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

(I)

FFL700DP
FFL800DP

UNITA' DI CONTROLLO MOTOPOMPA ANTINCENDIO (EN12845)

PROTOCOLLO DI COMUNICAZIONE MODBUS®

(GB)

FFL700DP
FFL800DP

DIESEL ENGINE FIRE PUMP CONTROLLERS (EN12845)

COMMUNICATION PROTOCOL MODBUS®

PROTOCOLLO MODBUS®

Le unità di controllo di motopompe diesel antincendio FFL…DP 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 FFL800DP:

- EXP 10 12T RS485
- EXP 10 13T 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 M17. E possibile configurare 3 porte di comunicaziione (n=1…3). La porta di comunicazione 1 è riservata alla comunicazione RS485 integrata.

MENU M17 – COMUNICAZIONE
Comunicazione seriale RS485

PAR	Funzione	Default	Range
P17.n.01	Indirizzo nodo	01	01-255
P17.n.02	Velocità seriale	9600	1200 2400 4800 9600 19200 38400 57600 115200
P17.n.03	Formato dati	8 bit – n	8 bit, no parità 8 bit, dispari 8 bit, pari 7 bit, dispari 7 bit, pari
P17.n.04	Bit di stop	1	1-2
P17.n.05	Protocollo	Modbus RTU	Modbus RTU Modbus ASCII Modbus TCP

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

PAR	Funzione	Default	Range
P17.n.06	Indirizzo IP	192.168.1.1	000.000.000.000 – 255.255.255.255
P17.n.07	Subnet mask	0.0.0.0	000.000.000.000 – 255.255.255.255
P17.n.08	Porta IP	1001	0-32000
P17.n.09	Funzione canale	Slave	Slave Gateway
P17.n.10	Client / server	Server	Client Server
P17.n.11	Indirizzo IP remoto	0.0.0.0	000.000.000.000 – 255.255.255.255
P17.n.12	Porta IP remota	1001	0-32000
P17.n.13	Indirizzo gateway IP	0.0.0.0	000.000.000.000 – 255.255.255.255

MODBUS® PROTOCOL

The diesel engine fire pump controllers FFL...DP supports the communication protocols Modbus RTU®, Modbus ASCII®, ModbusTCP® on optical interface, on the built in RS485 coomunicaqtion port and on the expansion modules for FFL800DP:

- EXP 10 12T RS485
- EXP 10 13T 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 M17 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 M17 – COMMUNICATION
Serial comunication RS485

PAR	Function	Default	Range
P17.n.01	Node address	01	01-255
P17.n.02	Serial port speed	9600	1200 2400 4800 9600 19200 38400 57600 115200
P17.n.03	Data format	8 bit – n	8 bit –no par. 8 bit, odd 8 bit, even 7 bit, odd 7 bit, even
P17.n.04	Stop bits	1	1-2
P17.n.05	Protocol	Modbus RTU	Modbus RTU Modbus ASCII Modbus TCP

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

PAR	Function	Default	Range
P17.n.06	IP address	192.168.1.1	000.000.000.000 – 255.255.255.255
P17.n.07	Subnet mask	0.0.0.0	000.000.000.000 – 255.255.255.255
P17.n.08	IP port	1001	0-32000
P17.n.09	Channel funcion	Slave	Slave Gateway
P17.n.10	Client / server	Server	Client Server
P17.n.11	Remote IP address	0.0.0.0	000.000.000.000 – 255.255.255.255
P17.n.12	Remote IP port	1001	0-32000
P17.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 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.
- La lunghezza massima consentita per il campo dati e' 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.

FFL800DP 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…DP con indirizzo 01 la tensione di batteria A, che si trova alla locazione 3854 (0F0E Hex), il messaggio da spedire è il seguente:

01	04	0F	0D	00	02	E3	1C

Dove:
01= indirizzo slave
04 = funzione di lettura locazione
0F 0D = indirizzo della locazione diminuito di un'unità, contenete il numero di allarmi commutazione dell'interruttore 1
00 02 = numero di registri da leggere a partire dall'indirizzo 0F0E
E3 1C = checksum CRC

La risposta dell' FFL…DP è la seguente:

01	04	04	00	00	04	B7	B9	32

Dove:
01= indirizzo della FFL…DP (Slave 01)
04 = funzione richiesta dal Master
04 = numero di byte inviati dalla FFL…DP

00 00 04 B7 = il valore della tensione di batteria A in esadecimale = 12.07 VDC
B9 32 = checksum CRC

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.
- 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.

FFL800DP 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 number battery A voltage, which resides at location 3854 (0F0E Hex), from the FFL…DP with serial address 01, the message to send is the following:

01	04	0F	0D	00	02	E3	1C

Where:
01= slave address
04 = Modbus® function 'Read input register'
0F 0D = Address of the required register (number of switching alarms of breaker 1) decreased by one
00 02 = Number of registers to be read beginning from address 0F0E
E3 1C = CRC Checksum

The FFL...DP answer is the following:

01	04	04	00	00	04	B7	B9	32

Where:
01 = FFL...DP address (Slave 01)
04 = Function requested by the master
04 = Number of bytes sent by the FFL...DP

00 00 04 B7 = Hex value battery A coltage = 12.07 VDC
B9 32 = 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 rispnderà 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 puo' essere utilizzata solo con i registri d'indirizzo superiore a 1000 Hex. E' 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 Tabelle 4.

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 dispositivo.

Richiesta Master:

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

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 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.

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

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 device.

Master query:

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

La tabella seguente riporta il significato del byte inviato dall'FFL800DP 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:	
Indirizzo slave	08h
Funzione	10h
MSB Indirizzo registro	20h
LSB Indirizzo registro	01h
MSB Numero registri	00h
LSB Numero registri	02h
Numero di byte (è il doppio di quelli sopra)	04h
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

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	73h
Data 02 (Sw revision)	01h
Data 03 (Hardware revision)	00h
Data 04 (Parameter revision)	01h
Data 05 (type of device) ❷	07h
Data 06 (reserved)	00h
Data 07 (reserved)	00h
Data 08 (reserved)	00h
LSB CRC	5Eh
MSB CRC	CDh

❶ 113 – 71h =FFL700DP / 115 - 73h = FFL800DP
❷ 7 – 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 FFL800DP as answer:

BIT	MEANING
0	Not used
1	Operative mode MAN
2	Operative mode AUT
3	Not used
4	On error
5	Engine 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 address	08h
Function	10h
MSB register address	20h
LSB register address	01h
MSB register number	00h
LSB register number	02h
Number of byte ((it is the double of the above)	04h
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

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	73h
Dato 02 (Revisione software)	01h
Dato 03 (Revisione hardware)	00h
Dato 04 (Revisione parametri)	01h
Dato 05 (tipoolgia di prodotto) ❷	07h
Dato 06 (riservato)	00h
Dato 07 (riservato)	00h
Dato 08 (riservato)	00h
LSB CRC	5Eh
MSB CRC	CDh

❶ 113 – 71h =FFL700DP / 115 - 73h = FFL800DP
❷ 7 – 07h = FFL series

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 P17.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

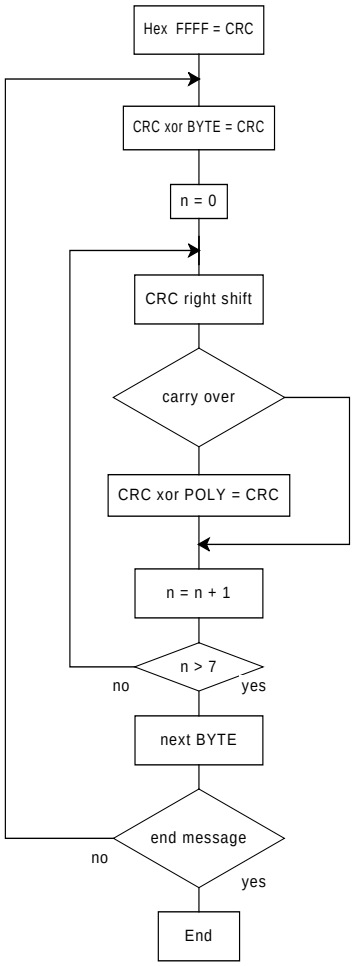
The Modbus® ASCII protocol is normally used in application that require to communicate through a couple of modems.

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 P17.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 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 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 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 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 VAC	VAC Voltage	V/100	Unsigned long
0006h	2	RPM	RPM	RPM	Unsigned long
0F80h	2	Ore di lavoro parziali	Partial working hours	hours	Unsigned long
0F84h	2	Ore di lavoro totali	Total working hours	hours	Unsigned long
0FA0h	2	Sensore temperatura motore 1	Engine temperature sensor 1	°C / °F	Signed long
0FA2h	2	Sensore pressione olio	Oil pressure sensor	(Bar / psi) / 10	Unsigned long
0FA4h	2	Sensore livello carburante	Fuel level sensor	% / l / gal	Unsigned long
0FA6h	2	Sensore temperatura motore 2	Engine temperature sensor 2	°C / °F	Signed long
0F12h	2	Sonda di temperature interna	Internal temperature probe	°C / °F	Signed long
0F0Ah	2	Sonda di temperatura esterna	External temperature probe	°C / °F	Signed long
0F1Eh	2	Temperatura a display	Temperature on the display	(°C / °F) /100	Signed long
0F0Eh	2	Tensione batteria A	Battery A voltage	VDC/100	Unsigned long
0F1Ah	2	Tensione batteria B	Battery B voltage	VDC/100	Unsigned long
0F10h	2	Tensione D+	D+ volteage	VDC	Unsigned long
0F18h	2	Feedback pignone	Feedback pignone	VDC	Unsigned long
0F28h	2	Riserva idrica	Water reserve	% / l	Unsigned long
2200h ❶	1	Allarmi da A01 a A16	Alarms from A01 to A16		Unsigned int
2201h ❷	1	Allarmi da A17 a A32	Alarms from A17 to A32		Unsigned int
2202h ❸	1	Allarmi da A33 a A48	Alarms from A33 to A48		Unsigned int
2203h ❹	1	Allarmi da A49 a A60	Alarms from A49 to A60		Unsigned int
2204h ❺	1	Allarmi da A69 a UA06	Alarms from A69 to UA06		Unsigned int
2205h ❻	1	UA07 - UA08	UA07 - UA08		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

❶ *Leggendo la word all'indirizzo 2200h vengono restituiti 16 bit con significato come da tabella:*
❶ *Reading the word at address 2200h will return 16 bits with the following meaning:*

Bit	Codice	Allarme	Code	Alarm
0	A01	Preallarme temperatura motore 1 (sensore analogico)	A01	Engine temperature 1 warning (Analog sensor)
1	A02	Alta temperatura motore 1 (sensore analogico)	A02	Engine temperature 1 high (Analog sensor)
2	A03	Guasto sensore temperatura 1 (sensore analogico)	A03	Engine temperature 1 sensor failure (Analog sensor)
3	A04	Bassa temperatura motore 1 (sensore analogico)	A04	Engine temperature 1 low (Analog sensor)
4	A05	Preallarme temperatura motore 2 (sensore analogico)	A05	Engine temperature 2 warning (Analog sensor)
5	A06	Alta temperatura motore 2 (sensore analogico)	A06	Engine temperature 2 high (Analog sensor)
6	A07	Guasto sensore analogico temperatura 2	A07	Engine temperature 2 sensor failure (Analog sensor)
7	A08	Bassa temperatura motore 2 (sensore analogico)	A08	Engine temperature 2 low (Analog sensor)
8	A09	Alta temperatura motore (sensore digitale)	A09	Engine temperature 2 high (Digital sensor)
9	A10	Temperatura motore troppo bassa (digitale). Avaria riscaldatore	A10	Engine temperature too low (Digital sensor). Heater failure
10	A11	Preallarme pressione olio (sensore analogico)	A11	Oil pressure warning (Analog sensor)
11	A12	Bassa pressione olio (sensore analogico)	A12	Low oil pressure (Analog sensor)
12	A13	Guasto sensore analogico pressione	A13	Analog pressure sensor failure
13	A14	Bassa pressione olio (sensore digitale)	A14	Low oil pressure (Digital sensor)
14	A15	Guasto sensore digitale pressione olio	A15	Digital pressure sensor failure
15	A16	Preallarme basso livello carburante (sensore analogico)	A16	Warning low fuel level (Analog sensor)

❷ *Leggendo la word all'indirizzo 0x2201 vengono restituiti 16 bit con significato come da tabella:*
❷ *Reading the word at address 0x2201 will return 16 bits with the following meaning:*

Bit	Codice	Allarme	Code	Alarm
0	A17	Basso livello carburante (sensore analogico)	A17	Low fuel level (Analog sensor)
1	A18	Preallarme alto livello carburante (sensore analogico)	A18	Warning high fuel level (Analog sensor)
2	A19	Alto livello carburante (sensore analogico)	A19	High fuel level (Analog sensor)
3	A20	Guasto sensore analogico livello	A20	Analog fuel level sensor failure
4	A21	Basso livello carburante (sensore digitale)	A21	Low fuel level (Digital sensor)
5	A22	Basso livello liquido radiatore	A22	Low level of radiator fluid
6	A23	Avaria segnale "W / pick-up"	A23	"W / pick-up" signal failure
7	A24	"W / pick-up" scollegato	A24	"W / pick-up" disconnected
8	A25	Bassa velocità motore "W / pick-up"	A25	Low engine speed "W / pick-up"
9	A26	Alta velocità motore "W / pick-up"	A26	High engine speed "W / pick-up"
10	A27	Pignone inserito (feedback on durante pausa)	A27	Pinion inserted (Feedback on during pause)
11	A28	Pignone non inserito (feedback off durante cranking)	A28	Pinion not inserted (Feedback off during cranking)
12	A29	Sensore pignone scollegato	A29	Pinion sensor disconnected

13	A30	Acqua nel carburante	A30	Water in the fuel
14	A31	Mancato avviamento	A31	Failure to start
15	A32	Arresto inaspettato	A32	Unexpected stop

❸ *Leggendo la word all'indirizzo 0x2202 vengono restituiti 16 bit con significato come da tabella:*
❸ *Reading the words starting at address 0x2202 will return 16 bits with the following meaning:*

Bit	Codice	Allarme	Code	Alarm
0	A33	Mancato arresto	A33	Failure to stop
1	A34	Tensione batteria A alta	A34	High battery A voltage
2	A35	Tensione batteria A bassa	A35	Low battery A voltage
3	A36	Batteria A inefficiente	A36	Battery A inefficient
4	A37	Allarme da carica batteria A	A37	Alarm from battery charger A
5	A38	Tensione batteria B alta	A38	High battery B voltage
6	A39	Tensione batteria B bassa	A39	Low battery B voltage
7	A40	Batteria B inefficiente	A40	Battery B inefficient
8	A41	Allarme da carica batteria B	A41	Alarm from battery charger B
9	A42	Avaria alternatore carica batteria	A42	Battery charger alternator failure
10	A43	Tensione ausiliaria troppo bassa	A43	Auxiliary voltage too low
11	A44	Tensione ausiliaria troppo alta	A44	Auxiliary voltage too high
12	A45	Errore di sistema	A45	System error
13	A46	Temperatura ambiente troppo bassa (analogica)	A46	Room temperature too low (analog)
14	A47	Temperatura ambiente troppo alta (analogica)	A47	Room temperature too high (Analog sensor)
15	A48	Riserva idrica (digitale)	A48	Water reserve (Digital sensor)

❹ *Leggendo la word all'indirizzo 0x2203 vengono restituiti 16 bit con significato come da tabella:*
❹ *Reading the word at address 0x2203 will return 16 bits with the following meaning:*

Bit	Codice	Allarme	Code	Alarm
0	A49	Basso livello riserva idrica (analogico)	A49	Water reserve low (Analog sensor)
1	A50	Riserva idrica vuota (analogico)	A50	Water reserve high (Analog sensor)
2	A51	Livello basso serbatoio adescamento	A51	Low priming tank level
3	A52	Alimentazione uscite disconnessa	A52	Supply of outputs disconnected
4	A54	Sistema non in modalità automatica (per 24 ore)	A54	System not in automatic mode (for 24 hours)
5	A55	Motopompa in funzione	A55	Engine pump running
6	A56	Pompa in avaria	A56	Pump failure
7	A57	Pompa in pressione (con motore spento)	A57	Pump in pressure (with engine off)
8	A58	Richiesta manutenzione 1	A58	Maintenance request 1
9	A59	Richiesta manutenzione 2	A59	Maintenance request 2
10	A60	Richiesta manutenzione 3	A60	Maintenance request 3
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

❺ *Leggendo la word all'indirizzo 0x2204 vengono restituiti 16 bit con significato come da tabella:*
❺ *Reading the word at address 0x2204 will return 16 bits with the following meaning:*

Bit	Codice	Allarme	Code	Alarm
0		Non utilizzato		not used
1		Non utilizzato		not used
2		Non utilizzato		not used
3	A69	Valvola aspirazione parzialmente aperta	A69	Suction valve partially open
4	A70	Valvola mandata parzialmente aperta	A70	Delivery valve partially open
5	A71	Sprinkler locale pompe	A71	Room pump sprinkler alarm
6	A72	Allarme avviamenti pompa Jockey	A72	Jockey pump starts alarm
7	A73	Allarme termico pompa jockey	A73	Thermal alarm jockey pump
8	A74	Allarme pompa di drenaggio	A74	Drain pump alarm
9	A75	Perdita liquido carburante	A75	Fuel tank leak
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

❻ *Leggendo la word all'indirizzo 0x2205 vengono restituiti 16 bit con significato come da tabella:*
❻ *Reading the word at address 0x2205 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

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

TABLE 3:
STATUS BITS
(To be used with functions 03 and 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) ❷	Unsigned integer
213Fh	1	Stato ingressi digitali (per pin) (17- 25) ❸	Digital inputs status (by pin) ❸	Unsigned integer
2140h	1	Stato uscite digitali (per pin) (1-16)❹	Digital outputs status (by pin) 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	SINSITRA	LEFT
2	FRECCIA GIU	DOWN
3	FRECCIA SU	UP
4	MAN/TEST	MAN/TEST
5	RESET	RESET
6	STOP	STOP
7	ENTER	ENTER
8	BAT A	BAT A
9	BAT B	BAT B
10…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 213Fh vengono restituiti 16 bit con significato come da tabella:

❸ 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

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

❹Following 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

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

❺Following 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)

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

Indirizzo Address	WORDS	STATI	STATUS
4F00 H	1	Imposta variabile remora REM1❶	Set remote variable REM1❶
4F01 H	1	Imposta variabile remora REM2	Set remote variable REM2
……			
4F0FH	1	Imposta variabile remora REM16	Set remote variable REM16
2F0AH	1	Simulazione pressione tasti pannello frontale❸	Front panel keystorke simulation ❸
2F03H	1	Valore 01h: Salvataggio eeprom	Value 01h: Eeprom save
		Valore 04h: reboot	Value 04h: reboot
		Valore 05h: Salvataggio e reboot	Valore 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 comandi❹	Command menu execution❹
28FAH	1	Valore 01h: Salvataggio impostazione orologio datario	Value 01H: Save real time clock setting

❶ Scrivendo il valore AAH all'indirizzo indicato viene impostata la variabile remota a 1, scrivendo BBH viene impostata a 0.

❶ Writing AAh to the indicated address the remte variable will be set to 1, writing BBH the remote variable will be set to 0

❸ La seguente tabella indica il valore da scrivere all'indirizzo 2F0AH per ottenere le corrispondenti funzioni.

❸ The following table shows the value to be written to address 2F0AH to achieve the correspondent function.

Value	SIGNIFICATO	MEANING
0001h	DESTRA	Key UP
0002h	SINSITRA	Key RIGHT
0004h	FRECCIA GIU	Key DOWN
0008h	FRECCIA SU	Key ENTER
0010h	MAN/TEST	Key LEFT
0020h	RESET	MAN mode
0040h	STOP	OFF mode
0080h	ENTER	AUT mode
0100h	BAT A	BAT A
0200h	BAT B	BAT B

❹Scrivendo il valore da 0 a 15 all'indirizzo indicato viene eseguito la corrispondente funzione

❹Writting value between 0 and 15 to the indicated address, the correspondent command will be executed

	SIGNIFICATO	MEANING		SIGNIFICATO	MEANING
0	Reset manutenzione ore 1	Reset maintenance 1	8	Reset contatore avviamenti	
1	Reset manutenzione ore 2	Reset maintenance 2	9	Reset lista eventi	Reset events list
2	Reset manutenzione ore 3	Reset maintenance 3	10	Ripristino parametri a default	Reset default parameters
3	Reset ore motore parziale	Reset partial engine hour	11	Salva parametri nella memoria backup	Save parameters in backup memory
4	Reset contatori generici CNTx	Reset generic counters CNTx	12	Ricarica parametri dalla memoria backup	Reload parameters from backup memory
5	Reset stato limiti LIMx	Reset LIMx limits	13	Non usato	Not used
6	Reset ore motore totale	Total engine hour reset	14	Non usato	Not used
7	Non usato	Not used	15	Reset programma PLC	Reset PLC program

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

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

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

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

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

BIT	SIGNIFICATO	MEANING
Bit 0	Non utilizzato	Not used
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	Mancato avviamento	Failure to start
Bit 8	Motore avviato	Engine started
Bit 9	Minimo livello combustibile	Low fuel level
Bit 10	Test automatico in corso	Automatic test in progress
Bit 11	(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 commando descritto nella tabella 4.

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.

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 a 64), il byte meno significativo viene incrementato ogni volta che un evento viene salvato (valore compreso tra 0 a 64).Una volta memorizzati 64 eventi l'MSB resterà a 64 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'FFL800DP. 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

FFL800DP Funzione = 4
Nr. byte. = 1 (01H)
MSB = 100 (64H)
LSB = 72 (48H)

01	04	02	64	48	93	C6

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

FFL800DP 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

FFL800DP Funzione = 4 (04H)
Indirizzo = 5030H (5030H – 0001H =502FH)
Nr. byte = 86 (56H)
Stringa = 2012/07/18;09:34:52;E1100,CAMBIO MODALITÁ IN: MODALITÁ OFF

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 100), the least significant byte (lsb) is incremented each time an event is saved (value between 0 to 100). Once stored the 100 events the msb will remain at 100 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 FFL800DP 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

FFL800DP Function = 4
Nr. bytes. = 1 (01H)
MSB = 100 (64H)
LSB = 2 (02H)

01	04	02	64	42	13	C1

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

FFL800DP 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

FFL800DP Function = 4 (04H)
Address = 5030H (5030H – 0001H =502FH)
Nr. bytes = 86 (56H)
String = 2012/07/18;09:34:52;E1100,CAMBIO MODALITÁ IN: MODALITÁ OFF

01	04	56	32	30	31	30	2F	30	31	2F	30	31	3B	30	30	3A	31	34	3A
30	31	3B	45	30															

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 unita' di misura, fare riferimento al manuale operativo del FFL800DP.

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 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 (es. M14.0x.06)	3 registri (6 byte)
Testo lunghezza 16 caratteri (es. M14.0x.05)	8 registri (16 byte)
Testo lunghezza 20 caratteri (es. M15.0x.03)	10 registri (20 byte)
Abs(Valore numerico)<32768 (es. M01.05)	1 registri (2 byte)
Abs(Valore numerico)>32768 (es. M12.01)	2 registri (4 byte)
Indirizzo IP (es. M08.0x.06 M08.0x.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*

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)

01	06	4F	FF	00	08	AE	E8

FFL800DP Funzione = 6
Indirizzo = 5000H (5000H – 0001H =4FFFH)
Valore = 8 (08H)

01	06	4F	FF	00	08	AE	E8

Passo 2: Impostazione sottomenu 01.

MASTER Funzione = 6
Indirizzo = 5001H (5001H – 0001H =5000H)
Valore = 1 (01H)

01	06	50	00	00	01	59	0A

FFL800DP Funzione = 6
Indirizzo = 5001H (5001H – 0001H =5000H)
Valore = 1 (01H)

01	06	50	00	00	01	59	0A

Passo 3: Impostazione parametro 01.

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

01	06	50	01	00	01	08	CA

FFL800DP Funzione = 6
Indirizzo = 5002H (5002H – 0001H =5001H)
Valore = 1 (02H)

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 FFL800DP 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 3.(write value 5 by using *function 6* at address *2F03H*)

TYPE OF PARAMETER	NUMBER OF REGISTER
Text length 6 characters (ex. M14.0x.06)	3 registers (6 byte)
Text length 16 characters (ex. M14.0x.05)	8 registers (16 byte)
Text length 20 characters (ex. M15.0x.03)	10 registers (20 byte)
Abs(Numeric value) < 32768 (ex M01.05)	1 registers (2 byte)
Abs(Numeric value) > 32768 (ex M12.01)	2 registers (4 byte)
IP address (ex. M08.0x.06 M08.0x.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*

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)

01	06	4F	FF	00	08	AE	E8

FFL800DP Function = 6
Address = 5000H (5000H – 0001H =4FFFH)
Value = 8 (08H)

01	06	4F	FF	00	08	AE	E8

Step 2: Set submenu 01.

MASTER Function = 6
Address = 5001H (5001H – 0001H =5000H)
Value = 1 (01H)

01	06	50	00	00	01	59	0A

FFL800DP Function = 6
Address = 5001H (5001H – 0001H =5000H)
Value = 1 (01H)

01	06	50	00	00	01	59	0A

Step 3: Set parameter 01.

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

01	06	50	01	00	01	08	CA

FFL800DP Function = 6
Address = 5002H (5002H – 0001H =5001H)
Value = 1 (02H)

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

FFL800DP Funzione = 16 (10H)
Indirizzo = 5004H (5004H – 0001H =5003H)
Valore = 2 (02H)

01	10	50	03	00	01	E0	C9

Passo 5: Salavataggio e riavvio.

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

01	06	2F	02	00	05	E0	DD

FFL800DP 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

FFL800DP Function = 16 (10H)
Address = 5004H (5004H – 0001H =5003H)
Value = 2 (02H)

01	10	50	03	00	01	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

FFL800DP No answer.