# Conto D4-Pt / Conto 72-Pt / Conto 96-Pt

## CONTENTS

## 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

## 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 :

<u>Answer containing data</u>

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

<u>Error answer</u>

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

## 2.1 Data field description

<u>Instrument address</u> : 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)

<u>Functional code</u> : command code
Used functional code :
Format : 1 BYTE
0x03 : reading of consecutive words
0x10 : writing of consecutive words

<u>Data</u> : they can be :
- the address and the number of the required words (in the demand)
- the data (in the answer)

<u>CRC word</u> : 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
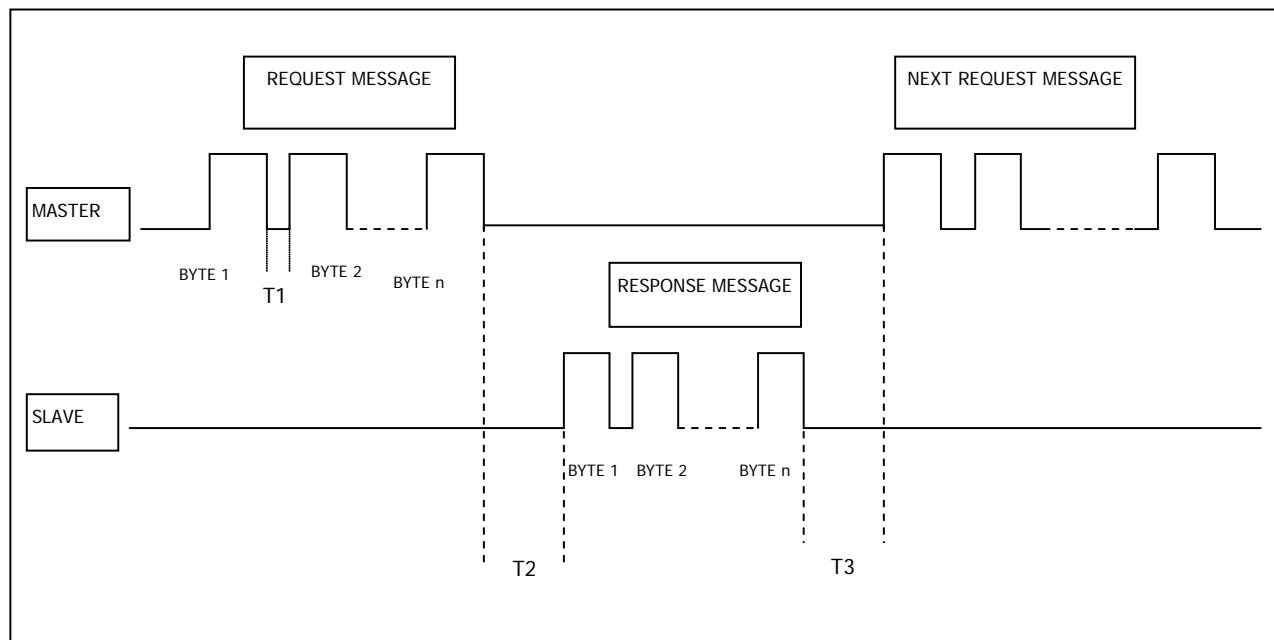


Values :

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 | BYTE | 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

### 4.1  Data addresses

Both variables and groups of variables can be required.
All the variables with consecutive addresses can be required within one request.
The following is the table with the addresses and the meaning of the variables.

| Address | | Read/Write | Length | Description |
|---|---|---|---|---|
| **HEX** | **DEC** | | | |
| | | | | **Energy** |
| 0x325 | 805 | R | Long | 3-phase : Total positive active energy |
| 0x329 | 809 | R | Long | 3-phase : Total positive reactive energy |
| 0x32d | 813 | R/W (1) | Long | 3-phase : Partial positive active energy |
| 0x331 | 817 | R/W (1) | Long | 3-phase : Partial positive reactive energy |
| | | | | **Average power** |
| 0x350 | 848 | R | Long | 3-phase : average power |
| 0x354 | 852 | R/W (1) | Long | 3-phase : peak maximum demand |
| 0x358 | 856 | R/W (1) | Long | 3-phase : peak maximum demand 2° tariffs (where available) |
| | | | | |
| 0x348 | 840 | R | Long | Operating time counter (where available) |

(1) The only writable value is 0x0000000 in order to reset the stored value.
   Different values won't have effect.

The following table must be used to retrieve all information of the real time measurements.

| 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 | 0 | |
| 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 | (1) |
| 0x1016 | Long | 3-phase : reactive power | (1) |
| 0x1018 | Long | 3-phase : apparent power | (1) |
| 0x101a | WORD | 3-phase : sign of active power | (2) |
| 0x101b | WORD | 3-phase : sign of reactive power | (2) |
| 0x101c | Long | 3-phase : total positive active energy | (3) |
| 0x101e | Long | 3-phase : total positive reactive energy | (3) |
| 0x1020 | Long | 0 | |
| 0x1022 | Long | 0 | |
| 0x1024 | WORD | 3-phase : power factor | 1/100 |
| 0x1025 | WORD | 3-phase : sector of power factor (cap or ind) | (4) |
| 0x1026 | WORD | Frequency | Hz/10 |
| 0x1027 | Long | 3-phase : average power | (1) |
| 0x1029 | Long | 3-phase : peak maximum demand | (1) |
| 0x102b | WORD | Time counter for average power | minutes |
| 0x102c | Long | Phase 1 : active power | (1) |
| 0x102e | Long | Phase 2 : active power | (1) |
| 0x1030 | Long | Phase 3 : active power | (1) |
| 0x1032 | WORD | Phase 1 : sign of active power | (2) |
| 0x1033 | WORD | Phase 2 : sign of active power | (2) |
| 0x1034 | WORD | Phase 3 : sign of active power | (2) |
| 0x1035 | Long | Phase 1 : reactive power | (1) |
| 0x1037 | Long | Phase 2 : reactive power | (1) |
| 0x1039 | Long | Phase 3 : reactive power | (1) |
| 0x103b | WORD | Phase 1 : sign of reactive power | (2) |
| 0x103c | WORD | Phase 2 : sign of reactive power | (2) |
| 0x103d | WORD | Phase 3 : sign of reactive power | (2) |
| 0x103e | Long | 3-phase : partial/second tariff positive active energy | (3) |
| 0x1040 | Long | 3-phase : partial/second tariff positive reactive energy | (3) |
| 0x1042 | Long | 3-phase : second tariff peak maximum demand | (1) |
| 0x1044 | Long | 0 | |
| 0x1046 | Long | 0 | |

| 0x0c8 | BYTE | Reset – bit to bit defined | (5) |
|-------|------|----------------------------|-----|
| 0x100 | WORD | Current transformer ratio  (KTA) | integer |
| 0x102 | WORD | Voltage transformer ratio  (KTV) | *10 <br> E.g. 1.0 => 10 |
| 0x300 | BYTE | Device identifier | 0x71 |

(1) ------------------------------------------------------------------------------

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


(2) ------------------------------------------------------------------------------

    0 : positive
    1 : negative

(3) ------------------------------------------------------------------------------

| Transformer   ratio | Measurement   unit | Display Format | Protocol Format |
|---------------------|--------------------|----------------|-----------------|
| 1 ≤ KTA*KTV < 10 | Wh(varh) * 10 | xxxxxx.yy k | xxxxxxyy |
| 10 ≤ KTA*KTV < 100 | Wh(varh) * 100 | xxxxxxx.y k | xxxxxxxy |
| 100 ≤ KTA*KTV < 1000 | kWh(kvarh) | xxxxxxxx k | xxxxxxxx |
| 1000 ≤ KTA*KTV < 10000 | kWh(kvarh) * 10 | xxxxxx.yy M | xxxxxxyy |
| 10000 ≤ KTA*KTV < 100000 | kWh(kvarh) * 100 | xxxxxxx.y M | xxxxxxxy |
| 100000 ≤ KTA*KTV | kWh(kvarh) * 1000 | xxxxxxxx M | xxxxxxxx |

(4) ------------------------------------------------------------------------------

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


(5) ------------------------------------------------------------------------------
    WRITABLE ONLY

    0x01  : partial active energy
    0x02  : partial reactive energy
    0x08  : operating time counter reset (where available)
    0x10  : peak maximum demand reset

**Example 1**

Reading of 4 WORDS (8 BYTES – 2 variables) starting from the address 0x101c :

**Request :**

| BYTE | BYTE | MSB \| LSB | MSB \| LSB | MSB \| LSB |
|---|---|---|---|---|
| Device address | F. code | 1st 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 positive active energy 0x0000648C = 25740
WORD 3 ,WORD 4 : Total positive reactive energy 0x00003554 = 13652

**Example 2**

Writing of 1 WORD at address 0xc8 (reset of operating time counter) :

**Command :**

| BYTE | BYTE | MSB \| LSB | MSB \| LSB | | MSB \| LSB | MSB \| LSB |
|---|---|---|---|---|---|---|
| Device address | F. code | 1st WORD address | WORDS number | BYTEs number | WORD | CRC16 |
| 0x01 | 0x10 | 0x00 \| 0xc8 | 0x00 \| 0x01 | 0x02 | 0x00 \| 0x08 | 0xb7 \| 0xde |