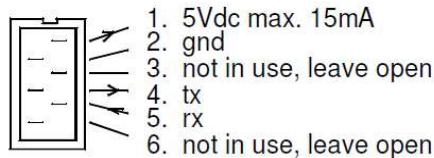


serial connector
micromatch 6 female



Connection

Devices can be connected for point to point communication directly using the onboard 'red' connector, image above. This is 5V logic serial line (not RS-485). When bus is needed a separate RS-485 adapter unit can be connected as well (EM-P189). Some devices have onboard RS-485 port for direct bus connection.

Serial line

The default Serial line configuration is 9600 bps, 8N1. Default slave address is 1. These can be changed with device's serial line parameters.

Modbus RTU

Serial line communication uses modbus protocol. Data is exchanged by modbus functions 03 read holding registers and 16 write multiple registers. Control and status data transfer can be also done with one function 23 read/write multiple registers.

Each device has common part of the messages, these are device info. with device's name and version, and messages for reading and writing parameters. These messages have the same data structure within all devices. Control and status messages are more device specific. Status messages are used to read device status. Some units may have status data divided in to two messages. First have data like drive direction, speed, motor current, supply voltage and input status. The other have data like start and run hour counters.

In this modbus implementation it is intended to use starting address to access the data. There is no random access to full address space. F.ex. all the parameters have to be written in one frame because device then saves all of them to a non volatile memory. For status data it can be chosen to read all or part of the data. Complete data structure can be seen in the following data charts.

Modbus map

Data	Start address	Access
Device info.	40001	read
Parameters	40101	read/write
Par. Range min.	40201	read
Par. Range max.	40301	read
Control	41001	write
Status 1	41101	read
Status 2	41201	read

Data	Address
Protocol version	40001
Device version	40002
Number of parameters	40003
Number of device name char.	40004
Name char. Ascii bytes	40005
.	.
.	.
Parameter 1	40101
Parameter 2	40102
Parameter 3	40103
.	.
.	.
Par. 1 range min.	40201
Par. 2 range min.	40202
Par. 3 range min.	40203
.	.
.	.
Par. 1 range max.	40301
Par. 2 range max.	40302
Par. 3 range max.	40303
.	.
.	.
Control data	41001
.	.
.	.
Status data 1	41101
.	.
.	.
Status data 2	41201
.	.
.	.

Read device info.

Data bytes	Value/range	Description
Slave address	1 – 247	
Read holding registers	3	
Address msb	0	Address 40001
Address lsb	0	
Quantity msb	0	
Quantity lsb	10	Quantity is not used, response have 20 bytes of information data.
CRC, lsb	0-255	
CRC, msb	0-255	

Read device info. response

Data bytes	Value/range	Description
Slave address	1 – 247	
Read holding registers	3	
Byte count	20	
Protocol version	2	
Protocol minor version	0-255	Default 0. Minor version can be used to distinct different control and status message configurations.
Not in use	0	
Device version	0-255	Version of the connected device sw. 10 means v1.0, 25 is v2.5 and so on.
Not in use	0	
Parameters	1-50	Number of configuration parameters.
Not in use	0	
Number of name characters	1-11	Number of connected unit's name characters.
Character 1	ASCII	Name characters ASCII. Unused characters have value 0.
Character 2	ASCII	
Character 3	ASCII	
Character 4	ASCII	
Character 5	ASCII	
Character 6	ASCII	
Character 7	ASCII	
Character 8	ASCII	
Character 9	ASCII	
Character 10	ASCII	
Character 11	ASCII	
Not in use	0	This keeps data size even if needed by modbus implementation.
CRC, lsb	0-255	
CRC, msb	0-255	

Parameter messages

Read all parameters

Data bytes	Value/range	Description
Slave address	1 – 247	
Read holding registers	3	
Address msb	0	Address 40101
Address lsb	0x64	
Quantity msb	0	Quantity is not used, response have data for all parameters. If needed for your modbus implementation you can use value here it's just ignored. Number of parameters in a device can be seen in device info.
Quantity lsb	1-50	
CRC lsb	0-255	
CRC msb	0-255	

Read all parameters response

Data bytes	Value/range	Description
Slave address	1 – 247	
Read holding registers	3	
Byte count	2-100	There can be 8bit and 16bit in a device, but all parameters are transferred as 16bit. With 8bit parameters msb is ignored.
Par. 1 msb	0-255	Parameter data, msb first
Par. 1 lsb	0-255	
Par. 2 msb	0-255	
Par. 2 lsb	0-255	
Par. 3 msb	0-255	
Par. 3 lsb	0-255	
...	...	
Par. N msb	0-255	Up to 50 parameters. Number of parameters in a device can be read with device info.
Par. N lsb	0-255	
CRC lsb	0-255	
CRC msb	0-255	

Write all parameters

Data bytes	Value/range	Description
Slave address	1 – 247	
Write multiple registers	16	
Address msb	0	Address 40101
Address lsb	0x64	
Quantity msb	0	<i>Quantity is not used, all parameters have to be sent at once. If needed for your modbus implementation you can use value there it's just ignored. Number of parameters in a device can be seen in device info.</i>
Quantity lsb	1-50	
Byte count	2-100	<i>2 x number of parameters.</i>
Par. 1 msb	0-255	<i>Parameter data, msb first. All parameters are send as a 16bit values. With 8bit parameters msb is ignored.</i>
Par. 1 lsb	0-255	
Par. 2 msb		
Par. 2 lsb		
Par. 3 msb		
Par. 3 lsb		
...	...	
Par. N msb	0-255	
Par. N lsb	0-255	
CRC lsb	0-255	
CRC msb	0-255	

This writes all parameters with one message.

Parameters are saved to memory which can take about 50ms – 150ms.

Response message is sent when saving is done.

Please note parameter memory can last only 100 000 saving times, when parameters like speed and current limit needs to be adjusted on the fly use control command instead.

Write all parameters response

Data bytes	Value/range	Description
Slave address	1 – 247	
Write multiple registers	16	
Address msb	0	Address 40101
Address lsb	0x64	
Quantity msb	0	<i>Quantity is not used, response have data for all parameters. If needed for your modbus implementation you can use value there it's just ignored. Number of parameters in a device can be seen in device info.</i>
Quantity lsb	1-50	
CRC lsb	0-255	
CRC msb	0-255	

Parameter range messages

Parameters have minimum and maximum value range.

There is no check for parameter values in a device. Read parameter max/min range and send parameter values only within these limits.

Read parameters range data minimum

Data bytes	Value/range	Description
Slave address	1 – 247	
Read holding registers	3	
Address msb	0	<i>Address 40201</i>
Address lsb	0xC8	
Quantity msb	0	<i>Quantity is not used, response have data for all parameters.</i>
Quantity lsb	1-50	
CRC lsb	0-255	
CRC msb	0-255	

Read all parameters range data minimum response

Data bytes	Value/range	Description
Slave address	1 – 247	
Read holding registers	3	
Byte count	2-100	<i>Byte count value is 2 x number of parameters.</i>
Par. 1 msb	0-255	<i>Parameter range data, msb first</i>
Par. 1 lsb	0-255	
Par. 2 msb	0-255	
Par. 2 lsb	0-255	
Par. 3 msb	0-255	
Par. 3 lsb	0-255	
...	...	
Par. N msb	0-255	<i>Up to 50 parameters.</i>
Par. N lsb	0-255	
CRC lsb	0-255	
CRC msb	0-255	

Read parameters range data maximum

Data bytes	Value/range	Description
Slave address	1 – 247	
Read holding registers	3	
Address msb	0x01	Address 40301
Address lsb	0x2C	
Quantity msb	0	Quantity is not used, response have data for all parameters.
Quantity lsb	1-50	If needed for your modbus implementation you can use value here it's just ignored.
CRC lsb	0-255	
CRC msb	0-255	

Read all parameters range data maximum response

Data bytes	Value/range	Description
Slave address	1 – 247	
Read holding registers	3	
Byte count	2-100	Byte count value is 2 x number of parameters.
Par. 1 msb	0-255	Parameter range data, msb first
Par. 1 lsb	0-255	
Par. 2 msb	0-255	
Par. 2 lsb	0-255	
Par. 3 msb	0-255	
Par. 3 lsb	0-255	
...	...	
Par. N msb	0-255	Up to 50 parameters.
Par. N lsb	0-255	
CRC lsb	0-255	
CRC msb	0-255	