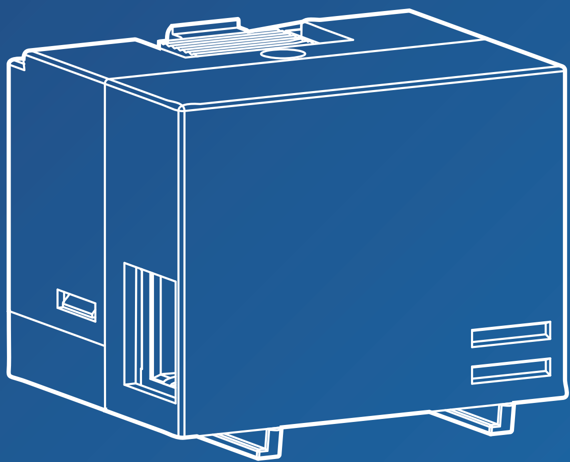
User guide

hw+

Modbus communication

sentinel Energy



hager

Contents

Page

01 About this manual **3**

1.1 Safety instructions 3
1.2 Using this manual 5

02 Modbus communication with an hw+ circuit breaker **6**

2.1 General description 6
2.2 Modbus-RTU communication module 7
2.3 Modbus-TCP communication module 11
2.4 Network Security 13
2.5 HTTPS server embedded in the Modbus-TCP communication module 14

03 Use of the Modbus protocol **18**

3.1 Modbus Functions 18
3.2 Exception codes 22
3.3 Structure of the sentinel Energy Modbus table 23
3.4 Secure write command procedure 26
3.5 Changing the power sign 28
3.6 Configuration and control of tariff meters 32
3.7 Inhibition of advanced protections 35
3.8 Switching between profile A and profile B 40
3.9 Remote opening and closing commands 45
3.10 Other embedded commands 50
3.11 Read the events history 51

04 Appendices **61**

4.1 Glossary 61
4.2 Modbus table of registers 62
4.3 SHA-224 hash function 63
4.4 Custom alarm identifiers 64
4.5 Information on the software licences 67

Warnings and instructions

This documentation contains safety advice which must be respected for your own safety and to prevent property damage.
Safety advice relating to your own safety is identified by a safety warning symbol in the documentation. Safety advice relating to damage to property is identified by "ATTENTION".
The safety warning symbols and the wording below are classified according to the risk level.

⚠ DANGER
DANGER indicates an imminent dangerous situation which, if not avoided, will result in death or serious injuries.

⚠ WARNING
WARNING indicates a potentially dangerous situation which, if not avoided, may result in serious injuries or even death.

⚠ CAUTION
CAUTION indicates a potentially dangerous situation which, if not avoided, may result in minor or moderate injuries.

ATTENTION
ATTENTION indicates a warning message relating to equipment damage. **ATTENTION** also indicates important instructions for use and particularly relevant information regarding the product, which must be respected to ensure effective and safe use.

Qualified personnel

The product or the system described in this documentation must be installed, operated and maintained by qualified personnel only. Hager Electro accepts no responsibility regarding the consequences of this equipment being used by unqualified personnel.
Qualified personnel are those people who have the necessary skills and knowledge for building, operating and installing electrical equipment, and who have received training enabling them to identify and avoid the risks incurred.

Appropriate use of Hager products

Hager products are designed to be used only for the applications described in the catalogues and in the technical documentation relating to them. If products and components from other manufacturers are used, they must be recommended or approved by Hager.
Appropriate use of Hager products during transport, storage, installation, assembly, commissioning, operation and maintenance is required to guarantee problem-free operation in complete safety.
The permissible ambient conditions must be respected. The information contained in the technical documentation must be respected.

Publication liability

The contents of this documentation have been reviewed in order to ensure that the information is correct at the time of publication.
Hager cannot, however, guarantee the accuracy of all the information contained in this documentation. Hager assumes no responsibility for printing errors and any damage they may cause.
Hager reserves the right to make the necessary corrections and modifications to subsequent versions.

Cybersecurity

The product or the system described in this documentation requires protective measures to be set up against the risks intrinsic to any wireless connection and transmission and the risks intrinsic to any cable-based connection and transmission.

⚠ WARNING

Risks that could affect the availability, integrity and confidentiality of the sentinel Energy system

- Change the default passwords during first use to prevent any unauthorised access to the device settings, controls and information.
- Disable unused ports and services, as well as default accounts, to reduce the risk of malicious attacks.
- Protect the devices in the network with several levels of cyberdefence (firewall, network segmentation, intrusion detection and network protection).
- Respect good cybersecurity practices (for example: least privilege principle, separation of tasks) to reduce the risks of intrusion, the loss or alteration of data and logs, or the interruption of services.

Failure to follow these instructions may result in death, serious injury or material damage.

Purpose of the document.

This manual is designed to provide users, electricians, panel builders and maintenance personnel with the technical information required for the use of the Modbus protocol on hw+ circuit breakers fitted with sentinel Energy electronic trip units

Field of application

This document is applicable to hw+ circuit breakers equipped with a Modbus RTU communication module or a Modbus-TCP communication module.

Revisions

Version	Date
6LE007964A	February 2023

Documents to consult

Document	Reference
sentinel Energy Modbus communication table	6LE009231A
HW1 installation manual	6LE007893A
HW2 / HW4 installation manual	6LE009206A
HW1 user maintenance guide	6LE007897A
User manual for sentinel Energy hw+ electronic trip units	6LE008147A
HTD210H panel display user manual	6LE002999A

You can download these publications and other technical information from our website: www.hager.com

Contact

Address	Hager Electro SAS 132 Boulevard d'Europe 67215 Obernai France
Phone	+ 33 (0)3 88 49 50 50
Website	www.hager.com

The hw+ circuit breaker fitted with a sentinel Energy trip unit can connect either to a Modbus master for an RTU module, or to a Modbus client for a TCP module.

It can connect:
- to an RS 485 serial link network using the Modbus RTU protocol with an HWY965H communication module;
- to an Ethernet network using the Modbus TCP/IP protocol with an HWY966H communication module.

Modbus communication with an hw+ sentinel Energy circuit breaker grants access to a wide range of functions, in particular:
- reading diagnostic and measurement data
- reading status conditions and remote operations
- displaying the protection settings
- reading circuit breaker identification and configuration data
- remote control of the circuit breaker
- setting the clock and synchronisation
- configuration of the protections and alarms
- remote control and configuration of tariff meters
- setting various other parameters
- remote inhibition of advanced protections
- remote switching between two protection profiles (A and B)
- the use of other embedded commands
- the transfer of the timestamped events history.

The Modbus-RTU and Modbus-TCP communication modules have been designed according to version 1.1.B3 of the Modbus Application protocol standard.
The Modbus-RTU communication module complies with the Modbus serial V.1.02 protocol standard.
The Modbus-TCP communication module complies with the protocol standard Modbus messaging on TCP/IP V.1.0B.

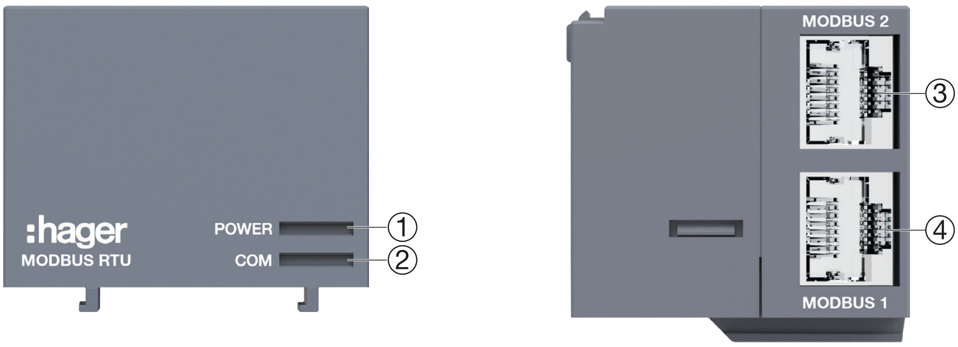
The Modbus-RTU communication module can be connected to an RS 485 serial link network using the Modbus- RTU protocol.

Modbus RTU module

This communication module is compatible with the agardio.manager HTG411H data server.

The Modbus RTU communication module is equipped with two RJ45 sockets on the left side of the product. This enables serial link connection to other Modbus participants in accordance with the "Daisy chain" principle.

The MODBUS 1 and MODBUS 2 ports can be used in an inlet/outlet direction as well as an out-going/in-going direction.



①	Power	Permanent green	Communication module powered and functional
		Flashing green	Synchronisation with the trip unit
		Flashing red	Internal fault
②	COM	Flashing green	Modbus data transmission
		Flashing red	Modbus transmission failure
③	RTU port	Modbus 1	
④	RTU port	Modbus 2	

Settings

From the sentinel Energy display or the Hager Power setup software.

MODBUS RTU	Value	Description
ADDRESS	1 to 247	Modbus address setting
BAUD	4800; 9600; 19200; 38400	BAUD rate setting
PARITY	Odd	1 stop bit
	Even	1 stop bit
	no parity	2 stop bits
STOP BIT	1 or 2	Setting the parity includes automatically managing the automatic adjustment of the number of Stop bits.

Wiring

The Modbus-RTU communication module uses a serial link in accordance with the EIA/TIA-485 standard, also known as the RS485 standard. It must be used in a Modbus system or a master device and one or more slave devices communicate via serial link.
All devices must be connected to a bus cable (shielded twisted pair) that complies with the Modbus-RTU specifications. This is required for bi-directional transmission of data.

Modbus cable recommendation

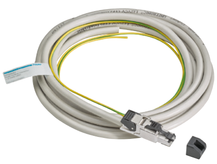
For internal installation, it is recommended that a data cable with the following characteristics is used:
- a shielded twisted pair with tin-plated copper braiding
- AWG 24 or 0.25 mm^2 cross section
- characteristic impedance of 100 to 120 ohms
- maximum linear resistance of 160 ohms/km
- maximum capacitance between the conductors 60 nF/km
- maximum capacitance between the conductors and shielding of 160 nF/km
- maximum bus cable length of 1200 m.

Cable example: transmission cable LAPP UNITRONIC Li2YCY (TP) fine-wire 2 x 2 x 0.25 mm^2.

Modbus-RTU cable accessories

The HTG471H to HTG484H reference models below are intended for cabling RJ45-compatible Modbus participants. These cables can be twisted on condition that they respect a minimum bend radius in fixed installation of 33 mm. They contain plastic material without halogen.

The reference items HTG465H and HTG485H can be cut to size. These cables can be twisted on condition that they respect a minimum bend radius in fixed installation of 41.5 mm. They contain halogen materials.

			Reference (factory-assembled)	**Reference (separate accessory)**
 HTG481H	RJ45 - RJ45	0.2 m	-	HTG480H
		1 m	-	HTG481H
		2 m	-	HTG482H
		5 m	-	HTG484H
 HTG471H	RJ45 - RJ45 with earth	1 m	-	HTG471H
		2 m	-	HTG472H
		5 m	-	HTG474H
 HTG465H	RJ45 with earth - bare wires	3 m	-	HTG465H
 HTG485H	Modbus cable	25 m	-	HTG485H
 HTG467H	120 Ohm RJ45 terminal resistor		-	HTG467H

The cable bus length can vary from 80 to 1200 metres depending on the transmission speed and repeater type.

The shielding protects the cable from electromagnetic influences.
It must be grounded at least at each distribution board to guarantee the equipotentiality of the shielding.

Terminal resistor

A terminal resistor of 120 Ω must be connected to both ends of the bus cable in order to stabilise the voltage level throughout the bus cable.
In the case of an installation including several Modbus-RTU communication modules, it is essential that the last module be connected with a 120 Ω resistor (reference model HTG467H) to integrate terminal impedance into the Modbus chain.

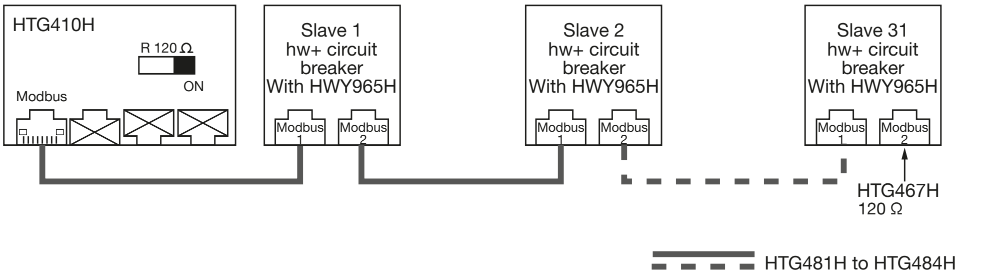
Cable recommendations to be followed

Use a shielded twisted pair cable, minimum cross-sectional area of 0.25 mm^2 or AWG 24, 2 pairs, characteristic impedance 120 Ω.
The HTG4xxH preassembled cables are recommended.

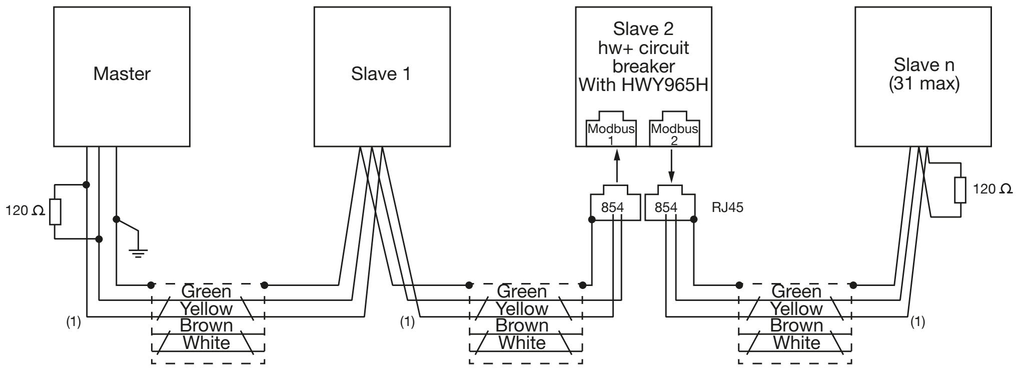
Daisy chain connection

Connecting the master and slaves in a daisy chain connection is essential for the system to operate correctly.

Connection with Modbus RJ45 cable between agardio.manager data server and hw+ circuit breakers

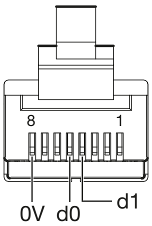


Connection between Modbus master, hw+ circuit breaker and other slaves



(1) Li2YCYTP cable type (HTG465H or HTG485H)

Position of the pins on a male/female RJ45 connector.



Pin 4 = D1 or B/B’ or (+)	Green conductor
Pin 5 = D0 or A/A’ or (-)	Yellow conductor
Pin 8 = Common or C/C’ or (0V)	Not used

ATTENTION

For a serial bus (Modbus-RTU), the slaves must be connected one after the other, in accordance with the daisy chain principle.
Star or branch connections are not permitted, as reflections may occur in the cable and data may be corrupted.

24 V power supply

The Modbus-RTU communication module is also powered by the TU terminal block of the hw+ circuit breaker (recommended reference model Hager HTG911H 24 V DC power supply).

HWY965H electrical characteristics:

Operating voltage	24 V DC (+/- 30 %) SELV
Current consumption	14 mA

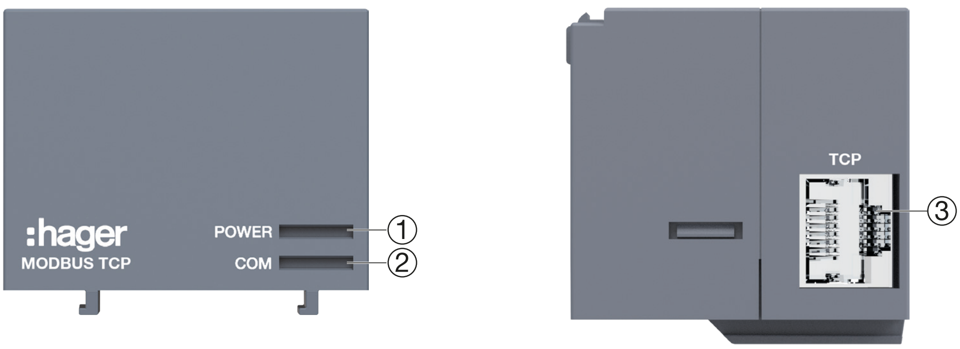
The Modbus-TCP communication module can be used to connect to an Ethernet network using the Modbus TCP protocol.

Modbus TCP module

This communication module is compatible with the agardio.manager HTG411H data server.

The Modbus-TCP communication module offers the possibility of deploying a Modbus-TCP/IP communication network that can be secured via TLS (See Chapter 2.4 Network Security).

The Modbus-TCP communication module is equipped with an RJ45 socket on the left side.



①	Power	Permanent green	Communication module powered and functional
		Flashing green	Synchronisation with the trip unit data
		Permanent red	Internal fault
②	COM	Flashing green	Modbus data transmission
		Flashing red	Modbus transmission failure
③	Ethernet port	Modbus 1	

Settings

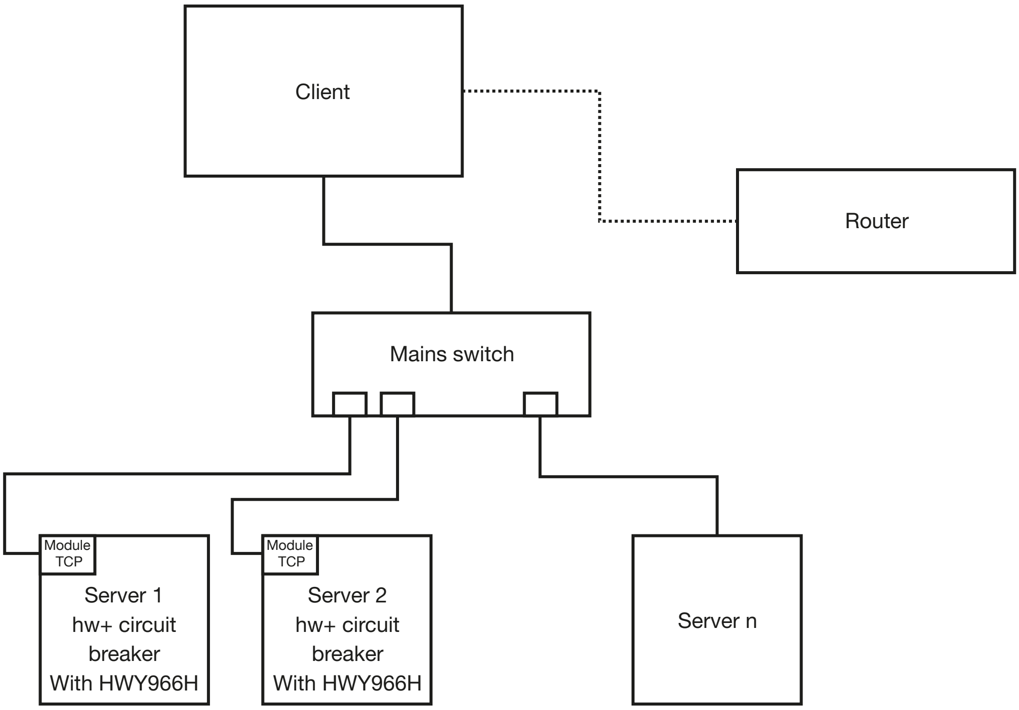
The Modbus TCP module can be configured from the sentinel Energy display or the Hager Power setup software for use of a fixed IP address. In cases where a DHCP dynamic IP address server is used or the Modbus Secure protocol is deployed, the Modbus TCP module is configured from the embedded HTTPS server.

MODBUS TCP	Default value	Description
DHCP	Off	Off: the IP address must be configured manually. On: the IP address is assigned automatically by the network.
IP ADDRESS	172.16.1.1	Either the fixed IP address is entered or it will be assigned automatically (DHCP setting On).
SUBNET MASK	255.255.255.0	Enter the sub net mask if necessary.
GATEWAY	0.0.0.0	Enter the network gateway if necessary.

Wiring

The Modbus-TCP communication module must be used in a Modbus system or a client device and one or more server devices communicate via IP cable bus.
It is recommended that a category 4 or 5 Ethernet cable be used.
The Modbus TCP communication module does not have a network switch function.
It must be connected to the client via a network switch.

Connection example



24 V power supply

ATTENTION
The Modbus-TCP communication module can communicate with a maximum of 5 clients at the same time.

The Modbus-TCP communication module is also powered by the TU terminal block of the hw+ circuit breaker (recommended reference model Hager HTG911H 24 V DC power supply).

HWY966H electrical characteristics:

Operating voltage	24 V DC (+/- 30 %) SELV
Current consumption	38 mA

Depending on the cybersecurity level of the IP network chosen and deployed on the site where the hw+ circuit breaker is installed, it may be necessary to reinforce the security of access to the sentinel Energy server (Modbus-TCP communication module).

TLS or Transport Layer Security, formerly known as SSL, is a technology intended to secure internal connections and protect sensitive data transmitted between two participants, preventing any cybercriminal from reading or altering the information transferred, including any personal information.
Here the two participants are the sentinel Energy server and a SCADA-type client system for example.

The Modbus-TCP communication module offers a choice of 3 security levels for network access to the Modbus-TCP communication module:
- unsecured Modbus (no TLS, no authentication)
- secure Modbus (TLS, authentication)
- secure Modbus (with TLS and mutual authentication).

Data encryption
Thanks to data encryption, data exchanged between two participants can no longer be read by third parties.

Authentication of actors
Each participant receives a public key certificate in accordance with the X.509 standard. This key is used by each of the participants to prove to the others that they are worthy of confidence.

Modbus not secured
The communication module is delivered with Modbus not secured in its factory settings, TLS deactivated.

Simply secure Modbus
TLS data encryption is activated but without mutual authentication.

Totally secure Modbus
Mutual authentication is activated in addition to the activation of TLS encryption. Thee client sends its identity to the server, while the server verifies the identity of the client. The server does exactly the same thing, i.e. it transmits its identity to the client while the client verifies the identity of the server.

The Modbus-TCP communication module includes an HTTPS server enabling advanced configuration of the IP network, particularly when DHCP dynamic configuration is used.

This allows the risks of intrusion to be sharply reduced in the context of cybersecurity measures.

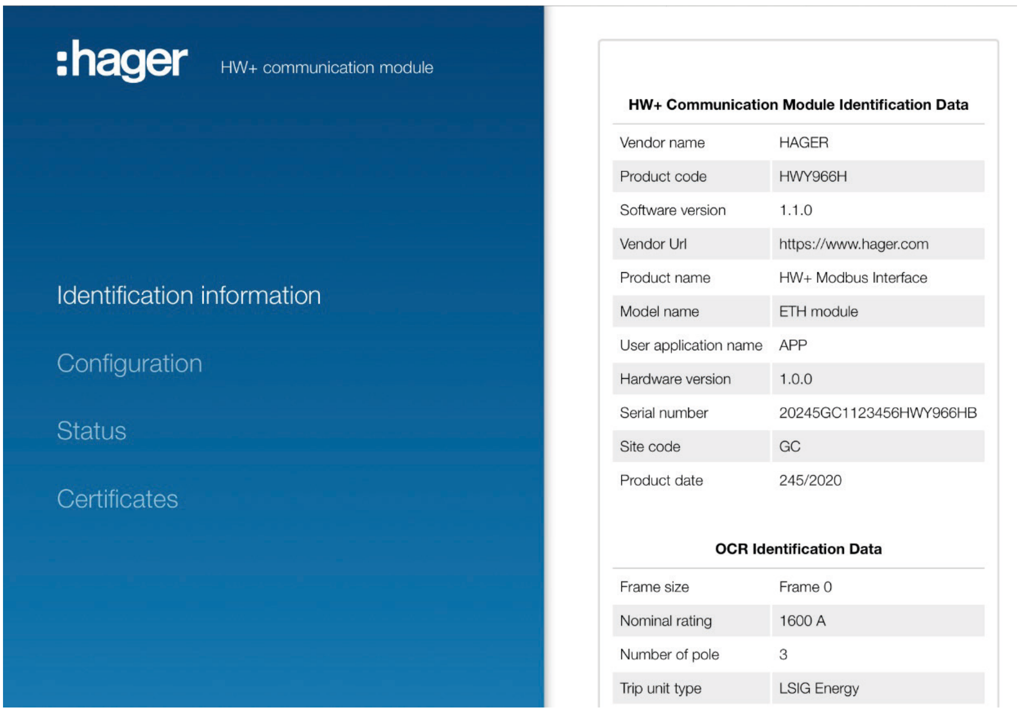
Connect a computer to the Ethernet port of the Modbus TCP module to access the interface. Then use a browser to open the link https://172.16.1.1 (default IP address),

Then enter the default login:
- Login: admin
- Password: admin

The password must be changed the first time a connection is made.
The Modbus-TCP interface is available in English and Chinese.

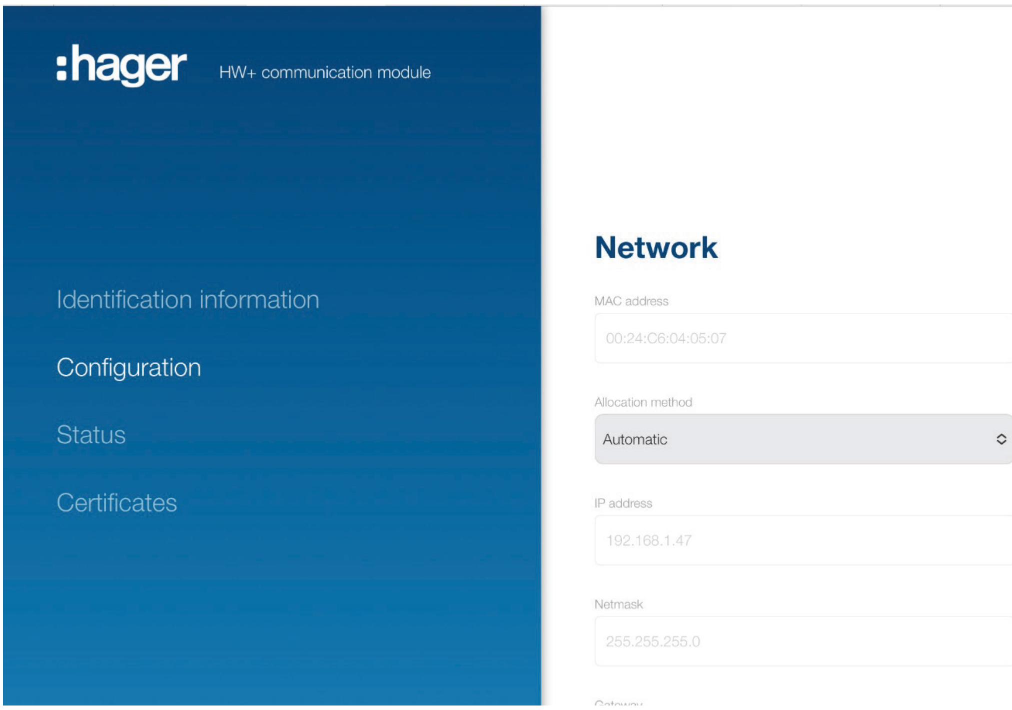
The interface includes 4 menus.

Identification information menu

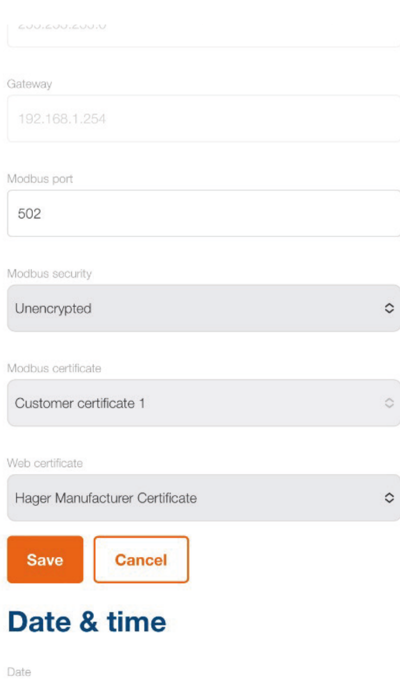


This page displays identification information about the communication module and the sentinel Energy electronic trip unit.

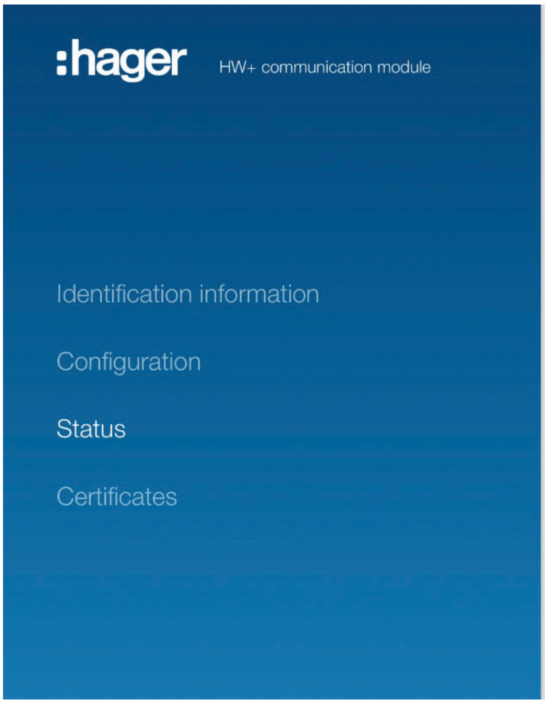
Configuration menu



This page allows Modbus-TCP communication to be configured:
- Static IP addressing mode or dynamic DHCP configuration
- Entry of the IP address of the sentinel Energy server, the subnet mask and the gateway
- Choice of Modbus port (502 by default)
- Choice of Modbus secure protocol
- Choice of web certificate
- Setting the date and time manually or automatically through an update request to an SNTP server.



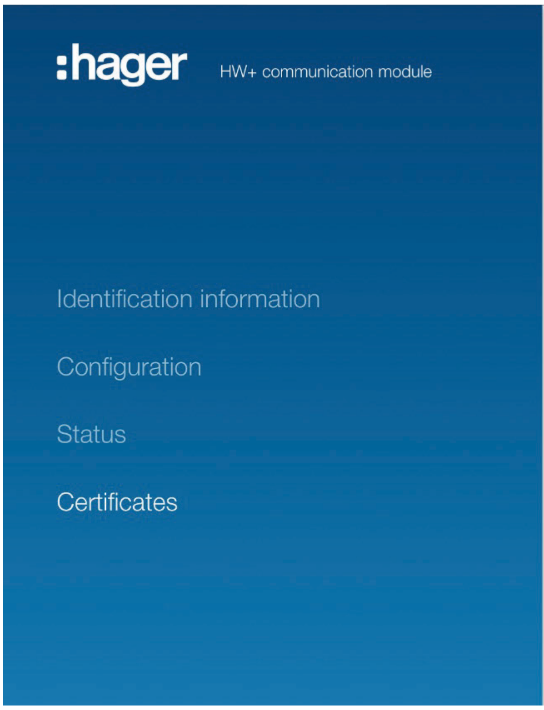
Status Menu

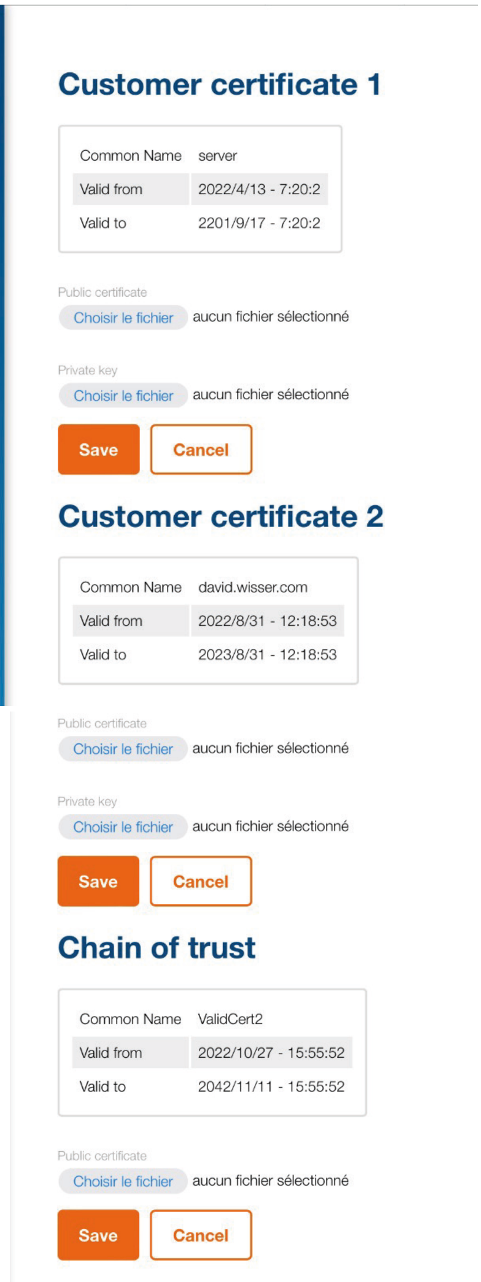


OCR communication status	Connected
Number of connected Modbus clients	0
Connected Modbus clients IP addresses	
Number of connected Web clients	1

This page displays the status of the communication with the sentinel energy trip unit. It displays the clients connected to the Modbus server as well as their IP address.

Certificates Menu





This page allows X.509 certificates to be filed that authenticate the Modbus server or the HTTPS server. It can also be used to store the chain of trust used for mutual authentication.

The hw+ circuit-breaker can perform up to 4 read, write or diagnostic functions using the Modbus protocol. It also allows one Hager function adapted to specific requirements to be performed.

Read functions

Function code	Name	Description
H'03 (0x03)	Read holding registers (Read holding registers)	Read n output registers or n internal registers
H'43 14 (0x2B / 0x0E)	Read device identification (Read Device Identification)	Read slave (or server) identification data

Example of reading the hold register

This examples shows how to read the frequency in registers 4458 and 4459. The address of the register 4458 is 4458 or 0x116A (Hex). The Modbus address of the slave is 14 = 0x0E (Hex).

Request from the master or client

Frame description	Value
Address of the slave	0x0E
Function code	0x03
Address of the register to be read (MSB)	0x11
Address of the register to be read (LSB)	0x6A
Number of registers (MSB)	0x00
Number of registers (LSB)	0x02
CRC (MSB)	0xXX
CRC (LSB)	0xXX

Response of the slave or the server

Frame description	Value
Address of the slave	0x0E
Function code	0x03
Length of the data in bytes	0x04
Value of register 1 (MSB, 1st byte)	0x00
Value of register 1 (LSB, 2nd byte)	0x00
Value of register 2 (MSB, 1st byte)	0xC3
Value of register 2 (LSB, 2nd byte)	0x46
CRC (MSB)	0xXX
CRC (LSB)	0xXX

The content of the registers 4458 and 4459 is 00 00 C3 46 which corresponds to a frequency of 49.990 Hz.

Write function

Function code	Name	Description
H'16 (0x10)	Write multiple registers (Write multiple registers)	Allows the master (or client) to write in the registers if the remote write authorisation parameter is activated on the sentinel Energy trip unit.

Example

This example shows an extract of the complete Modbus frame to change the power sign parameter with the secure write command.

Write request from the master or client

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x10	Write multiple registers function code (Write multiple registers)
Address of the first register (MSB)	0x30	Start of write at address 0x3000
Address of the first register (LSB)	0x00	
Number of registers to write (MSB)	0x00	18 registers
Number of registers to write (LSB)	0x12	
Number of bytes to write	0x00	Write over 36 bytes
Value in register 1 (MSB)	0x00	For details of the values refer to Chapter 3.5 Example secure write of the power sign command.
Value in register 1 (LSB)	0x65	
etc.	etc.	
Value in register 18 (MSB)	0x00	
Value in register 18 (LSB)	0x00	
CRC (MSB)	0xXX	-
CRC (LSB)	0xXX	

Read the response of the slave or the server

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x03	Holding register read function code
Address of the first register (MSB)	0x32	Start of read at address 0x3200
Address of the first register (LSB)	0x00	
Number of registers to write (MSB)	0x00	18 registers
Number of registers to write (LSB)	0x12	
Register 1 (MSB, 1st byte)	0x03	Identifier of command 1001 associated with the power sign parameter
Register 1 (LSB, 2nd byte)	0xEA	
Register 2 (MSB, 1st byte)	0x00	Success code = 0
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length of the data parameter in bytes
Register 3 (LSB, 2nd byte)	0x02	
Register 4 (MSB, 1st byte)	0x00	Value of the power sign parameter: "+" = 0
Register 4 (LSB, 2nd byte)	0x00	
CRC (MSB)		-
CRC (LSB)		

Diagnostic function

The list of diagnostic function codes supported by the HWY965H and HWY966H communication modules is shown below. For more detailed information about the diagnostic functions, see version 1.1.B3 of the Modbus application protocol standard.

Function code	Function sub-code	Name	Description
H’08 (0x08)	10 (0x000A)	Clear Counters and Diagnostic Register	Reset of all diagnostic counters
H’08 (0x08)	11 (0x000B)	Return Bus Message Count	Read correct bus message counter
H’08 (0x08)	12 (0x000C)	Return Bus Communication Error Count	Read incorrect bus message counter
H’08 (0x08)	13 (0x000D)	Return Bus Exception Error Count	Read exception responses counter
H’08 (0x08)	14 (0x000E)	Return Server Message Count	Read the counter of messages sent to the slave/server or sent to all slaves through boardcast message
H’08 (0x08)	15 (0x000F)	Return Server No Response Count	Read counter of messages sent to the slave without any response from it
H’08 (0x08)	16 (0x0010)	Return Server NAK Count	Read the counter of messages sent to the slave with a negative acknowledgement exception response ((NAK). For the list of exception responses returned refer to Chapter 7 of the Modbus Application protocol standard version 1.1.B3.
H’08 (0x08)	17 (0x0011)	Return Server Busy Count	Read the counter of messages sent to the slave/server with exception response "Server Device Busy"
H’08 (0x08)	18 (0x0012)	Return Bus Character Overrun Count	Read the counter of incorrect bus messages due to overload errors

The Modbus protocol uses diagnostic counters to activate error and performance management. The following counters are accessible using Modbus diagnostic functions:

Number	Description
1	Correct bus message counter
2	Incorrect bus message counter
3	Counter of exception responses of incorrect broadcast messages
4	Counter of messages sent to the slave
5	Counter of broadcast messages
6	Counter of messages sent to the slave but with no response due to exception code 07 negative acknowledgement
7	Counter of messages sent to the slave but with no response due to exception code 06 slave device busy
8	Counter of incorrect bus messages due to overload errors
9	Correct bus message counter

The diagnostic counters are reset to zero automatically each time the communication modules start or when they reach their maximum value of 65535.

A manual reset of the counters can be performed using the function H’08 (0x08), sub-function 0x000A: Clear Counters and Diagnostic Register.

Hager-specific time synchronisation function

Function code	Name	Description
0x41	Time synchronisation	Synchronisation of data and time in broadcast mode

The function uses the difference in seconds between the synchronisation date and 1 January 2000.

N.B.

Broadcast mode allows the master to address all slaves by using the address 0. Slaves do not respond to broadcast messages.

It is only used in Modbus RTU.

Example of a complete message frame

In this example we synchronise on 26 January 2023 5:46 pm 36 seconds,
which is 728,135,196 seconds since 1 January 2000.

Frame description	Value example
Address of the slave	0x00
Function code	0x41
Length of the data in bytes	0x06
Value of the 1st byte	0x16
Value of the 2nd byte	0x64
Value of the 3rd byte	0x2B
Value of the 4th byte	0x66
Value of the 5th byte	0x76
Value of the 6th byte	0x1C
CRC (MSB)	0xXX
CRC (LSB)	0xXX

The exception responses from the master (client) or a slave (server) may be the result of data-processing errors. One of the following events can occur after a request from the master (client):
- If the slave (server) receives the request without a communication error and manages the request correctly, it sends back a normal response.
- If the slave (server) does not receive the request due to a communication error and manages the request correctly, it sends back no response.
- If the slave (server) does not receive the request but detects a communication error, it does not send back a response.
- If the slave (server) receives the request without communication error but does not process it correctly (for example, the request consists of reading a register that does not exist), it returns an exception response specifying the nature of the error.

The exception response frame consists of the following fields:

Field	Definition	Size	Description
1	Number of the slave	1 byte	Address between 1 and 247
2	Exception function code	1 byte	Request function code + 128 (0x80)
3	Exception code	n bytes	See list of exception codes
4	CRC search errors	2 bytes	CRC16 (to verify the correct transmission of the message content)

Fields 2 and 3 are different from a normal response frame.

List of exception codes managed by the communication modules

Exception code	Name	Description
01	Illegal function	The server does not perform the function requested.
02	Illegal data access	The address targeted by the request is not managed by the server.
03	Illegal data value	The value sent in a request targeting data does not correspond to the format of this data. For example, the definition of a current threshold is rejected if the value is too high.

Example of reading the hold register with exception code response

This example shows the reading of non-existent registers 5312 and 5313. The Modbus address of the slave is 14 = 0x0E.

Request from the master or client

Message field	Value example
Address of the slave	0x0E
Function code	0x03
Address of the register to be read (MSB)	0x14
Address of the register to be read (LSB)	0xC0
Number of registers (1st byte)	0x00
Number of registers (2nd bytes)	0x02
CRC (MSB)	0xXX
CRC (LSB)	0xXX

Response of the slave or the server

Message field	Value example
Address of the slave	0x0E
Exception function code	0x81
Exception code	0x02
CRC (MSB)	0xXX
CRC (LSB)	0xXX

The sentinel Energy Modbus table consists of the 6 sheets:

- Identification
- Metering
- Indicators
- Configuration
- Trip unit commands
- Communication

The sentinel Energy Modbus table of registers is available for download at the address https://hgr.io/r/HW1E416FE.

Identification

Identification information of the trip unit and display.

Description

Description	Unit	Resolution	Address DEC	Address HEX	Length (word)	Data Type	Function	Further information

Description: name of the register or the command.
Unit: information measurement unit.
Resolution: resolution of the measurement unit.
Address DEC: a 16-bit register address in the form of a decimal number. The address corresponds to the data used in the Modbus frame.
Address HEX: a 16-bit register address in the form of a hexadecimal number. The address corresponds to the data used in the Modbus frame.
Length (word): length of the information in the number of bytes.
Data Type : type of data encoding: U16, U32, U64, S32 or STRING.
Function: Modbus function code.
Further information: additional explanation or example.

Metering

Description

Description	Unit	Resolution	Address DEC	Address HEX	Length (word)	Data Type	Function	Further information

Description: name of the register or the command.
Unit: information measurement unit.
Resolution: resolution of the measurement unit.
Address DEC: a 16-bit register address in the form of a decimal number. The address corresponds to the data used in the Modbus frame.
Address HEX: a 16-bit register address in the form of a hexadecimal number. The address corresponds to the data used in the Modbus frame.
Length (word): length of the information in the number of bytes.
Data Type : type of data encoding: U16, U32, U64, S32 or STRING.
Function: Modbus function code.
Further information: additional explanation or example.

Indicators

Description

Description	Unit	Resolution	Address DEC	Address HEX	Length (word)	Data Type	Function	Further information

Description: name of the register or the command.
Unit: information measurement unit.
Resolution: resolution of the measurement unit.
Address DEC: a 16-bit register address in the form of a hexadecimal number. The address corresponds to the data used in the Modbus frame.
Address HEX: a 16-bit register address in the form of a decimal number. The address corresponds to the data used in the Modbus frame.
Length (word): length of the information in the number of bytes.
Data Type : type of data encoding: U16, U32, U64, S32 or STRING.
Function: Modbus function code.
Further information: additional explanation or example.

Configuration

Description

Description	Unit	Resolution	Address DEC	Address HEX	Length (word)	Data Type	Function	R/W access	Secure write level	Command ID	Index	Command Length (word)	Further information

Description: name of the register or the command.
Unit: information measurement unit.
Resolution: resolution of the measurement unit.
Address DEC: a 16-bit register address in the form of a hexadecimal number. The address corresponds to the data used in the Modbus frame.
Address HEX: a 16-bit register address in the form of a decimal number. The address corresponds to the data used in the Modbus frame.
Length (word): length of the information in the number of bytes.
Data Type : type of data encoding: U16, U32, U64, S32 or STRING.
Function: Modbus function code.
R/W access: read/write access of the register.
Secure write level: secure write level of the register.
Command ID: identifier of command associated with the register.
Index : additional parameter of the command identifier.
Command length (word): command data length in bytes.
Further information: additional explanation or example.

Trip unit commands

Description

Description	Secure write level	Command ID	Index	Command Length (word)	Further information

Description: name of the register or the command.
Secure write level: secure write level of the register.
Command ID: identifier of command associated with the register.
Index: additional parameter of the command identifier.
Command length (word): command data length in bytes.
Further information: additional explanation or example.

Communication

Information on the communication errors.
Description

Description	Address DEC	Address HEX	Length (word)	Data Type	Function	Further information

Description: name of the register or the command.
Address DEC: a 16-bit register address in the form of a decimal number. The address corresponds to the data used in the Modbus frame.
Address HEX: a 16-bit register address in the form of a hexadecimal number. The address corresponds to the data used in the Modbus frame.
Length (word): length of the information in the number of bytes.
Data Type : type of data encoding: U16, U32, U64, S32 or STRING.
Function: Modbus function code.
Further information: additional explanation or example.

> **⚠ WARNING**
>
> **Risk of nuisance tripping or tripping failure**
> Remote modifications made to the Modbus registers may be dangerous for personnel near the circuit breaker or may cause damage to the equipment if the protection parameters are modified. For this reason remote control commands are protected by a password and the configuration of the sentinel Energy trip unit.

To prevent unintentional changes to the configuration of the sentinel Energy trip unit, remote modifications to the Modbus registers are protected by a remote write prohibition by factory settings.
In addition, the secure write command procedure must be followed to change Modbus registers remotely. This procedure uses the secure write level of the Modbus register to be changed.
If the conditions for the procedure are not met, an error code is generated and the operation is not executed.

The remote write prohibition must be removed by changing the corresponding parameter from the sentinel Energy trip unit or the Hager Power setup software.
See the hw+ sentinel Energy electronic trip unit user manual for more information on the configuration of the remote write authorisation.

Passwords

Registers with write access have a secure write level. This determines whether or not a Modbus password will be required.

Secure write level	Password
0	No password required
1	Password level 1
2	Password level 2

Modbus password	Default password
Password level 1	La1v%el1
Password level 2	La1v%el2

Password management

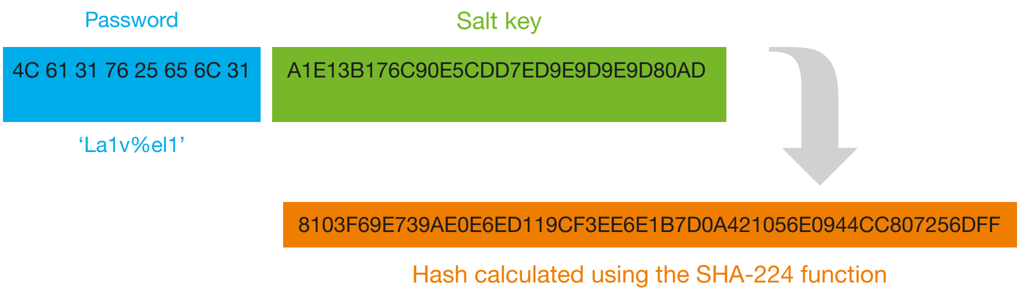
Modbus passwords can be changed using the Hager Power setup software.

Command procedure

The secure write command procedure uses an SHA-224 hash function of the password associated with the secure write level. See a more detailed explanation of this function in Appendix 4.3.

For each secure write procedure the communication module generates a salt key that must be used to calculate the password hash according to the SHA-224 hash function.

Example of password hash calculation.



The secure write command procedure is a 3-step sequence.

Step 1: request for the salt key
1.1: It consists of using the Modbus function H'16 (0x10), write multiple registers, to pass the command 101 for a salt key request to the address 0x3000.
1.2: The response to the salt key request is provided by reading the register 3200 using the function H'03 (0x03), Read holding registers.

N.B.
Step 1 is only necessary for a secure write command with a security level 1 or 2 password level.

Step 2: Implementation of the secure write command
2.1: It consists of using the Modbus function H'16 (0x10), write multiple registers, to pass the command associated with the target registers.
The hash of the password concerned must be calculated in advance using the salt key obtained at step 1 and the SHA-224 hash function.
Writing is done to address 0x3000.
The command is executed if the key provided by the client matches the one calculated by the server.

2.2: The response to the command is provided by reading the register 3200 using the function H'03 (0x03), Read holding registers.

Step 3: Reading the changed parameter(s)
This step consists of verifying that the parameters have been changed correctly by using the function H'0 (0x03), Read holding registers.

ATTENTION
The sequence of steps 1 and 2 of the procedure must be performed in 30 seconds, the maximum validity period of a salt key.

This example shows how to change the power sign parameter which is in register 6915.

Extract from the table of registers 6LE009231A

Description	Address DEC	Address HEX	Length (word)	Data Type	Function	R/W access	Secure write level	Command ID	Index	Command length (word)	Further information
Power sign convention	6915	1B03	1	U16	H'16 ; H'03	R/W	1	1002	-	2	Hex 00 00 = "+", Hex 00 01 = "-"

Step 1.1- Write request for the salt key

Here is what to write to the registers using the Modbus function H'16 (0x10) to obtain the salt key. The write is performed in registers 0x3000 to 0x3011.

Message write field	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 101 associated with this request
Register 1 (LSB, 2nd byte)	0x65	
Register 2 (MSB, 1st byte)	0x00	Security level 0 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x00	Value 0 in addresses 0x3002 to 0x300F
Register 3 (MSB, 1st byte)	0x00	
Register 3 (LSB, 2nd byte)	0x00	
Register 4 (MSB, 1st byte)	0x00	
Register 4 (LSB, 2nd byte)	0x00	
Register 5 (MSB, 1st byte)	0x00	
Register 5 (LSB, 2nd byte)	0x00	
Register 6 (MSB, 1st byte)	0x00	
Register 6 (LSB, 2nd byte)	0x00	
Register 7 (MSB, 1st byte)	0x00	
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	
Register 8 (LSB, 2nd byte)	0x00	
Register 9 (MSB, 1st byte)	0x00	
Register 9 (LSB, 2nd byte)	0x00	
Register 10 (MSB, 1st byte)	0x00	
Register 10 (LSB, 2nd byte)	0x00	
Register 11 (MSB, 1st byte)	0x00	
Register 11 (LSB, 2nd byte)	0x00	
Register 12 (MSB, 1st byte)	0x00	
Register 12 (LSB, 2nd byte)	0x00	
Register 13 (MSB, 1st byte)	0x00	
Register 13 (LSB, 2nd byte)	0x00	
Register 14 (MSB, 1st byte)	0x00	
Register 14 (LSB, 2nd byte)	0x00	
Register 15 (MSB, 1st byte)	0x00	
Register 15 (LSB, 2nd byte)	0x00	
Register 16 (MSB, 1st byte)	0x00	
Register 16 (LSB, 2nd byte)	0x00	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x02	
Register 18 (MSB, 1st byte)	0x00	Security level of the power sign parameter
Register 18 (LSB, 2nd byte)	0x01	

Step 1.2 - Reading the salt key

Here is an example of the function H’03 (0x03) to read from the register 0x3200.

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 101 associated for this request in the register 0x3200
Register 1 (LSB, 2nd byte)	0x65	
Register 2 (MSB, 1st byte)	0x00	Function code status, zero value for success
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length in number of bytes
Register 3 (LSB, 2nd byte)	0x10	
Register 4 (MSB, 1st byte)	0xA1	Salt key returned on 8 registers from 0x3203 to 0x320A
Register 4 (LSB, 2nd byte)	0xE1	
Register 5 (MSB, 1st byte)	0x3B	
Register 5 (LSB, 2nd byte)	0x17	
Register 6 (MSB, 1st byte)	0x6C	
Register 6 (LSB, 2nd byte)	0x90	
Register 7 (MSB, 1st byte)	0xE5	
Register 7 (LSB, 2nd byte)	0xCD	
Register 8 (MSB, 1st byte)	0XD7	
Register 8 (LSB, 2nd byte)	0xED	
Register 9 (MSB, 1st byte)	0x9E	
Register 9 (LSB, 2nd byte)	0x9D	
Register 10 (MSB, 1st byte)	0x3C	
Register 10 (LSB, 2nd byte)	0x23	
Register 11 (MSB, 1st byte)	0x80	
Register 11 (LSB, 2nd byte)	0xAD	

The salt key is A1 E1 3B 17 6C 90 E5 CD D7 ED 9E 9D 3C 23 80 AD.

Step 2.1 - Implementation of the secure write command

Here is the content of the registers 0x3000 to 0x3011 written using the Modbus function H'16 (0x10). The aim is to change the power sign parameter to "positive".

Message field	Value	Comment
Register 1 (MSB, 1st byte)	0x03	Identifier of command 1002 associated with the power sign parameter
Register 1 (LSB, 2nd byte)	0xEA	
Register 2 (MSB, 1st byte)	0x00	Security level 1 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x01	
Register 3 (MSB, 1st byte)	0x7F	Write of the level 1 password hash calculated using the SHA-224 function: 7F 73 59 79 5C A7 9D 35 54 2E 7F 86 1C C7 0C B6 EE E1 A1 68 DA 92 57 DF 42 F3 56 0B.
Register 3 (LSB, 2nd byte)	0x73	
Register 4 (MSB, 1st byte)	0x59	
Register 4 (LSB, 2nd byte)	0x79	
Register 5 (MSB, 1st byte)	0x5C	
Register 5 (LSB, 2nd byte)	0xA7	
Register 6 (MSB, 1st byte)	0x9D	
Register 6 (LSB, 2nd byte)	0x35	
Register 7 (MSB, 1st byte)	0x54	
Register 7 (LSB, 2nd byte)	0x2E	
Register 8 (MSB, 1st byte)	0x7F	
Register 8 (LSB, 2nd byte)	0x86	
Register 9 (MSB, 1st byte)	0x1C	
Register 9 (LSB, 2nd byte)	0xC7	
Register 10 (MSB, 1st byte)	0x0C	
Register 10 (LSB, 2nd byte)	0xB6	
Register 11 (MSB, 1st byte)	0xEE	
Register 11 (LSB, 2nd byte)	0xE1	
Register 12 (MSB, 1st byte)	0xA1	
Register 12 (LSB, 2nd byte)	0x68	
Register 13 (MSB, 1st byte)	0xDA	
Register 13 (LSB, 2nd byte)	0x92	
Register 14 (MSB, 1st byte)	0x57	
Register 14 (LSB, 2nd byte)	0xDF	
Register 15 (MSB, 1st byte)	0x42	
Register 15 (LSB, 2nd byte)	0xF3	
Register 16 (MSB, 1st byte)	0x56	
Register 16 (LSB, 2nd byte)	0x0B	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x02	
Register 18 (MSB, 1st byte)	0x00	Value of the power sign parameter: "+" = 0
Register 18 (LSB, 2nd byte)	0x00	

Step 2.2 - Reading the client response

Here are the registers to be read with the holding register H'03 read request.

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x03	Identifier of command 1002 associated with the power sign parameter
Register 1 (LSB, 2nd byte)	0xEA	
Register 2 (MSB, 1st byte)	0x00	Success code = 0
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length of the data in bytes
Register 3 (LSB, 2nd byte)	0x02	
Register 4 (MSB, 1st byte)	0x00	Value of the power sign parameter: "+" = 0
Register 4 (LSB, 2nd byte)	0x00	

Step 3: Reading the changed parameter(s)

Here are the message fields to read the power sign parameter in register 6915 using the function H'0 (0x03), Read holding registers.

Request from the master or client

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x03	Holding register read function code
Address of the register to be read (MSB)	0x1B	Register 6915
Address of the register to be read (LSB)	0x03	
Number of registers (1st byte)	0x00	-
Number of registers (2nd bytes)	0x01	
CRC (MSB)	0xXX	-
CRC (LSB)	0xXX	

Response of the slave or the server

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x03	Holding register read function code
Length of the data in bytes	0x02	-
Value of register 1 (MSB, 1st byte)	0x00	Value of the power sign parameter: "+" = 0
Value of register 1 (LSB, 2nd byte)	0x00	
CRC (MSB)	0xXX	-
CRC (LSB)	0xXX	

The Tariff Meter Switching control function is used to switch the metering order of multitariff energy meters.

The sentinel Energy trip unit makes it possible to manage up to 8 tariff slots for energy metering.

Control of more than 2 tariffs is only possible via Modbus communication.

The tariff meters first have to be activated in one of the following ways:
- from the sentinel Energy trip unit or
- from the Hager Power setup software
- by secure write command.

See the hw+ sentinel Energy electronic trip unit user manual for more information on the configuration of tariff meters.

The count order can then be given consecutively to each of the 8 groups of meters.

The activation parameter of the tariff meters is in register 6964.

The command (Command ID 4) consists of writing the number of the desired tariff slot in register 6965.

Extract from the table of registers 6LE009231A

Description	Address DEC	Address HEX	Length (word)	Data Type	Function	R/W access	Secure write level	Command ID	Index	Command length (word)	Further information
Tariff enable	6964	1B34	1	U16	H'16 ; H'03	R/W	1	1009	-	2	Hex 00 00 = disable, Hex 00 01 = enable
Tariff slot counter index	6965	1B35	1	U16	H'16 ; H'03	R/W	0	4	-	2	Hex 00 01 = T1, Hex 00 02 = T2, Hex 00 03 = T3, Hex 00 04 = T4, Hex 00 05 = T5, Hex 00 06 = T6, Hex 00 07 = T7, Hex 00 08 = T8

Here is an example of sending the count order to tariff meter T4.

The secure write command does not require a salt key request because the parameter "Tariff slot counter index" is of security level 0.

Step 2.1 - Implementation of the secure write command

Here is the content of the registers 0x3000 to 0x3011 written using the Modbus function H’16 (0x10). The aim is to pass the parameter "Tariff slot counter index" to "T4".

Message field	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 4 associated with the parameter
Register 1 (LSB, 2nd byte)	0x04	
Register 2 (MSB, 1st byte)	0x00	Security level 0 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Write of the level 0 password hash
Register 3 (LSB, 2nd byte)	0x00	
Register 4 (MSB, 1st byte)	0x00	
Register 4 (LSB, 2nd byte)	0x00	
Register 5 (MSB, 1st byte)	0x00	
Register 5 (LSB, 2nd byte)	0x00	
Register 6 (MSB, 1st byte)	0x00	
Register 6 (LSB, 2nd byte)	0x00	
Register 7 (MSB, 1st byte)	0x00	
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	
Register 8 (LSB, 2nd byte)	0x00	
Register 9 (MSB, 1st byte)	0x00	
Register 9 (LSB, 2nd byte)	0x00	
Register 10 (MSB, 1st byte)	0x00	
Register 10 (LSB, 2nd byte)	0x00	
Register 11 (MSB, 1st byte)	0x00	
Register 11 (LSB, 2nd byte)	0x00	
Register 12 (MSB, 1st byte)	0x00	
Register 12 (LSB, 2nd byte)	0x00	
Register 13 (MSB, 1st byte)	0x00	
Register 13 (LSB, 2nd byte)	0x00	
Register 14 (MSB, 1st byte)	0x00	
Register 14 (LSB, 2nd byte)	0x00	
Register 15 (MSB, 1st byte)	0x00	
Register 15 (LSB, 2nd byte)	0x00	
Register 16 (MSB, 1st byte)	0x00	
Register 16 (LSB, 2nd byte)	0x00	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x01	
Register 18 (MSB, 1st byte)	0x00	Parameter value: T4 = 0x0004
Register 18 (LSB, 2nd byte)	0x04	

Step 2.2 - Reading the client response

Here are the registers to be read with the holding register H’03 read request.

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 4 associated with the parameter
Register 1 (LSB, 2nd byte)	0x04	
Register 2 (MSB, 1st byte)	0x00	Success code = 0
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length of the data in bytes
Register 3 (LSB, 2nd byte)	0x02	
Register 4 (MSB, 1st byte)	0x00	Parameter value: T4 = 0x0004
Register 4 (LSB, 2nd byte)	0x04	

Step 3: Reading the changed parameter(s)

Here are the message fields to read the parameter "Tariff slot counter index" in register 6965 using the function H'0 (0x03), Read holding registers.

Request from the master or client

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x03	Holding register read function code
Address of the register to be read (MSB)	0x1B	Register 6965
Address of the register to be read (LSB)	0x35	
Number of registers (1st byte)	0x00	-
Number of registers (2nd bytes)	0x01	
CRC (MSB)	0xXX	-
CRC (LSB)	0xXX	

Response of the slave or the server

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x03	Holding register read function code
Length of the data in bytes	0x02	-
Value of register 1 (MSB, 1st byte)	0x00	Tariff T4 = 0x0004
Value of register 1 (LSB, 2nd byte)	0x04	
CRC (MSB)	0xXX	-
CRC (LSB)	0xXX	

The Inhibition control function momentarily deactivates the effect of advanced protection.

Only advanced protections with the INHIBIT parameter enabled will receive the inhibition command.

The advanced protections must previously have been configured and the INHIBIT parameter of the protections in question must have been activated. Configuration can be done in one of the following ways:
- from the sentinel Energy trip unit,
- from the Hager Power setup software,
- by secure write command.

See the hw+ sentinel Energy electronic trip unit user manual for more information on the configuration of advanced protections.

The inhibition command consists of writing in register 7727.

Extract from the table of registers 6LE009231A

Description	Address DEC	Address HEX	Length (word)	Data Type	Function	R/W access	Secure write level	Command ID	Index	Command length (word)	Further information
Inhibition command	7727	1E2F	1	U16	H'16 ; H'03	R/W	1	1307	-	2	Hex 00 00 = not inhibited, Hex 00 01 = inhibited

Here is an example of sending activation of the Inhibition command for advanced protections.

Step 1.1- Write request for the salt key

Here are the registers to write using the Modbus function H’16 (0x10) to obtain the salt key. The write is performed in registers 0x3000 to 0x3011.

Message field	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 101 associated with this request
Register 1 (LSB, 2nd byte)	0x65	
Register 2 (MSB, 1st byte)	0x00	Security level 0 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x00	Value 0 in addresses 0x3002 to 0x300F
Register 3 (MSB, 1st byte)	0x00	
Register 3 (LSB, 2nd byte)	0x00	
Register 4 (MSB, 1st byte)	0x00	
Register 4 (LSB, 2nd byte)	0x00	
Register 5 (MSB, 1st byte)	0x00	
Register 5 (LSB, 2nd byte)	0x00	
Register 6 (MSB, 1st byte)	0x00	
Register 6 (LSB, 2nd byte)	0x00	
Register 7 (MSB, 1st byte)	0x00	
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	
Register 8 (LSB, 2nd byte)	0x00	
Register 9 (MSB, 1st byte)	0x00	
Register 9 (LSB, 2nd byte)	0x00	
Register 10 (MSB, 1st byte)	0x00	
Register 10 (LSB, 2nd byte)	0x00	
Register 11 (MSB, 1st byte)	0x00	
Register 11 (LSB, 2nd byte)	0x00	
Register 12 (MSB, 1st byte)	0x00	
Register 12 (LSB, 2nd byte)	0x00	
Register 13 (MSB, 1st byte)	0x00	
Register 13 (LSB, 2nd byte)	0x00	
Register 14 (MSB, 1st byte)	0x00	
Register 14 (LSB, 2nd byte)	0x00	
Register 15 (MSB, 1st byte)	0x00	
Register 15 (LSB, 2nd byte)	0x00	
Register 16 (MSB, 1st byte)	0x00	
Register 16 (LSB, 2nd byte)	0x00	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x02	
Register 18 (MSB, 1st byte)	0x00	Register security level 7727: 1
Register 18 (LSB, 2nd byte)	0x01	

Step 1.2 - Reading the salt key

Here is an example of the response to be read using the function H’03 (0x03) from the register 0x3200

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 101 associated for this request in the register 0x3200
Register 1 (LSB, 2nd byte)	0x65	
Register 2 (MSB, 1st byte)	0x00	Function code status, zero value for success
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length in number of bytes
Register 3 (LSB, 2nd byte)	0x10	
Register 4 (MSB, 1st byte)	0xA1	Salt key returned on 8 registers from 0x3203 to 0x320A
Register 4 (LSB, 2nd byte)	0xE1	
Register 5 (MSB, 1st byte)	0x3B	
Register 5 (LSB, 2nd byte)	0x17	
Register 6 (MSB, 1st byte)	0x6C	
Register 6 (LSB, 2nd byte)	0x90	
Register 7 (MSB, 1st byte)	0xE5	
Register 7 (LSB, 2nd byte)	0xCD	
Register 8 (MSB, 1st byte)	0XD7	
Register 8 (LSB, 2nd byte)	0xED	
Register 9 (MSB, 1st byte)	0x9E	
Register 9 (LSB, 2nd byte)	0x9D	
Register 10 (MSB, 1st byte)	0x3C	
Register 10 (LSB, 2nd byte)	0x23	
Register 11 (MSB, 1st byte)	0x80	
Register 11 (LSB, 2nd byte)	0xAD	

The salt key is A1 E1 3B 17 6C 90 E5 CD D7 ED 9E 9D 3C 23 80 AD.

Step 2.1 - Implementation of the secure write command

Here is the content of the registers 0x3000 to 0x3011 written using the Modbus function H'16 (0x10). The aim is to switch the parameter "Inhibition command" to 1.

Message field	Value	Comment
Register 1 (MSB, 1st byte)	0x05	Identifier of command 1307 associated with the parameter
Register 1 (LSB, 2nd byte)	0x1B	
Register 2 (MSB, 1st byte)	0x00	Security level 1 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x01	
Register 3 (MSB, 1st byte)	0x7F	Write of the level 1 password hash calculated using the SHA-224 function: 7F 73 59 79 5C A7 9D 35 54 2E 7F 86 1C C7 0C B6 EE E1 A1 68 DA 92 57 DF 42 F3 56 0B.
Register 3 (LSB, 2nd byte)	0x73	
Register 4 (MSB, 1st byte)	0x59	
Register 4 (LSB, 2nd byte)	0x79	
Register 5 (MSB, 1st byte)	0x5C	
Register 5 (LSB, 2nd byte)	0xA7	
Register 6 (MSB, 1st byte)	0x9D	
Register 6 (LSB, 2nd byte)	0x35	
Register 7 (MSB, 1st byte)	0x54	
Register 7 (LSB, 2nd byte)	0x2E	
Register 8 (MSB, 1st byte)	0x7F	
Register 8 (LSB, 2nd byte)	0x86	
Register 9 (MSB, 1st byte)	0x1C	
Register 9 (LSB, 2nd byte)	0xC7	
Register 10 (MSB, 1st byte)	0x0C	
Register 10 (LSB, 2nd byte)	0xB6	
Register 11 (MSB, 1st byte)	0xEE	
Register 11 (LSB, 2nd byte)	0xE1	
Register 12 (MSB, 1st byte)	0xA1	
Register 12 (LSB, 2nd byte)	0x68	
Register 13 (MSB, 1st byte)	0xDA	
Register 13 (LSB, 2nd byte)	0x92	
Register 14 (MSB, 1st byte)	0x57	
Register 14 (LSB, 2nd byte)	0xDF	
Register 15 (MSB, 1st byte)	0x42	
Register 15 (LSB, 2nd byte)	0xF3	
Register 16 (MSB, 1st byte)	0x56	
Register 16 (LSB, 2nd byte)	0x0B	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x02	
Register 18 (MSB, 1st byte)	0x00	Parameter value: 0x0001
Register 18 (LSB, 2nd byte)	0x01	

Step 2.2: Reading the changed parameter(s)

Here are the message fields to read the parameter "Inhibit command" in register 7727 using the function H'03 (0x03), Read holding registers.

Request from the master or client

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x03	Holding register read function code
Address of the register to be read (MSB)	0x1E	Register 7727
Address of the register to be read (LSB)	0x2F	
Number of registers (1st byte)	0x00	-
Number of registers (2nd bytes)	0x01	
CRC (MSB)	0xXX	-
CRC (LSB)	0xXX	

Response of the slave or the server

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x03	Holding register read function code
Length of the data in bytes	0x02	-
Value of register 1 (MSB, 1st byte)	0x00	"Inhibition command" = 0x0001
Value of register 1 (LSB, 2nd byte)	0x01	
CRC (MSB)	0xXX	-
CRC (LSB)	0xXX	

WARNING

Risk of false tripping or trip failure
The inhibition command is intended to temporarily deactivate the advanced protections while test, maintenance or repair interventions are taking place.
It is essential to reset the inhibition control after the intervention has been carried out to restore correct operation of the circuit-breaker.

The Switch control functions enables the trip unit's operation to be switched between protection profile A and protection profile B. It is available after dual settings are activated and set

See the hw+ sentinel Energy electronic trip unit user manual for information on how to activate and configure the Dual settings.

Activation and configuration can be done in one of the following ways:
- from the sentinel Energy trip unit or
- from the Hager Power setup software
- by secure write command.

The command to switch between protection profile A and protection profile B consists of writing the profile to switch to into register 7681.

Extract from the table of registers 6LE009231A

Description	Address DEC	Address HEX	Length (word)	Data Type	Function	R/W access	Secure write level	Command ID	Index	Command length (word)	Further information
Dual protection setting active profile (A or B)	7681	1E01	1	U16	H'16 ; H'03	R/W	2	2002	-	2	Hex 00 00 = profile A, Hex 00 01 = profile B

Here is an example of switching to profile B.

Step 1.1- Write request for the salt key

Here are the registers to read using the Modbus function H'16 (0x10) to obtain the salt key. The write is performed in registers 0x3000 to 0x3011.

Message field	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 101 associated with this request
Register 1 (LSB, 2nd byte)	0x65	
Register 2 (MSB, 1st byte)	0x00	Security level 0 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x00	Value 0 in addresses 0x3002 to 0x300F
Register 3 (MSB, 1st byte)	0x00	
Register 3 (LSB, 2nd byte)	0x00	
Register 4 (MSB, 1st byte)	0x00	
Register 4 (LSB, 2nd byte)	0x00	
Register 5 (MSB, 1st byte)	0x00	
Register 5 (LSB, 2nd byte)	0x00	
Register 6 (MSB, 1st byte)	0x00	
Register 6 (LSB, 2nd byte)	0x00	
Register 7 (MSB, 1st byte)	0x00	
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	
Register 8 (LSB, 2nd byte)	0x00	
Register 9 (MSB, 1st byte)	0x00	
Register 9 (LSB, 2nd byte)	0x00	
Register 10 (MSB, 1st byte)	0x00	
Register 10 (LSB, 2nd byte)	0x00	
Register 11 (MSB, 1st byte)	0x00	
Register 11 (LSB, 2nd byte)	0x00	
Register 12 (MSB, 1st byte)	0x00	
Register 12 (LSB, 2nd byte)	0x00	
Register 13 (MSB, 1st byte)	0x00	
Register 13 (LSB, 2nd byte)	0x00	
Register 14 (MSB, 1st byte)	0x00	
Register 14 (LSB, 2nd byte)	0x00	
Register 15 (MSB, 1st byte)	0x00	
Register 15 (LSB, 2nd byte)	0x00	
Register 16 (MSB, 1st byte)	0x00	
Register 16 (LSB, 2nd byte)	0x00	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x02	
Register 18 (MSB, 1st byte)	0x00	Switching parameter security level
Register 18 (LSB, 2nd byte)	0x02	

Step 1.2 - Reading the salt key

Here is an example of the function H’03 (0x03) to read from the register 0x3200.

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 101 associated for this request in the register 0x3200
Register 1 (LSB, 2nd byte)	0x65	
Register 2 (MSB, 1st byte)	0x00	Function code status, zero value for success
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length in number of bytes
Register 3 (LSB, 2nd byte)	0x10	
Register 4 (MSB, 1st byte)	0xA1	Salt key returned on 8 registers from 0x3203 to 0x320A
Register 4 (LSB, 2nd byte)	0xE1	
Register 5 (MSB, 1st byte)	0x3B	
Register 5 (LSB, 2nd byte)	0x17	
Register 6 (MSB, 1st byte)	0x6C	
Register 6 (LSB, 2nd byte)	0x90	
Register 7 (MSB, 1st byte)	0xE5	
Register 7 (LSB, 2nd byte)	0xCD	
Register 8 (MSB, 1st byte)	0XD7	
Register 8 (LSB, 2nd byte)	0xED	
Register 9 (MSB, 1st byte)	0x9E	
Register 9 (LSB, 2nd byte)	0x9D	
Register 10 (MSB, 1st byte)	0x3C	
Register 10 (LSB, 2nd byte)	0x23	
Register 11 (MSB, 1st byte)	0x80	
Register 11 (LSB, 2nd byte)	0xAD	

The salt key is A1 E1 3B 17 6C 90 E5 CD D7 ED 9E 9D 3C 23 80 AD.

Step 2.1 - Implementation of the secure write command

Here is the content of the registers 0x3000 to 0x3011 written using the Modbus function H'16 (0x10). The aim is to switch the parameter "Dual protection setting active profile" to profile B.

Message e field	Value	Comment
Register 1 (MSB, 1st byte)	0x07	Identifier of command 2002 associated with the parameter
Register 1 (LSB, 2nd byte)	0xD2	
Register 2 (MSB, 1st byte)	0x00	Security level 2 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x02	
Register 3 (MSB, 1st byte)	0x90	Write of the level 2 password hash calculated using the SHA-224 function: 90 0D 09 D0 B6 86 AE CE 90 36 44 35 0E 34 BC 7D 7C 61 E5 AA DC 8B E6 1E F6 81 9D 65.
Register 3 (LSB, 2nd byte)	0x0D	
Register 4 (MSB, 1st byte)	0x09	
Register 4 (LSB, 2nd byte)	0xD0	
Register 5 (MSB, 1st byte)	0xB6	
Register 5 (LSB, 2nd byte)	0x86	
Register 6 (MSB, 1st byte)	0xAE	
Register 6 (LSB, 2nd byte)	0xCE	
Register 7 (MSB, 1st byte)	0x90	
Register 7 (LSB, 2nd byte)	0x36	
Register 8 (MSB, 1st byte)	0x44	
Register 8 (LSB, 2nd byte)	0x35	
Register 9 (MSB, 1st byte)	0x0E	
Register 9 (LSB, 2nd byte)	0x34	
Register 10 (MSB, 1st byte)	0xBC	
Register 10 (LSB, 2nd byte)	0x7D	
Register 11 (MSB, 1st byte)	0x7C	
Register 11 (LSB, 2nd byte)	0x61	
Register 12 (MSB, 1st byte)	0xE5	
Register 12 (LSB, 2nd byte)	0xAA	
Register 13 (MSB, 1st byte)	0xDC	
Register 13 (LSB, 2nd byte)	0x8B	
Register 14 (MSB, 1st byte)	0xE6	
Register 14 (LSB, 2nd byte)	0x1E	
Register 15 (MSB, 1st byte)	0xF6	
Register 15 (LSB, 2nd byte)	0x81	
Register 16 (MSB, 1st byte)	0x9D	
Register 16 (LSB, 2nd byte)	0x65	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x02	
Register 18 (MSB, 1st byte)	0x00	Value of the parameter: profile B = 0x0001
Register 18 (LSB, 2nd byte)	0x01	

Step 2.2 - Reading the client response

Here are the registers to be read with the holding register H'03 read request.

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x07	Identifier of command 2002 associated with the parameter
Register 1 (LSB, 2nd byte)	0xD2	
Register 2 (MSB, 1st byte)	0x00	Success code = 0
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length of the data in bytes
Register 3 (LSB, 2nd byte)	0x02	
Register 4 (MSB, 1st byte)	0x00	Value of the parameter: profile B = 0x0001
Register 4 (LSB, 2nd byte)	0x01	

Step 3: Reading the changed parameter(s)

Here are the message fields to read the parameter "Dual protection setting active profile (A or B)" in register 7681 using the function H'03 (0x03), Read holding registers.

Request from the master or client

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x03	Holding register read function code
Address of the register to be read (MSB)	0x1E	Register 7681
Address of the register to be read (LSB)	0x01	
Number of registers (1st byte)	0x00	-
Number of registers (2nd bytes)	0x01	
CRC (MSB)	0xXX	-
CRC (LSB)	0xXX	

Response of the slave or the server

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x03	Holding register read function code
Length of the data in bytes	0x02	-
Value of register 1 (MSB, 1st byte)	0x00	Value of the parameter: profile B = 0x0001
Value of register 1 (LSB, 2nd byte)	0x01	
CRC (MSB)	0xXX	-
CRC (LSB)	0xXX	

These commands allow the circuit breaker to be opened or closed remotely.

They require the following accessories to have been installed first:
- an SH shunt trip coil for the opening operation,
- a CC closing coil for the closing operation,
- an INS insulation module,
- an MO charging Motor for automatic reload of the closing spring.

These commands are accessible in any of the following ways:
- from the Hager Power touch application,
- from the Hager Power setup software,
- by secure write command.

These commands are embedded in the sentinel Energy trip unit.

Extract from the table of registers 6LE009231A

Description	**Secure write level**	**Command ID**	**Index**	**Command length (word)**	**Further information**
Opening operation command (drive the SH coil + INS insulation module)	1	215	-	0	SH coil and INS insulation module need to installed to use command
Closing operation command (drive the CC coil + INS insulation module)	1	216	-	0	CC coil and INS insulation module need to installed to use command

Here is an example opening command.

Step 1.1- Write request for the salt key

Here are the registers to write using the Modbus function H'16 (0x10) to obtain the salt key. The write is performed in registers 0x3000 to 0x3011.

Message field	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 101 associated with this request
Register 1 (LSB, 2nd byte)	0x65	
Register 2 (MSB, 1st byte)	0x00	Security level 0 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x00	Value 0 in addresses 0x3002 to 0x300F
Register 3 (MSB, 1st byte)	0x00	
Register 3 (LSB, 2nd byte)	0x00	
Register 4 (MSB, 1st byte)	0x00	
Register 4 (LSB, 2nd byte)	0x00	
Register 5 (MSB, 1st byte)	0x00	
Register 5 (LSB, 2nd byte)	0x00	
Register 6 (MSB, 1st byte)	0x00	
Register 6 (LSB, 2nd byte)	0x00	
Register 7 (MSB, 1st byte)	0x00	
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	
Register 8 (LSB, 2nd byte)	0x00	
Register 9 (MSB, 1st byte)	0x00	
Register 9 (LSB, 2nd byte)	0x00	
Register 10 (MSB, 1st byte)	0x00	
Register 10 (LSB, 2nd byte)	0x00	
Register 11 (MSB, 1st byte)	0x00	
Register 11 (LSB, 2nd byte)	0x00	
Register 12 (MSB, 1st byte)	0x00	
Register 12 (LSB, 2nd byte)	0x00	
Register 13 (MSB, 1st byte)	0x00	
Register 13 (LSB, 2nd byte)	0x00	
Register 14 (MSB, 1st byte)	0x00	
Register 14 (LSB, 2nd byte)	0x00	
Register 15 (MSB, 1st byte)	0x00	
Register 15 (LSB, 2nd byte)	0x00	
Register 16 (MSB, 1st byte)	0x00	
Register 16 (LSB, 2nd byte)	0x00	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x02	
Register 18 (MSB, 1st byte)	0x00	Parameter security level
Register 18 (LSB, 2nd byte)	0x01	

Step 1.2 - Reading the salt key

Here is an example of the response to be read using the function H’03 (0x03) from the register 0x3200

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 101 associated for this request in the register 0x3200
Register 1 (LSB, 2nd byte)	0x65	
Register 2 (MSB, 1st byte)	0x00	Function code status, zero value for success
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length in number of bytes
Register 3 (LSB, 2nd byte)	0x10	
Register 4 (MSB, 1st byte)	0xA1	Salt key returned on 8 registers from 0x3203 to 0x320A
Register 4 (LSB, 2nd byte)	0xE1	
Register 5 (MSB, 1st byte)	0x3B	
Register 5 (LSB, 2nd byte)	0x17	
Register 6 (MSB, 1st byte)	0x6C	
Register 6 (LSB, 2nd byte)	0x90	
Register 7 (MSB, 1st byte)	0xE5	
Register 7 (LSB, 2nd byte)	0xCD	
Register 8 (MSB, 1st byte)	0XD7	
Register 8 (LSB, 2nd byte)	0xED	
Register 9 (MSB, 1st byte)	0x9E	
Register 9 (LSB, 2nd byte)	0x9D	
Register 10 (MSB, 1st byte)	0x3C	
Register 10 (LSB, 2nd byte)	0x23	
Register 11 (MSB, 1st byte)	0x80	
Register 11 (LSB, 2nd byte)	0xAD	

The salt key is A1 E1 3B 17 6C 90 E5 CD D7 ED 9E 9D 3C 23 80 AD.

Step 2.1 - Implementation of the secure write command

Here is the content of the registers 0x3000 to 0x3011 written using the Modbus function H'16 (0x10). The aim is to perform the opening command.

Message write field	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 215
Register 1 (LSB, 2nd byte)	0xD7	
Register 2 (MSB, 1st byte)	0x00	Security level 1 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x01	
Register 3 (MSB, 1st byte)	0x7F	Write of the level 1 password hash calculated using the SHA-224 function: 7F 73 59 79 5C A7 9D 35 54 2E 7F 86 1C C7 0C B6 EE E1 A1 68 DA 92 57 DF 42 F3 56 0B.
Register 3 (LSB, 2nd byte)	0x73	
Register 4 (MSB, 1st byte)	0x59	
Register 4 (LSB, 2nd byte)	0x79	
Register 5 (MSB, 1st byte)	0x5C	
Register 5 (LSB, 2nd byte)	0xA7	
Register 6 (MSB, 1st byte)	0x9D	
Register 6 (LSB, 2nd byte)	0x35	
Register 7 (MSB, 1st byte)	0x54	
Register 7 (LSB, 2nd byte)	0x2E	
Register 8 (MSB, 1st byte)	0x7F	
Register 8 (LSB, 2nd byte)	0x86	
Register 9 (MSB, 1st byte)	0x1C	
Register 9 (LSB, 2nd byte)	0xC7	
Register 10 (MSB, 1st byte)	0x0C	
Register 10 (LSB, 2nd byte)	0xB6	
Register 11 (MSB, 1st byte)	0xEE	
Register 11 (LSB, 2nd byte)	0xE1	
Register 12 (MSB, 1st byte)	0xA1	
Register 12 (LSB, 2nd byte)	0x68	
Register 13 (MSB, 1st byte)	0xDA	
Register 13 (LSB, 2nd byte)	0x92	
Register 14 (MSB, 1st byte)	0x57	
Register 14 (LSB, 2nd byte)	0xDF	
Register 15 (MSB, 1st byte)	0x42	
Register 15 (LSB, 2nd byte)	0xF3	
Register 16 (MSB, 1st byte)	0x56	
Register 16 (LSB, 2nd byte)	0x0B	
Register 17 (MSB, 1st byte)	0x00	Zero value
Register 17 (LSB, 2nd byte)	0x00	

Step 2.2 - Reading the client response

Here are the registers to be read with the holding register H'03 read request.

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 215
Register 1 (LSB, 2nd byte)	0xD7	
Register 2 (MSB, 1st byte)	0x00	Success code = 0
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length of the data in bytes
Register 3 (LSB, 2nd byte)	0x02	
Register 4 (MSB, 1st byte)	0x00	Parameter value: 0
Register 4 (LSB, 2nd byte)	0x00	

Step 3: Reading the changed parameter(s)

To verify that the remote opening and closing commands have been performed correctly, the open or closed status of the circuit breaker is read in register 5376, On/Off indicator.

Request from the master or client

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x03	Holding register read function code
Address of the register to be read (MSB)	0x15	Register 5376
Address of the register to be read (LSB)	0x00	
Number of registers (1st byte)	0x00	-
Number of registers (2nd bytes)	0x01	
CRC (MSB)	0xXX	-
CRC (LSB)	0xXX	

Response of the slave or the server

Frame description	Value	Comment
Address of the slave	0x0E	-
Function code	0x03	Holding register read function code
Length of the data in bytes	0x02	-
Value of register 1 (MSB, 1st byte)	0x00	Open state = 0x0000
Value of register 1 (LSB, 2nd byte)	0x00	
CRC (MSB)	0xXX	-
CRC (LSB)	0xXX	

The sentinel Energy trip unit has several embedded commands to perform the following actions:
- perform a remote opening command,
- perform a remote closing command,
- reset the counters (OAC, PTA, On/Off, FS, min/max, optional alarms),
- reset the history sections,
- reset parameters to factory settings,
- perform bus synchronisation of the calculation interval windows of the current and power demands.

Extract from the table of registers 6LE009231A

Description	Secure write level	Command ID	Command length (word)	Further information
Reset OAC counters	1	201	0	
Reset history section: Alarm	1	202	0	
Reset history section: Diagnostic	1	203	0	
Reset history section: Error	1	204	0	
Reset history section: Operation	1	205	0	
Reset history section: Test	1	206	0	
Reset history section: Protection settings	1	207	0	
Reset history section: Metering settings	1	208	0	
Reset history section: Tripping	1	209	0	
Reset history section: Custom alarm	1	210	0	
Reset all history sections	1	211	0	
Reset PTA counter	1	212	0	
Reset user factory settings	1	213	0	
Reset On/Off and FS contact counters	1	214	0	
Opening operation command (drive the SH coil + INS insulation module)	1	215	0	SH coil and INS insulation module need to installed to use command
Closing operation command (drive the CC coil + INS insulation module)	1	216	0	CC coil and INS insulation module need to installed to use command
Reset min/max current values	1	217	0	
Reset min/max voltage values	1	218	0	
Reset min/max power values	1	219	0	
Reset min/max power factor values	1	220	0	
Reset min/max THD values	1	221	0	
Reset min/max frequency values	1	222	0	
Reset min/max energy values	1	223	0	
Reset min/max demand current values	1	224	0	
Reset min/max demand power values	1	225	0	
Reset all min/max counters	1	226	0	
Reset custom alarm counters	1	227	0	
Latch bus synchronisation for demand windows	1	228	0	

The secure write command procedure enables the use of embedded commands.

Access to event history

The events history is divided into the following sections:
- Trip
- Alarm
- Custom alarm
- Error
- Diagnostic
- Operation
- Protection setting
- Measurements setting
- Test

These events can be read via Modbus communication.
Refer to the hw+ sentinel Energy 6LE008147A electronic trip unit user manual for more information.

There are 3 ways to read the events history via Modbus communication.

Method	Description
0	Reading the 10 most recent events
1	Reading the events between 2 dates
2	Reading all events

History reading procedure

The procedure for obtaining the events is the same for each of the 3 methods.

1. Creation of a temporary view by applying the external command 111, sub-command 1 "create the view" using the Modbus function H’16 (0x10). This view must include the log section value indicating which sections are to be examined.
2. Reading the state of the command 111 indicating the number of event blocks copied into the holding registers using the Modbus function H’03 (0x03).
3. Block reading, to be iterated for each block :
   a. Request to read the content of the block by applying the external command 111, sub-command 2 "read a block" using the function Modbus H’16 (0x10). The event occurrences are copied into the dedicated registers (addresses 0x3208 and above).
   b. Reading the block in the holding registers where the events of the block considered are copied (Modbus function H’03 (0x03).
4. Deletion of the temporary view by applying the external command 111, sub-command 3 "delete the view" using the Modbus function H’16 (0x10).
5. Reading the status of the deletion of the temporary view to ensure that everything has taken place correctly even if this step is not mandatory.

ATTENTION
It is advisable to perform block read iterations without long interruptions, as the temporary view is automatically deleted after 30 seconds of inactivity on the read request. In addition, steps 1 to 3 must be performed in a maximum time of 2 minutes.

History section value

The desired history section value is written to register 0x3012 by combining the individual value of each section over 16 bits.

History section	Order number	Individual value over 16 bits
Alarm	0	0x00 01
Diagnostic	1	0x00 02
Error	2	0x00 04
Operation	3	0x00 08
Test	4	0x00 10
Protection settings	5	0x00 20
Measurements setting	6	0x00 40
Trip	7	0x00 80
Custom alarm	8	0x01 00

For example the section value to read all sections is 0x01 FF

Conversion of dates

With regard to the read method for events between 2 dates, each of the dates must be converted into timestamped time that the communication module can understand.
The conversion consists of translating the date into the number of seconds since 1st January 1970, then subtracting 946684800 seconds which corresponds to the 1st January 2000.
The dates are written into 0x3014 for the start date and 0x3016 for the end date.

For example here is the conversion to be carried out for a read between the 1st September 2021 at 1 hour 1 minute and 1 second and the 16 December 2022 at 1 hour 1 minute and 1 second.

Date	Conversion into seconds since 1970	Result after subtraction of 1st January 2000
1st September 2021 at 1 hour 1 minute and 1 second	1630458061 s	683766061 s = Hex 28 C1 71 2D
16 December 2022 at 1 hour 1 minute and 1 second	1671152461 s	724464061 s = Hex 2B 2E 71 BD

Example of history reading

The example below shows in detail how to read the history between 2 dates.

N.B.

Reading the history according to the 2 other methods follows the same sequence as this example with the difference of leaving the start date and end date registers at 0x0000.

Step 1 - Creation of a temporary view

Here is what to write to the registers using the Modbus function H’16 (0x10) to create a temporary view. The write is performed in registers 0x3000 to 0x3011.

Message field	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 111 associated with this request
Register 1 (LSB, 2nd byte)	0x6F	
Register 2 (MSB, 1st byte)	0x00	Security level 0 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x00	Password hash for level 0 = 0 in addresses 0x3002 à 0x300F
Register 3 (MSB, 1st byte)	0x00	
Register 3 (LSB, 2nd byte)	0x00	
Register 4 (MSB, 1st byte)	0x00	
Register 4 (LSB, 2nd byte)	0x00	
Register 5 (MSB, 1st byte)	0x00	
Register 5 (LSB, 2nd byte)	0x00	
Register 6 (MSB, 1st byte)	0x00	
Register 6 (LSB, 2nd byte)	0x00	
Register 7 (MSB, 1st byte)	0x00	
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	
Register 8 (LSB, 2nd byte)	0x00	
Register 9 (MSB, 1st byte)	0x00	
Register 9 (LSB, 2nd byte)	0x00	
Register 10 (MSB, 1st byte)	0x00	
Register 10 (LSB, 2nd byte)	0x00	
Register 11 (MSB, 1st byte)	0x00	
Register 11 (LSB, 2nd byte)	0x00	
Register 12 (MSB, 1st byte)	0x00	
Register 12 (LSB, 2nd byte)	0x00	
Register 13 (MSB, 1st byte)	0x00	
Register 13 (LSB, 2nd byte)	0x00	
Register 14 (MSB, 1st byte)	0x00	
Register 14 (LSB, 2nd byte)	0x00	
Register 15 (MSB, 1st byte)	0x00	
Register 15 (LSB, 2nd byte)	0x00	
Register 16 (MSB, 1st byte)	0x00	
Register 16 (LSB, 2nd byte)	0x00	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x16	
Register 18 (MSB, 1st byte)	0x00	Sub-command 1: create the view
Register 18 (LSB, 2nd byte)	0x01	
Register 19 (MSB, 1st byte)	0x01	Section value, here all sections
Register 19 (LSB, 2nd byte)	0xFF	
Register 20 (MSB, 1st byte)	0x00	Read method: 1 = between 2 dates
Register 20 (LSB, 2nd byte)	0x01	
Register 21 (MSB, 1st byte)	0x28	Start date: 1st September 2021 at 1 hour 1 minute and 1 second
Register 21 (LSB, 2nd byte)	0xC1	
Register 22 (MSB, 1st byte)	0x71	
Register 22 (LSB, 2nd byte)	0x2D	
Register 23 (MSB, 1st byte)	0x2B	End date: 16 December 2022 at 1 hour 1 minute and 1 second
Register 23 (LSB, 2nd byte)	0x2E	
Register 24 (MSB, 1st byte)	0x71	
Register 24 (LSB, 2nd byte)	0xBD	
Register 25 (MSB, 1st byte)	0x00	Block index (always 0 for the creation of the temporary view)
Register 25 (LSB, 2nd byte)	0x00	

Step 2 - Read the status of the command 111

Response to be read using the function H’03 (0x03) from the register 0x3200.

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 111 associated for this request in the register 0x3200
Register 1 (LSB, 2nd byte)	0x6F	
Register 2 (MSB, 1st byte)	0x00	Function code status, zero value for success
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length in number of bytes
Register 3 (LSB, 2nd byte)	0x10	
Register 4 (MSB, 1st byte)	0x01	Section value
Register 4 (LSB, 2nd byte)	0xFF	
Register 5 (MSB, 1st byte)	0x00	Read method: 1
Register 5 (LSB, 2nd byte)	0x01	
Register 6 (MSB, 1st byte)	0x00	Number of log blocks
Register 6 (LSB, 2nd byte)	0x02	
Register 7 (MSB, 1st byte)	0x00	Index of current block
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	Event number in the block: 0 for the creation of the temporary view
Register 8 (LSB, 2nd byte)	0x00	

Step 3.1 - Read request for the first block

Request using the Modbus function H'16 (0x10) to write in the registers 0x3000 to 0x3011.

Message field	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 111 associated with this request
Register 1 (LSB, 2nd byte)	0x6F	
Register 2 (MSB, 1st byte)	0x00	Security level 0 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x00	Password hash for level 0: 0 in addresses 0x3002 to 0x300F
Register 3 (MSB, 1st byte)	0x00	
Register 3 (LSB, 2nd byte)	0x00	
Register 4 (MSB, 1st byte)	0x00	
Register 4 (LSB, 2nd byte)	0x00	
Register 5 (MSB, 1st byte)	0x00	
Register 5 (LSB, 2nd byte)	0x00	
Register 6 (MSB, 1st byte)	0x00	
Register 6 (LSB, 2nd byte)	0x00	
Register 7 (MSB, 1st byte)	0x00	
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	
Register 8 (LSB, 2nd byte)	0x00	
Register 9 (MSB, 1st byte)	0x00	
Register 9 (LSB, 2nd byte)	0x00	
Register 10 (MSB, 1st byte)	0x00	
Register 10 (LSB, 2nd byte)	0x00	
Register 11 (MSB, 1st byte)	0x00	
Register 11 (LSB, 2nd byte)	0x00	
Register 12 (MSB, 1st byte)	0x00	
Register 12 (LSB, 2nd byte)	0x00	
Register 13 (MSB, 1st byte)	0x00	
Register 13 (LSB, 2nd byte)	0x00	
Register 14 (MSB, 1st byte)	0x00	
Register 14 (LSB, 2nd byte)	0x00	
Register 15 (MSB, 1st byte)	0x00	
Register 15 (LSB, 2nd byte)	0x00	
Register 16 (MSB, 1st byte)	0x00	
Register 16 (LSB, 2nd byte)	0x00	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x16	
Register 18 (MSB, 1st byte)	0x00	Sub-command 2: read a block
Register 18 (LSB, 2nd byte)	0x02	
Register 19 (MSB, 1st byte)	0x01	Section value, here all sections
Register 19 (LSB, 2nd byte)	0xFF	
Register 20 (MSB, 1st byte)	0x00	Read method: between 2 dates
Register 20 (LSB, 2nd byte)	0x01	
Register 21 (MSB, 1st byte)	0x28	Start date: 1st September 2021 at 1 hour 1 minute and 1 second
Register 21 (LSB, 2nd byte)	0xC1	
Register 22 (MSB, 1st byte)	0x71	
Register 22 (LSB, 2nd byte)	0x2D	
Register 23 (MSB, 1st byte)	0x2B	End date: 16 December 2022 at 1 hour 1 minute and 1 second
Register 23 (LSB, 2nd byte)	0x2E	
Register 24 (MSB, 1st byte)	0x71	
Register 24 (LSB, 2nd byte)	0xBD	
Register 25 (MSB, 1st byte)	0x00	Block index (0 for the first block)
Register 25 (LSB, 2nd byte)	0x00	

Step 3.2 - Reading the first block

Response to be read using the function H'03 (0x03) from the register 0x3200.

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 111 associated for this request in the register 0x3200
Register 1 (LSB, 2nd byte)	0x6F	
Register 2 (MSB, 1st byte)	0x00	Function code status, zero value for success
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length in number of bytes: 230 = 115 registers
Register 3 (LSB, 2nd byte)	0xE6	
Register 4 (MSB, 1st byte)	0x01	Section value
Register 4 (LSB, 2nd byte)	0xFF	
Register 5 (MSB, 1st byte)	0x00	Read method: 1
Register 5 (LSB, 2nd byte)	0x01	
Register 6 (MSB, 1st byte)	0x00	Number of log blocks
Register 6 (LSB, 2nd byte)	0x02	
Register 7 (MSB, 1st byte)	0x00	Index of current block
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	Event number in the block: 10
Register 8 (LSB, 2nd byte)	0x0A	
Register 9 (MSB, 1st byte)	0x00	Section order number (see table at the start of the chapter): 8 = Optional alarm
Register 9 (LSB, 2nd byte)	0x08	
Register 10 (MSB, 1st byte)	0x00	Event identifier see the table in Appendix 4. Here 37 = "Under instantaneous voltage U31".
Register 10 (LSB, 2nd byte)	0x25	
Register 11 (MSB, 1st byte)	0x00	Event status: 1 = Appearance (for the end state: 0, see Event Management chapter in the User manual for sentinel Energy hw+ electronic trip units)
Register 11 (LSB, 2nd byte)	0x01	
Register 12 (MSB, 1st byte)	0x00	Machine timestamping
Register 12 (LSB, 2nd byte)	0x7E	
Register 13 (MSB, 1st byte)	0xB4	
Register 13 (LSB, 2nd byte)	0x22	
Register 14 (MSB, 1st byte)	0x2B	User timestamping
Register 14 (LSB, 2nd byte)	0x2D	
Register 15 (MSB, 1st byte)	0x9B	
Register 15 (LSB, 2nd byte)	0x09	
Register 16 (MSB, 1st byte)	0x00	Event user data
Register 16 (LSB, 2nd byte)	0x00	
Register 17 (MSB, 1st byte)	0x00	Additional information
Register 17 (LSB, 2nd byte)	0x00	
Register 18 (MSB, 1st byte)	0x00	
Register 18 (LSB, 2nd byte)	0x00	
Register 19 (MSB, 1st byte)	0x00	
Register 19 (LSB, 2nd byte)	0x00	
Register 20 (MSB, 1st byte)	0x00	Section order number (see table at the start of the chapter): 0 = Alarm
Register 20 (LSB, 2nd byte)	0x00	
Register 21 (MSB, 1st byte)	0x00	Event identifier: 24 = Overload alarm. The identifier of the events in sections 0 to 7 corresponds to the event number listed in chapter 10 of the user manual for the sentinel Energy hw+ electronic trip units.
Register 21 (LSB, 2nd byte)	0x18	
etc.	etc.	etc.
etc.	etc.	
Register 117 (MSB, 1st byte)	0x00	Additional information
Register 117 (LSB, 2nd byte)	0x00	
Register 118 (MSB, 1st byte)	0x00	
Register 118 (LSB, 2nd byte)	0x00	

Step 3.3 - Read request for the second block

Request using the Modbus function H'16 (0x10) to write in the registers 0x3000 to 0x3011.

Message field	**Value**	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 111 associated with this request
Register 1 (LSB, 2nd byte)	0x6F	
Register 2 (MSB, 1st byte)	0x00	Security level 0 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x00	Password hash for level 0: 0 in addresses 0x3002 to 0x300F
Register 3 (MSB, 1st byte)	0x00	
Register 3 (LSB, 2nd byte)	0x00	
Register 4 (MSB, 1st byte)	0x00	
Register 4 (LSB, 2nd byte)	0x00	
Register 5 (MSB, 1st byte)	0x00	
Register 5 (LSB, 2nd byte)	0x00	
Register 6 (MSB, 1st byte)	0x00	
Register 6 (LSB, 2nd byte)	0x00	
Register 7 (MSB, 1st byte)	0x00	
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	
Register 8 (LSB, 2nd byte)	0x00	
Register 9 (MSB, 1st byte)	0x00	
Register 9 (LSB, 2nd byte)	0x00	
Register 10 (MSB, 1st byte)	0x00	
Register 10 (LSB, 2nd byte)	0x00	
Register 11 (MSB, 1st byte)	0x00	
Register 11 (LSB, 2nd byte)	0x00	
Register 12 (MSB, 1st byte)	0x00	
Register 12 (LSB, 2nd byte)	0x00	
Register 13 (MSB, 1st byte)	0x00	
Register 13 (LSB, 2nd byte)	0x00	
Register 14 (MSB, 1st byte)	0x00	
Register 14 (LSB, 2nd byte)	0x00	
Register 15 (MSB, 1st byte)	0x00	
Register 15 (LSB, 2nd byte)	0x00	
Register 16 (MSB, 1st byte)	0x00	
Register 16 (LSB, 2nd byte)	0x00	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x16	
Register 18 (MSB, 1st byte)	0x00	Sub-command 2: read a block
Register 18 (LSB, 2nd byte)	0x02	
Register 19 (MSB, 1st byte)	0x01	Section value, here all sections
Register 19 (LSB, 2nd byte)	0xFF	
Register 20 (MSB, 1st byte)	0x00	Read method: between 2 dates
Register 20 (LSB, 2nd byte)	0x01	
Register 21 (MSB, 1st byte)	0x28	Start date: 1st September 2021 at 1 hour 1 minute and 1 second
Register 21 (LSB, 2nd byte)	0xC1	
Register 22 (MSB, 1st byte)	0x71	
Register 22 (LSB, 2nd byte)	0x2D	
Register 23 (MSB, 1st byte)	0x2B	End date
Register 23 (LSB, 2nd byte)	0x2E	
Register 24 (MSB, 1st byte)	0x71	
Register 24 (LSB, 2nd byte)	0xBD	
Register 25 (MSB, 1st byte)	0x00	Block index (1 for the second block)
Register 25 (LSB, 2nd byte)	0x01	

Step 3.4 - Reading the second block

Response to be read using the function H'03 (0x03) from the register 0x3200.

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 111 associated for this request in the register 0x3200
Register 1 (LSB, 2nd byte)	0x6F	
Register 2 (MSB, 1st byte)	0x00	Function code status, zero value for success
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length in number of bytes: 230 = 115 registers
Register 3 (LSB, 2nd byte)	0xE6	
Register 4 (MSB, 1st byte)	0x01	Section value
Register 4 (LSB, 2nd byte)	0xFF	
Register 5 (MSB, 1st byte)	0x00	Read method: 1
Register 5 (LSB, 2nd byte)	0x01	
Register 6 (MSB, 1st byte)	0x00	Number of log blocks
Register 6 (LSB, 2nd byte)	0x02	
Register 7 (MSB, 1st byte)	0x00	Index of current block
Register 7 (LSB, 2nd byte)	0x01	
Register 8 (MSB, 1st byte)	0x00	Event number in the block: 10
Register 8 (LSB, 2nd byte)	0x0A	
Register 9 (MSB, 1st byte)	0x00	Section order number (see table at the start of the chapter): 2 = Error
Register 9 (LSB, 2nd byte)	0x02	
Register 10 (MSB, 1st byte)	0x00	Event identifier: 6 = E006: Critical Error 4. The identifier of the events in sections 0 to 7 corresponds to the event number listed in chapter 10 of the user manual for the sentinel Energy hw+ electronic trip units.
Register 10 (LSB, 2nd byte)	0x06	
Register 11 (MSB, 1st byte)	0x00	Event status: 1 = Appearance (for the end state: 0, see Chapter 10 of the User manual for sentinel Energy hw+ electronic trip units)
Register 11 (LSB, 2nd byte)	0x01	
Register 12 (MSB, 1st byte)	0x00	Machine timestamping
Register 12 (LSB, 2nd byte)	0x7E	
Register 13 (MSB, 1st byte)	0xB4	
Register 13 (LSB, 2nd byte)	0x22	
Register 14 (MSB, 1st byte)	0x2B	User timestamping
Register 14 (LSB, 2nd byte)	0x2D	
Register 15 (MSB, 1st byte)	0x9B	
Register 15 (LSB, 2nd byte)	0x09	
Register 16 (MSB, 1st byte)	0x00	Event user data
Register 16 (LSB, 2nd byte)	0x00	
Register 17 (MSB, 1st byte)	0x00	Additional information
Register 17 (LSB, 2nd byte)	0x00	
Register 18 (MSB, 1st byte)	0x00	
Register 18 (LSB, 2nd byte)	0x00	
Register 19 (MSB, 1st byte)	0x00	
Register 19 (LSB, 2nd byte)	0x00	
Register 20 (MSB, 1st byte)	0x00	Section order number (see table at the start of the chapter): 2 = Error
Register 20 (LSB, 2nd byte)	0x02	
Register 21 (MSB, 1st byte)	0x00	Event identifier: 21 = E021: Trip unit temperature
Register 21 (LSB, 2nd byte)	0x15	
etc.	etc.	etc.
etc.	etc.	
Register 117 (MSB, 1st byte)	0x00	Additional information
Register 117 (LSB, 2nd byte)	0x00	
Register 118 (MSB, 1st byte)	0x00	
Register 118 (LSB, 2nd byte)	0x00	

Step 4 - Deletion of the view

Request using the Modbus function H'16 (0x10) to write in the registers 0x3000 to 0x3018.

Message field	**Value**	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 111 associated with this request
Register 1 (LSB, 2nd byte)	0x6F	
Register 2 (MSB, 1st byte)	0x00	Security level 0 write at address 0x3001
Register 2 (LSB, 2nd byte)	0x00	Password hash for level 0: 0 in addresses 0x3002 to 0x300F
Register 3 (MSB, 1st byte)	0x00	
Register 3 (LSB, 2nd byte)	0x00	
Register 4 (MSB, 1st byte)	0x00	
Register 4 (LSB, 2nd byte)	0x00	
Register 5 (MSB, 1st byte)	0x00	
Register 5 (LSB, 2nd byte)	0x00	
Register 6 (MSB, 1st byte)	0x00	
Register 6 (LSB, 2nd byte)	0x00	
Register 7 (MSB, 1st byte)	0x00	
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	
Register 8 (LSB, 2nd byte)	0x00	
Register 9 (MSB, 1st byte)	0x00	
Register 9 (LSB, 2nd byte)	0x00	
Register 10 (MSB, 1st byte)	0x00	
Register 10 (LSB, 2nd byte)	0x00	
Register 11 (MSB, 1st byte)	0x00	
Register 11 (LSB, 2nd byte)	0x00	
Register 12 (MSB, 1st byte)	0x00	
Register 12 (LSB, 2nd byte)	0x00	
Register 13 (MSB, 1st byte)	0x00	
Register 13 (LSB, 2nd byte)	0x00	
Register 14 (MSB, 1st byte)	0x00	
Register 14 (LSB, 2nd byte)	0x00	
Register 15 (MSB, 1st byte)	0x00	
Register 15 (LSB, 2nd byte)	0x00	
Register 16 (MSB, 1st byte)	0x00	
Register 16 (LSB, 2nd byte)	0x00	
Register 17 (MSB, 1st byte)	0x00	Length of the parameter in bytes
Register 17 (LSB, 2nd byte)	0x16	
Register 18 (MSB, 1st byte)	0x00	Sub-command 3: delete the view
Register 18 (LSB, 2nd byte)	0x03	
Register 19 (MSB, 1st byte)	0x01	Section value, here all sections
Register 19 (LSB, 2nd byte)	0xFF	
Register 20 (MSB, 1st byte)	0x00	Read method: between 2 dates
Register 20 (LSB, 2nd byte)	0x01	
Register 21 (MSB, 1st byte)	0x28	Start date: 1st September 2021 at 1 hour 1 minute and 1 second
Register 21 (LSB, 2nd byte)	0xC1	
Register 22 (MSB, 1st byte)	0x71	
Register 22 (LSB, 2nd byte)	0x2D	
Register 23 (MSB, 1st byte)	0x2B	End date
Register 23 (LSB, 2nd byte)	0x2E	
Register 24 (MSB, 1st byte)	0x71	
Register 24 (LSB, 2nd byte)	0xBD	
Register 25 (MSB, 1st byte)	0x00	Block index (0 for the view)
Register 25 (LSB, 2nd byte)	0x00	

Step 5 - Read the status of the view deletion command

Response to be read using the function H’03 (0x03) from the register 0x3200.

Registers to be read from 0x3200	Value	Comment
Register 1 (MSB, 1st byte)	0x00	Identifier of command 111 associated for this request in the register 0x3200
Register 1 (LSB, 2nd byte)	0x6F	
Register 2 (MSB, 1st byte)	0x00	Function code status, zero value for success
Register 2 (LSB, 2nd byte)	0x00	
Register 3 (MSB, 1st byte)	0x00	Length in number of bytes: 5 registers
Register 3 (LSB, 2nd byte)	0x0A	
Register 4 (MSB, 1st byte)	0x01	Section value
Register 4 (LSB, 2nd byte)	0xFF	
Register 5 (MSB, 1st byte)	0x00	Reading method : 1
Register 5 (LSB, 2nd byte)	0x01	
Register 6 (MSB, 1st byte)	0x00	Number of log blocks
Register 6 (LSB, 2nd byte)	0x00	
Register 7 (MSB, 1st byte)	0x00	Index of current block
Register 7 (LSB, 2nd byte)	0x00	
Register 8 (MSB, 1st byte)	0x00	Event number in the block
Register 8 (LSB, 2nd byte)	0x00	

CRC
Cyclic Redundancy Check. The CRC is used to verify the integrity of the Modbus message sent.

DHCP
Dynamic Host Configuration Protocol. Dynamic Host Configuration Protocol used to manage IP addresses.

LSB
Least Significant Byte.

MSB
Most Significant Byte.

RTU
Modbus RTU (Remote Terminal Unit), is an Open Source serial protocol based on the master / slave design initially created by Modicon (currently Schneider Electric).

SALT
Name for a security key used to encrypt connection information.

SCADA
Supervisory Control and Data Acquisition. Industrial supervision system that processes a large number of measurements and remote control operations for installations in real time.

SHA
Secure Hash Algorithm.

SNTP
Simple Network Time Protocol. Used by a server managing the date and time of the communication network.

SELV
Safety Extra Low Voltage.

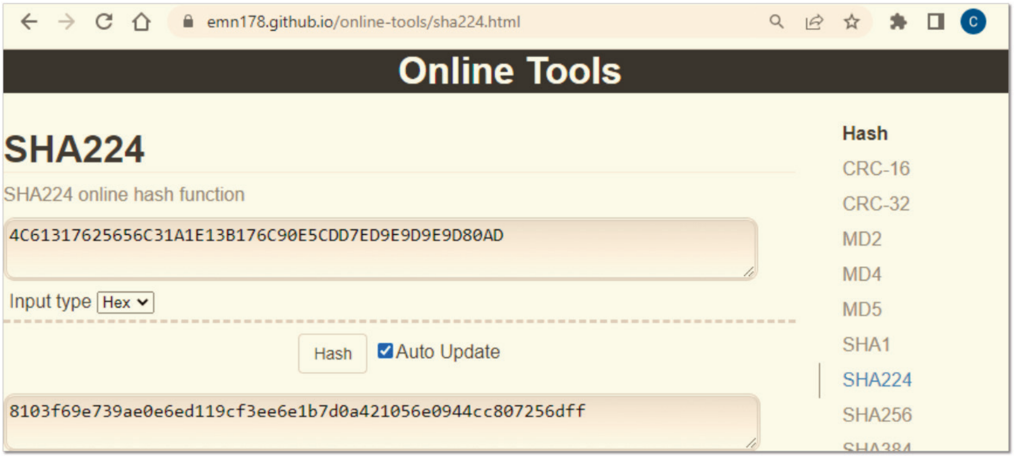
TCP
Transmission Control Protocol. TCP/IP is a set of standardised rules allowing computers to communicate on a network such as the Internet.

TLS
Transport Layer Security.

You can consult the sentinel Energy Modbus table of registers at this address: https://hgr.io/r/HW1E416FE

The SHA-224 function is part of the SHA-2 (Secure Hash Algorithm) family of hash functions designed originally by the National Security Agency of the United States (NSA).

The tool available on the following websites can verify the SHA-224 hash calculation based on a character string:
https://emn178.github.io/online-tools/sha224.html.



Several source code examples for the SHA-224 hash algorithm are available on the internet.

The SHA-224() function is available in the Python 3 language.

For the C# language, the Bouncy Castle open source library allows the SHA-224 hash algorithm to be used.

Find more information on the site: https://www.bouncycastle.org/index.html.

The BouncyCastle library is available at the following address:
https://www.nuget.org/packages/BouncyCastle.Cryptography.

ID	Description
1	Over instantaneous current I1
2	Over instantaneous current I2
3	Over instantaneous current I3
4	Over instantaneous current IN
5	Over instantaneous current I MAX
6	Under instantaneous current I1
7	Under instantaneous current I2
8	Under instantaneous current I3
9	Under instantaneous current IN
10	Earth fault alarm
11	Over unbalanced current value I1
12	Over unbalanced current value I2
13	Over unbalanced current value I3
14	Over unbalanced current max value
15	Over average current value Iavg
16	Under average current value Iavg
17	Over instantaneous voltage V1N
18	Over instantaneous voltage V2N
19	Over instantaneous voltage V3N
20	Over instantaneous voltage Vmax
21	Under instantaneous voltage V1N
22	Under instantaneous voltage V2N
23	Under instantaneous voltage V3N
24	Under instantaneous voltage Vmin
25	Over unbalanced voltage value V1N
26	Over unbalanced voltage value V2N
27	Over unbalanced voltage value V3N
28	Over unbalanced voltage phase-to-neutral max value
29	Over average voltage value Vavg
30	Under average voltage value Vavg
31	Over instantaneous voltage U12
32	Over instantaneous voltage U23
33	Over instantaneous voltage U31
34	Over instantaneous voltage Umax
35	Under instantaneous voltage U12
36	Under instantaneous voltage U23
37	Under instantaneous voltage U31
38	Under instantaneous voltage Umin
39	Over unbalanced voltage value U12
40	Over unbalanced voltage value U23
41	Over unbalanced voltage value U31
42	Over unbalanced phase voltage max value
43	Over import active power P1
44	Over import active power P2
45	Over import active power P3
46	Over import active power Total
47	Under import active power P1

48	Under import active power P2
49	Under import active power P3
50	Under import active power Total
51	Over export active power P1
52	Over export active power P2
53	Over export active power P3
54	Over export active power Total
55	Under export active power P1
56	Under export active power P2
57	Under export active power P3
58	Under export active power Total
59	Over import reactive power Q1
60	Over import reactive power Q2
61	Over import reactive power Q3
62	Over import reactive power Total
63	Under import reactive power Q1
64	Under import reactive power Q2
65	Under import reactive power Q3
66	Under import reactive power Total
67	Over export reactive power Q1
68	Over export reactive power Q2
69	Over export reactive power Q3
70	Over export reactive power Total
71	Under export reactive power Q1
72	Under export reactive power Q2
73	Under export reactive power Q3
74	Under export reactive power Total
75	Over apparent power S1
76	Over apparent power S2
77	Over apparent power S3
78	Over apparent power Total
79	Under apparent power S1
80	Under apparent power S2
81	Under apparent power S3
82	Under apparent power Total
83	Lagging power factor PF1 (IEEE) (under)
84	Lagging power factor PF2 (IEEE) (under)
85	Lagging power factor PF3 (IEEE) (under)
86	Lagging power factor Total (IEEE) (under)
87	Leading cos Phi 1 (IEEE) (under)
88	Leading cos Phi 2 (IEEE) (under)
89	Leading cos Phi 3 (IEEE) (under)
90	Leading cos Phi Total (IEEE) (under)
91	Lagging cos Phi 1 (IEEE) (under)
92	Lagging cos Phi 2 (IEEE) (under)
93	Lagging cos Phi 3 (IEEE) (under)
94	Lagging cos Phi Total (IEEE) (under)
95	Over THD current I1

96	Over THD current I2
97	Over THD current I3
98	Over THD voltage V1N
99	Over THD voltage V2N
100	Over THD voltage V3N
101	Over THD voltage U12
102	Over THD voltage U23
103	Over THD voltage U31
104	Over frequency
105	Under frequency
106	Over current demand I1
107	Over current demand I2
108	Over current demand I3
109	Over current demand IN
110	Over current demand Iavg
111	Under current demand I1
112	Under current demand I2
113	Under current demand I3
114	Under current demand IN
115	Under current demand Iavg
116	Over active power demand
117	Under active power demand
118	Over reactive power demand
119	Under reactive power demand
120	Over apparent power demand
121	Under apparent power demand
122	Operating quadrant 1
123	Operating quadrant 2
124	Operating quadrant 3
125	Operating quadrant 4
126	Phase sequence 1,2,3
127	Phase sequence 1,3,2
128	Lead
129	Lag
130	Leading power factor PF1 (IEEE) (under)
131	Leading power factor PF2 (IEEE) (under)
132	Leading power factor PF3 (IEEE) (under)
133	Leading power factor Total (IEEE) (under)
134	Over THD current IN

LWIP
Copyright: Copyright (c) 2001, 2002 Swedish Institute of Computer Science

License Text:

This file is part of the lwIP TCP/IP stack.
Author: Adam Dunkels <adam@sics.se>

MbedTLS
Copyright: Copyright The Mbed TLS Contributors

License Text:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

«License» shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

«Licensor» shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

«Legal Entity» shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
«control» means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

«You» (or «Your») shall mean an individual or Legal Entity
exercising permissions granted by this License.

«Source» form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

«Object» form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

«Work» shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

«Derivative Works» shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

«Contribution» shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, «submitted» means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as «Not a Contribution.»

«Contributor» shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a «NOTICE» text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an «AS IS» BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets «[]» replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same «printed page» as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the «License»); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an «AS IS» BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

FREE RTOS KERNEL
Copyright: Copyright (C) 2019 Amazon.com, Inc. or its affiliates

License Text:

FreeRTOS Kernel V10.2.1
Copyright (C) 2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the «Software»), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED «AS IS», WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

http://www.FreeRTOS.org
http://aws.amazon.com/freertos

CMSIS

License Text:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

«License» shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

«Licensor» shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

«Legal Entity» shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, «control» means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

«You» (or «Your») shall mean an individual or Legal Entity exercising permissions granted by this License.

«Source» form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

«Object» form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

«Work» shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

«Derivative Works» shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

«Contribution» shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, «submitted» means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as «Not a Contribution.»

«Contributor» shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a «NOTICE» text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an «AS IS» BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets «{}» replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same «printed page» as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the «License»);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an «AS IS» BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

CMSIS Device
Copyright: Copyright (c) 2019 STMicroelectronics

License Text:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

«License» shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

«Licensor» shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

«Legal Entity» shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, «control» means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

«You» (or «Your») shall mean an individual or Legal Entity exercising permissions granted by this License.

«Source» form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

«Object» form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

«Work» shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

«Derivative Works» shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

«Contribution» shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, «submitted» means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as «Not a Contribution.»

«Contributor» shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a «NOTICE» text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an «AS IS» BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets «{}» replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same «printed page» as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the «License»); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an «AS IS» BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS «AS IS» AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

STM32H7 HAL
Copyright: Copyright (c) 2017 STMicroelectronics.

License Text:

Copyright 2017 STMicroelectronics. All rights reserved

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS «AS IS» AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

STM32L4 HAL
Copyright: Copyright (c) 2017 STMicroelectronics.

License Text:

Copyright 2017 STMicroelectronics. All rights reserved

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS «AS IS» AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For third party technology that you receive from Hager Group or its affiliates in binary form which is licensed under an open source license, you can receive a copy of this specific source code on physical media by submitting a written request to :
Hager Electro SAS
Electronics Product Department
132 Boulevard de l'Europe
67215 Obernai - France
Or, you may send an email to Hager Group using the following email address: sourcecoderequest.grouplevel@hagergroup.com.
In both cases your request should include :

- The name of the component for which you are requesting the source code
- The reference and version number of the Hager product containing the binary
- The date you received the Hager product
- Your name
- Your company name (if applicable)
- Your return mailing address and email and
- A telephone number in the event we need to reach you.
- You may add additional comments to highlight your request.

We may charge you a fee to cover the cost of physical media and processing.
Your request must be sent :
(i) within three (3) years of the date you received the Hager product that included the component or binary file(s) that are the subject of your request,
or
(ii) in the case of code licensed under the GPL v3, for as long as Hager sales this product or customer support for that product in the country of the requester.

Hager Electro SAS
132 Boulevard d’Europe
BP3
67210 OBERNAI CEDEX

www.hager.com

07/2023

6LE007964A