



MEDAQLib

Micro-Epsilon
Data Acquisition Library

Application programming
interface (API) for
accessing digital sensors

Sensors

IFD2401	ILD1220	ILR110x/5x	DT306x/7x
IFD2421	ILD1302	ILR118x	DT3100
IFD2422	ILD1320	ILR1191	
IFD2431	ILD1401	ILR2250	DT6100
IFD2445	ILD1402		DT6120
IFD2451	ILD1420	ODC1202/20	DT62xx
IFD2461	ILD17xx	ODC2500	DT65xx
IFD2471	ILD1900	ODC2520	
	ILD22xx	ODC2600	KSS6380
ACS7000	ILD2300/10		KSS64xx
CFO100/200		MEBus	
MFA-7/14/21/28	IMC5400/10		
	IMC5600		

Interfaces

RS232	IF2001 (USB)	IF2008 (PCI+PCIe)	CSP2008
TCP/IP	IF2004 (PCI)	IF2008 (Ethernet)	C-Box
USB	IF2004 (USB)	IF1032 (Ethernet)	thicknessSENSOR

MEDAQLib

Micro-Epsilon Data Acqusition Library
X9751165

Application programming interface (API) for accessing digital sensors
V 4.11.1.31486

MICRO-EPSILON
MESSTECHNIK
GmbH & Co. KG
Königbacher Strasse 15

D-94496 Ortenburg

Tel. +49/8542/168-0
Fax +49/8542/168-90
e-mail info@micro-epsilon.de
www.micro-epsilon.de

Contents

1 Introduction	55
2 Installation	57
3 Accessing MEDAQLib in Visual C/C++	59
3.1 Setting up Visual Studio	59
3.1.1 Using MEDAQLib.lib, static approach	59
3.1.2 Using MEDAQLib.dll, dynamic approach	59
4 Using MEDAQLib	60
4.1 Main structure of a MEDAQLib program	60
4.2 How to call a function at sensor	61
4.3 Main structure in a LabView Sample	62
4.4 How to call a MEDAQLib function in LabView	63
4.5 How to decode signal order in TransferData or Poll returned data	63
4.6 Synchronized data from two ILD2300 at IF2008	64
4.7 Block based data acquisition vs Poll	66
4.7.1 Block based data acquisition	66
4.7.2 Polling	66
4.7.3 Additional hints	66
5 Samples	67
5.1 C-Sharp Example	67
5.2 Delphi Example	67
5.3 DLL Example	67
5.4 LabView	67
5.5 Lib Example	68
5.6 IF2008 Example	68
5.7 Unicode Example	68
5.8 VBA Example	68
5.9 VB2013Example	68
5.10 X64 Example	68
5.11 SensorFinder	68
5.12 SensorTest	69
6 Function Reference	70
6.1 Create a sensor instance (CreateSensorInst...)	70
6.2 Releasing sensor instance (ReleaseSensorInstance)	74
6.3 Set parameters (SetParameter...)	74
6.4 Get parameters (GetParameter...)	81
6.5 Clear internal parameter buffer (ClearAllParameters)	90
6.6 Connecting to the sensor (OpenSensor)	90
6.7 Closing connection to sensor (CloseSensor)	91
6.8 Sending commands to the sensor (SensorCommand)	91
6.9 Polling data from sensor (Poll)	93
6.10 Number of values available to read (DataAvail)	94
6.11 Block wise data acquisition from sensor (TransferData...)	95
6.12 Get additional error information (GetError...)	97
6.13 Get version of MEDAQLib dll (GetDLLVersion...)	100
6.14 EnableLogging wrapper function (EnableLogging...)	101
6.15 User logging functions (LogToFile...)	105

6.16	OpenSensor wrapper functions (OpenSensor...)	107
6.17	ExecSCmd wrapper functions (...ExecSCmd...)	115
7	Parameters	125
7.1	Naming conventions	125
7.1.1	Parameter prefix (IP_, SP_, SA_)	125
7.1.2	Parameter direction	125
7.1.3	Obligatory and optional parameters	125
7.1.4	Multiple commands at once	125
7.1.5	Multiple parameters at once	125
7.2	Interface parameters	125
7.2.1	All Interfaces	126
7.2.2	RS232	132
7.2.3	IF2004	136
7.2.4	IF2004_USB	138
7.2.5	IF2008	141
7.2.6	IF2008_ETH	144
7.2.7	TCP/IP	146
7.2.8	WinUSB	149
7.3	Sensor parameters	150
8	Sensor commands	151
8.1	Communication via SensorCommand	151
8.2	Sensor commands valid for each sensor	154
8.2.1	Get_SensorProperties	154
8.2.2	Get_OpenState	155
8.2.3	Enable.Logging	156
8.2.4	Automatic_Setup	158
8.2.5	Get_TransmittedDataInfo	158
8.2.6	Set_TransferredData	161
8.2.7	Get_TransferredData	161
8.2.8	Enable_ReferenceTimestamp	162
8.2.9	Use_Defaults	163
8.2.10	SettingsChanged	163
8.2.11	Get_DrvSetting	163
8.2.12	Get_HighResolutionTime	164
8.2.13	Exec_SleepTime	164
8.2.14	Cmd_Generic	164
8.2.15	Clear_Buffers	165
8.2.16	DataAvail_Event	165
8.2.17	Open_DataSocket	166
8.2.18	Set_Setup	168
8.2.19	Get_Setup	168
8.2.20	Update_... (Meta command)	170
8.2.21	Reset_... (Meta command)	171
8.3	SensorFinder commands	171
8.3.1	Start_FindSensor	174
8.3.2	Get_FindSensorProgress	178
8.3.3	Get_FoundSensor	179
8.3.4	Abort_FindSensor	189
8.3.5	Set_IPConfig	189

9 Commands for optoNCDT (ILR) sensors	194
9.1 Commands for ILR110x/5x	194
9.1.1 General commands	194
9.1.1.1 General	194
9.1.1.1.1 Get_Parameters (GAP)	194
9.1.1.1.2 Get_Energy (GDB)	198
9.1.1.1.3 Get_SerialNbr (GNR)	198
9.1.1.1.4 Get_ErrorStatus (GSI)	198
9.1.1.1.5 Get_Temperature (GTE)	199
9.1.1.1.6 Get_Version (GVE)	199
9.1.1.1.7 Set_Stand-by (ISB)	199
9.1.1.1.8 Set_VisibleLaser (IVL)	199
9.1.1.2 Triggering	200
9.1.1.2.1 Set_ContinousMode (ICM)	200
9.1.1.2.2 Exec_ContMeasure (ECM)	200
9.1.1.2.3 Trg_SingleMeasure (ESM)	200
9.1.1.3 Parameter management	200
9.1.1.3.1 Save_Parameters (EPW)	200
9.1.2 Measurement	200
9.1.2.1 General	200
9.1.2.1.1 Set_Offset (IDO)	200
9.1.3 Data output	201
9.1.3.1 Switching outputs	201
9.1.3.1.1 Set_HysteresisQ1 (IH1)	201
9.1.3.1.2 Set_HysteresisQ2 (IH2)	201
9.1.3.1.3 Set_LimitQ1-1 (IL1)	201
9.1.3.1.4 Set_LimitQ2-1 (IL2)	201
9.1.3.1.5 Set_LimitQ1-2 (IL4)	202
9.1.3.1.6 Set_LimitQ2-2 (IL5)	202
9.1.3.1.7 Set_ModeQ1 (IM1)	202
9.1.3.1.8 Set_ModeQ2 (IM2)	202
9.1.3.1.9 Set_NormQ1 (IN1)	203
9.1.3.1.10 Set_NormQ2 (IN2)	203
9.1.3.2 Analog output	203
9.1.3.2.1 Set_LimitQA-1 (IL3)	203
9.1.3.2.2 Set_LimitQA-2 (IL6)	203
9.1.3.2.3 Set_NormQA (INA)	204
9.2 Commands for ILR118x	204
9.2.1 General commands	204
9.2.1.1 General	204
9.2.1.1.1 Set_Autostart (AS)	204
9.2.1.1.2 Get_Autostart (AS)	205
9.2.1.1.3 Get_Info (ID)	205
9.2.1.1.4 Get_AllParameters (PA)	205
9.2.1.1.5 Get_Temperature (TP)	208
9.2.1.1.6 Laser_Off (LF)	209
9.2.1.1.7 Laser_On (LO)	209
9.2.1.2 Tracking	209
9.2.1.2.1 DistanceTracking (DT)	209
9.2.1.2.2 DistanceTracking7m (DS)	209
9.2.1.2.3 DistanceTracking10Hz (DW)	209
9.2.1.2.4 DistanceTracking50Hz (DX)	209
9.2.1.2.5 StopTracking (<ESC>)	209

9.2.1.3	Triggering	209
9.2.1.3.1	DistanceTriggered (DF)	209
9.2.1.3.2	DistanceMeasure (DM)	209
9.2.1.3.3	Set_MeasureTime (ST)	210
9.2.1.3.4	Get_MeasureTime (ST)	210
9.2.1.3.5	Set_TriggerDelay (TD)	210
9.2.1.3.6	Get_TriggerDelay (TD)	211
9.2.1.3.7	Set_TriggerMode (TM)	211
9.2.1.3.8	Get_TriggerMode (TM)	212
9.2.1.4	Interfaces	212
9.2.1.4.1	Set_Baudrate (BR)	212
9.2.1.4.2	Get_Baudrate (BR)	213
9.2.1.5	Parameter management	213
9.2.1.5.1	Reset_Parameters (PR)	213
9.2.2	Measurement	216
9.2.2.1	Measurement value processing	216
9.2.2.1.1	Set_ScaleFactor (SF)	216
9.2.2.1.2	Get_ScaleFactor (SF)	216
9.2.2.1.3	Set_Offset (OF)	217
9.2.2.1.4	Get_Offset (OF)	217
9.2.2.1.5	CurrentDistAsOffset (SO)	217
9.2.2.1.6	Set_AverageValue (SA)	217
9.2.2.1.7	Get_AverageValue (SA)	218
9.2.3	Data output	218
9.2.3.1	General	218
9.2.3.1.1	Set_OutputFormat (SD)	218
9.2.3.1.2	Get_OutputFormat (SD)	219
9.2.3.1.3	Set_RemovalMeasVal (RM)	219
9.2.3.1.4	Get_RemovalMeasVal (RM)	220
9.2.3.1.5	Set_ErrorMode (SE)	220
9.2.3.1.6	Get_ErrorMode (SE)	221
9.2.3.2	Switching outputs	221
9.2.3.2.1	Set_AlarmStart (AC)	221
9.2.3.2.2	Get_AlarmStart (AC)	221
9.2.3.2.3	Set_AlarmHysteresis (AH)	222
9.2.3.2.4	Get_AlarmHysteresis (AH)	222
9.2.3.2.5	Set_AlarmWidth (AW)	222
9.2.3.2.6	Get_AlarmWidth (AW)	223
9.2.3.3	Analog output	223
9.2.3.3.1	Set_RangeBegin4mA (RB)	223
9.2.3.3.2	Get_RangeBegin4mA (RB)	223
9.2.3.3.3	Set_RangeEnd20mA (RE)	223
9.2.3.3.4	Get_RangeEnd20mA (RE)	224
9.3	Commands for ILR1191	224
9.3.1	General commands	225
9.3.1.1	General	225
9.3.1.1.1	Set_Autostart (AS)	225
9.3.1.1.2	Get_Autostart (AS)	226
9.3.1.1.3	Set_PilotLaser (PL)	227
9.3.1.1.4	Get_PilotLaser (PL)	227
9.3.1.1.5	Get_Info (ID)	227
9.3.1.1.6	Get_AllParameters (PA)	228
9.3.1.1.7	Get_Temperature (TP)	232

9.3.1.1.8	Get_HWDiagnosis (HW)	232
9.3.1.1.9	Trigger_ColdStart (DR)	232
9.3.1.2	Tracking	232
9.3.1.2.1	DistanceTracking (DT)	232
9.3.1.2.2	SpeedTracking (VT)	232
9.3.1.2.3	StopTracking (<ESC>)	233
9.3.1.3	Triggering	233
9.3.1.3.1	DistanceTriggered (DF)	233
9.3.1.3.2	DistanceMeasure (DM)	233
9.3.1.3.3	SpeedMeasure (VM)	233
9.3.1.3.4	Set_TriggerDelay (TD)	233
9.3.1.3.5	Get_TriggerDelay (TD)	234
9.3.1.4	Interfaces	235
9.3.1.4.1	Set_Baudrate (BR)	235
9.3.1.4.2	Get_Baudrate (BR)	235
9.3.1.5	Parameter management	236
9.3.1.5.1	Reset_Parameters (PR)	236
9.3.2	Measurement	240
9.3.2.1	General	240
9.3.2.1.1	Set_MeasFreq (MF)	240
9.3.2.1.2	Get_MeasFreq (MF)	240
9.3.2.2	Measurement value processing	240
9.3.2.2.1	Set_ScaleFactor (SF)	240
9.3.2.2.2	Get_ScaleFactor (SF)	241
9.3.2.2.3	Set_Offset (OF)	241
9.3.2.2.4	Get_Offset (OF)	241
9.3.2.2.5	CurrentDistAsOffset (SO)	242
9.3.3	Data output	242
9.3.3.1	General	242
9.3.3.1.1	Set_OutputFormat (SD)	242
9.3.3.1.2	Get_OutputFormat (SD)	243
9.3.3.1.3	Set_TerminatingChar (TE)	243
9.3.3.1.4	Get_TerminatingChar (TE)	244
9.3.3.1.5	Set_ErrorMode (SE)	244
9.3.3.1.6	Get_ErrorMode (SE)	244
9.3.3.1.7	Set_MeasureWindow (MW)	245
9.3.3.1.8	Get_MeasureWindow (MW)	245
9.3.3.1.9	Set_AverageValue (SA)	246
9.3.3.1.10	Get_AverageValue (SA)	246
9.3.3.2	Switching outputs	246
9.3.3.2.1	Set_Out1Parameters (Q1)	246
9.3.3.2.2	Get_Out1Parameters (Q1)	247
9.3.3.2.3	Set_Out2Parameters (Q2)	248
9.3.3.2.4	Get_Out2Parameters (Q2)	249
9.3.3.3	Analog output	250
9.3.3.3.1	Set_AnalogOutLimits (QA)	250
9.3.3.3.2	Get_AnalogOutLimits (QA)	250
9.3.3.4	SSI	251
9.3.3.4.1	Set_FormatSSI (SC)	251
9.3.3.4.2	Get_FormatSSI (SC)	251
9.4	Commands for ILR2250	251
9.4.1	General commands	252
9.4.1.1	General	252

9.4.1.1.1	Get_Help (HELP)	252
9.4.1.1.2	Get_Info (GETINFO)	252
9.4.1.1.3	Get_Temperature (GETTEMP)	253
9.4.1.1.4	Get_OutputInfo_RS422 (GETOUTINFO_RS422)	253
9.4.1.1.5	Reset_Boot (RESET)	253
9.4.1.1.6	Reset_Counter (RESETCNT)	254
9.4.1.1.7	Get_AllParameters (PRINT)	254
9.4.1.2	Triggering	258
9.4.1.2.1	Set_TriggerMode (TRIGGER)	258
9.4.1.2.2	Get_TriggerMode (TRIGGER)	258
9.4.1.2.3	Set_TriggerMoment (TRIGGERAT)	258
9.4.1.2.4	Get_TriggerMoment (TRIGGERAT)	259
9.4.1.2.5	Set_TriggerLevel (TRIGGERLEVEL)	259
9.4.1.2.6	Get_TriggerLevel (TRIGGERLEVEL)	259
9.4.1.2.7	Set_TriggerCount (TRIGGERCOUNT)	259
9.4.1.2.8	Get_TriggerCount (TRIGGERCOUNT)	260
9.4.1.2.9	Software_Trigger (TRIGGERSW)	260
9.4.1.3	Interfaces	260
9.4.1.3.1	Set_Baudrate (BAUDRATE)	260
9.4.1.3.2	Get_Baudrate (BAUDRATE)	260
9.4.1.4	Parameter management	261
9.4.1.4.1	Save_InterfaceParameters (BASICSETTINGS STORE)	261
9.4.1.4.2	Load_InterfaceParameters (BASICSETTINGS READ)	261
9.4.1.4.3	Save_MeasureParameters (MEASSETTINGS STORE)	261
9.4.1.4.4	Load_MeasureParameters (MEASSETTINGS READ)	261
9.4.1.4.5	Get_MeasurePresetList (MEASSETTINGS PRESETLIST)	261
9.4.1.4.6	Set_MeasurePresetMode (MEASSETTINGS PRESETMODE)	261
9.4.1.4.7	Get_MeasurePresetMode (MEASSETTINGS PRESETMODE)	262
9.4.1.4.8	Set_Default (SETDEFAULT)	262
9.4.2	Measurement	262
9.4.2.1	General	262
9.4.2.1.1	Set_LaserPower (LASER)	262
9.4.2.1.2	Get_LaserPower (LASER)	263
9.4.2.2	Video signal	263
9.4.2.2.1	Set_ROI (ROI)	263
9.4.2.2.2	Get_ROI (ROI)	263
9.4.3	Data output	264
9.4.3.1	General	264
9.4.3.1.1	Set_DataOutInterface (OUTPUT)	264
9.4.3.1.2	Get_DataOutInterface (OUTPUT)	264
9.4.3.1.3	Set_HoldLastValid (OUTHOLD)	264
9.4.3.1.4	Get_HoldLastValid (OUTHOLD)	264
9.4.3.2	Selected measurement values	265
9.4.3.2.1	Set_OutputAdditional_RS422 (OUTADD_RS422)	265
9.4.3.2.2	Get_OutputAdditional_RS422 (OUTADD_RS422)	265
9.4.3.3	Switching outputs	265
9.4.3.3.1	Set_ErrorOutput<n> (ERRORROUT<n>)	265
9.4.3.3.2	Get_ErrorOutput<n> (ERRORROUT<n>)	265
9.4.3.3.3	Set_ErrorLevelOut<n> (ERRORLEVELOUT<n>)	266
9.4.3.3.4	Get_ErrorLevelOut<n> (ERRORLEVELOUT<n>)	266
9.4.3.3.5	Set_ErrorHysteresis<n> (ERRORHYSTERESIS<n>)	266
9.4.3.3.6	Get_ErrorHysteresis<n> (ERRORHYSTERESIS<n>)	266
9.4.3.3.7	Set_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)	267

9.4.3.3.8	Get_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)	267
9.4.3.3.9	Set_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)	267
9.4.3.3.10	Get_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)	267
9.4.3.3.11	Set_ErrorOutHoldTime (ERROROUTHOLD)	268
9.4.3.3.12	Get_ErrorOutHoldTime (ERROROUTHOLD)	268
9.4.3.4	Analog output	268
9.4.3.4.1	Set_AnalogScaleRange (ANALOGSCALERANGE)	268
9.4.3.4.2	Get_AnalogScaleRange (ANALOGSCALERANGE)	269
9.4.4	Internal commands	269
9.4.4.1	Get_FirmwareVersion	269
9.4.4.2	Prepare_UpdateFirmware	269
9.4.4.3	Start_UpdateFirmware	270
9.4.4.4	Get_UpdateFirmwareProgress	270
9.4.4.5	Generate_Firmware	270
10	Commands for optoNCDT (ILD) sensors	271
10.1	Commands for ILD1401	271
10.1.1	General commands	271
10.1.1.1	General	271
10.1.1.1.1	Get_Info (INFO)	271
10.1.1.1.2	Get_Version (VERSION)	272
10.1.1.1.3	Reset_Boot (BOOT)	273
10.1.2	Measurement	273
10.1.2.1	Set_Median (MEDIAN)	273
10.1.3	Data output	273
10.1.3.1	General	273
10.1.3.1.1	SaveLastMV (SAVELASTMV)	273
10.1.3.1.2	Set_OutputChannel (OUTPUTCHANNEL)	273
10.2	Commands for ILD1302 and ILD1402	274
10.2.1	General commands	274
10.2.1.1	General	274
10.2.1.1.1	Get_Info (GET_INFO)	274
10.2.1.1.2	Get_Settings (GET_SETTINGS)	277
10.2.1.1.3	Reset_Boot (RESET_BOOT)	280
10.2.1.2	User level	280
10.2.1.2.1	Set_KeyLock (SET_KEYLOCK)	280
10.2.1.3	Interfaces	281
10.2.1.3.1	Set_Baudrate (SET_BAUDRATE)	281
10.2.1.4	Parameter management	281
10.2.1.4.1	Set_Default (SET_DEFAULT)	281
10.2.1.4.2	Set_SaveSettingsMode (SET_SAVE_SETTINGS_MODE)	281
10.2.1.5	Internal controller commands	282
10.2.1.5.1	Laser_Off (LASER_OFF)	282
10.2.1.5.2	Laser_On (LASER_ON)	282
10.2.1.5.3	Set_ExtInputMode (SET_EXT_INPUT_MODE)	282
10.2.1.5.4	Get_CI_Mode (GET_CI_MODE)	282
10.2.1.5.5	Set_CI_Mode_1401 (SET_CI_MODE_1401)	282
10.2.1.5.6	Set_CI_Mode_1402 (SET_CI_MODE_1402)	283
10.2.1.5.7	Set_OperationMode (SET_MODE)	283
10.2.1.5.8	Set_LaserDiode<n> (SET_LD<n>)	283
10.2.1.5.9	Get_LaserDiodeError (GET_LDERROR)	283
10.2.1.5.10	Get_CurrentOfMonitor (GET_IMON)	284
10.2.2	Measurement	284

10.2.2.1	Set_Speed (SET_SCANRATE)	284
10.2.2.2	Set_PeakSearching (SET_PEAKSEARCHING)	284
10.2.2.3	Get_Video	284
10.2.2.4	Set_Threshold (SET_THRESHOLD)	285
10.2.2.5	Set_Av (SET_AV)	285
10.2.3	Data output	285
10.2.3.1	General	285
10.2.3.1.1	Dat_Out_Off (DAT_OUT_OFF)	285
10.2.3.1.2	Dat_Out_On (DAT_OUT_ON)	285
10.2.3.1.3	Set_ErrorHandler (SET_ANALOG_ERROR_HANDLER)	286
10.2.3.1.4	ASCII_Output (ASCII_OUTPUT)	286
10.2.3.1.5	Set_OutputType (SET_OUTPUT_CHANNEL)	286
10.2.3.1.6	Set_OutputMode (SET_OUTPUTMODE)	286
10.2.3.1.7	Set_OutputTime (SET_OUTPUTTIME_MS)	287
10.2.3.2	Analog output	287
10.2.3.2.1	Set_TeachValue (SET_TEACH_VALUE)	287
10.2.3.2.2	Reset_TeachValue (RESET_TEACH_VALUE)	287
10.3	Commands for ILD1220, ILD1320, ILD1420, ILD1750/6x and ILD1900	287
10.3.1	General commands	288
10.3.1.1	General	288
10.3.1.1.1	Get_Help (HELP)	288
10.3.1.1.2	Get_Info (GETINFO)	288
10.3.1.1.3	Get_OutputInfo_RS422 (GETOUTINFO_RS422)	290
10.3.1.1.4	Set_Unit (UNIT)	292
10.3.1.1.5	Get_Unit (UNIT)	292
10.3.1.1.6	Set_SyncMode (SYNC)	293
10.3.1.1.7	Get_SyncMode (SYNC)	293
10.3.1.1.8	Set_Termination (TERMINATION)	293
10.3.1.1.9	Get_Termination (TERMINATION)	294
10.3.1.1.10	Reset_Boot (RESET)	294
10.3.1.1.11	Reset_Counter (RESETCNT)	294
10.3.1.1.12	Set_Keylock (KEYLOCK)	295
10.3.1.1.13	Get_Keylock (KEYLOCK)	295
10.3.1.1.14	Set_KeyFunction (KEYFUNC)	296
10.3.1.1.15	Get_KeyFunction (KEYFUNC)	296
10.3.1.1.16	Set_Echo (ECHO)	296
10.3.1.1.17	Get_Echo (ECHO)	296
10.3.1.1.18	Get_AllParameters (PRINT)	297
10.3.1.2	User level	320
10.3.1.2.1	Logout (LOGOUT)	320
10.3.1.2.2	Login (LOGIN)	320
10.3.1.2.3	Get_UserLevel (GETUSERLEVEL)	320
10.3.1.2.4	Set_DefaultUser (STDUSER)	320
10.3.1.2.5	Get_DefaultUser (STDUSER)	320
10.3.1.2.6	Set_Password (PASSWD)	321
10.3.1.3	Triggering	321
10.3.1.3.1	Set_TriggerSource (TRIGGERSOURCE)	321
10.3.1.3.2	Get_TriggerSource (TRIGGERSOURCE)	321
10.3.1.3.3	Set_TriggerMode (TRIGGERMODE)	322
10.3.1.3.4	Get_TriggerMode (TRIGGERMODE)	322
10.3.1.3.5	Set_TriggerMode (TRIGGER)	322
10.3.1.3.6	Get_TriggerMode (TRIGGER)	323
10.3.1.3.7	Set_TriggerMoment (TRIGGERAT)	323

10.3.1.3.8	Get_TriggerMoment (TRIGGERAT)	323
10.3.1.3.9	Set_TriggerLevel (TRIGGERLEVEL)	324
10.3.1.3.10	Get_TriggerLevel (TRIGGERLEVEL)	324
10.3.1.3.11	Set_TriggerCount (TRIGGERCOUNT)	324
10.3.1.3.12	Get_TriggerCount (TRIGGERCOUNT)	324
10.3.1.3.13	Software_Trigger (TRIGGERSW)	325
10.3.1.3.14	Set_SoftwareTrigger (TRIGGERSW)	325
10.3.1.3.15	Get_SoftwareTrigger (TRIGGERSW)	325
10.3.1.4	Interfaces	325
10.3.1.4.1	Set_Baudrate (BAUDRATE)	325
10.3.1.4.2	Get_Baudrate (BAUDRATE)	326
10.3.1.4.3	Set_AppLanguage (LANGUAGE)	327
10.3.1.4.4	Get_AppLanguage (LANGUAGE)	327
10.3.1.5	Parameter management	327
10.3.1.5.1	Save_InterfaceParameters (BASICSETTINGS STORE)	327
10.3.1.5.2	Load_InterfaceParameters (BASICSETTINGS READ)	327
10.3.1.5.3	Save_MeasureParameters (MEASSETTINGS STORE)	327
10.3.1.5.4	Load_MeasureParameters (MEASSETTINGS READ)	327
10.3.1.5.5	Rename_MeasureParameters (MEASSETTINGS RENAME)	328
10.3.1.5.6	Get_CurrentMeasureSetting (MEASSETTINGS CURRENT)	328
10.3.1.5.7	Get_MeasureSettingsList (MEASSETTINGS LIST)	328
10.3.1.5.8	Delete_MeasureParameters (MEASSETTINGS DELETE)	328
10.3.1.5.9	Get_MeasurePresetList (MEASSETTINGS PRESETLIST)	328
10.3.1.5.10	Set_InitialMeasureSetting (MEASSETTINGS INITIAL)	329
10.3.1.5.11	Get_InitialMeasureSetting (MEASSETTINGS INITIAL)	329
10.3.1.5.12	Set_MeasurePresetMode (MEASSETTINGS PRESETMODE)	329
10.3.1.5.13	Get_MeasurePresetMode (MEASSETTINGS PRESETMODE)	330
10.3.1.5.14	Set_Default (SETDEFAULT)	330
10.3.1.5.15	Export_Parameters (EXPORT)	330
10.3.1.5.16	Import_Parameters (IMPORT)	331
10.3.1.6	Internal controller commands	331
10.3.1.6.1	Set_MultiFunctionInputMode (MFIFUNC)	331
10.3.1.6.2	Get_MultiFunctionInputMode (MFIFUNC)	332
10.3.1.6.3	Set_MultiFunctionInputLevel (MFILEVEL)	332
10.3.1.6.4	Get_MultiFunctionInputLevel (MFILEVEL)	332
10.3.1.6.5	Set_InitialChartType (CHARTTYPE_INITIAL)	333
10.3.1.6.6	Get_InitialChartType (CHARTTYPE_INITIAL)	333
10.3.1.6.7	Set_ChartType (CHARTTYPE)	333
10.3.1.6.8	Get_ChartType (CHARTTYPE)	334
10.3.1.6.9	Set_TargetMode (TARGETMODE)	334
10.3.1.6.10	Get_TargetMode (TARGETMODE)	334
10.3.1.6.11	Get_HTTP (GETHTTP)	335
10.3.2	Measurement	335
10.3.2.1	General	335
10.3.2.1.1	Set_MeasureMode (MEASMODE)	335
10.3.2.1.2	Get_MeasureMode (MEASMODE)	336
10.3.2.1.3	Set_MeasurePeak (MEASPEAK)	336
10.3.2.1.4	Get_MeasurePeak (MEASPEAK)	336
10.3.2.1.5	Set_PeakWidthMax (PEAK_WIDTHMAX)	337
10.3.2.1.6	Get_PeakWidthMax (PEAK_WIDTHMAX)	337
10.3.2.1.7	Set_PeakGlobalThreshold (PEAK_THRESGLOB)	337
10.3.2.1.8	Get_PeakGlobalThreshold (PEAK_THRESGLOB)	338
10.3.2.1.9	Set_ShutterMode (SHUTTERMODE)	338

10.3.2.1.10	Get_ShutterMode (SHUTTERMODE)	338
10.3.2.1.11	Set_Samplerate (MEASRATE)	339
10.3.2.1.12	Get_Samplerate (MEASRATE)	339
10.3.2.1.13	Set_ShutterTime (SHUTTER)	340
10.3.2.1.14	Get_ShutterTime (SHUTTER)	340
10.3.2.1.15	Set_ExposureMode (EXPOSUREMODE)	340
10.3.2.1.16	Get_ExposureMode (EXPOSUREMODE)	341
10.3.2.1.17	Set_ExposureLimit (EXPOSURELIMIT)	341
10.3.2.1.18	Get_ExposureLimit (EXPOSURELIMIT)	341
10.3.2.1.19	Set_LaserPower (LASERPOW)	342
10.3.2.1.20	Get_LaserPower (LASERPOW)	342
10.3.2.1.21	Get_VideoStreamSignal	342
10.3.2.2	Video signal	343
10.3.2.2.1	Set_ROI (ROI)	343
10.3.2.2.2	Get_ROI (ROI)	344
10.3.2.3	Measurement value processing	344
10.3.2.3.1	Set_Averaging (AVERAGE)	344
10.3.2.3.2	Get_Averaging (AVERAGE)	345
10.3.2.3.3	Set_MasterValue (MASTERMV)	346
10.3.2.3.4	Get_MasterValue (MASTERMV)	347
10.3.2.3.5	Set_MasterSignal (MASTERSIGNAL)	347
10.3.2.3.6	Get_MasterSignal (MASTERSIGNAL)	348
10.3.2.3.7	Reset_MasterSignal (MASTERSIGNAL)	348
10.3.2.3.8	Set_MasterSource (MASTERSOURCE)	349
10.3.2.3.9	Get_MasterSource (MASTERSOURCE)	349
10.3.2.3.10	Exec_Master (MASTER)	349
10.3.2.3.11	Get_Master (MASTER)	350
10.3.2.3.12	Set_Computation (COMP)	350
10.3.2.3.13	Get_Computation (COMP)	352
10.3.3	Data output	353
10.3.3.1	General	353
10.3.3.1.1	Set_DataOutInterface (OUTPUT)	353
10.3.3.1.2	Get_DataOutInterface (OUTPUT)	355
10.3.3.1.3	Set_ResamplingDevice (OUTREDUCEDEVICE)	356
10.3.3.1.4	Get_ResamplingDevice (OUTREDUCEDEVICE)	356
10.3.3.1.5	Set_ResamplingCount (OUTREDUCECOUNT)	357
10.3.3.1.6	Get_ResamplingCount (OUTREDUCECOUNT)	357
10.3.3.1.7	Set_HoldLastValid (OUTHOLD)	357
10.3.3.1.8	Get_HoldLastValid (OUTHOLD)	358
10.3.3.2	Selected measurement values	358
10.3.3.2.1	Set_Output_RS422 (OUT_RS422)	358
10.3.3.2.2	Get_Output_RS422 (OUT_RS422)	360
10.3.3.2.3	Set_OutputAdditional_RS422 (OUTADD_RS422)	363
10.3.3.2.4	Get_OutputAdditional_RS422 (OUTADD_RS422)	364
10.3.3.2.5	Set_OutputVideo_RS422 (OUTVIDEO_RS422)	365
10.3.3.2.6	Get_OutputVideo_RS422 (OUTVIDEO_RS422)	365
10.3.3.3	Switching outputs	366
10.3.3.3.1	Set_ErrorOutput1 (ERROROUT1)	366
10.3.3.3.2	Get_ErrorOutput1 (ERROROUT1)	366
10.3.3.3.3	Set_ErrorOutput2 (ERROROUT2)	366
10.3.3.3.4	Get_ErrorOutput2 (ERROROUT2)	367
10.3.3.3.5	Set_ErrorLimit (ERRORLIMIT)	367
10.3.3.3.6	Get_ErrorLimit (ERRORLIMIT)	367

10.3.3.3.7 Set_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)	368
10.3.3.3.8 Get_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)	368
10.3.3.3.9 Set_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)	369
10.3.3.3.10 Get_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)	369
10.3.3.3.11 Set_ErrorHysteresis (ERRORHYSTERESIS)	369
10.3.3.3.12 Get_ErrorHysteresis (ERRORHYSTERESIS)	369
10.3.3.3.13 Set_ErrorOutHoldTime (ERROROUTHOLD)	370
10.3.3.3.14 Get_ErrorOutHoldTime (ERROROUTHOLD)	370
10.3.3.3.15 Set_ErrorLevelOut1 (ERRORLEVELOUT1)	370
10.3.3.3.16 Get_ErrorLevelOut1 (ERRORLEVELOUT1)	370
10.3.3.3.17 Set_ErrorLevelOut2 (ERRORLEVELOUT2)	371
10.3.3.3.18 Get_ErrorLevelOut2 (ERRORLEVELOUT2)	371
10.3.3.4 Analog output	371
10.3.3.4.1 Set_AnalogRange (ANALOGRANGE)	371
10.3.3.4.2 Get_AnalogRange (ANALOGRANGE)	372
10.3.3.4.3 Set_AnalogScale (ANALOGSCALE)	372
10.3.3.4.4 Get_AnalogScale (ANALOGSCALE)	373
10.3.3.4.5 Set_AnalogScaleMode (ANALOGSCALEMODE)	373
10.3.3.4.6 Get_AnalogScaleMode (ANALOGSCALEMODE)	374
10.3.3.4.7 Set_AnalogScaleRange (ANALOGSCALERANGE)	374
10.3.3.4.8 Get_AnalogScaleRange (ANALOGSCALERANGE)	374
10.3.3.4.9 Set_AnalogScaleSource (ANALOGSCALESOURCE)	375
10.3.3.4.10 Get_AnalogScaleSource (ANALOGSCALESOURCE)	375
10.4 Commands for ILD17xx	375
10.4.1 General commands	376
10.4.1.1 General	376
10.4.1.1.1 Get_Info (GET_INFO)	376
10.4.1.1.2 Get_Settings (GET_SETTINGS)	379
10.4.1.1.3 Reset_Boot (RESET_BOOT)	382
10.4.1.2 User level	382
10.4.1.2.1 Set_KeyLock (SET_KEYLOCK)	382
10.4.1.3 Interfaces	383
10.4.1.3.1 Set_Baudrate (SET_BAUDRATE)	383
10.4.1.4 Parameter management	383
10.4.1.4.1 Set_Default (SET_DEFAULT)	383
10.4.1.4.2 WriteFlashZero (WriteFlashZero)	383
10.4.1.5 Internal controller commands	384
10.4.1.5.1 Laser_Off (LASER_OFF)	384
10.4.1.5.2 Laser_On (LASER_ON)	384
10.4.1.5.3 Set_Sync_TrgMode (SET_SYNCMODE/TRIGGERMODE)	384
10.4.1.5.4 Set_ErrorOutput (SET_ERROROUTPUT)	384
10.4.2 Measurement	384
10.4.2.1 Set_MeasureMode	384
10.4.2.2 Set_Speed (SET_SPEED)	385
10.4.2.3 Set_VideoMode	385
10.4.2.4 Get_Video	385
10.4.2.5 Set_Av0 (SET_AV0)	385
10.4.2.6 Set_Av1 (SET_AV1)	385
10.4.2.7 Set_Av2 (SET_AV2)	385
10.4.2.8 Set_Av3 (SET_AV3)	386
10.4.2.9 Set_AvX (SET_AVX)	386
10.4.2.10 Set_Av_T (SET_AV_T)	386
10.4.2.11 Get_MeasValue (GET_MEASVALUE)	386

10.4.3 Data output	386
10.4.3.1 General	386
10.4.3.1.1 Dat_Out_Off (DAT_OUT_OFF)	386
10.4.3.1.2 Dat_Out_On (DAT_OUT_ON)	387
10.4.3.1.3 Set_ErrorHandler (SET_ERRORHANDLER)	387
10.4.3.1.4 ASCII_Output (ASCII_OUTPUT)	387
10.4.3.1.5 Set_OutputType (SET_OUTPUTTYP)	387
10.4.3.2 Switching outputs	387
10.4.3.2.1 Set_Limits (SET_LIMITS)	387
10.4.3.2.2 Set_UpperLimit_F1 (SET_UPPERLIMIT_F1)	388
10.4.3.2.3 Set_LowerLimit_F1 (SET_LOWERLIMIT_F1)	388
10.4.3.3 Analog output	388
10.4.3.3.1 Zero (SET_ZERO)	388
10.5 Commands for ILD22xx	389
10.5.1 General commands	389
10.5.1.1 General	389
10.5.1.1.1 Get_Info (INFO)	389
10.5.1.1.2 Get_Settings (Get_Settings)	391
10.5.1.1.3 Get_Version	392
10.5.1.1.4 Reset_Boot (RESET)	392
10.5.1.2 User level	392
10.5.1.2.1 Set_KeyLock (SET_TASTENSPERRE)	392
10.5.1.3 Internal controller commands	393
10.5.1.3.1 Laser_Off (LASER_OFF)	393
10.5.1.3.2 Laser_On (LASER_ON)	393
10.5.2 Measurement	393
10.5.2.1 Set_Av0 (AVG 0)	393
10.5.2.2 Set_Av1 (AVG 1)	393
10.5.2.3 Set_Av2 (AVG 2)	393
10.5.2.4 Set_Av3 (AVG 3)	393
10.5.2.5 Set_AvX (AVG n)	393
10.5.2.6 Set_Av_T (AVGTYPE)	393
10.5.3 Data output	394
10.5.3.1 General	394
10.5.3.1.1 Dat_Out_Off (STOP)	394
10.5.3.1.2 Dat_Out_On (START)	394
10.5.3.2 Switching outputs	394
10.5.3.2.1 Transmit_Intensity	394
10.5.3.3 Analog output	394
10.5.3.3.1 Zero (ZERO)	394
10.6 Commands for ILD2300/10	394
10.6.1 General commands	395
10.6.1.1 General	395
10.6.1.1.1 Get_Help (HELP)	395
10.6.1.1.2 Get_Info (GETINFO)	395
10.6.1.1.3 Get_OutputInfo_RS422 (GETOUTINFO_RS422)	396
10.6.1.1.4 Get_OutputInfo_ETH (GETOUTINFO_ETH)	398
10.6.1.1.5 Set_SyncMode (SYNC)	400
10.6.1.1.6 Get_SyncMode (SYNC)	400
10.6.1.1.7 Set_Unit (UNIT)	401
10.6.1.1.8 Get_Unit (UNIT)	401
10.6.1.1.9 Reset_Boot (RESET)	401
10.6.1.1.10 Reset_Counter (RESETCNT)	401

10.6.1.1.11 Set_Echo (ECHO)	402
10.6.1.1.12 Get_Echo (ECHO)	402
10.6.1.1.13 Get_AllParameters (PRINT)	402
10.6.1.2 User level	416
10.6.1.2.1 Logout (LOGOUT)	416
10.6.1.2.2 Login (LOGIN)	416
10.6.1.2.3 Get_UserLevel (GETUSERLEVEL)	416
10.6.1.2.4 Set_DefaultUser (STDUSER)	416
10.6.1.2.5 Get_DefaultUser (STDUSER)	416
10.6.1.2.6 Set_Password (PASSWD)	417
10.6.1.3 Triggering	417
10.6.1.3.1 Set_TriggerMode (TRIGGER)	417
10.6.1.3.2 Get_TriggerMode (TRIGGER)	417
10.6.1.3.3 Set_TriggerMoment (TRIGGERAT)	418
10.6.1.3.4 Get_TriggerMoment (TRIGGERAT)	418
10.6.1.3.5 Set_TriggerLevel (TRIGGERLEVEL)	418
10.6.1.3.6 Get_TriggerLevel (TRIGGERLEVEL)	418
10.6.1.3.7 Set_TriggerCount (TRIGGERCOUNT)	419
10.6.1.3.8 Get_TriggerCount (TRIGGERCOUNT)	419
10.6.1.3.9 Software_Trigger (TRIGGERSW)	419
10.6.1.3.10 Set_TriggerOutput (TRIGGEROUT)	419
10.6.1.3.11 Get_TriggerOutput (TRIGGEROUT)	419
10.6.1.4 Interfaces	420
10.6.1.4.1 Set_IPConfiguration (IPCONFIG)	420
10.6.1.4.2 Get_IPConfiguration (IPCONFIG)	420
10.6.1.4.3 Set_IPDataTransferMode (MEATRANSFER)	421
10.6.1.4.4 Get_IPDataTransferMode (MEATRANSFER)	422
10.6.1.4.5 Set_EthernetMode (ETHERMODE)	422
10.6.1.4.6 Get_EthernetMode (ETHERMODE)	422
10.6.1.4.7 Set_Baudrate (BAUDRATE)	423
10.6.1.4.8 Get_Baudrate (BAUDRATE)	423
10.6.1.5 Parameter management	424
10.6.1.5.1 Save_Parameters (STORE)	424
10.6.1.5.2 Load_Parameters (READ)	424
10.6.1.5.3 Get_ActiveParameterSet (READ)	424
10.6.1.5.4 Set_Default (SETDEFAULT)	425
10.6.2 Measurement	425
10.6.2.1 General	425
10.6.2.1.1 Set_MeasureMode (MEASMODE)	425
10.6.2.1.2 Get_MeasureMode (MEASMODE)	425
10.6.2.1.3 Set_MeasurePeak (MEASPEAK)	426
10.6.2.1.4 Get_MeasurePeak (MEASPEAK)	426
10.6.2.1.5 Get_Video (GETVIDEO)	426
10.6.2.1.6 Set_ShutterMode (SHUTTERMODE)	427
10.6.2.1.7 Get_ShutterMode (SHUTTERMODE)	427
10.6.2.1.8 Set_Samplerate (MEASRATE)	428
10.6.2.1.9 Get_Samplerate (MEASRATE)	428
10.6.2.1.10 Set_ShutterTime (SHUTTER)	428
10.6.2.1.11 Get_ShutterTime (SHUTTER)	429
10.6.2.1.12 Set_ShutterFactor (SHUTTERFACTOR)	429
10.6.2.1.13 Get_ShutterFactor (SHUTTERFACTOR)	429
10.6.2.1.14 Set_LaserPower (LASERPOW)	429
10.6.2.1.15 Get_LaserPower (LASERPOW)	430

10.6.2.2	Video signal	430
10.6.2.2.1	Set_ROI (ROI)	430
10.6.2.2.2	Get_ROI (ROI)	430
10.6.2.2.3	Set_VideoAverage (VSAVERAGE)	431
10.6.2.2.4	Get_VideoAverage (VSAVERAGE)	431
10.6.2.2.5	Set_Threshold (THRESHOLD)	431
10.6.2.2.6	Get_Threshold (THRESHOLD)	432
10.6.2.3	Material database	432
10.6.2.3.1	Get_MaterialTable (MATERIALTABLE)	432
10.6.2.3.2	Set_ActiveMaterial (MATERIAL)	433
10.6.2.3.3	Get_ActiveMaterial (MATERIAL)	433
10.6.2.3.4	Get_MaterialInfo (MATERIALINFO)	433
10.6.2.3.5	Edit_Material (MATERIALEDIT)	433
10.6.2.3.6	Delete_Material (MATERIALDELETE)	434
10.6.2.3.7	Clear_MaterialTable	434
10.6.2.4	Measurement value processing	434
10.6.2.4.1	Set_Averaging (AVERAGE)	434
10.6.2.4.2	Get_Averaging (AVERAGE)	435
10.6.2.4.3	Set_SpikeCorrection (SPIKECORR)	436
10.6.2.4.4	Get_SpikeCorrection (SPIKECORR)	436
10.6.2.4.5	Set_StatisticDepth (STATISTICDEPTH)	437
10.6.2.4.6	Get_StatisticDepth (STATISTICDEPTH)	437
10.6.2.4.7	Reset_Statistic (RESETSTATISTIC)	437
10.6.2.4.8	Set_MasterValue (MASTERMV)	438
10.6.2.4.9	Get_MasterValue (MASTERMV)	438
10.6.3	Data output	438
10.6.3.1	General	438
10.6.3.1.1	Set_DataOutInterface (OUTPUT)	438
10.6.3.1.2	Get_DataOutInterface (OUTPUT)	439
10.6.3.1.3	Set_Resampling (OUTREDUCE)	439
10.6.3.1.4	Get_Resampling (OUTREDUCE)	440
10.6.3.1.5	Set_HoldLastValid (OUTHOLD)	440
10.6.3.1.6	Get_HoldLastValid (OUTHOLD)	440
10.6.3.1.7	Set_MeasureValueCnt (GETVALUE)	441
10.6.3.1.8	Get_MeasureValueCnt (GETVALUE)	441
10.6.3.2	Selected measurement values	441
10.6.3.2.1	Set_OutputDistance_RS422 (OUTDIST_RS422)	441
10.6.3.2.2	Get_OutputDistance_RS422 (OUTDIST_RS422)	441
10.6.3.2.3	Set_OutputThickness_RS422 (OUTTHICK_RS422)	442
10.6.3.2.4	Get_OutputThickness_RS422 (OUTTHICK_RS422)	442
10.6.3.2.5	Set_OutputStatistic_RS422 (OUTSTATISTIC_RS422)	442
10.6.3.2.6	Get_OutputStatistic_RS422 (OUTSTATISTIC_RS422)	443
10.6.3.2.7	Set_OutputStatistic_ETH (OUTSTATISTIC_ETH)	443
10.6.3.2.8	Get_OutputStatistic_ETH (OUTSTATISTIC_ETH)	444
10.6.3.2.9	Set_OutputAdditional_RS422 (OUTADD_RS422)	444
10.6.3.2.10	Get_OutputAdditional_RS422 (OUTADD_RS422)	445
10.6.3.2.11	Set_OutputAdditional_ETH (OUTADD_ETH)	446
10.6.3.2.12	Get_OutputAdditional_ETH (OUTADD_ETH)	447
10.6.3.2.13	Set_OutputVideo_ETH (OUTVIDEO)	448
10.6.3.2.14	Get_OutputVideo_ETH (OUTVIDEO)	448

11 Commands for confocalDT (IFD) sensors	449
11.1 Commands for IFD2401 and IFD2431	449
11.1.1 General commands	449
11.1.1.1 Get_Status (STS)	449
11.1.1.2 Get_Version (VER)	451
11.1.1.3 Reset (RST)	452
11.1.1.2 Sensor	452
11.1.1.2.1 Set_ActiveSensor (SEN)	452
11.1.1.2.2 Get_ActiveSensor (SEN?)	452
11.1.1.2.3 Get_Range (SCA)	452
11.1.1.2.4 Get_AllRanges (LUL)	452
11.1.1.2.5 Acquire_DarkSig (DRK)	453
11.1.1.2.6 FastDark (FDK)	453
11.1.1.2.7 Set_AutoDark (ADK)	453
11.1.1.2.8 Get_AutoDark (ADK?)	454
11.1.1.3 Triggering	454
11.1.1.3.1 Continue (CTN)	454
11.1.1.3.2 Start_Trigger (TRG)	454
11.1.1.3.3 End_Trigger	454
11.1.1.3.4 SingleShot_Trg (TRE)	454
11.1.1.3.5 Set_TrgMode_Edge (TRS)	454
11.1.1.3.6 Set_TrgMode_State (TRN)	455
11.1.1.3.7 Set_ActiveEdge (TRF)	455
11.1.1.3.8 Get_ActiveEdge (TRF?)	455
11.1.1.3.9 Software_Trigger (STR)	455
11.1.1.4 Encoder	455
11.1.1.4.1 RecenterEncoder (RCD)	455
11.1.1.5 Interfaces	456
11.1.1.5.1 Set_Baudrate (BAU)	456
11.1.1.5.2 Get_Baudrate (BAU?)	456
11.1.1.6 Parameter management	457
11.1.1.6.1 Save_Setup (SSU)	457
11.1.1.7 Internal controller commands	457
11.1.1.7.1 Set_LampTest (SLP)	457
11.1.1.7.2 Get_LampTest (SLP?)	457
11.1.1.7.3 Set_LampTestThr (CSL)	457
11.1.1.7.4 Get_LampTestThr (CSL?)	458
11.1.1.7.5 Set_Watchdog (WDE)	458
11.1.1.7.6 Get_Watchdog (WDE?)	458
11.1.1.7.7 Set_WatchdogPrd (WDP)	458
11.1.1.7.8 Get_WatchdogPrd (WDP?)	459
11.1.2 Measurement	459
11.1.2.1 General	459
11.1.2.1.1 Set_MeasureMode (MOD)	459
11.1.2.1.2 Get_MeasureMode (MOD?)	459
11.1.2.1.3 Set_FirstPeakMode (MSP)	459
11.1.2.1.4 Get_FirstPeakMode (MSP?)	460
11.1.2.1.5 Set_SRIndex (SRA)	460
11.1.2.1.6 Get_SRIndex (SRA?)	460
11.1.2.1.7 Set_FreeSR (FRQ)	461
11.1.2.1.8 Get_FreeSR (FRQ?)	461
11.1.2.1.9 Set_Exposure (TEX)	461

11.1.2.1.10	Get_Exposure (TEX?)	462
11.1.2.1.11	Get_MinSR (FRM)	462
11.1.2.1.12	Set_DoubleFreq (DFA)	462
11.1.2.1.13	Get_DoubleFreq (DFA?)	462
11.1.2.1.14	Set_Frequencies (DFF)	463
11.1.2.1.15	Get_Frequencies (DFF?)	463
11.1.2.1.16	Get_CCD (CCD)	463
11.1.2.1.17	Get_DarkSig (SGD)	464
11.1.2.1.18	Start_Spectrum	464
11.1.2.1.19	Get_Spectrum	465
11.1.2.1.20	End_Spectrum	466
11.1.2.2	Video signal	466
11.1.2.2.1	Set_Threshold (MNP)	466
11.1.2.2.2	Get_Threshold (MNP?)	466
11.1.2.2.3	Set_Threshold1 (SPP)	466
11.1.2.2.4	Get_Threshold1 (SPP?)	467
11.1.2.2.5	Set_Threshold2 (SDP)	467
11.1.2.2.6	Get_Threshold2 (SDP?)	467
11.1.2.2.7	Set_SpectralAv (AVS)	467
11.1.2.2.8	Get_SpectralAv (AVS?)	468
11.1.2.2.9	Set_Binning (FBH)	468
11.1.2.2.10	Get_Binning (FBH?)	468
11.1.2.3	Material database	468
11.1.2.3.1	Set_RefraclIndex (SRI)	468
11.1.2.3.2	Get_RefraclIndex (SRI?)	469
11.1.2.3.3	Set_RefIdxFile (INF)	469
11.1.2.3.4	Get_RefIdxFile (INF?)	470
11.1.2.4	Light source	470
11.1.2.4.1	Set_LEDIntensity (LED)	470
11.1.2.4.2	Get_LEDIntensity (LED?)	471
11.1.2.4.3	Set_AutoAdaptLED (AAL)	471
11.1.2.4.4	Get_AutoAdaptLED (AAL?)	471
11.1.2.4.5	Set_AdaptLEDTThr (VTH)	471
11.1.2.4.6	Get_AdaptLEDTThr (VTH?)	472
11.1.2.4.7	Set_IntLightSrc (CCL)	472
11.1.2.4.8	Get_IntLightSrc (CCL?)	472
11.1.2.5	Measurement value processing	472
11.1.2.5.1	Set_Averaging (AVR)	472
11.1.2.5.2	Get_Averaging (AVR?)	473
11.1.3	Data output	473
11.1.3.1	General	473
11.1.3.1.1	Set_IntensityMode (DFI)	473
11.1.3.1.2	Get_IntensityMode (DFI?)	473
11.1.3.1.3	Set_HoldLastValid (HLV)	473
11.1.3.1.4	Get_HoldLastValid (HLV?)	474
11.1.3.1.5	Set_MissingSignal (RSP)	474
11.1.3.1.6	Get_MissingSignal (RSP?)	474
11.1.3.1.7	Set_Reverse (RVS)	474
11.1.3.1.8	Get_Reverse (RVS?)	474
11.1.3.1.9	Set_Ascii (ASC)	475
11.1.3.1.10	Set_Binary (BIN)	475
11.1.3.2	Selected measurement values	475
11.1.3.2.1	Set_OutputData (SOD)	475

11.1.3.2.2	Get_OutputData (SOD?)	475
11.1.3.3	Analog output	476
11.1.3.3.1	Set_AnalogOut (ANA)	476
11.1.3.3.2	Get_AnalogOut (ANA?)	477
11.1.3.3.3	Set_AnalogZero (SOF)	478
11.1.4	Internal commands	478
11.1.4.1	Set_DataScale (CEE)	478
11.1.4.2	Get_DataScale (CEE?)	478
11.1.4.3	Set_BarycenterSca (CEB)	478
11.1.4.4	Get_BarycenterSca (CEB?)	479
11.1.4.5	Set_BarycenterOff (CRB)	479
11.1.4.6	Get_BarycenterOff (CRB?)	479
11.1.4.7	Upload_RefIdxFile	479
11.1.4.8	Get_WhiteRef (SGW)	480
11.1.4.9	Get_NormSig (SGN)	480
11.1.4.10	Get_CalibTable (SGC)	480
11.1.4.11	Upload_CalibTable	481
11.2	Commands for IFD2421 and IFD2422	481
11.2.1	General commands	482
11.2.1.1	General	482
11.2.1.1.1	Get_Help (HELP)	482
11.2.1.1.2	Get_Info (GETINFO)	482
11.2.1.1.3	Get_ChannelCount (GETCHANNELCNT)	483
11.2.1.1.4	Get_OutputInfo_RS422 (GETOUTINFO_RS422)	483
11.2.1.1.5	Get_OutputInfo_ETH (GETOUTINFO_ETH)	487
11.2.1.1.6	Set_Unit (UNIT)	491
11.2.1.1.7	Get_Unit (UNIT)	491
11.2.1.1.8	Set_Echo (ECHO)	491
11.2.1.1.9	Get_Echo (ECHO)	492
11.2.1.1.10	Get_AllParameters (PRINT)	492
11.2.1.1.11	Set_SyncMode (SYNC)	530
11.2.1.1.12	Get_SyncMode (SYNC)	530
11.2.1.1.13	Set_Termination (TERMINATION)	530
11.2.1.1.14	Get_Termination (TERMINATION)	531
11.2.1.1.15	Reset_Boot (RESET)	531
11.2.1.1.16	Reset_Counter (RESETCNT)	531
11.2.1.1.17	Set_Keylock (KEYLOCK)	531
11.2.1.1.18	Get_Keylock (KEYLOCK)	532
11.2.1.1.19	Set_KeyFunction1 (KEYFUNC1)	532
11.2.1.1.20	Get_KeyFunction1 (KEYFUNC1)	533
11.2.1.1.21	Set_KeyFunction2 (KEYFUNC2)	533
11.2.1.1.22	Get_KeyFunction2 (KEYFUNC2)	534
11.2.1.1.23	Set_KeyMasterSignalSelect (KEYMASTERSIGNALSELECT)	534
11.2.1.1.24	Get_KeyMasterSignalSelect (KEYMASTERSIGNALSELECT)	535
11.2.1.2	User level	535
11.2.1.2.1	Logout (LOGOUT)	535
11.2.1.2.2	Login (LOGIN)	536
11.2.1.2.3	Get_UserLevel (GETUSERLEVEL)	536
11.2.1.2.4	Set_DefaultUser (STDUSER)	536
11.2.1.2.5	Get_DefaultUser (STDUSER)	536
11.2.1.2.6	Set_Password (PASSWD)	536
11.2.1.3	Sensor	537
11.2.1.3.1	Get_SensorTable (SENSORTABLE)	537

11.2.1.3.2	Get_SensorTable_Ch<n> (SENSORTABLE_CH0<n>)	538
11.2.1.3.3	Set_ActiveSensor (SENSORHEAD)	539
11.2.1.3.4	Get_ActiveSensor (SENSORHEAD)	539
11.2.1.3.5	Set_ActiveSensor_Ch<n> (SENSORHEAD_CH0<n>)	539
11.2.1.3.6	Get_ActiveSensor_Ch<n> (SENSORHEAD_CH0<n>)	539
11.2.1.3.7	Get_SensorInfo (SENSORINFO)	540
11.2.1.3.8	Get_SensorInfo_Ch<n> (SENSORINFO_CH0<n>)	540
11.2.1.3.9	Set_LEDControl (LED)	541
11.2.1.3.10	Get_LEDControl (LED)	541
11.2.1.3.11	Set_LEDControl_Ch<n> (LED_CH0<n>)	541
11.2.1.3.12	Get_LEDControl_Ch<n> (LED_CH0<n>)	542
11.2.1.3.13	DarkCorr (DARKCORR)	542
11.2.1.3.14	DarkCorr_Ch<n> (DARKCORR_CH0<n>)	542
11.2.1.4	Triggering	542
11.2.1.4.1	Set_TriggerSource (TRIGGERSOURCE)	542
11.2.1.4.2	Get_TriggerSource (TRIGGERSOURCE)	543
11.2.1.4.3	Set_TriggerMode (TRIGGERMODE)	543
11.2.1.4.4	Get_TriggerMode (TRIGGERMODE)	543
11.2.1.4.5	Set_TriggerMoment (TRIGGERAT)	543
11.2.1.4.6	Get_TriggerMoment (TRIGGERAT)	544
11.2.1.4.7	Set_TriggerInLogic (TRIGINLEVEL)	544
11.2.1.4.8	Get_TriggerInLogic (TRIGINLEVEL)	544
11.2.1.4.9	Set_TriggerLevel (TRIGGERLEVEL)	544
11.2.1.4.10	Get_TriggerLevel (TRIGGERLEVEL)	545
11.2.1.4.11	Set_TriggerCount (TRIGGERCOUNT)	545
11.2.1.4.12	Get_TriggerCount (TRIGGERCOUNT)	545
11.2.1.4.13	Software_Trigger (TRIGGERSW)	545
11.2.1.4.14	Set_TriggerEncoderMin (TRIGGERENCMIN)	545
11.2.1.4.15	Get_TriggerEncoderMin (TRIGGERENCMIN)	546
11.2.1.4.16	Set_TriggerEncoderMax (TRIGGERENCMAX)	546
11.2.1.4.17	Get_TriggerEncoderMax (TRIGGERENCMAX)	546
11.2.1.4.18	Set_TriggerEncoderStepsize (TRIGGERENCSTEPSIZE)	546
11.2.1.4.19	Get_TriggerEncoderStepsize (TRIGGERENCSTEPSIZE)	547
11.2.1.5	Encoder	547
11.2.1.5.1	Set_EncoderInterpolation<n> (ENCINTERPOL<n>)	547
11.2.1.5.2	Get_EncoderInterpolation<n> (ENCINTERPOL<n>)	547
11.2.1.5.3	Set_EncoderMode<n> (ENCREF<n>)	547
11.2.1.5.4	Get_EncoderMode<n> (ENCREF<n>)	548
11.2.1.5.5	Set_EncoderPreload<n> (ENCVALUE<n>)	548
11.2.1.5.6	Get_EncoderPreload<n> (ENCVALUE<n>)	548
11.2.1.5.7	Load_Encoder<n> (ENCSET)	548
11.2.1.5.8	EnableRef_Encoder<n> (ENCRESET)	548
11.2.1.5.9	Set_Encoder.MaxValue<n> (ENCMAX<n>)	549
11.2.1.5.10	Get_Encoder.MaxValue<n> (ENCMAX<n>)	549
11.2.1.6	Interfaces	549
11.2.1.6.1	Set_IPConfiguration (IPCONFIG)	549
11.2.1.6.2	Get_IPConfiguration (IPCONFIG)	550
11.2.1.6.3	Set_IPDataTransferMode (MEATRANSFER)	550
11.2.1.6.4	Get_IPDataTransferMode (MEATRANSFER)	551
11.2.1.6.5	Set_TCPKeepAlive (TCPKEEPALIVE)	551
11.2.1.6.6	Get_TCPKeepAlive (TCPKEEPALIVE)	552
11.2.1.6.7	Set_EthernetMode (ETHERMODE)	552
11.2.1.6.8	Get_EthernetMode (ETHERMODE)	552

11.2.1.6.9	Set_Baudrate (BAUDRATE)	552
11.2.1.6.10	Get_Baudrate (BAUDRATE)	553
11.2.1.6.11	Set_AppLanguage (LANGUAGE)	553
11.2.1.6.12	Get_AppLanguage (LANGUAGE)	553
11.2.1.7	Parameter management	554
11.2.1.7.1	Save_InterfaceParameters (BASICSETTINGS STORE)	554
11.2.1.7.2	Load_InterfaceParameters (BASICSETTINGS READ)	554
11.2.1.7.3	Save_MeasureParameters (MEASSETTINGS STORE)	554
11.2.1.7.4	Load_MeasureParameters (MEASSETTINGS READ)	554
11.2.1.7.5	Rename_MeasureParameters (MEASSETTINGS RENAME)	554
11.2.1.7.6	Get_CurrentMeasureSetting (MEASSETTINGS CURRENT)	555
11.2.1.7.7	Get_MeasureSettingsList (MEASSETTINGS LIST)	555
11.2.1.7.8	Delete_MeasureParameters (MEASSETTINGS DELETE)	555
11.2.1.7.9	Get_MeasurePresetList (MEASSETTINGS PRESETLIST)	555
11.2.1.7.10	Set_InitialMeasureSetting (MEASSETTINGS INITIAL)	555
11.2.1.7.11	Get_InitialMeasureSetting (MEASSETTINGS INITIAL)	556
11.2.1.7.12	Set_MeasurePresetMode (MEASSETTINGS PRESETMODE)	556
11.2.1.7.13	Get_MeasurePresetMode (MEASSETTINGS PRESETMODE)	556
11.2.1.7.14	Set_Default (SETDEFAULT)	556
11.2.1.7.15	Export_Parameters (EXPORT)	557
11.2.1.7.16	Import_Parameters (IMPORT)	557
11.2.1.8	Internal controller commands	558
11.2.1.8.1	Set_ChartType (CHARTTYPE)	558
11.2.1.8.2	Get_ChartType (CHARTTYPE)	558
11.2.2	Measurement	558
11.2.2.1	General	558
11.2.2.1.1	Set_MeasurePeak (MEASPEAK)	558
11.2.2.1.2	Get_MeasurePeak (MEASPEAK)	559
11.2.2.1.3	Set_MeasurePeak_Ch<n> (MEASPEAK_CH0<n>)	559
11.2.2.1.4	Get_MeasurePeak_Ch<n> (MEASPEAK_CH0<n>)	559
11.2.2.1.5	Set_ShutterMode (SHUTTERMODE)	560
11.2.2.1.6	Get_ShutterMode (SHUTTERMODE)	560
11.2.2.1.7	Set_ShutterMode_Ch<n> (SHUTTERMODE_CH0<n>)	560
11.2.2.1.8	Get_ShutterMode_Ch<n> (SHUTTERMODE_CH0<n>)	561
11.2.2.1.9	Set_ShutterTime (SHUTTER)	561
11.2.2.1.10	Get_ShutterTime (SHUTTER)	561
11.2.2.1.11	Set_ShutterTime_Ch<n> (SHUTTER_CH0<n>)	562
11.2.2.1.12	Get_ShutterTime_Ch<n> (SHUTTER_CH0<n>)	562
11.2.2.1.13	Set_Samplerate (MEASRATE)	563
11.2.2.1.14	Get_Samplerate (MEASRATE)	563
11.2.2.1.15	Get_VideoStreamSignal	563
11.2.2.1.16	Get_DarkCorrTable (DARKCORR_PRINT)	565
11.2.2.1.17	Get_DarkCorrTable_Ch<n> (DARKCORR_PRINT_CH0<n>)	565
11.2.2.1.18	Get_LightCorrTable (LIGHTCORR_PRINT)	565
11.2.2.1.19	Get_LightCorrTable_Ch<n> (LIGHTCORR_PRINT_CH0<n>)	565
11.2.2.2	Video signal	566
11.2.2.2.1	Set_ROI (ROI)	566
11.2.2.2.2	Get_ROI (ROI)	566
11.2.2.2.3	Set_ROI_Ch<n> (ROI_CH0<n>)	566
11.2.2.2.4	Get_ROI_Ch<n> (ROI_CH0<n>)	567
11.2.2.3	Material database	567
11.2.2.3.1	Get_MaterialTable (MATERIALTABLE)	567
11.2.2.3.2	Set_ActiveMaterial (MATERIAL)	568

11.2.2.3.3	Get_ActiveMaterial (MATERIAL)	569
11.2.2.3.4	Set_ActiveMaterial_Ch<n> (MATERIAL_CH0<n>)	569
11.2.2.3.5	Get_ActiveMaterial_Ch<n> (MATERIAL_CH0<n>)	569
11.2.2.3.6	Get_MaterialInfo (MATERIALINFO)	569
11.2.2.3.7	Get_MaterialInfo_Ch<n> (MATERIALINFO_CH0<n>)	570
11.2.2.3.8	Edit_Material_Abbe (MATERIALEDIT)	571
11.2.2.3.9	Edit_Material_Nx (MATERIALEDIT)	572
11.2.2.3.10	Delete_Material (MATERIALDELETE)	572
11.2.2.3.11	Clear_MaterialTable	573
11.2.2.4	Peak processing	573
11.2.2.4.1	Set_RefractiveCorrection (REFRACCORR)	573
11.2.2.4.2	Get_RefractiveCorrection (REFRACCORR)	573
11.2.2.4.3	Set_RefractiveCorrection_Ch<n> (REFRACCORR_CH0<n>) .	573
11.2.2.4.4	Get_RefractiveCorrection_Ch<n> (REFRACCORR_CH0<n>) .	574
11.2.2.4.5	Set_PeakCount (PEAKCOUNT)	574
11.2.2.4.6	Get_PeakCount (PEAKCOUNT)	574
11.2.2.4.7	Set_PeakCount_Ch<n> (PEAKCOUNT_CH0<n>)	574
11.2.2.4.8	Get_PeakCount_Ch<n> (PEAKCOUNT_CH0<n>)	575
11.2.2.4.9	Set_MaterialMultiPeak (MATERIALMP)	575
11.2.2.4.10	Get_MaterialMultiPeak (MATERIALMP)	575
11.2.2.4.11	Set_MaterialMultiPeak_Ch<n> (MATERIALMP_CH0<n>) .	576
11.2.2.4.12	Get_MaterialMultiPeak_Ch<n> (MATERIALMP_CH0<n>) .	577
11.2.2.4.13	Set_MinimumThreshold (MIN_THRESHOLD)	577
11.2.2.4.14	Get_MinimumThreshold (MIN_THRESHOLD)	577
11.2.2.4.15	Set_MinimumThreshold_Ch<n> (MIN_THRESHOLD_CH0<n>) .	578
11.2.2.4.16	Get_MinimumThreshold_Ch<n> (MIN_THRESHOLD_CH0<n>) .	578
11.2.2.4.17	Set_PeakModulation (PEAK_MODULATION)	578
11.2.2.4.18	Get_PeakModulation (PEAK_MODULATION)	578
11.2.2.4.19	Set_PeakModulation_Ch<n> (PEAK_MODULATION_CH01) .	579
11.2.2.4.20	Get_PeakModulation_Ch<n> (PEAK_MODULATION_CH01) .	579
11.2.2.5	Measurement value processing	579
11.2.2.5.1	Set_SpikeCorrection (SPIKECORR)	579
11.2.2.5.2	Get_SpikeCorrection (SPIKECORR)	580
11.2.2.5.3	Set_SpikeCorrection_Ch<n> (SPIKECORR_CH0<n>) .	581
11.2.2.5.4	Get_SpikeCorrection_Ch<n> (SPIKECORR_CH0<n>) .	581
11.2.2.5.5	Set_StatisticSignal (STATISTICSIGNAL)	582
11.2.2.5.6	Get_StatisticSignal (STATISTICSIGNAL)	583
11.2.2.5.7	Reset_Statistic (RESETSTATISTIC)	584
11.2.2.5.8	Reset_StatisticSignal (STATISTIC RESET)	584
11.2.2.5.9	Set_MasterSignal (MASTERSIGNAL)	585
11.2.2.5.10	Get_MasterSignal (MASTERSIGNAL)	585
11.2.2.5.11	Reset_MasterSignal (MASTERSIGNAL)	587
11.2.2.5.12	Exec_Master (MASTER)	587
11.2.2.5.13	Get_Master (MASTER)	588
11.2.2.5.14	Set_Computation (COMP)	589
11.2.2.5.15	Get_Computation (COMP)	592
11.2.2.5.16	Set_SystemSignalRange (SYSSIGNALRANGE)	595
11.2.2.5.17	Get_SystemSignalRange (SYSSIGNALRANGE)	596
11.2.3	Data output	596
11.2.3.1	General	596
11.2.3.1.1	Set_DataOutInterface (OUTPUT)	596
11.2.3.1.2	Get_DataOutInterface (OUTPUT)	597
11.2.3.1.3	Set_ResamplingDevice (OUTREDUCEDEVICE)	598

11.2.3.1.4	Get_ResamplingDevice (OUTREDUCEDEVICE)	598
11.2.3.1.5	Set_ResamplingCount (OUTREDUCECOUNT)	599
11.2.3.1.6	Get_ResamplingCount (OUTREDUCECOUNT)	599
11.2.3.1.7	Set_HoldLastValid (OUTHOLD)	599
11.2.3.1.8	Get_HoldLastValid (OUTHOLD)	599
11.2.3.1.9	Set_FramesPerPacket_ETH (MEASCNT_ETH)	600
11.2.3.1.10	Get_FramesPerPacket_ETH (MEASCNT_ETH)	600
11.2.3.2	Selected measurement values	600
11.2.3.2.1	Set_Output_RS422 (OUT_RS422)	600
11.2.3.2.2	Get_Output_RS422 (OUT_RS422)	604
11.2.3.2.3	Set_Output_ETH (OUT_ETH)	608
11.2.3.2.4	Get_Output_ETH (OUT_ETH)	611
11.2.3.3	Switching outputs	615
11.2.3.3.1	Set_ErrorOutput<n> (ERRORROUT<n>)	615
11.2.3.3.2	Get_ErrorOutput<n> (ERRORROUT<n>)	616
11.2.3.3.3	Set_ErrorLimitSignal<n> (ERRORLIMITSIGNAL<n>)	616
11.2.3.3.4	Get_ErrorLimitSignal<n> (ERRORLIMITSIGNAL<n>)	617
11.2.3.3.5	Set_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)	617
11.2.3.3.6	Get_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)	617
11.2.3.3.7	Set_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)	618
11.2.3.3.8	Get_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)	618
11.2.3.3.9	Set_ErrorLevelOut<n> (ERRORLEVELOUT<n>)	618
11.2.3.3.10	Get_ErrorLevelOut<n> (ERRORLEVELOUT<n>)	619
11.2.3.4	Analog output	619
11.2.3.4.1	Set_AnalogOutput (ANALOGOUT)	619
11.2.3.4.2	Get_AnalogOutput (ANALOGOUT)	619
11.2.3.4.3	Set_AnalogRange (ANALOGRANGE)	620
11.2.3.4.4	Get_AnalogRange (ANALOGRANGE)	620
11.2.3.4.5	Set_AnalogScaleMode (ANALOGSCALEMODE)	620
11.2.3.4.6	Get_AnalogScaleMode (ANALOGSCALEMODE)	621
11.2.3.4.7	Set_AnalogScaleRange (ANALOGSCALERANGE)	621
11.2.3.4.8	Get_AnalogScaleRange (ANALOGSCALERANGE)	621
11.2.4	Internal commands	622
11.2.4.1	Upload_CalibTable	622
11.3	Commands for IFD2445, IFD2451, IFD2461 and IFD2471	622
11.3.1	General commands	623
11.3.1.1	General	623
11.3.1.1.1	Get_Help (HELP)	623
11.3.1.1.2	Get_Info (GETINFO)	623
11.3.1.1.3	Get_Temperature (GETTEMP)	624
11.3.1.1.4	Set_Echo (ECHO)	624
11.3.1.1.5	Get_Echo (ECHO)	624
11.3.1.1.6	Get_AllParameters (PRINT)	625
11.3.1.1.7	Set_SyncMode (SYNC)	653
11.3.1.1.8	Get_SyncMode (SYNC)	654
11.3.1.1.9	Reset_Boot (RESET)	654
11.3.1.2	User level	654
11.3.1.2.1	Logout (LOGOUT)	654
11.3.1.2.2	Login (LOGIN)	654
11.3.1.2.3	Get_UserLevel (GETUSERLEVEL)	654
11.3.1.2.4	Set_DefaultUser (STDUSER)	655
11.3.1.2.5	Get_DefaultUser (STDUSER)	655
11.3.1.2.6	Set_Password (PASSWD)	655

11.3.1.3	Sensor	655
11.3.1.3.1	Get_SensorTable (SENSORTABLE)	655
11.3.1.3.2	Set_ActiveSensor (SENSORHEAD)	656
11.3.1.3.3	Get_ActiveSensor (SENSORHEAD)	656
11.3.1.3.4	Get_SensorInfo (SENSORINFO)	657
11.3.1.3.5	Set_LEDControl (LED)	657
11.3.1.3.6	Get_LEDControl (LED)	657
11.3.1.3.7	DarkCorr (DARKCORR)	658
11.3.1.3.8	Set_DarkCorrThreshold (DARKCORRTHRES)	658
11.3.1.3.9	Get_DarkCorrThreshold (DARKCORRTHRES)	658
11.3.1.3.10	LightCorr (LIGHTCORR)	658
11.3.1.4	Triggering	658
11.3.1.4.1	Set_TriggerMode (TRIGGER)	658
11.3.1.4.2	Get_TriggerMode (TRIGGER)	659
11.3.1.4.3	Set_TriggerMoment (TRIGGERAT)	659
11.3.1.4.4	Get_TriggerMoment (TRIGGERAT)	659
11.3.1.4.5	Set_TriggerLevel (TRIGGERLEVEL)	659
11.3.1.4.6	Get_TriggerLevel (TRIGGERLEVEL)	660
11.3.1.4.7	Set_TriggerCount (TRIGGERCOUNT)	660
11.3.1.4.8	Get_TriggerCount (TRIGGERCOUNT)	660
11.3.1.4.9	Software_Trigger (TRIGGERSW)	660
11.3.1.4.10	Set_TriggerOnEncoder (TRIGGERENC)	660
11.3.1.4.11	Get_TriggerOnEncoder (TRIGGERENC)	661
11.3.1.5	Encoder	662
11.3.1.5.1	Set_EncoderInterpolation<n> (ENCINTERPOL<n>)	662
11.3.1.5.2	Get_EncoderInterpolation<n> (ENCINTERPOL<n>)	662
11.3.1.5.3	Set_EncoderMode<n> (ENCREF<n>)	662
11.3.1.5.4	Get_EncoderMode<n> (ENCREF<n>)	663
11.3.1.5.5	Set_EncoderPreload<n> (ENCVALUE<n>)	663
11.3.1.5.6	Get_EncoderPreload<n> (ENCVALUE<n>)	663
11.3.1.5.7	Load_Encoder<n> (ENCSET)	663
11.3.1.5.8	EnableRef_Encoder<n> (ENCRESET)	663
11.3.1.5.9	Set_Encoder.MaxValue<n> (ENCMAX<n>)	664
11.3.1.5.10	Get_Encoder.MaxValue<n> (ENCMAX<n>)	664
11.3.1.6	Interfaces	664
11.3.1.6.1	Set_IPConfiguration (IPCONFIG)	664
11.3.1.6.2	Get_IPConfiguration (IPCONFIG)	665
11.3.1.6.3	Set_IPDataTransferMode (MEATRANSFER)	665
11.3.1.6.4	Get_IPDataTransferMode (MEATRANSFER)	666
11.3.1.6.5	Set_EthernetMode (ETHERMODE)	666
11.3.1.6.6	Get_EthernetMode (ETHERMODE)	667
11.3.1.6.7	Set_Baudrate (BAUDRATE)	667
11.3.1.6.8	Get_Baudrate (BAUDRATE)	668
11.3.1.7	Parameter management	668
11.3.1.7.1	Save_Parameters (STORE)	668
11.3.1.7.2	Load_Parameters (READ)	668
11.3.1.7.3	Set_Default (SETDEFAULT)	669
11.3.2	Measurement	669
11.3.2.1	General	669
11.3.2.1.1	Set_MeasureMode (MEASMODE)	669
11.3.2.1.2	Get_MeasureMode (MEASMODE)	670
11.3.2.1.3	Set_MeasurePeak (MEASPEAK)	670
11.3.2.1.4	Get_MeasurePeak (MEASPEAK)	671

11.3.2.1.5	Set_ShutterMode (SHUTTERMODE)	671
11.3.2.1.6	Get_ShutterMode (SHUTTERMODE)	671
11.3.2.1.7	Set_Samplerate (MEASRATE)	672
11.3.2.1.8	Get_Samplerate (MEASRATE)	672
11.3.2.1.9	Set_ShutterTime (SHUTTER)	672
11.3.2.1.10	Get_ShutterTime (SHUTTER)	673
11.3.2.1.11	Get_Video (GETVIDEO)	673
11.3.2.1.12	Get_VideoStreamSignal	674
11.3.2.2	Video signal	676
11.3.2.2.1	Set_ROI (ROI)	676
11.3.2.2.2	Get_ROI (ROI)	676
11.3.2.2.3	Set_VideoAverage (VSAVERAGE)	677
11.3.2.2.4	Get_VideoAverage (VSAVERAGE)	677
11.3.2.2.5	Set_Threshold (THRESHOLD)	677
11.3.2.2.6	Get_Threshold (THRESHOLD)	678
11.3.2.3	Material database	678
11.3.2.3.1	Get_MaterialTable (MATERIALTABLE)	678
11.3.2.3.2	Set_ActiveMaterial (MATERIAL)	679
11.3.2.3.3	Get_ActiveMaterial (MATERIAL)	679
11.3.2.3.4	Get_MaterialInfo (MATERIALINFO)	679
11.3.2.3.5	Edit_Material_Abbe (MATERIALEDIT)	680
11.3.2.3.6	Edit_Material_Nx (MATERIALEDIT)	681
11.3.2.3.7	Delete_Material (MATERIALDELETE)	681
11.3.2.3.8	Clear_MaterialTable	682
11.3.2.4	Peak processing	682
11.3.2.4.1	Set_RefractiveCorrection (REFRACCORR)	682
11.3.2.4.2	Get_RefractiveCorrection (REFRACCORR)	682
11.3.2.4.3	Set_MaterialMultiPeak (MATERIALMP)	683
11.3.2.4.4	Get_MaterialMultiPeak (MATERIALMP)	683
11.3.2.5	Measurement value processing	684
11.3.2.5.1	Set_Averaging (AVERAGE)	684
11.3.2.5.2	Get_Averaging (AVERAGE)	685
11.3.2.5.3	Set_SpikeCorrection (SPIKECORR)	686
11.3.2.5.4	Get_SpikeCorrection (SPIKECORR)	686
11.3.2.5.5	Set_StatisticSignal (STATISTICSIGNAL)	687
11.3.2.5.6	Get_StatisticSignal (STATISTICSIGNAL)	688
11.3.2.5.7	Set_StatisticDepth (STATICDEPTH)	688
11.3.2.5.8	Get_StatisticDepth (STATICDEPTH)	689
11.3.2.5.9	Reset_Statistic (RESETSTATIC)	689
11.3.2.5.10	Set_MasterSignal (MASTERSIGNAL)	689
11.3.2.5.11	Get_MasterSignal (MASTERSIGNAL)	690
11.3.2.5.12	Set_MasterValue (MASTERMV)	690
11.3.2.5.13	Get_MasterValue (MASTERMV)	691
11.3.3	Data output	691
11.3.3.1	General	691
11.3.3.1.1	Set_DataOutInterface (OUTPUT)	691
11.3.3.1.2	Get_DataOutInterface (OUTPUT)	691
11.3.3.1.3	Set_Resampling (OUTREDUCE)	692
11.3.3.1.4	Get_Resampling (OUTREDUCE)	692
11.3.3.1.5	Set_HoldLastValid (OUTHOLD)	693
11.3.3.1.6	Get_HoldLastValid (OUTHOLD)	693
11.3.3.1.7	Set_FramesPerPacket_ETH (MEASCNT_ETH)	693
11.3.3.1.8	Get_FramesPerPacket_ETH (MEASCNT_ETH)	694

11.3.3.2 Selected measurement values	694
11.3.3.2.1 Set_OutputDistance_RS422 (OUTDIST_RS422)	694
11.3.3.2.2 Get_OutputDistance_RS422 (OUTDIST_RS422)	695
11.3.3.2.3 Set_OutputDistance_ETH (OUTDIST_ETH)	696
11.3.3.2.4 Get_OutputDistance_ETH (OUTDIST_ETH)	698
11.3.3.2.5 Set_OutputThickness_RS422 (OUTTHICK_RS422)	699
11.3.3.2.6 Get_OutputThickness_RS422 (OUTTHICK_RS422)	702
11.3.3.2.7 Set_OutputThickness_ETH (OUTTHICK_ETH)	706
11.3.3.2.8 Get_OutputThickness_ETH (OUTTHICK_ETH)	709
11.3.3.2.9 Set_OutputStatistic_RS422 (OUTSTATISTIC_RS422)	712
11.3.3.2.10 Get_OutputStatistic_RS422 (OUTSTATISTIC_RS422)	712
11.3.3.2.11 Set_OutputStatistic_ETH (OUTSTATISTIC_ETH)	713
11.3.3.2.12 Get_OutputStatistic_ETH (OUTSTATISTIC_ETH)	713
11.3.3.2.13 Set_OutputAdditional_RS422 (OUTADD_RS422)	714
11.3.3.2.14 Get_OutputAdditional_RS422 (OUTADD_RS422)	715
11.3.3.2.15 Set_OutputAdditional_ETH (OUTADD_ETH)	717
11.3.3.2.16 Get_OutputAdditional_ETH (OUTADD_ETH)	718
11.3.3.2.17 Set_UnlinearizedMode (SWITCHMD2)	719
11.3.3.2.18 Get_UnlinearizedMode (SWITCHMD2)	720
11.3.3.2.19 Set_OutputVideo_ETH (OUTVIDEO)	720
11.3.3.2.20 Get_OutputVideo_ETH (OUTVIDEO)	721
11.3.3.3 Switching outputs	722
11.3.3.3.1 Set_ErrorOutput<n> (ERROROUT<n>)	722
11.3.3.3.2 Get_ErrorOutput<n> (ERROROUT<n>)	722
11.3.3.3.3 Set_ErrorLimit (ERRORLIMIT)	722
11.3.3.3.4 Get_ErrorLimit (ERRORLIMIT)	723
11.3.3.3.5 Set_ErrorLevel (ERRORLEVEL)	724
11.3.3.3.6 Get_ErrorLevel (ERRORLEVEL)	724
11.3.3.4 Analog output	725
11.3.3.4.1 Set_AnalogOutput (ANALOGOUT)	725
11.3.3.4.2 Get_AnalogOutput (ANALOGOUT)	725
11.3.3.4.3 Set_AnalogRange (ANALOGRANGE)	726
11.3.3.4.4 Get_AnalogRange (ANALOGRANGE)	726
11.3.3.4.5 Set_AnalogScale (ANALOGSCALE)	726
11.3.3.4.6 Get_AnalogScale (ANALOGSCALE)	727
11.3.4 Internal commands	728
11.3.4.1 Set_PeakParameter (PEAKPARAM)	728
11.3.4.2 Get_PeakParameter (PEAKPARAM)	728
11.3.4.3 Set_BoreControlParam (BORECTRL)	728
11.3.4.4 Get_BoreControlParam (BORECTRL)	729
11.3.4.5 Get_SpecCal (SPECCAL)	729
11.3.4.6 Del_LinearisationTable (DELLINTAB)	730
11.3.4.7 Save_ParametersTemp (SAVE2TMP)	730
11.3.4.8 Load_ParametersTemp (RESTORETMP)	730
11.3.4.9 Set_CalibrationDefaults (SETCALDEFAULT)	730
11.3.4.10 Get_RegHyst (REGHYST)	730
11.3.4.11 Upload_CalibTable	731
11.3.4.12 Save_DarkCorrReference (SAVEDARKCORRREF)	731

12 Commands for interferoMETER (IMC) sensors	732
12.1 Commands for IMC5400/10 and IMC5600	732
12.1.1 General commands	732
12.1.1.1 General	732
12.1.1.1.1 Get_Help (HELP)	732
12.1.1.1.2 Get_Info (GETINFO)	733
12.1.1.1.3 Get_ChannelCount (GETCHANNELCNT)	734
12.1.1.1.4 Get_OutputInfo_RS422 (GETOUTINFO_RS422)	734
12.1.1.1.5 Get_OutputInfo_ETH (GETOUTINFO_ETH)	735
12.1.1.1.6 Set_Unit (UNIT)	737
12.1.1.1.7 Get_Unit (UNIT)	737
12.1.1.1.8 Set_Echo (ECHO)	737
12.1.1.1.9 Get_Echo (ECHO)	737
12.1.1.1.10 Get_AllParameters (PRINT)	738
12.1.1.1.11 Set_SyncMode (SYNC)	760
12.1.1.1.12 Get_SyncMode (SYNC)	760
12.1.1.1.13 Set_Termination (TERMINATION)	761
12.1.1.1.14 Get_Termination (TERMINATION)	761
12.1.1.1.15 Reset_Boot (RESET)	761
12.1.1.1.16 Reset_Counter (RESETCNT)	761
12.1.1.1.17 Set_Keylock (KEYLOCK)	762
12.1.1.1.18 Get_Keylock (KEYLOCK)	762
12.1.1.1.19 Set_KeyFunction1 (KEYFUNC1)	763
12.1.1.1.20 Get_KeyFunction1 (KEYFUNC1)	763
12.1.1.1.21 Set_KeyFunction2 (KEYFUNC2)	763
12.1.1.1.22 Get_KeyFunction2 (KEYFUNC2)	764
12.1.1.1.23 Set_KeyMasterSignalSelect (KEYMASTERSIGNALSELECT)	764
12.1.1.1.24 Get_KeyMasterSignalSelect (KEYMASTERSIGNALSELECT)	764
12.1.1.2 User level	765
12.1.1.2.1 Logout (LOGOUT)	765
12.1.1.2.2 Login (LOGIN)	765
12.1.1.2.3 Get_UserLevel (GETUSERLEVEL)	765
12.1.1.2.4 Set_DefaultUser (STDUSER)	765
12.1.1.2.5 Get_DefaultUser (STDUSER)	766
12.1.1.2.6 Set_Password (PASSWD)	766
12.1.1.3 Sensor	766
12.1.1.3.1 Get_SensorTable (SENSORTABLE)	766
12.1.1.3.2 Set_ActiveSensor (SENSORHEAD)	767
12.1.1.3.3 Get_ActiveSensor (SENSORHEAD)	767
12.1.1.3.4 Get_SensorInfo (SENSORINFO)	768
12.1.1.3.5 Set_SuperluminescentLED (SLED)	768
12.1.1.3.6 Get_SuperluminescentLED (SLED)	768
12.1.1.3.7 Set_PilotLaser (PILOTLASER)	769
12.1.1.3.8 Get_PilotLaser (PILOTLASER)	769
12.1.1.4 Triggering	769
12.1.1.4.1 Set_TriggerSource (TRIGGERSOURCE)	769
12.1.1.4.2 Get_TriggerSource (TRIGGERSOURCE)	769
12.1.1.4.3 Set_TriggerMode (TRIGGERMODE)	770
12.1.1.4.4 Get_TriggerMode (TRIGGERMODE)	770
12.1.1.4.5 Set_TriggerMoment (TRIGGERAT)	770
12.1.1.4.6 Get_TriggerMoment (TRIGGERAT)	770
12.1.1.4.7 Set_TriggerInLogic (TRIGINLEVEL)	771
12.1.1.4.8 Get_TriggerInLogic (TRIGINLEVEL)	771

12.1.1.4.9	Set_TriggerLevel (TRIGGERLEVEL)	771
12.1.1.4.10	Get_TriggerLevel (TRIGGERLEVEL)	771
12.1.1.4.11	Set_TriggerCount (TRIGGERCOUNT)	772
12.1.1.4.12	Get_TriggerCount (TRIGGERCOUNT)	772
12.1.1.4.13	Software_Trigger (TRIGGERSW)	772
12.1.1.4.14	Set_TriggerEncoderMin (TRIGGERENCMIN)	772
12.1.1.4.15	Get_TriggerEncoderMin (TRIGGERENCMIN)	772
12.1.1.4.16	Set_TriggerEncoderMax (TRIGGERENCMAX)	773
12.1.1.4.17	Get_TriggerEncoderMax (TRIGGERENCMAX)	773
12.1.1.4.18	Set_TriggerEncoderStepsize (TRIGGERENCSTEP SIZE)	773
12.1.1.4.19	Get_TriggerEncoderStepsize (TRIGGERENCSTEP SIZE)	773
12.1.1.5	Encoder	774
12.1.1.5.1	Set_EncoderInterpolation<n> (ENCINTERPOL<n>)	774
12.1.1.5.2	Get_EncoderInterpolation<n> (ENCINTERPOL<n>)	774
12.1.1.5.3	Set_Encoder3Enable (ENCODER3)	774
12.1.1.5.4	Get_Encoder3Enable (ENCODER3)	774
12.1.1.5.5	Set_EncoderMode<n> (ENCREF<n>)	775
12.1.1.5.6	Get_EncoderMode<n> (ENCREF<n>)	775
12.1.1.5.7	Set_EncoderPreload<n> (ENCVALUE<n>)	775
12.1.1.5.8	Get_EncoderPreload<n> (ENCVALUE<n>)	775
12.1.1.5.9	Load_Encoder<n> (ENCSET)	776
12.1.1.5.10	EnableRef_Encoder<n> (ENCRESET)	776
12.1.1.5.11	Set_Encoder.MaxValue<n> (ENCMAX<n>)	776
12.1.1.5.12	Get_Encoder.MaxValue<n> (ENCMAX<n>)	776
12.1.1.6	Interfaces	776
12.1.1.6.1	Set_IPConfiguration (IPCONFIG)	776
12.1.1.6.2	Get_IPConfiguration (IPCONFIG)	777
12.1.1.6.3	Set_IPDataTransferMode (MEATRANSFER)	778
12.1.1.6.4	Get_IPDataTransferMode (MEATRANSFER)	778
12.1.1.6.5	Set_TCPKeepAlive (TCPKEEPALIVE)	779
12.1.1.6.6	Get_TCPKeepAlive (TCPKEEPALIVE)	779
12.1.1.6.7	Set_EthernetMode (ETHERMODE)	779
12.1.1.6.8	Get_EthernetMode (ETHERMODE)	779
12.1.1.6.9	Set_Baudrate (BAUDRATE)	780
12.1.1.6.10	Get_Baudrate (BAUDRATE)	780
12.1.1.6.11	Set_AppLanguage (LANGUAGE)	781
12.1.1.6.12	Get_AppLanguage (LANGUAGE)	781
12.1.1.7	Parameter management	781
12.1.1.7.1	Save_InterfaceParameters (BASICSETTINGS STORE)	781
12.1.1.7.2	Load_InterfaceParameters (BASICSETTINGS READ)	781
12.1.1.7.3	Save_MeasureParameters (MEASSETTINGS STORE)	781
12.1.1.7.4	Load_MeasureParameters (MEASSETTINGS READ)	781
12.1.1.7.5	Rename_MeasureParameters (MEASSETTINGS RENAME)	782
12.1.1.7.6	Get_CurrentMeasureSetting (MEASSETTINGS CURRENT)	782
12.1.1.7.7	Get_MeasureSettingsList (MEASSETTINGS LIST)	782
12.1.1.7.8	Delete_MeasureParameters (MEASSETTINGS DELETE)	782
12.1.1.7.9	Get_MeasurePresetList (MEASSETTINGS PRESETLIST)	782
12.1.1.7.10	Set_InitialMeasureSetting (MEASSETTINGS INITIAL)	783
12.1.1.7.11	Get_InitialMeasureSetting (MEASSETTINGS INITIAL)	783
12.1.1.7.12	Set_MeasurePresetMode (MEASSETTINGS PRESETMODE)	783
12.1.1.7.13	Get_MeasurePresetMode (MEASSETTINGS PRESETMODE)	783
12.1.1.7.14	Set_Default (SETDEFAULT)	784
12.1.1.7.15	Export_Parameters (EXPORT)	784

12.1.1.7.16 Import_Parameters (IMPORT)	784
12.1.1.8 Internal controller commands	785
12.1.1.8.1 Set_ChartType (CHARTTYPE)	785
12.1.1.8.2 Get_ChartType (CHARTTYPE)	785
12.1.2 Measurement	785
12.1.2.1 General	785
12.1.2.1.1 Set_MeasurePeakSort (MEASPEAK_SORT)	785
12.1.2.1.2 Get_MeasurePeakSort (MEASPEAK_SORT)	786
12.1.2.1.3 Set_Samplerate (MEASRATE)	786
12.1.2.1.4 Get_Samplerate (MEASRATE)	786
12.1.2.1.5 Get_VideoStreamSignal	786
12.1.2.2 Video signal	787
12.1.2.2.1 Set_ROI (ROI)	787
12.1.2.2.2 Get_ROI (ROI)	788
12.1.2.3 Material database	788
12.1.2.3.1 Get_MaterialTable (MATERIALTABLE)	788
12.1.2.3.2 Set_ActiveMaterial (MATERIAL)	789
12.1.2.3.3 Get_ActiveMaterial (MATERIAL)	789
12.1.2.3.4 Get_MaterialInfo (MATERIALINFO)	789
12.1.2.3.5 Edit_Material (MATERIALEDIT)	790
12.1.2.3.6 Delete_Material (MATERIALDELETE)	790
12.1.2.3.7 Clear_MaterialTable	790
12.1.2.4 Peak processing	790
12.1.2.4.1 Set_PeakCount (PEAKCOUNT)	790
12.1.2.4.2 Get_PeakCount (PEAKCOUNT)	791
12.1.2.4.3 Set_MaterialMultiPeak (MATERIALMP)	791
12.1.2.4.4 Get_MaterialMultiPeak (MATERIALMP)	791
12.1.2.4.5 Set_MinimumThreshold (MIN_THRESHOLD)	792
12.1.2.4.6 Get_MinimumThreshold (MIN_THRESHOLD)	792
12.1.2.5 Measurement value processing	793
12.1.2.5.1 Set_StatisticSignal (STATISTICSIGNAL)	793
12.1.2.5.2 Get_StatisticSignal (STATISTICSIGNAL)	793
12.1.2.5.3 Reset_StatisticSignal (STATISTIC RESET)	795
12.1.2.5.4 Set_MasterSignal (MASTERSIGNAL)	795
12.1.2.5.5 Get_MasterSignal (MASTERSIGNAL)	796
12.1.2.5.6 Reset_MasterSignal (MASTERSIGNAL)	798
12.1.2.5.7 Exec_Master (MASTER)	798
12.1.2.5.8 Get_Master (MASTER)	799
12.1.2.5.9 Set_Computation (COMP)	800
12.1.2.5.10 Get_Computation (COMP)	803
12.1.3 Data output	806
12.1.3.1 General	806
12.1.3.1.1 Set_DataOutInterface (OUTPUT)	806
12.1.3.1.2 Get_DataOutInterface (OUTPUT)	806
12.1.3.1.3 Set_ResamplingDevice (OUTREDUCEDEVICE)	807
12.1.3.1.4 Get_ResamplingDevice (OUTREDUCEDEVICE)	807
12.1.3.1.5 Set_ResamplingCount (OUTREDUCECOUNT)	808
12.1.3.1.6 Get_ResamplingCount (OUTREDUCECOUNT)	808
12.1.3.1.7 Set_HoldLastValid (OUTHOLD)	808
12.1.3.1.8 Get_HoldLastValid (OUTHOLD)	809
12.1.3.1.9 Set_FramesPerPacket_ETH (MEASCNT_ETH)	809
12.1.3.1.10 Get_FramesPerPacket_ETH (MEASCNT_ETH)	809
12.1.3.2 Selected measurement values	809

12.1.3.2.1	Set_Output_RS422 (OUT_RS422)	809
12.1.3.2.2	Get_Output_RS422 (OUT_RS422)	811
12.1.3.2.3	Set_Output_ETH (OUT_ETH)	812
12.1.3.2.4	Get_Output_ETH (OUT_ETH)	814
12.1.3.3	Switching outputs	815
12.1.3.3.1	Set_ErrorLimitSignal<n> (ERRORLIMITSIGNAL<n>)	815
12.1.3.3.2	Get_ErrorLimitSignal<n> (ERRORLIMITSIGNAL<n>)	816
12.1.3.3.3	Set_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)	816
12.1.3.3.4	Get_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)	817
12.1.3.3.5	Set_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)	817
12.1.3.3.6	Get_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)	817
12.1.3.3.7	Set_ErrorLevelOut<n> (ERRORLEVELOUT<n>)	818
12.1.3.3.8	Get_ErrorLevelOut<n> (ERRORLEVELOUT<n>)	818
12.1.3.4	Analog output	818
12.1.3.4.1	Set_AnalogOutput (ANALOGOUT)	818
12.1.3.4.2	Get_AnalogOutput (ANALOGOUT)	819
12.1.3.4.3	Set_AnalogRange (ANALOGRANGE)	819
12.1.3.4.4	Get_AnalogRange (ANALOGRANGE)	820
12.1.3.4.5	Set_AnalogScaleMode (ANALOGSCALEMODE)	820
12.1.3.4.6	Get_AnalogScaleMode (ANALOGSCALEMODE)	820
12.1.3.4.7	Set_AnalogScaleRange (ANALOGSCALERANGE)	820
12.1.3.4.8	Get_AnalogScaleRange (ANALOGSCALERANGE)	821
13 Commands for optoCONTROL (ODC) sensors		822
13.1 Commands for ODC1202/20	822
13.1.1 General commands	822
13.1.1.1 General	822
13.1.1.1.1 Get_Version	822
13.1.1.2 User level	823
13.1.1.2.1 Get_Echo	823
13.1.1.3 Interfaces	823
13.1.1.3.1 Change_Baudrate	823
13.1.1.4 Parameter management	823
13.1.1.4.1 RefreshVideoThr	823
13.1.2 Measurement	824
13.1.2.1 Set_ParamSet1	824
13.1.2.2 Get_ParamSet1	826
13.1.2.3 Set_ParamSet2	829
13.1.2.4 Get_ParamSet2	831
13.1.2.5 Get_Video	834
13.1.2.6 Get_MeasValues	834
13.1.2.7 Get_Value_DataRecord	836
13.1.3 Data output	837
13.1.3.1 General	837
13.1.3.1.1 AutoDataTransfer	837
13.1.3.2 Switching outputs	838
13.1.3.2.1 Activate_Teach	838
13.1.3.3 Analog output	840
13.1.3.3.1 Reset_MinMax_Analog	840
13.2 Commands for ODC2500	840
13.2.1 General commands	841
13.2.1.1 General	841
13.2.1.1.1 Get_Info (INFO)	841

13.2.1.1.2 Reset_Boot (RESET)	842
13.2.1.2 Parameter management	842
13.2.1.2.1 Save_OptionData (SAVE_OPT_RAM_TO_FLASH)	842
13.2.1.2.2 Save_MeasProgData (SAVE_MPR_RAM_TO_FLASH)	842
13.2.2 Measurement	843
13.2.2.1 Choose_MeasProg (CHOOSE_MP)	843
13.2.2.2 Switch_Edge (SWITCH_EDGE)	843
13.2.2.3 Write_OptionData (WR_OPT_TO_RAM)	844
13.2.2.4 Read_OptionData (RD_OPT_RAM)	847
13.2.2.5 Update_OptionData	850
13.2.2.6 Write_MeasProgData (WR_MPR_TO_RAM)	853
13.2.2.7 Read_MeasProgData (RD_MPR_RAM)	856
13.2.2.8 Read_MinMax (RD_MINMAX)	858
13.2.2.9 Read_MinMaxReset (RD_MINMAX_RESET)	859
13.2.3 Data output	860
13.2.3.1 General	860
13.2.3.1.1 Dat_Out_Off (STOP)	860
13.2.3.1.2 Dat_Out_On (START)	860
13.3 Commands for ODC2520	860
13.3.1 General commands	861
13.3.1.1 General	861
13.3.1.1.1 Get_Help (HELP)	861
13.3.1.1.2 Get_Info (GETINFO)	861
13.3.1.1.3 Set_Echo (ECHO)	862
13.3.1.1.4 Get_Echo (ECHO)	862
13.3.1.1.5 Get_AllParameters (PRINT)	863
13.3.1.1.6 Set_SyncMode (SYNC)	891
13.3.1.1.7 Get_SyncMode (SYNC)	892
13.3.1.1.8 Set_Unit (UNIT)	892
13.3.1.1.9 Get_Unit (UNIT)	892
13.3.1.1.10 Reset_Boot (RESET)	893
13.3.1.2 User level	893
13.3.1.2.1 Logout (LOGOUT)	893
13.3.1.2.2 Login (LOGIN)	893
13.3.1.2.3 Get_UserLevel (GETUSERLEVEL)	893
13.3.1.2.4 Set_DefaultUser (STDUSER)	893
13.3.1.2.5 Get_DefaultUser (STDUSER)	894
13.3.1.2.6 Set_Password (PASSWD)	894
13.3.1.3 Measure distance	894
13.3.1.3.1 Get_MeasDistTable (MEASDISTTABLE)	894
13.3.1.3.2 Set_ActiveMeasDist (MEASDIST)	895
13.3.1.3.3 Get_ActiveMeasDist (MEASDIST)	895
13.3.1.3.4 Get_MeasDistInfo (MEASDISTINFO)	895
13.3.1.3.5 LightCorr (LIGHTCORR)	895
13.3.1.4 Triggering	896
13.3.1.4.1 Set_TriggerMode (TRIGGER)	896
13.3.1.4.2 Get_TriggerMode (TRIGGER)	896
13.3.1.4.3 Set_TriggerMoment (TRIGGERAT)	896
13.3.1.4.4 Get_TriggerMoment (TRIGGERAT)	897
13.3.1.4.5 Set_TriggerLevel (TRIGGERLEVEL)	897
13.3.1.4.6 Get_TriggerLevel (TRIGGERLEVEL)	897
13.3.1.4.7 Set_TriggerCount (TRIGGERCOUNT)	897
13.3.1.4.8 Get_TriggerCount (TRIGGERCOUNT)	898

13.3.1.4.9	Software_Trigger (TRIGGERSW)	898
13.3.1.5	Interfaces	898
13.3.1.5.1	Set_IPConfiguration (IPCONFIG)	898
13.3.1.5.2	Get_IPConfiguration (IPCONFIG)	899
13.3.1.5.3	Set_IPDataTransferMode (MEASTRANSFER)	899
13.3.1.5.4	Get_IPDataTransferMode (MEASTRANSFER)	900
13.3.1.5.5	Set_EthernetMode (ETHERMODE)	900
13.3.1.5.6	Get_EthernetMode (ETHERMODE)	900
13.3.1.5.7	Set_Baudrate (BAUDRATE)	901
13.3.1.5.8	Get_Baudrate (BAUDRATE)	901
13.3.1.6	Parameter management	902
13.3.1.6.1	Save_Parameters (STORE)	902
13.3.1.6.2	Load_Parameters (READ)	902
13.3.1.6.3	Set_Default (SETDEFAULT)	902
13.3.2	Measurement	903
13.3.2.1	General	903
13.3.2.1.1	Set_MeasureMode (MEASMODE)	903
13.3.2.1.2	Get_MeasureMode (MEASMODE)	903
13.3.2.1.3	Set_SearchDirection (SEARCHDIR)	903
13.3.2.1.4	Get_SearchDirection (SEARCHDIR)	904
13.3.2.1.5	Set_MeasureDirection (MEASDIR)	904
13.3.2.1.6	Get_MeasureDirection (MEASDIR)	904
13.3.2.1.7	Set_ExpectedEdges (EXPEDGES)	904
13.3.2.1.8	Get_ExpectedEdges (EXPEDGES)	905
13.3.2.1.9	Set_SegmentDefinition<n> (DEFSEG<n>)	905
13.3.2.1.10	Get_SegmentDefinition<n> (DEFSEG<n>)	905
13.3.2.1.11	Set_LaserPower (LASER)	906
13.3.2.1.12	Get_LaserPower (LASER)	906
13.3.2.1.13	Get_VideoStreamSignal	906
13.3.2.2	Video signal	907
13.3.2.2.1	Set_ROI (ROI)	907
13.3.2.2.2	Get_ROI (ROI)	908
13.3.2.2.3	Set_Threshold (THRESHOLD)	908
13.3.2.2.4	Get_Threshold (THRESHOLD)	908
13.3.2.3	Measurement value processing	909
13.3.2.3.1	Set_Averaging (AVERAGE)	909
13.3.2.3.2	Get_Averaging (AVERAGE)	910
13.3.2.3.3	Set_SpikeCorrection (SPIKECORR)	910
13.3.2.3.4	Get_SpikeCorrection (SPIKECORR)	911
13.3.2.3.5	Set_StatisticSignal (STATISTICSIGNAL)	912
13.3.2.3.6	Get_StatisticSignal (STATISTICSIGNAL)	912
13.3.2.3.7	Set_Statistic2Signal (STATISTIC2SIGNAL)	913
13.3.2.3.8	Get_Statistic2Signal (STATISTIC2SIGNAL)	914
13.3.2.3.9	Set_StatisticDepth (STATISTICDEPTH)	915
13.3.2.3.10	Get_StatisticDepth (STATISTICDEPTH)	916
13.3.2.3.11	Reset_ Statistic (RESETSTATISTIC)	916
13.3.2.3.12	Set_EdgeFilter<n> (EDGEFILTER<n>)	916
13.3.2.3.13	Get_EdgeFilter<n> (EDGEFILTER<n>)	917
13.3.2.3.14	Set_EdgeFilter<n>Signal (EDGEFILTER<n>SIGNAL)	917
13.3.2.3.15	Get_EdgeFilter<n>Signal (EDGEFILTER<n>SIGNAL)	918
13.3.2.3.16	Set_MasterSignal (MASTERSIGNAL)	919
13.3.2.3.17	Get_MasterSignal (MASTERSIGNAL)	920
13.3.2.3.18	Set_MasterValue (MASTERMV)	921

13.3.2.3.19	Get_MasterValue (MASTERMV)	922
13.3.3	Data output	922
13.3.3.1	General	922
13.3.3.1.1	Set_DataOutInterface (OUTPUT)	922
13.3.3.1.2	Get_DataOutInterface (OUTPUT)	923
13.3.3.1.3	Set_Resampling (OUTREDUCE)	923
13.3.3.1.4	Get_Resampling (OUTREDUCE)	923
13.3.3.1.5	Set_HoldLastValid (OUTHOLD)	924
13.3.3.1.6	Get_HoldLastValid (OUTHOLD)	924
13.3.3.2	Selected measurement values	925
13.3.3.2.1	Set_OutputEdgeLightDark_RS422 (OUTEDGEHL_RS422)	925
13.3.3.2.2	Get_OutputEdgeLightDark_RS422 (OUTEDGEHL_RS422)	925
13.3.3.2.3	Set_OutputEdgeLightDark_ETH (OUTEDGEHL_ETH)	925
13.3.3.2.4	Get_OutputEdgeLightDark_ETH (OUTEDGEHL_ETH)	925
13.3.3.2.5	Set_OutputEdgeDarkLight_RS422 (OUTEDGELH_RS422)	926
13.3.3.2.6	Get_OutputEdgeDarkLight_RS422 (OUTEDGELH_RS422)	926
13.3.3.2.7	Set_OutputEdgeDarkLight_ETH (OUTEDGELH_ETH)	926
13.3.3.2.8	Get_OutputEdgeDarkLight_ETH (OUTEDGELH_ETH)	926
13.3.3.2.9	Set_OutputDiameter_RS422 (OUTDIA_RS422)	927
13.3.3.2.10	Get_OutputDiameter_RS422 (OUTDIA_RS422)	927
13.3.3.2.11	Set_OutputDiameter_ETH (OUTDIA_ETH)	928
13.3.3.2.12	Get_OutputDiameter_ETH (OUTDIA_ETH)	928
13.3.3.2.13	Set_OutputGap_RS422 (OUTGAP_RS422)	929
13.3.3.2.14	Get_OutputGap_RS422 (OUTGAP_RS422)	930
13.3.3.2.15	Set_OutputGap_ETH (OUTGAP_ETH)	930
13.3.3.2.16	Get_OutputGap_ETH (OUTGAP_ETH)	931
13.3.3.2.17	Set_OutputSegment_RS422 (OUTSEG_RS422)	932
13.3.3.2.18	Get_OutputSegment_RS422 (OUTSEG_RS422)	936
13.3.3.2.19	Set_OutputSegment_ETH (OUTSEG_ETH)	940
13.3.3.2.20	Get_OutputSegment_ETH (OUTSEG_ETH)	944
13.3.3.2.21	Set_OutputStatistic_RS422 (OUTSTATISTIC_RS422)	948
13.3.3.2.22	Get_OutputStatistic_RS422 (OUTSTATISTIC_RS422)	949
13.3.3.2.23	Set_OutputStatistic_ETH (OUTSTATISTIC_ETH)	950
13.3.3.2.24	Get_OutputStatistic_ETH (OUTSTATISTIC_ETH)	951
13.3.3.2.25	Set_OutputAdditional_RS422 (OUTADD_RS422)	952
13.3.3.2.26	Get_OutputAdditional_RS422 (OUTADD_RS422)	953
13.3.3.2.27	Set_OutputAdditional_ETH (OUTADD_ETH)	954
13.3.3.2.28	Get_OutputAdditional_ETH (OUTADD_ETH)	954
13.3.3.2.29	Set_OutputVideo_ETH (OUTVID_ETH)	955
13.3.3.2.30	Get_OutputVideo_ETH (OUTVID_ETH)	956
13.3.3.3	Switching outputs	957
13.3.3.3.1	Set_ErrorOutput<n> (ERRORROUT<n>)	957
13.3.3.3.2	Get_ErrorOutput<n> (ERRORROUT<n>)	957
13.3.3.3.3	Set_ErrorLimit (ERRORLIMIT)	958
13.3.3.3.4	Get_ErrorLimit (ERRORLIMIT)	959
13.3.3.3.5	Set_ErrorOut<n>HoldTime (ERROROUTHOLD<n>)	960
13.3.3.3.6	Get_ErrorOut<n>HoldTime (ERROROUTHOLD<n>)	960
13.3.3.3.7	Set_ErrorLevelOut<n> (ERRORLEVELOUT<n>)	961
13.3.3.3.8	Get_ErrorLevelOut<n> (ERRORLEVELOUT<n>)	961
13.3.3.4	Analog output	961
13.3.3.4.1	Set_AnalogOutput (ANALOGOUT)	961
13.3.3.4.2	Get_AnalogOutput (ANALOGOUT)	962
13.3.3.4.3	Set_AnalogRange (ANALOGRANGE)	963

13.3.3.4.4	Get_AnalogRange (ANALOG RANGE)	963
13.3.3.4.5	Set_AnalogScale (ANALOG SCALE)	964
13.3.3.4.6	Get_AnalogScale (ANALOG SCALE)	964
13.4	Commands for ODC2600	965
13.4.1	General commands	965
13.4.1.1	General	965
13.4.1.1.1	Get_Info (INFO)	965
13.4.1.1.2	Reset_Boot (RESET)	967
13.4.1.2	Triggering	967
13.4.1.2.1	Triggermode_Reset (TRIGGERMODE RESET)	967
13.4.1.2.2	Triggermode_Trigger (TRIGGERMODE TRIGGER)	967
13.4.1.3	Parameter management	967
13.4.1.3.1	Save_OptionData (SAVE OPT RAM TO FLASH)	967
13.4.1.3.2	Save_MeasProgData (SAVE MPR RAM TO FLASH)	967
13.4.2	Measurement	967
13.4.2.1	Choose_MeasProg (CHOOSE MP)	967
13.4.2.2	Switch_Edge (SWITCH EDGE)	968
13.4.2.3	Write_OptionData (WR OPT TO RAM)	969
13.4.2.4	Read_OptionData (RD OPT RAM)	972
13.4.2.5	Update_OptionData	975
13.4.2.6	Write_MeasProgData (WR MPR TO RAM)	978
13.4.2.7	Read_MeasProgData (RD MPR RAM)	983
13.4.2.8	Set_LightRef (SET LIGHT REFERENCE TUNING)	987
13.4.2.9	Reset_LightRef (RESET LIGHT REFERENCE TUNING)	987
13.4.2.10	Read_MinMax (RD MINMAX)	987
13.4.2.11	Read_MinMax_Reset (RD MINMAX RESET)	988
13.4.3	Data output	989
13.4.3.1	General	989
13.4.3.1.1	Dat_Out_Off (STOP)	989
13.4.3.1.2	Dat_Out_On (START)	989
14	Commands for eddyNCDT sensors	990
14.1	Commands for DT306x/7x	990
14.1.1	User Level	990
14.1.1.1	Logout (LGO)	990
14.1.1.2	Login (LGI)	990
14.1.1.3	Set_Password (PWD)	991
14.1.2	Measurement	991
14.1.2.1	Get_SampleTime (STI?)	991
14.1.2.2	Get_Measure1 (GMD1)	991
14.1.2.3	Get_Measure2 (GMD2)	991
14.1.3	Data output	992
14.1.3.1	ChannelStatus (CHS)	992
14.1.3.2	Get_RawDataRange (MDF)	992
14.1.3.3	Get_RawDataRanges	993
14.1.4	Interfaces	993
14.1.4.1	Set_DataPort (SDP)	993
14.1.4.2	Get_DataPort (GDP)	993
14.1.4.3	Set_IPConfiguration (IPS)	994
14.1.4.4	Get_IPConfiguration (IPS?)	994
14.1.4.5	Set_AppLanguage (LNG)	995
14.1.4.6	Get_AppLanguage (LNG?)	995
14.1.5	Sensor interface	995

14.1.5.1	Set_Parameter (PAR)	995
14.1.5.2	Get_Parameter (PAR?)	996
14.1.6	Information	996
14.1.6.1	Get_Status (STS)	996
14.1.6.2	Get_Version (VER)	997
14.1.6.3	Get_ChannelInfo (CHI)	997
14.1.6.4	Get_ChannelInfos	998
14.1.6.5	Get_ControllerInfo (COI)	999
14.1.7	Internal commands	999
14.1.7.1	Update_Firmware	999
14.2	Commands for DT3100	1000
14.2.1	Information	1000
14.2.1.1	Get_Status (STS)	1000
14.2.1.2	Get_Settings (SET)	1000
14.2.1.3	Get_Error (ERR)	1002
14.2.1.4	Get_ControllerInfo (IND)	1002
14.2.1.5	Get_SensorInfo (SEN)	1003
14.2.1.6	Get_CableInfo1 (CB1)	1003
14.2.1.7	Get_CableInfo2 (CB2)	1004
14.2.1.8	DetectSensorChange (DSC)	1005
14.2.1.9	Set_TextField (ETF)	1005
14.2.1.10	Get_TextField (ETF?)	1005
14.2.2	Parameter Management	1005
14.2.2.1	Save_Settings (SSE)	1005
14.2.2.2	Read_Settings (RSE)	1005
14.2.2.3	Set_Defaults (DSE)	1005
14.2.3	Display	1007
14.2.3.1	Set_AppLanguage (LNG)	1007
14.2.3.2	Get_AppLanguage (LNG?)	1007
14.2.3.3	Set_AppYScale (SCA)	1007
14.2.3.4	Get_AppYScale (SCA?)	1008
14.2.3.5	Set_AppDispMode (DMD)	1008
14.2.3.6	Get_AppDispMode (DMD?)	1008
14.2.4	Measurement	1009
14.2.4.1	Set_SRIndex (SRA)	1009
14.2.4.2	Get_SRIndex (SRA?)	1009
14.2.4.3	Get_Measure (GMD)	1009
14.2.4.4	Set_MeasureCnt (VTT)	1009
14.2.4.5	Get_MeasureCnt (VTT?)	1009
14.2.4.6	Set_MeasureMode (MMD)	1010
14.2.4.7	Get_MeasureMode (MMD?)	1010
14.2.4.8	Set_AvrType (AVT)	1010
14.2.4.9	Get_AvrType (AVT?)	1010
14.2.4.10	Set_AvrNbr (AVN)	1011
14.2.4.11	Get_AvrNbr (AVN?)	1011
14.2.4.12	Get_ControllerTemp (GCT)	1011
14.2.4.13	Get_SensorTemp (GST)	1011
14.2.5	Linearization	1012
14.2.5.1	Get_CalibState (CST)	1012
14.2.5.2	Set_CalibTarget (TAR)	1012
14.2.5.3	Get_CalibTarget (TAR?)	1012
14.2.5.4	Get_CalibProgress (BSY)	1012
14.2.5.5	Set_CalibStart (SMR)	1013

14.2.5.6 Set_CalibEnd (EMR)	1013
14.2.5.7 Set_CalibMid (MMR)	1013
14.2.5.8 Exit_Calib (ECA)	1013
14.2.5.9 Set_FactoryCalib (FCA)	1013
14.2.5.10 Set_Poti (WPT)	1014
14.2.5.11 Get_Poti (RPT)	1014
14.2.5.12 Save_Poti (SPT)	1015
14.2.6 Internal commands	1015
14.2.6.1 Update_Firmware	1015
14.2.6.2 Get_BootLoaderMode (BOT)	1015
14.2.6.3 Get_BootLoaderVer (VBL)	1015
14.2.6.4 Get_FirmwareVer (VFW)	1016
14.2.6.5 Start_BootLoader (SBL)	1016
14.2.6.6 Erase_Flash (EFL)	1016
14.2.6.7 Write_Sensor (WDS)	1016
14.2.6.8 Read_Sensor (RDS)	1016
14.2.6.9 Write_AnalogEEPROM (WA1+2)	1016
14.2.6.10 Read_AnalogEEPROM (RA1+2)	1017
14.2.6.11 Write_SensorParam (WSP)	1017
14.2.6.12 Read_SensorParam (RSP)	1017
14.2.6.13 Write_AnalogParam (WP1+2)	1017
14.2.6.14 Read_AnalogParam (RP1+2)	1017
14.2.6.15 Exec_SysCalibOff (SCO)	1018
14.2.6.16 Exec_SysCalibGain (SCG)	1018
14.2.6.17 Write_Index (WRI)	1018
14.2.6.18 Read_Index (RDI)	1018
14.2.6.19 Write_IndexDigital (WRD)	1018
14.2.6.20 Read_IndexDigital (RDD)	1018
14.2.6.21 Get_ButtonState (CBT)	1019
14.2.6.22 Set_Multiplexer (MUX)	1019
15 Commands for capaNCDT and combiSENSOR sensors	1020
15.1 Commands for DT6100	1020
15.1.1 Measurement	1020
15.1.1.1 Set_SRIndex (SRA)	1020
15.1.1.2 Get_SRIndex (SRA?)	1021
15.1.1.3 Set_Trigger (TRG)	1021
15.1.1.4 Get_Trigger (TRG?)	1021
15.1.1.5 Get_Measure (GMD)	1021
15.1.1.6 Set_AvrType (AVT)	1022
15.1.1.7 Get_AvrType (AVT?)	1022
15.1.1.8 Set_AvrNbr (AVN)	1022
15.1.1.9 Get_AvrNbr (AVN?)	1022
15.1.2 Linearization	1023
15.1.2.1 Set_LinMode (LIN)	1023
15.1.2.2 Get_LinMode (LIN?)	1023
15.1.2.3 Set_LinPoint (SLP)	1023
15.1.2.4 Get_LinPoint (GLP)	1024
15.1.3 Information	1024
15.1.3.1 Get_Status (STS)	1024
15.1.3.2 Get_Version (VER)	1025
15.1.4 Parameter management	1025
15.1.4.1 Save_Setup (SSU)	1025

15.1.4.2	Read_Setup (RSU)	1025
15.1.4.3	Factory_Defaults (FDE)	1026
15.1.5	Internal commands	1027
15.1.5.1	Set_Coefficient (SCO)	1027
15.1.5.2	Get_Coefficient (GCO)	1027
15.1.6	Special commands	1027
15.1.6.1	Use_Defaults	1027
15.1.6.2	Get_DrvSetting	1028
15.2	Commands for DT6120	1028
15.2.1	Measurement	1028
15.2.1.1	Set_Samplerate	1028
15.2.1.2	Get_Samplerate	1029
15.2.1.3	Set_Averaging	1029
15.2.1.4	Get_Averaging	1030
15.2.1.5	Get_Measure	1030
15.2.2	Data output	1030
15.2.2.1	Set_Range	1030
15.2.3	Interfaces	1031
15.2.3.1	Set_Baudrate	1031
15.2.3.2	Get_Baudrate	1031
15.2.3.3	Set_SensorAddress	1032
15.2.4	Information	1032
15.2.4.1	Get_SensorInfo	1032
15.2.4.2	Get_ControllerInfo	1033
15.2.4.3	Read_AllBlocks	1033
15.3	Commands for DT62xx, KSS64xx and DT65xx	1034
15.3.1	User Level	1034
15.3.1.1	Logout (LGO)	1034
15.3.1.2	Login (LGI)	1034
15.3.1.3	Set_Password (PWD)	1034
15.3.2	Measurement	1035
15.3.2.1	Set_SRIndex (SRA)	1035
15.3.2.2	Get_SRIndex (SRA?)	1035
15.3.2.3	Set_SampleTime (STI)	1036
15.3.2.4	Get_SampleTime (STI?)	1036
15.3.2.5	Set_Trigger (TRG)	1036
15.3.2.6	Get_Trigger (TRG?)	1037
15.3.2.7	Get_Measure (GMD)	1037
15.3.2.8	Set_AvrType (AVT)	1037
15.3.2.9	Get_AvrType (AVT?)	1037
15.3.2.10	Set_AvrNbr (AVN)	1038
15.3.2.11	Get_AvrNbr (AVN?)	1038
15.3.2.12	Set_AnalogLowPass (ALP)	1038
15.3.2.13	Get_AnalogLowPass (ALP?)	1038
15.3.3	Data output	1039
15.3.3.1	ChannelStatus (CHS)	1039
15.3.3.2	Set_Channel (CHT)	1039
15.3.3.3	Get_Channel (CHT?)	1039
15.3.3.4	Set_Range (MRA)	1039
15.3.3.5	Get_RawDataRange (MDF)	1040
15.3.3.6	Get_RawDataRanges	1040
15.3.4	Display	1041
15.3.4.1	Set_Display (DIS)	1041

15.3.4.2	Get_Display (DIS?)	1041
15.3.5	Math	1042
15.3.5.1	Set_MathFunction (SMF)	1042
15.3.5.2	Get_MathFunction (GMF)	1042
15.3.5.3	Clr_MathFunction (CMF)	1043
15.3.5.4	Set_ThicknessMeasure (THM)	1043
15.3.5.5	Get_ThicknessMeasure (THM?)	1044
15.3.6	Interfaces	1044
15.3.6.1	Set_DataPort (SDP)	1044
15.3.6.2	Get_DataPort (GDP)	1045
15.3.6.3	Set_IPConfiguration (IPS)	1045
15.3.6.4	Get_IPConfiguration (IPS?)	1046
15.3.6.5	Set_EthernetMode (IFC)	1046
15.3.6.6	Get_EthernetMode (IFC?)	1047
15.3.6.7	Set_AppLanguage (LNG)	1047
15.3.6.8	Get_AppLanguage (LNG?)	1047
15.3.7	Linearization	1047
15.3.7.1	Set_LinMode (LIN)	1047
15.3.7.2	Get_LinMode (LIN?)	1048
15.3.7.3	Set_LinPoint (SLP)	1048
15.3.7.4	Set_LinPointOffline (SLP)	1049
15.3.7.5	Get_LinPoint (GLP)	1049
15.3.8	Information	1050
15.3.8.1	Get_Status (STS)	1050
15.3.8.2	Get_Version (VER)	1052
15.3.8.3	Get_ChannelInfo (CHI)	1052
15.3.8.4	Get_ChannelInfos	1053
15.3.8.5	Get_ControllerInfo (COI)	1054
15.3.9	Parameter management	1055
15.3.9.1	Save_Setup (SSU)	1055
15.3.9.2	Read_Setup (RSU)	1055
15.3.9.3	Factory_Defaults (FDE)	1055
15.3.10	Internal commands	1057
15.3.10.1	Update_Firmware	1057
15.3.11	Special commands	1057
15.3.11.1	Use_Defaults	1057
15.3.11.2	Get_DrvSetting	1057
15.4	Commands for KSS6380	1058
15.4.1	Measurement	1058
15.4.1.1	Set_SRIndex (SRA)	1058
15.4.1.2	Get_SRIndex (SRA?)	1058
15.4.1.3	Set_Trigger (TRG)	1059
15.4.1.4	Get_Trigger (TRG?)	1059
15.4.1.5	Get_Measure (GMD)	1059
15.4.1.6	Set_AvrType (AVT)	1059
15.4.1.7	Get_AvrType (AVT?)	1059
15.4.1.8	Set_AvrNbr (AVN)	1060
15.4.1.9	Get_AvrNbr (AVN?)	1060
15.4.2	Data output	1060
15.4.2.1	ChannelStatus (CHS)	1060
15.4.2.2	Set_Channel (CHT)	1060
15.4.2.3	Get_Channel (CHT?)	1061
15.4.3	Math	1061

15.4.3.1	Set_MathFunction (SMF)	1061
15.4.3.2	Get_MathFunction (GMF)	1062
15.4.3.3	Clr_MathFunction (CMF)	1062
15.4.4	Information	1062
15.4.4.1	Get_Status (STS)	1062
15.4.4.2	Get_Version (VER)	1063
15.4.5	Parameter management	1063
15.4.5.1	Save_Setup (SSU)	1063
15.4.5.2	Read_Setup (RSU)	1064
15.4.5.3	Factory_Defaults (FDE)	1064
15.4.6	Internal commands	1065
15.4.6.1	Set_TempCoeff (STC)	1065
15.4.6.2	Get_TempCoeff (GTC)	1065
15.4.6.3	Get_UnlinEddyVal (GUE)	1066
15.4.6.4	Set_LinPoint (SLP)	1066
15.4.6.5	Get_LinPoint (GLP)	1067
15.4.7	Special commands	1067
15.4.7.1	Use_Defaults	1067
15.4.7.2	Get_DrvSetting	1068
16	Commands for colorCONTROL sensors	1069
16.1	Commands for ACS7000	1069
16.1.1	General commands	1069
16.1.1.1	General	1069
16.1.1.1.1	Get_Help (HELP)	1069
16.1.1.1.2	Get_Info (GETINFO)	1070
16.1.1.1.3	Set_Echo (ECHO)	1071
16.1.1.1.4	Get_Echo (ECHO)	1071
16.1.1.1.5	Get_AllParameters (PRINT)	1071
16.1.1.1.6	Set_SyncMode (SYNC)	1091
16.1.1.1.7	Get_SyncMode (SYNC)	1091
16.1.1.1.8	Reset_Boot (RESET)	1091
16.1.1.1.9	Set_Keylock (KEYLOCK)	1091
16.1.1.1.10	Get_Keylock (KEYLOCK)	1092
16.1.1.2	User Level	1092
16.1.1.2.1	Logout (LOGOUT)	1092
16.1.1.2.2	Login (LOGIN)	1092
16.1.1.2.3	Get_UserLevel (GETUSERLEVEL)	1093
16.1.1.2.4	Set_DefaultUser (STDUSER)	1093
16.1.1.2.5	Get_DefaultUser (STDUSER)	1093
16.1.1.2.6	Set_Password (PASSWD)	1093
16.1.1.3	Sensor	1094
16.1.1.3.1	Set_Observer (OBSERVER)	1094
16.1.1.3.2	Get_Observer (OBSERVER)	1094
16.1.1.3.3	Set_LightSource (LQSRC)	1094
16.1.1.3.4	Get_LightSource (LQSRC)	1095
16.1.1.3.5	Set_LEDControl (LEDCTRL)	1095
16.1.1.3.6	Get_LEDControl (LEDCTRL)	1095
16.1.1.3.7	Set_LEDIntensityColdWhite (LEDKW)	1096
16.1.1.3.8	Get_LEDIntensityColdWhite (LEDKW)	1096
16.1.1.3.9	Set_LEDIntensityGreen (LEDGR)	1096
16.1.1.3.10	Get_LEDIntensityGreen (LEDGR)	1096
16.1.1.3.11	Set_LEDIntensityWarmWhite (LEDWW)	1096

16.1.1.3.12	Get_LEDIntensityWarmWhite (LEDWW)	1097
16.1.1.3.13	Set_LEDIntensityViolet (LEDUV)	1097
16.1.1.3.14	Get_LEDIntensityViolet (LEDUV)	1097
16.1.1.3.15	AutoLEDAdjustment (AUTOLEDADJ)	1097
16.1.1.3.16	DarkCorr (DARKCORR)	1097
16.1.1.3.17	LightCorr (LIGHTCORR)	1097
16.1.1.4	Triggering	1098
16.1.1.4.1	Set_TriggerMode (TRIGGER)	1098
16.1.1.4.2	Get_TriggerMode (TRIGGER)	1098
16.1.1.4.3	Set_TriggerLevel (TRIGGERLEVEL)	1098
16.1.1.4.4	Get_TriggerLevel (TRIGGERLEVEL)	1098
16.1.1.4.5	Set_TriggerCount (TRIGGERCOUNT)	1099
16.1.1.4.6	Get_TriggerCount (TRIGGERCOUNT)	1099
16.1.1.4.7	Software_Trigger (TRIGGERSW)	1099
16.1.1.5	Interfaces	1099
16.1.1.5.1	Set_IPConfiguration (IPCONFIG)	1099
16.1.1.5.2	Get_IPConfiguration (IPCONFIG)	1100
16.1.1.5.3	Set_IPDataTransferMode (MEASTRANSFER)	1101
16.1.1.5.4	Get_IPDataTransferMode (MEASTRANSFER)	1101
16.1.1.5.5	Set_EthernetMode (ETHERMODE)	1102
16.1.1.5.6	Get_EthernetMode (ETHERMODE)	1102
16.1.1.5.7	Set_Baudrate (BAUDRATE)	1102
16.1.1.5.8	Get_Baudrate (BAUDRATE)	1103
16.1.1.5.9	Set_DigitalOutColorFormat (COLOROUT_FORMAT)	1103
16.1.1.5.10	Get_DigitalOutColorFormat (COLOROUT_FORMAT)	1103
16.1.1.5.11	Set_DigitalOutBinFormat (BIN_FORMAT)	1104
16.1.1.5.12	Get_DigitalOutBinFormat (BIN_FORMAT)	1104
16.1.1.6	Parameter management	1104
16.1.1.6.1	Save_Parameters (STORE)	1104
16.1.1.6.2	Load_Parameters (READ)	1104
16.1.1.6.3	Set_Default (SETDEFAULT)	1105
16.1.2	Measurement	1105
16.1.2.1	General	1105
16.1.2.1.1	Set_MeasureMode (MEASMODE)	1105
16.1.2.1.2	Get_MeasureMode (MEASMODE)	1106
16.1.2.1.3	Set_ShutterMode (SHUTTERMODE)	1106
16.1.2.1.4	Get_ShutterMode (SHUTTERMODE)	1106
16.1.2.1.5	Get_Video (GETVIDEO)	1106
16.1.2.1.6	Set_Samplerate (MEASRATE)	1107
16.1.2.1.7	Get_Samplerate (MEASRATE)	1108
16.1.2.1.8	Set_DeltaMode (DELTAMODE)	1108
16.1.2.1.9	Get_DeltaMode (DELTAMODE)	1108
16.1.2.1.10	Set_DeltaKL (DELTA_KL)	1108
16.1.2.1.11	Get_DeltaKL (DELTA_KL)	1109
16.1.2.1.12	Set_DeltaKC (DELTA_KC)	1109
16.1.2.1.13	Get_DeltaKC (DELTA_KC)	1109
16.1.2.1.14	Set_DeltaKH (DELTA_KH)	1109
16.1.2.1.15	Get_DeltaKH (DELTA_KH)	1109
16.1.2.2	Color database	1110
16.1.2.2.1	Get_ColorTable (COLORTABLE)	1110
16.1.2.2.2	Edit_Color (COLORNEW)	1112
16.1.2.2.3	Set_ColorDescription (COLORDESCR)	1113
16.1.2.2.4	Get_ColorDescription (COLORDESCR)	1113

16.1.2.2.5	Set_ColorThresholds (THRESHOLDS)	1114
16.1.2.2.6	Get_ColorThresholds (THRESHOLDS)	1114
16.1.2.2.7	Set_ColorSpace (COLORSPACE)	1115
16.1.2.2.8	Get_ColorSpace (COLORSPACE)	1115
16.1.2.2.9	Move_Color (MOVECOLOR)	1115
16.1.2.2.10	Reset_ColorMapping (RESETMAPPING)	1116
16.1.2.2.11	Delete_Color (COLORDELETE)	1116
16.1.2.2.12	Clear_ColorTable	1116
16.1.2.3	Measurement value processing	1116
16.1.2.3.1	Set_VideoAverage (VSAVERAGE)	1116
16.1.2.3.2	Get_VideoAverage (VSAVERAGE)	1117
16.1.2.3.3	Set_Averaging (AVERAGE)	1117
16.1.2.3.4	Get_Averaging (AVERAGE)	1118
16.1.2.3.5	Set_StatisticSignal (STATISTICSIGNAL)	1119
16.1.2.3.6	Get_StatisticSignal (STATISTICSIGNAL)	1119
16.1.2.3.7	Set_StatisticDepth (STATISTICDEPTH)	1120
16.1.2.3.8	Get_StatisticDepth (STATISTICDEPTH)	1120
16.1.2.3.9	Reset_Statistic (RESETSTATISTIC)	1120
16.1.3	Data output	1121
16.1.3.1	General	1121
16.1.3.1.1	Set_DataOutInterface (OUTPUT)	1121
16.1.3.1.2	Get_DataOutInterface (OUTPUT)	1121
16.1.3.1.3	Set_Resampling (OUTREDUCE)	1121
16.1.3.1.4	Get_Resampling (OUTREDUCE)	1122
16.1.3.1.5	Set_HoldLastValid (OUTHOLD)	1122
16.1.3.1.6	Get_HoldLastValid (OUTHOLD)	1123
16.1.3.2	Selected measurement values	1123
16.1.3.2.1	Set_OutputVideo_ETH (OUTVIDEO)	1123
16.1.3.2.2	Get_OutputVideo_ETH (OUTVIDEO)	1124
16.1.3.2.3	Set_OutputColor_ETH (OUTCOLOR_ETH)	1124
16.1.3.2.4	Get_OutputColor_ETH (OUTCOLOR_ETH)	1125
16.1.3.2.5	Set_OutputColor_RS422 (OUTCOLOR_RS422)	1126
16.1.3.2.6	Get_OutputColor_RS422 (OUTCOLOR_RS422)	1127
16.1.3.2.7	Set_DistanceMode (DISTANCEMODE)	1128
16.1.3.2.8	Get_DistanceMode (DISTANCEMODE)	1129
16.1.3.2.9	Set_OutputDistance_ETH (OUTDIST_ETH)	1129
16.1.3.2.10	Get_OutputDistance_ETH (OUTDIST_ETH)	1131
16.1.3.2.11	Set_OutputDistance_RS422 (OUTDIST_RS422)	1134
16.1.3.2.12	Get_OutputDistance_RS422 (OUTDIST_RS422)	1134
16.1.3.2.13	Set_OutputStatistic_ETH (OUTSTATISTIC_ETH)	1135
16.1.3.2.14	Get_OutputStatistic_ETH (OUTSTATISTIC_ETH)	1135
16.1.3.2.15	Set_OutputStatistic_RS422 (OUTSTATISTIC_RS422)	1136
16.1.3.2.16	Get_OutputStatistic_RS422 (OUTSTATISTIC_RS422)	1136
16.1.3.2.17	Set_DigitalOutDistance (OUTDIST_COLOROUT)	1137
16.1.3.2.18	Get_DigitalOutDistance (OUTDIST_COLOROUT)	1137
16.1.3.2.19	Set_OutputStatus_ETH (OUTSTATUS_ETH)	1137
16.1.3.2.20	Get_OutputStatus_ETH (OUTSTATUS_ETH)	1139
16.1.3.2.21	Set_OutputStatus_RS422 (OUTSTATUS_RS422)	1140
16.1.3.2.22	Get_OutputStatus_RS422 (OUTSTATUS_RS422)	1142
16.2	Commands for CFO100/200	1143
16.2.1	Access	1144
16.2.1.1	Get_AccessControl (GET /api/access)	1144
16.2.1.2	Get_UserRoles (GET /api/access/roles)	1144

16.2.1.3	Add_User (POST /api/access/users)	1145
16.2.1.4	Set_User (PUT /api/access/users/{name})	1146
16.2.1.5	Get_User (GET /api/access/users/{name})	1147
16.2.1.6	Get_UserList (GET /api/access/users)	1147
16.2.1.7	Logout (DELETE /api/access/login, access logout)	1148
16.2.1.8	Login (POST /api/access/login, access login)	1148
16.2.1.9	Get_CurrentUser (GET /api/access/login, access session)	1149
16.2.1.10	Delete_User (DELETE /api/access/users/{name})	1150
16.2.1.11	Delete_UserList (DELETE /api/access/users)	1150
16.2.2	Defaults	1150
16.2.2.1	Add_Defaults (POST /api/defaults)	1150
16.2.2.2	Set_Defaults (PUT /api/defaults/{itemId}, matcher default set <hold_time tolerance>)	1151
16.2.2.3	Get_Defaults (GET /api/defaults/{itemId}, matcher default show)	1153
16.2.2.4	Get_DefaultsList (GET /api/defaults)	1155
16.2.2.5	Delete_Defaults (DELETE /api/defaults/{itemId})	1156
16.2.2.6	Delete_DefaultsList (DELETE /api/defaults)	1156
16.2.3	Device	1157
16.2.3.1	Get_Device (GET /api/device, device show)	1157
16.2.4	Firmware	1158
16.2.4.1	Get_CurrentFirmware (GET /api/firmware, firmware version)	1158
16.2.4.2	Get_RecoveryFirmware (GET /api/firmware/recovery, firmware recovery show)	1159
16.2.4.3	Get_Firmware (GET /api/firmware/images/{itemId})	1160
16.2.4.4	Get_FirmwareList (GET /api/firmware/images)	1161
16.2.4.5	Activate_Firmware (POST /api/firmware/images/{itemId}/apply)	1162
16.2.4.6	Set_FirmwareAsRecovery (POST /api/firmware/recovery/upgrade-from-current, firmware recovery upgrade)	1162
16.2.4.7	Set_FirmwareSettings (PUT /api/firmware/settings)	1162
16.2.4.8	Get_FirmwareSettings (GET /api/firmware/settings)	1163
16.2.4.9	Get_FirmwareStatus (GET /api/firmware/status)	1163
16.2.4.10	Delete_Firmware (DELETE /api/firmware/images/{itemId})	1163
16.2.5	Network	1163
16.2.5.1	Set_NetworkInterface (PUT /api/network/interfaces/{name}, network select INTERFACE set <ipv4 ipv6> <disabled static dhcp auto>)	1163
16.2.5.2	Get_NetworkInterface (GET /api/network/interfaces/{name}, network select INTERFACE show)	1168
16.2.5.3	Get_NetworkInterfaces (GET /api/network/interfaces, network list)	1170
16.2.5.4	Delete_Network (DELETE /api/network, network reset)	1172
16.2.6	Peripherals	1172
16.2.6.1	Set_Keypad (PUT /api/peripherals/keypad, keypad lock)	1172
16.2.6.2	Get_Keypad (GET /api/peripherals/keypad, keypad show)	1173
16.2.6.3	Exec_KeyPress (POST /api/peripherals/keypad/inputs/{name}/{event})	1173
16.2.6.4	Get_KeypadEvents (GET /api/peripherals/keypad/events)	1173
16.2.6.5	Get_KeypadInputs (GET /api/peripherals/keypad/inputs)	1174
16.2.6.6	Set_Outputs (PUT /api/peripherals/outputs)	1175
16.2.6.7	Get_Outputs (GET /api/peripherals/outputs)	1176
16.2.6.8	Set_RS232Settings (PUT /api/peripherals/rs232)	1176
16.2.6.9	Get_RS232Settings (GET /api/peripherals/rs232)	1177
16.2.6.10	Set_USBSettings (PUT /api/peripherals/usb)	1178
16.2.6.11	Get_USBSettings (GET /api/peripherals/usb)	1178
16.2.6.12	Get_TriggerSources (GET /api/peripherals/trigger-sources)	1179

16.2.6.13	<code>Get_KeypadOutput (GET /api/peripherals/keypad/outputs/{name})</code>	1180
16.2.6.14	<code>Get_KeypadOutputList (GET /api/peripherals/keypad/outputs)</code>	1181
16.2.7	<code>Sensor</code>	1183
16.2.7.1	<code>Capabilities</code>	1183
16.2.7.1.1	<code>Get_SensorCapabilities (GET /api/sensor/capabilities)</code>	1183
16.2.7.2	<code>Color spaces</code>	1187
16.2.7.2.1	<code>Get_SensorColorSpace (GET /api/sensor/colorsplaces/{space_id}, sensor colorspace show)</code>	1187
16.2.7.2.2	<code>Get_SensorColorSpaceList (GET /api/sensor/colorsplaces, sensor colorspace list)</code>	1189
16.2.7.3	<code>Detection profiles</code>	1190
16.2.7.3.1	<code>Add_DetectionProfile (POST /api/sensor/detection-profiles)</code>	1190
16.2.7.3.2	<code>Set_DetectionProfile (PUT /api/sensor/detection-profiles/{itemId}, sensor colorspace set COLORSPACE)</code>	1195
16.2.7.3.3	<code>Get_DetectionProfile (GET /api/sensor/detection-profiles/{itemId})</code>	1200
16.2.7.3.4	<code>Get_DetectionProfileList (GET /api/sensor/detection-profiles)</code>	1204
16.2.7.3.5	<code>Select_DetectionProfile (PUT /api/sensor/detection-profiles)</code>	1208
16.2.7.3.6	<code>Exec_DetectionProfileAutoGain (POST /api/sensor/detection-profiles/{itemId}/autogain, sensor autogain)</code>	1212
16.2.7.3.7	<code>Exec_DetectionProfileWhiteRef (POST /api/sensor/detection-profiles/{itemId}/white-reference, sensor white-reference sample)</code>	1215
16.2.7.3.8	<code>Get_DetectionProfileWhiteRef (GET /api/sensor/detection-profiles/{itemId}/white-reference)</code>	1215
16.2.7.3.9	<code>Delete_DetectionProfileWhiteRef (DELETE /api/sensor/detection-profiles/{itemId}/white-reference, sensor white-reference reset)</code>	1216
16.2.7.3.10	<code>Delete_DetectionProfile (DELETE /api/sensor/detection-profiles/{itemId})</code>	1216
16.2.7.3.11	<code>Delete_DetectionProfileList (DELETE /api/sensor/detection-profiles)</code>	1216
16.2.7.4	<code>Matchers</code>	1216
16.2.7.4.1	<code>Add_SensorMatcher (POST /api/sensor/matchers, matcher add)</code>	1216
16.2.7.4.2	<code>Set_SensorMatcher (PUT /api/sensor/matchers/{itemId}, matcher select MATCHER set <name hold_time output_pattern tolerance>)</code>	1219
16.2.7.4.3	<code>Get_SensorMatcher (GET /api/sensor/matchers/{itemId}, matcher select MATCHER show)</code>	1223
16.2.7.4.4	<code>Get_SensorMatcherList (GET /api/sensor/matchers, matcher list)</code>	1225
16.2.7.4.5	<code>Get_SensorMatcherIDList</code>	1226
16.2.7.4.6	<code>Delete_SensorMatcher (DELETE /api/sensor/matchers/{itemId}, matcher select MATCHER remove)</code>	1227
16.2.7.4.7	<code>Delete_SensorMatcherList (DELETE /api/sensor/matchers, matcher remove all)</code>	1227
16.2.7.5	<code>Detectables</code>	1227
16.2.7.5.1	<code>Learn_SensorDetectable (POST /api/sensor/detectables, matcher select MATCHER detectable add)</code>	1227
16.2.7.5.2	<code>Add_SensorDetectable (POST /api/sensor/detectables, matcher select MATCHER detectable add POSITION)</code>	1228
16.2.7.5.3	<code>Get_SensorDetectableList (GET /api/sensor/detectables, matcher select MATCHER detectable list)</code>	1229
16.2.7.5.4	<code>Set_SensorDetectable (PUT /api/sensor/detectables/{itemId}, matcher select MATCHER detectable select DETECTABLE set position POSITION)</code>	1230

16.2.7.5.5	Get_SensorDetectable (GET /api/sensor/detectables/{itemId}, matcher select MATCHER detectable select DETECTABLE show)	1230
16.2.7.5.6	Delete_SensorDetectable (DELETE /api/sensor/detectables/{itemId}, matcher select MATCHER detectable select DETECTABLE re- move)	1231
16.2.7.5.7	Delete_SensorDetectableList (DELETE /api/sensor/detectables, matcher select MATCHER detectable remove all)	1232
16.2.7.6	Action triggers	1232
16.2.7.6.1	Get_Action (GET /api/actions/{itemId})	1232
16.2.7.6.2	Get_ActionList (GET /api/actions)	1233
16.2.7.6.3	Exec_Action (POST /api/actions/{itemId}/execute)	1234
16.2.7.6.4	Add_ActionTrigger (POST /api/sensor/action-triggers)	1239
16.2.7.6.5	Set_ActionTrigger (PUT /api/sensor/action-triggers/{itemId})	1243
16.2.7.6.6	Get_ActionTrigger (GET /api/sensor/action-triggers/{itemId})	1248
16.2.7.6.7	Get_ActionTriggerList (GET /api/sensor/action-triggers)	1251
16.2.7.6.8	Delete_ActionTrigger (DELETE /api/sensor/action-triggers/{itemId})	1253
16.2.7.6.9	Delete_ActionTriggerList (DELETE /api/sensor/action-triggers)	1254
16.2.7.7	Samples	1254
16.2.7.7.1	Get_SensorSample (GET /api/sensor/samples/current, sam- ple show)	1254
16.2.7.7.2	Get_SensorSampleList (GET /api/sensor/samples)	1256
16.2.7.7.3	Start_SensorSampleStream (sample stream)	1258
16.2.7.7.4	Abort_SensorSampleStream (ETX, CTRL-C)	1258
16.2.8	Settings	1258
16.2.8.1	Update_Settings (POST /api/settings)	1258
16.2.8.2	Set_Settings (PUT /api/settings)	1259
16.2.8.3	Get_Settings (GET /api/settings)	1259
16.2.8.4	Delete_Settings (DELETE /api/settings, system settings reset)	1259
16.2.9	System	1259
16.2.9.1	Set_System (PUT /api/system, system hostname set HOST- NAME)	1259
16.2.9.2	Get_System (GET /api/system, system hostname show)	1260
16.2.9.3	Exec_FactoryReset (POST /api/system/factory-reset, firmware recovery restore)	1260
16.2.9.4	Exec_Reboot (POST /api/system/reboot, system reboot)	1260
16.2.9.5	Set_Time (PUT /api/system/time, system time set now TIME)	1260
16.2.9.6	Get_Time (GET /api/system/time, system time show)	1261
16.2.9.7	Set_Timezone (system time set timezone TIMEZONE)	1262
16.2.9.8	Get_Timezones (GET /api/system/time/zones, system timezones list)	1262
16.2.9.9	Set_RS232OutputFormat (set output-format FORMAT)	1262
16.2.9.10	Set_RS232Echo (set echo STATE)	1262
16.3	Commands for MFA-7/14/21/28	1263
16.3.1	General commands	1263
16.3.1.1	General	1263
16.3.1.1.1	Get_Help (HELP)	1263
16.3.1.1.2	Get_Info (GETINFO)	1264
16.3.1.1.3	Get_ChannelCount (GETCHANNELCNT)	1264
16.3.1.1.4	Get_Status (STATUS)	1265
16.3.1.1.5	Get_OutputInfo (GETOUTINFO)	1265
16.3.1.1.6	Reset_Boot (RESET)	1266
16.3.1.1.7	Reset_Counter (RESETCNT)	1266

16.3.1.1.8	Get_AllParameters (PRINT)	1266
16.3.1.2	Interfaces	1270
16.3.1.2.1	Set_Baudrate (BAUDRATE)	1270
16.3.1.2.2	Get_Baudrate (BAUDRATE)	1271
16.3.1.3	Parameter management	1271
16.3.1.3.1	Save_InterfaceParameters (BASICSETTINGS STORE)	1271
16.3.1.3.2	Load_InterfaceParameters (BASICSETTINGS READ)	1271
16.3.1.3.3	Save_MeasureParameters (MEASSETTINGS STORE)	1271
16.3.1.3.4	Load_MeasureParameters (MEASSETTINGS READ)	1271
16.3.1.3.5	Set_Default (SETDEFAULT)	1272
16.3.2	Measurement	1272
16.3.2.1	General	1272
16.3.2.1.1	Set_Samplerate (DATARATE)	1272
16.3.2.1.2	Get_Samplerate (DATARATE)	1272
16.3.2.1.3	Get_Measure (GETMEASURE)	1272
16.3.2.2	Color processing	1273
16.3.2.2.1	Set_ColorSpace (COLORSPACE)	1273
16.3.2.2.2	Get_ColorSpace (COLORSPACE)	1274
16.3.2.2.3	AutoGain (AUTOGAIN)	1274
16.3.2.2.4	Set_IntegrationTime (INTEGRATIONTIME)	1274
16.3.2.2.5	Get_IntegrationTime (INTEGRATIONTIME)	1275
16.3.2.2.6	Set_Gain (GAIN)	1275
16.3.2.2.7	Get_Gain (GAIN)	1275
16.3.2.2.8	WhiteCorr (WHITECORR)	1276
16.3.2.2.9	Set_WhiteCorrFactor (WHITECORR_FACTOR)	1276
16.3.2.2.10	Get_WhiteCorrFactor (WHITECORR_FACTOR)	1276
16.3.2.2.11	DarkCorr (DARKCORR)	1277
16.3.2.2.12	Set_DarkCorrOffset (DARKCORR_OFFSET)	1277
16.3.2.2.13	Get_DarkCorrOffset (DARKCORR_OFFSET)	1277
16.3.2.2.14	Set_WhiteLED (WHITELED)	1278
16.3.2.2.15	Get_WhiteLED (WHITELED)	1278
16.3.2.3	Measurement value processing	1278
16.3.2.3.1	Set_Averaging (AVERAGING)	1278
16.3.2.3.2	Get_Averaging (AVERAGING)	1279
16.3.3	Data output	1279
16.3.3.1	General	1279
16.3.3.1.1	Set_DataOutInterface (OUTPUT)	1279
16.3.3.1.2	Get_DataOutInterface (OUTPUT)	1279
16.3.3.2	Selected measurement values	1280
16.3.3.2.1	Set_Output (OUT)	1280
16.3.3.2.2	Get_Output (OUT)	1280
16.3.4	Internal commands	1281
16.3.4.1	Get_FirmwareVersion	1281
16.3.4.2	Prepare_UpdateFirmware	1281
16.3.4.3	Start_UpdateFirmware	1282
16.3.4.4	Get_UpdateFirmwareProgress	1282
16.3.4.5	Generate_Firmware	1282

17 Commands for Interfaces	1283
17.1 Commands for sensors on MEBus	1283
17.1.1 User Level	1283
17.1.1.1 Logout	1283
17.1.1.2 Login	1283
17.1.1.3 Get_UserLevel	1283
17.1.1.4 Set_Password	1284
17.1.2 Measurement	1284
17.1.2.1 Set_Samplerate	1284
17.1.2.2 Get_Samplerate	1284
17.1.2.3 Set_Trigger	1285
17.1.2.4 Get_Trigger	1285
17.1.2.5 Set_Averaging	1285
17.1.2.6 Get_Averaging	1286
17.1.2.7 Get_Measure	1286
17.1.2.8 Get_AlternateMeasure	1287
17.1.3 Data output	1287
17.1.3.1 Set_Range	1287
17.1.4 Interfaces	1288
17.1.4.1 Test_Baudrate	1288
17.1.4.2 Set_Baudrate	1288
17.1.4.3 Get_Baudrate	1289
17.1.4.4 Set_SensorAddress	1289
17.1.5 Information	1289
17.1.5.1 Get_SensorInfo	1289
17.1.5.2 Get_ChannellInfo	1290
17.1.5.3 Get_ChannellInfos	1291
17.1.5.4 Get_ControllerInfo	1293
17.1.5.5 Get_DiagnosticInfo	1293
17.1.5.6 Get_ConfigDescription	1293
17.1.5.7 Set_ConfigParameter	1294
17.1.5.8 Get_ConfigParameter	1294
17.1.5.9 Read_AllBlocks	1294
17.2 Commands for ETH_IF1032	1295
17.2.1 User Level	1295
17.2.1.1 Logout (LGO)	1295
17.2.1.2 Login (LGI)	1295
17.2.1.3 Set_Password (PWD)	1295
17.2.2 Measurement	1296
17.2.2.1 Set_SampleTime (STI)	1296
17.2.2.2 Get_SampleTime (STI?)	1296
17.2.2.3 Set_Trigger (TRG)	1296
17.2.2.4 Get_Trigger (TRG?)	1297
17.2.2.5 Set_AvrType (AVT)	1297
17.2.2.6 Get_AvrType (AVT?)	1297
17.2.2.7 Set_AvrNbr (AVN)	1298
17.2.2.8 Get_AvrNbr (AVN?)	1298
17.2.3 Data output	1298
17.2.3.1 ChannelStatus (CHS)	1298
17.2.3.2 Get_RawDataRange (MDF)	1298
17.2.3.3 Get_RawDataRanges	1299
17.2.4 Interfaces	1299
17.2.4.1 Set_DataPort (SDP)	1299

17.2.4.2	Get_DataPort (GDP)	1300
17.2.4.3	Set_IPConfiguration (IPS)	1300
17.2.4.4	Get_IPConfiguration (IPS?)	1300
17.2.4.5	Set_EthernetMode (IFC)	1301
17.2.4.6	Get_EthernetMode (IFC?)	1301
17.2.4.7	Set_AppLanguage (LNG)	1301
17.2.4.8	Get_AppLanguage (LNG?)	1302
17.2.5	Sensor interface	1302
17.2.5.1	Set_SensorInterface (SIF)	1302
17.2.5.2	Get_SensorInterface (SIF?)	1302
17.2.5.3	Set_SensorBaudrate (SBR)	1302
17.2.5.4	Get_SensorBaudrate (SBR?)	1303
17.2.5.5	Set_SensorAddress (SAD)	1303
17.2.5.6	Get_SensorAddress (SAD?)	1303
17.2.5.7	Set_ConfigParameter (PID)	1304
17.2.5.8	Get_ConfigParameter (PID?)	1304
17.2.5.9	Set_AnalogRange (ARA)	1304
17.2.5.10	Get_AnalogRange (ARA?)	1305
17.2.5.11	Set_AnalogOffset (AOF)	1305
17.2.5.12	Get_AnalogOffset (AOF?)	1305
17.2.5.13	Set_AnalogUnit (AUN)	1306
17.2.5.14	Get_AnalogUnit (AUN?)	1306
17.2.5.15	Set_AnalogMathFunction (AMF)	1307
17.2.5.16	Get_AnalogMathFunction (AMF?)	1308
17.2.6	Information	1308
17.2.6.1	Get_Status (STS)	1308
17.2.6.2	Get_Version (VER)	1309
17.2.6.3	Get_ChannelInfo (CHI)	1309
17.2.6.4	Get_ChannelInfos	1310
17.2.6.5	Get_ControllerInfo (COI)	1311
17.2.7	Internal commands	1312
17.2.7.1	Update_Firmware	1312
17.3	Commands for USBAdapter_IF2004	1312
17.3.1	Special commands	1312
17.3.1.1	Set_SyncMaster	1312
17.3.1.2	Set_Algorithm	1313
17.3.1.3	Get_Algorithm	1313
17.3.1.4	Set_SelectedSensors	1313
17.3.1.5	Get_SelectedSensors	1314
17.3.1.6	Set_SensorChannelBaudrate	1314
17.3.1.7	Get_SensorChannelBaudrate	1314
17.3.2	Parameter management	1315
17.3.2.1	Save_Parameters	1315
17.3.2.2	Load_Parameters	1315
17.3.3	Information	1315
17.3.3.1	Get_FPGAVersion	1315
17.3.3.2	Get_HWRevision	1315
17.3.3.3	Get_Option	1316
17.3.4	Timer	1316
17.3.4.1	Set_TimerFrequency	1316
17.3.4.2	Get_TimerFrequency	1317
17.3.5	RS422 (Sensor)	1317
17.3.5.1	Inputs	1317

17.3.5.1.1	Use_Gate	1317
17.3.5.1.2	Get_RxDValue	1318
17.3.5.2	Outputs	1318
17.3.5.2.1	Set_TxDSource	1318
17.3.5.2.2	Get_TxDSource	1318
17.3.5.3	Digital Inputs and Outputs	1319
17.3.5.3.1	Set_TrgSource	1319
17.3.5.3.2	Get_TrgSource	1319
17.3.6	RS422 (Computer)	1319
17.3.6.1	Set_DataMode	1319
17.3.6.2	Get_DataMode	1320
17.3.6.3	Set_RS422OutSource	1320
17.3.6.4	Get_RS422OutSource	1321
17.3.6.5	Set_RS422Baudrate	1321
17.3.6.6	Get_RS422Baudrate	1321
17.3.7	Switching inputs	1322
17.3.7.1	Set_ExtTriggerInSource	1322
17.3.7.2	Get_ExtTriggerInSource	1322
17.3.7.3	Set_ExtTriggerInDirection	1322
17.3.7.4	Get_ExtTriggerInDirection	1323
17.3.7.5	Get_ExtTriggerInValue	1323
17.3.8	Switching outputs	1323
17.3.8.1	Set_ExtTriggerOutSource	1323
17.3.8.2	Get_ExtTriggerOutSource	1324
17.3.9	LED	1324
17.3.9.1	Set_LEDMode	1324
17.3.9.2	Get_LEDMode	1325
17.4	Commands for PCI Card IF2004	1326
17.4.1	Special commands	1326
17.4.1.1	Use_Defaults	1326
17.4.1.2	Get_DrvSetting	1327
17.4.1.3	Set_SyncMaster	1328
17.4.2	General	1328
17.4.2.1	Get_FPGAVersion	1328
17.4.2.2	IF2004_SystemReset	1328
17.4.2.3	IF2004_SensorReset12	1328
17.4.2.4	IF2004_SensorReset34	1329
17.4.3	Encoder	1329
17.4.3.1	Actions	1329
17.4.3.1.1	Enc_ClearCounter	1329
17.4.3.1.2	Enc_SetLoadReg	1329
17.4.3.1.3	Enc_LoadCounter	1329
17.4.3.1.4	Enc_LatchCounter	1329
17.4.3.1.5	Enc_GetLatchReg	1329
17.4.3.1.6	Enc_UnlockTraceC	1330
17.4.3.1.7	Enc_CheckRef	1330
17.4.3.1.8	Enc_IsFirstRef	1330
17.4.3.1.9	Enc_IsSecondRef	1330
17.4.4	RS422	1330
17.4.4.1	Inputs	1330
17.4.4.1.1	IF2004_CheckGate	1330
17.5	Commands for PCI Card IF2008	1331
17.5.1	Special commands	1331

17.5.1.1	Use_Defaults	1331
17.5.1.2	Get_DrvSetting	1332
17.5.1.3	Set_SyncMaster	1333
17.5.1.4	Set_PowerSwitch	1334
17.5.2	Information	1334
17.5.2.1	Get_FPGAVersion	1334
17.5.2.2	Get_HWRevision	1334
17.5.2.3	Is_Channel56Available	1334
17.5.2.4	Is_ADCAvailable	1335
17.5.2.5	Is_DigitalIOAvailable	1335
17.5.2.6	Get_PowerError	1335
17.5.3	Timer	1335
17.5.3.1	Set_TimerFrequency	1335
17.5.4	Encoder	1336
17.5.4.1	Settings	1336
17.5.4.1.1	Set_EncoderInterpolation	1336
17.5.4.1.2	Get_EncoderInterpolation	1337
17.5.4.1.3	Set_EncoderDirection	1338
17.5.4.1.4	Get_EncoderDirection	1338
17.5.4.1.5	Set_EncoderMode	1338
17.5.4.1.6	Get_EncoderMode	1339
17.5.4.1.7	Set_EncoderLatchSource	1339
17.5.4.1.8	Get_EncoderLatchSource	1340
17.5.4.2	Actions	1341
17.5.4.2.1	Clear_Encoder	1341
17.5.4.2.2	Set_EncoderPreload	1341
17.5.4.2.3	Load_Encoder	1341
17.5.4.2.4	Latch_Encoder	1342
17.5.4.2.5	Get_EncoderValue	1342
17.5.4.2.6	EnableRef_Encoder	1342
17.5.4.2.7	Get_EncoderReference	1343
17.5.5	RS422	1343
17.5.5.1	Inputs	1343
17.5.5.1.1	Use_Gate	1343
17.5.5.1.2	Get_RxDValue	1344
17.5.5.2	Outputs	1344
17.5.5.2.1	Set_TxDSource	1344
17.5.5.2.2	Get_TxDSource	1344
17.5.5.2.3	Set_TxValue	1345
17.5.5.2.4	Get_TxValue	1345
17.5.5.3	Trigger	1345
17.5.5.3.1	Set_TrigSource	1345
17.5.5.3.2	Get_TrigSource	1346
17.5.5.3.3	Set_TrigValue	1346
17.5.5.3.4	Get_TrigValue	1346
17.5.6	Switching inputs	1347
17.5.6.1	Set_DigitalInLatchSource	1347
17.5.6.2	Get_DigitalInLatchSource	1347
17.5.6.3	Get_DigitalInValue	1347
17.5.7	Switching outputs	1348
17.5.7.1	Set_DigitalOutSource	1348
17.5.7.2	Get_DigitalOutSource	1348
17.5.7.3	Set_DigitalOutValue	1348

17.5.7.4	Get_DigitalOutValue	1349
17.5.8	Analog inputs	1349
17.5.8.1	Set_ADCLatchSource	1349
17.5.8.2	Get_ADCLatchSource	1350
17.5.8.3	Get_ADCValue	1350
17.6	Commands for EthAdapter_IF2008	1351
17.6.1	General commands	1351
17.6.1.1	Internal commands	1351
17.6.1.1.1	Use_Defaults	1351
17.6.1.1.2	Get_DrvSetting	1352
17.6.1.1.3	Set_SyncMaster	1352
17.6.1.2	Information	1353
17.6.1.2.1	Get_Info (GETINFO)	1353
17.6.1.2.2	Get_AllParameters (PRINT)	1353
17.6.1.3	Interfaces	1359
17.6.1.3.1	Set_IPConfiguration (IPCONFIG)	1359
17.6.1.3.2	Get_IPConfiguration (IPCONFIG)	1360
17.6.1.3.3	Set_IPDataTransferMode (MEATRANSFER)	1360
17.6.1.3.4	Get_IPDataTransferMode (MEATRANSFER)	1361
17.6.1.3.5	Set_FramesPerPacket_ETH (MEASCNT_ETH)	1361
17.6.1.3.6	Get_FramesPerPacket_ETH (MEASCNT_ETH)	1361
17.6.1.3.7	Set_AppLanguage (LANGUAGE)	1362
17.6.1.3.8	Get_AppLanguage (LANGUAGE)	1362
17.6.1.3.9	Set_ChannelMode<n> (CHANNELMODE<n>)	1362
17.6.1.3.10	Get_ChannelMode<n> (CHANNELMODE<n>)	1363
17.6.1.4	Timer	1363
17.6.1.4.1	Set_TimerFrequency<n> (TIMERFREQUENCY<n>)	1363
17.6.1.4.2	Get_TimerFrequency<n> (TIMERFREQUENCY<n>)	1363
17.6.1.4.3	Set_TimerRatio<n> (TIMERPULSEWIDTH<n>)	1363
17.6.1.4.4	Get_TimerRatio<n> (TIMERPULSEWIDTH<n>)	1364
17.6.1.5	Parameter management	1364
17.6.1.5.1	Save_Parameters (STORE)	1364
17.6.1.5.2	Load_Parameters (READ)	1364
17.6.1.5.3	Set_Default (SETDEFAULT)	1365
17.6.1.5.4	Reset_Boot (RESET)	1365
17.6.2	Sensor	1365
17.6.2.1	Settings	1365
17.6.2.1.1	Set_Baudrate<n> (BAUDRATE<n>)	1365
17.6.2.1.2	Get_Baudrate<n> (BAUDRATE<n>)	1366
17.6.2.1.3	Set_LaserPower<n> (LASERPOW<n>)	1366
17.6.2.1.4	Get_LaserPower<n> (LASERPOW<n>)	1366
17.6.2.1.5	Set_TriggerOutput<n> (TRIGGEROUTPUT<n>)	1367
17.6.2.1.6	Get_TriggerOutput<n> (TRIGGEROUTPUT<n>)	1367
17.6.2.2	Actions	1367
17.6.2.2.1	Get_SensorErrorInputs (SENSORERROR)	1367
17.6.2.2.2	SensorCommand<n> (TUNNEL<n>)	1368
17.6.3	Encoder	1368
17.6.3.1	Settings	1368
17.6.3.1.1	Set_EncoderInterpolation<n> (ENCINTERPOL<n>)	1368
17.6.3.1.2	Get_EncoderInterpolation<n> (ENCINTERPOL<n>)	1368
17.6.3.1.3	Set_EncoderMode<n> (ENCREF<n>)	1368
17.6.3.1.4	Get_EncoderMode<n> (ENCREF<n>)	1369
17.6.3.1.5	Set_EncoderPreload<n> (ENCVALUE<n>)	1369

17.6.3.1.6	Get_EncoderPreload<n> (ENCVALUE<n>)	1369
17.6.3.1.7	Set_EncoderDirection<n> (ENCDIR<n>)	1369
17.6.3.1.8	Get_EncoderDirection<n> (ENCDIR<n>)	1370
17.6.3.1.9	Set_EncoderLatchSource<n> (ENCLATCHSRC<n>) . .	1370
17.6.3.1.10	Get_EncoderLatchSource<n> (ENCLATCHSRC<n>) . .	1370
17.6.3.2	Actions	1371
17.6.3.2.1	Load_Encoder<n> (ENCSET)	1371
17.6.3.2.2	EnableRef_Encoder<n> (ENCRESET)	1371
17.6.3.2.3	Clear_Encoder<n> (ENCCLEAR)	1371
17.6.3.2.4	Get_EncoderValue<n> (GETENCVALUE<n>)	1371
17.6.3.2.5	Get_EncoderReference<n> (GETENCREF<n>)	1371
17.6.4	Digital I/O	1372
17.6.4.1	General	1372
17.6.4.1.1	Set_DigitalLevel (EXTLEVEL)	1372
17.6.4.1.2	Get_DigitalLevel (EXTLEVEL)	1372
17.6.4.2	Switching inputs	1372
17.6.4.2.1	Set_DigitalInLatchSource (EXTINLATCHSRC)	1372
17.6.4.2.2	Get_DigitalInLatchSource (EXTINLATCHSRC)	1373
17.6.4.2.3	Get_DigitalInValue (GETEXTINPUT)	1373
17.6.4.2.4	Set_DigitalInMode1 (EXTINPUTMODE1)	1373
17.6.4.2.5	Get_DigitalInMode1 (EXTINPUTMODE1)	1373
17.6.4.2.6	Set_DigitalInMode2 (EXTINPUTMODE2)	1374
17.6.4.2.7	Get_DigitalInMode2 (EXTINPUTMODE2)	1374
17.6.4.2.8	Set_DigitalInMode3 (EXTINPUTMODE3)	1374
17.6.4.2.9	Get_DigitalInMode3 (EXTINPUTMODE3)	1374
17.6.4.3	Switching outputs	1375
17.6.4.3.1	Set_DigitalOutSource<n> (EXTOUTSRC<n>)	1375
17.6.4.3.2	Get_DigitalOutSource<n> (EXTOUTSRC<n>)	1375
17.7	Commands for CSP2008	1375
17.7.1	Parameter management	1376
17.7.1.1	Save_Parameters (STORESETUP)	1376
17.7.1.2	Load_Parameters (LOADSETUP)	1376
17.7.2	Mastering	1376
17.7.2.1	Set_MasterValue (SWMASTER)	1376
17.7.2.2	Reset_MasterValue (SWREMASTER)	1376
17.7.3	Software trigger	1376
17.7.3.1	Set_SoftwareTriggerEnable (SWTRIGGERENABLE)	1376
17.7.3.2	Get_SoftwareTriggerEnable (SWTRIGGERENABLE)	1377
17.7.3.3	Set_SoftwareTrigger (SWTRIGGER)	1377
17.7.3.4	Get_SoftwareTrigger (SWTRIGGER)	1377
17.7.4	Special commands	1377
17.7.4.1	Use_Defaults	1377
17.7.4.2	Get_DrvSetting	1378
17.8	Commands for C-Box Analog, C-Box/2A and thicknessSENSOR .	1379
17.8.1	General commands	1380
17.8.1.1	General	1380
17.8.1.1.1	Get_Info (GETINFO)	1380
17.8.1.1.2	Set_Unit (UNIT)	1381
17.8.1.1.3	Get_Unit (UNIT)	1381
17.8.1.1.4	Set_SyncMode (SYNC)	1381
17.8.1.1.5	Get_SyncMode (SYNC)	1382
17.8.1.1.6	Reset_Boot (RESET)	1382
17.8.1.1.7	Get_ScanStatus (SCANSTATUS)	1382

17.8.1.1.8	Get_AllParameters (PRINT)	1383
17.8.1.2	Triggering	1400
17.8.1.2.1	Set_TriggerMode (TRIGGER)	1400
17.8.1.2.2	Get_TriggerMode (TRIGGER)	1400
17.8.1.2.3	Set_TriggerLevel (TRIGGERLEVEL)	1401
17.8.1.2.4	Get_TriggerLevel (TRIGGERLEVEL)	1401
17.8.1.2.5	Set_TriggerCount (TRIGGERCOUNT)	1401
17.8.1.2.6	Get_TriggerCount (TRIGGERCOUNT)	1402
17.8.1.2.7	Software_Trigger (TRIGGERSW)	1402
17.8.1.3	Interfaces	1402
17.8.1.3.1	Set_IPConfiguration (IPCONFIG)	1402
17.8.1.3.2	Get_IPConfiguration (IPCONFIG)	1403
17.8.1.3.3	Set_IPDataTransferMode (MEASTRANSFER)	1403
17.8.1.3.4	Get_IPDataTransferMode (MEASTRANSFER)	1404
17.8.1.3.5	Set_Baudrate (BAUDRATE)	1404
17.8.1.3.6	Get_Baudrate (BAUDRATE)	1405
17.8.1.3.7	Set_AppLanguage (LANGUAGE)	1405
17.8.1.3.8	Get_AppLanguage (LANGUAGE)	1405
17.8.1.4	Sensors	1406
17.8.1.4.1	Scan_Sensor<n> (SCAN<n>)	1406
17.8.1.4.2	Get_Info<n> (GETINFO<n>)	1406
17.8.1.4.3	Set_LaserPower<n> (LASERPOW<n>)	1406
17.8.1.4.4	Get_LaserPower<n> (LASERPOW<n>)	1406
17.8.1.4.5	Set_Averaging<n> (AVERAGE<n>)	1406
17.8.1.4.6	Get_Averaging<n> (AVERAGE<n>)	1407
17.8.1.4.7	SensorCommand<n> (TUNNEL<n>)	1408
17.8.1.4.8	Get_SensorInstance	1408
17.8.1.4.9	Set_MeasurePeak<n> (MEASPEAK)	1409
17.8.1.4.10	Get_MeasurePeak<n> (MEASPEAK)	1409
17.8.1.4.11	Set_TargetMode<n> (TARGETMODE)	1409
17.8.1.4.12	Get_TargetMode<n> (TARGETMODE)	1410
17.8.1.4.13	Get_Temperature<n> (GETTEMP<n>)	1410
17.8.1.5	Parameter management	1410
17.8.1.5.1	Save_Parameters (STORE)	1410
17.8.1.5.2	Load_Parameters (READ)	1411
17.8.1.5.3	Set_Default (SETDEFAULT)	1411
17.8.1.6	Internal controller commands	1412
17.8.1.6.1	Set_MultiFunctionInputMode (MFIFUNC)	1412
17.8.1.6.2	Get_MultiFunctionInputMode (MFFUNC)	1412
17.8.2	Measurement	1413
17.8.2.1	General	1413
17.8.2.1.1	Set_MeasureMode (MEASMODE)	1413
17.8.2.1.2	Get_MeasureMode (MEASMODE)	1413
17.8.2.1.3	Set_Samplerate (MEASRATE)	1413
17.8.2.1.4	Get_Samplerate (MEASRATE)	1414
17.8.2.2	Measurement value processing	1414
17.8.2.2.1	Set_Averaging (AVERAGE)	1414
17.8.2.2.2	Get_Averaging (AVERAGE)	1415
17.8.2.2.3	Set_MasterValue (MASTERMV)	1416
17.8.2.2.4	Get_MasterValue (MASTERMV)	1416
17.8.2.2.5	Set_DigitalScale<n> (DIGITALSCALE<n>)	1416
17.8.2.2.6	Get_DigitalScale<n> (DIGITALSCALE<n>)	1417
17.8.3	Data output	1417

17.8.3.1	General	1417
17.8.3.1.1	Set_DataOutInterface (OUTPUT)	1417
17.8.3.1.2	Get_DataOutInterface (OUTPUT)	1418
17.8.3.1.3	Set_Resampling (OUTREDUCE)	1418
17.8.3.1.4	Get_Resampling (OUTREDUCE)	1419
17.8.3.1.5	Set_HoldLastValid (OUTHOLD)	1420
17.8.3.1.6	Get_HoldLastValid (OUTHOLD)	1420
17.8.3.1.7	Set_FramesPerPacket_ETH (MEASCNT_ETH)	1420
17.8.3.1.8	Get_FramesPerPacket_ETH (MEASCNT_ETH)	1420
17.8.3.1.9	Set_OutputScale_RS422_USB (OUTSCALE_RS422_USB)	1421
17.8.3.1.10	Get_OutputScale_RS422_USB (OUTSCALE_RS422_USB)	1421
17.8.3.2	Selected measurement values	1422
17.8.3.2.1	Set_Output_RS422 (OUT_RS422)	1422
17.8.3.2.2	Get_Output_RS422 (OUT_RS422)	1423
17.8.3.2.3	Set_Output_USB (OUT_USB)	1424
17.8.3.2.4	Get_Output_USB (OUT_USB)	1426
17.8.3.2.5	Set_Output_ETH (OUT_ETH)	1428
17.8.3.2.6	Get_Output_ETH (OUT_ETH)	1431
17.8.3.2.7	Set_OutputAdditional (OUT_ADDITIONAL)	1433
17.8.3.2.8	Get_OutputAdditional (OUT_ADDITIONAL)	1433
17.8.3.3	Switching outputs	1434
17.8.3.3.1	Set_ErrorOutput<n> (ERRORROUT<n>)	1434
17.8.3.3.2	Get_ErrorOutput<n> (ERRORROUT<n>)	1434
17.8.3.3.3	Set_ErrorLimit<n> (ERRORLIMIT<n>)	1435
17.8.3.3.4	Get_ErrorLimit<n> (ERRORLIMIT<n>)	1435
17.8.3.4	Analog output	1436
17.8.3.4.1	Set_AnalogOutput (ANALOGOUT)	1436
17.8.3.4.2	Get_AnalogOutput (ANALOGOUT)	1436
17.8.3.4.3	Set_AnalogOutput<n> (ANALOGOUT<n>)	1437
17.8.3.4.4	Get_AnalogOutput<n> (ANALOGOUT<n>)	1437
17.8.3.4.5	Set_AnalogRange (ANALOGRANGE)	1438
17.8.3.4.6	Get_AnalogRange (ANALOGRANGE)	1438
17.8.3.4.7	Set_AnalogRange<n> (ANALOGRANGE<n>)	1439
17.8.3.4.8	Get_AnalogRange<n> (ANALOGRANGE<n>)	1439
17.8.3.4.9	Set_AnalogScale (ANALOGSCALE)	1439
17.8.3.4.10	Get_AnalogScale (ANALOGSCALE)	1440
17.8.3.4.11	Set_AnalogScale<n> (ANALOGSCALE<n>)	1441
17.8.3.4.12	Get_AnalogScale<n> (ANALOGSCALE<n>)	1441
17.8.4	Internal commands	1442
17.8.4.1	Get_FirmwareVersion	1442
17.8.4.2	Prepare_UpdateFirmware	1442
17.8.4.3	Start_UpdateFirmware	1443
17.8.4.4	Get_UpdateFirmwareProgress	1443
17.8.4.5	Generate_Firmware	1443
17.9	Commands for MultiSensor	1444
17.9.1	Special parameters at OpenSensor	1444
17.9.2	Get_DrvSetting	1445
17.9.3	Get_SensorInstance	1446
17.9.4	Get_HardwareInstance	1446
17.9.5	Set_MaximumTimeDiff	1447
17.9.6	Enable_ReferenceTimestamp	1447
17.9.7	Set_DriverSync	1447
17.9.8	Set_HardwareSync	1448

Contents

17.9.9 Set_LatestSync	1453
17.9.10 Set_ManualSync	1453
17.9.11 Sync_Start	1453
17.9.12 Any command for contained sensors and interfaces	1454
Index	1455

1 Introduction

MEDAQLib is a software library for easy data acquisition and communication with digital Micro-Epsilon sensors. The library is independent from any communication protocol or hardware interface, i.e. all sensors or controllers are accessed from your program in the same way independent whether via TCP/IP or USB or serial communication.

Currently the following sensors are supported:

ILR sensors

[optoNCDT ILR 110x/5x](#), [optoNCDT ILR 118x](#), [optoNCDT ILR 1191](#), [optoNCDT ILR 2250](#)

ILD sensors

[optoNCDT 1220](#), [optoNCDT 1302](#), [optoNCDT 1320](#), [optoNCDT 1401](#), [optoNCDT 1402](#), [optoNCDT 1420](#), [optoNCDT 17xx](#), [optoNCDT 1750/6x](#), [optoNCDT 1900](#), [optoNCDT 22xx](#), [optoNCDT 2300/10](#)

IFD sensors

[confocalDT 2401](#), [confocalDT 2421](#), [confocalDT 2422](#), [confocalDT 2431](#), [confocalDT 2445](#), [confocalDT 2451](#), [confocalDT 2461](#), [confocalDT 2471](#)

IMC sensors

[interferoMETER 5400/10](#), [interferoMETER 5600](#)

ODC sensors

[optoCONTROL 1202/20](#), [optoCONTROL 2500](#), [optoCONTROL 2520](#), [optoCONTROL 2600](#)

Eddy sensors

[eddyNCDT 306x/7x](#), [eddyNCDT 3100](#)

Capa sensors

[capaNCDT 6100](#), [capaNCDT 6120](#), [capaNCDT 62xx](#), [capaNCDT 65xx](#)

Combi sensors

[combiSENSOR 6380](#), [combiSENSOR 64xx](#)

Color sensors

[colorCONTROL ACS7000](#), [colorSENSOR CFO100/200](#), [colorSENSOR MFA-7/14/21/28](#)

Interface modules

[MEBus](#), [Eth_IF1032](#), [EthAdapter_IF2008](#), [IF2004_USB](#) adapter, [IF2004](#) and [IF2008](#) PCI(e) cards, [CSP2008](#), [C-Box](#) and [thicknessSENSOR](#)

The sensors can be accessed through interfaces [RS232](#), [USB](#) ([IF2001_USB](#) ([RS422](#)) or [RS485](#) converter and [RS232](#) high level interface), [USB](#) (via [WinUSB](#)), [IF2004](#) and [IF2008](#) PCI(e) card ([RS422](#)), [IF2004_USB](#) adapter ([RS422](#)), [IF2008_ETH](#) adapter ([RS422](#) over ethernet) and [TCP/IP](#).

The software consists of a DLL to be imported into your data acquisition project. As programming languages C/C++, Visual Basic, Delphi and many other languages are supported. For C/C++ an additional include file is provided to get you started.

1 Introduction

More than one sensor can be used over different interfaces at the same time (from one or several applications).

The functions of MEDAQLib are thread-safe, so you can call any function at the same time from concurrent threads.

Additionally Micro-Epsilon provides ICONNECT having special modules to access all the features of these sensors. ICONNECT is a programming environment giving you the possibility to develop applications without the need to "real programming".

2 Installation

MEDAQLib comes as setup executable. After installation to your destination folder the following directory structure and files will be created:

Release/MEDAQLib.dll

the 32 bit driver dll.

Release/MEDAQLib.lib

linker library for Visual C/C++ projects.

Release/MEDAQLib.NET.dll

the 32 bit assembly for using MEDAQLib in each .NET language.

Release/SensorTest.exe

simple 32 bit command line test program for any sensor (see Samples/[SensorTest](#)).

Release-x64/MEDAQLib.dll

the 64 bit version of driver dll.

Release-x64/MEDAQLib.lib

linker library 64 bit version for Visual C/C++ projects.

Release/MEDAQLib.NET.dll

the 64 bit assembly for using MEDAQLib in each .NET language.

Release-x64/SensorTest.exe

simple 64 bit command line test program for any sensor (see Samples/[SensorTest](#))

MEDAQLib.h

include file for C/C++ projects.

MEDAQLib.bas

module for easy access MEDAQLib from VB6 or VBA.

Documentation/MEDAQLib.pdf

the reference guide you are currently reading.

Documentation/Version History.txt

changes between the different released versions.

Documentation/Medaqlib-Quick_Reference.pdf

the quick start reference in english language.

Documentation/Medaqlib-Schnellstartanleitung.pdf

the quick start reference in german language.

Samples

examples showing usage of MEDAQLib.

Nothing is installed into any Windows system directory. Nothing is changed in Windows registry.

In addition the following drivers are also copied to installation folder (but not installed yet):

2 Installation

Driver/WinUSB for accessing the [SENSOR_IFD2401/SENSOR_IFD2431](#) and [CONTROLLER_CBOX](#) over [WinUSB](#).

Driver/IF2004 for accessing sensors over the [PCI_CARD_IF2004](#) PCI card.

Driver/IF2008 for accessing sensors over the [PCI_CARD_IF2008](#) PCI(e) card.

Driver/FTDI the driver for [USB_ADAPTER_IF2004](#) and the IF2001_USB (RS422).

To install a driver follow the driver wizard and manually select the directory.
These drivers are also included in the driver CD supplied with your hardware.

3 Accessing MEDAQLib in Visual C/C++

3.1 Setting up Visual Studio

For accessing MEDAQLib from Visual Studio copy the file MEDAQLib.h into your project directory, add it to your project (Project->Add Files) and include it in your C/C++ source file as follows:

```
#include "MEDAQLib.h"
```

Now you are able to compile your project accessing MEDAQLib. For linking in MEDAQLib.dll there are two possibilities:

3.1.1 Using MEDAQLib.lib, static approach

Copy the file MEDAQLib.lib into your project directory and add it to your project (Project->Add Files). Now you are able to also link your project.

3.1.2 Using MEDAQLib.dll, dynamic approach

Do not link your project against MEDAQLib.lib. For accessing the functions you have to load the MEDAQLib.dll using

```
HINSTANCE hInstance= LoadLibrary ("MEDAQLib.dll");
```

and get the functions pointers like

```
CREATESENSORINSTANCE pCreateSensorInstance= (CREATESENSORINSTANCE) GetProcAddress
(hInstance, "CreateSensorInstance");

RELEASESENSORINSTANCE pReleaseSensorInstance= (RELEASESENSORINSTANCE) GetProcAddress
(hInstance, "ReleaseSensorInstance");
```

Please check any return codes after calling these functions, which is not shown here. Calling a function then is done like

```
uint32_t result= pCreateSensorInstance (SENSOR_ILD2300);
```

4 Using MEDAQLib

In the following some programming examples in different programming languages are shown to give a first introduction into the main working structure of a MEDAQLib based program. Please note that these code examples contain no error handling, so they are not meant for production usage.

4.1 Main structure of a MEDAQLib program

As a first step the information about sensor and its interface have to be given to MEDAQLib. Also, it is a good idea for first tests to enable logfile writing by MEDAQLib. To get access to an ILD2300 via ethernet the following might be used:

```
// Tell MEDAQLib about sensor type to be used.
HANDLE hSensor = CreateSensorInstance(SENSOR_ILD2300);

// Tell MEDAQLib about interface to be used.
SetParameterString(hSensor, "IP_Interface", "TCP/IP");
SetParameterString(hSensor, "IP_RemoteAddr", "169.254.168.150");

// Enable Logfile writing
SetParameterInt (hSensor, "IP_EnableLogging", 1);

// Try to open communication to sensor via ethernet interface specified.
err= OpenSensor(hSensor);
```

To get access to an ILD1420 via USB-serial converter the following might be used:

```
// Tell MEDAQLib about sensor type to be used.
HANDLE hSensor = CreateSensorInstance(SENSOR_ILD1420);

// Tell MEDAQLib about interface to be used.
SetParameterString(hSensor, "IP_Interface", "RS232");
SetParameterString(hSensor, "IP_Port", "COM5");

// Enable Logfile writing
SetParameterInt (hSensor, "IP_EnableLogging", 1);

// Try to open communication to sensor via serial interface specified.
err= OpenSensor(hSensor);
```

After having successfully opened communication with the sensor it is possible to acquire data from sensor. The following sample shows continuous data acquisition from sensor. It will always ask for 200 values.

```
// In the following we try to always get 200 values from sensor.
//
#define EXPECTED_BLOCK_SIZE 200; // please adjust to your setting.

while(!bDone)
{
    // Check whether there's enough data to read in.
    int32_t currentlyAvailable = 0;
    err= DataAvail(hSensor, &currentlyAvailable);

    if (currentlyAvailable > EXPECTED_BLOCK_SIZE)
```

```

{
  // Allocate memory to get raw and scaled data into.
  int32_t rawData[EXPECTED_BLOCK_SIZE];
  double scaledData[EXPECTED_BLOCK_SIZE];

  // Set additional parameters for TransferData;
  int32_t expectedBlockSize = EXPECTED_BLOCK_SIZE;
  int32_t gotBlockSize = 0;

  // Fetch data from MEDAQLib's internal buffer.
  err= TransferData(hSensor, rawData, scaledData, expectedBlockSize, &gotBlockSize);

  // Now expectedBlockSize should be equal to gotBlockSize.
  // rawData contains original values from sensor, scaledData contains scaled data.
  // Do your computation on data ....
}

Sleep(10); // Sleep 10ms, allow other things to happen, ....
}

```

At end of program the communication channel and all MEDAQLib internal resources have to be returned to system.

```

// Close down by closing interface and releasing sensor instance.
CloseSensor(hSensor);
ReleaseSensorInstance(hSensor);

```

4.2 How to call a function at sensor

The following sample code shows how to call a specific function at a sensor. For the example we assume the connection to the sensor was already established by having called [CreateSensorInstance](#) and [OpenSensor](#).

The example is trying to get information about the current measurement mode of an ILD2300:

```

// Get measure mode of sensor.
int32_t iMeasureMode; /* will receive current measure mode */
SetParameterString(hSensor, "S_Command", "Get_MeasureMode");
err= SensorCommand(hSensor);
GetParameterInt(hSensor, "SA_MeasureMode", &iMeasureMode);

```

First the command has to be given to S_Command. Note that within this documentation the sensor commands are listed in the form "MEDAQLib-Command (ASCII-Command)". S_Command must be the "MEDAQLib-Command". The call to [SensorCommand](#) executes the command at the sensor itself. After having executed the command at the sensor the application is able to receive result values (sensor answers, SA_xxx) from MEDAQLib.

To set a value at sensor again first give the command to S_Command. Then the parameters for this call (sensor parameters, SP_xxx) have to be given to MEDAQLib. Only after this information is given to MEDAQLib a successful execution of the function at the sensor is possible:

```

// And set it to distance mode ( = 0, see Set_MeasureMode)
SetParameterString(hSensor, "S_Command", "Set_MeasureMode");
SetParameterInt(hSensor, "SP_MeasureMode", 0/* = Distance */);
err= SensorCommand(hSensor);

```

4 Using MEDAQLib

Another example is trying to get information from a confocal DT 2422 at which interfaces the sensor currently is sending its data:

```
// Inform MEDAQLib about command to be executed
SetParameterString(hSensor, "S_Command", "Get_DataOutInterface");
// Execute function at sensor
err = SensorCommand(hSensor);
int32_t iSendsToEthernet = 0;
int32_t iSendsToRS422 = 0;
// Ask MEDAQLib for parameters returned by sensor
GetParameterInt(hSensor, "SA_OutputEthernet", &iSendsToEthernet);
GetParameterInt(hSensor, "SA_OutputRS422", &iSendsToRS422);
```

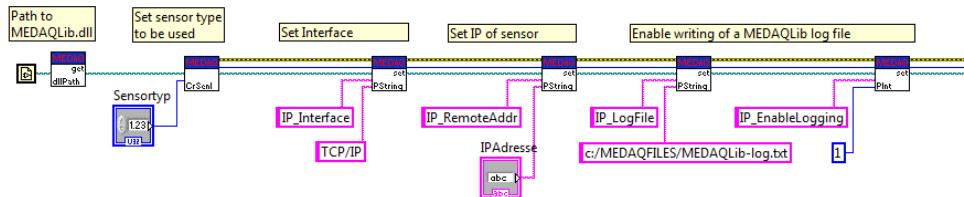
The examples above showed the principal functions to be called. For functions expecting a single argument only MEDAQLib provides short cut functions, so the above code may be written shorter like follows:

```
err= ExecSCmdGetInt(hSensor, "Get_MeasureMode", "SA_MeasureMode", &iMeasureMode);
err= SetIntExecSCmd(hSensor, "Set_MeasureMode", "SP_MeasureMode", 0/* = Distance */);
```

4.3 Main structure in a LabView Sample

The following sample shows how to implement the basic MEDAQLib application using LabView. Please note again that any error handling is omitted for clarity of structure.

The first LabView picture shows how to tell MEDAQLib which sensor type to use by calling [CreateSensorInstance](#), how to specify the communication interface and how to enable log file creation.

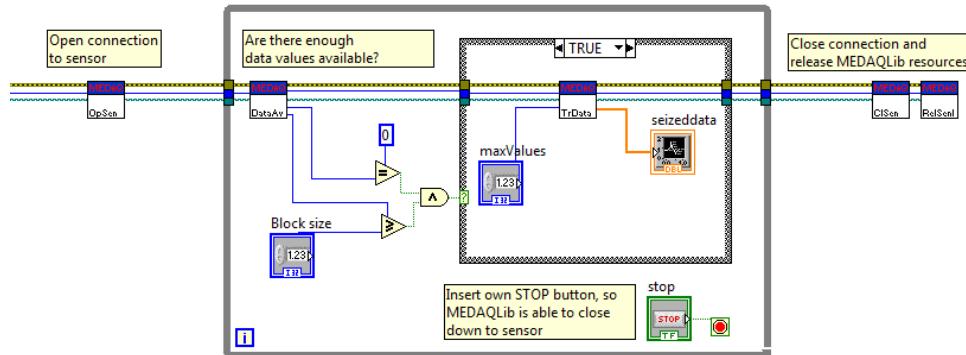


Up to this point only information is given to MEDAQLib, no communication is established to a sensor.

The second picture starts with a call to [OpenSensor](#). Now the settings from the first picture are used to try to establish a connection to the sensor using the interface specified. And only now is possible for MEDAQLib to detect any errors in the settings given.

The right part shows a [DataAvail](#) and [TransferData](#) loop to get data from sensor. At the end the interface used is closed by calling [CloseSensor](#) and the MEDAQLib internal resources are released by calling function [ReleaseSensorInstance](#).

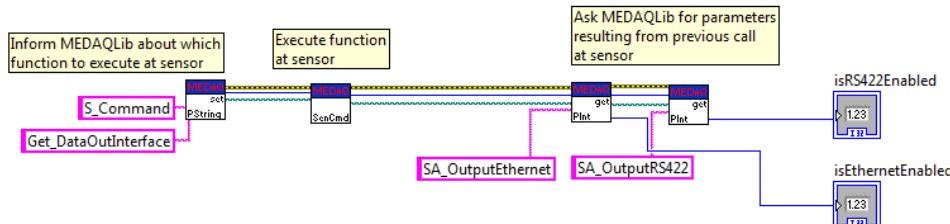
4 Using MEDAQLib



The FALSE part of the switch structure is empty and not shown. The additional STOP button is added as on pressing the LabView red stop button gives MEDAQLib no possibility to release any sensor related interface resulting then in a sensor meaning it is still connected to some application. On the next start it will refuse a new connection. Please note that also not implemented is any sleep/wait for new data as this is application dependent.

4.4 How to call a MEDAQLib function in LabView

The following sample shows how to get information about the interface an confocal DT 2421 is sending its data.

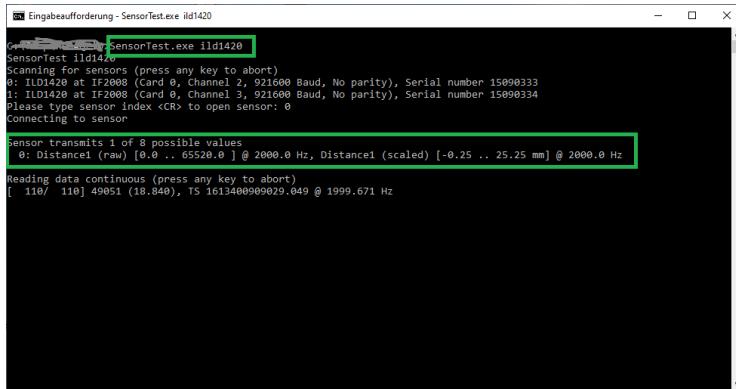


First the command `Get_DataOutInterface` has to be given to `S_Command`. The call to `SensorCommand` executes the command at the sensor itself. After having executed the command at the sensor the application is able to ask MEDAQLib for resulting sensor answers (`SA_xxx`). Here it asks for sensor answers `SA_OutputEthernet` and `SA_OutputRS422`.

4.5 How to decode signal order in TransferData or Poll returned data

Functions `TransferData` and `Poll` expect pointer to vector/array to fill in their values. To get information on how to interpret these values please use `SensorTest.exe` binary supplied within MEDAQLib distribution for your sensor as follows:

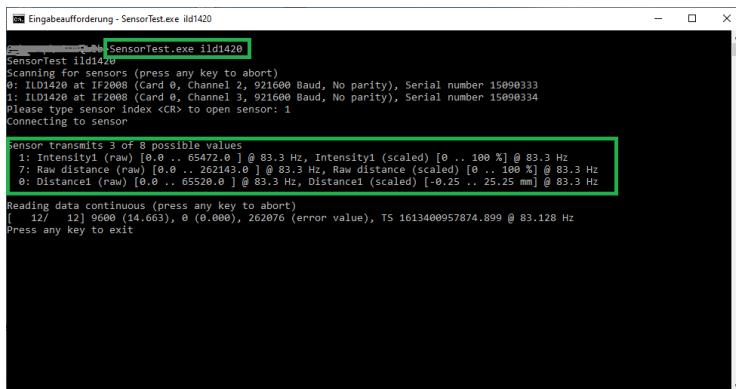
4 Using MEDAQLib



For the example shown above [TransferData](#) and [Poll](#) will fill array containing distance information like follows:



In contrast if sensor provides e.g. three signals within a data frame it will be shown by SensorTest.exe like follows:



In this example [TransferData](#) and [Poll](#) will fill the array provided by application like follows:



4.6 Synchronized data from two ILD2300 at IF2008

This example synchronizes two [SENSOR_ILD2300](#) sensors by [IF2008](#) card and reads data.

Please ensure that both sensors has equal settings.

For readability this example does not handle errors. Please add error handling to your code.

Create sensor and [IF2008](#) card instances and open them.

```

ERR_CODE err;
uint32_t sensor1= CreateSensorInstByName ("ILD2300");
uint32_t sensor2= CreateSensorInstByName ("ILD2300");
uint32_t ifCard= CreateSensorInstByName ("IF2008");
err= OpenSensorIF2008 (sensor1, 0, 0); /* First card, first channel */
err= OpenSensorIF2008 (sensor2, 0, 1); /* First card, second channel */
err= OpenSensorIF2008 (ifCard, 0, -1); /* First card, no data channel */

```

4 Using MEDAQLib

Now the sensors must be synchronized. The synchronization master is the [IF2008](#) card.

```

/* Set both sensors the same samplerate */
double samplerate= 10; // kHz
err= SetDoubleExecSCmd (sensor1, "Set_Samplerate", "SP_Measrate", samplerate);
err= SetDoubleExecSCmd (sensor2, "Set_Samplerate", "SP_Measrate", samplerate);

/* Set both sensors to synchronization slave */
SetParameterInt (sensor1, "SP_SyncMode", 1); /* Slave */
SetParameterInt (sensor1, "SP_SyncTermination", 1); /* On */
err= ExecSCmd (sensor1, "Set_SyncMode");
SetParameterInt (sensor2, "SP_SyncMode", 1); /* Slave */
SetParameterInt (sensor2, "SP_SyncTermination", 1); /* On */
err= ExecSCmd (sensor2, "Set_SyncMode");

// Set up the IF2008 card timer 1 frequency, convert from kHz to Hz
SetParameterInt (ifCard, "SP_TimerNumber", 1);
SetParameterDouble (ifCard, "SP_TimerFrequency", samplerate*1000.0);
SetParameterDouble (ifCard, "SP_TimerRatio", 0.5);
err= ExecSCmd (ifCard, "Set_TimerFrequency");

// and put it out at sync outputs
SetParameterInt (ifCard, "SP_TrgChannel1", 1); /* Timer 1 */
SetParameterInt (ifCard, "SP_TrgChannel2", 1); /* Timer 1 */
SetParameterInt (ifCard, "SP_TrgChannel3", 1); /* Timer 1 */
SetParameterInt (ifCard, "SP_TrgChannel4", 1); /* Timer 1 */
SetParameterInt (ifCard, "SP_TrgChannel5", 1); /* Timer 1 */
SetParameterInt (ifCard, "SP_TrgChannel6", 1); /* Timer 1 */
err= ExecSCmd (ifCard, "Set_TrigSource");

```

At the moment where you want to start data aquisition, clear all buffers so new data from sensors has same time base.

```

// Clear all buffers of all sensors
err= SetIntExecSCmd (sensor1, "Clear_Buffers", "SP_AllDevices", 1);

```

Then you can read synchronized data in a loop.

```

int32_t avail[2], maxValues, read[2];
int32_t raw[2][10000];
double scaled[2][10000];
while (true)
{
    err= DataAvail (sensor1, &avail[0]);
    err= DataAvail (sensor2, &avail[1]);
    maxValues= min (avail[0], avail[1]);
    if (maxValues>_countof (scaled)) /* if this happens each */
        { /* time, reduce sleep time or increase buffers, */
        maxValues= _countof (scaled); /* otherwise internal */
    } /* MEDAQLib buffer will overflow */

    err= TransferData (sensor1, &raw[0], &scaled[0], maxValues, &read[0]);
    err= TransferData (sensor2, &raw[1], &scaled[1], maxValues, &read[1]);
    assert (read[0]==read[1]); /* must be equal */

    /* Process data in raw and scaled ... */
    Sleep (100); /* Sleep until some new data has arrived */
}

```

4.7 Block based data acquisition vs Poll

MEDAQLib continuously collects all data from sensor in an own background thread or timer and stores this data into an internal ring buffer. No function call from user to MEDAQLib does interrupt reading the data, so no data is lost. For reading the data from MEDAQLib to customer application, MEDAQLib supports two operation modes:

4.7.1 Block based data acquisition

The measurement values are collected and retrieved by customer application in larger data blocks by calling [TransferData](#). The advantage of this method is that the transfer cycles do not have to be so fast and often. Because of this normally there is no loss of data.

4.7.2 Polling

Using function [Poll](#) only the very last measurement value is returned to the caller. So the response time is very short, but you may miss some data on not polling often enough.

4.7.3 Additional hints

Both modes may be used simultaneously because polling for data does not change any internal buffer of MEDAQLib.

The function [DataAvail](#) can be used to retrieve the number of available values in buffer.

To wait for enough data, the user can call [DataAvail](#) in a loop (combined with a Sleep function) or he can use the [DataAvail_Event](#).

5 Samples

For real examples please take a look at the given examples in subdirectory samples. They show you different possibilities on using MEDAQLib:

5.1 C-Sharp Example

This example on disk shows how to access an **SENSOR_ILD2300** via **TCP/IP** from C#. Before using it please change the IP address to suit your settings.

5.2 Delphi Example

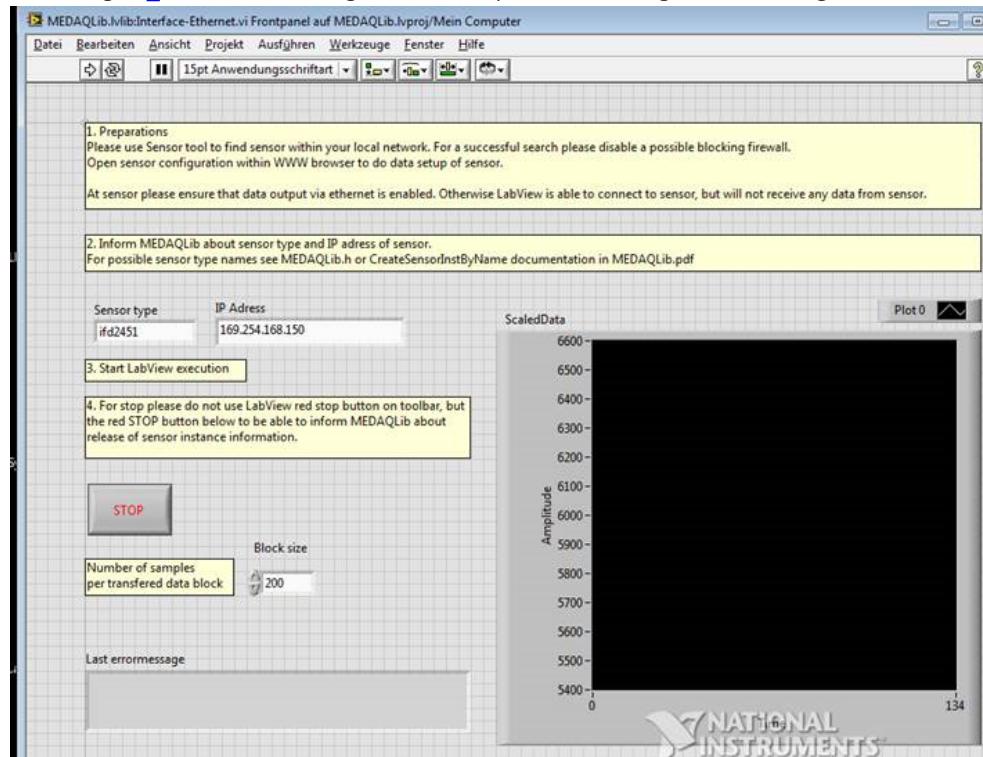
This example on disk shows how to access an **SENSOR_ILD1420** via **RS232** (RS422 to USB Converter, e.g. IF2001/USB) from Delphi. Before using it please change the COM port number to suit your settings.

5.3 DLL Example

This example on disk shows how to access an **SENSOR_ILD2300** via **IF2008_ETH** hardware card from Visual C++. In addition it shows how to access the functions from MEDAQLib without using MEDAQLib.lib, but using the dynamic approach using LoadLibrary and GetProcAddress.

5.4 LabView

Within your MEDAQLib/Samples/LabView/Public/Example directory please find three samples showing how to access a sensor via ethernet (interface-ethernet.vi), via IF2004/USB (Interface-IF2004_USB.vi) or via RS422 (Interface-RS232.vi, following **IP_Interface** naming convention) then showing the following GUI:



Please specify your sensor type as string (e.g. ILD2300 or IFC2451) at sensor type, and check right interface settings (like IP Adress or COM Port name) to suit your settings. Several VI's with base functionality are included which can be used for own applications.

5.5 Lib Example

This example on disk shows how to access an \fixlt{SENSOR_ILD1x02}{SENSOR_ILD1750} via [RS232](#) (RS422 to USB Converter, e.g. IF2001/USB) from Visual C++. Before using it please change the COM port number to suit your settings. In contrast to the DLLExample is links against the MEDAQLib.lib library directly.

5.6 IF2008 Example

This example on disk shows how to access [SENSOR_ILD2300](#) sensors via [IF2008](#) card and an encoder from Visual C++. Before using it please change the internal settings to suit your settings. It shows how to poll data and retrieve data synchronized.

5.7 Unicode Example

This example on disk shows how to access an [SENSOR_IFD2421](#) via [TCP/IP](#) from Visual C++. It uses the MEDAQLib dynamically and can be compiled for ANSI and UNICODE. To wait for data it uses event handling.

5.8 VBA Example

This example on disk shows how to access an [SENSOR_ODC2520](#) via [IF2004_USB](#) from Excel speaking Visual Basic for Applications. To see the code please right click the "Table 1" tab at the bottom of Excel. In the context menu select "Show code". The second example (VBAExample2.xls) does the same but with sensor \fixlt{SENSOR_ILD1x02}{SENSOR_ILD1750}.

5.9 VB2013Example

This example on disk shows how to access an [SENSOR_ILD1420](#) via [RS232](#) using Visual Basic 2013.

5.10 X64 Example

This example on disk shows how to access an [SENSOR_ILD2300](#) via [TCP/IP](#) from Visual C++ (64 Bit). Before using it please change the IP address to suit your settings. Additionally the usage of [Get_TransmittedDataInfo](#) is shown.

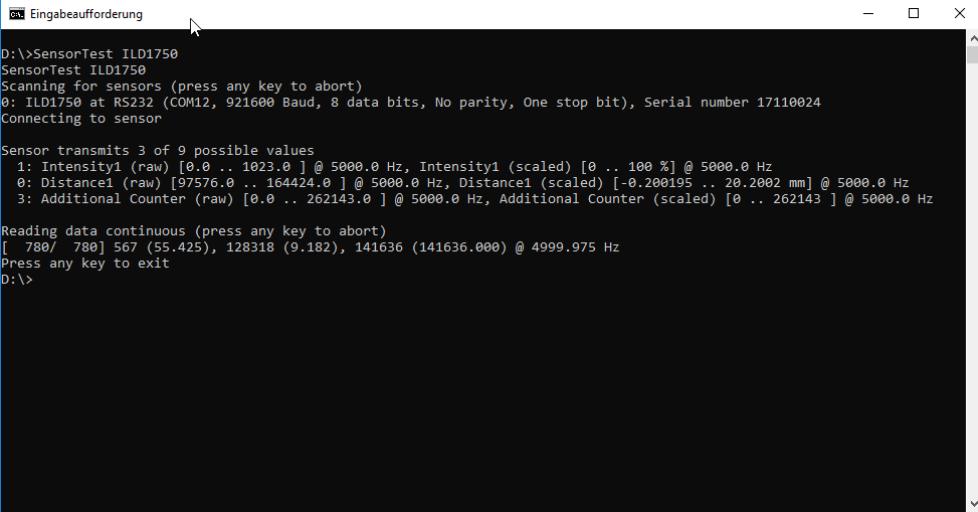
5.11 SensorFinder

This example on disk is a C command line utility which search for sensor [SENSOR_ILD2300](#) (using [SensorFinder commands](#)) and (if found over [TCP/IP](#)) change it's IP address (using [Set_IPConfig](#)).

5 Samples

5.12 SensorTest

This example on disk is a C command line utility which search/open any sensor, show information about transmitted data and continuously retrieves data.



```
D:\>SensorTest ILD1750
SensorTest ILD1750
Scanning for sensors (press any key to abort)
0: ILD1750 at RS232 (COM12, 921600 Baud, 8 data bits, No parity, One stop bit), Serial number 17110024
Connecting to sensor

Sensor transmits 3 of 9 possible values
 1: Intensity1 (raw) [0.0 .. 1023.0] @ 5000.0 Hz, Intensity1 (scaled) [0 .. 100 %] @ 5000.0 Hz
 0: Distance1 (raw) [97576.0 .. 164424.0] @ 5000.0 Hz, Distance1 (scaled) [-0.200195 .. 20.2002 mm] @ 5000.0 Hz
 3: Additional Counter (raw) [0.0 .. 262143.0] @ 5000.0 Hz, Additional Counter (scaled) [0 .. 262143] @ 5000.0 Hz

Reading data continuous (press any key to abort)
[ 780/ 780] 567 (55.425), 128318 (9.182), 141636 (141636.000) @ 4999.975 Hz
Press any key to exit
D:\>
```

The screenshot shows SensorTest working for ILD1750 connected at COM12 working with 921600 Baud and a data rate of 5kHz. In addition it supports all MEDAQLib parameters on command line, i.e. it is possible to create MEDAQLib log file by adding IP_EnableLogging=1 to end of command line.

6 Function Reference

6.1 Create a sensor instance (CreateSensorInst...)

Name: CreateSensorInstance

CreateSensorInstance

Description:

Creates an instance for the sensor which will be used in further function calls.

Attention! Please don't forget to call [ReleaseSensorInstance](#) at the end. Otherwise there will be memory and handle leaks.

Declaration:

```
uint32_t CreateSensorInstance (ME_SENSOR sensor);
```

Parameter: ME_SENSOR sensor

sensor

Direction: [IN]

Valid values:

- SENSOR_ILR110x_115x (19) - optoNCDT ILR
- SENSOR_ILR118x (20) - optoNCDT ILR
- SENSOR_ILR1191 (21) - optoNCDT ILR
- SENSOR_ILR2250 (60) - optoNCDT ILR
- SENSOR_ILD1220 (56) - optoNCDT
- SENSOR_ILD1302 (24) - optoNCDT
- SENSOR_ILD1320 (41) - optoNCDT
- SENSOR_ILD1401 (1) - optoNCDT
- SENSOR_ILD1402 (23) - optoNCDT
- SENSOR_ILD1420 (42) - optoNCDT
- SENSOR_ILD1700 (2) - optoNCDT
- SENSOR_ILD1750 (51) - optoNCDT
- SENSOR_ILD1900 (58) - optoNCDT
- SENSOR_ILD2200 (5) - optoNCDT
- SENSOR_ILD2300 (29) - optoNCDT
- SENSOR_IFD2401 (12) - confocalDT
- SENSOR_IFD2421 (46) - confocalDT
- SENSOR_IFD2422 (47) - confocalDT
- SENSOR_IFD2431 (13) - confocalDT
- SENSOR_IFD2445 (39) - confocalDT
- SENSOR_IFD2451 (30) - confocalDT
- SENSOR_IFD2461 (44) - confocalDT
- SENSOR_IFD2471 (26) - confocalDT
- SENSOR_ODC1202 (25) - optoCONTROL
- SENSOR_ODC2500 (8) - optoCONTROL
- SENSOR_ODC2520 (37) - optoCONTROL
- SENSOR_ODC2600 (9) - optoCONTROL
- SENSOR_LLT27xx - scanCONTROL and gapCONTROL (31), only for SensorFinder commands, [OpenSensor](#) will fail
- SENSOR_DT3060 (50) - eddyNCDT for whole DT306x and DT307x family
- SENSOR_DT3100 (28) - eddyNCDT
- SENSOR_IMC5400 (55) - interferoMETER
- SENSOR_IMC5600 (59) - interferoMETER

SENSOR_DT6100 (16) - capaNCDT
 SENSOR_DT6120 (40) - capaNCDT
 CONTROLLER_DT6200 (33) - capaNCDT
 CONTROLLER_KSS6380 (18) - combiSENSOR
 CONTROLLER_KSS64xx (45) - combiSENSOR
 CONTROLLER_DT6500 (15) - capaNCDT
 CONTROLLER_DT6536 (54) - capaNCDT
 SENSOR_MEBUS (43) - Generic sensor with MEbus protocol support, only for internal use
 PCI_CARD_IF2004 (10) - PCI card IF2004
 PCI_CARD_IF2008 (22) - PCI(e) card IF2008
 ETH_ADAPTER_IF2008 (52) - IF2008/ETH ethernet adapter
 CONTROLLER_CSP2008 (32) - Universal controller
 ETH_IF1032 (34) - Interface module Ethernet/EtherCAT
 USB_ADAPTER_IF2004 (36) - USB adapter IF2004/USB
 CONTROLLER_CBOX (38) - External controller C-Box
 THICKNESS_SENSOR (48) - thicknessSENSOR
 SENSOR_ACST7000 (35) - colorCONTROL
 SENSOR_CFO (53) - colorCONTROL
 SENSOR_MFA (61) - colorCONTROL

Description: Type of sensor used.

Returns: Number of the created sensor instance or zero, if parameter sensor is not valid.

Name: CreateSensorInstByName

CreateSensorInstByName

Description:

Creates an instance for the sensor which will be used in further function calls.

Attention! Please don't forget to call [ReleaseSensorInstance](#) at the end. Otherwise there will be memory and handle leaks.

Declaration:

```
uint32_t CreateSensorInstByName (const char *sensorName);
```

Parameter: const char * sensorName

sensorName

Direction: [IN]

Valid values:

- 'ILR110', 'ILR115' matches [SENSOR_ILR110x_115x](#)
- 'ILR118' matches [SENSOR_ILR118x](#)
- 'ILR1191' matches [SENSOR_ILR1191](#)
- 'ILR2250' matches [SENSOR_ILR2250](#)
- 'ILD1220', 'optoNCDT 1220' matches [SENSOR_ILD1220](#)
- 'ILD1302', 'optoNCDT 1302' matches [SENSOR_ILD1302](#)
- 'ILD1320', 'optoNCDT 1320' matches [SENSOR_ILD1320](#)
- 'ILD1401', 'optoNCDT 1401' matches [SENSOR_ILD1401](#)
- 'ILD1402', 'optoNCDT 1402' matches [SENSOR_ILD1402](#)
- 'ILD1420', 'optoNCDT 1420' matches [SENSOR_ILD1420](#)
- 'ILD1750', 'optoNCDT 1750' matches [SENSOR_ILD1750](#)
- 'ILD176', 'optoNCDT 176' matches [SENSOR_ILD1750](#)
- 'ILD17', 'optoNCDT 17' matches [SENSOR_ILD1700](#)

'ILD1900', 'optoNCDT 1900' matches [SENSOR_ILD1900](#)
 'ILD22', 'optoNCDT 22' matches [SENSOR_ILD2200](#)
 'ILD23', 'optoNCDT 23' matches [SENSOR_ILD2300](#)
 'IFC2401', 'IFD2401', 'confocalDT 2401' matches [SENSOR_IFD2401](#)
 'IFC2421', 'IFD2421', 'confocalDT 2421' matches [SENSOR_IFD2421](#)
 'IFC2422', 'IFD2422', 'confocalDT 2422' matches [SENSOR_IFD2422](#)
 'IFC2431', 'IFD2431', 'confocalDT 2431' matches [SENSOR_IFD2431](#)
 'IFC2445', 'IFD2445', 'confocalDT 2445' matches [SENSOR_IFD2445](#)
 'IFC2451', 'IFD2451', 'confocalDT 2451' matches [SENSOR_IFD2451](#)
 'IFC2461', 'IFD2461', 'confocalDT 2461' matches [SENSOR_IFD2461](#)
 'IFC2471', 'IFD2471', 'confocalDT 2471' matches [SENSOR_IFD2471](#)
 'ODC1202', 'optoCONTROL 1202', 'ODC1220', 'optoCONTROL
 1220' matches [SENSOR_ODC1202](#)
 'ODC2500', 'optoCONTROL 2500' matches [SENSOR_ODC2500](#)
 'ODC2520', 'optoCONTROL 2520' matches [SENSOR_ODC2520](#)
 'ODC2600', 'optoCONTROL 2600' matches [SENSOR_ODC2600](#)
 'IMC5400', 'interferoMETER 5400' matches [SENSOR_IMC5400](#)
 'IMC5600', 'interferoMETER 5600' matches [SENSOR_IMC5600](#)
 'LLT27', 'scanCONTROL 27', 'gapCONTROL 27' matches [SENSOR_LLTT27xx](#)
 'DT306', 'eddyNCDT 306', 'DT307', 'eddyNCDT 307' matches [SENSOR_DT306](#)
 'DT3100', 'eddyNCDT 3100' matches [SENSOR_DT3100](#)
 'DT6100', 'capaNCDT 6100' matches [SENSOR_DT6100](#)
 'DT6120', 'capaNCDT 6120' matches [SENSOR_DT6120](#)
 'DT62', 'capaNCDT 62' matches [CONTROLLER_DT6200](#)
 'KSS6380', 'combiSENSOR 6380' matches [CONTROLLER_KSS6380](#)
 'KSS64', 'combiSENSOR 64' matches [CONTROLLER_KSS64xx](#)
 'DT6536', 'capaNCDT 6536' matches [CONTROLLER_DT6536](#)
 'DT65', 'capaNCDT 65' matches [CONTROLLER_DT6500](#)
 'MEbus', 'ME-Bus' matches [SENSOR_MEBUS](#)
 'IF2008 Eth', 'EthAdapter_IF2008' matches [ETH_ADAPTER_IF2008](#)
 'IF2008' matches [PCI_CARD_IF2008](#)
 'CSP2008' matches [CONTROLLER_CSP2008](#)
 'IF1032' matches [ETH_IF1032](#)
 'IF2004 USB' matches [USB_ADAPTER_IF2004](#)
 'IF2004' matches [PCI_CARD_IF2004](#)
 'C-Box' matches [CONTROLLER_CBOX](#)
 'thicknessSENSOR' matches [THICKNESS_SENSOR](#)
 'ACS7000', 'colorCONTROL 7000' matches [SENSOR_ACS7000](#)
 'CFO100', 'CFO200', 'colorSENSOR CFO' matches [SENSOR_CFO](#)
 'MFA' matches [SENSOR_MFA](#)

Description: Any of the values above are tokenized (delimited at spaces) and searched (case insensitive) in parameter sensorName.

This will guarantee a high match rate. Only if all of the tokens are contained, the sensorName is valid.

Returns: Number of the created sensor instance or zero, if parameter sensorName cannot be interpreted.

Name: CreateSensorInstByNameU

CreateSensorInstByNameU

Description:

Creates an instance for the sensor which will be used in further function calls (Unicode version).

Attention! Please don't forget to call [ReleaseSensorInstance](#) at the end. Otherwise there will be memory and handle leaks.

Declaration:

```
uint32_t CreateSensorInstByNameU (const wchar_t * sensorName);
```

Parameter: const wchar_t * sensorName

sensorName

Direction: [IN]

Valid values:

- 'ILR110', 'ILR115' matches [SENSOR_ILR110x_115x](#)
- 'ILR118' matches [SENSOR_ILR118x](#)
- 'ILR1191' matches [SENSOR_ILR1191](#)
- 'ILR2250' matches [SENSOR_ILR2250](#)
- 'ILD1220', 'optoNCDT 1220' matches [SENSOR_ILD1220](#)
- 'ILD1302', 'optoNCDT 1302' matches [SENSOR_ILD1302](#)
- 'ILD1320', 'optoNCDT 1320' matches [SENSOR_ILD1320](#)
- 'ILD1401', 'optoNCDT 1401' matches [SENSOR_ILD1401](#)
- 'ILD1402', 'optoNCDT 1402' matches [SENSOR_ILD1402](#)
- 'ILD1420', 'optoNCDT 1420' matches [SENSOR_ILD1420](#)
- 'ILD1750', 'optoNCDT 1750' matches [SENSOR_ILD1750](#)
- 'ILD176', 'optoNCDT 176' matches [SENSOR_ILD1750](#)
- 'ILD17', 'optoNCDT 17' matches [SENSOR_ILD1700](#)
- 'ILD1900', 'optoNCDT 1900' matches [SENSOR_ILD1900](#)
- 'ILD22', 'optoNCDT 22' matches [SENSOR_ILD2200](#)
- 'ILD23', 'optoNCDT 23' matches [SENSOR_ILD2300](#)
- 'IFC2401', 'IFD2401', 'confocalDT 2401' matches [SENSOR_IFD2401](#)
- 'IFC2421', 'IFD2421', 'confocalDT 2421' matches [SENSOR_IFD2421](#)
- 'IFC2422', 'IFD2422', 'confocalDT 2422' matches [SENSOR_IFD2422](#)
- 'IFC2431', 'IFD2431', 'confocalDT 2431' matches [SENSOR_IFD2431](#)
- 'IFC2445', 'IFD2445', 'confocalDT 2445' matches [SENSOR_IFD2445](#)
- 'IFC2451', 'IFD2451', 'confocalDT 2451' matches [SENSOR_IFD2451](#)
- 'IFC2461', 'IFD2461', 'confocalDT 2461' matches [SENSOR_IFD2461](#)
- 'IFC2471', 'IFD2471', 'confocalDT 2471' matches [SENSOR_IFD2471](#)
- 'ODC1202', 'optoCONTROL 1202', 'ODC1220', 'optoCONTROL 1220' matches [SENSOR_ODC1202](#)
- 'ODC2500', 'optoCONTROL 2500' matches [SENSOR_ODC2500](#)
- 'ODC2520', 'optoCONTROL 2520' matches [SENSOR_ODC2520](#)
- 'ODC2600', 'optoCONTROL 2600' matches [SENSOR_ODC2600](#)
- 'IMC5400', 'interferoMETER 5400' matches [SENSOR_IMC5400](#)
- 'IMC5600', 'interferoMETER 5600' matches [SENSOR_IMC5600](#)
- 'LLT27', 'scanCONTROL 27', 'gapCONTROL 27' matches [SENSOR_LLT27xx](#)
- 'DT306', 'eddyNCDT 306', 'DT307', 'eddyNCDT 307' matches [SENSOR_DT306](#)
- 'DT3100', 'eddyNCDT 3100' matches [SENSOR_DT3100](#)
- 'DT6100', 'capaNCDT 6100' matches [SENSOR_DT6100](#)
- 'DT6120', 'capaNCDT 6120' matches [SENSOR_DT6120](#)
- 'DT62', 'capaNCDT 62' matches [CONTROLLER_DT6200](#)
- 'KSS6380', 'combiSENSOR 6380' matches [CONTROLLER_KSS6380](#)
- 'KSS64', 'combiSENSOR 64' matches [CONTROLLER_KSS64xx](#)

'DT6536', 'capaNCDT 6536' matches [CONTROLLER_DT6536](#)
 'DT65', 'capaNCDT 65' matches [CONTROLLER_DT6500](#)
 'MEbus', 'ME-Bus' matches [SENSOR_MEBUS](#)
 'IF2008 Eth', 'EthAdapter_IF2008' matches [ETH_ADAPTER_IF2008](#)
 'IF2008' matches [PCI_CARD_IF2008](#)
 'CSP2008' matches [CONTROLLER_CSP2008](#)
 'IF1032' matches [ETH_IF1032](#)
 'IF2004 USB' matches [USB_ADAPTER_IF2004](#)
 'IF2004' matches [PCI_CARD_IF2004](#)
 'C-Box' matches [CONTROLLER_CBOX](#)
 'thicknessSENSOR' matches [THICKNESS_SENSOR](#)
 'ACS7000', 'colorCONTROL 7000' matches [SENSOR_ACS7000](#)
 'CFO100', 'CFO200', 'colorSENSOR CFO' matches [SENSOR_CFO](#)
 'MFA' matches [SENSOR_MFA](#)

Description: Any of the values above are tokenized (delimited at spaces) and searched (case insensitive) in parameter `sensorName`. This will guarantee a high match rate. Only if all of the tokens are contained, the `sensorName` is valid.

Returns: Number of the created sensor instance or zero, if parameter `sensorName` cannot be interpreted.

6.2 Releasing sensor instance ([ReleaseSensorInstance](#))

Name: `ReleaseSensorInstance`

[ReleaseSensorInstance](#)

Description:

Free the specified sensor instance.

Declaration:

```
ERR_CODE ReleaseSensorInstance (uint32_t instanceHandle);
```

Parameter: `uint32_t instanceHandle`

`instanceHandle`

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Returns:

`ERR_NOERROR` (0) on success.

`ERR_INSTANCE_NOT_EXIST` (-24) if `instanceHandle` is not valid.

6.3 Set parameters ([SetParameter...](#))

Before connecting to the sensor ([OpenSensor](#)) or sending commands to the sensor ([SensorCommand](#)) parameters for both functions can be specified. Each parameter can be set by one of following functions.

Name: `SetParameterInt`

[SetParameterInt](#)

Description:

Set a 4 Byte integer parameter.

Declaration:

```
ERR_CODE SetParameterInt (uint32_t instanceHandle, const char *paramName,
                         int32_t paramValue);
```

Parameter: uint32_t instanceHandle instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * paramName paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: int32_t paramValue paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Name: SetParameterIntU

SetParameterIntU
Description:

Set a 4 Byte integer parameter (Unicode version).

Declaration:

```
ERR_CODE SetParameterIntU (uint32_t instanceHandle, const wchar_t *paramName,
                           int32_t paramValue);
```

Parameter: uint32_t instanceHandle instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * paramName paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: int32_t paramValue paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Name: SetParameterDWORD_PTR

**SetParameterDWORD_-
PTR**
Description:

Set a 4 Byte (Win32) or 8 Byte (Win64) pointer parameter.

Declaration:

```
ERR_CODE SetParameterDWORD_PTR (uint32_t instanceHandle, const char
                               *paramName, ptr_t paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by
[CreateSensorInstance](#).

Parameter: const char * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#)
and [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: ptr_t (void *) paramValue

paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Name: SetParameterDWORD_PTRU

**SetParameterDWORD_-
PTRU**
Description:

Set a 4 Byte (Win32) or 8 Byte (Win64) pointer parameter (Unicode ver-
sion).

Declaration:

```
ERR_CODE SetParameterDWORD_PTRU (uint32_t instanceHandle, const wchar_t
                                 *paramName, ptr_t paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by
[CreateSensorInstance](#).

Parameter: const wchar_t * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#)
and [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: ptr_t (void *) paramValue

paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Name: SetParameterDouble

SetParameterDouble
Description:

Set a 8 Byte double parameter.

Declaration:

```
ERR_CODE SetParameterDouble (uint32_t instanceHandle, const char *paramName,
                           double paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: double paramValue

paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Name: SetParameterDoubleU

SetParameterDoubleU
Description:

Set a 8 Byte double parameter (Unicode version).

Declaration:

```
ERR_CODE SetParameterDoubleU (uint32_t instanceHandle, const wchar_t *paramName,
                            double paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: double paramValue

paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Name: SetParameterString

SetParameterString
Description:

Set a string (pointer to a character array) parameter.

Declaration:

```
ERR_CODE SetParameterString (uint32_t instanceHandle, const char *paramName,
                            const char *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: const char * paramValue

paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Name: SetParameterStringU

SetParameterStringU
Description:

Set a string (pointer to a character array) parameter (Unicode version).

Declaration:

```
ERR_CODE SetParameterStringU (uint32_t instanceHandle, const wchar_t *
                             *paramName, const wchar_t *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: const wchar_t * paramValue

paramValue

Direction: [IN]

Description: Value (as unicode) of the parameter.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Name: SetParameterBinary

SetParameterBinary
Description:

Set binary data (pointer to a character array) to MEDAQLib.

Declaration:

```
ERR_CODE SetParameterBinary (uint32_t instanceHandle, const char *paramName,
                           const uint8_t *paramValue, uint32_t len);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as string. Useful for the parameter SP_CmdStr (in [Cmd_Generic](#))

Parameter: const uint8_t * paramValue

paramValue

Direction: [IN]

Description: Value of the parameter.

Parameter: uint32_t len

len

Direction: [IN]

Description: Length of the binary data in bytes.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Name: SetParameterBinaryU

SetParameterBinaryU
Description:

Set binary data (pointer to a character array) to MEDAQLib (Unicode version).

Declaration:

```
ERR_CODE SetParameterBinary (uint32_t instanceHandle, const wchar_t *paramName,
                           const uint8_t *paramValue, uint32_t len);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as unicode string. Useful for the parameter SP_CmdStr (in [Cmd_Generic](#))

Parameter: const uint8_t * paramValue paramValue

Direction: [IN]

Description: Value of the parameter.

Parameter: uint32_t len len

Direction: [IN]

Description: Length of the binary data in bytes.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Name: SetParameters [SetParameters](#)

Description:

Set a list of parameters at once.

Declaration:

```
ERR_CODE SetParameters (uint32_t instanceHandle, const char *parameterList);
```

Parameter: uint32_t instanceHandle instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * parameterList parameterList

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

The parameters must be separated by white spaces (space, tabulator, carriage return or line feed).

Parameter name and value must be separated by equality sign (=).

Attention! No space is allowed here.

Attention! No space is allowed here.

If parameter value only contains a sign and digits, it is interpreted as integer. If there is a decimal point and/or exponent additionally it is interpreted as double. Otherwise it is interpreted as string parameter.

If the string value should contain spaces, the whole string must be quoted by "..." or '...'.

Description: List of the parameters as string.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_WRONG_PARAMETER](#) (-18) if the parameter list has syntax errors.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Name: SetParametersU

SetParametersU
Description:

Set a list of parameters at once (Unicode version).

Declaration:

```
ERR_CODE SetParametersU (uint32_t instanceHandle, const wchar_t *parameterList);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by
[CreateSensorInstance](#).

Parameter: const wchar_t * parameterList

parameterList

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#)
 and [Sensor commands](#) and following.

The parameters must be separated by white spaces (space, tabulator,
 carriage return or line feed).

Parameter name and value must be separated by equality sign (=).

Attention! No space is allowed here.

If parameter value only contains a sign and digits, it is interpreted as
 integer. If there is a decimal point and/or exponent additionally it is
 interpreted as double. Otherwise it is interpreted as string parameter.

If the string value should contain spaces, the whole string must be
 quoted by "..." or '...'.

Description: List of the parameters as unicode string.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_WRONG_PARAMETER](#) (-18) if the parameter list has syntax errors.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

6.4 Get parameters (GetParameter...)

After sending a command to a sensor ([SensorCommand](#)) the answer can be retrieved at same manner as setting parameters.

Name: GetParameterInt

GetParameterInt
Description:

Get a 4 Byte integer parameter.

Declaration:

```
ERR_CODE GetParameterInt (uint32_t instanceHandle, const char *paramName,  

 int32_t *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by
[CreateSensorInstance](#).

Parameter: const char * paramName paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: int32_t * paramValue paramValue

Direction: [OUT]

Description: Pointer to a variable retrieving the parameter

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_WRONG_PARAMETER](#) (-18) if the pointer to paramValue is nullptr or the datatype does not match.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

[ERR_NOT_FOUND](#) (-25) if the parameter (specified by paramName) is not found.

Name: GetParameterIntU

GetParameterIntU

Description:

Get a 4 Byte integer parameter (Unicode version).

Declaration:

```
ERR_CODE GetParameterIntU (uint32_t instanceHandle, const wchar_t *paramName,
                           int32_t *paramValue);
```

Parameter: uint32_t instanceHandle instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * paramName paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: int32_t * paramValue paramValue

Direction: [OUT]

Description: Pointer to a variable retrieving the parameter

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_WRONG_PARAMETER](#) (-18) if the pointer to paramValue is nullptr or the datatype does not match.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

[ERR_NOT_FOUND](#) (-25) if the parameter (specified by paramName) is not found.

Name: GetParameterDWORD_PTR

**GetParameterDWORD_-
PTR**
Description:

Get a 4 Byte (Win32) or 8 Byte (Win64) pointer parameter.

Declaration:

```
ERR_CODE GetParameterDWORD_PTR (uint32_t instanceHandle, const char
                                *paramName, ptr_t *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: ptr_t * (void **) paramValue

paramValue

Direction: [OUT]

Description: Pointer to a variable retrieving the parameter

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_WRONG_PARAMETER](#) (-18) if the pointer to paramValue is nullptr or the datatype does not match.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

[ERR_NOT_FOUND](#) (-25) if the parameter (specified by paramName) is not found.

Name: GetParameterDWORD_PTRU

**GetParameterDWORD_-
PTRU**
Description:

Get a 4 Byte (Win32) or 8 Byte (Win64) pointer parameter (Unicode version).

Declaration:

```
ERR_CODE GetParameterDWORD_PTRU (uint32_t instanceHandle, const wchar_t
                                 *paramName, ptr_t *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: `ptr_t * (void **) paramValue` paramValue

Direction: [OUT]

Description: Pointer to a variable retrieving the parameter

Returns:

`ERR_NOERROR` (0) on success.

`ERR_WRONG_PARAMETER` (-18) if the pointer to `paramValue` is `nullptr` or the datatype does not match.

`ERR_INSTANCE_NOT_EXIST` (-24) if `instanceHandle` is not valid.

`ERR_NOT_FOUND` (-25) if the parameter (specified by `paramName`) is not found.

Name: `GetParameterDouble`

GetParameterDouble

Description:

Get a 8 Byte double parameter.

Declaration:

```
ERR_CODE GetParameterDouble (uint32_t instanceHandle, const char *paramName,
                            double *paramValue);
```

Parameter: `uint32_t instanceHandle`

`instanceHandle`

Direction: [IN]

Description: Number of the sensor instance, previously returned by `CreateSensorInstance`.

Parameter: `const char * paramName`

`paramName`

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: `double * paramValue`

`paramValue`

Direction: [OUT]

Description: Pointer to a variable retrieving the parameter

Returns:

`ERR_NOERROR` (0) on success.

`ERR_WRONG_PARAMETER` (-18) if the pointer to `paramValue` is `nullptr` or the datatype does not match.

`ERR_INSTANCE_NOT_EXIST` (-24) if `instanceHandle` is not valid.

`ERR_NOT_FOUND` (-25) if the parameter (specified by `paramName`) is not found.

Name: GetParameterDoubleU

GetParameterDoubleU
Description:

Get a 8 Byte double parameter (Unicode version).

Declaration:

```
ERR_CODE GetParameterDoubleU (uint32_t instanceHandle, const wchar_t *paramName, double *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: double * paramValue

paramValue

Direction: [OUT]

Description: Pointer to a variable retrieving the parameter

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_WRONG_PARAMETER](#) (-18) if the pointer to paramValue is nullptr or the datatype does not match.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

[ERR_NOT_FOUND](#) (-25) if the parameter (specified by paramName) is not found.

Name: GetParameterString

GetParameterString
Description:

Get a string (character array) parameter.

Declaration:

```
ERR_CODE GetParameterString (uint32_t instanceHandle, const char *paramName, char *paramValue, uint32_t *maxLen);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: char * paramValue paramValue

Direction: [OUT]

Description: Pointer to a variable (character buffer) retrieving the parameter.

Parameter: uint32_t * maxLen maxLen

Direction: [IN/OUT]

Description: The buffer must be allocated by the application. The size of the buffer is specified at maxLen. If the resulting string is larger than maxLen, it is truncated and **ERR_NOMEMORY** (-19) is returned. The real length of the string (maybe truncated) is returned in maxLen too. If paramValue is nullptr, the length of the containing string is returned in maxLen.

Returns:

ERR_NOERROR (0) on success.

ERR_WRONG_PARAMETER (-18) if the pointer to maxLen is nullptr or the datatype does not match.

ERR_NOMEMORY (-19) if the buffer is too short to hold the complete answer.

ERR_INSTANCE_NOT_EXIST (-24) if instanceHandle is not valid.

ERR_NOT_FOUND (-25) if the parameter (specified by paramName) is not found.

Name: GetParameterStringU

GetParameterStringU

Description:

Get a string (character array) parameter (Unicode version).

Declaration:

```
ERR_CODE GetParameterStringU (uint32_t instanceHandle, const wchar_t *paramName, wchar_t *paramValue, uint32_t *maxLen);
```

Parameter: uint32_t instanceHandle instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * paramName paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: wchar_t * paramValue paramValue

Direction: [OUT]

Description: Pointer to a variable (unicode buffer) retrieving the parameter.

Parameter: uint32_t * maxLen maxLen

Direction: [IN/OUT]

Description: The buffer must be allocated by the application. The size of the buffer (in characters, not bytes) is specified at maxLen. If the resulting string is larger than maxLen, it is truncated and **ERR_NOMEMORY** (-19) is returned. The real length of the string (maybe truncated) is returned in maxLen too.

If paramValue is nullptr, the length of the containing string is returned in maxLen.

Returns:

ERR_NOERROR (0) on success.
ERR_WRONG_PARAMETER (-18) if the pointer to maxLen is nullptr or the datatype does not match.
ERR_NOMEMORY (-19) if the buffer is too short to hold the complete answer.
ERR_INSTANCE_NOT_EXIST (-24) if instanceHandle is not valid.
ERR_NOT_FOUND (-25) if the parameter (specified by paramName) is not found.

Name: GetParameterBinary

GetParameterBinary
Description:

Get binary data from MEDAQLib.

Declaration:

```
ERR_CODE GetParameterBinary (uint32_t instanceHandle, const char *paramName,
                            uint8_t *paramValue, uint32_t *maxLen);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: uint8_t * paramValue

paramValue

Direction: [OUT]

Description: Pointer to a variable (character buffer) retrieving the parameter.

Parameter: uint32_t * maxLen

maxLen

Direction: [IN/OUT]

Description: The buffer must be allocated by the application. The size of the buffer (in bytes) is specified at maxLen. If the resulting binary data is larger than maxLen, it is truncated and **ERR_NOMEMORY** (-19) is returned. The real length of the binary data (maybe truncated) is returned in maxLen too.

If paramValue is nullptr, the length of the containing binary data is returned in maxLen.

Returns:

ERR_NOERROR (0) on success.
ERR_WRONG_PARAMETER (-18) if the pointer to maxLen is nullptr or the datatype does not match.
ERR_NOMEMORY (-19) if the buffer is too short to hold the complete answer.
ERR_INSTANCE_NOT_EXIST (-24) if instanceHandle is not valid.
ERR_NOT_FOUND (-25) if the parameter (specified by paramName) is not found.

Name: GetParameterBinaryU

GetParameterBinaryU
Description:

Get binary data from MEDAQLib (Unicode version).

Declaration:

```
ERR_CODE GetParameterBinaryU (uint32_t instanceHandle, const wchar_t *paramName, uint8_t *paramValue, uint32_t *maxLen);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Interface parameters](#) and [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: uint8_t * paramValue

paramValue

Direction: [OUT]

Description: Pointer to a variable (character buffer) retrieving the parameter.

Parameter: uint32_t * maxLen

maxLen

Direction: [IN/OUT]

Description: The buffer must be allocated by the application. The size of the buffer (in bytes) is specified at maxLen. If the resulting binary data is larger than maxLen, it is truncated and [ERR_NOMEMORY](#) (-19) is returned. The real length of the binary data (maybe truncated) is returned in maxLen too.

If paramValue is nullptr, the length of the containing binary data is returned in maxLen.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_WRONG_PARAMETER](#) (-18) if the pointer to maxLen is nullptr or the datatype does not match.

[ERR_NOMEMORY](#) (-19) if the buffer is too short to hold the complete answer.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

[ERR_NOT_FOUND](#) (-25) if the parameter (specified by paramName) is not found.

Name: GetParameters

GetParameters
Description:

Get all available parameters at once.

Declaration:

```
ERR_CODE GetParameters (uint32_t instanceHandle, char *parameterList, uint32_t *maxLen);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: `char * parameterList` parameterList

Direction: [OUT]

Valid value: The parameters are separated by a space. Parameter name and value is separated by equality sign (=). Integer parameters only contains a sign and digits. Double parameters contains a decimal point and/or exponent additionally. String parameters are always quoted by "...".

Description: List of the parameters as string.

Parameter: `uint32_t * maxLen` maxLen

Direction: [IN/OUT]

Description: The buffer must be allocated by the application. The size of the buffer is specified at maxLen. If the resulting string is larger than maxLen, it is truncated and `ERR_NOMEMORY` (-19) is returned. The real length of the string (maybe truncated) is returned in maxLen too. If parameterList is nullptr, the length of the containing string is returned in maxLen.

Returns:

`ERR_NOERROR` (0) on success.

`ERR_WRONG_PARAMETER` (-18) if the pointer to maxLen is nullptr.

`ERR_NOMEMORY` (-19) if the buffer is too short to hold the complete answer.

`ERR_INSTANCE_NOT_EXIST` (-24) if instanceHandle is not valid.

Name: `GetParametersU`

GetParametersU

Description:

Get all available parameters at once (Unicode version).

Declaration:

```
ERR_CODE GetParametersU (uint32_t instanceHandle, wchar_t *parameterList,
                        uint32_t *maxLen);
```

Parameter: `uint32_t instanceHandle` instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by `CreateSensorInstance`.

Parameter: `wchar_t * parameterList` parameterList

Direction: [OUT]

Valid value: The parameters are separated by a space. Parameter name and value is separated by equality sign (=). Integer parameters only contains a sign and digits. Double parameters contains a decimal point and/or exponent additionally. String parameters are always quoted by "...".

Description: List of the parameters as unicode string.

Parameter: `uint32_t * maxLen` maxLen

Direction: [IN/OUT]

Description: The buffer must be allocated by the application. The size of the buffer (in characters, not bytes) is specified at maxLen. If the resulting string is larger than maxLen, it is truncated and `ERR_NOMEMORY` (-19) is returned. The real length of the string (maybe truncated) is returned in maxLen too.

If parameterList is nullptr, the length of the containing string is returned in maxLen.

Returns:

ERR_NOERROR (0) on success.
 ERR_WRONG_PARAMETER (-18) if the pointer to maxLen is nullptr.
 ERR_NOMEMORY (-19) if the buffer is too short to hold the complete answer.
 ERR_INSTANCE_NOT_EXIST (-24) if instanceHandle is not valid.

6.5 Clear internal parameter buffer (ClearAllParameters)

Name: ClearAllParameters

ClearAllParameters
Description:

Before building up a new command using the [Set parameters](#) functions, the internal parameter buffer should be cleared. This avoids parameter mismatch between several commands.

Declaration:

```
ERR_CODE ClearAllParameters (uint32_t instanceHandle);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Returns:

ERR_NOERROR (0) on success.
 ERR_INSTANCE_NOT_EXIST (-24) if instanceHandle is not valid.

6.6 Connecting to the sensor (OpenSensor)

Name: OpenSensor

OpenSensor
Description:

Establish the connection to the sensor.

After connecting to the sensor, all internal parameters starting with SP_, CP_ and IP_ are cleared (like at function [ClearAllParameters](#)).

Attention! Please don't forget call [CloseSensor](#) at the end. Otherwise it can come to unpredictable behaviour when resources leave open.

Declaration:

```
ERR_CODE OpenSensor (uint32_t instanceHandle);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Returns:

ERR_NOERROR (0) on success.
 ERR_CANNOT_OPEN (-2) if interface cannot be opened.
 ERR_APPLYING_PARAMS (-4) if parameters cannot applied to driver.
 ERR_INTERFACE_NOT_SUPPORTED (-10) if the specified interface (parameter "IP_Interface") is not supported.
 ERR_ALREADY_OPEN (-11) if the connection is already open.
 ERR_CANNOT_CREATE_INTERFACE (-12) if the interface cannot be created.
 ERR_WRONG_PARAMETER (-18) if a parameter is not valid (e.g. out of range).
 ERR_NOMEMORY (-19) if there is not enough memory to allocate MEDAQLib ring buffer.
 ERR_INSTANCE_NOT_EXIST (-24) if instanceHandle is not valid.
 ERR_WARNING (-26) if data interface could not be opened.
 If first bit of IP_AutomaticMode is set (1), any error returned by [SensorCommand](#).
 Any other asynchronous error happened at interal threads or callbacks or timers.
 For full reference of all possible error codes please consult MEDAQLib.h file.

6.7 Closing connection to sensor ([CloseSensor](#))

Name: CloseSensor

CloseSensor
Description:

Close the connection to the sensor.

Declaration:

```
ERR_CODE CloseSensor (uint32_t instanceHandle);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Returns:

ERR_NOERROR (0) on success.

ERR_HW_COMMUNICATION (-7) if TCP/IP connection could not be closed.

ERR_INSTANCE_NOT_EXIST (-24) if instanceHandle is not valid.

6.8 Sending commands to the sensor ([SensorCommand](#))

Name: SensorCommand

SensorCommand
Description:

Before sending a command to the sensor the command name itself has to be provided in string parameter [S_Command](#) and all sensor parameters (named SP_...) for this function must be specified with corresponding [Set parameters](#) functions. After the command has been executed the sensor answer

parameters (named SA_...) can be read using [Get parameters](#) functions.

At start of this function, all internal parameters starting with SA_, and IA_ are cleared (like at function [ClearAllParameters](#)).

At end of this function, all internal parameters starting with SP_, CP_, and IP_ are cleared (like at function [ClearAllParameters](#)).

At section [How to call a function at sensor](#) some examples can be found.

Please note the also available short cut functions like [ExecSCmd](#) or [SetIntExecSCmd](#).

Declaration:

```
ERR_CODE SensorCommand (uint32_t instanceHandle);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_FUNCTION_NOT_SUPPORTED](#) (-1) if a function was called which is not supported by sensor or hardware interface.

[ERR_CANNOT_OPEN](#) (-2) for [SensorFinder commands](#).

[ERR_NOT_OPEN](#) (-3) if connection is not established

[ERR_APPLYING_PARAMS](#) (-4) if interface parameters changed by command (e.g. change baudrate) and cannot be applied.

[ERR_CLEARING_BUFFER](#) (-6) for command [Clear_Buffers](#) or if IP_ClearSendBuffer or IP_ClearReceiveBuffer is specified the driver could not be cleared or pending operations on USB could not be canceled.

[ERR_HW_COMMUNICATION](#) (-7) if communication with sensor failed.

[ERR_TIMEOUT_READING_FROM_SENSOR](#) (-8) if no answer is received within answer time.

[ERR_READING_SENSOR_DATA](#) (-9) if reading sensor answer over USB failed.

[ERR_INTERFACE_NOT_SUPPORTED](#) (-10) for internal command Open_DataSocket.

[ERR_CANNOT_CREATE_INTERFACE](#) (-12) for [SensorFinder commands](#).

[ERR_NO_SENSORDATA_AVAILABLE](#) (-13) for sensor command Get_Spectrum of SENSOR_IFD2401 and SENSOR_IFD2431.

[ERR_UNKNOWN_SENSOR_COMMAND](#) (-14) if parameter [S_Command](#) is not valid.

[ERR_UNKNOWN_SENSOR_ANSWER](#) (-15) if sensor answer is not an answer to a command.

[ERR_SENSOR_ANSWER_ERROR](#) (-16) if sensor answer cannot be interpreted or sensor returned an error number.

[ERR_SENSOR_ANSWER_TOO_SHORT](#) (-17) if sensor answer is too short.

[ERR_WRONG_PARAMETER](#) (-18) if a parameter is not valid (e.g. out of range).

[ERR_NOMEMORY](#) (-19) if there is not enough memory to allocate a buffer.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

[ERR_WARNING](#) (-26) for [SensorFinder commands](#).

[ERR_SENSOR_ANSWER_WARNING](#) (-27) if sensor returned a warning.

Any other asynchronous error happened at interal threads or callbacks or timers.

For full reference of all possible error codes please consult MEDAQLib.h file.

6.9 Polling data from sensor (Poll)

Name: Poll

Poll

Description:

Get the latest values from sensor.

Declaration:

```
ERR_CODE Poll (uint32_t instanceHandle, int32_t *rawData, double *scaledData,
               int32_t maxValues);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: int32_t * rawData

rawData

Direction: [OUT]

Description: Pointer to value (or array of values) to retrieve latest data frame from sensor as raw values. For signal data order within array please see [How to decode signal order](#).

If the pointer is null, no data is transferred for this parameter.

The meaning of the raw values is described at each sensor section.

Parameter: double * scaledData

scaledData

Direction: [OUT]

Description: Pointer to value (or array of values) to retrieve latest data frame from sensor scaled by sensor range. For signal data order within array please see [How to decode signal order](#).

If the pointer is null, no data is transferred for this parameter.

The meaning of the scaled values is described at each sensor section.

Please see parameter [IP_ScaleErrorValues](#) for scaling error values.

Parameter: int32_t maxValues

maxValues

Direction: [IN]

Description: Length of rawData and scaledData. Some sensors can measure more than one value per measure cycle (e.g. IFD's which measures distance, intensity, ... or ODC's which can measure multiple segments). In this case maxValues can be set up to frame size (values per measure cycle).

If maxValues is smaller than frame size only the first maxValues values are transferred.

If maxValues is greater than frame size, values of more than one frame are transferred.

Returns:

ERR_NOERROR (0) on success.

ERR_NOT_OPEN (-3) if connection is not established

ERR_NO_SENSORDATA_AVAILABLE (-13) if either the receive buffer does not contain any value from sensor. This error can occur if you poll for data immediately after connecting or after a sensor command. In this case, wait some time until data arrives at MEDAQLib. Or this error can occur, if MEDAQLib doesn't interpret data from sensor, because some information is missing. In this case, call the function as described at error text of **GetError**.

ERR_INSTANCE_NOT_EXIST (-24) if instanceHandle is not valid.

If the internal ring buffer overflows (because Poll does not remove data), no warning is output.

Any other asynchronous error happened at internal threads or callbacks or timers.

For full reference of all possible error codes please consult MEDAQLib.h file.

6.10 Number of values available to read (DataAvail)

Name: DataAvail

DataAvail

Description:

The values available returned may be divided by frame size to calculate the number of frames available. A frame is the set of values transmitted by the sensor for each one measurement.

Declaration:

```
ERR_CODE DataAvail (uint32_t instanceHandle, int32_t *avail);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by **CreateSensorInstance**.

Parameter: int32_t * avail

avail

Direction: [OUT]

Description: Pointer to value to retrieve number of values available from sensor

Returns:

ERR_NOERROR (0) on success.

ERR_NOT_OPEN (-3) if connection is not established.

ERR_TIMEOUT_READING_FROM_SENSOR (-8) if no data frame received for at least one second or three data cycles.

ERR_NO_SENSORDATA_AVAILABLE (-13) This error can occur, if MEDAQLib doesn't interpret data from sensor, because some information is missing. In this case, call the function as described at error text of **GetError**.

ERR_WRONG_PARAMETER (-18) if parameter avail is nullptr.

ERR_INSTANCE_NOT_EXIST (-24) if instanceHandle is not valid.

Any other asynchronous error happened at internal threads or callbacks or timers.

For full reference of all possible error codes please consult MEDAQLib.h file.

6.11 Block wise data acquisition from sensor (TransferData...)

Name: TransferData

TransferData

Description:

Transfer the data from driver to application.

Declaration:

```
ERR_CODE TransferData (uint32_t instanceHandle, int32_t *rawData, double
                      *scaledData, int32_t maxValues, int32_t *read);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: int32_t * rawData

rawData

Direction: [OUT]

Description: Pointer to array of values to retrieve data from sensor as raw values. For signal data order within array please see [How to decode signal order](#).

If the pointer is null, no data is transferred for this parameter. If rawData and scaledData are null all buffers used are emptied.

The meaning of the raw values is described at each sensor section.

Parameter: double * scaledData

scaledData

Direction: [OUT]

Description: Pointer to array of values to retrieve data frame from sensor scaled by sensor range. For signal data order within array please see [How to decode signal order](#).

If the pointer is null, no data is transferred for this parameter. If rawData and scaledData are null all buffers used are emptied.

The meaning of the scaled values is described at each sensor section.

Please see parameter [IP_ScaleErrorValues](#) for scaling error values.

Parameter: int32_t maxValues

maxValues

Direction: [IN]

Description: Length of rawData and scaledData. It should be a multiple of frame size, otherwise the rest of the last frame is lost, because TransferData always starts with frame start.

Parameter: int32_t * read

read

Direction: [OUT]

Description: Will receive the real number of data values transferred. This value returned will be less than the expected maxValues if not enough data is available.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_NOT_OPEN](#) (-3) if connection is not established.

[ERR_NO_SENSORDATA_AVAILABLE](#) (-13) This error can occur, if MEDAQLib doesn't can interpret data from sensor, because some information is missing. In this case, call the function as described at error text of [GetError](#).

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Any other asynchronous error happened at interal threads or callbacks or timers.

For full reference of all possible error codes please consult MEDAQLib.h file.

Name: TransferDataTS

TransferDataTS
Description:

Same as [TransferData](#) but with an additional parameter to retrieve timestamp of data.

Declaration:

```
ERR_CODE TransferDataTS (uint32_t instanceHandle, int32_t *rawData,
    double *scaledData, int32_t maxValues, int32_t *read, double *timestamp);
```

Parameter: uint32_t instanceHandle instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: int32_t * rawData rawData

Direction: [OUT]

Description: Pointer to array of values to retrieve data from sensor as raw values. For signal data order within array please see [How to decode signal order](#).

Parameter: double * scaledData scaledData

Direction: [OUT]

Description: Pointer to array of values to retrieve data frame from sensor scaled by sensor range. For signal data order within array please see [How to decode signal order](#).

If rawData is null or scaledData is null, no data is transferred for this parameter. If both parameters are null all buffers used are emptied.

The meaning of the raw and scaled values is described at each sensor section. Please see parameter [IP_ScaleErrorValues](#) for scaling error values.

Parameter: int32_t maxValues maxValues

Direction: [IN]

Description: Length of rawData and scaledData. It should be a multiple of frame size, otherwise the rest of the last frame is lost, because TransferDataTS always starts with frame start.

Parameter: int32_t * read read

Direction: [OUT]

Description: Will receive the real number of data values transferred. This value returned will be less than the expected maxValues if not enough data is available.

Parameter: double * timestamp timestamp

Direction: [OUT]

Description: Will receive the timestamp of the first (oldest) value in data array. It is in milli seconds starting at 01.01.1970 01:00.

It can be used to synchronize data from different sensors.

Each time when the internal ring buffer is cleared (e.g. at sensor command), it is resetted and when next data block arrives it is calculated by actual time minus acquire duration of data block (at sensor).

At TransferData(TS) it is automatically increased (by adding duration of block, calculated from expected datarate). But if data is discarded (because of the ring buffer overflow or synchronization lost), it is not incremented and therefore no longer correct. In this case, call sensor command [Clear_Buffers](#).

Returns:

ERR_NOERROR (0) on success.
ERR_NOT_OPEN (-3) if connection is not established.
ERR_NO_SENSORDATA_AVAILABLE (-13) This error can occur, if MEDAQLib doesn't can interpret data from sensor, because some information is missing. In this case, call the function as described at error text of **GetError**.
ERR_INSTANCE_NOT_EXIST (-24) if instanceHandle is not valid.
 Any other asynchronous error happened at interal threads or callbacks or timers.
 For full reference of all possible error codes please consult MEDAQLib.h file.

6.12 Get additional error information (**GetError...**)

Name: GetError

GetError
Description:

Get the extended error text of last error in MEDAQLib.
 This function can be used after an error return from any MEDAQLib function.

Declaration:

```
ERR_CODE GetError (uint32_t instanceHandle, char *errText, uint32_t maxLen);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by **CreateSensorInstance**.

Parameter: char * errText

errText

Direction: [OUT]

Description: String buffer to get extended error string.

Parameter: uint32_t maxLen

maxLen

Direction: [IN]

Description: Length of string buffer. If the error text is longer as maxLen it is truncated. The string is null terminated.

Returns:

ERR_NOERROR (0) if no error occurred previously.
ERR_WRONG_PARAMETER (-18) if the pointer to errText is nullptr and maxLen is not 0.
ERR_INSTANCE_NOT_EXIST (-24) if instanceHandle is not valid.
 Any other error value reported by previous function or happened asynchronous at interal threads or callbacks or timers.

Name: GetErrorU

GetErrorU
Description:

Get the last error in driver (Unicode version). Rest of functionality is identical to [GetError](#).

Declaration:

```
ERR_CODE GetErrorU (uint32_t instanceHandle, wchar_t *errText, uint32_t maxLen);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: wchar_t * errText

errText

Direction: [OUT]

Description: Unicode string buffer to get extended error string.

Parameter: uint32_t maxLen

maxLen

Direction: [IN]

Description: Length of string buffer (in characters, not bytes). If the error text is longer as maxLen it is truncated. The string is null terminated.

Returns:

[ERR_NOERROR](#) (0) on success.

[ERR_WRONG_PARAMETER](#) (-18) if the pointer to errText is nullptr and maxLen is not 0.

[ERR_INSTANCE_NOT_EXIST](#) (-24) if instanceHandle is not valid.

Any other error value reported by previous function or happened asynchronous at interal threads or callbacks or timers.

Following errors can occur in MEDAQLib:

ERR_NOERROR (0)

Function was successful, no error occurred.

ERR_FUNCTION_NOT_SUPPORTED (-1)

A function was called which is not supported by sensor or hardware interface.

ERR_CANNOT_OPEN (-2)

The hardware interface could not be opened.

ERR_NOT_OPEN (-3)

The connection to sensor is not established.

ERR_APPLYING_PARAMS (-4)

Internal parameters could not be applied.

ERR_SEND_CMD_TO_SENSOR (-5)

Internal error message, never returned at any MEDAQLib function.

ERR_CLEARING_BUFFER (-6)

Hardware or driver buffers failed to clear.

ERR_HW_COMMUNICATION (-7)

Communication with sensor failed or hardware interface detected asynchronous errors or a hardware interface was closed unexpected.

ERR_TIMEOUT_READING_FROM_SENSOR (-8)

A timeout occurred while reading from sensor.

ERR_READING_SENSOR_DATA (-9)

Reading sensor answer from sensor failed or asynchronous read operation failed or [DataAvail_Event](#) could not be set or transmitted data from sensor changed unexpected.

ERR_INTERFACE_NOT_SUPPORTED (-10)

The specified hardware interface does not exist or is not available for this sensor.

ERR_ALREADY_OPEN (-11)

The connection to sensor is already open.

ERR_CANNOT_CREATE_INTERFACE (-12)

The specified hardware interface cannot be created.

ERR_NO_SENSORDATA_AVAILABLE (-13)

Sensor data should be read but is not available.

ERR_UNKNOWN_SENSOR_COMMAND (-14)

The specified command is not available for the sensor.

ERR_UNKNOWN_SENSOR_ANSWER (-15)

The answer received from sensor is not known by MEDAQLib.

ERR_SENSOR_ANSWER_ERROR (-16)

The sensor returned an error message.

ERR_SENSOR_ANSWER_TOO_SHORT (-17)

The answer received from sensor is not complete.

ERR_WRONG_PARAMETER (-18)

An input parameter is not valid or the datatype does not match.

ERR_NOMEMORY (-19)

The answer does not fit into the available buffer or there is too less system memory to allocate a buffer.

ERR_NO_ANSWER RECEIVED (-20)

Internal error message, never returned at any MEDAQLib function.

ERR_SENSOR_ANSWER_DOES_NOT_MATCH_COMMAND (-21)

Internal error message, never returned at any MEDAQLib function.

ERR_BAUDRATE_TOO_LOW (-22)

Baudrate is too low for this command.

ERR_OVERFLOW (-23)

An asynchronous overflow occurred in any hardware or driver buffer.

ERR_INSTANCE_NOT_EXIST (-24)

The specified instance handle is not valid.

ERR_NOT_FOUND (-25)

The parameter (specified by paramName) is not found.

ERR_WARNING (-26)

A warning occurred in MEDAQLib, e.g. Ethernet packet counter mismatch, discarding data because of changed parameters, ring buffer overflow (only at [DataAvail](#) or [TransferData](#)), failure at setting IP configuration, ...

ERR_SENSOR_ANSWER_WARNING (-27)

The sensor returned a warning.

6.13 Get version of MEDAQLib dll (GetDLLVersion...)

Name: GetDLLVersion

[GetDLLVersion](#)
Description:

Retrieves the version of the MEDAQLib dll.

This function can be called at any time, no sensor instance is needed. The version is stored in versionStr and is limited to length of maxLen (should be at least 11 bytes).

Declaration:

```
ERR_CODE GetDLLVersion (char *versionStr, uint32_t maxLen);
```

Parameter: char * versionStr

versionStr

Direction: [OUT]

Description: String buffer to get version info.

Parameter: uint32_t maxLen

maxLen

Direction: [IN]

Description: Length of string buffer. If the version info is longer as maxLen it is truncated. The string is null terminated.

Returns:

[ERR_NOERROR](#) (0) on success.

Name: GetDLLVersionU

[GetDLLVersionU](#)
Description:

Retrieves the version of the MEDAQLib dll (Unicode version). Rest of functionality is identical to [GetDLLVersion](#).

Declaration:

```
ERR_CODE GetDLLVersionU (wchar_t *versionStr, uint32_t maxLen);
```

Parameter: wchar_t * versionStr

versionStr

Direction: [OUT]

Description: Unicode string buffer to get version info.

Parameter: uint32_t maxLen

maxLen

Direction: [IN]

Description: Length of string buffer (in characters, not bytes). If the version info is longer as maxLen it is truncated. The string is null terminated.

Returns:

[ERR_NOERROR](#) (0) on success.

6.14 EnableLogging wrapper function (EnableLogging...)

Wrapper functions for a set of [Set parameters](#) functions and executes the sensor command [EnableLogging](#). This usage of this functions makes the code shorter and more readable.

Name:	EnableLogging	EnableLogging
Description:		
Set the parameters to enable logging.		
Declaration:		
	ERR_CODE EnableLogging (uint32_t instanceHandle, int32_t enableLogging, int32_t logTypes, int32_t logLevels, const char *logFile, int32_t logAppend, int32_t logFlush, int32_t logSplitSize);	
Parameter:	uint32_t instanceHandle	instanceHandle
Direction:	[IN]	
Description:	Number of the sensor instance, previously returned by CreateSensorInstance .	
Parameter:	int32_t enableLogging	enableLogging
Direction:	[IN]	
Valid values:		
	0= FALSE	
	1= TRUE	
Description:	This parameter enables or disables logging to file for debugging purposes.	
Parameter:	int32_t logTypes	logTypes
Direction:	[IN]	
Valid values:		
	A bit combination of following values:	
	1= HIGH_TYPE (User <--> MEDAQLib)	
	2= MIDDLE_TYPE (Sensor layer <--> Interface layer)	
	4= LOW_TYPE (MEDAQLib <--> Hardware driver)	
	8= ERROR_TYPE (Any errors reported by MEDAQLib)	
	16= DRIVER_TYPE (Hardware driver <--> Sys driver)	
	32= APPL_TYPE (Application specific, see LogToFile and LogToFileU)	
	2147483647= ALL (all log types are active)	
Description:	This parameter specifies the type of messages to log.	
Parameter:	int32_t logLevels	logLevels
Direction:	[IN]	
Valid values:		
	A bit combination of following values:	
	1= EMERGENCY_LEVEL (logging emerging errors)	
	2= CRITICAL_LEVEL (logging critical errors)	
	4= ERROR_LEVEL (logging errors which occurs)	
	8= WARNING_LEVEL (logging warnings from MEDAQLib)	
	16= NOTICE_LEVEL (logging notices)	
	32= TRACE_LEVEL (logging function calls)	
	64= DATA_LEVEL (logging data in binary mode)	
	2147483647= ALL (all levels are active)	
Description:	This parameter specifies the kind of event to log.	

Parameter: const char * logFile logFile

Direction: [IN]

Description: File name of log file.

If it is empty or ends with '\' or '/', an automatic generated name ('SensorLog_%yyyy-%MM-%dd_%hh-%mm-%ss.%Us.txt') is appended.

Many placeholders (...) can be used for automatic name generation. Important ones are:

%h= hour in 24 hours format (0, 1, ..., 9, 10, ..., 23)

%hh= hour in 24 hours format (00, 01, ..., 09, 10, ..., 23)

%H= hour in 12 hours format (1, 2, ..., 9, 10, 11, 12)

%HH= hour in 12 hours format (01, 02, ..., 09, 10, 11, 12)

%m= minute (0, 1, ..., 9, 10, ..., 59)

%mm= minute (00, 01, ..., 09, 10, ..., 59)

%s= second (0, 1, ..., 9, 10, ..., 59)

%ss= second (00, 01, ..., 09, 10, ..., 59)

%Ms= millisecond (000, 001, ..., 999)

%Us= microsecond (000000, 000001, ..., 999999)
%PP= output for morning, afternoon for US-American time format
(AM, PM)

%d= day (1, 2, ..., 9, 10, ..., 31)

%dd= day (01, 02, ..., 09, 10, ..., 31)

%M= month (1, 2, ..., 9, 10, 11, 12)

%MM= month (01, 02, ..., 09, 10, 11, 12)

%yy= year (70, 71, ..., 99, 00, 01, ..., 38)

%yyyy= year (1970, 1971, ..., 1999, 2000, 2001, ..., 2038)

%DoW= day of week (0= Sunday, 1= Monday, ..., 6= Saturday)

%DoY= day of year (0, 1, ..., 9, 10, ..., 364, if a leap year 365)

%WoY= week of year (0, 1, ..., 9, 10, ..., 52), starting with 0= the
week with the first Sunday in year (US format)

Parameter: int32_t logAppend logAppend

Direction: [IN]

Valid values:

0= FALSE

1= TRUE

Description: This parameter specifies if the logfile should be cleared at
opening or if the new data should be appended to file.

Parameter: int32_t logFlush logFlush

Direction: [IN]

Valid values:

0= FALSE

1= TRUE

Description: This parameter specifies if the logfile should be flushed
after each output. In this case, it is sure that all information is stored
to logfile before proceeding. But depending on the storage device it
can slow down the MEDAQLib.

Parameter: int32_t logSplitSize logSplitSize

Direction: [IN]

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Unit: KB (1024 Bytes)

Description: If this parameter is greater than 0, logfile is closed and reopened when this size is reached. If the file name contains placeholders (%...), a new name is generated before opening. Otherwise, the same file is opened again and if appending is off, the old content is overwritten.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterInt](#), [SetParameterString](#) and [SensorCommand](#).

Name: EnableLoggingU

EnableLoggingU

Description:

Set the parameters to enable logging (Unicode version).

Declaration:

```
ERR_CODE EnableLoggingU (uint32_t instanceHandle, int32_t enableLogging,
    int32_t logTypes, int32_t logLevels, const wchar_t *logFile, int32_t
    logAppend, int32_t logFlush, int32_t logSplitSize);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: int32_t enableLogging

enableLogging

Direction: [IN]

Valid values:

0= FALSE

1= TRUE

Description: This parameter enables or disables logging to file for debugging purposes.

Parameter: int32_t logTypes

logTypes

Direction: [IN]

Valid values:

A bit combination of following values:

1= HIGH_TYPE (User <--> MEDAQLib)

2= MIDDLE_TYPE (Sensor layer <--> Interface layer)

4= LOW_TYPE (MEDAQLib <--> Hardware driver)

8= ERROR_TYPE (Any errors reported by MEDAQLib)

16= DRIVER_TYPE (Hardware driver <--> Sys driver)

32= APPL_TYPE (Application specific, see [LogToFile](#) and [LogToFileU](#))

2147483647= ALL (all log types are active)

Description: This parameter specifies the type of messages to log.

Parameter: int32_t logLevels logLevels

Direction: [IN]

Valid values:

A bit combination of following values:

- 1= EMERGENCY_LEVEL (logging emerging errors)
- 2= CRITICAL_LEVEL (logging critical errors)
- 4= ERROR_LEVEL (logging errors which occurs)
- 8= WARNING_LEVEL (logging warnings from MEDAQLib)
- 16= NOTICE_LEVEL (logging notices)
- 32= TRACE_LEVEL (logging function calls)
- 64= DATA_LEVEL (logging data in binary mode)
- 2147483647= ALL (all levels are active)

Description: This parameter specifies the kind of event to log.

Parameter: const wchar_t * logFile logFile

Direction: [IN]

Description: File name of log file as unicode string.

If it is empty or ends with '\' or '/', an automatic generated name ('SensorLog_%yyyy-%MM-%dd_%hh-%mm-%ss.%Us.txt') is appended.

Many placeholders (...) can be used for automatic name generation. Important ones are:

- %h= hour in 24 hours format (0, 1, ..., 9, 10, ..., 23)
- %hh= hour in 24 hours format (00, 01, ..., 09, 10, ..., 23)
- %H= hour in 12 hours format (1, 2, ..., 9, 10, 11, 12)
- %HH= hour in 12 hours format (01, 02, ..., 09, 10, 11, 12)
- %m= minute (0, 1, ..., 9, 10, ..., 59)
- %mm= minute (00, 01, ..., 09, 10, ..., 59)
- %s= second (0, 1, ..., 9, 10, ..., 59)
- %ss= second (00, 01, ..., 09, 10, ..., 59)
- %Ms= millisecond (000, 001, ..., 999)
- %Us= microsecond (000000, 000001, ..., 999999)
- %PP= output for morning, afternoon for US-American time format (AM, PM)
- %d= day (1, 2, ..., 9, 10, ..., 31)
- %dd= day (01, 02, ..., 09, 10, ..., 31)
- %M= month (1, 2, ..., 9, 10, 11, 12)
- %MM= month (01, 02, ..., 09, 10, 11, 12)
- %yy= year (70, 71, ..., 99, 00, 01, ..., 38)
- %yyyy= year (1970, 1971, ..., 1999, 2000, 2001, ..., 2038)
- %DoW= day of week (0= Sunday, 1= Monday, ..., 6= Saturday)
- %DoY= day of year (0, 1, ..., 9, 10, ..., 364, if a leap year 365)
- %WoY= week of year (0, 1, ..., 9, 10, ..., 52), starting with 0= the week with the first Sunday in year (US format)

Parameter: int32_t logAppend logAppend

Direction: [IN]

Valid values:

- 0= FALSE
- 1= TRUE

Description: This parameter specifies if the logfile should be cleared at opening or if the new data should be appended to file.

Parameter: int32_t logFlush	logFlush
Direction: [IN]	
Valid values:	
0= FALSE	
1= TRUE	
Description:	This parameter specifies if the logfile should be flushed after each output. In this case, it is sure that all information is stored to logfile before proceeding. But depending on the storage device it can slow down the MEDAQLib.
Parameter: int32_t logSplitSize	logSplitSize
Direction: [IN]	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Unit: KB (1024 Bytes)	
Description:	If this parameter is greater than 0, logfile is closed and reopened when this size is reached. If the file name contains placeholders (%...), a new name is generated before opening. Otherwise, the same file is opened again and if appending is off, the old content is overwritten.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterIntU](#), [SetParameterStringU](#) and [SensorCommand](#).

6.15 User logging functions (LogToFile...)

This functions allow the user to add own lines to MEDAQLib Logfile. It can be used for debugging purposes.

Name: LogToFile	LogToFile
Description:	Add a line to MEDAQLib Logfile.

Declaration:

```
ERR_CODE LogToFile (uint32_t instanceHandle, int32_t logLevel, const
                     char *location, const char *message, ...);
```

Parameter: uint32_t instanceHandle	instanceHandle
Direction: [IN]	
Description:	Number of the sensor instance, previously returned by CreateSensorInstance .

Parameter: int32_t logLevel	logLevel
Direction: [IN]	
Valid values:	
1= EMERGENCY_LEVEL	
2= CRITICAL_LEVEL	
4= ERROR_LEVEL	
8= WARNING_LEVEL	
16= NOTICE_LEVEL	
32= TRACE_LEVEL	
64= DATA_LEVEL	

Description: This parameter specifies the level for the line to log.

Parameter: const char * location	location
Direction: [IN]	
Description: Location in source code where the log line is generated. Is shown in LogFile.	
Parameter: const char * message	message
Direction: [IN]	
Description: Logging message. This parameter can be used at same as format string at C printf function (e.g. "%d, %f, %s").	
Parameter: variable argument list	variable argument list
Direction: [IN]	
Description: Depending on message string, additional parameters must be specified.	

Returns:

`ERR_NOERROR` (0) on success.
`ERR_INSTANCE_NOT_EXIST` (-24) if instanceHandle is not valid.

Name: LogToFileU **LogToFileU**

Description:

Add a line to MEDAQLib Logfile (Unicode version).

Declaration:

```
ERR_CODE LogToFileU (uint32_t instanceHandle, int32_t logLevel, const
                     wchar_t *location, const wchar_t *message, ...);
```

Parameter: uint32_t instanceHandle	instanceHandle
Direction: [IN]	
Description: Number of the sensor instance, previously returned by CreateSensorInstance .	

Parameter: int32_t logLevel	logLevel
Direction: [IN]	
Valid values:	

- 1= EMERGENCY_LEVEL
- 2= CRITICAL_LEVEL
- 4= ERROR_LEVEL
- 8= WARNING_LEVEL
- 16= NOTICE_LEVEL
- 32= TRACE_LEVEL
- 64= DATA_LEVEL

Description: This parameter specifies the level for the line to log.

Parameter: const wchar_t * location	location
Direction: [IN]	
Description: Location in source code where the log line is generated. Is shown in LogFile.	

Parameter: const wchar_t * message	message
Direction: [IN]	
Description: Logging message. This parameter can be used at same as format string at C printf function (e.g. "%d, %f, %s").	

Parameter: variable argument list variable argument list

Direction: [IN]

Description: Depending on message string, additional parameters must be specified.

Returns:

`ERR_NOERROR` (0) on success.

`ERR_INSTANCE_NOT_EXIST` (-24) if instanceHandle is not valid.

6.16 OpenSensor wrapper functions (OpenSensor...)

Wrapper functions for [OpenSensor](#) to open a specific interface. This usage of this functions makes the code shorter and more readable. This functions can be used in combination with any [Set parameters](#) functions, which must be called before.

Name: OpenSensorRS232

[**OpenSensorRS232**](#)

Description:

Set the parameters for [RS232](#) serial interface before calling [OpenSensor](#).

Declaration:

```
ERR_CODE OpenSensorRS232 (uint32_t instanceHandle, const char *port);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * port

port

Direction: [IN]

Valid values:

"COM1"

"COM2"

...

Description: Name of the serial interface. Before opening the interface using [CreateFile](#), the string is prefixed with "\.\\".

Returns:

Any error value which can be returned by the wrapped functions [SetParameterString](#) and [OpenSensor](#).

Name: OpenSensorRS232U

[**OpenSensorRS232U**](#)

Description:

Set the parameters for [RS232](#) serial interface before calling [OpenSensor](#) (Unicode version).

Declaration:

```
ERR_CODE OpenSensorRS232U (uint32_t instanceHandle, const wchar_t *port);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * port port

Direction: [IN]

Valid values:

"COM1"
"COM2"

...

Description: Name of the serial interface as unicode string. Before opening the interface using CreateFile, the string is prefixed with "\\.\\".

Returns:

Any error value which can be returned by the wrapped functions [SetParameterStringU](#) and [OpenSensor](#).

Name: OpenSensorIF2004

OpenSensorIF2004

Description:

Set the parameters for [IF2004](#) interface card before calling [OpenSensor](#).

Declaration:

```
ERR_CODE OpenSensorIF2004 (uint32_t instanceHandle, int32_t cardInstance,
                           int32_t channelNumber);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: int32_t cardInstance

cardInstance

Direction: [IN]

Valid values:

Minimum: 0
Maximum: 15

Description: Instance number of the IF2004 interface card. The cards are enumerated by the OS and the only way to distinguish is the card instance number. It does not change at least there are no changes at the PCI bus.

Parameter: int32_t channelNumber

channelNumber

Direction: [IN]

Valid values:

Minimum: PCI_CARD_IF2004: -1, otherwise 0
Maximum: 3

Description: Channel number on IF2004 Interface card. If the Encoder on the IF2004 card should be used to store values synchronously to a sensor, the channel number 3 is reserved for it. Otherwise (-1) the FIFO cannot be used for Encoder. Sensors can be carried on each channel.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterInt](#) and [OpenSensor](#).

Name: OpenSensorIF2004_USB

**OpenSensorIF2004_-
USB**
Description:

Set the parameters for USB adapter IF2004 before calling [OpenSensor](#).

Declaration:

```
ERR_CODE OpenSensorIF2004_USB (uint32_t instanceHandle, int32_t deviceInstance,
                               const char *serialNumber, const char *port, int32_t channelNumber);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by
[CreateSensorInstance](#).

Parameter: int32_t deviceInstance

deviceInstance

Direction: [IN]

Valid values:

Minimum: 0

Maximum: 255

Description: Instance number of the USB adapter IF2004. The devices
are enumerated by the OS. It may change if any devices are plugged
or unplugged at USB bus.

Parameter: const char * serialNumber

serialNumber

Direction: [IN]

Valid values:

Minimum: "0000001"

Maximum: "9999999"

Description: Serial number of the USB adapter (optional). If not specified
(null or empty), port or deviceInstance is used. Leading zeros can be
dismissed.

Parameter: const char * port

port

Direction: [IN]

Valid values:

"COM1"

"COM2"

...

Description: Name of the serial interface part of USB adapter, e.g.
COM1, COM2, ... (optional). If not specified (null or empty), device-
Instance is used.

Parameter: int32_t channelNumber

channelNumber

Direction: [IN]

Valid values:

Minimum: USB_ADAPTER_IF2004: -1, otherwise 0

Maximum: 4

Description: Channel number on USB adapter IF2004. Channel num-
bers 0-3 are used for sensors. Channel number 4 selects the
Digital inputs on the USB adapter IF2004 and provides their val-
ues synchronously to a sensor. Channel number -1 is valid for
USB_ADAPTER_IF2004 at CreateSensorInstance only and is used
for parameter setting in USB adapter IF2004 itself.

Returns:

Any error value which can be returned by the wrapped functions [SetParam-
eterInt](#), [SetParameterString](#) and [OpenSensor](#).

Name: OpenSensorIF2004_USBU

**OpenSensorIF2004_-
USBU**
Description:

Set the parameters for USB adapter IF2004 before calling [OpenSensor](#) (Unicode version).

Declaration:

```
ERR_CODE OpenSensorIF2004_USBU (uint32_t instanceHandle, int32_t deviceInstance,
                                const wchar_t *serialNumber, const wchar_t *port, int32_t channelNumber);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: int32_t deviceInstance

deviceInstance

Direction: [IN]

Valid values:

Minimum: 0

Maximum: 255

Description: Instance number of the USB adapter IF2004. The devices are enumerated by the OS. It may change if any devices are plugged or unplugged at USB bus.

Parameter: const wchar_t * serialNumber

serialNumber

Direction: [IN]

Valid values:

Minimum: "00000001"

Maximum: "9999999"

Description: Serial number of the USB adapter (optional). If not specified (null or empty), port or deviceInstance is used. Leading zeros can be dismissed.

Parameter: const wchar_t * port

port

Direction: [IN]

Valid values:

"COM1"

"COM2"

...

Description: Name of the serial interface part of USB adapter, e.g. COM1, COM2, ... (optional). If not specified (null or empty), deviceInstance is used.

Parameter: int32_t channelNumber

channelNumber

Direction: [IN]

Valid values:

Minimum: USB_ADAPTER_IF2004: -1, otherwise 0

Maximum: 4

Description: Channel number on USB adapter IF2004. Channel numbers 0-3 are used for sensors. Channel number 4 selects the Digital inputs on the USB adapter IF2004 and provides their values synchronously to a sensor. Channel number -1 is valid for USB_ADAPTER_IF2004 at CreateSensorInstance only and is used for parameter setting in USB adapter IF2004 itself.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterInt](#), [SetParameterStringU](#) and [OpenSensor](#).

Name: OpenSensorIF2008

OpenSensorIF2008

Description:

Set the parameters for IF2008 interface card before calling [OpenSensor](#).

Declaration:

```
ERR_CODE OpenSensorIF2008 (uint32_t instanceHandle, int32_t cardInstance,
                           int32_t channelNumber);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: int32_t cardInstance

cardInstance

Direction: [IN]

Valid values:

Minimum: 0

Maximum: 15

Description: Instance number of the IF2008 interface card. The cards are enumerated by the OS and the only way to distinguish is the card instance number. It does not change at least there are no changes at the PCI bus.

Parameter: int32_t channelNumber

channelNumber

Direction: [IN]

Valid values:

Condition: PCI_CARD_IF2008

-1= No data acquisition

6= Encoder 1

7= Encoder 2

8= Digital IN

9= Digital RxD

10= ADC 1 (Analog/Digital converter)

11= ADC 2 (Analog/Digital converter)

Valid values:

Condition: otherwise (sensors)

0= Sensor 1 (Base Board, Connector X1)

1= Sensor 2 (Base Board, Connector X1)

2= Sensor 3 (Base Board, Connector X2)

3= Sensor 4 (Base Board, Connector X2)

4= Sensor 5 (Extension Board, Connector X1)

5= Sensor 6 (Extension Board, Connector X1)

Description: Channel number on IF2008 Interface card.

Attention! Sensor 5 and 6 are only available if IF2008E extension card is installed. Digital IN is only available if IF2008E extension card or IF2008IO extension slot is installed. ADC is only available if IF2008E extension card is installed.

-1 means, no data channel is written to FIFO and cannot be read using [TransferData](#) or [Poll](#). This mode can be used if IF2008 should only be parametrized.

Returns:

Any error value which can be returned by the wrapped functions [SetParamaterInt](#) and [OpenSensor](#).

Name: OpenSensorIF2008_ETH

OpenSensorIF2008_ETH
Description:

Set the parameters for IF2008 ethernet adapter before calling [OpenSensor](#).

Declaration:

```
ERR_CODE OpenSensorIF2008_ETH (uint32_t instanceHandle, const char
    *remoteAddr, int32_t channelNumber);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * remoteAddr

remoteAddr

Direction: [IN]

Description: IP address of the remote sensor (TCP server).

Parameter: int32_t channelNumber

channelNumber

Direction: [IN]

Valid values:

- 1= No data acquisition
- 0= Channel 1
- 1= Channel 2
- 2= Channel 3
- 3= Channel 4
- 4= Channel 5
- 5= Channel 6
- 6= Channel 7
- 7= Channel 8
- 8= Digital input (only for [ETH_ADAPTER_IF2008](#))

Description: Channel number on IF2008 ethernet adapter.

-1 means, no data channel is written to FIFO and cannot be read using [TransferData](#) or [Poll](#). This mode can be used if IF2008 ethernet adapter should only be parametrized.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterInt](#) and [OpenSensor](#).

Name: OpenSensorIF2008_ETHU

OpenSensorIF2008_
ETHU
Description:

Set the parameters for IF2008 ethernet adapter before calling [OpenSensor](#) (Unicode version).

Declaration:

```
ERR_CODE OpenSensorIF2008_ETHU (uint32_t instanceHandle, const wchar_t
    *remoteAddr, int32_t channelNumber);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * remoteAddr remoteAddr

Direction: [IN]

Description: IP address of the remote sensor (TCP server) as unicode string.

Parameter: int32_t channelNumber channelNumber

Direction: [IN]

Valid values:

-1= No data acquisition

0= Channel 1

1= Channel 2

2= Channel 3

3= Channel 4

4= Channel 5

5= Channel 6

6= Channel 7

7= Channel 8

8= Digital input (only for [ETH_ADAPTER_IF2008](#))

Description: Channel number on IF2008 ethernet adapter.

-1 means, no data channel is written to FIFO and cannot be read using [TransferData](#) or [Poll](#). This mode can be used if IF2008 ethernet adapter should only be parametrized.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterInt](#) and [OpenSensor](#).

Name: OpenSensorTCPIP

OpenSensorTCPIP

Description:

Set the parameters for TCP/IP ethernet interface before calling [OpenSensor](#).

Declaration:

```
ERR_CODE OpenSensorTCPIP (uint32_t instanceHandle, const char *remoteAddr);
```

Parameter: uint32_t instanceHandle instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * remoteAddr remoteAddr

Direction: [IN]

Description: IP address of the remote sensor (TCP server).

Returns:

Any error value which can be returned by the wrapped functions [SetParameterString](#) and [OpenSensor](#).

Name: OpenSensorTCPIPU

OpenSensorTCPIPU
Description:

Set the parameters for TCP/IP ethernet interface before calling [OpenSensor](#) (Unicode version).

Declaration:

```
ERR_CODE OpenSensorTCPIPU (uint32_t instanceHandle, const wchar_t *remoteAddr);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * remoteAddr

remoteAddr

Direction: [IN]

Description: IP address of the remote sensor (TCP server) as unicode string.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterStringU](#) and [OpenSensor](#).

Name: OpenSensorWinUSB

OpenSensorWinUSB
Description:

Set the parameters for USB interface via [WinUSB](#) before calling [OpenSensor](#).

Declaration:

```
ERR_CODE OpenSensorWinUSB (uint32_t instanceHandle, int32_t deviceInstance);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: int32_t deviceInstance

deviceInstance

Direction: [IN]

Valid values:

Minimum: 0

Maximum: 255

Description: Instance number of the USB device. The devices are enumerated by the OS and the only way to distinguish is the device instance number. It does not change at least there are no changes at the USB bus (plug/unplug devices).

Returns:

Any error value which can be returned by the wrapped functions [SetParameterInt](#) and [OpenSensor](#).

6.17 ExecSCmd wrapper functions (...ExecSCmd...)

Wrapper functions for a set of [Set parameters](#) and [Get parameters](#) functions and [SensorCommand](#). This usage of this functions makes the code shorter and more readable. This functions can be used in combination with any [Set parameters](#) functions, which must be called before.

Name: ExecSCmd

ExecSCmd

Description:

Set the sensor command name and executes the sensor command.

Declaration:

```
ERR_CODE ExecSCmd (uint32_t instanceHandle, const char *sensorCommand);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)).

Returns:

Any error value which can be returned by the wrapped functions [SetParameterString](#) and [SensorCommand](#).

Name: ExecSCmdU

ExecSCmdU

Description:

Set the sensor command name and executes the sensor command (Unicode version).

Declaration:

```
ERR_CODE ExecSCmdU (uint32_t instanceHandle, const wchar_t *sensorCommand);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)) as unicode string.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterStringU](#) and [SensorCommand](#).

Name: SetIntExecSCmd

SetIntExecSCmd
Description:

Set the sensor command name and an integer parameter and executes the sensor command.

Declaration:

```
ERR_CODE SetIntExecSCmd (uint32_t instanceHandle, const char *sensorCommand,
                         const char *paramName, int32_t paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)).

Parameter: const char * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: int32_t paramValue

paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterString](#), [SetParameterInt](#) and [SensorCommand](#).

Name: SetIntExecSCmdU

SetIntExecSCmdU
Description:

Set the sensor command name and an integer parameter and executes the sensor command (Unicode version).

Declaration:

```
ERR_CODE SetIntExecSCmdU (uint32_t instanceHandle, const wchar_t *sensorCommand,
                          const wchar_t *paramName, int32_t paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)) as unicode string.

Parameter: const wchar_t * paramName paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: int32_t paramValue paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterStringU](#), [SetParameterIntU](#) and [SensorCommand](#).

Name: SetDoubleExecSCmd **SetDoubleExecSCmd**

Description:

Set the sensor command name and an double parameter and executes the sensor command.

Declaration:

```
ERR_CODE SetDoubleExecSCmd (uint32_t instanceHandle, const char *sensorCommand,
                            const char *paramName, double paramValue);
```

Parameter: uint32_t instanceHandle instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * sensorCommand sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)).

Parameter: const char * paramName paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: double paramValue paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterString](#), [SetParameterDouble](#) and [SensorCommand](#).

Name: SetDoubleExecSCmdU

SetDoubleExecSCmdU
Description:

Set the sensor command name and an double parameter and executes the sensor command (Unicode version).

Declaration:

```
ERR_CODE SetDoubleExecSCmdU (uint32_t instanceHandle, const wchar_t *sensorCommand, const wchar_t *paramName, double paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)) as unicode string.

Parameter: const wchar_t * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: double paramValue

paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterStringU](#), [SetParameterDoubleU](#) and [SensorCommand](#).

Name: SetStringExecSCmd

SetStringExecSCmd
Description:

Set the sensor command name and an string parameter and executes the sensor command.

Declaration:

```
ERR_CODE SetStringExecSCmd (uint32_t instanceHandle, const char *sensorCommand, const char *paramName, const char *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)).

Parameter: const char * paramName paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: const char * paramValue paramValue

Direction: [IN]

Description: Value of the parameter.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterString](#) and [SensorCommand](#).

Name: SetStringExecSCmdU

SetStringExecSCmdU

Description:

Set the sensor command name and an string parameter and executes the sensor command (Unicode version).

Declaration:

```
ERR_CODE SetStringExecSCmdU (uint32_t instanceHandle, const wchar_t *sensorCommand, const wchar_t *paramName, const wchar_t *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)) as unicode string.

Parameter: const wchar_t * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: const wchar_t * paramValue

paramValue

Direction: [IN]

Description: Value of the parameter as unicode string.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterStringU](#) and [SensorCommand](#).

Name: ExecSCmdGetInt

ExecSCmdGetInt
Description:

Set the sensor command name, executes the sensor command and get a integer parameter.

Declaration:

```
ERR_CODE ExecSCmdGetInt (uint32_t instanceHandle, const char *sensorCommand,
                        const char *paramName, int32_t *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)).

Parameter: const char * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: int32_t * paramValue

paramValue

Direction: [OUT]

Description: Pointer to a variable retrieving the parameter

Returns:

Any error value which can be returned by the wrapped functions [SetParameterString](#), [SensorCommand](#) and [GetParameterInt](#).

Name: ExecSCmdGetIntU

ExecSCmdGetIntU
Description:

Set the sensor command name, executes the sensor command and get a integer parameter (Unicode version).

Declaration:

```
ERR_CODE ExecSCmdGetIntU (uint32_t instanceHandle, const wchar_t *sensorCommand,
                          const wchar_t *paramName, int32_t *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)) as unicode string.

Parameter: const wchar_t * paramName paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: int32_t * paramValue paramValue

Direction: [OUT]

Description: Pointer to a variable retrieving the parameter

Returns:

Any error value which can be returned by the wrapped functions [SetParameterStringU](#), [SensorCommand](#) and [GetParameterIntU](#).

Name: ExecSCmdGetDouble

ExecSCmdGetDouble

Description:

Set the sensor command name, executes the sensor command and get a double parameter.

Declaration:

```
ERR_CODE ExecSCmdGetDouble (uint32_t instanceHandle, const char *sensorCommand,
                           const char *paramName, double *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)).

Parameter: const char * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: double * paramValue

paramValue

Direction: [OUT]

Description: Pointer to a variable retrieving the parameter

Returns:

Any error value which can be returned by the wrapped functions [SetParameterString](#), [SensorCommand](#) and [GetParameterDouble](#).

Name: ExecSCmdGetDoubleU

ExecSCmdGetDoubleU
Description:

Set the sensor command name, executes the sensor command and get a double parameter (Unicode version).

Declaration:

```
ERR_CODE ExecSCmdGetDoubleU (uint32_t instanceHandle, const wchar_t *sensorCommand, const wchar_t *paramName, double *paramValue);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)) as unicode string.

Parameter: const wchar_t * paramName

paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: double * paramValue

paramValue

Direction: [OUT]

Description: Pointer to a variable retrieving the parameter

Returns:

Any error value which can be returned by the wrapped functions [SetParameterStringU](#), [SensorCommand](#) and [GetParameterDoubleU](#).

Name: ExecSCmdGetString

ExecSCmdGetString
Description:

Set the sensor command name, executes the sensor command and get a string parameter.

Declaration:

```
ERR_CODE ExecSCmdGetString (uint32_t instanceHandle, const char *sensorCommand, const char *paramName, char *paramValue, uint32_t *maxLen);
```

Parameter: uint32_t instanceHandle

instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const char * sensorCommand

sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)).

Parameter: const char * paramName paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as string.

Parameter: char * paramValue paramValue

Direction: [OUT]

Description: Pointer to a variable (character buffer) retrieving the parameter.

Parameter: uint32_t * maxLen maxLen

Direction: [IN/OUT]

Description: The buffer must be allocated by the application. The size of the buffer is specified at maxLen. If the resulting string is larger than maxLen, it is truncated and [ERR_NOMEMORY](#) (-19) is returned. The real length of the string (maybe truncated) is returned in maxLen too. If paramValue is nullptr, the length of the containing string is returned in maxLen.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterString](#), [SensorCommand](#) and [GetParameterString](#).

Name: ExecSCmdGetStringU

ExecSCmdGetStringU

Description:

Set the sensor command name, executes the sensor command and get a string parameter (Unicode version).

Declaration:

```
ERR_CODE ExecSCmdGetStringU (uint32_t instanceHandle, const wchar_t *sensorCommand, const wchar_t *paramName, wchar_t *paramValue, uint32_t *maxLen);
```

Parameter: uint32_t instanceHandle instanceHandle

Direction: [IN]

Description: Number of the sensor instance, previously returned by [CreateSensorInstance](#).

Parameter: const wchar_t * sensorCommand sensorCommand

Direction: [IN]

Description: Name of the sensor command (used for parameter [S_Command](#)) as unicode string.

Parameter: const wchar_t * paramName paramName

Direction: [IN]

Valid value: All available values are listed at chapter [Sensor commands](#) and following.

Description: Name of the parameter as unicode string.

Parameter: wchar_t * paramValue paramValue

Direction: [OUT]

Description: Pointer to a variable (wide character buffer) retrieving the parameter.

Parameter: `uint32_t * maxLen`

`maxLen`

Direction: [IN/OUT]

Description: The buffer must be allocated by the application. The size of the buffer (in characters, not bytes) is specified at `maxLen`. If the resulting string is larger than `maxLen`, it is truncated and `ERR_NOMEMORY` (-19) is returned. The real length of the string (maybe truncated) is returned in `maxLen` too.

If `paramValue` is `nullptr`, the length of the containing string is returned in `maxLen`.

Returns:

Any error value which can be returned by the wrapped functions [SetParameterStringU](#), [SensorCommand](#) and [GetParameterStringU](#).

7 Parameters

7.1 Naming conventions

7.1.1 Parameter prefix (IP_, SP_, SA_)

Parameters for opening the interface (Interface parameters, IP_...) are used in the Function [OpenSensor](#).

Parameters for communicating with the sensor (Sensor parameters, SP_...) and answer from sensor (Sensor answer, SA_...) are used in Function [SensorCommand](#).

7.1.2 Parameter direction

Direction in parameters is defined as follows:

Down: From application to driver or sensor.

Up: From sensor or driver to application.

7.1.3 Obligatory and optional parameters

Parameters without a default value are obligatory and must be specified. For the other parameters, the default value is used if not specified.

7.1.4 Multiple commands at once

When multiple commands (only differ by a number) are available (e.g. for several channels), they may be combined into one section in documentation

The number is represented by a placeholder in terms of <n>. The range of n is defined within this chapter.

7.1.5 Multiple parameters at once

When a parameter contains two periods (e.g: SP_X1..16), it stands for a sequence of parameters (SP_X1, SP_X2, SP_X3, ..., SP_X16).

For sensor parameter, each parameter must be specified, except an [Update_...](#) or [Reset_...](#) meta command is available.

For answer parameters each parameter is available to read. If the number is variable (e.g. SA_Pos1..x), another parameter (above) contains the number.

7.2 Interface parameters

An interface is the hardware device directly opened by MEDAQLib to establish connection to the sensor. It communicates directly with the sensor.

Attention! The [CONTROLLER_CSP2008](#) is not an interface, it is treated as sensor (resp. controller). The sensors behind cannot be accessed by MEDAQLib.

7.2.1 All Interfaces

Parameter: String IP_Interface	IP_Interface
Direction: Down	
Valid values:	
"RS232" "IF2004_USB" "IF2004" "IF2008" "IF2008_ETH" "TCP/IP" "WinUSB"	
Description: Interface type where the sensor is plugged on. If the IF2001_USB (RS422) is used "RS232" must be set, because a RS232 interface is emulated. With an RS422/RS232 to Ethernet converter additional sensors can be connected via "TCP/IP".	
Parameter: int32_t IP_AutomaticMode	IP_AutomaticMode
Direction: Down	
Valid values:	
A bit combination of following values: "First bit (1)= Retrieve sensor information to setup MEDAQLib" "Second bit (2)= Activate data output at sensor"	
Default: 1	
Description: First bit (1) allows MEDAQLib to retrieve information from sensor automatically if needed. Normally this happens at OpenSensor , but in some cases, it can happen after SensorCommand , too. Second bit (2) allows MEDAQLib to change sensor interface parameters (if needed) so is outputs data. This only happens at OpenSensor .	
Parameter: int32_t IP_ScaleErrorValues	IP_ScaleErrorValues
Direction: Down	
Valid values:	
1= last valid value 2= set to fixed value 3= set to negative error value	
Default: 2	
Description: If sensor values are not valid, they cannot be scaled. So invalid values can be set to the last valid value or can be set to a fixed value (see IP_FixedErrorValue) or can be set to a negative error value: For SENSOR_ODC2500 and SENSOR_ODC2600 :	

-10	DSP No edge
-11	DSP At the beginning of the picture
-12	DSP At the end of the picture
-13	DSP Dark - bright edge
-14	DSP Bright - dark edge
-15	DSP Min. number of edges
-16	DSP Max. number of edges
-17	DSP Invalid measuring program
-18	DSP Segment 1st edge >= 2nd edge
-19	DSP Segment number of edges < last edge
-20	DSP Invalid working distance
-22	ARM Laser off
-23	ARM Invalid float
-24	ARM DMA setup error
-90	ASCII mode at IF2004 not supported

For [SENSOR_ILD1302](#), [SENSOR_ILD1401](#), [SENSOR_ILD1402](#), [SENSOR_ILD1700](#) and [SENSOR_ILD2200](#) (not every sensor supports all errors):

-1	F1 bad objekt (no objekt cognizable)
-2	F2 out of range + (to near at sensor)
-3	F3 out of range - (to far from sensor)
-4	F4 poor target (objekt not evaluable)
-5	F5 Laser off (external laser off)
-6	Measured object moves towards sensor. SENSOR_ILD1700 : Sensor in trigger mode Trigger and pulses comes to fast.
-7	Measured object moves away from sensor

For [SENSOR_ILD1220](#), [SENSOR_ILD1320](#), [SENSOR_ILD1420](#), [SENSOR_ILD1750](#), [SENSOR_ILD1900](#), [SENSOR_ILD2300](#), \fixh{SENSOR_ODC2520}, [SENSOR_IFD2445](#), [SENSOR_IFD2451](#), [SENSOR_IFD2461](#), [SENSOR_IFD2471](#), \fixlt{SENSOR_IFD242x}{SENSOR_IFD2421}, \fixlt{SENSOR_IFD242x}{SENSOR_IFD2422}, [SENSOR_ACS7000](#), [SENSOR_IMC5400](#), [SENSOR_IMC5600](#), [SENSOR_ILR2250](#), [SENSOR_MFA](#), [CONTROLLER_CSP2008](#), [CONTROLLER_CBOX](#), [ETH_ADAPTER_IF2008](#) (not every sensor supports all errors):

-1	Scaling error RS422 interface underflow
-2	Scaling errors RS422 interface overflow
-3	Too much data for selected baudrate
-4	No peak/edge available
-5	Peak is in front of the measuring range
-6	Peak is after the measuring range
-7	Measurement cannot be calculated
-8	Measurement cannot be evaluated, global error
-9	Peak is too wide
-10	Laser beam is off

For **SENSOR_ILR118x**:

For error values, scaled values are set to 0, raw values are set to positive error values:

15	E15 - Excessively poor reflexes. Distance sensor (Front edge) against target < 0.1m.
16	E16 - Excessively strong reflexes.
17	E17 - Too much steady light (for example sun).
18	E18 - Only in DX mode (50 Hz): Too much difference between measured and pre-calculated value.
19	E19 - Only in DX mode (50 Hz): Target motion speed > 10 m/s.
23	E23 - Temperature below -10 °C
24	E24 - Temperature above +60 °C
31	E31 - Faulty EEPROM checksum, hardware error.
51	E51 - Failure to set avalanche voltage of diode laser. 1. straylight or 2. hardware error.
52	E52 - Laser current too high / laser defective.
53	E53 - One or more parameters in the EEPROM not set (Consequence: Division by 0).
54	E54 - Hardware error (PLL).
55	E55 - Hardware error.
61	E61 - Used parameter is inadmissible, invalid command sent.
62	E62 - 1. Hardware error 2. wrong value in interface communication (Parity error SIO).
63	E63 - SIO overflow.
64	E64 - Framing-Error SIO.

For **SENSOR_ILR1191**:

For error values, scaled values are set to 0, raw values are set to positive error values:

2	E02 - No target.
4	E04 - Laser defect.

For **SENSOR_ACS7000**:

-1	RS422 scaling underflow
-2	RS422 scaling overflow
-3	Too much data for this baudrate

Parameter: double IP_FixedErrorValue

IP_FixedErrorValue

Direction: Down

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Default: -1.79769e+308 (-DBL_MAX)

Description: If IP_ScaleErrorValues is set to fixed value, this value is returned in case of an error.

Parameter: int32_t IP_RingBufferSize

IP_RingBufferSize

Direction: Down

Valid values:

Minimum: 512

Maximum: 1073741824 (1 GB)

Unit: Bytes

Default: 1048576 (1 MB)

Description: Data (values and sensor answer) is collected from sensor into a ring buffer. From there it is converted (value) or interpreted (sensor answer). If the ring buffer size is too small and data is not transferred fast enough an overflow occur. The buffer must be large enough to hold a complete sensor answer (especially IFD's with command Get_CCD or Get_DarkSig, ...).

Parameter: int32_t IP_EnableLogging

IP_EnableLogging

Direction: Down

Valid values:

0= FALSE

1= TRUE

Default: 0

Description: This parameter enables or disables logging to file for debugging purposes.

Parameter: int32_t IP_LogType

IP_LogType

Direction: Down

Valid values:

A bit combination of following values:

1= HIGH_TYPE (User <--> MEDAQLib)

2= MIDDLE_TYPE (Sensor layer <--> Interface layer)

4= LOW_TYPE (MEDAQLib <--> Hardware driver)

8= ERROR_TYPE (Any errors reported by MEDAQLib)

16= DRIVER_TYPE (Hardware driver <--> Sys driver)

32= APPL_TYPE (Application specific, see [LogFile](#) and [LogFileU](#))

2147483647= ALL (all log types are active)

Default: 2147483647 (INT32_MAX)

Description: This parameter specifies the type of messages to log.

Parameter: int32_t IP_LogLevel IP_LogLevel

Direction: Down

Valid values:

A bit combination of following values:

- 1= EMERGENCY_LEVEL (logging emerging errors)
- 2= CRITICAL_LEVEL (logging critical errors)
- 4= ERROR_LEVEL (logging errors which occurs)
- 8= WARNING_LEVEL (logging warnings from MEDAQLib)
- 16= NOTICE_LEVEL (logging notices)
- 32= TRACE_LEVEL (logging function calls)
- 64= DATA_LEVEL (logging data in binary mode)
- 2147483647= ALL (all levels are active)

Default: 2147483647 (INT32_MAX)

Description: This parameter specifies the kind of event to log.

Parameter: String IP_LogFile IP_LogFile

Direction: Down

Default:

Description: File name of log file.

If it is empty or ends with '\' or '/', an automatic generated name ('SensorLog_%yyyy-%MM-%dd %hh-%mm-%ss.%Us.txt') is appended.

Many placeholders (...) can be used for automatic name generation.

Important ones are:

- %h= hour in 24 hours format (0, 1, ..., 9, 10, ..., 23)
- %hh= hour in 24 hours format (00, 01, ..., 09, 10, ..., 23)
- %H= hour in 12 hours format (1, 2, ..., 9, 10, 11, 12)
- %HH= hour in 12 hours format (01, 02, ..., 09, 10, 11, 12)
- %m= minute (0, 1, ..., 9, 10, ..., 59)
- %mm= minute (00, 01, ..., 09, 10, ..., 59)
- %s= second (0, 1, ..., 9, 10, ..., 59)
- %ss= second (00, 01, ..., 09, 10, ..., 59)
- %Ms= millisecond (000, 001, ..., 999)
- %Us= microsecond (000000, 000001, ..., 999999)
- %PP= output for morning, afternoon for US-American time format (AM, PM)
- %d= day (1, 2, ..., 9, 10, ..., 31)
- %dd= day (01, 02, ..., 09, 10, ..., 31)
- %M= month (1, 2, ..., 9, 10, 11, 12)
- %MM= month (01, 02, ..., 09, 10, 11, 12)
- %yy= year (70, 71, ..., 99, 00, 01, ..., 38)
- %yyyy= year (1970, 1971, ..., 1999, 2000, 2001, ..., 2038)
- %DoW= day of week (0= Sunday, 1= Monday, ..., 6= Saturday)
- %DoY= day of year (0, 1, ..., 9, 10, ..., 364, if a leap year 365)
- %WoY= week of year (0, 1, ..., 9, 10, ..., 52), starting with 0= the week with the first Sunday in year (US format)
- %[NBR]= generates a consecutive number which increments with each new logfile up to the number NBR

Parameter: int32_t IP_LogAppend IP_LogAppend

Direction: Down

Valid values:

0= FALSE

1= TRUE

Default: 1 if %[NBR] is specified within IP_LogFile, otherwise 0

Description: This parameter specifies if the logfile should be cleared at opening or if the new data should be appended to file.

Parameter: int32_t IP_LogFlush

IP_LogFlush

Direction: Down

Valid values:

0= FALSE

1= TRUE

Default: 0

Description: This parameter specifies if the logfile should be flushed after each output. In this case, it is sure that all information is stored to logfile before proceeding. But depending on the storage device it can slow down the MEDAQLib.

Example how to enable MEDAQLib logging in your application:

```
uint32_t instance= CreateSensorInstance (...);
SetParameterString (instance, "IP_Interface", "...");
/* Set any other interface parameters */
SetParameterInt (instance, "IP_EnableLogging", 1);
SetParameterString (instance, "IP_LogFile", "C:\SensorLog.txt");
ERR_CODE err= OpenSensor (instance);
```

Alternatively you can use the function EnableLogging:

```
uint32_t instance= CreateSensorInstance (...);
EnableLogging (instance, 1, INT32_MAX, INT32_MAX, "C:\SensorLog.txt", 1,
0, 0);
ERR_CODE err= OpenSensor (instance);
```

Parameter: int32_t IP_LogSplitSize

IP_LogSplitSize

Direction: [IN]

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Unit: KB (1024 Bytes)

Default: 0

Description: If this parameter is greater than 0, logfile is closed and reopened when this size is reached. If the file name contains placeholders (%...), a new name is generated before opening. Otherwise, the same file is opened again and if appending is off, the old content is overwritten.

Parameter: int32_t IP_MaxPacketSize

IP_MaxPacketSize

Direction: Down

Valid values:

Minimum: 1

Maximum: 2147483647 (INT32_MAX)

Unit: Bytes

Default: SENSOR_ILD1401: 1, otherwise INT32_MAX

Description: Maximum size of a block which can be transferred to the sensor at once. Because of a small receive FIFO in ILD1401 only 1 Byte after another can be sent to sensor with a break between.

At IF2008_ETH this parameter is ignored.

Parameter: int32_t IP_PacketDelay IP_PacketDelay

Direction: Down

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Unit: ms

Default: [SENSOR_ILD1401](#): 1 (RS232) or 3 (TCP/IP), otherwise 0

Description: Break time between sending two blocks to a sensor (see IP_MaxPacketSize).

At [IF2008_ETH](#) this parameter is ignored.

Parameter: int32_t IP_TimerResolution IP_TimerResolution

Direction: Down

Unit: ms

Valid values:

-1 = Do not set timer resolution.

0 = Use greatest possible accuracy.

1..255 = Resolution in milliseconds.

Unit: ms

Default: [SENSOR_MEBUS](#): 0, -1

Description: Timer resolution (for Windows scheduler, set by timeBeginPeriod).

For some sensors which sends large amount of data in small packets, it could be useful to set lower timer resolution. This will instruct the scheduler to execute the acquisition thread more often.

Parameter: int32_t CP_BaseLevelTimeout CP_BaseLevelTimeout

Direction: Down

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Unit: ms

Default: 0

Description: This global timeout is added to the special timeout (specified by [CP_SensorAnswerTimeout](#) or internally) when waiting for any sensor answer at [SensorCommand](#).

7.2.2 RS232

Following sensors supports this interface:

[SENSOR_ILD1401](#), [SENSOR_ODC1202](#), [SENSOR_ODC2500](#), [SENSOR_ODC2600](#), [SENSOR_IFD2401](#), [SENSOR_IFD2431](#), [SENSOR_ILR110x_115x](#), [SENSOR_ILR118x](#), [SENSOR_ILR1191](#), [SENSOR_ILR2250](#), [SENSOR_MFA](#) (native).
[SENSOR_ILD1220](#), [SENSOR_ILD1302](#), [SENSOR_ILD1320](#), [SENSOR_ILD1402](#), [SENSOR_ILD1420](#), [SENSOR_ILD1700](#), [SENSOR_ILD1750](#), [SENSOR_ILD1900](#), [SENSOR_ILD2200](#), [SENSOR_ILD2300](#), [SENSOR_IFD2421](#), [SENSOR_IFD2422](#), [SENSOR_IFD2445](#), [SENSOR_IFD2451](#), [SENSOR_IFD2461](#), [SENSOR_IFD2471](#), [SENSOR_ODC2520](#), [SENSOR_IMC5400](#), [SENSOR_IMC5600](#), [SENSOR_ACS7000](#), [CONTROLLER_CSP2008](#), [CONTROLLER_CBOX](#) (additional, e.g. [IF2001_USB](#) (RS422) and RS232 high level interface).

[SENSOR_DT6120](#) and [SENSOR_MEBUS](#) (additional, e.g. RS485/USB converter and RS232 high level interface).

Parameter: String IP_Port IP_Port

Direction: Down

Valid values:

"COM1"
"COM2"

...

Description: Name of the serial interface. Before opening the interface using CreateFile, the string is prefixed with "\\.\\".

Parameter: int32_t IP_Baudrate IP_Baudrate

Direction: Down

Valid values:

SENSOR_ILD1401: 38400
 SENSOR_ILD1302, SENSOR_ILD1402: 115200, 57600, 38400, 19200,
 9600
 SENSOR_ILD1220, SENSOR_ILD1320, SENSOR_ILD1420: 1000000,
 921600, 691200, 460800, 256000, 230400, 128000, 115200, 56000,
 19200, 9600
 SENSOR_ILD1700: 115200, 57600, 19200, 9600
 SENSOR_ILD2200: 691200, 1250000
 SENSOR_IFD2401, SENSOR_IFD2431: 460800, 230400, 115200, 57600,
 38400, 19200, 9600
 SENSOR_ILD1750, SENSOR_ILD1900, SENSOR_IMC5400, SENSOR_IMC5600:
 4000000, 3000000, 2000000, 921600, 691200, 460800, 230400,
 115200, 9600
 SENSOR_IFD2421, SENSOR_IFD2422: 4000000, 3000000, 2000000,
 921600, 691200, 460800, 230400, 115200, 38400 (from firmware
 V001.041.086), 9600
 SENSOR_ILD2300, SENSOR_IFD2445, SENSOR_IFD2451, SENSOR_IFD2461,
 SENSOR_IFD2471, SENSOR_ODC2520: 4000000, 3500000, 3000000,
 2500000, 2000000, 1500000, 921600, 691200, 460800, 230400,
 115200, 9600
 SENSOR_ODC1202: 115200, 57600, 38400, 19200, 9600
 SENSOR_ODC2500, SENSOR_ODC2600: 115200, 38400, 19200, 9600
 SENSOR_ILR110x_115x: 57600, 38400, 19200, 9600, 4800
 SENSOR_ILR118x: 38400, 19200, 9600, 4800
 SENSOR_ILR1191: 460800, 230400, 115200, 57600, 38400, 19200,
 9600
 CONTROLLER_CSP2008: 115200, 691200
 SENSOR_DT6120: 921600, 460800, 230400, 115200, 9600
 SENSOR_MEBUS: 921600, 460800, 256000, 230400, 115200, 9600
 SENSOR_ACS7000: 4000000, 3500000, 2000000, 1500000, 921600,
 691200, 460800, 230400, 115200, 9600
 SENSOR_ILR2250, SENSOR_MFA: 230400, 115200, 9600
 CONTROLLER_CBOX: 8000000, 4000000, 3500000, 3000000, 2500000,
 2000000, 1500000, 921600, 691200, 460800, 230400, 115200, 9600

Unit: Baud

Default:

SENSOR_ILD1401, SENSOR_ILR118x, SENSOR_MEBUS: 9600,
 SENSOR_ODC1202: 19200,
 SENSOR_ILD1302, SENSOR_ILD1402, SENSOR_ILD1700, SENSOR_IFD2401,
 SENSOR_IFD2431, SENSOR_IFD2421, SENSOR_IFD2422, SENSOR_IFD2445,
 SENSOR_IFD2451, SENSOR_IFD2461, SENSOR_IFD2471, SENSOR_ILR1191,

SENSOR_IMC5400, SENSOR_IMC5600, SENSOR_ACS7000, SENSOR_ILR2250,
 SENSOR_MFA, SENSOR_ODC2500, SENSOR_ODC2520, SENSOR_ODC2600,
 SENSOR_DT6120, CONTROLLER_CSP2008, CONTROLLER_CBOX: 115200,
 SENSOR_ILD1220, SENSOR_ILD1320, SENSOR_ILD1420, SENSOR_ILD1750,
 SENSOR_ILD1900: 921600,
 SENSOR_ILD2200, SENSOR_ILD2300: 691200,
 SENSOR_ILR110x_115x: 38400

Description: Speed of the RS422 serial connection.

Parameter: int32_t IP_Stopbits IP_Stopbits

Direction: Down

Valid values:

SENSOR_ILR110x_115x, SENSOR_ODC2500, SENSOR_ODC2600: 0= ONESTOPBIT,
 2= TWOSTOPBITS
 otherwise 1= ONE5STOPBITS

Default:

SENSOR_ODC2500, SENSOR_ODC2600: 2,
 otherwise: 0

Description: Number of stop bits of the serial connection.

Parameter: int32_t IP_Parity IP_Parity

Direction: Down

Valid values:

SENSOR_ILR110x_115x, SENSOR_DT6120, SENSOR_MEBUG: 2= EVEN-
 PARITY
 SENSOR_ODC2500, SENSOR_ODC2600: 0= NOPARITY, 1= ODD-
 PARITY, 2= EVENPARITY
 otherwise: 0= NOPARITY

Default:

SENSOR_ILR110x_115x, SENSOR_DT6120, SENSOR_MEBUG: 2,
 otherwise: 0

Description: Parity of the serial connection.

Parameter: int32_t IP_ByteSize IP_ByteSize

Direction: Down

Valid values:

SENSOR_ILR110x_115x: 7, 8
 otherwise: 8

Unit: Bit

Default:

SENSOR_ILR110x_115x: 7,
 otherwise: 8

Description: Number of data bits per byte of the serial connection.

Parameter: int32_t IP_SensorAddress IP_SensorAddress

Direction: Down

Valid for sensor:

SENSOR_DT6120, SENSOR_MEBUG

Valid values:

Minimum: 0
Maximum: 126

Default: 126

Description: If the interface is RS485 (with RS232 software emulation), up to 32 slave devices (sensors) can be connected. Therefore a sensor address is required.

If several slaves are opened with different interface parameters (e.g. baudrate), the interface at computer is adjusted to last sensor. Other sensor may no longer be accessible. The same can happen when changing sensor baudrate (and interface baudrate implicit) by MEDAQLib command.

Parameter: int32_t IP_ReadTimeout

IP_ReadTimeout

Direction: Down

Valid values:

Minimum: -2147483648 (INT32_MIN)

Maximum: 2147483647 (INT32_MAX)

Unit: ms

Default:

SENSOR_DT6120, SENSOR_MEBUS: -8,

otherwise: 8

Description: Timeout for read function at RS232 read thread.

A value of 0 means, that read thread runs as fast as possible (one core has full load). A value greater than 0 means to wait for data up to this time (low processor load). A value less than 0 means to wait up to this time (negated) but returns immediately if any byte arrives (high processor load if data arrives continuously, fast response if data arrives intermittent).

Parameter: int32_t IP_EnableToggleDTR

IP_EnableToggleDTR

Direction: Down

Valid values:

0= Off

1= On

Default:

SENSOR_DT6120, SENSOR_MEBUS: 0,

otherwise: 1

Description: Some RS422 to USB converters (like IF2001_USB) has a short-circuit between DTR and DSR line (because sensors does not support DSR line). Toggling DTR tells the converter to send all buffered data to computer.

Parameter: String IP_EventChar

IP_EventChar

Direction: Down

Valid values:

Any single character.

Default:

SENSOR_DT6120, SENSOR_MEBUS: character 0x16,

otherwise: not set

Description: Some RS422 to USB converters (containing FTDI chips) can be forced to send it's internal buffer to computer when an event character is recognized.

Parameter: int32_t CP_ThreadPriority

CP_ThreadPriority

Direction: Down

Valid values:

-15= THREAD_PRIORITY_IDLE

-2= THREAD_PRIORITY_LOWEST

-1 = THREAD_PRIORITY_BELOW_NORMAL
 0 = THREAD_PRIORITY_NORMAL
 1 = THREAD_PRIORITY_ABOVE_NORMAL
 2 = THREAD_PRIORITY_HIGHEST
 15 = THREAD_PRIORITY_TIME_CRITICAL

Default: THREAD_PRIORITY_TIME_CRITICAL

Description: Priority of the RS232 read thread. It should be as high as possible to avoid data loss.

7.2.3 IF2004

Following sensors supports this interface:

[SENSOR_ILD1220](#), [SENSOR_ILD1302](#), [SENSOR_ILD1320](#), [SENSOR_ILD1402](#), [SENSOR_ILD1420](#), [SENSOR_ILD1700](#), [SENSOR_ILD1750](#), [SENSOR_ILD1900](#), [SENSOR_ILD2200](#), [SENSOR_ILD2300](#), [SENSOR_IFD2421](#), [SENSOR_IFD2422](#), [SENSOR_IFD2445](#), [SENSOR_IFD2451](#), [SENSOR_IFD2461](#), [SENSOR_IFD2471](#), [SENSOR_ODC2500](#), [SENSOR_ODC2520](#), [SENSOR_ODC2600](#), [SENSOR_ACST000](#), [SENSOR_MFA](#), [CONTROLLER_CSP2008](#), [CONTROLLER_CBOX](#), [PCI_CARD_IF2004](#) (native).

Parameter: int32_t IP_CardInstance

IP_CardInstance

Direction: Down

Valid values:

Minimum: 0

Maximum: 15

Default: 0

Description: Instance number of the IF2004 interface card. The cards are enumerated by the OS and the only way to distinguish is the card instance number. It does not change at least there are no changes at the PCI bus.

Parameter: int32_t IP_ChannelNumber

IP_ChannelNumber

Direction: Down

Valid values:

Minimum: [PCI_CARD_IF2004](#): -1, otherwise 0

Maximum: 3

Default: [PCI_CARD_IF2004](#): 3, otherwise obligatory

Description: Channel number on IF2004 Interface card. If the Encoder on the IF2004 card should be used to store values synchronously to a sensor, the channel number 3 is reserved for it. Otherwise (-1) the FIFO cannot be used for Encoder. Sensors can be carried on each channel.

Parameter: int32_t IP_UseGate

IP_UseGate

Direction: Down

Valid values:

0 = FALSE

1 = TRUE

Default: 0

Description: The gate input of the card (5V TTL signal) can be used to lock or free the FIFO for data from sensors. This parameter affects always two channels (0+1 or 2+3) because they are on the same connector. The encoder can be locked for FIFO too.

Parameter: int32_t IP_Baudrate IP_Baudrate

Direction: Down

Valid values:

SENSOR_ILD1302, SENSOR_ILD1402, SENSOR_ILD1700, SENSOR_MFA:
115200
SENSOR_ILD2200: 691200, 1250000
SENSOR_ILD1220, SENSOR_ILD1320, SENSOR_ILD1420, SENSOR_ILD1750,
SENSOR_ILD1900, SENSOR_ILD2300, SENSOR_IFD2421, SENSOR_IFD2422,
SENSOR_IFD2445, SENSOR_IFD2451, SENSOR_IFD2461, SENSOR_IFD2471,
SENSOR_ACS7000, CONTROLLER_CSP2008, CONTROLLER_CBOX,
SENSOR_ODC2500, SENSOR_ODC2520 and SENSOR_ODC2600:
115200, 691200

Unit: Baud

Default:

SENSOR_ILD1302, SENSOR_ILD1402, SENSOR_ILD1700, SENSOR_IFD2421,
SENSOR_IFD2422, SENSOR_IFD2445, SENSOR_IFD2451, SENSOR_IFD2461,
SENSOR_IFD2471, SENSOR_ODC2520, SENSOR_ACS7000, SENSOR_MFA,
CONTROLLER_CSP2008, CONTROLLER_CBOX: 115200,
SENSOR_ILD1220, SENSOR_ILD1320, SENSOR_ILD1420, SENSOR_ILD1750,
SENSOR_ILD1900: 921600,
SENSOR_ILD2200, SENSOR_ILD2300, SENSOR_ODC2500, SENSOR_ODC2600:
691200

Description: Speed of the RS422 serial connection. Only the ODC sensors can be used with different baud rates.

Parameter: int32_t IP_SyncMasterChannel IP_SyncMasterChannel

Direction: Down

Valid values:

-1 = No synchronization master
0 = Sensor channel 1 (Base Board, Connector 1/2)
1 = Sensor channel 2 (Base Board, Connector 1/2)
2 = Sensor channel 3 (Base Board, Connector 3/4)
3 = Sensor channel 4 (Base Board, Connector 3/4)

Default: -1

Description: Channel number to synchronize with.

Synchronization is done at driver layer.

Only sensor channels can be synchronized. To synchronize IF2004 internal channels (Encoder), set the latch source for this channel to the desired master.

If synchronization is active, a value for master channel and each slave channel is buffered before all values are transferred to MEDAQLib at once. If at a slave channel arrives two values while at master channel no value is available, the older value at the slave channel is deleted.

If at master channel arrives two values while at slave channel no value is available the last value at slave channel is duplicated. But if no last value is available (there was never data on slave channel before), the oldest value at master channel and all other slaves is deleted.

This ensures that the maximum time difference between a value at master and slave channel is one value at master channel, even if the datarates of the sensors are completely different.

To ensure that synchronized values have no offset (e.g. at start or after a sensor command) it is recommended to call ClearBuffer with SP_AllDevices set to true after opening all sensors and after each sensor command.

7.2.4 IF2004_USB

Following sensors supports this interface:

[SENSOR_ILD1401](#) (for sensor ILD1402 in compatibility mode).
[SENSOR_ILD1220](#), [SENSOR_ILD1302](#), [SENSOR_ILD1320](#), [SENSOR_ILD1402](#), [SENSOR_ILD1420](#), [SENSOR_ILD1700](#), [SENSOR_ILD1750](#), [SENSOR_ILD1900](#), [SENSOR_ILD2200](#), [SENSOR_ILD2300](#), [SENSOR_ODC2500](#), [SENSOR_ODC2520](#), [SENSOR_ODC2600](#), [USB_ADAPTER_IF2004](#), [SENSOR_IFD2401](#), [SENSOR_IFD2421](#), [SENSOR_IFD2422](#), [SENSOR_IFD2431](#), [SENSOR_IFD2445](#), [SENSOR_IFD2451](#), [SENSOR_IFD2461](#), [SENSOR_IFD2471](#), [SENSOR_IMC5400](#), [SENSOR_IMC5600](#), [SENSOR_ACS7000](#), [SENSOR_ILR110x_115x](#), [SENSOR_ILR118x](#), [SENSOR_ILR1191](#), [SENSOR_ILR2250](#), [SENSOR_MFA](#), [CONTROLLER_CSP2008](#), [CONTROLLER_CBOX](#) (native).

Parameter: int32_t IP_DeviceInstance

IP_DeviceInstance

Direction: Down

Valid values:

Minimum: 0

Maximum: 255

Default: 0

Description: Instance number of the USB device. The devices are enumerated by the OS and the only way to distinguish is the device instance number. It does not change at least there are no changes at the USB bus (plug/unplug devices).

Parameter: String IP_SerialNumber

IP_SerialNumber

Direction: Down

Valid values:

Minimum: "0000001"

Maximum: "9999999"

Description: Serial number of the USB adapter (optional). If not specified (or empty), IP_Port or IP_DeviceInstance is used. Leading zeros can be dismissed.

Parameter: String IP_Port

IP_Port

Direction: Down

Valid values:

"COM1"

"COM2"

...

Description: Name of the serial interface part of USB adapter, e.g. COM1, COM2, ... (optional). If not specified, IP_DeviceInstance is used.

Parameter: int32_t IP_ChannelNumber

IP_ChannelNumber

Direction: Down

Valid values:

Condition: [USB_ADAPTER_IF2004](#)

-1= No data acquisition

4= Digital IN

Valid values:

Condition: otherwise (sensors)

0= Sensor channel 1 (Connector 1/2)

1= Sensor channel 2 (Connector 1/2)

2= Sensor channel 3 (Connector 3/4)
 3= Sensor channel 4 (Connector 3/4)

Description: Channel number on USB adapter IF2004. Channel numbers 0-3 are used for sensors. Channel number 4 selects the Digital inputs on the USB adapter IF2004 and provides their values synchronously to a sensor. Channel number -1 is valid for USB_ADAPTER_IF2004 at CreateSensorInstance only and is used for parameter setting in USB adapter IF2004 itself.

Parameter: int32_t IP_Baudrate

IP_Baudrate

Direction: Down

Valid values:

SENSOR_ILD1401: 38400
 SENSOR_ILD1302, SENSOR_ILD1402: 115200, 57600, 38400, 19200,
 9600
 SENSOR_ILD1220, SENSOR_ILD1320, SENSOR_ILD1420: 1000000,
 921600, 691200, 460800, 256000, 230400, 128000, 115200, 56000,
 19200, 9600
 SENSOR_ILD1700: 115200, 57600, 19200, 9600
 SENSOR_ILD2200: 691200, 1250000
 SENSOR_IFD2401, SENSOR_IFD2431: 460800, 230400, 115200, 57600,
 38400, 19200, 9600
 SENSOR_ILD1750, SENSOR_ILD1900, SENSOR_IMC5400, SENSOR_IMC5600:
 4000000, 3000000, 2000000, 921600, 691200, 460800, 230400,
 115200, 9600
 SENSOR_IFD2421, SENSOR_IFD2422: 4000000, 3000000, 2000000,
 921600, 691200, 460800, 230400, 115200, 38400 (from firmware
 V001.041.086), 9600
 SENSOR_ILD2300, SENSOR_IFD2445, SENSOR_IFD2451, SENSOR_IFD2461,
 SENSOR_IFD2471, SENSOR_ODC2520: 4000000, 3500000, 3000000,
 2500000, 2000000, 1500000, 921600, 691200, 460800, 230400,
 115200, 9600
 SENSOR_ODC2500, SENSOR_ODC2600: 115200, 38400, 19200, 9600
 CONTROLLER_CSP2008: 115200, 691200
 SENSOR_ILR110x_115x: 57600, 38400, 19200, 9600, 4800
 SENSOR_ILR118x: 38400, 19200, 9600, 4800, 2400
 SENSOR_ILR1191: 460800, 230400, 115200, 57600, 38400, 19200,
 9600
 SENSOR_ACS7000: 4000000, 3500000, 2000000, 1500000, 921600,
 691200, 460800, 230400, 115200, 9600
 SENSOR_ILR2250, SENSOR_MFA: 230400, 115200, 9600
 CONTROLLER_CBOX: 8000000, 4000000, 3500000, 3000000, 2500000,
 2000000, 1500000, 921600, 691200, 460800, 230400, 115200, 9600

Unit: Baud

Default:

SENSOR_ILR118x: 9600,
 SENSOR_ILD1401, 38400,
 SENSOR_ILD1302, SENSOR_ILD1402, SENSOR_ILD1700, SENSOR_IFD2401,
 SENSOR_IFD2421, SENSOR_IFD2422, SENSOR_IFD2431, SENSOR_IFD2445,
 SENSOR_IFD2451, SENSOR_IFD2461, SENSOR_IFD2471, SENSOR_ODC2520,
 SENSOR_ILR1191, SENSOR_IMC5400, SENSOR_IMC5600, SENSOR_ACS7000,
 SENSOR_ILR2250, SENSOR_MFA, CONTROLLER_CSP2008, CONTROLLER_CBOX:
 115200,

SENSOR_ILD1220, SENSOR_ILD1320, SENSOR_ILD1420, SENSOR_ILD1750,
 SENSOR_ILD1900: 921600,
 SENSOR_ILD2200, SENSOR_ILD2300, SENSOR_ODC2500, SENSOR_ODC2600:
 691200,
 SENSOR_ILR110x_115x: 57600

Description: Baudrate of the RS422 serial connection.

Parameter: int32_t IP_Parity

IP_Parity

Direction: Down

Valid values:

SENSOR_ILR110x_115x: 2= EVENPARITY
 SENSOR_ODC2500, SENSOR_ODC2600: 0= NOPARITY, 2= EVEN-PARITY
 otherwise: 0= NOPARITY

Default: SENSOR_ILR110x_115x: 2, otherwise 0

Description: Parity of the RS422 serial connection.

Parameter: int32_t IP_SyncMasterChannel

IP_SyncMasterChannel

Direction: Down

Valid values:

-1= No synchronization master
 0= Sensor channel 1 (Connector 1/2)
 1= Sensor channel 2 (Connector 1/2)
 2= Sensor channel 3 (Connector 3/4)
 3= Sensor channel 4 (Connector 3/4)
 4= Digital IN

Default: -1

Description: Channel number to synchronize with.

Synchronization is done at driver layer.

If synchronization is active, a whole data frame for master channel and each slave channel is buffered before all frames are transferred to MEDAQLib at once.

If at a slave channel arrives two whole frames while at master channel no whole frame is available, the older frame at the slave channel is deleted.

If at master channel arrives two frames while at slave channel no whole frame is available the last complete frame at slave channel is duplicated.

But if no last complete frame is available (there was never data on slave channel before), the oldest frame at master channel and all other slaves is deleted.

This ensures that the maximum time difference between a value at master and slave channel is one value at master channel, even if the datarates of the sensors are completely different.

To ensure that synchronized values have no offset (e.g. at start or after a sensor command) it is recommended to call ClearBuffer with SP_AllDevices set to true after opening all sensors and after each sensor command.

Parameter: int32_t IA_Baudrate

IA_Baudrate

Direction: Up

Unit: Baud

Description: Baudrate cannot be set to any value. So it is set to the next matching baudrate the USB adapter IF2004 supports. This real baudrate is returned in this parameter.

7.2.5 IF2008

Following sensors supports this interface:

[SENSOR_ILD1401](#) (for sensor ILD1402 in compatibility mode).
[SENSOR_ILD1220](#), [SENSOR_ILD1302](#), [SENSOR_ILD1320](#), [SENSOR_ILD1402](#), [SENSOR_ILD1420](#), [SENSOR_ILD1700](#), [SENSOR_ILD1750](#), [SENSOR_ILD1900](#), [SENSOR_ILD2200](#), [SENSOR_ILD2300](#), [SENSOR_ODC2500](#), [SENSOR_ODC2520](#), [SENSOR_ODC2600](#), [PCI_CARD_IF2008](#), [SENSOR_IFD2401](#), [SENSOR_IFD2421](#), [SENSOR_IFD2422](#), [SENSOR_IFD2431](#), [SENSOR_IFD2445](#), [SENSOR_IFD2451](#), [SENSOR_IFD2461](#), [SENSOR_IFD2471](#), [SENSOR_IMC5400](#), [SENSOR_IMC5600](#), [SENSOR_AC57000](#), [SENSOR_ILR110x_115x](#), [SENSOR_ILR118x](#), [SENSOR_ILR1191](#), [SENSOR_ILR2250](#), [SENSOR_MFA](#), [CONTROLLER_CSP2008](#), [CONTROLLER_CBOX](#) (native).

Parameter: int32_t IP_CardInstance

IP_CardInstance

Direction: Down

Valid values:

Minimum: 0

Maximum: 15

Default: 0

Description: Instance number of the IF2008 interface card. The cards are enumerated by the OS and the only way to distinguish is the card instance number. It does not change at least there are no changes at the PCI(e) bus.

Parameter: int32_t IP_ChannelNumber

IP_ChannelNumber

Direction: Down

Valid values:

Condition: [PCI_CARD_IF2008](#)

-1= No data acquisition, i.e. for configuration only

6= Encoder 1

7= Encoder 2

8= Digital IN

9= Digital RxD

10= ADC 1 (Analog/Digital converter)

11= ADC 2 (Analog/Digital converter)

Valid values:

Condition: otherwise (sensors)

0= Sensor channel 1 (Base Board, Connector X1)

1= Sensor channel 2 (Base Board, Connector X1)

2= Sensor channel 3 (Base Board, Connector X2)

3= Sensor channel 4 (Base Board, Connector X2)

4= Sensor channel 5 (Extension Board, Connector X1)

5= Sensor channel 6 (Extension Board, Connector X1)

Description: Channel number on IF2008 Interface card.

Attention! Sensor channel 5 and 6 are only available if IF2008E extension card is installed. Digital IN is only available if IF2008E extension card or IF2008IO extension slot is installed. ADC is only available if IF2008E extension card is installed.

-1 means, no data channel is written to FIFO and cannot be read using [TransferData](#) or [Poll](#). This mode can be used if IF2008 should only be parametrized.

Parameter: int32_t IP_Baudrate IP_Baudrate

Direction: Down

Valid values:

SENSOR_ILD1401: 38400
 SENSOR_ILD1302, SENSOR_ILD1402: 115200, 57600, 38400, 19200,
 9600
 SENSOR_ILD1220, SENSOR_ILD1320, SENSOR_ILD1420: 1000000,
 921600, 691200, 460800, 256000, 230400, 128000, 115200, 56000,
 19200, 9600
 SENSOR_ILD1700: 115200, 57600, 19200, 9600
 SENSOR_ILD2200: 691200, 1250000
 SENSOR_IFD2401, SENSOR_IFD2431: 460800, 230400, 115200, 57600,
 38400, 19200, 9600
 SENSOR_ILD1750, SENSOR_ILD1900, SENSOR_IMC5400, SENSOR_IMC5600:
 4000000, 3000000, 2000000, 921600, 691200, 460800, 230400,
 115200, 9600
 SENSOR_IFD2421, SENSOR_IFD2422: 4000000, 3000000, 2000000,
 921600, 691200, 460800, 230400, 115200, 38400 (from firmware
 V001.041.086), 9600
 SENSOR_ILD2300, SENSOR_IFD2445, SENSOR_IFD2451, SENSOR_IFD2461,
 SENSOR_IFD2471, SENSOR_ODC2520: 4000000, 2500000, 2000000,
 1500000, 921600, 691200, 460800, 230400, 115200, 9600
 SENSOR_ODC2500, SENSOR_ODC2600: 115200, 38400, 19200, 9600
 CONTROLLER_CSP2008: 115200, 691200
 SENSOR_ILR110x_115x: 57600, 38400, 19200, 9600, 4800
 SENSOR_ILR118x: 38400, 19200, 9600, 4800, 2400
 SENSOR_ILR1191: 460800, 230400, 115200, 57600, 38400, 19200,
 9600
 SENSOR_ACS7000: 4000000, 2000000, 1500000, 921600, 691200,
 460800, 230400, 115200, 9600
 SENSOR_ILR2250, SENSOR_MFA: 230400, 115200, 9600
 CONTROLLER_CBOX: 8000000, 4000000, 3500000, 3000000, 2500000,
 2000000, 1500000, 921600, 691200, 460800, 230400, 115200, 9600

Unit: Baud

Default:

SENSOR_ILR118x: 9600,
 SENSOR_ILD1401, 38400,
 SENSOR_ILD1302, SENSOR_ILD1402, SENSOR_ILD1700, SENSOR_IFD2401,
 SENSOR_IFD2421, SENSOR_IFD2422, SENSOR_IFD2431, SENSOR_IFD2445,
 SENSOR_IFD2451, SENSOR_IFD2461, SENSOR_IFD2471, SENSOR_ODC2520,
 SENSOR_ILR1191, SENSOR_IMC5400, SENSOR_IMC5600, SENSOR_ACS7000,
 SENSOR_ILR2250, SENSOR_MFA, CONTROLLER_CSP2008, CONTROLLER_CBOX:
 115200,
 SENSOR_ILD1220, SENSOR_ILD1320, SENSOR_ILD1420, SENSOR_ILD1750,
 SENSOR_ILD1900: 921600,
 SENSOR_ILD2200, SENSOR_ILD2300, SENSOR_ODC2500, SENSOR_ODC2600:
 691200,
 SENSOR_ILR110x_115x: 38400

Description: Speed of the RS422 serial connection.

Parameter: int32_t IP_Parity IP_Parity

Direction: Down

Valid values:

SENSOR_ILR110x_115x: 2= EVENPARITY
SENSOR_ODC2500, SENSOR_ODC2600: 0= NOPARITY, 2= EVEN-PARITY
 otherwise: 0= NOPARITY

Default: `SENSOR_ILR110x_115x`: 2, otherwise 0

Description: Parity of the RS422 serial connection.

Parameter: `int32_t IP_SyncMasterChannel`

`IP_SyncMasterChannel`

Direction: Down

Valid values:

- 1= No synchronization master
- 0= Sensor channel 1 (Base Board, Connector X1)
- 1= Sensor channel 2 (Base Board, Connector X1)
- 2= Sensor channel 3 (Base Board, Connector X2)
- 3= Sensor channel 4 (Base Board, Connector X2)
- 4= Sensor channel 5 (Extension Board, Connector X1)
- 5= Sensor channel 6 (Extension Board, Connector X1)
- 6= Encoder 1
- 7= Encoder 2
- 8= Digital IN
- 9= RxD
- 10= ADC 1
- 11= ADC 2

Default: -1

Description: Channel number to synchronize with.

Synchronization is done at driver layer and is only supported for IF2008 cards with FPGA version 4 or higher.

Only sensor channels can be synchronized. To synchronize IF2008 internal channels (Encoder's, Digital IN, RxD or ADC's), set the latch source for this channel to the desired master.

If synchronization is active, a whole data frame for master channel and each slave channel is buffered before all frames are transferred to MEDAQLib at once.

If at a slave channel arrives two whole frames while at master channel no whole frame is available, the older frame at the slave channel is deleted.

If at master channel arrives two frames while at slave channel no whole frame is available the last complete frame at slave channel is duplicated.

But if no last complete frame is available (there was never data on slave channel before), the oldest frame at master channel and all other slaves is deleted.

This ensures that the maximum time difference between a value at master and slave channel is one value at master channel, even if the datarates of the sensors are completely different.

To ensure that synchronized values have no offset (e.g. at start or after a sensor command) it is recommended to call ClearBuffer with `SP_AllDevices` set to true after opening all sensors and after each sensor command.

Parameter: `int32_t IA_Baudrate`

`IA_Baudrate`

Direction: Up

Unit: Baud

Description: Baudrate cannot be set to any value. So it is set to the next matching baudrate the IF2008 card supports. This real baudrate is returned in this parameter.

7.2.6 IF2008_ETH

Following sensors supports this interface:

[SENSOR_ILD1401](#) (for sensor ILD1402 in compatibility mode).
[SENSOR_ILD1220](#), [SENSOR_ILD1302](#), [SENSOR_ILD1320](#), [SENSOR_ILD1402](#), [SENSOR_ILD1420](#), [SENSOR_ILD1700](#), [SENSOR_ILD1750](#), [SENSOR_ILD1900](#), [SENSOR_ILD2200](#), [SENSOR_ILD2300](#), [SENSOR_ODC2500](#), [SENSOR_ODC2520](#), [SENSOR_ODC2600](#), [ETH_ADAPTER_IF2008](#), [SENSOR_IFD2401](#), [SENSOR_IFD2421](#), [SENSOR_IFD2422](#), [SENSOR_IFD2431](#), [SENSOR_IFD2445](#), [SENSOR_IFD2451](#), [SENSOR_IFD2461](#), [SENSOR_IFD2471](#), [SENSOR_IMC5400](#), [SENSOR_IMC5600](#), [SENSOR_ACS7000](#), [SENSOR_ILR110x_115x](#), [SENSOR_ILR118x](#), [SENSOR_ILR1191](#), [SENSOR_ILR2250](#), [SENSOR_MFA](#).

Parameter: String IP_RemoteAddr IP_RemoteAddr

Direction: Down

Valid value: IP address

Description: IP address of the IF2008_Ethernet interface (TCP server). It has to be set in any case because the default address is not valid!

Parameter: int32_t IP_ChannelNumber IP_ChannelNumber

Direction: Down

Valid values:

Condition: [IF2008_ETH](#)

- 1= No data acquisition
- 0= Encoder channel 1
- 1= Encoder channel 2
- 2= Encoder channel 3
- 3= Encoder channel 4
- 4= Encoder channel 5
- 5= Encoder channel 6
- 6= Encoder channel 7
- 7= Encoder channel 8
- 8= Digital IN

Valid values:

Condition: otherwise (sensors)

- 0= Sensor channel 1
- 1= Sensor channel 2
- 2= Sensor channel 3
- 3= Sensor channel 4
- 4= Sensor channel 5
- 5= Sensor channel 6
- 6= Sensor channel 7
- 7= Sensor channel 8

Description: Channel number on IF2008_Ethernet Interface.

-1 means, no data channel is written to FIFO and cannot be read using [TransferData](#) or [Poll](#). This mode can be used if IF2008_Ethernet should only be parametrized.

Parameter: int32_t IP_Baudrate IP_Baudrate

Direction: Down

Valid values:

[SENSOR_ILD1401](#): 38400

SENSOR_ILD1302, SENSOR_ILD1402: 115200, 57600, 38400, 19200,
 9600
SENSOR_ILD1220, SENSOR_ILD1320, SENSOR_ILD1420: 1000000,
 921600, 691200, 460800, 256000, 230400, 128000, 115200, 56000,
 19200, 9600
SENSOR_ILD1700: 115200, 57600, 19200, 9600
SENSOR_ILD2200: 691200, 1250000
SENSOR_IFD2401, SENSOR_IFD2431: 460800, 230400, 115200, 57600,
 38400, 19200, 9600
SENSOR_ILD1750, SENSOR_ILD1900, SENSOR_IMC5400, SENSOR_IMC5600:
 4000000, 3000000, 2000000, 921600, 691200, 460800, 230400,
 115200, 9600
SENSOR_IFD2421, SENSOR_IFD2422: 4000000, 3000000, 2000000,
 921600, 691200, 460800, 230400, 115200, 38400 (from firmware
 V001.041.086), 9600
SENSOR_ILD2300, SENSOR_IFD2445, SENSOR_IFD2451, SENSOR_IFD2461,
SENSOR_IFD2471, SENSOR_ODC2520: 4000000, 2500000, 2000000,
 1500000, 921600, 691200, 460800, 230400, 115200, 9600
SENSOR_ODC2500, SENSOR_ODC2600: 115200, 38400, 19200, 9600
SENSOR_ILR110x_115x: 57600, 38400, 19200, 9600, 4800
SENSOR_ILR118x: 38400, 19200, 9600, 4800, 2400
SENSOR_ILR1191: 460800, 230400, 115200, 57600, 38400, 19200,
 9600
SENSOR_ACS7000: 4000000, 2000000, 1500000, 921600, 691200,
 460800, 230400, 115200, 9600
SENSOR_ILR2250, SENSOR_MFA: 230400, 115200, 9600

Unit: Baud

Default:

SENSOR_ILR118x: 9600,
SENSOR_ILD1401: 38400,
SENSOR_ILD1302, SENSOR_ILD1402, SENSOR_ILD1700, SENSOR_IFD2401,
SENSOR_IFD2421, SENSOR_IFD2422, SENSOR_IFD2431, SENSOR_IFD2445,
SENSOR_IFD2451, SENSOR_IFD2461, SENSOR_IFD2471, SENSOR_ODC2520,
SENSOR_ILR1191, SENSOR_IMC5400, SENSOR_IMC5600, SENSOR_ACS7000,
SENSOR_ILR2250, SENSOR_MFA: 115200,
SENSOR_ILD1220, SENSOR_ILD1320, SENSOR_ILD1420, SENSOR_ILD1750,
SENSOR_ILD1900: 921600,
SENSOR_ILD2200, SENSOR_ILD2300, SENSOR_ODC2500, SENSOR_ODC2600:
 691200,
SENSOR_ILR110x_115x: 38400

Description: Speed of the RS422 serial connection from IF2008_Ethernet to sensor.

Parameter: int32_t IP_SyncMasterChannel

IP_SyncMasterChannel

Direction: Down

Valid values:

- 1 = No synchronization master
- 0 = Sensor/Encoder channel 1
- 1 = Sensor/Encoder channel 2
- 2 = Sensor/Encoder channel 3
- 3 = Sensor/Encoder channel 4
- 4 = Sensor/Encoder channel 5
- 5 = Sensor/Encoder channel 6

6= Sensor/Encoder channel 7
 7= Sensor/Encoder channel 8
 8= Digital IN

Default: -1

Description: Channel number to synchronize with.

Only sensor channels can be synchronized. To synchronize IF2008_Ethernet internal channels (Encoder's or Digital IN), set the latch source for this channel to the desired master.

If synchronization is active, a whole data frame for master channel and each slave channel is buffered before all frames are transferred to MEDAQLib at once.

If at a slave channel arrives two whole frames while at master channel no whole frame is available, the older frame at the slave channel is deleted. If at master channel arrives two frames while at slave channel no whole frame is available the last complete frame at slave channel is duplicated. But if no last complete frame is available (there was never data on slave channel before), the oldest frame at master channel and all other slaves is deleted.

This ensures that the maximum time difference between a value at master and slave channel is one value at master channel, even is the datarates of the sensors are completely different.

To ensure that synchronized values has no offset (e.g. at start or after a sensor command) it is recommended to call ClearBuffer with SP_AllDevices set to true after opening all sensors and after each sensor command.

7.2.7 TCP/IP

Following sensors supports this interface:

[SENSOR_ILD1220](#), [SENSOR_ILD1302](#), [SENSOR_ILD1320](#), [SENSOR_ILD1401](#), [SENSOR_ILD1402](#), [SENSOR_ILD1420](#), [SENSOR_ILD1700](#), [SENSOR_ILD1750](#), [SENSOR_ILD1900](#), [SENSOR_ILD2200](#), [SENSOR_ODC1202](#), [SENSOR_ODC2500](#), [SENSOR_ODC2600](#), [SENSOR_IFD2401](#), [SENSOR_IFD2431](#), [SENSOR_ILR110x_115x](#), [SENSOR_ILR118x](#), [SENSOR_ILR1191](#), [SENSOR_ILR2250](#), [SENSOR_MFA](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface).
[SENSOR_ILD2300](#), [SENSOR_IFD2421](#), [SENSOR_IFD2422](#), [SENSOR_IFD2445](#), [SENSOR_IFD2451](#), [SENSOR_IFD2461](#), [SENSOR_IFD2471](#), [SENSOR_ODC2520](#), [SENSOR_DT3060](#), [SENSOR_DT3100](#), [SENSOR_DT6100](#), [SENSOR_IMC5400](#), [SENSOR_IMC5600](#), [SENSOR_ACST000](#), [SENSOR_CFO](#), [CONTROLLER_KSS6380](#), [CONTROLLER_DT6200](#), [CONTROLLER_KSS64xx](#), [CONTROLLER_DT6500](#), [CONTROLLER_DT6536](#), [CONTROLLER_CSP2008](#), [CONTROLLER_CBOX](#), [THICKNESS_SENSOR](#), [ETH_IF1032](#) (native).

Parameter: String IP_RemoteAddr

IP_RemoteAddr

Direction: Down

Valid value: IP address

Description: IP address of the remote sensor (TCP server). It has to be set in any case because the default address is not valid!

Parameter: int32_t IP_RemotePort IP_RemotePort

Direction: Down

Valid values:

Minimum: 1

Maximum: 65535

Default: SENSOR_ILD2300, SENSOR_IFD2421, SENSOR_IFD2422, SENSOR_IFD2445, SENSOR_IFD2451, SENSOR_IFD2461, SENSOR_IFD2471, SENSOR_ODC2520, SENSOR_IMC5400, SENSOR_IMC5600, SENSOR_ACS7000, CONTROLLER_CSP2008, CONTROLLER_CBOX, THICKNESS_SENSOR: 23 (Telnet), SENSOR_CFO: 80, otherwise 10001

Description: TCP port of the remote sensor (server). For SENSOR_ILD2300, SENSOR_IFD2421, SENSOR_IFD2422, SENSOR_IFD2445, SENSOR_IFD2451, SENSOR_IFD2461, SENSOR_IFD2471, SENSOR_ODC2520, SENSOR_IMC5400, SENSOR_IMC5600, SENSOR_ACS7000, CONTROLLER_CSP2008, CONTROLLER_CBOX and THICKNESS_SENSOR this parameter is used for command port. The command port 23 (Telnet) is fix. Both interfaces are generated by the driver.

Parameter: int32_t IP_AddressFamily IP_AddressFamily

Direction: Down

Valid values:

2= AF_INET (IPv4)

23= AF_INET6 (IPv6)

Default: Automatically detected depending on sensor address (AF_INET6 on a IPv6 address, otherwise AF_INET).

Description: TCP/IP address family, on which the sensor is accessible.

Parameter: int32_t IP_RemoteDataProtocol IP_RemoteDataProtocol

Direction: Down

Valid for sensor:

SENSOR_ILD2300

SENSOR_IFD2421

SENSOR_IFD2422

SENSOR_IFD2445

SENSOR_IFD2451

SENSOR_IFD2461

SENSOR_IFD2471

SENSOR_ODC2520

CONTROLLER_CSP2008

CONTROLLER_CBOX

THICKNESS_SENSOR

SENSOR_IMC5400

SENSOR_IMC5600

SENSOR_ACS7000

Valid values:

-1= Query automatic (not for CONTROLLER_CSP2008)

0= TCP server

1= TCP client (not for CONTROLLER_CBOX and THICKNESS_SENSOR)

2= UDP sender (not for CONTROLLER_CBOX and THICKNESS_SENSOR)

Default: CONTROLLER_CSP2008: 0, otherwise -1

Description: IP Protocol used for data transfer. Local host automatically takes opposite.

Parameter: int32_t IP_PortDetection IP_PortDetection

Direction: Down

Valid for sensor:

- SENSOR_DT3060
- CONTROLLER_DT6200
- CONTROLLER_KSS64xx
- CONTROLLER_DT6500
- CONTROLLER_DT6536
- ETH_IF1032

Valid values:

- 0= Set port(s) manually ([SENSOR_DT3060](#), [CONTROLLER_DT6200](#),
[CONTROLLER_KSS64xx](#), [CONTROLLER_DT6500](#), [CONTROLLER_DT6536](#)
+ [ETH_IF1032](#))
- 1= Query all ports automatically (only [CONTROLLER_DT6500](#))
- 2= Query data port automatically ([SENSOR_DT3060](#), [CONTROLLER_DT6200](#),
[CONTROLLER_KSS64xx](#), [CONTROLLER_DT6536](#) + [ETH_IF1032](#))

Default: [SENSOR_DT3060](#), [CONTROLLER_DT6200](#), [CONTROLLER_KSS64xx](#),
[CONTROLLER_DT6536](#) + [ETH_IF1032](#): 2, [CONTROLLER_DT6500](#): 1

Description: [CONTROLLER_DT6500](#) exists in two variations. The first one has one port for commands and data (always 10001), the new one has a command port (23, Telnet) and a data port (default 10001). If ports should be set manually, for first version both ports ([IP_RemotePort](#) and [IP_DataPort](#)) must be set to the same value (port at sensor). For the new version, [IP_RemotePort](#) must be set to port 23 and [IP_DataPort](#) must be set to data port at sensor. If ports should be queried automatically, [IP_RemotePort](#) and [IP_DataPort](#) is ignored. In this case, MEDAQLib first tries to open port 23. If successful, data port is queried and opened. If not, port 10001 used as command and data port.

At [SENSOR_DT3060](#), [CONTROLLER_DT6200](#), [CONTROLLER_KSS64xx](#),
[CONTROLLER_DT6536](#) and [ETH_IF1032](#), command port is always 23
(Telnet) and data port can be set manually or automatically queried.

Parameter: int32_t IP_DataPort IP_DataPort

Direction: Down

Valid for sensor:

- SENSOR_ILD2300
- SENSOR_IFD2421
- SENSOR_IFD2422
- SENSOR_IFD2445
- SENSOR_IFD2451
- SENSOR_IFD2461
- SENSOR_IFD2471
- SENSOR_ODC2520
- SENSOR_DT3060
- CONTROLLER_DT6200
- CONTROLLER_KSS64xx
- CONTROLLER_DT6500
- CONTROLLER_DT6536
- CONTROLLER_CSP2008
- CONTROLLER_CBOX
- THICKNESS_SENSOR
- ETH_IF1032
- SENSOR_IMC5400

[SENSOR_IMC5600](#)
[SENSOR_ACS7000](#)

Valid values:

Minimum: 0

Maximum: [CONTROLLER_CSP2008](#): 32384, otherwise 65535

Default: [CONTROLLER_DT6500](#): 10001, otherwise 1024

Description: If remote data protocol is TCP server this is the remote data port, otherwise it is the local data port. If remote data protocol is set to query automatic, this parameter is ignored. If this parameter is 0, the data port is not opened.

7.2.8 WinUSB

Following sensors supports this interface:

[SENSOR_IFD2401](#), [SENSOR_IFD2431](#) and [CONTROLLER_CBOX](#) (native).

Parameter: int32_t IP_DeviceInstance

IP_DeviceInstance

Direction: Down

Valid values:

Minimum: 0

Maximum: 255

Default: 0

Description: Instance number of the USB device. The devices are enumerated by the OS and the only way to distinguish is the device instance number. It does not change at least there are no changes at the USB bus (plug/unplug devices).

Parameter: int32_t IP_UsbReadBufCnt

IP_UsbReadBufCnt

Direction: Down

Valid values:

Minimum: 2

Maximum: 32

Default: 8

Description: Number of buffers for read operations on USB.

Parameter: int32_t IP_UsbReadBufSize

IP_UsbReadBufSize

Direction: Down

Valid values:

Minimum: 2

Maximum: 1048576

Unit: Bytes

Default: otherwise 512

Description: Buffer size for read operations on USB. The value is always ceiled to the next even number (2, 4, 6, ..., 1048574, 1048576).

Parameter: int32_t CP_ThreadPriority

CP_ThreadPriority

Direction: Down

Valid values:

-15= THREAD_PRIORITY_IDLE

-2= THREAD_PRIORITY_LOWEST

-1= THREAD_PRIORITY_BELOW_NORMAL

0= THREAD_PRIORITY_NORMAL

1= THREAD_PRIORITY_ABOVE_NORMAL

2= THREAD_PRIORITY_HIGHEST
15= THREAD_PRIORITY_TIME_CRITICAL

Default: THREAD_PRIORITY_TIME_CRITICAL

Description: Priority of the USB read thread. It should be as high as possible to avoid data loss.

7.3 Sensor parameters

The available sensor commands and its parameters are described at the specific sensor sections.

8 Sensor commands

8.1 Communication via SensorCommand

The function [SensorCommand](#) always must have one obligatory parameter:

Parameter: String S_Command

S_Command

Direction: Down

Valid for sensor: all

Valid value: See commands for sensors.

Description: The command to execute is specified by this parameter. Some commands are for the driver, the most commands are processed by the sensor. Note that within this documentation the sensor commands are listed in the form "MEDAQLib-Command (ASCII-Command)". S_Command must be the "MEDAQLib-Command".

The following parameters are used for each call to [SensorCommand](#) so they are described only once:

Parameter: int32_t CP_SensorAnswerTimeout

CP_SensorAnswerTimeout

Direction: Down

Valid for sensor: all

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Unit: ms

Default: Depending on sensor and command.

Description: This special timeout is added to the global timeout (specified by [CP_BaseLevelTimeout](#)) when waiting for the complete answer from sensor.

Parameter: String SA_CompleteAnswer

SA_CompleteAnswer

Direction: Up

Valid for sensor: all

Description: The raw (not interpreted) answer (to a command) from the sensor is stored here.

Parameter: int32_t SA_ErrorNumber

SA_ErrorNumber

Direction: Up

Valid for sensor:

[SENSOR_ILR2250](#)

[SENSOR_ILD1220](#)

[SENSOR_ILD1302](#)

[SENSOR_ILD1320](#)

[SENSOR_ILD1401](#)

[SENSOR_ILD1402](#)

[SENSOR_ILD1420](#)

[SENSOR_ILD1700](#)

[SENSOR_ILD1750](#)

[SENSOR_ILD1900](#)

[SENSOR_ILD2200](#)

[SENSOR_ILD2300](#)

[SENSOR_IFD2421](#)

SENSOR_IFD2422
 SENSOR_IFD2445
 SENSOR_IFD2451
 SENSOR_IFD2461
 SENSOR_IFD2471
 SENSOR_ODC2500
 SENSOR_ODC2520
 SENSOR_ODC2600
 SENSOR_IMC5400
 SENSOR_IMC5600
 SENSOR_ACS7000
 SENSOR_CFO
 SENSOR_MFA
 ETH_ADAPTER_IF2008
 CONTROLLER_CSP2008
 CONTROLLER_CBOX
 THICKNESS_SENSOR

Valid values:

Minimum: 0

Maximum: 255

Description: Error number returned by any sensor if a command was not successful.

Parameter: String SA_ErrorText

SA_ErrorText

Direction: Up

Valid for sensor:

SENSOR_ILR118x
 SENSOR_ILR1191
 SENSOR_ILR2250
 SENSOR_ILD1220
 SENSOR_ILD1302
 SENSOR_ILD1320
 SENSOR_ILD1402
 SENSOR_ILD1420
 SENSOR_ILD1700
 SENSOR_ILD1750
 SENSOR_ILD1900
 SENSOR_ILD2200
 SENSOR_ILD2300
 SENSOR_IFD2401
 SENSOR_IFD2421
 SENSOR_IFD2422
 SENSOR_IFD2431
 SENSOR_IFD2445
 SENSOR_IFD2451
 SENSOR_IFD2461
 SENSOR_IFD2471
 SENSOR_ODC2500
 SENSOR_ODC2520
 SENSOR_ODC2600
 SENSOR_DT3060
 SENSOR_DT3100
 SENSOR_DT6100

```

SENSOR_DT6120
SENSOR_MEBUS
CONTROLLER_DT6200
CONTROLLER_KSS64xx
CONTROLLER_DT6500
CONTROLLER_DT6536
ETH_ADAPTER_IF2008
CONTROLLER_CSP2008
CONTROLLER_CBOX
THICKNESS_SENSOR
SENSOR_IMC5400
SENSOR_IMC5600
SENSOR_ACS7000
SENSOR_CFO
SENSOR_MFA
ETH_IF1032

```

Description: Clear text for specific error number.

Following parameters affects the driver and interface (when communicating with the sensor):

Parameter: int32_t IP_ClearRingBuffer

IP_ClearRingBuffer

Direction: Down

Valid Interface: all

Valid for sensor: all

Valid values:

0= FALSE

1= TRUE

Default: 1

Description: Clears the ring buffer before sending the command to the sensor.

The containing data is discarded. So the next data in the ring buffer is the sensor answer. For **IF2004** card, the ring buffer is cleared after reading the answer too.

Parameter: int32_t IP_ClearSendBuffer

IP_ClearSendBuffer

Direction: Down

Valid Interface: all

Valid for sensor: all

Valid values:

0= FALSE

1= TRUE

Default: 1

Description: Clears the send buffer (PurgeComm (PURGE_TXCLEAR)) before sending the command to the sensor.

Parameter: int32_t IP_ClearReceiveBuffer

IP_ClearReceiveBuffer

Direction: Down

Valid Interface:

RS232

IF2004

IF2004_USB

IF2008

Valid for sensor: all

Valid values:

0= FALSE
1= TRUE

Default: 1

Description: Clears the receive buffer (PurgeComm (PURGE_RXCLEAR) for RS232, reading the FIFO for IF2004, IF2004_USB and IF2008) before sending the command to the sensor.

This example shows how to set measure speed of [SENSOR_ILD1700](#).

```
SetParameterString (instance, "S_Command", "Set_Speed");
SetParameterInt (instance, "SP_Speed", 1); /* 1 is 1.25 kHz */
err= SensorCommand (instance);
/* error handling, if err!=ERR_NOERROR */
```

Alternatively you can use the wrapper function [SetIntExecSCmd](#).

```
err= SetIntExecSCmd (instance, "Set_Speed", "SP_Speed", 1);
/* error handling, if err!=ERR_NOERROR */
```

The next example shows how to get information from [SENSOR_IFD2401](#).

```
int32_t sensor;
SetParameterString (instance, "S_Command", "Get_Status");
err= SensorCommand (instance);
/* error handling, if err!=ERR_NOERROR */
GetParameterInt (instance, "SA_Sensor", &sensor);
```

Or you can use the wrapper function [ExecSCmdGetInt](#).

```
int32_t sensor;
err= ExecSCmdGetInt (instance, "Get_Status", "SA_Sensor", &sensor);
/* error handling, if err!=ERR_NOERROR */
```

8.2 Sensor commands valid for each sensor

Commands are stored in Parameter [S_Command](#). Following chapters describe commands and the parameters for each command.

8.2.1 Get_SensorProperties

This command retrieve static information about capabilities of created sensor.
Can be called before sensor is opened, too.

Parameter: String CA_SupportedInterfaces

CA_SupportedInterfaces

Direction: Up

Valid values:

- RS232
- RS422
- RS485
- TCPIP
- USB

Description: List (newline separated) of interfaces supported by sensor.

Parameter: int32_t CA_SupportedInterfaceRS232	CA_SupportedInterfaceRS232
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Returns true, if sensors supports RS232.	
Parameter: int32_t CA_SupportedInterfaceRS422	CA_SupportedInterfaceRS422
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Returns true, if sensors supports RS422.	
Parameter: int32_t CA_SupportedInterfaceRS485	CA_SupportedInterfaceRS485
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Returns true, if sensors supports RS485.	
Parameter: int32_t CA_SupportedInterfaceTCPIP	CA_SupportedInterfaceTCPIP
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Returns true, if sensors supports TCP/IP.	
Parameter: int32_t CA_SupportedInterfaceUSB	CA_SupportedInterfaceUSB
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Returns true, if sensors supports USB.	

8.2.2 Get_OpenState

This command retrieve information from MEDAQLib about open interfaces to sensor.

Parameter: int32_t IA_SensorOpen	IA_SensorOpen
Direction: Up	
Valid values:	
0= FALSE	
1= TRUE	
Description: Connection to sensor is established. This does not mean, communication to sensor is possible (e.g. if wrong baudrate is selected).	

8.2.3 Enable_Logging

This command can control logging to file for debugging purposes at any time.

Parameter: int32_t IP_EnableLogging IP_EnableLogging

Direction: Down

Valid values:

0= FALSE

1= TRUE

Default: 0

Description: This parameter enables or disables logging to file for debugging purposes.

Parameter: int32_t IP_LogType IP_LogType

Direction: Down

Valid values:

A bit combination of following values:

1= HIGH_TYPE (User <--> MEDAQLib)

2= MIDDLE_TYPE (Sensor layer <--> Interface layer)

4= LOW_TYPE (MEDAQLib <--> Hardware driver)

8= ERROR_TYPE (Any errors reported by MEDAQLib)

16= DRIVER_TYPE (Hardware driver <--> Sys driver)

32= APPL_TYPE (Application specific, see [LogFile](#) and [LogFileU](#))

2147483647= ALL (all log types are active)

Default: 2147483647 (INT32_MAX)

Description: This parameter specifies the type of messages to log.

Parameter: int32_t IP_LogLevel IP_LogLevel

Direction: Down

Valid values:

A bit combination of following values:

1= EMERGENCY_LEVEL (logging emerging errors)

2= CRITICAL_LEVEL (logging critical errors)

4= ERROR_LEVEL (logging errors which occurs)

8= WARNING_LEVEL (logging warnings from MEDAQLib)

16= NOTICE_LEVEL (logging notices)

32= TRACE_LEVEL (logging function calls)

64= DATA_LEVEL (logging data in binary mode)

2147483647= ALL (all levels are active)

Default: 2147483647 (INT32_MAX)

Description: This parameter specifies the kind of event to log.

Parameter: String IP_LogFile IP_LogFile

Direction: Down

Default:

Description: File name of log file.

If it is empty or ends with '\' or '/', an automatic generated name ('SensorLog_%yyyy-%MM-%dd_%hh-%mm-%ss.%Us.txt') is appended.

Many placeholders (...) can be used for automatic name generation.

Important ones are:

%h= hour in 24 hours format (0, 1, ..., 9, 10, ..., 23)

%hh= hour in 24 hours format (00, 01, ..., 09, 10, ..., 23)

%H= hour in 12 hours format (1, 2, ..., 9, 10, 11, 12)

%HH= hour in 12 hours format (01, 02, ..., 09, 10, 11, 12)

```
%m= minute (0, 1, ..., 9, 10, ..., 59)
%mm= minute (00, 01, ..., 09, 10, ..., 59)
%s= second (0, 1, ..., 9, 10, ..., 59)
%ss= second (00, 01, ..., 09, 10, ..., 59)
%Ms= millisecond (000, 001, ..., 999)
%Us= microsecond (000000, 000001, ..., 999999)
%PP= output for morning, afternoon for US-American time format (AM,
PM)
%d= day (1, 2, ..., 9, 10, ..., 31)
%dd= day (01, 02, ..., 09, 10, ..., 31)
%M= month (1, 2, ..., 9, 10, 11, 12)
%MM= month (01, 02, ..., 09, 10, 11, 12)
%yy= year (70, 71, ..., 99, 00, 01, ..., 38)
%yyyy= year (1970, 1971, ..., 1999, 2000, 2001, ..., 2038)
%DoW= day of week (0= Sunday, 1= Monday, ..., 6= Saturday)
%DoY= day of year (0, 1, ..., 9, 10, ..., 364, if a leap year 365)
%WoY= week of year (0, 1, ..., 9, 10, ..., 52), starting with 0= the week
with the first Sunday in year (US format)
%[NBR]= generates a consecutive number which increments with each
new logfile up to the number NBR
```

Parameter: int32_t IP_LogAppend IP_LogAppend

Direction: Down

Valid values:

0= FALSE
1= TRUE

Default: 1 if %[NBR] is specified within IP_LogFile, otherwise 0

Description: This parameter specifies if the logfile should be cleared at opening or if the new data should be appended to file. It has only an effect if a new log file is opened.

Parameter: int32_t IP_LogFlush IP_LogFlush

Direction: Down

Valid values:

0= FALSE
1= TRUE

Default: 0

Description: This parameter specifies if the logfile should be flushed after each output. In this case, it is sure that all information is stored to logfile before proceeding. But depending on the storage device it can slow down the MEDAQLib.

Parameter: int32_t IP_LogSplitSize IP_LogSplitSize

Direction: [IN]

Valid values:

Minimum: 0
Maximum: 2147483647 (INT32_MAX)

Unit: KB (1024 Bytes)

Default: 0

Description: If this parameter is greater than 0, logfile is closed and reopened when this size is reached. If the file name contains placeholders (%...), a new name is generated before opening. Otherwise, the same file is opened again and if appending is off, the old content is overwritten.

8.2.4 Automatic_Setup

This command has two purposes.

On the one hand it setups MEDAQLib by retrieving required information from sensor.
On the other hand, it setups the sensor so it sends measurement data back to MEDAQLib.

Additionally this command modifies parameter [IP_AutomaticMode](#). So if it is set to 0, automatic setup after sensor command is disabled. Otherwise it is bitwise ORed.

Parameter: int32_t CP_AutomaticMode

CP_AutomaticMode

Direction: Down

Valid values:

A bit combination of following values:

"First bit (1)= Retrieve sensor information to setup MEDAQLib"

"Second bit (2)= Activate data output at sensor"

Default: 1

Description: First bit (1) allows MEDAQLib to retrieve information from sensor automatically if needed.

Second bit (2) allows MEDAQLib to change sensor interface parameters (if needed) so is outputs data.

8.2.5 Get_TransmittedDataInfo

Many sensors can transmit more than one value at each measurement (e.g. Distance + Intensity). All transmitted values are enclosed in a data frame. [TransferData](#) (and also [Poll](#)) returns data in an array, arranged frame by frame. For signal data order within array please see [How to decode signal order](#).

This function can be called to determine, how many and which kind of values are in a data frame.

Attention! The transmited data can change if sensor configuration has changed, e.g. by sending a sensor command.

Parameter: int32_t IA_MaxValuesPerFrame

IA_MaxValuesPerFrame

Direction: Up

Valid for sensor: all

Description: Maximum number of values which can be transferred from sensor for one measurement.

Parameter: int32_t IA_ValuesPerFrame

IA_ValuesPerFrame

Direction: Up

Valid for sensor: all

Valid values:

Minimum: 0

Maximum: IA_MaxValuesPerFrame

Description: If MEDAQLib does not know, which data is transferred from sensor, this value can be 0. All following parameters exists from 1 to this number, e.g. IA_Index1, IA_Index2, ...

Parameter: int32_t IA_Index1..x

IA_Index1..x

Direction: Up

Valid for sensor: all

Valid values:

Minimum: 0

Maximum: as many values the sensor can transfer in one frame

Description: This value is for identification. Each value in a frame has unique index, which does never change, even if sensor configuration is changed.

Parameter: String IA_Raw_Name1..x	IA_Raw_Name1..x
Direction: Up	
Valid for sensor: all	
Description: The name of the value in the raw data.	
Parameter: String IA_Scaled_Name1..x	IA_Scaled_Name1..x
Direction: Up	
Valid for sensor: all	
Description: The name of the value in the scaled data.	
Parameter: double IA_Raw_RangeMin1..x	IA_Raw_RangeMin1..x
Direction: Up	
Valid for sensor: all	
Description: The minimum range of the value in the raw data.	
Parameter: double IA_Raw_RangeMax1..x	IA_Raw_RangeMax1..x
Direction: Up	
Valid for sensor: all	
Description: The maximum range of the value in the raw data.	
Parameter: double IA_Scaled_RangeMin1..x	IA_Scaled_RangeMin1..x
Direction: Up	
Valid for sensor: all	
Description: The minimum range of the value in the scaled data.	
Parameter: double IA_Scaled_RangeMax1..x	IA_Scaled_RangeMax1..x
Direction: Up	
Valid for sensor: all	
Description: The maximum range of the value in the scaled data.	
Parameter: double IA_ViewRange1..x	IA_ViewRange1..x
Direction: Up	
Valid for sensor: all	
Description: The best range of the constant (not moving) value in the scaled data. Can be used to determine axis scaling if signal is drawn in a plot.	
Parameter: String IA_Raw_Unit1..x	IA_Raw_Unit1..x
Direction: Up	
Valid for sensor: all	
Description: The unit of the value in the raw data, e.g. mm.	
Parameter: String IA_Scaled_Unit1..x	IA_Scaled_Unit1..x
Direction: Up	
Valid for sensor: all	
Description: The unit of the value in the scaled data, e.g. mm.	
Parameter: String IA_AdditionalInfo1..x	IA_AdditionalInfo1..x
Direction: Up	
Valid for sensor: all	
Description: Additional information of this sensor channel	
. Can contain several new line separated "parameter=value" pairs or can be empty. Some possible parameters are SensorName, SerialNumber or ArticleNumber.	

Parameter: double IA_Raw_Datarate1..x	IA_Raw_Datarate1..x
Direction: Up	
Valid for sensor: all	
Description: The datarate of the value in the raw data. Datarates of all values are always the same.	
Parameter: double IA_Scaled_Datarate1..x	IA_Scaled_Datarate1..x
Direction: Up	
Valid for sensor: all	
Description: The datarate of the value in the scaled data. Datarates of all values are always the same.	
Parameter: int32_t IA_MajorSignal1..x	IA_MajorSignal1..x
Direction: Up	
Valid for sensor: all	
Valid values:	
0= No major signal, e.g. Intensity, Counter, Timestamp, Statistic signal	
1= Major sensor signal, e.g. Distance at distance sensors	
Description: Each sensor has one or more important major signals.	
Parameter: int32_t IA_VideoSignalsPerFrame	IA_VideoSignalsPerFrame
Direction: Up	
Valid for sensor: all	
Valid values:	
Minimum: 0	
Maximum: 6	
Description: Some sensors already can transmit video signals within data stream. If this number is greater than 0 you have to read data from MEDAQLib by calling Get_VideoStreamSignal (described at specific sensor chapters) and TransferData alternating.	
All following parameters exists from 1 to this number, e.g. IA_VideoSignalName1, IA_VideoSignalName2, ...	
Parameter: String IA_VideoSignalName1..x	IA_VideoSignalName1..x
Direction: Up	
Valid for sensor: all	
Description: The name of the video signal.	
Parameter: int32_t IA_VideoSignalPixelCount1..x	IA_VideoSignalPixel-Count1..x
Direction: Up	
Valid for sensor: all	
Valid values:	
Minimum: 256	
Maximum: 2048	
Description: The number of pixels of the video signal.	
Parameter: int32_t IA_VideoSignalPixelByteSize1..x	IA_VideoSignalPixelByte-Size1..x
Direction: Up	
Valid for sensor: all	
Valid values:	
Minimum: 2	
Maximum: 2	
Description: The length in bytes of each pixels in the video signal.	

8.2.6 Set_TransferredData

In addition to the sensor setting, this command can reduce the selection of data transferred in [TransferData](#) (and also [Poll](#)). It also works for signals that can not be deactivated in the sensor or are just generated in the MEDAQLib.

This function affects the signals returned by [Get_TransmittedDataInfo](#), too.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available (however, all unspecified signals are reactivated here).

The specified settings are not persistant (will be lost between [CloseSensor](#) and next [OpenSensor](#)).

Parameter: int32_t IP_TransferMode

IP_TransferMode

Direction: Down

Valid values:

0= Any

1= Selected

Description: Specify if any signal (received from sensor or generated by MEDAQLib) should be output or only selected signals should be output.

Parameter: int32_t IP_UnderIndex0..n

IP_UnderIndex0..n

Direction: Down

Valid for sensor:

0= no

1= yes

Description: Specifies if the signal with this index is transferred. For a list of all available indices, call [Get_TransmittedDataInfo](#) and see parameter IA_Index1..x. n is [IA_MaxValuesPerFrame](#)-1.

On TransferMode 'Any', this parameters are ignored.

8.2.7 Get_TransferredData

Returns the selection of transferred signals.

Parameter: int32_t IA_TransferMode

IA_TransferMode

Direction: Up

Valid values:

0= Any

1= Selected

Description: Specify if any signal (received from sensor or generated by MEDAQLib) should be output or only selected signals should be output.

Parameter: int32_t IA_UnderIndex0..n

IA_UnderIndex0..n

Direction: Down

Valid for sensor:

0= no

1= yes

Description: Specifies if the signal whith this index is transferred. n is [IA_MaxValuesPerFrame](#)-1.

On TransferMode any, this parameters are not returned.

8.2.8 Enable_ReferenceTimestamp

For improved visualization or logging it is possible to enable an additional timestamp trace. Each timestamp value is transmitted at end of the data frame and can be read using [TransferData](#) (and also [Poll](#)). The timestamp it is in milli seconds starting at 01.01.1970 01:00.

This function affects the signals returned by [Get_TransmittedDataInfo](#), too.

Parameter: int32_t IP_ReferenceTimestamp

IP_ReferenceTimestamp

Direction: Down

Valid values:

- 1= Keep unchanged
- 0= None
- 1= Highest accuracy (automatic)
- 2= Sensor timestamp
- 3= Sensor datarate
- 4= Interpolated receive time
- 5= Equidistant receive time
- 6= Exact receive time

Description: Specify the algorithm to generate a timestamp trace.

Keep unchanged does not change the current setting but only returns it at [IA_ReferenceTimestamp](#).

Highest accuracy (automatic) tries to take the best one from 'Sensor timestamp' to 'Equidistant receive time' (using this order).

Sensor timestamp uses the timestamp transmitted by sensor for each value. If sensor does not transmit a timestamp, 'None' is selected and returned at [IA_ReferenceTimestamp](#).

The precision depends on sensor quartz accuracy. If sensor timestamp flips around while no data is transferred (e.g. no trigger is active at triggered mode), MEDAQLib does not recognize is and at timestamp trace may lag behind real time.

Sensor datarate uses the timestamp of first data packet at datarate of sensor to calculate a sensor trace. If sensor datarate is unknown or not constant (e.g. at triggered mode), 'None' is selected and returned at [IA_ReferenceTimestamp](#).

The precision depends on computer quartz accuracy.

Interpolated receive time uses the timestamp of each received data packet and interpolates the timestamp of dataframes within one data packet. This is a quite inaccurate method, e.g. if sensor data is aquired by external trigger signals.

If sensor measrate is unknown, 'None' is selected and returned at [IA_ReferenceTimestamp](#).

Equidistant receive time uses the timestamp of last and current received data packet and set current values evenly distributed between this timestamps. This is a quite inaccurate method, e.g. if sensor data is aquired by external trigger signals.

Exact receive time uses the timestamp of each received data packet and set this timestamp for each dataframe within the packet.

Parameter: int32_t IA_ReferenceTimestamp

IA_ReferenceTimestamp

Direction: Up

Valid for sensor:

- 0= None

- 1= Highest accuracy (automatic)
- 2= Sensor timestamp
- 3= Sensor datarate
- 4= Interpolated receive time
- 5= Equidistant receive time
- 6= Exact receive time

Description: Returns the mode actually set.

8.2.9 Use_Defaults

This command tells the driver to use default values to operate with sensor data. If some parameters are not specified they are not changed. The sensor is not affected by this command.

The following parameters are interface parameters and already described at chapter [Interface parameters](#):

[IP_AutomaticMode](#), [IP_ScaleErrorValues](#), [IP_FixedErrorValue](#), [IP_RingBufferSize](#) and [CP_BaseLevelTimeout](#). They are valid for each sensor and interface and can be changed using this command (not only when opening the driver).

Parameters only valid for a specific sensor are not described here but at command [Use_Defaults](#) of specific sensor chapter.

8.2.10 SettingsChanged

Checks if sensor or driver settings have changed since last call of [Get_DrvSetting](#).

Parameter: int32_t IP_ResetFlag

IP_ResetFlag

Direction: Down

Valid values:

- 0= FALSE
- 1= TRUE

Description: If 1, the flag is resetted. Otherwise the flag is resetted at [Get_DrvSetting](#).

Parameter: int32_t IA_SettingsChanged

IA_SettingsChanged

Direction: Up

Valid values:

- 0= FALSE
- 1= TRUE

Description: 1 when settings have changed.

8.2.11 Get_DrvSetting

Returns the current settings of the driver used for operating with sensor data. It is the opposite of [Use_Defaults](#). The sensor is not affected by this command.

The following parameters returned, are already described at chapter [Interface parameters](#) (but instead of IP_... for interface parameter is called now IA_... for interface answer):

[IA_AutomaticMode](#), [IA_ScaleErrorValues](#), [IA_FixedErrorValue](#), [IA_RingBufferSize](#) and [CA_BaseLevelTimeout](#).

[IA_SensorType](#) and [IA_SensorTypeName](#) are described at [Get_FoundSensor](#).

They are returned for each sensor and interface.

Parameters only valid for a specific sensor are not described here but at command [Get_DrvSetting](#) of specific sensor chapter.

8.2.12 Get_HighResolutionTime

Returns the current timestamp (as high resolution time) used by MEDAQLib.

Parameter: double IA_HighResolutionTime	IA_HighResolutionTime
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.79769e+308 (DBL_MAX)	
Unit: ms	
Description: High resolution time in ms. It starts from midnight, January 1, 1970 (Unix time).	

8.2.13 Exec_SleepTime

Sleeps (not busy waiting) a specified period within MEDAQLib.

Parameter: int32_t IP_SleepTime	IP_SleepTime
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Unit: ms	
Description: High resolution time period in ms.	

8.2.14 Cmd_Generic

With Cmd_Generic, any data can be sent to the sensor.

Parameter: String SP_CmdStr	SP_CmdStr
Direction: Down	
Description: The command string as it is sent to the sensor. E.g. for ILD sensors it always starts with "+++\\rILD1" ('\\r' is carriage return, 0x0d).	
Parameter: int32_t CP_SensorAnswerTimeout	CP_SensorAnswerTimeout
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Unit: ms	
Default: 500	
Description: This special timeout is added to the global timeout (specified by CP_BaseLevelTimeout) when waiting for the complete answer from sensor.	
Parameter: int32_t CP_InitAfterSensorCommand	CP_InitAfterSensorCommand
Direction: Down	
Valid values:	
0= No	
1= Yes	
Default: 0	
Description: This parameter tells MEDAQLib if it should update internal states after processing generic sensor commands. Some sensor commands (as if IP_AutomaticMode is 1) are called in this case.	

Parameter: String SA_CompleteAnswer SA_CompleteAnswer
Direction: Up
Description: The raw (not interpreted) answer (to a command) from the sensor is stored here.

8.2.15 Clear_Buffers

This command is executed by the driver and does not affect the sensor. It erases the ring buffer and the input and output buffer of the interface.

Parameter: int32_t SP_AllDevices SP_AllDevices
Direction: Down
Valid values:
 0= FALSE
 1= TRUE
Default: 0
Description: If it is set to 1, not only this instance but all instances created from this driver are cleared. It can be used to synchronize data acquisition from several sensors because after clearing all buffers the next data from all sensors have same timestamp.

8.2.16 DataAvail_Event

This command registers an event which will be set when new data is available. The event can be used to wait for new data in your application.

Parameter: int32_t IP_EventOnAvailableValues IP_EventOnAvailableValues
Direction: Down
Valid values:
Minimum: -1
Maximum: 2147483647 (INT32_MAX)
Default: -1
Description: If it is set to -1, the event is not set by this condition.
 Otherwise the event is set if at least so many values are available as specified here.

Parameter: double IP_EventOnBufferFillsize IP_EventOnBufferFillsize
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 1.0
Default: 1.0
Description: If it is set to 1.0, the event is not set by this condition.
 Otherwise the event is set if the ring buffer fill size is at least as high as specified here (0.0 means empty, 1.0 means full).
 For setting the ring buffer size please see parameter [IP_RingBufferSize](#).

Parameter: ptr_t (void *) IA_DataAvailEvent IA_DataAvailEvent
Direction: Up
Description: The event handle for data avail.
 If IP_EventOnAvailableValues is -1 (default) and IP_EventOnBufferFillsize is 1.0 (default), the event is released and the value is nullptr.

Example how to use the event:

```

/* At your main initialisation function: */
ERR_CODE err;
err= SetIntExecSCmd (instance, "DataAvail_Event", "IP_EventOnAvailableValues", 1024);
/* error handling, if err!=ERR_NOERROR */
HANDLE event= nullptr;
GetParameterDWORD_PTR (instance, "IA_DataAvailEvent", (ptr_t *)&event);

/* At your working thread or function: */
/* Maybe the event was already set before, but we are not interested in old data,
so clear old data and reset event. */
err= ExecSCmd (instance, "Clear_Buffers");
/* error handling, if err!=ERR_NOERROR */
ResetEvent (event);

while (true)
{
    /* timeout is 1000 ms */
    if (WaitForSingleObject (event, 1000)==WAIT_OBJECT_0)
        err= TransferData (instance, raw, scaled, 1024, &read);
    /* error handling, verify that read==1024, process data */
}

```

8.2.17 Open_DataSocket

This command is only available for sensors with are connected via [TCP/IP](#) and different command and data port: [SENSOR_ILD2300](#), [SENSOR_IFD2421](#), [SENSOR_IFD2422](#), [SENSOR_IFD2445](#), [SENSOR_IFD2451](#), [SENSOR_IFD2461](#), [SENSOR_IFD2471](#), [SENSOR_ODC2520](#), [SENSOR_IMC5400](#), [SENSOR_IMC5600](#), [SENSOR_ACS7000](#), [CONTROLLER_CSP2008](#), [CONTROLLER_CBOX](#), [THICKNESS_SENSOR](#), [ETH_ADAPTER_IF2008](#), [ETH_IF1032](#), [SENSOR_DT3060](#), [CONTROLLER_DT6500](#) (only DT6530), [CONTROLLER_DT6536](#), [CONTROLLER_DT6200](#) and [CONTROLLER_KSS64xx](#).

Parameter: int32_t IP_RemoteDataProtocol

IP_RemoteDataProtocol

Direction: Down

Valid for sensor:

[SENSOR_ILD2300](#)
[SENSOR_IFD2421](#)
[SENSOR_IFD2422](#)
[SENSOR_IFD2445](#)
[SENSOR_IFD2451](#)
[SENSOR_IFD2461](#)
[SENSOR_IFD2471](#)
[SENSOR_ODC2520](#)
[CONTROLLER_CSP2008](#)
[CONTROLLER_CBOX](#)
[CONTROLLER_CBOX](#)
[THICKNESS_SENSOR](#)
[IF2008_ETH](#)
[SENSOR_IMC5400](#)
[SENSOR_IMC5600](#)
[SENSOR_ACS7000](#)

Valid values:

-1= Query automatic (not for [CONTROLLER_CSP2008](#))

0= TCP server

- 1= TCP client (not for **CONTROLLER_CBOX**, **THICKNESS_SENSOR** and **ETH_ADAPTER_IF2008**)
- 2= UDP sender (not for **CONTROLLER_CBOX**, **THICKNESS_SENSOR** and **ETH_ADAPTER_IF2008**)

Description: IP Protocol used for data transfer. Local host automatically takes opposite.

Parameter: int32_t IP_PortDetection

IP_PortDetection

Direction: Down

Valid for sensor:

- SENSOR_DT3060**
- CONTROLLER_DT6200**
- CONTROLLER_KSS64xx**
- CONTROLLER_DT6500**
- CONTROLLER_DT6536**
- ETH_IF1032**

Valid values:

- 0= Set port(s) manually (**SENSOR_DT3060**, **CONTROLLER_DT6200**, **CONTROLLER_KSS64xx**, **CONTROLLER_DT6500**, **CONTROLLER_DT6536** + **ETH_IF1032**)
- 1= Query all ports automatically (only **CONTROLLER_DT6500**)
- 2= Query data port automatically (**SENSOR_DT3060**, **CONTROLLER_DT6200**, **CONTROLLER_KSS64xx**, **CONTROLLER_DT6536** + **ETH_IF1032**)

Description: If ports should be queried automatically, **IP_DataPort** local setting is ignored and data port is queried automatically.

Parameter: int32_t IP_DataPort

IP_DataPort

Direction: Down

Valid for sensor:

- SENSOR_ILD2300**
- SENSOR_IFD2421**
- SENSOR_IFD2422**
- SENSOR_IFD2445**
- SENSOR_IFD2451**
- SENSOR_IFD2461**
- SENSOR_IFD2471**
- SENSOR_ODC2520**
- SENSOR_DT3060**
- CONTROLLER_DT6200**
- CONTROLLER_KSS64xx**
- CONTROLLER_DT6500**
- CONTROLLER_DT6536**
- CONTROLLER_CSP2008**
- CONTROLLER_CBOX**
- THICKNESS_SENSOR**
- ETH_IF1032**
- IF2008_ETH**
- SENSOR_IMC5400**
- SENSOR_IMC5600**
- SENSOR_ACS7000**

Valid values:

Minimum: 0

Maximum: [CONTROLLER_CSP2008](#): 32384, otherwise 65535

Description: If remote data protocol is TCP server this is the remote data port, otherwise it is the local data port. If remote data protocol is set to query automatic, this parameter is ignored. If this parameter is 0, the data port is not opened.

8.2.18 Set_Setup

This command import a parameter set to a sensor.

Parameter: String SP_SetupRecord

SP_SetupRecord

Direction: Down

Description: Parameter set which should be imported to sensor. Format is same as at [SensorTest](#).

First line must contain a header. Version and sensor type will be checked..

Following lines will be processed one by one. Comments (starting with #) will be ignored.

8.2.19 Get_Setup

This command exports all sensor settings.

It is only available for sensors [SENSOR_ILR2250](#), [SENSOR_ILD1220](#), [SENSOR_ILD1320](#), [SENSOR_ILD1420](#), [SENSOR_ILD1750](#), [SENSOR_ILD1900](#), [SENSOR_ILD2300](#), [SENSOR_IFD2421](#), [SENSOR_IFD2422](#), [SENSOR_IFD2445](#), [SENSOR_IFD2451](#), [SENSOR_IFD2461](#), [SENSOR_IFD2471](#), [SENSOR_ILR2250](#), [SENSOR_ODC2500](#), [SENSOR_ODC2520](#), [SENSOR_ODC2600](#), [SENSOR_IMC5400](#), [SENSOR_IMC5600](#), [SENSOR_DT3060](#), [SENSOR_DT3100](#), [SENSOR_DT6100](#), [CONTROLLER_DT6200](#), [CONTROLLER_KSS64xx](#), [CONTROLLER_KSS6380](#), [CONTROLLER_DT6500](#), [CONTROLLER_DT6536](#), [ETH_IF1032](#), [CONTROLLER_CBOX](#), [THICKNESS_SENSOR](#), [ETH_ADAPTER_IF2008](#), [SENSOR_ACS7000](#) and [SENSOR_MFA](#).

Parameter: int32_t SP_IncludeSavedSetups

SP_IncludeSavedSetups

Direction: Down

Valid for sensor:

- [SENSOR_ILR2250](#) - include saved parameter set
- [SENSOR_ILD1220](#) - include all existing parameter sets
- [SENSOR_ILD1320](#) - include all existing parameter sets
- [SENSOR_ILD1420](#) - include all existing parameter sets
- [SENSOR_ILD1750](#) - include all existing parameter sets
- [SENSOR_ILD1900](#) - include all existing parameter sets
- [SENSOR_ILD2300](#) - include all existing parameter sets
- [SENSOR_ILR2250](#) - include saved setup
- [SENSOR_ODC2500](#) - include OptionData and UserMeasProgData
- [SENSOR_ODC2520](#) - include all existing parameter sets
- [SENSOR_ODC2600](#) - include OptionData and UserMeasProgData
- [SENSOR_IFD2421](#) - include all existing parameter sets
- [SENSOR_IFD2422](#) - include all existing parameter sets
- [SENSOR_IFD2445](#) - include all existing parameter sets
- [SENSOR_IFD2451](#) - include all existing parameter sets
- [SENSOR_IFD2461](#) - include all existing parameter sets
- [SENSOR_IFD2471](#) - include all existing parameter sets
- [SENSOR_DT3100](#) - include saved settings

SENSOR_DT6100 - include saved setup
 CONTROLLER_KSS6380 - include saved setup
 CONTROLLER_DT6500 - include saved setup (not DT6530)
 CONTROLLER_CBOX - include all parameter sets
 THICKNESS_SENSOR - include all parameter sets
 ETH_ADAPTER_IF2008 - include all parameter sets
 SENSOR_IMC5400 - include all existing parameter sets
 SENSOR_IMC5600 - include all existing parameter sets
 SENSOR_ACS7000 - include all existing parameter sets
 SENSOR_MFA - include saved parameter set

Valid values:

0= No
 1= Yes

Default: 0

Description: Specifies if saved settings should be exported, too.

Parameter: int32_t SP_IncludeAdditionalParameters

SP_IncludeAdditionalParameters

Direction: Down

Valid for sensor:

SENSOR_ILD2300 - includes Material Table
 SENSOR_IFD2421 - includes Material Table
 SENSOR_IFD2422 - includes Material Table
 SENSOR_IFD2445 - includes Material Table
 SENSOR_IFD2451 - includes Material Table
 SENSOR_IFD2461 - includes Material Table
 SENSOR_IFD2471 - includes Material Table
 CONTROLLER_DT6200 - include Measure Ranges, Linearisation Modes
 and Linearisation Points
 CONTROLLER_KSS64xx - include Measure Ranges, Linearisation Modes
 and Linearisation Points
 CONTROLLER_DT6500 - include Measure Ranges (only DT6530), Lin-
 earisation Modes and Linearisation Points (only DT6530)
 CONTROLLER_DT6536 - include Measure Ranges, Linearisation Modes
 and Linearisation Points
 SENSOR_IMC5400 - includes Material Table
 SENSOR_ACS7000 - includes Color Table

Valid values:

0= No
 1= Yes

Default: 0

Description: Specifies if additional parameters should be exported, too.

Parameter: int32_t SP_IncludeInterfaceParameters

SP_IncludeInterfaceParam-
eters

Direction: Down

Valid for sensor:

SENSOR_ILR2250
 SENSOR_ILD1220
 SENSOR_ILD1320
 SENSOR_ILD1420
 SENSOR_ILD1750
 SENSOR_ILD1900
 SENSOR_ILD2300
 SENSOR_IFD2421

```

SENSOR_IFD2422
SENSOR_IFD2445
SENSOR_IFD2451
SENSOR_IFD2461
SENSOR_IFD2471
SENSOR_ILR2250
SENSOR_ODC2500
SENSOR_ODC2520
SENSOR_ODC2600
SENSOR_DT3060
CONTROLLER_DT6200
CONTROLLER_KSS64xx
CONTROLLER_DT6500 (only DT6530)
CONTROLLER_DT6536
ETH_IF1032
CONTROLLER_CBOX
THICKNESS_SENSOR
ETH_ADAPTER_IF2008
SENSOR_IMC5400
SENSOR_IMC5600
SENSOR_ACS7000
SENSOR_MFA
  
```

Valid values:

0 = No
1 = Yes

Default: 0

Description: Specifies if hardware interface parameters should be exported, too.

Attention!

Interface parameters at sensor are changed immediately, so connection may be lost at [Set_Setup](#) (if parameters are changed for currently used interface). Connection may be lost at [Get_Setup](#) too (if [SP_IncludeSavedSetup](#)s is 1 and interface parameters differs from current setup (RAM) and a saved setup).

Parameter: String SA_SetupRecord

SA_SetupRecord

Direction: Up

Description: Parameter set with exported settings. Format is same as at [SensorTest](#).

First line contains a header (version, sensor type, date and time of export). Following lines contains all sensor commands and it's parameters which are needed to setup a sensor with same settings.

8.2.20 Update_... (Meta command)

This command is a meta command for any sensors with Get_/Set_... command pairs.

It simplifies changing just a few parameters of a larger parameter set.

First, the command name is changed from Update_... to Get_... and this function is called (including the rest of the parameter set, too). The answer parameters from sensor (SA_...) are renamed to sensor parameters (SP_...) and the parameter set (given by the user) is copied over the working parameter set. The outcome of this

is a new parameter set, where all parameters are current sensor settings except the parameters which are specified by the user. Than the command name is changed to Set_... again and the command is send to sensor.

This command has no obligatory parameters, only parameters which should be changed at sensor must be specified.

It is also possible, to call an Update_... command, where the matching Get_... command needs own parameters, too (e.g. channel number). Just specify this parameter before calling Update_... command.

Following example shows how to setup **SENSOR_ILD2300** to output additional counter:

```
SetParameterString (instance, "S_Command", "Update_OutputAdditional_ETH");
SetParameterInt (instance, "SP_OutputAdditionalCounter_ETH", 1);
err= SensorCommand (instance);
/* error handling, if err!=ERR_NOERROR */
```

Another command which updates a mathematic channel at DT6200::

```
SetParameterDouble (instance, "SP_FactorCh3", 1.19);
err= SetIntExecSCmd (instance, "Update_MathFunction", "SP_Chан", 2);
/* error handling, if err!=ERR_NOERROR */
```

8.2.21 Reset_... (Meta command)

This command is a meta command for sensors with Set_... commands with multiple parameters.

Multiple parameters are send to sensor as a list and specified at MEDAQLib by 0 or 1.

It simplifies setting just a few parameters of a larger parameter set. All missing parameters are reset.

Internally the command is translated to a Set_... command with any multiple parameter optional. Only parameters which should be set at sensor must be specified.

It is also possible, to call a Reset_... command, where the multiple parameters are needed, too (e.g. resampling count at command Set_Resampling). Just specify this parameter before calling Reset_... command.

Following example shows how to setup **SENSOR_ILD2300** to output only one additional value (Timestamp). Any other additional value is switched off:

```
SetParameterString (instance, "S_Command", "Reset_OutputAdditional_ETH");
SetParameterInt (instance, "SP_OutputAdditionalTimestamp_ETH", 1);
err= SensorCommand (instance);
/* error handling, if err!=ERR_NOERROR */
```

8.3 SensorFinder commands

MEDAQLib supports commands to detect sensors at many different interfaces with any interface settings. Following table shows the sensors which can be detected and the possible interfaces.

SENSOR_ILR110x_115x	RS232 (IF2001_USB), IF2008, IF2008_ETH, IF2004_USB
SENSOR_ILR118x (01) - RS232	RS232
SENSOR_ILR118x (02) - RS422	RS232 (IF2001_USB), IF2008, IF2008_ETH, IF2004_USB
SENSOR_ILR1191 (01) - RS232	RS232
SENSOR_ILR1191 (02) - RS422	RS232 (IF2001_USB), IF2008, IF2008_ETH, IF2004_USB
SENSOR_ILR2250	RS232 (IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSOR_ODC1202	RS232
SENSORILD1220	RS232 (IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSORILD1302	RS232 (IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSORILD1320	RS232 (IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSORILD1401	RS232
SENSORILD1402	RS232, IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSORILD1402 (Comp. mode)	RS232 (IF2001_USB), IF2008, IF2008_ETH, IF2004_USB
SENSORILD1420	RS232 (IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSORILD1700	RS232 (IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSORILD1750	RS232 (IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSORILD1900	RS232 (IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSORILD2200	RS232 (IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSORILD2300	RS232 (IF2001_USB), IF2004, TCP/IP (local sub- net), IF2008, IF2008_ETH, IF2004_USB
SENSORIFD2401	RS232, WinUSB, IF2008, IF2008_ETH, IF2004_USB
SENSORIFD2421	RS232 (IF2001_USB), IF2004, TCP/IP (local sub- net), IF2008, IF2008_ETH, IF2004_USB
SENSORIFD2422	RS232 (IF2001_USB), IF2004, TCP/IP (local sub- net), IF2008, IF2008_ETH, IF2004_USB
SENSORIFD2431	RS232, WinUSB, IF2008, IF2008_ETH, IF2004_USB
SENSORIFD2445	RS232 (IF2001_USB), IF2004, TCP/IP (local sub- net), IF2008, IF2008_ETH, IF2004_USB
SENSORIFD2451	RS232 (IF2001_USB), IF2004, TCP/IP (local sub- net), IF2008, IF2008_ETH, IF2004_USB

SENSOR_IFD2461	RS232 (IF2001_USB), IF2004, TCP/IP (local subnet), IF2008, IF2008_ETH, IF2004_USB
SENSOR_IFD2471	RS232 (IF2001_USB), IF2004, TCP/IP (local subnet), IF2008, IF2008_ETH, IF2004_USB
SENSOR_ODC2500	RS232 (native, IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSOR_ODC2520	RS232 (IF2001_USB), IF2004, TCP/IP (local subnet), IF2008, IF2008_ETH, IF2004_USB
SENSOR_ODC2600	RS232 (native, IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
SENSOR_DT3060	TCP/IP (local subnet)
SENSOR_DT3100	TCP/IP (local subnet)
SENSOR_DT6100	TCP/IP (local subnet)
SENSOR_MEBUS	RS232 (RS485/USB converter)
SENSOR_DT6120	RS232 (RS485/USB converter)
CONTROLLER_KSS6380	TCP/IP (local subnet)
CONTROLLER_DT6200	TCP/IP (local subnet)
CONTROLLER_KSS64xx	TCP/IP (local subnet)
CONTROLLER_DT6500	TCP/IP (local subnet)
CONTROLLER_DT6536	TCP/IP (local subnet)
PCI_CARD_IF2004	IF2004 (channel 3)
USB_ADAPTER_IF2004	IF2004_USB (channel 5)
PCI_CARD_IF2008	IF2008 (channel 6 to 11)
ETH_ADAPTER_IF2008	IF2008_ETH (local subnet)
SENSOR_LLT27xx	TCP/IP (local subnet)
SENSOR_IMC5400	RS232 (IF2001_USB), TCP/IP (local subnet), IF2008, IF2008_ETH, IF2004_USB
SENSOR_IMC5600	RS232 (IF2001_USB), TCP/IP (local subnet), IF2008, IF2008_ETH, IF2004_USB
SENSOR_ACS7000	RS232 (IF2001_USB), IF2004, TCP/IP (local subnet), IF2008, IF2008_ETH, IF2004_USB
SENSOR_CFO	TCP/IP (local subnet)
SENSOR_MFA	RS232 (IF2001_USB), IF2004, IF2008, IF2008_ETH, IF2004_USB
CONTROLLER_CSP2008	TCP/IP (local subnet), on RS422 the CSP does not support detection
CONTROLLER_CBOX	RS232 (IF2001_USB), IF2004, TCP/IP (local subnet), IF2008, IF2004_USB, WinUSB
THICKNESS_SENSOR	TCP/IP (local subnet)
ETH_IF1032	TCP/IP (local subnet)

Because not every sensor does support same interface parameters as the interface at computer, there may be combinations which won't work.

Scanning for sensors runs asynchronous (each physical interface, e.g. COM1, COM2, IF2008[Card1, Channel1], IF2008[Card1, Channel2], ... in an own thread).

First, this will ensure the sensors are found as fast as possible, and second, your calling application is not blocked and can do anything else, e.g. showing an progress bar. To do a scan, first create a sensor instance. You can create it for any sensor, but it's beneficial to use the sensor you want to search for. Now you can use following sensor commands (internally processed by MEDAQLib) to perform a scan.

8.3.1 Start_FindSensor

This command starts a new scan for a sensor.

Parameter: int32_t IP_SensorType IP_SensorType

Direction: Down

Valid values:

- SENSOR_ILR110x_115x (19)
- SENSOR_ILR118x (20)
- SENSOR_ILR1191 (21)
- SENSOR_ILR2250 (60)
- SENSOR_ILD1220 (56)
- SENSOR_ILD1302 (24)
- SENSOR_ILD1320 (41)
- SENSOR_ILD1401 (1)
- SENSOR_ILD1402 (23)
- SENSOR_ILD1420 (42)
- SENSOR_ILD1700 (2)
- SENSOR_ILD1750 (51)
- SENSOR_ILD1900 (58)
- SENSOR_ILD2200 (5)
- SENSOR_ILD2300 (29)
- SENSOR_IFD2401 (12)
- SENSOR_IFD2421 (46)
- SENSOR_IFD2422 (47)
- SENSOR_IFD2431 (13)
- SENSOR_IFD2445 (39)
- SENSOR_IFD2451 (30)
- SENSOR_IFD2461 (44)
- SENSOR_IFD2471 (26)
- SENSOR_ODC1202 (25)
- SENSOR_ODC2500 (8)
- SENSOR_ODC2520 (37)
- SENSOR_ODC2600 (9)
- SENSOR_DT3060 (50)
- SENSOR_DT3100 (28)
- SENSOR_DT6100 (16)
- SENSOR_DT6120 (40)
- SENSOR_MEBUS (43)
- CONTROLLER_DT6200 (33)
- CONTROLLER_KSS64xx (45)
- CONTROLLER_DT6500 (15)
- CONTROLLER_DT6536 (54)
- CONTROLLER_KSS6380 (18)
- PCI_CARD_IF2004 (10)
- PCI_CARD_IF2008 (22)

[ETH_ADAPTER_IF2008](#) (52)
 SENSOR_LLT27xx, scanCONTROL and gapCONTROL (31)
[SENSOR_IMC5400](#) (55)
[SENSOR_IMC5600](#) (59)
[SENSOR_ACS7000](#) (35)
[SENSOR_CFO](#) (53)
[SENSOR_MFA](#) (61)
[CONTROLLER_CSP2008](#) (32)
[CONTROLLER_CBOX](#) (38)
[THICKNESS_SENSOR](#) (48)
[ETH_IF1032](#) (34)
[USB_ADAPTER_IF2004](#) (36)

Default: Automatic

Description: Type of sensor to scan for. If this parameter is not specified, the value specified by [CreateSensorInstance](#) it is automatically taken.

Parameter: int32_t IP_FindSimilarSensors

IP_FindSimilarSensors

Direction: Down

Valid values:

0= FALSE
1= TRUE

Default: 0

Description: When this parameter is true, the sensor name is not checked at sensor answer. So any sensor using same protocol can be found. This is an internal parameter. It should not be used by the customer.

Parameter: int32_t IP_ScanHWR232

IP_ScanHWR232

Direction: Down

Valid values:

0= FALSE
1= TRUE

Default: 1

Description: The sensor is scanned on serial interface [RS232](#) (also IF2001 - USB (RS422) or RS485/USB converter).

Parameter: int32_t IP_ScanHWIF2004

IP_ScanHWIF2004

Direction: Down

Valid values:

0= FALSE
1= TRUE

Default: 1

Description: The sensor is scanned on PCI card [IF2004](#).

Parameter: int32_t IP_ScanHWIF2004_USB

IP_ScanHWIF2004_USB

Direction: Down

Valid values:

0= FALSE
1= TRUE

Default: 1

Description: The sensor is scanned on USB adapter [IF2004_USB](#).

Parameter: int32_t IP_ScanHWTCPPIP IP_ScanHWTCPPIP

Direction: Down

Valid values:

0= FALSE
1= TRUE

Default: 1

Description: The sensor is scanned on network protocoll [TCP/IP](#).

Parameter: int32_t IP_ScanHWWinUSB IP_ScanHWWinUSB

Direction: Down

Valid values:

0= FALSE
1= TRUE

Default: 1

Description: The sensor is scanned on generic USB driver [WinUSB](#).

Parameter: int32_t IP_ScanHWIF2008 IP_ScanHWIF2008

Direction: Down

Valid values:

0= FALSE
1= TRUE

Default: 1

Description: The sensor is scanned on PCI(e) card [IF2008](#).

If IF2008 cards are scanned, IF2008 virtual COM ports are excepted from scanning automatically.

Parameter: int32_t IP_ScanHWIF2008_ETH IP_ScanHWIF2008_ETH

Direction: Down

Valid values:

0= FALSE
1= TRUE

Default: 1

Description: The sensor is scanned on ethernet adapter [IF2008_ETH](#).

Parameter: int32_t IP_ForceScanUsedChannel IP_ForceScanUsedChannel

Direction: Down

Valid values:

0= Do not interrupt sensor communication of other processes.
1= Switch channel mode and baudrate. Other processes will be disturbed.

Default: 0

Description: This parameter is only used at sensors connected at [IF2008_ETH](#).

If [Set_ChannelMode<n>](#) at [IF2008_ETH](#) is Sensor or Encoder, this channel is skipped from searching. Other programms connected to the sensor are not affected.

But it can happen, no other programms are connected but [Set_ChannelMode<n>](#) is not None anyway. In this case the parameter can be set to 1 to search at this channel nevertheless.

Parameter: int32_t IP_KeepChannelMode	IP_KeepChannelMode
Direction: Down	
Valid values:	
0= Switch channel mode to desired mode (encoder or sensor).	
1= Do not change current channel mode (at open and at close). If it does not match desired mode, an error is output.	
Default: 0	
Description: This parameter is only used at sensors connected at IF2008_ETH . Normally it will be only set if IP_ForceScanUsedChannel is set, too. If channel mode is changed at IF2008_ETH , Tx lines may be changed to Rx lines or become high resistance. Some (old) sensors cannot deal with this. To avoid communications problems for these sensors, IF2008_ETH card can be setup before and Set_ChannelMode<n> is not changed at open or close.	
Parameter: int32_t IP_QuickScanRS485	IP_QuickScanRS485
Direction: Down	
Valid values:	
0= All possible addresses are scanned (secure, but slow).	
1= Just broadcast address is scanned (only works if only one sensor is plugged at RS485 bus).	
Default: 0	
Description: This parameter is only used for sensors which supports RS485 interfaces. At RS485, several sensors can be connected, but just one is allowed to send data at same time. Therefore the fast scan only works if only one sensor is connected, otherwise there would be a data collision. If you are not sure, how much sensors are plugged, do not set this parameter. In this case the secure scan is processed.	
Parameter: String IP_RS232Ports	IP_RS232Ports
Direction: Down	
Valid values:	
Comma speareated list of COM ports.	
Default: Empty	
Description: Specify a list of serial ports for RS232 scanning. If not specified, (normal case), the available ports are detected automatically.	
Parameter: String IP_ChannelNumbers	IP_ChannelNumbers
Direction: Down	
Valid values:	
Comma speareated list of channel numbers.	
Default: Empty	
Description: Specify a list of channel numbers for scanning on serial multi channel interface (IF2004 , IF2004_USB , IF2008 and IF2008_ETH). If not specified, (normal case), all channels supported by interface card are used.	
Parameter: String IP_Baudrates	IP_Baudrates
Direction: Down	
Valid values:	
Comma speareated list of baudrates.	
Default: Empty	
Description: Specify a list of baudrates for scanning any serial interface (RS232 , IF2004 , IF2004_USB , IF2008 and IF2008_ETH). If not specified, (normal case), the baudrates supported by sensor are used.	

Parameter: String IP_ByteSizes IP_ByteSizes

Direction: Down

Valid values:

Comma separated list of byte sizes.

Default: Empty

Description: Specify a list of byte sizes for scanning any serial interface ([RS232](#), [IF2004](#), [IF2004_USB](#), [IF2008](#) and [IF2008_ETH](#)).

If not specified, (normal case), the byte sizes supported by sensor are used.

Parameter: String IP_Parities IP_Parities

Direction: Down

Valid values:

Comma separated list of parities (0= no, 1= odd, 2= even).

Default: Empty

Description: Specify a list of parities for scanning any serial interface ([RS232](#), [IF2004](#), [IF2004_USB](#), [IF2008](#) and [IF2008_ETH](#)).

If not specified, (normal case), the parities supported by sensor are used.

Parameter: String IP_Stopbits IP_Stopbits

Direction: Down

Valid values:

Comma separated list of stop bits (0= one, 2= two).

Default: Empty

Description: Specify a list of stop bits for scanning any serial interface ([RS232](#), [IF2004](#), [IF2004_USB](#), [IF2008](#) and [IF2008_ETH](#)).

If not specified, (normal case), the stop bits supported by sensor are used.

8.3.2 Get_FindSensorProgress

Get the progress of an active scan.

Parameter: double IA_Progress IA_Progress

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 100.0

Unit: %

Description: Progress of the scan.

Parameter: int32_t IA_Found IA_Found

Direction: Up

Description: Number of sensors found until now.

Parameter: String IA_ScanStatus IA_ScanStatus

Direction: Up

Description: Human readable multiline string containing scan status for debugging purpose.

8.3.3 Get_FoundSensor

Get information about a found sensor.

Parameter: int32_t IP_Index IP_Index

Direction: Down

Valid values:

Minimum: 0

Maximum: Number of sensors found -1

Description: Index (0 based) of sensor to get information.

Parameter: int32_t IP_SensorType IP_SensorType

Direction: Up

Valid values:

- SENSOR_ILR110x_115x (19)
- SENSOR_ILR118x (20)
- SENSOR_ILR1191 (21)
- SENSOR_ILR2250 (60)
- SENSOR_ILD1220 (56)
- SENSOR_ILD1302 (24)
- SENSOR_ILD1320 (41)
- SENSOR_ILD1401 (1)
- SENSOR_ILD1402 (23)
- SENSOR_ILD1420 (42)
- SENSOR_ILD1700 (2)
- SENSOR_ILD1750 (51)
- SENSOR_ILD1900 (58)
- SENSOR_ILD2200 (5)
- SENSOR_ILD2300 (29)
- SENSOR_IFD2401 (12)
- SENSOR_IFD2421 (46)
- SENSOR_IFD2422 (47)
- SENSOR_IFD2431 (13)
- SENSOR_IFD2445 (39)
- SENSOR_IFD2451 (30)
- SENSOR_IFD2461 (44)
- SENSOR_IFD2471 (26)
- SENSOR_ODC1202 (25)
- SENSOR_ODC2500 (8)
- SENSOR_ODC2520 (37)
- SENSOR_ODC2600 (9)
- SENSOR_DT3060 (50)
- SENSOR_DT3100 (28)
- SENSOR_DT6100 (16)
- SENSOR_DT6120 (40)
- SENSOR_MEBUS (43)
- CONTROLLER_DT6200 (33)
- CONTROLLER_KSS64xx (45)
- CONTROLLER_DT6500 (15)
- CONTROLLER_DT6536 (54)
- CONTROLLER_KSS6380 (18)
- PCI_CARD_IF2004 (10)
- PCI_CARD_IF2008 (22)

[ETH_ADAPTER_IF2008](#) (52)
 SENSOR_LLT27xx, scanCONTROL and gapCONTROL (31)
[SENSOR_IMC5400](#) (55)
[SENSOR_IMC5600](#) (59)
[SENSOR_ACS7000](#) (35)
[SENSOR_CFO](#) (53)
[SENSOR_MFA](#) (61)
[CONTROLLER_CSP2008](#) (32)
[CONTROLLER_CBOX](#) (38)
[THICKNESS_SENSOR](#) (48)
[ETH_IF1032](#) (34)
[USB_ADAPTER_IF2004](#) (36)
 0

Description: Type of found sensor. Is the same as searched for if sensor matches or 0 if sensor is similar to the sensor searched for.

Parameter: String IP_SensorTypeName

IP_SensorTypeName

Direction: Up

Valid values:

ILR110x_115x
 ILR118x
 ILR1191
 ILR2250
 ILD1220
 ILD1302
 ILD1320
 ILD1401
 ILD1402
 ILD1420
 ILD1700
 ILD1750
 ILD1900
 ILD2200
 ILD2300
 IFD2401
 IFD2421
 IFD2422
 IFD2431
 IFD2445
 IFD2451
 IFD2461
 IFD2471
 ODC1202
 ODC2500
 ODC2520
 ODC2600
 DT3060
 DT3100
 DT6100
 DT6120
 SensorOnMEbus
 DT6200
 KSS64xx

DT6500
 DT6536
 KSS6380
 PCICardIF2004
 PCICardIF2008
 EthAdapter_IF2008
 LLT27xx
 CSP2008
 ETH_IF1032
 IMC5400
 IMC5600
 ACS7000
 CFO
 MFA
 USBAdapterIF2004
 C-Box
 thicknessSENSOR
 Unknown

Description: Type name of found sensor.

Parameter: int32_t IP_InterfaceIdx

IP_InterfaceIdx

Direction: Up

Valid values:

[RS232](#) (1)
[IF2004](#) (2)
[TCP/IP](#) (3)
[IF2008](#) (6)
[WinUSB](#) (7)
[IF2004_USB](#) (8)
[IF2008_ETH](#) (10)

Description: Index of the interface, on which the sensor was detected.

Parameter: String IP_Interface

IP_Interface

Direction: Up

Valid values:

RS232
 IF2004
 TCP/IP
 IF2008
 WinUSB
[IF2004_USB](#)
[IF2008_ETH](#)

Description: Name of the interface, on which the sensor was detected.

Parameter: String IP_Port

IP_Port

Direction: Up

Valid Interface:

[RS232](#)
[IF2004_USB](#)

Valid values:

"COM1"
 "COM2"

...

Description: Name of the serial interface.".

Parameter: int32_t IP_Baudrate	IP_Baudrate
Direction: Up	
Valid Interface:	
RS232	
IF2004_USB	
IF2008	
IF2008_ETH	
Valid values:	
Minimum: 0	
Unit: Baud	
Description: Baudrate of the serial connection.	
Parameter: int32_t IP_Stopbits	IP_Stopbits
Direction: Up	
Valid Interface:	
RS232	
Valid values:	
0= ONESTOPBIT	
1= ONE5STOPBITS	
2= TWOSTOPBITS	
Description: Number of stop bits of the serial connection.	
Parameter: int32_t IP_Parity	IP_Parity
Direction: Up	
Valid Interface:	
RS232	
IF2004_USB	
IF2008	
Valid values:	
0= NOPARITY	
1= ODDPARITY (only at RS232)	
2= EVENPARITY	
3= MARKPARITY (only at RS232)	
4= SPACEPARITY (only at RS232)	
Description: Parity of the serial connection.	
Parameter: int32_t IP_ByteSize	IP_ByteSize
Direction: Up	
Valid Interface:	
RS232	
Valid values:	
7	
8	
Unit: Bit	
Description: Number of data bits per byte of the serial connection.	
Parameter: int32_t IP_SensorAddress	IP_SensorAddress
Direction: Up	
Valid Interface:	
RS232	
Valid values:	
Minimum: 0	
Maximum: 126	
Description: If the interface is RS485 (with RS232 software emulation), up to 32 slave devices (sensors) can be connected. Thus is the sensor address.	

Parameter: int32_t IP_CardInstance IP_CardInstance

Direction: Up

Valid Interface:

IF2004

IF2008

Valid values:

Minimum: 0

Maximum: 15

Description: Instance number of the IF2004 or IF2008 Interface card.

Parameter: int32_t IP_ChannelNumber IP_ChannelNumber

Direction: Up

Valid Interface:

IF2004

Valid values:

Minimum: -1

Maximum: 3

Description: Channel number on IF2004 Interface card. If [PCI_CARD_IF2004](#) is searched, the channel number -1 and 3 is returned.

Parameter: int32_t IP_Baudrate IP_Baudrate

Direction: Up

Valid Interface:

IF2004

Valid values:

SENSOR_ILD1302, SENSOR_ILD1402, SENSOR_ILD1700, SENSOR_ILR2250

and SENSOR_MFA, SENSOR_ODC2500 and SENSOR_ODC2600:

115200

SENSOR_ODC2500 and SENSOR_ODC2600: 691200

SENSOR_ILD2200: 691200 or 1250000

SENSOR_ILD1220, SENSOR_ILD1320, SENSOR_ILD1420, SENSOR_ILD1750,

SENSOR_ILD1900, SENSOR_ILD2300, SENSOR_IFD2421, SENSOR_IFD2422,

SENSOR_IFD2445, SENSOR_IFD2451, SENSOR_IFD2461, SENSOR_IFD2471,

SENSOR_ODC2520, SENSOR_ACS7000, CONTROLLER_CSP2008

and CONTROLLER_CBOX: 115200 or 691200

Unit: Baud

Description: Speed of the RS422 serial connection.

Parameter: int32_t IP_DeviceInstance IP_DeviceInstance

Direction: Up

Valid Interface:

IF2004_USB

WinUSB

Valid values:

Minimum: 0

Maximum: 255

Description: Instance number of the USB device.

Parameter: String IP_SerialNumber IP_SerialNumber

Direction: Up

Valid Interface:

IF2004_USB

Valid values:

Minimum: "0000001"

Maximum: "9999999"

Description: Serial number of USB adapter IF2004.

Parameter: int32_t IP_ChannelNumber	IP_ChannelNumber
Direction: Up	
Valid Interface:	
IF2004_USB	
Valid values:	
Minimum: -1	
Maximum: 4	
Description:	Channel number on USB adapter IF2004. If no sensor is found, the channel number is from 0 to 3, if USB_ADAPTER_IF2004 is searched, channel -1 and 4 is returned.
Parameter: String SA_UUID	SA_UUID
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Description:	Universal identifier of the sensor, if supported by sensor.
Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Description:	Name of the sensor, if supported by sensor.
Parameter: String SA_Manufacturer	SA_Manufacturer
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Description:	Manufacturer of the sensor, if supported by sensor.
Parameter: String SA_MACAddress	SA_MACAddress
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Description:	MAC address of the sensor, if known.
Parameter: String SA_CurrentIPAddresses	SA_CurrentIPAddresses
Direction: Up	
Valid Interface:	
TCP/IP	
Description:	Space separated list of IP addresses currently used by sensor in form of xx.xx.xx.xx/yy (yy are number of subnet bits).
Parameter: String IP_RemoteAddr	IP_RemoteAddr
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Description:	IP address of the sensor, extracted from answer packet.

Parameter: String SA_Address SA_Address

Direction: Up

Valid Interface:

TCP/IP

IF2008_ETH

Description: IP address of the sensor, returned by sensor. This parameter is output only if it differs to IP_RemoteAddr.

Parameter: String SA_SubnetMask SA_SubnetMask

Direction: Up

Valid Interface:

TCP/IP

IF2008_ETH

Description: IPv4 subnet mask of the sensor. 0.0.0.0 means standard netmask for class A, B, C is used.

Parameter: String SA_Gateway SA_Gateway

Direction: Up

Valid Interface:

TCP/IP

IF2008_ETH

Description: Default gateway of the sensor.

Parameter: int32_t SA_StaticIP SA_StaticIP

Direction: Up

Valid Interface:

TCP/IP

Valid values:

0= FALSE

1= TRUE

Description: Sensor has a static IP address.

Parameter: int32_t SA_AutoIPEnabled SA_AutoIPEnabled

Direction: Up

Valid Interface:

TCP/IP

Valid values:

0= FALSE

1= TRUE

Description: AutolP mode of the sensor.

Parameter: int32_t SA_DHCPEnabled SA_DHCPEnabled

Direction: Up

Valid Interface:

TCP/IP

IF2008_ETH

Valid values:

-1= Not supported by sensor

0= FALSE

1= TRUE

Description: DHCP mode of the sensor.

Parameter: int32_t SA_BootPEnabled	SA_BootPEnabled
Direction: Up	
Valid Interface:	
TCP/IP	
Valid values:	
0 = FALSE	
1 = TRUE	
Description: BootP mode of the sensor.	
Parameter: String SA_DHCPName	SA_DHCPName
Direction: Up	
Valid Interface:	
TCP/IP	
Description: Name which is used by sensor for registering at DHCP server.	
Parameter: String SA_DNSServer	SA_DNSServer
Direction: Up	
Valid Interface:	
TCP/IP	
Description: DNS server address of sensor.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Description: Software version of sensor.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Description: Serial number of sensor.	
Parameter: int32_t SA_AddressFamily	SA_AddressFamily
Direction: Up	
Valid Interface:	
TCP/IP	
Valid values:	
2 = AF_INET (IPv4)	
23 = AF_INET6 (IPv6)	
Description: TCP/IP address family, on which the sensor answer was received.	
Parameter: String SA_LocalIPAddress	SA_LocalIPAddress
Direction: Up	
Valid Interface:	
TCP/IP	
Description: IP address of local computer, on which the sensor answer was received. If the answer was received on multiple addresses, all addresses are set, separated by newline.	

Parameter: int32_t SA_NetworkBits	SA_NetworkBits
Direction: Up	
Valid Interface:	
TCP/IP	
Valid values:	
Minimum: 0	
Maximum: 32 (IPv4) or 128(IPv6)	
Description:	Number of network bits (prefix). Using this parameter the nework mask can be calculated.
	If the answer was received on multiple addresses, only first result is set.
Parameter: String SA_LocalSubnetMask	SA_LocalSubnetMask
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Description:	Subnet mask of local computer, on which the sensor answer was received. This parameter is only available on IPv4 networks.
	The number of subnet masks (separated by newline) is equal to the number at SA_LocalIPAddress.
Parameter: String SA_LocalDefaultGateway	SA_LocalDefaultGateway
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Description:	Default gateways of local computer (comma separated list), on which the sensor answer was received.
	The number of lists of default gateways (separated by newline) is equal to the number at SA_LocalIPAddress.
Parameter: int32_t SA_LocaldHCPEnabled	SA_LocaldHCPEnabled
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Valid values:	
Specific bit is 0= not enabled	
Specific bit is 1= enabled	
Description:	Shows if DHCP (or alternativly AutoIP) is enabled for the local IP address. The value is a bit field, bit 0 corresponds to the first local IP address, and so on.
Parameter: String SA_DHCPServer	SA_DHCPServer
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Description:	IP address of DHCP server, which was leasing the local IP address.
	The number of DHCP servers (separated by newline) is equal to the number at SA_LocalIPAddress.

Parameter: String SA_LocalAdapterDescription	SA_LocalAdapterDescription
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Description: The name of the local network adapter which owns the local IP address.	
The number of adapter descriptions (separated by newline) is equal to the number at SA_LocalIPAddress.	
Parameter: int32_t SA_SensorAccessible	SA_SensorAccessible
Direction: Up	
Valid Interface:	
TCP/IP	
IF2008_ETH	
Valid values:	
Specific bit is 0= not accessible	
Specific bit is 1= accessible	
Description: Sensor and local IP addresses and subnet masks are checked to determine if sensor is accessible by TCP/IP. The value is a bit field, bit 0 corresponds to the first local IP address, and so on.	
Parameter: int32_t IP_ChannelNumber	IP_ChannelNumber
Direction: Up	
Valid Interface:	
IF2008	
Valid values:	
Minimum: -1	
Maximum: 11	
Description: Channel number on IF2008 Interface card. If an sensor is searched, the channel number is from 0 to 5, if PCI_CARD_IF2008 is searched, entries for all valid channels from 6 to 11 (and -1) are returned.	
Parameter: int32_t IP_ChannelNumber	IP_ChannelNumber
Direction: Up	
Valid Interface:	
IF2008_ETH	
Valid values:	
Minimum: -1	
Maximum: 8	
Description: Channel number on IF2008_ETH ethernet adapter.	

Additionally a sensor command to get more sensor information is called. It depends on sensor and hardware interface, which command is called. Depending on the command, please refer to the specific command later in this documentation. Following table shows sensor, interface and command:

SENSOR_ILR110x_115x	any interface	Get_Version
SENSOR_ILR118x	any interface	Get_Info
SENSOR_ILR1191	any interface	Get_Info
SENSOR_ILR2250	any interface	Get_Info
SENSOR_ODC1202	any interface	Get_Version
SENSOR_ILD1220	any interface	Get_Info
SENSOR_ILD1302	any interface	Get_Info
SENSOR_ILD1320	any interface	Get_Info
SENSOR_ILD1401	any interface	Get_Info
SENSOR_ILD1402	any interface	Get_Info
SENSOR_ILD1420	any interface	Get_Info
SENSOR_ILD1700	any interface	Get_Info
SENSOR_ILD1750	any interface	Get_Info
SENSOR_ILD1900	any interface	Get_Info
SENSOR_ILD2200	any interface	Get_Info
SENSOR_ILD2300	any but TCP/IP	Get_Info
SENSOR_IFD2401	any interface	Get_Version
SENSOR_IFD2421	any but TCP/IP	Get_Info
SENSOR_IFD2422	any but TCP/IP	Get_Info
SENSOR_IFD2431	any interface	Get_Version
SENSOR_IFD2445	any but TCP/IP	Get_Info
SENSOR_IFD2451	any but TCP/IP	Get_Info
SENSOR_IFD2461	any but TCP/IP	Get_Info
SENSOR_IFD2471	any but TCP/IP	Get_Info
SENSOR_IMC5400	any but TCP/IP	Get_Info
SENSOR_IMC5600	any but TCP/IP	Get_Info
SENSOR_ACS7000	any but TCP/IP	Get_Info
SENSOR_MFA	any interface	Get_Info
SENSOR_ODC2500	any interface	Get_Info
SENSOR_ODC2520	any but TCP/IP	Get_Info
SENSOR_DT6120	any interface	Get_ControllerInfo
SENSOR_ODC2600	any interface	Get_Info
ETH_ADAPTER_IF2008	any interface	Get_Info
CONTROLLER_CBOX	any but TCP/IP	Get_Info

8.3.4 Abort_FindSensor

Abort an active scan.

8.3.5 Set_IPConfig

Set IP configuration of sensors with ethernet interface.

Parameter: int32_t IP_SensorType IP_SensorType

Direction: Down

Valid values:

- SENSOR_ILD2300 (29)
- SENSOR_IFD2421 (46)
- SENSOR_IFD2422 (47)
- SENSOR_IFD2445 (39)
- SENSOR_IFD2451 (30)
- SENSOR_IFD2461 (44)
- SENSOR_IFD2471 (26)
- SENSOR_ODC2520 (37)
- SENSOR_DT3060 (50)
- SENSOR_DT3100 (28)
- SENSOR_DT6100 (16)
- CONTROLLER_DT6200 (33)
- CONTROLLER_KSS64xx (45)
- CONTROLLER_DT6500 (15)
- CONTROLLER_DT6536 (54)
- CONTROLLER_KSS6380 (18)
- SENSOR_LLT27xx (31)
- CONTROLLER_CSP2008 (32)
- CONTROLLER_CBOX (38)
- THICKNESS_SENSOR (48)
- ETH_ADAPTER_IF2008 (52)
- ETH_IF1032 (34)
- SENSOR_IMC5400 (55)
- SENSOR_IMC5600 (59)
- SENSOR_ACS7000 (35)
- SENSOR_CFO (53)

Description: Type of sensor to set IP configuration.

Parameter: String IP_MACAddress IP_MACAddress

Direction: Down

Valid for sensor:

- SENSOR_DT3100 (28)
- SENSOR_DT6100 (16)
- CONTROLLER_DT6500 (15), not DT6530
- CONTROLLER_KSS6380 (18)
- SENSOR_LLT27xx (31)

Description: Mac address used to identify the sensor.

Parameter: String IP_OldIPAddress IP_OldIPAddress

Direction: Down

Valid for sensor:

- SENSOR_DT3100 (28)
- SENSOR_DT6100 (16)
- CONTROLLER_DT6500 (15), not DT6530
- CONTROLLER_KSS6380 (18)
- SENSOR_LLT27xx (31)

Description: Old IP address of sensor to be able to communicate with.

Parameter: String IP_UUID IP_UUID

Direction: Down

Valid for sensor:

- SENSOR_ILD2300 (29)
- SENSOR_IFD2421 (46)
- SENSOR_IFD2422 (47)
- SENSOR_IFD2445 (39)
- SENSOR_IFD2451 (30)
- SENSOR_IFD2461 (44)
- SENSOR_IFD2471 (26)
- SENSOR_ODC2520 (37)
- CONTROLLER_CSP2008 (32)
- CONTROLLER_CBOX (38)
- THICKNESS_SENSOR (48)
- SENSOR_DT3060 (50)
- CONTROLLER_DT6200 (33)
- CONTROLLER_KSS64xx (45)
- CONTROLLER_DT6500 (15), only DT6530
- CONTROLLER_DT6536 (54)
- ETH_ADAPTER_IF2008 (52)
- ETH_IF1032 (34)
- SENSOR_IMC5400 (55)
- SENSOR_IMC5600 (59)
- SENSOR_ACS7000 (35)
- SENSOR_CFO (53)

Description: UUID used to identify the sensor.

Parameter: String IP_Password IP_Password

Direction: Down

Valid for sensor:

- SENSOR_ILD2300 (29)
- SENSOR_IFD2421 (46)
- SENSOR_IFD2422 (47)
- SENSOR_IFD2445 (39)
- SENSOR_IFD2451 (30)
- SENSOR_IFD2461 (44)
- SENSOR_IFD2471 (26)
- SENSOR_ODC2520 (37)
- CONTROLLER_CSP2008 (32)
- SENSOR_DT3060 (50)
- CONTROLLER_DT6200 (33)
- CONTROLLER_KSS64xx (45)
- CONTROLLER_DT6500 (15), only DT6530
- CONTROLLER_DT6536 (54)
- SENSOR_IMC5400 (55)
- SENSOR_IMC5600 (59)
- SENSOR_ACS7000 (35)
- SENSOR_CFO (53)
- ETH_IF1032 (34)

Description: Password which allow changing IP configuration.

Parameter: String IP_NewIPAddress IP_NewIPAddress

Direction: Down

Valid for sensor: all

Description: New IP address to set at sensor.

Parameter: int32_t IP_AllowReserved	IP_AllowReserved
Direction: Down	
Valid for sensor: all	
Valid values:	
0= FALSE	
1= TRUE	
Default: 0	
Description: Sepcifies if new IP address may be in the reserverd range starting from 224.0.0.0 (Multicast, Class D, Class E). Not recommended, because sensor may no longer be accessible!	
Parameter: String IP_Gateway	IP_Gateway
Direction: Down	
Valid for sensor: all	
Description: New default gateway to set at sensor.	
Parameter: String IP_SubnetMask	IP_SubnetMask
Direction: Down	
Valid for sensor: all	
Description: New subnet mask to set at sensor.	
Parameter: int32_t IP_StaticIP	IP_StaticIP
Direction: Down	
Valid for sensor:	
SENSOR_LL27xx (31)	
Valid values:	
0= Static IP address should be set	
1= Do not set static IP address	
Description: Specify if sensor should use static IP address.	
Parameter: int32_t IP_DHCPEnabled	IP_DHCPEnabled
Direction: Down	
Valid for sensor: all	
Valid values:	
0= DHCP is disabled	
1= DHCP is enabled	
Description: Specify if sensor should use DHCP.	
Parameter: int32_t IP_AutoIPEnabled	IP_AutoIPEnabled
Direction: Down	
Valid for sensor:	
SENSOR_DT3100 (28)	
SENSOR_DT6100 (16)	
CONTROLLER_DT6500 (15), not DT6530	
CONTROLLER_KSS6380 (18)	
Valid values:	
0= AutoIP is disabled	
1= AutoIP is enabled	
Description: Specify if sensor should use AutoIP.	

Parameter: int32_t IP_BootPEnabled IP_BootPEnabled

Direction: Down

Valid for sensor:

SENSOR_DT3100 (28)
SENSOR_DT6100 (16)
CONTROLLER_DT6500 (15), not DT6530
CONTROLLER_KSS6380 (18)

Valid values:

0 = BootP is disabled
1 = BootP is enabled

Description: Specify if sensor should use BootP.

Parameter: int32_t IA_Success IA_Success

Direction: Up

Valid for sensor: all

Valid values:

0 = Operation failed
1 = Operation successful

Description: Returns result of Set_IPConfig operation. Some sensors may not return an acknowledge, in this case, success means that the command was send to sensor.

9 Commands for optoNCDT (ILR) sensors

9.1 Commands for ILR110x/5x

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (native)
- [TCP/IP](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface)
- [IF2004_USB](#) (native)
- [IF2008](#) (native)
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Parameters](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are already scaled at sensor.
- Scaled values are identical to raw values.

9.1.1 General commands

9.1.1.1 General

9.1.1.1.1 Get_Parameters (GAP)

Retrieve all parameters from the sensor.

Parameter: String SA_Version SA_Version
Direction: Up
Description: Firmware version.

Parameter: int32_t SA_PilotLaser SA_PilotLaser
Direction: Up
Valid values:
 0= off
 1= on
Description: Pilot laser behaviour.

Parameter: int32_t SA_SensorBaudrate	SA_SensorBaudrate
Direction: Up	
Valid values:	
4800	
9600	
19200	
38400	
57600	
Description: Sensor baudrate.	
Parameter: int32_t SA_SensorDatabits	SA_SensorDatabits
Direction: Up	
Valid values:	
7	
8	
Description: Sensor data bits.	
Parameter: int32_t SA_SensorStopbits	SA_SensorStopbits
Direction: Up	
Valid values:	
1	
2	
Description: Sensor stop bits.	
Parameter: int32_t SA_ContinuousMode	SA_ContinuousMode
Direction: Up	
Valid values:	
0= continuous	
1= single	
Description: Sensor is sending data continuous or only single values.	
Parameter: int32_t SA_Q1Value	SA_Q1Value
Direction: Up	
Valid values:	
0= low	
1= high	
Description: The actual state of output Q1.	
Parameter: int32_t SA_ModeQ1	SA_ModeQ1
Direction: Up	
Valid values:	
0= not active	
1= switching point	
2= switching points	
Description: The mode of output Q1.	
Parameter: int32_t SA_LimitQ1-1	SA_LimitQ1-1
Direction: Up	
Valid values:	
Minimum: -48000	
Maximum: 96000	
Description: The limit 1 of output Q1.	

Parameter: int32_t SA_LimitQ1-2	SA_LimitQ1-2
Direction: Up	
Valid values:	
Minimum: -48000	
Maximum: 96000	
Description: The limit 2 of output Q1.	
Parameter: int32_t SA_HysteresisQ1	SA_HysteresisQ1
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 999	
Description: The hysteresis of output Q1.	
Parameter: int32_t SA_NormQ1	SA_NormQ1
Direction: Up	
Valid values:	
0= normal	
1= inverted	
Description: Specifies if output Q1 is normal or inverted.	
Parameter: int32_t SA_Q2Value	SA_Q2Value
Direction: Up	
Valid values:	
0= low	
1= high	
Description: The actual state of output Q2.	
Parameter: int32_t SA_ModeQ2	SA_ModeQ2
Direction: Up	
Valid values:	
0= not active	
1= switching point	
2= switching points	
Description: The mode of output Q2.	
Parameter: int32_t SA_LimitQ2-1	SA_LimitQ2-1
Direction: Up	
Valid values:	
Minimum: -48000	
Maximum: 96000	
Description: The limit 1 of output Q2.	
Parameter: int32_t SA_LimitQ2-2	SA_LimitQ2-2
Direction: Up	
Valid values:	
Minimum: -48000	
Maximum: 96000	
Description: The limit 2 of output Q2.	
Parameter: int32_t SA_HysteresisQ2	SA_HysteresisQ2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 999	
Description: The hysteresis of output Q2.	

Parameter: int32_t SA_NormQ2	SA_NormQ2
Direction: Up	
Valid values:	
0= normal	
1= inverted	
Description: Specifies if output Q2 is normal or inverted.	
Parameter: int32_t SA_AnalogValue	SA_AnalogValue
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 4095	
Description: The actual value of analog output.	
Parameter: int32_t SA_LimitQA-1	SA_LimitQA-1
Direction: Up	
Valid values:	
Minimum: -48000	
Maximum: 96000	
Description: The limit 1 of analog output.	
Parameter: int32_t SA_LimitQA-2	SA_LimitQA-2
Direction: Up	
Valid values:	
Minimum: -48000	
Maximum: 96000	
Description: The limit 2 of analog output.	
Parameter: int32_t SA_NormQA	SA_NormQA
Direction: Up	
Valid values:	
0= normal	
1= inverted	
Description: Specifies if analog output is normal or inverted.	
Parameter: int32_t SA_OutputFormat	SA_OutputFormat
Direction: Up	
Valid values:	
0= mm	
1= inch*100	
Description: Output format of measured values.	
Parameter: int32_t SA_Offset	SA_Offset
Direction: Up	
Valid values:	
Minimum: -48000	
Maximum: 48000	
Description: Offset value.	
Parameter: int32_t SA_Password	SA_Password
Direction: Up	
Valid values:	
0= disabled	
1= enabled	
Description: Password protection for sensor menu.	

9.1. Commands for ILR110x/5x

Parameter: int32_t SA_ErrorStatus SA_ErrorStatus

Direction: Up

Valid values:

Minimum: 0x00

Maximum: 0xff

Description: Error status as 8 bit field.

00000000: no error.

00000010: PLL UNLOCKED - Counter error.

00000100: LOW VOLT - Error in supply voltage: voltage too low (or error in measurement of supply voltage).

00101000: OVERTEMP - Temperature too high (above 85 °C inside); Measurement switched off.

00010000: Dist (mm) >Maximum - No target in range or sensor badly aligne.

00100000: Temperature warning (below -10 °C or above 70 °C).

01000000: BLINDING - External light too strong or internal error.

10000000: LAS.ERR. - Measurement laser faulty.

9.1.1.1.2 Get_Energy (GDB)

Get the amount of receiving by sensor.

Parameter: int32_t SA_Energy SA_Energy

Direction: Up

Unit: dB

Valid values:

Minimum: -120

Maximum: 0

Description: Energy value.

9.1.1.1.3 Get_SerialNbr (GNR)

Get the serial number of sensor.

Parameter: String SA_SerialNumber SA_SerialNumber

Direction: Up

Description: Serial number.

9.1.1.1.4 Get_ErrorStatus (GSI)

Get error status from the sensor.

Parameter: int32_t SA_ErrorStatus SA_ErrorStatus

Direction: Up

Valid values:

Minimum: 0x00

Maximum: 0xff

9.1. Commands for ILR110x/5x

Description: Error status as 8 bit field.

- 00000000: no error.
- 00000010: PLL UNLOCKED - Counter error.
- 00000100: LOW VOLT - Error in supply voltage: voltage too low (or error in measurement of supply voltage).
- 00101000: OVERTEMP - Temperature too high (above 85 °C inside); Measurement switched off.
- 00010000: Dist (mm) >Maximum - No target in range or sensor badly aligne.
- 00100000: Temperature warning (below -10 °C or above 70 °C).
- 01000000: BLINDING - External light too strong or internal error.
- 10000000: LAS.ERR. - Measurement laser faulty.

9.1.1.5 Get_Temperature (GTE)

Retrieve the temperature inside of the sensor.

Parameter: int32_t SA_Temperature SA_Temperature
Direction: Up
Unit: °C
Description: Sensor temperature.

9.1.1.6 Get_Version (GVE)

Get the version of sensor firmware.

Parameter: String SA_Version SA_Version
Direction: Up
Description: Firmware version.

9.1.1.7 Set_Stand-by (ISB)

Set the sensor in stand-by mode or reactivates it.

Parameter: int32_t SP_Stand-by SP_Stand-by
Direction: Down
Valid values:
 0= operation
 1= stand-by
Description: Sensor stand-by mode.

9.1.1.8 Set_VisibleLaser (IVL)

Set the behaviour of the pilot laser of the sensor.

Parameter: int32_t SP_PilotLaser SP_PilotLaser
Direction: Down
Valid values:
 0= off
 1= on
Description: Pilot laser behaviour.

9.1. Commands for ILR110x/5x

9.1.1.2 Triggering

9.1.1.2.1 Set_ContinuousMode (ICM)

Set the measurement mode.

Parameter: int32_t SP_ContinuousMode

SP_ContinuousMode

Direction: Down

Valid values:

0= continuous

1= single

Description: Sensor is sending data continuous or only single values.

9.1.1.2.2 Exec_ContMeasure (ECM)

Continuous measurement output ist set and triggered by the next request for measured values (ESM).

9.1.1.2.3 Trg_SingleMeasure (ESM)

Request for measured value with single measurement output.

9.1.1.3 Parameter management

9.1.1.3.1 Save_Parameters (EPW)

Store all actual parameters in sensor memory.

9.1.2 Measurement

9.1.2.1 General

9.1.2.1.1 Set_Offset (IDO)

Set the offset which is added by the sensor to distance values.

Parameter: int32_t SP_Offset

SP_Offset

Direction: Down

Valid values:

Minimum: -12000 [mm] or -48000 [100*inch]

Maximum: 12000 [mm] or 48000 [100*inch]

Description: Offset value.

9.1.3 Data output

9.1.3.1 Switching outputs

9.1.3.1.1 Set_HysteresisQ1 (IH1)

Hysteresis setting around the switching point Q1 in [mm] or [100*inch].

Parameter: int32_t SP_HysteresisQ1

SP_HysteresisQ1

Direction: Down

Valid values:

Minimum: 0 [mm] / [100*inch]

Maximum: 254 [mm] or 999 [100*inch]

Description: Hysteresis Q1.

9.1.3.1.2 Set_HysteresisQ2 (IH2)

Hysteresis setting around the switching point Q2 in [mm] or [100*inch].

Parameter: int32_t SP_HysteresisQ2

SP_HysteresisQ2

Direction: Down

Valid values:

Minimum: 0 [mm] / [100*inch]

Maximum: 254 [mm] or 999 [100*inch]

Description: Hysteresis Q2.

9.1.3.1.3 Set_LimitQ1-1 (IL1)

Setting of the first switch point of Q1 in [mm] or [100*inch].

Parameter: int32_t SP_LimitQ1-1

SP_LimitQ1-1

Direction: Down

Valid values:

Minimum: Offset [mm] / [100*inch]

Maximum: 12000+Offset [mm] or 48000+Offset [100*inch]

Description: Limit Q1-1.

9.1.3.1.4 Set_LimitQ2-1 (IL2)

Setting of the first switch point of Q2 in [mm] or [100*inch].

Parameter: int32_t SP_LimitQ2-1

SP_LimitQ2-1

Direction: Down

Valid values:

Minimum: Offset [mm] / [100*inch]

Maximum: 12000+Offset [mm] or 48000+Offset [100*inch]

Description: Limit Q2-1.

9.1.3.1.5 Set_LimitQ1-2 (IL4)

Setting of the second switch point of Q1 in [mm] or [100*inch].

Parameter: int32_t SP_LimitQ1-2 SP_LimitQ1-2

Direction: Down

Valid values:

Minimum: Offset [mm] / [100*inch]

Maximum: 12000+Offset [mm] or 48000+Offset [100*inch]

Description: Limit Q1-2.

9.1.3.1.6 Set_LimitQ2-2 (IL5)

Setting of the second switch point of Q2 in [mm] or [100*inch].

Parameter: int32_t SP_LimitQ2-2 SP_LimitQ2-2

Direction: Down

Valid values:

Minimum: Offset [mm] / [100*inch]

Maximum: 12000+Offset [mm] or 48000+Offset [100*inch]

Description: Limit Q2-2.

9.1.3.1.7 Set_ModeQ1 (IM1)

Set the mode of output Q1.

Parameter: int32_t SP_ModeQ1 SP_ModeQ1

Direction: Down

Valid values:

0= not active

1= switching point

2= switching points

Description: The mode of output Q1.

9.1.3.1.8 Set_ModeQ2 (IM2)

Set the mode of output Q2.

Parameter: int32_t SP_ModeQ2 SP_ModeQ2

Direction: Down

Valid values:

0= not active

1= switching point

2= switching points

Description: The mode of output Q2.

9.1. Commands for ILR110x/5x

9.1.3.1.9 Set_NormQ1 (IN1)

Set the norm of output Q1.

Parameter: int32_t SP_NormQ1

SP_NormQ1

Direction: Down

Valid values:

0 = normal

1 = inverted

Description: Specifies if output Q1 is normal or inverted.

9.1.3.1.10 Set_NormQ2 (IN2)

Set the norm of output Q2.

Parameter: int32_t SP_NormQ2

SP_NormQ2

Direction: Down

Valid values:

0 = normal

1 = inverted

Description: Specifies if output Q2 is normal or inverted.

9.1.3.2 Analog output

9.1.3.2.1 Set_LimitQA-1 (IL3)

Setting of the 0% point of the analog characteristic. Only valid for sensors ILR1100 and ILR1150 (with analog output).

Parameter: int32_t SP_LimitQA-1

SP_LimitQA-1

Direction: Down

Valid values:

Minimum: Offset

Maximum: 12000+Offset

Description: Limit QA-1.

9.1.3.2.2 Set_LimitQA-2 (IL6)

Setting of the 100% point of the analog characteristic. Only valid for sensors ILR1100 and ILR1150 (with analog output).

Parameter: int32_t SP_LimitQA-2

SP_LimitQA-2

Direction: Down

Valid values:

Minimum: Offset

Maximum: 12000+Offset

Description: Limit QA-2.

9.2. Commands for ILR118x

9.1.3.2.3 Set_NormQA (INA)

Set the norm of analog output. Only valid for sensors ILR1100 and ILR1150 (with analog output).

Parameter: int32_t SP_NormQA

SP_NormQA

Direction: Down

Valid values:

0= normal

1= inverted

Description: Specifies if analog output is normal or inverted.

9.2 Commands for ILR118x

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (native).
- [TCP/IP](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_AllParameters](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking and to calculate datarate.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are already scaled at sensor.
- Scaled values are identical to raw values, except error values are scaled depending of [IP_ScaleErrorValues](#).

9.2.1 General commands

9.2.1.1 General

9.2.1.1.1 Set_Autostart (AS)

Set which function will be carried out when power becomes available to the sensor.

Parameter: int32_t SP_AutostartCommand

SP_AutostartCommand

Direction: Down

Valid values:

0= DT

1= DS

2= DW

3= DX

4= DF

5= DM

6= TP

7= ID

8= LO

Description: Autostart command.

9.2. Commands for ILR118x

Parameter: int32_t SA_AutostartCommand SA_AutostartCommand

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= DT
- 1= DS
- 2= DW
- 3= DX
- 4= DF
- 5= DM
- 6= TP
- 7= ID
- 8= LO

Description: Adapted value from sensor if parameter to set was invalid.

9.2.1.1.2 Get_Autostart (AS)

Get which function will be carried out when power becomes available to the sensor.

Parameter: int32_t SA_AutostartCommand SA_AutostartCommand

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= DT
- 1= DS
- 2= DW
- 3= DX
- 4= DF
- 5= DM
- 6= TP
- 7= ID
- 8= LO

Description: Autostart command.

9.2.1.1.3 Get_Info (ID)

Retrieve information (like serial number) of the sensor.

Parameter: String SA_Version SA_Version

Direction: Up

Description: Sensor name, range, serial number and version

9.2.1.1.4 Get_AllParameters (PA)

Retrieve all parameters from the sensor.

Parameter: int32_t SA_Average SA_Average

Direction: Up

Valid values:

- Minimum:** 1
- Maximum:** 20

Description: Average value.

9.2. Commands for ILR118x

Parameter: int32_t SA_OutputFormat	SA_OutputFormat
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= decimal distance	
1= hex distance	
3= decimal (distance + signal strength), since firmware version 7.02	
Description: Output format of values.	
Parameter: int32_t SA_MeasureTime	SA_MeasureTime
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 25	
Description: Measure time index.	
Parameter: double SA_ScaleFactor	SA_ScaleFactor
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Scaling factor.	
Parameter: int32_t SA_ErrorMode	SA_ErrorMode
Direction: Up	
Valid values:	
0= last valid value	
1= switch to bounds	
2= switch to negated bounds	
Description: Error mode.	
Parameter: double SA_AlarmStart	SA_AlarmStart
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Alarm start.	
Parameter: double SA_AlarmHysteresis	SA_AlarmHysteresis
Direction: Up	
Valid values:	
Minimum: -100000	
Maximum: 100000	
Description: Alarm hysteresis.	
Parameter: double SA_AlarmWidth	SA_AlarmWidth
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Alarm width.	

Parameter: double SA_RangeBegin	SA_RangeBegin
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Range begin.	
Parameter: double SA_RangeEnd	SA_RangeEnd
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Range end.	
Parameter: int32_t SA_PrecedingValues	SA_PrecedingValues
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10	
Description: Designates the number of preceding measuring values that will be evaluated in the case of non-conforming measurement.	
Parameter: double SA_ValidRange	SA_ValidRange
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Defines the range of permissible values. If this range is exceeded in negative or positive direction, the respective measuring value will be corrected accordingly.	
Parameter: int32_t SA_InvalidValues	SA_InvalidValues
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 100	
Description: Stands for the number of values that are out of the permissible value range; in the event of out-of-tolerance values arriving in succession, the most recently corrected value will be included in the correction process for the next out-of-tolerance value.	
Parameter: int32_t SA_TriggerDelay	SA_TriggerDelay
Direction: Up	
Unit: ms	
Valid values:	
Minimum: 0	
Maximum: 9999	
Description: Trigger delay.	
Parameter: int32_t SA_TriggerEdge	SA_TriggerEdge
Direction: Up	
Valid values:	
0= falling	
1= rising	
Description: Trigger edge.	

Parameter: int32_t SA_AutostartTrigger	SA_AutostartTrigger
Direction: Up	
Valid values:	
0= off	
1= on	
Description: Autostart trigger.	
Parameter: int32_t SA_AutostartEdge	SA_AutostartEdge
Direction: Up	
Valid values:	
0= falling	
1= rising	
Description: Autostart trigger edge.	
Parameter: int32_t SA_SensorBaudrate	SA_SensorBaudrate
Direction: Up	
Valid values:	
2400	
4800	
9600	
19200	
38400	
Description: Sensor baudrate.	
Parameter: int32_t SA_AutostartCommand	SA_AutostartCommand
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= DT	
1= DS	
2= DW	
3= DX	
4= DF	
5= DM	
6= TP	
7= ID	
8= LO	
Description: Autostart command.	
Parameter: double SA_Offset	SA_Offset
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Offset value which is set.	

9.2.1.5 Get_Temperature (TP)

Retrieve the temperature inside of the sensor.

Parameter: double SA_Temperature	SA_Temperature
Direction: Up	
Unit: °C	
Description: Sensor temperature.	

9.2. Commands for ILR118x

9.2.1.1.6 Laser_Off (LF)

Switch the laser off.

9.2.1.1.7 Laser_On (LO)

Switch the laser on.

9.2.1.2 Tracking

9.2.1.2.1 DistanceTracking (DT)

Start distance tracking mode.

9.2.1.2.2 DistanceTracking7m (DS)

Start distance tracking (7 m) mode.

9.2.1.2.3 DistanceTracking10Hz (DW)

Start distance tracking (10 Hz) mode.

Valid for sensor:

ILR1181

9.2.1.2.4 DistanceTracking50Hz (DX)

Start distance tracking (50 Hz) mode.

Valid for sensor:

ILR1182

9.2.1.2.5 StopTracking (<ESC>)

Stop any tracking mode.

9.2.1.3 Triggering

9.2.1.3.1 DistanceTriggered (DF)

Start distance tracking (with external trigger) mode.

9.2.1.3.2 DistanceMeasure (DM)

Measure one distance value.

9.2.1.3.3 Set_MeasureTime (ST)

Set an index for measure time of one distance value.

Parameter: int32_t SP_MeasureTime SP_MeasureTime

Direction: Down

Valid values:

Minimum: 0

Maximum: 25

Description: Measure time index.

Parameter: int32_t SA_MeasureTime SA_MeasureTime

Direction: Up

Valid values:

Minimum: 0

Maximum: 25

Description: Adapted value from sensor if parameter to set was invalid.

9.2.1.3.4 Get_MeasureTime (ST)

Get an index for measure time of one distance value.

Parameter: int32_t SA_MeasureTime SA_MeasureTime

Direction: Up

Valid values:

Minimum: 0

Maximum: 25

Description: Measure time index.

9.2.1.3.5 Set_TriggerDelay (TD)

Set the behaviour of the trigger input.

Parameter: int32_t SP_TriggerDelay SP_TriggerDelay

Direction: Down

Unit: ms

Valid values:

Minimum: 0

Maximum: 9999

Description: Trigger delay.

Parameter: int32_t SP_TriggerEdge SP_TriggerEdge

Direction: Down

Valid values:

0= falling

1= rising

Description: Trigger edge.

9.2. Commands for ILR118x

Parameter: int32_t SA_TriggerDelay SA_TriggerDelay

Direction: Up

Unit: ms

Valid values:

Minimum: 0

Maximum: 9999

Description: Adapted value from sensor if parameter to set was invalid.

Parameter: int32_t SA_TriggerEdge SA_TriggerEdge

Direction: Up

Valid values:

0= falling

1= rising

Description: Adapted value from sensor if parameter to set was invalid.

9.2.1.3.6 Get_TriggerDelay (TD)

Get the behaviour of the trigger input.

Parameter: int32_t SA_TriggerDelay SA_TriggerDelay

Direction: Up

Unit: ms

Valid values:

Minimum: 0

Maximum: 9999

Description: Trigger delay.

Parameter: int32_t SA_TriggerEdge SA_TriggerEdge

Direction: Up

Valid values:

0= falling

1= rising

Description: Trigger edge.

9.2.1.3.7 Set_TriggerMode (TM)

Set parameters for the auto-start trigger function which allows external triggering of the auto-start command that was set via parameter AS.

Parameter: int32_t SP_AutostartTrigger SP_AutostartTrigger

Direction: Down

Valid values:

0= off

1= on

Description: Autostart trigger.

Parameter: int32_t SP_AutostartEdge SP_AutostartEdge

Direction: Down

Valid values:

0= falling

1= rising

Description: Autostart trigger edge.

Parameter: int32_t SA_AutostartTrigger SA_AutostartTrigger

Direction: Up

Valid values:

- 0= off
- 1= on

Description: Adapted value from sensor if parameter to set was invalid.

Parameter: int32_t SA_AutostartEdge SA_AutostartEdge

Direction: Up

Valid values:

- 0= falling
- 1= rising

Description: Adapted value from sensor if parameter to set was invalid.

9.2.1.3.8 Get_TriggerMode (TM)

Get parameters for the auto-start trigger function which allows external triggering of the auto-start command that was set via parameter AS.

Parameter: int32_t SA_AutostartTrigger SA_AutostartTrigger

Direction: Up

Valid values:

- 0= off
- 1= on

Description: Autostart trigger.

Parameter: int32_t SA_AutostartEdge SA_AutostartEdge

Direction: Up

Valid values:

- 0= falling
- 1= rising

Description: Autostart trigger edge.

9.2.1.4 Interfaces

9.2.1.4.1 Set_Baudrate (BR)

Set the baudrate of the sensors serial interface. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Attention! To avoid synchronization problems, sending data by sensor should be disabled first (sensor command StopTracking).

Parameter: int32_t SP_SensorBaudrate SP_SensorBaudrate

Direction: Down

Valid values:

- 2400
- 4800
- 9600
- 19200
- 38400

Description: Sensor baudrate.

9.2. Commands for ILR118x

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate
Direction: Up
Valid values:
 2400
 4800
 9600
 19200
 38400
Description: Adapted value from sensor if parameter to set was invalid.

9.2.1.4.2 Get_Baudrate (BR)

Get the baudrate of the sensors serial interface.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate
Direction: Up
Valid values:
 2400
 4800
 9600
 19200
 38400
Description: Sensor baudrate.

9.2.1.5 Parameter management

9.2.1.5.1 Reset_Parameters (PR)

Reset all parameters of sensor to factory defaults and return new parameters.

Parameter: int32_t SA_Average SA_Average
Direction: Up
Valid values:
Minimum: 1
Maximum: 20
Description: Average value.

Parameter: int32_t SA_OutputFormat SA_OutputFormat
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = decimal distance
 1 = hex distance
 3 = decimal (distance + signal strength), since firmware version 7.02
Description: Output format of values.

Parameter: int32_t SA_MeasureTime SA_MeasureTime
Direction: Up
Valid values:
Minimum: 0
Maximum: 25
Description: Measure time index.

9.2. Commands for ILR118x

Parameter: double SA_ScaleFactor	SA_ScaleFactor
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Scaling factor.	
Parameter: int32_t SA_ErrorMode	SA_ErrorMode
Direction: Up	
Valid values:	
0= last valid value	
1= switch to bounds	
2= switch to negated bounds	
Description: Error mode.	
Parameter: double SA_AlarmStart	SA_AlarmStart
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Alarm start.	
Parameter: double SA_AlarmHysteresis	SA_AlarmHysteresis
Direction: Up	
Valid values:	
Minimum: -100000	
Maximum: 100000	
Description: Alarm hysteresis.	
Parameter: double SA_AlarmWidth	SA_AlarmWidth
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Alarm width.	
Parameter: double SA_RangeBegin	SA_RangeBegin
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Range begin.	
Parameter: double SA_RangeEnd	SA_RangeEnd
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Range end.	
Parameter: int32_t SA_PrecedingValues	SA_PrecedingValues
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10	
Description: Designates the number of preceding measuring values that will be evaluated in the case of non-conforming measurement.	

9.2. Commands for ILR118x

Parameter: double SA_ValidRange	SA_ValidRange
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Defines the range of permissible values. If this range is exceeded in negative or positive direction, the respective measuring value will be corrected accordingly.	
Parameter: int32_t SA_InvalidValues	SA_InvalidValues
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 100	
Description: Stands for the number of values that are out of the permissible value range; in the event of out-of-tolerance values arriving in succession, the most recently corrected value will be included in the correction process for the next out-of-tolerance value.	
Parameter: int32_t SA_TriggerDelay	SA_TriggerDelay
Direction: Up	
Unit: ms	
Valid values:	
Minimum: 0	
Maximum: 9999	
Description: Trigger delay.	
Parameter: int32_t SA_TriggerEdge	SA_TriggerEdge
Direction: Up	
Valid values:	
0= falling	
1= rising	
Description: Trigger edge.	
Parameter: int32_t SA_AutostartTrigger	SA_AutostartTrigger
Direction: Up	
Valid values:	
0= off	
1= on	
Description: Autostart trigger.	
Parameter: int32_t SA_AutostartEdge	SA_AutostartEdge
Direction: Up	
Valid values:	
0= falling	
1= rising	
Description: Autostart trigger edge.	
Parameter: int32_t SA_SensorBaudrate	SA_SensorBaudrate
Direction: Up	
Valid values:	
2400	
4800	
9600	
19200	
38400	
Description: Sensor baudrate.	

Parameter: int32_t SA_AutostartCommand SA_AutostartCommand

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= DT
- 1= DS
- 2= DW
- 3= DX
- 4= DF
- 5= DM
- 6= TP
- 7= ID
- 8= LO

Description: Autostart command.

Parameter: double SA_Offset SA_Offset

Direction: Up

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)

Description: Offset value which is set.

9.2.2 Measurement

9.2.2.1 Measurement value processing

9.2.2.1.1 Set_ScaleFactor (SF)

Set the scaling factor how the sensor scale distance values.

Parameter: double SP_ScaleFactor

SP_ScaleFactor

Direction: Down

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)

Description: Scaling factor.

Parameter: double SA_ScaleFactor

SA_ScaleFactor

Direction: Up

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)

Description: Adapted value from sensor if parameter to set was invalid.

9.2.2.1.2 Get_ScaleFactor (SF)

Get the scaling factor how the sensor scale distance values.

Parameter: double SA_ScaleFactor

SA_ScaleFactor

Direction: Up

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)

Description: Scaling factor.

9.2.2.1.3 Set_Offset (OF)

Set the offset which is added by the sensor to distance values.

Parameter: double SP_Offset SP_Offset

Direction: Down

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Offset value.

Parameter: double SA_Offset SA_Offset

Direction: Up

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Adapted value from sensor if parameter to set was invalid.

9.2.2.1.4 Get_Offset (OF)

Get the offset which is added by the sensor to distance values.

Parameter: double SA_Offset SA_Offset

Direction: Up

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Offset value.

9.2.2.1.5 CurrentDistAsOffset (SO)

Set the current distance value as offset.

Parameter: double SA_Offset SA_Offset

Direction: Up

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Offset value which is set.

9.2.2.1.6 Set_AverageValue (SA)

Set the average value for floating averaging.

Parameter: int32_t SP_Average SP_Average

Direction: Down

Valid values:

Minimum: 1

Maximum: 20

Description: Average value.

9.2. Commands for ILR118x

Parameter: int32_t SA_Average SA_Average
Direction: Up
Valid values:
Minimum: 1
Maximum: 20
Description: Adapted value from sensor if parameter to set was invalid.

9.2.2.1.7 Get_AverageValue (SA)

Get the average value for floating averaging.

Parameter: int32_t SA_Average SA_Average
Direction: Up
Valid values:
Minimum: 1
Maximum: 20
Description: Average value.

9.2.3 Data output

9.2.3.1 General

9.2.3.1.1 Set_OutputFormat (SD)

Set the output format how values are sent from sensor.

Parameter: int32_t SP_OutputFormat SP_OutputFormat
Direction: Down
Valid values:
 0= decimal distance
 1= hex distance
 3= decimal (distance + signal strength), since firmware version 7.02
Description: Output format of values.
 Mode binary is currently not supported by MEDAQLib. So if it is selected, no values can be read.

Parameter: int32_t SA_OutputFormat SA_OutputFormat
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= decimal distance
 1= hex distance
 3= decimal (distance + signal strength), since firmware version 7.02
Description: Adapted value from sensor if parameter to set was invalid.

9.2.3.1.2 Get_OutputFormat (SD)

Get the output format how values are sent from sensor.

Parameter: int32_t SA_OutputFormat	SA_OutputFormat
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = decimal distance	
1 = hex distance	
3 = decimal (distance + signal strength), since firmware version 7.02	
Description: Output format of values.	

9.2.3.1.3 Set_RemovalMeasVal (RM)

Set how invalid measure values should be treated by the sensor.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_PrecedingValues	SP_PrecedingValues
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 10	
Description: Designates the number of preceding measuring values that will be evaluated in the case of non-conforming measurement.	

Parameter: double SP_ValidRange	SP_ValidRange
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Defines the range of permissible values. If this range is exceeded in negative or positive direction, the respective measuring value will be corrected accordingly.	

Parameter: int32_t SP_InvalidValues	SP_InvalidValues
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 100	
Description: Stands for the number of values that are out of the permissible value range; in the event of out-of-tolerance values arriving in succession, the most recently corrected value will be included in the correction process for the next out-of-tolerance value.	

Parameter: int32_t SA_PrecedingValues	SA_PrecedingValues
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10	
Description: Adapted value from sensor if parameter to set was invalid.	

Parameter: double SA_ValidRange SA_ValidRange

Direction: Up

Valid values:

Minimum: 0

Maximum: 3.40282e+38 (FLT_MAX)

Description: Adapted value from sensor if parameter to set was invalid.

Parameter: int32_t SA_InvalidValues SA_InvalidValues

Direction: Up

Valid values:

Minimum: 0

Maximum: 100

Description: Adapted value from sensor if parameter to set was invalid.

9.2.3.1.4 Get_RemovalMeasVal (RM)

Get how invalid measure values should be treated by the sensor.

Parameter: int32_t SA_PrecedingValues SA_PrecedingValues

Direction: Up

Valid values:

Minimum: 0

Maximum: 10

Description: Designates the number of preceding measuring values that will be evaluated in the case of non-conforming measurement.

Parameter: double SA_ValidRange SA_ValidRange

Direction: Up

Valid values:

Minimum: 0

Maximum: 3.40282e+38 (FLT_MAX)

Description: Defines the range of permissible values. If this range is exceeded in negative or positive direction, the respective measuring value will be corrected accordingly.

Parameter: int32_t SA_InvalidValues SA_InvalidValues

Direction: Up

Valid values:

Minimum: 0

Maximum: 100

Description: Stands for the number of values that are out of the permissible value range; in the event of out-of-tolerance values arriving in succession, the most recently corrected value will be included in the correction process for the next out-of-tolerance value.

9.2.3.1.5 Set_ErrorMode (SE)

Set the behaviour of digital and analog outputs in case of an error.

Parameter: int32_t SP_ErrorMode SP_ErrorMode

Direction: Down

Valid values:

0= last valid value

1= switch to bounds

2= switch to negated bounds

Description: Error mode.

Parameter: int32_t SA_ErrorMode SA_ErrorMode

Direction: Up

Valid values:

- 0= last valid value
- 1= switch to bounds
- 2= switch to negated bounds

Description: Adapted value from sensor if parameter to set was invalid.

9.2.3.1.6 Get_ErrorMode (SE)

Get the behaviour of digital and analog outputs in case of an error.

Parameter: int32_t SA_ErrorMode SA_ErrorMode

Direction: Up

Valid values:

- 0= last valid value
- 1= switch to bounds
- 2= switch to negated bounds

Description: Error mode.

9.2.3.2 Switching outputs

9.2.3.2.1 Set_AlarmStart (AC)

Sets the beginning of the distance range, for which the switching output will be turned active.

Parameter: double SP_AlarmStart SP_AlarmStart

Direction: Down

Valid values:

- Minimum:** -3.40282e+38 (-FLT_MAX)
- Maximum:** 3.40282e+38 (FLT_MAX)

Description: Alarm start.

Parameter: double SA_AlarmStart SA_AlarmStart

Direction: Up

Valid values:

- Minimum:** -3.40282e+38 (-FLT_MAX)
- Maximum:** 3.40282e+38 (FLT_MAX)

Description: Adapted value from sensor if parameter to set was invalid.

9.2.3.2.2 Get_AlarmStart (AC)

Gets the beginning of the distance range, for which the switching output will be turned active.

Parameter: double SA_AlarmStart SA_AlarmStart

Direction: Up

Valid values:

- Minimum:** -3.40282e+38 (-FLT_MAX)
- Maximum:** 3.40282e+38 (FLT_MAX)

Description: Alarm start.

9.2.3.2.3 Set_AlarmHysteresis (AH)

Set the switching hysteresis at the beginning and the end point of the active range of the switching output.

Parameter: double SP_AlarmHysteresis SP_AlarmHysteresis

Direction: Down

Valid values:

Minimum: -100000

Maximum: 100000

Description: Alarm hysteresis.

Parameter: double SA_AlarmHysteresis SA_AlarmHysteresis

Direction: Up

Valid values:

Minimum: -100000

Maximum: 100000

Description: Adapted value from sensor if parameter to set was invalid.

9.2.3.2.4 Get_AlarmHysteresis (AH)

Get the switching hysteresis at the beginning and the end point of the active range of the switching output.

Parameter: double SA_AlarmHysteresis SA_AlarmHysteresis

Direction: Up

Valid values:

Minimum: -100000

Maximum: 100000

Description: Alarm hysteresis.

9.2.3.2.5 Set_AlarmWidth (AW)

Set the length of the active range for the switching output.

Parameter: double SP_AlarmWidth SP_AlarmWidth

Direction: Down

Valid values:

Minimum: 0

Maximum: 3.40282e+38 (FLT_MAX)

Description: Alarm width.

Parameter: double SA_AlarmWidth SA_AlarmWidth

Direction: Up

Valid values:

Minimum: 0

Maximum: 3.40282e+38 (FLT_MAX)

Description: Adapted value from sensor if parameter to set was invalid.

9.2.3.2.6 Get_AlarmWidth (AW)

Get the length of the active range for the switching output.

Parameter: double SA_AlarmWidth SA_AlarmWidth
Direction: Up
Valid values:
Minimum: 0
Maximum: 3.40282e+38 (FLT_MAX)
Description: Alarm width.

9.2.3.3 Analog output

9.2.3.3.1 Set_RangeBegin4mA (RB)

Set the starting point of the distance range that is provided at the analog output.

Parameter: double SP_RangeBegin SP_RangeBegin
Direction: Down
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Range begin.

Parameter: double SA_RangeBegin SA_RangeBegin
Direction: Up
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Adapted value from sensor if parameter to set was invalid.

9.2.3.3.2 Get_RangeBegin4mA (RB)

Get the starting point of the distance range that is provided at the analog output.

Parameter: double SA_RangeBegin SA_RangeBegin
Direction: Up
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Range begin.

9.2.3.3.3 Set_RangeEnd20mA (RE)

Set the end point of the distance range that is provided at the analog output.

Parameter: double SP_RangeEnd SP_RangeEnd
Direction: Down
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Range end.

9.3. Commands for ILR1191

Parameter: double SA_RangeEnd SA_RangeEnd
Direction: Up
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Adapted value from sensor if parameter to set was invalid.

9.2.3.3.4 Get_RangeEnd20mA (RE)

Get the end point of the distance range that is provided at the analog output.

Parameter: double SA_RangeEnd SA_RangeEnd
Direction: Up
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Range end.

9.3 Commands for ILR1191

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (native).
- [TCP/IP](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_AllParameters](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate and to assign values.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are already scaled at sensor.
- Scaled values are identical to raw values, except error values are scaled depending of [IP_ScaleErrorValues](#).

9.3.1 General commands

9.3.1.1 General

9.3.1.1.1 Set_Autostart (AS)

Set which function will be carried out when power becomes available to the sensor.

Parameter: int32_t SP_AutostartCommand

SP_AutostartCommand

Direction: Down

Valid values:

- 0= ID
- 1= DM
- 2= DT
- 3= DF
- 4= VM
- 5= VT
- 6= TP
- 7= HW
- 8= PA
- 9= MF
- 10= TD
- 11= SA
- 12= SF
- 13= MW
- 14= OF
- 15= SE
- 16= Q1
- 17= Q2
- 18= QA
- 19= BR
- 20= SD
- 21= TE
- 22= BB
- 23= AB
- 24= SC
- 25= PL
- 26= AS

Description: Autostart command.

Parameter: int32_t SA_AutostartCommand

SA_AutostartCommand

Direction: Up

Valid values:

- 0= ID
- 1= DM
- 2= DT
- 3= DF
- 4= VM
- 5= VT
- 6= TP
- 7= HW
- 8= PA
- 9= MF

```

 10= TD
 11= SA
 12= SF
 13= MW
 14= OF
 15= SE
 16= Q1
 17= Q2
 18= QA
 19= BR
 20= SD
 21= TE
 22= BB
 23= AB
 24= SC
 25= PL
 26= AS
 -1= unknown
  
```

Description: Autostart command.

9.3.1.1.2 Get_Autostart (AS)

Get which function will be carried out when power becomes available to the sensor.

Parameter: int32_t SA_AutostartCommand

SA_AutostartCommand

Direction: Up

Valid values:

```

 0= ID
 1= DM
 2= DT
 3= DF
 4= VM
 5= VT
 6= TP
 7= HW
 8= PA
 9= MF
 10= TD
 11= SA
 12= SF
 13= MW
 14= OF
 15= SE
 16= Q1
 17= Q2
 18= QA
 19= BR
 20= SD
 21= TE
 22= BB
 23= AB
 24= SC
  
```

9.3. Commands for ILR1191

25= PL

26= AS

-1= unknown

Description: Autostart command.

9.3.1.1.3 Set_PilotLaser (PL)

Set the behaviour of the pilot laser of the sensor.

Parameter: int32_t SP_PilotLaser SP_PilotLaser

Direction: Down

Valid values:

0= off

1= on

2= flashing (2 Hz)

3= flashing (5 Hz)

Description: Pilot laser behaviour.

Parameter: int32_t SA_PilotLaser SA_PilotLaser

Direction: Up

Valid values:

0= off

1= on

2= flashing (2 Hz)

3= flashing (5 Hz)

Description: Pilot laser behaviour.

9.3.1.1.4 Get_PilotLaser (PL)

Get the behaviour of the pilot laser of the sensor.

Parameter: int32_t SA_PilotLaser SA_PilotLaser

Direction: Up

Valid values:

0= off

1= on

2= flashing (2 Hz)

3= flashing (5 Hz)

Description: Pilot laser behaviour.

9.3.1.1.5 Get_Info (ID)

Retrieve information (like serial number) of the sensor.

Parameter: String SA_Version SA_Version

Direction: Up

Description: Firmware version.

9.3.1.1.6 Get_AllParameters (PA)

Retrieve all parameters from the sensor.

Parameter: int32_t SA_MeasFrequency	SA_MeasFrequency
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2000	
Description: Measure frequency.	
Parameter: double SA_TriggerDelay	SA_TriggerDelay
Direction: Up	
Unit: ms	
Valid values:	
Minimum: 0.0	
Maximum: 314.15	
Description: Trigger delay.	
Parameter: int32_t SA_TriggerEdge	SA_TriggerEdge
Direction: Up	
Valid values:	
0= falling	
1= rising	
Description: Trigger edge.	
Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
0= single	
1= measure	
Description: Specifies if one laser shot should be done or single measuring should be started.	
Parameter: int32_t SA_Average	SA_Average
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 30000	
Description: Average value.	
Parameter: double SA_ScaleFactor	SA_ScaleFactor
Direction: Up	
Valid values:	
Minimum: -10.0	
Maximum: 10.0	
Description: Scaling factor.	
Parameter: double SA_WindowMin	SA_WindowMin
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Window minimum value.	

Parameter: double SA_WindowMax	SA_WindowMax
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Window maximum value.	
Parameter: double SA_Offset	SA_Offset
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Offset value which is set.	
Parameter: int32_t SA_ErrorMode	SA_ErrorMode
Direction: Up	
Valid values:	
0= last valid value	
1= switch to bounds	
2= switch to negated bounds	
Description: Error mode.	
Parameter: double SA_Q1Start	SA_Q1Start
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Start value.	
Parameter: double SA_Q1Width	SA_Q1Width
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Width.	
Parameter: double SA_Q1Hysteresis	SA_Q1Hysteresis
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Hysteresis.	
Parameter: int32_t SA_Q1Negation	SA_Q1Negation
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Negation.	
Parameter: double SA_Q2Start	SA_Q2Start
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Start value.	

Parameter: double SA_Q2Width	SA_Q2Width
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Width.	
Parameter: double SA_Q2Hysteresis	SA_Q2Hysteresis
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Hysteresis.	
Parameter: int32_t SA_Q2Negation	SA_Q2Negation
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Negation.	
Parameter: double SA_RangeBegin	SA_RangeBegin
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Lower limit value.	
Parameter: double SA_RangeEnd	SA_RangeEnd
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Upper limit value.	
Parameter: int32_t SA_SensorBaudrate	SA_SensorBaudrate
Direction: Up	
Valid values:	
9600	
19200	
38400	
57600	
115200	
230400	
460800	
Description: Sensor baudrate.	
Parameter: int32_t SA_OutputFormat	SA_OutputFormat
Direction: Up	
Valid values:	
0= decimal	
1= hex	
2= binary	
Description: Output format of values.	

Parameter: int32_t SA_OutputContent	SA_OutputContent
Direction: Up	
Valid values:	
0= Measuring value	
1= Measuring value, signal strength	
2= Measuring value, sensor temperature	
3= Measuring value, signal strength, sensor temperature	
Description: Get which data is transmitted by sensor.	
Parameter: int32_t SA_TerminationChar	SA_TerminationChar
Direction: Up	
Valid values:	
0= <CRLF>	
1= <CR>	
2= <LF>	
3= <STX>	
4= <ETX>	
5= Tabulator	
6= Space	
7= Comma	
8= Colon	
9= Semicolon	
Description: Termination character.	
Parameter: int32_t SA_SSIFormat	SA_SSIFormat
Direction: Up	
Valid values:	
0= binary	
1= grey code	
Description: SSI transmission format.	
Parameter: int32_t SA_PilotLaser	SA_PilotLaser
Direction: Up	
Valid values:	
0= off	
1= on	
2= flashing (2 Hz)	
3= flashing (5 Hz)	
Description: Pilot laser behaviour.	
Parameter: int32_t SA_AutostartCommand	SA_AutostartCommand
Direction: Up	
Valid values:	
0= ID	
1= DM	
2= DT	
3= DF	
4= VM	
5= VT	
6= TP	
7= HW	
8= PA	
9= MF	
10= TD	

9.3. Commands for ILR1191

```

11= SA
12= SF
13= MW
14= OF
15= SE
16= Q1
17= Q2
18= QA
19= BR
20= SD
21= TE
22= BB
23= AB
24= SC
25= PL
26= AS
-1= unknown

```

Description: Autostart command.

9.3.1.1.7 Get_Temperature (TP)

Retrieve the temperature inside of the sensor.

Parameter: double SA_Temperature

SA_Temperature

Direction: Up

Unit: °C

Description: Sensor temperature.

9.3.1.1.8 Get_HWDiagnosis (HW)

Retrieve internal sensor diagnostic information.

Parameter: String SA_Diagnosis

SA_Diagnosis

Direction: Up

Description: Sensor diagnostic information.

9.3.1.1.9 Trigger_ColdStart (DR)

Reboots the sensor and executes the autostart command.

9.3.1.2 Tracking

9.3.1.2.1 DistanceTracking (DT)

Start distance tracking mode.

9.3.1.2.2 SpeedTracking (VT)

Start speed (velocity) tracking mode.

9.3.1.2.3 StopTracking (<ESC>)

Stop any tracking mode.

9.3.1.3 Triggering

9.3.1.3.1 DistanceTriggered (DF)

Start distance tracking (with external trigger) mode.

9.3.1.3.2 DistanceMeasure (DM)

Measure one distance value.

9.3.1.3.3 SpeedMeasure (VM)

Measure one speed (velocity) value.

Attention! To leave that mode with StopTracking, there is a delay of average value (SA) / measure frequency (MF) seconds.

For example, if average value is 20 and measure frequency is 10 Hz, it takes 2 seconds to leave this mode. In extreme case, this could be 30000/1 seconds, more than 8 hours.

9.3.1.3.4 Set_TriggerDelay (TD)

Set the behaviour of the trigger input.

For this command an [Update ...](#) meta command is available.

Parameter: double SP_TriggerDelay

SP_TriggerDelay

Direction: Down

Unit: ms

Valid values:

Minimum: 0.0

Maximum: 314.15

Description: Trigger delay.

Parameter: int32_t SP_TriggerEdge

SP_TriggerEdge

Direction: Down

Valid values:

0= falling

1= rising

Description: Trigger edge.

Parameter: int32_t SP_TriggerMode

SP_TriggerMode

Direction: Down

Valid values:

0= single

1= measure

Description: Specifies if one laser shot should be done or single measuring should be started.

Parameter: double SA_TriggerDelay	SA_TriggerDelay
Direction: Up	
Unit: ms	
Valid values:	
Minimum: 0.0	
Maximum: 314.15	
Description: Trigger delay.	
Parameter: int32_t SA_TriggerEdge	SA_TriggerEdge
Direction: Up	
Valid values:	
0= falling	
1= rising	
Description: Trigger edge.	
Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
0= single	
1= measure	
Description: Specifies if one laser shot should be done or single measuring should be started.	

9.3.1.3.5 Get_TriggerDelay (TD)

Get the behaviour of the trigger input.

Parameter: double SA_TriggerDelay	SA_TriggerDelay
Direction: Up	
Unit: ms	
Valid values:	
Minimum: 0.0	
Maximum: 314.15	
Description: Trigger delay.	
Parameter: int32_t SA_TriggerEdge	SA_TriggerEdge
Direction: Up	
Valid values:	
0= falling	
1= rising	
Description: Trigger edge.	
Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
0= single	
1= measure	
Description: Specifies if one laser shot should be done or single measuring should be started.	

9.3.1.4 Interfaces

9.3.1.4.1 Set_Baudrate (BR)

Set the baudrate of the sensors serial interface. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Attention! If baudrate is set to 9600, the command ID? (currently set to supported) will timeout.

If Set_Autostart is set to ID?, the sensor reboots cyclic and must be send back to manufacturer.

Attention! If baudrate is set to a baudrate, which the computer does not support (e.g. 230400 or 460800), it cannot be changed back and must be send back to manufacturer.

Attention! To avoid synchronization problems, sending data by sensor should be disabled first (sensor command StopTracking).

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

- 9600
- 19200
- 38400
- 57600
- 115200
- 230400
- 460800

Description: Sensor baudrate.

Parameter: int32_t SA_SensorBaudrate

SA_SensorBaudrate

Direction: Up

Valid values:

- 9600
- 19200
- 38400
- 57600
- 115200
- 230400
- 460800

Description: Sensor baudrate.

9.3.1.4.2 Get_Baudrate (BR)

Get the baudrate of the sensors serial interface.

Parameter: int32_t SA_SensorBaudrate

SA_SensorBaudrate

Direction: Up

Valid values:

- 9600
- 19200
- 38400
- 57600
- 115200
- 230400
- 460800

Description: Sensor baudrate.

9.3.1.5 Parameter management

9.3.1.5.1 Reset Parameters (PR)

Reset all parameters of sensor to factory defaults and return new parameters.

Parameter: int32_t SA_MeasFrequency	SA_MeasFrequency
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2000	
Description: Measure frequency.	
Parameter: double SA_TriggerDelay	SA_TriggerDelay
Direction: Up	
Unit: ms	
Valid values:	
Minimum: 0.0	
Maximum: 314.15	
Description: Trigger delay.	
Parameter: int32_t SA_TriggerEdge	SA_TriggerEdge
Direction: Up	
Valid values:	
0= falling	
1= rising	
Description: Trigger edge.	
Parameter: int32_t SA_Average	SA_Average
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 30000	
Description: Average value.	
Parameter: double SA_ScaleFactor	SA_ScaleFactor
Direction: Up	
Valid values:	
Minimum: -10.0	
Maximum: 10.0	
Description: Scaling factor.	
Parameter: double SA_WindowMin	SA_WindowMin
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Window minimum value.	
Parameter: double SA_WindowMax	SA_WindowMax
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Window maximum value.	

Parameter: double SA_Offset	SA_Offset
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Offset value which is set.	
Parameter: int32_t SA_ErrorMode	SA_ErrorMode
Direction: Up	
Valid values:	
0= last valid value	
1= switch to bounds	
2= switch to negated bounds	
Description: Error mode.	
Parameter: double SA_Q1Start	SA_Q1Start
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Start value.	
Parameter: double SA_Q1Width	SA_Q1Width
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Width.	
Parameter: double SA_Q1Hysteresis	SA_Q1Hysteresis
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Hysteresis.	
Parameter: int32_t SA_Q1Negation	SA_Q1Negation
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Negation.	
Parameter: double SA_Q2Start	SA_Q2Start
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Start value.	
Parameter: double SA_Q2Width	SA_Q2Width
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Width.	

9.3. Commands for ILR1191

Parameter: double SA_Q2Hysteresis	SA_Q2Hysteresis
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Hysteresis.	
Parameter: int32_t SA_Q2Negation	SA_Q2Negation
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Negation.	
Parameter: double SA_RangeBegin	SA_RangeBegin
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Lower limit value.	
Parameter: double SA_RangeEnd	SA_RangeEnd
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Upper limit value.	
Parameter: int32_t SA_SensorBaudrate	SA_SensorBaudrate
Direction: Up	
Valid values:	
9600	
19200	
38400	
57600	
115200	
230400	
460800	
Description: Sensor baudrate.	
Parameter: int32_t SA_OutputFormat	SA_OutputFormat
Direction: Up	
Valid values:	
0= decimal	
1= hex	
2= binary	
Description: Output format of values.	
Parameter: int32_t SA_OutputContent	SA_OutputContent
Direction: Up	
Valid values:	
0= Measuring value	
1= Measuring value, signal strength	
2= Measuring value, sensor temperature	
3= Measuring value, signal strength, sensor temperature	
Description: Get which data is transmitted by sensor.	

Parameter: int32_t SA_TerminationChar	SA_TerminationChar
Direction: Up	
Valid values:	
0= <CRLF>	
1= <CR>	
2= <LF>	
3= <STX>	
4= <ETX>	
5= Tabulator	
6= Space	
7= Comma	
8= Colon	
9= Semicolon	
Description: Termination character.	
Parameter: int32_t SA_SSIFormat	SA_SSIFormat
Direction: Up	
Valid values:	
0= binary	
1= grey code	
Description: SSI transmission format.	
Parameter: int32_t SA_PilotLaser	SA_PilotLaser
Direction: Up	
Valid values:	
0= off	
1= on	
2= flashing (2 Hz)	
3= flashing (5 Hz)	
Description: Pilot laser behaviour.	
Parameter: int32_t SA_AutostartCommand	SA_AutostartCommand
Direction: Up	
Valid values:	
0= ID	
1= DM	
2= DT	
3= DF	
4= VM	
5= VT	
6= TP	
7= HW	
8= PA	
9= MF	
10= TD	
11= SA	
12= SF	
13= MW	
14= OF	
15= SE	
16= Q1	
17= Q2	
18= QA	
19= BR	

20= SD
 21= TE
 22= BB
 23= AB
 24= SC
 25= PL
 26= AS
 -1= unknown

Description: Autostart command.

9.3.2 Measurement

9.3.2.1 General

9.3.2.1.1 Set_MeasFreq (MF)

Set the measure frequency of the sensor.

Parameter: int32_t SP_MeasFrequency

SP_MeasFrequency

Direction: Down

Valid values:

Minimum: 1

Maximum: 2000

Description: Measure frequency.

Parameter: int32_t SA_MeasFrequency

SA_MeasFrequency

Direction: Up

Valid values:

Minimum: 1

Maximum: 2000

Description: Measure frequency.

9.3.2.1.2 Get_MeasFreq (MF)

Get the measure frequency of the sensor.

Parameter: int32_t SA_MeasFrequency

SA_MeasFrequency

Direction: Up

Valid values:

Minimum: 1

Maximum: 2000

Description: Measure frequency.

9.3.2.2 Measurement value processing

9.3.2.2.1 Set_ScaleFactor (SF)

Set the scaling factor how the sensor scale distance values.

Parameter: double SP_ScaleFactor

SP_ScaleFactor

Direction: Down

Valid values:

Minimum: -10.0

Maximum: 10.0

Description: Scaling factor.

9.3. Commands for ILR1191

Parameter: double SA_ScaleFactor SA_ScaleFactor
Direction: Up
Valid values:
Minimum: -10.0
Maximum: 10.0
Description: Scaling factor.

9.3.2.2.2 Get_ScaleFactor (SF)

Get the scaling factor how the sensor scale distance values.

Parameter: double SA_ScaleFactor SA_ScaleFactor
Direction: Up
Valid values:
Minimum: -10.0
Maximum: 10.0
Description: Scaling factor.

9.3.2.2.3 Set_Offset (OF)

Set the offset which is added by the sensor to distance values.

Parameter: double SP_Offset SP_Offset
Direction: Down
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Offset value.

Parameter: double SA_Offset SA_Offset
Direction: Up
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Offset value.

9.3.2.2.4 Get_Offset (OF)

Get the offset which is added by the sensor to distance values.

Parameter: double SA_Offset SA_Offset
Direction: Up
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Offset value.

9.3.2.2.5 CurrentDistAsOffset (SO)

Set the current distance value as offset.

Parameter: double SA_Offset

SA_Offset

Direction: Up

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Offset value which is set.

9.3.3 Data output

9.3.3.1 General

9.3.3.1.1 Set_OutputFormat (SD)

Set the output format how values are sent from sensor.

Parameter: int32_t SP_OutputFormat

SP_OutputFormat

Direction: Down

Valid values:

0= decimal

1= hex

2= binary

Description: Output format of values.

Parameter: int32_t SP_OutputContent

SP_OutputContent

Direction: Down

Valid values:

0= Measuring value

1= Measuring value, signal strength

2= Measuring value, sensor temperature

3= Measuring value, signal strength, sensor temperature

Description: Set which data is transmitted by sensor.

Parameter: int32_t SA_OutputFormat

SA_OutputFormat

Direction: Up

Valid values:

0= decimal

1= hex

2= binary

Description: Output format of values.

Parameter: int32_t SA_OutputContent

SA_OutputContent

Direction: Up

Valid values:

0= Measuring value

1= Measuring value, signal strength

2= Measuring value, sensor temperature

3= Measuring value, signal strength, sensor temperature

Description: Get which data is transmitted by sensor.

9.3.3.1.2 Get_OutputFormat (SD)

Get the output format how values are sent from sensor.

Parameter: int32_t SA_OutputFormat

SA_OutputFormat

Direction: Up

Valid values:

- 0= decimal
- 1= hex
- 2= binary

Description: Output format of values.

Parameter: int32_t SA_OutputContent

SA_OutputContent

Direction: Up

Valid values:

- 0= Measuring value
- 1= Measuring value, signal strength
- 2= Measuring value, sensor temperature
- 3= Measuring value, signal strength, sensor temperature

Description: Get which data is transmitted by sensor.

9.3.3.1.3 Set_TerminatingChar (TE)

Set the termination character of each measurement.

Parameter: int32_t SP_TerminationChar

SP_TerminationChar

Direction: Down

Valid values:

- 0= <CRLF>
- 1= <CR>
- 2= <LF>
- 3= <STX>
- 4= <ETX>
- 5= Tabulator
- 6= Space
- 7= Comma
- 8= Colon
- 9= Semicolon

Description: Termination character.

Parameter: int32_t SA_TerminationChar

SA_TerminationChar

Direction: Up

Valid values:

- 0= <CRLF>
- 1= <CR>
- 2= <LF>
- 3= <STX>
- 4= <ETX>
- 5= Tabulator
- 6= Space
- 7= Comma
- 8= Colon
- 9= Semicolon

Description: Termination character.

9.3.3.1.4 Get_TerminatingChar (TE)

Get the termination character of each measurement.

Parameter: int32_t SA_TerminationChar

SA_TerminationChar

Direction: Up

Valid values:

- 0= <CRLF>
- 1= <CR>
- 2= <LF>
- 3= <STX>
- 4= <ETX>
- 5= Tabulator
- 6= Space
- 7= Comma
- 8= Colon
- 9= Semicolon

Description: Termination character.

9.3.3.1.5 Set_ErrorMode (SE)

Set the behaviour of digital and analog outputs in case of an error.

Parameter: int32_t SP_ErrorMode

SP_ErrorMode

Direction: Down

Valid values:

- 0= last valid value
- 1= switch to bounds
- 2= switch to negated bounds

Description: Error mode.

Parameter: int32_t SA_ErrorMode

SA_ErrorMode

Direction: Up

Valid values:

- 0= last valid value
- 1= switch to bounds
- 2= switch to negated bounds

Description: Error mode.

9.3.3.1.6 Get_ErrorMode (SE)

Get the behaviour of digital and analog outputs in case of an error.

Parameter: int32_t SA_ErrorMode

SA_ErrorMode

Direction: Up

Valid values:

- 0= last valid value
- 1= switch to bounds
- 2= switch to negated bounds

Description: Error mode.

9.3.3.1.7 Set_MeasureWindow (MW)

Set a metrological range by definition of a starting point x and an end point as limits for output of measured values.

Parameter: double SP_WindowMin SP_WindowMin

Direction: Down

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Window minimum value.

Parameter: double SP_WindowMax SP_WindowMax

Direction: Down

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Window maximum value.

Parameter: double SA_WindowMin SA_WindowMin

Direction: Up

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Window minimum value.

Parameter: double SA_WindowMax SA_WindowMax

Direction: Up

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Window maximum value.

9.3.3.1.8 Get_MeasureWindow (MW)

Get a metrological range by definition of a starting point x and an end point as limits for output of measured values.

Parameter: double SA_WindowMin SA_WindowMin

Direction: Up

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Window minimum value.

Parameter: double SA_WindowMax SA_WindowMax

Direction: Up

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Window maximum value.

9.3.3.1.9 Set_AverageValue (SA)

Set the average value for block wise averaging.

Parameter: int32_t SP_Average SP_Average

Direction: Down

Valid values:

Minimum: 1

Maximum: 30000

Description: Average value.

Parameter: int32_t SA_Average SA_Average

Direction: Up

Valid values:

Minimum: 1

Maximum: 30000

Description: Average value.

9.3.3.1.10 Get_AverageValue (SA)

Get the average value for block wise averaging.

Parameter: int32_t SA_Average SA_Average

Direction: Up

Valid values:

Minimum: 1

Maximum: 30000

Description: Average value.

9.3.3.2 Switching outputs

9.3.3.2.1 Set_Out1Parameters (Q1)

Set parameters of the switching outputs Q1.

For this command an [Update_...](#) meta command is available.

Parameter: double SP_Q1Start SP_Q1Start

Direction: Down

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Start value.

Parameter: double SP_Q1Width SP_Q1Width

Direction: Down

Valid values:

Minimum: -3.40282e+38 (-FLT_MAX)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Width.

Parameter: double SP_Q1Hysteresis	SP_Q1Hysteresis
Direction: Down	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Hysteresis.	
Parameter: int32_t SP_Q1Negation	SP_Q1Negation
Direction: Down	
Valid values:	
0= false	
1= true	
Description: Negation.	
Parameter: double SA_Q1Start	SA_Q1Start
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Start value.	
Parameter: double SA_Q1Width	SA_Q1Width
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Width.	
Parameter: double SA_Q1Hysteresis	SA_Q1Hysteresis
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Hysteresis.	
Parameter: int32_t SA_Q1Negation	SA_Q1Negation
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Negation.	

9.3.3.2.2 Get_Out1Parameters (Q1)

Get parameters of the switching outputs Q1.

Parameter: double SA_Q1Start	SA_Q1Start
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Start value.	

Parameter: double SA_Q1Width	SA_Q1Width
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Width.	
Parameter: double SA_Q1Hysteresis	SA_Q1Hysteresis
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Hysteresis.	
Parameter: int32_t SA_Q1Negation	SA_Q1Negation
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Negation.	

9.3.3.2.3 Set_Out2Parameters (Q2)

Set parameters of the switching outputs Q2.
 For this command an [Update...](#) meta command is available.

Parameter: double SP_Q2Start	SP_Q2Start
Direction: Down	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Start value.	
Parameter: double SP_Q2Width	SP_Q2Width
Direction: Down	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Width.	
Parameter: double SP_Q2Hysteresis	SP_Q2Hysteresis
Direction: Down	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Hysteresis.	
Parameter: int32_t SP_Q2Negation	SP_Q2Negation
Direction: Down	
Valid values:	
0= false	
1= true	
Description: Negation.	

Parameter: double SA_Q2Start	SA_Q2Start
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Start value.	
Parameter: double SA_Q2Width	SA_Q2Width
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Width.	
Parameter: double SA_Q2Hysteresis	SA_Q2Hysteresis
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Hysteresis.	
Parameter: int32_t SA_Q2Negation	SA_Q2Negation
Direction: Up	
Valid values:	
0= false	
1= true	
Description: Negation.	

9.3.3.2.4 Get_Out2Parameters (Q2)

Get parameters of the switching outputs Q2.

Parameter: double SA_Q2Start	SA_Q2Start
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Start value.	
Parameter: double SA_Q2Width	SA_Q2Width
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Width.	
Parameter: double SA_Q2Hysteresis	SA_Q2Hysteresis
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Hysteresis.	

Parameter: int32_t SA_Q2Negation SA_Q2Negation
Direction: Up
Valid values:
 0= false
 1= true
Description: Negation.

9.3.3.3 Analog output

9.3.3.3.1 Set_AnalogOutLimits (QA)

Set parameters of the analog output QA .

Parameter: double SP_RangeBegin SP_RangeBegin
Direction: Down
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Lower limit value.

Parameter: double SP_RangeEnd SP_RangeEnd
Direction: Down
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Upper limit value.

Parameter: double SA_RangeBegin SA_RangeBegin
Direction: Up
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Lower limit value.

Parameter: double SA_RangeEnd SA_RangeEnd
Direction: Up
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Upper limit value.

9.3.3.3.2 Get_AnalogOutLimits (QA)

Get parameters of the analog output QA.

Parameter: double SA_RangeBegin SA_RangeBegin
Direction: Up
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Lower limit value.

9.4. Commands for ILR2250

Parameter: double SA_RangeEnd SA_RangeEnd
Direction: Up
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Description: Upper limit value.

9.3.3.4 SSI

9.3.3.4.1 Set_FormatSSI (SC)

Set the transmission format of SSI output.

Parameter: int32_t SP_SSIFormat SP_SSIFormat
Direction: Down
Valid values:
 0= binary
 1= grey code
Description: SSI transmission format.

Parameter: int32_t SA_SSIFormat SA_SSIFormat
Direction: Up
Valid values:
 0= binary
 1= grey code
Description: SSI transmission format.

9.3.3.4.2 Get_FormatSSI (SC)

Get the transmission format of SSI output.

Parameter: int32_t SA_SSIFormat SA_SSIFormat
Direction: Up
Valid values:
 0= binary
 1= grey code
Description: SSI transmission format.

9.4 Commands for ILR2250

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

9.4. Commands for ILR2250

If first bit of `IP_AutomaticMode` is set (1), MEDAQLib calls automatically sensor command `Get_AllParameters` (`SP_Additional= 1`) after `OpenSensor`.

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate, to interpret and scale data and to assign values. If second bit of `IP_AutomaticMode` is set (2), MEDAQLib calls optionally sensor command `Set_DataOutInterface`, `Get_LaserPower` and optionally `Set_LaserPower` at `OpenSensor`.

Meaning of raw and scaled values (function `Poll` and `TransferData`):

- Raw values are as it comes directly from sensor.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command `Get_AllParameters` (`SP_Additional= 1`)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

9.4.1 General commands

9.4.1.1 General

9.4.1.1.1 Get_Help (HELP)

Retrieve a help text from sensor for a specific command.

Parameter: String `SP_Command` `SP_Command`
Direction: Down
Valid values:
 "" (empty string, means general help)
 or any command name
Description: Name of the command.

Parameter: String `SA_HelpText` `SA_HelpText`
Direction: Up
Description: Help text to the command.

9.4.1.1.2 Get_Info (GETINFO)

Retrieve information about the sensor.

Parameter: String `SA_Sensor` `SA_Sensor`
Direction: Up
Description: Name of the sensor.

Parameter: String `SA_SerialNumber` `SA_SerialNumber`
Direction: Up
Valid values:
 Numeric value
Description: Serial number of the sensor.

9.4. Commands for ILR2250

Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the sensor.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the sensor.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Unit: mm	
Description: Range of the sensor.	

9.4.1.1.3 Get_Temperature (GETTEMP)

Get temperature of controller.

Parameter: double SA_Temperature	SA_Temperature
Direction: Up	
Unit: °C	
Description: Temperature.	

9.4.1.1.4 Get_OutputInfo_RS422 (GETOUTINFO_RS422)

Retrieve information which data is output at RS422 interface.

Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputDistance1_RS422	SA_OutputDistance1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 is transmitted.	

9.4.1.1.5 Reset_Boot (RESET)

Resets the sensor.

At this command the sensor may change output data after reboot. If first bit of **IP_AutomaticMode** is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

9.4.1.1.6 Reset_Counter (RESETCNT)

Resets sensor counter values.

Parameter: int32_t SP_ResetTimestamp

SP_ResetTimestamp

Direction: Down

Valid values:

1 = Yes

Description: Reset timestamp value.

9.4.1.1.7 Get_AllParameters (PRINT)

Get all parameters from sensor.

Parameter: int32_t SP_Additional

SP_Additional

Direction: Down

Valid values:

0 = No

1 = Yes

Description: If set, additional information about sensor is output.

Parameter: int32_t SA_TriggerMode

SA_TriggerMode

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor

0 = None

1 = Edge

2 = Level (PULSE)

3 = Software

Description: Trigger mode.

Parameter: int32_t SA_TriggerMoment

SA_TriggerMoment

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor

0 = Input

1 = Output

Description: Trigger moment.

Parameter: int32_t SA_TriggerLevel

SA_TriggerLevel

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor

0 = High

1 = Low

Description: Trigger level.

Parameter: int32_t SA_TriggerCount

SA_TriggerCount

Direction: Up

Valid values:

Minimum: 1

Maximum: 2147483647 (INT32_MAX)

Description: Number of values to measure. 2147483647 means endless measurement (INFINITE).

9.4. Commands for ILR2250

Parameter: int32_t SA_SensorBaudrate	SA_SensorBaudrate
Direction: Up	
Valid values:	
9600	
115200	
230400	
Unit: Baud	
Description: Baudrate of sensor.	
Parameter: int32_t SA_LaserPower	SA_LaserPower
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Off	
1 = On	
2 = Measure	
Description: Laser power.	
Parameter: double SA_ROIStart	SA_ROIStart
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 150000.0	
Unit: mm	
Description: First position in region of interest.	
Parameter: double SA_ROIEnd	SA_ROIEnd
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 150000.0	
Unit: mm	
Description: Last position in region of interest.	
Parameter: int32_t SA_DataOutInterface	SA_DataOutInterface
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = None	
1 = RS422	
7 = RS422_ASCII (does not work with MEDAQLib)	
Description: Active interface for data output.	
Parameter: int32_t SA_HoldLastValid	SA_HoldLastValid
Direction: Up	
Valid values:	
Minimum: -1	
Maximum: 2147483646	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	

9.4. Commands for ILR2250

Parameter: int32_t SA_OutputDistance1_RS422	SA_OutputDistance1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_ErrorOutput1..3	SA_ErrorOutput1..3
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Output is disabled (NONE)	
1= Switch when result is outside bounds of analog range (TEACH)	
2= Switch on excess of ERRORLIMIT1..3 (LIMIT)	
Description: Condition for error output.	
Parameter: int32_t SA_ErrorLevelOut1..3	SA_ErrorLevelOut1..3
Direction: Up	
Valid values:	
0= NPN	
1= PNP	
2= Push-Pull (PUSHPULL)	
3= Push-Pull negated (PUSHPULLNEG)	
Description: Error level for error output.	
Parameter: double SA_ErrorHysteresis1..3	SA_ErrorHysteresis1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 150000.0	
Unit: mm	
Description: Error hysteresis.	
Parameter: int32_t SA_ErrorLimitCompOp1..3	SA_ErrorLimitCompOp1..3
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Lower	
1= Upper	
2= Both	
Description: Compare operation.	
Parameter: double SA_LowerLimit1..3	SA_LowerLimit1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 150000.0	
Unit: mm	
Description: Lower limit.	

9.4. Commands for ILR2250

Parameter: double SA_UpperLimit1..3	SA_UpperLimit1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 150000.0	
Unit: mm	
Description: Upper limit.	
Parameter: int32_t SA_ErrorOutHoldTime	SA_ErrorOutHoldTime
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10000	
Unit: ms	
Description: Hold time.	
Parameter: double SA_AnalogScaleRangeLowerLimit	SA_AnalogScaleRangeLowerLimit
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 150000.0	
Unit: mm	
Description: Value which represents lowest voltage/current (at two point scaling).	
Parameter: double SA_AnalogScaleRangeUpperLimit	SA_AnalogScaleRangeUpperLimit
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 150000.0	
Unit: mm	
Description: Value which represents highest voltage/current (at two point scaling).	
Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Description: Name of the sensor.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the sensor.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the sensor.	

9.4. Commands for ILR2250

Parameter: double SA_Range SA_Range
Direction: Up
Unit: mm
Description: Range of the sensor.

9.4.1.2 Triggering

9.4.1.2.1 Set_TriggerMode (TRIGGER)

Set the trigger mode.

Parameter: int32_t SP_TriggerMode SP_TriggerMode
Direction: Down
Valid values:
 0= None
 1= Edge
 2= Level (PULSE)
 3= Software
Description: Trigger mode.

9.4.1.2.2 Get_TriggerMode (TRIGGER)

Get the active trigger mode.

Parameter: int32_t SA_TriggerMode SA_TriggerMode
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= None
 1= Edge
 2= Level (PULSE)
 3= Software
Description: Trigger mode.

9.4.1.2.3 Set_TriggerMoment (TRIGGERAT)

Set the trigger time.

Parameter: int32_t SP_TriggerMoment SP_TriggerMoment
Direction: Down
Valid values:
 0= Input
 1= Output
Description: Trigger moment.

9.4. Commands for ILR2250

9.4.1.2.4 Get_TriggerMoment (TRIGGERAT)

Get the active trigger time.

Parameter: int32_t SA_TriggerMoment SA_TriggerMoment
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = Input
 1 = Output
Description: Trigger moment.

9.4.1.2.5 Set_TriggerLevel (TRIGGERLEVEL)

Set the trigger level.

Parameter: int32_t SP_TriggerLevel SP_TriggerLevel
Direction: Down
Valid values:
 0 = High
 1 = Low
Description: Trigger level.

9.4.1.2.6 Get_TriggerLevel (TRIGGERLEVEL)

Get the active trigger level.

Parameter: int32_t SA_TriggerLevel SA_TriggerLevel
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = High
 1 = Low
Description: Trigger level.

9.4.1.2.7 Set_TriggerCount (TRIGGERCOUNT)

Set the number of values to measure at trigger.

Parameter: int32_t SP_TriggerCount SP_TriggerCount
Direction: Down
Valid values:
Minimum: 1
Maximum: 2147483647 (INT32_MAX)
Description: Number of values to measure. 2147483647 means endless measurement (INFINITE).

9.4. Commands for ILR2250

9.4.1.2.8 Get_TriggerCount (TRIGGERCOUNT)

Get the number of values to measure at trigger.

Parameter: int32_t SA_TriggerCount

SA_TriggerCount

Direction: Up

Valid values:

Minimum: 1

Maximum: 2147483647 (INT32_MAX)

Description: Number of values to measure. 2147483647 means endless measurement (INFINITE).

9.4.1.2.9 Software_Trigger (TRIGGERSW)

Execute a software trigger.

9.4.1.3 Interfaces

9.4.1.3.1 Set_Baudrate (BAUDRATE)

Set baudrate of sensor for serial RS422 communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

9600

115200

230400

Unit: Baud

Description: Baudrate of sensor.

Parameter: int32_t CP_InterruptDataTransfer

CP_InterruptDataTransfer

Direction: Down

Valid values:

0 = false

1 = true

Default: 0

Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Set_DataOutInterface](#)) and enabling it again at end (only if enabled before).

9.4.1.3.2 Get_Baudrate (BAUDRATE)

Get baudrate of sensor for serial RS422 communication.

Parameter: int32_t SA_SensorBaudrate

SA_SensorBaudrate

Direction: Up

Valid values:

9600

115200

230400

Unit: Baud

Description: Baudrate of sensor.

9.4.1.4 Parameter management

9.4.1.4.1 Save_InterfaceParameters (BASICSETTINGS STORE)

Save actual interface parameters at controller.

9.4.1.4.2 Load_InterfaceParameters (BASICSETTINGS READ)

Load stored interface parameters into controller RAM.

9.4.1.4.3 Save_MeasureParameters (MEASSETTINGS STORE)

Save actual measurement parameters at controller.

9.4.1.4.4 Load_MeasureParameters (MEASSETTINGS READ)

Load stored measurement parameters into controller RAM.

At this command the controller may change output data after applying new setting.
 If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually.
 After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

9.4.1.4.5 Get_MeasurePresetList (MEASSETTINGS PRESETLIST)

Get list of all preset settings.

Parameter: String SA_PresetNames	SA_PresetNames
Direction: Up	
Description: List (separated by new line).	

9.4.1.4.6 Set_MeasurePresetMode (MEASSETTINGS PRESETMODE)

Set mode of current preset settings.

Parameter: int32_t SP_PresetMode	SP_PresetMode
Direction: Down	
Valid values:	
0= Auto	
1= Fast	
2= Accurate	
3= Precise	
Description: Preset mode.	

9.4.1.4.7 Get_MeasurePresetMode (MEASSETTINGS PRESETMODE)

Get mode of current preset settings.

Parameter: int32_t SA_PresetMode SA_PresetMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Auto
- 1= Fast
- 2= Accurate
- 3= Precise

Description: Preset mode.

9.4.1.4.8 Set_Default (SETDEFAULT)

Reset the sensor to default settings.

At this command the sensor may change output data after applying default settings.

If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually.

After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DefaultType SP_DefaultType

Direction: Down

Valid values:

- 0= Delete all settings and load the factory settings (ALL)
- 1= Delete all measuring settings (MEASSETTINGS)
- 2= Delete all basic settings (BASICSETTINGS)

Description: Specifies which settings should be reset.

9.4.2 Measurement

9.4.2.1 General

9.4.2.1.1 Set_LaserPower (LASER)

Specify the laser power at sensor.

Parameter: int32_t SP_LaserPower SP_LaserPower

Direction: Down

Valid values:

- 0= Off
- 1= On
- 2= Measure

Description: Laser power.

9.4.2.1.2 Get_LaserPower (LASER)

Get the laser power from sensor.

Parameter: int32_t SA_LaserPower SA_LaserPower
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = Off
 1 = On
 2 = Measure
Description: Laser power.

9.4.2.2 Video signal

9.4.2.2.1 Set_ROI (ROI)

Set the region of interest.

Parameter: double SP_ROIStart SP_ROIStart
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 150000.0
Unit: mm
Description: First position in region of interest.

Parameter: double SP_ROIEnd SP_ROIEnd
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 150000.0
Unit: mm
Description: Last position in region of interest.

9.4.2.2.2 Get_ROI (ROI)

Get the region of interest.

Parameter: double SA_ROIStart SA_ROIStart
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 150000.0
Unit: mm
Description: First position in region of interest.

Parameter: double SA_ROIEnd SA_ROIEnd
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 150000.0
Unit: mm
Description: Last position in region of interest.

9.4.3 Data output

9.4.3.1 General

9.4.3.1.1 Set_DataOutInterface (OUTPUT)

Set the active interface for data output.

At this command the sensor may change output data automatically. If first bit of `IP_AutomaticMode` is set (1), `Get_OutputInfo_RS422` is called automatically after this command. Otherwise, you have to call it manually. After this call `Get_TransmittedDataInfo` to retrieve new setting.

Parameter: int32_t SP_DataOutInterface

SP_DataOutInterface

Direction: Down

Valid values:

0= None

1= RS422

Description: Active interface for data output.

9.4.3.1.2 Get_DataOutInterface (OUTPUT)

Get the active interface for data output.

Parameter: int32_t SA_DataOutInterface

SA_DataOutInterface

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= None

1= RS422

7= RS422_ASCII (does not work with MEDAQLib)

Description: Active interface for data output.

9.4.3.1.3 Set_HoldLastValid (OUTHOLD)

Set the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SP_HoldLastValid

SP_HoldLastValid

Direction: Down

Valid values:

Minimum: -1

Maximum: 2147483646

Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).

9.4.3.1.4 Get_HoldLastValid (OUTHOLD)

Get the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SA_HoldLastValid

SA_HoldLastValid

Direction: Up

Valid values:

Minimum: -1

Maximum: 2147483646

Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).

9.4.3.2 Selected measurement values

9.4.3.2.1 Set_OutputAdditional_RS422 (OUTADD_RS422)

Set the additional data to be output at RS422 interface (available from firmware version 1.1.5).

For this command an [Update...](#) and a [Reset...](#) meta command is available.

Parameter: int32_t SP_OutputAdditionalTimestamp_RS422

Direction: Down

SP_OutputAdditionalTimestamp_RS422

Valid values:

0= no

1= yes

Description: Specify if timestamp is transmitted.

9.4.3.2.2 Get_OutputAdditional_RS422 (OUTADD_RS422)

Get the additional data which is output at RS422 interface (available from firmware version 1.1.5).

Parameter: int32_t SA_OutputAdditionalTimestamp_RS422

SA_OutputAdditionalTimestamp_RS422

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if timestamp is transmitted.

9.4.3.3 Switching outputs

9.4.3.3.1 Set_ErrorOutput<n> (ERROROUT<n>)

Set condition to be used to set error output n ∈ {1..3}.

Parameter: int32_t SP_ErrorOutput<n>

SP_ErrorOutput<n>

Direction: Down

Valid values:

0= Output is disabled (NONE)

1= Switch when result is outside bounds of analog range (TEACH)

2= Switch on excess of ERRORLIMIT<n> (LIMIT)

Description: Condition for error output.

9.4.3.3.2 Get_ErrorOutput<n> (ERROROUT<n>)

Get condition to be used to set error output n ∈ {1..3}.

Parameter: int32_t SA_ErrorOutput<n>

SA_ErrorOutput<n>

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Output is disabled (NONE)

1= Switch when result is outside bounds of analog range (TEACH)

2= Switch on excess of ERRORLIMIT<n> (LIMIT)

Description: Condition for error output .

9.4.3.3.3 Set_ErrorLevelOut<n> (ERRORLEVELOUT<n>)

Set level of error output n ∈ {1..3} on error.

Parameter: int32_t SP_ErrorLevelOut<n>

SP_ErrorLevelOut<n>

Direction: Down

Valid values:

- 0= NPN
- 1= PNP
- 2= Push-Pull (PUSHPULL)
- 3= Push-Pull negated (PUSHPULLNEG)

Description: Error level for error output.

9.4.3.3.4 Get_ErrorLevelOut<n> (ERRORLEVELOUT<n>)

Get level of error output n ∈ {1..3} on error.

Parameter: int32_t SA_ErrorLevelOut<n>

SA_ErrorLevelOut<n>

Direction: Up

Valid values:

- 0= NPN
- 1= PNP
- 2= Push-Pull (PUSHPULL)
- 3= Push-Pull negated (PUSHPULLNEG)

Description: Error level for error output.

9.4.3.3.5 Set_ErrorHysteresis<n> (ERRORHYSTERESIS<n>)

Set the hysteresis n ∈ {1..3} of the threshold function (Set_ErrorLimit<n>).

Parameter: double SP_ErrorHysteresis<n>

SP_ErrorHysteresis<n>

Direction: Down

Valid values:

- Minimum:** 0.0
- Maximum:** 150000.0

Unit: mm

Description: Error hysteresis.

9.4.3.3.6 Get_ErrorHysteresis<n> (ERRORHYSTERESIS<n>)

Get the hysteresis n ∈ {1..3} of the threshold function (Set_ErrorLimit<n>).

Parameter: double SA_ErrorHysteresis<n>

SA_ErrorHysteresis<n>

Direction: Up

Valid values:

- Minimum:** 0.0
- Maximum:** 150000.0

Unit: mm

Description: Error hysteresis.

9.4. Commands for ILR2250

9.4.3.3.7 Set_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)

Set the compare operation for limit n $\in \{1..3\}$.

Parameter: int32_t SP_ErrorLimitCompOp<n>

SP_ErrorLimitCompOp<n>

Direction: Down

Valid values:

- 0= Lower
- 1= Upper
- 2= Both

Description: Compare operation

9.4.3.3.8 Get_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)

Get the compare operation for limit n $\in \{1..3\}$.

Parameter: int32_t SA_ErrorLimitCompOp<n>

SA_ErrorLimitCompOp<n>

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Lower
- 1= Upper
- 2= Both

Description: Compare operation

9.4.3.3.9 Set_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)

Set the error limits n $\in \{1..3\}$.

Parameter: double SP_LowerLimit<n>

SP_LowerLimit<n>

Direction: Down

Valid values:

- Minimum:** 0.0
- Maximum:** 150000.0

Unit: mm

Description: Lower limit.

Parameter: double SP_UpperLimit<n>

SP_UpperLimit<n>

Direction: Down

Valid values:

- Minimum:** 0.0
- Maximum:** 150000.0

Unit: mm

Description: Upper limit.

9.4.3.3.10 Get_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)

Get the error limits n $\in \{1..3\}$.

Parameter: double SA_LowerLimit<n>

SA_LowerLimit<n>

Direction: Up

Valid values:

- Minimum:** 0.0
- Maximum:** 150000.0

Unit: mm

Description: Lower limit.

Parameter: double SA_UpperLimit<n> SA_UpperLimit<n>
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 150000.0
Unit: mm
Description: Upper limit.

9.4.3.3.11 Set_ErrorOutHoldTime (ERROROUTHOLD)

An error for a switch output will be hold at least the given time.

Parameter: int32_t SP_ErrorOutHoldTime SP_ErrorOutHoldTime
Direction: Down
Valid values:
Minimum: 0
Maximum: 10000
Unit: ms
Description: Hold time.

9.4.3.3.12 Get_ErrorOutHoldTime (ERROROUTHOLD)

An error for a switch output will be hold at least the given time.

Parameter: int32_t SA_ErrorOutHoldTime SA_ErrorOutHoldTime
Direction: Up
Valid values:
Minimum: 0
Maximum: 10000
Unit: ms
Description: Hold time.

9.4.3.4 Analog output

9.4.3.4.1 Set_AnalogScaleRange (ANALOGSCALERANGE)

Set the scaling factors for analog output.

Parameter: double SP_AnalogScaleRangeLowerLimit SP_AnalogScaleRangeLowerLimit
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 150000.0
Unit: mm
Description: Value which represents lowest voltage/current (at two point scaling).

9.4. Commands for ILR2250

Parameter: double SP_AnalogScaleRangeUpperLimit
Direction: Down
Valid values:
 Minimum: 0.0
 Maximum: 150000.0
Unit: mm
Description: Value which represents highest voltage/current (at two point scaling).

9.4.3.4.2 Get AnalogScaleRange (ANALOGSCALERANGE)

Get the scaling factor for analog output.

Parameter: double SA_AnalogScaleRangeLowerLimit
Direction: Up
Valid values:
 Minimum: 0.0
 Maximum: 150000.0
Unit: mm
Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SA_AnalogScaleRangeUpperLimit
Direction: Up
Valid values:
 Minimum: 0.0
 Maximum: 150000.0
Unit: mm
Description: Value which represents highest voltage/current (at two point scaling).

9.4.4 Internal commands

9.4.4.1 Get FirmwareVersion

Retrieve firmware version from sensor.
This is an internal command. It should not be used by the customer.

Parameter: String CA_FirmwareVersion
Direction: Up
Description: Firmware version

9.4.4.2 Prepare UpdateFirmware

Prepares a firmware update at sensor.
This is an internal command. It should not be used by the customer.

Parameter: Binary data XP_FirmwareFile
Direction: Down
Description: Firmware file

9.4. Commands for ILR2250

Parameter: String CA_FileName	CA_FileName
Direction: Up	
Description: Internal name of firmware file	
Parameter: String CA_Date	CA_Date
Direction: Up	
Description: Date of firmware file	
Parameter: String CA_ArticleNumber	CA_ArticleNumber
Direction: Up	
Description: Article number of destination device	
Parameter: String CA_SerialNumber	CA_SerialNumber
Direction: Up	
Description: Serial number of destination device	

9.4.4.3 Start_UpdateFirmware

Start firmware update at sensor.

This is an internal command. It should not be used by the customer.

9.4.4.4 Get_UpdateFirmwareProgress

Update firmware version at sensor.

Attention! This function can takes up to 1 minute. This is an internal command. It should not be used by the customer.

Parameter: double CA_Progress	CA_Progress
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: Progress of firmware update	
Parameter: String CA_Description	CA_Description
Direction: Up	
Description: Current state of firmware update.	
Parameter: int32_t CA_Result	CA_Result
Direction: Up	
Valid values:	
0= Failed	
1= Success	
Description: Result of firmware update (only available at end of update).	
Parameter: int32_t CA_Finished	CA_Finished
Direction: Up	
Valid values:	
-1= Firmware update is not prepared, call Prepare_UpdateFirmware first	
0= No	
1= Yes	
Description: Tell if firmware update is in progress.	

9.4.4.5 Generate_Firmware

This is an internal command. It should not be used by the customer.

10 Commands for optoNCDT (ILD) sensors

ILD1401 demo case, ILD1800 and ILD2000 are no longer supported.

10.1 Commands for ILD1401

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (native).
- [TCP/IP](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface).
- [IF2004_USB](#) (for sensor ILD1402 in compatibility mode).
- [IF2008](#) (for sensor ILD1402 in compatibility mode).
- [IF2008_ETH](#) (for sensor ILD1402 in compatibility mode).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Info](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to scale data.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor, from 0 to 4095.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_Info](#)), error values are scaled depending of [IP_ScaleErrorValues](#).

10.1.1 General commands

10.1.1.1 General

10.1.1.1.1 Get_Info (INFO)

Retrieve some information about the sensor.

Parameter: String SA_ArticleNumber SA_ArticleNumber
Direction: Up
Valid values:
 Numeric value
Description: Article number of the sensor

Parameter: String SA_Option SA_Option
Direction: Up
Valid values:
 Numeric value
Description: Option of the sensor

10.1. Commands for ILD1401

Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Unit: mm	
Description: Range of the sensor	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of the sensor	
Parameter: String SA_Date	SA_Date
Direction: Up	
Description: Software release date of the sensor	
Parameter: int32_t SA_OutputType	SA_OutputType
Direction: Up	
Valid values:	
0= analog	
1= RS232	
Description: Data output (only values, not answer) interface of the sensor	
Parameter: int32_t SA_ErrorHandler	SA_ErrorHandler
Direction: Up	
Valid values:	
0= hold last value	
1= error values	
Description: If the sensor cannot measure values, it can output the last valid value or it can output an error value (only at analog output).	
Parameter: int32_t SA_Median_OnOff	SA_Median_OnOff
Direction: Up	
Valid values:	
0= none	
1= Median 3	
Description: The sensor can perform averaging (Median over 3 values).	

10.1.1.2 Get_Version (VERSION)

Retrieve the sensor software version.

Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of the sensor.	

10.1.1.1.3 Reset_Boot (BOOT)

Resets the sensor. This command has no parameters.

10.1.2 Measurement

10.1.2.1 Set_Median (MEDIAN)

Set the internal averaging mode of the sensor.

Parameter: int32_t SP_Median_OnOff

SP_Median_OnOff

Direction: Down

Valid values:

0= none

1= Median 3

Description: The sensor can perform averaging (Median over 3 values).

10.1.3 Data output

10.1.3.1 General

10.1.3.1.1 SaveLastMV (SAVELASTMV)

Specifies the error handling of sensor if it cannot measure values.

Parameter: int32_t SP_ErrorHandler

SP_ErrorHandler

Direction: Down

Valid values:

0= hold last value

1= error values

Description: If the sensor cannot measure values, it can output the last valid value or it can output an error value (only at analog output).

10.1.3.1.2 Set_OutputChannel (OUTPUTCHANNEL)

Set the output channel of the sensor.

Parameter: int32_t SP_OutputType

SP_OutputType

Direction: Down

Valid values:

0= analog

1= RS232

Description: Specifies data output (only values, not answer) interface of the sensor.

10.2 Commands for ILD1302 and ILD1402

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (native) at ILD1402 and (additional, e.g. [IF2001_USB](#) (RS422) and RS232 high level interface) at ILD1302.
- [IF2004](#) (native).
- [TCP/IP](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_CIMode_Set_CIMode1402](#) (if not active) and [Get_Settings](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate and to scale data.

If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls sensor command [Dat_Out_On](#) and [Laser_On](#) at [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor, from 0 to 16383.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_Settings](#), error values are scaled depending of [IP_ScaleErrorValues](#)).

10.2.1 General commands

10.2.1.1 General

10.2.1.1.1 Get_Info (GET_INFO)

Retrieve some information about the sensor.

Parameter: String SA_Sensor SA_Sensor
Direction: Up
Description: Name of the sensor.

Parameter: String SA_SensorType SA_SensorType
Direction: Up
Description: Type of the sensor.

Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the sensor.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the sensor.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Unit: mm	
Description: Range of the sensor.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of the sensor.	
Parameter: String SA_BootLoaderVer	SA_BootLoaderVer
Direction: Up	
Description: Boot loader version of the sensor.	
Parameter: String SA_Date	SA_Date
Direction: Up	
Description: Software release date of the sensor.	
Parameter: int32_t SA_OutputType	SA_OutputType
Direction: Up	
Valid values:	
0= current (4..20mA)	
1= RS422	
Description: Data output (only values, not answer) interface of the sensor.	
Parameter: int32_t SA_ErrorHandler	SA_ErrorHandler
Direction: Up	
Valid values:	
0= hold last value	
1= error values	
2..99= hold last valid for n values	
Description: If the sensor cannot measure values, it can output the last valid value or it can output error values at analog interface.	

Parameter: int32_t SA_AvType	SA_AvType
Direction: Up	
Valid values:	
0= moving	
1= Median	
Description: The averaging type.	
Parameter: int32_t SA_MovingCount	SA_MovingCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 128	
Description: The moving averaging value, if AvType is moving.	
Parameter: int32_t SA_MedianIndex	SA_MedianIndex
Direction: Up	
Valid values:	
3	
5	
7	
9	
Description: The Median value, if AvType is Median.	
Parameter: int32_t SA_Speed	SA_Speed
Direction: Up	
Valid values:	
0= 1.5kHz	
1= 1.0kHz	
2= 750Hz	
3= 375Hz	
4= 50Hz	
Valid for sensor:	
ILD1402	
Description: The output speed of the sensor.	
Parameter: int32_t SA_ASCII	SA_ASCII
Direction: Up	
Valid values:	
0= off (binary 2 bytes/value)	
1= on (ASCII 6 bytes/value)	
Description: Returns the mode the sensor is sending data (only values).	
Parameter: int32_t SA_OutputMode	SA_OutputMode
Direction: Up	
Valid values:	
0= continuous	
1= timed	
2= triggered	
Description: The output mode of the sensor.	
Parameter: int32_t SA_Keylock	SA_Keylock
Direction: Up	
Valid values:	
0= off (keys enabled)	
1= on (keys locked)	
2= auto (locked after 5 minutes)	
Description: The keypad state at the sensor.	

Parameter: int32_t SA_SaveSettingsMode	SA_SaveSettingsMode
Direction: Up	
Valid values:	
0= temporary in RAM	
1= persistant in Flash	
Description: The mode if parameters should be temporay or stored persistant.	
Parameter: int32_t SA_ExtInputMode	SA_ExtInputMode
Direction: Up	
Valid values:	
0= used for teaching	
1= used as trigger	
Description: Specifies if the external input is used for teaching or as trigger input.	
Parameter: int32_t SA_PeakSearching	SA_PeakSearching
Direction: Up	
Valid values:	
0= global maximum	
1= first peak	
2= last peak	
Description: Specifies how the peak searching algorithm does work.	
Parameter: int32_t SA_Threshold	SA_Threshold
Direction: Up	
Valid values:	
0= lower	
1= normal	
2= higher	
3= highest	
Description: Specifies the spectral threshold.	
Parameter: double SA_TeachValue1	SA_TeachValue1
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 16368.0	
Description: The lower teach limit.	
Parameter: double SA_TeachValue2	SA_TeachValue2
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 16368.0	
Description: The higher teach limit.	

10.2.1.1.2 Get_Settings (GET_SETTINGS)

Retrieve detailed information about the sensor.

Parameter: int32_t SA_OutputType	SA_OutputType
Direction: Up	
Valid values:	
0= current (4..20mA)	
1= RS422	
Description: Data output (only values, not answer) interface of the sensor.	

Parameter: double SA_TeachValue1	SA_TeachValue1
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 16368.0	
Description: The lower teach limit.	
Parameter: double SA_TeachValue2	SA_TeachValue2
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 16368.0	
Description: The higher teach limit.	
Parameter: int32_t SA_ErrorHandler	SA_ErrorHandler
Direction: Up	
Valid values:	
0= hold last value	
1= error values	
2..99= hold last valid for n values	
Description: If the sensor cannot measure values, it can output the last valid value or it can output error values at analog interface.	
Parameter: int32_t SA_AvType	SA_AvType
Direction: Up	
Valid values:	
0= moving	
1= Median	
Description: The averaging type.	
Parameter: int32_t SA_MovingCount	SA_MovingCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 128	
Description: The moving averaging value, if AvType is moving.	
Parameter: int32_t SA_MedianIndex	SA_MedianIndex
Direction: Up	
Valid values:	
3	
5	
7	
9	
Description: The Median value, if AvType is Median.	
Parameter: int32_t SA_Speed	SA_Speed
Direction: Up	
Valid values:	
0= 1.5kHz	
1= 1.0kHz	
2= 750Hz	
3= 375Hz	
4= 50Hz	
Valid for sensor:	
ILD1402	
Description: The output speed of the sensor.	

Parameter: int32_t SA_SensorBaudrate	SA_SensorBaudrate
Direction: Up	
Valid values:	
0= 115200 Baud	
1= 57600 Baud	
2= 38400 Baud	
3= 19200 Baud	
4= 9600 Baud	
Description: The serial connection baudrate of the sensor.	
Parameter: int32_t SA_ASCII	SA_ASCII
Direction: Up	
Valid values:	
0= off (binary 2 bytes/value)	
1= on (ASCII 6 bytes/value)	
Description: Returns the mode the sensor is sending data (only values).	
Parameter: int32_t SA_OutputMode	SA_OutputMode
Direction: Up	
Valid values:	
0= continuous	
1= timed	
2= triggered	
Description: The output mode of the sensor.	
Parameter: int32_t SA_OutputTime	SA_OutputTime
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 65535	
Unit: ms	
Description: Data output time of the sensor. It is used for timeout check.	
Parameter: int32_t SA_Keylock	SA_Keylock
Direction: Up	
Valid values:	
0= off (keys enabled)	
1= on (keys locked)	
2= auto (locked after 5 minutes)	
Description: The keypad state at the sensor.	
Parameter: int32_t SA_SaveSettingsMode	SA_SaveSettingsMode
Direction: Up	
Valid values:	
0= temporary in RAM	
1= persistant in Flash	
Description: The mode if parameters should be temporay or stored persistant.	
Parameter: int32_t SA_ExtInputMode	SA_ExtInputMode
Direction: Up	
Valid values:	
0= used for teaching	
1= used as trigger	
Description: Specifies if the external input is used for teaching or as trigger input.	

Parameter: int32_t SA_PeakSearching	SA_PeakSearching
Direction: Up	
Valid values:	
0= global maximum	
1= first peak	
2= last peak	
Description:	Specifies how the peak searching algorithm does work.
Parameter: int32_t SA_Threshold	SA_Threshold
Direction: Up	
Valid values:	
0= lower	
1= normal	
2= higher	
3= highest	
Description:	Specifies the spectral threshold.
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Unit: mm	
Description:	Range of the sensor.
Parameter: int32_t SA_Reserved1	SA_Reserved1
Direction: Up	
Description:	Reserved for further use.
Parameter: int32_t SA_Reserved2	SA_Reserved2
Direction: Up	
Description:	Reserved for further use.
Parameter: int32_t SA_Reserved3	SA_Reserved3
Direction: Up	
Description:	Reserved for further use.
Parameter: int32_t SA_Reserved4	SA_Reserved4
Direction: Up	
Description:	Reserved for further use.

10.2.1.1.3 Reset_Boot (RESET_BOOT)

Resets the sensor.

If first bit of [IP_AutomaticMode](#) is set (1), [Get_Settings](#) is called automatically after this command. Otherwise, you have to call it manually.

10.2.1.2 User level

10.2.1.2.1 Set_KeyLock (SET_KEYLOCK)

Locks/Unlocks the keypad of sensor.

Parameter: int32_t SP_Keylock	SP_Keylock
Direction: Down	
Valid values:	
0= off (keys enabled)	
1= on (keys locked)	
2= auto (locked after 5 minutes)	
Description:	The keypad state at the sensor.

10.2.1.3 Interfaces

10.2.1.3.1 Set_Baudrate (SET_BAUDRATE)

Set the baudrate of the serial interface of sensor. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Attention! To avoid synchronization problems, sending data by sensor should be disabled first (sensor command [Dat_Out_Off](#)).

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

- 0= 115200 Baud
- 1= 57600 Baud
- 2= 38400 Baud
- 3= 19200 Baud
- 4= 9600 Baud

Description: The serial connection baudrate of the sensor.

Parameter: int32_t CP_InterruptDataTransfer

CP_InterruptDataTransfer

Direction: Down

Valid values:

- 0= false
- 1= true

Default: 0

Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Dat_Out_Off](#)) and enabling it again ([Dat_Out_On](#)) at end.

10.2.1.4 Parameter management

10.2.1.4.1 Set_Default (SET_DEFAULT)

Resets the sensor to factory settings.

If first bit of [IP_AutomaticMode](#) is set (1), [Get_Settings](#) is called automatically after this command. Otherwise, you have to call it manually.

10.2.1.4.2 Set_SaveSettingsMode (SET_SAVE_SETTINGS_MODE)

Set the save settings mode of sensor.

Parameter: int32_t SP_SaveSettingsMode

SP_SaveSettingsMode

Direction: Down

Valid values:

- 0= temporary in RAM
- 1= persistant in Flash

Description: The mode if parameters should be temporary or stored persistant.

10.2.1.5 Internal controller commands

10.2.1.5.1 Laser_Off (LASER_OFF)

Switch the laser off.

10.2.1.5.2 Laser_On (LASER_ON)

Switch the laser on.

10.2.1.5.3 Set_ExtInputMode (SET_EXT_INPUT_MODE)

Set the mode of external input at sensor.

Parameter: int32_t SP_ExtInputModule

SP_ExtInputModule

Direction: Down

Valid values:

0= used for teaching

1= used as trigger

Description: Specifies if the external input is used for teaching or as trigger input.

10.2.1.5.4 Get_CI_Mode (GET_CI_MODE)

Retrieve the sensor mode.

Valid for sensor:

ILD1402

Parameter: int32_t SA_CI_Mode

SA_CI_Mode

Direction: Up

Valid values:

0= ILD1401 compatibility mode

1= ILD1402 mode

Description: Sensor mode.

10.2.1.5.5 Set_CI_Mode_1401 (SET_CI_MODE_1401)

Set the sensor in compatibility mode for ILD1401.

No other commands expect Set_CI_Mode_1402 will work now.

Attention! If the interface is [IF2004](#), the sensor cannot be accessed any longer, because IF2004 does not support 38400 Baud (Baudrate of ILD1401).

Valid for sensor:

ILD1402

10.2.1.5.6 Set_CIMode_1402 (SET_CIMODE_1402)

Set the sensor back to ILD1402 mode.

If first bit of IP_AutomaticMode is set (1), Get_Settings is called automatically after this command. Otherwise, you have to call it manually.

Valid for sensor:

ILD1402

10.2.1.5.7 Set_OperationMode (SET_MODE)

Set the operation mode at sensor.

Valid for sensor:

ILD1402 with Option 207

Parameter: int32_t SP_OperationMode

SP_OperationMode

Direction: Down

Valid values:

- 1 = Measure mode
- 2 = User mode

Description: The operation mode.

10.2.1.5.8 Set_LaserDiode<n> (SET_LD<n>)

Set the power of laser diode n ∈ {1..2} at sensor.

Valid for sensor:

ILD1402 with Option 207

Parameter: int32_t SP_LaserDiode<n>

SP_LaserDiode<n>

Direction: Down

Valid values:

- Minimum: 0
- Maximum: 25

Description: The power of laser diode.

10.2.1.5.9 Get_LaserDiodeError (GET_LDERROR)

Get the error state of laser diode at sensor.

Valid for sensor:

ILD1402 with Option 207

Parameter: int32_t SA_LaserDiodeError

SA_LaserDiodeError

Direction: Up

Valid values:

Bit combination of eight bits

Description: The error state of laser diode.

10.2.1.5.10 Get_CurrentOfMonitor (GET_IMON)

Get the current at monitor diode.

Valid for sensor:

ILD1402 with Option 207

Parameter: int32_t SA_CurrentOfMonitor

SA_CurrentOfMonitor

Direction: Up

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Unit: nA

Description: Current at monitor diode.

10.2.2 Measurement

10.2.2.1 Set_Speed (SET_SCANRATE)

Set the data acquisition speed of the sensor.

Valid for sensor:

ILD1402

Parameter: int32_t SP_Speed

SP_Speed

Direction: Down

Valid values:

0= 1.5kHz

1= 1.0kHz

2= 750Hz

3= 375Hz

4= 50Hz

Description: The output speed of the sensor.

10.2.2.2 Set_PeakSearching (SET_PEAKSEARCHING)

Set the mode of external input at sensor.

Parameter: int32_t SP_PeakSearching

SP_PeakSearching

Direction: Down

Valid values:

0= global maximum

1= first peak

2= last peak

Description: Specifies how the peak searching algorithm does work.

10.2.2.3 Get_Video

Get recent video signal from sensor.

Parameter: Binary data SA_VideoSignal

SA_VideoSignal

Direction: Up

Valid values:

256 bytes, convertible to 128 words.

Description: Raw video signal

10.2.2.4 Set_Threshold (SET_THRESHOLD)

Set the spectral threshold of sensor.

Parameter: int32_t SP_Threshold

Direction: Down

Valid values:

- 0= lower
- 1= normal
- 2= higher
- 3= highest

Description: Specifies the spectral threshold.

10.2.2.5 Set_Av (SET_AV)

Set averaging type and value of sensor.

Parameter: int32_t SP_AvType

Direction: Down

Valid values:

- 0= moving
- 1= Median

Description: The averaging type.

Parameter: int32_t SP_MovingCount

Direction: Down

Valid values:

- Minimum: 1
- Maximum: 128

Description: The moving averaging value, if AvType is moving.

Parameter: int32_t SP_MedianIndex

Direction: Down

Valid values:

- 3
- 5
- 7
- 9

Description: The Median value, if AvType is Median.

10.2.3 Data output

10.2.3.1 General

10.2.3.1.1 Dat_Out_Off (DAT_OUT_OFF)

Switch off data output from sensor.

10.2.3.1.2 Dat_Out_On (DAT_OUT_ON)

Switch on data output from sensor.

10.2.3.1.3 Set_ErrorHandler (SET_ANALOG_ERROR_HANDLER)

Set the behaviour on invalid values at analog interface sensor.

Parameter: int32_t SP_ErrorHandler

SP_ErrorHandler

Direction: Down

Valid values:

0= hold last value

1= error values

2..99= hold last valid for n values

Description: If the sensor cannot measure values, it can output the last valid value or it can output error values at analog interface.

10.2.3.1.4 ASCII_Output (ASCII_OUTPUT)

Set digital data transfer (only values, no sensor answer) to ASCII or binary.

Parameter: int32_t SP_ASCII

SP_ASCII

Direction: Down

Valid values:

0= off (binary 2 bytes/value)

1= on (ASCII 6 bytes/value)

Description: Returns the mode the sensor is sending data (only values).

10.2.3.1.5 Set_OutputType (SET_OUTPUT_CHANNEL)

Set the output type of sensor.

Parameter: int32_t SP_OutputType

SP_OutputType

Direction: Down

Valid values:

0= current (4..20mA)

1= RS422

Description: Data output (only values, not answer) interface of the sensor.

10.2.3.1.6 Set_OutputMode (SET_OUTPUTMODE)

Set the output mode of sensor.

Parameter: int32_t SP_OutputMode

SP_OutputMode

Direction: Down

Valid values:

0= continuous

1= timed

2= triggered

Description: Data output mode of the sensor.

10.2.3.1.7 Set_OutputTime (SET_OUTPUTTIME_MS)

Set the output time of sensor.

Parameter: int32_t SP_OutputTime SP_OutputTime
Direction: Down
Valid values:
Minimum: 1
Maximum: 65535
Unit: ms
Description: Data output time of the sensor.

10.2.3.2 Analog output

10.2.3.2.1 Set_TeachValue (SET_TEACH_VALUE)

Set the teaching values at sensor.

Parameter: double SP_TeachValue1 SP_TeachValue1
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 16368.0
Description: The lower teach limit.

Parameter: double SP_TeachValue2 SP_TeachValue2
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 16368.0
Description: The higher teach limit.

10.2.3.2.2 Reset_TeachValue (RESET_TEACH_VALUE)

Reset the teaching values at sensor.

10.3 Commands for ILD1220, ILD1320, ILD1420, ILD1750/6x and ILD1900

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
- [IF2004](#) (native).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).

- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_AllParameters](#) (SP_Additional= 1) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate, to interpret and scale data and to assign values. If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls optionally sensor command [Set_DataOutInterface](#), [Get_LaserPower](#) and optionally [Set_LaserPower](#) at [OpenSensor](#) (not for ILD1220).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_AllParameters](#) (SP_Additional= 1)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

10.3.1 General commands

10.3.1.1 General

10.3.1.1.1 Get_Help (HELP)

Retrieve a help text from sensor for a specific command.

Parameter: String SP_Command

SP_Command

Direction: Down

Valid values:

"" (empty string, means general help)
or any command name

Description: Name of the command.

Parameter: String SA_HelpText

SA_HelpText

Direction: Up

Description: Help text to the command.

10.3.1.1.2 Get_Info (GETINFO)

Retrieve information about the sensor.

Parameter: String SA_Sensor

SA_Sensor

Direction: Up

Description: Name of the sensor.

Parameter: String SA_SerialNumber

SA_SerialNumber

Direction: Up

Valid values:

Numeric value

Description: Serial number of the sensor.

Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the sensor.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the sensor.	
Parameter: String SA_CableHead	SA_CableHead
Direction: Up	
Valid for sensor:	
ILD1220	
ILD1320	
ILD1420	
ILD1900	
Description: Sensor cable type.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Unit: mm	
Description: Range of the sensor.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the sensor.	
Parameter: String SA_HardwareRevision	SA_HardwareRevision
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of the controller board.	
Parameter: String SA_BuildID	SA_BuildID
Direction: Up	
Description: Build ID	
Parameter: String SA_BuildTimestamp	SA_BuildTimestamp
Direction: Up	
Description: Build timestamp	
Parameter: String SA_BootVersion	SA_BootVersion
Direction: Up	
Description: Boot version	

10.3.1.1.3 Get_OutputInfo_RS422 (GETOUTINFO_RS422)

Retrieve information which data is output at RS422 interface.

Valid for sensor:

- ILD1220 (from firmware version 001.062)
- ILD1320
- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SA_OutputVideoRaw_RS422

SA_OutputVideoRaw_RS422

Direction: Up

Valid values:

- 0 = no
- 1 = yes

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Description: Specify if raw video signal is transmitted.

Parameter: int32_t SA_OutputAdditionalShutterTime_RS422

SA_OutputAdditionalShutterTime_RS422

Direction: Up

Valid values:

- 0 = no
- 1 = yes

Valid for sensor:

- ILD1320
- ILD1420
- ILD1750
- ILD1900

Description: Specify if shutter time is transmitted.

Parameter: int32_t SA_OutputAdditionalCounter_RS422

SA_OutputAdditionalCounter_RS422

Direction: Up

Valid values:

- 0 = no
- 1 = yes

Valid for sensor:

- ILD1220 (from firmware version 001.062)
- ILD1320
- ILD1420
- ILD1750
- ILD1900

Description: Specify if counter is transmitted.

Parameter: int32_t SA_OutputAdditionalMeasrate_RS422

SA_OutputAdditionalMeasrate_RS422

Direction: Up

Valid values:

- 0 = no
- 1 = yes

Valid for sensor:

- ILD1750
- ILD1900

Description: Specify if measrate is transmitted.

Parameter: int32_t SA_OutputAdditionalTimestampLo_RS422	SA_OutputAdditionalTimestampLo_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description:	Specify if timestamp (lower 16 bit) is transmitted.
Parameter: int32_t SA_OutputAdditionalTimestampHi_RS422	SA_OutputAdditionalTimestampHi_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description:	Specify if timestamp (upper 16 bit) is transmitted.
Parameter: int32_t SA_OutputAdditionalIntensity1_RS422	SA_OutputAdditionalIntensity1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
Description:	Specify if intensity 1 is transmitted.
Parameter: int32_t SA_OutputAdditionalIntensity_RS422	SA_OutputAdditionalIntensity_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1900	
Description:	Specify if intensity is transmitted.
Parameter: int32_t SA_OutputAdditionalDistanceRaw1_RS422	SA_OutputAdditionalDistanceRaw1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description:	Specify if uncalibrated distance 1 is transmitted.

Parameter: int32_t SA_OutputDistanceUnlinearized_RS422	SA_OutputDistanceUnlinearized_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Specify if unlinearized distance 1 is transmitted.	
Parameter: int32_t SA_OutputDistance1_RS422	SA_OutputDistance1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1220 (from firmware version 001.062)	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description: Specify if distance 1 is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description: Specify if state is transmitted.	

10.3.1.1.4 Set_Unit (UNIT)

Set the unit for configuration and display in the web diagram.

Parameter: int32_t SP_DisplayUnit	SP_DisplayUnit
Direction: Down	
Valid values:	
0= mm	
1= Inch	
Description: Unit.	

10.3.1.1.5 Get_Unit (UNIT)

Get the unit for configuration and display in the web diagram.

Parameter: int32_t SA_DisplayUnit	SA_DisplayUnit
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= mm	
1= Inch	
Description: Unit.	

10.3.1.1.6 Set_SyncMode (SYNC)

Set the synchronization mode.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_SyncMode

SP_SyncMode

Direction: Down

Valid values:

- 0= None
- 1= Slave
- 2= Slave alternating (SLAVE_ALT)
- 3= Slave multi function input (SLAVE_MFI)
- 4= Master
- 5= Master alternating (MASTER_ALT)

Description: Synchronization mode.

10.3.1.1.7 Get_SyncMode (SYNC)

Get the synchronization mode.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_SyncMode

SA_SyncMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Slave
- 2= Slave alternating (SLAVE_ALT)
- 3= Slave multi function input (SLAVE_MFI)
- 4= Master
- 5= Master alternating (MASTER_ALT)

Description: Synchronization mode.

10.3.1.1.8 Set_Termination (TERMINATION)

Set connection of a termination resistor in sync line to prevent reflections.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_Termination

SP_Termination

Direction: Down

Valid values:

- 0= Off
- 1= On

Description: Termination resistor.

10.3.1.1.9 Get_Termination (TERMINATION)

Get connection of a termination resistor in sync line to prevent reflections.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_Termination

SA_Termination

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor
0 = Off
1 = On

Description: Termination resistor.

10.3.1.1.10 Reset_Boot (RESET)

Resets the sensor.

At this command the sensor may change output data after reboot. If first bit of **IP_AutomaticMode** is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

10.3.1.1.11 Reset_Counter (RESETCNT)

Resets sensor counter values.

Valid for sensor:

ILD1320
ILD1420
ILD1750
ILD1900

Parameter: int32_t SP_ResetTimestamp

SP_ResetTimestamp

Direction: Down

Valid values:

0 = No
1 = Yes

Description: Reset timestamp value.

Parameter: int32_t SP_ResetMeasCounter

SP_ResetMeasCounter

Direction: Down

Valid values:

0 = No
1 = Yes

Description: Reset counter value.

10.3.1.1.12 Set_Keylock (KEYLOCK)

Set key lock for sensor.

Parameter: int32_t SP_Keylock SP_Keylock

Direction: Down

Valid values:

0= Inactive (NONE)

1= Active

2= Automatic (AUTO)

Description: Keylock.

Parameter: int32_t SP_KeylockTime SP_KeylockTime

Direction: Down

Valid values:

Minimum: 1

Maximum: 60

Unit: Minutes

Description: Keylock time (only used at automatic keylock).

10.3.1.1.13 Get_Keylock (KEYLOCK)

Get key lock for sensor.

Parameter: int32_t SA_Keylock SA_Keylock

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Inactive (NONE)

1= Active

2= Automatic (AUTO)

Description: Keylock.

Parameter: int32_t SA_KeylockTime SA_KeylockTime

Direction: Up

Valid values:

Minimum: 0

Maximum: 60

Unit: Minutes

Description: Keylock time (only available at automatic keylock).

Parameter: int32_t SA_KeylockState SA_KeylockState

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Inactive (IS_INACTIVE)

1= Active (IS_ACTIVE)

Description: Actual keylock state (only available at automatic keylock).

10.3.1.1.14 Set_KeyFunction (KEYFUNC)

Set key function for sensor.

Valid for sensor:

- ILD1220
- ILD1320
- ILD1420

Parameter: int32_t SP_KeyFunction

SP_KeyFunction

Direction: Down

Valid values:

- 0= Key has no function (NONE)
- 1= Use key to master the measuring value (MASTER)
- 2= Use key to teach the analog output (TEACH)

Description: Key function.

10.3.1.1.15 Get_KeyFunction (KEYFUNC)

Get key function for sensor.

Valid for sensor:

- ILD1220
- ILD1320
- ILD1420

Parameter: int32_t SA_KeyFunction

SA_KeyFunction

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Key has no function (NONE)
- 1= Use key to master the measuring value (MASTER)
- 2= Use key to teach the analog output (TEACH)

Description: Key function.

10.3.1.1.16 Set_Echo (ECHO)

Set echo for sensor commands.

Parameter: int32_t SP_Echo

SP_Echo

Direction: Down

Valid values:

- 0= Off
- 1= On

Description: Echo mode.

10.3.1.1.17 Get_Echo (ECHO)

Get the echo mode.

Parameter: int32_t SA_Echo

SA_Echo

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Off
- 1= On

Description: Echo mode.

10.3.1.1.18 Get_AllParameters (PRINT)

Get all parameters from sensor.

Parameter: int32_t SP_Additional	SP_Additional
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description:	If set, additional information about sensor is output.
Parameter: int32_t SA_UserLevel	SA_UserLevel
Direction: Up	
Valid values:	
-1= Other user level (only for internal use)	
0= User	
1= Professional	
Description:	Actual user level.
Parameter: int32_t SA_DefaultUser	SA_DefaultUser
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= User	
1= Professional	
Description:	Default user level.
Parameter: int32_t SA_Echo	SA_Echo
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Description:	Echo mode.
Parameter: int32_t SA_DisplayUnit	SA_DisplayUnit
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= mm	
1= Inch	
Description:	Unit.
Parameter: int32_t SA_Keylock	SA_Keylock
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Inactive (NONE)	
1= Active	
2= Automatic (AUTO)	
Description:	Keylock.

Parameter: int32_t SA_KeylockTime SA_KeylockTime

Direction: Up

Valid values:

Minimum: 0

Maximum: 60

Unit: Minutes

Description: Keylock time (only available at automatic keylock).

Parameter: int32_t SA_KeylockState SA_KeylockState

Direction: Up

Valid values:

 -1 = Unknown parameter value from sensor

 0 = Inactive (IS_INACTIVE)

 1 = Active (IS_ACTIVE)

Description: Actual keylock state (only available at automatic keylock).

Parameter: int32_t SA_KeyFunction SA_KeyFunction

Direction: Up

Valid values:

 -1 = Unknown parameter value from sensor

 0 = Key has no function (NONE)

 1 = Use key to master the measuring value (MASTER)

 2 = Use key to teach the analog output (TEACH)

Valid for sensor:

 ILD1220

 ILD1320

 ILD1420

Description: Key function..

Parameter: int32_t SA_SyncMode SA_SyncMode

Direction: Up

Valid values:

 -1 = Unknown parameter value from sensor

 0 = None

 1 = Slave

 2 = Slave alternating (SLAVE_ALT)

 3 = Slave multi function input (SLAVE_MFI)

 4 = Master

 5 = Master alternating (MASTER_ALT)

Valid for sensor:

 ILD1750

 ILD1900

Description: Synchronization mode.

Parameter: int32_t SA_Termination SA_Termination

Direction: Up

Valid values:

 -1 = Unknown parameter value from sensor

 0 = Off

 1 = On

Valid for sensor:

 ILD1750

 ILD1900

Description: Termination resistor.

Parameter: int32_t SA_TriggerSource SA_TriggerSource

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = None
- 1 = Multi function input (MFI)
- 2 = Sync in-/output (SYNCIO)
- 3 = Software

Valid for sensor:

ILD1750

ILD1900

Description: Trigger source (input).

Parameter: int32_t SA_TriggerLevel SA_TriggerLevel

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = High
- 1 = Low

Valid for sensor:

ILD1750

ILD1900

Description: Trigger level.

Parameter: int32_t SA_TriggerMode SA_TriggerMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Edge
- 1 = Level (PULSE)

Valid for sensor:

ILD1750

ILD1900

Description: Trigger mode.

Parameter: int32_t SA_TriggerMode SA_TriggerMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = None
- 1 = Edge
- 2 = Level (PULSE)
- 3 = Software (only at ILD1320 and ILD1420)

Valid for sensor:

ILD1220

ILD1320

ILD1420

Description: Trigger mode.

Parameter: int32_t SA_TriggerMoment SA_TriggerMoment

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Input

1 = Output

Valid for sensor:

ILD1420
ILD1750
ILD1900

Description: Trigger moment.

Parameter: int32_t SA_TriggerCount

SA_TriggerCount

Direction: Up

Valid values:

Minimum: 0 (at ILD1220, ILD1320 and ILD1420), 1 (at ILD1750 and ILD1900)

Maximum: 16383

Description: Number of values to measure. 0 means no trigger (NONE), 16383 means endless measurement (INFINITE).

Parameter: int32_t SA_SensorBaudrate

SA_SensorBaudrate

Direction: Up

Valid values:

4000000 (only at ILD1750 and ILD1900)
3000000 (only at ILD1750 and ILD1900)
2000000 (only at ILD1750 and ILD1900)
1000000 (only at ILD1220, ILD1320 and ILD1420)
921600
691200
460800
256000 (only at ILD1220, ILD1320 and ILD1420)
230400
128000 (only at ILD1220, ILD1320 and ILD1420)
115200
56000 (only at ILD1220, ILD1320 and ILD1420)
19200 (only at ILD1220, ILD1320 and ILD1420)
9600

Unit: Baud

Description: Baudrate of sensor.

Parameter: int32_t SA_ApplicationLanguage

SA_ApplicationLanguage

Direction: Up

Valid values:

0= English (EN)
1= German (DE)

Description: Language of web interface.

Parameter: int32_t SA_MultiFunctionInputMode

SA_MultiFunctionInputMode

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= MFI has no function (NONE)
1= Use MFI to master the measuring value (MASTER)
2= Use MFI to teach the analog output (TEACH)
3= Use MFI to trigger the measuring process (TRIGGER)

Valid for sensor:

ILD1220
ILD1320
ILD1420

Description: Specifies how to use the multi function input.

Parameter: int32_t SA_MultiFunctionInputLevel	SA_MultiFunctionInputLevel
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = High (HTL_HIGH at ILD1220, ILD1320 and ILD1420, HTL at ILD1750 and ILD1900)	
1 = Low (HTL_LOW at ILD1220, ILD1320 and ILD1420, TTL at ILD1750 and ILD1900)	
Description: Multi function input level.	
Parameter: int32_t SA_InitialChartType	SA_InitialChartType
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Measure (MEAS)	
1 = Video signal (VIDEO)	
2 = Automatic (AUTO)	
Valid for sensor:	
ILD1220	
ILD1320	
ILD1420	
Description: Initial type of the chart at web interface.	
Parameter: int32_t SA_ChartType	SA_ChartType
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Measure (MEAS)	
1 = Video signal (VIDEO)	
Description: Type of the chart at web interface.	
Parameter: int32_t SA_TargetMode	SA_TargetMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Standard Target (STANDARD)	
1 = Multi-Surface Target (MULTISURFACE)	
2 = Light Penetration Target (PENETRATION)	
Valid for sensor:	
ILD1420	
ILD1750	
ILD1900	
Description: Target mode.	
Parameter: int32_t SA.MeasureMode	SA.MeasureMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Diffuse	
1 = Direct	
Valid for sensor:	
ILD1900 (only direct reflection variant)	
Description: Measure mode.	

Parameter: int32_t SA_MeasurePeak SA_MeasurePeak

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Greatest Amplitude (DISTA)
- 1 = First Peak (DIST1)
- 2 = Last Peak (DISTL)
- 3 = Largest Area (DISTW) (only at ILD1750 and ILD1900)

Valid for sensor:

ILD1320

ILD1420

ILD1750

ILD1900

Description: Peak to measure.

Parameter: int32_t SA_PeakWidthMax SA_PeakWidthMax

Direction: Up

Valid values:

- Minimum:** 1
- Maximum:** 512

Unit: Pixel

Valid for sensor:

ILD1900 (This parameter is only available at elevated user mode Professional+).

Description: Maximum peak width.

Parameter: double SA_PeakGlobalThreshold SA_PeakGlobalThreshold

Direction: Up

Valid values:

- Minimum:** 0.0
- Maximum:** 100.0

Unit: %

Valid for sensor:

ILD1900 (This parameter is only available at elevated user mode Professional+).

Description: Detection threshold.

Parameter: int32_t SA_ShutterMode SA_ShutterMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Exposure time is adapted automatically (MEAS)
- 1 = Exposure time is set manually (MANUAL)

Valid for sensor:

ILD1750

ILD1900

Description: Shutter mode.

Parameter: double SA_ShutterTime SA_ShutterTime

Direction: Up

Valid values:

- Minimum:** 0.1
- Maximum:** 3333.0 at ILD1750, 4000.0 at ILD1900

Unit: μ s

Valid for sensor:

ILD1750

ILD1900

Description: Exposure time.

Parameter: int32_t SA_ExposureMode

SA_ExposureMode

Direction: Up

Valid values:

0= Standard

1= Intelligent

2= Background (Background subtraction)

Valid for sensor:

ILD1900 (This parameter is only available at elevated user mode Professional+).

Description: Exposure mode.

Parameter: double SA_ExposureLimit

SA_ExposureLimit

Direction: Up

Valid values:
Minimum: 0.1

Maximum: 4000.0

Unit: μ s

Valid for sensor:

ILD1900 (This parameter is only available at elevated user mode Professional+).

Description: Upper limit of the exposure period, step size is 0.1.

Additionally, the range is internally limited by the selected measurement frequency.

Parameter: double SA_Measrate

SA_Measrate

Direction: Up

Valid values:

0.25

0.5

1.0

2.0 (only at ILD1320 and ILD1420)

4.0 (only at ILD1420)

Unit: kHz

Valid for sensor:

ILD1220

ILD1320

ILD1420

Description: Samplerate of measurement.

Parameter: double SA_Measrate

SA_Measrate

Direction: Up

Valid values:
Minimum: 0.3 at ILD1750, 0.25 at ILD1900

Maximum: 7.5 at ILD1750, 10.0 at ILD1900

Unit: kHz

Valid for sensor:

ILD1750

ILD1900

Description: Samplerate of measurement.

Parameter: int32_t SA_LaserPower SA_LaserPower

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Full
- 1= Off
- 2= Reduced (only at ILD1750 and ILD1900)
- 3= Medium (only at ILD1900)

Valid for sensor:

- ILD1220 (from firmware version 001.064)
- ILD1320
- ILD1420
- ILD1750
- ILD1900

Description: Laser power.

Parameter: int32_t SA_ROIStart SA_ROIStart

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 511

Unit: Pixel

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Description: First position on CCD.

Parameter: int32_t SA_ROIEnd SA_ROIEnd

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 511

Unit: Pixel

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Description: Last position on CCD.

Parameter: int32_t SA_AveragingType SA_AveragingType

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Moving average (MOVING)
- 2= Recursive averaging (RECURSIVE)
- 3= Median

Valid for sensor:

- ILD1420
- ILD1750

Description: Averaging type.

Parameter: int32_t SA_MovingCount SA_MovingCount

Direction: Up

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128

Valid for sensor:

- ILD1420

- ILD1750

Description: Number of value for the averaging window.

This parameter is only available at moving average.

Parameter: int32_t SA_RecursiveCount SA_RecursiveCount

Direction: Up

Valid values:

- Minimum:** 2
- Maximum:** 32768

Valid for sensor:

- ILD1420

- ILD1750

Description: Number of values for recursive averaging.

This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount SA_MedianCount

Direction: Up

Valid values:

- 3
- 5
- 7
- 9

Valid for sensor:

- ILD1420

- ILD1750

Description: Number of values to build median.

This parameter is only available at median.

Parameter: int32_t SA_Master SA_Master

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = no (NONE)
- 1 = yes (MASTER)

Valid for sensor:

- ILD1220 (from firmware version 001.064)

- ILD1320

- ILD1420

Description: Specifies if mastering is active.

Parameter: double SA_MasterValue	SA_MasterValue
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2 * measuring range	
Unit: mm	
Valid for sensor:	
ILD1220 (from firmware version 001.064)	
ILD1320	
ILD1420	
Description: Master value.	
Parameter: String SA_MasterSignalTable	SA_MasterSignalTable
Direction: Up	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_MasterSignalEntries	SA_MasterSignalEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MasterSignalEntry1, SA_MasterSignalEntry2, ...	
Parameter: String SA_MasterSignalEntry _{1..x}	SA_MasterSignalEntry _{1..x}
Direction: Up	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Each master signal as complete line.	
Parameter: int32_t SA_MasterSignalIndex _{1..x}	SA_MasterSignalIndex _{1..x}
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Distance 1	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Index of each master signal.	
Parameter: double SA_MasterValue _{1..x}	SA_MasterValue _{1..x}
Direction: Up	
Valid values:	
Minimum: -2* measuring range	
Maximum: +2* measuring range	
Unit: mm	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Master value.	

Parameter: int32_t SA_MasterSource	SA_MasterSource
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = None	
1 = Multi function input (MFI)	
2 = Select key (KEY_SELECT)	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Port which is used for mastering.	
Parameter: String SA_MasterTable	SA_MasterTable
Direction: Up	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_MasterEntries	SA_MasterEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MasterEntry1, SA_MasterEntry2, ...	
Parameter: String SA_MasterEntry _{1..x}	SA_MasterEntry _{1..x}
Direction: Up	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Each master as complete line.	
Parameter: int32_t SA_MasterIndex _{1..x}	SA_MasterIndex _{1..x}
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Distance 1	
1 = All	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Index of each master.	
Parameter: int32_t SA_MasterAction _{1..x}	SA_MasterAction _{1..x}
Direction: Up	
Valid values:	
0 = Inactive	
1 = Active	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Mastering action.	

Parameter: int32_t SA_DataOutInterface	SA_DataOutInterface
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None (at ILD1220 from firmware version 001.062)	
1= RS422 (at ILD1220 from firmware version 001.062)	
6= Analog	
Valid for sensor:	
ILD1220	
ILD1320	
ILD1420	
Description:	Active interface for data output.
Parameter: int32_t SA_OutputRS422	SA_OutputRS422
Direction: Up	
Valid values:	
0= No	
1= Yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description:	Output active at RS422 interface.
Parameter: int32_t SA_OutputAnalog	SA_OutputAnalog
Direction: Up	
Valid values:	
0= No	
1= Yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description:	Output active at analog interface.
Parameter: int32_t SA_OutputErrorOut1	SA_OutputErrorOut1
Direction: Up	
Valid values:	
0= No	
1= Yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description:	Output active at error output 1.
Parameter: int32_t SA_OutputErrorOut2	SA_OutputErrorOut2
Direction: Up	
Valid values:	
0= No	
1= Yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description:	Output active at error output 2.

Parameter: int32_t SA_Resampling SA_Resampling

Direction: Up

Valid values:

Minimum: 1

Maximum: 3000000

Valid for sensor:

ILD1420

ILD1750

ILD1900

Description: Resampling value.

Parameter: int32_t SA_ResampleRS422 SA_ResampleRS422

Direction: Up

Valid values:

0= no

1= yes

Valid for sensor:

ILD1420

ILD1750

ILD1900

Description: RS422 output is resampled.

Parameter: int32_t SA_ResampleAnalog SA_ResampleAnalog

Direction: Up

Valid values:

0= no

1= yes

Valid for sensor:

ILD1420

ILD1750

ILD1900

Description: Analog output is resampled.

Parameter: int32_t SA_ResampleChart SA_ResampleChart

Direction: Up

Valid values:

0= no

1= yes

Valid for sensor:

ILD1750

ILD1900

Description: Web output is resampled.

Parameter: int32_t SA_HoldLastValid SA_HoldLastValid

Direction: Up

Valid values:

Minimum: -1

Maximum: 1024

Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).

Parameter: int32_t SA_OutputDistance1_RS422	SA_OutputDistance1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1220 (from firmware version 001.062)	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description:	Specify if distance 1 is transmitted.
Parameter: int32_t SA_OutputAdditionalIntensity1_RS422	SA_OutputAdditionalIntensity1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
Description:	Specify if intensity 1 is transmitted.
Parameter: int32_t SA_OutputAdditionalIntensity_RS422	SA_OutputAdditionalIntensity_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1900	
Description:	Specify if intensity is transmitted.
Parameter: int32_t SA_OutputAdditionalDistanceRaw1_RS422	SA_OutputAdditionalDistanceRaw1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description:	Specify if uncalibrated distance 1 is transmitted.
Parameter: int32_t SA_OutputAdditionalShutterTime_RS422	SA_OutputAdditionalShutterTime_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description:	Specify if shutter time is transmitted.

Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1220 (from firmware version 001.062)	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalMeasrate_RS422	SA_OutputAdditionalMeasrate_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Specify if measrate is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestampLo_RS422	SA_OutputAdditionalTimestampLo_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description: Specify if timestamp (lower 16 bit) is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestampHi_RS422	SA_OutputAdditionalTimestampHi_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description: Specify if timestamp (upper 16 bit) is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if timestamp is transmitted.	

Parameter: int32_t SA_OutputAdditionalIntensity_RS422	SA_OutputAdditionalIntensity_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description:	Specify if intensity is transmitted.
Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description:	Specify if state is transmitted.
Parameter: int32_t SA_OutputAdditionalDistanceRaw_RS422	SA_OutputAdditionalDistanceRaw_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description:	Specify if uncalibrated distance is transmitted.
Parameter: int32_t SA_OutputDistanceUnlinearized_RS422	SA_OutputDistanceUnlinearized_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description:	Specify if unlinearized distance is transmitted.
Parameter: int32_t SA_OutputVideoRaw_RS422	SA_OutputVideoRaw_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1420	
ILD1750	
ILD1900	
Description:	Specify if raw video signal is transmitted.

Parameter: int32_t SA_ErrorOutput1 SA_ErrorOutput1

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Output trigger disabled (NONE) (only at ILD1220, ILD1320 and ILD1420)
- 1= No valid distance (no peak found, out of range) (DIST)
- 2= Distance is out of scaled analog range (TEACH)
- 3= Distance is above set threshold (L1)

Description: Condition for error output.

Parameter: int32_t SA_ErrorOutput2 SA_ErrorOutput2

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 1= No valid distance (no peak found, out of range) (DIST)
- 2= Distance is out of scaled analog range (TEACH)
- 3= Distance is above set threshold (L1)

Valid for sensor:

ILD1750
ILD1900

Description: Condition for second error output.

Parameter: int32_t SA_DataSource SA_DataSource

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Distance 1

Valid for sensor:

ILD1220
ILD1320
ILD1420

Description: Data source to be checked.

Parameter: double SA_UpperLimit SA_UpperLimit

Direction: Up

Valid values:

- Minimum:** -2* measuring range
Maximum: +2* measuring range

Unit: mm

Valid for sensor:

ILD1220
ILD1320
ILD1420

Description: Upper limit.

Parameter: double SA_LowerLimit_{1..2} SA_LowerLimit_{1..2}

Direction: Up

Valid values:

- Minimum:** -2* measuring range
Maximum: +2* measuring range

Unit: mm

Valid for sensor:

ILD1750
ILD1900

Description: Lower limit.

Parameter: double SA_UpperLimit1..2	SA_UpperLimit1..2
Direction: Up	
Valid values:	
Minimum: -2* measuring range	
Maximum: +2* measuring range	
Unit: mm	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Upper limit.	
Parameter: int32_t SA_ErrorLimitCompOp1..2	SA_ErrorLimitCompOp1..2
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Lower	
1 = Upper	
2 = Both	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Compare operation.	
Parameter: double SA_ErrorHysteresis	SA_ErrorHysteresis
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2* measuring range	
Unit: mm	
Description: Error hysteresis.	
Parameter: int32_t SA_ErrorOutHoldTime	SA_ErrorOutHoldTime
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1000	
Unit: ms	
Description: Minimum hold period.	
Parameter: int32_t SA_ErrorLevelOut1	SA_ErrorLevelOut1
Direction: Up	
Valid values:	
0 = NPN	
1 = PNP	
2 = Push-Pull (PUSHPULL)	
3 = Push-Pull negated (PUSHPULLNEG)	
Description: Error level for out 1.	

Parameter: int32_t SA_ErrorLevelOut2	SA_ErrorLevelOut2
Direction: Up	
Valid values:	
0= NPN	
1= PNP	
2= Push-Pull (PUSHPULL)	
3= Push-Pull negated (PUSHPULLNEG)	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Error level for out 2.	
Parameter: int32_t SA_AnalogRange	SA_AnalogRange
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
1= 0 - 5V	
2= 0 - 10V	
5= 4 - 20mA	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Analog output range.	
Parameter: int32_t SA_AnalogScaleMode	SA_AnalogScaleMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Standard	
1= Two point (TWOPOINT)	
Description: Analog scale mode.	
Parameter: double SA_MinValue	SA_MinValue
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2* measuring range	
Unit: mm	
Valid for sensor:	
ILD1220	
ILD1320	
ILD1420	
Description: Value which represents lowest voltage/current (at two point scaling).	
Parameter: double SA_MaxValue	SA_MaxValue
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2* measuring range	
Unit: mm	
Valid for sensor:	
ILD1220	
ILD1320	
ILD1420	
Description: Value which represents highest voltage/current (at two point scaling).	

Parameter: double SA_AnalogScaleRangeLowerLimit	SA_AnalogScaleRangeLowerLimit
Direction: Up	
Valid values:	
Minimum: -2* measuring range	
Maximum: +2* measuring range	
Unit: mm	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Value which represents lowest voltage/current (at two point scaling).	
Parameter: double SA_AnalogScaleRangeUpperLimit	SA_AnalogScaleRangeUpperLimit
Direction: Up	
Valid values:	
Minimum: -2* measuring range	
Maximum: +2* measuring range	
Unit: mm	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Value which represents highest voltage/current (at two point scaling).	
Parameter: int32_t SA_AnalogScaleSource	SA_AnalogScaleSource
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Multi function input (MFI)	
2= Select key (KEY_SELECT)	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Port which is used analog scaling.	
Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Description: Name of the sensor.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the sensor.	

Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the sensor.	
Parameter: String SA_CableHead	SA_CableHead
Direction: Up	
Valid for sensor:	
ILD1220	
ILD1320	
ILD1420	
Description: Sensor cable type.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Unit: mm	
Description: Range of the sensor.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the sensor.	
Parameter: String SA_HardwareRevision	SA_HardwareRevision
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of the controller board.	
Parameter: String SA_BuildID	SA_BuildID
Direction: Up	
Description: Build ID	
Parameter: String SA_BuildTimestamp	SA_BuildTimestamp
Direction: Up	
Description: Build timestamp	
Parameter: String SA_BootVersion	SA_BootVersion
Direction: Up	
Description: Boot version	
Parameter: String SA_ComputationTable	SA_ComputationTable
Direction: Up	
Valid for sensor:	
ILD1900	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_ComputationEntries	SA_ComputationEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2	
Valid for sensor:	
ILD1900	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_ComputationEntry1, SA_ComputationEntry2, ...	

Parameter: String SA_ComputationEntry1..x	SA_ComputationEntry1..x
Direction: Up	
Valid for sensor:	
ILD1900	
Description: Each computation as complete line.	
Parameter: int32_t SA_ComputationQueue1..x	SA_ComputationQueue1..x
Direction: Up	
Valid values:	
0= Channel 1	
Valid for sensor:	
ILD1900	
Description: Computation queue	
Parameter: int32_t SA_ComputationID1..x	SA_ComputationID1..x
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2	
Valid for sensor:	
ILD1900	
Description: Index within computation queue	
Parameter: int32_t SA_ComputationOperation1..x	SA_ComputationOperation1..x
Direction: Up	
Valid values:	
0= None	
1= Median	
2= Moving average	
3= Recursive averaging	
Valid for sensor:	
ILD1900	
Description: Computation operation	
Parameter: int32_t SA_ComputationSignalIndex1..x	SA_ComputationSignalIndex1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_ComputationSignalName1..x	
0= Distance 1 of first Channel	
Valid for sensor:	
ILD1900	
Description: Signal index.	
Parameter: String SA_ComputationSignalName1..x	SA_ComputationSignalName1..x
Direction: Up	
Valid for sensor:	
ILD1900	
Description: User defined signal.	
Parameter: int32_t SA_ComputationMovingCount1..x	SA_ComputationMovingCount1..x
Direction: Up	
Valid values:	
2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096	
Valid for sensor:	
ILD1900	
Description: Number of value for the averaging window. This parameter is only used at moving average.	

Parameter: int32_t SA_ComputationRecursiveCount1..x	SA_ComputationRecursive- Count1..x
Direction: Up	
Valid values:	
Minimum: 2	
Maximum: 32000	
Valid for sensor:	
ILD1900	
Description:	Number of values for recursive averaging. This parameter is only used at recursive average.
Parameter: int32_t SA_ComputationMedianCount1..x	SA_ComputationMedian- Count1..x
Direction: Up	
Valid values:	
3, 5, 7, 9	
Valid for sensor:	
ILD1900	
Description:	Number of values to build median. This parameter is only used at median.
Parameter: String SA_CurrentName	SA_CurrentName
Direction: Up	
Description:	Name of current (active) setting.
Parameter: String SA_SettingNames	SA_SettingNames
Direction: Up	
Description:	List (separated by new line).
Parameter: String SA_PresetNames	SA_PresetNames
Direction: Up	
Description:	List (separated by new line).
Parameter: int32_t SA_Automatic	SA_Automatic
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Automatic selection.
Parameter: String SA_InitialName	SA_InitialName
Direction: Up	
Description:	Name of setting.
Parameter: int32_t SA_PresetMode	SA_PresetMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Static (only at ILD1320, ILD1420, ILD1750 and ILD1900)	
1= Balanced	
2= Dynamic	
3= None (if no preset setting is active)	
4= No averaging (only at ILD1220, ILD1320 and ILD1420)	
Description:	Preset mode.

10.3.1.2 User level

10.3.1.2.1 Logout (LOGOUT)

Change user level to user.

10.3.1.2.2 Login (LOGIN)

Change user level to professional.

Parameter: String SP_Password

SP_Password

Direction: Down

Description: Valid password to login.

10.3.1.2.3 Get_UserLevel (GETUSERLEVEL)

Retrieve actual user level.

Parameter: int32_t SA_UserLevel

SA_UserLevel

Direction: Up

Valid values:

-1 = Other user level (only for internal use)

0 = User

1 = Professional

Description: Actual user level.

10.3.1.2.4 Set_DefaultUser (STDUSER)

Set the default user level after booting the system.

Parameter: int32_t SP_DefaultUser

SP_DefaultUser

Direction: Down

Valid values:

0 = User

1 = Professional

Description: Default user level.

10.3.1.2.5 Get_DefaultUser (STDUSER)

Get the default user level after booting the system.

Parameter: int32_t SA_DefaultUser

SA_DefaultUser

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor

0 = User

1 = Professional

Description: Default user level.

10.3.1.2.6 Set_Password (PASSWD)

Change the password for login.

Parameter: String SP_OldPassword SP_OldPassword
Direction: Down
Description: Old password.

Parameter: String SP_NewPassword SP_NewPassword
Direction: Down
Description: New password.

10.3.1.3 Triggering

10.3.1.3.1 Set_TriggerSource (TRIGGERSOURCE)

Set the trigger source (input).

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_TriggerSource SP_TriggerSource
Direction: Down
Valid values:
 0= None
 1= Multi function input (MFI)
 2= Sync in-/output (SYNCIO)
 3= Software
Description: Trigger source (input).

10.3.1.3.2 Get_TriggerSource (TRIGGERSOURCE)

Get the active trigger source (input).

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_TriggerSource SA_TriggerSource
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= None
 1= Multi function input (MFI)
 2= Sync in-/output (SYNCIO)
 3= Software
Description: Trigger source (input).

10.3.1.3.3 Set_TriggerMode (TRIGGERMODE)

Set the trigger mode.

Valid for sensor:

- ILD1750
- ILD1900

Parameter: int32_t SP_TriggerMode

SP_TriggerMode

Direction: Down

Valid values:

- 0= Edge
- 1= Level (PULSE)

Description: Trigger mode.

10.3.1.3.4 Get_TriggerMode (TRIGGERMODE)

Get the active trigger mode.

Valid for sensor:

- ILD1750
- ILD1900

Parameter: int32_t SA_TriggerMode

SA_TriggerMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Edge
- 1= Level (PULSE)

Description: Trigger mode.

10.3.1.3.5 Set_TriggerMode (TRIGGER)

Set the trigger mode.

Valid for sensor:

- ILD1220
- ILD1320
- ILD1420

Parameter: int32_t SP_TriggerMode

SP_TriggerMode

Direction: Down

Valid values:

- 0= None
- 1= Edge
- 2= Level (PULSE)
- 3= Software (only at ILD1320 and ILD1420)

Description: Trigger mode.

10.3.1.3.6 Get_TriggerMode (TRIGGER)

Get the active trigger mode.

Valid for sensor:

- ILD1220
- ILD1320
- ILD1420

Parameter: int32_t SA_TriggerMode

SA_TriggerMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = None
- 1 = Edge
- 2 = Level (PULSE)
- 3 = Software (only at ILD1320 and ILD1420)

Description: Trigger mode.

10.3.1.3.7 Set_TriggerMoment (TRIGGERAT)

Set the trigger time.

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SP_TriggerMoment

SP_TriggerMoment

Direction: Down

Valid values:

- 0 = Input
- 1 = Output

Description: Trigger moment.

10.3.1.3.8 Get_TriggerMoment (TRIGGERAT)

Get the active trigger time.

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SA_TriggerMoment

SA_TriggerMoment

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Input
- 1 = Output

Description: Trigger moment.

10.3.1.3.9 Set_TriggerLevel (TRIGGERLEVEL)

Set the trigger level.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_TriggerLevel

SP_TriggerLevel

Direction: Down

Valid values:

0= High
1= Low

Description: Trigger level.

10.3.1.3.10 Get_TriggerLevel (TRIGGERLEVEL)

Get the active trigger level.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_TriggerLevel

SA_TriggerLevel

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= High
1= Low

Description: Trigger level.

10.3.1.3.11 Set_TriggerCount (TRIGGERCOUNT)

Set the number of values to measure at trigger.

Parameter: int32_t SP_TriggerCount

SP_TriggerCount

Direction: Down

Valid values:

Minimum: 0 (at ILD1220, ILD1320 and ILD1420), 1 (at ILD1750 and ILD1900)

Maximum: 16383

Description: Number of values to measure. 0 means no trigger (NONE), 16383 means endless measurement (INFINITE).

10.3.1.3.12 Get_TriggerCount (TRIGGERCOUNT)

Get the number of values to measure at trigger.

Parameter: int32_t SA_TriggerCount

SA_TriggerCount

Direction: Up

Valid values:

Minimum: 0 (at ILD1220, ILD1320 and ILD1420), 1 (at ILD1750 and ILD1900)

Maximum: 16383

Description: Number of values to measure. 0 means no trigger (NONE), 16383 means endless measurement (INFINITE).

10.3.1.3.13 Software_Trigger (TRIGGERSW)

Execute a software trigger.

Valid for sensor:

ILD1320
ILD1420

10.3.1.3.14 Set_SoftwareTrigger (TRIGGERSW)

Set software trigger mode.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_SoftwareTriggerMode

SP_SoftwareTriggerMode

Direction: Down

Valid values:

0= Clear
1= Set

Description: Software trigger mode.

10.3.1.3.15 Get_SoftwareTrigger (TRIGGERSW)

Get software trigger mode.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_SoftwareTriggerMode

SA_SoftwareTriggerMode

Direction: Up

Valid values:

0= Clear
1= Set

Description: Software trigger mode.

10.3.1.4 Interfaces

10.3.1.4.1 Set_Baudrate (BAUDRATE)

Set baudrate of sensor for serial RS422 communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

4000000 (only at ILD1750 and ILD1900)
3000000 (only at ILD1750 and ILD1900)
2000000 (only at ILD1750 and ILD1900)

1000000 (only at ILD1220, ILD1320 and ILD1420)

921600

691200

460800

256000 (only at ILD1220, ILD1320 and ILD1420)

230400

128000 (only at ILD1220, ILD1320 and ILD1420)

115200

56000 (only at ILD1220, ILD1320 and ILD1420)

19200 (only at ILD1220, ILD1320 and ILD1420)

9600

Unit: Baud

Description: Baudrate of sensor.

Parameter: int32_t CP_InterruptDataTransfer

CP_InterruptDataTransfer

Direction: Down

Valid values:

0= false

1= true

Default: 0

Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Set_DataOutInterface](#)) and enabling it again at end (only if enabled before).

10.3.1.4.2 Get_Baudrate (BAUDRATE)

Get baudrate of sensor for serial RS422 communication.

Parameter: int32_t SA_SensorBaudrate

SA_SensorBaudrate

Direction: Up

Valid values:

4000000 (only at ILD1750 and ILD1900)

3000000 (only at ILD1750 and ILD1900)

2000000 (only at ILD1750 and ILD1900)

1000000 (only at ILD1220, ILD1320 and ILD1420)

921600

691200

460800

256000 (only at ILD1220, ILD1320 and ILD1420)

230400

128000 (only at ILD1220, ILD1320 and ILD1420)

115200

56000 (only at ILD1220, ILD1320 and ILD1420)

19200 (only at ILD1220, ILD1320 and ILD1420)

9600

Unit: Baud

Description: Baudrate of sensor.

10.3.1.4.3 Set_ApplLanguage (LANGUAGE)

Set language of web interface.

Parameter: int32_t SP_ApplicationLanguage

SP_ApplicationLanguage

Direction: Down

Valid values:

0= English (EN)

1= German (DE)

Description: Language of web interface.

10.3.1.4.4 Get_ApplLanguage (LANGUAGE)

Get language of web interface.

Parameter: int32_t SA_ApplicationLanguage

SA_ApplicationLanguage

Direction: Up

Valid values:

0= English (EN)

1= German (DE)

Description: Language of web interface.

10.3.1.5 Parameter management

10.3.1.5.1 Save_InterfaceParameters (BASICSETTINGS STORE)

Save actual interface parameters at controller.

10.3.1.5.2 Load_InterfaceParameters (BASICSETTINGS READ)

Load stored interface parameters into controller RAM.

10.3.1.5.3 Save_MeasureParameters (MEASSETTINGS STORE)

Save actual measurement parameters at controller.

Parameter: String SP_SettingName

SP_SettingName

Direction: Down

Description: Name of setting. If name is empty, current setting will be overwritten.

10.3.1.5.4 Load_MeasureParameters (MEASSETTINGS READ)

Load stored measurement parameters into controller RAM.

At this command the controller may change output data after applying new setting.

If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: String SP_SettingName

SP_SettingName

Direction: Down

Description: Name of setting.

10.3.1.5.5 Rename_MeasureParameters (MEASSETTINGS RENAME)

Rename stored measurement parameters at controller.

Parameter: String SP_OldName SP_OldName
Direction: Down
Description: Actual name of setting.

Parameter: String SP_NewName SP_NewName
Direction: Down
Description: New name of setting.

Parameter: int32_t SP_Overwrite SP_Overwrite
Direction: Down
Valid values:
 0 = no
 1 = yes
Description: Force overwriting existing setting.

10.3.1.5.6 Get_CurrentMeasureSetting (MEASSETTINGS CURRENT)

Get name of current (active) measurement parameters.

Parameter: String SA_CurrentName SA_CurrentName
Direction: Up
Description: Name of current (active) setting.

10.3.1.5.7 Get_MeasureSettingsList (MEASSETTINGS LIST)

Get list of all user settings.

Parameter: String SA_SettingNames SA_SettingNames
Direction: Up
Description: List (separated by new line).

10.3.1.5.8 Delete_MeasureParameters (MEASSETTINGS DELETE)

Delete stored measurement parameters at controller.

Parameter: String SP_SettingName SP_SettingName
Direction: Down
Description: Name of setting.

10.3.1.5.9 Get_MeasurePresetList (MEASSETTINGS PRESETLIST)

Get list of all preset settings.

Parameter: String SA_PresetNames SA_PresetNames
Direction: Up
Description: List (separated by new line).

10.3.1.5.10 Set_InitialMeasureSetting (MEASSETTINGS INITIAL)

Set initial (boot time) measure setting.

Parameter: int32_t SP_Automatic SP_Automatic

Direction: Down

Valid values:

0= no

1= yes

Description: Automatic selection.

Parameter: String SP_InitialName SP_InitialName

Direction: Down

Description: Name of setting.

10.3.1.5.11 Get_InitialMeasureSetting (MEASSETTINGS INITIAL)

Get initial (boot time) measure setting.

Parameter: int32_t SA_Automatic SA_Automatic

Direction: Up

Valid values:

0= no

1= yes

Description: Automatic selection.

Parameter: String SA_InitialName SA_InitialName

Direction: Up

Description: Name of setting.

10.3.1.5.12 Set_MeasurePresetMode (MEASSETTINGS PRESETMODE)

Set mode of current preset settings.

Parameter: int32_t SP_PresetMode SP_PresetMode

Direction: Down

Valid values:

0= Static (only at ILD1320, ILD1420, ILD1750 and ILD1900)

1= Balanced

2= Dynamic

3= None (if no preset setting is active)

4= No averaging (only at ILD1220, ILD1320 and ILD1420)

Description: Preset mode.

10.3.1.5.13 Get_MeasurePresetMode (MEASSETTINGS PRESETMODE)

Get mode of current preset settings.

Parameter: int32_t SA_PresetMode SA_PresetMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Static (only at ILD1320, ILD1420, ILD1750 and ILD1900)
- 1 = Balanced
- 2 = Dynamic
- 3 = None (if no preset setting is active)
- 4 = No averaging (only at ILD1220, ILD1320 and ILD1420)

Description: Preset mode.

10.3.1.5.14 Set_Default (SETDEFAULT)

Reset the sensor to default settings.

At this command the sensor may change output data after applying default settings. If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DefaultType SP_DefaultType

Direction: Down

Valid values:

- 0 = Delete all settings and load the factory settings (ALL)
- 1 = Delete all measuring settings (MEASSETTINGS)
- 2 = Delete all basic settings (BASICSETTINGS)

Description: Specifies which settings should be reset.

10.3.1.5.15 Export_Parameters (EXPORT)

Exports the settings of the sensor.

Parameter: int32_t SP_ExportType SP_ExportType

Direction: Down

Valid values:

- 0 = Exports only the measuring settings with name (MEASSETTINGS)
- 1 = Exports only the basic settings (BASICSETTINGS)
- 2 = Exports all measuring settings (MEASSETTINGS_ALL)
- 3 = Exports basic settings and all measuring settings (ALL)

Description: Export type.

Parameter: String SP_SettingName SP_SettingName

Direction: Down

Description: Name of setting to be exported.

This parameter is only used at export type 0.

Parameter: String SA_ExportData SA_ExportData

Direction: Up

Description: Exported data in ASCII format.

10.3.1.5.16 Import_Parameters (IMPORT)

Imports the settings of the sensor.

If SP_ApplyImmediately is 1, the controller may change output data after applying new setting. If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_ForceOverwrite

SP_ForceOverwrite

Direction: Down

Valid values:

0= Do not overwrite existing setting

1= Allow to overwrites existing settings (FORCE)

Description: Specify if existing settings can be overwritten.

Parameter: int32_t SP_ApplyImmediately

SP_ApplyImmediately

Direction: Down

Valid values:

0= Just store imported settings.

1= Apply the imported settings (APPLY)

Description: Specify if settings only should be stored or additionally applied.

Parameter: String SP_ImportData

SP_ImportData

Direction: Down

Description: Data to be imported (from a former call to Export_Parameters).

10.3.1.6 Internal controller commands

10.3.1.6.1 Set_MultiFunctionInputMode (MFIFUNC)

Set the mode of multi function input at sensor.

Valid for sensor:

ILD1220

ILD1320

ILD1420

Parameter: int32_t SP_MultiFunctionInputMode

SP_MultiFunctionInputMode

Direction: Down

Valid values:

0= MFI has no function (NONE)

1= Use MFI to master the measuring value (MASTER)

2= Use MFI to teach the analog output (TEACH)

3= Use MFI to trigger the measuring process (TRIGGER)

Description: Specifies how to use the multi function input.

10.3.1.6.2 Get_MultiFunctionInputMode (MFIFUNC)

Get the mode of multi function input at sensor.

Valid for sensor:

- ILD1220
- ILD1320
- ILD1420

Parameter: int32_t SA_MultiFunctionInputMode

SA_MultiFunctionInputMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = MFI has no function (NONE)
- 1 = Use MFI to master the measuring value (MASTER)
- 2 = Use MFI to teach the analog output (TEACH)
- 3 = Use MFI to trigger the measuring process (TRIGGER)

Description: Specifies how to use the multi function input.

10.3.1.6.3 Set_MultiFunctionInputLevel (MFILEVEL)

Set level of multi function input.

Parameter: int32_t SP_MultiFunctionInputLevel

SP_MultiFunctionInput-
Level

Direction: Down

Valid values:

- 0 = High (HTL_HIGH at ILD1220, ILD1320 and ILD1420, HTL at ILD1750 and ILD1900)
- 1 = Low (HTL_LOW at ILD1220, ILD1320 and ILD1420, TTL at ILD1750 and ILD1900)

Description: Multi function input level.

10.3.1.6.4 Get_MultiFunctionInputLevel (MFILEVEL)

Get level of multi function input.

Parameter: int32_t SA_MultiFunctionInputLevel

SA_MultiFunctionInput-
Level

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = High (HTL_HIGH at ILD1220, ILD1320 and ILD1420, HTL at ILD1750 and ILD1900)
- 1 = Low (HTL_LOW at ILD1220, ILD1320 and ILD1420, TTL at ILD1750 and ILD1900)

Description: Multi function input level.

10.3.1.6.5 Set_InitialChartType (CHARTTYPE_INITIAL)

Set initial type of the chart at web interface.

Valid for sensor:

ILD1320
ILD1420

Parameter: int32_t SP_InitialChartType

SP_InitialChartType

Direction: Down

Valid values:

0= Measure (MEAS)
1= Video signal (VIDEO)
2= Automatic (AUTO)

Description: Initial type of the chart at web interface.

10.3.1.6.6 Get_InitialChartType (CHARTTYPE_INITIAL)

Get initial type of the chart at web interface.

Valid for sensor:

ILD1320
ILD1420

Parameter: int32_t SA_InitialChartType

SA_InitialChartType

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= Measure (MEAS)
1= Video signal (VIDEO)
2= Automatic (AUTO)

Description: Initial type of the chart at web interface.

10.3.1.6.7 Set_ChartType (CHARTTYPE)

Set current type of the chart at web interface.

Valid for sensor:

ILD1320
ILD1420
ILD1750
ILD1900

Parameter: int32_t SP_ChartType

SP_ChartType

Direction: Down

Valid values:

0= Measure (MEAS)
1= Video signal (VIDEO)

Description: Type of the chart at web interface.

10.3.1.6.8 Get_ChartType (CHARTTYPE)

Get current type of the chart at web interface.

Valid for sensor:

- ILD1320
- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SA_ChartType

SA_ChartType

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Measure (MEAS)
- 1 = Video signal (VIDEO)

Description: Type of the chart at web interface.

10.3.1.6.9 Set_TargetMode (TARGETMODE)

Selecting a target loads a predefined sensor configuration which achieves the best results for the selected material.

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SP_TargetMode

SP_TargetMode

Direction: Down

Valid values:

- 0 = Standard Target (STANDARD)
- 1 = Multi-Surface Target (MULTISURFACE)
- 2 = Light Penetration Target (PENETRATION)

Description: Target mode.

10.3.1.6.10 Get_TargetMode (TARGETMODE)

Target for a predefined sensor configuration which achieves the best results for the selected material.

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SA_TargetMode

SA_TargetMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Standard Target (STANDARD)
- 1 = Multi-Surface Target (MULTISURFACE)
- 2 = Light Penetration Target (PENETRATION)

Description: Target mode.

10.3.1.6.11 Get_HTTP (GETHTTP)

Retrieve HTTP data from sensor.

Parameter: String SP_HTPPPath	SP_HTPPPath
Direction: Down	
Description: HTTP file name.	
Parameter: int32_t SP_HTTPMaxLength	SP_HTTPMaxLength
Direction: Down	
Valid values:	
Minimum: 512	
Maximum: 65535	
Unit: Bytes	
Description: Maximum transfer length.	
Parameter: String SP_HTTPEtage	SP_HTTPEtage
Direction: Down	
Description: HTTP eTag (data is not transmitted if eTag does match).	
Parameter: int32_t CP_HTTPRetryCount	CP_HTTPRetryCount
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Default: 5	
Description: Maximum retry count on error (0 means do not retry).	
Parameter: String SA_HTTPHeader	SA_HTTPHeader
Direction: Up	
Description: HTTP header.	
Parameter: Binary data SA_HTTPBody	SA_HTTPBody
Direction: Up	
Description: HTTP body.	

10.3.2 Measurement

10.3.2.1 General

10.3.2.1.1 Set_MeasureMode (MEASMODE)

Set the measure mode.

Valid for sensor:

ILD1900 (only direct reflection variant)

Parameter: int32_t SP_MeasureMode	SP_MeasureMode
Direction: Down	
Valid values:	
0= Diffuse	
1= Direct	
Description: Measure mode.	

10.3.2.1.2 Get_MeasureMode (MEASMODE)

Get the measure mode.

Valid for sensor:

ILD1900 (only direct reflection variant)

Parameter: int32_t SA_MeasureMode

SA_MeasureMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Diffuse
- 1 = Direct

Description: Measure mode.

10.3.2.1.3 Set_MeasurePeak (MEASPEAK)

Select the peak to measure.

Valid for sensor:

ILD1320

ILD1420

ILD1750

ILD1900

Parameter: int32_t SP_MeasurePeak

SP_MeasurePeak

Direction: Down

Valid values:

- 0 = Greatest Amplitude (DISTA)
- 1 = First Peak (DIST1)
- 2 = Last Peak (DISTL)
- 3 = Largest Area (DISTW) (only at ILD1750 and ILD1900)

Description: Peak to measure.

10.3.2.1.4 Get_MeasurePeak (MEASPEAK)

Get the selected peak to measure.

Valid for sensor:

ILD1320

ILD1420

ILD1750

ILD1900

Parameter: int32_t SA_MeasurePeak

SA_MeasurePeak

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Greatest Amplitude (DISTA)
- 1 = First Peak (DIST1)
- 2 = Last Peak (DISTL)
- 3 = Largest Area (DISTW) (only at ILD1750 and ILD1900)

Description: Peak to measure.

10.3.2.1.5 Set_PeakWidthMax (PEAK_WIDTHMAX)

Set the global maximal peak width threshold. A valid peak must be below this threshold value.

This command is only available at elevated user mode Professional+.

Valid for sensor:

ILD1900

Parameter: int32_t SP_PeakWidthMax

SP_PeakWidthMax

Direction: Down

Valid values:

Minimum: 1

Maximum: 512

Unit: Pixel

Description: Maximum peak width.

10.3.2.1.6 Get_PeakWidthMax (PEAK_WIDTHMAX)

Get the global maximal peak width threshold. A valid peak must be below this threshold value.

This command is only available at elevated user mode Professional+.

Valid for sensor:

ILD1900

Parameter: int32_t SA_PeakWidthMax

SA_PeakWidthMax

Direction: Up

Valid values:

Minimum: 1

Maximum: 512

Unit: Pixel

Description: Maximum peak width.

10.3.2.1.7 Set_PeakGlobalThreshold (PEAK_THRESGLOB)

Set the global minimal threshold. A valid peak must be above this threshold value.

This command is only available at elevated user mode Professional+.

Parameter: double SP_PeakGlobalThreshold

SP_PeakGlobalThreshold

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 100.0

Unit: %

Description: Detection threshold.

10.3.2.1.8 Get_PeakGlobalThreshold (PEAK_THRESGLOB)

Get the global minimal threshold. A valid peak must be above this threshold value.
 This command is only available at elevated user mode Professional+.

Parameter: double SA_PeakGlobalThreshold

SA_PeakGlobalThreshold

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 100.0

Unit: %

Description: Detection threshold.

10.3.2.1.9 Set_ShutterMode (SHUTTERMODE)

Set the shutter mode.

Valid for sensor:

ILD1750

ILD1900

Parameter: int32_t SP_ShutterMode

SP_ShutterMode

Direction: Down

Valid values:

0= Exposure time is adapted automatically (MEAS)

1= Exposure time is set manually (MANUAL)

Description: Shutter mode.

10.3.2.1.10 Get_ShutterMode (SHUTTERMODE)

Get the shutter mode.

Valid for sensor:

ILD1750

ILD1900

Parameter: int32_t SA_ShutterMode

SA_ShutterMode

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Exposure time is adapted automatically (MEAS)

1= Exposure time is set manually (MANUAL)

Description: Shutter mode.

10.3.2.1.11 Set_Samplerate (MEASRATE)

Set the samplerate.

Parameter: double SP_Measrate SP_Measrate

Direction: Down

Valid values:

- 0.25
- 0.5
- 1.0
- 2.0 (only at ILD1320 and ILD1420)
- 4.0 (only at ILD1420)

Unit: kHz

Valid for sensor:

- ILD1220
- ILD1320
- ILD1420

Description: Samplerate of measurement.

Parameter: double SP_Measrate SP_Measrate

Direction: Down

Valid values:

- Minimum:** 0.3 at ILD1750, 0.25 at ILD1900
- Maximum:** 7.5 at ILD1750, 10.0 at ILD1900

Unit: kHz

Valid for sensor:

- ILD1750
- ILD1900

Description: Samplerate of measurement.

10.3.2.1.12 Get_Samplerate (MEASRATE)

Get the samplerate.

Parameter: double SA_Measrate SA_Measrate

Direction: Up

Valid values:

- 0.25
- 0.5
- 1.0
- 2.0 (only at ILD1320 and ILD1420)
- 4.0 (only at ILD1420)

Unit: kHz

Valid for sensor:

- ILD1220
- ILD1320
- ILD1420

Description: Samplerate of measurement.

Parameter: double SA_Measrate SA_Measrate
Direction: Up
Valid values:
Minimum: 0.3 at ILD1750, 0.25 at ILD1900
Maximum: 7.5 at ILD1750, 10.0 at ILD1900
Unit: kHz
Valid for sensor:
 ILD1750
 ILD1900
Description: Samplerate of measurement.

10.3.2.1.13 Set_ShutterTime (SHUTTER)

Set the fixed exposure times.

Valid for sensor:

ILD1750
 ILD1900

Parameter: double SP_ShutterTime SP_ShutterTime
Direction: Down
Valid values:
Minimum: 0.1
Maximum: 3333.0 at ILD1750, 4000.0 at ILD1900
Unit: μ s
Description: Exposure time.

10.3.2.1.14 Get_ShutterTime (SHUTTER)

Get the fixed exposure times.

Valid for sensor:

ILD1750
 ILD1900

Parameter: double SA_ShutterTime SA_ShutterTime
Direction: Up
Valid values:
Minimum: 0.1
Maximum: 3333.0 at ILD1750, 4000.0 at ILD1900
Unit: μ s
Description: Exposure time.

10.3.2.1.15 Set_ExposureMode (EXPOSUREMODE)

Set the exposure mode.

This command is only available at elevated user mode Professional+.

Valid for sensor:

ILD1900

Parameter: int32_t SP_ExposureMode SP_ExposureMode
Direction: Down
Valid values:
 0= Standard
 1= Intelligent
 2= Background (Background subtraction)
Description: Exposure mode.

10.3.2.1.16 Get_ExposureMode (EXPOSUREMODE)

Get the exposure mode.

This command is only available at elevated user mode Professional+.

Valid for sensor:

ILD1900

Parameter: int32_t SA_ExposureMode

SA_ExposureMode

Direction: Up

Valid values:

0= Standard

1= Intelligent

2= Background (Background subtraction)

Description: Exposure mode.

10.3.2.1.17 Set_ExposureLimit (EXPOSURELIMIT)

Set the upper limit of the exposure period limiting the controlled range.

This command is only available at elevated user mode Professional+.

Valid for sensor:

ILD1900

Parameter: double SP_ExposureLimit

SP_ExposureLimit

Direction: Down

Valid values:

Minimum: 0.1

Maximum: 4000.0

Unit: μs

Description: Upper limit of the exposure period, step size is 0.1.

Additionally, the range is internally limited by the selected measurement frequency.

10.3.2.1.18 Get_ExposureLimit (EXPOSURELIMIT)

Get the upper limit of the exposure period limiting the controlled range.

This command is only available at elevated user mode Professional+.

Valid for sensor:

ILD1900

Parameter: double SA_ExposureLimit

SA_ExposureLimit

Direction: Up

Valid values:

Minimum: 0.1

Maximum: 4000.0

Unit: μs

Description: Upper limit of the exposure period, step size is 0.1.

Additionally, the range is internally limited by the selected measurement frequency.

10.3.2.1.19 Set_LaserPower (LASERPOW)

Specify the laser power at sensor.

Valid for sensor:

- ILD1220 (from firmware version 001.064)
- ILD1320
- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SP_LaserPower

SP_LaserPower

Direction: Down

Valid values:

- 0= Full
- 1= Off
- 2= Reduced (only at ILD1750 and ILD1900)
- 3= Medium (only at ILD1900)

Description: Laser power.

10.3.2.1.20 Get_LaserPower (LASERPOW)

Get the laser power from sensor.

Valid for sensor:

- ILD1220 (from firmware version 001.064)
- ILD1320
- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SA_LaserPower

SA_LaserPower

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Full
- 1= Off
- 2= Reduced (only at ILD1750 and ILD1900)
- 3= Medium (only at ILD1900)

Description: Laser power.

10.3.2.1.21 Get_VideoStreamSignal

Read one video signal from video stream.

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SP_ReadMode

SP_ReadMode

Direction: Down

Valid values:

- 0= Each video signal
- 1= Only newest video signal
- 2= Automatic

Description: This mode specifies if each video signal should be read or only the latest one. If set to automatic each video signal is read until the buffer does not overflow. If the buffer becomes full one or more video signals are discarded.

Parameter: int32_t SP_WaitVideoTimeout

SP_WaitVideoTimeout

Direction: Down

Unit: ms

Valid values:

- Minimum:** 0
- Maximum:** 2147483647 (INT32_MAX)

Description: Timeout to wait for a video signal.

Parameter: Binary data SA_VideoRaw

SA_VideoRaw

Direction: Up

Valid values:

- 512 words (each 2 byte), each word is an intensity value.

Description: Raw video signal

Parameter: double SA_VideoTimestamp

SA_VideoTimestamp

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 1.79769e+308 (DBL_MAX)

Unit: ms

Description: Timestamp of the video signal. It starts from 1970 Jan 01 at 01:00. It is generated when the video has arrived at TCP/IP socket.

Parameter: int32_t SA_SkippedVideo

SA_SkippedVideo

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 2147483647 (INT32_MAX)

Description: Number of skipped video signals, if SP_ReadMode is not 0.

10.3.2.2 Video signal

10.3.2.2.1 Set_ROI (ROI)

Set the region of interest for processing video signal.

Valid for sensor:

ILD1420

ILD1750

ILD1900

Parameter: int32_t SP_ROIStart

SP_ROIStart

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 511

Unit: Pixel

Description: First pixel.

Parameter: int32_t SP_ROIEnd SP_ROIEnd
Direction: Down
Valid values:
Minimum: 0
Maximum: 511
Unit: Pixel
Description: Last pixel.

10.3.2.2.2 Get_ROI (ROI)

Get the region of interest for processing video signal.

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SA_ROIStart SA_ROIStart
Direction: Up
Valid values:
Minimum: 0
Maximum: 511
Unit: Pixel
Description: First pixel.

Parameter: int32_t SA_ROIEnd SA_ROIEnd
Direction: Up
Valid values:
Minimum: 0
Maximum: 511
Unit: Pixel
Description: Last pixel.

10.3.2.3 Measurement value processing

10.3.2.3.1 Set_Averaging (AVERAGE)

Set data averaging at sensor.

Valid for sensor:

- ILD1420
- ILD1750

Parameter: int32_t SP_AveragingType SP_AveragingType
Direction: Down
Valid values:
 0= None
 1= Moving average (MOVING)
 2= Recursive averaging (RECURSIVE)
 3= Median
Description: Averaging type.

Parameter: int32_t SP_MovingCount SP_MovingCount

Direction: Down

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128

Description: Number of value for the averaging window. This parameter is only used at moving average.

Parameter: int32_t SP_RecursiveCount SP_RecursiveCount

Direction: Down

Valid values:

- Minimum:** 2
- Maximum:** 32768

Description: Number of values for recursive averaging. This parameter is only used at recursive average.

Parameter: int32_t SP_MedianCount SP_MedianCount

Direction: Down

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only used at median.

10.3.2.3.2 Get_Averaging (AVERAGE)

Get data averaging at sensor.

Valid for sensor:

ILD1420

ILD1750

Parameter: int32_t SA_AveragingType SA_AveragingType

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = None
- 1 = Moving average (MOVING)
- 2 = Recursive averaging (RECURSIVE)
- 3 = Median

Description: Averaging type.

Parameter: int32_t SA_MovingCount SA_MovingCount

Direction: Up

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128

Description: Number of value for the averaging window.

This parameter is only available at moving average.

Parameter: int32_t SA_RecursiveCount SA_RecursiveCount

Direction: Up

Valid values:

- Minimum: 2
- Maximum: 32768

Description: Number of values for recursive averaging.

This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount SA_MedianCount

Direction: Up

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median.

This parameter is only available at median.

10.3.2.3.3 Set_MasterValue (MASTERMV)

Set the master value.

Valid for sensor:

ILD1220 (from firmware version 001.064)

ILD1320

ILD1420

Parameter: int32_t SP_Master SP_Master

Direction: Down

Valid values:

- 0= no (NONE)
- 1= yes (MASTER)

Description: Specifies if mastering should be done or resetted.

Parameter: double SP_MasterValue SP_MasterValue

Direction: Down

Valid values:

- Minimum: 0
- Maximum: 2 * measuring range

Unit: mm

Description: Master value

10.3.2.3.4 Get_MasterValue (MASTERMV)

Get the master value.

Valid for sensor:

ILD1220 (from firmware version 001.064)
 ILD1320
 ILD1420

Parameter: int32_t SA_Master

SA_Master

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor
 0 = no (NONE)
 1 = yes (MASTER)

Description: Specifies if mastering is active.

Parameter: double SA_MasterValue

SA_MasterValue

Direction: Up

Valid values:

Minimum: 0
Maximum: 2 * measuring range

Unit: mm

Description: Master value.

10.3.2.3.5 Set_MasterSignal (MASTERSIGNAL)

Set the measured value which is used for mastering.

Valid for sensor:

ILD1750
 ILD1900

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

0 = Distance 1

Description: Signal which is used for mastering.

Parameter: double SP_MasterValue

SP_MasterValue

Direction: Down

Valid values:

Minimum: -2* measuring range
Maximum: +2* measuring range

Unit: mm

Description: Master value

10.3.2.3.6 Get_MasterSignal (MASTERSIGNAL)

Get the measured value which is used for mastering.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

0= Distance 1

Description: Signal which is used for mastering.

Parameter: String SA_MasterSignalTable

SA_MasterSignalTable

Direction: Up

Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_MasterSignalEntries

SA_MasterSignalEntries

Direction: Up

Valid values:

Minimum: 0

Maximum: 10

Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MasterSignalEntry1, SA_MasterSignalEntry2, ...

Parameter: String SA_MasterSignalEntry_{1..x}

SA_MasterSignalEntry_{1..x}

Direction: Up

Description: Each master signal as complete line.

Parameter: int32_t SA_MasterSignalIndex_{1..x}

SA_MasterSignalIndex_{1..x}

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Distance 1

Description: Index of each master signal.

Parameter: double SA_MasterValue_{1..x}

SA_MasterValue_{1..x}

Direction: Up

Valid values:

Minimum: -2* measuring range

Maximum: +2* measuring range

Unit: mm

Description: Master value

10.3.2.3.7 Reset_MasterSignal (MASTERSIGNAL)

Reset the measured value which is used for mastering.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

0= Distance 1

Description: Signal which should be reset.

10.3.2.3.8 Set_MasterSource (MASTERSOURCE)

Set the port used for commanding the mastering function

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_MasterSource

SP_MasterSource

Direction: Down

Valid values:

0= None
1= Multi function input (MFI)
2= Select key (KEY_SELECT)

Description: Port which is used for mastering.

10.3.2.3.9 Get_MasterSource (MASTERSOURCE)

Get the port used for commanding the mastering function

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_MasterSource

SA_MasterSource

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= None
1= Multi function input (MFI)
2= Select key (KEY_SELECT)

Description: Port which is used for mastering.

10.3.2.3.10 Exec_Master (MASTER)

Executes mastering for specified signal.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

0= Distance 1
1= All

Description: Signal which should be mastered.

Parameter: int32_t SP_MasterAction

SP_MasterAction

Direction: Down

Valid values:

0= Reset
1= Set

Description: Mastering action.

10.3.2.3.11 Get_Master (MASTER)

Get mastering for specified signal.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_Master

SP_Master

Direction: Down

Valid values:

0= Distance 1
1= All

Description: Signal which should be retrieved.

Parameter: String SA_MasterTable

SA_MasterTable

Direction: Up

Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_MasterEntries

SA_MasterEntries

Direction: Up

Valid values:

Minimum: 0
Maximum: 10

Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MasterEntry1, SA_MasterEntry2, ...

Parameter: String SA_MasterEntry_{1..x}

SA_MasterEntry_{1..x}

Direction: Up

Description: Each master as complete line.

Parameter: int32_t SA_MasterIndex_{1..x}

SA_MasterIndex_{1..x}

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= Distance 1
1= All

Description: Index of each master.

Parameter: int32_t SA_MasterAction_{1..x}

SA_MasterAction_{1..x}

Direction: Up

Valid values:

0= Inactive
1= Active

Description: Mastering action.

10.3.2.3.12 Set_Computation (COMP)

Create/modify customised measurement data processing computations.

Valid for sensor:

ILD1900

Parameter: int32_t SP_ComputationQueue

SP_ComputationQueue

Direction: Down

Valid values:

0= Channel 1

Description: Computation queue

Parameter: int32_t SP_ComputationID	SP_ComputationID
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 2	
Description: Index within computation queue	
Parameter: int32_t SP_ComputationOperation	SP_ComputationOperation
Direction: Down	
Valid values:	
0= None	
1= Median	
2= Moving average	
3= Recursive averaging	
Description: Computation operation	
Parameter: int32_t SP_ComputationSignalIndex	SP_ComputationSignalIndex
Direction: Down	
Valid values:	
-1= Use SP_ComputationSignalName	
0= Distance 1 of first Channel	
Description: Signal index.	
Parameter: String SP_ComputationSignalName	SP_ComputationSignalName
Direction: Down	
Description: User defined signal.	
Parameter: int32_t SP_ComputationMovingCount	SP_ComputationMovingCount
Direction: Down	
Valid values:	
2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096	
Description: Number of value for the averaging window. This parameter is only used at moving average.	
Parameter: int32_t SP_ComputationRecursiveCount	SP_ComputationRecursive-Count
Direction: Down	
Valid values:	
Minimum: 2	
Maximum: 32000	
Description: Number of values for recursive averaging. This parameter is only used at recursive average.	
Parameter: int32_t SP_ComputationMedianCount	SP_ComputationMedianCount
Direction: Down	
Valid values:	
3, 5, 7, 9	
Description: Number of values to build median. This parameter is only used at median.	

10.3.2.3.13 Get_Computation (COMP)

Get customised measurement data processing computations.

Valid for sensor:

ILD1900

Parameter: int32_t SP_ComputationQueue

SP_ComputationQueue

Direction: Down

Valid values:

-1= Any queue
0= Channel 1

Default: -1

Description: Computation queue

Parameter: int32_t SP_ComputationID

SP_ComputationID

Direction: Down

Valid values:

Minimum: 1
Maximum: 2

Description: Index within computation queue. Ignored if SP_Computation-Queue is -1.

Parameter: String SA_ComputationTable

SA_ComputationTable

Direction: Up

Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_ComputationEntries

SA_ComputationEntries

Direction: Up

Valid values:

Minimum: 0
Maximum: 2

Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_ComputationEntry1, SA_ComputationEntry2, ...

Parameter: String SA_ComputationEntry_{1..x}

SA_ComputationEntry_{1..x}

Direction: Up

Description: Each computation as complete line.

Parameter: int32_t SA_ComputationQueue_{1..x}

SA_ComputationQueue_{1..x}

Direction: Up

Valid values:

0= Channel 1

Description: Computation queue

Parameter: int32_t SA_ComputationID_{1..x}

SA_ComputationID_{1..x}

Direction: Up

Valid values:

Minimum: 1
Maximum: 2

Description: Index within computation queue

Parameter: int32_t SA_ComputationOperation1..x	SA_ComputationOperation1..x
Direction: Up	
Valid values:	
0= None	
1= Median	
2= Moving average	
3= Recursive averaging	
Description: Computation operation	
Parameter: int32_t SA_ComputationSignalIndex1..x	SA_ComputationSignalIndex1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_ComputationSignalName1..x	
0= Distance 1 of first Channel	
Description: Signal index.	
Parameter: String SA_ComputationSignalName1..x	SA_ComputationSignalName1..x
Direction: Up	
Description: User defined signal.	
Parameter: int32_t SA_ComputationMovingCount1..x	SA_ComputationMovingCount1..x
Direction: Up	
Valid values:	
2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096	
Description: Number of value for the averaging window. This parameter is only used at moving average.	
Parameter: int32_t SA_ComputationRecursiveCount1..x	SA_ComputationRecursiveCount1..x
Direction: Up	
Valid values:	
Minimum: 2	
Maximum: 32000	
Description: Number of values for recursive averaging. This parameter is only used at recursive average.	
Parameter: int32_t SA_ComputationMedianCount1..x	SA_ComputationMedianCount1..x
Direction: Up	
Valid values:	
3, 5, 7, 9	
Description: Number of values to build median. This parameter is only used at median.	

10.3.3 Data output

10.3.3.1 General

10.3.3.1.1 Set_DataOutInterface (OUTPUT)

Set the active interface for data output.

At this command the sensor may change output data automatically. If first bit of [IP_AutomaticMode](#) is set (1), [Get_OutputInfo_RS422](#) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DataOutInterface	SP_DataOutInterface
Direction: Down	
Valid values:	
0= None (at ILD1220 from firmware version 001.062)	
1= RS422 (at ILD1220 from firmware version 001.062)	
6= Analog	
Valid for sensor:	
ILD1220	
ILD1320	
ILD1420	
Description:	Active interface for data output.
Parameter: int32_t SP_OutputRS422	SP_OutputRS422
Direction: Down	
Valid values:	
0= No	
1= Yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description:	Output active at RS422 interface.
Parameter: int32_t SP_OutputAnalog	SP_OutputAnalog
Direction: Down	
Valid values:	
0= No	
1= Yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description:	Output active at analog interface.
Parameter: int32_t SP_OutputErrorOut1	SP_OutputErrorOut1
Direction: Down	
Valid values:	
0= No	
1= Yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description:	Output active at error output 1.
Parameter: int32_t SP_OutputErrorOut2	SP_OutputErrorOut2
Direction: Down	
Valid values:	
0= No	
1= Yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description:	Output active at error output 2.

10.3.3.1.2 Get_DataOutInterface (OUTPUT)

Get the active interface for data output.

Parameter: int32_t SA_DataOutInterface

SA_DataOutInterface

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None (at ILD1220 from firmware version 001.062)
- 1= RS422 (at ILD1220 from firmware version 001.062)
- 6= Analog

Valid for sensor:

- ILD1220
- ILD1320
- ILD1420

Description: Active interface for data output.

Parameter: int32_t SA_OutputRS422

SA_OutputRS422

Direction: Up

Valid values:

- 0= No
- 1= Yes

Valid for sensor:

- ILD1750
- ILD1900

Description: Output active at RS422 interface.

Parameter: int32_t SA_OutputAnalog

SA_OutputAnalog

Direction: Up

Valid values:

- 0= No
- 1= Yes

Valid for sensor:

- ILD1750
- ILD1900

Description: Output active at analog interface.

Parameter: int32_t SA_OutputErrorOut1

SA_OutputErrorOut1

Direction: Up

Valid values:

- 0= No
- 1= Yes

Valid for sensor:

- ILD1750
- ILD1900

Description: Output active at error output 1.

Parameter: int32_t SA_OutputErrorOut2

SA_OutputErrorOut2

Direction: Up

Valid values:

- 0= No
- 1= Yes

Valid for sensor:

- ILD1750
- ILD1900

Description: Output active at error output 1.

10.3.3.1.3 Set_ResamplingDevice (OUTREDUCEDEVICE)

Set the devices for which resampling is active.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SP_ResampleRS422

SP_ResampleRS422

Direction: Down

Valid values:

- 0= no
- 1= yes

Description: Specify if RS422 output should be resampled.

Parameter: int32_t SP_ResampleAnalog

SP_ResampleAnalog

Direction: Down

Valid values:

- 0= no
- 1= yes

Description: Specify if analog output should be resampled.

Parameter: int32_t SP_ResampleChart

SP_ResampleChart

Direction: Down

Valid values:

- 0= no
- 1= yes

Valid for sensor:

- ILD1750
- ILD1900

Description: Specify if web output should be resampled.

10.3.3.1.4 Get_ResamplingDevice (OUTREDUCEDEVICE)

Get the devices for which resampling is active.

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SA_ResampleRS422

SA_ResampleRS422

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: RS422 output is resampled.

Parameter: int32_t SA_ResampleAnalog

SA_ResampleAnalog

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: Analog output is resampled.

Parameter: int32_t SA_ResampleChart SA_ResampleChart
Direction: Up
Valid values:
 0= no
 1= yes
Valid for sensor:
 ILD1750
 ILD1900
Description: Web output is resampled.

10.3.3.1.5 Set_ResamplingCount (OUTREDUCECOUNT)

Set reduce count for resampling.

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SP_Resampling SP_Resampling
Direction: Down
Valid values:
Minimum: 1
Maximum: 3000000
Description: Resampling value.

10.3.3.1.6 Get_ResamplingCount (OUTREDUCECOUNT)

Get reduce count for resampling.

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SA_Resampling SA_Resampling
Direction: Up
Valid values:
Minimum: 1
Maximum: 3000000
Description: Resampling value.

10.3.3.1.7 Set_HoldLastValid (OUTHOLD)

Set the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SP_HoldLastValid SP_HoldLastValid
Direction: Down
Valid values:
Minimum: -1
Maximum: 1024
Description: Values to replace by last valid value. -1 means no value to hold,
 0 means never output an error value (always hold last valid value).

10.3.3.1.8 Get_HoldLastValid (OUTHOLD)

Get the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SA_HoldLastValid

SA_HoldLastValid

Direction: Up

Valid values:

Minimum: -1

Maximum: 1024

Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).

10.3.3.2 Selected measurement values

10.3.3.2.1 Set_Output_RS422 (OUT_RS422)

Set the data to be output at RS422 interface. For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Valid for sensor:

ILD1220 (from firmware version 001.062)

ILD1320

ILD1420

ILD1750

ILD1900

Parameter: int32_t SP_OutputDistance1_RS422

SP_OutputDistance1_RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if distance is transmitted.

Parameter: int32_t SP_OutputAdditionalShutterTime_RS422

SP_OutputAdditionalShutterTime_RS422

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

ILD1320

ILD1420

ILD1750

ILD1900

Description: Specify if shutter time is transmitted.

Parameter: int32_t SP_OutputAdditionalCounter_RS422

SP_OutputAdditionalCounter_RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if counter is transmitted.

Parameter: int32_t SP_OutputAdditionalMeasrate_RS422	SP_OutputAdditionalMeasrate_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Specify if measrate is transmitted.	
Parameter: int32_t SP_OutputAdditionalTimestampLo_RS422	SP_OutputAdditionalTimestampLo_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Specify if timestamp (lower 16 bit) is transmitted.	
Parameter: int32_t SP_OutputAdditionalTimestampHi_RS422	SP_OutputAdditionalTimestampHi_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Specify if timestamp (upper 16 bit) is transmitted.	
Parameter: int32_t SP_OutputAdditionalTimestamp_RS422	SP_OutputAdditionalTimestamp_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SP_OutputAdditionalIntensity_RS422	SP_OutputAdditionalIntensity_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description: Specify if intensity is transmitted.	

Parameter: int32_t SP_OutputAdditionalState_RS422	SP_OutputAdditionalState_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description: Specify if state is transmitted.	
Parameter: int32_t SP_OutputAdditionalDistanceRaw_RS422	SP_OutputAdditionalDistanceRaw_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if uncalibrated distance is transmitted.	
Parameter: int32_t SP_OutputDistanceUnlinearized_RS422	SP_OutputDistanceUnlinearized_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Specify if unlinearized distance is transmitted.	
Parameter: int32_t SP_OutputVideoRaw_RS422	SP_OutputVideoRaw_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1420	
ILD1750	
ILD1900	
Description: Specify if raw video signal is transmitted.	

10.3.3.2.2 Get_Output_RS422 (OUT_RS422)

Get the data which is output at RS422 interface.

Valid for sensor:

- ILD1220 (from firmware version 001.062)
- ILD1320
- ILD1420
- ILD1750
- ILD1900

Parameter: int32_t SA_OutputDistance1_RS422	SA_OutputDistance1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_RS422	SA_OutputAdditionalShutterTime_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalMeasrate_RS422	SA_OutputAdditionalMeasrate_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Specify if measrate is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestampLo_RS422	SA_OutputAdditionalTimestampLo_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Specify if timestamp (lower 16 bit) is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestampHi_RS422	SA_OutputAdditionalTimestampHi_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Specify if timestamp (upper 16 bit) is transmitted.	

Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_RS422	SA_OutputAdditionalIntensity_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description: Specify if intensity is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
ILD1750	
ILD1900	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalDistanceRaw_RS422	SA_OutputAdditionalDistanceRaw_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if uncalibrated distance is transmitted.	
Parameter: int32_t SA_OutputDistanceUnlinearized_RS422	SA_OutputDistanceUnlinearized_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1750	
ILD1900	
Description: Specify if unlinearized distance is transmitted.	

Parameter: int32_t SA_OutputVideoRaw_RS422 SA_OutputVideoRaw_RS422

Direction: Up

Valid values:

- 0= no
- 1= yes

Valid for sensor:

- ILD1420
- ILD1750
- ILD1900

Description: Specify if raw video signal is transmitted.

10.3.3.2.3 Set_OutputAdditional_RS422 (OUTADD_RS422)

Set the additional data to be output at RS422 interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Valid for sensor:

- ILD1220 (from firmware version 001.062)
- ILD1320
- ILD1420

Parameter: int32_t SP_OutputAdditionalShutterTime_RS422 SP_OutputAdditionalShutterTime_RS422

Direction: Down

Valid values:

- 0= no
- 1= yes

Valid for sensor:

- ILD1320
- ILD1420

Description: Specify if shutter time is transmitted.

Parameter: int32_t SP_OutputAdditionalCounter_RS422

SP_OutputAdditionalCounter_RS422

Direction: Down

Valid values:

- 0= no
- 1= yes

Description: Specify if counter is transmitted.

Parameter: int32_t SP_OutputAdditionalTimestamp_RS422

SP_OutputAdditionalTimestamp_RS422

Direction: Down

Valid values:

- 0= no
- 1= yes

Valid for sensor:

- ILD1320
- ILD1420

Description: Specify if timestamp is transmitted.

Parameter: int32_t SP_OutputAdditionalIntensity_RS422

SP_OutputAdditionalIntensity_RS422

Direction: Down

Valid values:

- 0= no
- 1= yes

Valid for sensor:

- ILD1320
- ILD1420

Description: Specify if intensity is transmitted.

Parameter: int32_t SP_OutputAdditionalState_RS422	SP_OutputAdditionalState_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if state is transmitted.	
Parameter: int32_t SP_OutputAdditionalDistanceRaw_RS422	SP_OutputAdditionalDistanceRaw_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if uncalibrated distance is transmitted.	

10.3.3.2.4 Get_OutputAdditional_RS422 (OUTADD_RS422)

Get the additional data which is output at RS422 interface.

Valid for sensor:

- ILD1220 (from firmware version 001.062)
- ILD1320
- ILD1420

Parameter: int32_t SA_OutputAdditionalShutterTime_RS422	SA_OutputAdditionalShutterTime_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if timestamp is transmitted.	

Parameter: int32_t SA_OutputAdditionalIntensity_RS422	SA_OutputAdditionalIntensity_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if intensity is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalDistanceRaw_RS422	SA_OutputAdditionalDistanceRaw_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
ILD1320	
ILD1420	
Description: Specify if uncalibrated distance is transmitted.	

10.3.3.2.5 Set_OutputVideo_RS422 (OUTVIDEO_RS422)

Set the video signal to be output at RS422 interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Valid for sensor:

ILD1420

Parameter: int32_t SP_OutputVideoRaw_RS422

SP_OutputVideoRaw_RS422

Direction: Down

Valid values:

 0= no

 1= yes

Description: Specify if raw video signal is transmitted.

10.3.3.2.6 Get_OutputVideo_RS422 (OUTVIDEO_RS422)

Get the video signal which is output at RS422 interface.

Valid for sensor:

ILD1420

Parameter: int32_t SA_OutputVideoRaw_RS422

SA_OutputVideoRaw_RS422

Direction: Up

Valid values:

 0= no

 1= yes

Description: Specify if raw video signal is transmitted.

10.3.3.3 Switching outputs

10.3.3.3.1 Set_ErrorOutput1 (ERROROUT1)

Set condition to be used to set error output 1.

Parameter: int32_t SP_ErrorOutput1

SP_ErrorOutput1

Direction: Down

Valid values:

- 0= Output trigger disabled (NONE) (only at ILD1220, ILD1320 and ILD1420)
- 1= No valid distance (no peak found, out of range) (DIST)
- 2= Distance is out of scaled analog range (TEACH)
- 3= Distance is above set threshold (LI1)

Description: Condition for error output.

10.3.3.3.2 Get_ErrorOutput1 (ERROROUT1)

Get condition to be used to set error output 1.

Parameter: int32_t SA_ErrorOutput1

SA_ErrorOutput1

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Output trigger disabled (NONE) (only at ILD1220, ILD1320 and ILD1420)
- 1= No valid distance (no peak found, out of range) (DIST)
- 2= Distance is out of scaled analog range (TEACH)
- 3= Distance is above set threshold (LI1)

Description: Condition for error output.

10.3.3.3.3 Set_ErrorOutput2 (ERROROUT2)

Set condition to be used to set error output 2.

Valid for sensor:

ILD1750

ILD1900

Parameter: int32_t SP_ErrorOutput2

SP_ErrorOutput2

Direction: Down

Valid values:

- 1= No valid distance (no peak found, out of range) (DIST)
- 2= Distance is out of scaled analog range (TEACH)
- 3= Distance is above set threshold (LI1)

Description: Condition for second error output.

10.3.3.3.4 Get_ErrorOutput2 (ERRORROUT2)

Get condition to be used to set error output 2.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_ErrorOutput2

SA_ErrorOutput2

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 1 = No valid distance (no peak found, out of range) (DIST)
- 2 = Distance is out of scaled analog range (TEACH)
- 3 = Distance is above set threshold (L1)

Description: Condition for second error output.

10.3.3.5 Set_ErrorLimit (ERRORLIMIT)

Set the error limits.

Valid for sensor:

ILD1220
ILD1320
ILD1420

Parameter: int32_t SP_DataSource

SP_DataSource

Direction: Down

Valid values:

- 0 = Distance 1

Description: Data source to be checked.

Parameter: double SP_UpperLimit

SP_UpperLimit

Direction: Down

Valid values:

- Minimum:** -2* measuring range
- Maximum:** +2* measuring range

Unit: mm

Description: Upper limit.

10.3.3.6 Get_ErrorLimit (ERRORLIMIT)

Get the error limits.

Valid for sensor:

ILD1220
ILD1320
ILD1420

Parameter: int32_t SA_DataSource

SA_DataSource

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Distance 1

Description: Data source to be checked.

Parameter: double SA_UpperLimit SA_UpperLimit
Direction: Up
Valid values:
 Minimum: -2* measuring range
 Maximum: +2* measuring range
Unit: mm
Description: Upper limit.

10.3.3.3.7 Set_ErrorLimitValues< n > (ERRORLIMITVALUES< n >)

Set the error limits $n \in \{1..2\}$.

Valid for sensor:

ILD1750
ILD1900

Parameter: double SP_LowerLimit< n > SP_LowerLimit< n >
Direction: Down
Valid values:
 Minimum: -2* measuring range
 Maximum: +2* measuring range
Unit: mm
Description: Lower limit.

Parameter: double SP_UpperLimit< n > SP_UpperLimit< n >
Direction: Down
Valid values:
 Minimum: -2* measuring range
 Maximum: +2* measuring range
Unit: mm
Description: Upper limit.

10.3.3.3.8 Get_ErrorLimitValues< n > (ERRORLIMITVALUES< n >)

Get the error limits $n \in \{1..2\}$.

Valid for sensor:

ILD1750
ILD1900

Parameter: double SA_LowerLimit< n > SA_LowerLimit< n >
Direction: Up
Valid values:
 Minimum: -2* measuring range
 Maximum: +2* measuring range
Unit: mm
Description: Lower limit.

Parameter: double SA_UpperLimit< n > SA_UpperLimit< n >
Direction: Up
Valid values:
 Minimum: -2* measuring range
 Maximum: +2* measuring range
Unit: mm
Description: Upper limit.

10.3.3.3.9 Set_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)

Set the compare operation for limit n $\in \{1..2\}$.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_ErrorLimitCompOp<n>

SP_ErrorLimitCompOp<n>

Direction: Down

Valid values:

0= Lower
1= Upper
2= Both

Description: Compare operation

10.3.3.3.10 Get_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)

Get the compare operation for limit n $\in \{1..2\}$.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_ErrorLimitCompOp<n>

SA_ErrorLimitCompOp<n>

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= Lower
1= Upper
2= Both

Description: Compare operation

10.3.3.3.11 Set_ErrorHysteresis (ERRORHYSTERESIS)

Set the hysteresis of the threshold function (Set_ErrorLimit).

Parameter: double SP_ErrorHysteresis

SP_ErrorHysteresis

Direction: Down

Valid values:

Minimum: 0
Maximum: 2* measuring range

Unit: mm

Description: Error hysteresis.

10.3.3.3.12 Get_ErrorHysteresis (ERRORHYSTERESIS)

Get the hysteresis of the threshold function (Set_ErrorLimit).

Parameter: double SA_ErrorHysteresis

SA_ErrorHysteresis

Direction: Up

Valid values:

Minimum: 0
Maximum: 2* measuring range

Unit: mm

Description: Error hysteresis.

10.3.3.3.13 Set_ErrorOutHoldTime (ERROROUTHOLD)

Set the minimum hold period of the threshold function (Set_ErrorLimit).

Parameter: int32_t SP_ErrorOutHoldTime SP_ErrorOutHoldTime
Direction: Down
Valid values:
 Minimum: 0
 Maximum: 1000
Unit: ms
Description: Minimum hold period.

10.3.3.3.14 Get_ErrorOutHoldTime (ERROROUTHOLD)

Get the minimum hold period of the threshold function (Set_ErrorLimit).

Parameter: int32_t SA_ErrorOutHoldTime SA_ErrorOutHoldTime
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 1000
Unit: ms
Description: Minimum hold period.

10.3.3.3.15 Set_ErrorLevelOut1 (ERRORLEVELOUT1)

Set level of error output 1 on error.

Parameter: int32_t SP_ErrorLevelOut1 SP_ErrorLevelOut1
Direction: Down
Valid values:
 0= NPN
 1= PNP
 2= Push-Pull (PUSHPULL)
 3= Push-Pull negated (PUSHPULLNEG)
Description: Error level for out 1.

10.3.3.3.16 Get_ErrorLevelOut1 (ERRORLEVELOUT1)

Get level of error output 1 on error.

Parameter: int32_t SA_ErrorLevelOut1 SA_ErrorLevelOut1
Direction: Up
Valid values:
 0= NPN
 1= PNP
 2= Push-Pull (PUSHPULL)
 3= Push-Pull negated (PUSHPULLNEG)
Description: Error level for out 1.

10.3.3.3.17 Set_ErrorLevelOut2 (ERRORLEVELOUT2)

Set level of error output 2 on error.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_ErrorLevelOut2

SP_ErrorLevelOut2

Direction: Down

Valid values:

0= NPN
1= PNP
2= Push-Pull (PUSHPULL)
3= Push-Pull negated (PUSHPULLNEG)

Description: Error level for out 2.

10.3.3.3.18 Get_ErrorLevelOut2 (ERRORLEVELOUT2)

Get level of error output 2 on error.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_ErrorLevelOut2

SA_ErrorLevelOut2

Direction: Up

Valid values:

0= NPN
1= PNP
2= Push-Pull (PUSHPULL)
3= Push-Pull negated (PUSHPULLNEG)

Description: Error level for out 2.

10.3.3.4 Analog output

10.3.3.4.1 Set_AnalogRange (ANALOG RANGE)

Set the analog output range.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_AnalogRange

SP_AnalogRange

Direction: Down

Valid values:

1= 0 - 5V
2= 0 - 10V
5= 4 - 20mA

Description: Analog output range.

10.3.3.4.2 Get_AnalogRange (ANALOGRANGE)

Get the analog output range.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_AnalogRange

SA_AnalogRange

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor
1 = 0 - 5V
2 = 0 - 10V
5 = 4 - 20mA

Description: Analog output range.

10.3.3.4.3 Set_AnalogScale (ANALOGSCALE)

Set the scaling factor for analog output.

Valid for sensor:

ILD1220
ILD1320
ILD1420

Parameter: int32_t SP_AnalogScaleMode

SP_AnalogScaleMode

Direction: Down

Valid values:

0 = Standard
1 = Two point (TWOPOINT)

Description: Analog scale mode.

Parameter: double SP_MinValue

SP_MinValue

Direction: Down

Valid values:

Minimum: 0
Maximum: 2* measuring range

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SP_MaxValue

SP_MaxValue

Direction: Down

Valid values:

Minimum: 0
Maximum: 2* measuring range

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

10.3.3.4.4 Get_AnalogScale (ANALOGSCALE)

Get the scaling factor for analog output.

Valid for sensor:

- ILD1220
- ILD1320
- ILD1420

Parameter: int32_t SA_AnalogScaleMode

SA_AnalogScaleMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Standard
- 1 = Two point (TWOPOINT)

Description: Analog scale mode.

Parameter: double SA_MinValue

SA_MinValue

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 2* measuring range

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SA_MaxValue

SA_MaxValue

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 2* measuring range

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

10.3.3.4.5 Set_AnalogScaleMode (ANALOGSCALEMODE)

Set the scaling mode for analog output.

Valid for sensor:

- ILD1750
- ILD1900

Parameter: int32_t SP_AnalogScaleMode

SP_AnalogScaleMode

Direction: Down

Valid values:

- 0 = Standard
- 1 = Two point (TWOPOINT)

Description: Analog scale mode.

10.3.3.4.6 Get_AnalogScaleMode (ANALOGSCALEMODE)

Get the scaling mode for analog output.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_AnalogScaleMode

SA_AnalogScaleMode

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor
0 = Standard
1 = Two point (TWOPOINT)

Description: Analog scale mode.

10.3.3.4.7 Set_AnalogScaleRange (ANALOGSCALERANGE)

Set the scaling factors for analog output.

Valid for sensor:

ILD1750
ILD1900

Parameter: double SP_AnalogScaleRangeLowerLimit

SP_AnalogScaleRangeLower-
Limit

Direction: Down

Valid values:

Minimum: -2* measuring range
Maximum: +2* measuring range

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SP_AnalogScaleRangeUpperLimit

SP_AnalogScaleRangeUpper-
Limit

Direction: Down

Valid values:

Minimum: -2* measuring range
Maximum: +2* measuring range

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

10.3.3.4.8 Get_AnalogScaleRange (ANALOGSCALERANGE)

Get the scaling factor for analog output.

Valid for sensor:

ILD1750
ILD1900

Parameter: double SA_AnalogScaleRangeLowerLimit

SA_AnalogScaleRangeLower-
Limit

Direction: Up

Valid values:

Minimum: -2* measuring range
Maximum: +2* measuring range

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

10.4. Commands for ILD17xx

Parameter: double SA_AnalogScaleRangeUpperLimit
Direction: Up
Valid values:
 Minimum: -2* measuring range
 Maximum: +2* measuring range
Unit: mm
Description: Value which represents highest voltage/current (at two point scaling).

10.3.3.4.9 Set_AnalogScaleSource (ANALOGSCALESOURCE)

Set the port used for commanding the teach function.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SP_AnalogScaleSource SP_AnalogScaleSource
Direction: Down
Valid values:
0= None
1= Multi function input (MFI)
2= Select key (KEY_SELECT)
Description: Port which is used for analog scaling.

10.3.3.4.10 Get_AnalogScaleSource (ANALOGSCALESOURCE)

Get the port used for commanding the teach function.

Valid for sensor:

ILD1750
ILD1900

Parameter: int32_t SA_AnalogScaleSource SA_AnalogScaleSource
Direction: Up
Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Multi function input (MFI)
- 2= Select key (KEY_SELECT)

Description: Port which is used analog scaling.

10.4 Commands for ILD17xx

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- **RS232** (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
 - **IF2004** (native).

- [TCP/IP](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Settings](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate and to scale data.

If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls sensor command [Dat_Out_On](#), [Laser_On](#) and optionally [Set_OutputType](#) at [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor, from 0 to 16383.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_Settings](#)), error values are scaled depending of [IP_ScaleErrorValues](#).

10.4.1 General commands

10.4.1.1 General

10.4.1.1.1 Get_Info (GET_INFO)

Retrieve some information about the sensor.

Parameter: String SA_Sensor

SA_Sensor

Direction: Up

Description: Name of the sensor.

Parameter: String SA_SensorType

SA_SensorType

Direction: Up

Description: Type of the sensor.

Parameter: String SA_Softwareversion

SA_Softwareversion

Direction: Up

Description: Software version of the sensor.

Parameter: int32_t SA_OutputType

SA_OutputType

Direction: Up

Valid values:

0= current (4..20mA)

1= voltage (0..10V)

2= RS422

Description: Data output (only values, not answer) interface of the sensor.

Parameter: int32_t SA_ErrorOutput SA_ErrorOutput

Direction: Up

Valid values:

- 0= sync error mode
- 1= sync switch mode
- 2= trigger error mode
- 3= trigger switch mode

Description: Sync/trigger and error/switch mode respectively of the sensor.

Parameter: int32_t SA_Speed SA_Speed

Direction: Up

Valid values:

- 0= 2.5kHz
- 1= 1.25kHz
- 2= 625Hz
- 3= 312.5Hz

Description: The output speed of the sensor.

Parameter: double SA_Samplerate SA_Samplerate

Direction: Up

Valid values:

- 2500
- 1250
- 625
- 312.5 (or other on specific sensor settings)

Unit: Hz

Description: The output speed of the sensor.

Parameter: int32_t SA_AvType SA_AvType

Direction: Up

Valid values:

- 0= recursive
- 1= moving
- 2= Median

Description: The averaging type.

Parameter: int32_t SA_AvIndex SA_AvIndex

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 15

Description: The averaging index. Averaging $N = 2^{AvIndex}$.

Parameter: int32_t SA_ErrorHandler SA_ErrorHandler

Direction: Up

Valid values:

- 0= error values
- 1= hold last value

Description: If the sensor cannot measure values, it can output the last valid value or it can output an error values.

Parameter: int32_t SA_Sync_TrgMode SA_Sync_TrgMode

Direction: Up

Valid values:

Condition: at sync mode (ErrorOutput is 0 or 1):

0= synchronous master off

1= synchronous master on

2= slave

3= alternating synchronous master

Valid values:

Condition: at trigger mode (ErrorOutput is 2 or 3):

0= edge L/H

1= edge H/L

2= level H

3= level L

Description: The sync mode or the trigger mode of the sensor.

Parameter: int32_t SA_ASCII SA_ASCII

Direction: Up

Valid values:

0= off (binary 2 bytes/value)

1= on (ASCII 6 bytes/value)

Description: Returns the mode the sensor is sending data (only values).

Parameter: int32_t SA_Keylock SA_Keylock

Direction: Up

Valid values:

0= off (keys enabled)

1= on (keys locked)

Description: The keypad state at the sensor.

Parameter: double SA_Range SA_Range

Direction: Up

Valid values:

Minimum: 0.0

Unit: mm

Description: Range of the sensor.

Parameter: String SA_SerialNumber SA_SerialNumber

Direction: Up

Valid values:

Numeric value

Description: Serial number of the sensor.

Parameter: String SA_Option SA_Option

Direction: Up

Valid values:

Numeric value

Description: Option of the sensor.

Parameter: String SA_ArticleNumber SA_ArticleNumber

Direction: Up

Valid values:

Numeric value

Description: Article number of the sensor.

Parameter: String SA_Date	SA_Date
Direction: Up	
Description: Software release date of the sensor.	
Parameter: String SA_BootLoaderVer	SA_BootLoaderVer
Direction: Up	
Description: Boot loader version of the sensor.	
Parameter: String SA_SWType	SA_SWType
Direction: Up	
Description: Software type of the sensor.	
Parameter: int32_t SA_EnableFlash	SA_EnableFlash
Direction: Up	
Valid values:	
0= locked	
1= enabled	
Description: The flash is locked or enabled for writing.	
10.4.1.1.2 Get_Settings (GET_SETTINGS)	
Retrieve detailed information about the sensor.	
Parameter: int32_t SA_OutputType	SA_OutputType
Direction: Up	
Valid values:	
0= current (4..20mA)	
1= voltage (0..10V)	
2= RS422	
Description: Data output (only values, not answer) interface of the sensor.	
Parameter: int32_t SA_Speed	SA_Speed
Direction: Up	
Valid values:	
0= 2.5kHz	
1= 1.25kHz	
2= 625Hz	
3= 312.5Hz	
Description: The output speed of the sensor.	
Parameter: int32_t SA_AvIndex	SA_AvIndex
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 15	
Description: The averaging index. Averaging $N = 2^{AvIndex}$.	
Parameter: int32_t SA_ErrorHandler	SA_ErrorHandler
Direction: Up	
Valid values:	
0= error values	
1= hold last value	
Description: If the sensor cannot measure values, it can output the last valid value or it can output an error values.	

Parameter: int32_t SA_Sync_TrgMode SA_Sync_TrgMode

Direction: Up

Valid values:

Condition: at sync mode (ErrorOutput is 0 or 1):

0= synchronous master off

1= synchronous master on

2= slave

3= alternating synchronous master

Valid values:

Condition: at trigger mode (ErrorOutput is 2 or 3):

0= edge L/H

1= edge H/L

2= level H

3= level L

Description: The sync mode or the trigger mode of the sensor.

Parameter: int32_t SA_AvType SA_AvType

Direction: Up

Valid values:

0= recursive

1= moving

2= Median

Description: The averaging type.

Parameter: int32_t SA_ErrorOutput SA_ErrorOutput

Direction: Up

Valid values:

0= sync error mode

1= sync switch mode

2= trigger error mode

3= trigger switch mode

Description: Sync/trigger and error/switch mode respectively of the sensor.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate

Direction: Up

Valid values:

0= 115200 Baud

1= 57600 Baud

2= 19200 Baud

3= 9600 Baud

Description: The serial connection baudrate of the sensor.

Parameter: int32_t SA_ASCII SA_ASCII

Direction: Up

Valid values:

0= off (binary 2 bytes/value)

1= on (ASCII 6 bytes/value)

Description: Returns the mode the sensor is sending data (only values).

Parameter: int32_t SA_Upper_limit SA_Upper_limit

Direction: Up

Valid values:

Minimum: 0

Maximum: 16368

Description: The upper limit of the sensor.

Parameter: int32_t SA_Lower_limit	SA_Lower_limit
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 16368	
Description: The lower limit of the sensor.	
Parameter: int32_t SA_Upper_hysteresis	SA_Upper_hysteresis
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 16368	
Description: The upper hysteresis of the sensor.	
Parameter: int32_t SA_Lower_hysteresis	SA_Lower_hysteresis
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 16368	
Description: The lower hysteresis of the sensor.	
Parameter: int32_t SA_Master_value	SA_Master_value
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 16368	
Description: The master value of the sensor.	
Parameter: int32_t SA_Master_MidPoint_Setup	SA_Master_MidPoint_Setup
Direction: Up	
Valid values:	
Condition: switch mode: 0/2= Not mastered 1/3= Mastered	
Valid values:	
Condition: error mode: 0/1= mid-point value not set 2/3= mid-point value set	
Description: The master and midpoint of the sensor.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Unit: mm	
Description: Range of the sensor.	
Parameter: int32_t SA_AssignLimits_ErrorOutput	SA_AssignLimits_ErrorOutput
Direction: Up	
Valid values:	
0= Set_LowerLimit_F1 1= Set_UpperLimit_F1	
Description: The assignment of the error outputs of the sensor.	

Parameter: int32_t SA_Keylock SA_Keylock

Direction: Up

Valid values:

0= off (keys enabled)
1= on (keys locked)

Description: The keypad state at the sensor.

Parameter: int32_t SA_DatOut SA_DatOut

Direction: Up

Valid values:

0= Dat_Out_Off
1= Dat_Out_On

Description: Data output from sensor.

Parameter: int32_t SA_LaserState SA_LaserState

Direction: Up

Valid values:

0= Laser_Off
1= Laser_On

Description: Laser state of sensor.

Parameter: int32_t SA_EnableFlash SA_EnableFlash

Direction: Up

Valid values:

0= locked
1= enabled

Description: The flash is locked or enabled for writing.

10.4.1.1.3 Reset_Boot (RESET_BOOT)

Resets the sensor.

10.4.1.2 User level

10.4.1.2.1 Set_KeyLock (SET_KEYLOCK)

Locks/Unlocks the keypad of sensor.

Parameter: int32_t SP_Keylock SP_Keylock

Direction: Down

Valid values:

0= off (keys enabled)
1= on (keys locked)

Description: The keypad state at the sensor.

10.4.1.3 Interfaces

10.4.1.3.1 Set_Baudrate (SET_BAUDRATE)

Set the baudrate of the serial interface of sensor. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Attention! To avoid synchronization problems, sending data by sensor should be disabled first (sensor command [Dat_Out_Off](#)).

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

0= 115200 Baud

1= 57600 Baud

2= 19200 Baud

3= 9600 Baud

Description: The serial connection baudrate of the sensor.

Parameter: int32_t CP_InterruptDataTransfer

CP_InterruptDataTransfer

Direction: Down

Valid values:

0= false

1= true

Default: 0

Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Dat_Out_Off](#)) and enabling it again ([Dat_Out_On](#)) at end.

10.4.1.4 Parameter management

10.4.1.4.1 Set_Default (SET_DEFAULT)

Resets the sensor to factory settings.

If first bit of [IP_AutomaticMode](#) is set (1), [Get_Settings](#) is called automatically after this command. Otherwise, you have to call it manually.

10.4.1.4.2 WriteFlashZero (WriteFlashZero)

Locks/enables the flash of sensor for writing.

Parameter: int32_t SP_EnableFlash

SP_EnableFlash

Direction: Down

Valid values:

0= locked

1= enabled

Description: The flash is locked or enabled for writing.

10.4.1.5 Internal controller commands

10.4.1.5.1 Laser_Off (LASER_OFF)

Switch the laser off.

10.4.1.5.2 Laser_On (LASER_ON)

Switch the laser on.

10.4.1.5.3 Set_Sync_TrMode (SET_SYNCMODE/TRIGGERMODE)

Set the synchronisation and trigger mode respectively.

Parameter: int32_t SP_Sync_TrMode

SP_Sync_TrMode

Direction: Down

Valid values:

Condition: at sync mode (ErrorOutput is 0 or 1):

- 0= synchronous master off
- 1= synchronous master on
- 2= slave
- 3= alternating synchronous master

Valid values:

Condition: at trigger mode (ErrorOutput is 2 or 3):

- 0= edge L/H
- 1= edge H/L
- 2= level H
- 3= level L

Description: The sync mode or the trigger mode of the sensor.

10.4.1.5.4 Set_ErrorOutput (SET_ERROROUTPUT)

Set the synchronisation or trigger mode and set the error or switch mode.

Parameter: int32_t SP_ErrorOutput

SP_ErrorOutput

Direction: Down

Valid values:

- 0= sync error mode
- 1= sync switch mode
- 2= trigger error mode
- 3= trigger switch mode

Description: Sync/trigger and error/switch mode respectively of the sensor.

10.4.2 Measurement

10.4.2.1 Set_MeasureMode

Set the measure mode of sensor. This command is only available for sensors with option 12

Parameter: int32_t SP_MeasureMode

SP_MeasureMode

Direction: Down

Valid values:

- 0= Diffuse reflexion
- 1= Direct reflexion

Description: The measure mode of the sensor.

10.4.2.2 Set_Speed (SET_SPEED)

Set the data acquisition speed of the sensor.

Parameter: int32_t SP_Speed

SP_Speed

Direction: Down

Valid values:

0= 2.5kHz

1= 1.25kHz

2= 625Hz

3= 312.5Hz

Description: The output speed of the sensor.

10.4.2.3 Set_VideoMode

Enter/Leave the video mode of the sensor.

Parameter: int32_t SP_VideoMode

SP_VideoMode

Direction: Down

Valid values:

0= off

1= on

Description: Switch video mode on or off.

10.4.2.4 Get_Video

Get recent video signal from sensor.

Parameter: Binary data SA_VideoSignal

SA_VideoSignal

Direction: Up

Valid values:

512 bytes, each byte is an intensity value.

Description: Raw video signal

10.4.2.5 Set_Av0 (SET_AV0)

Set averaging index AvIndex= 0, Averaging N= 1.

10.4.2.6 Set_Av1 (SET_AV1)

Set averaging index AvIndex= 2, Averaging N= 4.

10.4.2.7 Set_Av2 (SET_AV2)

Set averaging index AvIndex= 5, Averaging N= 32.

10.4.2.8 Set_Av3 (SET_AV3)

Set averaging index AvIndex= 7, Averaging N= 128.

10.4.2.9 Set_AvX (SET_AVX)

Set averaging index of sensor.

Parameter: int32_t SP_AvIndex SP_AvIndex
Direction: Down
Valid values:
Minimum: 0
Maximum: 15
Description: The averaging index. Averaging $N = 2^{AvIndex}$.

10.4.2.10 Set_Av_T (SET_AV_T)

Set averaging type of sensor.

Parameter: int32_t SP_AvType SP_AvType
Direction: Down
Valid values:
 0= recursive
 1= moving
 2= Median
Description: The averaging type.

10.4.2.11 Get_MeasValue (GET_MEASVALUE)

Is only useful in trigger mode (ErrorOutput is 2 or 3).

Tells the sensor to measure values without hardware trigger condition (software trigger).

Parameter: int32_t SP_MeasValue SP_MeasValue
Direction: Down
Valid values:
Minimum: 1
Maximum: 2147483647 (INT32_MAX)
Description: The number of values to measure.

10.4.3 Data output

10.4.3.1 General

10.4.3.1.1 Dat_Out_Off (DAT_OUT_OFF)

Switch off data output from sensor.

10.4.3.1.2 Dat_Out_On (DAT_OUT_ON)

Switch on data output from sensor.

10.4.3.1.3 Set_ErrorHandler (SET_ERRORHANDLER)

Set the behaviour on invalid values at sensor.

Parameter: int32_t SP_ErrorHandler

SP_ErrorHandler

Direction: Down

Valid values:

0= error values

1= hold last value

Description: If the sensor cannot measure values, it can output the last valid value or it can output an error values.

10.4.3.1.4 ASCII_Output (ASCII_OUTPUT)

Set digital data transfer (only values, no sensor answer) to ASCII or binary.

Parameter: int32_t SP_ASCII

SP_ASCII

Direction: Down

Valid values:

0= off (binary 2 bytes/value)

1= on (ASCII 6 bytes/value)

Description: Returns the mode the sensor is sending data (only values).

10.4.3.1.5 Set_OutputType (SET_OUTPUTTYP)

Set the output type of sensor.

Parameter: int32_t SP_OutputType

SP_OutputType

Direction: Down

Valid values:

0= current (4..20mA)

1= voltage (0..10V)

2= RS422

Description: Data output (only values, not answer) interface of the sensor.

10.4.3.2 Switching outputs

10.4.3.2.1 Set_Limits (SET_LIMITS)

Set sensor limits.

Parameter: int32_t SP_Upper_limit

SP_Upper_limit

Direction: Down

Valid values:

Minimum: 0

Maximum: 16368

Description: Upper limit.

Parameter: int32_t SP_Lower_limit	SP_Lower_limit
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 16368	
Description: Lower limit.	
Parameter: int32_t SP_Upper_hysteresis	SP_Upper_hysteresis
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 16368	
Description: Upper hysteresis.	
Parameter: int32_t SP_Lower_hysteresis	SP_Lower_hysteresis
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 16368	
Description: Lower hysteresis.	
Parameter: int32_t SP_Master_value	SP_Master_value
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 16368	
Description: Master value.	

10.4.3.2.2 Set_UpperLimit_F1 (SET_UPPERLIMIT_F1)

Assign the upper limit of sensor to error output 1 and lower limit to error output 2.

10.4.3.2.3 Set_LowerLimit_F1 (SET_LOWERLIMIT_F1)

Assign the lower limit of sensor to error output 1 and upper limit to error output 2.

10.4.3.3 Analog output

10.4.3.3.1 Zero (SET_ZERO)

Autozero the analog output value.

Parameter: int32_t SP_Zero	SP_Zero
Direction: Down	
Valid values:	
0= reset	
1= set	
Description: Set or reset autozero mode.	

10.5 Commands for ILD22xx

See sensor manual for detailed description of sensor commands. Driver for ILD2200 also includes ILD2220 support.

This sensor supports following interfaces:

- [RS232](#) (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
- [IF2004](#) (native).
- [TCP/IP](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Settings](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate and to scale data.

If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls sensor command [Dat_Out_On](#) and [Laser_On](#) at [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor (including two unused bits in high byte, mask it out by raw&0xffff), from 0 to 65535 (without unused bits).
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_Settings](#)), error values are scaled depending of [IP_ScaleErrorValues](#).

If [Transmit_Intensity](#) is enabled, distance and intensity values are filled in the arrays alternating. Each array always starts with an distance value.

10.5.1 General commands

10.5.1.1 General

10.5.1.1.1 Get_Info (INFO)

Retrieve some information about the sensor.

Parameter: String SA_Sensor

SA_Sensor

Direction: Up

Description: Name of the sensor.

Parameter: String SA_SensorType	SA_SensorType
Direction: Up	
Description: Type of the sensor.	
Parameter: double SA_Samplerate	SA_Samplerate
Direction: Up	
Valid values:	
20000	
10000	
5000	
2500	
Unit: Hz	
Description: The output speed of the sensor.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Unit: mm	
Description: Range of the sensor.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the sensor.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor.	
Parameter: int32_t SA_AvIndex	SA_AvIndex
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 15	
Description: The averaging index. Averaging $N = 2^{AvIndex}$.	
Parameter: int32_t SA_Output_Analog	SA_Output_Analog
Direction: Up	
Valid values:	
0= not avail	
1= voltage (+-5V)	
Description: The analog output type.	
Parameter: int32_t SA_Output_Digital	SA_Output_Digital
Direction: Up	
Valid values:	
0= not avail	
2= RS422	
Description: The digital output type.	

10.5.1.1.2 Get_Settings (Get_Settings)

Retrieve detailed information about the sensor.

Parameter: int32_t SA_Speed SA_Speed

Direction: Up

Valid values:

0= 10kHz

1= 5kHz

2= 2.5kHz

3= 20kHz

Description: The output speed of the sensor.

Parameter: int32_t SA_AvIndex SA_AvIndex

Direction: Up

Valid values:

Minimum: 0

Maximum: 15

Description: The averaging index. Averaging $N = 2^{AvIndex}$.

Parameter: int32_t SA_ErrorHandler SA_ErrorHandler

Direction: Up

Valid values:

0= error values

1= hold last value

Description: If the sensor cannot measure values, it can output the last valid value or it can output an error values.

Parameter: int32_t SA_AvType SA_AvType

Direction: Up

Valid values:

0= recursive

1= moving

2= Median

Description: The averaging type.

Parameter: double SA_OffsetValue SA_OffsetValue

Direction: Up

Valid values:

Minimum: -50.0

Maximum: +50.0

Unit: %

Description: The offset value of the sensor.

Parameter: int32_t SA_ZeroPoint SA_ZeroPoint

Direction: Up

Valid values:

0= absolute

1= relative (zero is set)

Description: Autozero off/on.

Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Unit: mm	
Description: Range of the sensor.	
Parameter: int32_t SA_Keylock	SA_Keylock
Direction: Up	
Valid values:	
0= off (keys enabled)	
1= on (keys locked)	
Description: The keypad state at the sensor.	
Parameter: int32_t SA_DatOut	SA_DatOut
Direction: Up	
Valid values:	
0= Dat_Out_Off	
1= Dat_Out_On	
Description: Data output from sensor.	
Parameter: int32_t SA_LaserState	SA_LaserState
Direction: Up	
Valid values:	
0= Laser_Off	
1= Laser_On	
Description: Laser state of sensor.	

10.5.1.1.3 Get_Version

Retrieve the sensor software version.

Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of the sensor.	

10.5.1.1.4 Reset_Boot (RESET)

Resets the sensor.

10.5.1.2 User level

10.5.1.2.1 Set_KeyLock (SET_TASTENSPERRE)

Locks/Unlocks the keypad of sensor.

Parameter: int32_t SP_Keylock	SP_Keylock
Direction: Down	
Valid values:	
0= off (keys enabled)	
1= on (keys locked)	
Description: The keypad state at the sensor.	

10.5.1.3 Internal controller commands

10.5.1.3.1 Laser_Off (LASER_OFF)

Switch off the laser.

10.5.1.3.2 Laser_On (LASER_ON)

Switch on the laser.

10.5.2 Measurement

10.5.2.1 Set_Av0 (AVG 0)

Set averaging index AvIndex= 0, Averaging N= 1.

10.5.2.2 Set_Av1 (AVG 1)

Set averaging index AvIndex= 2, Averaging N= 4.

10.5.2.3 Set_Av2 (AVG 2)

Set averaging index AvIndex= 5, Averaging N= 32.

10.5.2.4 Set_Av3 (AVG 3)

Set averaging index AvIndex= 7, Averaging N= 128.

10.5.2.5 Set_AvX (AVG n)

Set averaging index of sensor.

Parameter: int32_t SP_AvIndex

SP_AvIndex

Direction: Down

Valid values:

Minimum: 0

Maximum: 15

Description: The averaging index. Averaging $N = 2^{AvIndex}$.

10.5.2.6 Set_Av_T (AVGTYP)

Set averaging type of sensor.

Parameter: int32_t SP_AvType

SP_AvType

Direction: Down

Valid values:

0= recursive

1= moving

2= Median

Description: The averaging type.

10.5.3 Data output

10.5.3.1 General

10.5.3.1.1 Dat_Out_Off (STOP)

Switch off data output from sensor.

10.5.3.1.2 Dat_Out_On (START)

Switch on data output from sensor.

10.5.3.2 Switching outputs

10.5.3.2.1 Transmit_Intensity

Tells the sensor to transmit the intesity value after each distance value.

This command is only available with sensor option 204 and 243.

Option 204: This setting will not be stored persistently in the sensor so it has to be set after each reset or power on.

Option 243: This setting is stored persistently in the sensor. But to tell MEDAQLib about this you have to set it after connecting to sensor.

Parameter: int32_t SP_TransmitIntensity

SP_TransmitIntensity

Direction: Down

Valid values:

0 = no

1 = yes

Description: Transmit intensitiy value.

10.5.3.3 Analog output

10.5.3.3.1 Zero (ZERO)

Autozero the analog output value.

10.6 Commands for ILD2300/10

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
- [IF2004](#) (native).
- [TCP/IP](#) (native).
- [IF2004_USB](#) (native).

- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_AllParameters](#) (SP_Additional= 1) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate, to interpret and scale data and to assign values. If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls optionally sensor command [Set_IPDataTransferMode](#), [Set_DataOutInterface](#), [Get_LaserPower](#) and optionally [Set_LaserPower](#) at [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_AllParameters](#) (SP_Additional= 1)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

10.6.1 General commands

10.6.1.1 General

10.6.1.1.1 Get_Help (HELP)

Retrieve a help text from sensor for a specific command.

Parameter: String SP_Command SP_Command

Direction: Down

Valid values:

"" (empty string, means general help)
or any command name

Description: Name of the command.

Parameter: String SA_HelpText SA_HelpText

Direction: Up

Description: Help text to the command.

10.6.1.1.2 Get_Info (GETINFO)

Retrieve information about the sensor.

Parameter: String SA_Sensor SA_Sensor

Direction: Up

Description: Name of the sensor.

Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the sensor.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the sensor.	
Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description: MAC address (low level ethernet address) of the sensor.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the sensor.	
Parameter: String SA_ImageType	SA_ImageType
Direction: Up	
Description: Firmware image type used by the sensor.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Unit: mm	
Description: Range of the sensor.	
Parameter: String SA_CalibrationTable	SA_CalibrationTable
Direction: Up	
Description: Name of calibration table which is actually used.	

10.6.1.1.3 Get_OutputInfo_RS422 (GETOUTINFO_RS422)

Retrieve information which data is output at RS422 interface.

Parameter: int32_t SA_OutputAdditionalShutterTime_RS422	SA_OutputAdditionalShutterTime_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	

Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalTemperature_RS422	SA_OutputAdditionalTemperature_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity1_RS422	SA_OutputAdditionalIntensity1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity 1 is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity2_RS422	SA_OutputAdditionalIntensity2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity 2 is transmitted.	
Parameter: int32_t SA_OutputDistance1_RS422	SA_OutputDistance1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 is transmitted.	
Parameter: int32_t SA_OutputDistance2_RS422	SA_OutputDistance2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 2 is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	

Parameter: int32_t SA_OutputThickness12_RS422	SA_OutputThickness12_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if thickness between first and second peak is transmitted.	
Parameter: int32_t SA_OutputStatisticMin_RS422	SA_OutputStatisticMin_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if min value is transmitted.	
Parameter: int32_t SA_OutputStatisticMax_RS422	SA_OutputStatisticMax_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if max value is transmitted.	
Parameter: int32_t SA_OutputStatisticPeak2Peak_RS422	SA_OutputStatistic- Peak2Peak_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if peak to peak value is transmitted.	

10.6.1.1.4 Get_OutputInfo_ETH (GETOUTINFO_ETH)

Retrieve information which data is output at ETH interface.

Parameter: int32_t SA_OutputAdditionalShutterTime_ETH	SA_OutputAdditionalShut- terTime_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditional- Counter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimes- tamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	

Parameter: int32_t SA_OutputAdditionalTemperature_ETH	SA_OutputAdditionalTemperature_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity1_ETH	SA_OutputAdditionalIntensity1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity 1 is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity2_ETH	SA_OutputAdditionalIntensity2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity 2 is transmitted.	
Parameter: int32_t SA_OutputDistance1_ETH	SA_OutputDistance1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 is transmitted.	
Parameter: int32_t SA_OutputDistance2_ETH	SA_OutputDistance2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 2 is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_ETH	SA_OutputAdditionalState_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalTrgCounter_ETH	SA_OutputAdditionalTrgCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger counter is transmitted.	
Parameter: int32_t SA_OutputThickness12_ETH	SA_OutputThickness12_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if thickness between first and second peak is transmitted.	

Parameter: int32_t SA_OutputStatisticMin_ETH	SA_OutputStatisticMin_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if min value is transmitted.	
Parameter: int32_t SA_OutputStatisticMax_ETH	SA_OutputStatisticMax_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if max value is transmitted.	
Parameter: int32_t SA_OutputStatisticPeak2Peak_ETH	SA_OutputStatisticPeak2Peak_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if peak to peak value is transmitted.	

10.6.1.1.5 Set_SyncMode (SYNC)

Set the synchronization mode.

Parameter: int32_t SP_SyncMode	SP_SyncMode
Direction: Down	
Valid values:	
0= None	
1= Slave	
2= Master	
3= Master alternating (MASTER_ALT)	
Description: Synchronization mode.	
Parameter: int32_t SP_SyncTermination	SP_SyncTermination
Direction: Down	
Valid values:	
0= Off (TERMOFF)	
1= On 120 Ohm (TERMON)	
Description: Termination of external input.	

10.6.1.1.6 Get_SyncMode (SYNC)

Get the synchronization mode.

Parameter: int32_t SA_SyncMode	SA_SyncMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Slave	
2= Master	
3= Master alternating (MASTER_ALT)	
Description: Synchronization mode.	

Parameter: int32_t SA_SyncTermination SA_SyncTermination
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = Off (TERMOFF)
 1 = On 120 Ohm (TERMON)
Description: Termination of external input.

10.6.1.1.7 Set_Unit (UNIT)

Set the unit for configuration and display in the web diagram.

Parameter: int32_t SP_DisplayUnit SP_DisplayUnit
Direction: Down
Valid values:
 0 = mm
 1 = Inch
Description: Unit.

10.6.1.1.8 Get_Unit (UNIT)

Get the unit for configuration and display in the web diagram.

Parameter: int32_t SA_DisplayUnit SA_DisplayUnit
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = mm
 1 = Inch
Description: Unit.

10.6.1.1.9 Reset_Boot (RESET)

Resets the sensor.

At this command the sensor may change output data after reboot. If first bit of [IP_AutomaticMode](#) is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

10.6.1.1.10 Reset_Counter (RESETCNT)

Resets sensor counter values.

Parameter: int32_t SP_ResetTimestamp SP_ResetTimestamp
Direction: Down
Valid values:
 0 = No
 1 = Yes
Description: Reset timestamp value.

Parameter: int32_t SP_ResetMeasCounter SP_ResetMeasCounter

Direction: Down

Valid values:

- 0= No
- 1= Yes

Description: Reset counter value.

Parameter: int32_t SP_ResetTriggerCounter SP_ResetTriggerCounter

Direction: Down

Valid values:

- 0= No
- 1= Yes

Description: Reset trigger counter value.

10.6.1.1.11 Set_Echo (ECHO)

Set echo for sensor commands.

Parameter: int32_t SP_Echo SP_Echo

Direction: Down

Valid values:

- 0= Off
- 1= On

Description: Echo mode.

10.6.1.1.12 Get_Echo (ECHO)

Get the echo mode.

Parameter: int32_t SA_Echo SA_Echo

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Off
- 1= On

Description: Echo mode.

10.6.1.1.13 Get_AllParameters (PRINT)

Get all parameters from sensor.

Parameter: int32_t SP_Additional SP_Additional

Direction: Down

Valid values:

- 0= No
- 1= Yes

Description: If set, additional information about sensor, sensor and material is output.

Parameter: int32_t SA_SyncMode SA_SyncMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = None
- 1 = Slave
- 2 = Master
- 3 = Master alternating (MASTER_ALT)

Description: Synchronization mode.

Parameter: int32_t SA_SyncTermination SA_SyncTermination

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Off (TERMOFF)
- 1 = On 120 Ohm (TERMON)

Description: Specifies if termination should be activated.

Parameter: int32_t SA_UserLevel SA_UserLevel

Direction: Up

Valid values:

- 1 = Other user level (only for internal use)
- 0 = User
- 1 = Professional

Description: Actual user level.

Parameter: int32_t SA_DefaultUser SA_DefaultUser

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = User
- 1 = Professional

Description: Default user level.

Parameter: int32_t SA_Echo SA_Echo

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Off
- 1 = On

Description: Echo mode.

Parameter: int32_t SA_DisplayUnit SA_DisplayUnit

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = mm
- 1 = Inch

Description: Unit.

Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Edge	
2= Level (PULSE)	
3= Software	
Description: Trigger mode.	
Parameter: int32_t SA_TriggerTermination	SA_TriggerTermination
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off (TERMOFF)	
1= On 120 Ohm (TERMON)	
Description: Specifies if termination should be activated.	
Parameter: int32_t SA_TriggerMoment	SA_TriggerMoment
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Input	
1= Output	
Description: Trigger moment.	
Parameter: int32_t SA_TriggerLevel	SA_TriggerLevel
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= High	
1= Low	
Description: Trigger level.	
Parameter: int32_t SA_TriggerCount	SA_TriggerCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 16383	
Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.	
Parameter: int32_t SA_TriggerOutput	SA_TriggerOutput
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Triggered values (TRIGGERED)	
1= All values (ALL)	
Description: Output only triggered values or all values.	
Parameter: int32_t SA_EthernetMode	SA_EthernetMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Ethernet	
1= Ethercat	
Description: Ethernet mode.	

Parameter: int32_t SA_DHCPEnabled	SA_DHCPEnabled
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = FALSE	
1 = TRUE	
Description: Get settings if sensor should use a static IP address ask for IP at DHCP server (dynamic IP address).	
Parameter: String SA_Address	SA_Address
Direction: Up	
Valid values:	
Valid IP address in form of xxx.xxx.xxx.xxx	
Description: IP address of the sensor. If DHCP is enabled it returns the currently assigned IP address.	
Parameter: String SA_SubnetMask	SA_SubnetMask
Direction: Up	
Valid values:	
Valid network mask (e.g. 255.255.255.0 for a Class C network)	
Description: Network mask of the sensor. If DHCP is enabled it returns the currently assigned network mask.	
Parameter: String SA_Gateway	SA_Gateway
Direction: Up	
Valid values:	
Valid IP address of default gateway in form of xxx.xxx.xxx.xxx	
Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.	
Parameter: int32_t SA_Protocol	SA_Protocol
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = TCP server (SERVER/TCP)	
1 = TCP client (CLIENT/TCP)	
2 = UDP sender (CLIENT/UDP)	
3 = None	
Description: Specifies if data should be send using TCP or UDP.	
Parameter: String SA_RemoteAddress	SA_RemoteAddress
Direction: Up	
Valid values:	
Valid IP address of receiver of data	
Description: Address of remote computer to send data to.	
Parameter: int32_t SA_Port	SA_Port
Direction: Up	
Valid values:	
Minimum: 1024	
Maximum: 65535	
Description: Port to send data to or to listen for incoming requests.	

Parameter: int32_t SA_SensorBaudrate	SA_SensorBaudrate
Direction: Up	
Valid values:	
4000000	
3500000	
3000000	
2500000	
2000000	
1500000	
921600	
691200	
460800	
230400	
115200	
9600	
Unit: Baud	
Description: Baudrate of sensor.	
Parameter: int32_t SA_ActiveParameterSet	SA_ActiveParameterSet
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 8	
Description: Number of the currently active parameter set in RAM.	
Parameter: int32_t SA_MeasurePeak	SA_MeasurePeak
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Greatest Amplitude (DISTA)	
1= Greatest Area (DISTW)	
2= First Peak (DIST1)	
Description: Peak to measure.	
Parameter: int32_t SA_MeasureMode	SA_MeasureMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Distance diffuse (DIST_DIFFUSE)	
1= Distance direct (DIST_DIRECT)	
2= Thickness	
3= Video	
Description: Measure mode.	
Parameter: int32_t SA_ShutterMode	SA_ShutterMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Exposure time is adapted automatically (MEAS)	
1= Exposure time is set manually (MANUAL)	
2= Use two fixed exposure times alternately (2TIMEALT)	
3= Use adapted time and factor * adapted time (see Set_ShutterFactor) alternately (AUTOTIMEALT)	
Description: Shutter mode.	

Parameter: double SA_Measrate SA_Measrate

Direction: Up

Valid values:

- 1.5
- 2.5
- 5.0
- 10.0
- 20.0
- 30.0
- 49.0

Unit: kHz

Description: Samplerate of measurement.

Parameter: double SA_ShutterTime1 SA_ShutterTime1

Direction: Up

Valid values:

- Minimum:** 0.1
- Maximum:** 512.0

Unit: μ s

Description: First shutter time.

Parameter: double SA_ShutterTime2 SA_ShutterTime2

Direction: Up

Valid values:

- Minimum:** 0.1
- Maximum:** 512.0

Unit: μ s

Description: Second shutter time.

Parameter: int32_t SA_LaserPower SA_LaserPower

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Full
- 1 = Reduced
- 2 = Off

Description: Laser power.

Parameter: int32_t SA_ROIStart SA_ROIStart

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 511

Unit: Pixel

Description: First position on CCD.

Parameter: int32_t SA_ROIEnd SA_ROIEnd

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 511

Unit: Pixel

Description: Last position on CCD.

Parameter: int32_t SA_VideoAverage SA_VideoAverage

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Recursive over 2 lines (REC2)
- 2= Recursive over 4 lines (REC4)
- 3= Recursive over 8 lines (REC8)
- 4= Moving over 2 lines (MOV2)
- 5= Moving over 3 lines (MOV3)
- 6= Moving over 4 lines (MOV4)
- 7= Median over 3 lines (MED3)

Description: Averaging mode.

Parameter: double SA_Threshold SA_Threshold

Direction: Up

Valid values:

Minimum: 0.0
Maximum: 99.0

Unit: %

Description: Video threshold.

Parameter: String SA_ActiveMaterial SA_ActiveMaterial

Direction: Up

Description: Name of material.

Parameter: int32_t SA_AveragingType SA_AveragingType

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Moving average (MOVING)
- 2= Recursive averaging (RECURSIVE)
- 3= Median

Description: Averaging type.

Parameter: int32_t SA_MovingCount SA_MovingCount

Direction: Up

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128

Description: Number of value for the averaging window. This parameter is only available at moving average.

Parameter: int32_t SA_RecursiveCount SA_RecursiveCount

Direction: Up

Valid values:

Minimum: 2
Maximum: 32768

Description: Number of values for recursive averaging. This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount	SA_MedianCount
Direction: Up	
Valid values:	
3	
5	
7	
9	
Description: Number of values to build median. This parameter is only available at median.	
Parameter: int32_t SA_SpikeCorrection	SA_SpikeCorrection
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = off	
1 = on	
Description: Spike correction.	
Parameter: int32_t SA_NbrEvaluatedValues	SA_NbrEvaluatedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Number of values to evaluate for spike correction.	
Parameter: double SA_ToleranceRange	SA_ToleranceRange
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 100.0	
Description: Tolerance range for spike correction.	
Parameter: int32_t SA_NbrCorrectedValues	SA_NbrCorrectedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100	
Description: Number of values to correct at spike correction.	
Parameter: int32_t SA_StatisticDepth	SA_StatisticDepth
Direction: Up	
Valid values:	
Minimum: 2	
Maximum: 2147483647 (INT32_MAX)	
Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 16384). Value greater as 16384 calculates statistic over all values.	
Parameter: int32_t SA_Master	SA_Master
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = no (NONE)	
1 = yes (MASTER)	
Description: Specifies if mastering is active.	

Parameter: double SA_MasterValue	SA_MasterValue
Direction: Up	
Valid values:	
Minimum: -2* measuring range	
Maximum: +2* measuring range	
Unit: mm	
Description: Master value	
Parameter: int32_t SA_DataOutInterface	SA_DataOutInterface
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= RS422	
2= Ethernet	
3= HTTP	
Description: Active interface for data output.	
Parameter: int32_t SA_Resampling	SA_Resampling
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 3000000	
Description: Resampling value.	
Parameter: int32_t SA_ResampleRS422	SA_ResampleRS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: RS422 output is resampled.	
Parameter: int32_t SA_ResampleEthernet	SA_ResampleEthernet
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Output over ethernet is resampled.	
Parameter: int32_t SA_HoldLastValid	SA_HoldLastValid
Direction: Up	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	
Parameter: double SA_MeasureValueCnt	SA_MeasureValueCnt
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Number of values to be output. 0 means no output, 4294967295 means continuous output.	

Parameter: int32_t SA_OutputAdditionalIntensity1_RS422	SA_OutputAdditionalIntensity1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity 1 is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity2_RS422	SA_OutputAdditionalIntensity2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity 2 is transmitted.	
Parameter: int32_t SA_OutputDistance1_RS422	SA_OutputDistance1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 is transmitted.	
Parameter: int32_t SA_OutputDistance2_RS422	SA_OutputDistance2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 2 is transmitted.	
Parameter: int32_t SA_OutputThickness12_RS422	SA_OutputThickness12_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if thickness between first and second peak is transmitted.	
Parameter: int32_t SA_OutputStatisticMin_RS422	SA_OutputStatisticMin_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if min value is transmitted.	
Parameter: int32_t SA_OutputStatisticMax_RS422	SA_OutputStatisticMax_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if max value is transmitted.	
Parameter: int32_t SA_OutputStatisticPeak2Peak_RS422	SA_OutputStatisticPeak2Peak_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if peak to peak value is transmitted.	

Parameter: int32_t SA_OutputThickness12_ETH	SA_OutputThickness12_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if thickness between first and second peak is transmitted.	
Parameter: int32_t SA_OutputStatisticMin_ETH	SA_OutputStatisticMin_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if min value is transmitted.	
Parameter: int32_t SA_OutputStatisticMax_ETH	SA_OutputStatisticMax_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if max value is transmitted.	
Parameter: int32_t SA_OutputStatisticPeak2Peak_ETH	SA_OutputStatisticPeak2Peak_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if peak to peak value is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_RS422	SA_OutputAdditionalShutterTime_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_RS422	SA_OutputAdditionalIntensity_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity is transmitted.	

Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalTemperature_RS422	SA_OutputAdditionalTemperature_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity1_ETHERNET	SA_OutputAdditionalIntensity1_ETHERNET
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity 1 is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity2_ETHERNET	SA_OutputAdditionalIntensity2_ETHERNET
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity 2 is transmitted.	
Parameter: int32_t SA_OutputDistance1_ETHERNET	SA_OutputDistance1_ETHERNET
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 is transmitted.	
Parameter: int32_t SA_OutputDistance2_ETHERNET	SA_OutputDistance2_ETHERNET
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 2 is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_ETHERNET	SA_OutputAdditionalShutterTime_ETHERNET
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_ETHERNET	SA_OutputAdditionalCounter_ETHERNET
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	

Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_ETH	SA_OutputAdditionalIntensity_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_ETH	SA_OutputAdditionalState_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalTrgCounter_ETH	SA_OutputAdditionalTrgCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTemperature_ETH	SA_OutputAdditionalTemperature_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature is transmitted.	
Parameter: int32_t SA_OutputVideoRaw_ETH	SA_OutputVideoRaw_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal is transmitted.	
Parameter: int32_t SA_OutputVideoCorr_ETH	SA_OutputVideoCorr_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if corrected video signal is transmitted.	
Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Direction: Up	
Description: Name of the sensor.	

Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the sensor.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the sensor.	
Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description: MAC address (low level ethernet address) of the sensor.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the sensor.	
Parameter: String SA_ImageType	SA_ImageType
Direction: Up	
Description: Firmware image type used by the sensor.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Unit: mm	
Description: Range of the sensor.	
Parameter: String SA_CalibrationTable	SA_CalibrationTable
Direction: Up	
Description: Name of calibration table which is actually used.	
Parameter: String SA_MaterialName	SA_MaterialName
Direction: Up	
Description: Name of the active material.	
Parameter: String SA_Description	SA_Description
Direction: Up	
Description: Description of the active material.	
Parameter: double SA_RefractiveIndex_nF	SA_RefractiveIndex_nF
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material.	

Parameter: double SA_WaveLength SA_WaveLength
Direction: Up
Unit: nm
Description: The wave length where the refractive index taken from.

10.6.1.2 User level

10.6.1.2.1 Logout (LOGOUT)

Change user level to user.

10.6.1.2.2 Login (LOGIN)

Change user level to professional.

Parameter: String SP_Password SP_Password
Direction: Down
Description: Valid password to login.

10.6.1.2.3 Get_UserLevel (GETUSERLEVEL)

Retrieve actual user level.

Parameter: int32_t SA_UserLevel SA_UserLevel
Direction: Up
Valid values:
 -1 = Other user level (only for internal use)
 0 = User
 1 = Professional
Description: Actual user level.

10.6.1.2.4 Set_DefaultUser (STDUSER)

Set the default user level after booting the system.

Parameter: int32_t SP_DefaultUser SP_DefaultUser
Direction: Down
Valid values:
 0 = User
 1 = Professional
Description: Default user level.

10.6.1.2.5 Get_DefaultUser (STDUSER)

Get the default user level after booting the system.

Parameter: int32_t SA_DefaultUser SA_DefaultUser
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = User
 1 = Professional
Description: Default user level.

10.6.1.2.6 Set_Password (PASSWD)

Change the password for login.

Parameter: String SP_OldPassword	SP_OldPassword
Direction: Down	
Description: Old password.	
Parameter: String SP_NewPassword	SP_NewPassword
Direction: Down	
Description: New password.	

10.6.1.3 Triggering

10.6.1.3.1 Set_TriggerMode (TRIGGER)

Set the trigger mode.

Parameter: int32_t SP_TriggerMode	SP_TriggerMode
Direction: Down	
Valid values:	
0= None	
1= Edge	
2= Level (PULSE)	
3= Software	
Description: Trigger mode.	
Parameter: int32_t SP_TriggerTermination	SP_TriggerTermination
Direction: Down	
Valid values:	
0= Off (TERMOFF)	
1= On 120 Ohm (TERMON)	
Description: Termination of external input.	

10.6.1.3.2 Get_TriggerMode (TRIGGER)

Get the active trigger mode.

Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Edge	
2= Level (PULSE)	
3= Software	
Description: Trigger mode.	

Parameter: int32_t SA_TriggerTermination SA_TriggerTermination
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = Off (TERMOFF)
 1 = On 120 Ohm (TERMON)
Description: Termination of external input.

10.6.1.3.3 Set_TriggerMoment (TRIGGERAT)

Set the trigger time.

Parameter: int32_t SP_TriggerMoment SP_TriggerMoment
Direction: Down
Valid values:
 0 = Input
 1 = Output
Description: Trigger moment.

10.6.1.3.4 Get_TriggerMoment (TRIGGERAT)

Get the active trigger time.

Parameter: int32_t SA_TriggerMoment SA_TriggerMoment
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = Input
 1 = Output
Description: Trigger moment.

10.6.1.3.5 Set_TriggerLevel (TRIGGERLEVEL)

Set the trigger level.

Parameter: int32_t SP_TriggerLevel SP_TriggerLevel
Direction: Down
Valid values:
 0 = High
 1 = Low
Description: Trigger level.

10.6.1.3.6 Get_TriggerLevel (TRIGGERLEVEL)

Get the active trigger level.

Parameter: int32_t SA_TriggerLevel SA_TriggerLevel
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = High
 1 = Low
Description: Trigger level.

10.6.1.3.7 Set_TriggerCount (TRIGGERCOUNT)

Set the number of values to measure at trigger.

Parameter: int32_t SP_TriggerCount SP_TriggerCount
Direction: Down
Valid values:
Minimum: 0
Maximum: 16383
Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

10.6.1.3.8 Get_TriggerCount (TRIGGERCOUNT)

Get the number of values to measure at trigger.

Parameter: int32_t SA_TriggerCount SA_TriggerCount
Direction: Up
Valid values:
Minimum: 0
Maximum: 16383
Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

10.6.1.3.9 Software_Trigger (TRIGGERSW)

Execute a software trigger.

10.6.1.3.10 Set_TriggerOutput (TRIGGEROUT)

Specifies which values should be output at trigger mode.

Parameter: int32_t SP_TriggerOutput SP_TriggerOutput
Direction: Down
Valid values:
 0= Triggered values (TRIGGERED)
 1= All values (ALL)
Description: Output only triggered values or all values.

10.6.1.3.11 Get_TriggerOutput (TRIGGEROUT)

Specifies which values should be output at trigger mode.

Parameter: int32_t SA_TriggerOutput SA_TriggerOutput
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Triggered values (TRIGGERED)
 1= All values (ALL)
Description: Output only triggered values or all values.

10.6.1.4 Interfaces

10.6.1.4.1 Set_IPConfiguration (IPCONFIG)

Set the IP configuration at sensor.

For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_DHCPEnabled SP_DHCPEnabled

Direction: Down

Valid values:

- 0= FALSE
- 1= TRUE

Description: Specify if sensor should use a static IP address or ask for IP at DHCP server (dynamic IP address).

Parameter: String SP_Address SP_Address

Direction: Down

Valid values:

- Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the sensor. This parameter is only evaluated on static IP assignment.

Parameter: String SP_SubnetMask SP_SubnetMask

Direction: Down

Valid values:

- Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the sensor. This parameter is only evaluated on static IP assignment.

Parameter: String SP_Gateway SP_Gateway

Direction: Down

Valid values:

- Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: The default gateway must be specified if the sensor should communicate with peers in foreign subnets. This parameter is only evaluated on static IP assignment.

10.6.1.4.2 Get_IPConfiguration (IPCONFIG)

Get the IP configuration at sensor.

Parameter: int32_t SA_DHCPEnabled SA_DHCPEnabled

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= FALSE
- 1= TRUE

Description: Get settings if sensor should use a static IP address ask for IP at DHCP server (dynamic IP address).

Parameter: String SA_Address	SA_Address
Direction: Up	
Valid values:	Valid IP address in form of xxx.xxx.xxx.xxx
Description:	IP address of the sensor. If DHCP is enabled it returns the currently assigned IP address.
Parameter: String SA_SubnetMask	SA_SubnetMask
Direction: Up	
Valid values:	Valid network mask (e.g. 255.255.255.0 for a Class C network)
Description:	Network mask of the sensor. If DHCP is enabled it returns the currently assigned network mask.
Parameter: String SA_Gateway	SA_Gateway
Direction: Up	
Valid values:	Valid IP address of default gateway in form of xxx.xxx.xxx.xxx
Description:	Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

10.6.1.4.3 Set_IPDataTransferMode (MEATRANSFER)

Set IP protocol at sensor.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_Protocol	SP_Protocol
Direction: Down	
Valid values:	0= TCP server (SERVER/TCP) 1= TCP client (CLIENT/TCP) 2= UDP sender (CLIENT/UDP) 3= None
Description:	Specifies if data should be send using TCP or UDP.
Parameter: String SP_RemoteAddress	SP_RemoteAddress
Direction: Down	
Valid values:	Valid IP address of receiver of data
Description:	Address of remote computer to send data to. On TCP server this parameter is ignored.
Parameter: int32_t SP_Port	SP_Port
Direction: Down	
Valid values:	Minimum: 1024 Maximum: 65535
Description:	Port to send data to or to listen for incoming requests.

10.6.1.4.4 Get_IPDataTransferMode (MEASTRANSFER)

Get IP protocol at sensor.

Parameter: int32_t SA_Protocol

SA_Protocol

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= TCP server (SERVER/TCP)
- 1= TCP client (CLIENT/TCP)
- 2= UDP sender (CLIENT/UDP)
- 3= None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SA_RemoteAddress

SA_RemoteAddress

Direction: Up

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to.

Parameter: int32_t SA_Port

SA_Port

Direction: Up

Valid values:

- Minimum:** 1024
Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

10.6.1.4.5 Set_EthernetMode (ETHERMODE)

Switches ethernet mode between Ethernet and Ethercat.

Parameter: int32_t SP_EthernetMode

SP_EthernetMode

Direction: Down

Valid values:

- 0= Ethernet
- 1= Ethercat

Description: Ethernet mode.

10.6.1.4.6 Get_EthernetMode (ETHERMODE)

Get ethernet mode of sensor.

Parameter: int32_t SA_EthernetMode

SA_EthernetMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Ethernet
- 1= Ethercat

Description: Ethernet mode.

10.6.1.4.7 Set_Baudrate (BAUDRATE)

Set baudrate of sensor for serial RS422 communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

- 4000000
- 3500000 (does not work in combination with IF2008 PCI interface card)
- 3000000 (does not work in combination with IF2008 PCI interface card)
- 2500000
- 2000000
- 1500000
- 921600
- 691200
- 460800
- 230400
- 115200
- 9600

Unit: Baud

Description: Baudrate of sensor.

Parameter: int32_t CP_InterruptDataTransfer

CP_InterruptDataTransfer

Direction: Down

Valid values:

- 0= false
- 1= true

Default: 0

Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Set_DataOutInterface](#)) and enabling it again at end (only if enabled before).

10.6.1.4.8 Get_Baudrate (BAUDRATE)

Get baudrate of sensor for serial RS422 communication.

Parameter: int32_t SA_SensorBaudrate

SA_SensorBaudrate

Direction: Up

Valid values:

- 4000000
- 3500000
- 3000000
- 2500000
- 2000000
- 1500000
- 921600
- 691200
- 460800
- 230400
- 115200
- 9600

Unit: Baud

Description: Baudrate of sensor.

10.6.1.5 Parameter management

10.6.1.5.1 Save_Parameters (STORE)

Save actual parameters at sensor. There can be saved several settings on different locations. So it is easy to switch to another setting.

Parameter: int32_t SP_ParameterSet

SP_ParameterSet

Direction: Down

Valid values:

Minimum: 1
Maximum: 8

Description: Location to save the settings.

10.6.1.5.2 Load_Parameters (READ)

Load stored parameters into sensor RAM.

There can be loaded several settings from different locations. So it is easy to switch to another setting.

At this command the sensor may change output data after applying new setting.

If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_ParameterType

SP_ParameterType

Direction: Down

Valid values:

0= All settings (ALL)
1= Device settings (DEVICE)
2= Measurement settings (MEAS)

Description: Specifies which settings should be loaded.

Parameter: int32_t SP_ParameterSet

SP_ParameterSet

Direction: Down

Valid values:

Minimum: 1
Maximum: 8

Description: Location from where the settings should be loaded.

10.6.1.5.3 Get_ActiveParameterSet (READ)

Get the number of the currently active parameter set in RAM.

Parameter: int32_t SA_ActiveParameterSet

SA_ActiveParameterSet

Direction: Up

Valid values:

Minimum: 0
Maximum: 8

Description: Number of the currently active parameter set in RAM.

10.6.1.5.4 Set_Default (SETDEFAULT)

Reset the sensor to default settings.

At this command the sensor may change output data after applying default settings. If first bit of `IP_AutomaticMode` is set (1), `Get_AllParameters` (`SP_Additional= 1`) is called automatically after this command. Otherwise, you have to call it manually. After this call `Get_TransmittedDataInfo` to retrieve new setting.

Parameter: `int32_t SP_DefaultType`

`SP_DefaultType`

Direction: Down

Valid values:

- 0= All settings (ALL)
- 1= Keep device settings temporary (NODEVICE)
- 2= Only material table (MATERIAL)

Description: Specifies which settings should be reset.

10.6.2 Measurement

10.6.2.1 General

10.6.2.1.1 Set_MeasureMode (MEASMODE)

Set the measure mode.

At this command the sensor may change output data automatically. If first bit of `IP_AutomaticMode` is set (1), `Get_OutputInfo_RS422` resp. `Get_OutputInfo_ETH` (or `Get_AllParameters` (`SP_Additional= 0`) for older firmware versions) is called automatically after this command. Otherwise, you have to call it manually. After this call `Get_TransmittedDataInfo` to retrieve new setting.

Parameter: `int32_t SP_MeasureMode`

`SP_MeasureMode`

Direction: Down

Valid values:

- 0= Distance diffuse (DIST_DIFFUSE)
- 1= Distance direct (DIST_DIRECT)
- 2= Thickness
- 3= Video

Description: Measure mode.

10.6.2.1.2 Get_MeasureMode (MEASMODE)

Get the measure mode.

Parameter: `int32_t SA_MeasureMode`

`SA_MeasureMode`

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Distance diffuse (DIST_DIFFUSE)
- 1= Distance direct (DIST_DIRECT)
- 2= Thickness
- 3= Video

Description: Measure mode.

10.6.2.1.3 Set_MeasurePeak (MEASPEAK)

Select the peak to measure.

Parameter: int32_t SP_MeasurePeak SP_MeasurePeak
Direction: Down
Valid values:
 0= Greatest Amplitude (DISTA)
 1= Greatest Area (DISTW)
 2= First Peak (DIST1)
Description: Peak to measure.

10.6.2.1.4 Get_MeasurePeak (MEASPEAK)

Get the selected peak to measure.

Parameter: int32_t SA_MeasurePeak SA_MeasurePeak
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Greatest Amplitude (DISTA)
 1= Greatest Area (DISTW)
 2= First Peak (DIST1)
Description: Peak to measure.

10.6.2.1.5 Get_Video (GETVIDEO)

Get recent video signals from sensor.

Parameter: Binary data SA_VideoRaw SA_VideoRaw
Direction: Up
Valid values:
 512 words (each 2 byte), each word is an intensity value.
Description: Raw video signal

Parameter: Binary data SA_VideoCorr SA_VideoCorr
Direction: Up
Valid values:
 512 words (each 2 byte), each word is an intensity value.
Description: Filtered video signal

Parameter: double SA_VideoTimestamp SA_VideoTimestamp
Direction: Up
Valid values:
Minimum: 0
Maximum: 1.79769e+308 (DBL_MAX)
Unit: ms
Description: Timestamp of the video signal. It starts from 1970 Jan 01 at 01:00. It is generated when the video has arrived at TCP/IP socket.

Example how to read a video signal from sensor:

```

/* Do not forget to handle potential error after each call to MEDAQLib! */
/* Create sensor instance, open sensor via TCP/IP, set output to ethernet */
/* and then switch to video mode: */
err= SetIntExecSCmd (instance, "Set_MeasureMode", "SP_MeasureMode", 3 /*Video*/);

/* Select the desired video signal: */
err= SetParameterInt (instance, "SP_OutputVideoRaw_ETH", 1);
err= SetParameterInt (instance, "SP_OutputVideoCorr_ETH", 1);
err= ExecSCmd (instance, "Set_OutputVideo_ETH");

/* Acquire video signals: */
err= ExecSCmd (instance, "Get_Video");
uint16_t videoRaw[512], videoCorr[512];
uint32_t maxLen= sizeof (videoRaw);
err= GetParameterBinary (instance, "SA_VideoRaw", (uint8_t *)videoRaw, &maxLen);
assert (maxLen==sizeof (videoRaw)); // additional validity check
maxLen= sizeof (videoCorr);
err= GetParameterBinary (instance, "SA_VideoCorr", (uint8_t *)videoCorr, &maxLen);
assert (maxLen==sizeof (videoCorr)); // additional validity check

/* Do anything with the received video signals */

```

10.6.2.1.6 Set_ShutterMode (SHUTTERMODE)

Set the shutter mode. This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_ShutterMode

SP_ShutterMode

Direction: Down

Valid values:

- 0= Exposure time is adapted automatically (MEAS)
- 1= Exposure time is set manually (MANUAL)
- 2= Use two fixed exposure times alternately (2TIMEALT)
- 3= Use adapted time and factor * adapted time (see [Set_ShutterFactor](#)) alternately (AUTOTIMEALT)

Description: Shutter mode.

10.6.2.1.7 Get_ShutterMode (SHUTTERMODE)

Get the shutter mode. This is an internal command. It should not be used by the customer.

Parameter: int32_t SA_ShutterMode

SA_ShutterMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Exposure time is adapted automatically (MEAS)
- 1= Exposure time is set manually (MANUAL)
- 2= Use two fixed exposure times alternately (2TIMEALT)
- 3= Use adapted time and factor * adapted time (see [Set_ShutterFactor](#)) alternately (AUTOTIMEALT)

Description: Shutter mode.

10.6.2.1.8 Set_Samplerate (MEASRATE)

Set the samplerate.

Parameter: double SP_Measrate

SP_Measrate

Direction: Down

Valid values:

1.5
2.5
5.0
10.0
20.0
30.0
49.0

Unit: kHz

Description: Samplerate of measurement.

10.6.2.1.9 Get_Samplerate (MEASRATE)

Get the samplerate.

Parameter: double SA_Measrate

SA_Measrate

Direction: Up

Valid values:

1.5
2.5
5.0
10.0
20.0
30.0
49.0

Unit: kHz

Description: Samplerate of measurement.

10.6.2.1.10 Set_ShutterTime (SHUTTER)

Set the fixed exposure times. This is an internal command. It should not be used by the customer.

Parameter: double SP_ShutterTime1

SP_ShutterTime1

Direction: Down

Valid values:

Minimum: 0.1
Maximum: 1638.0

Unit: μ s

Description: First exposure time.

Parameter: double SP_ShutterTime2

SP_ShutterTime2

Direction: Down

Valid values:

Minimum: 0.1
Maximum: 1638.0

Unit: μ s

Description: Second exposure time.

10.6.2.1.11 Get_ShutterTime (SHUTTER)

Get the fixed exposure times. This is an internal command. It should not be used by the customer.

Parameter: double SA_ShutterTime1

SA_ShutterTime1

Direction: Up

Valid values:

Minimum: 0.1

Maximum: 512.0

Unit: μs

Description: First exposure time.

Parameter: double SA_ShutterTime2

SA_ShutterTime2

Direction: Up

Valid values:

Minimum: 0.1

Maximum: 512.0

Unit: μs

Description: Second exposure time.

10.6.2.1.12 Set_ShutterFactor (SHUTTERFACTOR)

Set the factor for [Set_ShutterMode](#) when SP_ShutterMode is 3. This is an internal command. It should not be used by the customer.

Parameter: double SP_ShutterFactor

SP_ShutterFactor

Direction: Down

Valid values:

Minimum: 1.0

Maximum: 100.0

Description: Factor for the second (alternating) exposure time.

10.6.2.1.13 Get_ShutterFactor (SHUTTERFACTOR)

Get the factor for [Set_ShutterMode](#) when SP_ShutterMode is 3. This is an internal command. It should not be used by the customer.

Parameter: double SA_ShutterFactor

SA_ShutterFactor

Direction: Up

Valid values:

Minimum: 1.0

Maximum: 100.0

Description: Factor for the second (alternating) exposure time.

10.6.2.1.14 Set_LaserPower (LASERPOW)

Specify the laser power at sensor.

Parameter: int32_t SP_LaserPower

SP_LaserPower

Direction: Down

Valid values:

0= Full

1= Reduced

2= Off

Description: Laser power.

10.6.2.1.15 Get_LaserPower (LASERPOW)

Get the laser power from sensor.

Parameter: int32_t SA_LaserPower	SA_LaserPower
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Full	
1 = Reduced	
2 = Off	
Description: Laser power.	

10.6.2.2 Video signal

10.6.2.2.1 Set_ROI (ROI)

Set the region of interest for processing video signal.

Parameter: int32_t SP_ROIStart	SP_ROIStart
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: First pixel.	
Parameter: int32_t SP_ROIEnd	SP_ROIEnd
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: Last pixel.	

10.6.2.2.2 Get_ROI (ROI)

Get the region of interest for processing video signal.

Parameter: int32_t SA_ROIStart	SA_ROIStart
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: First pixel.	
Parameter: int32_t SA_ROIEnd	SA_ROIEnd
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: Last pixel.	

10.6.2.2.3 Set_VideoAverage (VSAVERAGE)

Set video averaging (before processing).

Parameter: int32_t SP_VideoAverage

SP_VideoAverage

Direction: Down

Valid values:

- 0= None
- 1= Recursive over 2 lines (REC2)
- 2= Recursive over 4 lines (REC4)
- 3= Recursive over 8 lines (REC8)
- 4= Moving over 2 lines (MOV2)
- 5= Moving over 3 lines (MOV3)
- 6= Moving over 4 lines (MOV4)
- 7= Median over 3 lines (MED3)

Description: Averaging mode.

10.6.2.2.4 Get_VideoAverage (VSAVERAGE)

Get video averaging (before processing).

Parameter: int32_t SA_VideoAverage

SA_VideoAverage

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Recursive over 2 lines (REC2)
- 2= Recursive over 4 lines (REC4)
- 3= Recursive over 8 lines (REC8)
- 4= Moving over 2 lines (MOV2)
- 5= Moving over 3 lines (MOV3)
- 6= Moving over 4 lines (MOV4)
- 7= Median over 3 lines (MED3)

Description: Averaging mode.

10.6.2.2.5 Set_Threshold (THRESHOLD)

Set threshold for video processing. This is an internal command. It should not be used by the customer.

Parameter: double SP_Threshold

SP_Threshold

Direction: Down

Valid values:

- Minimum:** 0.0
- Maximum:** 100.0

Unit: %

Description: Video threshold.

10.6.2.2.6 Get_Threshold (THRESHOLD)

Get threshold for video processing. This is an internal command. It should not be used by the customer.

Parameter: double SA_Threshold SA_Threshold
Direction: Up
Valid values:
 Minimum: 0.0
 Maximum: 100.0
Unit: %
Description: Video threshold.

10.6.2.3 Material database

10.6.2.3.1 Get_MaterialTable (MATERIALTABLE)

Get a list of all materials for thickness calculation.

Parameter: String SA_MaterialTable SA_MaterialTable
Direction: Up
Description: Whole table in one string, separated by new lines and commas.

Parameter: double SA_WaveLength SA_WaveLength
Direction: Up
Unit: nm
Description: The wave length where the refractive index taken from.

Parameter: int32_t SA_MaterialTableCount SA_MaterialTableCount
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 20
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Pos1, SA_Pos2, ...

Parameter: int32_t SA_Pos1..x SA_Pos1..x
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 19
Description: Index of the material in the table.

Parameter: String SA_MaterialName1..x SA_MaterialName1..x
Direction: Up
Description: Name of the material in the table.

Parameter: double SA_RefRACTIVEINDEX_nF1..x SA_RefRACTIVEINDEX_nF1..x
Direction: Up
Valid values:
 Minimum: 1.0
 Maximum: 4.0
Description: Refractive index of the material.

Parameter: String SA_Description1..x SA_Description1..x
Direction: Up
Description: Description of the material in the table.

10.6.2.3.2 Set_ActiveMaterial (MATERIAL)

Set the active material for thickness calculation.

Parameter: String SP_ActiveMaterial SP_ActiveMaterial
Direction: Down
Description: Name of material.

10.6.2.3.3 Get_ActiveMaterial (MATERIAL)

Get the active material for thickness calculation.

Parameter: String SA_ActiveMaterial SA_ActiveMaterial
Direction: Up
Description: Name of material.

10.6.2.3.4 Get_MaterialInfo (MATERIALINFO)

Get information of active material.

Parameter: String SA_MaterialName SA_MaterialName
Direction: Up
Description: Name of the active material.

Parameter: String SA_Description SA_Description
Direction: Up
Description: Description of the active material.

Parameter: double SA_RefRACTIVEINDEX_nF SA_RefRACTIVEINDEX_nF
Direction: Up
Valid values:
Minimum: 1.0
Maximum: 4.0
Description: Refractive index of the active material.

Parameter: double SA_WaveLength SA_WaveLength
Direction: Up
Unit: nm
Description: The wave length where the refractive index taken from.

10.6.2.3.5 Edit_Material (MATERIALEDIT)

Edit or add new material by using refractive index.

Parameter: String SP_MaterialName SP_MaterialName
Direction: Down
Description: Name of the material.

Parameter: String SP_Description	SP_Description
Direction: Down	
Description: Description of the material.	
Parameter: double SP_RefRACTIVEINDEX_nF	SP_RefRACTIVEINDEX_nF
Direction: Down	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material.	

10.6.2.3.6 Delete_Material (MATERIALDELETE)

Deletes an existing material.

Parameter: String SP_MaterialName	SP_MaterialName
Direction: Down	
Description: Name of the material to delete.	

10.6.2.3.7 Clear_MaterialTable

Clear the whole material table.

10.6.2.4 Measurement value processing

10.6.2.4.1 Set_Averaging (AVERAGE)

Set data averaging at sensor.

Parameter: int32_t SP_AveragingType	SP_AveragingType
Direction: Down	
Valid values:	
0= None	
1= Moving average (MOVING)	
2= Recursive averaging (RECURSIVE)	
3= Median	
Description: Averaging type.	

Parameter: int32_t SP_MovingCount	SP_MovingCount
Direction: Down	
Valid values:	
2	
4	
8	
16	
32	
64	
128	

Description: Number of value for the averaging window. This parameter is only used at moving average.

Parameter: int32_t SP_RecursiveCount SP_RecursiveCount

Direction: Down

Valid values:

Minimum: 2

Maximum: 32768

Description: Number of values for recursive averaging. This parameter is only used at recursive average.

Parameter: int32_t SP_MedianCount SP_MedianCount

Direction: Down

Valid values:

3

5

7

9

Description: Number of values to build median. This parameter is only used at median.

10.6.2.4.2 Get_Averaging (AVERAGE)

Get data averaging at sensor.

Parameter: int32_t SA_AveragingType SA_AveragingType

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= None

1= Moving average (MOVING)

2= Recursive averaging (RECURSIVE)

3= Median

Description: Averaging type.

Parameter: int32_t SA_MovingCount SA_MovingCount

Direction: Up

Valid values:

2

4

8

16

32

64

128

Description: Number of value for the averaging window. This parameter is only available at moving average.

Parameter: int32_t SA_RecursiveCount SA_RecursiveCount

Direction: Up

Valid values:

Minimum: 2

Maximum: 32768

Description: Number of values for recursive averaging. This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount SA_MedianCount

Direction: Up

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only available at median.

10.6.2.4.3 Set_SpikeCorrection (SPIKECORR)

Set spike correction at sensor.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_SpikeCorrection SP_SpikeCorrection

Direction: Down

Valid values:

- 0= off
- 1= on

Description: Spike correction.

Parameter: int32_t SP_NbrEvaluatedValues SP_NbrEvaluatedValues

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 10

Description: Number of values to evaluate for spike correction.

Parameter: double SP_ToleranceRange SP_ToleranceRange

Direction: Down

Valid values:

- Minimum:** 0.0
- Maximum:** 100.0

Description: Tolerance range for spike correction.

Parameter: int32_t SP_NbrCorrectedValues SP_NbrCorrectedValues

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 100

Description: Number of values to correct at spike correction.

10.6.2.4.4 Get_SpikeCorrection (SPIKECORR)

Get spike correction at sensor.

Parameter: int32_t SA_SpikeCorrection SA_SpikeCorrection

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= off
- 1= on

Description: Spike correction.

Parameter: int32_t SA_NbrEvaluatedValues	SA_NbrEvaluatedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Number of values to evaluate for spike correction.	
Parameter: double SA_ToleranceRange	SA_ToleranceRange
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 100.0	
Description: Tolerance range for spike correction.	
Parameter: int32_t SA_NbrCorrectedValues	SA_NbrCorrectedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100	
Description: Number of values to correct at spike correction.	

10.6.2.4.5 Set_StatisticDepth (STATISTICDEPTH)

Set the window size for floating statistic calculation.

Parameter: int32_t SP_StatisticDepth	SP_StatisticDepth
Direction: Down	
Valid values:	
Minimum: 2	
Maximum: 2147483647 (INT32_MAX)	
Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 16384). Value greater as 16384 calculates statistic over all values.	

10.6.2.4.6 Get_StatisticDepth (STATISTICDEPTH)

Get the window size for floating statistic calculation.

Parameter: int32_t SA_StatisticDepth	SA_StatisticDepth
Direction: Up	
Valid values:	
Minimum: 2	
Maximum: 2147483647 (INT32_MAX)	
Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 16384). Value greater as 16384 calculates statistic over all values.	

10.6.2.4.7 Reset_Statistic (RESETSTATISTIC)

Reset the statistic (min and max values).

10.6.2.4.8 Set_MasterValue (MASTERMV)

Set the master value.

Parameter: int32_t SP_Master SP_Master

Direction: Down

Valid values:

0 = no (NONE)

1 = yes (MASTER)

Description: Specifies if mastering should be done or resetted.

Parameter: double SP_MasterValue SP_MasterValue

Direction: Down

Valid values:

Minimum: -2* measuring range

Maximum: +2* measuring range

Unit: mm

Description: Master value

10.6.2.4.9 Get_MasterValue (MASTERMV)

Get the master value.

Parameter: int32_t SA_Master SA_Master

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor

0 = no (NONE)

1 = yes (MASTER)

Description: Specifies if mastering is active.

Parameter: double SA_MasterValue SA_MasterValue

Direction: Up

Valid values:

Minimum: -2* measuring range

Maximum: +2* measuring range

Unit: mm

Description: Master value

10.6.3 Data output

10.6.3.1 General

10.6.3.1.1 Set_DataOutInterface (OUTPUT)

Set the active interface for data output.

At this command the sensor may change output data automatically. If first bit of **IP_AutomaticMode** is set (1), [Get_OutputInfo_RS422](#) resp. [Get_OutputInfo_ETH](#) (or [Get_AllParameters](#) (SP_Additional= 0) for older firmware versions) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DataOutInterface SP_DataOutInterface
Direction: Down
Valid values:
 0= None
 1= RS422
 2= Ethernet
 3= HTTP
Description: Active interface for data output.

10.6.3.1.2 Get_DataOutInterface (OUTPUT)

Get the active interface for data output.

Parameter: int32_t SA_DataOutInterface SA_DataOutInterface
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= None
 1= RS422
 2= Ethernet
 3= HTTP
Description: Active interface for data output.

10.6.3.1.3 Set_Resampling (OUTREDUCE)

Set resampling to reduce output data.
 For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_Resampling SP_Resampling
Direction: Down
Valid values:
Minimum: 1
Maximum: 3000000
Description: Resampling value.

Parameter: int32_t SP_ResampleRS422 SP_ResampleRS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if RS422 output should be resampled.

Parameter: int32_t SP_ResampleEthernet SP_ResampleEthernet
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if output over ethernet should be resampled.

10.6.3.1.4 Get_Resampling (OUTREDUCE)

Get resampling for reducing output data.

Parameter: int32_t SA_Resampling	SA_Resampling
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 3000000	
Description: Resampling value.	
Parameter: int32_t SA_ResampleRS422	SA_ResampleRS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: RS422 output is resampled.	
Parameter: int32_t SA_ResampleEthernet	SA_ResampleEthernet
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Output over ethernet is resampled.	

10.6.3.1.5 Set_HoldLastValid (OUTHOLD)

Set the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SP_HoldLastValid	SP_HoldLastValid
Direction: Down	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	

10.6.3.1.6 Get_HoldLastValid (OUTHOLD)

Get the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SA_HoldLastValid	SA_HoldLastValid
Direction: Up	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	

10.6.3.1.7 Set_MeasureValueCnt (GETVALUE)

Set the number of values to be output. If reached, the output stops.

Parameter: double SP_MeasureValueCnt

SP_MeasureValueCnt

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Number of values to be output. 0 means no output, 4294967295 means continuous output.

10.6.3.1.8 Get_MeasureValueCnt (GETVALUE)

Get the number of values to be output.

Parameter: double SA_MeasureValueCnt

SA_MeasureValueCnt

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Number of values to be output. 0 means no output, 4294967295 means continuous output.

10.6.3.2 Selected measurement values

10.6.3.2.1 Set_OutputDistance_RS422 (OUTDIST_RS422)

Set the distance data to be output at RS422 interface.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputDistance1_RS422

SP_OutputDistance1_RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if distance 1 is transmitted.

Parameter: int32_t SP_OutputDistance2_RS422

SP_OutputDistance2_RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if distance 2 is transmitted.

10.6.3.2.2 Get_OutputDistance_RS422 (OUTDIST_RS422)

Get the distance data to be output at RS422 interface.

Parameter: int32_t SA_OutputDistance1_RS422

SA_OutputDistance1_RS422

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if distance 1 is transmitted.

Parameter: int32_t SA_OutputDistance2_RS422 SA_OutputDistance2_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if distance 2 is transmitted.

10.6.3.2.3 Set_OutputThickness_RS422 (OUTTHICK_RS422)

Set the thickness data to be output at RS422 interface.
 For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputThickness12_RS422 SP_OutputThickness12_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if thickness between first and second peak is transmitted.

10.6.3.2.4 Get_OutputThickness_RS422 (OUTTHICK_RS422)

Get the thickness data to be output at RS422 interface.

Parameter: int32_t SA_OutputThickness12_RS422 SA_OutputThickness12_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if thickness between first and second peak is transmitted.

10.6.3.2.5 Set_OutputStatistic_RS422 (OUTSTATIC_RS422)

Set the statistic data to be output at RS422 interface.
 For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputStatisticMin_RS422 SP_OutputStatisticMin_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if min value is transmitted.

Parameter: int32_t SP_OutputStatisticMax_RS422 SP_OutputStatisticMax_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if max value is transmitted.

10.6.3.2.6 Get OutputStatistic_RS422 (OUTSTATISTIC_RS422)

Get the statistic data to be output at RS422 interface.

Parameter: int32_t SA_OutputStatisticPeak2Peak_RS422

Valid values:
0 = no
1 = yes

Description: Specify if peak to peak value is transmitted.

10.6.3.2.7 Set OutputStatistic ETH (OUTSTATISTIC ETH)

Set the statistic data to be output at ethernet interface.
For this command an [Update](#) and a [Reset](#) meta command is available.

Parameter: int32_t SP_OutputStatisticMin_ETH
Direction: Down
Valid values:
 0 = no
 1 = yes
Description: Specify if min value is transmitted

Parameter: int32_t SP_OutputStatisticMax_ETH
Direction: Down
Valid values:
 0 = no
 1 = yes
Description: Specify if max value is transmitted.

Parameter: int32_t SP_OutputStatisticPeak2Peak_ETH SP_OutputStatisticPeak2Peak_ETH
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if peak to peak value is transmitted.

10.6.3.2.8 Get_OutputStatistic_ETH (OUTSTATISTIC_ETH)

Get the statistic data to be output at ethernet interface.

Parameter: int32_t SA_OutputStatisticMin_ETH SA_OutputStatisticMin_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if min value is transmitted.

Parameter: int32_t SA_OutputStatisticMax_ETH SA_OutputStatisticMax_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if max value is transmitted.

Parameter: int32_t SA_OutputStatisticPeak2Peak_ETH SA_OutputStatisticPeak2Peak_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if peak to peak value is transmitted.

10.6.3.2.9 Set_OutputAdditional_RS422 (OUTADD_RS422)

Set the additional data to be output at RS422 interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputAdditionalShutterTime_RS422 SP_OutputAdditionalShutterTime_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if shutter time is transmitted.

Parameter: int32_t SP_OutputAdditionalCounter_RS422 SP_OutputAdditionalCounter_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if counter is transmitted.

Parameter: int32_t SP_OutputAdditionalTimestamp_RS422	SP_OutputAdditionalTimestamp_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SP_OutputAdditionalIntensity_RS422	SP_OutputAdditionalIntensity_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity is transmitted.	
Parameter: int32_t SP_OutputAdditionalState_RS422	SP_OutputAdditionalState_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SP_OutputAdditionalTemperature_RS422	SP_OutputAdditionalTemperature_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature is transmitted.	

10.6.3.2.10 Get_OutputAdditional_RS422 (OUTADD_RS422)

Get the additional data to be output at RS422 interface.

Parameter: int32_t SA_OutputAdditionalShutterTime_RS422	SA_OutputAdditionalShutterTime_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	

Parameter: int32_t SA_OutputAdditionalIntensity_RS422	SA_OutputAdditionalIntensity_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalTemperature_RS422	SA_OutputAdditionalTemperature_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature is transmitted.	

10.6.3.2.11 Set_OutputAdditional_ETH (OUTADD_ETH)

Set the additional data to be output at ethernet interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputAdditionalShutterTime_ETH	SP_OutputAdditionalShutterTime_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SP_OutputAdditionalCounter_ETH	SP_OutputAdditionalCounter_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SP_OutputAdditionalTimestamp_ETH	SP_OutputAdditionalTimestamp_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SP_OutputAdditionalIntensity_ETH	SP_OutputAdditionalIntensity_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity is transmitted.	

Parameter: int32_t SP_OutputAdditionalState_ETH	SP_OutputAdditionalState_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SP_OutputAdditionalTrgCounter_ETH	SP_OutputAdditionalTrgCounter_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger counter is transmitted.	
Parameter: int32_t SP_OutputAdditionalTemperature_ETH	SP_OutputAdditionalTemperature_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature is transmitted.	

10.6.3.2.12 Get_OutputAdditional_ETH (OUTADD_ETH)

Get the additional data to be output at ethernet interface.

Parameter: int32_t SA_OutputAdditionalShutterTime_ETH	SA_OutputAdditionalShutterTime_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditionalCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_ETH	SA_OutputAdditionalIntensity_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity is transmitted.	

Parameter: int32_t SA_OutputAdditionalState_ETH	SA_OutputAdditionalState_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalTrgCounter_ETH	SA_OutputAdditionalTrgCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTemperature_ETH	SA_OutputAdditionalTemperature_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature is transmitted.	

10.6.3.2.13 Set_OutputVideo_ETH (OUTVIDEO)

Set the video signal to be output at ethernet interface.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputVideoRaw_ETH	SP_OutputVideoRaw_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal is transmitted.	
Parameter: int32_t SP_OutputVideoCorr_ETH	SP_OutputVideoCorr_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if corrected video signal is transmitted.	

10.6.3.2.14 Get_OutputVideo_ETH (OUTVIDEO)

Get the video signal to be output at ethernet interface.

Parameter: int32_t SA_OutputVideoRaw_ETH	SA_OutputVideoRaw_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal is transmitted.	
Parameter: int32_t SA_OutputVideoCorr_ETH	SA_OutputVideoCorr_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if corrected video signal is transmitted.	

11 Commands for confocalDT (IFD) sensors

IFD2400 and IFD2430 are no longer supported.

11.1 Commands for IFD2401 and IFD2431

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (native).
- [TCP/IP](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface).
- [WinUSB](#) (native).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Status](#), [Get_SpectralAv](#), [Get_FreeSR](#), [Get_DoubleFreq](#) and [Get_Frequencies](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate, to interpret and scale data and to assign values.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor, from 0 to 32767.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_Status](#)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

11.1.1 General commands

11.1.1.1 General

11.1.1.1.1 Get_Status (STS)

Retrieve detailed information about the controller and sensor.

Parameter: int32_t SA_SRIndex

SA_SRIndex

Direction: Up

Valid values:

- 0= Free samplerate
- 1= 100 Hz (at IFD2401) or 500 Hz (at IFD2431)
- 2= 200 Hz (at IFD2401) or 1000 Hz (at IFD2431)
- 3= 400 Hz (at IFD2401) or 2000 Hz (at IFD2431)
- 4= 1000 Hz (at IFD2401) or 5000 Hz (at IFD2431)
- 5= 2000 Hz (at IFD2401) or 10000 Hz (at IFD2431)
- 6= 15625 Hz (only at IFD2431)
- 7= 20000 Hz (only at IFD2431)
- 8= 25000 Hz (only at IFD2431)
- 9= 31250 Hz (only at IFD2431)

Description: Samplerate index.

Parameter: int32_t SA_MeasureMode	SA_MeasureMode
Direction: Up	
Valid values:	
0= Distance	
1= Thickness	
Description: Measure mode of the sensor.	
Parameter: int32_t SA_Sensor	SA_Sensor
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Number of active sensor.	
Parameter: int32_t SA_ASCII	SA_ASCII
Direction: Up	
Valid values:	
0= Binary	
1= ASCII	
Description: Data Transfer mode (values) of sensor.	
Parameter: int32_t SA_Averaging	SA_Averaging
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 9999	
Description: Data averaging factor.	
Parameter: int32_t SA_X1..16	SA_X1..16
Direction: Up	
Valid values:	
0= off	
1= RS232/RS422	
9= USB	
Description: Selection of data of the sensor. This is necessary for data conversion and scaling.	
Depending on measure mode (distance / thickness) the selected value have different meaning. The following list shows the meaning for X1..16.	
X1 means (Distance / Thickness).	
X2 means (not used / Distance1).	
X3 means (Auto adaptive mode data / Distance2).	
X4 means (Intensity / Auto adaptive mode data).	
X5 means (not used / Intensity1).	
X6 means (not used / Intensity2).	
X7 means (Barycenter / Barycenter1).	
X8 means (not used / Barycenter2).	
X9 means (State Flags / State Flags).	
X10 means (Counter / Counter).	
X11 means (Encoder 1 LSB / Encoder 1 LSB).	
X12 means (Encoder 1 MSB / Encoder 1 MSB).	
X13 means (Encoder 2 LSB / Encoder 2 LSB).	
X14 means (Encoder 2 MSB / Encoder 2 MSB).	
X15 means (Encoder 3 LSB / Encoder 3 LSB).	
X16 means (Encoder 3 MSB / Encoder 3 MSB).	

Parameter: int32_t SA_0_SOD	SA_0_SOD
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7	
Description: Analog out BNC1: 0= X1, 1= X2, ..., 15= X16.	
Parameter: int32_t SA_0_0V	SA_0_0V
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 99999	
Description: Analog out BNC1: Specifies the value which shoud output 0 V.	
Parameter: int32_t SA_0_10V	SA_0_10V
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 99999	
Description: Analog out BNC1: Specifies the value which shoud output 10 V.	
Parameter: int32_t SA_1_SOD	SA_1_SOD
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7	
Description: Analog out BNC2: 0= X1, 1= X2, ..., 15= X16.	
Parameter: int32_t SA_1_0V	SA_1_0V
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 99999	
Description: Analog out BNC2: Specifies the value which shoud output 0 V.	
Parameter: int32_t SA_1_10V	SA_1_10V
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 99999	
Description: Analog out BNC2: Specifies the value which shoud output 10 V.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Unit: μ m	
Valid values:	
Minimum: 0.0	
Maximum: 2147483647.0 (INT32_MAX)	
Description: Range of active sensor.	

11.1.1.1.2 Get_Version (VER)

Get the software version of the controller.

Parameter: String SA_Version	SA_Version
Direction: Up	
Description: Controller name, serial number and version	

11.1.1.1.3 Reset (RST)

Reset the controller and set the default parameter values.

Only valid for very early firmware versions (any before V1.2.54), removed at later versions.

11.1.1.2 Sensor

11.1.1.2.1 Set_ActiveSensor (SEN)

Set the active sensor (up to 20 sensors can be stored in controller) for measurement. If first bit of [IP_AutomaticMode](#) is set (1), [Get_Range](#) is called automatically after this command. Otherwise, you have to call it manually.

Parameter: int32_t SP_Sensor

SP_Sensor

Direction: Down

Valid values:

Minimum: 0

Maximum: 19

Description: Number of active sensor.

11.1.1.2.2 Get_ActiveSensor (SEN?)

Get the active sensor.

Parameter: int32_t SA_Sensor

SA_Sensor

Direction: Up

Valid values:

Minimum: 0

Maximum: 19

Description: Number of active sensor.

11.1.1.2.3 Get_Range (SCA)

Get the range of active sensor.

Parameter: double SA_Range

SA_Range

Direction: Up

Unit: μm

Valid values:

Minimum: 0.0

Maximum: 2147483647.0 (INT32_MAX)

Description: Range of active sensor.

11.1.1.2.4 Get_AllRanges (LUL)

Get the ranges of all calibrated sensors.

Parameter: String SA_Ranges

SA_Ranges

Direction: Up

Unit: μm (after conversion from string to int32_t or double)

Valid values:

0.0 .. 2147483647.0 (INT32_MAX)

Description: Ranges of all 20 sensors in a string, separated by commas.

11.1.1.2.5 Acquire_DarkSig (DRK)

Acquire dark signal.

Parameter: int32_t SA_MinSRIndex SA_MinSRIndex

Direction: Up

Valid values:

- 1 = 100 Hz (at IFD2401) or 500 Hz (at IFD2431)
- 2 = 200 Hz (at IFD2401) or 1000 Hz (at IFD2431)
- 3 = 400 Hz (at IFD2401) or 2000 Hz (at IFD2431)
- 4 = 1000 Hz (at IFD2401) or 5000 Hz (at IFD2431)
- 5 = 2000 Hz (at IFD2401) or 10000 Hz (at IFD2431)
- 6 = 15625 Hz (only at IFD2431)
- 7 = 20000 Hz (only at IFD2431)
- 8 = 25000 Hz (only at IFD2431)
- 9 = 31250 Hz (only at IFD2431)

Description: Minimal samplerate index.

11.1.1.2.6 FastDark (FDK)

Acquire dark signal for active sensor and samplerate.

Parameter: int32_t SP_AveragingForDark SP_AveragingForDark

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 99

Description: Averaging factor for dark.

Parameter: int32_t SP_Weighting SP_Weighting

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 100

Description: Weighting factor.

11.1.1.2.7 Set_AutoDark (ADK)

Enables/Disables the "Automatic Fast Dark" mode.

Parameter: int32_t SP_AutoDark SP_AutoDark

Direction: Down

Valid values:

- 0 = off
- 1 = on

Description: Automatic Fast Dark.

11.1.1.2.8 Get_AutoDark (ADK?)

Get the state of the "Automatic Fast Dark" mode.

Parameter: int32_t SA_AutoDark

SA_AutoDark

Direction: Up

Valid values:

0 = off

1 = on

Description: Automatic Fast Dark.

11.1.1.3 Triggering

11.1.1.3.1 Continue (CTN)

Continues data acquisition (after trigger commands SingleShot_Trg, TriggerMode_Edge, TriggerMode_State or an error).

11.1.1.3.2 Start_Trigger (TRG)

Starts the trigger mode at controller.

11.1.1.3.3 End_Trigger

Stops the trigger mode at controller (after Start_Trigger).

11.1.1.3.4 SingleShot_Trg (TRE)

Starts the single shot trigger mode at controller.

Parameter: int32_t SP_NumberOfPoints

SP_NumberOfPoints

Direction: Down

Valid values:

Minimum: 0

Maximum: 9999

Description: Number of data frames (each frame can consist of different values) to read.

11.1.1.3.5 Set_TrgMode_Edge (TRS)

Enable/Disable the "Start/Stop on Edge Trigger" mode.

Parameter: int32_t SP_TriggerMode_Edge

SP_TriggerMode_Edge

Direction: Down

Valid values:

Minimum: 0

Maximum: 1

Description: Start/Stop on Edge Trigger state.

11.1.1.3.6 Set_TrgMode_State (TRN)

Enable/Disable the "Start/Stop on State Trigger" mode.

Parameter: int32_t SP_TriggerMode_State

SP_TriggerMode_State

Direction: Down

Valid values:

Minimum: 0

Maximum: 1

Description: Start/Stop on State Trigger state.

11.1.1.3.7 Set_ActiveEdge (TRF)

Set the active edge or state for Start_Trigger, TriggerMode_Edge or TriggerMode_State.

Parameter: int32_t SP_ActiveEdge

SP_ActiveEdge

Direction: Down

Valid values:

0= rising edge or high state

1= falling edge or low state

Description: Active state or edge.

11.1.1.3.8 Get_ActiveEdge (TRF?)

Get the active edge or state for Start_Trigger, TriggerMode_Edge or TriggerMode_State.

Parameter: int32_t SA_ActiveEdge

SA_ActiveEdge

Direction: Up

Valid values:

0= rising edge or high state

1= falling edge or low state

Description: Active state or edge.

11.1.1.3.9 Software_Trigger (STR)

Simulates an hardware trigger (SYNC IN) for trigger commands SingleShot_Trg, TriggerMode_Edge or TriggerMode_State.

Only available from firmware version V1.2.56

11.1.1.4 Encoder

11.1.1.4.1 RecenterEncoder (RCD)

Recenter encoder position.

Parameter: int32_t SP_Encoder1

SP_Encoder1

Direction: Down

Valid values:

Minimum: 0

Maximum: 1

Description: Recenter Encoder 1.

Parameter: int32_t SP_Encoder2 SP_Encoder2

Direction: Down

Valid values:

Minimum: 0

Maximum: 1

Description: Recenter Encoder 2.

Parameter: int32_t SP_Encoder3 SP_Encoder3

Direction: Down

Valid values:

Minimum: 0

Maximum: 1

Description: Recenter Encoder 3.

11.1.1.5 Interfaces

11.1.1.5.1 Set_Baudrate (BAU)

Set the baudrate of the serial interface of controller. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Attention! To avoid synchronization problems, sending data by sensor should be disabled after changing baudrate for a short time (sensor command Start_Trigger).

Parameter: int32_t SP_SensorBaudrate SP_SensorBaudrate

Direction: Down

Valid values:

9600

19200

38400

57600

115200

230400

460800

Description: Baudrate of controller.

11.1.1.5.2 Get_Baudrate (BAU?)

Get the baudrate of the serial interface of controller.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate

Direction: Up

Valid values:

9600

19200

38400

57600

115200

230400

460800

Description: Baudrate of controller.

11.1.1.6 Parameter management

11.1.1.6.1 Save_Setup (SSU)

Save the current setup of controller to flash.

11.1.1.7 Internal controller commands

11.1.1.7.1 Set_LampTest (SLP)

Enable/Disable the "Lamp Test" mode.

Valid for sensor:

IFD2401

Parameter: int32_t SP_LampTest

SP_LampTest

Direction: Down

Valid values:

Minimum: 0

Maximum: 1

Description: Lamp Test state.

11.1.1.7.2 Get_LampTest (SLP?)

Get the current "Lamp Test" mode.

Valid for sensor:

IFD2401

Parameter: int32_t SA_LampTest

SA_LampTest

Direction: Up

Valid values:

Minimum: 0

Maximum: 1

Description: Lamp Test state.

11.1.1.7.3 Set_LampTestThr (CSL)

Set the threshold for "Lamp Test" mode.

Valid for sensor:

IFD2401

Parameter: int32_t SP_LampTestThr

SP_LampTestThr

Direction: Down

Valid values:

Minimum: 0

Maximum: 998

Description: Lamp Test threshold.

11.1.1.7.4 Get_LampTestThr (CSL?)

Get the threshold for "Lamp Test" mode.

Valid for sensor:

IFD2401

Parameter: int32_t SA_LampTestThr

SA_LampTestThr

Direction: Up

Valid values:

Minimum: 0

Maximum: 998

Description: Lamp Test threshold.

11.1.1.7.5 Set_Watchdog (WDE)

Enable/Disable the Watchdog.

Parameter: int32_t SP_Watchdog

SP_Watchdog

Direction: Down

Valid values:

Minimum: 0

Maximum: 1

Description: Watchdog state.

11.1.1.7.6 Get_Watchdog (WDE?)

Get the Watchdog state.

Parameter: int32_t SA_Watchdog

SA_Watchdog

Direction: Up

Valid values:

Minimum: 0

Maximum: 1

Description: Watchdog state.

11.1.1.7.7 Set_WatchdogPrd (WDP)

Set the Watchdog period.

Parameter: int32_t SP_WatchdogPeriod

SP_WatchdogPeriod

Direction: Down

Unit: s

Valid values:

Minimum: 10

Maximum: 255

Description: Watchdog period.

11.1.1.7.8 Get_WatchdogPrd (WDP?)

Get the Watchdog period.

Parameter: int32_t SA_WatchdogPeriod

SA_WatchdogPeriod

Direction: Up

Unit: s

Valid values:

Minimum: 10

Maximum: 255

Description: Watchdog period.

11.1.2 Measurement

11.1.2.1 General

11.1.2.1.1 Set_MeasureMode (MOD)

Set the measure mode.

Parameter: int32_t SP_MeasureMode

SP_MeasureMode

Direction: Down

Valid values:

0= Distance

1= Thickness

Description: Measure mode of the sensor.

11.1.2.1.2 Get_MeasureMode (MOD?)

Get the current measure mode.

Parameter: int32_t SA_MeasureMode

SA_MeasureMode

Direction: Up

Valid values:

0= Distance

1= Thickness

Description: Measure mode of the sensor.

11.1.2.1.3 Set_FirstPeakMode (MSP)

Enable/Disable the "First peak" mode.

Parameter: int32_t SP_FirstPeak

SP_FirstPeak

Direction: Down

Valid values:

Minimum: 0

Maximum: 1

Description: First peak state.

11.1.2.1.4 Get_FirstPeakMode (MSP?)

Get the current "First peak" mode.

Parameter: int32_t SA_FirstPeak SA_FirstPeak

Direction: Up

Valid values:

Minimum: 0

Maximum: 1

Description: First peak state.

11.1.2.1.5 Set_SRIndex (SRA)

Set the samplerate for data acquisition.

Parameter: int32_t SP_SRIndex SP_SRIndex

Direction: Down

Valid values:

0= Free samplerate

1= 100 Hz (at IFD2401) or 500 Hz (at IFD2431)

2= 200 Hz (at IFD2401) or 1000 Hz (at IFD2431)

3= 400 Hz (at IFD2401) or 2000 Hz (at IFD2431)

4= 1000 Hz (at IFD2401) or 5000 Hz (at IFD2431)

5= 2000 Hz (at IFD2401) or 10000 Hz (at IFD2431)

6= 15625 Hz (only at IFD2431)

7= 20000 Hz (only at IFD2431)

8= 25000 Hz (only at IFD2431)

9= 31250 Hz (only at IFD2431)

Description: Samplerate index.

11.1.2.1.6 Get_SRIndex (SRA?)

Get the current samplerate.

Parameter: int32_t SA_SRIndex SA_SRIndex

Direction: Up

Valid values:

0= Free samplerate

1= 100 Hz (at IFD2401) or 500 Hz (at IFD2431)

2= 200 Hz (at IFD2401) or 1000 Hz (at IFD2431)

3= 400 Hz (at IFD2401) or 2000 Hz (at IFD2431)

4= 1000 Hz (at IFD2401) or 5000 Hz (at IFD2431)

5= 2000 Hz (at IFD2401) or 10000 Hz (at IFD2431)

6= 15625 Hz (only at IFD2431)

7= 20000 Hz (only at IFD2431)

8= 25000 Hz (only at IFD2431)

9= 31250 Hz (only at IFD2431)

Description: Samplerate index.

11.1.2.1.7 Set_FreeSR (FRQ)

Set free samplerate value for data acquisition.

Parameter: int32_t SP_FreeSR SP_FreeSR

Direction: Down

Valid values:

Minimum: 100

Maximum: 2000 (at IFD2401) or 31250 (at IFD2431)

Unit: Hz

Description: Free Frequency.

Parameter: int32_t SA_FreeSR SA_FreeSR

Direction: Up

Valid values:

Minimum: 100

Maximum: 2000 (at IFD2401) or 31250 (at IFD2431)

Unit: Hz

Description: Free Samplerate.

11.1.2.1.8 Get_FreeSR (FRQ?)

Get free samplerate value.

Parameter: int32_t SA_FreeSR SA_FreeSR

Direction: Up

Valid values:

Minimum: 100

Maximum: 2000 (at IFD2401) or 31250 (at IFD2431)

Unit: Hz

Description: Free Samplerate.

11.1.2.1.9 Set_Exposure (TEX)

Set free exposure time for data acquisition.

Parameter: int32_t SP_Exposure SP_Exposure

Direction: Down

Valid values:

Minimum: 500 (at IFD2401) or 32 (at IFD2431)

Maximum: 10000 (at IFD2401) or 2000 (at IFD2431)

Unit: μ s

Description: Free Exposure time.

Parameter: int32_t SA_Exposure SA_Exposure

Direction: Up

Valid values:

Minimum: 500 (at IFD2401) or 32 (at IFD2431)

Maximum: 10000 (at IFD2401) or 2000 (at IFD2431)

Unit: μ s

Description: Free Exposure time.

11.1.2.1.10 Get_Exposure (TEX?)

Get free exposure time.

Parameter: int32_t SA_Exposure SA_Exposure
Direction: Up
Valid values:
Minimum: 500 (at IFD2401) or 32 (at IFD2431)
Maximum: 10000 (at IFD2401) or 2000 (at IFD2431)
Unit: μ s
Description: Free Exposure time.

11.1.2.1.11 Get_MinSR (FRM)

Get the minimum authorized samplerate.

Parameter: int32_t SA_MinSR SA_MinSR
Direction: Up
Valid values:
Minimum: 100
Maximum: 2000 (at IFD2401) or 31250 (at IFD2431)
Unit: Hz
Description: Minimum samplerate (determined by dark signal).

11.1.2.1.12 Set_DoubleFreq (DFA)

Enables or disables the double frequency mode.

Valid for sensor:
 IFD2401 from firmware version V1.2.56

Parameter: int32_t SP_DoubleFrequency SP_DoubleFrequency
Direction: Down
Valid values:
 0= Disable
 1= Enable
Description: Double frequency mode.

11.1.2.1.13 Get_DoubleFreq (DFA?)

Get the double frequency mode.

Valid for sensor:
 IFD2401 from firmware version V1.2.56

Parameter: int32_t SA_DoubleFrequency SA_DoubleFrequency
Direction: Up
Valid values:
 0= Disable
 1= Enable
Description: Double frequency mode.

11.1.2.1.14 Set_Frequencies (DFF)

Set the frequencies for double frequency mode.
 SP_HighFrequency must be higher than SP_LowFrequency.

Valid for sensor:

IFD2401 from firmware version V1.2.56

Parameter: int32_t SP_LowFrequency

SP_LowFrequency

Direction: Down

Valid values:

Minimum: 100

Maximum: 1850

Unit: Hz

Description: Low frequency.

Parameter: int32_t SP_HighFrequency

SP_HighFrequency

Direction: Down

Valid values:

Minimum: 100

Maximum: 1850

Unit: Hz

Description: High frequency.

11.1.2.1.15 Get_Frequencies (DFF?)

Get the frequencies of double frequency mode.

Valid for sensor:

IFD2401 from firmware version V1.2.56

Parameter: int32_t SA_LowFrequency

SA_LowFrequency

Direction: Up

Valid values:

Minimum: 100

Maximum: 1850

Unit: Hz

Description: Low frequency.

Parameter: int32_t SA_HighFrequency

SA_HighFrequency

Direction: Up

Valid values:

Minimum: 100

Maximum: 1850

Unit: Hz

Description: High frequency.

11.1.2.1.16 Get_CCD (CCD)

Returns current CCD data.

Parameter: Binary data SA_CCD

SA_CCD

Direction: Up

Valid values:

4096 (at IFD2401) or 2048 (at IFD2431) bytes, convertible to 2048 (at IFD2401) or 1024 (at IFD2431) words.

Description: Raw CCD line.

11.1. Commands for IFD2401 and IFD2431

Parameter: Binary data SA_PreTreated SA_PreTreated
Direction: Up
Valid values:
 4096 (at IFD2401) or 2048 (at IFD2431) bytes, convertible to 2048 (at IFD2401) or 1024 (at IFD2431) words.
Description: Raw CCD line.

11.1.2.1.17 Get_DarkSig (SGD)

Get the dark signal table of controller.

Parameter: Binary data SA_DarkSig SA_DarkSig
Direction: Up
Valid values:
 4096 (at IFD2401) or 2048 (at IFD2431) bytes, convertible to 2048 (at IFD2401) or 1024 (at IFD2431) words.
Description: Dark signal table.

11.1.2.1.18 Start_Spectrum

Start the spectrum mode in controller. Only work via USB connection.
 On firmware versions below 1.2.36 the spectrum is shifted 8 bytes to left.
 This mode does work up to 4.7 kHz only. For higher samplerates data will be lost and the spectrums are corrupted. To avoid this, increase spectral averaging.

Parameter: int32_t IP_UsbReadBufSize IP_UsbReadBufSize
Direction: Down
Unit: Bytes
Valid values:
Minimum: 2
Maximum: 1048576
Default: 131072
Description: Buffer size for read operations on USB, while spectrum mode is active. The value is always ceiled to the next power of two (512, 1024, 2048, ..., 32768, 65536).

Parameter: int32_t IP_TimerResolution IP_TimerResolution
Direction: Down
Unit: ms
Valid values:
 -1 = Do not set timer resolution.
 0 = Use greatest possible accuracy.
 1..2147483647 (INT32_MAX) = Resolution in milliseconds.
Unit: ms
Default: 0
Description: Timer resolution (for Windows scheduler, set by timeBeginPeriod).
 It is automatically reset at End_Spectrum.

11.1.2.1.19 Get_Spectrum

Read one Spectrum (PreTreated Signal). It is returned with a frequency of Samplerate devided by Spectral Averaging.

Parameter: int32_t SP_WaitSpectrumTimeout	SP_WaitSpectrumTimeout
Direction: Down	
Unit: ms	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Timeout to wait for a spectrum.	
 Parameter: int32_t SP_ReadMode	SP_ReadMode
Direction: Down	
Valid values:	
0= Each spectrum	
1= Only newest spectrum	
2= Automatic	
Description: This mode specifies if each spectrum should be read or only the latest one. If set to automatic each spectrum is read until the buffer does not overflow. If the buffer becomes full one or more spectra are discarded.	
 Parameter: Binary data SA_Spectrum	SA_Spectrum
Direction: Up	
Valid values:	
4096 (at IFD2401) or 2048/1024 (IFD2431, depends on binning) bytes, convertible to 2048 (at IFD2401) or 1024/512 (IFD2431, depends on binning) words.	
Description: PreTreated Signal.	
 Parameter: double SA_Timestamp	SA_Timestamp
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.79769e+308 (DBL_MAX)	
Unit: ms	
Description: Timestamp of the signal. It starts from 1970 Jan 01 at 01:00. It is generated when the spectrum is read from USB	
 Parameter: int32_t SA_SkippedSpectra	SA_SkippedSpectra
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of skipped spectra, if SP_NewestSpectrum is set to 1.	

11.1.2.1.20 End_Spectrum

End the spectrum mode in controller.

Parameter: int32_t IP_UsbReadBufSize

IP_UsbReadBufSize

Direction: Down

Unit: Bytes

Valid values:

Minimum: 2

Maximum: 1048576

Default: 512

Description: Buffer size for read operations on USB, after spectrum mode was finished. The value is always ceiled to the next power of two (512, 1024, 2048, ..., 32768, 65536).

11.1.2.2 Video signal

11.1.2.2.1 Set_Threshold (MNP)

Set the detection threshold for distance mode.

Parameter: double SP_Threshold

SP_Threshold

Direction: Down

Valid values:

Minimum: 0.000001

Maximum: 1.0

Description: Threshold of the sensor.

11.1.2.2.2 Get_Threshold (MNP?)

Get the current detection threshold for distance mode.

Parameter: double SA_Threshold

SA_Threshold

Direction: Up

Valid values:

Minimum: 0.000001

Maximum: 1.0

Description: Threshold of the sensor.

11.1.2.2.3 Set_Threshold1 (SPP)

Set the detection threshold for the strongest peak in thickness mode.

Parameter: double SP_Threshold1

SP_Threshold1

Direction: Down

Valid values:

Minimum: 0.000001

Maximum: 1.0

Description: Threshold of the sensor.

11.1.2.2.4 Get_Threshold1 (SPP?)

Get the current detection threshold for the strongest peak in thickness mode.

Parameter: double SA_Threshold1 SA_Threshold1
Direction: Up
Valid values:
Minimum: 0.000001
Maximum: 1.0
Description: Threshold of the sensor.

11.1.2.2.5 Set_Threshold2 (SDP)

Set the detection threshold for the second peak in thickness mode.

Parameter: double SP_Threshold2 SP_Threshold2
Direction: Down
Valid values:
Minimum: 0.000001
Maximum: 1.0
Description: Threshold of the sensor.

11.1.2.2.6 Get_Threshold2 (SDP?)

Get the current detection threshold for the second peak in thickness mode.

Parameter: double SA_Threshold2 SA_Threshold2
Direction: Up
Valid values:
Minimum: 0.000001
Maximum: 1.0
Description: Threshold of the sensor.

11.1.2.2.7 Set_SpectralAv (AVS)

Set spectral averaging at controller.

Note! Acquisition speed will be reduced.

Parameter: int32_t SP_SpectralAv SP_SpectralAv
Direction: Down
Valid values:
Minimum: 1
Maximum: 9999
Description: Data averaging factor.

11.1.2.2.8 Get_SpectralAv (AVS?)

Returns current spectral averaging at controller.

Parameter: int32_t SA_SpectralAv	SA_SpectralAv
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 9999	
Description: Data averaging factor.	

11.1.2.2.9 Set_Binning (FBH)

Set the binning value.

This is an internal command. It should not be used by the customer.

Valid for sensor:

IFD2431

Parameter: int32_t SP_Binning	SP_Binning
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 2	
Description: Binning value	

11.1.2.2.10 Get_Binning (FBH?)

Get the binning value.

This is an internal command. It should not be used by the customer.

Valid for sensor:

IFD2431

Parameter: int32_t SA_Binning	SA_Binning
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2	
Description: Binning value	

11.1.2.3 Material database

11.1.2.3.1 Set_RefractIndex (SRI)

Set the refraction index for thickness measurement.

Parameter: double SP_RefraetIndex	SP_RefraetIndex
Direction: Down	
Valid values:	
Minimum: 0.0001	
Maximum: 9.9999	
Description: Refractive index used by the sensor for thickness measurement.	

11.1.2.3.2 Get_RefactIndex (SRI?)

Get the current refraction index.

Parameter: double SA_RefactIndex SA_RefactIndex
Direction: Up
Valid values:
Minimum: 0.0001
Maximum: 9.9999
Description: Refractive index used by the sensor for thickness measurement.

11.1.2.3.3 Set_RefIdxFile (INF)

Set the refractive index file.

Parameter: int32_t SP_RefactIndexFileIdx SP_RefactIndexFileIdx
Direction: Down
Valid values:
Minimum: 0
Maximum: 8
Description: Refractive index file index.

Parameter: int32_t SA_RefactIndexFileIdx SA_RefactIndexFileIdx
Direction: Up
Valid values:
Minimum: 0
Maximum: 8
Description: Refractive index file index.

Parameter: String SA_RefactIndexFileName SA_RefactIndexFileName
Direction: Up
Description: Refractive index file name.

Parameter: double SA_MinRefactIndex SA_MinRefactIndex
Direction: Up
Valid values:
Minimum: 0.0000
Maximum: 9.9999
Description: Minimum refractive index.

Parameter: double SA_MaxRefactIndex SA_MaxRefactIndex
Direction: Up
Valid values:
Minimum: 0.0001
Maximum: 9.9999
Description: Maximum refractive index.

Parameter: double SA_AvgRefactIndex SA_AvgRefactIndex
Direction: Up
Valid values:
Minimum: 0.0001
Maximum: 9.9999
Description: Averaging refractive index.

11.1.2.3.4 Get_RefIdxFile (INF?)

Request the selected refractive index file.

Parameter: int32_t SA_RefraIndexFileIdx	SA_RefraIndexFileIdx
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 8	
Description: Refractive index file index.	
Parameter: String SA_RefraIndexFileName	SA_RefraIndexFileName
Direction: Up	
Description: Refractive index file name.	
Parameter: double SA_MinRefractIndex	SA_MinRefractIndex
Direction: Up	
Valid values:	
Minimum: 0.0000	
Maximum: 9.9999	
Description: Minimum refractive index.	
Parameter: double SA_MaxRefractIndex	SA_MaxRefractIndex
Direction: Up	
Valid values:	
Minimum: 0.0001	
Maximum: 9.9999	
Description: Maximum refractive index.	
Parameter: double SA_AvgRefractIndex	SA_AvgRefractIndex
Direction: Up	
Valid values:	
Minimum: 0.0001	
Maximum: 9.9999	
Description: Averaging refractive index.	

11.1.2.4 Light source

11.1.2.4.1 Set_LEDIntensity (LED)

Set the LED intensity level.

Valid for sensor:
IFD2401

Parameter: int32_t SP_LEDIntensity	SP_LEDIntensity
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 100	
Description: LED intensity.	

11.1.2.4.2 Get_LEDIntensity (LED?)

Get the LED intensity level.

Valid for sensor:

IFD2401

Parameter: int32_t SA_LEDIntensity

SA_LEDIntensity

Direction: Up

Valid values:

Minimum: 0

Maximum: 100

Description: LED intensity.

11.1.2.4.3 Set_AutoAdaptLED (AAL)

Enable/Disable the "Auto-Adaptive LED" mode.

Valid for sensor:

IFD2401

Parameter: int32_t SP_AutoAdaptLED

SP_AutoAdaptLED

Direction: Down

Valid values:

Minimum: 0

Maximum: 1

Description: Auto-Adaptive LED state.

11.1.2.4.4 Get_AutoAdaptLED (AAL?)

Get the "Auto-Adaptive LED" mode.

Valid for sensor:

IFD2401

Parameter: int32_t SA_AutoAdaptLED

SA_AutoAdaptLED

Direction: Up

Valid values:

Minimum: 0

Maximum: 1

Description: Auto-Adaptive LED state.

11.1.2.4.5 Set_AdaptLEDThr (VTH)

Set the threshold value for the auto-adaptive LED mode.

Valid for sensor:

IFD2401

Parameter: int32_t SP_AutoAdaptLEDThr

SP_AutoAdaptLEDThr

Direction: Down

Valid values:

Minimum: 0

Maximum: 4095

Description: Auto-Adaptive LED threshold.

11.1.2.4.6 Get_AdaptLEDThr (VTH?)

Get the threshold value for the auto-adaptive LED mode.

Valid for sensor:

IFD2401

Parameter: int32_t SA_AutoAdaptLEDThr

SA_AutoAdaptLEDThr

Direction: Up

Valid values:

Minimum: 0

Maximum: 4095

Description: Auto-Adaptive LED threshold.

11.1.2.4.7 Set_IntLightSrc (CCL)

Set the state of the internal light source (LED).

Valid for sensor:

IFD2401

Parameter: int32_t SP_LED_Off

SP_LED_Off

Direction: Down

Valid values:

Minimum: 0

Maximum: 1

Description: LED on (0) or off (1).

11.1.2.4.8 Get_IntLightSrc (CCL?)

Get the state of the internal light source (LED).

Valid for sensor:

IFD2401

Parameter: int32_t SA_LED_Off

SA_LED_Off

Direction: Up

Valid values:

Minimum: 0

Maximum: 1

Description: LED on (0) or off (1).

11.1.2.5 Measurement value processing

11.1.2.5.1 Set_Averaging (AVR)

Set data averaging at controller. Note! Acquisition speed will be reduced.

Parameter: int32_t SP_Averaging

SP_Averaging

Direction: Down

Valid values:

Minimum: 1

Maximum: 9999

Description: Data averaging factor.

11.1.2.5.2 Get_Averaging (AVR?)

Returns current data averaging at controller.

Parameter: int32_t SA_Averaging	SA_Averaging
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 9999	

Description: Data averaging factor.

11.1.3 Data output

11.1.3.1 General

11.1.3.1.1 Set_IntensityMode (DFI)

Set normalized or raw intensity to transmit.

Valid for sensor:	IFD2401 from firmware version V1.2.56
--------------------------	---------------------------------------

Parameter: int32_t SP_TransmitIntensity	SP_TransmitIntensity
Direction: Down	
Valid values:	
0= Normalized	
1= Raw	

Description: Intensity mode.

11.1.3.1.2 Get_IntensityMode (DFI?)

Get if normalized or raw intensity is transmitted.

Valid for sensor:	IFD2401 from firmware version V1.2.56
--------------------------	---------------------------------------

Parameter: int32_t SA_TransmitIntensity	SA_TransmitIntensity
Direction: Up	
Valid values:	
0= Normalized	
1= Raw	

Description: Intensity mode.

11.1.3.1.3 Set_HoldLastValid (HLV)

Enable/Disable the "Hold Last Value" mode.

Parameter: int32_t SP_LastValid	SP_LastValid
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 999	

Description: Max number of points to hold.

11.1.3.1.4 Get_HoldLastValid (HLV?)

Get the current "Hold Last Value" mode.

Parameter: int32_t SA_LastValid SA_LastValid
Direction: Up
Valid values:
Minimum: 0
Maximum: 999
Description: Max number of points to hold.

11.1.3.1.5 Set_MissingSignal (RSP)

Set behaviour for missing second peak in thickness mode.

Parameter: int32_t SP_Option SP_Option
Direction: Down
Valid values:
Minimum: 0
Maximum: 1
Description: Mode for behaviour.

11.1.3.1.6 Get_MissingSignal (RSP?)

Returns behaviour for missing second peak in thickness mode.

Parameter: int32_t SA_Option SA_Option
Direction: Up
Valid values:
Minimum: 0
Maximum: 1
Description: Mode for behaviour.

11.1.3.1.7 Set_Reverse (RVS)

Reverse the distance signal.

Parameter: int32_t SP_Reverse SP_Reverse
Direction: Down
Valid values:
 0= Normal direction
 1= Reverse direction
Description: In reverse mode, measure range - distance is transmitted.

11.1.3.1.8 Get_Reverse (RVS?)

Returns setting for reverse mode.

Parameter: int32_t SA_Reverse SA_Reverse
Direction: Up
Valid values:
 0= Normal direction
 1= Reverse direction
Description: Reverse mode

11.1.3.1.9 Set_Ascii (ASC)

Set digital data transfer (only values, no sensor answer) to ASCII mode.

11.1.3.1.10 Set_Binary (BIN)

Set digital data transfer (only values, no sensor answer) to binary mode.

11.1.3.2 Selected measurement values

11.1.3.2.1 Set_OutputData (SOD)

Select the data to be output from controller.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_X1..16

SP_X1..16

Direction: Down

Valid values:

- 0= off
- 1= RS232/RS422
- 9= USB

Description: Selection of data of the sensor. This is necessary for data conversion and scaling.

Depending on measure mode (distance / thickness) the selected value have different meaning. The following list shows the meaning for X1..16.

X1 means (Distance / Thickness).

X2 means (not used / Distance1).

X3 means (Auto adaptive mode data / Distance2).

X4 means (Intensity / Auto adaptive mode data).

X5 means (not used / Intensity1).

X6 means (not used / Intensity2).

X7 means (Barycenter / Barycenter1).

X8 means (not used / Barycenter2).

X9 means (State Flags / State Flags).

X10 means (Counter / Counter).

X11 means (Encoder 1 LSB / Encoder 1 LSB).

X12 means (Encoder 1 MSB / Encoder 1 MSB).

X13 means (Encoder 2 LSB / Encoder 2 LSB).

X14 means (Encoder 2 MSB / Encoder 2 MSB).

X15 means (Encoder 3 LSB / Encoder 3 LSB).

X16 means (Encoder 3 MSB / Encoder 3 MSB).

11.1.3.2.2 Get_OutputData (SOD?)

Get the data output from controller.

Parameter: int32_t SA_X1..16

SA_X1..16

Direction: Up

Valid values:

0= off
 1= RS232/RS422
 9= USB

Description: Selection of data of the sensor. This is necessary for data conversion and scaling.

Depending on measure mode (distance / thickness) the selected value have different meaning. The following list shows the meaning for X1..16.

X1 means (Distance / Thickness).

X2 means (not used / Distance1).

X3 means (Auto adaptive mode data / Distance2).

X4 means (Intensity / Auto adaptive mode data).

X5 means (not used / Intensity1).

X6 means (not used / Intensity2).

X7 means (Barycenter / Barycenter1).

X8 means (not used / Barycenter2).

X9 means (State Flags / State Flags).

X10 means (Counter / Counter).

X11 means (Encoder 1 LSB / Encoder 1 LSB).

X12 means (Encoder 1 MSB / Encoder 1 MSB).

X13 means (Encoder 2 LSB / Encoder 2 LSB).

X14 means (Encoder 2 MSB / Encoder 2 MSB).

X15 means (Encoder 3 LSB / Encoder 3 LSB).

X16 means (Encoder 3 MSB / Encoder 3 MSB).

11.1.3.3 Analog output

11.1.3.3.1 Set_AnalogOut (ANA)

Setup the analog outputs.

Parameter: int32_t SP_OutNr

SP_OutNr

Direction: Down

Valid values:

0= BNC1

1= BNC2

Description: Number of analog output.

Parameter: int32_t SP_SOD

SP_SOD

Direction: Down

Valid values:

Minimum: 0

Maximum: 7

Description: Specifies the number (X1..X16) to output at analog out.

Parameter: int32_t SP_OV

SP_OV

Direction: Down

Valid values:

Minimum: 0

Maximum: 99999

Description: Specifies the value which shoud output 0 V.

Parameter: int32_t SP_10V SP_10V

Direction: Down

Valid values:

Minimum: 0

Maximum: 99999

Description: Specifies the value which shoud output 10 V.

11.1.3.3.2 Get_AnalogOut (ANA?)

Get settings of analog outputs.

Parameter: int32_t SA_0_SOD SA_0_SOD

Direction: Up

Valid values:

Minimum: 0

Maximum: 7

Description: Analog out BNC1: 0= X1, 1= X2, ..., 15= X16.

Parameter: int32_t SA_0_0V SA_0_0V

Direction: Up

Valid values:

Minimum: 0

Maximum: 99999

Description: Analog out BNC1: Specifies the value which shoud output 0 V.

Parameter: int32_t SA_0_10V SA_0_10V

Direction: Up

Valid values:

Minimum: 0

Maximum: 99999

Description: Analog out BNC1: Specifies the value which shoud output 10 V.

Parameter: int32_t SA_1_SOD SA_1_SOD

Direction: Up

Valid values:

Minimum: 0

Maximum: 7

Description: Analog out BNC2: 0= X1, 1= X2, ..., 15= X16.

Parameter: int32_t SA_1_0V SA_1_0V

Direction: Up

Valid values:

Minimum: 0

Maximum: 99999

Description: Analog out BNC2: Specifies the value which shoud output 0 V.

Parameter: int32_t SA_1_10V SA_1_10V

Direction: Up

Valid values:

Minimum: 0

Maximum: 99999

Description: Analog out BNC2: Specifies the value which shoud output 10 V.

11.1.3.3.3 Set_AnalogZero (SOF)

Set analog output to zero.

Parameter: int32_t SP_Zero

SP_Zero

Direction: Down

Valid values:

0 = normal

1 = set

Description: Set/reset the analog output 0V value.

11.1.4 Internal commands

11.1.4.1 Set_DataScale (CEE)

Set the scaling factor for X1, X2 and X3 in thickness mode.

This is an internal command. It should not be used by the customer.

Parameter: double SP_DataScale

SP_DataScale

Direction: Down

Valid values:

Minimum: 0.0001 (from firmware V1.2.56), 1.0 (for older versions)

Maximum: 5.0

Description: Scaling factor of the sensor.

11.1.4.2 Get_DataScale (CEE?)

Get the scaling factor for X1, X2 and X3 in thickness mode.

This is an internal command. It should not be used by the customer.

Parameter: double SA_DataScale

SA_DataScale

Direction: Up

Valid values:

Minimum: 0.0001 (from firmware V1.2.56), 1.0 (for older versions)

Maximum: 5.0

Description: Scaling factor of the sensor.

11.1.4.3 Set_BarycenterSca (CEB)

Set the scaling factor for barycenter values.

This is an internal command. It should not be used by the customer.

Parameter: double SP_BarycenterScale

SP_BarycenterScale

Direction: Down

Valid values:

Minimum: 32.0

Maximum: 32.0

Description: Scaling factor for Barycenter.

11.1.4.4 Get_BarycenterSca (CEB?)

Get the scaling factor for barycenter values.

This is an internal command. It should not be used by the customer.

Parameter: double SA_BarycenterScale

SA_BarycenterScale

Direction: Up

Valid values:

Minimum: 32.0

Maximum: 32.0

Description: Scaling factor for Barycenter.

11.1.4.5 Set_BarycenterOff (CRB)

Set the offset values for barycenter values.

This is an internal command. It should not be used by the customer.

Parameter: double SP_BarycenterOffset

SP_BarycenterOffset

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 1023.9

Description: Offset for Barycenter.

11.1.4.6 Get_BarycenterOff (CRB?)

Get the offset value for barycenter values.

This is an internal command. It should not be used by the customer.

Parameter: double SA_BarycenterOffset

SA_BarycenterOffset

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 1023.9

Description: Offset for Barycenter.

11.1.4.7 Upload_RefIdxFile

Send a refractive index file to the controller.

This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_RefraIndexFileIdx

SP_RefraIndexFileIdx

Direction: Down

Valid values:

Minimum: 1

Maximum: 8

Description: Index on which position the file should be stored.

11.1. Commands for IFD2401 and IFD2431

Parameter: String SP_RefraIndexFile SP_RefraIndexFile
Direction: Down
Valid values:
 2049 (at IFD2401) or 1025 (at IFD2431) lines
 First line is a name (limited to 19 characters), the other lines are refractive index values with precision four.
Description: Refractive index file.

11.1.4.8 Get_WhiteRef (SGW)

Get the white reference table table of controller.
 This is an internal command. It should not be used by the customer.

Parameter: Binary data SA_WhiteRef SA_WhiteRef
Direction: Up
Valid values:
 4096 (at IFD2401) or 2048 (at IFD2431) bytes, convertible to 2048 (at IFD2401) or 1024 (at IFD2431) words.
Description: White reference table.

11.1.4.9 Get_NormSig (SGN)

Get the norm signal table of controller.
 This is an internal command. It should not be used by the customer.

Parameter: Binary data SA_NormSig SA_NormSig
Direction: Up
Valid values:
 4096 (at IFD2401) or 2048 (at IFD2431) bytes, convertible to 2048 (at IFD2401) or 1024 (at IFD2431) words.
Description: Norm signal table.

11.1.4.10 Get_CalibTable (SGC)

Get the calibration table of controller.
 This is an internal command. It should not be used by the customer.

Parameter: Binary data SA_CalibTable SA_CalibTable
Direction: Up
Valid values:
 4096 (at IFD2401) or 2048 (at IFD2431) bytes, convertible to 2048 (at IFD2401) or 1024 (at IFD2431) words.
Description: Calibration table.

11.1.4.11 Upload_CalibTable

Send a calibration table for selected sensor to the controller.

This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_BlockSize SP_BlockSize

Direction: Down

Unit: Values

Valid values:

Minimum: 1

Maximum: 2147483647 (INT32_MAX)

Default: 64

Description: Size of blocks in which the calibration table is separated before sending.

Parameter: Binary data SP_CalibTable SP_CalibTable

Direction: Down

Valid values:

 2048 (at IFD2401) or 1024 (at IFD2431) values of datatype float (each 4 byte).

Description: Calibration table for selected sensor.

Parameter: double SP_FullRange SP_FullRange

Direction: Down

Unit: μm

Valid values:

Minimum: 1.17549e-38 (FLT_MIN)

Maximum: 3.40282e+38 (FLT_MAX)

Description: Measure range of selected sensor.

11.2 Commands for IFD2421 and IFD2422

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
- [IF2004](#) (native).
- [TCP/IP](#) (native).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_AllParameters](#) (SP_Additional= 1) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate, to interpret and scale data and to assign values. If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls optionally sensor command [Set_IPDataTransferMode](#) and [Set_DataOutInterface](#) at [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_AllParameters](#) (SP_Additional= 1)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

11.2.1 General commands

11.2.1.1 General

11.2.1.1.1 Get_Help (HELP)

Retrieve a help text from controller for a specific command.

Parameter: String SP_Command SP_Command
Direction: Down
Valid values:
 "" (empty string, means general help)
 or any command name
Description: Name of the command.

Parameter: String SA_HelpText SA_HelpText
Direction: Up
Description: Help text to the command.

11.2.1.1.2 Get_Info (GETINFO)

Retrieve information about the controller.

Parameter: String SA_Sensor SA_Sensor
Direction: Up
Description: Name of the controller.

Parameter: String SA_SerialNumber SA_SerialNumber
Direction: Up
Valid values:
 Numeric value
Description: Serial number of the controller.

Parameter: String SA_Option SA_Option
Direction: Up
Valid values:
 Numeric value
Description: Option of the controller.

Parameter: String SA_ArticleNumber SA_ArticleNumber
Direction: Up
Valid values:
 Numeric value
Description: Article number of the controller.

Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description: MAC address (low level ethernet address) of the controller.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the controller.	
Parameter: String SA_HardwareRevision	SA_HardwareRevision
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of the controller board.	
Parameter: String SA_BuildID	SA_BuildID
Direction: Up	
Description: Build ID	
Parameter: String SA_BuildTimestamp	SA_BuildTimestamp
Direction: Up	
Description: Build timestamp	
Parameter: String SA_BootVersion	SA_BootVersion
Direction: Up	
Description: Boot version	

11.2.1.3 Get_ChannelCount (GETCHANNELCNT)

Get number of channels of controller.

Parameter: int32_t SA_ChannelCount	SA_ChannelCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2	
Description: Number of channels.	

11.2.1.4 Get_OutputInfo_RS422 (GETOUTINFO_RS422)

Retrieve information which data is output at RS422 interface.

Parameter: int32_t SA_OutputVideoRaw_Ch1_RS422	SA_OutputVideoRaw_Ch1_- RS422
Direction: Up	
Valid values:	
0 = no	
1 = yes	
Description: Specify if raw video signal of first channel is transmitted.	

Parameter: int32_t SA_OutputVideoDark_Ch1_RS422	SA_OutputVideoDark_Ch1_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if dark corrected signal of first channel is transmitted.
Parameter: int32_t SA_OutputVideoLight_Ch1_RS422	SA_OutputVideoLight_Ch1_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if light corrected video signal of first channel is transmitted.
Parameter: int32_t SA_OutputVideoRaw_Ch2_RS422	SA_OutputVideoRaw_Ch2_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if raw video signal of second channel is transmitted.
Parameter: int32_t SA_OutputVideoDark_Ch2_RS422	SA_OutputVideoDark_Ch2_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if dark corrected signal of second channel is transmitted.
Parameter: int32_t SA_OutputVideoLight_Ch2_RS422	SA_OutputVideoLight_Ch2_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if light corrected video signal of second channel is transmitted.
Parameter: int32_t SA_OutputDistance1..6_Ch1_RS422	SA_OutputDistance1..6_- Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if distance 1 to 6 of first channel is transmitted.

Parameter: int32_t SA_OutputDistance1..6_Ch2_RS422	SA_OutputDistance1..6_- Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if distance 1 to 6 of second channel is transmitted.
Parameter: int32_t SA_OutputUnlinearizedDistance1..6_Ch1_RS422	SA_OutputUnlinearizedDis- tance1..6_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if unlinearized distance 1 to 6 of first channel is transmit- ted.
Parameter: int32_t SA_OutputUnlinearizedDistance1..6_Ch2_RS422	SA_OutputUnlinearizedDis- tance1..6_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if unlinearized distance 1 to 6 of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_RS422	SA_OutputAdditionalShut- terTime_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if shutter time of first channel is transmitted.
Parameter: int32_t SA_OutputAdditionalEncoder1_Ch1_RS422	SA_OutputAdditionalEn- coder1_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if encoder 1 of first channel is transmitted.
Parameter: int32_t SA_OutputAdditionalEncoder2_Ch1_RS422	SA_OutputAdditionalEn- coder2_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if encoder 2 of first channel is transmitted.
Parameter: int32_t SA_OutputAdditionalIntensity1..6_Ch1_RS422	SA_OutputAdditionalInten- sity1..6_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if intensity 1 to 6 of first channel is transmitted.

Parameter: int32_t SA_OutputAdditionalShutterTime_Ch2_RS422	SA_OutputAdditionalShutterTime_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if shutter time of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalEncoder1_Ch2_RS422	SA_OutputAdditionalEncoder1_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if encoder 1 of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalEncoder2_Ch2_RS422	SA_OutputAdditionalEncoder2_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if encoder 2 of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalIntensity1..6_Ch2_RS422	SA_OutputAdditionalIntensity1..6_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if intensity 1 to 6 of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalMeasrate_RS422	SA_OutputAdditionalMeasrate_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if measurement rate is transmitted.
Parameter: int32_t SA_OutputAdditionalTrgTimeDiff_RS422	SA_OutputAdditionalTrgTimeDiff_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if trigger time difference is transmitted.
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp is transmitted.

Parameter: int32_t SA_OutputAdditionalTimestampLo_RS422	SA_OutputAdditionalTimestampLo_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp (lower 16 bit) is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestampHi_RS422	SA_OutputAdditionalTimestampHi_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp (upper 16 bit) is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: String SA_OutputCalculation_RS422	SA_OutputCalculation_RS422
Direction: Up	
Description: List of calculated signals (generated by Set_Computation). The entries are separated by spaces.	
Parameter: String SA_OutputStatistic_RS422	SA_OutputStatistic_RS422
Direction: Up	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries are separated by spaces.	

11.2.1.1.5 Get_OutputInfo_ETH (GETOUTINFO_ETH)

Retrieve information which data is output at ethernet interface.

Parameter: int32_t SA_OutputVideoRaw_Ch1_ETH	SA_OutputVideoRaw_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoDark_Ch1_ETH	SA_OutputVideoDark_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if dark corrected signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoLight_Ch1_ETH	SA_OutputVideoLight_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if light corrected video signal of first channel is transmitted.	

Parameter: int32_t SA_OutputVideoRaw_Ch2_ETH	SA_OutputVideoRaw_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if raw video signal of second channel is transmitted.
Parameter: int32_t SA_OutputVideoDark_Ch2_ETH	SA_OutputVideoDark_Ch2_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if dark corrected signal of second channel is transmitted.
Parameter: int32_t SA_OutputVideoLight_Ch2_ETH	SA_OutputVideoLight_Ch2_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if light corrected video signal of second channel is transmitted.
Parameter: int32_t SA_OutputDistance1..6_Ch1_ETH	SA_OutputDistance1..6_- Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if distance 1 to 6 of first channel is transmitted.
Parameter: int32_t SA_OutputDistance1..6_Ch2_ETH	SA_OutputDistance1..6_- Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if distance 1 to 6 of second channel is transmitted.
Parameter: int32_t SA_OutputUnlinearizedDistance1..6_Ch1_ETH	SA_OutputUnlinearizedDis- tance1..6_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if unlinearized distance 1 to 6 of first channel is transmitted.

Parameter: int32_t SA_OutputUnlinearizedDistance1..6_Ch2_ETH	SA_OutputUnlinearizedDistance1..6_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if unlinearized distance 1 to 6 of second channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_ETH	SA_OutputAdditionalShutterTime_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1_Ch1_ETH	SA_OutputAdditionalEncoder1_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder2_Ch1_ETH	SA_OutputAdditionalEncoder2_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 2 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity1..6_Ch1_ETH	SA_OutputAdditionalIntensity1..6_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity 1 to 6 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch2_ETH	SA_OutputAdditionalShutterTime_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if shutter time of second channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1_Ch2_ETH	SA_OutputAdditionalEncoder1_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if encoder 1 of second channel is transmitted.	

Parameter: int32_t SA_OutputAdditionalEncoder2_Ch2_ETH	SA_OutputAdditionalEncoder2_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if encoder 2 of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalIntensity1..6_Ch2_ETH	SA_OutputAdditionalIntensity1..6_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if intensity 1 to 6 of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalMeasrate_ETH	SA_OutputAdditionalMeasrate_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if measurement rate is transmitted.
Parameter: int32_t SA_OutputAdditionalTrgTimeDiff_ETH	SA_OutputAdditionalTrgTimeDiff_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if trigger time difference is transmitted.
Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp is transmitted.
Parameter: int32_t SA_OutputAdditionalTimestampLo_ETH	SA_OutputAdditionalTimestampLo_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp (lower 16 bit) is transmitted.
Parameter: int32_t SA_OutputAdditionalTimestampHi_ETH	SA_OutputAdditionalTimestampHi_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp (upper 16 bit) is transmitted.

Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditionalCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: String SA_OutputCalculation_ETH	SA_OutputCalculation_ETH
Direction: Up	
Description: List of calculated signals (generated by Set_Computation). The entries are separated by spaces.	
Parameter: String SA_OutputStatistic_ETH	SA_OutputStatistic_ETH
Direction: Up	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries are separated by spaces.	

11.2.1.1.6 Set_Unit (UNIT)

Set the unit for configuration and display in the web diagram.

Parameter: int32_t SP_DisplayUnit	SP_DisplayUnit
Direction: Down	
Valid values:	
0= mm	
1= Inch	
Description: Unit.	

11.2.1.1.7 Get_Unit (UNIT)

Get the unit for configuration and display in the web diagram.

Parameter: int32_t SA_DisplayUnit	SA_DisplayUnit
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= mm	
1= Inch	
Description: Unit.	

11.2.1.1.8 Set_Echo (ECHO)

Set echo for sensor commands.

Parameter: int32_t SP_Echo	SP_Echo
Direction: Down	
Valid values:	
0= Off	
1= On	
Description: Echo mode.	

11.2.1.1.9 Get_Echo (ECHO)

Get the echo mode.

Parameter: int32_t SA_Echo

SA_Echo

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Off
- 1 = On

Description: Echo mode.

11.2.1.1.10 Get_AllParameters (PRINT)

Get all parameters from controller.

Parameter: int32_t SP_Additional

SP_Additional

Direction: Down

Valid values:

- 0 = No
- 1 = Yes

Description: If set, additional information about controller, sensor and material is output.

Parameter: int32_t SA_UserLevel

SA_UserLevel

Direction: Up

Valid values:

- 1 = Other user level (only for internal use)
- 0 = User
- 1 = Professional

Description: Actual user level.

Parameter: int32_t SA_DefaultUser

SA_DefaultUser

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = User
- 1 = Professional

Description: Default user level.

Parameter: int32_t SA_DisplayUnit

SA_DisplayUnit

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = mm
- 1 = Inch

Description: Unit.

Parameter: int32_t SA_ApplicationLanguage

SA_ApplicationLanguage

Direction: Up

Valid values:

- 0 = English (EN)
- 1 = German (DE)

Description: Language of web interface.

Parameter: int32_t SA_ChartType	SA_ChartType
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Measure (MEAS)	
1 = Video signal (VIDEO)	
Description: Type of the chart at web interface.	
Parameter: int32_t SA_Echo	SA_Echo
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Off	
1 = On	
Description: Echo mode.	
Parameter: double SA_Measrate	SA_Measrate
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 6.5	
Unit: kHz	
Description: Samplerate of measurement.	
Parameter: int32_t SA_EthernetMode	SA_EthernetMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Ethernet	
1 = Ethercat	
Description: Ethernet mode.	
Parameter: int32_t SA_DHCPEnabled	SA_DHCPEnabled
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = FALSE	
1 = TRUE	
Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).	
Parameter: String SA_Address	SA_Address
Direction: Up	
Valid values:	
Valid IP address in form of xxx.xxx.xxx.xxx	
Description: IP adress of the controller. If DHCP is enabled it returns the currently assigned IP address.	
Parameter: String SA_SubnetMask	SA_SubnetMask
Direction: Up	
Valid values:	
Valid network mask (e.g. 255.255.255.0 for a Class C network)	
Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.	

Parameter: String SA_Gateway SA_Gateway

Direction: Up

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

Parameter: int32_t SA_Protocol SA_Protocol

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= TCP server (SERVER/TCP)
- 1= TCP client (CLIENT/TCP)
- 2= UDP sender (CLIENT/UDP)
- 3= None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SA_RemoteAddress SA_RemoteAddress

Direction: Up

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to.

Parameter: int32_t SA_Port SA_Port

Direction: Up

Valid values:

- Minimum:** 1024
- Maximum:** 65535

Description: Port to send data to or to listen for incoming requests.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate

Direction: Up

Valid values:

- 4000000
- 3000000
- 2000000
- 921600
- 781250 (only available at Option 217)
- 691200
- 460800
- 230400
- 115200
- 38400 (since firmware V001.041.086)
- 9600

Unit: Baud

Description: Baudrate of controller.

Parameter: int32_t SA_KeyFunction1 SA_KeyFunction1

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= No function (NONE)
- 1= Dark correction, at IFD2422 the first channel is referenced (DARK-CORR)

- 2= The button triggers the command [Exec_Master](#) with action set for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERSET)
- 3= The button triggers the command [Exec_Master](#) with action reset for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERRESET)
- 4= The button is used as switch for the LED, at IFD2422 the first channel is referenced (LED)
- 5= Dark correction for channel 1, only available at IFD2422 (DARK-CORR_CH01)
- 6= Dark correction for channel 2, only available at IFD2422 (DARK-CORR_CH02)
- 7= The button is used as switch for the LED of channel 1, only available at IFD2422 (LED_CH01)
- 8= The button is used as switch for the LED of channel 2, only available at IFD2422 (LED_CH02)

Description: Button function

Parameter: int32_t SA_KeyFunction2

SA_KeyFunction2

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= No function (NONE)
- 1= Dark correction, at IFD2422 the first channel is referenced (DARK-CORR)
- 2= The button triggers the command [Exec_Master](#) with action set for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERSET)
- 3= The button triggers the command [Exec_Master](#) with action reset for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERRESET)
- 4= The button is used as switch for the LED, at IFD2422 the first channel is referenced (LED)
- 5= Dark correction for channel 1, only available at IFD2422 (DARK-CORR_CH01)
- 6= Dark correction for channel 2, only available at IFD2422 (DARK-CORR_CH02)
- 7= The button is used as switch for the LED of channel 1, only available at IFD2422 (LED_CH01)
- 8= The button is used as switch for the LED of channel 2, only available at IFD2422 (LED_CH02)

Description: Button function

Parameter: int32_t SA_MasterDistance1..6_Ch1

SA_MasterDistance1..6_Ch1

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: Specify if distance 1 to 6 of first channel is mastered.

Parameter: int32_t SA_MasterDistance1..6_Ch2

SA_MasterDistance1..6_Ch2

Direction: Up

Valid values:

- 0= no
- 1= yes

Valid for sensor:

IFD2422

Description: Specify if distance 1 to 6 of second channel is mastered.

Parameter: String SA_MasterSignalNames	SA_MasterSignalNames
Direction: Up	
Description: List of user defined signals (generated by Set_Computation or Set_StaticSignal). The entries are separated by spaces. The special value ALL specify to master all possible signals.	
Parameter: int32_t SA_Keylock	SA_Keylock
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Inactive (NONE)	
1 = Active	
2 = Automatic (AUTO)	
Description: Keylock.	
Parameter: int32_t SA_KeylockTime	SA_KeylockTime
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 60	
Unit: Minutes	
Description: Keylock time (only available at automatic keylock).	
Parameter: int32_t SA_KeylockState	SA_KeylockState
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Inactive (IS_INACTIVE)	
1 = Active (IS_ACTIVE)	
Description: Actual keylock state (only available at automatic keylock).	
Parameter: int32_t SA_OutputRS422	SA_OutputRS422
Direction: Up	
Valid values:	
0 = No	
1 = Yes	
Description: Output active at RS422 interface.	
Parameter: int32_t SA_OutputAnalog	SA_OutputAnalog
Direction: Up	
Valid values:	
0 = No	
1 = Yes	
Description: Output active at analog interface.	
Parameter: int32_t SA_OutputEthernet	SA_OutputEthernet
Direction: Up	
Valid values:	
0 = No	
1 = Yes	
Description: Output active at ethernet interface.	

Parameter: int32_t SA_OutputErrorOut	SA_OutputErrorOut
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at error outputs.	
Parameter: int32_t SA_ResampleRS422	SA_ResampleRS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: RS422 output is resampled.	
Parameter: int32_t SA_ResampleAnalog	SA_ResampleAnalog
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Analog output is resampled.	
Parameter: int32_t SA_ResampleEthernet	SA_ResampleEthernet
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Ethernet output is resampled.	
Parameter: int32_t SA_Resampling	SA_Resampling
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 3000000	
Description: Resampling value.	
Parameter: int32_t SA_MeasurePeak	SA_MeasurePeak
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= First peak at distance mode resp. first and last peak at thickness mode (F_L)	
1= Last peak at distance mode resp. last and next to last peak at thickness mode (L_SL)	
2= First peak at distance mode resp. first and second peak at thickness mode (F_S)	
3= Highest peak at distance mode resp. highest and second highest peak at thickness mode (H_SH)	
Description: Peaks to evalualte.	

Parameter: int32_t SA_MeasurePeak_Ch1..2	SA_MeasurePeak_Ch1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= First peak at distance mode resp. first and last peak at thickness mode (F_L)	
1= Last peak at distance mode resp. last and next to last peak at thickness mode (L_SL)	
2= First peak at distance mode resp. first and second peak at thickness mode (F_S)	
3= Highest peak at distance mode resp. highest and second highest peak at thickness mode (H_SH)	
Valid for sensor:	
IFD2422	
Description:	Peaks to evalualte.
Parameter: int32_t SA_SpikeCorrection	SA_SpikeCorrection
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Description:	Spike correction.
Parameter: int32_t SA_NbrEvaluatedValues	SA_NbrEvaluatedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description:	Number of values to evaluate for spike correction.
Parameter: double SA_ToleranceRange	SA_ToleranceRange
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 2147.0	
Unit:	mm
Description:	Tolerance range for spike correction.
Parameter: int32_t SA_NbrCorrectedValues	SA_NbrCorrectedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100	
Description:	Number of values to correct at spike correction.
Parameter: int32_t SA_SpikeCorrection_Ch1..2	SA_SpikeCorrection_Ch1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Valid for sensor:	
IFD2422	
Description:	Spike correction.

Parameter: int32_t SA_NbrEvaluatedValues_Ch1..2	SA_NbrEvaluatedValues_- Ch1..2
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Valid for sensor:	
IFD2422	
Description:	Number of values to evaluate for spike correction.
Parameter: double SA_ToleranceRange_Ch1..2	SA_ToleranceRange_Ch1..2
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 2147.0	
Unit: mm	
Valid for sensor:	
IFD2422	
Description:	Tolerance range for spike correction.
Parameter: int32_t SA_NbrCorrectedValues_Ch1..2	SA_NbrCorrectedValues_- Ch1..2
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100	
Valid for sensor:	
IFD2422	
Description:	Number of values to correct at spike correction.
Parameter: int32_t SA_TriggerSource	SA_TriggerSource
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = None	
1 = Sync/Trig	
2 = TrigIn	
3 = Software	
4 = Encoder1	
5 = Encoder2	
Description:	Trigger source (input).
Parameter: int32_t SA_TriggerInLogic	SA_TriggerInLogic
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = TTL	
1 = HTL	
Description:	Logic of trigger input
Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Edge	
1 = Level (PULSE)	
Description:	Trigger mode.

Parameter: int32_t SA_TriggerLevel	SA_TriggerLevel
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = High	
1 = Low	
Description: Trigger level.	
Parameter: int32_t SA_TriggerCount	SA_TriggerCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 16383	
Description: Number of values to measure. 0 means no trigger (NONE), 16383 means endless measurement (INFINITE).	
Parameter: int32_t SA_TriggerMoment	SA_TriggerMoment
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Input	
1 = Output	
Description: Trigger moment.	
Parameter: double SA_TriggerEncoderMin	SA_TriggerEncoderMin
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Lower encoder limit. Above this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: double SA_TriggerEncoderMax	SA_TriggerEncoderMax
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Upper encoder limit. Below this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: double SA_TriggerEncoderStepsize	SA_TriggerEncoderStepsize
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Number of encoder steps before trigger. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	

Parameter: int32_t SA_SyncMode SA_SyncMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Master
- 2= Slave (Sync/Trig)
- 3= Slave (TrigIn)

Description: Synchronization mode.

Parameter: int32_t SA_Termination SA_Termination

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Off
- 1= On

Description: Termination resistor

Parameter: int32_t SA_ErrorOutput1..2 SA_ErrorOutput1..2

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Channel 1 Intensity faults (01ER1)
- 1= Channel 1 Out of range values (01ER2)
- 2= 01ER1 or 01ER2 (01ER12)
- 3= Channel 2 Intensity faults, only available at IFD2422 (02ER1)
- 4= Channel 2 Out of range values, only available at IFD2422 (02ER2)
- 5= 02ER1 or 02ER2, only available at IFD2422 (02ER12)
- 6= 01ER1 or 01ER2 or 02ER1 or 02ER2, only available at IFD2422 (0102ER12)
- 7= Works as limit switch with the configuration of ERRORLIMIT1 (ER-RORLIMIT)
- 8= Falling below the intensity minimum channel 1, only available at Option 217 (01ER3)
- 9= Falling below the intensity minimum channel 2, only available at IFD2422 Option 217 (02ER3)
- 10= 01ER3 or 02ER3, only available at IFD2422 Option 217 (0102ER3)

Description: Condition for error output.

Parameter: int32_t SA_ErrorLimitSignalIndex1..2 SA_ErrorLimitSignalIndex1..2

Direction: Up

Valid values:

- 1= User defined name at SA_ErrorLimitSignalName1..2
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel
- 2= Distance 3 of first Channel
- 3= Distance 4 of first Channel
- 4= Distance 5 of first Channel
- 5= Distance 6 of first Channel
- 6= Distance 1 of second Channel (only available at IFD2422)
- 7= Distance 2 of second Channel (only available at IFD2422)
- 8= Distance 3 of second Channel (only available at IFD2422)
- 9= Distance 4 of second Channel (only available at IFD2422)
- 10= Distance 5 of second Channel (only available at IFD2422)
- 11= Distance 6 of second Channel (only available at IFD2422)

Description: Signal index to be checked.

Parameter: String SA_ErrorLimitSignalName1..2	SA_ErrorLimitSignalName1..2
Direction: Up	
Description: User defined signal which is used for error limit.	
Parameter: double SA_LowerLimit1..2	SA_LowerLimit1..2
Direction: Up	
Valid values:	
Minimum: -2147.0	
Maximum: 2147.0	
Unit: mm	
Description: Lower limit.	
Parameter: double SA_UpperLimit1..2	SA_UpperLimit1..2
Direction: Up	
Valid values:	
Minimum: -2147.0	
Maximum: 2147.0	
Unit: mm	
Description: Upper limit.	
Parameter: int32_t SA_ErrorLimitCompOp1..2	SA_ErrorLimitCompOp1..2
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Lower	
1 = Upper	
2 = Both	
Description: Compare operation	
Parameter: int32_t SA_ErrorLevelOut1..2	SA_ErrorLevelOut1..2
Direction: Up	
Valid values:	
0 = NPN	
1 = PNP	
2 = Push-Pull (PUSHPULL)	
3 = Push-Pull negated (PUSHPULLNEG)	
Description: Error level.	
Parameter: int32_t SA_EncoderInterpolation1..2	SA_EncoderInterpolation1..2
Direction: Up	
Valid values:	
1	
2	
4	
Description: Encoder interpolation.	
Parameter: int32_t SA_EncoderMode1..2	SA_EncoderMode1..2
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = No action (NONE)	
1 = Set encoder value to preset value only one time (ONE)	
2 = Set encoder value to preset value each time (EVER)	
Description: Mode of encoder when reference is reached.	

Parameter: double SA_EncoderPreload1..2	SA_EncoderPreload1..2
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: double SA_Encoder.MaxValue1..2	SA_Encoder.MaxValue1..2
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Maximum value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: int32_t SA_ROIStart	SA_ROIStart
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: First position on CCD.	
Parameter: int32_t SA_ROIEnd	SA_ROIEnd
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: Last position on CCD.	
Parameter: int32_t SA_ROIStart_Ch1..2	SA_ROIStart_Ch1..2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Valid for sensor:	
IFD2422	
Description: First position on CCD.	
Parameter: int32_t SA_ROIEnd_Ch1..2	SA_ROIEnd_Ch1..2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Valid for sensor:	
IFD2422	
Description: Last position on CCD.	

Parameter: double SA_MinimumThreshold	SA_MinimumThreshold
Direction: Up	
Valid values:	
Minimum: System minimum (usually 0.5)	
Maximum: 100.0	
Unit: %	
Description: A valid peak must be above this threshold.	
Parameter: double SA_MinimumThreshold_Ch1..2	SA_MinimumThreshold_- Ch1..2
Direction: Up	
Valid values:	
Minimum: System minimum (usually 0.5)	
Maximum: 100.0	
Unit: %	
Valid for sensor:	
IFD2422	
Description: A valid peak must be above this threshold.	
Parameter: double SA_ShutterTime1	SA_ShutterTime1
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 10000.0	
Unit: <i>μs</i>	
Description: First shutter time.	
Parameter: double SA_ShutterTime2	SA_ShutterTime2
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 10000.0	
Unit: <i>μs</i>	
Description: Second shutter time.	
Parameter: double SA_ShutterTime1_Ch1..2	SA_ShutterTime1_Ch1..2
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 10000.0	
Unit: <i>μs</i>	
Valid for sensor:	
IFD2422	
Description: First shutter time.	
Parameter: double SA_ShutterTime2_Ch1..2	SA_ShutterTime2_Ch1..2
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 10000.0	
Unit: <i>μs</i>	
Valid for sensor:	
IFD2422	
Description: Second shutter time.	

Parameter: int32_t SA_ShutterMode	SA_ShutterMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
1= Control exposure time, measuring rate is set manually (MEAS)	
2= Exposure time and measuring rate are set manually (MANUAL)	
3= Use two fixed exposure times alternately (2TIMEALT)	
4= Use the suitable exposure time of two fixed ones (2TIMES)	
Description: Shutter mode.	
Parameter: int32_t SA_ShutterMode_Ch1..2	SA_ShutterMode_Ch1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
1= Control exposure time, measuring rate is set manually (MEAS)	
2= Exposure time and measuring rate are set manually (MANUAL)	
3= Use two fixed exposure times alternately (2TIMEALT)	
4= Use the suitable exposure time of two fixed ones (2TIMES)	
Valid for sensor:	
IFD2422	
Description: Shutter mode.	
Parameter: double SA_PeakModulation	SA_PeakModulation
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 100.0	
Unit: %	
Description: Value how a peak is modulated.	
Parameter: double SA_PeakModulation_Ch1..2	SA_PeakModulation_Ch1..2
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 100.0	
Unit: %	
Valid for sensor:	
IFD2422	
Description: Value how a peak is modulated.	
Parameter: int32_t SA_LEDControl	SA_LEDControl
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: LED state.	
Parameter: int32_t SA_LEDControl_Ch1..2	SA_LEDControl_Ch1..2
Direction: Up	
Valid values:	
0= Off	
1= On	
Valid for sensor:	
IFD2422	
Description: LED state.	

Parameter: int32_t SA_ActiveSensor	SA_ActiveSensor
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of active sensor head.	
Parameter: int32_t SA_ActiveSensor_Ch1..2	SA_ActiveSensor_Ch1..2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Valid for sensor:	
IFD2422	
Description: Index of active sensor head.	
Parameter: int32_t SA_AnalogOutputSignalIndex	SA_AnalogOutputSignalIndex
Direction: Up	
Valid values:	
-1= User defined name at SA_AnalogOutputSignalName	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: Data to be used for analog output.	
Parameter: String SA_AnalogOutputSignalName	SA_AnalogOutputSignalName
Direction: Up	
Description: User defined signal which is used for analog output.	
Parameter: int32_t SA_AnalogRange	SA_AnalogRange
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
1= 0 - 5V	
2= 0 - 10V	
5= 4 - 20mA	
Description: Analog output range.	
Parameter: int32_t SA_AnalogScaleMode	SA_AnalogScaleMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Standard	
1= Two point (TWOPOINT)	
Description: Analog scale mode.	

Parameter: double SA_AnalogScaleRangeLowerLimit	SA_AnalogScaleRangeLowerLimit
Direction: Up	
Valid values:	
Minimum: -2147.0	
Maximum: 2147.0	
Unit: mm	
Description: Value which represents lowest voltage/current (at two point scaling).	
Parameter: double SA_AnalogScaleRangeUpperLimit	SA_AnalogScaleRangeUpperLimit
Direction: Up	
Valid values:	
Minimum: -2147.0	
Maximum: 2147.0	
Unit: mm	
Description: Value which represents highest voltage/current (at two point scaling).	
Parameter: int32_t SA_OutputVideoRaw_Ch1_RS422	SA_OutputVideoRaw_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoDark_Ch1_RS422	SA_OutputVideoDark_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if dark corrected signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoLight_Ch1_RS422	SA_OutputVideoLight_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if light corrected video signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoRaw_Ch2_RS422	SA_OutputVideoRaw_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if raw video signal of second channel is transmitted.	
Parameter: int32_t SA_OutputVideoDark_Ch2_RS422	SA_OutputVideoDark_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if dark corrected signal of second channel is transmitted.	

Parameter: int32_t SA_OutputVideoLight_Ch2_RS422	SA_OutputVideoLight_Ch2_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if light corrected video signal of second channel is transmitted.	
Parameter: int32_t SA_OutputDistance1..6_Ch1_RS422	SA_OutputDistance1..6_- Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 to 6 of first channel is transmitted.	
Parameter: int32_t SA_OutputDistance1..6_Ch2_RS422	SA_OutputDistance1..6_- Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if distance 1 to 6 of first channel is transmitted.	
Parameter: int32_t SA_OutputUnlinearizedDistance_Ch1_RS422	SA_OutputUnlinearizedDis- tance_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if unlinearized distance of first channel is transmitted.	
Parameter: int32_t SA_OutputUnlinearizedDistance1..6_Ch1_RS422	SA_OutputUnlinearizedDis- tance1..6_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if unlinearized distance 1 to 6 of first channel is transmitted.	
Parameter: int32_t SA_OutputUnlinearizedDistance_Ch2_RS422	SA_OutputUnlinearizedDis- tance_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if unlinearized distance of second channel is transmitted.	

Parameter: int32_t SA_OutputUnlinearizedDistance1..6_Ch2_RS422	SA_OutputUnlinearizedDistance1..6_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if unlinearized distance 1 to 6 of second channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_RS422	SA_OutputAdditionalShutterTime_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1_Ch1_RS422	SA_OutputAdditionalEncoder1_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder2_Ch1_RS422	SA_OutputAdditionalEncoder2_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 2 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_Ch1_RS422	SA_OutputAdditionalIntensity_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity1..6_Ch1_RS422	SA_OutputAdditionalIntensity1..6_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity 1 to 6 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch2_RS422	SA_OutputAdditionalShutterTime_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if shutter time of second channel is transmitted.	

Parameter: int32_t SA_OutputAdditionalEncoder1_Ch2_RS422	SA_OutputAdditionalEncoder1_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if encoder 1 of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalEncoder2_Ch2_RS422	SA_OutputAdditionalEncoder2_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if encoder 2 of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalIntensity_Ch2_RS422	SA_OutputAdditionalIntensity_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if intensity of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalIntensity1..6_Ch2_RS422	SA_OutputAdditionalIntensity1..6_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if intensity 1 to 6 of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalMeasrate_RS422	SA_OutputAdditionalMeasrate_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if measurement rate is transmitted.
Parameter: int32_t SA_OutputAdditionalTrgTimeDiff_RS422	SA_OutputAdditionalTrgTimeDiff_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if trigger time difference is transmitted.
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp is transmitted.

Parameter: int32_t SA_OutputAdditionalTimestampLo_RS422	SA_OutputAdditionalTimestampLo_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp (lower 16 bit) is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestampHi_RS422	SA_OutputAdditionalTimestampHi_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp (upper 16 bit) is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: String SA_OutputCalculation_RS422	SA_OutputCalculation_RS422
Direction: Up	
Description: List of calculated signals (generated by Set_Computation). The entries are separated by spaces.	
Parameter: String SA_OutputStatistic_RS422	SA_OutputStatistic_RS422
Direction: Up	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries are separated by spaces.	
Parameter: int32_t SA_OutputVideoRaw_Ch1_ETH	SA_OutputVideoRaw_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoDark_Ch1_ETH	SA_OutputVideoDark_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if dark corrected signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoLight_Ch1_ETH	SA_OutputVideoLight_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if light corrected video signal of first channel is transmitted.	

Parameter: int32_t SA_OutputVideoRaw_Ch2_ETH	SA_OutputVideoRaw_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if raw video signal of second channel is transmitted.
Parameter: int32_t SA_OutputVideoDark_Ch2_ETH	SA_OutputVideoDark_Ch2_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if dark corrected signal of second channel is transmitted.
Parameter: int32_t SA_OutputVideoLight_Ch2_ETH	SA_OutputVideoLight_Ch2_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if light corrected video signal of second channel is transmitted.
Parameter: int32_t SA_OutputDistance1..6_Ch1_ETH	SA_OutputDistance1..6_- Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if distance 1 to 6 of first channel is transmitted.
Parameter: int32_t SA_OutputDistance1..6_Ch2_ETH	SA_OutputDistance1..6_- Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if distance 1 to 6 of second channel is transmitted.
Parameter: int32_t SA_OutputUnlinearizedDistance_Ch1_ETH	SA_OutputUnlinearizedDis- tance_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if unlinearized distance of first channel is transmitted.

Parameter: int32_t SA_OutputUnlinearizedDistance1..6_Ch1_ETH	SA_OutputUnlinearizedDistance1..6_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if unlinearized distance 1 to 6 of first channel is transmitted.	
Parameter: int32_t SA_OutputUnlinearizedDistance_Ch2_ETH	SA_OutputUnlinearizedDistance_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if unlinearized distance of second channel is transmitted.	
Parameter: int32_t SA_OutputUnlinearizedDistance1..6_Ch2_ETH	SA_OutputUnlinearizedDistance1..6_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if unlinearized distance 1 to 6 of second channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_ETH	SA_OutputAdditionalShutterTime_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1_Ch1_ETH	SA_OutputAdditionalEncoder1_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder2_Ch1_ETH	SA_OutputAdditionalEncoder2_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 2 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_Ch1_ETH	SA_OutputAdditionalIntensity_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity of first channel is transmitted.	

Parameter: int32_t SA_OutputAdditionalIntensity1..6_Ch1_ETH	SA_OutputAdditionalIntensity1..6_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity 1 to 6 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch2_ETH	SA_OutputAdditionalShutterTime_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if shutter time of second channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1_Ch2_ETH	SA_OutputAdditionalEncoder1_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if encoder 1 of second channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder2_Ch2_ETH	SA_OutputAdditionalEncoder2_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if encoder 2 of second channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_Ch2_ETH	SA_OutputAdditionalIntensity_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if intensity of second channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity1..6_Ch2_ETH	SA_OutputAdditionalIntensity1..6_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if intensity 1 to 6 of second channel is transmitted.	

Parameter: int32_t SA_OutputAdditionalMeasrate_ETH	SA_OutputAdditionalMeasrate_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measurement rate is transmitted.	
Parameter: int32_t SA_OutputAdditionalTrgTimeDiff_ETH	SA_OutputAdditionalTrgTimeDiff_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger time difference is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestampLo_ETH	SA_OutputAdditionalTimestampLo_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp (lower 16 bit) is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestampHi_ETH	SA_OutputAdditionalTimestampHi_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp (upper 16 bit) is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditionalCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: String SA_OutputCalculation_ETH	SA_OutputCalculation_ETH
Direction: Up	
Description: List of calculated signals (generated by Set_Calculation). The entries are separated by spaces.	
Parameter: String SA_OutputStatistic_ETH	SA_OutputStatistic_ETH
Direction: Up	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries are separated by spaces.	

Parameter: int32_t SA_HoldLastValid	SA_HoldLastValid
Direction: Up	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	
Parameter: String SA_ActiveMaterial	SA_ActiveMaterial
Direction: Up	
Description: Name of material.	
Parameter: String SA_ActiveMaterial_Ch1..2	SA_ActiveMaterial_Ch1..2
Direction: Up	
Valid for sensor:	
IFD2422	
Description: Name of material.	
Parameter: String SA_MaterialMultiPeak12	SA_MaterialMultiPeak12
Direction: Up	
Description: Name of material between first and second peak.	
Parameter: String SA_MaterialMultiPeak23	SA_MaterialMultiPeak23
Direction: Up	
Description: Name of material between second and third peak.	
Parameter: String SA_MaterialMultiPeak34	SA_MaterialMultiPeak34
Direction: Up	
Description: Name of material between third and fourth peak.	
Parameter: String SA_MaterialMultiPeak45	SA_MaterialMultiPeak45
Direction: Up	
Description: Name of material between fourth and fifth peak.	
Parameter: String SA_MaterialMultiPeak56	SA_MaterialMultiPeak56
Direction: Up	
Description: Name of material between fifth and sixth peak.	
Parameter: String SA_MaterialMultiPeak12_Ch1..2	SA_MaterialMultiPeak12_-Ch1..2
Direction: Up	
Valid for sensor:	
IFD2422	
Description: Name of material between first and second peak.	
Parameter: String SA_MaterialMultiPeak23_Ch1..2	SA_MaterialMultiPeak23_-Ch1..2
Direction: Up	
Valid for sensor:	
IFD2422	
Description: Name of material between second and third peak.	
Parameter: String SA_MaterialMultiPeak34_Ch1..2	SA_MaterialMultiPeak34_-Ch1..2
Direction: Up	
Valid for sensor:	
IFD2422	
Description: Name of material between third and fourth peak.	

Parameter: String SA_MaterialMultiPeak45_Ch1..2	SA_MaterialMultiPeak45_- Ch1..2
Direction: Up	
Valid for sensor:	
IFD2422	
Description: Name of material between fourth and fifth peak.	
Parameter: String SA_MaterialMultiPeak56_Ch1..2	SA_MaterialMultiPeak56_- Ch1..2
Direction: Up	
Valid for sensor:	
IFD2422	
Description: Name of material between fifth and sixth peak.	
Parameter: int32_t SA_RefRACTiveCorrection	SA_RefRACTiveCorrection
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Description: Tells if refractive correction is be enabled.	
Parameter: int32_t SA_RefRACTiveCorrection_Ch1..2	SA_RefRACTiveCorrection_- Ch1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Valid for sensor:	
IFD2422	
Description: Tells if refractive correction is be enabled.	
Parameter: int32_t SA_NumberOfPeaks	SA_NumberOfPeaks
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2 (Standard) or 6 (Multipeak option)	
Description: Maximum number of peaks.	
Parameter: int32_t SA_NumberOfPeaks_Ch1..2	SA_NumberOfPeaks_Ch1..2
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2 (Standard) or 6 (Multipeak option)	
Valid for sensor:	
IFD2422	
Description: Maximum number of peaks.	
Parameter: int32_t SA_ChannelCount	SA_ChannelCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2	
Description: Number of channels.	

Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Description: Name of the controller.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the controller.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the controller.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the controller.	
Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description: MAC address (low level ethernet address) of the controller.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the controller.	
Parameter: String SA_HardwareRevision	SA_HardwareRevision
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of the controller board.	
Parameter: String SA_BuildID	SA_BuildID
Direction: Up	
Description: Build ID	
Parameter: String SA_BuildTimestamp	SA_BuildTimestamp
Direction: Up	
Description: Build timestamp	
Parameter: String SA_BootVersion	SA_BootVersion
Direction: Up	
Description: Boot version	
Parameter: int32_t SA_Pos	SA_Pos
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the active sensor head in the table.	

Parameter: String SA_Name	SA_Name
Direction: Up	
Description: Name of the active sensor head.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Description: Measurement range of the active sensor head.	
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description: Unit of the measurement range.	
Parameter: String SA_Serial	SA_Serial
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the active sensor head.	
Parameter: int32_t SA_Pos_Ch1..2	SA_Pos_Ch1..2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the active sensor head in the table.	
Parameter: String SA_Name_Ch1..2	SA_Name_Ch1..2
Direction: Up	
Description: Name of the active sensor head.	
Parameter: double SA_Range_Ch1..2	SA_Range_Ch1..2
Direction: Up	
Valid values:	
Minimum: 0.0	
Description: Measurement range of the active sensor head.	
Parameter: String SA_Unit_Ch1..2	SA_Unit_Ch1..2
Direction: Up	
Description: Unit of the measurement range.	
Parameter: String SA_Serial_Ch1..2	SA_Serial_Ch1..2
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the active sensor head.	
Parameter: String SA_ComputationTable	SA_ComputationTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	

Parameter: int32_t SA_ComputationEntries	SA_ComputationEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 30	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_ComputationEntry1, SA_ComputationEntry2, ...	
Parameter: String SA_ComputationEntry1..x	SA_ComputationEntry1..x
Direction: Up	
Description: Each computation as complete line.	
Parameter: int32_t SA_ComputationQueue1..x	SA_ComputationQueue1..x
Direction: Up	
Valid values:	
0= Channel 1	
1= Channel 2 (only available at IFD2422)	
2= System (only available at IFD2422)	
Description: Computation queue	
Parameter: int32_t SA_ComputationID1..x	SA_ComputationID1..x
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Index within computation queue	
Parameter: int32_t SA_ComputationOperation1..x	SA_ComputationOperation1..x
Direction: Up	
Valid values:	
0= None	
1= Median	
2= Moving average	
3= Recursive averaging	
4= Calculation	
5= Thickness	
6= Copy	
Description: Computation operation	
Parameter: int32_t SA_ComputationSignalIndex1..x	SA_ComputationSignalIndex1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_ComputationSignalName1..x	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: Signal index (if operation is averaging or copy).	

Parameter: String SA_ComputationSignalName1..x	SA_ComputationSignalName1..x
Direction: Up	
Description: User defined signal (if operation is averaging or copy).	
Parameter: int32_t SA_ComputationMovingCount1..x	SA_ComputationMovingCount1..x
Direction: Up	
Valid values:	
2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096	
Description: Number of value for the averaging window. This parameter is only used at moving average.	
Parameter: int32_t SA_ComputationRecursiveCount1..x	SA_ComputationRecursiveCount1..x
Direction: Up	
Valid values:	
Minimum: 2	
Maximum: 32768	
Description: Number of values for recursive averaging. This parameter is only used at recursive average.	
Parameter: int32_t SA_ComputationMedianCount1..x	SA_ComputationMedianCount1..x
Direction: Up	
Valid values:	
3, 5, 7, 9	
Description: Number of values to build median. This parameter is only used at median.	
Parameter: double SA_ComputationFactor1_1..x	SA_ComputationFactor1_1..x
Direction: Up	
Valid values:	
Minimum: -32767.0	
Maximum: 32767.0	
Description: Multiplication factor for first signal (if operation is calculation).	
Parameter: int32_t SA_ComputationSignalIndex1_1..x	SA_ComputationSignalIndex1_1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_ComputationSignalName1_1..x	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: First signal index (if operation is calculation or thickness).	
Parameter: String SA_ComputationSignalName1_1..x	SA_ComputationSignalName1_1..x
Direction: Up	
Description: User defined first signal (if operation is calculation or thickness).	

Parameter: double SA_ComputationFactor2_1..x	SA_ComputationFactor2_1..x
Direction: Up	
Valid values:	
Minimum: -32767.0	
Maximum: 32767.0	
Description: Multiplication factor for second signal (if operation is calculation).	
Parameter: int32_t SA_ComputationSignalIndex2_1..x	SA_ComputationSignalIndex2_1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_ComputationSignalName2_1..x	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: Second signal index (if operation is calculation or thickness).	
Parameter: String SA_ComputationSignalName2_1..x	SA_ComputationSignalName2_1..x
Direction: Up	
Description: User defined second signal (if operation is calculation or thickness).	
Parameter: double SA_ComputationOffset1..x	SA_ComputationOffset1..x
Direction: Up	
Valid values:	
Minimum: -2147.0	
Maximum: 2147.0	
Unit: mm	
Description: Additional offset (if operation is calculation).	
Parameter: String SA_ComputationName1..x	SA_ComputationName1..x
Direction: Up	
Description: Name of the newly created signal (if operation is calculation, thickness or copy).	
Parameter: String SA_MasterSignalTable	SA_MasterSignalTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_MasterSignalEntries	SA_MasterSignalEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MasterSignalEntry1, SA_MasterSignalEntry2, ...	

Parameter: String SA_MasterSignalEntry1..x	SA_MasterSignalEntry1..x
Direction: Up	
Description: Each master signal as complete line.	
Parameter: int32_t SA_MasterSignalIndex1..x	SA_MasterSignalIndex1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_MasterSignalName1..x	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: Index of each master signal.	
Parameter: String SA_MasterSignalName1..x	SA_MasterSignalName1..x
Direction: Up	
Description: Name of each master signal.	
Parameter: double SA_MasterValue1..x	SA_MasterValue1..x
Direction: Up	
Valid values:	
Minimum: -2147.0	
Maximum: 2147.0	
Unit: mm	
Description: Master value	
Parameter: String SA_StatisticSignalTable	SA_StatisticSignalTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_StatisticSignalEntries	SA_StatisticSignalEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_StatisticSignalEntry1, SA_StatisticSignalEntry2, ...	
Parameter: String SA_StatisticSignalEntry1..x	SA_StatisticSignalEntry1..x
Direction: Up	
Description: Each statistic signal as complete line.	

Parameter: int32_t SA_StatisticSignalIndex1..x	SA_StatisticSignalIndex1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_StatisticSignalName1..x	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: Index of each statistic signal.	
Parameter: String SA_StatisticSignalName1..x	SA_StatisticSignalName1..x
Direction: Up	
Description: Name of each statistic signal.	
Parameter: int32_t SA_StatisticDepth1..x	SA_StatisticDepth1..x
Direction: Up	
Valid values:	
2, 4, 8, ..., 8192, 16384	
16385= Infinite (endless statistic calculation)	
Description: Number of values (depth) for statistic calculation.	
Parameter: String SA_SensorTable	SA_SensorTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_SensorTableCount	SA_SensorTableCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 20	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Channel1, SA_Channel2, ...	
Parameter: int32_t SA_Channel1..x	SA_Channel1..x
Direction: Up	
Valid values:	
1	
Description: Sensor channel. Always 1	
Parameter: int32_t SA_Pos1..x	SA_Pos1..x
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the sensor head in the table.	

Parameter: String SA_Name1..x	SA_Name1..x
Direction: Up	
Description: Name of the sensor head in the table.	
Parameter: double SA_Range1..x	SA_Range1..x
Direction: Up	
Valid values:	
Minimum: 0.0	
Description: Measurement range of the sensor head in the table.	
Parameter: String SA_Unit1..x	SA_Unit1..x
Direction: Up	
Description: Unit of the measurement range.	
Parameter: String SA_Serial1..x	SA_Serial1..x
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor head in the table.	
Parameter: String SA_SensorTable_Ch1	SA_SensorTable_Ch1
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_SensorTableCount_Ch1	SA_SensorTableCount_Ch1
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 20	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Channel1_Ch1, SA_Channel2_Ch1, ...	
Parameter: int32_t SA_Channel1..x_Ch1	SA_Channel1..x_Ch1
Direction: Up	
Valid values:	
1	
Description: Sensor channel. Always 1	
Parameter: int32_t SA_Pos1..x_Ch1	SA_Pos1..x_Ch1
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the sensor head in the table.	
Parameter: String SA_Name1..x_Ch1	SA_Name1..x_Ch1
Direction: Up	
Description: Name of the sensor head in the table.	
Parameter: double SA_Range1..x_Ch1	SA_Range1..x_Ch1
Direction: Up	
Valid values:	
Minimum: 0.0	
Description: Measurement range of the sensor head in the table.	

Parameter: String SA_Unit1..x_Ch1	SA_Unit1..x_Ch1
Direction: Up	
Description: Unit of the measurement range.	
Parameter: String SA_Serial1..x_Ch1	SA_Serial1..x_Ch1
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor head in the table.	
Parameter: String SA_SensorTable_Ch2	SA_SensorTable_Ch2
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_SensorTableCount_Ch2	SA_SensorTableCount_Ch2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 20	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Channel1_Ch2, SA_Channel2_Ch2, ...	
Parameter: int32_t SA_Channel1..x_Ch2	SA_Channel1..x_Ch2
Direction: Up	
Valid values:	
2	
Description: Sensor channel. Always 2	
Parameter: int32_t SA_Pos1..x_Ch2	SA_Pos1..x_Ch2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the sensor head in the table.	
Parameter: String SA_Name1..x_Ch2	SA_Name1..x_Ch2
Direction: Up	
Description: Name of the sensor head in the table.	
Parameter: double SA_Range1..x_Ch2	SA_Range1..x_Ch2
Direction: Up	
Valid values:	
Minimum: 0.0	
Description: Measurement range of the sensor head in the table.	
Parameter: String SA_Unit1..x_Ch2	SA_Unit1..x_Ch2
Direction: Up	
Description: Unit of the measurement range.	
Parameter: String SA_Serial1..x_Ch2	SA_Serial1..x_Ch2
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor head in the table.	

Parameter: String SA_CurrentName	SA_CurrentName
Direction: Up	
Description: Name of current (active) setting.	
Parameter: String SA_SettingNames	SA_SettingNames
Direction: Up	
Description: List (separated by new line).	
Parameter: String SA_PresetNames	SA_PresetNames
Direction: Up	
Description: List (separated by new line).	
Parameter: int32_t SA_Automatic	SA_Automatic
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Automatic selection.	
Parameter: String SA_InitialName	SA_InitialName
Direction: Up	
Description: Name of setting.	
Parameter: int32_t SA_PresetMode	SA_PresetMode
Direction: Up	
Valid values:	
0= Static	
1= Balanced	
2= Dynamic	
3= None (if no preset setting is active)	
Description: Preset mode.	
Parameter: String SA_MaterialTable	SA_MaterialTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_MaterialTableCount	SA_MaterialTableCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 20	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Pos1, SA_Pos2, ...	
Parameter: int32_t SA_Pos1..x	SA_Pos1..x
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the material in the table.	
Parameter: String SA_MaterialName1..x	SA_MaterialName1..x
Direction: Up	
Description: Name of the material in the table.	

Parameter: double SA_RefractiveIndex_nF1..x	SA_RefractiveIndex_nF1..x
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 486 nm.	
Parameter: double SA_RefractiveIndex_nd1..x	SA_RefractiveIndex_nd1..x
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 587 nm.	
Parameter: double SA_RefractiveIndex_nC1..x	SA_RefractiveIndex_nC1..x
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 656 nm.	
Parameter: double SA_AbbeNumber_vd1..x	SA_AbbeNumber_vd1..x
Direction: Up	
Valid values:	
Minimum: 10.0	
Maximum: 100.0	
Description: Abbe number of the material or 0.0 if not specified.	
Parameter: String SA_Description1..x	SA_Description1..x
Direction: Up	
Description: Description of the material in the table.	
Parameter: String SA_MaterialName	SA_MaterialName
Direction: Up	
Description: Name of the active material.	
Parameter: String SA_Description	SA_Description
Direction: Up	
Description: Description of the active material.	
Parameter: double SA_RefractiveIndex_nF	SA_RefractiveIndex_nF
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 486 nm.	
Parameter: double SA_RefractiveIndex_nd	SA_RefractiveIndex_nd
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 587 nm.	

Parameter: double SA_RefractiveIndex_nC	SA_RefractiveIndex_nC
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 656 nm.	
Parameter: double SA_AbbeNumber_vd	SA_AbbeNumber_vd
Direction: Up	
Valid values:	
Minimum: 10.0	
Maximum: 100.0	
Description: Abbe number of the active material or 0.0 if not specified.	
Parameter: int32_t SA_Described_by	SA_Described_by
Direction: Up	
Valid values:	
0= nF, nd and nC	
1= nd and abbe value	
Description: Tells if nF, nd and nC or if nd and abbe value is valid.	
Parameter: String SA_MaterialName_Ch1..2	SA_MaterialName_Ch1..2
Direction: Up	
Description: Name of the active material.	
Parameter: String SA_Description_Ch1..2	SA_Description_Ch1..2
Direction: Up	
Description: Description of the active material.	
Parameter: double SA_RefractiveIndex_nF_Ch1..2	SA_RefractiveIndex_nF_- Ch1..2
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 486 nm.	
Parameter: double SA_RefractiveIndex_nd_Ch1..2	SA_RefractiveIndex_nd_- Ch1..2
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 587 nm.	
Parameter: double SA_RefractiveIndex_nC_Ch1..2	SA_RefractiveIndex_nC_- Ch1..2
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 656 nm.	
Parameter: double SA_AbbeNumber_vd_Ch1..2	SA_AbbeNumber_vd_Ch1..2
Direction: Up	
Valid values:	
Minimum: 10.0	
Maximum: 100.0	
Description: Abbe number of the active material or 0.0 if not specified.	

Parameter: int32_t SA_Described_by_Ch1..2 SA_Described_by_Ch1..2

Direction: Up

Valid values:

0= nF, nd and nC

1= nd and abbe value

Description: Tells if nF, nd and nC or if nd and abbe value is valid.

11.2.1.1.11 Set_SyncMode (SYNC)

Set the synchronization mode.

Parameter: int32_t SP_SyncMode SP_SyncMode

Direction: Down

Valid values:

0= None

1= Master

2= Slave (Sync/Trig)

3= Slave (TrigIn)

Description: Synchronization mode.

11.2.1.1.12 Get_SyncMode (SYNC)

Get the synchronization mode.

Parameter: int32_t SA_SyncMode SA_SyncMode

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= None

1= Master

2= Slave (Sync/Trig)

3= Slave (TrigIn)

Description: Synchronization mode.

11.2.1.1.13 Set_Termination (TERMINATION)

Set connection of a termination resistor in sync line to prevent reflections.

Parameter: int32_t SP_Termination SP_Termination

Direction: Down

Valid values:

0= Off

1= On

Description: Termination resistor

11.2.1.1.14 Get_Termination (TERMINATION)

Get connection of a termination resistor in sync line to prevent reflections.

Parameter: int32_t SA_Termination SA_Termination
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = Off
 1 = On
Description: Termination resistor

11.2.1.1.15 Reset_Boot (RESET)

Resets the sensor.

At this command the controller may change output data after reboot. If first bit of [IP_AutomaticMode](#) is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

11.2.1.1.16 Reset_Counter (RESETCNT)

Resets sensor counter values.

Parameter: int32_t SP_ResetTimestamp SP_ResetTimestamp
Direction: Down
Valid values:
 0 = No
 1 = Yes
Description: Reset timestamp value.

Parameter: int32_t SP_ResetMeasCounter SP_ResetMeasCounter
Direction: Down
Valid values:
 0 = No
 1 = Yes
Description: Reset counter value.

11.2.1.1.17 Set_Keylock (KEYLOCK)

Set key lock for sensor.

Parameter: int32_t SP_Keylock SP_Keylock
Direction: Down
Valid values:
 0 = Inactive (NONE)
 1 = Active
 2 = Automatic (AUTO)
Description: Keylock.

Parameter: int32_t SP_KeylockTime SP_KeylockTime
Direction: Down
Valid values:
 Minimum: 1
 Maximum: 60
Unit: Minutes
Description: Keylock time (only used at automatic keylock).

11.2.1.1.18 Get_Keylock (KEYLOCK)

Get key lock for sensor.

Parameter: int32_t SA_Keylock SA_Keylock
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = Inactive (NONE)
 1 = Active
 2 = Automatic (AUTO)
Description: Keylock.

Parameter: int32_t SA_KeylockTime SA_KeylockTime
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 60
Unit: Minutes
Description: Keylock time (only available at automatic keylock).

Parameter: int32_t SA_KeylockState SA_KeylockState
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = Inactive (IS_INACTIVE)
 1 = Active (IS_ACTIVE)
Description: Actual keylock state (only available at automatic keylock).

11.2.1.1.19 Set_KeyFunction1 (KEYFUNC1)

Set the button functionality for timeslot 1 (0...2s)

Parameter: int32_t SP_KeyFunction1 SP_KeyFunction1
Direction: Down
Valid values:
 0 = No function (NONE)
 1 = Dark correction, at IFD2422 the first channel is referenced (DARK-CORR)
 2 = The button triggers the command [Exec_Master](#) with action set for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERSET)
 3 = The button triggers the command [Exec_Master](#) with action reset for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERRESET)

- 4= The button is used as switch for the LED, at IFD2422 the first channel is referenced (LED)
- 5= Dark correction for channel 1, only available at IFD2422 (DARK-CORR_CH01)
- 6= Dark correction for channel 2, only available at IFD2422 (DARK-CORR_CH02)
- 7= The button is used as switch for the LED of channel 1, only available at IFD2422 (LED_CH01)
- 8= The button is used as switch for the LED of channel 2, only available at IFD2422 (LED_CH02)

Description: Button function

11.2.1.1.20 Get_KeyFunction1 (KEYFUNC1)

Get the button functionality for timeslot 1 (0...2s)

Parameter: int32_t SA_KeyFunction1

SA_KeyFunction1

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= No function (NONE)
- 1= Dark correction, at IFD2422 the first channel is referenced (DARK-CORR)
- 2= The button triggers the command [Exec_Master](#) with action set for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERSET)
- 3= The button triggers the command [Exec_Master](#) with action reset for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERRESET)
- 4= The button is used as switch for the LED, at IFD2422 the first channel is referenced (LED)
- 5= Dark correction for channel 1, only available at IFD2422 (DARK-CORR_CH01)
- 6= Dark correction for channel 2, only available at IFD2422 (DARK-CORR_CH02)
- 7= The button is used as switch for the LED of channel 1, only available at IFD2422 (LED_CH01)
- 8= The button is used as switch for the LED of channel 2, only available at IFD2422 (LED_CH02)

Description: Button function

11.2.1.1.21 Set_KeyFunction2 (KEYFUNC2)

Set the button functionality for timeslot 2 (2...5s)

Parameter: int32_t SP_KeyFunction2

SP_KeyFunction2

Direction: Down

Valid values:

- 0= No function (NONE)
- 1= Dark correction, at IFD2422 the first channel is referenced (DARK-CORR)
- 2= The button triggers the command [Exec_Master](#) with action set for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERSET)

- 3= The button triggers the command [Exec_Master](#) with action reset for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERRESET)
- 4= The button is used as switch for the LED, at IFD2422 the first channel is referenced (LED)
- 5= Dark correction for channel 1, only available at IFD2422 (DARK-CORR_CH01)
- 6= Dark correction for channel 2, only available at IFD2422 (DARK-CORR_CH02)
- 7= The button is used as switch for the LED of channel 1, only available at IFD2422 (LED_CH01)
- 8= The button is used as switch for the LED of channel 2, only available at IFD2422 (LED_CH02)

Description: Button function

11.2.1.1.22 [Get_KeyFunction2](#) (KEYFUNC2)

Get the button functionality for timeslot 2 (2...5s)

Parameter: int32_t SA_KeyFunction2

SA_KeyFunction2

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= No function (NONE)
- 1= Dark correction, at IFD2422 the first channel is referenced (DARK-CORR)
- 2= The button triggers the command [Exec_Master](#) with action set for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERSET)
- 3= The button triggers the command [Exec_Master](#) with action reset for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERRESET)
- 4= The button is used as switch for the LED, at IFD2422 the first channel is referenced (LED)
- 5= Dark correction for channel 1, only available at IFD2422 (DARK-CORR_CH01)
- 6= Dark correction for channel 2, only available at IFD2422 (DARK-CORR_CH02)
- 7= The button is used as switch for the LED of channel 1, only available at IFD2422 (LED_CH01)
- 8= The button is used as switch for the LED of channel 2, only available at IFD2422 (LED_CH02)

Description: Button function

11.2.1.1.23 [Set_KeyMasterSignalSelect](#) (KEYMASTERSIGNALSELECT)

Set selection of the measurement data signal(s) for mastering with key.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_MasterDistance1..6_Ch1

SP_MasterDistance1..6_Ch1

Direction: Down

Valid values:

- 0= no
- 1= yes

Description: Specify if distance 1 to 6 of first channel is mastered.

Parameter: int32_t SP_MasterDistance`1..6_Ch2` SP_MasterDistance`1..6_Ch2`

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

IFD2422

Description: Specify if distance 1 to 6 of second channel is mastered.

Parameter: String SP_MasterSignalNames SP_MasterSignalNames

Direction: Down

Description: List of user defined signals (generated by [Set_Computation](#) or [Set_StatisticSignal](#)). The entries must be separated by spaces.

The special value ALL specify to master all possible signals. It must not combined with any other parameter.

11.2.1.1.24 Get_KeyMasterSignalSelect (KEYMASTERSIGNALSELECT)

Get selection of the measurement data signal(s) for mastering with key.

Parameter: int32_t SA_MasterDistance`1..6_Ch1` SA_MasterDistance`1..6_Ch1`

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if distance 1 to 6 of first channel is mastered.

Parameter: int32_t SA_MasterDistance`1..6_Ch2` SA_MasterDistance`1..6_Ch2`

Direction: Up

Valid values:

0= no

1= yes

Valid for sensor:

IFD2422

Description: Specify if distance 1 to 6 of second channel is mastered.

Parameter: String SA_MasterSignalNames SA_MasterSignalNames

Direction: Up

Description: List of user defined signals (generated by [Set_Computation](#) or [Set_StatisticSignal](#)). The entries are separated by spaces.

The special value ALL specify to master all possible signals.

11.2.1.2 User level

11.2.1.2.1 Logout (LOGOUT)

Change user level to user.

11.2.1.2.2 Login (LOGIN)

Change user level to professional.

Parameter: String SP_Password SP_Password
Direction: Down
Description: Valid password to login.

11.2.1.2.3 Get_UserLevel (GETUSERLEVEL)

Retrieve actual user level.

Parameter: int32_t SA_UserLevel SA_UserLevel
Direction: Up
Valid values:
 -1 = Other user level (only for internal use)
 0 = User
 1 = Professional
Description: Actual user level.

11.2.1.2.4 Set_DefaultUser (STDUSER)

Set the default user level after booting the system.

Parameter: int32_t SP_DefaultUser SP_DefaultUser
Direction: Down
Valid values:
 0 = User
 1 = Professional
Description: Default user level.

11.2.1.2.5 Get_DefaultUser (STDUSER)

Get the default user level after booting the system.

Parameter: int32_t SA_DefaultUser SA_DefaultUser
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = User
 1 = Professional
Description: Default user level.

11.2.1.2.6 Set_Password (PASSWD)

Change the password for login.

Parameter: String SP_OldPassword SP_OldPassword
Direction: Down
Description: Old password.

Parameter: String SP_NewPassword SP_NewPassword
Direction: Down
Description: New password.

11.2.1.3 Sensor

11.2.1.3.1 Get_SensorTable (SENSORTABLE)

Get a list of all calibrated sensors.
 At IFD2422 the first channel is referenced.

Parameter: String SA_SensorTable SA_SensorTable
Direction: Up
Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_SensorTableCount SA_SensorTableCount
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 20
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Channel1, SA_Channel2, ...

Parameter: int32_t SA_Channel1..x SA_Channel1..x
Direction: Up
Valid values:
 1
Description: Sensor channel. Always 1

Parameter: int32_t SA_Pos1..x SA_Pos1..x
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 19
Description: Index of the sensor head in the table.

Parameter: String SA_Name1..x SA_Name1..x
Direction: Up
Description: Name of the sensor head in the table.

Parameter: double SA_Range1..x SA_Range1..x
Direction: Up
Valid values:
 Minimum: 0.0
Description: Measurement range of the sensor head in the table.

Parameter: String SA_Unit1..x SA_Unit1..x
Direction: Up
Description: Unit of the measurement range.

Parameter: String SA_Serial1..x SA_Serial1..x
Direction: Up
Valid values:
 Numeric value
Description: Serial number of the sensor head in the table.

11.2.1.3.2 Get_SensorTable_Ch<n> (SENSORTABLE_CH0<n>)

Get a list of all calibrated sensors for channel n $\in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: String SA_SensorTable_Ch<n>

SA_SensorTable_Ch<n>

Direction: Up

Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_SensorTableCount_Ch<n>

SA_SensorTableCount_Ch<n>

Direction: Up

Valid values:

Minimum: 0

Maximum: 20

Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Channel1_Ch1, SA_Channel2_Ch1, ...

Parameter: int32_t SA_Channel1..x_Ch<n>

SA_Channel1..x_Ch<n>

Direction: Up

Valid values:

1

2

Description: Sensor channel.

Parameter: int32_t SA_Pos1..x_Ch<n>

SA_Pos1..x_Ch<n>

Direction: Up

Valid values:

Minimum: 0

Maximum: 19

Description: Index of the sensor head in the table.

Parameter: String SA_Name1..x_Ch<n>

SA_Name1..x_Ch<n>

Direction: Up

Description: Name of the sensor head in the table.

Parameter: double SA_Range1..x_Ch<n>

SA_Range1..x_Ch<n>

Direction: Up

Valid values:

Minimum: 0.0

Description: Measurement range of the sensor head in the table.

Parameter: String SA_Unit1..x_Ch<n>

SA_Unit1..x_Ch<n>

Direction: Up

Description: Unit of the measurement range.

Parameter: String SA_Serial1..x_Ch<n>

SA_Serial1..x_Ch<n>

Direction: Up

Valid values:

Numeric value

Description: Serial number of the sensor head in the table.

11.2.1.3.3 Set_ActiveSensor (SENSORHEAD)

Change to another sensor head.

At IFD2422 the first channel is referenced.

Parameter: int32_t SP_ActiveSensor

SP_ActiveSensor

Direction: Down

Valid values:

Minimum: 0

Maximum: 19

Description: Index of new sensor head.

11.2.1.3.4 Get_ActiveSensor (SENSORHEAD)

Get active sensor head.

At IFD2422 the first channel is referenced.

Parameter: int32_t SA_ActiveSensor

SA_ActiveSensor

Direction: Up

Valid values:

Minimum: 0

Maximum: 19

Description: Index of active sensor head.

11.2.1.3.5 Set_ActiveSensor_Ch<n> (SENSORHEAD_CH0<n>)

Change to another sensor head for channel n $\in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: int32_t SP_ActiveSensor_Ch<n>

SP_ActiveSensor_Ch<n>

Direction: Down

Valid values:

Minimum: 0

Maximum: 19

Description: Index of new sensor head.

11.2.1.3.6 Get_ActiveSensor_Ch<n> (SENSORHEAD_CH0<n>)

Get active sensor head of channel n $\in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: int32_t SA_ActiveSensor_Ch<n>

SA_ActiveSensor_Ch<n>

Direction: Up

Valid values:

Minimum: 0

Maximum: 19

Description: Index of active sensor head.

11.2.1.3.7 Get_SensorInfo (SENSORINFO)

Get information of active sensor head.
At IFD2422 the first channel is referenced.

Parameter: int32_t SA_Pos	SA_Pos
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description:	Index of the active sensor head in the table.
Parameter: String SA_Name	SA_Name
Direction: Up	
Description:	Name of the active sensor head.
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Description:	Measurement range of the active sensor head.
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description:	Unit of the measurement range.
Parameter: String SA_Serial	SA_Serial
Direction: Up	
Valid values:	
Numeric value	
Description:	Serial number of the active sensor head.

11.2.1.3.8 Get_SensorInfo_Ch<n> (SENSORINFO_CH0<n>)

Get information of active sensor head for channel n $\in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: int32_t SA_Pos_Ch<n>	SA_Pos_Ch<n>
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description:	Index of the active sensor head in the table.
Parameter: String SA_Name_Ch<n>	SA_Name_Ch<n>
Direction: Up	
Description:	Name of the active sensor head.
Parameter: double SA_Range_Ch<n>	SA_Range_Ch<n>
Direction: Up	
Valid values:	
Minimum: 0.0	
Description:	Measurement range of the active sensor head.

Parameter: String SA_Unit_Ch<n> SA_Unit_Ch<n>

Direction: Up

Description: Unit of the measurement range.

Parameter: String SA_Serial_Ch<n> SA_Serial_Ch<n>

Direction: Up

Valid values:

Numeric value

Description: Serial number of the active sensor head.

11.2.1.3.9 Set_LEDControl (LED)

Switches the white light LED ON or OFF.

At IFD2422 the first channel is referenced.

Parameter: int32_t SP_LEDControl SP_LEDControl

Direction: Down

Valid values:

0 = Off

1 = On

Description: LED state

11.2.1.3.10 Get_LEDControl (LED)

Get state of white light LED.

At IFD2422 the first channel is referenced.

Parameter: int32_t SA_LEDControl SA_LEDControl

Direction: Up

Valid values:

0 = Off

1 = On

Description: LED state

11.2.1.3.11 Set_LEDControl_Ch<n> (LED_CH0<n>)

Switches the white light LED ON or OFF for channel n ∈ {1..2}.

Valid for sensor:

IFD2422

Parameter: int32_t SP_LEDControl_Ch<n> SP_LEDControl_Ch<n>

Direction: Down

Valid values:

0 = Off

1 = On

Description: LED state

11.2.1.3.12 Get_LEDControl_Ch<n> (LED_CH0<n>)

Get state of white light LED of channel n ∈ {1..2}.

Valid for sensor:

IFD2422

Parameter: int32_t SA_LEDControl_Ch<n>

SA_LEDControl_Ch<n>

Direction: Up

Valid values:

0= Off

1= On

Description: LED state

11.2.1.3.13 DarkCorr (DARKCORR)

Make a dark correction.

At IFD2422 the first channel is referenced.

11.2.1.3.14 DarkCorr_Ch<n> (DARKCORR_CH0<n>)

Make a dark correction for channel n ∈ {1..2}.

Valid for sensor:

IFD2422

11.2.1.4 Triggering

11.2.1.4.1 Set_TriggerSource (TRIGGERSOURCE)

Set the trigger source (input).

Parameter: int32_t SP_TriggerSource

SP_TriggerSource

Direction: Down

Valid values:

0= None

1= Sync/Trig

2= TrigIn

3= Software

4= Encoder1

5= Encoder2

Description: Trigger source (input).

11.2.1.4.2 Get_TriggerSource (TRIGGERSOURCE)

Get the active trigger source (input).

Parameter: int32_t SA_TriggerSource

SA_TriggerSource

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Sync/Trig
- 2= TrigIn
- 3= Software
- 4= Encoder1
- 5= Encoder2

Description: Trigger source (input).

11.2.1.4.3 Set_TriggerMode (TRIGGERMODE)

Set the trigger mode.

Parameter: int32_t SP_TriggerMode

SP_TriggerMode

Direction: Down

Valid values:

- 0= Edge
- 1= Level (PULSE)

Description: Trigger mode.

11.2.1.4.4 Get_TriggerMode (TRIGGERMODE)

Get the active trigger mode.

Parameter: int32_t SA_TriggerMode

SA_TriggerMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Edge
- 1= Level (PULSE)

Description: Trigger mode.

11.2.1.4.5 Set_TriggerMoment (TRIGGERAT)

Set the trigger time.

Parameter: int32_t SP_TriggerMoment

SP_TriggerMoment

Direction: Down

Valid values:

- 0= Input
- 1= Output

Description: Trigger moment.

11.2.1.4.6 Get_TriggerMoment (TRIGGERAT)

Get the active trigger time.

Parameter: int32_t SA_TriggerMoment

SA_TriggerMoment

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Input
- 1 = Output

Description: Trigger moment.

11.2.1.4.7 Set_TriggerInLogic (TRIGINLEVEL)

Set the logic of trigger input TrigIn.

Parameter: int32_t SP_TriggerInLogic

SP_TriggerInLogic

Direction: Down

Valid values:

- 0 = TTL
- 1 = HTL

Description: Logic of trigger input.

11.2.1.4.8 Get_TriggerInLogic (TRIGINLEVEL)

Get the logic of trigger input TrigIn.

Parameter: int32_t SA_TriggerInLogic

SA_TriggerInLogic

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = TTL
- 1 = HTL

Description: Logic of trigger input

11.2.1.4.9 Set_TriggerLevel (TRIGGERLEVEL)

Set the trigger level.

Parameter: int32_t SP_TriggerLevel

SP_TriggerLevel

Direction: Down

Valid values:

- 0 = High
- 1 = Low

Description: Trigger level.

11.2.1.4.10 Get_TriggerLevel (TRIGGERLEVEL)

Get the active trigger level.

Parameter: int32_t SA_TriggerLevel SA_TriggerLevel
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = High
 1 = Low
Description: Trigger level.

11.2.1.4.11 Set_TriggerCount (TRIGGERCOUNT)

Set the number of values to measure at trigger.

Parameter: int32_t SP_TriggerCount SP_TriggerCount
Direction: Down
Valid values:
Minimum: 0
Maximum: 16383
Description: Number of values to measure. 0 means no trigger (NONE),
 16383 means endless measurement (INFINITE).

11.2.1.4.12 Get_TriggerCount (TRIGGERCOUNT)

Get the number of values to measure at trigger.

Parameter: int32_t SA_TriggerCount SA_TriggerCount
Direction: Up
Valid values:
Minimum: 0
Maximum: 16383
Description: Number of values to measure. 0 means no trigger (NONE),
 16383 means endless measurement (INFINITE).

11.2.1.4.13 Software_Trigger (TRIGGERSW)

Execute a software trigger.

11.2.1.4.14 Set_TriggerEncoderMin (TRIGGERENCECMIN)

Set the minimum encoder value for triggering.

Parameter: double SP_TriggerEncoderMin SP_TriggerEncoderMin
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 4294967295.0 (UINT32_MAX)
Description: Lower encoder limit. Above this value it will be triggered. This
 value is in whole numbers. But to avoid (signed) integer overflow, it's
 type is double.

11.2.1.4.15 Get_TriggerEncoderMin (TRIGGERENCMIN)

Get the minimum encoder value for triggering.

Parameter: double SA_TriggerEncoderMin

SA_TriggerEncoderMin

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Lower encoder limit. Above this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.2.1.4.16 Set_TriggerEncoderMax (TRIGGERENCMAX)

Set the maximum encoder value for triggering.

Parameter: double SP_TriggerEncoderMax

SP_TriggerEncoderMax

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Upper encoder limit. Below this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.2.1.4.17 Get_TriggerEncoderMax (TRIGGERENCMAX)

Get the maximum encoder value for triggering.

Parameter: double SA_TriggerEncoderMax

SA_TriggerEncoderMax

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Upper encoder limit. Below this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.2.1.4.18 Set_TriggerEncoderStepsize (TRIGGERENCSTEP SIZE)

Set the number of steps between triggering.

Parameter: double SP_TriggerEncoderStepsize

SP_TriggerEncoderStepsize

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Number of encoder steps before trigger. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.2.1.4.19 Get_TriggerEncoderStepsize (TRIGGERENCSTEPSENSE)

Get the number of steps between triggering.

Parameter: double SA_TriggerEncoderStepsize

SA_TriggerEncoderStepsize

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Number of encoder steps before trigger. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.2.1.5 Encoder

11.2.1.5.1 Set_EncoderInterpolation<n> (ENCINTERPOL<n>)

Set the interpolation for encoder $n \in \{1..2\}$.

Parameter: int32_t SP_EncoderInterpolation<n>

SP_EncoderInterpolation<n>

Direction: Down

Valid values:

1

2

4

Description: Encoder interpolation.

11.2.1.5.2 Get_EncoderInterpolation<n> (ENCINTERPOL<n>)

Get the interpolation of encoder $n \in \{1..2\}$.

Parameter: int32_t SA_EncoderInterpolation<n>

SA_EncoderInterpolation<n>

Direction: Up

Valid values:

1

2

4

Description: Encoder interpolation.

11.2.1.5.3 Set_EncoderMode<n> (ENCREF<n>)

Set the behaviour of encoder $n \in \{1..2\}$ when reference is reached.

Parameter: int32_t SP_EncoderMode<n>

SP_EncoderMode<n>

Direction: Down

Valid values:

0= No action (NONE)

1= Set encoder value to preset value only one time (ONE)

2= Set encoder value to preset value each time (EVER)

Description: Mode of encoder when reference is reached.

11.2.1.5.4 Get_EncoderMode<n> (ENCREF<n>)

Get the behaviour of encoder n $\in \{1..2\}$ when reference is reached.

Parameter: int32_t SA_EncoderMode<n>

SA_EncoderMode<n>

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = No action (NONE)
- 1 = Set encoder value to preset value only one time (ONE)
- 2 = Set encoder value to preset value each time (EVER)

Description: Mode of encoder when reference is reached.

11.2.1.5.5 Set_EncoderPreload<n> (ENCVALUE<n>)

Set preload value for encoder n $\in \{1..2\}$.

Parameter: double SP_EncoderPreload<n>

SP_EncoderPreload<n>

Direction: Down

Valid values:

- Minimum:** 0.0
- Maximum:** 4294967295.0 (UINT32_MAX)

Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.2.1.5.6 Get_EncoderPreload<n> (ENCVALUE<n>)

Get preload value for encoder n $\in \{1..2\}$.

Parameter: double SA_EncoderPreload<n>

SA_EncoderPreload<n>

Direction: Up

Valid values:

- Minimum:** 0.0
- Maximum:** 4294967295.0 (UINT32_MAX)

Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.2.1.5.7 Load_Encoder<n> (ENCSET)

Load the encoder n $\in \{1..2\}$ with the preset value.

11.2.1.5.8 EnableRef_Encoder<n> (ENCRESET)

Reset reference counter of encoder n $\in \{1..2\}$. If encoder mode is 2, at next reference encoder value will be set to preset value again

11.2.1.5.9 Set_Encoder.MaxValue<n> (ENCMAX<n>)

Set maximum value for encoder n $\in \{1..2\}$ before it wraps around.

Parameter: double SP_Encoder.MaxValue<n>

SP_Encoder.MaxValue<n>

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Maximum value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.2.1.5.10 Get_Encoder.MaxValue<n> (ENCMAX<n>)

Get maximum value for encoder n $\in \{1..2\}$ before it wraps around.

Parameter: double SA_Encoder.MaxValue<n>

SA_Encoder.MaxValue<n>

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Maximum value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.2.1.6 Interfaces

11.2.1.6.1 Set_IPConfiguration (IPCONFIG)

Set the IP configuration at controller.

For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_DHCPEnabled

SP_DHCPEnabled

Direction: Down

Valid values:

0= FALSE

1= TRUE

Description: Specify if controller should use a static IP address or ask for IP at DHCP server (dynamic IP address).

Parameter: String SP_Address

SP_Address

Direction: Down

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_SubnetMask

SP_SubnetMask

Direction: Down

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_Gateway SP_Gateway

Direction: Down

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: The default gateway must be specified if the controller should communicate with peers in foreign subnets. This parameter is only evaluated on static IP assignment.

11.2.1.6.2 Get_IPConfiguration (IPCONFIG)

Get the IP configuration at controller.

Parameter: int32_t SA_DHCPEnabled SA_DHCPEnabled

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= FALSE

1= TRUE

Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).

Parameter: String SA_Address SA_Address

Direction: Up

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.

Parameter: String SA_SubnetMask SA_SubnetMask

Direction: Up

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.

Parameter: String SA_Gateway SA_Gateway

Direction: Up

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

11.2.1.6.3 Set_IPDataTransferMode (MEATRANSFER)

Set IP protocol at controller.

For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_Protocol SP_Protocol

Direction: Down

Valid values:

0= TCP server (SERVER/TCP)

1= TCP client (CLIENT/TCP)

2= UDP sender (CLIENT/UDP)

3= None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SP_RemoteAddress SP_RemoteAddress

Direction: Down

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to. On TCP server this parameter is ignored.

Parameter: int32_t SP_Port SP_Port

Direction: Down

Valid values:

Minimum: 1024

Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

11.2.1.6.4 Get_IPDataTransferMode (MEATRANSFER)

Get IP protocol at controller.

Parameter: int32_t SA_Protocol SA_Protocol

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= TCP server (SERVER/TCP)

1= TCP client (CLIENT/TCP)

2= UDP sender (CLIENT/UDP)

3= None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SA_RemoteAddress SA_RemoteAddress

Direction: Up

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to.

Parameter: int32_t SA_Port SA_Port

Direction: Up

Valid values:

Minimum: 1024

Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

11.2.1.6.5 Set_TCPKeepAlive (TCPKEEPALIVE)

Set the TCP keep alive feature (see RFC 1122).

Parameter: int32_t SP_TCPKeepAlive SP_TCPKeepAlive

Direction: Down

Valid values:

0= Off

1= On

Description: TCP keep alive feature.

11.2.1.6.6 Get_TCPKeepAlive (TCPKEEPALIVE)

Get the TCP keep alive feature (see RFC 1122).

Parameter: int32_t SA_TCPKeepAlive

SA_TCPKeepAlive

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Off
- 1 = On

Description: TCP keep alive feature.

11.2.1.6.7 Set_EthernetMode (ETHERMODE)

Switches ethernet mode between Ethernet and Ethercat.

Parameter: int32_t SP_EthernetMode

SP_EthernetMode

Direction: Down

Valid values:

- 0 = Ethernet
- 1 = Ethercat

Description: Ethernet mode.

11.2.1.6.8 Get_EthernetMode (ETHERMODE)

Get ethernet mode of controller.

Parameter: int32_t SA_EthernetMode

SA_EthernetMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Ethernet
- 1 = Ethercat

Description: Ethernet mode.

11.2.1.6.9 Set_Baudrate (BAUDRATE)

Set baudrate of controller for serial RS422 communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

- 4000000 (does not work in combination with IF2008 PCI(e) interface card less than FPGA 10 and IF2004 USB adapter less than FPGA 6)
- 3000000
- 2000000
- 921600
- 781250 (only available at Option 217)
- 691200
- 460800
- 230400
- 115200
- 38400 (since firmware V001.041.086)
- 9600

Unit: Baud

Description: Baudrate of controller.

Parameter: int32_t CP_InterruptDataTransfer CP_InterruptDataTransfer
Direction: Down
Valid values:
 0= false
 1= true
Default: 0
Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Set_DataOutInterface](#)) and enabling it again at end (only if enabled before).

11.2.1.6.10 Get_Baudrate (BAUDRATE)

Get baudrate of controller for serial RS422 communication.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate
Direction: Up
Valid values:
 4000000
 3000000
 2000000
 921600
 781250 (only available at Option 217)
 691200
 460800
 230400
 115200
 38400 (since firmware V001.041.086)
 9600
Unit: Baud
Description: Baudrate of controller.

11.2.1.6.11 Set_AppLanguage (LANGUAGE)

Set language of web interface.

Parameter: int32_t SP_ApplicationLanguage SP_ApplicationLanguage
Direction: Down
Valid values:
 0= English (EN)
 1= German (DE)
Description: Language of web interface.

11.2.1.6.12 Get_AppLanguage (LANGUAGE)

Get language of web interface.

Parameter: int32_t SA_ApplicationLanguage SA_ApplicationLanguage
Direction: Up
Valid values:
 0= English (EN)
 1= German (DE)
Description: Language of web interface.

11.2.1.7 Parameter management

11.2.1.7.1 Save_InterfaceParameters (BASICSETTINGS STORE)

Save actual interface parameters at controller.

11.2.1.7.2 Load_InterfaceParameters (BASICSETTINGS READ)

Load stored interface parameters into controller RAM.

11.2.1.7.3 Save_MeasureParameters (MEASSETTINGS STORE)

Save actual measurement parameters at controller.

Parameter: String SP_SettingName SP_SettingName
Direction: Down
Description: Name of setting. If name is empty, current setting will be overwritten.

11.2.1.7.4 Load_MeasureParameters (MEASSETTINGS READ)

Load stored measurement parameters into controller RAM.

At this command the controller may change output data after applying new setting. If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: String SP_SettingName SP_SettingName
Direction: Down
Description: Name of setting.

11.2.1.7.5 Rename_MeasureParameters (MEASSETTINGS RENAME)

Rename stored measurement parameters at controller.

Parameter: String SP_OldName SP_OldName
Direction: Down
Description: Actual name of setting.

Parameter: String SP_NewName SP_NewName
Direction: Down
Description: New name of setting.

Parameter: int32_t SP_Overwrite SP_Overwrite
Direction: Down
Valid values:
 0= no
 1= yes
Description: Force overwriting existing setting.

11.2.1.7.6 Get_CurrentMeasureSetting (MEASSETTINGS CURRENT)

Get name of current (active) measurement parameters.

Parameter: String SA_CurrentName SA_CurrentName
Direction: Up
Description: Name of current (active) setting.

11.2.1.7.7 Get_MeasureSettingsList (MEASSETTINGS LIST)

Get list of all user settings.

Parameter: String SA_SettingNames SA_SettingNames
Direction: Up
Description: List (separated by new line).

11.2.1.7.8 Delete_MeasureParameters (MEASSETTINGS DELETE)

Delete stored measurement parameters at controller.

Parameter: String SP_SettingName SP_SettingName
Direction: Down
Description: Name of setting.

11.2.1.7.9 Get_MeasurePresetList (MEASSETTINGS PRESETLIST)

Get list of all preset settings.

Parameter: String SA_PresetNames SA_PresetNames
Direction: Up
Description: List (separated by new line).

11.2.1.7.10 Set_InitialMeasureSetting (MEASSETTINGS INITIAL)

Set initial (boot time) measure setting.

Parameter: int32_t SP_Automatic SP_Automatic
Direction: Down
Valid values:
 0 = no
 1 = yes
Description: Automatic selection.

Parameter: String SP_InitialName SP_InitialName
Direction: Down
Description: Name of setting.

11.2.1.7.11 Get_InitialMeasureSetting (MEASSETTINGS INITIAL)

Get initial (boot time) measure setting.

Parameter: int32_t SA_Automatic

SA_Automatic

Direction: Up

Valid values:

0= no

1= yes

Description: Automatic selection.

Parameter: String SA_InitialName

SA_InitialName

Direction: Up

Description: Name of setting.

11.2.1.7.12 Set_MeasurePresetMode (MEASSETTINGS PRESETMODE)

Set mode of current preset settings.

Parameter: int32_t SP_PresetMode

SP_PresetMode

Direction: Down

Valid values:

0= Static

1= Balanced

2= Dynamic

3= None (if no preset setting is active)

Description: Preset mode.

11.2.1.7.13 Get_MeasurePresetMode (MEASSETTINGS PRESETMODE)

Get mode of current preset settings.

Parameter: int32_t SA_PresetMode

SA_PresetMode

Direction: Up

Valid values:

0= Static

1= Balanced

2= Dynamic

3= None (if no preset setting is active)

Description: Preset mode.

11.2.1.7.14 Set_Default (SETDEFAULT)

Reset the controller to default settings.

At this command the sensor may change output data after applying default settings.

If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually.

After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DefaultType

SP_DefaultType

Direction: Down

Valid values:

0= Delete all settings and load the factory settings (ALL)

1= Delete all measurement settings (MEASSETTINGS)

2= Delete all basic settings (BASICSETTINGS)

3= Overwrite and load the material table (MATERIAL)

Description: Specifies which settings should be reset.

11.2.1.7.15 Export_Parameters (EXPORT)

Exports the settings of the sensor.

Parameter: int32_t SP_ExportType SP_ExportType

Direction: Down

Valid values:

- 0= Exports only the measuring settings with name (MEASSETTINGS)
- 1= Exports only the basic settings (BASICSETTINGS)
- 2= Exports all measuring settings (MEASSETTINGS_ALL)
- 3= Exports basic settings and all measuring settings (ALL)
- 4= Exports only the table of materials (MATERIALTABLE)

Description: Export type.

Parameter: String SP_SettingName SP_SettingName

Direction: Down

Description: Name of setting to be exported.

This parameter is only used at export type 0.

Parameter: String SA_ExportData SA_ExportData

Direction: Up

Description: Exported data in ASCII format.

11.2.1.7.16 Import_Parameters (IMPORT)

Imports the settings of the sensor.

If SP_ApplyImmediately is 1, the controller may change output data after applying new setting. If first bit of [IP_AutomaticMode](#) is set (1), [Get_AllParameters](#) (SP_-Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_ForceOverwrite SP_ForceOverwrite

Direction: Down

Valid values:

- 0= Do not overwrite existing setting
- 1= Allow to overwrites existing settings (FORCE)

Description: Specify if existing settings can be overwritten.

Parameter: int32_t SP_ApplyImmediately SP_ApplyImmediately

Direction: Down

Valid values:

- 0= Just store imported settings.
- 1= Apply the imported settings (APPLY)

Description: Specify if settings only should be stored or additionally applied.

Parameter: String SP_ImportData SP_ImportData

Direction: Down

Description: Data to be imported (from a former call to Export_Parameters).

11.2.1.8 Internal controller commands

11.2.1.8.1 Set_ChartType (CHARTTYPE)

Set current type of the chart at web interface.

Parameter: int32_t SP_ChartType

SP_ChartType

Direction: Down

Valid values:

0= Measure (MEAS)

1= Video signal (VIDEO)

Description: Type of the chart at web interface.

11.2.1.8.2 Get_ChartType (CHARTTYPE)

Get current type of the chart at web interface.

Parameter: int32_t SA_ChartType

SA_ChartType

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Measure (MEAS)

1= Video signal (VIDEO)

Description: Type of the chart at web interface.

11.2.2 Measurement

11.2.2.1 General

11.2.2.1.1 Set_MeasurePeak (MEASPEAK)

Specifiy which peaks should be evaluated at controller.

At IFD2422 the first channel is referenced.

Parameter: int32_t SP_MeasurePeak

SP_MeasurePeak

Direction: Down

Valid values:

0= First peak at distance mode resp. first and last peak at thickness mode (F_L)

1= Last peak at distance mode resp. last and next to last peak at thickness mode (L_SL)

2= First peak at distance mode resp. first and second peak at thickness mode (F_S)

3= Highest peak at distance mode resp. highest and second highest peak at thickness mode (H_SH)

Description: Peaks to evalualte.

11.2.2.1.2 Get_MeasurePeak (MEASPEAK)

Retrieve which peaks are evaluated at controller.
At IFD2422 the first channel is referenced.

Parameter: int32_t SA_MeasurePeak

SA_MeasurePeak

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= First peak at distance mode resp. first and last peak at thickness mode (F_L)
- 1= Last peak at distance mode resp. last and next to last peak at thickness mode (L_SL)
- 2= First peak at distance mode resp. first and second peak at thickness mode (F_S)
- 3= Highest peak at distance mode resp. highest and second highest peak at thickness mode (H_SH)

Description: Peaks to evalualte.

11.2.2.1.3 Set_MeasurePeak_Ch<n> (MEASPEAK_CH0<n>)

Specifiy which peaks should be evaluated at controller for channel n ∈ {1..2}.

Valid for sensor:

IFD2422

Parameter: int32_t SP_MeasurePeak_Ch<n>

SP_MeasurePeak_Ch<n>

Direction: Down

Valid values:

- 0= First peak at distance mode resp. first and last peak at thickness mode (F_L)
- 1= Last peak at distance mode resp. last and next to last peak at thickness mode (L_SL)
- 2= First peak at distance mode resp. first and second peak at thickness mode (F_S)
- 3= Highest peak at distance mode resp. highest and second highest peak at thickness mode (H_SH)

Description: Peaks to evalualte.

11.2.2.1.4 Get_MeasurePeak_Ch<n> (MEASPEAK_CH0<n>)

Retrieve which peaks are evaluated at controller for channel n ∈ {1..2}.

Valid for sensor:

IFD2422

Parameter: int32_t SA_MeasurePeak_Ch<n>

SA_MeasurePeak_Ch<n>

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= First peak at distance mode resp. first and last peak at thickness mode (F_L)

- 1= Last peak at distance mode resp. last and next to last peak at thickness mode (L_SL)
- 2= First peak at distance mode resp. first and second peak at thickness mode (F_S)
- 3= Highest peak at distance mode resp. highest and second highest peak at thickness mode (H_SH)

Description: Peaks to evalualte.

11.2.2.1.5 Set_ShutterMode (SHUTTERMODE)

Set the shutter mode.

At IFD2422 the first channel is referenced.

Parameter: int32_t SP_ShutterMode

SP_ShutterMode

Direction: Down

Valid values:

- 1= Control exposure time, measuring rate is set manually (MEAS)
- 2= Exposure time and measuring rate are set manually (MANUAL)
- 3= Use two fixed exposure times alternately (2TIMEALT)
- 4= Use the suitable exposure time of two fixed ones (2TIMES)

Description: Shutter mode.

11.2.2.1.6 Get_ShutterMode (SHUTTERMODE)

Get the shutter mode.

At IFD2422 the first channel is referenced.

Parameter: int32_t SA_ShutterMode

SA_ShutterMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 1= Control exposure time, measuring rate is set manually (MEAS)
- 2= Exposure time and measuring rate are set manually (MANUAL)
- 3= Use two fixed exposure times alternately (2TIMEALT)
- 4= Use the suitable exposure time of two fixed ones (2TIMES)

Description: Shutter mode.

11.2.2.1.7 Set_ShutterMode_Ch<n> (SHUTTERMODE_CH0<n>)

Set the shutter mode for channel n $\in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: int32_t SP_ShutterMode_Ch<n>

SP_ShutterMode_Ch<n>

Direction: Down

Valid values:

- 1= Control exposure time, measuring rate is set manually (MEAS)
- 2= Exposure time and measuring rate are set manually (MANUAL)
- 3= Use two fixed exposure times alternately (2TIMEALT)
- 4= Use the suitable exposure time of two fixed ones (2TIMES)

Description: Shutter mode.

11.2.2.1.8 Get_ShutterMode_Ch<n> (SHUTTERMODE_CH0<n>)

Get the shutter mode for channel n $\in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: int32_t SA_ShutterMode_Ch<n>

SA_ShutterMode_Ch<n>

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 1= Control exposure time, measuring rate is set manually (MEAS)
- 2= Exposure time and measuring rate are set manually (MANUAL)
- 3= Use two fixed exposure times alternately (2TIMEALT)
- 4= Use the suitable exposure time of two fixed ones (2TIMES)

Description: Shutter mode.

11.2.2.1.9 Set_ShutterTime (SHUTTER)

Set the shutter time.

At IFD2422 the first channel is referenced.

Parameter: double SP_ShutterTime1

SP_ShutterTime1

Direction: Down

Valid values:

- Minimum:** 1.0
- Maximum:** 10000.0

Unit: μs

Description: First shutter time.

Parameter: double SP_ShutterTime2

SP_ShutterTime2

Direction: Down

Valid values:

- Minimum:** 1.0
- Maximum:** 10000.0

Unit: μs

Description: Second shutter time. Is automatically adapted (to first shutter time), if not less or equal to first shutter time.

11.2.2.1.10 Get_ShutterTime (SHUTTER)

Get the shutter time.

At IFD2422 the first channel is referenced.

Parameter: double SA_ShutterTime1

SA_ShutterTime1

Direction: Up

Valid values:

- Minimum:** 1.0
- Maximum:** 10000.0

Unit: μs

Description: First shutter time.

Parameter: double SA_ShutterTime2 SA_ShutterTime2
Direction: Up
Valid values:
Minimum: 1.0
Maximum: 10000.0
Unit: μs
Description: Second shutter time.

11.2.2.1.11 Set_ShutterTime_Ch<n> (SHUTTER_CH0<n>)

Set the shutter time for channel n $\in \{1..2\}$.

Valid for sensor:
 IFD2422

Parameter: double SP_ShutterTime1_Ch<n> SP_ShutterTime1_Ch<n>
Direction: Down
Valid values:
Minimum: 1.0
Maximum: 10000.0
Unit: μs
Description: First shutter time.

Parameter: double SP_ShutterTime2_Ch<n> SP_ShutterTime2_Ch<n>
Direction: Down
Valid values:
Minimum: 1.0
Maximum: 10000.0
Unit: μs
Description: Second shutter time. Is automatically adapted (to first shutter time), if not less or equal to first shutter time.

11.2.2.1.12 Get_ShutterTime_Ch<n> (SHUTTER_CH0<n>)

Get the shutter time for channel n $\in \{1..2\}$.

Valid for sensor:
 IFD2422

Parameter: double SA_ShutterTime1_Ch<n> SA_ShutterTime1_Ch<n>
Direction: Up
Valid values:
Minimum: 1.0
Maximum: 10000.0
Unit: μs
Description: First shutter time.

Parameter: double SA_ShutterTime2_Ch<n> SA_ShutterTime2_Ch<n>
Direction: Up
Valid values:
Minimum: 1.0
Maximum: 10000.0
Unit: μs
Description: Second shutter time.

11.2.2.1.13 Set_Samplerate (MEASRATE)

Set the samplerate.

Parameter: double SP_Measrate SP_Measrate
Direction: Down
Valid values:
Minimum: 0.1
Maximum: 6.5
Unit: kHz
Description: Samplerate of measurement.

11.2.2.1.14 Get_Samplerate (MEASRATE)

Get the samplerate.

Parameter: double SA_Measrate SA_Measrate
Direction: Up
Valid values:
Minimum: 0.1
Maximum: 6.5
Unit: kHz
Description: Samplerate of measurement.

11.2.2.1.15 Get_VideoStreamSignal

Read one video signal from video stream.

Parameter: int32_t SP_ReadMode SP_ReadMode
Direction: Down
Valid values:
 0 = Each video signal
 1 = Only newest video signal
 2 = Automatic
Description: This mode specifies if each video signal should be read or only the latest one. If set to automatic each video signal is read while the buffer does not overflow. If the buffer becomes full one or more video signals are discarded.

Parameter: int32_t SP_WaitVideoTimeout SP_WaitVideoTimeout
Direction: Down
Unit: ms
Valid values:
Minimum: 0
Maximum: 2147483647 (INT32_MAX)
Description: Timeout to wait for a video signal.

Parameter: Binary data SA_VideoRaw_Ch1 SA_VideoRaw_Ch1
Direction: Up
Valid values:
 512 words (each 2 byte), each word is an intensity value.
Description: Raw video signal of first channel

Parameter: Binary data SA_VideoDark_Ch1	SA_VideoDark_Ch1
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Description: Dark corrected video signal of first channel	
Parameter: Binary data SA_VideoLight_Ch1	SA_VideoLight_Ch1
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Description: Light corrected video signal of first channel	
Parameter: Binary data SA_VideoRaw_Ch2	SA_VideoRaw_Ch2
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Valid for sensor:	
IFD2422	
Description: Raw video signal of second channel.	
Parameter: Binary data SA_VideoDark_Ch2	SA_VideoDark_Ch2
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Valid for sensor:	
IFD2422	
Description: Dark corrected video signal of second channel.	
Parameter: Binary data SA_VideoLight_Ch2	SA_VideoLight_Ch2
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Valid for sensor:	
IFD2422	
Description: Light corrected video signal of second channel.	
Parameter: double SA_VideoTimestamp	SA_VideoTimestamp
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.79769e+308 (DBL_MAX)	
Unit: ms	
Description: Timestamp of the video signal. It starts from 1970 Jan 01 at 01:00. It is generated when the video has arrived at TCP/IP socket.	
Parameter: int32_t SA_SkippedVideo	SA_SkippedVideo
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of skipped video signals, if SP_ReadMode is not 0.	

11.2.2.1.16 Get_DarkCorrTable (DARKCORR_PRINT)

Read the dark correction table.

At IFD2422 the first channel is referenced.

Parameter: Binary data SA_DarkCorrTable

SA_DarkCorrTable

Direction: Up

Valid values:

512 words (each 2 byte), each word is an intensity value.

Description: Dark correction table.

11.2.2.1.17 Get_DarkCorrTable_Ch<n> (DARKCORR_PRINT_CH0<n>)

Read the dark correction table of channel n ∈ {1..2}.

Valid for sensor:

IFD2422

Parameter: Binary data SA_DarkCorrTable_Ch<n>

SA_DarkCorrTable_Ch<n>

Direction: Up

Valid values:

512 words (each 2 byte), each word is an intensity value.

Description: Dark correction table.

11.2.2.1.18 Get_LightCorrTable (LIGHTCORR_PRINT)

Read the light correction table.

At IFD2422 the first channel is referenced.

Parameter: Binary data SA_LightCorrTable

SA_LightCorrTable

Direction: Up

Valid values:

512 words (each 2 byte), each word is an intensity value.

Description: Light correction table.

11.2.2.1.19 Get_LightCorrTable_Ch<n> (LIGHTCORR_PRINT_CH0<n>)

Read the light correction table of channel n ∈ {1..2}.

Valid for sensor:

IFD2422

Parameter: Binary data SA_LightCorrTable_Ch<n>

SA_LightCorrTable_Ch<n>

Direction: Up

Valid values:

512 words (each 2 byte), each word is an intensity value.

Description: Light correction table.

11.2.2.2 Video signal

11.2.2.2.1 Set_ROI (ROI)

Set the region of intererst for processing video signal.
At IFD2422 the first channel is referenced.

Parameter: int32_t SP_ROIStart

SP_ROIStart

Direction: Down

Valid values:

Minimum: 0

Maximum: 511

Unit: Pixel

Description: First position on CCD.

Parameter: int32_t SP_ROIEnd

SP_ROIEnd

Direction: Down

Valid values:

Minimum: 0

Maximum: 511

Unit: Pixel

Description: Last position on CCD.

11.2.2.2.2 Get_ROI (ROI)

Get the region of intererst for processing video signal.
At IFD2422 the first channel is referenced.

Parameter: int32_t SA_ROIStart

SA_ROIStart

Direction: Up

Valid values:

Minimum: 0

Maximum: 511

Unit: Pixel

Description: First position on CCD.

Parameter: int32_t SA_ROIEnd

SA_ROIEnd

Direction: Up

Valid values:

Minimum: 0

Maximum: 511

Unit: Pixel

Description: Last position on CCD.

11.2.2.2.3 Set_ROI_Ch<n> (ROI_CH0<n>)

Set the region of intererst for processing video signal of channel n ∈ {1..2}.

Valid for sensor:

IFD2422

Parameter: int32_t SP_ROIStart_Ch<n>

SP_ROIStart_Ch<n>

Direction: Down

Valid values:

Minimum: 0

Maximum: 511

Unit: Pixel

Description: First position on CCD.

Parameter: int32_t SP_ROIEnd_Ch<n> SP_ROIEnd_Ch<n>
Direction: Down
Valid values:
 Minimum: 0
 Maximum: 511
Unit: Pixel
Description: Last position on CCD.

11.2.2.2.4 Get_ROI_Ch<n> (ROI_CH0<n>)

Get the region of interest for processing video signal of channel n ∈ {1..2}.

Valid for sensor:
 IFD2422

Parameter: int32_t SA_ROIStart_Ch<n> SA_ROIStart_Ch<n>
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 511
Unit: Pixel
Description: First position on CCD.

Parameter: int32_t SA_ROIEnd_Ch<n> SA_ROIEnd_Ch<n>
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 511
Unit: Pixel
Description: Last position on CCD.

11.2.2.3 Material database

11.2.2.3.1 Get_MaterialTable (MATERIALTABLE)

Get a list of all materials stored in the controller.

Parameter: String SA_MaterialTable SA_MaterialTable
Direction: Up
Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_MaterialTableCount SA_MaterialTableCount
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 20
Description: Number of entries in the table. All following parameters exist from 1 to this number, e.g. SA_Pos1, SA_Pos2, ...

Parameter: int32_t SA_Pos1..x	SA_Pos1..x
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the material in the table.	
Parameter: String SA_MaterialName1..x	SA_MaterialName1..x
Direction: Up	
Description: Name of the material in the table.	
Parameter: double SA_RefRACTIVEINDEX_nF1..x	SA_RefRACTIVEINDEX_nF1..x
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 486 nm.	
Parameter: double SA_RefRACTIVEINDEX_nd1..x	SA_RefRACTIVEINDEX_nd1..x
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 587 nm.	
Parameter: double SA_RefRACTIVEINDEX_nC1..x	SA_RefRACTIVEINDEX_nC1..x
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 656 nm.	
Parameter: double SA_AbbeNumber_vd1..x	SA_AbbeNumber_vd1..x
Direction: Up	
Valid values:	
Minimum: 10.0	
Maximum: 100.0	
Description: Abbe number of the material or 0.0 if not specified.	
Parameter: String SA_Description1..x	SA_Description1..x
Direction: Up	
Description: Description of the material in the table.	

11.2.2.3.2 Set_ActiveMaterial (MATERIAL)

Select the material between distances 1 and 2.
At IFD2422 the first channel is referenced.

Parameter: String SP_ActiveMaterial	SP_ActiveMaterial
Direction: Down	
Description: Name of material.	

11.2.2.3.3 Get_ActiveMaterial (MATERIAL)

Get the material between distances 1 and 2.
At IFD2422 the first channel is referenced.

Parameter: String SA_ActiveMaterial
Direction: Up
Description: Name of material.

SA_ActiveMaterial

11.2.2.3.4 Set_ActiveMaterial_Ch<n> (MATERIAL_CH0<n>)

Select the material between distances 1 and 2 for channel n $\in \{1..2\}$.

Valid for sensor:
IFD2422

Parameter: String SP_ActiveMaterial_Ch<n>
Direction: Down
Description: Name of material.

SP_ActiveMaterial_Ch<n>

11.2.2.3.5 Get_ActiveMaterial_Ch<n> (MATERIAL_CH0<n>)

Get the material between distances 1 and 2 of channel n $\in \{1..2\}$.

Valid for sensor:
IFD2422

Parameter: String SA_ActiveMaterial_Ch<n>
Direction: Up
Description: Name of material.

SA_ActiveMaterial_Ch<n>

11.2.2.3.6 Get_MaterialInfo (MATERIALINFO)

Get information of active material.
At IFD2422 the first channel is referenced.

Parameter: int32_t SP_MaterialIndex

SP_MaterialIndex

Direction: Down

Valid values:

Minimum: 0

Maximum: 5

Default: Empty, means first material.

Description: Index of material to get info. If this parameter is 0 or not set, MATERIALINFO without parameter is called (for compatibility mode)

Parameter: String SA_MaterialName

SA_MaterialName

Direction: Up

Description: Name of the active material.

Parameter: String SA_Description

SA_Description

Direction: Up

Description: Description of the active material.

Parameter: double SA_RefractiveIndex_nF	SA_RefractiveIndex_nF
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 486 nm.	
Parameter: double SA_RefractiveIndex_nd	SA_RefractiveIndex_nd
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 587 nm.	
Parameter: double SA_RefractiveIndex_nC	SA_RefractiveIndex_nC
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 656 nm.	
Parameter: double SA_AbbeNumber_vd	SA_AbbeNumber_vd
Direction: Up	
Valid values:	
Minimum: 10.0	
Maximum: 100.0	
Description: Abbe number of the active material or 0.0 if not specified.	
Parameter: int32_t SA_Described_by	SA_Described_by
Direction: Up	
Valid values:	
0= nF, nd and nC	
1= nd and abbe value	
Description: Tells if nF, nd and nC or if nd and abbe value is valid.	

11.2.2.3.7 Get_MaterialInfo_Ch<n> (MATERIALINFO_CH0<n>)

Get information of active material of channel n $\in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: int32_t SP_MaterialIndex_Ch<n>	SP_MaterialIndex_Ch<n>
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 5	
Default: Empty, means first material.	
Description: Index of material to get info. If this parameter is 0 or not set, MATERIALINFO without parameter is called (for compatibility mode)	
Parameter: String SA_MaterialName_Ch<n>	SA_MaterialName_Ch<n>
Direction: Up	
Description: Name of the active material.	

Parameter: String SA_Description_Ch<n>	SA_Description_Ch<n>
Direction: Up	
Description: Description of the active material.	
Parameter: double SA_RefractiveIndex_nF_Ch<n>	SA_RefractiveIndex_nF_- Ch<n>
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 486 nm.	
Parameter: double SA_RefractiveIndex_nd_Ch<n>	SA_RefractiveIndex_nd_- Ch<n>
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 587 nm.	
Parameter: double SA_RefractiveIndex_nC_Ch<n>	SA_RefractiveIndex_nC_- Ch<n>
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 656 nm.	
Parameter: double SA_AbbeNumber_vd_Ch<n>	SA_AbbeNumber_vd_Ch<n>
Direction: Up	
Valid values:	
Minimum: 10.0	
Maximum: 100.0	
Description: Abbe number of the active material or 0.0 if not specified.	
Parameter: int32_t SA_Described_by_Ch<n>	SA_Described_by_Ch<n>
Direction: Up	
Valid values:	
0= nF, nd and nC	
1= nd and abbe value	
Description: Tells if nF, nd and nC or if nd and abbe value is valid.	

11.2.2.3.8 Edit_Material_Abbe (MATERIALEDIT)

Edit or add new material by using Abbe number.

Parameter: String SP_MaterialName	SP_MaterialName
Direction: Down	
Description: Name of the material.	
Parameter: String SP_Description	SP_Description
Direction: Down	
Description: Description of the material.	

Parameter: double SP_RefractiveIndex_nd	SP_RefractiveIndex_nd
Direction: Down	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 587 nm.	
Parameter: double SP_AbbeNumber_vd	SP_AbbeNumber_vd
Direction: Down	
Valid values:	
Minimum: 10.0	
Maximum: 100.0	
Description: Abbe number of the material.	

11.2.2.3.9 Edit_Material_Nx (MATERIALEDIT)

Edit or add new material by using three refractive indices.

Parameter: String SP_MaterialName	SP_MaterialName
Direction: Down	
Description: Name of the material.	
Parameter: String SP_Description	SP_Description
Direction: Down	
Description: Description of the material.	
Parameter: double SP_RefractiveIndex_nF	SP_RefractiveIndex_nF
Direction: Down	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 486 nm.	
Parameter: double SP_RefractiveIndex_nd	SP_RefractiveIndex_nd
Direction: Down	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 587 nm.	
Parameter: double SP_RefractiveIndex_nC	SP_RefractiveIndex_nC
Direction: Down	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 656 nm.	

11.2.2.3.10 Delete_Material (MATERIALDELETE)

Deletes an existing material.

Parameter: String SP_MaterialName	SP_MaterialName
Direction: Down	
Description: Name of the material to delete.	

11.2.2.3.11 Clear_MaterialTable

Clear the whole material table.

11.2.2.4 Peak processing

11.2.2.4.1 Set_RefractiveCorrection (REFRACCORR)

Specify refractive correction.

At IFD2422 the first channel is referenced.

Parameter: int32_t SP_RefractionCorrection

SP_RefractionCorrection

Direction: Down

Valid values:

0= Off

1= On

Description: Specify if refractive correction should be enabled.

11.2.2.4.2 Get_RefractionCorrection (REFRACCORR)

Tells if refractive correction is enabled

At IFD2422 the first channel is referenced.

Parameter: int32_t SA_RefractionCorrection

SA_RefractionCorrection

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Off

1= On

Description: Tells if refractive correction is be enabled.

11.2.2.4.3 Set_RefractionCorrection_Ch<n> (REFRACCORR_CH0<n>)

Specify refractive correction of channel n $\in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: int32_t SP_RefractionCorrection_Ch<n>

SP_RefractionCorrection_-
Ch<n>

Direction: Down

Valid values:

0= Off

1= On

Description: Specify if refractive correction should be enabled.

11.2.2.4.4 Get_RefractiveCorrection_Ch<n> (REFRACCORR_CH0<n>)

Tells if refractive correction is enabled for channel n ∈ {1..2}.

Valid for sensor:

IFD2422

Parameter: int32_t SA_RefractiveCorrection_Ch<n>

SA_RefractiveCorrection_-
Ch<n>

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Off
- 1 = On

Description: Tells if refractive correction is be enabled.

11.2.2.4.5 Set_PeakCount (PEAKCOUNT)

Set the maximum number of peaks.

At IFD2422 the first channel is referenced.

Parameter: int32_t SP_NumberOfPeaks

SP_NumberOfPeaks

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 2 (Standard) or 6 (Multipeak option)

Description: Maximum number of peaks.

11.2.2.4.6 Get_PeakCount (PEAKCOUNT)

Get the maximum number of peaks.

At IFD2422 the first channel is referenced.

Parameter: int32_t SA_NumberOfPeaks

SA_NumberOfPeaks

Direction: Up

Valid values:

- Minimum:** 1
- Maximum:** 2 (Standard) or 6 (Multipeak option)

Description: Maximum number of peaks.

11.2.2.4.7 Set_PeakCount_Ch<n> (PEAKCOUNT_CH0<n>)

Set the maximum number of peaks for channel n ∈ {1..2}.

Valid for sensor:

IFD2422

Parameter: int32_t SP_NumberOfPeaks_Ch<n>

SP_NumberOfPeaks_Ch<n>

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 2 (Standard) or 6 (Multipeak option)

Description: Maximum number of peaks.

11.2.2.4.8 Get_PeakCount_Ch<n> (PEAKCOUNT_CH0<n>)

Get the maximum number of peaks of channel n $\in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: int32_t SA_NumberOfPeaks_Ch<n>

SA_NumberOfPeaks_Ch<n>

Direction: Up

Valid values:

Minimum: 1

Maximum: 2 (Standard) or 6 (Multipeak option)

Description: Maximum number of peaks.

11.2.2.4.9 Set_MaterialMultiPeak (MATERIALMP)

Set up to five materials for multipeak calculation.

At IFD2422 the first channel is referenced.

For this command an [Update_...](#) meta command is available.

Parameter: String SP_MaterialMultiPeak12

SP_MaterialMultiPeak12

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between first and second peak.

Parameter: String SP_MaterialMultiPeak23

SP_MaterialMultiPeak23

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between second and third peak.

Parameter: String SP_MaterialMultiPeak34

SP_MaterialMultiPeak34

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between third and fourth peak.

Parameter: String SP_MaterialMultiPeak45

SP_MaterialMultiPeak45

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between fourth and fifth peak.

Parameter: String SP_MaterialMultiPeak56

SP_MaterialMultiPeak56

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between fifth and sixth peak.

11.2.2.4.10 Get_MaterialMultiPeak (MATERIALMP)

Get all material names for multipeak calculation.

At IFD2422 the first channel is referenced.

Parameter: String SA_MaterialMultiPeak12

SA_MaterialMultiPeak12

Direction: Up

Description: Name of material between first and second peak.

Parameter: String SA_MaterialMultiPeak23	SA_MaterialMultiPeak23
Direction: Up	
Description: Name of material between second and third peak.	
Parameter: String SA_MaterialMultiPeak34	SA_MaterialMultiPeak34
Direction: Up	
Description: Name of material between third and fourth peak.	
Parameter: String SA_MaterialMultiPeak45	SA_MaterialMultiPeak45
Direction: Up	
Description: Name of material between fourth and fifth peak.	
Parameter: String SA_MaterialMultiPeak56	SA_MaterialMultiPeak56
Direction: Up	
Description: Name of material between fifth and sixth peak.	

11.2.2.4.11 Set_MaterialMultiPeak_Ch<n> (MATERIALMP_CH0<n>)

Set up to five materials for multipeak calculation of channel n $\in \{1..2\}$.
 For this command an [Update ...](#) meta command is available.

Valid for sensor:

IFD2422

Parameter: String SP_MaterialMultiPeak12_Ch<n>	SP_MaterialMultiPeak12_- Ch<n>
Direction: Down	
Default: "" (empty string, means refractive index 1.0)	
Description: Name of material between first and second peak.	
Parameter: String SP_MaterialMultiPeak23_Ch<n>	SP_MaterialMultiPeak23_- Ch<n>
Direction: Down	
Default: "" (empty string, means refractive index 1.0)	
Description: Name of material between second and third peak.	
Parameter: String SP_MaterialMultiPeak34_Ch<n>	SP_MaterialMultiPeak34_- Ch<n>
Direction: Down	
Default: "" (empty string, means refractive index 1.0)	
Description: Name of material between third and fourth peak.	
Parameter: String SP_MaterialMultiPeak45_Ch<n>	SP_MaterialMultiPeak45_- Ch<n>
Direction: Down	
Default: "" (empty string, means refractive index 1.0)	
Description: Name of material between fourth and fifth peak.	
Parameter: String SP_MaterialMultiPeak56_Ch<n>	SP_MaterialMultiPeak56_- Ch<n>
Direction: Down	
Default: "" (empty string, means refractive index 1.0)	
Description: Name of material between fifth and sixth peak.	

11.2.2.4.12 Get_MaterialMultiPeak_Ch<n> (MATERIALMP_CH0<n>)

Get all material names for multipeak calculation of channel n ∈ {1..2}.

Valid for sensor:

IFD2422

Parameter: String SA_MaterialMultiPeak12_Ch<n>

Direction: Up

Description: Name of material between first and second peak.

SA_MaterialMultiPeak12_-
Ch<n>

Parameter: String SA_MaterialMultiPeak23_Ch<n>

Direction: Up

Description: Name of material between second and third peak.

SA_MaterialMultiPeak23_-
Ch<n>

Parameter: String SA_MaterialMultiPeak34_Ch<n>

Direction: Up

Description: Name of material between third and fourth peak.

SA_MaterialMultiPeak34_-
Ch<n>

Parameter: String SA_MaterialMultiPeak45_Ch<n>

Direction: Up

Description: Name of material between fourth and fifth peak.

SA_MaterialMultiPeak45_-
Ch<n>

Parameter: String SA_MaterialMultiPeak56_Ch<n>

Direction: Up

Description: Name of material between fifth and sixth peak.

SA_MaterialMultiPeak56_-
Ch<n>

11.2.2.4.13 Set_MinimumThreshold (MIN_THRESHOLD)

Set the minimum threshold.

At IFD2422 the first channel is referenced.

Parameter: double SP_MinimumThreshold

SP_MinimumThreshold

Direction: Down

Valid values:

Minimum: System minimum (usually 0.5)

Maximum: 100.0

Unit: %

Description: A valid peak must be above this threshold.

11.2.2.4.14 Get_MinimumThreshold (MIN_THRESHOLD)

Get the minimum threshold.

At IFD2422 the first channel is referenced.

Parameter: double SA_MinimumThreshold

SA_MinimumThreshold

Direction: Up

Valid values:

Minimum: System minimum (usually 0.5)

Maximum: 100.0

Unit: %

Description: A valid peak must be above this threshold.

11.2.2.4.15 Set_MinimumThreshold_Ch< n > (MIN_THRESHOLD_CH0< n >)

Set the minimum threshold for channel $n \in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: double SP_MinimumThreshold_Ch< n >

SP_MinimumThreshold_Ch< n >

Direction: Down

Valid values:

Minimum: System minimum (usually 0.5)

Maximum: 100.0

Unit: %

Description: A valid peak must be above this threshold.

11.2.2.4.16 Get_MinimumThreshold_Ch< n > (MIN_THRESHOLD_CH0< n >)

Get the minimum threshold of channel $n \in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: double SA_MinimumThreshold_Ch< n >

SA_MinimumThreshold_Ch< n >

Direction: Up

Valid values:

Minimum: System minimum (usually 0.5)

Maximum: 100.0

Unit: %

Description: A valid peak must be above this threshold.

11.2.2.4.17 Set_PeakModulation (PEAK_MODULATION)

Set how a peak must be modulated, so that it is detected as a single peak.

At IFD2422 the first channel is referenced.

Parameter: double SP_PeakModulation

SP_PeakModulation

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 100.0

Unit: %

Description: Value how a peak is modulated.

11.2.2.4.18 Get_PeakModulation (PEAK_MODULATION)

Get how a peak is modulated, so that it is detected as a single peak.

At IFD2422 the first channel is referenced.

Parameter: double SA_PeakModulation

SA_PeakModulation

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 100.0

Unit: %

Description: Value how a peak is modulated.

11.2.2.4.19 Set_PeakModulation_Ch<n> (PEAK_MODULATION_CH01)

Set how a peak must be modulated at channel n $\in \{1..2\}$, so that it is detected as a single peak.

Valid for sensor:

IFD2422

Parameter: double SP_PeakModulation_Ch<n>

SP_PeakModulation_Ch<n>

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 100.0

Unit: %

Description: Value how a peak is modulated.

11.2.2.4.20 Get_PeakModulation_Ch<n> (PEAK_MODULATION_CH01)

Get how a peak is modulated at channel n $\in \{1..2\}$, so that it is detected as a single peak.

Valid for sensor:

IFD2422

Parameter: double SA_PeakModulation_Ch<n>

SA_PeakModulation_Ch<n>

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 100.0

Unit: %

Description: Value how a peak is modulated.

11.2.2.5 Measurement value processing

11.2.2.5.1 Set_SpikeCorrection (SPIKECORR)

Set spike correction at controller.

At IFD2422 the first channel is referenced.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_SpikeCorrection

SP_SpikeCorrection

Direction: Down

Valid values:

0= Off

1= On

Description: Spike correction.

Parameter: int32_t SP_NbrEvaluatedValues

SP_NbrEvaluatedValues

Direction: Down

Valid values:

Minimum: 1

Maximum: 10

Description: Number of values to evaluate for spike correction.

Parameter: double SP_ToleranceRange	SP_ToleranceRange
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 2147.0	
Unit: mm	
Description: Tolerance range for spike correction.	
Parameter: int32_t SP_NbrCorrectedValues	SP_NbrCorrectedValues
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 100	
Description: Number of values to correct at spike correction.	
11.2.2.5.2 Get_SpikeCorrection (SPIKECORR)	
Get spike correction at controller.	
At IFD2422 the first channel is referenced.	
Parameter: int32_t SA_SpikeCorrection	SA_SpikeCorrection
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Description: Spike correction.	
Parameter: int32_t SA_NbrEvaluatedValues	SA_NbrEvaluatedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Number of values to evaluate for spike correction.	
Parameter: double SA_ToleranceRange	SA_ToleranceRange
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 2147.0	
Unit: mm	
Description: Tolerance range for spike correction.	
Parameter: int32_t SA_NbrCorrectedValues	SA_NbrCorrectedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100	
Description: Number of values to correct at spike correction.	

11.2.2.5.3 Set_SpikeCorrection_Ch<n> (SPIKECORR_CH0<n>)

Set spike correction at controller for channel n $\in \{1..2\}$.

For this command an [Update_...](#) meta command is available.

Valid for sensor:

IFD2422

Parameter: int32_t SP_SpikeCorrection_Ch<n>

SP_SpikeCorrection_Ch<n>

Direction: Down

Valid values:

0 = Off

1 = On

Description: Spike correction.

Parameter: int32_t SP_NbrEvaluatedValues_Ch<n>

SP_NbrEvaluatedValues_-
Ch<n>

Direction: Down

Valid values:

Minimum: 1

Maximum: 10

Description: Number of values to evaluate for spike correction.

Parameter: double SP_ToleranceRange_Ch<n>

SP_ToleranceRange_Ch<n>

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 2147.0

Unit: mm

Description: Tolerance range for spike correction.

Parameter: int32_t SP_NbrCorrectedValues_Ch<n>

SP_NbrCorrectedValues_-
Ch<n>

Direction: Down

Valid values:

Minimum: 1

Maximum: 100

Description: Number of values to correct at spike correction.

11.2.2.5.4 Get_SpikeCorrection_Ch<n> (SPIKECORR_CH0<n>)

Get spike correction at controller of channel n $\in \{1..2\}$.

Valid for sensor:

IFD2422

Parameter: int32_t SA_SpikeCorrection_Ch<n>

SA_SpikeCorrection_Ch<n>

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor

0 = Off

1 = On

Description: Spike correction.

Parameter: int32_t SA_NbrEvaluatedValues_Ch<n>	SA_NbrEvaluatedValues_- Ch<n>
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Number of values to evaluate for spike correction.	
Parameter: double SA_ToleranceRange_Ch<n>	SA_ToleranceRange_Ch<n>
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 2147.0	
Unit: mm	
Description: Tolerance range for spike correction.	
Parameter: int32_t SA_NbrCorrectedValues_Ch<n>	SA_NbrCorrectedValues_- Ch<n>
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100	
Description: Number of values to correct at spike correction.	

11.2.2.5.5 Set_StatisticSignal (STATISTICSIGNAL)

Set the measured value which is used for statistic calculation.

Parameter: int32_t SP_StatisticSignal	SP_StatisticSignal
Direction: Down	
Valid values:	
-1= Use SP_StatisticSignalName	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: Signal which is used for statistic calculation.	
Parameter: String SP_StatisticSignalName	SP_StatisticSignalName
Direction: Down	
Description: User defined signal which is used for statistic calculation.	
Parameter: int32_t SP_StatisticDepth	SP_StatisticDepth
Direction: Down	
Valid values:	
0= None (deletes the specified statistic calculation)	
2, 4, 8, ..., 8192, 16384	
16385= Infinite (endless statistic calculation)	
Description: Number of values (depth) for statistic calculation.	

11.2.2.5.6 Get_StatisticSignal (STATISTICSIGNAL)

Get the measured value which is used for statistic calculation.

Parameter: int32_t SP_StatisticSignal

SP_StatisticSignal

Direction: Down

Valid values:

- 1= Use SP_StatisticSignalName
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel
- 2= Distance 3 of first Channel
- 3= Distance 4 of first Channel
- 4= Distance 5 of first Channel
- 5= Distance 6 of first Channel
- 6= Distance 1 of second Channel (only available at IFD2422)
- 7= Distance 2 of second Channel (only available at IFD2422)
- 8= Distance 3 of second Channel (only available at IFD2422)
- 9= Distance 4 of second Channel (only available at IFD2422)
- 10= Distance 5 of second Channel (only available at IFD2422)
- 11= Distance 6 of second Channel (only available at IFD2422)

Description: Signal which is used for statistic calculation.

Parameter: String SP_StatisticSignalName

SP_StatisticSignalName

Direction: Down

Description: User defined signal which is used for statistic calculation.

Parameter: String SA_StatisticSignalTable

SA_StatisticSignalTable

Direction: Up

Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_StatisticSignalEntries

SA_StatisticSignalEntries

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 10

Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_StatisticSignalEntry1, SA_StatisticSignalEntry2, ...

Parameter: String SA_StatisticSignalEntry_{1..x}

SA_StatisticSignalEntry_{1..x}

Direction: Up

Description: Each statistic signal as complete line.

Parameter: int32_t SA_StatisticSignalIndex_{1..x}

SA_StatisticSignalIndex_{1..x}

Direction: Up

Valid values:

- 1= User defined name at SA_StatisticSignalName_{1..x}
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel
- 2= Distance 3 of first Channel
- 3= Distance 4 of first Channel
- 4= Distance 5 of first Channel
- 5= Distance 6 of first Channel
- 6= Distance 1 of second Channel (only available at IFD2422)

- 7= Distance 2 of second Channel (only available at IFD2422)
- 8= Distance 3 of second Channel (only available at IFD2422)
- 9= Distance 4 of second Channel (only available at IFD2422)
- 10= Distance 5 of second Channel (only available at IFD2422)
- 11= Distance 6 of second Channel (only available at IFD2422)

Description: Index of each statistic signal.

Parameter: String SA_StatisticSignalName`1..x`

SA_StatisticSignal-
Name`1..x`

Direction: Up

Description: Name of each statistic signal.

Parameter: int32_t SA_StatisticDepth`1..x`

SA_StatisticDepth`1..x`

Direction: Up

Valid values:

- 2, 4, 8, ..., 8192, 16384
- 16385= Infinite (endless statistic calculation)

Description: Number of values (depth) for statistic calculation.

11.2.2.5.7 Reset_Statistic (RESETSTATISTIC)

Reset all statistic signals.

11.2.2.5.8 Reset_StatisticSignal (STATIC RESET)

Reset the values of the statistic signals.

Parameter: int32_t SP_StatisticSignal

SP_StatisticSignal

Direction: Down

Valid values:

- 1= Use SP_StatisticSignalName
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel
- 2= Distance 3 of first Channel
- 3= Distance 4 of first Channel
- 4= Distance 5 of first Channel
- 5= Distance 6 of first Channel
- 6= Distance 1 of second Channel (only available at IFD2422)
- 7= Distance 2 of second Channel (only available at IFD2422)
- 8= Distance 3 of second Channel (only available at IFD2422)
- 9= Distance 4 of second Channel (only available at IFD2422)
- 10= Distance 5 of second Channel (only available at IFD2422)
- 11= Distance 6 of second Channel (only available at IFD2422)
- 12= All

Description: Signal which is used for statistic calculation.

Parameter: String SP_StatisticSignalName

SP_StatisticSignalName

Direction: Down

Description: User defined signal which is used for statistic calculation.

11.2.2.5.9 Set_MasterSignal (MASTERSIGNAL)

Set the measured value which is used for mastering.

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

- 1= Use SP_MasterSignalName
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel
- 2= Distance 3 of first Channel
- 3= Distance 4 of first Channel
- 4= Distance 5 of first Channel
- 5= Distance 6 of first Channel
- 6= Distance 1 of second Channel (only available at IFD2422)
- 7= Distance 2 of second Channel (only available at IFD2422)
- 8= Distance 3 of second Channel (only available at IFD2422)
- 9= Distance 4 of second Channel (only available at IFD2422)
- 10= Distance 5 of second Channel (only available at IFD2422)
- 11= Distance 6 of second Channel (only available at IFD2422)

Description: Signal which is used for mastering.

Parameter: String SP_MasterSignalName

SP_MasterSignalName

Direction: Down

Description: User defined signal which is used for mastering.

Parameter: double SP_MasterValue

SP_MasterValue

Direction: Down

Valid values:

- Minimum:** -2147.0
- Maximum:** 2147.0

Unit: mm

Description: Master value

11.2.2.5.10 Get_MasterSignal (MASTERSIGNAL)

Get the measured value which is used for mastering.

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

- 1= Use SP_MasterSignalName
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel
- 2= Distance 3 of first Channel
- 3= Distance 4 of first Channel
- 4= Distance 5 of first Channel
- 5= Distance 6 of first Channel
- 6= Distance 1 of second Channel (only available at IFD2422)
- 7= Distance 2 of second Channel (only available at IFD2422)
- 8= Distance 3 of second Channel (only available at IFD2422)
- 9= Distance 4 of second Channel (only available at IFD2422)
- 10= Distance 5 of second Channel (only available at IFD2422)
- 11= Distance 6 of second Channel (only available at IFD2422)

Description: Signal which is used for mastering.

Parameter: String SP_MasterSignalName	SP_MasterSignalName
Direction: Down	
Description: User defined signal which is used for mastering.	
Parameter: String SA_MasterSignalTable	SA_MasterSignalTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_MasterSignalEntries	SA_MasterSignalEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MasterSignalEntry1, SA_MasterSignalEntry2, ...	
Parameter: String SA_MasterSignalEntry1..x	SA_MasterSignalEntry1..x
Direction: Up	
Description: Each master signal as complete line.	
Parameter: int32_t SA_MasterSignalIndex1..x	SA_MasterSignalIndex1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_MasterSignalName1..x	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: Index of each master signal.	
Parameter: String SA_MasterSignalName1..x	SA_MasterSignalName1..x
Direction: Up	
Description: Name of each master signal.	
Parameter: double SA_MasterValue1..x	SA_MasterValue1..x
Direction: Up	
Valid values:	
Minimum: -2147.0	
Maximum: 2147.0	
Unit: mm	
Description: Master value	

11.2.2.5.11 Reset_MasterSignal (MASTERSIGNAL)

Reset the measured value which is used for mastering.

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

- 1= Use SP_MasterSignalName
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel
- 2= Distance 3 of first Channel
- 3= Distance 4 of first Channel
- 4= Distance 5 of first Channel
- 5= Distance 6 of first Channel
- 6= Distance 1 of second Channel (only available at IFD2422)
- 7= Distance 2 of second Channel (only available at IFD2422)
- 8= Distance 3 of second Channel (only available at IFD2422)
- 9= Distance 4 of second Channel (only available at IFD2422)
- 10= Distance 5 of second Channel (only available at IFD2422)
- 11= Distance 6 of second Channel (only available at IFD2422)

Description: Signal which should be reset.

Parameter: String SP_MasterSignalName

SP_MasterSignalName

Direction: Down

Description: User defined signal which should be reset.

11.2.2.5.12 Exec_Master (MASTER)

Executes mastering for specified signal.

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

- 1= Use SP_MasterSignalName
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel
- 2= Distance 3 of first Channel
- 3= Distance 4 of first Channel
- 4= Distance 5 of first Channel
- 5= Distance 6 of first Channel
- 6= Distance 1 of second Channel (only available at IFD2422)
- 7= Distance 2 of second Channel (only available at IFD2422)
- 8= Distance 3 of second Channel (only available at IFD2422)
- 9= Distance 4 of second Channel (only available at IFD2422)
- 10= Distance 5 of second Channel (only available at IFD2422)
- 11= Distance 6 of second Channel (only available at IFD2422)
- 12= All

Description: Signal which should be mastered.

Parameter: String SP_MasterSignalName

SP_MasterSignalName

Direction: Down

Description: User defined signal which should be mastered.

Parameter: int32_t SP_MasterAction SP_MasterAction
Direction: Down
Valid values:
 0= Reset
 1= Set
Description: Mastering action

11.2.2.5.13 Get_Master (MASTER)

Get mastering for specified signal.

Parameter: int32_t SP_Master SP_Master
Direction: Down
Valid values:
 -1= Use SP_MasterName
 0= Distance 1 of first Channel
 1= Distance 2 of first Channel
 2= Distance 3 of first Channel
 3= Distance 4 of first Channel
 4= Distance 5 of first Channel
 5= Distance 6 of first Channel
 6= Distance 1 of second Channel (only available at IFD2422)
 7= Distance 2 of second Channel (only available at IFD2422)
 8= Distance 3 of second Channel (only available at IFD2422)
 9= Distance 4 of second Channel (only available at IFD2422)
 10= Distance 5 of second Channel (only available at IFD2422)
 11= Distance 6 of second Channel (only available at IFD2422)
 12= All
Description: Signal which should be retrieved.

Parameter: String SP_MasterName SP_MasterName
Direction: Down
Description: User defined which should be retrieved.

Parameter: String SA_MasterTable SA_MasterTable
Direction: Up
Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_MasterEntries SA_MasterEntries
Direction: Up
Valid values:
Minimum: 0
Maximum: 10
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MasterEntry1, SA_MasterEntry2, ...

Parameter: String SA_MasterEntry_{1..x} SA_MasterEntry_{1..x}
Direction: Up
Description: Each master as complete line.

Parameter: int32_t SA_MasterIndex1..x SA_MasterIndex1..x

Direction: Up

Valid values:

- 1= User defined name at SA_MasterName1..x
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel
- 2= Distance 3 of first Channel
- 3= Distance 4 of first Channel
- 4= Distance 5 of first Channel
- 5= Distance 6 of first Channel
- 6= Distance 1 of second Channel (only available at IFD2422)
- 7= Distance 2 of second Channel (only available at IFD2422)
- 8= Distance 3 of second Channel (only available at IFD2422)
- 9= Distance 4 of second Channel (only available at IFD2422)
- 10= Distance 5 of second Channel (only available at IFD2422)
- 11= Distance 6 of second Channel (only available at IFD2422)

Description: Index of each master.

Parameter: String SA_MasterName1..x SA_MasterName1..x

Direction: Up

Description: Name of each master.

Parameter: int32_t SA_MasterAction1..x SA_MasterAction1..x

Direction: Up

Valid values:

- 0= Inactive
- 1= Active

Description: Mastering action

11.2.2.5.14 Set_Computation (COMP)

Create/modify customised measurement data processing computations.

Parameter: int32_t SP_ComputationQueue SP_ComputationQueue

Direction: Down

Valid values:

- 0= Channel 1
- 1= Channel 2 (only available at IFD2422)
- 2= System (only available at IFD2422)

Description: Computation queue

Parameter: int32_t SP_ComputationID SP_ComputationID

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 10

Description: Index within computation queue

Parameter: int32_t SP_ComputationOperation	SP_ComputationOperation
Direction: Down	
Valid values:	
0= None	
1= Median	
2= Moving average	
3= Recursive averaging	
4= Calculation	
5= Thickness	
6= Copy	
Description: Computation operation	
Parameter: int32_t SP_ComputationSignalIndex	SP_ComputationSignalIndex
Direction: Down	
Valid values:	
-1= Use SP_ComputationSignalName	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: Signal index (if operation is averaging or copy).	
Parameter: String SP_ComputationSignalName	SP_ComputationSignalName
Direction: Down	
Description: User defined signal (if operation is averaging or copy).	
Parameter: int32_t SP_ComputationMovingCount	SP_ComputationMovingCount
Direction: Down	
Valid values:	
2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096	
Description: Number of value for the averaging window. This parameter is only used at moving average.	
Parameter: int32_t SP_ComputationRecursiveCount	SP_ComputationRecursiveCount
Direction: Down	
Valid values:	
Minimum: 2	
Maximum: 32768	
Description: Number of values for recursive averaging. This parameter is only used at recursive average.	
Parameter: int32_t SP_ComputationMedianCount	SP_ComputationMedianCount
Direction: Down	
Valid values:	
3, 5, 7, 9	
Description: Number of values to build median. This parameter is only used at median.	

Parameter: double SP_ComputationFactor1	SP_ComputationFactor1
Direction: Down	
Valid values:	
Minimum: -32767.0	
Maximum: 32767.0	
Description: Multiplication factor for first signal (if operation is calculation).	
Parameter: int32_t SP_ComputationSignalIndex1	SP_ComputationSignalIndex1
Direction: Down	
Valid values:	
-1= Use SP_ComputationSignalName1	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: First signal index (if operation is calculation or thickness).	
Parameter: String SP_ComputationSignalName1	SP_ComputationSignalName1
Direction: Down	
Description: User defined first signal (if operation is calculation or thickness).	
Parameter: double SP_ComputationFactor2	SP_ComputationFactor2
Direction: Down	
Valid values:	
Minimum: -32767.0	
Maximum: 32767.0	
Description: Multiplication factor for second signal (if operation is calculation).	
Parameter: int32_t SP_ComputationSignalIndex2	SP_ComputationSignalIndex2
Direction: Down	
Valid values:	
-1= Use SP_ComputationSignalName2	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: Second signal index (if operation is calculation or thickness).	

Parameter: String SP_ComputationSignalName2	SP_ComputationSignalName2
Direction: Down	
Description: User defined second signal (if operation is calculation or thickness).	
Parameter: double SP_ComputationOffset	SP_ComputationOffset
Direction: Down	
Valid values:	
Minimum: -2147.0	
Maximum: 2147.0	
Unit: mm	
Description: Additional offset (if operation is calculation).	
Parameter: String SP_ComputationName	SP_ComputationName
Direction: Down	
Description: Name of the newly created signal (if operation is calculation, thickness or copy).	

11.2.2.5.15 Get_Computation (COMP)

Get customised measurement data processing computations.

Parameter: int32_t SP_ComputationQueue	SP_ComputationQueue
Direction: Down	
Valid values:	
-1= Any queue	
0= Channel 1	
1= Channel 2 (only available at IFD2422)	
2= System (only available at IFD2422)	
Default: -1	
Description: Computation queue	
Parameter: int32_t SP_ComputationID	SP_ComputationID
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Index within computation queue. Ignored if SP_Computation-Queue is -1.	
Parameter: String SA_ComputationTable	SA_ComputationTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_ComputationEntries	SA_ComputationEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 30	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_ComputationEntry1, SA_ComputationEn-try2, ...	

Parameter: String SA_ComputationEntry1..x	SA_ComputationEntry1..x
Direction: Up	
Description: Each computation as complete line.	
Parameter: int32_t SA_ComputationQueue1..x	SA_ComputationQueue1..x
Direction: Up	
Valid values:	
0= Channel 1	
1= Channel 2 (only available at IFD2422)	
2= System (only available at IFD2422)	
Description: Computation queue	
Parameter: int32_t SA_ComputationID1..x	SA_ComputationID1..x
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Index within computation queue	
Parameter: int32_t SA_ComputationOperation1..x	SA_ComputationOperation1..x
Direction: Up	
Valid values:	
0= None	
1= Median	
2= Moving average	
3= Recursive averaging	
4= Calculation	
5= Thickness	
6= Copy	
Description: Computation operation	
Parameter: int32_t SA_ComputationSignalIndex1..x	SA_ComputationSignalIndex1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_ComputationSignalName1..x	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: Signal index (if operation is averaging or copy).	
Parameter: String SA_ComputationSignalName1..x	SA_ComputationSignalName1..x
Direction: Up	
Description: User defined signal (if operation is averaging or copy).	

Parameter: int32_t SA_ComputationMovingCount <code>1..x</code>	SA_ComputationMoving- Count <code>1..x</code>
Direction: Up	
Valid values:	
2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096	
Description: Number of value for the averaging window. This parameter is only used at moving average.	
Parameter: int32_t SA_ComputationRecursiveCount <code>1..x</code>	SA_ComputationRecursive- Count <code>1..x</code>
Direction: Up	
Valid values:	
Minimum: 2	
Maximum: 32768	
Description: Number of values for recursive averaging. This parameter is only used at recursive average.	
Parameter: int32_t SA_ComputationMedianCount <code>1..x</code>	SA_ComputationMedian- Count <code>1..x</code>
Direction: Up	
Valid values:	
3, 5, 7, 9	
Description: Number of values to build median. This parameter is only used at median.	
Parameter: double SA_ComputationFactor <code>1..x</code>	SA_ComputationFactor <code>1..x</code>
Direction: Up	
Valid values:	
Minimum: -32767.0	
Maximum: 32767.0	
Description: Multiplication factor for first signal (if operation is calculation).	
Parameter: int32_t SA_ComputationSignalIndex <code>1..x</code>	SA_ComputationSignalIn- dex <code>1..x</code>
Direction: Up	
Valid values:	
-1= User defined name at SA_ComputationSignalName <code>1..x</code>	
0= Distance 1 of first Channel	
1= Distance 2 of first Channel	
2= Distance 3 of first Channel	
3= Distance 4 of first Channel	
4= Distance 5 of first Channel	
5= Distance 6 of first Channel	
6= Distance 1 of second Channel (only available at IFD2422)	
7= Distance 2 of second Channel (only available at IFD2422)	
8= Distance 3 of second Channel (only available at IFD2422)	
9= Distance 4 of second Channel (only available at IFD2422)	
10= Distance 5 of second Channel (only available at IFD2422)	
11= Distance 6 of second Channel (only available at IFD2422)	
Description: First signal index (if operation is calculation or thickness).	
Parameter: String SA_ComputationSignalName <code>1..x</code>	SA_ComputationSignal- Name <code>1..x</code>
Direction: Up	
Description: User defined first signal (if operation is calculation or thickness).	

Parameter: double SA_ComputationFactor2_1..x	SA_ComputationFactor2_1..x
Direction: Up	
Valid values:	
Minimum: -32767.0	
Maximum: 32767.0	
Description: Multiplication factor for second signal (if operation is calculation).	
Parameter: int32_t SA_ComputationSignalIndex2_1..x	SA_ComputationSignalIndex2_1..x
Direction: Up	
Valid values:	
-1 = User defined name at SA_ComputationSignalName2_1..x	
0 = Distance 1 of first Channel	
1 = Distance 2 of first Channel	
2 = Distance 3 of first Channel	
3 = Distance 4 of first Channel	
4 = Distance 5 of first Channel	
5 = Distance 6 of first Channel	
6 = Distance 1 of second Channel (only available at IFD2422)	
7 = Distance 2 of second Channel (only available at IFD2422)	
8 = Distance 3 of second Channel (only available at IFD2422)	
9 = Distance 4 of second Channel (only available at IFD2422)	
10 = Distance 5 of second Channel (only available at IFD2422)	
11 = Distance 6 of second Channel (only available at IFD2422)	
Description: Second signal index (if operation is calculation or thickness).	
Parameter: String SA_ComputationSignalName2_1..x	SA_ComputationSignalName2_1..x
Direction: Up	
Description: User defined second signal (if operation is calculation or thickness).	
Parameter: double SA_ComputationOffset1..x	SA_ComputationOffset1..x
Direction: Up	
Valid values:	
Minimum: -2147.0	
Maximum: 2147.0	
Unit: mm	
Description: Additional offset (if operation is calculation).	
Parameter: String SA_ComputationName1..x	SA_ComputationName1..x
Direction: Up	
Description: Name of the newly created signal (if operation is calculation, thickness or copy).	

11.2.2.5.16 Set_SystemSignalRange (SYSSIGNALRANGE)

Set the output value range in mm for signals of COMP SYS.

Parameter: double SP_SystemSignalRangeLowerLimit	SP_SystemSignalRangeLowerLimit
Direction: Down	
Valid values:	
Minimum: -2147.0	
Maximum: 2147.0	
Unit: mm	
Description: Value which represents lowest raw value.	

Parameter: int32_t SP_OutputAnalog	SP_OutputAnalog
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description: Output active at analog interface.	
Parameter: int32_t SP_OutputEthernet	SP_OutputEthernet
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description: Output active at ethernet interface.	
Parameter: int32_t SP_OutputErrorOut	SP_OutputErrorOut
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description: Output active at error outputs.	

11.2.3.1.2 Get_DataOutInterface (OUTPUT)

Get the active interface for data output.

Parameter: int32_t SA_OutputRS422	SA_OutputRS422
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at RS422 interface.	
Parameter: int32_t SA_OutputAnalog	SA_OutputAnalog
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at analog interface.	
Parameter: int32_t SA_OutputEthernet	SA_OutputEthernet
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at ethernet interface.	
Parameter: int32_t SA_OutputErrorOut	SA_OutputErrorOut
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at error outputs.	

11.2.3.1.3 Set_ResamplingDevice (OUTREDUCEDEVICE)

Set the decives for which resampling is active.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_ResampleRS422

SP_ResampleRS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if RS422 output should be resampled.

Parameter: int32_t SP_ResampleAnalog

SP_ResampleAnalog

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if analog output should be resampled.

Parameter: int32_t SP_ResampleEthernet

SP_ResampleEthernet

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if ethernet output should be resampled.

11.2.3.1.4 Get_ResamplingDevice (OUTREDUCEDEVICE)

Get the decives for which resampling is active.

Parameter: int32_t SA_ResampleRS422

SA_ResampleRS422

Direction: Up

Valid values:

0= no

1= yes

Description: RS422 output is resampled.

Parameter: int32_t SA_ResampleAnalog

SA_ResampleAnalog

Direction: Up

Valid values:

0= no

1= yes

Description: Analog output is resampled.

Parameter: int32_t SA_ResampleEthernet

SA_ResampleEthernet

Direction: Up

Valid values:

0= no

1= yes

Description: Ethernet output is resampled.

11.2.3.1.5 Set_ResamplingCount (OUTREDUCECOUNT)

Set reduce count for resampling.

Parameter: int32_t SP_Resampling SP_Resampling
Direction: Down
Valid values:
Minimum: 1
Maximum: 3000000
Description: Resampling value.

11.2.3.1.6 Get_ResamplingCount (OUTREDUCECOUNT)

Get reduce count for resampling.

Parameter: int32_t SA_Resampling SA_Resampling
Direction: Up
Valid values:
Minimum: 1
Maximum: 3000000
Description: Resampling value.

11.2.3.1.7 Set_HoldLastValid (OUTHOLD)

Set the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SP_HoldLastValid SP_HoldLastValid
Direction: Down
Valid values:
Minimum: -1
Maximum: 1024
Description: Values to replace by last valid value. -1 means no value to hold,
 0 means never output an error value (always hold last valid value).

11.2.3.1.8 Get_HoldLastValid (OUTHOLD)

Get the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SA_HoldLastValid SA_HoldLastValid
Direction: Up
Valid values:
Minimum: -1
Maximum: 1024
Description: Values to replace by last valid value. -1 means no value to hold,
 0 means never output an error value (always hold last valid value).

11.2.3.1.9 Set_FramesPerPacket_ETH (MEASCNT_ETH)

Set the maximum number of frames in ethernet packet.

Parameter: int32_t SP_FramesPerPacket_ETH

SP_FramesPerPacket_ETH

Direction: Down

Valid values:

Minimum: 0

Maximum: 350

Description: Maximum number of frames in ethernet packet. 0 means automatic.

11.2.3.1.10 Get_FramesPerPacket_ETH (MEASCNT_ETH)

Get the maximum number of frames in ethernet packet.

Parameter: int32_t SA_FramesPerPacket_ETH

SA_FramesPerPacket_ETH

Direction: Up

Valid values:

Minimum: 0

Maximum: 350

Description: Maximum number of frames in ethernet packet. 0 means automatic.

11.2.3.2 Selected measurement values

11.2.3.2.1 Set_Output_RS422 (OUT_RS422)

Set the data to be output at RS422 interface.

For this command an [Update...](#) and a [Reset...](#) meta command is available.

Parameter: int32_t SP_OutputVideoRaw_Ch1_RS422

SP_OutputVideoRaw_Ch1_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if raw video signal of first channel is transmitted.

Parameter: int32_t SP_OutputVideoDark_Ch1_RS422

SP_OutputVideoDark_Ch1_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if dark corrected signal of first channel is transmitted.

Parameter: int32_t SP_OutputVideoLight_Ch1_RS422

SP_OutputVideoLight_Ch1_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if light corrected video signal of first channel is transmitted.

Parameter: int32_t SP_OutputVideoRaw_Ch2_RS422	SP_OutputVideoRaw_Ch2_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if raw video signal of second channel is transmitted.
Parameter: int32_t SP_OutputVideoDark_Ch2_RS422	SP_OutputVideoDark_Ch2_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if dark corrected signal of second channel is transmitted.
Parameter: int32_t SP_OutputVideoLight_Ch2_RS422	SP_OutputVideoLight_Ch2_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if light corrected video signal of second channel is transmitted.
Parameter: int32_t SP_OutputDistance1..6_Ch1_RS422	SP_OutputDistance1..6_- Ch1_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if distance 1 to 6 of first channel is transmitted.
Parameter: int32_t SP_OutputDistance1..6_Ch2_RS422	SP_OutputDistance1..6_- Ch2_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if distance 1 to 6 of second channel is transmitted.
Parameter: int32_t SP_OutputUnlinearizedDistance_Ch1_RS422	SP_OutputUnlinearizedDis- tance_Ch1_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if unlinearized distance of first channel is transmitted.

Parameter: int32_t SP_OutputUnlinearizedDistance_Ch2_RS422	SP_OutputUnlinearizedDistance_Ch2_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if unlinearized distance of second channel is transmitted.
Parameter: int32_t SP_OutputAdditionalShutterTime_Ch1_RS422	SP_OutputAdditionalShutterTime_Ch1_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if shutter time of first channel is transmitted.
Parameter: int32_t SP_OutputAdditionalEncoder1_Ch1_RS422	SP_OutputAdditionalEncoder1_Ch1_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if encoder 1 of first channel is transmitted.
Parameter: int32_t SP_OutputAdditionalEncoder2_Ch1_RS422	SP_OutputAdditionalEncoder2_Ch1_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if encoder 2 of first channel is transmitted.
Parameter: int32_t SP_OutputAdditionalIntensity_Ch1_RS422	SP_OutputAdditionalIntensity_Ch1_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if intensity of first channel is transmitted.
Parameter: int32_t SP_OutputAdditionalShutterTime_Ch2_RS422	SP_OutputAdditionalShutterTime_Ch2_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if shutter time of second channel is transmitted.
Parameter: int32_t SP_OutputAdditionalEncoder1_Ch2_RS422	SP_OutputAdditionalEncoder1_Ch2_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if encoder 1 of second channel is transmitted.

Parameter: int32_t SP_OutputAdditionalEncoder2_Ch2_RS422	SP_OutputAdditionalEncoder2_Ch2_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if encoder 2 of second channel is transmitted.
Parameter: int32_t SP_OutputAdditionalIntensity_Ch2_RS422	SP_OutputAdditionalIntensity_Ch2_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if intensity of second channel is transmitted.
Parameter: int32_t SP_OutputAdditionalMeasrate_RS422	SP_OutputAdditionalMeasrate_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if measurement rate is transmitted.
Parameter: int32_t SP_OutputAdditionalTrgTimeDiff_RS422	SP_OutputAdditionalTrgTimeDiff_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if trigger time difference is transmitted.
Parameter: int32_t SP_OutputAdditionalTimestamp_RS422	SP_OutputAdditionalTimestamp_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp is transmitted.
Parameter: int32_t SP_OutputAdditionalTimestampLo_RS422	SP_OutputAdditionalTimestampLo_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp (lower 16 bit) is transmitted.
Parameter: int32_t SP_OutputAdditionalTimestampHi_RS422	SP_OutputAdditionalTimestampHi_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp (upper 16 bit) is transmitted.

Parameter: int32_t SP_OutputAdditionalCounter_RS422	SP_OutputAdditionalCounter_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: String SP_OutputCalculation_RS422	SP_OutputCalculation_RS422
Direction: Down	
Description: List of calculated signals (generated by Set_Computation). The entries must be separated by spaces.	
Parameter: String SP_OutputStatistic_RS422	SP_OutputStatistic_RS422
Direction: Down	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries must be separated by spaces.	

11.2.3.2.2 Get_Output_RS422 (OUT_RS422)

Get the data which is output at RS422 interface.

Parameter: int32_t SA_OutputVideoRaw_Ch1_RS422	SA_OutputVideoRaw_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoDark_Ch1_RS422	SA_OutputVideoDark_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if dark corrected signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoLight_Ch1_RS422	SA_OutputVideoLight_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if light corrected video signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoRaw_Ch2_RS422	SA_OutputVideoRaw_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if raw video signal of second channel is transmitted.	

Parameter: int32_t SA_OutputVideoDark_Ch2_RS422	SA_OutputVideoDark_Ch2_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if dark corrected signal of second channel is transmitted.
Parameter: int32_t SA_OutputVideoLight_Ch2_RS422	SA_OutputVideoLight_Ch2_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if light corrected video signal of second channel is transmitted.
Parameter: int32_t SA_OutputDistance1..6_Ch1_RS422	SA_OutputDistance1..6_- Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if distance 1 to 6 of first channel is transmitted.
Parameter: int32_t SA_OutputDistance1..6_Ch2_RS422	SA_OutputDistance1..6_- Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if distance 1 to 6 of second channel is transmitted.
Parameter: int32_t SA_OutputUnlinearizedDistance_Ch1_RS422	SA_OutputUnlinearizedDis- tance_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if unlinearized distance of first channel is transmitted.
Parameter: int32_t SA_OutputUnlinearizedDistance_Ch2_RS422	SA_OutputUnlinearizedDis- tance_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if unlinearized distance of second channel is transmitted.

Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_RS422	SA_OutputAdditionalShutterTime_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1_Ch1_RS422	SA_OutputAdditionalEncoder1_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder2_Ch1_RS422	SA_OutputAdditionalEncoder2_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 2 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_Ch1_RS422	SA_OutputAdditionalIntensity_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch2_RS422	SA_OutputAdditionalShutterTime_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if shutter time of second channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1_Ch2_RS422	SA_OutputAdditionalEncoder1_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if encoder 1 of second channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder2_Ch2_RS422	SA_OutputAdditionalEncoder2_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if encoder 2 of second channel is transmitted.	

Parameter: int32_t SA_OutputAdditionalIntensity_Ch2_RS422	SA_OutputAdditionalIntensity_Ch2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if intensity of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalMeasrate_RS422	SA_OutputAdditionalMeasrate_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if measurement rate is transmitted.
Parameter: int32_t SA_OutputAdditionalTrgTimeDiff_RS422	SA_OutputAdditionalTrgTimeDiff_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if trigger time difference is transmitted.
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp is transmitted.
Parameter: int32_t SA_OutputAdditionalTimestampLo_RS422	SA_OutputAdditionalTimestampLo_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp (lower 16 bit) is transmitted.
Parameter: int32_t SA_OutputAdditionalTimestampHi_RS422	SA_OutputAdditionalTimestampHi_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp (upper 16 bit) is transmitted.
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if counter is transmitted.
Parameter: String SA_OutputCalculation_RS422	SA_OutputCalculation_RS422
Direction: Up	
Description:	List of calculated signals (generated by Set_Computation). The entries are separated by spaces.

Parameter: String SA_OutputStatistic_RS422 SA_OutputStatistic_RS422
Direction: Up
Description: List of statistic signals (generated by [Set_StatisticSignal](#)). The entries are separated by spaces.

11.2.3.2.3 Set_Output_ETH (OUT_ETH)

Set the data to be output at ethernet interface.
 For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputVideoRaw_Ch1_ETH SP_OutputVideoRaw_Ch1_ETH
Direction: Down
Valid values:
 0 = no
 1 = yes
Description: Specify if raw video signal of first channel is transmitted.

Parameter: int32_t SP_OutputVideoDark_Ch1_ETH SP_OutputVideoDark_Ch1_-ETH
Direction: Down
Valid values:
 0 = no
 1 = yes
Description: Specify if dark corrected signal of first channel is transmitted.

Parameter: int32_t SP_OutputVideoLight_Ch1_ETH SP_OutputVideoLight_Ch1_-ETH
Direction: Down
Valid values:
 0 = no
 1 = yes
Description: Specify if light corrected video signal of first channel is transmitted.

Parameter: int32_t SP_OutputVideoRaw_Ch2_ETH SP_OutputVideoRaw_Ch2_ETH
Direction: Down
Valid values:
 0 = no
 1 = yes
Valid for sensor:
 IFD2422
Description: Specify if raw video signal of second channel is transmitted.

Parameter: int32_t SP_OutputVideoDark_Ch2_ETH SP_OutputVideoDark_Ch2_-ETH
Direction: Down
Valid values:
 0 = no
 1 = yes
Valid for sensor:
 IFD2422
Description: Specify if dark corrected signal of second channel is transmitted.

Parameter: int32_t SP_OutputVideoLight_Ch2_ETH	SP_OutputVideoLight_Ch2_-ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if light corrected video signal of second channel is transmitted.	
Parameter: int32_t SP_OutputDistance1..6_Ch1_ETH	SP_OutputDistance1..6_-Ch1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 to 6 of first channel is transmitted.	
Parameter: int32_t SP_OutputDistance1..6_Ch2_ETH	SP_OutputDistance1..6_-Ch2_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if distance 1 to 6 of second channel is transmitted.	
Parameter: int32_t SP_OutputUnlinearizedDistance_Ch1_ETH	SP_OutputUnlinearizedDistance_Ch1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if unlinearized distance of first channel is transmitted.	
Parameter: int32_t SP_OutputUnlinearizedDistance_Ch2_ETH	SP_OutputUnlinearizedDistance_Ch2_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if unlinearized distance of second channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalShutterTime_Ch1_ETH	SP_OutputAdditionalShutterTime_Ch1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time of first channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalEncoder1_Ch1_ETH	SP_OutputAdditionalEncoder1_Ch1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 of first channel is transmitted.	

Parameter: int32_t SP_OutputAdditionalEncoder2_Ch1_ETH	SP_OutputAdditionalEncoder2_Ch1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 2 of first channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalIntensity_Ch1_ETH	SP_OutputAdditionalIntensity_Ch1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity of first channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalShutterTime_Ch2_ETH	SP_OutputAdditionalShutterTime_Ch2_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if shutter time of second channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalEncoder1_Ch2_ETH	SP_OutputAdditionalEncoder1_Ch2_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if encoder 1 of second channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalEncoder2_Ch2_ETH	SP_OutputAdditionalEncoder2_Ch2_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if encoder 2 of second channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalIntensity_Ch2_ETH	SP_OutputAdditionalIntensity_Ch2_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if intensity of second channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalMeasrate_ETH	SP_OutputAdditionalMeasrate_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measurement rate is transmitted.	

Parameter: int32_t SP_OutputAdditionalTrgTimeDiff_ETH	SP_OutputAdditionalTrgTimeDiff_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger time difference is transmitted.	
Parameter: int32_t SP_OutputAdditionalTimestamp_ETH	SP_OutputAdditionalTimestamp_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SP_OutputAdditionalTimestampLo_ETH	SP_OutputAdditionalTimestampLo_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp (lower 16 bit) is transmitted.	
Parameter: int32_t SP_OutputAdditionalTimestampHi_ETH	SP_OutputAdditionalTimestampHi_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp (upper 16 bit) is transmitted.	
Parameter: int32_t SP_OutputAdditionalCounter_ETH	SP_OutputAdditionalCounter_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: String SP_OutputCalculation_ETH	SP_OutputCalculation_ETH
Direction: Down	
Description: List of calculated signals (generated by Set_Calculation). The entries must be separated by spaces.	
Parameter: String SP_OutputStatistic_ETH	SP_OutputStatistic_ETH
Direction: Down	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries must be separated by spaces.	

11.2.3.2.4 Get_Output_ETH (OUT_ETH)

Get the data which is output at ethernet interface.

Parameter: int32_t SA_OutputVideoRaw_Ch1_ETH	SA_OutputVideoRaw_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal of first channel is transmitted.	

Parameter: int32_t SA_OutputVideoDark_Ch1_ETH	SA_OutputVideoDark_Ch1_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if dark corrected signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoLight_Ch1_ETH	SA_OutputVideoLight_Ch1_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if light corrected video signal of first channel is transmitted.	
Parameter: int32_t SA_OutputVideoRaw_Ch2_ETH	SA_OutputVideoRaw_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if raw video signal of second channel is transmitted.	
Parameter: int32_t SA_OutputVideoDark_Ch2_ETH	SA_OutputVideoDark_Ch2_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if dark corrected signal of second channel is transmitted.	
Parameter: int32_t SA_OutputVideoLight_Ch2_ETH	SA_OutputVideoLight_Ch2_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description: Specify if light corrected video signal of second channel is transmitted.	
Parameter: int32_t SA_OutputDistance1..6_Ch1_ETH	SA_OutputDistance1..6_- Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 to 6 of first channel is transmitted.	

Parameter: int32_t SA_OutputDistance1..6_Ch2_ETH	SA_OutputDistance1..6_-Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if distance 1 to 6 of second channel is transmitted.
Parameter: int32_t SA_OutputUnlinearizedDistance_Ch1_ETH	SA_OutputUnlinearizedDistance_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if unlinearized distance of first channel is transmitted.
Parameter: int32_t SA_OutputUnlinearizedDistance_Ch2_ETH	SA_OutputUnlinearizedDistance_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if unlinearized distance of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_ETH	SA_OutputAdditionalShutterTime_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if shutter time of first channel is transmitted.
Parameter: int32_t SA_OutputAdditionalEncoder1_Ch1_ETH	SA_OutputAdditionalEncoder1_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if encoder 1 of first channel is transmitted.
Parameter: int32_t SA_OutputAdditionalEncoder2_Ch1_ETH	SA_OutputAdditionalEncoder2_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if encoder 2 of first channel is transmitted.
Parameter: int32_t SA_OutputAdditionalIntensity_Ch1_ETH	SA_OutputAdditionalIntensity_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if intensity of first channel is transmitted.

Parameter: int32_t SA_OutputAdditionalShutterTime_Ch2_ETH	SA_OutputAdditionalShutterTime_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if shutter time of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalEncoder1_Ch2_ETH	SA_OutputAdditionalEncoder1_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if encoder 1 of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalEncoder2_Ch2_ETH	SA_OutputAdditionalEncoder2_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if encoder 2 of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalIntensity_Ch2_ETH	SA_OutputAdditionalIntensity_Ch2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2422	
Description:	Specify if intensity of second channel is transmitted.
Parameter: int32_t SA_OutputAdditionalMeasrate_ETH	SA_OutputAdditionalMeasrate_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if measurement rate is transmitted.
Parameter: int32_t SA_OutputAdditionalTrgTimeDiff_ETH	SA_OutputAdditionalTrgTimeDiff_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if trigger time difference is transmitted.
Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if timestamp is transmitted.

Parameter: int32_t SA_OutputAdditionalTimestampLo_ETH	SA_OutputAdditionalTimestampLo_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp (lower 16 bit) is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestampHi_ETH	SA_OutputAdditionalTimestampHi_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp (upper 16 bit) is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditionalCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: String SA_OutputCalculation_ETH	SA_OutputCalculation_ETH
Direction: Up	
Description: List of calculated signals (generated by Set_Calculation). The entries are separated by spaces.	
Parameter: String SA_OutputStatistic_ETH	SA_OutputStatistic_ETH
Direction: Up	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries are separated by spaces.	

11.2.3.3 Switching outputs

11.2.3.3.1 Set_ErrorOutput<n> (ERROROUT<n>)

Set condition to be used to set error output n ∈ {1..2}.

Parameter: int32_t SP_ErrorOutput<n>	SP_ErrorOutput<n>
Direction: Down	
Valid values:	
0= Channel 1 Intensity faults (01ER1)	
1= Channel 1 Out of range values (01ER2)	
2= 01ER1 or 01ER2 (01ER12)	
3= Channel 2 Intensity faults, only available at IFD2422 (02ER1)	
4= Channel 2 Out of range values, only available at IFD2422 (02ER2)	
5= 02ER1 or 02ER2, only available at IFD2422 (02ER12)	
6= 01ER1 or 01ER2 or 02ER1 or 02ER2, only available at IFD2422 (0102ER12)	
7= Works as limit switch with the configuration of ERRORLIMIT<n> (ERRORLIMIT)	
8= Falling below the intensity minimum channel 1, only available at Option 217 (01ER3)	
9= Falling below the intensity minimum channel 2, only available at IFD2422 Option 217 (02ER3)	
10= 01ER3 or 02ER3, only available at IFD2422 Option 217 (0102ER3)	
Description: Condition for error output.	

11.2.3.3.2 Get_ErrorOutput<n> (ERROROUT<n>)

Get condition to be used to set error output n ∈ {1..2}.

Parameter: int32_t SA_ErrorOutput<n>

SA_ErrorOutput<n>

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Channel 1 Intensity faults (01ER1)
- 1= Channel 1 Out of range values (01ER2)
- 2= 01ER1 or 01ER2 (01ER12)
- 3= Channel 2 Intensity faults, only available at IFD2422 (02ER1)
- 4= Channel 2 Out of range values, only available at IFD2422 (02ER2)
- 5= 02ER1 or 02ER2, only available at IFD2422 (02ER12)
- 6= 01ER1 or 01ER2 or 02ER1 or 02ER2, only available at IFD2422 (0102ER12)
- 7= Works as limit switch with the configuration of ERRORLIMIT<n> (ERRORLIMIT)
- 8= Falling below the intensity minimum channel 1, only available at Option 217 (01ER3)
- 9= Falling below the intensity minimum channel 2, only available at IFD2422 Option 217 (02ER3)
- 10= 01ER3 or 02ER3, only available at IFD2422 Option 217 (0102ER3)

Description: Condition for error output.

11.2.3.3.3 Set_ErrorLimitSignal<n> (ERRORLIMITSIGNAL<n>)

Set the signal for error limit n ∈ {1..2}.

Parameter: int32_t SP_ErrorLimitSignalIndex<n>

SP_ErrorLimitSignalIndex<n>

Direction: Down

Valid values:

- 1= Use SP_ErrorLimitSignalName<n>
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel
- 2= Distance 3 of first Channel
- 3= Distance 4 of first Channel
- 4= Distance 5 of first Channel
- 5= Distance 6 of first Channel
- 6= Distance 1 of second Channel (only available at IFD2422)
- 7= Distance 2 of second Channel (only available at IFD2422)
- 8= Distance 3 of second Channel (only available at IFD2422)
- 9= Distance 4 of second Channel (only available at IFD2422)
- 10= Distance 5 of second Channel (only available at IFD2422)
- 11= Distance 6 of second Channel (only available at IFD2422)

Description: Signal index to be checked.

Parameter: String SP_ErrorLimitSignalName<n>

SP_ErrorLimitSignalName<n>

Direction: Down

Description: User defined signal which is used for error limit.

11.2.3.3.4 Get_ErrorLimitSignal<n> (ERRORLIMITSIGNAL<n>)

Get the signal for error limit n $\in \{1..2\}$.

Parameter: int32_t SA_ErrorLimitSignalIndex<n>

Direction: Up

Valid values:

- 1 = User defined name at SA_ErrorLimitSignalName<n>
- 0 = Distance 1 of first Channel
- 1 = Distance 2 of first Channel
- 2 = Distance 3 of first Channel
- 3 = Distance 4 of first Channel
- 4 = Distance 5 of first Channel
- 5 = Distance 6 of first Channel
- 6 = Distance 1 of second Channel (only available at IFD2422)
- 7 = Distance 2 of second Channel (only available at IFD2422)
- 8 = Distance 3 of second Channel (only available at IFD2422)
- 9 = Distance 4 of second Channel (only available at IFD2422)
- 10 = Distance 5 of second Channel (only available at IFD2422)
- 11 = Distance 6 of second Channel (only available at IFD2422)

Description: Signal index to be checked.

SA_ErrorLimitSignalIndex<n>

Parameter: String SA_ErrorLimitSignalName<n>

Direction: Up

Description: User defined signal which is used for error limit.

SA_ErrorLimitSignalName<n>

11.2.3.3.5 Set_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)

Set the error limits n $\in \{1..2\}$.

Parameter: double SP_LowerLimit<n>

SP_LowerLimit<n>

Direction: Down

Valid values:

- Minimum:** -2147.0
Maximum: 2147.0

Unit: mm

Description: Lower limit.

Parameter: double SP_UpperLimit<n>

SP_UpperLimit<n>

Direction: Down

Valid values:

- Minimum:** -2147.0
Maximum: 2147.0

Unit: mm

Description: Upper limit.

11.2.3.3.6 Get_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)

Get the error limits n $\in \{1..2\}$.

Parameter: double SA_LowerLimit<n>

SA_LowerLimit<n>

Direction: Up

Valid values:

- Minimum:** -2147.0
Maximum: 2147.0

Unit: mm

Description: Lower limit.

Parameter: double SA_UpperLimit<n> SA_UpperLimit<n>
Direction: Up
Valid values:
 Minimum: -2147.0
 Maximum: 2147.0
Unit: mm
Description: Upper limit.

11.2.3.3.7 Set_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)

Set the compare operation for limit n ∈ {1..2}.

Parameter: int32_t SP_ErrorLimitCompOp<n> SP_ErrorLimitCompOp<n>
Direction: Down
Valid values:
 0= Lower
 1= Upper
 2= Both
Description: Compare operation

11.2.3.3.8 Get_ErrorLimitCompOp<n> (ERRORLIMITCOMPARETO<n>)

Get the compare operation for limit n ∈ {1..2}.

Parameter: int32_t SA_ErrorLimitCompOp<n> SA_ErrorLimitCompOp<n>
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Lower
 1= Upper
 2= Both
Description: Compare operation

11.2.3.3.9 Set_ErrorLevelOut<n> (ERRORLEVELOUT<n>)

Set level of error output n ∈ {1..2} on error.

Parameter: int32_t SP_ErrorLevelOut<n> SP_ErrorLevelOut<n>
Direction: Down
Valid values:
 0= NPN
 1= PNP
 2= Push-Pull (PUSHPULL)
 3= Push-Pull negated (PUSHPULLNEG)
Description: Error level.

11.2.3.3.10 Get_ErrorLevelOut<n> (ERRORLEVELOUT<n>)

Get level of error output n $\in \{1..2\}$ on error.

Parameter: int32_t SA_ErrorLevelOut<n>

SA_ErrorLevelOut<n>

Direction: Up

Valid values:

- 0= NPN
- 1= PNP
- 2= Push-Pull (PUSHPULL)
- 3= Push-Pull negated (PUSHPULLNEG)

Description: Error level.

11.2.3.4 Analog output

11.2.3.4.1 Set_AnalogOutput (ANALOGOUT)

Set the data to be used for analog output.

Parameter: int32_t SP_AnalogOutputSignalIndex

SP_AnalogOutputSignalIndex

Direction: Down

Valid values:

- 1= Use SP_AnalogOutputSignalName
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel
- 2= Distance 3 of first Channel
- 3= Distance 4 of first Channel
- 4= Distance 5 of first Channel
- 5= Distance 6 of first Channel
- 6= Distance 1 of second Channel (only available at IFD2422)
- 7= Distance 2 of second Channel (only available at IFD2422)
- 8= Distance 3 of second Channel (only available at IFD2422)
- 9= Distance 4 of second Channel (only available at IFD2422)
- 10= Distance 5 of second Channel (only available at IFD2422)
- 11= Distance 6 of second Channel (only available at IFD2422)

Description: Data to be used for analog output.

Parameter: String SP_AnalogOutputSignalName

SP_AnalogOutputSignalName

Direction: Down

Description: User defined signal which is used for analog output.

11.2.3.4.2 Get_AnalogOutput (ANALOGOUT)

Get the data to be used for analog output.

Parameter: int32_t SA_AnalogOutputSignalIndex

SA_AnalogOutputSignalIndex

Direction: Up

Valid values:

- 1= User defined name at SA_AnalogOutputSignalName
- 0= Distance 1 of first Channel
- 1= Distance 2 of first Channel

2= Distance 3 of first Channel
 3= Distance 4 of first Channel
 4= Distance 5 of first Channel
 5= Distance 6 of first Channel
 6= Distance 1 of second Channel (only available at IFD2422)
 7= Distance 2 of second Channel (only available at IFD2422)
 8= Distance 3 of second Channel (only available at IFD2422)
 9= Distance 4 of second Channel (only available at IFD2422)
 10= Distance 5 of second Channel (only available at IFD2422)
 11= Distance 6 of second Channel (only available at IFD2422)

Description: Data to be used for analog output.

Parameter: String SA_AnalogOutputSignalName

SA_AnalogOutputSignalName

Direction: Up

Description: User defined signal which is used for analog output.

11.2.3.4.3 Set_AnalogRange (ANALOG RANGE)

Set the analog output range.

Parameter: int32_t SP_AnalogRange

SP_AnalogRange

Direction: Down

Valid values:

1= 0 - 5V
 2= 0 - 10V
 5= 4 - 20mA

Description: Analog output range.

11.2.3.4.4 Get_AnalogRange (ANALOG RANGE)

Get the analog output range.

Parameter: int32_t SA_AnalogRange

SA_AnalogRange

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
 1= 0 - 5V
 2= 0 - 10V
 5= 4 - 20mA

Description: Analog output range.

11.2.3.4.5 Set_AnalogScaleMode (ANALOG SCALE MODE)

Set the scaling mode for analog output.

Parameter: int32_t SP_AnalogScaleMode

SP_AnalogScaleMode

Direction: Down

Valid values:

0= Standard
 1= Two point (TWOPOINT)

Description: Analog scale mode.

11.2.3.4.6 Get_AnalogScaleMode (ANALOGSCALEMODE)

Get the scaling mode for analog output.

Parameter: int32_t SA_AnalogScaleMode

SA_AnalogScaleMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Standard
- 1 = Two point (TWOPOINT)

Description: Analog scale mode.

11.2.3.4.7 Set_AnalogScaleRange (ANALOGSCALERANGE)

Set the scaling factors for analog output.

Parameter: double SP_AnalogScaleRangeLowerLimit

SP_AnalogScaleRangeLower-
Limit

Direction: Down

Valid values:

- Minimum:** -2147.0
- Maximum:** 2147.0

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SP_AnalogScaleRangeUpperLimit

SP_AnalogScaleRangeUpper-
Limit

Direction: Down

Valid values:

- Minimum:** -2147.0
- Maximum:** 2147.0

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

11.2.3.4.8 Get_AnalogScaleRange (ANALOGSCALERANGE)

Get the scaling factor for analog output.

Parameter: double SA_AnalogScaleRangeLowerLimit

SA_AnalogScaleRangeLower-
Limit

Direction: Up

Valid values:

- Minimum:** -2147.0
- Maximum:** 2147.0

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SA_AnalogScaleRangeUpperLimit

SA_AnalogScaleRangeUpper-
Limit

Direction: Up

Valid values:

- Minimum:** -2147.0
- Maximum:** 2147.0

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

11.2.4 Internal commands

11.2.4.1 Upload_CalibTable

Send a calibration table for selected sensor to the controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SP_CalibTable

SP_CalibTable

Direction: Down

Description: Calibration table for selected sensor.

11.3 Commands for IFD2445, IFD2451, IFD2461 and IFD2471

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- RS232 (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
- IF2004 (native).
- TCP/IP (native).
- IF2004_USB (native).
- IF2008 (native).
- IF2008_ETH (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_AllParameters](#) (SP_Additional= 1) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate, to interpret and scale data and to assign values. If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls optionally sensor command [Set_IPDataTransferMode](#) and [Set_DataOutInterface](#) at [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_AllParameters](#) (SP_Additional= 1)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

11.3.1 General commands

11.3.1.1 General

11.3.1.1.1 Get_Help (HELP)

Retrieve a help text from controller for a specific command.

Parameter: String SP_Command

SP_Command

Direction: Down

Valid values:

"" (empty string, means general help)
or any command name

Description: Name of the command.

Parameter: String SA_HelpText

SA_HelpText

Direction: Up

Description: Help text to the command.

11.3.1.1.2 Get_Info (GETINFO)

Retrieve information about the controller.

Parameter: String SA_Sensor

SA_Sensor

Direction: Up

Valid values:

Numeric value
Description: Name of the controller.

Parameter: String SA_SerialNumber

SA_SerialNumber

Direction: Up

Valid values:

Numeric value
Description: Serial number of the controller.

Parameter: String SA_Option

SA_Option

Direction: Up

Valid values:

Numeric value
Description: Option of the controller.

Parameter: String SA_ArticleNumber

SA_ArticleNumber

Direction: Up

Valid values:

Numeric value
Description: Article number of the controller.

Parameter: String SA_MacAddress

SA_MacAddress

Direction: Up

Valid values:

Valid MAC address in form of xx-xx-xx-xx-xx-xx
Description: MAC address (low level ethernet address) of the controller.

Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the controller.	
Parameter: String SA_ImageType	SA_ImageType
Direction: Up	
Description: Firmware image type used by the controller.	
Parameter: String SA_HardwareRevision	SA_HardwareRevision
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of the controller board.	

11.3.1.1.3 Get_Temperature (GETTEMP)

Get temperature of controller.

Parameter: double SA_Temperature	SA_Temperature
Direction: Up	
Unit: °C	
Description: Temperature.	

11.3.1.1.4 Set_Echo (ECHO)

Set echo for sensor commands.

Parameter: int32_t SP_Echo	SP_Echo
Direction: Down	
Valid values:	
0= Off	
1= On	
Description: Echo mode.	

11.3.1.1.5 Get_Echo (ECHO)

Get the echo mode.

Parameter: int32_t SA_Echo	SA_Echo
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Description: Echo mode.	

11.3.1.1.6 Get_AllParameters (PRINT)

Get all parameters from controller.

Parameter: int32_t SP_Additional	SP_Additional
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description:	If set, additional information about controller, sensor and material is output.
Parameter: int32_t SA_SyncMode	SA_SyncMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Master	
2= Slave	
3= Slave with external trigger (SLAVE_EXT)	
Description:	Synchronization mode.
Parameter: int32_t SA_UserLevel	SA_UserLevel
Direction: Up	
Valid values:	
-1= Other user level (only for internal use)	
0= User	
1= Professional	
Description:	Actual user level.
Parameter: int32_t SA_DefaultUser	SA_DefaultUser
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= User	
1= Professional	
Description:	Default user level.
Parameter: int32_t SA_Echo	SA_Echo
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Description:	Echo mode.
Parameter: int32_t SA_ActiveSensor	SA_ActiveSensor
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description:	Index of active sensor head.

Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Edge	
2= Level (PULSE)	
3= Software	
4= Encoder	
Description: Trigger mode.	
Parameter: int32_t SA_TriggerMoment	SA_TriggerMoment
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Input	
1= Output	
Description: Trigger moment.	
Parameter: int32_t SA_TriggerLevel	SA_TriggerLevel
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= High	
1= Low	
Description: Trigger level.	
Parameter: int32_t SA_TriggerCount	SA_TriggerCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 16383	
Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.	
Parameter: int32_t SA_EncoderNumber	SA_EncoderNumber
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 3	
Description: Encoder number to trigger on.	
Parameter: double SA_EncoderIncrements	SA_EncoderIncrements
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 2147483647.0 (INT32_MAX)	
Description: Number of encoder increments before trigger. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	

Parameter: double SA_EncoderMinValue	SA_EncoderMinValue
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Lower encoder limit. Above this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: double SA_Encoder.MaxValue	SA_Encoder.MaxValue
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Upper encoder limit. Below this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: int32_t SA_EncoderInterpolation1..3	SA_EncoderInterpolation1..3
Direction: Up	
Valid values:	
1	
2	
4	
Description: Encoder interpolation.	
Parameter: int32_t SA_EncoderMode1..3	SA_EncoderMode1..3
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = No action (NONE)	
1 = Set encoder value to preset value only one time (ONE)	
2 = Set encoder value to preset value each time (EVER)	
Description: Mode of encoder when reference is reached.	
Parameter: double SA_EncoderPreload1..3	SA_EncoderPreload1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: double SA_Encoder.MaxValue1..3	SA_Encoder.MaxValue1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Maximum value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	

Parameter: int32_t SA_EthernetMode SA_EthernetMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Ethernet
- 1 = Ethercat

Description: Ethernet mode.

Parameter: int32_t SA_DHCPEnabled SA_DHCPEnabled

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = FALSE
- 1 = TRUE

Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).

Parameter: String SA_Address SA_Address

Direction: Up

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.

Parameter: String SA_SubnetMask SA_SubnetMask

Direction: Up

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.

Parameter: String SA_Gateway SA_Gateway

Direction: Up

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

Parameter: int32_t SA_Protocol SA_Protocol

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = TCP server (SERVER/TCP)
- 1 = TCP client (CLIENT/TCP)
- 2 = UDP sender (CLIENT/UDP)
- 3 = None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SA_RemoteAddress SA_RemoteAddress

Direction: Up

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to.

Parameter: int32_t SA_Port SA_Port

Direction: Up

Valid values:

Minimum: 1024

Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate

Direction: Up

Valid values:

4000000

3500000

3000000

2500000

2000000

1500000

921600

691200

460800

230400

115200

9600

Unit: Baud

Description: Baudrate of controller.

Parameter: int32_t SA_MeasureMode SA_MeasureMode

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor

0 = Distance

1 = Thickness

2 = Video

3 = MultiLayer (only at IFD2451 and IFD2461 and IFD2471)

4 = VideoStream

Description: Measure mode.

Parameter: int32_t SA_RefractiveCorrection SA_RefractiveCorrection

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor

0 = Off

1 = On

Description: Tells if refractive correction is be enabled.

Parameter: int32_t SA_NumberOfPeaks SA_NumberOfPeaks

Direction: Up

Valid values:

Minimum: 2

Maximum: 6

Description: Number of peaks to detect.

Parameter: int32_t SA_ShutterMode	SA_ShutterMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Search the best exposure time and measuring rate automatically (SEARCH)	
1 = Control exposure time, measuring rate is set manually (MEAS)	
2 = Exposure time and measuring rate are set manually (MANUAL)	
3 = Use two fixed exposure times alternately (2TIMEALT)	
4 = Use the suitable exposure time of two fixed ones (2TIMES)	
Description: Shutter mode.	
Parameter: double SA_Measrate	SA_Measrate
Direction: Up	
Valid values:	
Minimum: 0.1 at IFD2445 and IFD2451 and IFD2461, 0.3 at IFD2471	
Maximum: 2.5 at IFD2445, 10.0 at IFD2451, 25.0 at IFD2461, 70.0 at IFD2471	
Unit: kHz	
Description: Samplerate of measurement. For older firmware versions (before V007.117.134.02), only discrete values are valid: 0.1, 0.2, 0.3, 1.0, 2.5, 5.0, 10.0, 25.0, 50.0, 70.0	
Parameter: double SA_ShutterTime1	SA_ShutterTime1
Direction: Up	
Valid values:	
Minimum: 0.075	
Maximum: 10000.0 (at IFD2445 and IFD2451 and IFD2461) or 3333.325 (at IFD2471)	
Unit: <i>μs</i>	
Description: First shutter time.	
Parameter: double SA_ShutterTime2	SA_ShutterTime2
Direction: Up	
Valid values:	
Minimum: 0.075	
Maximum: 10000.0 (at IFD2445 and IFD2451 and IFD2461) or 3333.325 (at IFD2471)	
Unit: <i>μs</i>	
Description: Second shutter time.	
Parameter: int32_t SA_ROIStart	SA_ROIStart
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: First position on CCD.	
Parameter: int32_t SA_ROIEnd	SA_ROIEnd
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: Last position on CCD.	

Parameter: int32_t SA_VideoAverage	SA_VideoAverage
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Recursive over 2 lines (REC2)	
2= Recursive over 4 lines (REC4)	
3= Recursive over 8 lines (REC8)	
4= Moving over 2 lines (MOV2)	
5= Moving over 3 lines (MOV3)	
6= Moving over 4 lines (MOV4)	
7= Median over 3 lines (MED3)	
Description: Averaging mode.	
Parameter: double SA_Threshold	SA_Threshold
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 99.0	
Unit: %	
Description: Video threshold.	
Parameter: String SA_ActiveMaterial	SA_ActiveMaterial
Direction: Up	
Description: Name of material.	
Parameter: String SA_MaterialMultiPeak12	SA_MaterialMultiPeak12
Direction: Up	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Name of material between first and second peak.	
Parameter: String SA_MaterialMultiPeak23	SA_MaterialMultiPeak23
Direction: Up	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Name of material between second and third peak.	
Parameter: String SA_MaterialMultiPeak34	SA_MaterialMultiPeak34
Direction: Up	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Name of material between third and fourth peak.	
Parameter: String SA_MaterialMultiPeak45	SA_MaterialMultiPeak45
Direction: Up	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Name of material between fourth and fifth peak.	

Parameter: String SA_MaterialMultiPeak56	SA_MaterialMultiPeak56
Direction: Up	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Name of material between fifth and sixth peak.	
Parameter: int32_t SA_AveragingType	SA_AveragingType
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Moving average (MOVING)	
2= Recursive averaging (RECURSIVE)	
3= Median	
Description: Averaging type.	
Parameter: int32_t SA_MovingCount	SA_MovingCount
Direction: Up	
Valid values:	
2	
4	
8	
16	
32	
64	
128	
256	
512	
1024	
Description: Number of value for the averaging window. This parameter is only available at moving average.	
Parameter: int32_t SA_RecursiveCount	SA_RecursiveCount
Direction: Up	
Valid values:	
Minimum: 2	
Maximum: 32768	
Description: Number of values for recursive averaging. This parameter is only available at recursive average.	
Parameter: int32_t SA_MedianCount	SA_MedianCount
Direction: Up	
Valid values:	
3	
5	
7	
9	
Description: Number of values to build median. This parameter is only available at median.	

Parameter: int32_t SA_SpikeCorrection	SA_SpikeCorrection
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= off	
1= on	
Description: Spike correction.	
Parameter: int32_t SA_NbrEvaluatedValues	SA_NbrEvaluatedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Number of values to evaluate for spike correction.	
Parameter: double SA_ToleranceRange	SA_ToleranceRange
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 100.0	
Description: Tolerance range for spike correction.	
Parameter: int32_t SA_NbrCorrectedValues	SA_NbrCorrectedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100	
Description: Number of values to correct at spike correction.	
Parameter: int32_t SA_StatisticSignal	SA_StatisticSignal
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Distance 1	
1= Distance 2	
2= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)	
3= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)	
4= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)	
5= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)	
6= Thickness 12	
7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)	
8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)	
9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)	
10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)	
11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)	
12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)	
13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)	
14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)	
15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)	
16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)	
17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)	
18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)	
19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)	
20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)	
Description: Value which is used for statistic calculation.	

Parameter: int32_t SA_MasterSignal SA_MasterSignal

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Distance 1
- 1= Distance 2
- 2= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)
- 3= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)
- 4= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)
- 5= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)
- 6= Thickness 12
- 7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)
- 8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)
- 9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)
- 10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)
- 11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)
- 12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)
- 13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)
- 14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)
- 15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)
- 16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)
- 17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)
- 18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)
- 19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)
- 20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)

Description: Value which is used for mastering.

Parameter: int32_t SA_MeasurePeak SA_MeasurePeak

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= First peak at distance mode resp. first and last peak at thickness mode (F_L)
- 1= Last peak at distance mode resp. last and next to last peak at thickness mode (L_SL)
- 2= First peak at distance mode resp. first and second peak at thickness mode (F_S)
- 3= Highest peak at distance mode resp. highest and second highest peak at thickness mode (H_SH)

Description: Peaks to evalualte.

Parameter: int32_t SA_StatisticDepth SA_StatisticDepth

Direction: Up

Valid values:

Minimum: 2

Maximum: 2147483647 (INT32_MAX)

Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 16384). Value greater as 16384 calculates statistic over all values.

Parameter: int32_t SA_Master SA_Master

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor

0= no (NONE)

1= yes (MASTER)

Description: Specifies if mastering is active.

Parameter: double SA_MasterValue

SA_MasterValue

Direction: Up

Valid values:

Minimum: -2* measuring range

Maximum: +2* measuring range

Unit: mm

Description: Master value

Parameter: int32_t SA_DataOutInterface

SA_DataOutInterface

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= None

1= RS422

2= Ethernet

3= HTTP

4= Ethercat

Description: Active interface for data output.

Parameter: int32_t SA_Resampling

SA_Resampling

Direction: Up

Valid values:

Minimum: 1

Maximum: 4200000

Description: Resampling value.

Parameter: int32_t SA_ResampleAnalog

SA_ResampleAnalog

Direction: Up

Valid values:

0= no

1= yes

Description: Analog output is resampled.

Parameter: int32_t SA_ResampleRS422

SA_ResampleRS422

Direction: Up

Valid values:

0= no

1= yes

Description: RS422 output is resampled.

Parameter: int32_t SA_ResampleEthernet

SA_ResampleEthernet

Direction: Up

Valid values:

0= no

1= yes

Description: Output over ethernet is resampled.

Parameter: int32_t SA_HoldLastValid	SA_HoldLastValid
Direction: Up	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	
Parameter: int32_t SA_OutputDistance1_RS422	SA_OutputDistance1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 is transmitted.	
Parameter: int32_t SA_OutputDistance2_RS422	SA_OutputDistance2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 2 is transmitted.	
Parameter: int32_t SA_OutputDistance3_RS422	SA_OutputDistance3_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 3 is transmitted.	
Parameter: int32_t SA_OutputDistance4_RS422	SA_OutputDistance4_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 4 is transmitted.	
Parameter: int32_t SA_OutputDistance5_RS422	SA_OutputDistance5_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 5 is transmitted.	

Parameter: int32_t SA_OutputDistance6_RS422	SA_OutputDistance6_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 6 is transmitted.	
Parameter: int32_t SA_OutputDistance1_ETH	SA_OutputDistance1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 is transmitted.	
Parameter: int32_t SA_OutputDistance2_ETH	SA_OutputDistance2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 2 is transmitted.	
Parameter: int32_t SA_OutputDistance3_ETH	SA_OutputDistance3_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 3 is transmitted.	
Parameter: int32_t SA_OutputDistance4_ETH	SA_OutputDistance4_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 4 is transmitted.	

Parameter: int32_t SA_OutputDistance5_ETH	SA_OutputDistance5_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 5 is transmitted.	
Parameter: int32_t SA_OutputDistance6_ETH	SA_OutputDistance6_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 6 is transmitted.	
Parameter: int32_t SA_OutputThickness12_RS422	SA_OutputThickness12_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if thickness between first and second peak is transmitted.	
Parameter: int32_t SA_OutputThickness13_RS422	SA_OutputThickness13_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between first and third peak is transmitted.	
Parameter: int32_t SA_OutputThickness14_RS422	SA_OutputThickness14_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between first and fourth peak is transmitted.	

Parameter: int32_t SA_OutputThickness15_RS422	SA_OutputThickness15_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between first and fifth peak is transmitted.	
Parameter: int32_t SA_OutputThickness16_RS422	SA_OutputThickness16_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between first and sixth peak is transmitted.	
Parameter: int32_t SA_OutputThickness23_RS422	SA_OutputThickness23_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and third peak is transmitted.	
Parameter: int32_t SA_OutputThickness24_RS422	SA_OutputThickness24_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and fourth peak is transmitted.	
Parameter: int32_t SA_OutputThickness25_RS422	SA_OutputThickness25_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and fifth peak is transmitted.	

Parameter: int32_t SA_OutputThickness26_RS422	SA_OutputThickness26_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and sixth peak is transmitted.	
Parameter: int32_t SA_OutputThickness34_RS422	SA_OutputThickness34_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between third and fourth peak is transmitted.	
Parameter: int32_t SA_OutputThickness35_RS422	SA_OutputThickness35_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between third and fifth peak is transmitted.	
Parameter: int32_t SA_OutputThickness36_RS422	SA_OutputThickness36_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between third and sixth peak is transmitted.	
Parameter: int32_t SA_OutputThickness45_RS422	SA_OutputThickness45_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between fourth and fifth peak is transmitted.	

Parameter: int32_t SA_OutputThickness46_RS422	SA_OutputThickness46_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between fourth and sixth peak is transmitted.
Parameter: int32_t SA_OutputThickness56_RS422	SA_OutputThickness56_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between fifth and sixth peak is transmitted.
Parameter: int32_t SA_OutputThickness12_ETH	SA_OutputThickness12_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if thickness between first and second peak is transmitted.
Parameter: int32_t SA_OutputThickness13_ETH	SA_OutputThickness13_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between first and third peak is transmitted.
Parameter: int32_t SA_OutputThickness14_ETH	SA_OutputThickness14_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between first and fourth peak is transmitted.

Parameter: int32_t SA_OutputThickness15_ETH	SA_OutputThickness15_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between first and fifth peak is transmitted.	
Parameter: int32_t SA_OutputThickness16_ETH	SA_OutputThickness16_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between first and sixth peak is transmitted.	
Parameter: int32_t SA_OutputThickness23_ETH	SA_OutputThickness23_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and third peak is transmitted.	
Parameter: int32_t SA_OutputThickness24_ETH	SA_OutputThickness24_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and fourth peak is transmitted.	
Parameter: int32_t SA_OutputThickness25_ETH	SA_OutputThickness25_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and fifth peak is transmitted.	

Parameter: int32_t SA_OutputThickness26_ETH	SA_OutputThickness26_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and sixth peak is transmitted.	
Parameter: int32_t SA_OutputThickness34_ETH	SA_OutputThickness34_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between third and fourth peak is transmitted.	
Parameter: int32_t SA_OutputThickness35_ETH	SA_OutputThickness35_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between third and fifth peak is transmitted.	
Parameter: int32_t SA_OutputThickness36_ETH	SA_OutputThickness36_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between third and sixth peak is transmitted.	
Parameter: int32_t SA_OutputThickness45_ETH	SA_OutputThickness45_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between fourth and fifth peak is transmitted.	

Parameter: int32_t SA_OutputThickness46_ETH	SA_OutputThickness46_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between fourth and sixth peak is transmitted.
Parameter: int32_t SA_OutputThickness56_ETH	SA_OutputThickness56_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between fifth and sixth peak is transmitted.
Parameter: int32_t SA_OutputStatisticMin_RS422	SA_OutputStatisticMin_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if min value is transmitted.
Parameter: int32_t SA_OutputStatisticMax_RS422	SA_OutputStatisticMax_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if max value is transmitted.
Parameter: int32_t SA_OutputStatisticPeak2Peak_RS422	SA_OutputStatisticPeak2Peak_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if peak to peak value is transmitted.
Parameter: int32_t SA_OutputStatisticMin_ETH	SA_OutputStatisticMin_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if min value is transmitted.
Parameter: int32_t SA_OutputStatisticMax_ETH	SA_OutputStatisticMax_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if max value is transmitted.

Parameter: int32_t SA_OutputStatisticPeak2Peak_ETH	SA_OutputStatistic-Peak2Peak_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if peak to peak value is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_RS422	SA_OutputAdditionalShutterTime_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1_RS422	SA_OutputAdditionalEncoder1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder2_RS422	SA_OutputAdditionalEncoder2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 2 is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder3_RS422	SA_OutputAdditionalEncoder3_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 3 is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_RS422	SA_OutputAdditionalIntensity_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity is transmitted.	

Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalTrgTimeDiff_RS422	SA_OutputAdditionalTrgTimeDiff_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger time difference is transmitted.	
Parameter: int32_t SA_OutputAdditionalMeasrate_RS422	SA_OutputAdditionalMeasrate_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measrate is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_ETH	SA_OutputAdditionalShutterTime_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1_ETH	SA_OutputAdditionalEncoder1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder2_ETH	SA_OutputAdditionalEncoder2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 2 is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder3_ETH	SA_OutputAdditionalEncoder3_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 3 is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditionalCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	

Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_ETH	SA_OutputAdditionalIntensity_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_ETH	SA_OutputAdditionalState_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalTrgTimeDiff_ETH	SA_OutputAdditionalTrgTimeDiff_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger time difference is transmitted.	
Parameter: int32_t SA_OutputAdditionalMeasrate_ETH	SA_OutputAdditionalMeasrate_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measrate is transmitted.	
Parameter: int32_t SA_OutputVideoRaw_ETH	SA_OutputVideoRaw_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal is transmitted.	
Parameter: int32_t SA_OutputVideoDark_ETH	SA_OutputVideoDark_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if dark preprocessed video signal is transmitted.	
Parameter: int32_t SA_OutputVideoLight_ETH	SA_OutputVideoLight_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if light preprocessed video signal is transmitted.	

Parameter: int32_t SA_OutputVideoDarkTable_ETH	SA_OutputVideoDarkTable_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if dark table is transmitted.	
Parameter: int32_t SA_OutputVideoLightTable_ETH	SA_OutputVideoLight- Table_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if light table is transmitted.	
Parameter: int32_t SA_OutputVideoThreshold_ETH	SA_OutputVideoThreshold_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if video threshold table is transmitted.	
Parameter: int32_t SA_ErrorOutput1..2	SA_ErrorOutput1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Intensity error (ER1)	
2= Out of range (ER2)	
3= Intensity error or out of range (ER12)	
4= Below low limit (LI1)	
5= Above high limit (LI2)	
6= Out of limits (LI12)	
Description: Condition for error output.	
Parameter: int32_t SA_DataSource	SA_DataSource
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Distance 1	
1= Distance 2	
3= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)	
4= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)	
5= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)	
6= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)	
2= Thickness 12	
7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)	
8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)	
9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)	
10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)	
11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)	
12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)	
13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)	
14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)	
15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)	

- 16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)
- 17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)
- 18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)
- 19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)
- 20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)

Description: Data source to be checked.

Parameter: double SA_LowerLimit SA_LowerLimit

Direction: Up

Valid values:

- Minimum:** -120.0
- Maximum:** 120.0

Unit: mm

Description: Lower limit.

Parameter: double SA_UpperLimit SA_UpperLimit

Direction: Up

Valid values:

- Minimum:** -120.0
- Maximum:** 120.0

Unit: mm

Description: Upper limit.

Parameter: int32_t SA_ErrorLevel SA_ErrorLevel

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= High
- 1= Low

Description: Error level.

Parameter: int32_t SA_AnalogOutput SA_AnalogOutput

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Distance 1
- 1= Distance 2
- 3= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)
- 4= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)
- 5= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)
- 6= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)
- 2= Thickness 12
- 7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)
- 8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)
- 9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)
- 10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)
- 11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)
- 12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)
- 13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)
- 14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)
- 15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)
- 16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)
- 17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)
- 18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)
- 19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)
- 20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)

Description: Data to be used for analog output.

Parameter: int32_t SA_AnalogRange SA_AnalogRange

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= 0 - 5V
- 2= 0 - 10V
- 3= -5 - 5V
- 4= -10 - 10V
- 5= 4 - 20mA

Description: Analog output range.

Parameter: int32_t SA_AnalogScaleMode SA_AnalogScaleMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Standard
- 1= Two point (TWOPOINT)

Description: Analog scale mode.

Parameter: double SA_MinValue SA_MinValue

Direction: Up

Valid values:

- Minimum:** -120.0
- Maximum:** 120.0

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SA_MaxValue SA_MaxValue

Direction: Up

Valid values:

- Minimum:** -120.0
- Maximum:** 120.0

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

Parameter: String SA_Sensor SA_Sensor

Direction: Up

Description: Name of the controller.

Parameter: String SA_SerialNumber SA_SerialNumber

Direction: Up

Valid values:

- Numeric value

Description: Serial number of the controller.

Parameter: String SA_Option SA_Option

Direction: Up

Valid values:

- Numeric value

Description: Option of the controller.

Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the controller.	
Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description: MAC address (low level ethernet address) of the controller.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the controller.	
Parameter: String SA_ImageType	SA_ImageType
Direction: Up	
Description: Firmware image type used by the controller.	
Parameter: String SA_HardwareRevision	SA_HardwareRevision
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of the controller board.	
Parameter: int32_t SA_Pos	SA_Pos
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the active sensor head in the table.	
Parameter: String SA_Name	SA_Name
Direction: Up	
Description: Name of the active sensor head.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Description: Measurement range of the active sensor head.	
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description: Unit of the measurement range.	
Parameter: String SA_Serial	SA_Serial
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the active sensor head.	
Parameter: String SA_SensorTable	SA_SensorTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	

Parameter: int32_t SA_SensorTableCount	SA_SensorTableCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 20	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Pos1, SA_Pos2, ...	
Parameter: int32_t SA_Pos1..x	SA_Pos1..x
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the sensor head in the table.	
Parameter: String SA_Name1..x	SA_Name1..x
Direction: Up	
Description: Name of the sensor head in the table.	
Parameter: double SA_Range1..x	SA_Range1..x
Direction: Up	
Valid values:	
Minimum: 0.0	
Description: Measurement range of the sensor head in the table.	
Parameter: String SA_Unit1..x	SA_Unit1..x
Direction: Up	
Description: Unit of the measurement range.	
Parameter: String SA_Serial1..x	SA_Serial1..x
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor head in the table.	
Parameter: String SA_MaterialName	SA_MaterialName
Direction: Up	
Description: Name of the active material.	
Parameter: String SA_Description	SA_Description
Direction: Up	
Description: Description of the active material.	
Parameter: double SA_RefRACTIVEINDEX_nF	SA_RefRACTIVEINDEX_nF
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 486 nm.	
Parameter: double SA_RefRACTIVEINDEX_nd	SA_RefRACTIVEINDEX_nd
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 587 nm.	

Parameter: double SA_RefractiveIndex_nC	SA_RefractiveIndex_nC
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 656 nm.	
Parameter: double SA_AbbeNumber_vd	SA_AbbeNumber_vd
Direction: Up	
Valid values:	
Minimum: 10.0	
Maximum: 100.0	
Description: Abbe number of the active material or 0.0 if not specified.	
Parameter: int32_t SA_Described_by	SA_Described_by
Direction: Up	
Valid values:	
0= nF, nd and nC	
1= nd and abbe value	
Description: Tells if nF, nd and nC or if nd and abbe value is valid.	
Parameter: int32_t SA_UnlinearizedMode	SA_UnlinearizedMode
Direction: Up	
Valid values:	
0= off	
1= on	
Description: Unlinearized data mode.	
Parameter: double SA_DarkCorrThreshold	SA_DarkCorrThreshold
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 99.0	
Unit: %	
Description: Darkcorr threshold.	

11.3.1.1.7 Set_SyncMode (SYNC)

Set the synchronization mode.

Parameter: int32_t SP_SyncMode	SP_SyncMode
Direction: Down	
Valid values:	
0= None	
1= Master	
2= Slave	
3= Slave with external trigger (SLAVE_EXT)	
Description: Synchronization mode.	

11.3.1.1.8 Get_SyncMode (SYNC)

Get the synchronization mode.

Parameter: int32_t SA_SyncMode

SA_SyncMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Master
- 2= Slave
- 3= Slave with external trigger (SLAVE_EXT)

Description: Synchronization mode.

11.3.1.1.9 Reset_Boot (RESET)

Resets the sensor.

At this command the controller may change output data after reboot. If first bit of [IP_AutomaticMode](#) is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

11.3.1.2 User level

11.3.1.2.1 Logout (LOGOUT)

Change user level to user.

11.3.1.2.2 Login (LOGIN)

Change user level to professional.

Parameter: String SP_Password

SP_Password

Direction: Down

Description: Valid password to login.

11.3.1.2.3 Get_UserLevel (GETUSERLEVEL)

Retrieve actual user level.

Parameter: int32_t SA_UserLevel

SA_UserLevel

Direction: Up

Valid values:

- 1= Other user level (only for internal use)
- 0= User
- 1= Professional

Description: Actual user level.

11.3.1.2.4 Set_DefaultUser (STDUSER)

Set the default user level after booting the system.

Parameter: int32_t SP_DefaultUser SP_DefaultUser
Direction: Down
Valid values:
 0= User
 1= Professional
Description: Default user level.

11.3.1.2.5 Get_DefaultUser (STDUSER)

Get the default user level after booting the system.

Parameter: int32_t SA_DefaultUser SA_DefaultUser
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= User
 1= Professional
Description: Default user level.

11.3.1.2.6 Set_Password (PASSWD)

Change the password for login.

Parameter: String SP_OldPassword SP_OldPassword
Direction: Down
Description: Old password.

Parameter: String SP_NewPassword SP_NewPassword
Direction: Down
Description: New password.

11.3.1.3 Sensor

11.3.1.3.1 Get_SensorTable (SENSORTABLE)

Get a list of all teached sensor heads.

Parameter: String SA_SensorTable SA_SensorTable
Direction: Up
Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_SensorTableCount SA_SensorTableCount
Direction: Up
Valid values:
Minimum: 0
Maximum: 20
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Pos1, SA_Pos2, ...

Parameter: int32_t SA_Pos1..x	SA_Pos1..x
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the sensor head in the table.	
Parameter: String SA_Name1..x	SA_Name1..x
Direction: Up	
Description: Name of the sensor head in the table.	
Parameter: double SA_Range1..x	SA_Range1..x
Direction: Up	
Valid values:	
Minimum: 0.0	
Description: Measurement range of the sensor head in the table.	
Parameter: String SA_Unit1..x	SA_Unit1..x
Direction: Up	
Description: Unit of the measurement range.	
Parameter: String SA_Serial1..x	SA_Serial1..x
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor head in the table.	

11.3.1.3.2 Set_ActiveSensor (SENSORHEAD)

Change to another sensor head.

Parameter: int32_t SP_ActiveSensor	SP_ActiveSensor
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of new sensor head.	

11.3.1.3.3 Get_ActiveSensor (SENSORHEAD)

Get active sensor head.

Parameter: int32_t SA_ActiveSensor	SA_ActiveSensor
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of active sensor head.	

11.3.1.3.4 Get_SensorInfo (SENSORINFO)

Get information of active sensor head.

Parameter: int32_t SA_Pos	SA_Pos
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description:	Index of the active sensor head in the table.
Parameter: String SA_Name	SA_Name
Direction: Up	
Description:	Name of the active sensor head.
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Description:	Measurement range of the active sensor head.
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description:	Unit of the measurement range.
Parameter: String SA_Serial	SA_Serial
Direction: Up	
Valid values:	
Numeric value	
Description:	Serial number of the active sensor head.

11.3.1.3.5 Set_LEDControl (LED)

Switches the white light LED ON or OFF.

Only available from firmware version V07.119.152.02

Parameter: int32_t SP_LEDControl	SP_LEDControl
Direction: Down	
Valid values:	
0= Off	
1= On	
Description:	LED state

11.3.1.3.6 Get_LEDControl (LED)

Get state of white light LED.

Only available from firmware version V07.119.152.02

Parameter: int32_t SA_LEDControl	SA_LEDControl
Direction: Up	
Valid values:	
0= Off	
1= On	
Description:	LED state

11.3.1.3.7 DarkCorr (DARKCORR)

Make a dark correction.

In measure mode MultiLayer and data output interface Ethernet the controller may change output data automatically. Next call to [DataAvail](#) or [TransferData](#) will return a warning. In this case call [Get_TransmittedDataInfo](#) to retrieve new setting.

11.3.1.3.8 Set_DarkCorrThreshold (DARKCORRTHRES)

Set the darkcorr threshold at the controller.

Parameter: double SP_DarkCorrThreshold

SP_DarkCorrThreshold

Direction: Down

Valid values:

Minimum: 1.0

Maximum: 99.0

Unit: %

Description: Darkcorr threshold.

11.3.1.3.9 Get_DarkCorrThreshold (DARKCORRTHRES)

Get the darkcorr threshold from controller.

Parameter: double SA_DarkCorrThreshold

SA_DarkCorrThreshold

Direction: Up

Valid values:

Minimum: 1.0

Maximum: 99.0

Unit: %

Description: Darkcorr threshold.

11.3.1.3.10 LightCorr (LIGHTCORR)

Make a light correction.

Valid for sensor:

IFD2471

11.3.1.4 Triggering

11.3.1.4.1 Set_TriggerMode (TRIGGER)

Set the trigger mode.

Parameter: int32_t SP_TriggerMode

SP_TriggerMode

Direction: Down

Valid values:

0= None

1= Edge

2= Level (PULSE)

3= Software

4= Encoder

Description: Trigger mode.

11.3.1.4.2 Get_TriggerMode (TRIGGER)

Get the active trigger mode.

Parameter: int32_t SA_TriggerMode

SA_TriggerMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Edge
- 2= Level (PULSE)
- 3= Software
- 4= Encoder

Description: Trigger mode.

11.3.1.4.3 Set_TriggerMoment (TRIGGERAT)

Set the trigger time.

Parameter: int32_t SP_TriggerMoment

SP_TriggerMoment

Direction: Down

Valid values:

- 0= Input
- 1= Output

Description: Trigger moment.

11.3.1.4.4 Get_TriggerMoment (TRIGGERAT)

Get the active trigger time.

Parameter: int32_t SA_TriggerMoment

SA_TriggerMoment

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Input
- 1= Output

Description: Trigger moment.

11.3.1.4.5 Set_TriggerLevel (TRIGGERLEVEL)

Set the trigger level.

Parameter: int32_t SP_TriggerLevel

SP_TriggerLevel

Direction: Down

Valid values:

- 0= High
- 1= Low

Description: Trigger level.

11.3.1.4.6 Get_TriggerLevel (TRIGGERLEVEL)

Get the active trigger level.

Parameter: int32_t SA_TriggerLevel

SA_TriggerLevel

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = High
- 1 = Low

Description: Trigger level.

11.3.1.4.7 Set_TriggerCount (TRIGGERCOUNT)

Set the number of values to measure at trigger.

Parameter: int32_t SP_TriggerCount

SP_TriggerCount

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 16383

Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

11.3.1.4.8 Get_TriggerCount (TRIGGERCOUNT)

Get the number of values to measure at trigger.

Parameter: int32_t SA_TriggerCount

SA_TriggerCount

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 16383

Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

11.3.1.4.9 Software_Trigger (TRIGGERSW)

Execute a software trigger.

11.3.1.4.10 Set_TriggerOnEncoder (TRIGGERENC)

Set the trigger on encoder mode.

For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 3

Description: Encoder number to trigger on.

Parameter: double SP_EncoderIncrements	SP_EncoderIncrements
Direction: Down	
Valid values:	
Minimum: 1.0	
Maximum: 2147483647.0 (INT32_MAX)	
Description: Number of encoder increments before trigger. 0 means output data continuous. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: double SP_EncoderMinValue	SP_EncoderMinValue
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Lower encoder limit. Above this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: double SP_Encoder.MaxValue	SP_Encoder.MaxValue
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Upper encoder limit. Below this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	

11.3.1.4.11 Get_TriggerOnEncoder (TRIGGERENC)

Get the trigger on encoder mode.

Parameter: int32_t SA_EncoderNumber	SA_EncoderNumber
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 3	
Description: Encoder number to trigger on.	
Parameter: double SA_EncoderIncrements	SA_EncoderIncrements
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 2147483647.0 (INT32_MAX)	
Description: Number of encoder increments before trigger. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: double SA_EncoderMinValue	SA_EncoderMinValue
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Lower encoder limit. Above this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	

Parameter: double SA_Encoder.MaxValue SA_Encoder.MaxValue
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 4294967295.0 (UINT32_MAX)
Description: Upper encoder limit. Below this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.3.1.5 Encoder

11.3.1.5.1 Set_EncoderInterpolation<n> (ENCINTERPOL<n>)

Set the interpolation for encoder n ∈ {1..3}.

Parameter: int32_t SP_EncoderInterpolation<n> SP_EncoderInterpolation<n>
Direction: Down
Valid values:
 1
 2
 4
Description: Encoder interpolation.

11.3.1.5.2 Get_EncoderInterpolation<n> (ENCINTERPOL<n>)

Get the interpolation of encoder n ∈ {1..3}.

Parameter: int32_t SA_EncoderInterpolation<n> SA_EncoderInterpolation<n>
Direction: Up
Valid values:
 1
 2
 4
Description: Encoder interpolation.

11.3.1.5.3 Set_EncoderMode<n> (ENCREF<n>)

Set the behaviour of encoder n ∈ {1..3} when reference is reached.

Parameter: int32_t SP_EncoderMode<n> SP_EncoderMode<n>
Direction: Down
Valid values:
 0= No action (NONE)
 1= Set encoder value to preset value only one time (ONE)
 2= Set encoder value to preset value each time (EVER)
Description: Mode of encoder when reference is reached.

11.3.1.5.4 Get_EncoderMode<n> (ENCREF<n>)

Get the behaviour of encoder n $\in \{1..3\}$ when reference is reached.

Parameter: int32_t SA_EncoderMode<n>

SA_EncoderMode<n>

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = No action (NONE)
- 1 = Set encoder value to preset value only one time (ONE)
- 2 = Set encoder value to preset value each time (EVER)

Description: Mode of encoder when reference is reached.

11.3.1.5.5 Set_EncoderPreload<n> (ENCVALUE<n>)

Set preload value for encoder n $\in \{1..3\}$.

Parameter: double SP_EncoderPreload<n>

SP_EncoderPreload<n>

Direction: Down

Valid values:

- Minimum:** 0.0
- Maximum:** 4294967295.0 (UINT32_MAX)

Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.3.1.5.6 Get_EncoderPreload<n> (ENCVALUE<n>)

Get preload value for encoder n $\in \{1..3\}$.

Parameter: double SA_EncoderPreload<n>

SA_EncoderPreload<n>

Direction: Up

Valid values:

- Minimum:** 0.0
- Maximum:** 4294967295.0 (UINT32_MAX)

Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.3.1.5.7 Load_Encoder<n> (ENCSET)

Load the encoder n $\in \{1..3\}$ with the preset value.

11.3.1.5.8 EnableRef_Encoder<n> (ENCRESET)

Reset reference counter of encoder n $\in \{1..3\}$. If encoder mode is 2, at next reference encoder value will be set to preset value again

11.3.1.5.9 Set_Encoder.MaxValue<n> (ENCMAX<n>)

Set maximum value for encoder n $\in \{1..3\}$ before it wraps around.

Parameter: double SP_Encoder.MaxValue<n>

SP_Encoder.MaxValue<n>

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Maximum value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.3.1.5.10 Get_Encoder.MaxValue<n> (ENCMAX<n>)

Get maximum value for encoder n $\in \{1..3\}$ before it wraps around.

Parameter: double SA_Encoder.MaxValue<n>

SA_Encoder.MaxValue<n>

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Maximum value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

11.3.1.6 Interfaces

11.3.1.6.1 Set_IPConfiguration (IPCONFIG)

Set the IP configuration at controller.

For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_DHCPEnabled

SP_DHCPEnabled

Direction: Down

Valid values:

0= FALSE

1= TRUE

Description: Specify if controller should use a static IP address or ask for IP at DHCP server (dynamic IP address).

Parameter: String SP_Address

SP_Address

Direction: Down

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_SubnetMask

SP_SubnetMask

Direction: Down

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_Gateway SP_Gateway

Direction: Down

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: The default gateway must be specified if the controller should communicate with peers in foreign subnets. This parameter is only evaluated on static IP assignment.

11.3.1.6.2 Get_IPConfiguration (IPCONFIG)

Get the IP configuration at controller.

Parameter: int32_t SA_DHCPEnabled SA_DHCPEnabled

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= FALSE

1= TRUE

Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).

Parameter: String SA_Address SA_Address

Direction: Up

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.

Parameter: String SA_SubnetMask SA_SubnetMask

Direction: Up

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.

Parameter: String SA_Gateway SA_Gateway

Direction: Up

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

11.3.1.6.3 Set_IPDataTransferMode (MEATRANSFER)

Set IP protocol at controller.

For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_Protocol SP_Protocol

Direction: Down

Valid values:

0= TCP server (SERVER/TCP)

1= TCP client (CLIENT/TCP)

2= UDP sender (CLIENT/UDP)

3= None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SP_RemoteAddress SP_RemoteAddress

Direction: Down

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to. On TCP server this parameter is ignored.

Parameter: int32_t SP_Port SP_Port

Direction: Down

Valid values:

Minimum: 1024

Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

11.3.1.6.4 Get_IPDataTransferMode (MEATRANSFER)

Get IP protocol at controller.

Parameter: int32_t SA_Protocol SA_Protocol

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= TCP server (SERVER/TCP)

1= TCP client (CLIENT/TCP)

2= UDP sender (CLIENT/UDP)

3= None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SA_RemoteAddress SA_RemoteAddress

Direction: Up

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to.

Parameter: int32_t SA_Port SA_Port

Direction: Up

Valid values:

Minimum: 1024

Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

11.3.1.6.5 Set_EthernetMode (ETHERMODE)

Switches ethernet mode between Ethernet and Ethercat.

Parameter: int32_t SP_EthernetMode SP_EthernetMode

Direction: Down

Valid values:

0= Ethernet

1= Ethercat

Description: Ethernet mode.

11.3.1.6.6 Get_EthernetMode (ETHERMODE)

Get ethernet mode of controller.

Parameter: int32_t SA_EthernetMode

SA_EthernetMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Ethernet
- 1 = Ethercat

Description: Ethernet mode.

11.3.1.6.7 Set_Baudrate (BAUDRATE)

Set baudrate of controller for serial RS422 communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

- 4000000
- 3500000 (does not work in combination with IF2008 PCI interface card)
- 3000000 (does not work in combination with IF2008 PCI interface card)
- 2500000
- 2000000
- 1500000
- 921600
- 691200
- 460800
- 230400
- 115200
- 9600

Unit: Baud

Description: Baudrate of controller.

Parameter: int32_t CP_InterruptDataTransfer

CP_InterruptDataTransfer

Direction: Down

Valid values:

- 0 = false
- 1 = true

Default: 0

Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Set_DataOutInterface](#)) and enabling it again at end (only if enabled before).

11.3.1.6.8 Get_Baudrate (BAUDRATE)

Get baudrate of controller for serial RS422 communication.

Parameter: int32_t SA_SensorBaudrate

SA_SensorBaudrate

Direction: Up

Valid values:

- 4000000
- 3500000
- 3000000
- 2500000
- 2000000
- 1500000
- 921600
- 691200
- 460800
- 230400
- 115200
- 9600

Unit: Baud

Description: Baudrate of controller.

11.3.1.7 Parameter management

11.3.1.7.1 Save_Parameters (STORE)

Save actual parameters at controller. There can be saved several settings on different locations. So it is easy to switch to another setting.

Parameter: int32_t SP_ParameterSet

SP_ParameterSet

Direction: Down

Valid values:

- Minimum: 1
- Maximum: 8

Description: Location to save the settings.

11.3.1.7.2 Load_Parameters (READ)

Load stored parameters into controller RAM.

There can be loaded several settings from different locations. So it is easy to switch to another setting.

At this command the controller may change output data after applying new setting.

If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually.

After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_ParameterType

SP_ParameterType

Direction: Down

Valid values:

- 0= All settings (ALL)
- 1= Device settings (DEVICE)
- 2= Measurement settings (MEAS)

Description: Specifies which settings should be loaded.

Parameter: int32_t SP_ParameterSet SP_ParameterSet
Direction: Down
Valid values:
 Minimum: 1
 Maximum: 8
Description: Location from where the settings should be loaded.

11.3.1.7.3 Set_Default (SETDEFAULT)

Reset the controller to default settings.

At this command the controller may change output data after applying default settings. If first bit of [IP_AutomaticMode](#) is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DefaultType SP_DefaultType
Direction: Down
Valid values:
 0= All settings (ALL)
 1= Keep device settings temporary (NODEVICE)
 2= Only material table (MATERIAL)
Description: Specifies which settings should be reset.

11.3.2 Measurement

11.3.2.1 General

11.3.2.1.1 Set_MeasureMode (MEASMODE)

Set the measure mode.

At this command the controller may change output data automatically. If first bit of [IP_AutomaticMode](#) is set (1), [Get_AllParameters](#) (SP_Additional= 0) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_MeasureMode SP_MeasureMode
Direction: Down
Valid values:
 0= Distance
 1= Thickness
 2= Video
 3= MultiLayer (only at IFD2451 and IFD2461 and IFD2471 with MultiLayer option)
 4= VideoStream
Description: Measure mode.

Parameter: int32_t IP_TimerResolution IP_TimerResolution

Direction: Down

Unit: ms

Valid values:

- 1= Do not set timer resolution.
- 0= Use greatest possible accuracy.
- 1..2147483647 (INT32_MAX)= Resolution in milliseconds.

Unit: ms

Default: 0 if measure mode is VideoStream, otherwise -1

Description: This parameter is necessary at video stream mode. Unless MEDAQLib waits for new video signals (using Sleep API function) and the default resolution for this function is 15 milli seconds there may be a time jitter at processing video signals (e.g. 15, 0, 0, 0 ..., 0, 15, 0, 0, ...). So this parameter changes the Windows timer resolution (for Windows scheduler, set by timeBeginPeriod).

If it is not reseted manually when switching off video stream mode (set to -1), it is autoamtically reseted at CloseSensor.

11.3.2.1.2 Get_MeasureMode (MEASMODE)

Get the measure mode.

Parameter: int32_t SA_MeasureMode SA_MeasureMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Distance
- 1= Thickness
- 2= Video
- 3= MultiLayer (only at IFD2451 and IFD2461 and IFD2471)
- 4= VideoStream

Description: Measure mode.

11.3.2.1.3 Set_MeasurePeak (MEASPEAK)

Specifiy which peaks should be evaluated at controller.

Parameter: int32_t SP_MeasurePeak SP_MeasurePeak

Direction: Down

Valid values:

- 0= First peak at distance mode resp. first and last peak at thickness mode (F_L)
- 1= Last peak at distance mode resp. last and next to last peak at thickness mode (L_SL)
- 2= First peak at distance mode resp. first and second peak at thickness mode (F_S)
- 3= Highest peak at distance mode resp. highest and second highest peak at thickness mode (H_SH)

Description: Peaks to evalualte.

11.3.2.1.4 Get_MeasurePeak (MEASPEAK)

Retrieve which peaks are evaluated at controller.

Parameter: int32_t SA_MeasurePeak

SA_MeasurePeak

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= First peak at distance mode resp. first and last peak at thickness mode (F_L)
- 1= Last peak at distance mode resp. last and next to last peak at thickness mode (L_SL)
- 2= First peak at distance mode resp. first and second peak at thickness mode (F_S)
- 3= Highest peak at distance mode resp. highest and second highest peak at thickness mode (H_SH)

Description: Peaks to evalualte.

11.3.2.1.5 Set_ShutterMode (SHUTTERMODE)

Set the shutter mode.

Parameter: int32_t SP_ShutterMode

SP_ShutterMode

Direction: Down

Valid values:

- 0= Search the best exposure time and measuring rate automatically (SEARCH)
- 1= Control exposure time, measuring rate is set manually (MEAS)
- 2= Exposure time and measuring rate are set manually (MANUAL)
- 3= Use two fixed exposure times alternately (2TIMEALT)
- 4= Use the suitable exposure time of two fixed ones (2TIMES)

Description: Shutter mode.

11.3.2.1.6 Get_ShutterMode (SHUTTERMODE)

Get the shutter mode.

Parameter: int32_t SA_ShutterMode

SA_ShutterMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Search the best exposure time and measuring rate automatically (SEARCH)
- 1= Control exposure time, measuring rate is set manually (MEAS)
- 2= Exposure time and measuring rate are set manually (MANUAL)
- 3= Use two fixed exposure times alternately (2TIMEALT)
- 4= Use the suitable exposure time of two fixed ones (2TIMES)

Description: Shutter mode.

11.3.2.1.7 Set_Samplerate (MEASRATE)

Set the samplerate.

Parameter: double SP_Measrate

SP_Measrate

Direction: Down

Valid values:

Minimum: 0.1 (at IFD2445 and IFD2451 and IFD2461), 0.3 (at IFD2471)

Maximum: 2.5 (at IFD2445), 10.0 (at IFD2451), 25.0 (at IFD2461), 70.0 (at IFD2471)

Unit: kHz

Description: Samplerate of measurement. For older firmware versions (before V007.117.134.02), only discrete values are valid: 0.1, 0.2, 0.3, 1.0, 2.5, 5.0, 10.0, 25.0, 50.0, 70.0

11.3.2.1.8 Get_Samplerate (MEASRATE)

Get the samplerate.

Parameter: double SA_Measrate

SA_Measrate

Direction: Up

Valid values:

Minimum: 0.1 (at IFD2445 and IFD2451 and IFD2461), 0.3 (at IFD2471)

Maximum: 2.5 (at IFD2445), 10.0 (at IFD2451), 25.0 (at IFD2461), 70.0 (at IFD2471)

Unit: kHz

Description: Samplerate of measurement. For older firmware versions (before V007.117.134.02), only discrete values are valid: 0.1, 0.2, 0.3, 1.0, 2.5, 5.0, 10.0, 25.0, 50.0, 70.0

11.3.2.1.9 Set_ShutterTime (SHUTTER)

Set the shutter time.

Parameter: double SP_ShutterTime1

SP_ShutterTime1

Direction: Down

Valid values:

Minimum: 0.075

Maximum: 10000.0 (at IFD2445 and IFD2451 and IFD2461) or 3333.325 (at IFD2471)

Unit: μ s

Description: First shutter time.

Parameter: double SP_ShutterTime2

SP_ShutterTime2

Direction: Down

Valid values:

Minimum: 0.075

Maximum: 10000.0 (at IFD2445 and IFD2451 and IFD2461) or 3333.325 (at IFD2471)

Unit: μ s

Description: Second shutter time. Is automatically adapted (to first shutter time), if not less or equal to first shutter time.

11.3.2.1.10 Get_ShutterTime (SHUTTER)

Get the shutter time.

Parameter: double SA_ShutterTime1 SA_ShutterTime1

Direction: Up

Valid values:

Minimum: 0.075

Maximum: 10000.0 (at IFD2445 and IFD2451 and IFD2461) or 3333.325
(at IFD2471)

Unit: μs

Description: First shutter time.

Parameter: double SA_ShutterTime2 SA_ShutterTime2

Direction: Up

Valid values:

Minimum: 0.075

Maximum: 10000.0 (at IFD2445 and IFD2451 and IFD2461) or 3333.325
(at IFD2471)

Unit: μs

Description: Second shutter time.

11.3.2.1.11 Get_Video (GETVIDEO)

Get recent video signals from sensor.

Parameter: Binary data SA_VideoRaw SA_VideoRaw

Direction: Up

Valid values:

512 words (each 2 byte), each word is an intensity value.

Description: Raw video signal

Parameter: Binary data SA_VideoDark SA_VideoDark

Direction: Up

Valid values:

512 words (each 2 byte), each word is an intensity value.

Description: Dark corrected video signal

Parameter: Binary data SA_VideoLight SA_VideoLight

Direction: Up

Valid values:

512 words (each 2 byte), each word is an intensity value.

Description: Light corrected video signal

Parameter: Binary data SA_VideoDarkTable SA_VideoDarkTable

Direction: Up

Valid values:

512 words (each 2 byte), each word is an intensity value.

Description: Dark table

Parameter: Binary data SA_VideoLightTable	SA_VideoLightTable
Direction: Up	
Valid values:	512 words (each 2 byte), each word is an intensity value.
Description: Light table	
Parameter: Binary data SA_VideoThreshold	SA_VideoThreshold
Direction: Up	
Valid values:	512 words (each 2 byte), each word is an intensity value.
Description: Threshold table	
Parameter: double SA_VideoTimestamp	SA_VideoTimestamp
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1.79769e+308 (DBL_MAX)	
Unit: ms	
Description: Timestamp of the video signal. It starts from 1970 Jan 01 at 01:00. It is generated when the video has arrived at TCP/IP socket.	

Example how to read a video signal from sensor:

```

/* Do not forget to handle potential error after each call to MEDAQLib! */
/* Create sensor instance, open sensor via TCP/IP, set output to ethernet */
/* and then switch to video mode: */
err= SetIntExecSCmd (instance, "Set_MeasureMode", "SP_MeasureMode", 2 /*Video*/);

/* Select the desired video signal: */
err= SetParameterInt (instance, "SP_OutputVideoRaw_ETH", 1);
err= SetParameterInt (instance, "SP_OutputVideoDark_ETH", 1);
err= SetParameterInt (instance, "SP_OutputVideoLight_ETH", 0);
err= SetParameterInt (instance, "SP_OutputVideoDarkTable_ETH", 0);
err= SetParameterInt (instance, "SP_OutputVideoLightTable_ETH", 0);
err= SetParameterInt (instance, "SP_OutputVideoThreshold_ETH", 0);
err= ExecSCmd (instance, "Set_OutputVideo_ETH");

/* Acquire video signals: */
err= ExecSCmd (instance, "Get_Video");
uint16_t videoRaw[512], videoDark[512];
uint32_t maxlen= sizeof (videoRaw);
err= GetParameterBinary (instance, "SA_VideoRaw", (uint8_t *)videoRaw, &maxlen);
assert (maxlen==sizeof (videoRaw)); // additional validity check
maxlen= sizeof (videoDark);
err= GetParameterBinary (instance, "SA_VideoDark", (uint8_t *)videoDark, &maxlen);
assert (maxlen==sizeof (videoDark)); // additional validity check

/* Do anything with the received video signals */

```

11.3.2.1.12 Get_VideoStreamSignal

Read one video signal from video stream.

Parameter: int32_t SP_ReadMode	SP_ReadMode
Direction: Down	
Valid values:	
0= Each video signal	
1= Only newest video signal	
2= Automatic	
Description: This mode specifies if each video signal should be read or only the latest one. If set to automatic each video signal is read until the buffer does not overflow. If the buffer becomes full one or more video signals are discarded.	

Parameter: int32_t SP_WaitVideoTimeout	SP_WaitVideoTimeout
Direction: Down	
Unit: ms	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Timeout to wait for a video signal.	
Parameter: Binary data SA_VideoRaw	SA_VideoRaw
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Description: Raw video signal	
Parameter: Binary data SA_VideoDark	SA_VideoDark
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Description: Dark corrected video signal	
Parameter: Binary data SA_VideoLight	SA_VideoLight
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Description: Light corrected video signal	
Parameter: Binary data SA_VideoDarkTable	SA_VideoDarkTable
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Description: Dark table	
Parameter: Binary data SA_VideoLightTable	SA_VideoLightTable
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Description: Light table	
Parameter: Binary data SA_VideoThreshold	SA_VideoThreshold
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Description: Threshold table	
Parameter: double SA_VideoTimestamp	SA_VideoTimestamp
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.79769e+308 (DBL_MAX)	
Unit: ms	
Description: Timestamp of the video signal. It starts from 1970 Jan 01 at 01:00. It is generated when the video has arrived at TCP/IP socket.	

Parameter: int32_t SA_SkippedVideo SA_SkippedVideo
Direction: Up
Valid values:
Minimum: 0
Maximum: 2147483647 (INT32_MAX)
Description: Number of skipped video signals, if SP_ReadMode is not 0.

11.3.2.2 Video signal

11.3.2.2.1 Set_ROI (ROI)

Set the region of intererst for processing video signal.

Parameter: int32_t SP_ROIStart SP_ROIStart
Direction: Down
Valid values:
Minimum: 0
Maximum: 511
Unit: Pixel
Description: First position on CCD.

Parameter: int32_t SP_ROIEnd SP_ROIEnd
Direction: Down
Valid values:
Minimum: 0
Maximum: 511
Unit: Pixel
Description: Last position on CCD.

11.3.2.2.2 Get_ROI (ROI)

Get the region of intererst for processing video signal.

Parameter: int32_t SA_ROIStart SA_ROIStart
Direction: Up
Valid values:
Minimum: 0
Maximum: 511
Unit: Pixel
Description: First position on CCD.

Parameter: int32_t SA_ROIEnd SA_ROIEnd
Direction: Up
Valid values:
Minimum: 0
Maximum: 511
Unit: Pixel
Description: Last position on CCD.

11.3.2.2.3 Set_VideoAverage (VSAVERAGE)

Set video averaging (before processing).

Parameter: int32_t SP_VideoAverage

SP_VideoAverage

Direction: Down

Valid values:

- 0= None
- 1= Recursive over 2 lines (REC2)
- 2= Recursive over 4 lines (REC4)
- 3= Recursive over 8 lines (REC8)
- 4= Moving over 2 lines (MOV2)
- 5= Moving over 3 lines (MOV3)
- 6= Moving over 4 lines (MOV4)
- 7= Median over 3 lines (MED3)

Description: Averaging mode.

11.3.2.2.4 Get_VideoAverage (VSAVERAGE)

Get video averaging (before processing).

Parameter: int32_t SA_VideoAverage

SA_VideoAverage

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Recursive over 2 lines (REC2)
- 2= Recursive over 4 lines (REC4)
- 3= Recursive over 8 lines (REC8)
- 4= Moving over 2 lines (MOV2)
- 5= Moving over 3 lines (MOV3)
- 6= Moving over 4 lines (MOV4)
- 7= Median over 3 lines (MED3)

Description: Averaging mode.

11.3.2.2.5 Set_Threshold (THRESHOLD)

Set threshold for video processing.

Parameter: double SP_Threshold

SP_Threshold

Direction: Down

Valid values:

- Minimum:** 0.0
Maximum: 99.0

Unit: %

Description: Video threshold.

11.3.2.2.6 Get_Threshold (THRESHOLD)

Get threshold for video processing.

Parameter: double SA_Threshold
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 99.0
Unit: %
Description: Video threshold.

11.3.2.3 Material database

11.3.2.3.1 Get_MaterialTable (MATERIALTABLE)

Get a list of all materials for thickness calculation.

Parameter: String SA_MaterialTable
Direction: Up
Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_MaterialTableCount
Direction: Up
Valid values:
Minimum: 0
Maximum: 20
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Pos1, SA_Pos2, ...

Parameter: int32_t SA_Pos1..x
Direction: Up
Valid values:
Minimum: 0
Maximum: 19
Description: Index of the material in the table.

Parameter: String SA_MaterialName1..x
Direction: Up
Description: Name of the material in the table.

Parameter: double SA_RefractiveIndex_nF1..x
Direction: Up
Valid values:
Minimum: 1.0
Maximum: 4.0
Description: Refractive index of the material at 486 nm.

Parameter: double SA_RefractiveIndex_nd1..x
Direction: Up
Valid values:
Minimum: 1.0
Maximum: 4.0
Description: Refractive index of the material at 587 nm.

Parameter: double SA_RefractiveIndex_nC1..x	SA_RefractiveIndex_nC1..x
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 656 nm.	
Parameter: double SA_AbbeNumber_vd1..x	SA_AbbeNumber_vd1..x
Direction: Up	
Valid values:	
Minimum: 10.0	
Maximum: 100.0	
Description: Abbe number of the material or 0.0 if not specified.	
Parameter: String SA_Description1..x	SA_Description1..x
Direction: Up	
Description: Description of the material in the table.	

11.3.2.3.2 Set_ActiveMaterial (MATERIAL)

Set the active material for thickness calculation.

Parameter: String SP_ActiveMaterial	SP_ActiveMaterial
Direction: Down	
Description: Name of material.	

11.3.2.3.3 Get_ActiveMaterial (MATERIAL)

Get the active material for thickness calculation.

Parameter: String SA_ActiveMaterial	SA_ActiveMaterial
Direction: Up	
Description: Name of material.	

11.3.2.3.4 Get_MaterialInfo (MATERIALINFO)

Get information of active material.

Parameter: int32_t SP_MaterialIndex	SP_MaterialIndex
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 5	
Default: Empty, means first material.	
Description: Index of material to get info. If this parameter is 0 or not set, MATERIALINFO without parameter is called (for compatibility mode)	

Parameter: String SA_MaterialName	SA_MaterialName
Direction: Up	
Description: Name of the active material.	

Parameter: String SA_Description	SA_Description
Direction: Up	
Description: Description of the active material.	
Parameter: double SA_RefRACTiveIndex_nF	SA_RefRACTiveIndex_nF
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 486 nm.	
Parameter: double SA_RefRACTiveIndex_nd	SA_RefRACTiveIndex_nd
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 587 nm.	
Parameter: double SA_RefRACTiveIndex_nC	SA_RefRACTiveIndex_nC
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the active material at 656 nm.	
Parameter: double SA_AbbeNumber_vd	SA_AbbeNumber_vd
Direction: Up	
Valid values:	
Minimum: 10.0	
Maximum: 100.0	
Description: Abbe number of the active material or 0.0 if not specified.	
Parameter: int32_t SA_Described_by	SA_Described_by
Direction: Up	
Valid values:	
0= nF, nd and nC	
1= nd and abbe value	
Description: Tells if nF, nd and nC or if nd and abbe value is valid.	

11.3.2.3.5 Edit_Material_Abbe (MATERIALEDIT)

Edit or add new material by using Abbe number.

Parameter: String SP_MaterialName	SP_MaterialName
Direction: Down	
Description: Name of the material.	
Parameter: String SP_Description	SP_Description
Direction: Down	
Description: Description of the material.	

Parameter: double SP_RefractiveIndex_nd	SP_RefractiveIndex_nd
Direction: Down	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 587 nm.	
Parameter: double SP_AbbeNumber_vd	SP_AbbeNumber_vd
Direction: Down	
Valid values:	
Minimum: 10.0	
Maximum: 100.0	
Description: Abbe number of the material.	

11.3.2.3.6 Edit_Material_Nx (MATERIALEDIT)

Edit or add new material by using three refractive indices.

Parameter: String SP_MaterialName	SP_MaterialName
Direction: Down	
Description: Name of the material.	
Parameter: String SP_Description	SP_Description
Direction: Down	
Description: Description of the material.	
Parameter: double SP_RefractiveIndex_nF	SP_RefractiveIndex_nF
Direction: Down	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 486 nm.	
Parameter: double SP_RefractiveIndex_nd	SP_RefractiveIndex_nd
Direction: Down	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 587 nm.	
Parameter: double SP_RefractiveIndex_nC	SP_RefractiveIndex_nC
Direction: Down	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Refractive index of the material at 656 nm.	

11.3.2.3.7 Delete_Material (MATERIALDELETE)

Deletes an existing material.

Parameter: String SP_MaterialName	SP_MaterialName
Direction: Down	
Description: Name of the material to delete.	

11.3.2.3.8 Clear_MaterialTable

Clear the whole material table.

11.3.2.4 Peak processing

11.3.2.4.1 Set_RefractiveCorrection (REFRACCORR)

Specify refractive correction and number of peaks to detect.

At this command the controller may change output data automatically. If first bit of `IP_AutomaticMode` is set (1), `Get_AllParameters` (SP_Additional= 0) is called automatically after this command. Otherwise, you have to call it manually. After this call `Get_TransmittedDataInfo` to retrieve new setting.

Parameter: int32_t SP_RefractiveCorrection

SP_RefractiveCorrection

Direction: Down

Valid values:

0= Off
1= On

Description: Specify if refractive correction should be enabled.

Parameter: int32_t SP_NumberOfPeaks

SP_NumberOfPeaks

Direction: Down

Valid values:

Minimum: 2
Maximum: 6

Description: Number of peaks to detect.

11.3.2.4.2 Get_RefractiveCorrection (REFRACCORR)

Tells if refractive correction is enabled and number of peaks to detect.

Parameter: int32_t SA_RefractiveCorrection

SA_RefractiveCorrection

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= Off
1= On

Description: Tells if refractive correction is be enabled.

Parameter: int32_t SA_NumberOfPeaks

SA_NumberOfPeaks

Direction: Up

Valid values:

Minimum: 2
Maximum: 6

Description: Number of peaks to detect.

11.3.2.4.3 Set_MaterialMultiPeak (MATERIALMP)

Set up to five materials for multipeak calculation.

For this command an [Update ...](#) meta command is available.

Valid for sensor:

- IFD2451 with MultiLayer option
- IFD2461 with MultiLayer option
- IFD2471 with MultiLayer option

Parameter: String SP_MaterialMultiPeak12

SP_MaterialMultiPeak12

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between first and second peak.

Parameter: String SP_MaterialMultiPeak23

SP_MaterialMultiPeak23

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between second and third peak.

Parameter: String SP_MaterialMultiPeak34

SP_MaterialMultiPeak34

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between third and fourth peak.

Parameter: String SP_MaterialMultiPeak45

SP_MaterialMultiPeak45

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between fourth and fifth peak.

Parameter: String SP_MaterialMultiPeak56

SP_MaterialMultiPeak56

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between fifth and sixth peak.

11.3.2.4.4 Get_MaterialMultiPeak (MATERIALMP)

Get all material names for multipeak calculation.

Valid for sensor:

- IFD2451 with MultiLayer option
- IFD2461 with MultiLayer option
- IFD2471 with MultiLayer option

Parameter: String SA_MaterialMultiPeak12

SA_MaterialMultiPeak12

Direction: Up

Description: Name of material between first and second peak.

Parameter: String SA_MaterialMultiPeak23

SA_MaterialMultiPeak23

Direction: Up

Description: Name of material between second and third peak.

Parameter: String SA_MaterialMultiPeak34	SA_MaterialMultiPeak34
Direction: Up	
Description: Name of material between third and fourth peak.	
Parameter: String SA_MaterialMultiPeak45	SA_MaterialMultiPeak45
Direction: Up	
Description: Name of material between fourth and fifth peak.	
Parameter: String SA_MaterialMultiPeak56	SA_MaterialMultiPeak56
Direction: Up	
Description: Name of material between fifth and sixth peak.	

11.3.2.5 Measurement value processing

11.3.2.5.1 Set_Averaging (AVERAGE)

Set data averaging at controller.

Parameter: int32_t SP_AveragingType	SP_AveragingType
Direction: Down	
Valid values:	
0= None	
1= Moving average (MOVING)	
2= Recursive averaging (RECURSIVE)	
3= Median	
Description: Averaging type.	

Parameter: int32_t SP_MovingCount	SP_MovingCount
Direction: Down	
Valid values:	
2	
4	
8	
16	
32	
64	
128	
256	
512	
1024	
Description: Number of value for the averaging window. This parameter is only used at moving average.	

Parameter: int32_t SP_RecursiveCount	SP_RecursiveCount
Direction: Down	
Valid values:	
Minimum: 2	
Maximum: 32768	
Description: Number of values for recursive averaging. This parameter is only used at recursive average.	

Parameter: int32_t SP_MedianCount SP_MedianCount

Direction: Down

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only used at median.

11.3.2.5.2 Get_Averaging (AVERAGE)

Get data averaging at controller.

Parameter: int32_t SA_AveragingType SA_AveragingType

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = None
- 1 = Moving average (MOVING)
- 2 = Recursive averaging (RECURSIVE)
- 3 = Median

Description: Averaging type.

Parameter: int32_t SA_MovingCount SA_MovingCount

Direction: Up

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128
- 256
- 512
- 1024

Description: Number of value for the averaging window. This parameter is only available at moving average.

Parameter: int32_t SA_RecursiveCount SA_RecursiveCount

Direction: Up

Valid values:

Minimum: 2
Maximum: 32768

Description: Number of values for recursive averaging. This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount SA_MedianCount

Direction: Up

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only available at median.

11.3.2.5.3 Set_SpikeCorrection (SPIKECORR)

Set spike correction at controller.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_SpikeCorrection	SP_SpikeCorrection
Direction: Down	
Valid values:	
0= off	
1= on	
Description: Spike correction.	
Parameter: int32_t SP_NbrEvaluatedValues	SP_NbrEvaluatedValues
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Number of values to evaluate for spike correction.	
Parameter: double SP_ToleranceRange	SP_ToleranceRange
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 100.0	
Description: Tolerance range for spike correction.	
Parameter: int32_t SP_NbrCorrectedValues	SP_NbrCorrectedValues
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 100	
Description: Number of values to correct at spike correction.	

11.3.2.5.4 Get_SpikeCorrection (SPIKECORR)

Get spike correction at controller.

Parameter: int32_t SA_SpikeCorrection	SA_SpikeCorrection
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= off	
1= on	
Description: Spike correction.	
Parameter: int32_t SA_NbrEvaluatedValues	SA_NbrEvaluatedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Number of values to evaluate for spike correction.	

Parameter: double SA_ToleranceRange	SA_ToleranceRange
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 100.0	
Description: Tolerance range for spike correction.	
Parameter: int32_t SA_NbrCorrectedValues	SA_NbrCorrectedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100	
Description: Number of values to correct at spike correction.	

11.3.2.5.5 Set_StatisticSignal (STATISTICSIGNAL)

Set the measured value which is used for statistic calculation.

In measure mode MultiLayer and data output interface Ethernet the controller may change output data automatically. Next call to [DataAvail](#) or [TransferData](#) will return a warning. In this case call [Get_TransmittedDataInfo](#) to retrieve new setting.

Valid for sensor:

- IFD2451 with MultiLayer option
- IFD2461 with MultiLayer option
- IFD2471 with MultiLayer option

Parameter: int32_t SP_StatisticSignal	SP_StatisticSignal
Direction: Down	
Valid values:	
0= Distance 1	
1= Distance 2	
2= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)	
3= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)	
4= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)	
5= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)	
6= Thickness 12	
7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)	
8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)	
9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)	
10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)	
11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)	
12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)	
13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)	
14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)	
15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)	
16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)	
17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)	
18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)	
19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)	
20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)	
Description: Value which is used for statistic calculation.	

11.3.2.5.6 Get_StatisticSignal (STATISTICSIGNAL)

Get the measured value which is used for statistic calculation.

Valid for sensor:

- IFD2451 with MultiLayer option
- IFD2461 with MultiLayer option
- IFD2471 with MultiLayer option

Parameter: int32_t SA_StatisticSignal

SA_StatisticSignal

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Distance 1
- 1= Distance 2
- 2= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)
- 3= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)
- 4= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)
- 5= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)
- 6= Thickness 12
- 7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)
- 8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)
- 9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)
- 10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)
- 11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)
- 12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)
- 13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)
- 14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)
- 15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)
- 16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)
- 17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)
- 18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)
- 19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)
- 20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)

Description: Value which is used for statistic calculation.

11.3.2.5.7 Set_StatisticDepth (STATISTICDEPTH)

Set the window size for floating statistic calculation.

Parameter: int32_t SP_StatisticDepth

SP_StatisticDepth

Direction: Down

Valid values:

Minimum: 2

Maximum: 2147483647 (INT32_MAX)

Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 16384). Value greater as 16384 calculates statistic over all values.

11.3.2.5.8 Get_StatisticDepth (STATISTICDEPTH)

Get the window size for floating statistic calculation.

Parameter: int32_t SA_StatisticDepth

SA_StatisticDepth

Direction: Up

Valid values:

Minimum: 2

Maximum: 2147483647 (INT32_MAX)

Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 16384). Value greater as 16384 calculates statistic over all values.

11.3.2.5.9 Reset_Statistic (RESETSTATISTIC)

Reset the statistic (min, max and peak to peak values).

11.3.2.5.10 Set_MasterSignal (MASTERSIGNAL)

Set the measured value which is used for mastering.

Valid for sensor:

IFD2451 with MultiLayer option

IFD2461 with MultiLayer option

IFD2471 with MultiLayer option

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

0= Distance 1

1= Distance 2

2= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)

3= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)

4= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)

5= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)

6= Thickness 12

7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)

8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)

9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)

10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)

11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)

12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)

13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)

14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)

15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)

16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)

17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)

18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)

19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)

20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)

Description: Value which is used for mastering.

11.3.2.5.11 Get_MasterSignal (MASTERSIGNAL)

Get the measured value which is used for mastering.

Valid for sensor:

- IFD2451 with MultiLayer option
- IFD2461 with MultiLayer option
- IFD2471 with MultiLayer option

Parameter: int32_t SA_MasterSignal

SA_MasterSignal

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Distance 1
- 1= Distance 2
- 2= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)
- 3= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)
- 4= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)
- 5= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)
- 6= Thickness 12
- 7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)
- 8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)
- 9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)
- 10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)
- 11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)
- 12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)
- 13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)
- 14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)
- 15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)
- 16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)
- 17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)
- 18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)
- 19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)
- 20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)

Description: Value which is used for mastering.

11.3.2.5.12 Set_MasterValue (MASTERMV)

Set the master value.

Parameter: int32_t SP_Master

SP_Master

Direction: Down

Valid values:

- 0= no (NONE)
- 1= yes (MASTER)

Description: Specifies if mastering should be done or resetted.

Parameter: double SP_MasterValue

SP_MasterValue

Direction: Down

Valid values:

- Minimum:** -2* measuring range
- Maximum:** +2* measuring range

Unit: mm

Description: Master value

11.3.2.5.13 Get_MasterValue (MASTERMV)

Get the master value.

Parameter: int32_t SA_Master

SA_Master

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = no (NONE)
- 1 = yes (MASTER)

Description: Specifies if mastering is active.

Parameter: double SA_MasterValue

SA_MasterValue

Direction: Up

Valid values:

- Minimum:** -2* measuring range
Maximum: +2* measuring range

Unit: mm

Description: Master value

11.3 Data output

11.3.3.1 General

11.3.3.1.1 Set_DataOutInterface (OUTPUT)

Set the active interface for data output.

At this command the controller may change output data automatically. If first bit of `IP_AutomaticMode` is set (1), `Get_AllParameters` (SP_Additional= 0) is called automatically after this command. Otherwise, you have to call it manually. After this call `Get_TransmittedDataInfo` to retrieve new setting.

Parameter: int32_t SP_DataOutInterface

SP_DataOutInterface

Direction: Down

Valid values:

- 0 = None
- 1 = RS422
- 2 = Ethernet
- 3 = HTTP
- 4 = Ethercat

Description: Active interface for data output.

11.3.3.1.2 Get_DataOutInterface (OUTPUT)

Get the active interface for data output.

Parameter: int32_t SA_DataOutInterface

SA_DataOutInterface

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = None
- 1 = RS422
- 2 = Ethernet
- 3 = HTTP
- 4 = Ethercat

Description: Active interface for data output.

11.3.3.1.3 Set_Resampling (OUTREDUCE)

Set resampling to reduce output data.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_Resampling	SP_Resampling
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 4200000	
Description: Resampling value.	
Parameter: int32_t SP_ResampleAnalog	SP_ResampleAnalog
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if analog output should be resampled.	
Parameter: int32_t SP_ResampleRS422	SP_ResampleRS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if RS422 output should be resampled.	
Parameter: int32_t SP_ResampleEthernet	SP_ResampleEthernet
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if output over ethernet should be resampled.	

11.3.3.1.4 Get_Resampling (OUTREDUCE)

Get resampling for reducing output data.

Parameter: int32_t SA_Resampling	SA_Resampling
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 4200000	
Description: Resampling value.	
Parameter: int32_t SA_ResampleAnalog	SA_ResampleAnalog
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Analog output is resampled.	

Parameter: int32_t SA_ResampleRS422	SA_ResampleRS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: RS422 output is resampled.	
Parameter: int32_t SA_ResampleEthernet	SA_ResampleEthernet
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Output over ethernet is resampled.	

11.3.3.1.5 Set_HoldLastValid (OUTHOLD)

Set the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SP_HoldLastValid	SP_HoldLastValid
Direction: Down	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	

11.3.3.1.6 Get_HoldLastValid (OUTHOLD)

Get the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SA_HoldLastValid	SA_HoldLastValid
Direction: Up	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	

11.3.3.1.7 Set_FramesPerPacket_ETH (MEASCNT_ETH)

Set the maximum number of frames in ethernet packet.

Parameter: int32_t SP_FramesPerPacket_ETH	SP_FramesPerPacket_ETH
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 1	
Description: Maximum number of frames in ethernet packet. 0 means automatic.	

11.3.3.1.8 Get_FramesPerPacket_ETH (MEASCNT_ETH)

Get the maximum number of frames in ethernet packet.

Parameter: int32_t SA_FramesPerPacket_ETH

SA_FramesPerPacket_ETH

Direction: Up

Valid values:

Minimum: 0

Maximum: 1

Description: Maximum number of frames in ethernet packet. 0 means automatic.

11.3.3.2 Selected measurement values

11.3.3.2.1 Set_OutputDistance_RS422 (OUTDIST_RS422)

Set the distance data to be output at RS422 interface.

At this command the controller may change output data (additional to specified parameters). If first bit of [IP_AutomaticMode](#) is set (1), [Get_OutputDistance_RS422](#) is called automatically after this command. Otherwise, you have to call it manually.

After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputDistance1_RS422

SP_OutputDistance1_RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if distance 1 is transmitted.

Parameter: int32_t SP_OutputDistance2_RS422

SP_OutputDistance2_RS422

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if distance 2 is transmitted (only available at measure mode thickness, multipeak or video).

Parameter: int32_t SP_OutputDistance3_RS422

SP_OutputDistance3_RS422

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if distance 3 is transmitted (only available at measure mode multipeak or video).

Parameter: int32_t SP_OutputDistance4_RS422	SP_OutputDistance4_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 4 is transmitted (only available at measure mode multipeak or video).	
Parameter: int32_t SP_OutputDistance5_RS422	SP_OutputDistance5_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 5 is transmitted (only available at measure mode multipeak or video).	
Parameter: int32_t SP_OutputDistance6_RS422	SP_OutputDistance6_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 6 is transmitted (only available at measure mode multipeak or video).	

11.3.3.2.2 Get_OutputDistance_RS422 (OUTDIST_RS422)

Get the distance data to be output at RS422 interface.

Parameter: int32_t SA_OutputDistance1_RS422	SA_OutputDistance1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 is transmitted.	
Parameter: int32_t SA_OutputDistance2_RS422	SA_OutputDistance2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 2 is transmitted.	

Parameter: int32_t SA_OutputDistance3_RS422	SA_OutputDistance3_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if distance 3 is transmitted.
Parameter: int32_t SA_OutputDistance4_RS422	SA_OutputDistance4_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if distance 4 is transmitted.
Parameter: int32_t SA_OutputDistance5_RS422	SA_OutputDistance5_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if distance 5 is transmitted.
Parameter: int32_t SA_OutputDistance6_RS422	SA_OutputDistance6_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if distance 6 is transmitted.

11.3.3.2.3 Set_OutputDistance_ETH (OUTDIST_ETH)

Set the distance data to be output at ethernet interface.

This command has only effect on measure mode video or multipeak.

In measure mode MultiLayer and data output interface Ethernet the controller may change output data (additional to specified parameters). Next call to [DataAvail](#) or [TransferData](#) will return a warning. In this case call [Get_TransmittedDataInfo](#) to retrieve new setting.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputDistance1_ETH	SP_OutputDistance1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if distance 1 is transmitted.	
Parameter: int32_t SP_OutputDistance2_ETH	SP_OutputDistance2_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 2 is transmitted (only available at measure mode thickness, multipeak or video).	
Parameter: int32_t SP_OutputDistance3_ETH	SP_OutputDistance3_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 3 is transmitted (only available at measure mode multipeak or video).	
Parameter: int32_t SP_OutputDistance4_ETH	SP_OutputDistance4_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 4 is transmitted (only available at measure mode multipeak or video).	
Parameter: int32_t SP_OutputDistance5_ETH	SP_OutputDistance5_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if distance 5 is transmitted (only available at measure mode multipeak or video).	

Parameter: int32_t SP_OutputDistance6_ETH SP_OutputDistance6_ETH
Direction: Down
Valid values:
 0= no
 1= yes
Valid for sensor:
 IFD2451
 IFD2461
 IFD2471
Description: Specify if distance 6 is transmitted (only available at measure mode multipeak or video).

11.3.3.2.4 Get_OutputDistance_ETH (OUTDIST_ETH)

Get the distance data to be output at ethernet interface.
 This command has only effect on measure mode video or multipeak.

Parameter: int32_t SA_OutputDistance1_ETH SA_OutputDistance1_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if distance 1 is transmitted.

Parameter: int32_t SA_OutputDistance2_ETH SA_OutputDistance2_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Valid for sensor:
 IFD2451
 IFD2461
 IFD2471
Description: Specify if distance 2 is transmitted.

Parameter: int32_t SA_OutputDistance3_ETH SA_OutputDistance3_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Valid for sensor:
 IFD2451
 IFD2461
 IFD2471
Description: Specify if distance 3 is transmitted.

Parameter: int32_t SA_OutputDistance4_ETH SA_OutputDistance4_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Valid for sensor:
 IFD2451
 IFD2461
 IFD2471
Description: Specify if distance 4 is transmitted.

Parameter: int32_t SA_OutputDistance5_ETH	SA_OutputDistance5_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if distance 5 is transmitted.
Parameter: int32_t SA_OutputDistance6_ETH	SA_OutputDistance6_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if distance 6 is transmitted.

11.3.3.2.5 Set_OutputThickness_RS422 (OUTTHICK_RS422)

Set the thickness data to be output at RS422 interface.

At this command the controller may change output data (additional to specified parameters). If first bit of [IP_AutomaticMode](#) is set (1), [Get_OutputThickness_RS422](#) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputThickness12_RS422	SP_OutputThickness12_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if thickness between first and second peak is transmitted (only available at measure mode thickness, multipeak or video).
Parameter: int32_t SP_OutputThickness13_RS422	SP_OutputThickness13_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between first and third peak is transmitted (only available at measure mode multipeak or video).

Parameter: int32_t SP_OutputThickness14_RS422	SP_OutputThickness14_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between first and fourth peak is transmitted (only available at measure mode multipeak or video).	
Parameter: int32_t SP_OutputThickness15_RS422	SP_OutputThickness15_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between first and fifth peak is transmitted (only available at measure mode multipeak or video).	
Parameter: int32_t SP_OutputThickness16_RS422	SP_OutputThickness16_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between first and sixth peak is transmitted (only available at measure mode multipeak or video).	
Parameter: int32_t SP_OutputThickness23_RS422	SP_OutputThickness23_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and third peak is transmitted (only available at measure mode multipeak or video).	
Parameter: int32_t SP_OutputThickness24_RS422	SP_OutputThickness24_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between second and fourth peak is transmitted (only available at measure mode multipeak or video).

Parameter: int32_t SP_OutputThickness25_RS422

SP_OutputThickness25_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between second and fifth peak is transmitted (only available at measure mode multipeak or video).

Parameter: int32_t SP_OutputThickness26_RS422

SP_OutputThickness26_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between second and sixth peak is transmitted (only available at measure mode multipeak or video).

Parameter: int32_t SP_OutputThickness34_RS422

SP_OutputThickness34_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between third and fourth peak is transmitted (only available at measure mode multipeak or video).

Parameter: int32_t SP_OutputThickness35_RS422

SP_OutputThickness35_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between third and fifth peak is transmitted (only available at measure mode multipeak or video).

Parameter: int32_t SP_OutputThickness36_RS422	SP_OutputThickness36_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between third and sixth peak is transmitted (only available at measure mode multipeak or video).	
Parameter: int32_t SP_OutputThickness45_RS422	SP_OutputThickness45_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between fourth and fifth peak is transmitted (only available at measure mode multipeak or video).	
Parameter: int32_t SP_OutputThickness46_RS422	SP_OutputThickness46_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between fourth and sixth peak is transmitted (only available at measure mode multipeak or video).	
Parameter: int32_t SP_OutputThickness56_RS422	SP_OutputThickness56_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between fifth and sixth peak is transmitted (only available at measure mode multipeak or video).	

11.3.3.2.6 Get_OutputThickness_RS422 (OUTTHICK_RS422)

Get the thickness data to be output at RS422 interface.

Parameter: int32_t SA_OutputThickness12_RS422	SA_OutputThickness12_- RS422
------------------------------------------------------	---------------------------------

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if thickness between first and second peak is transmitted.

Parameter: int32_t SA_OutputThickness13_RS422

SA_OutputThickness13_-
RS422

Direction: Up

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between first and third peak is transmitted.

Parameter: int32_t SA_OutputThickness14_RS422

SA_OutputThickness14_-
RS422

Direction: Up

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between first and fourth peak is transmitted.

Parameter: int32_t SA_OutputThickness15_RS422

SA_OutputThickness15_-
RS422

Direction: Up

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between first and fifth peak is transmitted.

Parameter: int32_t SA_OutputThickness16_RS422

SA_OutputThickness16_-
RS422

Direction: Up

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between first and sixth peak is transmitted.

Parameter: int32_t SA_OutputThickness23_RS422	SA_OutputThickness23_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and third peak is transmitted.	
Parameter: int32_t SA_OutputThickness24_RS422	SA_OutputThickness24_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and fourth peak is transmitted.	
Parameter: int32_t SA_OutputThickness25_RS422	SA_OutputThickness25_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and fifth peak is transmitted.	
Parameter: int32_t SA_OutputThickness26_RS422	SA_OutputThickness26_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and sixth peak is transmitted.	
Parameter: int32_t SA_OutputThickness34_RS422	SA_OutputThickness34_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between third and fourth peak is transmitted.	

Parameter: int32_t SA_OutputThickness35_RS422	SA_OutputThickness35_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between third and fifth peak is transmitted.
Parameter: int32_t SA_OutputThickness36_RS422	SA_OutputThickness36_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between third and sixth peak is transmitted.
Parameter: int32_t SA_OutputThickness45_RS422	SA_OutputThickness45_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between fourth and fifth peak is transmitted.
Parameter: int32_t SA_OutputThickness46_RS422	SA_OutputThickness46_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between fourth and sixth peak is transmitted.
Parameter: int32_t SA_OutputThickness56_RS422	SA_OutputThickness56_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between fifth and sixth peak is transmitted.

11.3.3.2.7 Set_OutputThickness_ETH (OUTTHICK_ETH)

Set the thickness data to be output at ethernet interface.

This command has only effect on measure mode multipeak or video.

In measure mode MultiLayer and data output interface Ethernet the controller may change output data (additional to specified parameters). Next call to [DataAvail](#) or [TransferData](#) will return a warning. In this case call [Get_TransmittedDataInfo](#) to retrieve new setting.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputThickness12_ETH

SP_OutputThickness12_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if thickness between first and second peak is transmitted.

Parameter: int32_t SP_OutputThickness13_ETH

SP_OutputThickness13_ETH

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between first and third peak is transmitted.

Parameter: int32_t SP_OutputThickness14_ETH

SP_OutputThickness14_ETH

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between first and fourth peak is transmitted.

Parameter: int32_t SP_OutputThickness15_ETH

SP_OutputThickness15_ETH

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

IFD2451

IFD2461

IFD2471

Description: Specify if thickness between first and fifth peak is transmitted.

Parameter: int32_t SP_OutputThickness16_ETH	SP_OutputThickness16_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between first and sixth peak is transmitted.	
Parameter: int32_t SP_OutputThickness23_ETH	SP_OutputThickness23_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and third peak is transmitted.	
Parameter: int32_t SP_OutputThickness24_ETH	SP_OutputThickness24_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and fourth peak is transmitted.	
Parameter: int32_t SP_OutputThickness25_ETH	SP_OutputThickness25_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and fifth peak is transmitted.	
Parameter: int32_t SP_OutputThickness26_ETH	SP_OutputThickness26_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and sixth peak is transmitted.	

Parameter: int32_t SP_OutputThickness34_ETH	SP_OutputThickness34_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between third and fourth peak is transmitted.
Parameter: int32_t SP_OutputThickness35_ETH	SP_OutputThickness35_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between third and fifth peak is transmitted.
Parameter: int32_t SP_OutputThickness36_ETH	SP_OutputThickness36_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between third and sixth peak is transmitted.
Parameter: int32_t SP_OutputThickness45_ETH	SP_OutputThickness45_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between fourth and fifth peak is transmitted.
Parameter: int32_t SP_OutputThickness46_ETH	SP_OutputThickness46_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between fourth and sixth peak is transmitted.

Parameter: int32_t SP_OutputThickness56_ETH SP_OutputThickness56_ETH

Direction: Down

Valid values:

- 0= no
- 1= yes

Valid for sensor:

- IFD2451
- IFD2461
- IFD2471

Description: Specify if thickness between fifth and sixth peak is transmitted.

11.3.3.2.8 Get_OutputThickness_ETH (OUTTHICK_ETH)

Get the thickness data to be output at ethernet interface.
This command has only effect on measure mode video.

Parameter: int32_t SA_OutputThickness12_ETH SA_OutputThickness12_ETH

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: Specify if thickness between first and second peak is transmitted.

Parameter: int32_t SA_OutputThickness13_ETH SA_OutputThickness13_ETH

Direction: Up

Valid values:

- 0= no
- 1= yes

Valid for sensor:

- IFD2451
- IFD2461
- IFD2471

Description: Specify if thickness between first and third peak is transmitted.

Parameter: int32_t SA_OutputThickness14_ETH SA_OutputThickness14_ETH

Direction: Up

Valid values:

- 0= no
- 1= yes

Valid for sensor:

- IFD2451
- IFD2461
- IFD2471

Description: Specify if thickness between first and fourth peak is transmitted.

Parameter: int32_t SA_OutputThickness15_ETH SA_OutputThickness15_ETH

Direction: Up

Valid values:

- 0= no
- 1= yes

Valid for sensor:

- IFD2451
- IFD2461
- IFD2471

Description: Specify if thickness between first and fifth peak is transmitted.

Parameter: int32_t SA_OutputThickness16_ETH	SA_OutputThickness16_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between first and sixth peak is transmitted.	
Parameter: int32_t SA_OutputThickness23_ETH	SA_OutputThickness23_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and third peak is transmitted.	
Parameter: int32_t SA_OutputThickness24_ETH	SA_OutputThickness24_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and fourth peak is transmitted.	
Parameter: int32_t SA_OutputThickness25_ETH	SA_OutputThickness25_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and fifth peak is transmitted.	
Parameter: int32_t SA_OutputThickness26_ETH	SA_OutputThickness26_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description: Specify if thickness between second and sixth peak is transmitted.	

Parameter: int32_t SA_OutputThickness34_ETH	SA_OutputThickness34_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between third and fourth peak is transmitted.
Parameter: int32_t SA_OutputThickness35_ETH	SA_OutputThickness35_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between third and fifth peak is transmitted.
Parameter: int32_t SA_OutputThickness36_ETH	SA_OutputThickness36_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between third and sixth peak is transmitted.
Parameter: int32_t SA_OutputThickness45_ETH	SA_OutputThickness45_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between fourth and fifth peak is transmitted.
Parameter: int32_t SA_OutputThickness46_ETH	SA_OutputThickness46_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
IFD2451	
IFD2461	
IFD2471	
Description:	Specify if thickness between fourth and sixth peak is transmitted.

Parameter: int32_t SA_OutputThickness56_ETH SA_OutputThickness56_ETH

Direction: Up

Valid values:

- 0= no
- 1= yes

Valid for sensor:

- IFD2451
- IFD2461
- IFD2471

Description: Specify if thickness between fifth and sixth peak is transmitted.

11.3.3.2.9 Set_OutputStatistic_RS422 (OUTSTATISTIC_RS422)

Set the statistic data to be output at RS422 interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputStatisticMin_RS422 SP_OutputStatisticMin_RS422

Direction: Down

Valid values:

- 0= no
- 1= yes

Description: Specify if min value is transmitted.

Parameter: int32_t SP_OutputStatisticMax_RS422 SP_OutputStatisticMax_RS422

Direction: Down

Valid values:

- 0= no
- 1= yes

Description: Specify if max value is transmitted.

Parameter: int32_t SP_OutputStatisticPeak2Peak_RS422 SP_OutputStatisticPeak2Peak_RS422

Direction: Down

Valid values:

- 0= no
- 1= yes

Description: Specify if peak to peak value is transmitted.

11.3.3.2.10 Get_OutputStatistic_RS422 (OUTSTATISTIC_RS422)

Get the statistic data to be output at RS422 interface.

Parameter: int32_t SA_OutputStatisticMin_RS422 SA_OutputStatisticMin_RS422

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: Specify if min value is transmitted.

Parameter: int32_t SA_OutputStatisticMax_RS422 SA_OutputStatisticMax_RS422

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: Specify if max value is transmitted.

Parameter: int32_t SA_OutputStatisticPeak2Peak_RS422 SA_OutputStatisticPeak2Peak_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if peak to peak value is transmitted.

11.3.3.2.11 Set_OutputStatistic_ETH (OUTSTATISTIC_ETH)

Set the statistic data to be output at ethernet interface.
 In measure mode MultiLayer and data output interface Ethernet the controller may change output data (additional to specified parameters). Next call to [DataAvail](#) or [TransferData](#) will return a warning. In this case call [Get_TransmittedDataInfo](#) to retrieve new setting.
 For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputStatisticMin_ETH SP_OutputStatisticMin_ETH
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if min value is transmitted.

Parameter: int32_t SP_OutputStatisticMax_ETH SP_OutputStatisticMax_ETH
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if max value is transmitted.

Parameter: int32_t SP_OutputStatisticPeak2Peak_ETH SP_OutputStatisticPeak2Peak_ETH
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if peak to peak value is transmitted.

11.3.3.2.12 Get_OutputStatistic_ETH (OUTSTATISTIC_ETH)

Get the statistic data to be output at ethernet interface.

Parameter: int32_t SA_OutputStatisticMin_ETH SA_OutputStatisticMin_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if min value is transmitted.

Parameter: int32_t SA_OutputStatisticMax_ETH SA_OutputStatisticMax_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if max value is transmitted.

Parameter: int32_t SA_OutputStatisticPeak2Peak_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if peak to peak value is transmitted.

SA_OutputStatistic-Peak2Peak_ETH

11.3.3.2.13 Set_OutputAdditional_RS422 (OUTADD_RS422)

Set the additional data to be output at RS422 interface.
 For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputAdditionalShutterTime_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if shutter time is transmitted.

SP_OutputAdditionalShutterTime_RS422

Parameter: int32_t SP_OutputAdditionalEncoder1_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if encoder 1 is transmitted.

SP_OutputAdditionalEncoder1_RS422

Parameter: int32_t SP_OutputAdditionalEncoder2_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if encoder 2 is transmitted.

SP_OutputAdditionalEncoder2_RS422

Parameter: int32_t SP_OutputAdditionalEncoder3_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if encoder 3 is transmitted.

SP_OutputAdditionalEncoder3_RS422

Parameter: int32_t SP_OutputAdditionalCounter_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if counter is transmitted.

SP_OutputAdditionalCounter_RS422

Parameter: int32_t SP_OutputAdditionalTimestamp_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if timestamp is transmitted.

SP_OutputAdditionalTimestamp_RS422

Parameter: int32_t SP_OutputAdditionalIntensity_RS422	SP_OutputAdditionalIntensity_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity is transmitted.	
Parameter: int32_t SP_OutputAdditionalState_RS422	SP_OutputAdditionalState_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SP_OutputAdditionalTrgTimeDiff_RS422	SP_OutputAdditionalTrgTimeDiff_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger time difference is transmitted.	
Parameter: int32_t SP_OutputAdditionalMeasrate_RS422	SP_OutputAdditionalMeasrate_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measrate is transmitted.	

11.3.3.2.14 Get_OutputAdditional_RS422 (OUTADD_RS422)

Get the additional data to be output at RS422 interface.

Parameter: int32_t SA_OutputAdditionalShutterTime_RS422	SA_OutputAdditionalShutterTime_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1_RS422	SA_OutputAdditionalEncoder1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder2_RS422	SA_OutputAdditionalEncoder2_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 2 is transmitted.	

Parameter: int32_t SA_OutputAdditionalEncoder3_RS422	SA_OutputAdditionalEncoder3_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 3 is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_RS422	SA_OutputAdditionalIntensity_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalTrgTimeDiff_RS422	SA_OutputAdditionalTrgTimeDiff_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger time difference is transmitted.	
Parameter: int32_t SA_OutputAdditionalMeasrate_RS422	SA_OutputAdditionalMeasrate_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measrate is transmitted.	

11.3.3.2.15 Set_OutputAdditional_ETH (OUTADD_ETH)

Set the additional data to be output at ethernet interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputAdditionalShutterTime_ETH

SP_OutputAdditionalShutterTime_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if shutter time is transmitted.

Parameter: int32_t SP_OutputAdditionalEncoder1_ETH

SP_OutputAdditionalEncoder1_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if encoder 1 is transmitted.

Parameter: int32_t SP_OutputAdditionalEncoder2_ETH

SP_OutputAdditionalEncoder2_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if encoder 2 is transmitted.

Parameter: int32_t SP_OutputAdditionalEncoder3_ETH

SP_OutputAdditionalEncoder3_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if encoder 3 is transmitted.

Parameter: int32_t SP_OutputAdditionalCounter_ETH

SP_OutputAdditionalCounter_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if counter is transmitted.

Parameter: int32_t SP_OutputAdditionalTimestamp_ETH

SP_OutputAdditionalTimestamp_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if timestamp is transmitted.

Parameter: int32_t SP_OutputAdditionalIntensity_ETH

SP_OutputAdditionalIntensity_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if intensity is transmitted.

Parameter: int32_t SP_OutputAdditionalState_ETH	SP_OutputAdditionalState_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SP_OutputAdditionalTrgTimeDiff_ETH	SP_OutputAdditionalTrgTimeDiff_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger time difference is transmitted.	
Parameter: int32_t SP_OutputAdditionalMeasrate_ETH	SP_OutputAdditionalMeasrate_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measrate is transmitted.	

11.3.3.2.16 Get_OutputAdditional_ETH (OUTADD_ETH)

Get the additional data to be output at ethernet interface.

Parameter: int32_t SA_OutputAdditionalShutterTime_ETH	SA_OutputAdditionalShutterTime_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1_ETH	SA_OutputAdditionalEncoder1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder2_ETH	SA_OutputAdditionalEncoder2_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 2 is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder3_ETH	SA_OutputAdditionalEncoder3_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 3 is transmitted.	

Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditionalCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalIntensity_ETH	SA_OutputAdditionalIntensity_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if intensity is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_ETH	SA_OutputAdditionalState_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalTrgTimeDiff_ETH	SA_OutputAdditionalTrgTimeDiff_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if trigger time difference is transmitted.	
Parameter: int32_t SA_OutputAdditionalMeasrate_ETH	SA_OutputAdditionalMeasrate_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measrate is transmitted.	

11.3.3.2.17 Set_UnlinearizedMode (SWITCHMD2)

Specifiy if data should be output without linearisation.

Parameter: int32_t SP_UnlinearizedMode	SP_UnlinearizedMode
Direction: Down	
Valid values:	
0= off	
1= on	
Description: Unlinearized data mode.	

11.3.3.2.18 Get_UnlinearizedMode (SWITCHMD2)

Retrieve if data is output without linearisation.

Parameter: int32_t SA_UnlinearizedMode

SA_UnlinearizedMode

Direction: Up

Valid values:

0= off

1= on

Description: Unlinearized data mode.

11.3.3.2.19 Set_OutputVideo_ETH (OUTVIDEO)

Set the video signal to be output at ethernet interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputVideoRaw_ETH

SP_OutputVideoRaw_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if raw video signal is transmitted.

Parameter: int32_t SP_OutputVideoDark_ETH

SP_OutputVideoDark_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if dark preprocessed video signal is transmitted.

Parameter: int32_t SP_OutputVideoLight_ETH

SP_OutputVideoLight_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if light preprocessed video signal is transmitted.

Parameter: int32_t SP_OutputVideoDarkTable_ETH

SP_OutputVideoDarkTable_-
ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if dark table is transmitted.

Parameter: int32_t SP_OutputVideoLightTable_ETH

SP_OutputVideoLight-
Table_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if light table is transmitted.

Parameter: int32_t SP_OutputVideoThreshold_ETH
Direction: Down
Valid values:
 0 = no
 1 = yes
Description: Specify if video threshold table is transmitted.

11.3.3.2.20 Get_OutputVideo_ETH (OUTVIDEO)

Get the video signal to be output at ethernet interface.

Parameter: int32_t SA_OutputVideoRaw_ETH
Direction: Up
Valid values:
 0 = no
 1 = yes
Description: Specify if raw video signal is transmitted.

Parameter: int32_t SA_OutputVideoDark_ETH
Direction: Up
Valid values:
 0 = no
 1 = yes
Description: Specify if dark preprocessed video signal is transmitted.

Parameter: int32_t SA_OutputVideoLight_ETH
Direction: Up
Valid values:
 0 = no
 1 = yes
Description: Specify if light preprocessed video signal is transmitted.

Parameter: int32_t SA_OutputVideoDarkTable_ETH
Direction: Up
Valid values:
 0 = no
 1 = yes
Description: Specify if dark table is transmitted.

Parameter: int32_t SA_OutputVideoLightTable_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if light table is transmitted

Parameter: int32_t SA_OutputVideoThreshold_ETH
Direction: Up
Valid values:
 0 = no
 1 = yes
Description: Specify if video threshold table is transmitted

11.3.3.3 Switching outputs

11.3.3.3.1 Set_ErrorOutput<n> (ERROROUT<n>)

Set condition to be used to set error output $n \in \{1..2\}$.

Parameter: int32_t SP_ErrorOutput<n>

SP_ErrorOutput<n>

Direction: Down

Valid values:

- 0= None
- 1= Intensity error (ER1)
- 2= Out of range (ER2)
- 3= Intensity error or out of range (ER12)
- 4= Below low limit (LI1)
- 5= Above high limit (LI2)
- 6= Out of limits (LI12)

Description: Condition for error output.

11.3.3.3.2 Get_ErrorOutput<n> (ERROROUT<n>)

Get condition to be used to set error output $n \in \{1..2\}$.

Parameter: int32_t SA_ErrorOutput<n>

SA_ErrorOutput<n>

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Intensity error (ER1)
- 2= Out of range (ER2)
- 3= Intensity error or out of range (ER12)
- 4= Below low limit (LI1)
- 5= Above high limit (LI2)
- 6= Out of limits (LI12)

Description: Condition for error output.

11.3.3.3.3 Set_ErrorLimit (ERRORLIMIT)

Set the error limits.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_DataSource

SP_DataSource

Direction: Down

Valid values:

- 0= Distance 1
- 1= Distance 2
- 3= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)
- 4= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)
- 5= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)
- 6= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)
- 2= Thickness 12
- 7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)

- 8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)
- 9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)
- 10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)
- 11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)
- 12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)
- 13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)
- 14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)
- 15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)
- 16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)
- 17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)
- 18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)
- 19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)
- 20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)

Description: Data source to be checked.

Parameter: double SP_LowerLimit SP_LowerLimit

Direction: Down

Valid values:

- Minimum:** -120.0
- Maximum:** 120.0

Unit: mm

Description: Lower limit.

Parameter: double SP_UpperLimit SP_UpperLimit

Direction: Down

Valid values:

- Minimum:** -120.0
- Maximum:** 120.0

Unit: mm

Description: Upper limit.

11.3.3.4 Get_ErrorLimit (ERRORLIMIT)

Get the error limits.

Parameter: int32_t SA_DataSource SA_DataSource

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Distance 1
- 1= Distance 2
- 3= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)
- 4= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)
- 5= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)
- 6= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)
- 2= Thickness 12
- 7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)
- 8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)
- 9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)
- 10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)
- 11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)
- 12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)
- 13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)

- 14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)
- 15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)
- 16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)
- 17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)
- 18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)
- 19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)
- 20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)

Description: Data source to be checked.

Parameter: double SA_LowerLimit SA_LowerLimit
Direction: Up
Valid values:
 Minimum: -120.0
 Maximum: 120.0
Unit: mm
Description: Lower limit.

Parameter: double SA_UpperLimit SA_UpperLimit
Direction: Up
Valid values:
 Minimum: -120.0
 Maximum: 120.0
Unit: mm
Description: Upper limit.

11.3.3.5 Set_ErrorLevel (ERRORLEVEL)

Set level of error output on error.

Parameter: int32_t SP_ErrorLevel SP_ErrorLevel
Direction: Down
Valid values:
 0= High
 1= Low
Description: Error level.

11.3.3.6 Get_ErrorLevel (ERRORLEVEL)

Get level of error output on error.

Parameter: int32_t SA_ErrorLevel SA_ErrorLevel
Direction: Up
Valid values:
 0= High
 1= Low
Description: Error level.

11.3.3.4 Analog output

11.3.3.4.1 Set_AnalogOutput (ANALOGOUT)

Set the data to be used for analog output.

In measure mode MultiLayer and data output interface Ethernet the controller may change output data automatically. Next call to [DataAvail](#) or [TransferData](#) will return a warning. In this case call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_AnalogOutput

SP_AnalogOutput

Direction: Down

Valid values:

- 0= Distance 1
- 1= Distance 2
- 3= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)
- 4= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)
- 5= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)
- 6= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)
- 2= Thickness 12
- 7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)
- 8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)
- 9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)
- 10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)
- 11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)
- 12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)
- 13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)
- 14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)
- 15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)
- 16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)
- 17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)
- 18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)
- 19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)
- 20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)

Description: Data to be used for analog output.

11.3.3.4.2 Get_AnalogOutput (ANALOGOUT)

Get the data to be used for analog output.

Parameter: int32_t SA_AnalogOutput

SA_AnalogOutput

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Distance 1
- 1= Distance 2
- 3= Distance 3 (only at IFD2451 and IFD2461 and IFD2471)
- 4= Distance 4 (only at IFD2451 and IFD2461 and IFD2471)
- 5= Distance 5 (only at IFD2451 and IFD2461 and IFD2471)
- 6= Distance 6 (only at IFD2451 and IFD2461 and IFD2471)
- 2= Thickness 12
- 7= Thickness 13 (only at IFD2451 and IFD2461 and IFD2471)
- 8= Thickness 14 (only at IFD2451 and IFD2461 and IFD2471)

- 9= Thickness 15 (only at IFD2451 and IFD2461 and IFD2471)
- 10= Thickness 16 (only at IFD2451 and IFD2461 and IFD2471)
- 11= Thickness 23 (only at IFD2451 and IFD2461 and IFD2471)
- 12= Thickness 24 (only at IFD2451 and IFD2461 and IFD2471)
- 13= Thickness 25 (only at IFD2451 and IFD2461 and IFD2471)
- 14= Thickness 26 (only at IFD2451 and IFD2461 and IFD2471)
- 15= Thickness 34 (only at IFD2451 and IFD2461 and IFD2471)
- 16= Thickness 35 (only at IFD2451 and IFD2461 and IFD2471)
- 17= Thickness 36 (only at IFD2451 and IFD2461 and IFD2471)
- 18= Thickness 45 (only at IFD2451 and IFD2461 and IFD2471)
- 19= Thickness 46 (only at IFD2451 and IFD2461 and IFD2471)
- 20= Thickness 56 (only at IFD2451 and IFD2461 and IFD2471)

Description: Data to be used for analog output.

11.3.3.4.3 Set_AnalogRange (ANALOG RANGE)

Set the analog output range.

Parameter: int32_t SP_AnalogRange

SP_AnalogRange

Direction: Down

Valid values:

- 0= None
- 1= 0 - 5V
- 2= 0 - 10V
- 3= -5 - 5V
- 4= -10 - 10V
- 5= 4 - 20mA

Description: Analog output range.

11.3.3.4.4 Get_AnalogRange (ANALOG RANGE)

Get the analog output range.

Parameter: int32_t SA_AnalogRange

SA_AnalogRange

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= 0 - 5V
- 2= 0 - 10V
- 3= -5 - 5V
- 4= -10 - 10V
- 5= 4 - 20mA

Description: Analog output range.

11.3.3.4.5 Set_AnalogScale (ANALOG SCALE)

Set the scaling factor for analog output. If both parameters are zero, the analog output is scaled to range of sensor head.

Parameter: int32_t SP_AnalogScaleMode

SP_AnalogScaleMode

Direction: Down

Valid values:

- 0= Standard
- 1= Two point (TWOPOINT)

Description: Analog scale mode.

Parameter: double SP_MinValue SP_MinValue

Direction: Down

Valid values:

Minimum: -120.0

Maximum: 120.0

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SP_MaxValue SP_MaxValue

Direction: Down

Valid values:

Minimum: -120.0

Maximum: 120.0

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

11.3.3.4.6 Get_AnalogScale (ANALOGSCALE)

Get the scaling factor for analog output. If both parameters are zero, the analog output is scaled to range of sensor head.

Parameter: int32_t SA_AnalogScaleMode SA_AnalogScaleMode

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Standard

1= Two point (TWOPOINT)

Description: Analog scale mode.

Parameter: double SA_MinValue SA_MinValue

Direction: Up

Valid values:

Minimum: -120.0

Maximum: 120.0

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SA_MaxValue SA_MaxValue

Direction: Up

Valid values:

Minimum: -120.0

Maximum: 120.0

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

11.3.4 Internal commands

11.3.4.1 Set_PeakParameter (PEAKPARAM)

Set peak detection parameters. This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_HalfMinDist SP_HalfMinDist

Direction: Down

Valid values:

Minimum: 0

Maximum: 255

Description: Half of the minimum distance between two peaks.

Parameter: int32_t SP_MinGap SP_MinGap

Direction: Down

Valid values:

Minimum: 0

Maximum: 255

Description: Minimum gap between two peaks (signal below threshold).

11.3.4.2 Get_PeakParameter (PEAKPARAM)

Get peak detection parameters. This is an internal command. It should not be used by the customer.

Parameter: int32_t SA_HalfMinDist SA_HalfMinDist

Direction: Up

Valid values:

Minimum: 0

Maximum: 255

Description: Half of the minimum distance between two peaks.

Parameter: int32_t SA_MinGap SA_MinGap

Direction: Up

Valid values:

Minimum: 0

Maximum: 255

Description: Minimum gap between two peaks (signal below threshold).

11.3.4.3 Set_BoreControlParam (BORECTRL)

Set firmware parameters for boreCONTROL.

Only available at Option 002 and 201.

This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_SplitPixel SP_SplitPixel

Direction: Down

Valid values:

Minimum: 0

Maximum: 511

Description: DarkCorr evaluates only CCD right of this pixel.

Parameter: int32_t SP_DarkCorrAverage SP_DarkCorrAverage
Direction: Down
Valid values:
Minimum: 0
Maximum: 2048
Description: Video averaging for DarkCorr.

11.3.4.4 Get_BoreControlParam (BORECTRL)

Get firmware parameters for boreCONTROL.
 Only available at Option 002 and 201.
 This is an internal command. It should not be used by the customer.

Parameter: int32_t SA_SplitPixel SA_SplitPixel
Direction: Up
Valid values:
Minimum: 0
Maximum: 511
Description: DarkCorr evaluates only CCD right of this pixel.

Parameter: int32_t SA_DarkCorrAverage SA_DarkCorrAverage
Direction: Up
Valid values:
Minimum: 0
Maximum: 2048
Description: Video averaging for DarkCorr.

11.3.4.5 Get_SpecCal (SPECCAL)

This is an internal command. It should not be used by the customer.

Parameter: double SA_lc0 SA_lc0
Direction: Up
Valid values:
Minimum: -1.79769e+308 (-DBL_MAX)
Maximum: 1.79769e+308 (DBL_MAX)
Description: lc0.

Parameter: double SA_lc1 SA_lc1
Direction: Up
Valid values:
Minimum: -1.79769e+308 (-DBL_MAX)
Maximum: 1.79769e+308 (DBL_MAX)
Description: lc1.

Parameter: double SA_lc2 SA_lc2
Direction: Up
Valid values:
Minimum: -1.79769e+308 (-DBL_MAX)
Maximum: 1.79769e+308 (DBL_MAX)
Description: lc2.

11.3.4.6 Del_LinearisationTable (DELLINTAB)

Delete a linearisation table from sensor. This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_SensorTable SP_SensorTable
Direction: Down
Valid values:
 Minimum: 0
 Maximum: 19
Description: Index of sensor table.

11.3.4.7 Save_ParametersTemp (SAVE2TMP)

Save parameters temporary. This is an internal command. It should not be used by the customer.

11.3.4.8 Load_ParametersTemp (RESTORETMP)

Restore parameters previously saved from temporary buffer. This is an internal command. It should not be used by the customer.

11.3.4.9 Set_CalibrationDefaults (SETCALDEFAULT)

Set default parameter set for calibration. This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_CalDefaultType SP_CalDefaultType
Direction: Down
Valid values:
 0= All settings (ALL)
 1= Only measurement settings (MEAS)
Description: Specifies which settings should be set to calibration defaults.

11.3.4.10 Get_RegHyst (REGHYST)

Get the destination value for regulation (peak height in raw signal). This is an internal command. It should not be used by the customer.

Parameter: int32_t SA_RegHyst SA_RegHyst
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 100
Unit: %
Description: Destination value for regulation.

11.3.4.11 **Upload_CalibTable**

Send a calibration table for selected sensor to the controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SP_CalibTable

SP_CalibTable

Direction: Down

Description: Calibration table for selected sensor.

11.3.4.12 **Save_DarkCorrReference (SAVEDARKCORRREF)**

Save the darkcorr reference value. This is an internal command. It should not be used by the customer.

12 Commands for interferoMETER (IMC) sensors

12.1 Commands for IMC5400/10 and IMC5600

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
- [TCP/IP](#) (native).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_AllParameters](#) (SP_Additional= 1) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate, to interpret and scale data and to assign values. If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls optionally sensor command [Set_IPDataTransferMode](#) and [Set_DataOutInterface](#) at [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_AllParameters](#) (SP_Additional= 1)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

12.1.1 General commands

12.1.1.1 General

12.1.1.1.1 Get_Help (HELP)

Retrieve a help text from controller for a specific command.

Parameter: String SP_Command SP_Command

Direction: Down

Valid values:

"" (empty string, means general help)
or any command name

Description: Name of the command.

Parameter: String SA_HelpText SA_HelpText
Direction: Up
Description: Help text to the command.

12.1.1.1.2 Get_Info (GETINFO)

Retrieve information about the controller.

Parameter: String SA_Sensor SA_Sensor
Direction: Up
Valid values:
 IMC5400
 IMC5410
 IMC5600
Description: Name of the controller.

Parameter: String SA_SerialNumber SA_SerialNumber
Direction: Up
Valid values:
 Numeric value
Description: Serial number of the controller.

Parameter: String SA_Option SA_Option
Direction: Up
Valid values:
 Numeric value
Description: Option of the controller.

Parameter: String SA_ArticleNumber SA_ArticleNumber
Direction: Up
Valid values:
 Numeric value
Description: Article number of the controller.

Parameter: String SA_MacAddress SA_MacAddress
Direction: Up
Valid values:
 Valid MAC address in form of xx-xx-xx-xx-xx-xx
Description: MAC address (low level ethernet address) of the controller.

Parameter: String SA_Softwareversion SA_Softwareversion
Direction: Up
Description: Software version of firmware in the controller.

Parameter: String SA_HardwareRevision SA_HardwareRevision
Direction: Up
Valid values:
 Numeric value
Description: Revision index of the controller board.

Parameter: String SA_BuildID SA_BuildID
Direction: Up
Description: Build ID

Parameter: String SA_BuildTimestamp SA_BuildTimestamp
Direction: Up
Description: Build timestamp

Parameter: String SA_BootVersion SA_BootVersion
Direction: Up
Description: Boot version

12.1.1.3 Get_ChannelCount (GETCHANNELCNT)

Get number of channels of controller.

Parameter: int32_t SA_ChannelCount SA_ChannelCount
Direction: Up
Valid values:
Minimum: 1
Maximum: 1
Description: Number of channels.

12.1.1.4 Get_OutputInfo_RS422 (GETOUTINFO_RS422)

Retrieve information which data is output at RS422 interface.

Parameter: int32_t SA_OutputVideoAbs_Ch1_RS422 SA_OutputVideoAbs_Ch1_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if absolute signal of first channel is transmitted.

Parameter: int32_t SA_OutputPeak1..16_Ch1_RS422 SA_OutputPeak1..16_Ch1_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if peak 1 to 16 of first channel is transmitted.

Parameter: int32_t SA_OutputAmount1..16_Ch1_RS422 SA_OutputAmount1..16_Ch1_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if amount 1 to 16 of first channel is transmitted.

Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_RS422 SA_OutputAdditionalShutterTime_Ch1_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if shutter time of first channel is transmitted.

Parameter: int32_t SA_OutputAdditionalEncoder1..3_Ch1_RS422	SA_OutputAdditionalEncoder1..3_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 to 3 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalMeasrate_RS422	SA_OutputAdditionalMeasrate_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measurement rate is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: String SA_OutputCalculation_RS422	SA_OutputCalculation_RS422
Direction: Up	
Description: List of calculated signals (generated by Set_Computation). The entries are separated by spaces.	
Parameter: String SA_OutputStatistic_RS422	SA_OutputStatistic_RS422
Direction: Up	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries are separated by spaces.	

12.1.1.5 Get_OutputInfo_ETH (GETOUTINFO_ETH)

Retrieve information which data is output at ethernet interface.

Parameter: int32_t SA_OutputVideoAbs_Ch1_ETH	SA_OutputVideoAbs_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if absolute signal of first channel is transmitted.	

Parameter: int32_t SA_OutputPeak1..16_Ch1_ETH	SA_OutputPeak1..16_Ch1_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if peak 1 to 16 of first channel is transmitted.	
Parameter: int32_t SA_OutputAmount1..16_Ch1_ETH	SA_OutputAmount1..16_- Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if amount 1 to 16 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_ETH	SA_OutputAdditionalShut- terTime_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1..3_Ch1_ETH	SA_OutputAdditionalEn- coder1..3_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 to 3 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalMeasrate_ETH	SA_OutputAdditionalMeas- rate_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measurement rate is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimes- tamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditional- Counter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_ETH	SA_OutputAdditional- State_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	

Parameter: String SA_OutputCalculation_ETH	SA_OutputCalculation_ETH
Direction: Up	
Description: List of calculated signals (generated by Set_Computation). The entries are separated by spaces.	
Parameter: String SA_OutputStatistic_ETH	SA_OutputStatistic_ETH
Direction: Up	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries are separated by spaces.	

12.1.1.1.6 Set_Unit (UNIT)

Set the unit for configuration and display in the web diagram.

Parameter: int32_t SP_DisplayUnit	SP_DisplayUnit
Direction: Down	
Valid values:	
0= mm	
1= Inch	

Description: Unit.

12.1.1.1.7 Get_Unit (UNIT)

Get the unit for configuration and display in the web diagram.

Parameter: int32_t SA_DisplayUnit	SA_DisplayUnit
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= mm	
1= Inch	

Description: Unit.

12.1.1.1.8 Set_Echo (ECHO)

Set echo for sensor commands.

Parameter: int32_t SP_Echo	SP_Echo
Direction: Down	
Valid values:	
0= Off	
1= On	

Description: Echo mode.

12.1.1.1.9 Get_Echo (ECHO)

Get the echo mode.

Parameter: int32_t SA_Echo	SA_Echo
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	

Description: Echo mode.

12.1.1.1.10 Get_AllParameters (PRINT)

Get all parameters from controller.

Parameter: int32_t SP_Additional	SP_Additional
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description:	If set, additional information about controller, sensor and material is output.
Parameter: int32_t SA_UserLevel	SA_UserLevel
Direction: Up	
Valid values:	
-1= Other user level (only for internal use)	
0= User	
1= Professional	
Description:	Actual user level.
Parameter: int32_t SA_DefaultUser	SA_DefaultUser
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= User	
1= Professional	
Description:	Default user level.
Parameter: int32_t SA_DisplayUnit	SA_DisplayUnit
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= mm	
1= Inch	
Description:	Unit.
Parameter: int32_t SA_ApplicationLanguage	SA_ApplicationLanguage
Direction: Up	
Valid values:	
0= English (EN)	
1= German (DE)	
Description:	Language of web interface.
Parameter: int32_t SA_ChartType	SA_ChartType
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Measure (MEAS)	
1= Video signal (VIDEO)	
Description:	Type of the chart at web interface.

Parameter: int32_t SA_Echo	SA_Echo
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Off	
1 = On	
Description: Echo mode.	
Parameter: double SA_Measrate	SA_Measrate
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 6.0	
Unit: kHz	
Description: Samplerate of measurement.	
Parameter: int32_t SA_EthernetMode	SA_EthernetMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Ethernet	
1 = Ethercat	
Description: Ethernet mode.	
Parameter: int32_t SA_DHCPEnabled	SA_DHCPEnabled
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = FALSE	
1 = TRUE	
Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).	
Parameter: String SA_Address	SA_Address
Direction: Up	
Valid values:	
Valid IP address in form of xxx.xxx.xxx.xxx	
Description: IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.	
Parameter: String SA_SubnetMask	SA_SubnetMask
Direction: Up	
Valid values:	
Valid network mask (e.g. 255.255.255.0 for a Class C network)	
Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.	
Parameter: String SA_Gateway	SA_Gateway
Direction: Up	
Valid values:	
Valid IP address of default gateway in form of xxx.xxx.xxx.xxx	
Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.	

Parameter: int32_t SA_Protocol SA_Protocol

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = TCP server (SERVER/TCP)
- 1 = TCP client (CLIENT/TCP)
- 2 = UDP sender (CLIENT/UDP)
- 3 = None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SA_RemoteAddress SA_RemoteAddress

Direction: Up

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to.

Parameter: int32_t SA_Port SA_Port

Direction: Up

Valid values:

- Minimum:** 1024
- Maximum:** 65535

Description: Port to send data to or to listen for incoming requests.

Parameter: int32_t SA_TCPKeepAlive SA_TCPKeepAlive

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Off
- 1 = On

Description: TCP keep alive feature.

Parameter: int32_t SA_FramesPerPacket_ETH SA_FramesPerPacket_ETH

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 350

Description: Maximum number of frames in ethernet packet. 0 means automatic.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate

Direction: Up

Valid values:

- 4000000
- 3000000
- 2000000
- 921600
- 691200
- 460800
- 230400
- 115200
- 9600

Unit: Baud

Description: Baudrate of controller.

Parameter: int32_t SA_Keylock SA_Keylock

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Inactive (NONE)
- 1= Active
- 2= Automatic (AUTO)

Description: Keylock.

Parameter: int32_t SA_KeylockTime SA_KeylockTime

Direction: Up

Valid values:

Minimum: 0
Maximum: 60

Unit: Minutes

Description: Keylock time (only available at automatic keylock).

Parameter: int32_t SA_KeylockState SA_KeylockState

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Inactive (IS_INACTIVE)
- 1= Active (IS_ACTIVE)

Description: Actual keylock state (only available at automatic keylock).

Parameter: int32_t SA_KeyFunction1 SA_KeyFunction1

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Key has no function (NONE)
- 1= The button triggers the command [Exec_Master](#) with action set for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERSET)
- 2= The button triggers the command [Exec_Master](#) with action reset for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERRESET)
- 3= Use key to toggle the pilot laser (PILOTLASER)
- 4= The button is used as a toggle for the SLED (see [Set_SuperluminescentLED](#) (SLED))

Description: Button function

Parameter: int32_t SA_KeyFunction2 SA_KeyFunction2

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Key has no function (NONE)
- 1= The button triggers the command \fixhh{SENSOR_IMC5x00}{Exec_Master} with action set for the signal(s) defined by \fixhh{SENSOR_IMC5x00}{Set_KeyMasterSignalSelect} (MASTERSET)
- 2= The button triggers the command \fixhh{SENSOR_IMC5x00}{Exec_Master} with action reset for the signal(s) defined by \fixhh{SENSOR_IMC5x00}{Set_KeyMasterSignalSelect} (MASTERRESET)
- 3= Use key to toggle the pilot laser (PILOTLASER)
- 4= The button is used as a toggle for the SLED (see \fixhh{SENSOR_IMC5x00}{Set_SuperluminescentLED}) (SLED)

Description: Button function

Parameter: int32_t SA_MasterPeak1..16_Ch1	SA_MasterPeak1..16_Ch1
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if peak 1 to 16 of first channel is mastered.	
Parameter: String SA_MasterSignalNames	SA_MasterSignalNames
Direction: Up	
Description: List of user defined signals (generated by Set_Computation or Set_StatisticSignal). The entries are separated by spaces.	
The special value ALL specify to master all possible signals. NONE means no signal.	
Parameter: int32_t SA_OutputRS422	SA_OutputRS422
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at RS422 interface.	
Parameter: int32_t SA_OutputAnalog	SA_OutputAnalog
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at analog interface.	
Parameter: int32_t SA_OutputEthernet	SA_OutputEthernet
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at ethernet interface.	
Parameter: int32_t SA_OutputErrorOut	SA_OutputErrorOut
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at error outputs.	
Parameter: int32_t SA_ResampleRS422	SA_ResampleRS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: RS422 output is resampled.	
Parameter: int32_t SA_ResampleAnalog	SA_ResampleAnalog
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Analog output is resampled.	

Parameter: int32_t SA_ResampleEthernet	SA_ResampleEthernet
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Ethernet output is resampled.	
Parameter: int32_t SA_Resampling	SA_Resampling
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 3000000	
Description: Resampling value.	
Parameter: int32_t SA_TriggerSource	SA_TriggerSource
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Sync/Trig	
2= TrigIn	
3= Software	
4= Encoder1	
5= Encoder2	
6= Encoder3	
Description: Trigger source (input).	
Parameter: int32_t SA_TriggerInLogic	SA_TriggerInLogic
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= TTL	
1= HTL	
Description: Logic of trigger input	
Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Edge	
1= Level (PULSE)	
Description: Trigger mode.	
Parameter: int32_t SA_TriggerLevel	SA_TriggerLevel
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= High	
1= Low	
Description: Trigger level.	

Parameter: int32_t SA_TriggerCount	SA_TriggerCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 16383	
Description: Number of values to measure. 0 means no trigger (NONE), 16383 means endless measurement (INFINITE).	
Parameter: int32_t SA_TriggerMoment	SA_TriggerMoment
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Input	
1= Output	
Description: Trigger moment.	
Parameter: double SA_TriggerEncoderMin	SA_TriggerEncoderMin
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Lower encoder limit. Above this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: double SA_TriggerEncoderMax	SA_TriggerEncoderMax
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Upper encoder limit. Below this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: double SA_TriggerEncoderStepsize	SA_TriggerEncoderStepsize
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 2147483647.0 (INT32_MAX)	
Description: Number of encoder steps before trigger. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: int32_t SA_SyncMode	SA_SyncMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Master	
2= Slave (Sync/Trig)	
3= Slave (TrigIn)	
Description: Synchronization mode.	

Parameter: int32_t SA_Termination	SA_Termination
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Description: Termination resistor	
Parameter: int32_t SA_ErrorLimitSignalIndex1..2	SA_ErrorLimitSignalIndex1..2
Direction: Up	
Valid values:	
-1= User defined name at SA_ErrorLimitSignalName1..2	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	
7= Peak 8 of first Channel	
8= Peak 9 of first Channel	
9= Peak 10 of first Channel	
10= Peak 11 of first Channel	
11= Peak 12 of first Channel	
12= Peak 13 of first Channel	
13= Peak 14 of first Channel	
14= Peak 15 of first Channel	
15= Peak 16 of first Channel	
Description: Signal index to be checked.	
Parameter: String SA_ErrorLimitSignalName1..2	SA_ErrorLimitSignalName1..2
Direction: Up	
Description: User defined signal which is used for error limit.	
Parameter: double SA_LowerLimit1..2	SA_LowerLimit1..2
Direction: Up	
Valid values:	
Minimum: -21.47	
Maximum: 21.47	
Unit: mm	
Description: Lower limit.	
Parameter: double SA_UpperLimit1..2	SA_UpperLimit1..2
Direction: Up	
Valid values:	
Minimum: -21.47	
Maximum: 21.47	
Unit: mm	
Description: Upper limit.	

Parameter: int32_t SA_ErrorLimitCompOp1..2	SA_ErrorLimitCompOp1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Lower	
1= Upper	
2= Both	
Description: Compare operation	
Parameter: int32_t SA_ErrorLevelOut1..2	SA_ErrorLevelOut1..2
Direction: Up	
Valid values:	
0= NPN	
1= PNP	
2= Push-Pull (PUSHPULL)	
3= Push-Pull negated (PUSHPULLNEG)	
Description: Error level.	
Parameter: int32_t SA_EncoderInterpolation1..3	SA_EncoderInterpolation1..3
Direction: Up	
Valid values:	
1	
2	
4	
Description: Encoder interpolation.	
Parameter: int32_t SA_Encoder3Enable	SA_Encoder3Enable
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Description: Third encoder is enabled or disabled.	
If it is enabled, then the Set_EncoderMode commands will be set to no action.	
Parameter: int32_t SA_EncoderMode1..2	SA_EncoderMode1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= No action (NONE)	
1= Set encoder value to preset value only one time (ONE)	
2= Set encoder value to preset value each time (EVER)	
Description: Mode of encoder when reference is reached.	
Parameter: double SA_EncoderPreload1..3	SA_EncoderPreload1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, its type is double.	

Parameter: double SA_Encoder.MaxValue1..3	SA_Encoder.MaxValue1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Maximum value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: int32_t SA_ROIStart	SA_ROIStart
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: First position on CCD.	
Parameter: int32_t SA_ROIEnd	SA_ROIEnd
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: Last position on CCD.	
Parameter: String SA_MaterialMultiPeak12	SA_MaterialMultiPeak12
Direction: Up	
Description: Name of material between first and second peak.	
Parameter: String SA_MaterialMultiPeak23	SA_MaterialMultiPeak23
Direction: Up	
Description: Name of material between second and third peak.	
Parameter: String SA_MaterialMultiPeak34	SA_MaterialMultiPeak34
Direction: Up	
Description: Name of material between third and fourth peak.	
Parameter: String SA_MaterialMultiPeak45	SA_MaterialMultiPeak45
Direction: Up	
Description: Name of material between fourth and fifth peak.	
Parameter: String SA_MaterialMultiPeak56	SA_MaterialMultiPeak56
Direction: Up	
Description: Name of material between fifth and sixth peak.	
Parameter: double SA_MinimumThreshold	SA_MinimumThreshold
Direction: Up	
Valid values:	
Minimum: 0.05 for IMC5410, otherwise 0.5	
Maximum: 100.0	
Unit: %	
Description: A valid peak must be above this threshold.	

Parameter: int32_t SA_MeasurePeakSort	SA_MeasurePeakSort
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Height	
1= Distance	
Description: Measure peak sort mode.	
Parameter: int32_t SA_ActiveSensor	SA_ActiveSensor
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of active sensor head.	
Parameter: int32_t SA_AnalogOutputSignalIndex	SA_AnalogOutputSignalIndex
Direction: Up	
Valid values:	
-1= User defined name at SA_AnalogOutputSignalName	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	
7= Peak 8 of first Channel	
8= Peak 9 of first Channel	
9= Peak 10 of first Channel	
10= Peak 11 of first Channel	
11= Peak 12 of first Channel	
12= Peak 13 of first Channel	
13= Peak 14 of first Channel	
14= Peak 15 of first Channel	
15= Peak 16 of first Channel	
Description: Data to be used for analog output.	
Parameter: String SA_AnalogOutputSignalName	SA_AnalogOutputSignalName
Direction: Up	
Description: User defined signal which is used for analog output.	
Parameter: int32_t SA_AnalogRange	SA_AnalogRange
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
1= 0 - 5V	
2= 0 - 10V	
5= 4 - 20mA	
Description: Analog output range.	
Parameter: int32_t SA_AnalogScaleMode	SA_AnalogScaleMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Standard	
1= Two point (TWOPOINT)	
Description: Analog scale mode.	

Parameter: double SA_AnalogScaleRangeLowerLimit	SA_AnalogScaleRangeLowerLimit
Direction: Up	
Valid values:	
Minimum: -21.47	
Maximum: 21.47	
Unit: mm	
Description: Value which represents lowest voltage/current (at two point scaling).	
Parameter: double SA_AnalogScaleRangeUpperLimit	SA_AnalogScaleRangeUpperLimit
Direction: Up	
Valid values:	
Minimum: -21.47	
Maximum: 21.47	
Unit: mm	
Description: Value which represents highest voltage/current (at two point scaling).	
Parameter: int32_t SA_OutputVideoAbs_Ch1_RS422	SA_OutputVideoAbs_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if absolute signal of first channel is transmitted.	
Parameter: int32_t SA_OutputPeak1..16_Ch1_RS422	SA_OutputPeak1..16_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if peak 1 to 16 of first channel is transmitted.	
Parameter: int32_t SA_OutputAmount1..16_Ch1_RS422	SA_OutputAmount1..16_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if amount 1 to 16 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_RS422	SA_OutputAdditionalShutterTime_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1..3_Ch1_RS422	SA_OutputAdditionalEncoder1..3_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 to 3 of first channel is transmitted.	

Parameter: int32_t SA_OutputAdditionalAmount_Ch1_RS422	SA_OutputAdditionalAmount_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if amount of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalMeasrate_RS422	SA_OutputAdditionalMeasrate_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measurement rate is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: String SA_OutputCalculation_RS422	SA_OutputCalculation_RS422
Direction: Up	
Description: List of calculated signals (generated by Set_Computation). The entries are separated by spaces.	
Parameter: String SA_OutputStatistic_RS422	SA_OutputStatistic_RS422
Direction: Up	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries are separated by spaces.	
Parameter: int32_t SA_OutputVideoAbs_Ch1_ETH	SA_OutputVideoAbs_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if absolute signal of first channel is transmitted.	

Parameter: int32_t SA_OutputPeak1..16_Ch1_ETH	SA_OutputPeak1..16_Ch1_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if peak 1 to 16 of first channel is transmitted.	
Parameter: int32_t SA_OutputAmount1..16_Ch1_ETH	SA_OutputAmount1..16_- Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if amount 1 to 16 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_ETH	SA_OutputAdditionalShut- terTime_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1..3_Ch1_ETH	SA_OutputAdditionalEn- coder1..3_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 to 3 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalAmount_Ch1_ETH	SA_OutputAdditionalAm- ount_Ch1_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if amount of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalMeasrate_ETH	SA_OutputAdditionalMeas- rate_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measurement rate is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimes- tamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditional- Counter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	

Parameter: int32_t SA_OutputAdditionalState_ETH	SA_OutputAdditionalState_ETH
Direction: Up	
Valid values:	
0 = no	
1 = yes	
Description: Specify if state is transmitted.	
Parameter: String SA_OutputCalculation_ETH	SA_OutputCalculation_ETH
Direction: Up	
Description: List of calculated signals (generated by Set_Computation). The entries are separated by spaces.	
Parameter: String SA_OutputStatistic_ETH	SA_OutputStatistic_ETH
Direction: Up	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries are separated by spaces.	
Parameter: int32_t SA_HoldLastValid	SA_HoldLastValid
Direction: Up	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	
Parameter: String SA_ActiveMaterial	SA_ActiveMaterial
Direction: Up	
Valid for sensor:	
IMC5400	
Description: Name of material.	
Parameter: int32_t SA_NumberOfPeaks	SA_NumberOfPeaks
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 16	
Description: Maximum number of peaks.	
Parameter: int32_t SA_ChannelCount	SA_ChannelCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 1	
Description: Number of channels.	
Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Valid values:	
IMC5400	
IMC5410	
IMC5600	
Description: Name of the controller.	

Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the controller.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the controller.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the controller.	
Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description: MAC address (low level ethernet address) of the controller.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the controller.	
Parameter: String SA_HardwareRevision	SA_HardwareRevision
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of the controller board.	
Parameter: String SA_BuildID	SA_BuildID
Direction: Up	
Description: Build ID	
Parameter: String SA_BuildTimestamp	SA_BuildTimestamp
Direction: Up	
Description: Build timestamp	
Parameter: String SA_BootVersion	SA_BootVersion
Direction: Up	
Description: Boot version	
Parameter: int32_t SA_Pos	SA_Pos
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the active sensor head in the table.	
Parameter: String SA_Name	SA_Name
Direction: Up	
Description: Name of the active sensor head.	

Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Description: Measurement range of the active sensor head.	
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description: Unit of the measurement range.	
Parameter: String SA_Serial	SA_Serial
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the active sensor head.	
Parameter: String SA_SensorType	SA_SensorType
Direction: Up	
Description: Type of the active sensor head, e.g. distance or thickness.	
Parameter: int32_t SA_PilotLaser	SA_PilotLaser
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Off	
1 = On	
Description: Pilot laser is enabled or disabled.	
Parameter: int32_t SA_SuperluminescentLED	SA_SuperluminescentLED
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Off	
1 = On	
Description: Superluminescent LED is enabled or disabled.	
Parameter: String SA_ComputationTable	SA_ComputationTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_ComputationEntries	SA_ComputationEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 30	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_ComputationEntry1, SA_ComputationEntry2, ...	
Parameter: String SA_ComputationEntry _{1..x}	SA_ComputationEntry _{1..x}
Direction: Up	
Description: Each computation as complete line.	

Parameter: int32_t SA_ComputationQueue1..x	SA_ComputationQueue1..x
Direction: Up	
Valid values:	
0= Channel 1	
Description: Computation queue	
Parameter: int32_t SA_ComputationID1..x	SA_ComputationID1..x
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Index within computation queue	
Parameter: int32_t SA_ComputationOperation1..x	SA_ComputationOperation1..x
Direction: Up	
Valid values:	
0= None	
1= Median	
2= Moving average	
3= Recursive averaging	
4= Calculation (only available at Option 201)	
5= Thickness (only available at Option 201)	
6= Copy	
Description: Computation operation	
Parameter: int32_t SA_ComputationSignalIndex1..x	SA_ComputationSignalIndex1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_ComputationSignalName1..x	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	
7= Peak 8 of first Channel	
8= Peak 9 of first Channel	
9= Peak 10 of first Channel	
10= Peak 11 of first Channel	
11= Peak 12 of first Channel	
12= Peak 13 of first Channel	
13= Peak 14 of first Channel	
14= Peak 15 of first Channel	
15= Peak 16 of first Channel	
Description: Signal index (if operation is averaging or copy).	
Parameter: String SA_ComputationSignalName1..x	SA_ComputationSignalName1..x
Direction: Up	
Description: User defined signal (if operation is averaging or copy).	

Parameter: int32_t SA_ComputationMovingCount1..x	SA_ComputationMoving- Count1..x
Direction: Up	
Valid values:	
2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096	
Description: Number of value for the averaging window. This parameter is only used at moving average.	
Parameter: int32_t SA_ComputationRecursiveCount1..x	SA_ComputationRecursive- Count1..x
Direction: Up	
Valid values:	
Minimum: 2	
Maximum: 32768	
Description: Number of values for recursive averaging. This parameter is only used at recursive average.	
Parameter: int32_t SA_ComputationMedianCount1..x	SA_ComputationMedian- Count1..x
Direction: Up	
Valid values:	
3, 5, 7, 9	
Description: Number of values to build median. This parameter is only used at median.	
Parameter: String SA_ComputationName1..x	SA_ComputationName1..x
Direction: Up	
Description: Name of the newly created signal (if operation is calculation, thickness or copy).	
Parameter: String SA_MasterSignalTable	SA_MasterSignalTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_MasterSignalEntries	SA_MasterSignalEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MasterSignalEntry1, SA_MasterSignalEntry2, ...	
Parameter: String SA_MasterSignalEntry1..x	SA_MasterSignalEntry1..x
Direction: Up	
Description: Each master signal as complete line.	
Parameter: int32_t SA_MasterSignalIndex1..x	SA_MasterSignalIndex1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_MasterSignalName1..x	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	

7= Peak 8 of first Channel
 8= Peak 9 of first Channel
 9= Peak 10 of first Channel
 10= Peak 11 of first Channel
 11= Peak 12 of first Channel
 12= Peak 13 of first Channel
 13= Peak 14 of first Channel
 14= Peak 15 of first Channel
 15= Peak 16 of first Channel

Description: Index of each master signal.

Parameter: String SA_MasterSignalName_{1..x}

SA_MasterSignalName_{1..x}

Direction: Up

Description: Name of each master signal.

Parameter: double SA_MasterValue_{1..x}

SA_MasterValue_{1..x}

Direction: Up

Valid values:

Minimum: -21.47

Maximum: 21.47

Unit: mm

Description: Master value

Parameter: String SA_StatisticSignalTable

SA_StatisticSignalTable

Direction: Up

Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_StatisticSignalEntries

SA_StatisticSignalEntries

Direction: Up

Valid values:

Minimum: 0

Maximum: 10

Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_StatisticSignalEntry1, SA_StatisticSignalEntry2, ...

Parameter: String SA_StatisticSignalEntry_{1..x}

SA_StatisticSignalEntry_{1..x}

Direction: Up

Description: Each statistic signal as complete line.

Parameter: int32_t SA_StatisticSignalIndex_{1..x}

SA_StatisticSignalIndex_{1..x}

Direction: Up

Valid values:

-1= User defined name at SA_StatisticSignalName_{1..x}

0= Peak 1 of first Channel

1= Peak 2 of first Channel

2= Peak 3 of first Channel

3= Peak 4 of first Channel

4= Peak 5 of first Channel

5= Peak 6 of first Channel

6= Peak 7 of first Channel

7= Peak 8 of first Channel

8= Peak 9 of first Channel

9= Peak 10 of first Channel

10= Peak 11 of first Channel
 11= Peak 12 of first Channel
 12= Peak 13 of first Channel
 13= Peak 14 of first Channel
 14= Peak 15 of first Channel
 15= Peak 16 of first Channel

Description: Index of each statistic signal.

Parameter: String SA_StatisticSignalName_{1..x}

SA_StatisticSignal-
Name_{1..x}

Direction: Up

Description: Name of each statistic signal.

Parameter: int32_t SA_StatisticDepth_{1..x}

SA_StatisticDepth_{1..x}

Direction: Up

Valid values:

2, 4, 8, ..., 8192, 16384
 16385= Infinite (endless statistic calculation)

Description: Number of values (depth) for statistic calculation.

Parameter: String SA_SensorTable

SA_SensorTable

Direction: Up

Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_SensorTableCount

SA_SensorTableCount

Direction: Up

Valid values:

Minimum: 0
Maximum: 20

Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Channel1, SA_Channel2, ...

Parameter: int32_t SA_Channel_{1..x}

SA_Channel_{1..x}

Direction: Up

Valid values:

1

Description: Sensor channel. Always 1

Parameter: int32_t SA_Pos_{1..x}

SA_Pos_{1..x}

Direction: Up

Valid values:

Minimum: 0
Maximum: 19

Description: Index of the sensor head in the table.

Parameter: String SA_Name_{1..x}

SA_Name_{1..x}

Direction: Up

Description: Name of the sensor head in the table.

Parameter: double SA_Range_{1..x}

SA_Range_{1..x}

Direction: Up

Valid values:

Minimum: 0.0

Description: Measurement range of the sensor head in the table.

Parameter: String SA_Unit1..x	SA_Unit1..x
Direction: Up	
Description: Unit of the measurement range.	
Parameter: String SA_Serial1..x	SA_Serial1..x
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor head in the table.	
Parameter: String SA_SensorType1..x	SA_SensorType1..x
Direction: Up	
Description: Type of the sensor head in the table, e.g. distance or thickness.	
Parameter: String SA_CurrentName	SA_CurrentName
Direction: Up	
Description: Name of current (active) setting.	
Parameter: String SA_SettingNames	SA_SettingNames
Direction: Up	
Description: List (separated by new line).	
Parameter: String SA_PresetNames	SA_PresetNames
Direction: Up	
Description: List (separated by new line).	
Parameter: int32_t SA_Automatic	SA_Automatic
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Automatic selection.	
Parameter: String SA_InitialName	SA_InitialName
Direction: Up	
Description: Name of setting.	
Parameter: int32_t SA_PresetMode	SA_PresetMode
Direction: Up	
Valid values:	
0= Static	
1= Balanced	
2= Dynamic	
3= None (if no preset setting is active)	
Description: Preset mode.	
Parameter: String SA_MaterialTable	SA_MaterialTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_MaterialTableCount	SA_MaterialTableCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 20	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MaterialName1, SA_MaterialName2, ...	

Parameter: String SA_MaterialName <code>1..x</code>	SA_MaterialName <code>1..x</code>
Direction: Up	
Description: Name of the material in the table.	
Parameter: double SA_GroupIndex <code>1..x</code>	SA_GroupIndex <code>1..x</code>
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Group index of the material.	
Parameter: String SA_Description <code>1..x</code>	SA_Description <code>1..x</code>
Direction: Up	
Description: Description of the material in the table.	
Parameter: String SA_MaterialName	SA_MaterialName
Direction: Up	
Description: Name of the active material.	
Parameter: String SA_Description	SA_Description
Direction: Up	
Description: Description of the active material.	
Parameter: double SA_GroupIndex	SA_GroupIndex
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 4.0	
Description: Group index.	

12.1.1.11 Set_SyncMode (SYNC)

Set the synchronization mode.

Parameter: int32_t SP_SyncMode	SP_SyncMode
Direction: Down	
Valid values:	
0= None	
1= Master	
2= Slave (Sync/Trig)	
3= Slave (TrigIn)	
Description: Synchronization mode.	

12.1.1.12 Get_SyncMode (SYNC)

Get the synchronization mode.

Parameter: int32_t SA_SyncMode	SA_SyncMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Master	
2= Slave (Sync/Trig)	
3= Slave (TrigIn)	
Description: Synchronization mode.	

12.1.1.1.13 Set_Termination (TERMINATION)

Set connection of a termination resistor in sync line to prevent reflections.

Parameter: int32_t SP_Termination SP_Termination

Direction: Down

Valid values:

- 0= Off
- 1= On

Description: Termination resistor

12.1.1.1.14 Get_Termination (TERMINATION)

Get connection of a termination resistor in sync line to prevent reflections.

Parameter: int32_t SA_Termination SA_Termination

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Off
- 1= On

Description: Termination resistor

12.1.1.1.15 Reset_Boot (RESET)

Resets the sensor.

At this command the controller may change output data after reboot. If first bit of **IP_AutomaticMode** is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

12.1.1.1.16 Reset_Counter (RESETCNT)

Resets sensor counter values.

Parameter: int32_t SP_ResetTimestamp SP_ResetTimestamp

Direction: Down

Valid values:

- 0= No
- 1= Yes

Description: Reset timestamp value.

Parameter: int32_t SP_ResetMeasCounter SP_ResetMeasCounter

Direction: Down

Valid values:

- 0= No
- 1= Yes

Description: Reset counter value.

12.1.1.1.17 Set_Keylock (KEYLOCK)

Set key lock for sensor.

Parameter: int32_t SP_Keylock SP_Keylock

Direction: Down

Valid values:

0= Inactive (NONE)

1= Active

2= Automatic (AUTO)

Description: Keylock.

Parameter: int32_t SP_KeylockTime SP_KeylockTime

Direction: Down

Valid values:

Minimum: 1

Maximum: 60

Unit: Minutes

Description: Keylock time (only used at automatic keylock).

12.1.1.1.18 Get_Keylock (KEYLOCK)

Get key lock for sensor.

Parameter: int32_t SA_Keylock SA_Keylock

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Inactive (NONE)

1= Active

2= Automatic (AUTO)

Description: Keylock.

Parameter: int32_t SA_KeylockTime SA_KeylockTime

Direction: Up

Valid values:

Minimum: 1

Maximum: 60

Unit: Minutes

Description: Keylock time (only available at automatic keylock).

Parameter: int32_t SA_KeylockState SA_KeylockState

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Inactive (IS_INACTIVE)

1= Active (IS_ACTIVE)

Description: Actual keylock state (only available at automatic keylock).

12.1.1.1.19 Set_KeyFunction1 (KEYFUNC1)

Set the button functionality for timeslot 1 (0...2s)

Parameter: int32_t SP_KeyFunction1

SP_KeyFunction1

Direction: Down

Valid values:

- 0= Key has no function (NONE)
- 1= The button triggers the command [Exec_Master](#) with action set for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERSET)
- 2= The button triggers the command [Exec_Master](#) with action reset for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERRESET)
- 3= Use key to toggle the pilot laser (PILOTLASER)
- 4= The button is used as a toggle for the SLED (see [Set_SuperluminescentLED](#)) (SLED)

Description: Button function

12.1.1.1.20 Get_KeyFunction1 (KEYFUNC1)

Get the button functionality for timeslot 1 (0...2s)

Parameter: int32_t SA_KeyFunction1

SA_KeyFunction1

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Key has no function (NONE)
- 1= The button triggers the command \fixhh{SENSOR_IMC5x00}{Exec_Master} with action set for the signal(s) defined by \fixhh{SENSOR_IMC5x00}{Set_KeyMasterSignalSelect} (MASTERSET)
- 2= The button triggers the command \fixhh{SENSOR_IMC5x00}{Exec_Master} with action reset for the signal(s) defined by \fixhh{SENSOR_IMC5x00}{Set_KeyMasterSignalSelect} (MASTERRESET)
- 3= Use key to toggle the pilot laser (PILOTLASER)
- 4= The button is used as a toggle for the SLED (see \fixhh{SENSOR_IMC5x00}{Set_SuperluminescentLED}) (SLED)

Description: Button function

12.1.1.1.21 Set_KeyFunction2 (KEYFUNC2)

Set the button functionality for timeslot 2 (2...5s)

Parameter: int32_t SP_KeyFunction2

SP_KeyFunction2

Direction: Down

Valid values:

- 0= Key has no function (NONE)
- 1= The button triggers the command [Exec_Master](#) with action set for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERSET)
- 2= The button triggers the command [Exec_Master](#) with action reset for the signal(s) defined by [Set_KeyMasterSignalSelect](#) (MASTERRESET)
- 3= Use key to toggle the pilot laser (PILOTLASER)
- 4= The button is used as a toggle for the SLED (see [Set_SuperluminescentLED](#)) (SLED)

Description: Button function

12.1.1.1.22 Get_KeyFunction2 (KEYFUNC2)

Get the button functionality for timeslot 2 (2...5s)

Parameter: int32_t SA_KeyFunction2

SA_KeyFunction2

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Key has no function (NONE)
- 1 = The button triggers the command \fixhh{SENSOR_IMC5x00}{Exec - Master} with action set for the signal(s) defined by \fixhh{SENSOR_IMC5x00}{Set_KeyMasterSignalSelect} (MASTERSET)
- 2 = The button triggers the command \fixhh{SENSOR_IMC5x00}{Exec - Master} with action reset for the signal(s) defined by \fixhh{SENSOR_IMC5x00}{Set_KeyMasterSignalSelect} (MASTERRESET)
- 3 = Use key to toggle the pilot laser (PILOTLASER)
- 4 = The button is used as a toggle for the SLED (see \fixhh{SENSOR_IMC5x00}{Set_SuperluminescentLED}) (SLED)

Description: Button function

12.1.1.1.23 Set_KeyMasterSignalSelect (KEYMASTERSIGNALSELECT)

Set selection of the measurement data signal(s) for mastering with key.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_MasterPeak1..16_Ch1

SP_MasterPeak1..16_Ch1

Direction: Down

Valid values:

- 0 = no
- 1 = yes

Description: Specify if peak 1 to 16 of first channel is mastered.

Parameter: String SP_MasterSignalNames

SP_MasterSignalNames

Direction: Down

Description: List of user defined signals (generated by [Set_Computation](#) or [Set_StatisticSignal](#)). The entries must be separated by spaces.

The special value ALL specify to master all possible signals. NONE disables any signals. It must not combined with any other parameter.

12.1.1.1.24 Get_KeyMasterSignalSelect (KEYMASTERSIGNALSELECT)

Get selection of the measurement data signal(s) for mastering with key.

Parameter: int32_t SA_MasterPeak1..16_Ch1

SA_MasterPeak1..16_Ch1

Direction: Up

Valid values:

- 0 = no
- 1 = yes

Description: Specify if peak 1 to 16 of first channel is mastered.

Parameter: String SA_MasterSignalNames SA_MasterSignalNames
Direction: Up
Description: List of user defined signals (generated by [Set_Computation](#) or [Set_StatisticSignal](#)). The entries are separated by spaces.
 The special value ALL specify to master all possible signals. NONE means no signal.

12.1.1.2 User level

12.1.1.2.1 Logout (LOGOUT)

Change user level to user.

12.1.1.2.2 Login (LOGIN)

Change user level to professional.

Parameter: String SP_Password SP_Password
Direction: Down
Description: Valid password to login.

12.1.1.2.3 Get_UserLevel (GETUSERLEVEL)

Retrieve actual user level.

Parameter: int32_t SA_UserLevel SA_UserLevel
Direction: Up
Valid values:
 -1 = Other user level (only for internal use)
 0 = User
 1 = Professional
Description: Actual user level.

12.1.1.2.4 Set_DefaultUser (STDUSER)

Set the default user level after booting the system.

Parameter: int32_t SP_DefaultUser SP_DefaultUser
Direction: Down
Valid values:
 0 = User
 1 = Professional
Description: Default user level.

12.1.1.2.5 Get_DefaultUser (STDUSER)

Get the default user level after booting the system.

Parameter: int32_t SA_DefaultUser SA_DefaultUser
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = User
 1 = Professional
Description: Default user level.

12.1.1.2.6 Set_Password (PASSWD)

Change the password for login.

Parameter: String SP_OldPassword SP_OldPassword
Direction: Down
Description: Old password.

Parameter: String SP_NewPassword SP_NewPassword
Direction: Down
Description: New password.

12.1.1.3 Sensor

12.1.1.3.1 Get_SensorTable (SENSORTABLE)

Get a list of all calibrated sensors.

Parameter: String SA_SensorTable SA_SensorTable
Direction: Up
Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_SensorTableCount SA_SensorTableCount
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 20
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Channel1, SA_Channel2, ...

Parameter: int32_t SA_Channel1..x SA_Channel1..x
Direction: Up
Valid values:
 1
Description: Sensor channel. Always 1

Parameter: int32_t SA_Pos1..x	SA_Pos1..x
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of the sensor head in the table.	
Parameter: String SA_Name1..x	SA_Name1..x
Direction: Up	
Description: Name of the sensor head in the table.	
Parameter: double SA_Range1..x	SA_Range1..x
Direction: Up	
Valid values:	
Minimum: 0.0	
Description: Measurement range of the sensor head in the table.	
Parameter: String SA_Unit1..x	SA_Unit1..x
Direction: Up	
Description: Unit of the measurement range.	
Parameter: String SA_Serial1..x	SA_Serial1..x
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor head in the table.	
Parameter: String SA_SensorType1..x	SA_SensorType1..x
Direction: Up	
Description: Type of the sensor head in the table, e.g. distance or thickness.	

12.1.1.3.2 Set_ActiveSensor (SENSORHEAD)

Change to another sensor head.

Parameter: int32_t SP_ActiveSensor	SP_ActiveSensor
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of new sensor head.	

12.1.1.3.3 Get_ActiveSensor (SENSORHEAD)

Get active sensor head.

Parameter: int32_t SA_ActiveSensor	SA_ActiveSensor
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description: Index of active sensor head.	

12.1.1.3.4 Get_SensorInfo (SENSORINFO)

Get information of active sensor head.

Parameter: int32_t SA_Pos	SA_Pos
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 19	
Description:	Index of the active sensor head in the table.
Parameter: String SA_Name	SA_Name
Direction: Up	
Description:	Name of the active sensor head.
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: 0.0	
Description:	Measurement range of the active sensor head.
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description:	Unit of the measurement range.
Parameter: String SA_Serial	SA_Serial
Direction: Up	
Valid values:	
Numeric value	
Description:	Serial number of the active sensor head.
Parameter: String SA_SensorType	SA_SensorType
Direction: Up	
Description:	Type of the active sensor head, e.g. distance or thickness.

12.1.1.3.5 Set_SuperluminescentLED (SLED)

Set state of superluminescent LED.

Parameter: int32_t SP_SuperluminescentLED	SP_SuperluminescentLED
Direction: Down	
Valid values:	
0= Off	
1= On	

Description: Superluminescent LED is enabled or disabled.

12.1.1.3.6 Get_SuperluminescentLED (SLED)

Get state of superluminescent LED.

Parameter: int32_t SA_SuperluminescentLED	SA_SuperluminescentLED
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	

Description: Superluminescent LED is enabled or disabled.

12.1.1.3.7 Set_PilotLaser (PIOTLASER)

Set state of pilot laser.

Parameter: int32_t SP_PilotLaser

SP_PilotLaser

Direction: Down

Valid values:

- 0= Off
- 1= On

Description: Pilot laser is enabled or disabled.

12.1.1.3.8 Get_PilotLaser (PIOTLASER)

Get state of pilot laser.

Parameter: int32_t SA_PilotLaser

SA_PilotLaser

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Off
- 1= On

Description: Pilot laser is enabled or disabled.

12.1.1.4 Triggering

12.1.1.4.1 Set_TriggerSource (TRIGGERSOURCE)

Set the trigger source (input).

Parameter: int32_t SP_TriggerSource

SP_TriggerSource

Direction: Down

Valid values:

- 0= None
- 1= Sync/Trig
- 2= TrigIn
- 3= Software
- 4= Encoder1
- 5= Encoder2
- 6= Encoder3

Description: Trigger source (input).

12.1.1.4.2 Get_TriggerSource (TRIGGERSOURCE)

Get the active trigger source (input).

Parameter: int32_t SA_TriggerSource

SA_TriggerSource

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Sync/Trig
- 2= TrigIn
- 3= Software
- 4= Encoder1
- 5= Encoder2
- 6= Encoder3

Description: Trigger source (input).

12.1.1.4.3 Set_TriggerMode (TRIGGERMODE)

Set the trigger mode.

Parameter: int32_t SP_TriggerMode SP_TriggerMode
Direction: Down
Valid values:
 0= Edge
 1= Level (PULSE)
Description: Trigger mode.

12.1.1.4.4 Get_TriggerMode (TRIGGERMODE)

Get the active trigger mode.

Parameter: int32_t SA_TriggerMode SA_TriggerMode
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Edge
 1= Level (PULSE)
Description: Trigger mode.

12.1.1.4.5 Set_TriggerMoment (TRIGGERAT)

Set the trigger time.

Parameter: int32_t SP_TriggerMoment SP_TriggerMoment
Direction: Down
Valid values:
 0= Input
 1= Output
Description: Trigger moment.

12.1.1.4.6 Get_TriggerMoment (TRIGGERAT)

Get the active trigger time.

Parameter: int32_t SA_TriggerMoment SA_TriggerMoment
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Input
 1= Output
Description: Trigger moment.

12.1.1.4.7 Set_TriggerInLogic (TRIGINLEVEL)

Set the logic of trigger input TrigIn.

Parameter: int32_t SP_TriggerInLogic

SP_TriggerInLogic

Direction: Down

Valid values:

0= TTL

1= HTL

Description: Logic of trigger input

12.1.1.4.8 Get_TriggerInLogic (TRIGINLEVEL)

Get the logic of trigger input TrigIn.

Parameter: int32_t SA_TriggerInLogic

SA_TriggerInLogic

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= TTL

1= HTL

Description: Logic of trigger input

12.1.1.4.9 Set_TriggerLevel (TRIGGERLEVEL)

Set the trigger level.

Parameter: int32_t SP_TriggerLevel

SP_TriggerLevel

Direction: Down

Valid values:

0= High

1= Low

Description: Trigger level.

12.1.1.4.10 Get_TriggerLevel (TRIGGERLEVEL)

Get the active trigger level.

Parameter: int32_t SA_TriggerLevel

SA_TriggerLevel

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= High

1= Low

Description: Trigger level.

12.1.1.4.11 Set_TriggerCount (TRIGGERCOUNT)

Set the number of values to measure at trigger.

Parameter: int32_t SP_TriggerCount

SP_TriggerCount

Direction: Down

Valid values:

Minimum: 0

Maximum: 16383

Description: Number of values to measure. 0 means no trigger (NONE), 16383 means endless measurement (INFINITE).

12.1.1.4.12 Get_TriggerCount (TRIGGERCOUNT)

Get the number of values to measure at trigger.

Parameter: int32_t SA_TriggerCount

SA_TriggerCount

Direction: Up

Valid values:

Minimum: 0

Maximum: 16383

Description: Number of values to measure. 0 means no trigger (NONE), 16383 means endless measurement (INFINITE).

12.1.1.4.13 Software_Trigger (TRIGGERSW)

Execute a software trigger.

12.1.1.4.14 Set_TriggerEncoderMin (TRIGGERENCMIN)

Set the minimum encoder value for triggering.

Parameter: double SP_TriggerEncoderMin

SP_TriggerEncoderMin

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Lower encoder limit. Above this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

12.1.1.4.15 Get_TriggerEncoderMin (TRIGGERENCMIN)

Get the minimum encoder value for triggering.

Parameter: double SA_TriggerEncoderMin

SA_TriggerEncoderMin

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Lower encoder limit. Above this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

12.1.1.4.16 Set_TriggerEncoderMax (TRIGGERENCMAX)

Set the maximum encoder value for triggering.

Parameter: double SP_TriggerEncoderMax

SP_TriggerEncoderMax

Direction: Down

Valid values:

Minimum: 1.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Upper encoder limit. Below this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

12.1.1.4.17 Get_TriggerEncoderMax (TRIGGERENCMAX)

Get the maximum encoder value for triggering.

Parameter: double SA_TriggerEncoderMax

SA_TriggerEncoderMax

Direction: Up

Valid values:

Minimum: 1.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Upper encoder limit. Below this value it will be triggered. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

12.1.1.4.18 Set_TriggerEncoderStepsize (TRIGGERENCSTEPSENSE)

Set the number of steps between triggering.

Parameter: double SP_TriggerEncoderStepsize

SP_TriggerEncoderStepsize

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 2147483647.0 (INT32_MAX)

Description: Number of encoder steps before trigger. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

12.1.1.4.19 Get_TriggerEncoderStepsize (TRIGGERENCSTEPSENSE)

Get the number of steps between triggering.

Parameter: double SA_TriggerEncoderStepsize

SA_TriggerEncoderStepsize

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 2147483647.0 (INT32_MAX)

Description: Number of encoder steps before trigger. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

12.1.1.5 Encoder

12.1.1.5.1 Set_EncoderInterpolation<n> (ENCINTERPOL<n>)

Set the interpolation for encoder n $\in \{1..3\}$.

Parameter: int32_t SP_EncoderInterpolation<n>

Direction: Down

Valid values:

1

2

4

SP_EncoderInterpolation<n>

Description: Encoder interpolation.

12.1.1.5.2 Get_EncoderInterpolation<n> (ENCINTERPOL<n>)

Get the interpolation of encoder n $\in \{1..3\}$.

Parameter: int32_t SA_EncoderInterpolation<n>

SA_EncoderInterpolation<n>

Direction: Up

Valid values:

1

2

4

Description: Encoder interpolation.

12.1.1.5.3 Set_Encoder3Enable (ENCODER3)

Set state of third encoder.

Parameter: int32_t SP_Encoder3Enable

SP_Encoder3Enable

Direction: Down

Valid values:

0= Off

1= On

Description: Third encoder is enabled or disabled.

If it is enabled, then the [Set_EncoderMode](#) commands will be set to no action.

12.1.1.5.4 Get_Encoder3Enable (ENCODER3)

Get state of third encoder.

Parameter: int32_t SA_Encoder3Enable

SA_Encoder3Enable

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Off

1= On

Description: Third encoder is enabled or disabled.

If it is enabled, then the [Set_EncoderMode](#) commands will be set to no action.

12.1.1.5.5 Set_EncoderMode<n> (ENCREF<n>)

Set the behaviour of encoder $n \in \{1..2\}$ when reference is reached.

Parameter: int32_t SP_EncoderMode<n>

SP_EncoderMode<n>

Direction: Down

Valid values:

0= No action (NONE)

1= Set encoder value to preset value only one time (ONE)

2= Set encoder value to preset value each time (EVER)

Description: Mode of encoder when reference is reached.

12.1.1.5.6 Get_EncoderMode<n> (ENCREF<n>)

Get the behaviour of encoder $n \in \{1..2\}$ when reference is reached.

Parameter: int32_t SA_EncoderMode<n>

SA_EncoderMode<n>

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= No action (NONE)

1= Set encoder value to preset value only one time (ONE)

2= Set encoder value to preset value each time (EVER)

Description: Mode of encoder when reference is reached.

12.1.1.5.7 Set_EncoderPreload<n> (ENCVALUE<n>)

Set preload value for encoder $n \in \{1..3\}$.

Parameter: double SP_EncoderPreload<n>

SP_EncoderPreload<n>

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

12.1.1.5.8 Get_EncoderPreload<n> (ENCVALUE<n>)

Get preload value for encoder $n \in \{1..3\}$.

Parameter: double SA_EncoderPreload<n>

SA_EncoderPreload<n>

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

12.1.1.5.9 Load_Encoder<n> (ENCSET)

Load the encoder n $\in \{1..3\}$ with the preset value.

12.1.1.5.10 EnableRef_Encoder<n> (ENCRESET)

Reset reference counter of encoder n $\in \{1..2\}$. If encoder mode is 2, at next reference encoder value will be set to preset value again

12.1.1.5.11 Set_Encoder.MaxValue<n> (ENCMAX<n>)

Set maximum value for encoder n $\in \{1..3\}$ before it wraps around.

Parameter: double SP_Encoder.MaxValue<n>

SP_Encoder.MaxValue<n>

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Maximum value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

12.1.1.5.12 Get_Encoder.MaxValue<n> (ENCMAX<n>)

Get maximum value for encoder n $\in \{1..3\}$ before it wraps around.

Parameter: double SA_Encoder.MaxValue<n>

SA_Encoder.MaxValue<n>

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Maximum value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

12.1.1.6 Interfaces

12.1.1.6.1 Set_IPConfiguration (IPCONFIG)

Set the IP configuration at controller.

For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_DHCPEnabled

SP_DHCPEnabled

Direction: Down

Valid values:

0= FALSE

1= TRUE

Description: Specify if controller should use a static IP address or ask for IP at DHCP server (dynamic IP address).

Parameter: String SP_Address	SP_Address
Direction: Down	
Valid values:	Valid IP address in form of xxx.xxx.xxx.xxx
Description:	IP address of the controller. This parameter is only evaluated on static IP assignment.
Parameter: String SP_SubnetMask	SP_SubnetMask
Direction: Down	
Valid values:	Valid network mask (e.g. 255.255.255.0 for a Class C network)
Description:	Network mask of the controller. This parameter is only evaluated on static IP assignment.
Parameter: String SP_Gateway	SP_Gateway
Direction: Down	
Valid values:	Valid IP address of default gateway in form of xxx.xxx.xxx.xxx
Description:	The default gateway must be specified if the controller should communicate with peers in foreign subnets. This parameter is only evaluated on static IP assignment.

12.1.1.6.2 Get_IPConfiguration (IPCONFIG)

Get the IP configuration at controller.

Parameter: int32_t SA_DHCPEnabled	SA_DHCPEnabled
Direction: Up	
Valid values:	-1 = Unknown parameter value from sensor 0 = FALSE 1 = TRUE
Description:	Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).
Parameter: String SA_Address	SA_Address
Direction: Up	
Valid values:	Valid IP address in form of xxx.xxx.xxx.xxx
Description:	IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.
Parameter: String SA_SubnetMask	SA_SubnetMask
Direction: Up	
Valid values:	Valid network mask (e.g. 255.255.255.0 for a Class C network)
Description:	Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.
Parameter: String SA_Gateway	SA_Gateway
Direction: Up	
Valid values:	Valid IP address of default gateway in form of xxx.xxx.xxx.xxx
Description:	Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

12.1.1.6.3 Set_IPDataTransferMode (MEATRANSFER)

Set IP protocol at controller.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_Protocol

SP_Protocol

Direction: Down

Valid values:

- 0= TCP server (SERVER/TCP)
- 1= TCP client (CLIENT/TCP)
- 2= UDP sender (CLIENT/UDP)
- 3= None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SP_RemoteAddress

SP_RemoteAddress

Direction: Down

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to. On TCP server this parameter is ignored.

Parameter: int32_t SP_Port

SP_Port

Direction: Down

Valid values:

- Minimum: 1024
- Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

12.1.1.6.4 Get_IPDataTransferMode (MEATRANSFER)

Get IP protocol at controller.

Parameter: int32_t SA_Protocol

SA_Protocol

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= TCP server (SERVER/TCP)
- 1= TCP client (CLIENT/TCP)
- 2= UDP sender (CLIENT/UDP)
- 3= None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SA_RemoteAddress

SA_RemoteAddress

Direction: Up

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to.

Parameter: int32_t SA_Port

SA_Port

Direction: Up

Valid values:

- Minimum: 1024
- Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

12.1.1.6.5 Set_TCPKeepAlive (TCPKEEPALIVE)

Set the TCP keep alive feature (see RFC 1122).

Parameter: int32_t SP_TCPKeepAlive SP_TCPKeepAlive
Direction: Down
Valid values:
 0= Off
 1= On
Description: TCP keep alive feature.

12.1.1.6.6 Get_TCPKeepAlive (TCPKEEPALIVE)

Get the TCP keep alive feature (see RFC 1122).

Parameter: int32_t SA_TCPKeepAlive SA_TCPKeepAlive
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Off
 1= On
Description: TCP keep alive feature.

12.1.1.6.7 Set_EthernetMode (ETHERMODE)

Switches ethernet mode between Ethernet and Ethercat.

Parameter: int32_t SP_EthernetMode SP_EthernetMode
Direction: Down
Valid values:
 0= Ethernet
 1= Ethercat
Description: Ethernet mode.

12.1.1.6.8 Get_EthernetMode (ETHERMODE)

Get ethernet mode of controller.

Parameter: int32_t SA_EthernetMode SA_EthernetMode
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Ethernet
 1= Ethercat
Description: Ethernet mode.

12.1.1.6.9 Set_Baudrate (BAUDRATE)

Set baudrate of controller for serial RS422 communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

4000000 (does not work in combination with IF2008 PCI(e) interface card less than FPGA 10 and IF2004 USB adapter less than FPGA 6)
 3000000
 2000000
 921600
 691200
 460800
 230400
 115200
 9600

Unit: Baud

Description: Baudrate of controller.

Parameter: int32_t CP_InterruptDataTransfer

CP_InterruptDataTransfer

Direction: Down

Valid values:

0= false
 1= true

Default: 0

Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Set_DataOutInterface](#)) and enabling it again at end (only if enabled before).

12.1.1.6.10 Get_Baudrate (BAUDRATE)

Get baudrate of controller for serial RS422 communication.

Parameter: int32_t SA_SensorBaudrate

SA_SensorBaudrate

Direction: Up

Valid values:

4000000
 3000000
 2000000
 921600
 691200
 460800
 230400
 115200
 9600

Unit: Baud

Description: Baudrate of controller.

12.1.1.6.11 Set_AppLanguage (LANGUAGE)

Set language of web interface.

Parameter: int32_t SP_ApplicationLanguage

SP_ApplicationLanguage

Direction: Down

Valid values:

0= English (EN)

1= German (DE)

Description: Language of web interface.

12.1.1.6.12 Get_AppLanguage (LANGUAGE)

Get language of web interface.

Parameter: int32_t SA_ApplicationLanguage

SA_ApplicationLanguage

Direction: Up

Valid values:

0= English (EN)

1= German (DE)

Description: Language of web interface.

12.1.1.7 Parameter management

12.1.1.7.1 Save_InterfaceParameters (BASICSETTINGS STORE)

Save actual interface parameters at controller.

12.1.1.7.2 Load_InterfaceParameters (BASICSETTINGS READ)

Load stored interface parameters into controller RAM.

12.1.1.7.3 Save_MeasureParameters (MEASSETTINGS STORE)

Save actual measurement parameters at controller.

Parameter: String SP_SettingName

SP_SettingName

Direction: Down

Description: Name of setting. If name is empty, current setting will be overwritten.

12.1.1.7.4 Load_MeasureParameters (MEASSETTINGS READ)

Load stored measurement parameters into controller RAM.

At this command the controller may change output data after applying new setting.

If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: String SP_SettingName

SP_SettingName

Direction: Down

Description: Name of setting.

12.1.1.7.5 Rename_MeasureParameters (MEASSETTINGS RENAME)

Rename stored measurement parameters at controller.

Parameter: String SP_OldName SP_OldName
Direction: Down
Description: Actual name of setting.

Parameter: String SP_NewName SP_NewName
Direction: Down
Description: New name of setting.

Parameter: int32_t SP_Overwrite SP_Overwrite
Direction: Down
Valid values:
 0 = no
 1 = yes
Description: Force overwriting existing setting.

12.1.1.7.6 Get_CurrentMeasureSetting (MEASSETTINGS CURRENT)

Get name of current (active) measurement parameters.

Parameter: String SA_CurrentName SA_CurrentName
Direction: Up
Description: Name of current (active) setting.

12.1.1.7.7 Get_MeasureSettingsList (MEASSETTINGS LIST)

Get list of all user settings.

Parameter: String SA_SettingNames SA_SettingNames
Direction: Up
Description: List (separated by new line).

12.1.1.7.8 Delete_MeasureParameters (MEASSETTINGS DELETE)

Delete stored measurement parameters at controller.

Parameter: String SP_SettingName SP_SettingName
Direction: Down
Description: Name of setting.

12.1.1.7.9 Get_MeasurePresetList (MEASSETTINGS PRESETLIST)

Get list of all preset settings.

Parameter: String SA_PresetNames SA_PresetNames
Direction: Up
Description: List (separated by new line).

12.1.1.7.10 Set_InitialMeasureSetting (MEASSETTINGS INITIAL)

Set initial (boot time) measure setting.

Parameter: int32_t SP_Automatic SP_Automatic

Direction: Down

Valid values:

0= no

1= yes

Description: Automatic selection.

Parameter: String SP_InitialName SP_InitialName

Direction: Down

Description: Name of setting.

12.1.1.7.11 Get_InitialMeasureSetting (MEASSETTINGS INITIAL)

Get initial (boot time) measure setting.

Parameter: int32_t SA_Automatic SA_Automatic

Direction: Up

Valid values:

0= no

1= yes

Description: Automatic selection.

Parameter: String SA_InitialName SA_InitialName

Direction: Up

Description: Name of setting.

12.1.1.7.12 Set_MeasurePresetMode (MEASSETTINGS PRESETMODE)

Set mode of current preset settings.

Parameter: int32_t SP_PresetMode SP_PresetMode

Direction: Down

Valid values:

0= Static

1= Balanced

2= Dynamic

3= None (if no preset setting is active)

Description: Preset mode.

12.1.1.7.13 Get_MeasurePresetMode (MEASSETTINGS PRESETMODE)

Get mode of current preset settings.

Parameter: int32_t SA_PresetMode SA_PresetMode

Direction: Up

Valid values:

0= Static

1= Balanced

2= Dynamic

3= None (if no preset setting is active)

Description: Preset mode.

12.1.1.7.14 Set_Default (SETDEFAULT)

Reset the controller to default settings.

At this command the sensor may change output data after applying default settings. If first bit of **IP_AutomaticMode** is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DefaultType

SP_DefaultType

Direction: Down

Valid values:

- 0= Delete all settings and load the factory settings (ALL)
- 1= Delete all measurement settings (MEASSETTINGS)
- 2= Delete all basic settings (BASICSETTINGS)
- 3= Overwrite and load the material table (MATERIAL)

Description: Specifies which settings should reseted.

12.1.1.7.15 Export_Parameters (EXPORT)

Exports the settings of the sensor.

Parameter: int32_t SP_ExportType

SP_ExportType

Direction: Down

Valid values:

- 0= Exports only the measuring settings with name (MEASSETTINGS)
- 1= Exports only the basic settings (BASICSETTINGS)
- 2= Exports all measuring settings (MEASSETTINGS_ALL)
- 3= Exports basic settings and all measuring settings (ALL)
- 4= Exports only the table of materials (MATERIALTABLE)

Description: Export type.

Parameter: String SP_SettingName

SP_SettingName

Direction: Down

Description: Name of setting to be exported.

This parameter is only used at export type 0.

Parameter: String SA_ExportData

SA_ExportData

Direction: Up

Description: Exported data in ASCII format.

12.1.1.7.16 Import_Parameters (IMPORT)

Imports the settings of the sensor.

If SP_ApplyImmediately is 1, the controller may change output data after applying new setting. If first bit of **IP_AutomaticMode** is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_ForceOverwrite

SP_ForceOverwrite

Direction: Down

Valid values:

- 0= Do not overwrite existing setting
- 1= Allow to overwrites existing settings (FORCE)

Description: Specify if existing settings can be overwritten.

Parameter: int32_t SP_ApplyImmediately SP_ApplyImmediately

Direction: Down

Valid values:

0= Just store imported settings.

1= Apply the imported settings (APPLY)

Description: Specify if settings only should be stored or additionally applied.

Parameter: String SP_ImportData SP_ImportData

Direction: Down

Description: Data to be imported (from a former call to Export_Parameters).

12.1.1.8 Internal controller commands

12.1.1.8.1 Set_ChartType (CHARTTYPE)

Set current type of the chart at web interface.

Parameter: int32_t SP_ChartType SP_ChartType

Direction: Down

Valid values:

0= Measure (MEAS)

1= Video signal (VIDEO)

Description: Type of the chart at web interface.

12.1.1.8.2 Get_ChartType (CHARTTYPE)

Get current type of the chart at web interface.

Parameter: int32_t SA_ChartType SA_ChartType

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Measure (MEAS)

1= Video signal (VIDEO)

Description: Type of the chart at web interface.

12.1.2 Measurement

12.1.2.1 General

12.1.2.1.1 Set_MeasurePeakSort (MEASPEAK_SORT)

Set how peaks will be sorted.

Parameter: int32_t SP_MeasurePeakSort SP_MeasurePeakSort

Direction: Down

Valid values:

0= Height

1= Distance

Description: Measure peak sort mode.

12.1.2.1.2 Get_MeasurePeakSort (MEASPEAK_SORT)

Get how peaks will be sorted.

Parameter: int32_t SA_MeasurePeakSort

SA_MeasurePeakSort

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Height
- 1 = Distance

Description: Measure peak sort mode.

12.1.2.1.3 Set_Samplerate (MEASRATE)

Set the samplerate.

Parameter: double SP_Measrate

SP_Measrate

Direction: Down

Valid values:

- Minimum:** 0.1
- Maximum:** 6.0

Unit: kHz

Description: Samplerate of measurement.

12.1.2.1.4 Get_Samplerate (MEASRATE)

Get the samplerate.

Parameter: double SA_Measrate

SA_Measrate

Direction: Up

Valid values:

- Minimum:** 0.1
- Maximum:** 6.0

Unit: kHz

Description: Samplerate of measurement.

12.1.2.1.5 Get_VideoStreamSignal

Read one video signal from video stream.

Parameter: int32_t SP_ReadMode

SP_ReadMode

Direction: Down

Valid values:

- 0 = Each video signal
- 1 = Only newest video signal
- 2 = Automatic

Description: This mode specifies if each video signal should be read or only the latest one. If set to automatic each video signal is read until the buffer does not overflow. If the buffer becomes full one or more video signals are discarded.

Parameter: int32_t SP_WaitVideoTimeout	SP_WaitVideoTimeout
Direction: Down	
Unit: ms	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Timeout to wait for a video signal.	
Parameter: Binary data SA_VideoAbs_Ch1	SA_VideoAbs_Ch1
Direction: Up	
Valid values:	
512 words (each 2 byte), each word is an intensity value.	
Description: Absolute signal of first channel	
Parameter: double SA_VideoTimestamp	SA_VideoTimestamp
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.79769e+308 (DBL_MAX)	
Unit: ms	
Description: Timestamp of the video signal. It starts from 1970 Jan 01 at 01:00. It is generated when the video has arrived at TCP/IP socket.	
Parameter: int32_t SA_SkippedVideo	SA_SkippedVideo
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of skipped video signals, if SP_ReadMode is not 0.	

12.1.2.2 Video signal

12.1.2.2.1 Set_ROI (ROI)

Set the region of interest for processing video signal.

Parameter: int32_t SP_ROIStart	SP_ROIStart
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: First position on CCD.	
Parameter: int32_t SP_ROIEnd	SP_ROIEnd
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 511	
Unit: Pixel	
Description: Last position on CCD.	

12.1.2.2.2 Get_ROI (ROI)

Get the region of interest for processing video signal

Parameter: int32_t SA_ROIStart SA_ROIStart

Direction: Up

Valid values:

Minimum: 0

Maximum: 511

Unit: Pixel

Description: First position on CCD.

Parameter: int32_t SA_ROIEnd SA_ROIEnd

Direction: Up

Valid values:

Minimum: 0

Maximum: 511

Unit: Pixel

Description: Last position on CCD.

12.1.2.3 Material database

12.1.2.3.1 Get_MaterialTable (MATERIALTABLE)

Get a list of all materials stored in the controller.

Valid for sensor:

IMC5400

Parameter: String SA_MaterialTable SA_MaterialTable

Direction: Up

Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_MaterialTableCount SA_MaterialTableCount

Direction: Up

Valid values:

Minimum: 0

Maximum: 20

Description: Number of entries in the table. All following parameters exist from 1 to this number, e.g. SA_MaterialName1, SA_MaterialName2, ...

Parameter: String SA_MaterialName_{1..x} SA_MaterialName_{1..x}

Direction: Up

Description: Name of the material in the table.

Parameter: double SA_GroupIndex_{1..x} SA_GroupIndex_{1..x}

Direction: Up

Valid values:

Minimum: 1.0

Maximum: 4.0

Description: Group index of the material.

Parameter: String SA_Description1..x SA_Description1..x
Direction: Up
Description: Description of the material in the table.

12.1.2.3.2 Set_ActiveMaterial (MATERIAL)

Select the material of thickness peak.

Valid for sensor:
 IMC5400

Parameter: String SP_ActiveMaterial SP_ActiveMaterial
Direction: Down
Description: Name of material.

12.1.2.3.3 Get_ActiveMaterial (MATERIAL)

Get the material of thickness peak.

Valid for sensor:
 IMC5400

Parameter: String SA_ActiveMaterial SA_ActiveMaterial
Direction: Up
Description: Name of material.

12.1.2.3.4 Get_MaterialInfo (MATERIALINFO)

Get information of active material.

Valid for sensor:
 IMC5400

Parameter: String SA_MaterialName SA_MaterialName
Direction: Up
Description: Name of the active material.

Parameter: String SA_Description SA_Description
Direction: Up
Description: Description of the active material.

Parameter: double SA_GroupIndex SA_GroupIndex
Direction: Up
Valid values:
Minimum: 1.0
Maximum: 4.0
Description: Group index.

12.1.2.3.5 Edit_Material (MATERIALEDIT)

Edit or add new material.

Valid for sensor:

IMC5400

Parameter: String SP_MaterialName SP_MaterialName

Direction: Down

Description: Name of the material.

Parameter: double SP_GroupIndex SP_GroupIndex

Direction: Down

Valid values:

Minimum: 1.0

Maximum: 4.0

Description: Group index of the material.

Parameter: String SP_Description SP_Description

Direction: Down

Description: Description of the material.

12.1.2.3.6 Delete_Material (MATERIALDELETE)

Deletes an existing material.

Valid for sensor:

IMC5400

Parameter: String SP_MaterialName SP_MaterialName

Direction: Down

Description: Name of the material to delete.

12.1.2.3.7 Clear_MaterialTable

Clear the whole material table.

Valid for sensor:

IMC5400

12.1.2.4 Peak processing

12.1.2.4.1 Set_PeakCount (PEAKCOUNT)

Set the maximum number of peaks.

This command is available from firmware version 001.050.032.

Parameter: int32_t SP_NumberOfPeaks SP_NumberOfPeaks

Direction: Down

Valid values:

Minimum: 1

Maximum: 16

Description: Maximum number of peaks.

12.1.2.4.2 Get_PeakCount (PEAKCOUNT)

Get the maximum number of peaks.

This command is available from firmware version 001.050.032.

Parameter: int32_t SA_NumberOfPeaks

SA_NumberOfPeaks

Direction: Up

Valid values:

Minimum: 1

Maximum: 16

Description: Maximum number of peaks.

12.1.2.4.3 Set_MaterialMultiPeak (MATERIALMP)

Set up to five materials for multipeak calculation.

This command is only available at Option 201.

For this command an [Update ...](#) meta command is available.

Parameter: String SP_MaterialMultiPeak12

SP_MaterialMultiPeak12

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between first and second peak.

Parameter: String SP_MaterialMultiPeak23

SP_MaterialMultiPeak23

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between second and third peak.

Parameter: String SP_MaterialMultiPeak34

SP_MaterialMultiPeak34

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between third and fourth peak.

Parameter: String SP_MaterialMultiPeak45

SP_MaterialMultiPeak45

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between fourth and fifth peak.

Parameter: String SP_MaterialMultiPeak56

SP_MaterialMultiPeak56

Direction: Down

Default: "" (empty string, means refractive index 1.0)

Description: Name of material between fifth and sixth peak.

12.1.2.4.4 Get_MaterialMultiPeak (MATERIALMP)

Get all material names for multipeak calculation.

This command is only available at Option 201.

At IFD2422 the first channel is referenced.

Parameter: String SA_MaterialMultiPeak12

SA_MaterialMultiPeak12

Direction: Up

Description: Name of material between first and second peak.

Parameter: String SA_MaterialMultiPeak23	SA_MaterialMultiPeak23
Direction: Up	
Description: Name of material between second and third peak.	
Parameter: String SA_MaterialMultiPeak34	SA_MaterialMultiPeak34
Direction: Up	
Description: Name of material between third and fourth peak.	
Parameter: String SA_MaterialMultiPeak45	SA_MaterialMultiPeak45
Direction: Up	
Description: Name of material between fourth and fifth peak.	
Parameter: String SA_MaterialMultiPeak56	SA_MaterialMultiPeak56
Direction: Up	
Description: Name of material between fifth and sixth peak.	

12.1.2.4.5 Set_MinimumThreshold (MIN_THRESHOLD)

Set the minimum threshold.

Parameter: double SP_MinimumThreshold	SP_MinimumThreshold
Direction: Down	
Valid values:	
Minimum: 0.05 for IMC5410, otherwise 0.5	
Maximum: 100.0	
Unit: %	
Description: A valid peak must be above this threshold.	

12.1.2.4.6 Get_MinimumThreshold (MIN_THRESHOLD)

Get the minimum threshold.

Parameter: double SA_MinimumThreshold	SA_MinimumThreshold
Direction: Up	
Valid values:	
Minimum: 0.05 for IMC5410, otherwise 0.5	
Maximum: 100.0	
Unit: %	
Description: A valid peak must be above this threshold.	

12.1.2.5 Measurement value processing

12.1.2.5.1 Set_StatisticSignal (STATISTICSIGNAL)

Set the measured value which is used for statistic calculation.

Parameter: int32_t SP_StatisticSignal

SP_StatisticSignal

Direction: Down

Valid values:

- 1= Use SP_StatisticSignalName
- 0= Peak 1 of first Channel
- 1= Peak 2 of first Channel
- 2= Peak 3 of first Channel
- 3= Peak 4 of first Channel
- 4= Peak 5 of first Channel
- 5= Peak 6 of first Channel
- 6= Peak 7 of first Channel
- 7= Peak 8 of first Channel
- 8= Peak 9 of first Channel
- 9= Peak 10 of first Channel
- 10= Peak 11 of first Channel
- 11= Peak 12 of first Channel
- 12= Peak 13 of first Channel
- 13= Peak 14 of first Channel
- 14= Peak 15 of first Channel
- 15= Peak 16 of first Channel

Description: Signal which is used for statistic calculation.

Parameter: String SP_StatisticSignalName

SP_StatisticSignalName

Direction: Down

Description: User defined signal which is used for statistic calculation.

Parameter: int32_t SP_StatisticDepth

SP_StatisticDepth

Direction: Down

Valid values:

- 0= None (deletes the specified statistic calculation)
- 2, 4, 8, ..., 8192, 16384
- 16385= Infinite (endless statistic calculation)

Description: Number of values (depth) for statistic calculation.

12.1.2.5.2 Get_StatisticSignal (STATISTICSIGNAL)

Get the measured value which is used for statistic calculation.

Parameter: int32_t SP_StatisticSignal

SP_StatisticSignal

Direction: Down

Valid values:

- 1= Use SP_StatisticSignalName
- 0= Peak 1 of first Channel
- 1= Peak 2 of first Channel
- 2= Peak 3 of first Channel
- 3= Peak 4 of first Channel

4= Peak 5 of first Channel
 5= Peak 6 of first Channel
 6= Peak 7 of first Channel
 7= Peak 8 of first Channel
 8= Peak 9 of first Channel
 9= Peak 10 of first Channel
 10= Peak 11 of first Channel
 11= Peak 12 of first Channel
 12= Peak 13 of first Channel
 13= Peak 14 of first Channel
 14= Peak 15 of first Channel
 15= Peak 16 of first Channel

Description: Signal which is used for statistic calculation.

Parameter: String SP_StatisticSignalName

SP_StatisticSignalName

Direction: Down

Description: User defined signal which is used for statistic calculation.

Parameter: String SA_StatisticSignalTable

SA_StatisticSignalTable

Direction: Up

Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_StatisticSignalEntries

SA_StatisticSignalEntries

Direction: Up

Valid values:

Minimum: 0

Maximum: 10

Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_StatisticSignalEntry1, SA_StatisticSignalEntry2, ...

Parameter: String SA_StatisticSignalEntry_{1..x}

SA_StatisticSignalEntry_{1..x}

Direction: Up

Description: Each statistic signal as complete line.

Parameter: int32_t SA_StatisticSignalIndex_{1..x}

SA_StatisticSignalIndex_{1..x}

Direction: Up

Valid values:

-1= User defined name at SA_StatisticSignalName1..x
 0= Peak 1 of first Channel
 1= Peak 2 of first Channel
 2= Peak 3 of first Channel
 3= Peak 4 of first Channel
 4= Peak 5 of first Channel
 5= Peak 6 of first Channel
 6= Peak 7 of first Channel
 7= Peak 8 of first Channel
 8= Peak 9 of first Channel
 9= Peak 10 of first Channel
 10= Peak 11 of first Channel
 11= Peak 12 of first Channel
 12= Peak 13 of first Channel
 13= Peak 14 of first Channel
 14= Peak 15 of first Channel
 15= Peak 16 of first Channel

Description: Index of each statistic signal.

Parameter: String SA_StatisticSignalName <code>1..x</code>	SA_StatisticSignal- Name <code>1..x</code>
Direction: Up	
Description: Name of each statistic signal.	
Parameter: int32_t SA_StatisticDepth <code>1..x</code>	SA_StatisticDepth <code>1..x</code>
Direction: Up	
Valid values:	
2, 4, 8, ..., 8192, 16384	
16385= Infinite (endless statistic calculation)	
Description: Number of values (depth) for statistic calculation.	

12.1.2.5.3 Reset_StatisticSignal (STATIC RESET)

Reset the values of the statistic signals.

Parameter: int32_t SP_StatisticSignal	SP_StatisticSignal
Direction: Down	
Valid values:	
-1= Use SP_StatisticSignalName	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	
7= Peak 8 of first Channel	
8= Peak 9 of first Channel	
9= Peak 10 of first Channel	
10= Peak 11 of first Channel	
11= Peak 12 of first Channel	
12= Peak 13 of first Channel	
13= Peak 14 of first Channel	
14= Peak 15 of first Channel	
15= Peak 16 of first Channel	
16= All	
Description: Signal which is used for statistic calculation.	

Parameter: String SP_StatisticSignalName	SP_StatisticSignalName
Direction: Down	
Description: User defined signal which is used for statistic calculation.	

12.1.2.5.4 Set_MasterSignal (MASTERSIGNAL)

Set the measured value which is used for mastering.

Parameter: int32_t SP_MasterSignal	SP_MasterSignal
Direction: Down	
Valid values:	
-1= Use SP_MasterSignalName	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	

2= Peak 3 of first Channel
 3= Peak 4 of first Channel
 4= Peak 5 of first Channel
 5= Peak 6 of first Channel
 6= Peak 7 of first Channel
 7= Peak 8 of first Channel
 8= Peak 9 of first Channel
 9= Peak 10 of first Channel
 10= Peak 11 of first Channel
 11= Peak 12 of first Channel
 12= Peak 13 of first Channel
 13= Peak 14 of first Channel
 14= Peak 15 of first Channel
 15= Peak 16 of first Channel

Description: Signal which is used for mastering.

Parameter: String SP_MasterSignalName

SP_MasterSignalName

Direction: Down

Description: User defined signal which is used for mastering.

Parameter: double SP_MasterValue

SP_MasterValue

Direction: Down

Valid values:

Minimum: -21.47

Maximum: 21.47

Unit: mm

Description: Master value

12.1.2.5.5 Get_MasterSignal (MASTERSIGNAL)

Get the measured value which is used for mastering.

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

-1= Use SP_MasterSignalName
 0= Peak 1 of first Channel
 1= Peak 2 of first Channel
 2= Peak 3 of first Channel
 3= Peak 4 of first Channel
 4= Peak 5 of first Channel
 5= Peak 6 of first Channel
 6= Peak 7 of first Channel
 7= Peak 8 of first Channel
 8= Peak 9 of first Channel
 9= Peak 10 of first Channel
 10= Peak 11 of first Channel
 11= Peak 12 of first Channel
 12= Peak 13 of first Channel
 13= Peak 14 of first Channel
 14= Peak 15 of first Channel
 15= Peak 16 of first Channel

Description: Signal which is used for mastering.

Parameter: String SP_MasterSignalName	SP_MasterSignalName
Direction: Down	
Description: User defined signal which is used for mastering.	
Parameter: String SA_MasterSignalTable	SA_MasterSignalTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_MasterSignalEntries	SA_MasterSignalEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MasterSignalEntry1, SA_MasterSignalEntry2, ...	
Parameter: String SA_MasterSignalEntry _{1..x}	SA_MasterSignalEntry _{1..x}
Direction: Up	
Description: Each master signal as complete line.	
Parameter: int32_t SA_MasterSignalIndex _{1..x}	SA_MasterSignalIndex _{1..x}
Direction: Up	
Valid values:	
-1= User defined name at SA_MasterSignalName1..x	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	
7= Peak 8 of first Channel	
8= Peak 9 of first Channel	
9= Peak 10 of first Channel	
10= Peak 11 of first Channel	
11= Peak 12 of first Channel	
12= Peak 13 of first Channel	
13= Peak 14 of first Channel	
14= Peak 15 of first Channel	
15= Peak 16 of first Channel	
Description: Index of each master signal.	
Parameter: String SA_MasterSignalName _{1..x}	SA_MasterSignalName _{1..x}
Direction: Up	
Description: Name of each master signal.	
Parameter: double SA_MasterValue _{1..x}	SA_MasterValue _{1..x}
Direction: Up	
Valid values:	
Minimum: -21.47	
Maximum: 21.47	
Unit: mm	
Description: Master value	

12.1.2.5.6 Reset_MasterSignal (MASTERSIGNAL)

Reset the measured value which is used for mastering.

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

- 1= Use SP_MasterSignalName
- 0= Peak 1 of first Channel
- 1= Peak 2 of first Channel
- 2= Peak 3 of first Channel
- 3= Peak 4 of first Channel
- 4= Peak 5 of first Channel
- 5= Peak 6 of first Channel
- 6= Peak 7 of first Channel
- 7= Peak 8 of first Channel
- 8= Peak 9 of first Channel
- 9= Peak 10 of first Channel
- 10= Peak 11 of first Channel
- 11= Peak 12 of first Channel
- 12= Peak 13 of first Channel
- 13= Peak 14 of first Channel
- 14= Peak 15 of first Channel
- 15= Peak 16 of first Channel

Description: Signal which should be reset.

Parameter: String SP_MasterSignalName

SP_MasterSignalName

Direction: Down

Description: User defined signal which should be reset.

12.1.2.5.7 Exec_Master (MASTER)

Executes mastering for specified signal.

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

- 1= Use SP_MasterSignalName
- 0= Peak 1 of first Channel
- 1= Peak 2 of first Channel
- 2= Peak 3 of first Channel
- 3= Peak 4 of first Channel
- 4= Peak 5 of first Channel
- 5= Peak 6 of first Channel
- 6= Peak 7 of first Channel
- 7= Peak 8 of first Channel
- 8= Peak 9 of first Channel
- 9= Peak 10 of first Channel
- 10= Peak 11 of first Channel
- 11= Peak 12 of first Channel
- 12= Peak 13 of first Channel
- 13= Peak 14 of first Channel
- 14= Peak 15 of first Channel
- 15= Peak 16 of first Channel
- 16= All

Description: Signal which should be mastered.

Parameter: String SP_MasterSignalName	SP_MasterSignalName
Direction: Down	
Description: User defined signal which should be mastered.	
Parameter: int32_t SP_MasterAction	SP_MasterAction
Direction: Down	
Valid values:	
0= Reset	
1= Set	
Description: Mastering action	

12.1.2.5.8 Get_Master (MASTER)

Get mastering for specified signal.

Parameter: int32_t SP_Master	SP_Master
Direction: Down	
Valid values:	
-1= Use SP_MasterName	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	
7= Peak 8 of first Channel	
8= Peak 9 of first Channel	
9= Peak 10 of first Channel	
10= Peak 11 of first Channel	
11= Peak 12 of first Channel	
12= Peak 13 of first Channel	
13= Peak 14 of first Channel	
14= Peak 15 of first Channel	
15= Peak 16 of first Channel	
16= All	
Description: Signal which should be retrieved.	

Parameter: String SP_MasterName	SP_MasterName
Direction: Down	
Description: User defined which should be retrieved.	

Parameter: String SA_MasterTable	SA_MasterTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	

Parameter: int32_t SA_MasterEntries	SA_MasterEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MasterEntry1, SA_MasterEntry2, ...	

Parameter: String SA_MasterEntry1..x	SA_MasterEntry1..x
Direction: Up	
Description: Each master as complete line.	
Parameter: int32_t SA_MasterIndex1..x	SA_MasterIndex1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_MasterName1..x	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	
7= Peak 8 of first Channel	
8= Peak 9 of first Channel	
9= Peak 10 of first Channel	
10= Peak 11 of first Channel	
11= Peak 12 of first Channel	
12= Peak 13 of first Channel	
13= Peak 14 of first Channel	
14= Peak 15 of first Channel	
15= Peak 16 of first Channel	
Description: Index of each master.	
Parameter: String SA_MasterName1..x	SA_MasterName1..x
Direction: Up	
Description: Name of each master.	
Parameter: int32_t SA_MasterAction1..x	SA_MasterAction1..x
Direction: Up	
Valid values:	
0= Inactive	
1= Active	
Description: Mastering action	

12.1.2.5.9 Set_Computation (COMP)

Create/modify customised measurement data processing computations.

Parameter: int32_t SP_ComputationQueue	SP_ComputationQueue
Direction: Down	
Valid values:	
0= Channel 1	
Description: Computation queue	
Parameter: int32_t SP_ComputationID	SP_ComputationID
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Index within computation queue	

Parameter: int32_t SP_ComputationOperation	SP_ComputationOperation
Direction: Down	
Valid values:	
0= None	
1= Median	
2= Moving average	
3= Recursive averaging	
4= Calculation (only available at Option 201)	
5= Thickness (only available at Option 201)	
6= Copy	
Description: Computation operation	
Parameter: int32_t SP_ComputationSignalIndex	SP_ComputationSignalIndex
Direction: Down	
Valid values:	
-1= Use SP_ComputationSignalName	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	
7= Peak 8 of first Channel	
8= Peak 9 of first Channel	
9= Peak 10 of first Channel	
10= Peak 11 of first Channel	
11= Peak 12 of first Channel	
12= Peak 13 of first Channel	
13= Peak 14 of first Channel	
14= Peak 15 of first Channel	
15= Peak 16 of first Channel	
Description: Signal index (if operation is averaging or copy).	
Parameter: String SP_ComputationSignalName	SP_ComputationSignalName
Direction: Down	
Description: User defined signal (if operation is averaging or copy).	
Parameter: int32_t SP_ComputationMovingCount	SP_ComputationMovingCount
Direction: Down	
Valid values:	
2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096	
Description: Number of value for the averaging window. This parameter is only used at moving average.	
Parameter: int32_t SP_ComputationRecursiveCount	SP_ComputationRecursiveCount
Direction: Down	
Valid values:	
Minimum: 2	
Maximum: 32768	
Description: Number of values for recursive averaging. This parameter is only used at recursive average.	

Parameter: int32_t SP_ComputationMedianCount	SP_ComputationMedianCount
Direction: Down	
Valid values:	
3, 5, 7, 9	
Description:	Number of values to build median. This parameter is only used at median.
Parameter: double SP_ComputationFactor1	SP_ComputationFactor1
Direction: Down	
Valid values:	
Minimum: -32767.0	
Maximum: 32767.0	
Description:	Multiplication factor for first signal (if operation is calculation).
Parameter: int32_t SP_ComputationSignalIndex1	SP_ComputationSignalIndex1
Direction: Down	
Valid values:	
-1= Use SP_ComputationSignalName1	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	
7= Peak 8 of first Channel	
8= Peak 9 of first Channel	
9= Peak 10 of first Channel	
10= Peak 11 of first Channel	
11= Peak 12 of first Channel	
12= Peak 13 of first Channel	
13= Peak 14 of first Channel	
14= Peak 15 of first Channel	
15= Peak 16 of first Channel	
Description:	First signal index (if operation is calculation or thickness).
Parameter: String SP_ComputationSignalName1	SP_ComputationSignalName1
Direction: Down	
Description:	User defined first signal (if operation is calculation or thickness).
Parameter: double SP_ComputationFactor2	SP_ComputationFactor2
Direction: Down	
Valid values:	
Minimum: -32767.0	
Maximum: 32767.0	
Description:	Multiplication factor for second signal (if operation is calculation).
Parameter: int32_t SP_ComputationSignalIndex2	SP_ComputationSignalIndex2
Direction: Down	
Valid values:	
-1= Use SP_ComputationSignalName2	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	

3= Peak 4 of first Channel
 4= Peak 5 of first Channel
 5= Peak 6 of first Channel
 6= Peak 7 of first Channel
 7= Peak 8 of first Channel
 8= Peak 9 of first Channel
 9= Peak 10 of first Channel
 10= Peak 11 of first Channel
 11= Peak 12 of first Channel
 12= Peak 13 of first Channel
 13= Peak 14 of first Channel
 14= Peak 15 of first Channel
 15= Peak 16 of first Channel

Description: Second signal index (if operation is calculation or thickness).

Parameter: String SP_ComputationSignalName2

SP_ComputationSignalName2

Direction: Down

Description: User defined second signal (if operation is calculation or thickness).

Parameter: double SP_ComputationOffset

SP_ComputationOffset

Direction: Down

Valid values:

Minimum: -21.47

Maximum: 21.47

Unit: mm

Description: Additional offset (if operation is calculation).

Parameter: String SP_ComputationName

SP_ComputationName

Direction: Down

Description: Name of the newly created signal (if operation is calculation, thickness or copy).

12.1.2.5.10 Get_Computation (COMP)

Get customised measurement data processing computations.

Parameter: int32_t SP_ComputationQueue

SP_ComputationQueue

Direction: Down

Valid values:

-1= Any queue

0= Channel 1

Default: -1

Description: Computation queue

Parameter: int32_t SP_ComputationID

SP_ComputationID

Direction: Down

Valid values:

Minimum: 1

Maximum: 10

Description: Index within computation queue. Ignored if SP_ComputationQueue is -1.

Parameter: String SA_ComputationTable	SA_ComputationTable
Direction: Up	
Description: Whole table in one string, separated by new lines and commas.	
Parameter: int32_t SA_ComputationEntries	SA_ComputationEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 30	
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_ComputationEntry1, SA_ComputationEntry2, ...	
Parameter: String SA_ComputationEntry1..x	SA_ComputationEntry1..x
Direction: Up	
Description: Each computation as complete line.	
Parameter: int32_t SA_ComputationQueue1..x	SA_ComputationQueue1..x
Direction: Up	
Valid values:	
0= Channel 1	
Description: Computation queue	
Parameter: int32_t SA_ComputationID1..x	SA_ComputationID1..x
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Index within computation queue	
Parameter: int32_t SA_ComputationOperation1..x	SA_ComputationOperation1..x
Direction: Up	
Valid values:	
0= None	
1= Median	
2= Moving average	
3= Recursive averaging	
4= Calculation (only available at Option 201)	
5= Thickness (only available at Option 201)	
6= Copy	
Description: Computation operation	
Parameter: int32_t SA_ComputationSignalIndex1..x	SA_ComputationSignalIndex1..x
Direction: Up	
Valid values:	
-1= User defined name at SA_ComputationSignalName1..x	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	
7= Peak 8 of first Channel	

8= Peak 9 of first Channel
 9= Peak 10 of first Channel
 10= Peak 11 of first Channel
 11= Peak 12 of first Channel
 12= Peak 13 of first Channel
 13= Peak 14 of first Channel
 14= Peak 15 of first Channel
 15= Peak 16 of first Channel

Description: Signal index (if operation is averaging or copy).

Parameter: String SA_ComputationSignalName`1..x`

Direction: Up

Description: User defined signal (if operation is averaging or copy).

SA_ComputationSignal-
Name`1..x`

Parameter: int32_t SA_ComputationMovingCount`1..x`

Direction: Up

Valid values:

2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096

Description: Number of value for the averaging window. This parameter is only used at moving average.

SA_ComputationMoving-
Count`1..x`

Parameter: int32_t SA_ComputationRecursiveCount`1..x`

Direction: Up

Valid values:

Minimum: 2

Maximum: 32768

Description: Number of values for recursive averaging. This parameter is only used at recursive average.

SA_ComputationRecursive-
Count`1..x`

Parameter: int32_t SA_ComputationMedianCount`1..x`

Direction: Up

Valid values:

3, 5, 7, 9

Description: Number of values to build median. This parameter is only used at median.

SA_ComputationMedian-
Count`1..x`

Parameter: double SA_ComputationFactor`1..x`

Direction: Up

Valid values:

Minimum: -32767.0

Maximum: 32767.0

Description: Multiplication factor for first signal (if operation is calculation).

SA_ComputationFactor`1..x`

Parameter: String SA_ComputationName`1..x`

Direction: Up

Description: Name of the newly created signal (if operation is calculation, thickness or copy).

SA_ComputationName`1..x`

12.1.3 Data output

12.1.3.1 General

12.1.3.1.1 Set_DataOutInterface (OUTPUT)

Set the active interface for data output.

At this command the controller may change output data automatically. If first bit of [IP_AutomaticMode](#) is set (1), [Get_OutputInfo_RS422](#) resp. [Get_OutputInfo_ETH](#) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputRS422

SP_OutputRS422

Direction: Down

Valid values:

0= No

1= Yes

Description: Output active at RS422 interface.

Parameter: int32_t SP_OutputAnalog

SP_OutputAnalog

Direction: Down

Valid values:

0= No

1= Yes

Description: Output active at analog interface.

Parameter: int32_t SP_OutputEthernet

SP_OutputEthernet

Direction: Down

Valid values:

0= No

1= Yes

Description: Output active at ethernet interface.

Parameter: int32_t SP_OutputErrorOut

SP_OutputErrorOut

Direction: Down

Valid values:

0= No

1= Yes

Description: Output active at error outputs.

12.1.3.1.2 Get_DataOutInterface (OUTPUT)

Get the active interface for data output.

Parameter: int32_t SA_OutputRS422

SA_OutputRS422

Direction: Up

Valid values:

0= No

1= Yes

Description: Output active at RS422 interface.

Parameter: int32_t SA_OutputAnalog	SA_OutputAnalog
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at analog interface.	
Parameter: int32_t SA_OutputEthernet	SA_OutputEthernet
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at ethernet interface.	
Parameter: int32_t SA_OutputErrorOut	SA_OutputErrorOut
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Output active at error outputs.	

12.1.3.1.3 Set_ResamplingDevice (OUTREDUCEDEVICE)

Set the devices for which resampling is active.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_ResampleRS422	SP_ResampleRS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if RS422 output should be resampled.	
Parameter: int32_t SP_ResampleAnalog	SP_ResampleAnalog
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if analog output should be resampled.	
Parameter: int32_t SP_ResampleEthernet	SP_ResampleEthernet
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if ethernet output should be resampled.	

12.1.3.1.4 Get_ResamplingDevice (OUTREDUCEDEVICE)

Get the devices for which resampling is active.

Parameter: int32_t SA_ResampleRS422	SA_ResampleRS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: RS422 output is resampled.	

Parameter: int32_t SA_ResampleAnalog	SA_ResampleAnalog
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Analog output is resampled.	
Parameter: int32_t SA_ResampleEthernet	SA_ResampleEthernet
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Ethernet output is resampled.	

12.1.3.1.5 Set_ResamplingCount (OUTREDUCECOUNT)

Set reduce count for resampling.

Parameter: int32_t SP_Resampling	SP_Resampling
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 3000000	
Description: Resampling value.	

12.1.3.1.6 Get_ResamplingCount (OUTREDUCECOUNT)

Get reduce count for resampling.

Parameter: int32_t SA_Resampling	SA_Resampling
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 3000000	
Description: Resampling value.	

12.1.3.1.7 Set_HoldLastValid (OUTHOLD)

Set the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SP_HoldLastValid	SP_HoldLastValid
Direction: Down	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	

12.1.3.1.8 Get_HoldLastValid (OUTHOLD)

Get the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SA_HoldLastValid

SA_HoldLastValid

Direction: Up

Valid values:

Minimum: -1

Maximum: 1024

Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).

12.1.3.1.9 Set_FramesPerPacket_ETH (MEASCNT_ETH)

Set the maximum number of frames in ethernet packet.

Parameter: int32_t SP_FramesPerPacket_ETH

SP_FramesPerPacket_ETH

Direction: Down

Valid values:

Minimum: 0

Maximum: 350

Description: Maximum number of frames in ethernet packet. 0 means automatic.

12.1.3.1.10 Get_FramesPerPacket_ETH (MEASCNT_ETH)

Get the maximum number of frames in ethernet packet.

Parameter: int32_t SA_FramesPerPacket_ETH

SA_FramesPerPacket_ETH

Direction: Up

Valid values:

Minimum: 0

Maximum: 350

Description: Maximum number of frames in ethernet packet. 0 means automatic.

12.1.3.2 Selected measurement values

12.1.3.2.1 Set_Output_RS422 (OUT_RS422)

Set the data to be output at RS422 interface.

For this command an [Update...](#) and a [Reset...](#) meta command is available.

Parameter: int32_t SP_OutputVideoAbs_Ch1_RS422

SP_OutputVideoAbs_Ch1_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if absolute signal of first channel is transmitted.

Parameter: int32_t SP_OutputPeak1..16_Ch1_RS422	SP_OutputPeak1..16_Ch1_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if peak 1 to 16 of first channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalShutterTime_Ch1_RS422	SP_OutputAdditionalShutterTime_Ch1_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time of first channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalEncoder1..3_Ch1_RS422	SP_OutputAdditionalEncoder1..3_Ch1_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 to 3 of first channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalAmount_Ch1_RS422	SP_OutputAdditionalAmount_Ch1_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if amount of first channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalMeasrate_RS422	SP_OutputAdditionalMeasrate_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measurement rate is transmitted.	
Parameter: int32_t SP_OutputAdditionalTimestamp_RS422	SP_OutputAdditionalTimestamp_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SP_OutputAdditionalCounter_RS422	SP_OutputAdditionalCounter_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SP_OutputAdditionalState_RS422	SP_OutputAdditionalState_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	

Parameter: String SP_OutputCalculation_RS422	SP_OutputCalculation_RS422
Direction: Down	
Description: List of calculated signals (generated by Set_Computation). The entries must be separated by spaces.	
Parameter: String SP_OutputStatistic_RS422	SP_OutputStatistic_RS422
Direction: Down	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries must be separated by spaces.	
12.1.3.2.2 Get_Output_RS422 (OUT_RS422)	
Get the data which is output at RS422 interface.	
Parameter: int32_t SA_OutputVideoAbs_Ch1_RS422	SA_OutputVideoAbs_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if absolute signal of first channel is transmitted.	
Parameter: int32_t SA_OutputPeak1..16_Ch1_RS422	SA_OutputPeak1..16_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if peak 1 to 16 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_RS422	SA_OutputAdditionalShutterTime_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalEncoder1..3_Ch1_RS422	SA_OutputAdditionalEncoder1..3_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 to 3 of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalAmount_Ch1_RS422	SA_OutputAdditionalAmount_Ch1_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if amount of first channel is transmitted.	
Parameter: int32_t SA_OutputAdditionalMeasrate_RS422	SA_OutputAdditionalMeasrate_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measurement rate is transmitted.	

Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: String SA_OutputCalculation_RS422	SA_OutputCalculation_RS422
Direction: Up	
Description: List of calculated signals (generated by Set_Computation). The entries are separated by spaces.	
Parameter: String SA_OutputStatistic_RS422	SA_OutputStatistic_RS422
Direction: Up	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries are separated by spaces.	

12.1.3.2.3 Set_Output_ETH (OUT_ETH)

Set the data to be output at ethernet interface.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputVideoAbs_Ch1_ETH	SP_OutputVideoAbs_Ch1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if absolute signal of first channel is transmitted.	
Parameter: int32_t SP_OutputPeak1..16_Ch1_ETH	SP_OutputPeak1..16_Ch1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if peak 1 to 16 of first channel is transmitted.	

Parameter: int32_t SP_OutputAdditionalShutterTime_Ch1_ETH	SP_OutputAdditionalShutterTime_Ch1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time of first channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalEncoder1..3_Ch1_ETH	SP_OutputAdditionalEncoder1..3_Ch1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if encoder 1 to 3 of first channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalAmount_Ch1_ETH	SP_OutputAdditionalAmount_Ch1_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if amount of first channel is transmitted.	
Parameter: int32_t SP_OutputAdditionalMeasrate_ETH	SP_OutputAdditionalMeasrate_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measurement rate is transmitted.	
Parameter: int32_t SP_OutputAdditionalTimestamp_ETH	SP_OutputAdditionalTimestamp_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SP_OutputAdditionalCounter_ETH	SP_OutputAdditionalCounter_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SP_OutputAdditionalState_ETH	SP_OutputAdditionalState_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: String SP_OutputCalculation_ETH	SP_OutputCalculation_ETH
Direction: Down	
Description: List of calculated signals (generated by Set_Computation). The entries must be separated by spaces.	

Parameter: String SP_OutputStatistic_ETH SP_OutputStatistic_ETH
Direction: Down
Description: List of statistic signals (generated by [Set_StatisticSignal](#)). The entries must be separated by spaces.

12.1.3.2.4 Get_Output_ETH (OUT_ETH)

Get the data which is output at ethernet interface.

Parameter: int32_t SA_OutputVideoAbs_Ch1_ETH SA_OutputVideoAbs_Ch1_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if absolute signal of first channel is transmitted.

Parameter: int32_t SA_OutputPeak1..16_Ch1_ETH SA_OutputPeak1..16_Ch1_-
ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if peak 1 to 16 of first channel is transmitted.

Parameter: int32_t SA_OutputAdditionalShutterTime_Ch1_ETH SA_OutputAdditionalShut-
terTime_Ch1_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if shutter time of first channel is transmitted.

Parameter: int32_t SA_OutputAdditionalEncoder1..3_Ch1_ETH SA_OutputAdditionalEn-
coder1..3_Ch1_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if encoder 1 to 3 of first channel is transmitted.

Parameter: int32_t SA_OutputAdditionalAmount_Ch1_ETH SA_OutputAdditionalAm-
ount_Ch1_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if amount of first channel is transmitted.

Parameter: int32_t SA_OutputAdditionalMeasrate_ETH SA_OutputAdditionalMeas-
rate_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if measurement rate is transmitted.

Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditionalCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_ETH	SA_OutputAdditionalState_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: String SA_OutputCalculation_ETH	SA_OutputCalculation_ETH
Direction: Up	
Description: List of calculated signals (generated by Set_Computation). The entries are separated by spaces.	
Parameter: String SA_OutputStatistic_ETH	SA_OutputStatistic_ETH
Direction: Up	
Description: List of statistic signals (generated by Set_StatisticSignal). The entries are separated by spaces.	

12.1.3.3 Switching outputs

12.1.3.3.1 Set_ErrorLimitSignal< n > (ERRORLIMITSIGNAL< n >)

Set the signal for error limit $n \in \{1..2\}$.

Parameter: int32_t SP_ErrorLimitSignalIndex< n >	SP_ErrorLimitSignalIndex< n >
Direction: Down	
Valid values:	
-1= Use SP_ErrorLimitSignalName< n >	
0= Peak 1 of first Channel	
1= Peak 2 of first Channel	
2= Peak 3 of first Channel	
3= Peak 4 of first Channel	
4= Peak 5 of first Channel	
5= Peak 6 of first Channel	
6= Peak 7 of first Channel	
7= Peak 8 of first Channel	
8= Peak 9 of first Channel	
9= Peak 10 of first Channel	
10= Peak 11 of first Channel	
11= Peak 12 of first Channel	
12= Peak 13 of first Channel	
13= Peak 14 of first Channel	
14= Peak 15 of first Channel	
15= Peak 16 of first Channel	
Description: Signal index to be checked.	

Parameter: String SP_ErrorLimitSignalName<n>
Direction: Down
Description: User defined signal which is used for error limit.

 SP_ErrorLimitSignal-
 Name<n>

12.1.3.3.2 Get_ErrorLimitSignal<n> (ERRORLIMITSIGNAL<n>)

Get the signal for error limit n ∈ {1..2}.

Parameter: int32_t SA_ErrorLimitSignalIndex<n>

 SA_ErrorLimitSignalIn-
 dex<n>

Direction: Up

Valid values:

- 1= User defined name at SA_ErrorLimitSignalName<n>
- 0= Peak 1 of first Channel
- 1= Peak 2 of first Channel
- 2= Peak 3 of first Channel
- 3= Peak 4 of first Channel
- 4= Peak 5 of first Channel
- 5= Peak 6 of first Channel
- 6= Peak 7 of first Channel
- 7= Peak 8 of first Channel
- 8= Peak 9 of first Channel
- 9= Peak 10 of first Channel
- 10= Peak 11 of first Channel
- 11= Peak 12 of first Channel
- 12= Peak 13 of first Channel
- 13= Peak 14 of first Channel
- 14= Peak 15 of first Channel
- 15= Peak 16 of first Channel

Description: Signal index to be checked.

Parameter: String SA_ErrorLimitSignalName<n>

 SA_ErrorLimitSignal-
 Name<n>

Direction: Up

Description: User defined signal which is used for error limit.

12.1.3.3.3 Set_ErrorLimitValues<n> (ERRORLIMITVALUES<n>)

Set the error limits n ∈ {1..2}.

Parameter: double SP_LowerLimit<n>

SP_LowerLimit<n>

Direction: Down

Valid values:

- Minimum:** -21.47
Maximum: 21.47

Unit: mm

Description: Lower limit.

Parameter: double SP_UpperLimit<n>

SP_UpperLimit<n>

Direction: Down

Valid values:

- Minimum:** -21.47
Maximum: 21.47

Unit: mm

Description: Upper limit.

12.1.3.3.4 Get_ErrorLimitValues< n > (ERRORLIMITVALUES< n >)

Get the error limits $n \in \{1..2\}$.

Parameter: double SA_LowerLimit< n >

SA_LowerLimit< n >

Direction: Up

Valid values:

Minimum: -21.47

Maximum: 21.47

Unit: mm

Description: Lower limit.

Parameter: double SA_UpperLimit< n >

SA_UpperLimit< n >

Direction: Up

Valid values:

Minimum: -21.47

Maximum: 21.47

Unit: mm

Description: Upper limit.

12.1.3.3.5 Set_ErrorLimitCompOp< n > (ERRORLIMITCOMPARETO< n >)

Set the compare operation for limit $n \in \{1..2\}$.

Parameter: int32_t SP_ErrorLimitCompOp< n >

SP_ErrorLimitCompOp< n >

Direction: Down

Valid values:

0= Lower

1= Upper

2= Both

Description: Compare operation

12.1.3.3.6 Get_ErrorLimitCompOp< n > (ERRORLIMITCOMPARETO< n >)

Get the compare operation for limit $n \in \{1..2\}$.

Parameter: int32_t SA_ErrorLimitCompOp< n >

SA_ErrorLimitCompOp< n >

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Lower

1= Upper

2= Both

Description: Compare operation

12.1.3.3.7 Set_ErrorLevelOut<n> (ERRORLEVELOUT<n>)

Set level of error output n $\in \{1..2\}$ on error.

Parameter: int32_t SP_ErrorLevelOut<n>

SP_ErrorLevelOut<n>

Direction: Down

Valid values:

- 0= NPN
- 1= PNP
- 2= Push-Pull (PUSHPULL)
- 3= Push-Pull negated (PUSHPULLNEG)

Description: Error level.

12.1.3.3.8 Get_ErrorLevelOut<n> (ERRORLEVELOUT<n>)

Get level of error output n $\in \{1..2\}$ on error.

Parameter: int32_t SA_ErrorLevelOut<n>

SA_ErrorLevelOut<n>

Direction: Up

Valid values:

- 0= NPN
- 1= PNP
- 2= Push-Pull (PUSHPULL)
- 3= Push-Pull negated (PUSHPULLNEG)

Description: Error level.

12.1.3.4 Analog output

12.1.3.4.1 Set_AnalogOutput (ANALOGOUT)

Set the data to be used for analog output.

Parameter: int32_t SP_AnalogOutputSignalIndex

SP_AnalogOutputSignalIndex

Direction: Down

Valid values:

- 1= Use SP_AnalogOutputSignalName
- 0= Peak 1 of first Channel
- 1= Peak 2 of first Channel
- 2= Peak 3 of first Channel
- 3= Peak 4 of first Channel
- 4= Peak 5 of first Channel
- 5= Peak 6 of first Channel
- 6= Peak 7 of first Channel
- 7= Peak 8 of first Channel
- 8= Peak 9 of first Channel
- 9= Peak 10 of first Channel
- 10= Peak 11 of first Channel
- 11= Peak 12 of first Channel
- 12= Peak 13 of first Channel
- 13= Peak 14 of first Channel
- 14= Peak 15 of first Channel
- 15= Peak 16 of first Channel

Description: Data to be used for analog output.

Parameter: String SP_AnalogOutputSignalName SP_AnalogOutputSignalName
Direction: Down
Description: User defined signal which is used for analog output.

12.1.3.4.2 Get_AnalogOutput (ANALOGOUT)

Get the data to be used for analog output.

Parameter: int32_t SA_AnalogOutputSignalIndex SA_AnalogOutputSignalIndex
Direction: Up
Valid values:

- 1= User defined name at SA_AnalogOutputSignalName
- 0= Peak 1 of first Channel
- 1= Peak 2 of first Channel
- 2= Peak 3 of first Channel
- 3= Peak 4 of first Channel
- 4= Peak 5 of first Channel
- 5= Peak 6 of first Channel
- 6= Peak 7 of first Channel
- 7= Peak 8 of first Channel
- 8= Peak 9 of first Channel
- 9= Peak 10 of first Channel
- 10= Peak 11 of first Channel
- 11= Peak 12 of first Channel
- 12= Peak 13 of first Channel
- 13= Peak 14 of first Channel
- 14= Peak 15 of first Channel
- 15= Peak 16 of first Channel

Description: Data to be used for analog output.

Parameter: String SA_AnalogOutputSignalName SA_AnalogOutputSignalName
Direction: Up
Description: User defined signal which is used for analog output.

12.1.3.4.3 Set_AnalogRange (ANALOG RANGE)

Set the analog output range.

Parameter: int32_t SP_AnalogRange SP_AnalogRange
Direction: Down
Valid values:

- 1= 0 - 5V
- 2= 0 - 10V
- 5= 4 - 20mA

Description: Analog output range.

12.1.3.4.4 Get_AnalogRange (ANALOG RANGE)

Get the analog output range.

Parameter: int32_t SA_AnalogRange SA_AnalogRange
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 1 = 0 - 5V
 2 = 0 - 10V
 5 = 4 - 20mA
Description: Analog output range.

12.1.3.4.5 Set_AnalogScaleMode (ANALOG SCALE MODE)

Set the scaling mode for analog output.

Parameter: int32_t SP_AnalogScaleMode SP_AnalogScaleMode
Direction: Down
Valid values:
 0 = Standard
 1 = Two point (TWOPOINT)
Description: Analog scale mode.

12.1.3.4.6 Get_AnalogScaleMode (ANALOG SCALE MODE)

Get the scaling mode for analog output.

Parameter: int32_t SA_AnalogScaleMode SA_AnalogScaleMode
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = Standard
 1 = Two point (TWOPOINT)
Description: Analog scale mode.

12.1.3.4.7 Set_AnalogScaleRange (ANALOG SCALE RANGE)

Set the scaling factors for analog output.

Parameter: double SP_AnalogScaleRangeLowerLimit SP_AnalogScaleRangeLowerLimit
Direction: Down
Valid values:
Minimum: -21.47
Maximum: 21.47
Unit: mm
Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SP_AnalogScaleRangeUpperLimit

SP_AnalogScaleRangeUpper-Limit

Direction: Down

Valid values:

Minimum: -21.47

Maximum: 21.47

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

12.1.3.4.8 Get_AnalogScaleRange (ANALOGSCALERANGE)

Get the scaling factor for analog output.

Parameter: double SA_AnalogScaleRangeLowerLimit

SA_AnalogScaleRangeLower-Limit

Direction: Up

Valid values:

Minimum: -21.47

Maximum: 21.47

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SA_AnalogScaleRangeUpperLimit

SA_AnalogScaleRangeUpper-Limit

Direction: Up

Valid values:

Minimum: -21.47

Maximum: 21.47

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

13 Commands for optoCONTROL (ODC) sensors

13.1 Commands for ODC1202/20

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (native).
- [TCP/IP](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Version](#) and [Get_ParamSet2](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking and to scale data.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor (including two status bits, mask it out by raw&0xffff), from 0 to 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28) (without segment bits).
- Scaled values are scaled using sensor range, from 0 to 75.0 (ODC1202-75) or 100.0 (ODC1202-100) or 28.0 (ODC1220-28), error values are scaled depending of [IP_ScaleErrorValues](#).

13.1.1 General commands

13.1.1.1 General

13.1.1.1.1 Get_Version

Read sensor version and other information.

Parameter: String SA_Sensor SA_Sensor
Direction: Up
Description: Name of the sensor.

Parameter: double SA_Range SA_Range
Direction: Up
Unit: mm
Valid values:
 75
 100
 28
Description: Sensor range.

Parameter: String SA_Softwareversion SA_Softwareversion
Direction: Up
Description: Sensor firmware version.

Parameter: String SA_Date SA_Date
Direction: Up
Description: Sensor firmware date.

13.1.1.2 User level

13.1.1.2.1 Get_Echo

Checks if the sensor is responding.

Parameter: int32_t SA_EchoValue SA_EchoValue
Direction: Up
Valid values:
 0x00aa
Description: Echo value, returned from sensor.

13.1.1.3 Interfaces

13.1.1.3.1 Change_Baudrate

Change baudrate at sensor. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_RS232Baudrate SP_RS232Baudrate
Direction: Down
Valid values:
 0= 9600 Baud
 1= 19200 Baud
 2= 38400 Baud
 3= 57600 Baud
 4= 115200 Baud
Description: Baudrate of RS232 interface.

13.1.1.4 Parameter management

13.1.1.4.1 RefreshVideoThr

Refresh auto video threshold at sensor.

Parameter: int32_t SP_Location SP_Location
Direction: Down
Valid values:
 0= RAM
 1= EEPROM
Description: Specify if threshold should be stored in temporary (RAM) until sensor is switch off or permanently (EEPROM).

Parameter: int32_t SA_VideoThrAuto SA_VideoThrAuto
Direction: Up
Valid values:
Minimum: 0
Maximum: 100
Description: Video threshold (in percent) of full ADC range for automatic video threshold mode.

13.1.2 Measurement

13.1.2.1 Set_ParamSet1

Set first part of parameters.
 For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_Location SP_Location
Direction: Down
Valid values:
 0= RAM
 1= EEPROM
Description: Specify if parameters should be stored in temporary (RAM) until sensor is switch off or permanently (EEPROM).

Parameter: int32_t SP_Power SP_Power
Direction: Down
Valid values:
Minimum: 0
Maximum: 1000
Description: Laser intensity.

Parameter: int32_t SP_PowerMode SP_PowerMode
Direction: Down
Valid values:
 0= Static
 1= Dynamic
Description: Transmitter mode, currently not used.

Parameter: int32_t SP_Polarity SP_Polarity
Direction: Down
Valid values:
 0= Direct
 1= Inverse
Description: Polarity setting for digital outputs OUT0 to OUT3.

Parameter: int32_t SP_EvalMode SP_EvalMode
Direction: Down
Valid values:
 0= Left edge
 1= Right edge
 2= Width
 3= Center
Description: Evaluation mode for sensor.

Parameter: int32_t SP_EvalBegin SP_EvalBegin

Direction: Down

Valid values:

Minimum: 0

Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)

Description: Evaluation start pixel.

Parameter: int32_t SP_EvalEnd SP_EvalEnd

Direction: Down

Valid values:

Minimum: 0

Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)

Description: Evaluation end pixel. SP_EvalEnd must be greater as SP_EvalBegin.

Parameter: int32_t SP_TeachValue SP_TeachValue

Direction: Down

Valid values:

Minimum: 1

Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)

Description: Teach value.

Parameter: int32_t SP_ToleranceHigh SP_ToleranceHigh

Direction: Down

Valid values:

Minimum: 0

Maximum: 2360/2 (ODC1202-75) or 3072/2 (ODC1202-100) or 16384/2 (ODC1220-28)

Description: Upper tolerance value.

Parameter: int32_t SP_ToleranceLow SP_ToleranceLow

Direction: Down

Valid values:

Minimum: 0

Maximum: 2360/2 (ODC1202-75) or 3072/2 (ODC1202-100) or 16384/2 (ODC1220-28)

Description: Lower tolerance value.

Parameter: int32_t SP_Averaging SP_Averaging

Direction: Down

Valid values:

1

2

4

8

16

32

64

128

256

Description: Average setting for measurement value.

Parameter: int32_t SP_TriggerMode	SP_TriggerMode
Direction: Down	
Valid values:	
0= Continuous	
1= In0 rising edge	
2= In0 high level	
3= Save video threshold	
4= Output width	
5= Laser on/off	
Description: Trigger mode.	
Parameter: int32_t SP_AnalogOutput	SP_AnalogOutput
Direction: Down	
Valid values:	
0= Direct 0..10V	
1= Maxima	
2= Minima	
3= Max-Min	
Description: Analog output mode.	
Parameter: int32_t SP_OperationMode	SP_OperationMode
Direction: Down	
Valid values:	
0= ADC	
1= Comperator	
Description: CCD operation mode. It can be sampled using the ADC or only an comperator (1 bit).	
Parameter: int32_t SP_HWMode	SP_HWMode
Direction: Down	
Valid values:	
0= Disable all	
1= Enable all	
2= Enable button	
3= Enable poti	
Description: Enabe or disable tolerance poti and button at sensor.	
Parameter: int32_t SP_VideoThrMode	SP_VideoThrMode
Direction: Down	
Valid values:	
0= Fix	
1= Auto	
Description: Video threshold mode.	

13.1.2.2 Get_ParamSet1

Get first part of parameters.

Parameter: int32_t SP_Location	SP_Location
Direction: Down	
Valid values:	
0= RAM	
1= EEPROM	
Description: Specify if parameters should be read from RAM or EEPROM.	

Parameter: int32_t SA_Power	SA_Power
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1000	
Description: Laser intensity.	
Parameter: int32_t SA_PowerMode	SA_PowerMode
Direction: Up	
Valid values:	
0= Static	
1= Dynamic	
Description: Transmitter mode, currently not used.	
Parameter: int32_t SA_Polarity	SA_Polarity
Direction: Up	
Valid values:	
0= Direct	
1= Inverse	
Description: Polarity setting for digital outputs OUT0 to OUT3.	
Parameter: int32_t SA_EvalMode	SA_EvalMode
Direction: Up	
Valid values:	
0= Left edge	
1= Right edge	
2= Width	
3= Center	
Description: Evaluation mode for sensor.	
Parameter: int32_t SA_EvalBegin	SA_EvalBegin
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Evaluation start pixel.	
Parameter: int32_t SA_EvalEnd	SA_EvalEnd
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Evaluation end pixel.	
Parameter: int32_t SA_TeachValue	SA_TeachValue
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Teach value.	

Parameter: int32_t SA_ToleranceHigh	SA_ToleranceHigh
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360/2 (ODC1202-75) or 3072/2 (ODC1202-100) or 16384/2 (ODC1220-28)	
Description: Upper tolerance value.	
Parameter: int32_t SA_ToleranceLow	SA_ToleranceLow
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360/2 (ODC1202-75) or 3072/2 (ODC1202-100) or 16384/2 (ODC1220-28)	
Description: Lower tolerance value.	
Parameter: int32_t SA_Averaging	SA_Averaging
Direction: Up	
Valid values:	
1	
2	
4	
8	
16	
32	
64	
128	
256	
Description: Average setting for measurement value.	
Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
0= Continuous	
1= In0 rising edge	
2= In0 high level	
3= Save video threshold	
4= Output width	
5= Laser on/off	
Description: Trigger mode.	
Parameter: int32_t SA_AnalogOutput	SA_AnalogOutput
Direction: Up	
Valid values:	
0= Direct 0..10V	
1= Maxima	
2= Minima	
3= Max-Min	
Description: Analog output mode.	
Parameter: int32_t SA_OperationMode	SA_OperationMode
Direction: Up	
Valid values:	
0= ADC	
1= Comperator	
Description: CCD operation mode. It can be sampled using the ADC or only an comperator (1 bit).	

Parameter: int32_t SA_HWMode SA_HWMode

Direction: Up

Valid values:

- 0= Disable all
- 1= Enable all
- 2= Enable button
- 3= Enable poti

Description: Enable or disable tolerance poti and button at sensor.

Parameter: int32_t SA_VideoThrMode SA_VideoThrMode

Direction: Up

Valid values:

- 0= Fix
- 1= Auto

Description: Video threshold mode.

13.1.2.3 Set_ParamSet2

Set second part of parameters.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_Location SP_Location

Direction: Down

Valid values:

- 0= RAM
- 1= EEPROM

Description: Specify if parameters should be stored in temporary (RAM) until sensor is switch off or permanently (EEPROM).

Parameter: int32_t SP_VideoThrFix SP_VideoThrFix

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 100

Description: Video threshold (in percent) of full ADC range for fix video threshold mode.

Parameter: int32_t SP_VideoThrAuto SP_VideoThrAuto

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 100

Description: Video threshold (in percent) of full ADC range for automatic video threshold mode.

Parameter: int32_t SP_RS232Mode SP_RS232Mode

Direction: Down

Valid values:

- 0= Static (record)
- 1= In0 rising edge (record)
- 2= In0 rising edge (3 byte)
- 3= Continous (3 byte)

Description: Transfer mode at RS232 interface.

Parameter: int32_t SP_RS232Baudrate	SP_RS232Baudrate
Direction: Down	
Valid values:	
0= 9600 Baud	
1= 19200 Baud	
2= 38400 Baud	
3= 57600 Baud	
4= 115200 Baud	
Description: Baudrate of RS232 interface.	
Parameter: int32_t SP_SmoothVideo	SP_SmoothVideo
Direction: Down	
Valid values:	
1	
2	
4	
8	
12	
16	
24	
32	
48	
64	
Description: Smooth video signal.	
Parameter: int32_t SP_AnalogZoom	SP_AnalogZoom
Direction: Down	
Valid values:	
0= Direct	
1= Zoom x1	
2= Zoom x2	
3= Zoom x4	
4= Zoom x8	
5= Zoom x15	
6= Tolerance Window 5V	
7= Tolerance Window 10V	
Description: Zoom mode for analog output.	
Parameter: int32_t SP_InternalTriggerMode	SP_InternalTriggerMode
Direction: Down	
Valid values:	
0= Disable	
1= Enable	
Description: Internal object trigger.	
Parameter: int32_t SP_DarkPixel	SP_DarkPixel
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Number of dark pixels for object trigger.	

Parameter: int32_t SP_OutMode	SP_OutMode
Direction: Down	
Valid values:	
0= [-]OUT0, [+]OUT1, [OK]OUT2	
Description: Output mode.	
Parameter: int32_t SP_Parameter10	SP_Parameter10
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved for further use.	
Parameter: int32_t SP_Parameter11	SP_Parameter11
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved for further use.	
Parameter: double SP_SlopeValue	SP_SlopeValue
Direction: Down	
Unit: $\mu\text{m}/\text{pixel}$	
Valid values:	
Minimum: 1.0	
Maximum: 1000.0	
Description: Slope value for calibration.	
Parameter: double SP_RefOffset	SP_RefOffset
Direction: Down	
Unit: μm	
Valid values:	
Minimum: -32767.0	
Maximum: 32767.0	
Description: Intersection parameter for calibration.	

13.1.2.4 Get_ParamSet2

Get second part of parameters.

Parameter: int32_t SP_Location	SP_Location
Direction: Down	
Valid values:	
0= RAM	
1= EEPROM	
Description: Specify if parameters should be read from RAM or EEPROM.	
Parameter: int32_t SA_VideoThrFix	SA_VideoThrFix
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 100	
Description: Video threshold (in percent) of full ADC range for fix video threshold mode.	

Parameter: int32_t SA_VideoThrAuto SA_VideoThrAuto

Direction: Up

Valid values:

Minimum: 0

Maximum: 100

Description: Video threshold (in percent) of full ADC range for automatic video threshold mode.

Parameter: int32_t SA_RS232Mode SA_RS232Mode

Direction: Up

Valid values:

 0= Static (record)

 1= In0 rising edge (record)

 2= In0 rising edge (3 byte)

 3= Continous (3 byte)

Description: Transfer mode at RS232 interface.

Parameter: int32_t SA_RS232Baudrate SA_RS232Baudrate

Direction: Up

Valid values:

 0= 9600 Baud

 1= 19200 Baud

 2= 38400 Baud

 3= 57600 Baud

 4= 115200 Baud

Description: Baudrate of RS232 interface.

Parameter: int32_t SA_SmoothVideo SA_SmoothVideo

Direction: Up

Valid values:

 1

 2

 4

 8

 12

 16

 24

 32

 48

 64

Description: Smooth video signal.

Parameter: int32_t SA_AnalogZoom SA_AnalogZoom

Direction: Up

Valid values:

 0= Direct

 1= Zoom x1

 2= Zoom x2

 3= Zoom x4

 4= Zoom x8

 5= Zoom x15

 6= Tolerance Window 5V

 7= Tolerance Window 10V

Description: Zoom mode for analog output.

Parameter: int32_t SA_InternalTriggerMode	SA_InternalTriggerMode
Direction: Up	
Valid values:	
0= Disable	
1= Enable	
Description: Internal object trigger.	
Parameter: int32_t SA_DarkPixel	SA_DarkPixel
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Number of dark pixels for object trigger.	
Parameter: int32_t SA_OutMode	SA_OutMode
Direction: Up	
Valid values:	
0= [-]OUT0, [+]OUT1, [OK]OUT2	
Description: Output mode.	
Parameter: int32_t SA_Parameter10	SA_Parameter10
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved for further use.	
Parameter: int32_t SA_Parameter11	SA_Parameter11
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved for further use.	
Parameter: double SA_SlopeValue	SA_SlopeValue
Direction: Up	
Unit: $\mu\text{m}/\text{pixel}$	
Valid values:	
Minimum: 1.0	
Maximum: 1000.0	
Description: Slope value for calibration.	
Parameter: double SA_RefOffset	SA_RefOffset
Direction: Up	
Unit: μm	
Valid values:	
Minimum: -32767.0	
Maximum: 32767.0	
Description: Intersection parameter for calibration.	

13.1.2.5 Get_Video

Get video signal from sensor. When starting reading video signal (first call to Get_Video), the whole video signal is stored in a buffer at sensor and can be read completely by further calls to Get_Video (sample and hold). If another command is executed, the buffer is released.

Parameter: int32_t SP_BlockIndex SP_BlockIndex

Direction: Down

Valid values:

Minimum: -1

Maximum: 3

Description: Video signal cannot be transmitted at once, so it is devided into four blocks (0..3). If block index is -1, MEDAQLib automatically reads all four blocks and concatenates the video signal.

Parameter: Binary data SA_VideoSignal SA_VideoSignal

Direction: Up

Valid values:

Each block (0..3) contains 128 bytes, convertible to 64 words.

Description: Raw video signal.

13.1.2.6 Get_MeasValues

Get measured values from sensor.

Parameter: int32_t SA_LeftEdge SA_LeftEdge

Direction: Up

Valid values:

Minimum: 0

Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)

Description: Pixel position of left edge.

Parameter: int32_t SA_RightEdge SA_RightEdge

Direction: Up

Valid values:

Minimum: 0

Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)

Description: Pixel position of right edge.

Parameter: int32_t SA_MeasValue SA_MeasValue

Direction: Up

Valid values:

Minimum: 0

Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)

Description: Unscaled measurement value.

Parameter: double SA_ScaledValue	SA_ScaledValue
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 75.0 (ODC1202-75) or 100.0 (ODC1202-100) or 28.0 (ODC1220-28)	
Description: Scaled measurement value.	
Parameter: int32_t SA_TeachValue	SA_TeachValue
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Teach value.	
Parameter: int32_t SA_ToleranceValue	SA_ToleranceValue
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1024	
Description: Tolerance value of potentiometer.	
Parameter: int32_t SA_EdgeCount	SA_EdgeCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 15	
Description: Number of detected edges.	
Parameter: int32_t SA_MV-First-8	SA_MV-First-8
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 255	
Description: Mean value of first eight pixel after begin of evaluation in video signal.	
Parameter: int32_t SA_MV-Last-8	SA_MV-Last-8
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 255	
Description: Mean value of last eight pixel before end of evaluation in video signal.	
Parameter: int32_t SA_FreePara1	SA_FreePara1
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65534	
Description: Reserved for further use.	

Parameter: int32_t SA_MaxValue	SA_MaxValue
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Maximum value.	
Parameter: int32_t SA_MinValue	SA_MinValue
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Minimum value.	
Parameter: int32_t SA_InState	SA_InState
Direction: Up	
Valid values:	
Bit 0 shows digital input In0	
Bit 1 shows digital input In1	
Bit 2 shows button at sensor	
Description: Bit coded state of digital input and button at sensor.	
Parameter: int32_t SA_VideoMax	SA_VideoMax
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 255	
Description: Maximum value of video signal.	
Parameter: int32_t SA_DarkPixel	SA_DarkPixel
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Dark pixel.	

13.1.2.7 Get_Value_DataRecord

Get measured values from sensor (less as at Get_MeasValues).

Parameter: int32_t SA_MeasValue	SA_MeasValue
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Unscaled measurement value.	

13.1. Commands for ODC1202/20

Parameter: int32_t SA_LeftEdge	SA_LeftEdge
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Pixel position of left edge.	
Parameter: int32_t SA_RightEdge	SA_RightEdge
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Pixel position of right edge.	
Parameter: double SA_ScaledValue	SA_ScaledValue
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 75.0 (ODC1202-75) or 100.0 (ODC1202-100) or 28.0 (ODC1220-28)	
Description: Scaled measurement value.	
Parameter: int32_t SA_EdgeCount	SA_EdgeCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 15	
Description: Number of detected edges.	

13.1.3 Data output

13.1.3.1 General

13.1.3.1.1 AutoDataTransfer

Starts or stops auto data transfer at sensor.

Parameter: int32_t SP_Active	SP_Active
Direction: Down	
Valid values:	
0= Stop	
1= Start	
Description: This mode can be activated only if parameter SP_RS232Mode is set to 3 (Continous, 3 byte) at command Set_ParamSet2.	

13.1.3.2 Switching outputs

13.1.3.2.1 Activate_Teach

Activate teaching mode at sensor.

Parameter: int32_t SA_Power SA_Power

Direction: Up

Valid values:

Minimum: 0

Maximum: 1000

Description: Laser intensity.

Parameter: int32_t SA_PowerMode SA_PowerMode

Direction: Up

Valid values:

0= Static

1= Dynamic

Description: Transmitter mode, currently not used.

Parameter: int32_t SA_Polarity SA_Polarity

Direction: Up

Valid values:

0= Direct

1= Inverse

Description: Polarity setting for digital outputs OUT0 to OUT3.

Parameter: int32_t SA_EvalMode SA_EvalMode

Direction: Up

Valid values:

0= Left edge

1= Right edge

2= Width

3= Center

Description: Evaluation mode for sensor.

Parameter: int32_t SA_EvalBegin SA_EvalBegin

Direction: Up

Valid values:

Minimum: 0

Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)

Description: Evaluation start pixel.

Parameter: int32_t SA_EvalEnd SA_EvalEnd

Direction: Up

Valid values:

Minimum: 0

Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)

Description: Evaluation end pixel.

Parameter: int32_t SA_TeachValue	SA_TeachValue
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2360 (ODC1202-75) or 3072 (ODC1202-100) or 16384 (ODC1220-28)	
Description: Teach value.	
Parameter: int32_t SA_ToleranceHigh	SA_ToleranceHigh
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360/2 (ODC1202-75) or 3072/2 (ODC1202-100) or 16384/2 (ODC1220-28)	
Description: Upper tolerance value.	
Parameter: int32_t SA_ToleranceLow	SA_ToleranceLow
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2360/2 (ODC1202-75) or 3072/2 (ODC1202-100) or 16384/2 (ODC1220-28)	
Description: Lower tolerance value.	
Parameter: int32_t SA_Averaging	SA_Averaging
Direction: Up	
Valid values:	
1	
2	
4	
8	
16	
32	
64	
128	
256	
Description: Average setting for measurement value.	
Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
0= Continuous	
1= In0 rising edge	
2= In0 high level	
3= Save video threshold	
4= Output width	
5= Laser on/off	
Description: Trigger mode.	
Parameter: int32_t SA_AnalogOutput	SA_AnalogOutput
Direction: Up	
Valid values:	
0= Direct 0..10V	
1= Maxima	
2= Minima	
3= Max-Min	
Description: Analog output mode.	

Parameter: int32_t SA_OperationMode	SA_OperationMode
Direction: Up	
Valid values:	
0= ADC	
1= Comperator	
Description:	CCD operation mode. It can be sampled using the ADC or only an comperator (1 bit).
Parameter: int32_t SA_HWMode	SA_HWMode
Direction: Up	
Valid values:	
0= Disable all	
1= Enable all	
2= Enable button	
3= Enable poti	
Description:	Enable or disable tolerance poti and button at sensor.
Parameter: int32_t SA_VideoThrMode	SA_VideoThrMode
Direction: Up	
Valid values:	
0= Fix	
1= Auto	
Description:	Video threshold mode.

13.1.3.3 Analog output

13.1.3.3.1 Reset_MinMax_Analog

Reset maximum/minimum values at analog output.

13.2 Commands for ODC2500

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (native).
- [IF2004](#) (native).
- [TCP/IP](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Info](#) and [Read_OptionData](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate, to interpret data and to assign values.

If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls sensor command [Dat_Out_On](#) at [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor (including two segment bits, mask it out by raw&0xffff), from 0 to 65535 (without segment bits).
- Scaled values are scaled using sensor range, error values are scaled depending of [IP_ScaleErrorValues](#).

The values of selected segments are filled in the arrays one after another. Each array always starts with first selected segment.

13.2.1 General commands

13.2.1.1 General

13.2.1.1.1 Get_Info (INFO)

Retrieve some information about the sensor.

Parameter: String SA_SerialNumber SA_SerialNumber

Direction: Up

Valid values:

Numeric value

Description: Serial number of the sensor.

Parameter: String SA_Option SA_Option

Direction: Up

Valid values:

Numeric value

Description: Option of the sensor.

Parameter: double SA_Range SA_Range

Direction: Up

Unit: mm

Valid values:

34.0

Description: Range of the sensor.

Parameter: int32_t SA_Reserve_1 SA_Reserve_1

Direction: Up

Valid values:

0

Description: Reserved.

Parameter: String SA_SoftArtBoot	SA_SoftArtBoot
Direction: Up	
Description: Sensor software versions.	
Parameter: String SA_SoftArtArm	SA_SoftArtArm
Direction: Up	
Description: Sensor software versions.	
Parameter: String SA_SoftArtDSP	SA_SoftArtDSP
Direction: Up	
Description: Sensor software versions.	
Parameter: int32_t SA_SoftVerBoot	SA_SoftVerBoot
Direction: Up	
Valid values:	
Numeric value	
Description: Sensor software versions.	
Parameter: int32_t SA_SoftVerArm	SA_SoftVerArm
Direction: Up	
Valid values:	
Numeric value	
Description: Sensor software versions.	
Parameter: int32_t SA_SoftVerDSP	SA_SoftVerDSP
Direction: Up	
Valid values:	
Numeric value	
Description: Sensor software versions.	

13.2.1.1.2 Reset_Boot (RESET)

Resets the sensor.

13.2.1.2 Parameter management

13.2.1.2.1 Save_OptionData (SAVE_OPT_RAM_TO_FLASH)

Save the option data to flash.

13.2.1.2.2 Save_MeasProgData (SAVE_MPR_RAM_TO_FLASH)

Save the measure program data to flash.

13.2.2 Measurement

13.2.2.1 Choose_MeasProg (CHOOSE_MP)

Select the measurement program of sensor.

Parameter: int32_t SP_MeasProgNumber

SP_MeasProgNumber

Direction: Down

Valid values:

- 0= EdgeHL
- 1= EdgeLH
- 2= DIA
- 3= GAP
- 4= SEG_2_4
- 5= 2-SEG
- 6= USER1
- 7= USER2
- 8= USER3
- 9= USER4

Description: Measure program of sensor.

13.2.2.2 Switch_Edge (SWITCH_EDGE)

Select the edges to measure.

Parameter: int32_t SP_FrontEdge_Seg1

SP_FrontEdge_Seg1

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 32

Description: Front edge of segment 1.

Parameter: int32_t SP_FrontEdge_Seg2

SP_FrontEdge_Seg2

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 32

Description: Front edge of segment 2.

Parameter: int32_t SP_RearEdge_Seg1

SP_RearEdge_Seg1

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 32

Description: Rear edge of segment 1.

Parameter: int32_t SP_RearEdge_Seg2

SP_RearEdge_Seg2

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 32

Description: Rear edge of segment 2.

13.2.2.3 Write_OptionData (WR_OPT_TO_RAM)

Write the option data to sensor.

Parameter: int32_t SP_MeasProgNumber

SP_MeasProgNumber

Direction: Down

Valid values:

- 0= EdgeHL
- 1= EdgeLH
- 2= DIA
- 3= GAP
- 4= SEG_2_4
- 5= 2-SEG
- 6= USER1
- 7= USER2
- 8= USER3
- 9= USER4

Description: Measure program of sensor.

Parameter: int32_t SP_Language

SP_Language

Direction: Down

Valid values:

- 0= German
- 1= English

Description: Language of sensor.

Parameter: int32_t SP_DispmesUnit

SP_DispmesUnit

Direction: Down

Valid values:

- 0= mm
- 1= inch

Description: Display measurement unit of sensor.

Parameter: int32_t SP_ErrorHandler

SP_ErrorHandler

Direction: Down

Valid values:

- 0= error output
- 1= retain last value

Description: If the sensor cannot measure values, it can output the last valid value or it can output an error values.

Parameter: int32_t SP_Reserve_2

SP_Reserve_2

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 65535

Description: Reserved.

Parameter: int32_t SP_Ext_LaserSwitch

SP_Ext_LaserSwitch

Direction: Down

Valid values:

- 0= not active
- 1= active

Description: Enable/Disable external laser switch.

Parameter: int32_t SP_LaserIntensity	SP_LaserIntensity
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: No effect.	
Parameter: int32_t SP_Contrast	SP_Contrast
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: Display contrast.	
Parameter: int32_t SP_Reserve_3	SP_Reserve_3
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Reserved.	
Parameter: int32_t SP_ActiveSerialIf	SP_ActiveSerialIf
Direction: Down	
Valid values:	
0 = RS422	
1 = RS232	
Description: Active serial interface of sensor.	
Parameter: int32_t SP_RS232_Baudrate	SP_RS232_Baudrate
Direction: Down	
Valid values:	
9600	
19200	
38400	
115200	
Unit: Baud	
Description: Baudrate of RS232 interface of sensor.	
Parameter: int32_t SP_RS232_Parity	SP_RS232_Parity
Direction: Down	
Valid values:	
0 = none	
1 = even	
2 = odd	
Description: Parity of RS232 interface of sensor.	
Parameter: int32_t SP_RS232_StopBits	SP_RS232_StopBits
Direction: Down	
Valid values:	
1	
2	
Description: Stop bits of RS232 interface of sensor.	

Parameter: int32_t SP_RS232_TimeoutSend	SP_RS232_TimeoutSend
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SP_RS232_TimeoutRecv	SP_RS232_TimeoutRecv
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SP_RS422_Baudrate	SP_RS422_Baudrate
Direction: Down	
Valid values:	
9600	
19200	
38400	
115200	
691200	
Unit: Baud	
Description: Baudrate of RS422 interface of sensor.	
Parameter: int32_t SP_RS422_Parity	SP_RS422_Parity
Direction: Down	
Valid values:	
0= none	
1= even	
2= odd	
Description: Parity of RS422 interface of sensor.	
Parameter: int32_t SP_RS422_StopBits	SP_RS422_StopBits
Direction: Down	
Valid values:	
1	
2	
Description: Stop bits of RS422 interface of sensor.	
Parameter: int32_t SP_RS422_TimeoutSend	SP_RS422_TimeoutSend
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SP_RS422_TimeoutRecv	SP_RS422_TimeoutRecv
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	

13.2.2.4 Read_OptionData (RD_OPT_RAM)

Read the option data from sensor.

Parameter: int32_t SA_MeasProgNumber

SA_MeasProgNumber

Direction: Up

Valid values:

- 0= EdgeHL
- 1= EdgeLH
- 2= DIA
- 3= GAP
- 4= SEG_2_4
- 5= 2-SEG
- 6= USER1
- 7= USER2
- 8= USER3
- 9= USER4

Description: Measure program of sensor.

Parameter: int32_t SA_Language

SA_Language

Direction: Up

Valid values:

- 0= German
- 1= English

Description: Language of sensor.

Parameter: int32_t SA_DispmesUnit

SA_DispmesUnit

Direction: Up

Valid values:

- 0= mm
- 1= inch

Description: Display measurement unit of sensor.

Parameter: int32_t SA_ErrorHandler

SA_ErrorHandler

Direction: Up

Valid values:

- 0= error output
- 1= retain last value

Description: If the sensor cannot measure values, it can output the last valid value or it can output an error values.

Parameter: int32_t SA_Reserve_2

SA_Reserve_2

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 65535

Description: Reserved.

Parameter: int32_t SA_Ext_LaserSwitch

SA_Ext_LaserSwitch

Direction: Up

Valid values:

- 0= not active
- 1= active

Description: Enable/Disable external laser switch.

Parameter: int32_t SA_LaserIntensity	SA_LaserIntensity
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: No effect.	
Parameter: int32_t SA_Contrast	SA_Contrast
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: Display contrast.	
Parameter: int32_t SA_Reserve_3	SA_Reserve_3
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Reserved.	
Parameter: int32_t SA_ActiveSerialIf	SA_ActiveSerialIf
Direction: Up	
Valid values:	
0 = RS422	
1 = RS232	
Description: Active serial interface of sensor.	
Parameter: int32_t SA_RS232_Baudrate	SA_RS232_Baudrate
Direction: Up	
Valid values:	
9600	
19200	
38400	
115200	
Unit: Baud	
Description: Baudrate of RS232 interface of sensor.	
Parameter: int32_t SA_RS232_Parity	SA_RS232_Parity
Direction: Up	
Valid values:	
0 = none	
1 = even	
2 = odd	
Description: Parity of RS232 interface of sensor.	
Parameter: int32_t SA_RS232_StopBits	SA_RS232_StopBits
Direction: Up	
Valid values:	
1	
2	
Description: Stop bits of RS232 interface of sensor.	

Parameter: int32_t SA_RS232_TimeoutSend	SA_RS232_TimeoutSend
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SA_RS232_TimeoutRecv	SA_RS232_TimeoutRecv
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SA_RS422_Baudrate	SA_RS422_Baudrate
Direction: Up	
Valid values:	
9600	
19200	
38400	
115200	
691200	
Unit: Baud	
Description: Baudrate of RS422 interface of sensor.	
Parameter: int32_t SA_RS422_Parity	SA_RS422_Parity
Direction: Up	
Valid values:	
0= none	
1= even	
2= odd	
Description: Parity of RS422 interface of sensor.	
Parameter: int32_t SA_RS422_StopBits	SA_RS422_StopBits
Direction: Up	
Valid values:	
1	
2	
Description: Stop bits of RS422 interface of sensor.	
Parameter: int32_t SA_RS422_TimeoutSend	SA_RS422_TimeoutSend
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SA_RS422_TimeoutRecv	SA_RS422_TimeoutRecv
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	

13.2.2.5 Update_OptionData

Update option data at sensor.

This is a meta command which first calls [Read_OptionData](#), renames all answer parameters (SA_...) to command parameters (SP_...), overwrites the parameter set by the parameters from user and finally calls [Write_OptionData](#).

The parameters of this command are NOT obligatory. Missing parameters are not overwritten, so the sensor keeps the original values.

Parameter: int32_t SP_MeasProgNumber

SP_MeasProgNumber

Direction: Down

Valid values:

- 0= EdgeHL
- 1= EdgeLH
- 2= DIA
- 3= GAP
- 4= SEG_2_4
- 5= 2-SEG
- 6= USER1
- 7= USER2
- 8= USER3
- 9= USER4

Description: Measure program of sensor.

Parameter: int32_t SP_Language

SP_Language

Direction: Down

Valid values:

- 0= German
- 1= English

Description: Language of sensor.

Parameter: int32_t SP_DispmesUnit

SP_DispmesUnit

Direction: Down

Valid values:

- 0= mm
- 1= inch

Description: Display measurement unit of sensor.

Parameter: int32_t SP_ErrorHandler

SP_ErrorHandler

Direction: Down

Valid values:

- 0= error output
- 1= retain last value

Description: If the sensor cannot measure values, it can output the last valid value or it can output an error values.

Parameter: int32_t SP_Reserve_2

SP_Reserve_2

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 65535

Description: Reserved.

Parameter: int32_t SP_Ext_LaserSwitch	SP_Ext_LaserSwitch
Direction: Down	
Valid values:	
0= not active	
1= active	
Description: Enable/Disable external laser switch.	
Parameter: int32_t SP_LaserIntensity	SP_LaserIntensity
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: No effect.	
Parameter: int32_t SP_Contrast	SP_Contrast
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: Display contrast.	
Parameter: int32_t SP_Reserve_3	SP_Reserve_3
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Reserved.	
Parameter: int32_t SP_ActiveSerialIf	SP_ActiveSerialIf
Direction: Down	
Valid values:	
0= RS422	
1= RS232	
Description: Active serial interface of sensor.	
Parameter: int32_t SP_RS232_Baudrate	SP_RS232_Baudrate
Direction: Down	
Valid values:	
9600	
19200	
38400	
115200	
Unit: Baud	
Description: Baudrate of RS232 interface of sensor.	
Parameter: int32_t SP_RS232_Parity	SP_RS232_Parity
Direction: Down	
Valid values:	
0= none	
1= even	
2= odd	
Description: Parity of RS232 interface of sensor.	

Parameter: int32_t SP_RS232_StopBits	SP_RS232_StopBits
Direction: Down	
Valid values:	
1	
2	
Description:	Stop bits of RS232 interface of sensor.
Parameter: int32_t SP_RS232_TimeoutSend	SP_RS232_TimeoutSend
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description:	No effect.
Parameter: int32_t SP_RS232_TimeoutRecv	SP_RS232_TimeoutRecv
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description:	No effect.
Parameter: int32_t SP_RS422_Baudrate	SP_RS422_Baudrate
Direction: Down	
Valid values:	
9600	
19200	
38400	
115200	
691200	
Unit: Baud	
Description:	Baudrate of RS422 interface of sensor.
Parameter: int32_t SP_RS422_Parity	SP_RS422_Parity
Direction: Down	
Valid values:	
0= none	
1= even	
2= odd	
Description:	Parity of RS422 interface of sensor.
Parameter: int32_t SP_RS422_StopBits	SP_RS422_StopBits
Direction: Down	
Valid values:	
1	
2	
Description:	Stop bits of RS422 interface of sensor.
Parameter: int32_t SP_RS422_TimeoutSend	SP_RS422_TimeoutSend
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description:	No effect.

Parameter: int32_t SP_RS422_TimeoutRecv SP_RS422_TimeoutRecv
Direction: Down
Valid values:
Minimum: 0
Maximum: 65535
Description: No effect.

13.2.2.6 Write_MeasProgData (WR_MPR_TO_RAM)

Write the measurement program data to sensor.

Parameter: int32_t SP_UserMeasProgNumber SP_UserMeasProgNumber
Direction: Down
Valid values:
 6= USER1
 7= USER2
 8= USER3
 9= USER4
Description: User measure program of sensor.

Parameter: String SP_MeasProgName SP_MeasProgName
Direction: Down
Description: Measure program name of sensor.

Parameter: double SP_AnalogOffset SP_AnalogOffset
Direction: Down
Valid values:
Minimum: -10.0
Maximum: 10.0
Unit: V
Description: Analog output offset of sensor.

Parameter: double SP_AnalogGain SP_AnalogGain
Direction: Down
Valid values:
Minimum: -3.4
Maximum: 3.4
Description: Analog output gain of sensor.

Parameter: double SP_DisplayOffset SP_DisplayOffset
Direction: Down
Valid values:
Minimum: -99.99
Maximum: 99.99
Unit: mm
Description: Display offset of sensor.

Parameter: double SP_DisplayGain SP_DisplayGain
Direction: Down
Valid values:
Minimum: -2.0
Maximum: 2.0
Description: Display gain of sensor.

Parameter: double SP_UpperLimit	SP_UpperLimit
Direction: Down	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Upper limit of sensor.	
Parameter: double SP_LowerLimit	SP_LowerLimit
Direction: Down	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Lower limit of sensor.	
Parameter: double SP_UpperWarning	SP_UpperWarning
Direction: Down	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Upper warning of sensor.	
Parameter: double SP_LowerWarning	SP_LowerWarning
Direction: Down	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Lower warning of sensor.	
Parameter: int32_t SP_Target_Distance	SP_Target_Distance
Direction: Down	
Valid values:	
0= 20 mm	
1= 50 mm	
2= 100 mm	
3= 150 mm	
Description: Target distance of sensor.	
Parameter: int32_t SP_Average_for_reading	SP_Average_for_reading
Direction: Down	
Valid values:	
1 to 128 sliding	
129 to 4096 recursive	
Description: Averaging mode and number of sensor.	
Parameter: int32_t SP_Reserve_4	SP_Reserve_4
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Reserved.	

Parameter: int32_t SP_MeasObject	SP_MeasObject
Direction: Down	
Valid values:	
1= EdgeHL	
2= EdgeLH	
3= DIA	
4= GAP	
5= SEG_2_4	
6= 2-SEG	
Description: Measurement program.	
Parameter: int32_t SP_FrontEdge_Seg1	SP_FrontEdge_Seg1
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 32	
Description: Front edge of segment 1.	
Parameter: int32_t SP_FrontEdge_Seg2	SP_FrontEdge_Seg2
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 32	
Description: Front edge of segment 2.	
Parameter: int32_t SP_RearEdge_Seg1	SP_RearEdge_Seg1
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 32	
Description: Rear edge of segment 1.	
Parameter: int32_t SP_RearEdge_Seg2	SP_RearEdge_Seg2
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 32	
Description: Rear edge of segment 2.	
Parameter: double SP_MasterValue	SP_MasterValue
Direction: Down	
Valid values:	
Minimum: -34.0	
Maximum: 34.0	
Unit: mm	
Description: Master value of sensor.	

13.2.2.7 Read_MeasProgData (RD_MPR_RAM)

Read the measurement program data from sensor.

Parameter: int32_t SA_MeasProgNumber

SA_MeasProgNumber

Direction: Up

Valid values:

- 0= EdgeHL
- 1= EdgeLH
- 2= DIA
- 3= GAP
- 4= SEG_2_4
- 5= 2-SEG
- 6= USER1
- 7= USER2
- 8= USER3
- 9= USER4

Description: User measure program of sensor.

Parameter: String SA_MeasProgName

SA_MeasProgName

Direction: Up

Description: Measure program name of sensor.

Parameter: double SA_AnalogOffset

SA_AnalogOffset

Direction: Up

Valid values:

- Minimum:** -10.0
- Maximum:** 10.0

Unit: V

Description: Analog output offset of sensor.

Parameter: double SA_AnalogGain

SA_AnalogGain

Direction: Up

Valid values:

- Minimum:** -3.4
- Maximum:** 3.4

Description: Analog output gain of sensor.

Parameter: double SA_DisplayOffset

SA_DisplayOffset

Direction: Up

Valid values:

- Minimum:** -99.99
- Maximum:** 99.99

Unit: mm

Description: Display offset of sensor.

Parameter: double SA_DisplayGain

SA_DisplayGain

Direction: Up

Valid values:

- Minimum:** -2.0
- Maximum:** 2.0

Description: Display gain of sensor.

Parameter: double SA_UpperLimit	SA_UpperLimit
Direction: Up	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Upper limit of sensor.	
Parameter: double SA_LowerLimit	SA_LowerLimit
Direction: Up	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Lower limit of sensor.	
Parameter: double SA_UpperWarning	SA_UpperWarning
Direction: Up	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Upper warning of sensor.	
Parameter: double SA_LowerWarning	SA_LowerWarning
Direction: Up	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Lower warning of sensor.	
Parameter: int32_t SA_Target_Distance	SA_Target_Distance
Direction: Up	
Valid values:	
0= 20 mm	
1= 50 mm	
2= 100 mm	
3= 150 mm	
Description: Target distance of sensor.	
Parameter: int32_t SA_Average_for_reading	SA_Average_for_reading
Direction: Up	
Valid values:	
1 to 128 sliding	
129 to 4096 recursive	
Description: Averaging mode and number of sensor.	
Parameter: int32_t SA_Reserve_4	SA_Reserve_4
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Reserved.	

Parameter: int32_t SA_MeasObject	SA_MeasObject
Direction: Up	
Valid values:	
1= EdgeHL	
2= EdgeLH	
3= DIA	
4= GAP	
5= SEG_2_4	
6= 2-SEG	
Description: Measurement program.	
Parameter: int32_t SA_FrontEdge_Seg1	SA_FrontEdge_Seg1
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 32	
Description: Front edge of segment 1.	
Parameter: int32_t SA_FrontEdge_Seg2	SA_FrontEdge_Seg2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 32	
Description: Front edge of segment 2.	
Parameter: int32_t SA_RearEdge_Seg1	SA_RearEdge_Seg1
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 32	
Description: Rear edge of segment 1.	
Parameter: int32_t SA_RearEdge_Seg2	SA_RearEdge_Seg2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 32	
Description: Rear edge of segment 2.	
Parameter: double SA_MasterValue	SA_MasterValue
Direction: Up	
Valid values:	
Minimum: -34.0	
Maximum: 34.0	
Unit: mm	
Description: Master value of sensor.	

13.2.2.8 Read_MinMax (RD_MINMAX)

Read the minimum and maximum values from sensor.

Parameter: int32_t SA_MinRaw	SA_MinRaw
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Minimum raw value.	

Parameter: int32_t SA_MaxRaw	SA_MaxRaw
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Maximum raw value.	
Parameter: double SA_MinScaled	SA_MinScaled
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 34.0	
Unit: mm	
Description: Minimum scaled value.	
Parameter: double SA_MaxScaled	SA_MaxScaled
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 34.0	
Unit: mm	
Description: Maximum scaled value.	

13.2.2.9 Read_MinMaxReset (RD_MINMAX_RESET)

Read the minimum and maximum values from sensor and reset the values.

Parameter: int32_t SA_MinRaw	SA_MinRaw
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Minimum raw value.	
Parameter: int32_t SA_MaxRaw	SA_MaxRaw
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Maximum raw value.	
Parameter: double SA_MinScaled	SA_MinScaled
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 34.0	
Unit: mm	
Description: Minimum scaled value.	
Parameter: double SA_MaxScaled	SA_MaxScaled
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 34.0	
Unit: mm	
Description: Maximum scaled value.	

13.2.3 Data output

13.2.3.1 General

13.2.3.1.1 Dat_Out_Off (STOP)

Switch off data output from sensor.

13.2.3.1.2 Dat_Out_On (START)

Switch on data output from sensor.

13.3 Commands for ODC2520

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
- [IF2004](#) (native).
- [TCP/IP](#) (native).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_AllParameters](#) (SP_Additional= 1) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate, to interpret and scale data and to assign values. If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls optionally sensor command [Set_IPDataTransferMode](#) and [Set_DataOutInterface](#) at [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_AllParameters](#) (SP_Additional= 1)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

13.3.1 General commands

13.3.1.1 General

13.3.1.1.1 Get_Help (HELP)

Retrieve a help text from controller for a specific command.

Parameter: String SP_Command

SP_Command

Direction: Down

Valid values:

"" (empty string, means general help)
or any command name

Description: Name of the command.

Parameter: String SA_HelpText

SA_HelpText

Direction: Up

Description: Help text to the command.

13.3.1.1.2 Get_Info (GETINFO)

Retrieve information about the controller.

Parameter: String SA_Sensor

SA_Sensor

Direction: Up

Description: Name of the controller.

Parameter: String SA_SerialNumber

SA_SerialNumber

Direction: Up

Valid values:

Numeric value

Description: Serial number of the controller.

Parameter: String SA_Option

SA_Option

Direction: Up

Valid values:

Numeric value

Description: Option of the controller.

Parameter: String SA_ArticleNumber

SA_ArticleNumber

Direction: Up

Valid values:

Numeric value

Description: Article number of the controller.

Parameter: double SA_Range

SA_Range

Direction: Up

Valid values:

46.0

95.0

Unit: mm

Description: Range of the sensor (from firmware 005.024.035).

13.3. Commands for ODC2520

Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description: MAC address (low level ethernet address) of the controller.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the controller.	
Parameter: String SA_Image	SA_Image
Direction: Up	
Description: Active software, ethernet or ethercat.	
Parameter: String SA_ImageType	SA_ImageType
Direction: Up	
Description: Firmware image type used by the controller.	
Parameter: String SA_HardwareRevision	SA_HardwareRevision
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of the controller board.	

13.3.1.1.3 Set_Echo (ECHO)

Set echo for sensor commands.

Parameter: int32_t SP_Echo	SP_Echo
Direction: Down	
Valid values:	
0= Off	
1= On	
Description: Echo mode.	

13.3.1.1.4 Get_Echo (ECHO)

Get the echo mode.

Parameter: int32_t SA_Echo	SA_Echo
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Description: Echo mode.	

13.3.1.1.5 Get_AllParameters (PRINT)

Get all parameters from controller.

Parameter: int32_t SP_Additional SP_Additional

Direction: Down

Valid values:

0= No

1= Yes

Description: If set, additional information about controller, sensor and material is output.

Parameter: int32_t SA_SyncMode SA_SyncMode

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= None

1= Slave

2= Master

Description: Synchronization mode.

Parameter: int32_t SA_SyncTermination SA_SyncTermination

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Off (TERMOFF)

1= On 120 Ohm (TERMON)

Description: Termination of external input.

Parameter: int32_t SA_UserLevel SA_UserLevel

Direction: Up

Valid values:

-1= Other user level (only for internal use)

0= User

1= Professional

Description: Actual user level.

Parameter: int32_t SA_DefaultUser SA_DefaultUser

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= User

1= Professional

Description: Default user level.

Parameter: int32_t SA_Echo SA_Echo

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Off

1= On

Description: Echo mode.

Parameter: int32_t SA_DisplayUnit	SA_DisplayUnit
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= mm	
1= Inch	
Description: Unit.	
Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Edge	
2= Level (PULSE)	
3= Software	
Description: Trigger mode.	
Parameter: int32_t SA_TriggerTermination	SA_TriggerTermination
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off (TERMOFF)	
1= On 120 Ohm (TERMON)	
Description: Termination of external input.	
Parameter: int32_t SA_TriggerMoment	SA_TriggerMoment
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Input	
1= Output	
Description: Trigger moment.	
Parameter: int32_t SA_TriggerLevel	SA_TriggerLevel
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= High	
1= Low	
Description: Trigger level.	
Parameter: int32_t SA_TriggerCount	SA_TriggerCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 16383	
Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.	
Parameter: int32_t SA_EthernetMode	SA_EthernetMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Ethernet	
1= Ethercat	
Description: Ethernet mode.	

Parameter: int32_t SA_DHCPEnabled	SA_DHCPEnabled
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = FALSE	
1 = TRUE	
Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).	
Parameter: String SA_Address	SA_Address
Direction: Up	
Valid values:	
Valid IP address in form of xxx.xxx.xxx.xxx	
Description: IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.	
Parameter: String SA_SubnetMask	SA_SubnetMask
Direction: Up	
Valid values:	
Valid network mask (e.g. 255.255.255.0 for a Class C network)	
Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.	
Parameter: String SA_Gateway	SA_Gateway
Direction: Up	
Valid values:	
Valid IP address of default gateway in form of xxx.xxx.xxx.xxx	
Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.	
Parameter: int32_t SA_Protocol	SA_Protocol
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = TCP server (SERVER/TCP)	
1 = TCP client (CLIENT/TCP)	
2 = UDP sender (CLIENT/UDP)	
3 = None	
Description: Specifies if data should be send using TCP or UDP.	
Parameter: String SA_RemoteAddress	SA_RemoteAddress
Direction: Up	
Valid values:	
Valid IP address of receiver of data	
Description: Address of remote computer to send data to.	
Parameter: int32_t SA_Port	SA_Port
Direction: Up	
Valid values:	
Minimum: 1024	
Maximum: 65535	
Description: Port to send data to or to listen for incoming requests.	

Parameter: int32_t SA_SensorBaudrate	SA_SensorBaudrate
Direction: Up	
Valid values:	
4000000	
3500000	
3000000	
2500000	
2000000	
1500000	
921600	
691200	
460800	
230400	
115200	
9600	
Unit: Baud	
Description: Baudrate of controller.	
Parameter: int32_t SA_MeasureMode	SA_MeasureMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Edge Light-Dark (EDGEHL)	
1 = Edge Dark-Light (EDGELH)	
2 = Diameter (DIA)	
3 = Gap	
4 = Segments (SEGMENT)	
Description: Measure mode.	
Parameter: int32_t SA_SearchDirection	SA_SearchDirection
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Standard	
1 = Inverse	
Description: Search direction for edges.	
Parameter: int32_t SA_MeasureDirection	SA_MeasureDirection
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Standard	
1 = Inverse	
Description: Reference point.	
Parameter: int32_t SA_ExpectedEdges	SA_ExpectedEdges
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 64	
Description: Number of expected edges.	

13.3. Commands for ODC2520

Parameter: int32_t SA_Edge1..8_A	SA_Edge1..8_A
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 64	
Description: First edge.	
Parameter: int32_t SA_Edge1..8_B	SA_Edge1..8_B
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 64	
Description: Second edge.	
Parameter: double SA_Threshold	SA_Threshold
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 99.0	
Unit: %	
Description: Video threshold.	
Parameter: int32_t SA_LaserPower	SA_LaserPower
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Off	
1= On	
Description: Laser power.	
Parameter: int32_t SA_ROIStart	SA_ROIStart
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 767 (for 46 mm range), 1535 (for 95 mm range)	
Unit: Pixel	
Description: First position on CCD.	
Parameter: int32_t SA_ROIEnd	SA_ROIEnd
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 767 (for 46 mm range), 1535 (for 95 mm range)	
Unit: Pixel	
Description: Last position on CCD.	
Parameter: int32_t SA_AveragingType	SA_AveragingType
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Moving average (MOVING)	
2= Recursive averaging (RECURSIVE)	
3= Median	
Description: Averaging type.	

13.3. Commands for ODC2520

Parameter: int32_t SA_MovingCount SA_MovingCount

Direction: Up

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128

Description: Number of value for the averaging window. This parameter is only available at moving average.

Parameter: int32_t SA_RecursiveCount SA_RecursiveCount

Direction: Up

Valid values:

- Minimum:** 2
- Maximum:** 32768

Description: Number of values for recursive averaging. This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount SA_MedianCount

Direction: Up

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only available at median.

Parameter: int32_t SA_SpikeCorrection SA_SpikeCorrection

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = off
- 1 = on

Description: Spike correction.

Parameter: int32_t SA_NbrEvaluatedValues SA_NbrEvaluatedValues

Direction: Up

Valid values:

- Minimum:** 1
- Maximum:** 10

Description: Number of values to evaluate for spike correction.

Parameter: double SA_ToleranceRange SA_ToleranceRange

Direction: Up

Valid values:

- Minimum:** 0.0
- Maximum:** measuring range

Unit: mm

Description: Tolerance range for spike correction.

Parameter: int32_t SA_NbrCorrectedValues	SA_NbrCorrectedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100	
Description: Number of values to correct at spike correction.	
Parameter: int32_t SA_StatisticSignal	SA_StatisticSignal
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Edge, Light-Dark (EHL)	
1= Edge, Dark-Light (ELH)	
2= Diameter, Edge A (DA)	
3= Diameter, Edge B (DB)	
4= Diameter, Difference (DD)	
5= Diameter, Center axis (DC)	
6= Gap, Edge A (GA)	
7= Gap, Edge B (GB)	
8= Gap, Difference (GD)	
9= Gap, Center axis (GC)	
10= Segment 1, Edge A (S1A)	
11= Segment 1, Edge B (S1B)	
12= Segment 1, Difference (S1D)	
13= Segment 1, Center axis (S1C)	
14= Segment 2, Edge A (S2A)	
15= Segment 2, Edge B (S2B)	
16= Segment 2, Difference (S2D)	
17= Segment 2, Center axis(S2C)	
18= Segment 3, Edge A (S3A)	
19= Segment 3, Edge B (S3B)	
20= Segment 3, Difference (S3D)	
21= Segment 3, Center axis (S3C)	
22= Segment 4, Edge A (S4A)	
23= Segment 4, Edge B (S4B)	
24= Segment 4, Difference (S4D)	
25= Segment 4, Center axis (S4C)	
26= Segment 5, Edge A (S5A)	
27= Segment 5, Edge B (S5B)	
28= Segment 5, Difference (S5D)	
29= Segment 5, Center axis (S5C)	
30= Segment 6, Edge A (S6A)	
31= Segment 6, Edge B (S6B)	
32= Segment 6, Difference (S6D)	
33= Segment 6, Center axis (S6C)	
34= Segment 7, Edge A (S7A)	
35= Segment 7, Edge B (S7B)	
36= Segment 7, Difference (S7D)	
37= Segment 7, Center axis (S7C)	
38= Segment 8, Edge A (S8A)	
39= Segment 8, Edge B (S8B)	
40= Segment 8, Difference (S8D)	
41= Segment 8, Center axis (S8C)	

Description: Value which is used for statistic calculation.

13.3. Commands for ODC2520

Parameter: int32_t SA_Statistic2Signal SA_Statistic2Signal

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)
- 2= Diameter, Edge A (DA)
- 3= Diameter, Edge B (DB)
- 4= Diameter, Difference (DD)
- 5= Diameter, Center axis (DC)
- 6= Gap, Edge A (GA)
- 7= Gap, Edge B (GB)
- 8= Gap, Difference (GD)
- 9= Gap, Center axis (GC)
- 10= Segment 1, Edge A (S1A)
- 11= Segment 1, Edge B (S1B)
- 12= Segment 1, Difference (S1D)
- 13= Segment 1, Center axis (S1C)
- 14= Segment 2, Edge A (S2A)
- 15= Segment 2, Edge B (S2B)
- 16= Segment 2, Difference (S2D)
- 17= Segment 2, Center axis (S2C)
- 18= Segment 3, Edge A (S3A)
- 19= Segment 3, Edge B (S3B)
- 20= Segment 3, Difference (S3D)
- 21= Segment 3, Center axis (S3C)
- 22= Segment 4, Edge A (S4A)
- 23= Segment 4, Edge B (S4B)
- 24= Segment 4, Difference (S4D)
- 25= Segment 4, Center axis (S4C)
- 26= Segment 5, Edge A (S5A)
- 27= Segment 5, Edge B (S5B)
- 28= Segment 5, Difference (S5D)
- 29= Segment 5, Center axis (S5C)
- 30= Segment 6, Edge A (S6A)
- 31= Segment 6, Edge B (S6B)
- 32= Segment 6, Difference (S6D)
- 33= Segment 6, Center axis (S6C)
- 34= Segment 7, Edge A (S7A)
- 35= Segment 7, Edge B (S7B)
- 36= Segment 7, Difference (S7D)
- 37= Segment 7, Center axis (S7C)
- 38= Segment 8, Edge A (S8A)
- 39= Segment 8, Edge B (S8B)
- 40= Segment 8, Difference (S8D)
- 41= Segment 8, Center axis (S8C)

Description: Value which is used for second statistic calculation.

Parameter: int32_t SA_StatisticDepth SA_StatisticDepth

Direction: Up

Valid values:

Minimum: 2

Maximum: 2147483647 (INT32_MAX)

Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 8192). Value greater as 8192 calculates statistic over all values.

Parameter: int32_t SA_MasterSignal

SA_MasterSignal

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)
- 2= Diameter, Edge A (DA)
- 3= Diameter, Edge B (DB)
- 4= Diameter, Difference (DD)
- 5= Diameter, Center axis (DC)
- 6= Gap, Edge A (GA)
- 7= Gap, Edge B (GB)
- 8= Gap, Difference (GD)
- 9= Gap, Center axis (GC)
- 10= Segment 1, Edge A (S1A)
- 11= Segment 1, Edge B (S1B)
- 12= Segment 1, Difference (S1D)
- 13= Segment 1, Center axis (S1C)
- 14= Segment 2, Edge A (S2A)
- 15= Segment 2, Edge B (S2B)
- 16= Segment 2, Difference (S2D)
- 17= Segment 2, Center axis(S2C)
- 18= Segment 3, Edge A (S3A)
- 19= Segment 3, Edge B (S3B)
- 20= Segment 3, Difference (S3D)
- 21= Segment 3, Center axis (S3C)
- 22= Segment 4, Edge A (S4A)
- 23= Segment 4, Edge B (S4B)
- 24= Segment 4, Difference (S4D)
- 25= Segment 4, Center axis (S4C)
- 26= Segment 5, Edge A (S5A)
- 27= Segment 5, Edge B (S5B)
- 28= Segment 5, Difference (S5D)
- 29= Segment 5, Center axis (S5C)
- 30= Segment 6, Edge A (S6A)
- 31= Segment 6, Edge B (S6B)
- 32= Segment 6, Difference (S6D)
- 33= Segment 6, Center axis (S6C)
- 34= Segment 7, Edge A (S7A)
- 35= Segment 7, Edge B (S7B)
- 36= Segment 7, Difference (S7D)
- 37= Segment 7, Center axis (S7C)
- 38= Segment 8, Edge A (S8A)
- 39= Segment 8, Edge B (S8B)
- 40= Segment 8, Difference (S8D)
- 41= Segment 8, Center axis (S8C)

Description: Value which is used for mastering.

Parameter: int32_t SA_Master

SA_Master

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= no (NONE)
- 1= yes (MASTER)

Description: Specifies if mastering is active.

Parameter: double SA_MasterValue

SA_MasterValue

Direction: Up

Valid values:

- Minimum:** -measuring range
Maximum: +measuring range

Unit: mm

Description: Master value

Parameter: int32_t SA_DataOutInterface

SA_DataOutInterface

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= RS422
- 2= Ethernet
- 3= HTTP

Description: Active interface for data output.

Parameter: int32_t SA_Resampling

SA_Resampling

Direction: Up

Valid values:

- Minimum:** 1
Maximum: 150000

Description: Resampling value.

Parameter: int32_t SA_ResampleAnalog

SA_ResampleAnalog

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: Analog output is resampled.

Parameter: int32_t SA_ResampleRS422

SA_ResampleRS422

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: RS422 output is resampled.

Parameter: int32_t SA_ResampleEthernet

SA_ResampleEthernet

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: Output over ethernet is resampled.

Parameter: int32_t SA_HoldLastValid	SA_HoldLastValid
Direction: Up	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	
Parameter: int32_t SA_OutputEdgeLightDark_RS422	SA_OutputEdgeLightDark_-RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge light-dark is transmitted.	
Parameter: int32_t SA_OutputEdgeLightDark_ETH	SA_OutputEdgeLightDark_-ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge light-dark is transmitted.	
Parameter: int32_t SA_OutputEdgeDarkLight_RS422	SA_OutputEdgeDarkLight_-RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge dark-light is transmitted.	
Parameter: int32_t SA_OutputEdgeDarkLight_ETH	SA_OutputEdgeDarkLight_-ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge dark-light is transmitted.	
Parameter: int32_t SA_OutputDiameterEdgeA_RS422	SA_OutputDiameterEdgeA_-RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A (first light-dark edge) is transmitted.	
Parameter: int32_t SA_OutputDiameterEdgeB_RS422	SA_OutputDiameterEdgeB_-RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B (last dark-light edge) is transmitted.	
Parameter: int32_t SA_OutputDiameterDifference_RS422	SA_OutputDiameterDifference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) is transmitted.	

13.3. Commands for ODC2520

Parameter: int32_t SA_OutputDiameterCenterAxis_RS422	SA_OutputDiameterCenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) is transmitted.	
Parameter: int32_t SA_OutputDiameterEdgeA_ETH	SA_OutputDiameterEdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A (first light-dark edge) is transmitted.	
Parameter: int32_t SA_OutputDiameterEdgeB_ETH	SA_OutputDiameterEdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B (last dark-light edge) is transmitted.	
Parameter: int32_t SA_OutputDiameterDifference_ETH	SA_OutputDiameterDifference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) is transmitted.	
Parameter: int32_t SA_OutputDiameterCenterAxis_ETH	SA_OutputDiameterCenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) is transmitted.	
Parameter: int32_t SA_OutputGapEdgeA_RS422	SA_OutputGapEdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A (first dark-light edge) is transmitted.	
Parameter: int32_t SA_OutputGapEdgeB_RS422	SA_OutputGapEdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B (following edge after A) is transmitted.	
Parameter: int32_t SA_OutputGapDifference_RS422	SA_OutputGapDifference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) is transmitted.	

Parameter: int32_t SA_OutputGapCenterAxis_RS422	SA_OutputGapCenterAxis_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) is transmitted.	
Parameter: int32_t SA_OutputGapEdgeA_ETH	SA_OutputGapEdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A (first dark-light edge) is transmitted.	
Parameter: int32_t SA_OutputGapEdgeB_ETH	SA_OutputGapEdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B (following edge after A) is transmitted.	
Parameter: int32_t SA_OutputGapDifference_ETH	SA_OutputGapDifference_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) is transmitted.	
Parameter: int32_t SA_OutputGapCenterAxis_ETH	SA_OutputGapCenterAxis_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) is transmitted.	
Parameter: int32_t SA_OutputSegment1EdgeA_RS422	SA_OutputSegment1EdgeA_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment1EdgeB_RS422	SA_OutputSegment1EdgeB_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment1Difference_RS422	SA_OutputSeg- ment1Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 1 is transmitted.	

Parameter: int32_t SA_OutputSegment1CenterAxis_RS422	SA_OutputSegment1CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment2EdgeA_RS422	SA_OutputSegment2EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment2EdgeB_RS422	SA_OutputSegment2EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment2Difference_RS422	SA_OutputSegment2Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment2CenterAxis_RS422	SA_OutputSegment2CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment3EdgeA_RS422	SA_OutputSegment3EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment3EdgeB_RS422	SA_OutputSegment3EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment3Difference_RS422	SA_OutputSegment3Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 3 is transmitted.	

Parameter: int32_t SA_OutputSegment3CenterAxis_RS422	SA_OutputSegment3CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment4EdgeA_RS422	SA_OutputSegment4EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment4EdgeB_RS422	SA_OutputSegment4EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment4Difference_RS422	SA_OutputSegment4Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment4CenterAxis_RS422	SA_OutputSegment4CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment5EdgeA_RS422	SA_OutputSegment5EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment5EdgeB_RS422	SA_OutputSegment5EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment5Difference_RS422	SA_OutputSegment5Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 5 is transmitted.	

Parameter: int32_t SA_OutputSegment5CenterAxis_RS422	SA_OutputSegment5CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment6EdgeA_RS422	SA_OutputSegment6EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment6EdgeB_RS422	SA_OutputSegment6EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment6Difference_RS422	SA_OutputSegment6Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment6CenterAxis_RS422	SA_OutputSegment6CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment7EdgeA_RS422	SA_OutputSegment7EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment7EdgeB_RS422	SA_OutputSegment7EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment7Difference_RS422	SA_OutputSegment7Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 7 is transmitted.	

Parameter: int32_t SA_OutputSegment7CenterAxis_RS422	SA_OutputSegment7CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment8EdgeA_RS422	SA_OutputSegment8EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment8EdgeB_RS422	SA_OutputSegment8EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment8Difference_RS422	SA_OutputSegment8Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment8CenterAxis_RS422	SA_OutputSegment8CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment1EdgeA_ETH	SA_OutputSegment1EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment1EdgeB_ETH	SA_OutputSegment1EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment1Difference_ETH	SA_OutputSegment1Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 1 is transmitted.	

Parameter: int32_t SA_OutputSegment1CenterAxis_ETH	SA_OutputSegment1CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment2EdgeA_ETH	SA_OutputSegment2EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment2EdgeB_ETH	SA_OutputSegment2EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment2Difference_ETH	SA_OutputSegment2Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment2CenterAxis_ETH	SA_OutputSegment2CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment3EdgeA_ETH	SA_OutputSegment3EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment3EdgeB_ETH	SA_OutputSegment3EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment3Difference_ETH	SA_OutputSegment3Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 3 is transmitted.	

Parameter: int32_t SA_OutputSegment3CenterAxis_ETH	SA_OutputSegment3CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment4EdgeA_ETH	SA_OutputSegment4EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment4EdgeB_ETH	SA_OutputSegment4EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment4Difference_ETH	SA_OutputSegment4Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment4CenterAxis_ETH	SA_OutputSegment4CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment5EdgeA_ETH	SA_OutputSegment5EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment5EdgeB_ETH	SA_OutputSegment5EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment5Difference_ETH	SA_OutputSegment5Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 5 is transmitted.	

Parameter: int32_t SA_OutputSegment5CenterAxis_ETH	SA_OutputSegment5CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment6EdgeA_ETH	SA_OutputSegment6EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment6EdgeB_ETH	SA_OutputSegment6EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment6Difference_ETH	SA_OutputSegment6Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment6CenterAxis_ETH	SA_OutputSegment6CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment7EdgeA_ETH	SA_OutputSegment7EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment7EdgeB_ETH	SA_OutputSegment7EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment7Difference_ETH	SA_OutputSegment7Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 7 is transmitted.	

Parameter: int32_t SA_OutputSegment7CenterAxis_ETH	SA_OutputSegment7CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment8EdgeA_ETH	SA_OutputSegment8EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment8EdgeB_ETH	SA_OutputSegment8EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment8Difference_ETH	SA_OutputSegment8Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment8CenterAxis_ETH	SA_OutputSegment8CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 8 is transmitted.	
Parameter: int32_t SA_OutputStatisticMin_RS422	SA_OutputStatisticMin_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic min value is transmitted.	
Parameter: int32_t SA_OutputStatisticMax_RS422	SA_OutputStatisticMax_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic max value is transmitted.	
Parameter: int32_t SA_OutputStatisticPeak2Peak_RS422	SA_OutputStatisticPeak2Peak_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic peak to peak value is transmitted.	

13.3. Commands for ODC2520

Parameter: int32_t SA_OutputStatistic2Min_RS422	SA_OutputStatistic2Min_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic min value is transmitted.	
Parameter: int32_t SA_OutputStatistic2Max_RS422	SA_OutputStatistic2Max_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic max value is transmitted.	
Parameter: int32_t SA_OutputStatistic2Peak2Peak_RS422	SA_OutputStatis- tic2Peak2Peak_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic peak to peak value is transmitted.	
Parameter: int32_t SA_OutputStatisticMin_ETH	SA_OutputStatisticMin_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic min value is transmitted.	
Parameter: int32_t SA_OutputStatisticMax_ETH	SA_OutputStatisticMax_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic max value is transmitted.	
Parameter: int32_t SA_OutputStatisticPeak2Peak_ETH	SA_OutputStatistic- Peak2Peak_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic peak to peak value is transmitted.	
Parameter: int32_t SA_OutputStatistic2Min_ETH	SA_OutputStatistic2Min_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic min value is transmitted.	
Parameter: int32_t SA_OutputStatistic2Max_ETH	SA_OutputStatistic2Max_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic max value is transmitted.	

Parameter: int32_t SA_OutputStatistic2Peak2Peak_ETH	SA_OutputStatistic2Peak2Peak_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second staticic peak to peak value is transmitted.	
Parameter: int32_t SA_OutputAdditionalNbrEdges_RS422	SA_OutputAdditionalNbrEdges_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if number of edges is transmitted.	
Parameter: int32_t SA_OutputAdditionalNbrPins_RS422	SA_OutputAdditionalNbrPins_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if number of pins (dark areas) is transmitted.	
Parameter: int32_t SA_OutputAdditionalNbrGaps_RS422	SA_OutputAdditionalNbrGaps_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if number of gaps (light areas) is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_RS422	SA_OutputAdditionalCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_RS422	SA_OutputAdditionalTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_RS422	SA_OutputAdditionalState_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputAdditionalNbrEdges_ETH	SA_OutputAdditionalNbrEdges_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if number of edges is transmitted.	

13.3. Commands for ODC2520

Parameter: int32_t SA_OutputAdditionalNbrPins_ETH	SA_OutputAdditionalNbrPins_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if number of pins (dark areas) is transmitted.	
Parameter: int32_t SA_OutputAdditionalNbrGaps_ETH	SA_OutputAdditionalNbrGaps_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if number of gaps (light areas) is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditionalCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_ETH	SA_OutputAdditionalState_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	
Parameter: int32_t SA_OutputVideoRaw_ETH	SA_OutputVideoRaw_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal is transmitted.	
Parameter: int32_t SA_OutputVideoLight_ETH	SA_OutputVideoLight_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if light preprocessed video signal is transmitted.	
Parameter: int32_t SA_OutputVideoLightTable_ETH	SA_OutputVideoLightTable_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if light table is transmitted.	

Parameter: int32_t SA_OutputVideoThreshold_ETH	SA_OutputVideoThreshold_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if video threshold table is transmitted.	
Parameter: int32_t SA_ErrorOutput1..2	SA_ErrorOutput1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Wrong number of edges (ER1)	
2= Measurement error (ER2)	
3= Below low limit (LI1)	
4= Above high limit (LI2)	
5= Out of limits (LI12)	
Description: Condition for error output.	
Parameter: int32_t SA_DataSource	SA_DataSource
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Edge, Light-Dark (EHL)	
1= Edge, Dark-Light (ELH)	
2= Diameter, Edge A (DA)	
3= Diameter, Edge B (DB)	
4= Diameter, Difference (DD)	
5= Diameter, Center axis (DC)	
6= Gap, Edge A (GA)	
7= Gap, Edge B (GB)	
8= Gap, Difference (GD)	
9= Gap, Center axis (GC)	
10= Segment 1, Edge A (S1A)	
11= Segment 1, Edge B (S1B)	
12= Segment 1, Difference (S1D)	
13= Segment 1, Center axis (S1C)	
14= Segment 2, Edge A (S2A)	
15= Segment 2, Edge B (S2B)	
16= Segment 2, Difference (S2D)	
17= Segment 2, Center axis (S2C)	
18= Segment 3, Edge A (S3A)	
19= Segment 3, Edge B (S3B)	
20= Segment 3, Difference (S3D)	
21= Segment 3, Center axis (S3C)	
22= Segment 4, Edge A (S4A)	
23= Segment 4, Edge B (S4B)	
24= Segment 4, Difference (S4D)	
25= Segment 4, Center axis (S4C)	
26= Segment 5, Edge A (S5A)	
27= Segment 5, Edge B (S5B)	
28= Segment 5, Difference (S5D)	
29= Segment 5, Center axis (S5C)	

30= Segment 6, Edge A (S6A)
 31= Segment 6, Edge B (S6B)
 32= Segment 6, Difference (S6D)
 33= Segment 6, Center axis (S6C)
 34= Segment 7, Edge A (S7A)
 35= Segment 7, Edge B (S7B)
 36= Segment 7, Difference (S7D)
 37= Segment 7, Center axis (S7C)
 38= Segment 8, Edge A (S8A)
 39= Segment 8, Edge B (S8B)
 40= Segment 8, Difference (S8D)
 41= Segment 8, Center axis (S8C)

Description: Data source to be checked.

Parameter: double SA_LowerLimit	SA_LowerLimit
Direction: Up	
Valid values:	
Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)	
Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)	
Unit: mm	
Description: Lower limit.	
Parameter: double SA_UpperLimit	SA_UpperLimit
Direction: Up	
Valid values:	
Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)	
Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)	
Unit: mm	
Description: Upper limit.	
Parameter: int32_t SA_ErrorOut1HoldTime	SA_ErrorOut1HoldTime
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10000	
Unit: ms	
Description: Minimum hold period. 0 means no value to hold (NONE).	
Parameter: int32_t SA_ErrorOut2HoldTime	SA_ErrorOut2HoldTime
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 10000	
Unit: ms	
Description: Minimum hold period. 0 means no value to hold (NONE).	
Parameter: int32_t SA_ErrorLevelOut1..2	SA_ErrorLevelOut1..2
Direction: Up	
Valid values:	
0= NPN	
1= PNP	
2= Push-Pull (PUSHPULL)	
3= Push-Pull negated (PUSHPULLNEG)	
Description: Error level.	

Parameter: int32_t SA_AnalogOutput SA_AnalogOutput

Direction: Up

Valid values:

- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)
- 2= Diameter, Edge A (DA)
- 3= Diameter, Edge B (DB)
- 4= Diameter, Difference (DD)
- 5= Diameter, Center axis (DC)
- 6= Gap, Edge A (GA)
- 7= Gap, Edge B (GB)
- 8= Gap, Difference (GD)
- 9= Gap, Center axis (GC)
- 10= Segment 1, Edge A (S1A)
- 11= Segment 1, Edge B (S1B)
- 12= Segment 1, Difference (S1D)
- 13= Segment 1, Center axis (S1C)
- 14= Segment 2, Edge A (S2A)
- 15= Segment 2, Edge B (S2B)
- 16= Segment 2, Difference (S2D)
- 17= Segment 2, Center axis(S2C)
- 18= Segment 3, Edge A (S3A)
- 19= Segment 3, Edge B (S3B)
- 20= Segment 3, Difference (S3D)
- 21= Segment 3, Center axis (S3C)
- 22= Segment 4, Edge A (S4A)
- 23= Segment 4, Edge B (S4B)
- 24= Segment 4, Difference (S4D)
- 25= Segment 4, Center axis (S4C)
- 26= Segment 5, Edge A (S5A)
- 27= Segment 5, Edge B (S5B)
- 28= Segment 5, Difference (S5D)
- 29= Segment 5, Center axis (S5C)
- 30= Segment 6, Edge A (S6A)
- 31= Segment 6, Edge B (S6B)
- 32= Segment 6, Difference (S6D)
- 33= Segment 6, Center axis (S6C)
- 34= Segment 7, Edge A (S7A)
- 35= Segment 7, Edge B (S7B)
- 36= Segment 7, Difference (S7D)
- 37= Segment 7, Center axis (S7C)
- 38= Segment 8, Edge A (S8A)
- 39= Segment 8, Edge B (S8B)
- 40= Segment 8, Difference (S8D)
- 41= Segment 8, Center axis (S8C)

Description: Data to be used for analog output.

Parameter: int32_t SA_AnalogRange SA_AnalogRange

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= 0 - 10V

Description: Analog output range.

Parameter: int32_t SA_AnalogScaleMode	SA_AnalogScaleMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Standard	
1 = Two point (TWOPOINT)	
Description: Analog scale mode.	
Parameter: double SA_MinValue	SA_MinValue
Direction: Up	
Valid values:	
Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)	
Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)	
Unit: mm	
Description: Value which represents lowest voltage/current (at two point scaling).	
Parameter: double SA_MaxValue	SA_MaxValue
Direction: Up	
Valid values:	
Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)	
Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)	
Unit: mm	
Description: Value which represents highest voltage/current (at two point scaling).	
Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Description: Name of the controller.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the controller.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the controller.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the controller.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
46.0	
95.0	
Unit: mm	
Description: Range of the sensor (from firmware 005.024.035).	

Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description:	MAC address (low level ethernet address) of the controller.
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description:	Software version of firmware in the controller.
Parameter: String SA_Image	SA_Image
Direction: Up	
Description:	Active software, ethernet or ethercat.
Parameter: String SA_ImageType	SA_ImageType
Direction: Up	
Description:	Firmware image type used by the controller.
Parameter: String SA_HardwareRevision	SA_HardwareRevision
Direction: Up	
Valid values:	
Numeric value	
Description:	Revision index of the controller board.
Parameter: int32_t SA_MeasDistIndex	SA_MeasDistIndex
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 8	
Description:	Index of the active measurement distance in the table.
Parameter: double SA_MeasDistance	SA_MeasDistance
Direction: Up	
Description:	Active measurement distance.
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description:	Unit of the measurement distance.
Parameter: String SA_MeasDistTable	SA_MeasDistTable
Direction: Up	
Description:	Whole table in one string, separated by new lines and commas.

13.3.1.1.6 Set_SyncMode (SYNC)

Set the synchronization mode.

Parameter: int32_t SP_SyncMode	SP_SyncMode
Direction: Down	
Valid values:	
0= None	
1= Slave	
2= Master	
Description:	Synchronization mode.

Parameter: int32_t SP_SyncTermination SP_SyncTermination
Direction: Down
Valid values:
 0= Off (TERMOFF)
 1= On 120 Ohm (TERMON)
Description: Termination of external input.

13.3.1.1.7 Get_SyncMode (SYNC)

Get the synchronization mode.

Parameter: int32_t SA_SyncMode SA_SyncMode
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= None
 1= Slave
 2= Master
Description: Synchronization mode.

Parameter: int32_t SA_SyncTermination SA_SyncTermination
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Off (TERMOFF)
 1= On 120 Ohm (TERMON)
Description: Termination of external input.

13.3.1.1.8 Set_Unit (UNIT)

Set the unit for configuration and display in the web diagram.

Parameter: int32_t SP_DisplayUnit SP_DisplayUnit
Direction: Down
Valid values:
 0= mm
 1= Inch
Description: Unit.

13.3.1.1.9 Get_Unit (UNIT)

Get the unit for configuration and display in the web diagram.

Parameter: int32_t SA_DisplayUnit SA_DisplayUnit
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= mm
 1= Inch
Description: Unit.

13.3.1.1.10 Reset_Boot (RESET)

Resets the sensor.

At this command the controller may change output data after reboot. If first bit of `IP_AutomaticMode` is set (1), [Get_AllParameters](#) (`SP_Additional= 1`) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

13.3.1.2 User level

13.3.1.2.1 Logout (LOGOUT)

Change user level to user.

13.3.1.2.2 Login (LOGIN)

Change user level to professional.

Parameter: String `SP_Password`

`SP_Password`

Direction: Down

Description: Valid password to login.

13.3.1.2.3 Get_UserLevel (GETUSERLEVEL)

Retrieve actual user level.

Parameter: int32_t `SA_UserLevel`

`SA_UserLevel`

Direction: Up

Valid values:

-1 = Other user level (only for internal use)

0 = User

1 = Professional

Description: Actual user level.

13.3.1.2.4 Set_DefaultUser (STDUSER)

Set the default user level after booting the system.

Parameter: int32_t `SP_DefaultUser`

`SP_DefaultUser`

Direction: Down

Valid values:

0 = User

1 = Professional

Description: Default user level.

13.3.1.2.5 Get_DefaultUser (STDUSER)

Get the default user level after booting the system.

Parameter: int32_t SA_DefaultUser SA_DefaultUser
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = User
 1 = Professional
Description: Default user level.

13.3.1.2.6 Set_Password (PASSWD)

Change the password for login.

Parameter: String SP_OldPassword SP_OldPassword
Direction: Down
Description: Old password.

Parameter: String SP_NewPassword SP_NewPassword
Direction: Down
Description: New password.

13.3.1.3 Measure distance

13.3.1.3.1 Get_MeasDistTable (MEASDISTTABLE)

Get a list of all calibrated measure distances.

Parameter: String SA_MeasDistTable SA_MeasDistTable
Direction: Up
Description: Whole table in one string, separated by new lines and commas.

Parameter: int32_t SA_MeasDistTableCount SA_MeasDistTableCount
Direction: Up
Valid values:
Minimum: 0
Maximum: 8
Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_MeasDistIndex1, SA_MeasDistIndex2, ...

Parameter: int32_t SA_MeasDistIndex1..x SA_MeasDistIndex1..x
Direction: Up
Valid values:
Minimum: 0
Maximum: 8
Description: Index of the sensor head in the table.

Parameter: double SA_MeasDistance1..x SA_MeasDistance1..x
Direction: Up
Description: Measurement distance of receiver in the table.

Parameter: String SA_Unit_{1..x} SA_Unit_{1..x}

Direction: Up

Description: Unit of the measurement distance.

13.3.1.3.2 Set_ActiveMeasDist (MEASDIST)

Change to another measurement distance.

Parameter: int32_t SP_MeasDistIndex SP_MeasDistIndex

Direction: Down

Valid values:

Minimum: 0

Maximum: 8

Description: Index of new measurement distance.

13.3.1.3.3 Get_ActiveMeasDist (MEASDIST)

Get active measurement distance.

Parameter: int32_t SA_MeasDistIndex SA_MeasDistIndex

Direction: Up

Valid values:

Minimum: 0

Maximum: 8

Description: Index of active measurement distance.

13.3.1.3.4 Get_MeasDistInfo (MEASDISTINFO)

Get information of active measurement distance.

Parameter: int32_t SA_MeasDistIndex SA_MeasDistIndex

Direction: Up

Valid values:

Minimum: 0

Maximum: 8

Description: Index of the active measurement distance in the table.

Parameter: double SA_MeasDistance SA_MeasDistance

Direction: Up

Description: Active measurement distance.

Parameter: String SA_Unit SA_Unit

Direction: Up

Description: Unit of the measurement distance.

13.3.1.3.5 LightCorr (LIGHTCORR)

Make a light correction.

13.3.1.4 Triggering

13.3.1.4.1 Set_TriggerMode (TRIGGER)

Set the trigger mode.

Parameter: int32_t SP_TriggerMode

SP_TriggerMode

Direction: Down

Valid values:

- 0= None
- 1= Edge
- 2= Level (PULSE)
- 3= Software

Description: Trigger mode.

Parameter: int32_t SP_TriggerTermination

SP_TriggerTermination

Direction: Down

Valid values:

- 0= Off (TERMOFF)
- 1= On 120 Ohm (TERMON)

Description: Termination of external input.

13.3.1.4.2 Get_TriggerMode (TRIGGER)

Get the active trigger mode.

Parameter: int32_t SA_TriggerMode

SA_TriggerMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Edge
- 2= Level (PULSE)
- 3= Software

Description: Trigger mode.

Parameter: int32_t SA_TriggerTermination

SA_TriggerTermination

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Off (TERMOFF)
- 1= On 120 Ohm (TERMON)

Description: Termination of external input.

13.3.1.4.3 Set_TriggerMoment (TRIGGERAT)

Set the trigger time.

Parameter: int32_t SP_TriggerMoment

SP_TriggerMoment

Direction: Down

Valid values:

- 0= Input
- 1= Output

Description: Trigger moment.

13.3.1.4.4 Get_TriggerMoment (TRIGGERAT)

Get the active trigger time.

Parameter: int32_t SA_TriggerMoment SA_TriggerMoment
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = Input
 1 = Output
Description: Trigger moment.

13.3.1.4.5 Set_TriggerLevel (TRIGGERLEVEL)

Set the trigger level.

Parameter: int32_t SP_TriggerLevel SP_TriggerLevel
Direction: Down
Valid values:
 0 = High
 1 = Low
Description: Trigger level.

13.3.1.4.6 Get_TriggerLevel (TRIGGERLEVEL)

Get the active trigger level.

Parameter: int32_t SA_TriggerLevel SA_TriggerLevel
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = High
 1 = Low
Description: Trigger level.

13.3.1.4.7 Set_TriggerCount (TRIGGERCOUNT)

Set the number of values to measure at trigger.

Parameter: int32_t SP_TriggerCount SP_TriggerCount
Direction: Down
Valid values:
Minimum: 0
Maximum: 16383
Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

13.3.1.4.8 Get_TriggerCount (TRIGGERCOUNT)

Get the number of values to measure at trigger.

Parameter: int32_t SA_TriggerCount SA_TriggerCount
Direction: Up
Valid values:
Minimum: 0
Maximum: 16383
Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

13.3.1.4.9 Software_Trigger (TRIGGERSW)

Execute a software trigger.

13.3.1.5 Interfaces

13.3.1.5.1 Set_IPConfiguration (IPCONFIG)

Set the IP configuration at controller.

Parameter: int32_t SP_DHCPEnabled SP_DHCPEnabled
Direction: Down
Valid values:
 0 = FALSE
 1 = TRUE
Description: Specify if controller should use a static IP address or ask for IP at DHCP server (dynamic IP address).

Parameter: String SP_Address SP_Address
Direction: Down
Valid values:
 Valid IP address in form of xxx.xxx.xxx.xxx
Description: IP address of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_SubnetMask SP_SubnetMask
Direction: Down
Valid values:
 Valid network mask (e.g. 255.255.255.0 for a Class C network)
Description: Network mask of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_Gateway SP_Gateway
Direction: Down
Valid values:
 Valid IP address of default gateway in form of xxx.xxx.xxx.xxx
Description: The default gateway must be specified if the controller should communicate with peers in foreign subnets. This parameter is only evaluated on static IP assignment.

13.3.1.5.2 Get_IPConfiguration (IPCONFIG)

Get the IP configuration at controller.

Parameter: int32_t SA_DHCPEnabled SA_DHCPEnabled

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= FALSE
- 1= TRUE

Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).

Parameter: String SA_Address SA_Address

Direction: Up

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.

Parameter: String SA_SubnetMask SA_SubnetMask

Direction: Up

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.

Parameter: String SA_Gateway SA_Gateway

Direction: Up

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

13.3.1.5.3 Set_IPDataTransferMode (MEATRANSFER)

Set IP protocol at controller.

Parameter: int32_t SP_Protocol SP_Protocol

Direction: Down

Valid values:

- 0= TCP server (SERVER/TCP)
- 1= TCP client (CLIENT/TCP)
- 2= UDP sender (CLIENT/UDP)
- 3= None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SP_RemoteAddress SP_RemoteAddress

Direction: Down

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to. On TCP server this parameter is ignored.

13.3. Commands for ODC2520

Parameter: int32_t SP_Port SP_Port
Direction: Down
Valid values:
Minimum: 1024
Maximum: 65535
Description: Port to send data to or to listen for incoming requests.

13.3.1.5.4 Get_IPDataTransferMode (MEASTRANSFER)

Get IP protocol at controller.

Parameter: int32_t SA_Protocol SA_Protocol
Direction: Up
Valid values:
-1= Unknown parameter value from sensor
0= TCP server (SERVER/TCP)
1= TCP client (CLIENT/TCP)
2= UDP sender (CLIENT/UDP)
3= None
Description: Specifies if data should be send using TCP or UDP.

Parameter: String SA_RemoteAddress SA_RemoteAddress
Direction: Up
Valid values:
Valid IP address of receiver of data
Description: Address of remote computer to send data to.
Parameter: int32_t SA_Port SA_Port
Direction: Up
Valid values:
Minimum: 1024
Maximum: 65535
Description: Port to send data to or to listen for incoming requests.

13.3.1.5.5 Set_EthernetMode (ETHERMODE)

Switches ethernet mode between Ethernet and Ethercat.

Parameter: int32_t SP_EthernetMode SP_EthernetMode
Direction: Down
Valid values:
0= Ethernet
1= Ethercat
Description: Ethernet mode.

13.3.1.5.6 Get_EthernetMode (ETHERMODE)

Get ethernet mode of controller.

Parameter: int32_t SA_EthernetMode SA_EthernetMode
Direction: Up
Valid values:
-1= Unknown parameter value from sensor
0= Ethernet
1= Ethercat
Description: Ethernet mode.

13.3.1.5.7 Set_Baudrate (BAUDRATE)

Set baudrate of controller for serial RS422 communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

- 4000000
- 3500000 (does not work in combination with IF2008 PCI interface card)
- 3000000 (does not work in combination with IF2008 PCI interface card)
- 2500000
- 2000000
- 1500000
- 921600
- 691200
- 460800
- 230400
- 115200
- 9600

Unit: Baud

Description: Baudrate of controller.

Parameter: int32_t CP_InterruptDataTransfer

CP_InterruptDataTransfer

Direction: Down

Valid values:

- 0= false
- 1= true

Default: 0

Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Set_DataOutInterface](#)) and enabling it again at end (only if enabled before).

13.3.1.5.8 Get_Baudrate (BAUDRATE)

Get baudrate of controller for serial RS422 communication.

Parameter: int32_t SA_SensorBaudrate

SA_SensorBaudrate

Direction: Up

Valid values:

- 4000000
- 3500000
- 3000000
- 2500000
- 2000000
- 1500000
- 921600
- 691200
- 460800
- 230400
- 115200
- 9600

Unit: Baud

Description: Baudrate of controller.

13.3.1.6 Parameter management

13.3.1.6.1 Save_Parameters (STORE)

Save actual parameters at controller. There can be saved several settings on different locations. So it is easy to switch to another setting.

Parameter: int32_t SP_ParameterSet

SP_ParameterSet

Direction: Down

Valid values:

Minimum: 1

Maximum: 8

Description: Location to save the settings.

13.3.1.6.2 Load_Parameters (READ)

Load stored parameters into controller RAM.

There can be loaded several settings from different locations. So it is easy to switch to another setting.

At this command the controller may change output data after applying new setting.

If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually.

After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_ParameterType

SP_ParameterType

Direction: Down

Valid values:

0= All settings (ALL)

1= Device settings (DEVICE)

2= Measurement settings (MEAS)

Description: Specifies which settings should be loaded.

Parameter: int32_t SP_ParameterSet

SP_ParameterSet

Direction: Down

Valid values:

Minimum: 1

Maximum: 8

Description: Location from where the settings should be loaded.

13.3.1.6.3 Set_Default (SETDEFAULT)

Reset the controller to default settings.

At this command the sensor may change output data after applying default settings.

If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually.

After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DefaultType

SP_DefaultType

Direction: Down

Valid values:

0= All settings (ALL)

1= Just the current setting (CURRENT)

Description: Specifies which settings should be resetted.

Parameter: int32_t SP_KeepDevice SP_KeepDevice

Direction: Down

Valid values:

0= no

1= yes

Description: Specifies if device settings should be kept temporary.

13.3.2 Measurement

13.3.2.1 General

13.3.2.1.1 Set_MeasureMode (MEASMODE)

Set the measure mode.

At this command the controller may change output data automatically. If first bit of [IP_AutomaticMode](#) is set (1), [Get_AllParameters](#) (SP_Additional= 0) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_MeasureMode SP_MeasureMode

Direction: Down

Valid values:

0= Edge Light-Dark (EDGEHL)

1= Edge Dark-Light (EDGELH)

2= Diameter (DIA)

3= Gap

4= Segments (SEGMENT)

Description: Measure mode.

13.3.2.1.2 Get_MeasureMode (MEASMODE)

Get the measure mode.

Parameter: int32_t SA_MeasureMode SA_MeasureMode

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Edge Light-Dark (EDGEHL)

1= Edge Dark-Light (EDGELH)

2= Diameter (DIA)

3= Gap

4= Segments (SEGMENT)

Description: Measure mode.

13.3.2.1.3 Set_SearchDirection (SEARCHDIR)

Specifiy the direction to search for edges.

Parameter: int32_t SP_SearchDirection SP_SearchDirection

Direction: Down

Valid values:

0= Standard

1= Inverse

Description: Search direction for edges.

13.3.2.1.4 Get_SearchDirection (SEARCHDIR)

Retrieve the direction to search for edges.

Parameter: int32_t SA_SearchDirection

SA_SearchDirection

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Standard
- 1 = Inverse

Description: Search direction for edges.

13.3.2.1.5 Set_MeasureDirection (MEASDIR)

Specifiy the reference point of measured value.

Parameter: int32_t SP_MeasureDirection

SP_MeasureDirection

Direction: Down

Valid values:

- 0 = Standard
- 1 = Inverse

Description: Reaference point.

13.3.2.1.6 Get_MeasureDirection (MEASDIR)

Retrieve the reference point of measured value.

Parameter: int32_t SA_MeasureDirection

SA_MeasureDirection

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Standard
- 1 = Inverse

Description: Reaference point.

13.3.2.1.7 Set_ExpectedEdges (EXPEDGES)

Set the number of expected edges.

Parameter: int32_t SP_ExpectedEdges

SP_ExpectedEdges

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 64

Description: Number of expected edges.

13.3.2.1.8 Get_ExpectedEdges (EXPEDGES)

Get the number of expected edges.

Parameter: int32_t SA_ExpectedEdges SA_ExpectedEdges
Direction: Up
Valid values:
Minimum: 1
Maximum: 64
Description: Number of expected edges.

13.3.2.1.9 Set_SegmentDefinition<n> (DEFSEG<n>)

Define segment n ∈ {1..8}.

Parameter: int32_t SP_Edge<n>_A SP_Edge<n>_A
Direction: Down
Valid values:
Minimum: 0
Maximum: 64
Description: First edge.

Parameter: int32_t SP_Edge<n>_B SP_Edge<n>_B
Direction: Down
Valid values:
Minimum: 0
Maximum: 64
Description: Second edge.

13.3.2.1.10 Get_SegmentDefinition<n> (DEFSEG<n>)

Retrieve segment n ∈ {1..8} definition.

Parameter: int32_t SA_Edge<n>_A SA_Edge<n>_A
Direction: Up
Valid values:
Minimum: 0
Maximum: 64
Description: First edge.

Parameter: int32_t SA_Edge<n>_B SA_Edge<n>_B
Direction: Up
Valid values:
Minimum: 0
Maximum: 64
Description: Second edge.

13.3.2.1.11 Set_LaserPower (LASER)

Specify the laser power at sensor.

Parameter: int32_t SP_LaserPower SP_LaserPower
Direction: Down
Valid values:
 0= Off
 1= On
Description: Laser power.

13.3.2.1.12 Get_LaserPower (LASER)

Get the laser power from sensor.

Parameter: int32_t SA_LaserPower SA_LaserPower
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Off
 1= On
Description: Laser power.

13.3.2.1.13 Get_VideoStreamSignal

Read one video signal from video stream.

Parameter: int32_t SP_ReadMode SP_ReadMode
Direction: Down
Valid values:
 0= Each video signal
 1= Only newest video signal
 2= Automatic
Description: This mode specifies if each video signal should be read or only the latest one. If set to automatic each video signal is read until the buffer does not overflow. If the buffer becomes full one or more video signals are discarded.

Parameter: int32_t SP_WaitVideoTimeout SP_WaitVideoTimeout
Direction: Down
Unit: ms
Valid values:
Minimum: 0
Maximum: 2147483647 (INT32_MAX)
Description: Timeout to wait for a video signal.

Parameter: Binary data SA_VideoRaw SA_VideoRaw
Direction: Up
Valid values:
 768 words (for 46 mm range), 1536 words (for 95 mm range), each word has 2 byte and is an intensity value.
Description: Raw video signal

Parameter: Binary data SA_VideoLight	SA_VideoLight
Direction: Up	
Valid values:	
768 words (for 46 mm range), 1536 words (for 95 mm range), each word has 2 byte and is an intensity value.	
Description: Light corrected video signal	
Parameter: Binary data SA_VideoLightTable	SA_VideoLightTable
Direction: Up	
Valid values:	
768 words (for 46 mm range), 1536 words (for 95 mm range), each word has 2 byte and is an intensity value.	
Description: Light table	
Parameter: Binary data SA_VideoThreshold	SA_VideoThreshold
Direction: Up	
Valid values:	
768 words (for 46 mm range), 1536 words (for 95 mm range), each word has 2 byte and is an intensity value.	
Description: Threshold table	
Parameter: double SA_VideoTimestamp	SA_VideoTimestamp
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1.79769e+308 (DBL_MAX)	
Unit: ms	
Description: Timestamp of the video signal. It starts from 1970 Jan 01 at 01:00. It is generated when the video has arrived at TCP/IP socket.	
Parameter: int32_t SA_SkippedVideo	SA_SkippedVideo
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of skipped video signals, if SP_ReadMode is not 0.	

13.3.2.2 Video signal

13.3.2.2.1 Set_ROI (ROI)

Set the region of interest for processing video signal.

Parameter: int32_t SP_ROIStart	SP_ROIStart
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 767 (for 46 mm range), 1535 (for 95 mm range)	
Unit: Pixel	
Description: First position on CCD.	

Parameter: int32_t SP_ROIEnd SP_ROIEnd
Direction: Down
Valid values:
Minimum: 0
Maximum: 767 (for 46 mm range), 1535 (for 95 mm range)
Unit: Pixel
Description: Last position on CCD.

13.3.2.2.2 Get_ROI (ROI)

Get the region of interest for processing video signal.

Parameter: int32_t SA_ROIStart SA_ROIStart
Direction: Up
Valid values:
Minimum: 0
Maximum: 767 (for 46 mm range), 1535 (for 95 mm range)
Unit: Pixel
Description: First position on CCD.

Parameter: int32_t SA_ROIEnd SA_ROIEnd
Direction: Up
Valid values:
Minimum: 0
Maximum: 767 (for 46 mm range), 1535 (for 95 mm range)
Unit: Pixel
Description: Last position on CCD.

13.3.2.2.3 Set_Threshold (THRESHOLD)

Set threshold for video processing.

Parameter: double SP_Threshold SP_Threshold
Direction: Down
Valid values:
Minimum: 1.0
Maximum: 99.0
Unit: %
Description: Video threshold.

13.3.2.2.4 Get_Threshold (THRESHOLD)

Get threshold for video processing.

Parameter: double SA_Threshold SA_Threshold
Direction: Up
Valid values:
Minimum: 1.0
Maximum: 99.0
Unit: %
Description: Video threshold.

13.3.2.3 Measurement value processing

13.3.2.3.1 Set_Averaging (AVERAGE)

Set data averaging at controller.

Parameter: int32_t SP_AveragingType

SP_AveragingType

Direction: Down

Valid values:

- 0= None
- 1= Moving average (MOVING)
- 2= Recursive averaging (RECURSIVE)
- 3= Median

Description: Averaging type.

Parameter: int32_t SP_MovingCount

SP_MovingCount

Direction: Down

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128

Description: Number of value for the averaging window. This parameter is only used at moving average.

Parameter: int32_t SP_RecursiveCount

SP_RecursiveCount

Direction: Down

Valid values:

- Minimum: 2
- Maximum: 32768

Description: Number of values for recursive averaging. This parameter is only used at recursive average.

Parameter: int32_t SP_MedianCount

SP_MedianCount

Direction: Down

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only used at median.

13.3.2.3.2 Get_Averaging (AVERAGE)

Get data averaging at controller.

Parameter: int32_t SA_AveragingType

SA_AveragingType

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = None
- 1 = Moving average (MOVING)
- 2 = Recursive averaging (RECURSIVE)
- 3 = Median

Description: Averaging type.

Parameter: int32_t SA_MovingCount

SA_MovingCount

Direction: Up

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128

Description: Number of value for the averaging window. This parameter is only available at moving average.

Parameter: int32_t SA_RecursiveCount

SA_RecursiveCount

Direction: Up

Valid values:

- Minimum: 2
- Maximum: 32768

Description: Number of values for recursive averaging. This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount

SA_MedianCount

Direction: Up

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only available at median.

13.3.2.3.3 Set_SpikeCorrection (SPIKECORR)

Set spike correction at controller.

Parameter: int32_t SP_SpikeCorrection

SP_SpikeCorrection

Direction: Down

Valid values:

- 0 = off
- 1 = on

Description: Spike correction.

Parameter: int32_t SP_NbrEvaluatedValues	SP_NbrEvaluatedValues
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Number of values to evaluate for spike correction.	
Parameter: double SP_ToleranceRange	SP_ToleranceRange
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: measuring range	
Unit: mm	
Description: Tolerance range for spike correction.	
Parameter: int32_t SP_NbrCorrectedValues	SP_NbrCorrectedValues
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 100	
Description: Number of values to correct at spike correction.	

13.3.2.3.4 Get_SpikeCorrection (SPIKECORR)

Get spike correction at controller.

Parameter: int32_t SA_SpikeCorrection	SA_SpikeCorrection
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = off	
1 = on	
Description: Spike correction.	
Parameter: int32_t SA_NbrEvaluatedValues	SA_NbrEvaluatedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 10	
Description: Number of values to evaluate for spike correction.	
Parameter: double SA_ToleranceRange	SA_ToleranceRange
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: measuring range	
Unit: mm	
Description: Tolerance range for spike correction.	
Parameter: int32_t SA_NbrCorrectedValues	SA_NbrCorrectedValues
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100	
Description: Number of values to correct at spike correction.	

13.3.2.3.5 Set_StatisticSignal (STATISTICSIGNAL)

Set the measured value which is used for statistic calculation.

Parameter: int32_t SP_StatisticSignal

SP_StatisticSignal

Direction: Down

Valid values:

- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)
- 2= Diameter, Edge A (DA)
- 3= Diameter, Edge B (DB)
- 4= Diameter, Difference (DD)
- 5= Diameter, Center axis (DC)
- 6= Gap, Edge A (GA)
- 7= Gap, Edge B (GB)
- 8= Gap, Difference (GD)
- 9= Gap, Center axis (GC)
- 10= Segment 1, Edge A (S1A)
- 11= Segment 1, Edge B (S1B)
- 12= Segment 1, Difference (S1D)
- 13= Segment 1, Center axis (S1C)
- 14= Segment 2, Edge A (S2A)
- 15= Segment 2, Edge B (S2B)
- 16= Segment 2, Difference (S2D)
- 17= Segment 2, Center axis(S2C)
- 18= Segment 3, Edge A (S3A)
- 19= Segment 3, Edge B (S3B)
- 20= Segment 3, Difference (S3D)
- 21= Segment 3, Center axis (S3C)
- 22= Segment 4, Edge A (S4A)
- 23= Segment 4, Edge B (S4B)
- 24= Segment 4, Difference (S4D)
- 25= Segment 4, Center axis (S4C)
- 26= Segment 5, Edge A (S5A)
- 27= Segment 5, Edge B (S5B)
- 28= Segment 5, Difference (S5D)
- 29= Segment 5, Center axis (S5C)
- 30= Segment 6, Edge A (S6A)
- 31= Segment 6, Edge B (S6B)
- 32= Segment 6, Difference (S6D)
- 33= Segment 6, Center axis (S6C)
- 34= Segment 7, Edge A (S7A)
- 35= Segment 7, Edge B (S7B)
- 36= Segment 7, Difference (S7D)
- 37= Segment 7, Center axis (S7C)
- 38= Segment 8, Edge A (S8A)
- 39= Segment 8, Edge B (S8B)
- 40= Segment 8, Difference (S8D)
- 41= Segment 8, Center axis (S8C)

Description: Value which is used for statistic calculation.

13.3.2.3.6 Get_StatisticSignal (STATISTICSIGNAL)

Get the measured value which is used for statistic calculation.

Parameter: int32_t SA_StatisticSignal SA_StatisticSignal

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)
- 2= Diameter, Edge A (DA)
- 3= Diameter, Edge B (DB)
- 4= Diameter, Difference (DD)
- 5= Diameter, Center axis (DC)
- 6= Gap, Edge A (GA)
- 7= Gap, Edge B (GB)
- 8= Gap, Difference (GD)
- 9= Gap, Center axis (GC)
- 10= Segment 1, Edge A (S1A)
- 11= Segment 1, Edge B (S1B)
- 12= Segment 1, Difference (S1D)
- 13= Segment 1, Center axis (S1C)
- 14= Segment 2, Edge A (S2A)
- 15= Segment 2, Edge B (S2B)
- 16= Segment 2, Difference (S2D)
- 17= Segment 2, Center axis (S2C)
- 18= Segment 3, Edge A (S3A)
- 19= Segment 3, Edge B (S3B)
- 20= Segment 3, Difference (S3D)
- 21= Segment 3, Center axis (S3C)
- 22= Segment 4, Edge A (S4A)
- 23= Segment 4, Edge B (S4B)
- 24= Segment 4, Difference (S4D)
- 25= Segment 4, Center axis (S4C)
- 26= Segment 5, Edge A (S5A)
- 27= Segment 5, Edge B (S5B)
- 28= Segment 5, Difference (S5D)
- 29= Segment 5, Center axis (S5C)
- 30= Segment 6, Edge A (S6A)
- 31= Segment 6, Edge B (S6B)
- 32= Segment 6, Difference (S6D)
- 33= Segment 6, Center axis (S6C)
- 34= Segment 7, Edge A (S7A)
- 35= Segment 7, Edge B (S7B)
- 36= Segment 7, Difference (S7D)
- 37= Segment 7, Center axis (S7C)
- 38= Segment 8, Edge A (S8A)
- 39= Segment 8, Edge B (S8B)
- 40= Segment 8, Difference (S8D)
- 41= Segment 8, Center axis (S8C)

Description: Value which is used for statistic calculation.

13.3.2.3.7 Set_Statistic2Signal (STATISTIC2SIGNAL)

Set the measured value which is used for second statistic calculation.

Parameter: int32_t SP_Statistic2Signal

SP_Statistic2Signal

Direction: Down

Valid values:

- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)
- 2= Diameter, Edge A (DA)
- 3= Diameter, Edge B (DB)
- 4= Diameter, Difference (DD)
- 5= Diameter, Center axis (DC)
- 6= Gap, Edge A (GA)
- 7= Gap, Edge B (GB)
- 8= Gap, Difference (GD)
- 9= Gap, Center axis (GC)
- 10= Segment 1, Edge A (S1A)
- 11= Segment 1, Edge B (S1B)
- 12= Segment 1, Difference (S1D)
- 13= Segment 1, Center axis (S1C)
- 14= Segment 2, Edge A (S2A)
- 15= Segment 2, Edge B (S2B)
- 16= Segment 2, Difference (S2D)
- 17= Segment 2, Center axis (S2C)
- 18= Segment 3, Edge A (S3A)
- 19= Segment 3, Edge B (S3B)
- 20= Segment 3, Difference (S3D)
- 21= Segment 3, Center axis (S3C)
- 22= Segment 4, Edge A (S4A)
- 23= Segment 4, Edge B (S4B)
- 24= Segment 4, Difference (S4D)
- 25= Segment 4, Center axis (S4C)
- 26= Segment 5, Edge A (S5A)
- 27= Segment 5, Edge B (S5B)
- 28= Segment 5, Difference (S5D)
- 29= Segment 5, Center axis (S5C)
- 30= Segment 6, Edge A (S6A)
- 31= Segment 6, Edge B (S6B)
- 32= Segment 6, Difference (S6D)
- 33= Segment 6, Center axis (S6C)
- 34= Segment 7, Edge A (S7A)
- 35= Segment 7, Edge B (S7B)
- 36= Segment 7, Difference (S7D)
- 37= Segment 7, Center axis (S7C)
- 38= Segment 8, Edge A (S8A)
- 39= Segment 8, Edge B (S8B)
- 40= Segment 8, Difference (S8D)
- 41= Segment 8, Center axis (S8C)

Description: Value which is used for second statistic calculation.

13.3.2.3.8 Get_Statistic2Signal (STATISTIC2SIGNAL)

Get the measured value which is used for second statistic calculation.

Parameter: int32_t SA_Statistic2Signal

SA_Statistic2Signal

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Edge, Light-Dark (EHL)
- 1 = Edge, Dark-Light (ELH)
- 2 = Diameter, Edge A (DA)
- 3 = Diameter, Edge B (DB)
- 4 = Diameter, Difference (DD)
- 5 = Diameter, Center axis (DC)
- 6 = Gap, Edge A (GA)
- 7 = Gap, Edge B (GB)
- 8 = Gap, Difference (GD)
- 9 = Gap, Center axis (GC)
- 10 = Segment 1, Edge A (S1A)
- 11 = Segment 1, Edge B (S1B)
- 12 = Segment 1, Difference (S1D)
- 13 = Segment 1, Center axis (S1C)
- 14 = Segment 2, Edge A (S2A)
- 15 = Segment 2, Edge B (S2B)
- 16 = Segment 2, Difference (S2D)
- 17 = Segment 2, Center axis (S2C)
- 18 = Segment 3, Edge A (S3A)
- 19 = Segment 3, Edge B (S3B)
- 20 = Segment 3, Difference (S3D)
- 21 = Segment 3, Center axis (S3C)
- 22 = Segment 4, Edge A (S4A)
- 23 = Segment 4, Edge B (S4B)
- 24 = Segment 4, Difference (S4D)
- 25 = Segment 4, Center axis (S4C)
- 26 = Segment 5, Edge A (S5A)
- 27 = Segment 5, Edge B (S5B)
- 28 = Segment 5, Difference (S5D)
- 29 = Segment 5, Center axis (S5C)
- 30 = Segment 6, Edge A (S6A)
- 31 = Segment 6, Edge B (S6B)
- 32 = Segment 6, Difference (S6D)
- 33 = Segment 6, Center axis (S6C)
- 34 = Segment 7, Edge A (S7A)
- 35 = Segment 7, Edge B (S7B)
- 36 = Segment 7, Difference (S7D)
- 37 = Segment 7, Center axis (S7C)
- 38 = Segment 8, Edge A (S8A)
- 39 = Segment 8, Edge B (S8B)
- 40 = Segment 8, Difference (S8D)
- 41 = Segment 8, Center axis (S8C)

Description: Value which is used for second statistic calculation.

13.3.2.3.9 Set_StatisticDepth (STATISTICDEPTH)

Set the window size for floating statistic calculation.

Parameter: int32_t SP_StatisticDepth

SP_StatisticDepth

Direction: Down

Valid values:
Minimum: 2

Maximum: 2147483647 (INT32_MAX)

Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 8192). Value greater as 8192 calculates statistic over all values.

13.3.2.3.10 Get_StatisticDepth (STATISTICDEPTH)

Get the window size for floating statistic calculation.

Parameter: int32_t SA_StatisticDepth

SA_StatisticDepth

Direction: Up

Valid values:
Minimum: 2

Maximum: 2147483647 (INT32_MAX)

Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 8192). Value greater as 8192 calculates statistic over all values.

13.3.2.3.11 Reset_Statistic (RESETSTATISTIC)

Reset the statistic (min, max and peak to peak values).

13.3.2.3.12 Set_EdgeFilter<n> (EDGEFILTER<n>)

Set edge filter n ∈ {1..2}.

Only available at Option 201.

Parameter: int32_t SP_EdgeFilter<n>

SP_EdgeFilter<n>

Direction: Down

Valid values:

0= off

1= on

Description: Specifies if filter should be active.

Parameter: double SP_LowerBound<n>

SP_LowerBound<n>

Direction: Down

Valid values:
Minimum: 0.0

Maximum: measuring range

Unit: mm

Description: Lower bound for deactivation.

Parameter: double SP_UpperBound<n>

SP_UpperBound<n>

Direction: Down

Valid values:
Minimum: 0.0

Maximum: measuring range

Unit: mm

Description: Upper bound for activation.

Parameter: int32_t SP_IgnoreValues<n> SP_IgnoreValues<n>
Direction: Down
Valid values:
 Minimum: 1
 Maximum: 32768
Description: Number of values to ignore.

13.3.2.3.13 Get_EdgeFilter<n> (EDGEFILTER<n>)

Get edge filter n ∈ {1..2}.
 Only available at Option 201.

Parameter: int32_t SA_EdgeFilter<n> SA_EdgeFilter<n>
Direction: Up
Valid values:
 0 = off
 1 = on
Description: Specifies if filter should be active.

Parameter: double SA_LowerBound<n> SA_LowerBound<n>
Direction: Up
Valid values:
 Minimum: 0.0
 Maximum: measuring range
Unit: mm
Description: Lower bound for deactivation.

Parameter: double SA_UpperBound<n> SA_UpperBound<n>
Direction: Up
Valid values:
 Minimum: 0.0
 Maximum: measuring range
Unit: mm
Description: Upper bound for activation.

Parameter: int32_t SA_IgnoreValues<n> SA_IgnoreValues<n>
Direction: Up
Valid values:
 Minimum: 1
 Maximum: 32768
Description: Number of values to ignore.

13.3.2.3.14 Set_EdgeFilter<n>Signal (EDGEFILTER<n>SIGNAL)

Set signal for edge filter n ∈ {1..2}.
 Only available at Option 201.

Parameter: int32_t SP_EdgeFilter<n>Signal SP_EdgeFilter<n>Signal
Direction: Down
Valid values:
 0 = Edge, Light-Dark (EHL)
 1 = Edge, Dark-Light (ELH)

2= Diameter, Edge A (DA)
 3= Diameter, Edge B (DB)
 4= Diameter, Difference (DD)
 5= Diameter, Center axis (DC)
 6= Gap, Edge A (GA)
 7= Gap, Edge B (GB)
 8= Gap, Difference (GD)
 9= Gap, Center axis (GC)
 10= Segment 1, Edge A (S1A)
 11= Segment 1, Edge B (S1B)
 12= Segment 1, Difference (S1D)
 13= Segment 1, Center axis (S1C)
 14= Segment 2, Edge A (S2A)
 15= Segment 2, Edge B (S2B)
 16= Segment 2, Difference (S2D)
 17= Segment 2, Center axis (S2C)
 18= Segment 3, Edge A (S3A)
 19= Segment 3, Edge B (S3B)
 20= Segment 3, Difference (S3D)
 21= Segment 3, Center axis (S3C)
 22= Segment 4, Edge A (S4A)
 23= Segment 4, Edge B (S4B)
 24= Segment 4, Difference (S4D)
 25= Segment 4, Center axis (S4C)
 26= Segment 5, Edge A (S5A)
 27= Segment 5, Edge B (S5B)
 28= Segment 5, Difference (S5D)
 29= Segment 5, Center axis (S5C)
 30= Segment 6, Edge A (S6A)
 31= Segment 6, Edge B (S6B)
 32= Segment 6, Difference (S6D)
 33= Segment 6, Center axis (S6C)
 34= Segment 7, Edge A (S7A)
 35= Segment 7, Edge B (S7B)
 36= Segment 7, Difference (S7D)
 37= Segment 7, Center axis (S7C)
 38= Segment 8, Edge A (S8A)
 39= Segment 8, Edge B (S8B)
 40= Segment 8, Difference (S8D)
 41= Segment 8, Center axis (S8C)

Description: Value which is used for filter.

13.3.2.3.15 Get_EdgeFilter< n >Signal (EDGEFILTER< n >SIGNAL)

Get signal for edge filter $n \in \{1..2\}$.

Only available at Option 201.

Parameter: int32_t SA_EdgeFilter< n >Signal

SA_EdgeFilter< n >Signal

Direction: Up

Valid values:

- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)

2= Diameter, Edge A (DA)
 3= Diameter, Edge B (DB)
 4= Diameter, Difference (DD)
 5= Diameter, Center axis (DC)
 6= Gap, Edge A (GA)
 7= Gap, Edge B (GB)
 8= Gap, Difference (GD)
 9= Gap, Center axis (GC)
 10= Segment 1, Edge A (S1A)
 11= Segment 1, Edge B (S1B)
 12= Segment 1, Difference (S1D)
 13= Segment 1, Center axis (S1C)
 14= Segment 2, Edge A (S2A)
 15= Segment 2, Edge B (S2B)
 16= Segment 2, Difference (S2D)
 17= Segment 2, Center axis (S2C)
 18= Segment 3, Edge A (S3A)
 19= Segment 3, Edge B (S3B)
 20= Segment 3, Difference (S3D)
 21= Segment 3, Center axis (S3C)
 22= Segment 4, Edge A (S4A)
 23= Segment 4, Edge B (S4B)
 24= Segment 4, Difference (S4D)
 25= Segment 4, Center axis (S4C)
 26= Segment 5, Edge A (S5A)
 27= Segment 5, Edge B (S5B)
 28= Segment 5, Difference (S5D)
 29= Segment 5, Center axis (S5C)
 30= Segment 6, Edge A (S6A)
 31= Segment 6, Edge B (S6B)
 32= Segment 6, Difference (S6D)
 33= Segment 6, Center axis (S6C)
 34= Segment 7, Edge A (S7A)
 35= Segment 7, Edge B (S7B)
 36= Segment 7, Difference (S7D)
 37= Segment 7, Center axis (S7C)
 38= Segment 8, Edge A (S8A)
 39= Segment 8, Edge B (S8B)
 40= Segment 8, Difference (S8D)
 41= Segment 8, Center axis (S8C)

Description: Value which is used for filter.

13.3.2.3.16 Set_MasterSignal (MASTERSIGNAL)

Set the measured value which is used for mastering.

Parameter: int32_t SP_MasterSignal

SP_MasterSignal

Direction: Down

Valid values:

- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)
- 2= Diameter, Edge A (DA)

3= Diameter, Edge B (DB)
 4= Diameter, Difference (DD)
 5= Diameter, Center axis (DC)
 6= Gap, Edge A (GA)
 7= Gap, Edge B (GB)
 8= Gap, Difference (GD)
 9= Gap, Center axis (GC)
 10= Segment 1, Edge A (S1A)
 11= Segment 1, Edge B (S1B)
 12= Segment 1, Difference (S1D)
 13= Segment 1, Center axis (S1C)
 14= Segment 2, Edge A (S2A)
 15= Segment 2, Edge B (S2B)
 16= Segment 2, Difference (S2D)
 17= Segment 2, Center axis (S2C)
 18= Segment 3, Edge A (S3A)
 19= Segment 3, Edge B (S3B)
 20= Segment 3, Difference (S3D)
 21= Segment 3, Center axis (S3C)
 22= Segment 4, Edge A (S4A)
 23= Segment 4, Edge B (S4B)
 24= Segment 4, Difference (S4D)
 25= Segment 4, Center axis (S4C)
 26= Segment 5, Edge A (S5A)
 27= Segment 5, Edge B (S5B)
 28= Segment 5, Difference (S5D)
 29= Segment 5, Center axis (S5C)
 30= Segment 6, Edge A (S6A)
 31= Segment 6, Edge B (S6B)
 32= Segment 6, Difference (S6D)
 33= Segment 6, Center axis (S6C)
 34= Segment 7, Edge A (S7A)
 35= Segment 7, Edge B (S7B)
 36= Segment 7, Difference (S7D)
 37= Segment 7, Center axis (S7C)
 38= Segment 8, Edge A (S8A)
 39= Segment 8, Edge B (S8B)
 40= Segment 8, Difference (S8D)
 41= Segment 8, Center axis (S8C)

Description: Value which is used for mastering.

13.3.2.3.17 Get_MasterSignal (MASTERSIGNAL)

Get the measured value which is used for mastering.

Parameter: int32_t SA_MasterSignal

SA_MasterSignal

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)
- 2= Diameter, Edge A (DA)

3= Diameter, Edge B (DB)
 4= Diameter, Difference (DD)
 5= Diameter, Center axis (DC)
 6= Gap, Edge A (GA)
 7= Gap, Edge B (GB)
 8= Gap, Difference (GD)
 9= Gap, Center axis (GC)
 10= Segment 1, Edge A (S1A)
 11= Segment 1, Edge B (S1B)
 12= Segment 1, Difference (S1D)
 13= Segment 1, Center axis (S1C)
 14= Segment 2, Edge A (S2A)
 15= Segment 2, Edge B (S2B)
 16= Segment 2, Difference (S2D)
 17= Segment 2, Center axis (S2C)
 18= Segment 3, Edge A (S3A)
 19= Segment 3, Edge B (S3B)
 20= Segment 3, Difference (S3D)
 21= Segment 3, Center axis (S3C)
 22= Segment 4, Edge A (S4A)
 23= Segment 4, Edge B (S4B)
 24= Segment 4, Difference (S4D)
 25= Segment 4, Center axis (S4C)
 26= Segment 5, Edge A (S5A)
 27= Segment 5, Edge B (S5B)
 28= Segment 5, Difference (S5D)
 29= Segment 5, Center axis (S5C)
 30= Segment 6, Edge A (S6A)
 31= Segment 6, Edge B (S6B)
 32= Segment 6, Difference (S6D)
 33= Segment 6, Center axis (S6C)
 34= Segment 7, Edge A (S7A)
 35= Segment 7, Edge B (S7B)
 36= Segment 7, Difference (S7D)
 37= Segment 7, Center axis (S7C)
 38= Segment 8, Edge A (S8A)
 39= Segment 8, Edge B (S8B)
 40= Segment 8, Difference (S8D)
 41= Segment 8, Center axis (S8C)

Description: Value which is used for mastering.

13.3.2.3.18 Set_MasterValue (MASTERMV)

Set the master value.

Parameter: int32_t SP_Master

SP_Master

Direction: Down

Valid values:

- 0= no (NONE)
- 1= yes (MASTER)

Description: Specifies if mastering should be done or resetted.

Parameter: double SP_MasterValue SP_MasterValue
Direction: Down
Valid values:
 Minimum: -measuring range
 Maximum: +measuring range
Unit: mm
Description: Master value

13.3.2.3.19 Get_MasterValue (MASTERMV)

Get the master value.

Parameter: int32_t SA_Master SA_Master
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = no (NONE)
 1 = yes (MASTER)
Description: Specifies if mastering is active.

Parameter: double SA_MasterValue SA_MasterValue
Direction: Up
Valid values:
 Minimum: -measuring range
 Maximum: +measuring range
Unit: mm
Description: Master value

13.3.3 Data output

13.3.3.1 General

13.3.3.1.1 Set_DataOutInterface (OUTPUT)

Set the active interface for data output.
 At this command the controller may change output data automatically. If first bit of **IP_AutomaticMode** is set (1), [Get_AllParameters](#) (SP_Additional= 0) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DataOutInterface SP_DataOutInterface
Direction: Down
Valid values:
 0 = None
 1 = RS422
 2 = Ethernet
 3 = HTTP
Description: Active interface for data output.

13.3.3.1.2 Get_DataOutInterface (OUTPUT)

Get the active interface for data output.

Parameter: int32_t SA_DataOutInterface

SA_DataOutInterface

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= RS422
- 2= Ethernet
- 3= HTTP

Description: Active interface for data output.

13.3.3.1.3 Set_Resampling (OUTREDUCE)

Set resampling to reduce output data.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_Resampling

SP_Resampling

Direction: Down

Valid values:

- Minimum:** 1
Maximum: 150000

Description: Resampling value.

Parameter: int32_t SP_ResampleAnalog

SP_ResampleAnalog

Direction: Down

Valid values:

- 0= no
- 1= yes

Description: Specify if analog output should be resampled.

Parameter: int32_t SP_ResampleRS422

SP_ResampleRS422

Direction: Down

Valid values:

- 0= no
- 1= yes

Description: Specify if RS422 output should be resampled.

Parameter: int32_t SP_ResampleEthernet

SP_ResampleEthernet

Direction: Down

Valid values:

- 0= no
- 1= yes

Description: Specify if output over ethernet should be resampled.

13.3.3.1.4 Get_Resampling (OUTREDUCE)

Get resampling for reducing output data.

Parameter: int32_t SA_Resampling

SA_Resampling

Direction: Up

Valid values:

- Minimum:** 1
Maximum: 150000

Description: Resampling value.

13.3. Commands for ODC2520

Parameter: int32_t SA_ResampleAnalog	SA_ResampleAnalog
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Analog output is resampled.	
Parameter: int32_t SA_ResampleRS422	SA_ResampleRS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: RS422 output is resampled.	
Parameter: int32_t SA_ResampleEthernet	SA_ResampleEthernet
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Output over ethernet is resampled.	

13.3.3.1.5 Set_HoldLastValid (OUTHOLD)

Set the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SP_HoldLastValid	SP_HoldLastValid
Direction: Down	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	

13.3.3.1.6 Get_HoldLastValid (OUTHOLD)

Get the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SA_HoldLastValid	SA_HoldLastValid
Direction: Up	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	

13.3.3.2 Selected measurement values

13.3.3.2.1 Set_OutputEdgeLightDark_RS422 (OUTEDGEHL_RS422)

Set the edge light-dark to be output at RS422 interface (at measure mode edge light-dark).

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputEdgeLightDark_RS422

SP_OutputEdgeLightDark_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if edge light-dark is transmitted.

13.3.3.2.2 Get_OutputEdgeLightDark_RS422 (OUTEDGEHL_RS422)

Retrieve if the edge light-dark to be output at RS422 interface (at measure mode edge light-dark).

Parameter: int32_t SA_OutputEdgeLightDark_RS422

SA_OutputEdgeLightDark_-
RS422

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if edge light-dark is transmitted.

13.3.3.2.3 Set_OutputEdgeLightDark_ETH (OUTEDGEHL_ETH)

Set the edge light-dark to be output at ethernet interface (at measure mode edge light-dark).

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputEdgeLightDark_ETH

SP_OutputEdgeLightDark_-
ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if edge light-dark is transmitted.

13.3.3.2.4 Get_OutputEdgeLightDark_ETH (OUTEDGEHL_ETH)

Retrieve if the edge light-dark to be output at ethernet interface (at measure mode edge light-dark).

Parameter: int32_t SA_OutputEdgeLightDark_ETH

SA_OutputEdgeLightDark_-
ETH

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if edge light-dark is transmitted.

13.3.3.2.5 Set_OutputEdgeDarkLight_RS422 (OUTEDGEELH_RS422)

Set the edge dark-light to be output at RS422 interface (at measure mode edge dark-light).

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputEdgeDarkLight_RS422

SP_OutputEdgeDarkLight_-
RS422

Direction: Down

Valid values:

0 = no

1 = yes

Description: Specify if edge dark-light is transmitted.

13.3.3.2.6 Get_OutputEdgeDarkLight_RS422 (OUTEDGEELH_RS422)

Retrieve if the edge dark-light to be output at RS422 interface (at measure mode edge dark-light).

Parameter: int32_t SA_OutputEdgeDarkLight_RS422

SA_OutputEdgeDarkLight_-
RS422

Direction: Up

Valid values:

0 = no

1 = yes

Description: Specify if edge dark-light is transmitted.

13.3.3.2.7 Set_OutputEdgeDarkLight_ETH (OUTEDGEELH_ETH)

Set the edge dark-light to be output at ethernet interface (at measure mode edge dark-light).

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputEdgeDarkLight_ETH

SP_OutputEdgeDarkLight_-
ETH

Direction: Down

Valid values:

0 = no

1 = yes

Description: Specify if edge dark-light is transmitted.

13.3.3.2.8 Get_OutputEdgeDarkLight_ETH (OUTEDGEELH_ETH)

Retrieve if the edge dark-light to be output at ethernet interface (at measure mode edge dark-light).

Parameter: int32_t SA_OutputEdgeDarkLight_ETH

SA_OutputEdgeDarkLight_-
ETH

Direction: Up

Valid values:

0 = no

1 = yes

Description: Specify if edge dark-light is transmitted.

13.3.3.2.9 Set_OutputDiameter_RS422 (OUTDIA_RS422)

Set the diameter data to be output at RS422 interface (at measure mode diameter).
 For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputDiameterEdgeA_RS422	SP_OutputDiameterEdgeA_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A (first light-dark edge) is transmitted.	
Parameter: int32_t SP_OutputDiameterEdgeB_RS422	SP_OutputDiameterEdgeB_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B (last dark-light edge) is transmitted.	
Parameter: int32_t SP_OutputDiameterDifference_RS422	SP_OutputDiameterDiffer- ence_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) is transmitted.	
Parameter: int32_t SP_OutputDiameterCenterAxis_RS422	SP_OutputDiameterCenter- Axis_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) is transmitted.	
13.3.3.2.10 Get_OutputDiameter_RS422 (OUTDIA_RS422)	
Get the diameter data to be output at RS422 interface (at measure mode diameter).	
Parameter: int32_t SA_OutputDiameterEdgeA_RS422	SA_OutputDiameterEdgeA_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A (first light-dark edge) is transmitted.	
Parameter: int32_t SA_OutputDiameterEdgeB_RS422	SA_OutputDiameterEdgeB_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B (last dark-light edge) is transmitted.	

13.3. Commands for ODC2520

Parameter: int32_t SA_OutputDiameterDifference_RS422	SA_OutputDiameterDifference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) is transmitted.	
Parameter: int32_t SA_OutputDiameterCenterAxis_RS422	SA_OutputDiameterCenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) is transmitted.	

13.3.3.2.11 Set_OutputDiameter_ETH (OUTDIA_ETH)

Set the diameter data to be output at ethernet interface (at measure mode diameter). For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputDiameterEdgeA_ETH	SP_OutputDiameterEdgeA_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A (first light-dark edge) is transmitted.	
Parameter: int32_t SP_OutputDiameterEdgeB_ETH	SP_OutputDiameterEdgeB_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B (last dark-light edge) is transmitted.	
Parameter: int32_t SP_OutputDiameterDifference_ETH	SP_OutputDiameterDifference_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) is transmitted.	
Parameter: int32_t SP_OutputDiameterCenterAxis_ETH	SP_OutputDiameterCenterAxis_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) is transmitted.	

13.3.3.2.12 Get_OutputDiameter_ETH (OUTDIA_ETH)

Get the diameter data to be output at ethernet interface (at measure mode diameter).

Parameter: int32_t SA_OutputDiameterEdgeA_ETH	SA_OutputDiameterEdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A (first light-dark edge) is transmitted.	

Parameter: int32_t SA_OutputDiameterEdgeB_ETH	SA_OutputDiameterEdgeB_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B (last dark-light edge) is transmitted.	
Parameter: int32_t SA_OutputDiameterDifference_ETH	SA_OutputDiameterDiffer- ence_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) is transmitted.	
Parameter: int32_t SA_OutputDiameterCenterAxis_ETH	SA_OutputDiameterCenter- Axis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) is transmitted.	

13.3.3.2.13 Set_OutputGap_RS422 (OUTGAP_RS422)

Set the gap data to be output at RS422 interface (at measure mode gap).
 For this command an [Update...](#) and a [Reset...](#) meta command is available.

Parameter: int32_t SP_OutputGapEdgeA_RS422	SP_OutputGapEdgeA_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A (first dark-light edge) is transmitted.	
Parameter: int32_t SP_OutputGapEdgeB_RS422	SP_OutputGapEdgeB_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B (following edge after A) is transmitted.	
Parameter: int32_t SP_OutputGapDifference_RS422	SP_OutputGapDifference_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) is transmitted.	
Parameter: int32_t SP_OutputGapCenterAxis_RS422	SP_OutputGapCenterAxis_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) is transmitted.	

13.3.3.2.14 Get_OutputGap_RS422 (OUTGAP_RS422)

Get the gap data to be output at RS422 interface (at measure mode gap).

Parameter: int32_t SA_OutputGapEdgeA_RS422

SA_OutputGapEdgeA_RS422

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if edge A (first dark-light edge) is transmitted.

Parameter: int32_t SA_OutputGapEdgeB_RS422

SA_OutputGapEdgeB_RS422

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if edge B (following edge after A) is transmitted.

Parameter: int32_t SA_OutputGapDifference_RS422

SA_OutputGapDifference_-
RS422

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if difference (|A-B|) is transmitted.

Parameter: int32_t SA_OutputGapCenterAxis_RS422

SA_OutputGapCenterAxis_-
RS422

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if center axis ((A+B)/2) is transmitted.

13.3.3.2.15 Set_OutputGap_ETH (OUTGAP_ETH)

Set the gap data to be output at ethernet interface (at measure mode gap).

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputGapEdgeA_ETH

SP_OutputGapEdgeA_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if edge A (first dark-light edge) is transmitted.

Parameter: int32_t SP_OutputGapEdgeB_ETH

SP_OutputGapEdgeB_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if edge B (following edge after A) is transmitted.

Parameter: int32_t SP_OutputGapDifference_ETH	SP_OutputGapDifference_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) is transmitted.	
Parameter: int32_t SP_OutputGapCenterAxis_ETH	SP_OutputGapCenterAxis_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis (($A+B$)/2) is transmitted.	

13.3.3.2.16 Get_OutputGap_ETH (OUTGAP_ETH)

Get the gap data to be output at ethernet interface (at measure mode gap).

Parameter: int32_t SA_OutputGapEdgeA_ETH	SA_OutputGapEdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A (first dark-light edge) is transmitted.	
Parameter: int32_t SA_OutputGapEdgeB_ETH	SA_OutputGapEdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B (following edge after A) is transmitted.	
Parameter: int32_t SA_OutputGapDifference_ETH	SA_OutputGapDifference_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) is transmitted.	
Parameter: int32_t SA_OutputGapCenterAxis_ETH	SA_OutputGapCenterAxis_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis (($A+B$)/2) is transmitted.	

13.3.3.2.17 Set_OutputSegment_RS422 (OUTSEG_RS422)

Set the segment data to be output at RS422 interface (at measure mode segments). At this command the controller may change output data (additional to specified parameters). If first bit of [IP_AutomaticMode](#) is set (1), [Get_OutputSegment_RS422](#) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputSegment1EdgeA_RS422

SP_OutputSegment1EdgeA_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if edge A of segment 1 is transmitted.

Parameter: int32_t SP_OutputSegment1EdgeB_RS422

SP_OutputSegment1EdgeB_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if edge B of segment 1 is transmitted.

Parameter: int32_t SP_OutputSegment1Difference_RS422

SP_OutputSeg-
ment1Difference_RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if difference ($|A-B|$) of segment 1 is transmitted.

Parameter: int32_t SP_OutputSegment1CenterAxis_RS422

SP_OutputSeg-
ment1CenterAxis_RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if center axis ((A+B)/2) of segment 1 is transmitted.

Parameter: int32_t SP_OutputSegment2EdgeA_RS422

SP_OutputSegment2EdgeA_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if edge A of segment 2 is transmitted.

Parameter: int32_t SP_OutputSegment2EdgeB_RS422

SP_OutputSegment2EdgeB_-
RS422

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if edge B of segment 2 is transmitted.

Parameter: int32_t SP_OutputSegment2Difference_RS422	SP_OutputSegment2Difference_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 2 is transmitted.	
Parameter: int32_t SP_OutputSegment2CenterAxis_RS422	SP_OutputSegment2CenterAxis_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis (($A+B$)/2) of segment 2 is transmitted.	
Parameter: int32_t SP_OutputSegment3EdgeA_RS422	SP_OutputSegment3EdgeA_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 3 is transmitted.	
Parameter: int32_t SP_OutputSegment3EdgeB_RS422	SP_OutputSegment3EdgeB_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 3 is transmitted.	
Parameter: int32_t SP_OutputSegment3Difference_RS422	SP_OutputSegment3Difference_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 3 is transmitted.	
Parameter: int32_t SP_OutputSegment3CenterAxis_RS422	SP_OutputSegment3CenterAxis_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis (($A+B$)/2) of segment 3 is transmitted.	
Parameter: int32_t SP_OutputSegment4EdgeA_RS422	SP_OutputSegment4EdgeA_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 4 is transmitted.	
Parameter: int32_t SP_OutputSegment4EdgeB_RS422	SP_OutputSegment4EdgeB_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 4 is transmitted.	

Parameter: int32_t SP_OutputSegment4Difference_RS422	SP_OutputSegment4Difference_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 4 is transmitted.	
Parameter: int32_t SP_OutputSegment4CenterAxis_RS422	SP_OutputSegment4CenterAxis_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 4 is transmitted.	
Parameter: int32_t SP_OutputSegment5EdgeA_RS422	SP_OutputSegment5EdgeA_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 5 is transmitted.	
Parameter: int32_t SP_OutputSegment5EdgeB_RS422	SP_OutputSegment5EdgeB_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 5 is transmitted.	
Parameter: int32_t SP_OutputSegment5Difference_RS422	SP_OutputSegment5Difference_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 5 is transmitted.	
Parameter: int32_t SP_OutputSegment5CenterAxis_RS422	SP_OutputSegment5CenterAxis_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 5 is transmitted.	
Parameter: int32_t SP_OutputSegment6EdgeA_RS422	SP_OutputSegment6EdgeA_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 6 is transmitted.	
Parameter: int32_t SP_OutputSegment6EdgeB_RS422	SP_OutputSegment6EdgeB_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 6 is transmitted.	

Parameter: int32_t SP_OutputSegment6Difference_RS422	SP_OutputSegment6Difference_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 6 is transmitted.	
Parameter: int32_t SP_OutputSegment6CenterAxis_RS422	SP_OutputSegment6CenterAxis_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 6 is transmitted.	
Parameter: int32_t SP_OutputSegment7EdgeA_RS422	SP_OutputSegment7EdgeA_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 7 is transmitted.	
Parameter: int32_t SP_OutputSegment7EdgeB_RS422	SP_OutputSegment7EdgeB_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 7 is transmitted.	
Parameter: int32_t SP_OutputSegment7Difference_RS422	SP_OutputSegment7Difference_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 7 is transmitted.	
Parameter: int32_t SP_OutputSegment7CenterAxis_RS422	SP_OutputSegment7CenterAxis_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 7 is transmitted.	
Parameter: int32_t SP_OutputSegment8EdgeA_RS422	SP_OutputSegment8EdgeA_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 8 is transmitted.	
Parameter: int32_t SP_OutputSegment8EdgeB_RS422	SP_OutputSegment8EdgeB_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 8 is transmitted.	

13.3. Commands for ODC2520

Parameter: int32_t SP_OutputSegment8Difference_RS422	SP_OutputSegment8Difference_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 8 is transmitted.	
Parameter: int32_t SP_OutputSegment8CenterAxis_RS422	SP_OutputSegment8CenterAxis_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 8 is transmitted.	
13.3.3.2.18 Get_OutputSegment_RS422 (OUTSEG_RS422)	
Get the segment data to be output at RS422 interface (at measure mode segments).	
Parameter: int32_t SA_OutputSegment1EdgeA_RS422	SA_OutputSegment1EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment1EdgeB_RS422	SA_OutputSegment1EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment1Difference_RS422	SA_OutputSegment1Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment1CenterAxis_RS422	SA_OutputSegment1CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment2EdgeA_RS422	SA_OutputSegment2EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 2 is transmitted.	

13.3. Commands for ODC2520

Parameter: int32_t SA_OutputSegment2EdgeB_RS422	SA_OutputSegment2EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment2Difference_RS422	SA_OutputSegment2Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment2CenterAxis_RS422	SA_OutputSegment2CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment3EdgeA_RS422	SA_OutputSegment3EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment3EdgeB_RS422	SA_OutputSegment3EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment3Difference_RS422	SA_OutputSegment3Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment3CenterAxis_RS422	SA_OutputSegment3CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment4EdgeA_RS422	SA_OutputSegment4EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 4 is transmitted.	

13.3. Commands for ODC2520

Parameter: int32_t SA_OutputSegment4EdgeB_RS422	SA_OutputSegment4EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment4Difference_RS422	SA_OutputSegment4Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment4CenterAxis_RS422	SA_OutputSegment4CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment5EdgeA_RS422	SA_OutputSegment5EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment5EdgeB_RS422	SA_OutputSegment5EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment5Difference_RS422	SA_OutputSegment5Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment5CenterAxis_RS422	SA_OutputSegment5CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment6EdgeA_RS422	SA_OutputSegment6EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 6 is transmitted.	

13.3. Commands for ODC2520

Parameter: int32_t SA_OutputSegment6EdgeB_RS422	SA_OutputSegment6EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment6Difference_RS422	SA_OutputSegment6Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment6CenterAxis_RS422	SA_OutputSegment6CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment7EdgeA_RS422	SA_OutputSegment7EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment7EdgeB_RS422	SA_OutputSegment7EdgeB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment7Difference_RS422	SA_OutputSegment7Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment7CenterAxis_RS422	SA_OutputSegment7CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment8EdgeA_RS422	SA_OutputSegment8EdgeA_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 8 is transmitted.	

Parameter: int32_t SA_OutputSegment8EdgeB_RS422	SA_OutputSegment8EdgeB_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment8Difference_RS422	SA_OutputSeg- ment8Difference_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment8CenterAxis_RS422	SA_OutputSeg- ment8CenterAxis_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 8 is transmitted.	

13.3.3.2.19 Set_OutputSegment_ETH (OUTSEG_ETH)

Set the segment data to be output at ethernet interface (at measure mode segments). In measure mode segment and data output interface Ethernet the controller may change output data (additional to specified parameters). Next call to [DataAvail](#) or [TransferData](#) will return a warning. In this case call [Get_TransmittedDataInfo](#) to retrieve new setting.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputSegment1EdgeA_ETH	SP_OutputSegment1EdgeA_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 1 is transmitted.	
Parameter: int32_t SP_OutputSegment1EdgeB_ETH	SP_OutputSegment1EdgeB_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 1 is transmitted.	
Parameter: int32_t SP_OutputSegment1Difference_ETH	SP_OutputSeg- ment1Difference_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 1 is transmitted.	

Parameter: int32_t SP_OutputSegment1CenterAxis_ETH	SP_OutputSegment1CenterAxis_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 1 is transmitted.	
Parameter: int32_t SP_OutputSegment2EdgeA_ETH	SP_OutputSegment2EdgeA_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 2 is transmitted.	
Parameter: int32_t SP_OutputSegment2EdgeB_ETH	SP_OutputSegment2EdgeB_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 2 is transmitted.	
Parameter: int32_t SP_OutputSegment2Difference_ETH	SP_OutputSegment2Difference_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 2 is transmitted.	
Parameter: int32_t SP_OutputSegment2CenterAxis_ETH	SP_OutputSegment2CenterAxis_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 2 is transmitted.	
Parameter: int32_t SP_OutputSegment3EdgeA_ETH	SP_OutputSegment3EdgeA_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 3 is transmitted.	
Parameter: int32_t SP_OutputSegment3EdgeB_ETH	SP_OutputSegment3EdgeB_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 3 is transmitted.	
Parameter: int32_t SP_OutputSegment3Difference_ETH	SP_OutputSegment3Difference_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 3 is transmitted.	

Parameter: int32_t SP_OutputSegment3CenterAxis_ETH	SP_OutputSegment3CenterAxis_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 3 is transmitted.	
Parameter: int32_t SP_OutputSegment4EdgeA_ETH	SP_OutputSegment4EdgeA_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 4 is transmitted.	
Parameter: int32_t SP_OutputSegment4EdgeB_ETH	SP_OutputSegment4EdgeB_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 4 is transmitted.	
Parameter: int32_t SP_OutputSegment4Difference_ETH	SP_OutputSegment4Difference_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 4 is transmitted.	
Parameter: int32_t SP_OutputSegment4CenterAxis_ETH	SP_OutputSegment4CenterAxis_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 4 is transmitted.	
Parameter: int32_t SP_OutputSegment5EdgeA_ETH	SP_OutputSegment5EdgeA_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 5 is transmitted.	
Parameter: int32_t SP_OutputSegment5EdgeB_ETH	SP_OutputSegment5EdgeB_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 5 is transmitted.	
Parameter: int32_t SP_OutputSegment5Difference_ETH	SP_OutputSegment5Difference_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 5 is transmitted.	

Parameter: int32_t SP_OutputSegment5CenterAxis_ETH	SP_OutputSegment5CenterAxis_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 5 is transmitted.	
Parameter: int32_t SP_OutputSegment6EdgeA_ETH	SP_OutputSegment6EdgeA_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 6 is transmitted.	
Parameter: int32_t SP_OutputSegment6EdgeB_ETH	SP_OutputSegment6EdgeB_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 6 is transmitted.	
Parameter: int32_t SP_OutputSegment6Difference_ETH	SP_OutputSegment6Difference_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 6 is transmitted.	
Parameter: int32_t SP_OutputSegment6CenterAxis_ETH	SP_OutputSegment6CenterAxis_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 6 is transmitted.	
Parameter: int32_t SP_OutputSegment7EdgeA_ETH	SP_OutputSegment7EdgeA_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 7 is transmitted.	
Parameter: int32_t SP_OutputSegment7EdgeB_ETH	SP_OutputSegment7EdgeB_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 7 is transmitted.	
Parameter: int32_t SP_OutputSegment7Difference_ETH	SP_OutputSegment7Difference_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 7 is transmitted.	

Parameter: int32_t SP_OutputSegment7CenterAxis_ETH	SP_OutputSegment7CenterAxis_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 7 is transmitted.	
Parameter: int32_t SP_OutputSegment8EdgeA_ETH	SP_OutputSegment8EdgeA_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 8 is transmitted.	
Parameter: int32_t SP_OutputSegment8EdgeB_ETH	SP_OutputSegment8EdgeB_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 8 is transmitted.	
Parameter: int32_t SP_OutputSegment8Difference_ETH	SP_OutputSegment8Difference_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if difference (A-B) of segment 8 is transmitted.	
Parameter: int32_t SP_OutputSegment8CenterAxis_ETH	SP_OutputSegment8CenterAxis_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 8 is transmitted.	

13.3.3.2.20 Get_OutputSegment_ETH (OUTSEG_ETH)

Get the segment data to be output at ethernet interface (at measure mode segments).

Parameter: int32_t SA_OutputSegment1EdgeA_ETH	SA_OutputSegment1EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment1EdgeB_ETH	SA_OutputSegment1EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 1 is transmitted.	

Parameter: int32_t SA_OutputSegment1Difference_ETH	SA_OutputSegment1Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment1CenterAxis_ETH	SA_OutputSegment1CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis (($A+B$)/2) of segment 1 is transmitted.	
Parameter: int32_t SA_OutputSegment2EdgeA_ETH	SA_OutputSegment2EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment2EdgeB_ETH	SA_OutputSegment2EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment2Difference_ETH	SA_OutputSegment2Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment2CenterAxis_ETH	SA_OutputSegment2CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis (($A+B$)/2) of segment 2 is transmitted.	
Parameter: int32_t SA_OutputSegment3EdgeA_ETH	SA_OutputSegment3EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment3EdgeB_ETH	SA_OutputSegment3EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 3 is transmitted.	

Parameter: int32_t SA_OutputSegment3Difference_ETH	SA_OutputSegment3Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment3CenterAxis_ETH	SA_OutputSegment3CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis (($A+B$)/2) of segment 3 is transmitted.	
Parameter: int32_t SA_OutputSegment4EdgeA_ETH	SA_OutputSegment4EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment4EdgeB_ETH	SA_OutputSegment4EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment4Difference_ETH	SA_OutputSegment4Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment4CenterAxis_ETH	SA_OutputSegment4CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis (($A+B$)/2) of segment 4 is transmitted.	
Parameter: int32_t SA_OutputSegment5EdgeA_ETH	SA_OutputSegment5EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment5EdgeB_ETH	SA_OutputSegment5EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 5 is transmitted.	

Parameter: int32_t SA_OutputSegment5Difference_ETH	SA_OutputSegment5Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment5CenterAxis_ETH	SA_OutputSegment5CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis (($A+B$)/2) of segment 5 is transmitted.	
Parameter: int32_t SA_OutputSegment6EdgeA_ETH	SA_OutputSegment6EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment6EdgeB_ETH	SA_OutputSegment6EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment6Difference_ETH	SA_OutputSegment6Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment6CenterAxis_ETH	SA_OutputSegment6CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis (($A+B$)/2) of segment 6 is transmitted.	
Parameter: int32_t SA_OutputSegment7EdgeA_ETH	SA_OutputSegment7EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment7EdgeB_ETH	SA_OutputSegment7EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 7 is transmitted.	

Parameter: int32_t SA_OutputSegment7Difference_ETH	SA_OutputSegment7Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment7CenterAxis_ETH	SA_OutputSegment7CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 7 is transmitted.	
Parameter: int32_t SA_OutputSegment8EdgeA_ETH	SA_OutputSegment8EdgeA_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge A of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment8EdgeB_ETH	SA_OutputSegment8EdgeB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if edge B of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment8Difference_ETH	SA_OutputSegment8Difference_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if difference ($ A-B $) of segment 8 is transmitted.	
Parameter: int32_t SA_OutputSegment8CenterAxis_ETH	SA_OutputSegment8CenterAxis_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if center axis ((A+B)/2) of segment 8 is transmitted.	

13.3.3.2.21 Set_OutputStatistic_RS422 (OUTSTATISTIC_RS422)

Set the statistic data to be output at RS422 interface.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputStatisticMin_RS422	SP_OutputStatisticMin_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic min value is transmitted.	

Parameter: int32_t SP_OutputStatisticMax_RS422	SP_OutputStatisticMax_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic max value is transmitted.	
Parameter: int32_t SP_OutputStatisticPeak2Peak_RS422	SP_OutputStatistic- Peak2Peak_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic peak to peak value is transmitted.	
Parameter: int32_t SP_OutputStatistic2Min_RS422	SP_OutputStatistic2Min_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic min value is transmitted.	
Parameter: int32_t SP_OutputStatistic2Max_RS422	SP_OutputStatistic2Max_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic max value is transmitted.	
Parameter: int32_t SP_OutputStatistic2Peak2Peak_RS422	SP_OutputStatis- tic2Peak2Peak_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic peak to peak value is transmitted.	

13.3.3.2.22 Get_OutputStatistic_RS422 (OUTSTATISTIC_RS422)

Get the statistic data to be output at RS422 interface.

Parameter: int32_t SA_OutputStatisticMin_RS422	SA_OutputStatisticMin_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic min value is transmitted.	
Parameter: int32_t SA_OutputStatisticMax_RS422	SA_OutputStatisticMax_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic max value is transmitted.	

Parameter: int32_t SA_OutputStatisticPeak2Peak_RS422	SA_OutputStatisticPeak2Peak_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic peak to peak value is transmitted.	
Parameter: int32_t SA_OutputStatistic2Min_RS422	SA_OutputStatistic2Min_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic min value is transmitted.	
Parameter: int32_t SA_OutputStatistic2Max_RS422	SA_OutputStatistic2Max_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic max value is transmitted.	
Parameter: int32_t SA_OutputStatistic2Peak2Peak_RS422	SA_OutputStatistic2Peak2Peak_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic peak to peak value is transmitted.	

13.3.3.2.23 Set_OutputStatistic_ETH (OUTSTATISTIC_ETH)

Set the statistic data to be output at ethernet interface.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputStatisticMin_ETH	SP_OutputStatisticMin_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic min value is transmitted.	
Parameter: int32_t SP_OutputStatisticMax_ETH	SP_OutputStatisticMax_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic max value is transmitted.	
Parameter: int32_t SP_OutputStatisticPeak2Peak_ETH	SP_OutputStatisticPeak2Peak_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic peak to peak value is transmitted.	

Parameter: int32_t SP_OutputStatistic2Min_ETH	SP_OutputStatistic2Min_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic min value is transmitted.	
Parameter: int32_t SP_OutputStatistic2Max_ETH	SP_OutputStatistic2Max_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic max value is transmitted.	
Parameter: int32_t SP_OutputStatistic2Peak2Peak_ETH	SP_OutputStatis- tic2Peak2Peak_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic peak to peak value is transmitted.	

13.3.3.2.24 Get_OutputStatistic_ETH (OUTSTATIC_ETH)

Get the statistic data to be output at ethernet interface.

Parameter: int32_t SA_OutputStatisticMin_ETH	SA_OutputStatisticMin_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic min value is transmitted.	
Parameter: int32_t SA_OutputStatisticMax_ETH	SA_OutputStatisticMax_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic max value is transmitted.	
Parameter: int32_t SA_OutputStatisticPeak2Peak_ETH	SA_OutputStatistic- Peak2Peak_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statictic peak to peak value is transmitted.	
Parameter: int32_t SA_OutputStatistic2Min_ETH	SA_OutputStatistic2Min_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second statictic min value is transmitted.	

Parameter: int32_t SA_OutputStatistic2Max_ETH	SA_OutputStatistic2Max_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second staticic max value is transmitted.	
Parameter: int32_t SA_OutputStatistic2Peak2Peak_ETH	SA_OutputStatis- tic2Peak2Peak_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if second staticic peak to peak value is transmitted.	

13.3.3.2.25 Set_OutputAdditional_RS422 (OUTADD_RS422)

Set the additional data to be output at RS422 interface.
 For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputAdditionalNbrEdges_RS422	SP_OutputAddition- alNbrEdges_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if number of edges is transmitted.	
Parameter: int32_t SP_OutputAdditionalNbrPins_RS422	SP_OutputAdditionalNbr- Pins_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if number of pins (dark areas) is transmitted.	
Parameter: int32_t SP_OutputAdditionalNbrGaps_RS422	SP_OutputAdditionalNbr- Gaps_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if number of gaps (light areas) is transmitted.	
Parameter: int32_t SP_OutputAdditionalCounter_RS422	SP_OutputAdditional- Counter_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SP_OutputAdditionalTimestamp_RS422	SP_OutputAdditionalTimes- tamp_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	

13.3. Commands for ODC2520

Parameter: int32_t SP_OutputAdditionalState_RS422
Direction: Down
Valid values:
 0= no
 1 = yes
Description: Specify if state is transmitted.

13.3.3.2.26 Get_OutputAdditional_RS422 (OUTADD_RS422)

Get the additional data to be output at RS422 interface.

Parameter: int32_t SA_OutputAdditionalNbrEdges_RS422	SA_OutputAdditionalNbrEdges_RS422
Direction: Up	
Valid values:	
0 = no	
1 = yes	
Description:	Specify if number of edges is transmitted.
Parameter: int32_t SA_OutputAdditionalNbrPins_RS422	SA_OutputAdditionalNbrPins_RS422
Direction: Up	
Valid values:	
0 = no	
1 = yes	

Parameter: int32_t SA_OutputAdditionalNbrGaps_RS422
Direction: Up
Valid values:
 0 = no
 1 = yes
Description: Specify if number of gaps (light areas) is transmitted

Parameter: int32_t SA_OutputAdditionalCounter_RS422
Direction: Up
Valid values:
 0 = no
 1 = yes
Description: Specify if counter is transmitted.

Parameter: int32_t SA_OutputAdditionalTimestamp_RS422
Direction: Up
Valid values:
 0 = no
 1 = yes
Description: Specify if timestamp is transmitted.

Parameter: int32_t SA_OutputAdditionalState_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if state is transmitted.

13.3.3.2.27 Set_OutputAdditional_ETH (OUTADD_ETH)

Set the additional data to be output at ethernet interface.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputAdditionalNbrEdges_ETH

SP_OutputAdditionalNbrEdges_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if number of edges is transmitted.

Parameter: int32_t SP_OutputAdditionalNbrPins_ETH

SP_OutputAdditionalNbrPins_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if number of pins (dark areas) is transmitted.

Parameter: int32_t SP_OutputAdditionalNbrGaps_ETH

SP_OutputAdditionalNbrGaps_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if number of gaps (light areas) is transmitted.

Parameter: int32_t SP_OutputAdditionalCounter_ETH

SP_OutputAdditionalCounter_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if counter is transmitted.

Parameter: int32_t SP_OutputAdditionalTimestamp_ETH

SP_OutputAdditionalTimestamp_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if timestamp is transmitted.

Parameter: int32_t SP_OutputAdditionalState_ETH

SP_OutputAdditionalState_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if state is transmitted.

13.3.3.2.28 Get_OutputAdditional_ETH (OUTADD_ETH)

Get the additional data to be output at ethernet interface.

Parameter: int32_t SA_OutputAdditionalNbrEdges_ETH

SA_OutputAdditionalNbrEdges_ETH

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if number of edges is transmitted.

13.3. Commands for ODC2520

Parameter: int32_t SA_OutputAdditionalNbrPins_ETH	SA_OutputAdditionalNbrPins_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if number of pins (dark areas) is transmitted.	
Parameter: int32_t SA_OutputAdditionalNbrGaps_ETH	SA_OutputAdditionalNbrGaps_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if number of gaps (light areas) is transmitted.	
Parameter: int32_t SA_OutputAdditionalCounter_ETH	SA_OutputAdditionalCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputAdditionalTimestamp_ETH	SA_OutputAdditionalTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputAdditionalState_ETH	SA_OutputAdditionalState_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if state is transmitted.	

13.3.3.2.29 Set_OutputVideo_ETH (OUTVID_ETH)

Set the video signal to be output at ethernet interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputVideoRaw_ETH	SP_OutputVideoRaw_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal is transmitted.	
Parameter: int32_t SP_OutputVideoLight_ETH	SP_OutputVideoLight_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if light preprocessed video signal is transmitted.	

13.3. Commands for ODC2520

Parameter: int32_t SP_OutputVideoLightTable_ETH	SP_OutputVideoLightTable_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if light table is transmitted.	
Parameter: int32_t SP_OutputVideoThreshold_ETH	SP_OutputVideoThreshold_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if video threshold table is transmitted.	
Parameter: int32_t IP_TimerResolution	IP_TimerResolution
Direction: Down	
Unit: ms	
Valid values:	
-1= Do not set timer resolution.	
0= Use greatest possible accuracy.	
1..255= Resolution in milliseconds.	
Unit: ms	
Default: 0 if any video signal is enabled, otherwise -1	
Description: This parameter is necessary at video stream mode. Unless MEDAQLib waits for new video signals (using Sleep API function) and the default resolution for this function is 15 milli seconds there may be a time jitter at processing video signals (e.g. 15, 0, 0, 0 ..., 0, 15, 0, 0, ...). So this parameter changes the Windows timer resolution (for Windows scheduler, set by timeBeginPeriod). If it is not resetted manually when switching off all video signals (set to 0), it is automatically resetted at CloseSensor.	

13.3.3.2.30 Get_OutputVideo_ETH (OUTVID_ETH)

Get the video signal to be output at ethernet interface.

Parameter: int32_t SA_OutputVideoRaw_ETH	SA_OutputVideoRaw_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if raw video signal is transmitted.	
Parameter: int32_t SA_OutputVideoLight_ETH	SA_OutputVideoLight_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if light preprocessed video signal is transmitted.	

Parameter: int32_t SA_OutputVideoLightTable_ETH	SA_OutputVideoLightTable_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if light table is transmitted.
Parameter: int32_t SA_OutputVideoThreshold_ETH	SA_OutputVideoThreshold_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if video threshold table is transmitted.

13.3.3.3 Switching outputs

13.3.3.3.1 Set_ErrorOutput<n> (ERROROUT<n>)

Set condition to be used to set error output n ∈ {1..2}.

Parameter: int32_t SP_ErrorOutput<n>	SP_ErrorOutput<n>
Direction: Down	
Valid values:	
0= None	
1= Wrong number of edges (ER1)	
2= Measurement error (ER2)	
3= Below low limit (LI1)	
4= Above high limit (LI2)	
5= Out of limits (LI12)	
Description:	Condition for error output.

13.3.3.3.2 Get_ErrorOutput<n> (ERROROUT<n>)

Get condition to be used to set error output n ∈ {1..2}.

Parameter: int32_t SA_ErrorOutput<n>	SA_ErrorOutput<n>
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Wrong number of edges (ER1)	
2= Measurement error (ER2)	
3= Below low limit (LI1)	
4= Above high limit (LI2)	
5= Out of limits (LI12)	
Description:	Condition for error output.

13.3.3.3.3 Set_ErrorLimit (ERRORLIMIT)

Set the error limits.

At this command the controller may change output data automatically. Next call to [DataAvail](#) or [TransferData](#) will return a warning. In this case call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DataSource SP_DataSource

Direction: Down

Valid values:

- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)
- 2= Diameter, Edge A (DA)
- 3= Diameter, Edge B (DB)
- 4= Diameter, Difference (DD)
- 5= Diameter, Center axis (DC)
- 6= Gap, Edge A (GA)
- 7= Gap, Edge B (GB)
- 8= Gap, Difference (GD)
- 9= Gap, Center axis (GC)
- 10= Segment 1, Edge A (S1A)
- 11= Segment 1, Edge B (S1B)
- 12= Segment 1, Difference (S1D)
- 13= Segment 1, Center axis (S1C)
- 14= Segment 2, Edge A (S2A)
- 15= Segment 2, Edge B (S2B)
- 16= Segment 2, Difference (S2D)
- 17= Segment 2, Center axis(S2C)
- 18= Segment 3, Edge A (S3A)
- 19= Segment 3, Edge B (S3B)
- 20= Segment 3, Difference (S3D)
- 21= Segment 3, Center axis (S3C)
- 22= Segment 4, Edge A (S4A)
- 23= Segment 4, Edge B (S4B)
- 24= Segment 4, Difference (S4D)
- 25= Segment 4, Center axis (S4C)
- 26= Segment 5, Edge A (S5A)
- 27= Segment 5, Edge B (S5B)
- 28= Segment 5, Difference (S5D)
- 29= Segment 5, Center axis (S5C)
- 30= Segment 6, Edge A (S6A)
- 31= Segment 6, Edge B (S6B)
- 32= Segment 6, Difference (S6D)
- 33= Segment 6, Center axis (S6C)
- 34= Segment 7, Edge A (S7A)
- 35= Segment 7, Edge B (S7B)
- 36= Segment 7, Difference (S7D)
- 37= Segment 7, Center axis (S7C)
- 38= Segment 8, Edge A (S8A)
- 39= Segment 8, Edge B (S8B)
- 40= Segment 8, Difference (S8D)
- 41= Segment 8, Center axis (S8C)

Description: Data source to be checked.

Parameter: double SP_LowerLimit SP_LowerLimit

Direction: Down

Valid values:

Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)

Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)

Unit: mm

Description: Lower limit.

Parameter: double SP_UpperLimit SP_UpperLimit

Direction: Down

Valid values:

Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)

Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)

Unit: mm

Description: Upper limit.

13.3.3.4 Get_ErrorLimit (ERRORLIMIT)

Get the error limits.

Parameter: int32_t SA_DataSource SA_DataSource

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)
- 2= Diameter, Edge A (DA)
- 3= Diameter, Edge B (DB)
- 4= Diameter, Difference (DD)
- 5= Diameter, Center axis (DC)
- 6= Gap, Edge A (GA)
- 7= Gap, Edge B (GB)
- 8= Gap, Difference (GD)
- 9= Gap, Center axis (GC)
- 10= Segment 1, Edge A (S1A)
- 11= Segment 1, Edge B (S1B)
- 12= Segment 1, Difference (S1D)
- 13= Segment 1, Center axis (S1C)
- 14= Segment 2, Edge A (S2A)
- 15= Segment 2, Edge B (S2B)
- 16= Segment 2, Difference (S2D)
- 17= Segment 2, Center axis (S2C)
- 18= Segment 3, Edge A (S3A)
- 19= Segment 3, Edge B (S3B)
- 20= Segment 3, Difference (S3D)
- 21= Segment 3, Center axis (S3C)
- 22= Segment 4, Edge A (S4A)
- 23= Segment 4, Edge B (S4B)
- 24= Segment 4, Difference (S4D)
- 25= Segment 4, Center axis (S4C)
- 26= Segment 5, Edge A (S5A)
- 27= Segment 5, Edge B (S5B)

28= Segment 5, Difference (S5D)
 29= Segment 5, Center axis (S5C)
 30= Segment 6, Edge A (S6A)
 31= Segment 6, Edge B (S6B)
 32= Segment 6, Difference (S6D)
 33= Segment 6, Center axis (S6C)
 34= Segment 7, Edge A (S7A)
 35= Segment 7, Edge B (S7B)
 36= Segment 7, Difference (S7D)
 37= Segment 7, Center axis (S7C)
 38= Segment 8, Edge A (S8A)
 39= Segment 8, Edge B (S8B)
 40= Segment 8, Difference (S8D)
 41= Segment 8, Center axis (S8C)

Description: Data source to be checked.

Parameter: double SA_LowerLimit

SA_LowerLimit

Direction: Up

Valid values:

Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)

Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)

Unit: mm

Description: Lower limit.

Parameter: double SA_UpperLimit

SA_UpperLimit

Direction: Up

Valid values:

Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)

Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)

Unit: mm

Description: Upper limit.

13.3.3.5 Set_ErrorOut< n > HoldTime (ERROROUTHOLD< n >)

Set the minimum hold period of the threshold function (Set_ErrorLimit) for error output $n \in \{1..2\}$.

Parameter: int32_t SP_ErrorOut< n > HoldTime

SP_ErrorOut< n > HoldTime

Direction: Down

Valid values:

Minimum: 0

Maximum: 10000

Unit: ms

Description: Minimum hold period. 0 means no value to hold (NONE).

13.3.3.6 Get_ErrorOut< n > HoldTime (ERROROUTHOLD< n >)

Get the minimum hold period of the threshold function (Set_ErrorLimit) for error output $n \in \{1..2\}$.

Parameter: int32_t SA_ErrorOut< n > HoldTime

SA_ErrorOut< n > HoldTime

Direction: Up

Valid values:

Minimum: 0

Maximum: 10000

Unit: ms

Description: Minimum hold period. 0 means no value to hold (NONE).

13.3.3.3.7 Set_ErrorLevelOut<n> (ERRORLEVELOUT<n>)

Set level of error output n ∈ {1..2} on error.

Parameter: int32_t SP_ErrorLevelOut<n>

SP_ErrorLevelOut<n>

Direction: Down

Valid values:

- 0= NPN
- 1= PNP
- 2= Push-Pull (PUSHPULL)
- 3= Push-Pull negated (PUSHPULLNEG)

Description: Error level.

13.3.3.8 Get_ErrorLevelOut<n> (ERRORLEVELOUT<n>)

Get level of error output n ∈ {1..2} on error.

Parameter: int32_t SA_ErrorLevelOut<n>

SA_ErrorLevelOut<n>

Direction: Up

Valid values:

- 0= NPN
- 1= PNP
- 2= Push-Pull (PUSHPULL)
- 3= Push-Pull negated (PUSHPULLNEG)

Description: Error level.

13.3.4 Analog output

13.3.4.1 Set_AnalogOutput (ANALOGOUT)

Set the data to be used for analog output.

Parameter: int32_t SP_AnalogOutput

SP_AnalogOutput

Direction: Down

Valid values:

- 0= Edge, Light-Dark (EHL)
- 1= Edge, Dark-Light (ELH)
- 2= Diameter, Edge A (DA)
- 3= Diameter, Edge B (DB)
- 4= Diameter, Difference (DD)
- 5= Diameter, Center axis (DC)
- 6= Gap, Edge A (GA)
- 7= Gap, Edge B (GB)
- 8= Gap, Difference (GD)
- 9= Gap, Center axis (GC)
- 10= Segment 1, Edge A (S1A)
- 11= Segment 1, Edge B (S1B)
- 12= Segment 1, Difference (S1D)
- 13= Segment 1, Center axis (S1C)
- 14= Segment 2, Edge A (S2A)
- 15= Segment 2, Edge B (S2B)

16= Segment 2, Difference (S2D)
 17= Segment 2, Center axis(S2C)
 18= Segment 3, Edge A (S3A)
 19= Segment 3, Edge B (S3B)
 20= Segment 3, Difference (S3D)
 21= Segment 3, Center axis (S3C)
 22= Segment 4, Edge A (S4A)
 23= Segment 4, Edge B (S4B)
 24= Segment 4, Difference (S4D)
 25= Segment 4, Center axis (S4C)
 26= Segment 5, Edge A (S5A)
 27= Segment 5, Edge B (S5B)
 28= Segment 5, Difference (S5D)
 29= Segment 5, Center axis (S5C)
 30= Segment 6, Edge A (S6A)
 31= Segment 6, Edge B (S6B)
 32= Segment 6, Difference (S6D)
 33= Segment 6, Center axis (S6C)
 34= Segment 7, Edge A (S7A)
 35= Segment 7, Edge B (S7B)
 36= Segment 7, Difference (S7D)
 37= Segment 7, Center axis (S7C)
 38= Segment 8, Edge A (S8A)
 39= Segment 8, Edge B (S8B)
 40= Segment 8, Difference (S8D)
 41= Segment 8, Center axis (S8C)

Description: Data to be used for analog output.

13.3.3.4.2 Get_AnalogOutput (ANALOGOUT)

Get the data to be used for analog output.

Parameter: int32_t SA_AnalogOutput

SA_AnalogOutput

Direction: Up

Valid values:

0= Edge, Light-Dark (EHL)
 1= Edge, Dark-Light (ELH)
 2= Diameter, Edge A (DA)
 3= Diameter, Edge B (DB)
 4= Diameter, Difference (DD)
 5= Diameter, Center axis (DC)
 6= Gap, Edge A (GA)
 7= Gap, Edge B (GB)
 8= Gap, Difference (GD)
 9= Gap, Center axis (GC)
 10= Segment 1, Edge A (S1A)
 11= Segment 1, Edge B (S1B)
 12= Segment 1, Difference (S1D)
 13= Segment 1, Center axis (S1C)
 14= Segment 2, Edge A (S2A)
 15= Segment 2, Edge B (S2B)
 16= Segment 2, Difference (S2D)

13.3. Commands for ODC2520

17= Segment 2, Center axis(S2C)
 18= Segment 3, Edge A (S3A)
 19= Segment 3, Edge B (S3B)
 20= Segment 3, Difference (S3D)
 21= Segment 3, Center axis (S3C)
 22= Segment 4, Edge A (S4A)
 23= Segment 4, Edge B (S4B)
 24= Segment 4, Difference (S4D)
 25= Segment 4, Center axis (S4C)
 26= Segment 5, Edge A (S5A)
 27= Segment 5, Edge B (S5B)
 28= Segment 5, Difference (S5D)
 29= Segment 5, Center axis (S5C)
 30= Segment 6, Edge A (S6A)
 31= Segment 6, Edge B (S6B)
 32= Segment 6, Difference (S6D)
 33= Segment 6, Center axis (S6C)
 34= Segment 7, Edge A (S7A)
 35= Segment 7, Edge B (S7B)
 36= Segment 7, Difference (S7D)
 37= Segment 7, Center axis (S7C)
 38= Segment 8, Edge A (S8A)
 39= Segment 8, Edge B (S8B)
 40= Segment 8, Difference (S8D)
 41= Segment 8, Center axis (S8C)

Description: Data to be used for analog output.

13.3.3.4.3 Set_AnalogRange (ANALOG RANGE)

Set the analog output range.

Parameter: int32_t SP_AnalogRange

SP_AnalogRange

Direction: Down

Valid values:

0= None

1= 0 - 10V

Description: Analog output range.

13.3.3.4.4 Get_AnalogRange (ANALOG RANGE)

Get the analog output range.

Parameter: int32_t SA_AnalogRange

SA_AnalogRange

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= None

1= 0 - 10V

Description: Analog output range.

13.3.3.4.5 Set_AnalogScale (ANALOGSCALE)

Set the scaling factor for analog output. If both parameters are zero, the analog output is scaled to default range.

Parameter: int32_t SP_AnalogScaleMode	SP_AnalogScaleMode
Direction: Down	
Valid values:	
0= Standard	
1= Two point (TWOPOINT)	
Description: Analog scale mode.	
Parameter: double SP_MinValue	SP_MinValue
Direction: Down	
Valid values:	
Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)	
Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)	
Unit: mm	
Description: Value which represents lowest voltage/current (at two point scaling).	
Parameter: double SP_MaxValue	SP_MaxValue
Direction: Down	
Valid values:	
Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)	
Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)	
Unit: mm	
Description: Value which represents highest voltage/current (at two point scaling).	

13.3.3.4.6 Get_AnalogScale (ANALOGSCALE)

Get the scaling factor for analog output. If both parameters are zero, the analog output is scaled to default range.

Parameter: int32_t SA_AnalogScaleMode	SA_AnalogScaleMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Standard	
1= Two point (TWOPOINT)	
Description: Analog scale mode.	
Parameter: double SA_MinValue	SA_MinValue
Direction: Up	
Valid values:	
Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)	
Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)	
Unit: mm	
Description: Value which represents lowest voltage/current (at two point scaling).	

Parameter: double SA_MaxValue SA_MaxValue
Direction: Up
Valid values:
Minimum: -100.0 (for 46 mm range), -200.0 (for 95 mm range)
Maximum: 100.0 (for 46 mm range), 200.0 (for 95 mm range)
Unit: mm
Description: Value which represents highest voltage/current (at two point scaling).

13.4 Commands for ODC2600

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (native).
- [IF2004](#) (native).
- [TCP/IP](#) (additional, e.g. RS232/RS422 to TCP/IP comm server and RS232 high level interface).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Info](#) and [Read_OptionData](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate, to interpret data and to assign values.

If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls sensor command [Dat_Out_On](#) at [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor (including two segment bits, mask it out by raw&0xffff), from 0 to 65535 (without segment bits).
- Scaled values are scaled using sensor range, error values are scaled depending of [IP_ScaleErrorValues](#).

The values of selected segments are filled in the arrays one after another. Each array always starts with first selected segment.

13.4.1 General commands

13.4.1.1 General

13.4.1.1.1 Get_Info (INFO)

Retrieve some information about the sensor.

Parameter: String SA_ArticleNumber SA_ArticleNumber
Direction: Up
Valid values:
 Numeric value
Description: Article number of the sensor.

Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the sensor.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Unit: mm	
Valid values:	
40.0	
Description: Range of the sensor.	
Parameter: int32_t SA_Reserve_1	SA_Reserve_1
Direction: Up	
Description: Reserved.	
Parameter: String SA_SoftArtBoot	SA_SoftArtBoot
Direction: Up	
Description: Sensor software versions.	
Parameter: String SA_SoftArtArm	SA_SoftArtArm
Direction: Up	
Description: Sensor software versions.	
Parameter: String SA_SoftArtDSP	SA_SoftArtDSP
Direction: Up	
Description: Sensor software versions.	
Parameter: int32_t SA_SoftVerBoot	SA_SoftVerBoot
Direction: Up	
Valid values:	
Numeric value	
Description: Sensor software versions.	
Parameter: int32_t SA_SoftVerArm	SA_SoftVerArm
Direction: Up	
Valid values:	
Numeric value	
Description: Sensor software versions.	
Parameter: int32_t SA_SoftVerDSP	SA_SoftVerDSP
Direction: Up	
Valid values:	
Numeric value	
Description: Sensor software versions.	

13.4.1.1.2 Reset_Boot (RESET)

Resets the sensor.

13.4.1.2 Triggering

13.4.1.2.1 Triggermode_Reset (TRIGGERMODE RESET)

Resets the values at trigger mode.

13.4.1.2.2 Triggermode_Trigger (TRIGGERMODE TRIGGER)

Activate output in trigger mode.

13.4.1.3 Parameter management

13.4.1.3.1 Save_OptionData (SAVE OPT RAM TO FLASH)

Save the option data to flash.

13.4.1.3.2 Save_MeasProgData (SAVE MPR RAM TO FLASH)

Save the measure program data to flash.

13.4.2 Measurement

13.4.2.1 Choose_MeasProg (CHOOSE MP)

Select the measurement program of sensor.

Parameter: int32_t SP_MeasProgNumber

SP_MeasProgNumber

Direction: Down

Valid values:

- 0= EdgeHL
- 1= EdgeLH
- 2= DIA
- 3= GAP
- 4= SEG_2_4
- 5= MULTISEG
- 6= USER1
- 7= USER2
- 8= USER3
- 9= USER4

Description: Measure program of sensor.

13.4.2.2 Switch_Edge (SWITCH EDGE)

Select the edges to measure.

Parameter: int32_t SP_FrontEdge_Seg1	SP_FrontEdge_Seg1
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 1.	
Parameter: int32_t SP_FrontEdge_Seg2	SP_FrontEdge_Seg2
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 2.	
Parameter: int32_t SP_FrontEdge_Seg3	SP_FrontEdge_Seg3
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 3.	
Parameter: int32_t SP_FrontEdge_Seg4	SP_FrontEdge_Seg4
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 4.	
Parameter: int32_t SP_RearEdge_Seg1	SP_RearEdge_Seg1
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Rear edge of segment 1.	
Parameter: int32_t SP_RearEdge_Seg2	SP_RearEdge_Seg2
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Rear edge of segment 2.	
Parameter: int32_t SP_RearEdge_Seg3	SP_RearEdge_Seg3
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Rear edge of segment 3.	

Parameter: int32_t SP_RearEdge_Seg4 SP_RearEdge_Seg4
Direction: Down
Valid values:
Minimum: 0
Maximum: 80
Description: Rear edge of segment 4.

13.4.2.3 Write_OptionData (WR OPT TO RAM)

Write option data to sensor.

Parameter: int32_t SP_MeasProgNumber SP_MeasProgNumber
Direction: Down
Valid values:
 0= EdgeHL
 1= EdgeLH
 2= DIA
 3= GAP
 4= SEG_2_4
 5= MULTISEG
 6= USER1
 7= USER2
 8= USER3
 9= USER4
Description: Measure program of sensor.

Parameter: int32_t SP_Language SP_Language
Direction: Down
Valid values:
 0= German
 1= English
Description: Language of sensor.

Parameter: int32_t SP_DispmesUnit SP_DispmesUnit
Direction: Down
Valid values:
 0= mm
 1= inch
Description: Display measurement unit of sensor.

Parameter: int32_t SP_ErrorHandler SP_ErrorHandler
Direction: Down
Valid values:
 0= error output
 1= retain last value
Description: If the sensor cannot measure values, it can output the last valid value or it can output an error values.

Parameter: int32_t SP_SerialOutFormat SP_SerialOutFormat
Direction: Down
Valid values:
 0= binary
 1= ASCII
Description: Serial output format of sensor.

Parameter: int32_t SP_Ext_LaserSwitch	SP_Ext_LaserSwitch
Direction: Down	
Valid values:	
0= not active	
1= active	
Description: Enable/Disable external laser switch.	
Parameter: int32_t SP_LaserIntensity	SP_LaserIntensity
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: No effect.	
Parameter: int32_t SP_Contrast	SP_Contrast
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: Display contrast.	
Parameter: int32_t SP_EdgeDetectThreshold	SP_EdgeDetectThreshold
Direction: Down	
Valid values:	
Minimum: 20	
Maximum: 90	
Unit: %	
Description: Contrast for edge detection.	
Parameter: int32_t SP_Reserve_2	SP_Reserve_2
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved.	
Parameter: int32_t SP_ActiveSerialIf	SP_ActiveSerialIf
Direction: Down	
Valid values:	
0= RS422	
1= RS232	
Description: Active serial interface of sensor.	
Parameter: int32_t SP_RS232_Baudrate	SP_RS232_Baudrate
Direction: Down	
Valid values:	
9600	
19200	
38400	
115200	
Unit: Baud	
Description: Baudrate of RS232 interface of sensor.	

Parameter: int32_t SP_RS232_Parity	SP_RS232_Parity
Direction: Down	
Valid values:	
0= none	
1= even	
2= odd	
Description: Parity of RS232 interface of sensor.	
Parameter: int32_t SP_RS232_StopBits	SP_RS232_StopBits
Direction: Down	
Valid values:	
1	
2	
Description: Stop bits of RS232 interface of sensor.	
Parameter: int32_t SP_RS232_TimeoutSend	SP_RS232_TimeoutSend
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SP_RS232_TimeoutRecv	SP_RS232_TimeoutRecv
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SP_RS422_Baudrate	SP_RS422_Baudrate
Direction: Down	
Valid values:	
9600	
19200	
38400	
115200	
691200	
Unit: Baud	
Description: Baudrate of RS422 interface of sensor.	
Parameter: int32_t SP_RS422_Parity	SP_RS422_Parity
Direction: Down	
Valid values:	
0= none	
1= even	
2= odd	
Description: Parity of RS422 interface of sensor.	
Parameter: int32_t SP_RS422_StopBits	SP_RS422_StopBits
Direction: Down	
Valid values:	
1	
2	
Description: Stop bits of RS422 interface of sensor.	

Parameter: int32_t SP_RS422_TimeoutSend	SP_RS422_TimeoutSend
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SP_RS422_TimeoutRecv	SP_RS422_TimeoutRecv
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	

13.4.2.4 Read_OptionData (RD OPT RAM)

Read the option data from sensor.

Parameter: int32_t SA_MeasProgNumber	SA_MeasProgNumber
Direction: Up	
Valid values:	
0= EdgeHL	
1= EdgeLH	
2= DIA	
3= GAP	
4= SEG_2_4	
5= MULTISEG	
6= USER1	
7= USER2	
8= USER3	
9= USER4	
Description: Measure program of sensor.	
Parameter: int32_t SA_Language	SA_Language
Direction: Up	
Valid values:	
0= German	
1= English	
Description: Language of sensor.	
Parameter: int32_t SA_DispmesUnit	SA_DispmesUnit
Direction: Up	
Valid values:	
0= mm	
1= inch	
Description: Display measurement unit of sensor.	
Parameter: int32_t SA_ErrorHandler	SA_ErrorHandler
Direction: Up	
Valid values:	
0= error output	
1= retain last value	
Description: If the sensor cannot measure values, it can output the last valid value or it can output an error values.	

Parameter: int32_t SA_SerialOutFormat	SA_SerialOutFormat
Direction: Up	
Valid values:	
0= binary	
1= ASCII	
Description: Serial output format of sensor.	
Parameter: int32_t SA_Ext_LaserSwitch	SA_Ext_LaserSwitch
Direction: Up	
Valid values:	
0= not active	
1= active	
Description: Enable/Disable external laser switch.	
Parameter: int32_t SA_LaserIntensity	SA_LaserIntensity
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: No effect.	
Parameter: int32_t SA_Contrast	SA_Contrast
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: Display contrast.	
Parameter: int32_t SA_EdgeDetectThreshold	SA_EdgeDetectThreshold
Direction: Up	
Valid values:	
Minimum: 20	
Maximum: 90	
Unit: %	
Description: Contrast for edge detection.	
Parameter: int32_t SA_Reserve_2	SA_Reserve_2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved.	
Parameter: int32_t SA_ActiveSerialIf	SA_ActiveSerialIf
Direction: Up	
Valid values:	
0= RS422	
1= RS232	
Description: Active serial interface of sensor.	

Parameter: int32_t SA_RS232_Baudrate	SA_RS232_Baudrate
Direction: Up	
Valid values:	
9600	
19200	
38400	
115200	
Unit: Baud	
Description: Baudrate of RS232 interface of sensor.	
Parameter: int32_t SA_RS232_Parity	SA_RS232_Parity
Direction: Up	
Valid values:	
0= none	
1= even	
2= odd	
Description: Parity of RS232 interface of sensor.	
Parameter: int32_t SA_RS232_StopBits	SA_RS232_StopBits
Direction: Up	
Valid values:	
1	
2	
Description: Stop bits of RS232 interface of sensor.	
Parameter: int32_t SA_RS232_TimeoutSend	SA_RS232_TimeoutSend
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SA_RS232_TimeoutRecv	SA_RS232_TimeoutRecv
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SA_RS422_Baudrate	SA_RS422_Baudrate
Direction: Up	
Valid values:	
9600	
19200	
38400	
115200	
691200	
Unit: Baud	
Description: Baudrate of RS422 interface of sensor.	
Parameter: int32_t SA_RS422_Parity	SA_RS422_Parity
Direction: Up	
Valid values:	
0= none	
1= even	
2= odd	
Description: Parity of RS422 interface of sensor.	

Parameter: int32_t SA_RS422_StopBits	SA_RS422_StopBits
Direction: Up	
Valid values:	
1	
2	
Description:	Stop bits of RS422 interface of sensor.
Parameter: int32_t SA_RS422_TimeoutSend	SA_RS422_TimeoutSend
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description:	No effect.
Parameter: int32_t SA_RS422_TimeoutRecv	SA_RS422_TimeoutRecv
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description:	No effect.

13.4.2.5 Update_OptionData

Update option data at sensor.

This is a meta command which first calls [Read_OptionData](#), renames all answer parameters (SA_...) to command parameters (SP_...), overwrites the parameter set by the parameters from user and finally calls [Write_OptionData](#).

The parameters of this command are NOT obligatory. Missing parameters are not overwritten, so the sensor keeps the original values.

Parameter: int32_t SP_MeasProgNumber	SP_MeasProgNumber
Direction: Down	
Valid values:	
0= EdgeHL	
1= EdgeLH	
2= DIA	
3= GAP	
4= SEG_2_4	
5= MULTISEG	
6= USER1	
7= USER2	
8= USER3	
9= USER4	
Description:	Measure program of sensor.
Parameter: int32_t SP_Language	SP_Language
Direction: Down	
Valid values:	
0= German	
1= English	
Description:	Language of sensor.

Parameter: int32_t SP_DispmesUnit	SP_DispmesUnit
Direction: Down	
Valid values:	
0= mm	
1= inch	
Description: Display measurement unit of sensor.	
Parameter: int32_t SP_ErrorHandler	SP_ErrorHandler
Direction: Down	
Valid values:	
0= error output	
1= retain last value	
Description: If the sensor cannot measure values, it can output the last valid value or it can output an error values.	
Parameter: int32_t SP_SerialOutFormat	SP_SerialOutFormat
Direction: Down	
Valid values:	
0= binary	
1= ASCII	
Description: Serial output format of sensor.	
Parameter: int32_t SP_Ext_LaserSwitch	SP_Ext_LaserSwitch
Direction: Down	
Valid values:	
0= not active	
1= active	
Description: Enable/Disable external laser switch.	
Parameter: int32_t SP_LaserIntensity	SP_LaserIntensity
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: No effect.	
Parameter: int32_t SP_Contrast	SP_Contrast
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 100	
Unit: %	
Description: Display contrast.	
Parameter: int32_t SP_EdgeDetectThreshold	SP_EdgeDetectThreshold
Direction: Down	
Valid values:	
Minimum: 20	
Maximum: 90	
Unit: %	
Description: Contrast for edge detection.	

Parameter: int32_t SP_Reserve_2	SP_Reserve_2
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved.	
Parameter: int32_t SP_ActiveSerialIf	SP_ActiveSerialIf
Direction: Down	
Valid values:	
0 = RS422	
1 = RS232	
Description: Active serial interface of sensor.	
Parameter: int32_t SP_RS232_Baudrate	SP_RS232_Baudrate
Direction: Down	
Valid values:	
9600	
19200	
38400	
115200	
Unit: Baud	
Description: Baudrate of RS232 interface of sensor.	
Parameter: int32_t SP_RS232_Parity	SP_RS232_Parity
Direction: Down	
Valid values:	
0 = none	
1 = even	
2 = odd	
Description: Parity of RS232 interface of sensor.	
Parameter: int32_t SP_RS232_StopBits	SP_RS232_StopBits
Direction: Down	
Valid values:	
1	
2	
Description: Stop bits of RS232 interface of sensor.	
Parameter: int32_t SP_RS232_TimeoutSend	SP_RS232_TimeoutSend
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SP_RS232_TimeoutRecv	SP_RS232_TimeoutRecv
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	

Parameter: int32_t SP_RS422_Baudrate	SP_RS422_Baudrate
Direction: Down	
Valid values:	
9600	
19200	
38400	
115200	
691200	
Unit: Baud	
Description: Baudrate of RS422 interface of sensor.	
Parameter: int32_t SP_RS422_Parity	SP_RS422_Parity
Direction: Down	
Valid values:	
0= none	
1= even	
2= odd	
Description: Parity of RS422 interface of sensor.	
Parameter: int32_t SP_RS422_StopBits	SP_RS422_StopBits
Direction: Down	
Valid values:	
1	
2	
Description: Stop bits of RS422 interface of sensor.	
Parameter: int32_t SP_RS422_TimeoutSend	SP_RS422_TimeoutSend
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	
Parameter: int32_t SP_RS422_TimeoutRecv	SP_RS422_TimeoutRecv
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: No effect.	

13.4.2.6 Write_MeasProgData (WR MPR TO RAM)

Write measurement program data to sensor.

Parameter: int32_t SP_UserMeasProgNumber	SP_UserMeasProgNumber
Direction: Down	
Valid values:	
6= USER1	
7= USER2	
8= USER3	
9= USER4	
Description: User measure program of sensor.	

Parameter: String SP_MeasProgName	SP_MeasProgName
Direction: Down	
Description: Measure program name of sensor.	
Parameter: double SP_AnalogOffset	SP_AnalogOffset
Direction: Down	
Valid values:	
Minimum: -10.0	
Maximum: 10.0	
Unit: V	
Description: Analog output offset of sensor.	
Parameter: double SP_AnalogGain	SP_AnalogGain
Direction: Down	
Valid values:	
Minimum: -4.0	
Maximum: 4.0	
Description: Analog output gain of sensor.	
Parameter: double SP_DisplayOffset	SP_DisplayOffset
Direction: Down	
Valid values:	
Minimum: -99.99	
Maximum: 99.99	
Unit: mm	
Description: Display offset of sensor.	
Parameter: double SP_DisplayGain	SP_DisplayGain
Direction: Down	
Valid values:	
Minimum: -2.0	
Maximum: 2.0	
Description: Display gain of sensor.	
Parameter: double SP_UpperLimit	SP_UpperLimit
Direction: Down	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Upper limit of sensor.	
Parameter: double SP_LowerLimit	SP_LowerLimit
Direction: Down	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Lower limit of sensor.	
Parameter: double SP_UpperWarning	SP_UpperWarning
Direction: Down	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Upper warning of sensor.	

Parameter: double SP_LowerWarning SP_LowerWarning

Direction: Down

Valid values:

Minimum: -168.876

Maximum: 168.876

Unit: mm

Description: Lower warning of sensor.

Parameter: int32_t SP_Reserve_3 SP_Reserve_3

Direction: Down

Valid values:

Minimum: 0

Maximum: 0

Description: Reserved.

Parameter: int32_t SP_MeasMode SP_MeasMode

Direction: Down

Valid values:

 0= Normal

 1= Max_Cont

 2= Min_Cont

 3= P-P_Cont

 4= Max_Trg

 5= Min_Trg

 6= P-P_Trg

 7= SC1_Trg

Description: Measure Mode.

Parameter: int32_t SP_Median SP_Median

Direction: Down

Valid values:

 0= no Median

 3

 5

 7

 9

Description: Median over n values.

Parameter: int32_t SP_Average_for_reading SP_Average_for_reading

Direction: Down

Valid values:

 1 to 128 sliding

 129 to 4096 recursive

Description: Averaging mode and number of sensor.

Parameter: int32_t SP_Reserve_4 SP_Reserve_4

Direction: Down

Valid values:

Minimum: 0

Maximum: 0

Description: Reserved.

Parameter: int32_t SP_MeasObject	SP_MeasObject
Direction: Down	
Valid values:	
1= EdgeHL	
2= EdgeLH	
3= DIA	
4= GAP	
5= SEG_2_4	
6= MULTISEG	
Description: Measurement program.	
Parameter: int32_t SP_NumberOfSegments	SP_NumberOfSegments
Direction: Down	
Valid values:	
1 for EdgeHL, EdgeLH, DIA, GAP and SEG_2_4	
2..4 for MULTISEG	
Description: Number of segments.	
Parameter: int32_t SP_FrontEdge_Seg1	SP_FrontEdge_Seg1
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 1.	
Parameter: int32_t SP_FrontEdge_Seg2	SP_FrontEdge_Seg2
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 2.	
Parameter: int32_t SP_FrontEdge_Seg3	SP_FrontEdge_Seg3
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 3.	
Parameter: int32_t SP_FrontEdge_Seg4	SP_FrontEdge_Seg4
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 4.	
Parameter: int32_t SP_Reserve_6	SP_Reserve_6
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved.	

Parameter: int32_t SP_Reserve_7	SP_Reserve_7
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved.	
Parameter: int32_t SP_RearEdge_Seg1	SP_RearEdge_Seg1
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Rear edge of segment 1.	
Parameter: int32_t SP_RearEdge_Seg2	SP_RearEdge_Seg2
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Rear edge of segment 2.	
Parameter: int32_t SP_RearEdge_Seg3	SP_RearEdge_Seg3
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Rear edge of segment 3.	
Parameter: int32_t SP_RearEdge_Seg4	SP_RearEdge_Seg4
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Rear edge of segment 4.	
Parameter: int32_t SP_Reserve_9	SP_Reserve_9
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved.	
Parameter: int32_t SP_Reserve_10	SP_Reserve_10
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved.	
Parameter: double SP_MasterValue	SP_MasterValue
Direction: Down	
Valid values:	
Minimum: -40.0	
Maximum: 40.0	
Unit: mm	
Description: Master value of sensor.	

13.4.2.7 Read_MeasProgData (RD MPR RAM)

Read measurement program data from sensor

Parameter: int32_t SA_MeasProgNumber	SA_MeasProgNumber
Direction: Up	
Valid values:	
0= EdgeHL	
1= EdgeLH	
2= DIA	
3= GAP	
4= SEG_2_4	
5= MULTISEG	
6= USER1	
7= USER2	
8= USER3	
9= USER4	
Description: User measure program of sensor.	
Parameter: String SA_MeasProgName	SA_MeasProgName
Direction: Up	
Description: Measure program name of sensor.	
Parameter: double SA_AnalogOffset	SA_AnalogOffset
Direction: Up	
Valid values:	
Minimum: -10.0	
Maximum: 10.0	
Unit: V	
Description: Analog output offset of sensor.	
Parameter: double SA_AnalogGain	SA_AnalogGain
Direction: Up	
Valid values:	
Minimum: -4.0	
Maximum: 4.0	
Description: Analog output gain of sensor.	
Parameter: double SA_DisplayOffset	SA_DisplayOffset
Direction: Up	
Valid values:	
Minimum: -99.99	
Maximum: 99.99	
Unit: mm	
Description: Display offset of sensor.	
Parameter: double SA_DisplayGain	SA_DisplayGain
Direction: Up	
Valid values:	
Minimum: -2.0	
Maximum: 2.0	
Description: Display gain of sensor.	

Parameter: double SA_UpperLimit	SA_UpperLimit
Direction: Up	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Upper limit of sensor.	
Parameter: double SA_LowerLimit	SA_LowerLimit
Direction: Up	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Lower limit of sensor.	
Parameter: double SA_UpperWarning	SA_UpperWarning
Direction: Up	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Upper warning of sensor.	
Parameter: double SA_LowerWarning	SA_LowerWarning
Direction: Up	
Valid values:	
Minimum: -168.876	
Maximum: 168.876	
Unit: mm	
Description: Lower warning of sensor.	
Parameter: int32_t SA_Reserve_3	SA_Reserve_3
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved.	
Parameter: int32_t SA_MeasMode	SA_MeasMode
Direction: Up	
Valid values:	
0= Normal	
1= Max_Cont	
2= Min_Cont	
3= P-P_Cont	
4= Max_Trg	
5= Min_Trg	
6= P-P_Trg	
7= SC1_Trg	
Description: Measure Mode.	

Parameter: int32_t SA_Median	SA_Median
Direction: Up	
Valid values:	
0= no Median	
3	
5	
7	
9	
Description: Median over n values.	
Parameter: int32_t SA_Average_for_reading	SA_Average_for_reading
Direction: Up	
Valid values:	
1 to 128 sliding	
129 to 4096 recursive	
Description: Averaging mode and number of sensor.	
Parameter: int32_t SA_Reserve_4	SA_Reserve_4
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved.	
Parameter: int32_t SA_MeasObject	SA_MeasObject
Direction: Up	
Valid values:	
1= EdgeHL	
2= EdgeLH	
3= DIA	
4= GAP	
5= SEG_2_4	
6= MULTISEG	
Description: Measurement program.	
Parameter: int32_t SA_NumberOfSegments	SA_NumberOfSegments
Direction: Up	
Valid values:	
1 for EdgeHL, EdgeLH, DIA, GAP and SEG_2_4	
2..4 for MULTISEG	
Description: Number of segments.	
Parameter: int32_t SA_FrontEdge_Seg1	SA_FrontEdge_Seg1
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 1.	
Parameter: int32_t SA_FrontEdge_Seg2	SA_FrontEdge_Seg2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 2.	

Parameter: int32_t SA_FrontEdge_Seg3	SA_FrontEdge_Seg3
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 3.	
Parameter: int32_t SA_FrontEdge_Seg4	SA_FrontEdge_Seg4
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Front edge of segment 4.	
Parameter: int32_t SA_Reserve_6	SA_Reserve_6
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved.	
Parameter: int32_t SA_Reserve_7	SA_Reserve_7
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 0	
Description: Reserved.	
Parameter: int32_t SA_RearEdge_Seg1	SA_RearEdge_Seg1
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Rear edge of segment 1.	
Parameter: int32_t SA_RearEdge_Seg2	SA_RearEdge_Seg2
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Rear edge of segment 2.	
Parameter: int32_t SA_RearEdge_Seg3	SA_RearEdge_Seg3
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Rear edge of segment 3.	
Parameter: int32_t SA_RearEdge_Seg4	SA_RearEdge_Seg4
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 80	
Description: Rear edge of segment 4.	

Parameter: int32_t SA_Reserve_9 SA_Reserve_9

Direction: Up

Valid values:

Minimum: 0

Maximum: 0

Description: Reserved.

Parameter: int32_t SA_Reserve_10 SA_Reserve_10

Direction: Up

Valid values:

Minimum: 0

Maximum: 0

Description: Reserved.

Parameter: double SA_MasterValue SA_MasterValue

Direction: Up

Valid values:

Minimum: -40.0

Maximum: 40.0

Unit: mm

Description: Master value of sensor.

13.4.2.8 Set_LightRef (SET LIGHT REFERENCE TUNING)

Set the light reference.

13.4.2.9 Reset_LightRef (RESET LIGHT REFERENCE TUNING)

Reset the light reference.

13.4.2.10 Read_MinMax (RD MINMAX)

Read the minimum and maximum values from sensor.

Parameter: int32_t SA_MinRaw SA_MinRaw

Direction: Up

Valid values:

Minimum: 0

Maximum: 65519

Description: Minimum raw value.

Parameter: int32_t SA_MaxRaw SA_MaxRaw

Direction: Up

Valid values:

Minimum: 0

Maximum: 65519

Description: Maximum raw value.

Parameter: double SA_MinScaled SA_MinScaled

Direction: Up

Valid values:

Minimum: -0.4204872

Maximum: 40.4035128

Unit: mm

Description: Minimum scaled value.

Parameter: double SA_MaxScaled SA_MaxScaled

Direction: Up

Valid values:

Minimum: -0.4204872

Maximum: 40.4035128

Unit: mm

Description: Maximum scaled value.

13.4.2.11 Read_MinMax_Reset (RD MINMAX RESET)

Read the minimum and maximum values from sensor and reset the values.

Parameter: int32_t SA_MinRaw SA_MinRaw

Direction: Up

Valid values:

Minimum: 0

Maximum: 65519

Description: Minimum raw value.

Parameter: int32_t SA_MaxRaw SA_MaxRaw

Direction: Up

Valid values:

Minimum: 0

Maximum: 65519

Description: Maximum raw value.

Parameter: double SA_MinScaled SA_MinScaled

Direction: Up

Valid values:

Minimum: -0.4204872

Maximum: 40.4035128

Unit: mm

Description: Minimum scaled value.

Parameter: double SA_MaxScaled SA_MaxScaled

Direction: Up

Valid values:

Minimum: -0.4204872

Maximum: 40.4035128

Unit: mm

Description: Maximum scaled value.

13.4.3 Data output

13.4.3.1 General

13.4.3.1.1 **Dat_Out_Off (STOP)**

Switch off data output from sensor.

13.4.3.1.2 **Dat_Out_On (START)**

Switch on data output from sensor.

14 Commands for eddyNCDT sensors

DT3909 and ESC4912 are no longer supported.

14.1 Commands for DT306x/7x

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [TCP/IP](#) (native).

If first bit of `IP_AutomaticMode` is set (1), MEDAQLib calls automatically sensor command [Get_Status](#), [Get_SampleTime](#), [Get_RawDataRanges](#) and [Get_ChannelInfos](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate and to assign values.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor, either from -2147483648 (`INT32_MIN`) to 2147483647 (`INT32_MAX`) or from 0 to 4294967295 (`UINT32_MAX`) or from -3.402823466e+38 (`-FLT_MAX`) to 3.402823466e+38 (`FLT_MAX`).
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_RawDataRanges](#) and [Get_ChannelInfos](#)).

The values of selected channels are filled in the arrays one after another. Each array always starts with first selected channel.

14.1.1 User Level

14.1.1.1 Logout (LGO)

Change user level to user at web interface.

14.1.1.2 Login (LGI)

Change user level to setup at web interface.

Parameter: String SP_Password	SP_Password
Direction: Down	
Description: Valid password to login.	

14.1. Commands for DT306x/7x

14.1.1.3 Set_Password (PWD)

Change the password for login.

Parameter: String SP_OldPassword
Direction: Down
Description: Old password.

SP_OldPassword

Parameter: String SP_NewPassword
Direction: Down
Description: New password.

SP_NewPassword

14.1.2 Measurement

14.1.2.1 Get_SampleTime (STI?)

Get the current sample time.

Parameter: double SA_SampleTime
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 4294967040.0
Unit: μ s
Description: Real sample time at controller

SA_SampleTime

14.1.2.2 Get_Measure1 (GMD1)

Get values from sensor asynchronously.

Parameter: int32_t SA_Status
Direction: Up
Valid values:
Bit combination of 16 bits
Description: State at controller

SA_Status

Parameter: double SA_SensorTemperature
Direction: Up
Unit: °C
Description: Temperature at sensor

SA_SensorTemperature

Parameter: double SA_ControllerTemperature
Direction: Up
Unit: °C
Description: Temperature at controller

SA_ControllerTemperature

14.1.2.3 Get_Measure2 (GMD2)

Get values from sensor asynchronously.

Parameter: int32_t SA_Status
Direction: Up
Valid values:
Bit combination of 16 bits
Description: State at controller

SA_Status

Parameter: double SA_MeasureValue	SA_MeasureValue
Direction: Up	
Description: Distance value of sensor	
Parameter: double SA_SensorTemperature	SA_SensorTemperature
Direction: Up	
Unit: °C	
Description: Temperature at sensor	
Parameter: double SA_ControllerTemperature	SA_ControllerTemperature
Direction: Up	
Unit: °C	
Description: Temperature at controller	

14.1.3 Data output

14.1.3.1 ChannelStatus (CHS)

Retrieve the available channels at controller.

Parameter: int32_t SA_ChExist1..3	SA_ChExist1..3
Direction: Up	
Valid values:	
0= Channel not available	
1= Measured channel	

Description: Channel 1 to 3 is available at controller.

14.1.3.2 Get_RawDataRange (MDF)

Read the raw data range for a channel from controller. The range is used by MEDAQLib for scaling raw data.

Parameter: int32_t SP_Chан	SP_Chан
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 3	

Description: Channel to read the raw data range for.

Parameter: double SA_RawRangeMin	SA_RawRangeMin
Direction: Up	
Valid values:	
Minimum: -2147483648.0 (INT32_MIN)	

Description: Minimum raw data range of channel.

Parameter: double SA_RawRangeMax	SA_RawRangeMax
Direction: Up	
Valid values:	
Minimum: -2147483648.0 (INT32_MIN)	

Description: Maximum raw data range of channel.

14.1.3.3 Get_RawDataRanges

Calls the sensor command Get_RawDataRange for any requested channel.

Parameter: int32_t SP_Complete

SP_Complete

Direction: Down

Valid values:

0 = FALSE

1 = TRUE

Description: Specifies if any possible channel should be requested or only known channels (from former call to [ChannelStatus](#) or [Get_Status](#)).

Parameter: double SA_RawRangeMin1..3

SA_RawRangeMin1..3

Direction: Up

Valid values:

Minimum: -2147483648.0 (INT32_MIN)

Maximum: 2147483647.0 (INT32_MAX)

Description: Minimum raw data range of channel 1 to 3.

Parameter: double SA_RawRangeMax1..3

SA_RawRangeMax1..3

Direction: Up

Valid values:

Minimum: -2147483648.0 (INT32_MIN)

Maximum: 2147483647.0 (INT32_MAX)

Description: Maximum raw data range of channel 1 to 3.

14.1.4 Interfaces

14.1.4.1 Set_DataPort (SDP)

Set the TCP/IP data port at controller.

Parameter: int32_t SP_DataPort

SP_DataPort

Direction: Down

Valid values:

Minimum: 1024

Maximum: 65535

Description: TCP/IP data port at controller.

14.1.4.2 Get_DataPort (GDP)

Retrieve the TCP/IP data port from controller.

Parameter: int32_t SA_DataPort

SA_DataPort

Direction: Up

Valid values:

Minimum: 1024

Maximum: 65535

Description: TCP/IP data port at controller.

14.1.4.3 Set_IPConfiguration (IPS)

Set the IP configuration at controller. For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_DHCPEnabled SP_DHCPEnabled

Direction: Down

Valid values:

0= FALSE
1= TRUE

Description: Specify if controller should use a static IP address or ask for IP at DHCP server (dynamic IP address).

Parameter: String SP_Address SP_Address

Direction: Down

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_SubnetMask SP_SubnetMask

Direction: Down

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_Gateway SP_Gateway

Direction: Down

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: The default gateway must be specified if the controller should communicate with peers in foreign subnets. This parameter is only evaluated on static IP assignment.

14.1.4.4 Get_IPConfiguration (IPS?)

Get the IP configuration at controller.

Parameter: int32_t SA_DHCPEnabled SA_DHCPEnabled

Direction: Up

Valid values:

0= FALSE
1= TRUE

Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).

Parameter: String SA_Address SA_Address

Direction: Up

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.

14.1. Commands for DT306x/7x

Parameter: String SA_SubnetMask	SA_SubnetMask
Direction: Up	
Valid values:	
Valid network mask (e.g. 255.255.255.0 for a Class C network)	
Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.	
Parameter: String SA_Gateway	SA_Gateway
Direction: Up	
Valid values:	
Valid IP address of default gateway in form of xxx.xxx.xxx.xxx	
Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.	

14.1.4.5 Set_AppLanguage (LNG)

Set language of web interface at controller.

Parameter: int32_t SP_ApplicationLanguage	SP_ApplicationLanguage
Direction: Down	
Valid values:	
0= System	
1= English	
2= German	
3= Chinese	
4= Japanise	
5= Korean	
6..99= Reverved	
Description: Language of web interface.	

14.1.4.6 Get_AppLanguage (LNG?)

Get language of web interface from controller.

Parameter: int32_t SA_ApplicationLanguage	SA_ApplicationLanguage
Direction: Up	
Valid values:	
0= System	
1= English	
2= German	
3= Chinese	
4= Japanise	
5= Korean	
6..99= Reverved	
Description: Language of web interface.	

14.1.5 Sensor interface

14.1.5.1 Set_Parameter (PAR)

Set sensor specific parameter.

Parameter: int32_t SP_Block	SP_Block
Direction: Down	
Description: Block index of parameter. See sensor documentation for further information.	

14.1. Commands for DT306x/7x

Parameter: int32_t SP_Page SP_Page

Direction: Down

Description: Page index in block. See sensor documentation for further information.

Parameter: int32_t SP_Offset SP_Offset

Direction: Down

Description: Offset in page. See sensor documentation for further information.

Parameter: String SP_Value SP_Value

Direction: Down

Description: Value of parameter. See sensor documentation for further information.

14.1.5.2 Get_Parameter (PAR?)

Get sensor specific parameter.

Parameter: int32_t SP_Block SP_Block

Direction: Down

Description: Block index of parameter. See sensor documentation for further information.

Parameter: int32_t SP_Page SP_Page

Direction: Down

Description: Page index in block. See sensor documentation for further information.

Parameter: int32_t SP_Offset SP_Offset

Direction: Down

Description: Offset in page. See sensor documentation for further information.

Parameter: String SA_Value SA_Value

Direction: Up

Description: Value of parameter. See sensor documentation for further information.

14.1.6 Information

14.1.6.1 Get_Status (STS)

Retrieve detailed information about the controller.

Parameter: double SA_SampleTime SA_SampleTime

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967040.0

Unit: μ s

Description: Real sample time at controller.

14.1. Commands for DT306x/7x

Parameter: int32_t SA_ChExist1..3 SA_ChExist1..3
Direction: Up
Valid values:
 0= Channel not available
 1= Measured channel
Description: Channel 1 to 3 is available at controller.

14.1.6.2 Get_Version (VER)

Retrieve the sensor software version.

Parameter: String SA_Version SA_Version
Direction: Up
Description: Software version of the controller.

14.1.6.3 Get_ChannelInfo (CHI)

Retrieve information about a sensor channel.

Parameter: int32_t SP_Chан SP_Chан
Direction: Down
Valid values:
Minimum: 1
Maximum: 3
Description: Channels to get information for.

Parameter: String SA_ArticleNumber SA_ArticleNumber
Direction: Up
Valid values:
 Numeric value
Description: Article number of the sensor channel.

Parameter: String SA_SensorName SA_SensorName
Direction: Up
Description: Name of the sensor channel.

Parameter: String SA_SerialNumber SA_SerialNumber
Direction: Up
Valid values:
 Numeric value
Description: Serial number of the sensor channel.

Parameter: double SA_Offset SA_Offset
Direction: Up
Valid values:
Minimum: -4294967040.0
Maximum: 4294967040.0
Unit: See parameter SA_Unit
Description: Offset of the sensor channel.

14.1. Commands for DT306x/7x

Parameter: double SA_Range SA_Range
Direction: Up
Valid values:
Minimum: -4294967040.0
Maximum: 4294967040.0
Unit: See parameter SA_Unit
Description: Range of sensor channel.

Parameter: String SA_Unit SA_Unit
Direction: Up
Description: Unit of parameter SA_Offset and SA_Range, e.g. μm or mm.

14.1.6.4 Get_ChannelInfos

Calls the sensor command Get_ChannelInfo for any requested channel.

Parameter: int32_t SP_Complete SP_Complete
Direction: Down
Valid values:
0 = FALSE
1 = TRUE
Description: Specifies if any possible channel should be requested or only known channels (from former call to [ChannelStatus](#) or [Get_Status](#)).

Parameter: String SA_ArticleNumber1..3 SA_ArticleNumber1..3
Direction: Up
Valid values:
Numeric value
Description: Article number of the sensor channel 1 to 3.

Parameter: String SA_SensorName1..3 SA_SensorName1..3
Direction: Up
Description: Name of the sensor channel 1 to 3.

Parameter: String SA_SerialNumber1..3 SA_SerialNumber1..3
Direction: Up
Valid values:
Numeric value
Description: Serial number of the sensor channel 1 to 3.

Parameter: double SA_Offset1..3 SA_Offset1..3
Direction: Up
Valid values:
Minimum: -4294967040.0
Maximum: 4294967040.0
Unit: See parameter SA_Unit
Description: Offset of the sensor channel 1 to 3.

Parameter: double SA_Range1..3 SA_Range1..3
Direction: Up
Valid values:
Minimum: -4294967040.0
Maximum: 4294967040.0
Unit: See parameter SA_Unit
Description: Range of sensor channel 1 to 3.

Parameter: String SA_Unit1..3 SA_Unit1..3

Direction: Up

Description: Unit of parameter SA_Offset and SA_Range, e.g. μm or mm.

14.1.6.5 Get_ControllerInfo (COI)

Retrieve information about the controller.

Parameter: String SA_ArticleNumber SA_ArticleNumber

Direction: Up

Description: Article number of the controller.

Parameter: String SA_ControllerName SA_ControllerName

Direction: Up

Description: Name of the controller.

Parameter: String SA_SerialNumber SA_SerialNumber

Direction: Up

Valid values:

Numeric value

Description: Serial number of the controller.

Parameter: String SA_Option SA_Option

Direction: Up

Valid values:

Numeric value

Description: Option of the controller.

Parameter: String SA_Softwareversion SA_Softwareversion

Direction: Up

Description: Software version of firmware in the controller.

14.1.7 Internal commands

14.1.7.1 Update_Firmware

Update firmware version at controller.

Attention! This function can take approx. 1 minute. This is an internal command. It should not be used by the customer.

Parameter: Binary data SP_FirmwareFile SP_FirmwareFile

Direction: Down

Valid values:

Firmware file as binary data.

Description: Firmware version, read from file.

Parameter: int32_t SA_Result SA_Result

Direction: Up

Valid values:

0= Failed

1= Success

Description: Result of firmware update.

14.2 Commands for DT3100

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [TCP/IP](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Settings](#) and [Get_SensorInfo](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate and to scale data.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor, from 0 to 16777215.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_SensorInfo](#)).

14.2.1 Information

14.2.1.1 Get_Status (STS)

Retrieve information about the controller.

Parameter: int32_t SA_CableCnt

SA_CableCnt

Direction: Up

Valid values:

Minimum: 0

Maximum: 2

Description: Number of cables.

Parameter: int32_t SA_AvailableTargets

SA_AvailableTargets

Direction: Up

Valid values:

Bit combination of:

1 (Bit 0)= Ferro magnetic

2 (Bit 1)= Non ferro magnetic

4 (Bit 2)= Custom1

8 (Bit 3)= Custom2

128 (Bit 7)= Custom (EPxx-LC)

Description: Available targets at controller.

14.2.1.2 Get_Settings (SET)

Retrieve detailed information about the controller settings.

Parameter: int32_t SA_MeasureMode

SA_MeasureMode

Direction: Up

Valid values:

0= Off

1= Continuous

2= Rising edge

3= Falling edge

4= High level

5= Low level

Description: Measure mode.

Parameter: int32_t SA_SRIndex	SA_SRIndex
Direction: Up	
Valid values:	
0= 3600 Hz	
1= 7200 Hz	
2= 14400 Hz	
Description: Samplerate index.	
Parameter: int32_t SA_AvrType	SA_AvrType
Direction: Up	
Valid values:	
0= off	
1= Moving average	
2= Recursive average	
3= Median	
Description: Averaging type at controller.	
Parameter: int32_t SA_AvrNbr	SA_AvrNbr
Direction: Up	
Valid values:	
0= 4 (moving + recursive), 3 (median)	
1= 8 (moving + recursive), 5 (median)	
2= 16 (moving + recursive), 7 (median)	
3= 32 (moving + recursive), 9 (median)	
Description: Averaging number at controller.	
Parameter: int32_t SA.MeasureCount	SA.MeasureCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 9999	
Description: Number of values to measure.	
Parameter: int32_t SA_CalibTarget	SA_CalibTarget
Direction: Up	
Valid values:	
1= Ferro magnetic	
2= Non ferro magnetic	
4= Custom1	
8= Custom2	
Description: Calibrated target at controller.	
Parameter: String SA_TextField	SA_TextField
Direction: Up	
Valid values:	
Up to 32 ASCII characters, no semicolon or new line.	
Description: ASCII text.	

14.2.1.3 Get_Error (ERR)

Get error information from controller.

Parameter: int32_t SA_Error

SA_Error

Direction: Up

Valid values:

- Bit combination of:
- 1 (Bit 0)= 1 wire shortcut
- 2 (Bit 1)= No EEPROM or 1 wire at +5V
- 4 (Bit 2)= Error reading EEPROM
- 8 (Bit 3)= More than one EEPROM
- 16 (Bit 4)= Invalid data at EEPROM
- 32 (Bit 5)= Error at Three point calibration: Poti reached lower limit
- 64 (Bit 6)= Error at Three point calibration: Poti reached upper limit
- 128 (Bit 7)= Invalid data at internal EEPROM

Description: Error field.

14.2.1.4 Get_ControllerInfo (IND)

Get information about controller.

Parameter: String SA_ControllerSerialNumber

SA_ControllerSerialNumber

Direction: Up

Valid values:

- Numeric value

Description: Serial number of controller.

Parameter: String SA_ControllerProductCode

SA_ControllerProductCode

Direction: Up

Valid values:

- Numeric value

Description: Product code of controller.

Parameter: String SA_ControllerRevisionIndex

SA_ControllerRevisionIndex

Direction: Up

Description: Revision index of controller.

Parameter: String SA_ControllerSoftwareVersion

SA_ControllerSoftwareVersion

Direction: Up

Description: Software version of controller.

Parameter: int32_t SA_ControllerOption

SA_ControllerOption

Direction: Up

Description: Option of controller.

Parameter: String SA_ControllerName

SA_ControllerName

Direction: Up

Valid values:

- DT3100

Description: Name of controller.

14.2.1.5 Get_SensorInfo (SEN)

Get information about attached sensor.

Parameter: String SA_SensorSerialNumber	SA_SensorSerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of sensor.	
Parameter: String SA_SensorProductCode	SA_SensorProductCode
Direction: Up	
Valid values:	
Numeric value	
Description: Product code of sensor.	
Parameter: String SA_SensorRevisionIndex	SA_SensorRevisionIndex
Direction: Up	
Description: Revision index of sensor.	
Parameter: int32_t SA_SensorOption	SA_SensorOption
Direction: Up	
Description: Option of sensor.	
Parameter: String SA_SensorName	SA_SensorName
Direction: Up	
Description: Name of sensor.	
Parameter: int32_t SA_SensorCableLength	SA_SensorCableLength
Direction: Up	
Description: Length of integrated sensor cable.	
Parameter: int32_t SA_SensorStartMeasRange	SA_SensorStartMeasRange
Direction: Up	
Description: Start of measurement range of sensor.	
Parameter: int32_t SA_SensorMidMeasRange	SA_SensorMidMeasRange
Direction: Up	
Description: Mid of measurement range of sensor.	
Parameter: int32_t SA_SensorEndMeasRange	SA_SensorEndMeasRange
Direction: Up	
Description: End of measurement range of sensor.	

14.2.1.6 Get_CableInfo1 (CB1)

Get information about cable 1.

Parameter: String SA_CableSerialNumber	SA_CableSerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of cable.	

Parameter: String SA_CableProductCode	SA_CableProductCode
Direction: Up	
Valid values:	
Numeric value	
Description: Product code of cable.	
Parameter: String SA_CableRevisionIndex	SA_CableRevisionIndex
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of cable.	
Parameter: int32_t SA_CableOption	SA_CableOption
Direction: Up	
Description: Option of cable.	
Parameter: int32_t SA_CableLength	SA_CableLength
Direction: Up	
Description: Length of cable.	

14.2.1.7 Get_CableInfo2 (CB2)

Get information about cable 2.

Parameter: String SA_CableSerialNumber	SA_CableSerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of cable.	
Parameter: String SA_CableProductCode	SA_CableProductCode
Direction: Up	
Valid values:	
Numeric value	
Description: Product code of cable.	
Parameter: String SA_CableRevisionIndex	SA_CableRevisionIndex
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of cable.	
Parameter: int32_t SA_CableOption	SA_CableOption
Direction: Up	
Description: Option of cable.	
Parameter: int32_t SA_CableLength	SA_CableLength
Direction: Up	
Description: Length of cable.	

14.2.1.8 DetectSensorChange (DSC)

Check if sensor was disconnected or reconnected since last check.

Parameter: int32_t SA_SensorChanged

SA_SensorChanged

Direction: Up

Valid values:

0= Sensor was not changed

1= Sensor was changed

Description: Flag if sensor was changed.

14.2.1.9 Set_TextField (ETF)

Set a text field at controller.

Parameter: String SP_TextField

SP_TextField

Direction: Down

Valid values:

Up to 32 ASCII characters, no semicolon or new line.

Description: ASCII text.

14.2.1.10 Get_TextField (ETF?)

Get a text field at controller.

Parameter: String SA_TextField

SA_TextField

Direction: Up

Valid values:

Up to 32 ASCII characters, no semicolon or new line.

Description: ASCII text.

14.2.2 Parameter Management

14.2.2.1 Save_Settings (SSE)

Save settings at controller.

14.2.2.2 Read_Settings (RSE)

Read settings at controller.

14.2.2.3 Set_Defaults (DSE)

Set and retrieve default settings at controller.

Parameter: int32_t SA_MeasureMode

SA_MeasureMode

Direction: Up

Valid values:

0= Off

1= Continuous

2= Rising edge

3= Falling edge

4= High level

5= Low level

Description: Measure mode.

Parameter: int32_t SA_SRIndex	SA_SRIndex
Direction: Up	
Valid values:	
0= 3600 Hz	
1= 7200 Hz	
2= 14400 Hz	
Description: Samplerate index.	
Parameter: int32_t SA_AvrType	SA_AvrType
Direction: Up	
Valid values:	
0= off	
1= Moving average	
2= Recursive average	
3= Median	
Description: Averaging type at controller.	
Parameter: int32_t SA_AvrNbr	SA_AvrNbr
Direction: Up	
Valid values:	
0= 4 (moving + recursive), 3 (median)	
1= 8 (moving + recursive), 5 (median)	
2= 16 (moving + recursive), 7 (median)	
3= 32 (moving + recursive), 9 (median)	
Description: Averaging number at controller.	
Parameter: int32_t SA_MeasureCount	SA_MeasureCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 9999	
Description: Number of values to measure.	
Parameter: int32_t SA_CalibTarget	SA_CalibTarget
Direction: Up	
Valid values:	
1= Ferro magnetic	
2= Non ferro magnetic	
4= Custom1	
8= Custom1	
Description: Calibrated target at controller.	
Parameter: String SA_TextField	SA_TextField
Direction: Up	
Valid values:	
Up to 32 ASCII characters, no semicolon or new line.	
Description: ASCII text.	

14.2.3 Display

14.2.3.1 Set_AppLanguage (LNG)

Set language of java applet at controller.

Parameter: int32_t SP_ApplicationLanguage

SP_ApplicationLanguage

Direction: Down

Valid values:

- 0= System
- 1= English
- 2= German
- 3..255 reserved

Description: Language of java applet.

14.2.3.2 Get_AppLanguage (LNG?)

Get language of java applet from controller.

Parameter: int32_t SA_ApplicationLanguage

SA_ApplicationLanguage

Direction: Up

Valid values:

- 0= System
- 1= English
- 2= German
- 3..255 reserved

Description: Language of java applet.

14.2.3.3 Set_AppYScale (SCA)

Set display scaling settings of java applet at controller.

For this command an [Update...](#) meta command is available.

Parameter: int32_t SP_ApplicationScaleMode

SP_ApplicationScaleMode

Direction: Down

Valid values:

- 0= User defined
- 1= Full scale

Description: Scaling mode of java applet.

Parameter: int32_t SP_ApplicationYMin

SP_ApplicationYMin

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 65535

Description: Lower bound at display of java applet.

Parameter: int32_t SP_ApplicationYMax

SP_ApplicationYMax

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 65535

Description: Upper bound at display of java applet.

14.2.3.4 Get_ApplyScale (SCA?)

Get display scaling settings of java applet at controller.

Parameter: int32_t SA_ApplicationScaleMode SA_ApplicationScaleMode

Direction: Up

Valid values:

0= User defined

1= Full scale

Description: Scaling mode of java applet.

Parameter: int32_t SA_ApplicationYMin SA_ApplicationYMin

Direction: Up

Valid values:

Minimum: 0

Maximum: 65535

Description: Lower bound at display of java applet.

Parameter: int32_t SA_ApplicationYMax SA_ApplicationYMax

Direction: Up

Valid values:

Minimum: 0

Maximum: 65535

Description: Upper bound at display of java applet.

14.2.3.5 Set_AppIDispMode (DMD)

Set display mode of java applet at controller.

Parameter: int32_t SP_ApplicationDisplayMode SP_ApplicationDisplayMode

Direction: Down

Valid values:

0= Roll

1= Single frame

Description: Display mode of java applet.

14.2.3.6 Get_AppIDispMode (DMD?)

Get display mode of java applet at controller.

Parameter: int32_t SA_ApplicationDisplayMode SA_ApplicationDisplayMode

Direction: Up

Valid values:

0= Roll

1= Single frame

Description: Display mode of java applet.

14.2.4 Measurement

14.2.4.1 Set_SRIndex (SRA)

Set the samplerate for data acquisition.

Parameter: int32_t SP_SRIndex

SP_SRIndex

Direction: Down

Valid values:

0= 3600 Hz
1= 7200 Hz
2= 14400 Hz

Description: Samplerate index

14.2.4.2 Get_SRIndex (SRA?)

Get the current samplerate.

Parameter: int32_t SA_SRIndex

SA_SRIndex

Direction: Up

Valid values:

0= 3600 Hz
1= 7200 Hz
2= 14400 Hz

Description: Samplerate index.

14.2.4.3 Get_Measure (GMD)

Retrieve one value from sensor, even if trigger is active.

14.2.4.4 Set_MeasureCnt (VTT)

Set the number of values to measure at several measure modes.

Parameter: int32_t SP_MeasureCount

SP_MeasureCount

Direction: Down

Valid values:

Minimum: 1
Maximum: 9999

Description: Number of values to measure.

14.2.4.5 Get_MeasureCnt (VTT?)

Get the number of values to measure at several measure modes.

Parameter: int32_t SA_MeasureCount

SA_MeasureCount

Direction: Up

Valid values:

Minimum: 1
Maximum: 9999

Description: Number of values to measure.

14.2.4.6 Set_MeasureMode (MMD)

Set the measure mode at sensor.

Parameter: int32_t SP_MeasureMode

SP_MeasureMode

Direction: Down

Valid values:

- 0= Off
- 1= Continuous
- 2= Rising edge
- 3= Falling edge
- 4= High level
- 5= Low level

Description: Measure mode.

14.2.4.7 Get_MeasureMode (MMD?)

Retrieve the measure mode from sensor.

Parameter: int32_t SA_MeasureMode

SA_MeasureMode

Direction: Up

Valid values:

- 0= Off
- 1= Continuous
- 2= Rising edge
- 3= Falling edge
- 4= High level
- 5= Low level

Description: Measure mode.

14.2.4.8 Set_AvrType (AVT)

Set the averaging type at controller.

Parameter: int32_t SP_AvrType

SP_AvrType

Direction: Down

Valid values:

- 0= off
- 1= Moving average
- 2= Recursive average
- 3= Median

Description: Averaging type.

14.2.4.9 Get_AvrType (AVT?)

Retrieve the averaging type at controller.

Parameter: int32_t SA_AvrType

SA_AvrType

Direction: Up

Valid values:

- 0= off
- 1= Moving average
- 2= Recursive average
- 3= Median

Description: Averaging type at controller.

14.2.4.10 Set_AvrNbr (AVN)

Set the averaging number at controller.

Parameter: int32_t SP_AvrNbr

SP_AvrNbr

Direction: Down

Valid values:

- 0= 4 (moving + recursive), 3 (median)
- 1= 8 (moving + recursive), 5 (median)
- 2= 16 (moving + recursive), 7 (median)
- 3= 32 (moving + recursive), 9 (median)

Description: Averaging number.

14.2.4.11 Get_AvrNbr (AVN?)

Retrieve the averaging number at controller.

Parameter: int32_t SA_AvrNbr

SA_AvrNbr

Direction: Up

Valid values:

- 0= 4 (moving + recursive), 3 (median)
- 1= 8 (moving + recursive), 5 (median)
- 2= 16 (moving + recursive), 7 (median)
- 3= 32 (moving + recursive), 9 (median)

Description: Averaging number at controller.

14.2.4.12 Get_ControllerTemp (GCT)

Get temperature at controller.

Parameter: double SA_ControllerTemperature

SA_ControllerTemperature

Direction: Up

Valid values:

- Minimum:** -55.0
Maximum: 125.0

Unit: °C

Description: Temperature at controller.

14.2.4.13 Get_SensorTemp (GST)

Get temperature at sensor.

Parameter: double SA_SensorTemperature

SA_SensorTemperature

Direction: Up

Unit: °C

Description: Temperature at sensor.

14.2.5 Linearization

14.2.5.1 Get_CalibState (CST)

Get state of calibration progress.

Parameter: int32_t SA_CalibState SA_CalibState
Direction: Up
Valid values:
Minimum: 0
Maximum: 6
Description: Calibration state.

14.2.5.2 Set_CalibTarget (TAR)

Set target for calibration.

Parameter: int32_t SP_CalibTarget SP_CalibTarget
Direction: Down
Valid values:
 1= Ferro magnetic
 2= Non ferro magnetic
 4= Custom1
 8= Custom2
Description: Calibration target.

14.2.5.3 Get_CalibTarget (TAR?)

Get target for calibration.

Parameter: int32_t SA_CalibTarget SA_CalibTarget
Direction: Up
Valid values:
 1= Ferro magnetic
 2= Non ferro magnetic
 4= Custom1
 8= Custom2
Description: Calibration target.

14.2.5.4 Get_CalibProgress (BSY)

Get calibration progress.

Parameter: int32_t SA_CalibProgress SA_CalibProgress
Direction: Up
Valid values:
Minimum: 0
Maximum: 255
Description: Calibration progress.

14.2.5.5 Set_CalibStart (SMR)

Set start value of measurement range for calibration.

Parameter: int32_t SP_CalibStart SP_CalibStart
Direction: Down
Valid values:
Minimum: -1
Maximum: 10000
Unit: μm
Description: Start value of measurement range. -1 means do not send this parameter.

14.2.5.6 Set_CalibEnd (EMR)

Set end value of measurement range for calibration.

Parameter: int32_t SP_CalibEnd SP_CalibEnd
Direction: Down
Valid values:
Minimum: -1
Maximum: 10000
Unit: μm
Description: End value of measurement range. -1 means do not send this parameter.

14.2.5.7 Set_CalibMid (MMR)

Set mid value of measurement range for calibration.

Parameter: int32_t SP_CalibMid SP_CalibMid
Direction: Down
Valid values:
Minimum: -1
Maximum: 10000
Unit: μm
Description: Mid value of measurement range. -1 means do not send this parameter.

14.2.5.8 Exit_Calib (ECA)

Abort active calibration progress.

14.2.5.9 Set_FactoryCalib (FCA)

Load factory default calibration.

14.2.5.10 Set_Poti (WPT)

Set values of digital potis at controller.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_NullPoti SP_NullPoti

Direction: Down

Valid values:

Minimum: 1

Maximum: 4095

Description: Value of NullPoti.

Parameter: int32_t SP_GainPoti SP_GainPoti

Direction: Down

Valid values:

Minimum: 1

Maximum: 4095

Description: Value of GainPoti.

Parameter: int32_t SP_LinPoti SP_LinPoti

Direction: Down

Valid values:

Minimum: 1

Maximum: 4095

Description: Value of LinPoti.

14.2.5.11 Get_Poti (RPT)

Get values of digital potis from controller.

Parameter: int32_t SA_NullPoti SA_NullPoti

Direction: Up

Valid values:

Minimum: 1

Maximum: 4095

Description: Value of NullPoti.

Parameter: int32_t SA_GainPoti SA_GainPoti

Direction: Up

Valid values:

Minimum: 1

Maximum: 4095

Description: Value of GainPoti.

Parameter: int32_t SA_LinPoti SA_LinPoti

Direction: Up

Valid values:

Minimum: 1

Maximum: 4095

Description: Value of LinPoti.

14.2.5.12 Save_Poti (SPT)

Save values of digital potis at controller.

14.2.6 Internal commands

14.2.6.1 Update_Firmware

Update firmware version at controller.

Attention! This function can take approx. 3 minutes.

Parameter: Binary data SP_FirmwareFile

SP_FirmwareFile

Direction: Down

Valid values:

Firmware file as binary data.

Description: Firmware version, read from file.

Parameter: int32_t SA_Result

SA_Result

Direction: Up

Valid values:

0= Failed

1= Success

Description: Result of firmware update.

14.2.6.2 Get_BootLoaderMode (BOT)

Get state of boot loader at controller. This is an internal command. It should not be used by the customer.

Parameter: int32_t SA_BootLoaderMode

SA_BootLoaderMode

Direction: Up

Valid values:

0= Inactive

1= Active

Description: State of boot loader.

14.2.6.3 Get_BootLoaderVer (VBL)

Get version of boot loader at controller.

Parameter: String SA_BootLoaderVersion

SA_BootLoaderVersion

Direction: Up

Valid values:

Numeric value

Description: Version of boot loader.

14.2.6.4 Get_FirmwareVer (VFW)

Get version of firmware at controller.

Parameter: String SA_FirmwareVersion

SA_FirmwareVersion

Direction: Up

Valid values:

Numeric value

Description: Version of firmware.

14.2.6.5 Start_BootLoader (SBL)

Start boot loader at controller. This is an internal command. It should not be used by the customer.

14.2.6.6 Erase_Flash (EFL)

Erase flash at controller. This is an internal command. It should not be used by the customer.

14.2.6.7 Write_Sensor (WDS)

Write data to Sensor EEPROM at controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SP_SensorData

SP_SensorData

Direction: Down

Valid values:

128 bytes

Description: Binary data.

14.2.6.8 Read_Sensor (RDS)

Read data from Sensor EEPROM at controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SA_SensorData

SA_SensorData

Direction: Up

Valid values:

128 bytes

Description: Binary data.

14.2.6.9 Write_AnalogEEPROM (WA1+2)

Write data to analog EEPROM at controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SP_AnalogEEPROMData

SP_AnalogEEPROMData

Direction: Down

Valid values:

256 bytes

Description: Binary data.

14.2.6.10 Read_AnalogEEPROM (RA1+2)

Read data from analog EEPROM at controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SA_AnalogEEPROMData

SA_AnalogEEPROMData

Direction: Up

Valid values:

256 bytes

Description: Binary data.

14.2.6.11 Write_SensorParam (WSP)

Write sensor parameter controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SP_SensorParamData

SP_SensorParamData

Direction: Down

Valid values:

78 bytes

Description: Binary data.

14.2.6.12 Read_SensorParam (RSP)

Read sensor parameter from controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SA_SensorParamData

SA_SensorParamData

Direction: Up

Valid values:

78 bytes

Description: Binary data.

14.2.6.13 Write_AnalogParam (WP1+2)

Write analog parameter to controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SP_AnalogParamData

SP_AnalogParamData

Direction: Down

Valid values:

256 bytes

Description: Binary data.

14.2.6.14 Read_AnalogParam (RP1+2)

Read analog parameter from controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SA_AnalogParamData

SA_AnalogParamData

Direction: Up

Valid values:

256 bytes

Description: Binary data.

14.2.6.15 Exec_SysCalibOff (SCO)

Assign system calibration offset. This is an internal command. It should not be used by the customer.

14.2.6.16 Exec_SysCalibGain (SCG)

Assign system calibration gain. This is an internal command. It should not be used by the customer.

14.2.6.17 Write_Index (WRI)

Write index data to controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SP_IndexData

SP_IndexData

Direction: Down

Valid values:

20 bytes

Description: Binary data.

14.2.6.18 Read_Index (RDI)

Read index data from controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SA_IndexData

SA_IndexData

Direction: Up

Valid values:

20 bytes

Description: Binary data.

14.2.6.19 Write_IndexDigital (WRD)

Write digital index data to controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SP_DigitalIndexData

SP_DigitalIndexData

Direction: Down

Valid values:

128 bytes

Description: Binary data.

14.2.6.20 Read_IndexDigital (RDD)

Read digital index data from controller. This is an internal command. It should not be used by the customer.

Parameter: Binary data SA_DigitalIndexData

SA_DigitalIndexData

Direction: Up

Valid values:

128 bytes

Description: Binary data.

14.2.6.21 Get_ButtonState (CBT)

Read button states at controller. This is an internal command. It should not be used by the customer.

Parameter: int32_t SA_ButtonState SA_ButtonState
Direction: Up
Valid values:
 Bit combination of:
 0= Nothing pressed
 1 (Bit 0)= Button 1 pressed
 2 (Bit 1)= Button 2 pressed
 4 (Bit 2)= Button 3 pressed
Description: Button states.

14.2.6.22 Set_Multiplexer (MUX)

Set multiplexer state at controller. This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_Multiplexer SP_Multiplexer
Direction: Down
Valid values:
 0= -U-DEMO
 1= -U-TEMP
 2= U-PHASE
 3= -U-LOG-A
Description: Multiplexer state.

Parameter: double SA_RawValue SA_RawValue
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 65535.0
Description: Raw value of multiplexer channel.

Parameter: double SA_ScaledValue SA_ScaledValue
Direction: Up
Valid values:
Minimum: -5.0
Maximum: +5.0
Unit: V
Description: Scaled value of multiplexer channel.

15 Commands for capaNCDT and combiSENSOR sensors

15.1 Commands for DT6100

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [TCP/IP](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Status](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor, from 0 to 16777215.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Use_Defaults](#) with parameter [IP_Range](#)).

15.1.1 Measurement

15.1.1.1 Set_SRIndex (SRA)

Set the samplerate for data acquisition.

Parameter: int32_t SP_SRIndex

SP_SRIndex

Direction: Down

Valid values:

0= 2.60 Hz
1= 5.21 Hz
2= 10.42 Hz
3= 15.63 Hz
4= 26.04 Hz
5= 31.25 Hz
6= 52.08 Hz
7= 62.50 Hz
8= 104.17 Hz
9= 520.83 Hz
10= 1041.67 Hz
11= 2083.33 Hz
12= 3906.25 Hz
13= 7812.50 Hz

Description: Samplerate index

15.1.1.2 Get_SRIndex (SRA?)

Get the current samplerate.

Parameter: int32_t SA_SRIndex

SA_SRIndex

Direction: Up

Valid values:

- 0= 2.60 Hz
- 1= 5.21 Hz
- 2= 10.42 Hz
- 3= 15.63 Hz
- 4= 26.04 Hz
- 5= 31.25 Hz
- 6= 52.08 Hz
- 7= 62.50 Hz
- 8= 104.17 Hz
- 9= 520.83 Hz
- 10= 1041.67 Hz
- 11= 2083.33 Hz
- 12= 3906.25 Hz
- 13= 7812.50 Hz

Description: Samplerate index.

15.1.1.3 Set_Trigger (TRG)

Activate/disable the trigger at sensor.

Parameter: int32_t SP_TrgMode

SP_TrgMode

Direction: Down

Valid values:

- 0= disabled
- 1= active

Description: Trigger active/disabled.

15.1.1.4 Get_Trigger (TRG?)

Retrieve the trigger mode at sensor.

Parameter: int32_t SA_TrgMode

SA_TrgMode

Direction: Up

Valid values:

- 0= disabled
- 1= active

Description: Trigger active/disabled.

15.1.1.5 Get_Measure (GMD)

Retrieve one value from sensor, even if trigger is active.

15.1. Commands for DT6100

15.1.1.6 Set_AvrType (AVT)

Set the averaging type at controller.

Parameter: int32_t SP_AvrType SP_AvrType
Direction: Down
Valid values:
 0= off
 1= Moving average
 2= Mean (arithmetic)
 3= Median
Description: Averaging type.

15.1.1.7 Get_AvrType (AVT?)

Retrieve the averaging type at controller.

Parameter: int32_t SA_AvrType SA_AvrType
Direction: Up
Valid values:
 0= off
 1= Moving average
 2= Mean (arithmetic)
 3= Median
Description: Averaging type at controller.

15.1.1.8 Set_AvrNbr (AVN)

Set the averaging number at controller.

Parameter: int32_t SP_AvrNbr SP_AvrNbr
Direction: Down
Valid values:
Minimum: 2
Maximum: 8
Description: Averaging number.

15.1.1.9 Get_AvrNbr (AVN?)

Retrieve the averaging number at controller.

Parameter: int32_t SA_AvrNbr SA_AvrNbr
Direction: Up
Valid values:
Minimum: 2
Maximum: 8
Description: Averaging number at controller.

15.1.2 Linearization

15.1.2.1 Set_LinMode (LIN)

Set the linearisation mode for sensor.

Parameter: int32_t SP_LinMode

SP_LinMode

Direction: Down

Valid values:

- 0= off
- 1= offset correction
- 2= 2 point linearization
- 3= 3 point linearization
- 4= 5 point linearization
- 5= 10 point linearization

Description: Linearisation mode.

15.1.2.2 Get_LinMode (LIN?)

Retrieve the linearisation mode for sensor.

Parameter: int32_t SA_LinMode

SA_LinMode

Direction: Up

Valid values:

- 0= off
- 1= offset correction
- 2= 2 point linearization
- 3= 3 point linearization
- 4= 5 point linearization
- 5= 10 point linearization

Description: Linearisation mode.

15.1.2.3 Set_LinPoint (SLP)

Set a linearisation point at sensor.

Parameter: int32_t SP_LinPos

SP_LinPos

Direction: Down

Valid values:

- 1= at 10%
- 2= at 20%
- 3= at 30%
- 4= at 40%
- 5= at 50%
- 6= at 60%
- 7= at 70%
- 8= at 80%
- 9= at 90%
- 10= at 100%

Description: Linearisation position to be set for.

15.1.2.4 Get_LinPoint (GLP)

Get a linearisation point at sensor.

Parameter: int32_t SP_LinPos

SP_LinPos

Direction: Down

Valid values:

- 1= at 10%
- 2= at 20%
- 3= at 30%
- 4= at 40%
- 5= at 50%
- 6= at 60%
- 7= at 70%
- 8= at 80%
- 9= at 90%
- 10= at 100%

Description: Linearisation position to be get for.

Parameter: double SA_LinPoint

SA_LinPoint

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 16777215 (0xffffffff)

Description: Linearisation point.

15.1.3 Information

15.1.3.1 Get_Status (STS)

Retrieve detailed information about the sensor.

Parameter: int32_t SA_SRIndex

SA_SRIndex

Direction: Up

Valid values:

- 0= 2.60 Hz
- 1= 5.21 Hz
- 2= 10.42 Hz
- 3= 15.63 Hz
- 4= 26.04 Hz
- 5= 31.25 Hz
- 6= 52.08 Hz
- 7= 62.50 Hz
- 8= 104.17 Hz
- 9= 520.83 Hz
- 10= 1041.67 Hz
- 11= 2083.33 Hz
- 12= 3906.25 Hz
- 13= 7812.50 Hz

Description: Samplerate index.

15.1. Commands for DT6100

Parameter: int32_t SA_AvrType	SA_AvrType
Direction: Up	
Valid values:	
0= off	
1= Moving average	
2= Mean (arithmetic)	
3= Median	
Description: Averaging type at controller.	
Parameter: int32_t SA_AvrNbr	SA_AvrNbr
Direction: Up	
Valid values:	
Minimum: 2	
Maximum: 8	
Description: Averaging number at controller.	
Parameter: int32_t SA_TrgMode	SA_TrgMode
Direction: Up	
Valid values:	
0= disabled	
1= active	
Description: Trigger active/disabled.	
Parameter: int32_t SA_LinMode	SA_LinMode
Direction: Up	
Valid values:	
0= off	
1= offset correction	
2= 2 point linearization	
3= 3 point linearization	
4= 5 point linearization	
5= 10 point linearization	
Description: Linearisation mode.	

15.1.3.2 Get_Version (VER)

Retrieve the sensor software version.

Parameter: String SA_Version	SA_Version
Direction: Up	
Description: Software version of the controller.	

15.1.4 Parameter management

15.1.4.1 Save_Setup (SSU)

Save the current setup of controller to flash.

15.1.4.2 Read_Setup (RSU)

Read the setup from controller flash.

15.1.4.3 Factory_Defaults (FDE)

Restore the sensor to factory defaults.

To save the default values call Save_Setup too. The new parameters are returned.

Parameter: int32_t SA_SRIndex SA_SRIndex

Direction: Up

Valid values:

- 0= 2.60 Hz
- 1= 5.21 Hz
- 2= 10.42 Hz
- 3= 15.63 Hz
- 4= 26.04 Hz
- 5= 31.25 Hz
- 6= 52.08 Hz
- 7= 62.50 Hz
- 8= 104.17 Hz
- 9= 520.83 Hz
- 10= 1041.67 Hz
- 11= 2083.33 Hz
- 12= 3906.25 Hz
- 13= 7812.50 Hz

Description: Samplerate index.

Parameter: int32_t SA_AvrType SA_AvrType

Direction: Up

Valid values:

- 0= off
- 1= Moving average
- 2= Mean (arithmetic)
- 3= Median

Description: Averaging type at controller.

Parameter: int32_t SA_AvrNbr SA_AvrNbr

Direction: Up

Valid values:

- Minimum:** 2
- Maximum:** 8

Description: Averaging number at controller.

Parameter: int32_t SA_TrgMode SA_TrgMode

Direction: Up

Valid values:

- 0= disabled
- 1= active

Description: Trigger active/disabled.

Parameter: int32_t SA_LinMode SA_LinMode

Direction: Up

Valid values:

- 0= off
- 1= offset correction
- 2= 2 point linearization
- 3= 3 point linearization
- 4= 5 point linearization
- 5= 10 point linearization

Description: Linearisation mode.

15.1.5 Internal commands

15.1.5.1 Set_Coefficient (SCO)

Set coefficient at controller. This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_CoeffParam SP_CoeffParam

Direction: Down

Valid values:

Minimum: 0

Maximum: 16

Description: Coeffizient parameter.

Parameter: double SP_Coeffizient SP_Coeffizient

Direction: Down

Valid values:

Minimum: 0

Maximum: 16777215 (0xffffffff)

Description: Coeffizient.

15.1.5.2 Get_Coefficient (GCO)

Get coefficient from controller. This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_CoeffParam SP_CoeffParam

Direction: Down

Valid values:

Minimum: 0

Maximum: 16

Description: Coeffizient parameter.

Parameter: double SA_Coeffizient SA_Coeffizient

Direction: Up

Valid values:

Minimum: 0

Maximum: 16777215 (0xffffffff)

Description: Coeffizient.

15.1.6 Special commands

15.1.6.1 Use_Defaults

This command tells the driver to use default values to operate with sensor data. If some parameters are not specified they are not changed. The sensor is not affected by this command.

Parameter: double IP_Range IP_Range

Direction: Down

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Unit: μm or mm

Description: Tells the driver the range of sensor. It is used to scale the raw sensor values into μm or mm. If it is zero, no scaling is done.

15.1.6.2 Get_DrvSetting

Returns the current settings of the driver used for operating with sensor data. It is the opposite of Use_Defaults. The sensor is not affected by this command.

Parameter: double IA_Range

IA_Range

Direction: Up

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Unit: μm or mm

Description: The range of sensor, used by driver to scale values into μm or mm.

15.2 Commands for DT6120

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- RS232 (RS485/USB converter).

If first bit of IP_AutomaticMode is set (1), MEDAQLib calls automatically sensor command [Read_AllBlocks](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate and to assign values.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor, range is depending on data format.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Read_AllBlocks](#)).

The values of selected channels are filled in the arrays one after another. Each array always starts with first selected channel.

15.2.1 Measurement

15.2.1.1 Set_Samplerate

Set the samplerate for data acquisition.

Parameter: double SP_Measrate

SP_Measrate

Direction: Down

Valid values:

5

10

20

40

80

160

320

639.79526551503523

1000

2000

Unit: Hz

Description: Desired samplerate

15.2. Commands for DT6120

Parameter: double SA_Measrate SA_Measrate

Direction: Up

Valid values:

- 5
- 10
- 20
- 40
- 80
- 160
- 320
- 639.79526551503523
- 1000
- 2000

Unit: Hz

Description: Real samplerate at sensor

15.2.1.2 Get_Samplerate

Get the current samplerate.

Parameter: double SA_Measrate SA_Measrate

Direction: Up

Valid values:

- 5
- 10
- 20
- 40
- 80
- 160
- 320
- 639.79526551503523
- 1000
- 2000

Unit: Hz

Description: Real samplerate at sensor

15.2.1.3 Set_Averaging

Set data averaging at sensor.

Parameter: int32_t SP_AveragingType SP_AveragingType

Direction: Down

Valid values:

- 0= off
- 1= Moving average
- 2= Mean (arithmetic)
- 3= Median
- 4= Dynamic noise rejection

Description: Averaging type.

Parameter: int32_t SP_AveragingValue SP_AveragingValue
Direction: Down
Valid values:
Minimum: 0
Maximum: 2147483647 (INT32_MAX)
Description: Averaging number (depending on averaging type).

15.2.1.4 Get_Averaging

Retrieve the averaging type at sensor.

Parameter: int32_t SA_AveragingType SA_AveragingType
Direction: Up
Valid values:
 0= off
 1= Moving average
 2= Mean (arithmetic)
 3= Median
 4= Dynamic noise rejection
Description: Averaging type at sensor.

Parameter: int32_t SA_AveragingValue SA_AveragingValue
Direction: Up
Valid values:
Minimum: 0
Maximum: 2147483647 (INT32_MAX)
Description: Averaging number (depending on averaging type).

15.2.1.5 Get_Measure

Retrieve a data block from controller. To get the values from MEDAQLib, use the functions [Poll](#) or [TransferData](#).

15.2.2 Data output

15.2.2.1 Set_Range

Write the measurement range to sensor.

Parameter: int32_t SP_Chан SP_Chан
Direction: Down
Valid values:
 0
Description: Channel number, always 0 for this sensor.

Parameter: int32_t SP_PageNo SP_PageNo
Direction: Down
Valid values:
 1
Description: Sensor page number, always 1 for this sensor.

Parameter: double SP_Range SP_Range
Direction: Down
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Unit: μm
Description: Range of sensor.

15.2.3 Interfaces

15.2.3.1 Set_Baudrate

Set baudrate of sensor for serial RS485 communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_SensorBaudrate SP_SensorBaudrate
Direction: Down
Valid values:
 921600
 460800
 230400
 115200
 9600
Unit: Baud
Description: Baudrate of sensor.

Parameter: int32_t SP_Store SP_Store
Direction: Down
Valid values:
 0 = Temporary (in RAM)
 1 = Permanent (in Flash)
Description: Specify if the baudrate should be changed persistently.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate
Direction: Up
Valid values:
 921600
 460800
 230400
 115200
 9600
Unit: Baud
Description: Real baudrate at sensor.

15.2.3.2 Get_Baudrate

Get baudrate of controller for serial RS485 communication.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate
Direction: Up
Valid values:
 921600

460800
230400
115200
9600

Unit: Baud

Description: Real baudrate at sensor.

15.2.3.3 Set_SensorAddress

Changes the address at sensor which is used to communicate over RS485 bus.

Parameter: int32_t SP_SensorAddress

SP_SensorAddress

Direction: Down

Valid values:

Minimum: 1
Maximum: 126

Description: Address of sensor.

15.2.4 Information

15.2.4.1 Get_SensorInfo

Retrieve information about the connected sensor.

Parameter: int32_t SP_Chан

SP_Chан

Direction: Down

Valid values:

0

Description: Channel number, always 0 for this sensor.

Parameter: int32_t SP_PageNo

SP_PageNo

Direction: Down

Valid values:

1

Description: Sensor page number, always 1 for this sensor.

Parameter: String SA_ArticleNumber

SA_ArticleNumber

Direction: Up

Valid values:

Numeric value

Description: Article number of the connected sensor.

Parameter: String SA_SensorName

SA_SensorName

Direction: Up

Description: Name of the connected sensor.

Parameter: String SA_SerialNumber

SA_SerialNumber

Direction: Up

Valid values:

Numeric value

Description: Serial number of the connected sensor.

Parameter: double SA_Offset	SA_Offset
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Unit: See parameter SA_Unit	
Description: Offset of the connected sensor.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Unit: See parameter SA_Unit	
Description: Range of connected sensor.	
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description: Unit of parameter SA_Offset and SA_Range, e.g. μm or mm.	

15.2.4.2 Get_ControllerInfo

Retrieve information about the controller part of sensor.

Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Description: Article number of the controller part of sensor.	
Parameter: String SA_ControllerName	SA_ControllerName
Direction: Up	
Description: Name of the controller part of sensor.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the controller part of sensor.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the controller part of sensor.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the controller part of sensor.	

15.2.4.3 Read_AllBlocks

Tell MEDAQLib to read all sensor information.

15.3 Commands for DT62xx, KSS64xx and DT65xx

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [TCP/IP](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Status](#), [Get_SampleTime](#) (only DT6530), [Get_RawDataRanges](#) and [Get_ChannelInfos](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate and to assign values.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor, either from 0 to 16777215 (for DT65x0) or from -2147483648 (INT32_MIN) to 2147483647 (INT32_MAX) or from 0 to 4294967295 (UINT32_MAX) or from -3.402823466e+38 (-FLT_MAX) to 3.402823466e+38 (FLT_MAX).
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_RawDataRanges](#) and [Get_ChannelInfos](#)) (not for DT6500) or [Use_Defaults](#) parameter [IP_Range1..8](#) for DT6500).

The values of selected channels are filled in the arrays one after another. Each array always starts with first selected channel.

15.3.1 User Level

15.3.1.1 Logout (LGO)

Change user level to user at web interface.

This command is not available for old controller DT6500.

15.3.1.2 Login (LGI)

Change user level to setup at web interface.

This command is not available for old controller DT6500.

Parameter: String SP_Password

SP_Password

Direction: Down

Description: Valid password to login.

15.3.1.3 Set_Password (PWD)

Change the password for login.

This command is not available for old controller DT6500.

Parameter: String SP_OldPassword

SP_OldPassword

Direction: Down

Description: Old password.

Parameter: String SP_NewPassword SP_NewPassword

Direction: Down

Description: New password.

15.3.2 Measurement

15.3.2.1 Set_SRIndex (SRA)

Set the samplerate for data acquisition.

This command is only available for controller DT65x0.

Parameter: int32_t SP_SRIndex

SP_SRIndex

Direction: Down

Valid values:

0= 2.60 Hz
 1= 5.21 Hz
 2= 10.42 Hz
 3= 15.63 Hz
 4= 26.04 Hz
 5= 31.25 Hz
 6= 52.08 Hz
 7= 62.50 Hz
 8= 104.17 Hz
 9= 520.83 Hz
 10= 1041.67 Hz
 11= 2083.33 Hz
 12= 3906.25 Hz
 13= 7812.50 Hz

Description: Samplerate index

15.3.2.2 Get_SRIndex (SRA?)

Get the current samplerate.

This command is only available for controller DT65x0.

Parameter: int32_t SA_SRIndex

SA_SRIndex

Direction: Up

Valid values:

0= 2.60 Hz
 1= 5.21 Hz
 2= 10.42 Hz
 3= 15.63 Hz
 4= 26.04 Hz
 5= 31.25 Hz
 6= 52.08 Hz
 7= 62.50 Hz
 8= 104.17 Hz
 9= 520.83 Hz
 10= 1041.67 Hz
 11= 2083.33 Hz
 12= 3906.25 Hz
 13= 7812.50 Hz

Description: Samplerate index.

15.3.2.3 Set_SampleTime (STI)

Set the sample time for data acquisition.

This command is not available for old controller DT6500. It replaces the obsolete sensor command Set_SRIndex.

Parameter: double SP_SampleTime SP_SampleTime

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 4294967040.0

Unit: μ s

Description: Desired sample time

Parameter: double SA_SampleTime SA_SampleTime

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967040.0

Unit: μ s

Description: Real sample time at controller

15.3.2.4 Get_SampleTime (STI?)

Get the current sample time.

This command is not available for old controller DT6500. It replaces the obsolete sensor command Get_SRIndex.

Parameter: double SA_SampleTime SA_SampleTime

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967040.0

Unit: μ s

Description: Real sample time at controller

15.3.2.5 Set_Trigger (TRG)

Activate/disable the trigger at controller.

Parameter: int32_t SP_TrMode SP_TrMode

Direction: Down

Valid values:

0= Off

1= Rising edge

2= High level

3= Gate at rising edge

Description: Trigger active/disabled.

15.3.2.6 Get_Trigger (TRG?)

Retrieve the trigger mode at controller.

Parameter: int32_t SA_TrgMode SA_TrgMode
Direction: Up
Valid values:
 0= Off
 1= Rising edge
 2= High level
 3= Gate at rising edge
Description: Trigger active/disabled.

15.3.2.7 Get_Measure (GMD)

Retrieve one value from controller, even if trigger is active. To get the values from MEDAQLib, use the functions [Poll](#) or [TransferData](#).

15.3.2.8 Set_AvrType (AVT)

Set the averaging type at controller.

Parameter: int32_t SP_AvrType SP_AvrType
Direction: Down
Valid values:
 0= off
 1= Moving average
 2= Mean (arithmetic)
 3= Median
 4= Dynamic noise rejection
Description: Averaging type.

15.3.2.9 Get_AvrType (AVT?)

Retrieve the averaging type at controller.

Parameter: int32_t SA_AvrType SA_AvrType
Direction: Up
Valid values:
 0= off
 1= Moving average
 2= Mean (arithmetic)
 3= Median
 4= Dynamic noise rejection
Description: Averaging type at controller.

15.3.2.10 Set_AvrNbr (AVN)

Set the averaging number at controller.

Parameter: int32_t SP_AvrNbr SP_AvrNbr
Direction: Down
Valid values:
Minimum: 2
Maximum: 7 (DT6536) or 8 (any other)
Description: Averaging number.

15.3.2.11 Get_AvrNbr (AVN?)

Retrieve the averaging number at controller.

Parameter: int32_t SA_AvrNbr SA_AvrNbr
Direction: Up
Valid values:
Minimum: 2
Maximum: 7 (DT6536) or 8 (any other)
Description: Averaging number at controller.

15.3.2.12 Set_AnalogLowPass (ALP)

Set the analog lowpass filter at controller.

This command is only available for DT6200.

Parameter: int32_t SP_LowPass SP_LowPass
Direction: Down
Valid values:
 0 = off
 1 = on
Description: Analog lowpass filter.

15.3.2.13 Get_AnalogLowPass (ALP?)

Retrieve the analog lowpass filter from controller.

This command is only available for DT6200.

Parameter: int32_t SA_LowPass SA_LowPass
Direction: Up
Valid values:
 0 = off
 1 = on
Description: Analog lowpass filter.

15.3.3 Data output

15.3.3.1 ChannelStatus (CHS)

Retrieve the available channels at controller.

Parameter: int32_t SA_ChExist1..4 (DT6200/KSS64xx) or 8 (DT65x0) or 6 (DT6536)

Direction: Up (DT6200/KSS64xx) or 8 (DT65x0) or 6 (DT6536)
Valid values:

- 0= Channel not available
- 1= Measured channel
- 2= Mathematic channel

Description: Channel 1 to 4 (DT6200/KSS64xx) or 8 (DT65x0) or 6 (DT6536) is available at controller.

15.3.3.2 Set_Channel (CHT)

Set the channels to transmit from controller.

This command is only available for DT65x0.

For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_ChTransmit1..8

SP_ChTransmit1..8

Direction: Down

Valid values:

- 0= no
- 1= yes

Description: Channel 1 to 8 is transmitted.

15.3.3.3 Get_Channel (CHT?)

Get the channels transmitted from controller.

This command is only available for DT65x0.

Parameter: int32_t SA_ChTransmit1..8

SA_ChTransmit1..8

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: Channel 1 to 8 is transmitted.

15.3.3.4 Set_Range (MRA)

Write the measurement range for a channel to controller.

This command is not available for old controller DT6500 and KSS64xx.

Parameter: int32_t SP_ChAn

SP_ChAn

Direction: Down

Valid values:

Minimum: 1

Maximum: 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)

Description: Sensor channel to set it's range.

Parameter: double SP_Range SP_Range
Direction: Down
Valid values:
Minimum: -4294967040.0
Maximum: 4294967040.0
Unit: μm
Description: Range of sensor channel.

15.3.3.5 Get_RawDataRange (MDF)

Read the raw data range for a channel from controller. The range is used by MEDAQLib for scaling raw data.

This command is not available for old controller DT6500.

Parameter: int32_t SP_Chан SP_Chан
Direction: Down
Valid values:
Minimum: 1
Maximum: 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)
Description: Channel to read the raw data range for.

Parameter: double SA_RawRangeMin SA_RawRangeMin
Direction: Up
Valid values:
Minimum: -2147483648.0 (INT32_MIN)
Maximum: 2147483647.0 (INT32_MAX)
Description: Minimum raw data range of channel.

Parameter: double SA_RawRangeMax SA_RawRangeMax
Direction: Up
Valid values:
Minimum: -2147483648.0 (INT32_MIN)
Maximum: 2147483647.0 (INT32_MAX)
Description: Maximum raw data range of channel.

15.3.3.6 Get_RawDataRanges

Calls the sensor command Get_RawDataRange for any requested channel.

Parameter: int32_t SP_Complete SP_Complete
Direction: Down
Valid values:
 0 = FALSE
 1 = TRUE
Description: Specifies if any possible channel should be requested or only known channels (from former call to [ChannelStatus](#) or [Get_Status](#)).

Parameter: double SA_RawRangeMin1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536) SA_RawRangeMin1..4
(DT6200/KSS64xx) or 8
(DT6530) or 6 (DT6536)
Direction: Up
Valid values:
Minimum: -2147483648.0 (INT32_MIN)
Maximum: 2147483647.0 (INT32_MAX)
Description: Minimum raw data range of channel 1 to 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536).

Parameter: double SA_RawRangeMax_{1..4} (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)

Direction: Up

Valid values:

- Minimum:** -2147483648.0 (INT32_MIN)
- Maximum:** 2147483647.0 (INT32_MAX)

Description: Maximum raw data range of channel 1 to 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536).

SA_RawRangeMax_{1..4}
(DT6200/KSS64xx) or 8
(DT6530) or 6 (DT6536)

15.3.4 Display

15.3.4.1 Set_Display (DIS)

Set the display settings at controller.
This command is only available for DT65x0.

Parameter: int32_t SP_ShowChannels

SP_ShowChannels

Direction: Down

Valid values:

- 0= none
- 1= all
- 2= selected

Description: Channels to be displayed.

Parameter: int32_t SP_ShowLinearized

SP_ShowLinearized

Direction: Down

Valid values:

- 0= off
- 1= on

Description: Channels should be displayed linearized.

15.3.4.2 Get_Display (DIS?)

Retrieve the display settings at controller.
This command is only available for DT65x0.

Parameter: int32_t SA_ShowChannels

SA_ShowChannels

Direction: Up

Valid values:

- 0= none
- 1= all
- 2= selected

Description: Channels to be displayed.

Parameter: int32_t SA_ShowLinearized

SA_ShowLinearized

Direction: Up

Valid values:

- 0= off
- 1= on

Description: Channels should be displayed linearized.

15.3.5 Math

15.3.5.1 Set_MathFunction (SMF)

Set mathematic function at controller.

The result of the math function is transmitted like normal sensor data at a specific channel.

This command is not available for KSS64xx.

Because of the mathematic operations (multiplication, addition) at DT65x0, the raw values can exceed the range of 24 bit. To avoid this, the controller does automatically divide the results by eight (Result = Result/8). MEDAQLib does output the raw values as it comes from sensor and multiply the scaled values by eight to compensate the dividing at controller.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_Chан

SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)

Description: Channels which returns the result.

Parameter: double SP_Offset

SP_Offset

Direction: Down

Valid values:

Minimum: -16777215 (-0xffffffff)

Maximum: 16777215 (0xffffffff)

Description: Offset to be added to result.

Parameter: double SP_FactorCh1..4 (DT6200) or 8 (DT6530) or 6 (DT6536)

SP_FactorCh1..4 (DT6200)
or 8 (DT6530) or 6
(DT6536)

Direction: Down

Valid values:

Minimum: -9.9

Maximum: 9.9

Description: Multiplication factor for channel 1 to 4 (DT6200) or 8 (DT6530) or 6 (DT6536).

15.3.5.2 Get_MathFunction (GMF)

Get mathematic function from controller. This command is not available for KSS64xx.

Parameter: int32_t SP_Chан

SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)

Description: Channels for which the function should be read.

Parameter: double SA_Offset

SA_Offset

Direction: Up

Valid values:

Minimum: -16777215 (-0xffffffff)

Maximum: 16777215 (0xffffffff)

Description: Offset to be added to result.

Parameter: double SA_FactorCh1..4 (DT6200) or 8 (DT6530) or 6 (DT6536)

Direction: Up

Valid values:

- Minimum:** -9.9
- Maximum:** 9.9

Description: Multiplication factor for channel 1 to 4 (DT6200) or 8 (DT6530) or 6 (DT6536). Not for KSS64xx.

SA_FactorCh1..4 (DT6200)
or 8 (DT6530) or 6
(DT6536)

15.3.5.3 Clr_MathFunction (CMF)

Clears mathematic function at controller. This command is not available for KSS64xx.

Parameter: int32_t SP_Chан

SP_Chан

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)

Description: Channels for which the function should be cleared.

15.3.5.4 Set_ThicknessMeasure (THM)

Set thickness measurement at controller.

This command is only available for KSS64xx.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_ThicknessMeasureEnable

SP_ThicknessMeasureEnable

Direction: Down

Valid values:

- 0= Disabled
- 1= Enabled

Description: Specify if thickness measurement is enabled.

Parameter: double SP_RelativePermittivity

SP_RelativePermittivity

Direction: Down

Valid values:

- Minimum:** 1.00999998
- Maximum:** 4294967040.0

Description: Relative permittivity of target

Parameter: double SP_OffsetCorrection

SP_OffsetCorrection

Direction: Down

Valid values:

- Minimum:** -4294967040.0
- Maximum:** 4294967040.0

Unit: μm

Description: Offset correction of thickness measurement

Parameter: double SP_Range SP_Range
Direction: Down
Valid values:
Minimum: 0.0000001
Maximum: 4294967040.0
Unit: μm
Description: Range of thickness measurement

15.3.5.5 Get_ThicknessMeasure (THM?)

Get thickness measurement at controller.
 This command is only available for KSS64xx.

Parameter: int32_t SA_ThicknessMeasureEnable SA_ThicknessMeasureEnable

Direction: Up

Valid values:

0= Disabled
 1= Enabled

Description: Specify if thickness measurement is enabled.

Parameter: double SA_RelativePermittivity SA_RelativePermittivity

Direction: Up

Valid values:

Minimum: 1.00999998
Maximum: 4294967040.0

Description: Relative permittivity of target

Parameter: double SA_OffsetCorrection SA_OffsetCorrection

Direction: Up

Valid values:

Minimum: -4294967040.0
Maximum: 4294967040.0

Unit: μm

Description: Offset correction of thickness measurement

Parameter: double SA_Range SA_Range

Direction: Up

Valid values:

Minimum: 0.0000001
Maximum: 4294967040.0

Unit: μm

Description: Range of thickness measurement

15.3.6 Interfaces

15.3.6.1 Set_DataPort (SDP)

Set the TCP/IP data port at controller.
 This command is not available for old controller DT6500.

Parameter: int32_t SP_DataPort SP_DataPort

Direction: Down

Valid values:

Minimum: 1024
Maximum: 65535

Description: TCP/IP data port at controller.

15.3.6.2 Get_DataPort (GDP)

Retrieve the TCP/IP data port from controller.

This command is not available for old controller DT6500.

Parameter: int32_t SA_DataPort

SA_DataPort

Direction: Up

Valid values:

Minimum: 1024

Maximum: 65535

Description: TCP/IP data port at controller.

15.3.6.3 Set_IPConfiguration (IPS)

Set the IP configuration at controller.

This command is not available for old controller DT6500.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_DHCPEnabled

SP_DHCPEnabled

Direction: Down

Valid values:

0 = FALSE

1 = TRUE

Description: Specify if controller should use a static IP address or ask for IP at DHCP server (dynamic IP address).

Parameter: String SP_Address

SP_Address

Direction: Down

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_SubnetMask

SP_SubnetMask

Direction: Down

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_Gateway

SP_Gateway

Direction: Down

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: The default gateway must be specified if the controller should communicate with peers in foreign subnets. This parameter is only evaluated on static IP assignment.

15.3.6.4 Get_IPConfiguration (IPS?)

Get the IP configuration at controller.
 This command is not available for old controller DT6500.

Parameter: int32_t SA_DHCPEnabled SA_DHCPEnabled

Direction: Up

Valid values:

0= FALSE

1= TRUE

Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).

Parameter: String SA_Address SA_Address

Direction: Up

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.

Parameter: String SA_SubnetMask SA_SubnetMask

Direction: Up

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.

Parameter: String SA_Gateway SA_Gateway

Direction: Up

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

15.3.6.5 Set_EthernetMode (IFC)

Switches ethernet mode between Ethernet and Ethercat. The controller must be rebooted to apply this setting.
 This command is not available for old controller DT6500.

Parameter: int32_t SP_EthernetMode SP_EthernetMode

Direction: Down

Valid values:

0= Ethernet

1= Ethercat

Description: Ethernet mode.

15.3.6.6 Get_EthernetMode (IFC?)

Get ethernet mode of controller.

This command is not available for old controller DT6500.

Parameter: int32_t SA_EthernetMode

SA_EthernetMode

Direction: Up

Valid values:

0= Ethernet

1= Ethercat

Description: Ethernet mode.

15.3.6.7 Set_AppLanguage (LNG)

Set language of web interface at controller.

This command is not available for old controller DT6500.

Parameter: int32_t SP_ApplicationLanguage

SP_ApplicationLanguage

Direction: Down

Valid values:

0= System

1= English

2= German

Description: Language of web interface.

15.3.6.8 Get_AppLanguage (LNG?)

Get language of web interface from controller.

This command is not available for old controller DT6500.

Parameter: int32_t SA_ApplicationLanguage

SA_ApplicationLanguage

Direction: Up

Valid values:

0= System

1= English

2= German

Description: Language of web interface.

15.3.7 Linearization

15.3.7.1 Set_LinMode (LIN)

Set the linearisation mode for a channel at controller.

For KSS64xx this is an internal command and should not be used by the customer.

Parameter: int32_t SP_Chан

SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 4 (DT6200/KSS64xx) or 8 (DT65x0) or 6 (DT6536)

Description: Channels to be linearized.

Parameter: int32_t SP_LinMode SP_LinMode

Direction: Down

Valid values:

- 0= off
- 1= offset correction
- 2= 2 point linearization
- 3= 3 point linearization
- 4= 5 point linearization
- 5= 10 point linearization

Description: Linearisation mode.

15.3.7.2 Get_LinMode (LIN?)

Retrieve the linearisation mode for all channels at controller.

For KSS64xx this is an internal command and should not be used by the customer.

Parameter: int32_t SA_LinMode1..4 (DT6200) or 8 (DT65x0) or 6 (DT6536)

SA_LinMode1..4 (DT6200)

or 8 (DT65x0) or 6
(DT6536)

Direction: Up

Valid values:

- 0= off
- 1= offset correction
- 2= 2 point linearization
- 3= 3 point linearization
- 4= 5 point linearization
- 5= 10 point linearization

Description: Linearisation mode for channel 1 to 4 (DT6200) or 8 (DT65x0) or 6 (DT6536).

15.3.7.3 Set_LinPoint (SLP)

Set a linearisation point for a channel at controller.

For KSS64xx this is an internal command and should not be used by the customer.

Parameter: int32_t SP_Chан

SP_Chан

Direction: Down

Valid values:

- Minimum:** 1
Maximum: 4 (DT6200/KSS64xx) or 8 (DT65x0) or 6 (DT6536)

Description: Channel to be set for.

Parameter: int32_t SP_LinPos

SP_LinPos

Direction: Down

Valid values:

- 0= at 0%
- 1= at 10%
- 2= at 20%
- 3= at 30%
- 4= at 40%
- 5= at 50%
- 6= at 60%
- 7= at 70%
- 8= at 80%
- 9= at 90%
- 10= at 100%

Description: Linearisation position to be set for.

15.3.7.4 Set_LinPointOffline (SLP)

Set a linearisation point for a channel at controller (specified at command).

This command is not available for old controller DT6500.

For KSS64xx this is an internal command and should not be used by the customer.

Parameter: int32_t SP_Chан SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 4 (DT6200/KSS64xx) or 8 (DT65x0) or 6 (DT6536)

Description: Channel to be set for.

Parameter: int32_t SP_LinPos SP_LinPos

Direction: Down

Valid values:

0= at 0%

1= at 10%

2= at 20%

3= at 30%

4= at 40%

5= at 50%

6= at 60%

7= at 70%

8= at 80%

9= at 90%

10= at 100%

Description: Linearisation position to be set for.

Parameter: double SP_LinPoint SP_LinPoint

Direction: Down

Valid values:

Minimum: -16777215 (-0xffffffff)

Maximum: 16777215 (0xffffffff)

Description: Linearisation point.

15.3.7.5 Get_LinPoint (GLP)

Get a linearisation point for a channel at controller.

For KSS64xx this is an internal command and should not be used by the customer.

Parameter: int32_t SP_Chан SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 4 (DT6200/KSS64xx) or 8 (DT65x0) or 6 (DT6536)

Description: Channel to be get for.

Parameter: int32_t SP_LinPos SP_LinPos

Direction: Down

Valid values:

- 0= at 0%
- 1= at 10%
- 2= at 20%
- 3= at 30%
- 4= at 40%
- 5= at 50%
- 6= at 60%
- 7= at 70%
- 8= at 80%
- 9= at 90%
- 10= at 100%

Description: Linearisation position to be get for.

Parameter: double SA_LinPoint SA_LinPoint

Direction: Up

Valid values:

- Minimum:** -268435455 (-0xffffffff)
- Maximum:** 268435455 (0xffffffff)

Description: Linearisation point.

15.3.8 Information

15.3.8.1 Get_Status (STS)

Retrieve detailed information about the controller.

Parameter: int32_t SA_SRIndex SA_SRIndex

Direction: Up

Valid values:

- 0= 2.60 Hz
- 1= 5.21 Hz
- 2= 10.42 Hz
- 3= 15.63 Hz
- 4= 26.04 Hz
- 5= 31.25 Hz
- 6= 52.08 Hz
- 7= 62.50 Hz
- 8= 104.17 Hz
- 9= 520.83 Hz
- 10= 1041.67 Hz
- 11= 2083.33 Hz
- 12= 3906.25 Hz
- 13= 7812.50 Hz

Description: Samplerate index. Only for DT65x0

Parameter: double SA_SampleTime SA_SampleTime

Direction: Up

Valid values:

- Minimum:** 0.0
- Maximum:** 4294967040.0

Unit: μ s

Description: Real sample time at controller. Not for DT65x0

Parameter: int32_t SA_AvrType SA_AvrType

Direction: Up

Valid values:

- 0= off
- 1= Moving average
- 2= Mean (arithmetic)
- 3= Median
- 4= Dynamic noise rejection

Description: Averaging type at controller.

Parameter: int32_t SA_AvrNbr SA_AvrNbr

Direction: Up

Valid values:

Minimum: 2

Maximum: 7 (DT6536) or 8 (any other)

Description: Averaging number at controller.

Parameter: int32_t SA_ChExist1..4 (DT6200/KSS64xx) or 8 (DT65x0) or 6 (DT6536) SA_ChExist1..4 (DT6200/KSS64xx) or 8 (DT65x0) or 6 (DT6536)

Direction: Up

Valid values:

- 0= Channel not available
- 1= Measured channel
- 2= Mathematic channel

Description: Channel 1 to 4 (DT6200/KSS64xx) or 8 (DT65x0) or 6 (DT6536) is available at controller.

Parameter: int32_t SA_ChTransmit1..8 SA_ChTransmit1..8

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: Channel 1 to 8 is transmitted. Only for DT65x0

Parameter: int32_t SA_TrgMode SA_TrgMode

Direction: Up

Valid values:

- 0= Off
- 1= Rising edge
- 2= High level
- 3= Gate at rising edge

Description: Trigger active/disabled.

Parameter: int32_t SA_LinMode1..8 SA_LinMode1..8

Direction: Up

Valid values:

- 0= off
- 1= offset correction
- 2= 2 point linearization
- 3= 3 point linearization
- 4= 5 point linearization
- 5= 10 point linearization

Description: Linearisation mode for channel 1 to 8. Only for DT65x0

Parameter: int32_t SA_ShowChannels SA_ShowChannels

Direction: Up

Valid values:

0= none

1= all

2= selected

Description: Channels to be displayed. Only for DT65x0

Parameter: int32_t SA_ShowLinearized SA_ShowLinearized

Direction: Up

Valid values:

0= off

1= on

Description: Channels should be displayed linearized. Only for DT65x0

15.3.8.2 Get_Version (VER)

Retrieve the sensor software version.

Parameter: String SA_Version SA_Version

Direction: Up

Description: Software version of the controller.

15.3.8.3 Get_ChannelInfo (CHI)

Retrieve information about a sensor channel.

This command is not available for old controller DT6500.

Parameter: int32_t SP_Chан SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)

Description: Channels to get information for.

Parameter: String SA_ArticleNumber SA_ArticleNumber

Direction: Up

Valid values:

Numeric value

Description: Article number of the sensor channel.

Parameter: String SA_SensorName SA_SensorName

Direction: Up

Description: Name of the sensor channel.

Parameter: String SA_SerialNumber SA_SerialNumber

Direction: Up

Valid values:

Numeric value

Description: Serial number of the sensor channel.

Parameter: double SA_Offset	SA_Offset
Direction: Up	
Valid values:	
Minimum: -4294967040.0	
Maximum: 4294967040.0	
Unit: See parameter SA_Unit	
Description: Offset of the sensor channel.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: -4294967040.0	
Maximum: 4294967040.0	
Unit: See parameter SA_Unit	
Description: Range of sensor channel.	
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description: Unit of parameter SA_Offset and SA_Range, e.g. μm or mm.	

15.3.8.4 Get_ChannelInfos

Calls the sensor command Get_ChannelInfo for any requested channel.

Parameter: int32_t SP_Complete	SP_Complete
Direction: Down	
Valid values:	
0 = FALSE	
1 = TRUE	
Description: Specifies if any possible channel should be requested or only known channels (from former call to ChannelStatus or Get_Status).	
Parameter: String SA_ArticleNumber _{1..4} (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)	SA_ArticleNumber _{1..4} (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the sensor channel 1 to 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536).	
Parameter: String SA_SensorName _{1..4} (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)	SA_SensorName _{1..4} (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)
Direction: Up	
Description: Name of the sensor channel 1 to 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536).	
Parameter: String SA_SerialNumber _{1..4} (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)	SA_SerialNumber _{1..4} (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor channel 1 to 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536).	

Parameter: double SA_Offset1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)	SA_Offset1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)
Direction: Up	
Valid values:	
Minimum: -4294967040.0	
Maximum: 4294967040.0	
Unit: See parameter SA_Unit	
Description: Offset of the sensor channel 1 to 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536).	
Parameter: double SA_Range1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)	SA_Range1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)
Direction: Up	
Valid values:	
Minimum: -4294967040.0	
Maximum: 4294967040.0	
Unit: See parameter SA_Unit	
Description: Range of sensor channel 1 to 4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536).	
Parameter: String SA_Unit1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)	SA_Unit1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536)
Direction: Up	
Description: Unit of parameter SA_Offset and SA_Range, e.g. μm or mm.	

15.3.8.5 Get_ControllerInfo (COI)

Retrieve information about the controller.

This command is not available for old controller DT6500.

Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Description: Article number of the controller.	
Parameter: String SA_ControllerName	SA_ControllerName
Direction: Up	
Description: Name of the controller.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the controller.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the controller.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the controller.	

15.3.9 Parameter management

15.3.9.1 Save_Setup (SSU)

Save the current setup of controller to flash.

This command is only available for old controller DT6500.

15.3.9.2 Read_Setup (RSU)

Read the setup from controller flash.

This command is only available for old controller DT6500.

15.3.9.3 Factory_Defaults (FDE)

Restore the controller to factory defaults.

To save the default values call Save_Setup too. The new parameters are returned.

This command is only available for controller DT65x0.

Parameter: int32_t SA_SRIndex

SA_SRIndex

Direction: Up

Valid values:

- 0= 2.60 Hz
- 1= 5.21 Hz
- 2= 10.42 Hz
- 3= 15.63 Hz
- 4= 26.04 Hz
- 5= 31.25 Hz
- 6= 52.08 Hz
- 7= 62.50 Hz
- 8= 104.17 Hz
- 9= 520.83 Hz
- 10= 1041.67 Hz
- 11= 2083.33 Hz
- 12= 3906.25 Hz
- 13= 7812.50 Hz

Description: Samplerate index.

Parameter: int32_t SA_AvrType

SA_AvrType

Direction: Up

Valid values:

- 0= off
- 1= Moving average
- 2= Mean (arithmetic)
- 3= Median
- 4= Dynamic noise rejection

Description: Averaging type at controller.

Parameter: int32_t SA_AvrNbr

SA_AvrNbr

Direction: Up

Valid values:

- Minimum:** 2
- Maximum:** 8

Description: Averaging number at controller.

Parameter: int32_t SA_ChExist1..8 SA_ChExist1..8

Direction: Up

Valid values:

- 0= Channel not available
- 1= Measured channel
- 2= Mathematic channel

Description: Channel 1 to 8 is available at controller.

Parameter: int32_t SA_ChTransmit1..8 SA_ChTransmit1..8

Direction: Up

Valid values:

- 0= no
- 1= yes

Description: Channel 1 to 8 is transmitted.

Parameter: int32_t SA_TrgMode SA_TrgMode

Direction: Up

Valid values:

- 0= Off
- 1= Rising edge
- 2= High level
- 3= Gate at rising edge

Description: Trigger active/disabled.

Parameter: int32_t SA_LinMode1..8 SA_LinMode1..8

Direction: Up

Valid values:

- 0= off
- 1= offset correction
- 2= 2 point linearization
- 3= 3 point linearization
- 4= 5 point linearization
- 5= 10 point linearization

Description: Linearisation mode for channel 1 to 8.

Parameter: int32_t SA_ShowChannels SA_ShowChannels

Direction: Up

Valid values:

- 0= none
- 1= all
- 2= selected

Description: Channels to be displayed.

Parameter: int32_t SA_ShowLinearized SA_ShowLinearized

Direction: Up

Valid values:

- 0= off
- 1= on

Description: Channels should be displayed linearized.

15.3.10 Internal commands

15.3.10.1 Update_Firmware

Update firmware version at controller.

This command is not available for old controller DT6500.

Attention! This function can take approx. 1 minute. This is an internal command. It should not be used by the customer.

Parameter: Binary data SP_FirmwareFile

SP_FirmwareFile

Direction: Down

Valid values:

Firmware file as binary data.

Description: Firmware version, read from file.

Parameter: int32_t SA_Result

SA_Result

Direction: Up

Valid values:

0= Failed

1= Success

Description: Result of firmware update.

15.3.11 Special commands

15.3.11.1 Use_Defaults

This command tells the driver to use default values to operate with sensor data. If some parameters are not specified they are not changed. The sensor is not affected by this command.

This command is only available for controller DT65x0.

Parameter: double IP_Range1..8

IP_Range1..8

Direction: Down

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Unit: μm or mm

Description: Tells the driver the range of sensor for channel 1 to 8. It is used to scale the raw sensor values into μm or mm. If it is zero, no scaling is done.

15.3.11.2 Get_DrvSetting

Returns the current settings of the driver used for operating with sensor data. It is the opposite of Use_Defaults. The sensor is not affected by this command.

This command is only available for controller DT65x0.

Parameter: double IA_Range1..8

IA_Range1..8

Direction: Up

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Unit: μm or mm

Description: The range of sensor for channel 1 to 8, used by driver to scale values into μm or mm.

15.4 Commands for KSS6380

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [TCP/IP](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Status](#) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate and to assign values.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor, from 0 to 16777215.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Use_Defaults](#) parameter [IP_Range1..4](#)).

The values of selected channels are filled in the arrays one after another. Each array always starts with first selected channel.

15.4.1 Measurement

15.4.1.1 Set_SRIndex (SRA)

Set the samplerate for data acquisition.

Parameter: int32_t SP_SRIndex

SP_SRIndex

Direction: Down

Valid values:

- 0= 26 Hz
- 1= 104 Hz
- 2= 520 Hz
- 3= 1040 Hz

Description: Samplerate index

15.4.1.2 Get_SRIndex (SRA?)

Get the current samplerate.

Parameter: int32_t SA_SRIndex

SA_SRIndex

Direction: Up

Valid values:

- 0= 26 Hz
- 1= 104 Hz
- 2= 520 Hz
- 3= 1040 Hz

Description: Samplerate index.

15.4.1.3 Set_Trigger (TRG)

Activate/disable the trigger at controller.

Parameter: int32_t SP_TrMode SP_TrMode

Direction: Down

Valid values:

0= disabled

1= active

Description: Trigger active/disabled.

15.4.1.4 Get_Trigger (TRG?)

Retrieve the trigger mode at controller.

Parameter: int32_t SA_TrMode SA_TrMode

Direction: Up

Valid values:

0= disabled

1= active

Description: Trigger active/disabled.

15.4.1.5 Get_Measure (GMD)

Retrieve one value from controller, even if trigger is active.

15.4.1.6 Set_AvrType (AVT)

Set the averaging type at controller.

Parameter: int32_t SP_AvrType SP_AvrType

Direction: Down

Valid values:

0= off

1= Moving average

Description: Averaging type.

15.4.1.7 Get_AvrType (AVT?)

Retrieve the averaging type at controller.

Parameter: int32_t SA_AvrType SA_AvrType

Direction: Up

Valid values:

0= off

1= Moving average

Description: Averaging type at controller.

15.4.1.8 Set_AvrNbr (AVN)

Set the averaging number at controller.

Parameter: int32_t SP_AvrNbr SP_AvrNbr
Direction: Down
Valid values:
Minimum: 2
Maximum: 4
Description: Averaging number.

15.4.1.9 Get_AvrNbr (AVN?)

Retrieve the averaging number at controller.

Parameter: int32_t SA_AvrNbr SA_AvrNbr
Direction: Up
Valid values:
Minimum: 2
Maximum: 4
Description: Averaging number at controller.

15.4.2 Data output

15.4.2.1 ChannelStatus (CHS)

Retrieve the available channels at controller.

Parameter: int32_t SA_ChExist1..4 SA_ChExist1..4
Direction: Up
Valid values:
 0= Channel not available
 1= Measured channel
 2= Mathematic channel (only for channel 1 to 3)
Description: Channel 1 to 4 is available at controller.

15.4.2.2 Set_Channel (CHT)

Set the channels to transmit from controller.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_ChTransmit1..4 SP_ChTransmit1..4
Direction: Down
Valid values:
 0= no
 1= yes
Description: Channel 1 to 4 is transmitted.

15.4.2.3 Get_Channel (CHT?)

Get the channels transmitted from controller.

Parameter: int32_t SA_ChTransmit1..4

SA_ChTransmit1..4

Direction: Up

Valid values:

0 = no

1 = yes

Description: Channel 1 to 4 is transmitted.

15.4.3 Math

15.4.3.1 Set_MathFunction (SMF)

Set mathematic function at controller.

The result of the math function is transmitted like normal sensor data at a specific channel.

Because of the mathematic operations (multiplication, addition) the raw values can exceed the range of 24 bit. In this case, it is set to maximum value.

For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_Chан

SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 3

Description: Channels which returns the result. Only channel 1 to 3 supports math function.

Parameter: double SP_Offset

SP_Offset

Direction: Down

Valid values:

Minimum: -16777215 (-0xffffffff)

Maximum: 16777215 (0xffffffff)

Description: Offset to be added to result.

Parameter: double SP_FactorCapa

SP_FactorCapa

Direction: Down

Valid values:

Minimum: -9.9

Maximum: 9.9

Description: Multiplication factor for capa.

Parameter: double SP_FactorEddy

SP_FactorEddy

Direction: Down

Valid values:

Minimum: -9.9

Maximum: 9.9

Description: Multiplication factor for eddy.

15.4.3.2 Get_MathFunction (GMF)

Get mathematic function from controller.

Parameter: int32_t SP_Chан

SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 3

Description: Channels for which the function should be read. Only channel 1 to 3 supports math function.

Parameter: double SA_Offset

SA_Offset

Direction: Up

Valid values:

Minimum: -16777215 (-0xffffffff)

Maximum: 16777215 (0xffffffff)

Description: Offset to be added to result.

Parameter: double SA_FactorCapa

SA_FactorCapa

Direction: Up

Valid values:

Minimum: -9.9

Maximum: 9.9

Description: Multiplication factor for capa.

Parameter: double SA_FactorEddy

SA_FactorEddy

Direction: Up

Valid values:

Minimum: -9.9

Maximum: 9.9

Description: Multiplication factor for eddy.

15.4.3.3 Clr_MathFunction (CMF)

Clears mathematic function at controller.

Parameter: int32_t SP_Chан

SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 3

Description: Channels for which the function should be cleared. Only channel 1 to 3 supports math function.

15.4.4 Information

15.4.4.1 Get_Status (STS)

Retrieve detailed information about the controller.

Parameter: int32_t SA_SRIndex

SA_SRIndex

Direction: Up

Valid values:

0= 26 Hz

1= 104 Hz

2= 520 Hz

3= 1040 Hz

Description: Samplerate index.

Parameter: int32_t SA_AvrType SA_AvrType

Direction: Up

Valid values:

0= off

1= Moving average

Description: Averaging type at controller.

Parameter: int32_t SA_AvrNbr SA_AvrNbr

Direction: Up

Valid values:

Minimum: 2

Maximum: 4

Description: Averaging number at controller.

Parameter: int32_t SA_ChExist1..4 SA_ChExist1..4

Direction: Up

Valid values:

0= Channel not available

1= Measured channel

2= Mathematic channel (only for channel 1 to 3)

Description: Channel 1 to 4 is available at controller.

Parameter: int32_t SA_ChTransmit1..4 SA_ChTransmit1..4

Direction: Up

Valid values:

0= no

1= yes

Description: Channel 1 to 4 is transmitted.

Parameter: int32_t SA_TrMode SA_TrMode

Direction: Up

Valid values:

0= disabled

1= active

Description: Trigger active/disabled.

15.4.4.2 Get_Version (VER)

Retrieve the sensor software version.

Parameter: String SA_Version SA_Version

Direction: Up

Description: Software version of the controller.

15.4.5 Parameter management

15.4.5.1 Save_Setup (SSU)

Save the current setup of controller to flash.

15.4.5.2 Read_Setup (RSU)

Read the setup from controller flash.

15.4.5.3 Factory_Defaults (FDE)

Restore the controller to factory defaults.

To save the default values call Save_Setup too. The new parameters are returned.

Parameter: int32_t SA_SRIndex	SA_SRIndex
Direction: Up	
Valid values:	
0= 26 Hz	
1= 104 Hz	
2= 520 Hz	
3= 1040 Hz	
Description: Samplerate index.	
Parameter: int32_t SA_AvrType	SA_AvrType
Direction: Up	
Valid values:	
0= off	
1= Moving average	
Description: Averaging type at controller.	
Parameter: int32_t SA_AvrNbr	SA_AvrNbr
Direction: Up	
Valid values:	
Minimum: 2	
Maximum: 4	
Description: Averaging number at controller.	
Parameter: int32_t SA_ChExist1..4	SA_ChExist1..4
Direction: Up	
Valid values:	
0= Channel not available	
1= Measured channel	
2= Mathematic channel (only for channel 1 to 3)	
Description: Channel 1 to 4 is available at controller.	
Parameter: int32_t SA_ChTransmit1..4	SA_ChTransmit1..4
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Channel 1 to 4 is transmitted.	
Parameter: int32_t SA_TrMode	SA_TrMode
Direction: Up	
Valid values:	
0= disabled	
1= active	
Description: Trigger active/disabled.	

15.4.6 Internal commands

15.4.6.1 Set_TempCoeff (STC)

Set temperature coefficient for a channel at controller. This is an internal command.
It should not be used by the customer.

Parameter: int32_t SP_Chан

SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 2

Description: Channel to be set for.

Parameter: int32_t SP_TempCoeffParam

SP_TempCoeffParam

Direction: Down

Valid values:

Minimum: 0

Maximum: 16

Description: Temperature coefficient parameter.

Parameter: double SP_TemperatureCoeffizient

SP_TemperatureCoeffizient

Direction: Down

Valid values:

Minimum: 0

Maximum: 16777215 (0xfffffff)

Description: Temperature coefficient.

15.4.6.2 Get_TempCoeff (GTC)

Get temperature coefficient for a channel at controller. This is an internal command.
It should not be used by the customer.

Parameter: int32_t SP_Chан

SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 2

Description: Channel to be set for.

Parameter: int32_t SP_TempCoeffParam

SP_TempCoeffParam

Direction: Down

Valid values:

Minimum: 0

Maximum: 16

Description: Temperature coefficient parameter.

Parameter: double SA_TemperatureCoeffizient

SA_TemperatureCoeffizient

Direction: Up

Valid values:

Minimum: -16777215 (-0xfffffff)

Maximum: 16777215 (0xfffffff)

Description: Temperature coefficient.

15.4.6.3 Get_UnlinEddyVal (GUE)

Get unlinearized eddy value from controller. This is an internal command. It should not be used by the customer.

Parameter: double SA_UnlinearizedEddyValue SA_UnlinearizedEddyValue
Direction: Up
Valid values:
Minimum: -16777215 (-0xffffffff)
Maximum: 16777215 (0xffffffff)
Description: Unlinearized eddy value .

15.4.6.4 Set_LinPoint (SLP)

Set a linearisation point for a channel at controller. This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_Chан SP_Chан
Direction: Down
Valid values:
Minimum: 1
Maximum: 2
Description: Channel to be set for.

Parameter: int32_t SP_LinPos SP_LinPos
Direction: Down
Valid values:
 0= at 0%
 1= at 5%
 2= at 10%
 3= at 15%
 4= at 20%
 5= at 25%
 6= at 30%
 7= at 35%
 8= at 40%
 9= at 45%
 10= at 50%
 11= at 55%
 12= at 60%
 13= at 65%
 14= at 70%
 15= at 75%
 16= at 80%
 17= at 85%
 18= at 90%
 19= at 95%
 20= at 100%
Description: Linearisation position to be set for.

15.4.6.5 Get_LinPoint (GLP)

Get a linearisation point for a channel at controller. This is an internal command. It should not be used by the customer.

Parameter: int32_t SP_Chан SP_Chан

Direction: Down

Valid values:

Minimum: 1

Maximum: 2

Description: Channel to be get for.

Parameter: int32_t SP_LinPos SP_LinPos

Direction: Down

Valid values:

0= at 0%

1= at 5%

2= at 10%

3= at 15%

4= at 20%

5= at 25%

6= at 30%

7= at 35%

8= at 40%

9= at 45%

10= at 50%

11= at 55%

12= at 60%

13= at 65%

14= at 70%

15= at 75%

16= at 80%

17= at 85%

18= at 90%

19= at 95%

20= at 100%

Description: Linearisation position to be get for.

Parameter: double SA_LinPoint SA_LinPoint

Direction: Up

Valid values:

Minimum: 0

Maximum: 16777215 (0xffffffff)

Description: Linearisation point.

15.4.7 Special commands

15.4.7.1 Use_Defaults

This command tells the driver to use default values to operate with sensor data. If some parameters are not specified they are not changed. The sensor is not affected by this command.

Parameter: double IP_Range`1..4` IP_Range`1..4`

Direction: Down

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Unit: μm or mm

Description: Tells the driver the range of sensor for channel 1 to 4. It is used to scale the raw sensor values into μm or mm. If it is zero, no scaling is done.

15.4.7.2 Get_DrvSetting

Returns the current settings of the driver used for operating with sensor data. It is the opposite of Use_Defaults. The sensor is not affected by this command.

Parameter: double IA_Range`1..4` IA_Range`1..4`

Direction: Up

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Unit: μm or mm

Description: The range of sensor for channel 1 to 4, used by driver to scale values into μm or mm.

16 Commands for colorCONTROL sensors

16.1 Commands for ACS7000

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
- [IF2004](#) (native).
- [TCP/IP](#) (native).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).
- [IF2008_ETH](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_AllParameters](#) (SP_Additional= 1) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate, to interpret and scale data and to assign values. If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls optionally sensor command [Set_IPDataTransferMode](#) and [Set_DataOutInterface](#) at [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_AllParameters](#) (SP_Additional= 1)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

16.1.1 General commands

16.1.1.1 General

16.1.1.1.1 Get_Help (HELP)

Retrieve a help text from controller for a specific command.

Parameter: String SP_Command

SP_Command

Direction: Down

Valid values:

"" (empty string, means general help)
or any command name

Description: Name of the command.

16.1. Commands for ACS7000

Parameter: String SA_HelpText SA_HelpText
Direction: Up
Description: Help text to the command.

16.1.1.1.2 Get_Info (GETINFO)

Retrieve information about the controller.

Parameter: String SA_Sensor SA_Sensor
Direction: Up
Description: Name of the controller.

Parameter: String SA_SerialNumber SA_SerialNumber
Direction: Up
Valid values:
 Numeric value
Description: Serial number of the controller.

Parameter: String SA_Option SA_Option
Direction: Up
Valid values:
 Numeric value
Description: Option of the controller.

Parameter: String SA_ArticleNumber SA_ArticleNumber
Direction: Up
Valid values:
 Numeric value
Description: Article number of the controller.

Parameter: String SA_MacAddress SA_MacAddress
Direction: Up
Valid values:
 Valid MAC address in form of xx-xx-xx-xx-xx-xx
Description: MAC address (low level ethernet address) of the controller.

Parameter: String SA_Softwareversion SA_Softwareversion
Direction: Up
Description: Software version of firmware in the controller.

Parameter: String SA_Imagetype SA_Imagetype
Direction: Up
Description: Firmware image type used by the controller.

Parameter: String SA_Webstatic SA_Webstatic
Direction: Up
Valid values:
 Numeric value
Description: Version of webstatic section.

16.1. Commands for ACS7000

16.1.1.1.3 Set_Echo (ECHO)

Set echo for sensor commands.

Parameter: int32_t SP_Echo SP_Echo
Direction: Down
Valid values:
 0= Off
 1= On
Description: Echo mode.

16.1.1.1.4 Get_Echo (ECHO)

Get the echo mode.

Parameter: int32_t SA_Echo SA_Echo
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Off
 1= On
Description: Echo mode.

16.1.1.1.5 Get_AllParameters (PRINT)

Get all parameters from controller.

Parameter: int32_t SP_Additional SP_Additional
Direction: Down
Valid values:
 0= No
 1= Yes
Description: If set, additional information about controller and color table is output.

Parameter: int32_t SA_AveragingType SA_AveragingType
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= None
 1= Moving average (MOVING)
 2= Recursive averaging (RECURSIVE)
 3= Median
Description: Averaging type.

Parameter: int32_t SA_MovingCount SA_MovingCount

Direction: Up

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128
- 256
- 512
- 1024

Description: Number of value for the averaging window. This parameter is only available at moving average.

Parameter: int32_t SA_RecursiveCount SA_RecursiveCount

Direction: Up

Valid values:

- Minimum: 2
- Maximum: 32768

Description: Number of values for recursive averaging. This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount SA_MedianCount

Direction: Up

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only available at median.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate

Direction: Up

Valid values:

- 4000000
- 3500000
- 3000000
- 2500000
- 2000000
- 1500000
- 921600
- 691200
- 460800
- 230400
- 115200
- 9600

Unit: Baud

Description: Baudrate of controller.

Parameter: int32_t SA_DigitalOutBinFormat	SA_DigitalOutBinFormat
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = MSB	
1 = LSB	
Description: Pin assignment at digital outputs.	
Parameter: int32_t SA_DigitalOutColorFormat	SA_DigitalOutColorFormat
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = None	
1 = Binary	
2 = Channel	
3 = Lab-Check	
Description: Color format at digital outputs.	
Parameter: int32_t SA_ColorSpace	SA_ColorSpace
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = XYZ	
1 = LAB	
Description: Actually used colorspace	
Parameter: double SA_DeltaKL	SA_DeltaKL
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3.0	
Description: Weighting factor	
Parameter: double SA_DeltaKC	SA_DeltaKC
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3.0	
Description: Weighting factor	
Parameter: double SA_DeltaKH	SA_DeltaKH
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3.0	
Description: Weighting factor	
Parameter: int32_t SA_DeltaMode	SA_DeltaMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Euclidean distance (EUKLID)	
1 = Distance according to DIN99	
2 = Distance according to CIE94	
3 = Distance according to CMC	
4 = Distance according to CIEDE2000	
5 = Cylindrical distance model (CYLINDER)	
6 = Box distance model (BOX)	
Description: Delta mode	

16.1. Commands for ACS7000

Parameter: int32_t SA_DistanceMode	SA_DistanceMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Best hit (BESTHIT)	
1 = Selection	
Description: Distance mode.	
Parameter: int32_t SA_Echo	SA_Echo
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Off	
1 = On	
Description: Echo mode.	
Parameter: int32_t SA_EthernetMode	SA_EthernetMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Ethernet	
1 = Ethercat	
Description: Ethernet mode.	
Parameter: int32_t SA_UserLevel	SA_UserLevel
Direction: Up	
Valid values:	
-1 = Other user level (only for internal use)	
0 = User	
1 = Professional	
Description: Actual user level.	
Parameter: int32_t SA_DHCPEnabled	SA_DHCPEnabled
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = FALSE	
1 = TRUE	
Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).	
Parameter: String SA_Address	SA_Address
Direction: Up	
Valid values:	
Valid IP address in form of xxx.xxx.xxx.xxx	
Description: IP adress of the controller. If DHCP is enabled it returns the currently assigned IP address.	
Parameter: String SA_SubnetMask	SA_SubnetMask
Direction: Up	
Valid values:	
Valid network mask (e.g. 255.255.255.0 for a Class C network)	
Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.	

Parameter: String SA_Gateway	SA_Gateway
Direction: Up	
Valid values:	
Valid IP address of default gateway in form of xxx.xxx.xxx.xxx	
Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.	
Parameter: int32_t SA_Keylock	SA_Keylock
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Inactive	
1 = Active	
2 = Automatic (AUTO)	
Description: Keylock.	
Parameter: int32_t SA_KeylockTime	SA_KeylockTime
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 60	
Unit: Minutes	
Description: Keylock time (only available at automatic keylock).	
Parameter: int32_t SA_KeylockState	SA_KeylockState
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Inactive (IS_INACTIVE)	
1 = Active (IS_ACTIVE)	
Description: Actual keylock state (only available at automatic keylock).	
Parameter: int32_t SA_LEDControl	SA_LEDControl
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Max	
1 = Min	
2 = Manual	
3 = Automatic (AUTO)	
4 = Off	
Description: LED control	
Parameter: int32_t SA_LEDIntensityColdWhite	SA_LEDIntensityColdWhite
Direction: Up	
Valid values:	
Minimum: 50	
Maximum: 1023	
Description: LED intensity of cold white quadrant	
Parameter: int32_t SA_LEDIntensityGreen	SA_LEDIntensityGreen
Direction: Up	
Valid values:	
Minimum: 50	
Maximum: 1023	
Description: LED intensity of green quadrant	

16.1. Commands for ACS7000

Parameter: int32_t SA_LEDIntensityWarmWhite	SA_LEDIntensityWarmWhite
Direction: Up	
Valid values:	
Minimum: 50	
Maximum: 1023	
Description: LED intensity of warm white quadrant	
Parameter: int32_t SA_LEDIntensityViolet	SA_LEDIntensityViolet
Direction: Up	
Valid values:	
Minimum: 50	
Maximum: 1023	
Description: LED intensity of violet quadrant	
Parameter: int32_t SA_LightSource	SA_LightSource
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = D65	
1 = D50	
2 = D75	
3 = A	
4 = C	
5 = E	
6 = F4	
7 = F7	
8 = F11	
Description: Light source (illuminant)	
Parameter: int32_t SA_MeasureMode	SA_MeasureMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Color measurement (COLORMEASURE)	
1 = Color detection (COLORDETECTION)	
2 = Video (VIDEOSPECTRUM)	
Description: Measure mode.	
Parameter: double SA_Measrate	SA_Measrate
Direction: Up	
Valid values:	
Minimum: 20	
Maximum: 2000	
Unit: Hz	
Description: Samplerate of measurement.	
Parameter: int32_t SA_Protocol	SA_Protocol
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = TCP server (SERVER/TCP)	
1 = TCP client (CLIENT/TCP)	
2 = UDP sender (CLIENT/UDP)	
3 = None	
Description: Specifies if data should be send using TCP or UDP.	

Parameter: String SA_RemoteAddress	SA_RemoteAddress
Direction: Up	
Valid values:	
Valid IP address of receiver of data	
Description: Address of remote computer to send data to.	
Parameter: int32_t SA_Port	SA_Port
Direction: Up	
Valid values:	
Minimum: 1024	
Maximum: 65535	
Description: Port to send data to or to listen for incoming requests.	
Parameter: int32_t SA_Observer	SA_Observer
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Two degree (TWO_DEGREE)	
1 = Ten degree (TEN_DEGREE)	
Description: Viewing angle of observer.	
Parameter: int32_t SA_OutputVideoRaw_ETH	SA_OutputVideoRaw_ETH
Direction: Up	
Valid values:	
0 = no	
1 = yes	
Description: Specify if raw video signal is transmitted.	
Parameter: int32_t SA_OutputVideoDark_ETH	SA_OutputVideoDark_ETH
Direction: Up	
Valid values:	
0 = no	
1 = yes	
Description: Specify if dark corrected video signal is transmitted.	
Parameter: int32_t SA_OutputVideoLinearized_ETH	SA_OutputVideoLinearized_ETH
Direction: Up	
Valid values:	
0 = no	
1 = yes	
Description: Specify if linearized video signal is transmitted.	
Parameter: int32_t SA_OutputVideoLightSpectrum_ETH	SA_OutputVideoLightSpectrum_ETH
Direction: Up	
Valid values:	
0 = no	
1 = yes	
Description: Specify if light referenced signal is transmitted.	
Parameter: int32_t SA_OutputStatusFramerate_ETH	SA_OutputStatusFramerate_ETH
Direction: Up	
Valid values:	
0 = no	
1 = yes	
Description: Specify if framerate is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SA_OutputStatusShutter_ETH	SA_OutputStatusShutter_-ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputStatusTempDetector_ETH	SA_OutputStatusTempDetector_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature of detector is transmitted.	
Parameter: int32_t SA_OutputStatusTempLightSrc_ETH	SA_OutputStatusTempLight-Src_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature of light source is transmitted.	
Parameter: int32_t SA_OutputLightSensorRed_ETH	SA_OutputLightSensorRed_-ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if red part of light sensor is transmitted.	
Parameter: int32_t SA_OutputLightSensorGreen_ETH	SA_OutputLightSensor-Green_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if green part of light sensor is transmitted.	
Parameter: int32_t SA_OutputLightSensorBlue_ETH	SA_OutputLightSensor-Blue_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if blue part of light sensor is transmitted.	
Parameter: int32_t SA_OutputLightSensorBright_ETH	SA_OutputLightSensor-Bright_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if brightness of light sensor is transmitted.	
Parameter: int32_t SA_OutputStatusCounter_ETH	SA_OutputStatusCounter_-ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	

Parameter: int32_t SA_OutputStatusTimestamp_ETH	SA_OutputStatusTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputStatusError_ETH	SA_OutputStatusError_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if error flags are transmitted.	
Parameter: int32_t SA_OutputColorXYZ_ETH	SA_OutputColorXYZ_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in XYZ is transmitted.	
Parameter: int32_t SA_OutputColorRGB_ETH	SA_OutputColorRGB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in RGB is transmitted.	
Parameter: int32_t SA_OutputColorLAB_ETH	SA_OutputColorLAB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b* is transmitted.	
Parameter: int32_t SA_OutputColorLUV_ETH	SA_OutputColorLUV_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*u*v* is transmitted.	
Parameter: int32_t SA_OutputColorLCH_ETH	SA_OutputColorLCH_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h* is transmitted.	
Parameter: int32_t SA_OutputColorLAB99_ETH	SA_OutputColorLAB99_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b*99 is transmitted.	

Parameter: int32_t SA_OutputColorLCH99_ETH	SA_OutputColorLCH99_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h*99 is transmitted.	
Parameter: int32_t SA_OutputDistDistance01_ETH	SA_OutputDistDistance01_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 1 is transmitted.	
Parameter: int32_t SA_OutputDistDistance02_ETH	SA_OutputDistDistance02_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 2 is transmitted.	
Parameter: int32_t SA_OutputDistDistance03_ETH	SA_OutputDistDistance03_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 3 is transmitted.	
Parameter: int32_t SA_OutputDistDistance04_ETH	SA_OutputDistDistance04_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 4 is transmitted.	
Parameter: int32_t SA_OutputDistDistance05_ETH	SA_OutputDistDistance05_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 5 is transmitted.	
Parameter: int32_t SA_OutputDistDistance06_ETH	SA_OutputDistDistance06_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 6 is transmitted.	
Parameter: int32_t SA_OutputDistDistance07_ETH	SA_OutputDistDistance07_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 7 is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SA_OutputDistDistance08_ETH	SA_OutputDistDistance08_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 8 is transmitted.	
Parameter: int32_t SA_OutputDistDistance09_ETH	SA_OutputDistDistance09_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 9 is transmitted.	
Parameter: int32_t SA_OutputDistDistance10_ETH	SA_OutputDistDistance10_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 10 is transmitted.	
Parameter: int32_t SA_OutputDistDistance11_ETH	SA_OutputDistDistance11_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 11 is transmitted.	
Parameter: int32_t SA_OutputDistDistance12_ETH	SA_OutputDistDistance12_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 12 is transmitted.	
Parameter: int32_t SA_OutputDistDistance13_ETH	SA_OutputDistDistance13_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 13 is transmitted.	
Parameter: int32_t SA_OutputDistDistance14_ETH	SA_OutputDistDistance14_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 14 is transmitted.	
Parameter: int32_t SA_OutputDistDistance15_ETH	SA_OutputDistDistance15_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 15 is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SA_OutputDistDistance16_ETH	SA_OutputDistDistance16_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 16 is transmitted.	
Parameter: int32_t SA_OutputDistMinDistance_ETH	SA_OutputDistMinDis- tance_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if minimum color distance is transmitted.	
Parameter: int32_t SA_OutputDistDetectedID_ETH	SA_OutputDistDetectedID_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of detected color is transmitted.	
Parameter: int32_t SA_OutputDistMinDistID_ETH	SA_OutputDistMinDistID_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of nearest color is transmitted.	
Parameter: int32_t SA_OutputStatisticMin_ETH	SA_OutputStatisticMin_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic minimum value is transmitted.	
Parameter: int32_t SA_OutputStatisticMax_ETH	SA_OutputStatisticMax_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic maximum value is transmitted.	
Parameter: int32_t SA_OutputStatisticPeak2Peak_ETH	SA_OutputStatistic- Peak2Peak_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic peak to peak value is transmitted.	
Parameter: int32_t SA_OutputStatusFramerate_RS422	SA_OutputStatusFramer- ate_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if framerate is transmitted.	

Parameter: int32_t SA_OutputStatusShutter_RS422	SA_OutputStatusShutter_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SA_OutputStatusTempDetector_RS422	SA_OutputStatusTempDetec- tor_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature of detector is transmitted.	
Parameter: int32_t SA_OutputStatusTempLightSrc_RS422	SA_OutputStatusTempLight- Src_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature of light source is transmitted.	
Parameter: int32_t SA_OutputLightSensorRed_RS422	SA_OutputLightSensorRed_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if red part of light sensor is transmitted.	
Parameter: int32_t SA_OutputLightSensorGreen_RS422	SA_OutputLightSensor- Green_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if green part of light sensor is transmitted.	
Parameter: int32_t SA_OutputLightSensorBlue_RS422	SA_OutputLightSensor- Blue_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if blue part of light sensor is transmitted.	
Parameter: int32_t SA_OutputLightSensorBright_RS422	SA_OutputLightSensor- Bright_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if brightness of light sensor is transmitted.	
Parameter: int32_t SA_OutputStatusCounter_RS422	SA_OutputStatusCounter_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	

Parameter: int32_t SA_OutputStatusTimestamp_RS422	SA_OutputStatusTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputStatusError_RS422	SA_OutputStatusError_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if error flags are transmitted.	
Parameter: int32_t SA_OutputColorXYZ_RS422	SA_OutputColorXYZ_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in XYZ is transmitted.	
Parameter: int32_t SA_OutputColorRGB_RS422	SA_OutputColorRGB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in RGB is transmitted.	
Parameter: int32_t SA_OutputColorLAB_RS422	SA_OutputColorLAB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b* is transmitted.	
Parameter: int32_t SA_OutputColorLUV_RS422	SA_OutputColorLUV_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*u*v* is transmitted.	
Parameter: int32_t SA_OutputColorLCH_RS422	SA_OutputColorLCH_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h* is transmitted.	
Parameter: int32_t SA_OutputColorLAB99_RS422	SA_OutputColorLAB99_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b*99 is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SA_OutputColorLCH99_RS422	SA_OutputColorLCH99_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h*99 is transmitted.	
Parameter: int32_t SA_OutputDistMinDistance_RS422	SA_OutputDistMinDistance_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if minimum color distance is transmitted.	
Parameter: int32_t SA_OutputDistDetectedID_RS422	SA_OutputDistDetectedID_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of detected color is transmitted.	
Parameter: int32_t SA_OutputDistMinDistID_RS422	SA_OutputDistMinDistID_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of nearest color is transmitted.	
Parameter: int32_t SA_OutputStatisticMin_RS422	SA_OutputStatisticMin_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic minimum value is transmitted.	
Parameter: int32_t SA_OutputStatisticMax_RS422	SA_OutputStatisticMax_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic maximum value is transmitted.	
Parameter: int32_t SA_OutputStatisticPeak2Peak_RS422	SA_OutputStatisticPeak2Peak_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic peak to peak value is transmitted.	
Parameter: int32_t SA_DigitalOutDetectedID	SA_DigitalOutDetectedID
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of detected color is output.	

Parameter: int32_t SA_HoldLastValid	SA_HoldLastValid
Direction: Up	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	
Parameter: int32_t SA_DataOutInterface	SA_DataOutInterface
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= RS422	
2= Ethernet	
3= HTTP	
4= Ethercat	
Description: Active interface for data output.	
Parameter: int32_t SA_Resampling	SA_Resampling
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2000	
Description: Resampling value.	
Parameter: int32_t SA_ResampleRS422	SA_ResampleRS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: RS422 output is resampled.	
Parameter: int32_t SA_ResampleEthernet	SA_ResampleEthernet
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Output over ethernet is resampled.	
Parameter: int32_t SA_ShutterMode	SA_ShutterMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Automatic samplerate and shutter time (SEARCH)	
1= Fixed samplerate, automatic shutter time (MEAS)	
2= Fixed samplerate and shutter time (MANUAL)	
Description: Shutter mode.	
Parameter: int32_t SA_StatisticDepth	SA_StatisticDepth
Direction: Up	
Valid values:	
Minimum: 2	
Maximum: 2147483647 (INT32_MAX)	
Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 16384). Value greater as 16384 calculates statistic over all values.	

16.1. Commands for ACS7000

Parameter: int32_t SA_StatisticSignal SA_StatisticSignal

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Color XYZ (XYZ)
- 2= Color RGB (RGB)
- 3= Color L*a*b* (LAB)
- 4= Color L*u*v* (LUV)
- 5= Color L*c*h* (LCH)
- 6= Color L*a*b*99 (LAB99)
- 7= Color L*c*h*99 (LCH99)
- 8= Distance 1 (DIST01)
- 9= Distance 2 (DIST02)
- 10= Distance 3 (DIST03)
- 11= Distance 4 (DIST04)
- 12= Distance 5 (DIST05)
- 13= Distance 6 (DIST06)
- 14= Distance 7 (DIST07)
- 15= Distance 8 (DIST08)
- 16= Distance 9 (DIST09)
- 17= Distance 10 (DIST10)
- 18= Distance 11 (DIST11)
- 19= Distance 12 (DIST12)
- 20= Distance 13 (DIST13)
- 21= Distance 14 (DIST14)
- 22= Distance 15 (DIST15)
- 23= Distance 16 (DIST16)
- 24= Minimum color distance (MINDIST)
- 25= No. of detected color (DETECTID)
- 26= No. of nearest color (MINDISTID)

Description: Value which is used for statistic calculation.

Parameter: int32_t SA_DefaultUser SA_DefaultUser

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= User
- 1= Professional

Description: Default user level.

Parameter: int32_t SA_SyncMode SA_SyncMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Master
- 2= Slave

Description: Synchronization mode.

Parameter: int32_t SA_TriggerMode SA_TriggerMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor

0= None
 1= Edge
 2= Level (PULSE)
 3= Software

Description: Trigger mode.

Parameter: int32_t SA_TriggerLevel

SA_TriggerLevel

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
 0= High (Rising edge / High-active)
 1= Low (Falling edge / Low-active)

Description: Trigger level.

Parameter: int32_t SA_TriggerCount

SA_TriggerCount

Direction: Up

Valid values:

Minimum: 0
Maximum: 16383

Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

Parameter: int32_t SA_VideoAverage

SA_VideoAverage

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
 0= None
 1= Recursive over 2 lines (REC2)
 2= Recursive over 4 lines (REC4)
 3= Recursive over 8 lines (REC8)
 4= Recursive over 16 lines (REC16)
 5= Recursive over 32 lines (REC32)
 6= Recursive over 64 lines (REC64)
 7= Recursive over 128 lines (REC128)

Description: Averaging mode.

Parameter: String SA_ColorTable

SA_ColorTable

Direction: Up

Description: Whole table in one string, separated by new lines.

Parameter: int32_t SA_ColorTableCount

SA_ColorTableCount

Direction: Up

Valid values:

Minimum: 0
Maximum: 16

Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Pos1, SA_Pos2, ...

Parameter: int32_t SA_Pos1..x

SA_Pos1..x

Direction: Up

Valid values:

Minimum: 1
Maximum: 16

Description: Index of the color in the table.

Parameter: String SA_ColorName <code>1..x</code>	SA_ColorName <code>1..x</code>
Direction: Up	
Description: Name of the color in the table.	
Parameter: int32_t SA_Observer <code>1..x</code>	SA_Observer <code>1..x</code>
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Two degree (2 deg)	
1= Ten degree (10 deg)	
Description: Viewing angle of observer.	
Parameter: int32_t SA_LightSource <code>1..x</code>	SA_LightSource <code>1..x</code>
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= D65	
1= D50	
2= D75	
3= A	
4= C	
5= E	
6= F4	
7= F7	
8= F11	
Description: Light source (illuminant)	
Parameter: double SA_L* <code>1..x</code>	SA_L* <code>1..x</code>
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 255.0	
Description: L*, only availabe if colorspace is LAB	
Parameter: double SA_a* <code>1..x</code>	SA_a* <code>1..x</code>
Direction: Up	
Valid values:	
Minimum: -256.0	
Maximum: 255.0	
Description: a*, only availabe if colorspace is LAB	
Parameter: double SA_b* <code>1..x</code>	SA_b* <code>1..x</code>
Direction: Up	
Valid values:	
Minimum: -256.0	
Maximum: 255.0	
Description: b*, only availabe if colorspace is LAB	
Parameter: double SA_X <code>1..x</code>	SA_X <code>1..x</code>
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 255.0	
Description: X, only availabe if colorspace is XYZ	

Parameter: double SA_Y1..x	SA_Y1..x
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 255.0	
Description: Y, only available if colorspace is XYZ	
Parameter: double SA_Z1..x	SA_Z1..x
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 255.0	
Description: Z, only available if colorspace is XYZ	
Parameter: int32_t SA_Spectrum1..x	SA_Spectrum1..x
Direction: Up	
Valid values:	
0= not available	
1= available	
Description: Spectrum	
Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Description: Name of the controller.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the controller.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the controller.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the controller.	
Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description: MAC address (low level ethernet address) of the controller.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the controller.	
Parameter: String SA_ImageType	SA_ImageType
Direction: Up	
Description: Firmware image type used by the controller.	

16.1. Commands for ACS7000

Parameter: String SA_Webstatic SA_Webstatic
Direction: Up
Valid values:
 Numeric value
Description: Version of webstatic section.

16.1.1.6 Set_SyncMode (SYNC)

Set the synchronization mode.

Parameter: int32_t SP_SyncMode SP_SyncMode
Direction: Down
Valid values:
 0= None
 1= Master
 2= Slave
Description: Synchronization mode.

16.1.1.7 Get_SyncMode (SYNC)

Get the synchronization mode.

Parameter: int32_t SA_SyncMode SA_SyncMode
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= None
 1= Master
 2= Slave
Description: Synchronization mode.

16.1.1.8 Reset_Boot (RESET)

Resets the sensor.

At this command the controller may change output data after reboot. If first bit of `IP_AutomaticMode` is set (1), `Get_AllParameters` (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call `Get_TransmittedDataInfo` to retrieve new setting.

16.1.1.9 Set_Keylock (KEYLOCK)

Set key lock for sensor.

Parameter: int32_t SP_Keylock SP_Keylock
Direction: Down
Valid values:
 0= Inactive
 1= Active
 2= Automatic (AUTO)
Description: Keylock.

Parameter: int32_t SP_KeylockTime SP_KeylockTime
Direction: Down
Valid values:
Minimum: 0
Maximum: 60
Unit: Minutes
Description: Keylock time (only used at automatic keylock).

16.1.1.1.10 Get_Keylock (KEYLOCK)

Get key lock for sensor.

Parameter: int32_t SA_Keylock SA_Keylock
Direction: Up
Valid values:
-1 = Unknown parameter value from sensor
0 = Inactive
1 = Active
2 = Automatic (AUTO)
Description: Keylock.

Parameter: int32_t SA_KeylockTime SA_KeylockTime
Direction: Up
Valid values:
Minimum: 0
Maximum: 60
Unit: Minutes
Description: Keylock time (only available at automatic keylock).

Parameter: int32_t SA_KeylockState SA_KeylockState
Direction: Up
Valid values:
-1 = Unknown parameter value from sensor
0 = Inactive (IS_INACTIVE)
1 = Active (IS_ACTIVE)
Description: Actual keylock state (only available at automatic keylock).

16.1.1.2 User Level

16.1.1.2.1 Logout (LOGOUT)

Change user level to user.

16.1.1.2.2 Login (LOGIN)

Change user level to professional.

Parameter: String SP_Password SP_Password
Direction: Down
Description: Valid password to login.

16.1.1.2.3 Get_UserLevel (GETUSERLEVEL)

Retrieve actual user level.

Parameter: int32_t SA_UserLevel SA_UserLevel
Direction: Up
Valid values:
 -1 = Other user level (only for internal use)
 0 = User
 1 = Professional
Description: Actual user level.

16.1.1.2.4 Set_DefaultUser (STDUSER)

Set the default user level after booting the system.

Parameter: int32_t SP_DefaultUser SP_DefaultUser
Direction: Down
Valid values:
 0 = User
 1 = Professional
Description: Default user level.

16.1.1.2.5 Get_DefaultUser (STDUSER)

Get the default user level after booting the system.

Parameter: int32_t SA_DefaultUser SA_DefaultUser
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = User
 1 = Professional
Description: Default user level.

16.1.1.2.6 Set_Password (PASSWD)

Change the password for login.

Parameter: String SP_OldPassword SP_OldPassword
Direction: Down
Description: Old password.

Parameter: String SP_NewPassword SP_NewPassword
Direction: Down
Description: New password.

16.1.1.3 Sensor

16.1.1.3.1 Set_Observer (OBSERVER)

Set the viewing angle.

Parameter: int32_t SP_Observer

SP_Observer

Direction: Down

Valid values:

0= Two degree (TWO_DEGREE)

1= Ten degree (TEN_DEGREE)

Description: Viewing angle of observer.

16.1.1.3.2 Get_Observer (OBSERVER)

Get the viewing angle.

Parameter: int32_t SA_Observer

SA_Observer

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Two degree (TWO_DEGREE)

1= Ten degree (TEN_DEGREE)

Description: Viewing angle of observer.

16.1.1.3.3 Set_LightSource (LQSRC)

Set the light source.

Parameter: int32_t SP_LightSource

SP_LightSource

Direction: Down

Valid values:

0= D65

1= D50

2= D75

3= A

4= C

5= E

6= F4

7= F7

8= F11

Description: Light source (illuminant)

16.1.1.3.4 Get_LightSource (LQSRC)

Get the light source.

Parameter: int32_t SA_LightSource SA_LightSource

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= D65
- 1= D50
- 2= D75
- 3= A
- 4= C
- 5= E
- 6= F4
- 7= F7
- 8= F11

Description: Light source (illuminant)

16.1.1.3.5 Set_LEDControl (LEDCTRL)

Control of the illumination LED.

Parameter: int32_t SP_LEDControl SP_LEDControl

Direction: Down

Valid values:

- 0= Max
- 1= Min
- 2= Manual
- 3= Automatic (AUTO)
- 4= Off

Description: LED control

16.1.1.3.6 Get_LEDControl (LEDCTRL)

Control of the illumination LED.

Parameter: int32_t SA_LEDControl SA_LEDControl

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Max
- 1= Min
- 2= Manual
- 3= Automatic (AUTO)
- 4= Off

Description: LED control

16.1.1.3.7 Set_LEDIntensityColdWhite (LEDKW)

Set LED intensity of cold white quadrant.

Parameter: int32_t SP_LEDIntensityColdWhite

SP_LEDIntensityColdWhite

Direction: Down

Valid values:

Minimum: 50

Maximum: 1023

Description: LED intensity of cold white quadrant

16.1.1.3.8 Get_LEDIntensityColdWhite (LEDKW)

Get LED intensity of cold white quadrant.

Parameter: int32_t SA_LEDIntensityColdWhite

SA_LEDIntensityColdWhite

Direction: Up

Valid values:

Minimum: 50

Maximum: 1023

Description: LED intensity of cold white quadrant

16.1.1.3.9 Set_LEDIntensityGreen (LEDGR)

Set LED intensity of green quadrant.

Parameter: int32_t SP_LEDIntensityGreen

SP_LEDIntensityGreen

Direction: Down

Valid values:

Minimum: 50

Maximum: 1023

Description: LED intensity of green quadrant

16.1.1.3.10 Get_LEDIntensityGreen (LEDGR)

Get LED intensity of green quadrant.

Parameter: int32_t SA_LEDIntensityGreen

SA_LEDIntensityGreen

Direction: Up

Valid values:

Minimum: 50

Maximum: 1023

Description: LED intensity of green quadrant

16.1.1.3.11 Set_LEDIntensityWarmWhite (LEDWW)

Set LED intensity of warm white quadrant.

Parameter: int32_t SP_LEDIntensityWarmWhite

SP_LEDIntensityWarmWhite

Direction: Down

Valid values:

Minimum: 50

Maximum: 1023

Description: LED intensity of warm white quadrant

16.1.1.3.12 Get_LEDIntensityWarmWhite (LEDWW)

Get LED intensity of warm white quadrant.

Parameter: int32_t SA_LEDIntensityWarmWhite

SA_LEDIntensityWarmWhite

Direction: Up

Valid values:

Minimum: 50

Maximum: 1023

Description: LED intensity of warm white quadrant

16.1.1.3.13 Set_LEDIntensityViolet (LEDUV)

Set LED intensity of violet quadrant.

Parameter: int32_t SP_LEDIntensityViolet

SP_LEDIntensityViolet

Direction: Down

Valid values:

Minimum: 50

Maximum: 1023

Description: LED intensity of violet quadrant

16.1.1.3.14 Get_LEDIntensityViolet (LEDUV)

Get LED intensity of violet quadrant.

Parameter: int32_t SA_LEDIntensityViolet

SA_LEDIntensityViolet

Direction: Up

Valid values:

Minimum: 50

Maximum: 1023

Description: LED intensity of violet quadrant

16.1.1.3.15 AutoLEDAdjustment (AUTOLEDADJ)

Brightness of the illumination LED is adjusted once, using the optimum range for the selected manual measuring frequency.

16.1.1.3.16 DarkCorr (DARKCORR)

Make a dark correction.

16.1.1.3.17 LightCorr (LIGHTCORR)

Make a light correction.

16.1.1.4 Triggering

16.1.1.4.1 Set_TriggerMode (TRIGGER)

Set the trigger mode.

Parameter: int32_t SP_TriggerMode

SP_TriggerMode

Direction: Down

Valid values:

- 0= None
- 1= Edge
- 2= Level (PULSE)
- 3= Software

Description: Trigger mode.

16.1.1.4.2 Get_TriggerMode (TRIGGER)

Get the active trigger mode.

Parameter: int32_t SA_TriggerMode

SA_TriggerMode

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Edge
- 2= Level (PULSE)
- 3= Software

Description: Trigger mode.

16.1.1.4.3 Set_TriggerLevel (TRIGGERLEVEL)

Set the trigger level.

Parameter: int32_t SP_TriggerLevel

SP_TriggerLevel

Direction: Down

Valid values:

- 0= High (Rising edge / High-active)
- 1= Low (Falling edge / Low-active)

Description: Trigger level.

16.1.1.4.4 Get_TriggerLevel (TRIGGERLEVEL)

Get the active trigger level.

Parameter: int32_t SA_TriggerLevel

SA_TriggerLevel

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= High (Rising edge / High-active)
- 1= Low (Falling edge / Low-active)

Description: Trigger level.

16.1.1.4.5 Set_TriggerCount (TRIGGERCOUNT)

Set the number of values to measure at trigger.

Parameter: int32_t SP_TriggerCount SP_TriggerCount
Direction: Down
Valid values:
Minimum: 0
Maximum: 16383
Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

16.1.1.4.6 Get_TriggerCount (TRIGGERCOUNT)

Get the number of values to measure at trigger.

Parameter: int32_t SA_TriggerCount SA_TriggerCount
Direction: Up
Valid values:
Minimum: 0
Maximum: 16383
Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

16.1.1.4.7 Software_Trigger (TRIGGERSW)

Execute a software trigger.

16.1.1.5 Interfaces

16.1.1.5.1 Set_IPConfiguration (IPCONFIG)

Set the IP configuration at controller.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_DHCPEnabled SP_DHCPEnabled
Direction: Down
Valid values:
 0= FALSE
 1= TRUE
Description: Specify if controller should use a static IP address or ask for IP at DHCP server (dynamic IP address).

Parameter: String SP_Address SP_Address
Direction: Down
Valid values:
 Valid IP address in form of xxx.xxx.xxx.xxx
Description: IP address of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_SubnetMask SP_SubnetMask

Direction: Down

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_Gateway SP_Gateway

Direction: Down

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: The default gateway must be specified if the controller should communicate with peers in foreign subnets. This parameter is only evaluated on static IP assignment.

16.1.1.5.2 Get_IPConfiguration (IPCONFIG)

Get the IP configuration at controller.

Parameter: int32_t SA_DHCPEnabled SA_DHCPEnabled

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor

0 = FALSE

1 = TRUE

Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).

Parameter: String SA_Address SA_Address

Direction: Up

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.

Parameter: String SA_SubnetMask SA_SubnetMask

Direction: Up

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.

Parameter: String SA_Gateway SA_Gateway

Direction: Up

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

16.1.1.5.3 Set_IPDataTransferMode (MEATRANSFER)

Set IP protocol at controller.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_Protocol

SP_Protocol

Direction: Down

Valid values:

- 0= TCP server (SERVER/TCP)
- 1= TCP client (CLIENT/TCP)
- 2= UDP sender (CLIENT/UDP)
- 3= None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SP_RemoteAddress

SP_RemoteAddress

Direction: Down

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to. On TCP server this parameter is ignored.

Parameter: int32_t SP_Port

SP_Port

Direction: Down

Valid values:

- Minimum: 1024
- Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

16.1.1.5.4 Get_IPDataTransferMode (MEATRANSFER)

Get IP protocol at controller.

Parameter: int32_t SA_Protocol

SA_Protocol

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= TCP server (SERVER/TCP)
- 1= TCP client (CLIENT/TCP)
- 2= UDP sender (CLIENT/UDP)
- 3= None

Description: Specifies if data should be send using TCP or UDP.

Parameter: String SA_RemoteAddress

SA_RemoteAddress

Direction: Up

Valid values:

Valid IP address of receiver of data

Description: Address of remote computer to send data to.

Parameter: int32_t SA_Port

SA_Port

Direction: Up

Valid values:

- Minimum: 1024
- Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

16.1.1.5.5 Set_EthernetMode (ETHERMODE)

Switches ethernet mode between Ethernet and Ethercat.

Parameter: int32_t SP_EthernetMode

SP_EthernetMode

Direction: Down

Valid values:

0= Ethernet

1= Ethercat

Description: Ethernet mode.

16.1.1.5.6 Get_EthernetMode (ETHERMODE)

Get ethernet mode of controller.

Parameter: int32_t SA_EthernetMode

SA_EthernetMode

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Ethernet

1= Ethercat

Description: Ethernet mode.

16.1.1.5.7 Set_Baudrate (BAUDRATE)

Set baudrate of controller for serial RS422 communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

4000000

3500000

3000000

2500000

2000000

1500000

921600

691200

460800

230400

115200

9600

Unit: Baud

Description: Baudrate of controller.

Parameter: int32_t CP_InterruptDataTransfer

CP_InterruptDataTransfer

Direction: Down

Valid values:

0= false

1= true

Default: 0

Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Set_DataOutInterface](#)) and enabling it again at end (only if enabled before).

16.1.1.5.8 Get_Baudrate (BAUDRATE)

Get baudrate of controller for serial RS422 communication.

Parameter: int32_t SA_SensorBaudrate

SA_SensorBaudrate

Direction: Up

Valid values:

- 4000000
- 3500000
- 3000000
- 2500000
- 2000000
- 1500000
- 921600
- 691200
- 460800
- 230400
- 115200
- 9600

Unit: Baud

Description: Baudrate of controller.

16.1.1.5.9 Set_DigitalOutColorFormat (COLOROUT_FORMAT)

Set the color format at digital outputs.

Parameter: int32_t SP_DigitalOutColorFormat

SP_DigitalOutColorFormat

Direction: Down

Valid values:

- 0= None
- 1= Binary
- 2= Channel
- 3= Lab-Check

Description: Color format at digital outputs.

16.1.1.5.10 Get_DigitalOutColorFormat (COLOROUT_FORMAT)

Get the color format at digital outputs.

Parameter: int32_t SA_DigitalOutColorFormat

SA_DigitalOutColorFormat

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Binary
- 2= Channel
- 3= Lab-Check

Description: Color format at digital outputs.

16.1.1.5.11 Set_DigitalOutBinFormat (BIN_FORMAT)

Set the pin assignment at digital outputs for Set_DigitalOutColorFormat (Binary).

Parameter: int32_t SP_DigitalOutBinFormat

SP_DigitalOutBinFormat

Direction: Down

Valid values:

0= MSB
1= LSB

Description: Pin assignent at digital outputs.

16.1.1.5.12 Get_DigitalOutBinFormat (BIN_FORMAT)

Get the pin assignment at digital outputs for Set_DigitalOutColorFormat (Binary).

Parameter: int32_t SA_DigitalOutBinFormat

SA_DigitalOutBinFormat

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= MSB
1= LSB

Description: Pin assignent at digital outputs.

16.1.1.6 Parameter management

16.1.1.6.1 Save_Parameters (STORE)

Save actual parameters at controller. There can be saved several settings on different locations. So it is easy to switch to another setting.

Parameter: int32_t SP_ParameterSet

SP_ParameterSet

Direction: Down

Valid values:

Minimum: 1
Maximum: 8

Description: Location to save the settings.

16.1.1.6.2 Load_Parameters (READ)

Load stored parameters into controller RAM.

There can be loaded several settings from different locations. So it is easy to switch to another setting.

At this command the controller may change output data after applying new setting. If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_ParameterType

SP_ParameterType

Direction: Down

Valid values:

0= All settings (ALL)
1= Device settings (DEVICE)
2= Measurement settings (MEAS)

Description: Specifies which settings should be loaded.

Parameter: int32_t SP_ParameterSet SP_ParameterSet
Direction: Down
Valid values:
 Minimum: 1
 Maximum: 8
Description: Location from where the settings should be loaded.

16.1.1.6.3 Set_Default (SETDEFAULT)

Reset the controller to default settings.
 At this command the sensor may change output data after applying default settings.
 If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually.
 After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DefaultType SP_DefaultType
Direction: Down
Valid values:
 0= All settings (ALL)
 1= Just the current setting (CURRENT)
 2= Only color table (COLOR)
Description: Specifies which settings should be reset.

Parameter: int32_t SP_KeepDevice SP_KeepDevice
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specifies if device settings should be kept temporary.

Parameter: int32_t SP_SaveTemporary SP_SaveTemporary
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specifies if temporary settings should be stored in a setup.

16.1.2 Measurement

16.1.2.1 General

16.1.2.1.1 Set_MeasureMode (MEASMODE)

Set the measure mode.
 At this command the controller may change output data automatically. If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 0) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_MeasureMode SP_MeasureMode
Direction: Down
Valid values:
 0= Color measurement (COLORMEASURE)
 1= Color detection (COLORDETECTION)
 2= Video (VIDEOSPECTRUM)
Description: Measure mode.

16.1.2.1.2 Get_MeasureMode (MEASMODE)

Get the measure mode.

Parameter: int32_t SA_MeasureMode SA_MeasureMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Color measurement (COLORMEASURE)
- 1 = Color detection (COLORDETECTION)
- 2 = Video (VIDEOSPECTRUM)

Description: Measure mode.

16.1.2.1.3 Set_ShutterMode (SHUTTERMODE)

Set the shutter mode.

Parameter: int32_t SP_ShutterMode SP_ShutterMode

Direction: Down

Valid values:

- 0 = Automatic samplerate and shutter time (SEARCH)
- 1 = Fixed samplerate, automatic shutter time (MEAS)
- 2 = Fixed samplerate and shutter time (MANUAL)

Description: Shutter mode.

16.1.2.1.4 Get_ShutterMode (SHUTTERMODE)

Get the shutter mode.

Parameter: int32_t SA_ShutterMode SA_ShutterMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Automatic samplerate and shutter time (SEARCH)
- 1 = Fixed samplerate, automatic shutter time (MEAS)
- 2 = Fixed samplerate and shutter time (MANUAL)

Description: Shutter mode.

16.1.2.1.5 Get_Video (GETVIDEO)

Get recent video signals from sensor.

Parameter: Binary data SA_VideoRaw SA_VideoRaw

Direction: Up

Valid values:

- 256 words (each 2 byte), each word is an intensity value.

Description: Raw video signal

16.1. Commands for ACS7000

Parameter: Binary data SA_VideoDark SA_VideoDark
Direction: Up
Valid values:
 256 words (each 2 byte), each word is an dark corrected value.
Description: Dark corrected video signal

Parameter: Binary data SA_VideoLin SA_VideoLin
Direction: Up
Valid values:
 256 words (each 2 byte), each word is an linearized value.
Description: Linearized video signal

Parameter: Binary data SA_VideoLight SA_VideoLight
Direction: Up
Valid values:
 256 words (each 2 byte), each word is an light referenced value.
Description: Light referenced video signal

Parameter: double SA_VideoTimestamp SA_VideoTimestamp
Direction: Up
Valid values:
Minimum: 0
Maximum: 1.79769e+308 (DBL_MAX)
Unit: ms
Description: Timestamp of the video signal. It starts from 1970 Jan 01 at 01:00. It is generated when the video has arrived at TCP/IP socket.

Example how to read a video signal from sensor:

```

/* Do not forget to handle potential error after each call to MEDAQLib! */
/* Create sensor instance, open sensor via TCP/IP, set output to ethernet */
/* and then switch to video mode: */
err= SetIntExecSCmd (instance, "Set_MeasureMode", "SP_MeasureMode", 3 /*Video*/);

/* Select the desired video signal: */
err= SetParameterInt (instance, "SP_OutputVideoRaw_ETH", 1);
err= SetParameterInt (instance, "SP_OutputVideoDark_ETH", 1);
err= ExecSCmd (instance, "Set_OutputVideo_ETH");

/* Acquire video signals: */
err= ExecSCmd (instance, "Get_Video");
uint16_t videoRaw[256], videoDark[256];
uint32_t maxlen= sizeof (videoRaw);
err= GetParameterBinary (instance, "SA_VideoRaw", (uint8_t *)videoRaw, &maxLen);
assert (maxlen==sizeof (videoRaw)); // additional validity check
maxLen= sizeof (videoDark);
err= GetParameterBinary (instance, "SA_VideoDark", (uint8_t *)videoDark, &maxLen);
assert (maxlen==sizeof (videoDark)); // additional validity check

/* Do anything with the received video signals */

```

16.1.2.1.6 Set_Samplerate (MEASRATE)

Set the samplerate.

Parameter: double SP_Measrate SP_Measrate
Direction: Down
Valid values:
Minimum: 20
Maximum: 2000
Unit: Hz
Description: Samplerate of measurement.

16.1.2.1.7 Get_Samplerate (MEASRATE)

Get the samplerate.

Parameter: double SA_Measrate

SA_Measrate

Direction: Up

Valid values:

Minimum: 20

Maximum: 2000

Unit: Hz

Description: Samplerate of measurement.

16.1.2.1.8 Set_DeltaMode (DELTAMODE)

Set the method of color difference calculation.

Parameter: int32_t SP_DeltaMode

SP_DeltaMode

Direction: Down

Valid values:

0= Euclidean distance (EUKLID)

1= Distance according to DIN99

2= Distance according to CIE94

3= Distance according to CMC

4= Distance according to CIEDE2000

5= Cylindrical distance model (CYLINDER)

6= Box distance model (BOX)

Description: Delta mode

16.1.2.1.9 Get_DeltaMode (DELTAMODE)

Get the method of color difference calculation.

Parameter: int32_t SA_DeltaMode

SA_DeltaMode

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Euclidean distance (EUKLID)

1= Distance according to DIN99

2= Distance according to CIE94

3= Distance according to CMC

4= Distance according to CIEDE2000

5= Cylindrical distance model (CYLINDER)

6= Box distance model (BOX)

Description: Delta mode

16.1.2.1.10 Set_DeltaKL (DELTA_KL)

Set weighting factor for delta mode CIE94, CMC and CIEDE2000.

Parameter: double SP_DeltaKL

SP_DeltaKL

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 3.0

Description: Weighting factor

16.1.2.1.11 Get_DeltaKL (DELTA_KL)

Get weighting factor for delta mode CIE94, CMC and CIEDE2000.

Parameter: double SA_DeltaKL SA_DeltaKL
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 3.0
Description: Weighting factor

16.1.2.1.12 Set_DeltaKC (DELTA_KC)

Set weighting factor for delta mode CIE94, CMC and CIEDE2000.

Parameter: double SP_DeltaKC SP_DeltaKC
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 3.0
Description: Weighting factor

16.1.2.1.13 Get_DeltaKC (DELTA_KC)

Get weighting factor for delta mode CIE94, CMC and CIEDE2000.

Parameter: double SA_DeltaKC SA_DeltaKC
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 3.0
Description: Weighting factor

16.1.2.1.14 Set_DeltaKH (DELTA_KH)

Set weighting factor for delta mode CIE94, CMC and CIEDE2000.

Parameter: double SP_DeltaKH SP_DeltaKH
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 3.0
Description: Weighting factor

16.1.2.1.15 Get_DeltaKH (DELTA_KH)

Get weighting factor for delta mode CIE94, CMC and CIEDE2000.

Parameter: double SA_DeltaKH SA_DeltaKH
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 3.0
Description: Weighting factor

16.1.2.2 Color database

16.1.2.2.1 Get_ColorTable (COLORTABLE)

Get a list of all colors in database.

Parameter: int32_t SA_ColorSpace

SA_ColorSpace

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= XYZ
- 1= LAB

Description: Actually used colorspace

Parameter: String SA_ColorTable

SA_ColorTable

Direction: Up

Description: Whole table in one string, separated by new lines.

Parameter: int32_t SA_ColorTableCount

SA_ColorTableCount

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 16

Description: Number of entries in the table. All following parameters exists from 1 to this number, e.g. SA_Pos1, SA_Pos2, ...

Parameter: int32_t SA_Pos1..x

SA_Pos1..x

Direction: Up

Valid values:

- Minimum:** 1
- Maximum:** 16

Description: Index of the color in the table.

Parameter: String SA_ColourName1..x

SA_ColourName1..x

Direction: Up

Description: Name of the color in the table.

Parameter: int32_t SA_Observer1..x

SA_Observer1..x

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Two degree (2 deg)
- 1= Ten degree (10 deg)

Description: Viewing angle of observer.

Parameter: int32_t SA_LightSource1..x

SA_LightSource1..x

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= D65
- 1= D50
- 2= D75
- 3= A
- 4= C
- 5= E
- 6= F4
- 7= F7
- 8= F11

Description: Light source (illuminant)

Parameter: double SA_L*1..x	SA_L*1..x
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 255.0	
Description: L*, only available if colorspace is LAB	
Parameter: double SA_a*1..x	SA_a*1..x
Direction: Up	
Valid values:	
Minimum: -256.0	
Maximum: 255.0	
Description: a*, only available if colorspace is LAB	
Parameter: double SA_b*1..x	SA_b*1..x
Direction: Up	
Valid values:	
Minimum: -256.0	
Maximum: 255.0	
Description: b*, only available if colorspace is LAB	
Parameter: double SA_X1..x	SA_X1..x
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 255.0	
Description: X, only available if colorspace is XYZ	
Parameter: double SA_Y1..x	SA_Y1..x
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 255.0	
Description: Y, only available if colorspace is XYZ	
Parameter: double SA_Z1..x	SA_Z1..x
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 255.0	
Description: Z, only available if colorspace is XYZ	
Parameter: int32_t SA_Spectrum1..x	SA_Spectrum1..x
Direction: Up	
Valid values:	
0 = not available	
1 = available	
Description: Spectrum	

16.1.2.2.2 Edit_Color (COLORNEW)

Add a new color or edit an existing color.

Parameter: int32_t SP_Pos

SP_Pos

Direction: Down

Valid values:

Minimum: 1

Maximum: 16

Description: Index of the color to add or edit.

Parameter: String SP_ColourName

SP_ColourName

Direction: Down

Description: Name of the color.

Parameter: int32_t SP_UsedColorSpace

SP_UsedColorSpace

Direction: Down

Valid values:

0= XYZ

1= LAB

2= Spectrum

Description: Color space to use

Parameter: int32_t SP_Observer

SP_Observer

Direction: Down

Valid values:

0= Two degree (2)

1= Ten degree (10)

Description: Viewing angle of observer (not used at color space spectrum).

Parameter: int32_t SP_LightSource

SP_LightSource

Direction: Down

Valid values:

0= D65

1= D50

2= D75

3= A

4= C

5= E

6= F4

7= F7

8= F11

Description: Light source (illuminant) (not used at color space spectrum).

Parameter: double SP_L*

SP_L*

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 255.0

Description: L*, if colorspace is LAB

Parameter: double SP_a*	SP_a*
Direction: Down	
Valid values:	
Minimum: -256.0	
Maximum: 255.0	
Description: a*, if colorspace is LAB	
Parameter: double SP_b*	SP_b*
Direction: Down	
Valid values:	
Minimum: -256.0	
Maximum: 255.0	
Description: b*, if colorspace is LAB	
Parameter: double SP_X	SP_X
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 255.0	
Description: X, if colorspace is XYZ	
Parameter: double SP_Y	SP_Y
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 255.0	
Description: Y, if colorspace is XYZ	
Parameter: double SP_Z	SP_Z
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 255.0	
Description: Z, if colorspace is XYZ	

16.1.2.2.3 Set_ColorDescription (COLORDESCR)

Set description of a color in color table.

Parameter: String SP_ColorName	SP_ColorName
Direction: Down	
Description: Name of the color.	
Parameter: String SP_Description	SP_Description
Direction: Down	
Description: Description of the color.	

16.1.2.2.4 Get_ColorDescription (COLORDESCR)

Get description of a color in color table.

Parameter: String SP_ColorName	SP_ColorName
Direction: Down	
Description: Name of the color.	

16.1. Commands for ACS7000

Parameter: String SA_Description SA_Description
Direction: Up
Description: Description of the color.

16.1.2.2.5 Set_ColorThresholds (THRESHOLDS)

Set thresholds for a color.
For this command an [Update ...](#) meta command is available.

Parameter: String SP_ColorName SP_ColorName
Direction: Down
Description: Name of the color.

Parameter: double SP_Delta_E_L SP_Delta_E_L
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 64.0
Description: Delta value for L*.

Parameter: double SP_Delta_A_AB SP_Delta_A_AB
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 64.0
Description: Delta value for a*b* resp. a*.

Parameter: double SP_Delta_B SP_Delta_B
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 64.0
Description: Delta value for b*.

16.1.2.2.6 Get_ColorThresholds (THRESHOLDS)

Get thresholds for a color.

Parameter: String SP_ColorName SP_ColorName
Direction: Down
Description: Name of the color.

Parameter: double SA_Delta_E_L SA_Delta_E_L
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 64.0
Description: Delta value for L*.

Parameter: double SA_Delta_A_AB SA_Delta_A_AB

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 64.0

Description: Delta value for a*b* resp. a*.

Parameter: double SA_Delta_B SA_Delta_B

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 64.0

Description: Delta value for b*.

16.1.2.2.7 Set_ColorSpace (COLORSPACE)

Set the Colorspace used at color table.

Parameter: int32_t SP_ColorSpace SP_ColorSpace

Direction: Down

Valid values:

0= XYZ

1= LAB

Description: Colorspace to use

16.1.2.2.8 Get_ColorSpace (COLORSPACE)

Get the Colorspace used by color table.

Parameter: int32_t SA_ColorSpace SA_ColorSpace

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= XYZ

1= LAB

Description: Actually used colorspace

16.1.2.2.9 Move_Color (MOVECOLOR)

Move a color from one to another position in color table.

Parameter: int32_t SP_ActualPos SP_ActualPos

Direction: Down

Valid values:

Minimum: 1

Maximum: 16

Description: Actual color index.

16.1. Commands for ACS7000

Parameter: int32_t SP_NewPos SP_NewPos
Direction: Down
Valid values:
Minimum: 1
Maximum: 16
Description: New color index.

16.1.2.2.10 Reset_ColorMapping (RESETMAPPING)

Resets all color positions in color table to original positions.

16.1.2.2.11 Delete_Color (COLORDELETE)

Deletes a color from color table.

Parameter: String SP_ColorName SP_ColorName
Direction: Down
Description: Name of the color.

16.1.2.2.12 Clear_ColorTable

Clear the whole color table.

16.1.2.3 Measurement value processing

16.1.2.3.1 Set_VideoAverage (VSAVERAGE)

Set video averaging (before processing).

Parameter: int32_t SP_VideoAverage SP_VideoAverage
Direction: Down
Valid values:
 0= None
 1= Recursive over 2 lines (REC2)
 2= Recursive over 4 lines (REC4)
 3= Recursive over 8 lines (REC8)
 4= Recursive over 16 lines (REC16)
 5= Recursive over 32 lines (REC32)
 6= Recursive over 64 lines (REC64)
 7= Recursive over 128 lines (REC128)
Description: Averaging mode.

16.1.2.3.2 Get_VideoAverage (VSAVERAGE)

Get video averaging (before processing).

Parameter: int32_t SA_VideoAverage

SA_VideoAverage

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Recursive over 2 lines (REC2)
- 2= Recursive over 4 lines (REC4)
- 3= Recursive over 8 lines (REC8)
- 4= Recursive over 16 lines (REC16)
- 5= Recursive over 32 lines (REC32)
- 6= Recursive over 64 lines (REC64)
- 7= Recursive over 128 lines (REC128)

Description: Averaging mode.

16.1.2.3.3 Set_Averaging (AVERAGE)

Set data averaging at controller.

Parameter: int32_t SP_AveragingType

SP_AveragingType

Direction: Down

Valid values:

- 0= None
- 1= Moving average (MOVING)
- 2= Recursive averaging (RECURSIVE)
- 3= Median

Description: Averaging type.

Parameter: int32_t SP_MovingCount

SP_MovingCount

Direction: Down

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128
- 256
- 512
- 1024

Description: Number of value for the averaging window. This parameter is only used at moving average.

Parameter: int32_t SP_RecursiveCount

SP_RecursiveCount

Direction: Down

Valid values:

- Minimum:** 2
- Maximum:** 32768

Description: Number of values for recursive averaging. This parameter is only used at recursive average.

Parameter: int32_t SP_MedianCount SP_MedianCount

Direction: Down

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only used at median.

16.1.2.3.4 Get_Averaging (AVERAGE)

Get data averaging at controller.

Parameter: int32_t SA_AveragingType SA_AveragingType

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Moving average (MOVING)
- 2= Recursive averaging (RECURSIVE)
- 3= Median

Description: Averaging type.

Parameter: int32_t SA_MovingCount SA_MovingCount

Direction: Up

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128
- 256
- 512
- 1024

Description: Number of value for the averaging window. This parameter is only available at moving average.

Parameter: int32_t SA_RecursiveCount SA_RecursiveCount

Direction: Up

Valid values:

- Minimum:** 2
- Maximum:** 32768

Description: Number of values for recursive averaging. This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount SA_MedianCount

Direction: Up

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only available at median.

16.1.2.3.5 Set_StatisticSignal (STATISTICSIGNAL)

Set the measured value which is used for statistic calculation.

Parameter: int32_t SP_StatisticSignal

SP_StatisticSignal

Direction: Down

Valid values:

- 0= None
- 1= Color XYZ (XYZ)
- 2= Color RGB (RGB)
- 3= Color L*a*b* (LAB)
- 4= Color L*u*v* (LUV)
- 5= Color L*c*h* (LCH)
- 6= Color L*a*b*99 (LAB99)
- 7= Color L*c*h*99 (LCH99)
- 8= Distance 1 (DIST01)
- 9= Distance 2 (DIST02)
- 10= Distance 3 (DIST03)
- 11= Distance 4 (DIST04)
- 12= Distance 5 (DIST05)
- 13= Distance 6 (DIST06)
- 14= Distance 7 (DIST07)
- 15= Distance 8 (DIST08)
- 16= Distance 9 (DIST09)
- 17= Distance 10 (DIST10)
- 18= Distance 11 (DIST11)
- 19= Distance 12 (DIST12)
- 20= Distance 13 (DIST13)
- 21= Distance 14 (DIST14)
- 22= Distance 15 (DIST15)
- 23= Distance 16 (DIST16)
- 24= Minimum color distance (MINDIST)
- 25= No. of detected color (DETECTID)
- 26= No. of nearest color (MINDISTID)

Description: Value which is used for statistic calculation.

16.1.2.3.6 Get_StatisticSignal (STATISTICSIGNAL)

Get the measured value which is used for statistic calculation.

Parameter: int32_t SA_StatisticSignal

SA_StatisticSignal

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 1= Color XYZ (XYZ)
- 2= Color RGB (RGB)
- 3= Color L*a*b* (LAB)
- 4= Color L*u*v* (LUV)
- 5= Color L*c*h* (LCH)
- 6= Color L*a*b*99 (LAB99)
- 7= Color L*c*h*99 (LCH99)
- 8= Distance 1 (DIST01)

```

9= Distance 2 (DIST02)
10= Distance 3 (DIST03)
11= Distance 4 (DIST04)
12= Distance 5 (DIST05)
13= Distance 6 (DIST06)
14= Distance 7 (DIST07)
15= Distance 8 (DIST08)
16= Distance 9 (DIST09)
17= Distance 10 (DIST10)
18= Distance 11 (DIST11)
19= Distance 12 (DIST12)
20= Distance 13 (DIST13)
21= Distance 14 (DIST14)
22= Distance 15 (DIST15)
23= Distance 16 (DIST16)
24= Minimum color distance (MINDIST)
25= No. of detected color (DETECTID)
26= No. of nearest color (MINDISTID)

```

Description: Value which is used for statistic calculation.

16.1.2.3.7 Set_StatisticDepth (STATISTICDEPTH)

Set the window size for floating statistic calculation.

Parameter: int32_t SP_StatisticDepth

SP_StatisticDepth

Direction: Down

Valid values:

Minimum: 2

Maximum: 2147483647 (INT32_MAX)

Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 16384). Value greater as 16384 calculates statistic over all values.

16.1.2.3.8 Get_StatisticDepth (STATISTICDEPTH)

Get the window size for floating statistic calculation.

Parameter: int32_t SA_StatisticDepth

SA_StatisticDepth

Direction: Up

Valid values:

Minimum: 2

Maximum: 2147483647 (INT32_MAX)

Description: Window size for floating statistic calculation. The value must be power of two (2, 4, 8, ..., 16384). Value greater as 16384 calculates statistic over all values.

16.1.2.3.9 Reset_Statistic (RESETSTATISTIC)

Reset the statistic (min and max values).

16.1.3 Data output

16.1.3.1 General

16.1.3.1.1 Set_DataOutInterface (OUTPUT)

Set the active interface for data output.

At this command the controller may change output data automatically. If first bit of [IP_AutomaticMode](#) is set (1), [Get_AllParameters](#) (SP_Additional= 0) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DataOutInterface

SP_DataOutInterface

Direction: Down

Valid values:

- 0= None
- 1= RS422
- 2= Ethernet
- 3= HTTP
- 4= Ethercat

Description: Active interface for data output.

16.1.3.1.2 Get_DataOutInterface (OUTPUT)

Get the active interface for data output.

Parameter: int32_t SA_DataOutInterface

SA_DataOutInterface

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= RS422
- 2= Ethernet
- 3= HTTP
- 4= Ethercat

Description: Active interface for data output.

16.1.3.1.3 Set_Resampling (OUTREDUCE)

Set resampling to reduce output data.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_Resampling

SP_Resampling

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 2000

Description: Resampling value.

16.1. Commands for ACS7000

Parameter: int32_t SP_ResampleRS422	SP_ResampleRS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if RS422 output should be resampled.
Parameter: int32_t SP_ResampleEthernet	SP_ResampleEthernet
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if output over ethernet should be resampled.

16.1.3.1.4 Get_Resampling (OUTREDUCE)

Get resampling for reducing output data.

Parameter: int32_t SA_Resampling	SA_Resampling
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2000	
Description:	Resampling value.
Parameter: int32_t SA_ResampleRS422	SA_ResampleRS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	RS422 output is resampled.
Parameter: int32_t SA_ResampleEthernet	SA_ResampleEthernet
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Output over ethernet is resampled.

16.1.3.1.5 Set_HoldLastValid (OUTHOLD)

Set the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SP_HoldLastValid	SP_HoldLastValid
Direction: Down	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description:	Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).

16.1.3.1.6 Get_HoldLastValid (OUTHOLD)

Get the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SA_HoldLastValid

SA_HoldLastValid

Direction: Up

Valid values:

Minimum: -1

Maximum: 1024

Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).

16.1.3.2 Selected measurement values

16.1.3.2.1 Set_OutputVideo_ETH (OUTVIDEO)

Set the video signal to be output at ethernet interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputVideoRaw_ETH

SP_OutputVideoRaw_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if raw video signal is transmitted.

Parameter: int32_t SP_OutputVideoDark_ETH

SP_OutputVideoDark_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if dark corrected video signal is transmitted.

Parameter: int32_t SP_OutputVideoLinearized_ETH

SP_OutputVideoLin-
earized_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if linearized video signal is transmitted.

Parameter: int32_t SP_OutputVideoLightSpectrum_ETH

SP_OutputVideoLightSpec-
trum_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if light referenced video signal is transmitted.

16.1.3.2.2 Get_OutputVideo_ETH (OUTVIDEO)

Get the video signal to be output at ethernet interface.

Parameter: int32_t SA_OutputVideoRaw_ETH

SA_OutputVideoRaw_ETH

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if raw video signal is transmitted.

Parameter: int32_t SA_OutputVideoDark_ETH

SA_OutputVideoDark_ETH

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if dark corrected video signal is transmitted.

Parameter: int32_t SA_OutputVideoLinearized_ETH

SA_OutputVideoLin-
earized_ETH

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if linearized video signal is transmitted.

Parameter: int32_t SA_OutputVideoLightSpectrum_ETH

SA_OutputVideoLightSpec-
trum_ETH

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if light referenced signal is transmitted.

16.1.3.2.3 Set_OutputColor_ETH (OUTCOLOR_ETH)

Set color data to be output at ETH interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputColorXYZ_ETH

SP_OutputColorXYZ_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if measured data in XYZ is transmitted.

Parameter: int32_t SP_OutputColorRGB_ETH

SP_OutputColorRGB_ETH

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if measured data in RGB is transmitted.

Parameter: int32_t SP_OutputColorLAB_ETH	SP_OutputColorLAB_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b* is transmitted.	
Parameter: int32_t SP_OutputColorLUV_ETH	SP_OutputColorLUV_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*u*v* is transmitted.	
Parameter: int32_t SP_OutputColorLCH_ETH	SP_OutputColorLCH_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h* is transmitted.	
Parameter: int32_t SP_OutputColorLAB99_ETH	SP_OutputColorLAB99_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b*99 is transmitted.	
Parameter: int32_t SP_OutputColorLCH99_ETH	SP_OutputColorLCH99_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h*99 is transmitted.	

16.1.3.2.4 Get_OutputColor_ETH (OUTCOLOR_ETH)

Get color data to be output at ETH interface.

Parameter: int32_t SA_OutputColorXYZ_ETH	SA_OutputColorXYZ_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in XYZ is transmitted.	
Parameter: int32_t SA_OutputColorRGB_ETH	SA_OutputColorRGB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in RGB is transmitted.	

Parameter: int32_t SA_OutputColorLAB_ETH	SA_OutputColorLAB_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b* is transmitted.	
Parameter: int32_t SA_OutputColorLUV_ETH	SA_OutputColorLUV_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*u*v* is transmitted.	
Parameter: int32_t SA_OutputColorLCH_ETH	SA_OutputColorLCH_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h* is transmitted.	
Parameter: int32_t SA_OutputColorLAB99_ETH	SA_OutputColorLAB99_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b*99 is transmitted.	
Parameter: int32_t SA_OutputColorLCH99_ETH	SA_OutputColorLCH99_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h*99 is transmitted.	

16.1.3.2.5 Set_OutputColor_RS422 (OUTCOLOR_RS422)

Set color data to be output at RS422 interface.

For this command an [Update...](#) and a [Reset...](#) meta command is available.

Parameter: int32_t SP_OutputColorXYZ_RS422	SP_OutputColorXYZ_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in XYZ is transmitted.	
Parameter: int32_t SP_OutputColorRGB_RS422	SP_OutputColorRGB_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in RGB is transmitted.	

Parameter: int32_t SP_OutputColorLAB_RS422	SP_OutputColorLAB_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b* is transmitted.	
Parameter: int32_t SP_OutputColorLUV_RS422	SP_OutputColorLUV_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*u*v* is transmitted.	
Parameter: int32_t SP_OutputColorLCH_RS422	SP_OutputColorLCH_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h* is transmitted.	
Parameter: int32_t SP_OutputColorLAB99_RS422	SP_OutputColorLAB99_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b*99 is transmitted.	
Parameter: int32_t SP_OutputColorLCH99_RS422	SP_OutputColorLCH99_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h*99 is transmitted.	

16.1.3.2.6 Get_OutputColor_RS422 (OUTCOLOR_RS422)

Get color data to be output at RS422 interface.

Parameter: int32_t SA_OutputColorXYZ_RS422	SA_OutputColorXYZ_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in XYZ is transmitted.	
Parameter: int32_t SA_OutputColorRGB_RS422	SA_OutputColorRGB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in RGB is transmitted.	

Parameter: int32_t SA_OutputColorLAB_RS422	SA_OutputColorLAB_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b* is transmitted.	
Parameter: int32_t SA_OutputColorLUV_RS422	SA_OutputColorLUV_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*u*v* is transmitted.	
Parameter: int32_t SA_OutputColorLCH_RS422	SA_OutputColorLCH_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h* is transmitted.	
Parameter: int32_t SA_OutputColorLAB99_RS422	SA_OutputColorLAB99_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*a*b*99 is transmitted.	
Parameter: int32_t SA_OutputColorLCH99_RS422	SA_OutputColorLCH99_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if measured data in L*c*h*99 is transmitted.	

16.1.3.2.7 Set_DistanceMode (DISTANCEMODE)

Set the distance mode.

If first bit of **IP_AutomaticMode** is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually.

Parameter: int32_t SP_DistanceMode	SP_DistanceMode
Direction: Down	
Valid values:	
0= Best hit (BESTHIT)	
1= Selection	
Description: Distance mode.	

16.1.3.2.8 Get_DistanceMode (DISTANCEMODE)

Get the distance mode.

Parameter: int32_t SA_DistanceMode

SA_DistanceMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Best hit (BESTHIT)
- 1 = Selection

Description: Distance mode.

16.1.3.2.9 Set_OutputDistance_ETH (OUTDIST_ETH)

Set the color distance data to be output at ethernet interface.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputDistDistance01_ETH

SP_OutputDistDistance01_-
ETH

Direction: Down

Valid values:

- 0 = no
- 1 = yes

Description: Specify if color distance 1 is transmitted.

Parameter: int32_t SP_OutputDistDistance02_ETH

SP_OutputDistDistance02_-
ETH

Direction: Down

Valid values:

- 0 = no
- 1 = yes

Description: Specify if color distance 2 is transmitted.

Parameter: int32_t SP_OutputDistDistance03_ETH

SP_OutputDistDistance03_-
ETH

Direction: Down

Valid values:

- 0 = no
- 1 = yes

Description: Specify if color distance 3 is transmitted.

Parameter: int32_t SP_OutputDistDistance04_ETH

SP_OutputDistDistance04_-
ETH

Direction: Down

Valid values:

- 0 = no
- 1 = yes

Description: Specify if color distance 4 is transmitted.

Parameter: int32_t SP_OutputDistDistance05_ETH

SP_OutputDistDistance05_-
ETH

Direction: Down

Valid values:

- 0 = no
- 1 = yes

Description: Specify if color distance 5 is transmitted.

16.1. Commands for ACS7000

Parameter: int32_t SP_OutputDistDistance06_ETH	SP_OutputDistDistance06_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 6 is transmitted.	
Parameter: int32_t SP_OutputDistDistance07_ETH	SP_OutputDistDistance07_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 7 is transmitted.	
Parameter: int32_t SP_OutputDistDistance08_ETH	SP_OutputDistDistance08_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 8 is transmitted.	
Parameter: int32_t SP_OutputDistDistance09_ETH	SP_OutputDistDistance09_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 9 is transmitted.	
Parameter: int32_t SP_OutputDistDistance10_ETH	SP_OutputDistDistance10_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 10 is transmitted.	
Parameter: int32_t SP_OutputDistDistance11_ETH	SP_OutputDistDistance11_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 11 is transmitted.	
Parameter: int32_t SP_OutputDistDistance12_ETH	SP_OutputDistDistance12_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 12 is transmitted.	
Parameter: int32_t SP_OutputDistDistance13_ETH	SP_OutputDistDistance13_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 13 is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SP_OutputDistDistance14_ETH	SP_OutputDistDistance14_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 14 is transmitted.	
Parameter: int32_t SP_OutputDistDistance15_ETH	SP_OutputDistDistance15_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 15 is transmitted.	
Parameter: int32_t SP_OutputDistDistance16_ETH	SP_OutputDistDistance16_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 16 is transmitted.	
Parameter: int32_t SP_OutputDistMinDistance_ETH	SP_OutputDistMinDis- tance_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if minimum color distance is transmitted.	
Parameter: int32_t SP_OutputDistDetectedID_ETH	SP_OutputDistDetectedID_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of detected color is transmitted.	
Parameter: int32_t SP_OutputDistMinDistID_ETH	SP_OutputDistMinDistID_- ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of nearest color is transmitted.	

16.1.3.2.10 Get_OutputDistance_ETH (OUTDIST_ETH)

Get the color distance data to be output at ethernet interface.

Parameter: int32_t SA_OutputDistDistance01_ETH	SA_OutputDistDistance01_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 1 is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SA_OutputDistDistance02_ETH	SA_OutputDistDistance02_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 2 is transmitted.	
Parameter: int32_t SA_OutputDistDistance03_ETH	SA_OutputDistDistance03_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 3 is transmitted.	
Parameter: int32_t SA_OutputDistDistance04_ETH	SA_OutputDistDistance04_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 4 is transmitted.	
Parameter: int32_t SA_OutputDistDistance05_ETH	SA_OutputDistDistance05_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 5 is transmitted.	
Parameter: int32_t SA_OutputDistDistance06_ETH	SA_OutputDistDistance06_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 6 is transmitted.	
Parameter: int32_t SA_OutputDistDistance07_ETH	SA_OutputDistDistance07_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 7 is transmitted.	
Parameter: int32_t SA_OutputDistDistance08_ETH	SA_OutputDistDistance08_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 8 is transmitted.	
Parameter: int32_t SA_OutputDistDistance09_ETH	SA_OutputDistDistance09_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 9 is transmitted.	

Parameter: int32_t SA_OutputDistDistance10_ETH	SA_OutputDistDistance10_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 10 is transmitted.	
Parameter: int32_t SA_OutputDistDistance11_ETH	SA_OutputDistDistance11_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 11 is transmitted.	
Parameter: int32_t SA_OutputDistDistance12_ETH	SA_OutputDistDistance12_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 12 is transmitted.	
Parameter: int32_t SA_OutputDistDistance13_ETH	SA_OutputDistDistance13_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 13 is transmitted.	
Parameter: int32_t SA_OutputDistDistance14_ETH	SA_OutputDistDistance14_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 14 is transmitted.	
Parameter: int32_t SA_OutputDistDistance15_ETH	SA_OutputDistDistance15_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 15 is transmitted.	
Parameter: int32_t SA_OutputDistDistance16_ETH	SA_OutputDistDistance16_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color distance 16 is transmitted.	
Parameter: int32_t SA_OutputDistMinDistance_ETH	SA_OutputDistMinDis- tance_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if minimum color distance is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SA_OutputDistDetectedID_ETH	SA_OutputDistDetectedID_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of detected color is transmitted.	
Parameter: int32_t SA_OutputDistMinDistID_ETH	SA_OutputDistMinDistID_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of nearest color is transmitted.	

16.1.3.2.11 Set_OutputDistance_RS422 (OUTDIST_RS422)

Set the color distance data to be output at RS422 interface.
 For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputDistMinDistance_RS422	SP_OutputDistMinDis- tance_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if minimum color distance is transmitted.	
Parameter: int32_t SP_OutputDistDetectedID_RS422	SP_OutputDistDetectedID_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of detected color is transmitted.	
Parameter: int32_t SP_OutputDistMinDistID_RS422	SP_OutputDistMinDistID_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of nearest color is transmitted.	

16.1.3.2.12 Get_OutputDistance_RS422 (OUTDIST_RS422)

Get the color distance data to be output at RS422 interface.

Parameter: int32_t SA_OutputDistMinDistance_RS422	SA_OutputDistMinDis- tance_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if minimum color distance is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SA_OutputDistDetectedID_RS422	SA_OutputDistDetectedID_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of detected color is transmitted.	
Parameter: int32_t SA_OutputDistMinDistID_RS422	SA_OutputDistMinDistID_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of nearest color is transmitted.	

16.1.3.2.13 Set_OutputStatistic_ETH (OUTSTATISTIC_ETH)

Set statistic data to be output at ETH interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputStatisticMin_ETH	SP_OutputStatisticMin_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic minimum value is transmitted.	
Parameter: int32_t SP_OutputStatisticMax_ETH	SP_OutputStatisticMax_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic maximum value is transmitted.	
Parameter: int32_t SP_OutputStatisticPeak2Peak_ETH	SP_OutputStatistic- Peak2Peak_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic peak to peak value is transmitted.	

16.1.3.2.14 Get_OutputStatistic_ETH (OUTSTATISTIC_ETH)

Get statistic data to be output at ETH interface.

Parameter: int32_t SA_OutputStatisticMin_ETH	SA_OutputStatisticMin_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic minimum value is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SA_OutputStatisticMax_ETH	SA_OutputStatisticMax_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic maximum value is transmitted.	
Parameter: int32_t SA_OutputStatisticPeak2Peak_ETH	SA_OutputStatisticPeak2Peak_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic peak to peak value is transmitted.	

16.1.3.2.15 Set_OutputStatistic_RS422 (OUTSTATISTIC_RS422)

Set statistic data to be output at RS422 interface.
 For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputStatisticMin_RS422	SP_OutputStatisticMin_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic minimum value is transmitted.	
Parameter: int32_t SP_OutputStatisticMax_RS422	SP_OutputStatisticMax_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic maximum value is transmitted.	
Parameter: int32_t SP_OutputStatisticPeak2Peak_RS422	SP_OutputStatisticPeak2Peak_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic peak to peak value is transmitted.	

16.1.3.2.16 Get_OutputStatistic_RS422 (OUTSTATISTIC_RS422)

Get statistic data to be output at RS422 interface.

Parameter: int32_t SA_OutputStatisticMin_RS422	SA_OutputStatisticMin_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic minimum value is transmitted.	

Parameter: int32_t SA_OutputStatisticMax_RS422	SA_OutputStatisticMax_- RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic maximum value is transmitted.	
Parameter: int32_t SA_OutputStatisticPeak2Peak_RS422	SA_OutputStatistic- Peak2Peak_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if statistic peak to peak value is transmitted.	

16.1.3.2.17 Set_DigitalOutDistance (OUTDIST_COLOROUT)

Set the color distance data at digital outputs.

Parameter: int32_t SP_DigitalOutDetectedID	SP_DigitalOutDetectedID
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of detected color is output.	

16.1.3.2.18 Get_DigitalOutDistance (OUTDIST_COLOROUT)

Get the color distance data at digital outputs.

Parameter: int32_t SA_DigitalOutDetectedID	SA_DigitalOutDetectedID
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if no. of detected color is output.	

16.1.3.2.19 Set_OutputStatus_ETH (OUTSTATUS_ETH)

Set status data to be output at ETH interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_OutputStatusFramerate_ETH	SP_OutputStatusFramer- ate_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if framerate is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SP_OutputStatusShutter_ETH	SP_OutputStatusShutter_-ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SP_OutputStatusTempDetector_ETH	SP_OutputStatusTempDetector_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature of detector is transmitted.	
Parameter: int32_t SP_OutputStatusTempLightSrc_ETH	SP_OutputStatusTempLight-Src_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature of light source is transmitted.	
Parameter: int32_t SP_OutputLightSensorRed_ETH	SP_OutputLightSensorRed_-ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if red part of light sensor is transmitted.	
Parameter: int32_t SP_OutputLightSensorGreen_ETH	SP_OutputLightSensor-Green_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if green part of light sensor is transmitted.	
Parameter: int32_t SP_OutputLightSensorBlue_ETH	SP_OutputLightSensor-Blue_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if blue part of light sensor is transmitted.	
Parameter: int32_t SP_OutputLightSensorBright_ETH	SP_OutputLightSensor-Bright_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if brightness of light sensor is transmitted.	
Parameter: int32_t SP_OutputStatusCounter_ETH	SP_OutputStatusCounter_-ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	

Parameter: int32_t SP_OutputStatusTimestamp_ETH
Direction: Down
Valid values:
 0 = no
 1 = yes
Description: Specify if timestamp is transmitted.

Parameter: int32_t SP_OutputStatusError_ETH SP_OutputStatusError_ETH
Direction: Down
Valid values:
 0 = no
 1 = yes
Description: Specify if error flags are transmitted.

16.1.3.2.20 Get_OutputStatus_ETH (OUTSTATUS_ETH)

Get status data to be output at ETH interface.

Parameter: int32_t SA_OutputStatusFramerate_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if framerate is transmitted.

Parameter: int32_t SA_OutputStatusShutter_ETH
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if shutter time is transmitted.

Parameter: int32_t SA_OutputStatusTempDetector_ETH
Direction: Up
Valid values:
 0 = no
 1 = yes
Description: Specify if temperature of detector is transmitted

Parameter: int32_t SA_OutputStatusTempLightSrc_ETH
Direction: Up
Valid values:
 0 = no
 1 = yes
Description: Specify if temperature of light source is transmitted

Parameter: int32_t SA_OutputLightSensorRed_ETH
Direction: Up
Valid values:
0 = no
1 = yes
Description: Specify if red part of light sensor is transmitted

Parameter: int32_t SA_OutputLightSensorGreen_ETH	SA_OutputLightSensor-Green_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if green part of light sensor is transmitted.	
Parameter: int32_t SA_OutputLightSensorBlue_ETH	SA_OutputLightSensor-Blue_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if blue part of light sensor is transmitted.	
Parameter: int32_t SA_OutputLightSensorBright_ETH	SA_OutputLightSensor-Bright_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if brightness of light sensor is transmitted.	
Parameter: int32_t SA_OutputStatusCounter_ETH	SA_OutputStatusCounter-ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputStatusTimestamp_ETH	SA_OutputStatusTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputStatusError_ETH	SA_OutputStatusError_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if error flags are transmitted.	

16.1.3.2.21 Set_OutputStatus_RS422 (OUTSTATUS_RS422)

Set status data to be output at RS422 interface.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputStatusFramerate_RS422	SP_OutputStatusFramerate_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if framerate is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SP_OutputStatusShutter_RS422	SP_OutputStatusShutter_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if shutter time is transmitted.	
Parameter: int32_t SP_OutputStatusTempDetector_RS422	SP_OutputStatusTempDetec- tor_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature of detector is transmitted.	
Parameter: int32_t SP_OutputStatusTempLightSrc_RS422	SP_OutputStatusTempLight- Src_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature of light source is transmitted.	
Parameter: int32_t SP_OutputLightSensorRed_RS422	SP_OutputLightSensorRed_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if red part of light sensor is transmitted.	
Parameter: int32_t SP_OutputLightSensorGreen_RS422	SP_OutputLightSensor- Green_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if green part of light sensor is transmitted.	
Parameter: int32_t SP_OutputLightSensorBlue_RS422	SP_OutputLightSensor- Blue_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if blue part of light sensor is transmitted.	
Parameter: int32_t SP_OutputLightSensorBright_RS422	SP_OutputLightSensor- Bright_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if brightness of light sensor is transmitted.	
Parameter: int32_t SP_OutputStatusCounter_RS422	SP_OutputStatusCounter_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	

16.1. Commands for ACS7000

Parameter: int32_t SP_OutputStatusTimestamp_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if timestamp is transmitted.

SP_OutputStatusTimestamp_RS422

Parameter: int32_t SP_OutputStatusError_RS422
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if error flags are transmitted.

SP_OutputStatusError_RS422

16.1.3.2.22 Get_OutputStatus_RS422 (OUTSTATUS_RS422)

Get status data to be output at RS422 interface.

Parameter: int32_t SA_OutputStatusFramerate_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if framerate is transmitted.

SA_OutputStatusFramerate_RS422

Parameter: int32_t SA_OutputStatusShutter_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if shutter time is transmitted.

SA_OutputStatusShutter_RS422

Parameter: int32_t SA_OutputStatusTempDetector_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if temperature of detector is transmitted.

SA_OutputStatusTempDetector_RS422

Parameter: int32_t SA_OutputStatusTempLightSrc_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if temperature of light source is transmitted.

SA_OutputStatusTempLightSrc_RS422

Parameter: int32_t SA_OutputLightSensorRed_RS422
Direction: Up
Valid values:
 0= no
 1= yes
Description: Specify if red part of light sensor is transmitted.

SA_OutputLightSensorRed_RS422

Parameter: int32_t SA_OutputLightSensorGreen_RS422	SA_OutputLightSensor-Green_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if green part of light sensor is transmitted.	
Parameter: int32_t SA_OutputLightSensorBlue_RS422	SA_OutputLightSensor-Blue_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if blue part of light sensor is transmitted.	
Parameter: int32_t SA_OutputLightSensorBright_RS422	SA_OutputLightSensor-Bright_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if brightness of light sensor is transmitted.	
Parameter: int32_t SA_OutputStatusCounter_RS422	SA_OutputStatusCounter_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if counter is transmitted.	
Parameter: int32_t SA_OutputStatusTimestamp_RS422	SA_OutputStatusTimestamp_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_OutputStatusError_RS422	SA_OutputStatusError_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if error flags are transmitted.	

16.2 Commands for CFO100/200

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [TCP/IP](#) (native).
- [RS232](#) (native, also over sensor internal USB interface and MEDAQLib RS232 high level interface).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are not used.
- Scaled values are filled from the JSON result received by WebSocket connection.

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

The binary array of bit coded outputs is represented by one number, placing first array entry at lowest bit.

The chosen matcher id (UUID) cannot be interpreted by one value. So a translation table is generated by MEDAQLib and can be read using command [Get_SensorMatcherIDList](#). The value here is the index (base 1) in the table.

The binary array of bit coded inputs is represented by one number, placing 'trigger_0_edge_falling' at lowest bit. The next bits are 'trigger_0_edge_rising', 'trigger_0_level_high', 'trigger_0_level_low', 'trigger_1_edge_falling', 'trigger_1_edge_rising', 'trigger_1_level_high', 'trigger_1_level_low', ..., 'trigger_3_level_low'.

16.2.1 Access

16.2.1.1 Get_AccessControl (GET /api/access)

Inspect the available aspects of the access control setup.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SA.Actions

SA.Actions

Direction: Up

Description: Newline separated array of available actions that can be allowed or denied via permissions.

Parameter: String SA.Scopes

SA.Scopes

Direction: Up

Description: Newline separated array of available scopes that can be accessed with the different actions.

16.2.1.2 Get_UserRoles (GET /api/access/roles)

Retrieve access roles.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SA.UserRoleEntries

SA.UserRoleEntries

Direction: Up

Valid values:

2

Description: Number of entries in the roles array. All following parameters exists from 1 to this number, e.g. SA.UserRole1ID, SA.UserRole2ID, ...

Parameter: String SA_UserRole1..xID	SA_UserRole1..xID
Direction: Up	
Valid values:	
anonymous	
administrator	
Description: ID of the role.	
Parameter: int32_t SA_UserRole1..xPermissionEntries	SA_User-
Direction: Up	Role1..xPermissionEntries
Valid values:	
6	
Description: Number of entries in the permission array. All following parameters exists from 1 to this number, e.g. SA_UserRole1Permission1Scope, SA_UserRole1Permission2Scope, ...	
Parameter: String SA_UserRole1..xPermission1..yScope	SA_User-
Direction: Up	Role1..xPermission1..yScope
Valid values:	
access	
network	
peripherals	
sensor	
settings	
system	
Description: Scope of the permission.	
Parameter: String SA_UserRole1..xPermission1..yAllow	SA_User-
Direction: Up	Role1..xPermission1..yAllow
Valid values:	
view, edit	
Description: Newline separated array of allowance of the permission.	

16.2.1.3 Add_User (POST /api/access/users)

Add user with name and password.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_UserName	SP_UserName
Direction: Down	
Description: Unique name identifying an account.	
Parameter: String SP_UserPassword	SP_UserPassword
Direction: Down	
Default: <not specified>, do not send this parameter to sensor.	
Description: Password assigned to this account. At least either a password or it's hash must be specified.	
Parameter: String SP_UserPasswordHash	SP_UserPasswordHash
Direction: Down	
Default: <not specified>, do not send this parameter to sensor.	
Description: Password hash assigned to this account.	

Parameter: String SP_UserRoles	SP_UserRoles
Direction: Down	
Valid values:	
administrator	
<empty>	
Description:	Newline separated array of roles assigned to an account define its set of permissions. Currently only one element is supported by sensor.
Parameter: String SA_UserName	SA_UserName
Direction: Up	
Description:	User name.
Parameter: String SA_UserPasswordHash	SA_UserPasswordHash
Direction: Up	
Description:	Crypted password.
Parameter: String SA_UserRoles	SA_UserRoles
Direction: Up	
Valid values:	
administrator	
Description:	Newline separated array of roles assigned to an account define its set of permissions.
16.2.1.4 Set_User (PUT /api/access/users/{name})	
Modify user.	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Parameter: String SP_UserName	SP_UserName
Direction: Down	
Description:	Unique name identifying an account.
Parameter: String SP_UserPassword	SP_UserPassword
Direction: Down	
Default:	<not specified>, do not send this parameter to sensor.
Description:	Password assigned to this account. At least either a password or it's hash must be specified.
Parameter: String SP_UserPasswordHash	SP_UserPasswordHash
Direction: Down	
Default:	<not specified>, do not send this parameter to sensor.
Description:	Password hash assigned to this account.
Parameter: String SP_UserRoles	SP_UserRoles
Direction: Down	
Valid values:	
administrator	
<empty>	
Description:	Newline separated array of roles assigned to an account define its set of permissions. Currently only one element is supported by sensor.

Parameter: String SA_UserName	SA_UserName
Direction: Up	
Description: User name.	
Parameter: String SA_UserPasswordHash	SA_UserPasswordHash
Direction: Up	
Description: Crypted password.	
Parameter: String SA_UserRoles	SA_UserRoles
Direction: Up	
Valid values:	
administrator	
Description: Newline separated array of roles assigned to an account define its set of permissions.	

16.2.1.5 Get_User (GET /api/access/users/{name})

Get single user.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_UserName	SP_UserName
Direction: Down	
Description: Unique name identifying an account.	
Parameter: String SA_UserName	SA_UserName
Direction: Up	
Description: Unique name identifying an account.	
Parameter: String SA_UserPasswordHash	SA_UserPasswordHash
Direction: Up	
Description: Password hash assigned to this account.	
Parameter: String SA_UserRoles	SA_UserRoles
Direction: Up	
Valid values:	
administrator	
Description: Newline separated array of roles assigned to an account define its set of permissions.	

16.2.1.6 Get_UserList (GET /api/access/users)

Get user management.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SA_UserEntries	SA_UserEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of entries in the user array. All following parameters exists from 1 to this number, e.g. SA_User1Name, SA_User2Name, ...	

Parameter: String SA_User1..xName	SA_User1..xName
Direction: Up	
Description: Unique name identifying an account.	
Parameter: String SA_User1..xPasswordHash	SA_User1..xPasswordHash
Direction: Up	
Description: Password hash assigned to this account.	
Parameter: String SA_User1..xRoles	SA_User1..xRoles
Direction: Up	
Valid values:	
administrator	
Description: Newline separated array of roles assigned to an account define its set of permissions.	

16.2.1.7 Logout (DELETE /api/access/login, access logout)

Invalidate any current credentials.

16.2.1.8 Login (POST /api/access/login, access login)

Login into an account.

Parameter: String SP_UserName	SP_UserName
Direction: Down	
Description: The name of the user that should be authenticated.	
Parameter: String SP_UserPassword	SP_UserPassword
Direction: Down	
Description: The password of the user that should be authenticated.	
Parameter: int32_t SP_UserSessionTimeout	SP_UserSessionTimeout
Direction: Down	
Unit: s	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: The lifetime of the session on the server-side in seconds. Passing 0 allows for checking credentials without creating a new session.	
Parameter: String SA_UserName	SA_UserName
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: The currently logged in user. This parameter is only available if returned from sensor.	

Parameter: int32_t SA_UserSessionTimeout	SA_UserSessionTimeout
Direction: Up	
Unit: s	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Number of seconds this session has left before expiring. This parameter is only available if returned from sensor.
Parameter: String SA_TimeCreated	SA_TimeCreated
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Representation of login time corresponding to ISO 8601.
Parameter: String SA_Timeout	SA_Timeout
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Representation of timeout.
Parameter: String SA_TimeExpires	SA_TimeExpires
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Representation of expiring time corresponding to ISO 8601.

16.2.1.9 Get_CurrentUser (GET /api/access/login, access session)

Retrieve Information about the currently used Credentials.

Parameter: String SA_UserName	SA_UserName
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	The currently logged in user. This parameter is only available if returned from sensor.
Parameter: int32_t SA_UserSessionTimeout	SA_UserSessionTimeout
Direction: Up	
Unit: s	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Number of seconds this session has left before expiring. This parameter is only available if returned from sensor.

Parameter: String SA_TimeCreated	SA_TimeCreated
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Representation of login time corresponding to ISO 8601. This parameter is only available if returned from sensor.
Parameter: String SA_Timeout	SA_Timeout
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Representation of timeout. This parameter is only available if returned from sensor.
Parameter: String SA_TimeExpires	SA_TimeExpires
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Representation of expiring time corresponding to ISO 8601. This parameter is only available if returned from sensor.

16.2.1.10 Delete_User (DELETE /api/access/users/{name})

Delete user.

Valid for sensor:
CFO100 / CFO200 over Ethernet

Parameter: String SP_UserName	SP_UserName
Direction: Down	
Description:	Unique name identifying an account.

16.2.1.11 Delete_UserList (DELETE /api/access/users)

Delete all users from sensor.

Valid for sensor:
CFO100 / CFO200 over Ethernet

16.2.2 Defaults

16.2.2.1 Add_Defaults (POST /api/defaults)

Add a default setting for special sensor actions.

Valid for sensor:
CFO100 / CFO200 over Ethernet

Parameter: String SP_DefaultObjectType	SP_DefaultObjectType
Direction: Down	
Description:	Name of the object.

Parameter: String SP_DefaultKey	SP_DefaultKey
Direction: Down	
Description: Name of the attribute.	
Parameter: String SP_DefaultValue	SP_DefaultValue
Direction: Down	
Description: Value as number, boolean, text or object in JSON notation.	
Parameter: String SA_DefaultUUID	SA_DefaultUUID
Direction: Up	
Description: Unique ID of the default setting.	
Parameter: String SA_DefaultObjectType	SA_DefaultObjectType
Direction: Up	
Description: Name of the object.	
Parameter: String SA_DefaultKey	SA_DefaultKey
Direction: Up	
Description: Name of the attribute.	
Parameter: String SA_DefaultValue	SA_DefaultValue
Direction: Up	
Description: Value as number, boolean, text or object in JSON notation.	
16.2.2.2 Set_Defaults (PUT /api/defaults/{itemId}, matcher default set <hold_time tolerance>)	
Set a default setting for special sensor actions.	
Parameter: String SP_DefaultUUID	SP_DefaultUUID
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Unique ID of the default setting.	
Parameter: String SP_DefaultValue	SP_DefaultValue
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Value as number, boolean, text or object in JSON notation.	
Parameter: int32_t SP_DefaultsProperty	SP_DefaultsProperty
Direction: Down	
Valid values:	
1 = Hold time	
2 = Tolerance	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description: Name of default setting.	

Parameter: double SP_DefaultsHoldTime	SP_DefaultsHoldTime
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 1.79769e+308 (DBL_MAX)	
Unit: s	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Default minimum duration of output signal in seconds. Only used if SP_DefaultsProperty is Hold time
Parameter: int32_t SP_DefaultsToleranceShape	SP_DefaultsToleranceShape
Direction: Down	
Valid values:	
0= Infinite	
1= Sphere	
2= Cylinder	
3= Box	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Default tolerance shape. Only used if SP_DefaultsProperty is Tolerance.
Parameter: double SP_DefaultsToleranceShapeSphereLimitRadius	SP_DefaultsTolerance-ShapeSphereLimitRadius
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Default radius for shape Sphere. Only used if SP_DefaultsToleranceShape is Sphere.
Parameter: double SP_DefaultsToleranceShapeCylinderLimitHalfHeight	SP_DefaultsTolerance-ShapeCylinderLimitHalfHeight
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Default half height for shape Cylinder. Only used if SP_DefaultsToleranceShape is Cylinder.
Parameter: double SP_DefaultsToleranceShapeCylinderLimitRadius	SP_DefaultsTolerance-ShapeCylinderLimitRadius
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Default radius for shape Cylinder. Only used if SP_DefaultsToleranceShape is Cylinder.

Parameter: double SP_DefaultsToleranceShapeBoxLimitHalfEdge1..3	SP_DefaultsTolerance-ShapeBoxLimitHalfEdge1..3
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Default half edges for shape Box. Only used if SP_DefaultsToleranceShape is Box.
Parameter: String SA_DefaultUUID	SA_DefaultUUID
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Unique ID of the default setting.
Parameter: String SA_DefaultObjectType	SA_DefaultObjectType
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Name of the object.
Parameter: String SA_DefaultKey	SA_DefaultKey
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Name of the attribute.
Parameter: String SA_DefaultValue	SA_DefaultValue
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Value as number, boolean, text or object in JSON notation.

16.2.2.3 Get_Defaults (GET /api/defaults/{itemId}, matcher default show)

Get a default setting for special sensor actions.

Parameter: String SP_DefaultUUID	SP_DefaultUUID
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Unique ID of the default setting.
Parameter: int32_t SP_DefaultsProperty	SP_DefaultsProperty
Direction: Down	
Valid values:	
0= All	
1= Hold time	
2= Tolerance	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Default:	<not specified>, do not send this parameter to sensor (means All).
Description:	Name of default setting.

Parameter: String SA_DefaultUUID	SA_DefaultUUID
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Unique ID of the default setting.	
Parameter: String SA_DefaultObjectType	SA_DefaultObjectType
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Name of the object.	
Parameter: String SA_DefaultKey	SA_DefaultKey
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Name of the attribute.	
Parameter: String SA_DefaultValue	SA_DefaultValue
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Value as number, boolean, text or object in JSON notation.	
Parameter: double SA_DefaultsHoldTime	SA_DefaultsHoldTime
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.79769e+308 (DBL_MAX)	
Unit: s	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description: Default minimum duration of output signal in seconds. Only available if SP_DefaultsProperty is all or Hold time	
Parameter: int32_t SA_DefaultsToleranceShape	SA_DefaultsToleranceShape
Direction: Up	
Valid values:	
0= Infinite	
1= Sphere	
2= Cylinder	
3= Box	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description: Default tolerance shape. Only available if SP_DefaultsProperty is all or Tolerance.	
Parameter: double SA_DefaultsToleranceShapeSphereLimitRadius	SA_DefaultsTolerance-ShapeSphereLimitRadius
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description: Default radius for shape Sphere. Only available if SA_DefaultsToleranceShape is Sphere.	

Parameter: double SA_DefaultsToleranceShapeCylinderLimitHalfHeight	SA_DefaultsToler- anceShapeCylinderLim- itHalfHeight
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Default half height for shape Cylinder. Only available if SA_- DefaultsToleranceShape is Cylinder.
Parameter: double SA_DefaultsToleranceShapeCylinderLimitRadius	SA_DefaultsToler- anceShapeCylinderLimitRadius
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Default radius for shape Cylinder. Only available if SA_Default- sToleranceShape is Cylinder.
Parameter: double SA_DefaultsToleranceShapeBoxLimitHalfEdge1..3	SA_DefaultsToler- anceShapeBoxLimitHalfEdge1..3
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Default half egdes for shape Box. Only available if SA_Default- sToleranceShape is Box.

16.2.2.4 Get_DefaultsList (GET /api/defaults)

Get list of default settings for special sensor actions.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SA_DefaultEntries	SA_DefaultEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description:	Number of entries in the defaults array. All following parameters exists from 1 to this number, e.g. SA_Default1UUID, SA_Default2UUID, ...
Parameter: String SA_Default1..xUUID	SA_Default1..xUUID
Direction: Up	
Description:	Unique ID of the default setting.
Parameter: String SA_Default1..xObjectType	SA_Default1..xObjectType
Direction: Up	
Description:	Name of the object.

Parameter: String SA_Default1..xKey	SA_Default1..xKey
Direction: Up	
Description: Name of the attribute.	
Parameter: String SA_Default1..xValue	SA_Default1..xValue
Direction: Up	
Description: Value as number, boolean, text or object in JSON notation.	
Parameter: int32_t SA_FactoryDefaultEntries	SA_FactoryDefaultEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of entries in the factory defaults array. All following parameters exists from 1 to this number, e.g. SA_FactoryDefault1UUID, SA_FactoryDefault2UUID, ...	
Parameter: String SA_FactoryDefault1..xUUID	SA_FactoryDefault1..xUUID
Direction: Up	
Description: Unique ID of the default setting.	
Parameter: String SA_FactoryDefault1..xObjectType	SA_FactoryDe- fault1..xObjectType
Direction: Up	
Description: Name of the object.	
Parameter: String SA_FactoryDefault1..xKey	SA_FactoryDefault1..xKey
Direction: Up	
Description: Name of the attribute.	
Parameter: String SA_FactoryDefault1..xValue	SA_FactoryDe- fault1..xValue
Direction: Up	
Description: Value as number, boolean, text or object in JSON notation.	

16.2.2.5 Delete_Defaults (DELETE /api/defaults/{itemId})

Delete a default setting for special sensor actions.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_DefaultUUID	SP_DefaultUUID
Direction: Down	
Description: Unique ID of the default setting.	

16.2.2.6 Delete_DefaultsList (DELETE /api/defaults)

Delete default settings for special sensor actions.

Valid for sensor:

CFO100 / CFO200 over Ethernet

16.2.3 Device

16.2.3.1 Get_Device (GET /api/device, device show)

Get information on sensor itself.

Parameter: int32_t SP_DeviceProperty	SP_DeviceProperty
Direction: Down	
Valid values:	
0= All	
1= Device ID	
2= Model Key	
3= Model Name	
4= Device Variant	
5= Vendor Key	
6= Vendor Name	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Default:	<not specified>, do not send this parameter to sensor (means All).
Description:	Name of device property.
Parameter: String SA_DeviceID	SA_DeviceID
Direction: Up	
Description:	Serial number.
Parameter: String SA_DeviceModelName	SA_DeviceModelName
Direction: Up	
Valid values:	
CFO100 at CFO200, CFO200 at CFO200	
Description:	Readable name of sensor model.
Parameter: String SA_DeviceModelKey	SA_DeviceModelKey
Direction: Up	
Valid values:	
me_cfo_100 at CFO100, me_cfo_200 at CFO200	
Description:	Unique key of sensor model.
Parameter: String SA_DeviceVariant	SA_DeviceVariant
Direction: Up	
Description:	Sensor option. This parameter is only available if returned from sensor.
Parameter: String SA_DeviceVendor	SA_DeviceVendor
Direction: Up	
Valid values:	
Micro-Epsilon Eltrotec GmbH	
Description:	Name of sensor manufacturer. Deprecated since firmware version 1.4.13 over Ethernet.
Parameter: String SA_DeviceVendorName	SA_DeviceVendorName
Direction: Up	
Valid values:	
Micro-Epsilon Eltrotec GmbH	
Description:	Name of sensor manufacturer. New at firmware version 1.4.13 over Ethernet.

Parameter: String SA_DeviceVendorKey SA_DeviceVendorKey
Direction: Up
Valid values:
 eltrotec
Description: Key of sensor manufacturer. New at firmware version 1.4.13 over Ethernet.

16.2.4 Firmware

16.2.4.1 Get_CurrentFirmware (GET /api/firmware, firmware version)

Get information about running firmware.

Parameter: String SA_CurrentFirmwareID SA_CurrentFirmwareID
Direction: Up
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Description: Unique ID for of this particular firmware image.

Parameter: String SA_CurrentFirmwareVersion SA_CurrentFirmwareVersion
Direction: Up
Description: Version of a firmware.

Parameter: String SA_CurrentFirmwareCreatedOn SA_CurrentFirmwareCreate-
dOn
Direction: Up
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Description: Time this firmware build was created.

Parameter: String SA_CurrentFirmwareWorksWith SA_CurrentFirmwareWork-
sWith
Direction: Up
Valid values:
 me_cfo_100 for CFO100, me_cfo_200 for CFO200
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Description: Newline separated array of valid sensor model keys.

Parameter: String SA_CurrentFirmwareName SA_CurrentFirmwareName
Direction: Up
Valid values:
 CFO
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Description: Human-readable name of this release.

Parameter: String SA_CurrentFirmwareChannel SA_CurrentFirmwareChannel
Direction: Up
Valid values:
 stable
 feature
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Description: Describes the kind of a publication.

Parameter: String SA_CurrentFirmwareNotes SA_CurrentFirmwareNotes
Direction: Up
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Description: Release notes formatted as markdown.

16.2.4.2 Get_RecoveryFirmware (GET /api/firmware/recovery, firmware recovery show)

Get information about recovery firmware.

Parameter: int32_t SP_RecoveryFirmwareProperty SP_RecoveryFirmwareProperty
Direction: Down
Valid values:
 0= All
 1= Channel
 2= Created on
 3= ID
 4= Name
 5= Version
Valid for sensor:
 CFO100 / CFO200 over RS232
Default: <not specified>, do not send this parameter to sensor (means All).
Description: Name of recovery firmware property.

Parameter: String SA_RecoveryFirmwareID SA_RecoveryFirmwareID
Direction: Up
Description: Unique ID for of this particular firmware image.

Parameter: String SA_RecoveryFirmwareVersion SA_RecoveryFirmwareVersion
Direction: Up
Description: Version of a firmware.

Parameter: String SA_RecoveryFirmwareCreatedOn SA_RecoveryFirmwareCreatedOn
Direction: Up
Description: Time this firmware build was created.

Parameter: String SA_RecoveryFirmwareWorksWith SA_RecoveryFirmwareWorksWith
Direction: Up
Valid values:
 me_cfo_100 at CFO100, me_cfo_200 at CFO200
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Description: Newline separated array of valid sensor model keys.

Parameter: String SA_RecoveryFirmwareName SA_RecoveryFirmwareName
Direction: Up
Valid values:
 CFO
Description: Human-readable name of this release.

Parameter: String SA_RecoveryFirmwareChannel	SA_RecoveryFirmwareChan-
Direction: Up	nel
Valid values:	
stable	
feature	
Description:	Describes the kind of a publication.
Parameter: String SA_RecoveryFirmwareNotes	SA_RecoveryFirmwareNotes
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Release notes formatted as markdown.
16.2.4.3 Get_Firmware (GET /api/firmware/images/{itemId})	
>Returns a single FirmwareImageUpload.	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Parameter: String SP_FirmwareUUID	SP_FirmwareUUID
Direction: Down	
Description:	Unique ID of the firmware.
Parameter: String SA_FirmwareUUID	SA_FirmwareUUID
Direction: Up	
Description:	Unique ID for of this particular firmware image.
Parameter: String SA_FirmwareBuildID	SA_FirmwareBuildID
Direction: Up	
Description:	Build ID for of this particular firmware image.
Parameter: String SA_FirmwareStatus	SA_FirmwareStatus
Direction: Up	
Valid values:	
incomplete	
complete	
invalid_signature	
processing_failure	
malformed_content	
device_mismatch	
Description:	Current status of the firmware upload.
Parameter: int32_t SA_FirmwareUploadedSize	SA_FirmwareUploadedSize
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description:	Number of uploaded bytes.
Parameter: int32_t SA_FirmwareExpectedSize	SA_FirmwareExpectedSize
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2147483647 (INT32_MAX)	
Description:	Expected total number of bytes for the firmware image.

Parameter: int32_t SA_FirmwareMaxChunkSize SA_FirmwareMaxChunkSize
Direction: Up
Valid values:
 Minimum: 1
 Maximum: 2147483647 (INT32_MAX)
Description: Maximum size for a data chunk uploaded to the device.

16.2.4.4 Get_FirmwareList (GET /api/firmware/images)

Retrieves a list of available FirmwareImageUploads.

Valid for sensor:
 CFO100 / CFO200 over Ethernet

Parameter: int32_t SA_FirmwareEntries SA_FirmwareEntries
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 2147483647 (INT32_MAX)
Description: Number of entries in the firmware image array. All following parameters exists from 1 to this number, e.g. SA_Firmware1UUID, SA_Firmware2UUID, ...

Parameter: String SA_Firmware1..xUUID SA_Firmware1..xUUID
Direction: Up
Description: Unique ID for of this particular firmware image.

Parameter: String SA_Firmware1..xBuildID SA_Firmware1..xBuildID
Direction: Up
Description: Build ID for of this particular firmware image.

Parameter: String SA_Firmware1..xStatus SA_Firmware1..xStatus
Direction: Up
Valid values:
 incomplete
 complete
 invalid_signature
 processing_failure
 malformed_content
 device_mismatch
Description: Current status of the firmware upload.

Parameter: int32_t SA_Firmware1..xUploadedSize SA_-
Firmware1..xUploadedSize
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 2147483647 (INT32_MAX)
Description: Number of uploaded bytes.

Parameter: int32_t SA_Firmware1..xExpectedSize SA_-
Firmware1..xExpectedSize
Direction: Up
Valid values:
 Minimum: 1
 Maximum: 2147483647 (INT32_MAX)
Description: Expected total number of bytes for the firmware image.

Parameter: int32_t SA_Firmware1...xMaxChunkSize

SA_-

Direction: Up

Firmware1...xMaxChunkSize

Valid values:

Minimum: 1

Maximum: 2147483647 (INT32_MAX)

Description: Maximum size for a data chunk uploaded to the device.

16.2.4.5 Activate_Firmware (POST /api/firmware/images/{itemId}/apply)

Applies the firmware to the device thus overwriting the current system image followed by a reboot of the device.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_FirmwareUUID

SP_FirmwareUUID

Direction: Down

Description: Unique ID of the firmware.

16.2.4.6 Set_FirmwareAsRecovery (POST /api/firmware/recovery/upgrade-from-current, firmware recovery upgrade)

Replaces the stored recovery image with the current system firmware.

16.2.4.7 Set_FirmwareSettings (PUT /api/firmware/settings)

Set information about current firmware.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_FirmwareReleaseChannel

SP_FirmwareReleaseChannel

Direction: Down

Valid values:

stable

feature

Default: <not specified>, do not send this parameter to sensor.

Description: Type of publication.

Parameter: String SA_FirmwareReleaseChannel

SA_FirmwareReleaseChannel

Direction: Up

Valid values:

stable

feature

Description: Type of publication.

16.2.4.8 Get_FirmwareSettings (GET /api/firmware/settings)

Get information about current firmware.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SA_FirmwareReleaseChannel

SA_FirmwareReleaseChannel

Direction: Up

Valid values:

stable

feature

Description: Type of publication.

16.2.4.9 Get_FirmwareStatus (GET /api/firmware/status)

Returns information about the currently running firmware version.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SA_FirmwareSourceURL

SA_FirmwareSourceURL

Direction: Up

Description: Absolute base URL of a firmware repository delivering firmware images suitable for this device. This parameter is only available if returned from sensor.

Parameter: String SA_FirmwareBuildID

SA_FirmwareBuildID

Direction: Up

Description: Unique ID for of this particular firmware image.

Parameter: String SA_FirmwareVersion

SA_FirmwareVersion

Direction: Up

Description: Version of a firmware.

16.2.4.10 Delete_Firmware (DELETE /api/firmware/images/{itemId})

Deletes a single FirmwareImageUpload.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_FirmwareUUID

SP_FirmwareUUID

Direction: Down

Description: Unique ID of the firmware.

16.2.5 Network

16.2.5.1 Set_NetworkInterface (PUT /api/network/interfaces/{name}, network select INTERFACE set <ipv4|ipv6> <disabled|static|dhcp|auto>)

Set new contents of network interfaces.

Parameter: String SP_NetworkInterfaceName

SP_NetworkInterfaceName

Direction: Down

Valid values:

eth0

Description: Unique name of interface.

Parameter: int32_t SP_NetworkInterfaceIPv4ConfigEntries	SP_NetworkInter- faceIPv4ConfigEntries
Direction: Down	
Valid values:	
-1 = deletes IPv4 configuration	
0 = leaves the IPv4 configuration untouched	
1	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: 0	
Description: Number of configured IPv4 address entries in the network inter- face array.	
Parameter: int32_t SP_NetworkInterfaceIPv4Config1Method	SP_NetworkInter- faceIPv4Config1Method
Direction: Down	
Valid values:	
1 = Static	
2 = DHCP	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Method of configured IPv4 entry. This parameter is only used if SP_NetworkInterfaceIPv4ConfigEntries is 1.	
Parameter: String SP_NetworkInterfaceIPv4Config1Address	SP_NetworkInter- faceIPv4Config1Address
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Address of configured IPv4 entry, if method is static. This parameter is only used if SP_NetworkInterfaceIPv4ConfigEntries is 1.	
Parameter: String SP_NetworkInterfaceIPv4Config1Gateway	SP_NetworkInter- faceIPv4Config1Gateway
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: Gateway of configured IPv4 entry, if method is static. This parameter is only used if SP_NetworkInterfaceIPv4ConfigEntries is 1.	
Parameter: int32_t SP_NetworkInterfaceIPv6ConfigEntries	SP_NetworkInter- faceIPv6ConfigEntries
Direction: Down	
Valid values:	
-1 = deletes IPv6 configuration	
0 = leaves the IPv6 configuration untouched	
1	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: 0	
Description: Number of configured IPv6 address entries in the network inter- face array.	

Parameter: int32_t SP_NetworkInterfaceIPv6Config1Method	SP_NetworkInterfaceIPv6Config1Method
Direction: Down	
Valid values:	
1= Static	
2= DHCP	
3= Auto	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Method of configured IPv6 entry. This parameter is only used if SP_NetworkInterfaceIPv6ConfigEntries is 1.
Parameter: String SP_NetworkInterfaceIPv6Config1Address	SP_NetworkInterfaceIPv6Config1Address
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Address of configured IPv6 entry, if method is static. This parameter is only used if SP_NetworkInterfaceIPv6ConfigEntries is 1.
Parameter: String SP_NetworkInterfaceIPv6Config1Gateway	SP_NetworkInterfaceIPv6Config1Gateway
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default:	<not specified>, do not send this parameter to sensor.
Description:	Gateway of configured IPv6 entry, if method is static. This parameter is only used if SP_NetworkInterfaceIPv6ConfigEntries is 1.
Parameter: int32_t SP_NetworkInterfaceFamily	SP_NetworkInterfaceFamily
Direction: Down	
Valid values:	
0= IPv4	
1= IPv6	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Network address family.
Parameter: int32_t SP_NetworkInterfaceMethod	SP_NetworkInterfaceMethod
Direction: Down	
Valid values:	
0= Disabled	
1= Static	
2= DHCP	
3= Auto	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Method of configured entry.
Parameter: String SP_NetworkInterfaceAddress	SP_NetworkInterfaceAddress
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Address of configured entry, if method is static.

Parameter: String SP_NetworkInterfaceGateway	SP_NetworkInterfaceGate-
Direction: Down	way
Valid for sensor:	
CFO100 / CFO200 over RS232	
Default: <not specified>, do not send this parameter to sensor.	
Description: Gateway of configured entry, if method is static.	
Parameter: String SA_NetworkInterfaceName	SA_NetworkInterfaceName
Direction: Up	
Valid values:	
eth0	
Description: Unique name of interface.	
Parameter: String SA_NetworkInterfaceHardwareAddress	SA_NetworkInterfaceHard-
Direction: Up	wareAddress
Description: Hardware (MAC) address of network interface.	
Parameter: int32_t SA_NetworkInterfaceHasLink	SA_NetworkInterface-
Direction: Up	HasLink
Valid values:	
0= Off	
1= On	
Description: Current interface status (cable connected or not).	
Parameter: int32_t SA_NetworkInterfaceIPv4CurrentEntries	SA_NetworkInter-
Direction: Up	faceIPv4CurrentEntries
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of currently used IPv4 address entries in the network	
interface array. All following parameters exists from 1 to this num-	
ber, e.g. SA_NetworkInterface1IPv4Current1Address, SA_NetworkInter-	
face1IPv4Current2Address, ...	
Parameter: String SA_NetworkInterfaceIPv4Current1..xAddress	SA_NetworkInter-
Direction: Up	faceIPv4Current1..xAddress
Description: Address of currently used IPv4 entry.	
Parameter: int32_t SA_NetworkInterfaceIPv4ConfigEntries	SA_NetworkInter-
Direction: Up	faceIPv4ConfigEntries
Valid values:	
Minimum: 0	
Maximum: 1	
Description: Number of configured IPv4 address entries in the network	
interface array. All following parameters exists from 1 to this num-	
ber, e.g. SA_NetworkInterface1IPv4Config1Address, SA_NetworkInter-	
face1IPv4Config2Address, ...	
Parameter: int32_t SA_NetworkInterfaceIPv4Config1..xMethod	SA_NetworkInter-
Direction: Up	faceIPv4Config1..xMethod
Valid values:	
-1= Unknown parameter value from sensor	
1= Static	
2= DHCP	
Description: Method of configured IPv4 entry.	

Parameter: String SA_NetworkInterfaceIPv4Config1..xAddress	SA_NetworkInterfaceIPv4Config1..xAddress
Direction: Up	
Description: Address of configured IPv4 entry, if method is static	
Parameter: String SA_NetworkInterfaceIPv4Config1..xGateway	SA_NetworkInterfaceIPv4Config1..xGateway
Direction: Up	
Description: Gateway of configured IPv4 entry, if available.	
Parameter: int32_t SA_NetworkInterfaceIPv6CurrentEntries	SA_NetworkInterfaceIPv6CurrentEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of currently used IPv6 address entries in the network interface array. All following parameters exists from 1 to this number, e.g. SA_NetworkInterface1IPv6Current1Address, SA_NetworkInterface1IPv6Current2Address, ...	
Parameter: String SA_NetworkInterfaceIPv6Current1..xAddress	SA_NetworkInterfaceIPv6Current1..xAddress
Direction: Up	
Description: Address of currently used IPv6 entry.	
Parameter: int32_t SA_NetworkInterfaceIPv6ConfigEntries	SA_NetworkInterfaceIPv6ConfigEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1	
Description: Number of configured IPv6 address entries in the network interface array. All following parameters exists from 1 to this number, e.g. SA_NetworkInterface1IPv6Config1Address, SA_NetworkInterface1IPv6Config2Address, ...	
Parameter: int32_t SA_NetworkInterfaceIPv6Config1..xMethod	SA_NetworkInterfaceIPv6Config1..xMethod
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
1= Static	
2= DHCP	
3= Auto	
Description: Method of configured IPv6 entry.	
Parameter: String SA_NetworkInterfaceIPv6Config1..xAddress	SA_NetworkInterfaceIPv6Config1..xAddress
Direction: Up	
Description: Address of configured IPv6 entry, if method is static.	
Parameter: String SA_NetworkInterfaceIPv6Config1..xGateway	SA_NetworkInterfaceIPv6Config1..xGateway
Direction: Up	
Description: Gateway of configured IPv6 entry, if available.	

16.2.5.2 Get_NetworkInterface (GET /api/network/interfaces/{name}, network select INTERFACE show)

Get network interface based on unique name.

Parameter: String SP_NetworkInterfaceName

SP_NetworkInterfaceName

Direction: Down

Valid values:

eth0

Description: Unique name of interface.

Parameter: int32_t SP_NetworkInterfaceProperty

SP_NetworkInterfaceProperty

Direction: Down

Valid values:

0= All

1= IPv4 current addresses

2= IPv4 config

3= IPv6 current addresses

4= IPv6 config

5= Has link

6= Hardware Address

7= Name

Valid for sensor:

CFO100 / CFO200 over RS232

Default: <not specified>, do not send this parameter to sensor (means All).

Description: Name of interface property.

Parameter: String SA_NetworkInterfaceName

SA_NetworkInterfaceName

Direction: Up

Valid values:

eth0

Description: Unique name of interface. Only available over Ethernet or if SP_NetworkInterfaceProperty is all or name.

Parameter: String SA_NetworkInterfaceHardwareAddress

SA_NetworkInterfaceHardwareAddress

Direction: Up

Description: Hardware (MAC) address of network interface. Only available over Ethernet or if SP_NetworkInterfaceProperty is all or hardware address.

Parameter: int32_t SA_NetworkInterfaceHasLink

SA_NetworkInterfaceHasLink

Direction: Up

Valid values:

0= Off

1= On

Description: Current interface status (cable connected or not). Only available over Ethernet or if SP_NetworkInterfaceProperty is all or has link.

Parameter: int32_t SA_NetworkInterfaceIPv4CurrentEntries

SA_NetworkInterfaceIPv4CurrentEntries

Direction: Up

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Description: Number of currently used IPv4 address entries in the network interface array. Only available over Ethernet or if SP_NetworkInterfaceProperty is all or IPv4 current addresses. All following parameters exists from 1 to this number, e.g. SA_NetworkInterfaceIPv4Current1Address, SA_NetworkInterfaceIPv4Current2Address, ...

Parameter: String SA_NetworkInterfaceIPv4Current1..xAddress	SA_NetworkInterfaceIPv4Current1..xAddress
Direction: Up	
Description: Address of currently used IPv4 entry.	
Parameter: int32_t SA_NetworkInterfaceIPv4ConfigEntries	SA_NetworkInterfaceIPv4ConfigEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1	
Description: Number of configured IPv4 address entries in the network interface array. Only available over Ethernet or if SP_NetworkInterfaceProperty is all or IPv4 config All following parameters exists from 1 to this number, e.g. SA_NetworkInterfaceIPv4Config1Address, SA_NetworkInterfaceIPv4Config2Address, ...	
Parameter: int32_t SA_NetworkInterfaceIPv4Config1..xMethod	SA_NetworkInterfaceIPv4Config1..xMethod
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
1 = Static	
2 = DHCP	
Description: Method of configured IPv4 entry.	
Parameter: String SA_NetworkInterfaceIPv4Config1..xAddress	SA_NetworkInterfaceIPv4Config1..xAddress
Direction: Up	
Description: Address of configured IPv4 entry, if method is static	
Parameter: String SA_NetworkInterfaceIPv4Config1..xGateway	SA_NetworkInterfaceIPv4Config1..xGateway
Direction: Up	
Description: Gateway of configured IPv4 entry, if available.	
Parameter: int32_t SA_NetworkInterfaceIPv6CurrentEntries	SA_NetworkInterfaceIPv6CurrentEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of currently used IPv6 address entries in the network interface array. Only available over Ethernet or if SP_NetworkInterfaceProperty is all or IPv6 current addresses All following parameters exists from 1 to this number, e.g. SA_NetworkInterfaceIPv6Current1Address, SA_NetworkInterfaceIPv6Current2Address, ...	
Parameter: String SA_NetworkInterfaceIPv6Current1..xAddress	SA_NetworkInterfaceIPv6Current1..xAddress
Direction: Up	
Description: Address of currently used IPv6 entry.	
Parameter: int32_t SA_NetworkInterfaceIPv6ConfigEntries	SA_NetworkInterfaceIPv6ConfigEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1	
Description: Number of configured IPv6 address entries in the network interface array. Only available over Ethernet or if SP_NetworkInterfaceProperty is all or IPv6 config All following parameters exists from 1 to this number, e.g. SA_NetworkInterfaceIPv6Config1Address, SA_NetworkInterfaceIPv6Config2Address, ...	

Parameter: int32_t SA_NetworkInterfaceIPv6Config1..xMethod	SA_NetworkInterfaceIPv6Config1..xMethod
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
1 = Static	
2 = DHCP	
3 = Auto	
Description: Method of configured IPv6 entry.	
Parameter: String SA_NetworkInterfaceIPv6Config1..xAddress	SA_NetworkInterfaceIPv6Config1..xAddress
Direction: Up	
Description: Address of configured IPv6 entry, if method is static.	
Parameter: String SA_NetworkInterfaceIPv6Config1..xGateway	SA_NetworkInterfaceIPv6Config1..xGateway
Direction: Up	
Description: Gateway of configured IPv6 entry, if available.	

16.2.5.3 Get_NetworkInterfaces (GET /api/network/interfaces, network list)

Get status and collection of network interfaces available in system.

Parameter: int32_t SA_NetworkInterfaceEntries	SA_NetworkInterfaceEntries
Direction: Up	
Valid values:	
1	
Description: Number of entries in the network interface array. All following parameters exists from 1 to this number, e.g. SA_NetworkInterface1Name, SA_NetworkInterface2Name, ...	
Parameter: String SA_NetworkInterface1..xName	SA_NetworkInterface1..xName
Direction: Up	
Valid values:	
eth0	
Description: Unique name of interface.	
Parameter: String SA_NetworkInterface1..xHardwareAddress	SA_NetworkInterface1..xHardwareAddress
Direction: Up	
Description: Hardware (MAC) address of network interfaces.	
Parameter: int32_t SA_NetworkInterface1..xHasLink	SA_NetworkInterface1..xHasLink
Direction: Up	
Valid values:	
0 = Off	
1 = On	
Description: Current interface status (cable connected or not).	
Parameter: int32_t SA_NetworkInterface1..xIPv4CurrentEntries	SA_NetworkInterface1..xIPv4CurrentEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of currently used IPv4 address entries in the network interface array. All following parameters exists from 1 to this number, e.g. SA_NetworkInterface1IPv4Current1Address, SA_NetworkInterface1IPv4Current2Address, ...	

Parameter: String SA_NetworkInterface1..xIPv4Current1..yAddress	SA_NetworkInter- face1..xIPv4Current1..yAddress
Direction: Up	
Description: Address of currently used IPv4 entry.	
Parameter: int32_t SA_NetworkInterface1..xIPv4ConfigEntries	SA_NetworkInter- face1..xIPv4ConfigEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1	
Description: Number of configured IPv4 address entries in the network interface array. All following parameters exists from 1 to this number, e.g. SA_NetworkInterface1IPv4Config1Address, SA_NetworkInterface1IPv4Config2Address, ...	
Parameter: int32_t SA_NetworkInterface1..xIPv4Config1..yMethod	SA_NetworkInter- face1..xIPv4Config1..yMethod
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
1 = Static	
2 = DHCP	
Description: Method of configured IPv4 entry.	
Parameter: String SA_NetworkInterface1..xIPv4Config1..yAddress	SA_NetworkInter- face1..xIPv4Config1..yAddress
Direction: Up	
Description: Address of configured IPv4 entry, if method is static	
Parameter: String SA_NetworkInterface1..xIPv4Config1..yGateway	SA_NetworkInter- face1..xIPv4Config1..yGateway
Direction: Up	
Description: Gateway of configured IPv4 entry, if available.	
Parameter: int32_t SA_NetworkInterface1..xIPv6CurrentEntries	SA_NetworkInter- face1..xIPv6CurrentEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of currently used IPv6 address entries in the network interface array. All following parameters exists from 1 to this number, e.g. SA_NetworkInterface1IPv6Current1Address, SA_NetworkInterface1IPv6Current2Address, ...	
Parameter: String SA_NetworkInterface1..xIPv6Current1..yAddress	SA_NetworkInter- face1..xIPv6Current1..yAddress
Direction: Up	
Description: Address of currently used IPv6 entry.	
Parameter: int32_t SA_NetworkInterface1..xIPv6ConfigEntries	SA_NetworkInter- face1..xIPv6ConfigEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1	
Description: Number of configured IPv6 address entries in the network interface array. All following parameters exists from 1 to this number, e.g. SA_NetworkInterface1IPv6Config1Address, SA_NetworkInterface1IPv6Config2Address, ...	

Parameter: int32_t SA_NetworkInterface <code>1..x</code> IPv6Config <code>1..y</code> Method	SA_NetworkInterface <code>1..x</code> IPv6Config <code>1..y</code> Method
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
1 = Static	
2 = DHCP	
3 = Auto	
Description: Method of configured IPv6 entry.	
Parameter: String SA_NetworkInterface <code>1..x</code> IPv6Config <code>1..y</code> Address	SA_NetworkInterface <code>1..x</code> IPv6Config <code>1..y</code> Address
Direction: Up	
Description: Address of configured IPv6 entry, if method is static.	
Parameter: String SA_NetworkInterface <code>1..x</code> IPv6Config <code>1..y</code> Gateway	SA_NetworkInterface <code>1..x</code> IPv6Config <code>1..y</code> Gateway
Direction: Up	
Description: Gateway of configured IPv6 entry, if available.	

16.2.5.4 Delete_Network (DELETE /api/network, network reset)

Network configuration of sensor.

16.2.6 Peripherals

16.2.6.1 Set_Keypad (PUT /api/peripherals/keypad, keypad lock)

Set keypad state.

Parameter: int32_t SP_KeypadLocked	SP_KeypadLocked
Direction: Down	
Valid values:	
0 = Off	
1 = On	
Default: <not specified>, do not send this parameter to sensor (only at Ethernet).	
Description: Key lock status.	
Parameter: int32_t SA_KeypadLocked	SA_KeypadLocked
Direction: Up	
Valid values:	
0 = Off	
1 = On	
Description: Key lock status.	
Parameter: String SA_KeypadVisualizationURL	SA_KeypadVisualizationURL
Direction: Up	
Valid values:	
/media/keypad.svg	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: URL of vector graph for keyboard (for simulation).	

16.2.6.2 Get_Keypad (GET /api/peripherals/keypad, keypad show)

Get keypad state.

Parameter: int32_t SA_KeypadLocked

SA_KeypadLocked

Direction: Up

Valid values:

0 = Off

1 = On

Description: Key lock status.

Parameter: String SA_KeypadVisualizationURL

SA_KeypadVisualizationURL

Direction: Up

Valid values:

/media/keypad.svg

Valid for sensor:

CFO100 / CFO200 over Ethernet

Description: URL of vector graph for keyboard (for simulation).

16.2.6.3 Exec_KeyPress (POST /api/peripherals/keypad/inputs/{name}/{event})

Triggering specified input action (key actuation).

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_KeyName

SP_KeyName

Direction: Down

Valid values:

clear

intensity

options

teach

tolerance

Description: Key name.

Parameter: String SP_KeyAction

SP_KeyAction

Direction: Down

Valid values:

up

down

Description: Key action.

16.2.6.4 Get_KeypadEvents (GET /api/peripherals/keypad/events)

Get keypad events.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SA_KeypadEventEntries

SA_KeypadEventEntries

Direction: Up

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Description: Number of entries in the keypad event array. All following parameters exists from 1 to this number, e.g. SA_KeypadEvent1Source, SA_KeypadEvent2Source, ...

Parameter: String SA_KeypadEvent1..xSource	SA_KeypadEvent1..xSource
Direction: Up	
Valid values:	
inputs	
Description: Event source.	
Parameter: String SA_KeypadEvent1..xName	SA_KeypadEvent1..xName
Direction: Up	
Valid values:	
clear	
intensity	
options	
teach	
tolerance	
Description: Event name.	
Parameter: String SA_KeypadEvent1..xEvent	SA_KeypadEvent1..xEvent
Direction: Up	
Valid values:	
up	
down	
Description: Event type.	
Parameter: int32_t SA_KeypadEvent1..xTimestamp	SA_KeypadEvent1..xTimestamp
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Unit: ms	
Description: Event timestamp.	

16.2.6.5 Get_KeypadInputs (GET /api/peripherals/keypad/inputs)

Determination of keypad interface inputs present.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SA_KeypadInputEntries	SA_KeypadInputEntries
Direction: Up	
Valid values:	
5	
Description: Number of entries in the keypad input array. All following parameters exists from 1 to this number, e.g. SA_KeypadInput1Name, SA_KeypadInput2Name, ...	
Parameter: String SA_KeypadInput1..xName	SA_KeypadInput1..xName
Direction: Up	
Valid values:	
clear	
intensity	
options	
teach	
tolerance	
Description: Input name.	

Parameter: int32_t SA_KeypadInput1..xCapabilityEntries	SA_KeypadIn-put1..xCapabilityEntries
Direction: Up	
Valid values:	
2	
Description: Number of entries in the capability array within keypad input.	
All following parameters exists from 1 to this number, e.g. SA_KeypadInput1Capability1Name, SA_KeypadInput1Capability2Name, ...	
Parameter: String SA_KeypadInput1..xCapability1..yName	SA_KeypadIn-put1..xCapability1..yName
Direction: Up	
Valid values:	
up	
down	
Description: Capability name.	
Parameter: String SA_KeypadInput1..xCapability1..yURL	SA_KeypadIn-put1..xCapability1..yURL
Direction: Up	
Description: URL of capability.	

16.2.6.6 Set_Outputs (PUT /api/peripherals/outputs)

Set digital output mode.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_OutputDriver	SP_OutputDriver
Direction: Down	
Valid values:	
off	
npn	
pnp	
push-pull	
Description: Mode for outputs. The list of available modes can be retrieved by Get_SensorCapabilities parameter SA_OutputDrivers	
Parameter: String SA_OutputDriver	SA_OutputDriver
Direction: Up	
Valid values:	
off	
npn	
pnp	
push-pull	
Description: Mode for outputs.	
Parameter: int32_t SA_OutputCount	SA_OutputCount
Direction: Up	
Valid values:	
3 at CFO100, 8 at CFO200	
Description: Number of available outputs.	

16.2.6.7 Get_Outputs (GET /api/peripherals/outputs)

Get information about digital outputs.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SA_OutputDriver

SA_OutputDriver

Direction: Up

Valid values:

- off
- npn
- pnp
- push-pull

Description: Mode for outputs.

Parameter: int32_t SA_OutputCount

SA_OutputCount

Direction: Up

Valid values:

- 3 at CFO100, 8 at CFO200

Description: Number of available outputs.

16.2.6.8 Set_RS232Settings (PUT /api/peripherals/rs232)

Modify interface configuration.

Valid for sensor:

CFO100 / CFO200 (Option 100) over Ethernet

Parameter: int32_t SP_RS232Protocol

SP_RS232Protocol

Direction: Down

Valid values:

- 0= None
- 1= Eliza
- 2= Modbus

Default: <not specified>, do not send this parameter to sensor.

Description: Serial protocol.

Parameter: int32_t SP_RS232SlaveID

SP_RS232SlaveID

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 247

Default: <not specified>, do not send this parameter to sensor.

Description: Modbus slave ID. 0 means no address, react on each packet.

Parameter: int32_t SP_RS232FrameFormat

SP_RS232FrameFormat

Direction: Down

Valid values:

- 0= RTU
- 1= ASCII

Default: <not specified>, do not send this parameter to sensor.

Description: Modbus frame format.

Parameter: int32_t SP_RS232Baudrate SP_RS232Baudrate
Direction: Down
Valid values:
 9600
 19200
 115200
Unit: Baud
Default: <not specified>, do not send this parameter to sensor.
Description: Serial baudrate.

16.2.6.9 Get_RS232Settings (GET /api/peripherals/rs232)

Get current interface configuration.

Valid for sensor:
 CFO100 / CFO200 (Option 100) over Ethernet

Parameter: int32_t SA_RS232Protocol SA_RS232Protocol
Direction: Up
Valid values:
 0= None
 1= Eliza
 2= Modbus
Description: Serial protocol.

Parameter: int32_t SA_RS232SlaveID SA_RS232SlaveID
Direction: Up
Valid values:
Minimum: 0
Maximum: 247
Description: Modbus slave ID. 0 means no address, react on each packet.

Parameter: int32_t SA_RS232FrameFormat SA_RS232FrameFormat
Direction: Up
Valid values:
 0= RTU
 1= ASCII
Description: Modbus frame format.

Parameter: int32_t SA_RS232Baudrate SA_RS232Baudrate
Direction: Up
Valid values:
 9600
 19200
 115200
Unit: Baud
Description: Serial baudrate.

16.2.6.10 Set_USBSettings (PUT /api/peripherals/usb)

Modify interface configuration.

Valid for sensor:

CFO100 / CFO200 (Option 100) over Ethernet

Parameter: int32_t SP_USBProtocol SP_USBProtocol

Direction: Down

Valid values:

0= None

1= Eliza

2= Modbus

Default: <not specified>, do not send this parameter to sensor.

Description: Serial protocol.

Parameter: int32_t SP_USBSlaveID SP_USBSlaveID

Direction: Down

Valid values:

Minimum: 0

Maximum: 247

Default: <not specified>, do not send this parameter to sensor.

Description: Modbus slave ID. 0 means no address, react on each packet.

Parameter: int32_t SP_USBFrameFormat SP_USBFrameFormat

Direction: Down

Valid values:

0= RTU

1= ASCII

Default: <not specified>, do not send this parameter to sensor.

Description: Modbus frame format.

16.2.6.11 Get_USBSettings (GET /api/peripherals/usb)

Get current interface configuration.

Valid for sensor:

CFO100 / CFO200 (Option 100) over Ethernet

Parameter: int32_t SA_USBProtocol SA_USBProtocol

Direction: Up

Valid values:

0= None

1= Eliza

2= Modbus

Description: Serial protocol.

Parameter: int32_t SA_USBSlaveID SA_USBSlaveID

Direction: Up

Valid values:

Minimum: 0

Maximum: 247

Description: Modbus slave ID. 0 means no address, react on each packet.

Parameter: int32_t SA_USBFrameFormat	SA_USBFrameFormat
Direction: Up	
Valid values:	
0= RTU	
1= ASCII	
Description: Modbus frame format.	
16.2.6.12 Get_TriggerSources (GET /api/peripherals/trigger-sources)	
The sensor has a number of input lines that can be used as trigger sources.	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Parameter: int32_t SA_TriggerEntries	SA_TriggerEntries
Direction: Up	
Valid values:	
1 at CFO100, 4 at CFO200	
Description: Number of entries in the trigger source array. All following parameters exists from 1 to this number, e.g. SA_Trigger1Source, SA_Trigger2Source, ...	
Parameter: int32_t SA_Trigger1..xSource	SA_Trigger1..xSource
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= TRG 0	
1= TRG 1 (only for CFO200)	
2= TRG 2 (only for CFO200)	
3= TRG 3 (only for CFO200)	
Description: Trigger input.	
Parameter: int32_t SA_Trigger1..xEventCntLevelHigh	SA_Trig- ger1..xEventCntLevelHigh
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Event count of this trigger source.	
Parameter: int32_t SA_Trigger1..xEventCntLevelLow	SA_Trig- ger1..xEventCntLevelLow
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Event count of this trigger source.	
Parameter: int32_t SA_Trigger1..xEventCntEdgeRising	SA_Trig- ger1..xEventCntEdgeRising
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Event count of this trigger source.	

Parameter: int32_t SA_Trigger1...xEventCntEdgeFalling

SA_Trig-

ger1...xEventCntEdgeFalling

Direction: Up

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Description: Event count of this trigger source.

16.2.6.13 Get_KeypadOutput (GET /api/peripherals/keypad/outputs/{name})

Determination of outputs for keypad interface.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_KeypadOutputName

SP_KeypadOutputName

Direction: Down

Valid values:

- button_clear
- button_intensity
- button_options
- button_teach
- button_tolerance
- hold_time
- keylock
- matching_box
- matching_cylinder
- matching_infinite
- matching_sphere
- multi_teach
- power
- trigger_output
- trigger_teach

Description: Name of output for keypad interface.

Parameter: String SA_KeypadOutputName

SA_KeypadOutputName

Direction: Up

Valid values:

- button_clear
- button_intensity
- button_options
- button_teach
- button_tolerance
- hold_time
- keylock
- matching_box
- matching_cylinder
- matching_infinite
- matching_sphere
- multi_teach
- power
- trigger_output
- trigger_teach

Description: Name of output for keypad interface.

Parameter: String SA_KeypadOutputType	SA_KeypadOutputType
Direction: Up	
Valid values:	
colored	
Description: Type of output for keypad interface.	
Parameter: int32_t SA_KeypadOutputAnimationEntries	SA_KeypadOutputAnimationEntries
Direction: Up	
Valid values:	
2	
Description: Number of entries in the animation array. All following parameters exists from 1 to this number, e.g. SA_KeypadOutputAnimation1Enabled, SA_KeypadOutputAnimation2Enabled, ...	
Parameter: int32_t SA_KeypadOutputAnimation1..xEnabled	SA_KeypadOutputAnimation1..xEnabled
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: State if animation is enabled for keypad output entry.	
Parameter: double SA_KeypadOutputAnimation1..xDuration	SA_KeypadOutputAnimation1..xDuration
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1.79769e+308 (DBL_MAX)	
Description: Duration of animation for keypad output entry.	
Parameter: String SA_KeypadOutputAnimation1..xColor	SA_KeypadOutputAnimation1..xColor
Direction: Up	
Valid values:	
green	
Description: Color of animation for keypad output entry. This parameter is only available if SA_KeypadOutputAnimation1..xEnabled is 1.	

16.2.6.14 Get_KeypadOutputList (GET /api/peripherals/keypad/outputs)

Get list of outputs for keypad interface

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SA_KeypadOutputEntries	SA_KeypadOutputEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of entries in the keypad output array. All following parameters exists from 1 to this number, e.g. SA_KeypadOutput1Name, SA_KeypadOutput2Name, ...	

Parameter: String SA_KeypadOutput1..xName	SA_KeypadOutput1..xName
Direction: Up	
Valid values:	
button_clear	
button_intensity	
button_options	
button_teach	
button_tolerance	
hold_time	
keylock	
matching_box	
matching_cylinder	
matching_infinite	
matching_sphere	
multi_teach	
power	
trigger_output	
trigger_teach	
Description: Name of output for keypad interface.	
Parameter: String SA_KeypadOutput1..xType	SA_KeypadOutput1..xType
Direction: Up	
Valid values:	
colored	
Description: Type of output for keypad interface.	
Parameter: int32_t SA_KeypadOutput1..xAnimationEntries	SA_KeypadOut- put1..xAnimationEntries
Direction: Up	
Valid values:	
2	
Description: Number of entries in the animation array. All following parameters exists from 1 to this number, e.g. SA_KeypadOutput1Animation1Enabled, SA_KeypadOutput1Animation2Enabled, ...	
Parameter: int32_t SA_KeypadOutput1..xAnimation1..yEnabled	SA_KeypadOut- put1..xAnimation1..yEnabled
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: State if animation is enabled for keypad output entry.	
Parameter: double SA_KeypadOutput1..xAnimation1..yDuration	SA_KeypadOut- put1..xAnimation1..yDuration
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1.79769e+308 (DBL_MAX)	
Description: Duration of animation for keypad output entry.	
Parameter: String SA_KeypadOutput1..xAnimation1..yColor	SA_KeypadOut- put1..xAnimation1..yColor
Direction: Up	
Valid values:	
green	
Description: Color of animation for keypad output entry. This parameter is only available if SA_KeypadOutput1..xAnimation1..yEnabled is 1.	

16.2.7 Sensor

16.2.7.1 Capabilities

16.2.7.1.1 Get_SensorCapabilities (GET /api/sensor/capabilities)

Get sensor capabilities.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SA_MaximumSamplerate

SA_MaximumSamplerate

Direction: Up

Valid values:

10000 at CFO100, 30000 at CFO200

Description: Highest sample rate supported by sensor.

Parameter: int32_t SA_MaximumMatchersCount

SA_MaximumMatchersCount

Direction: Up

Valid values:

255

Description: Maximum number of detection results (Matcher) be stored in a detection profile.

Parameter: int32_t SA_MaximumDetectablesCount

SA_MaximumDetectable-
sCount

Direction: Up

Valid values:

320

Description: Maximum number of color positions (Detectable) to be stored in a detection profile.

Parameter: int32_t SA_OutputPinCount

SA_OutputPinCount

Direction: Up

Valid values:

3 for CFO100, 8 for CFO200

Description: Number of sensor output pins.

Parameter: int32_t SA_ToleranceEntries

SA_ToleranceEntries

Direction: Up

Valid values:

4

Description: Number of entries in the tolerance array. All following parameters exists from 1 to this number, e.g. SA_Tolerance1Shape, SA_Tolerance2Shape, ...

Parameter: String SA_Tolerance1...xShape

SA_Tolerance1...xShape

Direction: Up

Valid values:

infinite

sphere

cylinder

box

Description: Name of tolerance shape.

Parameter: double SA_Tolerance _{1..x} LimitRadius	SA_Toler- ance _{1..x} LimitRadius
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Radius of tolerance shape shpere or cylinder. This parameter is only available for this shapes.	
Parameter: double SA_Tolerance _{1..x} LimitHalfHeight	SA_Toler- ance _{1..x} LimitHalfHeight
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Half height of tolerance shape cylinder. This parameter is only available for this shape.	
Parameter: double SA_Tolerance _{1..x} LimitHalfEdge _{1..3}	SA_Toler- ance _{1..x} LimitHalfEdge _{1..3}
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Half edges of tolerance shape box. This parameter is only available for this shape.	
Parameter: int32_t SA_TriggerEntries	SA_TriggerEntries
Direction: Up	
Valid values:	
1 at CFO100, 4 at CFO200	
Description: Number of entries in the trigger source array. All following parameters exists from 1 to this number, e.g. SA_Trigger1Source, SA_Trigger2Source, ...	
Parameter: int32_t SA_Trigger _{1..x} Source	SA_Trigger _{1..x} Source
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= TRG 0	
1= TRG 1 (only for CFO200)	
2= TRG 2 (only for CFO200)	
3= TRG 3 (only for CFO200)	
Description: Trigger input.	
Parameter: String SA_Trigger _{1..x} Events	SA_Trigger _{1..x} Events
Direction: Up	
Description: Newline separated array of event names of trigger sources.	
Parameter: String SA_SettingCategories	SA_SettingCategories
Direction: Up	
Description: Newline separated array of setting categories.	

Parameter: String SA_OutputDrivers	SA_OutputDrivers
Direction: Up	
Valid values:	
off	
nnp	
pnp	
push-pull	
Description: Newline separated array of output drivers.	
Parameter: int32_t SA_ActionEntries	SA_ActionEntries
Direction: Up	
Valid values:	
2	
6	
8	
Description: Number of entries in the actions array. Deprecated since firmware version 1.4.13. All following parameters exists from 1 to this number, e.g. SA_Action1Name, SA_Action2Name, ...	
Parameter: String SA_Action1..xName	SA_Action1..xName
Direction: Up	
Valid values:	
enable_switching_output	
teach_single	
keylock	
run_autogain	
remove_all_detectables	
remove_all_matchers	
api_view_lock (from firmware version 1.6)	
api_edit_lock (from firmware version 1.6)	
Description: Names of action. Deprecated since firmware version 1.4.13.	
Parameter: String SA_Action1..xArguments	SA_Action1..xArguments
Direction: Up	
Description: Array of possible arguments in JSON notation. Deprecated since firmware version 1.4.13.	
Parameter: int32_t SA_ColorSpaceEntries	SA_ColorSpaceEntries
Direction: Up	
Valid values:	
5	
Description: Number of entries in the colorspace array. All following parameters exists from 1 to this number, e.g. SA_ColorSpace1Name, SA_ColorSpace2Name, ...	
Parameter: String SA_ColorSpace1..xName	SA_ColorSpace1..xName
Direction: Up	
Valid values:	
L*a*b*	
L*u*v*	
L*u'v'	
xyY	
XYZ	
Description: Name of the color spaces.	

Parameter: int32_t SA_ColorSpace1..xID	SA_ColorSpace1..xID
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Lab	
1= Luv	
2= uvL	
3= xyY	
4= XYZ	
Description: ID of the color spaces.	
Parameter: String SA_ColorSpace1..xAxis1..3Label	SA_Col-
Direction: Up	orSpace1..xAxis1..3Label
Valid values:	
L*	
a*	
b*	
u*	
v*	
u'	
v'	
x	
y	
X	
Y	
Z	
Description: Label of the axes of the color spaces.	
Parameter: String SA_ColorSpace1..xAxis1..3ID	SA_Col-
Direction: Up	orSpace1..xAxis1..3ID
Valid values:	
L	
a	
b	
u	
v	
x	
y	
X	
Y	
Z	
Description: ID of the axes of the color spaces.	
Parameter: double SA_ColorSpace1..xAxis1..3Min	SA_Col-
Direction: Up	orSpace1..xAxis1..3Min
Valid values:	
Minimum: -500.0	
Maximum: 0.0	
Description: Minimum of the axes of the color spaces.	
Parameter: double SA_ColorSpace1..xAxis1..3Max	SA_Col-
Direction: Up	orSpace1..xAxis1..3Max
Valid values:	
Minimum: 1.0	
Maximum: 500.0	
Description: Maximum of the axes of the color spaces.	

Parameter: int32_t SA_ColorSpaceToleranceMapEntries	SA_ColorSpaceToleranceMapEntries
Direction: Up	
Valid values:	
10	
Description: Number of entries in the colorspace tolerance map. All following parameters exists from 1 to this number, e.g. SA_ColorSpaceToleranceMap1ID, SA_ColorSpaceToleranceMap2ID, ...	
Parameter: int32_t SA_ColorSpaceToleranceMap1..xID	SA_ColorSpaceToleranceMap1..xID
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Lab	
1= Luv	
2= uvL	
3= xyY	
4= XYZ	
Description: ID of the color spaces in the map.	
Parameter: String SA_ColorSpaceToleranceMap1..xShape	SA_ColorSpaceToleranceMap1..xShape
Direction: Up	
Valid values:	
cylinder	
box	
Description: Tolerance shape in the map.	
Parameter: String SA_ColorSpaceToleranceMap1..xLimitAxisRadius	SA_ColorSpaceToleranceMap1..xLimitAxisRadius
Direction: Up	
Description: Newline separated array of radius entries names of Tolerance shape cylinder in the map. This parameter is only available if SA_ColorSpaceToleranceMap1..xShape is cylinder.	
Parameter: String SA_ColorSpaceToleranceMap1..xLimitAxisHalfHeight	SA_ColorSpaceToleranceMap1..xLimitAxisHalfHeight
Direction: Up	
Description: Newline separated array of half height entries names of Tolerance shape cylinder in the map. This parameter is only available if SA_ColorSpaceToleranceMap1..xShape is cylinder.	
Parameter: String SA_ColorSpaceToleranceMap1..xLimitAxisHalfEdge	SA_ColorSpaceToleranceMap1..xLimitAxisHalfEdge
Direction: Up	
Description: Newline separated array of half edges entries names of Tolerance shape box in the map. This parameter is only available if SA_ColorSpaceToleranceMap1..xShape is box.	

16.2.7.2 Color spaces

16.2.7.2.1 Get_SensorColorSpace (GET /api/sensor/colorsplaces/{space_id}, sensor colorspace show)

Get information about a colorspace.

Parameter: int32_t SP_ColorSpaceID	SP_ColorSpaceID
Direction: Down	

Valid values:

0= Lab
 1= Luv
 2= uvL
 3= xyY
 4= XYZ

Valid for sensor:

CFO100 / CFO200 over Ethernet

Description: ID of the color space. Over RS232, the default colorspace of current detection profile.

Parameter: String SA_ColorSpaceName

SA_ColorSpaceName

Direction: Up

Valid values:

L*a*b*
 L*u*v*
 L*u'v'
 xyY
 XYZ

Description: Name of the color space.

Parameter: int32_t SA_ColorSpaceID

SA_ColorSpaceID

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
 0= Lab
 1= Luv
 2= uvL
 3= xyY
 4= XYZ

Description: ID of the color spaces.

Parameter: String SA_ColorSpaceAxis1..3Label

SA_Col-
orSpaceAxis1..3Label

Direction: Up

Valid values:

L*
 a*
 b*
 u*
 v*
 u'
 v'
 x
 y
 X
 Y
 Z

Description: Label of the axes of the color space.

Parameter: String SA_ColorSpaceAxis1..3ID

SA_ColorSpaceAxis1..3ID

Direction: Up

Valid values:

L
a

b
u
v
x
y
X
Y
Z

Description: ID of the axes of the color space.

Parameter: double SA_ColorSpaceAxis1..3Min

SA_ColorSpaceAxis1..3Min

Direction: Up

Valid values:

Minimum: -500.0
Maximum: 0.0

Description: Minimum of the axes of the color space.

Parameter: double SA_ColorSpaceAxis1..3Max

SA_ColorSpaceAxis1..3Max

Direction: Up

Valid values:

Minimum: 1.0
Maximum: 500.0

Description: Maximum of the axes of the color space.

16.2.7.2.2 Get_SensorColorSpaceList (GET /api/sensor/colorspheres, sensor colorspace list)

Get list of all colorspaces.

Parameter: int32_t SA_ColorSpaceEntries

SA_ColorSpaceEntries

Direction: Up

Valid values:

5

Description: Number of entries in the colorspace array. All following parameters exists from 1 to this number, e.g. SA_ColorSpace1Name, SA_ColorSpace2Name, ...

Parameter: String SA_ColorSpace1..xName

SA_ColorSpace1..xName

Direction: Up

Valid values:

L*a*b*
L*u*v*
L*u'v'
xyY
XYZ

Description: Name of the color spaces.

Parameter: int32_t SA_ColorSpace1..xID

SA_ColorSpace1..xID

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= Lab
1= Luv
2= uvL
3= xyY
4= XYZ

Description: ID of the color spaces.

Parameter: String SA_ColorSpace1..xAxis1..3Label	SA_Col- orSpace1..xAxis1..3Label
Direction: Up	
Valid values:	
L*	
a*	
b*	
u*	
v*	
u'	
v'	
x	
y	
X	
Y	
Z	
Description: Label of the axes of the color spaces.	
Parameter: String SA_ColorSpace1..xAxis1..3ID	SA_Col- orSpace1..xAxis1..3ID
Direction: Up	
Valid values:	
L	
a	
b	
u	
v	
x	
y	
X	
Y	
Z	
Description: ID of the axes of the color spaces.	
Parameter: double SA_ColorSpace1..xAxis1..3Min	SA_Col- orSpace1..xAxis1..3Min
Direction: Up	
Valid values:	
Minimum: -500.0	
Maximum: 0.0	
Description: Minimum of the axes of the color spaces.	
Parameter: double SA_ColorSpace1..xAxis1..3Max	SA_Col- orSpace1..xAxis1..3Max
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 500.0	
Description: Maximum of the axes of the color spaces.	

16.2.7.3 Detection profiles

16.2.7.3.1 Add_DetectionProfile (POST /api/sensor/detection-profiles)

Add a new detection profile.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_DetectionProfileName	SP_DetectionProfileName
Direction: Down	
Default: <not specified>, do not send this parameter to sensor.	
Description: Freely selectable profile name.	
Parameter: int32_t SP_DetectionProfileColorSpaceID	SP_DetectionProfileColorSpaceID
Direction: Down	
Valid values:	
0= Lab	
1= Luv	
2= uvL	
3= xyY	
4= XYZ	
Default: <not specified>, do not send this parameter to sensor.	
Description: ID of color space.	
Parameter: int32_t SP_DetectionProfileSamplingAverageCount	SP_DetectionProfileSamplingAverageCount
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 100000000	
Default: <not specified>, do not send this parameter to sensor.	
Description: Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.	
Parameter: double SP_DetectionProfileNonMatchingHoldTime	SP_DetectionProfileNonMatchingHoldTime
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Default: <not specified>, do not send this parameter to sensor.	
Description: Defined minimum duration of output signal in seconds.	
Parameter: int32_t SP_DetectionProfileNonMatchingOutputState	SP_DetectionProfileNonMatchingOutputState
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Default: <not specified>, do not send this parameter to sensor.	
Description: The combination of values describes a logical state of the switching outputs of the sensor.	
Parameter: String SA_DetectionProfileUUID	SA_DetectionProfileUUID
Direction: Up	
Description: Unique ID of detection profile.	
Parameter: String SA_DetectionProfileName	SA_DetectionProfileName
Direction: Up	
Description: Freely selectable profile name.	

Parameter: int32_t SA_DetectionProfileColorSpaceID	SA_DetectionProfileColorSpaceID
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Lab	
1= Luv	
2= uvL	
3= xyY	
4= XYZ	
Description: ID of color space.	
Parameter: String SA_DetectionProfileColorSpaceAxis1..3ID	SA_DetectionProfileColorSpaceAxis1..3ID
Direction: Up	
Valid values:	
L	
a	
b	
u	
v	
x	
y	
X	
Y	
Z	
Description: ID of axes for color space.	
Parameter: String SA_DetectionProfileColorSpaceAxis1..3Label	SA_DetectionProfileColorSpaceAxis1..3Label
Direction: Up	
Valid values:	
L*	
a*	
b*	
u*	
v*	
u'	
v'	
x	
y	
X	
Y	
Z	
Description: Label of axes for color space.	
Parameter: double SA_DetectionProfileColorSpaceAxis1..3Min	SA_DetectionProfileColorSpaceAxis1..3Min
Direction: Up	
Valid values:	
Minimum: -500.0	
Maximum: 0.0	
Description: Minimum value of axes for color space.	
Parameter: double SA_DetectionProfileColorSpaceAxis1..3Max	SA_DetectionProfileColorSpaceAxis1..3Max
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 500.0	
Description: Maximum value of axes for color space.	

Parameter: String SA_DetectionProfileColorSpaceName	SA_DetectionProfileColorSpaceName
Direction: Up	
Valid values:	
L*a*b*	
L*u*v*	
L*u'v'	
xyY	
XYZ	
Description: Name of color space.	
Parameter: int32_t SA_DetectionProfileWhiteReferenceEntries	SA_DetectionProfileWhiteReferenceEntries
Direction: Up	
Valid values:	
0	
3	
Description: Number of entries in the white reference array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfileWhiteReference1, SA_DetectionProfileWhiteReference2, ...	
Parameter: double SA_DetectionProfileWhiteReference1..x	SA_DetectionProfileWhiteReference1..x
Direction: Up	
Valid values:	
Minimum: -1.79769e+308 (-DBL_MAX)	
Maximum: 1.79769e+308 (DBL_MAX)	
Description: White reference value.	
Parameter: int32_t SA_DetectionProfileNormalizationConstantEntries	SA_DetectionProfileNormalizationConstantEntries
Direction: Up	
Valid values:	
0	
3	
Description: Number of entries in the normalization constant array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfileNormalizationConstant1, SA_DetectionProfileNormalizationConstant2, ...	
Parameter: double SA_DetectionProfileNormalizationConstant1..x	SA_DetectionProfileNormalizationConstant1..x
Direction: Up	
Valid values:	
Minimum: -1.79769e+308 (-DBL_MAX)	
Maximum: 1.79769e+308 (DBL_MAX)	
Description: Normalization constant value.	
Parameter: String SA_DetectionProfileCompensationSettings	SA_DetectionProfileCompensationSettings
Direction: Up	
Description: Object of compensation settings in JSON notation.	
Parameter: double SA_DetectionProfileSamplingLEDIntensity	SA_DetectionProfileSamplingLEDIntensity
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: Relative intensity of the internal emitter during the light phase.	

Parameter: double SA_DetectionProfileSamplingBaseSR	SA_DetectionProfileSamplingBaseSR
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The base sample rate determines the duration of a sampling period.	
Parameter: double SA_DetectionProfileSamplingEffectiveSR	SA_DetectionProfileSamplingEffectiveSR
Direction: Up	
Valid values:	
Minimum: 1.0e-8	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The effective sample rate is the numeric product of the base samplerate and the number of averages.	
Parameter: double SA_DetectionProfileSamplingMinDesiredSR	SA_DetectionProfileSamplingMinDesiredSR
Direction: Up	
Valid values:	
Minimum: 0.01	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: This informational value represents the sample rate that was requested during the most recent Autogain operation.	
Parameter: int32_t SA_DetectionProfileSamplingLightPhase	SA_DetectionProfileSamplingLightPhase
Direction: Up	
Valid values:	
0 = Off	
1 = On	
Description: Defines if the sensor should periodically activate the internal emitter for sampling.	
Parameter: int32_t SA_DetectionProfileSamplingDarkPhase	SA_DetectionProfileSamplingDarkPhase
Direction: Up	
Valid values:	
0 = Off	
1 = On	
Description: Defines if the sensor should periodically deactivate the internal emitter for sampling.	
Parameter: int32_t SA_DetectionProfileSamplingAverageCount	SA_DetectionProfileSamplingAverageCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100000000	
Description: Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.	

Parameter: int32_t SA_DetectionProfileSamplingAmplificationLevel	SA_DetectionProfileSamplingAmplificationLevel
Direction: Up	
Valid values:	
Minimum: -2147483648 (INT32_MIN)	
Maximum: 2147483647 (INT32_MAX)	
Description: The amplification level specifies the internal configuration of an amplifier.	
Parameter: String SA_DetectionProfileNonMatchingOutputUUID	SA_DetectionProfileNon-MatchingOutputUUID
Direction: Up	
Description: Logical treatment of outputs upon recognition failure.	
Parameter: double SA_DetectionProfileNonMatchingHoldTime	SA_DetectionProfileNon-MatchingHoldTime
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Description: Defined minimum duration of output signal in seconds.	
Parameter: int32_t SA_DetectionProfileNonMatchingOutputState	SA_DetectionProfileNon-MatchingOutputState
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Description: The combination of values describes a logical state of the switching outputs of the sensor.	

16.2.7.3.2 Set_DetectionProfile (PUT /api/sensor/detection-profiles/{itemId}, sensor colorspace set COLORSPACE)

Set a detection profile.

Parameter: String SP_DetectionProfileUUID	SP_DetectionProfileUUID
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: Unique ID of detection profile or "current".	
Parameter: String SP_DetectionProfileName	SP_DetectionProfileName
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Freely selectable profile name.	
Parameter: int32_t SP_DetectionProfileColorSpaceID	SP_DetectionProfileColorSpaceID
Direction: Down	
Valid values:	
0= Lab	
1= Luv	
2= uvL	
3= xyY	
4= XYZ	
Default: <not specified>, do not send this parameter to sensor (only over Ethernet).	
Description: ID of color space.	

Parameter: int32_t SP_DetectionProfileSamplingAverageCount	SP_DetectionProfileSamplingAverageCount
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 100000000	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.	
Parameter: double SP_DetectionProfileNonMatchingHoldTime	SP_DetectionProfileNonMatchingHoldTime
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: Defined minimum duration of output signal in seconds.	
Parameter: int32_t SP_DetectionProfileNonMatchingOutputState	SP_DetectionProfileNonMatchingOutputState
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: The combination of values describes a logical state of the switching outputs of the sensor.	
Parameter: String SA_DetectionProfileUUID	SA_DetectionProfileUUID
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Unique ID of detection profile.	
Parameter: String SA_DetectionProfileName	SA_DetectionProfileName
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Freely selectable profile name.	
Parameter: int32_t SA_DetectionProfileColorSpaceID	SA_DetectionProfileColorSpaceID
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Lab	
1 = Luv	
2 = uvL	
3 = xyY	
4 = XYZ	
Description: ID of color space.	

Parameter: String SA_DetectionProfileColorSpaceAxis1..3ID	SA_DetectionProfileColorSpaceAxis1..3ID
Direction: Up	
Valid values:	
L	
a	
b	
u	
v	
x	
y	
X	
Y	
Z	
Description: ID of axes for color space.	
Parameter: String SA_DetectionProfileColorSpaceAxis1..3Label	SA_DetectionProfileColorSpaceAxis1..3Label
Direction: Up	
Valid values:	
L*	
a*	
b*	
u*	
v*	
u'	
v'	
x	
y	
X	
Y	
Z	
Description: Label of axes for color space.	
Parameter: double SA_DetectionProfileColorSpaceAxis1..3Min	SA_DetectionProfileColorSpaceAxis1..3Min
Direction: Up	
Valid values:	
Minimum: -500.0	
Maximum: 0.0	
Description: Minimum value of axes for color space.	
Parameter: double SA_DetectionProfileColorSpaceAxis1..3Max	SA_DetectionProfileColorSpaceAxis1..3Max
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 500.0	
Description: Maximum value of axes for color space.	
Parameter: String SA_DetectionProfileColorSpaceName	SA_DetectionProfileColorSpaceName
Direction: Up	
Valid values:	
L*a*b*	
L*u*v*	
L*u'v'	
xyY	
XYZ	
Description: Name of color space.	

Parameter: int32_t SA_DetectionProfileWhiteReferenceEntries	SA_DetectionProfile- WhiteReferenceEntries
Direction: Up	
Valid values:	
0	
3	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Number of entries in the white reference array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfile-WhiteReference1, SA_DetectionProfileWhiteReference2, ...
Parameter: double SA_DetectionProfileWhiteReference1..x	SA_DetectionProfile- WhiteReference1..x
Direction: Up	
Valid values:	
Minimum: -1.79769e+308 (-DBL_MAX)	
Maximum: 1.79769e+308 (DBL_MAX)	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	White reference value.
Parameter: int32_t SA_DetectionProfileNormalizationConstantEntries	SA_DetectionProfileNor- malizationConstantEntries
Direction: Up	
Valid values:	
0	
3	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Number of entries in the normalization constant array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfileNormalizationConstant1, SA_DetectionProfileNormalizationConstant2, ...
Parameter: double SA_DetectionProfileNormalizationConstant1..x	SA_DetectionProfileNor- malizationConstant1..x
Direction: Up	
Valid values:	
Minimum: -1.79769e+308 (-DBL_MAX)	
Maximum: 1.79769e+308 (DBL_MAX)	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Normalization constant value.
Parameter: String SA_DetectionProfileCompensationSettings	SA_DetectionProfileCom- pensationSettings
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Object of compensation settings in JSON notation.
Parameter: double SA_DetectionProfileSamplingLEDIntensity	SA_DetectionProfileSam- plingLEDIntensity
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Relative intensity of the internal emitter during the light phase.

Parameter: double SA_DetectionProfileSamplingBaseSR	SA_DetectionProfileSamplingBaseSR
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	The base sample rate determines the duration of a sampling period.
Parameter: double SA_DetectionProfileSamplingEffectiveSR	SA_DetectionProfileSamplingEffectiveSR
Direction: Up	
Valid values:	
Minimum: 1.0e-8	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	The effective sample rate is the numeric product of the base samplerate and the number of averages.
Parameter: double SA_DetectionProfileSamplingMinDesiredSR	SA_DetectionProfileSamplingMinDesiredSR
Direction: Up	
Valid values:	
Minimum: 0.01	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	This informational value represents the sample rate that was requested during the most recent Autogain operation.
Parameter: int32_t SA_DetectionProfileSamplingLightPhase	SA_DetectionProfileSamplingLightPhase
Direction: Up	
Valid values:	
0= Off	
1= On	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Defines if the sensor should periodically activate the internal emitter for sampling.
Parameter: int32_t SA_DetectionProfileSamplingDarkPhase	SA_DetectionProfileSamplingDarkPhase
Direction: Up	
Valid values:	
0= Off	
1= On	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Defines if the sensor should periodically deactivate the internal emitter for sampling.

Parameter: int32_t SA_DetectionProfileSamplingAverageCount	SA_DetectionProfileSamplingAverageCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100000000	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.
Parameter: int32_t SA_DetectionProfileSamplingAmplificationLevel	SA_DetectionProfileSamplingAmplificationLevel
Direction: Up	
Valid values:	
Minimum: -2147483648 (INT32_MIN)	
Maximum: 2147483647 (INT32_MAX)	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	The amplification level specifies the internal configuration of an amplifier.
Parameter: String SA_DetectionProfileNonMatchingOutputUUID	SA_DetectionProfileNon-MatchingOutputUUID
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Logical treatment of outputs upon recognition failure.
Parameter: double SA_DetectionProfileNonMatchingHoldTime	SA_DetectionProfileNon-MatchingHoldTime
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Defined minimum duration of output signal in seconds.
Parameter: int32_t SA_DetectionProfileNonMatchingOutputState	SA_DetectionProfileNon-MatchingOutputState
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	The combination of values describes a logical state of the switching outputs of the sensor.

16.2.7.3.3 Get_DetectionProfile (GET /api/sensor/detection-profiles/{itemId})

Get a detection profile.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_DetectionProfileUUID

SP_DetectionProfileUUID

Direction: Down

Description: Unique ID of detection profile or "current".

Parameter: String SA_DetectionProfileUUID	SA_DetectionProfileUUID
Direction: Up	
Description: Unique ID of detection profile.	
Parameter: String SA_DetectionProfileName	SA_DetectionProfileName
Direction: Up	
Description: Freely selectable profile name.	
Parameter: int32_t SA_DetectionProfileColorSpaceID	SA_DetectionProfileColorSpaceID
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Lab	
1 = Luv	
2 = uvL	
3 = xyY	
4 = XYZ	
Description: ID of color space.	
Parameter: String SA_DetectionProfileColorSpaceAxis1..3ID	SA_DetectionProfileColorSpaceAxis1..3ID
Direction: Up	
Valid values:	
L	
a	
b	
u	
v	
x	
y	
X	
Y	
Z	
Description: ID of axes for color space.	
Parameter: String SA_DetectionProfileColorSpaceAxis1..3Label	SA_DetectionProfileColorSpaceAxis1..3Label
Direction: Up	
Valid values:	
L*	
a*	
b*	
u*	
v*	
u'	
v'	
x	
y	
X	
Y	
Z	
Description: Label of axes for color space.	

Parameter: double SA_DetectionProfileColorSpaceAxis1..3Min	SA_DetectionProfileColorSpaceAxis1..3Min
Direction: Up	
Valid values:	
Minimum: -500.0	
Maximum: 0.0	
Description: Minimum value of axes for color space.	
Parameter: double SA_DetectionProfileColorSpaceAxis1..3Max	SA_DetectionProfileColorSpaceAxis1..3Max
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 500.0	
Description: Maximum value of axes for color space.	
Parameter: String SA_DetectionProfileColorSpaceName	SA_DetectionProfileColorSpaceName
Direction: Up	
Valid values:	
L*a*b*	
L*u*v*	
L*u'v'	
xyY	
XYZ	
Description: Name of color space.	
Parameter: int32_t SA_DetectionProfileWhiteReferenceEntries	SA_DetectionProfileWhiteReferenceEntries
Direction: Up	
Valid values:	
0	
3	
Description: Number of entries in the white reference array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfileWhiteReference1, SA_DetectionProfileWhiteReference2, ...	
Parameter: double SA_DetectionProfileWhiteReference1..x	SA_DetectionProfileWhiteReference1..x
Direction: Up	
Valid values:	
Minimum: -1.79769e+308 (-DBL_MAX)	
Maximum: 1.79769e+308 (DBL_MAX)	
Description: White reference value.	
Parameter: int32_t SA_DetectionProfileNormalizationConstantEntries	SA_DetectionProfileNormalizationConstantEntries
Direction: Up	
Valid values:	
0	
3	
Description: Number of entries in the normalization constant array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfileNormalizationConstant1, SA_DetectionProfileNormalizationConstant2, ...	
Parameter: double SA_DetectionProfileNormalizationConstant1..x	SA_DetectionProfileNormalizationConstant1..x
Direction: Up	
Valid values:	
Minimum: -1.79769e+308 (-DBL_MAX)	
Maximum: 1.79769e+308 (DBL_MAX)	
Description: Normalization constant value.	

Parameter: String SA_DetectionProfileCompensationSettings	SA_DetectionProfileCompensationSettings
Direction: Up	
Description: Object of compensation settings in JSON notation.	
Parameter: double SA_DetectionProfileSamplingLEDIntensity	SA_DetectionProfileSamplingLEDIntensity
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: Relative intensity of the internal emitter during the light phase.	
Parameter: double SA_DetectionProfileSamplingBaseSR	SA_DetectionProfileSamplingBaseSR
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The base sample rate determines the duration of a sampling period.	
Parameter: double SA_DetectionProfileSamplingEffectiveSR	SA_DetectionProfileSamplingEffectiveSR
Direction: Up	
Valid values:	
Minimum: 1.0e-8	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The effective sample rate is the numeric product of the base samplerate and the number of averages.	
Parameter: double SA_DetectionProfileSamplingMinDesiredSR	SA_DetectionProfileSamplingMinDesiredSR
Direction: Up	
Valid values:	
Minimum: 0.01	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: This informational value represents the sample rate that was requested during the most recent Autogain operation.	
Parameter: int32_t SA_DetectionProfileSamplingLightPhase	SA_DetectionProfileSamplingLightPhase
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Defines if the sensor should periodically activate the internal emitter for sampling.	
Parameter: int32_t SA_DetectionProfileSamplingDarkPhase	SA_DetectionProfileSamplingDarkPhase
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Defines if the sensor should periodically deactivate the internal emitter for sampling.	

Parameter: int32_t SA_DetectionProfileSamplingAverageCount	SA_DetectionProfileSamplingAverageCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100000000	
Description: Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.	
Parameter: int32_t SA_DetectionProfileSamplingAmplificationLevel	SA_DetectionProfileSamplingAmplificationLevel
Direction: Up	
Valid values:	
Minimum: -2147483648 (INT32_MIN)	
Maximum: 2147483647 (INT32_MAX)	
Description: The amplification level specifies the internal configuration of an amplifier.	
Parameter: String SA_DetectionProfileNonMatchingOutputUUID	SA_DetectionProfileNon-MatchingOutputUUID
Direction: Up	
Description: Logical treatment of outputs upon recognition failure.	
Parameter: double SA_DetectionProfileNonMatchingHoldTime	SA_DetectionProfileNon-MatchingHoldTime
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Description: Defined minimum duration of output signal in seconds.	
Parameter: int32_t SA_DetectionProfileNonMatchingOutputState	SA_DetectionProfileNon-MatchingOutputState
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Description: The combination of values describes a logical state of the switching outputs of the sensor.	

16.2.7.3.4 Get_DetectionProfileList (GET /api/sensor/detection-profiles)

Get collection of detection profiles present.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SA_CurrentProfileUUID	SA_CurrentProfileUUID
Direction: Up	
Description: Unique ID of currently used detection profile.	
Parameter: int32_t SA_DetectionProfileEntries	SA_DetectionProfileEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of entries in the detection profile array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfile1UUID, SA_DetectionProfile2UUID, ...	

Parameter: String SA_DetectionProfile1..xUUID	SA_DetectionPro- file1..xUUID
Direction: Up	
Description: Unique ID of detection profile.	
Parameter: String SA_DetectionProfile1..xName	SA_DetectionPro- file1..xName
Direction: Up	
Description: Freely selectable profile name.	
Parameter: int32_t SA_DetectionProfile1..xColorSpaceID	SA_DetectionPro- file1..xColorSpaceID
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Lab	
1 = Luv	
2 = uvL	
3 = xyY	
4 = XYZ	
Description: ID of color space.	
Parameter: String SA_DetectionProfile1..xColorSpaceAxis1..3ID	SA_DetectionPro- file1..xColorSpaceAxis1..3ID
Direction: Up	
Valid values:	
L	
a	
b	
u	
v	
x	
y	
X	
Y	
Z	
Description: ID of axes for color space.	
Parameter: String SA_DetectionProfile1..xColorSpaceAxis1..3Label	SA_DetectionPro- file1..xColorSpaceAxis1..3Label
Direction: Up	
Valid values:	
L*	
a*	
b*	
u*	
v*	
u'	
v'	
x	
y	
X	
Y	
Z	
Description: Label of axes for color space.	

Parameter: double SA_DetectionProfile1..xColorSpaceAxis1..3Min

SA_DetectionPro-
file1..xColorSpaceAxis1..3Min

Direction: Up

Valid values:

Minimum: -500.0

Maximum: 0.0

Description: Minimum value of axes for color space.

Parameter: double SA_DetectionProfile1..xColorSpaceAxis1..3Max

SA_DetectionPro-
file1..xColorSpaceAxis1..3Max

Direction: Up

Valid values:

Minimum: 1.0

Maximum: 500.0

Description: Maximum value of axes for color space.

Parameter: String SA_DetectionProfile1..xColorSpaceName

SA_DetectionPro-
file1..xColorSpaceName

Direction: Up

Valid values:

L*a*b*

L*u*v*

L*u'v'

xyY

XYZ

Description: Name of color space.

Parameter: int32_t SA_DetectionProfile1..xWhiteReferenceEntries

SA_DetectionPro-
file1..xWhiteReferenceEntries

Direction: Up

Valid values:

0

3

Description: Number of entries in the white reference array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfile1WhiteReference1, SA_DetectionProfile1WhiteReference2, ...

Parameter: double SA_DetectionProfile1..xWhiteReference1..y

SA_DetectionPro-
file1..xWhiteReference1..y

Direction: Up

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Description: White reference value.

Parameter: int32_t SA_DetectionProfile1..xNormalizationConstantEntries

SA_DetectionPro-
file1..xNormalizationConstantEntries

Direction: Up

Valid values:

0

3

Description: Number of entries in the normalization constant array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfile1NormalizationConstant1, SA_DetectionProfile1NormalizationConstant2, ...

Parameter: double SA_DetectionProfile1..xNormalizationConstant1..y

SA_DetectionPro-
file1..xNormalizationConstant1..y

Direction: Up

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Description: Normalization constant value.

Parameter: String SA_DetectionProfile1..xCompensationSettings	SA_DetectionPro- file1..xCompensationSettings
Direction: Up	
Description: Object of compensation settings in JSON notation.	
Parameter: double SA_DetectionProfile1..xSamplingLEDIntensity	SA_DetectionPro- file1..xSamplingLEDIntensity
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: Relative intensity of the internal emitter during the light phase.	
Parameter: double SA_DetectionProfile1..xSamplingBaseSR	SA_DetectionPro- file1..xSamplingBaseSR
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The base sample rate determines the duration of a sampling period.	
Parameter: double SA_DetectionProfile1..xSamplingEffectiveSR	SA_DetectionPro- file1..xSamplingEffectiveSR
Direction: Up	
Valid values:	
Minimum: 1.0e-8	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The effective sample rate is the numeric product of the base samplerate and the number of averages.	
Parameter: double SA_DetectionProfile1..xSamplingMinDesiredSR	SA_DetectionPro- file1..xSamplingMinDesiredSR
Direction: Up	
Valid values:	
Minimum: 0.01	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: This informational value represents the sample rate that was requested during the most recent Autogain operation.	
Parameter: int32_t SA_DetectionProfile1..xSamplingLightPhase	SA_DetectionPro- file1..xSamplingLightPhase
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Defines if the sensor should periodically activate the internal emitter for sampling.	
Parameter: int32_t SA_DetectionProfile1..xSamplingDarkPhase	SA_DetectionPro- file1..xSamplingDarkPhase
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Defines if the sensor should periodically deactivate the internal emitter for sampling.	

Parameter: int32_t SA_DetectionProfile1..xSamplingAverageCount	SA_DetectionPro- file1..xSamplingAverageCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100000000	
Description: Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.	
Parameter: int32_t SA_DetectionProfile1..xSamplingAmplificationLevel	SA_DetectionPro- file1..xSamplingAmplificationLevel
Direction: Up	
Valid values:	
Minimum: -2147483648 (INT32_MIN)	
Maximum: 2147483647 (INT32_MAX)	
Description: The amplification level specifies the internal configuration of an amplifier.	
Parameter: String SA_DetectionProfile1..xNonMatchingOutputUUID	SA_DetectionPro- file1..xNonMatchingOutputUUID
Direction: Up	
Description: Logical treatment of outputs upon recognition failure.	
Parameter: double SA_DetectionProfile1..xNonMatchingHoldTime	SA_DetectionPro- file1..xNonMatchingHoldTime
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Description: Defined minimum duration of output signal in seconds.	
Parameter: int32_t SA_DetectionProfile1..xNonMatchingOutputState	SA_DetectionPro- file1..xNonMatchingOutputState
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Description: The combination of values describes a logical state of the switching outputs of the sensor.	

16.2.7.3.5 Select_DetectionProfile (PUT /api/sensor/detection-profiles)

Select an existing detection profiles.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_CurrentProfileUUID	SP_CurrentProfileUUID
Direction: Down	
Default: <not specified>, do not send this parameter to sensor.	
Description: Unique ID of detection profile.	
Parameter: String SA_CurrentProfileUUID	SA_CurrentProfileUUID
Direction: Up	
Description: Unique ID of currently used detection profile.	

Parameter: int32_t SA_DetectionProfileEntries	SA_DetectionProfileEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of entries in the detection profile array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfile1UUID, SA_DetectionProfile2UUID, ...	
Parameter: String SA_DetectionProfile1..xUUID	SA_DetectionProfile1..xUUID
Direction: Up	
Description: Unique ID of detection profile.	
Parameter: String SA_DetectionProfile1..xName	SA_DetectionProfile1..xName
Direction: Up	
Description: Freely selectable profile name.	
Parameter: int32_t SA_DetectionProfile1..xColorSpaceID	SA_DetectionProfile1..xColorSpaceID
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Lab	
1 = Luv	
2 = uvL	
3 = xyY	
4 = XYZ	
Description: ID of color space.	
Parameter: String SA_DetectionProfile1..xColorSpaceAxis1..3ID	SA_DetectionProfile1..xColorSpaceAxis1..3ID
Direction: Up	
Valid values:	
L	
a	
b	
u	
v	
x	
y	
X	
Y	
Z	
Description: ID of axes for color space.	
Parameter: String SA_DetectionProfile1..xColorSpaceAxis1..3Label	SA_DetectionProfile1..xColorSpaceAxis1..3Label
Direction: Up	
Valid values:	
L*	
a*	
b*	
u*	
v*	
u'	
v'	
x	

y
X
Y
Z

Description: Label of axes for color space.

Parameter: double SA_DetectionProfile1..xColorSpaceAxis1..3Min

Direction: Up

Valid values:

Minimum: -500.0
Maximum: 0.0

Description: Minimum value of axes for color space.

SA_DetectionPro-
file1..xColorSpaceAxis1..3Min

Parameter: double SA_DetectionProfile1..xColorSpaceAxis1..3Max

Direction: Up

Valid values:

Minimum: 1.0
Maximum: 500.0

Description: Maximum value of axes for color space.

SA_DetectionPro-
file1..xColorSpaceAxis1..3Max

Parameter: String SA_DetectionProfile1..xColorSpaceName

Direction: Up

Valid values:

L*a*b*
L*u*v*
L*u'v'
xyY
XYZ

Description: Name of color space.

SA_DetectionPro-
file1..xColorSpaceName

Parameter: int32_t SA_DetectionProfile1..xWhiteReferenceEntries

Direction: Up

Valid values:

0
3

Description: Number of entries in the white reference array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfile1WhiteReference1, SA_DetectionProfile1WhiteReference2, ...

SA_DetectionPro-
file1..xWhiteReferenceEntries

Parameter: double SA_DetectionProfile1..xWhiteReference1..y

Direction: Up

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)
Maximum: 1.79769e+308 (DBL_MAX)

Description: White reference value.

SA_DetectionPro-
file1..xWhiteReference1..y

Parameter: int32_t SA_DetectionProfile1..xNormalizationConstantEntries

Direction: Up

Valid values:

0
3

Description: Number of entries in the normalization constant array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfile1NormalizationConstant1, SA_DetectionProfile1NormalizationConstant2,

SA_DetectionPro-
file1..xNormalizationConstantEntries

...

Parameter: double SA_DetectionProfile1..xNormalizationConstant1..y	SA_DetectionPro- file1..xNormalizationConstant
Direction: Up	
Valid values:	
Minimum: -1.79769e+308 (-DBL_MAX)	
Maximum: 1.79769e+308 (DBL_MAX)	
Description: Normalization constant value.	
Parameter: String SA_DetectionProfile1..xCompensationSettings	SA_DetectionPro- file1..xCompensationSettings
Direction: Up	
Description: Object of compensation settings in JSON notation.	
Parameter: double SA_DetectionProfile1..xSamplingLEDIntensity	SA_DetectionPro- file1..xSamplingLEDIntensity
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: Relative intensity of the internal emitter during the light phase.	
Parameter: double SA_DetectionProfile1..xSamplingBaseSR	SA_DetectionPro- file1..xSamplingBaseSR
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The base sample rate determines the duration of a sampling period.	
Parameter: double SA_DetectionProfile1..xSamplingEffectiveSR	SA_DetectionPro- file1..xSamplingEffectiveSR
Direction: Up	
Valid values:	
Minimum: 1.0e-8	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The effective sample rate is the numeric product of the base samplerate and the number of averages.	
Parameter: double SA_DetectionProfile1..xSamplingMinDesiredSR	SA_DetectionPro- file1..xSamplingMinDesiredSR
Direction: Up	
Valid values:	
Minimum: 0.01	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: This informational value represents the sample rate that was requested during the most recent Autogain operation.	
Parameter: int32_t SA_DetectionProfile1..xSamplingLightPhase	SA_DetectionPro- file1..xSamplingLightPhase
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Defines if the sensor should periodically activate the internal emitter for sampling.	

Parameter: int32_t SA_DetectionProfile1..xSamplingDarkPhase	SA_DetectionPro- file1..xSamplingDarkPhase
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Defines if the sensor should periodically deactivate the internal emitter for sampling.	
Parameter: int32_t SA_DetectionProfile1..xSamplingAverageCount	SA_DetectionPro- file1..xSamplingAverageCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100000000	
Description: Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.	
Parameter: int32_t SA_DetectionProfile1..xSamplingAmplificationLevel	SA_DetectionPro- file1..xSamplingAmplificationLevel
Direction: Up	
Valid values:	
Minimum: -2147483648 (INT32_MIN)	
Maximum: 2147483647 (INT32_MAX)	
Description: The amplification level specifies the internal configuration of an amplifier.	
Parameter: String SA_DetectionProfile1..xNonMatchingOutputUUID	SA_DetectionPro- file1..xNonMatchingOutputUUID
Direction: Up	
Description: Logical treatment of outputs upon recognition failure.	
Parameter: double SA_DetectionProfile1..xNonMatchingHoldTime	SA_DetectionPro- file1..xNonMatchingHoldTime
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Description: Defined minimum duration of output signal in seconds.	
Parameter: int32_t SA_DetectionProfile1..xNonMatchingOutputState	SA_DetectionPro- file1..xNonMatchingOutputState
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Description: The combination of values describes a logical state of the switching outputs of the sensor.	

16.2.7.3.6 Exec_DetectionProfileAutoGain (POST /api/sensor/detection-profiles/{itemId}/autogain, sensor autogain)

Triggering of automatic level adjustment.

Parameter: String SP_DetectionProfileUUID	SP_DetectionProfileUUID
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Unique ID of detection profile or "current".	

Parameter: double SP_DetectionProfileLevel	SP_DetectionProfileLevel
Direction: Down	
Valid values:	
Minimum: 0.01	
Maximum: 1.0	
Default: <not specified>, do not send this parameter to sensor.	
Description: Target value for level adjustment. Over RS232 this parameter is only send if sampling rate is send, too.	
Parameter: double SP_DetectionProfileMinimumSamplingRate	SP_DetectionProfileMinimumSamplingRate
Direction: Down	
Valid values:	
Minimum: 0.01	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Default: <not specified>, do not send this parameter to sensor.	
Description: Desired sample rate.	
Parameter: int32_t SP_DetectionProfileEnableInternalEmitter	SP_DetectionProfileEnableInternalEmitter
Direction: Down	
Valid values:	
0= Off	
1= On	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: Activate or switch off internal light source.	
Parameter: int32_t SP_DetectionProfileEnableAmbientLightCompensation	SP_DetectionProfileEnableAmbientLightCompensation
Direction: Down	
Valid values:	
0= Off	
1= On	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: Controls ambient light compensation (only relevant if SP_DetectionProfileEnableInternalEmitter is 1).	
Parameter: String SA_DetectionProfileCompensationSettings	SA_DetectionProfileCompensationSettings
Direction: Up	
Description: Object of compensation settings in JSON notation.	
Parameter: double SA_DetectionProfileSamplingLEDIntensity	SA_DetectionProfileSamplingLEDIntensity
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: Relative intensity of the internal emitter during the light phase.	

Parameter: double SA_DetectionProfileSamplingBaseSR	SA_DetectionProfileSamplingBaseSR
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The base sample rate determines the duration of a sampling period.	
Parameter: double SA_DetectionProfileSamplingEffectiveSR	SA_DetectionProfileSamplingEffectiveSR
Direction: Up	
Valid values:	
Minimum: 1.0e-8	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The effective sample rate is the numeric product of the base samplerate and the number of averages.	
Parameter: double SA_DetectionProfileSamplingMinDesiredSR	SA_DetectionProfileSamplingMinDesiredSR
Direction: Up	
Valid values:	
Minimum: 0.01	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: This informational value represents the sample rate that was requested during the most recent Autogain operation.	
Parameter: int32_t SA_DetectionProfileSamplingLightPhase	SA_DetectionProfileSamplingLightPhase
Direction: Up	
Valid values:	
0 = Off	
1 = On	
Description: Defines if the sensor should periodically activate the internal emitter for sampling.	
Parameter: int32_t SA_DetectionProfileSamplingDarkPhase	SA_DetectionProfileSamplingDarkPhase
Direction: Up	
Valid values:	
0 = Off	
1 = On	
Description: Defines if the sensor should periodically deactivate the internal emitter for sampling.	
Parameter: int32_t SA_DetectionProfileSamplingAverageCount	SA_DetectionProfileSamplingAverageCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100000000	
Description: Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.	

Parameter: int32_t SA_DetectionProfileSamplingAmplificationLevel SA_DetectionProfileSamplingAmplificationLevel
Direction: Up
Valid values:
Minimum: -2147483648 (INT32_MIN)
Maximum: 2147483647 (INT32_MAX)
Description: The amplification level specifies the internal configuration of an amplifier.

16.2.7.3.7 Exec_DetectionProfileWhiteRef (POST /api/sensor/detection-profiles/{itemId}/white-reference, sensor white-reference sample)

Set white reference (without parameters; current sample is used).

Parameter: String SP_DetectionProfileUUID SP_DetectionProfileUUID
Direction: Down
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Description: Unique ID of detection profile or "current".

Parameter: int32_t SA_DetectionProfileWhiteReferenceEntries SA_DetectionProfileWhiteReferenceEntries
Direction: Up
Valid values:
 0
 3
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Description: Number of entries in the white reference array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfileWhiteReference1, SA_DetectionProfileWhiteReference2, ...

Parameter: double SA_DetectionProfileWhiteReference1..x SA_DetectionProfileWhiteReference1..x
Direction: Up
Valid values:
Minimum: -1.79769e+308 (-DBL_MAX)
Maximum: 1.79769e+308 (DBL_MAX)
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Description: White reference value.

16.2.7.3.8 Get_DetectionProfileWhiteRef (GET /api/sensor/detection-profiles/{itemId}/white-reference)

Read out non-factory white reference.

Valid for sensor:
 CFO100 / CFO200 over Ethernet

Parameter: String SP_DetectionProfileUUID SP_DetectionProfileUUID
Direction: Down
Description: Unique ID of detection profile or "current".

Parameter: int32_t SA_DetectionProfileWhiteReferenceEntries	SA_DetectionProfile-WhiteReferenceEntries
Direction: Up	
Valid values:	
0	
3	
Description: Number of entries in the white reference array. All following parameters exists from 1 to this number, e.g. SA_DetectionProfile-WhiteReference1, SA_DetectionProfileWhiteReference2, ...	
Parameter: double SA_DetectionProfileWhiteReference1..x	SA_DetectionProfile-WhiteReference1..x
Direction: Up	
Valid values:	
Minimum: -1.79769e+308 (-DBL_MAX)	
Maximum: 1.79769e+308 (DBL_MAX)	
Description: White reference value.	

16.2.7.3.9 Delete_DetectionProfileWhiteRef (DELETE /api/sensor/detection-profiles/{itemId}/white-reference, sensor white-reference reset)

Delete white reference.

Parameter: String SP_DetectionProfileUUID	SP_DetectionProfileUUID
Direction: Down	
Description: Unique ID of detection profile or "current".	

16.2.7.3.10 Delete_DetectionProfile (DELETE /api/sensor/detection-profiles/{itemId})

Delete a detection profile.

Valid for sensor:	
CFO100 / CFO200 over Ethernet	

Parameter: String SP_DetectionProfileUUID	SP_DetectionProfileUUID
Direction: Down	
Description: Unique ID of detection profile or "current".	

16.2.7.3.11 Delete_DetectionProfileList (DELETE /api/sensor/detection-profiles)

Delete collection of detection profiles.

Valid for sensor:	
CFO100 / CFO200 over Ethernet	

16.2.7.4 Matchers

16.2.7.4.1 Add_SensorMatcher (POST /api/sensor/matchers, matcher add)

Add a recognition result.

Parameter: String SP_MatcherName	SP_MatcherName
Direction: Down	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: Freely selectable matcher name.	

Parameter: int32_t SP_MatcherToleranceShape	SP_MatcherToleranceShape
Direction: Down	
Valid values:	
0= Infinite	
1= Sphere	
2= Cylinder	
3= Box	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	<not specified>, do not send this parameter to sensor.
Description:	Distance model.
Parameter: double SP_MatcherToleranceShapeSphereLimitRadius	SP_MatcherToleranceShape- SphereLimitRadius
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Radius of distance model sphere. This parameter is only used if SP_MatcherToleranceShape is Sphere.
Parameter: double SP_MatcherToleranceShapeCylinderLimitHalfHeight	SP_MatcherTolerance- ShapeCylinderLim- itHalfHeight
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Half height of distance model cylinder. This parameter is only used if SP_MatcherToleranceShape is Cylinder.
Parameter: double SP_MatcherToleranceShapeCylinderLimitRadius	SP_MatcherTolerance- ShapeCylinderLimitRadius
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Radius of distance model cylinder. This parameter is only used if SP_MatcherToleranceShape is Cylinder.
Parameter: double SP_MatcherToleranceShapeBoxLimitHalfEdge1..3	SP_MatcherToleranceShape- BoxLimitHalfEdge1..3
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description:	Half edge of distance model box. This parameter is only used if SP_MatcherToleranceShape is Box.

Parameter: double SP_MatcherHoldTime	SP_MatcherHoldTime
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: Defined minimum duration of output signal in seconds.	
Parameter: int32_t SP_MatcherOutputPattern	SP_MatcherOutputPattern
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Default: <not specified>, do not send this parameter to sensor (over Ethernet). Over RS232 this parameter is obligatory.	
Description: The combination of values describes a logical state of the switching outputs of the sensor.	
Parameter: int32_t SP_MatcherResetOutputAfterHoldTimeExpired	SP_MatcherResetOutputAfterHoldTimeExpired
Direction: Down	
Valid values:	
0= Off	
1= On	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: Falling back of the output circuit after the holding time has expired with the triggered output circuit configured.	
Parameter: String SA_MatcherUUID	SA_MatcherUUID
Direction: Up	
Description: Unique ID of matcher.	
Parameter: String SA_MatcherName	SA_MatcherName
Direction: Up	
Description: Freely selectable matcher name.	
Parameter: int32_t SA_MatcherToleranceShape	SA_MatcherToleranceShape
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Infinite	
1= Sphere	
2= Cylinder	
3= Box	
Description: Distance model.	
Parameter: double SA_MatcherToleranceShapeSphereLimitRadius	SA_MatcherToleranceShapeSphereLimitRadius
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Radius of distance model sphere. This parameter is only available if SA_MatcherToleranceShape is Sphere.	

Parameter: double SA_MatcherToleranceShapeCylinderLimitHalfHeight	SA_MatcherTolerance-ShapeCylinderLimitHalfHeight
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Half height of distance model cylinder. This parameter is only available if SA_MatcherToleranceShape is Cylinder.	
Parameter: double SA_MatcherToleranceShapeCylinderLimitRadius	SA_MatcherTolerance-ShapeCylinderLimitRadius
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Radius of distance model cylinder. This parameter is only available if SA_MatcherToleranceShape is Cylinder.	
Parameter: double SA_MatcherToleranceShapeBoxLimitHalfEdge1..3	SA_MatcherToleranceShape-BoxLimitHalfEdge1..3
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Half edge of distance model box. This parameter is only available if SA_MatcherToleranceShape is Box.	
Parameter: double SA_MatcherHoldTime	SA_MatcherHoldTime
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Description: Defined minimum duration of output signal in seconds.	
Parameter: int32_t SA_MatcherOutputPattern	SA_MatcherOutputPattern
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Description: The combination of values describes a logical state of the switching outputs of the sensor.	
Parameter: int32_t SA_MatcherResetOutputAfterHoldTimeExpired	SA_MatcherResetOutputAfterHoldTimeExpired
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Falling back of the output circuit after the holding time has expired with the triggered output circuit configured.	

16.2.7.4.2 Set_SensorMatcher (PUT /api/sensor/matchers/{itemId}, matcher select MATCHER set <name|hold_time|output_pattern|tolerance>)

Set a recognition result.

Parameter: String SP_MatcherUUID	SP_MatcherUUID
Direction: Down	
Description: Unique ID of matcher.	

Parameter: int32_t SP_MatcherProperty	SP_MatcherProperty
Direction: Down	
Valid values:	
1= Name	
2= Hold time	
3= Output pattern	
4= Tolerance	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description:	Name of matcher property.
Parameter: String SP_MatcherName	SP_MatcherName
Direction: Down	
Default: <not specified>, do not send this parameter to sensor.	
Description:	Freely selectable matcher name. This parameter is only used over Ethernet or if SP_MatcherProperty is Name.
Parameter: int32_t SP_MatcherToleranceShape	SP_MatcherToleranceShape
Direction: Down	
Valid values:	
0= Infinite	
1= Sphere	
2= Cylinder	
3= Box	
Default: <not specified>, do not send this parameter to sensor. This parameter is only used over Ethernet or if SP_MatcherProperty is Tolerance.	
Description:	Distance model.
Parameter: double SP_MatcherToleranceShapeSphereLimitRadius	SP_MatcherToleranceShape- SphereLimitRadius
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description:	Radius of distance model sphere. This parameter is only used if SP_MatcherToleranceShape is Sphere.
Parameter: double SP_MatcherToleranceShapeCylinderLimitHalfHeight	SP_MatcherTolerance- ShapeCylinderLim- itHalfHeight
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description:	Half height of distance model cylinder. This parameter is only used if SP_MatcherToleranceShape is Cylinder.
Parameter: double SP_MatcherToleranceShapeCylinderLimitRadius	SP_MatcherTolerance- ShapeCylinderLimitRadius
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description:	Radius of distance model cylinder. This parameter is only used if SP_MatcherToleranceShape is Cylinder.

Parameter: double SP_MatcherToleranceShapeBoxLimitHalfEdge1..3	SP_MatcherToleranceShape-BoxLimitHalfEdge1..3
Direction: Down	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Half edge of distance model box. This parameter is only used if SP_MatcherToleranceShape is Box.	
Parameter: double SP_MatcherHoldTime	SP_MatcherHoldTime
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Default: <not specified>, do not send this parameter to sensor.	
Description: Defined minimum duration of output signal in seconds. This parameter is only used over Ethernet or if SP_MatcherProperty is Hold time.	
Parameter: int32_t SP_MatcherOutputPattern	SP_MatcherOutputPattern
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Default: <not specified>, do not send this parameter to sensor.	
Description: The combination of values describes a logical state of the switching outputs of the sensor. This parameter is only used over Ethernet or if SP_MatcherProperty is Output pattern.	
Parameter: int32_t SP_MatcherResetOutputAfterHoldTimeExpired	SP_MatcherResetOutputAfterHoldTimeExpired
Direction: Down	
Valid values:	
0= Off	
1= On	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: Falling back of the output circuit after the holding time has expired with the triggered output circuit configured.	
Parameter: String SA_MatcherUUID	SA_MatcherUUID
Direction: Up	
Description: Unique ID of matcher.	
Parameter: String SA_MatcherName	SA_MatcherName
Direction: Up	
Description: Freely selectable matcher name.	
Parameter: int32_t SA_MatcherToleranceShape	SA_MatcherToleranceShape
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Infinite	
1= Sphere	
2= Cylinder	
3= Box	
Description: Distance model.	

Parameter: double SA_MatcherToleranceShapeSphereLimitRadius	SA_MatcherToleranceShapeSphereLimitRadius
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Radius of distance model sphere. This parameter is only available if SA_MatcherToleranceShape is Sphere.	
Parameter: double SA_MatcherToleranceShapeCylinderLimitHalfHeight	SA_MatcherToleranceShapeCylinderLimitHalfHeight
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Half height of distance model cylinder. This parameter is only available if SA_MatcherToleranceShape is Cylinder.	
Parameter: double SA_MatcherToleranceShapeCylinderLimitRadius	SA_MatcherToleranceShapeCylinderLimitRadius
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Radius of distance model cylinder. This parameter is only available if SA_MatcherToleranceShape is Cylinder.	
Parameter: double SA_MatcherToleranceShapeBoxLimitHalfEdge1..3	SA_MatcherToleranceShapeBoxLimitHalfEdge1..3
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Half edge of distance model box. This parameter is only available if SA_MatcherToleranceShape is Box.	
Parameter: double SA_MatcherHoldTime	SA_MatcherHoldTime
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Description: Defined minimum duration of output signal in seconds.	
Parameter: int32_t SA_MatcherOutputPattern	SA_MatcherOutputPattern
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Description: The combination of values describes a logical state of the switching outputs of the sensor.	
Parameter: int32_t SA_MatcherResetOutputAfterHoldTimeExpired	SA_MatcherResetOutputAfterHoldTimeExpired
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Falling back of the output circuit after the holding time has expired with the triggered output circuit configured.	

16.2.7.4.3 Get_SensorMatcher (GET /api/sensor/matchers/{itemId}, matcher select MATCHER show)

Get a recognition result.

Parameter: String SP_MatcherUUID	SP_MatcherUUID
Direction: Down	
Description: Unique ID of matcher.	
Parameter: int32_t SP_MatcherProperty	SP_MatcherProperty
Direction: Down	
Valid values:	
0= All	
1= Name	
2= Hold time	
3= Output pattern	
4= Tolerance	
5= Number of Detectables	
6= Reset output after hold time expired	
7= UUID	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Default: <not specified>, do not send this parameter to sensor (means All).	
Description: Name of matcher property.	
Parameter: String SA_MatcherUUID	SA_MatcherUUID
Direction: Up	
Description: Unique ID of matcher. This parameter is only available over Ethernet or if SP_MatcherProperty is All or UUID.	
Parameter: String SA_MatcherName	SA_MatcherName
Direction: Up	
Description: Freely selectable matcher name. This parameter is only available over Ethernet or if SP_MatcherProperty is All or Name.	
Parameter: int32_t SA_MatcherToleranceShape	SA_MatcherToleranceShape
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Infinite	
1= Sphere	
2= Cylinder	
3= Box	
Description: Distance model. This parameter is only available over Ethernet or if SP_MatcherProperty is All or Tolerance.	
Parameter: double SA_MatcherToleranceShapeSphereLimitRadius	SA_MatcherToleranceShape-SphereLimitRadius
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Radius of distance model sphere. This parameter is only available if SA_MatcherToleranceShape is Sphere.	

Parameter: double SA_MatcherToleranceShapeCylinderLimitHalfHeight	SA_MatcherToleranceShapeCylinderLimitHalfHeight
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Half height of distance model cylinder. This parameter is only available if SA_MatcherToleranceShape is Cylinder.	
Parameter: double SA_MatcherToleranceShapeCylinderLimitRadius	SA_MatcherToleranceShapeCylinderLimitRadius
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Radius of distance model cylinder. This parameter is only available if SA_MatcherToleranceShape is Cylinder.	
Parameter: double SA_MatcherToleranceShapeBoxLimitHalfEdge1..3	SA_MatcherToleranceShapeBoxLimitHalfEdge1..3
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Half edge of distance model box. This parameter is only available if SA_MatcherToleranceShape is Box.	
Parameter: double SA_MatcherHoldTime	SA_MatcherHoldTime
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Description: Defined minimum duration of output signal in seconds. This parameter is only available over Ethernet or if SP_MatcherProperty is All or Hold time.	
Parameter: int32_t SA_MatcherOutputPattern	SA_MatcherOutputPattern
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Description: The combination of values describes a logical state of the switching outputs of the sensor. This parameter is only available over Ethernet or if SP_MatcherProperty is All or Output pattern.	
Parameter: int32_t SA_MatcherResetOutputAfterHoldTimeExpired	SA_MatcherResetOutputAfterHoldTimeExpired
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Falling back of the output circuit after the holding time has expired with the triggered output circuit configured. This parameter is only available over Ethernet or if SP_MatcherProperty is All or Reset output after hold time expired.	

Parameter: int32_t SA_MatcherDetectableCount SA_MatcherDetectableCount
Direction: Up
Valid values:
Minimum: 0
Maximum: 320
Valid for sensor:
 CFO100 / CFO200 over RS232
Description: Number of detectables. This parameter is only available if SP_MatcherProperty is Number of Detectables.

16.2.7.4.4 Get_SensorMatcherList (GET /api/sensor/matchers, matcher list)

Get collection of learned recognition results.

Parameter: String SP_MatcherDetectionProfileUUID SP_MatcherDetectionProfileUUID
Direction: Down
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Default: <not specified>, do not send this parameter to sensor.
Description: Unique ID of detection profile.

Parameter: int32_t SA_MatcherEntries SA_MatcherEntries
Direction: Up
Valid values:
Minimum: 0
Maximum: 255
Description: Number of entries in matcher array. All following parameters exists from 1 to this number, e.g. SA_Matcher1UUID, SA_Matcher2UUID, ...

Parameter: String SA_Matcher1..xUUID SA_Matcher1..xUUID
Direction: Up
Description: Unique ID of matcher.

Parameter: String SA_Matcher1..xName SA_Matcher1..xName
Direction: Up
Description: Freely selectable matcher name.

Parameter: int32_t SA_Matcher1..xToleranceShape SA_Matcher1..xToleranceShape
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = Infinite
 1 = Sphere
 2 = Cylinder
 3 = Box
Description: Distance model.

Parameter: double SA_Matcher1..xToleranceShapeSphereLimitRadius SA_Matcher1..xToleranceShapeSphereLimitRadius
Direction: Up
Valid values:
Minimum: 0.1
Maximum: 100.0
Description: Radius of distance model sphere. This parameter is only available if SA_Matcher1..xToleranceShape is Sphere.

Parameter: double SA_Matcher1..xToleranceShapeCylinderLimitHalfHeight	SA_- Matcher1..xToleranceShapeCylind...
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Half height of distance model cylinder. This parameter is only available if SA_Matcher1..xToleranceShape is Cylinder.	
Parameter: double SA_Matcher1..xToleranceShapeCylinderLimitRadius	SA_- Matcher1..xToleranceShapeCylind...
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Radius of distance model cylinder. This parameter is only available if SA_Matcher1..xToleranceShape is Cylinder.	
Parameter: double SA_Matcher1..xToleranceShapeBoxLimitHalfEdge1..3	SA_- Matcher1..xToleranceShapeBoxLi...
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Description: Half edge of distance model box. This parameter is only available if SA_Matcher1..xToleranceShape is Box.	
Parameter: double SA_Matcher1..xHoldTime	SA_Matcher1..xHoldTime
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3153600000.0	
Unit: s	
Description: Defined minimum duration of output signal in seconds.	
Parameter: int32_t SA_Matcher1..xOutputPattern	SA_- Matcher1..xOutputPattern
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Description: The combination of values describes a logical state of the switching outputs of the sensor.	
Parameter: int32_t SA_Matcher1..xResetOutputAfterHoldTimeExpired	SA_- Matcher1..xResetOutputAfterHo...
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Falling back of the output circuit after the holding time has expired with the triggered output circuit configured.	

16.2.7.4.5 Get_SensorMatcherIDList

Get assignment of matcher UUID's to MEDAQLib interal ID's (used at TransferData).

Parameter: String SA_MatcherIDList	SA_MatcherIDList
Direction: Up	
Description: List of matchers (separated by new line) in form of <ID>: <UUID>.	

16.2.7.4.6 Delete_SensorMatcher (DELETE /api/sensor/matchers/{itemId}, matcher select MATCHER remove)

Delete a recognition result.

Parameter: String SP_MatcherUUID SP_MatcherUUID
Direction: Down
Description: Unique ID of matcher.

Parameter: String SA_MatcherUUID SA_MatcherUUID
Direction: Up
Valid for sensor:
 CFO100 / CFO200 over RS232
Description: Unique ID of matcher.

16.2.7.4.7 Delete_SensorMatcherList (DELETE /api/sensor/matchers, matcher remove all)

Delete collection of learned recognition results.

16.2.7.5 Detectables

16.2.7.5.1 Learn_SensorDetectable (POST /api/sensor/detectables, matcher select MATCHER detectable add)

Add a new color using current detected values.

Parameter: String SP_SensorDetectableMatcherID SP_SensorDe-
tectableMatcherID
Direction: Down
Description: Reference to associated matcher object.

Parameter: String SA_SensorDetectableUUID SA_SensorDetectableUUID
Direction: Up
Description: Unique ID of the color.

Parameter: String SA_SensorDetectableMatcherID SA_SensorDe-
tectableMatcherID
Direction: Up
Description: Reference to associated matcher object.

Parameter: double SA_SensorDetectableColor1..3 SA_SensorDe-
tectable-
Color1..3
Direction: Up
Valid values:
Minimum: Typical L*a*b* (0.0/-500.0/-200.0), L*u*v* (0.0/0.0/0.0), L*u'v'
 (0.0/0.0/0.0), xyY (0.0/0.0/0.0), XYZ (0.0/0.0/0.0)
Maximum: Typical L*a*b* (100.0/500.0/200.0), L*u*v* (100.0/100.0/100.0),
 L*u'v' (1.0/1.0/1.0), xyY (1.0/1.0/100.0), XYZ (120.0/100.0/120.0)
Description: Color information for color coordinate.

Parameter: double SA_SensorDetectableRepresentationsRGB_{1..3}

Direction: Up

Valid values:

- Minimum:** 0.0
- Maximum:** 1.0

Description: RGB representation for color values.

SA_SensorDetectableRepresentationsRGB_{1..3}

16.2.7.5.2 Add_SensorDetectable (POST /api/sensor/detectables, matcher select MATCHER detectable add POSITION)

Add a new color using parameter values.

Parameter: String SP_SensorDetectableMatcherID

Direction: Down

Description: Reference to associated matcher object.

SP_SensorDetectableMatcherID

Parameter: double SP_SensorDetectableColor_{1..3}

Direction: Down

Valid values:

- Minimum:** Typical L*a*b* (0.0/-500.0/-200.0), L*u*v* (0.0/0.0/0.0), L*u'v'
(0.0/0.0/0.0), xyY (0.0/0.0/0.0), XYZ (0.0/0.0/0.0)
- Maximum:** Typical L*a*b* (100.0/500.0/200.0), L*u*v* (100.0/100.0/100.0),
L*u'v' (1.0/1.0/1.0), xyY (1.0/1.0/100.0), XYZ (120.0/100.0/120.0)

Default: <not specified>, do not send this parameter to sensor.

Description: Color information for color coordinate.

SP_SensorDetectableColor_{1..3}

Parameter: String SA_SensorDetectableUUID

Direction: Up

Description: Unique ID of the color.

SA_SensorDetectableUUID

Parameter: String SA_SensorDetectableMatcherID

Direction: Up

Description: Reference to associated matcher object.

SA_SensorDetectableMatcherID

Parameter: double SA_SensorDetectableColor_{1..3}

Direction: Up

Valid values:

- Minimum:** Typical L*a*b* (0.0/-500.0/-200.0), L*u*v* (0.0/0.0/0.0), L*u'v'
(0.0/0.0/0.0), xyY (0.0/0.0/0.0), XYZ (0.0/0.0/0.0)
- Maximum:** Typical L*a*b* (100.0/500.0/200.0), L*u*v* (100.0/100.0/100.0),
L*u'v' (1.0/1.0/1.0), xyY (1.0/1.0/100.0), XYZ (120.0/100.0/120.0)

Description: Color information for color coordinate.

SA_SensorDetectableColor_{1..3}

Parameter: double SA_SensorDetectableRepresentationsRGB_{1..3}

Direction: Up

Valid values:

- Minimum:** 0.0
- Maximum:** 1.0

Description: RGB representation for color values.

SA_SensorDetectableRepresentationsRGB_{1..3}

16.2.7.5.3 Get_SensorDetectableList (GET /api/sensor/detectables, matcher select MATCHER detectable list)

Get collection of learned color coordinates.

Parameter: String SP_SensorDetectableProfileUUID

Direction: Down

Valid for sensor:

CFO100 / CFO200 over Ethernet

Default: <not specified>, do not send this parameter to sensor.

Description: Unique ID of detection profile.

SP_SensorDetectableProfileUUID

Parameter: String SP_SensorDetectableMatcherUUID

Direction: Down

Default: <not specified>, do not send this parameter to sensor (over Ethernet). Over RS232 this parameter is obligatory.

Description: Unique ID of matcher.

SP_SensorDetectableMatcherUUID

Parameter: int32_t SA_SensorDetectableEntries

Direction: Up

Valid values:

Minimum: 0

Maximum: 320

Description: Number of entries in the sensor detectable array. All following parameters exists from 1 to this number, e.g. SA_SensorDetectable1UUID, SA_SensorDetectable2UUID, ...

SA_SensorDetectableEntries

Parameter: String SA_SensorDetectable1..xUUID

Direction: Up

Description: Unique ID of the color.

SA_SensorDetectable1..xUUID

Parameter: String SA_SensorDetectable1..xMatcherID

Direction: Up

Description: Reference to associated matcher object.

SA_SensorDetectable1..xMatcherID

Parameter: double SA_SensorDetectable1..xColor1..3

Direction: Up

Valid values:

Minimum: Typical L*a*b* (0.0/-500.0/-200.0), L*u*v* (0.0/0.0/0.0), L*u'v'
(0.0/0.0/0.0), xyY (0.0/0.0/0.0), XYZ (0.0/0.0/0.0)

Maximum: Typical L*a*b* (100.0/500.0/200.0), L*u*v* (100.0/100.0/100.0),
L*u'v' (1.0/1.0/1.0), xyY (1.0/1.0/100.0), XYZ (120.0/100.0/120.0)

Description: Color information for color coordinate.

SA_SensorDetectable1..xColor1..3

Parameter: double SA_SensorDetectable1..xRepresentationsRGB1..3

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 1.0

Description: RGB representation for color values.

SA_SensorDetectable1..xRepresentationsRGB1..3

16.2.7.5.4 Set_SensorDetectable (PUT /api/sensor/detectables/{itemId}, matcher select MATCHER detectable select DETECTABLE set position POSITION)

Update an existing color.

Parameter: String SP_SensorDetectableUUID	SP_SensorDetectableUUID
Direction: Down	
Description: Unique ID of the color.	
Parameter: String SP_SensorDetectableMatcherID	SP_SensorDe- tectableMatcherID
Direction: Down	
Default: <not specified>, do not send this parameter to sensor.	
Description: Reference to associated matcher object.	
Parameter: double SP_SensorDetectableColor1..3	SP_SensorDetectable- Color1..3
Direction: Down	
Valid values:	
Minimum: Typical L*a*b* (0.0/-500.0/-200.0), L*u*v* (0.0/0.0/0.0), L*u'v' (0.0/0.0/0.0), xyY (0.0/0.0/0.0), XYZ (0.0/0.0/0.0)	
Maximum: Typical L*a*b* (100.0/500.0/200.0), L*u*v* (100.0/100.0/100.0), L*u'v' (1.0/1.0/1.0), xyY (1.0/1.0/100.0), XYZ (120.0/100.0/120.0)	
Default: <not specified>, do not send this parameter to sensor.	
Description: Color information for color coordinate.	
Parameter: String SA_SensorDetectableUUID	SA_SensorDetectableUUID
Direction: Up	
Description: Unique ID of the color.	
Parameter: String SA_SensorDetectableMatcherID	SA_SensorDe- tectableMatcherID
Direction: Up	
Description: Reference to associated matcher object.	
Parameter: double SA_SensorDetectableColor1..3	SA_SensorDetectable- Color1..3
Direction: Up	
Valid values:	
Minimum: Typical L*a*b* (0.0/-500.0/-200.0), L*u*v* (0.0/0.0/0.0), L*u'v' (0.0/0.0/0.0), xyY (0.0/0.0/0.0), XYZ (0.0/0.0/0.0)	
Maximum: Typical L*a*b* (100.0/500.0/200.0), L*u*v* (100.0/100.0/100.0), L*u'v' (1.0/1.0/1.0), xyY (1.0/1.0/100.0), XYZ (120.0/100.0/120.0)	
Description: Color information for color coordinate.	
Parameter: double SA_SensorDetectableRepresentationsRGB1..3	SA_SensorDetectableRe- presentationsRGB1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: RGB representation for color values.	

16.2.7.5.5 Get_SensorDetectable (GET /api/sensor/detectables/{itemId}, matcher select MATCHER detectable select DETECTABLE show)

Get a learned color.

Parameter: String SP_SensorDetectableUUID	SP_SensorDetectableUUID
Direction: Down	
Description: Unique ID of the color.	

Parameter: String SP_SensorDetectableMatcherID	SP_SensorDe-
Direction: Down	tectableMatcherID
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description: Reference to associated matcher object.	
Parameter: int32_t SP_SensorDetectableProperty	SP_SensorDetectableProp-
Direction: Down	erty
Valid values:	
0= All	
1= Matcher	
2= Position	
3= RGB	
4= UUID	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Default: <not specified>, do not send this parameter to sensor (means All).	
Description: Name of detectable property.	
Parameter: String SA_SensorDetectableUUID	SA_SensorDetectableUUID
Direction: Up	
Description: Unique ID of the color. This parameter is only available over Ethernet or if SP_SensorDetectableProperty is All or UUID.	
Parameter: String SA_SensorDetectableMatcherID	SA_SensorDe-
Direction: Up	tectableMatcherID
Description: Reference to associated matcher object. This parameter is only available over Ethernet or if SP_SensorDetectableProperty is All or Matcher.	
Parameter: double SA_SensorDetectableColor1..3	SA_SensorDetectable-
Direction: Up	Color1..3
Valid values:	
Minimum: Typical L*a*b* (0.0/-500.0/-200.0), L*u*v* (0.0/0.0/0.0), L*u'v' (0.0/0.0/0.0), xyY (0.0/0.0/0.0), XYZ (0.0/0.0/0.0)	
Maximum: Typical L*a*b* (100.0/500.0/200.0), L*u*v* (100.0/100.0/100.0), L*u'v' (1.0/1.0/1.0), xyY (1.0/1.0/100.0), XYZ (120.0/100.0/120.0)	
Description: Color information for color coordinate. This parameter is only available over Ethernet or if SP_SensorDetectableProperty is All or Position.	
Parameter: double SA_SensorDetectableRepresentationsRGB1..3	SA_SensorDetectableRepre-
Direction: Up	sentationsRGB1..3
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: RGB representation for color values. This parameter is only available over Ethernet or if SP_SensorDetectableProperty is All or RGB.	

16.2.7.5.6 Delete_SensorDetectable (DELETE /api/sensor/detectables/{itemId}, matcher select MATCHER detectable select DETECTABLE remove)

Delete a specific color.

Parameter: String SP_SensorDetectableUUID	SP_SensorDetectableUUID
Direction: Down	
Description: Unique ID of the color.	

Parameter: String SP_SensorDetectableMatcherID	SP_SensorDe-
Direction: Down	tectableMatcherID
Valid for sensor:	
CFO100 / CFO200 over RS232	
Description: Reference to associated matcher object.	
16.2.7.5.7 Delete_SensorDetectableList (DELETE /api/sensor/detectables, matcher select MATCHER detectable remove all)	
Delete collection of learned color coordinates.	
Parameter: String SP_SensorDetectableProfileUUID	SP_SensorDetectablePro-
Direction: Down	fileUUID
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Default: <not specified>, do not send this parameter to sensor.	
Description: Unique ID of detection profile.	
Parameter: String SP_SensorDetectableMatcherID	SP_SensorDe-
Direction: Down	tectableMatcherID
Default: <not specified>, do not send this parameter to sensor (over Ether-	
net). Over RS232 this parameter is obligatory.	
Description: Unique ID of matcher.	
16.2.7.6 Action triggers	
16.2.7.6.1 Get_Action (GET /api/actions/{itemId})	
Retrieves an action.	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Parameter: String SP_ActionName	SP_ActionName
Direction: Down	
Valid values:	
enable_switching_output	
teach_single	
keylock	
run_autogain	
remove_all_detectables	
remove_all_matchers	
api_view_lock (from firmware version 1.6)	
api_edit_lock (from firmware version 1.6)	
Description: Names of action.	

Parameter: String SA_ActionName SA_ActionName

Direction: Up

Valid values:

- enable_switching_output
- teach_single
- keylock
- run_autogain
- remove_all_detectables
- remove_all_matchers
- api_view_lock (from firmware version 1.6)
- api_edit_lock (from firmware version 1.6)

Description: Names of action.

Parameter: String SA_ActionArguments SA_ActionArguments

Direction: Up

Description: Array of possible arguments in JSON notation.

16.2.7.6.2 Get_ActionList (GET /api/actions)

Retrieves a list of available actions.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SA_ActionEntries SA_ActionEntries

Direction: Up

Valid values:

- 2
- 6
- 8

Description: Number of entries in the actions array. All following parameters exists from 1 to this number, e.g. SA_Action1Name, SA_Action2Name, ...

Parameter: String SA_Action1..xName SA_Action1..xName

Direction: Up

Valid values:

- enable_switching_output
- teach_single
- keylock
- run_autogain
- remove_all_detectables
- remove_all_matchers
- api_view_lock (from firmware version 1.6)
- api_edit_lock (from firmware version 1.6)

Description: Names of action.

Parameter: String SA_Action1..xArguments SA_Action1..xArguments

Direction: Up

Description: Array of possible arguments in JSON notation.

16.2.7.6.3 Exec_Action (POST /api/actions/{itemId}/execute)

Executes an action.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_ActionName

SP_ActionName

Direction: Down

Valid values:

- enable_switching_output
- teach_single
- keylock
- run_autogain
- remove_all_detectables
- remove_all_matchers
- api_view_lock (from firmware version 1.6)
- api_edit_lock (from firmware version 1.6)

Description: Names of action.

Parameter: String SP_ActionMatcherID

SP_ActionMatcherID

Direction: Down

Default: <not specified>, do not send this parameter to sensor.

Description: Unique ID of matcher for detectable. This parameter is only used if SP_ActionName is 'teach_single'. If this parameter is not specified, the first matcher (i.e. the one with the output bitmask 0x01) is used.

Parameter: int32_t SP_ActionRemoveMatcherDetectablesBefore

SP_ActionRemoveMatcherDe-
tectablesBefore

Direction: Down

Valid values:

- 0= No
- 1= Yes

Default: <not specified>, do not send this parameter to sensor.

Description: Previous detectables of specified matcher are removed before. This parameter is only used if SP_ActionName is 'teach_single'.

Parameter: int32_t SP_ActionLockState

SP_ActionLockState

Direction: Down

Valid values:

- 0= No
- 1= Yes

Description: Keylock state. This parameter is only used if SP_ActionName is 'keylock'.

Parameter: int32_t SP_ActionLockViewState

SP_ActionLockViewState

Direction: Down

Valid values:

- 0= No
- 1= Yes

Description: API view lock state. This parameter is only used if SP_ActionName is 'api_view_lock'.

Parameter: int32_t SP_ActionLockEditState	SP_ActionLockEditState
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description: API edit lock state. This parameter is only used if SP_ActionName is 'api_edit_lock'.	
Parameter: String SA_ActionUUID	SA_ActionUUID
Direction: Up	
Description: Unique ID of current sample. This parameter is only available if SP_ActionName is 'enable_switching_output'.	
Parameter: double SA_ActionTimestamp	SA_ActionTimestamp
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 18446744073709.551616	
Unit: s	
Description: Time stamp (sensor run time) for color value. This parameter is only available if SP_ActionName is 'enable_switching_output'.	
Parameter: double SA_ActionCorrectedColor1..3	SA_ActionCorrected-Color1..3
Direction: Up	
Valid values:	
Minimum: Typical XYZ (0.0/0.0/0.0)	
Maximum: Typical XYZ (120.0/100.0/120.0)	
Description: Color information for corrected color coordinate in XYZ color space. This parameter is only available if SP_ActionName is 'enable_switching_output'.	
Parameter: double SA_ActionTransformedColor1..3	SA_ActionTransformed-Color1..3
Direction: Up	
Valid values:	
Minimum: Typical L*a*b* (0.0/-500.0/-200.0), L*u*v* (0.0/0.0/0.0), L*u'v' (0.0/0.0/0.0), xyY (0.0/0.0/0.0), XYZ (0.0/0.0/0.0)	
Maximum: Typical L*a*b* (100.0/500.0/200.0), L*u*v* (100.0/100.0/100.0), L*u'v' (1.0/1.0/1.0), xyY (1.0/1.0/100.0), XYZ (120.0/100.0/120.0)	
Description: Color information for transformed color coordinate. This parameter is only available if SP_ActionName is 'enable_switching_output'.	
Parameter: double SA_ActionRepresentationsRGB1..3	SA_ActionRepresentation-sRGB1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: RGB representation for color values. This parameter is only available if SP_ActionName is 'enable_switching_output'.	
Parameter: String SA_ActionDetectionMatcher	SA_ActionDetectionMatcher
Direction: Up	
Description: UUID of matching recognition result. This parameter is only available if SP_ActionName is 'enable_switching_output'.	

Parameter: int32_t SA_ActionDetectionOutputPattern	SA_ActionDetectionOutputPattern
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Description: The combination of values describes a logical state of the switching outputs of the sensor. This parameter is only available if SP_ActionName is 'enable_switching_output'.	
Parameter: double SA_ActionDetectionDistance1..3	SA_ActionDetectionDistance1..3
Direction: Up	
Valid values:	
Minimum: -100.0	
Maximum: 100.0	
Description: Color distance. This parameter is only available if SP_ActionName is 'enable_switching_output'.	
Parameter: int32_t SA_ActionInputs	SA_ActionInputs
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Bit coded array of input conditions. Coding started from bit 0 with trigger_0_edge_falling, ...edge_rising, ...level_high, ...level_low, ... up to trigger_3_level_low. This parameter is only available if SP_ActionName is 'enable_switching_output'.	
Parameter: double SA_ActionSignalLevel	SA_ActionSignalLevel
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: The signal level indicates the usage of the internal ADC sampling range. This parameter is only available if SP_ActionName is 'enable_switching_output'.	
Parameter: String SA_ActionSensorDetectableUUID	SA_ActionSensorDetectableUUID
Direction: Up	
Description: Unique ID of the color. This parameter is only available if SP_ActionName is 'teach_single'.	
Parameter: String SA_ActionSensorDetectableMatcherID	SA_ActionSensorDetectableMatcherID
Direction: Up	
Description: Reference to associated matcher object. This parameter is only available if SP_ActionName is 'teach_single'.	
Parameter: double SA_ActionSensorDetectableColor1..3	SA_ActionSensorDetectableColor1..3
Direction: Up	
Valid values:	
Minimum: Typical L*a*b* (0.0/-500.0/-200.0), L*u*v* (0.0/0.0/0.0), L*u'v' (0.0/0.0/0.0), xyY (0.0/0.0/0.0), XYZ (0.0/0.0/0.0)	
Maximum: Typical L*a*b* (100.0/500.0/200.0), L*u*v* (100.0/100.0/100.0), L*u'v' (1.0/1.0/1.0), xyY (1.0/1.0/100.0), XYZ (120.0/100.0/120.0)	
Description: Color information for color coordinate. This parameter is only available if SP_ActionName is 'teach_single'.	

Parameter: double SA_ActionSensorDetectableRepresentationsRGB1..3	SA_ActionSensorDe- tectableRepresentation- sRGB1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: RGB representation for color values. This parameter is only available if SP_ActionName is 'teach_single'.	
Parameter: int32_t SA_ActionLockState	SA_ActionLockState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Keylock state. This parameter is only available if SP_ActionName is 'keylock'.	
Parameter: int32_t SA_ActionLockViewState	SA_ActionLockViewState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: API view lock state. This parameter is only available if SP_ActionName is 'api_view_lock'.	
Parameter: int32_t SA_ActionLockEditState	SA_ActionLockEditState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: API edit lock state. This parameter is only available if SP_ActionName is 'api_edit_lock'.	
Parameter: String SA_ActionCompensationSettings	SA_ActionCompensationSet- tings
Direction: Up	
Description: Object of compensation settings in JSON notation. This parameter is only available if SP_ActionName is 'run_autogain'.	
Parameter: double SA_ActionSamplingLEDIntensity	SA_ActionSamplingLEDIn- tensity
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: Relative intensity of the internal emitter during the light phase. This parameter is only available if SP_ActionName is 'run_autogain'.	
Parameter: double SA_ActionSamplingBaseSR	SA_ActionSamplingBaseSR
Direction: Up	
Valid values:	
Minimum: 1.0	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The base sample rate determines the duration of a sampling period. This parameter is only available if SP_ActionName is 'run_autogain'.	

Parameter: double SA_ActionSamplingEffectiveSR	SA_ActionSamplingEffectiveSR
Direction: Up	
Valid values:	
Minimum: 1.0e-8	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: The effective sample rate is the numeric product of the base samplerate and the number of averages. This parameter is only available if SP_ActionName is 'run_autogain'.	
Parameter: double SA_ActionSamplingMinDesiredSR	SA_ActionSamplingMinDesiredSR
Direction: Up	
Valid values:	
Minimum: 0.01	
Maximum: 10000.0 for CFO100, 30000.0 for CFO200	
Unit: Hz	
Description: This informational value represents the sample rate that was requested during the most recent Autogain operation. This parameter is only available if SP_ActionName is 'run_autogain'.	
Parameter: int32_t SA_ActionSamplingLightPhase	SA_ActionSamplingLightPhase
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Defines if the sensor should periodically activate the internal emitter for sampling. This parameter is only available if SP_ActionName is 'run_autogain'.	
Parameter: int32_t SA_ActionSamplingDarkPhase	SA_ActionSamplingDarkPhase
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Defines if the sensor should periodically deactivate the internal emitter for sampling. This parameter is only available if SP_ActionName is 'run_autogain'.	
Parameter: int32_t SA_ActionSamplingAverageCount	SA_ActionSamplingAverageCount
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 100000000	
Description: Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples. This parameter is only available if SP_ActionName is 'run_autogain'.	
Parameter: int32_t SA_ActionSamplingAmplificationLevel	SA_ActionSamplingAmplificationLevel
Direction: Up	
Valid values:	
Minimum: -2147483648 (INT32_MIN)	
Maximum: 2147483647 (INT32_MAX)	
Description: The amplification level specifies the internal configuration of an amplifier. This parameter is only available if SP_ActionName is 'run_autogain'.	

16.2.7.6.4 Add_ActionTrigger (POST /api/sensor/action-triggers)

Create a new action trigger.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SP_ActionTriggerSource

SP_ActionTriggerSource

Direction: Down

Valid values:

- 0= TRG 0
- 1= TRG 1
- 2= TRG 2
- 3= TRG 3

Description: Trigger input.

Parameter: int32_t SP_ActionTriggerEvent

SP_ActionTriggerEvent

Direction: Down

Valid values:

- 0= High level
- 1= Low level
- 2= Rising edge
- 3= Falling edge

Description: Trigger event.

Parameter: int32_t SP_ActionTriggerEnableSwitchingOutput

SP_ActionTriggerEnableSwitchingOutput

Direction: Down

Valid values:

- 0= Off
- 1= On

Default: 0

Description: Trigger enables switching output.

Parameter: int32_t SP_ActionTriggerTeachSingle

SP_ActionTriggerTeachSingle

Direction: Down

Valid values:

- 0= Off
- 1= On

Default: 0

Description: Trigger teaches single detectable.

Parameter: String SP_ActionTriggerMatcherID

SP_ActionTriggerMatcherID

Direction: Down

Default: <not specified>, send Null object to sensor.

Description: Unique ID of matcher for detectable. This parameter is only used if SP_ActionTriggerTeachSingle is On. If this parameter is not specified, the first matcher (i.e. the one with the output bitmask 0x01) is used.

Parameter: int32_t SP_ActionTriggerRemoveMatcherDetectablesBefore

SP_ActionTriggerRemoveMatcherDetectablesBefore

Direction: Down

Valid values:

- 0= No
- 1= Yes

Default: <not specified>, do not send this parameter to sensor.

Description: Previous detectables of specified matcher are removed before.

This parameter is only used if SP_TriggerTeachSingle is On.

Parameter: int32_t SP_ActionTriggerKeylock	SP_ActionTriggerKeylock
Direction: Down	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger enables/disables key lock.	
Parameter: int32_t SP_ActionTriggerLockState	SP_ActionTriggerLockState
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description: Keylock state. This parameter is only used if SP_TriggerKeylock is On.	
Parameter: int32_t SP_ActionTriggerAutogain	SP_ActionTriggerAutogain
Direction: Down	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger runs autogain.	
Parameter: int32_t SP_ActionTriggerRemoveAllDetectables	SP_ActionTriggerRemoveAllDetectables
Direction: Down	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all detectables.	
Parameter: int32_t SP_ActionTriggerRemoveAllMatchers	SP_ActionTriggerRemoveAllMatchers
Direction: Down	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all matchers.	
Parameter: int32_t SP_ActionTriggerAPIViewlock	SP_ActionTriggerAPIViewlock
Direction: Down	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger enables/disables API view lock. Available from firmware version 1.6	
Parameter: int32_t SP_ActionTriggerLockViewState	SP_ActionTriggerLockViewState
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description: API view lock state. This parameter is only used if SP_ActionTriggerAPIViewlock is On.	

Parameter: int32_t SP_ActionTriggerAPIEditlock	SP_ActionTriggerAPIEditlock
Direction: Down	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger enables/disables API edit lock. Available from firmware version 1.6	
Parameter: int32_t SP_ActionTriggerLockEditState	SP_ActionTriggerLockEditState
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description: API edit lock state. This parameter is only used if SP_ActionTriggerAPIEditlock is On.	
Parameter: String SA_ActionTriggerUUID	SA_ActionTriggerUUID
Direction: Up	
Description: Unique ID of action trigger.	
Parameter: int32_t SA_ActionTriggerSource	SA_ActionTriggerSource
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= TRG 0	
1= TRG 1	
2= TRG 2	
3= TRG 3	
Description: Trigger input.	
Parameter: int32_t SA_ActionTriggerEvent	SA_ActionTriggerEvent
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= High level	
1= Low level	
2= Rising edge	
3= Falling edge	
Description: Trigger event.	
Parameter: int32_t SA_ActionTriggerEnableSwitchingOutput	SA_ActionTriggerEnableSwitchingOutput
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables switching output.	
Parameter: int32_t SA_ActionTriggerTeachSingle	SA_ActionTriggerTeachSingle
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger teaches single detectable.	

Parameter: String SA_ActionTriggerMatcherID	SA_ActionTriggerMatcherID
Direction: Up	
Description: Unique ID of matcher for detectable. This parameter is only available if SA_ActionTriggerTeachSingle is On.	
Parameter: int32_t SA_ActionTriggerRemoveMatcherDetectablesBefore	SA_ActionTriggerRemove- MatcherDetectablesBefore
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Previous detectables of specified matcher are removed before. This parameter is only available if SA_ActionTriggerMatcherID is available, too.	
Parameter: int32_t SA_ActionTriggerKeylock	SA_ActionTriggerKeylock
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger enables/disables key lock.	
Parameter: int32_t SA_ActionTriggerLockState	SA_ActionTriggerLockState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Keylock state. This parameter is only available if SA_ActionTriggerKeylock is On.	
Parameter: int32_t SA_ActionTriggerAutogain	SA_ActionTriggerAutogain
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger runs autogain.	
Parameter: int32_t SA_ActionTriggerRemoveAllDetectables	SA_ActionTriggerRe- moveAllDetectables
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all detectables.	
Parameter: int32_t SA_ActionTriggerRemoveAllMatchers	SA_ActionTriggerRe- moveAllMatchers
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all matchers.	

Parameter: int32_t SA_ActionTriggerAPIViewlock	SA_ActionTrigger- APIViewlock
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables/disables API view lock. Available from firmware version 1.6	
Parameter: int32_t SA_ActionTriggerLockViewState	SA_ActionTriggerLockView- State
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: API view lock state. This parameter is only available if SA_ActionTriggerAPIViewlock is On.	
Parameter: int32_t SA_ActionTriggerAPIEditlock	SA_ActionTriggerAPIEdit- lock
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables/disables API edit lock. Available from firmware version 1.6	
Parameter: int32_t SA_ActionTriggerLockEditState	SA_ActionTriggerLockEdit- State
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: API edit lock state. This parameter is only available if SA_ActionTriggerAPIEditlock is On.	

16.2.7.6.5 Set_ActionTrigger (PUT /api/sensor/action-triggers/{itemId})

Modifies a single action trigger.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_ActionTriggerUUID	SP_ActionTriggerUUID
Direction: Down	
Description: Unique ID of action trigger.	
Parameter: int32_t SP_ActionTriggerSource	SP_ActionTriggerSource
Direction: Down	
Valid values:	
0= TRG 0	
1= TRG 1	
2= TRG 2	
3= TRG 3	
Description: Trigger input. Only used if SP_ActionTriggerEvent is specified	

Parameter: int32_t SP_ActionTriggerEvent	SP_ActionTriggerEvent
Direction: Down	
Valid values:	
0= High level	
1= Low level	
2= Rising edge	
3= Falling edge	
Default: <not specified>, do not send this parameter to sensor.	
Description: Trigger event.	
Parameter: int32_t SP_ActionTriggerEnableSwitchingOutput	SP_ActionTriggerEnableSwitchingOutput
Direction: Down	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger enables switching output.	
Parameter: int32_t SP_ActionTriggerTeachSingle	SP_ActionTriggerTeachSingle
Direction: Down	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger teaches single detectable.	
Parameter: String SP_ActionTriggerMatcherID	SP_ActionTriggerMatcherID
Direction: Down	
Default: <not specified>, do not send this parameter to sensor.	
Description: Unique ID of matcher for detectable. This parameter is only used if SP_ActionTriggerTeachSingle is On. If this parameter is not specified, the first matcher (i.e. the one with the output bitmask 0x01) is used.	
Parameter: int32_t SP_ActionTriggerRemoveMatcherDetectablesBefore	SP_ActionTriggerRemoveMatcherDetectablesBefore
Direction: Down	
Valid values:	
0= No	
1= Yes	
Default: <not specified>, do not send this parameter to sensor.	
Description: Previous detectables of specified matcher are removed before. This parameter is only used if SP_ActionTriggerTeachSingle is On.	
Parameter: int32_t SP_ActionTriggerKeylock	SP_ActionTriggerKeylock
Direction: Down	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger enables/disables key lock.	
Parameter: int32_t SP_ActionTriggerLockState	SP_ActionTriggerLockState
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description: Keylock state. This parameter is only used if SP_TriggerKeylock is On.	

Parameter: int32_t SP_ActionTriggerAutogain	SP_ActionTriggerAutogain
Direction: Down	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger runs autogain.	
Parameter: int32_t SP_ActionTriggerRemoveAllDetectables	SP_ActionTriggerRe-
Direction: Down	moveAllDetectables
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all detectables.	
Parameter: int32_t SP_ActionTriggerRemoveAllMatchers	SP_ActionTriggerRe-
Direction: Down	moveAllMatchers
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all matchers.	
Parameter: int32_t SP_ActionTriggerAPIViewlock	SP_ActionTrigger-
Direction: Down	APIViewlock
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger enables/disables API view lock. Available from firmware version 1.6	
Parameter: int32_t SP_ActionTriggerLockViewState	SP_ActionTriggerLockView-
Direction: Down	State
Valid values:	
0= No	
1= Yes	
Description: API view lock state. This parameter is only used if SP_ActionTriggerAPIViewlock is On.	
Parameter: int32_t SP_ActionTriggerAPIEditlock	SP_ActionTriggerAPIEdit-
Direction: Down	lock
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger enables/disables API edit lock. Available from firmware version 1.6	
Parameter: int32_t SP_ActionTriggerLockEditState	SP_ActionTriggerLockEdit-
Direction: Down	State
Valid values:	
0= No	
1= Yes	
Description: API edit lock state. This parameter is only used if SP_ActionTriggerAPIEditlock is On.	

Parameter: String SA_ActionTriggerUUID	SA_ActionTriggerUUID
Direction: Up	
Description: Unique ID of action trigger.	
Parameter: int32_t SA_ActionTriggerSource	SA_ActionTriggerSource
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= TRG 0	
1= TRG 1	
2= TRG 2	
3= TRG 3	
Description: Trigger input.	
Parameter: int32_t SA_ActionTriggerEvent	SA_ActionTriggerEvent
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= High level	
1= Low level	
2= Rising edge	
3= Falling edge	
Description: Trigger event.	
Parameter: int32_t SA_ActionTriggerEnableSwitchingOutput	SA_ActionTriggerEnableSwitchingOutput
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables switching output.	
Parameter: int32_t SA_ActionTriggerTeachSingle	SA_ActionTriggerTeachSingle
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger teaches single detectable.	
Parameter: String SA_ActionTriggerMatcherID	SA_ActionTriggerMatcherID
Direction: Up	
Description: Unique ID of matcher for detectable. This parameter is only available if SA_ActionTriggerTeachSingle is On.	
Parameter: int32_t SA_ActionTriggerRemoveMatcherDetectablesBefore	SA_ActionTriggerRemoveMatcherDetectablesBefore
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Previous detectables of specified matcher are removed before. This parameter is only available if SA_ActionTriggerMatcherID is available, too.	

Parameter: int32_t SA_ActionTriggerKeylock	SA_ActionTriggerKeylock
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger enables/disables key lock.	
Parameter: int32_t SA_ActionTriggerLockState	SA_ActionTriggerLockState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Keylock state. This parameter is only available if SA_ActionTriggerKeylock is On.	
Parameter: int32_t SA_ActionTriggerAutogain	SA_ActionTriggerAutogain
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger runs autogain.	
Parameter: int32_t SA_ActionTriggerRemoveAllDetectables	SA_ActionTriggerRemoveAllDetectables
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all detectables.	
Parameter: int32_t SA_ActionTriggerRemoveAllMatchers	SA_ActionTriggerRemoveAllMatchers
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all matchers.	
Parameter: int32_t SA_ActionTriggerAPIViewlock	SA_ActionTriggerAPIViewlock
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables/disables API view lock. Available from firmware version 1.6	
Parameter: int32_t SA_ActionTriggerLockViewState	SA_ActionTriggerLockViewState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: API view lock state. This parameter is only available if SA_ActionTriggerAPIViewlock is On.	

Parameter: int32_t SA_ActionTriggerAPIEditlock	SA_ActionTriggerAPIEditlock
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables/disables API edit lock. Available from firmware version 1.6	
Parameter: int32_t SA_ActionTriggerLockEditState	SA_ActionTriggerLockEditState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: API edit lock state. This parameter is only available if SA_ActionTriggerAPIEditlock is On.	

16.2.7.6.6 Get_ActionTrigger (GET /api/sensor/action-triggers/{itemId})

Returns a single action trigger.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_ActionTriggerUUID	SP_ActionTriggerUUID
Direction: Down	
Description: Unique ID of action trigger.	
Parameter: String SA_ActionTriggerUUID	SA_ActionTriggerUUID
Direction: Up	
Description: Unique ID of action trigger.	
Parameter: int32_t SA_ActionTriggerSource	SA_ActionTriggerSource
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= TRG 0	
1= TRG 1	
2= TRG 2	
3= TRG 3	
Description: Trigger input.	
Parameter: int32_t SA_ActionTriggerEvent	SA_ActionTriggerEvent
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= High level	
1= Low level	
2= Rising edge	
3= Falling edge	
Description: Trigger event.	

Parameter: int32_t SA_ActionTriggerEnableSwitchingOutput	SA_ActionTriggerEnableSwitchingOutput
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables switching output.	
Parameter: int32_t SA_ActionTriggerTeachSingle	SA_ActionTriggerTeachSingle
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger teaches single detectable.	
Parameter: String SA_ActionTriggerMatcherID	SA_ActionTriggerMatcherID
Direction: Up	
Description: Unique ID of matcher for detectable. This parameter is only available if SA_ActionTriggerTeachSingle is On.	
Parameter: int32_t SA_ActionTriggerRemoveMatcherDetectablesBefore	SA_ActionTriggerRemoveMatcherDetectablesBefore
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Previous detectables of specified matcher are removed before. This parameter is only available if SA_ActionTriggerMatcherID is available, too.	
Parameter: int32_t SA_ActionTriggerKeylock	SA_ActionTriggerKeylock
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger enables/disables key lock.	
Parameter: int32_t SA_ActionTriggerLockState	SA_ActionTriggerLockState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Keylock state. This parameter is only available if SA_ActionTriggerKeylock is On.	
Parameter: int32_t SA_ActionTriggerAutogain	SA_ActionTriggerAutogain
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger runs autogain.	

Parameter: int32_t SA_ActionTriggerRemoveAllDetectables	SA_ActionTriggerRemoveAllDetectables
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all detectables.	
Parameter: int32_t SA_ActionTriggerRemoveAllMatchers	SA_ActionTriggerRemoveAllMatchers
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all matchers.	
Parameter: int32_t SA_ActionTriggerAPIViewlock	SA_ActionTriggerAPIViewlock
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables/disables API view lock. Available from firmware version 1.6	
Parameter: int32_t SA_ActionTriggerLockViewState	SA_ActionTriggerLockViewState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: API view lock state. This parameter is only available if SA_ActionTriggerAPIViewlock is On.	
Parameter: int32_t SA_ActionTriggerAPIEditlock	SA_ActionTriggerAPIEditlock
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables/disables API edit lock. Available from firmware version 1.6	
Parameter: int32_t SA_ActionTriggerLockEditState	SA_ActionTriggerLockEditState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: API edit lock state. This parameter is only available if SA_ActionTriggerAPIEditlock is On.	

16.2.7.6.7 Get_ActionTriggerList (GET /api/sensor/action-triggers)

Retrieves a list of available action triggers.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SP_ActionTriggerSource

SP_ActionTriggerSource

Direction: Down

Valid values:

- 0= TRG 0
- 1= TRG 1
- 2= TRG 2
- 3= TRG 3

Description: Trigger input. Only used if SP_ActionTriggerEvent is specified

Parameter: int32_t SP_ActionTriggerEvent

SP_ActionTriggerEvent

Direction: Down

Valid values:

- 0= High level
- 1= Low level
- 2= Rising edge
- 3= Falling edge

Default: <not specified>, do not send this parameter to sensor.

Description: Trigger event.

Parameter: int32_t SA_ActionTriggerEntries

SA_ActionTriggerEntries

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 16

Description: Number of entries in action trigger array. All following parameters exists from 1 to this number, e.g. SA_ActionTrigger1UUID, SA_ActionTrigger2UUID, ...

Parameter: String SA_ActionTrigger1..xUUID

SA_ActionTrigger1..xUUID

Direction: Up

Description: Unique ID of action trigger.

Parameter: int32_t SA_ActionTrigger1..xSource

SA_ActionTrigger1..xSource

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= TRG 0
- 1= TRG 1
- 2= TRG 2
- 3= TRG 3

Description: Trigger input.

Parameter: int32_t SA_ActionTrigger1..xEvent

SA_ActionTrigger1..xEvent

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= High level
- 1= Low level
- 2= Rising edge
- 3= Falling edge

Description: Trigger event.

Parameter: int32_t SA_ActionTrigger1..xEnableSwitchingOutput	SA_ActionTrigger1..xEnableSwitchingOutput
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables switching output.	
Parameter: int32_t SA_ActionTrigger1..xTeachSingle	SA_ActionTrigger1..xTeachSingle
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger teaches single detectable.	
Parameter: String SA_ActionTrigger1..xMatcherID	SA_ActionTrigger1..xMatcherID
Direction: Up	
Description: Unique ID of matcher for detectable. This parameter is only available if SA_ActionTrigger1..xTeachSingle is On.	
Parameter: int32_t SA_ActionTrigger1..xRemoveMatcherDetectablesBefore	SA_ActionTrigger1..xRemoveMatcherDetectablesBefore
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Previous detectables of specified matcher are removed before. This parameter is only available if SA_ActionTrigger1..xMatcherID is available, too.	
Parameter: int32_t SA_ActionTrigger1..xKeylock	SA_ActionTrigger1..xKeylock
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger enables/disables key lock.	
Parameter: int32_t SA_ActionTrigger1..xLockState	SA_ActionTrigger1..xLockState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Keylock state. This parameter is only available if SA_ActionTrigger1..xKeylock is On.	
Parameter: int32_t SA_ActionTrigger1..xAutogain	SA_ActionTrigger1..xAutogain
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger runs autogain.	

Parameter: int32_t SA_ActionTrigger1..xRemoveAllDetectables	SA_ActionTrigger1..xRemoveAllDetectables
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all detectables.	
Parameter: int32_t SA_ActionTrigger1..xRemoveAllMatchers	SA_ActionTrigger1..xRemoveAllMatchers
Direction: Up	
Valid values:	
0= Off	
1= On	
Default: 0	
Description: Trigger removes all matchers.	
Parameter: int32_t SA_ActionTrigger1..xAPIViewlock	SA_ActionTrigger1..xAPIViewlock
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables/disables API view lock. Available from firmware version 1.6	
Parameter: int32_t SA_ActionTrigger1..xLockViewState	SA_ActionTrigger1..xLockViewState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: API view lock state. This parameter is only available if SA_ActionTrigger1..xAPIViewlock is On.	
Parameter: int32_t SA_ActionTrigger1..xAPIEditlock	SA_ActionTrigger1..xAPIEditlock
Direction: Up	
Valid values:	
0= Off	
1= On	
Description: Trigger enables/disables API edit lock. Available from firmware version 1.6	
Parameter: int32_t SA_ActionTrigger1..xLockEditState	SA_ActionTrigger1..xLockEditState
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: API edit lock state. This parameter is only available if SA_ActionTrigger1..xAPIEditlock is On.	

16.2.7.6.8 Delete_ActionTrigger (DELETE /api/sensor/action-triggers/{itemId})

Delete a single action trigger.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: String SP_ActionTriggerUUID

SP_ActionTriggerUUID

Direction: Down

Description: Unique ID of action trigger.

16.2.7.6.9 Delete_ActionTriggerList (DELETE /api/sensor/action-triggers)

Delete any action trigger.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SP_ActionTriggerSource

SP_ActionTriggerSource

Direction: Down

Valid values:

- 0= TRG 0
- 1= TRG 1
- 2= TRG 2
- 3= TRG 3

Description: Trigger input. Only used if SP_ActionTriggerEvent is specified

Parameter: int32_t SP_ActionTriggerEvent

SP_ActionTriggerEvent

Direction: Down

Valid values:

- 0= High level
- 1= Low level
- 2= Rising edge
- 3= Falling edge

Default: <not specified>, do not send this parameter to sensor.

Description: Trigger event.

16.2.7.7 Samples

16.2.7.7.1 Get_SensorSample (GET /api/sensor/samples/current, sample show)

Get the last recognition results.

Parameter: int32_t SP_SampleProperty

SP_SampleProperty

Direction: Down

Valid values:

- 0= All
- 1= Color
- 2= Detection
- 3= Output pattern
- 4= Timestamp
- 5= Trigger

Valid for sensor:

CFO100 / CFO200 over RS232

Default: <not specified>, do not send this parameter to sensor (means All).

Description: Name of sample property.

Parameter: String SA_SensorSampleUUID

SA_SensorSampleUUID

Direction: Up

Description: Unique ID of current sample. This parameter is only available over Ethernet or if SP_SampleProperty is All.

Parameter: double SA_SensorSampleTransformedColor1..3	SA_SensorSampleTransformedColor1..3
Direction: Up	
Valid values:	
Minimum: Typical L*a*b* (0.0/-500.0/-200.0), L*u*v* (0.0/0.0/0.0), L*u'v' (0.0/0.0/0.0), xyY (0.0/0.0/0.0), XYZ (0.0/0.0/0.0)	
Maximum: Typical L*a*b* (100.0/500.0/200.0), L*u*v* (100.0/100.0/100.0), L*u'v' (1.0/1.0/1.0), xyY (1.0/1.0/100.0), XYZ (120.0/100.0/120.0)	
Description: Color information for transformed color coordinate. This parameter is only available over Ethernet or if SP_SampleProperty is All or Color.	
Parameter: double SA_SensorSampleRepresentationsRGB1..3	SA_SensorSampleRepresentationsRGB1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: RGB representation for color values. This parameter is only available over Ethernet or if SP_SampleProperty is All.	
Parameter: double SA_SensorSampleCorrectedColor1..3	SA_SensorSampleCorrectedColor1..3
Direction: Up	
Valid values:	
Minimum: Typical XYZ (0.0/0.0/0.0)	
Maximum: Typical XYZ (120.0/100.0/120.0)	
Description: Color information for corrected color coordinate in XYZ color space. This parameter is only available over Ethernet or if SP_SampleProperty is All.	
Parameter: int32_t SA_SensorSampleDetectionOutputPattern	SA_SensorSampleDetectionOutputPattern
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Description: The combination of values describes a logical state of the switching outputs of the sensor. This parameter is only available over Ethernet or if SP_SampleProperty is All or Output pattern.	
Parameter: double SA_SensorSampleDetectionDistance1..3	SA_SensorSampleDetectionDistance1..3
Direction: Up	
Valid values:	
Minimum: -100.0	
Maximum: 100.0	
Description: Color distance. This parameter is only available over Ethernet or if SP_SampleProperty is All or Detection (and if returned from sensor).	
Parameter: String SA_SensorSampleDetectionMatcher	SA_SensorSampleDetectionMatcher
Direction: Up	
Description: UUID of matching recognition result. This parameter is only available over Ethernet or if SP_SampleProperty is All or Detection (and if returned from sensor).	

Parameter: int32_t SA_SensorSampleInputs	SA_SensorSampleInputs
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Bit coded array of input conditions. Coding started from bit 0 with trigger_0_edge_falling, ...edge_rising, ...level_high, ...level_low, ... up to trigger_3_level_low. This parameter is only available over Ethernet or if SP_SampleProperty is All or Trigger.	
Parameter: double SA_SensorSampleTimestamp	SA_SensorSampleTimestamp
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 18446744073709.551616	
Unit: s	
Description: Time stamp (sensor run time) for color value. This parameter is only available over Ethernet or if SP_SampleProperty is All or Timestamp.	
Parameter: double SA_SensorSampleSignalLevel	SA_SensorSampleSignal- Level
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: The signal level indicates the usage of the internal ADC sampling range. This parameter is only available over Ethernet or if SP_SampleProperty is All.	

16.2.7.7.2 Get_SensorSampleList (GET /api/sensor/samples)

Get list of historical recognition results.

Valid for sensor:

CFO100 / CFO200 over Ethernet

Parameter: int32_t SA_SensorSampleEntries	SA_SensorSampleEntries
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of entries in the sensor sample array. All following parameters exists from 1 to this number, e.g. SA_SensorSample1UUID, SA_SensorSample2UUID, ...	
Parameter: String SA_SensorSample1..xUUID	SA_SensorSample1..xUUID
Direction: Up	
Description: Unique ID of current sample.	
Parameter: double SA_SensorSample1..xTransformedColor1..3	SA_SensorSam- ple1..xTransformedColor1..3
Direction: Up	
Valid values:	
Minimum: Typical L*a*b* (0.0/-500.0/-200.0), L*u*v* (0.0/0.0/0.0), L*u'v' (0.0/0.0/0.0), xyY (0.0/0.0/0.0), XYZ (0.0/0.0/0.0)	
Maximum: Typical L*a*b* (100.0/500.0/200.0), L*u*v* (100.0/100.0/100.0), L*u'v' (1.0/1.0/1.0), xyY (1.0/1.0/100.0), XYZ (120.0/100.0/120.0)	
Description: Color information for transformed color coordinate.	

Parameter: double SA_SensorSample1..xRepresentationsRGB1..3	SA_SensorSam- ple1..xRepresentationsRGB1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: RGB representation for color values.	
Parameter: double SA_SensorSample1..xCorrectedColor1..3	SA_SensorSam- ple1..xCorrectedColor1..3
Direction: Up	
Valid values:	
Minimum: Typical XYZ (0.0/0.0/0.0)	
Maximum: Typical XYZ (120.0/100.0/120.0)	
Description: Color information for corrected color coordinate in XYZ color space.	
Parameter: int32_t SA_SensorSample1..xDetectionOutputPattern	SA_SensorSam- ple1..xDetectionOutputPattern
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 7 for CFO100, 255 for CFO200	
Description: The combination of values describes a logical state of the switching outputs of the sensor.	
Parameter: double SA_SensorSample1..xDetectionDistance1..3	SA_SensorSam- ple1..xDetectionDistance1..3
Direction: Up	
Valid values:	
Minimum: -100.0	
Maximum: 100.0	
Description: Color distance. This parameter is only available if returned from sensor.	
Parameter: String SA_SensorSample1..xDetectionMatcher	SA_SensorSam- ple1..xDetectionMatcher
Direction: Up	
Description: UUID of matching recognition result. This parameter is only available if returned from sensor.	
Parameter: int32_t SA_SensorSample1..xInputs	SA_SensorSample1..xInputs
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Bit coded array of input conditions. Coding started from bit 0 with trigger_0_edge_falling, ...edge_rising, ...level_high, ...level_low, ... up to trigger_3_level_low.	
Parameter: double SA_SensorSample1..xTimestamp	SA_SensorSam- ple1..xTimestamp
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 18446744073709.551616	
Unit: s	
Description: Time stamp (sensor run time) for color value.	

Parameter: double SA_SensorSample1..xSignalLevel

Direction: Up

Valid values:

- Minimum:** 0.0
- Maximum:** 1.0

Description: The signal level indicates the usage of the internal ADC sampling range.

SA_SensorSam-
ple1..xSignalLevel

16.2.7.7.3 Start_SensorSampleStream (sample stream)

Retrieve a continuous stream of color samples over RS232 from the sensor.

Valid for sensor:
CFO100 / CFO200 over RS232

Parameter: int32_t SP_SensorSampleCount

SP_SensorSampleCount

Direction: Down

Valid values:

- Minimum:** 0
- Maximum:** 2147483647 (INT32_MAX)

Default: <not specified>, do not send this parameter to sensor.

Description: Number of values to measure. 0 or not specified means infinite.

Parameter: double SP_SensorSampleFrequency

SP_SensorSampleFrequency

Direction: Down

Valid values:

- Minimum:** 2.22507e-308 (DBL_MIN)
- Maximum:** 1.79769e+308 (DBL_MAX)

Default: <not specified>, do not send this parameter to sensor.

Description: Measure frequency. This parameter is only used if SP_SensorSampleCount is specified. If it is not send to sensor it means infinite.

16.2.7.7.4 Abort_SensorSampleStream (ETX, CTRL-C)

Abort retrieve a continuous stream of color samples over RS232 from the sensor.
When MEDAQLib opens a connection to sensor over RS232 and before first command after start streaming, MEDAQLib automatically calls this command.

Valid for sensor:
CFO100 / CFO200 over RS232

16.2.8 Settings

16.2.8.1 Update_Settings (POST /api/settings)

Update specified settings at the device.

Valid for sensor:
CFO100 / CFO200 over Ethernet

Parameter: String SP_ImportCategories

SP_ImportCategories

Direction: Down

Valid values:

access
firmware
keypad
network
outputs
sensor
system

Description: Newline separated array of setting types to be imported. The list of available settings can be retrieved by [Get_SensorCapabilities](#) parameter [SA_SettingCategories](#)

Parameter: String SP_Settings

SP_Settings

Direction: Down

Description: BASE64 encoded sensor settings.

16.2.8.2 Set_Settings (PUT /api/settings)

Valid for sensor:

CFO100 / CFO200 over Ethernet

Set all settings at the device.

Parameter: String SP_Settings

SP_Settings

Direction: Down

Description: BASE64 encoded sensor settings.

16.2.8.3 Get_Settings (GET /api/settings)

Valid for sensor:

CFO100 / CFO200 over Ethernet

Export the complete configuration of the device.

Parameter: String SA_Settings

SA_Settings

Direction: Up

Description: BASE64 encoded sensor settings.

16.2.8.4 Delete_Settings (DELETE /api/settings, system settings reset)

Sensor configuration reset.

16.2.9 System

16.2.9.1 Set_System (PUT /api/system, system hostname set HOSTNAME)

Change of system settings.

Parameter: String SP_Hostname

SP_Hostname

Direction: Down

Description: Host name used in DHCP queries and elsewhere in network.

Parameter: String SA_Hostname SA_Hostname
Direction: Up
Description: Host name used in DHCP queries and elsewhere in network.

16.2.9.2 Get_System (GET /api/system, system hostname show)

Get system settings.

Parameter: String SA_Hostname SA_Hostname
Direction: Up
Description: Host name used in DHCP queries and elsewhere in network.

16.2.9.3 Exec_FactoryReset (POST /api/system/factory-reset, firmware recovery restore)

Sensor reset to factory settings followed by reboot.

16.2.9.4 Exec_Reboot (POST /api/system/reboot, system reboot)

Reboot sensor.

16.2.9.5 Set_Time (PUT /api/system/time, system time set now TIME)

Change time settings.

Parameter: String SP_Now SP_Now
Direction: Down
Default: <not specified>, do not send this parameter to sensor.
Description: Representation of date corresponding to ISO 8601.

Parameter: String SP_Timezone SP_Timezone
Direction: Down
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Default: <not specified>, do not send this parameter to sensor.
Description: Name of currently configured time zone.

Parameter: String SP_NTPServers SP_NTPServers
Direction: Down
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Default: <not specified>, do not send this parameter to sensor.
Description: Newline separated array of currently used time servers (IP / Hostname).

Parameter: String SA_NTPServers SA_NTPServers
Direction: Up
Valid for sensor:
 CFO100 / CFO200 over Ethernet
Description: Newline separated array of currently used time servers (IP / Hostname).

Parameter: String SA_DefaultNTPServers	SA_DefaultNTPServers
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Newline separated array of default time servers (IP / Hostname).	
Parameter: String SA_Now	SA_Now
Direction: Up	
Description: Representation of date corresponding to ISO 8601.	
Parameter: String SA_Timezone	SA_Timezone
Direction: Up	
Valid for sensor:	
CFO100 / CFO200 over Ethernet	
Description: Name of currently configured time zone.	

16.2.9.6 Get_Time (GET /api/system/time, system time show)

Get time settings.

Parameter: int32_t SP_TimeProperty	SP_TimeProperty
Direction: Down	
Valid values:	
0= All	
1= Now	
2= Timezone	
Valid for sensor:	
CFO100 / CFO200 over RS232	
Default: <not specified>, do not send this parameter to sensor (means All).	
Description: Name of time property.	
Parameter: String SA_Now	SA_Now
Direction: Up	
Description: Representation of date corresponding to ISO 8601. This parameter is only available over Ethernet or if SP_TimeProperty is All or Now.	
Parameter: String SA_Timezone	SA_Timezone
Direction: Up	
Description: Name of currently configured time zone. This parameter is only available over Ethernet or if SP_TimeProperty is All or Timezone.	
Parameter: String SA_NTPServers	SA_NTPServers
Direction: Up	
Description: Newline separated array of currently used time servers (IP / Hostname). This parameter is only available over Ethernet or if SP_TimeProperty is All.	
Parameter: String SA_DefaultNTPServers	SA_DefaultNTPServers
Direction: Up	
Description: Newline separated array of default time servers (IP / Hostname). This parameter is only available over Ethernet or if SP_TimeProperty is All.	

16.2.9.7 Set_Timezone (system time set timezone TIMEZONE)

Set time zones.

Valid for sensor:

CFO100 / CFO200 over RS232

Parameter: String SP_Timezone

SP_Timezone

Direction: Down

Description: Time zone to use.

Parameter: String SA_Timezone

SA_Timezone

Direction: Up

Description: Currently used time zone.

16.2.9.8 Get_Timezones (GET /api/system/time/zones, system timezones list)

Query for time zones available.

Parameter: String SA_Timezones

SA_Timezones

Direction: Up

Description: Newline separated array of time zones.

16.2.9.9 Set_RS232OutputFormat (set output-format FORMAT)

Set RS232 output format.

MEDAQLib ONLY supports JSON. When connecting via RS232 MEDAQLib call this command automatically.

Valid for sensor:

CFO100 / CFO200 over RS232

Parameter: int32_t SP_RS232OutputFormat

SP_RS232OutputFormat

Direction: Down

Valid values:

0= Human

1= JSON

Description: Output format of sensor data/answers.

16.2.9.10 Set_RS232Echo (set echo STATE)

Set RS232 echo.

Valid for sensor:

CFO100 / CFO200 over RS232

Parameter: int32_t SP_RS232Echo

SP_RS232Echo

Direction: Down

Valid values:

0= Off

1= On

Description: Switches echo off/on.

16.3 Commands for MFA-7/14/21/28

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- RS232 (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
- IF2004_USB (native).
- IF2008 (native).
- IF2008_ETH (native).

If first bit of `IP_AutomaticMode` is set (1), MEDAQLib calls automatically sensor command `Get_AllParameters` (`SP_Additional= 1`) after `OpenSensor`.

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate, to interpret and scale data and to assign values. If second bit of `IP_AutomaticMode` is set (2), MEDAQLib calls optionally sensor command `Set_DataOutInterface` at `OpenSensor`.

Meaning of raw and scaled values (function `Poll` and `TransferData`):

- Raw values are as it comes directly from sensor.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command `Get_AllParameters` (`SP_Additional= 1`)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

16.3.1 General commands

16.3.1.1 General

16.3.1.1.1 Get_Help (HELP)

Retrieve a help text from sensor for a specific command.

Parameter: String SP_Command

SP_Command

Direction: Down

Valid values:

"" (empty string, means general help)
or any command name

Description: Name of the command.

Parameter: String SA_HelpText

SA_HelpText

Direction: Up

Description: Help text to the command.

16.3.1.1.2 Get_Info (GETINFO)

Retrieve information about the sensor.

Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Description: Name of the sensor.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the sensor.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the sensor.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the sensor.	
Parameter: String SA_HardwareRevision	SA_HardwareRevision
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of the controller board.	

16.3.1.1.3 Get_ChannelCount (GETCHANNELCNT)

Get number of channels of controller.

Parameter: int32_t SA_ChannelCount	SA_ChannelCount
Direction: Up	
Valid values:	
7 (for MFA-7)	
14 (for MFA-14)	
21 (for MFA-21)	
28 (for MFA-28)	
Description: Number of channels.	

16.3.1.1.4 Get_Status (STATUS)

Returns information about the status of the sensor.

Parameter: int32_t SP_Channel

SP_Channel

Direction: Down

Valid values:

0= All channels

1..28= Channel 1 to 28

Description: Channel to process

Parameter: int32_t SA_Status01..28

SA_Status01..28

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Error

1= Autogain

2= Overflow

3= Configure

4= Measure

Description: Status of sensor channel.

16.3.1.1.5 Get_OutputInfo (GETOUTINFO)

Retrieve information which data is output at RS232/RS422/USB interface.

Parameter: int32_t SA_OutputCh01..28Color1

SA_OutputCh01..28Color1

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if color 1 is transmitted.

Parameter: int32_t SA_OutputCh01..28Color2

SA_OutputCh01..28Color2

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if color 2 is transmitted.

Parameter: int32_t SA_OutputCh01..28Color3

SA_OutputCh01..28Color3

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if color 3 is transmitted.

Parameter: int32_t SA_OutputCh01..28Temperature

SA_Out-
putCh01..28Temperature

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if temperature is transmitted.

Parameter: int32_t SA_OutputCh01..28Wavelength	SA_Out-putCh01..28Wavelength
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if wavelength is transmitted.	
Parameter: int32_t SA_OutputCh01..28Timestamp	SA_Out-putCh01..28Timestamp
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	

16.3.1.1.6 Reset_Boot (RESET)

Resets the sensor.

At this command the sensor may change output data after reboot. If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

16.3.1.1.7 Reset_Counter (RESETCNT)

Resets sensor timestamp values.

16.3.1.1.8 Get_AllParameters (PRINT)

Get all parameters from sensor.

Parameter: int32_t SP_Additional	SP_Additional
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description: If set, additional information about sensor is output.	
Parameter: int32_t SA_SensorBaudrate	SA_SensorBaudrate
Direction: Up	
Valid values:	
9600	
115200	
230400	
Unit: Baud	
Description: Baudrate of sensor.	
Parameter: int32_t SA_ChannelCount	SA_ChannelCount
Direction: Up	
Valid values:	
7 (for MFA-7)	
14 (for MFA-14)	
21 (for MFA-21)	
28 (for MFA-28)	
Description: Number of channels.	

Parameter: int32_t SA_ColorSpace	SA_ColorSpace
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= XYZ	
1= xyY	
2= Luv	
3= uvL	
4= RGB	
Description: Actually used colorspace	
Parameter: double SA_Measrate	SA_Measrate
Direction: Up	
Valid values:	
Minimum: 0.1	
Maximum: 100.0	
Unit: Hz	
Description: Datarate of measurement.	
Parameter: int32_t SA_DataOutInterface	SA_DataOutInterface
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= On (RS232/RS422/USB)	
Description: Active interface for data output.	
Parameter: int32_t SA_OutputCh01..28	SA_OutputCh01..28
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if channel is transmitted.	
Parameter: int32_t SA_OutputTemperature	SA_OutputTemperature
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature is transmitted.	
Parameter: int32_t SA_OutputWavelength	SA_OutputWavelength
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if wavelength is transmitted.	
Parameter: int32_t SA_OutputTimestamp	SA_OutputTimestamp
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	

Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Description: Name of the sensor.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the sensor.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the sensor.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the sensor.	
Parameter: String SA_HardwareRevision	SA_HardwareRevision
Direction: Up	
Valid values:	
Numeric value	
Description: Revision index of the controller board.	
Parameter: int32_t SA_OutputCh01..28Color1	SA_OutputCh01..28Color1
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color 1 is transmitted.	
Parameter: int32_t SA_OutputCh01..28Color2	SA_OutputCh01..28Color2
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color 2 is transmitted.	
Parameter: int32_t SA_OutputCh01..28Color3	SA_OutputCh01..28Color3
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if color 3 is transmitted.	

Parameter: int32_t SA_OutputCh01..28Temperature	SA_Out-putCh01..28Temperature
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if temperature is transmitted.	
Parameter: int32_t SA_OutputCh01..28Wavelength	SA_Out-putCh01..28Wavelength
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if wavelength is transmitted.	
Parameter: int32_t SA_OutputCh01..28Timestamp	SA_Out-putCh01..28Timestamp
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	
Parameter: int32_t SA_Status01..28	SA_Status01..28
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Error	
1= Autogain	
2= Overflow	
3= Configure	
4= Measure	
Description: Status of sensor channel.	
Parameter: double SA_DarkCorrOffset1_01..28	SA_DarkCorrOffset1_01..28
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Dark reference for color 1.	
Parameter: double SA_DarkCorrOffset2_01..28	SA_DarkCorrOffset2_01..28
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Dark reference for color 2.	
Parameter: double SA_DarkCorrOffset3_01..28	SA_DarkCorrOffset3_01..28
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Dark reference for color 3.	

Parameter: double SA_WhiteCorrFactor1_01..28	SA_WhiteCorrFactor1_01..28
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 2.0	
Description: White reference for color 1.	
Parameter: double SA_WhiteCorrFactor2_01..28	SA_WhiteCorrFactor2_01..28
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 2.0	
Description: White reference for color 2.	
Parameter: double SA_WhiteCorrFactor3_01..28	SA_WhiteCorrFactor3_01..28
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 2.0	
Description: White reference for color 3.	
Parameter: int32_t SA_IntegrationTime01..28	SA_IntegrationTime01..28
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 14	
Description: Real integration time [in ms] is $2^{(0..14)}$, this is 1..16384 ms.	
Parameter: int32_t SA_Gain01..28	SA_Gain01..28
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 11	
Description: Real gain level is $2^{(0..11)}$, this is 1..2048.	
Parameter: int32_t SA_Averaging01..28	SA_Averaging01..28
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of values for moving average.	

16.3.1.2 Interfaces

16.3.1.2.1 Set_Baudrate (BAUDRATE)

Set baudrate of sensor for serial RS232/RS422/USB communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_SensorBaudrate	SP_SensorBaudrate
Direction: Down	
Valid values:	
9600	
115200	
230400	
Unit: Baud	
Description: Baudrate of sensor.	

Parameter: int32_t CP_InterruptDataTransfer CP_InterruptDataTransfer
Direction: Down
Valid values:
 0= false
 1= true
Default: 0
Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Set_DataOutInterface](#)) and enabling it again at end (only if enabled before).

16.3.1.2.2 Get_Baudrate (BAUDRATE)

Get baudrate of sensor for serial RS232/RS422/USB communication.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate
Direction: Up
Valid values:
 9600
 115200
 230400
Unit: Baud
Description: Baudrate of sensor.

16.3.1.3 Parameter management

16.3.1.3.1 Save_InterfaceParameters (BASICSETTINGS STORE)

Save actual interface parameters at controller.

16.3.1.3.2 Load_InterfaceParameters (BASICSETTINGS READ)

Load stored interface parameters into controller RAM.

16.3.1.3.3 Save_MeasureParameters (MEASSETTINGS STORE)

Save actual measurement parameters at controller.

16.3.1.3.4 Load_MeasureParameters (MEASSETTINGS READ)

Load stored measurement parameters into controller RAM.

At this command the controller may change output data after applying new setting. If first bit of [IP_AutomaticMode](#) is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

16.3.1.3.5 Set_Default (SETDEFAULT)

Reset the sensor to default settings.

At this command the sensor may change output data after applying default settings. If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DefaultType

SP_DefaultType

Direction: Down

Valid values:

- 0= Delete all settings and load the factory settings (ALL)
- 1= Delete all measuring settings (MEASSETTINGS)
- 2= Delete all basic settings (BASICSETTINGS)

Description: Specifies which settings should be reset.

16.3.2 Measurement

16.3.2.1 General

16.3.2.1.1 Set_Samplerate (DATARATE)

Set the datarate.

Parameter: double SP_Measrate

SP_Measrate

Direction: Down

Valid values:

- Minimum: 0.1
- Maximum: 100.0

Unit: Hz

Description: Datarate of measurement.

16.3.2.1.2 Get_Samplerate (DATARATE)

Get the datarate.

Parameter: double SA_Measrate

SA_Measrate

Direction: Up

Valid values:

- Minimum: 0.1
- Maximum: 100.0

Unit: Hz

Description: Datarate of measurement.

16.3.2.1.3 Get_Measure (GETMEASURE)

Return a measurement result for all or the given channel.

Parameter: int32_t SP_Channel

SP_Channel

Direction: Down

Valid values:

- 0= All channels
- 1..28= Channel 1 to 28

Description: Channel to process

Parameter: double SA_Color1_01..28	SA_Color1_01..28
Direction: Up	
Valid values:	
Minimum: -256.0	
Maximum: 256.0	
Description: Color value 1.	
Parameter: double SA_Color2_01..28	SA_Color2_01..28
Direction: Up	
Valid values:	
Minimum: -256.0	
Maximum: 256.0	
Description: Color value 2.	
Parameter: double SA_Color3_01..28	SA_Color3_01..28
Direction: Up	
Valid values:	
Minimum: -256.0	
Maximum: 256.0	
Description: Color value 3.	
Parameter: double SA_Temperature01..28	SA_Temperature01..28
Direction: Up	
Valid values:	
Minimum: 1000.0	
Maximum: 27000.0	
Unit: K	
Description: Color temperature	
Parameter: double SA_Wavelength01..28	SA_Wavelength01..28
Direction: Up	
Valid values:	
Minimum: 10.0	
Maximum: 15000.0	
Unit: nm	
Description: Wave length	
Parameter: double SA_Timestamp01..28	SA_Timestamp01..28
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Unit: ms	
Description: Timestamp	

16.3.2.2 Color processing

16.3.2.2.1 Set_ColorSpace (COLORSPACE)

Set the Colorspace used for color calculation.

Parameter: int32_t SP_ColorSpace	SP_ColorSpace
Direction: Down	

Valid values:

0= XYZ
 1= xyY
 2= Luv
 3= uvL
 4= RGB

Description: Colorspace to use

16.3.2.2.2 Get_ColorSpace (COLORSPACE)

Get the Colorspace used for color calculation.

Parameter: int32_t SA_ColorSpace

SA_ColorSpace

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
 0= XYZ
 1= xyY
 2= Luv
 3= uvL
 4= RGB

Description: Actually used colorspace

16.3.2.2.3 AutoGain (AUTOGAIN)

Determine a reasonable combination of integration time and gain to get the best possible value range.

Parameter: int32_t SP_Channel

SP_Channel

Direction: Down

Valid values:

0= All channels
 1..28= Channel 1 to 28

Description: Channel to process

16.3.2.2.4 Set_IntegrationTime (INTEGRATIONTIME)

Set the hardware integration time.

Parameter: int32_t SP_Channel

SP_Channel

Direction: Down

Valid values:

0= All channels
 1..28= Channel 1 to 28

Description: Channel to process

Parameter: int32_t SP_IntegrationTime

SP_IntegrationTime

Direction: Down

Valid values:

Minimum: 0
Maximum: 14

Description: Real integration time [in ms] is $2^{(0..14)}$, this is 1..16384 ms.

16.3.2.2.5 Get_IntegrationTime (INTEGRATIONTIME)

Get the hardware integration time.

Parameter: int32_t SP_Channel

SP_Channel

Direction: Down

Valid values:

0= All channels

1..28= Channel 1 to 28

Description: Channel to process

Parameter: int32_t SA_IntegrationTime01..28

SA_IntegrationTime01..28

Direction: Up

Valid values:

Minimum: 0

Maximum: 14

Description: Real integration time [in ms] is $2^{(0..14)}$, this is 1..16384 ms.

16.3.2.2.6 Set_Gain (GAIN)

Set the hardware gain level.

Parameter: int32_t SP_Channel

SP_Channel

Direction: Down

Valid values:

0= All channels

1..28= Channel 1 to 28

Description: Channel to process

Parameter: int32_t SP_Gain

SP_Gain

Direction: Down

Valid values:

Minimum: 0

Maximum: 11

Description: Real gain level is $2^{(0..11)}$, this is 1..2048.

16.3.2.2.7 Get_Gain (GAIN)

Get the hardware gain level.

Parameter: int32_t SP_Channel

SP_Channel

Direction: Down

Valid values:

0= All channels

1..28= Channel 1 to 28

Description: Channel to process

Parameter: int32_t SA_Gain01..28

SA_Gain01..28

Direction: Up

Valid values:

Minimum: 0

Maximum: 11

Description: Real gain level is $2^{(0..11)}$, this is 1..2048.

16.3.2.2.8 WhiteCorr (WHITECORR)

Make a white correction.

Parameter: int32_t SP_Channel SP_Channel
Direction: Down
Valid values:
 0= All channels
 1..28= Channel 1 to 28
Description: Channel to process

16.3.2.2.9 Set_WhiteCorrFactor (WHITECORR_FACTOR)

Set the white reference.

Parameter: int32_t SP_Channel SP_Channel
Direction: Down
Valid values:
 0= All channels
 1..28= Channel 1 to 28
Description: Channel to process

Parameter: double SP_WhiteCorrFactor1..3 SP_WhiteCorrFactor1..3
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 2.0
Description: White reference

16.3.2.2.10 Get_WhiteCorrFactor (WHITECORR_FACTOR)

Get the white reference.

Parameter: int32_t SP_Channel SP_Channel
Direction: Down
Valid values:
 0= All channels
 1..28= Channel 1 to 28
Description: Channel to process

Parameter: double SA_WhiteCorrFactor1_01..28 SA_WhiteCorrFactor1_-
01..28
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 2.0
Description: White reference for color 1.

Parameter: double SA_WhiteCorrFactor2_01..28 SA_WhiteCorrFactor2_-
01..28
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 2.0
Description: White reference for color 2.

Parameter: double SA_WhiteCorrFactor3_01..28 SA_WhiteCorrFactor3_01..28
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 2.0
Description: White reference for color 3.

16.3.2.2.11 DarkCorr (DARKCORR)

Make a dark correction.

Parameter: int32_t SP_Channel SP_Channel
Direction: Down
Valid values:
 0= All channels
 1..28= Channel 1 to 28
Description: Channel to process

16.3.2.2.12 Set_DarkCorrOffset (DARKCORR_OFFSET)

Set the dark reference.

Parameter: int32_t SP_Channel SP_Channel
Direction: Down
Valid values:
 0= All channels
 1..28= Channel 1 to 28
Description: Channel to process

Parameter: double SP_DarkCorrOffset1..3 SP_DarkCorrOffset1..3
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 3.40282e+38 (FLT_MAX)
Description: Dark reference

16.3.2.2.13 Get_DarkCorrOffset (DARKCORR_OFFSET)

Get the dark reference.

Parameter: int32_t SP_Channel SP_Channel
Direction: Down
Valid values:
 0= All channels
 1..28= Channel 1 to 28
Description: Channel to process

Parameter: double SA_DarkCorrOffset1_01..28 SA_DarkCorrOffset1_01..28
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 3.40282e+38 (FLT_MAX)
Description: Dark reference for color 1.

Parameter: double SA_DarkCorrOffset2_01..28	SA_DarkCorrOffset2_01..28
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Dark reference for color 2.	
Parameter: double SA_DarkCorrOffset3_01..28	SA_DarkCorrOffset3_01..28
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 3.40282e+38 (FLT_MAX)	
Description: Dark reference for color 3.	

16.3.2.2.14 Set_WhiteLED (WHITELED)

Set the level of the white reference LED.

Parameter: int32_t SP_WhiteLED	SP_WhiteLED
Direction: Down	
Valid values:	
Minimum: 0= off	
Maximum: 15= maximum	
Description: Intensity of the LED	

16.3.2.2.15 Get_WhiteLED (WHITELED)

Get the level of the white reference LED.

Parameter: int32_t SA_WhiteLED	SA_WhiteLED
Direction: Up	
Valid values:	
Minimum: 0= off	
Maximum: 15= maximum	
Description: Intensity of the LED	

16.3.2.3 Measurement value processing

16.3.2.3.1 Set_Averaging (AVERAGING)

Set data averaging at sensor.

Parameter: int32_t SP_Channel	SP_Channel
Direction: Down	
Valid values:	
0= All channels	
1..28= Channel 1 to 28	
Description: Channel to process	

Parameter: int32_t SP_Averaging SP_Averaging
Direction: Down
Valid values:
Minimum: 1
Maximum: 2147483647 (INT32_MAX)
Description: Number of values for moving average.

16.3.2.3.2 Get_Averaging (AVERAGING)

Get data averaging at sensor.

Parameter: int32_t SP_Channel SP_Channel
Direction: Down
Valid values:
 0= All channels
 1..28= Channel 1 to 28
Description: Channel to process

Parameter: int32_t SA_Averaging01..28 SA_Averaging01..28
Direction: Up
Valid values:
Minimum: 1
Maximum: 2147483647 (INT32_MAX)
Description: Number of values for moving average.

16.3.3 Data output

16.3.3.1 General

16.3.3.1.1 Set_DataOutInterface (OUTPUT)

Set the active interface for data output.

At this command the sensor may change output data automatically. If first bit of **IP_AutomaticMode** is set (1), [Get_OutputInfo](#) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DataOutInterface SP_DataOutInterface
Direction: Down
Valid values:
 0= None
 1= On (RS232/RS422/USB)
Description: Active interface for data output.

16.3.3.1.2 Get_DataOutInterface (OUTPUT)

Get the active interface for data output.

Parameter: int32_t SA_DataOutInterface SA_DataOutInterface
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= None
 1= On (RS232/RS422/USB)
Description: Active interface for data output.

16.3.3.2 Selected measurement values

16.3.3.2.1 Set_Output (OUT)

Set the data to be output at RS232/RS422/USB interface. For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputCh01..28

SP_OutputCh01..28

Direction: Down

Valid values:

0= no
1= yes

Description: Specify if channel is transmitted.

Parameter: int32_t SP_OutputTemperature

SP_OutputTemperature

Direction: Down

Valid values:

0= no
1= yes

Description: Specify if temperature is transmitted.

Parameter: int32_t SP_OutputWavelength

SP_OutputWavelength

Direction: Down

Valid values:

0= no
1= yes

Description: Specify if wavelength is transmitted.

Parameter: int32_t SP_OutputTimestamp

SP_OutputTimestamp

Direction: Down

Valid values:

0= no
1= yes

Description: Specify if timestamp is transmitted.

16.3.3.2.2 Get_Output (OUT)

Get the data which is output at RS232/RS422/USB interface.

Parameter: int32_t SA_OutputCh01..28

SA_OutputCh01..28

Direction: Up

Valid values:

0= no
1= yes

Description: Specify if channel is transmitted.

Parameter: int32_t SA_OutputTemperature

SA_OutputTemperature

Direction: Up

Valid values:

0= no
1= yes

Description: Specify if temperature is transmitted.

Parameter: int32_t SA_OutputWavelength	SA_OutputWavelength
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if wavelength is transmitted.	
Parameter: int32_t SA_OutputTimestamp	SA_OutputTimestamp
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if timestamp is transmitted.	

16.3.4 Internal commands

16.3.4.1 Get_FirmwareVersion

Retrieve firmware version from sensor.
This is an internal command. It should not be used by the customer.

Parameter: String CA_FirmwareVersion	CA_FirmwareVersion
Direction: Up	
Description: Firmware version	

16.3.4.2 Prepare_UpdateFirmware

Prepares a firmware update at sensor.
This is an internal command. It should not be used by the customer.

Parameter: Binary data XP_FirmwareFile	XP_FirmwareFile
Direction: Down	
Description: Firmware file	

Parameter: String CA_FileName	CA_FileName
Direction: Up	
Description: Internal name of firmware file	

Parameter: String CA_Date	CA_Date
Direction: Up	
Description: Date of firmware file	

Parameter: String CA_ArticleNumber	CA_ArticleNumber
Direction: Up	
Description: Article number of destination device	

Parameter: String CA_SerialNumber	CA_SerialNumber
Direction: Up	
Description: Serial number of destination device	

16.3.4.3 Start_UpdateFirmware

Start firmware update at sensor.

This is an internal command. It should not be used by the customer.

16.3.4.4 Get_UpdateFirmwareProgress

Update firmware version at sensor.

Attention! This function can takes up to 1 minute. This is an internal command. It should not be used by the customer.

Parameter: double CA_Progress CA_Progress

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 1.0

Description: Progress of firmware update

Parameter: String CA_Description CA_Description

Direction: Up

Description: Current state of firmware update.

Parameter: int32_t CA_Result CA_Result

Direction: Up

Valid values:

0= Failed

1= Success

Description: Result of firmware update (only available at end of update).

Parameter: int32_t CA_Finished CA_Finished

Direction: Up

Valid values:

-1= Firmware update is not prepared, call Prepare_UpdateFirmware first

0= No

1= Yes

Description: Tell if firmware update is in progress.

16.3.4.5 Generate_Firmware

This is an internal command. It should not be used by the customer.

17 Commands for Interfaces

17.1 Commands for sensors on MEBus

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- RS232 (RS485/USB converter).

If first bit of `IP_AutomaticMode` is set (1), MEDAQLib calls automatically sensor command `Read_AllBlocks` after `OpenSensor`.

Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate and to assign values.

Meaning of raw and scaled values (function `Poll` and `TransferData`):

- Raw values are as it comes directly from sensor, range is depending on data format.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command `Read_AllBlocks`).

The values of selected channels are filled in the arrays one after another. Each array always starts with first selected channel.

17.1.1 User Level

17.1.1.1 Logout

Change user level to none.

17.1.1.2 Login

Change user level depending on password.

Parameter: String `SP_Password` `SP_Password`
Direction: Down
Description: Valid password to login.

17.1.1.3 Get_UserLevel

Retrieve actual user level.

Parameter: int32_t `SA_UserLevel` `SA_UserLevel`
Direction: Up
Valid values:
Minimum: 0
Maximum: 254
Description: Actual user level.

17.1. Commands for sensors on MEBus

Parameter: String SA_ActiveUserName	SA_ActiveUserName
Direction: Up	
Description: Name of current user level.	
Parameter: int32_t SA_PasswordChangeable	SA_PasswordChangeable
Direction: Up	
Valid values:	
0= No	
1= Yes	
Description: Password is changeable.	

17.1.1.4 Set_Password

Change the password for login.

Parameter: String SP_OldPassword	SP_OldPassword
Direction: Down	
Description: Old password.	
Parameter: String SP_NewPassword	SP_NewPassword
Direction: Down	
Description: New password.	

17.1.2 Measurement

17.1.2.1 Set_Samplerate

Set the samplerate for data acquisition.

Parameter: double SP_Measrate	SP_Measrate
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 65536000000.0	
Unit: Hz	
Description: Desired samplerate	
Parameter: double SA_Measrate	SA_Measrate
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 65536000000.0	
Unit: Hz	
Description: Real samplerate at sensor	

17.1.2.2 Get_Samplerate

Get the current samplerate.

Parameter: double SA_Measrate	SA_Measrate
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 65536000000.0	
Unit: Hz	
Description: Real samplerate at sensor	

17.1.2.3 Set_Trigger

Activate/disable the trigger.

Parameter: int32_t SP_TrMode

SP_TrMode

Direction: Down

Valid values:

- 0= Off
- 1= Rising edge
- 2= High level
- 3= Gate at rising edge
- 4= Falling edge
- 5= Low level
- 6= Gate at falling edge

Description: Trigger active/disabled.

17.1.2.4 Get_Trigger

Retrieve the trigger mode.

Parameter: int32_t SA_TrMode

SA_TrMode

Direction: Up

Valid values:

- 0= Off
- 1= Rising edge
- 2= High level
- 3= Gate at rising edge
- 4= Falling edge
- 5= Low level
- 6= Gate at falling edge

Description: Trigger active/disabled.

17.1.2.5 Set_Averaging

Set data averaging at sensor.

Parameter: int32_t SP_AveragingType

SP_AveragingType

Direction: Down

Valid values:

- 0= off
- 1= Moving average
- 2= Mean (arithmetic)
- 3= Median
- 4= Dynamic noise rejection
- 5= FIR filter
- 6= Kalman filter

Description: Averaging type.

Parameter: int32_t SP_AveragingValue SP_AveragingValue
Direction: Down
Valid values:
Minimum: 0
Maximum: 2147483647 (INT32_MAX)
Description: Averaging number (depending on averaging type).

17.1.2.6 Get_Averaging

Retrieve the averaging type at sensor.

Parameter: int32_t SA_AveragingType SA_AveragingType
Direction: Up
Valid values:
 0= off
 1= Moving average
 2= Mean (arithmetic)
 3= Median
 4= Dynamic noise rejection
 5= FIR filter
 6= Kalman filter
Description: Averaging type at sensor.

Parameter: int32_t SA_AveragingValue SA_AveragingValue
Direction: Up
Valid values:
Minimum: 0
Maximum: 2147483647 (INT32_MAX)
Description: Averaging number (depending on averaging type).

17.1.2.7 Get_Measure

Retrieve a data block from controller. To get the values from MEDAQLib, use the functions [Poll](#) or [TransferData](#).

Parameter: double SA_State SA_State
Direction: Up
Valid values:
Minimum: 0
Maximum: 4294967295.0 (UINT32_MAX)
Description: State of received telegram (status).

Parameter: double SA_PacketCounter SA_PacketCounter
Direction: Up
Valid values:
Minimum: 0
Maximum: 4294967295.0 (UINT32_MAX)
Description: Packet counter of received telegram (zaehler_messwerte).

Parameter: int32_t SA_FrameCount SA_FrameCount
Direction: Up
Valid values:
Minimum: 0
Maximum: 255
Description: Frame counter of received telegram (zaehler_messwertarray).

17.1.2.8 Get_AlternateMeasure

Retrieve an alternate data block from controller.

Parameter: int32_t SP_AlternateBlockIdx SP_AlternateBlockIdx
Direction: Down
Valid values:
Minimum: 129
Maximum: 143
Description: Alternate block index to read (starting with 0x81).

Parameter: String SA_AlternateMeasure SA_AlternateMeasure
Direction: Up
Valid values:
 Measure data as string, each value in a single line.
Description: Measured data.

17.1.3 Data output

17.1.3.1 Set_Range

Write the measurement range to sensor.

Parameter: int32_t SP_Chан SP_Chан
Direction: Down
Valid values:
Minimum: 0
Maximum: 31
Description: Channel number.

Parameter: int32_t SP_PageNo SP_PageNo
Direction: Down
Valid values:
Minimum: 1
Maximum: 255
Description: Sensor page number.

Parameter: double SP_Range SP_Range
Direction: Down
Valid values:
Minimum: -3.40282e+38 (-FLT_MAX)
Maximum: 3.40282e+38 (FLT_MAX)
Unit: μm
Description: Range of sensor.

17.1.4 Interfaces

17.1.4.1 Test_Baudrate

Test if a baudrate is valid for connected sensor.

Parameter: int32_t SP_DesiredBaudrate SP_DesiredBaudrate

Direction: Down

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Unit: Baud

Description: Baudrate to be test.

Parameter: int32_t SA_NegotiatedBaudrate SA_NegotiatedBaudrate

Direction: Up

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Unit: Baud

Description: Nearest possible baudrate to the desired baudrate.

17.1.4.2 Set_Baudrate

Set baudrate of sensor for serial RS485 communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Parameter: int32_t SP_SensorBaudrate SP_SensorBaudrate

Direction: Down

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Unit: Baud

Description: Baudrate of sensor.

Parameter: int32_t SP_Store SP_Store

Direction: Down

Valid values:

0= Temporary (in RAM)

1= Permanent (in Flash)

Description: Specify if the baudrate should be changed persistently.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate

Direction: Up

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Unit: Baud

Description: Real baudrate at sensor.

17.1.4.3 Get_Baudrate

Get baudrate of controller for serial RS485 communication.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate
Direction: Up
Valid values:
Minimum: 0
Maximum: 2147483647 (INT32_MAX)
Unit: Baud
Description: Real baudrate at sensor.

17.1.4.4 Set_SensorAddress

Changes the address at sensor which is used to communicate over RS485 bus.

Parameter: int32_t SP_SensorAddress SP_SensorAddress
Direction: Down
Valid values:
Minimum: 1
Maximum: 126
Description: Address of sensor.

17.1.5 Information

17.1.5.1 Get_SensorInfo

Retrieve information about the connected sensor.

Parameter: int32_t SP_Chан SP_Chан
Direction: Down
Valid values:
Minimum: 0
Maximum: 31
Description: Channel number.

Parameter: int32_t SP_PageNo SP_PageNo
Direction: Down
Valid values:
Minimum: 1
Maximum: 255
Description: Sensor page number.

Parameter: String SA_ArticleNumber SA_ArticleNumber
Direction: Up
Valid values:
 Numeric value
Description: Article number of the connected sensor.

Parameter: String SA_SensorName SA_SensorName
Direction: Up
Description: Name of the connected sensor.

Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the connected sensor.	
Parameter: double SA_Offset	SA_Offset
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Unit: See parameter SA_Unit	
Description: Offset of the connected sensor.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Unit: See parameter SA_Unit	
Description: Range of connected sensor.	
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description: Unit of parameter SA_Offset and SA_Range, e.g. μm or mm.	

17.1.5.2 Get_ChannelInfo

Retrieve information about a sensor channel.

Parameter: int32_t SP_Chан	SP_Chан
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 31	
Description: Channel number.	
Parameter: int32_t SP_PageNo	SP_PageNo
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 255	
Description: Sensor page number.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the connected sensor.	
Parameter: String SA_SensorName	SA_SensorName
Direction: Up	
Description: Name of the connected sensor.	

17.1. Commands for sensors on MEBus

Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the connected sensor.	
Parameter: double SA_Offset	SA_Offset
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Unit: See parameter SA_Unit	
Description: Offset of the connected sensor.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Unit: See parameter SA_Unit	
Description: Range of connected sensor.	
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description: Unit of parameter SA_Offset and SA_Range, e.g. μm or mm.	

17.1.5.3 Get_ChannelInfos

Retrieve information about all channels.

Parameter: int32_t SP_Read	SP_Read
Direction: Down	
Valid values:	
0= Only use MEDAQLib internal information	
1= Read information from sensor if not available	
Description: Use cached information.	
Parameter: int32_t SP_PageNo	SP_PageNo
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 255	
Default: 0	
Description: Sensor page number. 0 means read all available pages.	
Parameter: int32_t SA_ChannelCount	SA_ChannelCount
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 8160	
Description: Number of channels in the table. All following parameters exists from 1 to this number, e.g. SA_SensorIndex1, SA_SensorIndex2, ...	

Parameter: int32_t SA_SensorIndex1..x	SA_SensorIndex1..x
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 31	
Description: Sensor block index.	
Parameter: int32_t SA_SensorPage1..x	SA_SensorPage1..x
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 255	
Description: Sensor block page.	
Parameter: String SA_ArticleNumber1..x	SA_ArticleNumber1..x
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the connected sensor.	
Parameter: String SA_SensorName1..x	SA_SensorName1..x
Direction: Up	
Description: Name of the connected sensor.	
Parameter: String SA_SerialNumber1..x	SA_SerialNumber1..x
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the connected sensor.	
Parameter: double SA_Offset1..x	SA_Offset1..x
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Unit: See parameter SA_Unit	
Description: Offset of the connected sensor.	
Parameter: double SA_Range1..x	SA_Range1..x
Direction: Up	
Valid values:	
Minimum: -3.40282e+38 (-FLT_MAX)	
Maximum: 3.40282e+38 (FLT_MAX)	
Unit: See parameter SA_Unit	
Description: Range of connected sensor.	
Parameter: String SA_Unit1..x	SA_Unit1..x
Direction: Up	
Description: Unit of parameter SA_Offset1..x and SA_Range1..x, e.g. μm or mm.	

17.1.5.4 Get_ControllerInfo

Retrieve information about the controller part of sensor.

Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Description: Article number of the controller part of sensor.	
Parameter: String SA_ControllerName	SA_ControllerName
Direction: Up	
Description: Name of the controller part of sensor.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the controller part of sensor.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the controller part of sensor.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the controller part of sensor.	

17.1.5.5 Get_DiagnosticInfo

Read diagnostic information from sensor.

Parameter: String SA_DiagnosticInfo	SA_DiagnosticInfo
Direction: Up	
Description: Diagnostic information string.	

17.1.5.6 Get_ConfigDescription

Read config description table from sensor.

Parameter: int32_t CP_UseCachedParameter	CP_UseCachedParameter
Direction: Down	
Valid values:	
0= Read information from sensor in any case	
1= Use MEDAQLib internal information if available	
Default: 0	
Description: Use cached information.	
Parameter: String SA_ConfigDescription	SA_ConfigDescription
Direction: Up	
Description: Config description table.	

17.1.5.7 Set_ConfigParameter

Set specific parameter value to config description table.

Parameter: String SP_ParameterName	SP_ParameterName
Direction: Down	
Description: Parameter name.	
Parameter: int32_t SP_ParameterID	SP_ParameterID
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Parameter ID. It only is used if parameter name is empty.	
Parameter: String SP_Value	SP_Value
Direction: Down	
Valid values:	
Value in string representation.	
Description: Value to set.	

17.1.5.8 Get_ConfigParameter

Read specific parameter value from config description table.

Parameter: String SP_ParameterName	SP_ParameterName
Direction: Down	
Description: Parameter name.	
Parameter: int32_t SP_ParameterID	SP_ParameterID
Direction: Down	
Valid values:	
Minimum: 0	
Maximum: 65535	
Description: Parameter ID. It only is used if parameter name is empty.	
Parameter: int32_t CP_UseCachedParameter	CP_UseCachedParameter
Direction: Down	
Valid values:	
0= Read information from sensor in any case	
1= Use MEDAQLib internal information if available	
Default: 0	
Description: Use cached information.	
Parameter: String SA_Value	SA_Value
Direction: Up	
Valid values:	
Value in string representation.	
Description: Resulting value.	

17.1.5.9 Read_AllBlocks

Tell MEDAQLib to read all sensor information.

17.2 Commands for ETH_IF1032

See interface manual for detailed description of interface commands.

This interface supports following interfaces:

- [TCP/IP](#) (native).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_Status](#), [Get_RawDataRanges](#) and [Get_ChannelInfos](#) after [OpenSensor](#). Otherwise, you have to call it manually to allow MEDAQLib to calculate datarate and to assign values.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from interface, either from -2147483648 (INT32_MIN) to 2147483647 (INT32_MAX) or from 0 to 4294967295 (UINT32_MAX) or from -3.402823466e+38 (-FLT_MAX) to 3.402823466e+38 (FLT_MAX).
- Scaled values are scaled using sensor range (if known by MEDAQLib, use interface command [Get_RawDataRanges](#) and [Get_ChannelInfos](#)).

The values of selected channels are filled in the arrays one after another. Each array always starts with first selected channel.

17.2.1 User Level

17.2.1.1 Logout (LGO)

Change user level to user at web interface.

17.2.1.2 Login (LGI)

Change user level to setup at web interface.

Parameter: String SP_Password

SP_Password

Direction: Down

Description: Valid password to login.

17.2.1.3 Set_Password (PWD)

Change the password for login.

Parameter: String SP_OldPassword

SP_OldPassword

Direction: Down

Description: Old password.

Parameter: String SP_NewPassword SP_NewPassword
Direction: Down
Description: New password.

17.2.2 Measurement

17.2.2.1 Set_SampleTime (STI)

Set the sample time for data acquisition.

Parameter: double SP_SampleTime SP_SampleTime
Direction: Down
Valid values:
 Minimum: 0.0
 Maximum: 4294967040.0
Unit: μs
Description: Desired sample time

Parameter: double SA_SampleTime SA_SampleTime
Direction: Up
Valid values:
 Minimum: 0.0
 Maximum: 4294967040.0
Unit: μs
Description: Real sample time at interface

17.2.2.2 Get_SampleTime (STI?)

Get the current sample time.

Parameter: double SA_SampleTime SA_SampleTime
Direction: Up
Valid values:
 Minimum: 0.0
 Maximum: 4294967040.0
Unit: μs
Description: Real sample time at interface

17.2.2.3 Set_Trigger (TRG)

Activate/disable the trigger at interface.

Parameter: int32_t SP_TrgMode SP_TrgMode
Direction: Down
Valid values:
 0= Off
 1= Rising edge
 2= High level
 3= Gate at rising edge
 4= Falling edge
 5= Low level
 6= Gate at falling edge
Description: Trigger active/disabled.

17.2.2.4 Get_Trigger (TRG?)

Retrieve the trigger mode at interface.

Parameter: int32_t SA_TrMode

SA_TrMode

Direction: Up

Valid values:

- 0= Off
- 1= Rising edge
- 2= High level
- 3= Gate at rising edge
- 4= Falling edge
- 5= Low level
- 6= Gate at falling edge

Description: Trigger active/disabled.

17.2.2.5 Set_AvrType (AVT)

Set the averaging type at interface.

Parameter: int32_t SP_AvrType

SP_AvrType

Direction: Down

Valid values:

- 0= off
- 1= Moving average
- 2= Mean (arithmetic)
- 3= Median
- 4= Dynamic noise rejection
- 5= FIR filter
- 6= Kalman filter

Description: Averaging type.

17.2.2.6 Get_AvrType (AVT?)

Retrieve the averaging type at interface.

Parameter: int32_t SA_AvrType

SA_AvrType

Direction: Up

Valid values:

- 0= off
- 1= Moving average
- 2= Mean (arithmetic)
- 3= Median
- 4= Dynamic noise rejection
- 5= FIR filter
- 6= Kalman filter

Description: Averaging type at interface.

17.2.2.7 Set_AvrNbr (AVN)

Set the averaging number at interface.

Parameter: int32_t SP_AvrNbr SP_AvrNbr
Direction: Down
Valid values:
Minimum: 0
Maximum: 2147483647 (INT32_MAX)
Description: Averaging number.

17.2.2.8 Get_AvrNbr (AVN?)

Retrieve the averaging number at interface.

Parameter: int32_t SA_AvrNbr SA_AvrNbr
Direction: Up
Valid values:
Minimum: 0
Maximum: 2147483647 (INT32_MAX)
Description: Averaging number at interface.

17.2.3 Data output

17.2.3.1 ChannelStatus (CHS)

Retrieve the available channels at interface.

Parameter: int32_t SA_ChExist1..32 SA_ChExist1..32
Direction: Up
Valid values:
 0 = Channel not available
 1 = Measured channel
Description: Channel 1 to 32 is available at interface.

17.2.3.2 Get_RawDataRange (MDF)

Read the raw data range for a channel from interface. The range is used by MEDAQLib for scaling raw data.

Parameter: int32_t SP_Chан SP_Chан
Direction: Down
Valid values:
Minimum: 1
Maximum: 32
Description: Channel to read the raw data range for.

Parameter: double SA_RawRangeMin SA_RawRangeMin

Direction: Up

Valid values:

Minimum: -2147483648.0 (INT32_MIN)

Maximum: 2147483647.0 (INT32_MAX)

Description: Minimum raw data range of channel.

Parameter: double SA_RawRangeMax SA_RawRangeMax

Direction: Up

Valid values:

Minimum: -2147483648.0 (INT32_MIN)

Maximum: 2147483647.0 (INT32_MAX)

Description: Maximum raw data range of channel.

17.2.3.3 Get_RawDataRanges

Calls the sensor command Get_RawDataRange for any requested channel.

Parameter: int32_t SP_Complete SP_Complete

Direction: Down

Valid values:

0 = FALSE

1 = TRUE

Description: Specifies if any possible channel should be requested or only known channels (from former call to [ChannelStatus](#) or [Get_Status](#)).

Parameter: double SA_RawRangeMin1..32 SA_RawRangeMin1..32

Direction: Up

Valid values:

Minimum: -2147483648.0 (INT32_MIN)

Maximum: 2147483647.0 (INT32_MAX)

Description: Minimum raw data range of channel 1 to 32.

Parameter: double SA_RawRangeMax1..32 SA_RawRangeMax1..32

Direction: Up

Valid values:

Minimum: -2147483648.0 (INT32_MIN)

Maximum: 2147483647.0 (INT32_MAX)

Description: Maximum raw data range of channel 1 to 32.

17.2.4 Interfaces

17.2.4.1 Set_DataPort (SDP)

Set the TCP/IP data port at interface.

Parameter: int32_t SP_DataPort SP_DataPort

Direction: Down

Valid values:

Minimum: 1024

Maximum: 65535

Description: TCP/IP data port at interface.

17.2.4.2 Get_DataPort (GDP)

Retrieve the TCP/IP data port from interface.

Parameter: int32_t SA_DataPort

SA_DataPort

Direction: Up

Valid values:

Minimum: 1024

Maximum: 65535

Description: TCP/IP data port at interface.

17.2.4.3 Set_IPConfiguration (IPS)

Set the IP configuration at interface.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_DHCPEnabled

SP_DHCPEnabled

Direction: Down

Valid values:

0= FALSE

1= TRUE

Description: Specify if interface should use a static IP address or ask for IP at DHCP server (dynamic IP address).

Parameter: String SP_Address

SP_Address

Direction: Down

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the interface. This parameter is only evaluated on static IP assignment.

Parameter: String SP_SubnetMask

SP_SubnetMask

Direction: Down

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the interface. This parameter is only evaluated on static IP assignment.

Parameter: String SP_Gateway

SP_Gateway

Direction: Down

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: The default gateway must be specified if the interface should communicate with peers in foreign subnets. This parameter is only evaluated on static IP assignment.

17.2.4.4 Get_IPConfiguration (IPS?)

Get the IP configuration at interface.

Parameter: int32_t SA_DHCPEnabled

SA_DHCPEnabled

Direction: Up

Valid values:

0= FALSE

1= TRUE

Description: Get settings if interface should use a static IP address ask for IP at DHCP server (dynamic IP address).

Parameter: String SA_Address	SA_Address
Direction: Up	
Valid values:	Valid IP address in form of xxx.xxx.xxx.xxx
Description:	IP address of the interface. If DHCP is enabled it returns the currently assigned IP address.
Parameter: String SA_SubnetMask	SA_SubnetMask
Direction: Up	
Valid values:	Valid network mask (e.g. 255.255.255.0 for a Class C network)
Description:	Network mask of the interface. If DHCP is enabled it returns the currently assigned network mask.
Parameter: String SA_Gateway	SA_Gateway
Direction: Up	
Valid values:	Valid IP address of default gateway in form of xxx.xxx.xxx.xxx
Description:	Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

17.2.4.5 Set_EthernetMode (IFC)

Switches ethernet mode between Ethernet and Ethercat. The interface must be rebooted to apply this setting.

Parameter: int32_t SP_EthernetMode	SP_EthernetMode
Direction: Down	
Valid values:	0= Ethernet 1= Ethercat
Description:	Ethernet mode.

17.2.4.6 Get_EthernetMode (IFC?)

Get ethernet mode of interface.

Parameter: int32_t SA_EthernetMode	SA_EthernetMode
Direction: Up	
Valid values:	0= Ethernet 1= Ethercat
Description:	Ethernet mode.

17.2.4.7 Set_AppLangauge (LNG)

Set language of web interface at interface.

Parameter: int32_t SP_ApplicationLanguage	SP_ApplicationLanguage
Direction: Down	
Valid values:	0= System 1= English 2= German
Description:	Language of web interface.

17.2.4.8 Get_AppLanguage (LNG?)

Get language of web interface from interface.

Parameter: int32_t SA_ApplicationLanguage

SA_ApplicationLanguage

Direction: Up

Valid values:

0= System

1= English

2= German

Description: Language of web interface.

17.2.5 Sensor interface

17.2.5.1 Set_SensorInterface (SIF)

Selects the sensor interface of the module.

Parameter: int32_t SP_SensorInterface

SP_SensorInterface

Direction: Down

Valid values:

0= Analog

1= not assigned currently

2= RS232 (not available in EthIF1032)

3= RS485

Description: Sensor interface of the module

17.2.5.2 Get_SensorInterface (SIF?)

Get the sensor interface of the module.

Parameter: int32_t SA_SensorInterface

SA_SensorInterface

Direction: Up

Valid values:

0= Analog

1= not assigned currently

2= RS232 (not available in the IF1032/ETH)

3= RS485

Description: Sensor interface of the module

17.2.5.3 Set_SensorBaudrate (SBR)

Changes the Baudrate of the RS485 interface.

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

Minimum: 9200

Maximum: 6250000

Description: Baudrate of the RS485 interface

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate
Direction: Up
Valid values:
 Minimum: 9200
 Maximum: 6250000
Description: Real baudrate of the RS485 interface

17.2.5.4 Get_SensorBaudrate (SBR?)

Get the Baudrate of the RS485 interface.

Parameter: int32_t SA_SensorBaudrate SA_SensorBaudrate
Direction: Up
Valid values:
 Minimum: 9200
 Maximum: 6250000
Description: Baudrate of the RS485 interface

17.2.5.5 Set_SensorAddress (SAD)

Changes the address used to communicate with the RS485.

Parameter: int32_t SP_DeviceInstance SP_DeviceInstance
Direction: Down
Valid values:
 Minimum: 1
 Maximum: 1
Description: Device instance of the RS485 device

Parameter: int32_t SP_SensorAddress SP_SensorAddress
Direction: Down
Valid values:
 Minimum: 2
 Maximum: 126
Description: Address used to communicate with the RS485

17.2.5.6 Get_SensorAddress (SAD?)

Get the address used to communicate with the RS485.

Parameter: int32_t SP_DeviceInstance SP_DeviceInstance
Direction: Down
Valid values:
 Minimum: 1
 Maximum: 1
Description: Device instance of the RS485 device

Parameter: int32_t SA_SensorAddress SA_SensorAddress
Direction: Up
Valid values:
 Minimum: 2
 Maximum: 126
Description: Address used to communicate with the RS485

17.2.5.7 Set_ConfigParameter (PID)

Set sensor specific parameter value to config description table.

Parameter: int32_t SP_ParameterID SP_ParameterID

Direction: Down

Valid values:

Minimum: 0

Maximum: 65535

Description: Parameter ID.

Parameter: String SP_Value SP_Value

Direction: Down

Valid values:

Value in string representation.

Description: Value of parameter. See sensor documentation for further information.

17.2.5.8 Get_ConfigParameter (PID?)

Get sensor specific parameter.

Parameter: int32_t SP_ParameterID SP_ParameterID

Direction: Down

Valid values:

Minimum: 0

Maximum: 65535

Description: Parameter ID.

Parameter: String SA_Value SA_Value

Direction: Up

Valid values:

Value in string representation.

Description: Value of parameter. See sensor documentation for further information.

17.2.5.9 Set_AnalogRange (ARA)

Set the measuring range of the analog inputs. This changes only the scaling, not the actual input range.

Parameter: int32_t SP_AnalogChannel SP_AnalogChannel

Direction: Down

Valid values:

Minimum: 1

Maximum: 4

Description: Analog channel number

Parameter: double SP_AnalogRange SP_AnalogRange

Direction: Down

Valid values:

Minimum: -4294967040.0

Maximum: 4294967040.0

Description: Analog range value

17.2.5.10 Get_AnalogRange (ARA?)

Get the measuring range of the analog inputs.

Parameter: int32_t SP_AnalogChannel SP_AnalogChannel

Direction: Down

Valid values:

Minimum: 1

Maximum: 4

Description: Analog channel number

Parameter: double SA_AnalogRange SA_AnalogRange

Direction: Up

Valid values:

Minimum: -4294967040.0

Maximum: 4294967040.0

Description: Analog range value

17.2.5.11 Set_AnalogOffset (AOF)

Set the measuring range offset of the analog inputs. This only changes the scaling, not the actual input range.

Parameter: int32_t SP_AnalogChannel SP_AnalogChannel

Direction: Down

Valid values:

Minimum: 1

Maximum: 4

Description: Analog channel number

Parameter: double SP_AnalogOffset SP_AnalogOffset

Direction: Down

Valid values:

Minimum: -4294967040.0

Maximum: 4294967040.0

Description: Analog offset value

17.2.5.12 Get_AnalogOffset (AOF?)

Get the measuring range offset of the analog inputs.

Parameter: int32_t SP_AnalogChannel SP_AnalogChannel

Direction: Down

Valid values:

Minimum: 1

Maximum: 4

Description: Analog channel number

Parameter: double SA_AnalogOffset SA_AnalogOffset

Direction: Up

Valid values:

Minimum: -4294967040.0

Maximum: 4294967040.0

Description: Analog offset value

17.2.5.13 Set_AnalogUnit (AUN)

Set the unit of the analog inputs. This only changes the scaling, not the actual input range.

Parameter: int32_t SP_AnalogChannel

SP_AnalogChannel

Direction: Down

Valid values:

Minimum: 1

Maximum: 4

Description: Analog channel number

Parameter: int32_t SP_AnalogUnit

SP_AnalogUnit

Direction: Down

Valid values:

0= m

1= mm

2= μ m

3= V

4= digit

5= mA

6= °C

7= %

8= Hz

9= m/s

10= m/s^2

11= g (gravity)

12= °

13= °/s

14= rad

15= rad/s

16= -

17= °F

18= inch

19= mil

20= s

21= ms

22= μ s

Description: Analog unit

17.2.5.14 Get_AnalogUnit (AUN?)

Get the unit of the analog inputs.

Parameter: int32_t SP_AnalogChannel

SP_AnalogChannel

Direction: Down

Valid values:

Minimum: 1

Maximum: 4

Description: Analog channel number

Parameter: int32_t SA_AnalogUnit SA_AnalogUnit

Direction: Up

Valid values:

- 0= m
- 1= mm
- 2= μ m
- 3= V
- 4= digit
- 5= mA
- 6= °C
- 7= %
- 8= Hz
- 9= m/s
- 10= m/s^2
- 11= g (gravity)
- 12= °
- 13= °/s
- 14= rad
- 15= rad/s
- 16= -
- 17= °F
- 18= inch
- 19= mil
- 20= s
- 21= ms
- 22= μ s

Description: Analog unit

17.2.5.15 Set_AnalogMathFunction (AMF)

Set mathematic function for analog inputs at controller.

The result of the math function is transmitted like normal sensor data at a channel 4.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_AnalogMathFunctionEnable

SP_AnalogMathFunctionEnable

Direction: Down

Valid values:

- 0= Disabled
- 1= Enabled

Description: Specify if analog math function is enabled.

Parameter: double SP_AnalogOffset

SP_AnalogOffset

Direction: Down

Valid values:

- Minimum: -4294967040.0
- Maximum: 4294967040.0

Description: Offset to be added to result.

Parameter: double SP_AnalogFactor1..3

SP_AnalogFactor1..3

Direction: Down

Valid values:

- Minimum: -4294967040.0
- Maximum: 4294967040.0

Description: Multiplication factor for analog channel 1 to 3.

17.2.5.16 Get_AnalogMathFunction (AMF?)

Get mathematic function for analog inputs at controller.

Parameter: int32_t SA_AnalogMathFunctionEnable

SA_AnalogMathFunctionEnable

Direction: Up

Valid values:

0= Disabled

1= Enabled

Description: Specify if analog math function is enabled.

Parameter: double SA_AnalogOffset

SA_AnalogOffset

Direction: Up

Valid values:

Minimum: -4294967040.0

Maximum: 4294967040.0

Description: Offset to be added to result.

Parameter: double SA_AnalogFactor1..3

SA_AnalogFactor1..3

Direction: Up

Valid values:

Minimum: -4294967040.0

Maximum: 4294967040.0

Description: Multiplication factor for analog channel 1 to 3.

17.2.6 Information

17.2.6.1 Get_Status (STS)

Retrieve detailed information about the interface.

Parameter: double SA_SampleTime

SA_SampleTime

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967040.0

Unit: μ s

Description: Real sample time at interface

Parameter: int32_t SA_AvrType

SA_AvrType

Direction: Up

Valid values:

0= off

1= Moving average

2= Mean (arithmetic)

3= Median

4= Dynamic noise rejection

5= FIR filter

6= Kalman filter

Description: Averaging type at interface.

Parameter: int32_t SA_AvrNbr SA_AvrNbr

Direction: Up

Valid values:

Minimum: 0

Maximum: 2147483647 (INT32_MAX)

Description: Averaging number at interface.

Parameter: int32_t SA_ChExist1..32 SA_ChExist1..32

Direction: Up

Valid values:

0= Channel not available

1= Measured channel

Description: Channel 1 to 32 is available at interface.

Parameter: int32_t SA_TrgMode SA_TrgMode

Direction: Up

Valid values:

0= Off

1= Rising edge

2= High level

3= Gate at rising edge

4= Falling edge

5= Low level

6= Gate at falling edge

Description: Trigger active/disabled.

17.2.6.2 Get_Version (VER)

Retrieve the sensor software version.

Parameter: String SA_Version SA_Version

Direction: Up

Description: Software version of the interface.

17.2.6.3 Get_ChannelInfo (CHI)

Retrieve information about a sensor channel.

Parameter: int32_t SP_Chan SP_Chan

Direction: Down

Valid values:

Minimum: 1

Maximum: 32

Description: Channels to get information for.

Parameter: String SA_ArticleNumber SA_ArticleNumber

Direction: Up

Valid values:

Numeric value

Description: Article number of the sensor channel.

Parameter: String SA_SensorName	SA_SensorName
Direction: Up	
Description: Name of the sensor channel.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor channel.	
Parameter: double SA_Offset	SA_Offset
Direction: Up	
Valid values:	
Minimum: -4294967040.0	
Maximum: 4294967040.0	
Unit: See parameter SA_Unit	
Description: Offset of the sensor channel.	
Parameter: double SA_Range	SA_Range
Direction: Up	
Valid values:	
Minimum: -4294967040.0	
Maximum: 4294967040.0	
Unit: See parameter SA_Unit	
Description: Range of sensor channel.	
Parameter: String SA_Unit	SA_Unit
Direction: Up	
Description: Unit of parameter SA_Offset and SA_Range, e.g. μm or mm.	

17.2.6.4 Get_ChannelInfos

Calls the sensor command Get_ChannelInfo for any requested channel.

Parameter: int32_t SP_Complete	SP_Complete
Direction: Down	
Valid values:	
0= FALSE	
1= TRUE	
Description: Specifies if any possible channel should be requested or only known channels (from former call to ChannelStatus or Get_Status).	
Parameter: String SA_ArticleNumber1..32	SA_ArticleNumber1..32
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the sensor channel 1 to 32.	
Parameter: String SA_SensorName1..32	SA_SensorName1..32
Direction: Up	
Description: Name of the sensor channel 1 to 32.	

Parameter: String SA_SerialNumber1..32	SA_SerialNumber1..32
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the sensor channel 1 to 32.	
Parameter: double SA_Offset1..32	SA_Offset1..32
Direction: Up	
Valid values:	
Minimum: -4294967040.0	
Maximum: 4294967040.0	
Unit: See parameter SA_Unit	
Description: Offset of the sensor channel 1 to 32.	
Parameter: double SA_Range1..32	SA_Range1..32
Direction: Up	
Valid values:	
Minimum: -4294967040.0	
Maximum: 4294967040.0	
Unit: See parameter SA_Unit	
Description: Range of sensor channel 1 to 32.	
Parameter: String SA_Unit1..32	SA_Unit1..32
Direction: Up	
Description: Unit of parameter SA_Offset and SA_Range, e.g. μm or mm.	

17.2.6.5 Get_ControllerInfo (COI)

Retrieve information about the interface.

Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Description: Article number of the interface.	
Parameter: String SA_ControllerName	SA_ControllerName
Direction: Up	
Description: Name of the interface.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the interface.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the interface.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the interface.	

17.2.7 Internal commands

17.2.7.1 Update_Firmware

Update firmware version at interface.

Attention! This function can take approx. 1 minute. This is an internal command. It should not be used by the customer.

Parameter: Binary data SP_FirmwareFile

SP_FirmwareFile

Direction: Down

Valid values:

Firmware file as binary data.

Description: Firmware version, read from file.

Parameter: int32_t SA_Result

SA_Result

Direction: Up

Valid values:

0= Failed

1= Success

Description: Result of firmware update.

17.3 Commands for USBAdapter_IF2004

See USBAdapter_IF2004 manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [IF2004_USB](#) (native).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from IF2004_USB adapter, from 0 to 255 (ExtTriggerIn and RxD).
- Scaled values are same as raw values.

17.3.1 Special commands

17.3.1.1 Set_SyncMaster

Set a synchronization master for a specified sensor channel while interface is open.

Parameter: int32_t SP_SensorChannel

SP_SensorChannel

Direction: Down

Valid values:

0= Sensor channel 1 (Connector 1/2)
 1= Sensor channel 2 (Connector 1/2)
 2= Sensor channel 3 (Connector 3/4)
 3= Sensor channel 4 (Connector 3/4)

Description: Sensor channel to synchronize.

Parameter: int32_t SP_SyncMasterChannel SP_SyncMasterChannel
Direction: Down
Valid values:
 -1= No synchronization master
 0= Sensor channel 1 (Connector 1/2)
 1= Sensor channel 2 (Connector 1/2)
 2= Sensor channel 3 (Connector 3/4)
 3= Sensor channel 4 (Connector 3/4)
 4= Digital IN
Description: Channel number to synchronize with. See [IP_SyncMasterChannel](#) for further information.

17.3.1.2 Set_Algorithm

Set the data processing algorithm at comparator mode.
 This command is only available at special option 1.

Parameter: int32_t SP_Algorithm SP_Algorithm
Direction: Down
Valid values:
 0= Minimum value of selected sensor
 1= Maximum value of selected sensor
Description: Data processing algorithm for comparator.

17.3.1.3 Get_Algorithm

Get the data processing algorithm at comparator mode.
 This command is only available at special option 1.

Parameter: int32_t SA_Algorithm SA_Algorithm
Direction: Up
Valid values:
 0= Minimum value of selected sensor
 1= Maximum value of selected sensor
Description: Data processing algorithm for comparator.

17.3.1.4 Set_SelectedSensors

Set the selected sensors for comparator mode.
 This command is only available at special option 1.
 For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_SelectedSensor1..4 SP_SelectedSensor1..4
Direction: Down
Valid values:
 -1= Leave unchanged
 0= Off
 1= On
Description: Selected sensors for comparator mode.

17.3.1.5 Get_SelectedSensors

Get the selected sensors for comparator mode.
 This command is only available at special option 1.

Parameter: int32_t SA_SelectedSensor_{1..4}

SA_SelectedSensor_{1..4}

Direction: Up

Valid values:

0= Off

1= On

Description: Selected sensors for comparator mode.

17.3.1.6 Set_SensorChannelBaudrate

Set the baudrate of a sensor channel.

This command is normally not needed, because sensor instance set the baudrate itself.

Parameter: int32_t SP_SensorChannel

SP_SensorChannel

Direction: Down

Valid values:

0= Sensor channel 1 (Connector 1/2)

1= Sensor channel 2 (Connector 1/2)

2= Sensor channel 3 (Connector 3/4)

3= Sensor channel 4 (Connector 3/4)

Description: Sensor channel to set baudrate.

Parameter: int32_t SP_SensorChannelBaudrate

SP_SensorChannelBaudrate

Direction: Down

Valid values:

Minimum: 733

Maximum: 8000000

Unit: Baud

Description: Desired baudrate of sensor channel.

Parameter: int32_t SA_SensorChannelRealBaudrate

SA_SensorChannelRealBau-
drate

Direction: Up

Valid values:

Minimum: 733

Maximum: 8000000

Unit: Baud

Description: Real baudrate of sensor channel.

17.3.1.7 Get_SensorChannelBaudrate

Get the baudrate of a sensor channel.

Parameter: int32_t SA_SensorChannelBaudrate_{1..4}

SA_SensorChannelBau-
drate_{1..4}

Direction: Up

Valid values:

Minimum: 733

Maximum: 8000000

Unit: Baud

Description: Nominal baudrate of sensor channel 1 to 4.

Parameter: int32_t SA_SensorChannelRealBaudrate`1..4`

SA_SensorChannelRealBaudrate`1..4`

Direction: Up

Valid values:

Minimum: 733

Maximum: 8000000

Unit: Baud

Description: Real baudrate of sensor channel 1 to 4.

17.3.2 Parameter management

17.3.2.1 Save_Parameters

Save actual parameters at controller.

This command is only available from hardware revision 1.

17.3.2.2 Load_Parameters

Load stored parameters into controller RAM. This command is only available from hardware revision 1.

17.3.3 Information

17.3.3.1 Get_FPGAVersion

Get the version of the FPGA of the adapter.

Parameter: int32_t SA_FPGAVersion

SA_FPGAVersion

Direction: Up

Valid values:

Minimum: 0

Maximum: 255

Description: Version of the FPGA of the adapter.

17.3.3.2 Get_HWRevision

Get the hardware revision of the adapter.

Parameter: int32_t SA_HWRevision

SA_HWRevision

Direction: Up

Valid values:

Minimum: 0

Maximum: 255

Description: Hardware revision of the adapter.

17.3.3.3 Get_Option

Get the option of the adapter.

Parameter: int32_t SA_Option SA_Option
Direction: Up
Valid values:
Minimum: 0
Maximum: 15
Description: Option of the adapter.

17.3.4 Timer

17.3.4.1 Set_TimerFrequency

Set the frequency and pulse width for the timers on the adapter.

Parameter: int32_t SP_TimerNumber SP_TimerNumber
Direction: Down
Valid values:
 1= Timer 1
 2= Timer 2
Description: Number of the timer to parametrize.

Parameter: double SP_TimerFrequency SP_TimerFrequency
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 12000000.0
Unit: Hz
Description: Frequency of the timer.
 0.0 means the timer is off.

Parameter: double SP_TimerRatio SP_TimerRatio
Direction: Down
Valid values:
Minimum: 0.0
Maximum: 1.0
Description: Ratio of the high and low contingent of one period.
 If you are unsure, specify a value of 0.5.
 If SP_TimerFrequency is 0.0, a value of 0.0 means the timer is low,
 otherwise the timer is high.

Parameter: double SA_RealFrequency SA_RealFrequency
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 12000000.0
Unit: Hz
Description: The real frequency applied to the timer. Because of internal
 limitations not each value can be set.

Parameter: double SA_RealRatio SA_RealRatio
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 1.0
Description: The real ratio applied to the timer. Because of internal limitations not each value can be set.

17.3.4.2 Get_TimerFrequency

Get the frequency and pulse width for the timers on the adapter.

Parameter: int32_t SP_TimerNumber SP_TimerNumber
Direction: Down
Valid values:

- 1 = Timer 1
- 2 = Timer 2

Description: Number of the timer to parametrize.

Parameter: double SA_RealFrequency SA_RealFrequency
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 12000000.0
Unit: Hz
Description: The frequency applied to the timer.

Parameter: double SA_RealRatio SA_RealRatio
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 1.0
Description: The ratio applied to the timer.

17.3.5 RS422 (Sensor)

17.3.5.1 Inputs

17.3.5.1.1 Use_Gate

The external digital inputs 1 to 3 of the adapter (5V TTL signal) can be used to lock or free the FIFO for data from specific channels.

Parameter: int32_t SP_GateChannel SP_GateChannel
Direction: Down
Valid values:
 0 = External digital input 1 for Sensor channel 1+2 (Connector 1/2)
 1 = External digital input 1 for Sensor channel 1+2 (Connector 1/2)
 2 = External digital input 2 for Sensor channel 3+4 (Connector 3/4)
 3 = External digital input 2 for Sensor channel 3+4 (Connector 3/4)
 4 = External digital input 3 for external digital input and RxD

Description: Number of the channel to lock by external digital input. Because multiple channels are affected by one external digital input, the numbers 0..1 or 2..3 always affects the same external digital input.

Parameter: int32_t SP_ActivateGate SP_ActivateGate

Direction: Down

Valid values:

- 0= on
- 1= off

Description: Specifies if the gate function is enabled or disabled for the specific digital input.

17.3.5.1.2 Get_RxDValue

Get the RxD values.

Parameter: int32_t SA_RxDValue1..4 SA_RxDValue1..4

Direction: Up

Valid values:

- 0
- 1

Description: Value of the RxD line.

17.3.5.2 Outputs

17.3.5.2.1 Set_TxDSource

Set the source which is used for TxD line.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_TxDChannel1..4 SP_TxDChannel1..4

Direction: Down

Valid values:

- 1= Leave unchanged
- 0= Sensor output (transmitter)
- 1= Continuously low
- 2= Continuously high
- 3= Any sensor output (1-4)

Description: Source which is used for TxD line.

17.3.5.2.2 Get_TxDSource

Get the source which is used for TxD line.

Parameter: int32_t SA_TxDChannel1..4 SA_TxDChannel1..4

Direction: Up

Valid values:

- 1= Leave unchanged
- 0= sensor output (transmitter)
- 1= Continuously low
- 2= Continuously high
- 3= Any sensor output (1-4)

Description: Source which is used for TxD line.

17.3.5.3 Digital Inputs and Outputs

17.3.5.3.1 Set_TrgSource

Set the source which is used for Trg line to sensor.
 For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_TrgChannel`1..4`

SP_TrgChannel`1..4`

Direction: Down

Valid values:

- 1= Leave unchanged
- 0= Digital input 1
- 1= Digital input 2
- 2= Digital input 3
- 3= Digital input 4
- 4= Timer 1 pulse width
- 5= Timer 2 pulse width
- 6= Continuously low
- 7= Continuously high

Description: Source which is used for Trg line to sensor.

17.3.5.3.2 Get_TrgSource

Get the source which is used for Trg line to sensor.

Parameter: int32_t SA_TrgChannel`1..4`

SA_TrgChannel`1..4`

Direction: Up

Valid values:

- 1= Leave unchanged
- 0= Digital input 1
- 1= Digital input 2
- 2= Digital input 3
- 3= Digital input 4
- 4= Timer 1 pulse width
- 5= Timer 2 pulse width
- 6= Continuously low
- 7= Continuously high

Description: Source which is used for Trg line to sensor.

17.3.6 RS422 (Computer)

17.3.6.1 Set_DataMode

Set the data output format at RS422 and USB interface.
 This command is only available from hardware revision 1.

Parameter: int32_t SP_DataMode

SP_DataMode

Direction: Down

Valid values:

- 0= RS422 input for Sensor 1

17.3. Commands for USBAdapter_IF2004

- 1= Comparator at RS422 (three byte data format, like ILD2300, tuple at USB). Only valid at special option 1.
- 2= Off (no data at RS422, only at USB)
- 3= Tuple at RS422 (two bytes, address and data per sensor byte, no data at USB)

Description: Data output format at RS422 and USB interface.

If tuple mode is active, data is no longer output at USB interface.

If comparator mode is active, only sensor data is output at RS422, Digital IN is already output at USB. This mode does only work if all selected sensors supports three byte data format and outputs one value per frame.

17.3.6.2 Get_DataMode

Get the data output format at RS422 and USB interface.

This command is only available from hardware revision 1.

Parameter: int32_t SA_DataMode

SA_DataMode

Direction: Up

Valid values:

- 0= RS422 input for Sensor 1
- 1= Comparator at RS422 (three byte data format, like ILD2300, tuple at USB)
- 2= Off (no data at RS422, only at USB)
- 3= Tuple at RS422 (two bytes, address and data per sensor byte, no data at USB)

Description: Data output format at RS422 and USB interface.

17.3.6.3 Set_RS422OutSource

Set the source which is used for RS422 output at comparator mode.

This command is only available at special option 1.

Parameter: int32_t SP_RS422OutSource

SP_RS422OutSource

Direction: Down

Valid values:

- 0= Timer 1 pulse width
- 1= Timer 2 pulse width
- 2= Sensor channel 1
- 3= Sensor channel 3
- 4= External digital input 1
- 5= External digital input 2
- 6= External digital input 3
- 7= External digital input 4

Description: Source which is used for RS422 output at comparator mode.

17.3.6.4 Get_RS422OutSource

Get the source which is used for RS422 output at comparator mode.
 This command is only available at special option 1.

Parameter: int32_t SA_RS422OutSource

SA_RS422OutSource

Direction: Up

Valid values:

- 0= Timer 1 pulse width
- 1= Timer 2 pulse width
- 2= Sensor channel 1
- 3= Sensor channel 3
- 4= External digital input 1
- 5= External digital input 2
- 6= External digital input 3
- 7= External digital input 4

Description: Source which is used for RS422 output at comparator mode.

17.3.6.5 Set_RS422Baudrate

Set the baudrate of RS422 output.

This command is only available from hardware revision 1.

Parameter: int32_t SP_RS422Baudrate

SP_RS422Baudrate

Direction: Down

Valid values:

- Minimum:** 733
- Maximum:** 8000000

Description: Desired baudrate of RS422 output.

If the device should be found by [SensorFinder commands](#) over RS422,
 only following baudrates are allowed: 8000000, 4000000, 3500000,
 3000000, 2500000, 2000000, 1500000, 921600, 691200, 460800, 230400,
 115200, 9600.

Parameter: int32_t SA_RS422RealBaudrate

SA_RS422RealBaudrate

Direction: Up

Valid values:

- Minimum:** 733
- Maximum:** 8000000

Description: Real baudrate of RS422 output.

17.3.6.6 Get_RS422Baudrate

Get the baudrate of RS422 output.

This command is only available from hardware revision 1.

Parameter: int32_t SA_RS422Baudrate

SA_RS422Baudrate

Direction: Up

Valid values:

- Minimum:** 733
- Maximum:** 8000000

Description: Nominal baudrate of RS422 output.

Parameter: int32_t SA_RS422RealBaudrate SA_RS422RealBaudrate
Direction: Up
Valid values:
Minimum: 733
Maximum: 8000000
Description: Real baudrate of RS422 output.

17.3.7 Switching inputs

17.3.7.1 Set_ExtTriggerInSource

Set the source which is used for external digital inputs.

Parameter: int32_t SP_ExtTriggerInSource SP_ExtTriggerInSource
Direction: Down
Valid values:
 0= Sensor channel 1
 1= Sensor channel 3
 2= Timer 1 pulse width
 3= Timer 2 pulse width
Description: Source which is used for external digital inputs.

17.3.7.2 Get_ExtTriggerInSource

Get the source which is used for external digital inputs.

Parameter: int32_t SA_ExtTriggerInSource SA_ExtTriggerInSource
Direction: Up
Valid values:
 0= Sensor channel 1
 1= Sensor channel 3
 2= Timer 1 pulse width
 3= Timer 2 pulse width
Description: Source which is used for external digital inputs.

17.3.7.3 Set_ExtTriggerInDirection

Set the external digital input direction.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_ExtTriggerInDirection1..4 SP_ExtTriggerInDirection1..4
Direction: Down
Valid values:
 -1= Leave unchanged
 0= Normal
 1= Inverted
Description: External digital input direction.

17.3.7.4 Get_ExtTriggerInDirection

Get the external digital input direction.

Parameter: int32_t SA_ExtTriggerInDirection1..4

SA_ExtTriggerInDirection1..4

Direction: Up

Valid values:

-1= Leave unchanged

0= Normal

1= Inverted

Description: External digital input direction.

17.3.7.5 Get_ExtTriggerInValue

Get the external digital input values.

Parameter: int32_t SA_ExtTriggerInValue1..4

SA_ExtTriggerInValue1..4

Direction: Up

Valid values:

0

1

Description: External digital input value.

17.3.8 Switching outputs

17.3.8.1 Set_ExtTriggerOutSource

Set the source which is used for external digital outputs.

Parameter: int32_t SP_ExtTriggerOutSource1..2

SP_ExtTriggerOutSource1..2

Direction: Down

Valid values:

-1= Leave unchanged

0= Sensor channel 1

1= Sensor channel 2

2= Sensor channel 3

3= Sensor channel 4

4= Timer 1 pulse width

5= Timer 2 pulse width

6= Continuously low

7= Continuously high

Description: Source which is used for external digital outputs.

17.3.8.2 Get_ExtTriggerOutSource

Get the source which is used for external digital outputs.

Parameter: int32_t SA_ExtTriggerOutSource`1..2`

SA_ExtTriggerOut-
Source`1..2`

Direction: Up

Valid values:

- 1= Leave unchanged
- 0= Sensor channel 1
- 1= Sensor channel 2
- 2= Sensor channel 3
- 3= Sensor channel 4
- 4= Timer 1 pulse width
- 5= Timer 2 pulse width
- 6= Continuously low
- 7= Continuously high

Description: Source which is used for external digital outputs.

17.3.9 LED

17.3.9.1 Set_LEDMode

Set the mode of the LED's at device.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_LEDMode1

SP_LEDMode1

Direction: Down

Valid values:

- 1= Leave unchanged
- 0= Light with USB ready
- 1= Light with digital input 1
- 2= Light with RxD 1
- 3= Light with TxD 1

Description: Mode of LED 1.

Parameter: int32_t SP_LEDMode2

SP_LEDMode2

Direction: Down

Valid values:

- 1= Leave unchanged
- 0= Light with extern Power +24V
- 1= Light with digital input 2
- 2= Light with RxD 2
- 3= Light with TxD 2

Description: Mode of LED 2.

Parameter: int32_t SP_LEDMode3

SP_LEDMode3

Direction: Down

Valid values:

- 1= Leave unchanged
- 0= Light if FIFO has data
- 1= Light with digital input 3
- 2= Light with RxD 3
- 3= Light with TxD 3

Description: Mode of LED 3.

Parameter: int32_t SP_LEDMode4 SP_LEDMode4
Direction: Down
Valid values:
 -1= Leave unchanged
 0= Light with any TxD (1-4)
 1= Light with digital input 4
 2= Light with RxD 4
 3= Light with TxD 4
Description: Mode of LED 4.

17.3.9.2 Get_LEDMode

Get the mode of the LED's at device.

Parameter: int32_t SA_LEDMode1 SA_LEDMode1
Direction: Up
Valid values:
 -1= Leave unchanged
 0= Light with USB ready
 1= Light with digital input 1
 2= Light with RxD 1
 3= Light with TxD 1
Description: Mode of LED 1.

Parameter: int32_t SA_LEDMode2 SA_LEDMode2
Direction: Up
Valid values:
 -1= Leave unchanged
 0= Light with extern Power +24V
 1= Light with digital input 2
 2= Light with RxD 2
 3= Light with TxD 2
Description: Mode of LED 2.

Parameter: int32_t SA_LEDMode3 SA_LEDMode3
Direction: Up
Valid values:
 -1= Leave unchanged
 0= Light if FIFO has data
 1= Light with digital input 3
 2= Light with RxD 3
 3= Light with TxD 3
Description: Mode of LED 3.

Parameter: int32_t SA_LEDMode4 SA_LEDMode4
Direction: Up
Valid values:
 -1= Leave unchanged
 0= Light with any TxD (1-4)
 1= Light with digital input 4
 2= Light with RxD 4
 3= Light with TxD 4
Description: Mode of LED 4.

17.4 Commands for PCICardIF2004

See IF2004 manual for detailed description of sensor commands.

Attention! The command [Use_Defaults](#) must be called for correct functionality of encoder. Please see the example at command [Use_Defaults](#).

This sensor supports following interfaces:

- [IF2004](#) (native).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from encoder on IF2004, from 0 to 65536.
- Scaled values are scaled using encoder distance per count (if known by MEDAQLib, use sensor command [Use_Defaults](#) with parameter [IP_Range](#)).

17.4.1 Special commands

17.4.1.1 [Use_Defaults](#)

This command parametrizes the IF2004 PCI card. If same parameters are not specified they are not changed.

Parameter: double [IP_Range](#)

[IP_Range](#)

Direction: Down

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Unit: μm or mm

Description: It is the distance per count of the encoder used by the driver for scaling data.

If it is zero, no scaling is done.

Parameter: int32_t [IP_EncCountMode](#)

[IP_EncCountMode](#)

Valid values:

0= Counter without phase discriminator (Trace A is direction, Trace B is pulse, Trace C is load or latch signal).

1= Counter with phase discriminator and 1-fold discriminator.

2= Counter with phase discriminator and 2-fold discriminator.

3= Counter with phase discriminator and 4-fold discriminator.

Direction: Down

Description: The counting mode of the encoder

Parameter: int32_t [IP_EncSwapTraceA_B](#)

[IP_EncSwapTraceA_B](#)

Direction: Down

Valid values:

0= no swap

1= swap Trace A and B

Description: Trace A and B are swapped which negates the count direction

Parameter: int32_t IP_EncInvertTraceA_B	IP_EncInvertTraceA_B
Direction: Down	
Valid values:	
0= no invert	
1= invert Trace A and B	
Description: Trace A and B are inverted, for Encoders where Trace C is 1 when Trace A and B are 0	
Parameter: int32_t IP_EncLatchSrc	IP_EncLatchSrc
Direction: Down	
Valid values:	
0= Never	
1..4= Latch with start bit on channel 1 to 4.	
5= Latch with next reference on Trace C (if unlocked).	
6= Latch with second reference on Trace C (if unlocked).	
7= Latch with every reference on Trace C (always).	
Description: Specifies when the encoder value should be stored to FIFO (only useful when IP_ChannelNumber is 3)	
Parameter: int32_t IP_EncLoadOnRef	IP_EncLoadOnRef
Direction: Down	
Valid values:	
0= never	
1= Load with next reference on Trace C (if unlocked).	
2= Load with every reference on Trace C (always).	
3= Clear with every reference on Trace C (always) and load when count is -1.	
Description: Specifies when the encoder value should be changed by IF2004 card.	

Example how to parametrize the encoder:

```

uint32_t instance= CreateSensorInstance (PCI_CARD_IF2004);
ERR_CODE err= OpenSensorIF2004 (instance, 0, 3);
/* error handling, if err!=ERR_NOERROR */

SetParameterString (instance, "S_Command", "Use_Defaults");
/* Set encoder to increment or decrement each full period */
SetParameterInt (instance, "IP_EncCountMode", 1);
/* Each time when a value arrives at first sensor channel, put the encoder value into
FIFO */
SetParameterInt (instance, "IP_EncLatchSrc", 1);
err= SensorCommand (instance);
/* error handling, if err!=ERR_NOERROR*/

```

17.4.1.2 Get_DrvSetting

Returns the current settings of the driver used for operating with sensor data. It is the opposite of Use_Defaults.

Parameter: double IA_Range	IA_Range
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 1.79769e+308 (DBL_MAX)	
Unit: μm or mm	
Description: It is the distance per count of the encoder used by the driver for scaling data.	

17.4.1.3 Set_SyncMaster

Set a synchronization master for a specified sensor channel while interface is open.

Parameter: int32_t SP_SensorChannel

SP_SensorChannel

Direction: Down

Valid values:

- 0= Sensor channel 1 (Base Board, Connector 1/2)
- 1= Sensor channel 2 (Base Board, Connector 1/2)
- 2= Sensor channel 3 (Base Board, Connector 3/4)
- 3= Sensor channel 4 (Base Board, Connector 3/4)

Description: Sensor channel to synchronize.

Parameter: int32_t SP_SyncMasterChannel

SP_SyncMasterChannel

Direction: Down

Valid values:

- 1= No synchronization master
- 0= Sensor channel 1 (Base Board, Connector 1/2)
- 1= Sensor channel 2 (Base Board, Connector 1/2)
- 2= Sensor channel 3 (Base Board, Connector 3/4)
- 3= Sensor channel 4 (Base Board, Connector 3/4)

Description: Channel number to synchronize with. See [IP_SyncMasterChannel](#) for further information.

17.4.2 General

17.4.2.1 Get_FPGAVersion

Get the version of the FPGA on the card.

Parameter: int32_t SA_FPGAVersion

SA_FPGAVersion

Direction: Up

Valid values:

- 0= V1.1
- 1= V1.2 (115.2 KBaud, new data format for [SENSOR_ILD1700](#))
- 2= V1.3 (1.25 MBaud, support for two sensor values per data frame)

Description: Version of the FPGA on the card.

17.4.2.2 IF2004_SystemReset

Make a system reset: Clear FIFO and send a pulse (100 μ s) reset line at both sensor connectors.

17.4.2.3 IF2004_SensorReset12

Set the reset line of sensor connector 0+1 to specified value.

Parameter: int32_t SP_SensorReset12

SP_SensorReset12

Direction: Down

Valid values:

- 0= FALSE
- 1= TRUE

Description: Value for reset line.

17.4.2.4 IF2004_SensorReset34

Set the reset line of sensor connector 2+3 to specified value.

Parameter: int32_t SP_SensorReset34

SP_SensorReset34

Direction: Down

Valid values:

0 = FALSE

1 = TRUE

Description: Value for reset line.

17.4.3 Encoder

17.4.3.1 Actions

17.4.3.1.1 Enc_ClearCounter

Clears the counter (set to 0).

17.4.3.1.2 Enc_SetLoadReg

Set the load register for counter.

Parameter: int32_t SP_LoadReg

SP_LoadReg

Direction: Down

Valid values:

Minimum: 0

Maximum: 65535

Description: Counter value.

17.4.3.1.3 Enc_LoadCounter

Load the counter from the load register.

17.4.3.1.4 Enc_LatchCounter

Get the recent counter value to the latch register.

17.4.3.1.5 Enc_GetLatchReg

Get the counter value from the latch register.

Parameter: int32_t SA_LatchReg

SA_LatchReg

Direction: Up

Valid values:

Minimum: 0

Maximum: 65535

Description: Counter value.

17.4.3.1.6 Enc_UnlockTraceC

Unlocks Trace C.

17.4.3.1.7 Enc_CheckRef

Checks the actual state of trace C.

Parameter: int32_t SA_Ref

SA_Ref

Direction: Up

Valid values:

0= FALSE
1= TRUE

Description: True if reference (trace C) is set.

17.4.3.1.8 Enc_IsFirstRef

Checks if Ref (Trace C) was reached since last Unlock.

Parameter: int32_t SA_IsFirstRef

SA_IsFirstRef

Direction: Up

Valid values:

0= FALSE
1= TRUE

Description: True, if it is first reference.

17.4.3.1.9 Enc_IsSecondRef

Checks if Ref (Trace C) was reached twice since last Unlock.

Parameter: int32_t SA_IsSecondRef

SA_IsSecondRef

Direction: Up

Valid values:

0= FALSE
1= TRUE

Description: True, if it is second reference.

17.4.4 RS422

17.4.4.1 Inputs

17.4.4.1.1 IF2004_CheckGate

Checks the actual state of gate.

Parameter: int32_t SA_Gate

SA_Gate

Direction: Up

Valid values:

0= FALSE
1= TRUE

Description: True if gate is set.

17.5 Commands for PCICard_IF2008

See PCICard_IF2008 manual for detailed description of sensor commands.

Attention! The command [Use_Defaults](#) must be called if encoder or ADC values should be scaled.

This sensor supports following interfaces:

- [IF2008](#) (native).

To allow MEDAQLib to scale data, it is recommended to call sensor command [Use_Defaults](#) after [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from IF2008 card, from 0 to 4294967295 (Encoder) or 65535 (ADC) or 255 (Digital in).
- Scaled values are scaled using count factors or ranges (if known by MEDAQLib, use sensor command [Use_Defaults](#)).

17.5.1 Special commands

17.5.1.1 Use_Defaults

This command parametrizes the IF2008 PCI card. If same parameters are not specified they are not changed.

Parameter: double IP_Encoder1CountFactor

IP_Encoder1CountFactor

Direction: Down

Valid for sensor:

PCI_CARD_IF2008

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Default: 1.0

Description: It is the distance per count of the encoder 1 used by the driver for scaling data. If it is zero, no scaling is done.

Parameter: double IP_Encoder2CountFactor

IP_Encoder2CountFactor

Direction: Down

Valid for sensor:

PCI_CARD_IF2008

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Default: 1.0

Description: It is the distance per count of the encoder 2 used by the driver for scaling data. If it is zero, no scaling is done.

17.5. Commands for PCICard_IF2008

Parameter: int32_t IP_ADC1Range IP_ADC1Range

Direction: Down

Valid for sensor:

PCI_CARD_IF2008

Valid values:

0= 0..5V

1= 0..10V

2= +-5V

3= +-10V

Default: 3

Description: It is range of ADC 1 used by the driver for scaling data.

Parameter: int32_t IP_ADC2Range IP_ADC2Range

Direction: Down

Valid for sensor:

PCI_CARD_IF2008

Valid values:

0= 0..5V

1= 0..10V

2= +-5V

3= +-10V

Default: 3

Description: It is range of ADC 2 used by the driver for scaling data.

17.5.1.2 Get_DrvSetting

Returns the current settings of the driver used for operating. It is the opposite of Use_Defaults.

Parameter: double IA_Encoder1CountFactor IA_Encoder1CountFactor

Direction: Up

Valid for sensor:

PCI_CARD_IF2008

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Description: It is the distance per count of the encoder 1 used by the driver for scaling data. If it is zero, no scaling is done.

Parameter: double IA_Encoder2CountFactor IA_Encoder2CountFactor

Direction: Up

Valid for sensor:

PCI_CARD_IF2008

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Description: It is the distance per count of the encoder 2 used by the driver for scaling data. If it is zero, no scaling is done.

17.5. Commands for PCICard_IF2008

Parameter: int32_t IA_ADC1Range IA_ADC1Range

Direction: Up

Valid for sensor:

PCI_CARD_IF2008

Valid values:

0= 0..5V

1= 0..10V

2= +-5V

3= +-10V

Description: It is range of ADC 1 used by the driver for scaling data.

Parameter: int32_t IA_ADC2Range IA_ADC2Range

Direction: Up

Valid for sensor:

PCI_CARD_IF2008

Valid values:

0= 0..5V

1= 0..10V

2= +-5V

3= +-10V

Description: It is range of ADC 2 used by the driver for scaling data.

17.5.1.3 Set_SyncMaster

Set a synchronization master for a specified sensor channel while interface is open.

Parameter: int32_t SP_SensorChannel SP_SensorChannel

Direction: Down

Valid values:

0= Sensor channel 1 (Base Board, Connector X1)

1= Sensor channel 2 (Base Board, Connector X1)

2= Sensor channel 3 (Base Board, Connector X2)

3= Sensor channel 4 (Base Board, Connector X2)

4= Sensor channel 5 (Extension Board, Connector X1)

5= Sensor channel 6 (Extension Board, Connector X1)

Description: Sensor channel to synchronize.

Parameter: int32_t SP_SyncMasterChannel SP_SyncMasterChannel

Direction: Down

Valid values:

-1= No synchronization master

0= Sensor channel 1 (Base Board, Connector X1)

1= Sensor channel 2 (Base Board, Connector X1)

2= Sensor channel 3 (Base Board, Connector X2)

3= Sensor channel 4 (Base Board, Connector X2)

4= Sensor channel 5 (Extension Board, Connector X1)

5= Sensor channel 6 (Extension Board, Connector X1)

6= Encoder 1

7= Encoder 2

8= Digital IN

9= RxD

10= ADC 1

11= ADC 2

Description: Channel number to synchronize with. See [IP_SyncMasterChannel](#) for further information.

17.5.1.4 Set_PowerSwitch

Switches the sensor power on or off.

Parameter: int32_t SP_DisablePower

SP_DisablePower

Direction: Down

Valid values:

0= FALSE

1= TRUE

Description: Sensor power disabled (for all channels)

17.5.2 Information

17.5.2.1 Get_FPGAVersion

Get the version of the FPGA on the card.

Parameter: int32_t SA_FPGAVersion

SA_FPGAVersion

Direction: Up

Valid values:

Minimum: 0

Maximum: 63

Description: Version of the FPGA on the card.

17.5.2.2 Get_HWRevision

Get the hardware revision of the IF2008 card.

Parameter: int32_t SA_HWRevision

SA_HWRevision

Direction: Up

Valid values:

Minimum: 0

Maximum: 3

Description: Hardware revision of the IF2008 card.

17.5.2.3 Is_Channel56Available

Check if additional sensor channels at IF2008E extension card are available.

Parameter: int32_t SA_Channel56Available

SA_Channel56Available

Direction: Up

Valid values:

Minimum: 0= no

Maximum: 1= yes

Description: Availability of sensor channels at IF2008E extension card.

17.5.2.4 Is_ADCAvailable

Check if Analog/Digital converter (ADC) at IF2008E extension card are available.

Parameter: int32_t SA_ADCAvailable

SA_ADCAvailable

Direction: Up

Valid values:

Minimum: 0 = no

Maximum: 1 = yes

Description: Availability of Analog/Digital converter (ADC) at IF2008E extension card.

17.5.2.5 Is_DigitalIOAvailable

Check if digital IO at IF2008E extension card or IF2008IO extension slot is available.

Parameter: int32_t SA_DigitalIOAvailable

SA_DigitalIOAvailable

Direction: Up

Valid values:

Minimum: 0 = no

Maximum: 1 = yes

Description: Availability of digital IO at IF2008E extension card or IF2008IO extension slot.

17.5.2.6 Get_PowerError

Detect if sensor power is overloaded or short-circuited.

Parameter: int32_t SA_PowerError

SA_PowerError

Direction: Up

Valid values:

Minimum: 0 = OK

Maximum: 1 = overloaded

Description: State of sensor power supply.

17.5.3 Timer

17.5.3.1 Set_TimerFrequency

Set the frequency and pulse width for the timers on the card.

Parameter: int32_t SP_TimerNumber

SP_TimerNumber

Direction: Down

Valid values:

1 = Timer 1

2 = Timer 2

3 = Timer 3

Description: Number of the timer to parametrize.

Parameter: double SP_TimerFrequency	SP_TimerFrequency
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 20000000.0	
Unit: Hz	
Description: Frequency of the timer.	
0.0 means the timer is off.	
Parameter: double SP_TimerRatio	SP_TimerRatio
Direction: Down	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: Ratio of the high and low contingent of one period.	
If you are unsure, specify a value of 0.5.	
If SP_TimerFrequency is 0.0, a value of 0.0 means the timer is low, otherwise the timer is high.	
Parameter: double SA_RealFrequency	SA_RealFrequency
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 20000000.0	
Unit: Hz	
Description: The real frequency applied to the timer. Because of internal limitations not each value can be set.	
Parameter: double SA_RealRatio	SA_RealRatio
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: The real ratio applied to the timer. Because of internal limitations not each value can be set.	

17.5.4 Encoder

17.5.4.1 Settings

17.5.4.1.1 Set_EncoderInterpolation

Set the interpolation of the encoder.

Parameter: int32_t SP_EncoderNumber	SP_EncoderNumber
Direction: Down	
Valid values:	
6= Encoder 1	
7= Encoder 2	
Description: Number of the encoder to parametrize.	

Parameter: int32_t SP_EncoderInterpolation SP_EncoderInterpolation

Direction: Down

Valid values:

- 0= 1 fold (TTL & 1Vss)
- 1= 2 fold (TTL & 1Vss)
- 2= 3 fold (only 1Vss)
- 3= 4 fold (TTL & 1Vss)
- 4= 5 fold (only 1Vss)
- 5= 6 fold (only 1Vss)
- 6= 8 fold (only 1Vss)
- 7= 10 fold (only 1Vss)
- 8= 12 fold (only 1Vss)
- 9= 16 fold (only 1Vss)
- 10= 20 fold (only 1Vss)
- 11= 24 fold (only 1Vss)
- 12= 32 fold (only 1Vss)
- 13= 40 fold (only 1Vss)
- 14= 48 fold (only 1Vss)
- 15= 64 fold (only 1Vss)

Description: Interpolation mode of the encoder.

Attention! If a encoder with 5V TTL signal is used, only mode 0, 1 and 3 does work. If another mode is selected, the result is unpredictable.

17.5.4.1.2 Get_EncoderInterpolation

Get the interpolation of the encoder.

Parameter: int32_t SP_EncoderNumber SP_EncoderNumber

Direction: Down

Valid values:

- 6= Encoder 1
- 7= Encoder 2

Description: Number of the encoder to retrieve information.

Parameter: int32_t SA_EncoderInterpolation SA_EncoderInterpolation

Direction: Up

Valid values:

- 0= 1 fold (TTL & 1Vss)
- 1= 2 fold (TTL & 1Vss)
- 2= 3 fold (only 1Vss)
- 3= 4 fold (TTL & 1Vss)
- 4= 5 fold (only 1Vss)
- 5= 6 fold (only 1Vss)
- 6= 8 fold (only 1Vss)
- 7= 10 fold (only 1Vss)
- 8= 12 fold (only 1Vss)
- 9= 16 fold (only 1Vss)
- 10= 20 fold (only 1Vss)
- 11= 24 fold (only 1Vss)
- 12= 32 fold (only 1Vss)
- 13= 40 fold (only 1Vss)
- 14= 48 fold (only 1Vss)
- 15= 64 fold (only 1Vss)

Description: Interpolation mode of the encoder.

17.5.4.1.3 Set_EncoderDirection

Set the direction of the encoder.

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

6= Encoder 1

7= Encoder 2

Description: Number of the encoder to parametrize.

Parameter: int32_t SP_EncoderDirection

SP_EncoderDirection

Direction: Down

Valid values:

0= normal

1= reverse

Description: Count direction of the encoder.

17.5.4.1.4 Get_EncoderDirection

Get the direction of the encoder.

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

6= Encoder 1

7= Encoder 2

Description: Number of the encoder to parametrize.

Parameter: int32_t SA_EncoderDirection

SA_EncoderDirection

Direction: Up

Valid values:

0= normal

1= reverse

Description: Count direction of the encoder.

17.5.4.1.5 Set_EncoderMode

Set the behaviour of the encoder when a reference is reached.

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

6= Encoder 1

7= Encoder 2

Description: Number of the encoder to parametrize.

Parameter: int32_t SP_EncoderMode SP_EncoderMode

Direction: Down

Valid values:

- 0= no function
- 1= load counter with next reference mark
- 2= load counter with all reference marks
- 3= clear counter with all reference marks, load when -1 is reached
- 4= counter without phase discriminator (pulse counter)

Description: Behaviour when a reference is reached.

If counter is set to pulse counter, the encoder direction specifies the function of the input lines. On normal direction, A is for count direction and B is for pulse counting, on reverse direction it is reversed.

A reference mark only can be detected, if A, B and C are high at the same time. On encoders this is ensured, on other pulse generators it must be ensured by the hardware developer.

17.5.4.1.6 Get_EncoderMode

Get the behaviour of the encoder when a reference is reached.

Parameter: int32_t SP_EncoderNumber SP_EncoderNumber

Direction: Down

Valid values:

- 6= Encoder 1
- 7= Encoder 2

Description: Number of the encoder to parametrize.

Parameter: int32_t SA_EncoderMode SA_EncoderMode

Direction: Up

Valid values:

- 0= no function
- 1= load counter with next reference mark
- 2= load counter with all reference marks
- 3= clear counter with all reference marks, load when -1 is reached
- 4= counter without phase discriminator (pulse counter)

Description: Behaviour when a reference is reached.

17.5.4.1.7 Set_EncoderLatchSource

Set the latch source which triggers aquiring one value.

Parameter: int32_t SP_EncoderNumber SP_EncoderNumber

Direction: Down

Valid values:

- 6= Encoder 1
- 7= Encoder 2

Description: Number of the encoder to parametrize.

Parameter: int32_t SP_EncoderLatchSource SP_EncoderLatchSource

Direction: Down

Valid values:

- 0= never (locked)
- 1= Timer 1
- 2= Timer 2
- 3= Timer 3
- 4= Sensor channel 1
- 5= Sensor channel 2
- 6= Sensor channel 3
- 7= Sensor channel 4
- 8= Sensor channel 5
- 9= Sensor channel 6
- 10= Digital In 1
- 11= Digital In 2
- 12= Digital In 3
- 13= Digital In 4
- 14= second reference mark
- 15= all reference marks
- 16= Each counter pulse (only IF2008 PCIe from FPGA version 10)
- 17= Each counter pulse after first reference mark (only IF2008 PCIe from FPGA version 10)
- 18= Each counter pulse between first and second reference mark (only IF2008 PCIe from FPGA version 10)

Description: Latch source which triggers aquiring one value.

17.5.4.1.8 Get_EncoderLatchSource

Get the latch source which triggers aquiring one value.

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

- 6= Encoder 1
- 7= Encoder 2

Description: Number of the encoder to parametrize.

Parameter: int32_t SA_EncoderLatchSource

SA_EncoderLatchSource

Direction: Up

Valid values:

- 0= never (locked)
- 1= Timer 1
- 2= Timer 2
- 3= Timer 3
- 4= Sensor channel 1
- 5= Sensor channel 2
- 6= Sensor channel 3
- 7= Sensor channel 4
- 8= Sensor channel 5
- 9= Sensor channel 6
- 10= Digital In 1
- 11= Digital In 2

12= Digital In 3
 13= Digital In 4
 14= second reference mark
 15= all reference marks
 16= Each counter pulse (only IF2008 PCIe from FPGA version 10)
 17= Each counter pulse after first reference mark (only IF2008 PCIe from FPGA version 10)
 18= Each counter pulse between first and second reference mark (only IF2008 PCIe from FPGA version 10)

Description: Latch source which triggers acquiring one value.

17.5.4.2 Actions

17.5.4.2.1 Clear_Encoder

Clears the encoder value (set it to zero).

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

6= Encoder 1
7= Encoder 2

Description: Number of the encoder to clear.

17.5.4.2.2 Set_EncoderPreload

Set the preload value which is used when loading the encoder.

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

6= Encoder 1
7= Encoder 2

Description: Number of the encoder to parametrize.

Parameter: int32_t SP_EncoderPreloadValue

SP_EncoderPreloadValue

Direction: Down

Valid values:

Minimum: -2147483648 (INT32_MIN)
Maximum: 2147483647 (INT32_MAX)

Description: Preload value which is used when loading the encoder.

17.5.4.2.3 Load_Encoder

Loads the encoder value with the value set by function Set_EncoderPreload.

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

6= Encoder 1
7= Encoder 2

Description: Number of the encoder to load.

17.5.4.2.4 Latch_Encoder

Latch the encoder (get latest value into encoder register).
 The value can be retrieved using function Get_EncoderValue.

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

6= Encoder 1

7= Encoder 2

Description: Number of the encoder to latch.

17.5.4.2.5 Get_EncoderValue

Get the value of encoder register.

This function is useful in combination with function Latch_Encoder.

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

6= Encoder 1

7= Encoder 2

Description: Number of the encoder to latch.

Parameter: int32_t SA_EncoderRawValue

SA_EncoderRawValue

Direction: Up

Valid values:

Minimum: -2147483648 (INT32_MIN)

Maximum: 2147483647 (INT32_MAX)

Description: Raw value of encoder register.

Parameter: double SA_EncoderScaledValue

SA_EncoderScaledValue

Direction: Up

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Description: Scaled value of encoder register.

If IP_Encoder1CountFactor or IP_Encoder2CountFactor (at command

Use_Defaults) is not set, the scaled value is equal to the raw value.

17.5.4.2.6 EnableRef_Encoder

Enable loading encoder value at next reference (set by function Set_EncoderMode with parameter next reference).

Additionally it resets the internal encoder reference flags so the next reference can be detected (by Get_EncoderReference).

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

6= Encoder 1

7= Encoder 2

Description: Number of the encoder for enabling reference.

17.5.4.2.7 Get_EncoderReference

Get status if reference was reached first or second times.

Parameter: int32_t SP_EncoderNumber

SP_EncoderNumber

Direction: Down

Valid values:

6= Encoder 1

7= Encoder 2

Description: Number of the encoder to latch.

Parameter: int32_t SA_FirstReference

SA_FirstReference

Direction: Up

Valid values:

Minimum: 0= no

Maximum: 1= yes

Description: Reference was reached first time.

Parameter: int32_t SA_SecondReference

SA_SecondReference

Direction: Up

Valid values:

Minimum: 0= no

Maximum: 1= yes

Description: Reference was reached second time.

17.5.5 RS422

17.5.5.1 Inputs

17.5.5.1.1 Use_Gate

The digital inputs In 1 to In 4 of the card (5V TTL signal) can be used to lock or free the FIFO for data from specific channels.

Parameter: int32_t SP_GateChannel

SP_GateChannel

Direction: Down

Valid values:

0= Digital In 1 for Sensor channel 1+2 (Base Board, Connector X1)

1= Digital In 1 for Sensor channel 1+2 (Base Board, Connector X1)

2= Digital In 2 for Sensor channel 3-6 (Base Board, Connector X2 + Extension Board, Connector X1)

3= Digital In 2 for Sensor channel 3-6 (Base Board, Connector X2 + Extension Board, Connector X1)

4= Digital In 2 for Sensor channel 3-6 (Base Board, Connector X2 + Extension Board, Connector X1)

5= Digital In 2 for Sensor channel 3-6 (Base Board, Connector X2 + Extension Board, Connector X1)

6= Digital In 3 for Encoder 1+2

7= Digital In 3 for Encoder 1+2

8= Digital In 4 for Digital In+RxD, ADC 1+2

9= Digital In 4 for Digital In+RxD, ADC 1+2

10= Digital In 4 for Digital In+RxD, ADC 1+2

11= Digital In 4 for Digital In+RxD, ADC 1+2

Description: Number of the channel to lock by digital in. Because multiple channels are affected by one digital input, the numbers 0..1, 2..5, 6..7 or 8..11 always affects the same digital input.

Parameter: int32_t SP_ActivateGate SP_ActivateGate

Direction: Down

Valid values:

- 0= off
- 1= on

Description: Specifies if the gate function is enabled or disabled for the specific digital input.

17.5.5.1.2 Get_RxDValue

Get the RxD values.

Parameter: int32_t SA_RxDValue1..6 SA_RxDValue1..6

Direction: Up

Valid values:

- 0
- 1

Description: Value of the RxD line.

17.5.5.2 Outputs

17.5.5.2.1 Set_TxDSource

Set the source which is used for TxD line.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_TxDChannel1..6 SP_TxDChannel1..6

Direction: Down

Valid values:

- 1= Leave unchanged
- 0= Sensor output (transmitter)
- 1= User (output function)

Description: Source which is used for TxD line.

-1 means the channel will not be changed. 0 means the value is set by sensor output (if a sensor command is send to the sensor), 1 means the value is set by function Set_TxDValue.

17.5.5.2.2 Get_TxDSource

Get the source which is used for TxD line.

Parameter: int32_t SA_TxDChannel1..6 SA_TxDChannel1..6

Direction: Up

Valid values:

- 0= sensor output (transmitter)
- 1= user (output function)

Description: Source which is used for TxD line.

17.5.5.2.3 Set_TxDValue

Set the values of TxD lines.

This function does only work when Set_TxDSource is set to 1 (output function).

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_TxDValue_{1..6}

SP_TxDValue_{1..6}

Direction: Down

Valid values:

- 1 = Leave unchanged
- 0
- 1

Description: TxD value.

-1 means leave the value unchanged.

17.5.5.2.4 Get_TxDValue

Get the values of TxD lines.

Parameter: int32_t SA_TxDValue_{1..6}

SA_TxDValue_{1..6}

Direction: Up

Valid values:

- 0
- 1

Description: TxD value.

17.5.5.3 Trigger

17.5.5.3.1 Set_TrgSource

Set the source which is used for Trg line.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_TrgChannel_{1..6}

SP_TrgChannel_{1..6}

Direction: Down

Valid values:

- 1 = Leave unchanged
- 0 = User (output function)
- 1 = Timer 1 pulse width
- 2 = Timer 2 pulse width
- 3 = Timer 3 pulse width
- 4 = Digital input 1
- 5 = Digital input 2
- 6 = Digital input 3
- 7 = Digital input 4

Description: Source which is used used for Trg line.

-1 means the value is not set. 0 means the value is set by function

Set_TrgValue, Timer 1..3 means the value is equal to the timer value.

Digital input is only available from FPGA version 8.

17.5.5.3.2 Get_TrgSource

Get the source which is used used for Trg line.

Parameter: int32_t SA_TrгChannel`1..6`

SA_TrгChannel`1..6`

Direction: Up

Valid values:

- 1= Leave unchanged
- 0= User (output function)
- 1= Timer 1 pulse width
- 2= Timer 2 pulse width
- 3= Timer 3 pulse width
- 4= Digital input 1
- 5= Digital input 2
- 6= Digital input 3
- 7= Digital input 4

Description: Source which is used used for Trg line.

17.5.5.3.3 Set_TrgValue

Set the values of Trg lines.

This function does only work when Set_TrgSource is set to 0 (output function).

For this command an [Update_...](#) meta command is available.

Parameter: int32_t SP_TrгValue`1..6`

SP_TrгValue`1..6`

Direction: Down

Valid values:

- 1= Leave unchanged
- 0
- 1

Description: TxD value.

-1 means leave the value unchanged.

17.5.5.3.4 Get_TrgValue

Get the values of Trg lines.

Parameter: int32_t SA_TrгValue`1..6`

SA_TrгValue`1..6`

Direction: Up

Valid values:

- 0
- 1

Description: Trg value.

17.5.6 Switching inputs

17.5.6.1 Set_DigitalInLatchSource

Set the latch source which triggers aquiring one value.

Attention! This command is only allowed if IF2008E extension card or IF2008IO extension slot is installed

Parameter: int32_t SP_DigitalInLatchSource

SP_DigitalInLatchSource

Direction: Down

Valid values:

- 0= never (locked)
- 1= Timer 1 pulse width
- 2= Timer 2 pulse width
- 3= Timer 3 pulse width
- 4= Sensor channel 1
- 5= Sensor channel 2
- 6= Sensor channel 3
- 7= Sensor channel 4

Description: Latch source which triggers aquiring one value.

17.5.6.2 Get_DigitalInLatchSource

Get the latch source which triggers aquiring one value.

Attention! This command is only allowed if IF2008E extension card or IF2008IO extension slot is installed

Parameter: int32_t SA_DigitalInLatchSource

SA_DigitalInLatchSource

Direction: Up

Valid values:

- 0= never (locked)
- 1= Timer 1 pulse width
- 2= Timer 2 pulse width
- 3= Timer 3 pulse width
- 4= Sensor channel 1
- 5= Sensor channel 2
- 6= Sensor channel 3
- 7= Sensor channel 4

Description: Latch source which triggers aquiring one value.

17.5.6.3 Get_DigitalInValue

Get the digital In values.

Attention! This command is only allowed if IF2008E extension card or IF2008IO extension slot is installed

Parameter: int32_t SA_DigitalInValue_{1..4}

SA_DigitalInValue_{1..4}

Direction: Up

Valid values:

- 0
- 1

Description: Digital In value.

17.5.7 Switching outputs

17.5.7.1 Set_DigitalOutSource

Set the source which is used to output one value.

Attention! This command is only allowed if IF2008E extension card or IF2008IO extension slot is installed.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_DigitalOut1..4

SP_DigitalOut1..4

Direction: Down

Valid values:

- 1= Leave unchanged
- 0= User (output function)
- 1= Timer 1 pulse width
- 2= Timer 2 pulse width
- 3= Timer 3 pulse width

Description: Source which is used to output one value.

-1 means the value is not set. 0 means the value is set by function Set_DigitalOutValue, Timer 1..3 means the value is equal to the timer value.

17.5.7.2 Get_DigitalOutSource

Get the source which is used to output one value.

Attention! This command is only allowed if IF2008E extension card or IF2008IO extension slot is installed

Parameter: int32_t SA_DigitalOut1..4

SA_DigitalOut1..4

Direction: Up

Valid values:

- 0= user (output function)
- 1= Timer 1 pulse width
- 2= Timer 2 pulse width
- 3= Timer 3 pulse width

Description: Source which is used to output one value.

17.5.7.3 Set_DigitalOutValue

Set the digital Out values.

This function does only work when Set_DigitalOutSource is set to 0 (output function)

Attention! This command is only allowed if IF2008E extension card or IF2008IO extension slot is installed.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_DigitalOutValue1..4

SP_DigitalOutValue1..4

Direction: Down

Valid values:

- 1= Leave unchanged
- 0
- 1

Description: Digital Out value. -1 menas leave the value unchanged.

17.5.7.4 Get_DigitalOutValue

Get the digital Out values.

Attention! This command is only allowed if IF2008E extension card or IF2008IO extension slot is installed

Parameter: int32_t SA_DigitalOutValue1..4

SA_DigitalOutValue1..4

Direction: Up

Valid values:

0

1

Description: Digital Out value.

17.5.8 Analog inputs

17.5.8.1 Set_ADCLatchSource

Set the latch source which triggers aquiring one value.

Attention! This command is only allowed if IF2008E extension card is installed

Parameter: int32_t SP_ADCNumber

SP_ADCNumber

Direction: Down

Valid values:

10= ADC 1

11= ADC 2

Description: Number of the ADC (Analog/Digital converter) to parametrize.

Parameter: int32_t SP_ADCLatchSource

SP_ADCLatchSource

Direction: Down

Valid values:

0= never (locked)

1= Timer 1

2= Timer 2

3= Timer 3

4= Sensor channel 1

5= Sensor channel 2

6= Sensor channel 3

7= Sensor channel 4

8= Sensor channel 5

9= Sensor channel 6

10= Digital In 1

11= Digital In 2

12= Digital In 3

13= Digital In 4

Description: Latch source which triggers aquiring one value.

17.5.8.2 Get_ADCLatchSource

Get the latch source which triggers aquiring one value.

Attention! This command is only allowed if IF2008E extension card is installed

Parameter: int32_t SP_ADCNumber

SP_ADCNumber

Direction: Down

Valid values:

10= ADC 1
11= ADC 2

Description: Number of the ADC (Analog/Digital converter) to parametrize.

Parameter: int32_t SA_ADCLatchSource

SA_ADCLatchSource

Direction: Up

Valid values:

0= never (locked)
1= Timer 1
2= Timer 2
3= Timer 3
4= Sensor channel 1
5= Sensor channel 2
6= Sensor channel 3
7= Sensor channel 4
8= Sensor channel 5
9= Sensor channel 6
10= Digital In 1
11= Digital In 2
12= Digital In 3
13= Digital In 4

Description: Latch source which triggers aquiring one value.

17.5.8.3 Get_ADCValue

Get the value of the ADC.

Attention! This command is only allowed if IF2008E extension card is installed

Parameter: int32_t SP_ADCNumber

SP_ADCNumber

Direction: Down

Valid values:

10= ADC 1
11= ADC 2

Description: Number of the ADC to return value.

Parameter: int32_t SA_ADCRawValue

SA_ADCRawValue

Direction: Up

Valid values:

Minimum: 0
Maximum: 65535

Description: Raw value of ADC.

Parameter: double SA_ADCScaledValue SA_ADCScaledValue
Direction: Up
Valid values:
 Minimum: 0, -5 or -10, depending on analog range
 Maximum: 5 or 10, depending on analog range
Unit: V
Description: Scaled value of ADC.
 The range depends on IP_ADC1Range or IP_ADC2Range (at command Use_Defaults).

17.6 Commands for EthAdapter_IF2008

See IF2008/ETH manual for detailed description of sensor commands.

Attention! The command [Use_Defaults](#) must be called if encoder values should be scaled.

This sensor supports following interfaces:

- [IF2008_ETH](#) (native).

To allow MEDAQLib to scale data, it is recommended to call sensor command [Use_Defaults](#) after [OpenSensor](#).

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from IF2008/ETH card, from 0 to 4294967295 (Encoder) or 15 (Digital in).
- Scaled values are scaled using count factors (if known by MEDAQLib, use sensor command [Use_Defaults](#)).

17.6.1 General commands

17.6.1.1 Internal commands

17.6.1.1.1 [Use_Defaults](#)

This command parametrizes the IF2008/ETH adapter. If same parameters are not specified they are not changed.

Parameter: double IP_EncoderCountFactor IP_EncoderCountFactor
Direction: Down
Valid for sensor:
 ETH_ADAPTER_IF2008
Valid values:
 Minimum: -1.79769e+308 (-DBL_MAX)
 Maximum: 1.79769e+308 (DBL_MAX)
Default: 1.0
Description: It is the distance per count of the encoder used by the driver for scaling data.

17.6.1.1.2 Get_DrvSetting

Returns the current settings of the driver used for operating. It is the opposite of Use_Defaults.

Parameter: double IA_EncoderCountFactor IA_EncoderCountFactor
Direction: Up
Valid for sensor:
 ETH_ADAPTER_IF2008
Valid values:
Minimum: -1.79769e+308 (-DBL_MAX)
Maximum: 1.79769e+308 (DBL_MAX)
Description: It is the distance per count of the encoder used by the driver for scaling data.

17.6.1.1.3 Set_SyncMaster

Set a synchronization master for a specified sensor channel while interface is open.

Parameter: int32_t SP_SensorChannel SP_SensorChannel
Direction: Down
Valid values:
 0= Sensor/Encoder channel 1
 1= Sensor/Encoder channel 2
 2= Sensor/Encoder channel 3
 3= Sensor/Encoder channel 4
 4= Sensor/Encoder channel 5
 5= Sensor/Encoder channel 6
 6= Sensor/Encoder channel 7
 7= Sensor/Encoder channel 8
Description: Sensor channel to synchronize.

Parameter: int32_t SP_SyncMasterChannel SP_SyncMasterChannel
Direction: Down
Valid values:
 -1= No synchronization master
 0= Sensor/Encoder channel 1
 1= Sensor/Encoder channel 2
 2= Sensor/Encoder channel 3
 3= Sensor/Encoder channel 4
 4= Sensor/Encoder channel 5
 5= Sensor/Encoder channel 6
 6= Sensor/Encoder channel 7
 7= Sensor/Encoder channel 8
 8= Digital IN
Description: Channel number to synchronize with. See [IP_SyncMasterChannel](#) for further information.

17.6.1.2 Information

17.6.1.2.1 Get_Info (GETINFO)

Retrieve information about the adapter.

Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Description: Name of the adapter.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the adapter.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the adapter.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the adapter.	
Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description: MAC address (low level ethernet address) of the adapter.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the adapter.	
Parameter: String SA_FPGAVersion	SA_FPGAVersion
Direction: Up	
Description: FPGA version in the adapter.	
Parameter: String SA_WebpageVersion	SA_WebpageVersion
Direction: Up	
Description: Version of the Web pages in the adapter.	

17.6.1.2.2 Get_AllParameters (PRINT)

Get all parameters from adapter.

If SP_Additional is 1 there also may be contained the answer parameters of [Get_Info](#).

Parameter: int32_t SP_Additional	SP_Additional
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description: If set, additional information about controller and sensors are output.	

Parameter: String SA_Sensor	SA_Sensor
Direction: Up	
Description: Name of the adapter.	
Parameter: String SA_SerialNumber	SA_SerialNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Serial number of the adapter.	
Parameter: String SA_Option	SA_Option
Direction: Up	
Valid values:	
Numeric value	
Description: Option of the adapter.	
Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the adapter.	
Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description: MAC address (low level ethernet address) of the adapter.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the adapter.	
Parameter: String SA_FPGAVersion	SA_FPGAVersion
Direction: Up	
Description: FPGA version in the adapter.	
Parameter: String SA_WebpageVersion	SA_WebpageVersion
Direction: Up	
Description: Version of the Web pages in the adapter.	
Parameter: int32_t SA_DHCPEnabled	SA_DHCPEnabled
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = FALSE	
1 = TRUE	
Description: Get settings if adapter should use a static IP address ask for IP at DHCP server (dynamic IP address).	
Parameter: String SA_Address	SA_Address
Direction: Up	
Valid values:	
Valid IP address in form of xxx.xxx.xxx.xxx	
Description: IP address of the adapter. If DHCP is enabled it returns the currently assigned IP address.	

Parameter: String SA_SubnetMask	SA_SubnetMask
Direction: Up	
Valid values:	Valid network mask (e.g. 255.255.255.0 for a Class C network)
Description:	Network mask of the adapter. If DHCP is enabled it returns the currently assigned network mask.
Parameter: String SA_Gateway	SA_Gateway
Direction: Up	
Valid values:	Valid IP address of default gateway in form of xxx.xxx.xxx.xxx
Description:	Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.
Parameter: int32_t SA_Protocol	SA_Protocol
Direction: Up	
Valid values:	-1= Unknown parameter value from sensor 0= TCP server (SERVER/TCP)
Description:	Only one protocol is supported.
Parameter: int32_t SA_Port	SA_Port
Direction: Up	
Valid values:	Minimum: 1024 Maximum: 65535
Description:	Port to send data to or to listen for incoming requests.
Parameter: int32_t SA_FramesPerPacket_ETH	SA_FramesPerPacket_ETH
Direction: Up	
Valid values:	Minimum: 0 Maximum: 716
Description:	Maximum number of frames in ethernet packet. 0 means automatic, so each 10 ms a packet is output. For other values, the adapter wait until enough tuples are captured. If packets cannot be send fast enough, this value may be exceeded.
Parameter: String SA_ApplicationLanguage	SA_ApplicationLanguage
Direction: Up	
Valid values:	BROWSER ENGLISH GERMAN Or any other language (from Firmware version 2.13).
Description:	Language of web interface.
Parameter: int32_t SA_ChannelMode1..8	SA_ChannelMode1..8
Direction: Up	
Valid values:	0= None 1= Encoder 2= Sensor
Description:	Operation mode of the channel. If it is set to encoder, Set_EncoderLatchSource<n> must be called to enable automatic data acquisition.

Parameter: double SA_TimerFrequency1..3	SA_TimerFrequency1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 12000000.0	
Unit: Hz	
Description: Frequency of the timer.	
0.0 means the timer is off.	
Parameter: double SA_TimerRatio1..3	SA_TimerRatio1..3
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 1.0	
Description: Ratio of the high and low contingent of one period.	
If SA_TimerFrequency is 0.0, a value of 0.0 means the timer is low, otherwise the timer is high.	
Parameter: int32_t SA_SensorBaudrate1..8	SA_SensorBaudrate1..8
Direction: Up	
Valid values:	
Minimum: 9600	
Maximum: 8000000	
Unit: Baud	
Description: Baudrate of sensor channel.	
Parameter: int32_t SA_LaserPower1..8	SA_LaserPower1..8
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Off	
1 = On	
Description: Laser power.	
Parameter: int32_t SA_TriggerOutput1..8	SA_TriggerOutput1..8
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Low	
1 = High	
2 = Timer1	
3 = Timer2	
4 = Timer3	
5 = Input1	
6 = Input2	
7 = Input3	
8 = Input4	
Description: Source of trigger/synchronisation source line.	
Parameter: int32_t SA_SensorErrorBits	SA_SensorErrorBits
Direction: Up	
Valid values:	
Bit combination of 8 bits	
Description: The error state of the sensors.	

Parameter: int32_t SA_EncoderInterpolation1..8	SA_EncoderInterpolation1..8
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Counter	
1= Single (1)	
2= Double (2)	
3= Quadruple (4)	
Description: Encoder interpolation.	
Parameter: int32_t SA_EncoderMode1..8	SA_EncoderMode1..8
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= No action (NONE)	
1= Set encoder value to preset value only one time (ONE)	
2= Set encoder value to preset value each time (EVER)	
3= Encoder is limited between 0 and encoder preset value (LIMIT), see Set_EncoderPreload<n> .	
Description: Mode of encoder when reference is reached.	
Parameter: double SA_EncoderPreload1..8	SA_EncoderPreload1..8
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: int32_t SA_EncoderDirection1..8	SA_EncoderDirection1..8
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Normal	
1= Reverse	
Description: Counting direction of encoder.	
Parameter: int32_t SA_EncoderLatchSource1..8	SA_EncoderLatchSource1..8
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Timer1	
2= Timer2	
3= Timer3	
4= Sensor1	
5= Sensor2	
6= Sensor3	
7= Sensor4	
8= Input1	
9= Input2	
10= Input3	
11= Input4	
12= Second reference	
13= Any reference	
Description: Latch source of encoder.	

Parameter: double SA_EncoderValue1..8	SA_EncoderValue1..8
Direction: Up	
Valid values:	
Minimum: 0.0	
Maximum: 4294967295.0 (UINT32_MAX)	
Description: Encoder value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.	
Parameter: int32_t SA_EncoderReference1..8	SA_EncoderReference1..8
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= First	
2= Second	
Description: Encoder reference count.	
Parameter: int32_t SA_DigitalLevel	SA_DigitalLevel
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Low level logic (LLL)	
1= High level logic (HLL)	
Description: Digital I/O logic level.	
Parameter: int32_t SA_DigitalInLatchSource	SA_DigitalInLatchSource
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= never (NONE)	
1= Timer 1	
2= Timer 2	
3= Timer 3	
4= Sensor channel 1	
5= Sensor channel 2	
6= Sensor channel 3	
7= Sensor channel 4	
Description: Latch source which triggers reading digital inputs.	
Parameter: int32_t SA_DigitalInValueValue	SA_DigitalInValueValue
Direction: Up	
Valid values:	
Bit combination of 4 bits	
Description: Digital input value.	
Parameter: int32_t SA_DigitalInMode1	SA_DigitalInMode1
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Laser power (LASERPOW)	
Description: Special function for digital input 1.	

Parameter: int32_t SA_DigitalInMode2	SA_DigitalInMode2
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = None	
1 = FIFO gate for sensor/encoder channels 1 to 4 (FIFOGATE)	
Description: Special function for digital input 2.	
Parameter: int32_t SA_DigitalInMode3	SA_DigitalInMode3
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = None	
1 = FIFO gate for sensor/encoder channels 5 to 8 (FIFOGATE)	
Description: Special function for digital input 3.	
Parameter: int32_t SA_DigitalOutSource1..4	SA_DigitalOutSource1..4
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Low	
1 = High	
2 = Timer1	
3 = Timer2	
4 = Timer3	
Description: Digital output source.	

17.6.1.3 Interfaces

17.6.1.3.1 Set_IPConfiguration (IPCONFIG)

Set the IP configuration at adapter.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_DHCPEnabled	SP_DHCPEnabled
Direction: Down	
Valid values:	
0 = FALSE	
1 = TRUE	
Description: Specify if adapter should use a static IP address or ask for IP at DHCP server (dynamic IP address).	
Parameter: String SP_Address	SP_Address
Direction: Down	
Valid values:	
Valid IP address in form of xxx.xxx.xxx.xxx	
Description: IP address of the adapter. This parameter is only evaluated on static IP assignment.	
Parameter: String SP_SubnetMask	SP_SubnetMask
Direction: Down	
Valid values:	
Valid network mask (e.g. 255.255.255.0 for a Class C network)	
Description: Network mask of the adapter. This parameter is only evaluated on static IP assignment.	

Parameter: String SP_Gateway SP_Gateway

Direction: Down

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: The default gateway must be specified if the adapter should communicate with peers in foreign subnets. This parameter is only evaluated on static IP assignment.

17.6.1.3.2 Get_IPConfiguration (IPCONFIG)

Get the IP configuration at adapter.

Parameter: int32_t SA_DHCPEnabled SA_DHCPEnabled

Direction: Up

Valid values:

-1 = Unknown parameter value from sensor

0 = FALSE

1 = TRUE

Description: Get settings if adapter should use a static IP address ask for IP at DHCP server (dynamic IP address).

Parameter: String SA_Address SA_Address

Direction: Up

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the adapter. If DHCP is enabled it returns the currently assigned IP address.

Parameter: String SA_SubnetMask SA_SubnetMask

Direction: Up

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the adapter. If DHCP is enabled it returns the currently assigned network mask.

Parameter: String SA_Gateway SA_Gateway

Direction: Up

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

17.6.1.3.3 Set_IPDataTransferMode (MEASTRANSFER)

Set IP protocol at adapter.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_Protocol SP_Protocol

Direction: Down

Valid values:

0 = TCP server (SERVER/TCP)

Description: Only one protocol is supported.

17.6. Commands for EthAdapter_IF2008

Parameter: int32_t SP_Port SP_Port
Direction: Down
Valid values:
Minimum: 1024
Maximum: 65535
Description: Port to send data to or to listen for incoming requests.

17.6.1.3.4 Get_IPDataTransferMode (MEATRANSFER)

Get IP protocol at adapter.

Parameter: int32_t SA_Protocol SA_Protocol
Direction: Up
Valid values:
-1 = Unknown parameter value from sensor
0 = TCP server (SERVER/TCP)
Description: Only one protocol is supported.

Parameter: int32_t SA_Port SA_Port
Direction: Up
Valid values:
Minimum: 1024
Maximum: 65535
Description: Port to send data to or to listen for incoming requests.

17.6.1.3.5 Set_FramesPerPacket_ETH (MEASCNT_ETH)

Set the desired number of data tuples in ethernet packet. Each tuple consists of an address (channel, byte counter, ...) and a data byte.

Parameter: int32_t SP_FramesPerPacket_ETH SP_FramesPerPacket_ETH
Direction: Down
Valid values:
Minimum: 0
Maximum: 716
Description: Maximum number of tuples in ethernet packet. 0 means automatic, so each 10 ms a packet is output. For other values, the adapter wait until enough tuples are captured. If packets cannot be send fast enough, this value may be exceeded.

17.6.1.3.6 Get_FramesPerPacket_ETH (MEASCNT_ETH)

Get the desired number of data tuples in ethernet packet. Each tuple consists of an address (channel, byte counter, ...) and a data byte.

Parameter: int32_t SA_FramesPerPacket_ETH SA_FramesPerPacket_ETH
Direction: Up
Valid values:
Minimum: 0
Maximum: 716
Description: Maximum number of tuples in ethernet packet. 0 means automatic, so each 10 ms a packet is output. For other values, the adapter wait until enough tuples are captured. If packets cannot be send fast enough, this value may be exceeded.

17.6.1.3.7 Set_ApplLanguage (LANGUAGE)

Set language of web interface.

Parameter: String SP_ApplicationLanguage

SP_ApplicationLanguage

Direction: Down

Valid values:

BROWSER

ENGLISH

GERMAN

Or any other language (from Firmware version 2.13).

Description: Language of web interface.

17.6.1.3.8 Get_ApplLanguage (LANGUAGE)

Get language of web interface.

Parameter: String SA_ApplicationLanguage

SA_ApplicationLanguage

Direction: Up

Valid values:

BROWSER

ENGLISH

GERMAN

Or any other language (from Firmware version 2.13).

Description: Language of web interface.

17.6.1.3.9 Set_ChannelMode<n> (CHANNELMODE<n>)

Set operation mode for channel $n \in \{1..8\}$ on the adapter.

This command is normally not used by customer. MEDAQLib calls it automatically when any sensor or [ETH_ADAPTER_IF2008](#) instance is opened or closed at this channel.

Parameter: int32_t SP_ChannelMode<n>

SP_ChannelMode<n>

Direction: Down

Valid values:

0= None

1= Encoder

2= Sensor

Description: Operation mode of the channel.

If it is set to encoder, [Set_EncoderLatchSource<n>](#) must be called to enable automatic data acquisition.

Attention! MEDAQLib will automatically changes this parameter when any sensor or [ETH_ADAPTER_IF2008](#) instance is opened or closed at this channel. To avoid this, open [ETH_ADAPTER_IF2008](#) instance with [IP_ChannelNumber](#) set to -1.

17.6.1.3.10 Get_ChannelMode<n> (CHANNELMODE<n>)

Get operation mode for channel n $\in \{1..8\}$ on the adapter.

Parameter: int32_t SA_ChannelMode<n>

SA_ChannelMode<n>

Direction: Up

Valid values:

0= None

1= Encoder

2= Sensor

Description: Operation mode of the channel.

If it is set to encoder, [Set_EncoderLatchSource<n>](#) must be called to enable automatic data acquisition.

17.6.1.4 Timer

17.6.1.4.1 Set_TimerFrequency<n> (TIMERFREQUENCY<n>)

Set the frequency for the timer n $\in \{1..3\}$ on the adapter.

Parameter: double SP_TimerFrequency<n>

SP_TimerFrequency<n>

Direction: Down

Valid values:

Minimum: 0.000

Maximum: 12000000.0

Unit: Hz

Description: Frequency of the timer.

0.0 means the timer is off.

17.6.1.4.2 Get_TimerFrequency<n> (TIMERFREQUENCY<n>)

Get the frequency for the timer n $\in \{1..3\}$ on the adapter.

Parameter: double SA_TimerFrequency<n>

SA_TimerFrequency<n>

Direction: Up

Valid values:

Minimum: 0.000

Maximum: 12000000.0

Unit: Hz

Description: Frequency of the timer.

0.0 means the timer is off.

17.6.1.4.3 Set_TimerRatio<n> (TICKERPULSEWIDTH<n>)

Set the pulse width for the timer n $\in \{1..3\}$ on the adapter.

Parameter: double SP_TimerRatio<n>

SP_TimerRatio<n>

Direction: Down

Valid values:

Minimum: 0.0

Maximum: 1.0

Description: Ratio of the high and low contingent of one period.

If you are unsure, specify a value of 0.5.

If SP_TimerFrequency is 0.0, a value of 0.0 means the timer is low, otherwise the timer is high.

17.6.1.4.4 Get_TimerRatio<n> (TIMERPULSEWIDTH<n>)

Get the pulse width for the timer n ∈ {1..3} on the adapter.

Parameter: double SA_TimerRatio<n>

SA_TimerRatio<n>

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 1.0

Description: Ratio of the high and low contingent of one period.

If SA_TimerFrequency is 0.0, a value of 0.0 means the timer is low, otherwise the timer is high.

17.6.1.5 Parameter management

17.6.1.5.1 Save_Parameters (STORE)

Save actual parameters at adapter. There can be saved several settings on different locations. So it is easy to switch to another setting.

Parameter: int32_t SP_ParameterSet

SP_ParameterSet

Direction: Down

Valid values:

Minimum: 1

Maximum: 8

Description: Location to save the settings.

17.6.1.5.2 Load_Parameters (READ)

Load stored parameters into adapter RAM.

There can be loaded several settings from different locations. So it is easy to switch to another setting.

At this command the adapter may change output data after applying new setting.

If first bit of IP_AutomaticMode is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually.

After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_ParameterType

SP_ParameterType

Direction: Down

Valid values:

0= All settings (ALL)

1= Device settings (DEVICE)

2= Measurement settings (MEAS)

Description: Specifies which settings should be loaded.

Parameter: int32_t SP_ParameterSet

SP_ParameterSet

Direction: Down

Valid values:

Minimum: 1

Maximum: 8

Description: Location from where the settings should be loaded.

17.6.1.5.3 Set_Default (SETDEFAULT)

Reset the adapter to default settings.

At this command the sensor may change output data after applying default settings. If first bit of `IP_AutomaticMode` is set (1), `Get_AllParameters` (`SP_Additional= 1`) is called automatically after this command. Otherwise, you have to call it manually. After this call `Get_TransmittedDataInfo` to retrieve new setting.

Parameter: int32_t `SP_JustActualSetup`

`SP_JustActualSetup`

Direction: Down

Valid values:

0= Reset all settings (ALL)

1= Just reset actual setting

Description: Specifies which settings should be resetted.

Parameter: int32_t `SP_KeepDevice`

`SP_KeepDevice`

Direction: Down

Valid values:

0= Reset actual interface parameters

1= Keep device settings temporary (NODEVICE)

Description: Specifies if parameters should be held temporary.

17.6.1.5.4 Reset_Boot (RESET)

Resets the adapter.

At this command the adapter may change output data after reboot. If first bit of `IP_AutomaticMode` is set (1), `Get_AllParameters` (`SP_Additional= 1`) is called automatically after this command. Otherwise, you have to call it manually. After this call `Get_TransmittedDataInfo` to retrieve new setting.

17.6.2 Sensor

17.6.2.1 Settings

17.6.2.1.1 Set_Baudrate<n> (BAUDRATE<n>)

Set baudrate of sensor channel n $\in \{1..8\}$ on the adapter.

This command is normally not used by customer. MEDAQLib calls it automatically when any sensor instance is opened at this channel.

Parameter: int32_t `SP_SensorBaudrate<n>`

`SP_SensorBaudrate<n>`

Direction: Down

Valid values:

Minimum: 9600

Maximum: 8000000

Unit: Baud

Description: Baudrate of sensor channel.

Attention! MEDAQLib will automatically change this parameter when any sensor instance is opened at this channel. To avoid this, open `ETH_ADAPTER_IF2008` instance with `IP_ChannelNumber` set to -1.

Parameter: int32_t CP_InterruptDataTransfer CP_InterruptDataTransfer
Direction: Down
Valid values:
 0= false
 1= true
Default: 0
Description: After switch baudrate, synchronization may be impossible if serial line has high load.
 Enabling this parameter resolves the problem by disabling data transfer first ([Set_ChannelMode<n>](#)) and enabling it again at end (only if enabled before).

17.6.2.1.2 Get_Baudrate<n> (BAUDRATE<n>)

Get baudrate of sensor channel n $\in \{1..8\}$ on the adapter.

Parameter: int32_t SA_SensorBaudrate<n> SA_SensorBaudrate<n>
Direction: Up
Valid values:
Minimum: 9600
Maximum: 8000000
Unit: Baud
Description: Baudrate of sensor channel.

17.6.2.1.3 Set_LaserPower<n> (LASERPOW<n>)

Set the laser state (if supported by connected sensor) for sensor channel n $\in \{1..8\}$.
 A special line between IF2008/ETH and sensor is used to switch the laser at sensor.

Parameter: int32_t SP_LaserPower<n> SP_LaserPower<n>
Direction: Down
Valid values:
 0= Off
 1= On
Description: Laser power.

17.6.2.1.4 Get_LaserPower<n> (LASERPOW<n>)

Get the laser state for sensor channel n $\in \{1..8\}$.
 A special line between IF2008/ETH and sensor is used to switch the laser at sensor.

Parameter: int32_t SA_LaserPower<n> SA_LaserPower<n>
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Off
 1= On
Description: Laser power.

17.6.2.1.5 Set_TriggerOutput<n> (TRIGGEROUTPUT<n>)

Set the trigger/synchronisation source (if supported by connected sensor) for sensor channel n ∈ {1..8}.

A special line between IF2008/ETH and sensor is used to trigger/synchronize the sensor.

Parameter: int32_t SP_TriggerOutput<n>

SP_TriggerOutput<n>

Direction: Down

Valid values:

- 0= Low
- 1= High
- 2= Timer1
- 3= Timer2
- 4= Timer3
- 5= Input1
- 6= Input2
- 7= Input3
- 8= Input4

Description: Source of trigger/synchronisation source line.

17.6.2.1.6 Get_TriggerOutput<n> (TRIGGEROUTPUT<n>)

Get the trigger/synchronisation source for sensor channel n ∈ {1..8}.

A special line between IF2008/ETH and sensor is used to trigger/synchronize the sensor.

Parameter: int32_t SA_TriggerOutput<n>

SA_TriggerOutput<n>

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Low
- 1= High
- 2= Timer1
- 3= Timer2
- 4= Timer3
- 5= Input1
- 6= Input2
- 7= Input3
- 8= Input4

Description: Source of trigger/synchronisation source line.

17.6.2.2 Actions

17.6.2.2.1 Get_SensorErrorInputs (SENSORERROR)

Retrieve state of sensor error lines (if supported by connected sensor) for all sensor channels.

A special line between IF2008/ETH and sensor is used to get sensor error state.

Parameter: int32_t SA_SensorErrorBits

SA_SensorErrorBits

Direction: Up

Valid values:

Bit combination of 8 bits

Description: The error state of the sensors.

17.6.2.2.2 SensorCommand<n> (TUNNEL<n>)

Send a command directly to the attached sensor at sensor channel n $\in \{1..8\}$.
 This command is normally not used by customer. MEDAQLib calls it automatically when any sensor instance processes a sensor command over this channel.

Parameter: Binary data SP_Command<n>

SP_Command<n>

Direction: Down

Description: Generic sensor command. For sensors using ASCII protocol, command must be postfixed by <CR><LF>.

17.6.3 Encoder

17.6.3.1 Settings

17.6.3.1.1 Set_EncoderInterpolation<n> (ENCINTERPOL<n>)

Set the interpolation of encoder channel n $\in \{1..8\}$.

Parameter: int32_t SP_EncoderInterpolation<n>

SP_EncoderInterpolation<n>

Direction: Down

Valid values:

- 0= Counter
- 1= Single (1)
- 2= Double (2)
- 3= Quadruple (4)

Description: Encoder interpolation.

17.6.3.1.2 Get_EncoderInterpolation<n> (ENCINTERPOL<n>)

Get the interpolation of encoder channel n $\in \{1..8\}$.

Parameter: int32_t SA_EncoderInterpolation<n>

SA_EncoderInterpolation<n>

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Counter
- 1= Single (1)
- 2= Double (2)
- 3= Quadruple (4)

Description: Encoder interpolation.

17.6.3.1.3 Set_EncoderMode<n> (ENCREF<n>)

Set the behaviour of encoder channel n $\in \{1..8\}$ when reference is reached.

Parameter: int32_t SP_EncoderMode<n>

SP_EncoderMode<n>

Direction: Down

Valid values:

- 0= No action (NONE)
- 1= Set encoder value to preset value only one time (ONE)
- 2= Set encoder value to preset value each time (EVER)
- 3= Encoder is limited between 0 and encoder preset value (LIMIT), see
[Set_EncoderPreload<n>](#).

Description: Mode of encoder when reference is reached.

17.6.3.1.4 Get_EncoderMode<n> (ENCREF<n>)

Get the behaviour of encoder channel n ∈ {1..8} when reference is reached.

Parameter: int32_t SA_EncoderMode<n>

SA_EncoderMode<n>

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = No action (NONE)
- 1 = Set encoder value to preset value only one time (ONE)
- 2 = Set encoder value to preset value each time (EVER)
- 3 = Encoder is limited between 0 and encoder preset value (LIMIT), see [Set_EncoderPreload<n>](#).

Description: Mode of encoder when reference is reached.

17.6.3.1.5 Set_EncoderPreload<n> (ENCVALUE<n>)

Set preload value of encoder channel n ∈ {1..8} on the adapter.

Parameter: double SP_EncoderPreload<n>

SP_EncoderPreload<n>

Direction: Down

Valid values:

- Minimum:** 0.0
- Maximum:** 4294967295.0 (UINT32_MAX)

Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

17.6.3.1.6 Get_EncoderPreload<n> (ENCVALUE<n>)

Get preload value of encoder channel n ∈ {1..8} on the adapter.

Parameter: double SA_EncoderPreload<n>

SA_EncoderPreload<n>

Direction: Up

Valid values:

- Minimum:** 0.0
- Maximum:** 4294967295.0 (UINT32_MAX)

Description: Preload value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

17.6.3.1.7 Set_EncoderDirection<n> (ENCDIR<n>)

Set counting direction of encoder channel n ∈ {1..8} on the adapter.

Parameter: int32_t SP_EncoderDirection<n>

SP_EncoderDirection<n>

Direction: Down

Valid values:

- 0 = Normal
- 1 = Reverse

Description: Counting direction of encoder.

17.6.3.1.8 Get_EncoderDirection<n> (ENCDIR<n>)

Get counting direction of encoder channel n ∈ {1..8} on the adapter.

Parameter: int32_t SA_EncoderDirection<n>

SA_EncoderDirection<n>

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Normal
- 1= Reverse

Description: Counting direction of encoder.

17.6.3.1.9 Set_EncoderLatchSource<n> (ENCLATCHSRC<n>)

Set latch source of encoder channel n ∈ {1..8} on the adapter.

Parameter: int32_t SP_EncoderLatchSource<n>

SP_EncoderLatchSource<n>

Direction: Down

Valid values:

- 0= None
- 1= Timer1
- 2= Timer2
- 3= Timer3
- 4= Sensor1
- 5= Sensor2
- 6= Sensor3
- 7= Sensor4
- 8= Input1
- 9= Input2
- 10= Input3
- 11= Input4
- 12= Second reference
- 13= Any reference

Description: Latch source of encoder.

17.6.3.1.10 Get_EncoderLatchSource<n> (ENCLATCHSRC<n>)

Get latch source of encoder channel n ∈ {1..8} on the adapter.

Parameter: int32_t SA_EncoderLatchSource<n>

SA_EncoderLatchSource<n>

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Timer1
- 2= Timer2
- 3= Timer3
- 4= Sensor1
- 5= Sensor2
- 6= Sensor3
- 7= Sensor4

8= Input1
 9= Input2
 10= Input3
 11= Input4
 12= Second reference
 13= Any reference

Description: Latch source of encoder.

17.6.3.2 Actions

17.6.3.2.1 Load_Encoder< n > (ENCSET)

Load the encoder $n \in \{1..8\}$ with the preset value.

17.6.3.2.2 EnableRef_Encoder< n > (ENCRESET)

Reset reference counter of encoder $n \in \{1..8\}$. If Set_EncoderMode< n > is set to 1 (ONE) or 2 (EVER), at next reference encoder value will be set to preset value again

17.6.3.2.3 Clear_Encoder< n > (ENCCLEAR)

Clear the encoder $n \in \{1..8\}$ (to value 0).

17.6.3.2.4 Get_EncoderValue< n > (GETENCVALUE< n >)

Get the value of encoder $n \in \{1..8\}$.

Parameter: double SA_EncoderValue< n >

SA_EncoderValue< n >

Direction: Up

Valid values:

Minimum: 0.0

Maximum: 4294967295.0 (UINT32_MAX)

Description: Encoder value. This value is in whole numbers. But to avoid (signed) integer overflow, it's type is double.

17.6.3.2.5 Get_EncoderReference< n > (GETENCREF< n >)

Get the reference count of encoder $n \in \{1..8\}$.

Parameter: int32_t SA_EncoderReference< n >

SA_EncoderReference< n >

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= None

1= First

2= Second

Description: Encoder reference count.

17.6.4 Digital I/O

17.6.4.1 General

17.6.4.1.1 Set_DigitalLevel (EXTLEVEL)

Set the logic level of c.

Parameter: int32_t SP_DigitalLevel

SP_DigitalLevel

Direction: Down

Valid values:

0= Low level logic (LLL)

1= High level logic (HLL)

Description: Digital I/O logic level.

17.6.4.1.2 Get_DigitalLevel (EXTLEVEL)

Get the logic level of digital I/O.

Parameter: int32_t SA_DigitalLevel

SA_DigitalLevel

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Low level logic (LLL)

1= High level logic (HLL)

Description: Digital I/O logic level.

17.6.4.2 Switching inputs

17.6.4.2.1 Set_DigitalInLatchSource (EXTINLATCHSRC)

Set the latch source digital inputs are written into FIFO of IF2008/ETH adapter.

Parameter: int32_t SP_DigitalInLatchSource

SP_DigitalInLatchSource

Direction: Down

Valid values:

0= never (NONE)

1= Timer 1

2= Timer 2

3= Timer 3

4= Sensor channel 1

5= Sensor channel 2

6= Sensor channel 3

7= Sensor channel 4

Description: Latch source which triggers reading digital inputs.

17.6.4.2.2 Get_DigitalInLatchSource (EXTINLATCHSRC)

Get the latch source digital inputs are written into FIFO of IF2008/ETH adapter.

Parameter: int32_t SA_DigitalInLatchSource

SA_DigitalInLatchSource

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= never (NONE)
- 1= Timer 1
- 2= Timer 2
- 3= Timer 3
- 4= Sensor channel 1
- 5= Sensor channel 2
- 6= Sensor channel 3
- 7= Sensor channel 4

Description: Latch source which triggers reading digital inputs.

17.6.4.2.3 Get_DigitalInValue (GETEXTINPUT)

Get the value of digital inputs.

Parameter: int32_t SA_DigitalInValueValue

SA_DigitalInValueValue

Direction: Up

Valid values:

- Bit combination of 4 bits

Description: Digital input value.

17.6.4.2.4 Set_DigitalInMode1 (EXTINPUTMODE1)

Set special function of digital input 1.

Parameter: int32_t SP_DigitalInMode1

SP_DigitalInMode1

Direction: Down

Valid values:

- 0= None
- 1= Laser power (LASERPOW)

Description: Special function for digital input 1.

17.6.4.2.5 Get_DigitalInMode1 (EXTINPUTMODE1)

Get special function of digital input 1.

Parameter: int32_t SA_DigitalInMode1

SA_DigitalInMode1

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Laser power (LASERPOW)

Description: Special function for digital input 1.

17.6.4.2.6 Set_DigitalInMode2 (EXTINPUTMODE2)

Set special function of digital input 2.

Parameter: int32_t SP_DigitalInMode2

SP_DigitalInMode2

Direction: Down

Valid values:

0= None

1= FIFO gate for sensor/encoder channels 1 to 4 (FIFOGATE)

Description: Special function for digital input 2.

17.6.4.2.7 Get_DigitalInMode2 (EXTINPUTMODE2)

Get special function of digital input 2.

Parameter: int32_t SA_DigitalInMode2

SA_DigitalInMode2

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= None

1= FIFO gate for sensor/encoder channels 1 to 4 (FIFOGATE)

Description: Special function for digital input 2.

17.6.4.2.8 Set_DigitalInMode3 (EXTINPUTMODE3)

Set special function of digital input 3.

Parameter: int32_t SP_DigitalInMode3

SP_DigitalInMode3

Direction: Down

Valid values:

0= None

1= FIFO gate for sensor/encoder channels 5 to 8 (FIFOGATE)

Description: Special function for digital input 3.

17.6.4.2.9 Get_DigitalInMode3 (EXTINPUTMODE3)

Get special function of digital input 3.

Parameter: int32_t SA_DigitalInMode3

SA_DigitalInMode3

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= None

1= FIFO gate for sensor/encoder channels 5 to 8 (FIFOGATE)

Description: Special function for digital input 3.

17.6.4.3 Switching outputs

17.6.4.3.1 Set_DigitalOutSource< n > (EXTOUTSRC< n >)

Set the source for digital output $n \in \{1..4\}$.

Parameter: int32_t SP_DigitalOutSource< n >

SP_DigitalOutSource< n >

Direction: Down

Valid values:

- 0= Low
- 1= High
- 2= Timer1
- 3= Timer2
- 4= Timer3

Description: Digital output source.

17.6.4.3.2 Get_DigitalOutSource< n > (EXTOUTSRC< n >)

Get the source for digital output $n \in \{1..4\}$.

Parameter: int32_t SA_DigitalOutSource< n >

SA_DigitalOutSource< n >

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Low
- 1= High
- 2= Timer1
- 3= Timer2
- 4= Timer3

Description: Digital output source.

17.7 Commands for CSP2008

See sensor manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (additional, e.g. IF2001_USB (RS422) and RS232 high level interface).
- [IF2004](#) (native).
- [TCP/IP](#) (native).
- [IF2004_USB](#) (native).
- [IF2008](#) (native).

You have to call [Use_Defaults](#) manually to allow MEDAQLib timeout checking, to calculate datarate and to interpret and scale data.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from sensor.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Use_Defaults](#)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

The CSP2008 does not act as an interface in this case. You do not have direct access to the sensors connected to CSP2008.

17.7.1 Parameter management

17.7.1.1 Save_Parameters (STORESETUP)

Save actual parameters at controller. There can be saved several settings on different locations. So it is easy to switch to another setting.

Parameter: int32_t SP_ParameterSet

SP_ParameterSet

Direction: Down

Valid values:

Minimum: 1

Maximum: 20

Description: Location to save the settings.

17.7.1.2 Load_Parameters (LOADSETUP)

Load stored parameters into controller RAM. There can be loaded several settings from different locations. So it is easy to switch to another setting.

Parameter: int32_t SP_ParameterSet

SP_ParameterSet

Direction: Down

Valid values:

Minimum: 1

Maximum: 20

Description: Location from where the settings should be loaded.

17.7.2 Mastering

17.7.2.1 Set_MasterValue (SWMASTER)

Set the master value.

17.7.2.2 Reset_MasterValue (SWREMASTER)

Reset the master value.

17.7.3 Software trigger

17.7.3.1 Set_SoftwareTriggerEnable (SWTRIGGERENABLE)

Enable/Disable the software trigger mode.

Parameter: int32_t SP_SoftwareTriggerEnable

SP_SoftwareTriggerEnable

Direction: Down

Valid values:

0= Off

1= On

Description: Software trigger mode.

17.7.3.2 Get_SoftwareTriggerEnable (SWTRIGGERENABLE)

Get the software trigger mode.

Parameter: int32_t SA_SoftwareTriggerEnable

SA_SoftwareTriggerEnable

Direction: Up

Valid values:

0= Off

1= On

Description: Software trigger mode.

17.7.3.3 Set_SoftwareTrigger (SWTRIGGER)

Set a software trigger.

Parameter: int32_t SP_SoftwareTrigger

SP_SoftwareTrigger

Direction: Down

Valid values:

0= Off

1= On

Description: Software trigger.

17.7.3.4 Get_SoftwareTrigger (SWTRIGGER)

Get the software trigger.

Parameter: int32_t SA_SoftwareTrigger

SA_SoftwareTrigger

Direction: Up

Valid values:

0= Off

1= On

Description: Software trigger.

17.7.4 Special commands

17.7.4.1 Use_Defaults

This command tells the driver to use default values to operate with sensor data. If some parameters are not specified they are not changed. The sensor is not affected by this command.

Parameter: double IP_Samplerate

IP_Samplerate

Direction: Down

Valid values:

Minimum: 0

Maximum: 1.79769e+308 (DBL_MAX)

Unit: Hz

Description: Tells the driver the sampling rate of the controller.

Parameter: double IP_Datarate IP_Datarate

Direction: Down

Valid values:

Minimum: 0

Maximum: 1.79769e+308 (DBL_MAX)

Unit: Hz

Description: Tells the driver the datarate (output rate) of the controller. It is used for timeout checking. If it is zero, no timeout check is performed.

Parameter: double IP_RangeMin IP_RangeMin

Direction: Down

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Unit: Either mm or μm or any other

Description: Tells the driver the minimum range of sensor. It is used to scale the raw sensor values into mm or μm . If IP_RangeMin and IP_RangeMax is zero, no scaling is done.

Parameter: double IP_RangeMax IP_RangeMax

Direction: Down

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Unit: Either mm or μm or any other

Description: Tells the driver the maximum range of sensor. It is used to scale the raw sensor values into mm or μm . If IP_RangeMin and IP_RangeMax is zero, no scaling is done.

17.7.4.2 Get_DrvSetting

Returns the current settings of the driver used for operating with sensor data. It is the opposite of Use_Defaults. The sensor is not affected by this command.

Parameter: double IA_Samplerate IA_Samplerate

Direction: Up

Valid values:

Minimum: 0

Maximum: 1.79769e+308 (DBL_MAX)

Unit: Hz

Description: The sampling rate of the controller used by driver. 0 means the samplerate is unknown.

Parameter: double IA_Datarate IA_Datarate

Direction: Up

Valid values:

Minimum: 0

Maximum: 1.79769e+308 (DBL_MAX)

Unit: Hz

Description: The datarate (output rate) of the controller used by driver to check timeout from sensor. 0 means the datarate is unknown.

Parameter: double IA_RangeMin IA_RangeMin

Direction: Up

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Unit: Either mm or μm or any other

Description: Minimum range of sensor assumed by the driver

Parameter: double IA_RangeMax IA_RangeMax

Direction: Up

Valid values:

Minimum: -1.79769e+308 (-DBL_MAX)

Maximum: 1.79769e+308 (DBL_MAX)

Unit: Either mm or μm or any other

Description: Maximum range of sensor assumed by the driver

17.8 Commands for C-Box Analog, C-Box/2A and thicknessSENSOR

See C-Box/thicknessSENSOR manual for detailed description of sensor commands.

This sensor supports following interfaces:

- [RS232](#) (additional, e.g. IF2001_USB (RS422) and RS232 high level interface) (only at C-Box Analog).
- [IF2004](#) (native) (only at C-Box Analog).
- [TCP/IP](#) (native).
- [IF2004_USB](#) (native) (only at C-Box Analog).
- [IF2008](#) (native) (only at C-Box Analog).
- [WinUSB](#) (native) (only at C-Box Analog and C-Box/2A).

If first bit of [IP_AutomaticMode](#) is set (1), MEDAQLib calls automatically sensor command [Get_AllParameters](#) (SP_Additional= 1) after [OpenSensor](#).

Otherwise, you have to call it manually to allow MEDAQLib timeout checking, to calculate datarate, to interpret and scale data and to assign values. If second bit of [IP_AutomaticMode](#) is set (2), MEDAQLib calls optionally sensor command [Set_DataOutInterface](#), [Get_LaserPower<n>](#) and optionally [Set_LaserPower<n>](#) for attached sensors at [OpenSensor](#).

At open, MEDAQLib calls [Get_ScanStatus](#) periodically up to 20 seconds until C-Box has finished scanning.

Meaning of raw and scaled values (function [Poll](#) and [TransferData](#)):

- Raw values are as it comes directly from C-Box and the sensors behind.
- Scaled values are scaled using sensor range (if known by MEDAQLib, use sensor command [Get_AllParameters](#) (SP_Additional= 1)).

The values of one data frame are filled in the arrays one after another. Each array always starts with a new data frame.

17.8.1 General commands

17.8.1.1 General

17.8.1.1.1 Get_Info (GETINFO)

Retrieve information about the controller.

Parameter: String SA_Sensor

SA_Sensor

Direction: Up

Description: Name of the controller.

Parameter: String SA_SerialNumber

SA_SerialNumber

Direction: Up

Valid values:

Numeric value

Description: Serial number of the controller.

Parameter: String SA_Option

SA_Option

Direction: Up

Valid values:

Numeric value

Description: Option of the controller.

Parameter: String SA_ArticleNumber

SA_ArticleNumber

Direction: Up

Valid values:

Numeric value

Description: Article number of the controller.

Parameter: String SA_MacAddress

SA_MacAddress

Direction: Up

Valid values:

Valid MAC address in form of xx-xx-xx-xx-xx-xx

Description: MAC address (low level ethernet address) of the controller.

Parameter: String SA_Softwareversion

SA_Softwareversion

Direction: Up

Description: Software version of firmware in the controller.

Parameter: String SA_FPGAVersion

SA_FPGAVersion

Direction: Up

Description: FPGA version in the controller.

Parameter: String SA_WebpageVersion

SA_WebpageVersion

Direction: Up

Description: Version of the Web pages in the controller.

17.8.1.1.2 Set_Unit (UNIT)

Set the unit for configuration and display in the web diagram.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: int32_t SP_DisplayUnit

SP_DisplayUnit

Direction: Down

Valid values:

0= mm
1= Inch

Description: Unit.

17.8.1.1.3 Get_Unit (UNIT)

Get the unit for configuration and display in the web diagram.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: int32_t SA_DisplayUnit

SA_DisplayUnit

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= mm
1= Inch

Description: Unit.

17.8.1.1.4 Set_SyncMode (SYNC)

Set the synchronization mode.

Valid for sensor:

C-Box Analog
C-Box/2A

Parameter: int32_t SP_SyncMode

SP_SyncMode

Direction: Down

Valid values:

0= None
1= Internal
2= External

Description: Synchronization mode.

Parameter: int32_t SP_SyncLogic

SP_SyncLogic

Direction: Down

Valid values:

0= Low level logic (LLL)
1= High level logic (HLL)

Description: Logic level of external input.

17.8.1.1.5 Get_SyncMode (SYNC)

Get the synchronization mode.

Valid for sensor:

- C-Box Analog
- C-Box/2A

Parameter: int32_t SA_SyncMode

SA_SyncMode

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = None
- 1 = Internal
- 2 = External

Description: Synchronization mode.

Parameter: int32_t SA_SyncLogic

SA_SyncLogic

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Low level logic (LLL)
- 1 = High level logic (HLL)

Description: Logic level of external input.

17.8.1.1.6 Reset_Boot (RESET)

Resets the controller.

At this command the sensor may change output data after reboot. If first bit of [IP_AutomaticMode](#) is set (1), [Get_AllParameters](#) (SP_Additional= 1) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_AllDevices

SP_AllDevices

Direction: Down

Valid values:

- 0 = Only C-Box/thicknessSENSOR
- 1 = C-Box/thicknessSENSOR and plugged sensors (ALL)

Description: Reset only C-Box/thicknessSENSOR or plugged sensors additionally.

17.8.1.1.7 Get_ScanStatus (SCANSTATUS)

Ask the controller if scanning for sensors is active.

Parameter: int32_t SA_ScanStatus

SA_ScanStatus

Direction: Up

Valid values:

- 0 = No scan in progress (READY)
- 1 = Active

Description: Scan state

17.8.1.1.8 Get_AllParameters (PRINT)

Get all parameters from controller.

If SP_Additional is 1 there also may be contained the answer parameters of [Get_Info<n>](#).

Parameter: int32_t SP_Additional	SP_Additional
Direction: Down	
Valid values:	
0= No	
1= Yes	
Description: If set, additional information about controller and sensors are output.	
 Parameter: int32_t SA_ScanStatus	SA_ScanStatus
Direction: Up	
Valid values:	
0= No scan in progress (READY)	
1= Active	
Description: Scan state	
 Parameter: int32_t SA_SensorBaudrate	SA_SensorBaudrate
Direction: Up	
Valid values:	
8000000	
4000000	
3500000	
3000000	
2500000	
2000000	
1500000	
921600	
691200	
460800	
230400	
115200	
9600	
Unit: Baud	
Valid for sensor:	
C-Box Analog	
Description: Baudrate of controller.	
 Parameter: int32_t SA_DHCPEnabled	SA_DHCPEnabled
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= FALSE	
1= TRUE	
Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).	

Parameter: String SA_Address	SA_Address
Direction: Up	
Valid values:	
Valid IP address in form of xxx.xxx.xxx.xxx	
Description:	IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.
Parameter: String SA_SubnetMask	SA_SubnetMask
Direction: Up	
Valid values:	
Valid network mask (e.g. 255.255.255.0 for a Class C network)	
Description:	Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.
Parameter: String SA_Gateway	SA_Gateway
Direction: Up	
Valid values:	
Valid IP address of default gateway in form of xxx.xxx.xxx.xxx	
Description:	Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.
Parameter: int32_t SA_Protocol	SA_Protocol
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = TCP server (SERVER/TCP)	
Description:	Only one protocol is supported.
Parameter: int32_t SA_Port	SA_Port
Direction: Up	
Valid values:	
Minimum: 1024	
Maximum: 65535	
Description:	Port to send data to or to listen for incoming requests.
Parameter: int32_t SA_LaserPower1..2	SA_LaserPower1..2
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Off	
1 = On	
Description:	Laser power.
Parameter: int32_t SA_MeasureMode	SA_MeasureMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Value of sensor 1 (SENSOR1VALUE)	
1 = Thickness between sensor 1 and sensor 2 (SENSOR12THICK)	
2 = Step from sensor 1 to sensor 2 (SENSOR12STEP)	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description:	Measure mode.

Parameter: double SA_Measrate	SA_Measrate
Direction: Up	
Valid values:	
Set_SyncMode is None: 0.1/0.4 ... 80.0 (only at C-Box Analog and C-Box/2A)	
Set_SyncMode is Internal: Depending on connected sensor.	
Unit: kHz	
Description: Samplerate of measurement.	
Parameter: int32_t SA_DisplayUnit	SA_DisplayUnit
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = mm	
1 = Inch	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description: Unit.	
Parameter: int32_t SA_SyncMode	SA_SyncMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = None	
1 = Internal	
2 = External	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Synchronization mode.	
Parameter: int32_t SA_SyncLogic	SA_SyncLogic
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = Low level logic (LLL)	
1 = High level logic (HLL)	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Logic level of external input.	
Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = None	
1 = Edge	
2 = Level (PULSE)	
3 = Software	
Description: Trigger mode.	

Parameter: int32_t SA_TriggerLevel SA_TriggerLevel

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = High
- 1 = Low

Description: Trigger level.

Parameter: int32_t SA_TriggerLogic SA_TriggerLogic

Direction: Up

Valid values:

- 0 = Low level logic (LLL)
- 1 = High level logic (HLL)

Description: Logic level of external input.

Parameter: int32_t SA_TriggerCount SA_TriggerCount

Direction: Up

Valid values:

- Minimum:** 0
- Maximum:** 16383

Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

Parameter: int32_t SA_AveragingType SA_AveragingType

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = None
- 1 = Moving average (MOVING)
- 2 = Recursive averaging (RECURSIVE)
- 3 = Median

Description: Averaging type.

Parameter: int32_t SA_MovingCount SA_MovingCount

Direction: Up

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128
- 256
- 512

Description: Number of value for the averaging window.

This parameter is only available at moving average.

Parameter: int32_t SA_RecursiveCount SA_RecursiveCount

Direction: Up

Valid values:

- Minimum:** 2, 4, 8, ...,
- Maximum:** 32768

Description: Number of values for recursive averaging.

This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount	SA_MedianCount
Direction: Up	
Valid values:	
3	
5	
7	
9	
Description: Number of values to build median.	
This parameter is only available at median.	
Parameter: int32_t SA_AveragingType1..2	SA_AveragingType1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Moving average (MOVING)	
2= Recursive averaging (RECURSIVE)	
3= Median	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Averaging type.	
Parameter: int32_t SA_MovingCount1..2	SA_MovingCount1..2
Direction: Up	
Valid values:	
Depending on the connected sensor	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Number of value for the averaging window.	
This parameter is only available at moving average.	
Parameter: int32_t SA_RecursiveCount1..2	SA_RecursiveCount1..2
Direction: Up	
Valid values:	
Depending on the connected sensor	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Number of values for recursive averaging.	
This parameter is only available at recursive average.	
Parameter: int32_t SA_MedianCount1..2	SA_MedianCount1..2
Direction: Up	
Valid values:	
Depending on the connected sensor	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Number of values to build median.	
This parameter is only available at median.	

Parameter: int32_t SA_MultiFunctionInputMode	SA_MultiFunctionInputMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from controller	
0= MFI has no function (NONE)	
1= Use MFI to master the measuring value (MASTER)	
2= Pass through MFI to sensor1 (SENSOR1)	
3= Pass through MFI to sensor2 (SENSOR2)	
4= Pass through MFI to both sensor1 and sensor2 (SENSOR12)	
Valid for sensor:	
C-Box/2A	
Description:	Specifies how to use the multi function input.
Parameter: int32_t SA_MultiFunctionInputLogic	SA_MultiFunctionInputLogic
Direction: Up	
Valid values:	
0= Low level logic (LLL)	
1= High level logic (HLL)	
Valid for sensor:	
C-Box/2A	
Description:	Logic level of multi function input.
Parameter: int32_t SA_Master	SA_Master
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= no (NONE)	
1= yes (MASTER)	
Description:	Specifies if mastering is active.
Parameter: double SA_MasterValue	SA_MasterValue
Direction: Up	
Valid values:	
Minimum: -1024.0	
Maximum: +1024.0	
Unit: mm	
Description:	Master value
Parameter: int32_t SA_DataOutInterface	SA_DataOutInterface
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= RS422 (only at C-Box Analog)	
2= Ethernet	
3= HTTP	
5= USB (only at C-Box Analog and C-Box/2A)	
Description:	Active interface for data output.
Parameter: int32_t SA_Resampling	SA_Resampling
Direction: Up	
Valid values:	
Minimum: 1	
Maximum: 1000	
Description:	Resampling value.

Parameter: int32_t SA_ResampleAnalog	SA_ResampleAnalog
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Analog output is resampled.	
Parameter: int32_t SA_ResampleRS422	SA_ResampleRS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
Description: RS422 output is resampled.	
Parameter: int32_t SA_ResampleUSB	SA_ResampleUSB
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Output on USB is resampled.	
Parameter: int32_t SA_ResampleEthernet	SA_ResampleEthernet
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Output over ethernet is resampled.	
Parameter: int32_t SA_OutputScaleMode	SA_OutputScaleMode
Direction: Up	
Valid values:	
0= Standard	
1= Two point (TWOPOINT)	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Output scale mode.	
Parameter: double SA_OutputMinValue	SA_OutputMinValue
Direction: Up	
Valid values:	
Minimum: -1024.0	
Maximum: 1024.0	
Unit: mm	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Lowest possible value to transmit over RS422 (at C-Box Analog) or USB (at C-Box Analog and C-Box/2A).	

Parameter: double SA_Output.MaxValue	SA_Output.MaxValue
Direction: Up	
Valid values:	
Minimum: -1024.0	
Maximum: 1024.0	
Unit: mm	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Highest possible value to transmit over RS422 (at C-Box Analog) or USB (at C-Box Analog and C-Box/2A).	
Parameter: int32_t SA_HoldLastValid	SA_HoldLastValid
Direction: Up	
Valid values:	
Minimum: -1	
Maximum: 1024	
Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).	
Parameter: int32_t SA_FramesPerPacket_ETH	SA_FramesPerPacket_ETH
Direction: Up	
Valid values:	
Minimum: 0	
Maximum: 200	
Description: Desired number of frames in ethernet packet.	
0 means automatic (each 10 ms a packet). If packets cannot be send as fast as needed, the number of frames may be exceeded. If maximum packet size is not sufficient, the number of frames is reduced to fit into a packet.	
Parameter: int32_t SA_OutputAdditional	SA_OutputAdditional
Direction: Up	
Valid values:	
0= C-Box counter (C-BOXCOUNTER)	
1= C-Box timestamp (C-BOXTIMESTAMP)	
2= C-Box trigger input (TRG-IN)	
Valid for sensor:	
C-Box Analog	
Description: Specify which additional value is transmitted.	
Parameter: int32_t SA_ErrorOutput1..2	SA_ErrorOutput1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Pass through first error output of sensor1 (SENSOR1ERROROUT1)	
1= Pass through second error output of sensor1 (SENSOR1ERROROUT2)	
2= Pass through first error output of sensor2 (SENSOR2ERROROUT1)	
3= Pass through second error output of sensor2 (SENSOR2ERROROUT2)	
4= Range check measure value of sensor 1 (SENSOR1VALUE)	
5= Range check intensity value of sensor 1 (SENSOR1INTENSITY)	
6= Range check shutter time of sensor 1 (SENSOR1SHUTTER)	
7= Range check reflectivity of sensor 1 (SENSOR1REFLECTIVITY)	
8= Range check measure value of sensor 2 (SENSOR2VALUE)	

9= Range check intensity value of sensor 2 (SENSOR2INTENSITY)
 10= Range check shutter time of sensor 2 (SENSOR2SHUTTER)
 11= Range check reflectivity of sensor 2 (SENSOR2REFLECTIVITY)
 12= Range check calculated value of C-Box (C-BOXVALUE)
 13= Always low (LOW)
 14= Always high (HIGH)

Valid for sensor:

C-Box/2A
thicknessSENSOR

Description: Condition for error output.

Parameter: double SA_LowerLimit1..2

SA_LowerLimit1..2

Direction: Up

Valid values:

Minimum: -1024.0 (for distance/thickness), 0 (for intensity/shutter)
Maximum: 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0
 (for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_ErrorOutput<n>](#)

Valid for sensor:

C-Box/2A
thicknessSENSOR

Description: Lower limit.

Parameter: double SA_UpperLimit1..2

SA_UpperLimit1..2

Direction: Up

Valid values:

Minimum: -1024.0 (for distance/thickness), 0 (for intensity/shutter)
Maximum: 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0
 (for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_ErrorOutput<n>](#)

Valid for sensor:

C-Box/2A
thicknessSENSOR

Description: Upper limit.

Parameter: int32_t SA_OutputSensor1Value_RS422

SA_OutputSensor1Value_-
RS422

Direction: Up

Valid values:

0= no
1= yes

Valid for sensor:

C-Box Analog

Description: Specify if sensor 1 measure value is transmitted.

Parameter: int32_t SA_OutputSensor1Additional_RS422

SA_OutputSen-
sor1Additional_RS422

Direction: Up

Valid values:

0= no
1= yes

Valid for sensor:

C-Box Analog

Description: Specify if sensor 1 additional value is transmitted.

Parameter: int32_t SA_OutputSensor2Value_RS422	SA_OutputSensor2Value_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
Description:	Specify if sensor 2 measure value is transmitted.
Parameter: int32_t SA_OutputSensor2Additional_RS422	SA_OutputSensor2Additional_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
Description:	Specify if sensor 2 additional value is transmitted.
Parameter: int32_t SA_OutputC-BoxValue_RS422	SA_OutputC-BoxValue_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
Description:	Specify if C-Box measure value is transmitted.
Parameter: int32_t SA_OutputC-BoxAdditional_RS422	SA_OutputC-BoxAdditional_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
Description:	Specify if C-Box additional value is transmitted.
Parameter: int32_t SA_OutputSensor1Value_USB	SA_OutputSensor1Value_USB
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description:	Specify if sensor 1 measure value is transmitted.
Parameter: int32_t SA_OutputSensor1Additional_USB	SA_OutputSensor1Additional_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description:	Specify if sensor 1 additional value is transmitted.

Parameter: int32_t SA_OutputSensor1Intensity_USB	SA_OutputSensor1Intensity_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 1 intensity value is transmitted.	
Parameter: int32_t SA_OutputSensor1Shutter_USB	SA_OutputSensor1Shutter_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 1 shutter value is transmitted.	
Parameter: int32_t SA_OutputSensor1Reflectivity_USB	SA_OutputSensor1Reflectivity_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 1 reflectivity value (calculated by C-Box) is transmitted.	
Parameter: int32_t SA_OutputSensor2Value_USB	SA_OutputSensor2Value_USB
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Specify if sensor 2 measure value is transmitted.	
Parameter: int32_t SA_OutputSensor2Additional_USB	SA_OutputSensor2Additional_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description: Specify if sensor 2 additional value is transmitted.	
Parameter: int32_t SA_OutputSensor2Intensity_USB	SA_OutputSensor2Intensity_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 2 intensity value is transmitted.	
Parameter: int32_t SA_OutputSensor2Shutter_USB	SA_OutputSensor2Shutter_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 2 shutter value is transmitted.	

Parameter: int32_t SA_OutputSensor2Reflectivity_USB	SA_OutputSensor2Reflectivity_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 2 reflectivity value (calculated by C-Box) is transmitted.	
Parameter: int32_t SA_OutputC-BoxValue_USB	SA_OutputC-BoxValue_USB
Direction: Up	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
C-Box/2A	
Description: Specify if C-Box measure value is transmitted.	
Parameter: int32_t SA_OutputC-BoxAdditional_USB	SA_OutputC-BoxAdditional_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description: Specify if C-Box additional value is transmitted.	
Parameter: int32_t SA_OutputC-BoxCounter_USB	SA_OutputC-BoxCounter_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if C-Box counter value is transmitted.	
Parameter: int32_t SA_OutputC-BoxTimestamp_USB	SA_OutputC-BoxTimestamp_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if C-Box timestamp value is transmitted.	
Parameter: int32_t SA_OutputC-BoxDigital_USB	SA_OutputC-BoxDigital_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if C-Box digital signals are transmitted.	
Parameter: int32_t SA_OutputSensor1Value_ETH	SA_OutputSensor1Value_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 1 measure value is transmitted.	

Parameter: int32_t SA_OutputSensor1Additional_ETH	SA_OutputSensor1Additional_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description: Specify if sensor 1 additional value is transmitted.	
Parameter: int32_t SA_OutputSensor1Intensity_ETH	SA_OutputSensor1Intensity_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if sensor 1 intensity value is transmitted.	
Parameter: int32_t SA_OutputSensor1Shutter_ETH	SA_OutputSensor1Shutter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if sensor 1 shutter value is transmitted.	
Parameter: int32_t SA_OutputSensor1Reflectivity_ETH	SA_OutputSensor1Reflectivity_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if sensor 1 reflectivity value (calculated by C-Box) is transmitted.	
Parameter: int32_t SA_OutputSensor2Value_ETH	SA_OutputSensor2Value_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 2 measure value is transmitted.	
Parameter: int32_t SA_OutputSensor2Additional_ETH	SA_OutputSensor2Additional_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description: Specify if sensor 2 additional value is transmitted.	
Parameter: int32_t SA_OutputSensor2Intensity_ETH	SA_OutputSensor2Intensity_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if sensor 2 intensity value is transmitted.	
Parameter: int32_t SA_OutputSensor2Shutter_ETH	SA_OutputSensor2Shutter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if sensor 2 shutter value is transmitted.	

Parameter: int32_t SA_OutputSensor2Reflectivity_ETH	SA_OutputSensor2Reflectivity_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if sensor 2 reflectivity value (calculated by C-Box) is transmitted.	
Parameter: int32_t SA_OutputC-BoxValue_ETH	SA_OutputC-BoxValue_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if C-Box measure value is transmitted.	
Parameter: int32_t SA_OutputC-BoxAdditional_ETH	SA_OutputC-BoxAdditional_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description: Specify if C-Box additional value is transmitted.	
Parameter: int32_t SA_OutputC-BoxCounter_ETH	SA_OutputC-BoxCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if C-Box counter value is transmitted.	
Parameter: int32_t SA_OutputC-BoxTimestamp_ETH	SA_OutputC-BoxTimestamp_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if C-Box timestamp value is transmitted.	
Parameter: int32_t SA_OutputC-BoxDigital_ETH	SA_OutputC-BoxDigital_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if C-Box digital signals are transmitted.	
Parameter: int32_t SA_AnalogOutput	SA_AnalogOutput
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Fixed	
1= Sensor 1 value (SENSOR1VALUE)	
2= Sensor 2 value (SENSOR2VALUE)	
3= C-Box value (C-BOXVALUE)	
Valid for sensor:	
C-Box Analog	
Description: Data to be used for analog output.	

Parameter: double SA_AnalogValue SA_AnalogValue

Direction: Up

Valid values:

Minimum: 0, -5, -10 or 4, depending on analog range

Maximum: 5, 10 or 20, depending on analog range

Unit: V or mA, depending on analog range

Valid for sensor:

C-Box Analog

Description: Fixed output value for analog output.

Parameter: int32_t SA_AnalogOutput1..2 SA_AnalogOutput1..2

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Fixed

1= Sensor 1 value (SENSOR1VALUE)

2= Sensor 2 value (SENSOR2VALUE)

3= C-Box value (C-BOXVALUE)

4= Sensor 1 intensity value (SENSOR1INTENSITY)

5= Sensor 1 shutter time (SENSOR1SHUTTER)

6= Sensor 1 reflectivity (SENSOR1REFLECTIVITY)

7= Sensor 2 intensity value (SENSOR2INTENSITY)

8= Sensor 2 shutter time (SENSOR2SHUTTER)

9= Sensor 2 reflectivity (SENSOR2REFLECTIVITY)

Valid for sensor:

C-Box/2A

thicknessSENSOR

Description: Data to be used for analog output.

Parameter: double SA_AnalogValue1..2 SA_AnalogValue1..2

Direction: Up

Valid values:

Minimum: 0, -5, -10 or 4, depending on analog range

Maximum: 5, 10 or 20, depending on analog range

Unit: V or mA, depending on analog range

Valid for sensor:

C-Box/2A

thicknessSENSOR

Description: Fixed output value for analog output.

Parameter: int32_t SA_AnalogRange SA_AnalogRange

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= None

1= 0 - 5V

2= 0 - 10V

3= -5 - 5V

4= -10 - 10V

5= 4 - 20mA

Valid for sensor:

C-Box Analog

Description: Analog output range.

Parameter: int32_t SA_AnalogRange1..2	SA_AnalogRange1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= 0 - 5V	
2= 0 - 10V	
3= -5 - 5V	
4= -10 - 10V	
5= 4 - 20mA	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description: Analog output range.	
Parameter: int32_t SA_AnalogScaleMode	SA_AnalogScaleMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Standard	
1= Two point (TWOPOINT)	
Valid for sensor:	
C-Box Analog	
Description: Analog scale mode.	
Parameter: double SA_MinValue	SA_MinValue
Direction: Up	
Valid values:	
Minimum: -1024.0	
Maximum: 1024.0	
Unit: mm	
Valid for sensor:	
C-Box Analog	
Description: Value which represents lowest voltage/current (at two point scaling).	
Parameter: double SA_MaxValue	SA_MaxValue
Direction: Up	
Valid values:	
Minimum: -1024.0	
Maximum: 1024.0	
Unit: mm	
Valid for sensor:	
C-Box Analog	
Description: Value which represents highest voltage/current (at two point scaling).	
Parameter: int32_t SA_AnalogScaleMode1..2	SA_AnalogScaleMode1..2
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= Standard	
1= Two point (TWOPOINT)	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description: Analog scale mode.	

Parameter: double SA_MinValue1..2 SA_MinValue1..2

Direction: Up

Valid values:

Minimum: -1024.0 (for distance/thickness), 0 (for intensity/shutter)

Maximum: 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0 (for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_AnalogOutput<n>](#)

Valid for sensor:

C-Box/2A

thicknessSENSOR

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SA_MaxValue1..2 SA_MaxValue1..2

Direction: Up

Valid values:

Minimum: -1024.0 (for distance/thickness), 0 (for intensity/shutter)

Maximum: 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0 (for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_AnalogOutput<n>](#)

Valid for sensor:

C-Box/2A

thicknessSENSOR

Description: Value which represents highest voltage/current (at two point scaling).

Parameter: String SA_ApplicationLanguage SA_ApplicationLanguage

Direction: Up

Valid values:

BROWSER

ENGLISH

GERMAN

Or any other language (from Firmware version 3.34??).

Description: Language of web interface.

Parameter: String SA_Sensor SA_Sensor

Direction: Up

Description: Name of the controller.

Parameter: String SA_SerialNumber SA_SerialNumber

Direction: Up

Valid values:

Numeric value

Description: Serial number of the controller.

Parameter: String SA_Option SA_Option

Direction: Up

Valid values:

Numeric value

Description: Option of the controller.

Parameter: String SA_ArticleNumber	SA_ArticleNumber
Direction: Up	
Valid values:	
Numeric value	
Description: Article number of the controller.	
Parameter: String SA_MacAddress	SA_MacAddress
Direction: Up	
Valid values:	
Valid MAC address in form of xx-xx-xx-xx-xx-xx	
Description: MAC address (low level ethernet address) of the controller.	
Parameter: String SA_Softwareversion	SA_Softwareversion
Direction: Up	
Description: Software version of firmware in the controller.	
Parameter: String SA_FPGAVersion	SA_FPGAVersion
Direction: Up	
Description: FPGA version in the controller.	
Parameter: String SA_WebpageVersion	SA_WebpageVersion
Direction: Up	
Description: Version of the Web pages in the controller.	

17.8.1.2 Triggering

17.8.1.2.1 Set_TriggerMode (TRIGGER)

Set the trigger mode.

Parameter: int32_t SP_TriggerMode	SP_TriggerMode
Direction: Down	
Valid values:	
0= None	
1= Edge	
2= Level (PULSE)	
3= Software	
Description: Trigger mode.	

17.8.1.2.2 Get_TriggerMode (TRIGGER)

Get the active trigger mode.

Parameter: int32_t SA_TriggerMode	SA_TriggerMode
Direction: Up	
Valid values:	
-1= Unknown parameter value from sensor	
0= None	
1= Edge	
2= Level (PULSE)	
3= Software	
Description: Trigger mode.	

17.8.1.2.3 Set_TriggerLevel (TRIGGERLEVEL)

Set the trigger level.

Parameter: int32_t SP_TriggerLevel

SP_TriggerLevel

Direction: Down

Valid values:

0= High

1= Low

Description: Trigger level.

Parameter: int32_t SP_TriggerLogic

SP_TriggerLogic

Direction: Down

Valid values:

0= Low level logic (LLL)

1= High level logic (HLL)

Description: Logic level of external input.

17.8.1.2.4 Get_TriggerLevel (TRIGGERLEVEL)

Get the active trigger level.

Parameter: int32_t SA_TriggerLevel

SA_TriggerLevel

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= High

1= Low

Description: Trigger level.

Parameter: int32_t SA_TriggerLogic

SA_TriggerLogic

Direction: Up

Valid values:

0= Low level logic (LLL)

1= High level logic (HLL)

Description: Logic level of external input.

17.8.1.2.5 Set_TriggerCount (TRIGGERCOUNT)

Set the number of values to measure at trigger.

Parameter: int32_t SP_TriggerCount

SP_TriggerCount

Direction: Down

Valid values:

Minimum: 0

Maximum: 16383

Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

17.8.1.2.6 Get_TriggerCount (TRIGGERCOUNT)

Get the number of values to measure at trigger.

Parameter: int32_t SA_TriggerCount SA_TriggerCount
Direction: Up
Valid values:
 Minimum: 0
 Maximum: 16383
Description: Number of values to measure. 0 means no trigger, 16383 means endless measurement.

17.8.1.2.7 Software_Trigger (TRIGGERSW)

Execute a software trigger.

17.8.1.3 Interfaces

17.8.1.3.1 Set_IPConfiguration (IPCONFIG)

Set the IP configuration at controller.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_DHCPEnabled SP_DHCPEnabled
Direction: Down

Valid values:

0= FALSE
 1= TRUE

Description: Specify if controller should use a static IP address or ask for IP at DHCP server (dynamic IP address).

Parameter: String SP_Address SP_Address

Direction: Down

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_SubnetMask SP_SubnetMask

Direction: Down

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. This parameter is only evaluated on static IP assignment.

Parameter: String SP_Gateway SP_Gateway

Direction: Down

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: The default gateway must be specified if the controller should communicate with peers in foreign subnets. This parameter is only evaluated on static IP assignment.

17.8.1.3.2 Get_IPConfiguration (IPCONFIG)

Get the IP configuration at controller.

Parameter: int32_t SA_DHCPEnabled SA_DHCPEnabled

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = FALSE
- 1 = TRUE

Description: Get settings if controller should use a static IP address ask for IP at DHCP server (dynamic IP address).

Parameter: String SA_Address SA_Address

Direction: Up

Valid values:

Valid IP address in form of xxx.xxx.xxx.xxx

Description: IP address of the controller. If DHCP is enabled it returns the currently assigned IP address.

Parameter: String SA_SubnetMask SA_SubnetMask

Direction: Up

Valid values:

Valid network mask (e.g. 255.255.255.0 for a Class C network)

Description: Network mask of the controller. If DHCP is enabled it returns the currently assigned network mask.

Parameter: String SA_Gateway SA_Gateway

Direction: Up

Valid values:

Valid IP address of default gateway in form of xxx.xxx.xxx.xxx

Description: Address of the default gateway. If DHCP is enabled it returns the currently assigned default gateway.

17.8.1.3.3 Set_IPDataTransferMode (MEATRANSFER)

Set IP protocol at controller.

For this command an [Update ...](#) meta command is available.

Parameter: int32_t SP_Protocol SP_Protocol

Direction: Down

Valid values:

0 = TCP server (SERVER/TCP)

Description: Only one protocol is supported.

Parameter: int32_t SP_Port SP_Port

Direction: Down

Valid values:

Minimum: 1024

Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

17.8.1.3.4 Get_IPDataTransferMode (MEASTRANSFER)

Get IP protocol at controller.

Parameter: int32_t SA_Protocol

SA_Protocol

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = TCP server (SERVER/TCP)

Description: Only one protocol is supported.

Parameter: int32_t SA_Port

SA_Port

Direction: Up

Valid values:

Minimum: 1024

Maximum: 65535

Description: Port to send data to or to listen for incoming requests.

17.8.1.3.5 Set_Baudrate (BAUDRATE)

Set baudrate of controller for serial RS422 communication. After receiving the answer of the command, the baudrate of the local serial interface is changed, too.

Valid for sensor:

C-Box Analog

Parameter: int32_t SP_SensorBaudrate

SP_SensorBaudrate

Direction: Down

Valid values:

- 8000000
- 4000000
- 3500000
- 3000000
- 2500000
- 2000000
- 1500000
- 921600
- 691200
- 460800
- 230400
- 115200
- 9600

Unit: Baud

Description: Baudrate of controller.

Parameter: int32_t CP_InterruptDataTransfer

CP_InterruptDataTransfer

Direction: Down

Valid values:

- 0 = false
- 1 = true

Default: 0

Description: After switch baudrate, synchronization may be impossible if serial line has high load. Enabling this parameter resolves the problem by disabling data transfer first ([Set_DataOutInterface](#)) and enabling it again at end (only if enabled before).

17.8.1.3.6 Get_Baudrate (BAUDRATE)

Get baudrate of controller for serial RS422 communication.

Valid for sensor:

C-Box Analog

Parameter: int32_t SA_SensorBaudrate

SA_SensorBaudrate

Direction: Up

Valid values:

- 8000000
- 4000000
- 3500000
- 3000000
- 2500000
- 2000000
- 1500000
- 921600
- 691200
- 460800
- 230400
- 115200
- 9600

Unit: Baud

Description: Baudrate of controller.

17.8.1.3.7 Set_ApplLanguage (LANGUAGE)

Set language of web interface.

Parameter: String SP_ApplicationLanguage

SP_ApplicationLanguage

Direction: Down

Valid values:

- BROWSER
- ENGLISH
- GERMAN

Or any other language (from Firmware version 3.34??).

Description: Language of web interface.

17.8.1.3.8 Get_ApplLanguage (LANGUAGE)

Get language of web interface.

Parameter: String SA_ApplicationLanguage

SA_ApplicationLanguage

Direction: Up

Valid values:

- BROWSER
- ENGLISH
- GERMAN

Or any other language (from Firmware version 3.34??).

Description: Language of web interface.

17.8.1.4 Sensors

17.8.1.4.1 Scan_Sensor<n> (SCAN<n>)

Scan for a sensor at connector Sensor n ∈ {1..2}. The answer parameters are the same as at command [Get_Info<n>](#).

17.8.1.4.2 Get_Info<n> (GETINFO<n>)

Retrieve information about the sensor behind C-Box at connector n ∈ {1..2}. The answer parameters are depending on the connected sensor.

See the sensor command Get_Info at the specific sensor section. Use this parameter names with postfix <n>, e.g. SA_SerialNumber<n> for serial number of [SENSOR_ILD2300](#).

17.8.1.4.3 Set_LaserPower<n> (LASERPOW<n>)

Set the laser state (if supported by connected sensor) at connector n ∈ {1..2}. A special line between C-Box and sensor is used to switch the laser at sensor.

Parameter: int32_t SP_LaserPower<n>

SP_LaserPower<n>

Direction: Down

Valid values:

0= Off

1= On

Description: Laser power.

17.8.1.4.4 Get_LaserPower<n> (LASERPOW<n>)

Get the laser state at connector n ∈ {1..2}.

A special line between C-Box and sensor is used to switch the laser at sensor.

Parameter: int32_t SA_LaserPower<n>

SA_LaserPower<n>

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Off

1= On

Description: Laser power.

17.8.1.4.5 Set_Averaging<n> (AVERAGE<n>)

Set data averaging of sensor n ∈ {1..2} behind C-Box.

Valid for sensor:

C-Box Analog

C-Box/2A

Parameter: int32_t SP_AveragingType<n>

SP_AveragingType<n>

Direction: Down

Valid values:

0= None

1= Moving average (MOVING)

2= Recursive averaging (RECURSIVE)

3= Median

Description: Averaging type.

Parameter: int32_t SP_MovingCount<n>	SP_MovingCount<n>
Direction: Down	
Valid values:	
Depending on the connected sensor	
Description:	Number of value for the averaging window. This parameter is only used at moving average.
Parameter: int32_t SP_RecursiveCount<n>	SP_RecursiveCount<n>
Direction: Down	
Valid values:	
Depending on the connected sensor	
Description:	Number of values for recursive averaging. This parameter is only used at recursive average.
Parameter: int32_t SP_MedianCount<n>	SP_MedianCount<n>
Direction: Down	
Valid values:	
Depending on the connected sensor	
Description:	Number of values to build median. This parameter is only used at median.

17.8.1.4.6 Get_Averaging<n> (AVERAGE<n>)

Get data averaging of sensor $n \in \{1..2\}$ behind C-Box.

Valid for sensor:

C-Box Analog
C-Box/2A

Parameter: int32_t SA_AveragingType<n>	SA_AveragingType<n>
Direction: Up	
Valid values:	
-1 = Unknown parameter value from sensor	
0 = None	
1 = Moving average (MOVING)	
2 = Recursive averaging (RECURSIVE)	
3 = Median	
Description:	Averaging type.
Parameter: int32_t SA_MovingCount<n>	SA_MovingCount<n>
Direction: Up	
Valid values:	
Depending on the connected sensor	
Description:	Number of value for the averaging window. This parameter is only available at moving average.
Parameter: int32_t SA_RecursiveCount<n>	SA_RecursiveCount<n>
Direction: Up	
Valid values:	
Depending on the connected sensor	
Description:	Number of values for recursive averaging. This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount<n>

SA_MedianCount<n>

Direction: Up

Valid values:

Depending on the connected sensor

Description: Number of values to build median. This parameter is only available at median.

17.8.1.4.7 SensorCommand<n> (TUNNEL<n>)

Send a command directly to the attached sensor at connector n ∈ {1..2}.

Valid for sensor:

C-Box Analog

C-Box/2A

Parameter: String SP_Command<n>

SP_Command<n>

Direction: Down

Description: Generic sensor command.

Parameter: String SA_Answer<n>

SA_Answer<n>

Direction: Up

Description: Answer from sensor.

17.8.1.4.8 Get_SensorInstance

This internal command returns the MEDAQLib instance handle to the sensors connected to C-Box.

The handle returned can be used for any calling sensor command. [OpenSensor](#) or [CloseSensor](#) using this handle is not allowed. After calling [CloseSensor](#) or [ReleaseSensorInstance](#) using C-Box handle, this handle is no longer valid.

Valid for sensor:

C-Box Analog

C-Box/2A

Parameter: int32_t SP_Sensor

SP_Sensor

Direction: Down

Valid values:

1 = First sensor

2 = Second sensor

Description: Number of the sensor.

Parameter: int32_t SA_InstanceHandle

SA_InstanceHandle

Direction: Up

Valid values:

Minimum: 1

Maximum: 2147483647 (INT32_MAX)

Description: Instance number which now can be used to access sensor directly.

17.8.1.4.9 Set_MeasurePeak<n> (MEASPEAK)

Select the peak to measure for sensor $n \in \{1..2\}$.

Valid for sensor:

thicknessSENSOR

Parameter: int32_t SP_MeasurePeak<n>

SP_MeasurePeak<n>

Direction: Down

Valid values:

- 0= Greatest Amplitude (DISTA)
- 1= First Peak (DIST1)
- 2= Last Peak (DISTL)

Description: Peak to measure.

17.8.1.4.10 Get_MeasurePeak<n> (MEASPEAK)

Get the selected peak to measure of sensor $n \in \{1..2\}$.

Valid for sensor:

thicknessSENSOR

Parameter: int32_t SA_MeasurePeak<n>

SA_MeasurePeak<n>

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Greatest Amplitude (DISTA)
- 1= First Peak (DIST1)
- 2= Last Peak (DISTL)

Description: Peak to measure.

17.8.1.4.11 Set_TargetMode<n> (TARGETMODE)

Selecting a target loads a predefined sensor configuration which achieves the best results for the selected material for sensor $n \in \{1..2\}$.

Valid for sensor:

thicknessSENSOR

Parameter: int32_t SP_TargetMode<n>

SP_TargetMode<n>

Direction: Down

Valid values:

- 0= Standard Target (STANDARD)
- 1= Multi-Surface Target (MULTISURFACE)
- 2= Light Penetration Target (PENETRATION)

Description: Target mode.

17.8.1.4.12 Get_TargetMode<n> (TARGETMODE)

Target for a predefined sensor configuration which achieves the best results for the selected material of sensor n $\in \{1..2\}$.

Valid for sensor:

thicknessSENSOR

Parameter: int32_t SA_TargetMode<n>

SA_TargetMode<n>

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = Standard Target (STANDARD)
- 1 = Multi-Surface Target (MULTISURFACE)
- 2 = Light Penetration Target (PENETRATION)

Description: Target mode.

17.8.1.4.13 Get_Temperature<n> (GETTEMP<n>)

Get the temperature of sensor ILD2300 at connector n $\in \{1..2\}$.

A special line between C-Box and sensor is used to switch the laser at sensor.

Valid for sensor:

C-Box/2A from firmware version 36 with ILD2300 sensor

Parameter: double SA_Temperature<n>

SA_Temperature<n>

Direction: Up

Valid values:

Minimum: -275.50

Unit: °C

Description: Sensor temperature. If sensor does not transmit temperature, -275.50 is returned.

17.8.1.5 Parameter management

17.8.1.5.1 Save_Parameters (STORE)

Save actual parameters at controller. There can be saved several settings on different locations. So it is easy to switch to another setting.

Parameter: int32_t SP_ParameterSet

SP_ParameterSet

Direction: Down

Valid values:

Minimum: 1

Maximum: 8

Description: Location to save the settings.

17.8.1.5.2 Load_Parameters (READ)

Load stored parameters into controller RAM.

There can be loaded several settings from different locations. So it is easy to switch to another setting.

At this command the controller may change output data after applying new setting. If first bit of `IP_AutomaticMode` is set (1), [Get_AllParameters](#) (`SP_Additional= 1`) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t `SP_ParameterType`

`SP_ParameterType`

Direction: Down

Valid values:

- 0= All settings (ALL)
- 1= Device settings (DEVICE)
- 2= Measurement settings (MEAS)

Description: Specifies which settings should be loaded.

Parameter: int32_t `SP_ParameterSet`

`SP_ParameterSet`

Direction: Down

Valid values:

- Minimum: 1
- Maximum: 8

Description: Location from where the settings should be loaded.

17.8.1.5.3 Set_Default (SETDEFAULT)

Reset the controller to default settings.

At this command the sensor may change output data after applying default settings.

If first bit of `IP_AutomaticMode` is set (1), [Get_AllParameters](#) (`SP_Additional= 1`) is called automatically after this command. Otherwise, you have to call it manually.

After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t `SP_JustActualSetup`

`SP_JustActualSetup`

Direction: Down

Valid values:

- 0= Reset all settings (ALL)
- 1= Just reset actual setting

Description: Specifies which settings should be resetted.

Parameter: int32_t `SP_KeepDevice`

`SP_KeepDevice`

Direction: Down

Valid values:

- 0= Reset actual interface parameters
- 1= Keep device settings temporary (NODEVICE)

Description: Specifies if parameters should be hold temporary.

17.8.1.6 Internal controller commands

17.8.1.6.1 Set_MultiFunctionInputMode (MFIFUNC)

Set the mode of multi function input at controller.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: int32_t SP_MultiFunctionInputMode

SP_MultiFunctionInputMode

Direction: Down

Valid values:

- 0= MFI has no function (NONE)
- 1= Use MFI to master the measuring value (MASTER)
- 2= Pass through MFI to sensor1 (SENSOR1)
- 3= Pass through MFI to sensor2 (SENSOR2)
- 4= Pass through MFI to both sensor1 and sensor2 (SENSOR12)

Valid for sensor:

C-Box/2A

Description: Specifies how to use the multi function input.

Parameter: int32_t SP_MultiFunctionInputLogic

SP_MultiFunctionInput-
Logic

Direction: Down

Valid values:

- 0= Low level logic (LLL)
- 1= High level logic (HLL)

Description: Logic level of multi function input.

17.8.1.6.2 Get_MultiFunctionInputMode (MFIFUNC)

Get the mode of multi function input at controller.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: int32_t SA_MultiFunctionInputMode

SA_MultiFunctionInputMode

Direction: Up

Valid values:

- 1= Unknown parameter value from controller
- 0= MFI has no function (NONE)
- 1= Use MFI to master the measuring value (MASTER)
- 2= Pass through MFI to sensor1 (SENSOR1)
- 3= Pass through MFI to sensor2 (SENSOR2)
- 4= Pass through MFI to both sensor1 and sensor2 (SENSOR12)

Valid for sensor:

C-Box/2A

Description: Specifies how to use the multi function input.

Parameter: int32_t SA_MultiFunctionInputLogic SA_MultiFunctionInputLogic
Direction: Up
Valid values:
 0= Low level logic (LLL)
 1= High level logic (HLL)
Description: Logic level of multi function input.

17.8.2 Measurement

17.8.2.1 General

17.8.2.1.1 Set_MeasureMode (MEASMODE)

Set the measure mode.

Valid for sensor:

- C-Box Analog
- C-Box/2A

Parameter: int32_t SP_MeasureMode SP_MeasureMode
Direction: Down
Valid values:
 0= Value of sensor 1 (SENSOR1VALUE)
 1= Thickness between sensor 1 and sensor 2 (SENSOR12THICK)
 2= Step from sensor 1 to sensor 2 (SENSOR12STEP)
Description: Measure mode.

17.8.2.1.2 Get_MeasureMode (MEASMODE)

Get the measure mode.

Valid for sensor:

- C-Box Analog
- C-Box/2A

Parameter: int32_t SA_MeasureMode SA_MeasureMode
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Value of sensor 1 (SENSOR1VALUE)
 1= Thickness between sensor 1 and sensor 2 (SENSOR12THICK)
 2= Step from sensor 1 to sensor 2 (SENSOR12STEP)
Description: Measure mode.

17.8.2.1.3 Set_Samplerate (MEASRATE)

Set the samplerate.

Parameter: double SP_Measrate SP_Measrate
Direction: Down
Valid values:
Set_SyncMode is None: 0.1/0.4 ... 80.0 (only at C-Box Analog and C-Box/2A)
Set_SyncMode is Internal: Depending on connected sensor.
Unit: kHz
Description: Samplerate of measurement.

17.8.2.1.4 Get_Samplerate (MEASRATE)

Get the samplerate.

Parameter: double SA_Measrate SA_Measrate
Direction: Up
Valid values:
 Set_SyncMode is None: 0.1/0.4 ... 80.0 (only at C-Box Analog and C-Box/2A)
 Set_SyncMode is Internal: Depending on connected sensor.
Unit: kHz
Description: Samplerate of measurement.

17.8.2.2 Measurement value processing

17.8.2.2.1 Set_Averaging (AVERAGE)

Set data averaging at controller.

Parameter: int32_t SP_AveragingType SP_AveragingType
Direction: Down
Valid values:
 0= None
 1= Moving average (MOVING)
 2= Recursive averaging (RECURSIVE)
 3= Median
Description: Averaging type.

Parameter: int32_t SP_MovingCount SP_MovingCount
Direction: Down
Valid values:
 2
 4
 8
 16
 32
 64
 128
 256
 512
Description: Number of value for the averaging window. This parameter is only used at moving average.

Parameter: int32_t SP_RecursiveCount SP_RecursiveCount
Direction: Down
Valid values:
Minimum: 2, 4, 8, ...,
Maximum: 32768
Description: Number of values for recursive averaging. This parameter is only used at recursive average.

Parameter: int32_t SP_MedianCount SP_MedianCount

Direction: Down

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only used at median.

17.8.2.2.2 Get_Averaging (AVERAGE)

Get data averaging at controller.

Parameter: int32_t SA_AveragingType SA_AveragingType

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= Moving average (MOVING)
- 2= Recursive averaging (RECURSIVE)
- 3= Median

Description: Averaging type.

Parameter: int32_t SA_MovingCount SA_MovingCount

Direction: Up

Valid values:

- 2
- 4
- 8
- 16
- 32
- 64
- 128
- 256
- 512

Description: Number of value for the averaging window. This parameter is only available at moving average.

Parameter: int32_t SA_RecursiveCount SA_RecursiveCount

Direction: Up

Valid values:

- Minimum:** 2, 4, 8, ...,
- Maximum:** 32768

Description: Number of values for recursive averaging. This parameter is only available at recursive average.

Parameter: int32_t SA_MedianCount SA_MedianCount

Direction: Up

Valid values:

- 3
- 5
- 7
- 9

Description: Number of values to build median. This parameter is only available at median.

17.8.2.2.3 Set_MasterValue (MASTERMV)

Set the master value.

Parameter: int32_t SP_Master

SP_Master

Direction: Down

Valid values:

0= no (NONE)

1= yes (MASTER)

Description: Specifies if mastering should be done or resetted.

Parameter: double SP_MasterValue

SP_MasterValue

Direction: Down

Valid values:

Minimum: -1024.0

Maximum: +1024.0

Unit: mm

Description: Master value

17.8.2.2.4 Get_MasterValue (MASTERMV)

Get the master value.

Parameter: int32_t SA_Master

SA_Master

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= no (NONE)

1= yes (MASTER)

Description: Specifies if mastering is active.

Parameter: double SA_MasterValue

SA_MasterValue

Direction: Up

Valid values:

Minimum: -1024.0

Maximum: +1024.0

Unit: mm

Description: Master value

17.8.2.2.5 Set_DigitalScale<n> (DIGITALSCALE<n>)

Set the scaling factor for digital output n ∈ {1..2}.

Valid for sensor:

C-Box/2A

thicknessSENSOR

Parameter: int32_t SP_DigitalScaleMode<n>

SP_DigitalScaleMode<n>

Direction: Down

Valid values:

0= None

1= Master

Description: Digital scale mode.

Parameter: double SP_DigitalScaleValue<n> SP_DigitalScaleValue<n>
Direction: Down
Valid values:
Minimum: -1024.0
Maximum: 1024.0
Unit: mm
Description: Value which represents first/second offset value.

17.8.2.2.6 Get_DigitalScale<n> (DIGITALSCALE<n>)

Get the scaling factor for digital output n ∈ {1..2}.

Valid for sensor:
 C-Box/2A
 thicknessSENSOR

Parameter: int32_t SA_DigitalScaleMode<n> SA_DigitalScaleMode<n>
Direction: Up
Valid values:
 -1 = Unknown parameter value from sensor
 0 = None
 1 = Master
Description: Digital scale mode.

Parameter: double SA_DigitalScaleValue<n> SA_DigitalScaleValue<n>
Direction: Up
Valid values:
Minimum: -1024.0
Maximum: 1024.0
Unit: mm
Description: Value which represents first/second offset value.

17.8.3 Data output

17.8.3.1 General

17.8.3.1.1 Set_DataOutInterface (OUTPUT)

Set the active interface for data output.

At this command the sensor may change output data automatically. If first bit of [IP_AutomaticMode](#) is set (1), [Get_AllParameters](#) (SP_Additional= 0) is called automatically after this command. Otherwise, you have to call it manually. After this call [Get_TransmittedDataInfo](#) to retrieve new setting.

Parameter: int32_t SP_DataOutInterface SP_DataOutInterface
Direction: Down
Valid values:
 0 = None
 1 = RS422 (only at C-Box Analog)
 2 = Ethernet
 3 = HTTP
 5 = USB (only at C-Box Analog and C-Box/2A)
Description: Active interface for data output.

17.8.3.1.2 Get_DataOutInterface (OUTPUT)

Get the active interface for data output.

Parameter: int32_t SA_DataOutInterface

SA_DataOutInterface

Direction: Up

Valid values:

- 1 = Unknown parameter value from sensor
- 0 = None
- 1 = RS422 (only at C-Box Analog)
- 2 = Ethernet
- 3 = HTTP
- 5 = USB (only at C-Box Analog and C-Box/2A)

Description: Active interface for data output.

17.8.3.1.3 Set_Resampling (OUTREDUCE)

Set resampling to reduce output data.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Parameter: int32_t SP_Resampling

SP_Resampling

Direction: Down

Valid values:

- Minimum:** 1
- Maximum:** 1000

Description: Resampling value.

Parameter: int32_t SP_ResampleAnalog

SP_ResampleAnalog

Direction: Down

Valid values:

- 0 = no
- 1 = yes

Description: Specify if analog output should be resampled.

Parameter: int32_t SP_ResampleRS422

SP_ResampleRS422

Direction: Down

Valid values:

- 0 = no
- 1 = yes

Valid for sensor:

C-Box Analog

Description: Specify if RS422 output should be resampled.

Parameter: int32_t SP_ResampleUSB

SP_ResampleUSB

Direction: Down

Valid values:

- 0 = no
- 1 = yes

Valid for sensor:

C-Box Analog

C-Box/2A

Description: Specify if output on USB should be resampled.

Parameter: int32_t SP_ResampleEthernet SP_ResampleEthernet
Direction: Down
Valid values:
 0= no
 1= yes
Description: Specify if output over ethernet should be resampled.

17.8.3.1.4 Get_Resampling (OUTREDUCE)

Get resampling for reducing output data.

Parameter: int32_t SA_Resampling SA_Resampling
Direction: Up
Valid values:
Minimum: 1
Maximum: 1000
Description: Resampling value.

Parameter: int32_t SA_ResampleAnalog SA_ResampleAnalog
Direction: Up
Valid values:
 0= no
 1= yes
Description: Analog output is resampled.

Parameter: int32_t SA_ResampleRS422 SA_ResampleRS422
Direction: Up
Valid values:
 0= no
 1= yes
Valid for sensor:
 C-Box Analog
Description: RS422 output is resampled.

Parameter: int32_t SA_ResampleUSB SA_ResampleUSB
Direction: Up
Valid values:
 0= no
 1= yes
Valid for sensor:
 C-Box Analog
 C-Box/2A
Description: Output on USB is resampled.

Parameter: int32_t SA_ResampleEthernet SA_ResampleEthernet
Direction: Up
Valid values:
 0= no
 1= yes
Description: Output over ethernet is resampled.

17.8.3.1.5 Set_HoldLastValid (OUTHOLD)

Set the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SP_HoldLastValid

SP_HoldLastValid

Direction: Down

Valid values:

Minimum: -1

Maximum: 1024

Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).

17.8.3.1.6 Get_HoldLastValid (OUTHOLD)

Get the number of values to be replaced by last valid value instead of error values.

Parameter: int32_t SA_HoldLastValid

SA_HoldLastValid

Direction: Up

Valid values:

Minimum: -1

Maximum: 1024

Description: Values to replace by last valid value. -1 means no value to hold, 0 means never output an error value (always hold last valid value).

17.8.3.1.7 Set_FramesPerPacket_ETH (MEASCNT_ETH)

Set the maximum number of frames in ethernet packet.

Parameter: int32_t SP_FramesPerPacket_ETH

SP_FramesPerPacket_ETH

Direction: Down

Valid values:

Minimum: 0

Maximum: 200

Description: Desired number of frames in ethernet packet.

0 means automatic (each 10 ms a packet). If packets cannot be send as fast as needed, the number of frames may be exceeded. If maximum packet size is not sufficient, the number of frames is reduced to fit into a packet.

17.8.3.1.8 Get_FramesPerPacket_ETH (MEASCNT_ETH)

Get the maximum number of frames in ethernet packet.

Parameter: int32_t SA_FramesPerPacket_ETH

SA_FramesPerPacket_ETH

Direction: Up

Valid values:

Minimum: 0

Maximum: 200

Description: Desired number of frames in ethernet packet.

0 means automatic (each 10 ms a packet). If packets cannot be send as fast as needed, the number of frames may be exceeded. If maximum packet size is not sufficient, the number of frames is reduced to fit into a packet.

17.8.3.1.9 Set_OutputScale_RS422_USB (OUTSCALE_RS422_USB)

Set the scaling factor for RS422 (only C-Box Analog) and USB output.
 If first bit of [IP_AutomaticMode](#) is set (1), [Get_OutputScale_RS422_USB](#) is called automatically after this command. Otherwise, you have to call it manually.
 For this command an [Update_...](#) meta command is available.

Valid for sensor:

C-Box Analog
 C-Box/2A

Parameter: int32_t SP_OutputScaleMode

SP_OutputScaleMode

Direction: Down

Valid values:

0= Standard
 1= Two point (TWOPOINT)

Description: Output scale mode.

Parameter: double SP_OutputMinValue

SP_OutputMinValue

Direction: Down

Valid values:

Minimum: -1024.0
Maximum: 1024.0

Unit: mm

Description: Lowest possible value to transmit over RS422 (only at C-Box Analog) or USB.

Parameter: double SP_OutputMaxValue

SP_OutputMaxValue

Direction: Down

Valid values:

Minimum: -1024.0
Maximum: 1024.0

Unit: mm

Description: Highest possible value to transmit over RS422 (only at C-Box Analog) or USB.

17.8.3.1.10 Get_OutputScale_RS422_USB (OUTSCALE_RS422_USB)

Get the scaling factor for RS422 (only C-Box Analog) and USB output.

Valid for sensor:

C-Box Analog
 C-Box/2A

Parameter: int32_t SA_OutputScaleMode

SA_OutputScaleMode

Direction: Up

Valid values:

0= Standard
 1= Two point (TWOPOINT)

Description: Output scale mode.

Parameter: double SA_OutputMinValue	SA_OutputMinValue
Direction: Up	
Valid values:	
Minimum: -1024.0	
Maximum: 1024.0	
Unit: mm	
Description: Lowest possible value to transmit over RS422 (only at C-Box Analog) or USB.	
Parameter: double SA_Output.MaxValue	SA_Output.MaxValue
Direction: Up	
Valid values:	
Minimum: -1024.0	
Maximum: 1024.0	
Unit: mm	
Description: Highest possible value to transmit over RS422 (only at C-Box Analog) or USB.	

17.8.3.2 Selected measurement values

17.8.3.2.1 Set_Output_RS422 (OUT_RS422)

Set the data to be output at RS422 interface.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Valid for sensor:

C-Box Analog

Parameter: int32_t SP_OutputSensor1Value_RS422	SP_OutputSensor1Value_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 1 measure value is transmitted.	
Parameter: int32_t SP_OutputSensor1Additional_RS422	SP_OutputSen- sor1Additional_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 1 additional value is transmitted.	
Parameter: int32_t SP_OutputSensor2Value_RS422	SP_OutputSensor2Value_- RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 2 measure value is transmitted.	
Parameter: int32_t SP_OutputSensor2Additional_RS422	SP_OutputSen- sor2Additional_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 2 additional value is transmitted.	

Parameter: int32_t SP_OutputC-BoxValue_RS422	SP_OutputC-BoxValue_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if C-Box measure value is transmitted.	
Parameter: int32_t SP_OutputC-BoxAdditional_RS422	SP_OutputC-BoxAdditional_RS422
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if C-Box additional value is transmitted.	

17.8.3.2.2 Get_Output_RS422 (OUT_RS422)

Get the data to be output at RS422 interface.

Valid for sensor:

C-Box Analog

Parameter: int32_t SA_OutputSensor1Value_RS422	SA_OutputSensor1Value_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 1 measure value is transmitted.	
Parameter: int32_t SA_OutputSensor1Additional_RS422	SA_OutputSensor1Additional_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 1 additional value is transmitted.	
Parameter: int32_t SA_OutputSensor2Value_RS422	SA_OutputSensor2Value_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 2 measure value is transmitted.	
Parameter: int32_t SA_OutputSensor2Additional_RS422	SA_OutputSensor2Additional_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 2 additional value is transmitted.	
Parameter: int32_t SA_OutputC-BoxValue_RS422	SA_OutputC-BoxValue_RS422
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if C-Box measure value is transmitted.	

Parameter: int32_t SA_OutputC-BoxAdditional_RS422

SA_OutputC-
BoxAdditional_RS422

Direction: Up

Valid values:

0= no

1= yes

Description: Specify if C-Box additional value is transmitted.

17.8.3.2.3 Set_Output_USB (OUT_USB)

Set the data to be output at USB interface.

For this command an [Update_...](#) and a [Reset_...](#) meta command is available.

Valid for sensor:

C-Box Analog

C-Box/2A

Parameter: int32_t SP_OutputSensor1Value_USB

SP_OutputSensor1Value_USB

Direction: Down

Valid values:

0= no

1= yes

Description: Specify if sensor 1 measure value is transmitted.

Parameter: int32_t SP_OutputSensor1Additional_USB

SP_OutputSen-
sor1Additional_USB

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

C-Box Analog

Description: Specify if sensor 1 additional value is transmitted.

Parameter: int32_t SP_OutputSensor1Intensity_USB

SP_OutputSen-
sor1Intensity_USB

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

C-Box/2A

Description: Specify if sensor 1 intensity value is transmitted.

Parameter: int32_t SP_OutputSensor1Shutter_USB

SP_OutputSensor1Shutter_-
USB

Direction: Down

Valid values:

0= no

1= yes

Valid for sensor:

C-Box/2A

Description: Specify if sensor 1 shutter value is transmitted.

Parameter: int32_t SP_OutputSensor1Reflectivity_USB	SP_OutputSensor1Reflectivity_USB
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
Description: Specify if sensor 1 reflectivity value (calculated by C-Box) is transmitted.	
Parameter: int32_t SP_OutputSensor2Value_USB	SP_OutputSensor2Value_USB
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 2 measure value is transmitted.	
Parameter: int32_t SP_OutputSensor2Additional_USB	SP_OutputSensor2Additional_USB
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
Description: Specify if sensor 2 additional value is transmitted.	
Parameter: int32_t SP_OutputSensor2Intensity_USB	SP_OutputSensor2Intensity_USB
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
Description: Specify if sensor 2 intensity value is transmitted.	
Parameter: int32_t SP_OutputSensor2Shutter_USB	SP_OutputSensor2Shutter_USB
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
Description: Specify if sensor 2 shutter value is transmitted.	
Parameter: int32_t SP_OutputSensor2Reflectivity_USB	SP_OutputSensor2Reflectivity_USB
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
Description: Specify if sensor 2 reflectivity value (calculated by C-Box) is transmitted.	

Parameter: int32_t SP_OutputC-BoxValue_USB	SP_OutputC-BoxValue_USB
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if C-Box measure value is transmitted.	
Parameter: int32_t SP_OutputC-BoxAdditional_USB	SP_OutputC-BoxAdditional_USB
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
Description: Specify if C-Box additional value is transmitted.	
Parameter: int32_t SP_OutputC-BoxCounter_USB	SP_OutputC-BoxCounter_USB
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
Description: Specify if C-Box counter value is transmitted.	
Parameter: int32_t SP_OutputC-BoxTimestamp_USB	SP_OutputC-BoxTimestamp_USB
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
Description: Specify if C-Box timestamp value is transmitted.	
Parameter: int32_t SP_OutputC-BoxDigital_USB	SP_OutputC-BoxDigital_USB
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
Description: Specify if C-Box digital signals are transmitted.	

17.8.3.2.4 Get_Output_USB (OUT_USB)

Get the data to be output at USB interface.

Valid for sensor:

 C-Box Analog
 C-Box/2A

Parameter: int32_t SA_OutputSensor1Value_USB	SA_OutputSensor1Value_USB
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 1 measure value is transmitted.	

Parameter: int32_t SA_OutputSensor1Additional_USB	SA_OutputSensor1Additional_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description: Specify if sensor 1 additional value is transmitted.	
Parameter: int32_t SA_OutputSensor1Intensity_USB	SA_OutputSensor1Intensity_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 1 intensity value is transmitted.	
Parameter: int32_t SA_OutputSensor1Shutter_USB	SA_OutputSensor1Shutter_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 1 shutter value is transmitted.	
Parameter: int32_t SA_OutputSensor1Reflectivity_USB	SA_OutputSensor1Reflectivity_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 1 reflectivity value (calculated by C-Box) is transmitted.	
Parameter: int32_t SA_OutputSensor2Value_USB	SA_OutputSensor2Value_USB
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 2 measure value is transmitted.	
Parameter: int32_t SA_OutputSensor2Additional_USB	SA_OutputSensor2Additional_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description: Specify if sensor 2 additional value is transmitted.	
Parameter: int32_t SA_OutputSensor2Intensity_USB	SA_OutputSensor2Intensity_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 2 intensity value is transmitted.	
Parameter: int32_t SA_OutputSensor2Shutter_USB	SA_OutputSensor2Shutter_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 2 shutter value is transmitted.	

Parameter: int32_t SA_OutputSensor2Reflectivity_USB	SA_OutputSensor2Reflectivity_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if sensor 2 reflectivity value (calculated by C-Box) is transmitted.	
Parameter: int32_t SA_OutputC-BoxValue_USB	SA_OutputC-BoxValue_USB
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if C-Box measure value is transmitted.	
Parameter: int32_t SA_OutputC-BoxAdditional_USB	SA_OutputC-BoxAdditional_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description: Specify if C-Box additional value is transmitted.	
Parameter: int32_t SA_OutputC-BoxCounter_USB	SA_OutputC-BoxCounter_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if C-Box counter value is transmitted.	
Parameter: int32_t SA_OutputC-BoxTimestamp_USB	SA_OutputC-BoxTimestamp_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if C-Box timestamp value is transmitted.	
Parameter: int32_t SA_OutputC-BoxDigital_USB	SA_OutputC-BoxDigital_USB
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A)	
Description: Specify if C-Box digital signals are transmitted.	

17.8.3.2.5 Set_Output_ETH (OUT_ETH)

Set the data to be output at ethernet interface.

For this command an [Update ...](#) and a [Reset ...](#) meta command is available.

Parameter: int32_t SP_OutputSensor1Value_ETH	SP_OutputSensor1Value_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 1 measure value is transmitted.	

Parameter: int32_t SP_OutputSensor1Additional_ETH	SP_OutputSensor1Additional_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
Description:	Specify if sensor 1 additional value is transmitted.
Parameter: int32_t SP_OutputSensor1Intensity_ETH	SP_OutputSensor1Intensity_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description:	Specify if sensor 1 intensity value is transmitted.
Parameter: int32_t SP_OutputSensor1Shutter_ETH	SP_OutputSensor1Shutter_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description:	Specify if sensor 1 shutter value is transmitted.
Parameter: int32_t SP_OutputSensor1Reflectivity_ETH	SP_OutputSensor1Reflectivity_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description:	Specify if sensor 1 reflectivity value (calculated by C-Box) is transmitted.
Parameter: int32_t SP_OutputSensor2Value_ETH	SP_OutputSensor2Value_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description:	Specify if sensor 2 measure value is transmitted.
Parameter: int32_t SP_OutputSensor2Additional_ETH	SP_OutputSensor2Additional_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
Description:	Specify if sensor 2 additional value is transmitted.

Parameter: int32_t SP_OutputSensor2Intensity_ETH	SP_OutputSensor2Intensity_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description: Specify if sensor 2 intensity value is transmitted.	
Parameter: int32_t SP_OutputSensor2Shutter_ETH	SP_OutputSensor2Shutter_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description: Specify if sensor 2 shutter value is transmitted.	
Parameter: int32_t SP_OutputSensor2Reflectivity_ETH	SP_OutputSensor2Reflectivity_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description: Specify if sensor 2 reflectivity value (calculated by C-Box) is transmitted.	
Parameter: int32_t SP_OutputC-BoxValue_ETH	SP_OutputC-BoxValue_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Description: Specify if C-Box measure value is transmitted.	
Parameter: int32_t SP_OutputC-BoxAdditional_ETH	SP_OutputC-BoxAdditional_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box Analog	
Description: Specify if C-Box additional value is transmitted.	
Parameter: int32_t SP_OutputC-BoxCounter_ETH	SP_OutputC-BoxCounter_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description: Specify if C-Box counter value is transmitted.	

Parameter: int32_t SP_OutputC-BoxTimestamp_ETH	SP_OutputC-BoxTimestamp_
Direction: Down	ETH
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description:	Specify if C-Box timestamp value is transmitted.
Parameter: int32_t SP_OutputC-BoxDigital_ETH	SP_OutputC-BoxDigital_ETH
Direction: Down	
Valid values:	
0= no	
1= yes	
Valid for sensor:	
C-Box/2A	
thicknessSENSOR	
Description:	Specify if C-Box digital signals are transmitted.

17.8.3.2.6 Get_Output_ETH (OUT_ETH)

Get the data to be output at ethernet interface.

Parameter: int32_t SA_OutputSensor1Value_ETH	SA_OutputSensor1Value_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description:	Specify if sensor 1 measure value is transmitted.
Parameter: int32_t SA_OutputSensor1Additional_ETH	SA_OutputSen-
Direction: Up	sor1Additional_ETH
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description:	Specify if sensor 1 additional value is transmitted.
Parameter: int32_t SA_OutputSensor1Intensity_ETH	SA_OutputSen-
Direction: Up	sor1Intensity_ETH
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description:	Specify if sensor 1 intensity value is transmitted.
Parameter: int32_t SA_OutputSensor1Shutter_ETH	SA_OutputSensor1Shutter_-
Direction: Up	ETH
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description:	Specify if sensor 1 shutter value is transmitted.

Parameter: int32_t SA_OutputSensor1Reflectivity_ETH	SA_OutputSensor1Reflectivity_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if sensor 1 reflectivity value (calculated by C-Box) is transmitted.	
Parameter: int32_t SA_OutputSensor2Value_ETH	SA_OutputSensor2Value_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if sensor 2 measure value is transmitted.	
Parameter: int32_t SA_OutputSensor2Additional_ETH	SA_OutputSensor2Additional_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description: Specify if sensor 2 additional value is transmitted.	
Parameter: int32_t SA_OutputSensor2Intensity_ETH	SA_OutputSensor2Intensity_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if sensor 2 intensity value is transmitted.	
Parameter: int32_t SA_OutputSensor2Shutter_ETH	SA_OutputSensor2Shutter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if sensor 2 shutter value is transmitted.	
Parameter: int32_t SA_OutputSensor2Reflectivity_ETH	SA_OutputSensor2Reflectivity_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if sensor 2 reflectivity value (calculated by C-Box) is transmitted.	
Parameter: int32_t SA_OutputC-BoxValue_ETH	SA_OutputC-BoxValue_ETH
Direction: Up	
Valid values:	
0= no	
1= yes	
Description: Specify if C-Box measure value is transmitted.	
Parameter: int32_t SA_OutputC-BoxAdditional_ETH	SA_OutputC-BoxAdditional_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box Analog)	
Description: Specify if C-Box additional value is transmitted.	

Parameter: int32_t SA_OutputC-BoxCounter_ETH	SA_OutputC-BoxCounter_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if C-Box counter value is transmitted.	
Parameter: int32_t SA_OutputC-BoxTimestamp_ETH	SA_OutputC-BoxTimestamp_- ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if C-Box timestamp value is transmitted.	
Parameter: int32_t SA_OutputC-BoxDigital_ETH	SA_OutputC-BoxDigital_ETH
Direction: Up	
Valid values:	
0= no	
1= yes (only at C-Box/2A and thicknessSENSOR)	
Description: Specify if C-Box digital signals are transmitted.	

17.8.3.2.7 Set_OutputAdditional (OUT_ADDITIONAL)

Set the additional data to be output.

Valid for sensor:

C-Box Analog

Parameter: int32_t SP_OutputAdditional	SP_OutputAdditional
Direction: Down	
Valid values:	
0= C-Box counter (C-BOXCOUNTER)	
1= C-Box timestamp (C-BOXTIMESTAMP)	
2= C-Box trigger input (TRG-IN)	
Description: Specify which additional value should be transmitted.	

17.8.3.2.8 Get_OutputAdditional (OUT_ADDITIONAL)

Get the additional data to be output.

Valid for sensor:

C-Box Analog

Parameter: int32_t SA_OutputAdditional	SA_OutputAdditional
Direction: Up	
Valid values:	
0= C-Box counter (C-BOXCOUNTER)	
1= C-Box timestamp (C-BOXTIMESTAMP)	
2= C-Box trigger input (TRG-IN)	
Description: Specify which additional value is transmitted.	

17.8.3.3 Switching outputs

17.8.3.3.1 Set_ErrorOutput<n> (ERROROUT<n>)

Set condition to be used to set error output n ∈ {1..2}.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: int32_t SP_ErrorOutput<n>

SP_ErrorOutput<n>

Direction: Down

Valid values:

- 0= Pass through first error output of sensor1 (SENSOR1ERROROUT1)
- 1= Pass through second error output of sensor1 (SENSOR1ERROROUT2)
- 2= Pass through first error output of sensor2 (SENSOR2ERROROUT1)
- 3= Pass through second error output of sensor2 (SENSOR2ERROROUT2)
- 4= Range check measure value of sensor 1 (SENSOR1VALUE)
- 5= Range check intensity value of sensor 1 (SENSOR1INTENSITY)
- 6= Range check shutter time of sensor 1 (SENSOR1SHUTTER)
- 7= Range check reflectivity of sensor 1 (SENSOR1REFLECTIVITY)
- 8= Range check measure value of sensor 2 (SENSOR2VALUE)
- 9= Range check intensity value of sensor 2 (SENSOR2INTENSITY)
- 10= Range check shutter time of sensor 2 (SENSOR2SHUTTER)
- 11= Range check reflectivity of sensor 2 (SENSOR2REFLECTIVITY)
- 12= Range check calculated value of C-Box (C-BOXVALUE)
- 13= Always low (LOW)
- 14= Always high (HIGH)

Description: Condition for error output.

17.8.3.3.2 Get_ErrorOutput<n> (ERROROUT<n>)

Get condition to be used to set error output n ∈ {1..2}.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: int32_t SA_ErrorOutput<n>

SA_ErrorOutput<n>

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Pass through first error output of sensor1 (SENSOR1ERROROUT1)
- 1= Pass through second error output of sensor1 (SENSOR1ERROROUT2)
- 2= Pass through first error output of sensor2 (SENSOR2ERROROUT1)
- 3= Pass through second error output of sensor2 (SENSOR2ERROROUT2)
- 4= Range check measure value of sensor 1 (SENSOR1VALUE)
- 5= Range check intensity value of sensor 1 (SENSOR1INTENSITY)
- 6= Range check shutter time of sensor 1 (SENSOR1SHUTTER)
- 7= Range check reflectivity of sensor 1 (SENSOR1REFLECTIVITY)
- 8= Range check measure value of sensor 2 (SENSOR2VALUE)
- 9= Range check intensity value of sensor 2 (SENSOR2INTENSITY)

- 10= Range check shutter time of sensor 2 (SENSOR2SHUTTER)
- 11= Range check reflectivity of sensor 2 (SENSOR2REFLECTIVITY)
- 12= Range check calculated value of C-Box (C-BOXVALUE)
- 13= Always low (LOW)
- 14= Always high (HIGH)

Description: Condition for error output.

17.8.3.3.3 Set_ErrorLimit<n> (ERRORLIMIT<n>)

Set the error limits (at range check) of error output n $\in \{1..2\}$.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: double SP_LowerLimit<n>

SP_LowerLimit<n>

Direction: Down

Valid values:

- Minimum:** -1024.0 (for distance/thickness), 0 (for intensity/shutter)
- Maximum:** 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0 (for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_ErrorOutput<n>](#)

Description: Lower limit.

Parameter: double SP_UpperLimit<n>

SP_UpperLimit<n>

Direction: Down

Valid values:

- Minimum:** -1024.0 (for distance/thickness), 0 (for intensity/shutter)
- Maximum:** 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0 (for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_ErrorOutput<n>](#)

Description: Upper limit.

17.8.3.3.4 Get_ErrorLimit<n> (ERRORLIMIT<n>)

Get the error limits (at range check) of error output n $\in \{1..2\}$.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: double SA_LowerLimit<n>

SA_LowerLimit<n>

Direction: Up

Valid values:

- Minimum:** -1024.0 (for distance/thickness), 0 (for intensity/shutter)
- Maximum:** 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0 (for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_ErrorOutput<n>](#)

Description: Lower limit.

Parameter: double SA_UpperLimit<n> SA_UpperLimit<n>

Direction: Up

Valid values:

- Minimum:** -1024.0 (for distance/thickness), 0 (for intensity/shutter)
- Maximum:** 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0 (for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_ErrorOutput<n>](#)

Description: Upper limit.

17.8.3.4 Analog output

17.8.3.4.1 Set_AnalogOutput (ANALOGOUT)

Set the data to be used for analog output.

For C-Box/2A or thicknessSensor, this command is identical to [Set_AnalogOutput<n>](#).

Parameter: int32_t SP_AnalogOutput SP_AnalogOutput

Direction: Down

Valid values:

- 0= Fixed
- 1= Sensor 1 value (SENSOR1VALUE)
- 2= Sensor 2 value (SENSOR2VALUE)
- 3= C-Box value (C-BOXVALUE)

Description: Data to be used for analog output.

Parameter: double SP_AnalogValue SP_AnalogValue

Direction: Down

Valid values:

- Minimum:** 0, -5, -10 or 4, depending on analog range
- Maximum:** 5, 10 or 20, depending on analog range

Unit: V or mA, depending on analog range

Description: Fixed output value for analog output.

17.8.3.4.2 Get_AnalogOutput (ANALOGOUT)

Get the data to be used for analog output.

. For C-Box/2A or thicknessSensor, this command is identical to [Get_AnalogOutput<n>](#).

Parameter: int32_t SA_AnalogOutput SA_AnalogOutput

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= Fixed
- 1= Sensor 1 value (SENSOR1VALUE)
- 2= Sensor 2 value (SENSOR2VALUE)
- 3= C-Box value (C-BOXVALUE)

Description: Data to be used for analog output.

Parameter: double SA_AnalogValue SA_AnalogValue
Direction: Up
Valid values:
Minimum: 0, -5, -10 or 4, depending on analog range
Maximum: 5, 10 or 20, depending on analog range
Unit: V or mA, depending on analog range
Description: Fixed output value for analog output.

17.8.3.4.3 Set_AnalogOutput<n> (ANALOGOUT<n>)

Set the data to be used for analog output n ∈ {1..2}.

Valid for sensor:
 C-Box/2A
 thicknessSENSOR

Parameter: int32_t SP_AnalogOutput<n> SP_AnalogOutput<n>
Direction: Down
Valid values:
 0= Fixed
 1= Sensor 1 value (SENSOR1VALUE)
 2= Sensor 2 value (SENSOR2VALUE)
 3= C-Box value (C-BOXVALUE)
 4= Sensor 1 intensity value (SENSOR1INTENSITY)
 5= Sensor 1 shutter time (SENSOR1SHUTTER)
 6= Sensor 1 reflectivity (SENSOR1REFLECTIVITY)
 7= Sensor 2 intensity value (SENSOR2INTENSITY)
 8= Sensor 2 shutter time (SENSOR2SHUTTER)
 9= Sensor 2 reflectivity (SENSOR2REFLECTIVITY)
Description: Data to be used for analog output.

Parameter: double SP_AnalogValue<n> SP_AnalogValue<n>
Direction: Down
Valid values:
Minimum: 0, -5, -10 or 4, depending on analog range
Maximum: 5, 10 or 20, depending on analog range
Unit: V or mA, depending on analog range
Description: Fixed output value for analog output.

17.8.3.4.4 Get_AnalogOutput<n> (ANALOGOUT<n>)

Get the data to be used for analog output n ∈ {1..2}.

Valid for sensor:
 C-Box/2A
 thicknessSENSOR

Parameter: int32_t SA_AnalogOutput<n> SA_AnalogOutput<n>
Direction: Up
Valid values:
 -1= Unknown parameter value from sensor
 0= Fixed

- 1= Sensor 1 value (SENSOR1VALUE)
- 2= Sensor 2 value (SENSOR2VALUE)
- 3= C-Box value (C-BOXVALUE)
- 4= Sensor 1 intensity value (SENSOR1INTENSITY)
- 5= Sensor 1 shutter time (SENSOR1SHUTTER)
- 6= Sensor 1 reflectivity (SENSOR1REFLECTIVITY)
- 7= Sensor 2 intensity value (SENSOR2INTENSITY)
- 8= Sensor 2 shutter time (SENSOR2SHUTTER)
- 9= Sensor 2 reflectivity (SENSOR2REFLECTIVITY)

Description: Data to be used for analog output.

Parameter: double SA_AnalogValue<n>

SA_AnalogValue<n>

Direction: Up

Valid values:

Minimum: 0, -5, -10 or 4, depending on analog range

Maximum: 5, 10 or 20, depending on analog range

Unit: V or mA, depending on analog range

Description: Fixed output value for analog output.

17.8.3.4.5 Set_AnalogRange (ANALOG RANGE)

Set the analog output range.

At C-Box/2A and thicknessSENSOR, this command is identical to [Set_AnalogRange<n>](#).

Parameter: int32_t SP_AnalogRange

SP_AnalogRange

Direction: Down

Valid values:

- 0= None
- 1= 0 - 5V
- 2= 0 - 10V
- 3= -5 - 5V
- 4= -10 - 10V
- 5= 4 - 20mA

Description: Analog output range.

17.8.3.4.6 Get_AnalogRange (ANALOG RANGE)

Get the analog output range. At C-Box/2A and thicknessSENSOR, this command is identical to [Get_AnalogRange<n>](#).

Parameter: int32_t SA_AnalogRange

SA_AnalogRange

Direction: Up

Valid values:

- 1= Unknown parameter value from sensor
- 0= None
- 1= 0 - 5V
- 2= 0 - 10V
- 3= -5 - 5V
- 4= -10 - 10V
- 5= 4 - 20mA

Description: Analog output range.

17.8.3.4.7 Set_AnalogRange<n> (ANALOG RANGE<n>)

Set the analog output range $n \in \{1..2\}$.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: int32_t SP_AnalogRange<n>

SP_AnalogRange<n>

Direction: Down

Valid values:

0= None
1= 0 - 5V
2= 0 - 10V
3= -5 - 5V
4= -10 - 10V
5= 4 - 20mA

Description: Analog output range.

17.8.3.4.8 Get_AnalogRange<n> (ANALOG RANGE<n>)

Get the analog output range $n \in \{1..2\}$.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: int32_t SA_AnalogRange<n>

SA_AnalogRange<n>

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= None
1= 0 - 5V
2= 0 - 10V
3= -5 - 5V
4= -10 - 10V
5= 4 - 20mA

Description: Analog output range.

17.8.3.4.9 Set_AnalogScale (ANALOG SCALE)

Set the scaling factor for analog output. If both parameters are zero, the standard scaling is used.

At C-Box/2A and thicknessSENSOR, this command is identical to [Set_AnalogScale<n>](#).

Parameter: int32_t SP_AnalogScaleMode

SP_AnalogScaleMode

Direction: Down

Valid values:

0= Standard
1= Two point (TWOPOINT)

Description: Analog scale mode.

Parameter: double SP_MinValue SP_MinValue

Direction: Down

Valid values:

Minimum: -1024.0

Maximum: 1024.0

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SP_MaxValue SP_MaxValue

Direction: Down

Valid values:

Minimum: -1024.0

Maximum: 1024.0

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

17.8.3.4.10 Get_AnalogScale (ANALOGSCALE)

Get the scaling factor for analog output. If both parameters are zero, the standard scaling is used.

At C-Box/2A and thicknessSENSOR, this command is identical to [Get_AnalogScale<n>](#).

Parameter: int32_t SA_AnalogScaleMode SA_AnalogScaleMode

Direction: Up

Valid values:

-1= Unknown parameter value from sensor

0= Standard

1= Two point (TWOPOINT)

Description: Analog scale mode.

Parameter: double SA_MinValue SA_MinValue

Direction: Up

Valid values:

Minimum: -1024.0

Maximum: 1024.0

Unit: mm

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SA_MaxValue SA_MaxValue

Direction: Up

Valid values:

Minimum: -1024.0

Maximum: 1024.0

Unit: mm

Description: Value which represents highest voltage/current (at two point scaling).

17.8.3.4.11 Set_AnalogScale<n> (ANALOGSCALE<n>)

Set the scaling factor for analog output $n \in \{1..2\}$. If both parameters are zero, the standard scaling is used.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: int32_t SP_AnalogScaleMode<n>

SP_AnalogScaleMode<n>

Direction: Down

Valid values:

0= Standard
1= Two point (TWOPOINT)

Description: Analog scale mode.

Parameter: double SP_MinValue<n>

SP_MinValue<n>

Direction: Down

Valid values:

Minimum: -1024.0 (for distance/thickness), 0 (for intensity/shutter)
Maximum: 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0
(for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_AnalogOutput<n>](#)

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SP_MaxValue<n>

SP_MaxValue<n>

Direction: Down

Valid values:

Minimum: -1024.0 (for distance/thickness), 0 (for intensity/shutter)
Maximum: 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0
(for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_AnalogOutput<n>](#)

Description: Value which represents highest voltage/current (at two point scaling).

17.8.3.4.12 Get_AnalogScale<n> (ANALOGSCALE<n>)

Get the scaling factor for analog output $n \in \{1..2\}$. If both parameters are zero, the standard scaling is used.

Valid for sensor:

C-Box/2A
thicknessSENSOR

Parameter: int32_t SA_AnalogScaleMode<n>

SA_AnalogScaleMode<n>

Direction: Up

Valid values:

-1= Unknown parameter value from sensor
0= Standard
1= Two point (TWOPOINT)

Description: Analog scale mode.

Parameter: double SA_MinValue<n> SA_MinValue<n>

Direction: Up

Valid values:

Minimum: -1024.0 (for distance/thickness), 0 (for intensity/shutter)
Maximum: 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0
 (for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_AnalogOutput<n>](#)

Description: Value which represents lowest voltage/current (at two point scaling).

Parameter: double SA_MaxValue<n> SA_MaxValue<n>

Direction: Up

Valid values:

Minimum: -1024.0 (for distance/thickness), 0 (for intensity/shutter)
Maximum: 1024.0 (for distance/thickness), 100.0 (for intensity), 4000.0
 (for shutter)

Unit: mm (for distance/thickness), % (for intensity) or μs (for shutter), depending on [Set_AnalogOutput<n>](#)

Description: Value which represents highest voltage/current (at two point scaling).

17.8.4 Internal commands

17.8.4.1 Get_FirmwareVersion

Retrieve firmware version from controller.

This is an internal command. It should not be used by the customer.

Parameter: String CA_FirmwareVersion CA_FirmwareVersion

Direction: Up

Description: Firmware version

17.8.4.2 Prepare_UpdateFirmware

Prepares a firmware update at controller.

This is an internal command. It should not be used by the customer.

Parameter: Binary data XP_FirmwareFile XP_FirmwareFile

Direction: Down

Description: Firmware file

Parameter: String CA_FileName CA_FileName

Direction: Up

Description: Internal name of firmware file

Parameter: String CA_Date CA_Date

Direction: Up

Description: Date of firmware file

Parameter: String CA_ArticleNumber CA_ArticleNumber

Direction: Up

Description: Article number of destination device

Parameter: String CA_SerialNumber CA_SerialNumber
Direction: Up
Description: Serial number of destination device

17.8.4.3 Start_UpdateFirmware

Start firmware update at controller.
 This is an internal command. It should not be used by the customer.

17.8.4.4 Get_UpdateFirmwareProgress

Update firmware version at controller.
 Attention! This function can takes up to 1 minute. This is an internal command. It should not be used by the customer.

Parameter: double CA_Progress CA_Progress
Direction: Up
Valid values:
Minimum: 0.0
Maximum: 1.0
Description: Progress of firmware update

Parameter: String CA_Description CA_Description
Direction: Up
Description: Current state of firmware update.

Parameter: int32_t CA_Result CA_Result
Direction: Up
Valid values:
 0= Failed
 1= Success
Description: Result of firmware update (only available at end of update).

Parameter: int32_t CA_Finished CA_Finished
Direction: Up
Valid values:
 -1= Firmware update is not prepared, call Prepare_UpdateFirmware first
 0= No
 1= Yes
Description: Tell if firmware update is in progress.

17.8.4.5 Generate_Firmware

This is an internal command. It should not be used by the customer.

17.9 Commands for MultiSensor

This sensor is a container for any sensors to retrieve synchronized data.

At [OpenSensor](#), all desired sensors for this container must be specified. Than the synchronization mode must be selected.

At [Poll](#) or [TransferData](#), the synchronized values from sensors are filled at the data arrays frame by frame. A data frame consists of the data frames of each sensor (in order of specifying the sensors at [OpenSensor](#)).

Example of values in data array (two sensors, first with two values and second with three values per frame):

```
Sensor1_Value1 Sensor1_Value2 Sensor2_Value1 Sensor2_Value2 Sensor2_Value3
Sensor1_Value1 Sensor1_Value2 Sensor2_Value1 Sensor2_Value2 Sensor2_Value3
```

...

For this sensor, no [SensorFinder commands](#) are possible. Search for the physical sensors first and use the returned interface parameters at [OpenSensor](#) than.

Several standard commands are not available for this container nor its contained sensors/interfaces: [Clear_Buffers](#) (use [Sync_Start](#) instead), [DataAvail_Event](#).

Several standard commands are not available for this container (but for contained sensors): [Automatic_Setup](#), [Cmd_Generic](#) (no commands available), [Open_DataSocket](#) (no ethernet sensor).

17.9.1 Special parameters at OpenSensor

All parameters specified here can be specified before calling [OpenSensor](#) applies to any sensor in this container. To specify a parameter for one sensor, the parameter must be postfixed by '#' and the channel number (starting with 1). If a parameter for a special channel is specified, this one is used, otherwise the common parameter (without '#' and channel number) is used. Therefore a parameter can be specified both for common use and as special parameter.

Parameter: int32_t IP_SensorCount	IP_SensorCount
Direction: Down	
Valid values:	
Minimum: 1	
Maximum: 2147483647 (INT32_MAX)	
Description: Number of sensors in this container.	
Parameter: int32_t IP_SensorType	IP_SensorType
Direction: Down	
Valid values:	
Any type which is allowed at CreateSensorInstance (except SENSOR_LL27xx and MULTI_SENSOR itself).	
Description: Type of sensor which should be synchronized. This parameter can be used as special parameter, too.	
Parameter: String IP_SensorTypeName	IP_SensorTypeName
Direction: Down	
Valid values:	
Any name which is allowed at CreateSensorInstByName (except SENSOR_LL27xx and MULTI_SENSOR itself).	
Description: Type name of found sensor. If IP_SensorType is specified, this parameter is ignored. This parameter can be used as special parameter, too.	

Any other parameter used at [OpenSensor](#) can be specified, too. They can be used as special parameter, too.

Example how to open a MultiSensor consists of two sensors.

```

DWORD instance= CreateSensorInstance (MULTI_SENSOR);
SetParameterInt (instance, "IP_SensorCount", 2);
/* Following parameter is same for both sensors, so use common variant.*/
SetParameterString (instance, "IP_Interface", "IF2004_USB");
SetParameterString (instance, "IP_SensorTypeName#1", "ILD1750");
SetParameterInt (instance, "IP_ChannelNumber#1", 0);
SetParameterInt (instance, "IP_SensorType#2", SENSOR_IFD2422);
SetParameterInt (instance, "IP_ChannelNumber#2", 2);
DWORD err= OpenSensor (instance);

```

17.9.2 Get_DrvSetting

Returns the current settings of this MultiSensor instance. The sensors are not affected by this command.

Parameter: int32_t IA_SensorCount

IA_SensorCount

Direction: Up

Valid values:

Minimum: 1

Maximum: 2147483647 (INT32_MAX)

Description: Number of sensors in this container. All following parameters exists from 1 to this number, e.g. IA_SensorType#1, IA_SensorType#2,

...

Parameter: int32_t IA_SensorType# 1..x

IA_SensorType# 1..x

Direction: Up

Valid values:

Any type which is allowed at [CreateSensorInstance](#) (except SENSOR_LLT27xx and MULTI_SENSOR itself).

Description: Type of sensor which is synchronized.

Parameter: String IA_SensorTypeName# 1..x

IA_SensorTypeName# 1..x

Direction: Up

Valid values:

Any name which is allowed at [CreateSensorInstByName](#) (except SENSOR_LLT27xx and MULTI_SENSOR itself).

Description: Type name of found sensor.

Parameter: String IA_Interface# 1..x

IA_Interface# 1..x

Direction: Up

Valid values:

- "RS232"
- "IF2004_USB"
- "IF2004"
- "IF2008"
- "IF2008_ETH"
- "TCP/IP"
- "WinUSB"

Description: Interface type where the sensor is plugged on.

For the used interface the interface specific parameters are returned as described at [RS232](#) and following. But each parameter is prefixed by IA_... (instead of IP_...) and postfixed by sensor index (...#1, ...#2, ...).

17.9.3 Get_SensorInstance

This internal command returns the MEDAQLib instance handle to the sensors in the container.

The handle returned can be used for any calling sensor command. [OpenSensor](#) or [CloseSensor](#) using this handle is not allowed. After calling [CloseSensor](#) or [ReleaseSensorInstance](#) using MultiSensor handle, this handle is no longer valid.

Parameter: int32_t IP_Sensor

IP_Sensor

Direction: Down

Valid values:

Minimum: 1 = First sensor

Maximum: Up to [IP_SensorCount](#)

Description: Number of the sensor.

Parameter: int32_t SA_InstanceHandle

SA_InstanceHandle

Direction: Up

Valid values:

Minimum: 1

Maximum: 2147483647 (INT32_MAX)

Description: Instance number which now can be used to access sensor directly.

17.9.4 Get_HardwareInstance

This internal command returns the MEDAQLib instance handle to the hardware interface used by sensors in the container.

The handle returned can be used for any calling sensor command. [OpenSensor](#) or [CloseSensor](#) using this handle is not allowed. After calling [CloseSensor](#) or [ReleaseSensorInstance](#) using MultiSensor handle, this handle is no longer valid.

Parameter: int32_t IP_Sensor

IP_Sensor

Direction: Down

Valid values:

Minimum: 1 = First sensor

Maximum: Up to [IP_SensorCount](#)

Description: Number of the sensor to return it's hardware interface instance handle.

Parameter: int32_t SA_InstanceHandle

SA_InstanceHandle

Direction: Up

Valid values:

Minimum: 1

Maximum: 2147483647 (INT32_MAX)

Description: Instance number which now can be used to access hardware interface directly.

17.9.5 Set_MaximumTimeDiff

MultiSensor generates an error message at [TransferData](#), if the data amount of the sensors in container differs by a greater value as allowed.

At [OpenSensor](#) IP_MaximumTimeDiff is set to 200.0 ms and IP_MaxFrameDiff is set to 1000 frames.

Parameter: double IP_MaximumTimeDiff

IP_MaximumTimeDiff

Direction: Down

Unit: ms

Valid values:

Minimum: -1.0

Maximum: 1.79769e+308 (DBL_MAX)

Default: keep old value

Description: Maximum allowed time in ms (number of data frames will be devided by datarate at calculation).

If it is set to -1.0, this check is disabled.

Parameter: int32_t IP_MaximumFrameDiff

IP_MaximumFrameDiff

Direction: Down

Unit: frames

Valid values:

Minimum: -1

Maximum: 2147483647 (INT32_MAX)

Default: keep old value

Description: Maximum allowed number of data frames which may differ in ring buffer of sensor instance.

If it is set to -1, this check is disabled.

17.9.6 Enable_ReferenceTimestamp

Call [Enable_ReferenceTimestamp](#) for each sensor instance.

For the instance specified by IP_RefTSChannel, the parameter [IP_ReferenceTimestamp](#) is transmitted and [IA_ReferenceTimestamp](#) is returned. For all other instances, it is disabled (set to None).

Parameter: int32_t IP_RefTSChannel

IP_RefTSChannel

Direction: Down

Valid values:

Minimum: 1 = First sensor

Maximum: Up to [IP_SensorCount](#)

Description: Sensor instance, which should be used as reference timestamp.

17.9.7 Set_DriverSync

Enable sensor synchronisation at driver level. This command is only available if any sensor is connected at the same hardware device of type [IF2004_USB](#), [IF2008](#) or [IF2008_ETH](#). At end of this command [Sync_Start](#) is called automatically.

Parameter: int32_t IP_UseGate

IP_UseGate

Direction: Down

Valid values:

0 = FALSE

1 = TRUE

Default: 0

Description: This parameter uses the mechanism of [IF2004_USB Use_Gate](#), [IF2008 Use_Gate](#) or [Set_DigitalInMode2](#) resp. [Set_DigitalInMode3](#).

Parameter: int32_t IP_SyncMasterChannel IP_SyncMasterChannel

Direction: Down

Valid values:

Depending on the hardware interface, where the sensors are connected.

Description: For sensor channels, this parameter uses the mechanism of IF2004_USB [Set_SyncMaster](#), IF2008 [Set_SyncMaster](#) or IF2008_ETH [Set_SyncMaster](#).

For extended channels (Encoder, Digital In, ...), this parameter uses the mechanism of IF2004_USB [Set_ExtTriggerInSource](#), IF2008 [Set_EncoderLatchSource](#), IF2008 [Set_DigitalInLatchSource](#), IF2008 [Set_ADCLatchSource](#), IF2008_ETH [Set_EncoderLatchSource<n>](#) or IF2008_ETH [Set_DigitalInLatchSource](#).

17.9.8 Set_HardwareSync

Enable sensor synchronisation at hardware level (sync line at all sensors must be wired). This command is only available if any sensor is connected at the same hardware device of type [IF2004_USB](#), [IF2008](#) or [IF2008_ETH](#). At end of this command [Sync_Start](#) is called automatically.

Parameter: int32_t IP_UseGate IP_UseGate

Direction: Down

Valid values:

0= FALSE

1= TRUE

Default: 0

Description: This parameter uses the mechanism of IF2004_USB [Use_Gate](#), IF2008 [Use_Gate](#) or [Set_DigitalInMode2](#) resp. [Set_DigitalInMode3](#).

Parameter: int32_t IP_SyncSource IP_SyncSource

Direction: Down

Valid values:

1= Timer (on a [IF2004_USB](#), [IF2008](#) or [IF2008_ETH](#))

2= Sensor

Description: Specifies the synchronization master type.

It can be either a timer on a hardware interface. In this case, the sync inputs at the sensors must be wired to the trigger outputs of hardware interface.

Or it can be a sensor. In this case, the sync inputs of all slave sensors must be wired to the sync outputs of the master sensor.

Please read sensor documentation for detailed wiring description.

Parameter: int32_t IP_SyncMaster IP_SyncMaster

Direction: Down

Valid values:

Minimum: 0

Description: If [IP_SyncSource](#) is 1 (Timer), it specifies the timer number (counting from 1) to use for synchronization. [IF2004_USB](#) supports two timers, [IF2008](#) and [IF2008_ETH](#) supports three timers.

If [IP_SyncSource](#) is 2 (Sensor), it specifies the number of the sensor (counting from 0 up to [IP_SensorCount](#)-1) to use as synchronization master.

Parameter: double IP_SyncFrequency IP_SyncFrequency
Direction: Down
Valid values:
Minimum: 0
Maximum: 1.79769e+308 (DBL_MAX)
Default: 0
Description: If [IP_SyncSource](#) is 1 (Timer), the desired timer is setup to this frequency (which must be greater than 0). The trigger output for each channel where a sensor is connected is set to the timer. For extended channels (Encoder, Digital In, ...), the functions of IF2004_USB [Set_ExtTriggerInSource](#), IF2008 [Set_EncoderLatchSource](#), IF2008_Set_DigitalInLatchSource, IF2008_Set_ADCLatchSource, IF2008_ETH [Set_EncoderLatchSource<n>](#) or IF2008_ETH [Set_DigitalInLatchSource](#) are called to set the latch source to the timer.
 If [IP_SyncSource](#) is 2 (Sensor), the frequency may be 0. In this case, it is retrieved from the master sensor. Than the master sensor is eventually set to desired measure frequency, synchronization mode, the electrical level and termination.
 The sensor slaves are set to desired measure frequency, synchronization mode and eventually the electrical level and termination in both cases.

Parameter: int32_t IP_Resampling IP_Resampling
Direction: Down
Valid values:
Minimum: 1
Maximum: 2147483647 (INT32_MAX)
Default: 1 = no resampling
Valid for sensor:
 ILD1750
 ILD1900
 ILD2300
 IFD2421
 IFD2422
 IFD2445
 IFD2451
 IFD2461
 IFD2471
 ODC2520
 IMC5400
 IMC5600
 ACS7000
Description: Many sensors can reduce values internally (transmit each n'th value).
 At [PCI_CARD_IF2004](#), [USB_ADAPTER_IF2004](#), [PCI_CARD_IF2008](#) and [ETH_ADAPTER_IF2008](#) resampling will work too, if any other sensor channel transmits data at this device.

Parameter: int32_t SP_Sync_TrgMode SP_Sync_TrgMode
Direction: Down
Valid values:
 -1 = Use current setting of sensor, if valid (1 or 3). Otherwise it is set to 1.
 1 = synchronous master on
 3 = alternating synchronous master

Default: -1

Valid for sensor:

ILD1700 as master

Description: The sync mode or the trigger mode of the sensor.

This parameter can be specified for ILD1700 sensors used as master, see

[SP_Sync_TrgMode](#).

It only can be used as special parameter (with #).

Parameter: int32_t SP_ErrorOutput

SP_ErrorOutput

Direction: Down

Valid values:

- 1= Use current setting of sensor, if valid (0 or 1). Otherwise it is subtracted 2 of current setting.

- 0= sync error mode

- 1= sync switch mode

Default: -1

Valid for sensor:

ILD1700

Description: Sync/trigger and error/switch mode respectively of the sensor.

This parameter can be specified for ILD1700 sensors used as master or slave, see [SP_ErrorOutput](#).

It only can be used as special parameter (with #).

Parameter: int32_t SP_SyncMode

SP_SyncMode

Direction: Down

Valid values:

- 1= Use current setting of sensor, if valid (4 or 5). Otherwise it is set to 4.

- 4= Master

- 5= Master alternating (MASTER_ALT)

Default: -1

Valid for sensor:

ILD1750 as master

ILD1900 as master

Description: Synchronization mode.

This parameter can be specified for ILD1750 and ILD1900 sensors used as master, see [SP_SyncMode](#).

It only can be used as special parameter (with #).

Parameter: int32_t SP_Termination

SP_Termination

Direction: Down

Valid values:

- 1= Use current setting of sensor.

- 0= Off

- 1= On

Default: -1

Valid for sensor:

ILD1750

ILD1900

IFD2421

IFD2422

IMC5400

IMC5600

Description: Termination resistor.

This parameter can be specified for ILD1750 and ILD1900 (see [SP_Termination](#)), IFD2421 and IFD2422 (see [SP_Termination](#)), IMC5400 and IMC5600 (see [SP_Termination](#)) sensors used as master or slave.
It only can be used as special parameter (with #).

Parameter: int32_t SP_SyncMode

SP_SyncMode

Direction: Down

Valid values:

- 1= Use current setting of sensor, if valid (2 or 3). Otherwise it is set to 2.
- 2= Master
- 3= Master alternating (MASTER_ALT)

Default: -1

Valid for sensor:

ILD2300 as master

Description: Synchronization mode.

This parameter can be specified for ILD2300 sensors used as master, see [SP_SyncMode](#).
It only can be used as special parameter (with #).

Parameter: int32_t SP_SyncTermination

SP_SyncTermination

Direction: Down

Valid values:

- 1= If synchronisation mode has to be changed, set termination on (for master) or off (for slave), otherwise termination is not changed.
- 0= Off (TERMOFF)
- 1= On 120 Ohm (TERMON)

Default: -1

Valid for sensor:

ILD2300

ODC2520

Description: Termination of external input.

This parameter can be specified for ILD2300 (see [SP_SyncTermination](#)) and ODC2520 (see [SP_SyncTermination](#)) sensors used as master or slave.

It only can be used as special parameter (with #).

Parameter: int32_t SP_SyncMode

SP_SyncMode

Direction: Down

Valid values:

- 1= Use current setting of sensor, if valid (1 to 3). Otherwise it is set to 1.
- 1= Slave
- 2= Slave alternating (SLAVE_ALT)
- 3= Slave multi function input (SLAVE_MFI)

Default: -1

Valid for sensor:

ILD1750 as slave

ILD1900 as slave

Description: Synchronization mode.

This parameter can be specified for ILD1750 and ILD1900 sensors used as slave, see [SP_SyncMode](#).

It only can be used as special parameter (with #).

Parameter: int32_t SP_MultiFunctionInputLevel	SP_MultiFunctionInputLevel
Direction: Down	
Valid values:	
-1 = Use current setting of sensor.	
0 = HTL	
1 = TTL	
Default: -1	
Valid for sensor:	
ILD1750 as slave	
ILD1900 as slave	
Description: Multi function input level. It is only used if SP_SyncMode is 3 (SLAVE_MFI).	
This parameter can be specified for ILD1750 and ILD1900 sensors used as slave, see SP_MultiFunctionInputLevel .	
It only can be used as special parameter (with #).	
Parameter: int32_t SP_SyncMode	SP_SyncMode
Direction: Down	
Valid values:	
-1 = Use current setting of sensor, if valid (2 or 3). Otherwise it is set to 2.	
2 = Slave (Sync/Trig)	
3 = Slave (TrigIn)	
Valid for sensor:	
IFD2421 as slave	
IFD2422 as slave	
IMC5400 as slave	
IMC5600 as slave	
Description: Synchronization mode.	
This parameter can be specified for IFD2421 and IFD2422 sensors used as slave, see SP_SyncMode .	
It only can be used as special parameter (with #).	
Parameter: int32_t SP_TriggerInLogic	SP_TriggerInLogic
Direction: Down	
Valid values:	
-1 = Use current setting of sensor.	
0 = TTL	
1 = HTL	
Default: -1	
Valid for sensor:	
IFD2421 as slave	
IFD2422 as slave	
IMC5400 as slave	
IMC5600 as slave	
Description: Logic of trigger input. It is only used if SP_SyncMode is 3 (Slave TrigIn).	
This parameter can be specified for IFD2421 and IFD2422 sensors used as slave, see SP_TriggerInLogic .	
It only can be used as special parameter (with #).	
Parameter: int32_t SP_SyncMode	SP_SyncMode
Direction: Down	
Valid values:	
-1 = Use current setting of sensor, if valid (2 or 3). Otherwise it is set to 2.	

2= Slave
 3= Slave with external trigger (SLAVE_EXT)

Default: -1

Valid for sensor:

IFD2445 as slave
 IFD2451 as slave
 IFD2461 as slave
 IFD2471 as slave

Description: Synchronization mode.

This parameter can be specified for IFD2445, IFD2451, IFD2461 and IFD2471 sensors used as slave, see [SP_SyncMode](#).
 It only can be used as special parameter (with #).

17.9.9 Set_LatestSync

Enable latest sensor data synchronisation. This command is available for any sensor on any hardware interface. When [TransferData](#) is called, all data expect latest frame is deleted and rest is returned (but not removed after reading). At end of this command [Sync_Start](#) is called automatically.

Parameter: double IP_CycleTime

IP_CycleTime

Direction: Down

Unit: ms

Valid values:

Minimum: 0.0
Maximum: 1.79769e+308 (DBL_MAX)

Default: 2000.0

Description: To avoid high CPU load, this parameter regulates the transfer frequency. If cycle time is not reached since last [TransferData](#) than IP_LatestSyncMasterChannel will be checked.

Parameter: int32_t IP_LatestSyncMasterChannel

IP_LatestSyncMasterChan-
nel

Direction: Down

Valid values:

-1= Any sensor channel
 0= No sensor channel
 1..IP_SensorCount= Specific sensor channel

Default: -1

Description: This parameter for any (or all, if -1) channel(s) allows calling [TransferData](#) if new data has arrived. 0 disables this check.

17.9.10 Set_ManualSync

Enable manual sensor synchronisation. This command is available for any sensor on any hardware interface. The user is responsible to parametrize the sensors first, so each sensor delivers same amount of data frames. At end of this command [Sync_Start](#) is called automatically.

17.9.11 Sync_Start

Starts synchronized data acquisition. Use [DataAvail](#) and [TransferData](#) to read data. While synchronized data acquisition is active, any other sensor command (particularly for a physical sensor) would disturb this mode and data is no longer synchronized.

17.9.12 Any command for contained sensors and interfaces

It is possible to send sensor commands to one or all contained sensors or interfaces. To do this specify the command name and parameters and set additional parameters IP_SensorIndex and/or IP_InterfaceIndex.

The answer of each sensor and interface is contained, postfixed by "#S1..IP_SensorCount" (for sensor instances) or "#I1..IP_InterfaceCount" (for interface instances).

Parameter: int32_t IP_SensorIndex

IP_SensorIndex

Direction: Down

Valid values:

- 1= Send command to any contained sensor
- 0= Do not send command to sensor
- 1 .. up to [IP_SensorCount](#)= Send command to specific sensor

Default: 0

Description: Number of the sensor.

Parameter: int32_t IP_InterfaceIndex

IP_InterfaceIndex

Direction: Down

Valid values:

- 1= Send command to interfaces of any contained sensor
- 0= Do not send command to interface
- 1 .. up to [IP_SensorCount](#)= Send command to interface of specific sensor

Default: 0

Description: Number of the sensor.

Index

avail, 94
 CA_ArticleNumber, 270, 1281, 1442
 CA_Date, 270, 1281, 1442
 CA_Description, 270, 1282, 1443
 CA_FileName, 270, 1281, 1442
 CA_Finished, 270, 1282, 1443
 CA_FirmwareVersion, 269, 1281, 1442
 CA_Progress, 270, 1282, 1443
 CA_Result, 270, 1282, 1443
 CA_SerialNumber, 270, 1281, 1443
 CA_SupportedInterfaceRS232, 155
 CA_SupportedInterfaceRS422, 155
 CA_SupportedInterfaceRS485, 155
 CA_SupportedInterfaces, 154
 CA_SupportedInterfaceTCPIP, 155
 CA_SupportedInterfaceUSB, 155
 cardInstance, 108, 111
 channelNumber, 108--113
 ClearAllParameters, 90
 CloseSensor, 91
 CP_AutomaticMode, 158
 CP_BaseLevelTimeout, 132
 CP_HTTPRetryCount, 335
 CP_InitAfterSensorCommand, 164
 CP_InterruptDataTransfer, 260, 281, 326,
 383, 423, 553, 667, 780, 901,
 1102, 1271, 1366, 1404
 CP_SensorAnswerTimeout, 151, 164
 CP_ThreadPriority, 135, 149
 CP_UseCachedParameter, 1293, 1294
 CreateSensorInstance, 70
 CreateSensorInstByName, 71
 CreateSensorInstByNameU, 72

 DataAvail, 94
 deviceInstance, 109, 110, 114

 EnableLogging, 101
 enableLogging, 101, 103
 EnableLoggingU, 103
 errText, 97, 98
 ExecSCmd, 115
 ExecSCmdGetDouble, 121
 ExecSCmdGetDoubleU, 122
 ExecSCmdGetInt, 120
 ExecSCmdGetIntU, 120
 ExecSCmdGetString, 122
 ExecSCmdGetStringU, 123
 ExecSCmdU, 115

 GetDLLVersion, 100
 GetDLLVersionU, 100
 GetError, 97
 GetErrorU, 98
 GetParameterBinary, 87
 GetParameterBinaryU, 88
 GetParameterDouble, 84
 GetParameterDoubleU, 85
 GetParameterDWORD_PTR, 83
 GetParameterDWORD_PTRU, 83
 GetParameterInt, 81
 GetParameterIntU, 82
 GetParameters, 88
 GetParameterString, 85
 GetParameterStringU, 86
 GetParametersU, 89

 IA_ADC1Range, 1333
 IA_ADC2Range, 1333
 IA_AdditionalInfo1..x, 159
 IA_Baudrate, 140, 143
 IA_DataAvailEvent, 165
 IA_Datarate, 1378
 IA_Encoder1CountFactor, 1332
 IA_Encoder2CountFactor, 1332
 IA_EncoderCountFactor, 1352
 IA_Found, 178
 IA_HighResolutionTime, 164
 IA_Index1..x, 158
 IA_Interface# 1..x, 1445
 IA_MajorSignal1..x, 160
 IA_MaxValuesPerFrame, 158
 IA_Progress, 178
 IA_Range, 1028, 1327
 IA_Range1..4, 1068
 IA_Range1..8, 1057
 IA_RangeMax, 1379
 IA_RangeMin, 1379
 IA_Raw_Datarate1..x, 160
 IA_Raw_Name1..x, 159
 IA_Raw_RangeMax1..x, 159
 IA_Raw_RangeMin1..x, 159
 IA_Raw_Unit1..x, 159
 IA_ReferenceTimestamp, 162
 IA_Samplerate, 1378
 IA_Scaled_Datarate1..x, 160
 IA_Scaled_Name1..x, 159
 IA_Scaled_RangeMax1..x, 159
 IA_Scaled_RangeMin1..x, 159
 IA_Scaled_Unit1..x, 159

IA_ScanStatus, 178
IA_SensorCount, 1445
IA_SensorOpen, 155
IA_SensorType# 1..x, 1445
IA_SensorTypeName# 1..x, 1445
IA_SettingsChanged, 163
IA_Success, 193
IA_TransferMode, 161
IA_UniqueIndex0..n, 161
IA_ValuesPerFrame, 158
IA_VideoSignalName1..x, 160
IA_VideoSignalPixelByteSize1..x, 160
IA_VideoSignalPixelCount1..x, 160
IA_VideoSignalsPerFrame, 160
IA_ViewRange1..x, 159
instanceHandle, 74–98, 101, 103, 105–
 123
IP_ADC1Range, 1332
IP_ADC2Range, 1332
IP_AddressFamily, 147
IP_AllowReserved, 192
IP_AutoIPEnabled, 192
IP_AutomaticMode, 126
IP_Baudrate, 133, 137, 139, 142, 144,
 182, 183
IP_Baudrates, 177
IP_BootPEnabled, 193
IP_ByteSize, 134, 182
IP_BytSizes, 178
IP_CardInstance, 136, 141, 183
IP_ChannelNumber, 136, 138, 141, 144,
 183, 184, 188
IP_ChannelNumbers, 177
IP_ClearReceiveBuffer, 153
IP_ClearRingBuffer, 153
IP_ClearSendBuffer, 153
IP_CycleTime, 1453
IP_DataPort, 148, 167
IP_Datarate, 1378
IP_DeviceInstance, 138, 149, 183
IP_DHCPEnabled, 192
IP_EnableLogging, 129, 156
IP_EnableToggleDTR, 135
IP_EncCountMode, 1326
IP_EncInvertTraceA_B, 1327
IP_EncLatchSrc, 1327
IP_EncLoadOnRef, 1327
IP_Encoder1CountFactor, 1331
IP_Encoder2CountFactor, 1331
IP_EncoderCountFactor, 1351
IP_EncSwapTraceA_B, 1326
IP_EventChar, 135
IP_EventOnAvailableValues, 165
IP_EventOnBufferFillsize, 165
IP_FindSimilarSensors, 175
IP_FixedErrorValue, 129
IP_ForceScanUsedChannel, 176
IP_Gateway, 192
IP_Index, 179
IP_Interface, 126, 181
IP_InterfacIdx, 181
IP_InterfaceIndex, 1454
IP_KeepChannelMode, 177
IP_LatestSyncMasterChannel, 1453
IP_LogAppend, 130, 157
IP_LogFile, 130, 156
IP_LogFlush, 131, 157
IP_LogLevel, 130, 156
IP_LogSplitSize, 131, 157
IP_LogType, 129, 156
IP_MACAddress, 190
IP_MaximumFrameDiff, 1447
IP_MaximumTimeDiff, 1447
IP_MaxPacketSize, 131
IP_NewIPAddress, 191
IP_OldIPAddress, 190
IP_PacketDelay, 132
IP_Parities, 178
IP_Parity, 134, 140, 142, 182
IP_Password, 191
IP_Port, 133, 138, 181
IP_PortDetection, 148, 167
IP_QuickScanRS485, 177
IP_Range, 1027, 1326
IP_Range1..4, 1068
IP_Range1..8, 1057
IP_RangeMax, 1378
IP_RangeMin, 1378
IP_ReadTimeout, 135
IP_ReferenceTimestamp, 162
IP_RefTSChannel, 1447
IP_RemoteAddr, 144, 146, 184
IP_RemoteDataProtocol, 147, 166
IP_RemotePort, 147
IP_Resampling, 1449
IP_ResetFlag, 163
IP_RingBufferSize, 129
IP_RS232Ports, 177
IP_Samplerate, 1377
IP_ScaleErrorValues, 126
IP_ScanHWIF2004, 175
IP_ScanHWIF2004_USB, 175
IP_ScanHWIF2008, 176
IP_ScanHWIF2008_ETH, 176
IP_ScanHWRS232, 175
IP_ScanHWTCP/IP, 176

IP_ScanHWWinUSB, 176
 IP_Sensor, 1446
 IP_SensorAddress, 134, 182
 IP_SensorCount, 1444
 IP_SensorIndex, 1454
 IP_SensorType, 174, 179, 190, 1444
 IP_SensorTypeName, 180, 1444
 IP_SerialNumber, 138, 183
 IP_SleepTime, 164
 IP_StaticIP, 192
 IP_Stopbits, 134, 178, 182
 IP_SubnetMask, 192
 IP_SyncFrequency, 1449
 IP_SyncMaster, 1448
 IP_SyncMasterChannel, 137, 140, 143, 145, 1448
 IP_SyncSource, 1448
 IP_TimerResolution, 132, 464, 670, 956
 IP_TransferMode, 161
 IP_UniqueIndex0..n, 161
 IP_UsbReadBufCnt, 149
 IP_UsbReadBufSize, 149, 464, 466
 IP_UseGate, 136, 1447, 1448
 IP_UUID, 191

 len, 79, 80
 location, 106
 logAppend, 102, 104
 logFile, 102, 104
 logFlush, 102, 105
 logLevel, 105, 106
 logLevels, 101, 104
 logSplitSize, 102, 105
 LogToFile, 105
 LogToFileU, 106
 logTypes, 101, 103

 maxLen, 86--89, 97, 98, 100, 123, 124
 maxValues, 93, 95, 96
 message, 106

 OpenSensor, 90
 OpenSensorIF2004, 108
 OpenSensorIF2004_USB, 109
 OpenSensorIF2004_USBU, 110
 OpenSensorIF2008, 111
 OpenSensorIF2008_ETH, 112
 OpenSensorIF2008_ETHU, 112
 OpenSensorRS232, 107
 OpenSensorRS232U, 107
 OpenSensorTCPIP, 113
 OpenSensorTCPIPU, 114
 OpenSensorWinUSB, 114

 parameterList, 80, 81, 89
 paramName, 75--79, 82--88, 116--123
 paramValue, 75--80, 82--88, 116--123
 Poll, 93
 port, 107--110

 rawData, 93, 95, 96
 read, 95, 96
 ReleaseSensorInstance, 74
 remoteAddr, 112--114

 S_Command, 151
 SA_0_0V, 451, 477
 SA_0_10V, 451, 477
 SA_0_SOD, 451, 477
 SA_1_0V, 451, 477
 SA_1_10V, 451, 477
 SA_1_SOD, 451, 477
 SA_a*1..x, 1089, 1111
 SA_AbbeNumber_vd, 529, 570, 653, 680
 SA_AbbeNumber_vd_Ch<n>, 571
 SA_AbbeNumber_vd_Ch1..2, 529
 SA_AbbeNumber_vd1..x, 528, 568, 679
 SA_Action1..xArguments, 1185, 1233
 SA_Action1..xName, 1185, 1233
 SA_ActionArguments, 1233
 SA_ActionCompensationSettings, 1237
 SA_ActionCorrectedColor1..3, 1235
 SA_ActionDetectionDistance1..3, 1236
 SA_ActionDetectionMatcher, 1235
 SA_ActionDetectionOutputPattern, 1236
 SA_ActionEntries, 1185, 1233
 SA_ActionInputs, 1236
 SA_ActionLockEditState, 1237
 SA_ActionLockState, 1237
 SA_ActionLockViewState, 1237
 SA_ActionName, 1233
 SA_ActionRepresentationsRGB1..3, 1235
 SA.Actions, 1144
 SA_ActionSamplingAmplificationLevel, 1238
 SA_ActionSamplingAverageCount, 1238
 SA_ActionSamplingBaseSR, 1237
 SA_ActionSamplingDarkPhase, 1238
 SA_ActionSamplingEffectiveSR, 1238
 SA_ActionSamplingLEDIntensity, 1237
 SA_ActionSamplingLightPhase, 1238
 SA_ActionSamplingMinDesiredSR, 1238
 SA_ActionSensorDetectableColor1..3, 1236
 SA_ActionSensorDetectableMatcherID, 1236
 SA_ActionSensorDetectableRepresentation-sRGB1..3, 1237
 SA_ActionSensorDetectableUUID, 1236
 SA_ActionSignalLevel, 1236

SA_ActionTimestamp, 1235
 SA_ActionTransformedColor1..3, 1235
 SA_ActionTrigger1..xAPIEditlock, 1253
 SA_ActionTrigger1..xAPIViewlock, 1253
 SA_ActionTrigger1..xAutogain, 1252
 SA_ActionTrigger1..xEnableSwitchingOutpSA_ActiveParameterSet, 406, 424
 1252
 SA_ActionTrigger1..xEvent, 1251
 SA_ActionTrigger1..xKeylock, 1252
 SA_ActionTrigger1..xLockEditState, 1253
 SA_ActionTrigger1..xLockState, 1252
 SA_ActionTrigger1..xLockViewState, 1253
 SA_ActionTrigger1..xMatcherID, 1252
 SA_ActionTrigger1..xRemoveAllDetectablesSA_ActiveSensor, 506, 539, 625, 656,
 1253
 SA_ActionTrigger1..xRemoveAllMatchers, 1253
 1253
 SA_ActionTrigger1..xRemoveMatcherDetectablesBeforeSA_ActiveSensor_Ch<n>, 539
 1252
 SA_ActionTrigger1..xSource, 1251
 SA_ActionTrigger1..xTeachSingle, 1252
 SA_ActionTrigger1..xUUID, 1251
 SA_ActionTriggerAPIEditlock, 1243, 1248
 1250
 SA_ActionTriggerAPIViewlock, 1243, 1247
 1250
 SA_ActionTriggerAutogain, 1242, 1247,
 1249
 SA_ActionTriggerEnableSwitchingOutput, 1241, 1246, 1249
 SA_ActionTriggerEntries, 1251
 SA_ActionTriggerEvent, 1241, 1246, 1248
 SA_ActionTriggerKeylock, 1242, 1247,
 1249
 SA_ActionTriggerLockEditState, 1243, 1248
 1250
 SA_ActionTriggerLockState, 1242, 1247,
 1249
 SA_ActionTriggerLockViewState, 1243, 1247,
 1250
 SA_ActionTriggerMatcherID, 1242, 1246,
 1249
 SA_ActionTriggerRemoveAllDetectables,
 1242, 1247, 1250
 SA_ActionTriggerRemoveAllMatchers, 1242
 1247, 1250
 SA_ActionTriggerRemoveMatcherDetecta-
 blesBefore, 1242, 1246, 1249
 SA_ActionTriggerSource, 1241, 1246, 1248
 SA_ActionTriggerTeachSingle, 1241, 1246,
 1249
 SA_ActionTriggerUUID, 1241, 1246, 1248
 SA_ActionUUID, 1235
 SA_ActiveEdge, 455
 SA_ActiveMaterial, 408, 433, 516, 569,
 631, 679, 752, 789
 SA_ActiveMaterial_Ch<n>, 569
 SA_ActiveMaterial_Ch1..2, 516
 SA_ActiveParameterSet, 406, 424
 SA_ActiveSensor, 506, 539, 625, 656,
 748, 767
 SA_ActiveSensor_Ch<n>, 539
 SA_ActiveSensor_Ch1..2, 506
 SA_ActiveSerialIf, 848, 973
 SA_ActiveUserLevelName, 1284
 SA_ADCAvailable, 1335
 SA_ADCLatchSource, 1350
 SA_ADCRawValue, 1350
 SA_ADCScaledValue, 1351
 SA_Address, 185, 405, 421, 493, 550,
 SA_AddressFamily, 186
 SA_AlarmHysteresis, 206, 214, 222
 SA_AlarmStart, 206, 214, 221
 SA_AlarmWidth, 206, 214, 222, 223
 SA_Algorithm, 1313
 SA_AlternateMeasure, 1287
 SA_AnalogEEPROMData, 1017
 SA_AnalogFactor1..3, 1308
 SA_AnalogGain, 856, 983
 SA_AnalogMathFunctionEnable, 1308
 SA_AnalogOffset, 856, 983, 1305, 1308
 SA_AnalogOutput, 649, 725, 828, 839,
 889, 962, 1396, 1436
 SA_AnalogOutput<n>, 1437
 SA_AnalogOutput1..2, 1397
 SA_AnalogOutputSignalIndex, 506, 619,
 748, 819
 SA_AnalogOutputSignalName, 506, 620,
 748, 819
 SA_AnalogParamData, 1017
 SA_AnalogRange, 315, 372, 506, 620,
 650, 726, 748, 820, 889, 963,
 1305, 1397, 1438
 SA_AnalogRange<n>, 1439
 SA_AnalogRange1..2, 1398
 SA_AnalogScaleMode, 315, 373, 374,
 506, 621, 650, 727, 748, 820,
 890, 964, 1398, 1440
 SA_AnalogScaleMode<n>, 1441
 SA_AnalogScaleMode1..2, 1398
 SA_AnalogScaleRangeLowerLimit, 257,
 269, 316, 374, 507, 621, 749,
 821

SA_AnalogScaleRangeUpperLimit, 257, 269, 316, 375, 507, 621, 749, 821
 SA_AnalogScaleSource, 316, 375
 SA_AnalogUnit, 1307
 SA_AnalogValue, 197, 1397, 1437
 SA_AnalogValue<n>, 1438
 SA_AnalogValue1..2, 1397
 SA_AnalogZoom, 832
 SA_Answer<n>, 1408
 SA_ApplicationDisplayMode, 1008
 SA_ApplicationLanguage, 300, 327, 492, 553, 738, 781, 995, 1007, 1047, 1302, 1355, 1362, 1399, 1405
 SA_ApplicationScaleMode, 1008
 SA_ApplicationYMax, 1008
 SA_ApplicationYMin, 1008
 SA_ArticleNumber, 253, 257, 271, 275, 289, 317, 378, 396, 415, 482, 518, 623, 651, 733, 753, 861, 890, 965, 997, 999, 1032, 1033, 1052, 1054, 1070, 1090, 1264, 1268, 1289, 1290, 1293, 1309, 1311, 1353, 1354, 1380, 1400
 SA_ArticleNumber1..32, 1310
 SA_ArticleNumber1..3, 998
 SA_ArticleNumber1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536), 1053
 SA_ArticleNumber1..x, 1292
 SA_ASCII, 276, 279, 378, 380, 450
 SA_AssignLimits_ErrorOutput, 381
 SA_AutoAdaptLED, 471
 SA_AutoAdaptLEDTThr, 472
 SA_AutoDark, 454
 SA_AutoIPEnabled, 185
 SA_Automatic, 319, 329, 527, 556, 759, 783
 SA_AutostartCommand, 205, 208, 216, 225, 226, 231, 239
 SA_AutostartEdge, 208, 212, 215
 SA_AutostartTrigger, 208, 212, 215
 SA_AvailableTargets, 1000
 SA_Average, 205, 213, 218, 228, 236, 246
 SA_Average_for_reading, 857, 985
 SA_Averaging, 450, 473, 828, 839
 SA_Averaging01..28, 1270, 1279
 SA_AveragingType, 304, 345, 408, 435, 632, 685, 867, 910, 1030, 1071, 1118, 1286, 1386, 1415
 SA_AveragingType<n>, 1407
 SA_AveragingType1..2, 1387
 SA_AveragingValue, 1030, 1286
 SA_AvgRefractIndex, 469, 470
 SA_AvlIndex, 377, 379, 390, 391
 SA_AvrNbr, 1001, 1006, 1011, 1022, 1025, 1026, 1038, 1051, 1055, 1060, 1063, 1064, 1298, 1309
 SA_AvrType, 1001, 1006, 1010, 1022, 1025, 1026, 1037, 1051, 1055, 1059, 1063, 1064, 1297, 1308
 SA_AvType, 276, 278, 377, 380, 391
 SA_b*1..x, 1089, 1111
 SA_BarycenterOffset, 479
 SA_BarycenterScale, 479
 SA_Binning, 468
 SA_BootLoaderMode, 1015
 SA_BootLoaderVer, 275, 379
 SA_BootLoaderVersion, 1015
 SA_BootPEnabled, 186
 SA_BootVersion, 289, 317, 483, 518, 734, 753
 SA_BuildID, 289, 317, 483, 518, 733, 753
 SA_BuildTimestamp, 289, 317, 483, 518, 734, 753
 SA_ButtonState, 1019
 SA_CableCnt, 1000
 SA_CableHead, 289, 317
 SA_CableLength, 1004
 SA_CableOption, 1004
 SA_CableProductCode, 1004
 SA_CableRevisionIndex, 1004
 SA_CableSerialNumber, 1003, 1004
 SA_CalibProgress, 1012
 SA_CalibrationTable, 396, 415
 SA_CalibState, 1012
 SA_CalibTable, 480
 SA_CalibTarget, 1001, 1006, 1012
 SA_CCD, 463
 SA_Channel56Available, 1334
 SA_Channel1..x, 524, 537, 758, 766
 SA_Channel1..x_Ch1, 525
 SA_Channel1..x_Ch2, 526
 SA_Channel1..x_Ch<n>, 538
 SA_ChannelCount, 483, 517, 734, 752, 1264, 1266, 1291
 SA_ChannelMode<n>, 1363
 SA_ChannelMode1..8, 1355
 SA_ChartType, 301, 334, 493, 558, 738, 785
 SA_ChExist1..32, 1298, 1309
 SA_ChExist1..3, 992, 997
 SA_ChExist1..4, 1060, 1063, 1064
 SA_ChExist1..4 (DT6200/KSS64xx) or 8 (DT65x0) or 6 (DT6536), 1039,

1051
 SA_ChExist1..8, 1056
 SA_ChTransmit1..4, 1061, 1063, 1064
 SA_ChTransmit1..8, 1039, 1051, 1056
 SA_CIMode, 282
 SA_Coefficient, 1027
 SA_Color1_01..28, 1273
 SA_Color2_01..28, 1273
 SA_Color3_01..28, 1273
 SA_ColorName1..x, 1089, 1110
 SA_ColorSpace, 1073, 1110, 1115, 1267,
 1274
 SA_ColorSpace1..xAxis1..3ID, 1186, 1190
 SA_ColorSpace1..xAxis1..3Label, 1186,
 1190
 SA_ColorSpace1..xAxis1..3Max, 1186, 1190
 SA_ColorSpace1..xAxis1..3Min, 1186, 1190
 SA_ColorSpace1..xID, 1186, 1189
 SA_ColorSpace1..xName, 1185, 1189
 SA_ColorSpaceAxis1..3ID, 1188
 SA_ColorSpaceAxis1..3Label, 1188
 SA_ColorSpaceAxis1..3Max, 1189
 SA_ColorSpaceAxis1..3Min, 1189
 SA_ColorSpaceEntries, 1185, 1189
 SA_ColorSpaceID, 1188
 SA_ColorSpaceName, 1188
 SA_ColorSpaceToleranceMap1..xID, 1187
 SA_ColorSpaceToleranceMap1..xLimitAxisHalfEdge, 1054, 1293, 1311
 1187
 SA_ColorSpaceToleranceMap1..xLimitAxis
 1187
 SA_ColorSpaceToleranceMap1..xLimitAxis
 1187
 SA_ColorSpaceToleranceMap1..xShape, 1187
 SA_ColorSpaceToleranceMapEntries, 1187
 SA_ColorTable, 1088, 1110
 SA_ColorTableCount, 1088, 1110
 SA_CompleteAnswer, 151, 165
 SA_ComputationEntries, 317, 352, 520,
 592, 754, 804
 SA_ComputationEntry1..x, 318, 352, 520,
 593, 754, 804
 SA_ComputationFactor1_1..x, 521, 594,
 805
 SA_ComputationFactor2_1..x, 522, 595
 SA_ComputationID1..x, 318, 352, 520,
 593, 755, 804
 SA_ComputationMedianCount1..x, 319,
 353, 521, 594, 756, 805
 SA_ComputationMovingCount1..x, 318,
 353, 521, 594, 756, 805
 SA_ComputationName1..x, 522, 595, 756,
 805
 SA_ComputationOffset1..x, 522, 595
 SA_ComputationOperation1..x, 318, 353,
 520, 593, 755, 804
 SA_ComputationQueue1..x, 318, 352, 520,
 593, 755, 804
 SA_ComputationRecursiveCount1..x, 319,
 353, 521, 594, 756, 805
 SA_ComputationSignalIndex1_1..x, 521,
 594
 SA_ComputationSignalIndex2_1..x, 522,
 595
 SA_ComputationSignalIndex1..x, 318, 353,
 520, 593, 755, 804
 SA_ComputationSignalName1_1..x, 521,
 594
 SA_ComputationSignalName2_1..x, 522,
 595
 SA_ComputationSignalName1..x, 318, 353,
 521, 593, 755, 805
 SA_ComputationTable, 317, 352, 519,
 592, 754, 804
 SA_ConfigDescription, 1293
 SA_ContinuousMode, 195
 SA_Contrast, 848, 973
 SA_ControllerName, 999, 1002, 1033,
 SA_ControllerOption, 1002
 SA_ControllerProductCode, 1002
 SA_ControllerRevisionIndex, 1002
 SA_ControllerSerialNumber, 1002
 SA_ControllerSoftwareVersion, 1002
 SA_ControllerTemperature, 991, 992, 1011
 SA_CurrentFirmwareChannel, 1158
 SA_CurrentFirmwareCreatedOn, 1158
 SA_CurrentFirmwareID, 1158
 SA_CurrentFirmwareName, 1158
 SA_CurrentFirmwareNotes, 1159
 SA_CurrentFirmwareVersion, 1158
 SA_CurrentFirmwareWorksWith, 1158
 SA_CurrentIPAddresses, 184
 SA_CurrentName, 319, 328, 527, 555,
 759, 782
 SA_CurrentOfMonitor, 284
 SA_CurrentProfileUUID, 1204, 1208
 SA_DarkCorrAverage, 729
 SA_DarkCorrOffset1_01..28, 1269, 1277
 SA_DarkCorrOffset2_01..28, 1269, 1278
 SA_DarkCorrOffset3_01..28, 1269, 1278
 SA_DarkCorrTable, 565
 SA_DarkCorrTable_Ch<n>, 565
 SA_DarkCorrThreshold, 653, 658

SA_DarkPixel, 833, 836
 SA_DarkSig, 464
 SA_DataMode, 1320
 SA_DataOutInterface, 255, 264, 308, 355, 410, 439, 635, 691, 872, 923, 1086, 1121, 1267, 1279, 1388, 1418
 SA_DataPort, 993, 1045, 1300
 SA_DataScale, 478
 SA_DataSource, 313, 367, 648, 723, 887, 959
 SA_Date, 272, 275, 379, 823
 SA_DatOut, 382, 392
 SA_Default1..xKey, 1156
 SA_Default1..xObjectType, 1155
 SA_Default1..xUUID, 1155
 SA_Default1..xValue, 1156
 SA_DefaultEntries, 1155
 SA_DefaultKey, 1151, 1153, 1154
 SA_DefaultNTPServers, 1261
 SA_DefaultObjectType, 1151, 1153, 1154
 SA_DefaultsHoldTime, 1154
 SA_DefaultsToleranceShape, 1154
 SA_DefaultsToleranceShapeBoxLimitHalfE-
 itHalfHeight, 1155
 SA_DefaultsToleranceShapeCylinderLim-
 itRadius, 1155
 SA_DefaultsToleranceShapeSphereLimi-
 tRadius, 1154
 SA_DefaultUser, 297, 320, 403, 416, 492, 536, 625, 655, 738, 766, 863, 894, 1087, 1093
 SA_DefaultUUID, 1151, 1153, 1154
 SA_DefaultValue, 1151, 1153, 1154
 SA_Delta_A_AB, 1115
 SA_Delta_B, 1115
 SA_Delta_E_L, 1114
 SA_DeltaKC, 1073, 1109
 SA_DeltaKH, 1073, 1109
 SA_DeltaKL, 1073, 1109
 SA_DeltaMode, 1073, 1108
 SA_Described_by, 529, 570, 653, 680
 SA_Described_by_Ch<n>, 571
 SA_Described_by_Ch1..2, 530
 SA_Description, 415, 433, 528, 569, 652, 680, 760, 789, 1114
 SA_Description_Ch<n>, 571
 SA_Description_Ch1..2, 529
 SA_Description1..x, 433, 528, 568, 679, 760, 789
 SA_DetectionProfile1..xColorSpaceAxis1..3ID, 1205, 1209
 SA_DetectionProfile1..xColorSpaceAxis1..3Label, 1205, 1209
 SA_DetectionProfile1..xColorSpaceAxis1..3Max, 1206, 1210
 SA_DetectionProfile1..xColorSpaceAxis1..3Min, 1206, 1210
 SA_DetectionProfile1..xColorSpaceID, 1205, 1209
 SA_DetectionProfile1..xColorSpaceName, 1206, 1210
 SA_DetectionProfile1..xCompensationSettings, 1207, 1211
 SA_DetectionProfile1..xName, 1205, 1209
 SA_DetectionProfile1..xNonMatchingHoldTime, 1208, 1212
 SA_DetectionProfile1..xNonMatchingOutputState, 1208, 1212
 SA_DetectionProfile1..xNonMatchingOutputUUID, 1208, 1212
 SA_DetectionProfile1..xNormalizationConstant1..y, 1206, 1211
 SA_DetectionProfile1..xNormalizationConstantEntries, 1206, 1210
 SA_DetectionProfile1..xSamplingAmplificationLevel, 1208, 1212
 SA_DetectionProfile1..xSamplingAverageCount, 1208, 1212
 SA_DetectionProfile1..xSamplingBaseSR, 1207, 1211
 SA_DetectionProfile1..xSamplingDarkPhase, 1207, 1212
 SA_DetectionProfile1..xSamplingEffectiveSR, 1207, 1211
 SA_DetectionProfile1..xSamplingLEDIntensity, 1207, 1211
 SA_DetectionProfile1..xSamplingLightPhase, 1207, 1211
 SA_DetectionProfile1..xSamplingMinDesiredSR, 1207, 1211
 SA_DetectionProfile1..xUUID, 1205, 1209
 SA_DetectionProfile1..xWhiteReference1..y, 1206, 1210
 SA_DetectionProfile1..xWhiteReferenceEntries, 1206, 1210
 SA_DetectionProfileColorSpaceAxis1..3ID, 1192, 1197, 1201
 SA_DetectionProfileColorSpaceAxis1..3Label, 1192, 1197, 1201
 SA_DetectionProfileColorSpaceAxis1..3Max, 1192, 1197, 1202
 SA_DetectionProfileColorSpaceAxis1..3Min,

1192, 1197, 1202
 SA_DetectionProfileColorSpaceID, 1192, 1196, 1201
 SA_DetectionProfileColorSpaceName, 1193, 1197, 1202
 SA_DetectionProfileCompensationSettings, 1193, 1198, 1203, 1213
 SA_DetectionProfileEntries, 1204, 1209
 SA_DetectionProfileName, 1191, 1196, 1201
 SA_DetectionProfileNonMatchingHoldTime, 1195, 1200, 1204
 SA_DetectionProfileNonMatchingOutput- State, 1195, 1200, 1204
 SA_DetectionProfileNonMatchingOutputU- UID, 1195, 1200, 1204
 SA_DetectionProfileNormalizationConstant- tEntries, 1193, 1198, 1202
 SA_DetectionProfileSamplingAmplification- Level, 1195, 1200, 1204, 1215
 SA_DetectionProfileSamplingAverageCount, 1194, 1200, 1204, 1214
 SA_DetectionProfileSamplingBaseSR, 1194, 1199, 1203, 1214
 SA_DetectionProfileSamplingDarkPhase, 1194, 1199, 1203, 1214
 SA_DetectionProfileSamplingEffectiveSR, 1194, 1199, 1203, 1214
 SA_DetectionProfileSamplingLEDIntensity, 1193, 1198, 1203, 1213
 SA_DetectionProfileSamplingLightPhase, 1194, 1199, 1203, 1214
 SA_DetectionProfileSamplingMinDesiredS- 1194, 1199, 1203, 1214
 SA_DetectionProfileUUID, 1191, 1196, 1201
 SA_DetectionProfileWhiteReference1..x, 1198, 1202, 1215, 1216
 SA_DetectionProfileWhiteReferenceEntries, 1193, 1198, 1202, 1215, 1216
 SA_DeviceID, 1157
 SA_DeviceModelKey, 1157
 SA_DeviceModelName, 1157
 SA_DeviceVariant, 1157
 SA_DeviceVendor, 1157
 SA_DeviceVendorKey, 1158
 SA_DeviceVendorName, 1157
 SA_DHCPEnabled, 185, 405, 420, 493, 550, 628, 665, 739, 777, 865, 899, 994, 1046, 1074, 1100, 1300, 1354, 1360, 1383, 1403
 SA_DHCPName, 186
 SA_DHCPServer, 187
 SA_Diagnosis, 232
 SA_DiagnosticInfo, 1293
 SA_DigitalIndexData, 1018
 SA_DigitalInLatchSource, 1347, 1358, 1373
 SA_DigitalInMode1, 1358, 1373
 SA_DigitalInMode2, 1359, 1374
 SA_DigitalInMode3, 1359, 1374
 SA_DigitalInValue1..4, 1347
 SA_DigitalInValueValue, 1358, 1373
 SA_DigitalIOAvailable, 1335
 SA_DigitalLevel, 1358, 1372
 SA_DigitalOut1..4, 1348
 SA_DigitalOutBinFormat, 1073, 1104
 SA_DigitalOutColorFormat, 1073, 1103
 SA_DigitalOutDetectedID, 1085, 1137
 SA_DigitalOutSource<n>, 1375
 SA_DigitalOutSource1..4, 1359
 SA_DigitalOutValue1..4, 1349
 SA_DigitalScaleMode<n>, 1417
 SA_DigitalScaleValue<n>, 1417
 SA_DisplayGain, 856, 983
 SA_DisplayOffset, 856, 983
 SA_DisplayUnit, 292, 297, 401, 403, 491, 492, 737, 738, 864, 892, 1381, 1385
 SA_DispmesUnit, 847, 972
 SA_DistanceMode, 1074, 1129
 SA_DNSServer, 186
 SA_DoubleFrequency, 462
 SA_Echo, 296, 297, 402, 403, 492, 493, 624, 625, 737, 739, 862, 863,
 SA_Edge<n>_A, 905
 SA_Edge<n>_B, 905
 SA_Edge1..8_A, 867
 SA_Edge1..8_B, 867
 SA_EdgeCount, 835, 837
 SA_EdgeDetectThreshold, 973
 SA_EdgeFilter<n>, 917
 SA_EdgeFilter<n>Signal, 918
 SA_EnableFlash, 379, 382
 SA_Encoder3Enable, 746, 774
 SA_EncoderDirection, 1338
 SA_EncoderDirection<n>, 1370
 SA_EncoderDirection1..8, 1357
 SA_EncoderIncrements, 626, 661
 SA_EncoderInterpolation, 1337
 SA_EncoderInterpolation<n>, 547, 662, 774, 1368
 SA_EncoderInterpolation1..2, 502
 SA_EncoderInterpolation1..3, 627, 746

SA_EncoderInterpolation_{1..8}, 1357
 SA_EncoderLatchSource, 1340
 SA_EncoderLatchSource_{<n>}, 1370
 SA_EncoderLatchSource_{1..8}, 1357
 SA_Encoder.MaxValue, 627, 662
 SA_Encoder.MaxValue<_n>, 549, 664, 776
 SA_Encoder.MaxValue_{1..2}, 503
 SA_Encoder.MaxValue_{1..3}, 627, 747
 SA_Encoder.MinValue, 627, 661
 SA_EncoderMode, 1339
 SA_EncoderMode_{<n>}, 548, 663, 775, 1369
 SA_EncoderMode_{1..2}, 502, 746
 SA_EncoderMode_{1..3}, 627
 SA_EncoderMode_{1..8}, 1357
 SA_EncoderNumber, 626, 661
 SA_EncoderPreload_{<n>}, 548, 663, 775, 1369
 SA_EncoderPreload_{1..2}, 503
 SA_EncoderPreload_{1..3}, 627, 746
 SA_EncoderPreload_{1..8}, 1357
 SA_EncoderRawValue, 1342
 SA_EncoderReference_{<n>}, 1371
 SA_EncoderReference_{1..8}, 1358
 SA_EncoderScaledValue, 1342
 SA_EncoderValue_{<n>}, 1371
 SA_EncoderValue_{1..8}, 1358
 SA_Energy, 198
 SA_Error, 1002
 SA_ErrorHandler, 272, 275, 278, 377, 379, 391, 847, 972
 SA_ErrorHysteresis, 314, 369
 SA_ErrorHysteresis_{<n>}, 266
 SA_ErrorHysteresis_{1..3}, 256
 SA_ErrorLevel, 649, 724
 SA_ErrorLevelOut1, 314, 370
 SA_ErrorLevelOut2, 315, 371
 SA_ErrorLevelOut_{<n>}, 266, 619, 818, 961
 SA_ErrorLevelOut_{1..2}, 502, 746, 888
 SA_ErrorLevelOut_{1..3}, 256
 SA_ErrorLimitCompOp_{<n>}, 267, 369, 618, 817
 SA_ErrorLimitCompOp_{1..2}, 314, 502, 746
 SA_ErrorLimitCompOp_{1..3}, 256
 SA_ErrorLimitSignalIndex_{<n>}, 617, 816
 SA_ErrorLimitSignalIndex_{1..2}, 501, 745
 SA_ErrorLimitSignalName_{<n>}, 617, 816
 SA_ErrorLimitSignalName_{1..2}, 502, 745
 SA_ErrorMode, 206, 214, 221, 229, 237, 244
 SA_ErrorNumber, 151
 SA_ErrorOut1HoldTime, 888
 SA_ErrorOut2HoldTime, 888
 SA_ErrorOut<_n>HoldTime, 960
 SA_ErrorOutHoldTime, 257, 268, 314, 370
 SA_ErrorOutput, 377, 380
 SA_ErrorOutput1, 313, 366
 SA_ErrorOutput2, 313, 367
 SA_ErrorOutput_{<n>}, 265, 616, 722, 957, 1434
 SA_ErrorOutput_{1..2}, 501, 648, 887, 1390
 SA_ErrorOutput_{1..3}, 256
 SA_ErrorStatus, 198
 SA_ErrorText, 152
 SA_EthernetMode, 404, 422, 493, 552, 628, 667, 739, 779, 864, 900, 1047, 1074, 1102, 1301
 SA_EvalBegin, 827, 838
 SA_EvalEnd, 827, 838
 SA_EvalMode, 827, 838
 SA_ExpectedEdges, 866, 905
 SA_ExportData, 330, 557, 784
 SA_Exposure, 461, 462
 SA_ExposureLimit, 303, 341
 SA_ExposureMode, 303, 341
 SA_Ext_LaserSwitch, 847, 973
 SA_ExtInputMode, 277, 279
 SA_ExtTriggerInDirection_{1..4}, 1323
 SA_ExtTriggerInSource, 1322
 SA_ExtTriggerInValue_{1..4}, 1323
 SA_ExtTriggerOutSource_{1..2}, 1324
 SA_FactorCapa, 1062
 SA_FactorCh_{1..4} (DT6200) or 8 (DT6530) or 6 (DT6536), 1043
 SA_FactorEddy, 1062
 SA_FactoryDefault1..xKey, 1156
 SA_FactoryDefault1..xObjectType, 1156
 SA_FactoryDefault1..xUUID, 1156
 SA_FactoryDefault1..xValue, 1156
 SA_FactoryDefaultEntries, 1156
 SA_Firmware1..xBuildID, 1161
 SA_Firmware1..xExpectedSize, 1161
 SA_Firmware1..xMaxChunkSize, 1162
 SA_Firmware1..xStatus, 1161
 SA_Firmware1..xUploadedSize, 1161
 SA_Firmware1..xUUID, 1161
 SA_FirmwareBuildID, 1160, 1163
 SA_FirmwareEntries, 1161
 SA_FirmwareExpectedSize, 1160
 SA_FirmwareMaxChunkSize, 1161
 SA_FirmwareReleaseChannel, 1162, 1163
 SA_FirmwareSourceURL, 1163
 SA_FirmwareStatus, 1160
 SA_FirmwareUploadedSize, 1160

SA_FirmwareUUID, 1160
 SA_FirmwareVersion, 1016, 1163
 SA_FirstPeak, 460
 SA_FirstReference, 1343
 SA_FPGAVersion, 1315, 1328, 1334, 1353, 1354, 1380, 1400
 SA_FrameCount, 1287
 SA_FramesPerPacket_ETH, 600, 694, 740, SA_Keylock, 276, 279, 295, 297, 378, 809, 1355, 1361, 1390, 1420
 SA_FreePara1, 835
 SA_FreeSR, 461
 SA_FrontEdge_Seg1, 858, 985
 SA_FrontEdge_Seg2, 858, 985
 SA_FrontEdge_Seg3, 986
 SA_FrontEdge_Seg4, 986
 SA_Gain01..28, 1270, 1275
 SA_GainPoti, 1014
 SA_Gate, 1330
 SA_Gateway, 185, 405, 421, 494, 550, 628, 665, 739, 777, 865, 899, 995, 1046, 1075, 1100, 1301, 1355, 1360, 1384, 1403
 SA_GroupIndex, 760, 789
 SA_GroupIndex1..x, 760, 788
 SA_HalfMinDist, 728
 SA_HardwareRevision, 289, 317, 483, 518, 624, 651, 733, 753, 862, 891, 1264, 1268
 SA_HelpText, 252, 288, 395, 482, 623, 733, 861, 1070, 1263
 SA_HighFrequency, 463
 SA_HoldLastValid, 255, 264, 309, 358, 410, 440, 516, 599, 636, 693, 752, 809, 873, 924, 1086, 1123, 1390, 1420
 SA_Hostname, 1260
 SA_HTTPBody, 335
 SA_HTTPHeader, 335
 SA_HWMode, 829, 840
 SA_HWRevision, 1315, 1334
 SA_HysteresisQ1, 196
 SA_HysteresisQ2, 196
 SA_IgnoreValues<n>, 917
 SA_Image, 862, 891
 SA_ImageType, 396, 415, 624, 651, 862, 891, 1070, 1090
 SA_IndexData, 1018
 SA_InitialChartType, 301, 333
 SA_InitialName, 319, 329, 527, 556, 759, 783
 SA_InstanceHandle, 1408, 1446
 SA_InState, 836
 SA_IntegrationTime01..28, 1270, 1275
 SA_InternalTriggerMode, 833
 SA_InvalidValues, 207, 215, 220
 SA_IsFirstRef, 1330
 SA_IsSecondRef, 1330
 SA_KeyFunction, 296, 298
 SA_KeyFunction1, 494, 533, 741, 763
 SA_KeyFunction2, 495, 534, 741, 764
 SA_Keylock, 276, 279, 295, 297, 378, 382, 392, 496, 532, 741, 762, 1075, 1092
 SA_KeylockState, 295, 298, 496, 532, 741, 762, 1075, 1092
 SA_KeylockTime, 295, 298, 496, 532, 741, 762, 1075, 1092
 SA_KeypadEvent1..xEvent, 1174
 SA_KeypadEvent1..xName, 1174
 SA_KeypadEvent1..xSource, 1174
 SA_KeypadEvent1..xTimestamp, 1174
 SA_KeypadEventEntries, 1173
 SA_KeypadInput1..xCapability1..yName, 1175
 SA_KeypadInput1..xCapability1..yURL, 1175
 SA_KeypadInput1..xCapabilityEntries, 1175
 SA_KeypadInput1..xName, 1174
 SA_KeypadInputEntries, 1174
 SA_KeypadLocked, 1172, 1173
 SA_KeypadOutput1..xAnimation1..yColor, 1182
 SA_KeypadOutput1..xAnimation1..yDuration, 1182
 SA_KeypadOutput1..xAnimation1..yEnabled, 1182
 SA_KeypadOutput1..xAnimationEntries, 1182
 SA_KeypadOutput1..xName, 1182
 SA_KeypadOutput1..xType, 1182
 SA_KeypadOutputAnimation1..xColor, 1181
 SA_KeypadOutputAnimation1..xDuration, 1181
 SA_KeypadOutputAnimation1..xEnabled, 1181
 SA_KeypadOutputAnimationEntries, 1181
 SA_KeypadOutputEntries, 1181
 SA_KeypadOutputName, 1180
 SA_KeypadOutputType, 1181
 SA_KeypadVisualizationURL, 1172, 1173
 SA_L*1..x, 1089, 1111
 SA_LampTest, 457
 SA_LampTestThr, 458
 SA_Language, 847, 972
 SA_LaserDiodeError, 283
 SA_LaserIntensity, 848, 973
 SA_LaserPower, 255, 263, 304, 342, 407, 430, 867, 906

SA_LaserPower<n>, 1366, 1406
 SA_LaserPower1..2, 1384
 SA_LaserPower1..8, 1356
 SA_LaserState, 382, 392
 SA_LastValid, 474
 SA_LatchReg, 1329
 SA_lc0, 729
 SA_lc1, 729
 SA_lc2, 729
 SA_LED_Off, 472
 SA_LEDControl, 505, 541, 657, 1075, 1095
 SA_LEDControl_Ch<n>, 542
 SA_LEDControl_Ch1..2, 505
 SA_LEDIntensity, 471
 SA_LEDIntensityColdWhite, 1075, 1096
 SA_LEDIntensityGreen, 1075, 1096
 SA_LEDIntensityViolet, 1076, 1097
 SA_LEDIntensityWarmWhite, 1076, 1097
 SA_LEDMode1, 1325
 SA_LEDMode2, 1325
 SA_LEDMode3, 1325
 SA_LEDMode4, 1325
 SA_LeftEdge, 834, 837
 SA_LightCorrTable, 565
 SA_LightCorrTable_Ch<n>, 565
 SA_LightSource, 1076, 1095
 SA_LightSource1..x, 1089, 1110
 SA_LimitQ1-1, 195
 SA_LimitQ1-2, 196
 SA_LimitQ2-1, 196
 SA_LimitQ2-2, 196
 SA_LimitQA-1, 197
 SA_LimitQA-2, 197
 SA_LinMode, 1023, 1025, 1026
 SA_LinMode1..4 (DT6200) or 8 (DT65x0) or 6 (DT6536), 1048
 SA_LinMode1..8, 1051, 1056
 SA_LinPoint, 1024, 1050, 1067
 SA_LinPoti, 1014
 SA_LocalAdapterDescription, 188
 SA_LocalDefaultGateway, 187
 SA_LocalDHCPEnabled, 187
 SA_LocalIPAddress, 186
 SA_LocalSubnetMask, 187
 SA_Lower_hysteresis, 381
 SA_Lower_limit, 381
 SA_LowerBound<n>, 917
 SA_LowerLimit, 649, 724, 857, 888, 960, 984
 SA_LowerLimit<n>, 267, 368, 617, 817, 1435
 SA_LowerLimit1..2, 313, 502, 745, 1391
 SA_LowerLimit1..3, 256
 SA_LowerWarning, 857, 984
 SA_LowFrequency, 463
 SA_LowPass, 1038
 SA_MACAddress, 184
 SA_MacAddress, 396, 415, 483, 518, 623, 651, 733, 753, 862, 891, 1070, 1090, 1353, 1354, 1380, 1400
 SA_Manufacturer, 184
 SA_Master, 305, 347, 409, 438, 634, 691, 871, 922, 1388, 1416
 SA_Master_MidPoint_Setup, 381
 SA_Master_value, 381
 SA_MasterAction1..x, 307, 350, 589, 800
 SA_MasterDistance1..6_Ch1, 495, 535
 SA_MasterDistance1..6_Ch2, 495, 535
 SA_MasterEntries, 307, 350, 588, 799
 SA_MasterEntry1..x, 307, 350, 588, 800
 SA_MasterIndex1..x, 307, 350, 589, 800
 SA_MasterName1..x, 589, 800
 SA_MasterPeak1..16_Ch1, 742, 764
 SA_MasterSignal, 634, 690, 871, 920
 SA_MasterSignalEntries, 306, 348, 522, 586, 756, 797
 SA_MasterSignalEntry1..x, 306, 348, 523, 586, 756, 797
 SA_MasterSignallIndex1..x, 306, 348, 523, 586, 756, 797
 SA_MasterSignalName1..x, 523, 586, 757, 797
 SA_MasterSignalNames, 496, 535, 742, 765
 SA_MasterSignalTable, 306, 348, 522, 586, 756, 797
 SA_MasterSource, 307, 349
 SA_MasterTable, 307, 350, 588, 799
 SA_MasterValue, 306, 347, 410, 438, 635, 691, 858, 872, 922, 987, 1388, 1416
 SA_MasterValue1..x, 306, 348, 523, 586, 757, 797
 SA_Matcher1..xHoldTime, 1226
 SA_Matcher1..xName, 1225
 SA_Matcher1..xOutputPattern, 1226
 SA_Matcher1..xResetOutputAfterHoldTimeExpired, 1226
 SA_Matcher1..xToleranceShape, 1225
 SA_Matcher1..xToleranceShapeBoxLimitHalfEdge1..3, 1226
 SA_Matcher1..xToleranceShapeCylinderLimitHalfHeight, 1226
 SA_Matcher1..xToleranceShapeCylinderLimitRadius,

1226
 SA_Matcher1..xToleranceShapeSphereLimitRadius, 678, 759, 788
 1225
 SA_Matcher1..xUUID, 1225
 SA_MatcherDetectableCount, 1225
 SA_MatcherEntries, 1225
 SA_MatcherHoldTime, 1219, 1222, 1224
 SA_MatcherIDList, 1226
 SA_MatcherName, 1218, 1221, 1223
 SA_MatcherOutputPattern, 1219, 1222, 1224
 SA_MatcherResetOutputAfterHoldTimeExpired, 1219, 1222, 1224
 SA_MatcherToleranceShape, 1218, 1221, 1223
 SA_MatcherToleranceShapeBoxLimitHalfHeight, 1219, 1222, 1224
 SA_MatcherToleranceShapeCylinderLimitHalfHeight, 1219, 1222, 1224
 SA_MatcherToleranceShapeCylinderLimitRadius, 1219, 1222, 1224
 SA_MatcherToleranceShapeSphereLimitRadius, 1218, 1222, 1223
 SA_MatcherUUID, 1218, 1221, 1223, 1227
 SA_MaterialMultiPeak12, 516, 575, 631, 683, 747, 791
 SA_MaterialMultiPeak12_Ch<n>, 577
 SA_MaterialMultiPeak12_Ch1..2, 516
 SA_MaterialMultiPeak23, 516, 576, 631, 683, 747, 792
 SA_MaterialMultiPeak23_Ch<n>, 577
 SA_MaterialMultiPeak23_Ch1..2, 516
 SA_MaterialMultiPeak34, 516, 576, 631, 684, 747, 792
 SA_MaterialMultiPeak34_Ch<n>, 577
 SA_MaterialMultiPeak34_Ch1..2, 516
 SA_MaterialMultiPeak45, 516, 576, 631, 684, 747, 792
 SA_MaterialMultiPeak45_Ch<n>, 577
 SA_MaterialMultiPeak45_Ch1..2, 517
 SA_MaterialMultiPeak56, 516, 576, 632, 684, 747, 792
 SA_MaterialMultiPeak56_Ch<n>, 577
 SA_MaterialMultiPeak56_Ch1..2, 517
 SA_MaterialName, 415, 433, 528, 569, 652, 679, 760, 789
 SA_MaterialName_Ch<n>, 570
 SA_MaterialName_Ch1..2, 529
 SA_MaterialName1..x, 432, 527, 568, 678, 760, 788
 SA_MaterialTable, 432, 527, 567, 678, 759, 788
 SA_MaterialTableCount, 432, 527, 567,

SA_MaximumDetectablesCount, 1183
 SA_MaximumMatchersCount, 1183
 SA_MaximumSamplerate, 1183
 SA_MaxRaw, 859, 987, 988
 SA_MaxRefractIndex, 469, 470
 SA_MaxScaled, 859, 988
 SA_MaxValue, 315, 373, 650, 727, 836, 890, 965, 1398, 1440
 SA_MaxValue<n>, 1442
 SA_MaxValue1..2, 1399
 SA_MeasDistance, 891, 895
 SA_MeasDistance1..x, 894
 SA_MeasDistIndex, 891, 895
 SA_MeasDistIndex1..x, 894
 SA_MeasDistTable, 891, 894
 SA_MeasDistTableCount, 894
 SA_MeasFrequency, 228, 236, 240
 SA_MeasMode, 984
 SA_MeasObject, 858, 985
 SA_MeasProgName, 856, 983
 SA_MeasProgNumber, 847, 856, 972, 983
 SA_Measrate, 303, 339, 340, 407, 428, 493, 563, 630, 672, 739, 786, 1029, 1076, 1108, 1267, 1272, 1284, 1385, 1414
 SA_MeasureCount, 1001, 1006, 1009
 SA_MeasureDirection, 866, 904
 SA_MeasureMode, 301, 336, 406, 425, 450, 459, 629, 670, 866, 903, 1000, 1005, 1010, 1076, 1106, 1384, 1413
 SA_MeasurePeak, 302, 336, 406, 426, 497, 559, 634, 671
 SA_MeasurePeak_Ch<n>, 559
 SA_MeasurePeak_Ch1..2, 498
 SA_MeasurePeak<n>, 1409
 SA_MeasurePeakSort, 748, 786
 SA_MeasureTime, 206, 210, 213
 SA_MeasureValue, 992
 SA_MeasureValueCnt, 410, 441
 SA_MeasValue, 834, 836
 SA_Median, 985
 SA_Median_OnOff, 272
 SA_MedianCount, 305, 346, 409, 436, 632, 685, 868, 910, 1072, 1118, 1387, 1415
 SA_MedianCount<n>, 1408
 SA_MedianCount1..2, 1387
 SA_MedianIndex, 276, 278
 SA_MinGap, 728

SA_MinimumThreshold, 504, 577, 747, 792
 SA_MinimumThreshold_Ch<n>, 578
 SA_MinimumThreshold_Ch1..2, 504
 SA_MinRaw, 858, 859, 987, 988
 SA_MinRefractIndex, 469, 470
 SA_MinScaled, 859, 988
 SA_MinSR, 462
 SA_MinSRIIndex, 453
 SA_MinValue, 315, 373, 650, 727, 836, 890, 964, 1398, 1440
 SA_MinValue<n>, 1442
 SA_MinValue1..2, 1399
 SA_ModeQ1, 195
 SA_ModeQ2, 196
 SA_MovingCount, 276, 278, 305, 346, 408, 435, 632, 685, 868, 910, 1072, 1118, 1386, 1415
 SA_MovingCount<n>, 1407
 SA_MovingCount1..2, 1387
 SA_MultiFunctionInputLevel, 301, 332
 SA_MultiFunctionInputLogic, 1388, 1413
 SA_MultiFunctionInputMode, 300, 332, 1388, 1412
 SA_MV-First-8, 835
 SA_MV-Last-8, 835
 SA_Name, 519, 540, 651, 657, 753, 768
 SA_Name_Ch<n>, 540
 SA_Name_Ch1..2, 519
 SA_Name1..x, 525, 537, 652, 656, 758, 767
 SA_Name1..x_Ch1, 525
 SA_Name1..x_Ch2, 526
 SA_Name1..x_Ch<n>, 538
 SA_NbrCorrectedValues, 409, 437, 498, 580, 633, 687, 869, 911
 SA_NbrCorrectedValues_Ch<n>, 582
 SA_NbrCorrectedValues_Ch1..2, 499
 SA_NbrEvaluatedValues, 409, 437, 498, 580, 633, 686, 868, 911
 SA_NbrEvaluatedValues_Ch<n>, 582
 SA_NbrEvaluatedValues_Ch1..2, 499
 SA_NegotiatedBaudrate, 1288
 SA_NetworkBits, 187
 SA_NetworkInterface1..xHardwareAddress, 1170
 SA_NetworkInterface1..xHasLink, 1170
 SA_NetworkInterface1..xIPv4Config1..yAddress, 1171
 SA_NetworkInterface1..xIPv4Config1..yGateway, 1171
 SA_NetworkInterface1..xIPv4Config1..yMethod, 1171
 SA_NetworkInterface1..xIPv4ConfigEntries, 1171
 SA_NetworkInterface1..xIPv4Current1..yAddress, 1171
 SA_NetworkInterface1..xIPv4CurrentEntries, 1170
 SA_NetworkInterface1..xIPv6Config1..yAddress, 1172
 SA_NetworkInterface1..xIPv6Config1..yGateway, 1172
 SA_NetworkInterface1..xIPv6ConfigEntries, 1171
 SA_NetworkInterface1..xIPv6Current1..yAddress, 1171
 SA_NetworkInterface1..xIPv6CurrentEntries, 1171
 SA_NetworkInterface1..xName, 1170
 SA_NetworkInterfaceEntries, 1170
 SA_NetworkInterfaceHardwareAddress, 1166, 1168
 SA_NetworkInterfaceHasLink, 1166, 1168
 SA_NetworkInterfaceIPv4Config1..xAddress, 1167, 1169
 SA_NetworkInterfaceIPv4Config1..xGateway, 1167, 1169
 SA_NetworkInterfaceIPv4Config1..xMethod, 1166, 1169
 SA_NetworkInterfaceIPv4ConfigEntries, 1166, 1169
 SA_NetworkInterfaceIPv4Current1..xAddress, 1166, 1169
 SA_NetworkInterfaceIPv4CurrentEntries, 1166, 1168
 SA_NetworkInterfaceIPv6Config1..xAddress, 1167, 1170
 SA_NetworkInterfaceIPv6Config1..xGateway, 1167, 1170
 SA_NetworkInterfaceIPv6Config1..xMethod, 1167, 1170
 SA_NetworkInterfaceIPv6ConfigEntries, 1167, 1169
 SA_NetworkInterfaceIPv6Current1..xAddress, 1167, 1169
 SA_NetworkInterfaceIPv6CurrentEntries, 1167, 1169
 SA_NetworkInterfaceName, 1166, 1168
 SA_NormQ1, 196
 SA_NormQ2, 197
 SA_NormQA, 197
 SA_NormSig, 480
 SA_Now, 1261

SA_NTPServers, 1260, 1261
 SA_NullPoti, 1014
 SA_NumberOfPeaks, 517, 574, 629, 682, 752, 791
 SA_NumberOfPeaks_Ch<n>, 575
 SA_NumberOfPeaks_Ch1..2, 517
 SA_NumberOfSegments, 985
 SA_Observer, 1077, 1094
 SA_Observer1..x, 1089, 1110
 SA_Offset, 197, 208, 216, 217, 229, 237, 241, 242, 997, 1033, 1042, 1053, 1062, 1290, 1291, 1310
 SA_Offset1..32, 1311
 SA_Offset1..3, 998
 SA_Offset1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536), 1054
 SA_Offset1..x, 1292
 SA_OffsetCorrection, 1044
 SA_OffsetValue, 391
 SA_OperationMode, 828, 840
 SA_Option, 253, 257, 271, 275, 289, 316, 378, 390, 396, 415, 474, 482, 518, 623, 650, 733, 753, 841, 861, 890, 966, 999, 1033, 1054, 1070, 1090, 1264, 1268, 1293, 1311, 1316, 1353, 1354, 1380, 1399
 SA_OutMode, 833
 SA_Output_Analog, 390
 SA_Output_Digital, 390
 SA_OutputAdditional, 1390, 1433
 SA_OutputAdditionalAmount_Ch1_ETH, 758
 SA_OutputAdditionalAmount_Ch1_RS422, 750, 811
 SA_OutputAdditionalCounter_ETH, 398, 413, 447, 491, 515, 615, 646, 719, 736, 751, 815, 886, 955
 SA_OutputAdditionalCounter_RS422, 290, 311, 361, 364, 397, 412, 445, 487, 511, 607, 645, 716, 735, 750, 812, 885, 953
 SA_OutputAdditionalDistanceRaw1_RS422, 291, 310
 SA_OutputAdditionalDistanceRaw_RS422, 312, 362, 365
 SA_OutputAdditionalEncoder1_Ch1_ETH, 489, 513, 613
 SA_OutputAdditionalEncoder1_Ch1_RS422, 485, 509, 606
 SA_OutputAdditionalEncoder1_Ch2_ETH, 489, 514, 614
 SA_OutputAdditionalEncoder1_Ch2_RS422, 485, 509
 SA_OutputAdditionalEncoder2_Ch1_ETH, 489, 513, 613
 SA_OutputAdditionalEncoder2_Ch1_RS422, 485, 509, 606
 SA_OutputAdditionalEncoder2_Ch2_ETH, 490, 514, 614
 SA_OutputAdditionalEncoder2_Ch2_RS422, 486, 510, 606
 SA_OutputAdditionalEncoder2_ETH, 646, 718
 SA_OutputAdditionalEncoder2_RS422, 645, 715
 SA_OutputAdditionalIntensity1_ETH, 399, 413
 SA_OutputAdditionalIntensity1_RS422, 291, 310, 397, 411
 SA_OutputAdditionalIntensity2_ETH, 399, 413
 SA_OutputAdditionalIntensity2_RS422, 397, 411
 SA_OutputAdditionalIntensity_Ch1_ETH, 513, 613
 SA_OutputAdditionalIntensity_Ch1_RS422, 509, 606
 SA_OutputAdditionalIntensity_Ch2_ETH, 514, 614
 SA_OutputAdditionalIntensity_Ch2_RS422, 510, 607
 SA_OutputAdditionalIntensity_ETH, 414, 447, 647, 719
 SA_OutputAdditionalIntensity_RS422, 291, 310, 312, 362, 365, 412, 446, 645, 716
 SA_OutputAdditionalIntensity1..6_Ch1_ETH, 489, 514
 SA_OutputAdditionalIntensity1..6_Ch1_RS422, 485, 509
 SA_OutputAdditionalIntensity1..6_Ch2_ETH, 490, 514
 SA_OutputAdditionalIntensity1..6_Ch2_RS422, 485, 509

486, 510
 SA_OutputAdditionalMeasrate_ETH, 490, SA_OutputAdditionalTimestampLo_ETH,
 515, 614, 647, 719, 736, 751, 90, 515, 615
 814
 SA_OutputAdditionalTimestampLo_RS422,
 SA_OutputAdditionalMeasrate_RS422, 290, 291, 311, 361, 487, 511, 607
 311, 361, 486, 510, 607, 646, SA_OutputAdditionalTrgCounter_ETH, 399,
 716, 735, 750, 811 414, 448
 SA_OutputAdditionalNbrEdges_ETH, 885, SA_OutputAdditionalTrgTimeDiff_ETH, 490,
 954 515, 614, 647, 719
 SA_OutputAdditionalNbrEdges_RS422, 885, SA_OutputAdditionalTrgTimeDiff_RS422,
 953 486, 510, 607, 646, 716
 SA_OutputAdditionalNbrGaps_ETH, 886, SA_OutputAmount1..16_Ch1_ETH, 736,
 955 751
 SA_OutputAdditionalNbrGaps_RS422, 885, SA_OutputAmount1..16_Ch1_RS422, 734,
 953 749
 SA_OutputAdditionalNbrPins_ETH, 886, SA_OutputAnalog, 308, 355, 496, 597,
 955 742, 807
 SA_OutputAdditionalNbrPins_RS422, 885, SA_OutputC-BoxAdditional_ETH, 1396, 1432
 953
 SA_OutputC-BoxAdditional_RS422, 1392,
 SA_OutputAdditionalShutterTime_Ch1_ETH, 1424
 489, 513, 613, 736, 751, 814
 SA_OutputC-BoxAdditional_USB, 1394, 1428
 SA_OutputAdditionalShutterTime_Ch1_RS422, SA_OutputC-BoxCounter_ETH, 1396, 1433
 485, 509, 606, 734, 749, 811
 SA_OutputC-BoxCounter_USB, 1394, 1428
 SA_OutputAdditionalShutterTime_Ch2_ETH, SA_OutputC-BoxDigital_ETH, 1396, 1433
 489, 514, 614
 SA_OutputC-BoxDigital_USB, 1394, 1428
 SA_OutputAdditionalShutterTime_Ch2_RS422, SA_OutputC-BoxTimestamp_ETH, 1396,
 486, 509, 606 1433
 SA_OutputAdditionalShutterTime_ETH, 398, SA_OutputC-BoxTimestamp_USB, 1394,
 413, 447, 646, 718 1428
 SA_OutputAdditionalShutterTime_RS422, SA_OutputC-BoxValue_ETH, 1396, 1432
 290, 310, 361, 364, 396, 412, SA_OutputC-BoxValue_RS422, 1392, 1423
 445, 645, 715
 SA_OutputC-BoxValue_USB, 1394, 1428
 SA_OutputAdditionalState_ETH, 399, 414, SA_OutputCalculation_ETH, 491, 515, 615,
 448, 647, 719, 736, 752, 815, 737, 752, 815
 886, 955
 SA_OutputCalculation_RS422, 487, 511,
 SA_OutputAdditionalState_RS422, 292, 312, 607, 735, 750, 812
 362, 365, 397, 413, 446, 646, SA_OutputCh01..28, 1267, 1280
 716, 735, 750, 812, 885, 953
 SA_OutputCh01..28Color1, 1265, 1268
 SA_OutputAdditionalTemperature_ETH, 398, SA_OutputCh01..28Color2, 1265, 1268
 414, 448
 SA_OutputCh01..28Color3, 1265, 1268
 SA_OutputAdditionalTemperature_RS422, SA_OutputCh01..28Temperature, 1265,
 397, 413, 446 1269
 SA_OutputAdditionalTimestamp_ETH, 398, SA_OutputCh01..28Timestamp, 1266, 1269
 414, 447, 490, 515, 614, 647, SA_OutputCh01..28Wavelength, 1266, 1269
 719, 736, 751, 815, 886, 955
 SA_OutputColorLAB99_ETH, 1079, 1126
 SA_OutputAdditionalTimestamp_RS422, SA_OutputColorLAB99_RS422, 1084, 1128
 253, 256, 265, 311, 362, 364, SA_OutputColorLAB_ETH, 1079, 1126
 397, 412, 445, 486, 510, 607, SA_OutputColorLAB_RS422, 1084, 1128
 645, 716, 735, 750, 812, 885, SA_OutputColorLCH99_ETH, 1080, 1126
 953
 SA_OutputColorLCH99_RS422, 1085, 1128
 SA_OutputAdditionalTimestampHi_ETH, 498, SA_OutputColorLCH_ETH, 1079, 1126
 515, 615
 SA_OutputColorLCH_RS422, 1084, 1128
 SA_OutputAdditionalTimestampHi_RS422, SA_OutputColorLUV_ETH, 1079, 1126

SA_OutputColorLUV_RS422, 1084, 1128
 SA_OutputColorRGB_ETH, 1079, 1125
 SA_OutputColorRGB_RS422, 1084, 1127
 SA_OutputColorXYZ_ETH, 1079, 1125
 SA_OutputColorXYZ_RS422, 1084, 1127
 SA_OutputContent, 231, 238, 242, 243
 SA_OutputCount, 1175, 1176
 SA_OutputDiameterCenterAxis_ETH, 874, 929
 SA_OutputDiameterCenterAxis_RS422, 874, 928
 SA_OutputDiameterDifference_ETH, 874, 929
 SA_OutputDiameterDifference_RS422, 873, 928
 SA_OutputDiameterEdgeA_ETH, 874, 928
 SA_OutputDiameterEdgeA_RS422, 873, 927
 SA_OutputDiameterEdgeB_ETH, 874, 929
 SA_OutputDiameterEdgeB_RS422, 873, 927
 SA_OutputDistance1_ETH, 399, 413, 637, 698
 SA_OutputDistance1_RS422, 253, 256, 292, 310, 361, 397, 411, 441, 636, 695
 SA_OutputDistance2_ETH, 399, 413, 637, 698
 SA_OutputDistance2_RS422, 397, 411, 442, 636, 695
 SA_OutputDistance3_ETH, 637, 698
 SA_OutputDistance3_RS422, 636, 696
 SA_OutputDistance4_ETH, 637, 698
 SA_OutputDistance4_RS422, 636, 696
 SA_OutputDistance5_ETH, 638, 699
 SA_OutputDistance5_RS422, 636, 696
 SA_OutputDistance6_ETH, 638, 699
 SA_OutputDistance6_RS422, 637, 696
 SA_OutputDistance1..6_Ch1_ETH, 488, 512, 612
 SA_OutputDistance1..6_Ch1_RS422, 484, 508, 605
 SA_OutputDistance1..6_Ch2_ETH, 488, 512, 613
 SA_OutputDistance1..6_Ch2_RS422, 485, 508, 605
 SA_OutputDistanceUnlinearized_RS422, 292, 312, 362
 SA_OutputDistDetectedID_ETH, 1082, 1134
 SA_OutputDistDetectedID_RS422, 1085, 1135
 SA_OutputDistDistance01_ETH, 1080, 1131
 SA_OutputDistDistance02_ETH, 1080, 1132
 SA_OutputDistDistance03_ETH, 1080, 1132
 SA_OutputDistDistance04_ETH, 1080, 1132
 SA_OutputDistDistance05_ETH, 1080, 1132
 SA_OutputDistDistance06_ETH, 1080, 1132
 SA_OutputDistDistance07_ETH, 1080, 1132
 SA_OutputDistDistance08_ETH, 1081, 1132
 SA_OutputDistDistance09_ETH, 1081, 1132
 SA_OutputDistDistance10_ETH, 1081, 1133
 SA_OutputDistDistance11_ETH, 1081, 1133
 SA_OutputDistDistance12_ETH, 1081, 1133
 SA_OutputDistDistance13_ETH, 1081, 1133
 SA_OutputDistDistance14_ETH, 1081, 1133
 SA_OutputDistDistance15_ETH, 1081, 1133
 SA_OutputDistDistance16_ETH, 1082, 1133
 SA_OutputDistMinDistance_ETH, 1082, 1133
 SA_OutputDistMinDistance_RS422, 1085, 1134
 SA_OutputDistMinDistID_ETH, 1082, 1134
 SA_OutputDistMinDistID_RS422, 1085, 1135
 SA_OutputDriver, 1175, 1176
 SA_OutputDrivers, 1185
 SA_OutputEdgeDarkLight_ETH, 873, 926
 SA_OutputEdgeDarkLight_RS422, 873, 926
 SA_OutputEdgeLightDark_ETH, 873, 925
 SA_OutputEdgeLightDark_RS422, 873, 925
 SA_OutputErrorOut, 497, 597, 742, 807
 SA_OutputErrorOut1, 308, 355
 SA_OutputErrorOut2, 308, 355
 SA_OutputEthernet, 496, 597, 742, 807
 SA_OutputFormat, 197, 206, 213, 218, 219, 230, 238, 242, 243
 SA_OutputGapCenterAxis_ETH, 875, 931
 SA_OutputGapCenterAxis_RS422, 875, 930
 SA_OutputGapDifference_ETH, 875, 931
 SA_OutputGapDifference_RS422, 874, 930
 SA_OutputGapEdgeA_ETH, 875, 931
 SA_OutputGapEdgeA_RS422, 874, 930
 SA_OutputGapEdgeB_ETH, 875, 931
 SA_OutputGapEdgeB_RS422, 874, 930
 SA_OutputLightSensorBlue_ETH, 1078, 1140
 SA_OutputLightSensorBlue_RS422, 1083, 1143
 SA_OutputLightSensorBright_ETH, 1078, 1140
 SA_OutputLightSensorBright_RS422, 1083, 1143
 SA_OutputLightSensorGreen_ETH, 1078, 1143
 SA_OutputLightSensorGreen_RS422, 1083, 1143

SA_OutputLightSensorRed_RS422, 1083, SA_OutputSegment3EdgeB_RS422, 876,
1142 937
SA_Output.MaxValue, 1390, 1422 SA_OutputSegment4CenterAxis_ETH, 881,
SA_Output.MinValue, 1389, 1422 946
SA_OutputMode, 276, 279 SA_OutputSegment4CenterAxis_RS422, 877,
SA_OutputPeak1..16_Ch1_ETH, 736, 751, 938
814 SA_OutputSegment4Difference_ETH, 881,
SA_OutputPeak1..16_Ch1_RS422, 734, 749, 946
811 SA_OutputSegment4Difference_RS422, 877,
SA_OutputPinCount, 1183 938
938
SA_OutputRS422, 308, 355, 496, 597, SA_OutputSegment4EdgeA_ETH, 881, 946
742, 806 SA_OutputSegment4EdgeA_RS422, 877,
SA_OutputScaleMode, 1389, 1421 937
SA_OutputSegment1CenterAxis_ETH, 880, SA_OutputSegment4EdgeB_ETH, 881, 946
945 SA_OutputSegment4EdgeB_RS422, 877,
SA_OutputSegment1CenterAxis_RS422, 876, 938
936 SA_OutputSegment5CenterAxis_ETH, 882,
SA_OutputSegment1Difference_ETH, 879, 947
945 SA_OutputSegment5CenterAxis_RS422, 878,
SA_OutputSegment1Difference_RS422, 875, 938
936 SA_OutputSegment5Difference_ETH, 881,
SA_OutputSegment1EdgeA_ETH, 879, 944 947
SA_OutputSegment1EdgeA_RS422, 875, SA_OutputSegment5Difference_RS422, 877,
936 938
SA_OutputSegment1EdgeB_ETH, 879, 944 SA_OutputSegment5EdgeA_ETH, 881, 946
SA_OutputSegment1EdgeB_RS422, 875, SA_OutputSegment5EdgeA_RS422, 877,
936 938
SA_OutputSegment2CenterAxis_ETH, 880, SA_OutputSegment5EdgeB_ETH, 881, 946
945 SA_OutputSegment5EdgeB_RS422, 877,
SA_OutputSegment2CenterAxis_RS422, 876, 938
937 SA_OutputSegment6CenterAxis_ETH, 882,
SA_OutputSegment2Difference_ETH, 880, 947
945 SA_OutputSegment6CenterAxis_RS422, 878,
SA_OutputSegment2Difference_RS422, 876, 939
937 SA_OutputSegment6Difference_ETH, 882,
SA_OutputSegment2EdgeA_ETH, 880, 945 947
SA_OutputSegment2EdgeA_RS422, 876, SA_OutputSegment6Difference_RS422, 878,
936 939
SA_OutputSegment2EdgeB_ETH, 880, 945 SA_OutputSegment6EdgeA_ETH, 882, 947
SA_OutputSegment2EdgeB_RS422, 876, SA_OutputSegment6EdgeA_RS422, 878,
937 938
SA_OutputSegment3CenterAxis_ETH, 881, SA_OutputSegment6EdgeB_ETH, 882, 947
946 SA_OutputSegment6EdgeB_RS422, 878,
SA_OutputSegment3CenterAxis_RS422, 877, 939
937 SA_OutputSegment7CenterAxis_ETH, 883,
SA_OutputSegment3Difference_ETH, 880, 948
946 SA_OutputSegment7CenterAxis_RS422, 879,
SA_OutputSegment3Difference_RS422, 876, 939
937 SA_OutputSegment7Difference_ETH, 882,
SA_OutputSegment3EdgeA_ETH, 880, 945 948
SA_OutputSegment3EdgeA_RS422, 876, SA_OutputSegment7Difference_RS422, 878,
937 939
SA_OutputSegment3EdgeB_ETH, 880, 945 SA_OutputSegment7EdgeA_ETH, 882, 947

SA_OutputSegment7EdgeA_RS422, 878, 939 SA_OutputSensor2Shutter_USB, 1393, 1427
SA_OutputSegment7EdgeB_ETH, 882, 947 SA_OutputSensor2Value_ETH, 1395, 1432
SA_OutputSegment7EdgeB_RS422, 878, 939 SA_OutputSensor2Value_RS422, 1392, 1423
SA_OutputSegment8CenterAxis_ETH, 883, 948 SA_OutputSensor2Value_USB, 1393, 1427
SA_OutputSegment8CenterAxis_RS422, 879, 940 SA_OutputStatistic2Max_ETH, 884, 952
SA_OutputSegment8Difference_ETH, 883, 948 SA_OutputStatistic2Max_RS422, 884, 950
SA_OutputSegment8Difference_RS422, 879, 940 SA_OutputStatistic2Min_ETH, 884, 951
SA_OutputSegment8EdgeA_ETH, 883, 948 SA_OutputStatistic2Peak2Peak_ETH, 885,
SA_OutputSegment8EdgeA_RS422, 879, 939 SA_OutputStatistic2Peak2Peak_RS422, 884,
SA_OutputSegment8EdgeB_ETH, 883, 948 952
SA_OutputSegment8EdgeB_RS422, 879, 940 SA_OutputStatistic_ETH, 491, 515, 615,
SA_OutputSegment8EdgeA_RS422, 879, 939 737, 752, 815
SA_OutputSegment8EdgeB_RS422, 879, 940 SA_OutputStatistic_RS422, 487, 511, 608,
SA_OutputSegment8EdgeA_RS422, 879, 939 735, 750, 812
SA_OutputSegment8EdgeB_ETH, 883, 948 SA_OutputStatisticMax_ETH, 400, 412,
SA_OutputSegment8EdgeB_RS422, 879, 940 444, 644, 713, 884, 951, 1082,
SA_OutputSensor1Additional_ETH, 1395, 1431 1136
SA_OutputSensor1Additional_RS422, 1391, 1423 SA_OutputStatisticMax_RS422, 398, 411,
SA_OutputSensor1Additional_USB, 1392, 1427 SA_OutputStatisticMin_ETH, 400, 412, 444,
SA_OutputSensor1Intensity_ETH, 1395, 1431 644, 713, 884, 951, 1082, 1135
SA_OutputSensor1Intensity_USB, 1393, 1427 SA_OutputStatisticMin_RS422, 398, 411,
SA_OutputSensor1Reflectivity_ETH, 1395, 1432 443, 644, 712, 883, 949, 1085,
SA_OutputSensor1Reflectivity_USB, 1393, 1427 1136
SA_OutputSensor1Shutter_ETH, 1395, 1432 SA_OutputStatusCounter_ETH, 1078, 1140
SA_OutputSensor1Shutter_USB, 1393, 1422 SA_OutputStatusCounter_RS422, 1083,
SA_OutputSensor1Value_ETH, 1394, 1431 1143
SA_OutputSensor1Value_RS422, 1391, 1423 SA_OutputStatusError_ETH, 1079, 1140
SA_OutputSensor1Value_USB, 1392, 1426 SA_OutputStatusError_RS422, 1084, 1143
SA_OutputSensor2Additional_ETH, 1395, 1432 SA_OutputStatusFramerate_ETH, 1077,
1139
SA_OutputSensor2Additional_RS422, 1392, 1423 SA_OutputStatusFramerate_RS422, 1082,
1142
SA_OutputSensor2Additional_USB, 1393, 1427 SA_OutputStatusShutter_ETH, 1078, 1139
SA_OutputSensor2Intensity_ETH, 1395, 1432 SA_OutputStatusShutter_RS422, 1083, 1142
SA_OutputSensor2Intensity_USB, 1393, 1427 SA_OutputStatusTempDetector_ETH, 1078,
1139
SA_OutputSensor2Reflectivity_ETH, 1396, 1432 SA_OutputStatusTempDetector_RS422, 1083,
1142
SA_OutputSensor2Reflectivity_USB, 1394, 1428 SA_OutputStatusTempLightSrc_ETH, 1079,
1142
SA_OutputSensor2Shutter_ETH, 1395, 1432 SA_OutputStatusTimestamp_ETH, 1079,

1140
SA_OutputStatusTimestamp_RS422, 1084, 1143
SA_OutputTemperature, 1267, 1280
SA_OutputThickness12_ETH, 399, 412, 641, 709
SA_OutputThickness12_RS422, 398, 411, 442, 638, 702
SA_OutputThickness13_ETH, 641, 709
SA_OutputThickness13_RS422, 638, 703
SA_OutputThickness14_ETH, 641, 709
SA_OutputThickness14_RS422, 638, 703
SA_OutputThickness15_ETH, 642, 709
SA_OutputThickness15_RS422, 639, 703
SA_OutputThickness16_ETH, 642, 710
SA_OutputThickness16_RS422, 639, 703
SA_OutputThickness23_ETH, 642, 710
SA_OutputThickness23_RS422, 639, 704
SA_OutputThickness24_ETH, 642, 710
SA_OutputThickness24_RS422, 639, 704
SA_OutputThickness25_ETH, 642, 710
SA_OutputThickness25_RS422, 639, 704
SA_OutputThickness26_ETH, 643, 710
SA_OutputThickness26_RS422, 640, 704
SA_OutputThickness34_ETH, 643, 711
SA_OutputThickness34_RS422, 640, 704
SA_OutputThickness35_ETH, 643, 711
SA_OutputThickness35_RS422, 640, 705
SA_OutputThickness36_ETH, 643, 711
SA_OutputThickness36_RS422, 640, 705
SA_OutputThickness45_ETH, 643, 711
SA_OutputThickness45_RS422, 640, 705
SA_OutputThickness46_ETH, 644, 711
SA_OutputThickness46_RS422, 641, 705
SA_OutputThickness56_ETH, 644, 712
SA_OutputThickness56_RS422, 641, 705
SA_OutputTime, 279
SA_OutputTimestamp, 1267, 1281
SA_OutputType, 272, 275, 277, 376, 379
SA_OutputUnlinearizedDistance_Ch1_ETH
SA_OutputUnlinearizedDistance_Ch1_RS422, 508, 605
SA_OutputUnlinearizedDistance_Ch2_ETH
SA_OutputUnlinearizedDistance_Ch2_RS422, 508, 605
SA_OutputUnlinearizedDistance1..6_Ch1_ETH, 488, 513
SA_OutputUnlinearizedDistance1..6_Ch1_RS422, 485, 508
SA_OutputUnlinearizedDistance1..6_Ch2_ETH, 489, 513
SA_OutputUnlinearizedDistance1..6_Ch2_RS422, 485, 509
SA_OutputVideoAbs_Ch1_ETH, 735, 750, 814
SA_OutputVideoAbs_Ch1_RS422, 734, 749, 811
SA_OutputVideoCorr_ETH, 414, 448
SA_OutputVideoDark_Ch1_ETH, 487, 511, 612
SA_OutputVideoDark_Ch1_RS422, 484, 507, 604
SA_OutputVideoDark_Ch2_ETH, 488, 512, 612
SA_OutputVideoDark_Ch2_RS422, 484, 507, 605
SA_OutputVideoDark_ETH, 647, 721, 1077, 1124
SA_OutputVideoDarkTable_ETH, 648, 721
SA_OutputVideoLight_Ch1_ETH, 487, 511, 612
SA_OutputVideoLight_Ch1_RS422, 484, 507, 604
SA_OutputVideoLight_Ch2_ETH, 488, 512, 612
SA_OutputVideoLight_Ch2_RS422, 484, 508, 605
SA_OutputVideoLight_ETH, 647, 721, 886, 956
SA_OutputVideoLightSpectrum_ETH, 1077, 1124
SA_OutputVideoLightTable_ETH, 648, 721, 886, 957
SA_OutputVideoLinearized_ETH, 1077, 1124
SA_OutputVideoRaw_Ch1_ETH, 487, 511, 611
SA_OutputVideoRaw_Ch1_RS422, 483, 507, 604
SA_OutputVideoRaw_Ch2_ETH, 488, 512, 612
SA_OutputVideoRaw_Ch2_RS422, 484, 507, 604
SA_OutputVideoRaw_ETH, 414, 448, 647, 721, 886, 956, 1077, 1124
SA_OutputVideoRaw_RS422, 290, 312, 363, 365
SA_OutputVideoThreshold_ETH, 648, 721, 887, 957
SA_OutputWavelength, 1267, 1281
SA_PacketCounter, 1286
SA_Parameter10, 833
SA_Parameter11, 833
SA_Password, 197
SA_PasswordChangeable, 1284

SA_PeakGlobalThreshold, 302, 338
 SA_PeakModulation, 505, 578
 SA_PeakModulation_Ch<n>, 579
 SA_PeakModulation_Ch1..2, 505
 SA_PeakSearching, 277, 280
 SA_PeakWidthMax, 302, 337
 SA_PilotLaser, 194, 227, 231, 239, 754, 769
 SA_Polarity, 827, 838
 SA_Port, 405, 422, 494, 551, 629, 666, 740, 778, 865, 900, 1077, 1101, 1355, 1361, 1384, 1404
 SA_Pos, 518, 540, 651, 657, 753, 768
 SA_Pos_Ch<n>, 540
 SA_Pos_Ch1..2, 519
 SA_Pos1..x, 432, 524, 527, 537, 568, 652, 656, 678, 758, 767, 1088, 1110
 SA_Pos1..x_Ch1, 525
 SA_Pos1..x_Ch2, 526
 SA_Pos1..x_Ch<n>, 538
 SA_Power, 827, 838
 SA_PowerError, 1335
 SA_PowerMode, 827, 838
 SA_PrecedingValues, 207, 214, 219, 220
 SA_PresetMode, 262, 319, 330, 527, 556, 759, 783
 SA_PresetNames, 261, 319, 328, 527, 555, 759, 782
 SA_PreTreated, 464
 SA_Protocol, 405, 422, 494, 551, 628, 666, 740, 778, 865, 900, 1076, 1101, 1355, 1361, 1384, 1404
 SA_Q1Hysteresis, 229, 237, 247, 248
 SA_Q1Negation, 229, 237, 247, 248
 SA_Q1Start, 229, 237, 247
 SA_Q1Value, 195
 SA_Q1Width, 229, 237, 247, 248
 SA_Q2Hysteresis, 230, 238, 249
 SA_Q2Negation, 230, 238, 249, 250
 SA_Q2Start, 229, 237, 249
 SA_Q2Value, 196
 SA_Q2Width, 230, 237, 249
 SA_Range, 253, 258, 272, 275, 280, 289, 317, 378, 381, 390, 392, 396, 415, 451, 452, 519, 540, 651, 657, 754, 768, 822, 841, 861, 890, 966, 998, 1033, 1044, 1053, 1290, 1291, 1310
 SA_Range_Ch<n>, 540
 SA_Range_Ch1..2, 519
 SA_Range1..32, 1311
 SA_Range1..3, 998
 SA_Range1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536), 1054
 SA_Range1..x, 525, 537, 652, 656, 758, 767, 1292
 SA_Range1..x_Ch1, 525
 SA_Range1..x_Ch2, 526
 SA_Range1..x_Ch<n>, 538
 SA_RangeBegin, 207, 214, 223, 230, 238, 250
 SA_RangeEnd, 207, 214, 224, 230, 238, 250, 251
 SA_Ranges, 452
 SA_RawRangeMax, 992, 1040, 1299
 SA_RawRangeMax1..32, 1299
 SA_RawRangeMax1..3, 993
 SA_RawRangeMax1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536), 1041
 SA_RawRangeMin, 992, 1040, 1299
 SA_RawRangeMin1..32, 1299
 SA_RawRangeMin1..3, 993
 SA_RawRangeMin1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536), 1040
 SA_RawValue, 1019
 SA_RealFrequency, 1316, 1317, 1336
 SA_RealRatio, 1317, 1336
 SA_RearEdge_Seg1, 858, 986
 SA_RearEdge_Seg2, 858, 986
 SA_RearEdge_Seg3, 986
 SA_RearEdge_Seg4, 986
 SA_RecoveryFirmwareChannel, 1160
 SA_RecoveryFirmwareCreatedOn, 1159
 SA_RecoveryFirmwareID, 1159
 SA_RecoveryFirmwareName, 1159
 SA_RecoveryFirmwareNotes, 1160
 SA_RecoveryFirmwareVersion, 1159
 SA_RecoveryFirmwareWorksWith, 1159
 SA_RecursiveCount, 305, 346, 408, 435, 632, 685, 868, 910, 1072, 1118, 1386, 1415
 SA_RecursiveCount<n>, 1407
 SA_RecursiveCount1..2, 1387
 SA_Ref, 1330
 SA_RefOffset, 833
 SA_RefractIndex, 469
 SA_RefractIndexFileIdx, 469, 470
 SA_RefractIndexFileName, 469, 470
 SA_RefractiveCorrection, 517, 573, 629, 682
 SA_RefractiveCorrection_Ch<n>, 574
 SA_RefractiveCorrection_Ch1..2, 517

SA_RefractiveIndex_nC, 529, 570, 653, 680
 SA_RefractiveIndex_nC_Ch<n>, 571
 SA_RefractiveIndex_nC_Ch1..2, 529
 SA_RefractiveIndex_nC1..x, 528, 568, 679
 SA_RefractiveIndex_nd, 528, 570, 652, 680
 SA_RefractiveIndex_nd_Ch<n>, 571
 SA_RefractiveIndex_nd_Ch1..2, 529
 SA_RefractiveIndex_nd1..x, 528, 568, 678
 SA_RefractiveIndex_nF, 415, 433, 528, 570, 652, 680
 SA_RefractiveIndex_nF_Ch<n>, 571
 SA_RefractiveIndex_nF_Ch1..2, 529
 SA_RefractiveIndex_nF1..x, 432, 528, 568, 678
 SA_RegHyst, 730
 SA_RelativePermittivity, 1044
 SA_RemoteAddress, 405, 422, 494, 551, 628, 666, 740, 778, 865, 900, 1077, 1101
 SA_ResampleAnalog, 309, 356, 497, 598, 635, 692, 742, 808, 872, 924, 1389, 1419
 SA_ResampleChart, 309, 357
 SA_ResampleEthernet, 410, 440, 497, 598, 635, 693, 743, 808, 872, 924, 1086, 1122, 1389, 1419
 SA_ResampleRS422, 309, 356, 410, 440, 497, 598, 635, 693, 742, 807, 872, 924, 1086, 1122, 1389, 1419
 SA_ResampleUSB, 1389, 1419
 SA_Resampling, 309, 357, 410, 440, 497, 599, 635, 692, 743, 808, 872, 923, 1086, 1122, 1388, 1419
 SA_Reserve_1, 841, 966
 SA_Reserve_10, 987
 SA_Reserve_2, 847, 973
 SA_Reserve_3, 848, 984
 SA_Reserve_4, 857, 985
 SA_Reserve_6, 986
 SA_Reserve_7, 986
 SA_Reserve_9, 987
 SA_Reserved1, 280
 SA_Reserved2, 280
 SA_Reserved3, 280
 SA_Reserved4, 280
 SA_Result, 999, 1015, 1057, 1312
 SA_Reverse, 474
 SA_RightEdge, 834, 837
 SA_ROIEnd, 255, 263, 304, 344, 407, 430, 503, 566, 630, 676, 747, 788, 867, 908
 SA_ROIEnd_Ch<n>, 567
 SA_ROIEnd_Ch1..2, 503
 SA_ROIStart, 255, 263, 304, 344, 407, 430, 503, 566, 630, 676, 747, 788, 867, 908
 SA_ROIStart_Ch<n>, 567
 SA_ROIStart_Ch1..2, 503
 SA_RS232_Baudrate, 848, 974
 SA_RS232_Parity, 848, 974
 SA_RS232_StopBits, 848, 974
 SA_RS232_TimeoutRecv, 849, 974
 SA_RS232_TimeoutSend, 849, 974
 SA_RS232Baudrate, 832, 1177
 SA_RS232FrameFormat, 1177
 SA_RS232Mode, 832
 SA_RS232Protocol, 1177
 SA_RS232SlaveID, 1177
 SA_RS422_Baudrate, 849, 974
 SA_RS422_Parity, 849, 974
 SA_RS422_StopBits, 849, 975
 SA_RS422_TimeoutRecv, 849, 975
 SA_RS422_TimeoutSend, 849, 975
 SA_RS422Baudrate, 1321
 SA_RS422OutSource, 1321
 SA_RS422RealBaudrate, 1321, 1322
 SA_RxDValue1..4, 1318
 SA_RxDValue1..6, 1344
 SA_Samplerate, 377, 390
 SA_SampleTime, 991, 996, 1036, 1050, 1296, 1308
 SA_SaveSettingsMode, 277, 279
 SA_ScaledValue, 835, 837, 1019
 SA_ScaleFactor, 206, 214, 216, 228, 236, 241
 SA_ScanStatus, 1382, 1383
 SA_Scopes, 1144
 SA_SearchDirection, 866, 904
 SA_SecondReference, 1343
 SA_SelectedSensor1..4, 1314
 SA_Sensor, 184, 252, 257, 274, 288, 316, 376, 389, 395, 414, 450, 452, 482, 518, 623, 650, 733, 752, 822, 861, 890, 1070, 1090, 1264, 1268, 1353, 1354, 1380, 1399
 SA_SensorAccessible, 188
 SA_SensorAddress, 1303
 SA_SensorBaudrate, 195, 208, 213, 215, 230, 235, 238, 255, 260, 279, 300, 326, 380, 406, 423, 456, 494, 553, 629, 668, 740, 780, 866, 901, 1031, 1072, 1103,

1266, 1271, 1288, 1289, 1303, 1383, 1405
SA_SensorBaudrate<n>, 1366
SA_SensorBaudrate1..8, 1356
SA_SensorCableLength, 1003
SA_SensorChanged, 1005
SA_SensorChannelBaudrate1..4, 1314
SA_SensorChannelRealBaudrate, 1314
SA_SensorChannelRealBaudrate1..4, 1315
SA_SensorData, 1016
SA_SensorDatabits, 195
SA_SensorDetectable1..xColor1..3, 1229
SA_SensorDetectable1..xMatcherID, 1229
SA_SensorDetectable1..xRepresentationsRGB1..3, 1229
SA_SensorDetectable1..xUUID, 1229
SA_SensorDetectableColor1..3, 1227, 1228, 1230, 1231
SA_SensorDetectableEntries, 1229
SA_SensorDetectableMatcherID, 1227, 1228
SA_SensorDetectableRepresentationsRGB1..3, 1228, 1230, 1231
SA_SensorDetectableUUID, 1227, 1228, 1230, 1231
SA_SensorEndMeasRange, 1003
SA_SensorErrorBits, 1356, 1367
SA_SensorIndex1..x, 1292
SA_SensorInterface, 1302
SA_SensorMidMeasRange, 1003
SA_SensorName, 997, 1003, 1032, 1052, 1289, 1290, 1310
SA_SensorName1..32, 1310
SA_SensorName1..3, 998
SA_SensorName1..4 (DT6200/KSS64xx)
 or 8 (DT6530) or 6 (DT6536), 1053
SA_SensorName1..x, 1292
SA_SensorOption, 1003
SA_SensorPage1..x, 1292
SA_SensorParamData, 1017
SA_SensorProductCode, 1003
SA_SensorRevisionIndex, 1003
SA_SensorSample1..xCorrectedColor1..3, 1257
SA_SensorSample1..xDetectionDistance1..3, 1257
SA_SensorSample1..xDetectionMatcher, 1257
SA_SensorSample1..xDetectionOutputPattern, 1257
SA_SensorSample1..xInputs, 1257
SA_SensorSample1..xRepresentationsRGB1..3, 1257
SA_SensorSample1..xSignalLevel, 1258
SA_SensorSample1..xTimestamp, 1257
SA_SensorSample1..xTransformedColor1..3, 1256
SA_SensorSample1..xUUID, 1256
SA_SensorSampleCorrectedColor1..3, 1255
SA_SensorSampleDetectionDistance1..3, 1255
SA_SensorSampleDetectionMatcher, 1255
SA_SensorSampleDetectionOutputPattern, 1255
SA_SensorSampleEntries, 1256
SA_SensorSampleInputs, 1256
SA_SensorSampleRepresentationsRGB1..3, 1255
SA_SensorSampleSignalLevel, 1256
SA_SensorSampleTimestamp, 1256
SA_SensorSampleTransformedColor1..3, 1255
SA_SensorSampleUUID, 1254
SA_SensorSerialNumber, 1003
SA_SensorStartMeasRange, 1003
SA_SensorStopbits, 195
SA_SensorTable, 524, 537, 651, 655, 758, 766
SA_SensorTable_Ch1, 525
SA_SensorTable_Ch2, 526
SA_SensorTable_Ch<n>, 538
SA_SensorTableCount, 524, 537, 652, 655, 758, 766
SA_SensorTableCount_Ch1, 525
SA_SensorTableCount_Ch2, 526
SA_SensorTableCount_Ch<n>, 538
SA_SensorTemperature, 991, 992, 1011
SA_SensorType, 274, 376, 390, 754, 768
SA_SensorType1..x, 759, 767
SA_Serial, 519, 540, 651, 657, 754, 768
SA_Serial_Ch<n>, 541
SA_Serial_Ch1..2, 519
SA_Serial1..x, 525, 537, 652, 656, 759, 767
SA_Serial1..x_Ch1, 526
SA_Serial1..x_Ch2, 526
SA_Serial1..x_Ch<n>, 538
SA_SerialNumber, 186, 198, 252, 257, 272, 275, 288, 316, 378, 390, 396, 415, 482, 518, 623, 650, 733, 753, 841, 861, 890, 966, 997, 999, 1032, 1033, 1052, 1054, 1070, 1090, 1264, 1268, 1290, 1291, 1293, 1310, 1311,

1353, 1354, 1380, 1399
SA_SerialNumber1..32, 1311
SA_SerialNumber1..3, 998
SA_SerialNumber1..4 (DT6200/KSS64xx)
 or 8 (DT6530) or 6 (DT6536),
 1053
SA_SerialNumber1..x, 1292
SA_SerialOutFormat, 973
SA_SettingCategories, 1184
SA_SettingNames, 319, 328, 527, 555,
 759, 782
SA_Settings, 1259
SA_SetupRecord, 170
SA_ShowChannels, 1041, 1052, 1056
SA_ShowLinearized, 1041, 1052, 1056
SA_ShutterFactor, 429
SA_ShutterMode, 302, 338, 406, 427,
 505, 560, 630, 671, 1086, 1106
SA_ShutterMode_Ch<n>, 561
SA_ShutterMode_Ch1..2, 505
SA_ShutterTime, 302, 340
SA_ShutterTime1, 407, 429, 504, 561,
 630, 673
SA_ShutterTime1_Ch<n>, 562
SA_ShutterTime1_Ch1..2, 504
SA_ShutterTime2, 407, 429, 504, 562,
 630, 673
SA_ShutterTime2_Ch<n>, 562
SA_ShutterTime2_Ch1..2, 504
SA_SkippedSpectra, 465
SA_SkippedVideo, 343, 564, 676, 787,
 907
SA_SlopeValue, 833
SA_SmoothVideo, 832
SA_SoftArtArm, 842, 966
SA_SoftArtBoot, 842, 966
SA_SoftArtDSP, 842, 966
SA_SoftVerArm, 842, 966
SA_SoftVerBoot, 842, 966
SA_SoftVerDSP, 842, 966
SA_SoftwareTrigger, 1377
SA_SoftwareTriggerEnable, 1377
SA_SoftwareTriggerMode, 325
SA_Softwareversion, 186, 272, 275, 289,
 317, 376, 392, 396, 415, 483,
 518, 624, 651, 733, 753, 822,
 862, 891, 999, 1033, 1054, 1070,
 1090, 1264, 1268, 1293, 1311,
 1353, 1354, 1380, 1400
SA_SpectralAv, 468
SA_Spectrum, 465
SA_Spectrum1..x, 1090, 1111
SA_Speed, 276, 278, 377, 379, 391
SA_SpikeCorrection, 409, 436, 498, 580,
 633, 686, 868, 911
SA_SpikeCorrection_Ch<n>, 581
SA_SpikeCorrection_Ch1..2, 498
SA_SplitPixel, 729
SA_SRIndex, 449, 460, 1001, 1006, 1009,
 1021, 1024, 1026, 1035, 1050,
 1055, 1058, 1062, 1064
SA_SSIFormat, 231, 239, 251
SA_State, 1286
SA_StaticIP, 185
SA_Statistic2Signal, 870, 914
SA_StatisticDepth, 409, 437, 634, 689,
 870, 916, 1086, 1120
SA_StatisticDepth1..x, 524, 584, 758, 795
SA_StatisticSignal, 633, 688, 869, 913,
 1087, 1119
SA_StatisticSignalEntries, 523, 583, 757,
 794
SA_StatisticSignalEntry1..x, 523, 583, 757,
 794
SA_StatisticSignalIndex1..x, 524, 583, 757,
 794
SA_StatisticSignalName1..x, 524, 584, 758,
 795
SA_StatisticSignalTable, 523, 583, 757,
 794
SA_Status, 991
SA_Status01..28, 1265, 1269
SA_SubnetMask, 185, 405, 421, 493,
 550, 628, 665, 739, 777, 865,
 899, 995, 1046, 1074, 1100,
 1301, 1355, 1360, 1384, 1403
SA_SuperluminescentLED, 754, 768
SA_SWType, 379
SA_Sync_TrgMode, 378, 380
SA_SyncLogic, 1382, 1385
SA_SyncMode, 293, 298, 400, 403, 501,
 530, 625, 654, 744, 760, 863,
 892, 1087, 1091, 1382, 1385
SA_SyncTermination, 401, 403, 863, 892
SA_SystemSignalRangeLowerLimit, 596
SA_SystemSignalRangeUpperLimit, 596
SA_Target_Distance, 857
SA_TargetMode, 301, 334
SA_TargetMode<n>, 1410
SA_TCPKeepAlive, 552, 740, 779
SA_TeachValue, 827, 835, 839
SA_TeachValue1, 277, 278
SA_TeachValue2, 277, 278
SA_Temperature, 199, 208, 232, 253,
 624
SA_Temperature<n>, 1410

SA_Temperature01..28, 1273
 SA_TemperatureCoeffizient, 1065
 SA_Termination, 294, 298, 501, 531, 745, 761
 SA_TerminationChar, 231, 239, 243, 244
 SA_TextField, 1001, 1005, 1006
 SA_ThicknessMeasureEnable, 1044
 SA_Threshold, 277, 280, 408, 432, 466, 631, 678, 867, 908
 SA_Threshold1, 467
 SA_Threshold2, 467
 SA_TimeCreated, 1149, 1150
 SA_TimeExpires, 1149, 1150
 SA_Timeout, 1149, 1150
 SA_TimerFrequency<n>, 1363
 SA_TimerFrequency1..3, 1356
 SA_TimerRatio<n>, 1364
 SA_TimerRatio1..3, 1356
 SA_Timestamp, 465
 SA_Timestamp01..28, 1273
 SA_Timezone, 1261, 1262
 SA_Timezones, 1262
 SA_Tolerance1..xLimitHalfEdge1..3, 1184
 SA_Tolerance1..xLimitHalfHeight, 1184
 SA_Tolerance1..xLimitRadius, 1184
 SA_Tolerance1..xShape, 1183
 SA_ToleranceEntries, 1183
 SA_ToleranceHigh, 828, 839
 SA_ToleranceLow, 828, 839
 SA_ToleranceRange, 409, 437, 498, 580, 633, 687, 868, 911
 SA_ToleranceRange_Ch<n>, 582
 SA_ToleranceRange_Ch1..2, 499
 SA_ToleranceValue, 835
 SA_TransmitIntensity, 473
 SA_TrgChannel1..4, 1319
 SA_TrgChannel1..6, 1346
 SA_TrgMode, 1021, 1025, 1026, 1037, 1051, 1056, 1059, 1063, 1064, 1285, 1297, 1309
 SA_TrgValue1..6, 1346
 SA_Trigger1..xEventCntEdgeFalling, 1180
 SA_Trigger1..xEventCntEdgeRising, 1179
 SA_Trigger1..xEventCntLevelHigh, 1179
 SA_Trigger1..xEventCntLevelLow, 1179
 SA_Trigger1..xEvents, 1184
 SA_Trigger1..xSource, 1179, 1184
 SA_TriggerCount, 254, 260, 300, 324, 404, 419, 500, 545, 626, 660, 744, 772, 864, 898, 1088, 1099, 1386, 1402
 SA_TriggerDelay, 207, 211, 215, 228, 234, 236
 SA_TriggerEdge, 207, 211, 215, 228, 234, 236
 SA_TriggerEncoderMax, 500, 546, 744, 773
 SA_TriggerEncoderMin, 500, 546, 744, 772
 SA_TriggerEncoderStepsize, 500, 547, 744, 773
 SA_TriggerEntries, 1179, 1184
 SA_TriggerInLogic, 499, 544, 743, 771
 SA_TriggerLevel, 254, 259, 299, 324, 404, 418, 500, 545, 626, 660, 743, 771, 864, 897, 1088, 1098, 1386, 1401
 SA_TriggerLogic, 1386, 1401
 SA_TriggerMode, 228, 234, 254, 258, 299, 322, 323, 404, 417, 499, 543, 626, 659, 743, 770, 828, 839, 864, 896, 1087, 1098, 1385, 1400
 SA_TriggerMoment, 254, 259, 299, 323, 404, 418, 500, 544, 626, 659, 744, 770, 864, 897
 SA_TriggerOutput, 404, 419
 SA_TriggerOutput<n>, 1367
 SA_TriggerOutput1..8, 1356
 SA_TriggerSource, 299, 321, 499, 543, 743, 769
 SA_TriggerTermination, 404, 418, 864, 896
 SA_TxDChannel1..4, 1318
 SA_TxDChannel1..6, 1344
 SA_TxDValue1..6, 1345
 SA_Unit, 519, 540, 651, 657, 754, 768, 891, 895, 998, 1033, 1053, 1290, 1291, 1310
 SA_Unit_Ch<n>, 541
 SA_Unit_Ch1..2, 519
 SA_Unit1..32, 1311
 SA_Unit1..3, 999
 SA_Unit1..4 (DT6200/KSS64xx) or 8 (DT6530) or 6 (DT6536), 1054
 SA_Unit1..x, 525, 537, 652, 656, 759, 767, 895, 1292
 SA_Unit1..x_Ch1, 526
 SA_Unit1..x_Ch2, 526
 SA_Unit1..x_Ch<n>, 538
 SA_UnlinearizedEddyValue, 1066
 SA_UnlinearizedMode, 653, 720
 SA_Upper_hysteresis, 381
 SA_Upper_limit, 380
 SA_UpperBound<n>, 917
 SA_UpperLimit, 313, 368, 649, 724, 857,

888, 960, 984
SA_UpperLimit< n >, 268, 368, 618, 817,
 1436
SA_UpperLimit1..2, 314, 502, 745, 1391
SA_UpperLimit1..3, 257
SA_UpperWarning, 857, 984
SA_USBFrameFormat, 1179
SA_USBProtocol, 1178
SA_USBSlaveID, 1178
SA_User1..xName, 1148
SA_User1..xPasswordHash, 1148
SA_User1..xRoles, 1148
SA_UserEntries, 1147
SA_UserLevel, 297, 320, 403, 416, 492,
 536, 625, 654, 738, 765, 863,
 893, 1074, 1093, 1283
SA_UserName, 1146--1149
SA_UserPasswordHash, 1146, 1147
SA_UserRole1..xID, 1145
SA_UserRole1..xPermission1..yAllow, 1145
SA_UserRole1..xPermission1..yScope, 114
SA_UserRole1..xPermissionEntries, 1145
SA_UserRoleEntries, 1144
SA_UserRoles, 1146, 1147
SA_UserSessionTimeout, 1149
SA_UUID, 184
SA_ValidRange, 207, 215, 220
SA_Value, 996, 1294, 1304
SA_Version, 194, 199, 205, 227, 451,
 997, 1025, 1052, 1063, 1309
SA_VideoAbs_Ch1, 787
SA_VideoAverage, 408, 431, 631, 677,
 1088, 1117
SA_VideoCorr, 426
SA_VideoDark, 673, 675, 1107
SA_VideoDark_Ch1, 564
SA_VideoDark_Ch2, 564
SA_VideoDarkTable, 673, 675
SA_VideoLight, 673, 675, 907, 1107
SA_VideoLight_Ch1, 564
SA_VideoLight_Ch2, 564
SA_VideoLightTable, 674, 675, 907
SA_VideoLin, 1107
SA_VideoMax, 836
SA_VideoRaw, 343, 426, 673, 675, 906,
 1106
SA_VideoRaw_Ch1, 563
SA_VideoRaw_Ch2, 564
SA_VideoSignal, 284, 385, 834
SA_VideoThrAuto, 824, 832
SA_VideoThreshold, 674, 675, 907
SA_VideoThrFix, 831
SA_VideoThrMode, 829, 840
SA_VideoTimestamp, 343, 426, 564, 674,
 675, 787, 907, 1107
SA_Watchdog, 458
SA_WatchdogPeriod, 459
SA_WaveLength, 416, 432, 433
SA_Wavelength01..28, 1273
SA_WebpageVersion, 1353, 1354, 1380,
 1400
SA_Webstatic, 1070, 1091
SA_WhiteCorrFactor1_01..28, 1270, 1276
SA_WhiteCorrFactor2_01..28, 1270, 1276
SA_WhiteCorrFactor3_01..28, 1270, 1277
SA_WhiteLED, 1278
SA_WhiteRef, 480
SA_WindowMax, 229, 236, 245
SA_WindowMin, 228, 236, 245
SA_X1..16, 450, 475
SA_X1..x, 1089, 1111
SA_Y1..x, 1090, 1111
SA_Z1..x, 1090, 1111
SA_ZeroPoint, 391
scaledData, 93, 95, 96
sensor, 70
SensorCommand, 91
sensorCommand, 115--123
sensorName, 71, 73
serialNumber, 109, 110
SetDoubleExecSCmd, 117
SetDoubleExecSCmdU, 118
SetIntExecSCmd, 116
SetIntExecSCmdU, 116
SetParameterBinary, 79
SetParameterBinaryU, 79
SetParameterDouble, 77
SetParameterDoubleU, 77
SetParameterDWORD_PTR, 76
SetParameterDWORD_PTRU, 76
SetParameterInt, 74
SetParameterIntU, 75
SetParameters, 80
SetParameterString, 78
SetParameterStringU, 78
SetParametersU, 81
SetStringExecSCmd, 118
SetStringExecSCmdU, 119
SP_0V, 476
SP_10V, 477
SP_a*, 1113
SP_AbbeNumber_vd, 572, 681
SP_ActionLockEditState, 1235
SP_ActionLockState, 1234
SP_ActionLockViewState, 1234
SP_ActionMatcherID, 1234

SP_ActionName, 1232, 1234
 SP_ActionRemoveMatcherDetectablesBefore, 1234
 SP_ActionTriggerAPIEditlock, 1241, 1245
 SP_ActionTriggerAPIViewlock, 1240, 1245
 SP_ActionTriggerAutogain, 1240, 1245
 SP_ActionTriggerEnableSwitchingOutput, 1239, 1244
 SP_ActionTriggerEvent, 1239, 1244, 1251, 1254
 SP_ActionTriggerKeylock, 1240, 1244
 SP_ActionTriggerLockEditState, 1241, 1245
 SP_ActionTriggerLockState, 1240, 1244
 SP_ActionTriggerLockViewState, 1240, 1245
 SP_ActionTriggerMatcherID, 1239, 1244
 SP_ActionTriggerRemoveAllDetectables, 1240, 1245
 SP_ActionTriggerRemoveAllMatchers, 1245
 SP_ActionTriggerRemoveMatcherDetectablesBefore, 1239, 1244
 SP_ActionTriggerSource, 1239, 1243, 1251, 1254
 SP_ActionTriggerTeachSingle, 1239, 1244
 SP_ActionTriggerUUID, 1243, 1248, 1253
 SP_ActivateGate, 1318, 1344
 SP_Active, 837
 SP_ActiveEdge, 455
 SP_ActiveMaterial, 433, 568, 679, 789
 SP_ActiveMaterial_Ch<n>, 569
 SP_ActiveSensor, 539, 656, 767
 SP_ActiveSensor_Ch<n>, 539
 SP_ActiveSerialIf, 845, 851, 970, 977
 SP_ActualPos, 1115
 SP_ADCLatchSource, 1349
 SP_ADCNumber, 1349, 1350
 SP_Additional, 254, 297, 402, 492, 625, 738, 863, 1071, 1266, 1353, 1383
 SP_Address, 420, 549, 664, 777, 898, 994, 1045, 1099, 1300, 1359, 1402
 SP_AlarmHysteresis, 222
 SP_AlarmStart, 221
 SP_AlarmWidth, 222
 SP_Algorithm, 1313
 SP_AllDevices, 165, 1382
 SP_AlternateBlockIdx, 1287
 SP_AnalogChannel, 1304--1306
 SP_AnalogEEPROMData, 1016
 SP_AnalogFactor1..3, 1307
 SP_AnalogGain, 853, 979
 SP_AnalogMathFunctionEnable, 1307
 SP_AnalogOffset, 853, 979, 1305, 1307
 SP_AnalogOutput, 725, 826, 961, 1436
 SP_AnalogOutput<n>, 1437
 SP_AnalogOutputSignalIndex, 619, 818
 SP_AnalogOutputSignalName, 619, 819
 SP_AnalogParamData, 1017
 SP_AnalogRange, 371, 620, 726, 819, 963, 1304, 1438
 SP_AnalogRange<n>, 1439
 SP_AnalogScaleMode, 372, 373, 620, 726, 820, 964, 1439
 SP_AnalogScaleMode<n>, 1441
 SP_AnalogScaleRangeLowerLimit, 268, 374, 621, 820
 SP_AnalogScaleRangeUpperLimit, 269, 374, 621, 821
 SP_AnalogScaleSource, 375
 SP_AnalogUnit, 1306
 SP_AnalogValue, 1436
 SP_AnalogValue<n>, 1437
 SP_AnalogZoom, 830
 SP_ApplicationDisplayMode, 1008
 SP_ApplicationLanguage, 327, 553, 781, 995, 1007, 1047, 1301, 1362, 1405
 SP_ApplicationScaleMode, 1007
 SP_ApplicationYMax, 1007
 SP_ApplicationYMin, 1007
 SP_ApplyImmediately, 331, 557, 785
 SP_ASCII, 286, 387
 SP_AutoAdaptLED, 471
 SP_AutoAdaptLEDTThr, 471
 SP_AutoDark, 453
 SP_Automatic, 329, 555, 783
 SP_AutostartCommand, 204, 225
 SP_AutostartEdge, 211
 SP_AutostartTrigger, 211
 SP_Average, 217, 246
 SP_Average_for_reading, 854, 980
 SP_Averaging, 472, 825, 1279
 SP_AveragingForDark, 453
 SP_AveragingType, 344, 434, 684, 909, 1029, 1117, 1285, 1414
 SP_AveragingType<n>, 1406
 SP_AveragingValue, 1030, 1286
 SP_AvlIndex, 386, 393
 SP_AvrNbr, 1011, 1022, 1038, 1060, 1298
 SP_AvrType, 1010, 1022, 1037, 1059, 1297
 SP_AvType, 285, 386, 393
 SP_b*, 1113
 SP_BarycenterOffset, 479
 SP_BarycenterScale, 478

SP_Binning, 468
 SP_Block, 995, 996
 SP_BlockIndex, 834
 SP_BlockSize, 481
 SP_CalDefaultType, 730
 SP_CalibEnd, 1013
 SP_CalibMid, 1013
 SP_CalibStart, 1013
 SP_CalibTable, 481, 622, 731
 SP_CalibTarget, 1012
 SP_Chan, 992, 997, 1030, 1032, 1039, 1040, 1042, 1043, 1047--1049, 1052, 1061, 1062, 1065--1067, 1287, 1289, 1290, 1298, 1309
 SP_Channel, 1265, 1272, 1274--1279
 SP_ChannelMode<n>, 1362
 SP_ChartType, 333, 558, 785
 SP_ChTransmit1..4, 1060
 SP_ChTransmit1..8, 1039
 SP_CmdStr, 164
 SP_Coefficient, 1027
 SP_CoeffParam, 1027
 SP_ColorName, 1112--1114, 1116
 SP_ColorSpace, 1115, 1273
 SP_ColorSpaceID, 1187
 SP_Command, 252, 288, 395, 482, 623, 732, 861, 1069, 1263
 SP_Command<n>, 1368, 1408
 SP_Complete, 993, 998, 1040, 1053, 1299, 1310
 SP_ComputationFactor1, 591, 802
 SP_ComputationFactor2, 591, 802
 SP_ComputationID, 351, 352, 589, 592, 800, 803
 SP_ComputationMedianCount, 351, 590, 802
 SP_ComputationMovingCount, 351, 590, 801
 SP_ComputationName, 592, 803
 SP_ComputationOffset, 592, 803
 SP_ComputationOperation, 351, 590, 801
 SP_ComputationQueue, 350, 352, 589, 592, 800, 803
 SP_ComputationRecursiveCount, 351, 590, 801
 SP_ComputationSignalIndex, 351, 590, 801
 SP_ComputationSignalIndex1, 591, 802
 SP_ComputationSignalIndex2, 591, 802
 SP_ComputationSignalName, 351, 590, 801
 SP_ComputationSignalName1, 591, 802
 SP_ComputationSignalName2, 592, 803
 SP_ContinuousMode, 200
 SP_Contrast, 845, 851, 970, 976
 SP_CurrentProfileUUID, 1208
 SP_DarkCorrAverage, 729
 SP_DarkCorrOffset1..3, 1277
 SP_DarkCorrThreshold, 658
 SP_DarkPixel, 830
 SP_DataMode, 1319
 SP_DataOutInterface, 264, 354, 439, 691, 922, 1121, 1279, 1417
 SP_DataPort, 993, 1044, 1299
 SP_DataScale, 478
 SP_DataSource, 367, 722, 958
 SP_DefaultKey, 1151
 SP_DefaultObjectType, 1150
 SP_DefaultsHoldTime, 1152
 SP_DefaultsProperty, 1151, 1153
 SP_DefaultsToleranceShape, 1152
 SP_DefaultsToleranceShapeBoxLimitHalfEdge1..3, 1153
 SP_DefaultsToleranceShapeCylinderLimitHalfHeight, 1152
 SP_DefaultsToleranceShapeCylinderLimitRadius, 1152
 SP_DefaultsToleranceShapeSphereLimitRadius, 1152
 SP_DefaultType, 262, 330, 425, 556, 669, 784, 902, 1105, 1272
 SP_DefaultUser, 320, 416, 536, 655, 765, 893, 1093
 SP_DefaultUUID, 1151, 1153, 1156
 SP_DefaultValue, 1151
 SP_Delta_A_AB, 1114
 SP_Delta_B, 1114
 SP_Delta_E_L, 1114
 SP_DeltaKC, 1109
 SP_DeltaKH, 1109
 SP_DeltaKL, 1108
 SP_DeltaMode, 1108
 SP_Description, 434, 571, 572, 680, 681, 790, 1113
 SP_DesiredBaudrate, 1288
 SP_DetectionProfileColorSpaceID, 1191, 1195
 SP_DetectionProfileEnableAmbientLightCompensation, 1213
 SP_DetectionProfileEnableInternalEmitter, 1213
 SP_DetectionProfileLevel, 1213
 SP_DetectionProfileMinimumSamplingRate, 1213
 SP_DetectionProfileName, 1191, 1195

SP_DetectionProfileNonMatchingHoldTime
 1191, 1196
 SP_DetectionProfileNonMatchingOutput-
 State, 1191, 1196
 SP_DetectionProfileSamplingAverageCount
 1191, 1196
 SP_DetectionProfileUUID, 1195, 1200, 1212
 1215, 1216
 SP_DeviceInstance, 1303
 SP_DeviceProperty, 1157
 SP_DHCPEnabled, 420, 549, 664, 776,
 898, 994, 1045, 1099, 1300,
 1359, 1402
 SP_DigitalIndexData, 1018
 SP_DigitalInLatchSource, 1347, 1372
 SP_DigitalInMode1, 1373
 SP_DigitalInMode2, 1374
 SP_DigitalInMode3, 1374
 SP_DigitalLevel, 1372
 SP_DigitalOut1..4, 1348
 SP_DigitalOutBinFormat, 1104
 SP_DigitalOutColorFormat, 1103
 SP_DigitalOutDetectedID, 1137
 SP_DigitalOutSource<n>, 1375
 SP_DigitalOutValue1..4, 1348
 SP_DigitalScaleMode<n>, 1416
 SP_DigitalScaleValue<n>, 1417
 SP_DisablePower, 1334
 SP_DisplayGain, 853, 979
 SP_DisplayOffset, 853, 979
 SP_DisplayUnit, 292, 401, 491, 737, 892,
 1381
 SP_DispMeasUnit, 844, 850, 969, 976
 SP_DistanceMode, 1128
 SP_DoubleFrequency, 462
 SP_Echo, 296, 402, 491, 624, 737, 862,
 1071
 SP_Edge<n>_A, 905
 SP_Edge<n>_B, 905
 SP_EdgeDetectThreshold, 970, 976
 SP_EdgeFilter<n>, 916
 SP_EdgeFilter<n>Signal, 917
 SP_EnableFlash, 383
 SP_Encoder1, 455
 SP_Encoder2, 456
 SP_Encoder3, 456
 SP_Encoder3Enable, 774
 SP_EncoderDirection, 1338
 SP_EncoderDirection<n>, 1369
 SP_EncoderIncrements, 661
 SP_EncoderInterpolation, 1337
 SP_EncoderInterpolation<n>, 547, 662,
 774, 1368
 SP_EncoderLatchSource, 1340
 SP_EncoderLatchSource<n>, 1370
 SP_Encoder.MaxValue, 661
 SP_Encoder.MaxValue<n>, 549, 664, 776
 SP_Encoder.MinValue, 661
 SP_Encoder.Mode, 1339
 SP_Encoder.Mode<n>, 547, 662, 775,
 1368
 SP_Encoder.Number, 660, 1336--1343
 SP_Encoder.Preload<n>, 548, 663, 775,
 1369
 SP_Encoder.PreloadValue, 1341
 SP_ErrorHandler, 273, 286, 387, 844,
 850, 969, 976
 SP_ErrorHysteresis, 369
 SP_ErrorHysteresis<n>, 266
 SP_ErrorLevel, 724
 SP_ErrorLevelOut1, 370
 SP_ErrorLevelOut2, 371
 SP_ErrorLevelOut<n>, 266, 618, 818,
 961
 SP_ErrorLimitCompOp<n>, 267, 369,
 618, 817
 SP_ErrorLimitSignalIndex<n>, 616, 815
 SP_ErrorLimitSignalName<n>, 616, 816
 SP_ErrorMode, 220, 244
 SP_ErrorOut<n>HoldTime, 960
 SP_ErrorOutHoldTime, 268, 370
 SP_ErrorOutput, 384, 1450
 SP_ErrorOutput1, 366
 SP_ErrorOutput2, 366
 SP_ErrorOutput<n>, 265, 615, 722, 957,
 1434
 SP_EthernetMode, 422, 552, 666, 779,
 900, 1046, 1102, 1301
 SP_EvalBegin, 825
 SP_EvalEnd, 825
 SP_EvalMode, 824
 SP_ExpectedEdges, 904
 SP_ExportType, 330, 557, 784
 SP_Exposure, 461
 SP_ExposureLimit, 341
 SP_ExposureMode, 340
 SP_Ext_LaserSwitch, 844, 851, 970, 976
 SP_ExtInputMode, 282
 SP_ExtTriggerInDirection1..4, 1322
 SP_ExtTriggerInSource, 1322
 SP_ExtTriggerOutSource1..2, 1323
 SP_FactorCapa, 1061
 SP_FactorCh1..4 (DT6200) or 8 (DT6530)
 or 6 (DT6536), 1042
 SP_FactorEddy, 1061
 SP_FirmwareFile, 999, 1015, 1057, 1312

SP_FirmwareReleaseChannel, 1162
 SP_FirmwareUUID, 1160, 1162, 1163
 SP_FirstPeak, 459
 SP_ForceOverwrite, 331, 557, 784
 SP_FramesPerPacket_ETH, 600, 693, 809, 1361, 1420
 SP_FreeSR, 461
 SP_FrontEdge_Seg1, 843, 855, 968, 981
 SP_FrontEdge_Seg2, 843, 855, 968, 981
 SP_FrontEdge_Seg3, 968, 981
 SP_FrontEdge_Seg4, 968, 981
 SP_FullRange, 481
 SP_Gain, 1275
 SP_GainPoti, 1014
 SP_GateChannel, 1317, 1343
 SP_Gateway, 420, 550, 665, 777, 898, 994, 1045, 1100, 1300, 1360, 1402
 SP_GroupIndex, 790
 SP_HalfMinDist, 728
 SP_HighFrequency, 463
 SP_HoldLastValid, 264, 357, 440, 599, 693, 808, 924, 1122, 1420
 SP_Hostname, 1259
 SP_HTTPPeTag, 335
 SP_HTTPMaxLength, 335
 SP_HTTPPath, 335
 SP_HWMode, 826
 SP_HysteresisQ1, 201
 SP_HysteresisQ2, 201
 SP_IgnoreValues<n>, 917
 SP_ImportCategories, 1258
 SP_ImportData, 331, 557, 785
 SP_IncludeAdditionalParameters, 169
 SP_IncludedInterfaceParameters, 169
 SP_IncludeSavedSetups, 168
 SP_IndexData, 1018
 SP_InitialChartType, 333
 SP_InitialName, 329, 555, 783
 SP_IntegrationTime, 1274
 SP_InternalTriggerMode, 830
 SP_InvalidValues, 219
 SP_JustActualSetup, 1365, 1411
 SP_KeepDevice, 903, 1105, 1365, 1411
 SP_KeyAction, 1173
 SP_KeyFunction, 296
 SP_KeyFunction1, 532, 763
 SP_KeyFunction2, 533, 763
 SP_Keylock, 280, 295, 382, 392, 531, 762, 1091
 SP_KeylockTime, 295, 532, 762, 1092
 SP_KeyName, 1173
 SP_KeypadLocked, 1172
 SP_KeypadOutputName, 1180
 SP_L*, 1112
 SP_LampTest, 457
 SP_LampTestThr, 457
 SP_Language, 844, 850, 969, 975
 SP_LaserDiode<n>, 283
 SP_LaserIntensity, 845, 851, 970, 976
 SP_LaserPower, 262, 342, 429, 906
 SP_LaserPower<n>, 1366, 1406
 SP_LastValid, 473
 SP_LED_Off, 472
 SP_LEDControl, 541, 657, 1095
 SP_LEDControl_Ch<n>, 541
 SP_LEDIntensity, 470
 SP_LEDIntensityColdWhite, 1096
 SP_LEDIntensityGreen, 1096
 SP_LEDIntensityViolet, 1097
 SP_LEDIntensityWarmWhite, 1096
 SP_LEDMode1, 1324
 SP_LEDMode2, 1324
 SP_LEDMode3, 1324
 SP_LEDMode4, 1325
 SP_LightSource, 1094, 1112
 SP_LimitQ1-1, 201
 SP_LimitQ1-2, 202
 SP_LimitQ2-1, 201
 SP_LimitQ2-2, 202
 SP_LimitQA-1, 203
 SP_LimitQA-2, 203
 SP_LinMode, 1023, 1048
 SP_LinPoint, 1049
 SP_LinPos, 1023, 1024, 1048--1050, 1066, 1067
 SP_LinPoti, 1014
 SP_LoadReg, 1329
 SP_Location, 823, 824, 826, 829, 831
 SP_Lower_hysteresis, 388
 SP_Lower_limit, 388
 SP_LowerBound<n>, 916
 SP_LowerLimit, 723, 854, 959, 979
 SP_LowerLimit<n>, 267, 368, 617, 816, 1435
 SP_LowerWarning, 854, 980
 SP_LowFrequency, 463
 SP_LowPass, 1038
 SP_Master, 346, 350, 438, 588, 690, 799, 921, 1416
 SP_Master_value, 388
 SP_MasterAction, 349, 588, 799
 SP_MasterDistance1..6_Ch1, 534
 SP_MasterDistance1..6_Ch2, 535
 SP_MasterName, 588, 799
 SP_MasterPeak1..16_Ch1, 764

SP_MasterSignal, 347–349, 585, 587, 689, 795, 796, 798, 919
 SP_MasterSignalName, 585–587, 796–799
 SP_MasterSignalNames, 535, 764
 SP_MasterSource, 349
 SP_MasterValue, 346, 347, 438, 585, 690, 796, 855, 922, 982, 1416
 SP_MatcherDetectionProfileUUID, 1225
 SP_MatcherHoldTime, 1218, 1221
 SP_MatcherName, 1216, 1220
 SP_MatcherOutputPattern, 1218, 1221
 SP_MatcherProperty, 1220, 1223
 SP_MatcherResetOutputAfterHoldTimeEx- pired, 1218, 1221
 SP_MatcherToleranceShape, 1217, 1220
 SP_MatcherToleranceShapeBoxLimitHalfE- 1217, 1221
 SP_MatcherToleranceShapeCylinderLim- itHalfHeight, 1217, 1220
 SP_MatcherToleranceShapeCylinderLim- itRadius, 1217, 1220
 SP_MatcherToleranceShapeSphereLim- itRadius, 1217, 1220
 SP_MatcherUUID, 1219, 1223, 1227
 SP_MaterialIndex, 569, 679
 SP_MaterialIndex_Ch<n>, 570
 SP_MaterialMultiPeak12, 575, 683, 791
 SP_MaterialMultiPeak12_Ch<n>, 576
 SP_MaterialMultiPeak23, 575, 683, 791
 SP_MaterialMultiPeak23_Ch<n>, 576
 SP_MaterialMultiPeak34, 575, 683, 791
 SP_MaterialMultiPeak34_Ch<n>, 576
 SP_MaterialMultiPeak45, 575, 683, 791
 SP_MaterialMultiPeak45_Ch<n>, 576
 SP_MaterialMultiPeak56, 575, 683, 791
 SP_MaterialMultiPeak56_Ch<n>, 576
 SP_MaterialName, 433, 434, 571, 572, 680, 681, 790
 SP_MaxValue, 372, 727, 964, 1440
 SP_MaxValue<n>, 1441
 SP_MeasDistIndex, 895
 SP_MeasFrequency, 240
 SP_MeasMode, 980
 SP_MeasObject, 855, 981
 SP_MeasProgName, 853, 979
 SP_MeasProgNumber, 843, 844, 850, 967, 969, 975
 SP_Measrate, 339, 428, 563, 672, 786, 1028, 1107, 1272, 1284, 1413
 SP_MeasureCount, 1009
 SP_MeasureDirection, 904
 SP_MeasureMode, 335, 384, 425, 459, 669, 903, 1010, 1105, 1413
 SP_MeasurePeak, 336, 426, 558, 670
 SP_MeasurePeak_Ch<n>, 559
 SP_MeasurePeak<n>, 1409
 SP_MeasurePeakSort, 785
 SP_MeasureTime, 210
 SP_MeasureValueCnt, 441
 SP_MeasValue, 386
 SP_Median, 980
 SP_Median_OnOff, 273
 SP_MedianCount, 345, 435, 685, 909, 1118, 1415
 SP_MedianCount<n>, 1407
 SP_MedianIndex, 285
 SP_MinGap, 728
 SP_MinimumThreshold, 577, 792
 SP_MinimumThreshold_Ch<n>, 578
 SP_MinValue, 372, 727, 964, 1440
 SP_MinValue<n>, 1441
 SP_ModeQ1, 202
 SP_ModeQ2, 202
 SP_MovingCount, 285, 345, 434, 684, 909, 1117, 1414
 SP_MovingCount<n>, 1407
 SP_MultiFunctionInputLevel, 332, 1452
 SP_MultiFunctionInputLogic, 1412
 SP_MultiFunctionInputMode, 331, 1412
 SP_Multiplexer, 1019
 SP_NbrCorrectedValues, 436, 580, 686, 911
 SP_NbrCorrectedValues_Ch<n>, 581
 SP_NbrEvaluatedValues, 436, 579, 686, 911
 SP_NbrEvaluatedValues_Ch<n>, 581
 SP_NetworkInterfaceAddress, 1165
 SP_NetworkInterfaceFamily, 1165
 SP_NetworkInterfaceGateway, 1166
 SP_NetworkInterfaceIPv4Config1Address, 1164
 SP_NetworkInterfaceIPv4Config1Gateway, 1164
 SP_NetworkInterfaceIPv4Config1Method, 1164
 SP_NetworkInterfaceIPv4ConfigEntries, 1164
 SP_NetworkInterfaceIPv6Config1Address, 1165
 SP_NetworkInterfaceIPv6Config1Gateway, 1165
 SP_NetworkInterfaceIPv6Config1Method, 1165
 SP_NetworkInterfaceIPv6ConfigEntries, 1164
 SP_NetworkInterfaceMethod, 1165

SP_NetworkInterfaceName, 1163, 1168
 SP_NetworkInterfaceProperty, 1168
 SP_NewName, 328, 554, 782
 SP_NewPassword, 321, 417, 537, 655, 766, 894, 991, 1035, 1093, 1284, 1296
 SP_NewPos, 1116
 SP_NormQ1, 203
 SP_NormQ2, 203
 SP_NormQA, 204
 SP_Now, 1260
 SP_NTPServers, 1260
 SP_NullPoti, 1014
 SP_NumberOfPeaks, 574, 682, 790
 SP_NumberOfPeaks_Ch<n>, 574
 SP_NumberOfPoints, 454
 SP_NumberOfSegments, 981
 SP_Observer, 1094, 1112
 SP_Offset, 200, 217, 241, 996, 1042, 1061
 SP_OffsetCorrection, 1043
 SP_OldName, 328, 554, 782
 SP_OldPassword, 321, 417, 536, 655, 766, 894, 991, 1034, 1093, 1284, 1295
 SP_OperationMode, 283, 826
 SP_Option, 474
 SP_OutMode, 831
 SP_OutNr, 476
 SP_OutputAdditional, 1433
 SP_OutputAdditionalAmount_Ch1_ETH, 81
 SP_OutputAdditionalAmount_Ch1_RS422, 810
 SP_OutputAdditionalCounter_ETH, 446, 611, 717, 813, 954
 SP_OutputAdditionalCounter_RS422, 358, 363, 444, 604, 714, 810, 952
 SP_OutputAdditionalDistanceRaw_RS422, 360, 364
 SP_OutputAdditionalEncoder1_Ch1_ETH, 609
 SP_OutputAdditionalEncoder1_Ch1_RS422, 602
 SP_OutputAdditionalEncoder1_Ch2_ETH, 610
 SP_OutputAdditionalEncoder1_Ch2_RS422, 602
 SP_OutputAdditionalEncoder1_ETH, 717
 SP_OutputAdditionalEncoder1_RS422, 714
 SP_OutputAdditionalEncoder2_Ch1_ETH, 610
 SP_OutputAdditionalEncoder2_Ch1_RS422, 602
 SP_OutputAdditionalEncoder2_Ch2_ETH, 610
 SP_OutputAdditionalEncoder2_Ch2_RS422, 603
 SP_OutputAdditionalEncoder2_ETH, 717
 SP_OutputAdditionalEncoder2_RS422, 714
 SP_OutputAdditionalEncoder3_ETH, 717
 SP_OutputAdditionalEncoder3_RS422, 714
 SP_OutputAdditionalEncoder1..3_Ch1_ETH, 813
 SP_OutputAdditionalEncoder1..3_Ch1_RS422, 810
 SP_OutputAdditionalIntensity_Ch1_ETH, 610
 SP_OutputAdditionalIntensity_Ch1_RS422, 602
 SP_OutputAdditionalIntensity_Ch2_ETH, 610
 SP_OutputAdditionalIntensity_Ch2_RS422, 603
 SP_OutputAdditionalIntensity_ETH, 446, 717
 SP_OutputAdditionalIntensity_RS422, 359, 363, 445, 715
 SP_OutputAdditionalMearstate_ETH, 610, 718, 813
 SP_OutputAdditionalMearstate_RS422, 359, 603, 715, 810
 SP_OutputAdditionalNbrEdges_ETH, 954
 SP_OutputAdditionalNbrEdges_RS422, 952
 SP_OutputAdditionalNbrGaps_ETH, 954
 SP_OutputAdditionalNbrGaps_RS422, 952
 SP_OutputAdditionalNbrPins_ETH, 954
 SP_OutputAdditionalNbrPins_RS422, 952
 SP_OutputAdditionalShutterTime_Ch1_ETH, 609, 813
 SP_OutputAdditionalShutterTime_Ch1_RS422, 602, 810
 SP_OutputAdditionalShutterTime_Ch2_ETH, 610
 SP_OutputAdditionalShutterTime_Ch2_RS422, 602
 SP_OutputAdditionalShutterTime_ETH, 446, 813, 954
 SP_OutputAdditionalShutterTime_RS422, 358, 363, 444, 714
 SP_OutputAdditionalState_ETH, 447, 718, 813, 953
 SP_OutputAdditionalState_RS422, 360, 364, 445, 715, 810, 953
 SP_OutputAdditionalTemperature_ETH, 447
 SP_OutputAdditionalTemperature_RS422, 445

SP_OutputAdditionalTimestamp_ETH, 446 SP_OutputDiameterEdgeA_RS422, 927
611, 717, 813, 954 SP_OutputDiameterEdgeB_ETH, 928
SP_OutputAdditionalTimestamp_RS422, 265 SP_OutputDiameterEdgeB_RS422, 927
359, 363, 445, 603, 714, 810, 952 SP_OutputDistance1_ETH, 697
SP_OutputDistance1_RS422, 358, 441,
SP_OutputAdditionalTimestampHi_ETH, 611 694
SP_OutputAdditionalTimestampHi_RS422, SP_OutputDistance2_ETH, 697
359, 603 SP_OutputDistance2_RS422, 441, 694
SP_OutputAdditionalTimestampLo_ETH, 611 SP_OutputDistance3_ETH, 697
SP_OutputAdditionalTimestampLo_RS422, SP_OutputDistance3_RS422, 694
359, 603 SP_OutputDistance4_ETH, 697
SP_OutputDistance4_RS422, 695
SP_OutputAdditionalTrgCounter_ETH, 447 SP_OutputDistance5_ETH, 697
SP_OutputAdditionalTrgTimeDiff_ETH, 611 SP_OutputDistance5_RS422, 695
718 SP_OutputDistance6_ETH, 698
SP_OutputAdditionalTrgTimeDiff_RS422, 603, 715 SP_OutputDistance6_RS422, 695
SP_OutputAnalog, 354, 597, 806 SP_OutputDistance1..6_Ch1_ETH, 609
SP_OutputC-BoxAdditional_ETH, 1430 SP_OutputDistance1..6_Ch1_RS422, 601
SP_OutputC-BoxAdditional_RS422, 1423 SP_OutputDistance1..6_Ch2_ETH, 609
SP_OutputC-BoxAdditional_USB, 1426 SP_OutputDistance1..6_Ch2_RS422, 601
SP_OutputC-BoxCounter_ETH, 1430 SP_OutputDistanceUnlinearized_RS422,
360
SP_OutputC-BoxCounter_USB, 1426 SP_OutputDistDetectedID_ETH, 1131
SP_OutputC-BoxDigital_ETH, 1431 SP_OutputDistDetectedID_RS422, 1134
SP_OutputC-BoxDigital_USB, 1426 SP_OutputDistDistance01_ETH, 1129
SP_OutputC-BoxTimestamp_ETH, 1431 SP_OutputDistDistance02_ETH, 1129
SP_OutputC-BoxTimestamp_USB, 1426 SP_OutputDistDistance03_ETH, 1129
SP_OutputC-BoxValue_ETH, 1430 SP_OutputDistDistance04_ETH, 1129
SP_OutputC-BoxValue_RS422, 1423 SP_OutputDistDistance05_ETH, 1129
SP_OutputC-BoxValue_USB, 1426 SP_OutputDistDistance06_ETH, 1130
SP_OutputCalculation_ETH, 611, 813 SP_OutputDistDistance07_ETH, 1130
SP_OutputCalculation_RS422, 604, 811 SP_OutputDistDistance08_ETH, 1130
SP_OutputCh01..28, 1280 SP_OutputDistDistance09_ETH, 1130
SP_OutputColorLAB99_ETH, 1125 SP_OutputDistDistance10_ETH, 1130
SP_OutputColorLAB99_RS422, 1127 SP_OutputDistDistance11_ETH, 1130
SP_OutputColorLAB_ETH, 1125 SP_OutputDistDistance12_ETH, 1130
SP_OutputColorLAB_RS422, 1127 SP_OutputDistDistance13_ETH, 1130
SP_OutputColorLCH99_ETH, 1125 SP_OutputDistDistance14_ETH, 1131
SP_OutputColorLCH99_RS422, 1127 SP_OutputDistDistance15_ETH, 1131
SP_OutputColorLCH_ETH, 1125 SP_OutputDistDistance16_ETH, 1131
SP_OutputColorLCH_RS422, 1127 SP_OutputDistMinDistance_ETH, 1131
SP_OutputColorLUV_ETH, 1125 SP_OutputDistMinDistance_RS422, 1134
SP_OutputColorLUV_RS422, 1127 SP_OutputDistMinDistID_ETH, 1131
SP_OutputColorRGB_ETH, 1124 SP_OutputDistMinDistID_RS422, 1134
SP_OutputColorRGB_RS422, 1126 SP_OutputDriver, 1175
SP_OutputColorXYZ_ETH, 1124 SP_OutputEdgeDarkLight_ETH, 926
SP_OutputColorXYZ_RS422, 1126 SP_OutputEdgeDarkLight_RS422, 926
SP_OutputContent, 242 SP_OutputEdgeLightDark_ETH, 925
SP_OutputDiameterCenterAxis_ETH, 928 SP_OutputEdgeLightDark_RS422, 925
SP_OutputDiameterCenterAxis_RS422, 927 SP_OutputErrorOut, 597, 806
SP_OutputDiameterDifference_ETH, 928 SP_OutputErrorOut1, 354
SP_OutputDiameterDifference_RS422, 927 SP_OutputErrorOut2, 354
SP_OutputDiameterEdgeA_ETH, 928 SP_OutputEthernet, 597, 806

SP_OutputFormat, 218, 242
 SP_OutputGapCenterAxis_ETH, 931
 SP_OutputGapCenterAxis_RS422, 929
 SP_OutputGapDifference_ETH, 931
 SP_OutputGapDifference_RS422, 929
 SP_OutputGapEdgeA_ETH, 930
 SP_OutputGapEdgeA_RS422, 929
 SP_OutputGapEdgeB_ETH, 930
 SP_OutputGapEdgeB_RS422, 929
 SP_OutputLightSensorBlue_ETH, 1138
 SP_OutputLightSensorBlue_RS422, 1141
 SP_OutputLightSensorBright_ETH, 1138
 SP_OutputLightSensorBright_RS422, 1141
 SP_OutputLightSensorGreen_ETH, 1138
 SP_OutputLightSensorGreen_RS422, 1141
 SP_OutputLightSensorRed_ETH, 1138
 SP_OutputLightSensorRed_RS422, 1141
 SP_Output.MaxValue, 1421
 SP_Output.MinValue, 1421
 SP_OutputMode, 286
 SP_OutputPeak1..16_Ch1_ETH, 812
 SP_OutputPeak1..16_Ch1_RS422, 810
 SP_OutputRS422, 354, 596, 806
 SP_OutputScaleMode, 1421
 SP_OutputSegment1CenterAxis_ETH, 941
 SP_OutputSegment1CenterAxis_RS422, 939
 SP_OutputSegment1Difference_ETH, 940
 SP_OutputSegment1Difference_RS422, 938
 SP_OutputSegment1EdgeA_ETH, 940
 SP_OutputSegment1EdgeA_RS422, 932
 SP_OutputSegment1EdgeB_ETH, 940
 SP_OutputSegment1EdgeB_RS422, 932
 SP_OutputSegment2CenterAxis_ETH, 941
 SP_OutputSegment2CenterAxis_RS422, 939
 SP_OutputSegment2Difference_ETH, 941
 SP_OutputSegment2Difference_RS422, 939
 SP_OutputSegment2EdgeA_ETH, 941
 SP_OutputSegment2EdgeA_RS422, 932
 SP_OutputSegment2EdgeB_ETH, 941
 SP_OutputSegment2EdgeB_RS422, 932
 SP_OutputSegment3CenterAxis_ETH, 942
 SP_OutputSegment3CenterAxis_RS422, 939
 SP_OutputSegment3Difference_ETH, 941
 SP_OutputSegment3Difference_RS422, 939
 SP_OutputSegment3EdgeA_ETH, 941
 SP_OutputSegment3EdgeA_RS422, 933
 SP_OutputSegment3EdgeB_ETH, 941
 SP_OutputSegment3EdgeB_RS422, 933
 SP_OutputSegment4CenterAxis_ETH, 942
 SP_OutputSegment4CenterAxis_RS422, 939
 SP_OutputSegment4Difference_ETH, 942
 SP_OutputSegment4Difference_RS422, 939
 SP_OutputSegment4EdgeA_ETH, 942
 SP_OutputSegment4EdgeA_RS422, 933
 SP_OutputSegment4EdgeB_ETH, 942
 SP_OutputSegment4EdgeB_RS422, 933
 SP_OutputSegment5CenterAxis_ETH, 943
 SP_OutputSegment5CenterAxis_RS422, 934
 SP_OutputSegment5Difference_ETH, 942
 SP_OutputSegment5Difference_RS422, 934
 SP_OutputSegment5EdgeA_ETH, 942
 SP_OutputSegment5EdgeA_RS422, 934
 SP_OutputSegment5EdgeB_ETH, 942
 SP_OutputSegment5EdgeB_RS422, 934
 SP_OutputSegment6CenterAxis_ETH, 943
 SP_OutputSegment6CenterAxis_RS422, 935
 SP_OutputSegment6Difference_ETH, 943
 SP_OutputSegment6Difference_RS422, 935
 SP_OutputSegment6EdgeA_ETH, 943
 SP_OutputSegment6EdgeA_RS422, 934
 SP_OutputSegment6EdgeB_ETH, 943
 SP_OutputSegment6EdgeB_RS422, 934
 SP_OutputSegment7CenterAxis_ETH, 944
 SP_OutputSegment7CenterAxis_RS422, 935
 SP_OutputSegment7Difference_ETH, 943
 SP_OutputSegment7Difference_RS422, 935
 SP_OutputSegment7EdgeA_ETH, 943
 SP_OutputSegment7EdgeA_RS422, 935
 SP_OutputSegment7EdgeB_ETH, 943
 SP_OutputSegment7EdgeB_RS422, 935
 SP_OutputSegment8CenterAxis_ETH, 944
 SP_OutputSegment8CenterAxis_RS422, 936
 SP_OutputSegment8Difference_ETH, 944
 SP_OutputSegment8Difference_RS422, 936
 SP_OutputSegment8EdgeA_ETH, 944
 SP_OutputSegment8EdgeA_RS422, 935
 SP_OutputSegment8EdgeB_ETH, 944
 SP_OutputSegment8EdgeB_RS422, 935
 SP_OutputSensor1Additional_ETH, 1429
 SP_OutputSensor1Additional_RS422, 1422
 SP_OutputSensor1Additional_USB, 1424
 SP_OutputSensor1Intensity_ETH, 1429
 SP_OutputSensor1Intensity_USB, 1424
 SP_OutputSensor1Reflectivity_ETH, 1429
 SP_OutputSensor1Reflectivity_USB, 1425
 SP_OutputSensor1Shutter_ETH, 1429
 SP_OutputSensor1Shutter_USB, 1424
 SP_OutputSensor1Value_ETH, 1428
 SP_OutputSensor1Value_RS422, 1422
 SP_OutputSensor1Value_USB, 1424
 SP_OutputSensor2Additional_ETH, 1429
 SP_OutputSensor2Additional_RS422, 1422
 SP_OutputSensor2Additional_USB, 1425
 SP_OutputSensor2Intensity_ETH, 1430
 SP_OutputSensor2Intensity_USB, 1425
 SP_OutputSensor2Reflectivity_ETH, 1430

SP_OutputSensor2Reflectivity_USB, 1425
 SP_OutputSensor2Shutter_ETH, 1430
 SP_OutputSensor2Shutter_USB, 1425
 SP_OutputSensor2Value_ETH, 1429
 SP_OutputSensor2Value_RS422, 1422
 SP_OutputSensor2Value_USB, 1425
 SP_OutputStatistic2Max_ETH, 951
 SP_OutputStatistic2Max_RS422, 949
 SP_OutputStatistic2Min_ETH, 951
 SP_OutputStatistic2Min_RS422, 949
 SP_OutputStatistic2Peak2Peak_ETH, 951
 SP_OutputStatistic2Peak2Peak_RS422, 948
 SP_OutputStatistic_ETH, 611, 814
 SP_OutputStatistic_RS422, 604, 811
 SP_OutputStatisticMax_ETH, 443, 713, 950, 1135
 SP_OutputStatisticMax_RS422, 442, 712, 949, 1136
 SP_OutputStatisticMin_ETH, 443, 713, 950, 1135
 SP_OutputStatisticMin_RS422, 442, 712, 948, 1136
 SP_OutputStatisticPeak2Peak_ETH, 444, 713, 950, 1135
 SP_OutputStatisticPeak2Peak_RS422, 443, 712, 949, 1136
 SP_OutputStatusCounter_ETH, 1138
 SP_OutputStatusCounter_RS422, 1141
 SP_OutputStatusError_ETH, 1139
 SP_OutputStatusError_RS422, 1142
 SP_OutputStatusFramerate_ETH, 1137
 SP_OutputStatusFramerate_RS422, 1140
 SP_OutputStatusShutter_ETH, 1138
 SP_OutputStatusShutter_RS422, 1141
 SP_OutputStatusTempDetector_ETH, 1138
 SP_OutputStatusTempDetector_RS422, 1138
 SP_OutputStatusTempLightSrc_ETH, 1138
 SP_OutputStatusTempLightSrc_RS422, 1138
 SP_OutputStatusTimestamp_ETH, 1139
 SP_OutputStatusTimestamp_RS422, 1142
 SP_OutputTemperature, 1280
 SP_OutputThickness12_ETH, 706
 SP_OutputThickness12_RS422, 442, 699
 SP_OutputThickness13_ETH, 706
 SP_OutputThickness13_RS422, 699
 SP_OutputThickness14_ETH, 706
 SP_OutputThickness14_RS422, 700
 SP_OutputThickness15_ETH, 706
 SP_OutputThickness15_RS422, 700
 SP_OutputThickness16_ETH, 707
 SP_OutputThickness16_RS422, 700
 SP_OutputThickness23_ETH, 707
 SP_OutputThickness23_RS422, 700
 SP_OutputThickness24_ETH, 707
 SP_OutputThickness24_RS422, 700
 SP_OutputThickness25_ETH, 707
 SP_OutputThickness25_RS422, 701
 SP_OutputThickness26_ETH, 707
 SP_OutputThickness26_RS422, 701
 SP_OutputThickness34_ETH, 708
 SP_OutputThickness34_RS422, 701
 SP_OutputThickness35_ETH, 708
 SP_OutputThickness35_RS422, 701
 SP_OutputThickness36_ETH, 708
 SP_OutputThickness36_RS422, 702
 SP_OutputThickness45_ETH, 708
 SP_OutputThickness45_RS422, 702
 SP_OutputThickness46_ETH, 708
 SP_OutputThickness46_RS422, 702
 SP_OutputThickness56_ETH, 709
 SP_OutputThickness56_RS422, 702
 SP_OutputTime, 287
 SP_OutputTimestamp, 1280
 SP_OutputType, 273, 286, 387
 SP_OutputUnlinearizedDistance_Ch1_ETH, 609
 SP_OutputUnlinearizedDistance_Ch1_RS422, 601
 SP_OutputUnlinearizedDistance_Ch2_ETH, 609
 SP_OutputUnlinearizedDistance_Ch2_RS422, 602
 SP_OutputVideoAbs_Ch1_ETH, 812
 SP_OutputVideoAbs_Ch1_RS422, 809
 SP_OutputVideoCorr_ETH, 448
 SP_OutputVideoDark_Ch1_ETH, 608
 SP_OutputVideoDark_Ch1_RS422, 600
 SP_OutputVideoDark_Ch2_ETH, 608
 SP_OutputVideoDark_Ch2_RS422, 601
 SP_OutputVideoDarkTable_ETH, 720, 1123
 SP_OutputVideoDarkTable_RS422, 1123
 SP_OutputVideoLight_Ch1_ETH, 608
 SP_OutputVideoLight_Ch1_RS422, 600
 SP_OutputVideoLight_Ch2_ETH, 609
 SP_OutputVideoLight_Ch2_RS422, 601
 SP_OutputVideoLight_ETH, 720, 955
 SP_OutputVideoLightSpectrum_ETH, 1123
 SP_OutputVideoLightTable_ETH, 720, 956
 SP_OutputVideoLinearized_ETH, 1123
 SP_OutputVideoRaw_Ch1_ETH, 608
 SP_OutputVideoRaw_Ch1_RS422, 600
 SP_OutputVideoRaw_Ch2_ETH, 608
 SP_OutputVideoRaw_Ch2_RS422, 601
 SP_OutputVideoRaw_ETH, 448, 720, 955, 1123
 SP_OutputVideoRaw_RS422, 360, 365

SP_OutputVideoThreshold_ETH, 721, 956
 SP_OutputWavelength, 1280
 SP_Overwrite, 328, 554, 782
 SP_Page, 996
 SP_PageNo, 1030, 1032, 1287, 1289–
 1291
 SP_Parameter10, 831
 SP_Parameter11, 831
 SP_ParameterID, 1294, 1304
 SP_ParameterName, 1294
 SP_ParameterSet, 424, 668, 669, 902,
 1104, 1105, 1364, 1376, 1410,
 1411
 SP_ParameterType, 424, 668, 902, 1104,
 1364, 1411
 SP_Password, 320, 416, 536, 654, 765,
 893, 990, 1034, 1092, 1283,
 1295
 SP_PeakGlobalThreshold, 337
 SP_PeakModulation, 578
 SP_PeakModulation_Ch<n>, 579
 SP_PeakSearching, 284
 SP_PeakWidthMax, 337
 SP_PilotLaser, 199, 227, 769
 SP_Polarity, 824
 SP_Port, 421, 551, 666, 778, 900, 1101,
 1361, 1403
 SP_Pos, 1112
 SP_Power, 824
 SP_PowerMode, 824
 SP_PrecedingValues, 219
 SP_PresetMode, 261, 329, 556, 783
 SP_Protocol, 421, 550, 665, 778, 899,
 1101, 1360, 1403
 SP_Q1Hysteresis, 247
 SP_Q1Negation, 247
 SP_Q1Start, 246
 SP_Q1Width, 246
 SP_Q2Hysteresis, 248
 SP_Q2Negation, 248
 SP_Q2Start, 248
 SP_Q2Width, 248
 SP_Range, 1031, 1040, 1044, 1287
 SP_RangeBegin, 223, 250
 SP_RangeEnd, 223, 250
 SP_Read, 1291
 SP_ReadMode, 342, 465, 563, 674, 786,
 906
 SP_RearEdge_Seg1, 843, 855, 968, 982
 SP_RearEdge_Seg2, 843, 855, 968, 982
 SP_RearEdge_Seg3, 968, 982
 SP_RearEdge_Seg4, 969, 982
 SP_RecoveryFirmwareProperty, 1159
 SP_RecursiveCount, 345, 435, 684, 909,
 1117, 1414
 SP_RecursiveCount<n>, 1407
 SP_RefOffset, 831
 SP_RefractIndex, 468
 SP_RefractIndexFile, 480
 SP_RefractIndexFileIdx, 469, 479
 SP_RefractionCorrection, 573, 682
 SP_RefractionCorrection_Ch<n>, 573
 SP_RefractionIndex_nC, 572, 681
 SP_RefractionIndex_nd, 572, 681
 SP_RefractionIndex_nF, 434, 572, 681
 SP_RelativePermittivity, 1043
 SP_RemoteAddress, 421, 551, 666, 778,
 899, 1101
 SP_ResampleAnalog, 356, 598, 692, 807,
 923, 1418
 SP_ResampleChart, 356
 SP_ResampleEthernet, 439, 598, 692,
 807, 923, 1122, 1419
 SP_ResampleRS422, 356, 439, 598, 692,
 807, 923, 1122, 1418
 SP_ResampleUSB, 1418
 SP_Resampling, 357, 439, 599, 692, 808,
 923, 1121, 1418
 SP_Reserve_10, 982
 SP_Reserve_2, 844, 850, 970, 977
 SP_Reserve_3, 845, 851, 980
 SP_Reserve_4, 854, 980
 SP_Reserve_6, 981
 SP_Reserve_7, 982
 SP_Reserve_9, 982
 SP_ResetMeasCounter, 294, 402, 531,
 761
 SP_ResetTimestamp, 254, 294, 401, 531,
 761
 SP_ResetTriggerCounter, 402
 SP_Reverse, 474
 SP_ROIEnd, 263, 344, 430, 566, 676,
 787, 908
 SP_ROIEnd_Ch<n>, 567
 SP_ROIStart, 263, 343, 430, 566, 676,
 787, 907
 SP_ROIStart_Ch<n>, 566
 SP_RS232_Baudrate, 845, 851, 970, 977
 SP_RS232_Parity, 845, 851, 971, 977
 SP_RS232_StopBits, 845, 852, 971, 977
 SP_RS232_TimeoutRecv, 846, 852, 971,
 977
 SP_RS232_TimeoutSend, 846, 852, 971,
 977
 SP_RS232Baudrate, 823, 830, 1177
 SP_RS232Echo, 1262

SP_RS232FrameFormat, 1176
 SP_RS232Mode, 829
 SP_RS232OutputFormat, 1262
 SP_RS232Protocol, 1176
 SP_RS232SlaveID, 1176
 SP_RS422_Baudrate, 846, 852, 971, 978
 SP_RS422_Parity, 846, 852, 971, 978
 SP_RS422_StopBits, 846, 852, 971, 978
 SP_RS422_TimeoutRecv, 846, 853, 972, 978
 SP_RS422_TimeoutSend, 846, 852, 972, 978
 SP_RS422Baudrate, 1321
 SP_RS422OutSource, 1320
 SP_SampleProperty, 1254
 SP_SampleTime, 1036, 1296
 SP_SaveSettingsMode, 281
 SP_SaveTemporary, 1105
 SP_ScaleFactor, 216, 240
 SP_SearchDirection, 903
 SP_SelectedSensor1..4, 1313
 SP_Sensor, 452, 1408
 SP_SensorAddress, 1032, 1289, 1303
 SP_SensorBaudrate, 212, 235, 260, 281, 325, 383, 423, 456, 552, 667, 780, 901, 1031, 1102, 1270, 1288, 1302, 1404
 SP_SensorBaudrate<n>, 1365
 SP_SensorChannel, 1312, 1314, 1328, 1333, 1352
 SP_SensorChannelBaudrate, 1314
 SP_SensorData, 1016
 SP_SensorDetectableColor1..3, 1228, 1230
 SP_Store, 1031, 1288
 SP_SensorDetectableMatcherID, 1227, 1228
 SP_SubnetMask, 420, 549, 664, 777, 898, 994, 1045, 1100, 1300, 1359, 1402
 SP_SensorDetectableMatcherUUID, 1229
 SP_SensorDetectableProfileUUID, 1229, 1232
 SP_SensorDetectableProperty, 1231
 SP_SensorDetectableUUID, 1230, 1231
 SP_SensorInterface, 1302
 SP_SensorParamData, 1017
 SP_SensorReset12, 1328
 SP_SensorReset34, 1329
 SP_SensorSampleCount, 1258
 SP_SensorSampleFrequency, 1258
 SP_SensorTable, 730
 SP_SerialOutFormat, 969, 976
 SP_SettingName, 327, 328, 330, 554, 555, 557, 781, 782, 784
 SP_Settings, 1259
 SP_SetupRecord, 168
 SP_ShowChannels, 1041
 SP_ShowLinearized, 1041
 SP_ShutterFactor, 429
 SP_ShutterMode, 338, 427, 560, 671, 1106
 SP_ShutterMode_Ch<n>, 560
 SP_ShutterTime, 340
 SP_ShutterTime1, 428, 561, 672
 SP_ShutterTime1_Ch<n>, 562
 SP_ShutterTime2, 428, 561, 672
 SP_ShutterTime2_Ch<n>, 562
 SP_SlopeValue, 831
 SP_SmoothVideo, 830
 SP_SOD, 476
 SP_SoftwareTrigger, 1377
 SP_SoftwareTriggerEnable, 1376
 SP_SoftwareTriggerMode, 325
 SP_SpectralAv, 467
 SP_Speed, 284, 385
 SP_SpikeCorrection, 436, 579, 686, 910
 SP_SpikeCorrection_Ch<n>, 581
 SP_SplitPixel, 728
 SP_SRIndex, 460, 1009, 1020, 1035, 1058
 SP_SSIFormat, 251
 SP_Stand-by, 199
 SP_Statistic2Signal, 913
 SP_StatisticDepth, 437, 582, 688, 793, 915, 1120
 SP_StatisticSignal, 582--584, 687, 793, 795, 912, 1119
 SP_StatisticSignalName, 582--584, 793--795
 SP_SuperluminescentLED, 768
 SP_Sync_TrgMode, 384, 1449
 SP_SyncLogic, 1381
 SP_SyncMasterChannel, 1313, 1328, 1333, 1352
 SP_SyncMode, 293, 400, 530, 653, 760, 891, 1091, 1381, 1450--1452
 SP_SyncTermination, 400, 892, 1451
 SP_SystemSignalRangeLowerLimit, 595
 SP_SystemSignalRangeUpperLimit, 596
 SP_Target_Distance, 854
 SP_TargetMode, 334
 SP_TargetMode<n>, 1409
 SP_TCPKeepAlive, 551, 779
 SP_TeachValue, 825
 SP_TeachValue1, 287
 SP_TeachValue2, 287

SP_TempCoeffParam, 1065
 SP_TemperatureCoeffizient, 1065
 SP_Termination, 293, 530, 761, 1450
 SP_TerminationChar, 243
 SP_TextField, 1005
 SP_ThicknessMeasureEnable, 1043
 SP_Threshold, 285, 431, 466, 677, 908
 SP_Threshold1, 466
 SP_Threshold2, 467
 SP_TimeProperty, 1261
 SP_TimerFrequency, 1316, 1336
 SP_TimerFrequency<n>, 1363
 SP_TimerNumber, 1316, 1317, 1335
 SP_TimerRatio, 1316, 1336
 SP_TimerRatio<n>, 1363
 SP_Timezone, 1260, 1262
 SP_ToleranceHigh, 825
 SP_ToleranceLow, 825
 SP_ToleranceRange, 436, 580, 686, 911
 SP_ToleranceRange_Ch<n>, 581
 SP_TransmitIntensity, 394, 473
 SP_TrgChannel1..4, 1319
 SP_TrgChannel1..6, 1345
 SP_TrgMode, 1021, 1036, 1059, 1285,
 1296
 SP_TrgValue1..6, 1346
 SP_TriggerCount, 259, 324, 419, 545,
 660, 772, 897, 1099, 1401
 SP_TriggerDelay, 210, 233
 SP_TriggerEdge, 210, 233
 SP_TriggerEncoderMax, 546, 773
 SP_TriggerEncoderMin, 545, 772
 SP_TriggerEncoderStepsize, 546, 773
 SP_TriggerInLogic, 544, 771, 1452
 SP_TriggerLevel, 259, 324, 418, 544,
 659, 771, 897, 1098, 1401
 SP_TriggerLogic, 1401
 SP_TriggerMode, 233, 258, 322, 417,
 543, 658, 770, 826, 896, 1098,
 1400
 SP_TriggerMode_Edge, 454
 SP_TriggerMode_State, 455
 SP_TriggerMoment, 258, 323, 418, 543,
 659, 770, 896
 SP_TriggerOutput, 419
 SP_TriggerOutput<n>, 1367
 SP_TriggerSource, 321, 542, 769
 SP_TriggerTermination, 417, 896
 SP_TxDChannel1..4, 1318
 SP_TxDChannel1..6, 1344
 SP_TxDValue1..6, 1345
 SP_UnlinearizedMode, 719
 SP_Upper_hysteresis, 388
 SP_Upper_limit, 387
 SP_UpperBound<n>, 916
 SP_UpperLimit, 367, 723, 854, 959, 979
 SP_UpperLimit<n>, 267, 368, 617, 816,
 1435
 SP_UpperWarning, 854, 979
 SP_USBFrameFormat, 1178
 SP_USBProtocol, 1178
 SP_USBSlaveID, 1178
 SP_UsedColorSpace, 1112
 SP_UserMeasProgNumber, 853, 978
 SP_UserName, 1145--1148, 1150
 SP_UserPassword, 1145, 1146, 1148
 SP_UserPasswordHash, 1145, 1146
 SP_UserRoles, 1146
 SP_UserSessionTimeout, 1148
 SP_ValidRange, 219
 SP_Value, 996, 1294, 1304
 SP_VideoAverage, 431, 677, 1116
 SP_VideoMode, 385
 SP_VideoThrAuto, 829
 SP_VideoThrFix, 829
 SP_VideoThrMode, 826
 SP_WaitSpectrumTimeout, 465
 SP_WaitVideoTimeout, 343, 563, 675,
 787, 906
 SP_Watchdog, 458
 SP_WatchdogPeriod, 458
 SP_Weighting, 453
 SP_WhiteCorrFactor1..3, 1276
 SP_WhiteLED, 1278
 SP_WindowMax, 245
 SP_WindowMin, 245
 SP_X, 1113
 SP_X1..16, 475
 SP_Y, 1113
 SP_Z, 1113
 SP_Zero, 388, 478
 timestamp, 96
 TransferData, 95
 TransferDataTS, 96
 variable argument list, 106, 107
 versionStr, 100
 XP_FirmwareFile, 269, 1281, 1442