COMMUNICATION PROTOCOL

MODBUS MANUAL

M4M Network analyzers

Table of Contents

1 General

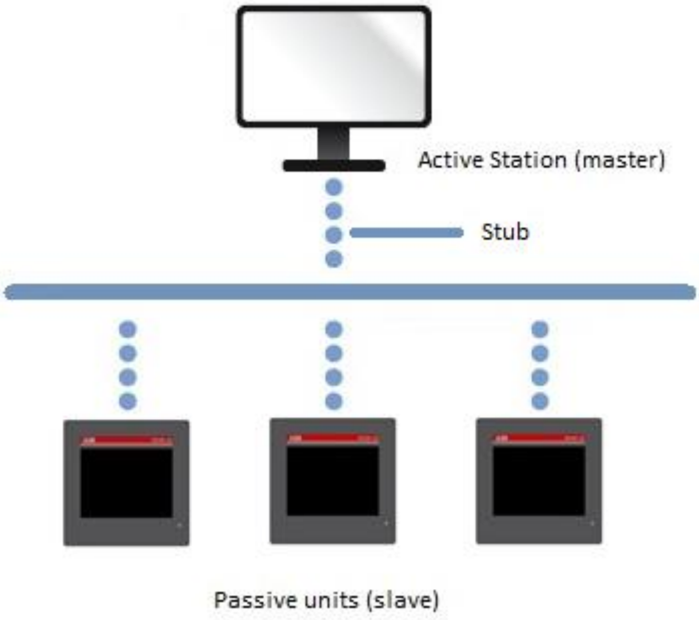
M4M network analyzers offer includes versions with both Modbus RTU and Modbus TCP/IP protocols. The Modbus protocol is specified in its entirety in Modbus Application Protocol Specification available at http://www.modbus.org

1.1 Modbus RTU

Modbus RTU communication in the M4M network analyzers is done on a 3-wire (A, B and Common) polarity dependent bus according to the RS-485 standard. Modbus is a master-slave communication protocol that can support up to 247 slaves (which is the same as the individual device address range in Modbus RTU) organized as a multidrop bus. The communication is half duplex. Services on Modbus are specified by function codes.

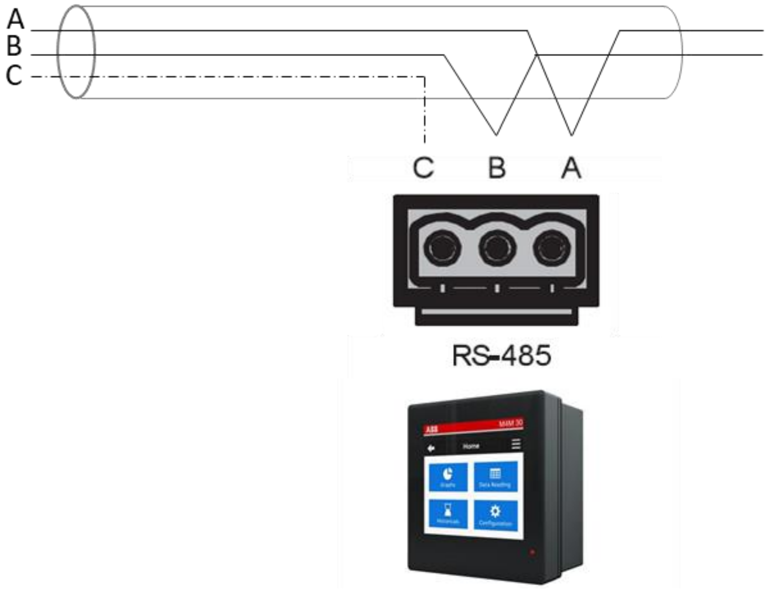
1.1.1 Topology

The RS-485 bus uses line topology, see figure below. Stubs at the meter connections are allowed but should be kept as short as possible and no longer than 1 m. Bus termination in both ends of the line should be used. The resistors should have the same values as the characteristic impedance of the cable which normally is 120 Ohm.


Active Station (master)
Stub
Passive units (slave)

1.1.2 RS-485 wiring on M4M

Each M4M provided with Modbus RTU communication is equipped with RS-485 port. The RS485 terminal is a 3-pole plug contact. A and B are mandatory for the correct communication of the device. C can be connected to the data common ground, if available and needed.

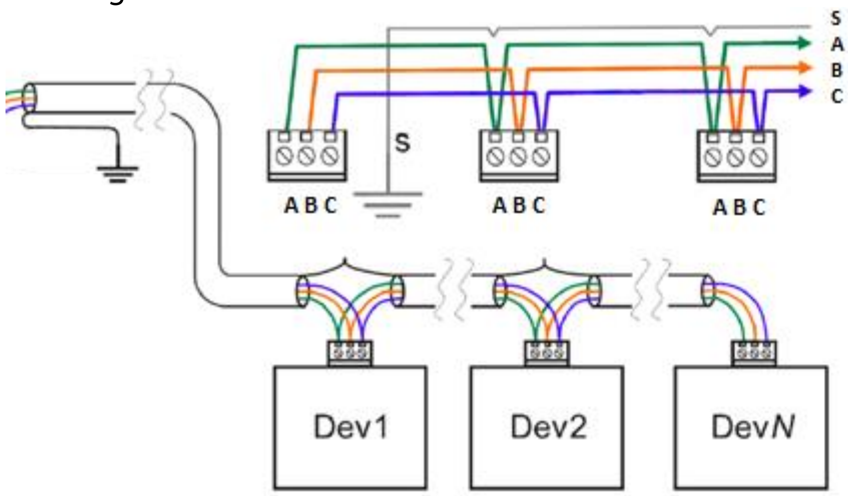


RS485 is a differential signal so no common ground is required.
Given the differential nature of the RS485 signal, the signal can be recovered without any reference to a ground as such - the signal is the difference between the A and B voltages, not the different between one voltage and ground.
The third wire C (Common) helps to ensure that the common mode requirements (-7 V to +12 V) of the transceivers are maintained.

1.1.3 Cable

Cable used is non-shielded or shielded twisted pair cable with wire area of 0.35-1.5 mm2. Maximum length of the bus is 700 m.

The cable recommended in this type of connection has 2 twisted pairs. A pair will be used for “A” and “B”, one of the wires of the second pair will be used as common wire and the fourth wire will be not used.
See image below:



1.1.3.1 Recommended practice

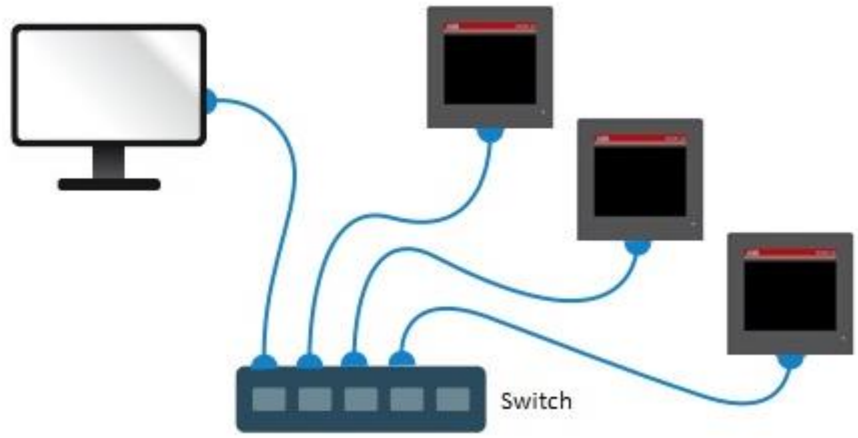
- Good quality shielded twisted pair cable should be used.
- If shielded cable is used the shield should be connected to ground in one end.
- Do not put communications cables and power cables in the same race-ways.
- Route communications cables to avoid potential noise sources such as high-power equipment.
- Ferrite should be used especially when long cables are used.

1.2 Modbus TCP/IP

Modbus TCP/IP is a Modbus variant used for communications over TCP/IP networks, typically connecting over port 502 (default value).

1.2.1 Topology

Hereafter an example of Modbus TCP/IP topology. Please note that M4M 30 Ethernet allows to daisy-chain the Modbus TCP/IP communication through 2 RJ45 ports on the devices.



1.2.2 Cable

Suitable cables for Modbus TCP/IP connection:

CATEGORY	SHIELDING
Category 5	Unshielded
Category 5e	Unshielded
Category 6	Shielded or Unshielded
Category 6a	Shielded
Category 7	Shielded

1.2.2.1 Recommended practice

- If shielded cable is used the shield should be connected to ground in one end.
- Do not put communications cables and power cables in the same raceways.
- Route communications cables to avoid potential noise sources such as high-power equipment.
- Ferrite should be used especially when long cables are used.

2 Supported function codes

The function codes are used to read or write 16 bits registers. All metering data, such as voltage, current power, active energy or firmware version, is represented by one or more such registers. For further information about the relation between register number and metering data, refer to “Mapping Tables”.

The following function codes are supported:

- Function code 3 (Read holding registers)
- Function code 6 (Write single register)
- Function code 16 (Write multiple registers)

3 Modbus frame

A Modbus request frame generally has the following structure:

Slave Address	Function Code	Data	Error Check

Slave address: Modbus slave address, 1 byte.

Function code: Decides the service to be performed.

Data: Dependent on the function code. The length varies.

Error check: CRC, 2 bytes

The network messages can be query-response or broadcast type. The query-response command sends a query from the master to an individual slave and is generally followed by a response.

The broadcast command sends a message to all slaves and is never followed by a response. Broadcast is supported by function code 6 and 16.

<u>Function Code 3 (Read holding registers)</u>

Function code 3 is used to read measurement values or other information from the electricity meter. Multiple Modbus register can be read in one request.

Request frame

Slave Address	Function Code	Address	No. of Register	Error Check

Example of a request

The following is an example of a request (read phase voltage L1).

Slave address	0x01
Function code	0x03
Start address, high byte	0x5B
Start address, low byte	0x02
No. of registers, high byte	0x00
No. of registers, low byte	0x02
Error check (CRC), high byte	0x76
Error check (CRC), low byte	0xEF

Response frame

Slave Address	Function Code	Byte count	Register Values	Error Check

Example of a response

The following is an example of a response.

Slave address	0x01
Function code	0x03
Byte count	0x04
Value of register 0x5B02, high byte	0x00
Value of register 0x5B02, low byte	0x00
Value of register 0x5B03, high byte	.....
Value of register 0x5B03, low byte	…..
Error check (CRC), high byte	0xXX
Error check (CRC), low byte	0xXX

<u>Function Code 16 (Write multiple registers)</u>

Function code 16 is used to modify settings in the meter. It is possible to write up to 123 consecutive registers in a single request. This means that several settings can be modified in a single request.

Request frame

Slave Address	Function Code	Start Address	No. of Register	Byte Count	Register Values	Error Check

Example of a request

The following is an example of a request (set Date/Time to November 11, 2019,12:13:14).

Slave address	0x01
Function code	0x10
Start address, high byte	0x8A
Start address, low byte	0x00
No. of registers, high byte	0x00
No. of registers, low byte	0x03
Byte count	0x06
Value of register 0x8A00, high byte	0x13
Value of register 0x8A00, low byte	0x0B
Value of register 0x8A01, high byte	0x0B
Value of register 0x8A01, low byte	0x0C
Value of register 0x8A02, high byte	0x0D
Value of register 0x8A02, low byte	0x0E
Error check (CRC), high byte	0x8C
Error check (CRC), low byte	0x82

Response frame

Slave Address	Function Code	Start Address	No. of Register	Error Check

Example of a response

The following is an example of a response.

Slave address	0x01
Function code	0x10
Register address, high byte	0x8A
Register address, low byte	0x00
No. of registers, high byte	0x00
No. of registers, low byte	0x03
Error check (CRC), high byte	0xAA
Error check (CRC), low byte	0x10

Function Code 6 (Write single register)

Function code 6 can be used as an alternative to function code 16 if there is only one register to be written.

Request frame

Slave Address	Function Code	Register Address	Register Value	Error Check

Example of a request

The following is an example of a request (reset power fail counter).

Slave address	0x01
Function code	0x06
Register address, high byte	0x8F
Register address, low byte	0x00
No. of registers, high byte	0x00
No. of registers, low byte	0x01
Error check (CRC), high byte	0x62
Error check (CRC), low byte	0xDE

Response frame

Using function code 6, the response frame is an echo of the request frame.

Exception Responses

If an error should occur while processing a request, the power meter gives an exception response that contains an exception code.

An exception frame has the following structure:

Slave Address	Function Code	Exception Code	Error Check

In the exception response the function code is set to the function code of the request plus 0x80.

The exception codes that are used are listed in the following table:

Exception Code	Exception	Definition
01	Illegal Function	A function code that is not supported has been used.
02	Illegal data address	The requested register is outside the allowed range.
03	Illegal data value	The structure of a received message is incorrect.
04	Slave device failure	Processing the request fail due to an internal error in the meter.

Reading and Writing to Registers

Readable register

The readable range in the Modbus mapping are registers 1000-8EFF (hexadecimal). Reading any registers within this range will result in a normal Modbus response. It is possible to read any number of registers between 1 and 125, i.e., it is not necessary to read all registers of a quantity listed on one line in the mapping tables. Any attempt to read outside this range will result in an illegal data address exception (Modbus exception code 2).

Multi-register values

For quantities that are represented as more than 1 register, the most significant byte is found in the high byte of the first (lowest) register. The least significant byte is found in the low byte of the last (highest) register.

Unused register

Unused registers within the mapping range, for example missing quantities in the connected meter, will result in a normal Modbus response but the value of the register will be set to “invalid”. For quantities with data type “unsigned”, the value will be FFFF in all registers. For quantities with data type “signed”, the value is the highest value possible to express. That means that a quantity that is represented by only one register will have the value 7FFF. A quantity that is represented by 2 registers will have the value 7FFFFFFF, and so on.

Writing to register

Writing to registers is only permitted to the registers listed as writable in the mapping tables. Attempting to write to a register that is listed as writable but that is not supported by the meter will not result in an error indication. It is not possible to modify parts of a setting, e.g. to set only the year and month of the Date/time setting.

Confirm set value

After you set a value in the meter, it is recommended that you read the value to confirm the result, since it is not possible to confirm if a write was successful from the Modbus response.

4 Mapping Tables

The purpose of this section is to explain the relation between register number and metering data.

Please visit the ABB Library at this link to get the full Modbus table file.

Quantity	Name of the meter quantity or other information available in the meter.
Unit	Unit for the Quantity (if applicable).
Details	Refinement of the Quantity column.
Resolution	Resolution of the value for this Quantity (if applicable).
Data type	Data type for this Quantity, i.e. how the value in the Modbus registers should be interpreted.
Access	Read / Write access.
Start Reg (Hex)	Hexadecimal number for the first (lowest) Modbus Register for this quantity. It is expressed exactly as it is sent on the bus.
Start Reg (Dec)	Decimal representation of Modbus Register.
Nr of quantities (Dec)	Number of quantities.
Size (Dec)	Size for the meter Quantity.
Nr of registers (Dec)	Number of Modbus registers for the meter Quantities. A Modbus Register is 16 bits long.
Product type	Product for which the quantity is available.
Functional block	Functionality to which the quantity belongs.

5 Historicals

Readout of all types of historical values is made by writing to a group of registers called Header and reading from one or more groups of registers called Data blocks.

In the Modbus mapping all historical data are organized as entries. This concerns Energy Snapshots, Energy Trend, Max/Min Demand, Load profile functionalities.

Entry number 1 is the most recent entry, entry number 2 is the second most recent, and so on. Entry number 0 is not used.

The Header is used for controlling readout with respect to date/time or entry numbers, and for loading new entries into the Data blocks. The data blocks contain the actual data, for example energy values.

When there are no more entries to read all registers in the Data blocks are set to 0xFFFF.

Header register

There are number of standard commands that are used in the same way when reading out any type of historical data. These are represented by registers in the Header, separately mapped for each functionality, but with the same names.

The following table describes the common header registers:

Function	size	Description	Data type	Read/Write
Get next entry	1	Write the value 1 to this register to new values in the Data block(s)	Unsigned	R/W
Entry number	1	Write to this register to choose an entry number to start reading from	Unsigned	R/W
Date/Time	3	Write to this register to choose a date/time to start reading from	Date/Time (see below)	R/W
Direction	1	Write to this register to choose the direction of reading	Unsigned	R/W

Get next entry register:

The Get next entry register is used to continue an ongoing readout, which was started by writing to any of the Entry number, Date/Time or Direction registers.

If the direction in Direction register is set to backward the Data block is loaded with older data; correspondingly, if the direction is set to forward the Data block is loaded with more recent data.

Entry number register:

The Entry number register is used to specify an entry number to start reading from. When a value is written to the Entry number register the Data block is loaded with values for that entry number.

Subsequent writes to Get next entry register will update the Entry number register (increment or decrement depending on direction in the Direction register), as well as loading new values to the Data block.

The default value of Entry number register after a restart is 0.

Date/Time register:

The Date/Time register is used to specify a date and time to start reading from. When a value is written to the Date/Time register the Data block is loaded with values for that date and time. The Entry number register is also automatically updated, to reflect which entry number the values for this date and time has.

If there is no entry for the date and time chosen, and the reading direction is set to backward, the nearest older entry will be loaded into the Data block. If the reading direction is instead forward, the nearest newer entry will be loaded.

Subsequent writes to Get next entry register will load new data into the Data block, in the order indicated by the Direction register. The Entry number register will also be automatically updated (incremented or decremented depending on the direction in the Direction register).

Direction register:

The Direction register is used to control the direction in time in which the entries are read. Possible values are shown in the table below:

Value	**Description**
0	Backwards, i.e. from recent entries towards older entries
1	Forward, i.e. from old entries towards recent entries

The default value of Entry number register after a restart is 0, i.e. backwards.

Data block register

There are number of standard data items that are used in the same way when reading out any type of historical data. These are represented by registers in the Data block, separately mapped for each functionality, but with the same names.

The following table describes the common Data block registers:

Function	size	Description	Data type	Read/Write
Timestamp	3	The date and time on which the value was stored	Date/Time	R/W
Quantity	3	OBIS code for the quantity concerned	6 bytes sequence	R/W
Data type	1	Data type for the value of the quantity concerned	Unsigned	R/W
Scaler	1	Scaling of the value for the quantity concerned	Signed	R/W

<u>Timestamp:</u>

The same date and time format are used wherever a date and time occurs in the registers, e.g. the Date/Time register in the Header or a timestamp in the Data block.

The following table shows the structure of date and time in the mapping:

Byte number	Description	Details
0	Year	Most significant byte of lowest register
1	Month	Least significant byte of lowest register
2	Day	....
3	Hour	....
4	Minute	....
5	Second	Least significant byte of highest register

Quantity register:

The OBIS code for a quantity in a channel. List of OBIS codes is present in the Modbus table file. The table below shows an example of how an OBIS code is mapped to the Quantity registers. The OBIS code used is for active energy import total: 1.0.1.8.0.255.

Byte number	Details	Value (for active energy import total)
0	Most significant byte of lowest register	1
1	Least significant byte of lowest register	0
2	....	1
3	....	8
4	....	0
5	Least significant byte of highest register	255

Data type register:

The data type register contains a data type identifier. The identifier for 64-bit unsigned integer is 21 and the identifier for 64-bit signed integer is 20.

The following table shows the available data type:

Data type	Decimal value	Hexadecimal value
Int64	20	0x14
UInt64	21	0x15

Scaler register:

The scaler register shows the resolution of the value. The measured value in the Value register should be interpreted as: value*10scaler. For example, the prefix “kilo” is represented by scaler 3 while “milli” is -3. An energy accumulator with the resolution 0,01 kWh consequently has scaler 1.

Response times

The Headers for reading out historical values include one or more of the registers Entry number, Date/Time, Direction and Get next entry for controlling the readout.

When writing to any of the registers Entry number, Date/Time or Direction a new search is started in the persistent storage, which can take a long time depending on how old the entry searched for is. The response from Modbus is given after the search is finished, i.e. when the requested entry has been found.

Recent entries are found fast, whereas finding the oldest can take seconds or even up to about a minute if there are many thousands of newer values. It is therefore preferable to start reading from a recent entry number or date/time and then go backwards in time.

Writing to the Get next entry register continues the ongoing search and consequently goes fast.

6 Energy Snapshots, Energy Trend

At the end of a defined period, up to 20 configurable channels, which can contain energy register values, input energy counter values and currency/CO2 values, are stored together with the time/date for the end of the period.

Each channel can store up to 730 periods.

The period length can be 1 hour, 6 hours, 12hours, a day, a week or a month. Changing time/date into another period than the pending period will store the current period and start a new one. If a power fail occurs that lasts over the end of an ongoing period, the period will be stored when the meter powers up again and a new period will start. If the meter has lost time and date/time is not set when the meter powers up again, Energy Snapshots and Energy Trend will enter a waiting state until time/date is set.

The period date and time is stored as end of period. For instance, if a period starts 2019.01.01 00:00.00 and ends 2019.01.02 00:00.00, then the stored period will be 2019.01.02 00:00.00.

Memory works with a FIFO logic.

It is possible to configure and read Energy Snapshots and Energy Trend via Modbus communication.

Mapping Table - Energy Snapshots:

Function	**Details**	**Start Reg (Hex)**	**Size**
Energy Snapshots	Header	8000	16
Energy Snapshots	Data block 1	8010	83
Energy Snapshots	Data block 2	8070	83
Energy Snapshots	Data block 3	80D0	83
Energy Snapshots	Data block 4	8130	83
Energy Snapshots	Data block 5	8190	83
Energy Snapshots	Data block 6	81F0	83
Energy Snapshots	Data block 7	8250	83

Header for Energy Snapshots registers:

The following table describes the Energy Snapshots headers:

Function	Start Reg (Hex)	Size	Description	Read/Write
Get next entry	8000	1	Write value 1 to this register to load the next block of values and timestamp.	R/W
Entry number	8001	1	Write to this register to choose an entry number to start reading from	R/W
Date/Time	8004	3	Write to this register to choose a date/time to start reading from	R/W
Direction	8007	1	Write to this register to choose the direction of reading	R/W

Mapping Table – Energy Trend:

Function	Details	Start Reg (Hex)	Size
Energy Trend	Header	8300	16
Energy Trend	Data block 1	8310	83
Energy Trend	Data block 2	8370	83
Energy Trend	Data block 3	83D0	83
Energy Trend	Data block 4	8430	83
Energy Trend	Data block 5	8490	83
Energy Trend	Data block 6	84F0	83
Energy Trend	Data block 7	8550	83

Header for Energy Trend registers:

The following table describes the Header about Energy Trend:

Function	Start Reg (Hex)	Size	Description	Read/Write
Get next entry	8300	1	Write value 1 to this register to load the next block of values and timestamp.	R/W
Entry number	8301	1	Write to this register to choose an entry number to start reading from	R/W
Date/Time	8304	3	Write to this register to choose a date/time to start reading from	R/W
Direction	8307	1	Write to this register to choose the direction of reading	R/W

The Data blocks contain the history of Energy Snapshots / Energy Trend. Data block 1 to 7 have the same structure. Each block can contain up to 8 channels. Consequently, in a meter with 20 previous values channels, there are 8 channels in block 1 and block 2 and 4 channels in block 3.

The registers of unused channels are filled with 0xFFFF.

Structure of the Data blocks:

The following table describes the structure of the Energy Snapshots Data blocks. This structure is used also for Trend functionality, with different registers.

Channel	**Contents**	**Start Reg (Hex)**	**Size**	**Description**
Common for all channels	Timestamp	8010	3	Date and time for the end of period
Channel 1	Quantity	8013	3	OBIS code for the quantity stored in channel 1.
Channel 1	Data type	8016	1	Data type for quantity stored in channel 1.
Channel 1	Scaler	8017	1	Scaler for quantity stored in channel 1.
Channel 1	Status	8018	1	Status for quantity stored in channel 1.
Channel 1	Value	8019	4	Value for quantity stored in channel 1.
…				
…				
Channel 8	Quantity	8059	3	OBIS code for the quantity stored in channel 8.
Channel 8	Data type	805C	1	Data type for quantity stored in channel 8.
Channel 8	Scaler	805D	1	Scaler for quantity stored in channel 8.
Channel 8	Status	805E	1	Status for quantity stored in channel 8.
Channel 8	Value	805F	4	Value for quantity stored in channel 8.

Status register:

The status register shows the status for a value stored at a given timestamp. Possible values are shown in the table below:

Status	**Description**
0	OK
1	Not available
2	Data error

6.1 Reading Energy Snapshots and Energy Trend

Readout of Energy Snapshots and Energy Trend is controlled by the Entry number register or Date/ Time register.

After writing to any of those registers, the values of all channels for the given entry number or date/time are available in the registers of data block 1 to 7, together with status and timestamp information.

In the data blocks, the registers Quantity, Data type and Scaler provide further information about the data stored in each channel. To get the next block of previous values, write the value 1 to the Get next entry register, and then read again from the registers in the data blocks.

Read the most recent values

Follow the steps in the table below to read the most recent Energy Snapshots / Energy Trend entry:

Step	Action
1	Write the value 1 to the entry number register.
2	Read the data blocks (from 1 to 7 or data blocks of interest).

Example of Energy Snapshots reading (Hex Format)

01 10 80 01 00 01 02 00 01 E7 89 (Request: write Entry number register)

01 10 80 01 00 01 79 C9 (Response: write Entry number register)

01 03 80 10 00 03 2D CE (Request: read Energy Snapshots Data Block 1, Timestamp)

01 03 06 0A 01 01 03 01 01 2D B3 (Response: Timestamp Data Block 1 – 01/01/2010, 03:01:01)

01 03 80 13 00 50 9D F3 (Request: read Energy Snapshots Data Block 1, Data)

01 03 A0 01 00 01 08 00 FF 00 14 00 01 00 00 00 00 00 00 00 00 00 00 ……..(Response: Energy Snapshots Data Block 1 and channel 1: Active Energy Import Total with value of ‘0’)

Read the entire history

Follow the steps in the table below to read the entire history of Energy Snapshots / Energy Trend:

Step	Action
1	Write the value 0 to the Entry number register to make sure the reading starts from the most recent entry.
2	Write the value 1 to the Get next entry register.
3	Read the data blocks (from 1 to 7 or data blocks of interest).
4	Repeat steps 2 and 3 until there are no more entries stored. When all entries have been read, all registers (Timestamp included) in the data blocks are set to 0xFFFF.

Example of Energy Snapshots reading (Hex Format)

01 10 80 01 00 01 02 00 00 26 49 (Request: write Entry number register)

01 10 80 01 00 01 79 C9 (Response: write Entry number register)

01 10 80 00 00 01 02 00 01 E6 58 (Request: write Get next entry number register)

01 10 80 00 00 01 28 09 (Response: write Get next entry number register)

01 03 80 10 00 03 2D CE (Request: read Energy Snapshots Data Block 1, Timestamp)

01 03 06 0A 01 01 03 01 01 2D B3 (Response: Timestamp Data Block 1 – 01/01/2010, 03:01:01)

01 03 80 13 00 50 9D F3 (Request: read Energy Snapshots Data Block 1, Data)

01 03 A0 01 00 01 08 00 FF 00 14 00 01 00 00 00 00 00 00 00 00 00 00 ……..(Response: Energy Snapshots Data Block 1 and channel 1: Active Energy Import Total with value of ‘0’)

Read forward or backwards from a specified date and time

Follow the steps in the table below to read forward or backwards in time from a specified date/time:

Step	Action
1	Write a date and time to the Date/Time registers.
2	Write to the Direction register. Writing value 0 means backwards and value 1 means forward.
3	Read the data blocks (from 1 to 7 or data blocks of interest).
4	Write the value 1 to the Get next entry register.
5	Repeat steps 3 and 4 until there are no more entries stored. When all entries have been read, all registers in the data blocks are set to 0xFFFF.

Example of Energy Snapshots reading (Hex Format)

01 10 80 04 00 03 06 09 04 13 0A 1F 35 13 6F (Request: write Date/Time register, 09-04-2019 10:31:53)

01 10 80 04 00 03 E8 09 (Response: write Date/Time register)

01 10 80 07 00 01 02 00 00 26 2F (Request: write Direction register, backward reading)

01 10 80 07 00 01 99 C8 (Response: write Direction register, backward reading)

01 03 80 10 00 03 2D CE (Request: read Energy Snapshots Data Block 1, Timestamp)

01 03 06 0A 01 01 03 01 01 2D B3 (Response: Timestamp Data Block 1 – 01/01/2010, 03:01:01)

01 03 80 13 00 50 9D F3 (Request: read Energy Snapshots Data Block 1, Data)

01 03 A0 01 00 01 08 00 FF 00 14 00 01 00 00 00 00 00 00 00 00 00 00 ……...(Response: Energy Snapshots Data Block 1 and channel 1: Active Energy Import Total with value of ‘0’)

01 10 80 00 00 01 02 00 01 E6 58 (Request: write Get next entry register, value 1)

01 10 80 00 00 01 28 09 (Response: Get next entry register, value 1)

6.2 Energy Snapshots and Energy Trend configuration

Energy Snapshots and Energy Trend configuration defines the set of quantities to store at the end of a period. It is also defining the period with which values are stored.

The following table show an overview of the mapping table:

Quantity	Details	Start Reg (Hex)	Size
Energy Snapshots / Energy Trend	Quantity configuration	8C50	5
Energy Snapshots / Energy Trend	Period configuration	8C55	1

The following table describes the group of registers for configuring quantities to store in Energy Snapshots and Energy Trend:

Quantity	Start Reg (Hex)	Size	Description	Read / Write
Number of channels	8C50	1	The number of channel used (up to a max of 20)	R / W
Channel number	8C51	1	Current channel number during read or write of configuration	R
Quantity	8C52	3	OBIS code for the quantity in this channel	R / W

Follow the steps in the table below to configure the set of quantities to store in Energy Snapshots and Trend:

Step	Action
1	Write the number of channels that shall be configured to the Number of channels register. This is a value between 1 and 20.
2	Write the OBIS code for the quantity to store in the first channel to the Quantity registers.
3	Repeat step 2 for all channels that shall be used, i.e. the same number of times as the value written in step 1.

Example (Hex Format)

01 10 8C 50 00 01 02 00 14 E7 C7 (Request: write Number of channels register)

01 10 8C 50 00 01 2B 48 (Response: write Number of channels register)

01 03 8C 51 00 01 FF 4B (Request: select Channel number)

01 03 02 00 01 79 84 (Response: Current channel number, channel 1)

01 10 8C 52 00 03 06 01 00 01 08 00 FF 13 36 (Request: write OBIS code for the Quantity register

01 10 8C 52 00 03 0B 49 (Response: Quantity register writing)

01 03 8C 51 00 01 FF 4B (Request: select Channel number)

01 03 02 00 02 39 85 (Response: Current channel number channel 2)

01 10 8C 52 00 03 06 01 00 02 08 00 FF 13 72 (Request: write OBIS code for the Quantity register)

01 10 8C 52 00 03 0B 49 (Response: Quantity register writing)

and so on.

01 10 8C 55 00 01 02 00 00 E7 9D (Request: write Period configuration register, period Day)

01 10 8C 55 00 01 3B 49 (Response: Period configuration register)

Follow the steps in the table below to read the current configuration of quantities to store in Energy Snapshots and Trend:

Step	Action
1	Read the Number of channels register to find out how many channels are used.
2	Read from the Quantity registers to get the OBIS code for the quantity configured in the first channel.
3	Repeat step 2 for each channel, until all OBIS codes have been read. This means step 2 shall be performed the same number of times as the value read from the Number of channels register

Example (Hex Format)

01 03 8C 50 00 01 AE 8B (Request: read Number of channels register)

01 03 02 00 14 B8 4B (Response: Number of channels configured, 20 channels)

01 03 8C 51 00 04 3F 48 (Request: get OBIS code for quantity of the channel 1)

01 03 08 00 01 01 00 01 08 00 FF 44 B8 (Response: OBIS code of the quantity into channel 1)

01 03 8C 51 00 04 3F 48 (Request: get OBIS code for quantity of the channel 2)

01 03 08 00 02 01 00 01 08 00 FF 77 B8 (Response: OBIS code of the quantity into channel 2)

01 03 8C 51 00 04 3F 48 (Request: get OBIS code for quantity of the channel 3)

01 03 08 00 03 01 00 01 08 00 FF 67 78 (Response: OBIS code of the quantity into channel 3)

and so on.

Note – Step 1 initiates the readout procedure and can NOT be left out, even if the number of channels used is already known.

Note – The Channel number register can optionally be read together with the Quantity registers in step 2. The Channel number register holds the current channel number, starting from 1 after reading the Number of channels register. It is incremented every time the Quantity registers are read.

The Period configuration register is used to read or write the period with which Energy Snapshots and Energy Trend are stored. The table below describes the contents of the Period configuration register:

Byte number	Description	Possible values
0 (High byte)	Configured period	0 = Day 1 = Week 2 = Month 3 = 12 Hours 4 = 6 Hours 5 = 1 Hour
1 (Low byte)	Day of week, in case of weekly storage	1-7 (1 = Monday)

Example (Hex Format)

01 03 8C 55 00 01 BE 8A (Request: read Period configuration register)

01 03 02 00 FF F8 04 (Response: actual Period configured)

7 Max / Min Demand

Max/Min Demand functionality defines the set of max/min values to store at the end of a period and the number of levels for these quantities. It is also defining the period with which values are stored, and the intervals for calculation of minimum and maximum values. Memory works with a FIFO logic.

It is possible to configure and read Max/Min Demand via Modbus.

Mapping Table – Max / Min Demand:

Function	Details	Start Reg (Hex)	Size
Max / Min Demand	Header	8F72	16
Max / Min Demand	Data block 1	8F82	115
Max / Min Demand	Data block 2	9002	115
Max / Min Demand	Data block 3	9082	115
Max / Min Demand	Data block 4	9102	115
Max / Min Demand	Data block 5	9182	115
Max / Min Demand	Data block 6	9202	115
Max / Min Demand	Data block 7	9282	115
Max / Min Demand	Data block 8	9302	115
Max / Min Demand	Data block 9	9382	115
Max / Min Demand	Data block 10	9402	115
Max / Min Demand	Data block 11	9482	115
Max / Min Demand	Data block 12	9502	115
Max / Min Demand	Data block 13	9582	115
Max / Min Demand	Data block 14	9602	115
Max / Min Demand	Data block 15	9682	115
Max / Min Demand	Data block 16	9702	115
Max / Min Demand	Data block 17	9782	115
Max / Min Demand	Data block 18	9802	115
Max / Min Demand	Data block 19	9882	115

Header for Max / Min Demand registers:

Function	Start Reg (Hex)	Size	Description	Read/Write
Get next entry	8F72	1	Write value 1 to this register to load the next block of values and timestamp.	R/W
Entry number	8F73	1	Write to this register to choose an entry number to start reading from	R/W
Date/Time	8F76	3	Write to this register to choose a date/time to start reading from	R/W
Direction	8F79	1	Write to this register to choose the direction of reading	R/W

Data block registers:

The Data blocks contain the history of max/min demand values. Data block 1 to 19 have the same structure. Each block can contain up to 8 channels. Consequently, in a meter with 150 demand channels (25 channels, for each up to 3 max and 3 min), there are 8 channels in each of block 1 to block 18 and 6 channels in block 19. The registers of unused channels are filled with 0xFFFF.

Structure of the Data blocks:

The following table describes the structure of the Max/Min Demand blocks.

Channel	**Contents**	**Start Reg (Hex)**	**Size**	**Description**
Common for all channels	Timestamp	8F82	3	Date and time for the end if this period, i.e. when this entry was stored. (Date/Time format)
Channel 1	Quantity	8F85	3	OBIS code for the quantity monitored in channel 1.
Channel 1	Level	8F88	1	Demand level for channel 1.
Channel 1	Data type	8F89	1	Data type for quantity monitored in channel 1.
Channel 1	Scaler	8F8A	1	Scaler for quantity monitored in channel 1.
Channel 1	Capture time	8F8B	3	Date and time when the minimum or maximum occurred for the quantity monitored in channel 1.
Channel 1	Status	8F8E	1	Status for quantity monitored in channel 1.
Channel 1	Value	8F8F	4	Value for quantity monitored in channel 1.
….				
….				
Channel 8	Quantity	8FDE	3	OBIS code for the quantity monitored in channel 8.
Channel 8	Level	8FE1	1	Demand level for channel 8.
Channel 8	Data type	8FE2	1	Data type for quantity monitored in channel 8.
Channel 8	Scaler	8FE3	1	Scaler for quantity monitored in channel 8.
Channel 8	Capture time	8FE4	3	Date and time when the minimum or maximum occurred for the quantity monitored in channel 8.
Channel 8	Status	8FE7	1	Status for quantity monitored in channel 8.
Channel 8	Value	8FE8	4	Value for quantity monitored in channel 8.

Level register:

The Level register shows which demand level is configured for this channel. Possible values are shown in the table below:

Value	Description
1	Highest/Lowest value during the demand period
2	Second highest/lowest value during the demand period
3	Third highest/lowest value during the demand period

Capture time register:

The Capture time register shows the date and time when the minimum or maximum value for this entry occurred.

Status register:

The status register shows the status for a value stored at a given timestamp. Possible values are shown in the table below:

Status	Description
0	OK
1	Not available
2	Data error

7.1 Reading Max / Min Demand

Readout of max/min demand is controlled by the Entry number register or Date/Time register. Entry n. 0 is used for current demand, that is the pending period, and entry n. equal or bigger than 1 are used for historic demand periods.

After writing to any of those registers, the values of all channels for the given entry number or date/time are available in the registers of data block 1 to 19, together with status and timestamp information.

In the data blocks, the registers Quantity, Level, Data type and Scaler provide further information about the data stored in each channel. To get the next block of demand values, write the value 1 to the Get next entry register, and then read again from the registers in the data blocks.

Read the most recent historic entry

Follow the steps in the table below to read the most recent entry:

Step	Action
1	Write the value 1 to the entry number register.
2	Read the data blocks (from 1 to 19 or data blocks of interest).

Read part of or the entire demand

Follow the steps in the table below to read part of or the entire demand:

Step	Action
1	Write the value for the starting entry number. Entry number 0 makes the reading to start with current demand and 1 makes the reading to start with most recent historic entry.
2	Read the data blocks of interest.
3	Write the value 1 to the Get next entry register.
4	Repeat steps 2 and 3 as many times as required or until there are no more entries stored. When all entries have been read, all registers in the data blocks are set to 0xFFFF.

Read forward or backwards from a specified date/ time

Follow the steps in the table below to read forward or backwards in time from a specified date/time:

Step	Action
1	Write a date and time to the Date/Time registers.
2	Write to the Direction register. Writing value 0 means backwards and value 1 means forward.
3	Read the data blocks of interest.
4	Write the value 1 to the Get next entry register.
5	Repeat steps 3 and 4 until there are no more entries stored. When all entries have been read, all registers in the data blocks are set to 0xFFFF.

7.2 Max / Min Demand configuration

Max/min demand configuration defines the set of quantities to store at the end of a period and the number of levels for these quantities. It is also defining the period with which values are stored, and the intervals for calculation of minimum and maximum values.

Mapping table:

The following table shows an overview of the mapping table:

Quantity	**Details**	**Start Reg (Hex)**	**Size**
Max / Min Demand	Quantity configuration	8C30	5
Max / Min Demand	Level configuration	8C35	4
Max / Min Demand	Interval configuration	8C39	1
Max / Min Demand	Sub interval configuration	8C3A	1
Max / Min Demand	Period configuration	8C3B	1

Quantity configuration registers:

The following table describes the group of registers for configuring quantities to store in demand:

Function	**Start Reg (Hex)**	**Size**	**Description**	**Read / Write**
Number of quantities	8C30	1	The number of quantities to store in Demand (Minimum 50, maximum 150)	R / W
Quantity number	8C31	1	Current quantity number during read or write of configuration	R
Quantity	8C32	3	OBIS code for the quantity	R / W

Follow the steps in the table below to configure the set of quantities to store in Demand:

Step	**Action**
1	Write the number of quantities that shall be configured to the Number of quantities register. Minimum 50, maximum 150.
2	Write the OBIS code for the first quantity to the Quantity registers.
3	Repeat step 2 for all quantities that shall be used, i.e. the same number of times as the value written in step 1.

Follow the steps in the table below to read the current configuration of quantities stored in Max/min demand:

Step	Action
1	Read the Number of quantities register to find out how many quantities are used.
2	Read from the Quantity registers to get the OBIS code for the first quantity.
3	Repeat step 2 for each quantity, until all OBIS codes have been read. This means step 2 shall be performed the same number of times as the value read from the Number of quantities register.

Note – Step 1 initiates the readout procedure and can NOT be left out, even if the number of quantities used is already known.

Note – The Quantity number register can optionally be read together with the Quantity registers in step 2. The Quantity number register holds the current quantity number, starting from 1 after reading the Number of quantities register. It is incremented every time the Quantity registers are read.

<u>Max/Min level configuration registers:</u>

The following table describes the group of registers for configuring the number of levels for all quantities stored in demand:

Function	Start Reg (Hex)	Size	Description	Read / Write
Level quantity	8C35	3	OBIS code for the quantity	R/W
Number of levels	8C38	1	Number of levels to store for the quantity	R/W

Follow the steps in the table below to configure the number of levels for each of the quantities stored in demand:

Step	Action
1	Write the OBIS code for the first quantity to the Level quantity registers.
2	Write the number of levels to use for the quantity chosen in step 1 to the Number of levels register. Allowed values are 1-3.
3	Repeat step 1 and 2 for all quantities used in demand.

Follow the steps in the table below to read the current configuration of levels for all quantities stored in demand:

Step	Action
1	Write the OBIS code for the first quantity to the Level quantity registers.
2	Read the number of levels used for the quantity chosen in step 1 from the Number of levels register.
3	Repeat step 1 and 2 for all quantities used in demand.

Interval configuration register:

The Interval configuration register is used to read or write the length of the period with which average values are calculated. The interval is expressed in minutes.

Sub interval configuration register:

The Sub interval configuration register is used to read or write the length of the short period in case of sliding demand. The sub interval is expressed in minutes.

Function	Start Reg (Hex)	Size	Read / Write
Interval	8C39	1	R/W
Sub interval	8C3A	1	R/W
Period	8C3B	1	R/W

Period configuration register:

The Period configuration register is used to read or write the period with which demand values are stored. The table below describes the contents of the Period configuration register:

Byte number	Description	Possible values
0 (High byte)	Demand period	0 = Daily 1 = Weekly 2 = Monthly
1 (Low byte)	Day of week, in case of weekly storage	1-7 (1 = Monday)

8 Load Profile

Load profile configuration defines the quantity to store for each channel. It is also defining the interval by which values are stored and the maximum number of snapshots. All settings are individual for every channel. If there is no free memory space available, the oldest period will be erased to make room for the most recent one.

It is possible to configure and read Load Profile via Modbus communication.

Mapping Table – Load Profile

Function	Details	Start Reg (Hex)	Size
Load Profile	Header	8700	16
Load Profile	Channel information	8710	7
Load Profile	Data block	8720	120

Header for Load profile registers:

The following table describes Load Profile header registers:

Function	Start Reg (Hex)	Size	Description	Read/Write
Get next block	8700	1	Write value 1 to this register to load the next block of load profile entries.	R/W
Channel number	8703	1	Write to this register to choose a load profile channel. Possible values are 1-25.	R/W
Date/Time	8704	3	Write to this register to choose a date/time to start reading from.	R/W
Direction	8707	1	Write to this register to choose the direction of reading.	R/W

Channel information registers:

The following table describes the channel information registers:

Function	Start Reg (Hex)	Size	Description	Read/Write
Quantity	8710	3	OBIS code for the quantity stored in this channel	R/W
Scaler	8713	1	Scaling of the values stored in this channel	R/W
Interval	8714	2	Interval with which values are stored in this channel. Expressed in minutes.	R/W
Data type	8716	1	Data type of the values stored in this channel	R/W

Data block registers:

The data block contains the load profile entries, consisting of timestamp, status and value. There is space for up to 15 entries in the data block. The load profile is read by repeatedly loading new values into the data block in backward or forward direction in time. In case of backwards reading the entries in the data block are placed in ascending entry number order, i.e. going towards older entries. In case of forward reading the entries are placed in descending entry number order, i.e. going towards more recent entries.

Structure of the Data blocks:

Entry	Contents	Start Reg (Hex)	Size	Description
Channel 1	Timestamp	8720	3	Date and time when the entry was stored (Date/Time format)
Channel 1	Status	8723	1	The status for this entry
Channel 1	Value	8724	4	The value for this entry
....				
....				
Channel 15	Timestamp	8789	3	Date and time when the entry was stored. (Date/Time format)
Channel 15	Status	8792	1	The status for this entry
Channel 15	Value	8793	4	The value for this entry

Status registers:

The status register holds status information for a load profile entry.

The following table describes the meaning of the individual bits in the status register:

Status bit	Contents	Description
0	Entry available	This bit is set if the value register contains a valid value
1	Restart	This bit is set if a restart occurred during the interval
2	Interval long	This bit is set if the interval was longer than the configured interval. This happens if the date and time have been adjusted backwards in time
3	Interval short	This bit is set if the interval was shorter than the configured interval. This happens if the date and time have been adjusted forward in time
4	Time change	This bit is set if an adjustment to the date and time was made during the interval
5	Bad value	This bit is set if the value register contains a doubtful value
6-7	Not used	

8.1 Reading Load Profile

Readout of load profile is controlled by the Date/Time register. After writing to the Date/Time register, the load profile entries are available in the registers of the data block. To get the next set of entries the Get next entry register is used.

Follow the steps in the table below to read the 15 most recent load profile entries:

Step	Description
1	Write a date and time in the future to the Date/Time registers, e.g. 2099-01-01 00:00:00.
2	Write the value 0 to the Direction register.
3	Read the data block.

Follow the steps in the table below to read forward or backwards in time from a specified date/time:

Status bit	Description
1	Write a date and time to the Date/Time registers.
2	Write to the Direction register. Writing value 0 means backwards and value 1 means forward.
3	Read data block.
4	Write the value 1 to the Get next entry register.
5	Repeat steps 3 and 4 until there are no more entries stored. When all entries have been read, all registers in the data block are set to 0xFFFF.

8.2 Load Profile configuration

Load profile configuration defines the quantity to store for each channel. It is also defining the interval by which values are stored and the maximum number of snapshots. All settings are individual for every channel.

The following table shows the registers used for load profile configuration:

Quantity	Details	Start Reg (Hex)	Size	Description
Load profile	Channel number	8C20	1	The channel you want to configure
Load profile	Quantity	8C21	3	OBIS code for the quantity
Load profile	Interval	8C24	2	Interval value in minutes
Load profile	Max number of snapshots	8C26	2	Maximum amount of Ch. snapshots

Follow the steps in the table below to configure all load profile channels:

Step	Description
1	Choose the channel to configure by writing a number to the Channel number register. Allowed values are 1-25.
2	Write the OBIS code for the quantity to store in the chosen channel to the Quantity registers.
3	Write the desired storing interval to the Interval registers. The interval is expressed in minutes.
4	Write the desired maximum number of snapshots to the Max number of snapshots registers.
5	Repeat steps 1 to 4 for all channels.

Follow the steps in the table below to read the current configuration of the load profile channels:

Step	Description
1	Choose the channel to read configuration for by writing a number to the Channel number register. Allowed values are 1-25.
2	Read from the Quantity registers to get the OBIS code for the quantity configured in the chosen channel.
3	Read from the Interval registers to get the storing interval for the chosen channel. The interval is expressed in minutes.
4	Read from the Max number of snapshots registers to get the maximum number of snapshots that can be stored in the chosen channel.
5	Repeat steps 1 to 4 for all channels.

9 Notifications

9.1 Errors Logs, Alarms Logs and Warnings Logs

Notifications are divided into Alarms, Warnings and Errors. Each notification type has a header and a data Block, according to the table below.

Mapping Table – Notifications

Log type	Details	Start Reg (Hex)	Size
Errors	Header	6500	16
Errors	Data Block	6510	105
Alarms	Header	65B0	16
Alarms	Data Block	65C0	105
Warnings	Header	6710	16
Warnings	Data Block	6720	105

Header for Notifications registers:
The Header is used for controlling the readout and populate the Data Block. The Data Block contains the actual data and it's initialized with all registers to 0xFFFF.

Errors Header:

Function	Start Reg (Hex)	Size	Description	Read/Write
Get next	6500	1	Write value 1 to this register to load the next block of audit log entries.	W
Entry number	6501	1	Write to this register to choose an entry number to start reading from.	R/W
Direction	6507	1	Write to this register to choose the direction of reading. 1- newer to older blocks 0- older to newer blocks	R/W

Alarms Header:

Function	Start Reg (Hex)	Size	Description	Read/Write
Get next	65B0	1	Write value 1 to this register to load the next block of audit log entries.	W
Entry number	65B1	1	Write to this register to choose an entry number to start reading from.	R/W
Direction	65B7	1	Write to this register to choose the direction of reading. 1- newer to older blocks 0- older to newer blocks	R/W

Warnings Header:

Function	Start Reg (Hex)	Size	Description	Read/Write
Get next	6710	1	Write value 1 to this register to load the next block of audit log entries.	W
Entry number	6711	1	Write to this register to choose an entry number to start reading from.	R/W
Direction	6517	1	Write to this register to choose the direction of reading. 1- newer to older blocks 0- older to newer blocks	R/W

Data block registers:

The Data Block contains 15 entries. Each entry contains:

Contents	Size	Description	Data type
Time-stamp	3	The date and time on which the value was stored in format 0xYYMMDDHHMMSS	Timestamp
Category	1	Possible values for the category register are: ➢ 2 - Error ➢ 4 - Warning ➢ 8 - Alarm	Unsigned
Event ID	1	Contanis a code related to the triggered alarm number or to the error or warning (see related table)	Unsigned
Duration	2	Numer of second related to the duration of the event	Unsigned

The Data blocks structure is the same for Errors Log, Alarms Log and Warnings Log; there follows an example for the Alarms Log data block:

Entry	Contents	Start Reg (Hex)	Size	Description
Entry 1	Timestamp	65C0	3	The date and time on which the value was stored in format 0xYYMMDDHHMMSS
Entry 1	Category	65C3	1	Fixed value (8- Alarm)
Entry 1	Event ID	65C4	1	Contains a code related to the triggered alarm number or to the error or warning (see related table)
Entry 1	Duration	65C5	2	Number of second related to the duration of the event
…..				
Entry 15	Timestamp	65C7	3	The date and time on which the value was stored in format 0xYYMMDDHHMMSS
Entry 15	Category	65CA	1	Fixed value (8- Alarm)
Entry 15	Event ID	65CB	1	Contains a code related to the triggered alarm number or to the error or warning (see related table)
Entry 15	Duration	65CC	2	Number of second related to the duration of the event
…		…	…	…

Event ID
The ID for specific log entry, identifying what has happened. For each kind of Notification, the Event ID register value correspond to a specific meaning.

Errors Logs Event IDs:

Code	Description
40	Audit log error
41	Firmware CRC error
42	Persistent storage error
43	RAM Memory CRC error
44	Firmware upgrade invalid image
45	Firmware upgrade maximum count
46	Firmware upgrade error
47	Firmware upgrade maximum invalid image count
51	Analog circuit reference error
52	Analog circuit temperature error
53	RTC circuit error

Warnings Logs Event IDs:

Code	**Description**
1000	U1 missing
1001	U2 missing (and it is not single-phase system)
1002	U3 missing (and it is not single-phase system)
1003	Not Locked device
1004	Power on line 1 < 0
1005	Power on line 2 < 0
1006	Power on line 3 < 0
1007	Total power < 0
1008	Frequency out of the metering limit
1010	Date not set
1011	Time not set
1012	U2 connected for single phase wires setup
1013	U3 connected for single phase wires setup
1014	I1 missing
1015	I2 missing (and it is not single-phase system)
1016	I3 missing (and it is not single-phase system)
1017	I2 connected for single phase wires setup
1018	I3 connected for single phase wires setup
1019	IN missing for 4 wires connection
1020	IN connected in non 4 wires connection
1021	Phase 1 connected to neutral
1022	Phase 2 connected to neutral
1023	Phase 3 connected to neutral
1024	Pulse 1 merged (2 high frequency or pulse length for measured power)
1025	Pulse 2 merged (2 high frequency or pulse length for measured power)
1026	Pulse 3 merged (2 high frequency or pulse length for measured power)
1027	Pulse 4 merged (2 high frequency or pulse length for measured power)
1028	Pulse 5 merged (2 high frequency or pulse length for measured power)
1029	Pulse 6 merged (2 high frequency or pulse length for measured power)
1030	Power Fail

Alarms Logs Event IDs:

Code	Description
2013	Notification of simple alarm 1
2014	Notification of simple alarm 2
2015	Notification of simple alarm 3
2016	Notification of simple alarm 4
2017	Notification of simple alarm 5
2018	Notification of simple alarm 6
2019	Notification of simple alarm 7
2020	Notification of simple alarm 8
2021	Notification of simple alarm 9
2022	Notification of simple alarm 10
2023	Notification of simple alarm 11
2024	Notification of simple alarm 12
2025	Notification of simple alarm 13
2026	Notification of simple alarm 14
2027	Notification of simple alarm 15
2028	Notification of simple alarm 16
2029	Notification of simple alarm 17
2030	Notification of simple alarm 18
2031	Notification of simple alarm 19
2032	Notification of simple alarm 20
2033	Notification of simple alarm 21
2034	Notification of simple alarm 22
2035	Notification of simple alarm 23
2036	Notification of simple alarm 24
2037	Notification of simple alarm 25
2038	Notification of complex alarm 1
2039	Notification of complex alarm 2
2040	Notification of complex alarm 3
2041	Notification of complex alarm 4
2042	Notification of complex alarm 5
2043	Notification of complex alarm 6

9.2 Reading Notifications

In order to read notification first the user has to write the related header.
There are several ways to read the notifications, the following statement apply to all the scenario:

- Writing 0 to the Entry Number register, restore a clean situation of the registers
- The Direction register of the header has to be written first. Then the user can write the Get next or the Entry Number register according to the reading scenario
- After writing the header the user can read back the Entry Number register of the header to have the information about the instance of the data block in first position
- After writing the Get next or the Entry Number register in the header there should be no other writing prior to read back the data block
- Writing a value in the Entry Number register bigger than the total amount of Notifications gives a Modbus error

Read most recent notification

Step	Action
1	Write 0 in the Entry Number register of the header to start a new enquiry.
2	Write 0 in the Direction register of the header.
3	Write 1 in the Get Next register of the header or write 1 in the Entry Number register of the header (same result).
4	(Optional) read back the Entry Number register of the header (it will be equal to 1).
5	Read the Data Block. The first entry is the latest (most recent) Notification.

Example of Alarms Log reading (Hex Format)
01 06 65 B1 00 00 C7 21 (Request: write Entry Number register)
01 06 65 B1 00 00 C7 21 (Response: write Entry Number register)
01 06 65 B7 00 00 27 20 (Request: write Direction register)
01 06 65 B7 00 00 27 20 (Response: write Direction register)
01 06 65 B0 00 01 57 21 (Request: write Get Next register)
01 06 65 B0 00 01 57 21 (Response: write Get Next register)
01 03 65 C0 00 69 9B 14 (Request: read the Data Block)
01 03 D2 14 07 09 0A 2E 17 00 08 07 DD FF FF FF FF 14 06 1D 0B 21 31 00 08 07 DD 00 00 1F E5 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF B3 62

Reading all the notifications from most recent back in time

Step	Action
1	Write 0 in the “Entry Number” register of the header to start a new enquiry
2	Write 0 in the “Direction” register of the header
3	Write 1 in the “Get Next” register of the header
4	Read back the “Entry Number” register of the header in order to understand what the entry number is related to the first position of the data block (the first reading will be 1, the second reading will be 16 …)
5	Read the Data Block.
6	Repeat steps from 3 to 5 until the data are filled with FFFF. For each “Get Next” writing, the Data Block is filled with new data.

9.3 Errors Flag, Alarms Flag and Warnings Flag

The following Modbus register are useful to get information on which are the alarms active and which are the errors and warnings are active.

Flag type	Start Reg (Hex)	Size	Description
Error flags	8A13	4	64 bits flags: 0 = error not active 1 = error active
Warning flags	8A1F	4	64 bits flags: 0 = warning not ac-tive 1 = warning active
Alarm flags (simple and com-plex)	8A25	4	64 bits flags: 0 = alarm not active 1 = alarm active

Each bit of the above registers represents respectively an alarm, an error or a warning according to the information described in the section: Event ID.
Hereafter an example for the Alarms Flag:

Bit	Bit 0	Bit 1	…….	Bit 25	…….	Bit 63
Value	1	0	…….	1	…….	0
Description	Simple alarm 1 active	Simple alarm 2 not active	…….	Complex alarm 1 active	…….	Not used

10 Alarms and Complex alarms

Alarm configuration defines the set of quantities to monitor. It is also defining the threshold and hysteresis values, delays, type and actions to perform for each alarm. Each alarm is configured individually.

When specified conditions are met, alarms have turned on or off. Triggering of alarms can be registered in the devices log. In addition to that, they can be set up to control digital outputs of the device.

The following table describes the group of registers for configuring the alarm parameters:

Function	**Start Reg (Hex)**	**Size**	**Description**	**Read / Write**
Alarm number	8C60	1	The number (identifier) for the alarm to configure	R/W
Parameter	8C61	3	The parameter to monitor (OBIS code)	R/W
Thresholds	8C64	4	ON and OFF thresholds to use to decide when the alarm is active	R/W
Hysteresis	8C68	1	Hysteresis to be applied to the turn off threshold	R/W
Delays	8C69	2	ON and OFF delays, defining the time that the measured value must be above/below the configured thresholds before the alarm triggers	R/W
Type	8C6B	1	The type of alarm: cross up or down	R/W
Actions	8C6C	2	Actions to perform when alarm is triggered	R/W

<u>Thresholds registers:</u>

The Thresholds registers are used to read and write the ON and OFF threshold registers values for an alarm. The scaling is the same as where the quantity appears in the mapping tables. The first (lowest) 2 registers are the ON threshold and the last 2 registers are the OFF threshold.

<u>Hysteresis register:</u>

The hysteresis register is used to calculate the Turn off threshold

*turn off threshold=threshold – (threshold * hysteresis)*

the value is intended as a percentage (%).

<u>Delays registers:</u>

The Delays registers are used to read or write the ON and OFF delays for an alarm. The delay is expressed in seconds. The first (lowest) registers is the ON delay and the second one is the OFF delay.

Type register:

The type register is used to define whether alarm trips on cross up or down

Possible Values	Description
0	None
1	Cross up
2	Cross down

Actions registers:

The Actions registers are used to read or write the actions to be performed when an alarm trigger. The first (lowest) register holds the actions to perform. The second register holds the number of the output to set, in case set output action is used.

Register nr (Hex)	Bit number	Description	Possible values
8C6C	0 (least significant bit)	Write entry to log	1 = use this action 0 = don't use
	1	Set output	1 = use this action 0 = don't use
	2	Set bit in alarm status register	1 = use this action 0 = don't use
8C6D	(Entire register)	Number of the output to turn on. Ignored if Set output bit above is set to 0.	

Follow the steps in the table below to configure the parameters for monitoring parameters for alarms

Step	Description
1	Write the number of the alarm to configure to the Alarm number register. This is a value between 1 and 25.
2	Write the OBIS code for the quantity to monitor to the Quantity registers.
3	Write the ON and OFF thresholds to the Thresholds registers.
4	Write the percentage value of hysteresis to the hysteresis register.
5	Write the ON and OFF delays to the Delays registers.
6	Write the cross up or cross down type to the type register.
7	Write the actions to perform to the Action registers.
8	Repeat these steps for all alarms that shall be used.

Follow the steps in the table below to read the current configuration of monitoring parameters for alarms:

Step	Description
1	Write the number of the alarm to read configuration for to the Alarm number register. This is a value between 1 and 25.
2	Read the Quantity registers to get the quantity monitored in the chosen alarm.
3	Read the Thresholds registers to get the ON and OFF thresholds.
4	Read the hysteresis registers to get the hysteresis value.
5	Read the Delays registers to get the ON and OFF delays.
6	Read the Type registers to get the type of alarm: cross up / cross down.
7	Read the Action registers to get the actions performed when an alarm is triggered.
8	Repeat these steps for all alarms that shall be used.

10.1 Complex alarm

There can be up to 4 complex alarms defined on the device. This type of alarm is used to combine simple alarm into single entity. It is possible to create complex alarms by combining the output of up to four alarms with logical AND and OR operators.

Alarm will be tripped every time result of logical equation will turn positive. According to configuration it will be logged, and/or digital output will be turned on. Once logical equation will turn negative again, turn off will get logged and/or digital output will be turned off.

The following table describes the group of registers for configuring the complex alarm parameters:

Function	Start Reg (Hex)	Size	Description	Read / Write
Complex alarm number	8C80	1	The number (identifier) for the complex alarm to configure	R/W
Components	8C81	4	Simple alarm active	R/W
Operator	8C85	1	Logical operator to use	R/W
Actions	8C86	2	Actions to perform when alarm is triggered	R/W

Component register:

The Component registers are used to define which are the simple alarms configured along with complex alarm.

Operator registers:

The Operator register is used to define which logical operator shall be used to combine simple alarms into complex alarm. Possible values are shown in the table below:

Possible values	Description
0	None
1	AND logical operator
2	OR logical operator

Actions registers:

The Actions registers are used to read or write the actions to be performed when an alarm trigger. The first (lowest) register holds the actions to perform. The second register holds the number of the output to set, in case set output action is used.

Register nr (Hex)	Bit number	Description	Possible values
8C86	0 (least significant bit)	Write entry to log	1 = use this action 0 = don't use
	1	Set output	1 = use this action 0 = don't use
	2	Set bit in alarm status register	1 = use this action 0 = don't use
8C87	(Entire register)	Number of the output to turn on. Ignored if Set output bit above is set to 0.	

Follow the steps in the table below to configure the parameters for monitoring parameters for complex alarms:

Step	Description
1	Write the number of the complex alarm to configure to the Complex alarm number register. This is a value between 1 and 4.
2	Write the corresponding bit of the Components register to assign a simple alarm to the complex alarm.
3	Write the desired logical operator to the Operator register.
4	Write the actions to perform to the Action registers.
5	Repeat these steps for all complex alarms that shall be used.

Follow the steps in the table below to read the current configuration of monitoring parameters for complex alarms:

Step	Description
1	Write the number of the alarm to read configuration for to the Complex alarm number register. This is a value between 1 and 4.
2	Read the Components registers to get the simple alarm monitored in the chosen complex alarm.
3	Read the Operator register to get the logical operator.
4	Read the Action registers to get the actions performed when a complex alarm is triggered.
5	Repeat these steps for all alarms that shall be used.

10.2 Alarm Status register

The following table describes the Status alarm registers used to read the status of each simple and complex alarm:

Function	Start Reg (Hex)	Size	Description	Read / Write
Alarm status	8A25	4	Bits flag representing the status of each simple/complex alarm	R

Each alarm status is represented by one bit

Possible Values	Description
0	Alarm not active
1	Alarm active

11 Inputs / Outputs

Inputs and outputs configuration define the function for each physical I/O port. It also defines the parameters for the logical pulse outputs.

The following table describes the group of registers for configuring the function for physical I/O ports:

Function	Start Reg (Hex)	Size	Description	Read / Write
I/O port 1	8C0C	1	Function of first, I/O port	R/W
I/O port 2	8C0D	1	Function of second, I/O port	R/W
I/O port 3	8C0E	1	Function of third, I/O port	R/W
I/O port 4	8C0F	1	Function of fourth, I/O port	R/W
I/O port 5	8C10	1	Function of fifth I/O port, only on M4M 30 I/O and M4M 20 I/O	R/W
I/O port 6	8C11	1	Function of sixth I/O port, only on M4M 30 I/O and M4M 20 I/O	R/W

The following table lists the possible values for I/O port function:

Possible Values	Function
0	Pulse input
1	Communication output
2	Alarm output
3	Pulse output
4	Tariff input
5	Output always ON
6	Output always OFF

I/O port can be configured as a Pulse output. The following table describes the group of registers for configuring the pulse outputs:

Function	Start Reg (Hex)	Size	Description	Read / Write
Pulse output instance	8C12	1	The instance number of the pulse output.	R/W
Port number	8C13	1	The physical I/O port on which the pulses are sent out.	R/W
Energy quantity	8C14	3	The OBIS code for the quantity	R/W
Pulse frequency active energy	8C17	2	The pulse frequency measured in pulses/MWh. This is relevant only if Energy quantitiy is set to active energy.	R/W
Pulse frequency reactive energy	8C19	2	The pulse frequency measured in pulses/Mvarh. This is relevant only if Energy quantity is set to reactive energy.	R/W
Pulse length	8C1B	2	The duration of a pulse measured in milliseconds.	R/W
Turn off pulse output	8C1D	1	Write the value 1 to this register to turn off the chosen pulse output instance.	R/W

Follow the steps in the table below to configure the pulse outputs:

Step	Description
1	Choose the pulse output instance to configure by writing a number to the Pulse output instance register. Allowed values are 1-6.
2	Write to the Port number register to decide to which physical port the pulses are sent out for the chosen pulse output. Allowed values are 0-6, where 0 means No Output.
3	Write the OBIS code of the quantity that shall be used for the chosen pulse output to the Energy quantity registers. OBIS codes are available on the Modbus map document.
4	Write the desired pulse frequency to the Pulse frequency active or reactive energy registers, depending on the chosen energy type.
5	Write the desired pulse length to the Pulse length registers.
6	Repeat steps 1 to 5 for all pulse outputs.

Follow the steps in the table below to turn off a pulse output instance:

Step	Description
1	Choose the pulse output instance to configure by writing a number to the Pulse output instance register. Allowed values are 1-6.
2	Write the value 1 to the Turn off pulse output register.

Follow the steps in the table below to read the current pulse output configuration:

Step	Description
1	Choose the pulse output instance to read configuration for by writing a number to the Pulse output instance register. Allowed values are 1-6.
2	Read the Port number register to get the I/O port number used by the chosen pulse output instance.
3	Read the Energy quantity registers to get the OBIS code of the quantity used for the chosen pulse output instance.
4	Read the Pulse frequency active or reactive energy registers, depending on the chosen energy type, to get the pulse frequency used by the chosen pulse output instance.
5	Read the Pulse length registers to get the pulse length used by the chosen pulse output instance.
6	Repeat steps 1 to 5 for all pulse outputs.

11.1 Input / Output status register

The following table describes the registers used to read the status of the I/O configured as output. The same registers are user to set the value of the register is the I/O is configured as "Communication Output". I/O slots n. 5 and 6 are only available for M4M 20 I/O and M4M 30 I/O.

Function	Start Reg (Hex)	Size	Description	Read / Write
Output 1	6300	1	Status: ON=1, OFF=0	R/W
Output 2	6301	1	Status: ON=1, OFF=0	R/W
Output 3	6302	1	Status: ON=1, OFF=0	R/W
Output 4	6303	1	Status: ON=1, OFF=0	R/W
Output 5	6304	1	Status: ON=1, OFF=0	R/W
Output 6	6305	1	Status: ON=1, OFF=0	R/W

If I/O is configured as input, then Output registers =-1 (0xFFFF). Write is allowed only if the I/O is configured as "Communication Output.

The following table describes the registers used read the status of the I/O if configured as "Tariff Input".

Function	Start Reg (Hex)	Size	Description	Read / Write
Input 1 status	6308	1	Status: ON=1, OFF=0	R
Input 2 status	6309	1	Status: ON=1, OFF=0	R
Input 3 status	630A	1	Status: ON=1, OFF=0	R
Input 4 status	630B	1	Status: ON=1, OFF=0	R
Input 5 status	630C	1	Status: ON=1, OFF=0	R
Input 6 status	630D	1	Status: ON=1, OFF=0	R

12 Tariff

Tariff configuration defines the currently used tariff source, i.e. communication, clock or inputs. It also defines the settings that are specific for each of these sources.

The following table describes the group of registers for configuring day profiles:

Function	Start Reg (Hex)	Size	Description	Read / Write
Number of day profile	8CCD	1	The number of day profiles used (1-16)	R/W
Day profile number	8CCE	1	Current day profile number during read or write of configuration	R
Number of actions	8CCF	1	The number of actions during a day profile (1-30)	R/W
Action number	8CD0	1	Current action number during read or write of configuration	R
Action	8CD1	2	Time when the action shall be performed, and what to do	R/W

The following table describes the group of registers for configuring a day profile action:

Function	Byte number	Description
Execution time	0 (High byte)	Hour when the action shall be performed.
	1 (Low byte)	Minute when the action shall be performed.
Action ID	Both bytes	Decides the action to perform. See the list of possible actions below.

Note: Both registers in the table above must be written in one operation, otherwise the values will not take effect.

Possible actions to perform are activating tariffs and setting or resetting outputs. The possible values for action id are listed in the table below:

Possible Values	Description
0	Activate tariff 1
….	….
3	Activate tariff 4
….	….
100	Set output 1
101	Reset output 1
….	….
106	Set output 4
107	Reset output 4
….	….

Follow the steps in the table below to configure the day profiles:

Step	Description
1	Write the number of day profiles to use to the Number of day profiles register. This is a value between 1 and 16.
2	Write the number of actions to perform for the first day profile to the Number of actions register. This is a number between 1 and 30.
3	Write the execution time and action id for the first action to perform during the day to the Action registers.
4	Repeat step 3 for all actions that shall be performed during the day, i.e. the same number of times as the value written in step 2.
5	Repeat step 2-4 for all day profiles, i.e. the same number of times as the value written in step 1.

Follow the steps in the table below to read the current day profile configuration:

Step	Description
1	Read the Number of day profiles register to find out how many day profiles are used.
2	Read the Number of actions register to find out how many actions are configured for the first day profile.
3	Read from the Action registers to get the execution time and action id for the first action.
4	Repeat step 3 for all actions that are configured for the day, i.e. the same number of times as the value read in step 2.
5	Repeat step 2-4 for all day profiles, i.e. the same number of times as the value read in step 1.

Note: Step 1 and 2 initiate the readout procedure and can NOT be left out, even if the number of day profiles and actions used are already known.

Note: The Day profile number register can optionally be read together with the Number of actions register in step 2. The Day profile number register holds the current day profile number, starting from 1 after reading the Number of day profiles register. It is incremented every time the last action during the day is read from Action registers. In the same way the Action number register can optionally be read together with the Action registers in step 3. The Action number register holds the current action number, starting from 1 after reading the Number of actions register. It is incremented every time the Action registers are read.

The following table describes the group of registers for configuring special days:

Function	Start Reg (Hex)	Size	Description	Read / Write
Number of special days	8CD3	1	The number of special days used (1-50).	R/W
Special day number	8CD4	1	Current special day number during read or write of configuration.	R
Special day	8CD5	3	Date and associated day ID for the special day.	R/W

The following table describes the group of registers for configuring a week profile:

Contents	Register	Byte nr	Description
Date	8CD5	0 (high byte)	Year
		1	Month
	8CD6	0	Day
		1	Not used
Day id	8CD7	Both	Day ID associated with the special day

Note: All 3 registers in the table above must be written in one operation, otherwise the values will not take effect.

12.1 Current tariff register

The following table describes the Current tariff register used to read or write which tariff is activated:

Function	Start Reg (Hex)	Size	Description	Read / Write
Current tariff	8A07	1	Represents the active current tariff	R

If the tariff source is set to communication, then those register can be written and the user can set the active tariff.

12.2 Daylight Savings Time

Daylight savings time (DST) can be enabled and if enabled it has a start and end time that can be defined. The following table shows an overview of the mapping table:

Function	Details	Start Reg (Hex)	Size
DST	DST start	8CE6	2
DST	DST end	8CE8	2
DST	DST enabled	8CEA	1

DTS start register:

The following table shows the contents of the registers for configuring DST start. The structure of the DST end registers is the same as for DST start.

Function	Start Reg (Hex)	Size	Description	Read / Write
DST start	8CE6	2	Byte 0: month (highest byte of lowest register) Byte 1: day of month Byte 2: day (e.g. 1=Monday, 7=Sunday) Byte 3: hour	R/W

For month, day of month and day of week wild cards can be used:

Setting	Wild cards available
Month	Not specified: 255
Day of month	Not specified: 255; Last day of month: 254; Second last day of month: 253
Day of week	Not specified: 255
Hour	-

DTS enabled register:

The DST enabled register decides whether the DST functionality of the meter is turned on or not.

Possible Values	Description
0	OFF
1	ON

Example of DST start/end:

- Month = 3, Day of month = 254, Day of week = 7, Hour = 2 means last Sunday of March 02:00.
- Month = 3, Day of month = 254, Day of week = 255, Hour = 2 means last day of March 02:00, regardless on which weekday it occurs.
- Month = 3, Day of month = 2, Day of week = 7, Hour = 2 means second Sunday of March 02:00.

13 Audit Log

Audit Log stores an event after an attempt has been made to upgrade the firmware and/or CT ratio, VT ratio or Type of network is modified.
It is possible to read Audit Log via Modbus communication.

Mapping table – Audit Log
Audit Log has a header and a data Block, according to the table below.

Function	Details	Start Reg (Hex)	Size
Audit Log	Header	6660	7
Audit Log	Data Block	6670	68

Header for Audit Log registers:
The following table describes Audit Log header registers:

Function	Start Reg (Hex)	Size	Description	Read/ Write
Get next entry	6660	1	Write value 1 to this register to load the next block of audit log entries.	W
Entry number	6661	1	Write to this register to choose an entry number to start reading from.	R/W
Direction	6667	1	Write to this register to choose the direction of reading. 1- newer to older blocks 0- older to newer blocks	R/W

<u>Data block registers:</u>
The data block contains the audit log entries.

Function	Start Reg (Hex)	Size	Read/Write
Entry number	6670	2	R
Timestamp (second from 2010-01-01)	6672	2	R
Upgrade counter	6674	1	R
Firmware version	6675	6	R
VT ratio primary	667B	2	R
VT ratio secondary	667D	2	R
CT ratio primary	667F	2	R
CT ratio secondary	6681	2	R
CT ratio (N) primary	6683	2	R
CT ratio (N) secondary	6685	2	R
Type of network	6687	1	R
Active energy import Total	6688	4	R
Active energy import L1	668C	4	R
Active energy import L2	6690	4	R
Active energy import L3	6694	4	R
Active energy Tariff 1	6698	4	R
Active energy Tariff 2	669C	4	R
Active energy Tariff 3	66A0	4	R
Active energy Tariff 4	66A4	4	R
Active energy Tariff 5	66A8	4	R
Active energy Tariff 6	66AC	4	R
Active energy export Total	66B0	1	R

13.1 Reading Audit Log

Readout is controlled by the Entry number register. After writing the Entry number register, log entries are available in the registers of the data block. To get next set of entries, the Get next entry register is used.
Follow the steps below to read the most recent Audit log entries:

Step	Action
1	Write the value 1 to the entry number register.
2	Read the data block.

Follow the steps below to read the history of Audit logs, backwards in time:

Step	Action
1	Write the value 0 to the entry number register to make sure the reading starts from the most recent entry.
2	Write the value 1 to the Get next entry register.
3	Read the Data block.
4	Repeat the steps 2 and 3 until there are no more entries stored. When all entries have been read, all registers in the block are set to 0xFFFF.

14 LED Source

The functionality of the Led, placed on top of the device, is to flash in proportion to the energy measured. Its related modbus register allow to the user to set proportional flashing according to Active Energy or Reactive Energy measured.

Quantity	**Start Reg (Hex)**	**Size**	**Description**	**Read/Write**
LED Source	8CE4	1	0 = Active Energy 1 = Reactive Energy	R/W

MID version: LED source Modbus register is fixed to Active Energy and it is a read-only register.