Via L. Da Vinci, 100, 50028, Tavarnelle V.P. – Loc. Sambuca (FI) – ITA, P.IVA 0438 634 0485
Ph +39 0558071267 Fax.+ 39 0558071338 telegroup@telegroup.it www.telegroup.it

(I)

REGOLATORE AUTOMATICO
DEL FATTORE DI POTENZA

PCRL3-PCRL5-PCRL8

PROTOCOLLO DI COMUNICAZIONE
MODBUS®

ADDENDUM

(GB)

DIGITAL POWER FACTOR CONTROLLER

PCRL3-PCRL5-PCRL8

MODBUS COMUNICATION PROTOCOL®

ADDENDUM

CE

PROTOCOLLO MODBUS®

I regolatori di rifasamento automatici della serie PCRL supportano i protocolli di comunicazione Modbus RTU®, Modbus ASCII® e Modbus TCP® (solo PCRL8) sull'interfaccia ottica e sui moduli di espansione:

- EXP 10 20 USB
- EXP 10 30 RS 232
- EXP 10 40 RS485
- EXP 10 50 Ethernet (solo PCRL8)

Grazie a questa funzione è possibile leggere lo stato degli apparecchi e controllare gli stessi tramite 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 ADV: è possibile configurare una sola porta di comunicazione.

MENU AVANZATO

Comunicazione seriale

PAR	Funzione	Default	Range
P.49	Indirizzo nodo	01	01-255
P.50	Velocità seriale	9600	1200 2400 4800 9600 19200 38400
P.51	Formato dati	8 bit – n	8 bit, no parità 8 bit, dispari 8 bit, pari 7 bit, dispari 7 bit, pari
P.52	Bit di stop	1	1-2
P.53	Protocollo	Modbus RTU	Modbus RTU Modbus ASCII *Modbus TCP* (Only PCRL8)

Per il modulo di espansione EXP 1050 (Ethernet solo PCRL8) esistono altri parametri.
Accedere al SETUP MENU e selezionare il menu FUN.

PAR	Funzione	Default	Range
F.01	Indirizzo IP	IP1 192 IP2 168 IP3 001 IP4 001	000.000.000.000 – 255.255.255.255
F.02	Subnet mask	SUB1 0 SUB2 0 SUB3 0 SUB4 0	000.000.000.000 – 255.255.255.255
F.03	Porta IP	1001	0-32000
F.04	Client / server	Server	Client Server
F.05	Indirizzo IP remoto	RIP1 0 RIP2 0 RIP3 0 RIP4 0	000.000.000.000 – 255.255.255.255
F.06	Porta IP remota	1001	0-32000
F.07	Indirizzo gateway IP	GW1 0 GW2 0 GW3 0 GW4 0	000.000.000.000 – 255.255.255.255

MODBUS® PROTOCOL

The PCRL series of power factor controller support the communication protocols Modbus RTU® , Modbus ASCII®, ModbusTCP® (Only PCRL8) on optical interface and the expansion modules:

- EXP 10 20 USB
- EXP 10 30 RS 232
- EXP 10 40 RS485
- EXP 10 50 Ethernet (solo PCRL8)

Using this function it is possible to read the device status and to control the units through 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 ADV menu: it is possible to configure only one communication ports.

ADVANCE MENU

Serial comunication

PAR	Function	Default	Range
P.49	Node address	01	01-255
P.50	Serial port speed	9600	1200 2400 4800 9600 19200 38400
P.51	Data format	8 bit – n	8 bit –no par. 8 bit, odd 8 bit, even 7 bit, odd 7 bit, even
P.52	Stop bits	1	1-2
P.53	Protocol	Modbus RTU	Modbus RTU Modbus ASCII *Modbus TCP* (Only PCRL8)

For expansion module EXP 10 50 (Ethernet only PCRL8), there are other parameters.
Enter on the SETUP MENU and select the menu FUN.

PAR	Function	Default	Range
F.01	IP address	IP1 192 IP2 168 IP3 001 IP4 001	000.000.000.000 – 255.255.255.255
F.02	Subnet mask	SUB1 0 SUB2 0 SUB3 0 SUB4 0	000.000.000.000 – 255.255.255.255
F.03	IP port	1001	0-32000
F.04	Client / server	Server	Client Server
F.05	Remote IP address	RIP1 0 RIP2 0 RIP3 0 RIP4 0	000.000.000.000 – 255.255.255.255
F.06	Remote IP port	1001	0-32000
F.07	IP gateway address	GW1 0 GW2 0 GW3 0 GW4 0	000.000.000.000 – 255.255.255.255

TELEGROUP S.r.L
Via L. Da Vinci, 100
50028, Tavarnelle Val di Pesa, Loc. Sambuca (FI) – Italy
P. IVA 0438 634 0485

Tel. +39 055 80 71 267
Fax + 39 055 80 71 338
telegroup@telegroup.it
www.telegroup.it

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

T1 T2 T3	Indirizzo (8 bit)	Funzione (8 bit)	Dati (N x 8 bit)	CRC (16 bit)	T1 T2 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 PCRL la lunghezza massima consentita per il campo dati e' 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 sia 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 dispositivo 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 sul dispositivo
04 = Read input register	Consente la lettura delle misure disponibili sul dispositivo
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 dalla PCRL con indirizzo 01 il valore della temperatura interna quadro che si trova alla locazione 14 (0E Hex), il messaggio da spedire è il seguente:

01	04	00	2D	00	02	E0	08

Dove:
01= indirizzo slave
04 = funzione di lettura locazione
002D = indirizzo della locazione <u>diminuito di un'unità</u>, contenete il valore della temperatura interna quadro
0002 = numero di registri da leggere a partire dall'indirizzo 2D
E0 08 = checksum CRC

La risposta del dispositivo è la seguente:

01	04	04	00	00	00	1C	FA	4D

Dove:
01= indirizzo del dispositivo (Slave 01)
04 = funzione richiesta dal Master
04 = numero di byte inviati dal dispositivo
00 00 00 1C = valore esadecimale della temperatura = 28 = 28 °C
FA 4D = 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 è indicato nella Tabella 2. 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 dispositivo ritorna un messaggio di errore (vedi tabella errori).

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

T1 T2 T3	Address (8 bit)	Function (8 bit)	Data (N x 8 bit)	CRC (16 bit)	T1 T2 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 PCRL 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 device 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 device measures.
04 = Read input register	Allows to read the device 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 the internal panel temperature, which resides at location 14 (0E Hex), from the PCRL with serial address 01, the message to send is the following:

01	04	00	2D	00	02	E0	08

Where:
01= slave address
04 = Modbus® function 'Read input register'
002D = Address of the required register (temperature) <u>decreased by one</u>
00 02 = Number of registers to be read beginning from address 2D
E0 08 = CRC Checksum

The device answer is the following:

01	04	04	00	00	00	1C	FA	4D

Where:
01 = device address (Slave 01)
04 = Function requested by the master
04 = Number of bytes sent by the DEVICE
00 00 00 1C = Hex value of the temperature =28 = 28 °C
FA 4D = 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 table 2. 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 device will return an error code (see error table).

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.

FUNZIONE 06: PRESET SINGLE REGISTER

Questa funzione permette di scrivere nei registri. Essa può essere utilizzata solo con i registri d'indirizzo superiore a 1000 Hex. E' possibile ad esempio impostare i parametri del set-up. Qualora il valore impostato non rientri nel valore minimo e massimo della tabella il dispositivo risponderà con un messaggio di errore. Se viene richiesto un parametro ad un indirizzo inesistente verrà risposto con un messaggio di errore.

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 regolatore di rifasamento.

Richiesta Master:

Indirizzo slave	01h
Funzione	07h
LSB CRC	41h
MSB CRC	E2h

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

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.

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 device will answer with an error message. In the same way, if the parameter address is not recognised, the DEVICE will send an error response.

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 power factor controller.

Master query:

Slave address	01h
Function	07h
LSB CRC	41h
MSB CRC	E2h

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

FUNZIONE 17: REPORT SLAVE ID
Questa funzione permette di identificare il tipo di dispositivo.

Richiesta Master.

Indirizzo slave	01h
Funzione	11h
LSB CRC	C0h
MSB CRC	2Ch

Risposta Slave:

Indirizzo slave	01h
Funzione	11h
Contatore bytes	08h
Dato 01 (Tipo) ❶	49h
Dato 02 (Revisione software)	00h
Dato 03 (Revisione hardware)	00h
Dato 04 (Revisione parametri)	00h
Dato 05 (tipoolgia di prodotto) ❷	01h
Dato 06 (riservato)	00h
Dato 07 (riservato)	08h
LSB CRC	00h
MSB CRC	F7h

❶
67 - 43h = PCRL3
69 - 45h = PCRL5
73 - 49h = PCRL8 ❷
1 – 01h= Serie PCRL

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 17: REPORT SLAVE ID
This function allows to identify the device type.

Master query.

Slave address	01h
Function	11h
LSB CRC	C0h
MSB CRC	2Ch

Slave response:

Slave address	01h
Function	11h
Contatore bytes	08h
Data 01 (Type)❶	49h
Data 02 (Sw revision)	00h
Data 03 (Hardware revision)	00h
Data 04 (Parameter revision)	00h
Data 05 (type of device) ❷	01h
Data 06 (reserved)	00h
Data 07 (reserved)	00h
LSB CRC	00h
MSB CRC	F7h

❶
67 - 43h = PCRL3
69 - 45h = PCRL5
73 - 49h = PCRL8 ❷
1 – 01h= Serie PCRL

ERRORS
In case the slave receives an incorrect message, it answers with a massage 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

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 P.53 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 dispositivo con indirizzo 8 il valore della tensione equivalente che si trova alla locazione 4 (04 Hex), il messaggio da spedire è il seguente:

:	08	04	00	03	00	02	56	CRLF

Dove:
: = ASCII 3Ah = Delimitatore inizio messaggio
08 = indirizzo slave.
04 = funzione di lettura locazione.
00 03 = indirizzo della locazione <u>diminuito di un'unità</u>, contenente il valore della corrente di fase L3
00 02 = numero di registri da leggere a partire dall'indirizzo 04.
56 = checksum LRC.
CRLF = ASCII 0Dh 0Ah = delimitatore fine messaggio

La risposta del dispositivo è la seguente:

:	08	04	04	00	00	01	3D	9B	CRLF

Dove:
: = ASCII 3Ah = Delimitatore inizio messaggio
08 = indirizzo del dispositivo (Slave 08).
04 = funzione richiesta dal Master.
04 = numero di byte inviati dallo slave.
00 00 01 3D = valore esadecimale della tensione misurata (=317 V)
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 P.53 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 current phase L3, which resides at location 12 (0C Hex) from the slave with serial address 08, the message to send is the following:

:	08	04	00	03	00	02	56	CRLF

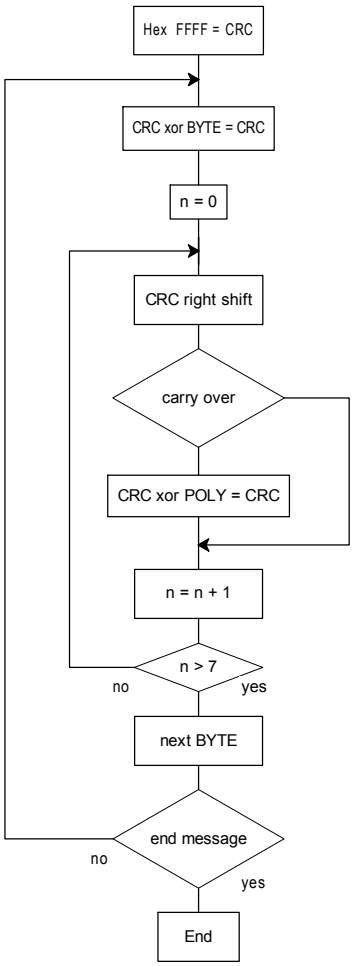
Whereas:
: = ASCII 3Ah message start delimiter
08 = slave address.
04 = Modbus® function 'Read input register'
00 03 = Address of the required register (L3 current phase) <u>decreased by one</u>

00 02 = Number of registers to be read beginning from address 04.
56 = LRC Checksum.
CRLF = ASCII 0Dh 0Ah = Message end delimiter

The answer of the device is the following:

:	08	04	04	00	00	01	3D	9B	CRLF

Whereas:
: = ASCII 3Ah message start delimiter
08 = PCRL address (Slave 08)
04 = Function requested by the master
04 = Number of bytes sent by the multimeter
00 00 01 3D = Hex value of the measured voltage (=317 V)
9B = LRC checksum
CRLF = ASCII 0Dh 0Ah = Message end delimiter



CALCOLO DEL CRC (CHECKSUM per RTU)
Esempio di calcolo:
Frame = 0207h

Inizializzazione CRC	1111	1111	1111	1111	
Carica primo byte			0000	0010	
Esegue xor con il primo Byte della frame	1111	1111	1111	1101	
Esegue primo shift a 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**
Risultato CRC	**0001**	**0010**			
0100	**0001**				
	12h	**41h**			

Nota: Il byte 41h viene spedito per primo (anche se è 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

CRC initialization	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**
CRC Result	**0001**	**0010**		**0100**	
	0001				
	12h	**41h**			

Note: The byte 41h is sent first(even if it is the LSB), then12h 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 COMUNICAZIONE
(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
02h	2	Cosfi attuale ❶	Actual cos-phi ❶		Unsigned long
04h	2	Tensione	Voltage	V	Unsigned long
06h	2	Corrente	Current	A / 1000	Unsigned long
08h	2	Delta-kvar ❷	Delta-kvar ❷	kvar	Signed long
0Ah	2	Fattore di potenza medio settimanale	Weekly average power factor	/100	Unsigned long
0Ch	1	Corrente sovraccarico condensatori %	Capacitor current overload %	%	Unsigned integer
0Eh	2	Temperatura quadro ❷	Panel temperature ❷	°C	Signed long
10h	2	Stato uscite	Output status		Unsigned long
12h	2	Sfasamento	Phase angle offset	° / 4	Unsigned long
14h	2	Fattore di potenza attuale	Actual power factor		Signed long
16h	2	Stato bit di errore ❸	Error bits status ❸		Unsigned long
18h	2	MAX tensione	MAX voltage	V	Unsigned long
1Ah	2	MAX corrente	MAX current	A / 100	Unsigned long
1Ch	2	MAX sovraccarico condensatori	MAX capacitor overload	%	Unsigned long
1Eh	2	MAX temperatura ❷	MAX temperature ❷	°C	Unsigned long
200h	2	Tempo di inserzione gradino 1	Time insertion step 1	min	Unsigned long
---					
21Ah	2	Tempo di inserzione gradino 14	Time insertion step 14	min	Unsigned long
2020h	1	Numero inserzioni gradino 1	Number of insertion step 1		Unsigned integer
----	---	---	---	---	---
202Dh	1	Numero inserzioni gradino 14	Number of insertion step 14		Unsigned integer
2040h	1	Potenza residua step 1	Step 1 residual power	KVAr	Unsigned integer
---	---	---	---	---	---
204Dh	1	Potenza residua step 1	Step 1 residual power	KVAr	Unsigned integer
2008h	1	THDV	THDV	% /10	
2080h	1	2° Armonica di tensione	2nd voltage harmonics	% /10	Unsigned integer
---	---	---	---	---	---
208Dh	1	15° Armonica di tensione	15th voltage harmonics	% /10	Unsigned integer
2009h	1	THDI	THDI	%	Unsigned integer
20A0h	1	2° Armonica di corrente	2nd current harmonic	% /10	Unsigned integer
---	---	---	---	---	---
20ADh	1	15° Armonica di corrente	15th current harmonics	% /10	Unsigned integer
2007h	1	Scadenza manutenzione	Hours to maintenance	h	Signed integer

❶ Il bit 31 indica il segno (0=pos, 1=neg), mentre il bit 30 indica induttivo/capacitivo(0=ind, 1=cap).
Bit 31 indicates the sign (0=pos., 1=neg.), while bit 30 indicates inductive/capacitive load (0=ind, 1=cap).
❷ Il bit 31 indica il segno (0=pos, 1=neg)
Bit 31 indicates the sign (0=pos., 1=neg.)
❸Leggendo le word all'indirizzo 16h vengono restituiti 32 bit con significato come da tabella:
Reading the words starting at address 16h will return 32 bits with the following meaning:

TABELLA 3: **TABLE 3:**

Bit	*Codice*	Allarme	*Code*	Alarm
0	**A01**	Sottocompensazione	**A01**	*Under compensation*
1	**A02**	Sovracompensazione	**A02**	*Over compensation*
2	**A03**	Corrente troppo bassa	**A03**	*Low current*
3	**A04**	Corrente troppo alta	**A04**	*High current*
4	**A05**	Tensione troppo bassa	**A05**	*Low voltage*
5	**A06**	Tensione troppo alta	**A06**	*High voltage*
6	**A07**	Sovraccarico condensatori	**A07**	*Capacitor overload*
7	**A08**	Temperatura troppo alta	**A08**	*Overtemperature*
8	**A09**	Microinterruzione	**A09**	*No-voltage release*
9	**A10**	THD tensione troppo alto	**A10**	*Voltage THD too high*
10	**A11**	THD corrente impianto troppo alto	**A11**	*Current THD too high*
11	**A12**	Richiesta manutenzione	**A12**	*Maintenance requested*
12	**A13**	Step difettoso	**A13**	*Step failure*
13..31	**-**	Bit liberi	**-**	Free bits

TABELLA 4: COMANDI
(Utilizzabili con funzione 06)

TABLE 4: COMMANDS
(To be used with function 06)

INDIRIZZO ADDRESS	WORDS	FUNZIONE	FUNCTION	FORMATO FORMAT
2FF0h	1	Esecuzione comando menu comandi ❶	Commands list ❶	Unsigned integer
3000h	1	Cambio modalità operativa❷	Operative mode change❷	Unsigned integer
3001h	1	Reset apparecchio (warm boot) ❸	Device reset (warm boot) ❸	Unsigned integer
3005h	1	Attivazione di uno step ❹	Step activation ❹	Unsigned integer
3006h	1	Disattivazione step ❹	Step de-activation ❹	Unsigned integer
3007h	1	Blocco tastiera ON/OFF❺	Keyboard lock ON/OFF❺	Unsigned integer

❶ La tabella indica le funzioni associate al valore da scrivere all'indirizzo 2FF0h. E possibile eseguire più di una funzione contemporaneamente.
The following table shows functions generated by the single bits of the value written to address 2FF0h. It is possible to execute several function at the same time.

TABELLA 5: **TABLE 5:**

Valore Value	Funzione	*Function*
00h	C01 Azzera intervallo di manutenzione	*C01 Reset maintenance service interval.*
01h	C02 Azzera i contatore di manovre step.	*C02 Reset step operation counters.*
02h	C03 Ripristina le potenze originali nell'aggiustamento step.	*C03 Reload originally programmed power into step trimming.*
03h	C04 Azzera i conta ore di funzionamento step.	*C04 Reset step operation hour meters.*
04h	C05 Azzera i picchi massimi registrati delle misure.	*C05 Reset maximum peak values.*
05h	C06 Azzera memoria TPF settimanale.	*C06 Resets weekly total power factor history.*
06h	C07 Ripristina i parametri al default di fabbrica.	*C07 Resets setup programming to factory default.*
07h	C08 Salva una copia di backup delle impostazioni di setup dell' utente .	*C08 Makes a backup copy of user setup parameters settings.*
08h	C09 Ripristina i parametri al valore della copia utente.	*C09 Reloads setup parameters with the backup of user settings.*

❷ La seguente tabella indica i valori da scrivere all'indirizzo 3000h per ottenere le corrispondenti funzioni.
The following table shows the values to be written to address 3000h to achieve the correspondent functions.

Valore Value	Funzione	*Function*
0	Passaggio da MAN ad AUT e viceversa	*Switch from AUT to MAN and vice versa*
1	Passaggio a modalita' MAN	*Switch to MAN mode*
2	Passaggio a modalita' AUT	*Switch to AUT mode*

❸ Scrivendo il valore 01 all'indirizzo indicato viene eseguita la corrispondente funzione.
Writing value 01 to the indicated address, the correspondent function will be executed.

❹ Scrivere nel registro indicato il numero dello step da attivare/disattivare. Se si tenta di attivare uno step per il quale è in corso il tempo di riconnessione, il comando non verrà eseguito.
Write in the correspondent register the number of the step to be activated/ deactivated. Trying to activate a step while the correspondent reconnection time is running, the command will be ignored.

❺ Scrivendo il valore 01 all'indirizzo indicato viene bloccata la tastiera scrivendo 00 viene sbloccata.
Writing value 01 to the indicated address the keyboard is locked writing 00 it's unlocked.

IMPOSTAZIONE PARAMETRI

Tramite il protocollo Modbus® è possibile accedere ai parametri dei menu. Per interpretare correttamente la corrispondenza fra valore numerico e funzione selezionata e/o unità di misura, fare riferimento al manuale operativo del dispositivo.

PROCEDURA PER LA LETTURA DEI PARAMETRI

1. Scrivere il valore del parametro che si vuole leggere tramite la ***funzione 6*** all'indirizzo ***5002h*** **❶.**
2. Eseguire la ***funzione 4*** all'indirizzo ***5003h***, di un numero di registri appropriato alla lunghezza del parametro (vedi tabella).
3. Se si vuole leggere il parametro successivo ripetere il passo 2, altrimenti eseguire il passo 1.

PROCEDURA PER LA SCRITTURA DEI PARAMETRI

1. Scrivere il valore parametro che si vuole modificare tramite la ***funzione 6*** all'indirizzo ***5002h***❶
2. Eseguire la ***funzione 16*** all'indirizzo ***5003h***, di un numero di registri appropriato alla lunghezza del parametro.
3. Se si vuole scrivere il parametro successivo ripetere il passo 2, altrimenti eseguire il passo 1, se non bisogna scrivere ulteriori parametri eseguire il passo 4.
4. Per rendere effettivo un cambiamento nel menu di set-up è necessario memorizzare i valori in EEPROM utilizzando l'apposito comando. Scrivere il valore 5 con la ***funzione 6*** all'indirizzo ***2F03h***.

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 device operating manual.

PROCEDURE FOR THE READING OF PARAMETERS

1. Write the value of the parameter that you want to read by using the ***function 6*** at address ***5002h***❶.
2. Perform the ***function 4*** at the address ***5003h***, with a number of registers appropriate to the length of the parameter (see table).
3. If you want to read the next parameter repeat step 2, otherwise perform step 1.

PROCEDURE FOR THE WRITING OF PARAMETERS

1. Write the value of the parameter that you want to change by using the ***function 6*** at address ***5002h***❶
2. Perform the ***function 16*** at address ***5003h***, with a number of registers appropriate to the length of the parameter
3. If you want to write the next parameter repeat step 2, otherwise perform step 1, if you do not have to write additional parameters go to step 4.
4. To make effective the changes made to setup parameters it is necessary to store the values in EEPROM using the dedicated command. Write value 5 by using ***function 6*** at address ***2F03h***.

TELEGROUP S.r.l.
Via L. Da Vinci, 100
50028, Tavarnelle Val di Pesa, Loc. Sambuca (FI) – Italy
P. IVA 0438 634 0485

Tel. +39 055 80 71 267
Fax + 39 055 80 71 338
telegroup@telegroup.it
www.telegroup.it

TIPO DI PARAMETRO	NUMERO REGISTRI
Valore numerico <= 65535 (es P.30)	1 registri (2 byte)
Valore numerico > 65535 (es P.6)	2 registri (4 byte)

❶ E possibile leggere il valore del parametro memorizzati all'indirizzo ***5002h*** utilizzando la ***funzione 4***

Vedere esempio

ESEMPIO
Impostare a 250 il valore del parametro P.01

Passo 1: Impostazione parametro P.01

MASTER Funzione = 6
Indirizzo = 5002h (5002h – 0001h =5001h)
Valore = 1 (01h)

01	06	50	01	00	01	08	CA

PCRL Funzione = 6
Indirizzo = 5002h (5002h – 0001h =5001h)
Valore = 1 (01h)

01	06	50	01	00	01	08	CA

Passo 2: Impostazione valore 250.

MASTER Funzione = 16 (10h)
Indirizzo = 5003h (5003h – 0001h =5002h)
Nr. registri = 2 (02h)
Nr. byte = 4 (04h)
Valore = 250 (000000FAh)

01	10	50	02	00	02	04	00	00	00	FA	0E	36

PCRL Funzione = 16 (10h)
Indirizzo = 5003H (5004h – 0001h =5003h)
Valore = 250 (000000FAh)

01	10	50	02	00	02	F1	08

Passo 3: Salvataggio e riavvio.

MASTER Funzione = 6 (06h)
Indirizzo = 2F03h (2F03h – 0001h =2F02h)
Valore = 5 (05h)

01	6	2F	02	00	04	21	1D

PCRL Nessuna risposta

TYPE OF PARAMETER	NUMBER OF REGISTER
Numeric value <= 65535 (es P.30)	1 registers (2 byte)
Numeric value > 65535 (es P.6)	2 registers (4 byte)

❶It's possible to read the parameter stored at the addresse ***5002h*** by using the ***function 4***

See the example:

EXAMPLE
Set to 250 the value of parameter P.01

Step 1: Set parameter P.01

MASTER Function = 6
Address = 5002h (5002h – 0001h =5001h)
Value = 1 (01h)

01	06	50	01	00	01	08	CA

PCRL Function = 6
Address = 5002h (5002h – 0001h =5001h)
Value = 1 (01h)

01	06	50	01	00	01	08	CA

Step 2: Set value 250.

MASTER Function = 16 (10h)
Address = 5003h (5003h – 0001h =5002h)
Nr. register = 1 (02h)
Nr. bytes = 2 (04h)
Value = 250 (000000FAh)

01	10	50	02	00	02	04	00	00	00	FA	0E	36

PCRL Function = 16 (10h)
Address = 5003H (5004h – 0001h =5003h)
Value = 250 (000000FAh)

01	10	50	02	00	02	F1	08

Step 3: Save and reboot.

MASTER Funzione = 6 (06h)
Indirizzo = 2F03h (2F03h – 0001h =2F02h)
Valore = 5 (04h)

01	6	2F	02	00	04	21	1D

PCRL Nessuna risposta/No answer