Application Solutions

InPro 6860 i

Modbus Programmers Guide

METTLER TOLEDO

Content

1 General Modbus Information

1.1 Documentation of the Modbus protocol

For detailed information about Modbus specifications, please refer to the following documents:

– Modbus_over_serial_line_V1_02.pdf
– Modbus_Application_Protocol_V1_1b3.pdf

These documents are available on the Modbus website: www.modbus.org
Go to tab „Technical Resources/Technical Specifications".

1.2 Modbus testing tool

On the web, there are plenty of Modbus testing tools or modbus libraries available for C++, Phyton or other programming languages.
For manually accessing the InPro 6860 i over Modbus, for instance to predefine the Modbus address, baudrate, or other items, the tool "Modbus Poll" is a valuable choice. It can be purchased at www.modbustools.com.

1.3 Protocol definitions, as implemented in InPro 6860 i

Modbus mode:	RTU
Start bits:	1
Data bits:	8
Stop bits:	2
Parity:	None
Baudrate:	19200 (default), 4800, 9600, 38400, 56600, 115000
Device address:	1 (default) to 247

The device does not implement any line polarization nor any line termination.
External line polarization and line termination is required.

1.4 Modbus RTU function codes implemented in InPro 6860 i

#3	Read Holding Registers
#4	Read Input Registers
#16	Write Multiple Registers

For detailed description of these functions please consult the document "Modbus_Application_Protocol_V1_1b3.pdf".

With InPro 6860 i, reading any register is performed by either command #3 or #4.
There is no difference in handling the information between these two commands.

1.5 Data representation

Each Modbus register contains two bytes, the data length of a command and an answer is always a multiple of two registers. Thus, the length of one data entry is 4 bytes, enabling also 32-bit floating point data.

The high byte (first byte) of a register contains the last digit of a value or string, the first digit of a value or string is found on the low byte (second byte) of the last register of the interesting register chain.
The first byte of a register always contains the higher order bits, the second byte contains the lower order bits.

Decimal values:
Integer decimal values are translated to hexadecimal numbers.
Non-integer decimal values are represented as single precision float values.
See below for examples.

For integer values:

Example:	A 32-bit value of 12'345'678.
Converted to hex:	BC614E
First register:	Lower bytes of the value (bytes 1, 2): 0x614E
Second register:	Higher bytes of the value (bytes 3, 4): 0x00BC

For float values:
The mantissa of the value is stored on the second register, its exponent in the first register.
The float data format is implemented according to IEEE 754, single precision.

Example:	2.5, converted to a 32-bit float value.
First register:	0x0000 (highest bit of first register is the sign; exponent bias is 127)
Second register:	0x4020

When using Modbus Poll select "float CDAB" to correctly interpret float values.

For ASCII-text strings:
Same data order as for integer values.

Example:	Text sample: "Text". ASCII-code is: 0x54 0x65 0x78 0x74
First register:	0x7478
Second register:	0x6554

1.6 Addressing scheme

The addressing scheme of InPro 6860 i is "Base 0" (first register number is 0).

A register offset is available on register number 0000.
Using this register offset, one can adjust the absolute starting point of the register bank to fit for instance already existing implementations.
The register offset is unsigned with a range of 0...32767.
For instance by setting the offset to 1, the sensor is becoming „Base 1".
By default, the register offset is set to 999, thus the first user register is on number 1031.
Please note: The register offset is always found on register number 0000, independent of its value.
The offset affects only register numbers 0002 and up.

The register numbers given on the following pages are always relative numbers.
The absolute number of a register is calculated by adding the register offset to the relative address.

Examples:
Register offset as default (999), the device address shall be read:
Relative register number of device address is 3096.
The effective absolute register number to be transmitted in the command is: 4095.

1.7 Error handling

Transmission errors (corrupt telegrams) are detected by the InPro 6860 i. Corrupt telegrams are discarded and the sensor is waiting for a next, correct telegram.

Errors on application layer are answered with an error message. In case the answer consists of an error code, the leading bit (0x80) of the function code is set, signaling the error condition.

The following error codes are implemented in InPro 6860i:

Error code, hex	Error type
0x00	No error
0x01	Illegal function code was sent to the sensor
0x02	Illegal data address (invalid register number, access denied)
0x03	Illegal data value (value out of range)
0x04	Slave device error (operation not successfully completed)

Error code 0x01 is returned when a function code other than #3, #4, #16 is sent to the sensor.

Error code 0x02 is returned in the following cases:
– Any attempts to undefined registers
– Any attempts to registers on a higher operator level then actually selected (access denied)
– When reading too many registers, so undefined registers would be attempted
– When writing too many or not enough registers at once, or on a wrong starting address

Error code 0x03 is returned when writing invalid data to a register. Invalid data means any value out of the range of the specific register (value below or above limits, value not part of a list of possible values).
In this case, the last valid data is restored on the specific Modbus register and no change is active.

Error code 0x04 is typically returned when trying to log-in to a higher user level with a wrong password or to an inexistent user level. In these cases, the log-in fail, the operation is not successfully completed.

1.8 User levels, password protection

InPro 6860 i sensors have implemented three user levels, level 0, 1 and 2.
Reading registers is possible on any user level, except some specific registers used for calibration.
Writing registers of the sensor typically means changing the configuration, which also changes the behavior of the sensor. To prevent of any unwanted configuration changes, most writing attempts are possible only on user level 2.
For all user levels, default passwords are stored in the sensor. These passwords can be changed by the user. Changed passwords are stored in the non-volatile memory of the sensor.

User levels and default passwords of InPro 6860 i:

User level	Code, hex	Default password, hex
0	0x03	0x00000000
1	0x0C	0x01145DEA
2	0x30	0x00F479CE

After each power-up, the sensor is reset to user level 0.
When trying to change the user level to an invalid level or using a wrong password, the sensor remains on the last valid user level, error code 0x04 is returned.

1.9 Writing registers, data retention

In nearly all cases, writing any registers of the InPro 6860 i means changing the configuration of the sensor. Any configuration data are stored in the non-volatile memory (FLASH) of the sensor. Thus, the changed configuration will not get lost by a power-down of the sensor.

It is a well-known fact, that FLASH memories only allows about 100'000 write attempts. By exceeding this limit, the FLASH memory might get damaged; resulting in data lost or corrupted data. A device with a damaged FLASH is no longer operable.

Attention!
The Modbus Master controller must make sure to write any configuration data only upon change and only during the commissioning phase of a system!
Automatic, periodic writes of data during normal operation must be prohibited!

2 Implemented Modbus registers in InPro 6860 i

Except register 0000, all register addresses are relative to the offset stored in register 0000.

Example:
Register offset is 999, as by default. Register 3288 shall be read.
The controller must read from register 4287.

Registers sorted in ascending register number order:

Register			Access Levels		Comments
Start-Register	Name	Count	Read	Write	
0000 fix	Register offset	2	0	0	Starting point of Modbus registers
32	User end firmware version	8	0	–	
72	User end hardware version	8	0	–	
280	Part number	8	0	–	
288	Sensor name	8	0	–	
296	Work order	8	0	–	
312	Serial number	8	0	–	
320	Manufacturer	8	0	–	
336	Sensor type	8	0	–	
928	Unit string °C	4	0	–	
932	Unit string °F	4	0	–	
936	Unit string %-vol	4	0	–	
940	Unit string %-sat	4	0	–	
944	Unit string µg/l	4	0	–	
948	Unit string mg/l	4	0	–	
960	Unit string mS/cm	4	0	–	
1012	Unit string mbar	4	0	–	
1048	Available measurement channels	2	0	–	
1080	Description of oxygen measurement channel	8	0	–	
1088	Available units for oxygen measurement channel	2	0	–	
1090	Unit of oxygen measurement	2	0	2	
1090	Oxygen measurement channel	10	0	–	
1400	Description of temperature measurement channel	8	0	–	
1408	Available units for temp. measurement channel	2	0	–	
1410	Unit of temperature measurement	2	0	2	
1410	Temperature measurement channel	10	0	–	
2072	Available parameters	2	0	–	
2104	Description of process salinity	8	0	–	
2112	Available units for process salinity	2	0	–	

Register			Access Levels		Comments
Start-Register	Name	Count	Read	Write	
2114	Unit and value of process salinity	4	0	2	
2114	Read all values of process salinity	8	0	–	
2136	Description of process pressure	8	0	–	
2144	Available units for process pressure	2	0	–	
2146	Unit and value of process pressure	4	0	2	
2146	Read all values of process pressure	8	0	–	
2168	Description of process humidity	8	0	–	
2176	Available units for process humidity	2	0	–	
2178	Unit and value of process humidity	4	0	2	
2178	Read all values of process humidity	8	0	–	
2520	Description of Sampling rate	8	0	–	
2528	Available units for Sampling rate	2	0	–	
2530	Unit and value of sampling rate	4	0	2	
2530	Read all values of sampling rate	8	0	–	
2552	Description of LED-off-temp	8	0	–	
2560	Available units for LED-off-temp	2	0	–	
2562	Unit and value of LED-off-temp	4	0	2	
2562	Read all values of LED-off-temp	8	0	–	
2584	Description of LED-on-hysteresis	8	0	–	
2592	Available units for LED-on-hysteresis	2	0	–	
2594	Unit and value of LED-on-Hysteresis	4	0	2	
2594	Read all values of LED-on-hysteresis	8	0	–	
3096	Device address	2	0	2	
3098	Address limits	4	0	–	
3102	Baudrate	2	0	2	
3104	Baudrate limits	4	0	–	
3288	User level	4	0	0	
3292	User level passwords	4	–	2	
3320	Available analog outputs	2	0	–	
3322	Available analog output modes	8	0	–	
3352	Description of analog output	8	0	–	
3360	Mode of analog output	2	0	2	
3362	Available measurement channels for AO	2	0	–	

Register			Access Levels		Comments
Start-Register	Name	Count	Read	Write	
3364	Selected measurement channels for AO	2	0	2	
3366	Min/max output current, mA mode	4	0	–	
3370	Current at min/max measurement value, mA	6	0	2	
3376	Unit of measured data reflected by the AO	2	0	–	
3378	Measurement value at min/max current	6	0	2	
3384	Constant current for analog output, mA mode	2	0	2	
3394	Min/max output current, nA mode	4	0	–	
3398	Current in zero oxygen/air, nA mode	6	0	–	
3404	Constant current for analog output, nA mode	2	0	2	
3414	Read measured output current	4	0	–	Reads always 0
3608	Operation temperature range	4	0	–	
3612	Measurement temperature range	4	0	–	
3676	Operating hours	6	0	–	
3688	SIP/CIP counts	4	0	–	
3692	Maximum for SIP/CIP counters	4	0	2	
3736	Pending warnings	8	0	–	
3800	Pending errors	8	0	–	
3988	SIP definition (temp./time)	8	0	–	
3988	Minimum SIP temperature	2	0	2	
3996	CIP definition (temp./time)	8	0	–	
3996	Minimum CIP temperature	2	0	2	
4120	Available calibration points	2	0	–	
4472	OptoCap quality (% value)	2	0	–	
4474	Read DLI/ACT	4	0	–	
4600	G100 data	10	2	–	
4612	Calibration range check	6	2	–	
4620	Calibration parameters	10	–	2	
4630	Calibration time stamp	8	–	2	
4640	Calibration control	16	–	2	
7192	Recall/factory reset	2	–	2	
7194	Clear counters (SIP, CIP, ACT, DLI)	2	–	2	

3 Detailed description of the implemented Modbus registers

3.1 User levels and passwords

After power-up, InPro 6860 i is set to user level 0.
User levels 1 or 2 can be selected by logging in with password.
The password of each access level can be changed by the user.

Set user level

To change or check the user level, write or read relative register number 3288:

Register		**Register usage**		**Access user level**	
Start	Count	Register 1/2	Register 3/4	Read	Write
3288	4	User level code	Password	0	2
Example		0x0030, 0x0000	0x79CE, 0x00F4		

The selected user level stays active until next power-down of the sensor. After power-up, user level 0 is active. Invalid login trials are discarded and user level 0 is activated.

Change passwords for user levels

To change the password of a user level, write relative register number 3290:

Register		**Register usage**		**Access user level**	
Start	Count	Register 1/2	Register 3/4	Read	Write
3292	4	User level code (hex)	Password (hex)	–	2
Example		0x0030, 0x0000	0x0202, 0x1905		

The invalid user level settings are discarded and no password will be changed.
Checking the valid passwords is performed by reading the user level.

3.2 Modbus register offset

By default, the Modbus register offset is defined to 999. If necessary, this offset can be changed to any number in the range of 0 … 32767.

Register		**Register usage**	**Access user level**	
Start	Count	Register 1/2	Read	Write
0000	2	Modbus register offset (unsigned integer)	0	2
Example		999 (hex-value on register #1: 0x03E7, register #2: 0x0000)		

3.3 Configuration of the RS485 interface

The factory settings of the RS485 interface are mentioned in chapter "Protocol definitions". The device address, as well as the baudrate can be adjusted to fit the installation.

Device address

By default, the device address is set to 1. By reading relative register 3098, the valid address range can be evaluated. The device address can be changed to any number within this range by writing register 3096:

Register		Register usage		Access user level	
Start	Count	Register 1/2	Register 3/4	Read	Write
3096	2	Device address (unsigned int)		0	2
3098	4	Min address (unsigned int)	Max address (unsigned int)	0	–

Baudrate

By default, the baudrate is set to 19200. Relative register 3104 reports the baudrate limits. The baudrate can be changed to any number within this range by writing register 3102:

Register		Register usage		Access user level	
Start	Count	Register 1/2	Register 3/4	Read	Write
3096	2	Device address (unsigned int)		0	2
3104	4	Min Baudrate code (unsigned int)	Max Baudrate code (unsigned int)	0	–

The Baudrate is represented as a decimal code:

Baudrate	4800	9600	19200	38400	57600	115200
Code	2	3	4 (default)	5	6	7

3.4 Configuration of the measurement

To ensure the highest possible measurement performance, the process conditions like pressure, humidity or salinity must be taken in account, or other parameters need to be adjusted. By reading or writing the appropriate registers, these values can be checked or modified.

Evaluation of the available parameters

Register		Register usage	Access user level	
Start	Count	Register 1/2	Read	Write
2072	2	Available parameters (bit array)	0	–

The meaning of the bitwise definition of the available parameters for InPro 6860 i:

Bit number	Hex value	Represented parameter
0	0x00000001	Process salinity
1	0x00000002	Process pressure
2	0x00000004	Process humidity
3 … 13	n.a.	
14	0x00004000	Sampling rate
15	0x00008000	LED off temperature
16	0x00010000	LED on hysteresis
17 … 31		n.a.

Description of the represented parameters

There is a text description available for every parameter, which could for instance be displayed on the screen of the Modbus controller. The text descriptions are located on the following relative register addresses:

Register		Register usage	Access user level	
Start	Count	Register 1 … 8	Read	Write
2104	8	Description of process salinity (ASCII)	0	–
2136	8	Description of process pressure (ASCII)	0	–
2168	8	Description of process humidity (ASCII)	0	–
2520	8	Description of sampling rate (ASCII)	0	–
2552	8	Description of LED off temperature (ASCII)	0	–
2584	8	Description of LED on hysteresis (ASCII)	0	–

Setting unit and value of the parameters

For each parameter, the available units can be evaluated from a register. For information about how to interpret the bit arrays, please consult chapter 3.5 Measurement via Modbus.

Register		Register usage	Access user level	
Start	Count	Register 1/2	Read	Write
2112	2	Available units for process salinity (bit array)	0	–
2144	2	Available units for process pressure (bit array)	0	–
2176	2	Available units for process humidity (bit array)	0	–
2528	2	Available units for sampling rate (bit array)	0	–
2560	2	Available units for LED off temperature (bit array)	0	–
2592	2	Available units for LED on hysteresis (bit array)	0	–

Setting or checking a parameter's unit and value is possible by accessing the following registers:

Register		Register usage		Access user level	
Start	Count	Register 1/2	Register 3/4	Read	Write
2114	4	Unit of process salinity (bitwise)	Value of process salinity (float)	0	2
2146	4	Unit of process pressure (bitwise)	Value of process pressure (float)	0	2
2178	4	Unit of process humidity (bitwise)	Value of process humidity (float)	0	2
2530	4	Unit of sampling rate (bitwise)	Value of sampling rate (float)	0	2
2562	4	Unit of LED off temp (bitwise)	Value of LED off temp. (float)	0	2
2594	4	Unit of LED on hyst. (bitwise)	Value of LED on hyst. (float)	0	2

Evaluation of each parameter's value ranges is possible by reading the following registers:

Register		Register usage		Access user level	
Start	Count	Register 1/2	Register 3/4	Read	Write
2118	4	Min value, process salinity	Max value, process salinity	0	–
2150	4	Min value, process pressure	Max value, process pressure	0	–
2182	4	Min value, process humidity	Max value, process humidity	0	–
2534	4	Min value, sampling rate	Max value, sampling rate	0	–
2566	4	Min value, LED off temperature (float)	Max value, LED off temperature (float)	0	–
2598	4	Min value, LED on hysteresis (float)	Max value, LED on hysteresis (float)	0	–

The complete settings of the parameters can be read at once from the following registers:

Register		Register usage				Access user level	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Reg 7/8	Read	Write
2114	8	Unit of process salinity	Actual value	Min value	Max value	0	–
2146	8	Unit of process pressure	Actual value	Min value	Max value	0	–
2178	8	Unit of process humidity	Actual value	Min value	Max value	0	–
2530	8	Unit of sampling rate	Actual value	Min value	Max value	0	–
2562	8	Unit of LED off. temperature	Actual value	Min value	Max value	0	–
2594	8	Unit of LED on hysteresis	Actual value	Min value	Max value	0	–
		(bitwise)	(float)	(float)	(float)		

3.5 Measurement via Modbus

The InPro 6860 i sensors offer two measurement channels: Oxygen and temperature.
The measured values of both channels are available on appropriate Modbus registers in single precision float format.
For each measurement channel, a test description, minimum/maximum of measurement range are available on Modbus registers. The measuring unit of each channel can be defined by the controller.

Evaluation of the available measurement channels

Register		Register usage	Access user level	
Start	Count	Register 1/2	Read	Write
1048	2	Available measurement channels (bit array)	0	–

The meaning of the bitwise definition of the available parameters for InPro 6860 i:

Bit number	Hex value	Represented measurement channel
0	0x00000001	Process salinity
0	0x00000001	Oxygen
1...4		n.a.
5	0x00000020	Temperature
17...31		n.a.

Description of the represented measurement channels

There is a text description available for each measurement channel, which could for instance be displayed on the screen of the Modbus controller. The text descriptions are located on the following relative register addresses:

Register		Register usage	Access user level	
Start	Count	Register 1 … 8	Read	Write
1080	8	Description of oxygen measurem. channel (ASCII)	0	–
1400	8	Description of temperature measurem. channel (ASCII)	0	–

Setting unit and value of the measurement channels

For each parameter, the available units can be evaluated from a register.

Register		Register usage	Access user level	
Start	Count	Register 1 / 2	Read	Write
1088	2	Available units for oxygen meas. channel (bit array)	0	–
1408	2	Available units for temp. meas. channel (bit array)	0	–

The meaning of the bitwise definition of the available measurement units for InPro 6860 i:

Bit number	Hex value	Represented measurement channel
0 … 1		n.a.
2	0x00000004	°C
3	0x00000008	°F
4	0x00000010	%-vol
5	0x00000020	%-sat
6	0x00000040	µg/l
7	0x00000080	mg/l
8 … 9	n.a.	
10	0x00000400	mS/cm
11 … 22	n.a.	
23	0x00800000	mbar
24 … 31		n.a.

The unit strings of the units are available on the following registers:

Register		**Register usage**	**Access user level**	
Start	Count	Register 1 … 8	Read	Write
928	4	Unit string °C	0	–
932	4	Unit string °F	0	–
936	4	Unit string %-vol	0	–
940	4	Unit string %-sat	0	–
944	4	Unit string µg/l	0	–
948	4	Unit string mg/l	0	–
960	4	Unit string mS/cm	0	–
1012	4	Unit string ppm	0	–

Setting or checking each measurement channel's unit is possible by accessing the following registers:

Register		**Register usage**	**Access user level**	
Start	Count	Register 1/2	Read	Write
1090	2	Unit of oxygen measure (bitwise)	0	2
1410	2	Unit of temperature measure (bitwise)	0	2

Each measurement channel can be read at once from the following registers:

Register		**Register usage**					**Access user level**	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Reg 7/8	Reg 9/10	Read	Write
1090	10	Unit, oxygen	Act. value	Meas. status	Min value	Max value	0	–
1410	10	Unit, temp.	Act. value	Meas. status	Min value	Max value	0	–
		(bitwise)	(float)	(bit array)	(float)	(float)		

The meaning of the bitwise definition of the measurement status is:

Bit number	**Hex value**	**Meaning**
0	0x00000001	Temperature out of measurement range
1	0x00000002	Temperature out of operating range
2	0x00000004	n. a.
3	0x00000008	Warning is pending
4	0x00000010	Error is pending

3.6 Configuration of the analog output

The sensors InPro 6860 i offer an analog output to represent the measured O_2 value. Depending on the detailed type of the sensor, either a nA output or a mA output is available. The scaling of this output can be configured as following. On a separate, resistive output, the temperature is represented. This output is not configurable.

The number of configurable analog outputs can be evaluated by reading relative register 3320:

Register		**Register usage**	**Access user level**	
Start	Count	Register 1 … 8	Read	Write
3320	2	Available analog outputs (Hex)	0	–

Description of the analog output

A text description for the analog output is available on the following relative register address:

Register		**Register usage**	**Access user level**	
Start	Count	Register 1 … 8	Read	Write
3352	8	Description of analog output (ASCII)	0	–

Evaluation of the available analog output configuration

The available output modes of the connected sensor can be read from the following registers:

Register		**Register usage**				**Access user level**	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Reg 7/8	Read	Write
3322	8	Available output modes (bit array)	Reserved	Reserved	Reserved	0	–

The meaning of the bitwise definition of the output modes for InPro 6860 i is:

Bit number	**Hex value**	**Available output mode**
–	0x00000000	No output available/output switched off[1]
0	0x00000001	mA output, fixed current for test purpose
1	0x00000002	4–20 mA output, reflecting actual O_2-value
2 … 7		n.a.
8	0x00000100	nA output, fixed current for test purpose
9	0x00000200	nA output, reflecting actual O_2-value, 70 nA range
10	0x00000400	nA output, reflecting actual O_2-value, 400 nA range
11 … 31		n.a.

1 for InPro 6860 i/mA sensors the output switched off means that a fix current of 4 mA is sourced to enable HART Multidrop operation.

Setting the analog output mode

The output mode is selected by setting the appropriate bit, as shown in the above table.

Register		**Register usage**	**Access user level**	
Start	Count	Register 1/2	Read	Write
3360	2	Mode of analog output (bitwise)	0	2

Note: Only one bit can be set at a time. Wrong settings will be discarded and the last valid output mode will stay active.

Defining a constant current for testing the analog output

In all versions of InPro 6860 i, a constant current for testing purposes can be defined by the following registers. There are different register available for nA and mA output:

Register		Register usage	Access user level	
Start	Count	Register 1/2	Read	Write
3384	2	Constant current for analog output configured to mA (float)	0	2
3404	2	Constant current for analog output configured to nA (float)	0	2

Defining the measurement channel reflected by the analog output

The available measurement channels to be reflected on the analog output is evaluated by checking register 3362, the measurement channel reflected by the analog output is selected by setting the appropriate bit of register 3364.

The meaning of the bitwise definition of the measurement channels represented by the analog output is:

Bit number	Hex value	Represented measurement channel
0	0x00000001	Oxygen
1 … 32		n. a.

Register		Register usage	Access user level	
Start	Count	Register 1/2	Read	Write
3362	2	Available measurement channels for analog output (bit array)	0	–
3364	2	Selected measurement channel for analog output (bitwise)	0	2

Note: Only one bit can be set at a time. Wrong settings will be discarded and the last valid output mode will stay active.

Minimum/maximum output currents of the analog output

The physical current limits of the output, as well as the configured measurement range can be evaluated by reading the following registers. Separate registers are used for nA mode and for mA mode:

Register		Register usage			Access user level	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Read	Write
3366	4	Min current, mA mode (float)	Max current, mA mode (float)	–	0	–
3394	4	Min current, nA mode (float)	Max current, nA mode (float)	–	0	–
3370	6	Current at minimum measurement value mA mode (float)	Current at maximum measurement value mA mode (float)	No meaning (float)	0	2
3398	6	Current in zero oxygen at 25 °C nA mode (float)	Current in air at 25 °C nA mode (float)	No meaning (float)	0	–

Unit measurement data reflected by the analog output

The physical unit of the measurement data reflected by the analog output is given by the measurment channel mapped to the analog output. The actual unit can be evaluated by reading the following registers. The same bit definition is valid as shown in chapter 3.5:

Register		Register usage	Access user level	
Start	Count	Register 1/2	Read	Write
3376	2	Unit of measurement data reflected by the analog output (bitwise)	0	–

Minimum/maximum measurement values for mA-output current

The configured measurement range can be evaluated by reading the following registers:

Register		Register usage			Access user level	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Read	Write
3378	6	Measurement value at min current (float)	Measurement value at max current (float)	No meaning (float)	0	2

3.7 Sensor calibration

For a detailed description of the calibration work flow, please refer to chapter 4. Within this chapter, the used Modbus registers are listed and explained in detail.

Available Calibration points

These Modbus registers are implemented for legacy reason only. Its content is always 0x00000000.

Register		Register usage	Access user level	
Start	Count	Reg 1/2	Read	Write
4120	2	Available calibration points [ ] (bit array)	0	–

G100 data

These Modbus registers provide the measured values used during calibration.

Register		Register usage					Access user level	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Reg 7/8	Reg 9/10	Read	Write
4600	10	O2_measured [%]	Phi_Tbcorr [°]	Phi_Tref [°]	Tm_measured [°C / °F]	Max value	0	–
		(float)	(float)	(float)	(float)	(bit array)		

Bitwise definition of Ext_status is:

Bit number	Hex value	Meaning
0, 1	0x00000003	0x00000000 → Signal stable in Nitrogen (0% air) 0x00000002 → Signal stable in air (100% air) 0x00000001/0x00000003 → No calibration point detected
2…31		n. a.

Calibration range check

These Modbus registers provide information about the validity of the evaluated calibration values.

Register		Register usage			Access user level	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Read	Write
4612	6	Cal range status	Phi0_Tref [°]	Phi100_Tref [°]	0	–
		(bit array)	(float)	(float)		

Bitwise definition of Cal range status is:

Bit number	Hex value	Meaning
0	0x00000001	n. a.
1	0x00000002	Phi0 out of range
2	0x00000004	Phi100 out of range
3	0x00000008	Phi out of range
3	0x00000010	KSV out of range
4 … 31		n. a.

Calibration parameters

These Modbus registers define the conditions valid for the following calibration.

Register		Register usage					Access user level	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Reg 7/8	Reg 9/10	Read	Write
4620	10	Tb [°C / °F]	Cal pressure [mBar]	Cal salinity [mS/cm]	Cal humidity [%]	O2-Value set [unit_ code]	–	2
		(float)	(float)	(float)	(float)	(float)		

Calibration time stamp

Calibration time stamp consists of two parts:

The calibration date string and the calibration time string.

Register		Register usage		Access user level	
Start	Count	Register 1/2/3/4	Register 5/6/7/8	Read	Write
4630	8	Calibration Time (ASCII)	Calibration Date (ASCII)	0	2

The data format of the strings is as follows. Make sure to use exactly 8 characters per string:

String name	Data format	Example
Cal Time string (ASCII)	HH:MM:SS	14:30:00
Cal Date string (ASCII)	YY/MM/DD	15/01/31

Calibration control

These Modbus registers control the calibration workflow and define the values used for calibration.

Register		Register usage								Access user level	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Reg 7/8	Reg 9/10	Reg 11/12	Reg 13/14	Reg 15/16	Read	Write
4640	16	Cal selector [-]	Adj selector [-]	Phi0_Tbcorr [°]	Phi100_Tb-corr [°]	Phi100_Tb-corr [°]	Phi100_Tref [°]	Tm0 [°C/°F]	Tm100 [°C/°F]	–	2
		(int)	(int)	(float)	(float)	(float)	(float)	(float)	(float)		

Definition of Cal selector is:

Value	Meaning
0	No calibration
2	One point air calibration
3	2-point calibration
4	Process calibration
Any other	n. a.

Definition of Adj selector is:

Value	Meaning
1	Trigger sensor adjustment
2	Trigger sensor calibration (store data on the sensor, but do not apply)
3	Trigger calibration range check

3.8 Sensor identification

By interrogating the Modbus registers mentioned below, detailed identification of the connected sensor is available. All information is formatted as readable ASCII-strings.

Register		**Register usage**	**Access user level**	
Start	Count	Register 1 … 8	Read	Write
32	8	User end firmware version (ASCII)	0	–
72	8	User end hardware version (ASCII)	0	–
280	8	Part number (ASCII)	0	–
288	8	Sensor name (ASCII)	0	–
296	8	Work order (ASCII)	0	–
312	8	Serial number (ASCII)	0	–
320	8	Manufacturer (ASCII)	0	–
336	8	Sensor type (ASCII)	0	–

3.9 Sensor status

Several status information, i.e. operating time, sensor quality, etc., is available by accessing the appropriate Modbus registers.

Temperature ranges

The operation temperature range, as well as the measurement temperature range of the sensor can be evaluated by reading the following registers:

Register		**Register usage**		**Access user level**	
Start	Count	Reg 1/2	Reg 3/4	Read	Write
3608	4	Min operation temp. (float)	Max operation temp. (float)	0	–
3612	4	Min measurement temp. (float)	Max measurement temp. (float)	0	–

Lifetime and SIP/CIP information

By reading the following registers, several life time information enable a detailed qualification of the sensor quality, determination of next calibration date or checking the expected life time of the OptoCap.

Register		Register usage			Access user level	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Read	Write
3676	6	Operating hours (integer)	No meaning	No meaning	0	–
3688	4	SIP counts (integer)	CIP counts (integer)	–	0	–
3692	4	Max. SIP counts (integer)	Max. CIP counts (integer)	–	0	2
4472	2	OptoCap quality [%] (float)	–	–	0	–
4474	4	DLI [days] (float)	ACT [days] (float)	–	0	–

SIP/CIP definition

The definition of SIP and CIP can be evaluated or changed by accessing the following registers. When changing the SIP/CIP definition, only the minimum temperature can be changed:

Register		Register usage				Access user level	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Reg 7/8	Read	Write
3988	8	SIP min temp. [°C] (float)	SIP max temp. [°C] (float)	SIP min time [min] (float)	No meaning (float)	0	–
3988	2	SIP min temp. [°C] (float)	–	–	–	0	2
3996	8	CIP min temp. [°C] (float)	CIP max temp. [°C] (float)	CIP min time [min] (float)	No meaning (float)	0	–
3996	2	CIP min temp. [°C] (float)	–	–	–	0	2

Warnings and errors

If measurement parameters are outside their ranges, or other problems should occur, warnings or errors are generated, available on the following registers:

Register		Register usage				Access user level	
Start	Count	Reg 1/2	Reg 3/4	Reg 5/6	Reg 7/8	Read	Write
3736	8	Measurement warning (bit array)	Calibration warning (bit array)	No meaning (bit array)	No meaning (bit array)	0	–
3800	8	Measurement errors (bit array)	No meaning (bit array)	No meaning (bit array)	Hardware errors (bit array)	0	–

Definition for measurement warnings

Bit number	Hex value	Warning
0	0x00000001	DO below minimum
1	0x00000002	DO above maximum
2	0x00000004	DO measurement unstable
3 … 24		n.a.
25	0x02000000	Temperature below minimum
26	0x04000000	Temperature above maximum
26 … 31		n.a.

Definition for calibration warnings

Bit number	Hex value	Warning
0	0x00000001	Calibration recommended (ACT below limit)
1		n.a.
2	0x00000004	OptoCap needs replacement (DLI below limit)
3 … 31		n.a.

Definition for measurement errors

Bit number	Hex value	Warning
0	0x00000001	DO measurement failure
1 … 24		n.a.
25	0x02000000	Temperature measurement failure
26 … 31		n.a.

Definition for hardware errors

Bit number	Hex value	Warning
0 … 1		n.a.
2	0x00000004	Temperature reading far below minimum
3	0x00000008	Temperature reading far above maximum
4 … 31		n.a.

3.10 System commands

In some cases, one must recall the factory settings, or after changing the OptoCap, the counters must be reset. Both actions can be performed by writing the correct key number to the following registers.
Recall/factory reset executs the same actions as with the METTLER TOLEDO transmitters.

Register		Register usage	Access user level	
Start	Count	Register 1/2	Read	Write
7192	2	Recall/factory reset Keyword: 732255 (decimal)	–	2
7194	2	Clear counters (SIP, CIP, ACT, DLI) Keyword: 0x696E6974	–	2

Note: Invalid keywords will be discarded, no action will be performed.

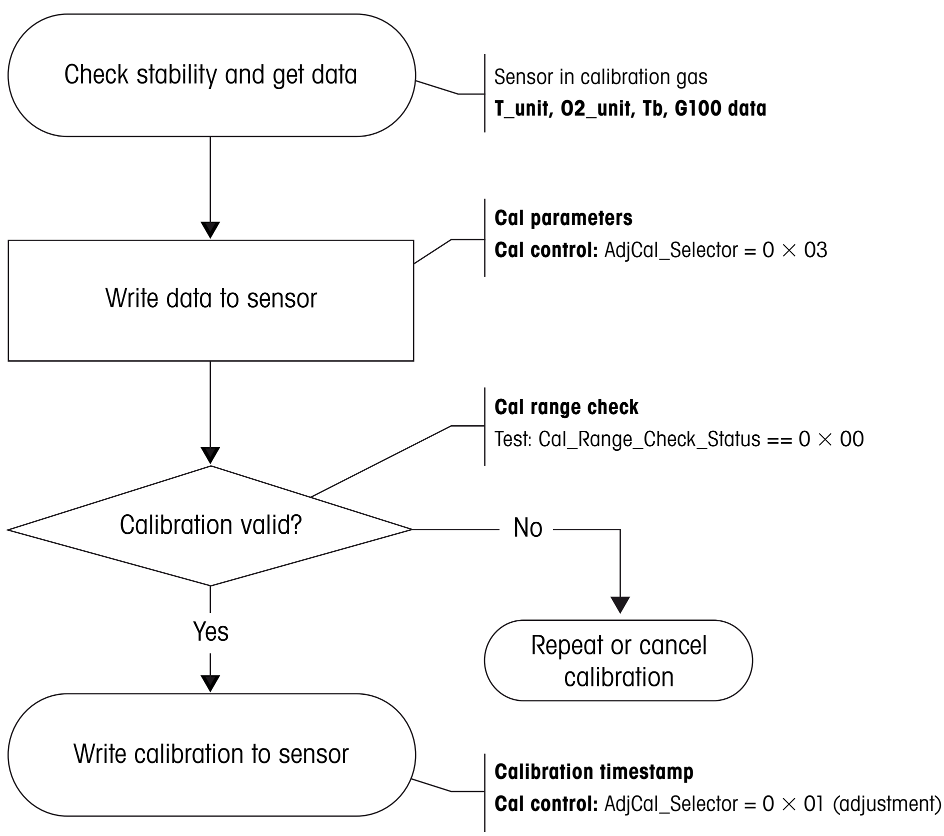
4 Description of the calibration workflow

4.1 Calibration workflow

All calibrations routines share a common structure, which contains following steps:

- Check measured values for stability
- Get data from sensor
- Write data to sensor
- Check if calibration possible
- Write calibration to sensor

This structure is depicted in following flowchart for reference:



In order to fully understand the reasons behind this process flow, one has to know that the sensor is not aware that it is being calibrated. This is why we need to read the data from the sensor and write it back along with a calibration command.

The first three steps are repeated in a 2-point calibration, while the first loop is the air calibration point, second loop is the nitrogen calibration point.

4.2 1-Point air calibration

4.2.1 Check stability and get data

First, get the actual units for O_2 and for temperature by reading the following registers:

Register name	**Register**	**Count**	**Variable name for future use**	**Meaning**
T_unitcode	1410	2	Temp_Unit_Code_100	Selected temp. unit
O2_unitcode	1090	2		Selected O_2 unit

Alternatively, you can define the units according to your favor by writing the appropriate unit codes to the registers.
Now read and store the sensor temperature:

Register name	**Register**	**Count**	**Variable name for future use**	**Meaning**
Tb	4620	2	Tb_100	Sensor temperature

Then start reading the "G100 data" registers shown below. Repeat read-out until the condition is true: Ext_status & 0x0003 == 2.
If the result is true, temporarily store the read register contents for ensuing steps:

G100 data

Register name	**Register**	**Count**	**Variable name for future use**	**Meaning**
O2_measured	4600	2		Unit as defined above
Phi_Tbcorr	4602	2	Phi_Tbcorr_100	Measured phase at Tm
Phi_Tref	4604	2	Phi_Tref_100	
Tm_measured	4606	2	Tm_100	Medium temperature
Ext_status	4608	2	Ext_status	

Go on with step 4.2.2.

4.2.2 Write data to sensor

Now the calibration parameters are written to the appropriate registers. These data define the conditions valid for the following calibration:

Calibration parameters

Register name	**Register**	**Count**	**Data**	**Notes**
Tb	4620	2	Tb_100	From step 4.2.1
Cal pressure	4622	2	e.g. 980	Enter accordingly to your calibration conditions
Cal salinity	4624	2	e.g. 50	
Cal humidity	4626	2	e.g. 0	
O2-Value set	4628	2	e.g. 100	Oxygen in air, unit as defined in step 4.2.1

After that, the calibration range check is triggered by writing following data to the calibration control register:

Calibration control

Register name	Register	Count	Data	Notes
Cal selector	4640	2	2	One-point air calibration
Adj selector	4642	2	3	Trigger calibration range check
Phi0_Tbcorr	4644	2	0.0	
Phi100_Tbcorr	4646	2	Phi100_Tbcorr	From step 4.2.1
Phi0_Tref	4648	2	0.0	
Phi100_Tref	4650	2	Phi_Tref_100	From step 4.2.1
Tm0	4652	2	0.0	
Tm100	4654	2	Tm_100	From step 4.2.1

After receiving this data the sensor computes a calibration based on the data but keeps measuring with the previous calibration. Go on with step 4.2.3.

4.2.3 Check calibration validity

Get information about the tentative calibration by reading the calibration range check data. The sensor compares the calibration parameters with some internal limits and issues four bits that can be evaluated as described in the appropriate section of chapter 3.7.

If Cal_range_status is 0, calibration values are valid and calibration can be executed.

If Cal_range_status is > 0, the calibration values are out of range, calibration must be discarded. In the latter case, no further action is needed or – if required – a new attempt can be started by going back to step 4.2.1.

Calibration range check

Register name	Register	Count	Variable name for future use	Meaning
Cal range status	4612	2		If value = 0, calibration is valid
Phi0_Tref	4614	2		
Phi100_Tref	4616	2		

To accept and write the calibration, carry out step 4.2.4.

4.2.4 Write calibration to sensor

Before writing the calibration, it is recommended to define the actual calibration date and time, to ensure a proper calibration history:

Calibration time stamp

Register name	Register	Count	Data	Notes
Cal time string	4630	4	e.g. 14:30:00	ASCII-string, exactly 8 characters long
Cal date string	4634	4	e.g. 15/01/31	ASCII-string, exactly 8 characters long

In order to write the calibration you need to re-write the calibration control register with the same dataset as in step 4.2.1. The only difference is in the AdjCal_Selector field:

Calibration control

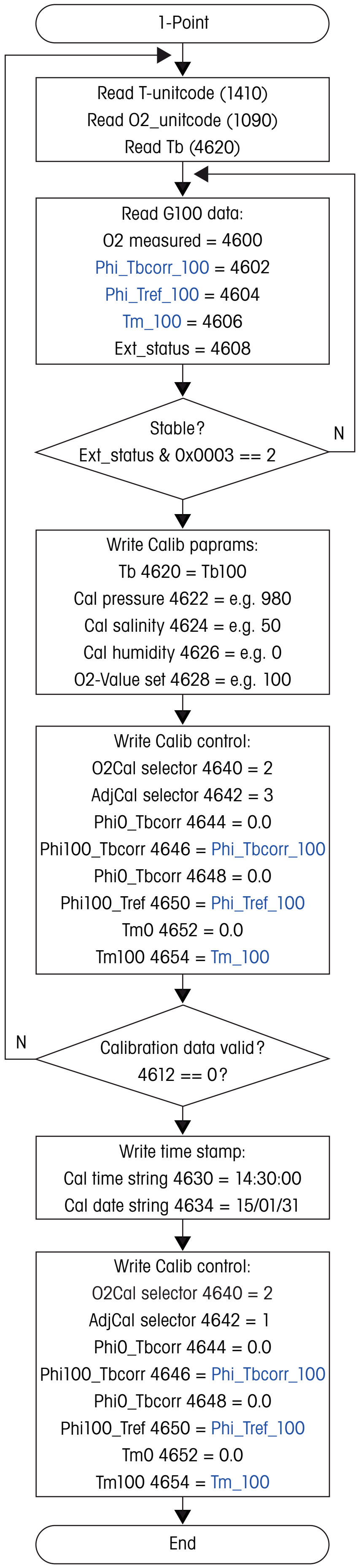
Register name	Register	Count	Data	Notes
Cal selector	4640	2	2	One-point air calibration
Adj selector	4642	2	1	Trigger sensor adjustment
Phi0_Tbcorr	4644	2	0.0	
Phi100_Tbcorr	4646	2	Phi100_Tbcorr	From step 4.2.1
Phi0_Tref	4648	2	0.0	
Phi100_Tref	4650	2	Phi_Tref_100	From step 4.2.1
Tm0	4652	2	0.0	
Tm100	4654	2	Tm_100	From step 4.2.1

Adjustment means the calibration gets active immediately. Instead of activating the evaluated calibration, it alternatively can be stored on the sensor without having any impact to the measurement.

If one wants to use this alternative approach, the Adj_selector must be set to 2 instead of 1.

Note:
Upon request, a VBA-script for Excel is available to demonstrate the sensor calibration.

4.2.5 1-point calibration detailed workflow



4.3 2-Point calibration

This calibration will be done in two steps. First step needs to be processed in air with 100 % oxygen saturation, second step is done in nitrogen (0 % oxygen saturation).

4.3.1 Check stability and get data for the first gas

Place the sensor in the first calibration gas, for instance in air (100 % oxygen saturation).
First get the actual units for O_2 and for temperature by reading the following registers:

Register name	Register	Count	Variable name for future use	Meaning
T_unitcode	1410	2	Temp_Unit_Code_100	Selected temp. unit
O2_unitcode	1090	2		Selected O_2 unit

Alternatively, you can define the units according to your favor by writing the appropriate unit codes to the registers.
Now read and store the sensor temperature:

Register name	Register	Count	Variable name for future use	Meaning
Tb	4620	2	Tb_100	Sensor temperature

Then start reading the "G100 data" registers shown below. Repeat read-out until the condition is true: Ext_status & 0x0003 == 2.
If the result is true, temporarily store the read register contents for ensuing steps:

G100 data

Register name	Register	Count	Variable name for future use	Meaning
O2_measured	4600	2		Unit as defined above
Phi_Tbcorr	4602	2	Phi_Tbcorr_100	Measured phase at Tm
Phi_Tref	4604	2	Phi_Tref_100	
Tm_measured	4606	2	Tm_100	Medium temperature
Ext_status	4608	2	Ext_status	

Go on with step 4.3.2.

4.3.2 Check stability and get data for the second gas

Move the sensor to the second calibration gas (e.g. nitrogen) and read again the "G100 data" registers shown below. Repeat read-out until the condition is true: Ext_status & 0x0003 == 2.
If the result is true, temporarily store the read register contents for ensuing steps:

G100 data

Register name	Register	Count	Variable name for future use	Meaning
O2_measured	4600	2		Unit as defined above
Phi_Tbcorr	4602	2	Phi_Tbcorr_100	Measured phase at Tm
Phi_Tref	4604	2	Phi_Tref_100	
Tm_measured	4606	2	Tm_100	Medium temperature
Ext_status	4608	2	Ext_status	

Go on with step 4.3.3.

4.3.3 Write data to sensor

Now the calibration parameters are written to the appropriate registers. These data define the conditions valid for the following calibration:

Calibration parameters

Register name	Register	Count	Data	Notes
Tb	4620	2	Tb_100	From step 4.3.1
Cal pressure	4622	2	e.g. 980	Enter accordingly to your calibration conditions
Cal salinity	4624	2	e.g. 50	
Cal humidity	4626	2	e.g. 0	
O2-Value set	4628	2	e.g. 100	Oxygen in air, unit as defined in step 4.3.1

After that, the calibration range check is triggered by writing following data to the calibration control register:

Calibration control

Register name	Register	Count	Data	Notes
Cal selector	4640	2	3	Two-point calibration
Adj selector	4642	2	3	Trigger calibration range check
Phi0_Tbcorr	4644	2	Phi_Tbcorr_0	From step 4.3.2
Phi100_Tbcorr	4646	2	Phi_Tbcorr_100	From step 4.3.1
Phi0_Tref	4648	2	Phi_Tref_0	From step 4.3.2
Phi100_Tref	4650	2	Phi_Tref_100	From step 4.3.1
Tm0	4652	2	Tm_0	From step 4.3.2
Tm100	4654	2	Tm_100	From step 4.3.1

After receiving this data the sensor computes a calibration based on the data but keeps measuring with the previous calibration. Go on with step 4.3.4.

4.3.4 Check calibration validity

Get information about the tentative calibration by reading the calibration range check data. The sensor compares the calibration parameters with some internal limits and issues four bits that can be evaluated as described in the appropriate section of chapter 3.7.

If Cal_range_status is 0, calibration values are valid and calibration can be executed.

If Cal_range_status is > 0, the calibration values are out of range, calibration must be discarded. In the latter case, no further action is needed or – if required – a new attempt can be started by going back to step 4.3.1.

Calibration range check

Register name	Register	Count	Variable name for future use	Meaning
Cal range status	4612	2		If value = 0, calibration is valid
Phi0_Tref	4614	2		
Phi100_Tref	4616	2		

To accept and write the calibration, carry out step 4.3.5.

4.3.5 Write calibration to sensor

Before writing the calibration, it is recommended to define the actual calibration date and time, to ensure a proper calibration history:

Calibration time stamp

Register name	Register	Count	Data	Notes
Cal time string	4630	4	e.g. 14:30:00	ASCII-string, exactly 8 characters long
Cal date string	4634	4	e.g. 15/01/31	ASCII-string, exactly 8 characters long

In order to write the calibration you need to re-write the calibration control register with the same dataset as in step 4.3.3. The only difference is in the AdjCal_Selector field:

Calibration control

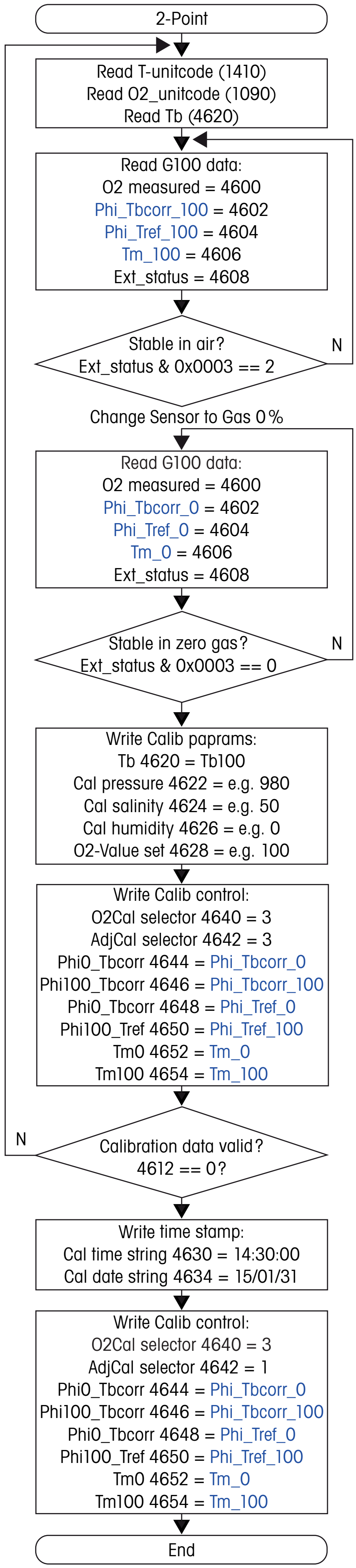
Register name	Register	Count	Data	Notes
Cal selector	4640	2	3	Two-point calibration
Adj selector	4642	2	1	Trigger sensor adjustment
Phi0_Tbcorr	4644	2	Phi_Tbcorr_0	From step 4.3.2
Phi100_Tbcorr	4646	2	Phi_Tbcorr_100	From step 4.3.1
Phi0_Tref	4648	2	Phi_Tref_0	From step 4.3.2
Phi100_Tref	4650	2	Phi_Tref_100	From step 4.3.1
Tm0	4652	2	Tm_0	From step 4.3.2
Tm100	4654	2	Tm_100	From step 4.3.1

Adjustment means the calibration gets active immediately. Instead of activating the evaluated calibration, it alternatively can be stored on the sensor without having any impact to the measurement.

If one wants to use this alternative approach, the Adj_selector must be set to 2 instead of 1.

Note:
Upon request, a VBA-script for Excel is available to demonstrate the sensor calibration.

4.3.6 2-point calibration detailed workflow



4.4 Process calibration

4.4.1 Check stability and get data

First, get the actual units for O_2 and for temperature by reading the following registers:

Register name	Register	Count	Variable name for future use	Meaning
T_unitcode	1410	2	Temp_Unit_Code_X	Selected temp. unit
O2_unitcode	1090	2		Selected O_2 unit

Alternatively, you can define the units according to your favor by writing the appropriate unit codes to the registers.
Now read and store the sensor temperature:

Register name	Register	Count	Variable name for future use	Meaning
Tb	4620	2	Tb_X	Sensor temperature

Then start reading the "G100 data" registers shown below. Other than for the calibrations in air and nitrogen, one cannot use the automatic stability detection when performing a process calibration. Compare the difference (or standard deviation) between two (or more) subsequent samples and when the variation is small enough for your application, store the data listed in the following table. Be sure to poll the sensor at a slower rate than its sampling rate to avoid a wrong stability detection.
If the result is true, temporarily store the read register contents for ensuing steps:

G100 data

Register name	Register	Count	Variable name for future use	Meaning
O2_measured	4600	2		Unit as defined above
Phi_Tbcorr	4602	2	Phi_Tbcorr_X	Measured phase at Tm
Phi_Tref	4604	2	Phi_Tref_X	
Tm_measured	4606	2	Tm_X	Medium temperature
Ext_status	4608	2		

Go on with step 4.4.2.

4.4.2 Write data to sensor

Now the calibration parameters are written to the appropriate registers. These data define the conditions valid for the following calibration. Note that O2-Value set is no longer a fixed parameter: enter your current process value against which you calibrate the sensor.:

Calibration parameters

Register name	Register	Count	Data	Notes
Tb	4620	2	Tb_X	From step 4.4.1
Cal pressure	4622	2	e.g. 980	Enter accordingly to your calibration conditions
Cal salinity	4624	2	e.g. 50	
Cal humidity	4626	2	e.g. 0	
O2-Value set	4628	2	e.g. 77.5	Oxygen in air, unit as defined in step 4.4.1

After that, the calibration range check is triggered by writing following data to the calibration control register:

Calibration control

Register name	Register	Count	Data	Notes
Cal selector	4640	2	4	Process calibration
Adj selector	4642	2	3	Trigger calibration range check
Phi0_Tbcorr	4644	2	0.0	
Phi100_Tbcorr	4646	2	Phi_Tbcorr_X	From step 4.4.1
Phi0_Tref	4648	2	0.0	
Phi100_Tref	4650	2	Phi_Tref_X	From step 4.4.1
Tm0	4652	2	0.0	
Tm100	4654	2	Tm_X	From step 4.4.1

After receiving this data the sensor computes a calibration based on the data but keeps measuring with the previous calibration. Go on with step 4.4.3.

4.4.3 Check calibration validity

Get information about the tentative calibration by reading the calibration range check data. The sensor compares the calibration parameters with some internal limits and issues four bits that can be evaluated as described in the appropriate section of chapter 3.7.

If Cal_range_status is 0, calibration values are valid and calibration can be executed.

If Cal_range_status is > 0, the calibration values are out of range, calibration must be discarded. In the latter case, no further action is needed or – if required – a new attempt can be started by going back to step 4.4.1.

Calibration range check

Register name	Register	Count	Variable name for future use	Meaning
Cal range status	4612	2		If value = 0, calibration is valid
Phi0_Tref	4614	2		
Phi100_Tref	4616	2		

To accept and write the calibration, carry out step 4.2.4.

4.4.4 Write calibration to sensor

Before writing the calibration, it is recommended to define the actual calibration date and time, to ensure a proper calibration history:

Calibration time stamp

Register name	Register	Count	Data	Notes
Cal time string	4630	4	e.g. 14:30:00	ASCII-string, exactly 8 characters long
Cal date string	4634	4	e.g. 15/01/31	ASCII-string, exactly 8 characters long

In order to write the calibration you need to re-write the calibration control register with the same dataset as in step 4.4.3. The only difference is in the AdjCal_Selector field:

Calibration control

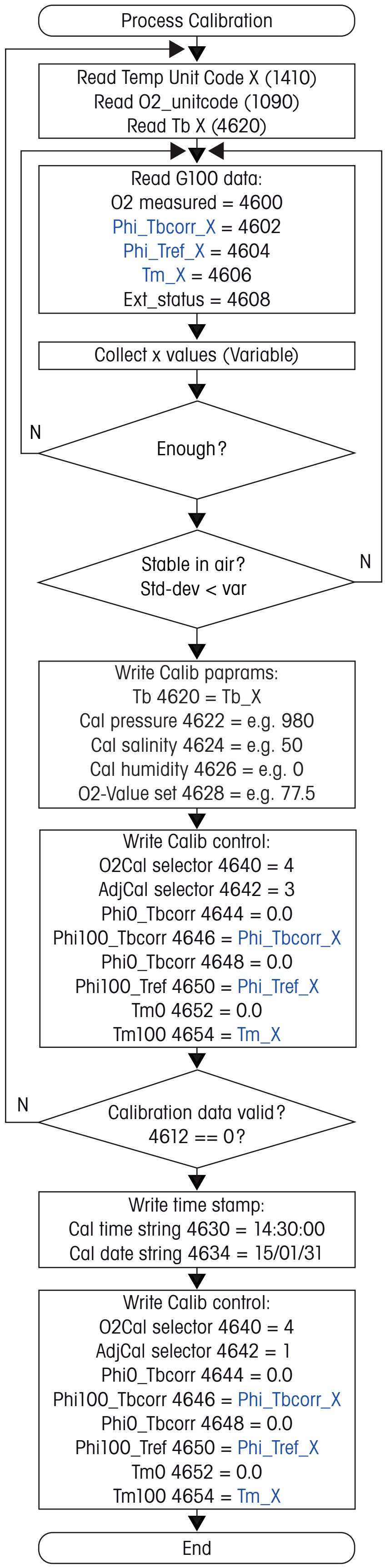
Register name	Register	Count	Data	Notes
Cal selector	4640	2	4	Process calibration
Adj selector	4642	2	1	Trigger sensor adjustment
Phi0_Tbcorr	4644	2	0.0	
Phi100_Tbcorr	4646	2	Phi_Tbcorr_X	From step 4.4.1
Phi0_Tref	4648	2	0.0	
Phi100_Tref	4650	2	Phi_Tref_X	From step 4.4.1
Tm0	4652	2	0.0	
Tm100	4654	2	Tm_X	From step 4.4.1

Adjustment means the calibration gets active immediately. Instead of activating the evaluated calibration, it alternatively can be stored on the sensor without having any impact to the measurement.

If one wants to use this alternative approach, the Adj_selector must be set to 2 instead of 1.

Note:
Upon request, a VBA-script for Excel is available to demonstrate the sensor calibration.

4.4.5 Process calibration detailed workflow



4.5 Scaling

4.5.1 Check stability and get data

First, get the actual units for O_2 and for temperature by reading the following registers:

Register name	Register	Count	Variable name for future use	Meaning
T_unitcode	1410	2	Temp_Unit_Code_X	Selected temp. unit
O2_unitcode	1090	2		Selected O_2 unit

Alternatively, you can define the units according to your favor by writing the appropriate unit codes to the registers.
Now read and store the sensor temperature:

Register name	Register	Count	Variable name for future use	Meaning
Tb	4620	2	Tb_X	Sensor temperature

Then start reading the "G100 data" registers shown below. Other than for the calibrations in air and nitrogen, one cannot use the automatic stability detection when performing a process calibration. Compare the difference (or standard deviation) between two (or more) subsequent samples and when the variation is small enough for your application, store the data listed in the following table. Be sure to poll the sensor at a slower rate than its sampling rate to avoid a wrong stability detection.
If the result is true, temporarily store the read register contents for ensuing steps:

G100 data

Register name	Register	Count	Variable name for future use	Meaning
O2_measured	4600	2		Unit as defined above
Phi_Tbcorr	4602	2	Phi_Tbcorr_X	Measured phase at Tm
Phi_Tref	4604	2	Phi_Tref_X	
Tm_measured	4606	2	Tm_X	Medium temperature
Ext_status	4608	2		

Go on with step 4.4.2.

4.5.2 Write data to sensor

Now the calibration parameters are written to the appropriate registers. These data define the conditions valid for the following calibration. Note that O2-Value set is no longer a fixed parameter: enter your current process value against which you calibrate the sensor.:

Calibration parameters

Register name	Register	Count	Data	Notes
Tb	4620	2	Tb_X	From step 4.5.1
Cal pressure	4622	2	e.g. 980	Enter accordingly to your calibration conditions
Cal salinity	4624	2	e.g. 50	
Cal humidity	4626	2	e.g. 0	
O2-Value set	4628	2	e.g. 77.5	Oxygen in air, unit as defined in step 4.5.1

After that, the calibration range check is triggered by writing following data to the calibration control register:

Calibration control

Register name	Register	Count	Data	Notes
Cal selector	4640	2	5	Scaling requested
Adj selector	4642	2	1	Trigger sensor adjustment
Phi0_Tbcorr	4644	2	0.0	
Phi100_Tbcorr	4646	2	Phi_Tbcorr_X	From step 4.5.1
Phi0_Tref	4648	2	0.0	
Phi100_Tref	4650	2	Phi_Tref_X	From step 4.5.1
Tm0	4652	2	0.0	
Tm100	4654	2	Tm_X	From step 4.5.1

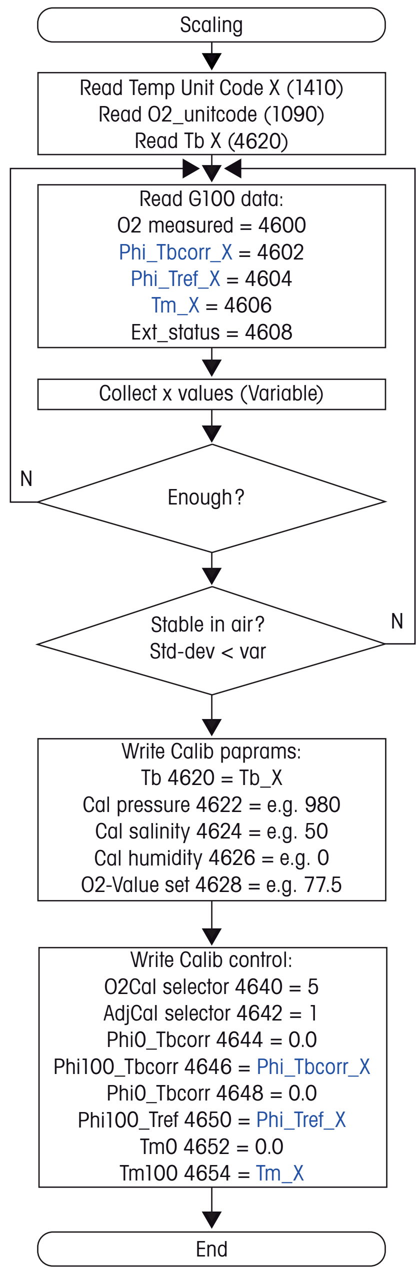
No modification is done to the current calibration data. The oxygen measurement is linearly scaled instead so that the linear function goes through the current process point and zero.

When scaling, no verification step is needed.

Note:
Upon request, a VBA-script for Excel is available to demonstrate the sensor calibration.

Modbus Programmers Guide

4.5.3 Scaling detailed workflow



Notes

The information you want
is at www.mt.com/pro

The METTLER TOLEDO Process Analytics website contains a vast amount of up-to-date information on all our products and services. Content is localized for your country and tailored to suit your selections. Simple layout allows you to quickly find the information and features you are looking for.

- Learn about our most recent product developments
- Register for free webinars
- Request further information on products and services
- Obtain a quote quickly and easily
- Download our latest white papers
- Read case studies relevant to your industry
- Access buffer and electrolyte solution certificate
- and more …

www.mt.com/pro

Visit for more information

Mettler-Toledo GmbH
Process Analytics
Im Hackacker 15
CH-8902 Urdorf
Phone +41 44 729 62 11
Fax +41 44 729 66 36