MCM-12
DATASHEET

PRODUCT INFORMATION

IMPORTANT NOTICE

This product is a development tool and primarily intended for use in lab environment. All electrical connections are designed according to best practices, but mechanical strength and durability is not as an industrial product.
The software functionality of the product, is still in pre-production state and all samples are intended for evaluation purposes only.

USE

The Embedded Modbus Analyser Tool is designed to simplify testing and integration of Modbus systems. The only configuration required is to set the communication settings. Everything else is handled automatically. The device allows all data between units, to be monitored and analysed. All devices and register access/values can be read directly from the display. Either in text format or as graphs (1 second resolution).

Communication log can either be streamed (raw or analysed data) over CDC port (Virtual Com Port), or saved to internal SD card for later analysis. Either all information is saved or communication with specific unit(s) can be filtered.

TYPE SUMMERY

Type (ASN)	Ordering number	Connactivity	Singals
MCM-12	HAT-M12	Modbus RTU	RS485

ACCESSORIES

Type (ASN)	Ordering number	Description
MCM-USB-AC	MCM-USB-AC	1m USB A to USB C
MCM-USB-CC	MCM-USB-CC	1m USB C to USB C
MCM-M12-1	MCM-M12-1	1 m M12 cable. One open end
MCM-M12-3	MCM-M12-3	3 m M12 cable. One open end

ORDERING

order@hemsek.com

NEW FEATURES

This is a development tool with active development. Proposals for new features are highly appreciated and all idea's feasibility will be investigated. Over the coming months, there are planned new features to be added. See later paragraphs for more information.

TECHNICAL DATA

Electrical interface	
Power supply Supply voltage Power consumption Alternative power supply	SELV 24 VAC/VDC +/- 20% 50/60 Hz < 2 VA 5 VDC @ 150 mA over USB-C
Functional data	
RS485 Signals USB Signals	Max - 7 .. 12 VDC Max 5.5 VDC
Protection class	
IEC 60 529	IP21
v	
Electrical connections - 5 pin M12 connector - USB-C	
Environmental conditions	
Permissible ambient temperature - Operation - Transport/storage Permissible ambient humidity	 10 .. 30 °C (non-condensing) -10 .. 40 °C (non-condensing) < 90% R.H. (without condensation)
Directives, standards	
conformity as per EMC guidelines - Immunity - Emissions	 No testing completed at this time No testing completed at this time
Weight	
Product including package	< 150 g
Size	
Enclosure (W x L x H)	109 x 93 x 22 mm
Total (W x L x H)	150 x 110 x 35 mm
Made in Sweden	

CONNACTIVITY

The device include
- one RS485 interface capable of operating with a baud rate up to 115200 with standard firmware. And higher with custom firmware.
- one CDC USB interface over USB-C.

CONFIGURATIONS

Baudrate	9600, 19200, 38400, 57600, 115200
Format	Modbus RTU
Line termination	NONE / ODD / EVEN
Hardware	RS485

ENGINEERING NOTES

The device does not have galvanic separation between RS485 interface, external power and/or USB-C connection. Caution must be taken mains power computers is connected over USB-C.
For standalone operation with power and RS485 connection over M12 connector, the special cautions are required.
For full galvanic separation it is recommended to operate the device from a power bank.

DEVICE EMULATION

In applications where firmware development takes place or where new features are tested. It is possible to let the device emulate other end devices. This allows values to be manually modified from the display or allow system to continue operate, while other devices for example are restarted or reprogrammed.
This require the real units to be used at startup, so registers are identified, and base values are stored.

The function is implemented using the average response time of each device and if an end device doesn't respond within 200% of the average, this unit will answer the request instead.
Enabling/disabling the emulation feature is system global, and single units, multiple units or all units in a system can be emulated at once.

All values can be modified directly on the display if emulation is activated, and device is emulated. Any access directly with physical device is automatically restore values to physical register value.

This feature is not enabled in firmware version 1. The function is implemented in will be included in next release in May 2023.

USB LOGGING

When USB cable is connected, it is possible to configure the device to send communication data over virtual com port.
At the PC or MAC end, only a terminal program is required.

Information sent can be selected between the following formats.

Mode	Description
trace	All received Modbus requests are encoded in hex format. Any new request starts with either < or > > = controller to end device communication < = end device to controller communication
decoded simple	All information from trace, but also decode the frame with information about Modbus command code, registers accessed and more.
decoded full	All information from decoded simple but full decoding of messages including data values.
trace encode	Reserved mode for communication with PC application.
raw	Reserved mode for later implementations

SD CARD LOGGING

An internal SD card of up to 32 GB can be used to store settings and communication logs. More information about this will be included with next firmware release.

This feature is not enabled in firmware version 1. The function is implemented in will be included in next release in May 2023.

COMMUNICATION FILTERS

When USB or SD card logging is used it is possible to configure the device to filter communication with specific units. By default all communication is exchanged. But by selecting the filter option in the menu only activated units are included in communication.

The filter enable/disable functionality is access under each device.

DISPLAY AND HMI

The unit includes a 480x272 4.3 inch full color display with capacitive touch.
The screen is used for all user inpu.

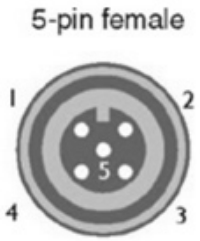
On the right side of the unit there are two 2 color LEDs used for debugging functionality.

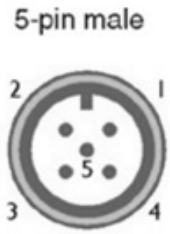
DISPLAY TOOLBAR

At the left side of the screen all function options are always visible. Just click on the options to change the status and/or functtionality.

	USB connection status. Icon will turn green when USB cable is connected and driver installed correctly at PC/MAC side.
	Filtering enabled/disabled. Icon will be green when filtering is enable.
	Emulation enabled/disabled. Icon will be green when emulation is enable.
	SD card logging. Icon will be green when SD card is present and function enabled.

TERMINALS





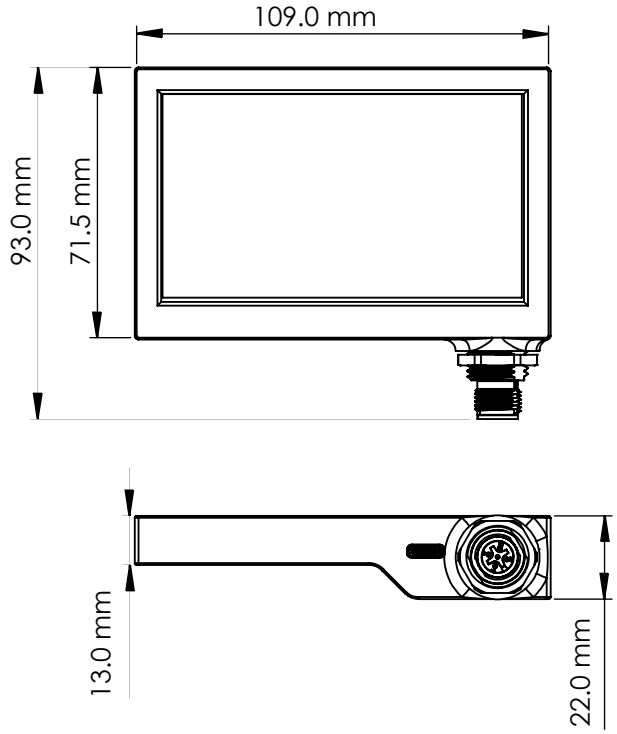
Signal description				
	Amphenol	**Phoenix**	**Other**	
1	Brown			Shielf/REF
2	White			Operating voltage AC/DC
3	Blue			GND (G0)
4	Black			Modbus Comm A (+)
5	Green	Gray	Green/Yellow	Modbus Comm B (-)

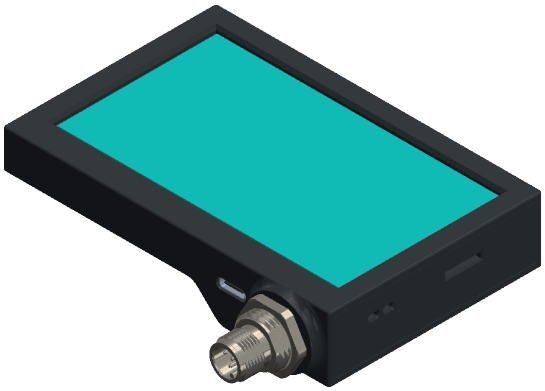
DIMENSIONS

NOTES

The device is included in a small ergonomic enclosure where the right side of the device is ticker due to M12 connector and internal capacitors. For a right-handed person, with not too big hands, it should be quite comfortable to hold the device. Most actions can be executed with only one hand.

Do not try to open the hatch on the right-hand side. It is not intended to be open, and the display cables are easily damaged if trying to take out the board.





MODBUS INTEGRATION

General information

The following section provides only a brief overview of the Modbus protocol. For the full specification, refer to "Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. J".

Master/slave protocol

The Modbus is a master/slave protocol. By definition, this means that a Modbus network contains one, and only one, master and at least one slave. The Modbus master starts the transactions on the network with a slave query. The slave either responds positively with the requested service (response) or transmits an "exception message". In the remainder of this document, these query/response sequences are also referred to as "Modbus telegrams".

Function codes

The type of transaction is defined by the function code transmitted in the Modbus telegrams. A function code defines the following:

- Structure of the telegram, query and response
- Direction of data transmission (master/slave or slave/master)
- Data format of data point (bit or 16-bit register)

Transmission mode

The Modbus protocol defines two alternative serial transmission modes. These modes have the following characteristics:
RTU (Remote Terminal Unit) mode Binary-coded data

- Start and end of telegrams marked by timed pauses (a "silent interval") between the characters transmitted.
- Check sum algorithm: CRC (Cyclic Redundancy Check)
- ASCII mode
- Data in hexadecimal notation
- Beginning and end of telegrams marked by start and end characters.
- Check sum algorithm: LRC (Longitudinal Redundancy Check)

Telegrams with points

Certain types of Modbus transactions permit the transmission of a multiple data variable number of Modbus data points (bit or 16-bit register) in a single telegram.

RS485 Network

RS485 is a balanced line, half-duplex transmission system that meets the requirements for a truly multi-point communications network, and the standard specifies up to 64 drivers and 64 receivers on a single (2-wire + common) bus.
Half-duplex data transmission means that data can be transmitted in both directions on a signal carrier, but not at the same time.

Tools

Modbus slave devices can be tested with several Modbus master simulation tools, like "Modbus Poll" or "ModScan", from a computer. Modbus Poll can be downloaded from www.modbustools.com.
A RS485/RS232 converter or a Modbus RTU/TCP gateway may be needed to connect to a computer.

Troubleshooting, Tipst

The slave address must be unique in the network, valid addresses are from 1-247.
Only reference addresses that are generated can be read/write, see chapter modbus register map for more information about the specific application.
Baud rate, Parity and stop bits must match the network and the Master.
The 2-wire bus is NOT interchangeable and must be connected correctly. Common wire must be used at all times !
In case of long distance and/or high baud rate, please consider end of line resistors like 120 Ohm on both sides (according to RS485 rules)

Register map and function codes

Modbus registers are organized into reference types identified by the leading number of the reference address: The "x" following the leading character represents a four-digit reference address.

Function code	Function	Code
Coil status	0x<addr>	Read/Write Discrete Outputs or Coils. A 0x reference address is used to drive output data to a digital 1-bit output channel.

Input status	1x<addr>	Read Discrete Inputs. The 1-bit status of a 1x reference address is controlled by the corresponding digital input channel.
Input registers	3x<addr>	Read Input Registers. A 3x reference register contains a 16-bit number received from an external source - e.g. an analog signal.
Holding registers	4x<addr>	Read/Write Output or Holding Registers. A 4x register is used to store 16-bits of numerical data (binary or decimal), or to send the data from the CPU to an output channel.

Function code

Function code	Function	Code	Note
01	0x<addr>	Read coil status	1)
02	1x<addr>	Read input status	1)
03	4x<addr>	Read holding registers	Implemented
04	3x<addr>	Read input registers	1)
05	0x<addr>	Force single coil	1)
06	4x<addr>	Preset single register	Implemented
15	0x<addr>	Force multiple coils	1)
16	4x<addr>	Force multiple register	Implemented

Register values

Modbus registers are normally constructed from 16 bit unsigned integers. In some cases more information are packed into each register with the following coding. For those register that each bit have a spcific function the each bit within the register is described with the b prefix.

Example

b0 represent bit 0
b1 represents bit 1
b15 reprsents bit 15

For those registers where multiple information is stored in the same registers a range of bits are used for each value.

Example

b0..b7 represents the lowest 8 bits in the register.
value = <register value> AND 255
b8..b15 represents the higest 8 bits in the register.
value = (<register value> DIV 256) AND 255

Reference addresses

This chapter describes the reference addresses used in the application. Reference addresses marked "Not used", "Not specified" or "Reserved" can be accessed freely with read or write access.

It is recommended not to read or write any addresses not mentioned in this manual. If So there will be an exception response and the communication fails with current firmware or any later update.

All address types starts with 1, and due to that some master devices starts with 0 it's in that case necessary to subtract all addresses in this manual with 1.

- 16 bit real values are presented in their actual value/unit. E.g. °C, %, Pa, l/s
- 16 bit states are presented either as a number or as a bit pattern, see the reference address description
- 1 bit status are presented as 0=Off and 1=On
- 1 bit alarms are presented as 0=Normal and 1=Alarm