

DESCRIZIONE SOFTWARE ALB (ABACO® LINK BUS)

Il protocollo di comunicazione **ALB**, permette di sfruttare tutte le risorse della scheda tramite una serie di comandi sulla linea seriale; per questo motivo tutto quanto ricevuto viene interpretato ed eseguito, ed eventuali codici di risposta vengono restituiti all'unità master di controllo. Il firmware prevede inoltre, una modalità di funzionamento Setup, che permette di configurare, nel modo desiderato, tutte le varie sezioni del dispositivo.

SELEZIONE DELLA MODALITA' DI FUNZIONAMENTO

Il firmware di bordo della **GPC® R/T94** è in grado di gestire due modalità distinte, in particolare può essere attivato il **modo SETUP**, oppure il **modo RUN (ALB)**. La selezione dell'uno o dell'altro funzionamento, avviene durante la fase di **Power-ON**, verificando lo stato logico in cui si trova il jumper **J1 (RUN/DEBUG)** in particolare:

J1 = **CONNESSO** -> *Modo SETUP*
J1 = **NON CONNESSO** -> *Modo RUN*

MODALITA' DI FUNZIONAMENTO SETUP

Tramite il modo **SETUP** è possibile definire tutti i vari parametri d'inizializzazione, relativi al settaggio del Baud Rate, del Tipo di comunicazione e del Nome del dispositivo. Tali dati saranno memorizzati all'interno della EEPROM, e permetteranno di definire la configurazione di lavoro nel modo RUN. Per una corretta programmazione dei vari parametri seguire i seguenti punti:

- 1) Chiudere il jumper **J1** ed alimentare la scheda.
- 2) Se la modalità SETUP è stata riconosciuta viene attivata l'uscita OUT0 (LD1 acceso). A questo punto l'utente deve impostare il BAUD RATE ed il TIPO DI COMUNICAZIONE utilizzando gli ingressi IN0÷IN7 seguendo le indicazioni delle seguenti tabelle:

BAUD RATE	IN0	IN1	IN2	IN3	IN4	IN5
38400	OFF	OFF	OFF	OFF	OFF	OFF
19200	ON	OFF	OFF	OFF	OFF	OFF
9600	OFF	ON	OFF	OFF	OFF	OFF
4800	OFF	OFF	ON	OFF	OFF	OFF
2400	OFF	OFF	OFF	ON	OFF	OFF
1200	OFF	OFF	OFF	OFF	ON	OFF

TIPO DI COMUNICAZIONE	IN6	IN7
Punto-Punto	OFF	OFF
Master-Slave 9 bits	ON	OFF

FIGURA 37: TABELLE SETTAGGIO BAUD RATE E TIPO COMUNICAZIONE

Per confermare e salvare i valori selezionati, l'utente deve chiudere (ON) e successivamente riaprire (OFF) l'ingresso AUX (P3.2).

- 3) Se l'operazione precedente non ha avuto successo viene attivato anche OUT3 (LD4 acceso) e l'utente deve ripetere il punto 2. Al contrario se l'operazione ha avuto successo il firmware attiva l'uscita OUT1 (LD2 acceso), indicando che é possibile inserire il NOME della scheda utilizzato nella comunicazione seriale. Per l'inserimento del NOME (valori settabili=128÷255) si deve utilizzare la stessa tecnica del punto 2 ossia si devono settare opportunamente le linee di input IN0÷IN7 e si deve confermare il dato dando un impulso sull'ingresso AUX. Di seguito viene riportata una tabella che serve per chiarire come settare il NOME in modalità binaria:

NOME	IN0	IN1	IN2	IN3	IN4	IN5	IN6	IN7
128	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
129	ON	OFF	OFF	OFF	OFF	OFF	OFF	ON
...
255	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

FIGURA 38: TABELLE SETTAGGIO NOME

- 4) Se l'operazione precedente non ha avuto successo viene attivato anche OUT3 (LD4 acceso) e l'utente deve ripetere il punto 3. Al contrario se l'operazione ha avuto successo il firmware salva tutti i parametri in EEPROM ed entra in un loop infinito. Se l'operazione precedente non ha avuto successo vengono attivate tutte le uscite OUT0÷OUT3 (LD1÷LD4 accessi) al contrario se l'operazione ha avuto successo il firmware setta le uscite nel seguente modo per dichiarare la fine del SETUP:

OUT0	=	Attiva	(LD1 acceso)
OUT1	=	Disattiva	(LD2 spento)
OUT2	=	Attiva	(LD3 acceso)
OUT3	=	Disattiva	(LD4 spento)

- 5) Togliere l'alimentazione alla scheda.
- 6) Aprire il jumper J1 ed alimentare la scheda (RUN mode attivo = LD1÷LD4 spenti).

Da notare che un ingresso optoisolato NPN é attivo (ON - LED Attivo), quando il relativo contatto d'ingresso é chiuso verso la linea di massa della tensione di alimentazione degli optoisolatori (GND opto).

MODALITA' DI FUNZIONAMENTO RUN

Quando si accede al modo **RUN**, vengono verificati i parametri memorizzati nell'EEPROM, relativi al protocollo di comunicazione. Se questi non sono validi (EEPROM non inizializzata) la scheda entra in un loop di attesa; a questo punto é possibile solo togliere alimentazione dal dispositivo.

N.B.: Di default l' EEPROM non é inizializzata perciò l' utente deve provvedere ad **inizializzarla (SETUP mode) prima di provare a gestire la scheda.**

Il protocollo di comunicazione del modo **RUN** é settabile in termini di Baud Rate e Tipo di Comunicazione (mediante la modalit  SETUP), mentre il formato dei dati é in funzione della modalit  selezionata, in particolare:

Comunicazione Punto-Punto: **8 bit, 1 Stop, NO Parity**
Master-Slave: **9 bit, 1 Stop, NO Parity**

I comandi riconosciuti nella modalit  di funzionamento **RUN**, sono descritti di seguito.

COMANDI VARI

RESET GENERALE

Sequenza di Input:

<i>Codice Dec:</i>	65	97
<i>Codice Hex:</i>	41	61
<i>Mnemonico:</i>	A	a

A seguito di questo comando, il firmware ripristina la condizione di inizializzazione che si verifica all'atto del Power-ON; in particolare:

OUT0÷OUT3: Vengono resettati e posti nella condizione iniziale, quindi settati allo stato logico 0. Eventuali temporizzazioni vengono interrotte.

Counter a 16 Bit: Viene inizializzato con il valore 0.

SCRITTURA DEL BYTE DI PRESENZA

Sequenza di Input:

<i>Codice Dec:</i>	89	<NibL VAL>	<NibH VAL>
<i>Codice Hex:</i>	59	<NibL VAL>	<NibH VAL>
<i>Mnemonico:</i>	Y	ASCII(NibL VAL)	ASCII(NibH VAL)

Il dato "**VAL**" (0÷255) viene salvato nella parte non accessibile all' utente della EEPROM (0÷31). Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Esempio:

Se si desidera memorizzare, il dato di presenza 65 sar  necessario inviare la seguente sequenza:
89 1 4.

LETTURA DAL BYTE DI PRESENZA

Sequenza di Input:

Codice Dec: **121**

Codice Hex: **79**

Mnemonico: **y**

Codici di Risposta:

Viene restituito un dato (0÷255), tramite i due nibble che lo compongono secondo, il formato visto per il comando precedente: <NibL VAL>, <NibH VAL>.

LETTURA DELLA VERSIONE DEL FIRMWARE

Sequenza di Input:

Codice Dec: **86**

Codice Hex: **56**

Mnemonico: **V**

Codici di Risposta:

Vengono restituiti due nibble che compongono il numero di versione del firmware:

Versione X.Y= Y --> <NibL VAL>
X --> <NibH VAL>

COMANDI RELATIVI ALLA GESTIONE A PORT DELL' I/O DIGITALE

SETTAGGIO DEL PORT DI OUTPUT

Sequenza di Input:

<i>Codice Dec:</i>	87	1	<Dato>	0
<i>Codice Hex:</i>	57	1	<Dato>	0
<i>Mnemonico:</i>	W	SOH	ASCII(Dato)	NUL

Il byte <Dato> deve essere inviato secondo il seguente formato:

(MSB) 0 0 0 0 **OUT3 OUT2 OUT1 OUT0** (LSB)

Dove **OUTn** indica lo stato logico, **0** o **1**, che devono assumere le relative uscite.
Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:

Se si vogliono attivare le uscite OUT0 e OUT3 sarà necessario inviare la seguente sequenza: **87 1 9 0**.

LETTURA DEL PORT DI INPUT

Sequenza di Input:

<i>Codice Dec:</i>	82	0
<i>Codice Hex:</i>	52	0
<i>Mnemonico:</i>	R	NUL

Viene restituito in seriale il dato letto dal Port di INPUT.

Codici di Risposta:

Il dato acquisito dal Port viene restituito sotto forma di nibble secondo il seguente formato:

Nibble L:	(MSB)	0	0	0	0	IN3	IN2	IN1	IN0	(LSB)
Nibble H:		0	0	0	0	IN7	IN6	IN5	IN4	

Dove i bit **INn** di tali bytes, indicano gli stati logici **0** o **1**, in cui si trovano le relative linee di INPUT.
Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:

Se si vuole leggere il Port di INPUT, sul quale é presente il dato **90** (5A Hex) sarà necessario inviare la seguente sequenza: **82 0**, e come risposta si avranno i seguenti bytes: **10 5**.

COMANDI RELATIVI ALLA GESTIONE A BIT DELL' I/O DIGITALE

SETTAGGIO DI UN BIT DEL PORT DI OUTPUT

Sequenza di Input:

<i>Codice Dec:</i>	83	1	<Bit>
<i>Codice Hex:</i>	53	1	<Bit>
<i>Mnemonico:</i>	S	SOH	ASCII(Bit)

Viene posto allo stato logico 1, l' uscita indicata da **<Bit>** che può assumere i valori **0, 1, 2 e 3**. Se sulla linea di output in questione sono in corso delle temporizzazioni, queste vengono interrotte. Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:

Se si vuole settare la linea **OUT2** é necessario inviare la seguente sequenza: **83 1 2**.

SETTAGGIO TEMPORIZZATO DI UN BIT DEL PORT DI OUTPUT

Sequenza di Input:

<i>Codice Dec:</i>	115	1	<Bit>	<NibL>	<NibH>
<i>Codice Hex:</i>	73	1	<Bit>	<NibL>	<NibH>
<i>Mnemonico:</i>	s	SOH	ASCII(Bit)	ASCII(NibL)	ASCII(NibH)

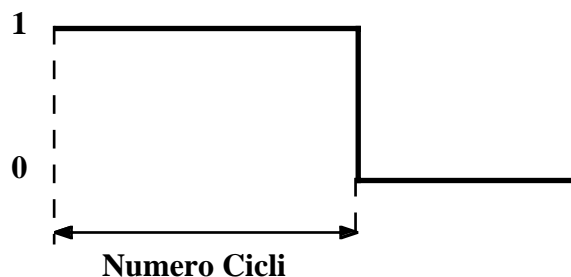


FIGURA 39: COMANDO DI SET TEMPORIZZATO

Viene posto allo stato logico 1, per un determinato tempo l' uscita indicata da **<Bit>** che può assumere i valori **0, 1, 2 e 3**. L' uscita selezionata rimane allo stato logico 1 per un tempo determinato dai bytes **<Nib>**, quindi ritorna allo stato logico 0. Il valore di temporizzazione deve essere compreso nel range **1÷255**, dove 1 ciclo corrisponde a 10 msec e deve essere inviato a nibble secondo il seguente formato:

Nibble L:	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
Nibble H:		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4

Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:

Se si vuole settare la linea **OUT2** per un tempo di 500 msec, corrispondente a **50** cicli é necessario inviare la seguente sequenza: **115 1 2 2 3**.

CLEAR DI UN BIT DEL PORT DI OUTPUT

Sequenza di Input:

<i>Codice Dec:</i>	67	1	<Bit>
<i>Codice Hex:</i>	43	1	<Bit>
<i>Mnemonico:</i>	C	SOH	ASCII(Bit)

Viene posto allo stato logico 0, l'uscita indicata da <Bit> che può assumere i valori **0, 1, 2 e 3**. Se sulla linea di output in questione sono in corso delle temporizzazioni, queste vengono interrotte. Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:

Se si vuole resettare la linea **OUT2** é necessario inviare la seguente sequenza: **67 1 2**.

CLEAR TEMPORIZZATO DI UN BIT DEL PORT DI OUTPUT

Sequenza di Input:

<i>Codice Dec:</i>	99	1	<Bit>	<NibL>	<NibH>
<i>Codice Hex:</i>	63	1	<Bit>	<NibL>	<NibH>
<i>Mnemonico:</i>	c	SOH	ASCII(Bit)	ASCII(NibL)	ASCII(NibH)

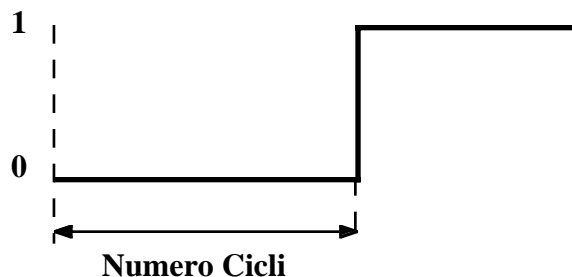


FIGURA 40: COMANDO DI CLEAR TEMPORIZZATO

Viene posto allo stato logico 0, per un determinato tempo l'uscita indicata da <Bit> che può assumere i valori **0, 1, 2 e 3**. L'uscita selezionata rimane allo stato logico 0 per un tempo determinato dai bytes <Nib>, quindi ritorna allo stato logico 1. Il valore di temporizzazione deve essere compreso nel range **1÷255**, dove 1 ciclo corrisponde a 10 msec e deve essere inviato a nibble secondo il seguente formato:

Nibble L:	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0	(LSB)
Nibble H:		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	

Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:

Se si vuole resettare la linea **OUT2** per un tempo di 500 msec, corrispondente a **50** cicli é necessario inviare la seguente sequenza: **99 1 2 2 3**.

ONDA QUADRA SU UN BIT DEL PORT DI OUTPUT

Sequenza di Input:

<i>Codice Dec:</i>	80	1	<Bit>	<NibL>	<NibH>
<i>Codice Hex:</i>	50	1	<Bit>	<NibL>	<NibH>
<i>Mnemonico:</i>	P	SOH	ASCII(Bit)	ASCII(NibL)	ASCII(NibH)

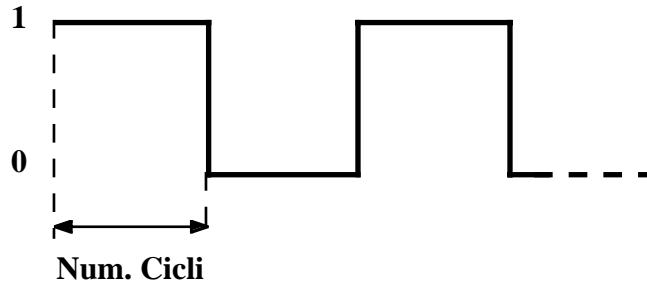


FIGURA 41: COMANDO DI ONDA QUADRA

Viene generata un' onda quadra con Duty Cycle del 50%, sull' uscita indicata da <Bit> che può assumere i valori **0, 1, 2 e 3**. Il periodo del segnale generato, viene determinato dal valore dei bytes <Nib>, che esprime il tempo di durata di una semi-onda. Il valore deve essere compreso nel range **1÷255**, dove 1 ciclo corrisponde a 10 msec e deve essere inviato a nibble secondo il seguente formato:

Nibble L:	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0	(LSB)
Nibble H:		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	

Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:

Se si vuole resettare la linea **OUT2** per un tempo di 200 msec, corrispondente a **20** cicli é necessario inviare la seguente sequenza: **99 1 2 4 1**.

ONDA QUADRA TEMPORIZZATA DA "1" SU UN BIT DEL PORT DI OUTPUT

Sequenza di Input:

<i>Codice Dec:</i>	112 1	<Bit>	<NibL>	<NibH>	<StaL>	<StaH>
<i>Codice Hex:</i>	70 1	<Bit>	<NibL>	<NibH>	<StaL>	<StaH>
<i>Mnemonico:</i>	p SOH	ASCII(Bit)	ASCII(NibL)	ASCII(NibH)	ASCII(StaL)	ASCII(StaH)

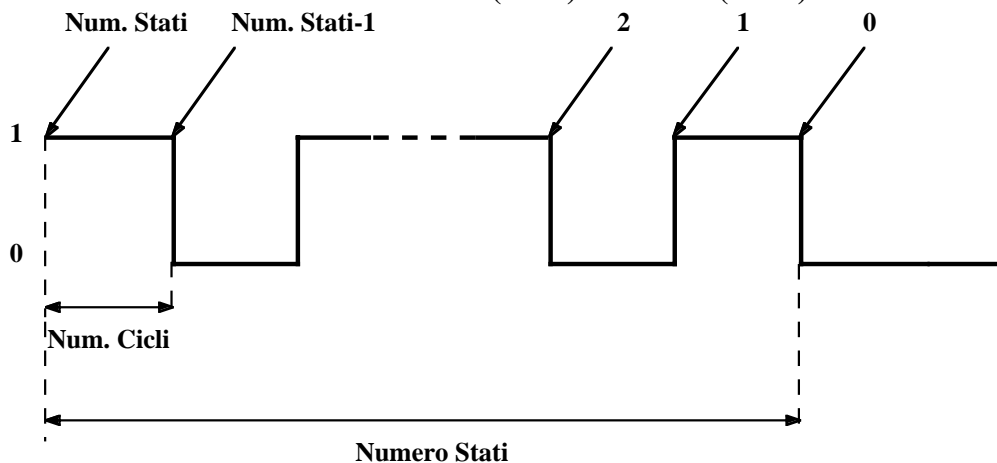


FIGURA 42: COMANDO DI ONDA QUADRA TEMPORIZZATA

Viene generata un' onda quadra, **con stato logico iniziale 1**, per un determinato tempo con Duty Cycle del 50%, sull'uscita indicata da **<Bit>** che può assumere i valori **0, 1, 2 e 3**. Il periodo del segnale generato, viene determinato dal valore dei bytes **<Nib>**, che esprime il tempo di durata di una semi-onda. Il valore deve essere compreso nel range **1÷255**, dove 1 ciclo corrisponde a 10 msec e deve essere inviato a nibble secondo il seguente formato:

Nibble L:	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0	(LSB)
Nibble H:		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	

Il tempo di permanenza di tale segnale sull' uscita selezionata é determinato dal byte **<Sta>**, che deve essere inviato anch'esso a nibble secondo il formato indicato in precedenza, e deve essere compreso nel range **1÷255**. Mediante questo valore é infatti possibile indicare il numero di cambiamenti di stato che devono avvenire sull' uscita indicata, prima che quest'ultima ritorni definitivamente allo stato logico 0; come si può notare dalla Figura, il numero di cambiamenti di stato é: **<Sta>+1**.
Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:

Se si vuole resettare la linea **OUT2** per un tempo di 200 msec, corrispondente a **20** cicli e con **10** cambiamenti di stato, é necessario inviare la seguente sequenza: **112 1 2 4 1 9 0**.

ONDA QUADRA TEMPORIZZATA DA "0" SU UN BIT DEL PORT DI OUTPUT

Sequenza di Input:

<i>Codice Dec:</i>	119 1	<Bit>	<NibL>	<NibH>	<StaL>	<StaH>
<i>Codice Hex:</i>	77 1	<Bit>	<NibL>	<NibH>	<StaL>	<StaH>
<i>Mnemonico:</i>	w SOH	ASCII(Bit)	ASCII(NibL)	ASCII(NibH)	ASCII(StaL)	ASCII(StaH)

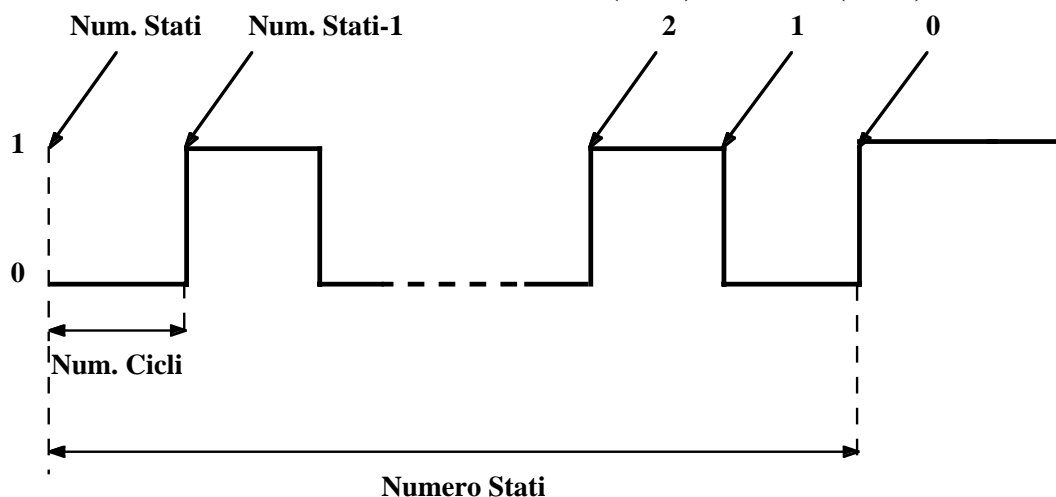


FIGURA 43: COMANDO DI ONDA QUADRA TEMPORIZZATA

Viene generata un' onda quadra, **con stato logico iniziale 0**, per un determinato tempo con Duty Cycle del 50%, sull'uscita indicata da **<Bit>** che può assumere i valori **0, 1, 2 e 3**. Il periodo del segnale generato, viene determinato dal valore dei bytes **<Nib>**, che esprime il tempo di durata di una semi-onda. Il valore deve essere compreso nel range **1÷255**, dove 1 ciclo corrisponde a 10 msec e deve essere inviato a nibble secondo il seguente formato:

Nibble L:	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0	(LSB)
Nibble H:		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	

Il tempo di permanenza di tale segnale sull'uscita selezionata é determinato dal byte <Sta>, che deve essere inviato anch'esso a nibble secondo il formato indicato in precedenza, e deve essere compreso nel range **1÷255**. Mediante questo valore é infatti possibile indicare il numero di cambiamenti di stato che devono avvenire sull'uscita indicata, prima che quest'ultima ritorni definitivamente allo stato logico 1; come si può notare dalla Figura, il numero di cambiamenti di stato é: <Sta>+1. Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:

Se si vuole resettare la linea **OUT2** per un tempo di 200 msec, corrispondente a **20** cicli e con **10** cambiamenti di stato, é necessario inviare la seguente sequenza: **119 1 2 4 1 9 0**.

LETTURA DELL' INPUT AUX O DI UN BIT DEL PORT DI INPUT

Sequenza di Input:

<i>Codice Dec:</i>	114	<Port>	<Bit>
<i>Codice Hex:</i>	72	<Port>	<Bit>
<i>Mnemonic:</i>	r	ASCII(Port)	ASCII(Bit)

Viene restituito lo stato logico, 0 o 1, dell'input indicato da <Bit> relativo al Port indicato in <Port>. Questi bytes possono avere i seguenti valori: 0÷7, per quel che riguarda il numero del Bit, mentre i valori validi per il numero del Port sono 0 (PORT di INPUT) o 3 (ingresso AUX=Bit 0, gli altri bits non hanno alcun significato ed il comando viene ignorato).

Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Codici di Riposta:

Viene restituito lo stato dell' INPUT richiesto, in particolare:

- 0 La linea di input si trova allo stato logico 0
- 1 La linea di input si trova allo stato logico 1

Esempio:

Se si vuole leggere lo stato della linea AUX é necessario inviare la seguente sequenza: **114 3 0** e sarà restituito il valore 1 o 0.

LETTURA DELLA CONFIGURAZIONE DEL PIN AUX

Sequenza di Input:

<i>Codice Dec:</i>	102
<i>Codice Hex:</i>	66
<i>Mnemonic:</i>	f

Codici di risposta:

Il byte di configurazione del pin AUX, che viene restituito dal firmware può avere i seguenti valori:

- 0** Pin AUX settato come linea di INPUT
- 2** Pin AUX settato come Trigger per il Counter a 16 Bits

SETTAGGIO DELLA CONFIGURAZIONE DEL PIN AUX

Sequenza di Input:

<i>Codice Dec:</i>	85	<Byte>
<i>Codice Hex:</i>	55	<Byte>
<i>Mnemonico:</i>	U	ASCII(Bit)

Il pin AUX può essere configurato come **INPUT** generico o come **COUNTER** a seconda del contenuto del byte **<Byte>**:

0	<i>Pin AUX settato come linea di INPUT</i>
2	<i>Pin AUX settato come Trigger per il Counter a 16 Bits</i>

Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:

Se si vuole settare la linea **AUX** come **COUNTER** é necessario inviare la seguente sequenza: **85 2**.

COMANDI RELATIVI ALLA GESTIONE DEL COUNTER A 16 BIT

Sono di seguito riportati i comandi relativi alla gestione del Counter a 16 bit. Tale contatore viene incrementato dalle variazioni di stato sul pin AUX, quando questo é configurato come segnale di Trigger.

LETTURA DEL CONTATORE A 16 BIT

Sequenza di Input:

Codice Dec: **73**
Codice Hex: **49**
Mnemonic: **I**

Questo comando permette di acquisire il valore attuale del contatore a 16 bit.

Codici di Risposta:

La sequenza restituita a seguito del comando, è formata da 4 codici che esprimono la combinazione a 16 bit attualmente memorizzata all'interno dei registri del contatore; questa é trasmessa a nibble secondo il seguente formato:

Contatore (bit 0÷3) :	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
(bit 4÷7) :		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4
(bit 8÷11) :		0	0	0	0	Bit 11	Bit 10	Bit 9	Bit 8
(bit 12÷15) :		0	0	0	0	Bit 15	Bit 14	Bit 13	Bit 12

Quando il contatore raggiunge il suo massimo valore, corrispondente a 65535 (FFFF Hex), ad un successivo impulso sul Trigger, il nuovo valore della combinazione sarà 0. Se il pin AUX é configurato come Input, a seguito di questo comando, sarà sempre restituita la combinazione 0.

Esempio:

Se i registri del contatore a 16 bit contengono la combinazione 23055 (5A0F Hex), a seguito della trasmissione del comando **73**, sarà restituita la seguente sequenza: 15 0 10 5.

AZZERAMENTO DEL COUNTER A 16 BIT

Sequenza di Input:

Codice Dec: **88** **120**
Codice Hex: **58** **78**
Mnemonic: **X** **x**

A seguito di questo comando, il firmware azzerà i registri del counter a 16 bit, scrivendo in essi la combinazione **0**.

COMANDI PER LA GESTIONE DEI MESSAGGI

LETTURA NUMERO ULTIMO MESSAGGIO MEMORIZZABILE

Sequenza di Input:

Codice Dec: **77**
Codice Hex: **4D**
Mnemonico: **M**

Questo comando permette di acquisire il numero dell'ultimo messaggio memorizzabile nella EEPROM. Questo varia in funzione del dispositivo installato e corrisponde a quanto riportato nella seguente tabella.

EEPROM	N.MAX
24c02 (256 Bytes)	23
24c04 (512 Bytes)	48
24c08 (1024 Bytes)	99

FIGURA 44: NUMERO DELL'ULTIMO MESSAGGIO MEMORIZZABILE IN EEPROM

Codici di Riposta:

Il numero dell'ultimo messaggio memorizzabile viene restituito in due nibble; in particolare:

Nibble L: (MSB) 0 0 0 0 **Bit 3 Bit 2 Bit 1 Bit 0** (LSB)
 Nibble H: 0 0 0 0 **Bit 7 Bit 6 Bit 5 Bit 4**

RICHIESTA DI EEPROM PRONTA

Sequenza di Input:

Codice Dec: **66**
Codice Hex: **42**
Mnemonico: **B**

Mediante questo comando é possibile richiedere al firmware se é pronto per gestire un nuovo messaggio in EEPROM, esso va eseguito tutte le volte che si deve inviare uno dei comandi descritti in seguito per la gestione dei messaggi.

Codici di Risposta:

Il codice restituito dal firmware a seguito di tale comando sono i seguenti:

0 *EEPROM non disponibile per gestire un nuovo messaggio*
1 *EEPROM disponibile per gestire un nuovo messaggio*

MEMORIZZAZIONE DI UN MESSAGGIO

Sequenza di Input:

<i>Codice Dec:</i>	69	<NibL Mess>	<NibH Mess>
		<NibL Car.0>	<NibH Car.0>...<NibL Car.9>
		<NibH Car.9>	
<i>Codice Hex:</i>	45	<NibL Mess>	<NibH Mess>
		<NibL Car.0>	<NibH Car.0>...<NibL Car.9>
		<NibH Car.9>	
<i>Mnemonic:</i>	E	ASCII(NibL Mess)	ASCII(NibH Mess)
		ASCII(NibL Car.0)	ASCII(NibH Car.0)...
		ASCII(NibL Car.9)	ASCII(NibH Car.9)

Il messaggio, di 10 caratteri, con codice indicato in "Mess" viene memorizzato nella EEPROM. Il numero del messaggio, che deve essere inviato a nibble secondo il formato indicato in precedenza, deve essere compreso nel range $0 \div N.MAX$, dove "N.MAX" é il numero dell'ultimo messaggio memorizzabile, visto in precedenza, e ricavabile direttamente dalla tabella precedente o tramite l'apposito comando.

Car. x (nibble L) :	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
(nibble H) :		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4

Questi byte possono essere compresi in tutto il range $0 \div 255$ ($0 \div FF$ Hex).

Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Esempio:

Se si desidera memorizzare, come messaggio numero 16, la stringa "ABCDEFGHJIJ" (corrispondente ai codici: 65, 66, 67, 68, 69, 70, 71, 72, 73, 74), sarà necessario inviare la seguente sequenza:

69 0 1 1 4 2 4 3 4 4 4 5 4 6 4 7 4 8 4 9 4 10 4.

LETTURA DI UN MESSAGGIO

Sequenza di Input:

<i>Codice Dec:</i>	76	<NibL Mess>	<NibH Mess>
<i>Codice Hex:</i>	4C	<NibL Mess>	<NibH Mess>
<i>Mnemonic:</i>	L	ASCII(NibL Mess)	ASCII(NibH Mess)

Il messaggio, di 10 caratteri, con codice indicato in "Mess" viene letto dalla EEPROM e restituito in seriale. Il numero del messaggio, che deve essere inviato a nibble secondo il formato indicato in precedenza, deve essere compreso nel range $0 \div N.MAX$, dove "N.MAX" ha lo stesso significato visto in precedenza e ricavabile direttamente dalla tabella precedente o tramite l'apposito comando.

Codici di Risposta:

I codici dei 10 caratteri che compongono il messaggio sono restituiti a nibble secondo, il formato visto per il comando precedente.

Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Esempio:

Facendo riferimento all'esempio del comando precedente, se si vuole leggere il messaggio numero 16 dalla EEPROM, sarà necessario inviare la seguente sequenza: **76 0 1**. La risposta a seguito del tale comando sarà la sequenza: **1 4 2 4 3 4 4 4 5 4 6 4 7 4 8 4 9 4 10 4.**

COMANDI PER LA GESTIONE DELL' RTC E DELLA RAM-RTC

SETTAGGIO OROLOGIO

Sequenza di Input:

<i>Codice Dec:</i>	79	<NibL ORE>	<NibH ORE>	<NibL MIN>	<NibH MIN>
		<NibL SEC>	<NibH SEC>	<NibL GIO>	<NibH GIO>
		<NibL MES>	<NibH MES>	<NibL ANN>	<NibH ANN>
		<NibL GSE>	<NibH GSE>		
<i>Codice Hex:</i>	4F	<NibL ORE>	<NibH ORE>	<NibL MIN>	<NibH MIN>
		<NibL SEC>	<NibH SEC>	<NibL GIO>	<NibH GIO>
		<NibL MES>	<NibH MES>	<NibL ANN>	<NibH ANN>
		<NibL GSE>	<NibH GSE>		
<i>Mnemonic:</i>	O	ASCII(NibL ORE)	ASCII(NibH ORE)		
		ASCII(NibL MIN)	ASCII(NibH MIN)		
		ASCII(NibL SEC)	ASCII(NibH SEC)		
		ASCII(NibL GIO)	ASCII(NibH GIO)		
		ASCII(NibL MES)	ASCII(NibH MES)		
		ASCII(NibL ANN)	ASCII(NibH ANN)		
		ASCII(NibL GSE)	ASCII(NibH GSE)		

BYTE	RANGE	SIGNIFICATO
ORE	0 ... 23	ORE
MIN	0 ... 59	MINUTI
SEC	0 ... 59	SECONDI
GIO	1 ... 31	GIORNO
MES	1... 12	MESE
ANN	0 ... 99	ANNO
GSE	0 ... 6	Giorno della settimana: 0 -> DOMENICA 6 -> SABATO

FIGURA 45: RANGE DI VALIDITÀ DEI 7 BYTES DI INIZIALIZZAZIONE RTC

Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Esempio:

Se si desidera inizializzare l' RTC con la seguente data: **LUNEDI' 11-MAGGIO-1998** e la seguente ora: **12:30:40** sarà necessario inviare i codici: **79 12 0 14 1 8 2 11 0 5 0 2 6 1 0**.

LETTURA OROLOGIO

Sequenza di Input:

Codice Dec: **111**
Codice Hex: **6F**
Mnemonico: **o**

Codici di Risposta:

I codici dei 7 bytes che compongono l' ora e la data sono restituiti a nibble secondo, il formato visto per il comando precedente:

<NibL ORE>	<NibH ORE>	<NibL MIN>	<NibH MIN>
<NibL SEC>	<NibH SEC>	<NibL GIO>	<NibH GIO>
<NibL MES>	<NibH MES>	<NibL ANN>	<NibH ANN>
<NibL GSE>	<NibH GSE>		

Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

SCRITTURA NELLA RAM DELL' RTC

Sequenza di Input:

<i>Codice Dec:</i>	71	<NibL IND>	<NibH IND>	<NibL VAL>	<NibH VAL>
<i>Codice Hex:</i>	47	<NibL IND>	<NibH IND>	<NibL VAL>	<NibH VAL>
<i>Mnemonico:</i>	G	ASCII(NibL IND)	ASCII(NibH IND)		
		ASCII(NibL VAL)	ASCII(NibH VAL)		

Il dato "VAL" (0÷255) viene salvato nella RAM dell' RTC all' indirizzo "IND" (32÷255).

Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Esempio:

Se si desidera memorizzare, il dato 65 all' indirizzo 100 sarà necessario inviare la seguente sequenza:
71 4 6 1 4.

LETTURA DALLA RAM DELL' RTC

Sequenza di Input:

<i>Codice Dec:</i>	103	<NibL IND>	<NibH IND>
<i>Codice Hex:</i>	67	<NibL IND>	<NibH IND>
<i>Mnemonico:</i>	g	ASCII(NibL IND)	ASCII(NibH IND)

Viene letto un byte dalla RAM dell' RTC all' indirizzo "IND" (32÷255).

Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Codici di Risposta:

Viene restituito il dato (0÷255), tramite i due nibble che lo compongono secondo, il formato visto per il comando precedente: <NibL VAL>, <NibH VAL>.

COMANDO	CODICE	C. HEX	MNEMONICO
Reset Generale	65 97	41 61	A a
Scrittura del Byte di Presenza	89 val.L val.H	59 val.L val.H	Y ASCII(val.L) ASCII(val.H)
Lettura del Byte di Presenza	121	79	y
Lettura versione firmware	86	56	V
Settaggio del Port di OUTPUT	87 1 Dato 0	57 1 Dato 0	W SOH ASCII(Dato) NUL
Lettura del Port di INPUT	82 0	52 0	R NUL
SET Port.Bit	83 1 bit	53 1 bit	S SOH ASCII(bit)
CLEAR Port.Bit	67 1 bit	43 1 bit	C SOH ASCII(bit)
SET Port.Bit temporizzato	115 1 bit nib.L nib.H	73 1 bit nib.L nib.H	s SOH ASCII(bit) ASCII(nib.L) ASCII(nib.H)
CLEAR Port.Bit temporizzato	99 1 bit nib.L nib.H	63 1 bit nib.L nib.H	c SOH ASCII(bit) ASCII(nib.L) ASCII(nib.H)
Onda Quadra su Port.Bit	80 1 bit nib.L nib.H	50 1 bit nib.L nib.H	P SOH ASCII(bit) ASCII(nib.L) ASCII(nib.H)
Onda Quadra da "1" su Port.Bit temporizzata	112 1 bit nib.L nib.H sta.L sta.H	70 1 bit nib.L nib.H sta.L sta.H	p SOH ASCII(bit) ASCII(nib.L) ASCII(nib.H) ASCII(sta.L) ASCII(sta.H)
Onda Quadra da "0" su Port.Bit temporizzata	119 1 bit nib.L nib.H sta.L sta.H	77 1 bit nib.L nib.H sta.L sta.H	w SOH ASCII(bit) ASCII(nib.L) ASCII(nib.H) ASCII(sta.L) ASCII(sta.H)
Lettura Port.Bit	114 port bit	72 port bit	r ASCII(port) ASCII(bit)
Lettura Config. del pin AUX	102	66	f
Settaggio Config. del pin AUX	85 byte	55 byte	U ASCII(byte)

FIGURA 46: TABELLA 1 DEI COMANDI

COMANDO	CODICE	C. HEX	MNEMONICO
Reset Counter	88 120	58 78	X x
Lettura Counter	73	49	I
Rischiesta di EEPROM pronta	66	42	B
Lettura Numero ultimo messaggio	77	4D	M
Memorizzazione Messaggio	69 nib.L nib.H nib.L0 nib.H0 nib.L9 nib.H9	45 nib.L nib.H nib.L0 nib.H0 nib.L9 nib.H9	E ASCII(nib.L) ASCII(nib.H) ASCII(nib.L0) ASCII(nib.H0) ASCII(nib.L9) ASCII(nib.H9)
Lettura Messaggio	76 nib.L nib.H	4C nib.L nib.H	L ASCII(nib.L) ASCII(nib.H)
Settaggio RTC	79 nib.L0 nib.H0 nib.L6 nib.H6	4F nib.L0 nib.H0 nib.L6 nib.H6	O ASCII(nib.L0) ASCII(nib.H0) ASCII(nib.L6) ASCII(nib.H6)
Lettura RTC	111	6F	o
Scrittura RAM RTC	71 nib.L0 nib.H0 nib.L1 nib.H1	47 nib.L0 nib.H0 nib.L1 nib.H1	G ASCII(nib.L0) ASCII(nib.H0) ASCII(nib.L1) ASCII(nib.H1)
Lettura RAM RTC	103 nib.L nib.H	67 nib.L nib.H	g ASCII(nib.L) ASCII(nib.H)

FIGURA 47: TABELLA 2 DEI COMANDI

MODALITA' DI COMUNICAZIONE MASTER-SLAVE A 9 BIT

La modalità Master-Slave, sfrutta la tecnica di comunicazione a 9 bit. In particolare oltre agli 8 bit di dati, viene gestito un nono bit che serve a distinguere una chiamata, da parte della apparecchiatura "**Master**" ad una delle strutture "**Slave**", da un normale passaggio di informazioni tra il Master e il dispositivo attualmente selezionato. Quando il nono bit è posto a 1, il byte di dati deve contenere il nome, o codice di identificazione, del dispositivo con il quale si vuole comunicare, mentre ponendo questo particolare bit a 0 è poi possibile prelevare o fornire informazioni a tale dispositivo. Nel caso particolare della comunicazione con il protocollo **ALB**, il codice di identificazione deve essere quello settato in **SETUP mode (NOME)**. Quando viene inviato questo byte (**con il nono bit posto a 1**), il dispositivo si riconosce e si pone in attesa della stringa contenente, dati o comandi (**con il nono bit posto a 0**); questa deve essere al massimo di **24 byte**. In questa stringa, può esserci solo un comando che comporta la restituzione di una informazione in seriale da parte del dispositivo, se ve ne è un numero superiore, i restanti comandi di questo tipo verranno ignorati. Tra la trasmissione di un carattere ed il successivo, deve passare un tempo inferiore al tempo di **Time-Out**, in quanto, trascorso questo ritardo, si considera finita la stringa di dati ed inizia la fase di risposta.

I tempi di Time-Out relativi ai vari Baud Rate sono indicati di seguito:

<i>Baud Rate</i>	<i>Time-Out</i>
38400 Baud	550 μ sec
19200 Baud	990 μ sec
9600 Baud	1.54 msec
4800 Baud	3.08 msec
2400 Baud	6.105 msec
1200 Baud	12,1 msec

Quando scade il tempo di Time-Out, inizia la sequenza di risposta; questa consiste in un byte contenente il codice di presenza **6 (6 Hex)**, oppure da un dato o sequenza di dati, relativi ad un comando di lettura inviato nella chiamata precedente.

Esempio:

Se viene trasmessa una stringa contenente il comando di lettura Port, si avrà che in quella particolare chiamata sarà restituito il codice di presenza, mentre nella successiva, verrà trasmesso il dato acquisito dal Port, richiesto in precedenza.

Dopo che è stata completata la trasmissione dell'ultimo carattere della stringa, sarà necessario attendere un tempo di: "**Tempo di trasmissione di un carattere**" + **Time-Out** prima che arrivi il primo carattere della sequenza di risposta, trasmessa dal dispositivo.

Esempio:

Se si lavora a 38.4 KBaud, quando è stata completata la trasmissione dell'ultimo carattere, è necessario attendere un tempo di circa 810 μ sec, prima che sia completata la ricezione del primo dato di risposta trasmesso dal dispositivo.

Note:

Tra una chiamata e la successiva, è necessario attendere un tempo che è funzione del numero di comandi inviati e del tipo di operazioni che questi comportano. Una stringa di dati o comandi, trasmessa dal "Master" deve sempre contenere delle sequenze di comandi complete. Se una di queste risulta incompleta può venire ignorata, insieme alla sequenza di comando successiva, anche se completa.

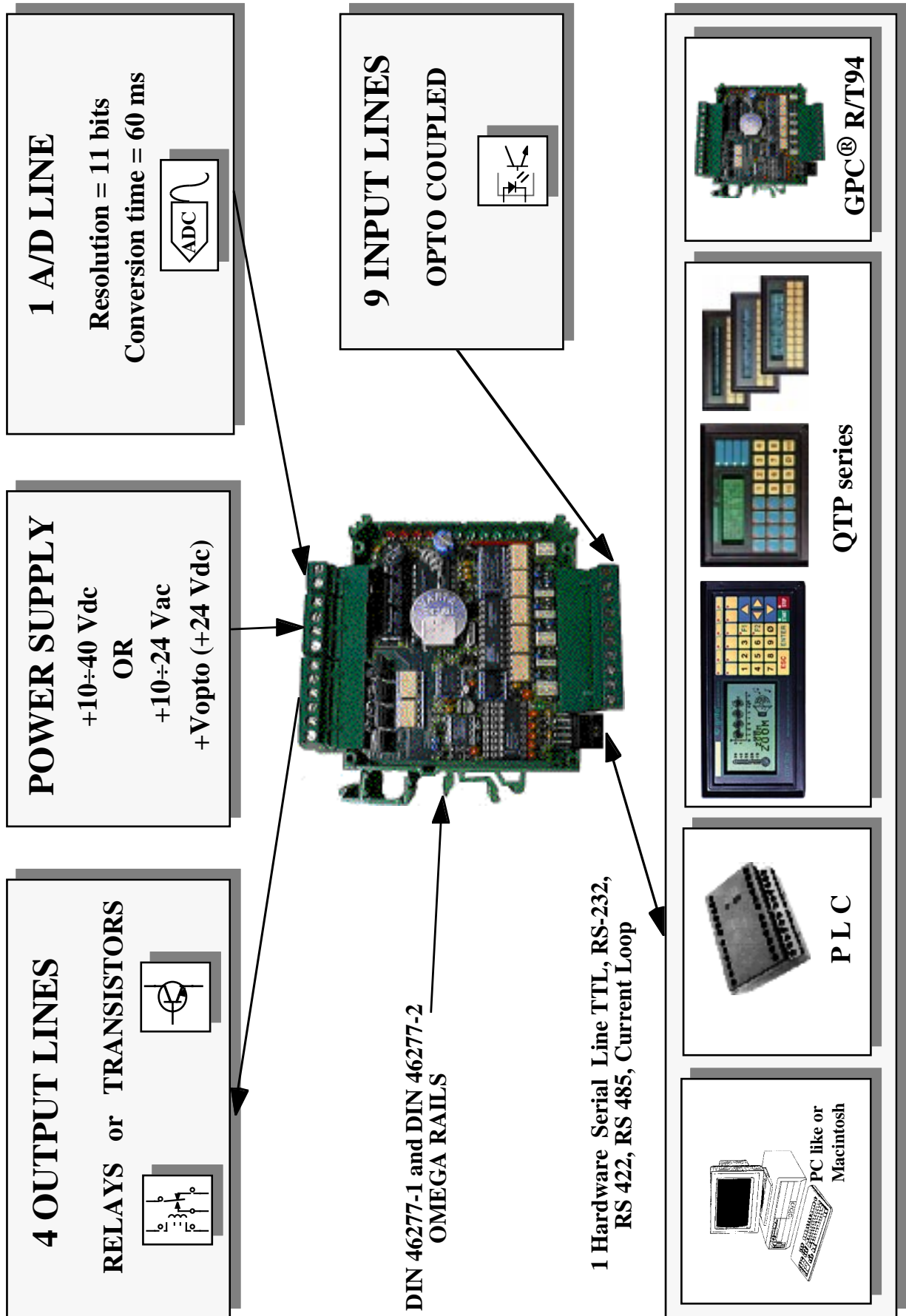


FIGURA 48: SCHEMA DELLE POSSIBILI CONNESSIONI PER LA GPC® R/T94