

# grafica

di Francesco Petroni

## Ancora sullo standard IBM

Nel lontano settembre del 1985 MC eseguì una nutrita serie di prove sui Computer IBM compatibili, in quello stesso numero apparve un articolo, realizzato dal sottoscritto, in cui si esaminavano le conseguenze che l'affermazione dello standard IBM stava comportando nel campo della grafica.

Vogliamo riprendere l'argomento per documentare gli sviluppi che ha avuto, nei pochi mesi trascorsi, il settore, sia quello dei PC compatibili sia quello della grafica su standard IBM, e anche per portare ulteriori contributi ai numerosi lettori che utilizzano questo tipo di macchine.

### Il fenomeno dei cinesi

Uno degli elementi che ha contribuito all'affermazione dello standard IBM sta senza dubbio nella diffusione dei «cinesi», che si sono affermati in porzioni di mercato particolari come gli hobbysti o i cantinieri. In pratica una fetta di mercato di scarso interesse per le grosse case ma che, come importanza, ai fini della diffusione dello standard, ne ha molta.

Dal punto di vista hardware è noto che ormai i «cinesi» hanno superato gli originali, nel senso che ormai tutti i

cinesi offrono, nella configurazione base, opzioni che il PC IBM non ha.

I tre standard dei compatibili sono: clock a 4.77/7.20 MHz switch-abile; espandibilità a 640 kbyte su piastra base; scheda grafica Hercules in dotazione.

E in effetti questi sono i tre limiti maggiori del PC IBM, la velocità (rispetto alle macchine 68000 o 8086), la espandibilità che costa slot (oltre che soldi) e la mancanza della grafica nella configurazione base.

Nel campo della grafica in questi ultimi mesi non ci sono stati sviluppi decisivi, anzi in pratica esiste più confusione di prima.

Nella sola casa IBM esistono una scheda monocromatica alfanumerica, e tre schede grafiche a colori, inoltre esistono in catalogo 4 tipi differenti di monitor. Poiché, inoltre, le schede possono convivere sulla stessa macchina risultano possibili una decina di configurazioni differenti.

Allargandosi poi alla produzione

Figura 1 - Tipologia Business Graphic. Nell'ambito della Business Graphic esistono diverse tipologie di grafici, ciascuno con un proprio range di utilizzabilità. Quello mostrato si usa nell'analisi di stock di magazzino, ed è presente nella tipologia disponibile sul Symphony 1.1.

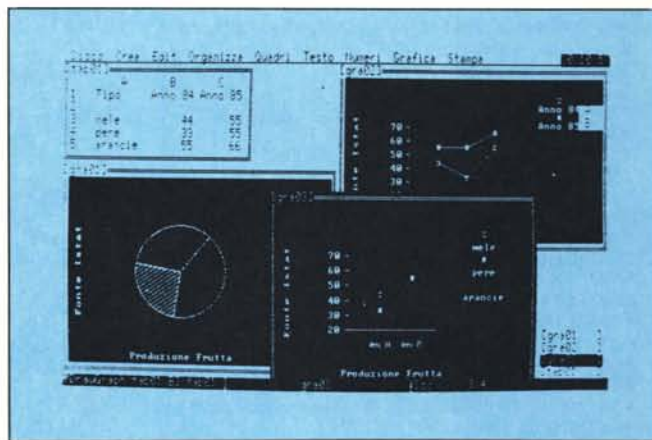
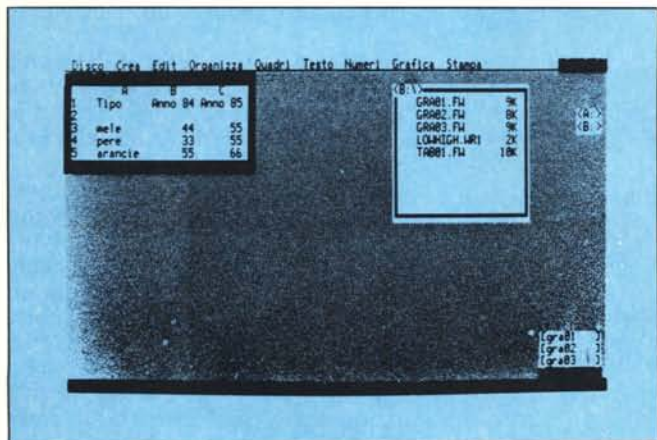
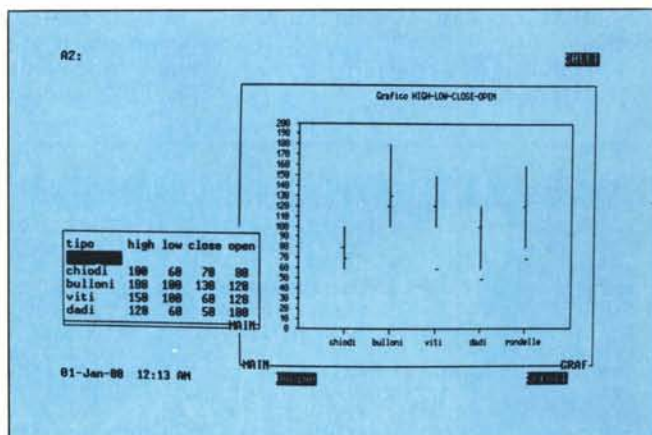


Figura 2 - Grafici con Strumenti Multifunzionali. È realizzato con il Framework 1.1 della Ashton Tate (presentato nel numero 41 di MC). Da una stessa tabella di dati numerici si possono realizzare più grafici, ognuno dei quali, in FW, corrisponde ad un quadro riposto in una delle vaschette in basso a destra sul video.

Figura 3 - Grafici con Strumenti Multifunzionali. Così appare il video con tutti i quadri richiamati e aperti, ovvero la scrivania con tutte le pratiche aperte e sparse sulla sua superficie.

non IBM, esiste l'ormai standard Hercules, che permette grafica monocromatica ad alta definizione e a basso costo in quanto non richiede monitor particolari, esiste inoltre uno standard Olivetti, ormai riconosciuto in fase di installazione dai nuovi pacchetti software americani, in quanto in America ha un buon successo l'Olivetti «Americano» AT/T 6300.

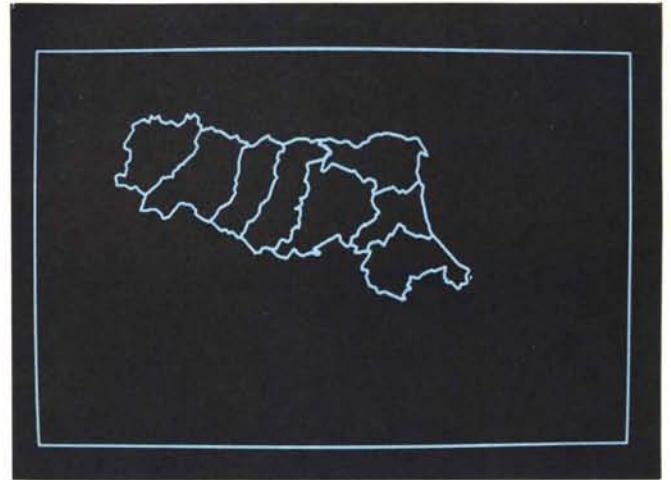
In flessione le schede grafiche a colori ed ad alta risoluzione non di produzione IBM, in quanto superate dalla EGA. La loro diffusione è legata alla utilizzazione di software dedicati specifici per la produzione televisiva.

Per tali schede poi risultano indispensabili monitor ad altissima definizione (quelli che permettono di distinguere un pixel di un certo colore tra centinaia di migliaia di pixel di altre centinaia di colori). In questo caso l'investimento tra scheda, monitor e software diventa di decine di milioni.

Per fare un esempio concreto delle vastità delle possibilità riportiamo l'elenco delle installazioni previste per il prodotto Symphony 1.1, che essendo molto recente riconosce anche le ultime schede.

- IBM Display Adapter (alfanumerico)
- Hercules 80 per 25
- Hercules 90 per 38

Figura 5  
Out del Programma di Lettura. Nel nostro caso i dati vengono subito visualizzati in forma grafica, per controllo, e poi memorizzati in un file sequenziale, riutilizzabile in successivi programmi.



- IBM Color Graphic Adapter (colore)
- IBM Color Graphic Adapter (monocromo)
- IBM Enhanced Graphic Adapter (varie)
- IBM Portable
- AT&T monocromo (Olivetti M24)
- AT&T colore (Olivetti M24)
- Tecmar, Plantronics, ecc.

Inoltre esiste l'opzione grafica combinata con il testo, e, nel caso di due unità video installate, la modalità Duale. Questo vuol dire che dello stesso

prodotto un utente che lo usi su più macchine ne deve avere circa dieci versioni.

Per rimanere sul Symphony 1.1 vediamo una novità, e cioè il grafico di tipo High-Low-Close-Open, che permette di rappresentare valori di quantità di magazzino. I due valori High e Low identificano un segmento sul quale vengono riportati nelle due direzioni opposte i valori Close e Open. Poiché di problematiche di magazzino non me ne intendo molto, per evitare di fare figuracce, rinvio gli interessati alla figura 1.

Rimanendo in tema di prodotti multifunzionali facciamo una puntatina sul Framework, che possiede anche funzionalità di grafica commerciale, integrata con le altre modalità di Tabellone Elettronico, di Word Processor, di gestore di Data Base.

La filosofia di lavoro del Framework, presentato sul numero 41 di MC, è quella della scrivania sulla quale sono posizionate «le vaschette» con le pratiche da evadere. Fisicamente le vaschette appaiono in basso a destra sul video e il loro contenuto può essere richiamato, modificato, cancellato, ecc.

Per fornire una esemplificazione delle potenzialità grafiche del FW abbiamo realizzato una semplice tabella, che assume il nome di TAB01 nella vaschetta e nell'archivio su disco.

Partendo dai dati contenuti in questa tabella abbiamo realizzato tre grafici differenti nei quali si comprende come gli stessi dati si possono vedere in vari modi, ognuno dei quali ha un suo significato. In pratica la rappresentazione grafica non è che uno dei modi attraverso i quali si possono «leggere» i dati stessi, e anche nell'ambito delle rappresentazioni grafiche i modi sono numerosi.

Gli esempi realizzati in figure 2 e 3 permettono di comprendere il concetto di vaschetta, e il secondo concetto di modalità di lettura dei dati. La dif-

```

100 REM                                     inizializzazione video
110 CLS:SCREEN 2:DIM A%(1000,3)
120 WINDOW (0,0)-(3800,2600):LINE (0,0)-(3800,2600),,B
130 REM                                     lettura stringa ascii da digitizer
140 GOSUB 180
150 FX=VAL(MID$(B%,11,1)):X%=VAL(MID$(B%,1,4)):Y%=VAL(MID$(B%,6,4))
160 K=K+1:A%(K,1)=X%:A%(K,2)=Y%:A%(K,3)=FX:GOSUB 250
170 IF F%=8 THEN 270 ELSE GOTO 140
180 REM                                     inizializzazione digitizer
190 DEF SEG=0:CLOSE B%="":OPEN "COM1:1200,n,7,1" AS #1
200 IF EOF(1) THEN 200
210 A%=INPUT$(LOC(1),#1)
220 B%=B%+A%:IF LEN(B%)>12 THEN 230 ELSE GOTO 200
230 CLOSE #1:DEF SEG:RETURN
240 REM                                     visualizzazione passo passo
250 IF F%=2 THEN PSET (X%,Y%):RETURN
260 LINE -(X%,Y%):RETURN
270 REM                                     visualizzazione finale
280 KK=K:CLS
290 FOR K=1 TO KK:X%=A%(K,1):Y%=A%(K,2):F%=A%(K,3):GOSUB 240:NEXT K
300 REM                                     scrittura file
310 OPEN "dati.dat" FOR OUTPUT AS 1
320 PRINT #1,KK:FOR K=1 TO KK
330 PRINT #1,A%(K,1),A%(K,2),A%(K,3):NEXT K:CLOSE #1:END
340 REM                                     predisposizione videc
350 CLS:SCREEN 2:DIM B%(1000,3)
360 WINDOW (0,0)-(3800,2600):LINE (0,0)-(3800,2600),,B
370 OPEN "dati.dat" FOR INPUT AS 1
380 REM                                     lettura file
390 INPUT #1,KK:FOR K=1 TO KK
400 INPUT #1,B%(K,1),B%(K,2),B%(K,3):NEXT K:CLOSE #1
410 REM                                     visualizzazione per controllo
420 FOR K=1 TO KK
430 X%=B%(K,1):Y%=B%(K,2):F%=B%(K,3)
440 GOSUB 240:NEXT K:END

```

Figura 4 - Programma di lettura da Digitizer. Per prelevare dati da una piantina geografica via digitizer, occorre un programma che contenga una routine di lettura dei dati che arrivano via porta RS 232, e che li interpreti così come occorre al programma.



Figure 6, 7 - Out di una Videata Realizzata con StoryBoard. Per mezzo del modulo Picture Taker (acchiappaimmagini dello StoryBoard (numero scorso di MC)), è possibile prendere lo scheletro di una immagine realizzata con il programma precedente e sottoporlo ad una cura di «bellezza» con le varie funzionalità interattive dello strumento.

ferenza tra GRF02 e GRF03 non è che l'ordine di lettura dei dati, tipo per anno il primo e anno per tipo il secondo.

### Integrazione tra prodotti

Un utilizzatore evoluto in genere non si accontenta dei risultati ottenuti con un prodotto, ma cerca di spingerne la utilizzazione oltre i suoi limiti, o se anche questo ambito gli è angusto, fa da sé le procedure di proprio interesse. È però evidente che realizzare software è in termini economici, ovvero in termini di ore-uomo impiegate, molto costoso. Se per esempio per realizzare un programma adatto alle nostre necessità impieghiamo 100 ore, proviamo a calcolarne il costo dando un valore in lire alla nostra ora di lavoro, zero lire in caso di studente squattrinato .. o 100.000 lire in caso di idraulico specialista.

Questo discorso non vale ovviamente per gli hobbysti, ma credo che nell'area dei PC, non esista hobbysta che in fondo non pensi che la sua attività con il computer non possa essere produttiva, o immediatamente, in quanto pensa di commercializzare il programma su cui sta lavorando, o a lungo termine, in quanto sta creandosi una «cultura» informatica «rivendibile» sul mercato del lavoro.

Un altro campo di attività su cui si cimenta spesso l'hobbysta è quello della traduzione dei prodotti americani, per lo meno per la parte relativa alla messaggistica del prodotto stesso. Ad esempio spesso l'utilizzabilità di un prodotto è compromessa dal fatto che i messaggi che questo invia sono in inglese. Le opzioni grafiche del Lotus 123, ad esempio, producono scritte tipo «millions» o «thousands» in maniera non facilmente controllabile, oppure le cifre in valuta vengono prodotte con il simbolo del dollaro.

Molti prodotti dispongono di «librerie» di simboli, di immagini raccolte per argomento, di piantine geografiche, e anche questi risentono della loro origine americana. L'utilizzatore italiano è anche in questo caso costretto ad arrangiarsi. Se si è in cerca di immagini per un certo lavoro è molto più facile trovare piantine geografiche dell'Alaska che non della Lombardia.

Per dare un senso pratico a quanto detto presentiamo un programma in Basic con il quale si può realizzare una piantina geografica, contenente ad esempio solo i confini delle provincie di una regione italiana.

Il risultato di questo programma viene parzialmente elaborato in Basic, in quanto i dati vengono memorizzati

in un semplice file sequenziale.

L'immagine prodotta viene «ripescata» con lo Storyboard, e sottoposta alla cura di bellezza, consistente nell'apposizione di scritte, nel riempimento delle aree, e, eventualmente, nell'aggiunta di altri elementi (vedi figg. 6,7).

Nel programma, listato in figura 4 e out in figura 5, si identificano le varie routine eseguite in sequenza.

La prima, da riga 110 e riga 120, predispone il video grafico ad alta definizione (screen 2) e ne stabilisce un'area di lavoro esattamente proporzionale a quella del digitizer.

Utilizziamo un digitizer Graphtec DT 1000, che «ragiona» in millimetri e su un'area di 2600 per 3800 decimi.

```

100 REM pseudo graphics
110 REM lettura colori
120 DIM C%(15):FOR I=0 TO 15:READ C%(I):NEXT
130 REM predisposizione schermo
140 KEY OFF:SCREEN 0,0,0:WIDTH 80:OUT 984,9
150 OUT 980,4:OUT 981,127:OUT 980,6:OUT 981,100
160 OUT 980,7:OUT 981,112:OUT 980,9:OUT 981,1
170 B=3:B1=B*17:DEF SEG=&HB800
180 FOR I%=0 TO 16382 STEP 2
190 POKE I%,222:POKE I%+1,B1:NEXT I%: PRINT CHR$(7)
191 GOTO 260
200 REM random
210 C=INT(RND*16):X0=INT(RND*16):Y0=INT(RND*10)
220 FOR X1=0 TO 9:FOR Y1=0 TO 9:X=X0*10+X1:Y=Y0*10+Y1
230 S=X AND 1:A=160*Y+(X OR 1)
240 POKE A,(PEEK(A) AND (15+S*225))+(C*(16-15*S))
250 NEXT Y1:NEXT X1:GOTO 210
260 REM palette
270 FOR M=0 TO 15:PRINT CHR$(7)
280 FOR N=0 TO 9:X=M*10+N:FOR Y=0 TO 99:C=C%(M)
290 S=X AND 1:A=160*Y+(X OR 1)
300 POKE A,(PEEK(A) AND (15+S*225))+(C*(16-15*S))
310 NEXT Y:NEXT N:NEXT M:DEF SEG
320 REM data colori
330 DATA 0,8,1,9,4,12,5,13,6,14,2,10,3,11,7,15

```

Figura 8 - Listato Programma in Basic I. Viene realizzata una modalità pseudo-grafica di 160 per 100 pixel a 16 colori, non presente tra quelle «classiche» del BASICA standard. Si parte dalla configurazione alfanumerica a colori permessa dalla scheda Color Graphic Adapter del PC IBM.

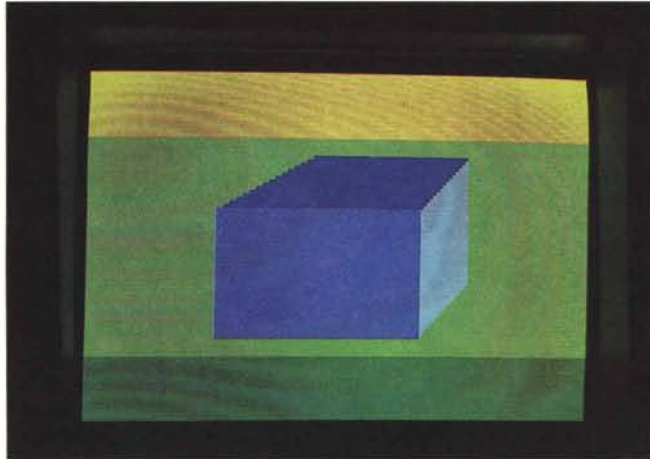
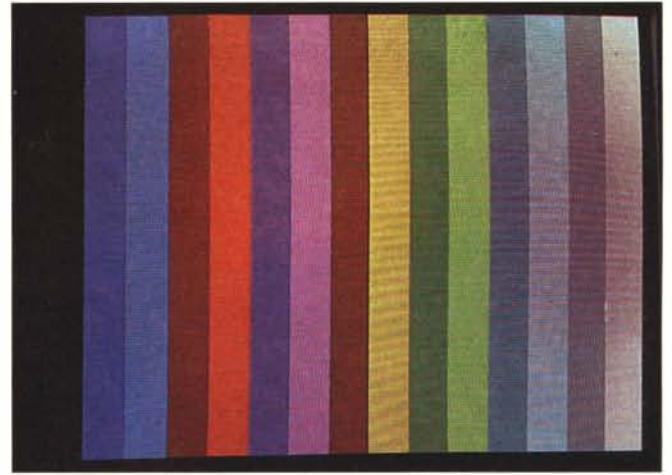
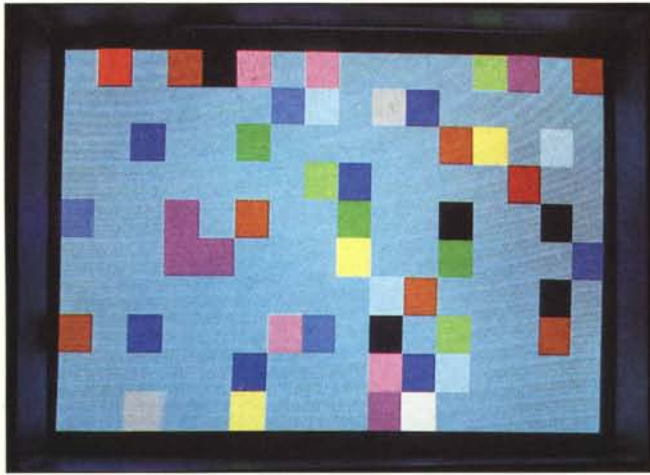


Figura 9, 10, 11  
Risultati. Sono i 16  
colori reali dello  
SCREEN 0, ma i  
pixel vanno  
maneggiati con  
routine scritte  
all'uopo, in quanto  
non sono riconosciute,  
ovviamente, le  
istruzioni grafiche  
BASICA che lavorano  
con gli SCREEN 1 e  
SCREEN 2.

In Basic va fatto lo «scaling» dell'immagine base, mentre è possibile spostarla o rifinirla con il programma autore. Con l'autore è possibile, anzi è meglio, fare il riempimento delle aree, in quanto si tratta di un'operazione estetica e non strutturale.

## Due programmi in Basic

Il Basic, pur essendo caduto in disuso per il vasto mondo degli utilizzatori dei PC compatibili non esperti e non interessati a diventarlo, è pur sempre il linguaggio che permette di sfruttare direttamente per mezzo di semplici comandi tutte le caratteristiche hard della macchina e in particolare tutte le caratteristiche grafiche.

Presentiamo due programmi, ri-elaborati da articoli apparsi su PC Magazine, che permettono di spingere un po' più in là i limiti di utilizzazione della grafica in Basic su PC. Il primo realizza una modalità pseudo-grafica (pseudo in quanto lavora su video alfanumerico) in 16 colori con una definizione di 160 per 100 pixel su video grafico 0 con modalità alfanumerica, il secondo fornisce una serie di funzioni tramite le quali, sul video grafico 2 (monocromatico con definizione di 640 per 200 pixel) si possono utilizzare altri colori, ovviamente uno per volta, oltre al nero.

Ambedue lavorano con la scheda IBM CGA, Color Graphic Adapter.

### Programma in Basic num. 1

Il programma, listato in figura 8, si divide in tre parti. Nella prima, da riga 130 a riga 190, viene predisposto lo schermo pseudo grafico, nella seconda viene attivata una routine di quadratura casuale e nella terza una routine di tracciamento righe verticali.

I 16 kbyte della mappa della memoria riservati alla grafica a colori risiedono sul CGA a partire dalla locazione &HB800 (47104 in decimale). Que-

La modalità di trasmissione è a caratteri ASCII, tramite la quale ad ogni lettura eseguita il digitizer invia, per mezzo dell'interfaccia seriale, una stringa di 12 caratteri, che vanno decodificati per mezzo di funzioni di stringa BASIC. Viene così identificato il tasto premuto sul puntatore, e la coppia di coordinate.

In riga 150 inizia la routine di decodifica, mentre la routine di accensione della periferica è in riga 190. Come si vede il protocollo di trasmissione è settato, via istruzione BASIC «OPEN COM», a 1200 baud, nessuna parità, 7 bit, 1 bit di stop.

I caratteri, inviati uno per volta, vengono accumulati su una stringa, la BS, che quando arriva a 12 caratteri, viene inviata alla decodifica.

Il programma è il più semplice possibile, vengono riconosciute solo tre posizioni del Flag F%, che identifica quale dei quattro tasti del tastierino numerico del puntatore è stato premuto. Le condizioni riconosciute sono quelle di fine linea, e quella di fine lavoro. La condizione di Default è quella di continuazione linea.

Durante l'immissione è possibile controllare la figura in quanto viene

eseguita la semplice routine di «plottaggio» di riga 240, che attiva l'istruzione PSET in caso di inizio linea e LINE in caso di continuazione.

Alla fine dell'immissione c'è un'ulteriore visualizzazione di controllo, prima della fase di memorizzazione del profilo (routine di riga 270).

La successiva routine è quella di scrittura del file sequenziale (riga 300).

La parte finale è un nuovo controllo dei dati, eseguito stavolta sui dati già immagazzinati nel file, ha come senso quello del controllo dell'andata a buon fine della scrittura del file (routine 340 predisposizione video, 380 lettura file e 410 visualizzazione finale).

L'immagine così realizzata, oppure semplicemente letta dal file, in caso di lavoro con un archivio di immagini, può essere prelevata con un prodotto che fissa la video memory, come il PSAVER del Polaroid Palette o il modulo Picture Taker dello Storyboard, e riutilizzata al di fuori del Basic.

Appare quindi evidente che alcune manipolazioni debbono essere fatte in Basic, altre possono essere fatte con il programma autore del prodotto, nel caso dello Storyboard con il Picture Maker.

st'area può essere utilizzata in tre modi.

Nel modo alfanumerico l'area si divide in 4 pagine da 80 colonne o in 8 pagine da 40 colonne. Ad ogni pagina da ottanta colonne sono quindi destinati 4000 byte, che per una videata di 80 per 25 caratteri significa 2 byte per carattere.

I due byte indicano il primo carattere ASCII e il secondo attributi di visualizzazione. Le varie tipologie e specifiche sono tutte richiamabili con gli specifici comandi BASICA tipo LOCATE, SCREEN, ecc. o con i comandi che hanno accesso diretto alla pagina video POKE, PEEK, OUT, ecc.

Nel nostro caso, tramite una serie di istruzioni OUT, con le quali vengono inviati byte al dispositivo video, e un loop di POKE si predispongono la pagina PSEUDO GRAFICA, sulla quale si potrà accedere solo con istruzioni POKE.

La routine di visualizzazione del pixel X, Y nel colore C, sarà:

```
REM entrano X, Y coordinate e C colore
S=X AND 1
A=160*Y+(X OR 1)
POKE A, (PEEK (A) AND (15+S*225))
+ (C*(16-15*S))
```

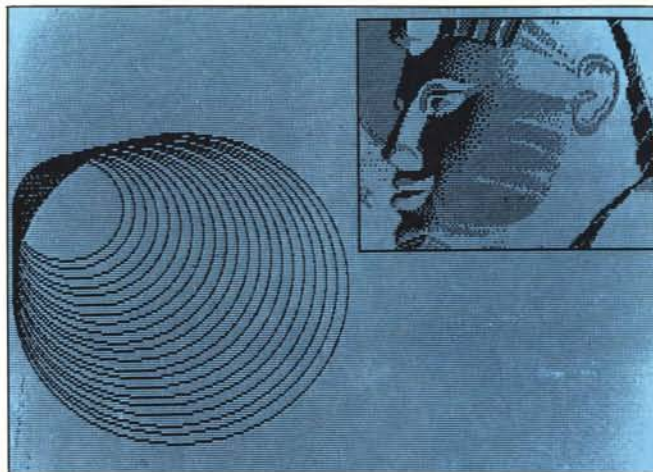
Tale metodo viene utilizzato in due routine che permettono non tanto di valutare la definizione in pixel, quanto le possibilità «cromatiche» del metodo stesso.

Con la prima, da riga 200 a riga 250, viene realizzata una routine di randomizzazione delle coordinate e del colore, e si disegnano quadratini di 10 per 10 pixel.

Con la seconda routine, da riga 260 alla fine, vengono eseguite delle righe verticali con i colori indicati nella sequenza dei dati. Questa routine può essere utile per valutare gli accostamenti cromatici di un disegno.

Quando si dispone di più colori la realizzazione di un disegno si complica, non solo in quanto il colore diventa un elemento che entra in ciascuna istruzione di tracciamento, ma in

Figura 13  
Out del Programma  
in Basic 2.  
Visualizziamo il  
risultato del  
programma  
precedente, applicato  
al disegno geometrico  
e a una immagine, la  
Sfinge delle librerie  
ExecuVision,  
rimanipolata per  
l'occasione.



quanto i vari colori devono essere messi con una certa logica.

Disponendo di pochi colori la logica è quella della campitura delle aree, con la quale i disegni sono composti da tante aree di ugual colore, ma disponendo di molti colori si possono ottenere effetti di «sfumatura» o chiaro scuro nei quali vanno accostati colori che differiscono di poco l'un l'altro. In questa situazione il programma diventa particolarmente pesante.

Oltre alla composizione dell'immagine diventa anche molto pesante la sua memorizzazione che richiede aree di memoria da 128 k in su e su un singolo dischetto di formato IBM possono entrarci anche una sola. Una immagine 640 per 350 pixel per 256 colori fa ... 256 kbyte.

Limiti impensabili fino a due tre anni fa, raggiungibili a costi alti oggi, ma che domani, con le nuove generazioni di microprocessori, di RAM, e con le ROM su Compact Disk saranno disponibili anche sugli Home Computer.

I due esempi realizzati, più un terzo che permette di valutare anche la definizione di 16.000 pixel, sono in figure 9, 10, 11.

## Programma Basic num. 2

Il secondo programma presentato, figura 12 e out in figura 13, mostra diverse utility sulla pagina video in modalità SCREEN 2, per la gestione dei due colori, o meglio dell'altro colore rispetto al nero.

Come test utilizziamo l'immagine della Sfinge già utilizzata per altri esperimenti, caricata con la routine da riga 120 a riga 140, che richiama la routine di caricamento da disco di riga 370, e un disegno geometrico di tanti cerchi uno dentro l'altro, routine da riga 150 a riga 180.

Le cose che il programma fa sono:

- con il loop di riga 140, chiuso con la riga 340, viene settato il colore alternativo;
- nella riga 190 avviene la memorizzazione dell'intera videata in un vettore di interi;
- nelle righe dalla 210 alla 250 la videata viene sparata con varie modalità;
- con la riga 270 l'immagine viene fatta sparire, e con la 290 riapparire;
- l'immagine viene invertita con un loop che «spazzola» tutta la pagina video.

Tutto qui.

MC

```
100 REM setta schermo di lavoro
110 SCREEN 2:CLS:DEF SEG=&HB800:OPTION BASE 1
120 REM carica sfinge
130 DIM A$(8200):GOSUB 370
140 FOR C=1 TO 15:OUT 985,C
150 REM disegno base
160 FOR I=50 TO 150 STEP 5
170 CIRCLE (I+20,I/3+70),I:NEXT I
180 REM memorizza in vettore l'immagine
190 GET (0,0)-(619,199),A$
200 REM varie visualizzazioni
210 PUT (0,0),A$,AND:GOSUB 360
220 PUT (0,0),A$,PSET:GOSUB 360
230 PUT (0,0),A$,PRESET:GOSUB 360
240 PUT (0,0),A$,XOR:GOSUB 360
250 PUT (0,0),A$,PSET:GOSUB 360

260 REM scompare l' immagine
270 OUT 984,18:GOSUB 360
280 REM riappare l' immagine
290 OUT 984,26:GOSUB 360
300 REM inversione
310 FOR I%=0 TO 16191
320 POKE I%,PEEK(I%) XOR 255
330 NEXT I:GOSUB 360
340 NEXT C:DEF SEG:END
350 REM attesa di un tasto
360 I$=INKEY$:IF I$="" THEN 360 ELSE RETURN
370 REM carica sfinge
380 OPEN "sfinge.dat" FOR INPUT AS 1
390 INPUT #1,D:DIM B$(D)
400 FOR I=1 TO D:INPUT #1,B$(I):NEXT I:CLOSE #1
410 PUT (330,10),B$,PSET:RETURN
```

Figura 12 - Listato Programma in Basic 2. In questo listato sono presenti varie modalità di uso dello SCREEN 2, che è monocromatico, ma il cui colore alternativo al nero può essere variato.

# Elettronica 2000

MISTER KIT

ELETRONICA APPLICATA, SCIENZA E TECNICA



SOUND  
LIGHT  
MANIA

L'ESCA A TRANSISTOR  
HI-FI CAR BOOSTER  
LA CHIAVE DIGITALE  
RIVELATORE SSB  
MINI BF MIXER  
MODEM HACKER NEWS

OGNI  
MESE  
IN  
EDICOLA!