# WM4-96

(Rev. C04 and following)

# SERIAL COMMUNICATION PROTOCOL

Vers. 1 Rev. 3

February 15th, 2005

# Index

# 1 SERIAL COMMUNICATION PROTOCOL

## 1.1 INTRODUCTION

WM4-96 can be equipped with a RS485 or RS232 serial interface. The serial communication protocol, MODBUS-RTU, is the same on both interfaces. When using RS485, it is possible to connect up to 255 instruments using MODBUS protocol. When using RS232 it is only possible to connect a single instrument (multidrop feature is not available).

The command's structure of the protocol allows the user to read and write from/in the μP RAM memory and the peripheral devices (EEPROM, real time clock, flash memory), so that all the functions are completely transparent.
The communication parameters are configurable when using both the RS485 interface and the RS232 one, in accordance with the following table:

| Interface | Baud rate (bps) | Parity | Stop bit |
|---|---|---|---|
| RS232 | 2400 to 38400 | None | 1 |
| RS485 | 1200 to 9600 | None, even, odd | 1 |

**NOTE:** please refer to the instruction manual for any detail on the instrument programming.

The communication can be started only by the HOST unit, which sends the request frame. Each frame contains the following information:

- slave address: is a number from 1 to 255, which identifies the instrument connected to the network. Address 0 (zero) is accepted (in write frames only) by all the instruments, which will execute the relevant command but won't send any answer frame.
  **NOTE:** The request frame must always contain the address even if, when using RS232, it is not considered (every legal value is accepted).
- command: it defines the command type (e.g. read function, write function etc.).
- data fields: these numbers define the operating parameters of the command (e.g. the address of the word, the value of the word to be written, etc.).
- CRC word: it allows detecting transmission errors that may occur. CRC calculation is carried out by the MASTER unit once it has defined address, command and data fields. When the frame is received by the SLAVE, it is stored in a temporary buffer. The CRC is calculated and then compared with the received one. If they correspond and the address is recognised by the SLAVE unit, the command is executed and an answer frame is sent.

If the CRC is not correct, the frame is discarded and no answer is sent.

## 1.2 FUNCTIONS

WM4-96 accepts the following three commands:

- Read words (code 04h)
- Write one word (code 06h)
- Read words from Flash Memory (code 80h)

**NOTE:** the memory addressing is different according to the used function. It is explained in detail in the paragraph 1.3.
Function 04h and 06h are compatible with the Modbus coding, while function 80h is very similar but differs slightly from Modbus coding.

### 1.2.1 Function 04 (read words)

*Request frame*

| Address | Function | Data address | | n° of words | | CRC | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 byte | 1 byte | 2 byte | | 2 byte | | 2 byte | |
| From 1 to 255 | 04h | MSB | LSB | MSB | LSB | MSB | LSB |

**NOTE:** - The maximum number of word is 120 (240 byte).
- The address 00 is not allowed (it generates no answer)

*Answer frame*

| Address | Function | n° byte (=2 x n° word) | | Values | CRC | |
| --- | --- | --- | --- | --- | --- | --- |
| 1 byte | 1 byte | 1 byte | | n° byte (=2 x n° word) | 2 byte | |
| From 1 to 255 | 04h | MSB | LSB | … | MSB | LSB |

### 1.2.2 Function 06 (write one word)

*Request frame*

| Address | Function | Data address | | Value | | CRC | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 byte | 1 byte | 2 byte | | 2 byte | | 2 byte | |
| From 1 to 255 | 06h | MSB | LSB | MSB | LSB | MSB | LSB |

*Answer frame*

| Address | Function | Data address | | Value | | CRC | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 byte | 1 byte | 2 byte | | 2 byte | | 2 byte | |
| From 1 to 255 | 06h | MSB | LSB | MSB | LSB | MSB | LSB |

**NOTE**: the answer frame is an echo of the request frame, which confirm the execution of the command.
The write function cannot be used to modify the contents of the energy meter memory area.

### 1.2.3 Function 80 (read words from Memory Flash)

*Request frame*

| Address | Function | Word address and number of words | | | | CRC | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 byte | 1 byte | 4 byte (see Flash Memory addressing table) | | | | 2 byte | |
| From 1 to 255 | 80h | MSB | … | … | LSB | MSB | LSB |

*Answer frame*

| Address | Function | Number of byte | Value | CRC | |
| --- | --- | --- | --- | --- | --- |
| 1 byte | 1 byte | 1 byte | n byte | 2 byte | |
| From 1 to 255 | 80h | Number of words x 2 | … | MSB | LSB |

**IMPORTANT**: if the address is 00 (zero) all the instruments connected to the network will execute the command but won't send an answer frame.

### 1.3 MEMORY AREA

WM4-96 manages four different memory areas addressed as follows.

Function 04h and 06h:

| Memory area | Area | | Byte reading order |
| --- | --- | --- | --- |
| Internal RAM | 0000h | 00E7h | MSB, LSB |
| Internal RAM | 00E8h | 1FFFh | LSB, MSB |
| Data storage EEPROM | 2000h | 3FFFh | MSB, LSB |
| Real Time Clock | 4000h | 5FFFh | LSB |

Function 80h:

| Memory area | Area | Byte reading order |
| --- | --- | --- |
| Flash Memory | From 0000h | MSB, LSB |

The Flash Memory is composed by 4095 pages, shared in three different blocks containing different kind of information:

- from page 0000 to page 3993: logged data
- from page 3394 to page 3395: telephone numbers and SMS messages
- from page 3396 to page 4095: load profile data

The Flash Memory addressing requires to indicate the page number (from 0 to 4095), followed by the word address in that specific page (from 0 to 527) and from the number of words to be read. Therefore the addressing requires 4 bytes to be written as a single 32-bit word: the 14 more significant bits indicate the page, the following 10 bits define the address inside the page, the 8 less significant bits indicate the number of word to be read (maximum 132 word).

*Flash Memory addressing table*

| Memory area | Word address and number of words (4 byte) | | Byte reading order |
| --- | --- | --- | --- |
| Flash Memory | 00pppppppppppppp | iiiiiiii wwwwwwww | MSB, LSB |

where
pppppppppppppp = page number (14 bit)
iiiiiiii = word address inside the pppppppppppppp page (8 bit)
wwwwwwww= number of words to be read

It is only possible to reset the logged data from the Flash Memory using fixed frames (see paragraph 5.4)

**NOTE:** in the following pages the following notation will be used:
1 int = 4 byte;
1 short = 2 byte;
1 word = 2 byte;
1 byte = 8 bit.

### 1.4 WM4-96 IDENTIFICATION CODE AND SERIAL NUMBER

Every Carlo Gavazzi instrument is identified by means of a code stored in address 0Bh, in order to recognise the type of the instrument via serial communication. The WM4-96 code is 0010h.
This code can be read with the following fixed frame:

*Instrument code request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 00h | 0Bh | 00h | 01h | CRC | CRC |

*Instrument code answer frame (7 byte):*

| | | | | | | |
|---|---|---|---|---|---|---|
| 01h | 04h | 02h | **00h** | **10h** | CRC | CRC |

The serial number of the instrument is stored as short (2 bytes) on location 0218h.

## 2 RAM VARIABLES MAP

### 2.1 INSTANTANEOUS VARIABLES MAP

| Word | ADDRESS | BYTE | VARIABLE | Type | Word | ADDRESS | BYTE | VARIABLE | Type |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 000 | 4 | V L1-N | V | 31 | 078 | 4 | THD V2 | D |
| 2 | 004 | 4 | A L1 | A | 32 | 07C | 4 | THDe V2 | D |
| 3 | 008 | 4 | W L1 | P | 33 | 080 | 4 | THDo V2 | D |
| 4 | 00C | 4 | V L2-N | V | 34 | 084 | 4 | THD V3 | D |
| 5 | 010 | 4 | A L2 | A | 35 | 088 | 4 | THDe V3 | D |
| 6 | 014 | 4 | W L2 | P | 36 | 08C | 4 | THDo V3 | D |
| 7 | 018 | 4 | V L3-N | V | 37 | 090 | 4 | THD A1 | D |
| 8 | 01C | 4 | A L3 | A | 38 | 094 | 4 | THDe A1 | D |
| 9 | 020 | 4 | W L3 | P | 39 | 098 | 4 | THDo A1 | D |
| 10 | 024 | 4 | V L1 | V | 40 | 09C | 4 | THD A2 | D |
| 11 | 028 | 4 | V L2 | V | 41 | 0A0 | 4 | THDe A2 | D |
| 12 | 02C | 4 | V L3 | V | 42 | 0A4 | 4 | THDo A2 | D |
| 13 | 030 | 4 | VA L1 | P | 43 | 0A8 | 4 | THD A3 | D |
| 14 | 034 | 4 | var L1 | P | 44 | 0AC | 4 | THDe A3 | D |
| 15 | 038 | 4 | PF L1 | C | 45 | 0B0 | 4 | THDo A3 | D |
| 16 | 03C | 4 | VA L2 | P | 46 | 0B4 | 4 | A dmd | A |
| 17 | 040 | 4 | var L2 | P | 47 | 0B8 | 4 | VA dmd | P |
| 18 | 044 | 4 | PF L2 | C | 48 | 0BC | 4 | PF avg | C |
| 19 | 048 | 4 | VA L3 | P | 49 | 0C0 | 4 | W dmd | P |
| 20 | 04C | 4 | var L3 | P | 50 | 0C4 | 4 | Hz | H |
| 21 | 050 | 4 | PF L3 | C | 51 | 0C8 | 4 | ASY | D |
| 22 | 054 | 4 | V ∑ | V | 52 | 0CC | 4 | VL-N ∑ | V |
| 23 | 058 | 4 | A ∑ | A | 53 | 0D0 | 4 | var dmd | P |
| 24 | 05C | 4 | W ∑ | P | 54 | | 4 | | |
| 25 | 060 | 4 | VA ∑ | P | 55 | | 4 | | |
| 26 | 064 | 4 | Var ∑ | P | 56 | | 4 | | |
| 27 | 068 | 4 | PF ∑ | C | 57 | | 4 | | |
| 28 | 06C | 4 | THD V1 | D | 58 | | 4 | | |
| 29 | 070 | 4 | THDe V1 | D | 59 | 0E8 | 1+1+1+1 | Unit V,A/P | Inf1/2 |
| 30 | 074 | 4 | THDo V1 | D | | | | | |

**NOTE:** all the variables in this table are contiguous. It is possible to read the whole area with a single command sending, in the request frame, 000h as data address and 0076h as number of words (that is 118 in decimal).
The values of the instantaneous variables are stored in the addresses from 000h to 0E7h. The data are sent in 4-byte groups in the following order: MSB, ..., ..., LSB.

## 2.2 VARIABLE FORMAT

The value of all the instantaneous variables is stored as a 4 byte (2 word) integer value. The decimal point and the multiplier have to be set according to the **inf1/2** word coding (see the following table) for voltage (V), current (A) and power (P), in the position "**111.1**" for the THD-type (%) and H-type (Hz) type variables and in position "**1.111**" for the G-type variables (PF). The single phase PF variables are stored with a positive value if the power factor is "L" (inductive), and with a negative value if the power factor is "C" (capacitive). The variable "PF ∑" has neither L nor C sign indication.

**Variable format info map**

| Address | Byte | Variable | Type |
|---|---|---|---|
| 0E8 | 1 | Info voltage value | inf1 |
| 0E9 | 1 | Info current value | inf1 |
| 0EA | 1 | Info power value | inf2 |
| 0EB | 1 | --- | |

**Decimal point and multiplier coding**

| INF value | d.p | INF value | d.p |
|---|---|---|---|
| 0 | 1.111m | 8 | 111.1k |
| 1 | 11.11m | 9 | 1111k |
| 2 | 111.1m | 10 | 11.11M |
| 3 | 1.111 | 11 | 111.1M |
| 4 | 11.11 | 12 | 1111M |
| 5 | 111.1 | 13 | 11.11G |
| 6 | 1111 | 14 | 111.1G |
| 7 | 11.11K | 15 | |

**NOTE**: if a power value exceeds 9999, the autoranging function will intervene and modify the inf2 value. If the power value is lower than 99999 the inf2 will be increased of 1 unit, if the power value is greater than 99999 but lower than 999999 the inf2 will be increased of 2 units and so on.

***Example 1:* reading of an int variable stored at address 100h**

An int variable is 4 byte (2 word) long, so a 2-word read request must be sent:

*Read command request frame*

| Address | Function | Word address | | n° of words | | CRC | |
|---|---|---|---|---|---|---|---|
| 1 byte | 1 byte | 2 byte | | 2 byte | | 2 byte | |
| from 1 to 255 | 04 | 01h | 00h | 00h | 02h | MSB | LSB |

*Read command answer frame*

| Address | Function | n° byte | Value of int type variable | | | | CRC | |
|---|---|---|---|---|---|---|---|---|
| 1 byte | 1 byte | 1 byte | 1° byte | 2° byte | 3° byte | 4° byte | 2 byte | |
| from 1 to 255 | 04 | 04 | LSB | | | MSB | MSB | LSB |

***Example 2*: reading of 4 char variables (4 bytes=2 words) starting from address 1C0h**

Char type variables (1 byte) must always be read carrying out a 1-word (2-bytes) read request and taking only the needed byte into account. Note that the first sent byte is the byte relevant to the specified word address. The following bytes are relevant to the previous address+1.

*Read command request frame*

| Address | Function | Word address | | n° of words | | CRC | |
|---|---|---|---|---|---|---|---|
| 1 byte | 1 byte | 2 byte | | 2 byte | | 2 byte | |
| From 1 to 255 | 04 | 01h | C0h | 00h | 02h | MSB | LSB |

*Read command answer frame*

| Address | Functi on | n° byte | Value | Value | Value | Value | CRC | |
|---|---|---|---|---|---|---|---|---|
| 1 byte | 1 byte | 1 byte | 1° byte | 2° byte | 3° byte | 4° byte | 2 byte | |
| From 1 to 255 | 04 | 04 | **01C0h** | **01C1h** | **01C2h** | **01C3h** | MSB | LSB |

## 2.3 INSTANTANEOUS VARIABLES READING

***Example 3*: Reading of a single variable: `W1`**

*Value request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 00h | 08h | 00h | 02h | CRC | CRC |

*Value answer frame (9 byte):*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 04h | 00h | 00h | 63h | 8Dh | CRC | CRC |

*Info request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 00h | E8h | 00h | 02h | CRC | CRC |

*Info answer frame (frame 9 byte):*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 04h | 07h | 07h | 06h | 00h | CRC | CRC |

Stored value: 638Dh (25485 decimal)
Info value (P type): 06h

Since 9999<25485<99999, the inf2 value to be considered is 06+1=07 (11.11K)
Variable value (W1): 25.48 kW

***Example 4*: Reading of all the instantaneous variables:**

*All instantaneous values (+ info) request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 00h | 00h | 00h | 76h | CRC | CRC |

*All instantaneous values (+ info) answer frame (241 byte):*

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | 04h | ECh | 00h | 00h | 01h | 37h | - - - | 07h | 07h | 0Ah | 03h | CRC | CRC |

`VL1-N` stored value: 0137h (0311 decimal)
.....................
.....................
Info (V type) value: 07h
Info (A type) value: 07h
Info (P type) value: 0Ah

Variable value (`VL1-N`): 3.11 kV

## 2.4 ENERGY METERS MAP

Table 1

| ADDRESS | BYTE | SEASON | PERIOD | METER TYPE |
|---|---|---|---|---|
| 0EC | 4 | TOTALE | | KWh+ (LSB) |
| 0F0 | 4 | | | KWh- (LSB) |
| 0F4 | 4 | | | Kvarh+ (LSB) |
| 0F8 | 4 | | | Kvarh- (LSB) |
| 0FC | 1 | | | KWh+ (MSB) |
| 0FD | 1 | | | KWh- (MSB) |
| 0FE | 1 | | | Kvarh+ (MSB) |
| 0FF | 1 | | | Kvarh- (MSB) |
| 100 | 4 | WINTER | 1 | KWh+ (LSB) |
| 104 | 4 | | | KWh- (LSB) |
| 108 | 4 | | | Kvarh+ (LSB) |
| 10C | 4 | | | Kvarh- (LSB) |
| 110 | 4 | | 2 | KWh+ (LSB) |
| 114 | 4 | | | KWh- (LSB) |
| 118 | 4 | | | Kvarh+ (LSB) |
| 11C | 4 | | | Kvarh- (LSB) |
| 120 | 4 | | 3 | KWh+ (LSB) |
| 124 | 4 | | | KWh- (LSB) |
| 128 | 4 | | | Kvarh+ (LSB) |
| 12C | 4 | | | Kvarh- (LSB) |
| 130 | 4 | | 4 | KWh+ (LSB) |
| 134 | 4 | | | KWh- (LSB) |
| 138 | 4 | | | Kvarh+ (LSB) |
| 13C | 4 | | | Kvarh- (LSB) |
| 140 | 4 | SUMMER | 1 | KWh+ (LSB) |
| 144 | 4 | | | KWh- (LSB) |
| 148 | 4 | | | Kvarh+ (LSB) |
| 14C | 4 | | | Kvarh- (LSB) |
| 150 | 4 | | 2 | KWh+ (LSB) |
| 154 | 4 | | | KWh- (LSB) |
| 158 | 4 | | | Kvarh+ (LSB) |
| 15C | 4 | | | Kvarh- (LSB) |
| 160 | 4 | | 3 | KWh+ (LSB) |
| 164 | 4 | | | KWh- (LSB) |
| 168 | 4 | | | Kvarh+ (LSB) |
| 16C | 4 | | | Kvarh- (LSB) |
| 170 | 4 | | 4 | KWh+ (LSB) |
| 174 | 4 | | | KWh- (LSB) |
| 178 | 4 | | | Kvarh+ (LSB) |
| 17C | 4 | | | Kvarh- (LSB) |
| 180 | 4 | HOLYDAY | 1 | KWh+ (LSB) |
| 184 | 4 | | | KWh- (LSB) |
| 188 | 4 | | | Kvarh+ (LSB) |
| 18C | 4 | | | Kvarh- (LSB) |
| 190 | 4 | | 2 | KWh+ (LSB) |
| 194 | 4 | | | KWh- (LSB) |
| 198 | 4 | | | Kvarh+ (LSB) |
| 19C | 4 | | | Kvarh- (LSB) |
| 1A0 | 4 | | 3 | KWh+ (LSB) |
| 1A4 | 4 | | | KWh- (LSB) |
| 1A8 | 4 | | | Kvarh+ (LSB) |
| 1AC | 4 | | | Kvarh- (LSB) |
| 1B0 | 4 | | 4 | KWh+ (LSB) |
| 1B4 | 4 | | | KWh- (LSB) |
| 1B8 | 4 | | | Kvarh+ (LSB) |
| 1BC | 4 | | | Kvarh- (LSB) |

Table 2

| ADDRESS | BYTE | SEASON | PERIOD | METER TYPE |
|---|---|---|---|---|
| 8E8 | 1 | WINTER | 1 | KWh+ (MSB) |
| 8E9 | 1 | | | KWh- (MSB) |
| 8EA | 1 | | | Kvarh+ (MSB) |
| 8EB | 1 | | | Kvarh- (MSB) |
| 8EC | 1 | | 2 | KWh+ (MSB) |
| 8ED | 1 | | | KWh- (MSB) |
| 8EE | 1 | | | Kvarh+ (MSB) |
| 8EF | 1 | | | Kvarh- (MSB) |
| 8F0 | 1 | | 3 | KWh+ (MSB) |
| 8F1 | 1 | | | KWh- (MSB) |
| 8F2 | 1 | | | Kvarh+ (MSB) |
| 8F3 | 1 | | | Kvarh- (MSB) |
| 8F4 | 1 | | 4 | KWh+ (MSB) |
| 8F5 | 1 | | | KWh- (MSB) |
| 8F6 | 1 | | | Kvarh+ (MSB) |
| 8F7 | 1 | | | Kvarh- (MSB) |
| 8F8 | 1 | SUMMER | 1 | KWh+ (MSB) |
| 8F9 | 1 | | | KWh- (MSB) |
| 8FA | 1 | | | Kvarh+ (MSB) |
| 8FB | 1 | | | Kvarh- (MSB) |
| 8FC | 1 | | 2 | KWh+ (MSB) |
| 8FD | 1 | | | KWh- (MSB) |
| 8FE | 1 | | | Kvarh+ (MSB) |
| 8FF | 1 | | | Kvarh- (MSB) |
| 900 | 1 | | 3 | KWh+ (MSB) |
| 901 | 1 | | | KWh- (MSB) |
| 902 | 1 | | | Kvarh+ (MSB) |
| 903 | 1 | | | Kvarh- (MSB) |
| 904 | 1 | | 4 | KWh+ (MSB) |
| 905 | 1 | | | KWh- (MSB) |
| 906 | 1 | | | Kvarh+ (MSB) |
| 907 | 1 | | | Kvarh- (MSB) |
| 908 | 1 | HOLYDAY | 1 | KWh+ (MSB) |
| 909 | 1 | | | KWh- (MSB) |
| 90A | 1 | | | Kvarh+ (MSB) |
| 90B | 1 | | | Kvarh- (MSB) |
| 90C | 1 | | 2 | KWh+ (MSB) |
| 90D | 1 | | | KWh- (MSB) |
| 90E | 1 | | | Kvarh+ (MSB) |
| 90F | 1 | | | Kvarh- (MSB) |
| 910 | 1 | | 3 | KWh+ (MSB) |
| 911 | 1 | | | KWh- (MSB) |
| 912 | 1 | | | Kvarh+ (MSB) |
| 913 | 1 | | | Kvarh- (MSB) |
| 914 | 1 | | 4 | KWh+ (MSB) |
| 915 | 1 | | | KWh- (MSB) |
| 916 | 1 | | | Kvarh+ (MSB) |
| 917 | 1 | | | Kvarh- (MSB) |

A further table, relevant to the monthly energy meters, will be explained afterwards.

**NOTE**: Table 1 and Table 2 are not contiguous. The variables included in each table are contiguous, so that it is possible to read every variables with two request frames. With the first request frame the 106 words included in Table 1 could be read, with the second request frame the 24 words included in Table 2 could be read.
The values of all the total and partial energy meters are stored as a 5-byte integer (the first 4 bytes are the less significant part, the 5$^{th}$ is the most significant one). The resolution of the meters is 10W

(the decimal point position has to be set to "**1.11Kwh (Kvarh)**").
Whereas the total meters MSB (5th byte) is contiguous to the less significant bytes, the partial meters MSB (5th byte) is stored in a different area of the memory. For this reason it is required to carry out two different read commands in order to get all the energy meter information.

## 2.5 READING OF THE ENERGY METER VALUES

*8-bytes request frame (read command, 10 word):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 00h | ECh | 00h | 0Ah | CRC | CRC |

*25-bytes answer frame (read command):*

| | | | 1 ECh | 2 EDh | 3 EEh | 4 EFh | 5 F0h | 6 F1h | 7 F2h | 8 F3h | 9 F4h | 10 F5h |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 14h | **00h** | **00h** | **00h** | **00h** | **94h** | **59h** | **FFh** | **FFh** | **94h** | **02h** |

| 11 F6h | 12 F7h | 13 F8h | 14 F9h | 15 FAh | 16 FBh | 17 FCh | 18 FDh | 19 Feh | 20 FFh | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **00h** | **00h** | **BEh** | **FEh** | **FFh** | **FFh** | **00h** | **00h** | **00h** | **00h** | CRC | CRC |

Starting from address ECh, it is possible to read all the energy meters by means of a single read command (10 word, see the example above).

*Reconstruction of the kWh+ total meter*
The first 4 data bytes (less significant bytes) have to be placed side by side in the opposite order:

| 4 Efh | 3 EEh | 2 EDh | 1 ECh |
|---|---|---|---|
| **00h** | **00h** | **00h** | **00h** |

00000000h=0

The obtained 32-bit value has to be interpreted as a two's complement value.
The relevant kWh+ MSB (byte n° 17), which has to be interpreted as a two's complement value too, must be multiplied by 1000000000 (decimal value). The result has to be algebraically added to the previous value.

| 17 FCh |
|---|
| **00h** |

1000000000*0=0
Finally the last result has to be divided by 100.
0+0/100=0 kWh

***Example 5*: reconstruction of the kWh- total meter**

| 5 F0h | 6 F1h | 7 F2h | 8 F3h |
|---|---|---|---|
| **94h** | **59h** | **FFh** | **FFh** |

FF FF 59 94h = -42604

| 18 FDh |
|---|
| **00h** |

1000000000*0 = 0

(- 42604 + 0*1000000000)/100 = - 426.04 kWh

## 2.6 WRITING OF THE ENERGY METER VALUES

The user is not allowed to write in the energy meter memory area. It is only possible to reset the energy meter using fixed frames.

## 2.7 ENERGY METERS RESET COMMANDS

The fixed frames to be used to reset the energy meters are listed below:

1. **General reset command (reset of all the total, partial and monthly meters)**

*Reset request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 00h | ECh | D4h | F0h | CRC | CRC |

*Reset answer frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 00h | ECh | D4h | F0h | CRC | CRC |

2. **Total positive energy meters (kWh+ and kvarh+) and monthly meters reset command**

*Reset request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 01h | 00h | A5h | F0h | CRC | CRC |

*Reset answer frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 01h | 00h | A5h | F0h | CRC | CRC |

3. **Total negative energy meters (kWh- and kvarh-) and monthly meters reset command**

*Reset request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 01h | 04h | 23h | 44h | CRC | CRC |

*Reset answer frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 01h | 04h | 23h | 44h | CRC | CRC |

4. **Partial positive energy meters (kWh+ and kvarh+) and monthly meters reset command**

*Reset request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 01h | 08h | 87h | 35h | CRC | CRC |

*Reset answer frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 01h | 08h | 87h | 35h | CRC | CRC |

5. **Partial negative energy meters (kWh- and kvarh-) and monthly meters reset command**

*Reset request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 01h | C0h | 59h | 12h | CRC | CRC |

*Reset answer frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 01h | C0h | 59h | 12h | CRC | CRC |

## 2.8 GAS AND WATER METERS MAP

Table 3

| ADDRESS | BYTE | METER TYPE |
|---|---|---|
| 9C8 | 4 | GAS: TOTAL |
| 9CC | 4 | GAS: DAY |
| 9D0 | 4 | GAS: NIGHT |
| 9D4 | 4 | WATER: TOTAL |

The utility meters contain integer values. It is possible to enable/disable the utility metering modifying the contents of the address 2036h (see EEPROM map).
Their resolution is 0.1 $m^3$ and, after reaching 99999999.9 $m^3$ the total meters will be reset and start again from 0. The day/night gas meters full maximum value is 50000000.0 $m^3$.

To reset an utility meter two write requests (write 0000h) are to be sent to the instrument: the first to reset the two most significant bytes, the second to reset the two less significant bytes.

***Example 6*: Reset of the day gas meter:**

*1st Reset request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 09h | CEh | 00h | 00h | CRC | CRC |

*1st Reset answer frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 09h | CEh | 00h | 00h | CRC | CRC |

*2nd Reset request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 09h | CCh | 00h | 00h | CRC | CRC |

*2nd Reset answer frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 09h | CCh | 00h | 00h | CRC | CRC |

## 2.9 ALARM STATUS MAP

Table 4

| ADDRESS | BYTE | Variable type |
|---|---|---|
| 1C0 | 1+1 | Diagn0 , diagn1 |
| 1C2 | 1+1 | Diagn2 , diagn3 |
| 1C4 | 1+1 | Alarm0 , alarm1 |
| 1C6 | 1+1 | Alarm2 , alarm3 |
| 1C8 | 2 | control 0 type |
| 1CA | 2 | control 1 type |
| 1CC | 2 | control 2 type |
| 1CE | 2 | control 3 type |
| 1D0 | 2 | status Relay 0 |
| 1D2 | 2 | status Relay 1 |
| 1D4 | 2 | status Relay 2 |
| 1D6 | 2 | status Relay 3 |
| 1D8 | 2 | Variable associated to alarm 0 |
| 1DA | 2 | Variable associated to alarm 1 |
| 1DC | 2 | Variable associated to alarm 2 |
| 1DE | 2 | Variable associated to alarm 3 |
| 1E0 | 2 | ON set-point 0 |
| 1E2 | 2 | ON set-point 1 |
| 1E4 | 2 | ON set-point 2 |
| 1E6 | 2 | ON set-point 3 |
| 1E8 | 2 | OFF set-point 0 |
| 1EA | 2 | OFF set-point 1 |
| 1EC | 2 | OFF set-point 2 |
| 1EE | 2 | OFF set-point 3 |
| 1F0 | 2 | delay 0 |
| 1F2 | 2 | delay 1 |
| 1F4 | 2 | delay 2 |
| 1F6 | 2 | delay 3 |

Table 5

| ADDRESS | BYTE | Variable type |
|---|---|---|
| 8D8 | 1+1 | Remote 1, Remote 2 |
| 8DA | 1+1 | Remote 3, Remote 4 |

**NOTE**: the variables included in each of the previous tables are contiguous: it is possible to read every variables with two request frames. With the first request frame the 28 words included in Table 4 can be read, with the second request frame the 2 words included in Table 5 can be read.
In order to know the current digital output settings, see the EEPROM map paragraph.

## 2.10 READING OF ALARM, DIAGNOSTIC AND REMOTE CONTROL OUTPUT STATUS

The $n^{th}$ digital output can work as pulse output, alarm output, diagnostic output or remote control output.
In order to know if the $n^{th}$ digital output is set as alarm, the $n^{th}$ alarm byte (“alarm n”) must be read. If the byte is equal to 0 it means that the digital output is not set as alarm, if it is equal to 1 the alarm status is OFF, if it is equal to 2 the alarm status is ON.
The same considerations are valid in case of diagnostic output (“diagn n” byte must be read) or remote control output (“Remote n” byte must be read).
Of course, only one among “alarm n”, “diagn n” and “remote n” byte can be different from 0. If all these three bytes are equal to 0, it means that the $n^{th}$ digital output is set as pulse output.

If the digital outputs are set as alarm, the values stored in addresses from 1C8h to 1CEh indicate the control type, coded as follows:
0 = UP
1 = UP-LATCH
2 = DOWN
3 = DOWN LATCH

The values stored in addresses from 1D0h to 1D6h explain if the relay is normally energised or de-energised:
0 = Normally de-energised
1 = Normally energised

In the addresses from 1D8h to 1DEh the variables associated to the alarms are stored, according to the "Variable type coding" table (see paragraph 3.1.1).

*Example:* if a control on variable W1 has been associated to alarm1, in the address 1DAh the value 12 must be stored

The set-point ON and OFF values are stored as unsigned short.
The delay values are stored as short and must be included in the range from 0 to 255 seconds.

***Example 7*: "Diagnostic" read command**

*2-word read command request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 01h | C0h | 00h | 02h | CRC | CRC |

*Read command answer frame (9 byte):*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 04h | **00h** | **00h** | **01h** | **00h** | CRC | CRC |

Digital output 0: NO Diagnostic
Digital output 1: NO Diagnostic
Digital output 2: Diagnostic OFF
Digital output 3: NO Diagnostic

***Example 8*: "Alarm" read command**

*2-word read command request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 01h | C4h | 00h | 02h | CRC | CRC |

*Read command answer frame (9 byte):*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 04h | **00h** | **01h** | **00h** | **00h** | CRC | CRC |

Digital output 0: NO Alarm
Digital output 1: Alarm OFF
Digital output 2: NO Alarm
Digital output 3: NO Alarm

**_Example 9_: "Control type" read command**

*4-word read command request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 01h | C8h | 00h | 04h | CRC | CRC |

*Read command answer frame (13 byte):*

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | **LSB** | **MSB** | **LSB** | **MSB** | **LSB** | **MSB** | **LSB** | **MSB** | | |
| 01h | 04h | 08h | **00h** | **00h** | **00h** | **00h** | **00h** | **00h** | **00h** | **00h** | CRC | CRC |

Digital output 0: Not used (digital output 0 is not set as alarm, see previous example)
Digital output 1: UP control
Digital output 2: Not used
Digital output 3: Not used

**_Example 10_: "Relay status" read command**

*4-word read command request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 01h | D0h | 00h | 04h | CRC | CRC |

*Read command answer frame (13 byte):*

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 08h | **00h** | **00h** | **01h** | **00h** | **00h** | **00h** | **00h** | **00h** | CRC | CRC |

Digital output 0: Not used (digital output 0 is not set as alarm, see example 7)
Digital output 1: Normally energised
Digital output 2: Not used
Digital output 3: Not used

**_Example 11_: "Variable associated to the alarm" read command**

*4-word read command request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 01h | D8h | 00h | 04h | CRC | CRC |

*Read command answer frame (13 byte):*

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 08h | **00h** | **00h** | **26h** | **00h** | **00h** | **00h** | **00h** | **00h** | CRC | CRC |

Digital output 0: not used (digital output 0 is not set as alarm, see example 7)
Digital output 1: THD A1
Digital output 2: not used
Digital output 3: not used

**_Example 12_: "ON Set-point" (alarm activation) read command**

*4-word read command request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 01h | E0h | 00h | 04h | CRC | CRC |

*Read command answer frame (13 byte):*

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 08h | **00h** | **00h** | **64h** | **00h** | **00h** | **00h** | **00h** | **00h** | CRC | CRC |

Digital output 0: not used (digital output 0 is not set as alarm, see example 7)
Digital output 1: 10.0% (0064h = 100 decimal)
Digital output 2: not used
Digital output 3: not used

***Example 13*: "OFF Set-point" (alarm deactivation) read command**

*4-word read command request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 01h | E8h | 00h | 04h | CRC | CRC |

*Read command answer frame (13 byte):*

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 08h | **00h** | **00h** | **32h** | **00h** | **00h** | **00h** | **00h** | **00h** | CRC | CRC |

Digital output 0: not used (digital output 0 is not set as alarm, see example 7)
Digital output 1: 5.0% (0032h = 50 decimal)
Digital output 2: not used
Digital output 3: not used

***Example 14*: "Alarm activation delay" read command**

*4-word read command request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 01h | F0h | 00h | 04h | CRC | CRC |

*Read command answer frame (13 byte):*

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 08h | **00h** | **00h** | **04h** | **00h** | **00h** | **00h** | **00h** | **00h** | CRC | CRC |

Digital output 0: not used (digital output 0 is not set as alarm, see example 7)
Digital output 1: 4 seconds
Digital output 2: not used
Digital output 3: not used

***Example 15*: "Latch alarm" reset command**

To reset a UP-LATCH or DOWN-LATCH alarm, the relevant alarm byte must be set to 1.

*Reset command request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | **01h** | **C4h** | **00h** | **01h** | CRC | CRC |

*Reset command answer frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | **01h** | **C4h** | **00h** | **01h** | CRC | CRC |

To reset the alarm 1, the byte at address 01C5h must be set to 1. The byte at address 01C4h must be set to 00h, since it is relevant to alarm 0.

### 2.11 WRITE COMMAND FOR REMOTE CONTROL OUTPUT

The remote control digital output memory area is described in Table 5 and consists in 4 bytes starting from address 08D8h (Remote1=8D8h, Remote2=8D9h, and so on).
To switch ON the $n^{th}$ remote control output, the value 02h must be written in the "Remote n" byte, while to switch OFF the $n^{th}$ remote control output, the value 01h must be written in the "Remote n" byte. Note again that the write command always writes 1 word (2 bytes).

*Request frame: R1 = ON and R2 = OFF (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | **08h** | **D8h** | **02h** | **01h** | CRC | CRC |

*Answer frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | **08h** | **D8h** | **02h** | **01h** | CRC | CRC |

*Request frame: R1 = OFF and R2 = OFF (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | **08h** | **D8h** | **01h** | **01h** | CRC | CRC |

*Answer frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | **08h** | **D8h** | **01h** | **01h** | CRC | CRC |

**NOTE:** a digital output can be used as remote control output only if the relevant "digital output type" variable stored in EEPROM is correctly set (see paragraph 3.1.19).

### 2.12 FORMAT OF THE "PRESENT MODULES+DIGITAL INPUT STATUS" WORD

| ADDRESS | BYTE | Code | Variable type |
|---|---|---|---|
| 800 | 2 | XXXXXXXXXXXXXXXX | Module |

Code

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Gas | F1 | kvarh+ | kWh+ | | | | S3A | S4A | S2A | 232 | CLK | 485 | F2 | Water | |

Serial output

| 485 | RS485 module |
|---|---|
| 0 | Not present |
| 1 | Present |

| 232 | RS232 module |
|---|---|
| 0 | Not present |
| 1 | Present |

| CLK | RTC Clock |
|---|---|
| 0 | Not present |
| 1 | Present |

Digital output code

| S3A | S4A | S2A | Available digital outputs on the inserted modules |
|---|---|---|---|
| 0 | 0 | 0 | 1,2,3,4 |
| 0 | 0 | 1 | 1,2,3,4 |
| 0 | 1 | 0 | 1,2,3,4 |
| 0 | 1 | 1 | 1,2 |
| 1 | 0 | 0 | 3,4 |
| 1 | 0 | 1 | 3,4 |
| 1 | 1 | 0 | 1,2,3,4 |
| 1 | 1 | 1 | None |

Digital inputs code

| kWh | Digital input |
|---|---|
| 0 | ON |
| 1 | OFF |

| kvarh | Digital input |
|---|---|
| 0 | ON |
| 1 | OFF |

| F1 | Digital input |
|---|---|
| 0 | ON |
| 1 | OFF |

| F2 | Digital input |
|---|---|
| 0 | ON |
| 1 | OFF |

| Gas | Digital input |
|---|---|
| 0 | ON |
| 1 | OFF |

| Water | Digital input |
|---|---|
| 0 | ON |
| 1 | OFF |

The “Gas” digital input can be used to collect either the gas pulses or the kWh- pulses according to the chosen working mode (see paragraph 3.1.17).
The “Water” digital input can be used, according to the chosen working mode, to collect the water pulses, the kvarh- pulses, or to select between day or night gas tariff (see paragraph 3.1.17).

***Example 16*: reading of the “present modules+digital input status” word**

*1-word read request frame (8 byte)*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 08h | 00h | 00h | 01h | CRC | CRC |

*1-word read answer frmae (8 byte):*

| | | | | | | |
|---|---|---|---|---|---|---|
| 01h | 04h | 02h | **70h** | **F4h** | CRC | CRC |

Module variable value: 70F4h = 0111000011110100

Available modules: RS232, clock, digital output 1 and 2.
Digital input status: Gas = ON
Water = ON
kWh+ = OFF
kvarh+ = OFF
Tariff F1= OFF
tariff F2= OFF

## 2.13 HARMONIC ANALYSIS MAP

| Harmonic order | Voltages[1] (%) | | | Currents (%) | | | Relative angles[2] (°) | | |
|---|---|---|---|---|---|---|---|---|---|
| | L1-N | L2-N | L3-N | L1 | L2 | L3 | L1 | L2 | L3 |
| | ADD. | ADD. | ADD. | ADD. | ADD. | ADD. | ADD. | ADD. | ADD. |
| THD | 224 | 226 | 228 | 22A | 22C | 22E | | | |
| 1° | 238 | 23A | 23C | 23E | 240 | 242 | 244 | 246 | 248 |
| 2° | 24C | 24E | 250 | 252 | 254 | 256 | 258 | 25A | 25C |
| 3° | 260 | 262 | 264 | 266 | 268 | 26A | 26C | 26E | 270 |
| 4° | 274 | 276 | 278 | 27A | 27C | 27E | 280 | 282 | 284 |
| 5° | 288 | 28A | 28C | 28E | 290 | 292 | 294 | 296 | 298 |
| 6° | 29C | 29E | 2A0 | 2A2 | 2A4 | 2A6 | 2A8 | 2AA | 2AC |
| 7° | 2B0 | 2B2 | 2B4 | 2B6 | 2B8 | 2BA | 2BC | 2BE | 2C0 |
| 8° | 2C4 | 2C6 | 2C8 | 2CA | 2CC | 2CE | 2D0 | 2D2 | 2D4 |
| 9° | 2D8 | 2DA | 2DC | 2DE | 2E0 | 2E2 | 2E4 | 2E6 | 2E8 |
| 10° | 2EC | 2EE | 2F0 | 2F2 | 2F4 | 2F6 | 2F8 | 2FA | 2FC |
| 11° | 300 | 302 | 304 | 306 | 308 | 30A | 30C | 30E | 310 |
| 12° | 314 | 316 | 318 | 31A | 31C | 31E | 320 | 322 | 324 |
| 13° | 328 | 32A | 32C | 32E | 330 | 332 | 334 | 336 | 338 |
| 14° | 33C | 33E | 340 | 342 | 344 | 346 | 348 | 34A | 34C |
| 15° | 350 | 352 | 354 | 356 | 358 | 35A | 35C | 35E | 360 |
| 16° | 364 | 366 | 368 | 36A | 36C | 36E | 370 | 372 | 374 |
| 17° | 378 | 37A | 37C | 37E | 380 | 382 | 384 | 386 | 388 |
| 18° | 38C | 38E | 390 | 392 | 394 | 396 | 398 | 39A | 39C |
| 19° | 3A0 | 3A2 | 3A4 | 3A6 | 3A8 | 3AA | 3AC | 3AE | 3B0 |
| 20° | 3B4 | 3B6 | 3B8 | 3BA | 3BC | 3BE | 3C0 | 3C2 | 3C4 |
| 21° | 3C8 | 3CA | 3CC | 3CE | 3D0 | 3D2 | 3D4 | 3D6 | 3D8 |
| 22° | 3DC | 3DE | 3E0 | 3E2 | 3E4 | 3E6 | 3E8 | 3EA | 3EC |
| 23° | 3F0 | 3F2 | 3F4 | 3F6 | 3F8 | 3FA | 3FC | 3FE | 400 |
| 24° | 404 | 406 | 408 | 40A | 40C | 40E | 410 | 412 | 414 |
| 25° | 418 | 41A | 41C | 41E | 420 | 422 | 424 | 426 | 428 |
| 26° | 42C | 42E | 430 | 432 | 434 | 436 | 438 | 43A | 43C |
| 27° | 440 | 442 | 444 | 446 | 448 | 44A | 44C | 44E | 450 |
| 28° | 454 | 456 | 458 | 45A | 45C | 45E | 460 | 462 | 464 |
| 29° | 468 | 46A | 46C | 46E | 470 | 472 | 474 | 476 | 478 |
| 30° | 47C | 47E | 480 | 482 | 484 | 486 | 488 | 48A | 48C |
| 31° | 490 | 492 | 494 | 496 | 498 | 49A | 49C | 49E | 4A0 |
| 32° | 4A4 | 4A6 | 4A8 | 4AA | 4AC | 4AE | 4B0 | 4B2 | 4B4 |
| 33° | 4B8 | 4BA | 4BC | 4BE | 4C0 | 4C2 | 4C4 | 4C6 | 4C8 |
| 34° | 4CC | 4CE | 4D0 | 4D2 | 4D4 | 4D6 | 4D8 | 4DA | 4DC |
| 35° | 4E0 | 4E2 | 4E4 | 4E6 | 4E8 | 4EA | 4EC | 4EE | 4F0 |
| 36° | 4F4 | 4F6 | 4F8 | 4FA | 4FC | 4FE | 500 | 502 | 504 |
| 37° | 508 | 50A | 50C | 50E | 510 | 512 | 514 | 516 | 518 |
| 38° | 51C | 51E | 520 | 522 | 524 | 526 | 528 | 52A | 52C |
| 39° | 530 | 532 | 534 | 536 | 538 | 53A | 53C | 53E | 540 |
| 40° | 544 | 546 | 548 | 54A | 54C | 54E | 550 | 552 | 554 |
| 41° | 558 | 55A | 55C | 55E | 560 | 562 | 564 | 566 | 568 |
| 42° | 56C | 56E | 570 | 572 | 574 | 576 | 578 | 57A | 57C |
| 43° | 580 | 582 | 584 | 586 | 588 | 58A | 58C | 58E | 590 |
| 44° | 594 | 596 | 598 | 59A | 59C | 59E | 5A0 | 5A2 | 5A4 |
| 45° | 5A8 | 5AA | 5AC | 5AE | 5B0 | 5B2 | 5B4 | 5B6 | 5B8 |
| 46° | 5BC | 5BE | 5C0 | 5C2 | 5C4 | 5C6 | 5C8 | 5CA | 5CC |
| 47° | 5D0 | 5D2 | 5D4 | 5D6 | 5D8 | 5DA | 5DC | 5DE | 5E0 |
| 48° | 5E4 | 5E6 | 5E8 | 5EA | 5EC | 5EE | 5F0 | 5F2 | 5F4 |
| 49° | 5F8 | 5FA | 5FC | 5FE | 600 | 602 | 604 | 606 | 608 |
| 50° | 60C | 60E | 610 | 612 | 614 | 616 | 618 | 61A | 61C |
| THDo | 620 | 622 | 624 | 626 | 628 | 62A | | | |
| THDe | 634 | 636 | 638 | 63A | 63C | 63E | | | |

**NOTE:**
[1] According to the selected electrical system, the voltages can be Phase to Phase Voltage or Phase to Neutral Voltages.
[2] Negligible values when the selected system is without neutral.

All the variables of the previous table are contiguous. Note that using a single read command it is possible to read at most 120 words.

The values of the harmonic and distortion variables are represented as short (2 byte long). The decimal point must be set to "**111.1**" for distortion and angle variables (THD, THDo, THDe), and to "**111.11**" for the harmonic variables (h).
The stored values have physical meaning only if the harmonic analysis of the relevant phase is enabled (please refer to the user manual for FFT enable function, see also EEPROM map, address 200Ch).

***Example 17*: reading of the VL1 3rd order harmonic**

*"Value" request frame (frame 8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 02h | 5Ch | 00h | 01h | CRC | CRC |

*"Value" answer frame (frame 7 byte):*

| | | | | | | |
|---|---|---|---|---|---|---|
| 01h | 04h | 02h | 13h | 0Dh | CRC | CRC |

| | | |
|---|---|---|
| Variable value: | 0D13h | 3347 (decimal) |
| Value format: | 111.11 | |
| VL1 3rd order harmonic value | 33.47% | (the display shows 33.4%) |

***Example 18*: reading of the phase 1 - 3rd order relative angle**

*"Value" read request frame (frame 8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 02h | 68h | 00h | 01h | CRC | CRC |

*"Value" read answer frame (frame 7 byte):*

| | | | | | | |
|---|---|---|---|---|---|---|
| 01h | 04h | 02h | EFh | 06h | CRC | CRC |

| | | |
|---|---|---|
| Variable value: | 06EFh | 1775 (decimal) |
| Value format: | 111.1 | |
| Phase 1–3rd order relative angle: | 177.5° | (the display shows 177°) |

# 3 EEPROM VARIABLE MAP

**NOTE**: f.s. means full scale; b.s. means beginning of the scale

**WM4-96 configuration map**

| ADD. | VARIABLE | MAX | MIN | DEFAULT | BIT CHECK |
|---|---|---|---|---|---|
| 2000 | Password | 500 | 0 | 0 | 0101XXXX XXXXXXXX |
| 2002 | System | 4 | 0 | 2 | 0101XXXX XXXXXXXX |
| 2004 | CT | 60000 | 1 | 10 | not present |
| 2006 | VT | 60000 | 1 | 10 | not present |
| 2008 | type avg | 1 | 0 | 0 | 0101XXXX XXXXXXXX |
| 200A | time avg | 30 | 1 | 15 | 0101XXXX XXXXXXXX |
| 200C | enable FFT | See EEPROM data format tables | | | 0100XXXX XXXXXXXX |
| 200E | type digit | 1 | 0 | 0 | 0101XXXX XXXXXXXX |
| 2010 | field 1 e 2 | See EEPROM data format tables | | | 0101XXXX XXXXXXXX |
| 2012 | field 3 e 4 | See EEPROM data format tables | | | 0101XXXX XXXXXXXX |
| 2014 | RS485: address | 255 | 1 | 1 | not present |
| 2016 | RS485: baud rate | 3 | 0 | 3 (9600) | not present |
| 2018 | RS485: parity | 2 | 0 | 0 (none) | not present |
| 201A | RS232: baud rate | 3 | 0 | 3 (38400) | not present |
| 201C | Reserved | | | | |
| 201E | filter range | 1000 | 1 | 10 | not present |
| 2020 | filter coefficient | 255 | 1 | 3 | not present |
| 2022 | event selection | See EEPROM data format tables | | | not present |
| 2024 | event selection | See EEPROM data format tables | | | not present |
| 2026 | F.S. digital input | See EEPROM data format tables | | | 0101XXXX XXXXXXXX |
| 2028 | USA/EUROclockformat | See EEPROM data format tables | | | 0101XXXX XXXXXXXX |
| 202A | Language | See EEPROM data format tables | | | not present |
| 202C | Pulse-retransmitted tariff | See EEPROM data format tables | | | not present |
| 202E | Inputs type | See EEPROM data format tables | | | 0101 see note 1 |
| 2030 | Data log time interval | 60 | 1 | 15 | 0101XXXX XXXXXXXX |
| 2032 | --- | | | | |
| 2034 | --- | | | | |
| 2036 | Enabled dig.inputs | See EEPROM data format tables | | | see note 1 |
| 2038 | Pulses/m$^3$ Gas | 1000 | 1 | 1 | |
| 203A | Pulses/m$^3$ Water | 1000 | 1 | 1 | |
| 203C | dmd tariff alarm | 4 | 0 | 0 | 0101XXXX XXXXXXXX |
| 203E | Dig. out type | See EEPROM data format tables | | | not present |
| 2040 | Pulses/kWh out1 | 1000 | 1 | 1 | 01XXXXXX XXXXXXXX |
| 2042 | Pulses/kWh out2 | 1000 | 1 | 1 | 01XXXXXX XXXXXXXX |
| 2044 | Pulses/kWh out3 | 1000 | 1 | 1 | 01XXXXXX XXXXXXXX |
| 2046 | Pulses/kWh out4 | 1000 | 1 | 1 | 01XXXXXX XXXXXXXX |
| 2048 | Info dig. out 1 | See EEPROM data format tables | | | not present |
| 204A | Delay out 1 | 255 | 0 | 0 | 0101XXXX XXXXXXXX |
| 204C | Set-point out 1 | f.s | b.s. | 0 | not present |
| 204E | Hysteresis out 1 | f.s | 0 | 0 | not present |
| 2050 | Info dig. out 2 | See EEPROM data format tables | | | not present |
| 2052 | Delay out 2 | 255 | 0 | 0 | 0101XXXX XXXXXXXX |
| 2054 | Set-point out 2 | f.s | b.s. | 0 | not present |
| 2056 | Hysteresis out 2 | f.s | 0 | 0 | not present |
| 2058 | Info dig. out 3 | See EEPROM data format tables | | | not present |
| 205A | Delay out 3 | 255 | 0 | 0 | 0101XXXX XXXXXXXX |
| 205C | Set-point out 3 | f.s | b.s. | 0 | not present |
| 205E | Hysteresis out 3 | f.s | 0 | 0 | not present |
| 2060 | Info dig. out 4 | See EEPROM data format tables | | | not present |
| 2062 | Delay out 4 | 255 | 0 | 0 | 0101XXXX XXXXXXXX |
| 2064 | Set-point out 4 | f.s | b.s. | 0 | not present |
| 2066 | Hysteresis out 4 | f.s | 0 | 0 | not present |

**WM4-96 configuration map** (continue)

| ADD. | VARIABLE | MAX | MIN | DEFAULT | BIT CHECK |
|---|---|---|---|---|---|
| 2068 | Daylight saving time | 1 | 0 | 1 | 0101xxxx xxxxxxxx |
| 206A | Installed power t1 | See EEPROM data format tables | | | not present |
| 206C | Installed power t2 | See EEPROM data format tables | | | not present |
| 206E | Installed power t3 | See EEPROM data format tables | | | not present |
| 2070 | Installed power t4 | See EEPROM data format tables | | | not present |
| 2072 | Installed power min | See EEPROM data format tables | | | not present |
| 2074 | dmd variables | See EEPROM data format tables | | | not present |
| 2076 | dmd variables | See EEPROM data format tables | | | not present |
| 2078 | SMS events | See EEPROM data format tables | | | not present |
| 207A | SMS events | See EEPROM data format tables | | | not present |
| 207C | Phone numbers | 5 | 0 | | not present |
| 207E | SMS password | 255 | 1 | | not present |
| 2080 | Modem/SMS | 1 | 0 | | not present |
| 2082<br>2084 | Pulses/kWh input | 1000000 | 1 | 1000 | not present |
| 2086<br>2088 | Pulses/kvarh input | 1000000 | 1 | 1000 | not present |
| 208A | Tariff management type | See EEPROM data format tables | | | not present |
| 20C0 | type MAX1 | --- | --- | | 0101XXXX XXXXXXXX |
| 20C2 | type MAX2 | --- | --- | | 0101XXXX XXXXXXXX |
| 20C4 | type MAX3 | --- | --- | | 0101XXXX XXXXXXXX |
| 20C6 | type MAX4 | --- | --- | | 0101XXXX XXXXXXXX |
| 20C8 | type MAX5 | --- | --- | | 0101XXXX XXXXXXXX |
| 20CA | type MAX6 | --- | --- | | 0101XXXX XXXXXXXX |
| 20CC | type MAX7 | --- | --- | | 0101XXXX XXXXXXXX |
| 20CE | type MAX8 | --- | --- | | 0101XXXX XXXXXXXX |
| 20D0 | type MAX9 | --- | --- | | 0101XXXX XXXXXXXX |
| 20D2 | type MAX10 | --- | --- | | 0101XXXX XXXXXXXX |
| 20D4 | type MAX11 | --- | --- | | 0101XXXX XXXXXXXX |
| 20D6 | type MAX12 | --- | --- | | 0101XXXX XXXXXXXX |
| 20D8 | type MIN1 | --- | --- | | 0101XXXX XXXXXXXX |
| 20DA | type MIN2 | --- | --- | | 0101XXXX XXXXXXXX |
| 20DC | type MIN3 | --- | --- | | 0101XXXX XXXXXXXX |
| 20DE | type MIN4 | --- | --- | | 0101XXXX XXXXXXXX |
| 20E0 | type MIN5 | --- | --- | | 0101XXXX XXXXXXXX |
| 20E2 | type MIN6 | --- | --- | | 0101XXXX XXXXXXXX |
| 20E4 | type MIN7 | --- | --- | | 0101XXXX XXXXXXXX |
| 20E6 | type MIN8 | --- | --- | | 0101XXXX XXXXXXXX |

**NOTE 1**: these cells are to be set to 0 if no modules are installed.

**WM4-96 configuration map** (continue)

| ADD. | VARIABLE | MAX | MIN | DEFAULT | BIT CHECK |
|---|---|---|---|---|---|
| 2100 | val MAX1(msb) | | | | |
| 2101 | val MAX1(lsb) | | | | |
| 2102 | val MAX2(msb) | | | | |
| 2103 | val MAX2(lsb) | | | | |
| 2104 | val MAX3(msb) | | | | |
| 2105 | val MAX3(lsb) | | | | |
| 2106 | val MAX4(msb) | | | | |
| 2107 | val MAX4(lsb) | | | | |
| 2108 | val MAX5(msb) | | | | |
| 2109 | val MAX5(lsb) | | | | |
| 210A | val MAX6(msb) | | | | |
| 210B | val MAX6(lsb) | | | | |
| 210C | val MAX7(msb) | | | | |
| 210D | val MAX7(lsb) | | | | |
| 210E | val MAX8(msb) | | | | |
| 210F | val MAX8(lsb) | | | | |
| 2110 | val MAX9(msb) | | | | |
| 2111 | val MAX9(lsb) | | | | |
| 2112 | val MAX10(msb) | | | | |
| 2113 | val MAX10(lsb) | | | | |
| 2114 | val MAX11(msb) | | | | |
| 2115 | val MAX11(lsb) | | | | |
| 2116 | val MAX12(msb) | | | | |
| 2117 | val MAX12(lsb) | | | | |
| | | | | | |
| | | | | | |
| 2120 | val MIN1(msb) | | | | |
| 2121 | val MIN1(lsb) | | | | |
| 2122 | val MIN2(msb) | | | | |
| 2123 | val MIN2(lsb) | | | | |
| 2124 | val MIN3(msb) | | | | |
| 2125 | val MIN3(lsb) | | | | |
| 2126 | val MIN4(msb) | | | | |
| 2127 | val MIN4(lsb) | | | | |
| 2128 | val MIN5(msb) | | | | |
| 2129 | val MIN5(lsb) | | | | |
| 212A | val MIN6(msb) | | | | |
| 212B | val MIN6(lsb) | | | | |
| 212C | val MIN7(msb) | | | | |
| 212D | val MIN7(lsb) | | | | |
| 212E | val MIN8(msb) | | | | |
| 212F | val MIN8(lsb) | | | | |

## 3.1 EEPROM CONFIGURATION DATA FORMAT

### 3.1.1 Variable type coding

| VARIABLE | CODE | VARIABLE | CODE |
|---|---|---|---|
| V L1-N | 0 | PF L3 | 26 |
| V L2-N | 1 | PF ∑ | 27 |
| V L3-N | 2 | Hz | 28 |
| VL-N ∑ | 3 | THD V1 | 29 |
| V L1 | 4 | THDe V1 | 30 |
| V L2 | 5 | THDo V1 | 31 |
| V L3 | 6 | THD V2 | 32 |
| V ∑ | 7 | THDe V2 | 33 |
| A L1 | 8 | THDo V2 | 34 |
| A L2 | 9 | THD V3 | 35 |
| A L3 | 10 | THDe V3 | 36 |
| A n | 11 | THDo V3 | 37 |
| W L1 | 12 | THD A1 | 38 |
| W L2 | 13 | THDe A1 | 39 |
| W L3 | 14 | THDo A1 | 40 |
| W ∑ | 15 | THD A2 | 41 |
| Var L1 | 16 | THDe A2 | 42 |
| Var L2 | 17 | THDo A2 | 43 |
| Var L3 | 18 | THD A3 | 44 |
| Var ∑ | 19 | THDe A3 | 45 |
| VA L1 | 20 | THDo A3 | 46 |
| VA L2 | 21 | A dmd | 47 |
| VA L3 | 22 | VA dmd | 48 |
| VA ∑ | 23 | TPF avg | 49 |
| PF L1 | 24 | W dmd | 50 |
| PF L2 | 25 | var dmd | 51 |
| | | ASY | 52 |

### 3.1.2 System coding

| System | selection |
|---|---|
| 010XXXXX XXXXX000 | 1-phase |
| 010XXXXX XXXXX001 | 3+N phases bal |
| 010XXXXX XXXXX010 | 3+N phases unbal |
| 010XXXXX XXXXX011 | 3 phases bal |
| 010XXXXX XXXXX100 | 3 phases unbal |

### 3.1.3 Average type coding

| Average type | selection |
|---|---|
| XXXXXXXX XXXXXXX0 | avg fixed |
| XXXXXXXX XXXXXXX1 | avg float |
| 0101XXXX XXXXXXXX | bit check |

### 3.1.4 FFT enable coding

| FFT enable | selection |
|---|---|
| XXXXXXXX XXXXXX11 | fft V1-I1 disable |
| XXXXXXXX XXXXXX00 | fft V1-I1 enable |
| XXXXXXXX XXXX11XX | fft V2-I2 disable |
| XXXXXXXX XXXX00XX | fft V2-I2 enable |
| XXXXXXXX XX11XXXX | fft V3-I3 disable |
| XXXXXXXX XX00XXXX | fft V3-I3 enable |

### 3.1.5 Digit type coding

| digit type | Selection |
|---|---|
| XXXXXXXX XXXXXXX0 | 4 digit visualisation |
| XXXXXXXX XXXXXXX1 | 3½ digit visualisation |

### 3.1.6 Field n coding

The field n (n = 0, 1, 2, 3) variables are the variables chosen by the user to be shown on page 0 of the WM4 display.

### 3.1.7 Field (1 and 2) coding

| Field | Selection |
|---|---|
| XXXXXXXX XX000000 | field 1 variable |
| XXXX0000 00XXXXXX | field 2 variable |
| 0101XXXX XXXXXXXX | bit check |

### 3.1.8 Field (3 and 4) coding

| Field | Selection |
|---|---|
| XXXXXXXX XX000000 | field 3 variable |
| XXXX0000 00XXXXXX | field 4 variable |
| 0101XXXX XXXXXXXX | bit check |

### 3.1.9 RS485 baud rate coding

| RS485 baud rate | Selection |
|---|---|
| XXXXXXXX XXXXXX00 | 1200b |
| XXXXXXXX XXXXXX01 | 2400b |
| XXXXXXXX XXXXXX10 | 4800b |
| XXXXXXXX XXXXXX11 | 9600b |

### 3.1.10 RS485 parity coding

| RS485 parity | Selection |
|---|---|
| XXXXXXXX XXXXXX00 | None |
| XXXXXXXX XXXXXX01 | Odd |
| XXXXXXXX XXXXXX10 | Even |

### 3.1.11 RS232 baud rate coding

| RS232 baud rate | Selection |
|---|---|
| XXXXXXXX XXXXXX00 | 2400 baud |
| XXXXXXXX XXXXXX01 | 4800 baud |
| XXXXXXXX XXXXXX10 | 9600 baud |
| XXXXXXXX XXXXXX11 | 38400 baud |

### 3.1.12 Selected events coding

| EVENT | Bit | EVENT | bit |
|---|---|---|---|
| MAX 1 | 0 | MAX 11 | 10 |
| MAX 2 | 1 | MAX 12 | 11 |
| MAX 3 | 2 | MIN 1 | 12 |
| MAX 4 | 3 | MIN 2 | 13 |
| MAX 5 | 4 | MIN 3 | 14 |
| MAX 6 | 5 | MIN 4 | 15 |
| MAX 7 | 6 | MIN 5 | 16 |
| MAX 8 | 7 | MIN 6 | 17 |
| MAX 9 | 8 | MIN 7 | 18 |
| MAX 10 | 9 | MIN 8 | 19 |

| 4 BYTE | |
|---|---|
| XXXXXXXX XXXXXXXX | $n^{th}$ bit=0 event not selected; $n^{th}$ bit=1 event selected |
| XXXXXXXX XXXXXXXX | $n^{th}$ bit=0 event not selected; $n^{th}$ bit=1 event selected |

### 3.1.13 Digital inputs full scale coding

| Full scale | Selection |
|---|---|
| XXXXXXXX XXXXX000 | 32.00 kW or kvar |
| XXXXXXXX XXXXX001 | 320.0 kW or kvar |
| XXXXXXXX XXXXX010 | 3.200 MW or Mvar |
| XXXXXXXX XXXXX011 | 32.00 MW or Mvar |
| XXXXXXXX XXXXX100 | 320.0 MW or Mvar |
| 0101XXXX XXXXXXXX | bit check |

### 3.1.14 USA/EURO clock format

| Clock format | Selection |
|---|---|
| XXXXXXXX XXXXXXX0 | USA clock format |
| XXXXXXXX XXXXXXX1 | European clock format |
| 0101XXXX XXXXXXXX | bit check |

### 3.1.15 Language

| Language format | Selection |
|---|---|
| XXXXXXXX XXXXX000 | English |
| XXXXXXXX XXXXX001 | Italian |
| XXXXXXXX XXXXX010 | German |
| XXXXXXXX XXXXX011 | French |
| XXXXXXXX XXXXX100 | Spanish |
| 0101XXXX XXXXXXXX | bit check |

### 3.1.16 Pulse type selection

| Pulse type selection | Selection |
|---|---|
| XXXXXXXX XXXXX000 | Out 1 pulses related to: total energy meter |
| XXXXXXXX XXXXX001 | Out 1 pulses related to: period 1 energy meter |
| XXXXXXXX XXXXX010 | Out 1 pulses related to: period 2 energy meter |
| XXXXXXXX XXXXX011 | Out 1 pulses related to: period 3 energy meter |
| XXXXXXXX XXXXX100 | Out 1 pulses related to: period 4 energy meter |
| XXXXXXXX XX000XXX | Out 2 pulses related to: total energy meter |
| XXXXXXXX XX001XXX | Out 2 pulses related to: period 1 energy meter |
| XXXXXXXX XX010XXX | Out 2 pulses related to: period 2 energy meter |
| XXXXXXXX XX011XXX | Out 2 pulses related to: period 3 energy meter |
| XXXXXXXX XX100XXX | Out 2 pulses related to: period 4 energy meter |
| XXXXXXX0 00XXXXXX | Out 3 pulses related to: total energy meter |
| XXXXXXX0 01XXXXXX | Out 3 pulses related to: period 1 energy meter |
| XXXXXXX0 10XXXXXX | Out 3 pulses related to: period 2 energy meter |
| XXXXXXX0 11XXXXXX | Out 3 pulses related to: period 3 energy meter |
| XXXXXXX1 00XXXXXX | Out 3 pulses related to: period 4 energy meter |
| XXXX000X XXXXXXXX | Out 4 pulses related to: total energy meter |
| XXXX001X XXXXXXXX | Out 4 pulses related to: period 1 energy meter |
| XXXX010X XXXXXXXX | Out 4 pulses related to: period 2 energy meter |
| XXXX011X XXXXXXXX | Out 4 pulses related to: period 3 energy meter |
| XXXX100X XXXXXXXX | Out 4 pulses related to: period 4 energy meter |

### 3.1.17 Input type coding

| Field | Selection |
|---|---|
| XXXXXXXX XXXXXXX0 | Measure (from analogue input module) |
| XXXXXXXX XXXXXXX1 | Pulse (pulses from official watthourmeter) |
| 0101XXXX XXXXXXXX | bit check |

### 3.1.18 Enabled digital input coding

| Diginput_on | Selection |
|---|---|
| ------- --- --- --X | Input A1 (1=enabled, 0=disabled) |
| ------- --- --- -X- | Input A2 (1=enabled, 0=disabled) |
| ------- --- --- X-- | Input A3 (1=enabled, 0=disabled) |
| ------- --- --X --- | Input C1 (1=enabled, 0=disabled) |
| ------- --- -X- --- | Input C2 (1=enabled, 0=disabled) |
| ------- --- X-- --- | Input C3 (1=enabled, 0=disabled) |
| ------X --- --- --- | Input A2 (1=dual tariff gas metering, 0=water metering) |
| -----X- --- --- --- | Inputs A1 and A2<br>(1=negative energies metering, 0=gas and water metering) |

### 3.1.19 Coding of the tariff associated to the Wdmd alarm

| Alarm tariff | Selection |
|---|---|
| ------- --- --- XXX | Tariff of alarm 1 (from 0 to 4, 0 means total) |
| ------- --- XXX --- | Tariff of alarm 2 (from 0 to 4, 0 means total) |
| ------- XXX --- --- | Tariff of alarm 3 (from 0 to 4, 0 means total) |
| ----XXX --- --- --- | Tariff of alarm 4 (from 0 to 4, 0 means total) |
| 0101XXXX XXXXXXXX | bit check |

### 3.1.20 Digital output type coding

| type dig out | selection |
|---|---|
| XXXXXXXX XXXXXX00 | dig out 1 pulse (default type out 1) |
| XXXXXXXX XXXXXX01 | dig out 1 control |
| XXXXXXXX XXXXXX10 | dig out 1 alarm |
| XXXXXXXX XXXX00XX | dig out 2 pulse (default type out 2) |
| XXXXXXXX XXXX01XX | dig out 2 control |
| XXXXXXXX XXXX10XX | dig out 2 alarm |
| XXXXXXXX XX00XXXX | dig out 3 pulse (default type out 3) |
| XXXXXXXX XX01XXXX | dig out 3 control |
| XXXXXXXX XX10XXXX | dig out 3 alarm |
| XXXXXXXX 00XXXXXX | dig out 4 pulse (default type out 4) |
| XXXXXXXX 01XXXXXX | dig out 4 control |
| XXXXXXXX 10XXXXXX | dig out 4 alarm |
| XXXXXX00 XXXXXXXX | pulse 1 Kwh+ (default) (see note 1) |
| XXXXXX01 XXXXXXXX | pulse 1 Kwh- (see note 1) |
| XXXXXX10 XXXXXXXX | pulse 1 Kvarh+ (see note 1) |
| XXXXXX11 XXXXXXXX | pulse 1 Kvarh- (see note 1) |
| XXXX00XX XXXXXXXX | pulse 2 Kwh+ (default) (see note 1) |
| XXXX01XX XXXXXXXX | pulse 2 Kwh- (see note 1) |
| XXXX10XX XXXXXXXX | pulse 2 Kvarh+ (see note 1) |
| XXXX11XX XXXXXXXX | pulse 2 Kvarh- (see note 1) |
| XX00XXXX XXXXXXXX | pulse 3 Kwh+ (default) (see note 1) |
| XX01XXXX XXXXXXXX | pulse 3 Kwh- (see note 1) |
| XX10XXXX XXXXXXXX | pulse 3 Kvarh+ (see note 1) |
| XX11XXXX XXXXXXXX | pulse 3 Kvarh- (see note 1) |
| 00XXXXXX XXXXXXXX | pulse 4 Kwh+ (default) (see note 1) |
| 01XXXXXX XXXXXXXX | pulse 4 Kwh- (see note 1) |
| 10XXXXXX XXXXXXXX | pulse 4 Kvarh+ (see note 1) |
| 11XXXXXX XXXXXXXX | pulse 4 Kvarh- (see note 1) |

**NOTE 1**: the multiplier type depends on the “info P” variable (refer to the instantaneous variables map).

### 3.1.21 Info out (1,2,3,4) coding

| info out | Selection |
|---|---|
| XXXXXXXX XX000000 | Variable type(from 000000 to 110100, see table 3.1.1) |
| XXXXXX00 00XXXXXX | Control type «up» (default) |
| XXXXXX00 01XXXXXX | Control type «up.l» |
| XXXXXX00 10XXXXXX | Control type «do» |
| XXXXXX00 11XXXXXX | Control type «do.l» |
| XXXXX0XX XXXXXXXX | Normally de-energised relay |
| XXXXX1XX XXXXXXXX | Normally energised relay |

### 3.1.22 Daylight saving time coding

| Tariff management | Selection |
|---|---|
| XXXXXXXX XXXXXX00 | ON |
| XXXXXXXX XXXXXX01 | OFF |
| 0101XXXX XXXXXXXX | bit check |

### 3.1.23 Installed power coding

| | |
|---|---|
| Installed power | |
| XXXXXXXX XXXXYYYY | XXXXXXXXXXXX -→ MANTISSA YYYY -→ EXPONENT |

The mantissa is lower than 2000 and the exponent lower than 16.
The YYYY value corresponds to the value on the "INF" table, with which it is possible to find the engineering unit to be associated to the XXXXXXXXXXXX mantissa.

### 3.1.24 dmd variable coding

| **VARIABLE** | **Bit** | **VARIABLE** | **bit** |
|---|---|---|---|
| V L1-N | 0 | W L2 | 13 |
| V L2-N | 1 | W L3 | 14 |
| V L3-N | 2 | W ∑ | 15 |
| VL-N ∑ | 3 | var L1 | 16 |
| V L1 | 4 | var L2 | 17 |
| V L2 | 5 | var L3 | 18 |
| V L3 | 6 | VAR ∑ | 19 |
| V ∑ | 7 | VA L1 | 20 |
| A L1 | 8 | VA L2 | 21 |
| A L2 | 9 | VA L3 | 22 |
| A L3 | 10 | VA ∑ | 23 |
| A ∑ | 11 | PF ∑ | 24 |
| W L1 | 12 | | |

| | |
|---|---|
| 4 BYTE | |
| XXXXXXXX XXXXXXXX | $n^{th}$ bit=0 variable not selected; $n^{th}$ bit=1 variable selected |
| XXXXXXXX XXXXXXXX | $n^{th}$ bit=0 variable not selected; $n^{th}$ bit=1 variable selected |

The maximum number of selectable variable is 8.

### 3.1.25 SMS event coding

| **VARIABLE** | **bit** |
|---|---|
| Alarm1 | 0 |
| Alarm2 | 1 |
| Alarm3 | 2 |
| Alarm4 | 3 |

| | |
|---|---|
| 4 BYTE | |
| XXXXXXXX XXXXXXXX | $n^{th}$ bit=0 variable not selected; $n^{th}$ bit=1 variable selected |
| XXXXXXXX XXXXYYYY | $n^{th}$ bit=0 variable not selected; $n^{th}$ bit=1 variable selected |

Only the first 4 bits (YYYY) are considered.

### 3.1.26 Pulses/kWh (or kvarh) coding.

| 4 BYTE | MAX value |
|---|---|
| XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX | 1000000 |

The stored value is an integer from 1 to 1000000. The real value is a fixed point number (two decimals) from 0.01 to 10000.00.

### 3.1.27 Tariff management coding

| Tariff management | Selection |
|---|---|
| XXXXXXXX XXXXXX00 | Single tariff |
| XXXXXXXX XXXXXX01 | Dual tariff |
| XXXXXXXX XXXXXX10 | Multi tariff |

### 3.1.28 MAX and MIN type coding

| MAX and MIN type | Selection |
|---|---|
| XXXXXXXX XX000000 | variable field (from 000000 to 110100, see TABLE 3.1.1) |
| 0101XXXX XXXXXXXX | bit check |

## 3.2 EVENT LOGGING

**Event logging map**

| | | |
|---|---|---|
| 2300 | 4 words | Event 1 |
| 2308 | 4 words | Event 2 |
| 2310 | 4 words | Event 3 |
| 2318 | 4 words | Event 4 |
| 2320 | 4 words | Event 5 |
| 2328 | 4 words | Event 6 |
| ------- | ------- | ------- |
| 31F8 | 4 words | Event 480 |

The stored information relevant to every event are the following: event type, hour, minutes, seconds, day, month, year, value.
All these data are included in the relevant 4 words, coded as follow.

To reset the events, it is necessary to write 0 in every of the sideways listed addresses and to reset the event counter, placed at the address 80Ch.

**N$^{th}$ event coding**

| | hour | min | event type |
|---|---|---|---|
| Word1 | XXXXX | XXXXXX | XXXXX |

| | month | day | year |
|---|---|---|---|
| Word2 | XXXX | XXXXX | XXXXXXX |

| | Seconds | variable type |
|---|---|---|
| Word3 | 0101XXXXXX | XXXXXX |

| | value |
|---|---|
| Word4 | XXXXXXXXXXXXXXXX |

**Variable type coding:**

Refer to the relevant table in paragraph 3.1.1.

**Power variable format:**

| Power variable format | |
|---|---|
| XXXXXXXX XXXXYYYY | XXXXXXXXXXXX -→ MANTISSA YYYY -→ EXPONENT |

The mantissa is lower than 2000 and the exponent lower than 16.
The YYYY value corresponds to the value on the "INF" table, with which it is possible to find the engineering unit to be associated to the XXXXXXXXXXXX mantissa.

**Event type coding:**

| | | |
|---|---|---|
| MAX | | 1 |
| MIN | | 2 |
| DIAGNOSTIC 1 | ON | 3 |
| DIAGNOSTIC 2 | ON | 4 |
| DIAGNOSTIC 3 | ON | 5 |
| DIAGNOSTIC 4 | ON | 6 |
| DIAGNOSTIC 1 | OFF | 7 |
| DIAGNOSTIC 2 | OFF | 8 |
| DIAGNOSTIC 3 | OFF | 9 |
| DIAGNOSTIC 4 | OFF | 10 |
| REMOTE 1 | ON | 11 |
| REMOTE 2 | ON | 12 |
| REMOTE 3 | ON | 13 |
| REMOTE 4 | ON | 14 |
| REMOTE 1 | OFF | 15 |
| REMOTE 2 | OFF | 16 |
| REMOTE 3 | OFF | 17 |
| REMOTE 4 | OFF | 18 |
| ALARM 1 | ON | 19 |
| ALARM 2 | ON | 20 |
| ALARM 3 | ON | 21 |
| ALARM 4 | ON | 22 |
| ALARM 1 | OFF | 23 |
| ALARM 2 | OFF | 24 |
| ALARM 3 | OFF | 25 |
| ALARM 4 | OFF | 26 |

### 3.3 MONTHLY ENERGY METERS

The reading of the values of the energy meters relevant to the previous three months is feasible by reading the data stored in the three tables described below. The tables have the same structure: they are composed of 14 32-bytes pages where the total and partial meter values are stored on the first day of the month at 0.00.00. The storing order of the table is the following (assuming, for example, to begin the WM4 use in January): January data = table A, February data = table B, March data = table C, April data = table A (overwriting the January data), and so on.

Pages structure:

Page 1: the initial 16 bytes, grouped 4 by 4, are the four-total meter LSB part
(KWh+ ,KWh-, Kvarh+, Kvarh-)
Page 2: the initial 20 bytes, grouped 5 by 5, are the four-winter tariff 1 partial meters values
Page 3: the initial 20 bytes, grouped 5 by 5, are the four-winter tariff 2 partial meters values
Page 4: the initial 20 bytes, grouped 5 by 5, are the four-winter tariff 3 partial meters values
Page 5: the initial 20 bytes, grouped 5 by 5, are the four-winter tariff 4 partial meters values
Page 6: the initial 20 bytes, grouped 5 by 5, are the four-summer tariff 1 partial meters values
Page 7: the initial 20 bytes, grouped 5 by 5, are the four-summer tariff 2 partial meters values
Page 8: the initial 20 bytes, grouped 5 by 5, are the four-summer tariff 3 partial meters values
Page 9: the initial 20 bytes, grouped 5 by 5, are the four-summer tariff 4 partial meters values
Page 10: the initial 20 bytes, grouped 5 by 5, are the four-holiday tariff 1 partial meters values
Page 11: the initial 20 bytes, grouped 5 by 5, are the four-holiday tariff 2 partial meters values
Page 12: the initial 20 bytes, grouped 5 by 5, are the four-holiday tariff 3 partial meters values
Page 13: the initial 20 bytes, grouped 5 by 5, are the four-holiday tariff 4 partial meters values
Page 14: the initial 4 bytes are the four-total meter MSB part, then 10 not used bytes follow, then the following two bytes are relevant respectively to the year and month when the table were stored.

*How to reconstruct the energy meter values:*

The energy values have to be reconstructed according to the procedure described in paragraph 2.5. The value of byte 5, multiplied by 1000000000, must be added to the byte1-byte2-byte3-byte4 value and the sum divided by 100.

- Total meters: byte5 is stored at page 14 of the relevant monthly table. byte1-byte2-byte3-byte4 are stored at page 1 (byte 1 has the lower address).

- Partial meters: byte5 and byte1-byte2-byte3-byte4 are consecutively stored starting from the address of the required meter (byte 5 has the lower address, then byte 1 is stored, etc.).

To obtain the energy consumption relevant to a given month, the tables relevant to the end and the beginning of that month must be read, and the difference between the respective values must be carried out.

### 3.3.1 Monthly energy meters map

| ADDRESS | BYTE | SEASON | PERIOD | METER TYPE |
|---|---|---|---|---|
| 3220 (page 1) | 4 | TOTAL LSB | | Kwh+ (LSB) |
| 3224 | 4 | | | Kwh- (LSB) |
| 3228 | 4 | | | Kvarh+ (LSB) |
| 322C | 4 | | | Kvarh- (LSB) |
| | 4 | | | |
| | 4 | | | |
| | 4 | | | |
| | 4 | | | |
| 3240 (page 2) | 4 | WINTER | 1 | Kwh+ (LSB) |
| 3244 | 1 | | | Kwh+ (MSB) |
| 3245 | 4 | | | Kvarh+ (LSB) |
| 3249 | 1 | | | Kvarh+ (MSB) |
| 324A | 4 | | | Kwh- (LSB) |
| 324E | 1 | | | Kwh- (MSB) |
| 324F | 4 | | | Kvarh- (LSB) |
| 3253 | 1 | | | Kvarh- (MSB) |
| 3254 | 12 | | | |
| 3260 (page 3) | 4 | WINTER | 2 | Kwh+ (LSB) |
| 3264 | 1 | | | Kwh+ (MSB) |
| 3265 | 4 | | | Kvarh+ (LSB) |
| 3269 | 1 | | | Kvarh+ (MSB) |
| 326A | 4 | | | Kwh- (LSB) |
| 326E | 1 | | | Kwh- (MSB) |
| 326F | 4 | | | Kvarh- (LSB) |
| 3263 | 1 | | | Kvarh- (MSB) |
| 3264 | 12 | | | |
| | | | | |
| … | … | … | … | … |
| | | | | |
| … | … | … | … | … |
| | | | | |
| … | … | … | … | … |
| | | | | |
| 33A0 (page 13) | 4 | HOLIDAY | 4 | Kwh+ (LSB) |
| 33A4 | 1 | | | Kwh+ (MSB) |
| 33A5 | 4 | | | Kvarh+ (LSB) |
| 33A9 | 1 | | | Kvarh+ (MSB) |
| 33AA | 4 | | | Kwh- (LSB) |
| 33AE | 1 | | | Kwh- (MSB) |
| 33AF | 4 | | | Kvarh- (LSB) |
| 33B3 | 1 | | | Kvarh- (MSB) |
| 33B4 | 12 | | | |
| 33C0 (page 14) | 1 | TOTAL MSB | | Kwh+ (MSB) |
| 33C1 | 1 | | | Kwh- (MSB) |
| 33C2 | 1 | | | Kvarh+ (MSB) |
| 33C3 | 1 | | | Kvarh- (MSB) |
| 33C4 | 10 | | | |
| 33CE | 1 | YEAR/MONTH | | YEAR |
| 33CF | 1 | | | MONTH |
| 33D0 | 16 | | | |

Table A

### 3.3.2 Monthly energy meters map

| ADDRESS | BYTE | SEASON | PERIOD | METER TYPE |
|---|---|---|---|---|
| 33E0 (page 1) | 4 | TOTAL LSB | | Kwh+ (LSB) |
| 33E4 | 4 | | | Kwh- (LSB) |
| 33E8 | 4 | | | Kvarh+ (LSB) |
| 33EC | 4 | | | Kvarh- (LSB) |
| | 4 | | | |
| | 4 | | | |
| | 4 | | | |
| | 4 | | | |
| 3400 (page 2) | 4 | WINTER | 1 | Kwh+ (LSB) |
| 3404 | 1 | | | Kwh+ (MSB) |
| 3405 | 4 | | | Kvarh+ (LSB) |
| 3409 | 1 | | | Kvarh+ (MSB) |
| 340A | 4 | | | Kwh- (LSB) |
| 340E | 1 | | | Kwh- (MSB) |
| 340F | 4 | | | Kvarh- (LSB) |
| 3413 | 1 | | | Kvarh- (MSB) |
| 3414 | 12 | | | |
| 3420 (page 3) | 4 | WINTER | 2 | Kwh+ (LSB) |
| 3424 | 1 | | | Kwh+ (MSB) |
| 3425 | 4 | | | Kvarh+ (LSB) |
| 3429 | 1 | | | Kvarh+ (MSB) |
| 342A | 4 | | | Kwh- (LSB) |
| 342E | 1 | | | Kwh- (MSB) |
| 342F | 4 | | | Kvarh- (LSB) |
| 3433 | 1 | | | Kvarh- (MSB) |
| 3434 | 12 | | | |
| | | | | |
| … | … | … | … | …. |
| | | | | |
| … | … | … | … | … |
| | | | | |
| … | … | … | … | … |
| | | | | |
| 3540 (page 13) | 4 | HOLIDAY | 4 | Kwh+ (LSB) |
| 3544 | 1 | | | Kwh+ (MSB) |
| 3545 | 4 | | | Kvarh+ (LSB) |
| 3549 | 1 | | | Kvarh+ (MSB) |
| 354A | 4 | | | Kwh- (LSB) |
| 354E | 1 | | | Kwh- (MSB) |
| 354F | 4 | | | Kvarh- (LSB) |
| 3553 | 1 | | | Kvarh- (MSB) |
| 3554 | 12 | | | |
| 3560 (page 14) | 1 | TOTAL MSB | | Kwh+ (MSB) |
| 3561 | 1 | | | Kwh- (MSB) |
| 3562 | 1 | | | Kvarh+ (MSB) |
| 3563 | 1 | | | Kvarh- (MSB) |
| 3564 | 10 | | | |
| 356E | 1 | YEAR/MONTH | | YEAR |
| 356F | 1 | | | MONTH |
| 3570 | 16 | | | |

Table B

### 3.3.3 Monthly energy meters map

| ADDRESS | BYTE | SEASON | PERIOD | METER TYPE |
|---|---|---|---|---|
| 35A0 (page 1) | 4 | TOTAL LSB | | Kwh+ (LSB) |
| 35A4 | 4 | | | Kwh- (LSB) |
| 35A8 | 4 | | | Kvarh+ (LSB) |
| 35AC | 4 | | | Kvarh- (LSB) |
| | 4 | | | |
| | 4 | | | |
| | 4 | | | |
| | 4 | | | |
| 35C0 (page 2) | 4 | WINTER | 1 | Kwh+ (LSB) |
| 35C4 | 1 | | | Kwh+ (MSB) |
| 35C5 | 4 | | | Kvarh+ (LSB) |
| 35C9 | 1 | | | Kvarh+ (MSB) |
| 35CA | 4 | | | Kwh- (LSB) |
| 35CE | 1 | | | Kwh- (MSB) |
| 35CF | 4 | | | Kvarh- (LSB) |
| 35C3 | 1 | | | Kvarh- (MSB) |
| 35C4 | 12 | | | |
| 35E0 (page 3) | 4 | WINTER | 2 | Kwh+ (LSB) |
| 35E4 | 1 | | | Kwh+ (MSB) |
| 35E5 | 4 | | | Kvarh+ (LSB) |
| 35E9 | 1 | | | Kvarh+ (MSB) |
| 35EA | 4 | | | Kwh- (LSB) |
| 35EE | 1 | | | Kwh- (MSB) |
| 35EF | 4 | | | Kvarh- (LSB) |
| 35E3 | 1 | | | Kvarh- (MSB) |
| 35E4 | 12 | | | |
| | | | | |
| … | … | … | … | … |
| | | | | |
| … | … | … | … | … |
| | | | | |
| … | … | … | … | … |
| | | | | |
| 3720 (page 13) | 4 | HOLIDAY | 4 | Kwh+ (LSB) |
| 3724 | 1 | | | Kwh+ (MSB) |
| 3725 | 4 | | | Kvarh+ (LSB) |
| 3729 | 1 | | | Kvarh+ (MSB) |
| 372A | 4 | | | Kwh- (LSB) |
| 372E | 1 | | | Kwh- (MSB) |
| 372F | 4 | | | Kvarh- (LSB) |
| 3733 | 1 | | | Kvarh- (MSB) |
| 3734 | 12 | | | |
| 3740 (page 14) | 1 | TOTAL MSB | | Kwh+ (MSB) |
| 3741 | 1 | | | Kwh- (MSB) |
| 3742 | 1 | | | Kvarh+ (MSB) |
| 3743 | 1 | | | Kvarh- (MSB) |
| 3744 | 10 | | | |
| 374E | 1 | YEAR/MONTH | | YEAR |
| 374F | 1 | | | MONTH |
| 3750 | 16 | | | |

Table C

### 3.4 EXAMPLES: HOW TO READ THE DATA FROM EEPROM

**NOTE**: EEPROM is structured in word (if not differently advised) which are sent in the order MSB, LSB.

The value of the variables stored in EEPROM are 4-byte integer except from the values of the power which are stored in a different way. Refer to example 22 to know how to read the power values.

#### 3.4.1 Reading and resetting maximum and minumum

***Example 19*: "12th MAXIMUM variable type" read command**

*4-word read command request frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 20h | D6h | 00h | 01h | CRC | CRC |

*read command answer frame (7 byte):*

| | | | | | | |
|---|---|---|---|---|---|---|
| 01h | 04h | 02h | **50h** | **0Ah** | CRC | CRC |

12th MAX-variable type address: 20D6h
Stored variable value: 0Ah = 10 (decimal)
Variable type: A L3 (phase 3 current)

***Example 20*: "Current info" read command**

*"Info A" read request frame (frame 8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 00h | E8h | 00h | 01h | CRC | CRC |

*"Info A" read answer frame (frame 7 byte):*

| | | | | | | |
|---|---|---|---|---|---|---|
| 01h | 04h | 02h | 06h | 04h | CRC | CRC |

Info V value: 06h decimal point position: 1111
Info A value: 04h decimal point position: 11.11

***Example 21*: value of the "12th MAXIMUM" read command**

*1-word read request command (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 21h | 16h | 00h | 01h | CRC | CRC |

*read answer frame (7 byte):*

| | | | | | | |
|---|---|---|---|---|---|---|
| 01h | 04h | 02h | **03h** | **6Ch** | CRC | CRC |

Address of 12th MAX value: 2116h
Stored value: 036Ch = 876 (decimal)
Taking into account the results of the previous examples:
A L3 value: 8.76 A

**_Example 22_: value of the "12th MAXIMUM" read command in case of "power type" variable**

The structure of the value of the power stored in EEPROM is the following:

| Word 1 | | Word 2 | |
|---|---|---|---|
| 2116h | | 2117h | |
| MSB | … | LSB | d.p. |

*1-word read request command (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 21h | 16h | 00h | 01h | CRC | CRC |

*read answer frame (7 byte):*

| | | | | | | |
|---|---|---|---|---|---|---|
| 01h | 04h | 02h | **19h** | **E8h** | CRC | CRC |

| Value | | | d.p. |
|---|---|---|---|
| **1h** | **9h** | **Eh** | **8h** |

Address of 12th MAX value: 2116h
Stored value: 19E8h ➔ value = 19Eh = 414 (decimal);
decimal point position code = 8h = 111.1 k
Considering that, for example, the variable code is 12 (W L1):

W L1 value: 4.14 kW

**_Example 23_: "12th MAXIMUM" reset command**

*1-word write request command (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 21h | 16h | 00h | 00h | CRC | CRC |

*Write answer command (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 06h | 21h | 16h | 00h | 00h | CRC | CRC |

### 3.4.2 Event reading

***Example 24*: read command of the event stored at address 2240h**

The reading of the information regarding an event is carried out by transferring 4 words starting from the first address of the selected event location, according to the Event Logging Map table (see paragraph 3.2).
The description of the event is obtained by decoding the data contained in the 4 words, according to "n$^{th}$ event coding" table.

In accordance to the above listed procedure, before reading a MAX or MIN event, the variable associated to the MAX or MIN must be known. Then the info of the variable (decimal point position) must be acquired. Finally the stored value must be read.

4-word read command frame (8 byte):

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 04h | 22h | 40h | 00h | 04h | FAh | 65h |

read command answer frame (13 byte):

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 08h | **7Bh** | **C1h** | **61h** | **80h** | **5Bh** | **42h** | **00h** | **36h** | 88h | DFh |

| | |
|---|---|
| Word 1: | 7BC1h = 01111 011110 00001 |
| Word 2: | 6180h = 0110 00011 0000000 |
| Word 3: | 5B42h = 0101 101101 000010 |
| Word 4: | 0036h = 0000000000110110 |

| | | | |
|---|---|---|---|
| Event type | 0001 | 1 | MAX |
| Minutes | 011110 | 30 | |
| Hour | 01111 | 15 | |
| Year | 0000000 | 00 | |
| Day | 00011 | 03 | |
| Month | 0110 | 06 | |
| Variable type | 001000 | 08 | A L1 |
| Seconds | 101101 | 45 | |
| Value | 110110 | 54 | |

The engineering unit and the decimal point position of the variable are obtained by reading the "info" value in the instantaneous variables area (see example 2).

### 3.5 RTC MAPPING

| ADD. | BYTE | VARIABLE | Coding (with bit check) |
|---|---|---|---|
| 4000 | 1 | Seconds | Hex value |
| 4001 | 1 | Minutes | Hex value |
| 4002 | 1 | Hours | Hex value |
| 4003 | 1 | Day of the week | Hex value |
| 4004 | 1 | Day of the month | Hex value |
| 4005 | 1 | Month | Hex value |
| 4006 | 1 | Year | Hex value |
| 4007 | 1 | | |
| 4008 | 1 | | |
| 4009 | 1 | | |
| 400A | 1(LSB) | Winter starting date | XXXYYYYY |
| | 1(MSB) | | 0101010X |
| 400C | 1(LSB) | Winter finish date | XXXYYYYY |
| | 1(MSB) | | 0101010X |
| 400E | 1(LSB) | End of 1st Winter period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 4010 | 1(LSB) | End of 2nd Winter period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 4012 | 1(LSB) | End of 3rd Winter period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 4014 | 1(LSB) | End of 4th Winter period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 4016 | 1(LSB) | End of 5th Winter period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 4018 | 1(LSB) | End of 6th Winter period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 401A | 1(LSB) | End of 7th Winter period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 401C | 1(LSB) | End of 8th Winter period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 401E | 1(LSB) | End of 1st Summer period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 4020 | 1(LSB) | End of 2nd Summer period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 4022 | 1(LSB) | End of 3rd Summer period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 4024 | 1(LSB) | End of 4th Summer period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 4026 | 1(LSB) | End of 5th Summer period | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 4028 | 1(LSB) | Start Holiday | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 402A | 1(LSB) | End Holiday | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 402C | 1(LSB) | Holiday rate | XXYYYYYY |
| | 1(MSB) | | 010WWXXX |
| 402E | 1 | | |

**NOTE**: the MSB/LSB notation is used to define how the data should be reconstructed (see below). The LSB is always sent before the MSB, in both reading and writing frames.

The first 7 bytes are relevant to the system clock. If WM4 is programmed for dual tariff or multitariff management, the locations from 400Ah to 402Dh are to be programmed according to the following coding.

Winter starting date

| 2 BYTE | |
|---|---|
| 0101010X XXXYYYYY | bit Y (0←→4) first winter day<br>bit X (5←→8) month of the first winter day |

Winter finish date

| 2 BYTE | |
|---|---|
| 0101010X XXXYYYYY | bit Y (0←→4) last winter day<br>bit X (5←→8) month of the last winter day |

The summer season is defined automatically, after setting the winter one.

Holiday starting date

| 2 BYTE | |
|---|---|
| 0101010X XXXYYYYY | bit Y (0←→4) first holiday day<br>bit X (5←→8) month of the first holiday day |

Holiday finish date

| 2 BYTE | |
|---|---|
| 0101010X XXXYYYYY | bit Y (0←→4) last holiday day<br>bit X (5←→8) month of the last holiday day |

Day periods finish hour (Winter, Summer, Holiday).

| 2 BYTE | |
|---|---|
| 010WWXXX XXYYYYYY | 010 check bit, bit Y Minutes, bit X Hours, bit W Tariff |

**WARNING**: to read the RTC RAM, the bytes are to be addressed two by two starting only from the even addresses.

**NOTE**: when the time is updated by serial commands, the AM-PM coding is not allowed

***Example 24*: RTC data read command**

*4-word read command frame (8 byte):*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01 | 04h | 40h | 00h | 00h | 04h | E4h | 09h |

*read command answer frame (13 byte):*

| | | | **Sec.** | **Min.** | **Hour** | **Week day .** | **Month day** | **Month** | **Year** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | 04h | 08h | **12h** | **08h** | **11h** | **01h** | **08h** | **0Ah** | **01h** | **00h** | 30h | 6Bh |

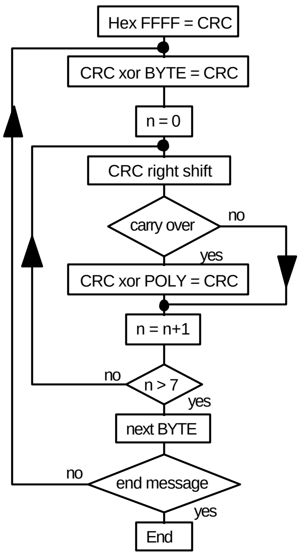
| | |
|---|---|
| Seconds: | 12h = 18 |
| Minutes: | 08h = 8 |
| Hour: | 11h = 17 |
| Day of the week | 01h = 1 (Monday) |
| Day of the month: | 08h = 8 |
| Month: | 0Ah = 10 |
| Year: | 0001h= 1 (2001) |

## 4 CRC CALCULATION ALGORITHM

CRC is calculated according to the relevant flow diagram (see below). An explanatory example will follow.

***Example 25*: calculation of CRC starting from frame 0207h**

| | | | | | |
|---|---|---|---|---|---|
| CRC Inizialization | 1111 | 1111 | 1111 | 1111 | |
| Load first byte | | | 0000 | 0010 | |
| Execute XOR with the first byte of the frame | 1111 | 1111 | 1111 | 1101 | |
| Execute 1st right Shift | 0111 | 1111 | 1111 | 1110 | **1** |
| Carry = 1 , load polynomial | 1010 | 0000 | 0000 | 0001 | |
| Execute XOR with the polynomial | 1101 | 1111 | 1111 | 1111 | |
| Execute 2nd right Shift | 0110 | 1111 | 1111 | 1111 | **1** |
| Carry = 1 , load polynomial | 1010 | 0000 | 0000 | 0001 | |
| Execute XOR with the polynomial | 1100 | 1111 | 1111 | 1110 | |
| Execute 3rd right Shift | 0110 | 0111 | 1111 | 1111 | **0** |
| Execute 4th right Shift | 0011 | 0011 | 1111 | 1111 | **1** |
| Carry = 1 , load polynomial | 1010 | 0000 | 0000 | 0001 | |
| Execute XOR with the polynomial | 1001 | 0011 | 1111 | 1110 | |
| Execute 5th right Shift | 0100 | 1001 | 1111 | 1111 | **0** |
| Execute 6th right Shift | 0010 | 0100 | 1111 | 1111 | **1** |
| Carry = 1 , load polynomial | 1010 | 0000 | 0000 | 0001 | |
| Execute XOR with the polynomial | 1000 | 0100 | 1111 | 1110 | |
| Execute 7th right Shift | 0100 | 0010 | 0111 | 1111 | **0** |
| Execute 8th right Shift | 0010 | 0001 | 0011 | 1111 | **1** |
| Carry = 1 , load polynomial | 1010 | 0000 | 0000 | 0001 | |
| Execute XOR with the polynomial | 1000 | 0001 | 0011 | 1110 | |
| | | | | | |
| Load the second byte of the frame | | | 0000 | 0111 | |
| Execute XOR with the second byte of the frame | 1000 | 0001 | 0011 | 1001 | |
| Execute 1st right Shift | 0100 | 0000 | 1001 | 1100 | **1** |
| Carry = 1 , load polynomial | 1010 | 0000 | 0000 | 0001 | |
| Execute XOR with the polynomial | 1110 | 0000 | 1001 | 1101 | |
| Execute 2nd right Shift | 0111 | 0000 | 0100 | 1110 | **1** |
| Carry = 1 , load polynomial | 1010 | 0000 | 0000 | 0001 | |
| Execute XOR with the polynomial | 1101 | 0000 | 0100 | 1111 | |
| Execute 3rd right Shift | 0110 | 1000 | 0010 | 0111 | **1** |
| Carry = 1 , load polynomial | 1010 | 0000 | 0000 | 0001 | |
| Execute XOR with the polynomial | 1100 | 1000 | 0010 | 0110 | |
| Execute 4th right Shift | 0110 | 0100 | 0001 | 0011 | **0** |
| Execute 5° right Shift | 0011 | 0010 | 0000 | 1001 | **1** |
| Carry = 1 , load polynomial | 1010 | 0000 | 0000 | 0001 | |
| Execute XOR with the polynomial | 1001 | 0010 | 0000 | 1000 | |
| Execute 6th right Shift | 0100 | 1001 | 0000 | 0100 | **0** |
| Execute 7th right Shift | 0010 | 0100 | 1000 | 0010 | **0** |
| Execute 8th right Shift | 0001 | 0010 | 0100 | 0001 | **0** |
| | | | | | |
| **CRC Result** | **0001** | **0010** | **0100** | **0001** | |
| | **12h** | | **41h** | | |




POLY = crc calculation polynominal: A001h

**NOTE:** the byte 41h is sent first (even if it's the LSB), then byte 12h is sent.

# 5 FLASH MEMORY MAP

The Flash Memory is composed by 4095 pages, shared in three different blocks containing different kind of information:

- from page 0000 to page 3993: logged data
- from page 3394 to page 3395: telephone numbers and SMS messages
- from page 3396 to page 4095: load profile data

Refer to paragraph 1.3 for the Flash memory read command.

**Configuration map in case of 2-variable logging (pages from 00 to 3993)**

Page 000h

| ADD. | BYTE | VARIABLE | MAX | MIN | DEFAULT | BIT CHECK |
|---|---|---|---|---|---|---|
| 000 | | Sample 1 | | | | |
| 00C | | Sample 2 | | | | |
| 018 | | Sample 3 | | | | |
| | | | | | | |
| | | | | | | |
| 204 | | Sample 44 | | | | |

Page 001h

| ADD. | BYTE | VARIABLE | MAX | MIN | DEFAULT | BIT CHECK |
|---|---|---|---|---|---|---|
| 000 | | Sample 45 | | | | |
| 00C | | Sample 46 | | | | |
| 018 | | Sample 47 | | | | |
| | | | | | | |
| | | | | | | |
| 204 | | Sample 88 | | | | |

**NOTE**: the quantity of samples logged in the Flash memory pages up to 3994 is stored as integer in the RAM at address 0210h.
Refer to paragraph 5.1 for details.

**Configuration map for telephone numbers and SMS messages (pages 3994 and 3995)**
Refer to paragraph 5.2.

**Configuration map for load profiles (pages from 3996 to 4095)**

Page 000h

| ADD. | BYTE | VARIABLE | MAX | MIN | DEFAULT | BIT CHECK |
|---|---|---|---|---|---|---|
| 000 | | Sample 1 | | | | |
| 006 | | Sample 2 | | | | |
| 00C | | Sample 3 | | | | |
| | | | | | | |
| | | | | | | |
| 20A | | Sample 88 | | | | |

**NOTE**: the quantity of samples logged in the Flash memory pages from 3996 to 4095 is stored as integer in the RAM at address 09ECh.
Refer to paragraph 5.1 for details.

### 5.1 DATA LOGGING SAMPLE FORMAT

Every sample includes the following information:

- Number of logged variable (1 <= n <= 8)
- Type of variable 1
- Value of variable 1
- …
- Type of variable n
- Value of variable n
- Hour and date

The sample is coded in a number of bytes which is depending on the number of logged variables (n) according to the following formula: byte=6 + nx3. In the next example the coding of a sample is explained in detail.

***Example 25*: data logging sample coding**

In the following example every sample includes 2 variables.

| Byte | ADDRESS | BYTE | VARIABLE | VALUE | MEANING |
|---|---|---|---|---|---|
| 1 | 000 | 1 | N° of logged variables | 02h | 2 variables |
| 2 | 001 | 1 | Type of variable 1 | 00h | VL1-N |
| 3 | 002 | 1 | Value (MSB) | 01h | |
| 4 | 003 | 1 | Value (LSB) | 7Eh | 017Eh = 382 V |
| 5 | 004 | 1 | Type of variable 2 | 0Fh | WΣ |
| 6 | 005 | 1 | Value (MSB) | 09h | |
| 7 | 006 | 1 | Value (LSB) | 65h | 15.0 W |
| 8,9 | 007 | 2 | Year, month, day (see note) | 0593h | 19 December 2002 |
| 10 | 009 | 1 | Hour | 0Dh | 13 |
| 11 | 00A | 1 | Minute | 01h | 01 |
| 12 | 00B | 1 | Second | 00h | 00 |
| 13 | 00C | 1 | N° of logged variables | 02h | 2 variables |
| 14 | 00D | 1 | Type of variable 1 | 00h | VL1-N |
| 15 | 00E | 1 | Value (MSB) | 01h | |
| 16 | 00F | 1 | Value (LSB) | 7F | 017Fh = 383 V |
| 17 | 010 | 1 | Type of variable 2 | 0Fh | WΣ |
| 18 | 011 | 1 | Value (MSB) | 09h | |
| 19 | 012 | 1 | Value (LSB) | 75h | 15.1 W |
| 20,21 | 013 | 2 | Year, month, day (see note) | 0593h | 19 December 2002 |
| 22 | 015 | 1 | Hour | 0Dh | 13 |
| 23 | 016 | 1 | Minute | 02h | 02 |
| 24 | 017 | 1 | Second | 00h | 00 |
| … | … | … | … | … | … |
| | | | | | |

**NOTE**: the year-month-day format is the following:

| | Year | Month | Day |
|---|---|---|---|
| Byte8 + Byte9 | XXXXXXX | XXXX | XXXXX |

Example: 0593h=0000 0101 1001 0011
Year = 0000 010 = 2 ➔ 2002
Month = 1 100 = 12 ➔ December
Day = 1 0011 = 19

**NOTE**: the format of the power variables in EEPROM and FLASH memory is the following:

| Power variable format | |
|---|---|
| XXXXXXXX XXXXYYYY | XXXXXXXXXXXX -→ MANTISSA YYYY -→ EXPONENT |

The mantissa is lower than 2000 and the exponent lower than 16.
The YYYY value corresponds to the value on the "INF" table, with which it is possible to find the engineering unit to be associated to the XXXXXXXXXXXX mantissa.

Example:
Power = 0965h
096h = 150
5h ➔ 111.1
Power = 15.0 W

## 5.2 PAGES 3994 AND 3995 MAP (SMS MESSAGES)

**PAGE 3994**

| BYTE | ADDRESS | BYTE | VARIABLE | |
|---|---|---|---|---|
| 1 | 000 | 1 | The first 100-byte area contain the text of | |
| 2 | 001 | 1 | the SMS message associated to the activation | |
| 3 | 002 | 1 | of alarm 1 | |
| 4 | 003 | 1 | | |
| | | | | |
| | | | | |
| 100 | 099 | 1 | | |
| 101 | 100 | 1 | The second 100-byte area contain the text of | |
| 102 | 101 | 1 | the SMS message associated to the activation | |
| 103 | 102 | 1 | of alarm 2 | |
| 104 | 103 | 1 | | |
| | | | | |
| | | | | |
| 200 | 199 | 1 | | |
| 201 | 200 | 1 | The third 100-byte area contain the text of | |
| 202 | 201 | 1 | the SMS message associated to the activation | |
| 203 | 202 | 1 | of alarm 3 | |
| 204 | 203 | 1 | | |
| | | | | |
| | | | | |
| 300 | 299 | 1 | | |
| 301 | 300 | 1 | The fourth 100-byte area contain the text of | |
| 302 | 301 | 1 | the SMS message associated to the activation | |
| 303 | 302 | 1 | of alarm 4 | |
| 304 | 303 | 1 | | |
| | | | | |
| | | | | |
| 400 | 399 | 1 | | |
| 401 | 400 | 1 | Area dedicated to the 5 GSM phone numbers. | |
| 402 | 401 | 1 | Every 16-byte area contain one of the five | |
| 403 | 402 | 1 | GSM phone number | |
| | | | | |
| 480 | | | | |

**NOTE**: the number must begin with the international prefix without 00 or + (for example 39xxxxxxxxxx for Italy).

**NOTE**: messages and phone numbers must be stored using only the first 127 ASCII characters. The last character of each message must be the CR (13 decimal).

**PAGE 3995**

| Byte | ADDRESS | BYTE | VARIABLE | |
|---|---|---|---|---|
| 1 | 000 | 1 | The first 100-byte area contain the text of | |
| 2 | 001 | 1 | the SMS message associated to the | |
| 3 | 002 | 1 | deactivation of alarm 1 | |
| 4 | 003 | 1 | | |
| | | | | |
| | | | | |
| 100 | 099 | 1 | | |
| 101 | 100 | 1 | The second 100-byte area contain the text of | |
| 102 | 101 | 1 | the SMS message associated to the | |
| 103 | 102 | 1 | deactivation of alarm 2 | |
| 104 | 103 | 1 | | |
| | | | | |
| | | | | |
| 200 | 199 | 1 | | |
| 201 | 200 | 1 | The third 100-byte area contain the text of | |
| 202 | 201 | 1 | the SMS message associated to the | |
| 203 | 202 | 1 | deactivation of alarm 3 | |
| 204 | 203 | 1 | | |
| | | | | |
| | | | | |
| 300 | 299 | 1 | | |
| 301 | 300 | 1 | The fourth 100-byte area contain the text of | |
| 302 | 301 | 1 | the SMS message associated to the | |
| 303 | 302 | 1 | deactivation of alarm 4 | |
| 304 | 303 | 1 | | |
| | | | | |
| | | | | |
| 400 | 399 | 1 | | |
| 401 | 400 | 1 | Area dedicated to the 5 fixed phone numbers. | |
| 402 | 401 | 1 | Every 16-byte area contain one of the five | |
| 403 | 402 | 1 | fixed phone number | |
| 404 | 403 | 1 | | |
| | | | | |
| 480 | | | | |

**NOTE**: the number must begin with the international prefix without 00 or + (for example 39xxxxxxxxxx for Italy).

**NOTE**: messages and phone numbers must be stored using only the first 127 ASCII characters. The last character of each message must be the CR (13 decimal).

## 5.3 LOAD PROFILE SAMPLE FORMAT

Every sample includes the following information:

- Wdmd value
- Hour
- Date

The sample is coded in a 6 bytes.

**Map of one of the 100 pages dedicated to the load profile**

| Byte | ADDRESS | BYTE | VARIABLE | |
|---|---|---|---|---|
| Sample 1 | | | | |
| 1 | 000 | 1 | Value (MSB) | |
| 2 | 001 | 1 | Value (LSB) | |
| 3,4 | 002 | 2 | Second, Month, Day | |
| 5,6 | 004 | 2 | Tariff, Hour, Minute | |
| Sample 2 | | | | |
| 7 | 006 | 1 | Value (MSB) | |
| 8 | 007 | 1 | Value (LSB) | |
| 9,10 | 008 | 2 | Second, Month, Day | |
| 11,12 | 00A | 2 | Tariff, Hour, Minute | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| Sample 88 | | | | |
| 523 | 20A | 1 | Value (MSB) | |
| 524 | 20B | 1 | Value (LSB) | |
| 525,526 | 20C | 2 | Second, Month, Day | |
| 527,528 | 20E | 2 | Tariff, Hour, Minute | |

Load profile sample coding

| | Value1 MSB | Value1 LSB |
|---|---|---|
| Byte1 + Byte2 | XXXXXXXX | XXXXXXXX |

| | Second | Month | Day |
|---|---|---|---|
| Byte3 + Byte4 | XXXXXXX | XXXX | XXXXX |

| | Tariff | Hour | Minute |
|---|---|---|---|
| Byte5 + Byte6 | XXXXXXX | XXXX | XXXXX |

**NOTE**: the format of the power variables in EEPROM and FLASH memory is the following:

| Power variable format | |
|---|---|
| XXXXXXXX XXXXYYYY | XXXXXXXXXXXX -→ MANTISSA YYYY -→ EXPONENT |

The mantissa is lower than 2000 and the exponent lower than 16.
The YYYY value corresponds to the value on the "INF" table, with which it is possible to find the engineering unit to be associated to the XXXXXXXXXXXX mantissa.

## 5.4 FLASH MEMORY RESET

It is only possible to reset the flash memory using a fixed frame, where the address of the instrument, the 80h read command and the value DFh in the sixth byte must be written. Byte nr. 3, 4 and 5 can assume any value.

Flash memory reset request frame (8 byte):

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 80h | XXh | XXh | XXh | DFh | XXh | XXh |

Flash memory reset answer frame (8 byte):

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01h | 80h | XXh | XXh | XXh | DFh | XXh | XXh |

# 6 HARDWARE SPECIFICATIONS

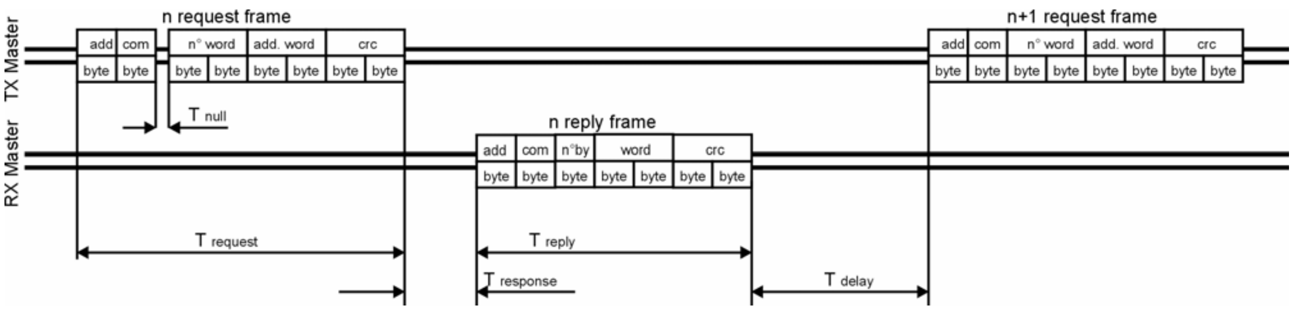
## 6.1 RS485 INTERFACE

| **General technical specifications** | | **Note** |
|---|---|---|
| Baud rate | 1200, 2400, 4800, 9600bps | |
| Data format | 1 start / 8 data / 1 stop bit / no parity<br>1 start / 8 data / 1 stop bit / even parity<br>1 start / 8 data / 1 stop bit / odd parity | |
| Address | 1 to 255 | |
| Broadcast | Yes (address 0 with function 06) | |
| Standard functions | 04: Read function (max 108 words) | |
| | 06: Write function (max 1 word) | |
| Special functions | 80: Read from Flash memory (data-logging) | |
| Answer buffer | 264+5 byte | A |
| Identification code | 16 (10h) | B |
| Synchr. Time-out | 3 chars | C |
| Physical interface | MAX1482 | |
| RX termination | Jumper between Rx+ and T terminals | |
| Available connections | 4-wire (RS422 half duplex interface) | D |
| | 2-wire (RS485 interface) | |

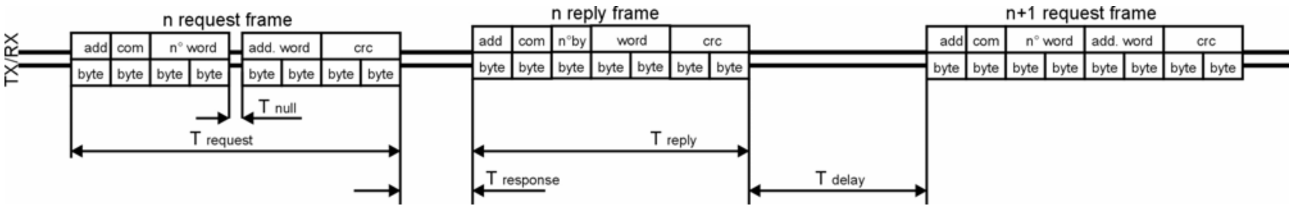
**Note:**

A. With a single request maximum 132 words can be read from WM4.
B. See paragraph 1.4 WM4-96 identification code.
C. It is the time that must elapse without receiving any character before starting the analysis of the received frame.
D. RS422 interface is managed with the same protocol of the RS485 one: in this way only the half-duplex communication is allowed (TX and RX not simultaneous).

| **Timing characteristics for 4-wire communication** | **msec** |
|---|---|
| T response: max answering time | 600ms |
| T response: typical answering time | 100ms |
| T delay1: minimum time for a new query on the same address | 10ms |
| T delay2: minimum time for a new query on a different address | 10ms |
| T null: maximum interruption time on the request frame | 3 char |



| Timing characteristics for 2-wire communication | msec |
|---|---|
| T response: max answering time | 600ms |
| T response: typical answering time | 100ms |
| T delay1: minimum time for a new query on the same address | 10ms |
| T delay2: minimum time for a new query on a different address | 10ms |
| T null: maximum interruption time on the request frame | 3 char |




**APPLICATION NOTES**

1. If the 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 must be considered as not connected, faulty or having a different address. The same consideration is valid in case of CRC errors or incomplete frames.
2. By entering the programming mode (by pressing the “S” key) the communication is interrupted. Any data received during the programming mode are ignored.
3. EEPROM read and write commands must be carried out to manage “static” variables. Use them only during the instrument set-up and not during the normal measuring mode in order to avoid to extend the answer time and to limit the writing in EEPROM (max 100.000).
4. To avoid reflections or couplings between the communication wires it is suggested to terminate the last instrument of the network and of the host. If some problems persist, bias the host transmission. It is advisable to terminate the network also in case of short point to point connections.
5. If the connection is longer than 1200 m a signal amplifier has to be used.
6. To calculate the time required to scan all the instruments of a network, the following formulae are to be used:

$$Trequest = \frac{N^\circ bit}{Baud\_rate} * 8$$

$$Treply = \frac{N^\circ bit}{Baud\_rate} * N^\circ char$$

$$TS = T\_request + T\_response + T\_reply + T\_delay1$$

$$TA = TS * N^\circ request$$

$$TM = (TS + Tdelay2) * N^\circ instruments$$

| | |
|---|---|
| N° bit | 10 (no parity), 11 (even or odd parity) |
| N° char | 5 + number of Words*2 (function 04), 8 (function 06) |
| N° word | Number of words to be read in the same request |
| TS | Reading execution time |
| Tdelay1 | Minimum time for a new query on the same address |
| TA | Instrument data acquisition time |
| TM | Total network scanning time |
| N° instruments | Number of instruments connected in the network |
| Tdelay2 | Minimum time for a new query on a different address |

### 6.2 RS232 INTERFACE

| General technical specifications | | Note |
|---|---|---|
| Baud rate | 2400, 4800, 9600, 38400 bps | |
| Data format | 1start/8 data / 1 stop bit / no parity | |
| Address | Not managed | A |

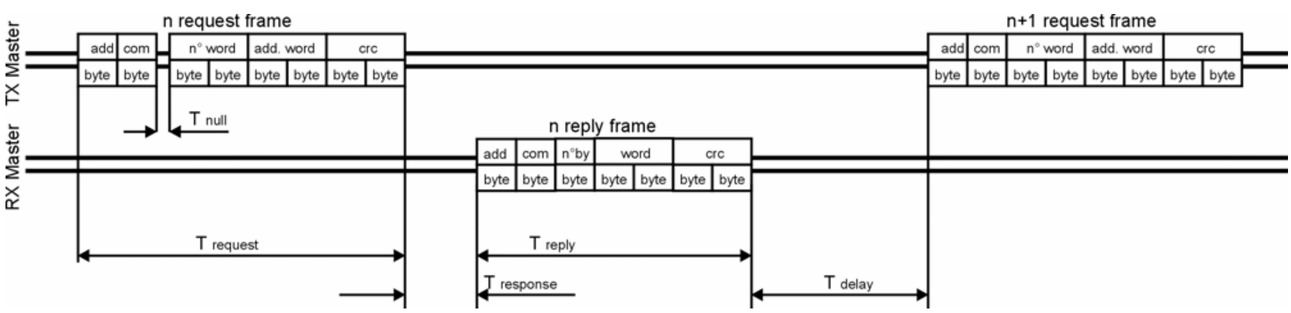
Note:
A. Nevertheless in the address cell a value from 1 to 255 must be.

| 9-pole female RS232 connector | | Note |
|---|---|---|
| Pin 1 | DCD | Used only for modem connection |
| Pin 2 | TX | To be connected to the RX terminal of the PC COM |
| Pin 3 | RX | To be connected to the TX terminal of the PC COM |
| Pin 4 | Not used | |
| Pin 5 | GND | To be connected to the GND terminal of the PC COM |
| Pin 6 | Not used | |
| Pin 7 | Not used | |
| Pin 8 | Not used | |
| Pin 9 | RING | Used only for modem connection |

**Note**: to connect WM4 with a PC use a serial cable with “pin to pin” connections.

| Timing characteristics for RS232 communication | msec |
|---|---|
| T response: max answering time | 600ms |
| T response: typical answering time | 100ms |
| T delay: minimum time for a new query | 10ms |
| T null: maximum interruption time on the request frame | 50msec |

**Note**: T null is independent of the selected baud-rate value



**APPLICATION NOTES**

1. If the 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 must be considered as not connected, faulty or having a different address. The same consideration is valid in case of CRC errors or incomplete frames.
2. By entering the programming mode (by pressing the “S” key) the communication is interrupted. Any data received during the programming mode are ignored.
3. EEPROM read and write commands must be carried out to manage “static” variables. Use them only during the instrument set-up and not during the normal measuring mode in order to avoid to extend the answer time and to limit the writing in EEPROM (max 100.000).
4. Control lines are not managed.

# 7 MODEM MANAGEMENT

## 7.1 INTRODUCTION

WM4-96 can be connected to the fixed telephone network by means of analogue modems or to the mobile telephone network by means of a GSM modem. The different possibilities are listed in the following table:

| **WM4 interfacing capabilities** |
|---|
| WM4 ↔ PC |
| WM4 ↔ Analogue Modem ↔ Fixed phone network ↔ Analogue Modem ↔ PC |
| WM4 ↔ GSM Modem ↔ Mobile phone network ↔ Mobile phone |

## 7.2 VALIDATED MODEM

The following models of modem and relevant accessories have been tested and validated:

- 56K Faxmodem by U.S. Robotics (analogue modem);
- TC35 Terminal by Siemens (GSM modem, code SMTTC35Terminal).
  Power supply for TC35T (code SMTALIM-M20T-TC35T).
  Vehicular antenna SME (code XAT573/2)
  Magnetic antenna SME (code XAT574)

The communication parameters (connection between WM4 and analogue or GSM modem) are the following:

- Baud rate: 9600 bps
- No parity
- 1 stop bit.

## 7.3 CONNECTIONS

To connect WM4 with a PC 9-pole serial cables with "pin to pin" connections are to be used.

| RS232 (PC) | | RS232 (WM4-96) |
|---|---|---|
| 2 | ←======→ | 2 |
| 3 | ←======→ | 3 |
| 5 | ←======→ | 5 |

To connect WM4 with a modem 9-pole Null modem serial cables are to be used.

| RS232 (MODEM) | | RS232 (WM4-96) |
|---|---|---|
| 1, 6 | ←======→ | 4 |
| 2 | ←======→ | 3 |
| 3 | ←======→ | 2 |
| 4 | ←======→ | 1, 6 |
| 5 | ←======→ | 5 |
| 7 | ←======→ | 8 |
| 8 | ←======→ | 7 |

## 7.4 MODEM CONFIGURATION

Both an analogue modem and a GSM one are to be interfaced respecting the following conditions:

- Transparent working mode, 9600 bps, no parity, 1 stop bit.
- Automatic answer after 3 rings (only for analogue modems).
- Carrier Detect (CD) control signal activated.
- Flow control deactivated.

To configure the modem (both analogue and GSM):

- connect the modem to the PC RS232 serial port using a "pin to pin" serial cable;
- launch Hyperterminal software.
- send to the modem the relevant AT commands (please refer to the modem manual).

## 7.5 AT COMMANDS FOR MODEM CONFIGURATION

56K FaxModem by U.S. Robotics:

- ATS0=3
- AT&N6
- AT&U6
- AT&I0
- AT&H0
- AT&K0
- AT&C1
- AT&D0
- AT&Y1
- AT&W1

GSM TC35i Terminal Modem by Siemens:

- ATS0=3
- AT\Q0
- AT&C1
- AT&D0
- AT&S1
- AT+CBST=7,0,1
- AT+CMGF=1
- AT+CMGS=1
- AT&W0

## 7.6 GSM MODEM COMMUNICATION

To enable the communication via GSM modem, WM4 serial port is to be opportunely set (menu serial output/RS232/GSM).

The communication from/to WM4 via GSM modem is performed using the SMS messaging, both in active and passive working mode.

GSM modem is to be equipped with a SIM CARD whose P.I.N. is to be previously disabled (for example Vodafone Omnitel rechargeable Fax/modem Internet Card).

### 7.6.1 Active working mode

WM4 is able to call up to 5 different phone numbers and send an SMS message in case of activation or deactivation of up to 4 alarms.
The connection between WM4 and the GSM modem is to be carried out after storing on the Flash memory the phone number to which the messages are to be sent and the text messages to be associated both to the alarm activation and alarm deactivation (see paragraph 5.2).
The text messages associated to the alarms will be sent only if this function is enabled (see EEPROM addresses 2078h and 207Ah). It is required to specify the number of telephone number towards which the messages are to be sent (maximum 5, see EEPROM address 207Ch). The messages will be sent to every selected phone numbers in the specified order.

### 7.6.2 Passive working mode

WM4 is able to answer to a SMS message, sending the required values of the selected variables (instantaneous variables, data logging variables, energy meters or alarm status).
The connection between WM4 and the GSM modem is to be carried out after storing the 3–digit password value to be used to identify the instrument (see EEPROM address 207Eh).
The instrument can be interrogated by means of a SMS message whose fixed text is to be inferred by the following table.

| Variables to be read | Request SMS text | Answer SMS text |
| --- | --- | --- |
| VL1, VL2. VL3, VLΣ | WM4.**xxx**.VN | WM4.**xxx**: |
| V1-2, V2-3, V3-1, VΣ | WM4.**xxx**.VL | WM4.**xxx**: |
| AL1, AL2, AL3, AΣ | WM4.**xxx**.A | WM4.**xxx**: |
| WL1, WL2, WL3, WΣ | WM4.**xxx**.W | WM4.**xxx**: |
| VAL1, VAL2, VAL3, VAΣ | WM4.**xxx**.VA | WM4.**xxx**: |
| varL1, varL2, varL3, varΣ | WM4.**xxx**.VAR | WM4.**xxx**: |
| PFL1, PFL2, PFL3, PFΣ | WM4.**xxx**.PF | WM4.**xxx**: |
| Wdmd, VAdmd, vardmd, PFdmd | WM4.**xxx**.DMD | WM4.**xxx**: |
| Log1, Log2, Log3, Log4, Log5, Log6, Log7, Log8 | WM4.**xxx**.LOG | WM4.**xxx**: |
| Total kWh+, kvar+ , kWh-, kvar- | WM4.**xxx**.TOTAL | WM4.**xxx**: |
| Winter kWh+, kvar+ , kWh-, kvar- | WM4.**xxx**.ENGWn | WM4.**xxx**: |
| Summer kWh+, kvar+ , kWh-, kvar- | WM4.**xxx**.ENGSn | WM4.**xxx**: |
| Holiday KWh+, Kvar+ , KWh-, Kvar- | WM4.**xxx**.ENGHn | WM4.**xxx**: |
| Status Alarm1, Alarm2, ….. | WM4.**xxx**.ALARM | WM4.**xxx**: |

Where:
“**xxx**” is the SMS password
**"ENGW1" .... "ENGH4"** are the partial energy meter of each season: for example ENGW1 means “energy meter winter tariff1”; ENGH4 means “energy meter holiday tariff4”.
“**Log1**” … “**Log8**” are the last logged variables stored on the Flash memory.

***Example 26*: dmd variables request (SMS password 123)**

Request SMS text: WM4.123.DMD
Answer SMS text: WM4.123.DMD.12.45KW dmd, 16.04KVA dmd, 3.45Kvar dmd, 0.79PF

### 7.7 ANALOGUE MODEM COMMUNICATION.

To enable the communication via analogue modem, WM4 serial port is to be opportunely set (menu serial output/RS232/MODEM).

#### 7.7.1 Passive working mode

WM4 automatically answers the remote PC after 3 rings. The communication will start using the Modbus protocol as the WM4 is directly connected to the PC.

Note: the modem connected to WM4 is to be configured with automatic answer after 3 rings (see the relevant AT commands on the modem manual).

#### 7.7.2 Active working mode

WM4 is able to call a remote PC and communicate the activation or deactivation of up to 4 alarms.

The connection between WM4 and the analogue modem is to be carried out after storing on the Flash memory the phone number to which the message is to be sent and the text messages to be associated both to the alarm activation and alarm deactivation (see paragraph 5.2).

It is required to specify the number of telephone number towards which WM4 has to try to send the message (maximum 5, see EEPROM address 207Ch). The messages will be to the first telephone number. Only if the connection is not possible, WM4 will try with the second number and so on.

WM4 has to send a fixed frame request to the remote PC to establish if the connection is correctly settled. The PC has to answer with another fixed frame to confirm the correct connection and to communicate it is in waiting status.

Every message has to finish with the ASCII code 26 (Ctrl+Z) and has to include less than 100 characters.

**Connection request (from WM4-96 to the remote PC, 7-byte frame)**

| Address | Function | | | | CRC | |
|---|---|---|---|---|---|---|
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 byte | |
| **from 1 to 255** | **04** | **AA** | **55** | **AA** | MSB | LSB |

**Correct connection answer (from the remote PC to WM4-96, 8-byte frame)**

| Address | | | | | | CRC | |
|---|---|---|---|---|---|---|---|
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 2 byte | |
| **from 1 to 255** | **BE** | **AD** | **9C** | **8B** | **7A** | MSB | LSB |