

## Lavoriamo un po' con il Digitizer

*La diffusione dei Personal Computer sta comportando una rivoluzione del mercato, in cui appaiono macchine sempre più interessanti a prezzi sempre più bassi. In questa rivoluzione è stato coinvolto anche il mercato delle periferiche, in cui sono particolarmente attive le case giapponesi che propongono componenti o addirittura linee di componenti adeguate, come prestazioni e prezzi, ai personal computer.*

*Particolarmente interessante è lo sviluppo delle periferiche adatte alla computer grafica, con hardware e software di base sempre più efficiente.*

*Ma trattandosi di apparecchiature di uso personale o al massimo semiprofessionale, è pur sempre l'operatore che deve decidere come utilizzarle e quindi deve sapere come programmarle.*

*In questo articolo parleremo di alcuni problemi che si incontrano nel realizzare software applicativo per digitizer, nel nostro caso un WATANABE DT 1000, presentato nel numero scorso.*

*Come al solito non è nostra intenzione fornire software applicativo del tipo "copia e usa", ma solo spunti che possano suggerire a chi usa personalmente tali macchine, soluzioni a problemi che è sicuramente interessato a risolvere.*

Abbiamo più volte trattato il problema dell'immissione dei dati nelle varie applicazioni di Computer Grafica. In talune di queste, in genere in quelle dove il volume dei dati da immettere è ingente, è pressoché indispensabile l'uso della tavoletta grafica, cioè del digitizer. Il digitizer permette infatti di eliminare il lavoro di predisposizione dei dati (se si tratta di dati grafici esprimibili a mezzo di coordinate geometriche) e di velocizzare la fase di input vera e propria.

In sostanza, lo diciamo ai meno esperti, il digitizer sostituisce la tastiera, e nel programma grafico la istruzione che preleva i dati dalla periferica è una INPUT vera e propria.

È anche immediato capire che l'input da tastiera permette una gran varietà di possibilità in quanto ad ogni tasto viene assegnato uno o più significati, mentre l'input da digitizer, poiché questo fornisce solo una coppia di coordinate X, Y, è molto più semplificato.

Né è molto conveniente realizzare programmi che prevedono una doppia modalità di input, da tastiera per la parte codifica e da digitizer per la parte grafica, in quanto se l'operatore è solo deve spostarsi

continuamente, se sono due debbono lavorare in perfetto sincronismo.

La soluzione al problema è il famoso MENU, di cui abbiamo parlato decine di volte, che tutti i package di Computer Grafica che supportano digitizer prevedono e che permette di dialogare con il computer esclusivamente da digitizer, sia quando si inviano coordinate grafiche, sia quando si scambiano altri messaggi.

Per utilizzare il menu occorre riservare una areola della superficie della periferica, non all'immissione dei dati grafici, ma all'immissione di altri tipi di dati. Il menu fisicamente è un foglietto rettangolare suddiviso in quadratini, che può essere fissato direttamente sul piano di lavoro (come nel caso dell'Apple Tablet, o della tavoletta grafica di MC).

Anche i grossi e costosi package di Computer Grafica Applicativa Professionale, prevedono l'uso di menu, specifici del package, che possono essere posizionati in qualsiasi punto sul piano di lavoro (preferibilmente in una zona non interessata dai dati grafici). L'operatore all'inizio della seduta, comunica dove ha posizionato il menu, semplicemente puntando due suoi vertici opposti.

Ma dietro all'aspetto "hardware" del menu (cioè il foglietto di plastica) c'è l'aspetto software, cioè la routine del programma applicativo che deve riconoscere quando si sta puntando il menu e quale funzione specifica di questo viene richiamata.

Ci deve essere assoluta rispondenza tra la zona del menu e la routine di riconoscimento.

Chi segue la nostra rivista ha trovato più volte esaminata questa problematica, in particolare chi ha usato la MC tablet di MCmicrocomputer ed il suo software ne ha apprezzato una applicazione pratica spinta.

L'occasione di tornare sull'argomento ci è data dal Digitizer DT 1000 della Watanabe già utilizzato nei numeri scorsi. Tale digitizer infatti ha in dotazione un cursore dotato di tastierino a quattro tasti.

Avere a disposizione direttamente sul cursore del digitizer una tastiera che manda messaggi al computer apre nuove possibilità di utilizzazione. In pratica si possono realizzare programmi di input senza dover ricorrere a pesanti routine di riconoscimento e di gestione del menu, e che sottraggono una certa porzione della superficie utile al digitizer.

Il tastierino del Watanabe DT 1000 ha quattro tasti, che come vedremo sono sufficienti per programmi di una certa complessità. Quello che vogliamo realizzare è quindi un programma grafico che utilizzi, per l'input dei dati grafici e non, il digitizer

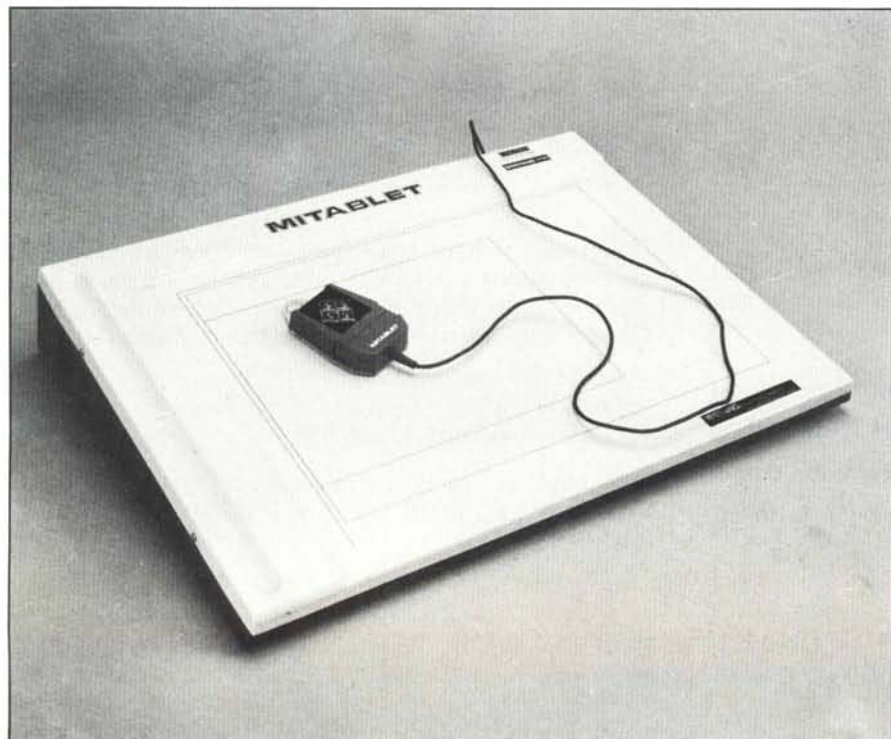


Figura 1 - Il digitizer WATANABE DT 1000. Il formato di lavoro è A3, l'area sensibile è di 3800 per 2600 decimi di millimetro. È una macchina di caratteristiche semiprofessionali.



e il suo tastierino. Abbiamo diviso le varie fasi.

La prima consiste nel realizzare una routine di utilizzo del tastierino e di riconoscimento dei messaggi da questo inviati. Il relativo programma costituisce in pratica una struttura che si presta a varie utilizzazioni.

Poi abbiamo realizzato attorno a questa struttura un programma applicativo schematico in cui sono affrontate le varie problematiche relative al dialogo tra periferica e computer. Abbiamo in sostanza suddiviso i messaggi operativi in tre gruppi logici: messaggi che non prevedono l'invio di dati grafici (ad es. CLEAR, END, ecc.), messaggi che prevedono l'invio di un numero prefissato di dati grafici (ad es. BOX, CIRCLE, WINDOW, ecc.), messaggi che prevedono l'invio di un numero non prefissato di dati grafici (ad es. SPEZZATA APERTA, SPEZZATA ORIZZ. VERT.).

### Riconoscimento dei messaggi inviati via tastierino

Il DT 1000 invia messaggi, direttamente in codice ASCII, del tipo XXXX, YYYY,

Z lfc. Sono 12 caratteri, i primi quattro indicano il valore X, poi c'è la virgola, il valore Y, un'altra virgola, la Z e il salto riga e a capo (line feed, carriage return).

La Z indica il valore del tasto che è stato premuto e quindi fornisce valori 1,2,4,8. Tale valore può essere tradotto in corrispondenti valori 1,2,3,4 con una semplice formuletta matematica  $G = INT(F \cdot 6 + .9)$ , molto più utile per una gestione di subroutine, tramite l'istruzione ON G GOSUB...

Poiché i tasti sono quattro le funzioni distinte possono essere solo quattro. Combinando però due tasti premuti in sequenza i codici diventano 16.

Nel caso in cui alcuni dati non richiedano l'uso del tastierino in quanto non necessitano di altri messaggi, occorrerà rinunciare a qualcuna delle 16 possibilità.

Ad esempio se stiamo immettendo una linea spezzata, di cui non abbiamo predefinito la lunghezza e che immettiamo quindi punto per punto, non occorre utilizzare il flag Z, se non per dare il messaggio di fine immissione.

Negli esempi pubblicati tratteremo a fondo questa tematica, nel primo caso rea-

lizzando una routine di riconoscimento e di gestione di 12 subroutine in funzione delle coppie di tasti premuti in sequenza. Nel secondo programma tale routine viene corredata di subroutine specificamente grafiche.

### Il programma struttura

È, come detto, un programma che utilizza per l'input dei dati esclusivamente il digitizer e che permette la gestione di un menu di 12 opzioni e l'immissione di punti grafici in sequenza che non interferiscono con queste.

Può costituire la struttura di un qualsiasi programma di input o di input/visualizzazione o di input/visualizzazione/memorizzazione di dati grafici.

La routine di riga 9000 è quella che legge il dato dal digitizer. In riga 9000 c'è un beep per l'autorizzazione all'immissione, l'azzeramento della variabile stringa K\$ che riceve i 12 caratteri ASCII inviati dal digitizer e l'accensione di questo (tramite l'istruzione APPLESOFT IN#3).

In riga 9020 c'è il caricamento della variabile K\$, cui viene aggiunto, uno alla



Figura 2 - Il tastierino del cursore del WATANABE DT 100. Premendo i tasti Z, 1, 2, 3 vengono trasmessi rispettivamente i valori 1, 2, 4, 8 utilizzabili via software.

```

100 HOME
110 GOSUB 9000: ON G% GOSUB 200,1000,2000,3000: GOTO 110
200 REM PUNTO A SEGUIRE
210 K = K + 1: PRINT "PUNTO ";K: TAB(10)X%; TAB(16)Y%: RETURN
1000 REM ROUTINE SENZA PUNTI
1010 GOSUB 9000: ON G% GOSUB 1100,1200,1300,1400: RETURN
1100 PRINT "1/2": RETURN
1200 PRINT "1/1 CLEAR": K = 0: RETURN
1300 PRINT "1/2": RETURN
1400 PRINT "1/3 FINE": RETURN
2000 REM ROUTINE PUNTI FISSI
2010 GOSUB 9000: ON G% GOSUB 2100,2200,2300,2400: RETURN
2100 PRINT "2/2": RETURN
2200 PRINT "2/1": RETURN
2300 PRINT "2/2": RETURN
2400 PRINT "2/3": RETURN
3000 REM ROUTINE PUNTI VARIABILI
3010 GOSUB 9000: ON G% GOSUB 3100,3200,3300,3400: RETURN
3100 PRINT "3/2": RETURN
3200 PRINT "3/1": RETURN
3300 PRINT "3/2": RETURN
3400 PRINT "3/3": RETURN
9000 REM LETTURA DA DIGITIZER
9010 PRINT CHR$(7):K$ = "": IN# 3
9020 FOR I = 1 TO 12: GET J#:K$ = K$ + J#: NEXT I
9030 X% = VAL < MID$(K$,1,4):Y% = VAL < MID$(K$,6,4)
9040 F% = VAL < MID$(K$,11,1):G% = F% * 6 + 9
9050 IN# 0: RETURN
    
```

Figura 3 Programma STRUTTURA. La routine di riga 9000 legge i dati coordinate X%, Y% e il flag F% inviati dal digitizer. Le varie subroutine operative sono gestite dal valore di tale flag.

PUNTO 1	1779	1355
PUNTO 2	2484	1821
PUNTO 3	1721	2234
3/2		
1/2		
PUNTO 4	2287	2061
PUNTO 5	2297	1866
PUNTO 6	2226	1818
3/3		
1/2		
2/2		
1/1 CLEAR		
PUNTO 1	1906	1984
PUNTO 2	1906	1984
PUNTO 3	1576	1979
PUNTO 4	1576	1979

Figura 4 - Output del programma STRUTTURA. L'output su video si limita a visualizzare le subroutine richiamate (premendo due tasti in sequenza), oppure il valore del punto immesso premendo il tasto Z.

volta, il carattere J\$ in arrivo tramite la istruzione GET\$.

Con BASIC di derivazione MICROSOFT occorre cambiare la istruzione di gestione della porta RS 232, e utilizzare, al posto del GET\$, che riceve un carattere alla volta solo quando viene premuto un tasto, l'istruzione

100 I\$ = INKEY\$: IF I\$ = "" THEN 100 che esegue un loop continuo che si interrompe solo quando "arriva" un carattere.

In riga 9030 la stringa K\$ viene sezionata e tradotta in valori numerici interi X%, Y%. Ricordiamo che le coordinate fornite dal digitizer sono in decimi di millimetro e che il formato utile è 3800 per 2600.



```

10 DATA BOX,CIRCLE,ARC,SPEZZ,CHIUSA,SPEZZ,APERTA,SPEZZ,ORIZZ,VERT
20 FOR I = 1 TO 6: READ N%(I): NEXT I
30 HGR : HCOLOR= 3: TEXT : HOME
40 P = 3.14159: SP = P / 64: DIM K%(200,3)
50 S = 280 / 3800: T = 192 / 2600: REM SCALING DI DEFAULT
60 GOSUB 1720: ON G% GOSUB 70,100,600,1200: GOTO 60
70 REM PUNTO A SEGUIRE
80 K = K + 1: PRINT "PUNTO":K: TAB( 10)X%: TAB( 16)Y%: RETURN
100 REM ROUTINE SENZA PUNTI
110 GOSUB 1720: ON G% GOSUB 200,300,400,500: RETURN
200 PRINT "1/2 SWITCH "
210 IF NOT SW THEN POKE 49234,0: POKE 49232,0: SW = 1: RETURN
220 POKE 49233,0: SW = 0: RETURN
300 PRINT "1/1 CLEAR TOTALE"
310 FOR J = 1 TO K: K%(J,1) = 0: K%(J,2) = 0: K%(J,3) = 0: NEXT J
320 K = 0: HOME : HGR : POKE 49233,0: RETURN
400 PRINT "1/2 CLEAR PARZIALE":K = KI: RETURN
500 HOME : PRINT "1/3 VISUALIZZAZIONE DATI": PRINT
510 FOR J = 1 TO K: IF K = 0 THEN 540
520 PRINT J: TAB( 6)K%(J,2): TAB( 12)K%(J,3)
530 IF K%(J,1) < > 0 THEN PRINT TAB( 20)N%(K%(J,1))
540 PRINT : NEXT J: RETURN
600 REM ROUTINE PUNTI FISSI
610 GOSUB 1720: ON G% GOSUB 700,800,900,1100: RETURN
700 PRINT "2/2 BOX "
710 KI = K: K%(K + 1,1) = 1
720 GOSUB 1700: AX = X%: BX = Y%
730 GOSUB 1700: CX = X%: DX = Y%
740 HPLLOT S + AX: T + BX TO S + AX: T + DX
750 HPLLOT TO S + CX: T + BX TO S + AX: T + BX: RETURN
800 PRINT "2/1 CIRCLE "
810 KI = K: K%(K + 1,1) = 2
820 GOSUB 1700: AX = X%: BX = Y%
830 GOSUB 1700: CX = X%: DX = Y%
840 EX = SQR ((AX - CX) ^ 2 + (BX - DX) ^ 2)
850 HPLLOT S + (AX + EX): T + BX: FOR A = SP TO 2 * P STEP SP
860 HPLLOT TO S + (AX + EX * COS (A)): T + (BX + EX * SIN (A))
870 NEXT A: RETURN
900 PRINT "2/2 ARC "
910 KI = K: K%(K + 1,1) = 3
920 GOSUB 1700: AX = X%: BX = Y%: HPLLOT S + X%: T + Y%
930 GOSUB 1700: CX = X%: DX = Y%: HPLLOT S + X%: T + Y%
940 GOSUB 1700: EX = X%: FX = Y%: HPLLOT S + X%: T + Y%
950 D = CX ^ 2 + DX ^ 2 - EX ^ 2 - FX ^ 2
960 E = AX ^ 2 + BX ^ 2 - CX ^ 2 - DX ^ 2
970 C = AX * (FX - DX) + CX * (BX - FX) + EX * (DX - BX)
980 A = ((BX - DX) * D - (DX - FX) * E) / C / 2
990 B = ((CX - EX) * E - (AX - CX) * D) / C / 2
1000 R = SQR ((AX - A) ^ 2 + (BX - B) ^ 2)
1010 A1 = AX - A: B1 = BX - B: A2 = EX - A: B2 = FX - B
1020 L1 = ATN (B1 / A1): IF A1 < 0 THEN L1 = L1 + P: GOTO 1040
1030 IF B1 < 0 THEN L1 = L1 + 2 * P
1040 L2 = ATN (B2 / A2): IF A2 < 0 THEN L2 = L2 + P: GOTO 1060
1050 IF B2 < 0 THEN L2 = L2 + 2 * P
1060 X = A + R * COS (L1) + .5: Y = B + R * SIN (L1) + .5: HPLLOT S +
X, T * Y
1070 FOR I = L1 TO L2 STEP SP * SGN (L2 - L1)
1080 X = A + R * COS (I) + .5: Y = B + R * SIN (I) + .5
1090 HPLLOT TO S + X, T * Y: NEXT I: RETURN
1100 PRINT "2/3 WINDOW"
1110 IF K < > 0 THEN PRINT "LA WINDOW PUO' ESSERE SETTATA":
1120 IF K < > 0 THEN PRINT "SOLO ALL'INIZIO DEL PROGRAMMA "
1130 GOSUB 1700: AX = X%: BX = Y%: S = 279: AX: T = 191 / BX: RETURN
1200 REM ROUTINE PUNTI VARIABILI
1210 GOSUB 1720: ON G% GOSUB 1300,1400,1500,1600: RETURN
1300 PRINT "3/2 SPEZZATA CHIUSA"
1310 KI = K: K%(K + 1,1) = 4
1320 GOSUB 1700: AX = X%: BX = Y%: HPLLOT S + AX: T + BX
1330 GOSUB 1700: CX = X%: DX = Y%
1340 IF FX < > 1 THEN HPLLOT TO S + AX: T + BX: K = K - 1: RETURN
1350 HPLLOT TO S + CX: T + DX: GOTO 1330
1400 PRINT "3/1 SPEZZATA APERTA"
1410 KI = K: K%(K + 1,1) = 5
1420 GOSUB 1700: AX = X%: BX = Y%: HPLLOT S + AX: T + BX
1430 GOSUB 1700: CX = X%: DX = Y%
1440 IF FX < > 1 THEN K = K - 1: RETURN
1450 HPLLOT TO S + CX: T + DX: GOTO 1430
1500 PRINT "3/2 SPEZZATA ORIZZ / VERT. "
1510 KI = K: K%(K + 1,1) = 6
1520 GOSUB 1700: AX = X%: BX = Y%: HPLLOT S + AX: T + BX: EX = BX
1530 GOSUB 1700: CX = X%: HPLLOT TO S + CX: T + EX
1540 GOSUB 1700
1550 IF FX < > 1 THEN HPLLOT TO S + AX: T + BX: K = K - 1: RETURN
1560 DX = Y%: HPLLOT TO S + CX: T + DX: EX = Y%: GOTO 1530
1600 PRINT "3/3 END ": END
1700 REM LETTURA DA DIGITIZER
1710 K = K + 1: FF = 0
1720 PRINT CHR$( 7): K% = "": IN# 3
1730 FOR I = 1 TO 12: GET J#: K% = K% + J#: NEXT I
1740 X% = VAL ( MID$( K%, 1, 4)): Y% = 2600 - VAL ( MID$( K%, 6, 4))
1750 F% = VAL ( MID$( K%, 11, 1)): G% = F% ^ .6 + .9
1760 IF FF THEN 1780
1770 K%(K,2) = X%: K%(K,3) = Y%
1780 FF = 1: IN# 0: RETURN

```

Figura 5 - Programma GRAFICO. La struttura è stata completata con una serie di routine grafiche che permettono la immissione di dati grafici e la contemporanea realizzazione del relativo disegno.

Il flag F% ritorna con valori 1,2,4,8 a seconda che siano premuti rispettivamente i tasti Z,1,2,3. E poiché non possiamo utilizzare nessuna di queste due forme, usiamo il citato metodo brutale, ma istantaneo, per tradurli in valori da 1 a 4 necessari per poter utilizzare l'istruzione ON G GOSUB...

Il main program è la sola riga 110, dalla quale sono richiamate le quattro subroutine principali.

La prima, di riga 200-210, è richiamata premendo il tasto Z, è quella di immissione del punto a seguire, cioè del punto che può essere richiesto in una sequenza di lunghezza prefissata o libera, e che quindi necessita di un settaggio della Z che non interferisca con gli altri codici.

Alle altre routine si accede premendo una prima volta i tasti 1,2,3 e una seconda volta uno qualsiasi dei tasti, per un totale di 12 combinazioni (1/Z, 1/1, ..., 3/3).

Nel programma "Struttura" le 12 routine sono vuote, se richiamate appare solo l'indicazione dei tasti premuti.

In termini di righe di programma, dalla riga 110, a seconda del tasto premuto, la prima volta si arriva alle righe 1000, 2000, 3000 (routine di passaggio) da dove, in funzione del tasto premuto la seconda volta, sono richiamate le 12 subroutine operative. Di queste sono implementate, a puro titolo esemplificativo, l'END e il CLEAR TOTALE, che azzerano il contatore dei punti.

Evidentemente questo programma non ha nessuna utilità pratica, può servire solo da ossatura di un programma realizzato inserendo delle routine operative nelle apposite righe.

## Un programma un po' più complesso

Il digitizer serve dunque per immettere dati, in genere grafici, opportunamente codificati. Ora occorre decidere cosa fare di questi dati, e cioè che uso deve prevedere il programma di questi dati?

Trattandosi di dati grafici la prima cosa da fare è la visualizzazione, oltre che in forma numerica, anche in forma grafica, per poterne controllare la validità. E questo può essere fatto "on line" oppure "off line", cioè a mano a mano che si caricano i dati o tutto insieme alla fine.

Nel primo caso è facilitata la correzione istantanea del dato, nel secondo caso occorre costruire pesanti routine di correzione, oppure più semplicemente cominciare da capo.

È quindi preferibile la prima soluzione. C'è poi bisogno di memorizzare i dati e anche qui ci sono due possibilità, quella di memorizzarli in forma grafica, come SLIDE, e quella di memorizzarli come FILE di dati e codici.

In questo secondo caso sarà un'altro programma che leggerà il file e rielaborerà i dati visualizzandoli in una o più maniere differenti.

Tutte queste tematiche andrebbero con-





# Cin, cin... brindiamo ad una scelta azzeccata!

Perché ho trovato  
un elaboratore  
che ha grandi prestazioni  
ed un piccolo prezzo!

Perché il Gruppo BAGSH  
mi garantisce programmi  
personalizzati di elevata qualità!

Perché le diverse esperienze  
di un gruppo di qualificate  
aziende ha risolto i miei problemi  
ed aumentato i miei profitti!



**ICL**  
trader point

memoria RAM da 64KB a 1024KB  
memoria di massa  
da 1.6MB a 30MB  
da 1 a 8 utenti in reale  
multiprogrammazione



**il punto d'incontro delle esperienze più qualificate.**

Via Nicolò dell'Arca, 1 - 40129 BOLOGNA - Tel. (051) 35.32.31/37.10.99 (3 linee)

BOLOGNA, BRESCIA, CARPI, CATTOLICA, CESENA, FERRARA, FIRENZE  
FOLIGNO, MILANO, MODENA, PADOVA, PARMA, REGGIO EMILIA, TRIESTE



template e quindi inserite in un programma di INPUT completo.

Esempio di software per digitizer complesso e "quasi" completo è quello della MC tablet (che si usa in collegamento con l'Apple II) che prevede ben 40 funzioni di vario genere, di disegno vero e proprio, di TEXT, di settaggio WINDOW, di SWITCH tra le pagine, di gestione archivio SLIDE, ecc.

Essendo però un software orientato all'utilizzo del monitor grafico dell'Apple II non contiene routine per la memorizzazione dei dati in forma di file.

## Il programma grafico

Partendo dal programma STRUTTURATA, abbiamo realizzato una sua imple-

11/3	VISUALIZZAZIONE DATI		
1	2656	1665	BOX
2	1607	592	
3	2276	749	CIRCLE
4	1632	1115	
5	2245	1561	ARC
6	2500	1270	
7	2271	755	
8	2157	723	SPEZZ. APERTA
9	1466	628	
10	1404	1176	
11	2296	1389	
12	2646	844	
13	1482	460	BOX
14	2286	1276	

Figura 6 - Output del programma GRAFICO. La subroutine, richiamata premendo in sequenza i tasti 1 e 3, permette la visualizzazione in forma alfanumerica dei dati immessi e conservati nella matrice  $K\% (K,3)$ . Viene conservato anche un semplice codice che specifica la routine grafica richiamata.

mentazione che, utilizzando esclusivamente il digitizer WATANABE DT 1000 per l'immissione dei dati, realizza un disegno, memorizzando contemporaneamente i dati in una matrice.

Mancano due funzioni fondamentali in quanto avrebbero allungato il programma in maniera da renderlo non pubblicabile, senza aggiungere nulla alla problematica "input" oggetto dell'articolo.

La prima consiste nella routine di correzione grafica, e questo deriva dal fatto che il disegno viene realizzato via via che si immettono i dati. Mentre è possibile riimmettere una funzione che ci si è accorti di aver sbagliato, semplicemente "ritornando indietro" con il contatore, non è stata inserita la routine che cancella dal disegno la funzione sbagliata.

La seconda assenza, ancora più grave, è quella della routine di archiviazione della figura composta in una "slide" oppure dei dati codificati, che producono quella stessa figura, in forma di "file".

Ma questo argomento è stato trattato più volte nel corso di questi articoli, per cui ci è facile inviare gli interessati alla consultazione di numeri precedenti di MCmicrocomputer.

Passiamo ad esaminare il programma. Le routine implementate sono 12, raggruppate in tre gruppi da quattro.

Il primo gruppo viene richiamato con il tasto 1, premendo il quale si entra nella routine di riga 100, chiamata "routine senza punti", in quanto le funzioni implementate non producono incrementi del contatore, e quindi immagazzinamento di dati. Sono quindi routine di utilità per il lavoro, e cioè: la SWITCH, di riga 200, routine interruttore che, se richiamata, provoca alternativamente il passaggio dalla pagina di testo, sulla quale si controllano i dati alfanumerici, alla pagina HGR, nella quale gli stessi dati assumono forma grafica.

C'è poi la routine di "Clear totale" di riga 300, che permette la cancellazione del disegno e l'abblanckamento del vettore in cui sono caricati i dati.

La routine di "Clear parziale" (riga 400) consiste invece nel semplice ritorno all'"in-

chiamata solo all'inizio del lavoro e prevede l'immissione di un solo punto.

Se non viene richiamata, il programma utilizza i fattori di scaling di default, che sono quelli massimi  $S = 280/3800$  e  $T = 192/2600$ , per cui ciascun punto rilevato sulla superficie del digitizer può essere rappresentato sul monitor APPLE II traducendo le sue coordinate  $X_m = S \cdot X_d$ ,  $Y_m = T \cdot Y_d$ .

Fortunatamente S e T sono pressoché uguali e quindi il disegno non viene schiacciato o allungato sul video.

Se il disegno da digitare è più piccolo e lo si vuole vedere comunque su tutto il formato monitor, occorre alterare opportunamente i valori S e T. E questo si fa con la routine di WINDOW, semplificata, posizionando l'origine del disegno (il vertice in alto a sinistra) sull'origine del digitizer e immettendo il punto in basso a destra, così facendo vengono automaticamente settati i nuovi valori S e T.

L'ultimo gruppo di routine (riga 1200) è costituito dalle routine "a punti variabili" in cui cioè il numero dei punti da immettere non è noto a priori. In ognuna di queste routine c'è un controllo di fine immissione. In pratica i punti interni alla routine vanno immessi premendo il tasto 1, l'ultimo va immesso premendo un qualsiasi altro tasto.

Le routine sono: la spezzata chiusa (riga 1300) in cui finita l'immissione, l'ultimo punto viene collegato al primo, la spezzata aperta (riga 1400) in cui questa chiusura non c'è.


C'è poi la routine di spezzata orizz./vert. (riga 1500), con la quale si immettono alternativamente segmenti orizzontali e verticali. La routine in pratica impedisce che per un errore, anche piccolo, di puntamento, un segmento che sappiamo a priori orizzontale o verticale diventi obliquo.

La routine di lettura da digitizer (riga 1700) gestisce anche l'incremento del contatore (riga 1710) e, tramite il flag FF, il caricamento o meno della matrice, ovvero se la routine viene richiamata da una funzione "senza punti" il contatore non viene incrementato e la matrice non viene caricata.

L'uso di questo programma richiede un minimo di pratica, fatta la quale è assolutamente immediato. Dalle prove fatte risulta più facile lavorare sullo schermo alfanumerico facendo fugaci puntate sullo schermo grafico.

## Le implementazioni possibili

Come detto tale programma ha solo uno scopo dimostrativo ed è quindi carente in talune funzioni. C'è poi il fatto che non è vero che si sia obbligati alla limitazione di 12 routine.

Infatti quella di riga 70, di "punto a seguire", se si opera correttamente con le altre routine non viene mai richiamata. Quindi si può ampliare il numero delle funzioni da 12 a 16 e aggiungere alcune di quelle che mancano. 



# DIGITEK HA SCELTO BENE. SCEGLI BENE ANCHE TU.



MPF III  
CPU: 6502, 1 MHz  
ROM: 24 K (con interprete basic apple  
soft compatibile)  
RAM: 64 K dinamiche più 2 K statiche  
per le 80 colonne di testo  
VIDEO  
- Testo: 40x24 / 80x24 (2 pagine)  
- Grafica GR: 40x48 16 colori (2 pagine)  
- Grafica HGR: 280x192 6 colori  
(2 pagine)

## MPF III

Lo con-  
fessiamo subito.

È questo, a sinistra,  
l'unico componente che non  
consegnamo con MPF III. Però, è necessario per rendere il sistema MPF III vera-  
mente completo!

MPF III, personal computer, ha una struttura compatta, solida ed ergonomica. Scrive maiuscolo e minuscolo ed ha funzioni di editing. Il suo funzionamento è comodo ed immediato, grazie al ricchissimo software disponibile. L'integrità dei drivers è salvaguardata dall'unica possibilità di connessione ad MPF III. In alta e bassa risoluzione, sono tanti i colori possibili sul monitor. Il volume dell'altoparlante interno è regolabile a piacere. MPF III, personal computer. Bello, dinamico e forte. Unisce sorprendenti possibilità ad un ottimo rapporto prestazioni/prezzo.

### DIGITEK COMPUTER

VIA VALLI, 28 - 42011 BAGNOLO IN PIANO (Reggio Emilia) Tel. (0522) 61623 r.a.

