


# VMU-MC

## COMMUNICATION PROTOCOL

Version 1 Revision 2

February 15th, 2018

# Index


1.1 Introduction ..... 3
1.2 MODBUS functions ..... 3
*Function 03h (Read Holding Registers)* ..... *3*
*Function 04h (Read Input Registers)* ..... *4*
*Function 06h (Write Single Holding Register)* ..... *4*
*Function 10h (Write Multiple Register)* ..... *5*
*Function 08h (Diagnostic with sub-function code 00h)* ..... *5*
*Broadcast mode* ..... *6*
1.3 Application notes ..... 7
*RS485 general considerations* ..... *7*
*MODBUS timing* ..... *7*

**2 TABLES ..... 8**

2.1 Data format representation In Carlo Gavazzi instruments ..... 8
2.2 Totalizers ..... 9
2.3 Input status / Active tariff / System status ..... 10
2.4 Firmware version and revision code ..... 11
2.5 Carlo Gavazzi Controls identification code ..... 11
2.6 Password (to locally enter programming menu) ..... 11
2.7 Serial port configuration ..... 12
2.8 Modules configuration ..... 12
2.9 Input configuration ..... 12
2.10 Totalizers “Reset” and “Reset/Overwrite Enable” commands ..... 14
2.11 “Serial configuration update” command ..... 18
2.12 Serial number and production year ..... 18

**3 REVISIONS ..... 19**

## 1.1 Introduction

The RS485 serial interface supports the MODBUS/JBUS (RTU) protocol. In this document only the information necessary to read/write from/to VMU-MC has been reported (not all the parts of the protocol have been implemented).
For a complete description of the MODBUS protocol please refer to the "Modbus_Application_Protocol_V1_1a.pdf" document that is downloadable from the www.modbus.org web site.

## 1.2 MODBUS functions

These functions are available on VMU-MC:

- Reading of n "Holding Registers" (code 03h)
- Reading of n "Input Register" (code 04h)
- Writing of one "Holding Registers" (code 06h)
- Diagnostic (code 08h with sub-function code 00h)
- Broadcast mode (writing instruction on address 00h)

**IMPORTANT:**

1) In this document the "Modbus address" field is indicated in two modes:
   1.1) "**Modicon address**": it is the "6-digit Modicon" representation with Modbus function code 04 (Read Input Registers). It is possible to read the same values with function code 03 (Read Holding Registers) replacing the first digit ("3") with the number "4".
   1.2) "**Physical address**": it is the "word address" value to be included in the communication frame.
2) The functions 03h and 04h have exactly the same effect and can be used indifferently.
3) The communication parameters are to be set according to the configuration of the instrument (refer to VMU-MC instruction manual)

### Function 03h (Read Holding Registers)

This function is used to read the contents of a contiguous block of holding registers (word). The Request frame specifies the starting register address and the number of registers to be read. The register data in the response message are packed as two bytes per register (word), with the binary contents right justified within each byte. For each register, the first byte contains the high order bits (MSB) and the second contains the low order bits (LSB).

Request frame

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7h (1 to 247) | |
| Function code | 1 byte | 03h | |
| Starting address | 2 bytes | 0000h to FFFFh | Byte order: MSB, LSB |
| Quantity of registers (**N** word) | 2 bytes | 1 to 7Dh (1 to 125) | Byte order: MSB, LSB |
| CRC | 2 bytes | | |

Response frame (correct action)

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7h (1 to 247) | |
| Function code | 1 byte | 03h | |
| Quantity of requested bytes | 1 byte | **N** word * 2 | |
| Register value | **N***2 bytes | | Byte order: MSB, LSB |
| CRC | 2 bytes | | |

Response frame (incorrect action)

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7h (1 to 247) | Possible exception :<br>01h: illegal function<br>02h: illegal data address<br>03h: illegal data value<br>04h: slave device failure |
| Function code | 1 byte | 83h | |
| Exception code | 1 byte | 01h, 02h, 03h, 04h (see note) | |
| CRC | 2 bytes | | |

### Function 04h (Read Input Registers)

This function code is used to read the contents of a contiguous block of input registers (word). The Request frame specifies the starting register address and the number of registers to be read.
The register data in the response message are packed as two bytes per register (word), with the binary contents right justified within each byte. For each register, the first byte contains the high order bits (MSB) and the second contains the low order bits (LSB).

Request frame

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7h (1 to 247) | |
| Function code | 1 byte | 04h | |
| Starting address | 2 bytes | 0000h to FFFFh | Byte order: MSB, LSB |
| Quantity of registers (**N** word) | 2 bytes | 1 to 7Dh (1 to 125) | Byte order: MSB, LSB |
| CRC | 2 bytes | | |

Response frame (correct action)

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7h (1 to 247) | |
| Function code | 1 byte | 04h | |
| Quantity of requested bytes | 1 byte | **N** word * 2 | |
| Register value | **N***2 bytes | | Byte order: MSB, LSB |
| CRC | 2 bytes | | |

Response frame (incorrect action)

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7h (1 to 247) | Possible exception :<br>01h: illegal function<br>02h: illegal data address<br>03h: illegal data value<br>04h: slave device failure |
| Function code | 1 byte | 84h | |
| Exception code | 1 byte | 01h, 02h, 03h, 04h | |
| CRC | 2 bytes | | |

### Function 06h (Write Single Holding Register)

This function code is used to write a single holding register. The Request frame specifies the address of the register (word) to be written and its content.
The correct response is an echo of the request, returned after the register content has been written.

Request frame

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7h (1 to 247) | |
| Function code | 1 byte | 06h | |
| Starting address | 2 bytes | 0000h to FFFFh | Byte order: MSB, LSB |
| Register value | 2 bytes | 0000h to FFFFh | Byte order: MSB, LSB |
| CRC | 2 bytes | | |

Response frame (correct action)

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7h (1 to 247) | |
| Function code | 1 byte | 06h | |
| Starting address | 2 bytes | 0000h to FFFFh | Byte order: MSB, LSB |
| Register value | 2 bytes | 0000h to FFFFh | Byte order: MSB, LSB |
| CRC | 2 bytes | | |

Response frame (incorrect action)

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7h (1 to 247) | Possible exception :<br>01h: illegal function<br>02h: illegal data address<br>03h: illegal data value<br>04h: slave device failure |
| Function code | 1 byte | 86h | |
| Exception code | 1 byte | 01h, 02h, 03h, 04h | |
| CRC | 2 bytes | | |

**Function 10h (Write Multiple Register)**

This function code is used to write a block of contiguous registers (maximum 120). The requested values to be written are specified in the request data field. Data is packed as two bytes per register.
The correct response returns the function code, starting address, and the quantity of written registers.

Request frame

| Description | Length | Value | Note |
|---|---|---|---|
| Physical Address | 1 byte | 1 to F7 (1 to 255) | |
| Function code | 1 byte | 10h | |
| Starting Address | 2 bytes | 0000h to FFFFh | Byte order: MSB, LSB |
| Quantity of Registers (N word) | 2 bytes | 0001h to 0078h | Byte order: MSB, LSB |
| Byte count | 1 byte | N word * 2 | |
| Register value | N * 2 bytes | value | Byte order: MSB, LSB |
| CRC | 2 bytes | | |

Response frame (correct action)

| Description | Length | Value | Note |
|---|---|---|---|
| Physical Address | 1 byte | 1 to F7 (1 to 255) | |
| Function code | 1 byte | 10h | |
| Starting Address | 2 bytes | 0000h to FFFFh | Byte order: MSB, LSB |
| Quantity of Registers (N word) | 2 bytes | 0001h to 0078h | Byte order: MSB, LSB |
| CRC | 2 bytes | | |

Response frame (incorrect action)

| Description | Length | Value | Note |
|---|---|---|---|
| Physical Address | 1 byte | 1 to F7 (1 to 255) | Possible exception :<br>01h: illegal function<br>02h: illegal data address<br>03h: illegal data value<br>04h: slave device failure |
| Function code | 1 byte | 90h | |
| Exception code | 1 byte | 01h, 02h, 03h, 04h | |
| CRC | 2 bytes | | |

**Function 08h (Diagnostic with sub-function code 00h)**

MODBUS function 08h provides a series of tests to check the communication system between a client (Master) device and a server (Slave), or to check various internal error conditions in a server.
VMU-MC supports only 0000h sub-function code (Return Query Data). With this sub-function the data passed in the request data field is to be returned (looped back) in the response. The entire response message should be identical to the request.

Request frame

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7h (1 to 247) | |
| Function code | 1 byte | 08h | |
| Sub-function | 2 bytes | 0000h | |
| Data (**N** word) | **N** *2 bytes | Data | Byte order: MSB, LSB |
| CRC | 2 bytes | | |

Response frame (correct action)

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7 (1 to 247) | |
| Function code | 1 byte | 08h | |
| Sub-function | 2 bytes | 0000h | |

| | | | |
|---|---|---|---|
| Data (**N** word) | **N** *2 bytes | Data | Byte order: MSB, LSB |
| CRC | 2 bytes | | |

Response frame (incorrect action)

| Description | Length | Value | Note |
|---|---|---|---|
| Physical address | 1 byte | 1 to F7h (1 to 247) | Possible exception :<br>01h: illegal function<br>02h: illegal data address<br>03h: illegal data value<br>04h: slave device failure |
| Function code | 1 byte | 88h | |
| Exception code | 1 byte | 01h, 02h, 03h, 04h | |
| CRC | 2 bytes | | |

**Broadcast mode**

In broadcast mode the master can send a request (command) to all the slaves. No response is returned to broadcast requests sent by the master. It is possible to send the broadcast message only with function code 06h using address 00h.

## 1.3 Application notes

### RS485 general considerations

1. To avoid errors due to the signal reflections or line coupling, it is necessary to terminate the bus at the beginning and at the end (inserting a 120 ohm 1/2W 5% resistor between line B and A in the last instrument and in the Host interface or by using the accessory terminal block provided with the product).
2. The network termination is necessary even in case of point-to-point connection and/or of short distances.
3. For connections longer than 1000m or if in the network there are more than 160 instruments (with 1/5 unit load as used in VMU-MC interface), a signal repeater is necessary.
4. For bus connection it is suggested to use an AWG24 balanced pair cable and to add a third wire for GND connection. Connect GND to the shield if a shielded cable is used.
5. The GND is to be connected to ground only at the host side.
6. If an instrument does not answer within the “max answering time”, it is necessary to repeat the query. If the instrument does not answer after 2 or 3 consecutive queries, it is to be considered as not connected, faulty or reached with a wrong address. The same consideration is valid in case of CRC errors or incomplete response frames.

### MODBUS timing

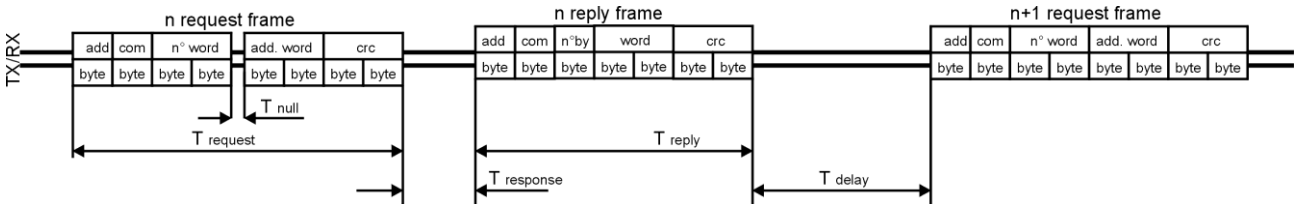



Fig. 1 : 2-wire timing diagram

| Timing characteristics of reading function: | msec |
|---|---|
| T response: Max answering time | 500ms |
| T response: Typical answering time | 40ms |
| T delay: Minimum time before a new query | 3.5char |
| T null: Max interruption time during the request frame | 2.5char |

# 2 TABLES

## 2.1 Data format representation In Carlo Gavazzi instruments

The variables are represented by integers or floating numbers, with 2's complement notation in case of "signed" format, using the following:

| Format | IEC data type | Description | Bits | Range |
|---|---|---|---|---|
| INT16 | INT | Integer | 16 | -32768 .. 32767 |
| UINT16 | UINT | Unsigned integer | 16 | 0 .. 65535 |
| INT32 | DINT | Double integer | 32 | $-2^{31}$ .. $2^{31}$ |
| UINT32 | UDINT | Unsigned double int | 32 | 0 .. $2^{32}-1$ |
| UINT64 | ULINT | Unsigned long integer | 64 | 0 .. $2^{64}-1$ |
| IEEE754 SP | | Single-precision floating-point | 32 | $-(1+[1-2^{-23}])x2^{127}$ .. $2^{128}$ |

For all the formats the byte order (inside the single word) is MSB->LSB. In INT32, UINT32 and UINT64 formats, the word order is LSW-> MSW.

## 2.2 Totalizers

The totalizers values are given as unsigned integer without the decimal point; to obtain the correct value the number has to be divided by the relevant factor.
Each totalizer can be overwritten after the operation has been enabled with the appropriate command (see 2.10).

**MODBUS**: read and write mode with functions code 0x03, 0x04, 0x06, 0x10 Table 2.2-1

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| 300001 | 0000h | 2 | VMU-MC: Cnt_tot_In1 | UINT32 | Totalizers: total<br>Decimal point position:<br>See Table 2.9-1 |
| 300003 | 0002h | 2 | VMU-MC: Cnt_tot_In2 | UINT32 | |
| 300005 | 0004h | 2 | VMU-OC, pos. 1: Cnt_tot_In1 | UINT32 | |
| 300007 | 0006h | 2 | VMU-OC, pos. 1: Cnt_tot_In2 | UINT32 | |
| 300009 | 0008h | 2 | VMU-OC, pos. 1: Cnt_tot_In3 | UINT32 | |
| 300011 | 000Ah | 2 | VMU-OC, pos. 2: Cnt_tot_In1 | UINT32 | |
| 300013 | 000Ch | 2 | VMU-OC, pos. 2: Cnt_tot_In2 | UINT32 | |
| 300015 | 000Eh | 2 | VMU-OC, pos. 2: Cnt_tot_In3 | UINT32 | |
| 300017 | 0010h | 2 | VMU-OC, pos. 3: Cnt_tot_In1 | UINT32 | |
| 300019 | 0012h | 2 | VMU-OC, pos. 3: Cnt_tot_In2 | UINT32 | |
| 300021 | 0014h | 2 | VMU-OC, pos. 3: Cnt_tot_In3 | UINT32 | |
| 300023 | 0016h | 2 | VMU-MC: Cnt_T1_In1 | UINT32 | Totalizers: tariff (T1, T2, T3, T4)<br>Decimal point position:<br>See Table 2.9-1 |
| 300025 | 0018h | 2 | VMU-MC: Cnt_T2_In1 | UINT32 | |
| 300027 | 001Ah | 2 | VMU-MC: Cnt_T3_In1 | UINT32 | |
| 300029 | 001Ch | 2 | VMU-MC: Cnt_T4_In1 | UINT32 | |
| 300031 | 001Eh | 2 | VMU-MC: Cnt_T1_In2 | UINT32 | |
| 300033 | 0020h | 2 | VMU-MC: Cnt_T2_In2 | UINT32 | |
| 300035 | 0022h | 2 | VMU-MC: Cnt_T3_In2 | UINT32 | |
| 300037 | 0024h | 2 | VMU-MC: Cnt_T4_In2 | UINT32 | |
| 300039 | 0026h | 2 | VMU-OC, pos. 1: Cnt_T1_In1 | UINT32 | |
| 300041 | 0028h | 2 | VMU-OC, pos. 1: Cnt_T2_In1 | UINT32 | |
| 300043 | 002Ah | 2 | VMU-OC, pos. 1: Cnt_T3_In1 | UINT32 | |
| 300045 | 002Ch | 2 | VMU-OC, pos. 1: Cnt_T4_In1 | UINT32 | |
| 300047 | 002Eh | 2 | VMU-OC, pos. 1: Cnt_T1_In2 | UINT32 | |
| 300049 | 0030h | 2 | VMU-OC, pos. 1: Cnt_T2_In2 | UINT32 | |
| 300051 | 0032h | 2 | VMU-OC, pos. 1: Cnt_T3_In2 | UINT32 | |
| 300053 | 0034h | 2 | VMU-OC, pos. 1: Cnt_T4_In2 | UINT32 | |
| 300055 | 0036h | 2 | VMU-OC, pos. 1: Cnt_T1_In3 | UINT32 | |
| 300057 | 0038h | 2 | VMU-OC, pos. 1: Cnt_T2_In3 | UINT32 | |
| 300059 | 003Ah | 2 | VMU-OC, pos. 1: Cnt_T3_In3 | UINT32 | |
| 300061 | 003Ch | 2 | VMU-OC, pos. 1: Cnt_T4_In3 | UINT32 | |
| 300063 | 003Eh | 2 | VMU-OC, pos. 2: Cnt_T1_In1 | UINT32 | |
| 300065 | 0040h | 2 | VMU-OC, pos. 2: Cnt_T2_In1 | UINT32 | |
| 300067 | 0042h | 2 | VMU-OC, pos. 2: Cnt_T3_In1 | UINT32 | |
| 300069 | 0044h | 2 | VMU-OC, pos. 2: Cnt_T4_In1 | UINT32 | |
| 300071 | 0046h | 2 | VMU-OC, pos. 2: Cnt_T1_In2 | UINT32 | |
| 300073 | 0048h | 2 | VMU-OC, pos. 2: Cnt_T2_In2 | UINT32 | |
| 300075 | 004Ah | 2 | VMU-OC, pos. 2: Cnt_T3_In2 | UINT32 | |
| 300077 | 004Ch | 2 | VMU-OC, pos. 2: Cnt_T4_In2 | UINT32 | |
| 300079 | 004Eh | 2 | VMU-OC, pos. 2: Cnt_T1_In3 | UINT32 | |
| 300081 | 0050h | 2 | VMU-OC, pos. 2: Cnt_T2_In3 | UINT32 | |
| 300083 | 0052h | 2 | VMU-OC, pos. 2: Cnt_T3_In3 | UINT32 | |
| 300085 | 0054h | 2 | VMU-OC, pos. 2: Cnt_T4_In3 | UINT32 | |
| 300087 | 0056h | 2 | VMU-OC, pos. 3: Cnt_T1_In1 | UINT32 | |
| 300089 | 0058h | 2 | VMU-OC, pos. 3: Cnt_T2_In1 | UINT32 | |
| 300091 | 005Ah | 2 | VMU-OC, pos. 3: Cnt_T3_In1 | UINT32 | |
| 300093 | 005Ch | 2 | VMU-OC, pos. 3: Cnt_T4_In1 | UINT32 | |
| 300095 | 005Eh | 2 | VMU-OC, pos. 3: Cnt_T1_In2 | UINT32 | |
| 300097 | 0060h | 2 | VMU-OC, pos. 3: Cnt_T2_In2 | UINT32 | |
| 300087 | 0062h | 2 | VMU-OC, pos. 3: Cnt_T3_In2 | UINT32 | |
| 300089 | 0064h | 2 | VMU-OC, pos. 3: Cnt_T4_In2 | UINT32 | |
| 300091 | 0066h | 2 | VMU-OC, pos. 3: Cnt_T1_In3 | UINT32 | |
| 300093 | 0068h | 2 | VMU-OC, pos. 3: Cnt_T2_In3 | UINT32 | |
| 300095 | 006Ah | 2 | VMU-OC, pos. 3: Cnt_T3_In3 | UINT32 | |
| 300097 | 006Ch | 2 | VMU-OC, pos. 3: Cnt_T4_In3 | UINT32 | |

## 2.3 Input status / Active tariff / System status

**MODBUS**: read only mode with functions code 03 and 04 Table 2.3-1

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| 300257 | 0100h | 1 | Digital input status | UINT16 | Bit 0: VMU-MC, In1<br>Bit 1: VMU-MC, In2<br>Bit 2: VMU-OC pos. 1, In1<br>Bit 3: VMU-OC pos. 1, In2<br>Bit 4: VMU-OC pos. 1, In3<br>Bit 5: VMU-OC pos. 2, In1<br>Bit 6: VMU-OC pos. 2, In2<br>Bit 7: VMU-OC pos. 2, In3<br>Bit 8: VMU-OC pos. 3, In1<br>Bit 9: VMU-OC pos. 3, In2<br>Bit 10: VMU-OC pos. 3, In3<br>Bit 11 – Bit 15: unused<br><br>0 = input not active<br>1 = input active<br>(see also input HW configuration) |
| 300258 | 0101h | 1 | Overrun status Input, VMU-MC, In1 | UINT16 | The bit sequence indicates the overrun read condition of the last 16 input changes (bit15 the furthest):<br><br>Bit = 0: no overrun<br>Bit = 1: overrun<br><br>Bit overrun detect when polling frequency protocol from PC is too low if compared with the input status change frequency. Bit overrun is set when the input status is change and PC has not been updated with information of the change. |
| 300259 | 0102h | 1 | Overrun status Input, VMU-MC, In2 | UINT16 | |
| 300260 | 0103h | 1 | Overrun status Input, VMU-OC pos. 1, In1 | UINT16 | |
| 300261 | 0104h | 1 | Overrun status Input, VMU-OC pos. 1, In2 | UINT16 | |
| 300262 | 0105h | 1 | Overrun status Input, VMU-OC pos. 1, In3 | UINT16 | |
| 300263 | 0106h | 1 | Overrun status Input, VMU-OC pos. 2, In1 | UINT16 | |
| 300264 | 0107h | 1 | Overrun status Input, VMU-OC pos. 2, In2 | UINT16 | |
| 300265 | 0108h | 1 | Overrun status Input, VMU-OC pos. 2, In3 | UINT16 | |
| 300266 | 0109h | 1 | Overrun status Input, VMU-OC pos. 3, In1 | UINT16 | |
| 300267 | 010Ah | 1 | Overrun status Input, VMU-OC pos. 3, In2 | UINT16 | |
| 300268 | 010Bh | 1 | Overrun status Input, VMU-OC pos. 3, In3 | UINT16 | |
| 300269 | 010Ch | 1 | Active tariff | UINT16 | 0 = T1<br>1 = T2<br>2 = T3<br>3 = T4<br>FFFF = No tariff<br>If no inputs is configured as tariff, and no tariff selected via serial, the value is 0xFFFF. |
| 300270 | 010Dh | 1 | System status | UINT16 | Bit 0: 0<br>Bit 1: VMU-OC pos. 1<br>Bit 2: VMU-OC pos. 2<br>Bit 3: VMU-OC pos. 3<br>Bit 4 – 15: 0 (unused)<br><br>Bit = 1: module error (configured but not detected or detected but not configured)<br>Bit = 0: module ok (configured and detected or not configured and not detected). |

## 2.4 Firmware version and revision code

**MODBUS**: read only mode with functions code 03 and 04 Table 2.4-1

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| 300771 | 0300h | 1 | Version code VMU-MC | UINT 16 | Value= 65: Version "A" (ASCI table)<br>Value=66: Version "B"<br>...<br><br>First standard release is "A"(value = 65) |
| 300772 | 0301h | 1 | Revision code VMU-MC | UINT 16 | Value=0: Revision 0<br>Value=1: Revision 1<br>...<br><br>First standard release is 0 (value = 0) |
| 300773 | 0302h | 1 | Version code Module pos 1 | UINT 16 | Value= 65: Version "A" (ASCI table)<br>Value=66: Version "B"<br>...<br><br>First standard release is "A"(value = 65) |
| 300774 | 0303h | 1 | Revision code Module pos 1 | UINT 16 | Value=0: Revision 0<br>Value=1: Revision 1<br>...<br><br>First standard release is 0 (value = 0) |
| 300775 | 0304h | 1 | Version code Module pos 2 | UINT 16 | Value= 65: Version "A" (ASCI table)<br>Value=66: Version "B"<br>...<br><br>First standard release is "A"(value = 65) |
| 300776 | 0305h | 1 | Revision code Module pos 2 | UINT 16 | Value=0: Revision 0<br>Value=1: Revision 1<br>...<br><br>First standard release is 0 (value = 0) |
| 300777 | 0306h | 1 | Version code Module pos 3 | UINT 16 | Value= 65: Version "A" (ASCI table)<br>Value=66: Version "B"<br>...<br><br>First standard release is "A"(value = 65) |
| 300778 | 0307h | 1 | Revision code Module pos 3 | UINT 16 | Value=0: Revision 0<br>Value=1: Revision 1<br>...<br><br>First standard release is 0 (value = 0) |

## 2.5 Carlo Gavazzi Controls identification code

**MODBUS**: read only mode with functions code 03 and 04 limited to a word at a time Table 2.5-1

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| **3**00012 | 000Bh | 1 | Carlo Gavazzi Controls identification code | UINT 16 | Value = 105<br>(69h) |

.

## 2.6 Password (to locally enter programming menu)

**MODBUS**: read and write mode Table 2.6-1

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| **3**04097 | 1000h | 1 | PASSWORD | UINT 16 | Minimum valid value: 0d<br>Maximum valid value: 9999d<br>Default: 0d |

## 2.7 Serial port configuration

**MODBUS**: read and write mode

Table 2.7-1

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| **3**08193 | 2000h | 1 | RS485 instrument address | UINT 16 | Value min = 1<br>Value max = 247<br>Default: 1 |
| **3**08194 | 2001h | 1 | RS485 baud rate | UINT 16 | Value 0 = 9.6 kbps (default)<br>Value 1 = 19.2 kbps<br>Value 2 = 38.4 kbps<br>Default: 0 (9.6 kbps) |
| **3**08195 | 2002h | 1 | RS485 parity | UINT 16 | Value 0 = no parity (default)<br>Value 1 = even parity<br>Value 2 = odd parity<br>Default: 0 (no parity) |
| **3**08196 | 2003h | 1 | RS485 Stop bit | UINT 16 | Value 0 = 1 stop bit (default)<br>Value 1 = 2 stop bit<br>Default: 0 (1 stop bit) |

The values are updated after a power on or after a “serial configuration update” command (ref. tab command)

## 2.8 Modules configuration

**MODBUS**: read and write mode

Table 2.8-1

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| **3**08449 | 2100h | 1 | Working Mode | UINT 16 | Bit 0-1: VMU-MC input configuration<br>0 = In1 and In2 as counter/status AND tariff by serial communication DISABLED<br>1 = In1 as Tariff, In2 as counter/status<br>2 = In1 and In2 as Tariff<br>3 = In1 and In2 as counter/status AND tariff by serial communication ENABLED<br><br>Bit 2-3: VMU-OC modules connect<br>0 .. 3<br><br>Default: 0000h |
| **3**08705 | 2200h | 1 | Tariff managed via serial communication | UINT 16 located in RAM | From Power ON value is 0xFFFF (no tariff)<br>Valid Values from 0 to 3 (Tariff setting)<br><br>always writeable<br><br>valid only if Working Mode Bits 0-1 = 11 VMU-MC configuration = 3 (no input configured as tariff but tariff by serial ENABLED) |

## 2.9 Input configuration

**MODBUS**: read and write mode

Table 2.9-1

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| 312289 | 3000h | 1 | Pulse Weight VMU-MC In1 | UINT 16 | Pulse Weight: 0..65535<br>Each pulse counted is multiplied by the relevant Pulse Weight and added to the Totalizer (total and tariff)<br><br>Default = 1 |
| 312290 | 3001h | 1 | Pulse Weight VMU-MC In2 | UINT 16 | |
| 312291 | 3002h | 1 | Pulse Weight VMU-OC pos.1, In1 | UINT 16 | |
| 312292 | 3003h | 1 | Pulse Weight VMU-OC pos.1, In2 | UINT 16 | |
| 312293 | 3004h | 1 | Pulse Weight VMU-OC pos.1, In3 | UINT 16 | |
| 312294 | 3005h | 1 | Pulse Weight VMU-OC pos.2, In1 | UINT 16 | |

| | | | | | |
|---|---|---|---|---|---|
| 312295 | 3006h | 1 | Pulse Weight<br>VMU-OC pos.2, In2 | UINT 16 | |
| 312296 | 3007h | 1 | Pulse Weight<br>VMU-OC pos.2, In3 | UINT 16 | |
| 312297 | 3008h | 1 | Pulse Weight<br>VMU-OC pos.3, In1 | UINT 16 | |
| 312298 | 3009h | 1 | Pulse Weight<br>VMU-OC pos.3, In2 | UINT 16 | |
| 312299 | 300Ah | 1 | Pulse Weight<br>VMU-OC pos.3, In3 | UINT 16 | |
| 312305 | 3010h | 1 | Decimal Point position<br>VMU-MC, In1 | UINT 16 | Values:<br>0: no decimals (integer)<br>1: 0,1<br>2: 0,01<br>3: 0,001<br>…<br>9: 0,000000001<br>10 and above: Reserved<br><br>Default: 0<br><br>Note: these values are only for external data interpretation, not internally applied and/or displayed |
| 312306 | 3011h | 1 | Decimal Point position<br>VMU-MC, In2 | UINT 16 | |
| 312307 | 3012h | 1 | Decimal Point position<br>VMU-OC, pos. 1, In1 | UINT 16 | |
| 312308 | 3013h | 1 | Decimal Point position<br>VMU-OC, pos. 1, In2 | UINT 16 | |
| 312309 | 3014h | 1 | Decimal Point position<br>VMU-OC, pos. 1, In3 | UINT 16 | |
| 312310 | 3015h | 1 | Decimal Point position<br>VMU-OC, pos. 2, In1 | UINT 16 | |
| 312311 | 3016h | 1 | Decimal Point position<br>VMU-OC, pos. 2, In2 | UINT 16 | |
| 312312 | 3017h | 1 | Decimal Point position<br>VMU-OC, pos. 2, In3 | UINT 16 | |
| 312313 | 3018h | 1 | Decimal Point position<br>VMU-OC, pos. 3, In1 | UINT 16 | |
| 312314 | 3019h | 1 | Decimal Point position<br>VMU-OC, pos. 3, In2 | UINT 16 | |
| 312315 | 301Ah | 1 | Decimal Point position<br>VMU-OC, pos. 3, In3 | UINT 16 | |
| 312321 | 3020h | 1 | Totalizer Base Unit<br>VMU-MC, In1 | UINT 16 | Values:<br>0: kWh<br>1: kVarh<br>2: kVAh<br>3: kJ<br>4: kcal<br>5: $m^3$ (cubic meter)<br>6: $Nm^3$ (Normal cubic meter)<br>7: hr (hour counter)<br>8: pcs (pieces)<br>9: kg<br>10 : Reserved<br>…<br>999: Reserved<br>1000 and above: Free<br><br>Default = 0<br><br>Note: these units are only for external data interpretation, not internally applied and/or displayed |
| 312322 | 3021h | 1 | Totalizer Base Unit<br>VMU-MC, In2 | UINT 16 | |
| 312323 | 3022h | 1 | Totalizer Base Unit<br>VMU-OC, pos. 1, In1 | UINT 16 | |
| 312324 | 3023h | 1 | Totalizer Base Unit<br>VMU-OC, pos. 1, In2 | UINT 16 | |
| 312325 | 3024h | 1 | Totalizer Base Unit<br>VMU-OC, pos. 1, In3 | UINT 16 | |
| 312326 | 3025h | 1 | Totalizer Base Unit<br>VMU-OC, pos. 2, In1 | UINT 16 | |
| 312327 | 3026h | 1 | Totalizer Base Unit<br>VMU-OC, pos. 2, In2 | UINT 16 | |
| 312328 | 3027h | 1 | Totalizer Base Unit<br>VMU-OC, pos. 2, In3 | UINT 16 | |
| 312329 | 3028h | 1 | Totalizer Base Unit<br>VMU-OC, pos. 3, In1 | UINT 16 | |
| 312330 | 3029h | 1 | Totalizer Base Unit<br>VMU-OC, pos. 3, In2 | UINT 16 | |
| 312331 | 302Ah | 1 | Totalizer Base Unit<br>VMU-OC, pos. 3, In3 | UINT 16 | |
| 312337 | 3030h | 1 | Input filter setting<br>VMU-MC | UINT 16 | Filtering apply to all the inputs belonging to the specified module.<br>LSB: Ton filter<br>MSB: Toff filter<br>Value:<br>0: min<br>1: 10ms<br>2: 20ms<br>3: 30ms<br>4: 40ms<br>5: 50ms<br>6: 100ms<br>7: 200ms<br>8: 300ms<br><br>Default: 0303h (30ms/30ms) |
| 312338 | 3031h | 1 | Input filter setting<br>VMU-OC, pos. 1 | UINT 16 | |
| 312339 | 3032h | 1 | Input filter setting<br>VMU-OC, pos. 2 | UINT 16 | |
| 312340 | 3033h | 1 | Input filter setting<br>VMU-OC, pos. 3 | UINT 16 | |
| 312353 | 3040h | 1 | Digital input HW logic | UINT16 | Bit 0: VMU-MC, In1 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | Bit 1: VMU-MC, In2<br>Bit 2: VMU-OC pos. 1, In1<br>Bit 3: VMU-OC pos. 1, In2<br>Bit 4: VMU-OC pos. 1, In3<br>Bit 5: VMU-OC pos. 2, In1<br>Bit 6: VMU-OC pos. 2, In2<br>Bit 7: VMU-OC pos. 2, In3<br>Bit 8: VMU-OC pos. 3, In1<br>Bit 9: VMU-OC pos. 3, In2<br>Bit 10: VMU-OC pos. 3, In3<br>Bit 11 – Bit 15: unused<br><br>Bit = 0: logic not inverted<br>Bit = 1: inverted logic<br><br>Default: 0000h |
| 312369 | 3050h | 1 | Totalizers Reset/Overwrite Enable Mask | UINT16 | Bit 0: VMU-MC, In1<br>Bit 1: VMU-MC, In2<br>Bit 2: VMU-OC pos. 1, In1 totalizers<br>Bit 3: VMU-OC pos. 1, In2 totalizers<br>Bit 4: VMU-OC pos. 1, In3 totalizers<br>Bit 5: VMU-OC pos. 2, In1 totalizers<br>Bit 6: VMU-OC pos. 2, In2 totalizers<br>Bit 7: VMU-OC pos. 2, In3 totalizers<br>Bit 8: VMU-OC pos. 3, In1 totalizers<br>Bit 9: VMU-OC pos. 3, In2 totalizers<br>Bit 10: VMU-OC pos. 3, In3 totalizers<br>Bit 11 – Bit 15: unused<br><br>Bit = 0: Reset/Overwrite Disabled for the totalizers correspondent to the bit position<br>Bit = 1: Reset/Overwrite Enabled for the totalizers correspondent to the bit position<br><br>The enable/disable flag is valid for both total and tariff totalizers.<br><br>Default: 0000h (reset/overwrite commands are disabled for all totalizers) |

## 2.10 Totalizers “Reset” and “Reset/Overwrite Enable” commands

***Reset/Overwrite enable commands:***
If allowed by the “Totalizers Reset/Overwrite Enable Mask” settings, after writing 1 in the bit corresponding to a totalizer (total or tariff), it can be reset or overwrite within a fixed timeout of 3 s (see 2.2).
The commands are read/write mode.
After the timeout expires or once the totalizer has been reset or written, the command bits are automatically reset (Bit = 0).

***Reset commands:***
If allowed by the “Totalizers Reset/Overwrite Enable Mask” settings, write 1, within the timeout after the Reset/Overwrite command, in the bit corresponding to a totalizer (total or tariff) cause its immediate reset (totalizer = 0). If the bit is 0, the relevant totalizer is not affected.
The commands are read/write mode.
After the execution of the reset, the command bits are automatically reset (Bit = 0).

**MODBUS**: read and write mode

Table 2.10-1

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| **3**16385 | 4000h | 1 | Reset totalizers - Total | UINT 16 | Bit 0: VMU-MC, Cnt_tot_In1<br>Bit 1: VMU-MC, Cnt_tot_In2<br>Bit 2: VMU-OC pos.1, Cnt_tot_In1<br>Bit 3: VMU-OC pos.1, Cnt_tot_In2<br>Bit 4: VMU-OC pos.1, Cnt_tot_In3<br>Bit 5: VMU-OC pos.1, Cnt_tot_In4<br>Bit 6: VMU-OC pos.2, Cnt_tot_In1<br>Bit 7: VMU-OC pos.2, Cnt_tot_In2<br>Bit 8: VMU-OC pos.2, Cnt_tot_In3<br>Bit 9: VMU-OC pos.2, Cnt_tot_In4<br>Bit 10: VMU-OC pos.3, Cnt_tot_In1<br>Bit 11: VMU-OC pos.3, Cnt_tot_In2<br>Bit 12: VMU-OC pos.3, Cnt_tot_In3<br>Bit 13: VMU-OC pos.3, Cnt_tot_In4<br>Bit 14 – 15: Not used<br><br>Bit = 1, reset totalizer command<br>Bit = 0, no reset command<br>Bits automatically reset after command execution |
| **3**16386 | 4001h | 1 | Reset totalizers – Tariff T1 | UINT 16 | Bit 0: VMU-MC, Cnt_T1_In1<br>Bit 1: VMU-MC, Cnt_T1_In2<br>Bit 2: VMU-OC pos.1, Cnt_T1_In1<br>Bit 3: VMU-OC pos.1, Cnt_T1_In2<br>Bit 4: VMU-OC pos.1, Cnt_T1_In3<br>Bit 5: VMU-OC pos.1, Cnt_T1_In4<br>Bit 6: VMU-OC pos.2, Cnt_T1_In1<br>Bit 7: VMU-OC pos.2, Cnt_T1_In2<br>Bit 8: VMU-OC pos.2, Cnt_T1_In3<br>Bit 9: VMU-OC pos.2, Cnt_T1_In4<br>Bit 10: VMU-OC pos.3, Cnt_T1_In1<br>Bit 11: VMU-OC pos.3, Cnt_T1_In2<br>Bit 12: VMU-OC pos.3, Cnt_T1_In3<br>Bit 13: VMU-OC pos.3, Cnt_T1_In4<br>Bit 14 – 15: Not used<br><br>Bit = 1 reset totalizer command<br>Bit = 0, no reset command<br>Bits automatically reset after command execution |
| **3**16387 | 4002h | 1 | Reset totalizers – Tariff T2 | UINT 16 | Bit 0: VMU-MC, Cnt_T2_In1<br>Bit 1: VMU-MC, Cnt_T2_In2<br>Bit 2: VMU-OC pos.1, Cnt_T2_In1<br>Bit 3: VMU-OC pos.1, Cnt_T2_In2<br>Bit 4: VMU-OC pos.1, Cnt_T2_In3<br>Bit 5: VMU-OC pos.1, Cnt_T2_In4<br>Bit 6: VMU-OC pos.2, Cnt_T2_In1<br>Bit 7: VMU-OC pos.2, Cnt_T2_In2<br>Bit 8: VMU-OC pos.2, Cnt_T2_In3<br>Bit 9: VMU-OC pos.2, Cnt_T2_In4<br>Bit 10: VMU-OC pos.3, Cnt_T2_In1<br>Bit 11: VMU-OC pos.3, Cnt_T2_In2<br>Bit 12: VMU-OC pos.3, Cnt_T2_In3<br>Bit 13: VMU-OC pos.3, Cnt_T2_In4<br>Bit 14 – 15: Not used<br><br>Bit = 1 reset totalizer command<br>Bit = 0, no reset command<br>Bits automatically reset after command execution |

| | | | | | |
|---|---|---|---|---|---|
| **3**16388 | 4003h | 1 | Reset totalizers – Tariff T3 | UINT 16 | Bit 0: VMU-MC, Cnt_T3_In1<br>Bit 1: VMU-MC, Cnt_T3_In2<br>Bit 2: VMU-OC pos.1, Cnt_T3_In1<br>Bit 3: VMU-OC pos.1, Cnt_T3_In2<br>Bit 4: VMU-OC pos.1, Cnt_T3_In3<br>Bit 5: VMU-OC pos.1, Cnt_T3_In4<br>Bit 6: VMU-OC pos.2, Cnt_T3_In1<br>Bit 7: VMU-OC pos.2, Cnt_T3_In2<br>Bit 8: VMU-OC pos.2, Cnt_T3_In3<br>Bit 9: VMU-OC pos.2, Cnt_T3_In4<br>Bit 10: VMU-OC pos.3, Cnt_T3_In1<br>Bit 11: VMU-OC pos.3, Cnt_T3_In2<br>Bit 12: VMU-OC pos.3, Cnt_T3_In3<br>Bit 13: VMU-OC pos.3, Cnt_T3_In4<br>Bit 14 – 15: Not used<br><br>Bit = 1, reset totalizer command<br>Bit = 0, no reset command<br>Bits automatically reset after command execution |
| **3**16389 | 4004h | 1 | Reset totalizers – Tariff T4 | UINT 16 | Bit 0: VMU-MC, Cnt_T4_In1<br>Bit 1: VMU-MC, Cnt_T4_In1<br>Bit 2: VMU-OC pos.1, Cnt_T4_In1<br>Bit 3: VMU-OC pos.1, Cnt_T4_In2<br>Bit 4: VMU-OC pos.1, Cnt_T4_In3<br>Bit 5: VMU-OC pos.1, Cnt_T4_In4<br>Bit 6: VMU-OC pos.2, Cnt_T4_In1<br>Bit 7: VMU-OC pos.2, Cnt_T4_In2<br>Bit 8: VMU-OC pos.2, Cnt_T4_In3<br>Bit 9: VMU-OC pos.2, Cnt_T4_In4<br>Bit 10: VMU-OC pos.3, Cnt_T4_In1<br>Bit 11: VMU-OC pos.3, Cnt_T4_In2<br>Bit 12: VMU-OC pos.3, Cnt_T4_In3<br>Bit 13: VMU-OC pos.3, Cnt_T4_In4<br>Bit 14 – 15: Not used<br><br>Bit = 1, reset totalizer command<br>Bit = 0, no reset command<br>Bits automatically reset after command execution |

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| **3**16390 | 4100h | 1 | Reset/Overwrite totalizers enable – Total | UINT 16 | Bit 0: VMU-MC, Cnt_tot_In1<br>Bit 1: VMU-MC, Cnt_tot_In2<br>Bit 2: VMU-OC pos.1, Cnt_tot_In1<br>Bit 3: VMU-OC pos.1, Cnt_tot_In2<br>Bit 4: VMU-OC pos.1, Cnt_tot_In3<br>Bit 5: VMU-OC pos.1, Cnt_tot_In4<br>Bit 6: VMU-OC pos.2, Cnt_tot_In1<br>Bit 7: VMU-OC pos.2, Cnt_tot_In2<br>Bit 8: VMU-OC pos.2, Cnt_tot_In3<br>Bit 9: VMU-OC pos.2, Cnt_tot_In4<br>Bit 10: VMU-OC pos.3, Cnt_tot_In1<br>Bit 11: VMU-OC pos.3, Cnt_tot_In2<br>Bit 12: VMU-OC pos.3, Cnt_tot_In3<br>Bit 13: VMU-OC pos.3, Cnt_tot_In4<br>Bit 14 – 15: Not used<br><br>Bit = 1, enable reset/overwrite<br>Bit = 0, reset/overwrite disabled<br>Bits automatically reset after timeout expires (3 s) |

| | | | | | |
|---|---|---|---|---|---|
| **3**16391 | 4101h | 1 | Reset/Overwrite totalizers enable – Tariff T1 | UINT 16 | Bit 0: VMU-MC, Cnt_T1_In1<br>Bit 1: VMU-MC, Cnt_T1_In2<br>Bit 2: VMU-OC pos.1, Cnt_T1_In1<br>Bit 3: VMU-OC pos.1, Cnt_T1_In2<br>Bit 4: VMU-OC pos.1, Cnt_T1_In3<br>Bit 5: VMU-OC pos.1, Cnt_T1_In4<br>Bit 6: VMU-OC pos.2, Cnt_T1_In1<br>Bit 7: VMU-OC pos.2, Cnt_T1_In2<br>Bit 8: VMU-OC pos.2, Cnt_T1_In3<br>Bit 9: VMU-OC pos.2, Cnt_T1_In4<br>Bit 10: VMU-OC pos.3, Cnt_T1_In1<br>Bit 11: VMU-OC pos.3, Cnt_T1_In2<br>Bit 12: VMU-OC pos.3, Cnt_T1_In3<br>Bit 13: VMU-OC pos.3, Cnt_T1_In4<br>Bit 14 – 15: Not used<br><br>Bit = 1, enable reset/overwrite<br>Bit = 0, reset/overwrite disabled<br>Bits automatically reset after timeout expires (3 s) |
| **3**16392 | 4102h | 1 | Reset/Overwrite totalizers enable – Tariff T2 | UINT 16 | Bit 0: VMU-MC, Cnt_T2_In1<br>Bit 1: VMU-MC, Cnt_T2_In2<br>Bit 2: VMU-OC pos.1, Cnt_T2_In1<br>Bit 3: VMU-OC pos.1, Cnt_T2_In2<br>Bit 4: VMU-OC pos.1, Cnt_T2_In3<br>Bit 5: VMU-OC pos.1, Cnt_T2_In4<br>Bit 6: VMU-OC pos.2, Cnt_T2_In1<br>Bit 7: VMU-OC pos.2, Cnt_T2_In2<br>Bit 8: VMU-OC pos.2, Cnt_T2_In3<br>Bit 9: VMU-OC pos.2, Cnt_T2_In4<br>Bit 10: VMU-OC pos.3, Cnt_T2_In1<br>Bit 11: VMU-OC pos.3, Cnt_T2_In2<br>Bit 12: VMU-OC pos.3, Cnt_T2_In3<br>Bit 13: VMU-OC pos.3, Cnt_T2_In4<br>Bit 14 – 15: Not used<br><br>Bit = 1, enable reset/overwrite<br>Bit = 0, reset/overwrite disabled<br>Bits automatically reset after timeout expires (3 s) |
| **3**16393 | 4103h | 1 | Reset/Overwrite totalizers enable – Tariff T3 | UINT 16 | Bit 0: VMU-MC, Cnt_T3_In1<br>Bit 1: VMU-MC, Cnt_T3_In2<br>Bit 2: VMU-OC pos.1, Cnt_T3_In1<br>Bit 3: VMU-OC pos.1, Cnt_T3_In2<br>Bit 4: VMU-OC pos.1, Cnt_T3_In3<br>Bit 5: VMU-OC pos.1, Cnt_T3_In4<br>Bit 6: VMU-OC pos.2, Cnt_T3_In1<br>Bit 7: VMU-OC pos.2, Cnt_T3_In2<br>Bit 8: VMU-OC pos.2, Cnt_T3_In3<br>Bit 9: VMU-OC pos.2, Cnt_T3_In4<br>Bit 10: VMU-OC pos.3, Cnt_T3_In1<br>Bit 11: VMU-OC pos.3, Cnt_T3_In2<br>Bit 12: VMU-OC pos.3, Cnt_T3_In3<br>Bit 13: VMU-OC pos.3, Cnt_T3_In4<br>Bit 14 – 15: Not used<br><br>Bit = 1, enable reset/overwrite<br>Bit = 0, reset/overwrite disabled<br>Bits automatically reset after timeout expires (3 s) |

| | | | | | |
|---|---|---|---|---|---|
| **3**16394 | 4104h | 1 | Reset/Overwrite totalizers enable – Tariff T4 | UINT 16 | Bit 0: VMU-MC, Cnt_T4_In1<br>Bit 1: VMU-MC, Cnt_T4_In2<br>Bit 2: VMU-OC pos.1, Cnt_T4_In1<br>Bit 3: VMU-OC pos.1, Cnt_T4_In2<br>Bit 4: VMU-OC pos.1, Cnt_T4_In3<br>Bit 5: VMU-OC pos.1, Cnt_T4_In4<br>Bit 6: VMU-OC pos.2, Cnt_T4_In1<br>Bit 7: VMU-OC pos.2, Cnt_T4_In2<br>Bit 8: VMU-OC pos.2, Cnt_T4_In3<br>Bit 9: VMU-OC pos.2, Cnt_T4_In4<br>Bit 10: VMU-OC pos.3, Cnt_T4_In1<br>Bit 11: VMU-OC pos.3, Cnt_T4_In2<br>Bit 12: VMU-OC pos.3, Cnt_T4_In3<br>Bit 13: VMU-OC pos.3, Cnt_T4_In4<br>Bit 14 – 15: Not used<br><br>Bit = 1, enable reset/overwrite<br>Bit = 0, reset/overwrite disabled<br>Bits automatically reset after timeout expires (3 s) |

## 2.11 "Serial configuration update" command

**MODBUS**: read and write mode

Table 2.11-1

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| **3**17665 | 4500h | 1 | Update serial configuration | UINT 16 | Bit 0: write 1 to update serial configuration<br>Bit 1 – 15: Not used<br><br>Bits automatically reset after command execution |

## 2.12 Serial number and production year

**MODBUS**: read only mode

Table 2.12-1

| Modicon address | Physical address | Length (words) | VARIABLE ENG. UNIT | Data Format | Notes |
|---|---|---|---|---|---|
| **3**20481 | 5000h | 1 | Letter 1 (from SX)<br>Letter 2 (from SX) | UINT 16 | MSB: ASCII code<br>LSB: ASCII code |
| **3**20482 | 5001h | 1 | Letter 3 (from SX)<br>Letter 4 (from SX) | UINT 16 | MSB: ASCII code<br>LSB: ASCII code |
| **3**20483 | 5002h | 1 | Letter 5 (from SX)<br>Letter 6 (from SX) | UINT 16 | MSB: ASCII code<br>LSB: ASCII code |
| **3**20484 | 5003h | 1 | Letter 7 (from SX)<br>Letter 8 (from SX) | UINT 16 | MSB: ASCII code<br>LSB: ASCII code |
| **3**20485 | 5004h | 1 | Letter 9 (from SX)<br>Letter 10 (from SX) | UINT 16 | MSB: ASCII code<br>LSB: ASCII code |
| **3**20486 | 5005h | 1 | Letter 11 (from SX)<br>Letter 12 (from SX) | UINT 16 | MSB: ASCII code<br>LSB: ASCII code |
| **3**20487 | 5006h | 1 | Letter 13 (from SX) | UINT 16 | MSB: ASCII code<br>LSB: Not Used |
| **3**20488 | 5007h | 1 | Instrument production year | UINT 16 | Value = year (e.g. 2016) |

# 3 Revisions

Ver 1, rev 1 (2017-10-30): Corrected totalizers table because there was a gap of addresses between VMU-MC: Cnt_T4_In1 and VMU-MC: Cnt_T1_In2
Ver 1, rev 2 (2018-02-15): Corrected some labels.