


**GB AUTOMATIC TRANSFER SWITCH CONTROLLER**
**Communication protocol Modbus®**
**I COMMUTATORE DI RETE AUTOMATICO**
**Protocollo di comunicazione Modbus®**
**ATL610**

**MODBUS® PROTOCOL**

The ATL610 automatic transfer switch controller support the communication protocols Modbus-RTU®, Modbus-ASCII®, Modbus-TCP® on optical interface and the expansion modules:

- EXP1010 USB
- EXP1011 RS232
- EXP1012 RS485
- EXP1013 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 M08 menu: It is possible to configure 2 different communication ports (n=1...2).

**MENU M08 – COMMUNICATION**

Serial communication.

PAR	Function / Funzione	Default	Range
P08.n.01	Node address / Indirizzo nodo	01	01-255
P08.n.02	Serial port speed / Velocità seriale	9600	1200 2400 4800 9600 19200 38400 57600 115200
P08.n.03	Data format / Formato dati	8 bit – n	8 bit–no par. 8 bit, odd 8 bit, even 7 bit, odd 7 bit, even
P08.n.04	Stop bits / Bit di stop	1	1-2
P08.n.05	Protocol / Protocollo	Modbus-RTU	Modbus-RTU Modbus-ASCII Modbus-TCP

For expansion module EXP1013 (Ethernet), there are other parameters.

Per il modulo di espansione EXP1013 (Ethernet) esistono altri parametri.

PAR	Function / Funzione	Default	Range
P08.n.06	IP address / Indirizzo IP	192.168.1.1	000.000.000.000 – 255.255.255.255
P08.n.07	Subnet mask / Subnet mask	0.0.0.0	000.000.000.000 – 255.255.255.255
P08.n.08	IP port / Porta IP	1001	0-32000
P08.n.09	Channel function / Funzione canale	Slave	Slave Gateway
P08.n.10	Client/server / Client/server	Server	Client Server
P08.n.11	Remote IP address / Indirizzo IP remoto	0.0.0.0	000.000.000.000 – 255.255.255.255
P08.n.12	Remote IP port / Porta IP remota	1001	0-32000
P08.n.13	IP gateway address / Indirizzo gateway IP	0.0.0.0	000.000.000.000 – 255.255.255.255

## MODBUS-RTU® PROTOCOL

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

T1	Address ( 8 bit)	Function (8 bit)	Data (N x 8 bit)	CRC (16 bit)	T1
T2					T2
T3					T3

- the Address field holds the serial address of the slave destination device
- the Function field holds the code of the function that must be executed by the slave
- the Data field contains data sent to the slave or data received from the slave in response to a query
- the maximum length for the data field is 80 16 bit registers (160 bytes)
- the CRC field allows the master and slave devices to check the message integrity  
If a message has been corrupted by electrical noise or interference, the CRC field allows the devices to recognize the error and thereby to ignore the message
- the T1 T2 T3 sequence corresponds to a time in which data must not be exchanged on the communication bus to allow the connected devices to recognize the end of one message and the beginning of another. This time must be at least 3.5 times the time required to send one character.

The ATL610 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 ATL measures
04 = Read input register	Allows to read the ATL measures
06 = Preset single register	Allows writing parameters
07 = Read exception	Allows to read the device status
10 = Preset multiple register	Allows writing several parameters
17 = Report slave ID	Allows to read information about the device

For instance, to read the number number of switching alarms of breaker 1, which resides at location 58 (3A Hex), from the ATL610 with serial address 01, the message to send is the following:

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

Where:

- 01 = slave address
- 04 = Modbus® function "Read input register"
- 00 39 = 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 22
- A1 C6 = CRC checksum.

The ATL610 answer is the following:

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

Where:

- 01 = ATL610 address (slave 01)
- 04 = function requested by the master
- 04 = number of bytes sent by the ATL610
- 00 00 00 07 = Hex value of number of switching alarms of breaker 1 = 7
- BA 46 = CRC checksum.

## FUNCTION 04: READ INPUT REGISTER

The 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 ATL610 will return an error code (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

## PROTOCOLLO MODBUS-RTU®

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

T1	Indirizzo ( 8 bit)	Funzione (8 bit)	Dati (N x 8 bit)	CRC (16 bit)	T1
T2					T2
T3					T3

- il campo Indirizzo contiene l'indirizzo dello strumento slave cui il messaggio viene inviato;
- il campo Funzione contiene il codice della funzione che deve essere eseguita dallo slave;
- il campo Dati contiene i dati inviati allo slave o quelli inviati dallo slave come risposta ad una domanda;
- la lunghezza massima consentita per il campo dati è di 80 registri da 16 bit (160 bytes);
- il campo CRC consente sia al master 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.

L'ATL610 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 nell'ATL
04 = Read input register	Consente la lettura delle misure disponibili nell'ATL
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 dall' ATL610 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 dell' ATL610 è la seguente:

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

Dove:

- 01 = indirizzo dell' ATL610 (slave 01);
- 04 = funzione richiesta dal master;
- 04 = numero di byte inviati dall'ATL610;
- 00 00 00 07 = valore esadecimale del numero di allarmi di commutazione dell'interruttore 1 = 7;
- BA 46 = 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, l' ATL610 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

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

Slave response:

Slave address	08h
Function	04h
Byte number	10h
MSB register 10h	00h
LSB register 10h	00h
-----	----
MSB register 17h	00h
LSB register 17h	00h
LSB CRC	8Ah
MSB CRC	B1h

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 ATL610 will answer with an error message. In the same way, if the parameter address is not recognised, the ATL610 will send an error response.

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

Master message:

Slave address	08h
Function	06h
MSB address	2Fh
LSB address	0Fh
MSB register	00h
LSB register	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 automatic transfer switch.

Master query:

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

The following table gives the meaning of the status byte sent by the ATL610 as answer:

BIT	MEANING
0	Operative mode OFF / Reset
1	Operative mode MAN
2	Operative mode AUT
3	Operative mode TEST
4	On error
5	AC power supply ok
6	DC power supply ok
7	Global alarm ON

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

Risposta Slave:

Indirizzo slave	08h
Funzione	04h
Numero di byte	10h
MSB dato 10h	00h
LSB dato 10h	00h
-----	----
MSB dato 17h	00h
LSB dato 17h	00h
LSB CRC	8Ah
MSB CRC	B1h

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

#### FUNZIONE 06: PRESET SINGLE REGISTER

Questa funzione permette di scrivere nei registri. Essa può essere utilizzata solo con i registri d'indirizzo superiore a 1000 Hex. È possibile ad esempio impostare i parametri del setup. Qualora il valore impostato non rientri nel valore minimo e massimo della tabella l'ATL610 risponderà con un messaggio di errore. Se viene richiesto un parametro ad un indirizzo inesistente verrà risposto con un messaggio di errore.

L'indirizzo ed il range valido per i vari parametri può essere trovato nella Tabella 4.

Richiesta master:

Indirizzo slave	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 commutatore di linea.

Richiesta master:

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

La tabella seguente riporta il significato del byte inviato dall'ATL610 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

## FUNCTION 16: PRESET MULTIPLE REGISTER

This function allows to modify multiple parameters with a single message, or to preset a value longer than one register.

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

## FUNCTION 17: REPORT SLAVE ID

This function allows to identify the device type.

Master query		Slave response	
Slave address	08h	Slave address	08h
Function	11h	Function	11h
LSB CRC	C6h	Contatore bytes	08h
MSB CRC	7Ch	Data 01 (type) ❶	76h
		Dato 02 (sw revision)	01h
		Dato 03 (hardware revision)	00h
		Dato 04 (parameter revision)	01h
		Dato 05 (type of device) ❷	04h
		Dato 06 (reserved)	00h
		Dato 07 (reserved)	00h
		Dato 08 (reserved)	00h
		LSB CRC	B0h
		MSB CRC	2Ah

❶ 118 - 76h = ATL610.

❷ 4 - 04h = ATL series.

## 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 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

## 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 P7.x.05 or P7.05 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. 2) 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).

## FUNZIONE 16: PRESET MULTIPLE REGISTER

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

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

## FUNZIONE 17: REPORT SLAVE ID

Questa funzione permette di identificare il tipo di dispositivo.

Richiesta master		Risposta slave	
Indirizzo slave	08h	Indirizzo slave	08h
Funzione	11h	Funzione	11h
LSB CRC	C6h	Contatore bytes	08h
MSB CRC	7Ch	Dato 01 (tipo) ❶	76h
		Data 02 (revisione software)	01h
		Data 03 (revisione hardware)	00h
		Data 04 (revisione parametri)	01h
		Data 05 (tipologia di prodotto) ❷	04h
		Data 06 (riservato)	00h
		Data 07 (riservato)	00h
		Data 08 (riservato)	00h
		LSB CRC	B0h
		MSB CRC	2Ah

❶ 118 - 76h = ATL610.

❷ 4 - 04h = Serie ATL.

## 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 P7.x.05 o 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	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. 2) 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).

#### Example

For instance, to read the value of the current phase L3, which resides at location 12 (0C Hex) from the slave with serial address 08, the message to send is the following:

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

#### Where:

: = ASCII 3Ah message start delimiter  
08 = slave address  
04 = Modbus® function "Read input register"  
00 0B = address of the required register (L3 current phase ) decreased by one  
00 02 = number of registers to be read beginning from address 04  
E7 = LRC checksum  
CRLF = ASCII 0Dh 0Ah = message end delimiter.

The ATL610 answer is the following:

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

#### Where:

: = ASCII 3Ah message start delimiter  
08 = ATL610 address (slave 08)  
04 = function requested by the master  
04 = number of bytes sent by the multimeter  
00 00 A8 AE = hex value of the current phase of L3 (= 4.3182 A)  
9B = LRC checksum  
CRLF = ASCII 0Dh 0Ah = message end delimiter.

#### Esempio

Per esempio, se si vuole leggere dall'ATL610 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 3Ah = 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 0Dh 0Ah = delimitatore fine messaggio.

La risposta dell'ATL610 è la seguente:

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

#### Dove:

: = ASCII 3Ah = Delimitatore inizio messaggio;  
08 = indirizzo dell'ATL610 (slave 08);  
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 0Dh 0Ah = delimitatore fine messaggio.

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			
	12h	41h			

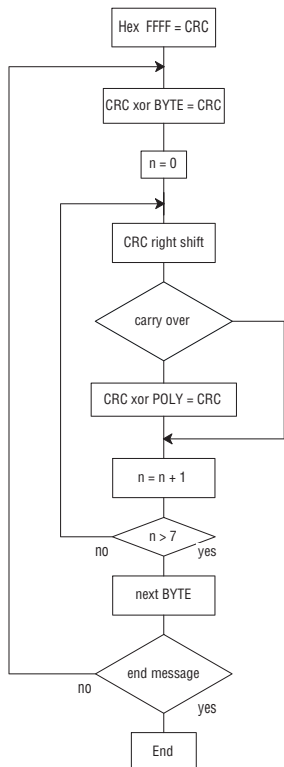
CALCOLO DEL CRC (CHECKSUM per RTU)

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 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			
	12h	41h			

Note: the byte 41h is sent first (even if it is the LSB) then 12h is sent.

Nota: il byte 41h viene spedito per primo (anche se è il LSB), poi viene trasmesso 12h.



CRC calculation algorithm.  
Algoritmo di calcolo del CRC.

## 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	

## 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	

TABLE 2:  
MEASURES SUPPLIED BY SERIAL COMMUNICATION PROTOCOL  
(to be used with functions 03 and 04).

TABELLA 2:  
MISURE FORNITE DAL PROTOCOLLO DI COMUNICAZIONE  
(utilizzabili con funzioni 03 e 04).

ADDRESS INDIRIZZO	WORDS	MEASURE	MISURA	UNIT UNITA'	FORMAT FORMATO
02h	2	Voltage of LINE 1 L1-N	Tensione LINEA 1 L1-N	V	Unsigned long
04h	2	Voltage of LINE 1 L2-N	Tensione LINEA 1 L2-N	V	Unsigned long
06h	2	Voltage of LINE 1 L3-N	Tensione LINEA 1 L3-N	V	Unsigned long
08h	2	Voltage of LINE 1 L1-L2	Tensione LINEA 1 L1-L2	V	Unsigned long
0Ah	2	Voltage of LINE 1 L2-L3	Tensione LINEA 1 L2-L3	V	Unsigned long
0Ch	2	Voltage of LINE 1 L3-L1	Tensione LINEA 1 L3-L1	V	Unsigned long
0Eh	2	Voltage of LINE 2 L1-N	Tensione LINEA 2 L1-N	V	Unsigned long
10h	2	Voltage of LINE 2 L2-N	Tensione LINEA 2 L2-N	V	Unsigned long
12h	2	Voltage of LINE 2 L3-N	Tensione LINEA 2 L3-N	V	Unsigned long
14h	2	Voltage of LINE 2 L1-L2	Tensione LINEA 2 L1-L2	V	Unsigned long
16h	2	Voltage of LINE 2 L2-L3	Tensione LINEA 2 L2-L3	V	Unsigned long
18h	2	Voltage of LINE 2 L3-L1	Tensione LINEA 2 L3-L1	V	Unsigned long
1Ah	2	Frequency of LINE 1	Frequenza LINEA 1	Hz/10	Unsigned long
1Ch	2	Frequency of LINE 2	Frequenza LINEA 2	Hz/10	Unsigned long
1Eh	2	Battery voltage (DC power supply)	Tensione batteria (alimentazione DC)	VDC/10	Unsigned long
20h	2	Total operation time	Tempo di lavoro totale	s	Unsigned long
22h	2	LINE 1 ok total time	Tempo totale LINEA 1 ok	s	Unsigned long
24h	2	LINE 2 ok total time	Tempo totale LINEA 2 ok	s	Unsigned long
26h	2	LINE 1 not ok total time	Tempo totale LINEA 1 non ok	s	Unsigned long
28h	2	LINE 2 not ok total time	Tempo totale LINEA 2 non ok	s	Unsigned long
2Ah	2	LINE 1 breaker closed total time	Tempo totale interruttore LINEA 1 chiuso	s	Unsigned long
2Ch	2	LINE 2 breaker closed total time	Tempo totale interruttore LINEA 2 chiuso	s	Unsigned long
2Eh	2	TBreaker opened total time	Tempo totale interruttori aperti	s	Unsigned long
30h	2	(not used)	(non usato)	--	Unsigned long
32h	2	Number of operations of LINE 1 breaker in AUT	Numero commutazioni interruttore 1 in AUT	nr	Unsigned long
34h	2	Number of operations of LINE 2 breaker in AUT	Numero commutazioni interruttore 2 in AUT	nr	Unsigned long
36h	2	Number of operations of LINE 1 breaker in MAN	Numero commutazioni interruttore 1 in MAN	nr	Unsigned long
38h	2	Number of operations of LINE 2 breaker in MAN	Numero commutazioni interruttore 2 in MAN	nr	Unsigned long
3Ah	2	Number of switching alarms of breaker 1	Numero allarmi commutazione interruttore 1	nr	Unsigned long
3Ch	2	Number of switching alarms of breaker 2	Numero allarmi commutazione interruttore 2	nr	Unsigned long
3Eh	2	(not used)	(non usato)	--	Unsigned long
40h	2	Alarms <b>!</b>	Allarmi <b>!</b>	bits	Unsigned long
50h	2	Minimum battery voltage	Tensione batteria minima	V	Unsigned long
52h	2	Maximum battery voltage	Tensione batteria massima	V	Unsigned long
54h	2	Maintenance hours LINE 1	Manutenzione ore LINEA 1	nr	Unsigned long
56h	2	Maintenance hours LINE 2	Manutenzione ore LINEA 2	nr	Unsigned long
58h	2	Operations to the maintenance of the LINE 1	Manovre mancanti manutenzione ore LINEA 1	nr	Signed long
5Ah	2	Operations to the maintenance of the LINE 2	Manovre mancanti manutenzione ore LINEA 2	nr	Signed long
21C0h	1	OR of all limits	OR di tutti i limiti	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
1D00h	2	Counter CNT 1	Contatore CNT 1	UM1	long
1D02h	2	Counter CNT 2	Contatore CNT 2	UM2	long
1D04h	2	Counter CNT 3	Contatore CNT 3	UM3	long
1D06h	2	Counter CNT 4	Contatore CNT 4	UM4	long

❶ Reading the words starting at address 40h will return 32 bits with the following meaning:

❶ Leggendo word all'indirizzo 40h vengono restituiti 32 bit con significato come da tabella:

Bit	Code	Allarme	Codice	Alarm
0	A01	Battery voltage too low	A01	Tensione batteria troppo bassa
1	A02	Battery voltage too high	A02	Tensione batteria troppo alta
2	A03	LINE 1 circuit breaker timeout	A03	Timeout interruttore LINEA 1
3	A04	LINE 2 circuit breaker timeout	A04	Timeout interruttore LINEA 2
4	A05	LINE 1 wrong phase sequence	A05	Errata sequenza fase LINEA 1
5	A06	LINE 2 wrong phase sequence	A06	Errata sequenza fase LINEA 2
6	A07	Timeout load not powered	A07	Timeout carico non alimentato
7	A08	External battery charger failure	A08	Avaria caricabatteria esterno
8	A09	Emergency	A09	Emergenza
9	A10	LINE 1 breaker protection trip	A10	Intervento protezione Interruttore LINEA 1 (trip)
10	A11	LINE 2 breaker protection trip	A11	Intervento protezione Interruttore LINEA 2 (trip)
11	A12	LINE 1 generator not available	A12	Generatore LINEA 1 non disponibile
12	A13	LINE 2 generator not available	A13	Generatore LINEA 2 non disponibile
13	A14	LINE 1 maintenance hours elapsed	A14	Manutenzione ore LINEA 1
14	A15	LINE 2 maintenance hours elapsed	A15	Manutenzione ore LINEA 2
15	A16	LINE 1 maintenance operations	A16	Manutenzione manovre LINEA 1
16	A17	LINE 2 maintenance operations	A17	Manutenzione manovre LINEA 2
17	A18	Auxiliary voltage failure	A18	Allarme tensione ausiliaria
18	UA1	User alarm	UA1	Allarme utente
19	UA2	User alarm	UA2	Allarme utente
20	UA3	User alarm	UA3	Allarme utente
21	UA4	User alarm	UA4	Allarme utente
22 - 31	---	Not used	---	Non usati

TABLE 3:  
STATUS BITS  
(to be used with functions 03 and 04).

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

ADDRESS INDIRIZZO	WORDS	FUNCTION	FUNZIONE	FORMAT FORMATO
2070h	1	Front panel keyboard status ❶	Stato tastiera frontale ❶	Unsigned integer
2100h	1	Digital inputs status (by pin) ❷	Stato ingressi digitali (per pin) ❷	Unsigned integer
2140h	1	Digital outputs status (by pin) ❸	Stato uscite digitali (per pin) ❸	Unsigned integer
--	-	-	-	
2074h	1	LINE 1 voltage status ❹	Stato tensione LINEA 1 ❹	Unsigned integer
2075h	1	LINE 1 breaker status ❺	Stato interruttore LINEA 1 ❺	Unsigned integer
2176h	1	LINE 2 voltage status ❹	Stato tensione LINEA 2 ❹	Unsigned integer
2177h	1	LINE 2 breaker status ❺	Stato interruttore LINEA 2 ❺	Unsigned integer
2078h	2	Input function status ❻	Stato funzioni di ingresso ❻	Unsigned integer
207Ah	1	Output function status ❼	Stato funzioni di uscita ❼	Unsigned integer
207Bh	1	Display messages status ❽	Stato messaggi su display ❽	Unsigned integer
207Ch	1	Controller general status ❾	Stato generale controller ❾	Unsigned integer
207Eh	1	Frontal LED status	Stato LED frontali	Unsigned integer
207Fh	1	Frontal LED status	Stato LED frontali	Unsigned integer

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

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

Bit	Key	Tasto
0	UP	FRECCIA SU
1	OFF/RESET	OFF/RESET
2	MAN	MAN
3	DOWN	FRECCIA GIU'
4	AUT/ENTER	AUT/ENTER
5...15	Not used	Non usati

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

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

Bit	Input	Ingresso
0	Input 1	Ingresso 1
1	Input 2	Ingresso 2
2	Input 3	Ingresso 3
3	Input 4	Ingresso 4
4	Input 5	Ingresso 5
5	Input 6	Ingresso 6
6	Input 7	Ingresso 7
7	Input 8	Ingresso 8
8	Input 9	Ingresso 9
9	Input 10	Ingresso 10
10	Input 11	Ingresso 11
11	Input 12	Ingresso 12
12	Input 13	Ingresso 13
13	Input 14	Ingresso 14
14-15	Not used	Non usati



- ④ Reading address 2072h, the 16 bits have the meaning like the following table:  
 ⑤ Leggendo word all'indirizzo 2072h vengono restituiti 16 bit con significato come da tabella:

Bit	Output	Uscita
0	Output 1	Stato uscita 1
1	Output 2	Stato uscita 2
2	Output 3	Stato uscita 3
3	Output 4	Stato uscita 4
4	Output 5	Stato uscita 5
5	Output 6	Stato uscita 6
6	Output 7	Stato uscita 7
7	Output 8	Stato uscita 8
8	Output 9	Stato uscita 9
9	Output 10	Stato uscita 10
10	Output 11	Stato uscita 11
11	Output 12	Stato uscita 12
12	Output 13	Stato uscita 13
13	Output 14	Stato uscita 14
14	Output 15	Stato uscita 15
15	Not used	Non usato

- ④ Reading address 2075h (LINE 1) or 2177h (LINE 2), the 16 bits have the meaning like the following table:  
 ⑤ Leggendo word all'indirizzo 2075h (LINEA 1) o 2177h (LINEA 2) vengono restituiti 16 bit con significato come da tabella:

Bit	Breaker status	Stato interruttore
0	Breaker closed	Interruttore chiuso
1	Trip alarm	Allarme Trip
2	Not used	Non usato
3	Command status (1 = close)	Stato comandato (1= chiuso)
4	Close command output	Uscita comando chiusura
5	Open command output	Uscita comando apertura
6...15	Not used	Non usati

- ④ Reading address 207Ah, the 16 bits have the meaning like the following table:  
 ⑤ Leggendo word all'indirizzo 207Ah vengono restituiti 16 bit con significato come da tabella:

Bit	Output functions status	Stato funzioni uscita
0	LINE 1 breaker open	Apertura LINEA 1
1	LINE 2 breaker open	Apertura LINEA 2
2	LINE 1 breaker close	Chiusura LINEA 1
3	LINE 2 breaker close	Chiusura LINEA 2
4	Global alarm	Allarme globale
5	Generator 1 start	Start generatore 1
6	Generator 2 start	Start generatore 2
7	ATS ready	ATS pronto
8	Load shed	Load shed
9	Not used	Non usato
10	Not used	Non usato
11	Open all	Apri tutto
12	Undervoltage coil 1	Bobina minima 1
13	Undervoltage coil 2	Bobina minima 2
14	LINE 1 OK	LINEA 1 OK
15	LINE 2 OK	LINEA 2 OK

- ④ Following table shows meaning of bits of the word at address 2074h (LINE 1) or 2176h (LINE 2):  
 ⑤ Leggendo word all'indirizzo 2074h (LINEA 1) o 2176h (LINEA 2) vengono restituiti 16 bit con significato come da tabella:

Bit	Line status	Stato linea
0	Line values into limits	Linea nei limiti
1	Line values into limits delayed	Linea nei limiti + ritardo
2	Voltage into limits	Tensione nei limiti
3	Voltage ok	Tensione ok
4	Frequency into limits	Frequenza nei limiti
5	Frequency ok	Frequenza ok
6	Voltage below min	Tensione < minima
7	Voltage above max	Tensione > massima
8	Voltage asymmetry	Tensioni fuori soglia asimmetria
9	Voltage phase loss	Tensione < soglia mancanza fase
10	Frequency below min	Frequenza < minima
11	Frequency above max	Frequenza > massima
12	Wrong phase sequence	Sequenza fasi errata
13	All line parameters ok	Tutti i parametri linea ok
14-15	Not used	Non usato

- ④ Reading address 2178h, the 16 bits have the meaning like the following table:  
 ⑤ Leggendo word all'indirizzo 2178h vengono restituiti 16 bit con significato come da tabella:

Bit	Input functions status	Stato funzioni ingresso
0	LINE 1 breaker closed feedback	Interruttore LINEA 1 chiuso
1	LINE 1 breaker trip	Interruttore LINEA 1 Trip
2	Not used	Non usato
3	LINE 2 breaker closed feedback	Interruttore LINEA 2 chiuso
4	LINE 2 breaker trip	Interruttore LINEA 2 Trip
5	Not used	Non usato
6	Transfer to secondary LINE	Forzatura su LINEA secondaria
7	Inhibit return to main LINE	Inibizione ritorno su LINEA principale
8	Emergency pushbutton	Pulsante emergenza
9	Generator start	Start generatore
10	Generator 1 ready	Generatore 1 pronto
11	Generator 2 ready	Generatore 2 pronto
12	Keyboard locked	Blocco tastiera
13	Lock parameters	Blocco programmazione
14	Not used	Non usato
15	Alarms inhibited	Allarmi inibiti

- ④ Reading address 207Bh, the 16 bits have the meaning like the following table:  
 ⑤ Leggendo word all'indirizzo 207Bh vengono restituiti 16 bit con significato come da seguente tabella:

Bit	Display message status	Stato messaggi display
0	Generator 1 start	Start generatore 1
1	Generator 2 start	Start generatore 2
2	Generator 1 cooling	Raffreddamento generatore 1
3	Generator 2 cooling	Raffreddamento generatore 2
4	Load transfer 2 → 1	Trasferimento carico 2 → 1
5	Load transfer 1 → 2	Trasferimento carico 1 → 2

- ④ Reading address 207Ch, the 16 bits have the meaning like the following table:  
 ⑤ Leggendo word all'indirizzo 207Ch vengono restituiti 16 bit con significato come da seguente tabella:

Bit	Output functions status	Stato funzioni uscita
0	Operative mode OFF / Reset	Modo operativo OFF / Reset
1	Operative mode MAN	Modo operativo MAN
2	Operative mode AUT	Modo operativo AUT
3	Operative mode TEST	Modo operativo TEST
4	Error on	In errore
5	AC power supply present	Alimentazione AC presente
6	DC power supply present	Alimentazione DC presente
7	Global alarm on	Allarme globale attivato
8...15	Not used	Non usati

TABLE 4:  
COMMANDS  
(to be used with function 06).

ADDRESS INDIRIZZO	WORDS	STATUS	STATO
4F00h	1	Set remote variable REM1❶	Imposta variabile remora REM1❶
4F01h	1	Set remote variable REM2	Imposta variabile remora REM2
.....			
4F07h	1	Set remote variable REM8	Imposta variabile remora REM8
2F00h		Operative mode change❷	Cambio modalità operativa❷
2FOAh	1	Front panel keystore simulation❸	Simulazione pressione tasti pannello frontale❸
2F03h	1	Value 01h: E <sup>2</sup> prom save	Valore 01h: salvataggio E <sup>2</sup> prom
		Value 04h: reboot	Valore 04h: reboot
		Value 05h: save and reboot	Valore 05h: salvataggio e reboot
2F07h	1	Value 00h: reset device	Valore 00h: reset apparecchio
		Value 01h: reset device and save fram	Valore 01h: reset apparecchio con salvataggio in fram
2FF0h	1	Command menu execution❹	Esecuzione comando menu comandi❹
28FAh	1	Value 01h: save real time clock setting	Valore 01h: salvataggio impostazione orologio datario
1C02h	1	Value 100h: start generator 1	Valore 100h: start generatore 1
		Value 01h: stop generator 1	Valore 01h: stop generatore 1
		Value 200h: start generator 2	Valore 200h: start generatore 2
		Value 02h: stop generator 2	Valore 02h: stop generatore 2
1C00h	1	Value 100h: BREAKER 1 close	Valore 100h: chiudere BREAKER 1
		Value 01h: BREAKER 1 open	Valore 01h: aprire BREAKER 1
		Value 200h: BREAKER 2 close	Valore 200h: chiudere BREAKER 2
		Value 02h: BREAKER 2 open	Valore 02h: aprire BREAKER 2

- ❶ Writing AAh to the indicated address the remote variable will be set to 1, writing BBh the remote variable will be set to 0:  
 ❷ The following table shows the values to be written to address 2F00h to achieve the correspondent function:

VALUE VALORE	FUNCTION	FUNZIONE
0	Switch to OFF mode	Passaggio a modalità OFF
1	Switch to MAN mode	Passaggio a modalità MAN
2	Switch to AUT mode	Passaggio a modalità AUT

- ❸ The following table shows the value to be written to address 2FOAh to achieve the correspondent function.  
 ❹ La seguente tabella indica il valore da scrivere all'indirizzo 2FOAh per ottenere le corrispondenti funzioni.

VALUE VALORE	MEANING	SIGNIFICATO
0x8001	Key UP	Tasto UP
0x8004	Key RIGHT	Tasto RIGHT
0x8200	Key DOWN	Tasto DOWN
0x8400	Key ENTER	Tasto ENTER
0x8800	Key LEFT	Tasto LEFT
0x0002	MAN mode	Modalità MAN
0x0020	OFF mode	Modalità OFF
0x0040	AUT mode	Modalità AUT

TABELLA 4:  
COMANDI  
(utilizzabili con funzione 06).

ADDRESS INDIRIZZO	WORDS	STATUS	STATO
4F00h	1	Set remote variable REM1❶	Imposta variabile remora REM1❶
4F01h	1	Set remote variable REM2	Imposta variabile remora REM2
.....			
4F07h	1	Set remote variable REM8	Imposta variabile remora REM8
2F00h		Operative mode change❷	Cambio modalità operativa❷
2FOAh	1	Front panel keystore simulation❸	Simulazione pressione tasti pannello frontale❸
2F03h	1	Value 01h: E <sup>2</sup> prom save	Valore 01h: salvataggio E <sup>2</sup> prom
		Value 04h: reboot	Valore 04h: reboot
		Value 05h: save and reboot	Valore 05h: salvataggio e reboot
2F07h	1	Value 00h: reset device	Valore 00h: reset apparecchio
		Value 01h: reset device and save fram	Valore 01h: reset apparecchio con salvataggio in fram
2FF0h	1	Command menu execution❹	Esecuzione comando menu comandi❹
28FAh	1	Value 01h: save real time clock setting	Valore 01h: salvataggio impostazione orologio datario
1C02h	1	Value 100h: start generator 1	Valore 100h: start generatore 1
		Value 01h: stop generator 1	Valore 01h: stop generatore 1
		Value 200h: start generator 2	Valore 200h: start generatore 2
		Value 02h: stop generator 2	Valore 02h: stop generatore 2
1C00h	1	Value 100h: BREAKER 1 close	Valore 100h: chiudere BREAKER 1
		Value 01h: BREAKER 1 open	Valore 01h: aprire BREAKER 1
		Value 200h: BREAKER 2 close	Valore 200h: chiudere BREAKER 2
		Value 02h: BREAKER 2 open	Valore 02h: aprire BREAKER 2

- ❶ Scrivendo il valore AAh all'indirizzo indicato viene impostata la variabile remota a 1, scrivendo BBh viene impostata a 0.  
 ❷ La seguente tabella indica i valori da scrivere all'indirizzo 2F00h per ottenere le corrispondenti funzioni.

- ❸ Writing value between 0 and 15 to the indicated address, the correspondent command will be executed.  
 ❹ Scrivendo il valore da 0 a 15 all'indirizzo indicato viene eseguito la corrispondente funzione.

	MEANING	SIGNIFICATO
0	Reset maintenance 1	Reset manutenzione ore 1
1	Reset maintenance 2	Reset manutenzione ore 2
2	Reset maintenance operations 1	Reset manutenzione manovre 1
3	Reset maintenance operations 2	Reset manutenzione manovre 2
4	Reset generic counters CNTx	Reset contatori generici CNTx
5	Reset LIMx limits	Reset stato limiti LIMx
6	Reset hours counter LINE 1/LINE 2	Reset conta ore LINEA 1/ LINEA 2
7	Reset conta ore BREAKER 1/ BREAKER 2	Reset hours counter BREAKER 1/ BREAKER 2
8	Reset breaker operation	Reset manovre interruttori
9	Reset events list	Reset lista eventi
10	Reset default parameters	Ripristino parametri a default
11	Save parameters in backup memory	Salva parametri nella memoria backup
12	Reload parameters from backup memory	Ricarica parametri dalla memoria backup
14	Reset A03 – A04 alarms	Reset allarmi A03 – A04
15	Simulate line failure	Simula mancanza linea prioritaria

TABLE 5:  
DEVICE GLOBAL STATUS  
(to be used with function 03 and 04).

ADDRESS INDIRIZZO	WORDS	STATUS	STATI	FORMAT FORMATO
2210h	2	Device global status (bit 0-bit 31)Ⓜ	Stato globale dispositivo (bit 0-bit 31)Ⓜ	Unsigne integer

Ⓜ Reading two words at address 2210h will return 32 bits with the following meaning:

Ⓜ Leggendo 2 word agli indirizzi 2210h vengono restituiti 32 Bit con significato come da tabella:

Bit	MEANING	SIGNIFICATO
Bit 0	Device OFF	Dispositivo in OFF
Bit 1	Device in MAN mode	Dispositivo in MAN
Bit 2	Device in AUT mode	Dispositivo in AUT
Bit 3	Device TEST mode	Dispositivo in TEST
Bit 4	Voltage LINE 1 OK	Tensione di LINEA 1 OK
Bit 5	Voltage LINE 2 OK	Tensione di LINEA 2 OK
Bit 6	LED LINE 1 on	LED LINEA 1 acceso
Bit 7	LED LINE 2 on	LED LINEA 2 acceso
Bit 8	LED LINE 1 on the load	LED LINEA 1 sul carico
Bit 9	LED LINE 1 on the load	LED LINEA 1 sul carico
Bit 10	Mains contactor closed	Teleruttore rete chiuso
Bit 11	Generator contactor closed	Teleruttore generatore chiuso
Bit 12	Global alarm	Allarme globale
Bit 13	AC power supply	Alimentazione AC
Bit 14	Start Generator 1	Start Generatore 1
Bit 15	Start Generator 2	Start Generatore 2
Bit 16	LINE 1 max Volt	LINEA 1 max Volt
Bit 17	LINE 1 min Volt	LINEA 1 min Volt
Bit 18	LINE 1 max Hz	LINEA 1 max Hz
Bit 19	LINE 1 min Hz	LINEA 1 min Hz
Bit 20	LINE 1 phase lost	LINEA 1 perdita di fase
Bit 21	LINE 1 asym	LINEA 1 asym
Bit 22	LINE 2 max Volt	LINEA 2 max Volt
Bit 23	LINE 2 min Volt	LINEA 2 min Volt
Bit 24	LINE 2 Hz max	LINEA 2 max Hz
Bit 25	LINE 2 min Hz	LINEA 2 min Hz
Bit 26	LINE 2 phase lost	LINEA 2 perdita di fase
Bit 27	LINE 2 asym	LINEA 2 asym
Bit 28	(not used)	(non usato)
...	.....	.....
Bit 31	(not used)	(non usato)

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

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.

ADDRESS INDIRIZZO	WORDS	FUNZIONE	FUNCTION	RANGE
28F0h	1	Year	Anno	2000...2099
28F1h	1	Month	Mese	1-12
28F2h	1	Day	Giorno	1-31
28F3h	1	Hours	Ora	0-23
28F4h	1	Minutes	Minuti	0-59
28F5h	1	Seconds	Secondi	0-59

TABELLA 6:  
OROLOGIO DATARIO  
(utilizzabili con funzioni 04 e 06).

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

## 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 perform 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 ATL610 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 3, if you want to read any other event do step 2.

## 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
----	----	----	----	----	----	----	----

ATL 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
----	----	----	----	----	----	----	----

ATL 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
----	----	----	----	----	----	----	----

ATL 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															

## 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 ATL610 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 ❷.
3. Write the value of the parameter that you want to read by using the function 6 at address 5002h ❸.
4. Perform the function 4 at the address 5004h, with a number of registers appropriate to the length of the parameter (see table "Type of parameter").
5. If you want to read the next parameter (in the same menu/submenu) repeat step 4, otherwise perform step 1.

❶ It's possible to read the menu, submenus, and parameter stored at the addresses 5000h, 5001h and 5002h by using the function 4.

## 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 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 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'ATL610. 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
----	----	----	----	----	----	----	----

ATL Funzione = 4  
Nr. bytes. = 1 (01h)  
MSB = 100 (64h)  
LSB = 2 (02h)

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

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
----	----	----	----	----	----	----	----

ATL 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
----	----	----	----	----	----	----	----

ATL 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

## 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 dell'ATL610.

## 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 ❷.
3. Scrivere il valore del parametro che si vuole leggere tramite la funzione 6 all'indirizzo 5002h ❸.
4. Eseguire la funzione 4 all'indirizzo 5004h, di un numero di registri appropriato alla lunghezza del parametro (vedi tabella "Tipo di parametro").
5. Se si vuole leggere il parametro successivo, (all'interno dello stesso menu/sottomenu) ripetere il passo 4, altrimenti eseguire il passo 1.

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

## 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 E<sup>2</sup>PROM, using the dedicated command described in Table 4 (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 register (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
----	----	----	----	----	----	----	----

ALT 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
----	----	----	----	----	----	----	----

ALT 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
----	----	----	----	----	----	----	----

ALT Function = 6  
Address = 5002h (5002h - 0001h = 5001h)  
Value = 1 (01h)

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

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
----	----	----	----	----	----	----	----	----	----	----

ALT 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
----	----	----	----	----	----	----	----

ALT No answer.

## 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 E<sup>2</sup>PROM, 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 registro (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 È possibile leggere il valore del menu, sottomenu e parametro memorizzati agli indirizzi 5000h, 5001h e 5002h utilizzando la funzione 4.

## ESEMPIO

Impostare a 8 il valore del parametro M08.01.01

Passo 1: impostazione menu 08.

MASTER Funzione = 6  
Indirizzo = 5000h (5000h - 0001h = 4FFFh)  
Valore = 8 (08h)

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

ALT 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
----	----	----	----	----	----	----	----

ALT 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
----	----	----	----	----	----	----	----

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

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
----	----	----	----	----	----	----	----	----	----	----

ALT Funzione = 16 (10h)  
Indirizzo = 5004h (5004h - 0001h = 5003h)  
Valore = 2 (02h)

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

Passo 5: salvataggio e riavvio.

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

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

ALT Nessuna risposta.