	COMMUNICATION PROTOCOL	PR 115	rev. 0
		11/11/2011	Pagina 1 di 11
ELECTRICITY ENERGY METER		FIRMWARE ≥ 1.3	
CE4DMID01			

CONTENTS

1.0 INTRODUCTION

2.0 DATA MESSAGE DESCRIPTION

- 2.1 Data field description
- 2.2 Data format
- 2.3 Description of CRC calculation
- 2.4 Error management
- 2.5 Timing

3.0 COMMANDS

4.0 VARIABLES

- 4.1 Data addresses

1.0 ABSTRACT

1.0 INTRODUCTION

Data link level

The communication protocol used is MODBUS / JBUS compatible.
Up to 255 different instruments can be managed by the protocol.
Data are transmitted in messages and are checked by mean of a CRC16 WORD
There are no limitations to the number of possible retries done by the master.

Physical level

The physical communication line respects the EIA-RS485 standard in half-duplex modality.
In this case, as only two wires are used, only one instrument at a time can engage the line; this means that there must be a master polling the slave instruments and waiting for the answers.

On the same physical line only 32 instruments can be attached (master included). In order to increase the number of the slave instrument, the necessary repeaters must be used.

The communication parameters are :

speed : programmable
19200, 9600, 4800 Baud
bit n. : 8
stop bit : 1
parity : programmable

2.0 DATA MESSAGE DESCRIPTION

The generic data message is composed as following :

Instrument address	Functional code	Data	CRC word
--------------------	-----------------	------	----------

Two answers are possible :

Answer containing data

Instrument address	Functional code	Data	CRC word
--------------------	-----------------	------	----------

Error answer

Instrument address	Functional code + 0x80	Error code	CRC word
--------------------	---------------------------	------------	----------

2.1 Data field description

Instrument address : instrument identification number in the network

It must be the same for the demand and the answer.

Format : 1 BYTE from 0 to 0xff - 0 is for broadcast messages with no answer (not used)

Functional code : command code

Used functional code :

Format : 1 BYTE

0x03 : reading of consecutive words

0x10 : writing of consecutive words

Data : they can be :

- the address and the number of the required words (in the demand)
- the data (in the answer)

CRC word : it is the result of the calculation made on all the bytes in the message

2.2 Data format

Three types of format are used for the data :

- * BYTE
- * WORD : two BYTES
- * long : two WORDS

Three types of format are used for the data :

- * BYTE
- * WORD : two BYTES
- * long : two WORDS

The base data format is the WORD.

If the required data is in a BYTE format, a WORD with the MSB (Most Significant Byte) set to 0 is anyway transmitted and this BYTE comes before the LSB (Least Significant Byte).

If the required data is in a long format, 2 WORDS are transmitted and the MSW comes before the LSW.

MSB	LSB	MSB	LSB
Most Significant WORD		Least Significant WORD	

Example : 1000 = 0x 03 e8 or
0x 00 00 03 e8 (if long)

MSB	LSB	MSB	LSB
0x00	0x00	0x03	0xe8

All data are positive and the sign indications are readable in other variables.

2.3 Description of CRC calculation

The following is an example of the CRC calculation in C language.

```

unsigned int calc_crc (char *ptbuf, unsigned int num)
/* *****
 *   Descrizione : calculates a data buffer CRC WORD
 *   Input       : ptbuf = pointer to the first byte of the buffer
 *                 num  = number of bytes
 *   Output      : //
 *   Return      :
 **  *****/
{
    unsigned int crc16;
    unsigned int temp;
    unsigned char c, flag;

    crc16 = 0xffff;                               /* init the CRC WORD */
    for (num; num>0; num--) {
        temp = (unsigned int) *ptbuf;             /* temp has the first byte */
        temp &= 0x00ff;                           /* mask the MSB */
        crc16 = crc16 ^ temp;                     /* crc16 XOR with temp */
        for (c=0; c<8; c++) {
            flag = crc16 & 0x01;                 /* LSBit di crc16 is kept */
            crc16 = crc16 >> 1;                 /* LSBit di crc16 is lost */
            if (flag != 0)
                crc16 = crc16 ^ 0x0a001;        /* crc16 XOR with 0x0a001 */
        }
        ptbuf++;                                  /* points the next byte */
    }

    crc16 = (crc16 >> 8) | (crc16 << 8);        /* LSB is exchanged with MSB */

    return (crc16);

} /* calc_crc */

```

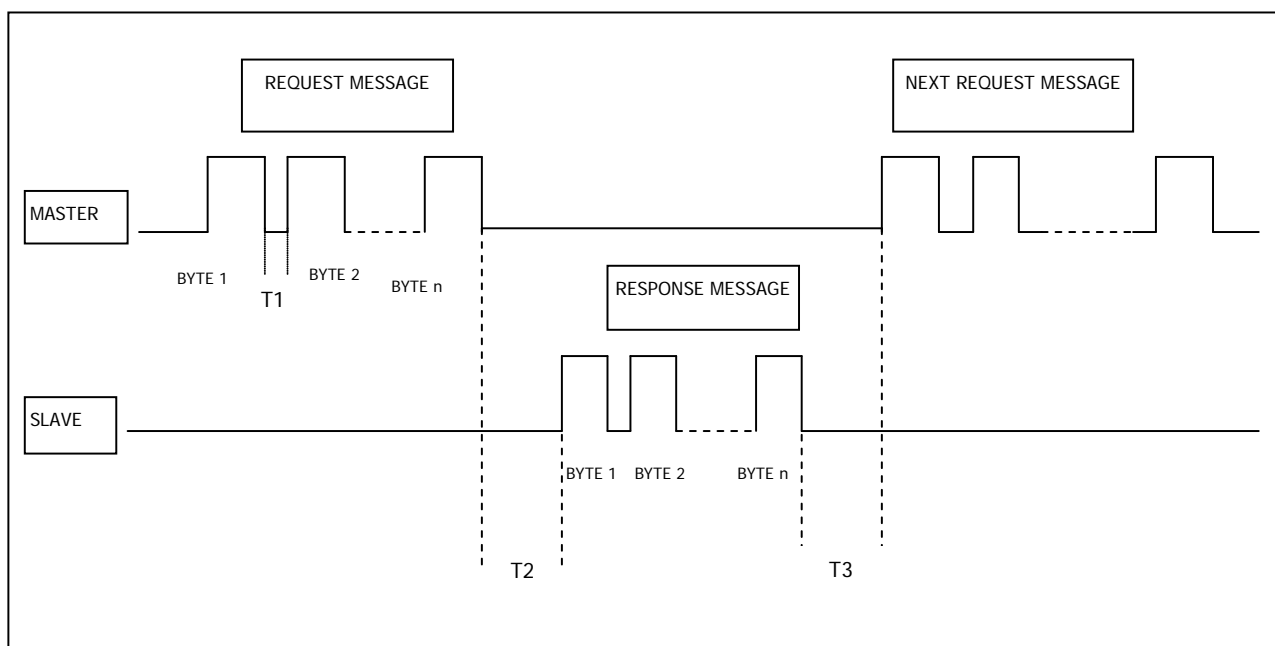
2.4 Error management

If the received message is incorrect (CRC16 is wrong) the polled slave doesn't answer.

If the message is correct but there are errors (wrong functional code or data) so it can't be accepted, the slave answers with an error message.

The error codes are defined in the following part of the document.

2.5 Timing



Values :

T1 (time between characters) = 25 msec (max)

T2 (slave response time) = 100 msec (max)

T3 (delay time) = 25 msec (min)

3.0 COMMANDS

Code 0x03 : reading of one or more consecutive WORDS

Command format :

BYTE	BYTE	MSB	LSB	MSB	LSB	MSB	LSB
Instrument Address	Funct. Code	First WORD address		WORDS number		CRC16	

Answer format (containing data) :

BYTE	BYTE	BYTE	MSB	LSB	MSB	LSB	MSB	LSB
Instrument Address	Funct. Code	BYTES number	WORD 1		WORD N.		CRC16	

The BYTES number must always match the WORDS number (in the demand) * 2.

Answer format (wrong request) :

BYTE	BYTE	BYTE	MSB	LSB
Instrument Address	Funct. Code + 0x80	Error code	CRC16	

Error codes :

- * 0x01 : incorrect functional code
- * 0x02 : wrong first WORD address
- * 0x03 : incorrect data

Code 0x10 : writing of more consecutive WORDS

Command format :

BYTE	BYTE	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB
Instr. address	Funct. Code	First WORD address	WORDS number	BYTE numbers	Word Value		CRC16		

Answer format (containing data) :

BYTE	BYTE	BYTE	MSB	LSB	MSB	LSB	MSB	LSB
Instrument Address	Funct. Code	BYTES number	First WORD address		00	00	CRC16	

The BYTES number must always match the WORDS number (in the demand) * 2.

Answer format (wrong request) :

BYTE	BYTE	BYTE	MSB	LSB
Instrument Address	Funct. Code + 0x80	Error code	CRC16	

Error codes :

- * 0x01 : wrong functional code
- * 0x02 : wrong first WORD address
- * 0x03 : wrong data

4.0 VARIABLES

Variables or groups of variables may be required up to 100 BYTES.

Address	Byte n.	Description	Unit
0x301	Long	Phase 1 : phase voltage	mV
0x305	Long	Phase 2 : phase voltage	mV
0x309	Long	Phase 3 : phase voltage	mV
0x30d	Long	Phase 1 : current	mA
0x311	Long	Phase 2 : current	mA
0x315	Long	Phase 3 : current	mA
0x319	Long	3-phase : active power	(3)
0x31d	Long	3-phase : reactive power	(3)
0x321	Long	3-phase : apparent power	(3)
0x325	Long	3-phase : indirect positive active energy	(4)
0x329	Long	Chained voltage : L1-L2	mV
0x32d	Long	Chained voltage : L2-L3	mV
0x331	Long	Chained voltage : L3-L1	mV
0x335	Long	3-phase : direct positive active energy	(4)
0x339	WORD	Frequency	Hz/10
0x33b	WORD	0	-
0x33d	BYTE	3-phase : power factor	1/100
0x33f	BYTE	3-phase : sector of power factor (cap or ind)	(1)
0x340	BYTE	Reserved	-
0x341	WORD	CRC of the software	-
0x343	Long	3-phase : direct positive reactive energy	(4)
0x347	BYTE	3-phase : sign of active power	(5)
0x348	Long	Operating time counter	sec.
0x34c	BYTE	3-phase : sign of reactive power	(5)
0x34d	BYTE	Reserved	-
0x34e	BYTE	0	
0x34f	BYTE	0	
0x350	Long	3-phase : average power	(3)
0x354	Long	3-phase : peak maximum demand	(3)
0x358	BYTE	Time counter for average power	minutes
0x359	Long	Neutral current	mA
0x35d	Long	Phase 1 : active power	(3)
0x361	Long	Phase 2 : active power	(3)
0x365	Long	Phase 3 : active power	(3)
0x369	BYTE	Phase 1 : sign of active power	(5)
0x36a	BYTE	Phase 2 : sign of active power	(5)
0x36b	BYTE	Phase 3 : sign of active power	(5)
0x36c	Long	Phase 1 : reactive power	(3)
0x370	Long	Phase 2 : reactive power	(3)
0x374	Long	Phase 3 : reactive power	(3)
0x378	BYTE	Phase 1 : sign of reactive power	(5)
0x379	BYTE	Phase 2 : sign of reactive power	(5)
0x37a	BYTE	Phase 3 : sign of reactive power	(5)
0x0c8	BYTE	Reset - bit to bit defined	(6)
0x100	WORD	Current transformer ratio (KTA)	integer
0x102	WORD	Voltage transformer ratio (KTV)	*10 always
0x300	BYTE	Device identifier	0x11

A second address table is implemented in the software and the user may decide freely which use.

Address	Length	Description	Unit
0x1000	Long	Phase 1 : phase voltage	mV
0x1002	Long	Phase 2 : phase voltage	mV
0x1004	Long	Phase 3 : phase voltage	mV
0x1006	Long	Phase 1 : current	mA
0x1008	Long	Phase 2 : current	mA
0x100a	Long	Phase 3 : current	mA
0x100c	Long	Neutral current	mA
0x100e	Long	Chained voltage : L1-L2	mV
0x1010	Long	Chained voltage : L2-L3	mV
0x1012	Long	Chained voltage : L3-L1	mV
0x1014	Long	3-phase : active power	(3)
0x1016	Long	3-phase : reactive power	(3)
0x1018	Long	3-phase : apparent power	(3)
0x101a	WORD	3-phase : sign of active power	(5)
0x101b	WORD	3-phase : sign of reactive power	(5)
0x101c	Long	3-phase : indirect positive active energy	(4) (*)
0x101e	Long	3-phase : direct positive reactive energy	(4)
0x1020	Long	3-phase : direct positive active energy	(4)
0x1022	Long	Operating time counter	sec.
0x1024	WORD	3-phase : power factor	1/100
0x1025	WORD	3-phase : sector of power factor (cap or ind)	(1)
0x1026	WORD	Frequency	Hz/10
0x1027	Long	3-phase : average power	(3)
0x1029	Long	3-phase : peak maximum demand	(3)
0x102b	WORD	Time counter for average power	minutes
0x102c	Long	Phase 1 : active power	(3)
0x102e	Long	Phase 2 : active power	(3)
0x1030	Long	Phase 3 : active power	(3)
0x1032	WORD	Phase 1 : sign of active power	(5)
0x1033	WORD	Phase 2 : sign of active power	(5)
0x1034	WORD	Phase 3 : sign of active power	(5)
0x1035	Long	Phase 1 : reactive power	(3)
0x1037	Long	Phase 2 : reactive power	(3)
0x1039	Long	Phase 3 : reactive power	(3)
0x103b	WORD	Phase 1 : sign of reactive power	(5)
0x103c	WORD	Phase 2 : sign of reactive power	(5)
0x103d	WORD	Phase 3 : sign of reactive power	(5)
0x1200	WORD	Current transformer ratio (KTA)	integer
0x1201	WORD	Voltage transformer ratio (KTV)	*10 always
0x1206	WORD	Device identifier	0x11

(1) -----

0 : PF = 0 or 1
 1 : ind
 2 : cap

(3) -----

W, var, VA / 100 if KTA*KTV < 6000
 W, var, VA if KTA*KTV >= 6000

(4) -----

For indirect positive active energy :

Format : xxxxxx.yy kWh always

(*) Indirect energy = metrological energy at terminal side without taking in account the transformer ratios.

Otherwise:

Transformer ratio	Measurement unit	Display Format	Protocol Format
$1 \leq KTA \cdot KTV < 10$	Wh(varh) * 10	xxxxxx.yy k	xxxxxyy
$10 \leq KTA \cdot KTV < 100$	Wh(varh) * 100	xxxxxxx.y k	xxxxxxy
$100 \leq KTA \cdot KTV < 1000$	kWh(kvarh)	Xxxxxxxxx k	xxxxxxxx
$1000 \leq KTA \cdot KTV < 10000$	kWh(kvarh) * 10	xxxxxx.yy M	xxxxxyy
$10000 \leq KTA \cdot KTV < 100000$	kWh(kvarh) * 100	xxxxxxx.y M	xxxxxxy
$100000 \leq KTA \cdot KTV < 1000000$	kWh(kvarh) * 1000	Xxxxxxxxx M	xxxxxxxx

(5) -----

0 : positive
 1 : negative

(6) -----

0x08 : operating time counter reset
 0x010 : peak maximum demand reset

Example 1

Reading of 4 WORDS (8 BYTES – 2 variables) starting from the address 0x100c :

Request :

BYTE	BYTE	MSB LSB	MSB LSB	MSB LSB
Device address	F. code	1 st WORD address	WORDS number	CRC16
0x01	0x03	0x10 0x1c	0x00 0x04	0x81 0x0f

Answer :

BYTE	BYTE	BYTE	MSB LSB	MSB LSB	MSB LSB	MSB LSB	MSB LSB
		BYTES number	WORD 1	WORD 2	WORD 3	WORD 4	CRC16
0x01	0x03	0x08	0x00 0x00	0x64 0x8c	0x00 0x00	0x35 0x54	0x9a 0x83

In the above case, the information is :

WORD 1 ,WORD 2 : Total indirect active energy 0x0000648C = 25740

WORD 3 ,WORD 4 : Total direct reactive energy 0x00003554 = 13652

Example 2

Writing of 1 WORD at address 0xc8 (reset of partial active energy) :

Command :

BYTE	BYTE	MSB LSB	MSB LSB		MSB LSB	MSB LSB
Device address	F. code	1 st WORD address	WORDS number	BYTES number	WORD	CRC16
0x01	0x10	0x00 0xc8	0x00 0x01	0x02	0x00 0x10	0x72 0xE4