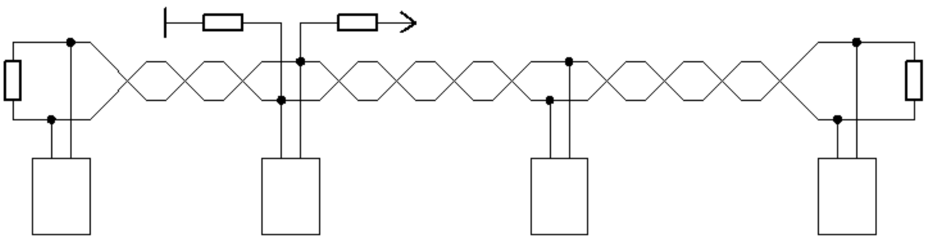
# Technical Documentation

# EMM5 – MODBUS

Beluk GmbH
Taubenstrasse 1
86956 Schongau
Germany

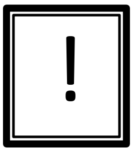
Tel.: +49/(0)8861/2332-0
Fax: +49/(0)8861/2332-22
E-Mail: blr@beluk.de
Web: http://www.beluk.de

## Document History

| Date | Name | Revision | Change |
|---|---|---|---|
| 21.11.05 | ATh | 01 | Initial document release |
| | | | Features of new software revision (V1.8.x) |
| 14.12.06 | ATh | 02 | Features of new software revision (V1.9.x) |
| 08.08.07 | CE | 03 | New connector on MODBUS hardware |
| 20.09.07 | Le | 04 | New label for 3 pin connector |
| 06.09.10 | Le | 05 | New Modbus addresses for revision V1.12.x |
| 27.08.18 | SMi | 06 | Layout changes |

# Table of Contents

# Important Information!



**If the upper sign appears besides a text passage in the manual, the reader is strongly advised to read the corresponding information as it may be very important for the usage of the device.**

**It can highlight safety advices or information about the correct handling of the device. If the information is disregarded, the device may be inoperable or even damaged!**

**Additional documentation of the MODBUS protocol can be found at www.modbus.org. The MODBUS standards are also available there.**

# 1. Overview

The MODBUS extension of the EMM5 offers the possibility to read values from the device via a remote connection, set up with a computer system.

This document describes the data transmission using the MODBUS-protocol. This protocol defines methods for data transmission and access control, but doesn't restrict the user to one single physical transmission system. In case of the EMM5, RS485 is used as physical layer. Since this is a bus-capable interface, it is possible to connect more than one EMM5 to a single pair of wire and access the units by use of an ID number.

A lot of commercial devices and PLCs are able to use the MODBUS protocol either as bus master or as slave. Various SCADA solutions are also available from different vendors. So, the integration of the EMM5 in an existing bus-system or setting up a new bus system is only a minor issue.

# 2. MODBUS / RS485

The implementation basically consists of two parts:

- The RS485 transmission is used for serial data transport. It is able to interconnect more than one device in a bus-like configuration. The RS485 protocol offers its “services” to the higher-level MODBUS protocol.

- The MODBUS protocol uses the serial data transport layer (RS485 in this case) to communicate with several bus devices. It defines commands, address structures and data structures to access the slave device.

## 2.1 RS485 (defined in EIA485/ISO8482)

RS485 offers basic serial data transport to the higher-level MODBUS protocol layers. Therefore, it is called the “physical layer” of the bus system. Higher layers use the lower physical layer as a basic “service” for data transport.
RS485 uses two data wires for serial transmission. Each of them is driven to 0V or 5V by the transmitting device. The two data wires always have different voltage levels. One state (one wire 5V, other wire GND) represents the logic “OFF” state. The two wires exchange their voltages for the logic “ON” state. This differential transmission mode makes the RS485 bus very resistant against electro-magnetic distortions and therefore allows long transmission distances of more than 1000 meters.
The data transmission rates of the EMM5 can be selected between 1200, 2400, 9600, 19200 or 38400 baud. The parity can be selected between even, odd and no parity. All bus devices need to be set up equally. The standard settings are: 9600 baud and even parity.

There are two different types of RS485:

- 2-wire RS485: This type uses only two data wires, which form one data channel. This means, that after sending a request, the bus master has to deactivate its transmitter to make the data line free for the answering device. (Half-duplex mode)
- 4-wire RS485: This type uses one data line (=two wires) for the master -> slave direction and another one (two more wires) for the slave -> master direction.

**The EMM5 supports 2 wire RS485 only!**

Although it is not mentioned explicitly, both types, 2-wire and 4-wire, need additionally a common ground GND. Thus, the 2-wire version requires a cable with 3 wires and the 4-wire version requires a cable with 5 wires! Moreover, a shielded cable is recommended. Here, the shield must not be used as GND connection. It should only be connected to protective ground to reduce electromagnetic influences.

The RS485 bus is capable to interconnect more than one device (typically up to 32). To accomplish this, the data signals of all bus members must be interconnected. This concerns the two data lines and the common ground GND. All devices are connected in parallel. It is recommended to avoid using taps, because they tend to be the source of transmission errors if they are too long. The connection between the bus and the device must be kept short.

A single bus cable, connecting several devices, is called a “bus segment”. Several segments can be interconnected using “repeaters”.

### 2.1.1 Connection

There are two different MODBUS interfaces:

**a) 9-pin D-SUB**

| | | | | |
|---|---|---|---|---|
| 5 | 4 | 3 | 2 | 1 |
| 9 | 8 | 7 | 6 | |

**PIN1:** +5V only for data line bias. Do not supply other circuits by this pin!

**PIN2:** GND for biasing and as common ground for all bus participants

**PIN5:** D (B) – Data signal B

**PIN9:** D (A) – Data signal A

**b) 3-pin connector**

This connection variant uses a 3-pin connector. The respective pinout is shown in the picture. In order to set up a connection, connect the data lines + and – as well as the common ground.


### 2.1.2 Line Termination

A very important issue is the termination of the bus line. In order to cancel echoes and interferences, resistors must be added at both ends of the bus cable. The value of the resistor must match the cables impedance. Usually, 120Ω is a good value to start with. At each end of the bus segment a 120 Ω must be added.

Some devices, especially bus converters have built-in resistors. Please check the respective manuals of all devices used on the bus. If these internal resistors cannot be disabled, this has an influence on the bus: This device must be placed at the end of the bus! As the bus has only two ends, you can use only two devices with integrated resistors per bus segment!

### 2.1.3 Line Biasing

Another important issue is line biasing. If no device is actually transmitting, the data wires are left floating. Because of the termination resistors, both lines have almost the same voltage. This could result in spurious data signals because of external influences. Line biasing is used to give the data wires a defined “OFF”-state in this case.

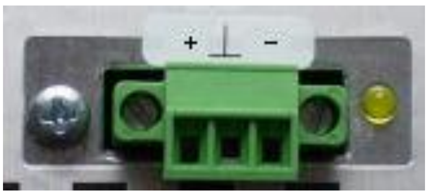
Two resistors of approx. 500 - 600Ω have to be connected from D(+) line to +5 V and from D(-) to GND. The two bias resistors are needed only once per bus, the position of the bias resistors is not important. They may be placed anywhere on the bus, even in the “middle”. Please check the manuals of all bus devices, if these resistors are internally provided!

When the device is equipped with connection variant a) (9-pin sub-d), the voltage levels 5V and GND are available on the bus connector, so these two resistors can be soldered inside the connector case.

Unfortunately, this is not possible for connection variant b) (3-pin).

Attention must be paid, using different vendors. Here, assuming A = + and B = - is not correct. This has to be checked from case to case.

### 2.1.4 Communication Indicator



The yellow LED on the backside of the device indicates an active transmission. It flashes only, if the device is actually communication with the bus master.

The communication indicator LED is available for both connector variants

## 2.2 The MODBUS Protocol

### 2.2.1 MODBUS - Description

The MODBUS protocol uses the RS485 as an underlying physical layer and implements the data transmission control mechanisms. It is therefore located on layer 2 ("link layer") of the OSI layer model for data exchange systems.

### 2.2.2 Serial data format and framing

The data is transmitted in fixed frames. The frames are separated by the bus being inactive for at least 3,5 character times. All data is organized in “protocol data units” (PDUs), which are transmitted over the serial bus system by the underlying physical protocol layer.

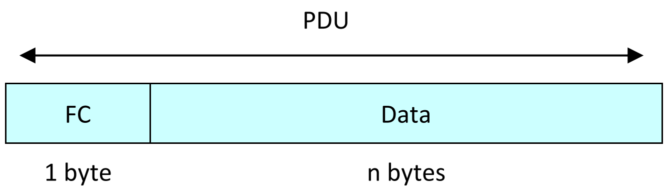



**Figure 1: “Protocol Data Unit” – PDU-Frame**

The PDU consists of two parts:

- The "function code" (FC) gives a command, which defines how the slave device has to answer the request.
- The data block includes the corresponding data to a FC. Its content depends on the FC, it can contain pure data but also register addresses for slave data access.

The PDU defines a single data unit, which has to reach a certain bus device in order to perform a task. The transmission type differs, depending on the used physical layer.

To be able to control the transmission, the PDU is extended with additional blocks of data for transmission control purposes. For RS485, this extension results in the "application data unit" (ADU)

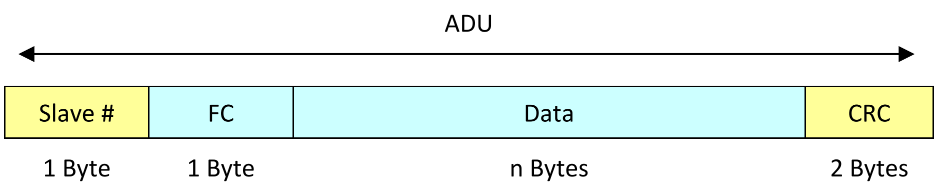



**Figure 2: Schematic of the "Application Data Unit" – ADU-Frame**

The application data unit, as it is used with the serial transmission over RS485, contains two additional blocks of data:

- The first field specifies the target for the data block, the "slave number" (= slave address)
- The transmission is additionally secured by a CRC16 error correction code.

### 2.2.3 Serial Transmission Modes

The protocol defines two different encodings for the frame's data content. The EMM5 always uses the RTU-mode! ASCII mode is not implemented and is only mentioned for the purpose of completeness.

**"Remote Terminal Unit" (RTU)**

In this transmission mode every 8-bit data word contains two 4-bit hexadecimal numbers. They are transmitted as one complete byte to reach the maximum transmission density. With every data word, the following information is transmitted:

- 1 start bit
- 8 data bits, "least significant bit" first
- 1 parity bit (if set)
- 1 stop bit for parity even or odd / 2 if parity is none to compensate missing parity bit

**"American Standard Code for Information Interchange" (ASCII)**

In ASCII-mode, the two 4-byte nibbles of an 8-bit data word are transmitted separately in ASCII-code representation. A data byte which contents $5B_{hex}$ will be divided into two parts and transmitted separately as one byte each. The result is that TWO data bytes are transmitted with contents $35_{hex}$ (=ASCII-Code "5") and $42_{hex}$ (=ASCII-Code "B"). This data mode is intended for compatibility reasons and makes debugging on the transmission line easier but it also decreases transmission speed significantly.

### 2.2.4 Function Codes

As mentioned earlier, the data packet contains "function codes", which specify a command from the bus master to the bus slave. The slave executes the command (if possible) and answers with the same function code to acknowledge the command. The valid range for function codes is specified from 1 to 255, but only a part of it is actually used. Please refer to the MODBUS specifications for detailed information. If the function code is not known by the slave, it replies with an exception (=an error code). The function code of an exception package includes the function code of the received command, which caused the error, and the most significant bit is set to high, in order to signal the error condition to the master. The contents of the data block specify the error in more detail.
The EMM5 supports the function codes 03hex (read holding register), 04 hex (read input register) and 06hex (write single register).

### 2.2.5 Exception Codes

If a slave is not able to execute a command, which was sent by the master, it answers with exception codes. A full list of codes can be found in the MODBUS specification. A list of exceptions is not included at this point, because the master software will be able to handle most exceptions automatically.

### 2.2.6 Master-Slave Protocol

For communication, a master-slave protocol is used. Only the bus master is permitted to initiate a transfer. The "master" starts the data exchange by sending a data frame with the corresponding function code (= command) to the slave, which will execute it subsequently.

- The unicast-mode is normally used to communicate on a Modbus system. One single slave is addressed by the slave number in the master's data package. The valid address range is between 1 and 247. The slave then executes the command and answers by sending a data package as acknowledgement back to the master.
- In multicast-mode all slaves on the bus are addressed in parallel. They all execute the same command, but none of them will respond. The master initiates a multicast communication by using "0" as slave number.

### 2.2.7 EMM5 - MODBUS Setup

If your device supports MODBUS, an additional entry is available in the device's "setup" – menu. After entering it, the following settings can be performed:

- ADDRESS: This is the devices slave address (slave ID). The valid range is 1-247.
- BAUD RATE: Select the baud rate here. The valid range is 1200 - 38400 baud.
- PARITY: Select the parity to be EVEN, ODD or NONE.

The settings for baud rate and parity must be the same for all bus devices, but the address must be unique for each device.

### 2.2.8 Adress Space

The data in the EMM5 is organized and accessed by means of addresses. Each address accesses a single data word. The data words are always 16-bit wide.

The EMM5 does not differentiate the addresses between the function codes. There is one big address space available and to access each address's data, any valid function code can be used. Nevertheless, the data will only make sense when it is interpreted in the correct way!

The data can be of the following types:

- REAL: this is a 32bit floating-point number, as defined in IEEE Standard 754.
- INT: this is a 16bit integer value.
- LONG: this is a 32bit integer value.

As the data is organized in 16 bit wide words, longer items must be read sequentially. For this data, the base address is given in the following tables. To read a REAL type with base address 12, two 16-bit words must be read (address 12 AND address 13). These two values need to be assembled to form the desired result of 32-bit. The most SCADA software packages or PLCs do this without user interaction.

There are different types of addresses:

- The MODBUS address always starts with 0 and can go up to 65535.
  It can be used with any function code.
- Certain PLCs lack correct handling of the 0 and therefore add 1 to the address. So their addresses (MODBUS address +1) always start with 1.
- Some SCADA tools add an offset to determine the function code, which shall be used to access the device at the given address. They also add 1 to the MODBUS address sometimes. As an example, address 40001 would be "read MODBUS address 0 with function code $03_{hex}$", 30012 would be "read MODBUS address 11 with function code $04_{hex}$".

The following tables always give the MODBUS addresses mentioned first in above list.

### 2.2.9 Measured Values

The measured values are available beginning from address 0 in intervals of 6 data words. For each value in this table, the maximum and minimum values are also available. To read the maximum, just add 2 to the address of a value, for the minimum add 4. (Example: to get the minimum voltage L1-N, read from address 00034 = 00030+4). The following values can be accessed with function codes $03_{\text{hex and}}$ $04_{\text{hex}}$.

| Adress | Value | Words | Type | Unit |
|---|---|---|---|---|
| 00000 | Frequency (taken from voltage L1-N) | 2 | REAL | Hz |
| 00006 | Current I-L1 | 2 | REAL | A |
| 00012 | Current I-L2 | 2 | REAL | A |
| 00018 | Current I-L3 | 2 | REAL | A |
| 00024 | Current I-N | 2 | REAL | A |
| 00030 | Voltage L1-N | 2 | REAL | V |
| 00036 | Voltage L2-N | 2 | REAL | V |
| 00042 | Voltage L3-N | 2 | REAL | V |
| 00048 | Voltage L1-L2 | 2 | REAL | V |
| 00054 | Voltage L2-L3 | 2 | REAL | V |
| 00060 | Voltage L3-L1 | 2 | REAL | V |
| 00066 | Fundamental current If-L1 | 2 | REAL | A |
| 00072 | Fundamental current If-L2 | 2 | REAL | A |
| 00078 | Fundamental current If-L3 | 2 | REAL | A |
| 00084 | Fundamental current If-N | 2 | REAL | A |
| 00090 | Apparent power S-L1 | 2 | REAL | VA |
| 00096 | Apparent power S-L2 | 2 | REAL | VA |
| 00102 | Apparent power S-L3 | 2 | REAL | VA |
| 00108 | Apparent power S-sum | 2 | REAL | VA |
| 00114 | Active power P-L1 | 2 | REAL | W |
| 00120 | Active power P-L2 | 2 | REAL | W |
| 00126 | Active power P-L3 | 2 | REAL | W |
| 00132 | Active power P-sum | 2 | REAL | W |
| 00138 | Reactive power Q-L1 | 2 | REAL | var |
| 00144 | Reactive power Q-L2 | 2 | REAL | var |
| 00150 | Reactive power Q-L3 | 2 | REAL | var |
| 00156 | Reactive power Q-sum | 2 | REAL | var |

| | | | | |
|---|---|---|---|---|
| 00162 | Power factor pf-L1 | 2 | REAL | - |
| 00168 | Power factor pf-L2 | 2 | REAL | - |
| 00174 | Power factor pf-L3 | 2 | REAL | - |
| 00180 | Power factor pf-sum | 2 | REAL | - |
| 00186 | Fundamental phase angle phi-L1 | 2 | REAL | degrees |
| 00192 | Fundamental phase angle phi-L2 | 2 | REAL | degrees |
| 00198 | Fundamental phase angle phi-L3 | 2 | REAL | degrees |
| 00204 | Fundamental cos φ cp-L1 | 2 | REAL | - |
| 00210 | Fundamental cos φ cp-L2 | 2 | REAL | - |
| 00216 | Fundamental cos φ cp-L3 | 2 | REAL | - |
| 00222 | Total harmonic distortion THD-I1 | 2 | REAL | % |
| 00228 | Total harmonic distortion THD-I2 | 2 | REAL | % |
| 00234 | Total harmonic distortion THD-I3 | 2 | REAL | % |
| 00240 | Total harmonic distortion THD-I-N | 2 | REAL | % |
| 00246 | Total harmonic distortion THD-U1 | 2 | REAL | % |
| 00252 | Total harmonic distortion THD-U2 | 2 | REAL | % |
| 00258 | Total harmonic distortion THD-U3 | 2 | REAL | % |
| 00264 | Ambient temperature TEMP | 2 | REAL | °C |
| 00270 | Damped current Ith-L1 | 2 | REAL | A |
| 00276 | Damped current Ith-L2 | 2 | REAL | A |
| 00282 | Damped current Ith-L3 | 2 | REAL | A |
| 00288 | Damped current Ith-N | 2 | REAL | A |
| 00294 | Damped active power Pth-L1 | 2 | REAL | W |
| 00300 | Damped active power Pth-L2 | 2 | REAL | W |
| 00306 | Damped active power Pth-L3 | 2 | REAL | W |
| 00312 | Damped active power Pth-sum | 2 | REAL | W |

### 2.2.10 Harmonics

Harmonics are stored in separate arrays of FLOAT values for each current / voltage. The table below depicts the corresponding base addresses. Each data array contains 63 values with 2 words length each. The first table entry holds the fundamental wave. After the fundamental wave (= harmonic of order 1), the other harmonics follow up to the 63rd order. Each FLOAT value occupies 2 words, so the address must be increased by 2, in order to retrieve the next value.

If the current or voltage is too small to calculate valid harmonics from it, the value at the base address (= the fundamental) is 0.0%. This indicates, that the higher current or voltage harmonics are also invalid!

The following values can be accessed with function codes $03_{\text{hex and}}$ $04_{\text{hex}}$.

| Adress | Value | Word | Type | Unit |
|---|---|---|---|---|
| 768 | Base address of harmonics I - L1 | 63*2 | REAL | % |
| 898 | Base address of harmonics I - L2 | 63*2 | REAL | % |
| 1028 | Base address of harmonics I - L3 | 63*2 | REAL | % |
| 1158 | Base address of harmonics I - N | 63*2 | REAL | % |
| 1288 | Base address of harmonics U - L1-N | 63*2 | REAL | % |
| 1418 | Base address of harmonics U - L2-N | 63*2 | REAL | % |
| 1548 | Base address of harmonics U - L3-N | 63*2 | REAL | % |

### 2.2.11 Work Counter

The work counters/accumulators are arranged in a special way. This is necessary to protect them from precision degradation. Each counter consists of two parts:

a) A FLOAT-type base counter which simply accumulates/integrates the power. If this counter reaches 1000000.0, the extended counter is increased by one and 1000000.0 is subtracted from the base counter.

b) A LONG-type extended counter, which is used to count portions of MW / Mvar up to $(2^{32} - 1) * 10^6$.

To get the real work value, the extended counter has to be multiplied with 1,000,000 and added to the base counter. This keeps the precision of the FLOAT-type base counter in acceptable range, so no work is lost for big counter values.

The following values can be accessed with function codes $03_{\text{hex and}}$ $04_{\text{hex}}$.

| Adress | Value | Words | Type | Unit |
|---|---|---|---|---|
| 512 | WP import L1 Tariff 1 - base counter WP | 2 | REAL | Wh |
| 514 | import L1 Tariff 1 - extended counter WP | 2 | LONG | MWh |
| 516 | import L2 Tariff 1 - base counter | 2 | REAL | Wh |
| 518 | WP import L2 Tariff 1 - extended counter | 2 | LONG | MWh |
| 520 | WP import L3 Tariff 1 - base counter WP | 2 | REAL | Wh |
| 522 | import L3 Tariff 1 - extended counter WP | 2 | LONG | MWh |
| 524 | import sum Tariff 1 - base counter | 2 | REAL | Wh |
| 526 | WP import sum Tariff 1 - extended counter | | LONG | MWh |
| 528 | WP export L1 Tariff 1 - base counter | 2 | REAL | Wh |
| 530 | WP export L1 Tariff 1 - extended counter | 2 | LONG | MWh |
| 532 | WP export L2 Tariff 1 - base counter | 2 | REAL | Wh |
| 534 | WP export L2 Tariff 1 - extended counter | 2 | LONG | MWh |
| 536 | WP export L3 Tariff 1 - base counter | 2 | REAL | Wh |
| 538 | WP export L3 Tariff 1 - extended counter | 2 | LONG | MWh |
| 540 | WP export sum Tariff 1 - base counter | 2 | REAL | Wh |
| 542 | WP export sum Tariff 1 - extended counter | 2 | LONG | MWh |
| 544 | WQ inductive L1 Tariff 1 - base counter | 2 | REAL | varh |
| 546 | WQ inductive L1 Tariff 1 - extended counter | 2 | LONG | Mvarh |
| 548 | WQ inductive L2 Tariff 1 - base counter | 2 | REAL | varh |
| 550 | WQ inductive L2 Tariff 1 - extended counter | 2 | LONG | Mvarh |
| 552 | WQ inductive L3 Tariff 1 - base counter | 2 | REAL | varh |

| | | | | |
|---|---|---|---|---|
| 554 | WQ inductive L3 Tariff 1 - extended counter | 2 | LONG | Mvarh |
| 556 | WQ inductive sum Tariff 1 - base counter | 2 | REAL | varh |
| 558 | WQ inductive sum Tariff 1 - extended counter | 2 | LONG | Mvarh |
| 560 | WQ capacitive L1 Tariff 1 - base counter | 2 | REAL | varh |
| 562 | WQ capacitive L1 Tariff 1 - extended counter | 2 | LONG | Mvarh |
| 564 | WQ capacitive L2 Tariff 1 - base counter | 2 | REAL | varh |
| 566 | WQ capacitive L2 Tariff 1 - extended counter | 2 | LONG | Mvarh |
| 568 | WQ capacitive L3 Tariff 1 - base counter | 2 | REAL | varh |
| 570 | WQ capacitive L3 Tariff 1 - extended counter | 2 | LONG | Mvarh |
| 572 | WQ capacitive sum Tariff 1 - base counter | 2 | REAL | varh |
| 574 | WQ capacitive sum Tariff 1 - extended counter | 2 | LONG | Mvarh |

From software Revision 1.12.x counters for different tariffs are available. The table below depicts the new counters including the tariffs.

**Attention:** Tariff 2 counters are only available when the device is equipped with option –DM.

| Adress | Value | Words | Type | Unit |
|---|---|---|---|---|
| 2048 | WP import L1 Tarif 1 | 2 | REAL | kWh |
| 2050 | WP import L2 Tarif 1 | 2 | REAL | kWh |
| 2052 | WP import L3 Tarif 1 | 2 | REAL | kWh |
| 2054 | WP import sum Tarif 1 | 2 | REAL | kWh |
| 2056 | WP export L1 Tarif 1 | 2 | REAL | kWh |
| 2058 | WP export L2 Tarif 1 | 2 | REAL | kWh |
| 2060 | WP export L3 Tarif 1 | 2 | REAL | kWh |
| 2062 | WP export sum Tarif 1 | 2 | REAL | kWh |
| 2064 | WQ inductive L1 Tarif 1 | 2 | REAL | kvarh |
| 2066 | WQ inductive L2 Tarif 1 | 2 | REAL | kvarh |
| 2068 | WQ inductive L3 Tarif 1 | 2 | REAL | kvarh |
| 2070 | WQ inductive sum Tarif 1 | 2 | REAL | kvarh |
| 2072 | WQ capacitive L1 Tarif 1 | 2 | REAL | kvarh |
| 2074 | WQ capacitive L2 Tarif 1 | 2 | REAL | kvarh |
| 2076 | WQ capacitive L3 Tarif 1 | 2 | REAL | kvarh |
| 2078 | WQ capacitive sum Tarif 1 | 2 | REAL | kvarh |

| Adress | Value | Words | Type | Unit |
|---|---|---|---|---|
| 2080 | WP import L1 Tariff 2 | 2 | REAL | kWh |
| 2082 | WP import L2 Tariff 2 | 2 | REAL | kWh |
| 2084 | WP import L3 Tariff 2 | 2 | REAL | kWh |
| 2086 | WP import sum Tariff 2 | 2 | REAL | kWh |
| 2088 | WP export L1 Tariff 2 | 2 | REAL | kWh |
| 2090 | WP export L2 Tariff 2 | 2 | REAL | kWh |
| 2092 | WP export L3 Tariff 2 | 2 | REAL | kWh |
| 2094 | WP export sum Tariff 2 | 2 | REAL | kWh |
| 2096 | WQ inductive L1 Tariff 2 | 2 | REAL | kvarh |
| 2098 | WQ inductive L2 Tarif 2 | 2 | REAL | kvarh |
| 2100 | WQ inductive L3 Tarif 2 | 2 | REAL | kvarh |
| 2102 | WQ inductive sum Tarif 2 | 2 | REAL | kvarh |
| 2104 | WQ capacitive L1 Tarif 2 | 2 | REAL | kvarh |
| 2106 | WQ capacitive L2 Tarif 2 | 2 | REAL | kvarh |
| 2108 | WQ capacitive L3 Tarif 2 | 2 | REAL | kvarh |
| 2110 | WQ capacitive sum Tarif 2 | 2 | REAL | kvarh |

!

The device’s memory includes more registers than mentioned above. They can contain important setup data for the device, which is why it is allowed to change the content of the mentioned addresses only.

# 3. Trouble Shooting

If the bus connection isn't working correctly, please check the following points:

a) If there is no communication at all, then the error must be looked for between the EMM5 and the PC! Possible causes can be:
   - Check adjustment of baud rate, parity and address at the EMM5, possibly make changes in the configuration
   - Possibly the bus lines A and B are interchanged, if necessary rectify
   - Check adjustments of the converter RS485/RS232, possibly use the data sheet of the converter
   - Perhaps there is a multiple reservation of the interface at the PC, if necessary stop this multiple reservation
   - Check termination and bias resistors, if necessary rectify

b) Does the cable of the bus connection have any damages? All plug connections are correct? If necessary replace.

c) Is the pin assignment of the RS485 connection correct? If necessary rectify.

d) The shielding of the bus connection may not be grounded. If this is the case, separate the shielding from the grounding.

e) Is the communication possible, but there are problems with the software of the customer, then please check the following points:
   - Check adjustment of bus address, parity and baud rate in the software
   - Check data format