

Byte nell'etere

di Fabio Marzocca

La programmazione delle Eprom per i manipolatori automatici CW

Abbiamo visto, nella precedente puntata di "Byte nell'etere", come si può automatizzare completamente una stazione radio RTTY con l'ausilio di un computer e di un buon programma di gestione. Rimanendo in tema di controlli automatici, questo mese vedremo come realizzare una procedura software per la programmazione delle Eprom dei manipolatori automatici CW.

Il termine CW, nel settore radiotelegrafico, sta ad indicare una trasmissione radiotelegrafica con codice Morse (fig. 1). Questo famoso codice di punti e linee, ideato da Samuel Morse nel 1843, è caratterizzato dalla sua "temporizzazione relativa"; in pratica ciò significa che nel Morse è importante soltanto il rapporto relativo di durata fra un punto ed una linea, mentre non esistono regole di temporizzazione fissa come nel Baudot o in altri codici.

Inoltre il Morse è un codice a lunghezza variabile, fondato sulla frequenza delle lettere dell'alfabeto nella lingua inglese, per cui lettere più frequenti, sono state codificate con sequenze più corte.

Una trasmissione CW viene realizzata con l'attivazione-esclusione della portante RF del trasmettitore, al ritmo del codice. Il corrispondente in ricezione rivelerà il segnale sottoponendolo a battimento in media frequenza, per ottenere una nota intermittente nella banda audio.

Il manipolatore CW

Il manipolatore CW è un dispositivo elettronico che, collegato all'opportuna presa sul trasmettitore, genera un messaggio automatico in CW, con ripetizione ciclica. In commercio ne esistono di diversi tipi, ma lo schema di principio è essenzialmente lo stesso, come riportato in figura 2.

Un circuito di clock variabile, non necessariamente a quarzo, stabilisce la cadenza dell'emissione dei simboli e quindi la velocità di trasmissione.

Questo oscillatore è collegato ad un address decoder il quale, ad ogni fronte di salita del clock, incrementa di una unità l'indirizzo presentato sulla Eprom. Sul data bus dell'Eprom saranno quindi disponibili otto messaggi (uno per ogni bit del data bus).

Con un selettore ad otto posizioni, che potrà essere sia meccanico che elettronico, sarà quindi possibile scegliere il messaggio desiderato, ed inviarlo poi al transistor switch, prima di entrare nella presa "cw-

key" del trasmettitore. A questo punto il manipolatore inizierà a trasmettere il suo messaggio ciclicamente leggendo di seguito il contenuto della memoria.

L'impiego di questo dispositivo trova la sua maggiore applicazione per i radiofari (beacon) o per i ponti radio, dove è appunto richiesta la trasmissione di un messaggio di identificazione con il QTH della stazione.

Ma anche nella stazione del radioamatore è possibile applicare il manipolatore CW.

In particolare, su gamme di frequenza con scarso traffico (UHF, SHF), il manipolatore potrà trasmettere ciclicamente il messaggio di chiamata della stazione, per interrompersi automaticamente, tramite un dispositivo di controllo sul ricevitore, qualora giungesse risposta alla chiamata.

La programmazione dell'Eprom

Prenderemo in esame una Eprom tipo 2716, ma i discorsi sono lo stesso validi anche per 2732 o altri tipi.

Il contenuto di una singola cella di memoria, viene indirizzato e collegato sulla base di un transistor di commutazione. Se il bit è 1, il transistor satura, se è 0 interdice. Fin qui tutto bene: sarà perciò solo necessario stabilire i rapporti di durata fra il punto e la linea, e le separazioni fra simboli, lettere e parole.

Esiste già in tal senso una normativa; considerando come unità la durata del punto, le temporizzazioni sono regolate dalla seguente tabella:

Simbolo o intervallo	Durata
Punto	1
Linea	3
Separazione punto/linea	1

A	• _	J	• _ _ _	S	• • •
B	_ • • •	K	_ • _	T	_
C	_ • _ •	L	• _ • •	U	• • _
D	_ • •	M	_ _	V	• • • _
E	•	N	_ •	W	• _ _ _
F	• • _ •	O	_ _ _	X	_ • • _
G	_ _ •	P	• _ _ •	Y	_ • _ _
H	• • • •	Q	_ _ • _	Z	_ _ • •
I	• •	R	• _ •	1	• _ _ _ _
2	• • _ _ _	3	• • • _ _	4	• • • • _
5	• • • • •	6	_ • • • •	7	_ _ • • •
8	_ _ _ • •	9	_ _ _ _ •	0	_ _ _ _ _

Figura 1
Alfabeto
Morse.

Separazione fra caratteri 3

Separazione fra parole 9

Questo significa che per rappresentare una linea, nella memoria dovranno essere registrati tre "1" consecutivi, mentre per separare due parole, si dovranno registrare nove "0" consecutivi. Ciò è anche intuitivo in quanto il trasmettitore non eccita la portante se il bit di memoria è zero, ma solo quando legge un uno.

Vediamo quindi come sarà codificata nella Eprom la chiamata "CQ". Dalla figura 1 ricaviamo:

C •••••
Q •••••

perciò:

C 1 1 1 0 1 0 1 1 1 0 1

separatore 0 0 0

Q 1 1 1 0 1 1 1 0 1 0 1 1 1

seguiti da una serie di nove "0" per separare dalla successiva parola. Complessivamente, perciò, si ha un'occupazione di 36 bit di memoria.

Durante la programmazione di questa Eprom, bisognerà però tener conto del fatto che ognuno degli otto messaggi va scritto sempre sullo stesso ordine di bit, nei vari byte di memoria. In particolare il messaggio numero 4 dovrà essere scritto nel quarto bit di ogni byte.

Dato che la programmazione dell'Eprom avviene per byte, nella fase di preparazione del file da registrare, bisognerà far attenzione a mascherare i restanti sette bit, mentre si scrive uno dei messaggi.

In tal modo, una volta indirizzato il byte di memoria, la Eprom presenterà in parallelo gli otto messaggi che contiene, e sarà il selettore a scegliere quale dovrà andare al trasmettitore. Sarà inoltre possibile realizzare un selettore elettronico che al termine di un messaggio commuti automaticamente sul successivo. Così facendo si avranno a

```

1
LIST
30 HIMEM: 32766
100 DIM EP$(90):D$ = CHR$(4)
110 FOR I = 46 TO 90: READ EP$(I): NEXT
150 FOR I = 0 TO 58
160 READ A: POKE (768 + I),A: NEXT I
500 D$ = ""
510 DIM H(20),V(20)
520 GOTO 5000
4900 TEXT : HOME : INVERSE : FOR I = 1 TO 40: PRINT "@": NEXT : FOR I =
1 TO 8: PRINT "@": HTAB 40: PRINT "@": VTAB I: NEXT : FOR I = 1 TO
40: PRINT "@": NEXT : NORMAL : RETURN
5000 ONERR GOTO 7040
5030 GOSUB 4900: VTAB 3: HTAB 9: PRINT "PROGRAMMA CODIFICA EPROM": VTAB 6
: HTAB 13: PRINT "FABIO MARZOCCA": VTAB 4: HTAB 15: PRINT "- 8 CANALI
5040 GOSUB 5100
5050 END
5100 PRINT : PRINT : PRINT : PRINT : PRINT
5120 PRINT "SCEGLI:"
5130 PRINT : PRINT
5140 HTAB 5: PRINT " 1 CREARE UN MESSAGGIO"
5150 PRINT : HTAB 5: PRINT " 2 CODIFICARE GRUPPO MESSAGG."
5160 PRINT : HTAB 5: PRINT " 3 VERIFICARE FILE BINARIO"
5170 PRINT : HTAB 5: PRINT " 4 REGISTRARE EPROM"
5180 PRINT : HTAB 5: PRINT " 5 USCIRE DAL PROGRAMMA"
5190 PRINT
5200 PRINT "INSERISCI 1,2,3,4 O 5":
5210 GET N: PRINT N
5220 IF N > 5 OR N < 1 THEN PRINT "": HOME : GOTO 5030
5230 ON N GOTO 5270,7480,8000,9000,6320
5240 REM
5250 REM
5260 REM
5270 REM CREA IL MESSAGGIO
5280 REM
5290 REM
5300 REM
5390 HOME
5400 AS = 32768
5410 M = 1
5420 I = 1
5430 GOSUB 6210
5450 GOSUB 6420
5460 GOTO 5030
6150 REM
6160 REM
6170 REM TITOLO
6180 REM
6190 REM
6200 REM
6210 TEXT : HOME
6220 VTAB 2
6230 PRINT TAB(13)"INSERISCI MESSAGGIO"
6240 PRINT
6250 PRINT TAB(13)"PER FINIRE PREMI RETURN"
6260 PRINT "-----"
6270 POKE 34,7
6280 RETURN
6290 REM
6300 REM
6310 REM
6320 HOME : END
6330 REM
6340 REM
6350 REM
6360 REM
6370 REM LEGGI LA TASTIERA
6390 REM
6400 REM
6410 REM
6420 VTAB 13: POKE 34,12
6430 CH = 0
6440 FOR J = 1 TO 20:H(J) = 1: NEXT
3
1100 CH = CH - 2: IF CH < = 0 THEN CH = CO
1110 IF PEEK (32767 + I) < > 141 THEN 7140
1120 ONERR GOTO 7150
1130 IF PEEK (32767 + I) = 141 THEN PRINT " ": VTAB V(20): HTAB H(20):
FOR J = 19 TO 1 STEP -1:H(J + 1) = H(J):V(J + 1) = V(J): NEXT : PRINT
CHR$(222):: PRINT CHR$(136):
RETURN
1140
1150 ONERR GOTO 7180
1160 PRINT " ": PRINT CHR$(136): VTAB PEEK (37) - 1
1170 GOTO 7140
1180 TEXT : HOME
1190 VTAB 10: PRINT TAB(6)"SEI TORNATO TROPPO INDIETRO"
1200 PRINT ""
1210 FOR I = 1 TO 2500: NEXT : RUN
12450 REM
12455 REM
12470 REM CODIFICA MESSAGGIO
12480 HOME : FOR I = 32768 TO 34816: POKE I,0: NEXT
HOME : VTAB 10: PRINT "INSERISCI IL NOME DEL FILE DI TESTO": VTAB 11
: PRINT "DA CODIFICARE": INPUT F$
VTAB 15: PRINT "RIPEITTORE (R) O BEACON (B)? ": GET RB$: IF RB$ < >
"R" AND RB$ < > "B" THEN 7510
7512 GOSUB 12000: POKE (805),(2 ^ (CN - 1))
7515 ST = 32767:LU = 0:B = 0
7517 HOME : VTAB 11: HTAB 15: INVERSE : FLASH : PRINT "ATTENDERE"
7519 NORMAL
7520 PRINT D$"OPEN ":F$: PRINT D$"READ ":F$
7522 GET A$: IF A$ = CHR$(32) THEN FOR I = 1 TO 9: POKE (804), PEEK (S
+ LU + I): CALL 778: POKE (ST + LU + I), PEEK (804): NEXT I:LU = LU
+ 9: GOTO 7522
7523 GOTO 7531
7530 GET A$
7531 IF A$ = CHR$(13) THEN 7560
7537 IF A$ = CHR$(32) THEN FOR I = 1 TO (6 + B): POKE (804), PEEK (ST +
LU + I): CALL 778: POKE (ST + LU + I), PEEK (804): NEXT I:LU = LU + 6
+ B:B = 3: GOTO 7530
7538 B = 0
7540 L = LEN (EP$(ASC (A$))): FOR I = 1 TO L:M = VAL ( MID$( EP$( ASC (
A$)),I,1))
FOR K = 1 TO M: POKE (804), PEEK (ST + LU + K): CALL 768: POKE (ST +
LU + K), PEEK (804): NEXT K:LU = LU + M: POKE (804), PEEK (ST + LU +
1): CALL 778: POKE (ST + LU + 1), PEEK (804):LU = LU + 1: NEXT I
POKE 804, PEEK (ST + LU + 1): CALL 778: POKE (ST + LU + 1), PEEK (80
4): POKE (804), PEEK (ST + LU + 2): CALL 778: POKE (ST + LU + 2), PEEK
(804):LU = LU + 2
7550 GOTO 7530
7560 PRINT CHR$(1): PRINT D$"CLOSE ":F$
7565 RB = 778: IF RB$ = "B" THEN RB = 768
7570 FOR I = (ST + LU + 1) TO 34816: POKE (804), PEEK (I): CALL RB: POKE
I, PEEK (804): NEXT
7575 HOME : INVERSE : PRINT "IL FILE CODIFICATO E' STATO GENERATO ": PRINT
"DALL'INDIRIZZO $8000 A $8800 (2K) "
7576 NORMAL : VTAB 10: PRINT "ALTRI MESSAGGI (S/N)? ": GET A$: IF A$ = "S
" THEN PRINT A$: GOTO 7500
7580 NORMAL : VTAB 10: PRINT "INSERISCI IL NOME DEL FILE BINARIO ": INPUT
"PER QUESTA SERIE DI MESSAGGI":FB$
7590 PRINT D$"BSAVE":FB$:BIN,A32768,L2048"
7600 PRINT : GOTO 5000
7950 REM
7960 REM
7979 REM VERIFICA FILE BINARIO
7990 REM
8000 HOME : VTAB 11: PRINT "INSERISCI IL NOME DEL FILE": PRINT "DA VERIFI
CARE ": INPUT F$
8010 PRINT D$"LOAD ":F$:BIN,A$8000"
8012 GOSUB 12000: POKE (805),(2 ^ (CN - 1))
8020 HOME : PRINT "IL FILE ":F$:".BIN N,":CN:" CONTIENE": PRINT "(ESC P
ER USCIRE)": VTAB 5
8025 ST = 32767:L = 1
8030 K$ = "":K = 0:V = 0:J = 0
8031 FOR I = 0 TO 8: POKE (804), PEEK (ST + I + L): CALL 806:J = J + PEEK
(804): NEXT I: IF J = 0 THEN PRINT CHR$(32): CHR$(222): CHR$(136
)::L = L + 9: GOTO 8031

```

Figura 3 - Listato del programma EPROMCW.

```

2
6450 X = PEEK (49152); REM
6460 IF X < 127 THEN 6450
6470 POKE 49168,0
6480 IF X = 141 THEN 6480
6490 PRINT " "; PRINT CHR$ (136);
6500 IF X < 141 THEN 6590
6510 CO = CHICH = 0
6520 FOR J = 1 TO 19
6530 H(J) = H(J + 1)
6540 IF V(20) > 22 THEN 6560
6550 V(J) = V(J + 1)
6560 NEXT
6570 H(20) = PEEK (36) + 1; V(20) = PEEK (37) + 1
6580 IF V(20) = 24 THEN V(20) = 23
6590 CH = CH + 1
6600 PRINT CHR$ (X);
6610 NORMAL
6620 PRINT CHR$ (222);
6630 PRINT CHR$ (136);
6640 IF X = 136 THEN GOSUB 7080; GOTO 6450
6650 POKE (32767 + I),X
6660 I = I + 1
6670 GOTO 6450
6680 AE = 32767 + I
6685 POKE AE,141
6692 PRINT
6890 TEXT : HOME : VTAB 10
6900 VTAB 12: PRINT "INSERISCI IL NOME DI QUESTO ": VTAB 13: PRINT "FILE
DI TESTO: "; INPUT F$
6920 D$ = CHR$ (4): PRINT
6960 PRINT D$; "OPEN "; F$: PRINT D$; "WRITE "; F$
6961 FOR K = 32768 TO AE: PRINT CHR$ (PEEK (K));: NEXT
6962 PRINT
6965 PRINT D$; "CLOSE "; F$
6970 RETURN
6980 REM
7000 REM
7010 REM -MESSAGGIO D'ERRORE
7020 REM
7030 REM
7040 HOME : PRINT " ": VTAB 10: INVERSE : PRINT "SI E' VERIFICATO UN ERROR
E: RIPETI "
7050 GET BH$
7070 RUN
7080 REM VERIFICA BACKSPACE
7090 I = I - 1

```

disposizione tutti i 16 Kbyte della 2716 per un unico messaggio.

In figura 6 è rappresentato un esempio di una Eprom programmata con soli 4 messaggi. Ovviamente nelle caselle sono riportate le lettere in chiaro per facilitare l'esempio, ma nella realtà queste saranno sostituite dalle sequenze di 1 e 0, come visto sopra.

Il programma EPROMCW

In figura 3 è riportato il listato del programma EPROMCW, scritto per Apple II, ma dopo quanto detto finora, non dovrebbero esserci problemi ad organizzare un programma simile per altri computer.

Lo scopo del programma è di creare una zona di memoria da 2Kbyte (\$8000 - \$8800) nella quale verranno codificati gli 8 messaggi desiderati; quest'area di memoria potrà essere controllata e decodificata, prima di essere trasferita in un'Eprom tipo 2716.

Lanciando il programma e scegliendo l'opzione 1 del menu principale, si avrà la possibilità di inserire uno dei file di testo che costituiranno i messaggi dell'Eprom. Alla fine del messaggio, il computer richiede il nome con cui salvare il file di testo e ripropone il menu.

Si consiglia di inserire inizialmente su un disco dati, tutti i testi dei messaggi che dovranno essere successivamente registrati sull'Eprom, e quindi passare alla fase di codifica.

Con l'opzione 2, il programma passa alla fase di codifica. Vengono richiesti i nomi dei testi da codificare ed il rispettivo canale su cui dovranno essere registrati. Non è necessario disporre otto messaggi sulla Eprom; nel caso occorresse registrarne meno di otto, gli altri canali saranno automaticamente riempiti con 0.

Prima della codifica, viene chiesto se l'Eprom sarà impiegata su un beacon o su un ripetitore. La differenza sta nel riempimento degli spazi vuoti dopo il messaggio: nel caso di beacon, la memoria restante dopo il messaggio, sarà riempita da una serie di 1, nel caso di ripetitore, da 0.

Questo perché in un beacon, funzionando il manipolatore ad interruzione di portante, occorre che, al termine del messaggio, la portante rimanga inserita; in un ponte ripetitore, invece, la portante deve necessariamente sganciare per permettere gli accessi agli utenti.

Dopo aver codificato tutta l'area di memoria che dovrà essere copiata sull'Eprom, si potrà verificarne il contenuto tramite l'opzione 3. Questa opzione può essere utilizzata anche per rileggere un file binario codificato su disco precedentemente, e del quale non se ne ricorda il contenuto.

Per registrare l'Eprom, occorrerà scegliere l'opzione 4. Il computer richiederà in quale slot è inserito il programmatore di Eprom, e successivamente il controllo passerà a quest'ultimo.

La routine di codifica

La corrispondenza fra punti/linee dell'alfabeto Morse e la codifica elettronica sulla Eprom, può essere risolta associando ai relativi caratteri dell'alfabeto, una stringa composta dai simboli "1" e "3", che stanno ad indicare, rispettivamente, punto e linea.

Come visto in precedenza, infatti, una linea è codificata come un punto lungo tre volte, quindi se nella Eprom per il punto si registra un bit 1, per una linea se ne registreranno tre in sequenza.

La tabella dei simboli è rappresentata dal vettore EPS(i), con i variabile da 46 a 90 per comprendere simboli, lettere e numeri. Nelle linee da 10000 a 10020, sono contenute le stringhe di codifica associate ad ogni carattere.

In questo modo, letto il carattere da codificare, il programma ricaverà la stringa associata (es. C=3131) e scriverà in memoria quattro stringhe di 1 della lunghezza specificata, ognuna separata dall'altra da uno 0. Alla fine del carattere, verranno registrati tre 0 per segnalare l'inizio di un nuovo carattere. Quindi il carattere C diverrà in memoria, come già visto: 11101011101000

In figura 5 è riportato il flow-chart della routine di codifica messaggio (linee 7470-7600).

La routine parte con l'azzeramento

0300-	AD 24 03	LDA	#0324
0303-	0D 25 03	ORA	#0325
0306-	8D 24 03	STA	#0324
0309-	60	RTS	
030A-	A9 FF	LDA	##FF
030C-	4D 25 03	EOR	#0325
030F-	8D 25 03	STA	#0325
0312-	AD 24 03	LDA	#0324
0315-	2D 25 03	AND	#0325
0318-	8D 24 03	STA	#0324
031B-	A9 FF	LDA	##FF
031D-	4D 25 03	EOR	#0325
0320-	8D 25 03	STA	#0325
0323-	60	RTS	
0324-	00	BRK	
0325-	00	BRK	
0326-	AD 24 03	LDA	#0324
0329-	EA	NOP	
032A-	2D 25 03	AND	#0325
032D-	F0 06	BEQ	#0335
032F-	A9 01	LDA	##01
0331-	8D 24 03	STA	#0324
0334-	60	RTS	
0335-	A9 00	LDA	##00
0337-	8D 24 03	STA	#0324
033A-	60	RTS	
033B-	00	BRK	

Figura 4 - Breve routine in linguaggio macchina contenuta nel programma EPROMCW.

completo dell'area di memoria da \$8000 a \$8800 e quindi richiede il nome del primo file di testo da codificare con il relativo canale di registrazione.

A questo punto viene creata la maschera di selezione bit in relazione al numero di canale del messaggio. Se CN è il numero di canale, il bit da non mascherare sarà dato da:

$$\text{bit} = 2^{(\text{CN}-1)}$$

Successivamente il programma legge un carattere, e verifica se il primo è un carattere di space, nel qual caso registra nove 0 consecutivi.

Da qui inizia il loop della routine che consiste nella lettura di un carattere, verifica se è l'ultimo carattere del messaggio — CHR\$(13) —, verifica se è uno space, oppure scrittura del codice corrispondente.

Dato che alla fine di ogni carattere vengono registrati tre 0, per gli space incontrati all'interno del messaggio si dovranno aggiungere solo sei 0, per averne così nove in totale, come richiesto dalla tabella delle temporizzazioni che abbiamo visto in precedenza.

Si è voluto dare un maggior risalto nel dettaglio della routine di codifica, sia perché essa rappresenta il cuore del programma, sia per dare la possibilità ad utenti di altri computer, di scriverne una propria versione in un Basic diverso dall'Applesoft.

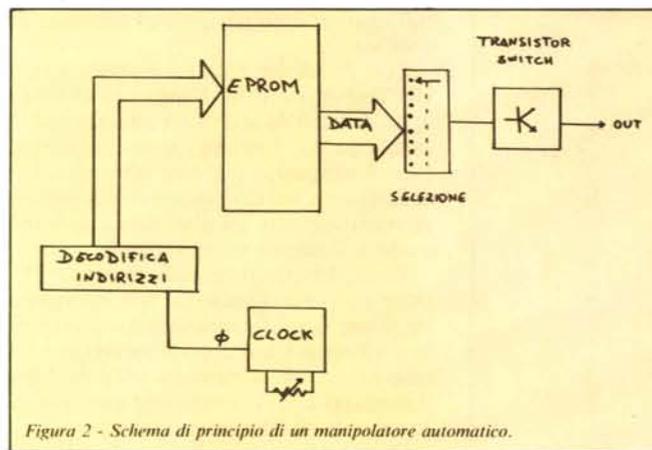


Figura 2 - Schema di principio di un manipolatore automatico.

DATA								ADDRESS
7	6	5	4	3	2	1	0	
V	C	I	A					000
V	Q	Ø	T					001
V		A	N					002
	C							003
D	Q	J	A					004
E		N	T					005
	D	G	N					006
I	E	I						...
								...
								...

Figura 6 Esempio parziale di una Eprom programmata con 4 messaggi.

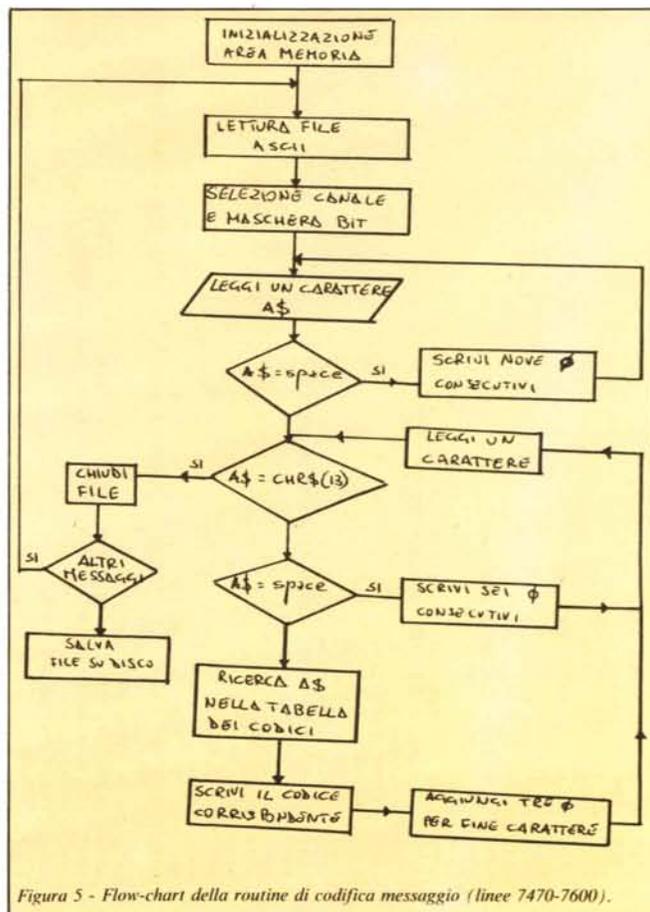


Figura 5 - Flow-chart della routine di codifica messaggio (linee 7470-7600).

2ª MOSTRA DI «HI-FI ESOTERICO»

Sono presenti tutte le
maggiori aziende
del settore,
che presentano i marchi
più prestigiosi
con dimostrazioni e prove
acustiche in ampie sale.

**31
OTTOBRE
3
NOVEMBRE
1985**

STEREOMANIA



6ª RASSEGNA ESPOSITIVA DI APPARECCHIATURE
HI-FI, MUSICA, COMPONENTI AUDIO

BOLOGNA:
PALAZZO DEI
CONGRESSI
(quartiere fiera)

Orario mostra
10.00-20.00

Per informazioni:
Segreteria mostra

PROMO EXPO

Via Barberia, 22
40123 Bologna
Tel.: 051/333657



SALONE DELLA VIDEOREGISTRAZIONE, MICRO
COMPUTER, TV COLOR, HI-FI, TECNICA VIDEO