

Terminale Logica TL32A

manuale di riferimento

aggiornamento: 21-01-97

Detail[®]

ELETTRONICA INDUSTRIALE

TL32A - manuale di riferimento

Gennaio 1997

Tutti i diritti sono riservati. E' vietata la riproduzione di qualsiasi parte di questo manuale, in qualsiasi forma, senza esplicita autorizzazione da parte della proprietà del marchio **Detail®**.

Il costruttore si riserva di modificare, senza alcun preavviso, le caratteristiche delle apparecchiature riportate in questo manuale.

Ogni cura è stata posta nella raccolta e nella verifica della documentazione contenuta in questo manuale, tuttavia la proprietà del marchio **Detail®** non può assumersi alcuna responsabilità derivante dall'utilizzo della stessa.

Sommario

Caratteristiche tecniche	5
Montaggio e connessioni	6
<i>Figura 1. Dimensioni e fissaggio del terminale logica TL32A</i>	<i>6</i>
<i>Figura 2. Connessioni e descrizione del lay-out scheda</i>	<i>7</i>
Programmazione della logica	8
<i>Tabella 1. Mappatura delle risorse esterne</i>	<i>9</i>
<i>Tabella 2. Codifica del tasto premuto</i>	<i>9</i>
<i>Tabella 3. Posizione dei 32 caratteri del display</i>	<i>10</i>
Programmazione dei messaggi	11
<i>Tabella 4. Set di caratteri ASCII del display</i>	<i>12</i>
Esempi applicativi	13

Caratteristiche tecniche

Il terminale logica TL32A rappresenta un elemento di punta nella linea dei prodotti **Detail®** per l'automazione industriale e civile.

In un unico dispositivo sono state associate le funzioni di logica programmabile e di interfaccia operatore con caratteristiche e prestazioni elevatissime. Basti pensare che il terminale logica TL32A può contenere nella sua memoria fino a 10K istruzioni di programma e fino a 1000 messaggi alfanumerici; gli 8 ingressi e le 8 uscite digitali disponibili a bordo possono essere ampliate connettendo in modo remoto, tramite la linea RS485, fino a 31 elementi SLAVE scelti nell'ampia gamma dei prodotti **Detail®**.

Un vero e proprio computer d'automazione è dunque il terminale logica TL32A; basta solo alimentarlo ed è subito pronto per controllare il vostro impianto. Naturalmente tutto questo in modo completamente programmabile con una semplice interfaccia per il PC che consente la messa a punto di tutte le funzioni e dei messaggi direttamente nella memoria FLASH-EPROM.

Le caratteristiche tecniche evidenziano che si tratta di un prodotto del tutto innovativo con un rapporto prestazioni/prezzo elevatissimo; ovviamente il TL32A è frutto di lunga esperienza nel campo delle normative di compatibilità elettromagnetica e dispone di tutto ciò che serve per superare con ampio margine le severe prove richieste dalla marcatura CE.

INTERFACCIE ESTERNE E FUNZIONALI

- 8 ingressi optoisolati 24Vdc logica PNP/NPN con led di stato logico
- 8 uscite a relè 8A/250Vac con led di stato logico. Disponibili uscite statiche da 3Aac/dc
- Display alfanumerico LCD (2 righe da 16 caratteri) disponibile con retroilluminazione
- Tastiera a 16 tasti numerici + funzionali con beeper per la segnalazione di conferma
- Beeper attivabile dal programma utente per segnalazioni acustiche dei codici allarmi
- Seriale veloce RS485 per espansione remota su doppino fino a 31 elementi SLAVE
- Connettore per modulo PRG1 interfaccia PC per la programmazione e supervisione

CARATTERISTICHE DI PROGRAMMAZIONE

- Capacità memoria programma: 10K istruzioni (min. 5 K con 1000 messaggi da 32 car.)
- Tempo esecuzione medio per istruzione (media su tutte le istruzioni a bits): 4 μ s
- 16384 relè interni (dei quali 8192 ritentivi) + 128 timers/contatori + 3840 bytes di I/O
- Vasto set di istruzioni e di funzioni matematiche a 32 BITS
- Istruzioni di Calcolatrice per la valutazione delle espressioni a 32 BITS con segno
- Creazione di nuove istruzioni personalizzate da aggiungere a quelle base
- Orologio/calendario (opzionale) con informazione dal secondo all'anno
- Possibilità di supervisione mediante PC di tutte le variabili interne alla RAM

FUNZIONI DELL'INTERFACCIA OPERATORE

- Lettura diretta di tutti i 16 tasti e lettura diretta della variabile digitata a 32 BITS
- Visualizzazione diretta mediante indirizzo di messaggi fino a 32 caratteri
- Programmazione fino a 1000 messaggi da 32 caratteri nella memoria FLASH-EPROM
- Visualizzazione diretta di variabili (a 32 BITS con segno e punto decimale)

CARATTERISTICHE GENERALI

- Alimentazione 24Vac/dc +/-20% dotata di regolazione switching e filtro antistallo EMI
- Dimensioni del frontale: 200x150mm. Profondità massima: 40mm
- Realizzazione su unica scheda fissata sul pannello frontale. Morsettiera a vite estraibili

Montaggio e connessioni

Per il fissaggio sul quadro elettrico del terminale logica TL32A riferirsi alla seguente Figura 1.

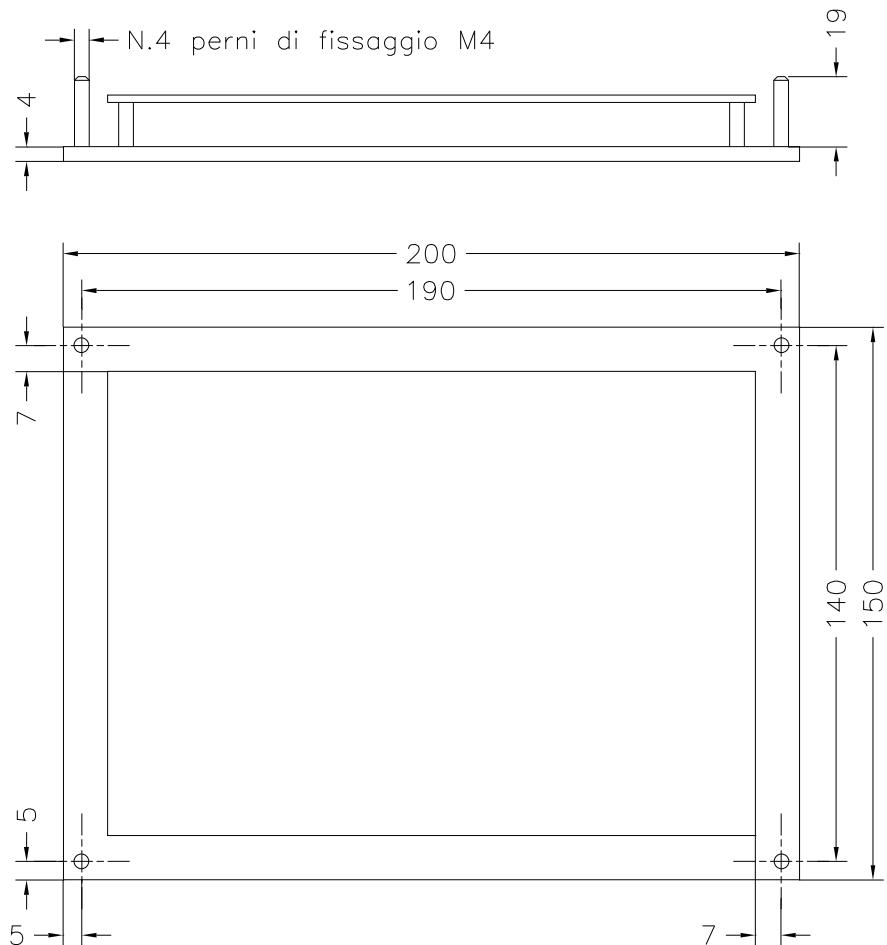


Figura 1. Dimensioni e fissaggio del terminale logica TL32A

Al fine di rendere efficaci i dispositivi di protezione dai radiodisturbi previsti sul terminale si consiglia vivamente di fissarlo al quadro o allo sportello di questo facendo attenzione a rendere conduttrice la superficie di contatto tra questi. Anche il fissaggio mediante i 4 perni M4 dovrebbe contribuire, nel miglior modo possibile, alla connessione elettrica tra la lamiera del quadro ed il pannello metallico frontale del TL32A; verificare inoltre la buona messa a terra del quadro mediante conduttori di elevata sezione.

Per le connessioni elettriche del terminale logica TL32A riferirsi alla Figura 2.

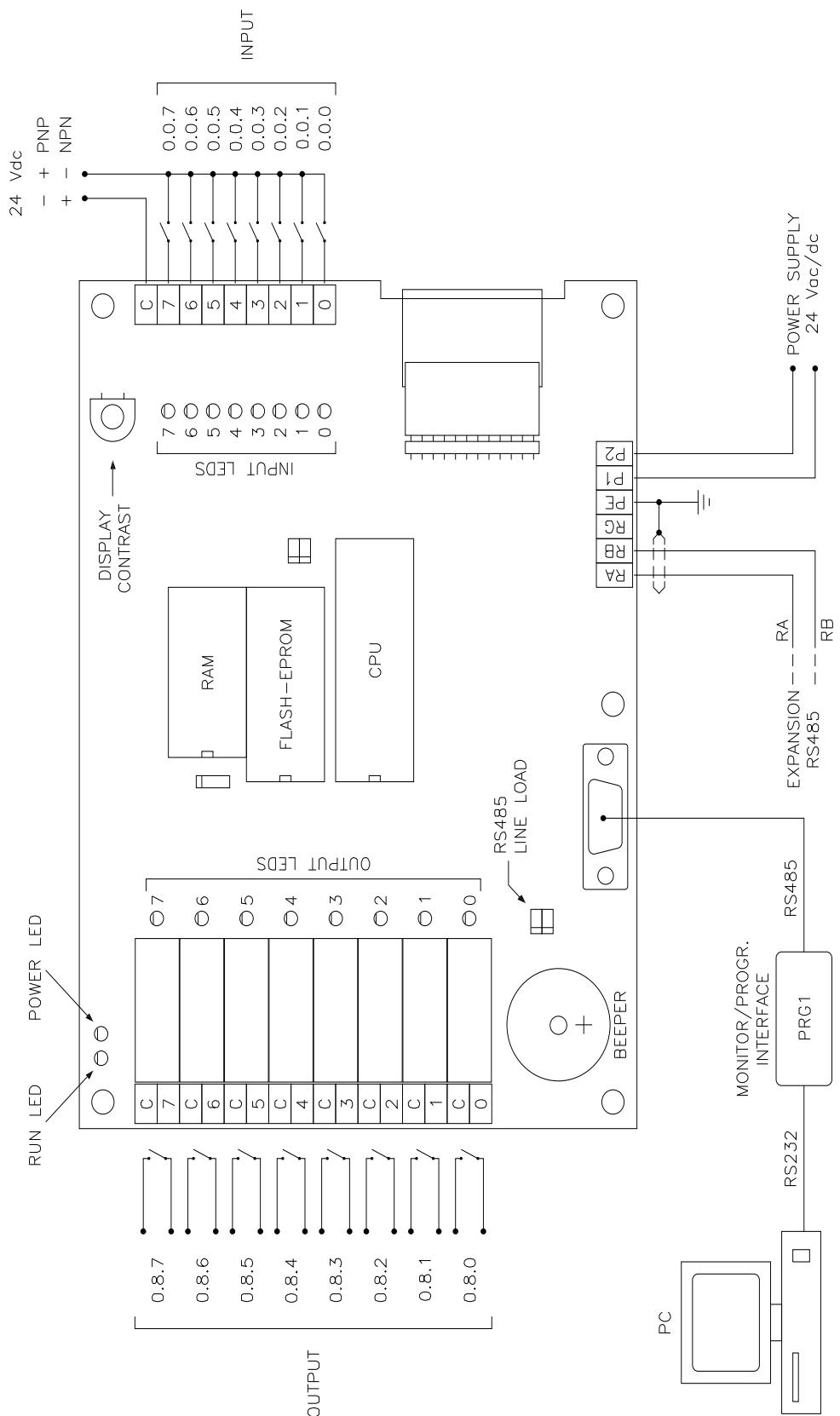


Figura 2. Connessioni e descrizione del lay-out scheda

Gli ingressi digitali sono a logica universale PNP/NPN a 24Vdc. Per l'alimentazione degli ingressi e dello stesso terminale TL32A si consiglia di utilizzare una sorgente di alimentazione apposita distinta da quella che alimenta eventuali attuatori di potenza. Non comporta alcun inconveniente l'utilizzo di questa stessa sorgente anche per eventuali alimentazioni di sensori attivi connessi agli ingressi del terminale. Si raccomanda estrema cura nel disaccoppiare fisicamente tutte le connessioni degli ingressi digitali e dell'alimentazione del terminale da qualsiasi altro cavo o apparecchiatura interessate da elevate potenze o di tipo fortemente induttivo (motori, teleruttori, elettrovalvole).

E' buona norma inserire in parallelo ad ogni attuatore induttivo (bobine teleruttori ed elettrovalvole) un filtro di tipo RC allo scopo di proteggere i contatti dei relè di uscita del terminale TL32A da eccessive scintille di commutazione. Si ricordi che le norme di compatibilità elettromagnetica, il cui soddisfacimento è obbligato dalla marcatura CE, impongono di non introdurre nell'ambiente circostante un livello di radiodisturbo superiore ad un certo limite. Questo lo si ottiene predisponendo adeguati filtri e soppressori di picchi su tutte le possibili fonti di disturbo. Il pannello terminale TL32A è caratterizzato da un grado di immunità ai disturbi esterni, ben superiore ai limiti imposti dalla norma, il che significa che eventuali malfunzionamenti di questo sono possono essere imputabili ad un cattivo utilizzo ed alla presenza di impianti esterni ben lontani dal rispetto delle norme di emissione dei disturbi.

Per la connessione della linea RS485 di espansione remota utilizzare cavo schermato ed intrecciato, avendo cura di collegare lo schermo ad un efficace punto di terra (lo stesso del terminale). Nel caso di utilizzo del TL32A ad uno degli estremi della linea RS485 inserire gli appositi ponticelli ai 2 jumper (J1 e J2) per la chiusura della linea sulla resistenza di terminazione.

Programmazione della logica

Il terminale logica TL32A è programmabile mediante il linguaggio ICL51 release 4.0; per le nozioni generali del linguaggio si rimanda alla documentazione relativa a tale software di programmazione. In questo manuale verranno illustrate esclusivamente le risorse esterne disponibili dalla specifica logica TL32A.

Per la programmazione occorre utilizzare il modulo PRG1 di interfaccia RS485/RS232 che consente la connessione della logica alla seriale RS232 del Personal Computer. Si vuole evidenziare che non esistono particolari punti di intervento sulla scheda del TL32A in fase di programmazione; infatti il programma ed i messaggi sono memorizzati tramite il modulo PRG1 direttamente nella memoria FLASH-EPROM. Non sono presenti jumper che necessitano di configurazione per la programmazione; in particolare la funzione di STOP forzato all'accensione è ottenuta mediante la funzione F9 del menu di trasferimento del software su PC.

Il terminale TL32A opera quindi come MASTER di sistema e quindi è un dispositivo già autonomo, mediante le proprie risorse esterne, nel controllo di un impianto. Eventuali ampliamenti degli I/O sono possibili mediante la linea RS485 che consente il controllo fino a 31 elementi SLAVE.

Vediamo ora quali sono le particolari risorse esterne della logica TL32A utilizzabili dal programma utente.

In Tabella 1 viene riportata la mappatura di tutte le risorse del terminale logica TL32A. L'area di ingresso (informazioni lette dal programma) è costituita dai bytes compresi tra 0.0 e 0.7 mentre l'area di uscita (informazioni scritte dal programma) dai bytes compresi tra 0.8 e 0.18.

BYTE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0.0	8 INGRESSI DIGITALI							
0.1	"7"	"6"	"5"	"4"	"3"	"2"	"1"	"0"
0.2	"DOWN"	"UP"	"CLR"	"F2"	"F1"	"ENTER"	"9"	"8"
0.3	CODIFICA TASTO PREMUTO (255, 0-15)							
0.4	VARIABILE INGRESSO DIGITATA (32 BITS SENZA SEGNO)							
0.5								
0.6								
0.7								
0.8	8 USCITE DIGITALI							
0.9	CLEAR							BEEPER
0.10	INDIRIZZO MESSAGGIO DISPLAY (0-999)							
0.11								
0.12	PUNTATORE SINISTRO POSIZIONE MESSAGGIO (0, 1-32)							
0.13	VARIABILE USCITA DISPLAY (32 BITS CON SEGNO)							
0.14								
0.15								
0.16								
0.17	PUNTATORE DESTRO POSIZIONE USCITA (0, 1-32)							
0.18	NUMERO CIFRE DECIMALI USCITA (0, 1-10)							

Tabella 1. Mappatura delle risorse esterne

Il byte 0.0 permette la lettura dello stato logico degli 8 ingressi digitali; questi sono ingressi di utilizzo generico aggiornati ad ogni ciclo di programma.

I bytes da 0.1 a 0.2 riportano lo stato dei singoli tasti della tastiera del terminale; il valore ad “uno” logico dei bits permane per tutto il tempo per cui il tasto è premuto. E’ disponibile per la lettura della tastiera anche il byte 0.3 contenente la codifica del tasto attualmente premuto. Il valore di riposo di tale byte è 255 (nessun tasto premuto); la pressione di un tasto cambia il contenuto di tale byte secondo la codifica numerica dei 16 tasti riportata in Tabella 2.

TASTO	CODIFICA
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
ENTER	10
F1	11
F2	12
CLR	13
UP	14
DOWN	15
nessuno tasto	255

Tabella 2. Codifica del tasto premuto

Per una più semplice gestione da parte del programma utente della funzione di inserimento di una variabile numerica, è stata prevista l'area di ingresso dal byte 0.4 al byte 0.7; questa rappresenta una variabile a 4 bytes aggiornata automaticamente dal terminale logica TL32A e disponibile alla lettura diretta da parte del programma. Il valore contenuto in tale variabile non è altro che l'ultimo numero digitato mediante la tastiera numerica (tasti 0-9); tale valore viene aggiornato alla pressione di ogni tasto numerico fino al massimo consentito da una variabile a 32 bits senza segno.

Il tasto CLR permette in ogni momento di azzerare tale digitazione; esistono tuttavia altri due modi per azzerare il valore della variabile. Il primo è quello di utilizzare il bit di uscita 0.9.7: forzando ad “uno” logico tale bit si azzerà la variabile digitata. L’altro è automatico all’introduzione di una nuova sequenza numerica dopo aver premuto almeno uno degli altri tasti della tastiera (ENTER, F1, F2, UP, DOWN). Infatti la pressione del tasto ENTER può essere utilizzata dal programma utente per campionare il valore introdotto; in questo modo quando si introduce un nuovo valore la variabile di ingresso viene preventivamente ed automaticamente azzerata.

Il byte di uscita 0.8 contiene le 8 uscite digitali; per l’attivazione delle uscite occorre forzare con le istruzioni del programma lo stato logico “uno” nei relativi bits.

Il byte 0.9 viene utilizzato solo per quanto riguarda il bit 0 ed il bit 7. Il bit di uscita 0 consente il pilotaggio diretto del beeper presente sul terminale; forzando ad “uno” logico tale bit il beeper viene attivato fino a che permane tale valore. Il bit 7 di uscita può essere utilizzato per l’azzeramento forzato della variabile di input digitata con la tastiera.

Dal byte di uscita 0.10 al byte 0.18 si trova l’area di comunicazione con il display LCD del terminale logica TL32A. In particolare vi sono due distinte vie per modificare il contenuto del display: la scrittura di un testo alfanumerico fisso e preprogrammato sulla memoria programma e la scrittura di un valore numerico fornito direttamente dal programma utente.

Entrambe le possibilità permettono di scegliere in quale punto dello schermo agire fornendo i relativi due puntatori di schermo; il puntatore di schermo è un valore numerico scritto in apposita area di uscita che indica uno dei 32 caratteri del display LCD. Il valore 0 del puntatore disabilita la rispettiva operazione di visualizzazione, mentre un valore compreso tra 1 e 32 causa la scrittura del testo e della variabile ad iniziare dal carattere puntato.

I 32 caratteri del display LCD sono numerati secondo quanto indicato dalla Tabella 3.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

Tabella 3. Posizione dei 32 caratteri del display

Per il posizionamento del messaggio alfanumerico il valore del puntatore schermo coincide con il primo carattere a sinistra del messaggio stesso, mentre per la variabile numerica coincide con il carattere più a destra della stringa che ne rappresenta il valore (eventuali cifre decimali comprese); questo comporta una giustificazione a sinistra dei messaggi e a destra dei valori numerici.

Le due distinte operazioni di scrittura sul display avvengono contemporaneamente ad ogni ciclo di scansione del programma utente. Per primo viene scritto il testo del messaggio a partire dal relativo puntatore e procedendo verso destra per quanti caratteri dispone il testo (max 32); successivamente viene scritta la stringa corrispondente al valore numerico iniziando, con la cifra meno significativa, dal puntatore fornito e continuando verso sinistra fino ad arrivare alla cifra più significativa o all’eventuale segno negativo o punto decimale se il valore visualizzato è minore di 1.

Si consideri che i caratteri necessari alla visualizzazione del valore numerico sovrascrivono gli eventuali caratteri forzati con il testo del messaggio. Per questo, nell’utilizzo di entrambe le tecniche allo scopo di visualizzare una dicitura seguita da un valore, occorre fare attenzione ad inserire un certo numero di caratteri “spazio” a destra della dicitura fino ad arrivare al carattere più a destra del valore numerico (coincidente con il puntatore della variabile); così facendo si avrà la garanzia di “pulire” la parte di schermo utilizzata per la visualizzazione della variabile da eventuali cifre più significative di valori numerici precedenti, che richiedevano un numero di cifre superiore.

Ogni ciclo di scansione del programma utente vengono dunque valutati i valori presenti nell’area di uscita corrispondente al display e conseguentemente viene aggiornato lo schermo. Nel caso occorra realizzare schermate costituite da più messaggi e valori numerici distinti si possono definire valori diversi nell’area di uscita in cicli di scansione successivi del programma utente. Riferirsi al paragrafo “Esempi applicativi” per meglio comprendere tale tecnica.

Ritornando all’area di uscita, utilizzata per la gestione del display, dalla Tabella 1 risulta che il messaggio alfanumerico che si desidera scrivere sullo schermo viene individuato mediante il valore di indirizzo scritto nel doppio byte 0.10-0.11; questa word può assumere valori compresi tra 0 e 999 per indicare uno dei possibili 1000 messaggi disponibili a bordo della memoria programma. Nel paragrafo successivo verrà spiegato come fare per inserire i testi dei messaggi nella memoria di programma della logica. Il byte 0.12 rappresenta invece il puntatore sinistro del testo indirizzato correntemente; come già illustrato un valore 0 in tale byte disabilita l’operazione di scrittura del messaggio.

La variabile a 32 bits con segno forzata nei bytes di uscita da 0.13 a 0.16 costituisce il valore numerico da visualizzare sul display; il byte 0.17 è il puntatore destro della stringa corrispondente visualizzata sul display. Il byte 0.18 permette di indicare la posizione dell’eventuale punto decimale della variabile; se questo byte contiene il valore 0 nessun punto decimale è visualizzato e dunque la stringa scritta sullo schermo è un numero intero corrispondente al valore fornito (con eventuale segno negativo). Se il byte 0.18 contiene un valore compreso tra 1 e 10 all’interno delle stesse cifre del numero intero fornito comparirà il carattere “punto” nella posizione tale che a destra del punto ci saranno un numero di cifre pari al valore del byte 0.18.

Programmazione dei messaggi

L’insieme dei testi dei messaggi alfanumerici visualizzabili sul display viene archiviato in modo permanente a bordo della stessa memoria FLASH-EPROM utilizzata per il programma utente di funzionamento.

Questa operazione viene fatta inserendo all’interno del listato programma ed in punti qualsiasi le stringhe volute, precedute dal numero identificativo del messaggio:

N >MESSAGGIO N

(N = 0÷999)

Il carattere > viene utilizzato come indicatore di inizio stringa: il carattere successivo a tale indicatore viene considerato il primo carattere del messaggio (quello scritto nel carattere individuato dal puntatore display). La stringa digitata nel file sorgente del programma può contenere al massimo 32 caratteri ma può anche essere lunga un solo carattere; il terminale logica TL32A invierà al display un numero di caratteri pari alla lunghezza della stringa digitata (compresi gli eventuali caratteri “spazio” trovati al termine).

E’ ovvio che la scrittura dei testi all’interno della stessa memoria programma determina una occupazione parziale della stessa, a scapito delle istruzioni massime programmabili. Tuttavia sul terminale logica TL32A è installata una memoria FLASH-EPROM da 64 Kbyte che permette, nel caso peggiore di inserimento di 1000 testi da 32 caratteri, di avere ancora disponibile lo spazio per 5000 istruzioni di programma di tipo base. Il sistema è stato comunque concepito per compattare il più possibile i testi all’interno della memoria programma; l’unica avvertenza da rispettare, per permettere la riduzione massima ottenibile della memoria occupata, è quella di iniziare a numerare i messaggi da 0 e di procedere progressivamente da tale numero senza lasciare grossi “buchi” nella numerazione.

I testi inseriti nel listato programma costituiscono un unico insieme i cui singoli elementi sono reperiti mediante l’indirizzo identificatore. Non esiste alcun limite a far comparire più volte lo stesso messaggio in aree diverse dello schermo (ovviamente utilizzando la tecnica dei cicli di scansione successivi) oppure nel suddividere gli indirizzi in gruppi omogenei di lingue diverse. Per esempio i messaggi da 0 a 255 possono essere in lingua Italiana mentre i messaggi da 256 a 511 in lingua Inglese e così via; semplicemente gestendo il tipo di messaggio con il byte basso dell’indirizzo ed il tipo di lingua con il byte alto, si realizza in modo molto semplice una gestione multilingua.

In Tabella 4 riportiamo il set di caratteri ASCII visualizzabili dal display LCD del terminale logica TL32A.

ASCII CODE			D0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
			D1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
			D2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	
			D3	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
D7	D6	D5	D4	hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	0	2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0	0	1	1	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0	1	0	0	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	0
0	1	0	1	5	P	Q	R	S	T	U	V	W	X	Y	Z	[¥]	^	-
0	1	1	0	6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0	1	1	1	7	p	q	r	s	t	u	v	w	x	y	z	{		}	Righ	Left

Tabella 4. Set di caratteri ASCII del display

Esempi applicativi

Riportiamo nel seguito due semplici programmi applicativi che illustrano alcune tecniche di gestione del display del terminale logica TL32A.

Il primo esempio è una traccia di come realizzare la scrittura di più messaggi e più valori numerici sulla stessa schermata del display: in cicli di scansione alterni vengono definiti valori diversi nell'area di uscita corrispondente al display.

Il secondo mostra una tecnica di gestione completamente alternativa del display. Infatti vengono definiti un certo insieme di messaggi fatti solo da un carattere ASCII: il valore di indirizzo messaggio corrisponde al valore ASCII del singolo carattere da visualizzare nella posizione indicata dal puntatore di messaggio. In questo modo il display è completamente libero da ogni gestione interna del sistema e può essere scritto liberamente dal programma utente definendo il contenuto dell'area di schema.

```
=====
`           Programma 1 dimostrativo Terminale Logica TL32A
`



=====
External_Input      = 0.0           'ingressi esterni
Tasto_0             = 0.1.0         'tasto "0"
Tasto_1             = 0.1.1         'tasto "1"
Tasto_2             = 0.1.2         'tasto "2"
Tasto_3             = 0.1.3         'tasto "3"
Tasto_4             = 0.1.4         'tasto "4"
Tasto_5             = 0.1.5         'tasto "5"
Tasto_6             = 0.1.6         'tasto "6"
Tasto_7             = 0.1.7         'tasto "7"
Tasto_8             = 0.2.0         'tasto "8"
Tasto_9             = 0.2.1         'tasto "9"
Tasto_ENTER         = 0.2.2         'tasto "10"
Tasto_F1            = 0.2.3         'tasto "11"
Tasto_F2            = 0.2.4         'tasto "12"
Tasto_CLR           = 0.2.5         'tasto "13"
Tasto_UP            = 0.2.6         'tasto "14"
Tasto_DOWN          = 0.2.7         'tasto "15"
Key_Code            = 0.3           'codifica tasto premuto
Input_Variable      = 0.4           'variabile ingresso digitata

External_Output      = 0.8           'uscite esterne
Beeper_Clear         = 0.9.0         'abilitazione beeper
Clear_Input_Variable = 0.9.7         'azzeramento variabile ingresso
Message_Number       = 0.10          'numero di messaggio
Message_Pointer      = 0.12          'puntatore display messaggio
Output_Variable      = 0.13          'variabile uscita display
Output_Variable_Pointer = 0.17        'puntatore display variabile
```

```

Output_Variable_Decimal = 0.18           'cifre decimali variabile
\\
LD      F.1
CPL    M.100.0

LD      T.200
OUT    P.0.IN

LD      P.0.OUTU
DEC4   M.200
INC4   M.204

LD      M.100.0           'scrittura 2 testi + 2 variabili
MOV2   Message_Number    K.1
MOV1   Message_Pointer   K.1
MOV4   Output_Variable   M.200
MOV1   Output_Variable_Pointer K.16
MOV1   Output_Variable_Decimal K.1

LDNOT  M.100.0
MOV2   Message_Number    K.2
MOV1   Message_Pointer   K.17
MOV4   Output_Variable   M.204
MOV1   Output_Variable_Pointer K.32
MOV1   Output_Variable_Decimal K.3

`      12345678901234567890123456789012
0      >
1      >VAR1=
2      >VAR2=


END

```

Programma 2 dimostrativo Terminale Logica TI-32A

Message_Number	= 0.10	'numero di messaggio
Message_Pointer	= 0.12	'puntatore display messaggio

```
#####
LD      F.1
MOVASC M.510  | PROVA SCRITTURA CARATTERI ASCII |
```

```
#####
LD      T.500          'loop di scarico caratteri ASCII
OUT    P.0.IN
```

```
LD      P.0.OUTU
INC2    M.500
MOVADD M.502    M.510
ADD2    M.502    M.502    M.500
SUB2    M.502    M.502    K.1
MOV1    M.504    @M.502
MOV1    M.505    K.0
MOV2    Message_Number    M.504
MOV1    Message_Pointer    M.500
CMP2    M.500    K.32
AND     F.=
MOV2    M.500    K.0
```

'Caratteri ASCII riconosciuti

```
32    >
33    >!
34    >"
35    >#
36    >$
37    >%
38    >&
amp;39    >'
40    >(
41    >)
42    >*
43    >+
44    >,
45    >-
46    >.
47    >/
48    >0
49    >1
50    >2
51    >3
```

```
52    >4
53    >5
54    >6
55    >7
56    >8
57    >9
58    >:
59    >;
60    ><
61    >=
62    >>
63    >?
64    >@
65    >A
66    >B
67    >C
68    >D
69    >E
70    >F
71    >G
72    >H
73    >I
74    >J
75    >K
76    >L
77    >M
78    >N
79    >O
80    >P
81    >Q
82    >R
83    >S
84    >T
85    >U
86    >V
87    >W
88    >X
89    >Y
90    >Z
```

```
91    >[  
93    >]  
94    >^  
95    >_  
96    >`  
97    >a  
98    >b  
99    >c  
100   >d  
101   >e  
102   >f  
103   >g  
104   >h  
105   >i  
106   >j  
107   >k  
108   >l  
109   >m  
110   >n  
111   >o  
112   >p  
113   >q  
114   >r  
115   >s  
116   >t  
117   >u  
118   >v  
119   >w  
120   >x  
121   >y  
122   >z  
123   >{  
125   >}
```

.....

END