

Terminale Slave PTD1B

manuale di riferimento

aggiornamento: 27-11-97

Detail[®]

ELETTRONICA INDUSTRIALE

PTD1B - manuale di riferimento

Ottobre 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

Montaggio e connessioni	5
<i>Figura 1. Dimensioni e fissaggio del terminale slave PTD1B</i>	<i>5</i>
<i>Figura 2. Connessioni e descrizione del lay-out scheda</i>	<i>6</i>
Gestione delle risorse	7
<i>Tabella 1. Mappatura delle risorse esterne</i>	<i>7</i>
<i>Tabella 2. Codifica del tasto premuto</i>	<i>8</i>
<i>Tabella 3. Posizione dei 24 caratteri del display</i>	<i>9</i>
Programmazione dei messaggi	10
<i>Tabella 4. Set di caratteri ASCII del display</i>	<i>11</i>
Esempi applicativi	12

Montaggio e connessioni

Per il fissaggio sul quadro elettrico del terminale slave PTD1B riferirsi alla seguente Figura 1.

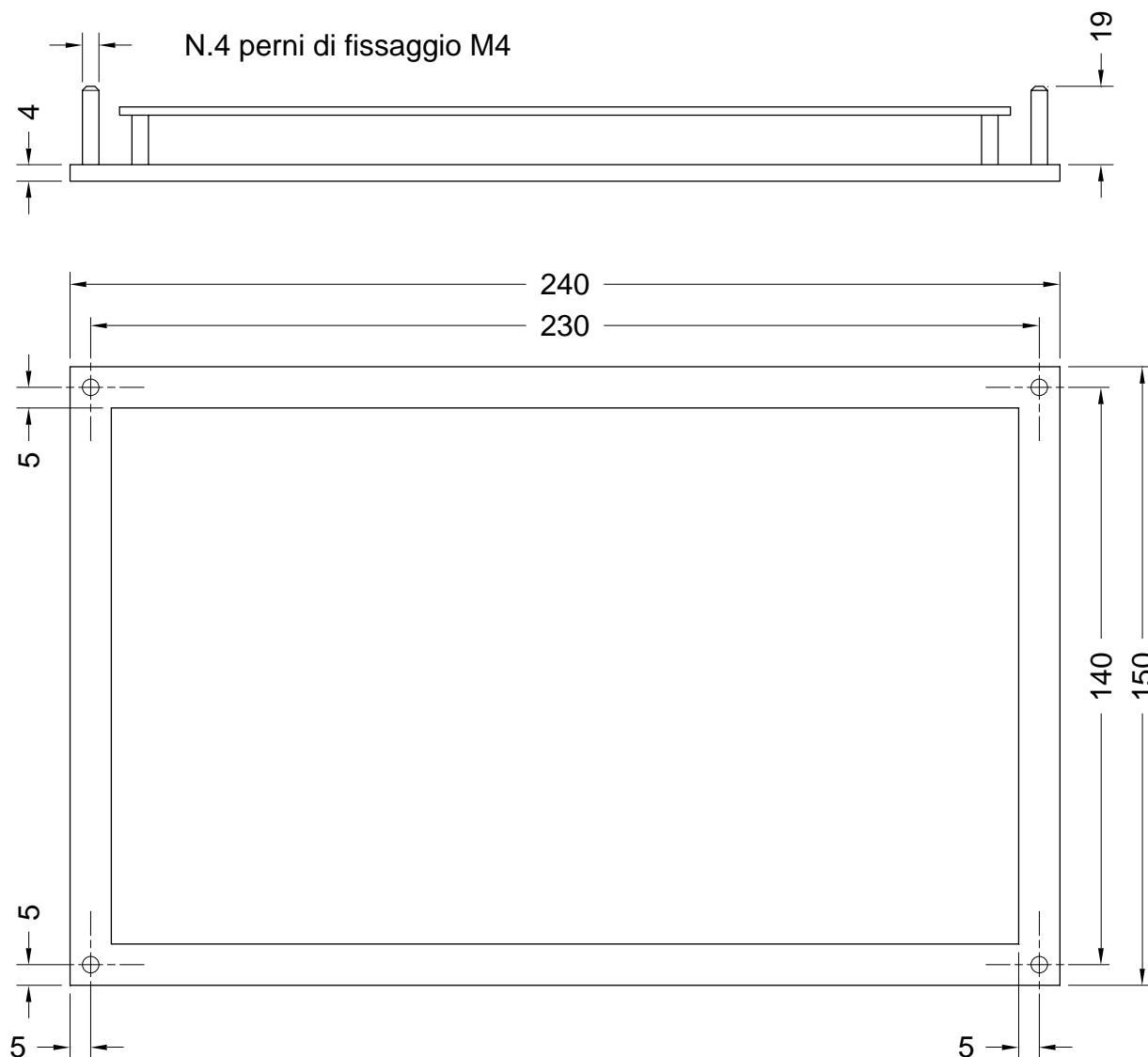


Figura 1. Dimensioni e fissaggio del terminale slave PTD1B

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 conduttiva 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 PTD1B; verificare inoltre la buona messa a terra del quadro mediante conduttori di elevata sezione.

Per le connessioni elettriche del terminale slave PTD1B riferirsi alla Figura 2.

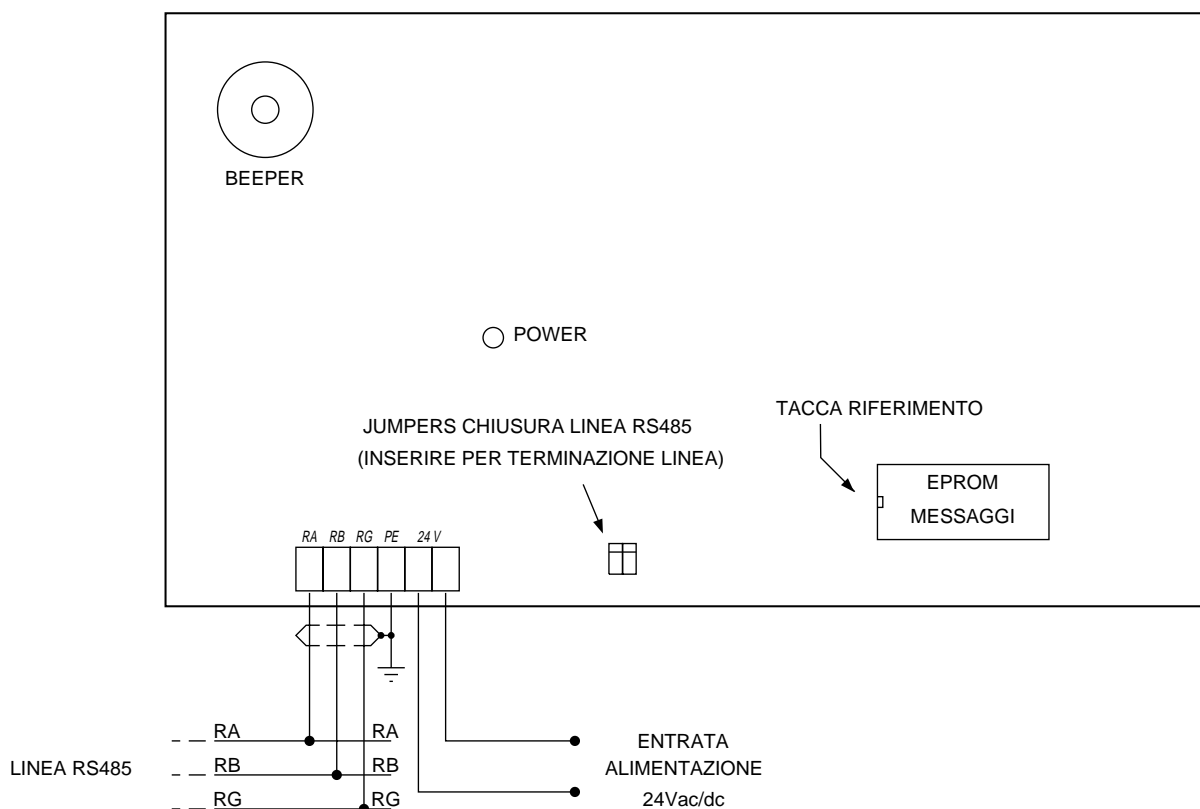


Figura 2. Connessioni e descrizione del lay-out scheda

Per l'alimentazione del terminale PTD1B si consiglia di utilizzare una sorgente di alimentazione apposita distinta da quella che alimenta eventuali attuatori di potenza. Si raccomanda estrema cura nel disaccoppiare fisicamente le connessioni dell'alimentazione del terminale da qualsiasi altro cavo o apparecchiatura interessate da elevate potenze o di tipo fortemente induttivo (motori, teleruttori, elettrovalvole).

Per la connessione della linea RS485 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 PTD1B ad uno degli estremi della linea RS485 inserire gli appositi ponticelli ai 2 jumper per la chiusura della linea sulla resistenza di terminazione.

Gestione delle risorse

Vediamo ora quali sono le particolari risorse esterne del terminale slave PTD1B utilizzabili dal programma utente sul master di sistema.

In Tabella 1 viene riportata la mappatura di tutte le risorse del terminale slave PTD1B. L'area di ingresso (informazioni lette dal programma) è costituita dai bytes compresi tra *.0 e *.6 mentre l'area di uscita (informazioni scritte dal programma) dai bytes compresi tra *.7 e *.18. Il carattere * indica il numero di slave scelto per il terminale in questione e può essere programmato sulla stessa EPROM messaggi come spiegato nel prossimo paragrafo.

BYTE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
*.0	"7"	"6"	"5"	"4"	"3"	"2"	"1"	"0"
*.1	"DOWN"	"UP"	"CLR"	"F2"	"F1"	"ENTER"	"9"	"8"
*.2	CODIFICA TASTO PREMUTO (255, 0-15)							
*.3	VARIABILE INGRESSO DIGITATA (32 BITS SENZA SEGNO)							
*.4								
*.5								
*.6								
*.7	LED "7"	LED "6"	LED "5"	LED "4"	LED "3"	LED "2"	LED "1"	LED "0"
*.8	LED "DW"	LED "UP"	LED "CL"	LED "F2"	LED "F1"	LED "EN"	LED "9"	LED "8"
*.9	CLEAR							BEEPER
*.10	INDIRIZZO MESSAGGIO DISPLAY (0-1193)							
*.11								
*.12	PUNTATORE SINISTRO POSIZIONE MESSAGGIO (0, 1-24)							
*.13	VARIABILE USCITA DISPLAY (32 BITS CON SEGNO)							
*.14								
*.15								
*.16								
*.17	PUNTATORE DESTRO POSIZIONE VARIABILE USCITA (0, 1-24)							
*.18	NUMERO CIFRE DECIMALI VARIABILE USCITA (0, 1-10)							

Tabella 1. Mappatura delle risorse esterne

I bytes da *.0 a *.1 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 *.2 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 *.3 al byte *.6; questa rappresenta una variabile a 4 bytes aggiornata automaticamente dal terminale slave PTD1B 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 *.9.7: forzando ad "uno" logico tale bit si azzerla 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.

I due bytes di uscita *.7 ed *.8 consentono di accendere direttamente i 16 leds presenti in corrispondenza di ciascun tasto del pannello terminale PTD1B. Il valore logico "1" del bit corrisponde all'accensione del relativo led mentre il valore logico "0" lo spegne.

Il byte di uscita *.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 *.10 al byte *.18 si trova l'area di comunicazione con il display LCD del terminale slave PTD1B. In particolare vi sono due distinte vie per modificare il contenuto del display: la scrittura di un testo alfanumerico fisso e preprogrammato sulla memoria EPROM del terminale 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 24 caratteri del display a matrice di led. Il valore 0 del puntatore

disabilita la rispettiva operazione di visualizzazione, mentre un valore compreso tra 1 e 24 causa la scrittura del testo e della variabile ad iniziare dal carattere puntato.

I 24 caratteri del display 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
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Tabella 3. Posizione dei 24 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 24); 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. A questo proposito occorre tener conto che il terminale slave PTD1B necessita di almeno 3ms per elaborare le informazioni inviate in un ciclo di scansione dal master; per questo motivo quando si inviano diverse configurazioni del display, in cicli di scansione successivi, occorre verificare che il tempo di ciclo del programma utente sia superiore a 3ms. Normalmente programmi utenti con un minimo di complessità soddisfano tale condizione; tuttavia se non fosse noto il tempo di ciclo o si verificassero visualizzazioni parzialmente bloccate (over-running) si consiglia di inviare le distinte configurazioni dell'area display ogni due o più cicli di scansione successivi. 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 *.10-*.11; questa word può assumere valori compresi tra 0 e 1193 per indicare uno dei possibili 1194 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 *.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 *.13 a *.16 costituisce il valore numerico da visualizzare sul display; il byte *.17 è il puntatore destro della stringa corrispondente visualizzata sul display. Il byte *.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 *.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 *.18.

Programmazione dei messaggi

L'insieme dei testi dei messaggi alfanumerici visualizzabili sul display viene archiviato in modo permanente sulla memoria di tipo EPROM presente a bordo del terminale.

Questa operazione viene fatta digitando un file di testo con estensione .MES contenente l'elenco dei messaggi preceduti dal numero identificativo degli stessi:

N >MESSAGGIO N (N = 0÷1193)

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 può contenere al massimo 24 caratteri ma può anche essere lunga un solo carattere; il terminale slave PTD1B invierà al display un numero di caratteri pari alla lunghezza della stringa digitata (compresi gli eventuali caratteri "spazio" trovati al termine).

Il file sorgente .MES permette anche di definire un indirizzo di slave, scelto per il terminale, diverso da quello di default (3). Per questa operazione utilizzare l'istruzione seguente:

SLAVE S (S = 1÷31)

L'istruzione SLAVE può comparire in un punto qualsiasi del listato sorgente dei messaggi; con tale istruzione si viene a definire un particolare indirizzo di slave memorizzandolo in una ben precisa locazione della EPROM messaggi, nota al sistema operativo.

Infine il file .MES può contenere linee di commento facendo precedere il testo del commento dal carattere speciale ' (apice); linee bianche (solo carattere CR) permettono di spaziare e rendere più leggibili le linee del file messaggi.

Successivamente il file sorgente dei messaggi (.MES) deve essere compilato mediante il programma PTD1B.EXE. Questo compilatore provvederà alla creazione di un file .EPR da programmare nella EPROM del terminale utilizzando un qualsiasi programmatore di tipo commerciale. Il file .EPR è ottenuto mediante l'unione del sistema operativo del terminale, contenuto nel file PTD1B.EPR (che deve essere disponibile nella directory corrente di compilazione), e l'insieme dei testi dei messaggi trovati nel file sorgente .MES.

La sintassi del comando di compilazione è la seguente:

PTD1B PROVA

dove PROVA.MES è l'ipotetico file sorgente dei messaggi.

I testi inseriti nella memoria EPROM 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 del terminale slave PTD1B.

ASCII CODE				D0	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	1	1	1	1	1	1	1	1
D6	D5	D4	hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	0	0	0	0	ì	Up	Right	Down	Left	¿	à			ò	ù	Ñ	ç	ê	É	é
0	0	1	1	1	è	Æ	æ	Å	å	Ä	ä	Ö	ö	Ü	ü			ß	£	¥
0	1	0	2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
0	1	1	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
1	0	0	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	0	1	5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_	
1	1	0	6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
1	1	1	7	p	q	r	s	t	u	v	w	x	y	z	{		}	~		

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 slave PTD1B.

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 Slave PTD1B      \
\                               \
////////////////////////////////////

```

Tasto_0	= 3.0.0	\tasto "0"
Tasto_1	= 3.0.1	\tasto "1"
Tasto_2	= 3.0.2	\tasto "2"
Tasto_3	= 3.0.3	\tasto "3"
Tasto_4	= 3.0.4	\tasto "4"
Tasto_5	= 3.0.5	\tasto "5"
Tasto_6	= 3.0.6	\tasto "6"
Tasto_7	= 3.0.7	\tasto "7"
Tasto_8	= 3.1.0	\tasto "8"
Tasto_9	= 3.1.1	\tasto "9"
Tasto_ENTER	= 3.1.2	\tasto "10"
Tasto_F1	= 3.1.3	\tasto "11"
Tasto_F2	= 3.1.4	\tasto "12"
Tasto_CLR	= 3.1.5	\tasto "13"
Tasto_UP	= 3.1.6	\tasto "14"
Tasto_DOWN	= 3.1.7	\tasto "15"
Key_Code	= 3.2	\codifica tasto premuto
Input_Variable	= 3.3	\variabile ingresso digitata
Led_0	= 3.7.0	\led tasto "0"
Led_1	= 3.7.1	\led tasto "1"
Led_2	= 3.7.2	\led tasto "2"

Led_3	= 3.7.3	`led tasto "3"
Led_4	= 3.7.4	`led tasto "4"
Led_5	= 3.7.5	`led tasto "5"
Led_6	= 3.7.6	`led tasto "6"
Led_7	= 3.7.7	`led tasto "7"
Led_8	= 3.8.0	`led tasto "8"
Led_9	= 3.8.1	`led tasto "9"
Led_ENTER	= 3.8.2	`led tasto "ENTER"
Led_F1	= 3.8.3	`led tasto "F1"
Led_F2	= 3.8.4	`led tasto "F2"
Led_CLR	= 3.8.5	`led tasto "CLR"
Led_UP	= 3.8.6	`led tasto "UP"
Led_DOWN	= 3.8.7	`led tasto "DOWN"
Beeper_Clear	= 3.9.0	`abilitazione beeper
Clear_Input_Variable	= 3.9.7	`azzeramento variabile ingresso
Message_Number	= 3.10	`numero di messaggio
Message_Pointer	= 3.12	`puntatore display messaggio
Output_Variable	= 3.13	`variabile uscita display
Output_Variable_Pointer	= 3.17	`puntatore display variabile
Output_Variable_Decimal	= 3.18	`cifre decimali variabile

////////////////////////////////////

```
LD      F.1
INC1    M.100
```

```
LD      T.200
OUT     P.0.IN
```

```
LD      P.0.UTU
DEC4    M.200
INC4    M.204
```

```
LD      M.100.1           `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.11
MOV1    Output_Variable_Decimal K.1
```

```
LDNOT    M.100.1
MOV2     Message_Number    K.2
MOV1     Message_Pointer   K.14
MOV4     Output_Variable    M.204
MOV1     Output_Variable_Pointer K.24
MOV1     Output_Variable_Decimal K.3
```

```
END
```

```

\ //////////////////////////////////////
\          Programma 1 dimostrativo Terminale Slave PTD1B          \
\          - Elenco Messaggi -                                     \
\ //////////////////////////////////////

```

```

\      123456789012345678901234
0      >
1      >VAR1=
2      >VAR2=

```

Tasto_0	= 3.0.0	`tasto "0"
Tasto_1	= 3.0.1	`tasto "1"
Tasto_2	= 3.0.2	`tasto "2"
Tasto_3	= 3.0.3	`tasto "3"
Tasto_4	= 3.0.4	`tasto "4"
Tasto_5	= 3.0.5	`tasto "5"
Tasto_6	= 3.0.6	`tasto "6"
Tasto_7	= 3.0.7	`tasto "7"
Tasto_8	= 3.1.0	`tasto "8"
Tasto_9	= 3.1.1	`tasto "9"
Tasto_ENTER	= 3.1.2	`tasto "10"
Tasto_F1	= 3.1.3	`tasto "11"
Tasto_F2	= 3.1.4	`tasto "12"
Tasto_CLR	= 3.1.5	`tasto "13"
Tasto_UP	= 3.1.6	`tasto "14"
Tasto_DOWN	= 3.1.7	`tasto "15"
Key_Code	= 3.2	`codifica tasto premuto
Input_Variable	= 3.3	`variabile ingresso digitata
Led_0	= 3.7.0	`led tasto "0"
Led_1	= 3.7.1	`led tasto "1"
Led_2	= 3.7.2	`led tasto "2"
Led_3	= 3.7.3	`led tasto "3"
Led_4	= 3.7.4	`led tasto "4"
Led_5	= 3.7.5	`led tasto "5"
Led_6	= 3.7.6	`led tasto "6"
Led_7	= 3.7.7	`led tasto "7"
Led_8	= 3.8.0	`led tasto "8"
Led_9	= 3.8.1	`led tasto "9"
Led_ENTER	= 3.8.2	`led tasto "ENTER"
Led_F1	= 3.8.3	`led tasto "F1"
Led_F2	= 3.8.4	`led tasto "F2"
Led_CLR	= 3.8.5	`led tasto "CLR"
Led_UP	= 3.8.6	`led tasto "UP"
Led_DOWN	= 3.8.7	`led tasto "DOWN"
Beeper_Clear	= 3.9.0	`abilitazione beeper
Clear_Input Variable	= 3.9.7	`azzeramento variabile ingresso

Message_Number	= 3.10	'numero di messaggio
Message_Pointer	= 3.12	'puntatore display messaggio
Output_Variable	= 3.13	'variabile uscita display
Output_Variable_Pointer	= 3.17	'puntatore display variabile
Output_Variable_Decimal	= 3.18	'cifre decimali variabile

//

```
LD      F.1
MOVASC  M.510    |PROVA 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.24
AND     F.=
MOV2    M.500    K.0
```

//

```
END
```



```

\/////////////////////////////////////////////////////////////////
\                Programma 2 dimostrativo Terminale Slave PTD1B                \
\                - Elenco Messaggi -                \
\/////////////////////////////////////////////////////////////////

```

`Caratteri ASCII riconosciuti

```

32  >
33  >!
34  >"
35  >#
36  >$
37  >%
38  >&
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 >}