

PROTOCOLLO MODBUS®

La centralina di rifasamento DCRG8 supporta i protocolli di comunicazione Modbus RTU® e Modbus ASCII® sulla porta seriale RS-485.

La centralina di rifasamento DCRG8 supporta i protocolli di comunicazione Modbus RTU® e Modbus ASCII® sui moduli di espansione:

- EXP 10 10 USB
- EXP 10 11 RS232
- EXP 10 12 RS485
- EXP 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 (DCRJ 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 M16.

E' possibile configurare 2 moduli di espansione (n=1..2).

MENU M16 – COMUNICAZIONE SERIALE

| PAR | Funzione | Range | Default |
|----------|------------------------|---|---------------|
| P16.n.01 | Indirizzo | 1 ..245 | 1 |
| P16.n.02 | Velocità RS-232 (baud) | 1200 2400 4800 9600 19200 38400 57600 115200 | 9600 baud |
| P16.n.03 | Formato dati | 8 bit Nessuna 8 bit Dispari 8 bit Pari 7 bit Dispari 7 bit Pari | 8 bit Nessuna |
| P16.n.04 | Stop bit | 1 2 | 1 |
| P16.n.05 | Protocollo | Modbus RTU Modbus ASCII Modbus TCP | Modbus RTU |

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

| PAR | Funzione | Range | Default |
|----------|--------------|------------------------------------|-----------------|
| P16.n.06 | Indirizzo IP | 000.000.000.000 255.255.255.255 | 000.000.000.000 |
| P16.n.07 | Subnet MASK | 000.000.000.000 255.255.255.255 | 000.000.000.000 |
| P16.n.08 | TCP-IP Port | 0 - 9999 | 1001 |

MODBUS® PROTOCOL

The digital power factor controller DCRG8 supports the communication protocols Modbus RTU® and Modbus ASCII® on the RS-485 serial port.

The digital power factor controller DCRG8 support the communication protocols Modbus RTU® and Modbus ASCII® on the expansion modules:

- EXP 10 10 USB
- EXP 10 11 RS232
- EXP 10 12 RS485
- EXP 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 (DCRJ 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 M16 menu:

It is possible to configure 2 different expansion modules (n=1..2).

MENU M16 – SERIAL COMMUNICATION

| PAR | Function | Range | Default |
|----------|------------------|---|--------------|
| P16.n.01 | Address | 1 ..245 | 1 |
| P16.n.02 | RS-232 Baud Rate | 1200 2400 4800 9600 19200 38400 57600 115200 | 9600 baud |
| P16.n.03 | Data format | 8 bit None 8 bit Odd 8 bit Even 7 bit Odd 7 bit Even | 8 bit None |
| P16.n.04 | Stop bit | 1 2 | 1 |
| P16.n.05 | Protocol | Modbus RTU Modbus ASCII Modbus TCP | Modbus RTU |

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

| PAR | Function | Range | Default |
|----------|-------------|------------------------------------|-----------------|
| P16.n.06 | IP Address | 000.000.000.000 255.255.255.255 | 000.000.000.000 |
| P16.n.07 | Subnet MASK | 000.000.000.000 255.255.255.255 | 000.000.000.000 |
| P16.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 DCRG8, la lunghezza massima consentita per il campo dati è di :

80 registri da 16 bit (160 bytes)

- 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 DCRG8 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:

| | |
|-------------------------------|---|
| 03 = Read input register | Consente la lettura delle misure disponibili nel DCRG8 |
| 04 = Read input register | Consente la lettura delle misure disponibili nel DCRG8. |
| 06 = Preset single register | Permette la scrittura dei parametri |
| 07 = Read exception | Permette di leggere lo stato dell'apparecchio |
| 10 = Preset multiple register | Permette la scrittura di più parametri |
| 17 = Report slave ID | Permette di leggere informazioni relative all'apparecchio |

Per esempio, se si vuole leggere dal DCRG8 con indirizzo 01 il valore del cos phi totale che si trova alla locazione 0 (0 Hex), il messaggio da spedire è il seguente:

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 01 | 04 | FF | FF | 00 | 02 | 71 | EF |
|----|----|----|----|----|----|----|----|

Dove:

- 01= indirizzo slave
- 04 = funzione di lettura locazione
- FF FF = indirizzo della locazione diminuito di un'unità, contenete il valore delcos phi totale.
- 00 02 = numero di registri da leggere a partire dall'indirizzo 0
- 71 EF = 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 DCRG8 series, the maximum length for the data field is:

80 16-bit registers (160 bytes)

- 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 DCRG8 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:

| | |
|-------------------------------|--|
| 03 = Read input register | Allows to read the DCRG8 measures. |
| 04 = Read input register | Allows to read the DCRG8 measures. |
| 06 = Preset single register | Allows writing parameters |
| 07 = Read exception | Allows to read the device status |
| 10 = Preset multiple register | Allows writing several parameters |
| 17 = Report slave ID | Allows to read information about the device. |

For instance, to read the value of total cos phi , which resides at location 0 (0 Hex) from the DCRG8 with serial address 01, the message to send is the following:

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 01 | 04 | FF | FF | 00 | 02 | 71 | EF |
|----|----|----|----|----|----|----|----|

Whereas:

- 01= slave address
- 04 = Modbus® function 'Read input register'
- FF FF = Address of the required register (total cos phi) decreased by one
- 00 02 = Number of registers to be read beginning from address 0
- 71 EF = CRC Checksum

La risposta del DCRG8 è la seguente:

| | | | | | | | | |
|----|----|----|----|----|----|----|----|----|
| 01 | 04 | 04 | 00 | 00 | 03 | B4 | FB | 03 |
|----|----|----|----|----|----|----|----|----|

Dove:

01= indirizzo del DCRG8 (Slave 01)

04 = funzione richiesta dal Master

04 = numero di byte inviati dal DCRG8

00 00 03 B4 = valore esadecimale cos phi totale

= 948 = 0.948

FB 03 = 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 DCRG8 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 |
| LSB CRC | C1h |
| MSB 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 |
| LSB CRC | 5Eh |
| MSB 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 DCRG8 answer is the following:

| | | | | | | | | |
|----|----|----|----|----|----|----|----|----|
| 01 | 04 | 04 | 00 | 00 | 03 | B4 | FB | 03 |
|----|----|----|----|----|----|----|----|----|

Where:

01 = DCRG8 address (Slave 01)

04 = Function requested by the master

04 = Number of bytes sent by the DCRG8

00 00 03 B4 = Hex value of total cos phi

=948= 0.948

FB 03 = 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 DCRG8 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 |
| LSB CRC | C1h |
| MSB 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 |
| LSB CRC | 5Eh |
| MSB 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 DCRG8 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 |
| LSB CRC | 31h |
| MSB 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 |
| LSB CRC | 47h |
| MSB CRC | B2h |

La tabella seguente riporta il significato del byte inviato dal DCRG8 come risposta:

| BIT | SIGNIFICATO |
|-----|--------------------|
| 0 | |
| 1 | Modo operativo MAN |
| 2 | Modo operativo AUT |
| 3 | TEST utente |
| 4 | In errore |

FUNZIONE 17: REPORT SLAVE ID

Questa funzione permette di identificare il tipo di strumento.

Richiesta Master:

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

FUNCTION 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 DCRG8 will answer with an error message. In the same way, if the parameter address is not recognised, the DMG 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 |
| LSB CRC | 31h |
| MSB 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.

FUNCTION 07: READ EXCEPTION STATUS

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

Master query:

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

The following table gives the meaning of the status byte sent by the DCRG8 as answer:

| BIT | MEANING |
|-----|--------------------|
| 0 | |
| 1 | Operative mode MAN |
| 2 | Operative mode AUT |
| 3 | User TEST |
| 4 | Error on |

FUNZIONE 17: REPORT SLAVE ID

This function allows to identify the device type.

Master query:

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

Risposta Slave:

| | |
|------------------------------|------|
| Indirizzo slave | 08h |
| Funzione | 11h |
| Contatore bytes | 04 h |
| Dato 1 (Tipo) ❶ | 82h |
| Dato 2 (Revisione software) | 04h |
| Dato 3 (Revisione hardware) | 00h |
| Dato 4 (Revisione parametri) | 01h |
| LSB CRC | ...h |
| MSB CRC | ...h |

❶ 48h = DCRG8

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 |
| LSB CRC | 85h |
| MSB 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 |
| LSB CRC | 1Bh |
| MSB CRC | 51h |

Slave response:

| | |
|--------------------------------|------|
| Slave address | 08h |
| Function | 11h |
| Byte count | 04 h |
| Data 01 -Type ❶ | 82h |
| Data 02 - (Sw revision) | 04h |
| Data 03 - (Hardware revision) | 00h |
| Data 04 - (Parameter revision) | 01h |
| LSB CRC | ...h |
| MSB CRC | ...h |

❶ 48h = DCRG8

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 |
| LSB CRC | 85h |
| MSB 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 |
| LSB CRC | 1Bh |
| MSB 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 e' effettuata a tempo ma con dei caratteri di ritorno a capo.

Se si seleziona il parametro P16.x.05 o P16.05 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 e' 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 all' indirizzo 2112h il valore dello stato del terzo step (indice 2), il messaggio da spedire è il seguente:

| | | | | | | | | |
|---|----|----|----|----|----|----|----|------|
| : | 08 | 04 | 21 | 11 | 00 | 01 | C1 | CRLF |
|---|----|----|----|----|----|----|----|------|

Dove:

- : = ASCII 3Ah = Delimitatore inizio messaggio
- 08 = indirizzo slave.
- 04 = funzione di lettura locazione.
- 21 11 = indirizzo della locazione diminuito di un'unità, dello stato del terzo step (indice 2)
- 00 01 = numero di registri da leggere a partire dall'indirizzo 2112.
- C1 = checksum LRC.
- CRLF = ASCII 0Dh 0Ah = delimitatore fine messaggio

La risposta del DCRG8 è la seguente:

| | | | | | | | |
|---|----|----|----|----|----|----|----------|
| : | 08 | 04 | 04 | 00 | 01 | F1 | CR LF |
|---|----|----|----|----|----|----|----------|

Dove:

- : = ASCII 3Ah = Delimitatore inizio messaggio
- 08 = indirizzo del DCRG8 (Slave 08).
- 04 = funzione richiesta dal Master.
- 04 = numero di byte inviati dallo slave.
- 00 01 = stato dello step (1 = inserito).
- F1 = 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 P16.x.05 or P16.05 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 status of third step (index 2), which resides at location 2112h from the slave with serial address 08, the message to send is the following:

| | | | | | | | | |
|---|----|----|----|----|----|----|----|------|
| : | 08 | 04 | 21 | 11 | 00 | 01 | C1 | CRLF |
|---|----|----|----|----|----|----|----|------|

Whereas:

- : = ASCII 3Ah message start delimiter
- 08 = slave address
- 04 = Modbus® function 'Read input register'
- 21 11 = Address of the required register (L3 current phase) decreased by one
- 00 01 = Number of registers to be read beginning from address 04
- C1 = LRC Checksum
- CRLF = ASCII 0Dh 0Ah = Message end delimiter

The DCRG8 answer is the following:

| | | | | | | | |
|---|----|----|----|----|----|----|----------|
| : | 08 | 04 | 04 | 00 | 01 | F1 | CR LF |
|---|----|----|----|----|----|----|----------|

Whereas:

- : = ASCII 3Ah message start delimiter
- 08 = DCRG8 address (Slave 08)
- 04 = Function requested by the master
- 04 = Number of bytes sent by the multimeter
- 00 01 = step status (1 = inserted)
- F1 = LRC checksum
- CRLF = ASCII 0Dh 0Ah = Message end delimiter

TABELLA 2:
MISURE FORNITE DAL PROTOCOLLO DI COM.
(Utilizzabili con funzioni 03 e 04)

| Indirizzo Address | WORDS | MISURA | MEASURE | UNITA' UNIT | FORMATO FORMAT | From Fw.Rev. |
|----------------------|-------|-----------------------------------|--------------------------------|----------------|-------------------|-----------------|
| | | MISURA ISTANTANEA (IN) | INSTANTANEOUS MEASURE (IN) | | | |
| 0000H | 2 | ① Cos phi / Cos phi totale | ① Cos phi / Total Cos phi | Value/1000 | Signed long | 4 |
| 0002H | 2 | ① Sen phi / ND | ① Sen phi / NA | Value/1000 | Signed long | 4 |
| 0004H | 2 | ① Tan phi / ND | ① Tan phi / NA | Value/1000 | Signed long | 4 |
| 0006H | 2 | ① Tensione / Tensione Equivalente | ① Voltage / Equivalent Voltage | V/10 | Signed long | 4 |
| 0008H | 2 | ① Corrente / Corrente Equivalente | ① Current / Equivalent Current | A/1000 | Signed long | 4 |
| 000AH | 2 | ① Potenza Reattiva / ND | ① Reactive Power / NA | Var | Signed long | 4 |

TABLE 2:
MEASURES SUPPLIED BY SERIAL COMMUNICATION PROTOCOL
(To be used with functions 03 and 04)

| | | | | | | |
|-------|---|------------------------------------|----------------------------------|------------|---------------|---|
| 000CH | 2 | ① ND / Potenza Reattiva 1 | ① NA / Reactive Power 1 | Var | Signed long | 4 |
| 000EH | 2 | ① ND / Potenza Reattiva 2 | ① NA / Reactive Power 2 | Var | Signed long | 4 |
| 0010H | 2 | ① ND / Potenza Reattiva 3 | ① NA / Reactive Power 3 | Var | Signed long | 4 |
| 0012H | 2 | ① Delta Var / Delta Var | ① Delta Var / Delta Var | Var | Signed long | 4 |
| 0016H | 2 | Energia attiva totale importata | Total imported active energy | kWh/10 | Unsigned long | 4 |
| 0018H | 2 | Energia reattiva induttiva totale | Total inductive reactive energy | kVarh/10 | Unsigned long | 4 |
| 001AH | 2 | Energia apparente totale | Total apparent energy | kVA/10 | Unsigned long | 4 |
| 001CH | 2 | Energia attiva totale esportata | Total exported active energy | kWh/10 | Unsigned long | 4 |
| 001EH | 2 | Energia reattiva capacitiva totale | Total capacitive reactive energy | kVarh/10 | Unsigned long | 4 |
| 0040H | 2 | ① ND / Tensione L1 | ① NA / L1 Voltage | V/10 | Signed long | 4 |
| 0042H | 2 | ① ND / Tensione L2 | ① NA / L2 Voltage | V/10 | Signed long | 4 |
| 0044H | 2 | ① ND / Tensione L3 | ① NA / L3 Voltage | V/10 | Signed long | 4 |
| 0046H | 2 | ① ND / Corrente L1 | ① NA / L1 Current | A/1000 | Signed long | 4 |
| 0048H | 2 | ① ND / Corrente L2 | ① NA / L2 Current | A/1000 | Signed long | 4 |
| 004AH | 2 | ① ND / Corrente L3 | ① NA / L3 Current | A/1000 | Signed long | 4 |
| 004CH | 2 | ① ND / Tensione L1-L2 | ① NA / L1-L2 Voltage | V/10 | Signed long | 4 |
| 004EH | 2 | ① ND / Tensione L2-L3 | ① NA / L2-L3 Voltage | V/10 | Signed long | 4 |
| 0050H | 2 | ① ND / Tensione L3-L1 | ① NA / L3-L1 Voltage | V/10 | Signed long | 4 |
| 0052H | 2 | ① ND / Cos phi 1 | ① NA / Cos phi 1 | Value/1000 | Signed long | 4 |
| 0054H | 2 | ① ND / Cos phi 2 | ① NA / Cos phi 2 | Value/1000 | Signed long | 4 |
| 0056H | 2 | ① ND / Cos phi 3 | ① NA / Cos phi 3 | Value/1000 | Signed long | 4 |
| 0058H | 2 | ① ND / Sen phi 1 | ① NA / Sen phi 1 | Value/1000 | Signed long | 4 |
| 005AH | 2 | ① ND / Sen phi 2 | ① NA / Sen phi 2 | Value/1000 | Signed long | 4 |
| 005CH | 2 | ① ND / Sen phi 3 | ① NA / Sen phi 3 | Value/1000 | Signed long | 4 |
| 005EH | 2 | ① ND / Tan phi 1 | ① NA / Tan phi 1 | Value/1000 | Signed long | 4 |
| 0060H | 2 | ① ND / Tan phi 2 | ① NA / Tan phi 2 | Value/1000 | Signed long | 4 |
| 0062H | 2 | ① ND / Tan phi 3 | ① NA / Tan phi 3 | Value/1000 | Signed long | 4 |
| 1500H | 2 | ① Potenza Attiva / ND | ① Active Power / NA | W | Unsigned long | 4 |
| 1502H | 2 | ① ND / Potenza Attiva 1 | ① NA / Active Power 1 | W | Unsigned long | 4 |
| 1504H | 2 | ① ND / Potenza Attiva 2 | ① NA / Active Power 2 | W | Unsigned long | 4 |
| 1506H | 2 | ① ND / Potenza Attiva 3 | ① NA / Active Power 3 | W | Unsigned long | 4 |
| 1508H | 2 | ① Potenza Apparente / ND | ① Apparent Power / NA | VA | Unsigned long | 4 |
| 150AH | 2 | ① ND / Potenza Apparente 1 | ① NA / Apparent Power 1 | VA | Unsigned long | 4 |
| 150CH | 2 | ① ND / Potenza Apparente 2 | ① NA / Apparent Power 2 | VA | Unsigned long | 4 |
| 150EH | 2 | ① ND / Potenza Apparente 3 | ① NA / Apparent Power 3 | VA | Unsigned long | 4 |

Nota ①: Se la centralina è cablata e programmata con una tensione ed una corrente verrà restituita la prima misura; Se invece sono disponibili 3 tensioni e 3 correnti verrà restituita la seconda misura.

Note ②: If the device has been wired and programmed with one voltage and one current inputs, then the first measurement will be returned; Otherwise, if three voltages and three current are available, the second measurement will be returned.

| Indirizzo Address | WORDS | MISURA | MEASURE | UNITA' UNIT | FORMATO FORMAT | From Fw.Rev. |
|----------------------------------|-------|-------------------------------|-----------------------------|---|----------------|--------------|
| 1300H + (2 * i), 0 <= i <= 31 | 2 | Potenza reattiva step i | Reactive Power Step i | kVar/100 | Unsigned long | 4 |
| 1200H + (2 * i), 0 <= i <= 31 | 2 | Tempo inserzione step i | Insertion Time Step i | Sec | Unsigned long | 4 |
| 1100H + (2 * i), 0 <= i <= 31 | 2 | Numero inserzioni step i | Insertion Count Step i | - | Unsigned long | 4 |
| 2001H | 1 | Flag_CAP-IND Corrente / ND | Current CAP-IND Flag / ND | 0 = CAP / 1 = IND | Unsigned short | 4 |
| 2002H | 1 | ND / Flag_CAP-IND Corrente 1 | ND / Current 1 CAP-IND Flag | 0 = CAP / 1 = IND | Unsigned short | 4 |
| 2003H | 1 | ND / Flag_CAP-IND Corrente 2 | ND / Current 2 CAP-IND Flag | 0 = CAP / 1 = IND | Unsigned short | 4 |
| 2004H | 1 | ND / Flag_CAP-IND Corrente 3 | ND / Current 3 CAP-IND Flag | 0 = CAP / 1 = IND | Unsigned short | 4 |
| 2005H | 1 | PF medio settimanale | Average weekly PF | /1000 | Unsigned short | 4 |
| 2110H + (1 * i), 0 <= i <= 31 | 1 | Stato Step | Step Status | 0 = step not ins. 1 = step ins. 3 = step moving 2 = step not config. | Unsigned short | 4 |
| 202AH | 1 | Temperatura °C/°F | Temperature °C/°F | Value/10 | Signed short | 4 |
| 202BH | 1 | Temperatura massima °C/°F | Max Temperature °C/°F | Value/10 | Signed short | 4 |
| 2C00H + (1 * i), 0 <= i <= 29 | 1 | (i+2) armonica di tensione L1 | (i+2)° L1 harmonic voltage | Value/10 | Unsigned short | 4 |
| 2C20H | 1 | THD di Tensione L1 | L1 voltage THD | Value/10 | Unsigned short | 4 |
| 2C30H + (1 * i), 0 <= i <= 29 | 1 | (i+2) armonica di tensione L2 | (i+2)° L2 harmonic voltage | Value/10 | Unsigned short | 4 |
| 2C50H | 1 | THD di Tensione L2 | L2 voltage THD | Value/10 | Unsigned short | 4 |
| 2C60H + (1 * i), 0 <= i <= 29 | 1 | (i+2) armonica di tensione L3 | (i+2)° L3 harmonic voltage | Value/10 | Unsigned short | 4 |
| 2C80H | 1 | THD di Tensione L3 | L3 voltage THD | Value/10 | Unsigned short | 4 |

| Indirizzo Address | WORDS | MISURA | MEASURE | UNITA' UNIT | FORMATO FORMAT | From Fw.Rev. |
|----------------------------------|-------|-------------------------------|----------------------------|-------------|------------------|--------------|
| | | MISURA ISTANTANEA (IN) | INSTANTANEOUS MEASURE (IN) | | | 4 |
| 2C90H + (1 * i), 0 <= i <= 29 | 1 | (i+2) armonica di corrente L1 | (i+2)° L1 harmonic current | Value/10 | Unsigned integer | 4 |
| 2CB0H | 1 | THD di corrente L1 | L1 current THD | Value/10 | Unsigned integer | 4 |
| 2CC0H + (1 * i), 0 <= i <= 29 | 1 | (i+2) armonica di corrente L2 | (i+2)° L2 harmonic current | Value/10 | Unsigned integer | 4 |
| 2CE0H | 1 | THD di corrente L2 | L2 current THD | Value/10 | Unsigned integer | 4 |

| | | | | | | |
|----------------------------------|---|-----------------------------------|-------------------------------|----------|------------------|---|
| 2CF0H + (1 * i), 0 <= i <= 29 | 1 | (i+2)° Armonica di corrente L3 | (i+2)° L3 harmonic current | Value/10 | Unsigned integer | 4 |
| 2D10H | 1 | THD di corrente L3 | L3 current THD | Value/10 | Unsigned integer | 4 |
| 2D20H + (1 * i), 0 <= i <= 29 | 1 | (i+2)° armonica di tensione L1-L2 | (i+2)° L1-L2 harmonic voltage | Value/10 | Unsigned integer | 4 |
| 2D40 | 1 | THD di Tensione L1-L2 | L1-L2 voltage THD | Value/10 | Unsigned integer | 4 |
| 2D50H + (1 * i), 0 <= i <= 29 | 1 | (i+2)° armonica di tensione L2-L3 | (i+2)° L2-L3 harmonic voltage | Value/10 | Unsigned integer | 4 |
| 2D70H | 1 | THD di Tensione L2-L3 | L2-L3 voltage THD | Value/10 | Unsigned integer | 4 |
| 2D80 + (1 * i), 0 <= i <= 29 | 1 | (i+2)° armonica di tensione L3-L1 | (i+2)° L3-L1 harmonic voltage | Value/10 | Unsigned integer | 4 |
| 2DA0H | 1 | THD di Tensione L3-L1 | L3-L1 voltage THD | Value/10 | Unsigned integer | 4 |
| 1A80H | 2 | Contatore 1 | Counter 1 | /1 | Unsigned long | 4 |
| 1A82H | 2 | Contatore 2 | Counter 2 | /1 | Unsigned long | 4 |
| 1A84H | 2 | Contatore 3 | Counter 3 | /1 | Unsigned long | 4 |
| 1A86H | 2 | Contatore 4 | Counter 4 | /1 | Unsigned long | 4 |
| 1A88H | 2 | Contatore 5 | Counter 5 | /1 | Unsigned long | 4 |
| 1A8AH | 2 | Contatore 6 | Counter 6 | /1 | Unsigned long | 4 |
| 1A8CH | 2 | Contatore 7 | Counter 7 | /1 | Unsigned long | 4 |
| 1A8EH | 2 | Contatore 8 | Counter 8 | /1 | Unsigned long | 4 |

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 | From Fw. Rev. |
|-------------------|-------|--------------------------------------|----------------------------|--------------|----------------|---------------|
| 2FF0H | 1 | Azzerà energia parziale | Reset Partial Energy | 1 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà contatori | Reset External counter | 2 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà limiti | Reset Limits | 3 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà temp. massima | Reset max temperatue | 4 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà sovraccarico condensatori | Reset capacitor overload | 5 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà tempi inserzione condensatori | Reset step insertion time | 6 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà manovre condensatori | Reset step insertion count | 7 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà Step trimming | Reset Step trimming | 8 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà energia totale | Reset Total Energy | 9 | Unsigned int | 4 |
| 2FF0H | 1 | Ripristina modalità test | Restore test mode | 10 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà memoria eventi | Reset events log | 11 | Unsigned int | 4 |
| 2FF0H | 1 | Setup a default ❶ | Setup to default ❶ | 12 | Unsigned int | 4 |
| 2FF0H | 1 | Salva copia setup | Backup parameters | 13 | Unsigned int | 4 |
| 2FF0H | 1 | Ripristina setup ❶ | Restore parameters ❶ | 14 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà TPF settimanale | Reset week TPF | 15 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà manutenzione 1 | Reset maintenance 1 | 16 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà manutenzione 2 | Reset maintenance 2 | 17 | Unsigned int | 4 |
| 2FF0H | 1 | Azzerà manutenzione 3 | Reset maintenance 3 | 18 | Unsigned int | 4 |

❶ **ATTENZIONE**

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

❶ **ATTENTION**

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

TABELLA 4:
EVENTI

TABLE 4:
EVENTS

| Indirizzo Address | WORDS | MISURA | MEASURE | UNITA' UNIT | FORMATO FORMAT |
|--|-------|--|--|-------------|------------------|
| 5030H | 1 | PUNTATORE EVENTI Indica l'ultimo evento registrato (LSB) / CONTATORE EVENTI Indica il numero totale eventi (MSB) | EVENTS POINTER Last event stored (LSB)/ EVENTS COUNTER Total events stored (MSB) | | Unsigned integer |
| 5032H | 43 | Descrizione Evento nella lingua corrente | Event description using current language | | Unsigned integer |
| Procedura per la lettura degli eventi 1 - Lettura di contatore e puntatore eventi con funzione 04 dall'indirizzo 5030H 2 - Il contatore è nel MSB del valore restituito, il puntatore nel LSB 3 - Scrittura con funzione 06 del numero di evento desiderato, scrivendolo all'indirizzo 5030H 4 - Lettura del testo dell'evento con funzione 04, dall'indirizzo 5032H per 43 registri 5 - Vengono ritornati 86 byte: i primi 10 sono la data, 11=separator, dal 12 fino al 19 = ora, 20=separator, dal 21.mo alla fine = testo evento Dati: cnt = numero di eventi da scaricare, pnt = puntatore all'ultimo evento | | | Procedure for events reading 1 - Reading of event counter and pointer with function 04 from address 5030H 2 - The counter is in the MSB of the return value, the pointer to the LSB 3 - Writing with 06 function of the number of desired event, writing to the address 5030H 4 - Read of the text of the event with function 04, 43 registers from address 5032H 5 - 86 bytes are returned: the first 10 are the date, 11 = separator, from 12 to 19 = hour, 20 = separator, from 21.st to the end = event text Given: cnt = number of events to download, pnt = pointer to the last event | | |
| <ul style="list-style-type: none"> Dopo reset si parte da cnt= 0 , pnt = 0 Quando si verificano degli eventi si incrementano sia pnt che cnt Quando cnt arriva al valore max (250) il contatore non si incrementa più e il puntatore riparte da 1 Quando si vuole leggere l'evento più giovane, richiedere il record nr. 1 (write al 5030 scrivendo 1) Quando si vuole leggere l'evento più vecchio, richiedere il record nr. cnt (write 5030 del valore di cnt) | | | <ul style="list-style-type: none"> After reset, start from cnt = 0 , pnt = 0 When events occur that will increase both pnt and cnt When cnt reaches the maximum value (250), the counter does not increment the pointer over and starts again from 1 When you want to read the younger event, ask for the record no. 1 (by writing 1 to 5030) When you want to read the oldest event, ask for the record no. cnt (write 5030 the value of cnt) | | |

TABELLA 5:
EVENTI:

Il numero massimo eventi memorizzati è 250 (buffer circolare).

TABLE 5:
EVENTS:

The max events number stored is 250 (circular buffer).

| Indirizzo Address | CLASSE EVENTO | TIPO EVENTO | SORGENTE SOURCE | EVENT CLASS | EVENT TYPE |
|----------------------|----------------|--|--------------------|---------------|---|
| 0 | POWER | 0- POWER ON 1- POWER DOWN 2- REBOOT | | POWER | 0- POWER ON 1- POWER DOWN 2- REBOOT |
| 2 | ALLARME | 0- INIZIO ALLARME 1- FINE ALLARME 2- RESET ALLARME | | ALLARM | 0- ALARM BEGIN 1- ALARM END 2- ALARM RESET |
| 3 | LIMITE | 0- LIMITE ON 1- LIMITE OFF | | LIMIT | 0- LIMIT ON 1- LIMIT OFF |
| 4 | REMOTO | 0- COMANDO REMOTO ON 1- COMANDO REMOTO OFF | | REMOTE | 0- REMOTE COMMAND ON 1- REMOTE COMMAND OFF |
| 5 | COMUNICAZIONE | 0- COMUNICAZIONE ON 1- COMUNICAZIONE OFF | | COMMUNICATION | 0- COMMUNICATION ON 1- COMMUNICATION OFF |
| 7 | SETUP | 0- MENU PARAMETRI 2- MPOSTAZIONE OROLOGIO | | SETUP | 0- PARAMETERS MENU 2- CLOCK SETUP |
| 8 | MENU COMANDI | 0- AZZERA ENERGIA PARZIALE 1- AZZERA CONTATORI 2- AZZERA LIMITI 3- AZZERA TEMP. MASSIMA 4- AZZERA STEP OVERLOAD 5- AZZERA TEMPI INSERZIONE STEP 6- AZZERA MANOVRE CONDENSATORI 7- AZZERA STEP TRIMMING 8- AZZERA ENERGIA TOTALE 9- RIPRISTINA MODO TEST 10- AZZERA EVENTI 11- SETUP A DEFAULT 12- SALVA COPIA SETUP 13- RIPRISTINA SETUP 14- AZZERA TPF SETTIMANALE 15- AZZERA MANUTENZIONE 1 16- AZZERA MANUTENZIONE 2 17- AZZERA MANUTENZIONE 3 | | COMMAND MENU | 0- RESET PARTIAL ENERGY 1- RESET COUNTERS 2- RESET LIMITS 3- RESET MAX TEMP. 4- RESET STEP OVERLOAD 5- RESET STEP INSERTION TIME 6- RESET STEP INSERTION COUNT 7- RESET STEP TRIMMING 8- RESET TOTAL ENERGY 9- RESTORE TEST MODE 10- RESET EVENTS 11- SETUP TO DEFAULT 12- SAVE SETUP 13- RESTORE SETUP 14- RESET WEEKLY TPF 15- RESET SERVICE 1 16- RESET SERVICE 2 17- RESET SERVICE 3 |
| 9 | PASSWORD | 0- PASSWORD UTENTE 1- PASSWORD AMMINISTRATORE | | PASSWORD | 0- USER PASSWORD 1- ADMINISTARTOR PASSWORD |
| 10 | CONFIGURAZIONE | 0- CAMBIO CONFIGURAZIONE MODULI | | CONFIGURATION | 0- CHANGE MODULE CONFIGURATION |

TABELLA 6:
STATO ALLARMI

TABLE 6:
ALARM STATUS

| Indirizzo Address | WORDS | MISURA | MEASURE | UNITA' UNIT | FORMATO FORMAT | From Fw.Rev. |
|----------------------|-------|---------------|--------------|----------------|-------------------|-----------------|
| 1400H | 2 | STATO ALLARMI | ALARM STATUS | | Unsigned long | 4 |

TABELLA 7:
CODIFICA ALLARMI

TABLE 7:
ALARM SCHEME

| Bit # | TIPO ALLARME | ALARM TYPE |
|----------|---|----------------------------------|
| 0 | A01 SOTTOCOMPENSAZIONE | A01 UNDER COMPENSATION |
| 1 | A02 SOVRACCOMPENSAZIONE | A02 OVER COMPENSATION |
| 2 | A03 CORRENTE IMPIANTO TROPPO BASSA | A03 LOW CURRENT |
| 3 | A04 CORRENTE IMPIANTO TROPPO ALTA | A04 HIGH CURRENT |
| 4 | A05 TENSIONE IMPIANTO TROPPO BASSA | A05 LOW VOLTAGE |
| 5 | A06 TENSIONE IMPIANTO TROPPO ALTA | A06 HIGH VOLTAGE |
| 6 | A07 TEMPERATURA IMPIANTO TROPPO ALTA | A07 OVERTEMPERATURE |
| 7 | A08 SOVRACCARICO CONDENSATORE | A08 CAPACITOR OVERLOAD |
| 8 | A09 MICROINTERRUZIONE | A09 NO-VOLTAGE RELEASE |
| 9 | A10 STEP DIFETTOSO | A10 STEP FAILURE |
| 10 | A11 EVENTO ARMONICO: CORRENTE TROPPO ALTA | A11 HARMONIC EVENT: HIGH CURRENT |
| 11 | A12 EVENTO ARMONICO: THD TROPPO ALTA | A12 HARMONIC EVENT: HIGH THD |
| 12 | A13 EVENTO ARMONICO: 5. ARMONICA | A13 HARMONIC EVENT: 5. HARM. |
| 13 | A14 EVENTO ARMONICO: 7. ARMONICA | A14 HARMONIC EVENT: 7. HARM. |
| 14 | A15 EVENTO ARMONICO: 11. ARMONICA | A15 HARMONIC EVENT: 11. HARM. |
| 15 | A16 EVENTO ARMONICO: 13. ARMONICA | A16 HARMONIC EVENT: 13. HARM. |
| 16 | A17 PROT. 1 ALTA TEMP. | A17 PROT. 1 HI TEMP. |
| 17 | A18 PROT. 2 ALTA TEMP. | A18 PROT. 2 HI TEMP. |
| 18 | A19 ERRORE COLLEGAMENTO | A19 LINK ERROR |
| 19 | UA1 ALLARME UTENTE 1 | UA1 USER ALARM 1 |
| 20 | UA2 ALLARME UTENTE 2 | UA2 USER ALARM 2 |

| | | |
|----|----------------------------|-----------------------|
| 21 | UA3 ALLARME UTENTE 3 | UA3 USER ALARM 3 |
| 22 | UA4 ALLARME UTENTE 4 | UA4 USER ALARM 4 |
| 23 | UA5 ALLARME UTENTE 5 | UA5 USER ALARM 5 |
| 24 | UA6 ALLARME UTENTE 6 | UA6 USER ALARM 6 |
| 25 | UA7 ALLARME UTENTE 7 | UA7 USER ALARM 7 |
| 26 | UA8 ALLARME UTENTE 8 | UA8 USER ALARM 8 |
| 27 | A01 MANUTENZIONE 1 SCADUTA | A01 SERVICE 1 EXPIRED |
| 28 | A02 MANUTENZIONE 2 SCADUTA | A02 SERVICE 2 EXPIRED |
| 29 | A03 MANUTENZIONE 3 SCADUTA | A03 SERVICE 3 EXPIRED |

TABELLA 8:
OROLOGIO DATARIO

TABLE 8:
REAL TIME CLOCK

| Indirizzo Address | WORDS | FUNZIONE | FUNCTION | RANGE | From Fw.Rev. |
|-------------------|-------|---|---|------------|--------------|
| 28F0H | 1 | Anno | Year | 2000..2099 | 4 |
| 28F1H | 1 | Mese | Month | 1-12 | 4 |
| 28F2H | 1 | Giorno | Day | 1-31 | 4 |
| 28F3H | 1 | Ora | Hours | 0-23 | 4 |
| 28F4H | 1 | Minuti | Minutes | 0-59 | 4 |
| 28F5H | 1 | Secondi | Seconds | 0-59 | 4 |
| 28FAH | 1 | Valore 01h: Salvataggio impostazione orologio datario | Value 01H: Save real time clock setting | - | 4 |

TABELLA 9:

TABLE 9:

| Indirizzo Address | WORDS | STATI | STATUS | From Fw.Rev. |
|-------------------|-------|--|--|--------------|
| 2F00H | 1 | Cambio modalit  operativa Valore 01h: Modalit  Manuale Valore 02h: Modalit  Automatica | Operative mode change Valore 01h: Manual Mode Valore 02h: Automatic Mode | 4 |
| 2F03H | 1 | Valore 01h: Salvataggio eeprom | Value 01h: Eeprom save | 4 |
| | | Valore 02h: Salvataggio Fram | Value 02h: Fram save | 4 |
| | | Valore 04h: Salvataggio fram e reboot | Value 04h: Fram save and reboot | 4 |
| | | Valore 08h: Salvataggio eeprom e fram | Value 08h: Eeprom, Fram save | 4 |
| 2F07H | 1 | Valore 00h: Reset apparecchio | Value 00h: Reset device | 4 |
| | | Valore 01h: Reset apparecchio con salvataggio in fram | Value 01h: Reset device and save Fram | 4 |
| 28FAH | 1 | Valore 01h: Salvataggio impostazione orologio datario | Value 01H: Save real time clock setting | 4 |

IMPOSTAZIONE PARAMETRI

Tramite il protocollo Modbus® e' possibile accedere ai parametri dei menu.

Per interpretare correttamente la corrispondenza fra valore numerico e funzione selezionata e/o unita' di misura, fare riferimento al manuale operativo del DCRG.

PROCEDURA PER LA LETTURA DEI PARAMETRI

1. Scrivere il valore del menu che si vuole leggere tramite la **funzione 6** all'indirizzo **5000H** ❶.
2. Scrivere il valore del sottomenu (se esiste) che si vuole leggere tramite la **funzione 6** all'indirizzo **5001H** ❶.
3. Scrivere il valore del parametro che si vuole leggere tramite la **funzione 6** all'indirizzo **5002H** ❶.
4. Eseguire la **funzione 4** all'indirizzo **5004H**, di un numero di registri appropriato alla lunghezza del parametro (vedi tabella).
5. Se si vuole leggere il parametro successivo, (all'interno dello stesso menu/sottomenu) ripetere il passo 4, altrimenti eseguire il passo 1.

PROCEDURA PER LA SCRITTURA DEI PARAMETRI

1. Scrivere il valore del menu che si vuole modificare tramite la **funzione 6** all'indirizzo **5000H** ❶.
2. Scrivere il valore del sottomenu (se esiste) che si vuole modificare tramite la **funzione 6** all'indirizzo **5001H** ❶.
3. Scrivere il valore parametro che si vuole modificare tramite la **funzione 6** all'indirizzo **5002H** ❶.
4. Eseguire la **funzione 16** all'indirizzo **5004H**, di un numero di registri appropriato alla lunghezza del parametro.
5. Se si vuole scrivere il parametro successivo, all'interno dello stesso menu/sottomenu ripetere il passo 4, altrimenti eseguire il passo 1, se non bisogna scrivere ulteriori parametri eseguire il passo 6.
6. Per rendere effettivo un cambiamento nel menu di setup e' necessario memorizzare i valori in EEPROM, utilizzando l'apposito comando descritto nella tabella 9.(scrivere il valore 01 con la **funzione 6** all' indirizzo **2F03H**) e poi successivamente il valore 04 sempre all'indirizzo **2F03H**.

| TIPO DI PARAMETRO | NUMERO REGISTRI |
|---|--------------------------|
| Testo lunghezza 6 caratteri (es. M21.01.06) | 3 registri (6 byte) |
| Testo lunghezza 16 caratteri (es. M21.01.05) | 8 registri (16 byte) |
| Testo lunghezza 20 caratteri (es. M01.9) | 10 registri (20 byte) |
| Valore numerico < 32768 (es M01.07) | 1 registri (2 byte) |
| Valore numerico > 32768 (es M02.07) | 2 registri (4 byte) |
| Indirizzo IP (es. M16.0x.06 M16.0x.07) | 2 registri (4 byte) |

❶ E' possibile leggere il valore del menu, sottomenu e parametro memorizzati agli indirizzi **5000H**, **5001H** e **5002H** utilizzando la **funzione 4**

Vedere esempio seguente

PARAMETER SETTING

Using the Modbus® protocol it is possible to access the menu parameters.

To correctly understand the correspondence between the numeric value and the selected function and/or the unit of measure, please see the DCRG operating manual.

PROCEDURE FOR THE READING OF PARAMETERS

1. Write the value of the menu that you want to read by using the **function 6** at address **5000H** ❶.
2. Write the value of the submenu (if it is present) that you want to read by using the **function 6** at address **5001H** ❶.
3. Write the value of the parameter that you want to read by using the **function 6** at address **5002H** ❶.
4. Perform the **function 4** at the address **5004H**, with a number of registers appropriate to the length of the parameter (see table).
5. If you want to read the next parameter (in the same menu/submenu) repeat step 4, otherwise perform step 1.

PROCEDURE FOR THE WRITING OF PARAMETERS

1. Write the value of the menu that you want to change by using the **function 6** at address **5000H** ❶.
2. Write the value of the submenu (if it is present) that you want to change by using the **function 6** at address **5001H** ❶.
3. Write the value of the parameter that you want to change by using the **function 6** at address **5002H** ❶.
4. Perform the **function 16** at address **5004H**, with a number of registers appropriate to the length of the parameter
5. If you want to write the next parameter, in the same menu / submenu repeat step 4, otherwise perform step 1, if you do not have to write additional parameters go to step 6.
6. To make effective the changes made to setup parameters it is necessary to store the values in EEPROM, using the dedicated command described in table 9.(write value 01 by using **function 6** at address **2F03H**) and then write value 04 again at address **2F03H**.

| TYPE OF PARAMETER | NUMBER OF REGISTER |
|--|---------------------------|
| Text length 6 characters (ex. M25.01.06) | 3 registers (6 byte) |
| Text length 16 characters (ex. M21.01.05) | 8 registers (16 byte) |
| Text length 20 characters (ex. M01.9) | 10 registers (20 byte) |
| Numeric value < 32768 (ex M01.07) | 1 registers (2 byte) |
| Numeric value > 32768 (ex M02.07) | 2 registers (4 byte) |
| IP address (ex. M16.0x.06 M16.0x.07) | 2 registers (4 byte) |

❶ It's possible to read the menu, submenus, and parameter stored at the addresses **5000H**, **5001H** and **5002H** by using the **function 4**

See the following example

ESEMPIO / EXAMPLE

Impostare a 200 il valore del parametro P02.01 (Primario TA)
Set to 200 the value of parameter P02.01 (CT primary)

Passo 1 :Selezione menu 02.

Step 1 :Select menu 02.

MASTER Funzione / Function = 6
Indirizzo / Address = 5000H (5000H - 0001H =4FFFH)
Valore / Value = 2 (02H)

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 01 | 06 | 4F | FF | 00 | 02 | 2E | EF |
|----|----|----|----|----|----|----|----|

DCRG Funzione / Function = 6
Indirizzo / Address = 5000H (5000H - 0001H =4FFFH)
Valore / Value = 4 (04H)

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 01 | 06 | 4F | FF | 00 | 02 | 2E | EF |
|----|----|----|----|----|----|----|----|

(Nota: in questo caso di esempio non è necessaria impostazione del sotto-menu all'indirizzo 5001)
(Note: In this example it is not necessary to set the sub-menu number at address 5001)

Passo 2 :Impostazione parametro 01.

Step 2 :Set parameter 01.

MASTER Funzione / Function = 6
Indirizzo / Address = 5002H (5002H - 0001H =5001H)
Valore / Value = 1 (01H)

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 01 | 06 | 50 | 01 | 00 | 01 | 08 | CA |
|----|----|----|----|----|----|----|----|

DCRG Funzione / Function = 6
Indirizzo / Address = 5002H (5002H - 0001H =5001H)
Valore / Value = 2 (02H)

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 01 | 06 | 50 | 01 | 00 | 01 | 08 | CA |
|----|----|----|----|----|----|----|----|

Passo 3 :Impostazione valore 200.

Step 3 :Set value 200.

MASTER Funzione / Function = 16 (10H)
Indirizzo / Address = 5004H (5004H - 0001H =5003H)
Nr. registri / Nr. register = 1 (01H)
Nr. byte / Nr. bytes = 2 (02H)
Valore / Value = 200 (000000C8H)

| | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|
| 01 | 10 | 50 | 03 | 00 | 01 | 02 | 00 | C8 | F7 | F0 |
|----|----|----|----|----|----|----|----|----|----|----|

DCRG Funzione / Function = 16 (10H)
Indirizzo / Address = 5004H (5004H - 0001H =5003H)
Valore / Value = 2 (02H)

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 01 | 10 | 50 | 03 | 00 | 01 | E0 | C9 |
|----|----|----|----|----|----|----|----|

Passo 4 :Salvataggio eeprom.

Step 4 :Save eeprom.

MASTER Funzione / Function = 6 (06H)
Indirizzo / Address = 2F03H (2F03H - 0001H =2F02H)
Valore / Value = 1 (01H)

| | | | | | | | |
|----|---|----|----|----|----|----|----|
| 01 | 6 | 2F | 02 | 00 | 01 | E1 | 1E |
|----|---|----|----|----|----|----|----|

DCRG Nessuna risposta/No answer

Passo 5 :Riavvio.

Step 5 :Reboot.

MASTER Funzione / Function = 6 (06H)
Indirizzo / Address = 2F03H (2F03H - 0001H =2F02H)
Valore / Value = 4 (04H)

| | | | | | | | |
|----|---|----|----|----|----|----|----|
| 01 | 6 | 2F | 02 | 00 | 04 | 21 | 1D |
|----|---|----|----|----|----|----|----|

DCRG Nessuna risposta/No answer