Morphy

di Riccardo e Corrado Tesio - Cuneo

Si tratta di un gioco che utilizza le sprite generabili con il Basic residente, secondo le modalità esaminate il mese scorso in questa rubrica. La procedura da seguire per caricare il programma è la seguente:

1) Caricate il programma generatore di file pubblicato il mese scorso e che riportiamo per coloro che non avessero il n. 33 della rivista. Dopo avere dato il RUN inserite i valori 5,15.

2) Alla richiesta del programma inserite nel registratore la cassetta su cui volete incidere il programma; seguendo la procedura indicata sullo schermo registrate sulla cassetta lo "pseudogramma" prodotto dal generatore di file.

```
10 REM*                    *
20 REM* GENERATORE FILES *
30 REM*                    *
40 REM* PER MODIFICARE I *
50 REM*                    *
60 REM* REGISTRI DEL VDP *
70 REM*                    *
80 REM* (C) BY TESIOWARE *
90 REM*                    *
100 CALL CLEAR
110 INPUT "REGISTRO(0-7),DATO(0-255)?  ":
R,D
120 A=18429-(256*R+D)
130 X$=CHR$(0)
140 OPEN #1:"CS1",OUTPUT,FIXED
150 PRINT #1:X$&X$&X$&X$&X$&X$&X$&CHR$(INT(A
/256))&CHR$(A)
```

3) Registrate di seguito sulla cassetta il programma "Morphy". Ogni volta che vorrete usare il programma dovrete:

4) Disconnettere ogni dispositivo attaccato al computer tramite la porta laterale.

5) Resettare la macchina tramite QUIT o BYE

6) Caricare il primo programma sulla cassetta (quello cortissimo)

7) La macchina al termine della procedura di caricamento non restituirà il cursore ma dopo un po' di tempo lo schermo cambierà repentinamente più volte colore per poi stabilizzarsi sul nero a righe verticali bianche. A questo punto battete un solo carattere e poi premete ENTER; dovrebbe ritornare quasi tutto normale e comparire il messaggio "MEMORY FULL". Se qualcosa non funziona ripetete la procedura a partire dal punto 4.

```
10 CALL CHAR(144,"D")
20 OPTION BASE 1
30 RANDOMIZE
40 DIM H$(32),F$(6),C$(7)
50 CALL CLEAR
60 READ A,B
70 FOR I=A TO A+B
80 READ A$
90 CALL CHAR(I,A$)
100 NEXT I
110 IF A THEN 60
120 FOR I=2 TO 32 STEP 2
130 A$=SEG$("0123456789ABCDEF",I/2,1)
140 H$(I-1)=A$&"0"
150 H$(I)=A$&"8"
160 NEXT I
170 FOR I=1 TO 10
180 SC(I)=1000
190 N$(I)="M O R P H Y"
200 NEXT I
210 FOR I=1 TO 6
220 READ F$(I)
230 NEXT I
240 FOR I=1 TO 7
250 READ C$(I),A$
260 C$(I)=C$(I)&A$
270 NEXT I
280 GOSUB 1210
290 IF B THEN 360
300 GOSUB 980
310 IF B THEN 360
320 GOSUB 1040
330 IF B THEN 360
340 GOSUB 1110
350 IF B=0 THEN 280
360 CALL SCREEN(14)
370 CALL CHAR(145,"D")
380 S=0
390 O=3
400 R=0
410 X=24
420 Y=23
430 K=X
440 C=4
450 P=0
460 E=400
465 F=0
470 Z=0
480 B=3
490 FOR A=1 TO 9
500 Q(A)=INT(B+2*RND)
510 B=2*(Q(A)=1)-2*(Q(A)=6)-5*(Q(A)(4)-(Q(A))3)
520 NEXT A
530 Q(10)=7
540 FOR A=1 TO 12
550 CALL COLOR(A,1,1)
560 NEXT A
570 B=16
580 FOR A=23 TO 9 STEP -1
590 T=4*RND)1
600 B=B+T
610 CALL HCHAR(A,3,106,B)
620 CALL HCHAR(A,B+3,36+T*(1+RND))
630 NEXT A
640 PRINT "jjjjjsvquuvqtjjj#":"svquq_w______qtjj#":"`aba``a_^__]^__jjj$":"defgfe
g_r__rr_)jjj$"
650 PRINT "ky__ikhijkijjkijjs":"jky_tswquuqqvuquw_":"jjky__`abaa`w_____":"jjjky_
egfged_____":"jjjjkhhhhhhhhm_hhhhhhhhhhhhh";
660 T=13.0411
670 GOSUB 1440
680 FOR A=10 TO 30
690 CALL SOUND(-100,55*A,8)
700 NEXT A
```

```
710 GOTO 4250
720 DATA 34,7,C0F0F0F0F0F8F8FE,80C0C0C080DCFCFE,80C0E0E0E0C0C0C,0101030707777FFF
,070F0F0F1F3F3FFF,0303070707070F07
730 DATA 0C52272717D2B01,40C0201C1E0F0F07,6D,2,60106808D2252418,FFFFFFFFFFFFFFFF
,,91,18,3C18D88D7E3C183C,00000080E3FFFFFF,,,
740 DATA FFFF7FFE76561C1C,FF882F2B23230202,FFDFFF5E5C1C1818,1818181818181818,181
858585C547EFF,10109094945B5FFF
750 DATA 1818181C3D3575FF,1011111535378FFF,FFFFFFDF49,FEFAF0E0A08,,7F3F2F070501,
FFFFFFFAF0D0C08,FFFFFF7F2F0701
760 DATA 112,13,00808185A585B7F7,FF75756424242020,181818383C7C7EFF,FEFEFAF8E8A88
88,FF7F5F5F1B090101
770 DATA DF9D951514141404,FFFF7D5C5C585858,5858484808080808,,F0F0F0F0FFFFFFFF,0F
0F0F0FFFFFFFFF
780 DATA 00384492A2924438,00008081,00010281,128,13,7CD4D7FFFC382433
790 DATA 7CD4D7FFFC38684C,3E2BEBFF3F1C24CC,3E2BEBFF3F1C1632,0C0C9FF93F79DF1E,4C6
838FCFFD7D47C,78FB9EFC9FF9303
800 DATA 1C3E68497F2A141C,6010680850222418,0018242320D2120C,002828104A322418,001
8181048332418,3C3C7EA5A5A5A5A5
810 DATA 3C3C7EA5A5995A5A,153,2,0F09CD81FF7C187C,103810547C7C287C,FEAAFE7C381010
7C,0,-1,E00F,E10F,E20F,E30F,EC0D,ED0D
820 DATA jj__vvuvquqqvuququuqquvutjjjjt__ww_w__bab``___]__^w_pjjjj_a`ab`___c=dcc
__(r__r_)jsvjj_=c=cc_zlc=hcclhijkijkis_wj
830 DATA j_=c=ffzisc==dfquuquuqqu___jjp=c=_zij(fgg_____________pjjjedgzijjhhhhm_
lhy_((_(__Cjjjjkhhijjjjjjjj_jjkhhhhhhhijj
840 DATA js__tjjsvuuquqjjvvuqjjsqvvtjs_`a_vv_w_____qv_w__uq__ww_t_bc=bww__zlhh__
w_____________
850 DATA pcc=cb`_zijsw_____^_^]______jfdgdffzijjC__((_(r_rr__(__(jkhhhhhijjjkhhh
hhijhjjkhm_li
860 DATA suqqvtsuquqqus_tvququuququj_____________]_ba`a_]`ba`___ujC_________(r_e=
cg_red=c__(_jhmy__lm__)ijkh=c)lm_j
870 DATA jjky_tv__vququ=cqjvud=ctqu_tjjjky_w]^w_]__ef_jw__ef_^___jjjjky_rr__r(__
__jC_((__r__Cjjjjjkijjkijkhhhijkhhhhij_hi
880 DATA squqvtjjjjuvq__tjsvjjvuvuqvj_`baw_quuq_w_`bau_wuqw_w__wj_cc=`_]_]^_a`=c
=a____^_____(j
890 DATA Cfc=f_r_rr_edgfeg((_(r__zhijkmdglijhjjkhhhhhhhhhijkhijjj
900 DATA jsuqququuqvquququuqvquuqu_tjs_)____^_____]^__ba`abab`a_t__lmy_(r_____rr
__f=cgdefc=`a__qvkhijk)__ijjkhh
910 DATA efzhy_d=c=____ujsq___]_ququ__zijky_=c=____(jC_((_r(_____zijjjkyefgkm_li
jkhhhijkhhhhhijjjjjkhhi
920 DATA suqququqvvuqquuquvquuqus_tv`b__C____ww____]^_w_____^__wccalhmy____`a__
rr_______r__(cc=svtky___c=zijjkmy_lhij_licc=bw_jkm)_chiuquv
930 DATA jky_tjs_vvcc=ca_quuv_c=w___wqjky_q__wwcc=c=_____dg__^___jjky____pfdefg
((_(r_((__r__Cjjjky___jkhhhhhhhijkhhm_jkhijjjjkhhij
940 DATA js__tjjjsuqvvs_tjjjjjjjsv_tjs____uvq___ww___quvquuv_w__v_____w________
___w___w____w
950 DATA khhm\\\\\\\\\lm\\\\\\\\\\lhi
960 DATA 409,508,606,301
970 DATA 4470A70F5470A10F,6470AD0F7470A50F,4488AF0F5488860F,6488A50F7488820F,
980 CALL SCREEN(12)
990 C=2.09
1000 GOSUB 1670
1010 PRINT TAB(10);"ISTRUZIONI":::"RIPORTA ALLA LUCE I 3 CIMELI"::"DELLA CIVILT
A' PERDUTA DI"::"KHNOS."::"USA:"
1020 PRINT :" S  SINISTRA"::" D  DESTRA"::" E  SALTARE O SALIRE"::" Q  BUTTARE C
IBO"::" P  PUNTEGGIO"::::
1030 GOTO 1720
1040 CALL SCREEN(14)
1050 PRINT TAB(10);"CLASSIFICA"
1060 FOR I=1 TO 10
1070 PRINT :SC(I);TAB(10);N$(I)
1080 NEXT I
1090 PRINT ::
1100 GOTO 1720
1110 CALL SCREEN(5)
1120 CALL COLOR(2,9,16)
1130 CALL COLOR(8,13,16)
1140 CALL COLOR(16,11,16)
1150 C=5.16
1160 GOSUB 1670
1170 PRINT TAB(8);"TABELLA  PUNTI":::"   )   COSCIOTTO":::"   C   ANFORA    200
 PTI"
1180 PRINT ::"   ";CHR$(154);"   CORONA    800  PTI":::"   ";CHR$(155);"   CALIC
E    800  PTI":::"   ";
1190 PRINT CHR$(153);"   LAMPADA   800  PTI":::"   (   FRUTTO    150  CAL":::
1200 GOTO 1720
1210 CALL SCREEN(2)
1220 C=8.02
1230 GOSUB 1670
1240 CALL COLOR(13,14,1)
1250 CALL COLOR(14,1,1)
1260 CALL HCHAR(4,1,136,224)
1270 FOR I=6 TO 8
1280 CALL HCHAR(I,5,32,24)
1290 NEXT I
1300 A$="M  O  R  P  H  Y"
1310 FOR I=1 TO 16
1320 CALL HCHAR(7,8+I,131+I-4*INT(I/4))
1330 CALL SOUND(-400,348-8*I,6)
1340 CALL HCHAR(7,8+I,ASC(SEG$(A$,I,1)))
1350 NEXT I
1360 C=INT(12*RND+3)
1370 CALL COLOR(14,C,C)
1380 FOR I=6 TO 30
1390 CALL SOUND(-100,110,I,165,I,220,I)
1400 NEXT I
1410 A$="A9( 1984 TESIOWARE"
```

(continua a pag. 154)

8) Digitate NEW; tutto dovrebbe tornare normale.

9) Caricate il programma "Morphy"; se durante il caricamento vedeste apparire sullo schermo strani caratteri di vari colori non preoccupatevi. A questo punto date il RUN e buon divertimento.

Chi possiede la Mini Memory può evitare tutta questa trafila aggiungendo al programma:

15 CALL PEEKV (-753,X)

Il gioco consiste nel dirigere Morphy attraverso le impervie caverne alle pendici del monte Rocciascura al fine di recuperare tre preziosi cimeli della perduta civiltà di Khnos. La riuscita della missione è minacciata dalle voracissime piante carnivore che infestano la grotta e dagli imprevedibili spruzzi di acqua bollente che sgorgano da protuberanze della roccia. Inoltre la caverna è abitata da Griper, un curioso essere che è innocuo quando è chiuso nel suo guscio (potete camminarci sopra) ma che è letale quando è aperto.

Attenti al livello di energia che cala rapidamente: 400 unità all'inizio, -2 ogni passo, -10 ogni salto. Attenti anche a non saltare contro gli ostacoli (muri o stalattiti) per non rimediare rumorose zuccate o anche la perdita di una vita. Nelle grotte trovate: anfore che rendono 200 punti; frutti che rendono 150 unità di energia; cosciotti che serviranno all'ultima schermata.

Giunti al fondo delle grotte si accede ad un'ampia caverna, parzialmente allagata, che custodisce sul fondale tre tesori. Dovete portarli in superficie uno alla volta, evitando la grossa piovra che li sorveglia. Potete tenerla occupata lanciando i cosciotti raccolti in precedenza: essa li mangerà sospendendo momentaneamente l'inseguimento. Anche in acqua l'energia si consuma: -2 per ogni movimento, -10 lanciando un cosciotto.

Una volta recuperati i tre cimeli riceverete un bonus in base all'energia avanzata; poi il gioco ricomincerà da capo.

I tasti da utilizzare sono:

S per andare a sinistra

D per andare a destra

E per saltare (se si vuole saltare in una determinata direzione bisogna aver preso almeno un passo di rincorsa)

P per vedere i valori di punteggio, energia, il numero di cosciotti raccolti e il numero di vite rimaste

Q per lanciare i cosciotti.

Il gioco è estremamente ben curato sia nella grafica che nella realizzazione. Il per-

Questo programma è disponibile su cassetta presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 126.

sonaggio di Morphy e la piovra sono realizzati mediante sprite. Abbiamo trovato il gioco piuttosto difficile e, specie all'inizio, è assai facile perdere le quattro vite che si hanno a disposizione. Dobbiamo confessare che per poter provare il gioco fino all'ultimo quadro, dovendo verificare il corretto funzionamento del programma, abbiamo dovuto alterare l'istruzione che contava le "vite" (linea 2390), così da disporre di omini a volontà senza dover ricominciare ogni volta da capo. Vi possiamo assicurare però che una volta presa un po' di pratica il gioco risulta assai avvincente, anche grazie all'ottima grafica.

La sequenza delle caverne generata dal calcolatore non è sempre la stessa, ma varia da partita a partita; se quindi vi trovate in difficoltà e non riuscite a superare il primo livello perché gli ostacoli sono troppo difficili per voi, provate a ridare il RUN.

Il programma occupa 11369 byte: per limitare l'occupazione di memoria alcune linee superano la lunghezza di 4 righe di schermo; per inserirle bisogna digitare caratteri fino a riempire completamente le quattro righe, premere ENTER, richiamare la linea con EDIT ed aggiungere quello che è rimasto.

Per aumentare la velocità del programma, l'autore ha anche posto le parti che devono essere eseguite più rapidamente in fondo al programma affinché vengano trovate prima da GOTO e IF...THEN. Si tratta di una tecnica di cui prendiamo atto senza tuttavia essere certi dell'utilità (in effetti è possibile che l'interprete Basic in presenza di un salto cominci a scandire il programma dal fondo alla ricerca della linea a cui trasferire il controllo).

Variabili principali

H$(32) Vettore che contiene i valori esadecimali delle coordinate dello schermo (per avere corrispondenza diretta tra posizione degli sprite e caratteri)

F$(6) Valori esadecimali delle forme dei personaggi

SC(10) I 10 migliori punteggi

N$(10) Chi li ha realizzati

C$(7) Vettore che contiene la forma delle grotte

Q(10) Vettore che contiene la successione delle forme delle grotte

X,Y Coordinate di Morphy

S Punteggio

E Energia

O Numero omini

R Numero cosciotti

Analisi del listato

10 - 270 Inizializzazione: definizione caratteri, caricamento vettori

280 - 350 successione degli schermi, classifica, istruzioni, eccetera, finché non viene premuto un tasto

360 - 470 inizio partita

(segue da pag. 153)

```
1420 GOSUB 1610
1430 GOTO 1720
1440 A=100*(T-INT(T))
1450 B=100*(A-INT(A))
1460 CALL COLOR(1,T,B)
1480 CALL COLOR(8,6,A)
1490 CALL COLOR(9,A,2)
1500 CALL COLOR(10,B,T)
1510 CALL COLOR(11,T,A)
1520 CALL COLOR(12,B,A)
1530 CALL COLOR(13,16,A)
1540 CALL COLOR(14,2,A)
1550 CALL COLOR(16,11,T)
1560 CALL SCREEN(T)
1570 FOR B=2 TO 7
1580 CALL COLOR(B,2,A)
1590 NEXT B
1600 RETURN
1610 T=ASC(A$)-48
1620 Z=ASC(SEG$(A$,2,1))-51
1630 FOR I=3 TO LEN(A$)
1640 CALL HCHAR(T,Z+I,ASC(SEG$(A$,I,1)))
1650 NEXT I
1660 RETURN
1670 A=100*(C-INT(C))
1680 FOR B=1 TO 12
1690 CALL COLOR(B,C,A)
1700 NEXT B
1710 RETURN
1720 PRINT " PREMI UN TASTO PER INIZIARE";
1730 FOR I=0 TO 99
1740 CALL KEY(3,A,B)
1750 IF B THEN 1770
1760 NEXT I
1770 CALL CLEAR
1780 CALL VCHAR(1,31,31,96)
1790 RETURN
1800 B=4
1810 C=25
1820 A$="JJJJJJJJJJJJJJJJJJJJJJJJJJJJ"
1830 CALL HCHAR(Y,X,130-D)
1840 CALL CHAR(144,"D")
1850 FOR A=1 TO 19
1860 F=INT(-2*RND*(B(12))
1870 T=-INT(-2*RND*(C)17))
1880 PRINT SEG$(A$,1,B);CHR$(36-F*(1+RND));TAB(C);CHR$(39+T*(1+RND));SEG$(A$,1,2
8-C)
1890 B=B+F
1900 C=C+T
1910 NEXT A
1920 PRINT A$;
1930 FOR A=4 TO 6
1940 CALL HCHAR(24,B+A,149+A)
1950 NEXT A
1960 CALL HCHAR(3,X,95)
1970 F=0
1990 A=16
2000 B=23
2010 W=0
2020 GOTO 3760
2030 IF E<0 THEN 2340
2040 CALL SOUND(-80,-1,E)
2050 FOR I=20 TO E STEP -.05
2060 NEXT I
2070 IF P=11 THEN 3850 ELSE 4380
2080 CALL CHAR(144,H$(Y-1)&H$(X)&"E40D")
2090 FOR B=0 TO 2
2100 FOR A=136 TO 139
2110 CALL HCHAR(Y,X,A)
2120 CALL SOUND(-400,250-A,0)
2130 NEXT A
2140 NEXT B
2150 CALL HCHAR(Y,X,60)
2160 GOTO 2340
2170 A$=H$(Y)&H$(X)&F$(3-C)
2180 B$=H$(INT(B))&H$(INT(A))
2190 FOR C=1E3 TO 8E3 STEP 1E3
2200 I=I=0
2210 CALL CHAR(144,B$&F$(6+I)&A$)
2220 CALL SOUND(-600,C,4)
2230 NEXT C
2240 GOTO 2340
2250 X=X-D
2260 Y=Y+1
2270 A$=H$(Y-1)&H$(X)&"E"
2280 RESTORE 960
2290 FOR A=550 TO 220 STEP -110
2300 READ B$
2310 CALL SOUND(-400,A,6)
2320 CALL CHAR(144,A$&B$)
2330 NEXT A
2340 CALL CHAR(144,H$(Y-1)&H$(X)&"E70F")
2350 FOR A=0 TO 30
2360 CALL SOUND(-200,110,A,-5,A)
```

```
2370 NEXT A
2390 O=O-1
2400 IF O<0 THEN 2460
2405 IF F=3 THEN 2910
2408 E=400
2410 IF P=11 THEN 1990
2420 Y=23
2430 X=K
2440 C=3
2450 GOTO 4250
2460 RESTORE 970
2470 FOR I=144 TO 147
2480 READ A$
2490 CALL CHAR(I+1,"0")
2500 CALL CHAR(I,A$)
2510 NEXT I
2520 FOR I=21 TO 23
2530 CALL HCHAR(I,5,95,24)
2540 NEXT I
2550 A$="F6PUNTEGGIO_TOTALE:_"&STR$(S)
2560 GOSUB 1610
2570 GOSUB 1730
2580 CALL CHAR(144,"0")
2590 FOR I=1 TO 10
2600 IF S>SC(I)THEN 2630
2610 NEXT I
2620 GOTO 350
2630 CALL SCREEN(11)
2640 C=2.01
2650 GOSUB 1670
2660 PRINT "BRAVO!":"SEI ";STR$(I);") NELLA CLASSIFICA":"DI OGGI: SCRIVI IL TUO
NOME:"
2670 FOR A=9 TO I STEP -1
2680 N$(A+1)=N$(A)
2690 SC(A+1)=SC(A)
2700 NEXT A
2710 INPUT A$
2720 N$(I)=SEG$(A$,1,16)
2730 SC(I)=S
2740 GOSUB 1770
2750 GOSUB 1040
2760 GOTO 350
2770 W=0
2780 Z=Z=0
2790 IF Z THEN 2820
2800 CALL CHAR(62,"819999D8FFD87E3C")
2810 GOTO 4290
2820 CALL CHAR(62,"003C7EFFFFD87E3C")
2830 GOTO 4290
2840 S=S+600
2850 FOR I=1 TO 1.4 STEP .1
2860 CALL SOUND(-100,220*I,4)
2870 CALL SOUND(-100,292*I,4)
2880 NEXT I
2890 CALL HCHAR(3,F+2,K)
2900 IF F<3 THEN 3760
2910 FOR I=2 TO 22
2920 CALL SOUND(-200,110*I,4,55*I,4)
2930 NEXT I
2940 CALL CHAR(144,"0")
2950 FOR A=20 TO 24
2960 CALL HCHAR(A,1,95,14)
2970 NEXT A
2980 A$="E3ENERGIA:"&STR$(E)
2990 GOSUB 1610
3000 A=E*10
3010 A$="G3BONUS:"&STR$(A)
3020 GOSUB 1610
3030 S=S+A
3040 FOR I=0 TO 9
3050 FOR A=9 TO 16
3060 CALL COLOR(16,A,2)
3070 CALL SOUND(-400,A*440,18)
3080 NEXT A
3090 NEXT I
3100 GOSUB 1770
3110 GOTO 410
3120 S=S+200
3130 F=F+1
3140 K=T
3150 FOR T=220 TO 660 STEP 55
3160 CALL SOUND(-100,T,4)
3170 NEXT T
3180 CALL HCHAR(Y,X,106)
3190 Y=Y-1
3200 GOTO 3780
3210 FOR A=660 TO 330 STEP -110
3220 CALL SOUND(-100,A,4)
3230 NEXT A
3240 FOR A=0 TO 30 STEP 2
3250 CALL SOUND(-100,660,A)
3260 NEXT A
3270 S=S+200
3280 CALL HCHAR(Y,X,95)
3290 GOTO 4290
3300 R=R+1
```

(continua a pag. 156)

480 - 530 caricamento vettore Q(10)
540 - 670 disegno della prima schermata
980 - 1790 subroutine varie
1800 - 1960 stampa dell'ultimo schermo
2030 - 2070 loop di ritardo quando l'energia è agli sgoccioli
2080 - 2160 mangiato dalla pianta carnivora
2170 - 2240 mangiato dalla piovra
2250 - 2450 perdita di una vita
2460 - 2760 movimento Griper
2840 - 3420 punteggi e bonus
3430 - 3510 getti d'acqua bollente
3600 - 4000 movimento omino e piovra
4010 - 4610 movimento omino nella grotta.

Modifiche al programma "Labirinto in 3-D"

Il programma "labirinto in 3-d", pubblicato sul numero 31 di MC del giugno scorso, ha riscosso un buon successo tra i nostri lettori; si tratta senza dubbio di un ottimo programma che tuttavia può essere oggetto di ulteriori miglioramenti. Per esempio prevede che l'orientamento dei tasti cursore per muoversi all'interno del labirinto sia quello assoluto, riferito alla mappa del labirinto e non ai corridoi visualizzati in quel momento sullo schermo. Come indicammo a suo tempo nell'articolo che accompagnava il programma, l'ostacolo principale per qualunque modifica di un certo rilievo era la quasi totale occupazione della memoria del computer durante l'esecuzione.

A suo tempo chiedemmo la vostra collaborazione, invitandovi a inviarci i vostri suggerimenti su come apportare migliorie al programma. Il nostro appello è stato recepito e diverse persone, una perfino dalla Yugoslavia, ci hanno scritto proponendoci le loro modifiche. Qui abbiamo raccolto le più interessanti, dovute all'opera di Arturo Campana di Bergamo e di Alessandro Forlani di Lodi.

Il primo punto è il seguente: modificare i DATA alle linee 2570-2750 sostituendo ciascuna costante numerica con un carattere il cui codice ASCII sia pari al valore della costante stessa. In queste linee DATA si incontrano solo i due valori: 32 (corridoio) e 96 (muro); a questi due valori corrispondono rispettivamente lo spazio bianco (ottenuto premendo la barra spaziatrice) e l'accento grave (ottenuto premendo FCTN C). Questi caratteri dovranno essere scritti uno di seguito all'altro senza interporre alcun separatore, cosicché la linea DATA risulterà contenere una sola costante di tipo stringa di caratteri. Per esempio la linea

```
2580 DATA 96,32,96,32...
```

diverrà

```
2580 DATA ' '... e così via.
```

Fatto ciò occorre modificare la lettura dei DATA. Noterete la presenza del nuovo vettore G$ in cui vengono memorizzate le stringhe. Tale scelta non è casuale ma serve a velocizzare in maniera drastica la stampa della mappa del labirinto.

Questa non è più ottenuta mediante 532 chiamate della procedura HCHAR, ma attraverso la stampa di 19 velocissime PRINT. Il tempo necessario per la stampa della mappa passa da circa mezzo minuto a meno di 5 secondi.

Grazie alla semplificazione dei DATA si ha una notevole riduzione della memoria occupata dal programma. Si passa infatti da liste di dati lunghe 83 caratteri, con un totale di $83 \times 19 = 1577$ byte, a liste di data lunghe 28 caratteri, cioè $28 \times 19 = 532$ byte.

Considerando poi il dimensionamento del vettore G$ e il suo riempimento (circa 560 byte), nonché le linee di listato modificate, si ha un risparmio globale di memoria di circa mezzo kbyte: abbastanza per poter apportare ulteriori migliorie.

A questo punto possiamo dedicarci alla seconda e più impegnativa modifica: la possibilità di riprogrammare i tasti delle frecce per muoversi riferendosi alla nostra visione sullo schermo e non a quella della piantina del labirinto vista dall'alto. Ad onore del vero a suo tempo noi affermammo che tale operazione avrebbe richiesto una quantità di memoria non indifferente; al contrario il lettore Forlani è riuscito ad ottenere un ottimo risultato soltanto aggiungendo pochissime linee e spostandone altre già presenti.

Modificando il programma nella maniera indicata si può ovviare anche ad altri inconvenienti: dopo la stampa del labirinto l'unico tasto utile per visualizzare di nuovo il corridoio è la barra spaziatrice, dopodiché ci si ritrova esattamente nello stesso punto rappresentato dal quadratino nero sulla cartina.

Ciò contrariamente a prima, quando premendo un tasto con le frecce oppure un altro tasto qualunque (cosa che ora equivale ad un passo in avanti), la situazione si modificava nel passaggio da piantina a corridoio, con ovvia perdita dell'orientamento appena acquistato. Inoltre non è neanche più necessario negli incroci ad L o a T andare a "sbattere contro il muro" per poter girare, ma è sufficiente portarsi all'altezza della svolta. Analogamente per i corridoi che si aprono ai lati di una stanza, non è più necessario che essi scompaiano completamente dal video per poterli imboccare.

Le linee che devono essere cancellate del tutto sono:
660, 690, 720, 750, 2460, 2470, 2480, 2490, 2500, 600, 601.

Nel listato trovate le linee da aggiungere, come vedete alcune linee sono state semplicemente spostate.

(segue da pag. 155)

```
3310 GOTO 3330
3320 E=E+150
3330 FOR A=0 TO 3
3340 CALL SOUND(-100,523,4)
3350 CALL SOUND(-100,440,4)
3360 NEXT A
3370 GOTO 3280
3380 CALL HCHAR(1,3,95,28)
3390 A$="13PTI_"&STR$(S)&"___EN._"&STR$(E)&"___)_"&STR$(R)
3400 GOSUB 1610
3410 CALL HCHAR(1,31-0,128,0)
3420 IF P=11 THEN 3780 ELSE 4290
3430 IF F THEN 3480
3440 F=INT(2*RND+1)
3450 CALL CHAR(92+F,"2856AABA1A90121")
3460 CALL SOUND(-4E3,-5,24)
3470 GOTO 4310
3480 CALL CHAR(92+F,"")
3490 CALL SOUND(-1,-6,30)
3500 F=0
3510 GOTO 4310
3520 IF T<0 THEN 2250
3530 Y=Y-T
3540 X=X-D
3550 CALL SOUND(-60,-3,4)
3560 GOTO 4070
3570 X=X-D
3580 CALL SOUND(-20,-2,8)
3590 GOTO 4270
3600 IF R<1 THEN 3780
3610 R=R-1
3620 C=X
3630 D=Y
3640 A=INT(A)
3650 B=INT(B)
3660 CALL SOUND(-4E3,-8,16)
3670 C=C+SGN(A-C)
3680 D=D+SGN(B-D)
3690 CALL CHAR(145,H$(D)&H$(C)&"8906D")
3700 IF C-A+D-B THEN 3670
3710 CALL SOUND(-10,880,10)
3720 CALL CHAR(145,"D")
3730 E=E-10
3740 W=14
3750 GOTO 3780
3760 K=0
3770 Y=5
3780 CALL KEY(0,T,T)
3790 IF T=80 THEN 3380
3800 IF T=81 THEN 3600
3810 C=(T=83)-(T=68)
3820 D=(T=69)-(T=-1)
3830 E=E-2
3840 IF E<20 THEN 2030
3850 X=X+C
3860 Y=Y+D
3870 CALL GCHAR(Y,X,T)
3880 IF (T=32)+(T=92)THEN 3940
3890 IF (T=95)*K THEN 2840
3900 IF (T)152)*(K=0)THEN 3120
3910 X=X-C
3920 Y=Y-D
3930 CALL SOUND(-40,-3,4)
3940 W=W-1
3950 IF W)0 THEN 3980
3960 A=A+.6*SGN(X-A)
3970 B=B+.6*SGN(Y-B)
3980 I=I=0
3990 CALL CHAR(144,H$(B)&H$(A)&F$(6+I)&H$(Y)&H$(X)&F$(3-C))
4000 IF (INT(A)=X)*(INT(B)=Y)THEN 2170 ELSE 3780
4010 IF A-62 THEN 4030
4020 IF Z THEN 4290 ELSE 2260
4030 IF Y=23 THEN 4090
4040 Y=Y+1
4050 CALL CHAR(144,H$(Y)&H$(X)&F$(C))
4060 IF A=60 THEN 2080
4070 CALL GCHAR(Y+1,X,A)
4080 IF A<104 THEN 4030 ELSE 4260
4090 P=P+1
4100 IF P-3 THEN 4130
4110 T=7.1011
4120 GOSUB 1440
4130 IF P-6 THEN 4160
4140 T=5.0815
4150 GOSUB 1440
4160 IF P-9 THEN 4200
4170 T=2.1504
4180 GOSUB 1440
4190 CALL COLOR(8,8,15)
4200 IF P=11 THEN 1800
4210 PRINT C$(Q(P));
4220 K=X
4230 GOTO 4260
4240 Y=Y-1
```

```
4250 CALL CHAR(144,H$(Y)&H$(X)&F$(C))
4260 CALL SOUND(-1,2E4,0)
4270 W=W+1
4280 IF W>7 THEN 2770
4290 CALL KEY(0,A,B)
4300 IF RND(.1 THEN 3430
4310 IF A=69 THEN 4510
4320 IF A=80 THEN 3380
4330 D=(A=83)-(A=68)
4340 IF B=0 THEN 4270
4350 S=S+1
4360 E=E-2
4370 IF E<20 THEN 2030
4380 I=I=0
4390 X=X+D
4400 C=I-D+3
4410 CALL GCHAR(Y,X,A)
4420 IF A=121-(D=1)THEN 4240
4430 IF (A=31)+(A=114)+(A=106)THEN 3570
4440 CALL CHAR(144,H$(Y)&H$(X)&F$(C))
4450 IF A=91 THEN 3210
4460 IF A=60 THEN 2080
4470 IF A=40 THEN 3320
4480 IF A=41 THEN 3300
4490 CALL GCHAR(Y+1,X,A)
4500 IF A<104 THEN 4010 ELSE 4260
4510 E=E-10
4520 FOR T=-1 TO 1
4530 Y=Y+T
4540 X=X+D
4550 CALL GCHAR(Y,X,A)
4560 IF (A=92+F)*F THEN 2270
4570 IF (A)103)+(A=31)THEN 3520
4580 IF A=60 THEN 2080
4590 CALL CHAR(144,H$(Y)&H$(X)&F$(C))
4600 NEXT T
4610 GOTO 4490
```

Usare le sprite con il TI Basic

Seconda parte: la teoria

di Riccardo Tesio e Fabio Schiattarella

In questo secondo articolo dedicato all'uso delle sprite in TI Basic vorremmo dare un fondamento teorico all'argomento che il mese scorso avevamo trattato dal punto di vista esclusivamente procedurale. Purtroppo data la complessità e la vastità dell'argomento non si può prescindere da un certo livello di conoscenza di base sull'hardware dei microcomputer e del TI 99 in particolare; il nostro impegno è quello di essere il più chiari possibile. Per un'introduzione agli argomenti trattati, potete consultare l'articolo apparso nella rubrica "I segreti del TI 99/4A" sul numero 21 di MC del luglio 1983. In ogni caso è nei nostri programmi trattare l'argomento hardware e gestione delle risorse nel TI 99/4A, in uno o più articoli nei mesi a venire.

Come molti di voi già sapranno tra i chip presenti all'interno del nostro "Texas" spicca il processore grafico (in inglese VDP: video display processor) TMS 9918A. La A nella sigla distingue tale integrato dal suo predecessore TMS9918, il quale non possedeva ancora la grafica in alta risoluzione secondo la modalità "bit-map". Questo spiega anche la A presente nella sigla del TI 99/4A che monta l'integrato più recente a differenza del più vetusto TI 99/4, di cui forse avrete sentito parlare. Tuttavia il TI 99/4A conserva del modello precedente tutto il Basic che quindi manca di istruzioni grafiche del tipo PLOT o DRAW che troviamo in quasi tutti i computer, pur disponendo tecnicamente delle possibilità di implementazione. Al processore grafico, che d'ora in poi chiameremo VDP per brevità, sono connessi 16 K di memoria RAM che perciò viene detta VDP RAM. In questa memoria trovano posto le tabelle con le mappe dello schermo, dei caratteri, dei colori e delle sprite, i buffer per lo scambio di dati con le periferiche, e soprattutto, nella macchina non espansa, vi risiedono i programmi Basic. Per i più esperti diciamo che tale dispositivo è mappato nell'area di memoria della CPU la quale vi accede per via indiretta attraverso un'apposita routine, caricando in una locazione della CPU RAM l'indirizzo di accesso e usandone un'altra come buffer per leggere o scrivere dati nella VDP RAM. Questa laboriosa procedura di accesso indiretto alla memoria dovrebbe essere tra i maggiori responsabili della famigerata lentezza del TI 99.

La VDP RAM è mappata come memoria del VDP tra gli indirizzi 0 e 16383 (esadecimale 3FFF). Il VDP dispone inoltre di 8 registri in cui è possibile solo scrivere (e perciò detti VDP write only register) nei quali vengono memorizzate le informazio-

Modifiche al programma "labirinto in 3D".

```
260 DIM G(19,28),G$(19)
360 FOR A=1 TO 19
370 READ G$(A)
380 FOR B=1 TO 28
390 G(A,B)=ASC(SEG$(G$(A),B,1))
400 NEXT B
410 NEXT A
551 FOR A=1 TO PP
553 SS(A)=0
555 DD(A)=0
557 NEXT A
559 PP=0
560 ON DI GOSUB 785,895,1005,1115
650 IF Q=69 THEN 670 ELSE 680
670 DI=DI+0
680 IF Q=83 THEN 700 ELSE 710
700 DI=DI+1
710 IF Q=88 THEN 730 ELSE 740
730 DI=DI+2
740 IF Q=68 THEN 760 ELSE 762
762 IF DI<=4 THEN 770
764 DI=DI-4
785 UU=UU+(G(UU-1,V)<33)
895 VV=VV+(G(UU,VV-1)<33)
1005 UU=UU-(G(UU+1,VV)<33)
1115 VV=VV-(G(UU,VV+1)<33)
2570 DATA ````````````````````````````
2580 REM  INSERIRE LE ALTRE DATA
2750 DATA ````````````````````````````
2871 CALL KEY(3,Q,W)
2872 IF W=0 THEN 2871
2873 IF Q<>32 THEN 2871
2874 GOSUB 1240
3005 FOR W=1 TO 19
3010 PRINT G$(W)
3020 NEXT W
3030 PRINT
3040 PRINT
760  DI=DI+3
```

ni relative al modo grafico desiderato ed ai puntatori alle aree di VDP RAM che contengono le mappe del video e delle sprite. Tali 8 registri sono lunghi 8 bit ciascuno e possono essere indirizzati anche essi come se si trattasse di normale memoria. Tale indirizzamento però è piuttosto particolare in quanto contiene già al suo interno il dato da memorizzare. Infatti dei 16 bit che costituiscono l'indirizzo, il più significativo discrimina tra VDP RAM e registri, i 4 bit successivi non sono significativi, tre bit selezionano uno tra gli otto registri e i restanti otto bit meno significativi costituiscono il dato da trasferire nel registro. Questo ci fa capire come sia possibile, possedendo la Mini Memory, scrivere in questi registri utilizzando l'istruzione CALL PEEKV (A,X) che normalmente serve per leggere il valore contenuto in una cella di memoria; la variabile di ritorno X viene messa soltanto per rispettare la sintassi dell'istruzione e non restituisce alcun valore significativo.

Il registro che ci interessa in maniera particolare è il numero 5, che contiene il puntatore all'inizio della mappa delle sprite. Spostando tale mappa in una zona di memoria accessibile al TI Basic e per l'esattezza al posto dei caratteri definibili dall'utente (cossicché sia accessibile tramite il sottoprogramma CHAR), saremo in grado di attivare le sprite anche se il linguaggio espressamente non lo prevede. La mappa dei caratteri ASCII (32-159) inizia alla locazione 1024 e termina alla 2047. La mappa degli sprite può essere spostata a 1792 o a 1920, ovvero a partire dai caratteri ASCII 128 o 144, ponendo il valore 14 ovvero 15 nel registro suddetto secondo le modalità sopra descritte.

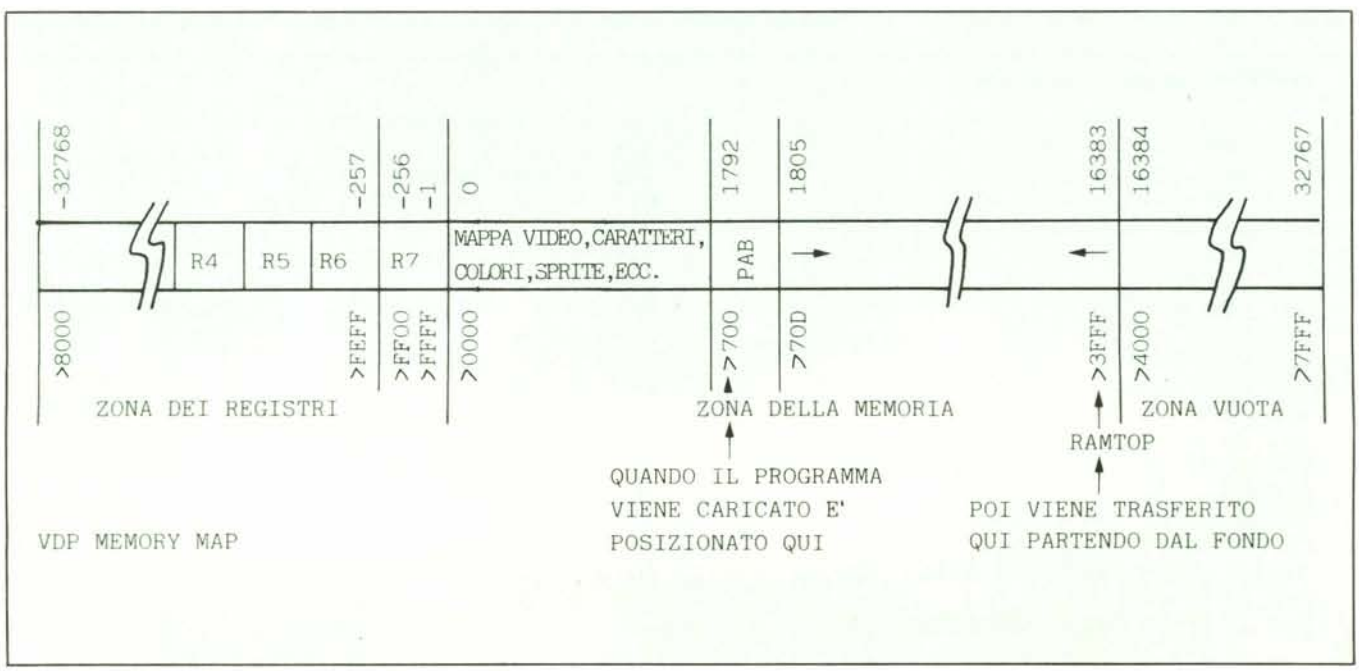
Il nostro problema a questo punto è quello di scrivere un valore in questo registro, non disponendo il Basic di nessuna istruzione del tipo CALL PEEKV. In TI Basic si può ottenere qualcosa di simile sfruttando il modo con cui sono registrati i programmi su cassetta. Prima del programma vero e proprio vengono registrati otto byte; di questi i primi due sono di controllo (XOR dei quattro che seguono); la seconda coppia di byte punta all'inizio della tabella dei numeri di linea, mentre la terza coppia punta alla fine; la quarta coppia indica la fine del programma ed è in genere (macchina non espansa) posta a 3FFFH (16383). Quando il programma viene caricato da nastro viene posizionato a partire da 70H (1805 decimale), distruggendo i caratteri grafici; se non intervengono errori, alla fine della lettura viene trasferito al fondo della RAM e rilocato (forse per adattarsi a diverse configurazioni di memoria) confrontando il puntatore di fine programma con l'indicatore della RAMTOP (posto a H8370 ovvero -31888 della CPU RAM) e ricalcolando eventualmente gli altri puntatori.

Il metodo che permette di utilizzare gli sprite consiste nel creare un programma con particolari valori nei primi otto byte, in modo che appaia all'interprete come un programma lunghissimo, oltre 16 Kbyte. Quando viene caricato in memoria, l'interprete non si accorge di nulla e inizia a rilocarlo partendo dal fondo come di consueto. Ad un certo punto la memoria finisce, ma l'interprete non se ne accorge e continua a trasferire byte fino a che non scrive all'indirizzo desiderato. Il programma che genera un tale programma lo abbiamo presentato il mese scorso.

In effetti il programma generatore di file da noi pubblicato funziona in maniera leggermente diversa da quanto abbiamo sopra esposto, in quanto si avvale del fatto che i programmi sono rilocabili, in ogni caso si tratta di differenze soltanto marginali e l'essenza del ragionamento rimane la stessa.

In tale programma la linea 110 chiede in quale registro e quale dato vogliamo scrivere; la linea 120 calcola l'indirizzo che dovrà raggiungere il programma. Bisogna ricordare che, nonostante l'input di linea 110 non si può scrivere in tutti gli otto registri a disposizione perché i registri 2, 3, 4 e 7 sono riportati al loro valore di default non appena viene commesso un errore (ad esempio il *MEMORY FULL usato nella procedura di caricamento per utilizzare gli sprite), il registro 1 viene ripristinato appena è premuto un tasto. Rimane il registro 0 (ponendo 2 in questo registro si entra in BIT-MAP mode, che però è inutilizzabile), il registro 5 di cui abbiamo già detto e il registro 6 che contiene il puntatore alla mappa degli "attributi" degli sprite (anche questa possibilità non è sfruttabile). Inoltre, come si vede dalla figura pubblicata in questa pagina, durante il rilocamento, prima che il programma raggiunga ad esempio il registro 4, passerà sui registri 7,6 e 5 azzerandoli. Per fortuna scrivendo nel registro 5 non sorgono problemi poiché il valore di default del registro 6 è 0.

La linea 150 del programma scrive il file su nastro. I puntatori di inizio e fine della tabella dei numeri di linea sono posti ad ugual valore per non dover calcolare la coppia dei byte di controllo.

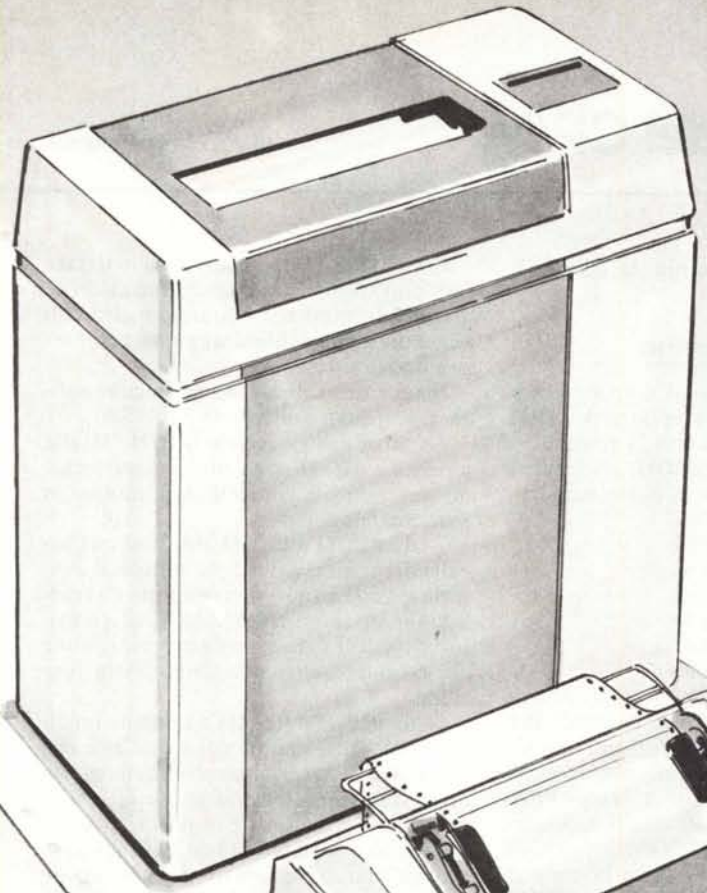


MANNESMANN
TALLY

le stampanti a stock presso

silverstar
componenti e periferiche

Sede: 20146 Milano - Via dei Gracchi, 20 - Tel. (02) 4996 (12 linee) - Telex 332187
40122 Bologna - Via del Porto, 30 - Tel. (051) 522231
00198 Roma - Via Paisiello, 30 - Tel. (06) 8448841 (5 linee) - Telex 610511
10139 Torino - P.za Adriano, 9 - Tel. (011) 443275/6 - 442321 - Telex 220181
35100 Padova - Via S. Sofia, 15 - Tel. (049) 22338



MT 600

- Velocità di stampa: 600 LPM
- Doppia tecnologia di stampa: Data processing e scrittura carattere std
- Doppia risoluzione grafica: 100 x 100 punti per pollice e 60 x 75 punti per pollice.

Accessori
- BAR CODES OCR - A o B
- Interfaccia seriale RS 232
- Interfaccia parallela standard

MT 440 L/D

- Velocità di stampa: 200/400 cps o 50/100 cps ad alta definizione
- Caratteri per linea: 132 caratteri a 10 cpi

Accessori
- Stampa: a 2/4 colori
- Interfaccia: Seriale 24 V / RS - 232 o parallela standard
- Alimentazione foglio singolo: Automatico per 250 fogli formato 210 x 297 mm (UNI A4)

MT 160/180

- Velocità di stampa: 160/200 cps (carattere std) e 40 cps (in stampa di qualità)
- Caratteri per linea: 80/132 a 10 cpi
- Matrice di stampa: 9 x 7
- Alimentazione foglio: a frizione o a trattore
- Interfaccia: Seriale integrale 24 V./RS 232 C e a 8 BIT parallela
- Disponibile con opzioni: alimentatore fogli singoli o introduttore automatico

Costo estremamente contenuto

PIXY PLOTTER

- Numero delle penne: 3
- Dimensione foglio: UNI A4
- Velocità di esecuzione: 200 mm/sec
- Risoluzione: 0,1 mm
- Ripetibilità: 0,3 mm
- Interfaccia: RS 232 C o parallela
- Esecuzioni circolari, ad asco, a spirale
- Adattatore per curve.

Realmente economico

MT 80

- 80 colonne, 80 cps, stampa bidirezionale
- Grafica indirizzabile a singolo ago
- Matrice di stampa: 9 x 8
- Alimentazione foglio: a frizione e con trattore di spinta
- Interfaccia: a 8 BIT parallela o 2 K buffer RS 232

Bassissimo costo