



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
 E-mail info@LovatoElectric.com

**FFL700EP****FFL800EP**

UNITA' DI CONTROLLO ELETTROPOMPA
 ANTINCENDIO (EN12845)

PROTOCOLLO DI COMUNICAZIONE MODBUS®**PROTOCOLLO MODBUS®**

Le unità di controllo di elettropompe antincendio FFL...EP supportano i protocolli di comunicazione Modbus RTU®, Modbus ASCII® e Modbus TCP® sull'interfaccia ottica, sulla porta di comunicazione RS485 integrata e sui moduli di espansione per FFL800EP:

- EXP1012T RS485
- EXP1013T Ethernet

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 M11. È possibile configurare 3 porte di comunicazione (n=1...3). La porta di comunicazione 1 è riservata alla comunicazione RS485 integrata.

MENU M11 – COMUNICAZIONE**Comunicazione seriale RS485**

PAR	Funzione	Default	Range
P11.n.01	Indirizzo nodo	01	01-255
P11.n.02	Velocità seriale	9600	1200 2400 4800 9600 19200 38400 57600 115200
P11.n.03	Formato dati	8 bit – n	8 bit, no parità 8 bit, dispari 8 bit, pari 7 bit, dispari 7 bit, pari
P11.n.04	Bit di stop	1	1-2
P11.n.05	Protocollo	Modbus RTU	Modbus RTU Modbus ASCII Modbus TCP

Per il modulo di espansione EXP1013T (Ethernet) esistono altri parametri.

PAR	Funzione	Default	Range
P11.n.06	Indirizzo IP	192.168.1.1	000.000.000.000 – 255.255.255.255
P11.n.07	Subnet mask	0.0.0.0	000.000.000.000 – 255.255.255.255
P11.n.08	Porta IP	1001	0-32000
P11.n.09	Funzione canale	Slave	Slave Gateway
P11.n.10	Client / server	Server	Client Server
P11.n.11	Indirizzo IP remoto	0.0.0.0	000.000.000.000 – 255.255.255.255
P11.n.12	Porta IP remota	1001	0-32000
P11.n.13	Indirizzo gateway IP	0.0.0.0	000.000.000.000 – 255.255.255.255

**FFL700EP****FFL800EP**

ELECTRIC FIRE PUMP CONTROLLERS
 (EN12845)

COMMUNICATION PROTOCOL MODBUS®**MODBUS® PROTOCOL**

The electric fire pump controllers FFL...EP supports the communication protocols Modbus RTU®, Modbus ASCII®, Modbus TCP® on optical interface, on the built in RS485 communication port and on the expansion modules for FFL800EP:

- EXP1012T RS485
- EXP1013T Ethernet

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 M11 menu. It is possible to configure 3 different communication ports (n=1...3). The communication port 1 is reserved for the built in RS485 communication.

MENU M11 – COMMUNICATION**Serial communication RS485**

PAR	Function	Default	Range
P11.n.01	Node address	01	01-255
P11.n.02	Baudrate	9600	1200 2400 4800 9600 19200 38400 57600 115200
P11.n.03	Data format	8 bit – n	8 bit – no par. 8 bit, odd 8 bit, even 7 bit, odd 7 bit, even
P11.n.04	Stop bits	1	1-2
P11.n.05	Protocol	Modbus RTU	Modbus RTU Modbus ASCII Modbus TCP

For expansion module EXP1013T (Ethernet), there are other parameters.

PAR	Function	Default	Range
P11.n.06	IP address	192.168.1.1	000.000.000.000 – 255.255.255.255
P11.n.07	Subnet mask	0.0.0.0	000.000.000.000 – 255.255.255.255
P11.n.08	IP port	1001	0-32000
P11.n.09	Channel function	Slave	Slave Gateway
P11.n.10	Client / server	Server	Client Server
P11.n.11	Remote IP address	0.0.0.0	000.000.000.000 – 255.255.255.255
P11.n.12	Remote IP port	1001	0-32000
P11.n.13	IP gateway address	0.0.0.0	000.000.000.000 – 255.255.255.255

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.
- La lunghezza massima consentita per il campo dati è di 80 registri da 16 bit (160 bytes).
- Il campo CRC consente sia al master sia 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.

FFL 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.
04 = Read input register	Consente la lettura delle misure disponibili.
06 = Preset single register	Permette la scrittura dei parametri
07 = Read exception	Permette di leggere lo stato dell'apparecchio
16 = 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 da una FFL...EP con indirizzo 01 la temperatura dell'acqua, che si trova alla locazione 3850 (0F0A Hex), il messaggio da spedire è il seguente:

01	04	0F	09	00	02	A2	DD
----	----	----	----	----	----	----	----

Dove:

01= indirizzo slave

04 = funzione di lettura locazione

0F 09 = indirizzo della locazione diminuito di un'unità

00 02 = numero di registri da leggere a partire dall'indirizzo 0F0A

A2 DD = checksum CRC

La risposta dell'FFL...EP è la seguente:

01	04	04	00	00	09	EC	FC	59
----	----	----	----	----	----	----	----	----

Dove:

01= indirizzo della FFL...EP (Slave 01)

04 = funzione richiesta dal Master

04 = numero di byte inviati dalla FFL...EP

00 00 09 EC = il valore della temperatura acqua in esadecimale = 25.40 °C

FC 59 = 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.
- 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.

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

For instance, to read the water temperature, which resides at location 3850 (0F0A Hex), from the FFL...EP with serial address 01, the message to send is the following:

01	04	0F	09	00	02	A2	DD
----	----	----	----	----	----	----	----

Where:

01= slave address

04 = Modbus® function 'Read input register'

0F 09 = Address of the required register, decreased by one

00 02 = Number of registers to be read beginning from address 0F0A

A2 DD = CRC Checksum

The FFL...EP answer is the following:

01	04	04	00	00	09	EC	FC	59
----	----	----	----	----	----	----	----	----

Where:

01 = FFL...EP address (Slave 01)

04 = Function requested by the master

04 = Number of bytes sent by the FFL...EP

00 00 09 EC = Hex value of water temperature = 25.40 °C

FC 59 = 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 risponderà con un messaggio di errore (vedi tabella errori).

Richiesta Master:

Indirizzo slave	01h
Funzione	04h
MSB Indirizzo registro	00h
LSB Indirizzo registro	01h
MSB Numero registri	00h
LSB Numero registri	02h
LSB CRC	20h
MSB CRC	0Bh

Nell'esempio vengono richiesti, allo slave numero 1, 2 registri consecutivi a partire dall'indirizzo 01h. Quindi vengono letti i registri dall' 01h al 02h.

Il comando termina sempre con il valore di checksum CRC.

Risposta Slave:

Indirizzo slave	01h
Funzione	04h
Numero di byte	04h
MSB Dato 01h	00h
LSB Dato 01h	00h
MSB Dato 02h	00h
LSB Dato 02h	00h
LSB CRC	FBh
MSB CRC	84h

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. È possibile ad esempio impostare i parametri del setup. 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. L'indirizzo ed il range valido per i vari parametri può essere trovato nella Tabella 4.

Richiesta Master:

Indirizzo slave	01h
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 dispositivo.

Richiesta Master:

Indirizzo slave	01h
Funzione	07h
LSB CRC	47h
MSB CRC	B2h

FUNZIONE 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 answer with an error code (see error table).

Master query:

Slave address	01h
Function	04h
MSB address	00h
LSB address	01h
MSB register number	00h
LSB register number	02h
LSB CRC	20h
MSB CRC	0Bh

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	01h
Function	04h
Byte number	04h
MSB register 01h	00h
LSB register 01h	00h
MSB register 02h	00h
LSB register 02h	00h
LSB CRC	FBh
MSB CRC	84h

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

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. The address and the valid range for each parameter are indicated in Table 4.

Master message:

Indirizzo slave	01h
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.

FUNZIONE 07: READ EXCEPTION STATUS

This function allows to read the status of the device.

Master query:

Slave address	01h
Function	07h
LSB CRC	47h
MSB CRC	B2h

La tabella seguente riporta il significato del byte inviato dall'FFL...EP come risposta:

BIT	SIGNIFICATO
0	Non usato
1	Modo operativo MAN
2	Modo operativo AUT
3	Non usato
4	In errore
5	Richiesta avviamento motore
6	Non usato
7	Allarme globale attivato

FUNZIONE 16: PRESET MULTIPLE REGISTER

Questa funzione permette di modificare più parametri consecutivamente o parametri composti da più di 2 byte.

Richiesta Master:		Risposta Slave:	
Indirizzo slave	01h	Indirizzo slave	08h
Funzione	10h	Funzione	10h
MSB Indirizzo registro	20h	MSB Indirizzo registro	20h
LSB Indirizzo registro	01h	LSB Indirizzo registro	01h
MSB Numero registri	00h	MSB Numero byte	00h
LSB Numero registri	02h	LSB Numero byte	02h
Numero di byte (è il doppio di quelli sopra)	04h	LSB CRC	1Bh
MSB Dato	00h	MSB CRC	51h
LSB Dato	00h		
MSB Dato	00h		
LSB Dato	00h		
LSB CRC	85h		
MSB CRC	3Eh		

FUNZIONE 17: REPORT SLAVE ID

Questa funzione permette di identificare il tipo di dispositivo.

Richiesta Master:		Risposta Slave:	
Indirizzo slave	01h	Indirizzo slave	01h
Funzione	17h	Funzione	17h
LSB CRC	34h	Contatore bytes	08h
MSB CRC	44h	Dato 01 (Tipo) ①	91h
		Data 02 (Sw revision)	01h
		Data 03 (Hardware revision)	00h
		Data 04 (Parameter revision)	01h
		Data 05 (type of device) ②	07h
		Data 06 (reserved)	00h
		Data 07 (reserved)	00h
		Data 08 (reserved)	00h
		LSB CRC	80h
		MSB CRC	33h

- ① 91h = FFL700EP / 93h = FFL800EP
 ② 07h = Serie FFL

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

The following table gives the meaning of the status byte sent by the FFL...EP as answer:

BIT	MEANING
0	Not used
1	Operative mode MAN
2	Operative mode AUT
3	Not used
4	On error
5	Pump starting request
6	Not used
7	Global alarm on

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.

Master message:		Slave response:	
Slave address	01h	Slave address	08h
Function	10h	Function	10h
MSB register address	20h	MSB register address	20h
LSB register address	01h	LSB register address	01h
MSB register number	00h	MSB byte number	00h
LSB register number	02h	LSB byte number	02h
Number of byte (it is the double of the above)	04h	LSB CRC	1Bh
MSB data	00h	MSB CRC	51h
LSB data	00h		
MSB data	00h		
LSB data	00h		
LSB CRC	85h		
MSB CRC	3Eh		

FUNZIONE 17: REPORT SLAVE ID

This function allows to identify the device type.

Master query:		Slave response:	
Slave address	01h	Slave address	01h
Function	17h	Function	17h
LSB CRC	34h	Contatore bytes	08h
MSB CRC	44h	Data 01 (Type) ①	91h
		Data 02 (Revisione software)	01h
		Data 03 (Revisione hardware)	00h
		Data 04 (Revisione parametri)	01h
		Data 05 (tipologia di prodotto) ②	07h
		Data 06 (riservato)	00h
		Data 07 (riservato)	00h
		Data 08 (riservato)	00h
		LSB CRC	80h
		MSB CRC	33h

- ① 91h = FFL700EP / 93h = FFL800EP
 ② 07h = FFL series

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

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 P11.x.05 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).

MODBUS® ASCII PROTOCOL

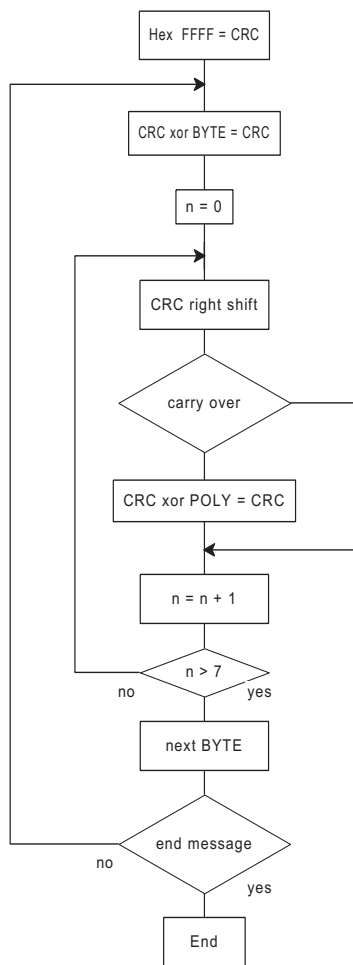
The Modbus® ASCII protocol is normally used in application that require to communicate through a couple of modems.

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 P11.x.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).



Algoritmo di calcolo del CRC
CRC calculation algorithm

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	1111	1111	1111	1101
Byte della frame				
Esegue primo shift dx	0111	1111	1111	1110 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il	1101	1111	1111	1111
polinomio				
Esegue secondo shift dx	0110	1111	1111	1111 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il	1100	1111	1111	1110
polinomio				
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	1001	0011	1111	1110
Polinomio				
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			0000	0111
della frame				
Esegue xor con il	1000	0001	0011	1001
Secondo byte della frame				
Esegue primo shift dx	0100	0000	1001	1100 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il	1110	0000	1001	1101
polinomio				
Esegue secondo shift dx	0111	0000	0100	1110 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il	1101	0000	0100	1111
polinomio				
Esegue terzo shift dx	0110	1000	0010	0111 1
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il	1100	1000	0010	0110
polinomio				
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	1001	0010	0000	1000
polinomio				
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 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

CRC initialization	1111	1111	1111	1111
Load the first byte			0000	0010
Execute xor with the first	1111	1111	1111	1101
Byte of the frame				
Execute 1st right shift	0111	1111	1111	1110 1
Carry=1,load polynomial	1010	0000	0000	0001
Execute xor with the	1101	1111	1111	1111
polynomial				
Execute 2nd right shift	0110	1111	1111	1111 1
Carry=1,load polynomial	1010	0000	0000	0001
Execute xor with the	1100	1111	1111	1110
polynomial				
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	1001	0011	1111	1110
polynomial				
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	1000	0100	1111	1110
polynomial				
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			0000	0111
of the frame				
Execute xor with the	1000	0001	0011	1001
Second byte of the frame				
Execute 1st right shift	0100	0000	1001	1100 1
Carry=1,load polynomial	1010	0000	0000	0001
Execute xor with the	1110	0000	1001	1101
polynomial				
Execute 2nd right shift	0111	0000	0100	1110 1
Carry=1,load polynomial	1010	0000	0000	0001
Execute xor with the	1101	0000	0100	1111
polynomial				
Execute 3rd right shift	0110	1000	0010	0111 1
Carry=1,load polynomial	1010	0000	0000	0001
Execute xor with the	1100	1000	0010	0110
polynomial				
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	1001	0010	0000	1000
polynomial				
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), 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
0002h	2	Tensione di fase L1	L1 phase voltage	V/100	Unsigned long
0004h	2	Tensione di fase L2	L2 phase voltage	V/100	Unsigned long
0006h	2	Tensione di fase L3	L3 phase voltage	V/100	Unsigned long
0008h	2	Corrente di fase L1	L1 phase current	A/10000	Unsigned long
000Ah	2	Corrente di fase L2	L2 phase current	A/10000	Unsigned long
000Ch	2	Corrente di fase L3	L3 phase current	A/10000	Unsigned long
000Eh	2	Tensione L1-L2	L1-L2 voltage	V/100	Unsigned long
0010h	2	Tensione L2-L3	L2-L3 voltage	V/100	Unsigned long
0012h	2	Tensione L3-L1	L3-L1 voltage	V/100	Unsigned long
0014h	2	Potenza attiva L1	L1 active power	W/10	Signed long
0016h	2	Potenza attiva L2	L2 active power	W/10	Signed long
0018h	2	Potenza attiva L3	L3 active power	W/10	Signed long
001Ah	2	Potenza reattiva L1	L1 reactive power	Var/10	Signed long
001Ch	2	Potenza reattiva L2	L2 reactive power	Var/10	Signed long
001Eh	2	Potenza reattiva L3	L3 reactive power	Var/10	Signed long
0020h	2	Potenza apparente L1	L1 apparent power	VA/10	Unsigned long
0022h	2	Potenza apparente L2	L2 apparent power	VA/10	Unsigned long
0024h	2	Potenza apparente L3	L3 apparent power	VA/10	Unsigned long
0026h	2	Fattore Di Potenza L1	L1 power factor	/10000	Signed long
0028h	2	Fattore Di Potenza L2	L2 power factor	/10000	Signed long
002Ah	2	Fattore Di Potenza L3	L3 power factor	/10000	Signed long
002Ch	2	Tensione di fase equivalente	Equivalent phase voltage	V/100	Unsigned long
002Eh	2	Corrente equivalente	Equivalent current	A/10000	Unsigned long
0030h	2	Tensione concatenata equivalente	Equivalent phase-to-phase voltage	V/100	Unsigned long
0032h	2	Potenza attiva equivalente	Equivalent active power	W/10	Signed long
0034h	2	Potenza reattiva equivalente	Equivalent reactive power	Var/10	Signed long
0036h	2	Potenza apparente equivalente	Equivalent apparent power	VA/10	Unsigned long
0038h	2	Fattore di potenza equivalente	Equivalent power factor	/10000	Signed long
003Ah	2	Frequenza	Frequency	Hz/1000	Unsigned long
003Ch	2	Asimmetria tensione fase-fase	Phase-to-phase voltage asymmetry	%/10	Unsigned long
003Eh	2	Asimmetria tensione fase-neutro	Phase-to-neutral voltage asymmetry	%/100	Unsigned long
0040h	2	Asimmetria corrente	Current asymmetry	%/100	Unsigned long
0042h	2	Corrente equivalente massima	Max equivalent current	A/10000	Unsigned long
0044h	2	Tensione fase-fase equivalente massima	Max equivalent phase-to-phase voltage	V/100	Unsigned long
0046h	2	Tensione fase-fase equivalente minima	Min equivalent phase-to-phase voltage	V/100	Unsigned long
0F0Eh	2	Temperatura acqua	Water temperature	*/100	Signed long
0F80h	2	Ore di lavoro totali	Total working hours	hours	Unsigned long
0F84h	2	Ore di lavoro parziali	Partial working hours	hours	Unsigned long
2200h	1	Allarmi da A01 a A16	Alarms from A01 to A16		Unsigned int
2201h	1	Allarmi da A17 a A32	Alarms from A17 to A32		Unsigned int
2202h	1	Allarmi da A33 a UA6	Alarms from A33 to UA6		Unsigned int
2203h	1	Allarmi da UA7 a UA12	Alarms from UA7 to UA12		Unsigned int
21C0h	1	OR di tutti i limiti	OR of all limits	bits	Unsigned int
21C1h	1	LIM 1	LIM 1	bits	Unsigned int
21C2h	1	LIM 2	LIM 2	bits	Unsigned int
21C3h	1	LIM 3	LIM 3	bits	Unsigned int
21C4h	1	LIM 4	LIM 4	bits	Unsigned int
21C5h	1	LIM 5	LIM 5	bits	Unsigned int
21C6h	1	LIM 6	LIM 6	bits	Unsigned int
21C7h	1	LIM 7	LIM 7	bits	Unsigned int
21C8h	1	LIM 8	LIM 8	bits	Unsigned int
1D00h	2	Contatore CNT 1	Counter CNT 1	UM1	Unsigned long
1D02h	2	Contatore CNT 2	Counter CNT 2	UM2	Unsigned long
1D04h	2	Contatore CNT 3	Counter CNT 3	UM3	Unsigned long
1D06h	2	Contatore CNT 4	Counter CNT 4	UM4	Unsigned long
1D08h	2	Contatore CNT 5	Counter CNT 5	UM5	Unsigned long
1D0Ah	2	Contatore CNT 6	Counter CNT 6	UM6	Unsigned long
1D0Ch	2	Contatore CNT 7	Counter CNT 7	UM7	Unsigned long
1D0Eh	2	Contatore CNT 8	Counter CNT 8	UM8	Unsigned long
0F50h	2	Ingresso analogico 1	Analog input 1		Signed long
0F52h	2	Ingresso analogico 2	Analog input 2		Signed long
0F54h	2	Ingresso analogico 3	Analog input 3		Signed long
0F56h	2	Ingresso analogico 4	Analog input 4		Signed long
0F60h	2	Uscita analogica 1	Analog output 1		Signed long
0F62h	2	Uscita analogica 2	Analog output 2		Signed long
0F64h	2	Uscita analogica 3	Analog output 3		Signed long
0F66h	2	Uscita analogica 4	Analog output 4		Signed long

① Leggendo la word all'indirizzo 2200h vengono restituiti 16 bit con significato come da tabella:

① Reading the word at address 2200h will return 16 bits with the following meaning:

Bit	Codice	Allarme	Code	Alarm
0	A01	Bassa tensione rete	A01	Low mains voltage
1	A02	Alta tensione rete	A02	High mains voltage
2	A03	Bassa frequenza rete	A03	Low mains frequency
3	A04	Alta frequenza rete	A04	High mains frequency
4	A05	Asimmetria tensione rete	A05	Voltage asymmetry mains
5	A06	Mancanza fase	A06	Phase failure
6	A07	Errata sequenza fasi	A07	Incorrect phase sequence
7	A08	Mancato avviamento pompa	A08	Pump starting failure
8	A09	Rotore bloccato	A09	Rotor blocked
9	A10	Marcia a secco	A10	Dry running
10	A11	Corrente troppo bassa	A11	Current too low
11	A12	Corrente troppo alta	A12	Current too high
12	A13	Correnti sbilanciate	A13	Unbalanced current
13	A14	Corrente inaspettata	A14	Unexpected current
14	A15	Collegamento TA errato	A15	Wrong CT connection
15	A16	Errore di sistema xx	A16	System error xx

② Leggendo la word all'indirizzo 0x2201 vengono restituiti 16 bit con significato come da tabella:

② Reading the word at address 0x2201 will return 16 bits with the following meaning:

Bit	Codice	Allarme	Code	Alarm
0	A17	Bassa temperatura locale pompe	A17	Low temperature in pump room
1	A18	Alta temperatura locale pompe	A18	High temperature in pump room
2	A19	Riserva idrica	A19	Water reserve
3	A20	Basso livello cisterna	A20	Low water tank level
4	A21	Cisterna vuota	A21	Water tank empty
5	A22	Basso livello serbatoio adescamento	A22	Low level priming tank
6	A23	Sistema non in modalità automatica	A23	System is not in automatic mode
7	A24	Elettropompa in funzione	A24	Fire pump running
8	A25	Pompa non in pressione	A25	Fire pump not in pressure
9	A26	Pompa in pressione	A26	Pump in pressure
10	A27	Richiesta manutenzione 1	A27	Maintenance 1 requested
11	A28	Richiesta manutenzione 2	A28	Maintenance 2 requested
12	A29	Richiesta manutenzione 3	A29	Maintenance 3 requested
13	A30	Valvola aspirazione parzialmente aperta	A30	Suction valve partially opened
14	A31	Valvola mandata parzialmente aperta	A31	Discharge valve partially opened
15	A32	Sprinkler locale pompe in funzione	A32	Sprinkler in pump room activated

③ Leggendo la word all'indirizzo 0x2202 vengono restituiti 16 bit con significato come da tabella:

③ Reading the words starting at address 0x2202 will return 16 bits with the following meaning:

Bit	Codice	Allarme	Code	Alarm
0	A33	Numero massimo avviamenti pompa pilota	A33	Max number of start-up jockey pump
1	A34	Avaria pompa pilota	A34	Jockey pump alarm failure
2	A35	Tempo massimo pompa pilota	A35	Timeout jockey pump
3	A36	Avaria pompa drenaggio	A36	Drainage pump alarm failure
4	A37	Errore di comunicazione	A37	Communication error
5	A38	Errore test pressostato	A38	Pressure input test failed
6	A39	Valvola di test aperta	A39	Valve test opened
7	A40	Potenza troppo bassa	A40	Power too low
8	A41	Potenza troppo alta	A41	Power too high
9	A42	Elettropompa in funzione	A42	Electric fire pump running
10	UA1	Allarme utente 1	UA1	User alarm 1
11	UA2	Allarme utente 2	UA2	User alarm 2
12	UA3	Allarme utente 3	UA3	User alarm 3
13	UA4	Allarme utente 4	UA4	User alarm 4
14	UA5	Allarme utente 5	UA5	User alarm 5
15	UA6	Allarme utente 6	UA6	User alarm 6

④ Leggendo la word all'indirizzo 0x2203 vengono restituiti 16 bit con significato come da tabella:

④ Reading the words starting at address 0x2203 will return 16 bits with the following meaning:

Bit	Codice	Allarme	Code	Alarm
0	UA7	Allarme utente 7	UA7	User alarm 7
1	UA8	Allarme utente 8	UA8	User alarm 8
2	UA9	Allarme utente 9	UA9	User alarm 9
3	UA10	Allarme utente 10	UA10	User alarm 10
4	UA11	Allarme utente 11	UA11	User alarm 11
5	UA12	Allarme utente 12	UA12	User alarm 12
6	-	Non utilizzato	-	Not used
7	-	Non utilizzato	-	Not used
8	-	Non utilizzato	-	Not used
9	-	Non utilizzato	-	Not used
10	-	Non utilizzato	-	Not used
11	-	Non utilizzato	-	Not used
12	-	Non utilizzato	-	Not used
13	-	Non utilizzato	-	Not used
14	-	Non utilizzato	-	Not used
15	-	Non utilizzato	-	Not used

**TABELLA 3:
BIT DI STATO**
(Utilizzabili con funzioni 03 e 04)

INDIRIZZO ADDRESS	WORDS	FUNZIONE	FUNCTION	FORMATO FORMAT
2070h	1	Stato tastiera frontale ❶	Front panel keyboard status ❶	Unsigned integer
2100h	1	Stato ingressi digitali (per pin) (1-16) ❷	Digital inputs status (by pin) (1-16) ❷	Unsigned integer
213Fh	1	Stato ingressi digitali (per pin) (17-20) ❸	Digital inputs status (by pin) ❸	Unsigned integer
2140h	1	Stato uscite digitali (per pin) (1-16) ❹	Digital outputs status (by pin) (1-16) ❹	Unsigned integer
217Fh	1	Stato uscite digitali (per pin) (17-20) ❺	Digital outputs status (by pin) ❺	Unsigned integer

❶ Following table shows meaning of bits of the word at address 2070h:

❶ Leggendo le word all'indirizzo 2070h vengono restituiti 16 bit con significato come da tabella:

Bit	Tasto	Key
0	DESTRA	RIGHT
1	SINISTRA	LEFT
2	FRECCIA GIU	DOWN
3	FRECCIA SU	UP
4	STOP	STOP
5	RESET	RESET
6	Non usato	Not used
7	ENTER	ENTER
8	START	START
9...15	Non usati	Not used

❷ Leggendo le word all'indirizzo 2100h vengono restituiti 16 bit con significato come da tabella:

❷ Following table shows meaning of bits of the word at address 2100h

Bit	Ingresso	Input
0	Ingresso 1	Input 1
1	Ingresso 2	Input 2
2	Ingresso 3	Input 3
3	Ingresso 4	Input 4
4	Ingresso 5	Input 5
5	Ingresso 6	Input 6
6	Ingresso 7	Input 7
7	Ingresso 8	Input 8
8	Ingresso 9	Input 9
9	Ingresso 10	Input 10
10	Ingresso 11	Input 11
11	Ingresso 12	Input 12
12	Ingresso 13	Input 13
13	Ingresso 14	Input 14
14	Ingresso 15	Input 15
15	Ingresso 16	Input 16

❸ Leggendo le word all'indirizzo 2140h vengono restituiti 16 bit con significato come da tabella:

❸ Following table shows meaning of bits of the word at address 2140h:

Bit	Uscita	Output
0	Stato uscita 1	Output 1
1	Stato uscita 2	Output 2
2	Stato uscita 3	Output 3
3	Stato uscita 4	Output 4
4	Stato uscita 5	Output 5
5	Stato uscita 6	Output 6
6	Stato uscita 7	Output 7
7	Stato uscita 8	Output 8
8	Stato uscita 9	Output 9
9	Stato uscita 10	Output 10
10	Stato uscita 11	Output 11
11	Stato uscita 12	Output 12
12	Stato uscita 13	Output 13
13	Stato uscita 14	Output 14
14	Stato uscita 15	Output 15
15	Stato uscita 16	Output 16

❹ Leggendo le word all'indirizzo 213Fh vengono restituiti 16 bit con significato come da tabella:

❹ Following table shows meaning of bits of the word at address 213Fh

Bit	Ingresso	Input
0	Ingresso 17	Input 17
1	Ingresso 18	Input 18
2	Ingresso 19	Input 19
3	Ingresso 20	Input 20
4	Ingresso 21	Input 21
5	Ingresso 22	Input 22
6	Ingresso 23	Input 23
7	Ingresso 24	Input 24
8	Ingresso 25	Input 25
9-15	Non usati	Not used

❺ Leggendo le word all'indirizzo 217F h vengono restituiti 16 bit con significato come da tabella:

❺ Following table shows meaning of bits of the word at address 217F h:

Bit	Uscita	Output
0	Stato uscita 17	Output 17
1	Stato uscita 18	Output 18
2	Stato uscita 19	Output 19
3	Stato uscita 20	Output 20
4-15	Non usato	Not used

TABELLA 4:
COMANDI
(Utilizzabili con funzione 06)

Indirizzo Address	WORDS	STATI	STATUS
4F00h	1	Imposta variabile remota REM1	Set remote variable REM1
4F01h	1	Imposta variabile remota REM2	Set remote variable REM2
.....
4F0Fh	1	Imposta variabile remota REM16	Set remote variable REM16
2F0Ah	1	Simulazione pressione tasti pannello frontale	Front panel keystroke simulation
2F03h	1	Valore 01h: Salvataggio eeprom	Value 01h: Eeprom save
		Valore 04h: reboot	Value 04h: reboot
		Valore 05h: Salvataggio e reboot	Value 05h: save and reboot
2F07h	1	Valore 00h: Reset apparecchio Valore 01h: Reset apparecchio con salvataggio in fram	Value 00h: Reset device Value 01h: Reset device and save Fram
2FF0h	1	Esecuzione comando menu comandi	Command menu execution

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

1 Scrivendo il valore AAh all'indirizzo indicato viene impostata la variabile remota a 1, scrivendo BBh viene impostata a 0.

1 Writing AAh to the indicated address the remote variable will be set to 1, writing BBh the remote variable will be set to 0

2 La seguente tabella indica il valore da scrivere all'indirizzo 2F0Ah per ottenere le corrispondenti funzioni.

2 The following table shows the value to be written to address 2F0Ah to achieve the correspondent function.

Value	SIGNIFICATO	MEANING
0001h	FRECCIA SU	UP key
0002h	RESET	RESET key
0004h	DESTRA	RIGHT key
0020h	STOP	STOP key
0100h	START	START key
0200h	FRECCIA GIU	DOWN key
0400h	ENTER	ENTER key
0800h	SINISTRA	LEFT key

3 Scrivendo il valore da 0 a 15 all'indirizzo indicato viene eseguita la corrispondente funzione

3 Writing value between 0 and 15 to the indicated address, the correspondent command will be executed

SIGNIFICATO	MEANING	SIGNIFICATO	MEANING
0 C01 Reset manutenzione ore 1	C01 Reset maintenance 1	8 C09 Reset contatore avviamenti	C09 Reset number starts counter
1 C02 Reset manutenzione ore 2	C02 Reset maintenance 2	9 C10 Reset valori massimi e minimi	C10 Reset MAX MIN values
2 C03 Reset manutenzione ore 3	C03 Reset maintenance 3	10 C11 Reset lista eventi	C11 Reset event list
3 C04 Reset contaore motore parziale	C04 Reset electric partial hour counter	11 C12 Ripristino parametri a default	C12 Setup to default
4 C05 Reset contatori generici CNTx	C5 Reset generic counters CNTx	12 C13 Salva parametri nella memoria backup	C13 Backup setup
5 C06 Reset stato limiti LIMx	C06 Reset limits status LIMx	13 C14 Ricarica parametri dalla memoria backup	C14 Restore setup
6 C07 Reset contaore motore totale	C07 Reset electric total hour counter	14 C15 Forzatura I/O	C15 Forced I/O
7 C08 Impostazione contaore motore	C08 Set pump hours	15 C16 Azzeramento programma PLC	C16 Reset PLC program

TABELLA 5:
STATO GLOBALE DISPOSITIVO
(Utilizzabili con funzioni 03 e 04)

Indirizzo Address	WORDS	STATI	STATUS	FORMATO FORMAT
2210H	1	Stato globale dispositivo (bit 0-bit15)	Device global status(bit 0-bit15)	Unsigne integer

TABLE 5:
DEVICE GLOBAL STATUS
(To be used with function 03 e 04)

4 Leggendo 1 word agli indirizzi 2210H vengono restituiti 16 Bit con significato come da tabella

2 Reading one word at address 2210H will return 16 bits with the following mean

BIT	SIGNIFICATO	MEANING
Bit 0	Dispositivo in OFF	Device in OFF mode
Bit 1	Dispositivo in MAN	Device in MAN mode
Bit 2	Dispositivo in AUT	Device in AUT mode
Bit 3	Allarme globale	Global alarm
Bit 4	Allarme tipo A	Type A alarm
Bit 5	Allarme tipo B	Type B alarm
Bit 6	Allarme sirena	Siren alarms
Bit 7	Mancanza tensione	Mains failure
Bit 8	Mancato avviamento	Failure to start
Bit 9	Motore avviato	Pump running
Bit 10	Richiesta avviamento	Starting request
Bit 11	Test automatico in corso	Automatic test in progress
Bit 12	Clock 100ms	Clock 100ms
Bit 13	Stop ritardo abilitato	Stop delay enabled
Bit 14	(non usato)	(not used)
Bit 15	(non usato)	(not used)

TABELLA 6:
OROLOGIO DATARIO

(Utilizzabili con funzioni 04 e 06)

Per rendere effettivi i cambiamenti, memorizzare le impostazioni utilizzando l'apposito comando descritto nella tabella 4.

Indirizzo Address	WORDS	FUNZIONE	FUNCTION	RANGE
28F0H	1	Anno	Year	2000..2099
28F1H	1	Mese	Month	1-12
28F2H	1	Giorno	Day	1-31
28F3H	1	Ora	Hours	0-23
28F4H	1	Minuti	Minutes	0-59
28F5H	1	Secondi	Seconds	0-59

LETTURA LISTA EVENTI

Per leggere gli eventi bisogna svolgere la seguente procedura:

1. Eseguire la lettura di 1 registro con la **funzione 4** all'indirizzo **5030H**, il byte più significativo (MSB) indica quanti eventi sono memorizzati (valore compreso tra 0 e 128), il byte meno significativo (LSB) viene incrementato ogni volta che un evento viene salvato (valore compreso tra 0 e 128). Una volta memorizzati 128 eventi l'MSB resterà a 128 mentre l'LSB tornerà a zero e poi continuerà ad incrementare.
2. Impostare l'indice dell'evento che si vuole leggere (minore del numero massimo di eventi memorizzati), per fare questo bisogna eseguire la **funzione 6** all'indirizzo **5030H**, specificando quale evento leggere.
3. Eseguire una lettura di 43 registri (con un'unica **funzione 4**) all'indirizzo **5032H**
4. Il valore tornato è una stringa di 86 caratteri ASCII, che riportano la stessa descrizione dell'evento visibile sul display dell'FFL...EP. L'indice dell'evento che si vuole leggere viene incrementato in automatico dopo la lettura del registro **5032H**, al fine di velocizzare il download degli eventi.
5. Se si vuole leggere l'evento successivo eseguire il punto 3, se si vuole leggere un qualsiasi altro evento eseguire il passo 2.

ESEMPIO

Passo 1: Lettura eventi memorizzati.

MASTER Funzione = 4 (04H)
Indirizzo = 5030H (5030H - 0001H = 502FH)
Nr. registri = 1 (01H)

01	04	50	2F	00	01	11	03
----	----	----	----	----	----	----	----

FFL800EP Funzione = 4
Nr. byte. = 2 (02h)
MSB = 128 (80h)
LSB = 101 (65h)

01	04	02	80	65	18	DB
----	----	----	----	----	----	----

Passo 2: Impostare l'indice dell'evento da leggere.

MASTER Funzione = 6(06H)
Indirizzo = 5030H (5030H - 0001H = 502FH)
Valore = 1 (01H)

01	06	50	2F	00	01	68	C3
----	----	----	----	----	----	----	----

FFL800EP Funzione = 6
Indirizzo = 5030H (5030H - 0001H = 502FH)
Valore = 1 (01H)

01	06	50	2F	00	01	68	C3
----	----	----	----	----	----	----	----

Passo 3: Leggere l'evento.

MASTER Funzione = 4 (04H)
Indirizzo = 5032H (5032H - 0001H = 5031H)
Nr. registri = 43 (2BH)

01	04	50	31	00	2B	F0	DA
----	----	----	----	----	----	----	----

FFL800EP Funzione = 4 (04H)
Nr. byte = 86 (56H)
Stringa = 2018/01/01;14:44:45;E1100,CAMBIO MODALITÀ IN:
MODALITÀ OFF

TABLE 6:
REAL TIME CLOCK

(To be used with functions 04 and 06)

To make effective the changes, store them using the dedicated command described in table 4.

EVENT LOG READING

To read the events must do the following:

1. Perform the read of 1 register by using the **function 4** at address **5030H**, the most significant byte (MSB) indicates how many events are stored (value between 0 to 128), the least significant byte (LSB) is incremented each time an event is saved (value between 0 to 128). Once stored the 128 events the MSB will remain at 128 while the LSB will back to zero and after will continue to increase.
2. Set the index of the event that you want to read (less than the maximum number of events stored), to do this you performe the **function 6** at **5030H**, specifying which event read.
3. Perform a read of 43 registers (with a single **function 4**) at address **5032H**
4. The value returned is a string of 86 ASCII characters, showing the same event description FFL...EP visible on the display. The index of the event to be read is incremented automatically after a reading of the register **5032H**, in order to speed up the download of events.
5. If you want to read the next event performing step 4, if you want to read any other event do step 3.

EXAMPLE

Step 1: Reading events stored.

MASTER Function = 4 (04H)
Address = 5030H (5030H - 0001H = 502FH)
Nr. registers = 1 (01H)

01	04	50	2F	00	01	11	03
----	----	----	----	----	----	----	----

FFL800EP Function = 4
Nr. bytes. = 2 (02h)
MSB = 128 (64h)
LSB = 101 (65h)

01	04	02	80	65	18	DB
----	----	----	----	----	----	----

Step 2: Set the index of the event to read.

MASTER Function = 6(06H)
Address = 5030H (5030H - 0001H = 502FH)
Value = 1 (01H)

01	06	50	2F	00	01	68	C3
----	----	----	----	----	----	----	----

FFL800EP Function = 6
Address = 5030H (5030H - 0001H = 502FH)
Value = 1 (01H)

01	06	50	2F	00	01	68	C3
----	----	----	----	----	----	----	----

Step 3: Read the event.

MASTER Function = 4 (04H)
Address = 5032H (5032H - 0001H = 5031H)
Nr. registers = 43 (2BH)

01	04	50	31	00	2B	F0	DA
----	----	----	----	----	----	----	----

FFL800EP Function = 4 (04H)
Nr. bytes = 86 (56H)
String = 2018/01/01;14:44:45;E1100,MODE CHANGE TO: MAN MODE

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 FFL800EP.

PROCEDURA PER LA LETTURA DEI PARAMETRI

1. Scrivere il valore del menu che si vuole leggere tramite la **funzione 6** all'indirizzo **5000H**.
2. Scrivere il valore del sottomenu (se esiste) che si vuole leggere tramite la **funzione 6** all'indirizzo **5001H**.
3. Scrivere il valore del parametro che si vuole leggere tramite la **funzione 6** all'indirizzo **5002H**.
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**.
2. Scrivere il valore del sottomenu (se esiste) che si vuole modificare tramite la **funzione 6** all'indirizzo **5001H**.
3. Scrivere il valore del parametro che si vuole modificare tramite la **funzione 6** all'indirizzo **5002H**.
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 è necessario memorizzare i valori in EEPROM, utilizzando l'apposito comando descritto nella tabella 4. (scrivere il valore 5 con la **funzione 6** all'indirizzo **2F03H**)

TIPO DI PARAMETRO	NUMERO REGISTRI
Testo lunghezza 6 caratteri	3 registri (6 byte)
Testo lunghezza 16 caratteri	8 registri (16 byte)
Testo lunghezza 20 caratteri	10 registri (20 byte)
Abs(Valore numerico) < 32768	1 registri (2 byte)
Abs(Valore numerico) > 32768	2 registri (4 byte)
Indirizzo IP	2 registri (4 byte)

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

ESEMPIO

Impostare a 8 il valore del parametro M08.01.01

Passo 1: Impostazione menu 08.

MASTER	Funzione = 6								
	Indirizzo = 5000H (5000H - 0001H = 4FFFH)								
	Valore = 8 (08H)								
	<table border="1"><tr><td>01</td><td>06</td><td>4F</td><td>FF</td><td>00</td><td>08</td><td>AE</td><td>E8</td></tr></table>	01	06	4F	FF	00	08	AE	E8
01	06	4F	FF	00	08	AE	E8		

FFL800EP	Funzione = 6								
	Indirizzo = 5000H (5000H - 0001H = 4FFFH)								
	Valore = 8 (08H)								
	<table border="1"><tr><td>01</td><td>06</td><td>4F</td><td>FF</td><td>00</td><td>08</td><td>AE</td><td>E8</td></tr></table>	01	06	4F	FF	00	08	AE	E8
01	06	4F	FF	00	08	AE	E8		

Passo 2: Impostazione sottomenu 01.

MASTER	Funzione = 6								
	Indirizzo = 5001H (5001H - 0001H = 5000H)								
	Valore = 1 (01H)								
	<table border="1"><tr><td>01</td><td>06</td><td>50</td><td>00</td><td>00</td><td>01</td><td>59</td><td>0A</td></tr></table>	01	06	50	00	00	01	59	0A
01	06	50	00	00	01	59	0A		

FFL800EP	Funzione = 6								
	Indirizzo = 5001H (5001H - 0001H = 5000H)								
	Valore = 1 (01H)								
	<table border="1"><tr><td>01</td><td>06</td><td>50</td><td>00</td><td>00</td><td>01</td><td>59</td><td>0A</td></tr></table>	01	06	50	00	00	01	59	0A
01	06	50	00	00	01	59	0A		

Passo 3: Impostazione parametro 01.

MASTER	Funzione = 6								
	Indirizzo = 5002H (5002H - 0001H = 5001H)								
	Valore = 1 (01H)								
	<table border="1"><tr><td>01</td><td>06</td><td>50</td><td>01</td><td>00</td><td>01</td><td>08</td><td>CA</td></tr></table>	01	06	50	01	00	01	08	CA
01	06	50	01	00	01	08	CA		

FFL800EP	Funzione = 6								
	Indirizzo = 5002H (5002H - 0001H = 5001H)								
	Valore = 1 (02H)								
	<table border="1"><tr><td>01</td><td>06</td><td>50</td><td>01</td><td>00</td><td>01</td><td>08</td><td>CA</td></tr></table>	01	06	50	01	00	01	08	CA
01	06	50	01	00	01	08	CA		

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 FFL800EP 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**.
2. Write the value of the submenu (if it is present) that you want to read by using the **function 6** at address **5001H**.
3. Write the value of the parameter that you want to read by using the **function 6** at address **5002H**.
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**.
2. Write the value of the submenu (if it is present) that you want to change by using the **function 6** at address **5001H**.
3. Write the value of the parameter that you want to change by using the **function 6** at address **5002H**.
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 3. (write value 5 by using **function 6** at address **2F03H**)

TYPE OF PARAMETER	NUMBER OF REGISTER
Text length 6 characters	3 registers (6 byte)
Text length 16 characters	8 registers (16 byte)
Text length 20 characters	10 registers (20 byte)
Abs(Numeric value) < 32768	1 registers (2 byte)
Abs(Numeric value) > 32768	2 registers (4 byte)
IP address	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**.

EXAMPLE

Set to 8 the value of parameter M08.01.01

Step 1: Set menu 08.

MASTER	Function = 6								
	Address = 5000H (5000H - 0001H = 4FFFH)								
	Value = 8 (08H)								
	<table border="1"><tr><td>01</td><td>06</td><td>4F</td><td>FF</td><td>00</td><td>08</td><td>AE</td><td>E8</td></tr></table>	01	06	4F	FF	00	08	AE	E8
01	06	4F	FF	00	08	AE	E8		

FFL800EP	Function = 6								
	Address = 5000H (5000H - 0001H = 4FFFH)								
	Value = 8 (08H)								
	<table border="1"><tr><td>01</td><td>06</td><td>4F</td><td>FF</td><td>00</td><td>08</td><td>AE</td><td>E8</td></tr></table>	01	06	4F	FF	00	08	AE	E8
01	06	4F	FF	00	08	AE	E8		

Step 2: Set submenu 01.

MASTER	Function = 6								
	Address = 5001H (5001H - 0001H = 5000H)								
	Value = 1 (01H)								
	<table border="1"><tr><td>01</td><td>06</td><td>50</td><td>00</td><td>00</td><td>01</td><td>59</td><td>0A</td></tr></table>	01	06	50	00	00	01	59	0A
01	06	50	00	00	01	59	0A		

FFL800EP	Function = 6								
	Address = 5001H (5001H - 0001H = 5000H)								
	Value = 1 (01H)								
	<table border="1"><tr><td>01</td><td>06</td><td>50</td><td>00</td><td>00</td><td>01</td><td>59</td><td>0A</td></tr></table>	01	06	50	00	00	01	59	0A
01	06	50	00	00	01	59	0A		

Step 3: Set parameter 01.

MASTER	Function = 6								
	Address = 5002H (5002H - 0001H = 5001H)								
	Value = 1 (01H)								
	<table border="1"><tr><td>01</td><td>06</td><td>50</td><td>01</td><td>00</td><td>01</td><td>08</td><td>CA</td></tr></table>	01	06	50	01	00	01	08	CA
01	06	50	01	00	01	08	CA		

FFL800EP	Function = 6								
	Address = 5002H (5002H - 0001H = 5001H)								
	Value = 1 (02H)								
	<table border="1"><tr><td>01</td><td>06</td><td>50</td><td>01</td><td>00</td><td>01</td><td>08</td><td>CA</td></tr></table>	01	06	50	01	00	01	08	CA
01	06	50	01	00	01	08	CA		

Passo 4: Impostazione valore 8.

MASTER Funzione = 16 (10H)
 Indirizzo = 5004H (5004H – 0001H =5003H)
 Nr. registri = 1 (01H)
 Nr. byte = 2 (02H)
 Valore = 8 (0008H)

01	10	50	03	00	01	02	00	08	F7	A0
----	----	----	----	----	----	----	----	----	----	----

FFL800EP Funzione = 16 (10H)
 Indirizzo = 5004H (5004H – 0001H =5003H)
 Valore = 2 (02H)

01	10	50	03	00	08	E0	C9
----	----	----	----	----	----	----	----

Passo 5: Salvataggio e riavvio.

MASTER Funzione = 6 (06H)
 Indirizzo = 2F03H (2F03H – 0001H =2F02H)
 Valore = 5 (05H)

01	06	2F	02	00	05	E0	DD
----	----	----	----	----	----	----	----

FFL800EP Nessuna risposta.

Step 4: Set value 8.

MASTER Function = 16 (10H)
 Address = 5004H (5004H – 0001H =5003H)
 Nr. register = 1 (01H)
 Nr. bytes = 2 (02H)
 Value = 8 (0008H)

01	10	50	03	00	01	02	00	08	F7	A0
----	----	----	----	----	----	----	----	----	----	----

FFL800EP Function = 16 (10H)
 Address = 5004H (5004H – 0001H =5003H)
 Value = 2 (02H)

01	10	50	03	00	08	E0	C9
----	----	----	----	----	----	----	----

Step 5: Save and reboot.

MASTER Function = 6 (06H)
 Address = 2F03H (2F03H – 0001H =2F02H)
 Value = 5 (04H)

01	06	2F	02	00	05	E0	DD
----	----	----	----	----	----	----	----

FFL800EP No answer.