

# Interfacciamo la TI-57

*I4IBR, Marco Ibridi*

*"Interfacciare" è certamente uno dei verbi più in voga in questo periodo in cui l'informatica galoppa verso mete pressoché inimmaginabili, vuoi a cavallo di magnifici puledri IBM o altri, vuoi in sella a splendidi Apple, Pet o Trs-80 ma anche sulle tastiere di straordinarie programmabili dalle possibilità fantascientifiche solo fino a qualche anno fa. Ciò che io voglio ora proporre è la possibilità di dare, per l'appunto, una voce nuova a una magnifica programmabile quale è la TI57 della Texas Instruments.*

*Detta programmabile è infatti il cavallo di battaglia di parecchie persone che, più o meno passionate di elettronica "vulgaris", vogliono scoprire cosa si nasconde dietro questo appassionante mondo dell'"intelligenza artificiale" e che allo stesso tempo non vogliono impegnare cifre considerevoli.*

*Il prezzo infatti è ciò che strizza più l'occholino ma non ci dobbiamo fermare qui, la '57 è un vero, seppur piccolo, computer.*

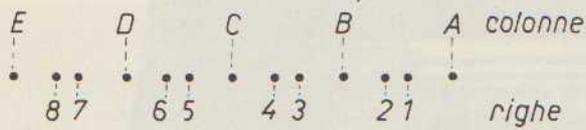
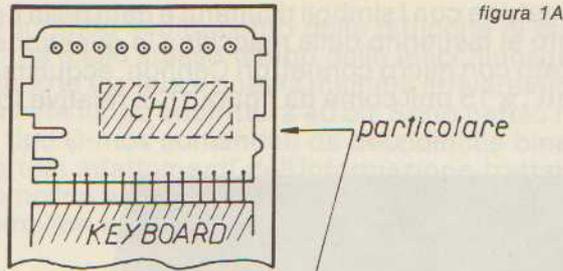
*Presto però tutti, chi prima chi poi, ci si accorge con rammarico dei limiti che detta macchina ci impone e allora cominciano i pensieri.*

*C'è chi pensa subito di passare alle sorelle maggiori e chi invece, come me, decide di guardarci un po' "dentro".*

*Appaiono così, un po' su tutte le riviste del ramo, vari suggerimenti e varie modifiche, ma quasi tutte si limitano a usarla né più né meno come timer intelligente. Certamente io non ho scoperto l'America, chissà a quanti prima di me sarà venuta l'idea, ma ho tentato di "nobilitarla" dandole la possibilità di colloquiare, o meglio "ascoltare" il mondo esterno.*

*Di darle un 'bus' più o meno standard nemmeno a pensarlo e allora la classica lampadina: perché non interfacciare la tastiera? Da qui ogni sorta di immondo pensiero, dalla possibilità di registrare i programmi su nastro magnetico, per i più bravi e i più zelanti, o su banda perforata, per i più pigri come me, e via sulla strada dell'implementazione di giochi o cose più utili. Quello che vi presenterò, più che un progetto finito e a sé stante è una **strada nuova** su cui sbizzarrirvi con la vostra splendida '57.*

*I contatti della tastiera sono perfettamente visibili nello spaccato di figura 1A e sono "quantificabili" nelle lettere da me assegnate, in modo assolutamente arbitrario, in calce a detta figura.*



**facile  
per tutti**

figura 1B

		colonne				
		A	B	C	D	E
righe	4	2nd	INV	lnX	CE	CLR
	2	LRN	X $\approx$ 1	x2	yx	1/x
	6	SST	STO	RCL	SUM	YX
	5	BST	EE	(	)	+
	7	GTO	7	8	9	X
	8	SBR	4	5	6	-
	1	RST	1	2	3	+
	3	RIS	0	.	+/-	=



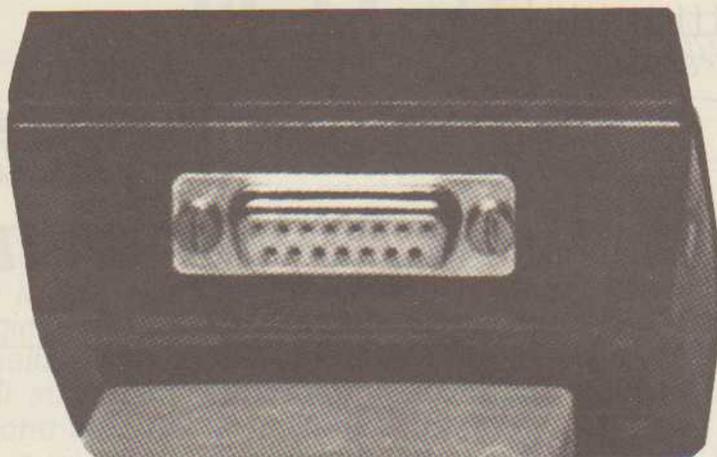
microconnettore (femmina) 15SB "Cannon" 15 poli

- 1 = 1
- 2 = 2
- 3 = 3
- 4 = 4
- 5 = 5
- 6 = 6
- 7 = 7
- 8 = 8
- 9 = n.c.
- 10 = n.c.
- 11 = A
- 12 = B
- 13 = C
- 14 = D
- 15 = E

figura 1C

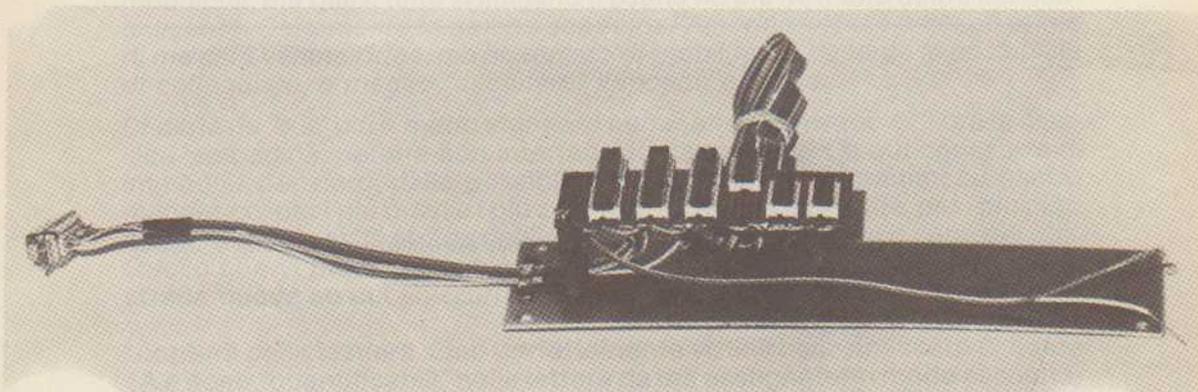
Collegamenti consigliati per l'utilizzo del connettore "Cannon"

La relazione che lega detta matrice con i simboli risultanti è data dalla figura 1B, che si rifà per l'appunto al tastierino della macchina in questione; il tutto elegantemente realizzato con micro connettori Cannon, acquistabili alla GBC e abbastanza "salati", a 15 poli come da figura 1C e relative foto.



A questo punto la domanda: cosa ne facciamo? Ai più bravi nessuna risposta, o hanno già snobbato tutto quanto o hanno già deciso cosa fare, agli altri, forse non meno bravi ma più curiosi, illustrerò parte della mia mirabolante costruzione.

**L'idea base è quella di far corrispondere a ogni tasto un ben determinato byte di sette bit**, otto nel caso si inserisca un particolare bit di controllo, in modo da poter memorizzare o manipolare il dato con relativa facilità. Infatti, se osserviamo di nuovo la figura 1B notiamo che, per ottenere, ad esempio, il carattere '8' si deve congiungere la riga n. 7 con la colonna 'C', come pure per ottenere la funzione 'LRN' la riga n. 2 con la colonna 'A'.



Ci si rende subito conto delle difficoltà di manipolare i dati sotto questa forma e allo stesso tempo della macchinosità della cosa, in quanto utilizzando la tastiera a questa maniera arriveremmo al limite della praticità con un byte lungo addirittura 40 bit. Sono perciò ricorso a switches elettronici di tipo C-mos comandati da decodifiche binario-decimale attraverso opportuni adattamenti dell'informazione trattata. Semplice eh? ma se osservate lo schema di figura 2 tutto sarà ancora più semplice e chiaro!

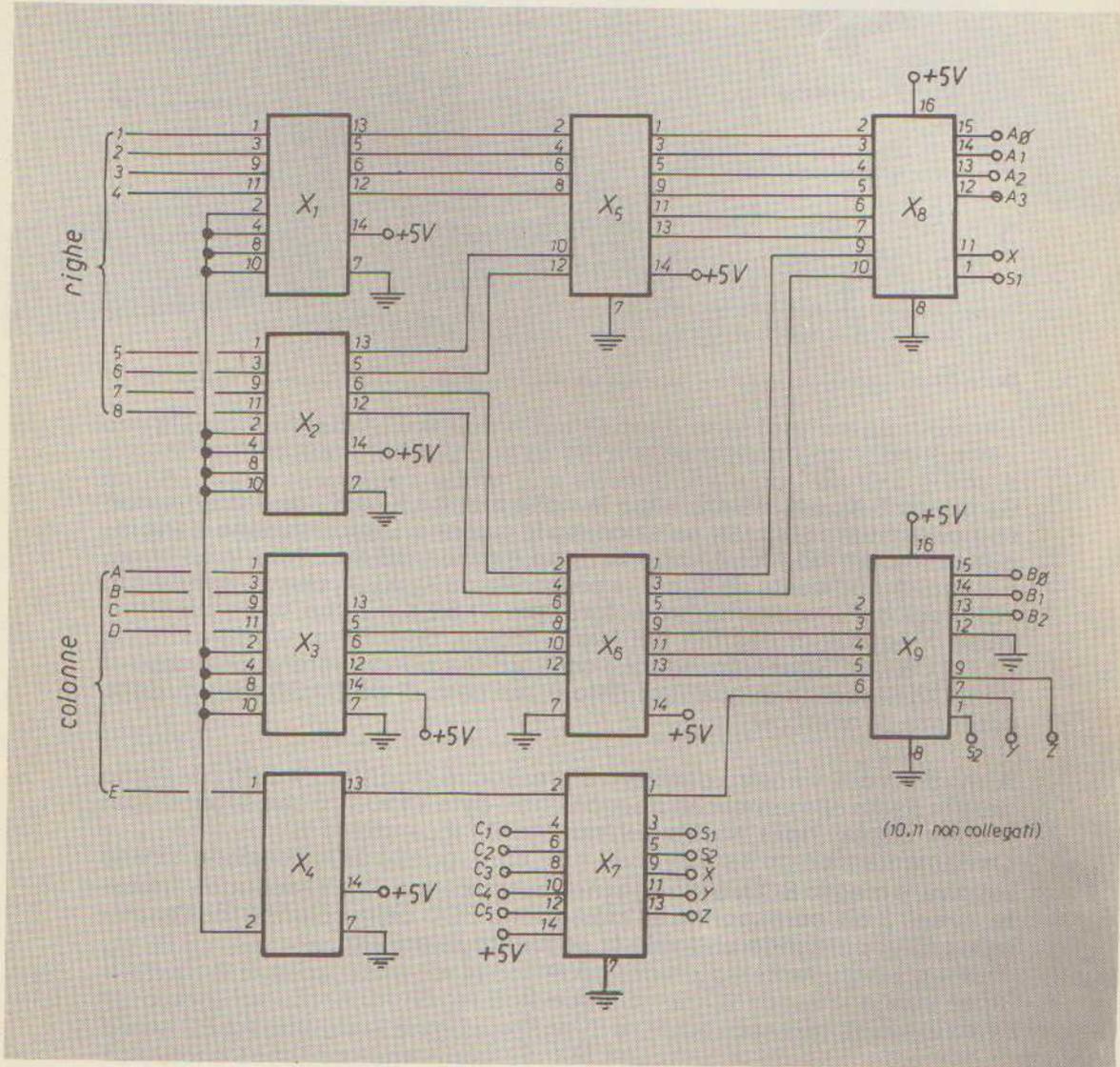
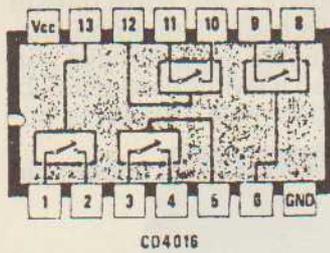


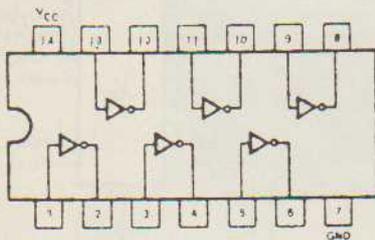
figura 2  
Zoccolature degli integrati a pagina seguente.

Interfacciamo la TI-57



CD4016  
 $X_1, X_2, X_3, X_4$

SN 7404 HEX INVERTER



$X_5, X_6, X_7$

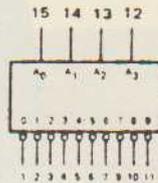
CONNECTION DIAGRAM  
DIP (TOP VIEW)



SN 7442

$X_8, X_9$

SN 7442  
LOGIC SYMBOL



$V_{CC} = \text{Pin } 16$   
 $GND = \text{Pin } 8$

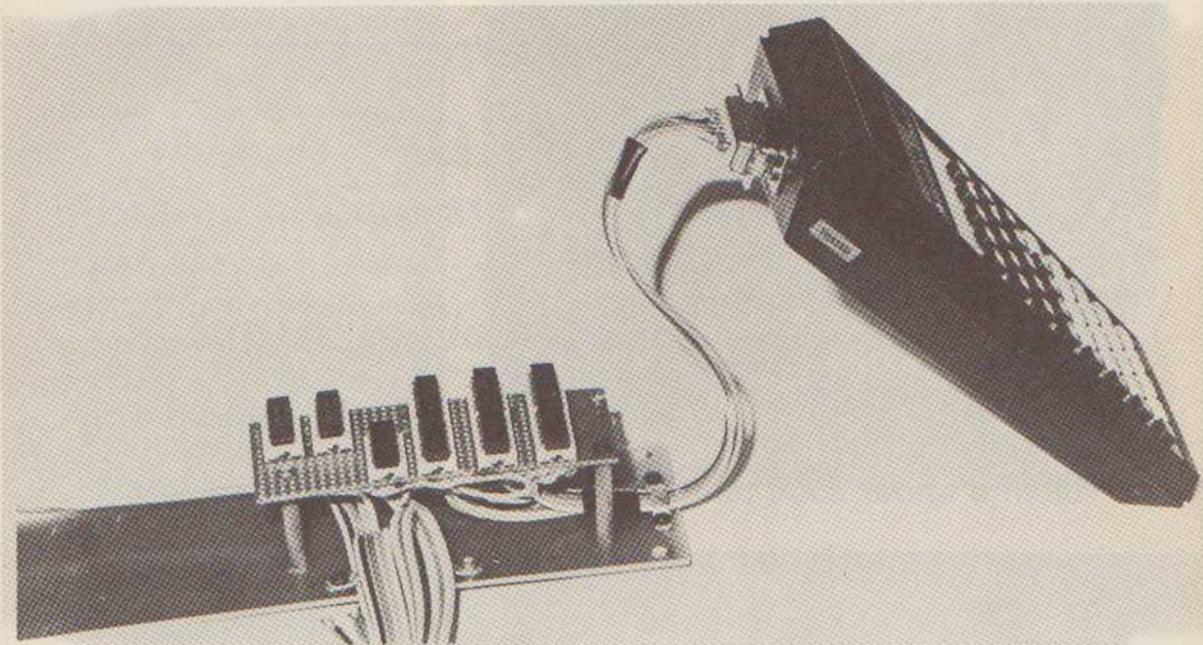
segue figura 2  
Zoccolature degli integrati.

Gli integrati  $X_1$  e  $X_2$  selezionano le righe mentre  $X_3$  e  $X_4$ , ovviamente non completamente utilizzati, selezionano le colonne. L'informazione viene iniettata dai pin  $A_0, A_1, A_2, A_3$  per le righe, e dai pin  $B_0, B_1, B_2$  per le colonne; è quindi decodificata da  $X_8$  e  $X_9$  e preparata da  $X_5, X_6, X_7$  che ne inverte il livello logico. Possiamo inoltre notare i pin  $S_1, S_2, X, Y, Z$  che, transitati attraverso  $X_7$  e quindi ribaltati nel livello logico, diventano rispettivamente  $C_1, C_2, C_3, C_4, C_5$  per comandare piccoli led che ci indicheranno, nel caso di  $C_1$  e  $C_2$ , l'assenza di qualsiasi dato e, nel caso di  $C_3, C_4, C_5$  attivati, della presenza di overflow.

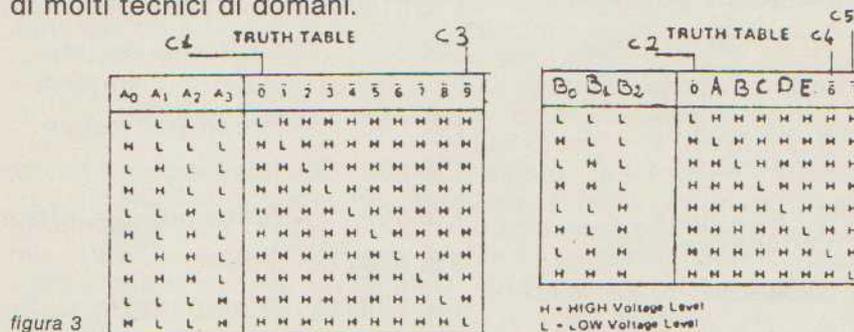
Arrivati sin qui, il nostro obiettivo è pressoché raggiunto: infatti possiamo dare in pasto alle fameliche decodifiche il byte 1110110 che subito apparirà nel display della nostra beneamata '57 il carattere '8'. Ovviamente il primo semibyte da 4 bit corrisponde all'indicazione binaria relativa, o meglio BCD, della riga (nell'esempio la n. 7) e il secondo semibyte, di soli 3 bit, corrisponde all'indicazione BCD della colonna (nell'esempio la 'C') ricordando che vale la seguente relazione:

$A = 1$   
 $B = 2$   
 $C = 3$   
 $D = 4$   
 $E = 5$

Ora entra in gioco la vostra fantasia e tutta l'ingegnosità di cui disponete; io personalmente ho utilizzato il byte così ottenuto per registrare i programmi su banda perforata leggendo l'informazione con comuni fotoresistenze



che pilotavano i relativi transistori, ma non ci si deve fermare qui: volendo si può trasformare l'informazione da parallela a seriale, registrare quindi i dati su nastro magnetico, addirittura trasmetterli e così via sino al limite della nostra immaginazione: l'importante è avere la coscienza di fare qualcosa per restare al passo di questo sviluppo tecnologico travolgente e di essere entrati in un mondo che ancora a molti fa paura, ma che sarà il pane di molti tecnici di domani.



Un ultimo sguardo ora alla figura 3 che rappresenta la tavola della verità del byte di ingresso utile per la codificazione dei caratteri usati e per l'utilizzo degli indicatori C1 (Ø nel primo semibyte), C2 (Ø nel secondo semibyte), C3 (overflow nel primo semibyte), e C4,C5 (due condizioni di overflow per il secondo semibyte).

Dalle foto, inoltre, per le quali va un particolare ringraziamento all'amico **Claudio Gatti**, notiamo i particolari del montaggio dell'interfaccia e della modifica alla TI57. \*\*\*\*\*