

Documentation and Sample Programmes for the Windows Dynamic Link Libraries **GMH3X32E.DLL** as of Version 2.07

CONTENTS

1	GENERAL	2
1.1	SCOPE OF SUPPLY	2
1.2	SOFTWARE LICENCE AGREEMENT	2
1.3	SOFTWARE REGISTRATION.....	2
1.4	SYSTEM REQUIREMENTS	2
2	FUNCTIONS OF THE GMH3000-DLL.....	3
2.1	ABOUT THE DLL IN GENERAL.....	3
2.2	DATA COMMUNICATION.....	3
2.3	BUS ADDRESSES.....	3
2.4	PRIORITY MESSAGES.....	3
2.5	LANGUAGE-OFFSET.....	3
2.6	GMH_OPENCOM.....	4
2.7	GMH_CLOSECOM	4
2.8	GMH_TRANSMIT.....	4
2.8.1	Error -36 at GMH_Transmit-function 'GMH_GetValue'	4
2.8.2	Function Calls of GMH_Transmit.....	5
2.9	GMH_GETTYPE.....	5
2.10	GMH_GETUNIT.....	6
2.11	GMH_GETMEASUREMENT.....	6
2.12	GMH_GETERRORMESSAGERET.....	6
2.13	GMH_GETSTATUSMESSAGE	6
2.14	GMH_GETERRORMESSAGEFL.....	6
2.15	GMH_GETVERSIONNUMBER	6
2.16	GETADDITIONALDELAY	7
2.17	SETADDITIONALDELAY	7
2.18	READ OUT THE CYCLIC LOGGER MEMORY: COMMAND GMH_READLOGGER.....	7
2.19	READ OUT THE MANUAL LOGGER MEMORY (STOR)	8
3	ERROR CODES OF EASYBUS-DLL	9
4	EXTENDED ERROR CODES (ON ERROR -36 USE GETERRORMESSAGEFL)	10
5	UNITS.....	11
6	MEASURMENT.....	12
7	PRIORITY MESSAGES	13
8	CONFIGURATION FLAGS.....	14

1 General

1.1 Scope of Supply

The documentation includes a programme disk, which contains:

- | | |
|----------------|---|
| - GMH3x32E.dll | 32bit Windows Dynamic Link Library |
| - GMH3x32E.lib | 32bit Windows Dynamic Link Library- declarations for C-applications |
| - subdirectory | Sample programme for several program languages |

1.2 Software Licence Agreement

Please read carefully the software licence agreement at the disk package. By opening the package you agree with the mentioned terms.

1.3 Software Registration

Get Your software registrated! It pays!

Registrated customers do profit from the following advantages:

Technical Support:

Only registrated customers get support concerning technical questions about the software.

Low Priced Update Offers:

Only registrated customers get our low priced update offers!

Up To Date Information:

Our products palette is constantly growing. Only registrated customers will be informed about new or improved products.

To be always up to date!

Your Opinion Is Important To Us!

Let us know your proposals of improvement. Your proposal may already be included in the next product version!

A registration form is included in the scope of supply. Please fill out completely and send or fax it to us.

1.4 System Requirements

- IBM compatible PC pentium or higher
- at least 1 MB of free harddisk space, cd drive
- at least 128 MB RAM
- Windows 2000 or higher
- a free serial interface (COM1...255)
- Measuring instrument of the GMH3000-Series
- interface converter (GRS3100, GRS3105, ...)

2 Functions of the GMH3000-DLL

2.1 About the DLL in general

All function deliver return values, if they are negative, a error occurred.

Before performing a communication via interface, the interface principally has to be opened. After finishing communication, its good practice to close the interface again, otherwise You may 'loose' the interface until the programme/computer is restarted.

When using the dll or building your installation software, please consider that the dll has to be copied to the windows-system folder first.

2.2 Data Communication

Data communication will happen in 'polling-mode', which means that first the PC sends a 'Question' to the instrument, which answers then. All of this polling actions are included within the function GMH_Transmit
You don't need to care about!

2.3 Bus Addresses

One address exists for each measuring of an instrument.

Instruments which only have one measuring (e.g. GMH 3160) just do have one address (address 1), others, such as the 3510 do have more (address 1,2,3...). Instruments, which do need an GRS3000 (not GRS3100!) converter do have the bus address 0 instead of 1. The 'base-address' of High End instruments, such as the GMH3250 for example can be changed from 1 to 11, 21, 31..91. Therefore several instruments can be connected parallel to a special interface adapter (GRS3105: 5 instruments) without getting an address conflict.

2.4 Priority Messages

If *PRIO was set, one or more additional priority messages are present.

With the EBTransmit-function 3 the status word can be read. Depending on the bits set in this flag word, the returned measuring value may not be valid! See table Priority Messages

2.5 Language-Offset

A Language-Offset can be added to the code when using the functions GetUnit, GetMeasurement, GetStatus, GetErrorMessageRet, GetErrorMessageFL. The DLL returns the message in the corresponding language.

0 = German

4096 = English

8192 = Czech

12288 = Spanish

16384 = French

2.6 GMH_OpenCom

Function: Opens the referring interface port
Call: GMH_OpenCom (PORT)
Return (16bit Integer) 0..3, if opening was successful
 <0 if an error occurred (p.r.t. chapter ,3 Error Codes')
PORT (16bit Integer) Interface number: 1 = COM1, 2 = COM2, 3 = COM3, ... 255 = COM255

2.7 GMH_CloseCom

Function: Closes an opened port
Call: GMH_CloseCom
Return (Integer) 0 or positive, if closing was successful
 negative if an error occurred (p.r.t. chapter ,3 Error Codes')

2.8 GMH_Transmit

Function: With GMH_Transmit all data transfers are performed after the interface was opened by GMH_OpenCom. The parameter FUNCTION determines the function.
 If data are sent, they have to be contained in the parameters *FLOATINGPOINT or *INTEGERNUMBER.
 If data is read, the wished data will be returned in *FLOATINGPOINT or *INTEGERNUMBER.

Call: GMH_Transmit (ADDRESS,FUNCTION,*PRIO,*FLOATINGPOINT,*INTEGERNUMBER)
Return (Integer) 0, if transmission was successful
 <0 if an error occurred (p.r.t. Appendix A: Error Codes)

ADDRESS (16bit Integer) valid addresses for measurings (addresses 0 to 99)
FUNCTION (16bit Integer) 0..260 Number of the referring function
 (p.r.t. chapter ,2.6.3 Function Calls of GMH_Transmit')

***PRIO** pointer for 16bit Integer, will be set by GMH3000:
 0: no priority message, 1: priority message!
 (p.r.t. chapter ,2.6.2 Priority Messages')

***FLOATINGPOINT** pointer of 64bit Double, Parameter for floating point values
***INTEGERNUMBER** pointer of 32bit Integer, Parameter for integer values

2.8.1 Error -36 at GMH_Transmit-function 'GMH_GetValue'

If error -36 occurs by using the GMH_Transmit-Function 'GMH_GetValue' (function code 0), then the returned floating point value is a error message, not a valid measuring! Refer Extended Error Codes

2.8.2 Function Calls of GMH_Transmit

Function Call		Parameter			Return Values		
		Address	*FLOATINGPOINT	*INTEGERNUMBER	Func.	*FLOATINGPOINT	*INTEGERNUMBER
0	GMH_GetValue	0..99			error	Measuring value	
1	GMH_SetValue	0..99	display value		error	Display value	
3	GMH_GetStatus	0..99			error		BW: system state
6	GMH_GetMinValue	0..99			error	Min.-value	
7	GMH_GetMaxValue	0..99			error	Max.-value	
12	GMH_GetType	0..99			error		Instrument type
22	GMH_GetMinAlarm	0..99			error	min. alarm rail	
23	GMH_GetMaxAlarm	0..99			error	Max. alarm rail	
24	GMH_GetAlarmdelay	0..99			error		alarmdelay[min]
25	GMH_GetAlarmfunction	0..99			error		BW: alarm function
32	GMH_GetConfigFlag	0..99	BitNr{0..32768}		error		state
100	GMH_SetStatus	0..99		BW: system state	error		BW: system state
102	GMH_SetMinAlarm	0..99	min. alarm rail		error	min. alarm rail	
103	GMH_SetMaxAlarm	0..99	max. alarm rail		error	Max. alarm rail	
104	GMH_SetAlarmdelay	0..99		alarmdelay[min]	error		alarmdelay[min]
105	GMH_SetAlarmfunction	0..99		BW: alarm function	Error		BW: alarm function
160	GMH_SetConfigFlag	0..99	BitNr{0..32768}	State	error	BitNr{0..32768}	State
174	GMH_ClearMinValue	0..99		1	error	Min.-value	1
175	GMH_ClearMaxValue	0..99		1	error	Max.-value	1
176	GMH_GetMinRange	0..99			error	min. range	
177	GMH_GetMaxRange	0..99			error	Max. range	
178	GMH_GetUnitCode	0..99			error		unit
179	GMH_GetDecimalPoint	0..99			error		decimal point
180	GMH_GetMeasCode	0..99			error		measuring
194	GMH_SetDispUnitCode	0..99		unit of display	error		unit of display
195	GMH_SetDispDecPoint	0..99		decimal point of disp.	error		decimal point of displ.
199	GMH_GetDispMeasuring	0..99			error		measuring of display
200	GMH_GetDispMinRange	0..99			error	min. display range	
201	GMH_GetDispMaxRange	0..99			error	Max. display range	
202	GMH_GetDispUnitCode	0..99			error		unit of display
203	GMH_GetBatteryState	0..99			error		bat state[%]{0..100}
204	GMH_GetDispDecPoint	0..99			error		decimal point of display
208	GMH_GetChannelcount	0..99			error		channels per instrument
210	GMH_GetElectrodeState	0..99			error		state[%]{0..100}
214	GMH_GetSlopeCorrection	0..99			error	Correction in %	
215	GMH_SetSlopeCorrection	0..99	correction in percent		error	Correction in %	
216	GMH_GetOffsetCorrection	0..99			error	Offset correction	
217	GMH_SetOffsetCorrection	0..99	offset correction		error	Offset correction	
218	GMH_GetCorrFactor	0..99			error		correction factor*1000
219	GMH_SetCorrFactor	0..99		correction factor*1000	error		correction factor*1000
220	GMH_GetAltitude	0..99			error		Altitude [m]
221	GMH_SetAltitude	0..99		Altitude [m]	error		Altitude [m]
222	GMH_GetPowerOffTime	0..99			error		power off time [min]
223	GMH_SetPowerOffTime	0..99		power off time [min]	error		power off time [min]
224	GMH_GetLoggerData	0..99	offset in logger mem.	size of data block	error	Data	size of data block
225	GMH_GetLoggerCycle	0..99			error		Loggerzyklus[s]
226	GMH_SetLoggerCycle	0..99		logger cycle [s]	error		Loggerzyklus[s]
227	GMH_StartLogger	0..99	start delay [min]	start condition	error		
228	GMH_GetLoggerCount	0..99			error		data count
229	GMH_GetLoggerstate	0..99			error		logger state
							0=stop, 1=active, 6=direct
233	GMH_GetRealtimeClock	0..99			error	Real time (DOS/WINDOWS date format)	
234	GMH_SetRealtimeClock	0..99	real time (DOS/WINDOWS date format)		error	Real time (DOS/WINDOWS date format)	
236	GMH_GetLoggerSize	0..99			error		logger size
240	GMH_Reset	0..99,255			error		
254	GMH_GetSoftwareInfo	0..99			error	Version {0..255}	identifier{0..255}
260	GMH_GetLoggerDataMan	0..99	loggermem. address	size of data block	error	Data	size of data block

Table 2.3: function calls of GMH_Transmit

2.9 GMH_GetType

Function:

Returns the referring type-string to the type code, which was read from the instrument

Call:

GMH_GetType (TYPECODE, *TYPESTRING)

TYPECODE (32bit integer)

Code of Type, which is read by GMH_Tansmit-function ' GMH_GetTypeCode'

***TYPESTRING**

Pointer of String, which contains the Type afterwards (null-terminated).

Return (Char)

Length of string

2.10 GMH_GetUnit

Function: Returns the referring unit-string to the unit code, which was read from the instrument

Call: GMH_GetUnit (UNITCODE, *UNITSTRING)

UNITCODE (16bit Integer) Code of Unit: UNITCODE = UNIT + 0 for German units
UNITCODE = UNIT + 1000h for English units
UNIT: Code which is read by GMH_Transmit-func. 'GMH_GetDispUnitCode' or 'GMH_GetUnitCode'

***UNITSTRING** Pointer of String, contains the unit afterwards (null-terminated).

Return (Char) Length of string

2.11 GMH_GetMeasurement

Function: Returns the referring measurement-string to the measurement code, which was read from the instrument

Call: GMH_GetMeasurement (MEASCODE, *MEASSTRING)

MEASCODE (16bit Integer) Code of Measurement: MEASCODE = MEAS + 0 for German
MEASCODE = MEAS + 1000h for English
MEAS: Code which is read by GMH_Transmit-function 'GMH_GetMeasCode'

***MEASSTRING** Pointer of String, contains the measurement afterwards (null-terminated).

Return (Char) Length of string

2.12 GMH_GetErrorMessageRet

Function: Returns the referring error message to the returned error code (negative).
The error code is the return value of the GMH_Transmit function

Call: GMH_GetErrorMessageRet (ERRORCODE, *ERRORSTRING)

ERRORCODE (16bit integer) error code: ERRORCODE = ERR + LANGUAGEOFFSET
ERR: returned errorcode, read with GMH_Transmit (negative value)
LANGUAGEOFFSET: 0 German
01000h English

***ERRORSTRING** Pointer to string to be returned

Return (char) Length of returned string (maximum of 70 characters)

2.13 GMH_GetStatusMessage

Function: Returns the referring error message to the returned error code (negative).
The error code is the return value of the GMH_Transmit function

Call: GMH_GetErrorMessageRet (ERRORCODE, *ERRORSTRING)

ERRORCODE (16bit integer) error code: ERRORCODE = ERR + LANGUAGEOFFSET
ERR: returned statuscode, read with GMH_Transmit code 3
LANGUAGEOFFSET: 0 German
01000h English

***ERRORSTRING** Pointer to string to be returned

Return (char) Length of returned string (maximum of 70 characters)

2.14 GMH_GetErrorMessageFL

Function: Returns the referring error message to the returned error code
The error code is the floating point value of the Function GMH_Transmit, if the return value was -36 (code is error code)

Call: GMH_GetErrorMessageFL (ERRORCODE, *ERRORSTRING)

ERRORCODE (16bit integer) error code: ERRORCODE = ERR + LANGUAGEOFFSET
ERR: returned errorcode, read with GMH_Transmit
LANGUAGEOFFSET: 0 German
01000h English

***ERRORSTRING** Pointer to string to be returned

Return (char) Length of returned string (maximum of 40 characters)

2.15 GMH_GetVersionNumber

Function: Returns the version information of the dll

Call: GMH_GetVersionNumber

Return (16bit Integer) Version information, calculation example: Return = 1027d => 0403h => Version 4.03

2.16 GetAdditionalDelay

Function: Among other things simulated COM-Interfaces which are handled by the computer like a local COM-Port (e.g. Ethernet to serial converters), but are indeed a remote interface, can retard the communication so that errors occur. The standard timeouts for local interfaces can be expanded by „AdditionalDelay“. The execution time can be slowed down by this, especially at slave-searching-commands. Usually additional delays of 50..500 are sufficient, the unit is milliseconds.
GetAdditionalDelay: Reading the current setting of „AdditionalDelay“.

Call: GetAdditionalDelay()

Return (32bit Integer) 0...20000, delay in milliseconds

2.17 SetAdditionalDelay

Function: AdditionalDelay: see above
SetAdditionalDelay: Setting of „AdditionalDelay“.

Call: SetAdditionalDelay(DELAY)

DELAY (32bit integer) 0...20000. New delay in milliseconds (standard setting: 0)

Return (32bit Integer) 0...20000, delay in milliseconds

2.18 Read out the cyclic logger memory: Command GMH_ReadLogger

Function: Read out of the logger memory of the cyclic logger (Logg CYCL) of instruments supporting this feature. Attention: The manual logger (Logg Stor) cannot be read out by this command.
The execution time of the command can be very long (~150 data sets per sec -> 9999 data sets up to 1.2min, instruments with 4 ¾ display double of this time). If the application should not be 'locked' that long by the execution, we suggest to split the read out in several sequential calls with smaller count of data sets.

Call: GMH_ReadLogger
(ADDRESS,*FILENUMBER,*DATASETCOUNT,*STARTDATE,*LOGGERDATA())

Return (16bit Integer) 0, if transmission was successful
<0 if an error occurred (p.r.t. Error Codes)

ADDRESS (16bit Integer) Valid addresses for sensor modules:
1...99 Standard addresses for normal data transfer

***FILENUMBER** Pointer to 32Bit Integer
The file-number of the desired Dataset (see below *DATASETCOUNT) is handed over in here. For devices that store their datasets in only one file, a 0 has to be handed over. The file-number-count begins at 0. By now the GMH3151 and GMH3156 are supporting to save their datasets in more than one file.

***DATASETCOUNT** Pointer to 32bit integer
Here the desired count of data sets is handed over.
If a return value of -103 or -105 comes back, the data set count was altered by the command, the loggerdata return is valid.

***STARTDATE** Pointer to 64bit double
Here the desired start date of the loggerdata is handed over.
(in windows date format, e.g. 11.11.2004 10:39:13 ~ 38302,4439)
If loaded with 0 the loggerdata returned will be starting from the earliest available date.
The returned start date will automatically be corrected by the EBReadLogger command to the exact value.

***LOGGERDATA()** Pointer to one dimensional field of 64bit double variables.
The field has to have at least the size of the requested data set count.
Returned values are the logged measuring values or error codes in floating point format.

2.19 Read out the manual logger memory (Stor)

The read out of the manual logger memory is easily done by the GMH_Transmit command, code 260.

Per EB_Transmit command the data which are stored at a certain point of time, can be read (= data set).

To read out 10 data sets, therefore the command has to be called 10 times.

Call: GMH_Transmit (BASEADDRESS,FUNCTION,*PRIO,*BLOCK(),* BLOCKLENGTH)

Return (16bit Integer) 0, if transmission was successful
negative, if error occurred (p.r.t. Appendix A: Error Codes)

BASEADDRESS (16bit Integer 0...255) p.r.t. chapter 2.6 GMH_Transmit

FUNCTION (16bit Integer) 260, p.r.t. chapter 2.6 GMH_Transmit

***PRIO** Pointer auf 16bit Integer (p.r.t. chapter 2.6 GMH_Transmit)

***BLOCK()** Pointer to field of 64bit Doubles, passing of data set number (starting from 0)
return in floatingpoint format

***BLOCKLENGTH** Pointer to 32bit Integer, passing of the block length of data set to be read out.

Example: You want to read out an instrument with 3 meas. Channels, configured to base address 01.

It contains 2 data sets. The blocklength is calculated as follows: Count of Channels + 2 = 5

The field BLOCK therefore has to have at least the size of five 64bit Doubles.

Calling of the first data set:

GMH_Transmit (BASISADRESSE(=1), FUNKTION(=260), *PRIO, *BLOCK(=0), * BLOCKLENGTH =5

Values in brackets have to be set before the call of the command

The returned data set looks like following:

- BLOCK(0): 0 (reserved)
- BLOCK(1): Date and time of the data set in Windows date format
- BLOCK(2): Measuring value channel 1
- BLOCK(3): Measuring value channel 2
- BLOCK(4): Measuring value channel 3

Calling of the first data set:

GMH_Transmit (BASISADRESSE(=1), FUNKTION(=260), *PRIO, *BLOCK(=1), * BLOCKLENGTH =5

3 Error Codes of EASYBUS-DLL

Code	Meaning
0	(Add language offset for different language than german)
-1	Port number not valid and/or supported (CreateFile)
-2	Port already open (CreateFile)
-3	Port not ready (CreateFile)
-4	Queue cannot be set up (CreateFile)
-5	Standard parameter error (CreateFile)
-6	Converter type not supported/valid (CreateFile)
-10	Hardware (port) does not exist (CreateFile)
-11	Byte size not valid (CreateFile)
-12	Baud rate is not supported (OpenCom)
-13	The hardware detected a break condition (ReadFile)
-14	The hardware detected a framing error (ReadFile)
-15	A character-buffer overrun has occurred (ReadFile)
-16	An input-buffer overflow has occurred (ReadFile)
-17	The hardware detected a parity error (ReadFile)
-20	DCB formation not possible (CreateFile)
-21	Interface initialisation not possible (CreateFile)
-22	Message could not be sent (WriteFile)
-23	Transmission error: sensor module not responding
-24	Transmission error: wrong message length
-25	Transmission error: CRC code wrong
-26	Transmission error: address wrong
-27	Closing of port not possible (CloseComm)
-28	Testcode
-29	Echo data not identical
-30	Invalid operation code
-31	Transmission error: decimal point information invalid
-32	Transmission error: F-field incorrect
-33	Value not within allowed area
-34	Echo data not received
-35	Echo data not complete
-36	Return value is error code
-37	Message could not be sent: CTS Time Out (WriteFile)
-38	No acknowledge from sensor module received
-39	Transmission error: control code F2 incorrect
-40	Transmission error: length-of-message-bits incorrect
-41	Data received within locked range (3eb1h-3fffh)
-42	Transmission error: CRC code for echo data incorrect
-43	Message could not be sent: CTS changes (WriteFile)
-44	Transmission error: value transmitted <> value received, CRC ok
-45	More than 240 sensor modules found
-46	Number of data to be read exceeds max value
-47	Error in system initialisation
-48	Answer not expected
-51	Modem does not respond
-52	Wrong Modem response
-53	Could not establish connection
-54	Could not establish connection, timeout during dial attempt
-55	Cannot read state of modem connection
-56	Wrong security access prompt received from modem
-57	Enter password
-58	Modem error during entering of password
-59	Password was wrong!
-60	Problem during establishing connection to MODEM
-61	Answer cannot be decoded
-70	EBW 240 break or overload detected (ReadFile)
-71	Levelconverter overload detected

4 Extended Error Codes (on Error -36 use GetErrorMessageFI)

Float	Error
100000000	measuring range overrun
100000001	measuring range underrun
100000010	no value
100000011	system error
100000012	battery empty
100000013	no sensor
100000014	Recording error: EEPROM error
100000015	EEPROM checksum error
100000016	Recording error, Error 6: System restarted
100000017	Recording error: data pointer
100000018	Recording error: marker, data invalid
100000019	data invalid

5 Units

Temperature	°C	1
	°F	2
	K	3

Frequency	U/min	50
	Hz	53
	Pulse	55

Current	A	100
	mA	101
	µA	102

Voltage	V	105
	mV	106
	µV	107

Humidity	%r.H.	10
----------	-------	----

Speed	m/s	60
	km/h	61

Power	W	111
	kW	112
	Wh	115
	kWh	116
	Wh/m²	119

Pressure	inHg(0°C)	18
	inHg(60°F)	19
	bar	20
	mbar	21
	Pascal	22
	hPascal	23
	kPascal	24
	mPascal	25
	mm Hg	27
	PSI	28
	mm H ₂ O	29
Conductivity	S/cm	30
	mS/cm	31
	µS/cm	32

Distance	mm	70
	m	71
	inch	72
	ft	73

Resistance	mOhm	120
	Ohm	121
	kOhm	122
	MOhm	123

spec. Resistance	kOhm*cm	125
	MOhm*cm	126

Flow	l/s	79
	l/h	80
	l/min	81
	m³/h	82
	m³/min	83
	Nm³/h	84

Common	%	150
	°	151
	ppm	152
	g/kg	160
	g/m³	161
	mg/m³	162
	kJ/kg	170
	kcal/kg	171
	mg/l	172
	dB	175
	dBm	176
	dBA	177

pH/Redox	pH	40
	rH	41

Weigth & Force	g	90
	kg	91
	N	92
	Nm	93

Oxygen	mg/l O ₂	45
	%Sat O ₂	46
	%O ₂	47

6 Measurement

Temperature	Temperature	1	GAS	CO Concentration (gaseous)	60
	Differenz-Temperature	2		CO2 Concentration (gaseous)	61
Humidity	Rel. Air Humidity	10	Sonstige	Frequency	100
	Atmospheric Humidity	11		Revolutions	101
	Enthalpy	12		Counter	102
	Dewpoint Temperature	13		Pulses	103
	Dewpoint Distance	14			
	Wet Bulb Temperature	15			
	Material moisture u	16			
	Absolute humidity	17			
	Wet-basis moisture w	18			
Pressure	Absolute Pressure	20		Average absolute pressure	110
	Relative Pressure	21		Average relative pressure	111
	Absolute Pressure Min.	22			
	Absolute Pressure Max.	23			
	Relative Pressure Min.	24			
	Relative Pressure Max.	25			
	Pressure Difference	26			
	Pressure Difference Min.	27			
	Pressure Difference Max.	28			
Conductivity	Conductivity of Fluid	30		Velocity	120
	Resistance of Fluid	31		Flow	140
	Salinity	32		Current	160
	Total Dissolved Solids	33		Voltage	180
	Conductance of Fluid	34		Resistance	200
	Resistivity of Fluid	35		Conductance	220
PH/Redox	pH-Value	40		Conductivity	221
	ORP	41		Wind Direction	223
	ORP (Hydrogen Electrode)	42		Carboxy-haemoglobin	225
	rH-Value	43		Interface Operation	239
Oxygen	Oxygen Partial Pressure (dissolved)	50			
	Oxygen Concentration (dissolved)	51			
	Oxygen Saturation (dissolved)	52			
	Oxygen Partial Pressure (gaseous)	53			
	Oxygen Concentration (gaseous)	54			
Distance	Distance	70			
	Height	71			
	Level	72			

7 Priority Messages

Bit0	0x0001	1	max. alarm	
Bit1	0x0002	2	min. alarm	
Bit2	0x0004	4	display range overrun	FE3
Bit3	0x0008	8	display range underrun	FE4
Bit8	0x0100	256	measuring range overrun	FE1
Bit9	0x0200	512	measuring range underrun	FE2
Bit10	0x0400	1024	sensor error	FE9
Bit12	0x1000	4096	system fault	FE7
Bit13	0x2000	8192	calculation not possible	FE11
BitB15	0x8000	32768	low battery	FE8

8 Configuration Flags

0	Alarm-Hold	0 = clearing, 1 = storing
1	Alarm-Function	0 = disabled, 1 = enabled
2	Min-/Max-Alarm	0 = common 1 = independent
3	Alarm-Signal	0 = disabled, 1 = enabled
4	Alarm-output	0= standard, 1= inverted
32	CorrectToSeaLevel	0 = disabled, 1= enabled
33	PeakDetection	0= disabled, 1= enabled
34	FastFiltered	0= disabled, 1= enabled
47	Stop an active logger with buttons	0= enabled, 1= disabled
48	Stop an active logger	0= enabled, 1= disabled
49	Ring memory	0= enabled, 1= disabled
50	Logger	0 = Logger enabled, 1 = Logger disabled
51	Cyclic Logger	0 = manual logger, 1 = cyclic logger
52	Energy saving logger	0 = disabled, 1 = enabled (measuring only when logger is running)
60	Extended Range	0= Normal Range, 1= Extended Range
61	Displaying FE1 disabled	0= FE1 is shown, 1= max. measuring range is shown
62	Displaying FE2 disabled	0= FE2 is shown, 1= min. measuring range is shown
70	Change display cyclically	0= disabled, 1 = enabled
256	Repeater	0 = enabled, 1 = disabled
257	Sensorbus-access	0 = enabled, 1 = disabled
258	Timeout-supervision	0 = enabled, 1 = disabled