MKS interface

MKS_PH, MKS_TMP, MKS_ADAPT, MKS_OXY, MKS_COND, MKS_CO2, MKS_OUT, MKS_PH_ISM, MKS_OXY_ISM, MKS_DS Modules

Version	Change *)	Date
0.6	First version (preliminary, Oxy is missing in the presentation layer)	28.06.06
0.61	Additions from the technical description	21.09.06
0.7	Change in the CRC in the data link layer. Addition from the technical description	13.03.07
0.71	MKS_TMP, physical layer interface HE, data history	11.06.07
0.72	Change in error messages. In the case of MKS_PH, add the sensor temperature and temperature-compensated measured values. MKS_Adapt	27.07.07
0.80	MKS_OXY, MKS_COND, EEPROM instead of ROM	09.10.07
1.00	MKS_COND temperature measurement, measured value status	08.01.08
1.10	MKS_TMP temperature correction	23.07.09
1.20	MKS_CO2, MKS_OUT	03.03.11
1.30	MKS_PH_ISM, MK_OXY_ISM, MKS_DS (preliminary)	11.07.12
1.40	MKS_DS (preliminary)	22.11.12

1.50	Sensoface criteria	29.10.13
1.60	Adjustment of oxy sensors for MKS_DS	04.03.15
1.70	Adjustment of redox, oxy and cond sensors for MKS_DS	18.12.15
1.80	Sensor version information for MKS_DS	21.01.16

*) Changes are indicated by a vertical dash on the margin

Knick
Elektronische Messgeräte GmbH & Co. KG Sartorius Stedim Systems GmbH

Berlin, Germany, [date] Melsungen, Germany, [date]

....

Distribution list:	Record 1:	Knick	Filed: PM	Copy: Project manager/support personnel/EWS
	Record 2:	Custome	Sartorius	

The description of the MKS interface is based on the ISO-OSI reference model

1 General Requirements

1.1 Module Overview

Modules		Master base
Module name	Sensor connection	RS-485 32 channels
MKS_PH Glas	Ports	√
MKS_OXY	Ports	√
MKS_COND	Ports	√
MKS_ADAPT	Ports	√
MKS_TMP	Ports	√
MKS_CO2	Ports	√
MKS_OUT	Ports	√
MKS_PH_ISM	Ports	√
MKS_OXY_ISM	Ports	√
MKS_DS	Ports	√

1.2 Galvanically Isolated

- In multi-channel devices, the modules must be gal. isolated from one another and from the master.
- Cut-off voltage 500V (8kV ESD)

2 Physical Layer

2.1 Interface

Function		MKS
HE	Current consumption	< 25 mA MKS_ADAPT < 200 mA MKS_DS < 120 mA
	Voltage range	22.8 … 25.2 V
Communication	Modulation	RS-485, 5V ±5%
	Addressing	5 DIL switch for 32 addresses
	Transmission rate	19200 baud
	Terminating resistor	> 50kΩ

2.2 Pin Assignment

10-pin plug connector

Pin	Name	Description
1	n.c.	n.c.
2	n.c.	n.c.
3	n.c.	n.c.
4	n.c.	n.c.
5	MKS_B	RS-485 signal line B
6	n.c.	n.c.
7	n.c.	n.c.
8	MKS_A	RS-485 signal line A
9	MKS_GND	GND
10	MKS_Vcc	HE

3 Data Link Layer

3.1 Requirements

- Configuration of module address in polling mode (master / slave)
- Data protection for SIL2
- Short communication interval, even at low baud rates
- Temporal decoupling of communication from task processing

3.2 Frame Format

3.2.1 UART

The UART configuration is 8 data bits, no parity bit and 1 stop bit. Within a byte, the MSB is always sent first (standard procedure for UARTs).
The baud rate is 19200 bps.

3.2.2 Byte Structure

In values that consist of more than one byte, the LSB is always sent first (INTEL).

3.2.3 Frame Structure

The master can send frames with commands to the slave as well as frames to query the slave queue.

The slave can send command frame responses back to the master as well as frames with information on the slave queue state.

Value range	Preamble	Delimiter	Byte count	Reference data					Check sum
				Slave address	Slave queue state	Command	Address	Data	
Size / byte	0…9	1	1	4	1	1	2	Bytecountrest	4
Master sends	0xFF	0xFA	7…0xFF	32 bit	-	0…0x1F	0…0xFFFF	0…0xFF	0…0xFFFFFFFF
	0xFF	0xFA	4	32 bit	-	-	-	-	0…0xFFFFFFFF
Slave sends	0xFF	0xFA	7…0xFF	32 bit	-	0x80…0x9F	0…0xFFFF	0…0xFF	0…0xFFFFFFFF
	0xFF	0xFA	5	32 bit	0x80…0x82	-	-	-	0…0xFFFFFFFF

<u>Preamble</u>
Can be used, e.g. with automatic direction change of the RS-485.

<u>Delimiter</u>
Frame begin, the value 0xFA may also occur in other frame ranges and is not reserved for the delimiter.

<u>Byte count</u>
Contains the number of bytes in the reference data field.
Not all slaves support the full byte count up to 0xFF.

<u>Slave address</u>
Only the addressed slave may respond. The slave address contains the bus address configured via the DIL switch in the value range 0...31. If the DIL switches are configured to slave address 0, the slave is also addressable via its serial number. The serial number corresponds to the parameter mk_seriesno of the addressed module as the unique bus address.

Slave queue state
Feedback from the slave frame processing.

Command
Contains the action to be executed. Address and data are interpreted differently depending on the command.

Address
Contains the memory start address starting at which the master wants to have read or write access. RAM and EEPROM use separate address ranges.

Data
All data, including integer and floating point numbers are stored in memory starting with the low byte.

Check sum
Secures the range from the delimiter up to and including the reference data.

```
/*******************************************************************************
// crc_32_calc:  Checksum calculation CRC32/8
//               CRC-32/8 (Castagnoli 1993)
//
// IN call:      mem[]   -> Pointer pointing to data beginning
//               len     -> Number of data bytes starting from mem
// IN Global:    -
//
// OUT Return:   CRC32 as execution result
// OUT Pointer:  -
// OUT Global:   -
//
// Side effects: -
//*****************************************************************************
**    u32 crc_32_calc(u8 mem[], u16 len)
{
  // Local variable declarations u8  i,
  indata;
  u16 n;
  u32 crc_input;

  // Program code
  crc_input = 1; // CRC32 start value is 1, not 0
  for(n=0; n<len; n++){
    // Edit next byte indata =
    mem[n];
    // Link byte bitwise for (i = 0; i
    < 8; i++)
    {
      if (((crc_input & 0x80000000) != 0) != ((indata & 0x80) != 0))
        crc_input = (crc_input << 1) ^ 0xF1922815; // XOR = 1 => Shift + process poly
      else
        crc_input = crc_input << 1;             // XOR = 0 => pure shift
      indata = indata << 1;
    }
  }

  return crc_input;
}
```

3.3 Communications Sequence

After expiry of the timegap_nextframe or timeout_responseframe time, the master can send both commands and queries to the slave queue. If the last command is not yet completed, the master polls the slave with slave queue queries until the command is answered or timegap_command has expired.

The slave must begin the frame response on each correctly received frame (check sum test) within a time window (timegap_response). The response can be the final response frame with result (Data_Return) or feedback from the slave queue (Slavequeuestate). The slave must complete the command within the period specified under timegap_command.

Time	Definition	min	max	Unit
timegap_framebyte	The next byte in a frame must start within this time. Permitted time gap between two bytes within a frame. Period between the end of the last bit and the start of the first bit.	0 0	10 2	tbit ms (respectively smaller value applies)
timegap_responseframe	The slave must start its response frame within this time (minimum time gap serves to change the direction in the master). Period between the last byte from the master and the first byte from the slave.	2	30	ms
timeout_responseframe	The master may be switched to send again if the slave has not started his answer after this time has elapsed. The master can then assume a lost frame. Period between the last byte of the 1st master telegram and the first byte of the 2nd master telegram	40	∞	ms
timegap_nextframe	The master may start a new frame after the slave response frame within this time (minimum time gap serves to change the direction in the slave). Time between the last byte of the slave and the first byte of the master.	2	∞	ms
timegap_command	Permitted time gap between the last byte of the master command frame and the first byte of the slave command response frame. The master queries the slave queue status cyclically during command processing until it gets the command result. After this time has elapsed, the master can assume the slave has malfunctioned.	2	2000	ms

4 Transport Layer

4.1 Requirements

- Download/upload configuration data
- Download measured values
- Download error messages

4.2 Command

4.2.1 Memory Access

Command Byte	Command Source	Memory Access	Address Memory storage	Address Scale interval	Data	Check sum
0x01	Master	read	EEPROM	1 byte	Data_Count	CRC32
0x81	Slave	read	EEPROM	1 byte	Data_Content	CRC32
0x02	Master	read	RAM	1 byte	Data_Count	CRC32
0x82	Slave	read	RAM	1 byte	Data_Content	CRC32
0x11	Master	write	EEPROM	1 byte	Data_Content	CRC32
0x91	Slave	write	EEPROM	1 byte	Data_Return	CRC32
0x12	Master	write	RAM	1 byte	Data_Content	CRC32
0x92	Slave	write	RAM	1 byte	Data_Return	CRC32

4.2.2 Structure of the Data Range

4.2.2.1 Data_Count

Number of bytes to be sent starting from and including the address from slave to master. Data type: u8
Value range (RAM, EEPROM): 0…240

4.2.2.2 Data_Content

Memory contents starting with the low byte. Data type: u8[n]
Value range: 0…0xFF

4.2.2.3 Data_Return

Feedback for write command data type: u8
Value range:

Data_Return	Result	Definition
0x00	ok	Command executed completely
0x01	Warn	Can only write to memory in SUmode
0x02	Warn	Memory address not allowed
0x03	Warn	Byte count not allowed
0x04	Warn	Command not allowed
0x40	Fail	Command cannot be executed

4.2.2.4 Read Command Feedback

On a successful read command, the memory content is returned.
Given an invalid read command or an invalid memory range, the read command is returned without Data_Content.

4.3 Slave Queue

Commands have processing times of different lengths. Status messages are therefore needed for processing in the slave. As long as a command frame is still being processed, the slave can give a busy signal or working answer.

Feedback from the slave frame processing data
type: u8

Slave queue state	Result	Definition
0x80	ready	Slave queue is empty, ready for next command frame
0x81	working	Command frame processing, query the slave queue further until reply frame sent with Data_Return or data are sent
0x82	busy	New command frame cannot be edited, continue querying the slave queue until result ready is sent

5 Presentation Layer

5.1 Requirements

- Module type, serial number, module version no., SW version no., HW version no., options, manufacturer labeling, certificates
- Operating mode (power management, power consumption)
- Self-test trigger (in addition to the cyclical self-test)
- Scaled measured value (V, A, □) with status and numbered (instead of a timestamp)
- Error messages
- Sensor change detection (Tempsensor)
- Configuration data

5.2 Operating Modes

A distinction is made between numerous operating modes (see 5.3.1.6 un_mode).

5.3 Memory Access

Depending on the requirement, the individual parameters are divided into different types of memory. There are clearly defined memory areas and those that are still free. There are application specific (pH, Oxy) and universal parameters (ID, type, version, modes, etc.).

5.3.1 Definitions

5.3.1.1 Class

Depending on the application, not all parameters are supported. A multiple condition (AND or OR condition) for a parameter is possible (for example, in multi-parameter modules).

Class	Definition
Common	Always supported
pH	Only for Sartorius pH and CO2 modules
Redox	Only for Redox or ORP sensors
Oxy	Only for Oxy modules
Cond	Only for Cond modules
Adapter_S	Only for Sartorius adapter modules
Temp_S	Only for Sartorius temperature modules
S	Only for Sartorius modules
CO2_S	Only for Sartorius CO2 modules
OUT_S	Only for Sartorius OUT modules
DS_S	Only for Sartorius digital sensor and ISM modules

5.3.1.2 Parameter

Name of a memory range, can be further divided via a data type (array, struct). Parameters can be read or written individually or connected with a command.

5.3.1.3 Access

Allowable conditions under which the parameter may be used

Access	Definition
r	Read only
rw	Read and write

5.3.1.4 *Address*

Start address of the parameter, byte-oriented. Multi-byte values are ascending addresses (0…0xFFFF).

5.3.1.5 *Data type*

Simple or structured data type (array [], struct {}).
Data types may only be read or written in conjunction with a command. In values that consist of more than one byte, the LSB is always sent first (INTEL).

Data type	Definition
u8	Unsigned 1 byte
u16	Unsigned 2 byte
u32	Unsigned 4 byte
i8	Signed 1 byte
i16	Signed 2 byte
i32	Signed 4 byte
float	IEEE 754 (4 byte)
st_mw64	struct { float value; u8 state; u8 history; i8 res; u8 count; }
st_justpH	struct { float pH; float mV; float tempC; u32 time; u32 state; }
st_justORP	struct { float ORP; float mV; float tempC; u32 time; u32 state; }
st_justOxy	struct { float mbar; float reading; float tempC; u32 time; u32 state; }
st_justCond	struct { float conductivity; float resistance; float tempC; u32 time; u32 state; }

5.3.1.6 *Data unit*

Unit of measure of the parameter in []

Data unit	Data type	Definition
un_oem	u8	Manufacturer labeling 7:
un_identifier	u8	Module type 1: pH_K 3: Cond_K 5: Oxy_K 30: Adapter_S 31: Temp_S 32: CO2_S 33: OUT_S
un_versionHW	u8	Hardware version designation Bit 4…7: Major version (change is dependent on software version) Bit 0…3: Minor version (change does not require software change)
un_versionSW	u16	Software version designation according to NA53 Bit 12...15: Level 1 (hidden) Bit 8...11: Level 2 (major version) Bit 4...7: Level 3 (minor version) bit 0...3: Addition (special
un_variant	u8	Module variant (0 ... 254: depending on un_identifier; 255: undef.) pH_K 11: MK_SARTORIUS Oxy_K 11: MK_SARTORIUS Cond_K 11: MK_SARTORIUS Adapter_S 11: MK_SARTORIUS Temp_S 11: MK_SARTORIUS CO2_S 11: MK_SARTORIUS OUT_S 11: MK_SARTORIUS DS_S 11: MK_SARTORIUS

Data unit	Data type	Definition
un_option	u16	Supported options, binary coded, 0 = no option Bit 0: ISM digital Bit 1…15: Free
un_certificate	u16	Valid certificates, binary coded, 0 = no certificate Bit 0…15: Free
un_seriesno	u32	Module serial number, 0 = undefined
un_mode	u8	Operating mode of the module 2: Switch on self-test (no measuring) 3: Self-test (no measuring)
un_modefct	u16	Toggle additional functions (Set by the master and, if necessary, reset by the slave. Initial value 0) Bit 3: Reset module status restart (un_state.0) Bit 4: Reset module sensor change (un_state.1)
un_state	u16	Module status Bit 0: Restart performed (reset by un_modefct.3) Bit 1: Sensor change detected (reset by un_modefct.4) Bit 2: Temperature sensor 1 is connected (measured value is in the measuring range) Bit 5: Temperature sensor 2 is connected (measured value is in the
un_error	u16	Module error messages that set the measured value status to BAD (Bits indicate the current error, no history) Bit 3: AD converter fault (hardware or reference inaccurate, optional) Bit 4: Module temperature error (module temperature inadmissible, optional) Bit 5: RAM error detected (reset by successful self-test) Bit 6: EEPROM error detected in plant-internal area (with access rsw, reset by successful self-test) Bit 7: EEPROM error detected in the user area (optional) Only parameters that influence the measured value in the module. (with access rw, reset by successful self-test) Bit 8: FLASH error detected (reset by successful self-test) Bit 9: Oscillator error (CPU frequency inaccurate, optional)
un_mw64	st_mw64	st_mw64.value: Measured value (float) st_mw64.state: State of the measured value, data status 5.3.1.7 (Includes messages from un_error, un_state.0 and .1) st_mw64.history: Pre-processing of the measured value, data history 5.3.1.8 st_mw64.res: Resolution of the measured value, data resolution 0 st_mw64.count: Measured value counter, incremented for each new value, begins at 0 after an overflow

Data unit	Data type	Definition
un_cond_ mode	u8	Cond operating mode 0: 2-electrode sensor 1: 4-electrode sensor 255: Automatic selection (not used)
un_cond_error	u16	Cond error messages Bit 0: Sensor is not connected (cable capacitance too high) Bit 1: Sensor polarized
un_ph_sensor	u8	pH sensor selection 0: Analog 2: ISM digital
un_oxy_sensor	u8	Oxy sensor selection 0: Standard 1: Traces 2: Subtraces 3: ISM digital 255: Automatic selection via the module or the sensor
un_justpH	st_just pH	st_justpH.pH: pH value of the buffer or the sample st_justpH.mV: pH voltage [mV] of the buffer or the sample st_justpH.tempC: Temperature [°C] of the buffer or during sampling st_justpH.time: Time point [s] of the adjustment or sampling st_justpH.state: Status of the adjustment in accordance with Hamilton specifications (document "EPHUM014" Ch.2.7.4.1)
un_justORP	st_just ORP	st_justORP.ORP ORP value of the buffer or the sample st_justORP.mV: ORP voltage [mV] of the buffer or the sample st_justORP.tempC: Temperature [°C] of the buffer or during sampling st_justORP.time: Time point [s] of the adjustment or sampling st_justORP.state: Status of the adjustment in accordance with Hamilton specifications (document "ERXUM014" Ch.2.7.4.1)
un_justOxy	st_just Oxy	st_justOxy.mbar: O_2 partial pressure [mbar] during adjustment st_justOxy.reading: Phase angle [°] or current [nA] during adjustment st_justOxy.tempC: Temperature [°C] during adjustment st_justOxy.time: Time point [s] during adjustment st_justOxy.state: Status of the adjustment in accordance with Hamilton specifications (document "ODOUM042" Ch.2.7.4.1) or (document "EDOUM014" Ch.2.7.4.1)
un_justCond	st_just Cond	st_justCond.conductivity: Conductivity of the buffer or the sample st_justCond.resistance: Resistivity [Ω] of the buffer or the sample st_justCond.tempC: Temperature [°C] of the buffer or during sampling st_justCond.time: Time point [s] of the adjustment or sampling st_justCond.state: Status of the adjustment in accordance with Hamilton specifications (document "CONUM014" Ch.2.7.4.1) or (document "CPWUM014" Ch.2.7.4.1)

Data unit	Data type	Definition
un_ds_typ	u8	Sensor type for digital sensors, defined by selector switch 0: no digital sensor 1: Memosens sensor (RS-485 with HE 3VDC) 2: ARC Modbus RTU sensor (RS-485 with HE 24VDC) 3: ISM (onewire with HE 5VDC)
un_ds_variant	u8	Sensor variant for digital sensors 0: no valid sensor 1: pH 2: pH isfet 3: Redox 4: pH+Redox 5: Cond 2pol 6: Cond 4pol 7: Oxy standard 8: Oxy traces 9: Oxy subtraces 10: Oxy optical 11…255: free
un_secTime	u32	Seconds since 01/01/2000 at 00:00:00
un_ds_error	u16	Current digital sensor error messages Bit 0: Digital sensor failure Bit 1: Measured value is unstable (status: BAD) Bit 2: Calibration is unstable Bit 3: HE sensor failure
un_just_function	u8	Carry out adjustment functions Not all functions are available for each sensor variant. The function is started when writing. The current status of the function can be determined by a read-out. Before a new function is started, the last should be completed and the parameter should be 0 or 255 (function complete with PASS or FAIL), otherwise the current function may be aborted.

Function index	ISM		Memosens				Hamilton			
	pH	Oxy	pH	Redox	Oxy	Cond	pH	Redox	Oxy	Cond
0: No function, last function was PASS	√	√	√	√	√	√	√	√	√	√
1: Adjustment acceptance and ACT counter reset	√	√	√	√	√	√				
2: Increment autoclaving counter	√	√					√	√	√	√
3: Reset TTM counter	√	√			√					
4: Reset DLI counter		√								
5: Adjustment with buffer 1 or low pressure					√		√	√	√	√
6: Adjustment with buffer 2 or high pressure					√		√		√	
7: Sampling							√	√	√	√
8: Accept sample adjustment							√	√	√	√
9: Delete sample adjustment							√	√	√	√
10…254: Free										
255: No function, last function was FAIL	√	√	√	√	√	√	√	√	√	√

Input parameters used by data unit "un_just_function”

Process para	Function index	Input parameters used in the RAM	Description of the function
pH	1	mp_just_time mp_just_npoint mp_just_slope	Zero point and slope are automatically accepted.
	5	mp_just_time mp_just_pH	Adjustment with buffer 1. The currently measured temperature and the sensor voltage are used.
	6	mp_just_time mp_just_pH	Adjustment with buffer 2. The currently measured temperature and the sensor voltage are used.
	7	mp_just_time	The sensor saves the current measured value for the subsequent sample adjustment.
	8	mp_just_time mp_just_pH	Accept adjustment with the determined laboratory value
	9		Delete sample adjustment.
Redox	1	mp_just_time mp_just_ORPnpoint	Zero point is automatically accepted.
	5	mp_just_time mp_just_ORP	Adjustment with buffer. The currently measured temperature and the sensor voltage are used.
	7	mp_just_time	The sensor saves the current measured value for the subsequent sample adjustment.
	8	mp_just_time mp_just_ORP	Accept adjustment with the determined laboratory value
	9		Delete sample adjustment.
Oxy	1	mp_just_time mp_just_zeroCurrent mp_just_satCurrent	Zero point and slope are automatically accepted.
	5	mp_just_time	The oxygen content is assumed to be zero. The currently measured temperature and the sensor value are used.
	6	mp_just_time mp_just_airpercent mp_airPressure mp_humidity	The oxygen content corresponds to the parameters. The currently measured temperature and the sensor value are used.
	7	mp_just_time mp_airPressure	The sensor saves the current measured value for the subsequent sample adjustment.
	8	mp_just_time mp_just_saturationO2	Accept adjustment with the determined laboratory value. Air pressure is taken from the time of sampling Humidity RH = 100%
	9		Delete sample adjustment.
Cond	1	mp_just_time mp_just_cellconst	Cell constant is automatically accepted.
	5	mp_just_time mp_just_cond	Adjustment with buffer. The currently measured temperature and the conductivity are used.
	7	mp_just_time	The sensor saves the current measured value for the subsequent sample adjustment.
	8	mp_just_time mp_just_cond	Accept adjustment with the determined laboratory value.
	9		Delete sample adjustment.

Knick

5.3.1.7 Data Status

The status is modified according to the status definition of PROFIBUS: (Only **boldface** status values are used)

Quality		Substatus				Limits		Value	Description
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Decimal	
1	0	x	x	x	x	x	x	128…191	**Good**
1	0	0	0	0	0	0	0	**128**	Measured value ok
0	1	x	x	x	x	x	x	64…127	**Uncertain**
0	1	0	1	1	0	0	0	**88**	Measured value imprecise
0	1	0	0	1	1	0	0	**76**	Measured value imprecise, initial value (after restart)
0	1	0	0	0	1	0	0	**68**	Measured value imprecise, last usable value (sensor failure)
0	0	x	x	x	x	x	x	0…63	**Bad**
0	0	0	0	1	1	0	0	**12**	Measured value bad (device failure)
0	0	0	1	0	0	0	0	**16**	Measured value bad (sensor failure)
0	0	0	1	0	0	0	1	**17**	Measured value bad, lower range excursion
0	0	0	1	0	0	1	0	**18**	Measured value bad, upper range excursion

5.3.1.8 Data History

Describes the previous processing of the measured value. When the bit is set, the definition applies.

Importance of the data history

Bit	Definition
0	Restart was performed (corresponds to state un_state.0)
1	Sensor replacement was detected (corresponds to state un_state.1)
2	Free
3	Free
4	Free
5	Measured value comes from the ISM digital sensor
6	Measured value complies with the accuracy requirements of the specification (no filtering necessary)
7	Measured value is filtered (a measured value jump is relayed with a delay)

5.3.1.9 Data Resolution

Resolution limit of the float point value, gives no indication of the measured value symbol. The resolution in the format i8 delivers the decimal place, which is regarded as the resolution limit.

Example range:

Value (i8)	Description	Example [Ω] resolution
-128	no information	–
-3	0.001 digit	0.001 Ω
-2	0.01 digit	0.01 Ω
-1	0.1 digit	0.1 Ω
0	1 digit	1 Ω
1	10 digits	10 Ω
2	100 digits	100 Ω
3	1000 digits	1000 Ω

The range from ± 127 digits based on the basic unit of measure is possible. The basic unit does not need to be an SI unit (e.g. mV). The basic unit is indicated in the description of the measured value address.

5.3.1.10 Name Groups

Meaning of name groups used

Name header	Definition
un_	Dataunit
en_	Datatype enumeration
st_	Datatype struct
mp_	Module parameter representing the current state of the module but can also be changed by the master
mv_	Module variable representing the current state of the module and cannot be changed by the master
mw_	Module variable representing the current measured value of the sensor and cannot be changed by the master

5.3.2 EEPROM

Class	Parameter	Access	Address[1]	Data type	Data unit	Definition
Common	mk_oem	r	0x0002	u8	un_oem	Manufacturer labeling
	mk_identifier	r	0x0003	u8	un_identifier	Module type
	mk_versionHW	r	0x0004	u8	un_versionHW	Module hardware version
	mk_variant	r	0x0005	u8	un_variant	Module variant
	mk_versionSW	r	0x0006	u16	un_versionSW	Module software version
	mk_versionSWcom	r	0x0008	u16	un_versionSW	Module software version that is still compatible
	mk_option	r	0x000A	u16	un_option	Module options
	mk_certificate	r	0x000C	u16	un_certificate	Valid certificates
	mk_seriesno	r	0x000E	u32	un_seriesno	Module serial number
Cond	mp_cond_mode	rw	0x1200	u8	un_cond_mode	Selection 2/4 El. 0: 2-electrode sensor 1: 4-electrode sensor (Initial value: 0)
Oxy	mp_pol	rw	0x0400	float	mV	Polarization voltage (range: − 1000.0 mV … 1000.0 mV; initial value: − 675.0 mV)
	mp_tkmem	rw	0x0404	float	dimensionless	Membrane correction factor (Range: 0.0 … 10.0; initial value: 1.0)
Temp_S	mp_temp1_offset	rw	0x0400	float	K	Temperature 1 offset (initial value: 0.0)
	mp_temp2_offset	rw	0x0404	float	K	Temperature 2 offset (initial value: 0.0)
	mp_temp1_ factor	rw	0x0408	float	dimensionless	Temperature 1 factor (initial value: 1.0)
	mp_temp2_ factor	rw	0x040C	float	dimensionless	Temperature 2 factor (initial value: 1.0)
CO2_S	mp_CO2_pH_zero	rw	0x0900	float	pH	Zero point pH value in the sensor (initial value:
	mp_CO2_pH_slope	rw	0x0904	float	mV/pH	Slope pH value in the sensor (initial value: 59.2)
	mp_CO2_salinity	rw	0x0908	float	g/kg	Salinity (initial value: 28.0)
	mp_CO2_hydroconc	rw	0x090C	float	mol/l	Bicarbonate conc. (initial value: 0.05)
	mp_CO2_pressure	rw	0x0910	float	mbar	Process pressure (initial value: 1013.0)
	mp_CO2_factor	rw	0x0914	float	dimensionless	CO2 value correction factor (initial value: 1.0)
OUT_S	mp_temp_offset	rw	0x2900	float	K	Temperature offset (initial value: 0.0)
	mp_temp_ factor	rw	0x2904	float	dimensionless	Temperature factor (initial value: 1.0)
Oxy && DS_S	mp_init_airPressure	rw	0x0500	float	mbar	Initial air pressure value (initial value: 1013.0)
	mp_init_humidity	rw	0x0504	float	%rH	Initial humidity (initial value: 50.0)
	mp_refTemperature	rw	0x0508	float	°C	Reference temperature (initial value: 25.0)
	mp_saltCorrection	rw	0x050C	float	g/kg	Salinity correction factor (initial value: 0.0)

[1] Organize data types 16 bitwise, so that the data types >1 byte start at even addresses.

5.3.3 RAM

Class	Parameter	Access	Address	Data type	Data unit	Definition
Common	mp_mode	rw	0x0000	u8	un_mode	Preferred operating mode of the module
	mp_modefct	rw	0x0002	u16	un_modefct	Start module functions
	mv_state	r	0x0004	u16	un_state	Current module status
	mv_error	r	0x0006	u16	un_error	Current error messages of the module
	mv_mode	r	0x0008	u8	un_mode	Current operating mode of the module
pH, Redox	mw_tempR	r	0x0240	st_mw64	un_mw64(Ω, °C)	Analog sensor: Sensor temperature resistivity ISM sensor: Sensor temperature in degrees
	mw_phU	r	0x0248	st_mw64	un_mw64(mV)	pH voltage
	mw_orpU	r	0x0250	st_mw64	un_mw64(mV)	ORP voltage
	mw_glassZ	r	0x0258	st_mw64	un_mw64(Ω)	Glass electrode impedance
	mw_refZ	r	0x0260	st_mw64	un_mw64(Ω)	Ref. electrode impedance
	mp_ph_sensor	rw	0x0280	u8	un_ph_sensor	Select sensor type pH (initial value: 255)
S&&pH, Redox	mw_tempC	r	0x0400	st_mw64	un_mw64(°C)	Sensor temperature in degrees Celsius
	mw_phU_TC	r	0x0408	st_mw64	un_mw64(mV)	pH voltage temperature comp.
	mw_glassZ_TC	r	0x0410	st_mw64	un_mw64(Ω)	Glass el. impedance temperature comp.
	mw_refZ_TC	r	0x0418	st_mw64	un_mw64(Ω)	Ref. el. impedance temperature comp.
Cond	mv_cond_error	r	0x100E	u16	un_cond_error	Current cond error messages
	mw_tempR	r	0x1010	st_mw64	un_mw64(Ω)	Sensor temperature resistivity
	mw_cond	r	0x1018	st_mw64	un_mw64(S)	Conductance
	mw_resist	r	0x1020	st_mw64	un_mw64(Ω)	Resistivity
S&&Cond	mw_tempC	r	0x1200	st_mw64	un_mw64(°C)	Sensor temperature (Pt100/Pt1000)
Oxy	mw_tempR	r	0x0234	st_mw64	un_mw64(Ω, °C)	Analog sensor: Sensor temperature resistivity ISM sensor: Sensor temperature in degrees
	mw_i	r	0x023C	st_mw64	un_mw64(nA)	Sensor current
	mw_imp	r	0x0244	st_mw64	un_mw64(Ω)	Sensor impedance
	mw_tki	r	0x0254	st_mw64	un_mw64(nA)	Temperature-compensated sensor current
	mw_tempC	r	0x025C	st_mw64	un_mw64(°C)	Temperature (NTC 22kΩ specified)
	mp_oxy_sensor	rw	0x0271	u8	un_oxy_sensor	Select sensor type Oxy (initial value: 255)
Temp_S	mw_temp1	r	0x0224	st_mw64	un_mw64(°C)	Temperature 1
	mw_temp2	r	0x022C	st_mw64	un_mw64(°C)	Temperature 2
	mw_tempR1	r	0x0234	float	Ω	Sensor temperature resistivity 1
	mw_tempR2	r	0x0238	float	Ω	Sensor temperature resistivity 2
Adapter_S	mw_ain	r	0x020C	st_mw64	un_mw64(mV)	Input supply voltage
CO2_S	mw_CO2_pH	r	0x0800	st_mw64	un_mw64(pH)	pH value in the CO2 sensor
	mw_CO2_si_per	r	0x0808	st_mw64	un_mw64(%)	CO2 saturation
	mw_CO2_conc_mgl	r	0x0810	st_mw64	un_mw64(mg/l)	CO2 concentration
	mw_CO2_pp_hpa	r	0x0818	st_mw64	un_mw64(hPa)	CO2 partial pressure
OUT_S	mw_tempR	r	0x2900	st_mw64	un_mw64(Ω)	Sensor temperature resistivity
	mw_tempC	r	0x2908	st_mw64	un_mw64(°C)	Sensor temperature
	mp_out_nA	rw	0x2A00	float	nA	Current output for value pO2
	mp_out_mV	rw	0x2A04	float	mV	Voltage output for value pH

Class	Parameter	Access	Address	Data type	Data unit	Definition
pH, Redox &&DS_S	mw_pH	r	0x0420	st_mw64	un_mw64(pH)	pH value
	mv_just_time	r	0x0428	u32	un_secTime	Sensor-internal adjustment time point
	mv_just_asymmetry	r	0x042C	float	mV	Sensor-internal asymmetry potential ISFET
	mv_just_ORPnpoint	r	0x0430	float	mV	Sensor-internal Redox zero point
	mv_just_npoint	r	0x0434	float	pH	Sensor-internal zero point
	mv_just_slope	r	0x0438	float	mV/pH	Sensor-internal slope
	mv_just_buffer1	r	0x043C	st_justpH	un_justpH	Adjustment pH Buffer 1
	mv_just_buffer2	r	0x0450	st_justpH	un_justpH	Adjustment pH Buffer 2
	mv_just_product	r	0x0464	st_justpH	un_justpH	Sample adjustment pH
	mv_just_bufferORP	r	0x0478	st_justORP	un_justORP	Adjustment ORP buffer
	mv_just_productORP	r	0x048C	st_justORP	un_justORP	Sample adjustment ORP
			0x04A0			
		n	0x0500	u8		Free
	mp_just_function	rw	0x0501	u8	un_just_function	Controlling adjustment function
	mp_just_time	rw	0x0502	u32	un_secTime	Sensor adjustment time point
	mp_just_asymmetry	rw	0x0506	float	mV	Sensor asymmetry potential ISFET
	mp_just_ORPnpoint	rw	0x050A	float	mV	Sensor Redox zero point
	mp_just_npoint	rw	0x050E	float	pH	Sensor zero point
	mp_just_slope	rw	0x0512	float	mV/pH	Sensor slope
	mp_just_pH	rw	0x0516	float	pH	Sensor setpoint pH
	mp_just_ORP	rw	0x051A	float	mV	Sensor setpoint ORP
			0x051E			
Oxy && DS_S	mw_oxy_partialPressure	r	0x0400	st_mw64	un_mw64(mbar)	O_2 partial pressure
	mw_oxy_saturationO2	r	0x0408	st_mw64	un_mw64(%sat)	O_2 saturation (in rel. to air)
	mw_oxy_gasvolConc	r	0x0410	st_mw64	un_mw64(vol%)	O_2 volume concentration in gases
	mw_oxy_liquidConc	r	0x0418	st_mw64	un_mw64(mg/l)	O_2 concentration in liquids
	mv_just_time	r	0x0420	u32	un_secTime	Sensor-internal adjustment time point
	mv_just_zeroCurrent	r	0x0424	float	nA	Sensor-internal zero current
	mv_just_satCurrent	r	0x0428	float	nA	Sensor-internal current at 100%rH in air
	mv_just_phase	r	0x042C	float	Grad	Sensor-internal phase angle
	mv_just_sensitivity	r	0x0430	float	-	Sensor-internal sensitivity (Stern Volmer)
	mv_just_pressureLow	r	0x0434	st_justOxy	un_justOxy	Adjustment at low pressure
	mv_just_pressureHigh	r	0x0448	st_justOxy	un_justOxy	Adjustment at high pressure
	mv_just_product	r	0x045C	st_justOxy	un_justOxy	Sample adjustment
			0x0470			
	mp_airPressure	rw	0x0500	float	mbar	Air pressure value. Default: mp_init_airPressure
	mp_humidity	rw	0x0504	float	%rH	Humidity. Default: mp_init_humidity
		n	0x0508	u8		Free
	mp_just_function	rw	0x0509	u8	un_just_function	Controlling adjustment function
	mp_just_time	rw	0x050A	u32	un_secTime	Sensor adjustment time point
	mp_just_zeroCurrent	rw	0x050E	float	nA	Sensor zero current
	mp_just_satCurrent	rw	0x0512	float	nA	Sensor current at 100% RH in air
	mp_just_saturationO2	rw	0x0516	float	%sat	Sensor setpoint O_2 saturation (in rel. to air)
	mp_just_airpercent	rw	0x051A	float	%	Air proportion during adjustment. Default: 100%
			0x051E			
Cond && DS_S	mw_conductivity	r	0x1400	st_mw64	un_mw64(S/m)	Conductivity
	mw_massicResist	r	0x1408	st_mw64	un_mw64(□m)	Resistivity
	mv_just_time	r	0x1410	u32	s	Sensor-internal adjustment time point
	mv_just_cellconst	r	0x1414	float	1/cm	Sensor-internal cell constant
	mv_just_buffer	r	0x1418	st_justCond	un_justCond	Adjustment buffer
	mv_just_product	r	0x142C	st_justCond	un_justCond	Sample adjustment
			0x1440			
		n	0x1600	u8		Free
	mp_just_function	rw	0x1601	u8	un_just_function	Controlling adjustment function
	mp_just_time	rw	0x1602	u32	un_secTime	Sensor adjustment time point
	mp_just_cellconst	rw	0x1606	float	1/cm	Sensor cell constant
	mp_just_cond	rw	0x160A	float	S/m	Sensor setpoint
			0x160E			

Class	Parameter	Access	Address	Data type	Data unit	Definition
DS_S	mv_ds_typ	r	0x3002	u8	un_ds_typ	Sensor type, defined by selector switch
	mv_ds_variant	r	0x3003	u8	un_ds_variant	Sensor variant
	mv_ds_oem	r	0x3004	u8[16]	ASCII	Sensor manufacturer
	mv_ds_description	r	0x3014	u8[16]	ASCII	Sensor designation
	mv_ds_orderNumber	r	0x3024	u8[16]	ASCII	Sensor order number
	mv_ds_serialNumber	r	0x3034	u8[16]	ASCII	Sensor serial number
	mv_ds_fabricationTime	r	0x3044	u32	un_secTime	Sensor time of manufacture
	mv_ds_versionSW	r	0x3048	u8[16]	ASCII	Sensor software version
	mv_ds_versionHW	r	0x3058	u8[16]	ASCII	Sensor hardware version
	mv_ds_operatingTime	r	0x3100	u32	s	Operating period
	mv_ds_wear	r	0x3104	float	%	Wear, 100% = end of life reached (not Memosens Cond). Deviations due to internal rounding are possible.
	mv_ds_CIP_cycle	r	0x3108	u16	Meter reading	CIP cycles (see un_just_function)
	mv_ds_SIP_cycle	r	0x310A	u16	Meter reading	SIP cycles (see un_just_function)
	mv_ds_autoclav_cycle	r	0x310C	u16	Meter reading	Autoclave cycles (see un_just_function)
	mv_ds_DLI	r	0x310E	u32	s	Residual life period, DLI (see un_just_function)
	mv_ds_TTM	r	0x3112	u32	s	Period until next maintenance, TTM (see un_just_function)
	mv_ds_ACT	r	0x3116	u32	s	Period until next adjustment, ACT (see un_just_function)
	mv_ds_error	r	0x311A	u16	un_ds_error	Current digital sensor error messages

6 Application Layer

6.1 Requirements

- Realization of a master slave bus structure with up to 32 slaves
- Each slave is uniquely addressable

6.2 Bus Structure

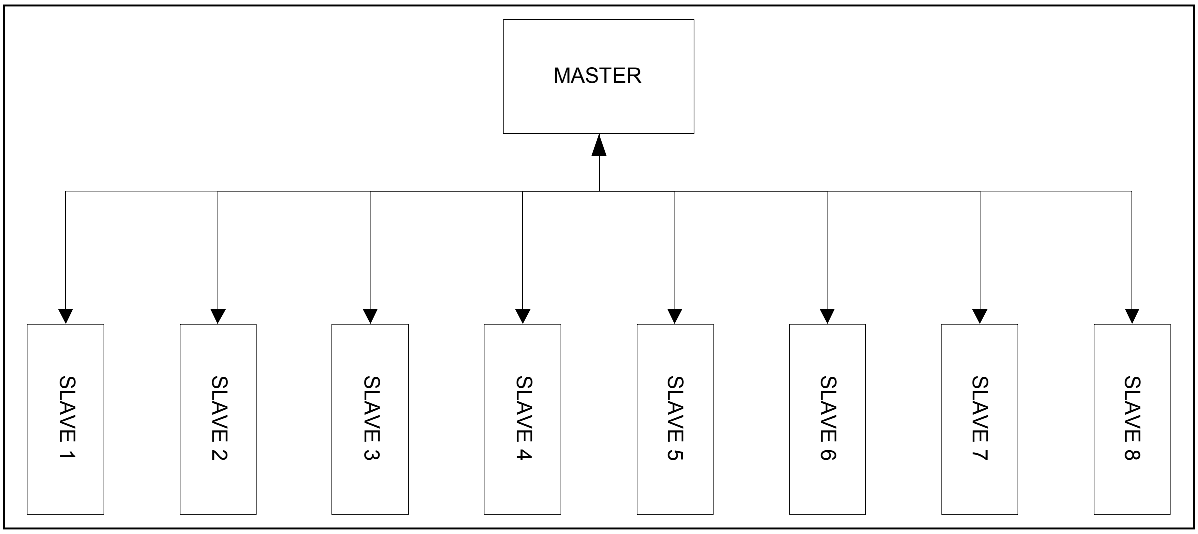

Figure 1: Master slave bus (expandable up to 32 participants)

6.2.1 Bus Operation

Each slave is assigned a unique address. If two slaves are configured to the same address, communication is not possible as both respond simultaneously.

Only the addressed slave may respond to a master frame. The slave address corresponds to the value set via the DIP switch in the value range 0 ... 31.

If the DIP switches are set to address 0, the slave is also addressable via its serial number. The serial number corresponds to the parameter mk_seriesno of the addressed module as the unique bus address. The serial number can be read directly on the slave on the type label.

7 Sensoface

7.1 General Information

- For the current calculation of Sensoface, parameters from the MKS interface and from the sensor calibration are required cyclically, which must make control available.
- In order to always provide a current Sensoface for a sensor, the Sensoface calculations made in a framework must be carried out cyclically. The syntax corresponds to that of the programming language "C".
- The respective Sensoface assessment only applies to the sensor types listed in parentheses. If a sensor type is not listed, there is no assessment for this property. The Sensoface assessments listed correspond to those of Stratos Evo series. The assessments were adopted for the Hamilton sensors.
- The priority of the Sensoface assessments corresponds to the sequence of the Sensoface calculations carried out in the frame. The lowest priority is evaluated first and the highest priority last.

7.1.1 General Constants

Name of the constants	Description
FINE, MEDIUM, BAD	Possible Sensoface states
PH_NOM_NPKT = 7.0;	Nominal pH zero point in pH
PH_NOM_SLOPE = 59.2;	Nominal pH slope in mV/pH
STD, TRACE, SUBTR	Possible oxy sensor types
OXYSAT_NOM[STD] = -50.0;	Nominal oxy sensor current in air, standard in nA
OXYSAT_NOM[TRACE] = -375.0;	Nominal oxy sensor current in air, traces in nA
OXYSAT_NOM[SUBTR] = -5000.0;	Nominal oxy sensor current in air, subtraces in nA
OXYSAT_MIN[STD] = 2.2 * OXYSAT_NOM[STD];	Minimum oxy sensor current in air, standard in nA
OXYSAT_MIN[TRACE] = 1.4 * OXYSAT_NOM[TRACE];	Minimum oxy sensor current in air, traces in nA
OXYSAT_MIN[SUBTR] = 1.6 * OXYSAT_NOM[SUBTR];	Minimum oxy sensor current in air, subtraces in nA
OXYSAT_MAX[STD] = 0.6 * OXYSAT_NOM[STD];	Maximum oxy sensor current in air, standard in nA
OXYSAT_MAX[TRACE] = 0.6 * OXYSAT_NOM[TRACE];	Maximum oxy sensor current in air, traces in nA
OXYSAT_MAX[SUBTR] = 0.5 * OXYSAT_NOM[SUBTR];	Maximum oxy sensor current in air, subtraces in nA
OXYZERO_MAX[STD] = 1.0;	Maximum oxy sensor current zero point, standard in nA
OXYZERO_MAX[TRACE] = 1.0;	Maximum oxy sensor current zero point, traces in nA
OXYZERO_MAX[SUBTR] = 3.0;	Maximum oxy sensor current zero point, subtraces in nA
COND_NOM_CELL = 1.0;	Nominal Cond cell factor in 1/cm
COND_CONCTEMP_MIN = 0.0;	Lower temperature limit of the concentration tables in °C
COND_CONCTEMP_MAX = 100.0;	Upper temperature limit of the concentration tables in °C

7.1.2 General Variables

Name of the variables	Description
sensoface	Sensoface with the states FINE, MEDIUM, BAD
act_time	Current time in the format un_secTime
cal_timespan	Calibration interval in seconds
cal_response	Configuration period in seconds
cip_cyclelimit	Max. count of cleaning cycles
sip_cyclelimit	Max. count of sterilization cycles
autoclav_cyclelimit	Max. count of autoclave cycles
oxysensor	Oxy sensor type with the states STD, TRACE, SUBTR
x, gcal, rcal	Local variable in the format float
mp_..., mw_..., mv_...	Variables from the MKS interface specification

7.1.3 General Functions

Function name	Description
abs(x);	Returns the value of x in the format float
max(x, y);	Returns the higher value in float format
bad(x, y);	Returns the state of the worst Sensoface

7.2 pH Sensors

First Sensoface assessment. Is always executed and defines the Sensoface default value.

```
sensoface = FINE;    // default
```

7.2.1 Zero Point and Slope (All Sensors)

Calibration of the pH glass electrode

Calibration range asymmetry potential: (PH_NOM_NPKT 1.0–)…(PH_NOM_NPKT + 1.0)

Calibration range slope: (PH_NOM_SLOPE * 0.8)…(PH_NOM_SLOPE * 1.03)

```
x = abs( mv_just_npoint – PH_NOM_NPKT ) +
    abs(( mv_just_slope – PH_NOM_SLOPE * 0.94 ) / 20.0 );
if( x >= 0.8 )then sensoface = MEDIUM;
if( x > 1.0 )then sensoface = BAD;
```

7.2.2 SIP Cycles (ISM, Memosens, Hamilton)

The sterilization cycles are counted by the sensor and compared with the parameter "sip_cyclelimit" defined by the user.

```
if( mv_ds_SIP_cycle == sip_cyclelimit )then sensoface = bad( sensoface, MEDIUM );
if( mv_ds_SIP_cycle > sip_cyclelimit )then sensoface = BAD;
```

7.2.3 CIP Cycles (ISM, Hamilton)

The cleaning cycles are counted by the sensor and compared with the parameter "cip_cyclelimit" defined by the user.

```
If( mv_ds_CIP_cycle == cip_cyclelimit )then sensoface = bad( sensoface, MEDIUM );
if( mv_ds_CIP_cycle > cip_cyclelimit )then sensoface = BAD;
```

7.2.4 Autoclave Cycles (ISM, Hamilton)

The autoclaving cycles are counted by the user in the sensor and compared with the parameter "autoclav_cyclelimit" defined by the user.

```
if( mv_ds_autoclav_cycle == autoclav_cyclelimit )then sensoface = bad( sensoface, MEDIUM );
if( mv_ds_autoclave_cycle > autoclav_cyclelimit )then sensoface = BAD;
```

7.2.5 Wear (Memosens, ISM, Hamilton)

The wear is determined from the operating time of the sensor as a function of the operating temperature and the measured value.

```
x = mv_ds_wear / 100%;
if( x >= 0.8 )then sensoface = bad( sensoface, MEDIUM );
if( x > 1.0 )then sensoface = BAD;
```

7.2.6 Calibration Interval Without ACT (All Sensors)

The calibration interval parameter "cal_timespan” defined by the user represents the max. period of validity of the last calibration.

To determine the Sensoface, the elapsed time since the last calibration is calculated based on the current time and the timestamp "mv_just_time" and assessed in relation to the calibration interval.

```
x = ( act_time - mv_just_time ) / cal_timespan;
if( x >= 0.8 )then sensoface = bad( sensoface, MEDIUM );
if( x > 1.0 )then sensoface = BAD;
```

7.2.7 Calibration Interval with ACT (ISM)

The calibration interval is specified at 10h operating time and represents the max. period of validity of the last calibration.

To determine the Sensoface, the current remaining duration is considered in relation to the calibration interval for the Sensoface assessment.

```
x = mv_ds_ACT / ( 10 * 60 * 60 );
if( x <= 0.2 )then sensoface = bad( sensoface, MEDIUM );
if( x == 0.0 )then sensoface = BAD;
```

7.2.8 Time To Maintenance – TTM (ISM)

The maintenance interval parameter defined by the sensor is 20 days and represents the maximum period for the maintenance interval.

To determine the Sensoface, the elapsed operating time of the sensor since the last maintenance is calculated from the variable "mv_ds_TTM” and compared to the maintenance interval.

```
x = mv_ds_TTM / ( 20 * 24 * 60 * 60 );
if( x <= 0.2 )then sensoface = bad( sensoface, MEDIUM );
if( x == 0.0 )then sensoface = BAD;
```

7.2.9 Response Time (All Sensors)

Assessment of the response time that the sensor requires for calibration in the buffer. This does not apply to the sample adjustment and is not stored in the digital sensor.

The configuration period "cal_response” provides stability monitoring of the measured value during calibration in seconds.

Start of cal_response: Start of calibration

End of cal_response: Measured value drift is smaller than 0.5mV over 10s.

```
if( cal_response > 36 )then sensoface = bad( sensoface, MEDIUM );
if( cal_response > 72 )then sensoface = BAD;
```

7.2.10 Sensocheck Reference Electrode – if Present on your Model (Analog, ISM, Hamilton)

For Sensocheck reference there are variable limits (rcal / 1.6) ... (rcal / 2 + 100kΩ) that are determined during calibration. To do this, the temperature-compensated impedance of the reference electrode (rcal = mw_refZ_TC.value) is determined during calibration and used as a reference until the next calibration.

```
If(( mw_refZ_TC.value  < ( rcal / 1.6 )) || ( mw_refZ_TC.value > ( rcal / 2 + 100e3 )))then
  sensoface = BAD;
```

7.2.11 Sensocheck Glass Electrode (Analog, ISM, Hamilton)

For Sensocheck glass there are fixed limits 10 MΩ … 1000 MΩ (25 °C) and variable limits (gcal / 3.5) ... (gcal * 3.5) that are determined during calibration. To do this, the temperature-compensated impedance of the glass electrode (gcal = mw_glassZ_TC.value) is determined during calibration and used as a reference until the next calibration.

```
If(( mw_glassZ_TC.value  < ( gcal / 3.5 )) || ( mw_glassZ_TC.value  > ( gcal * 3.5 )))then
  sensoface = BAD;
If(( mw_glassZ_TC.value  < 10e6 ) || ( mw_glassZ_TC.value  > 1e9 ))then
  sensoface = BAD;
```

7.2.12 Error Messages (All Sensors)

Error messages such as 'module failure', 'not a valid sensor', and 'digital sensor failure' lead to a poor Sensoface.

```
If( mv_error != 0 )then sensoface = BAD;
If( mk_identifier == 1 ){              // Analog,
  ISM If( (mv_state & 0x40) != 0 ){        // ISM
    if( mv_ds_typ != 3 )then sensoface = BAD;
    If(( mv_ds_variant  != 1 ) && ( mv_ds_variant  != 4 ))then sensoface = BAD;
    If( mv_ds_error != 0 )then sensoface = BAD;
  }
}
If( mk_identifier == 34 ){             // Memosens, Hamilton
  if(( mv_ds_typ != 1 ) && ( mv_ds_typ !=2 ))then sensoface = BAD;
  If(( mv_ds_variant  != 1 ) && ( mv_ds_variant  != 4 ))then sensoface = BAD;
  If( mv_ds_error != 0 )then sensoface = BAD;
}
```

7.3 Oxy Sensors

First Sensoface assessment. Is always executed and defines the Sensoface default value.

```
sensoface = FINE;    //
oxysensor =          // Sensor selection with STD, TRACE,
```

7.3.1 SIP Cycles (ISM, Memosens, Hamilton)

The sterilization cycles are counted by the sensor and compared with the parameter "sip_cyclelimit" defined by the user.

```
if( mv_ds_SIP_cycle == sip_cyclelimit )then sensoface = bad( sensoface, MEDIUM );
if( mv_ds_SIP_cycle > sip_cyclelimit )then sensoface = BAD;
```

7.3.2 CIP Cycles (ISM, Hamilton)

The cleaning cycles are counted by the sensor and compared with the parameter "cip_cyclelimit" defined by the user.

```
If( mv_ds_CIP_cycle == cip_cyclelimit )then sensoface = bad( sensoface, MEDIUM );
if( mv_ds_CIP_cycle > cip_cyclelimit )then sensoface = BAD;
```

7.3.3 Autoclave Cycles (ISM, Hamilton)

The autoclaving cycles are counted by the user in the sensor and compared with the parameter "autoclav_cyclelimit" defined by the user.

```
if( mv_ds_autoclav_cycle == autoclav_cyclelimit )then sensoface = bad( sensoface, MEDIUM );
if( mv_ds_autoclave_cycle > autoclav_cyclelimit )then sensoface = BAD;
```

7.3.4 Wear (Memosens, ISM, Hamilton)

The wear is determined from the operating time of the sensor as a function of the operating temperature and the measured value.

```
x = mv_ds_wear / 100%;
if( x >= 0.8 )then sensoface = bad( sensoface, MEDIUM );
if( x > 1.0 )then sensoface = BAD;
```

7.3.5 DLI (ISM)

The sensor counts the remaining lifetime.

```
if( mv_ds_DLI <= 0 )then sensoface = BAD;
```

7.3.6 Response Time (All Sensors)

Assessment of the response time that the sensor requires for calibration in air. The configuration period "cal_response" provides stability monitoring of the measured value during calibration in seconds.

Start of cal_response: Start of calibration

End of cal_response: Measured value drift is smaller than 1/256 of the measured value over 60s.

```
if( cal_response >= 300 )then sensoface = bad( sensoface, MEDIUM );
if( cal_response > 600 )then sensoface = BAD;
```

7.3.7 Sensor Calibration at the Zero Point (All Sensors)

Calibration in an oxygen-free environment.

The calibration limits are 30% larger than the OXYZERO_MAX [] limit to 0.0 nA.

```
x = abs( mv_just_zeroCurrent / OXYZERO_MAX[oxysensor] );
if( x >= 0.8 )then sensoface = MEDIUM;
if( x > 1.0 )then sensoface = BAD;
```

7.3.8 Sensor Calibration in Air (All Sensors)

Calibration in air at 25 °C, 1013.25 mbar, rH 100%.

The calibration limits are each 30% larger than the OXYSAT_MIN [] and OXYSAT_MAX [] limits against OXYSAT_NOM [].

```
x = max( abs(( OXYSAT_NOM[oxysensor] - mv_just_satCurrent ) /
               ( OXYSAT_NOM[oxysensor] - OXYSAT_MIN[oxysensor] )),
          abs(( OXYSAT_NOM[oxysensor] - mv_just_satCurrent ) /
               ( OXYSAT_NOM[oxysensor] - OXYSAT_MAX[oxysensor] )));
if( x >= 0.8 )then sensoface = MEDIUM;
if( x > 1.0 )then sensoface = BAD;
```

7.3.9 Calibration Interval Without ACT (All Sensors)

The calibration interval parameter "cal_timespan" defined by the user represents the max. period of validity of the last calibration.

To determine the Sensoface, the elapsed time since the last calibration is calculated based on the current time and the timestamp "mv_just_time" and assessed in relation to the calibration interval.

```
x = ( act_time - mv_just_time ) / cal_timespan;
if( x >= 0.8 )then sensoface = bad( sensoface, MEDIUM );
if( x > 1.0 )then sensoface = BAD;
```

7.3.10 Calibration Interval with ACT (ISM)

The calibration interval is specified at 336h operating time and represents the max. period of validity of the last calibration.

To determine the Sensoface, the current remaining duration is considered in relation to the calibration interval for the Sensoface assessment.

```
x = mv_ds_ACT / ( 336 * 60 * 60 );
if( x <= 0.2 )then sensoface = bad( sensoface, MEDIUM );
if( x == 0.0 )then sensoface = BAD;
```

7.3.11 Time To Maintenance – TTM (ISM)

The maintenance interval parameter defined by the sensor is 366 days and represents the maximum period for the maintenance interval.

To determine the Sensoface, the elapsed operating time of the sensor since the last maintenance is calculated from the variable "mv_ds_TTM" and compared to the maintenance interval.

```
x = mv_ds_TTM / ( 366 * 24 * 60 * 60 );
if( x <= 0.2 )then sensoface = bad( sensoface, MEDIUM );
if( x == 0.0 )then sensoface = BAD;
```

7.3.12 Error Messages (All Sensors)

Error messages such as ‘module failure’, ‘not a valid sensor’, and ‘digital sensor failure’ lead to a poor Sensoface.

```
If( mv_error != 0 )then sensoface = BAD;
If( mk_identifier == 1 ){             // Analog,
  ISM If( (mv_state & 0x40) != 0 ){        // ISM
    if( mv_ds_typ != 3 )then sensoface = BAD;
    If(( mv_ds_variant  < 7 ) || ( mv_ds_variant  > 9 ))then sensoface = BAD;
    If( mv_ds_error != 0 )then sensoface = BAD;
  }
}
If( mk_identifier == 34 ){           // Memosens, Hamilton
  if(( mv_ds_typ != 1 ) && ( mv_ds_typ !=2 ))then sensoface = BAD;
  If(( mv_ds_variant  < 7 ) || ( mv_ds_variant  > 9 ))then sensoface = BAD;
  If( mv_ds_error != 0 )then sensoface = BAD;
}
```

7.4 Cond. Sensors

First Sensoface assessment. Is always executed and defines the Sensoface default value.

```
sensoface = FINE;    // default
```

7.4.1 Cell Factor for Sensor with Unknown Nominal Cell Factor (All Sensors)

The cell factor is assessed the same for 2-electrode and 4-electrode sensors.

```
If(( mv_just_cellconst  < 0.005 ) || (mv_just_cellcon > 19.9999 ))then sensoface = BAD;
```

7.4.2 Cell Factor for Sensor with Known Nominal Cell Factor (All Sensors)

The cell factor is assessed as a function of the nominal cell factor.

```
x = mv_just_cellconst  / COND_NOM_CELL;
If(( x  < 0.5 ) || ( x > 2.0 ))then sensoface = BAD;
```

7.4.3 Concentration Tables (All Sensors)

The sensor temperature affects the evaluation of concentration tables.

If the sensor temperature is outside the range specified in the tables, this will affect the Sensoface.

```
If(( mw_tempC.value < COND_CONCTEMP_MIN ) ||
   ( mw_tempC.value > COND_CONCTEMP_MAX ))then sensoface = BAD;
```

7.4.4 Measured Value Limits (All Sensors)

If, for analog sensors, the measured value is above the conductance limit of 3500mS, this will have an effect on the measured value state. In digital sensors, the sensor itself specifies the measured value limit that acts on the measured value state.

```
If(( mw_cond.state <= 63 ) || ( mw_conductivity.state <= 63 ))then sensoface = BAD;
```

7.4.5 Polarization Recognition (All Sensors)

Polarization recognition is assessed the same for 2-electrode and 4-electrode sensors.

```
If( mv_cond_error.bit1 != 0 )then sensoface = BAD;
```

7.4.6 Error Messages (All Sensors)

Error messages such as ‘module failure’, ‘not a valid sensor’, and ‘digital sensor failure’ lead to a poor Sensoface.

```
If( mv_error != 0 )then sensoface = BAD;
If( mk_identifier == 34 ){            // Memosens, Hamilton
  if(( mv_ds_typ != 1 ) && ( mv_ds_typ !=2 ))then sensoface = BAD;
  If(( mv_ds_variant  != 5 ) && ( mv_ds_variant  != 6 ))then sensoface = BAD;
  If( mv_ds_error != 0 )then sensoface = BAD;
}
```