KROHNE

KROHNE CT Products

OPTISONIC V6
Modbus manual

Protocol description
&
set-up

Liability, disclaimer

KROHNE takes great care to make sure the information in this document is correct, accurate and as complete as deemed necessary. Despite this, KROHNE rejects any responsibility for the consequences of missing or incorrect information in this manual.

Responsibility for setting up and operating a Modbus communication link, as described in this manual, rests solely with parties involved in the implementation and operation of such a data communication link.

It is KROHNE policy to maintain and update this document as necessary, however this is not a controlled document. Owners of this document can not claim that KROHNE is responsible for replacing this version of this document in case an updated version becomes available.

TABLE OF CONTENTS

1 INTRODUCTION

Scope

This manual describes how to use the Modbus protocol with the OPTISONIC V6 gas flow meter.

Introduction to Modbus

Modbus communication is based on the master-slave principle. Only the master can initiate transactions (requests), and only the addressed device (slave) responds. The master can also send a broadcast message (“message to all”); none of the slaves will respond to such a message.

The OPTISONIC V6 flow meter always acts as a Modbus compatible slave when communicating with host systems. Slaves are identified by means of a “device address”. Check the documentation to find the preset device address of your OPTISONIC V6 flow meter. In case necessary, the address can be reprogrammed. Contact the manufacturer for information about the procedures and tools that are needed for reprogramming the device address.

The Modbus protocol defines a message structure that Modbus enabled controllers will recognise and use, regardless of the type of network over which they communicate. It describes:

- the process a controller uses to request access to other devices,
- how to respond to requests from the other devices, and
- how errors will be detected and reported.

The Modbus request consists of:

- an address,
- a function code defining the requested action,
- data (if necessary for the requested function), and
- an error check for testing the integrity of the message.

The slave’s response contains:

- the slave address,
- data conform the request type, and
- an error check.

If the data integrity test fails, no response is sent back.
If a request cannot be processed an exception message is returned.

2 PHYSICAL COMMUNICATION LAYER

The Modbus over serial line protocol is a master-slave protocol. The physical layer can be half-duplex or full-duplex.

In case of the OPTISONIC V6 the physical layer is a half-duplex (two-wire) connection according to RS 485 specifications.

The end of a RS485 line has to be terminated by means of a resistor. This terminating resistor is included in the line driving circuit in the OPTISONIC V6.

Multiple OPTISONIC V6 meters may be connected to the same RS 485 line. In this case only the terminating resistor in the OPTISONIC V6 at the end of the line should be connected. The terminating resistors in the other OPTISONIC V6 meters on the line should be disconnected by means of the switch on the RS 485 driver printed circuit board. Default this switch is set to connect the line terminating resistor.

Because of the half-duplex operation, the RS 485 communication circuit in the OPTISONIC V6 is normally always in data receiving mode. Only in case it is requested to send it will automatically switch to data transmit mode for the time needed.

3 SERIAL TRANSMISSION FORMAT

Two transmission modes are defined for a Modbus data communication link:

- Modbus ASCII
- Modbus RTU.

Both transmission modes are supported, the user can select the desired mode along with the serial communication parameters (baud rate, parity).

The default configuration of the OPTISONIC V6 is Modbus RTU communication mode with "standard" Modbus settings.

Check chapter 7 for a list of programmable parameters and the default settings of these parameters. Except for the device addresses all these parameters must be the same for all controllers in the network.

3.1 ASCII mode

In the Modbus message each byte of data is coded as 2 ASCII characters; one to represent the upper 4 bits and another to represent the lower 4 bits. Each group of 4 bits is represented by a hexadecimal number, transmitted as an ASCII character from the range 0-9, A-F.

Standard serial communication parameters:

Start bits:	1
Data bits:	7
Parity:	odd/even/none
Stop bits:	1 stop bit if parity is used, or 2 stop bits if no parity is used.
Error check field:	Longitudinal Redundancy Check (LRC).

An advantage of ASCII mode is that it allows for a time interval up to 1 second between characters without causing a timeout. A disadvantage of ASCII mode is the larger message length.

3.2 RTU mode

Each byte of data is represented in the message by an equivalent number of bits (8).

The number of bits transmitted in the process of communicating one byte of information is sometimes also referred to as a "character". Note that this is not the same as an ASCII character.

Default serial communication parameters:

Baud rate:	19200
Data bits:	8
Parity:	even
Stop bits:	1
Error check field:	Cyclic Redundancy Check (CRC).

4 MODBUS MESSAGE FRAMING

ASCII mode

In ASCII mode a message starts with a colon character (:) and ends with a carriage return-linefeed.

Intervals up to one second can elapse between characters within the message. If the interval is longer, a timeout error occurs and the message is rejected.

RTU mode

In RTU mode a message starts with a silent time interval equivalent to at least 3.5 characters. The entire message frame must be transmitted as a continuous stream. If a silent interval of more than 3.5 character times occurs before completion of the frame, the receiving device flushes the incoming message and assumes that the next byte will be the address field for the new message.

Example of a typical message frame:

Mode	START	ADDRESS	FUNCTION	DATA	CHECKSUM	END
ASCII Mode	':'	2 characters	2 characters	N*2 characters	LRC 2 characters	CR-LF
RTU Mode	3.5 characters silent interval	8 bits	8 bits	N*8 bits	CRC 16 bits	3.5 character silent interval

4.1 The Address Field (Device Address)

The address field of a message frame contains:
ASCII mode: 2 characters
RTU mode: 8 bits

Valid slave addresses are 1 to 247.
Address 0 is used for a broadcast to address all slaves.

4.2 The Function Field

The function field of a message frame contains:
ASCII mode: 2 characters
RTU mode: 8 bits

Valid function codes lie in a range of 1 to 127.

The function code tells the slave which kind of action to perform.

The supported functions are listed in chapter 5.

A slave response always contains the function code of the request. If a function is not applicable, the slave sends an exception response. An exception is indicated by a returned function code with bit 8 (most significant bit) set.

4.3 The Data Field

The data field contains 8 bit values (bytes) in the range of 0 to FF hexadecimal.

In ASCII mode each 8 bit value is represented by 2 ASCII characters.

The data field of messages contains information which both master and slave use to perform an action. This includes the register address, quantity of registers, and the necessary data.

4.4 Error Checking Methods

Two error checking methods are defined for the Modbus protocol:

- Optional: an additional bit (parity bit) is appended to each character (or byte) for detecting errors during the transmission of individual characters (or bytes)
- Obligatory: two bytes (or characters) are appended to the message for detecting errors during the transmission of the message

As an even number of bit errors in one character (or byte) will not be detected using a parity bit, the second method is used to check the contents of the entire message.

Both character check and message check are generated in the transmitting device and appended to the message before transmission. The slave checks each character and the entire message frame during receipt.

The contents of the error checking field for the entire message depend on the transmission mode.

4.4.1 Error check in ASCII mode transmission

For detecting errors in the entire message the error-checking field contains two ASCII characters. The error check characters are the result of a Longitudinal Redundancy Check (LRC) calculation. This is performed on the message contents with exception of the beginning colon, the carriage return and line feed characters. The LRC characters are appended to the message as the last field preceding the CR-LF characters.

4.4.2 Error check in RTU mode transmission

For detecting errors in the entire message the error-checking field contains a 16-bit value implemented as two bytes. The error check value is the result of a Cyclic Redundancy Check (CRC) calculation performed on the message contents. The CRC field is appended to the message as the last field.

4.5 Transmission gaps

Gaps that exceed a specific value during the transmission of a message will be qualified as a transmission error.

4.5.1 ASCII mode

In ASCII mode the maximum time between 2 characters is one second. If a longer interval occurs, the message will be ignored and the search for a starting character (colon) is resumed.

4.5.2 RTU mode

In RTU mode the entire message frame must be transmitted as a continuous stream. If a silent interval of more than 3.5 character times occurs before completion of the frame, the receiving device ignores the message and assumes the next byte will be the device-address field of a new message.

4.6 Response time out

The master device has a predetermined time-out interval before aborting a transaction.

This interval shall be set long enough for any slave to respond normally.

5 SUPPORTED FUNCTIONS

A number of functions is available to perform operations on variables in the slave.

An operation can be a "read" operation to obtain the value of a variable or a "write" operation to assign a value to a variable. Variables are identified by means of their register number (address).

Typically in a Modbus slave, data can be stored in multiple areas that can be seen as different memories:

- Discrete Inputs: data from logical (also called binary, Boolean, or ON/OFF) inputs. By nature the data in this area is "read-only": the master has only access to read this data.
- Coils: logical (also called binary, Boolean, or ON/OFF) outputs. The master device may read the current state of an output, but may also set or change the state of an output.
- Input Registers: data, for example originating from electrical inputs of the slave or results from calculations in the slave, can be stored in "input registers". By nature the data in this area is "read-only": the master has only access to read this data.
- Holding Registers: the master has access to this area to read the data but as well to set or change the value of data (write).

As these register groups are located in apparently different memories, the addresses may overlap: for example, an input register having address 100 can exist and at the same time a holding register having address 100. These are not the same: which one will be selected for an operation is implied from the function code, referring to an input register or to a holding register, for example.

The OPTISONIC V6 does not use Discrete Inputs or Coils, but only Input Registers and Holding Registers.

Variables are grouped according to data type and dependant of being input registers (read-only data) or being holding registers (read/write data). An address range is assigned to each variable type, subdivided in input registers (read-only) and holding registers (read/write).

The OPTISONIC V6's address ranges of Input Registers and Holding Registers do not overlap. Accessing a specific register address is therefore unambiguous. The functions "read register" and "read input" could both be used to effectively access the same register/address. However, in this application, functions shall still be used consistent with the type of memory they are intended to be used for.

In the master and the slave register addresses are referenced (counted) starting from 1. However, the address range used in the message during in the transmission starts from 0. As an example, when referencing address 4001, the address actually present in the message will be 4000.

On an application level the user will not notice this, as during the coding and decoding of the message this offset of 1 will be taken into account. However, when the message – as it is transmitted – is analyzed and checked one has to be aware of this offset.

When functions which do not support broadcast requests, are accessed with a broadcast address, the request will be ignored.

5.1 Function 01: READ COILS

Function 01 reads the status of 1 to 2000 contiguous logical (Boolean or ON/OFF) variables.

This function is not used, as in this application Boolean (or logical) variables are not used as individual entities. Boolean variables are represented by means of specific bits packed in 32 bit data word (type "Long").

5.2 Function 02: READ DISCRETE INPUTS

Function 02 reads the status of 1 to 2000 contiguous logical (Boolean or ON/OFF) variables.

This function is not used, as in this application Boolean (or logical) variables are not used as individual entities. Boolean or logical variables are represented by means of specific bits packed in 32 bit data word (type "Long").

5.3 Function 03: READ HOLDING REGISTERS

Function 03 reads the contents of 1 to 125 contiguous holding registers in the slave.

The maximum number of registers at each request is limited to 125 16-bit registers: 125 integers, 62 long integers, 62 floats, 31 doubles or 31 long longs.

Request

The request message specifies the starting register and the quantity of registers to be read. Registers are addressed starting from zero. Registers 1-16 are addressed as 0-15.

Example

A request to read from slave device 17, registers 40108-40110 (decimal), or starting from 9CAC (hex):

Header	Slave Address	Function	Starting address		Number of data items		Error check	Trailer
			Hi	Low	Hi	Low		
--	11(h)	03(h)	9C(h)	AB(h)	00(h)	03(h)	--	--

Response:

Header	Slave address	Function	Byte count	Data						Error check	Trailer
				(Reg. 40108 Hi)	(Reg. 40108 Low)	(Reg. 40109 Hi)	(Reg. 40109 Low)	(Reg. 40110 Hi)	(Reg. 40110 Low)		
--	11(h)	03(h)	06(h)	02(h)	2B(h)	00(h)	00(h)	00(h)	64(h)	--	--

For each register the first byte contains the high order byte, the second the low order byte.

The contents of register 40108 are shown as the two byte values of 02 2B hex (555 decimal). The contents of register 40109 is 00 00 hex (0 decimal) and of register 40110 is 00 64 hex (100 decimal).

If the request is not applicable, an exception response will be sent. **See chapter 5.10 for exception responses.**

5.4 Function 04: READ INPUT REGISTERS

Function 04 performs a “read” operation, similar to function 03. The difference is that function 04 addresses input registers (which are “read-only”), whereas function 03 addresses holding registers (which are “read/write”).

5.5 Function 05: WRITE SINGLE COIL

Function 05 writes the status of a logical (Boolean or ON/OFF) variable.

This function is not used, as in this application Boolean variables are not used as individual entities. Boolean variables are represented by means of specific bits packed in 32 bit data word (type “Long”).

5.6 Function 06: WRITE SINGLE HOLDING REGISTER

Function 06 presets a value into a single holding register.

When the address is a broadcast, all slaves will process the request.

Request

The request specifies the register reference to be written. Registers are addressed starting from zero.

Registers 1-16 are addressed as 0-15. The value to be written is specified in the data field, which is a 16-bit value.

Example

Request for slave 17 to preset register 40002 (decimal), 9C42 (hex) to 00 03 (hex).

Header	Slave Address	Function	Register Address		Data		Error Check	Trailer
			Hi	Low	Hi	Low		
--	11(h)	06(h)	9C(h)	41(h)	00(h)	03(h)	--	--

Response

The response message is an echo of the request, returned after the register contents has been written.

Header	Slave Address	Function	Register Address		Data		Error Check	Trailer
			Hi	Low	Hi	Low		
--	11(h)	06(h)	9C(h)	41(h)	00(h)	03(h)	--	--

If the request is not applicable, an exception response will be sent. **See chapter 5.10 for exception responses.**

5.7 Function 8: DIAGNOSTICS

Function 8 provides a test for checking the communication system between the master and the slave.

Request

The function uses a two-byte sub-function field in the request to define the test to be performed:

Header	Slave address	Function	Sub-function	Data Hi+Lo	Error check	Trailer
--	11(h)	08(h)	00 00(h)	A1B8 (h)	--	--

All sub-functions are supported.

5.8 Function 15: WRITE MULTIPLE COILS

Function 15 writes the status of 1 to 2000 contiguous logical (Boolean or ON/OFF) variables.

This function is not used, as in this application Boolean variables are not used as individual entities. Boolean variables are represented by means of specific bits packed in 32 bit data word (type “Long”).

5.9 Function 16: WRITE MULTIPLE HOLDING REGISTERS

Function 16 writes the contents of 1 to 123 contiguous holding registers in the slave.

When the address is a broadcast, the function pre-sets the same register references in all attached slaves.

Request

The request message specifies the register references to be pre-set. Registers are addressed starting at zero (register 1 is addressed as 0).

Example

An example of a request for slave device 17 to pre-set two registers starting at 40002 (decimal), 9C42 (hexadecimal) to 00 0A end 01 02 hex:

Header	Slave Address	Func-tion	Starting address		Quantity Registers		Byte counts	Data				Error check	Trailer
			Hi	Low	Hi	Low		Hi	Low	Hi	Low		
--	11(h)	10(h)	9C(h)	41(h)	00(h)	02(h)	04(h)	00(h)	0A(h)	01(h)	02(h)	--	--

Response

The normal response returns the slave address, the function code, starting address, and quantity of registers pre-set:

Header	Slave Address	Function	Starting Address		Quantity Of points		Error check	Trailer
			Hi	Low	Hi	Low		
--	11(h)	10(h)	9C(h)	41(h)	00(h)	02(h)	--	--

If the request is not applicable, an exception response will be sent. **See chapter** 5.10 **for exception responses**.

5.10 Exception Responses

Except for broadcast messages, a master device expects a normal response, when it sends a request to a slave device. One of the four possible events can occur upon the master device’s request:

- If the slave device receives the request without a communication error and can handle the request normally, it returns a normal response.
- If the slave does not receive the request due to a communication error, no response is returned. The master program will eventually process a timeout condition for the request.
- If the slave receives the request, but detects a communication error (parity, CRC, LRC), no response is returned. The master program will eventually process a timeout condition for the request.
- If the slave receives the request without a communication error, but cannot handle it, the slave will return an exception response informing the master of the nature of the error.

The exception response message has two fields that differentiate it from a normal response.

Function Code Field

In a normal response the slave echoes the function code of the original request in the function code field of the response. In an exception response the slave sets the most significant bit of the function code to 1.

The master recognises the exception response by means of this bit and can examine the data field for the exception code.

Data field

In an exception response the slave returns an exception code in the data field. By means of this exception code the slave reports a reason for not being able to respond normally.

The exception response message:

Header	Slave address	Function	Exception code	Error check	Trailer

Exception codes

Code	Name	Meaning
01	Illegal function	The function code in the request is not an allowable action for the slave.
02	Illegal data address	The data address received in the request is not an allowable address for the slave.

6 HANDLING OF LARGE DATA TYPES

The standard Modbus specification does not explain how data types larger than 16 bits should be handled. As larger data types are stored in a multiple of 16 bit registers, such data can be accessed by means of “read” or “write” operation on a series of consecutive 16 bit registers.

Function 03 (read multiple holding registers), function 04 (read input registers), function 06 (write single holding register), and function 16 (write multiple holding registers) are used to read or modify these data types.

Taking the data type into account, the addressing can be “optimized” accordingly, this is implemented in an addressing mode not compatible with the original Modicon concept:

- In the original “Modicon compatible mode” one address is assigned and counted for each 16 bit register. To hold for example a 64-bit integer value, 4 addresses would be occupied. Therefore, in order to address the next variable of this type, the address has to be incremented by 4.
- In “not-Modicon compatible mode” addresses are incremented by 1 for each next variable. For example, in order to read or write the next 64-bit variable, the register area to be read will automatically be shifted by 4 16 bit registers.

The OPTISONIC V6 is by default configured for Modicon compatible addressing.

The supported data types are:

- Integer (16 bit)
- Long integer (32 bit)
- Float (single precision floating-point, 32 bit)
- Double (double precision floating-point, 64 bit)
- Long long (64-bit Integer)

The register ranges for each data type:

Data type	Address range	Number of registers to request for each data type	
		Modicon compatible	Not Modicon compatible
Integer (16 bit)	3000..3999	1	1
Long integer (32 bit)	5000..5999	2	1
Double (64 bit)	6000..6999	4	1
Float (32 bit)	7000..7999	2	1
Long long (64 bit)	8000..8999	4	1

Note that in **Modicon compatible mode** each data type larger than 16 bits should be addressed as an appropriate number of 16-bit registers. For instance the first float is located at address 7000/7001; the next float is located at address 7002/7003.

A double would be accessed by four 16-bit registers, so the first double 6000/6001/6002/6003 and the next double 6004/6005/6006/6007.

The data in the chapter 8, “MODBUS REGISTER MAPPING”, is printed both as it should be addressed in Modicon compatible and as well as in **not-Modicon compatible mode**.

6.1 Integer (16 bit), Transmit Sequence

Integers are transmitted and stored with the most significant part first.

Example

Integer value 1790 decimal (6FE hexadecimal) is transmitted as:

First transmitted byte in data field	Second transmitted byte in data field
06	FE

6.2 Long integer (32 bit), Transmit Sequence

Example

Long integer value 305419896 (12345678 hexadecimal).

Long integers could be transmitted in two possible ways. The transmit order in both modes:

Normal mode	(1) 12_h	(2) 34_h	(3) 56_h	(4) 78_h
Reversed mode	(3) 56_h	(4) 78_h	(1) 12_h	(2) 34_h

6.3 Single precision floating-point (32 bit), Transmit Sequence

Single precision floating-point numbers are stored in 32-bit registers, represented using the IEEE 754 encoding. In IEEE 754-2008 the 32-bit base 2 format is officially referred to as **binary32**. It was called **single** in IEEE 754-1985.

The IEEE 754 standard specifies a binary32 as having:
- Sign bit: 1 bit
- Exponent width: 8 bits
- Significand (also known as mantissa) precision: 24 (23 explicitly stored)

The true significand (mantissa) includes an implicit leading bit with value 1 unless the exponent is stored with all zeros. Thus only 23 bits of the significand (mantissa) appear in the memory format but the total precision is 24 bits (equivalent to $\log10(2^{24}) \approx 7.225$ decimal digits). The bits are laid out as follows:

Sign + (Biased) Exponent	Exponent + Mantissa 3 (high)	Mantissa 2	Mantissa 1 (low)
SEEE EEEE	E MMM MMMM	MMMM MMMM	MMMM MMMM

The single precision binary floating-point exponent is encoded using an offset binary representation, with the zero offset being 127; also known as exponent bias in the IEEE 754 standard.

Example:

The float number 4.125977 will give the IEEE 754representation.

Sign	Exponent	Mantissa
0	1000 0001	(1) 000 0100 0000 1000 0000 0000

A positive sign
A biased exponent of 129 (81 hexadecimal) is exponent 2.
Mantissa = 4 + 1/8 + 1/1024. Note that the first bit is not stored!

Floats could be transmitted in two ways. The transmit order in both modes:

IEEE 754	(1) 40_h	(2) 84_h	(3) 08_h	(4) 00_h
Normal mode	(1) 40_h	(2) 84_h	(3) 08_h	(4) 00_h
Reversed mode	(3) 08_h	(4) 00_h	(1) 40_h	(2) 84_h

6.4 Double precision floating-point (64 bit), Transmit Sequence

Double precision floating-point numbers are stored in 64-bit registers, represented using the IEEE 754 encoding. In IEEE 754-2008 the 64-bit base 2 format is officially referred to as **binary64**. It was called **double** in IEEE 754-1985.

The IEEE 754 standard specifies a binary64 as having:

- Sign bit: 1 bit
- Exponent width: 11 bits
- Significand (also known as mantissa) precision: 53 (52 explicitly stored)

The true significand (mantissa) includes an implicit leading bit with value 1 unless the exponent is stored with all zeros. Thus only 52 bits of the significand (mantissa) appear in the memory format but the total precision is 53 bits (equivalent to $\log10(2^{53}) \approx 16$ decimal digits). The bits are laid out as follows:

Sign + (Biased) Exponent	Exponent + Mantissa	Mantissa 6	Mantissa 5
SEEE EEEE	EEEE MMMM	MMMM MMMM	MMMM MMMM

Mantissa 4	Mantissa 3	Mantissa 2	Mantissa 1
MMMM MMMM	MMMM MMMM	MMMM MMMM	MMMM MMMM

Example

The double number 4.125000001862645 will give the IEEE representation:

Sign	Exponent	Mantissa
0	100 0000 0001	(1)0000 1000 0000 0000 0000 0000 0000 0010 0000 0000 0000 0000 0000

A positive sign
A biased exponent of 1025 (401 hexadecimal) is exp. 2
Mantissa = 4 + 1/8 + 1/536870912. Note that the first bit is not stored!

Doubles could be transmitted in two ways. The transmit order in both modes:

IEEE 754	(1) 40_h	(2) 10_h	(3) 80_h	(4) 00_h	(5) 00_h	(6) 20_h	(7) 00_h	(8) 00_h
Normal mode	(1) 40_h	(2) 10_h	(3) 80_h	(4) 00_h	(5) 00_h	(6) 20_h	(7) 00_h	(8) 00_h
Reversed mode	(3) 80_h	(4) 00_h	(1) 40_h	(2) 10_h	(7) 00_h	(8) 00_h	(5) 00_h	(6) 20_h

6.5 Long long (64 bit integer), Transmit Sequence

Example

64 bit integer value 4.616.330.355.545.210.880 (= 4010 8000 0020 0000 hexadecimal).

64 bit integers could be transmitted in two ways. The transmit order in both modes:

Normal mode	(1) 40_h	(2) 10_h	(3) 80_h	(4) 00_h	(5) 00_h	(6) 20_h	(7) 00_h	(8) 00_h
Reversed mode	(3) 80_h	(4) 00_h	(1) 40_h	(2) 10_h	(7) 00_h	(8) 00_h	(5) 00_h	(6) 20_h

6.6 Maximum number requested items

The maximum amount of data that can be sent in a single response limits the amount of items that can be requested in a single query. The table below shows the maximum number of items per data type:

Data type	**Number of items**
Boolean	2000
Integer (16 bit)	125
Long integer (32 bit)	62
Float (32 bit floating point)	62
Double (64 bit floating point)	31
Long long (64 bit integer)	31

7 DEFAULT SETTINGS

By means of a number of parameters the Modbus communication link can be adjusted to one's needs or preferences. When the OPTISONIC V6 meter is delivered these parameters are set to default values as listed below:

Port 0:

Baud rate:	19200
Data bits:	8
Stop bits:	1
Parity:	Even
Modbus Mode:	RTU
Modbus End Code:	2
Addressing Mode:	Modicon compatible
Representation Mode:	Normal
Modbus Address:	237
Modbus Start Gap:	40 (bits)
Modbus End Gap:	20 (bits)
Modbus Start Code:	":" (colon character)
Modbus End Code1:	ASCII 13 (carriage return, CR)
Modbus End Code2:	ASCII 10 (line feed, LF)
Modbus Time Out:	1 second

Port 1:

Baud rate:	115200
Data bits:	8
Stop bits:	1
Parity:	Even
Modbus Mode:	RTU
Modbus End Code:	2
Addressing Mode:	Modicon compatible
Representation Mode:	Normal
Modbus Address:	237
Modbus Start Gap:	40 (bits)
Modbus End Gap:	20 (bits)
Modbus Start Code:	":" (colon character)
Modbus End Code1:	ASCII 13 (carriage return, CR)
Modbus End Code2:	ASCII 10 (line feed, LF)
Modbus Time Out:	1 second

8 MODBUS REGISTER MAPPING

Registers are mapped to specific address ranges according to both data and register type:

Data type	Register type	Read command(s)	Write command(s)	Address range
Integer (16 bit)	Input Register	4	n.a.	3000..3499
	Holding Register	3	6, 16	3500..3999
Long integer (32 bit)	Input Register	4	n.a.	5000..5499
	Holding Register	3	6, 16	5500..5999
Double (64 bit)	Input Register	4	n.a.	6000..6499
	Holding Register	3	6, 16	6500..6999
Float (32 bit)	Input Register	4	n.a.	7000..7499
	Holding Register	3	6, 16	7500..7999
Long long (64 bit)	Input Register	4	n.a.	8000..8499
	Holding Register	3	6, 16	8500..8999

Note: The relative addresses listed in the tables below are addresses relative to the starting address of the designated register group.

8.1 Input Registers (read-only): Integer (16-bit); address range 3000-3499

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Test register	TestRegister uint16	Register reserved for testing communications and protocol handling with this type of register, without affecting the operation of the flow meter.	0	3000	3000-3001 (2 regs)	-

8.2 Holding Registers (read/write): Integer (16-bit); address range 3500-3999

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Test register	TestRegister uint16	Register reserved for testing communications and protocol handling with this type of register, without affecting the operation of the flow meter.	0	3500	3500-3501 (2 regs)	-

8.3 Input Registers (read-only): Long integer (32-bit); address range 5000-5499

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Test register	TestRegister uint32	Register reserved for testing communications and protocol handling with this type of register, without affecting the operation of the flow meter.	0	5000	5000-5001 (2 regs)	-

Alarm / status events

Parameter / Variable (short description)	Register Name	Bit	Bit Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Down All Channels, no custody transfer standard (FAILED)	Alarm&Status Pipe	0	A FailAll	Alarm flag indicating none of the acoustic paths is operational.	1	5001	5002-5003 (2 regs)	[status]
Down Channels, Custody Transfer unreliable (FAILED)	Alarm&Status Pipe	1	A FailUnreliable	Alarm flag indicating some paths out of operation, reading **NOT valid for custody transfer** measurement.	1	5001	5002-5003 (2 regs)	[status]
Down Channels, but Custody Transfer reliable	Alarm&Status Pipe	2	A FailReliable	Alarm flag indicating some paths out of operation, reading **valid for custody transfer** measurement.	1	5001	5002-5003 (2 regs)	[status]

Parameter / Variable (short description)	Register Name	Bit	Bit Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Out of range FLOW	Alarm&Status Pipe	4	A OOR_Flow	Status signal flow rate is out of range, activated at overspeeding by 25%.	1	5001	5002-5003 (2 regs)	[status]
Correction out of range REYNOLDS	Alarm&Status Pipe	5	A OOR_Reynolds	Alarm flag indicating Reynolds number out of range	1	5001	5002-5003 (2 regs)	[status]
Correction out of range PATH SUBSITITION	Alarm&Status Pipe	6	A OOR_Substit	Alarm flag indicating gas velocity is out of range where path substitution can be applied. This signal only appears in case of active path substitution (at least one path failing) and gas velocity is out min/max limits OR Reynolds number is out of min/max limits OR no valid velocity profile is available.	1	5001	5002-5003 (2 regs)	[status]
Flow direction	Alarm&Status Pipe	16	S Flow_Direction	Status signal indicating flow direction: 0 = Forward flow 1 = Reverse flow	1	5001	5002-5003 (2 regs)	[status]
Low flow cut-off	Alarm&Status Pipe	17	S LowFlowCutOff	Status signal indicating low flow cut-off: 0 = Flow velocity above threshold. 1 = Flow velocity below threshold.	1	5001	5002-5003 (2 regs)	[status]
Reset occurred: All Totalisers	Alarm&Status Pipe	18	S ResetTotals	Status signal indicating totalisers have been reset. Note: This operation is only allowed for specially authorized personnel.	1	5001	5002-5003 (2 regs)	[status]
Channel 1 Unreliable	Alarm&Status Channel_1	0	A Ch_Unreliable	Alarm flag indicating path error (channel 1): channel unreliable.	2	5002	5004-5005 (2 regs)	[status]
Channel 1 Down	Alarm&Status Channel_1	1	A Ch_Down	Alarm flag indicating path error (channel 1): channel down.	2	5002	5004-5005 (2 regs)	[status]
Channel 1 Down: Deviation SoS too large	Alarm&Status Channel_1	2	A Ch_Deviation_SOS	Alarm flag indicating path error (channel 1): SOS deviation too large.	2	5002	5004-5005 (2 regs)	[status]
Channel 1 Down: Signal lost	Alarm&Status Channel_1	3	A Ch_Signal_Lost	Alarm flag indicating path failure (channel 1): signal lost.	2	5002	5004-5005 (2 regs)	[status]

Parameter / Variable (short description)	Register Name	Bit	Bit Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Channel 2 Unreliable	Alarm&Status Channel_2	0	A Ch_Unreliable	Alarm flag indicating path error (channel 2): channel unreliable.	3	5003	5006-5007 (2 regs)	[status]
Channel 2 Down	Alarm&Status Channel_2	1	A Ch_Down	Alarm flag indicating path error (channel 2): channel down	3	5003	5006-5007 (2 regs)	[status]
Channel 2 Down: SoS Deviation too large	Alarm&Status Channel_2	2	A Ch_Deviation_SOS	Alarm flag indicating path error (channel 2): SOS deviation too large.	3	5003	5006-5007 (2 regs)	[status]
Channel 2 Down: Signal lost	Alarm&Status Channel_2	3	A Ch_Signal_Lost	Alarm flag indicating path failure (channel 2): signal lost	3	5003	5006-5007 (2 regs)	[status]
Channel 3 Unreliable	Alarm&Status Channel_3	0	A Ch_Unreliable	Alarm flag indicating path error (channel 3): channel unreliable.	4	5004	5008-5009 (2 regs)	[status]
Channel 3 Down	Alarm&Status Channel_3	1	A Ch_Down	Alarm flag indicating path error (channel 3): channel down.	4	5004	5008-5009 (2 regs)	[status]
Channel 3 Down: SoS Deviation too large	Alarm&Status Channel_3	2	A Ch_Deviation_SOS	Alarm flag indicating path error (channel 3): SOS deviation too large.	4	5004	5008-5009 (2 regs)	[status]
Channel 3 Down: Signal lost	Alarm&Status Channel_3	3	A Ch_Signal_Lost	Alarm flag indicating path failure (channel 3): signal lost.	4	5004	5008-5009 (2 regs)	[status]
Reserved	Reserved	-	Reserved	Reserved	5-7	5005-5007 (3 regs)	5010-5015 (6 regs)	-
Process Temperature Out of range	Alarm Inputs	0	A_Temperature_OOR	Alarm flag, measured body temperature is out of range.	8	5008	5016-5017 (2 regs)	[status]
Process temperature: Override value used	Alarm Inputs	1	A_Temperature_OVR	Status/alarm flag indicating manual input of temperature value overrides measured temperature (process temperature input); will appear in case alarm parameter nr 6 set to enabled and Override Mode (User Control) set to enabled (mode 1 or 2); Mode 3 will not generate an alarm.	8	5008	5016-5017 (2 regs)	[status]

Parameter / Variable (short description)	Register Name	Bit	Bit Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Rollover Totaliser Process Forward	Status Totalisers	1	S_Fwd_Process_Roll	Status signal, totaliser has reached maximum value that can be represented and displayed, totalising is continued starting from zero.	9	5009	5018-5019 (2 regs)	[status]
Rollover Totaliser Process Reverse	Status Totalisers	5	S_Rev_Process_Roll	Status signal, totaliser has reached maximum value that can be represented and displayed, totalising is continued starting from zero.	9	5009	5018-5019 (2 regs)	[status]
Rollover Totaliser FAIL Process Forward	Status Totalisers	9	S_FwdFail_Process_Roll	Status signal, totaliser has reached maximum value that can be represented and displayed, totalising is continued starting from zero.	9	5009	5018-5019 (2 regs)	[status]
Rollover Totaliser FAIL Process Reverse	Status Totalisers	13	S_RevFail_Process_Roll	Status signal, totaliser has reached maximum value that can be represented and displayed, totalising is continued starting from zero.	9	5009	5018-5019 (2 regs)	[status]

8.4 Holding Registers (read/write): Long integer (32-bit), address range 5500-5999

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Test register	TestRegister uint32	Register reserved for testing communications and protocol handling with this type of register, without affecting the operation of the flow meter.	0	5500	5500-5501 (2 regs)	-

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Date and time	ANSI time	Built-in real time clock: date and time according to ANSI standard: number of seconds elapsed since January 1st, 1970, 00:00:00 hours. Note: Although this is a Holding Register, this register is read-only. Use the KROHNE Flowmeter Configuration and Monitoring software to set or adjust the real-time clock.	1	5501	5502-5503 (2 regs)	s

8.5 Input Registers (read-only): Double (64-bit floating-point), address range 6000-6499

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Test register	TestRegister double64	Register reserved for testing communications and protocol handling with this type of register, without affecting the operation of the flow meter.	0	6000	6000-6003 (4 regs)	-

8.6 Holding Registers (read/write): Double (64-bit floating-point), address range 6500-6999

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Test register	TestRegister double64	Register reserved for testing communications and protocol handling with this type of register, without affecting the operation of the flow meter.	0	6500	6500-6503 (4 regs)	-

8.7 Input Registers (read-only): Float (32-bit floating-point), address range 7000-7499

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Test register	TestRegister float32	Register reserved for testing communications and protocol handling with this type of register, without affecting the operation of the flow meter.	0	7000	7000-7001 (2 regs)	-

Transit Time Measuring Process Values

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Gain "downstream" (= gain for transducer B), Channel 1-3	Ch_GainAB	Gain for transmission of signal from transducer A towards transducer B, array with values for channel 1 through 3.	1-3	7001-7003 (3 regs)	7002-7007 (6 regs)	dB
Reserved	Reserved	Reserved	4-6	7004-7006 (3 regs)	7008-7013 (6 regs)	-
Gain "upstream" (= gain for transducer A), Channel 1-3	Ch_GainBA	Gain for transmission of signal from transducer B towards transducer A, array with values for channel 1 through 3.	7-9	7007-7009 (3 regs)	7014-7019 (6 regs)	dB
Reserved	Reserved	Reserved	10-12	7010-7012 (3 regs)	7020-7025 (6 regs)	-
SNR "downstream" (=received at trd. B), Channel 1-3	Ch_SNratioAB	Signal-to-noise ratio for transmission of signal from transducer A towards transducer B, array with values for channel 1 through 3.	13-15	7013-7015 (3 regs)	7026-7031 (6 regs)	dB
Reserved	Reserved	Reserved	16-18	7016-7018 (3 regs)	7032-7037 (6 regs)	-
SNR "upstream" (=received at trd. A), Channel 1-3	Ch_SNratioBA	Signal-to-noise ratio for transmission of signal from transducer B towards transducer A, array with values for channel 1 through 3.	19-21	7019-7021 (3 regs)	7038-7043 (6 regs)	dB
Reserved	Reserved	Reserved	22-24	7022-7024 (3 regs)	7044-7049 (6 regs)	-

Measured / entered process parameters

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Temperature process value	TemperatureProces	Value used for calculations, can either be the measured process value, a calculated (indirectly determined) value or a manual setting (fixed value).	25	7025	7050-7051 (2 regs)	°C
Dynamic Viscosity at Process conditions	ViscosityDynamic-Proces	Value used for calculations, can either be the measured process value, a calculated (indirectly determined) value or a manual setting (fixed value).	26	7026	7052-7053 (2 regs)	cP
Density Process	DensityProces	Value used for calculations, can either be the measured process value, a calculated (indirectly determined) value or a manual setting (fixed value).	27	7027	7054-7055 (2 regs)	kg/m^3

Calculated flow variables

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Flow Process	Flow_Proces	Measured volume flow rate at process conditions.	28	7028	7056-7057 (2 regs)	m^3/s
Velocity Process	Velo_Proces	Measured gas velocity (integrated value from all paths) at process conditions.	29	7029	7058-7059 (2 regs)	m/s
Speed of Sound	SoS	Measured speed of sound, average of all acoustic paths, corrected for mach effect.	30	7030	7060-7061 (2 regs)	m/s
Channel Velocity, Channel 1-3	Ch_VeloRaw	“Raw” gas velocity as observed on each channel; array with values for channel 1 through 3.	31-33	7031-7033 (3 regs)	7062-7067 (6 regs)	m/s
Reserved	Reserved	Reserved	34-36	7034-7036 (3 regs)	7068-7073 (6 regs)	-
Channel SOS, Channel 1-3	Ch_SoS	Speed of sound as observed on each channel; array with values for channel 1 through 3.	37-39	7037-7039 (3 regs)	7074-7079 (6 regs)	m/s

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Reserved	Reserved	Reserved	40-42	7040-7042 (3 regs)	7080-7085 (6 regs)	-

Statistics / diagnostics variables

Averages and standard deviations are calculated for a number of flow variables. The number of measurement values that are included in this calculation are specified by means of a preset parameter. Averages and standard deviations are evaluated and updated each time a number of measurements equal to this parameter has been collected. Collecting data for a new block of data to be evaluated then starts again from there.

As an exception to the rule above, for the flow standard deviation tau, the preset number of measurement values is evaluated every time a new measurement value becomes available. The series of measurements values then contains the most recent values and shifts with each acquired sample (measurement data), like a running average.

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Channel reliability, channel 1-3	Ch_Reliab	Reliability per channel; array with values for channel 1 through 3	43-44	7043-7045 (3 regs)	7086-7091 (6 regs)	%
Reserved	Reserved	Reserved	46-48	7046-7048 (3 regs)	7092-7097 (6 regs)	-
Channel velocity, average, channel 1-3	Ch_AV_Velocity	Average gas velocity, as observed on each channel; array with values for channel 1 through 3.	49-51	7049-7051 (3 regs)	7098-7103 (6 regs)	m/s
Reserved	Reserved	Reserved	52-54	7052-7054 (3 regs)	7104-7109 (6 regs)	-
Channel SoS, average, channel 1-3	Ch_AV_SoS	Average speed-of-sound, as observed on each channel; array with values for channel 1 through 3.	55-57	7055-7057 (3 regs)	7110-7115 (6 regs)	m/s
Reserved	Reserved	Reserved	58-60	7058-7060 (3 regs)	7116-7121 (6 regs)	-
SoS Average	AV_SoS	Speed of Sound value, average.	61	7061	7122-7123 (2 regs)	m/s

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Flow average	AV_FlowProces	Volume flow rate value, average.	62	7062	7124-7125 (2 regs)	m^3/s
Flow standard deviation	SD_FlowProces	Standard deviation of volume flow rate (“batchwise calculation”).	63	7063	7126-7127 (2 regs)	%
Flow standard deviation Tau	SD_FlowProcesTau	Standard deviation of volume flow rate ("running calculation”).	64	7064	7128-7129 (2 regs)	%
SoS standard deviation	SD_SoS	Speed of Sound standard deviation.	65	7065	7130-7131 (2 regs)	%
Channel velocity, standard deviation, channel 1-3	Ch_SD_Velocity	Gas velocity, standard deviation, as observed on each channel; array with values for channel 1 through 3.	66-68	7066-7068 (3 regs)	7132-7137 (6 regs)	%
Reserved	Reserved	Reserved	69-71	7069-7071 (3 regs)	7138-7143 (6 regs)	-
Channel SoS, standard deviation, channel 1-3	Ch_SD_SoS	Speed of Sound, standard deviation, as observed on each channel; array with values for channel 1 through 3.	72-74	7072-7074 (3 regs)	7144-7149 (6 regs)	%
Reserved	Reserved	Reserved	75-77	7075-7077 (3 regs)	7150-7155 (6 regs)	-
Overall weighted reliability	Weighted_Reliability	The reliabilities of the individual paths are combined in an overall figure using the weighing factors of the paths.	78	7078	7156-7157 (2 regs)	%

8.8 Holding Registers (read/write): Float (32-bit) floating-point, address range 7500-7999

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Test register	TestRegister float32	Register reserved for testing communications and protocol handling with this type of register, without affecting the operation of the flow meter.	0	7500	7500-7501 (2 regs)	-

Flowmeter configuration

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Internal diameter	Diameter	Internal diameter of meter body at the measuring section.	1	7501	7502-7503 (2 regs)	m
Meter Constant forward	MeterConstant_Fwd	Meter constant, forward flow direction.	2	7502	7504-7505 (2 regs)	-
Meter Constant reverse	MeterConstant_Rev	Meter constant, reverse flow direction.	3	7503	7506-7507 (2 regs)	-
Thermal expansion factor pipe material Alpha	ThermExp_Alpha_Pipe	Property of the material of the meter body, used to calculate the effect of thermal expansion.	4	7504	7508-7509 (2 regs)	1/°C
Reference temperature Body expansion	ThermExp_Reference-Temp	Temperature at which the internal diameter of meter body at the measuring section is specified.	5	7505	7510-7511 (2 regs)	°C
Channel Calibrated Path Length L, channel 1-3	Ch_PathLength_Cal	Length of acoustic path length between two transducers of a measuring path; array with values for channel 1 through 3.	6-8	7506-7508 (3 regs)	7512-7517 (6 regs)	m
Reserved	Reserved	Reserved	9-11	7509-7511 (3 regs)	7518-7523 (6 regs)	-
Channel Measuring Angle, channel 1-3	Ch_Angle_Measuring	Angle at which the ultrasonic beam intersects with the meter body; array with values for channel 1 through 3.	12-14	7512-7514 (3 regs)	7524-7529 (6 regs)	Deg
Reserved	Reserved	Reserved	15-17	7515-7517 (3 regs)	7530-7535 (6 regs)	-
Flow Full Scale Forward	Fwd_MaxFlowRate	Indicated value (full scale) at flow of 100% of rated volume flow in forward flow direction	18	7518	7536-7537 (2 regs)	m^3/s
Flow Full Scale Reverse	Rev_MaxFlowRate	Indicated value (full scale) at flow of 100% of rated volume flow in reverse flow direction	19	7519	7538-7539 (2 regs)	m^3/s
Low Flow cutoff Forward	Fwd_LowFlowCutoff	For forward flow rates with smaller magnitude than this cut-off value, totalisers stop counting and outputs (such as the frequency output) will be disabled; flow rate indication will still be available	20	7520	7540-7541 (2 regs)	m/s
Low Flow cutoff Reverse	Rev_LowFlowCutoff	For reverse flow rates with smaller magnitude than this cut-off value, totalisers stop counting and outputs (such as the frequency output) will be disabled; flow rate indication will still be available	21	7521	7542-7543 (2 regs)	m/s

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Low Flow cutoff Threshold	Low_Flowcut_-ThresHold	Hysteresis preventing frequent on/off switching of low flow cut-off	22	7522	7544-7545 (2 regs)	%

Alarm configuration

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Allowed Deviation SOS per channel on SOS average Pipe	SOSDev	Defines alarm limit for deviation of measured SOS of any acoustic path from average value of all paths (all SoS values calculated with mach-correction).	23	7523	7546-7547 (2 regs)	m/s
Input Temperature Proces: LimitHigh	T_LimitHigh	Upper limit for process temperature measurement.	24	7524	7548-7549 (2 regs)	°C
Input Temperature Proces: LimitLow	T_LimitLow	Lower limit for process temperature measurement.	25	7525	7550-7551 (2 regs)	°C

Viscosity configuration

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Dynamic Viscosity at Reference Temperature	VD_Reference_Visc	Input value of dynamic viscosity, used as base value to correct reference viscosity to actual viscosity at process temperature and pressure.	26	7526	7552-7553 (2 regs)	cP
Reference Density	D_Reference_Dens	Input value of density, used as base value to correct reference density to actual density at process temperature and pressure.	27	7527	7554-7555 (2 regs)	kg/m^3

Override control

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Temperature Process Manual override value	Ovr_Value_T	Manually set temperature value, to be taken into effect in case of failing temperature measurement	28	7528	7556-7557 (2 regs)	°C

8.9 Input Registers (read-only): Long long (64-bit integer), address range 8000-8499

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Test register	TestRegister uint64	Register reserved for testing communications and protocol handling with this type of register, without affecting the operation of the flow meter.	0	8000	8000-8003 (4 regs)	-

Totalisers

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Totaliser PROCESS Forward	Fwd_Proces	Accumulated volume (with normal working meter) in forward flow direction, at line conditions.	1	8001	8004-8007 (4 regs)	ml (=10^{-6} m^3)
Totaliser PROCESS Reverse	Rev_Proces	Accumulated volume (with normal working meter) in reverse flow direction, at line conditions.	2	8002	8008-8011 (4 regs)	ml (=10^{-6} m^3)
Totaliser FAIL PROCESS Forward	FwdFail_Proces	Accumulated volume (with meter in error state) in forward flow direction, at line condition.	3	8003	8012-8015 (4 regs)	ml (=10^{-6} m^3)
Totaliser FAIL PROCESS Reverse	RevFail_Proces	Accumulated volume (with meter in error state) in reverse flow direction, at line condition.	4	8004	8016-8019 (4 regs)	ml (=10^{-6} m^3)

8.10 Holding Registers (read/write): Long long (64-bit integer), address range 8500-8999

Parameter / Variable (short description)	Name	Explanation	Relative address	Absolute address, non-Modicon	Absolute address, Modicon-comp.	Units
Test register	TestRegister uint64	Register reserved for testing communications and protocol handling with this type of register, without affecting the operation of the flow meter.	0	8500	8500-8503 (4 regs)	-