

ATTENZIONE

Le attività di installazione, regolazione, ispezione e manutenzione devono essere svolte solo da personale qualificato incaricato. Ogni operazione deve essere effettuata in conformità a leggi e regolamenti in vigore concernenti la sicurezza del personale e l'uso di adeguati dispositivi di protezione.

INTRODUZIONE

I regolatori di rifasamento 5LGA e 8LGA supportano i protocolli di comunicazione Modbus® RTU, Modbus® ASCII e Modbus® TCP (solo 8LGA) sull'interfaccia ottica e sui moduli di espansione:

- USB
- RS232
- RS485
- Ethernet (solo 8LGA)

Grazie a questa funzione è possibile leggere lo stato degli apparecchi e controllare gli stessi tramite software di supervisione standard forniti da terze parti (SCADA) oppure tramite apparecchiature dotate di interfaccia Modbus® quali PLC e terminali intelligenti.

IMPOSTAZIONE DEI PARAMETRI

Per configurare il protocollo Modbus®, accedere al SETUP MENU e selezionare il menu ADV. E' possibile configurare una sola porta di comunicazione.

MENU AVANZATO — COMUNICAZIONE SERIALE

PAR	FUNZIONE	RANGE	DEFAULT
P49	Indirizzo nodo	1 ..255	1
P50	Velocità seriale	1200, 2400, 4800, 9600, 19200, 38400	9600
P51	Formato dati	8 bit Nessuna, 8 bit Dispari, 8 bit Pari, 7 bit Dispari, 7 bit Pari	8 bit Nessuna
P52	Stop bit	1, 2	1
P53	Protocollo	Modbus RTU, Modbus ASCII, Modbus TCP (solo 8LGA)	Modbus RTU

Per il modulo di espansione Ethernet (solo 8LGA) esistono altri parametri. Accedere al SETUP MENU e selezionare il menu FUN.

PAR	FUNZIONE	RANGE	DEFAULT
F.01	Indirizzo IP	000.000.000.000 - 255.255255.255	000.000.000.000
F.02	Subnet MASK	000.000.000.000 - 255.255255.255	000.000.000.000
F.03	Porta IP	0 - 32000	1001
F.04	Client / Server	Client / Server	Server
F.05	Indirizzo IP remoto	000.000.000.000 - 255.255255.255	000.000.000.000
F.06	Porta IP remota	0 - 32000	1001
F.07	Indirizzo gateway IP	000.000.000.000 - 255.255255.255	000.000.000.000

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 a 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 la presenza di errori di trasmissione. Questo consente, in caso di disturbo sulla linea di trasmissione, di ignorare il messaggio inviato per evitare problemi sia dal lato master, sia da quello 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 almeno a 3.5 volte il tempo richiesto per l'invio di un carattere. Il controllore 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 carattere successivo viene considerato come inizio di un nuovo messaggio.

FUNZIONI MODBUS®

Le funzioni disponibili sono:

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

Per esempio, se si vuole leggere dal controllore con indirizzo 01 il valore della temperatura interno quadro che si trova alla locazione 14 (0E hex), il messaggio da spedire è il seguente:

01	04	00	2D	00	02	E0	08
----	----	----	----	----	----	----	----

dove:

01 = indirizzo slave

04 = funzione di lettura locazione

00 2D = indirizzo della locazione contenente il valore della temperatura interno quadro diminuito di un'unità

00 02 = numero di registri da leggere a partire dall'indirizzo 2D

E0 08 = checksum CRC

La risposta del controllore 8BGA è la seguente:

01	04	04	00	00	00	1C	FA	4D
----	----	----	----	----	----	----	----	----

dove:

01 = indirizzo del controllore (slave 01)

04 = funzione richiesta dal Master

04 = numero di byte inviati da 8BGA

00 00 00 1C = valore esadecimale temperatura = 28 = 28°C

FA 4D = checksum CRC

Funzione 04: read input register

La funzione 04 permette di leggere una o più grandezze consecutive in memoria. L'indirizzo di ciascuna grandezza è indicato nella tabella 2 riportata più in là nel presente documento.

Come da standard Modbus®, l'indirizzo specificato nel messaggio va diminuito di 1 rispetto a quello effettivo riportato nelle tabelle. Se l'indirizzo richiesto non è compreso nella tabella o il numero di registri richiesti è maggiore del numero consentito, il controllore ritorna un messaggio di errore (vedi tabella errori).

Richiesta Master:

Indirizzo slave	08h
Funzione	04h
MSB indirizzo registro	00h
LSB indirizzo registro	0Fh
MSB numero registri	00h
LSB numero registri	08h
LSB CRC	C1h
MSB CRC	56h

Nell'esempio, allo slave numero 08 vengono richiesti 8 registri consecutivi a partire dall'indirizzo 10h. Quindi vengono letti i registri da 10h a 17h. Il comando termina sempre con il valore di checksum CRC.

Risposta Slave:

Indirizzo slave	08h
Funzione	04h
Numero di byte	10h
MSB dato 10h	00h
LSB dato 10h	00h
-----	---
MSB dato 17h	00h
LSB dato 17h	00h
LSB CRC	5Eh
MSB CRC	83h

La risposta è composta sempre dall'indirizzo dello slave, dalla funzione richiesta dal Master e dai dati dei registri richiesti. La risposta termina sempre con il valore di checksum CRC.

Funzione 06: preset single register

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

L'indirizzo e il range valido per i vari parametri può essere trovato nelle tabelle 5, 6 e 7.

Richiesta Master:

Indirizzo slave	08h
Funzione	06h
MSB indirizzo registro	2Fh
LSB indirizzo registro	0Fh
MSB dato	00h
LSB dato	0Ah
LSB CRC	31h
MSB CRC	83h

Risposta slave:

La risposta è un'eco della domanda, cioè sono inviati 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 controllore.

Richiesta Master:

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

La tabella seguente riporta il significato del byte inviato dal controllore 8BGA come risposta:

BIT	SIGNIFICATO
0	Verifica checksum memoria programma
1	
2	
3	
4	
5	
6	
7	

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 1 (Tipo) ❶	49h
Dato 2 (Revisione software)	00h
Dato 3 (Revisione hardware)	00h
Dato 4 (Revisione parametri)	00h
Dato 5 (tipologia prodotto) ❷	01h
Dato 6 (riservato)	00h
Dato 7 (riservato)	00h
LSB CRC	00h
MSB CRC	F7h

- ❶ 67 – 43h = non usato
69 – 45h = 5LGA
73 – 49h = 8LGA

- ❷ 1 – 01h controllori LGA

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

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 P53 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 a una domanda. La massima lunghezza consentita è di 80 registri consecutivi.

- Il campo LRC 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.
- Il messaggio termina sempre con i caratteri di controllo CRLF (OD OA).

Per esempio, se si vuole leggere dal dispositivo con indirizzo 8 il valore della tensione equivalente che si trova alla locazione 4 (04 Hex), il messaggio da spedire è il seguente:

:	08	04	00	03	00	02	56	CRLF
---	----	----	----	----	----	----	----	------

dove:

: = ASCII 3Ah = Delimitatore inizio messaggio

08 = indirizzo slave.

04 = funzione di lettura locazione.

00 03= indirizzo della locazione diminuito di un'unità contenente il valore della tensione equivalente

00 02 = numero di registri da leggere a partire dall'indirizzo 04.

56 = checksum LRC.

CRLF = ASCII 0Dh 0Ah = delimitatore fine messaggio

La risposta è la seguente:

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

dove:

: = ASCII 3Ah = Delimitatore inizio messaggio

08 = indirizzo del controllore (Slave 08).

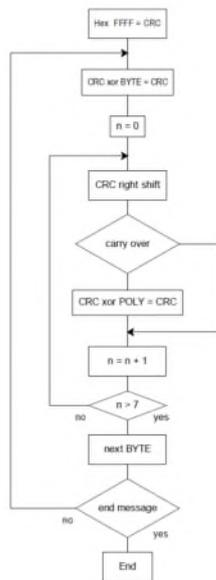
04 = funzione richiesta dal Master.

04 = numero di byte inviati dallo slave.

00 00 01 3D = valore esadecimale della tensione misurata (317V)

F1 = checksum LRC.

CRLF = ASCII 0Dh 0Ah = delimitatore fine messaggio



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
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
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il	1100	1111	1111	1110
polinomio				
Esegue terzo shift dx	0110	0111	1111	1111
Esegue quarto shift dx	0011	0011	1111	1111
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
Esegue sesto shift dx	0010	0100	1111	1111
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1000	0100	1111	1110
Polinomio				
Esegue settimo shift dx	0100	0010	0111	1111
Esegue ottavo shift dx	0010	0001	0111	1111
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1000	0001	0111	1110
Polinomio				
Esegue primo shift dx	0100	0000	1001	1100
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1110	0000	1001	1101
Polinomio				
Esegue secondo shift dx	0111	0000	0100	1110
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1101	0000	0100	1111
Polinomio				
Esegue terzo shift dx	0110	1000	0010	0111
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1100	1000	0010	0110
Polinomio				
Esegue quarto shift dx	0110	0100	0001	0011
Esegue quinto shift dx	0010	0100	0000	1001
Carry=1, carica polinomio	1010	0000	0000	0001
Esegue xor con il polinomio	1001	0010	0000	1000
Polinomio				
Esegue sesto shift dx	0100	1001	0000	0100
Esegue settimo shift dx	0010	0100	1000	0010
Esegue ottavo shift dx	0001	0010	0100	0001
Risultato CRC	0001	0010		
0100	0001			
	12h	41h		

Nota: il byte 41h viene spedito per primo (anche se è il LRC), 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

tabella 2: misure fornite dal protocollo di comunicazione (utilizzabili con funzioni 03 e 04)

INDIRIZZO	WORDS	MISURA ISTANTANEA (IN)	UNITA'	FORMATO
02h	2	Cos phi attuale		Unsigned long
04h	2	Tensione	V	Unsigned long

INDIRIZZO	WORDS	MISURA ISTANTANEA (IN)	UNITA'	FORMATO
06h	2	Corrente	A / 1000	Unsigned long
08h	2	Delta kvar ❷	kvar	Signed long
0Ah	2	Fattore di potenza medio settimanale	/100	Unsigned long
0Ch	1	Corrente sovraccarico condensatori %	%	Unsigned integer
0Eh	2	Temperatura quadro ❷	°C	Signed long
10h	2	Stato uscite		Unsigned long
12h	2	Sfasamento	° / 4	Unsigned long
14h	2	Fattore di potenza attuale		Signed long
16h	2	Stato bit di errore ❸		Unsigned long
18h	2	Max tensione	V	Unsigned long
1Ah	2	Max corrente	A / 100	Unsigned long
1Ch	2	Max sovraccarico condensatori	%	Unsigned long
1Eh	2	Max temperatura	°C	Unsigned long
200h	2	Tempo inserzione gradino 1	min	Unsigned integer

21Ah	2	Tempo inserzione gradino 14	min	Unsigned integer
2020h	2	Numero inserzioni gradino 1		Unsigned integer

202Dh	2	Numero inserzioni gradino 14		Unsigned integer
2040h	2	Potenza residua gradino 1	kvar	Unsigned integer

204Dh	2	Potenza residua gradino 14	kvar	Unsigned integer
2008h	1	THDV	% / 10	Unsigned integer
2080h	1	2° armonica tensione	% / 10	Unsigned integer

208Dh	1	15° armonica tensione	% / 10	Unsigned integer
2009h	1	THDI	% / 10	Unsigned integer
20A0h	1	2° armonica corrente	% / 10	Unsigned integer

20ADh	1	15° armonica corrente	% / 10	Unsigned integer
2007h	1	Scadenza manutenzione	h	Signed integer

❶ Il bit 31 indica il segno (0=positivo; 1=negativo), mentre il bit 30 indica induttivo/capacitivo (0=ind; 1=cap)

❷ Il bit 31 indica il segno (0=positivo; 1=negativo)

❸ Leggendo le word all'indirizzo 16h vengono restituiti 32 bit con significato come da tabella 3:

tabella 3: codifica allarmi

BIT #	TIPO ALLARME
0	A01 sottocompensazione
1	A02 sovracompensazione
2	A03 corrente impianto troppo bassa
3	A04 corrente impianto troppo alta
4	A05 tensione impianto troppo bassa
5	A06 tensione impianto troppo alta
6	A07 sovraccarico condensatore
7	A08 temperatura troppo alta
8	A09 microinterruzione
9	A10 THD tensione troppo alto
10	A11 THD corrente impianto troppo alta
11	A12 Richiesta manutenzione
12	A13 Step difettoso
13 - 31	Bit liberi

tabella 4: Comandi (utilizzabili con funzione 06)

INDIRIZZO	WORDS	COMANDO	FORMATO
2FF0h	1	Esecuzione comandi menu comandi ❶	Unsigned integer
3000h	1	Cambio modalità operative ❷	Unsigned integer
3001h	1	Reset apparecchio ❸	Unsigned integer
3005h	1	Attivazione di un gradino ❹	Unsigned integer
3006h	1	Disattivazione gradino ❺	Unsigned integer
3007h	1	Blocco tastiera ON/OFF ❻	Unsigned integer

❶ La tabella indica le funzioni associate al valore da scrivere all'indirizzo 2FF0h. E' possibile eseguire più di una funzione contemporaneamente.

tabella 5:

VALORE	FUNZIONE
00h	C01 azzerà intervallo manutenzione
01h	C02 azzerà il contatore di manovre gradini
02h	C03 ripristina le potenze originali dell'aggiustamento gradino
03h	C04 azzerà il contatore di funzionamento gradino
04h	C05 azzerà i picchi massimi registrati
05h	C06 azzerà memoria fattore di potenza totale settimanale
06h	C07 ripristina i parametri al default di fabbrica
07h	C08 salva copia di backup delle impostazioni utente
08h	C09 ripristina i parametri al valore della copia utente

② La seguente tabella indica i valori da scrivere all'indirizzo 3000h per ottenere le corrispondenti funzioni.

VALORE	FUNZIONE
0	Passaggio MAN / AUT e viceversa
1	Passaggio a modalità MAN
2	Passaggio a modalità AUT

③ Scrivendo il valore 01 all'indirizzo indicato viene eseguita la funzione corrispondente.

④ Scrivere nel registro indicato il numero del gradino da attivare / disattivare. Se si tenta di attivare un gradino per il quale è in corso il tempo di riconnessione, il comando non verrà eseguito.

⑤ Scrivendo il valore 01 all'indirizzo indicato viene bloccata la tastiera, scrivendo 00 la stessa viene sbloccata.

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 controllore.

PROCEDURA PER LA LETTURA DEI PARAMETRI

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

PROCEDURA PER LA SCRITTURA DEI PARAMETRI

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

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

① E' possibile leggere il valore del parametro memorizzato all'indirizzo **5002h** utilizzando la **funzione 4**.

ESEMPIO

Impostare a 250 il valore del parametro P.01

Passo 1: impostazione parametro P.01.

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

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

controllore Funzione = 6
Indirizzo = 5002h (500h - 0001h = 5001h)
Valore = 1 (01h)

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

Passo 2: Impostazione valore 250

MASTER Funzione = 16 (10h)
Indirizzo = 5003h (5003h - 0001h = 5002h)
Nr. registri = 2 (02h)
Nr. Byte = 4 (04h)
Valore = 250 (00000FAh)

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

controllore Funzione = 16 (10h)
Indirizzo = 5003h (5004h - 0001h = 5003h)
Valore = 250 (00000FAh)

01	10	50	02	00	02	F1	08
----	----	----	----	----	----	----	----

Passo 3: Salvataggio e riavvio

MASTER Funzione = 6 (06h)
Indirizzo = 2F03h (2F03h - 0001h = 2F02h)
Valore = 1 (01H)

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

controllore Nessuna risposta

INFORMAZIONI E ASSISTENZA TECNICA



ICAR by ORTEA NEXT

www.next.ortea.com - ortea@ortea.com



ORTEA SpA
Via dei Chiosi, 21
20873 Cavenago Brianza - Milan - ITALY
Tel.: ++39 02 95917800

WARNING

Installation, setting, inspection and maintenance operations must be performed only by qualified personnel in charge of it. Any operation must be carried out in compliance with the enforced regulations and legislation concerning personal safety and the use of adequate protective tools.

INTRODUCTION

The digital power factor controller 5LGA and 8LGA support the communication protocols Modbus® RTU, Modbus® ASCII and Modbus® TCP (only 8LGA) on the optical interface and the expansion modules:

- USB
- RS232
- RS485
- Ethernet (only 8LGA)

Using this function, it is possible to read the device status and to control the units through standard third party supervision software (SCADA) or through other devices fitted with Modbus® interface, such as PLCs and smart terminals.

PARAMETER SETTING

To configure the Modbus® protocol, enter the SETUP MENU and choose the ADV menu. Only one communication port can be configured.

ADVANCED MENU – SERIAL COMMUNICATION

PAR	FUNCTION	RANGE	DEFAULT
P49	Indirizzo nodo	1 ..255	1
P50	Velocità seriale	1200, 2400, 4800, 9600, 19200, 38400	9600
P51	Formato dati	8 bit Nessuna, 8 bit Dispari, 8 bit Pari, 7 bit Dispari, 7 bit Pari	8 bit Nessuna
P52	Stop bit	1, 2	1
P53	Protocollo	Modbus RTU, Modbus ASCII, Modbus TCP (solo 8LGA)	Modbus RTU

For the Ethernet expansion module (only 8LGA), additional parameters are available. Enter the SETUP MENU and select the FUN menu.

PAR	FUNCTION	RANGE	DEFAULT
F.01	IP address	000.000.000.000 - 255.255255.255	000.000.000.000
F.02	Subnet mask	000.000.000.000 - 255.255255.255	000.000.000.000
F.03	IP port	0 - 32000	1001
F.04	Client / Server	Client / Server	Server
F.05	Remote IP address	000.000.000.000 - 255.255255.255	000.000.000.000
F.06	Remote IP port	0 - 32000	1001
F.07	IP gateway address	000.000.000.000 - 255.255255.255	000.000.000.000

MODBUS® RTU PROTOCOL

If the Modbus® RTU protocol is selected, 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 device to which the message is destined.
- 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 no.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 the time during which data must not be exchanged on the communication bus to allow the connected devices to recognize the end of one message and the beginning of another. This time must be at least 3.5 times the time required to send one character.

The controller measures the time that elapses from the reception of one character and the following one . 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 controller measures.
04 = Read input register	Allows to read the controller measures.
06 = Preset single register	Allows to write the parameters
07 = Read exception	Allows to read the device status
10 = Preset multiple register	Allows to write several parameters
17 = Report slave ID	Allows to read information about the device.

For instance, in order to read from the controller with serial address 01 the value of the temperature inside the panel (residing at location 14 (0E Hex), the message to send is the following:

01	04	00	2D	00	02	E0	08
----	----	----	----	----	----	----	----

where:

01= slave address

04 = Modbus® function 'Read input register'

00 2D = Address of the required register (temperature) decreased by one

00 02 = Number of registers to be read beginning from address 2D

EE 08 = CRC Checksum

The controller answer is the following:

01	04	04	00	00	00	1C	FA	4D
----	----	----	----	----	----	----	----	----

where:

01 = device address (Slave 01)

04 = Function requested by the master

04 = Number of bytes sent by the device

00 00 00 1C = Hex value of the temperature = 28 = 28°C

FA 4D = CRC checksum

Function 04: read input register

The Modbus® function 04 allows to read one or more consecutive registers from the slave memory. The address of each measure is given in tables 2 later on in this document.

As per 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 controller will return an error message (see error table).

Master query:

Slave address	08h
Function	04h
MSB address	00h
LSB address	0Fh
MSB register number	00h
LSB register number	08h
LSB CRC	C1h
MSB CRC	56h

In the above example, slave 08 is requested for 8 consecutive registers beginning with address 10h. Thus, registers from 10h to 17h will be returned. As usual, the message ends with the CRC checksum.

Slave response:

address	08h
Function	04h
Byte number	10h
MSB register 10h	00h
LSB register 10h	00h
-----	---
MSB register 17h	00h
LSB register 17h	00h
LSB CRC	5Eh
MSB CRC	83h

The response is always composed of the slave address, the function code requested by the master and the contents of the requested registers. The answer ends with the CRC.

Function 06: preset single register

This function allows to write in the registers. It can be used only with registers with address higher than 1000 Hex. For instance, it is possible to change setup parameters. If the value is not in the correct range, the 8BGA 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 Tables 5, 6 and 7.

Master query:

Slave address	08h
Function	06h
MSB register address	2Fh
LSB register address	0Fh
MSB data	00h
LSB data	0Ah
LSB CRC	31h
MSB CRC	83h

Slave response:

The slave response is an echo to the query, which means that the slave sends back to the master the address and the new value of the variable.

Function 07: read exception status

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

Master query:

Slave address	08h
Function	07h
LSB CRC	41h
MSB CRC	B2h

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

BIT	MEANING
0	Program memory checksum check
1	
2	
3	
4	
5	
6	
7	

Function 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
Byte counter	08h
Date 1 (Type) ❶	49h
Date 2 (software revision)	00h
Date 3 (hardware revision)	00h
Date 4 (parameter revision)	00h
Date 5 (device type) ❷	01h
Date 6 (reserved)	00h
Date 7 (reserved)	00h
LSB CRC	00h
MSB CRC	F7h

- ❶ 67 – 43h = not used
69 – 45h = 5LGA
73 – 49h = 8LGA

- ❷ 1 – 01h LGA controllers

Errors

In case the slave receives an incorrect message, it answers with a message composed by the queried function OR with 80 Hex, followed by an error code byte. In the following table the error codes sent by the slave to the master are listed:

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

MODBUS® ASCII PROTOCOL

The Modbus® ASCII protocol is normally used in applications that require to communicate through modem.

Functions and addresses available are the same as the ones 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 the parameter P.53 as Modbus® ASCII protocol is selected, the communication message on the correspondent communication port has the following structure:

:	Address 2 chars	Function 2 chars	Data (N. chars)	LRC 2 chars	CR LF
---	--------------------	---------------------	--------------------	----------------	-------

- The Address field holds the serial address of the slave device to which the message is destined.
- 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 allowed length is of 8 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 therefore ignore the message.
- The message always ends with CRLF control character (0D 0A).

For instance, in order to read from the slave with serial address 08 the equivalent voltage residing at location 4 (04 Hex), the message to send is the following:

:	08	04	00	03	00	02	56	CRLF
---	----	----	----	----	----	----	----	------

where:

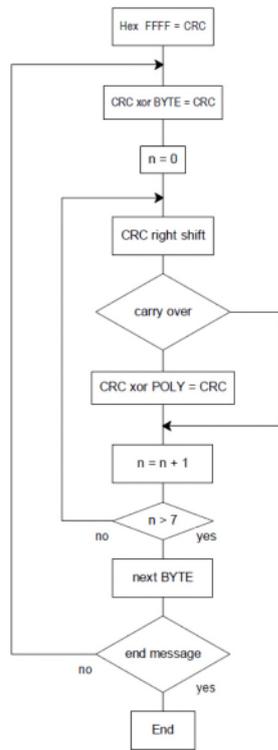
- : = ASCII 3Ah message start delimiter
- 08 = slave address
- 04 = Modbus® function 'Read input register'
- 00 03 = Address of the required register (voltage) decreased by one
- 00 02 = Number of registers to be read beginning from address 04
- 56= LRC Checksum
- CRLF = ASCII 0Dh 0Ah = Message end delimiter

The response is the following:

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

where:

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



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
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1101 1111 1111 1111
Execute 2nd right shift
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1100 1111 1111 1110
Execute 3rd right shift
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1001 0011 1111 1110
Execute 4th right shift
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1000 0100 1111 1110
Execute 5th right shift
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1000 0100 1111 1110
Execute 6th right shift
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1000 0100 1111 1110
Execute 7th right shift
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1000 0100 1111 1110
Execute 8th right shift
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
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1110 0000 1001 1101
Execute 2nd right shift
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1101 0000 0100 1111
Execute 3rd right shift
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1100 1000 0010 0110
Execute 4th right shift
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1001 0010 0000 1000
Execute 5th right shift
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           1000 1001 0000 0100
Execute 6th right shift
Carry=1, load polynomial 1010 0000 0000 0001
Execute xor with the
polynomial           0001 0010 0100 0001
CRC Result           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
  
```

table 2: measured supplied by serial communication protocol (to be used with functions 03 e 04)

ADDRESS	WORDS	MEASURE	UNIT	FORMAT
02h	2	Cos phi ❶		Unsigned long
04h	2	Voltage	V	Unsigned long
06h	2	Current	A / 1000	Unsigned long
08h	2	Delta kvar ❷	kvar	Signed long
0Ah	2	Weekly average power factor	/100	Unsigned long
0Ch	1	Capacitor overload current	%	Unsigned integer
0Eh	2	Panel temperature ❸	°C	Signed long

ADDRESS	WORDS	MEASURE	UNIT	FORMAT
10h	2	Output status		Unsigned long
12h	2	Phase angle offset	° / 4	Unsigned long
14h	2	Power factor		Signed long
16h	2	Error bit status ❶		Unsigned long
18h	2	Max voltage	V	Unsigned long
1Ah	2	Max current	A / 100	Unsigned long
1Ch	2	Max capacitor overload	%	Unsigned long
1Eh	2	Max temperature	°C	Unsigned long
200h	2	Step 1 insertion time	min	Unsigned integer

21Ah	2	Step 14 insertion time	min	Unsigned integer
2020h	2	Step 1 no. of insertions		Unsigned integer

202Dh	2	Step 14 no. of insertions		Unsigned integer
2040h	2	Step 1 residual power	kvar	Unsigned integer

204Dh	2	Step 14 residual power	kvar	Unsigned integer
2008h	1	THDV	% / 10	Unsigned integer
2080h	1	2nd order voltage harmonic	% / 10	Unsigned integer

208Dh	1	15th order voltage harmonic	% / 10	Unsigned integer
2009h	1	THDI	% / 10	Unsigned integer
20A0h	1	2nd order current harmonic	% / 10	Unsigned integer

20ADh	1	15th order current harmonic	% / 10	Unsigned integer
2007h	1	Hours to maintenance	h	Signed integer

❶ Bit 31 indicates the sign (0=positive; 1=negative), while bit 30 indicates inductive/capacitive load (0=ind; 1=cap)

❷ Bit 31 indicates the sign (0=positive; 1=negative)

❸ By reading the words starting at address 16h, 32 bits will be returned with the meaning as per table 3:

table 3: codifica allarmi

BIT #	ALARM
0	A01 undercompensation
1	A02 overcompensation
2	A03 low current
3	A04 high current
4	A05 low voltage
5	A06 high voltage
6	A07 capacitor overload
7	A08 overtemperature
8	A09 microinterruption
9	A10 voltage THD too high
10	A11 current THD too high
11	A12 maintenance required
12	A13 step failure
13 - 31	Free bits

table 4: Commands (to be used with function 06)

ADDRESS	WORDS	COMMAND	FORMAT
2FF0h	1	Commands list ❶	Unsigned integer
3000h	1	Operational mode change ❷	Unsigned integer
3001h	1	Device reset ❸	Unsigned integer
3005h	1	Step activation ❹	Unsigned integer
3006h	1	Step de-activation ❹	Unsigned integer
3007h	1	Keyboard lock ON/OFF ❺	Unsigned integer

❶ The following table shows the functions associated to the value to be written at the address 2FF0h. It is possible to activate several functions at the same time.

table 5:

VALUE	FUNCTION
00h	C01 reset maintenance service interval
01h	C02 reset step operation counter
02h	C03 reset original powers in step adjusting
03h	C04 reset the step operation hour counter
04h	C05 reset maximum peak values
05h	C06 reset weekly total power factor history
06h	C07 reset parameters to factory default
07h	C08 create a backup copy of the user settings
08h	C09 reload parameters using the user backup copy

❷ The following table indicates the values to be written at address 3000h to achieve the corresponding functions:

VALUE	FUNCTION
0	Switch from MAN to AUT and vice-versa

VALUE	FUNCTION
1	Switch to MAN mode
2	Switch to AUT mode

- ③ By writing 01 at the indicated address, the corresponding function will be activated.
- ④ Write in the corresponding register the number of the step to be activated/de-activated. An attempt to activate a step while the reconnection time is running will be ignored.
- ⑤ By writing 01 at the indicated address the keyboard will be locked. By writing 00, it will be unlocked.

PARAMETER SETTING

By using the Modbus® protocol, the menu parameters can be accessed. In order to understand correctly the correspondence between the numeric value and the selected function and/or the unit of measure, please see the 8BGA operating manual.

PROCEDURE FOR READING PARAMETERS

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

PROCEDURE FOR WRITING PARAMETERS

1. Write the value of the menu to be changed by using **function 6** at address **5002h**. ①
2. Perform **function 16** at address **5003h**, with a number of registers appropriate to the length of the parameter
3. To write the next parameter, repeat step 2, otherwise perform step 1. If no additional parameters need to be written, go to step 4.
4. To make the changes to setup parameters effective, it is necessary to store the values in the EEPROM using the dedicated command. Write 5 by using **function 6** at address **2F03h**.

TYPE OF PARAMETER	NUMBER OF REGISTER
Numeric value ≤ 65535 (for example, P.30)	1 register (2 byte)
Numeric value > 65535 (for example, P.6)	2 registers (4 byte)

① It is possible to read the parameter stored at the address **5002h** by using **function 4**.

Example

Set the value of parameter P.01 to 250

Step 1: P.01 parameter setup

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

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

controller Function = 6
 Address = 5002h (500h – 0001h = 5001h)
 Value = 1 (01hH)

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

Step 2: value 250 setup

MASTER Function = 16 (10h)
 Address = 5003h (5003h – 0001h = 5002h)
 Register no. = 2 (02h)
 Byte no. = 4 (04h)
 Value = 250 (00000FAh)

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

controller Function = 16 (10h)
 Address = 5003h (5004h – 0001h = 5003h)
 Value = 250 (00000FAh)

01	10	50	02	00	02	F1	08
----	----	----	----	----	----	----	----

Step 3: save and reboot

MASTER Function = 6 (06h)
 Address = 2F03h (2F03h – 0001h = 2F02h)
 Value = 1 (01H)

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

controller No answer

SERVICE



ICAR by ORTEA NEXT
 www.icar.com – ortea@ortea.com



ORTEA SpA
 Via dei Chiosi, 21
 20873 Cavenago Brianza – Milan – ITALY
 Tel.: ++39 02 95917800