

Software

SHARP PC-1500

Code Editor

di Giacomo Borlizzi - Pisa

Il programma proposto da Borlizzi è essenzialmente un'utility da impiegare in fase di correzione dei programmi in linguaggio macchina: consente di trasferire il contenuto di locazioni di memoria in linee di "DATA".

L'utente a questo punto potrà effettuare la correzione dei codici e successivamente, tramite le istruzioni READ e RESTORE, riposizionare in memoria il programma I/m

Dopo aver caricato il programma di figura 1, inserire una serie di linee non etichettate del tipo:

```
350 DATA &00,&00,&00,.....&00
```

È necessario che ogni linea termini con uno &00, pena una possibile macchiatura dell'area Basic.

Per facilitare l'operazione di immissione delle linee DATA, si può assegnare ad un tasto definibile la frase (,&00' e quindi digitare DEF S seguito dall'indirizzo di partenza della lettura, quello finale, ed il numero di linea.

Ad esempio, inserendo &C000, &C005 e &350, al termine dell'elaborazione nella linea 350 appariranno i primi 5 codici della ROM.

Dopo le eventuali modifiche, si potrà far girare ad esempio il seguente

```
10 'S' INPUT 'L.IN' ; T, 'L.FI' ; A, 'LINEA' ; R : WAIT
20 M = STATUS 2 - STATUS 1
30 IF R = 16*16*PEEK M + PEEK(M+1) THEN 60
40 M = M+3+PEEK(M+2) : IF M<STATUS 2-2 THEN 30
50 PRINT 'NO LINEA' : END
55 IF M >= STATUS 2-2 THEN 50
60 K = M+2 : IF 16*16*PEEK(K+1)+PEEK(K+2)<>61837 PRINT 'NO DATA' : END
70 M = M+3+PEEK(M+2)
80 K = K+4 : IF K>M-3 THEN 55
90 IF T>A PRINT 'FINE' : END
100 Q = PEEK T/16 : P = INT Q : Q = 16*(Q-P)
110 GOSUB 120 : POKE K,P : P=Q : GOSUB 120 : POKE(K+1),P : T=T+1 : GOT0 80
120 P=P+48 : IF P>57 LET P=P+7
130 RETURN
```

Figura 1 - Programma Code Editor.

programma che legge dai DATA e scrive in memoria:

```
200 'D' INPUT "L.IN";A, "L.FI"; B, "LINEA";
C
210 RESTORE C : FOR K=A TO B
220 READ Q: POKE K,Q: NEXT K: END
```

In tale programma non c'è un controllo tra il numero di locazioni fornite e i codici da leggere, pertanto se le locazioni sono in numero inferiore alla differenza tra i primi due valori forniti, sarà segnalato un errore.

Vediamo una descrizione del programma Code Editor:

linea 10: chiede la prima e l'ultima locazione da copiare, segue la richiesta del numero di linea dal quale dovrà iniziare la copia dei codici voluti;

linee 20-50: ricerca della linea vista sopra; la 50 visualizza una eventuale non presenza di tale linea;

linea 55: controlla se eventuali linee successive a quella data esistono;

linea 60: controlla se nella suddetta linea esiste 'DATA' come prima istruzione;

linea 70: calcola la successiva linea a quella fornita; se non esiste tale linea o non esiste il DATA iniziale, verrà visualizzato 'NO LINEA' o 'NO DATA';

linea 80: calcola il punto della linea nella quale mettere il codice;

linea 90: segnala che il programma ha fatto il suo dovere;

linea 100-130: lettura della locazione da copiare e inserimento (previa conversione) del codice letto al punto opportuno nella linea contenente il 'DATA'.

Graphic Editor

di Fulvio Peruggi - Qualiano (NA)

La creazione di nuovi caratteri grafici o di disegni da rappresentare sul display del PC-1500 causa in generale una notevole perdita di tempo con fogli di carta millimetrata, bozze piene di quadratini anneriti, ed i conti associati alle conversioni binario/decimale/esadecimale necessarie per trovare i codici da inserire nei nostri futuri GPRINT.

Questo approccio viene immediatamente considerato "preistorico" non appena si usa il Graphic Editor. L'idea alla base di questo programma è che il posto migliore per progettare le nostre opere è evidentemente... il display stesso. I problemi da risolvere erano: 1) accedere velocemente ai singoli pixel dal Basic, e 2) realizzare un programma interattivo col vincolo di non cancellare mai il prezioso contenuto del display. Il primo è stato risolto con un uso massiccio delle funzioni logiche, ed il secondo realizzando l'acquisizione di tutti i comandi mediante l'istruzione INKEYS.

Lanciato il Graphic Editor, compare il suo cursore, ovvero un pixel lampeggiante che inizialmente si trova in alto e a sinistra del display. Esso può essere spostato in tutte le direzioni premendo i tasti <↑>, <↓>, <▶>, <◀>. Per disegnare basta portare il cursore sui pixel che interessano ed accenderli (se sono spenti) o spegnerli (se sono già stati accesi) premendo I (per "Invert"). Una volta realizzato un progetto, nel caso si sia insoddisfatti dei risultati ottenuti, si può premere R ("Reset") ottenendo la cancellazione del display e il ritorno del cursore in

```

1:REM Graphic Editor
10:CLS: CLEAR :Z$="0123456789ABCDEF"
20:X=X-(A$=CHR$ 8)+(A$=CHR$ 12):Y=Y-(A$=CHR$ 11)+(A$=CHR$ 10)
30:X=X-((X=156)-(X=-1))*156:Y=Y-((Y=7)-(Y=-1))*7
40:A=POINT X:B=2^Y:C=AOR B:D=A AND B:E=C-D
50:WAIT 9:GCURSOR X:GPRINT E
60:WAIT 0:GCURSOR X:GPRINT A
70:A$=INKEY$:IF A$<CHR$ 32THEN 20
80:IF A$="I"
    GCURSOR X:GPRINT E:BEEP 1,1,30:GOTO 40
90:IF A$="D"OR A$="H"GOSUB 140
100:IF A$>CHR$ 47 AND A$<CHR$ 58 LET S=2+VAL A$:I=220/S<=X+1:A$="C":GRAPH :GOSUB 140
110:IF A$="R"THEN 10
120:IF A$="E"WAIT :GCURSOR X:GPRINT A:END
130:GOTO 40
140:FOR J=0TO X:A=POINT J:GOSUB A$:NEXT J
150:IF A$="C"GLCURSOR (0,-(X-6)*I+7)*S):TEXT
160:LPRINT :RETURN
170:"D":LPRINT STR$ A;"";:RETURN
180:"H":M=INT (A/6):N=A-M*16
190:LPRINT MID$ (Z$,M+1,1);MID$ (Z$,N+1,1);:RETURN
200:"C":IF A=0 RETURN
210:FOR K=0TO 6:B=2^K:C=AAND B:IF C=0THEN 250
220:FOR L=0TO S-1:Z=L+J*S:T=Z:U=-K*S:V=U-S:U=-W:U=U
230:IF 1LET W=Z:Z=150+U:T=150+U:
240:LINE (Z,U)-(T,U),0:NEXT L
250:NEXT K:RETURN
STATUS 1
    
```

Figura 2 - Programma Graphic Editor.

alto a sinistra. Se invece il risultato è soddisfacente si può ottenere la stampa dei corrispondenti codici decimali o esadecimali da usare con GPRINT per ricreare in altra occasione il disegno, e se ne può anche ottenere la hard copy con scelta fra 10 fattori di scala. In ogni caso le stampe prodotte saranno relative alla zona del display che va dall'estrema sinistra fino alla posizione del cursore: si dovrà quindi spostare preventivamente quest'ultimo

sulla colonnina di pixel più opportuna. Per far stampare i codici decimali è necessario premere il tasto D, per quelli esadecimali serve il tasto H, mentre per la hard copy si deve premere un tasto numerico. In quest'ultimo caso, il fattore di scala più piccolo corrisponde al tasto 0 e quello più grande al 9: se la scala selezionata non permette di copiare l'intero disegno in senso orizzontale, si ottiene automaticamente una stampa in verticale (rota-

zione di 90° in senso orario). Alla fine di ogni stampa, il cursore ricompare sul display e si possono dare altri comandi o modificare ancora il proprio progetto. Al termine del lavoro, la pressione del tasto E ("End") provoca l'arresto del programma facendo permanere il contenuto del display.

Struttura del programma

- 10 Inizializzazione
- 20 Spostamento del cursore
- 30 Condizioni toroidali al contorno (se il cursore esce da un lato del display rientra da quello opposto)
- 40 ÷ 60 Visualizzazione del cursore (variare il WAIT nella linea 50 per regolare la frequenza di lampeggiamento)
- 70 Accettazione comandi e diramazione allo spostamento del cursore
- 80 Inversione dello stato del pixel puntato dal cursore, con conferma acustica dell'esecuzione
- 90 Lancio della subroutine di gestione stampa (codici numerici)
- 100 Lancio e inizializzazione della subroutine di gestione stampa (hard copy)
- 110 Reset
- 120 Fine programma
- 130 Riciclaggio
- 140 ÷ 160 Gestione stampa
- 170 Stampa dei codici decimali
- 180 ÷ 190 Stampa dei codici esadecimali
- 200 ÷ 250 Hard copy

Configurazioni della RAM utente
 Senza moduli aggiuntivi essa va dall'indirizzo &4000 all'indirizzo &47FF nel PC-1500 e da &4000 a &57FF nel PC-1500A (area RESERVE compresa). Il modulo CE-151 si fa rilocare dal microprocessore nei 4 Kb successivi alla RAM utente originale. Il modulo CE-155 si fa rilocare disponendo 2 Kb prima e 6 Kb dopo la RAM utente originale. Il modulo CE-161 fa indirizzare i suoi 16 Kb da &0000 a &3FFF.

Caricamento di routine in linguaggio macchina - Oltre al comando: CLOAD M "etichetta" che provoca il caricamento di una routine dalla locazione &XXXX alla locazione &YYYY in cui si trovava al momento del salvataggio fatto con CSAVE M "etichetta"; &XXXX, &YYYY si può anche dare il comando: CLOAD M "etichetta"; &ZZZZ che provoca il caricamento a partire da una locazione prescelta &ZZZZ. (Le etichette sono facoltative). **MC**

α	30404830484400	τ	04043C442400	Ω	4C52610161524C
β	7E29292E1000	υ	0C1078140800	00	
γ	023448340200	κ	08442810284420	Ξ	63415D495D4163
δ	324D493100	00		00	
ε	2854544400	00		Alfabeto greco	
ζ	01414955552300	ω	30484030404830	Hard copy	
η	04380404784000	00		(Scala: 0)	
θ	3E49493E00	Γ	417F4101010300	Codici esadecimali	
ι	3C40402000	Δ	40704C434C7040	(Le lettere simili	
κ	7C1028444400	00		a quelle latine e	
λ	41221C204000	Θ	3E414949413E00	quelle poco usate	
μ	7C20201C2000	00		sono omesse)	
ν	047C20100C00	Α	40700C030C7040	Nel Graphic Editor	
ξ	01414155552B00	00		la linea 150 e'	
ο	7824241800	Π	417F4101417F41	stata modificata	
σ	3844444C340400	00		per l'occasione:	
		Σ	41635549416300		
		00		150:IF A\$="C"	
		00		GLCURSOR (0,-(
		00		(X-6)*I+7)*S):	
		00		TEXT:LCURSOR	
		00		4:A\$="H":GOTO	
		00		140	

Figura 3 - Esempio di caratteri grafici.