



**LOVATO ELECTRIC S.P.A.**

24020 GORLE (BERGAMO) ITALIA  
VIA DON E. MAZZA, 12  
TEL. 035 4282111  
TELEFAX (Nazionale): 035 4282200  
TELEFAX (International): +39 035 4282400  
Web [www.LovatoElectric.com](http://www.LovatoElectric.com)  
E-mail [info@LovatoElectric.com](mailto:info@LovatoElectric.com)

13341 GB 0211



## DME CD

Concentratore dati

**PROTOCOLLO DI COMUNICAZIONE  
MODBUS®**



## DME CD

Data concentrator

**MODBUS® COMMUNICATION PROTOCOL**

### PROTOCOLLO MODBUS®

Il concentratore dati DME\_CD supporta i protocolli di comunicazione Modbus RTU®, Modbus ASCII® e Modbus TCP®.

Il concentratore dati DME\_CD supporta i protocolli di comunicazione sui moduli di espansione:

- EXM 10 11 RS 232
- EXM 10 12 RS485
- EXM 10 20 RS 485 + 2 relè
- EXM 10 10 USB
- EXM 10 13 Ethernet

Grazie a questa funzione e' possibile leggere lo stato degli apparecchi e controllare gli stessi tramite il software di controllo remoto dedicato (DMK remote control), software di supervisione standard forniti da terze parti (SCADA) oppure tramite apparecchiature dotate di interfaccia Modbus® quali PLC e terminali intelligenti.

### IMPOSTAZIONE DEI PARAMETRI

Per configurare il protocollo Modbus®, accedere al SETUP MENU e selezionare il menu M06.

Se n = 1 la programmazione è riferita alla RS485 del dispositivo.

Se n = 2 la programmazione è riferita al modulo d'espansione.

### MENU M6 – COMUNICAZIONE SERIALE

PAR	Funzione	Range	Default
P06.n.01	Indirizzo	1 ..245	1
P06.n.02	Velocità RS-232 (baud)	1200 2400 4800 9600 19200 38400	9600 baud
P06.n.03	Formato dati	8 bit Nessuna 8 bit Dispari 8 bit Pari 7 bit Dispari 7 bit Pari	8 bit Nessuna
P06.n.04	Stop bit	1 2	1
P6.n.05	Protocollo	Modbus RTU Modbus ASCII Modbus TCP	Modbus RTU
P06.n.09	Funzione Gateway	OFF	OFF/ON

Per il modulo di espansione EXM 10 13 e EXP 10 13 ( Ethernet ) esistono altri tre parametri.

PAR	Funzione	Range	Default
P07.n.06	Indirizzo IP	000.000.000.000 255.255.255.255	000.000.000.000
P07.n.07	Subnet MASK	000.000.000.000 255.255.255.255	000.000.000.000
P07.n.08	TCP-IP Port	0 - 9999	1001

### MODBUS® PROTOCOL

The data concentrator DME\_CD supports the communication protocols Modbus RTU®, Modbus ASCII® and Modbus TCP®.

The data concentrator DME\_CD supports the communication protocols on the expansion modules:

- EXM 10 11 RS 232
- EXM 10 12 RS485
- EXM 10 20 RS 485 + 2 relè
- EXM 10 10 USB
- EXM 10 13 Ethernet

Using this function it is possible to read the device status and to control the units through the dedicated Remote control software (DMK remote control), third-party supervision software (SCADA) or through other intelligent devices supporting Modbus®, like PLCs.

### PARAMETER SETTING

To configure the Modbus® protocol, enter SETUP MENU and choose the M06 menu:

If n = 1, the programming is related to the RS485 on the device.

If n = 2, the programming is related to the expansion module.

### MENU M7 – SERIAL COMMUNICATION

PAR	Function	Range	Default
P06.n.01	Address	1 ..245	1
P06.n.02	RS-232 Baud Rate	1200 2400 4800 9600 19200 38400	9600 baud
P06.n.03	Data format	8 bit None 8 bit Odd 8 bit Even 7 bit Odd 7 bit Even	8 bit None
P06.n.04	Stop bit	1 2	1
P06.n.05	Protocol	Modbus RTU Modbus ASCII Modbus TCP	Modbus RTU
P06.n.09	Gateway Funtion	OFF	OFF/ON

For expansion module EXM 10 13 and EXP 10 13 (Ethernet), there are other three parameters.

PAR	Function	Range	Default
P07.n.06	IP Address	000.000.000.000 255.255.255.255	000.000.000.000
P07.n.07	Subnet MASK	000.000.000.000 255.255.255.255	000.000.000.000
P07.n.08	TCP-IP Port	0 - 9999	1001

### PROTOCOLLO MODBUS® RTU

Quando si utilizza il protocollo Modbus® RTU, la struttura del messaggio di comunicazione è così costituita:

T1	Indirizzo ( 8 bit)	Funzione (8 bit)	Dati (N x 8 bit)	CRC (16 bit)	T1
T2					T2
T3					T3

- Il campo Indirizzo contiene l'indirizzo dello strumento slave cui il messaggio viene inviato.
- Il campo Funzione contiene il codice della funzione che deve essere eseguita dallo slave.
- Il campo Dati contiene i dati inviati allo slave o quelli inviati dallo slave come risposta ad una domanda.
- Per la serie DME\_CD, la lunghezza massima consentita per il campo dati è di 80 registri da 16 bit

• Il campo CRC consente sia al master che allo slave di verificare se ci sono errori di trasmissione. Questo consente, in caso di disturbo sulla linea di trasmissione, di ignorare il messaggio inviato per evitare problemi sia dal lato master che slave.

- La sequenza T1 T2 T3 corrisponde al tempo durante il quale non devono essere scambiati dati sul bus di comunicazione, per consentire agli strumenti collegati di riconoscere la fine di un messaggio e l'inizio del successivo. Questo tempo deve essere pari a 3.5 caratteri.

Il DME\_CD misura il tempo trascorso tra la ricezione di un carattere e il successivo e se questo tempo supera quello necessario per trasmettere 3.5 caratteri, riferiti al baud rate impostato, il prossimo carattere viene considerato l'inizio di un nuovo messaggio.

### FUNZIONI MODBUS®

Le funzioni disponibili sono:

<b>03 = Read input register</b>	Consente la lettura delle misure disponibili nel DME_CD
<b>04 = Read input register</b>	Consente la lettura delle misure disponibili nel DME_CD.
<b>06 = Preset single register</b>	Permette la scrittura dei parametri
<b>07 = Read exception</b>	Permette di leggere lo stato dell'apparecchio
<b>10 = Preset multiple register</b>	Permette la scrittura di più parametri
<b>17 = Report slave ID</b>	Permette di leggere informazioni relative all'apparecchio

Per esempio, se si vuole leggere dal DME\_CD con indirizzo 01 il valore del contatore totale 1 che si trova alla locazione 256 (100 Hex), il messaggio da spedire è il seguente:

01	04	00	FF	00	02	41	FB
----	----	----	----	----	----	----	----

Dove:

01= indirizzo slave

04 = funzione di lettura locazione

00 FF = indirizzo della locazione diminuito di un'unità, contenete il valore del contatore totale 1

00 02 = numero di registri da leggere a partire dall'indirizzo 22

41 FB = checksum CRC

### MODBUS® RTU PROTOCOL

If one selects the Modbus® RTU protocol, the communication message has the following structure:

T1	Address ( 8 bit)	Function (8 bit)	Data (N x 8 bit)	CRC (16 bit)	T1
T2					T2
T3					T3

- The Address field holds the serial address of the slave destination device.
- The Function field holds the code of the function that must be executed by the slave.
- The Data field contains data sent to the slave or data received from the slave in response to a query.
- For the DME\_CD series, the maximum length for the data field is 80 16-bit registers

• The CRC field allows the master and slave devices to check the message integrity. If a message has been corrupted by electrical noise or interference, the CRC field allows the devices to recognize the error and thereby to ignore the message.

- The T1 T2 T3 sequence corresponds to a time in which data must not be exchanged on the communication bus to allow the connected devices to recognize the end of one message and the beginning of another. This time must be at least 3.5 times the time required to send one character.

The DME\_CD measures the time that elapses from the reception of one character and the following. If this time exceeds the time necessary to send 3.5 characters at the selected baudrate, then the next character will be considered as the first of a new message.

### MODBUS® FUNCTIONS

The available functions are:

<b>03 = Read input register</b>	Allows to read the DME_CD measures.
<b>04 = Read input register</b>	Allows to read the DME_CD measures.
<b>06 = Preset single register</b>	Allows writing parameters
<b>07 = Read exception</b>	Allows to read the device status
<b>10 = Preset multiple register</b>	Allows writing several parameters
<b>17 = Report slave ID</b>	Allows to read information about the device.

For instance, to read the value of total counter 1 , which resides at location 256 (100 Hex) from the DME\_CD with serial address 01, the message to send is the following:

01	04	00	FF	00	02	41	FB
----	----	----	----	----	----	----	----

Whereas:

01= slave address

04 = Modbus® function 'Read input register'

00 FF = Address of the required register (total counter 1) decreased by one

00 02 = Number of registers to be read beginning from address 22

41 FB = CRC Checksum

La risposta del DME\_CD è la seguente:

01	04	04	00	00	7C	C4	DA	D7
----	----	----	----	----	----	----	----	----

Dove:

01 = indirizzo del DME\_CD (Slave 01)

04 = funzione richiesta dal Master

04 = numero di byte inviati dal DME\_CD

00 00 7C C4 = valore esadecimale del contatore totale 1

= 31940 = 319,40

DA D7 = checksum CRC

#### FUNZIONE 04: READ INPUT REGISTER

La funzione 04 permette di leggere una o più grandezze consecutive in memoria. L'indirizzo di ciascuna grandezza e' indicato nelle Tabelle 2-4 riportate nelle ultime pagine del presente manuale.

Come da standard Modbus®, l'indirizzo specificato nel messaggio va diminuito di 1 rispetto a quello effettivo riportato nella tabella.

Se l'indirizzo richiesto non è compreso nella tabella o il numero di registri richiesti è maggiore del numero consentito il DME\_CD ritorna un messaggio di errore (vedi tabella errori).

#### Richiesta Master:

Indirizzo slave	08h
Funzione	04h
MSB Indirizzo registro	00h
LSB Indirizzo registro	0Fh
MSB Numero registri	00h
LSB Numero registri	08h
MSB CRC	C1h
LSB CRC	56h

Nell'esempio vengono richiesti ,allo slave numero 8, 8 registri consecutivi a partire dall'indirizzo 10h.

Quindi vengono letti i registri dall' 10h al 17h.

Il comando termina sempre con il valore di checksum CRC.

#### Risposta Slave:

Indirizzo slave	08h
Funzione	04h
Numero di byte	10h
MSB Dato 10h	00h
LSB Dato 10h	00h
-----	----
MSB Dato 17h	00h
LSB Dato 17h	00h
MSB CRC	5Eh
LSB CRC	83h

La risposta è composta sempre dall'indirizzo dello slave, dalla funzione richiesta dal Master e dai dati dei registri richiesti. La risposta termina sempre con il valore di checksum CRC.

The DME\_CD answer is the following:

01	04	04	00	00	7C	C4	DA	D7
----	----	----	----	----	----	----	----	----

Where:

01 = DME\_CD address (Slave 01)

04 = Function requested by the master

04 = Number of bytes sent by the DME\_CD

00 00 7C C4 = Hex value of the total counter 1

= 31940 = 319,40

DA D7 = CRC checksum

#### FUNCTION 04: READ INPUT REGISTER

The Modbus® function 04 allows to read one or more consecutive registers from the slave memory. The address of each measure is given in the tables 2-4 on the final pages of this manual.

As for Modbus® standard, the address in the query message must be decreased by one from the effective address reported in the table.

If the measure address is not included in the table or the number of requested registers exceeds the acceptable max number, the DME\_CD will return an error code (see error table).

#### Master query:

Slave address	08h
Function	04h
MSB address	00h
LSB address	0Fh
MSB register number	00h
LSB register number	08h
MSB CRC	C1h
LSB CRC	56h

In the above example, slave 08 is requested for 8 consecutive registers beginning with address 10h.

Thus, registers from 10h to 17h will be returned. As usual, the message ends with the CRC checksum.

#### Slave response:

Slave address	08h
Function	04h
Byte number	10h
MSB register 10h	00h
LSB register 10h	00h
-----	----
MSB register 17h	00h
LSB register 17h	00h
MSB CRC	5Eh
LSB CRC	83h

The response is always composed of the slave address, the function code requested by the master and the contents of the requested registers. The answer ends with the CRC.

#### FUNZIONE 06: PRESET SINGLE REGISTER

Questa funzione permette di scrivere nei registri. Essa può essere utilizzata solo con i registri di indirizzo superiore a 1000 Hex. È possibile ad esempio impostare i parametri del setup. Qualora il valore impostato non rientri nel valore minimo e massimo della tabella il DME\_CD risponderà con un messaggio di errore. Se viene richiesto un parametro ad un indirizzo inesistente verrà risposto con un messaggio di errore. L'indirizzo ed il range valido per i vari parametri può essere trovato nelle Tabelle 5, 6 e 7.

##### Richiesta Master:

Indirizzo slave	08h
Funzione	06h
MSB Indirizzo registro	2Fh
LSB Indirizzo registro	0Fh
MSB Dato	00h
LSB Dato	0Ah
MSB CRC	31h
LSB CRC	83h

##### Risposta Slave:

La risposta è un eco della domanda, cioè viene inviato al master l'indirizzo del dato da modificare e il nuovo valore del parametro.

#### FUNZIONE 07: READ EXCEPTION STATUS

Tale funzione permette di leggere lo stato in cui si trova il commutatore di linea.

##### Richiesta Master:

Indirizzo slave	08h
Funzione	07h
MSB CRC	47h
LSB CRC	B2h

La tabella seguente riporta il significato del byte inviato dal DME\_CD come risposta:

BIT	SIGNIFICATO
0	Verifica checksum memoria programma
1	
2	
3	
4	
5	
6	
7	

#### FUNZIONE 17: REPORT SLAVE ID

Questa funzione permette di identificare il tipo di multimetro.

##### Richiesta Master.

Indirizzo slave	08h
Funzione	11h
MSB CRC	C6h
LSB CRC	7Ch

#### FUNZIONE 06: PRESET SINGLE REGISTER

This function allows to write in the registers. It can be used only with registers with address higher than 1000 Hex. For instance, it is possible to change setup parameters. If the value is not in the correct range, the DME\_CD will answer with an error message. In the same way, if the parameter address is not recognised, the DME\_CD will send an error response.

The address and the valid range for each parameter are indicated in Tables 5, 6 and 7.

##### Master message:

Indirizzo slave	08h
Funzione	06h
MSB Indirizzo registro	2Fh
LSB Indirizzo registro	0Fh
MSB Dato	00h
LSB Dato	0Ah
MSB CRC	31h
LSB CRC	83h

##### Slave response:

The slave response is an echo to the query, that is the slave sends back to the master the address and the new value of the variable.

#### FUNZIONE 07: READ EXCEPTION STATUS

This function allows to read the status of the automatic transfer switch.

##### Master query:

Slave address	08h
Function	07h
MSB CRC	47h
LSB CRC	B2h

The following table gives the meaning of the status byte sent by the DME\_CD as answer:

BIT	MEANING
0	Checksum verify to program memory
1	
2	
3	
4	
5	
6	
7	

#### FUNZIONE 17: REPORT SLAVE ID

This function allows to identify the multimeter type.

##### Master query.

Slave address	08h
Function	11h
MSB CRC	C6h
LSB CRC	7Ch

**Risposta Slave:**

Indirizzo slave	08h
Funzione	11h
Contatore bytes	04 h
Dato 1 (Tipo)	C8h
Dato 2 (Revisione software)	04h
Dato 3 (Revisione hardware)	00h
Dato 4 (Revisione parametri)	01h
MSB CRC	...h
LSB CRC	...h

**ERRORI**

Nel caso lo slave riceva un messaggio errato, segnala la condizione al master rispondendo con un messaggio composto dalla funzione richiesta in OR con 80 Hex, seguita da un codice di errore.

Nella seguente tabella vengono riportati i codici di errore inviati dallo slave al master:

**TABELLA 1: CODICI ERRORE**

COD	ERRORE
01	Funzione non valida
02	Indirizzo registro illegale
03	Valore del parametro fuori range
04	Impossibile effettuare operazione
06	Slave occupato, funzione momentaneamente non disponibile

**FUNZIONE 16: PRESET MULTIPLE REGISTER**

Questa funzione permette di modificare più parametri consecutivamente o parametri composti da più di 2 byte. L'indirizzo ed il range valido per i vari parametri possono essere trovati nella Tabella 8.

**Richiesta Master:**

Indirizzo slave	08h
Funzione	10h
MSB Indirizzo registro	20h
LSB Indirizzo registro	01h
MSB Numero registri	00h
LSB Numero registri	02h
MSB Dato	00h
LSB Dato	00h
MSB Dato	00h
LSB Dato	00h
MSB CRC	85h
LSB CRC	3Eh

**Risposta Slave:**

Indirizzo slave	08h
Funzione	10h
MSB Indirizzo registro	20h
LSB Indirizzo registro	01h
MSB Numero byte	00h
LSB Numero byte	02h
MSB CRC	1Bh
LSB CRC	51h

**Slave response:**

Slave address	08h
Function	11h
Byte count	04 h
Data 01 –Type	C8h
Data 02 – (Sw revision)	04h
Data 03 – (Hardware revision)	00h
Data 04 – (Parameter revision)	01h
MSB CRC	...h
LSB CRC	...h

**ERRORS**

In case the slave receives an incorrect message, it answers with a message composed by the queried function ORed with 80 Hex, followed by an error code byte.

In the following table are reported the error codes sent by the slave to the master:

**TABLE 1: ERROR CODES**

CODE	ERROR
01	Invalid function
02	Invalid address
03	Parameter out of range
04	Function execution impossible
06	Slave busy, function momentarily not available

**FUNZIONE 16: PRESET MULTIPLE REGISTER**

This function allows to modify multiple parameters with a single message, or to preset a value longer than one register. The address and the valid range for each parameter are stated in Table 8.

**Master message:**

Slave address	08h
Function	10h
MSB register address	20h
LSB register address	01h
MSB register number	00h
LSB register number	02h
MSB data	00h
LSB data	00h
MSB data	00h
LSB data	00h
MSB CRC	85h
LSB CRC	3Eh

**Slave response:**

Slave address	08h
Function	10h
MSB register address	20h
LSB register address	01h
MSB byte number	00h
LSB byte number	02h
MSB CRC	1Bh
LSB CRC	51h

### PROTOCOLLO MODBUS® ASCII

Il protocollo Modbus® ASCII viene utilizzato normalmente nelle applicazioni che richiedono di comunicare via modem.

Le funzioni e gli indirizzi disponibili sono gli stessi della versione RTU, ma i caratteri trasmessi sono in ASCII e la terminazione del messaggio non è effettuata a tempo ma con dei caratteri di ritorno a capo.

Se si seleziona il parametro P7.x.05 o P7.05 o come protocollo Modbus® ASCII, la struttura del messaggio di comunicazione sulla relativa porta di comunicazione è così costituita:

:	Indirizzo 2 chars	Funzione 2 chars	Dati (N chars)	LRC 2 chars	CR LF
---	----------------------	---------------------	-------------------	----------------	----------

- Il campo Indirizzo contiene l'indirizzo dello strumento slave cui il messaggio viene inviato.
- Il campo Funzione contiene il codice della funzione che deve essere eseguita dallo slave.
- Il campo Dati contiene i dati inviati allo slave o quelli inviati dallo slave come risposta ad una domanda. La massima lunghezza consentita è di (ved. Pag. 3) registri consecutivi.
- Il campo LRC consente sia al master che allo slave di verificare se ci sono errori di trasmissione. Questo consente, in caso di disturbo sulla linea di trasmissione, di ignorare il messaggio inviato per evitare problemi sia dal lato master che slave.
- Il messaggio termina sempre con i caratteri di controllo CRLF (0D 0A).

#### Esempio:

Per esempio, se si vuole leggere dal DME\_CD con indirizzo 8 il valore del contatore totale 2 che si trova alla locazione 259 (103 Hex), il messaggio da spedire è il seguente:

:	08	04	01	02	00	02	E7	CRLF
---	----	----	----	----	----	----	----	------

Dove:

: = ASCII 3Ah = Delimitatore inizio messaggio  
08 = indirizzo slave.  
04 = funzione di lettura locazione.  
01 02 = indirizzo della locazione diminuito di un'unità, contenente il valore del contatore totale 2  
00 02 = numero di registri da leggere a partire dall'indirizzo 04.  
E7 = checksum LRC.  
CRLF = ASCII 0Dh 0Ah = delimitatore fine messaggio

La risposta del DME\_CD è la seguente:

:	08	04	04	00	00	07	30	9B	CR LF
---	----	----	----	----	----	----	----	----	----------

Dove:

: = ASCII 3Ah = Delimitatore inizio messaggio  
08 = indirizzo del DME\_CD (Slave 08).  
04 = funzione richiesta dal Master.  
04 = numero di byte inviati dallo slave.  
00 00 07 30 = valore esadecimale del contatore totale 2 = 1840 = 18,40.  
9B = checksum LRC.  
CRLF = ASCII 0Dh 0Ah = delimitatore fine messaggio

### MODBUS® ASCII PROTOCOL

The Modbus® ASCII protocol is normally used in application that require to communicate through a couple of modems.

The functions and addresses available are the same as for the RTU version, but the transmitted characters are in ASCII and the message end is delimited by Carriage return/ Line Feed instead of a transmission pause.

If one selects the parameter P7.x.05 or P7.05 as Modbus® ASCII protocol, the communication message on the correspondent communication port has the following structure:

:	Address (2 chars)	Function (2 chars)	Dates (N chars)	LRC (2 chars)	CR LF
---	----------------------	-----------------------	--------------------	---------------------	----------

- The Address field holds the serial address of the slave destination device.
- The Function field holds the code of the function that must be executed by the slave.
- The Data field contains data sent to the slave or data received from the slave in response to a query. The maximum allowable length is of (read pag. 3) consecutive registers.
- The LRC field allows the master and slave devices to check the message integrity. If a message has been corrupted by electrical noise or interference, the LRC field allows the devices to recognize the error and thereby ignore the message.
- The message terminates always with CRLF control character (0D 0A).

#### Example:

For instance, to read the value of the total counter 2, which resides at location 259 (103 Hex), from the slave with serial address 08, the message to send is the following:

:	08	04	01	02	00	02	E7	CRLF
---	----	----	----	----	----	----	----	------

Whereas:

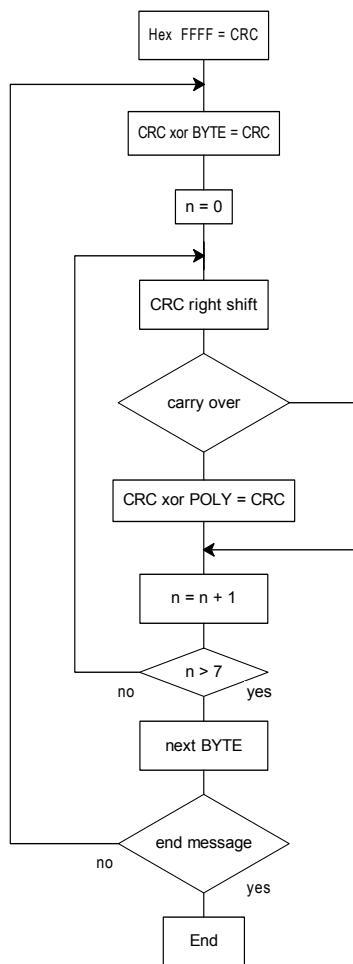
: = ASCII 3Ah message start delimiter  
08 = slave address  
04 = Modbus® function 'Read input register'  
01 02 = Address of the required register (total counter 2) decreased by one  
00 02 = Number of registers to be read beginning from address 04  
E7 = LRC Checksum  
CRLF = ASCII 0Dh 0Ah = Message end delimiter

The DME\_CD answer is the following:

:	08	04	04	00	00	07	30	9B	CR LF
---	----	----	----	----	----	----	----	----	----------

Whereas:

: = ASCII 3Ah message start delimiter  
08 = DME\_CD address (Slave 08)  
04 = Function requested by the master  
04 = Number of bytes sent by the multimeter  
00 00 07 30 = Hex value of the current phase of total counter 2 = 1840 = 18,40.  
9B = LRC checksum  
CRLF = ASCII 0Dh 0Ah = Message end delimiter



**Algoritmo di calcolo del CRC**  
CRC calculation algorithm

**CALCOLO DEL CRC (CHECKSUM per RTU)**

Esempio di calcolo:  
Frame = 0207h

<b>Inizializzazione CRC</b>	1111	1111	1111	1111
Carica primo byte			0000	0010
Esegue xor con il primo Byte della frame	1111	1111	1111	1101
Esegue primo shift dx	0111	1111	1111	1110 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1101	1111	1111	1111
Esegue secondo shift dx	0110	1111	1111	1111 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1100	1111	1111	1110
Esegue terzo shift	0110	0111	1111	1111 0
Esegue quarto shift	0011	0011	1111	1111 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il Polinomio	1001	0011	1111	1110
Esegue quinto shift dx	0100	1001	1111	1111 0
Esegue sesto shift dx	0010	0100	1111	1111 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con polinomio	1000	0100	1111	1110
Esegue settimo shift dx	0100	0010	0111	1111 0
Esegue ottavo shift dx	0010	0001	0011	1111 1
Carry=1, carica polinomio	1010	0000	0000	0001
Carica secondo byte della frame			0000	0111
Esegue xor con il Secondo byte della frame	1000	0001	0011	1001
Esegue primo shift dx	0100	0000	1001	1100 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1110	0000	1001	1101
Esegue secondo shift dx	0111	0000	0100	1110 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1101	0000	0100	1111
Esegue terzo shift dx	0110	1000	0010	0111 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1100	1000	0010	0110
Esegue quarto shift dx	0110	0100	0001	0011 0
Esegue quinto shift dx	0010	0100	0000	1001 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1001	0010	0000	1000
Esegue sesto shift dx	0100	1001	0000	0100 0
Esegue settimo shift dx	0010	0100	1000	0010 0
Esegue ottavo shift dx	0001	0010	0100	0001 0
<b>Risultato CRC</b>	<b>0001</b>	<b>0010</b>		
	<b>0100</b>	<b>0001</b>		
	<b>12h</b>	<b>41h</b>		

**Nota:** Il byte 41h viene spedito per primo (anche se e' il LSB), poi viene trasmesso 12h.

**CALCOLO LRC (CHECKSUM per ASCII)**

Esempio di calcolo:

Indirizzo	01	00000001
Funzione	04	00000100
Start address hi.	00	00000000
Start address lo.	00	00000000
Numero registri	08	00001000
	Somma	00001101
	Complemento a 1	11110010
	+ 1	00000001
	Complemento a 2	11110101

**Risultato LRC** **F5**

**CRC CALCULATION (CHECKSUM for RTU)**

Example of CRC calculation:  
Frame = 0207h

<b>CRC initialization</b>	1111	1111	1111	1111
Load the 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
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 5th right shift	0010	0100	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
<b>CRC Result</b>	<b>0001</b>	<b>0010</b>		
	<b>0100</b>	<b>0001</b>		
	<b>12h</b>	<b>41h</b>		

**Note:** The byte 41h is sent first(even if it is the LSB), then 12h is sent.

**LRC CALCULATION (CHECKSUM for ASCII)**

Example of LRC calculation:

Address	01	00000001
Function	04	00000100
Start address hi.	00	00000000
Start address lo.	00	00000000
Number of registers	08	00001000
	Sum	00001101
	1. complement	11110010
	+ 1	00000001
	2. complement	11110101

**LRC result** **F5**

TABELLA 2:

MISURE FORNITE DAL PROTOCOLLO DI COM.

(Utilizzabili con funzioni 03 e 04)

TABLE 2:

MEASURES SUPPLIED BY SERIAL COMMUNICATION  
PROTOCOL

(To be used with functions 03 and 04)

Indirizzo Address	WORDS	MISURA	MEASURE	UNITA' UNIT	FORMATO FORMAT
0100H	2	Contatore totale 1	Total counter 1	UM/100	signed long
0102H	2	Contatore totale 2	Total counter 2	UM/100	signed long
0104H	2	Contatore totale 3	Total counter 3	UM/100	signed long
0106H	2	Contatore totale 4	Total counter 4	UM/100	signed long
0108H	2	Contatore totale 5	Total counter 5	UM/100	signed long
010AH	2	Contatore totale 6	Total counter 6	UM/100	signed long
010CH	2	Contatore totale 7	Total counter 7	UM/100	signed long
010EH	2	Contatore totale 8	Total counter 8	UM/100	signed long
0110H	2	Contatore totale 9	Total counter 9	UM/100	signed long
0112H	2	Contatore totale 10	Total counter 10	UM/100	signed long
0114H	2	Contatore totale 11	Total counter 11	UM/100	signed long
0116H	2	Contatore totale 12	Total counter 12	UM/100	signed long
0118H	2	Contatore totale 13	Total counter 13	UM/100	signed long
011AH	2	Contatore totale 14	Total counter 14	UM/100	signed long
011CH	2	Contatore totale 15	Total counter 15	UM/100	signed long
011EH	2	Contatore totale 16	Total counter 16	UM/100	signed long
0140H	2	Contatore parziale 1	Partial counter 1	UM/100	signed long
0142H	2	Contatore parziale 2	Partial counter 2	UM/100	signed long
0144H	2	Contatore parziale 3	Partial counter 3	UM/100	signed long
0146H	2	Contatore parziale 4	Partial counter 4	UM/100	signed long
0148H	2	Contatore parziale 5	Partial counter 5	UM/100	signed long
014AH	2	Contatore parziale 6	Partial counter 6	UM/100	signed long
014CH	2	Contatore parziale 7	Partial counter 7	UM/100	signed long
014EH	2	Contatore parziale 8	Partial counter 8	UM/100	signed long
0150H	2	Contatore parziale 9	Partial counter 9	UM/100	signed long
0152H	2	Contatore parziale 10	Partial counter 10	UM/100	signed long
0154H	2	Contatore parziale 11	Partial counter 11	UM/100	signed long
0156H	2	Contatore parziale 12	Partial counter 12	UM/100	signed long
0158H	2	Contatore parziale 13	Partial counter 13	UM/100	signed long
015AH	2	Contatore parziale 14	Partial counter 14	UM/100	signed long
015CH	2	Contatore parziale 15	Partial counter 15	UM/100	signed long
015EH	2	Contatore parziale 16	Partial counter 16	UM/100	signed long
0180H	2	Derivata contatore 1	Derivative counter 1	UM/100	signed long
0182H	2	Derivata contatore 2	Derivative counter 2	UM/100	signed long
0184H	2	Derivata contatore 3	Derivative counter 3	UM/100	signed long
0186H	2	Derivata contatore 4	Derivative counter 4	UM/100	signed long
0188H	2	Derivata contatore 5	Derivative counter 5	UM/100	signed long
018AH	2	Derivata contatore 6	Derivative counter 6	UM/100	signed long
018CH	2	Derivata contatore 7	Derivative counter 7	UM/100	signed long
018EH	2	Derivata contatore 8	Derivative counter 8	UM/100	signed long
0190H	2	Derivata contatore 9	Derivative counter 9	UM/100	signed long
0192H	2	Derivata contatore 10	Derivative counter 10	UM/100	signed long
0194H	2	Derivata contatore 11	Derivative counter 11	UM/100	signed long
0196H	2	Derivata contatore 12	Derivative counter 12	UM/100	signed long
0198H	2	Derivata contatore 13	Derivative counter 13	UM/100	signed long
019AH	2	Derivata contatore 14	Derivative counter 14	UM/100	signed long
019CH	2	Derivata contatore 15	Derivative counter 15	UM/100	signed long
019EH	2	Derivata contatore 16	Derivative counter 16	UM/100	signed long

Indirizzo Address	WORDS	MISURA	MEASURE	UNITA' UNIT	FORMATO FORMAT
0200H	2	Contatore 1 tar 01	Counter 1 tar 01	UM/100	signed long
0202H	2	Contatore 1 tar 02	Counter 1 tar 02	UM/100	signed long
0204H	2	Contatore 1 tar 03	Counter 1 tar 03	UM/100	signed long
0206H	2	Contatore 1 tar 04	Counter 1 tar 04	UM/100	signed long
0208H	2	Contatore 2 tar 01	Counter 2 tar 01	UM/100	signed long
020AH	2	Contatore 2 tar 02	Counter 2 tar 02	UM/100	signed long
020CH	2	Contatore 2 tar 03	Counter 2 tar 03	UM/100	signed long
020EH	2	Contatore 2 tar 04	Counter 2 tar 04	UM/100	signed long
.....		.....	.....	.....	
0278H	2	Contatore 16 tar 01	Counter 16 tar 01	UM/100	signed long
027AH	2	Contatore 16 tar 02	Counter 16 tar 02	UM/100	signed long
027CH	2	Contatore 16 tar 03	Counter 16 tar 03	UM/100	signed long
027EH	2	Contatore 16 tar 04	Counter 16 tar 04	UM/100	signed long
0300H	4	Matematico 1	Mathematics 1	UM/100	signed long long
0304H	4	Matematico 2	Mathematics 2	UM/100	signed long long
0308H	4	Matematico 3	Mathematics 3	UM/100	signed long long
030CH	4	Matematico 4	Mathematics 4	UM/100	signed long long
0310H	4	Matematico 5	Mathematics 5	UM/100	signed long long
0314H	4	Matematico 6	Mathematics 6	UM/100	signed long long
0318H	4	Matematico 7	Mathematics 7	UM/100	signed long long
031CH	4	Matematico 8	Mathematics 8	UM/100	signed long long
0320H	4	Matematico 9	Mathematics 9	UM/100	signed long long
0324H	4	Matematico 10	Mathematics 10	UM/100	signed long long
0328H	4	Matematico 11	Mathematics 11	UM/100	signed long long
032CH	4	Matematico 12	Mathematics 12	UM/100	signed long long
0330H	4	Matematico 13	Mathematics 13	UM/100	signed long long
0324H	4	Matematico 14	Mathematics 14	UM/100	signed long long
0338H	4	Matematico 15	Mathematics 15	UM/100	signed long long
033CH	4	Matematico 16	Mathematics 16	UM/100	signed long long
2810H	1	Flag divisione per zero del MAT01	Flag division by zero of MAT01	bool	Unsigned int
2811H	1	Flag divisione per zero del MAT02	Flag division by zero of MAT02	bool	Unsigned int
2812H	1	Flag divisione per zero del MAT03	Flag division by zero of MAT03	bool	Unsigned int
2813H	1	Flag divisione per zero del MAT04	Flag division by zero of MAT04	bool	Unsigned int
2814H	1	Flag divisione per zero del MAT05	Flag division by zero of MAT05	bool	Unsigned int
2815H	1	Flag divisione per zero del MAT06	Flag division by zero of MAT06	bool	Unsigned int
2816H	1	Flag divisione per zero del MAT07	Flag division by zero of MAT07	bool	Unsigned int
2817H	1	Flag divisione per zero del MAT08	Flag division by zero of MAT08	bool	Unsigned int
2818H	1	Flag divisione per zero del MAT09	Flag division by zero of MAT09	bool	Unsigned int
2819H	1	Flag divisione per zero del MAT10	Flag division by zero of MAT10	bool	Unsigned int
281AH	1	Flag divisione per zero del MAT11	Flag division by zero of MAT11	bool	Unsigned int
281BH	1	Flag divisione per zero del MAT12	Flag division by zero of MAT12	bool	Unsigned int
281CH	1	Flag divisione per zero del MAT13	Flag division by zero of MAT13	bool	Unsigned int
281DH	1	Flag divisione per zero del MAT14	Flag division by zero of MAT14	bool	Unsigned int
281EH	1	Flag divisione per zero del MAT15	Flag division by zero of MAT15	bool	Unsigned int
281FH	1	Flag divisione per zero del MAT16	Flag division by zero of MAT16	bool	Unsigned int

Indirizzo Address	WORDS	MISURA	MEASURE	UNITA' UNIT	FORMATO FORMAT
0400H	2	Contatore totale 1	Total counter 1	①	signed long
0402H	2	Contatore totale 2	Total counter 2	①	signed long
0404H	2	Contatore totale 3	Total counter 3	①	signed long
0406H	2	Contatore totale 4	Total counter 4	①	signed long
0408H	2	Contatore totale 5	Total counter 5	①	signed long
040AH	2	Contatore totale 6	Total counter 6	①	signed long
040CH	2	Contatore totale 7	Total counter 7	①	signed long
040EH	2	Contatore totale 8	Total counter 8	①	signed long
0410H	2	Contatore totale 9	Total counter 9	①	signed long
0412H	2	Contatore totale 10	Total counter 10	①	signed long
0414H	2	Contatore totale 11	Total counter 11	①	signed long
0416H	2	Contatore totale 12	Total counter 12	①	signed long
0418H	2	Contatore totale 13	Total counter 13	①	signed long
041AH	2	Contatore totale 14	Total counter 14	①	signed long
041CH	2	Contatore totale 15	Total counter 15	①	signed long
041EH	2	Contatore totale 16	Total counter 16	①	signed long
0440H	2	Contatore parziale 1	Partial counter 1	①	signed long
0442H	2	Contatore parziale 2	Partial counter 2	①	signed long
0444H	2	Contatore parziale 3	Partial counter 3	①	signed long
0446H	2	Contatore parziale 4	Partial counter 4	①	signed long
0448H	2	Contatore parziale 5	Partial counter 5	①	signed long
044AH	2	Contatore parziale 6	Partial counter 6	①	signed long
044CH	2	Contatore parziale 7	Partial counter 7	①	signed long
044EH	2	Contatore parziale 8	Partial counter 8	①	signed long
0450H	2	Contatore parziale 9	Partial counter 9	①	signed long
0452H	2	Contatore parziale 10	Partial counter 10	①	signed long
0454H	2	Contatore parziale 11	Partial counter 11	①	signed long
0456H	2	Contatore parziale 12	Partial counter 12	①	signed long
0458H	2	Contatore parziale 13	Partial counter 13	①	signed long
045AH	2	Contatore parziale 14	Partial counter 14	①	signed long
045CH	2	Contatore parziale 15	Partial counter 15	①	signed long
045EH	2	Contatore parziale 16	Partial counter 16	①	signed long
0500H	2	Contatore 1 tar 01	Counter 1 tar 01	①	signed long
0502H	2	Contatore 1 tar 02	Counter 1 tar 02	①	signed long
0504H	2	Contatore 1 tar 03	Counter 1 tar 03	①	signed long
0506H	2	Contatore 1 tar 04	Counter 1 tar 04	①	signed long
0508H	2	Contatore 2 tar 01	Counter 2 tar 01	①	signed long
050AH	2	Contatore 2 tar 02	Counter 2 tar 02	①	signed long
050CH	2	Contatore 2 tar 03	Counter 2 tar 03	①	signed long
050EH	2	Contatore 2 tar 04	Counter 2 tar 04	①	signed long
.....		.....	.....	.....	
0578H	2	Contatore 16 tar 01	Counter 16 tar 01	①	signed long
057AH	2	Contatore 16 tar 02	Counter 16 tar 02	①	signed long
057CH	2	Contatore 16 tar 03	Counter 16 tar 03	①	signed long
057EH	2	Contatore 16 tar 04	Counter 16 tar 04	①	signed long

① Se il valore del parametro P01.n.09 è uguale a 1 il valore letto non contiene decimali altrimenti contiene 2 decimali.  
If the value of the parameter P01.n.09 is equal to 1 the value read doesn't contain a decimal otherwise contains 2 decimal.

Indirizzo Address	WORDS	MISURA	MEASURE	UNITA' UNIT	FORMATO FORMAT
2100H	1	OR di tutti gli ingressi	OR of all Inputs	②	Unsigned int
2101H	1	Ingresso 1	Input 1	bool	Unsigned int
2102	1	Ingresso 2	Input 2	bool	Unsigned int
2103	1	Ingresso 3	Input 3	bool	Unsigned int
2104	1	Ingresso 4	Input 4	bool	Unsigned int
2105	1	Ingresso 5	Input 5	bool	Unsigned int
2106	1	Ingresso 6	Input 6	bool	Unsigned int
2107	1	Ingresso 7	Input 7	bool	Unsigned int
2108	1	Ingresso 8	Input 8	bool	Unsigned int
2150H	1	Ingresso 9	Input 9	bool	Unsigned int
2151	1	Ingresso 10	Input 10	bool	Unsigned int
2152	1	Ingresso 11	Input 11	bool	Unsigned int
2153	1	Ingresso 12	Input 12	bool	Unsigned int
2154	1	Ingresso 13	Input 13	bool	Unsigned int
2155	1	Ingresso 14	Input 14	bool	Unsigned int
2156	1	Ingresso 15	Input 15	bool	Unsigned int
2157	1	Ingresso 16	Input 16	bool	Unsigned int
2110H	1	OR di tutte le uscite	OR of all Outputs	②	Unsigned int
2111H	1	Uscite 1	Output 1	bool	Unsigned int
.....					
2118H	1	Uscite 8	Output 8	bool	Unsigned int
2120H	1	OR di tutti gli allarmi	OR of all Alarms	②	Unsigned int
2121H	1	Allarme 1	Alarm 1	③	Unsigned int
....					
2128H	1	Allarme 8	Alarm 8	③	Unsigned int
2130H	1	OR di tutti i booleani	OR of all Boolean	②	Unsigned int
2131H	1	Booleano 1	Boolean 1	bool	Unsigned int
...					
2138H	1	Booleano 8	Boolean 8	bool	Unsigned int
2140H	1	OR Tutti i limiti	OR All Limits	②	Unsigned int
2141H	1	Limite 1	Limit 1	bool	Unsigned int
...					
0148H	1	Limite 8	Limit 8	bool	Unsigned int
4F00H	1	Remoto 1	Remote 1	bool	Unsigned int
....					
4F0F	1	Remoto 16	Remote 16	bool	Unsigned int

② Esempio:

Il valore all'indirizzo 2100H è 0x05 (esadecimale),  
= 0x00000101 vuol dire che gli ingressi 1 e 3 sono  
attivi.

③ Lo stato del bit 0 indica l'allarme in ritenuta.  
Lo stato del bit 1 indica l'allarme attivo.

② Example:

The value at address 2100H is 0x05 (hexadecimal)  
= 0x00000101 means that the inputs 1 and 3 are  
active

③ The status of bit 0 indicates a latched alarm.  
The status of bit 1 indicates an active alarm.

**TABELLA 3:  
COMANDI  
(Utilizzabili con funzione 06)**

**TABLE 3:  
COMMANDS  
(To be used with function 06)**

INDIRIZZO ADDRESS	WORDS	COMANDO	COMMAND	VALORE VALUE	FORMATO FORMAT
2FF0H	1	Azzera contatori parziali	Reset partial counters	0	Unsigned int
2FF0H	1	Azzera contaore parziale	Reset partial hour	1	Unsigned int
2FF0H	1	Azzera tariffe	Reset Energy Tariff	2	Unsigned int
2FF0H	1	Azzera allarmi	Reset Alarms	3	Unsigned int
2FF0H	1	Azzera limiti	Reset Limits	4	Unsigned int
2FF0h	1	Azzera contatori totali	Reset partial total	9	Unsigned int
2FF0H	1	Azzera contaore totale	Reset Totat hour	10	Unsigned int
2FF0H	1	Setup a default	Parametrs to default	①11	Unsigned int
2FF0H	1	Salva copia setup	Backup parameters	①12	Unsigned int
2FF0H	1	Ripristina setup	Restore parameters	①13	Unsigned int
4200H	1	Impostazione tariffa energ.	Set Energy tariff	②1÷4	Unsigned int

**① ATTENZIONE**

Dopo aver usato questo comando è preferibile utilizzare il comando di REBOOT.

**② ATTENZIONE**

Dopo avere eseguito questo comando, per ottenere il risultato del test bisogna eseguire una domanda 4 all'indirizzo 0x1F20, il significato dei bit della risposta è riportato nella tabella sottostante.

**③ ATTENZIONE**

Questa funzione è attiva solo se nessun ingresso è programmato con la funzione tariffa (TAR-A e TAR-B)

**① ATTENTION**

After using of this command it is recommended to send REBOOT command.

**② ATTENTION**

After executing this command, to get the test result you can use the query 4 at address 0x1F20; the meaning of the bits of the response is shown in the table below.

**③ ATTENTION**

This function is enabled only if none of the inputs is set with the tariff function (TAR-A and TAR-B).

**TABELLA 8:  
PARAMETRI SETUP  
(Utilizzabili con funzioni 04 e 06)**

**TABLE 8:  
SETUP PARAMETERS  
(To be used with functions 04 and 06)**

CODE	MENU	MENU	MIN	MAX	DEF	WORDS	ADDRESS
<b>M01</b>	<b>Generale</b>	<b>General</b>					
P01.n.01	Visualizzazione contatore	Counter viewing	0	1	ON (CNT01..08) OFF (CNT09..16)	1	①5000H+ (n-1) * 80H
P01.n.02	Descrizione del contatore	Counter description			CNTn	8	①5002H+ (n-1) * 80H
P01.n.03	Unità di misura contatore	Counter unit of measure			kWh+	3	①5022H+ (n-1) * 80H
P01.n.04	Sorgente incremento conteggio	Increment signal source	0	5	INP (da 1 a 8) OFF (da 1 a 15)	1	①5032H+ (n-1) * 80H
P01.n.05	Numero canale (x)	Channel number (x)	1	16	n	1	①5034H+ (n-1) * 80H
P01.n.06	Sorgente decremento conteggio	Decrement signal source	0	5	0	1	①5036H+ (n-1) * 80H
P01.n.07	Numero canale (x)	Channel number (x)	1	16	n	1	①5038H+ (n-1) * 80H
P01.n.08	Moltiplicatore	Multiplier	1	1000	1	1	①503AH+ (n-1) * 80H
P01.n.09	Divisore	Divider	1	1000	10	1	①503CH+ (n-1) * 80H
P01.n.10	Sorgente di azzeramento contatore parziale	Partial counter reset signal source	0	5	0	1	①503EH+ (n-1) * 80H
P01.n.11	Numero canale (x)	Channel number (x)	1	16	n	1	①5040H+ (n-1) * 80H
P01.n.12	Sorgente di azzeramento contatore totale TOT	Reset source total counter TOT	0	5	0	1	①5042H+ (n-1) * 80H
P01.n.13	Numero canale (x)	Channel number (x)	1	16	n	1	①5044H+ (n-1) * 80H
P01.n.14	Tempo di calcolo derivata	Time period for derivative calculation	0	60	0	1	①5046H+ (n-1) * 80H
P01.n.15	Moltiplicatore derivata	Derivative multiplier	1	1000	60	1	①5048H+ (n-1) * 80H
P01.n.16	Divisore derivata	Derivative divider	1	1000	1	1	①504AH+ (n-1) * 80H
P01.n.17	Unità di misura derivata	Derivative unit of measure			kW+	3	①504CH+ (n-1) * 80H
<b>M02</b>	<b>Utilità</b>	<b>Utility</b>					
P02.01	Lingua	Language	0	4	0	1	5800H
P02.02	Contrasto LCD	Display contrast	0	50	100	1	5802H
P02.03	Intensità retroilluminazione display alta	High backlight level	10	100	100	1	5804H
P02.04	Intensità retroilluminazione display bassa	Low backlight level	10	100	30	1	5806H
P02.05	Tempo passaggio a retroilluminazione bassa	Delay to low backl.	5	600	30	1	5808H
P02.06	Ritorno a pagina di default	Default page return	9	600	60	1	580AH
P02.07	Pagina di default	Default page	1	32	1	1	580CH
P02.08	Sotto-pagina di default	Default sub-page	0	13	0	1	580EH
P02.09	Tempo di aggiornamento display	Display update time	1	50	5	1	5810H
<b>M03</b>	<b>Password</b>	<b>Password</b>					
P03.01	Abilitazione password	Enable passwords	0	1	0	1	5880H
P03.02	Password livello utente	User level Password	0	9999	1000	1	5882H
P03.03	Password livello avanzato	Advanced level Password	0	9999	2000	1	5884H
<b>M04</b>	<b>Contaore</b>	<b>Hour counters</b>					
P04.01	Abil. generale contaore	Hour counters enable	0	1	1	1	5900H
P04.02	Abil. contaore parziale	Partial hour counter enable	0	4	1	1	5902H
P04.03	Numero canale (x)	Channel number (x)	1	8	1	1	5904H
<b>M05</b>	<b>Grafico trend</b>	<b>Trend graph</b>					
P05.n.01	Misura per pagina trend	Trend graph measure	0	3	1	1	①5980H+ (n-1) * 80H
P05.n.02	Autorange scala	Autorange	0	1	1	1	①5982H+ (n-1) * 80H
P05.n.03	Valore fondo scala	Full scale value	0	1000	1000	1	①5984H+ (n-1) * 80H
P05.n.04	Moltiplicatore fondo scala	Full scale multiplier	0	2	0	1	①5986H+ (n-1) * 80H

CODE	MENU	MENU	MIN	MAX	DEF	WORDS	ADDRESS
<b>M06</b>	<b>Comunicazione</b>	<b>Communication</b>					
P06.n.01	Indirizzo seriale nodo	Serial node address	1	255	1	1	①6180H+ (n-1) * 80H
P06.n.02	Velocità seriale	Serial speed	0	5	3	1	①6182H+ (n-1) * 80H
P06.n.03	Formato dati	Data format	0	4	0	1	①6184H+ (n-1) * 80H
P06.n.04	Bit di stop	Stop bits	0	1	0	1	①6186H+ (n-1) * 80H
P06.n.05	Protocollo	Protocol	0	1	0	1	①6188H+ (n-1) * 80H
P06.n.06	Indirizzo IP	IP address	0	255	0.0.0.0	2	①618AH+ (n-1) * 80H
P06.n.07	Subnet mask	Subnet mask	0	255	0.0.0.0	2	①619AH+ (n-1) * 80H
P06.n.08	Porta IP	IP port	0	9999	1001	1	①61AAH+ (n-1) * 80H
P06.n.09	Funzione Gateway	Gateway function	OFF	ON	ON		①61ACH+ (n-1) * 80H
<b>M07</b>	<b>Soglie limite</b>	<b>Limit thresholds</b>					
P07.n.01	Misura riferimento	Reference measure	0	41	0	1	①6280H+ (n-1) * 80H
P07.n.02	Funzione	Function	0	2	0	1	①6282H+ (n-1) * 80H
P07.n.03	Soglia superiore	Upper threshold	-9999	9999	0	1	①6284H+ (n-1) * 80H
P07.n.04	Moltiplicatore	Multiplier	0	6	2	1	①6286H+ (n-1) * 80H
P07.n.05	Ritardo	Delay	0	6000	0	1	①6288H+ (n-1) * 80H
P07.n.06	Soglia inferiore	Lower threshold	-9999	9999	0	1	①628AH+ (n-1) * 80H
P07.n.07	Moltiplicatore	Multiplier	0	6	2	1	①628CH+ (n-1) * 80H
P07.n.08	Ritardo	Delay	0	6000	0	1	①628EH+ (n-1) * 80H
P07.n.09	Stato di riposo	Normal status	0	1	0	1	①6290H+ (n-1) * 80H
P07.n.10	Memoria	Latch	0	1	0	1	①6292H+ (n-1) * 80H
<b>M08</b>	<b>Allarmi</b>	<b>Alarms</b>					
P08.n.01	Sorgente allarme	Alarm source	0	3	0	1	①6680H + (n-1) * 80H
P08.n.02	Numero canale (x)	Channel number (x)	1	8	1	1	①6682H + (n-1) * 80H
P08.n.03	Memoria	Latch	0	1	0	1	①6684H + (n-1) * 80H
P08.n.04	Priorità	Priority	0	1	0	1	①6686H + (n-1) * 80H
P08.n.05	Testo	Text			ALAn	8	①6688H + (n-1) * 80H
<b>M09</b>	<b>Logica booleana</b>	<b>Boolean Logic</b>					
P09.n.01	Operando 1	Operand 1	0	5	0	1	①6A80H + (n-1) * 80H
P09.n.02	Numero canale (x)	Channel number (x)	1	8	1	1	①6A82H + (n-1) * 80H
P09.n.03	Operatore logico 1	Logic operator 1	0	6	0	1	①6A84H + (n-1) * 80H
P09.n.04	Operando 2	Operand 2	0	5	0	1	①6A86H + (n-1) * 80H
P09.n.05	Numero canale (x)	Channel number (x)	1	8	1	1	①6A88H + (n-1) * 80H
P09.n.06	Operatore logico 2	Logic operator 2	0	6	0	1	①6A8AH + (n-1) * 80H
P09.n.07	Operando 3	Operand 3	0	5	0	1	①6A8CH + (n-1) * 80H
P09.n.08	Numero canale (x)	Channel number (x)	1	8	1	1	①6A8EH + (n-1) * 80H
P09.n.09	Operatore logico 3	Logic operator 3	0	6	0	1	①6A90H + (n-1) * 80H
P09.n.10	Operando 4	Operand 4	0	5	0	1	①6A92H + (n-1) * 80H
P09.n.11	Numero canale (x)	Channel number (x)	1	8	1	1	①6A94H + (n-1) * 80H

CODE	MENU	MENU	MIN	MAX	DEF	WORDS	ADDRESS
<b>M10</b>	<b>Ingressi</b>	<b>Inputs</b>					
P10.n .01	Funzione ingresso	Input function	0	5	0	1	① 6E80H + (n -1) * 80H
P10.n .02	Stato a riposo	Normal status	0	1	0	1	① 6E82H + (n -1) * 80H
P10.n .03	Ritardo ON	ON delay	0	60000	5	2	① 6E84H + (n-1) * 80H
P10.n .04	Ritardo OFF	OFF delay	0	60000	5	2	① 6E86H + (n -1) * 80H
<b>M11</b>	<b>Uscite</b>	<b>Outputs</b>					
P11.n .01	Funzione uscita	Output function	0	7	0	1	① 7680H + (n -1) * 80H
P11. n.02	Numero canale (x)	Channel number (x)	1	8	1	1	① 7682H + (n -1) * 80H
P11. n.03	Stato a riposo	Idle status	0	1	0	1	① 7684H + (n -1) * 80H
<b>M11</b>	<b>Matematica</b>	<b>Mathematics</b>					
P12.n.01	Visualizzazione variabile matematica	Enable viewing of math variable	0	1	0	1	①7A80 + (n -1) * 80H
P12.n.02	Descrizione della variabile matematica	Description of math variable			MATn	8	①7A82 + (n -1) * 80H
P12.n.03	Unità di misura	Unit of measure			UM	3	①7A92 + (n -1) * 80H
P12.n.04	Operando 1	Operand 1	0	64	OFF	1	①7AA2 + (n -1) * 80H
P12.n.05	Operatore 1	Operator 1	0	4	0	1	①7AA4 + (n -1) * 80H
P12.n.06	Operando 2	Operand 2	0	64	OFF	8	①7AA6 + (n -1) * 80H
P12.n.07	Operatore 2	Operator 2	0	4	0	1	①7AA8 + (n -1) * 80H
P12.n.08	Operando costante	Constant operand	-9999	9999	0	1	①7AAA + (n -1) * 80H
P12.n.09	Moltiplicatore costante	Constant multiplier	0	6	3	1	①7AAC + (n -1) * 80H

#### ① ESEMPIO

- Se si vuole scrivere/leggere il registro STATUS dell'ingresso 1 (P10.1.02), bisogna utilizzare la formula:  
 $6E82H (n -1) * 80H$   
dove n = 1 quindi:  
 $6E82H (1 -1) * 80H = 6E82H$
- Se si vuole scrivere/leggere il registro STATO A RIPOSO dell'uscita 2 (P11.2.03), bisogna utilizzare la formula :  
 $7684H + (n -1) * 80H$   
dove n = 2 quindi:  
 $7684H + (2 -1) * 80H = 7704H$
- Se si vuole scrivere/leggere il registro STATO A RIPOSO dell'uscita 8 (P11.8.01), bisogna utilizzare la formula:  
 $7684H + (n -1) * 80H$   
dove n = 8 quindi:  
 $7684H + (8 -1) * 80H = 7A04H$

#### ① EXAMPLE

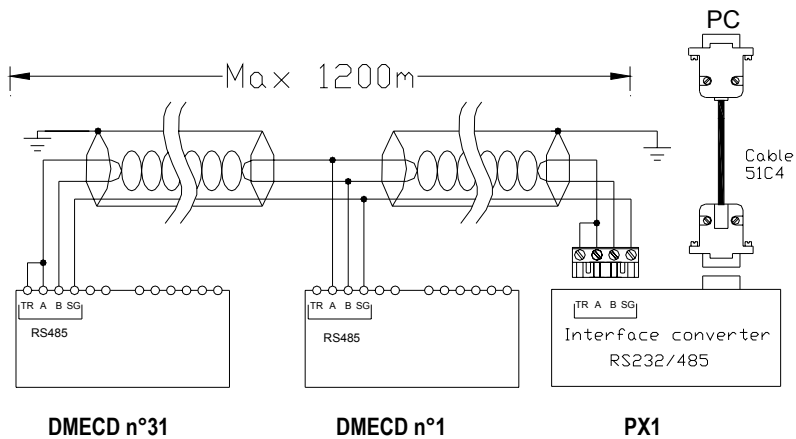
- If you want to read/write the STATUS register of input 1 (P10.1.02), you must use the formula:  
 $6E82H (n -1) * 80H$   
where n = 1 thus:  
 $6E82H (1 -1) * 80H = 6E82H$
- If you want read/write to the IDLE STATUS register of pulse 2 (P11.2.03), you must use the formula:  
 $7684H + (n -1) * 80H$   
where n = 2 thus:  
 $7684H + (2 -1) * 80H = 7704H$
- If you want read/write to the IDLE STATUS register of pulse 8 (P11.8.03), you must use the formula:  
 $7684H + (n -1) * 80H$   
where n = 8 thus:  
 $7684H + (8 -1) * 80H = 7A04H$

TABELLA 9:  
RTC  
(Utilizzabili con funzioni 04 e 06)

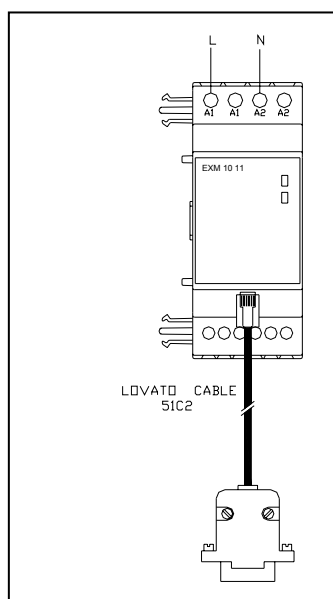
TABLE 9:  
RTC  
(To be used with functions 04 and 06)

Indirizzo Address	WORDS	MISURA	MEASURE	UNITA' UNIT	FORMATO FORMAT
28F0	1	Anno	Year		Unsigned int
28F1	1	Mese	Month		Unsigned int
28F2	1	Giorno	Day		Unsigned int
28F3	1	Ora	Hour		Unsigned int
28F4	1	Minuti	Minutes		Unsigned int
28F5	1	Secondi	Seconds		Unsigned int

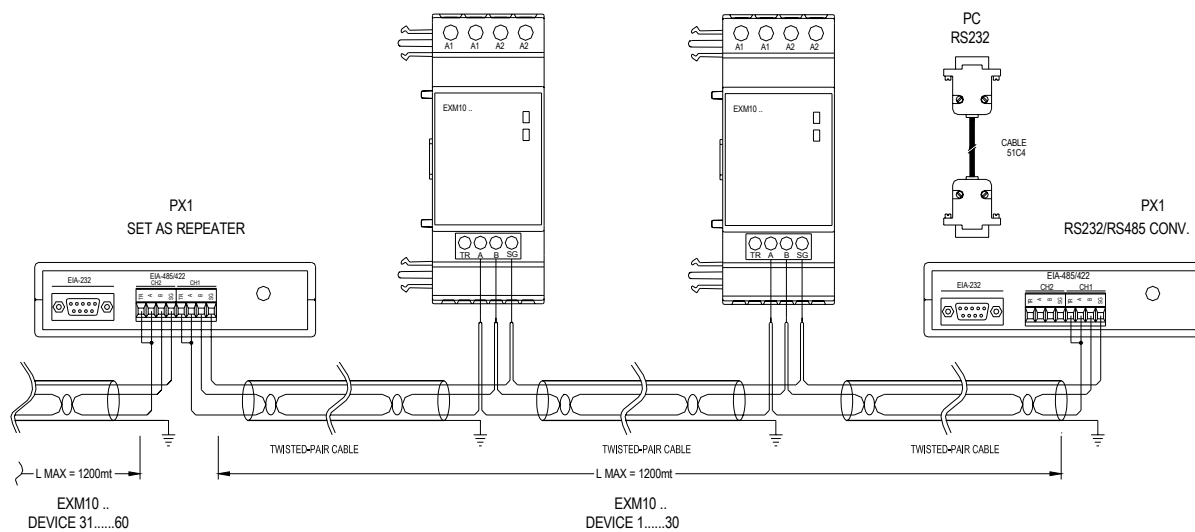
• DMECD



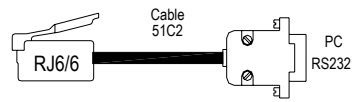
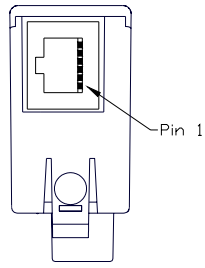
• EXM 10 11 RS 232



• EXM 10 12 RS485



- EXP 10 11 RS 232



- EXP 10 12 RS 485

