GlucoSense RS485 Arc Sensors

Modbus RTU Programmer's Manual

Firmware version:

GLOUM101

Important Notice

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form without written permission from Hamilton Bonaduz AG.

The contents of this manual are subject to change without notice. Technical changes reserved.

All efforts have been made to ensure the accuracy of the contents of this manual. However, should any errors be detected, Hamilton Bonaduz AG would greatly appreciate being informed of them.

The above notwithstanding, Hamilton Bonaduz AG can assume no responsibility for any errors in this manual or their consequences.

Copyright © 2025 Hamilton Bonaduz AG, Switzerland.

Rev.	Revision Date	Author	Change Description
00	06.11.2025	GRi	Initial version GLOUM101

Table of Content

1 MODBUS RTU GENERAL INFORMATION 5

1.1 INTRODUCTION 5
1.2 HAMILTON ARC SENSORS: MODBUS COMMAND STRUCTURE 5
1.2.1 Modbus RTU: Definitions According to Modbus IDA 5
1.2.2 Command Structure 6
1.2.3 Modbus RTU Transmission Mode 6
1.2.4 Modbus RTU Message Framing 7
1.2.5 Modbus RTU CRC Checking 8
1.3 IMPLEMENTATION OF MODBUS RTU IN HAMILTON ARC SENSORS 9
1.4 MODBUS RTU FUNCTION CODES USED FOR ARC SENSORS 9
1.4.1 Modbus function code #3: Read Holding Registers 9
1.4.2 Modbus function code #4: Read Input Registers 10
1.4.3 Modbus Function Code #16: Write Multiple Registers 11
1.5 DATA FORMATS USED IN ARC SENSORS 12
1.5.1 Float 12
1.5.2 Character 13
1.5.3 Decimal 13
1.6 MODBUS RTU ERROR MESSAGES 14

2 GLUCOSENSE RS485 COMMANDS IN MODBUS RTU 15

2.1 GENERAL 15
2.2 OPERATOR LEVELS AND PASSWORDS 15
2.2.1 Reading / Setting Operator Level 15
2.2.2 Changing Passwords for Operator Level 16
2.3 CONFIGURATION OF THE SERIAL RS485 INTERFACE 16
2.3.1 Device Address 16
2.3.2 Baud Rate 17
2.3.3 Parity and Stop Bits 17
2.4 CONFIGURATION OF THE ANALOG INTERFACES (4-20 MA) 18
2.4.1 Available Analog Interfaces 18
2.4.2 Available Analog Interface Modes 18
2.4.3 Description of the Analog Interface 19
2.4.4 Selection of an Analog Interface Mode 19
2.4.5 Configuration of the 4-20 mA Interface 19
2.4.6 Reading the Internally Calculated Output Current 26
MEASUREMENT 27
2.4.7 Definition of Measurement Channels and Physical Units 27
2.4.8 Primary Measurement Channel 1 (Glucose Concentration) 29
2.4.9 Definition of the Measurement Status for PMC1 / PMC6 34
2.4.10 Secondary Measurement Channel 1 (T act) 35
2.4.11 Secondary Measurement Channel 8 (S1) 36
2.4.12 Secondary Measurement Channel 9 (S2) 37
2.4.13 Secondary Measurement Channel 10 (S3) 38
2.4.14 Secondary Measurement Channel 11 (S4) 39
2.5 MEASUREMENT HOLD MODE 40
2.6 CONFIGURATION OF THE MEASUREMENT 41
2.6.1 Available Parameters 41
2.6.2 PA9: Moving Average 42
2.7 VERIFICATION 44
2.7.1 Initial Setup 44
2.7.2 Product Calibration 44
2.7.3 Glucose Standard Measurement 44
2.7.4 Verification of Tolerance 44
2.7.5 Further Action 44
2.8 CALIBRATION 45
2.8.1 Available Calibration Points 45
2.8.2 Definitions of Calibration Points 46

2.8.3 Calibration Procedure 53
2.8.4 Reading the Calibration Status 59
2.9 SENSOR STATUS 60
2.9.1 Temperature Ranges 60
2.9.2 Operating Hours, Counters and System Time 61
2.9.3 Warnings 63
2.9.4 Errors 65
2.9.5 PMCs output in case of sensor error / warning 67
2.9.6 Reading Definition of SIP and CIP 69
2.9.7 Quality Indicator 70
2.10 SENSOR IDENTIFICATION AND INFORMATION 71
2.10.1 General Information 71
2.10.2 Sensor Identification 71
2.10.3 Free User Memory Space 72
2.10.4 Membrane Information 72
2.11 SYSTEM COMMANDS 73
2.11.1 Recall Sensor's Factory Settings 73
3 APPENDIX 74
3.1 LIST OF TABLES 74
3.2 LIST OF FIGURES 75

1 Modbus RTU General Information

1.1 Introduction

This document describes in detail the GlucoSense RS485 (Arc) Modbus RTU interface. It is addressed to software programmers.
The general information about Modbus command structures and its implementation in the Hamilton Arc Sensor family is described in detail in chapter 1.

1.2 Hamilton Arc Sensors: Modbus Command Structure

In the present manual, only the specific command structure for the GlucoSense RS485 Sensor is described. It is valid for the firmware:

GLOUM101

Please check by reading register 1032 (see chapter 2.10.1).

This present definition of the command structure is an additional document to the Operating Instructions of the specific GlucoSense RS485 sensors. Before reading this manual, the operating instructions of the sensors should be read and understood.

1.2.1 Modbus RTU: Definitions According to Modbus IDA

The definitions in chapter 1.2 are an excerpt from the document:

- "Modbus over serial line - Specification and Implementation Guide V1.02" and
- "Modbus Application Protocol Specification V1.1b"

For more detailed information please consult http://www.modbus.org.



Attention:

- In this manual the register counting starts per definition at address 1. Some Modbus client protocols operate with register-count starting at address 0. Usually, the Modbus client software translates the addressing. Thus, the register address of 2088 will be translated by Modbus client software to 2087 which is sent to the sensor (Modbus server). This must be observed during programming. Please check the specifications of the Modbus client that you are using.
- Representation of data formats in this document:
  - decimal values are displayed as numbers without any prefix, for example 256
  - hexadecimal values are displayed as: 0x2A
  - ASCII-characters or ASCII strings are displayed as: "Text"

1.2.2 Command Structure

The Modbus application protocol defines a simple **P**rotocol **D**ata **U**nit (PDU) independent of the underlying communication layers:

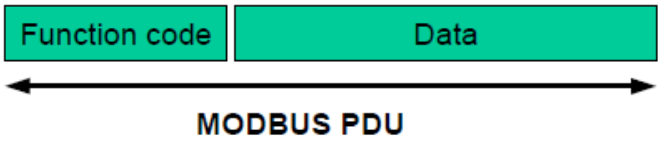


Figure 1 Modbus Protocol Data Unit.

The mapping of Modbus protocol on a specific bus or network introduces some additional fields on the **P**rotocol **D**ata **U**nit. The client that initiates a Modbus transaction builds the Modbus PDU, and then adds fields in order to build the appropriate communication PDU.

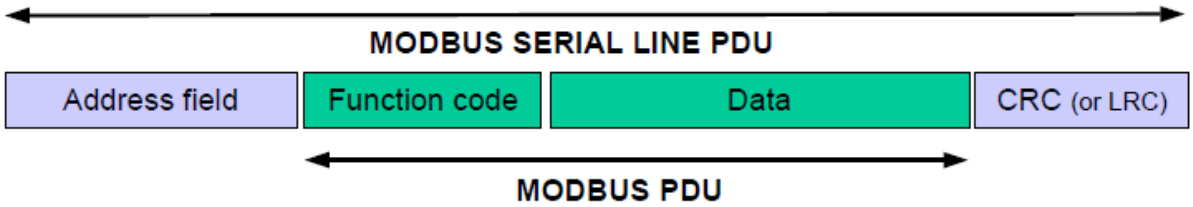


Figure 2 Modbus frame over Serial Line.

- On Modbus Serial Line, the Address field only contains the server address.

Note:
Arc Sensors support only server addresses 1 to 32.

A client addresses a server by placing the server address in the address field of the message. When the server returns its response, it places its own address in the response address field to let the client know which server is responding.

- The function code indicates to the server what kind of action to perform. The function code can be followed by a data field that contains request and response parameters.
- The CRC field is the result of a "Redundancy Checking" calculation that is performed on the message contents.

1.2.3 Modbus RTU Transmission Mode

When devices communicate on a Modbus serial line using the RTU (Remote Terminal Unit) mode, each 8-bit byte in a message contains two 4-bit hexadecimal characters. The main advantage of this mode is that its greater character density allows better data throughput than ASCII mode for the same baud rate. Each message must be transmitted in a continuous stream of characters.

The format (11bits) for each byte in RTU mode is:
Coding System: 8 bit binary
Bits per Byte: 1 start bit
8 data bits, least significant bit sent first
1 bit for parity completion
1 stop bit

Remark: The use of no parity requires 2 stop bits.

How characters are transmitted serially:
Each character or byte is sent in this order (left to right):
Least Significant Bit (LSB)…Most Significant Bit (MSB)

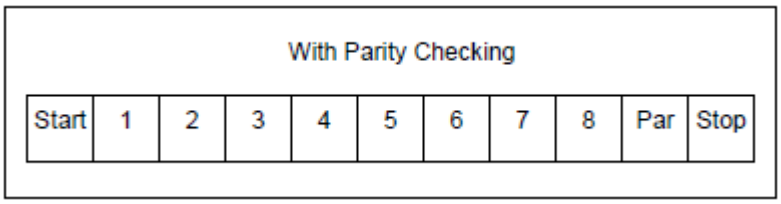


Figure 3 Bit sequence in RTU mode.

Frame Checking Field:
Cyclical Redundancy Checking (CRC)

Frame description:

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 bytes CRC Low CRC Hi

Figure 4 RTU Message Frame.

=> The maximum size of a Modbus RTU frame is 256 bytes.

1.2.4 Modbus RTU Message Framing

A Modbus message is placed by the transmitting device into a frame that has a known beginning and ending point. This allows devices that receive a new frame to begin at the start of the message, and to know when the message is completed. Partial messages must be detected and errors must be set as a result.

In RTU mode, message frames are separated by a silent interval of at least 3.5 character times.

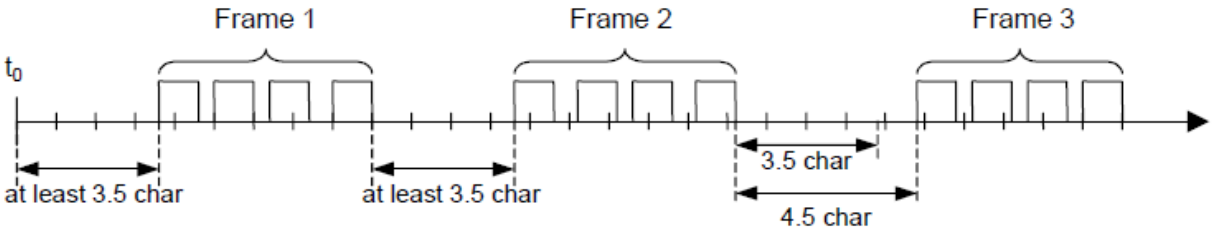


Figure 5 Valid frames with silent intervals.

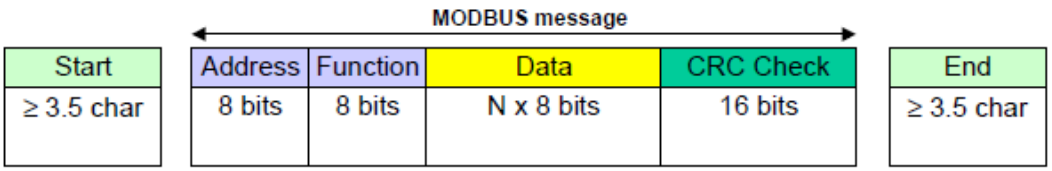


Figure 6 RTU Message Frame.

The entire message frame must be transmitted as a continuous stream of characters.

If a silent interval of more than 1.5 character times occurs between two characters, the message frame is declared incomplete and should be discarded by the receiver.

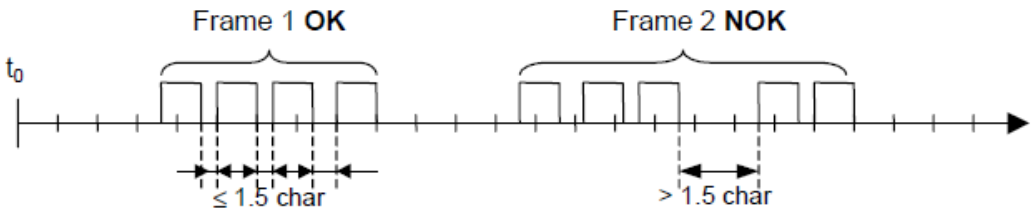


Figure 7 Data transmission of a frame.

1.2.5 Modbus RTU CRC Checking

The RTU mode includes an error-checking field that is based on a Cyclical Redundancy Checking (CRC) method performed on the message contents.

The CRC field checks the contents of the entire message. It is applied regardless of any parity checking method used for the individual characters of the message.

The CRC field contains a 16-bit value implemented as two 8-bit bytes.

The CRC field is appended to the message as the last field in the message. When this is done, the low-order byte of the field is appended first, followed by the high-order byte. The CRC high-order byte is the last byte to be sent in the message.

The CRC value is calculated by the sending device, which appends the CRC to the message. The receiving device recalculates a CRC during receipt of the message and compares the calculated value to the current value it received in the CRC field. If the two values are not equal, an error results.

The CRC calculation is started by first pre-loading a 16-bit register to all 1's. Then a process begins of applying successive 8-bit bytes of the message to the current contents or the register. Only the eight bits of data in each character are used for generating the CRC. Start and stop bits and the parity bit do not apply to the CRC.

During generation of the CRC, each 8-bit character is exclusive OR-ed with the register contents. Then the result is shifted in the direction of the least significant bit (LSB), with a zero filled into the most significant bit (MSB) position. The LSB is extracted and examined. If the LSB was a 1, the register is then exclusive OR-ed with a preset, fixed value. If the LSB was a 0, no exclusive OR takes place.

This process is repeated until eight shifts have been performed. After the last (eight) shift, the next 8-bit byte is exclusive OR-ed with the register's current value, and the process repeats for eight more shifts as described above. The final content of the register, after all the bytes of the message have been applied, is the CRC value.

When the CRC is appended to the message, the low-order byte is appended first, followed by the high-order byte.

A detailed introduction to CRC generation can be found in the manual
"MODBUS over Serial Line, Specification and Implementation Guide, V1.02"
in chapter 6.2 "Appendix B - LRC/CRC Generation" form http://www.modbus.org.

1.3 Implementation of Modbus RTU in Hamilton Arc Sensors

According to the official Modbus definition, the start of a command is initiated with a pause of ≥ 3.5 characters. Also, the end of a command is indicated with a pause of ≥ 3.5 char.
The device address and the Modbus function code have 8 bits.
The data string consists of n*8 bits. The data string contains the starting address of the register and the number of registers to read/write.
The checksum CRC is 16 bits long.

Table 1 Modbus definition for data transmission

	start	**device address**	**function**	**data**	**Checksum**		**end**
value	no signal during ≥ 3.5 char	1-32	function code according to Modbus specs	data according to Modbus specs	CRC L	CRC H	No signal during ≥ 3.5 char
bytes	≥ 3.5	1	1	n	1	1	≥ 3.5

The RS485 interface is configured as follows:

Table 2 RS485 definitions for Arc Sensors

Modbus RTU implementation in Hamilton Arc Sensors	
Start Bits	1
Data Bits	8
Parity	none
Stop Bit	2
String length	11 Bits
Baud Rate	19200 (default), other baud rate can be configured

1.4 Modbus RTU Function Codes Used for Arc Sensors

Arc Sensors use only 3 Modbus function codes:

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

These three function codes are described below in detail using excerpts from "Modbus Application Protocol Specification V1.1b "(http://www.modbus.org).

1.4.1 Modbus function code #3: Read Holding Registers

This function code is used to read the contents of a contiguous block of holding registers in a remote device. The Request PDU specifies the starting register address and the number or registers. The PDU Registers are addressed starting at zero. Therefore, registers numbered 1 – 16 are addressed as 0 – 15.

The register data in the response message are packed as two bytes per register. For each register, the first byte contains the high order bits and the second contains the low order bits.

Request

Function code	1 Byte	**0x03**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 125 (0x7D)

Response

Function code	1 Byte	**0x03**
Byte count	1 Byte	2 x **N***
Register value	**N*** x 2 Bytes	

***N** = Quantity of Registers

Error

Error code	1 Byte	**0x83**
Exception code	1 Byte	01 or 02 or 03 or 04

Figure 8 Definition of Holding Registers.

Request		**Response**	
Field Name	*(Hex)*	*Field Name*	*(Hex)*
Function	**03**	Function	**03**
Starting Address Hi	**00**	Byte Count	**06**
Starting Address Lo	**6B**	Register value Hi (108)	**02**
No. of Registers Hi	**00**	Register value Lo (108)	**2B**
No. of Registers Lo	**03**	Register value Hi (109)	**00**
		Register value Lo (109)	**00**
		Register value Hi (110)	**00**
		Register value Lo (110)	**64**

Figure 9 Example of reading holding registers 108 – 110. The contents of register 108 are read as the two-byte values 0x022B. The contents of registers 109 – 110 are 0x00 00 and 0x0064.

1.4.2 Modbus function code #4: Read Input Registers

The function code is used to read from 1 to 125 contiguous input registers in a remote device. The Request PDU specifies the starting register address and the number of registers. The PDU Registers are addressed starting at zero. Therefore, input registers numbered 1 – 16 are addressed as 0 – 15.

The register data in the response message are packed as two bytes per register. For each register, the first byte contains the high order bits and the second contains the low order bits.

Request

Function code	1 Byte	**0x04**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Input Registers	2 Bytes	0x0001 to 0x007D

Response

Function code	1 Byte	**0x04**
Byte count	1 Byte	2 x **N***
Input Registers	**N*** x 2 Bytes	

***N** = Quantity of Input Registers

Error

Error code	1 Byte	**0x84**
Exception code	1 Byte	01 or 02 or 03 or 04

Figure 10 Definition of Input Registers.

Request		Response	
Field Name	*(Hex)*	*Field Name*	*(Hex)*
Function	04	Function	04
Starting Address Hi	00	Byte Count	02
Starting Address Lo	08	Input Reg. 9 Hi	00
Quantity of Input Reg. Hi	00	Input Reg. 9 Lo	0A
Quantity of Input Reg. Lo	01		

Figure 11 Example of reading input register 9. The contents of input register 9 are read as the two-byte value 0x000A.

1.4.3 Modbus Function Code #16: Write Multiple Registers

This function code is used to write a block of contiguous registers (1 to 123 registers) in a remote device. The requested values are specified in the request data field. Data is packed as two bytes per register. The response returns the function code, starting address, and quantity of registers written.

Request

Function code	1 Byte	0x10
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	0x0001 to 0x007B
Byte Count	1 Byte	2 x **N***
Registers Value	**N*** x 2 Bytes	value

***N** = Quantity of Registers

Response

Function code	1 Byte	0x10
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 123 (0x7B)

Error

Error code	1 Byte	0x90
Exception code	1 Byte	01 or 02 or 03 or 04

Figure 12 Definition of Write Multiple Registers.

Request		Response	
Field Name	*(Hex)*	*Field Name*	*(Hex)*
Function	10	Function	10
Starting Address Hi	00	Starting Address Hi	00
Starting Address Lo	01	Starting Address Lo	01
Quantity of Registers Hi	00	Quantity of Registers Hi	00
Quantity of Registers Lo	02	Quantity of Registers Lo	02
Byte Count	04		
Registers Value Hi	00		
Registers Value Lo	0A		
Registers Value Hi	01		
Registers Value Lo	02		

Figure 13 Example of writing the value 0x000A and 0x0102 to two registers starting at address 2.

1.5 Data Formats Used in Arc Sensors

1.5.1 Float

Definition: Floating point according to IEEE 754 (Single Precision)

Table 3 Definition Floating point Single Precision (4 bytes resp. 2 Modbus registers)

Explanation:	sign	exponent	mantissa	total
Bit:	31	30 to 23	22 to 0	32
Exponent bias	127			

Example: translate the decimal value 62.85 into binary

Step 1: Conversion of the decimal value into binary fixed-point number

62 / 2	= 31	residue 0	LSB	**0.85** * 2	= 1.70	=> 1	MSB
31 / 2	= 15	residue 1		0.70 * 2	= 1.40	=> 1	
15 / 2	= 7	residue 1		0.40 * 2	= 0.80	=> 0	
7 / 2	= 3	residue 1		0.80 * 2	= 1.60	=> 1	
3 / 2	= 1	residue 1		0.60 * 2	= 1.20	=> 1	
1 / 2	= 0	residue 1	MSB	0.20 * 2	= 0.40	=> 0	
	= **111110**			0.40 * 2	= 0.80	=> 0	LSB
					…		
					= **0.11011001100110011001100**…		

62.85 = 111110.11011001100110011001100…

Step 2: Normalizing (in order to obtain 1 bit on the left side of the fraction point)

111110**.**11011001100110011001100… *2^**0** = 1**.**11110110110011001100110011001100… *2^**5**

Sep 3: Calculation of the dual exponent

2^5 => Exponent **5**
Exponent + Exponent bias = **5** + 127 = **132**

132 / 2	= 66	residue 0	LSB
66 / 2	= 33	residue 0	
33 / 2	= 16	residue 1	
16 / 2	= 8	residue 0	
8 / 2	= 4	residue 0	
4 / 2	= 2	residue 0	
2 / 2	= 1	residue 0	
1 / 2	= 0	residue 1	MSB
	= **10000100**		

Sep 4: Definition of the sign bit

Positive = 0
Negative = 1
= **0**

Step 5: conversion into floating-point

1 Bit Sign + 8 Bit Exponent + 23 Bit Mantissa
0 10000100 11110110110011001100110 (corresponds to 0x427B6666)

One important note for the 23 Bit Mantissa: The first bit (so-called hidden bit) is not represented. The hidden bit is the bit to the left of the fraction point. This bit is per definition always 1 and therefore suppressed.
Example: translate the binary float 0100 0010 0111 1011 0110 0110 0110 0110 to a decimal value

Step 1: Separating the binary value into Sign, Exponent and Mantissa

0 10000100 11110110110011001100110
1 Bit Sign + 8 Bit Exponent + 23 Bit Mantissa

S: **0** $_{binary}$ = **0** (positive sign)
E: **10000100** $_{binary}$ = $1*2^7 + 0*2^6 + 0*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 0*2^0$
= 128 + 0 + 0 + 0 + 0 + 4 + 0 + 0
= **132**
M: **11110110110011001100110** $_{binary}$ = **8087142**

Step 2: Calculate the decimal value

D = $(-1)^S * (1.0 + M/2^{23}) * 2^{E-127}$
= $(-1)^0 * (1.0 + 8087142/2^{23}) * 2^{132-127}$
= 1 * 1.964062452316284 * 32
= **62.85**

1.5.2 Character

Definition: The numerical representation of characters is defined in 8-Bit ASCII-Code-Table (ANSI X3.4-1986). Accordingly, each Modbus register in Arc Sensors can store two ASCII characters.

Example: translate the ASCII-string "2076" to Hex representation

The following interpretation is made according to the ASCII Codes-Table:

"2" => ASCII code table => 0x32 Low Byte
"**0"** => ASCII code table => 0x30
"**7"** => ASCII code table => 0x37
"**6"** => ASCII code table => 0x36 High Byte

"2076" => **0x36373032**

1.5.3 Decimal

Example: translate Decimal 2227169 to Hex

2227169 / 16	= 139198	residue 1 Low Byte
139198 / 16	= 8699	residue 14 => E
8699 / 16	= 543	residue 11 => B
543 / 16	= 33	residue 15 => F
33/16	= 2	residue 1
2 / 16	= 0	residue 2 High Byte
	= **0x21FBE1**	

1.6 Modbus RTU Error Messages

Here are listed the Modbus standard error-codes we have implemented in Arc Sensors.

Table 4 Implemented Error-Codes

Error-Code Hex	**Status-Text**
0x00	OK
0x01	Illegal function
0x02	Illegal data address
0x03	Illegal data value
0X04	Server device failure

See "Modbus_Application_Protocol_V1.1b" (www.modbus.org) for details.

If a server device failure exception occurs, try to repeat the command that has thrown the exception. If the exception remains, check the sensor status.

2 GlucoSense RS485 Commands in Modbus RTU

2.1 General

In order to communicate with a GlucoSense RS485 sensor over Modbus RTU protocol a Modbus client terminal application software is needed. The Modbus RTU is an open standard and a number of free and commercial application toolkits are available.

This manual contains examples and illustrations from WinTECH Modbus client ActiveX Control tool: WinTECH (www.win-tech.com) "Modbus Master OCX for Visual Basic". The Modbus Organization (www.modbus.org/tech.php) provides other links to a wide variety of Modbus terminal software.

In the present manual the addressing of the Modbus registers starts at 1. But the Modbus client protocol operates with register addresses starting at 0. Usually, the Modbus client software translates the addressing. Thus, the register address of 2090 will be translated by the Modbus client software to 2089 which is sent to the sensor (Modbus server).

 Attention:
When configuring and calibrating the sensor, please limit write operations to a reasonable number. More than 100'000 write operations will physically damage the memory of the sensor.

2.2 Operator levels and Passwords

2.2.1 Reading / Setting Operator Level

GlucoSense RS485 can be operated in three different operator levels. Each operator level allows a defined access to a specific set of commands.

Table 5 Definition of operator level and default passwords

Abbreviation	Description	Code (hex)	Password (default)	
			(decimal)	(hex)
U	User (lowest level)	0x03	0	0x00000000
A	Administrator	0x0C	18111978	0x01145DEA
S	Specialist	0x30	16021966	0x00F479CE

At each power up or processor reset, the operator level falls back to the default level U.

The active operator level can be read and written in register 4288.
The Password itself can't be read, the information in Reg 3/ Reg 4 is always read as zero.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (uint)	Modbus function code	Read access	Write access
4288	4	Operator Level	Password	3, 4, 16	U/A/S	U/A/S

 Attention:
If the password is wrong, the operator level falls back to operator level U. To make sure that the operator level switch was successful, read back register 4288.

2.2.2 Changing Passwords for Operator Level

The passwords for accessing the operator levels A and S can be modified by S (Specialist) only. U (User) and A (Administrator) have no right to change any password. If they try anyway, an illegal data address exception (0x02) is returned.
The new password will remain stored after power down.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (uint)	Modbus function code	Read access	Write access
4292	4	Level	New password	16	None	S

2.3 Configuration of the serial RS485 Interface

Factory settings for the RS485 interface:

Table 6 RS485 factory settings

Start Bits	1
Data Bits	8
Parity	None
Stop Bits	2
Baud Rate	19200

2.3.1 Device Address

2.3.1.1 Reading and Writing the Device Address

The sensor specific device address can be read and written in register 4096.

Start register	Number of registers	Reg1 / Reg2 (uint)	Modbus function code	Read access	Write access
4096	2	device address	3, 4, 16	U/A/S	S

The device address can be set by S (Specialist), default value is 1. If the address limits are not met when setting a new address, the former address stays active.

 Attention:

The device address changes immediately, what means that the next Modbus access must be done using the new address.

2.3.1.2 Reading the Device Address Limits

The device address limits can be read in register 4098.

Start register	Number of registers	Reg1 / Reg2 (uint)	Reg3 / Reg4 (uint)	Modbus function code	Read access	Write access
4098	4	Min. device address	Max. device address	3, 4	U/A/S	none

Device address limits are:
Minimal device address: 1
Maximal device address: 32

2.3.1.3 Broadcast

Independent from the selected device address, the sensor responds to broadcasted Modbus commands (address 0).

2.3.2 Baud Rate

2.3.2.1 Reading and Writing the Baud Rate

The baud rate can be read and written in register 4102.

Start register	Number of registers	Reg1 / Reg2 (uint)	Modbus function code	Read access	Write access
4102	2	Baud rate code (definition see below)	3, 4, 16	U/A/S	S

The code for the baud rate is defined as follows:

Table 7 Code for the baud rates.

Baud rate	4800	9600	19200	38400	57600	115200
Code	2	3	4	5	6	7

 Attention:

If the baud rate limits are not met when setting a new baud rate, the former baud rate stays active.
If the baud rate is valid, it will change immediately, which means that the next Modbus access must be done using the new baud rate.

2.3.2.2 Reading the Baud Rate Limits

The baud rate limits can be read in register 4104.

Start register	Number of registers	Reg1 / Reg2 (uint)	Reg3 / Reg4 (uint)	Modbus function code	Read access	Write access
4104	4	Min. Baud rate code	Max. Baud rate code	3, 4	U/A/S	none

The baud rate limits are:
Minimal baud rate code: 2
Maximal baud rate code: 7

2.3.3 Parity and Stop Bits

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
4108	2	Interface parameter	3, 4, 16	U/A/S	S

The interface parameter is coded as following:
0xAABBCCDD where
AA = no meaning (reading: 0x00)
BB = Parity (0x00: no parity, 0x01: even, 0x02: odd)
CC = Stop bits (0x00: 1 stop bit, 0x04: 2 stop bits)
DD = no meaning (reading 0x00)

The interface parameters do not change before the next power up!
When writing to register 4108 set Bytes AA and DD to 0x00.

 Attention:

If one of the parameter limits is not met, the old configuration stays active! Parity option (even or odd) is only available with one stop bit (max. string length of 11 bits).
The configuration changes immediately, what means that the next Modbus access must be done using the new baud rate.

2.4 Configuration of the Analog Interfaces (4-20 mA)

Note:

GlucoSense has no physical analog output interface. The 4-20mA analog interface is only provided by an Arc Wi Adapter BT, but the registers to configure these interfaces are available on the sensor. That means, that these registers can be read and written with or without Arc Wi Adapter BT. Therefore, the sensor has no error or warning information if the measurement values is below or above the limit of the analog interface.

2.4.1 Available Analog Interfaces

Equipped with an Arc Wi Adapter BT, the GlucoSense RS485 Sensor has two individual physical analog interfaces that have identical functionalities, but can be configured independently from each other.

- Analog Output Interface 1 (AO1)
- Analog Output Interface 2 (AO2)

The number of analog interfaces is defined in register 4320.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
4320	2	Available analog outputs	3, 4	U/A/S	none

The answer always is "0x03" meaning that there exists an Analog Interface 1 (AO1) and Analog Interface 2 (AO2).

2.4.2 Available Analog Interface Modes

With register 4322, the available analog interface modes for AO1 and AO2 are defined

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (bitwise defined)	Reg5 / Reg6	Reg7 / Reg8	Modbus function code	Read access	Write access
4322	8	Available Analog Interface Modes for AO1	Available Analog Interface Modes for AO2	reserved	reserved	3,4	U/A/S	none

Register 4322 defines the analog interface modes available for AO1 and AO2. The analog interface modes are described in Table 8.

Table 8 Definition of the analog interface modes

Code (Hex)	Analog Interface Mode	Description
0x0000	4-20 mA inactive	Analog interface is deactivated
0x0001	4-20 mA fixed	Set to a constant output value for current loop testing See 2.4.5.7
0x0002	4-20 mA linear	Linear output of measurement (PMC1 / 6)
0x0004	4-20 mA bilinear	Bilinear output of measurement (PMC1 / 6)

The answer is a bitwise combination (OR) of the available modes defined in Table 8.
Reg1/Reg2 return "0x0007" meaning that 4-20 mA fixed, 4-20 mA linear and 4-20 mA bilinear mode are available.
Reg3/Reg4 return "0x0007" meaning that 4-20 mA fixed, 4-20 mA linear and 4-20 mA bilinear mode are available.
Reg5/Reg6 return always zero.
Reg7/Reg8 return always zero.
How to select or change the analog interface mode, see 2.4.4.

2.4.3 Description of the Analog Interface

Register 46010 / 46138 contain the descriptions of AO1 / AO2 as plain text ASCII. AO1 for GlucoSense RS485 is called "mA interface #1". AO2 for GlucoSense RS485 is called "mA interface #2".

Start register	Number of registers	Reg1 to Reg8 16 ASCII characters	Modbus function code	Read access	Write access
46010	8	Description of the analog interface AO1	3, 4	U/A/S	none
46138	8	Description of the analog interface AO2	3, 4	U/A/S	none

2.4.4 Selection of an Analog Interface Mode

The analog interface mode of AO1 / AO2 is selected by programming the analog interface mode in register 46018/ 46146.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
46018	2	Analog interface mode for AO1	3, 4, 16	U/A/S	S
46146	2	Analog interface mode for AO2	3, 4, 16	U/A/S	S

For available analog interface modes are see register 4322.

Only one bit can be set. Using not allowed interface mode codes will leave the selection unchanged.

2.4.5 Configuration of the 4-20 mA Interface

Note:
The configuration of the 4-20 mA interface is only effective if register 46018 / 46146 (analog interface mode) is set to the value 0x01, 0x02 or 0x04 (see chapter 2.4.4).

2.4.5.1 Reading the Available Primary Measurement Channels to be Mapped to the Analog Output

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
46020	2	Available Primary Measurement Channels for AO1	3, 4	U/A/S	none
46148	2	Available Primary Measurement Channels for AO2	3, 4	U/A/S	none

Table 9 Code for selection of the primary measurement channel

Code (Hex)	Primary Measurement Channel (PMC)
0x01	PMC1
0x20	PMC6

Reading the available Primary Measurement Channels (PMC) always return the hexadecimal value of "0x21" meaning that PMC1 or PMC6 can be mapped to AO1 respectively AO2.

2.4.5.2 Selecting the Primary Measurement Channel to be Mapped to the Analog Interface

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
46022	2	Selected PMC for AO1	3, 4, 16	U/A/S	S
46150	2	Selected PMC for AO2	3, 4, 16	U/A/S	S

Write this register to change the mapped measurement channel to AO1 / AO2. Make sure that only one bit is set, according to Table 9. Writing 0 or an illegal code will leave the selection unchanged. Only one bit can be set!
Reading this register returns the selected PMC for AO1 / AO2 according to Table 9.

The factory setting for register 46022 is "0x01" mapping PMC1 to AO1.
The factory setting for register 46150 is "0x20" mapping PMC6 to AO2.

2.4.5.3 Reading the Minimal and Maximal Possible Physical Output Current

Register 46024 / 46142 delivers the limits of the physical output current for AO1/AO.

Start register	Number of registers	Reg1 / Reg2 (float)	Reg3 / Reg4 (float)	Modbus function code	Read access	Write access
46024	4	Min physical output current for AO1 [mA]	Max physical output current for AO1 [mA]	3, 4	U/A/S	none
46152	4	Min physical output current for AO2 [mA]	Max physical output current for AO2 [mA]	3, 4	U/A/S	none

The limits are fixed to:
Minimum is 3.5mA
Maximum is 22 mA



Note:

Currents above 20 and below 4 mA indicate erroneous measurements or errors.

2.4.5.4 Reading the Minimum, Maximum and Mid Current for Measurement Value Output

Start register	Number of registers	Reg1 / Reg2 (float)	Reg3 / Reg4 (float)	Reg5 / Reg6 (float)	Modbus function code	Read access	Write access
46028	6	Min output for measurement value for AO1 [mA]	Max output for measurement values for AO1 [mA]	Mid output for measurement values for AO1 [mA]	3, 4	U/A/S	none
46156	6	Min output for measurement value for AO2 [mA]	Max output for measurement values for AO2 [mA]	Mid output for measurement values for AO2 [mA]	3, 4	U/A/S	none

These registers deliver the minimal, maximal and middle output current for AO1/AO2 in mA during normal operation. They are fixed to 4, 20 and 12 mA.

Note:
Mid current must always be defined. However, in linear output mode, the mid current value has no physical meaning and will not affect the 4-20 mA output.

2.4.5.5 Reading the Selected Physical Unit for Analog Interface

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
46034	2	Selected physical unit of AO1 (see Table 14)	3, 4	U/A/S	none
46162	2	Selected physical unit of AO2 (see Table 14)	3, 4	U/A/S	none

Read the selected unit of the selected PMC of AO1/AO2. The value returned has to be interpreted as a hex value that represents the unit according to Table 14.
The physical unit for the PMC is defined in Reg. 2090 or 2410 and applies automatically for 4-20 mA output.

2.4.5.6 Defining the Measurement Values for 4, 12 and 20 mA Output

Start register	Number of registers	Reg1 / Reg2 (float)	Reg3 / Reg4 (float)	Reg5 / Reg6 (float)	Modbus function code	Read access	Write access
46036	6	Measurement value at Min Output Current (4 mA) for AO1	Measurement value at Max Output Current (20 mA) for AO1	Measurement value at Mid Output Current (12 mA) for AO1	3, 4, 16	U/A/S	S
46164	6	Measurement value at Min Output Current (4 mA) for AO2	Measurement value at Max Output Current (20 mA) for AO2	Measurement value at Mid Output Current (12 mA) for AO2	3, 4, 16	U/A/S	S

These registers define the relation between measurement value and output current in linear mode. Reg1/Reg2 define the measurement value at 4mA and Reg3/Reg4 define the measurement value at 20mA. Reg5/Reg6 do not affect the 4-20mA output. When writing, write 0 or any random value. When reading, Reg5/Reg6 return half of Min + Max.
The corresponding physical unit can be read in register 46034/ 46162 respectively in the corresponding PMC register (2090 for PMC1 and 2410 for PMC6).

Attention:

When assigning measurement values to 4-20 mA analog output by using register 46036/ 46164 you need to consider the following:

- The PMC you have mapped to AO1 / AO2 (register 46022 / 46150)
- The physical unit currently in use for the selected PMC (register 2090 for PMC1, and register 2410 for PMC6.

Therefore, when the operator redefines any of the above-mentioned register, the definitions of the register 46034 / 46162 should be reviewed. If not, the current output at the 4-20 mA interfaces may suddenly be unexpected!

Note:

Mid current must always be defined. However, in linear output mode, the mid current value has no physical meaning and will not affect the 4-20 mA output.

Example 1:
Register 46022 is set to 1 (PMC1 is mapped to AO1).
Register 2090 is set to 0x01000000 (the unit "mmol/l" is assigned to PMC1).
Register 46018 is set to 0x00000002 (linear analog output of PMC1).
Register 46036 is set to 0 and 2000 (4 mA = 0 mmol/l, 20 mA = 2000 mmol/l) as shown in Figure 14.
The sensor is placed in a bioreactor and currently reads a value of 900 mmol/l glucose concentration, the output at the 4-20 mA accordingly is 11.2 mA (900 mmol/l).
The operator now re-assigns register 2090 to the value of 0x00000020 (mg/l) but does not modify all other registers. The sensor reads now 162144 mg/l. At the analog output, as 20 mA is programmed to a value of 2000 by register 4378, the current will go to the maximum value of 20 mA. This will not generate an interface warning.

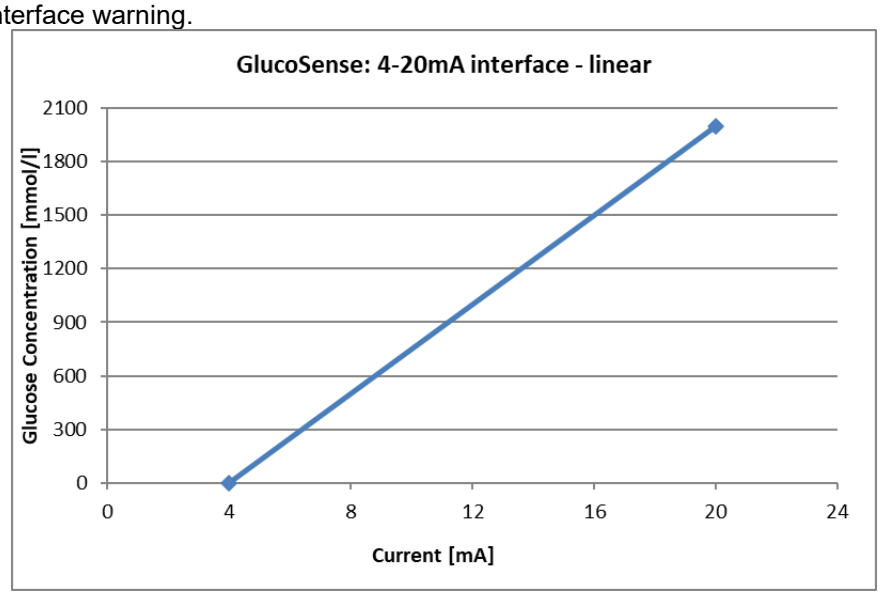


Figure 14 Example of linear 4-20mA output characteristics

Example 2:
Register 46022 is set to 1 (PMC1 is mapped to AO1).
Register 2090 is set to 0x00000100 (the unit "g/l" is assigned to PMC1).
Register 46018 is set to 0x00000004 (bilinear analog output of PMC1).
Register 46036 is set to 0, 62.85 and 10.0 (4 mA = 0 g/l, 20 mA = 62.85 g/l, 12 mA = 10.0 g/l) as shown in Figure 15.
The sensor is placed in a bioreactor and currently reads a value of 20.95 g/l glucose concentration, the output at the 4-20 mA accordingly is 13.66 mA (20.95 g/l).

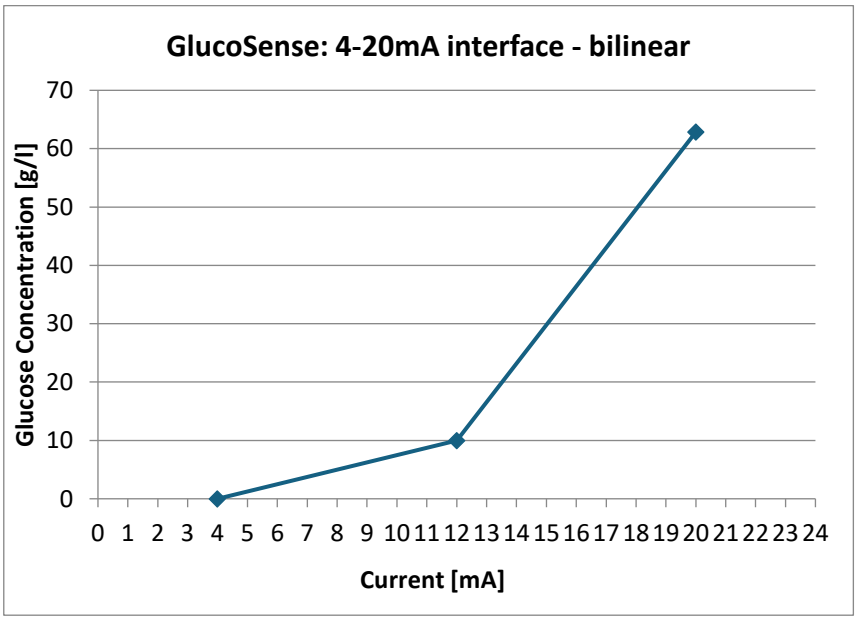


Figure 15 Example of bilinear 4-20mA output characteristics



Attention:
When assigning measurement values to 4-20 mA analog output by using register 46036 / 46164 you need to consider the following:

- The PMC you have mapped to 4-20 mA analog output (register 46020 / 46148)
- The unit currently in use for the selected PMC (register 2090 for PMC1 (glucose) and register 2410 for PMC6 (temperature).

Therefore, when the operator redefines one of the registers 46022 / 46150 / 2090 or 2410, the definitions of the register 46036 / 46164 should be reviewed. If not, the current output at the 4-20 mA interface may be wrong.

The 4-20 mA analog output freezes if

- the measure mode is in Hold Mode, see chapter 2.5
  or
- the current temperature is outside the user defined measurement temperature range (reg. 4624).

Note:
The physical unit of the analog output corresponds always to the unit that is set for the selected PMC (register 2090 for PMC1 or register 2410 for PMC6).

2.4.5.7 Defining a Constant Current Output for Testing

Note:

For constant current output, the AO1 / AO2 must be set to analogue interface mode 4-20 mA fixed (0x01) (see Table 8):

Start register	Number of registers	Reg1 / Reg2 (float)	Modbus function code	Read access	Write access
46042	2	Constant current output value for AO1 [mA]	3, 4, 16	U/A/S	S
46170	2	Constant current output value for AO2 [mA]	3, 4, 16	U/A/S	S

Values lower than 4mA respectively higher than 20mA will automatically be set within the limits.

2.4.5.8 Defining the Error and Warning Output of the 4-20 mA Interface

Errors and warnings can be mapped to the AO1 / AO2.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (float)	Reg5 / Reg6 (float)	Reg7 / Reg8 (float)	Modbus function code	Read access	Write access
46044	8	Code of warnings and errors for AO1	Current in case of "warning" for AO1 [mA]	Current in case of "error" for AO1 [mA]	Current in case of "T exceed" for AO1 [mA]	3, 4, 16	U/A/S	S
46172	8	Code of warnings and errors for AO2	Current in case of "warning" for AO2 [mA]	Current in case of "error" for AO2 [mA]	Current in case of "T exceed" for AO2 [mA]	3, 4, 16	U/A/S	S

Table 10 Code for the 4-20 mA interface in case of errors and warnings.

Code (Hex)	Behavior of the 4-20 mA interface in case of errors and warnings
0x000001	Error continuous output
0x010000	Warning continuous output

If the corresponding bits for the errors and warnings are not set (=0), the respective options are inactive.
"T exceed" is always active. What means that in case of a measurement temperature limit violation, the output current will be as the specified value. "T exceed" values lower than 3.5mA respectively higher than 4mA will automatically be set within the limits.

The "T exceed" limits are:
Minimal current value: 3.5 mA
Maximal current value: 4 mA

The default settings are:

- Code 0x01
- current in case of warnings: 3.5 mA
- current in case of errors: 3.5 mA
- current in case of measurement temperature limits violation: 3.5 mA

Table 11 Example: Read the settings for AO1 in case of warnings and errors

Command: ErrorWarnings AO1		Modbus address: **46044**	Length: **8**	Type: **3** Read
Parameter:	Warning code	Current in case of warning [mA]	Current in case of error [mA]	Current in case of temperature exceed [mA]
Format:	Hex	Float	Float	Float
Value:	**0x010001**	**3.5**	**3.5**	**3.5**

Warning code 0x010001 corresponds to the continuous output current in case of warning (0x010000) and continuous output current in case of error (0x01) of 3.5 mA. The output current in case of temperature exceed is 3.5 mA.

For more information about warnings, errors and temperature limits, see chapter 2.9.

2.4.6 Reading the Internally Calculated Output Current

Reg. 46072/ 46200 provides internal calculated output current of AO1 / AO2. These values are helpful to compare against the externally measured electrical current.

Start register	Number of registers	Reg1 / Reg2 (float)	Reg3 / Reg4 (float)	Modbus function code	Read access	Write access
46072	4	Set point [mA] AO1	Internally measured [mA] AO1	3, 4	U/A/S	none
46200	4	Set point [mA] AO2	Internally measured [mA] AO2	3, 4	U/A/S	none



Attention:

The current outputs are provided by the Arc Wi Adapter BT. Therefore, the sensor cannot internally measure any output currents. Reg3/Reg4 always deliver the same value as Reg1/Reg2 even though there is no Arc Wi Adapter BT connected. This is due to compatibility to other Arc Sensors.

Measurement

2.4.7 Definition of Measurement Channels and Physical Units

The Arc Modbus register structure allows the definition of 6 individual Primary Measurement Channels (PMC), and 16 individual Secondary Measurement Channels (SMC). The SMCs describe the raw measured values from which the PMCs are then calculated.

Table 12 Definition of PMC1 to 6 and SMC1 to 16

Code (Hex)	Description	Definition
0x000001	PMC1	Glucose concentration
0x000020	PMC6	Temperature
0x000040	SMC1	Temperature act
0x002000	SMC8	S1
0x004000	SMC9	S2
0x008000	SMC10	S3
0x010000	SMC11	S4

In Register 2048, the available PMC and SMC are defined for a specific sensor and a specific operator level.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
2048	2	Available measurement channels PMC and SMC (bitwise set)	3, 4	U/A/S	none

Table 13 Example to read Reg. 2048

Command: Avail. meas. channels	Modbus address: **2048**	Length: **2**	Type: **3**	Read
Parameter:	Avail. meas. ch. PMC and SMC			
Format:	Hex			
Value:	**0x21**			

In case of operator level U and A, the value 0x21 is returned. In other words: PMC1 and PMC6 are available to the operator U and A. In case of operator level S, the value 0x1E061 is returned. In other words: PMC1, PMC6, SMC1, SMC8, SMC9, SMC10 and SMC11 are available to the operator S.

Using SMC8 to SMC11, PMC1 is calculated using the calibration coefficients g1 to g5 (2.8.2.5.7):

$$PMC1 = g_5 + \sum_{i=1}^{4} g_i S_i$$

The GlucoSense RS485 Modbus register structure uses the following physical units used for Primary or Secondary Measurement Channels.

Table 14 Definition of physical units

Code (Hex)	Physical unit	Start register. (8 ASCII characters, length 4 registers, Type 3, read for U/A/S)
0x00000001	none	1920
0x00000002	K	1924
0x00000004	°C	1928
0x00000008	°F	1932
0x00000010	PCV	1936
0x00000020	mg/l	1940
0x00000040	mg/dl	1944
0x00000080	NTU	1948
0x00000100	g/l	1952
0x00000200	uS/cm	1956
0x00000400	mS/cm	1960
0x00000800	1/cm	1964
0x00001000	%	1968
0x00002000	pF	1972
0x00004000	kOhm	1976
0x00008000	MOhm	1980
0x00010000	pA	1984
0x00020000	nA	1988
0x00040000	uA	1992
0x00080000	mA	1996
0x00100000	uV	2000
0x00200000	mV	2004
0x00400000	V	2008
0x00800000	CFU	2012
0x01000000	mmol/l	2016
0x02000000	Ohm	2020
0x04000000	%/K	2024
0x08000000	°	2028
0x10000000	E6 c/ml	2032
0x20000000	percent	2036
0x40000000	kHz	2040
0x80000000	SPECIAL	2044

Table 15 Example to read the physical unit in plain text ASCII in register 1952

Command: Unit text		Modbus address: **1952**	Length: **4**	Type: **3**	Read
Parameter:	Text				
Format:	8 ASCII characters				
Value:	**"g/l"**				

2.4.8 Primary Measurement Channel 1 (Glucose Concentration)

2.4.8.1 Definition of PMC1

Start register	Data size (registers)	Function	Data type	Modbus function code	Read access	Write access
2080	8	Description of PMC1	ASCII chars	3, 4	U/A/S	none
2088	2	Available physical units of PMC1	uint	3, 4	U/A/S	none
2090	2	Selected physical unit for PMC1	uint	16	none	S

In register 2080, a plain text ASCII description of PMC1 is given. PMC1 for GlucoSense RS485 is called "GLC".

In register 2088, the available physical units for this channel are defined. The available physical units for Calibration: 0x01000160 → g/l, mg/dl, mg/l and mmol/l.

In register 2090, the active physical unit for this channel can be selected, by choosing one of the physical units that are defined in register 2088.
Selecting an invalid unit code will leave the current unit unchanged.

Table 16 Example to set the physical unit of PMC1 to g/l (0x00000100)

Command: PMC1 set unit		Modbus address: **2090**	Length: **2**	Type: **16**	Write
Parameter:	Unit				
Format:	Hex				
Value:	**0x00000100**				



Attention:
Changing the physical unit of PMC1 has also an influence on the output of the 4-20 mA analog output, as the same physical unit is active for 4-20 mA. All limits of the 4-20 mA analog output must be redefined after changing the physical unit! See chapter 2.4.5 for more details.

Example 1: Reading the available physical units of PMC1 (Glucose concentration)

Read the available physical units of PMC1 at register 2088 with length 2 and function-code 3.

Request:

The command in HEX-format sent to the sensor:
0x 01 03 08 27 00 02 76 60

0x 01 Server address (decimal "01")
0x 03 Function code "Read Holding Registers"
0x 08 27 Starting address (decimal "2087")
0x 00 02 Number of registers (decimal "02")
0x 76 60 CRC

Response:

The answer received in HEX-format from the sensor:
0x 01 03 04 01 60 01 00 FA 41

0x 01 Server address (decimal "01")
0x 03 Function code "Read Holding Registers"
0x 04 Data-Byte count (decimal "04")
0x 01 00 Register 1 (decimal "352") Available physical unit – low register
0x 01 60 Register 2 (decimal "256") Available physical unit – high register
0x FA 41 CRC
Registers 2 and 1 together result in 0x 01600100 which represents the available physical units g/l (0x00000100), mg/l (0x00000020), mg/dl (0x00000040) and mmol/l (0x01000000) see chapter 2.5.1

Example 2: Writing the physical unit of PMC1 (Glucose concentration)

Writing to register 2090 with length 2 and function-code 16:

Request:

The command in HEX-format sent to the sensor:
0x 01 10 08 29 00 02 04 01 00 00 00 57 E1

0x 01 Server address (decimal "01")
0x 10 Function code "Write Multiple Registers"
0x 08 29 Starting address (decimal "2089")
0x 00 02 Quantity of registers (decimal "2")
0x 04 Byte count (decimal "4")
0x 01 00 Register 1 (hex) Physical unit – low register
0x 00 00 Register 2 (hex) Physical unit – high register
0x 57 E1 CRC

Response:

The answer received in HEX-format from the sensor:
0x 01 10 08 29 00 02 92 60

0x 01 Server address (decimal "01")
0x 10 Function code "Write Multiple Registers"
0x 08 29 Starting address (decimal "2089")
0x 00 02 Quantity of registers (decimal "2")
0x 92 60 CRC

2.4.8.2 Reading the measurement value of PMC1

Register 2090 is also used to read the measurement values of PMC1.

Start reg.	Num-ber of reg.	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (float)	Reg5 / Reg6 (bitwise defined)	Reg7 / Reg8 (float)	Reg9 / Reg10 (float)	Modbus function code	Read access	Write access
2090	10	Selected physical unit	Measure-ment value of PMC1 (1)	Measure-ment status (2)	Min allowed value (1)	Max allowed value (1)	3, 4	U/A/S	none

(1) Value is always in the physical unit defined in register 2090.
(2) Definition of the status see chapter 2.4.9. All bits set to zero means: no problem.

 Attention:

You cannot read selectively the registers 3 and 4 for the measurement value only. You must read the entire length of the command (10 registers) and extract the desired information.
For the definition of the Measurement Status see chapter 2.4.9.

Example: Reading PMC1 (Glucose concentration)

Reading register 2090 with length 10 and function-code 3 the five different values at once:

- Physical unit (registers 1 and 2) interpret as hex value
- Measurement value PMC1 (registers 3 and 4) transform to a floating-point value
- Status (registers 5 and 6) interpret as hex value
- Min allowed value (registers 7 and 8) transform to a floating-point value
- Max allowed value (registers 9 and 10) transform to a floating-point value

Request:

The command in HEX-format send to the sensor:
0x 01 03 08 29 00 0A 16 65

0x 01 Server address (decimal "01")
0x 03 Function code "Read Holding Registers"
0x 08 29 Starting address (decimal "2089")
0x 00 0A Number of registers (decimal "10")
0x 16 65 CRC

Response:

The answer received in HEX-format from the sensor:
0x 01 03 14 01 00 00 00 7B C4 41 A8 00 00 00 00 00 00 00 00 00 00 41 C8 D7 58

0x 01 Server address (decimal "01")
0x 03 Function code "Read Holding Registers"
0x 14 Data-Byte count (decimal "20")
0x 01 00 Register 1 (hex) Physical unit – low register
0x 00 00 Register 2 (hex) Physical unit – high register
0x 7B C4 Register 3 (floating point) Measurement value PMC1 – low register
0x 41 A8 Register 4 (floating point) Measurement value PMC1 – high register
0x 00 00 Register 5 (hex) Status – low register
0x 00 00 Register 6 (hex) Status – high register
0x 00 00 Register 7 (floating point) Min allowed value – low register
0x 00 00 Register 8 (floating point) Min allowed value – high register
0x 00 00 Register 9 (floating point) Max allowed value – low register
0x 41 C8 Register 10 (floating point) Max allowed value – high register
0x D7 58 CRC

Now extract the desired information:

- The physical unit 0x00000100 of registers 2 and 1 represents g/l, see chapter 2.4.7.
- The measurement value 0x41C80000 of registers 4 and 3 transformed to a floating-point figure results in the decimal value 21.06043.
- The status 0x00000000 of registers 6 and 5 indicate that there is no error or warning.
- The min allowed value 0x00000000 in registers 8 and 7 transformed to a floating-point figure results in the decimal value 0.
- The max allowed value 0x427BCF8D in registers 10 and 9 transformed to a floating-point figure results in the decimal value 25.0.

2.4.8.3 Primary Measurement Channel 6 (Temperature)

The GlucoSense RS485 sensor has a built-in temperature sensor. This temperature sensor is to be used only for monitoring the sensor conditions, not for controlling the process temperature.

2.4.8.4 Definition of PMC6

Start register	Data size (registers)	Function	Data type	Modbus function code	Read access	Write access
2400	8	Description of PMC6	ASCII chars	3, 4	U/A/S	none
2408	2	Available physical units of PMC6	uint	3, 4	U/A/S	none
2410	2	Selected physical unit for PMC6	uint	16	none	S

In register 2400, a plain text ASCII description of PMC6 is given. PMC6 is called "T".

In register 2408, the available physical units for this channel are defined. The available physical units for PMC6: 0x0000000E => K, °C and °F

In register 2410, the active physical unit for this channel can be selected, by choosing one of the physical units that are defined in register 2408.
Selecting an invalid unit code will leave the current unit unchanged.

 Attention:

Changing the physical unit of PMC6 has also an influence on the output of AO1 / AO2, as the same physical unit is active for the analog outputs. All limits of the 4-20 mA analog output must be redefined after changing the physical unit! See chapter 2.4.5 for more details.

2.4.8.5 Reading the measurement value of PMC6

Register 2410 is also used to read the measurement values of PMC6.

Start reg.	Num-ber of reg.	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (float)	Reg5 / Reg6 (bitwise defined)	Reg7 / Reg8 (float)	Reg9 / Reg10 (float)	Modbus function code	Read access	Write access
2410	10	Selected physical unit	Measure-ment value of PMC6 (1)	Measure-ment status (2)	Min allowed value (1)	Max allowed value (1)	3, 4	U/A/S	none

(1) Value is always in the physical unit defined in register 2410, length 2.
(2) For definition of the status see chapter 2.4.9. All bits set to zero means: no problem.

Table 17 Example to read register 2410

Command: PMC6 read		Modbus address: **2410**		Length: **10**	Type: **3** Read
Parameter:	Unit	Value	Status	Min limit	Max limit
Format:	Hex	Float	Hex	Float	Float
Value:	**0x04**	**27.42447**	**0x00**	**-20**	**140**

Physical unit is set to °C, PMC6 is 27.42 (°C), Status is 0x00, Min allowed value is -20 °C, Max allowed value is 140 °C.

 Attention:

You cannot read selectively the registers 3 and 4 for the measurement value only. You must read the entire length of the command (10 registers) and extract the desired information.

Example: Reading PMC6 (Temperature)

Reading register 2410 with length 10 and function-code 3:

- Physical unit (registers 1 and 2) interpret as hex value
- Measurement value PMC6 (registers 3 and 4) transform to a floating-point value
- Status (registers 5 and 6) interpret as hex value
- Min allowed value (registers 7 and 8) transform to a floating-point value
- Max allowed value (registers 9 and 10) transform to a floating-point value

Request:

The command in HEX-format send to the sensor:
0x 01 03 09 69 00 0A 16 4D

0x 01 Server address (decimal "01")
0x 03 Function code "Read Holding Registers"
0x 09 69 Starting address (decimal "2409")
0x 00 0A Number of registers (decimal "10")
0x 16 4D CRC

Response:

The answer received in HEX-format from the sensor:
0x 01 03 14 00 04 00 00 2A E0 41 D1 00 00 00 00 00 00 C2 20 00 00 43 02 70 E5

0x 01 Server address (decimal "01")
0x 03 Function code "Read Holding Registers"
0x 14 Data-Byte count (decimal "20")
0x 00 04 Register 1 (hex) Physical unit – low register
0x 00 00 Register 2 (hex) Physical unit – high register
0x 2A E0 Register 3 (floating point) Measurement value PMC6 – low register
0x 41 D1 Register 4 (floating point) Measurement value PMC6 – high register
0x 00 00 Register 5 (hex) Status – low register
0x 00 00 Register 6 (hex) Status – high register
0x 00 00 Register 7 (floating point) Min allowed value – low register
0x C2 20 Register 8 (floating point) Min allowed value – high register
0x 00 00 Register 9 (floating point) Max allowed value – low register
0x 43 02 Register 10 (floating point) Max allowed value – high register
0x 70 E5 CRC
Now extract the desired information:

- The physical unit 0x00000004 of registers 2 and 1 represents °C, see chapter 2.4.7.
- The measurement value 0x41D12AE0 in registers 4 and 3 transformed to a floating-point figure represents the decimal value 26.14594.
- The status 0x00000000 of registers 6 and 5 indicates that there is no error or warning
- The min allowed value 0xC2200000 in registers 8 and 7 transformed to a floating-point figure represents the decimal value -40.
- The max allowed value 0x43020000 in registers 10 and 9 transformed into a floating-point figure represents the decimal value 130.

2.4.9 Definition of the Measurement Status for PMC1 / PMC6

This is the definition of the status registers read in registers 2090 (PMC1) and 2410 (PMC6):

Table 18 Definition of measurement status for Primary Measurement Channels

Code (Hex)	Description
0x01	Temperature out of user defined measurement temperature range (see chapter 2.9.1)
0x02	Temperature out of operating range (see chapter 2.9.1)
0x04	Not available
0x08	Warning not zero (see chapter 2.9.3)
0x10	Error not zero (see chapter 2.9.4)

2.4.10 Secondary Measurement Channel 1 (T act)

2.4.10.1 Definition of SMC1

Start register	Data size (registers)	Function	Data type	Modbus function code	Read access	Write access
2464	8	Description of SMC1	ASCII chars	3, 4	U/A/S	none

In register 2465, a plain text ASCII description of SMC1 is given. SMC1 for GlucoSense RS485 is called "T act". In contrast to PMC6, the PA9: Moving Average has no influence on SMC1.

This channel has unit K (0x00000002).

2.4.10.2 Reading the measurement value of SMC1

Register 2472 is used to read the measurement values of SMC1 with or without the limits.

Start reg.	Num-ber of reg.	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (float)	Reg5 / Reg6	Reg7 / Reg8 (float)	Reg9 / Reg10 (float)	Modbus function code	Read access	Write access
2472	6	Physical unit	Measure-ment value of SMC1	Not used	Not used	Not used	3, 4	U/A/S	none
2472	10	Physical unit	Measure-ment value of SMC1	Not used	Min allowed value	Max allowed value	3, 4	U/A/S	none

Attention:

You cannot read selectively the registers 3 and 4 for the measurement value only. You must read the entire length of the command (6 or 10 registers) and extract the desired information.

2.4.11 Secondary Measurement Channel 8 (S1)

2.4.11.1 Definition of SMC8

Start register	Data size (registers)	Function	Data type	Modbus function code	Read access	Write access
2688	8	Description of SMC8	ASCII chars	3, 4	U/A/S	none

In register 2688, a plain text ASCII description of SMC8 is given. SMC8 for GlucoSense RS485 is called "S1".

This channel has no unit (0x00000001).

2.4.11.2 Reading the measurement value of SMC8

Register 2696 is used to read the measurement values of SMC8 with or without the limits.

Start reg.	Num-ber of reg.	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (float)	Reg5 / Reg6	Reg7 / Reg8 (float)	Reg9 / Reg10 (float)	Modbus function code	Read access	Write access
2696	6	Physical unit	Measure-ment value of SMC8	Not used	Not used	Not used	3, 4	U/A/S	none
2696	10	Physical unit	Measure-ment value of SMC8	Not used	Min allowed value	Max allowed value	3, 4	U/A/S	none

 Attention:

You cannot read selectively the registers 3 and 4 for the measurement value only. You must read the entire length of the command (6 or 10 registers) and extract the desired information.

2.4.12 Secondary Measurement Channel 9 (S2)

2.4.12.1 Definition of SMC9

Start register	Data size (registers)	Function	Data type	Modbus function code	Read access	Write access
2720	8	Description of SMC9	ASCII chars	3, 4	U/A/S	none

In register 2720, a plain text ASCII description of SMC9 is given. SMC9 for GlucoSense RS485 is called "S2".

This channel has no unit (0x00000001).

2.4.12.2 Reading the measurement value of SMC9

Register 2728 is used to read the measurement values of SMC9 with or without the limits.

Start reg.	Num-ber of reg.	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (float)	Reg5 / Reg6	Reg7 / Reg8 (float)	Reg9 / Reg10 (float)	Modbus function code	Read access	Write access
2728	6	Physical unit	Measure-ment value of SMC9	Not used	Not used	Not used	3, 4	U/A/S	none
2728	10	Physical unit	Measure-ment value of SMC9	Not used	Min allowed value	Max allowed value	3, 4	U/A/S	none



Attention:

You cannot read selectively the registers 3 and 4 for the measurement value only. You must read the entire length of the command (6 or 10 registers) and extract the desired information.

2.4.13 Secondary Measurement Channel 10 (S3)

2.4.13.1 Definition of SMC10

Start register	Data size (registers)	Function	Data type	Modbus function code	Read access	Write access
2752	8	Description of SMC10	ASCII chars	3, 4	U/A/S	none

In register 2752, a plain text ASCII description of SMC10 is given. SMC10 for GlucoSense RS485 is called "S3".

This channel has no unit (0x00000001).

2.4.13.2 Reading the measurement value of SMC10

Register 2760 is used to read the measurement values of SMC10 with or without the limits.

Start reg.	Num-ber of reg.	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (float)	Reg5 / Reg6	Reg7 / Reg8 (float)	Reg9 / Reg10 (float)	Modbus function code	Read access	Write access
2760	6	Physical unit	Measure-ment value of SMC10	Not used	Not used	Not used	3, 4	U/A/S	none
2760	10	Physical unit	Measure-ment value of SMC10	Not used	Min allowed value	Max allowed value	3, 4	U/A/S	none

 Attention:

You cannot read selectively the registers 3 and 4 for the measurement value only. You must read the entire length of the command (6 or 10 registers) and extract the desired information.

2.4.14 Secondary Measurement Channel 11 (S4)

2.4.14.1 Definition of SMC11

Start register	Data size (registers)	Function	Data type	Modbus function code	Read access	Write access
2784	8	Description of SMC11	ASCII chars	3, 4	U/A/S	none

In register 2784, a plain text ASCII description of SMC11 is given. SMC11 for GlucoSense RS485 is called "S4".

This channel has no unit (0x00000001).

2.4.14.2 Reading the measurement value of SMC11

Register 2792 is used to read the measurement values of SMC11 with or without the limits.

Start reg.	Num-ber of reg.	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (float)	Reg5 / Reg6	Reg7 / Reg8 (float)	Reg9 / Reg10 (float)	Modbus function code	Read access	Write access
2792	6	Physical unit	Measure-ment value of SMC11	Not used	Not used	Not used	3, 4	U/A/S	none
2792	10	Physical unit	Measure-ment value of SMC11	Not used	Min allowed value	Max allowed value	3, 4	U/A/S	none



Attention:

You cannot read selectively the registers 3 and 4 for the measurement value only. You must read the entire length of the command (6 or 10 registers) and extract the desired information.

2.5 Measurement Hold Mode

The measurement hold mode is not persistently stored in the GlucoSense RS485, therefore on every sensor start up, the measurement hold mode is in the "on" mode (default mode).
A detailed description of the measurement hold mode is described in Figure 16.

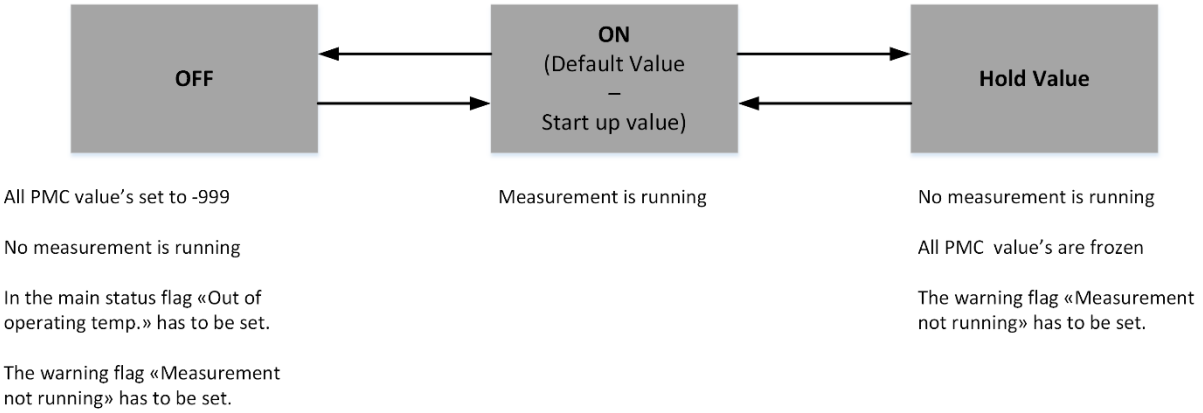


Figure 16: Description of the measurement hold mode functionality

The measurement hold mode for PMC1 and PMC6 is selected by programming the mode in register 2064.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
2064	2	Measurement Hold Mode, see Table 19	3, 4, 16	S	S

Table 19 Code for selection of the measurement hold mode

Code (Hex)	Description
0x00	Off • Values set to -999 • Measurement warning "Measurement not running" is set, see chapter 2.9.3
0x01	On • Current measured value output
0x02	Hold Value • Values are frozen • Measurement warning "Measurement not running" is set, see chapter 2.9.3

2.6 Configuration of the Measurement

This chapter describes the configuration of PMC1 and PMC6 by means of measurement parameters (PA). Only one parameter exists for GlucoSense RS485.

2.6.1 Available Parameters

In register 3072, all available parameters (PA) are given.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
3072	2	Available parameters (see Table 20)	3, 4	U/A/S	none

Table 20 Bitwise definition of all parameters PA1 to PA16

Code (Hex)	Description	Definition in GlucoSense RS485
0x0100	PA9	Moving average

Table 21 Example to read the available Parameters with operator level S

Command: Available parameters		Modbus address: **3072**	Length: **2**	Type: **3**	Read
Parameter:	Measurement parameters				
Format:	Hex				
Value:	**0x0100**				

The hex value 0x0100 corresponds to 0x0100 (PA9).

Note:

- PA1 to PA8 use FLOAT as data format for its values
- PA9 to PA16 use UNSIGNED INT as data format for its values

2.6.2 PA9: Moving Average

GlucoSense RS485 calculates new Glucose- concentrations readings with a measurement interval of one second. One has the possibility to smoothen the glucose and temperature reading (PMC1 and PMC6) by means of a moving average.
Figure 17 shows a comparison between no moving average (n=1) and a moving average over 50 readings at a measurement interval of 3 second.

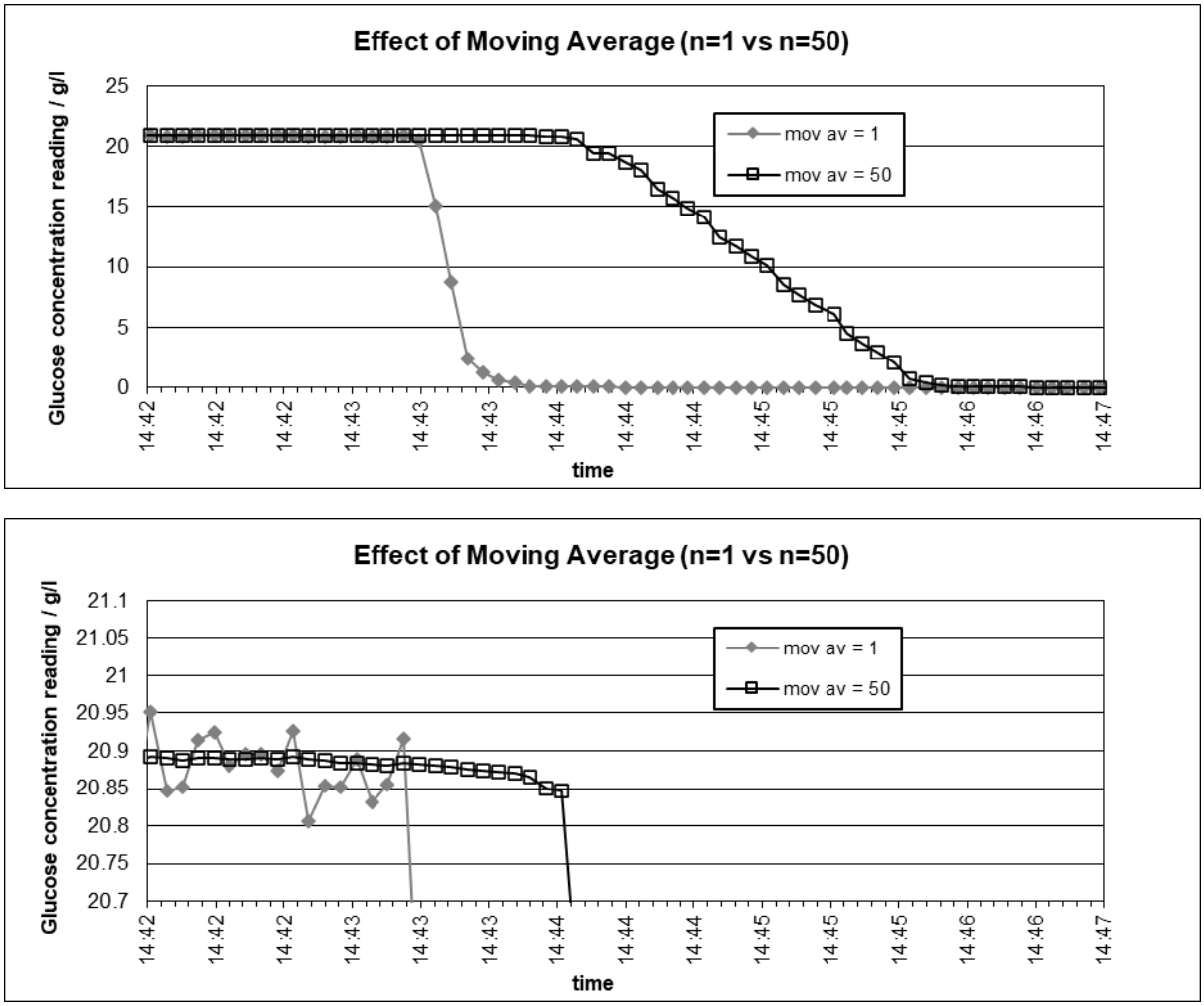


Figure 17 Comparison of the response of GlucoSense RS485 to a change from a glucose mixture to water

Using moving average, the short-term signal stability can be improved; on the other hand, the response time of the sensor increases with increasing moving average. A moving average over 50 samples results in a response time of at least 50 seconds.

Note:
The moving average defined by PA9 is applied to all primary measurement channels PMC1 and PMC6.

2.6.2.1 Description of PA9 (Moving Average)

In register 3360, a plain text ASCII description of PA9 is given. It returns "Moving average".

Start register	Number of registers	Reg1 to Reg8 16 ASCII characters	Modbus function code	Read access	Write access
3360	8	Description of PA9	3, 4	U/A/S	none

2.6.2.2 Selecting the Physical Unit and Writing the Value for PA9

In register 3368, the available physical units for PA9 are defined. The only one available here is "none" (0x01). For the definition of the physical units see Table 14.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
3368	2	Available physical units for PA9	3, 4	U/A/S	none

By writing to register 3370/4, the active physical unit for PA9 can be selected, by choosing one of the physical units that are defined in register 3368. According to register 3368 only one bit for the physical unit can be set. The value of the parameter can be set as well.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (uint)	Modbus function code	Read access	Write access
3370	4	Select physical unit for PA9	Value for PA9 (1-600, default: 10)	16	none	S

PA9 can take values between 1 and 600. The value of 1 does not influence the response time of the sensor, the value of 10 increases the response time for 10 seconds.

2.6.2.3 Reading all Values for PA9

By reading register 3370/8, the active physical unit of measurement, the current value, and the min and max values can be read.
The unit is always none (0x01), the min is 1 and the max is 600.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (uint)	Reg5 / Reg6 (uint)	Reg7 / Reg8 (uint)	Modbus function code	Read access	Write access
3370	8	Physical unit	Current value	Min value	Max value	3, 4	U/A/S	none

The default settings are:

- Physical unit of PA9: 0x00000001 (none)
- Current value of PA9: 10
- Min value of PA9: 1
- Max value of PA9: 600

2.7 Verification

Please note that GlucoSense RS485 can only be recalibrated by Hamilton. The purpose of this verification procedure is to determine whether the sensor requires recalibration. Regular verification is essential to ensure the accuracy of the sensor and to identify if recalibration is necessary.

The feature of verification must be switched on or off with the command in register 3684. The verification command is not stored persistently into the sensor, a restart of the sensor the verification command is automatically switched off.

To activate the verification function, it must first be enabled in register 3684 by writing the verification command code **0x00000001** (Start Verification), as shown in Table 22. The subsequent verification steps (Sections 2.7.1 to 2.7.5) can only be carried out after the verification function has been successfully activated.

By sending the verification command code **0x00000000** (Stop Verification) in register 3684 the verification feature is deactivated.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
3684	2	Verification Command (see Table 22)	3, 4, 16	U/A/S	S

Table 22: Bitwise definition of the verification command

Hex value	Description
0x00000000	Stop Verification
0x00000001	Start Verification

2.7.1 Initial Setup

- Immerse the sensor in setup solution and allow it to measure for a duration of 10 minutes. This step ensures the sensor is properly initialized.

2.7.2 Product Calibration

- After the 10-minute measurement in setup solution, perform a product calibration on the sensor to ensure accurate readings.

2.7.3 Glucose Standard Measurement

- Place the sensor into the glucose standard solution and let it measure for another 10 minutes. This step ensures the sensor is correctly calibrated for glucose measurements.

2.7.4 Verification of Tolerance

- After the 10-minute glucose measurement, check if the sensor's readings are within the tolerance range set by user.
- If the sensor is within this tolerance range, it is functioning properly.

2.7.5 Further Action

- If the sensor's readings fall outside of the ±1.0 g/L tolerance, contact Technical Support for further assistance.

2.8 Calibration

The GlucoSense RS485 sensor is calibrated at the Hamilton factory and can thereafter only be recalibrated by Technical Support. Only the product calibration must be carried out before each measurement. Product calibration can be performed for PMC1.

2.8.1 Available Calibration Points

In register 10256 the available number of Calibration Points (CP) for Primary Measurement Channel 1 (PMC1) are defined.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
10256	2	Available number of CP for PMC1 (see Table 23)	3, 4	U/A/S	none

Table 23 Bitwise definition of CP6

Code (Hex)	Description	Definition in GlucoSense RS485
0x40	CP6	Product calibration

Table 24 Example to read the available CPs of PMC1

Command: Available cali points		Modbus address: **10256**	Length: **2**	Type: **3**	Read
Parameter:	Points				
Format:	Hex				
Value:	**0x40**				

The hex value 0x40 in Table 24 corresponds to 0x40 (CP6).

The GlucoSense RS485 sensor allows only a product calibration (CP6)
As shown in Figure 18 the product calibration (CP6) is used to adjust the factory calibration function to specific process conditions. When performing a product calibration, an offset shift to the desired operating point is carried out

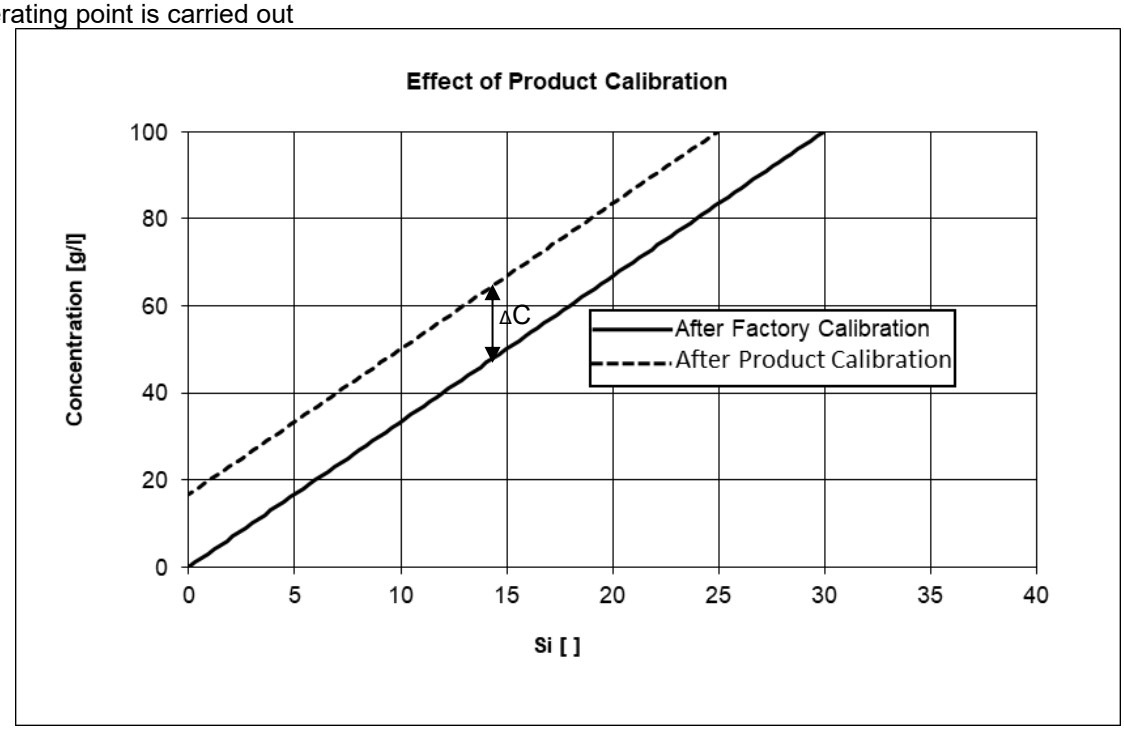


Figure 18 Effect of the product calibration CP6 on an existing factory calibration function defined by Hamilton.

2.8.2 Definitions of Calibration Points

2.8.2.1 Calibration Unit

The available calibration units for CP6 for PMC1 is defined in register 10272.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
10272	2	Available calibration units for PMC1, see chapter 2.4.7	3, 4	U/A/S	none
10274	2	Selected physical unit for calibration for PMC1, see chapter 2.4.7	3, 4	U/A/S	S

In register 10272 the available physical units for product calibration are defined. The available physical units for Calibration: 0x01000160 → mmol/l, g/l, mg/dl and mg/l

In register 10274 the active physical unit for the calibration can be selected, by choosing one of the physical units that are defined in register 10272. As only g/l is available for the Product calibration for the GlucoSense RS485 senor, only this unit can be selected.
Selecting an invalid unit code will leave the current unit unchanged.

Table 25 Example to set the physical calibration unit to g/l (0x00000100) for PMC1

Command: Calibration unit		Modbus address: **10274**	Length: **2**	Type: **16**	Write
Parameter:	Unit				
Format:	Hex				
Value:	**0x00000100**				

Note:
A change of the calibration unit has no impact on the PMC1 unit.

2.8.2.2 Calibration limits

The min and max limits for product calibration (CP6) are defined in register 13918 for PMC1.

Start register	Number of registers	Reg1 / Reg2 (float)	Reg3 / Reg4 (float)	Modbus function code	Read access	Write access
13918	4	Min value for CP6 (in the physical unit as defined in register 10274)	Max value for CP6 (in the physical unit as defined in register 10274)	3, 4	U/A/S	none

2.8.2.3 Calibration stability criteria

If a product calibration is to be initialized, the environmental conditions are checked first. The measurement for temperature and glucose must be stable for the last 10 minutes. The stability criteria can be found in register 10278 and apply to PMC1.

Start register	Number of registers	Reg1 / Reg2 (float)	Reg3 / Reg4 (float)	Modbus function code	Read access	Write access
10278	4	Max. Drift SMC8 – SMC11 [1/h]	Max. Drift SMC1 Temperature [K/h]	3, 4, 16	U/A/S	S

2.8.2.4 Actual stability

There is the possibility to read out the actual stability of the sensor. Therefore, it's necessary to carry out read access first. To do this, the corresponding bit mask of the SMC to be read must first be written in register 10282 with length 2.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
10282	2	Max. Drift SMCx bitmask see Table 27	16	none	S

Table 26 Bit masks of the SMCs from which the stability can be read

Code (Hex)	Description	Definition
0x000040	SMC1	T act
0x002000	SMC8	S1
0x004000	SMC9	S2
0x008000	SMC10	S3
0x010000	SMC11	S4

After the read access has been executed, the stability can be read with the register 10282 with the length 8.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (uint)	Reg5 / Reg6 (float)	Reg7 / Reg8 (uint)	Modbus function code	Read access	Write access
10282	8	SMCx bitmask see Table 27	Actual drift flag SMCx 0: not stable 1: stable	Actual drift SCMx	reserved	3, 4	U/A/S	none

2.8.2.5 Calibration Parameters and Coefficients

In register 10276, available calibration parameters and calibration coefficients are defined. Bitmask is valid for calibration point CP6.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
10276	2	Available calibration parameters and calibration coefficients for PMC1, see Table 27	3, 4	U/A/S	none

The available calibration parameters and calibration coefficients has a bit mask of 0x07C003FF:

- CPA1 (Calibration parameter index 1) – CPA10 (Calibration parameter index 10)
- CCO1 (Calibration coefficient index 1) – CCO5 (Calibration coefficient index 5)

Table 27 Bitwise definition of available calibration parameters and calibration coefficients.

Code (Hex)	Definition
0x00000001	CPA1 (Calibration parameter index 1)
0x00000002	CPA2 (Calibration parameter index 2)
0x00000004	CPA3 (Calibration parameter index 3)
0x00000008	CPA4 (Calibration parameter index 4)
0x00000010	CPA5 (Calibration parameter index 5)
0x00000020	CPA6 (Calibration parameter index 6)
0x00000040	CPA7 (Calibration parameter index 7)
0x00000080	CPA8 (Calibration parameter index 8)
0x00000100	CPA9 (Calibration parameter index 9)
0x00000200	CPA10 (Calibration parameter index 10)
0x00400000	CCO1 (Calibration coefficient index 1)
0x00800000	CCO2 (Calibration coefficient index 2)
0x01000000	CCO3 (Calibration coefficient index 3)
0x02000000	CCO4 (Calibration coefficient index 4)
0x04000000	CCO5 (Calibration coefficient index 5)

2.8.2.5.1 CPA1 - Measured value

At the time of a valid CP6 calibration, the current measured value of PMC1 value is stored within CPA1.

Start register	Number of registers	Reg1 to Reg8 (16 ASCII characters)	Calibration point	Modbus function code	Read access	Write access
13928	8	Description of CPA1 (PMC1)	CP6 (PMC1)	3, 4	U/A/S	none

In register 13928 (PMC1) for CP6, a plain text ASCII description of CPA1 is given. CPA1 for GlucoSense RS485 is called "Measured value".

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (bitwise defined)	Reg5 / Reg6 (float)	Calibration point	Modbus function code	Read access	Write access
13936	6	Not relevant	Unit (PMC1)	Value (PMC1)	CP6 (PMC1)	3, 4	U/A/S	none

From the last valid CP6 calibration, registers 13936 for CP6 can be used to read the PMC1 values at that time with the fixed unit g/l (0x00000100).

Note:
The first parameter is not relevant and is always zero.

2.8.2.5.2 CPA2 - Assigned value

At the time of a valid CP6 calibration, the assigned calibration value (glucose- concentration) is stored within CPA2.

Start register	Number of registers	Reg1 to Reg8 (16 ASCII characters)	Calibration point	Modbus function code	Read access	Write access
13960	8	Description of CPA2 (PMC1)	CP6	3, 4	U/A/S	none

In register 13960 for CP6, a plain text ASCII description of CPA2 is given. CPA2 for GlucoSense RS485 is called "Assigned value".

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (bitwise defined)	Reg5 / Reg6 (float)	Calibration point	Modbus function code	Read access	Write access
13968	6	Not relevant	Unit (PMC1)	Value (PMC1)	CP6	3, 4	U/A/S	none

From the last valid CP6 calibration, register 13968 can be used to read the assigned values with the calibration unit, set at that time.
In register 10272, the available physical units for calibration are defined, see chapter 2.8.2.1.

Note:
The first parameter is not relevant and is always zero.

2.8.2.5.3 CPA3 - Temperature

At the time of a valid CP6 calibration, the current temperature value is stored within CPA3.

Start register	Number of registers	Reg1 to Reg8 (16 ASCII characters)	Calibration point	Modbus function code	Read access	Write access
13992	8	Description of CPA3 (PMC1)	CP6	3, 4	U/A/S	none

In register 13992, a plain text ASCII description of CPA3 is given. CPA3 for GlucoSense RS485 is called "Temperature".

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (bitwise defined)	Reg5 / Reg6 (float)	Calibration point	Modbus function code	Read access	Write access
14000	6	Not relevant	Unit (PMC1)	Value (PMC1)	CP6	3, 4	U/A/S	none

From the last valid CP6 calibration, register 14000 can be used to read the temperature values with the PMC6 unit, set at that time.
In register 2408, the available physical units for PMC6 are defined, see chapter 2.4.8.4.

Note:
The first parameter is not relevant and is always zero.

2.8.2.5.4 CPA4 - Number of calib.

At the time of a valid CP6 calibration, the quantity performed of calibration value is stored within CPA4.

Start register	Number of registers	Reg1 to Reg8 (16 ASCII characters)	Calibration point	Modbus function code	Read access	Write access
14024	8	Description of CPA4 (PMC1)	CP6	3, 4	U/A/S	none

In register 14024, a plain text ASCII description of CPA4 is given. CPA4 for GlucoSense RS485 is called "Number of calib.".

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (bitwise defined)	Reg5 / Reg6 (uint)	Calibration point	Modbus function code	Read access	Write access
14032	6	Not relevant	Unit (PMC1)	Value (PMC1)	CP6	3, 4	U/A/S	none

From the last valid CP6 calibration, registers 14032 can be used to read the Amount of product-calibration values at that time with no unit (0x00000001).

Note:
The first parameter is not relevant and is always zero.

Note:
The calibration counter is only incremented after a successful assign step (second step of CP6).

2.8.2.5.5 CPA5 - Operating hours

At the time of a valid CP6 calibration, the sensor operating hours value is stored within CPA5.

Start register	Number of registers	Reg1 to Reg8 (16 ASCII characters)	Calibration point	Modbus function code	Read access	Write access
14056	8	Description of CPA5 (PMC1)	CP6	3, 4	U/A/S	none

In registers 14056, a plain text ASCII description of CPA5 is given. CPA5 for GlucoSense RS485 is called "Operating hours".

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (bitwise defined)	Reg5 / Reg6 (uint)	Calibration point	Modbus function code	Read access	Write access
14064	6	Not relevant	Unit (PMC1)	Value (PMC1)	CP6	3, 4	U/A/S	none

From the last valid CP6 calibration, register 14064 can be used to read the sensor operating hours values at that time with no unit (0x00000001).

Note:
The first parameter is not relevant and is always zero.

Note:
The "operating hour" is the moment of the "initial measurement" (first step of CP6).

2.8.2.5.6 CPA6 - Time stamp

At the time of a valid CP6 calibration, the sensor Time stamp (the current system time, see chapter 2.9.2) value is stored within CPA6.

Start register	Number of registers	Reg1 to Reg8 (16 ASCII characters)	Calibration point	Modbus function code	Read access	Write access
14088	8	Description of CPA6 (PMC1)	CP6	3, 4	U/A/S	none

In register 14088, a plain text ASCII description of CPA6 is given. CPA6 for GlucoSense RS485 is called "Time stamp".

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (bitwise defined)	Reg5 / Reg6 (uint)	Calibration point	Modbus function code	Read access	Write access
14096	6	Not relevant	Unit (PMC1)	Value (PMC1)	CP6	3, 4	U/A/S	none

From the last valid CP6 calibration, register 14096 can be used to read the sensor Time stamp values at that time with no unit (0x00000001).

Note:
The first parameter is not relevant and is always zero.

Note:
The system time is set during the action "initial measurement". The system time is explained in chapter 2.9.2.

According to Table 28 the Time Stamp value has no unit (0x00000001), the value is 1614089002 (no unit), means the initial measurement of the product calibration has been performed on February 23th 2021 at 15:03.

Table 28 Example to read the unit and value of CPA6 (Time Stamp) of CP6 of PMC1

Command: Time Stamp CP6	Modbus address: 14096	Length: 6	Type: **3**	Read
Parameter:	Cali. Par.6 - Not relevant	Cali. Par.6 - Unit	Cali. Par.6 - Value	
Format:	Hex	Hex	Decimal	
Value:	**0x00000000**	**0x00000001**	**1614089002**	

2.8.2.5.7 CCO1…CCO5 – Calibration Coefficients

At the time of a valid CP6 calibration the calibration coefficient CCO5 (ΔC: shown in Figure 18) value is stored.
The coefficients CCO1 to CCO4 determined during the factory calibration remain unchanged.

Start register	Number of registers	Reg1 to Reg8 (16 ASCII characters)	Modbus function code	Read access	Write access
14632	8	Description of CCO1 (Factory Calibration, PMC1)	3, 4	U/A/S	none
14664	8	Description of CCO2 (Factory Calibration, PMC1)	3, 4	U/A/S	none
14696	8	Description of CCO3 (Factory Calibration, PMC1)	3, 4	U/A/S	none
14696	8	Description of CCO4 (Factory Calibration, PMC1)	3, 4	U/A/S	none
14696	8	Description of CCO5 (Product Calibration PMC1)	3, 4	U/A/S	none

In registers 14632 to 14696 a plain text ASCII description of CCO1 to CCO5 is given. CCO1 to CCO5 for GlucoSense RS485 is called "g1", "g2", "g3", "g4" and "g5" for PMC1.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (bitwise defined)	Reg5 / Reg6 (float)	Calibration Coefficient Index	Modbus function code	Read access	Write access
14640	6	Not relevant	Unit (PMC1)	Value (PMC1)	CCO1	3, 4	U/A/S	none
14672	6	Not relevant	Unit (PMC1)	Value (PMC1)	CCO2	3, 4	U/A/S	none
14704	6	Not relevant	Unit (PMC1)	Value (PMC1)	CCO3	3, 4	U/A/S	none
14736	6	Not relevant	Unit (PMC1)	Value (PMC1)	CCO4	3, 4	U/A/S	none
14768	6	Not relevant	Unit (PMC1)	Value (PMC1)	CCO5	3, 4	U/A/S	none

From the last valid CP6 calibration, register 14768 can be used to read the product-calibration coefficients g5 (Offset value shown in Figure 18) value with the fixed unit g/l (0x00000100). The other coefficients are not changed by the product calibration

Note:
The first parameter is not relevant and is always zero.

2.8.3 Calibration Procedure

2.8.3.1 Calibration at CP6 (Product Calibration)

The product calibration is a process to adjust the measurement of a factory calibrated GlucoSense RS485 sensor to specific process conditions.

GlucoSense RS485 has a unique product calibration routine. When initiating the product calibration, the data set of the GlucoSense RS485 is automatically traced back within the last 600 seconds and a decision is made immediately if the product calibration is successful or not. The operator therefore gets an immediate result. The criteria for a successful calibration are:

- the stability of S1 to S4 and temperature over the last 600 seconds
- the input value content is in the limits defined for CP6

 Attention:

If the Measurement Hold Mode is in Hold Mode or Off, see chapter 2.5, the initial measurement command has no effect.

 Attention:

It is important that GlucoSense RS485 is in a defined calibration media at least 600 seconds BEFORE the calibration is started.

The product calibration (CP6) is initiated for PMC1 by writing to register 13914.

The calibration value (glucose value) must be given in the physical unit defined within register 10274.

Product calibration is a two-stage process and is performed independently for PMC1.

1. An initial measurement is performed while the operator takes a sample of the process solution. At that time point the GlucoSense RS485 sensor stores its raw measurement value, temperature and operating hour in the memory.

   While the operator takes the sample to the analytics lab for reference analysis the GlucoSense RS485 sensor is still running on its factory calibration while the initial measurement data for the ongoing product calibration is kept in the GlucoSense RS485 sensors memory.

2. When the result of the reference analysis is available, this value is assigned to a second time point to the former initial measurement data, which is stored in the GlucoSense RS485 sensor.

   The sensor is now, after valid assignment, running on a calibration function which is compensated for the correct process conditions. The product calibration (CP6) is now active.

Performing a cancel command for the product calibration (CP6) brings the sensor back to its stored factory calibration.
If the operator needs to adjust an active product calibration (old CP6) by a new product calibration (new CP6) the above process applies in the same way. After initial measurement the GlucoSense RS485 sensor is running on the first product calibration (old CP6) until a valid assignment has been done (new CP6).

What happens to the GlucoSense RS485 calibration function during product calibration (CP6)?
During a product calibration for the GlucoSense RS485 sensor, the factory calibration is shifted with an offset. This offset is calculated after initialization and assignment of the product calibration.

Example:
The operator starts with a factory calibration:

Calibration coeffitient	Value in g/l	PMCx
g1	0.064	PMC1
g2	0.0012	PMC1
g3	-0.0355	PMC1
g4	0.026	PMC1
g5	0	PMC1

The sensor internally calculates the calibration function, using the calibration coefficients g1 to g5 for PMC1. The resulting calibration function, compensated to the standard temperature 25°C, is shown as a straight line. The calibration function is shown in Figure 18. How to read these values is described in chapter 2.8.2.5.7.
After a factory calibration has been carried out, the coefficient g5 for PMC1 is 0. If a process is now started, a product calibration must always be carried out for PMC1, in other words adjusting the factory calibration function to the process conditions:

Calibration point	Glucose Concentration value [g/l]	Temperature in °C	Measured Measured Glucose Concentration [g/l]
CP6	2	28.79	5

The sensor now calculates an offset (3 g/l) and uses it to shift the operating point of the existing factory calibration. The new calibration function, compensated to the standard temperature 25°C, is shown as a dotted line in Figure 18.

Another special feature of this calibration point is to switch off and back on again a product calibration. These functions are called "restore factory calibration" and "restore product calibration".

The sensor's internal criteria for a successful product calibration are:

- the sensor is currently in an environment corresponding to the GlucoSense RS485 measurement range.
- the glucose- concentration content is within the calibration limits defined for CP6 (see above)
- the glucose- measurement is stable for the last 10 minutes (see register 10278).
- the temperature is within the calibration limits defined for CP6 (see above)
- the temperature is stable for the last 10 minutes (see register 10278).

The different functionalities of product calibration (CP6) are accessible through the following sensor commands:

- Initial measurement
- Assignment
- Cancel
- Restore standard calibration
- Restore product calibration

All commands are executed by writing a command to the register 13914 for PMC1. The command codes are defined according to Table 29.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (float)	Modbus function code	Read access	Write access
13914	4	Code as defined in Table 29 (PMC1)	Calibration value (PMC1)	3, 4, 16	U/A/S	S

Table 29 Definition of the commands related to the product calibration

Code (Hex)	Definition of commands
0x0010	Perform initial measurement
0x0020	Assign product calibration
0x0040	Cancel an active product calibration
0x0080	Restore a standard calibration from an active product calibration
0x0100	Restore a product calibration from an active standard calibration

2.8.3.1.1 Product calibration: Initial measurement

Upon process sample collection for laboratory analysis the command for initial measurement is sent to the sensor.
This is achieved by writing the command 0x0010 to register 13914 which performs the initial measurement and stores the corresponding measurement values in the sensor.

Table 30 Example to start the product calibration procedure for PMC1

Command: CP6: Initial measurement	Modbus address: **13914**	Length: **4**	Type: **16**	Write
Parameter:	Command	Value		
Format:	Hex	Float		
Value:	**0x00000010**	**0.0**		

Note:

The second parameter is for the product calibration initialisation not relevant and is ignored by the sensor. As default value during a product calibration initialisation, it is recommended to set the value to zero.

As shown in Table 31, after successful initial measurement the corresponding calibration status is "CP6: Initial measurement" (0x20000000, see also Table 29).

The sensor continues the measurement with the previous standard calibration.

Table 31 Example to read the current product calibration status of PMC1

Command: CP6: Initial measurement	Modbus address: **13912**	Length: **2**	Type: **3**	Read
Parameter:	Command			
Format:	Hex			
Value:	**0x20000000**			



Attention:

If the Measurement Hold Mode is in Hold Mode or Off, see chapter 2.5, the initial measurement command has no effect.

2.8.3.1.2 Product calibration: Assignment

After successful initial measurement a correct value must be assigned to the initially stored measurement data. As shown in Table 32, this is achieved by writing the glucose- concentration value in the g/l during initial measurement (here 4.5 g/l) to 13914.

Table 32 Example to assign a calibration value to the above performed initial measurement for PMC1

Command: CP6: Assignment	Modbus address: **13914**	Length: **4**	Type: **16**	Write
Parameter:	Command	Value		
Format:	Hex	Float		
Value:	**0x00000020**	**4.5**		

From now on the sensor is measuring using the here performed product calibration.

According to Table 33 the calibration status of the sensor is 0x50000000 meaning that a correct value has been assigned and that the product calibration is active (see Table 40).

Table 33 Example to read the current product calibration status, after performing a product calibration.

Command: CP6: Initial measurement	Modbus address: **13912**	Length: **2**	Type: **3**	Read
Parameter:	Command			
Format:	Hex			
Value:	**0x50000000**			

2.8.3.1.3 Product calibration: Cancel

To cancel an active product calibration or an active initial measurement the command 0x00000040 is written to register 13914 (see Table 34). The calibration value is not considered in this product calibration cancel phase and the value is not saved within the sensor.

Table 34 Example to cancel an active product calibration or an initial measurement for PMC2

Command: CP6: Cancel	Modbus address: **13914**	Length: **4**	Type: **16**	Write
Parameter:	Command	Value		
Format:	Hex	Float		
Value:	**0x00000040**	**0.0**		

Performing this action, the product calibration or any initial measurements are canceled. The values of the prior product calibration are removed from the sensor's memory. From now on the sensor is measuring using its factory calibration.

As shown in Table 35 after cancelling a product calibration the sensors calibration status will be reading 0x00000000 again (see Table 40).

Table 35 Example to read the current product calibration status of PMC1

Command: CP6: Initial measurement	Modbus address: **13912**	Length: **2**	Type: **3**	Read
Parameter:	Command			
Format:	Hex			
Value:	**0x00000000**			

2.8.3.1.4 Product calibration: Restore standard calibration

If a product calibration is active this product calibration can be temporarily switched off by writing the command "restore standard calibration" (0x00000080) (see Table 40) to register 13914 (see Table 36). The calibration value is not considered in this restore standard calibration phase and the value is not saved within the sensor.
Performing this action, the values of the product calibration remain stored in the sensor's memory.

Table 36 Example to restore a standard calibration from an active product calibration for PMC1

Command: CP6: Restore standard	Modbus address: **13914**	Length: **4**	Type: **16**	Write
Parameter:	Command	Value		
Format:	Hex	Float		
Value:	**0x00000080**	**0.0**		

From now on the sensor is measuring using its factory calibration.
As shown in Table 37 the sensor's calibration status will be reading "CP6 assigned" (0x40000000) (see Table 40) meaning that a valid assignment for a product calibration is available in the sensor's memory.

Table 37 Example to read the current product calibration status after a restoring a standard calibration of PMC1

Command: CP6: Initial measurement	Modbus address: **13912**	Length: **2**	Type: **3**	Read
Parameter:	Command			
Format:	Hex			
Value:	**0x40000000**			

2.8.3.1.5 Product calibration: Restore product calibration

If a valid but inactive product calibration is available in the sensor's memory, the calibration status is reading 0x40000000 ("CP 6 assigned", see Table 37). This stored product calibration can be restored or reactivated by writing command "restore product calibration" 0x00000100 (see Table 40) to register 13914 according to Table 38. The calibration value is not considered in this restore product calibration phase and the value is not saved within the sensor.

Table 38 Example to restore an available product calibration from the factory calibration for PMC1

Command: CP6: Restore product	Modbus address: **13914**	Length: **4**	Type: **16**	Write
Parameter:	Command	Value		
Format:	Hex	Float		
Value:	**0x00000100**	**0.0**		

If this command is performed without available product calibration in the sensors memory the sensor will respond with a Modbus exception since this command is not valid.

From now on the sensor is measuring using its prior CP6 product calibration. As shown in Table 39 the sensors calibration status will be reading 0x50000000 (corresponding to "CP6 assigned" and "CP6 active") again. (see Table 40)

Table 39 Example to read the current product calibration status, after restoring a product calibration for PMC1

Command: CP6: Initial measurement	Modbus address: **13912**	Length: **2**	Type: **3**	Read
Parameter:	Command			
Format:	Hex			
Value:	**0x50000000**			

2.8.4 Reading the Calibration Status

A calibration is not always successful. To analyze what has gone wrong, three different calibration status registers can be read:

- Register 13912 for CP6 of PMC1

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Modbus function code	Read access	Write access
13912	2	Status CP6 (see Table 40)	3, 4	U/A/S	none

Table 40 Definition of the calibration status for registers 13912

Code (Hex)	Definition
0x00000000	calibration successful
0x00000100	current temperature reading is too low (see register 4616)
0x00000200	current temperature reading is too high (see register 4616)
0x00001000	Temperature reading during calibration is not stable (see register 10278)
0x00002000	Glucose reading during calibration is not stable (see register 10278)
0x00004000	out of calibration range (wrong glucose value entered)
0x10000000	active
0x20000000	initial measurement
0x40000000	assigned

2.9 Sensor Status

2.9.1 Temperature Ranges

In registers 4608, 4612, 4616 and 4624 four different temperature ranges are defined:

- Operation is the maximum temperature range to which the sensor can be exposed to during operation and storage. If the current temperature is out of this range, the corresponding bit in the measurement status register is set, see chapter 2.4.9.
- Measurement – is the maximum allowable range where glucose concentration measurement is possible.
- Product Calibration – in this range a product calibration can be performed (CP6).
- User defined measurement – the specialist can adjust the range in which glucose concentration reading is active. The user defined measurement temperature range is a sub range of the measurement temperature range.

Start register	Number of registers	Reg1 / Reg2 (float)	Reg3 / Reg4 (float)	Modbus function code	Read access	Write access
4608	4	Operating temperature min	Operating temperature max	3, 4	U/A/S	none
4612	4	Measurement temperature min	Measurement temperature max)	3, 4	U/A/S	none
4616	4	Product Calibration temperature min	Product Calibration temperature max	3, 4	U/A/S	none
4624	4	User defined measurement temperature min	User defined measurement temperature max	3, 4, 16	U/A/S	S

The unit of the temperatures is according to the selected unit of PMC6 (see 2.4.8.4 Definition of PMC6).

If operating- or the measurement temperature range is exceeded, the measurement status will be set accordingly (see chapter 2.4.9).

For GlucoSense RS485 the ranges are as follows:

- Operating temperature: -20 .. +140°C
- Measurement temperature: 0 .. +80°C
- Calibration temperature: +5 .. +50°C
- User defined measurement temperature: 0 .. +50°C (default values)



Note:

Temperature reading is active at any time, regardless of the current temperature.

2.9.2 Operating Hours, Counters and System Time

Start register	Number of registers	Reg1 / Reg2	Reg3 / Reg4	Reg5 / Reg6	Modbus function code	Read access	Write access
4676	6	Operating hours [h] (float)	Operating hours above max measurement temperature [h] (float)	Operating hours above max operating temperature [h] (float)	3, 4	U/A/S	none
4682	6	Number of Power ups (uint)	Number of Watchdog resets (uint)	Heartbeat (uint)	3, 4	U/A/S	none
4686	1	Heartbeat (uint 16)			3, 4	U/A/S	none
4688	4	Number of SIP cycles (uint)	Number of CIP cycles (uint)		3, 4	U/A/S	none
4692	2	Number of autoclavings (uint)			3, 4, 16	U/A/S	S
8232	2	System Time Counter [s] (uint)			3, 4, 16	U/A/S	S

In register 4676 are stored:

- total operating hours
- operating hours above max measurement temperature (see chapter 2.9.1)
- the operating hours above max operating temperature (see chapter 2.9.1)

In register 4682 are stored:

- number of power ups
- number of watchdog resets
- number of heartbeats. The number gets increased every second by 1 and ranges from 0 to 19.

In register 4686 is stored:

- number of heartbeats.
  The Number is stored as an unsigned integer with the length of 16 bit. The number gets increased every second by 1 and ranges from 0 to 19.

In register 4688 are stored:

- number of sterilizations in place (SIP) (see chapter 2.9.6)
- number of cleanings in place (CIP) (see chapter 2.9.6)

In register 4692 is stored

- number of autoclavings
  This register has no effect on the sensor and is only for the user to store the number of autoclavings.

In register 8232 is stored

- system time counter.
  When the sensor is powered up, the system time is set to 0. A value between 0 and 2^{32} - 1 can be written into this register. From this value, the sensor increments this value every second. We recommend using as base date the so-called UNIX timestamp (hint: www.epochconverter.com) which starts at 1st of January 1970 GMT. Be sure to update this register if needed after every power up of the sensor.

Note:

Accuracy of the system time, if not updated by the operator: The deviation of the system time is less than one minute per week.

2.9.3 Warnings

A "Warning" is a notification message which still allows further functioning of the system. This message alerts the operator of a possible problem that could lead to uncertain results.

2.9.3.1 Currently Active Warnings

The currently active warnings are stored in register 4736. For the definition of warnings, see chapter 2.9.3.2.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (bitwise defined)	Reg5 / Reg6 (bitwise defined)	Reg7 / Reg8 (bitwise defined)	Modbus function code	Read access	Write access
4736	8	Active warnings measurement	Active warnings calibration and membrane	Active warnings interface	Active warnings hardware	3, 4	U/A/S	none

2.9.3.2 Definition of Warnings

Table 41 Definition of warnings "measurement".

Bit #	Code (Hex)	Description
0 (LSB)	0x00000001	PMC1 GLC reading below lower limit
	The glucose reading is too low (SMC8, SMC9, SMC10 or SMC11 < -1000). Check for Air bubbles	
1	0x00000002	PMC1 GLC reading above upper limit
	The glucose reading is too high (SMC8, SMC9, SMC10 or SMC11 > 1000).	
25	0x02000000	PMC6 T reading below lower limit
	The temperature is below the user defined measurement temperature range. If outside this range, the sensor will not perform GLC readings.	
26	0x04000000	PMC6 T reading above upper limit
	The temperature is above the user defined measurement temperature range. If outside this range, the sensor will not perform GLC readings.	
31	0x80000000	Measurement not running
	Causes that trigger this warning: • Measurement Hold Mode is in Hold- or Off mode, see chapter 2.5. • Sensor operating voltage range is not between 10-27 VDC • The temperature measurement is outside the user defined temperature range, see chapter 2.9.1.	

Table 42 Definition of warnings "calibration and membrane".

Bit #	Code (Hex)	Description
0 (LSB)	0x00000001	PMC1 GLC new product calibration recommended
	Perform a correct product calibration for PMC1 to ensure reliable measurement.	
2	0x00000004	PMC1 GLC product calibration not active
	Perform a product calibration for PMC1	

Table 43 Definition of warnings "interface".

Bit #	Code (Hex)	Description
		Not available

Table 44 Definition of warnings "hardware".

Bit #	Code (Hex)	Description
0 (LSB)	0x00000001	Sensor supply voltage too low
	The sensor supply voltage is too low for the sensor to operate correctly. Ensure stable supply voltage within the sensor's specifications.	
1	0x00000002	Sensor supply voltage too high
	The sensor supply voltage is too high for the sensor to operate correctly. Ensure stable supply voltage within the sensor's specifications.	
9	0x00000200	Replace Sensor recommended
	The Sensor Quality Indicator is below 40%. The quality of the sensor is sufficient for reliable measurement, but replacement of the sensor will be needed in near future	

2.9.4 Errors

An "Error" message indicates a serious problem of the sensor which does not allow further proper functioning of the sensor. This problem must be solved.

2.9.4.1 Currently Active Errors

The currently active errors are stored in register 4800.

Start register	Number of registers	Reg1 / Reg2 (bitwise defined)	Reg3 / Reg4 (bitwise defined)	Reg5 / Reg6 (bitwise defined)	Reg7 / Reg8 (bitwise defined)	Modbus function code	Read access	Write access
4800	8	Active errors measure-ment	Active errors calibration and membrane	Active errors interface	Active errors hardware	3, 4	U/A/S	none

2.9.4.2 Definition of Errors

Table 45 Definition of errors "measurement".

Bit #	Code (Hex)	Description
0 (LSB)	0x00000001	PMC1 Glucose concentration reading failure
	If there is no communication with the frontend device or something went wrong when analyzing the measurement (INF or NAN). In the latter case, please call our Technical Support.	
25	0x02000000	PMC6 T sensor defective
	The internal temperature sensor is defective.	

Table 46 Definition of errors "calibration and membrane".

Bit #	Code (Hex)	Description
		Not available

Table 47 Definition of errors "interface".

Bit #	Code (Hex)	Description
		Not available

Table 48 Definition of errors "hardware".

Bit #	Code (Hex)	Description
2	0x00000004	Temperature reading far below min
	The measured temperature is below the operation temperature (Reg. 4608)	
3	0x00000008	Temperature reading far above max
	The measured temperature is above the operation temperature (Reg. 4608)	
9	0x00000200	Sensor Defective
	Sensor is defective or Sensor Quality Indicator is below 10%. The quality of the sensor is not sufficient for reliable measurement. Sensor needs to be replaced.	
22	0x40000000	EEPROM comm. (I2C) error Userend
	Internal I2C communication error Reset the sensor and try again. Please call our Technical Support.	
24	0x01000000	Internal communication (I2C) failure Userend
	Internal I2C communication error Reset the sensor and try again. Please call our Technical Support.	
25	0x02000000	Internal communication failure (frontend)
	No communication between Frontend and Userend. Reset the sensor and try again. Please call our Technical Support.	
26	0x04000000	Stackoverflow
	Internal memory failure Reset the sensor and try again. Please call our Technical Support.	

Note:

If an internal error occurs, reset the sensor and try again or get in contact with our Technical Support.

2.9.5 PMCs output in case of sensor error / warning

2.9.5.1 Measurement values - Exceeds Temperature range, and GLC measurement errors

In case that the current measurement temperature exceeds the user defined temperature range, following warning bits are set:

Category	Code (Hex)	Description
Measurement Warning	0x02000000	T reading below lower limit
	0x04000000	T reading above upper limit
	0x80000000	Measurement not running

In case that the current measurement temperature exceeds the operating temperature range or the glucose concentration measurements is erroneous, following error bits must be set:

Category	Code (Hex)	Description
Hardware Error	0x00000004	Temperature reading far below min
	0x00000008	Temperature reading far above max
Measurement Error	0x00000001	GLC reading failure

The measurement output values for these cases are defined in Table 49.

Table 49 Measurement values (PMCs) - Exceeds Temperature range, and GLC measurement errors

Bit #	Code (Hex)	Description	Definition	Measurement output values (float)
0 (LSB)	0x00000001	PMC1	GLC	-999.0
5	0x00000020	PMC6	T (temperature measurement value)	Current temp. value

2.9.5.2 Measurement values – no measurement within the frontend is running

In case that the current supply voltage is not within the defined supply voltage range, following warning bits are set:

Category	Code (Hex)	Description
Hardware Warning	0x00000001	Sensor supply voltage too low
	0x00000002	Sensor supply voltage too high
Measurement Warning	0x80000000	Measurement not running

The measurement output values for these cases, are defined in Table 50.

Table 50 Measurement values (PMCs) in case no communication within the frontend is running

Bit #	Code (Hex)	Description	Definition	Measurement output values (float)
0 (LSB)	0x00000001	PMC1	GLC	-999.0
5	0x00000020	PMC6	T (temperature measurement value)	-999.0

2.9.5.3 Measurement values – Hardware related errors

In case that the current measurement temperature exceeds the operating temperature range or the glucose concentration measurements is erroneous, following error bits must be set:

Category	Code (Hex)	Description
Hardware Error	0x00000200	Sensor Defective
	0x00400000	EEPROM comm. (I2C) error
	0x01000000	Internal communication (I2C) failure
	0x02000000	Internal communication failure (frontend)
	0x04000000	Stackoverflow
Measurement Error	0x02000000	T sensor defective

The measurement output values for these cases, are defined in Table 51.

Table 51 Measurement values (PMCs) in case Hardware errors

Bit #	Code (Hex)	Description	Definition	Measurement output values (float)
0 (LSB)	0x00000001	PMC1	GLC	-999.0
5	0x00000020	PMC6	T (temperature measurement value)	-999.0

2.9.6 Reading Definition of SIP and CIP

GlucoSense RS485 is counting special cleaning events such as sterilizations or cleaning cycles by means of tracking typical temperature profiles (see chapter 2.9.2).

Register 4988 defines the temperature profile for SIP (sterilization in place) and register 4996 the temperature profile for CIP (cleaning in place). For the explanation the following values are given:

CIP temperature min: 80 °C CIP temperature max: 100 °C CIP time min: 30 minutes
SIP temperature min: 120 °C SIP temperature max: 140 °C SIP time min: 30 minutes

CIP / SIP time maximum values: 180 min

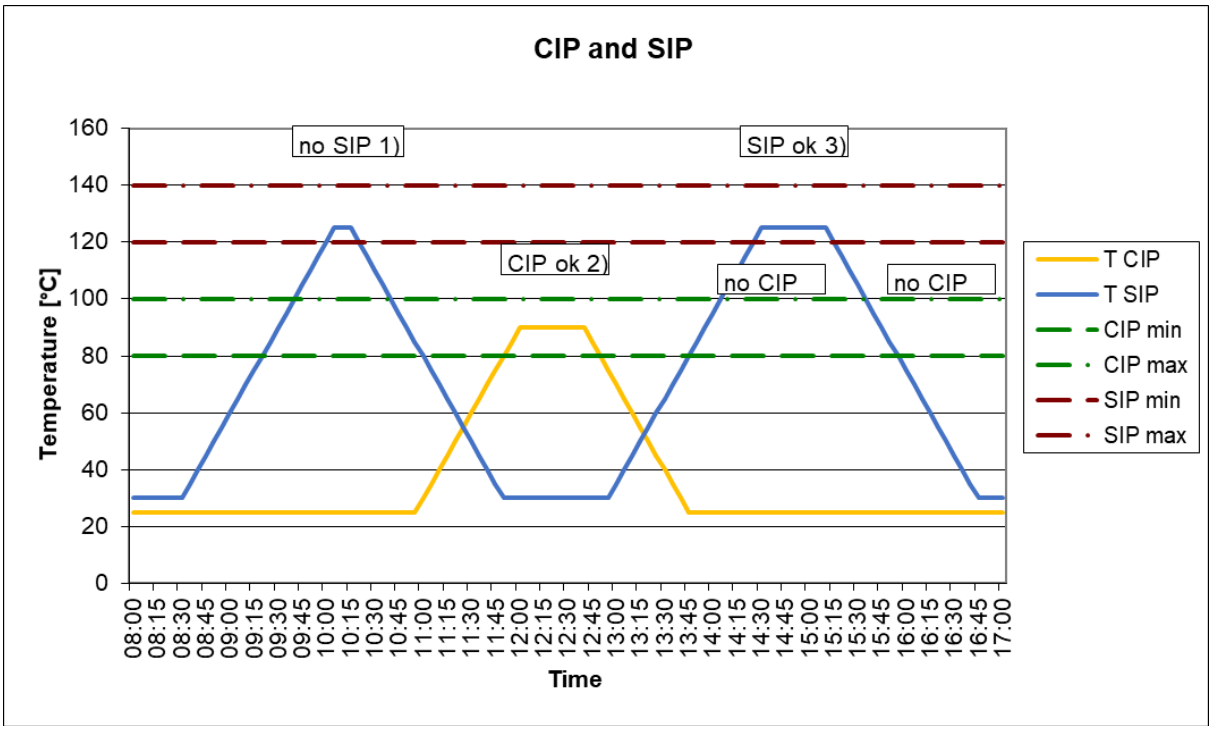


Figure 19: Definition of CIP and SIP cycles.
1) no SIP-cycle counted, because time too short less than 30 minutes.
2) CIP-cycle counted, because time greater than 30 minutes and in CIP temperature range.
3) SIP-cycle counted, because time greater than 30 minutes and in SIP temperature range.
4) no CIP-cycle counted, because of reaching the SIP-min limit.

Start register	Number of registers	Reg1 / Reg2 (float)	Reg3 / Reg4 (float)	Reg5 / Reg6 (float)	Reg7 / Reg8	Modbus function code	Read access	Write access
4988	8	SIP Tempera-ture min [°C]	SIP Tempera-ture max [°C]	SIP Process time min [min]	Empty	3, 4	U/A/S	S
4996	8	CIP Tempera-ture min [°C]	CIP Tempera--ture max [°C]	CIP Process time min [min]	Empty	3, 4	U/A/S	S

The unit of the temperatures is according to the selected unit of PMC6 (see 2.4.8.4 Definition of PMC6).

2.9.7 Quality Indicator

2.9.7.1 Reading the Sensor Cap Quality

In register 5472 with a command length of 2, the sensor quality (0-100%) is given in 10% percent steps. Hamilton recommends the replacement of the sensor at a value less than 40 % (see Table 44).

Start register	Number of registers	Reg1 / Reg2 (float)	Modbus function code	Read access	Write access
5472	2	Sensor Cap Quality [%]	3, 4	U/A/S	none

Table 52 Example to read the Sensor Quality Indicator with a command length of 2

Command: Sensor cap quality		Modbus address: **5472**	Length: **2**	Type: **3**	Read
Parameter:	Sensor quality [%]				
Format:	Float				
Value:	**100**				

2.9.7.2 Reading the Measurement Quality Indicator

In register 5472 with a command length of 6, the sensor quality (0-100%) in 10% percent steps and the measurement quality is given.

Start register	Number of registers	Reg1 / Reg2 (float)	Reg3 / Reg4 (float)	Reg5 / Reg6 (float)	Modbus function code	Read access	Write access
5472	6	Sensor Quality [%]	Not used	Measurement quality, see Table 53	3, 4	U/A/S	none

Table 53 Definition of measurement quality values

Value	Description
100	Good: => 40 % sensor quality (no actions)
50	Bad: < 40 % sensor quality (maintenance required)
0	Defect: < 10 % sensor quality (senser defective)

Table 54 Example to read the sensor- and measurement quality with a command length of 6

Command: Measurement quality		Modbus address: **5472**	Length: **6**	Type: **3**	Read
Parameter:	Sensor quality [%]	Not used	Measurement quality		
Format:	Float	Float	Float		
Value:	**100**	**-999**	**100**		

2.10 Sensor Identification and Information

2.10.1 General Information

General information about the sensor is available as shown in the Table below.

Start register	Number of registers	Reg1 to Reg8 16 ASCII characters	Example of content	Modbus function code	Read access	Write access
1024	8	Userend FW Date	2025-11-04	3, 4	U/A/S	none
1032	8	Userend FW	GLOUM101	3, 4	U/A/S	none
1040	8	Userend BL Date	2022-05-16	3, 4	U/A/S	none
1048	8	Userend BL	BL5UX101	3, 4	U/A/S	none
1056	8	Userend Ref	10151002/00	3, 4	U/A/S	none
1064	8	Userend SN	1234	3, 4	U/A/S	none
1072	8	Userend (space holder)	not available	3, 4	U/A/S	none
1080	8	Userend (space holder)	not available	3, 4	U/A/S	none
1088	8	Frontend FW Date	2025-10-24	3, 4	U/A/S	none
1096	8	Frontend FW	GLOFJ001	3, 4	U/A/S	none
1104	8	Frontend BL Date	not available	3, 4	U/A/S	none
1112	8	Frontend BL	not available	3, 4	U/A/S	none
1120	8	Frontend Ref	not available	3, 4	U/A/S	none
1128	8	Frontend SN	not available	3, 4	U/A/S	none
1136	8	Frontend (space holder)	not available	3, 4	U/A/S	none
1144	8	Frontend (space holder)	not available	3, 4	U/A/S	none

2.10.2 Sensor Identification

Definition of registers containing sensor identification:

Start register	Number of registers	Reg1 to Reg8 16 ASCII characters	Example of content	Modbus function code	Read access	Write access
1280	8	Sensor Ref	10184106/00	3, 4	U/A/S	none
1288	8	Sensor name	GlucoSense RS485	3, 4	U/A/S	none
1296	8	Sensor Lot	1354271	3, 4	U/A/S	none
1304	8	Sensor Lot date	2025-08-09	3, 4	U/A/S	none
1312	8	Sensor SN	2076	3, 4	U/A/S	none
1320	8	Manufacturer part 1	HAMILTON Bonaduz	3, 4	U/A/S	none
1328	8	Manufacturer part 2	AG Switzerland	3, 4	U/A/S	none
1336	8	Sensor type	ARC GLO Sensor	3, 4	U/A/S	none
1344	8	Power supply	10 - 27V 1.5W	3, 4	U/A/S	none
1352	8	Pressure range	10 - 12000mbar	3, 4	U/A/S	none
1360	8	Sensor ID	10184106-1-11	3, 4	U/A/S	none
1368	8	a-length	120	3, 4	U/A/S	none
1376	8	(space holder)	not available	3, 4	U/A/S	none
1384	8	Electrical connection	VP 8.0	3, 4	U/A/S	none
1392	8	Process connection	PG 13.5	3, 4	U/A/S	none
1400	8	Sensing material	ATR Crystal	3, 4	U/A/S	none

2.10.3 Free User Memory Space

These registers can be used to store any customer specific information in the sensor. There are different registers which can be read by everybody, but only specific operators can write them.

Start register	Number of registers	Reg1 to Reg8 16 ASCII characters	Example of content	Modbus function code	Read access	Write access
1536	8	Userspace	*FREE_USERSPACE*	3, 4, 16	U/A/S	U/A/S
1544	8	Userspace	*FREE_USERSPACE*	3, 4, 16	U/A/S	U/A/S
1552	8	Userspace	*FREE_USERSPACE*	3, 4, 16	U/A/S	U/A/S
1560	8	Userspace	*FREE_USERSPACE*	3, 4, 16	U/A/S	U/A/S
1568	8	Userspace	*FREE_USERSPACE*	3, 4, 16	U/A/S	A/S
1576	8	Userspace	*FREE_USERSPACE*	3, 4, 16	U/A/S	A/S
1584	8	Userspace	*FREE_USERSPACE*	3, 4, 16	U/A/S	A/S
1592	8	Userspace	*FREE_USERSPACE*	3, 4, 16	U/A/S	A/S
1600	8	Measuring Point	10118255-2076	3, 4, 16	U/A/S	S
1608	8	Userspace	*FREE_USERSPACE*	3, 4, 16	U/A/S	S
1616	8	Userspace	*FREE_USERSPACE*	3, 4, 16	U/A/S	S
1624	8	Userspace	*FREE_USERSPACE*	3, 4, 16	U/A/S	S
1632	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1640	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1648	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1656	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1664	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1672	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1680	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1688	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1696	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1704	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1712	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1720	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1728	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1736	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1744	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none
1752	8	Userspace	*FREE_USERSPACE*	3, 4	U/A/S	none

Important registers are 1600, 1608 and 1616 which are the measuring point, the membrane part number and the membrane lot number. This information is used by ArcAir to identify individual sensors and to trace the membrane.

2.10.4 Membrane Information

Definition of registers containing Membrane information:

Start register	Number of registers	Reg1 to Reg8 16 ASCII characters	Example of content	Modbus function code	Read access	Write access
1792	8	Membrane PN	10190449	3, 4, 16	U/A/S	A/S
1800	8	Membrane Name	Glu. Membrane	3, 4, 16	U/A/S	A/S
1808	8	Membrane Lot	9999999999	3, 4, 16	U/A/S	A/S
1816	8	Membrane Lot Date	1999-01-01	3, 4, 16	U/A/S	A/S
1824	8	Membrane (space holder)	not available	3, 4, 16	U/A/S	A/S



Attention:
The Free User Memory Space is located in a memory which allows in total max 1'000'000 write operations.

2.11 System Commands

2.11.1 Recall Sensor's Factory Settings

Using register 8192 you can recall the sensor manufacturer values (interfaces, calibration data and passwords), except:

- SIP counter,
- CIP counter,
- Power up counter,
- Number of Watchdog Resets,
- Number of autoclaving's,
- Operating hours,
- Operating hours above max measurement temperature,
- Operating hours above max operating temperature and
- all calibration counter values
- measuring point

remains unchanged.

By sending the recall value "911", all configuration values will be set to default.

Start register	Number of registers	Reg1 / Reg2	Modbus function code	Read access	Write access
8192	2	Recall by value 911	16	none	S

3 Appendix

3.1 List of tables

Table 1 Modbus definition for data transmission 9
Table 2 RS485 definitions for Arc Sensors 9
Table 3 Definition Floating point Single Precision (4 bytes resp. 2 Modbus registers) 12
Table 4 Implemented Error-Codes 14
Table 5 Definition of operator level and default passwords 15
Table 6 RS485 factory settings 16
Table 7 Code for the baud rates. 17
Table 8 Definition of the analog interface modes 19
Table 9 Code for selection of the primary measurement channel 20
Table 10 Code for the 4-20 mA interface in case of errors and warnings. 25
Table 11 Example: Read the settings for AO1 in case of warnings and errors 26
Table 12 Definition of PMC1 to 6 and SMC1 to 16 27
Table 13 Example to read Reg. 2048 27
Table 14 Definition of physical units 28
Table 15 Example to read the physical unit in plain text ASCII in register 1952 28
Table 16 Example to set the physical unit of PMC1 to g/l (0x00000100) 29
Table 17 Example to read register 2410 32
Table 18 Definition of measurement status for Primary Measurement Channels 34
Table 19 Code for selection of the measurement hold mode 40
Table 20 Bitwise definition of all parameters PA1 to PA16 41
Table 21 Example to read the available Parameters with operator level S 41
Table 22: Bitwise definition of the verification command 44
Table 23 Bitwise definition of CP6 45
Table 24 Example to read the available CPs of PMC1 45
Table 25 Example to set the physical calibration unit to g/l (0x00000100) for PMC1 46
Table 26 Bit masks of the SMCs from which the stability can be read 47
Table 27 Bitwise definition of available calibration parameters and calibration coefficients. 48
Table 28 Example to read the unit and value of CPA6 (Time Stamp) of CP6 of PMC1 52
Table 29 Definition of the commands related to the product calibration 55
Table 30 Example to start the product calibration procedure for PMC1 55
Table 31 Example to read the current product calibration status of PMC1 56
Table 32 Example to assign a calibration value to the above performed initial measurement for PMC1 56
Table 33 Example to read the current product calibration status, after performing a product calibration. 56
Table 34 Example to cancel an active product calibration or an initial measurement for PMC2 56
Table 35 Example to read the current product calibration status of PMC1 57
Table 36 Example to restore a standard calibration from an active product calibration for PMC1 57
Table 37 Example to read the current product calibration status after a restoring a standard calibration of PMC1 57
Table 38 Example to restore an available product calibration from the factory calibration for PMC1 57
Table 39 Example to read the current product calibration status, after restoring a product calibration for PMC1 58
Table 40 Definition of the calibration status for registers 13912 59
Table 41 Definition of warnings "measurement". 63
Table 42 Definition of warnings "calibration and membrane". 64
Table 43 Definition of warnings "interface". 64
Table 44 Definition of warnings "hardware". 64
Table 45 Definition of errors "measurement". 65
Table 46 Definition of errors "calibration and membrane". 65
Table 47 Definition of errors "interface". 66
Table 48 Definition of errors "hardware". 66
Table 49 Measurement values (PMCs) - Exceeds Temperature range, and GLC measurement errors 67
Table 50 Measurement values (PMCs) in case no communication within the frontend is running 67
Table 51 Measurement values (PMCs) in case Hardware errors 68

Table 52 Example to read the Sensor Quality Indicator with a command length of 2 70
Table 53 Definition of measurement quality values....70
Table 54 Example to read the sensor- and measurement quality with a command length of 6 70

3.2 List of figures

Figure 1 Modbus Protocol Data Unit. 6
Figure 2 Modbus frame over Serial Line. 6
Figure 3 Bit sequence in RTU mode. 7
Figure 4 RTU Message Frame. 7
Figure 5 Valid frames with silent intervals. 7
Figure 6 RTU Message Frame. 7
Figure 7 Data transmission of a frame. 8
Figure 8 Definition of Holding Registers. 10
Figure 9 Example of reading holding registers 108 – 110. The contents of register 108 are read as the two-byte values 0x022B. The contents of registers 109 – 110 are 0x00 00 and 0x0064. 10
Figure 10 Definition of Input Registers. 10
Figure 11 Example of reading input register 9. The contents of input register 9 are read as the two-byte value 0x000A. 11
Figure 12 Definition of Write Multiple Registers. 11
Figure 13 Example of writing the value 0x000A and 0x0102 to two registers starting at address 2. 11
Figure 14 Example of linear 4-20mA output characteristics 23
Figure 15 Example of bilinear 4-20mA output characteristics 23
Figure 16: Description of the measurement hold mode functionality 40
Figure 17 Comparison of the response of GlucoSense RS485 to a change from a glucose mixture to water 42
Figure 18 Effect of the product calibration CP6 on an existing factory calibration function defined by Hamilton. 45
Figure 19: Definition of CIP and SIP cycles. 1) no SIP-cycle counted, because time too short less than 30 minutes. 2) CIP-cycle counted, because time greater than 30 minutes and in CIP temperature range. 3) SIP-cycle counted, because time greater than 30 minutes and in SIP temperature range. 4) no CIP-cycle counted, because of reaching the SIP-min limit. 69

Abbreviations

AO	Analog Output Interface
CP	Calibration Point
PA	Parameter
CPA	Calibration Parameter
CCO	Calibration Coefficient
PMC	Primary Measurement Channel
SMC	Secondary Measurement Channel
MC	Measurement Channel
SIP	Sterilization In Place
CIP	Cleaning In Place



Hamilton Europe, Asia & Africa
Parc Industrial Vial 10
CH-7013 Domat/Ems
Switzerland

Tel. +41 58 610 10 10

contact.pa.ch@hamilton.ch
www.hamiltoncompany.com

November 6th 2025
P/N: 111009574/00