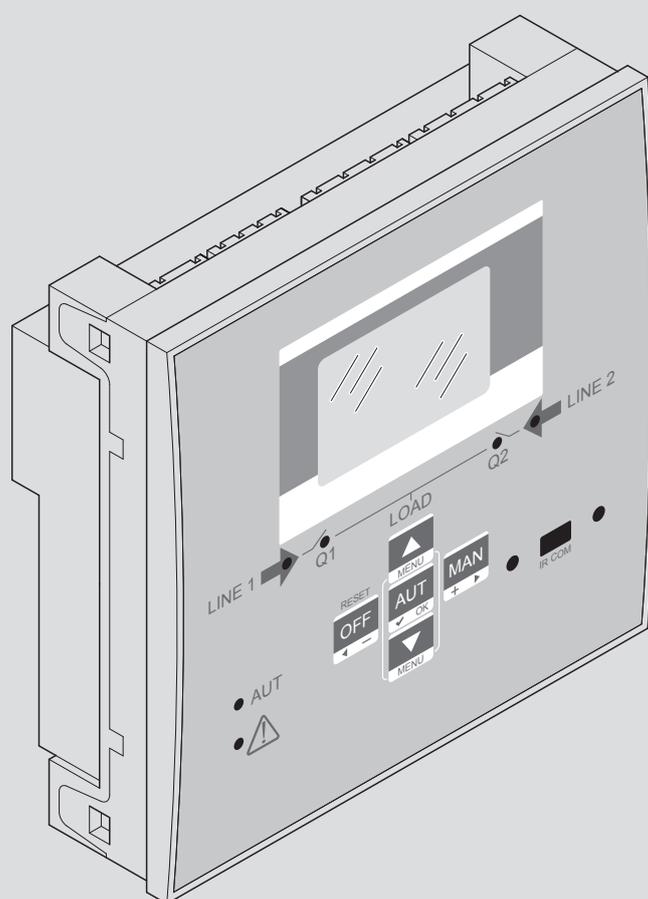


Centralina di commutazione automatica

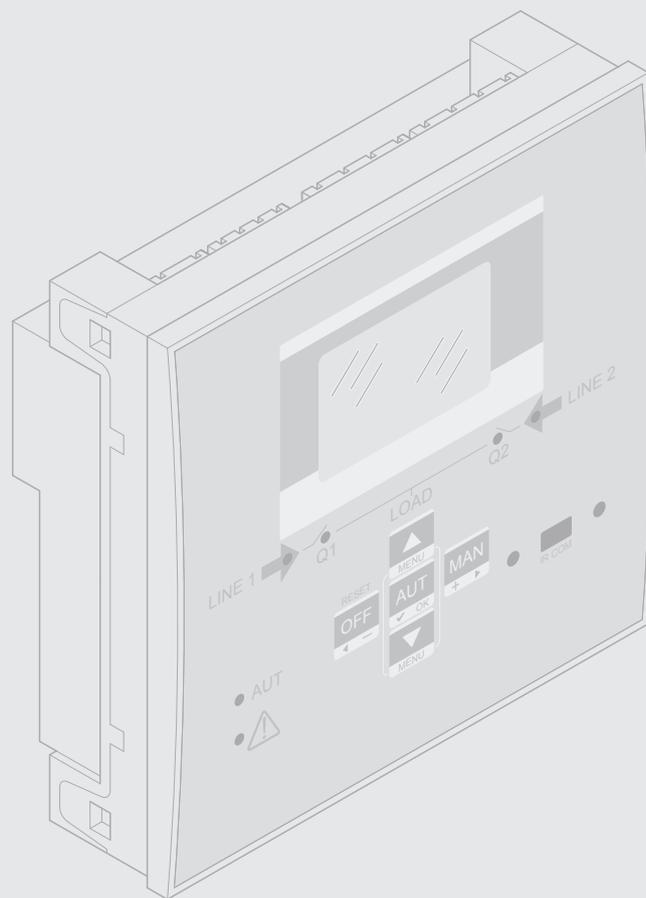
Protocollo di comunicazione Modbus



Centralina di commutazione automatica

IT

ITALIANO

3

Sommario

1. Protocollo Modbus	4
2. Settaggio dei parametri	4
3. Protocollo Modbus RTU	5
4. Funzioni Modbus	6
4.1 Funzione 04: read input register	6
4.2 Funzione 06: preset single register	7
4.3 Funzione 07: read exception status	8
4.4 Funzione 16: preset multiple register	8
4.5 Funzione 17: report slave ID	9
5. Errori	9
6. Protocollo Modbus ASCII	10
7. Tabelle	11
7.1 Misure fornite dal protocollo di comunicazione	11
7.2 BIT di stato	13
7.3 Comandi	16
7.4 Stato globale dispositivo	18
7.5 Orologio/datario	18
8. Lettura lista eventi	19
9. Impostazione parametri	20
A. Calcolo CRC (Checksum per RTU)	22
B. Calcolo LRC (Checksum per ASCII)	23

1. Protocollo Modbus

La centralina di commutazione automatica Bticino M7000CBNCU03 supporta il protocollo di comunicazione Modbus (RTU e ASCII) su interface di comunicazione:

- M7000CBNUSB: interfaccia frontale optoisolata IR-USB
- M7000CBNWIFI: interfaccia frontale optoisolata IR-WiFi
- M7000CBNRS485: modulo di espansione serial RS485

Utilizzando la comunicazione Modbus è possibile leggere e scrivere parametri del dispositivo e controllarlo attraverso sistemi di supervisione o sistemi di controllo master (come PLC o SCADA).

2. Settaggio dei parametri

Per configurare il protocollo Modbus, entrare nel Menu "Impostazioni" e selezionare il menu M08, impostando i parametri per il canale desiderato (COM1 o COM2 a seconda di come posizionato il modulo di espansione M7000CBNRS485).

Menu parametri

M08 – COMUNICAZIONE (COMn, n=1...2)		Default	Range
P08.n.01	Indirizzo nodo seriale	05	01-247 (248 ... 255 per uso interno)
P08.n.02	Velocità	19200	1200 2400 4800 9600 19200 38400 57600 115200
P08.n.03	Formato dati	8 bit, even	8 bit –no par. 8 bit, dispari 8 bit, pari 7 bit, dispari 7 bit, pari
P08.n.04	Stop bit	1	1-2
P08.n.05	Protocollo	Modbus RTU	Modbus RTU Modbus ASCII

Nota: in caso di utilizzo delle interface frontali USB o WiFi non è necessario effettuare un settaggio di parametri, in quanto già preconfigurate e pronte all'uso.

- **P08.n.01** – Indirizzo seriale (nodo) della rete Modbus a cui fa riferimento il dispositivo.
- **P08.n.02** – Velocità della porta di comunicazione.
- **P08.n.03** – Formato dati. Settaggio a 7 bit disponibile solo per protocollo Modbus ASCII.
- **P08.n.04** – Numero di Stop bit.
- **P08.n.05** – Tipo di protocollo di comunicazione.
- **P08.n.06...P08.n.13** – Non disponibili con l'attuale offerta.

3. Protocollo Modbus RTU

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

T1T2T3	Indirizzo (8 bit)	Funzione (8 bit)	Dati (N x 8 bit)	CRC (16 bit)	T1T2T3
--------	----------------------	---------------------	---------------------	-----------------	--------

- Il campo Indirizzo contiene l'indirizzo del dispositivo 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, quindi 160 byte).
- Il campo CRC consente sia al master che allo slave di verificare se ci sono errori di trasmissione. Questo consente, in caso di disturbo sulla linea di trasmissione, di ignorare il messaggio inviato per evitare problemi sia dal lato master che slave.
- La sequenza T1 T2 T3 corrisponde al tempo durante il quale non devono essere scambiati dati sul bus di comunicazione, per consentire agli strumenti collegati di riconoscere la fine di un messaggio e l'inizio del successivo. Questo tempo deve essere pari a 3.5 caratteri.

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

4. Funzioni Modbus

Le funzioni disponibili sono:

03 = Read Multiple Holding Registers	Consente la lettura dei valori disponibili
04 = Read input register	Consente la lettura dei registri di input
06 = Preset single register	Permette la scrittura di parametri
07 = Read exception	Permette di leggere lo stato del dispositivo
10 = Preset multiple register	Permette la scrittura di più parametri
17 = Report slave ID	Permette di leggere informazioni relative all'apparecchio

Esempio:

Per leggere dal dispositivo con indirizzo 01 il numero di allarmi commutazione dell'interruttore 1, che si trova alla locazione 58 (3A_{hex}), il messaggio da spedire è il seguente:

01	04	00	39	00	02	A1	C6
----	----	----	----	----	----	----	----

Dove:

01= indirizzo slave

04 = funzione di lettura locazione

00 39 = 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 3A

A1 C6 = checksum CRC

La risposta è la seguente:

01	04	04	00	00	00	07	BA	46
----	----	----	----	----	----	----	----	----

Dove:

01= indirizzo del dispositivo (Slave 01)

04 = funzione richiesta dal Master

04 = numero di byte inviati

00 00 00 07 = valore esadecimale del numero di allarmi di commutazione dell'interruttore 1 (7 allarmi, in questo esempio)

BA 46 = checksum CRC

4.1 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 7.1. Come da standard Modbus, l'indirizzo specificato nel messaggio va diminuito di 1 rispetto a quello effettivo riportato nella tabella.

Se l'indirizzo richiesto non è compreso nella tabella o il numero di registri richiesti è maggiore del numero consentito, il dispositivo ritorna un messaggio di errore (vedi tabella errori in cap. 5).

Esempio:

Richiesta master:

Indirizzo slave	08 _{hex}
Funzione	04 _{hex}
MSB Indirizzo registro	00 _{hex}
LSB Indirizzo registro	0F _{hex}
MSB Numero registri	00 _{hex}
LSB Numero registri	08 _{hex}
LSB CRC	C1 _{hex}
MSB CRC	56 _{hex}

Nell'esempio vengono richiesti, allo slave numero 8, 8 registri consecutivi a partire dall'indirizzo 10_{hex} . Quindi vengono letti i registri dal 10_{hex} al 17_{hex} .
Il comando termina sempre con il valore di checksum CRC.

Risposta Slave:

Indirizzo slave	08_{hex}
Funzione	04_{hex}
Numero di byte	10_{hex}
MSB Dato 10_{hex}	00_{hex}
LSB Dato 10_{hex}	00_{hex}
-----	----
MSB Dato 17_{hex}	00_{hex}
LSB Dato 17_{hex}	00_{hex}
LSB CRC	$8A_{hex}$
MSB CRC	$B1_{hex}$

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.

4.2 Funzione 06: preset single register

Questa funzione permette di scrivere nei registri. Essa può essere utilizzata solo con i registri d'indirizzo superiore a 1000_{hex} . E' possibile, ad esempio, impostare i parametri di setup. Qualora il valore impostato non rientri nel valore minimo e massimo della tabella, il dispositivo risponderà con un messaggio di errore. Se venisse 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 7.3.

Esempio:

Richiesta Master:

Indirizzo slave	08_{hex}
Funzione	06_{hex}
MSB Indirizzo registro	$2F_{hex}$
LSB Indirizzo registro	$0F_{hex}$
MSB Dato	00_{hex}
LSB Dato	$0A_{hex}$
LSB CRC	31_{hex}
MSB CRC	83_{hex}

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.

4. Funzioni Modbus

4.3 Funzione 07: read exception status

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

Esempio:

Richiesta Master:

Indirizzo slave	08 _{hex}
Funzione	07 _{hex}
LSB CRC	47 _{hex}
MSB CRC	B2 _{hex}

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

BIT	SIGNIFICATO
0	Modo operativo OFF / Reset
1	Modo operativo MAN
2	Modo operativo AUT
3	Modo operativo TEST
4	In errore
5	Alimentazione AC presente
6	Alimentazione DC presente
7	Allarme globale attivato

4.4 Funzione 16: preset multiple register

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

Esempio:

Richiesta Master:

Indirizzo slave	08 _{hex}
Funzione	10 _{hex}
MSB Indirizzo registro	20 _{hex}
LSB Indirizzo registro	01 _{hex}
MSB Numero registri	00 _{hex}
LSB Numero registri	02 _{hex}
Numero di byte (<i>è il doppio di quelli sopra</i>)	04 _{hex}
MSB Dato	00 _{hex}
LSB Dato	00 _{hex}
MSB Dato	00 _{hex}
LSB Dato	00 _{hex}
LSB CRC	85 _{hex}
MSB CRC	3E _{hex}

Risposta Slave:

Indirizzo slave	08 _{hex}
Funzione	10 _{hex}
MSB Indirizzo registro	20 _{hex}
LSB Indirizzo registro	01 _{hex}
MSB Numero byte	00 _{hex}
LSB Numero byte	02 _{hex}
LSB CRC	1B _{hex}
MSB CRC	51 _{hex}

4.5 Funzione 17: report slave ID

Questa funzione permette di identificare il tipo di dispositivo.

Esempio:

Richiesta Master:

Indirizzo slave	08 _{hex}
Funzione	11 _{hex}
LSB CRC	C6 _{hex}
MSB CRC	7C _{hex}

Risposta Slave:

Indirizzo slave	08 _{hex}
Funzione	11 _{hex}
Contatore byte	08 _{hex}
Dato 01 (Tipo) ❶	76 _{hex}
Data 02 (Revisione Fw)	01 _{hex}
Data 03 (Revisione Hardware)	00 _{hex}
Data 04 (Revisione Parametri)	01 _{hex}
Data 05 (tipo dispositivo) ❷	04 _{hex}
Data 06 (riservato)	00 _{hex}
Data 07 (riservato)	00 _{hex}
Data 08 (riservato)	00 _{hex}
LSB CRC	B0 _{hex}
MSB CRC	2A _{hex}

❶ 118 - 76_{hex} = M7000CBNCU03

❷ 3 - 03_{hex} = Serie Bticino

5. 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:

CODICE	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

6. Protocollo Modbus ASCII

Il protocollo Modbus ASCII viene utilizzato normalmente nelle applicazioni che richiedono di comunicare via modem. Le funzioni e gli indirizzi disponibili sono gli stessi della versione RTU, ma i caratteri trasmessi sono in ASCII e la terminazione del messaggio non è effettuata a tempo ma con dei caratteri di ritorno a capo. Se si seleziona il parametro P7.05 o come protocollo Modbus ASCII, la struttura del messaggio di comunicazione sulla relativa porta di comunicazione è così costituita:

:	Indirizzo (2 chars)	Funzione (2 chars)	Dati (N chars)	LRC (2 chars)	CRLF
---	------------------------	-----------------------	-------------------	------------------	------

- Il campo Indirizzo contiene l'indirizzo del dispositivo 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 80 registri consecutivi.
- Il campo LRC consente sia al master che allo slave di verificare se ci sono errori di trasmissione. Questo permette, in caso di disturbo sulla linea di trasmissione, di ignorare il messaggio inviato per evitare problemi sia dal lato master che slave.
- Il messaggio termina sempre con i caratteri di controllo CRLF (0D 0A).

Esempio:

Per leggere dal dispositivo con indirizzo 8 il valore della corrente di fase L3 equivalente che si trova alla locazione 12 ($0C_{hex}$), il messaggio da spedire è il seguente:

:	08	04	00	0B	00	02	E7	CRLF
---	----	----	----	----	----	----	----	------

Dove:

- : = ASCII $3A_{hex}$ = Delimitatore inizio messaggio
- 08 = indirizzo slave.
- 04 = funzione di lettura locazione.
- 00 0B = indirizzo della locazione diminuito di un'unità, contenente il valore della corrente di fase L3
- 00 02 = numero di registri da leggere a partire dall'indirizzo 04.
- E7 = checksum LRC.
- CRLF = ASCII $0D_{hex}$ $0A_{hex}$ = delimitatore fine messaggio

La risposta è la seguente:

:	08	04	04	00	00	A8	AE	9B	CRLF
---	----	----	----	----	----	----	----	----	------

Dove:

- : = ASCII $3A_{hex}$ = Delimitatore inizio messaggio
- 08 = indirizzo Slave
- 04 = funzione richiesta dal Master.
- 04 = numero di byte inviati dallo slave.
- 00 00 A8 AE = valore esadecimale della corrente di fase L3 = 4.3182 A
- 9B = checksum LRC.
- CRLF = ASCII $0D_{hex}$ $0A_{hex}$ = delimitatore fine messaggio

7. Tabelle

7.1 Misure fornite dal protocollo di comunicazione

Da usare con le funzioni 03 e 04.

INDIRIZZO	WORDS	MISURE	UNITÀ	FORMATO
02 _{hex}	2	Tensione Linea 1 L1-N	V	Unsigned long
04 _{hex}	2	Tensione Linea 1 L2-N	V	Unsigned long
06 _{hex}	2	Tensione Linea 1 L3-N	V	Unsigned long
08 _{hex}	2	Tensione Linea 1 L1-L2	V	Unsigned long
0A _{hex}	2	Tensione Linea 1 L2-L3	V	Unsigned long
0C _{hex}	2	Tensione Linea 1 L3-L1	V	Unsigned long
0E _{hex}	2	Tensione Linea 2 L1-N	V	Unsigned long
10 _{hex}	2	Tensione Linea 2 L2-N	V	Unsigned long
12 _{hex}	2	Tensione Linea 2 L3-N	V	Unsigned long
14 _{hex}	2	Tensione Linea 2 L1-L2	V	Unsigned long
16 _{hex}	2	Tensione Linea 2 L2-L3	V	Unsigned long
18 _{hex}	2	Tensione Linea 2 L3-L1	V	Unsigned long
1A _{hex}	2	Frequenza linea 1	Hz/10	Unsigned long
1C _{hex}	2	Frequenza linea 2	Hz/10	Unsigned long
1E _{hex}	2	Tensione batteria (alimentazione DC)	VDC / 10	Unsigned long
20 _{hex}	2	Tempo di lavoro totale	s	Unsigned long
22 _{hex}	2	Tempo totale linea 1 ok	s	Unsigned long
24 _{hex}	2	Tempo totale linea 2 ok	s	Unsigned long
26 _{hex}	2	Tempo totale linea 1 non ok	s	Unsigned long
28 _{hex}	2	Tempo totale linea 2 non ok	s	Unsigned long
2A _{hex}	2	Tempo totale interruttore linea 1 chiuso	s	Unsigned long
2C _{hex}	2	Tempo totale interruttore linea 2 chiuso	s	Unsigned long
2E _{hex}	2	Tempo totale interruttori aperti	s	Unsigned long
30 _{hex}	2	(non usato)	--	Unsigned long
32 _{hex}	2	Numero commutazioni interruttore 1 in AUT	nr	Unsigned long
34 _{hex}	2	Numero commutazioni interruttore 2 in AUT	nr	Unsigned long
36 _{hex}	2	Numero commutazioni interruttore 1 in MAN	nr	Unsigned long
38 _{hex}	2	Numero commutazioni interruttore 2 in MAN	nr	Unsigned long
3A _{hex}	2	Numero allarmi commutazione interruttore 1	nr	Unsigned long
3C _{hex}	2	Numero allarmi commutazione interruttore 2	nr	Unsigned long
3E _{hex}	2	(non usato)	--	Unsigned long
40 _{hex}	2	Allarmi 🚫	bits	Unsigned long
50 _{hex}	2	Tensione batteria minima	V	Unsigned long
52 _{hex}	2	Tensione batteria massima	V	Unsigned long

(continua)

7. Tabelle

INDIRIZZO	WORDS	MISURE	UNITÀ	FORMATO
54 _{hex}	2	Manutenzione ore linea 1	nr	Unsigned long
56 _{hex}	2	Manutenzione ore linea 2	nr	Unsigned long
58 _{hex}	2	Manovre mancanti manutenzione ore linea 1	nr	Signed long
5A _{hex}	2	Manovre mancanti manutenzione ore linea 2	nr	Signed long
21C0 _{hex}	1	OR di tutti i limiti	bits	Unsigned int
21C1 _{hex}	1	LIM 1	bits	Unsigned int
21C2 _{hex}	1	LIM 2	bits	Unsigned int
21C3 _{hex}	1	LIM 3	bits	Unsigned int
21C4 _{hex}	1	LIM 4	bits	Unsigned int
1D00 _{hex}	2	Contatore CNT 1	UM1	long
1D02 _{hex}	2	Contatore CNT 2	UM2	long
1D04 _{hex}	2	Contatore CNT 3	UM3	long
1D06 _{hex}	2	Contatore CNT 4	UM4	long

❶ Leggendo le word all'indirizzo 40_{hex} vengono restituiti 32 bit con significato come da tabella:

BIT	CODICE	ALLARME
0	A01	Tensione batteria troppo bassa
1	A02	Tensione batteria troppo alta
2	A03	Timeout interruttore Linea 1
3	A04	Timeout interruttore Linea 2
4	A05	Errata sequenza fase Linea 1
5	A06	Errata sequenza fase Linea 2
6	A07	Timeout carico non alimentato
7	A08	Avaria caricabatteria esterno
8	A09	Emergenza
9	A10	Intervento protezione Interruttore Linea 1 (trip)
10	A11	Intervento protezione Interruttore Linea 2 (trip)
11	A12	Generatore linea 1 non disponibile

BIT	CODICE	ALLARME
12	A13	Generatore linea 2 non disponibile
13	A14	Manutenzione ore 1
14	A15	Manutenzione ore 2
15	A16	Manutenzione manovre 1
16	A17	Manutenzione manovre 2
17	A18	Allarme tensione ausiliaria
18	UA1	Allarme utente
19	UA2	Allarme utente
20	UA3	Allarme utente
21	UA4	Allarme utente
22 - 31	---	Non usati

7.2 BIT di stato

Da usare con le funzioni 03 e 04.

INDIRIZZO	WORDS	FUNZIONE	FORMATO
2070 _{hex}	1	Stato tastiera frontale ❶	Unsigned integer
2100 _{hex}	1	Stato ingressi digitali (per pin) ❷	Unsigned integer
2140 _{hex}	1	Stato uscite digitali (per pin) ❸	Unsigned integer
-	-	-	-
2074 _{hex}	1	Stato tensione linea 1 ❹	Unsigned integer
2075 _{hex}	1	Stato interruttore linea 1 ❺	Unsigned integer
2176 _{hex}	1	Stato tensione linea 2 ❹	Unsigned integer
2177 _{hex}	1	Stato interruttore linea 2 ❺	Unsigned integer
2078 _{hex}	2	Stato funzioni di ingresso ❻	Unsigned integer
207A _{hex}	1	Stato funzioni di uscita ❼	Unsigned integer
207B _{hex}	1	Stato messaggi su display ❸	Unsigned integer
207C _{hex}	1	Stato generale controller ❾	Unsigned integer
207E _{hex}	1	Stato LED frontali	Unsigned integer
207F _{hex}	1	Stato LED frontali	Unsigned integer

❶ Leggendo le word all'indirizzo 2070_{hex} vengono restituiti 16 bit con significato come da tabella:

BIT	TASTO
0	FRECCIA SU
1	OFF/RESET
2	MAN
3	FRECCIA GIU'
4	AUT/ENTER
5...15	Non usati

7. Tabelle

② Leggendo le word all'indirizzo 2100_{hex} vengono restituiti 16 bit con significato come da tabella:

BIT	INPUT
0	Input 1
1	Input 2
2	Input 3
3	Input 4
4	Input 5
5	Input 6
6	Input 7
7	Input 8

BIT	INPUT
8	Input 9
9	Input 10
10	Input 11
11	Input 12
12	Input 13
13	Input 14
14-15	Not used

③ Leggendo le word all'indirizzo 2072_{hex} vengono restituiti 16 bit con significato come da tabella:

BIT	OUTPUT
0	Output 1
1	Output 2
2	Output 3
3	Output 4
4	Output 5
5	Output 6
6	Output 7
7	Output 8

BIT	OUTPUT
8	Output 9
9	Output 10
10	Output 11
11	Output 12
12	Output 13
13	Output 14
14	Output 15
15	Not used

④ Leggendo le word all'indirizzo 2074_{hex} (Linea 1) o 2176_{hex} (linea 2) vengono restituiti 16 bit con significato come da tabella:

BIT	STATO LINEA
0	Linea nei limiti
1	Linea nei limit + ritardo
2	Tensione nei limiti
3	Tensione ok
4	Frequenza nei limiti
5	Frequenza ok
6	Tensione < minima
7	Tensione > massima

BIT	STATO LINEA
8	Tensioni fuori soglia asimmetria
9	Tensione < soglia mancanza fase
10	Frequenza < minima
11	Frequenza > massima
12	Sequenza fasi errata
13	Tutti i parametri linea ok
14-15	Non usato

⑤ Leggendo le word all'indirizzo 2075_{hex} (linea 1) o 2177_{hex} (linea 2) vengono restituiti 16 bit con significato come da tabella:

BIT	STATO INTERRUTTORE
0	Interruttore chiuso
1	Allarme Trip
2	Non usato
3	Stato comandato (1= chiuso)
4	Uscita comando chiusura
5	Uscita comando apertura
6...15	Non usati

⑥ Leggendo le word all'indirizzo 2178_{hex} vengono restituiti 16 bit con significato come da tabella:

BIT	STATO FUNZIONI INGRESSO
0	Interruttore Linea1 chiuso
1	Interruttore Linea1 Trip
2	Non usato
3	Interruttore Linea2 chiuso
4	Interruttore Linea2 Trip
5	Non usato
6	Forzata su Linea secondaria
7	Inibizione ritorno su Linea principale

BIT	STATO FUNZIONI INGRESSO
8	Pulsante emergenza
9	Start generatore
10	Generatore 1 pronto
11	Generatore 2 pronto
12	Blocco tastiera
13	Blocco programmazione
14	Non usato
15	Allarmi inibiti

⑦ Leggendo le word all'indirizzo 207A_{hex} vengono restituiti 16 bit con significato come da tabella:

BIT	STATO FUNZIONI USCITA
0	Apertura linea 1
1	Apertura linea 2
2	Chiusura linea 1
3	Chiusura Linea 2
4	Allarme globale
5	Start generatore 1
6	Start generatore 2
7	ATS pronto

BIT	STATO FUNZIONI USCITA
8	Load shed
9	Non usato
10	Non usato
11	Apri tutto
12	Bobina minima 1
13	Bobina minima 2
14	Linea 1 ok
15	Linea 2 ok

7. Tabelle

⑧ Leggendo le word all'indirizzo 207B_{hex} vengono restituiti 16 bit con significato come da tabella:

BIT	STATO MESSAGGI DISPLAY
0	Start generatore 1
1	Start generatore 2
2	Raffreddamento generatore 1
3	Raffreddamento generatore 2
4	Trasferimento carico 2 → 1
5	Trasferimento carico 1 → 2

⑨ Leggendo le word all'indirizzo 207C_{hex} vengono restituiti 16 bit con significato come da tabella:

BIT	STATO FUNZIONI USCITA
0	Modo operativo OFF / Reset
1	Modo operativo MAN
2	Modo operativo AUT
3	Modo operativo TEST
4	In Errore
5	Alimentazione AC presente
6	Alimentazione DC presente
7	Allarme globale attivato
8...15	Non usati

7.3 Comandi

Da usare con la funzione 06.

INDIRIZZO	WORDS	STATO
4F00 _{hex}	1	Imposta variabile remora REM1 ①
4F01 _{hex}	1	Imposta variabile remora REM2
.....		
4F07 _{hex}	1	Imposta variabile remora REM8
2F00 _{hex}		Cambio modalità operativa ②
2F0A _{hex}	1	Simulazione pressione tasti pannello frontale ③
2F03 _{hex}	1	Valore 01 _{hex} : Salvataggio memoria
		Valore 04 _{hex} : reboot
2F07 _{hex}	1	Valore 00 _{hex} : Reset apparecchio
		Valore 01 _{hex} : Reset apparecchio con salvataggio in memoria
2FF0 _{hex}	1	Esecuzione comando menu comandi ④
28FA _{hex}	1	Valore 01 _{hex} : Salvataggio impostazione orologio datario

- ① Scrivendo il valore AA_{hex} all'indirizzo indicato viene impostata la variabile remota a 1, scrivendo BB_{hex} viene impostata a 0
- ② La seguente tabella indica i valori da scrivere all'indirizzo 2F00_{hex} per ottenere le corrispondenti funzioni:

VALORE	FUNZIONE
0	Passaggio a modalità OFF
1	Passaggio a modalità MAN
2	Passaggio a modalità AUT

- ③ La seguente tabella indica i valori da scrivere all'indirizzo 2F0A_{hex} per ottenere le corrispondenti funzioni:

BIT	SIGNIFICATO
0	Tasto Su
1	Modalità MAN
2	Tasto destra
3	START
4	Modalità TEST
5	Modalità OFF
6	Modalità AUT
7	STOP

- ④ Scrivendo il valore da 0 a 15 all'indirizzo indicato viene eseguita la corrispondente funzione:

BIT	SIGNIFICATO
0	Reset manutenzione ore 1
1	Reset manutenzione ore 2
2	Reset manutenzione manovre 1
3	Reset manutenzione manovre 2
4	Reset contatori generici CNTx
5	Reset stato limiti LIMx
6	Reset conta ore linea 1/ Linea 2
7	Reset conta ore Q1/Q2

BIT	SIGNIFICATO
8	Reset manovre interruttori
9	Reset lista eventi
10	Ripristino parametri a default
11	Salva parametri nella memoria backup
12	Ricarica parametri dalla memoria backup
13	Forzatura I/O
14	Reset allarmi A03 – A04
15	Simula mancanza linea prioritaria

7. Tabelle

7.4 Stato globale dispositivo

Da usare con le funzioni 03 e 04.

INDIRIZZO	WORDS	STATO	FORMATO
2210 _{hex}	2	Stato globale dispositivo (bit 0-bit31) ❶	Unsigned integer

❶ Leggendo 2 word agli indirizzi 2210_{hex} vengono restituiti 32 Bit con significato come da tabella:

BIT	SIGNIFICATO
Bit 0	Dispositivo in OFF
Bit 1	Dispositivo in MAN
Bit 2	Dispositivo in AUT
Bit 3	Dispositivo in TEST
Bit 4	Tensione di Linea1 OK
Bit 5	Tensione di Linea2 OK
Bit 6	Led Linea 1 acceso
Bit 7	Led Linea 2 acceso
Bit 8	Led Linea 1 sul carico
Bit 9	Led Linea 1 sul carico
Bit 10	Teleruttore rete chiuso
Bit 11	Teleruttore generatore chiuso
Bit 12	Allarme globale
Bit 13	Alimentazione AC
Bit 14	Start Generatore 1
Bit 15	Start Generatore 2

BIT	SIGNIFICATO
Bit 16	Linea 1 max Volt
Bit 17	Linea 1 min Volt
Bit 18	Linea 1 max Hz
Bit 19	Linea 1 min Hz
Bit 20	Linea 1 perdita di fase
Bit 21	Linea 1 asym
Bit 22	Linea 2 max Volt
Bit 23	Linea 2 min Volt
Bit 24	Linea 2 max Hz
Bit 25	Linea 2 min Hz
Bit 26	Linea 2 perdita di fase
Bit 27	Linea 2 asym
Bit 28	(non usato)
...
Bit 31	(non usato)

7.5 Orologio/datario

Da usare con le funzioni 04 e 06.

Per rendere effettivi i cambiamenti, memorizzare le impostazioni utilizzando l'apposito comando descritto nella tabella 7.3.

INDIRIZZO	WORDS	FUNZIONE	RANGE
28F0 _{hex}	1	Anno	2000..2099
28F1 _{hex}	1	Mese	1-12
28F2 _{hex}	1	Giorno	1-31
28F3 _{hex}	1	Ora	0-23
28F4 _{hex}	1	Minuti	0-59
28F5 _{hex}	1	Secondi	0-59

8. Lettura lista eventi

Per leggere gli eventi bisogna svolgere la seguente procedura:

1. Eseguire la lettura di 1 registro con la **funzione 04** all'indirizzo 5030_{hex} , il byte più significativo (MSB) indica quanti eventi sono memorizzati (valore compreso tra 0 a 100), il byte meno significativo viene incrementato ogni volta che un evento viene salvato (valore compreso tra 0 a 100). Una volta memorizzati 100 eventi l'MSB resterà a 100 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 5030_{hex} specificando quale evento leggere.
3. Eseguire una lettura di 43 registri (con un'unica **funzione 04**) all'indirizzo 5032_{hex} .
4. Il valore tornato è una stringa di 86 caratteri ASCII, che riportano la stessa descrizione dell'evento visibile sul display del dispositivo. L'indice dell'evento che si vuole leggere viene incrementato in automatico dopo la lettura del registro 5032_{hex} 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:

Step 1: Lettura eventi memorizzati.

MASTER Funzione = 4 (04_{hex})
 Indirizzo = 5030_{hex} $(5030_{hex} - 0001_{hex} = 502F_{hex})$
 Nr. registri = 1 (01_{hex})

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

ATS Funzione = 4 (04_{hex})
 Nr. byte = 1 (01_{hex})
 MSB = 100 (64_{hex})
 LSB = 2 (02_{hex})

01	04	02	64	42	13	C1
----	----	----	----	----	----	----

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

MASTER Funzione = 6 (06_{hex})
 Indirizzo = 5030_{hex} $(5030_{hex} - 0001_{hex} = 502F_{hex})$
 Valore = 1 (01_{hex})

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

ATS Funzione = 6 (06_{hex})
 Indirizzo = 5030_{hex} $(5030_{hex} - 0001_{hex} = 502F_{hex})$
 Valore = 1 (01_{hex})

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

Step 3: Lettura evento.

MASTER Funzione = 4 (04_{hex})
 Indirizzo = 5032_{hex} $(5032_{hex} - 0001_{hex} = 5031_{hex})$
 Nr. registri = 43 $(2B_{hex})$

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

ATS Funzione = 4 (04_{hex})
 Indirizzo = 5030_{hex} $(5030_{hex} - 0001_{hex} = 502F_{hex})$
 Nr. byte = 86 (56_{hex})

Stringa = 2012/07/18;09:34:52;E1 100, CAMBIO MODALITÀ IN: MODALITÀ OFF

9. Impostazione parametri

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															

Tramite il protocollo Modbus è possibile accedere ai parametri dei menu.

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

PROCEDURA PER LA LETTURA DEI PARAMETRI

1. Scrivere il valore del menu che si vuole leggere tramite la **funzione 06** all'indirizzo 5000_{hex} ①.
2. Scrivere il valore del sottomenu (se esiste) che si vuole leggere tramite la **funzione 06** all'indirizzo 5001_{hex} ①.
3. Scrivere il valore del parametro che si vuole leggere tramite la **funzione 06** all'indirizzo 5002_{hex} ①.
4. Eseguire la **funzione 4** all'indirizzo 5004_{hex} di un numero di registri appropriato alla lunghezza del parametro.
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 06** all'indirizzo 5000_{hex} ①.
2. Scrivere il valore del sottomenu (se esiste) che si vuole modificare tramite la **funzione 06** all'indirizzo 5001_{hex} ①.
3. Scrivere il valore parametro che si vuole modificare tramite la **funzione 06** all'indirizzo 5002_{hex} ①.
4. Eseguire la **funzione 16** all'indirizzo 5004_{hex} 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 memoria, utilizzando l'apposito comando descritto nella tabella 7 (scrivere il valore 04 con la **funzione 06** all'indirizzo $2F03_{hex}$).

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)

- ① E' possibile leggere il valore del menu, sottomenu e parametro memorizzati agli indirizzi 5000_{hex} , 5001_{hex} e 5002_{hex} utilizzando la **funzione 04**.

Esempio:

Impostare a 8 il valore del parametro M08.01.01

Step 1: Set menu 08.

MASTER Funzione = 6 (06_{hex})
 Indirizzo = 5000_{hex} ($5000_{hex} - 0001_{hex} = 4FFF_{hex}$)
 Valore = 8 (08_{hex})

01	06	4F	FF	00	08	AE	E8
----	----	----	----	----	----	----	----

ATS Funzione = 6 (06_{hex})
 Indirizzo = 5000_{hex} ($5000_{hex} - 0001_{hex} = 4FFF_{hex}$)
 Valore = 8 (08_{hex})

01	06	4F	FF	00	08	AE	E8
----	----	----	----	----	----	----	----

Step 2: Set sottomenu 01.

MASTER Funzione = 6 (06_{hex})
 Indirizzo = 5001_{hex} (5001_{hex} - 0001_{hex} = 5000_{hex})
 Valore = 1 (01_{hex})

01	06	50	00	00	01	59	0A
----	----	----	----	----	----	----	----

ATS Funzione = 6 (06_{hex})
 Indirizzo = 5001_{hex} (5001_{hex} - 0001_{hex} = 5000_{hex})
 Valore = 1 (01_{hex})

01	06	50	00	00	01	59	0A
----	----	----	----	----	----	----	----

Step 3: Set parametro 01.

MASTER Funzione = 6 (06_{hex})
 Indirizzo = 5002_{hex} (5002_{hex} - 0001_{hex} = 5001_{hex})
 Valore = 1 (01_{hex})

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

ATS Funzione = 6 (06_{hex})
 Indirizzo = 5002_{hex} (5002_{hex} - 0001_{hex} = 5001_{hex})
 Valore = 1 (01_{hex})

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

Step 3: Set value 8.

MASTER Funzione = 16 (10_{hex})
 Indirizzo = 5004_{hex} (5004_{hex} - 0001_{hex} = 5003_{hex})
 Nr. Register = 1 (01_{hex})
 Nr. bytes = 2 (02_{hex})
 Valore = 8 (0008_{hex})

01	10	50	03	00	02	04	00	00	00	08	4E	7F
----	----	----	----	----	----	----	----	----	----	----	----	----

ATS Funzione = 16 (10_{hex})
 Indirizzo = 5004_{hex} (5004_{hex} - 0001_{hex} = 5003_{hex})
 Valore = 2 (02_{hex})

01	10	50	03	00	02	A0	C8
----	----	----	----	----	----	----	----

Step 6: Salva e riavvia.

MASTER Funzione = 6 (06_{hex})
 Indirizzo = 2F03_{hex} (2F03_{hex} - 0001_{hex} = 2F02_{hex})
 Valore = 4 (04_{hex})

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

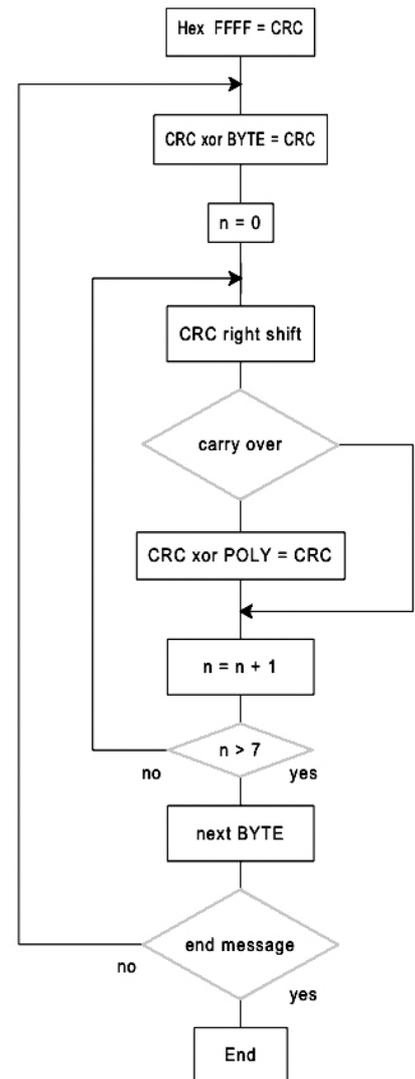
ATS Nessuna risposta.

A. Calcolo CRC (Checksum per RTU)

Algoritmo calcolo CRC

Esempio: Frame = 0207_{hex}

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		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		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		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
			12_{hex}	41_{hex}



Nota: Il byte 41_{hex} viene spedito per primo (anche se è il LSB), poi viene trasmesso

B. Calcolo LRC (Checksum per ASCII)

Esempio:

Indirizzo	01	00000001
Funzione	04	00000100
Start address high	00	00000000
Start address low	00	00000000
Numero registri	08	00001000
	Somma	00001101
	Complemento a 1	11110010
	+ 1	00000001
	Complemento a 2	11110101
Risultato LRC		F5_{hex}

BTicino SpA

Viale Borri, 231
21100 Varese - Italy
www.bticino.com

Servizio clienti



Timbro installatore