# Conto D1
# MODBUS COMMUNICATION PROTOCOL

## CONTENTS

## 1.0 ABSTRACT

**Physical level**

The electrical communication line complies with 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 which polls the slave instruments so the demand and the request are alternated.

On the same 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 :

Baud rate : programmable (device dependant)
bit n. : 8
stop bit : 1
parity : none

**Data link level**

The data are transmitted in a packet form (message) and are checked by a word (CRC).
See the description of the data packet in the next paragraphs for more details.

**Application level**

The communication protocol used is ModBus RTU compatible.
Up to 247 different instruments can be managed by the protocol.
There are no limitations to the number of possible retries done by the master.
A delay between the response from the slave and the next command could be necessary and it is specified for each device (timing).

## 2.0 DATA MESSAGE DESCRIPTION

The generic data message is composed as following :

| Device address | Functional code | Data | CRC word |
|---|---|---|---|

Two answers are possible :

Answer containing data

| Device address | Functional code | Data | CRC word |
|---|---|---|---|

Error answer

| Device address | Functional code + 0x80 | Error code | CRC word |
|---|---|---|---|

## 2.1 Parameters description

Device address : device 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

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 of the required words (in the demand)
- the data (in the answer)

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

## 2.2 Data format

The following types of format are used for the data values :

* U_WORD : one WORD - unsigned
* S_WORD : one WORD - signed

## 2.3 Description of CRC calculation

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

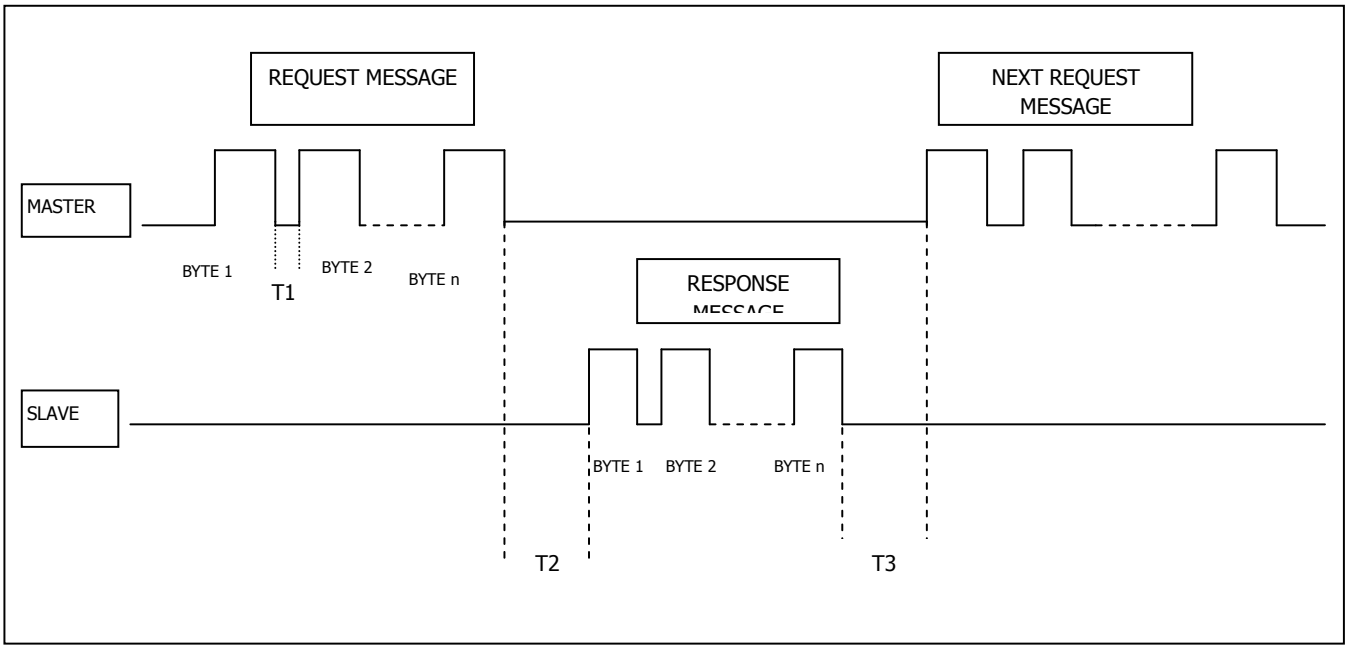
```
unsigned int calc_crc (char *ptbuf, unsigned int num)
/*     ************************************************************
 *     Descrizione : calculates a data buffer CRC WORD
 *     Input       :       ptbuf = pointer to the first byte of the buffer
 *                  num    = number of bytes
 *     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 mantained */
              crc16 = crc16 >> 1;              /* Lsbit di crc16 is lost */
              if (flag != 0)
                   crc16 = crc16 ^ 0x0a001;    /* crc16 XOR with 0x0a001 */
        }
        ptbuf++;                               /* pointer to 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) it can't be accepted, so the slave answers with an error message.

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

## 2.5 Timing




| TIME | DESCRIPTION | Min & Max VALUES |
|---|---|---|
| T1 | **Time between characters.** If this time exceeds the max. time allowed, the message is not considered by device. | Max < 20 ms. |
| T2 | **2.6 Slave response time** Minimum and maximum response time of device to the Master request. | Min = 20 ms. Max = 150ms. |
| T3 | Time before a new message request from the Master | Min = 20 ms. |

## 3.0 COMMANDS

**Code 0x03 : reading of one or more consecutive WORDS**

Command format :

| BYTE | BYTE | MSB LSB | MSB LSB | | |
|---|---|---|---|---|---|
| Device address | Funct. Code | First WORD address | WORDS number | CRC16 | |

Answer format (containing data) :

| BYTE | BYTE | BYTE | MSB LSB | MSB LSB | | |
|---|---|---|---|---|---|---|
| Device 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 (the demand was wrong) :

| BYTE | BYTE | BYTE | | |
|---|---|---|---|---|
| Device 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 | BYTE | MSB LSB | MSB LSB | | |
|---|---|---|---|---|---|---|---|---|
| Device address | Funct. Code | First WORD address | WORDS number | BYTE numbers | Word Value | | CRC16 | |

Answer format (containing data) :

| BYTE | BYTE | BYTE | MSB | LSB | MSB | LSB | | |
|---|---|---|---|---|---|---|---|---|
| Device 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 (the demand was wrong) :

| BYTE | BYTE | BYTE | | |
|---|---|---|---|---|
| Device address | Funct. Code + 0x80 | Error code | CRC16 | |

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

## 4.0 VARIABLES

Variables or groups of variables may be required up to 14 BYTES

| Address | Format | Description | Unit | Type |
|---|---|---|---|---|
| 0x00 | **U_WORD** | Voltage | xxx,x V | Read |
| 0x01 | **U_WORD** | Current | xxx,x A | Read |
| 0x02 | **U_WORD** | Frequency | Xx,xx Hz | Read |
| 0x03 | **U_WORD** | Active Power | Xxxx W | Read |
| 0x04 | **U_WORD** | Reactive Power | Xxxx var | Read |
| 0x05 | **U_WORD** | Apparent Power | Xxxx VA | Read |
| 0x06 | **U_WORD** | Power Factor | - | - |
| 0x07 | **U_DWORD** | Total Active Energy | Xxxxxx wh | Read |
| 0x11 | **U_DWORD** | Total Reactive Energy | Xxxxxx varh | Read |
| 0x2A | **U_WORD** | Baud rate | - | Read/Write |
| 0x2B | **U_WORD** | Node Address | - | Read/Write |

*WRITE ADDRESS TABLE*

| Address | Format | Description | 2.7 Value |
|---|---|---|---|
| 0x2A | **U_WORD** | Baud rate | 1=1200/2=2400/3=4800/4=9600 |
| 0x2B | **U_WORD** | Node address | 1 .. 247 |