Partiamo questo mese con il tanto atteso software per il C-128 proponendovi alcune routine di tutto rispetto che contribuiranno a svelare un po' di dubbi sul comportamento di questa macchina su cui ancora non esiste una documentazione sufficiente.

La prima che vi proponiamo, è un Boot editor con cui potrete editare una serie di istruzioni per il vostro dischetto che ne determinerà il comportamento quando viene data tensione al sistema, con il dischetto nel drive, o dopo un reset. A proposito del programma, ci è capitato che il bootstrap non andasse in esecuzione su un vecchio 1541. È bastato però dissaldare il filo collegato al piedino centrale del connettore posto sul cavo di collegamento tra drive e computer (pin 6 - reset) per mandare le cose a posto.

La seconda parte del software riguarda invece, in esclusiva, un argomento di cui molto si parla ma poco si riesce a fare perché mancano le dovute informazioni.

Con i brevi programmi dimostrativi, potrete finalmente fare della grafica sullo schermo da 80 colonne. Con essi, disegnerete con risoluzione 640x200 o, a scelta, 768x168 e 512x256.

Le varie routine nascono dall'esperienza diretta sulla macchina di due lettori che, a detta loro, le hanno provate tutte fino a che non sono riusciti ad ottenere qualcosa! Prendiamo quindi il loro breve articolo come degli appunti di laboratorio, cioè come l'estratto di un brogliaccio su cui si fanno alcune ipotesi di comportamento.

Naturalmente, le routine proposte funzionano egregiamente: provare per credere! Il loro uso è semplice. Basterà caricare per prima cosa la relativa routine di plottaggio (che rendiamo utilizzabile fornendola in un caricatore Basic) e poi caricare ed avviare il corrispondente programma di Hires.

Ad esempio, volendo disegnare la sinusoide proposta nell'esempio di risoluzione 640x200, caricheremo ed eseguiremo il programma plot 640x200. Fatto questo, caricheremo ed eseguiremo Hires 640x200.

A proposito sempre delle esperienze degli arditi lettori di cui stiamo parlando, abbiano un altro po' di risultati molto interessanti (racchiusi in una cospicua tabella riassuntiva) che vi proporremo la prossima volta.

In essi sono stati esaminati alcuni comportamenti del chip video che ci permettono effetti speciali, caratteri in doppia altezza, operazioni sul sincronismo del video e...

Per il momento ci fermiamo qui lasciandovi con un po' di curiosità. Naturalmente, ora che la corsa al C-128 è aperta, aspettiamo ansiosi i vostri programmi, le scoperte ed i trucchetti che riguardano questa macchina.

C-128 Boot Editor

di Felice Sobrero - Milano

Come avrete certamente notato, sul C-128 il sistema operativo CP/M si auto-inizializza, caricandosi ed installandosi da disco all'accensione del computer o all'atto del RESET: basta che sia presente nel drive l'apposito dischetto. Sarebbe interessante poter ottenere effetti simili con i propri dischi o, più in generale, poter determinare delle procedure di bootstrap automatico. È quanto vi propongo col mio programma.

Come funziona

All'accensione il sistema operativo, prima di presentare il prompt READY, controlla se un drive è collegato e acceso. Se sì, legge la traccia 1 settore 0. Esamina quindi il blocco, per vedere se i primi tre byte siano i caratteri Ascii CBM: trovato questo identificatore, esegue senz'altro qualunque routine in LM si trovi scritta sul blocco, più oltre. Se non trova l'identificatore, torna al Basic prompt READY.

Potremmo quindi anche noi prendere un disco ed incidere l'identificatore seguito da apposita routine in LM, ma il procedimento è lungo e poco pratico. Vi propongo un mezzo comodo per preparare qualsiasi vostro disco con una procedura auto-start da voi stessi definita, sia essa un messaggio,

Notizie sul sistema operativo

Per i patiti del monitor, riporto alcune notizie ricavate dalle mie esplorazioni nel sistema operativo del 128, a proposito del programma: (potete vederle con l'opzione M del monitor)

locazione dec. **209** = lunghezza in byte della KEY$
210 = offset dall'inizio dell'area delle stringhe KEY$, per giungere al testo da stampare.
Pokando queste due locazioni, si forza l'esecuzione.
locaz. decimali **4168 a 4177** = 10 byte indicano la lunghezza delle 10 KEY$, il cui contenuto è memorizzato subito dopo.
locazione dec. **215** = riporta il formato attuale dello schermo; il programma la testa per vedere se siamo in modo 80 o 40 colonne per regolarsi di conseguenza.

Per chi volesse eventualmente servirsi del Boot Editor per scopi di protezione consiglio di rendere invisibile la directory e di disabilitare il tasto RUN/STOP e RESTORE con POKE 808, 100; questo non permetterà di fermare il programma autostart. La manovra inversa consiste in POKE 808, 110.

```
1000 REM ********************************
1010 REM *                              *
1020 REM *      C-128 BOOT EDITOR       *
1030 REM *                              *
1040 REM *      DI FELICE SOBRERO       *
1050 REM *                              *
1080 REM ********************************
1090 REM
1100 COLOR0,7:COLOR4,7
1110 COLOR6,1:COLOR5,2
1120 PRINT CHR$(14):TB=0
1130 U1=7:U2=4:U3=39:U4=9
1140 U5=0:U6=0:U7=39:U8=24
1150 IF PEEK(215)=128 THEN BEGIN
1160 FAST:TB=20:U1=20:U2=4:U3=59:U4=8
1170 U5=0:U6=0:U7=79:U8=24: BEND
```

```
1180 WINDOW U5,U6,U7,U8,1
1190 SCNCLR:PRINTCHR$(17);CHR$(17)
1200 PRINT TAB(TB+6);"------------------------"
1210 PRINT TAB(TB+6);"!                      !"
1220 PRINT TAB(TB+6);"!  C-128 BOOT EDITOR  !"
1230 PRINT TAB(TB+6);"!                      !"
1240 PRINT TAB(TB+6);"------------------------"
1250 PRINT TAB(TB+3);CHR$(17);CHR$(17);"INSERIRE IL DISCO DA PREPARARE"
1260 PRINT TAB(TB+3);CHR$(17);"E PREMERE UN TASTO..."
1270 GETKEY A$:PRINTCHR$(7);
1280 PRINTCHR$(17);CHR$(17);TAB(TB+6);
1290 PRINT"ATTENDERE PREGO..."
1300 CLOSE15:OPEN15,8,15,"I":GOSUB1890
1310 SCNCLR
1320 FORA=1TO16:PRINT:NEXT
1330 PRINT TAB(TB);"------ CONTENUTO ATTUALE ASCII : ------":PRINT
1340 PRINTR$:COLOR5,2:PRINTCHR$(19);TAB(TB);
1350 PRINTCHR$(18);" (MAX 160 CAR.)  TERMINARE PREMENDO  ↑  ";CHR$(146)
1360 PRINT:PRINT:PRINT
1370 REM
1380 REM------ ROUTINE DI INPUT--------
1390 REM
1400 B$="←":CC$="":CC=0:CH%=0:A$=""
1410 PRINT"KEY 10,":PRINTCHR$(18);
1420 WINDOW U1,U2,U3,U4,1:FORA=1TO160:PRINTCHR$(32);:NEXT:PRINTCHR$(19);
1430 DO UNTIL CH%=94
1440 IF CC>160 THEN 1310
1450 TRAP 1400
1460 IFCH%=13 THEN PRINT B$;:ELSE PRINT A$;
1470 IF CH%=20 THEN BEGIN
1480 CC$=LEFT$(CC$,CC-2):CC=CC-1
1490 BEND:ELSE CC$=CC$+A$:CC=CC+1
1500 GETKEY A$
1520 CH%=ASC(A$)
1530 LOOP:PRINTCHR$(146);
1540 CC$=LEFT$(CC$,CC)
1550 CC$=CC$+CHR$(0)
1560 REM
1570 REM----- PREPARAZIONE DATI ------
1580 REM
1590 RESTORE:WINDOW U5,U6,U7,U8,1
1600 DO UNTIL X=96
1610 READ X
1620 IF X= -1 THEN BEGIN
1630 G$=G$+CHR$(CC-1)
1640 BEND:ELSE G$=G$+CHR$(X)
1650 LOOP:G$=G$+CC$
1660 REM
1670 REM--- SCRITTURA SETTORE 0 ------
1680 REM
1690 SCNCLR:PRINT:PRINT:PRINT
1700 PRINT"VALIDAZIONE E SCRITTURA DISCO..."
1710 PRINT:COLLECT
1720 CLOSE15:OPEN15,8,15
1730 CLOSE5:OPEN5,8,5,"#"
1740 PRINT#15,"B-P:5,0"
1750 PRINT#5,G$;
1760 PRINT#15,"B-A:0,1,0"
1770 IF DS=65THEN BEGIN
1780 PRINT
1790 PRINT "BLOCCO OCCUPATO!    "
1800 PRINT
1810 PRINT"POSSO SCRIVERCI SOPRA ? (S/N)"
1820 GETKEY A$:IF A$<>"S" THEN 1880
1830 PRINT:PRINT "OK..."
1840 PRINT#15,"B-F:0,1,0"
1850 BEND
1860 PRINT#15,"U2:5,0,1,0"
1870 PRINT:PRINT DS$
1880 CLOSE5:CLOSE15:END
1890 REM
1900 REM---- LETTURA SETTORE 0 --------
1910 REM
1920 CLOSE15:OPEN15,8,15:R$=""
1930 CLOSE5:OPEN5,8,5,"#"
1940 PRINT#15,"U1:5,0,1,0"
1950 FOR X=0 TO 33:GET#5,A$:NEXT
1960 FOR X=34 TO 195
1970 A=0:FLAG=0:GET#5,A$
1980 IFA$=CHR$(0)THEN2030:ELSE A=ASC(A$)
1990 IFA=13THEN R$=R$+CHR$(95):GOTO2030
2000 IF((A>31)AND(A<96))OR((A>182)AND(A<219)) THEN FLAG=1
2010 IF FLAG THEN R$=R$+CHR$(A):ELSE R$=R$+CHR$(32)
2020 BEND
2030 NEXT X
2040 CLOSE5:CLOSE15
2050 RETURN
2060 REM
2070 REM
2080 REM------- ROUTINE L/M -----------
2090 REM
2100 DATA67,66,77,0,0,0,0,0,0
2110 DATA169,-1,141,9,16
2120 DATA162,0,189,34,11,157,72,16,232,208,247
2130 DATA169,-1,133,209
2140 DATA169,62,133,210
2150 DATA96
2160 REM
2170 REM---------- FINE ---------------
```

un caricamento immediato di un programma o di un altro loader o, perché no, un bel GO 64 per tutti i vecchi dischi e programmi del glorioso sessantaquattro.

Il programma vi mette a disposizione 160 byte per scriverci qualunque comando accettabile in modo diretto, in Basic 7.0, per un totale di 4 righe di schermo, compresa la virgola, i doppi apici e il return, dato anche più volte. Il tutto sarà trasformato in modo che, al reset, il C-128 leggerà il blocco 0, ridefinirà in un attimo la KEY 10 (che normalmente contiene HELP), e ne forzerà l'esecuzione immediata. Con le opportune differenze, questa tecnica assomiglia alla programmazione del buffer di tastiera, di cui si è ampiamente parlato su MCmicrocomputer a proposito del C64.

Infine, ad operazione avvenuta, il tasto HELP che, ripeto, è la KEY 10, resterà addetto al nostro comando. Invariati gli altri tasti di funzione F1-F8 è il CHR$(131) o RUN/STOP/SHIFT, che corrisponde per il sistema operativo alla KEY 9.

Il programma è autoesplicativo. Sono mostrati sullo schermo i limiti del testo Basic inseribile. Siccome è di per sé instampabile, il RETURN sarà visualizzato come freccia sinistra, per analogia con i word-processor. Tutto il resto appare come sarà eseguito. È possibile usare INST/DEL per correzioni.

Esempi

RUN"*" ← caricherà ed eseguirà il 1° prg basic del disco

BOOT"*" ← caricherà ed eseguirà il 1° prg LM del disco.

Quasi comico sarà l'uso di GO64←YES←! Il computer infatti si domanderà se è sicuro e si risponderà di sì in una frazione di secondo, poi apparirà tutto da solo il fatidico messaggio di cold star del 64.

Sono naturalmente possibili anche comandi abbastanza complessi, tipo: PRINT"rosso":COLOR 4, 1:PRINT" BUONGIORNO":DLOAD"MENU": ecc... ecc... ecc...

Il programma controlla da solo se, dopo validazione, il blocco 0 risulta disponibile. Se non lo fosse, significa che fa parte di un file della directory e ci avvertirà prima di combinare guai.

Il tutto tramite i comandi DOS Block-Allocate per il controllo e poi U2 per la scrittura. Alla fine viene letto il canale di errore per sapere l'esito dell'operazione. Non resta che premere il pulsante di reset o spegnere e riaccendere il C-128 per provare!

L'integrato 8563: 80 colonne e 640x200

di Stefano Lonardi
e Valerio Tagliabue - Verona

Questo integrato, di cui per ora non si sa quasi nulla, supporta le 80 colonne e una non meglio specificata grafica 640x200.
Può darsi che alcune delle nostre considerazioni che seguono, e che abbiamo ricavato dalle prove pratiche effettuate, saranno chiarite dalla tanto attesa «Guida di riferimento» quindi i lettori dovranno avere un po' di pazienza per qualche notizia non tanto precisa.

Saremo comunque felici di confrontare i nostri risultati con quelli di altri 128ttisti.

A nostro giudizio oltre ai 128Kb di RAM di cui è provvisto il computer ve ne sono 16 dedicati all'8563 che sono organizzati in questo modo:

Screenmemory	0000-07FF	2K
Color-Status memory.	0800-0FFF	2K
Free memory	1000-1FFF	4K
Char memory	2000-3FFF	8K

Purtroppo essendo questa zona distaccata non è possibile accedervi direttamente con il monitor.

Al momento dell'accensione (o del Reset) il S.O. ricopia la charrom delle 40 colonne nella ram del chip video ad 80 colonne tramite la routine allocata all'indirizzo $FCE0C.

L'8563 ha due registri allocati rispettivamente in $fd600 (che chiameremo reg. 0) e in $fd601 (che chiameremo reg. 1). Non si tratta però delle solite locazioni tipo SID, CIA, VIC-II che hanno funzioni ben determinate e fisse.

Il nostro chip è una specie di microprocessore al quale bisogna fornire il codice dell'istruzione sul reg. 0 e il dato sul reg. 1.

Provando una alla volta i codici abbiamo composto una cospicua tabella *(che sarà pubblicata sul prossimo numero, n.d.r.)*; ad esempio, per entrare nel modo grafico si dovrà caricare nel registro 0 l'istruzione $19 e nel registro 1 il dato $87.

A questo punto però non è possibile intervenire direttamente nelle locazioni dell'area in alta risoluzione. Si dovrà perciò utilizzare una routine presente all'indirizzo $FCDCA che permette di mandare al chip un byte tramite l'istruzione $1F. Utilizzata nel modo testo, stampa il carattere corrispondente mentre, nel modo grafico, memorizza semplicemente il dato.

Naturalmente allo stesso modo è possibile leggere un byte o il codice relativo ad un carattere nell'area video ($FCDD8).

Queste due routine dovranno essere accompagnate con l'informazione riguardante il «dove» mandare il byte all'interno dei 16K. Le istruzioni da eseguire sono la $12 (nel reg. 0), la parte alta dell'indirizzo (nel reg. 1) e poi la $13 con la parte bassa dell'indirizzo.

Se, ad esempio, vogliamo azzerare tutti 16K utilizzeremo una routine del tipo che segue:

```
      LDY #$3F   ; numero pagine.
      LDX #$00   ; contatore byte.
      LDA #$12   ; istruzione $12.
      STA $D600
      LDA #$00   ; azzera puntatore (H).
      STA $D601
      LDA #$13   ; istruzione $13.
      STA $D600
      LDA #$00   ; azzera puntatore (L).
      STA $D601
LOOP  LDA #$1F   ; istruzione $1F.
      STA $D600
WAIT  BIT $D600  ; aspetta il 'ready'.
      BPL WAIT
      LDA #$00   ; manda il byte 00.
      STA $D601
      DEX
      BNE LOOP
      DEY        ; * ripete $3FFF volte.
      BNE LOOP
      RTS
```

Come si può notare dopo aver azzerato il puntatore non ci siamo più curati di quest'ultimo: infatti esso si incrementa automaticamente quando scrive o legge una locazione.

Vogliamo ora proporvi una routine in linguaggio macchina per il Plot: ci sembra infatti abbastanza valida in quanto è priva di cicli (e quindi un po' più lunga di quanto potrebbe essere), ma risulta essere più veloce. Per poterla utilizzare da Basic si dovrà «pokare» il valore di Y nella locazione $FA (250) ed X in due, dividendolo in parte alta e bassa ($FE, $FF). Il listato è nella figura 1.

Vorremmo farvi notare una particolarità, che nelle prime ci era sfuggita, delle istruzioni $01 e $06 che definiscono il numero di colonne e di righe della pagina testo.

Utilizzandole nella pagina grafica per modificare il numero di pixel vi-

Figura 1

```
LDX $FA  ; Duplica $FA in $FC e si ha
STX $FC  ; (coord.Y) => (FA-FB) => (FC-FD)
LDX #$00
STX $FB
STX $FD

ASL $FA  ; Esegue (FA-FB)*64 => (FA-FB)
ROL $FB  ; shiftando a sinistra 6 volte (FA-FB).
ASL $FA
ROL $FB
ASL $FA
ROL $FB
ASL $FA
ROL $FB
ASL $FA
ROL $FB
ASL $FA
ROL $FB

ASL $FC  ; Esegue (FC-FD)*16 => (FC-FD)
ROL $FD  ; shiftando a sinistra 3 volte (FC-FD).
ASL $FC
ROL $FD
ASL $FC
ROL $FD

CLC      ; Esegue la somma a 16 bit
LDA $FA  ; (FA-FB) + (FC-FD) => (FA-FB).
ADC $FC
STA $FA
LDA $FB
ADC $FD
STA $FB

LDA $FE  ; Salva gli ultimi 3 bit della coordinata X
AND $#07 ; relativi al pixel all'interno del byte.
STA $F9
```

HIRES 640x200

Esempio di utilizzo della grafica gestibile con l'8563:«plotta» la funzione del seno utilizzando la routine LM relativa. La prima parte riguarda l'inizializzazione della pagina grafica. Segue il tracciamento del reticolo e il «plottaggio» della funzione stessa.

Routine	Parametri	SYS	Descrizione
PLOT	0<= X <=639 0<= Y <=199	12288	Poke 250 , Y Poke 254 , X and 255 Poke 255 , X / 256
CLEAR		12430	Cancella la pagina.
ORIZZON.	Come per il Plot	12463	Traccia una linea orizzontale di lunghezza X partendo dal punto (0,Y).
VERTIC.	Come per il Plot	12495	Traccia una linea verticale di lunghezza Y partendo dal punto (X,0).

PLOT 640x200

Routine LM allocata da $3000. In essa sono presenti il PLOT, il CLEAR e il tracciamento di linee orizzontali e verticali e sono così distribuite:

```
10 REM ------------------------ PLOT 640X200 -------------------------------
20 '
30 FORI=12288TO12521:READ A:POKEI,A:NEXT
12288 DATA 166,250,134,252,162, 0,134,251,134,253, 6,250, 38,251, 6,250
12304 DATA  38,251, 6,250, 38,251, 6,250, 38,251, 6,250, 38,251, 6,250
12320 DATA  38,251, 6,252, 38,253, 6,252, 38,253, 6,252, 38,253, 6,252
12336 DATA  38,253, 24,165,250,101,252,133,250,165,251,101,253,133,251,165
12352 DATA 254, 41, 7,133,249, 70,255,102,254, 70,255,102,254, 70,255,102
12368 DATA 254, 24,165,250,101,254,133,250,165,251,101,255,133,251, 32,112
12384 DATA  48, 32,216,205,166,249, 29,133, 48, 32,112, 48, 32,202,205, 96
12400 DATA 162, 18,142, 0,214,166,251,142, 1,214,162, 19,142, 0,214,166
12416 DATA 250,142, 1,214, 96,128, 64, 32, 16, 8, 4, 2, 1, 0,162, 0
12432 DATA 134,250,134,251, 32,112, 48,160, 63,169, 31,141, 0,214, 44, 0
12448 DATA 214, 16,251,169, 0,141, 1,214,232,208,238,136,208,235, 96,164
12464 DATA 250,166,254,134,180,166,255,134,181,132,250,166,180,134,254,166
12480 DATA 181,134,255, 32, 0, 48,198,180,208,239,198,181, 16,235, 96,164
12496 DATA 250,166,254,134,180,166,255,134,181,132,250,166,180,134,254,166
12512 DATA 181,134,255, 32, 0, 48,136,208,240,96
```

Naturalmente in caso di programmi molto lunghi sarà opportuno allocare più in alto la routine curandosi di modificare in modo appropriato i salti assoluti.

```
------------ HIRES 640 X 200 -----------

40 BANK15:FAST:TRAP 380
50 REM"_____________________________________
60 REM"*************** COLORS *****************
70 REM"_____________________________________
80 POKEDEC("D600"),DEC("1A")
90 POKEDEC("D601"),DEC("81")
100 REM"____________________________________
110 REM"*************** HI-RES ON **************
120 REM"____________________________________
130 POKEDEC("D600"),DEC("19")
140 POKEDEC("D601"),DEC("87")
150 REM"____________________________________
160 REM"************ CLEAR HI RES **************
170 REM"____________________________________
180 SYS 12430:REM HIRES (80*8=640)X(25*8=200)
190 REM"____________________________________
200 REM"************ DRAW X/Y AXIS **************
210 REM"____________________________________
220 FOR Y=0 TO 199 STEP 39.8
230 : POKE250,Y:POKE255,639/256:POKE254,639AND255:SYS12
240 NEXT
250 FOR X=0 TO 639 STEP 79.8
260 : POKE250,199:POKE255,X/256:POKE254,XAND255:SYS1249
270 NEXT
280 REM"____________________________________
290 REM"************ PLOT FUNCTION **************
300 REM"____________________________________
310 FOR X=0 TO 639
320 : Y=(SIN(X/100)*100)+100
330 : POKE 255,X/256:POKE254,X AND 255
340 : POKE 250,Y:SYS12288
350 NEXT
360 GETKEY A$
370 REM"____________________________________
380 REM"************* HI-RES OFF ***************
390 REM"____________________________________
400 POKEDEC("D600"),DEC("19")
410 POKEDEC("D601"),DEC("07")
420 PRINT"□":SYS 52748
```

```
LSR $FF   ; Esegue (FE-FF)/16 => (FE/FF)
ROR $FE   ; shiftando a destra 3 volte (FE-FF)
LSR $FF
ROR $FE
LSR $FF
ROR $FE

CLC       ; Esegue la somma a 16 bit
LDA $FA   ; (FA-FB) + (FE-FF) => (FA-FB) che e'
ADC $FE   ; finalmente l'indirizzo effettivo del byte.
STA $FA
LDA $FB
ADC $FF
STA $FB

JSR POINT    ; Punta gli indici all'indirizzo effettivo.
JSR $CDD8    ; Carica il byte gia' presente sulla pagine Hires.
LDX $F8      ; Recupera gli ultimi 3 bit della coordinata X ...
ORA TABLE,X  ; e in base ad esso estrae il valore dalla tavola.
JSR POINT    ; Ri-punta gli indici.....
JSR $CDCA    ; e finalmente plotta il punto.
RTS

$80,$40$,$20,$10,$08,$04,$02,$01

LDX #$12     ; Istruz. $12 per il posizionamento dell'indice (H).
STX $D600
BIT $D600
BPL WAIT1    ; Aspetta il Ready.
LDX $FB
STX $D601
LDX #$13     ; Istruz. $13 per il posizionamento dell'indice (L).
STX $D600
BIT $D600
BPL WAIT2    ; Aspetta il Ready.
LDX $FC
STX $D601
RTS
```

Naturalmente, la routine di Plot sopra descritta andrà modificata in modo opportuno a seconda del tipo di grafica scelta.

N.B.: cambiando la risoluzione si dovrà «centrare» lo schermo usando le istruzioni $02 e $07 (ne parleremo la prossima volta).

sualizzabili abbiamo ottenuto risultati stupefacenti.

Pensate che siamo riusciti ad ottenere una risoluzione a tutto schermo di 768 punti!

L'integrato permette di arrivare oltre i 950 punti, ma purtroppo non sono visualizzabili normalmente (bisognerà «restringere» lo schermo).

Usando queste maxi-risoluzioni c'è, naturalmente, un inconveniente: l'8563 ha a disposizione solo 16K di memoria e se i punti visualizzati sono più di 131072 (cioè 16384*8) i pixel «sovrabbondanti» saranno copiati dalla stessa area di memoria e perciò risulteranno uguali.

Seguendo questo ragionamento, se raddoppieremo la risoluzione orizzontale, dovremo dimezzare quella verticale.

I «formati» che ci sono sembrati più interessanti sono, oltre alla 640x200, la 768x168 e in particolare la 512x256.

Risoluzione 640 x 200

```
100 REM"_____
110 REM **********
120 REM        TOROIDE
130 REM     TEMPO:6 MIN          640x200
140 REM **********
150 REM"_____

170 FAST:BANK15:SYS12430:TRAP510
180 REM"_____
190 REM ************ HI RES ON ***************
200 REM"_____
210 POKE DEC("D600"),DEC("19")
220 POKE DEC("D601"),DEC("87")
230 REM"_____
240 REM"************** CONSTANTS ***************
250 REM"_____
260 P=π/180
270 C=10:D=10:CC=COS(C*P):CD=COS(D*P):SC=SIN(C*P):SD=SIN(D*P)
280 RAGGIO=90:XCENTRO=320:YCENTRO=100:N=70
290 REM"_____
300 REM ********** CALCULATING **************
310 REM"_____
320 PN=2*π/N
330 FORI=0TON/2:A=I*PN-π/2
340 :FORL=0TON :B=L*PN
350 : X=RA*COS(A)*COS(B)
360 : Y=RA*COS(A)*SIN(B)
370 : Z=RA*SIN(A)*COS(B)
380 : XS=XC-(X*CC+Y*CD)*2
390 : YS=YC-X*SC+Y*SD+Z
400 : REM"_____
410 : REM ********** PLOT ROUTINE ***********
420 : REM"_____
430 : POKE255,XS/256:POKE254,XS AND 255
440 : POKE250,YS:SYS12288
450 :NEXT
460 NEXT
470 GETKEY A$
480 REM"_____
490 REM ************ ERROR *****************
500 REM"_____
510 POKE DEC("D600"),DEC("19")
520 POKE DEC("D601"),DEC("07")
530 PRINT"□":SYS52748
```

HIRES 768x168

Questo programma ricalca in parte quello relativo alla 640x200 eccetto per alcune istruzioni che devono modificare il formato grafico e che centrano la pagina.

In questo caso il programma utilizza una routine differenziata rispetto a quella precedente.

```
---------- HIRES 768 X 168 ----------

40 FAST:TRAP 560:BANK 15
50 REM"_____
60 REM"**************** COLORS ****************
70 REM"_____
80 POKE DEC("D600"),DEC("1A")
90 POKE DEC("D601"),DEC("81")
100 REM"_____
110 REM"************** HI-RES ON ***************
120 REM"_____
130 POKE DEC("D600"),DEC("19")
140 POKE DEC("D601"),DEC("87")
150 REM"_____
160 REM"********* # RIGHE DISPLAY *************
170 REM"_____
180 POKE DEC("D600"),DEC("06")
190 POKE DEC("D601"),21
200 REM"_____
210 REM"********* # COLONNE DISPLAY ***********
220 REM"_____
230 POKE DEC("D600"),DEC("01")
240 POKE DEC("D601"),96
250 REM"_____
260 REM"** CENTRA LA PAGINA IN ORRIZONTALE ***
270 REM"_____
280 POKE DEC("D600"),DEC("02")
290 POKE DEC("D601"),100
300 REM"_____
310 REM"*** CENTRA LA PAGINA IN VERTICALE ****
320 REM"_____
330 POKE DEC("D600"),DEC("07")
340 POKE DEC("D601"),30
350 REM"_____
360 REM"************ CLEAR  HI-RES *************
370 REM"_____
380 SYS 12444:REM" HIRES (96*8=768)X(21*8=168)
390 REM"_____
400 REM"********** DRAW X/Y AXIS **************
410 REM"_____
420 FOR Y=0 TO 167 STEP 41.75
430 : POKE250,Y:POKE254,767AND255:POKE255,767/256:SYS12477
440 NEXT
450 FOR X=0 TO 767 STEP 62.6
460 : POKE250,167:POKE254,XAND255:POKE255,X/256:SYS12509
470 NEXT
480 REM"_____
490 REM"************ PLOT FUNCTION *************
500 REM"_____
510 FOR X=0 TO 767
520 : Y=(SIN(X/79)*84)+84
530 : POKE250,Y:POKE254,XAND255:POKE255,X/256:SYS12288
540 NEXT
550 GETKEYA$
560 REM"_____
570 REM"************** ALL O.K. ****************
580 REM"_____
590 POKE DEC("D600"),DEC("19")
600 POKE DEC("D601"),7
610 POKE DEC("D600"),DEC("01")
620 POKE DEC("D601"),80
630 POKE DEC("D600"),DEC("06")
640 POKE DEC("D601"),25
650 POKE DEC("D600"),DEC("07")
660 POKE DEC("D601"),32
670 POKE DEC("D600"),DEC("02")
680 POKE DEC("D601"),100
690 PRINT"□":SYS 52748
```

PLOT 768x168

Anche questa è allocata in $3000 ed è composta dagli stessi «sottoprogrammi»:

Routine	Parametri	SYS	Descrizione
PLOT	0<= X <=767 0<= Y <=167	12288	Poke 250 , Y Poke 254 , X and 255 Poke 255 , X / 256
CLEAR		12444	Cancella la pagina.
ORIZZON.	Come per il Plot	12477	Traccia una linea orizzontale di lunghezza X partendo dal punto (0,Y).
VERTIC.	Come per il Plot	12509	Traccia una linea verticale di lunghezza Y partendo dal punto (X,0).

```
10 REM ---------- PLOT 768X168 ----------
20 :
30 FORI=12288TO12535:READA:POKEI,A:NEXT
12288 DATA 166,250,134,252,162, 0,134,251,134,253, 6,250, 38,251, 6,250
12304 DATA  38,251, 6,250, 38,251, 6,250, 38,251, 6,250, 38,251, 6,250
12320 DATA  38,251, 6,252, 38,253, 6,252, 38,253, 6,252, 38,253, 6,252
12336 DATA  38,253, 6,252, 38,253, 24,165,250,101,252,133,250,165,251,101
12352 DATA 253,133,251,165,254, 41, 7,133,249, 70,255,102,254, 70,255,102
12368 DATA 254, 70,255,102,254, 24,165,250,101,254,133,250,165,251,101,25
12384 DATA 133,251, 32,116, 48, 32,214,205,166,249, 29,147, 48, 32,116, 4
12400 DATA  32,202,205, 96,162, 18,142, 0,214, 44, 0,214, 16,251,166,251
12416 DATA 142, 1,214,162, 19,142, 0,214, 44, 0,214, 16,251,166,250,142
12432 DATA  1,214, 96,128, 64, 32, 16, 8, 4, 2, 1, 0,162, 0,134,250
12448 DATA 134,251, 32,116, 48,160, 63,169, 31,141, 0,214, 44, 0,214, 16
12464 DATA 251,169, 0,141, 1,214,232,208,238,136,208,235, 96,164,250,166
12480 DATA 254,134,180,166,255,134,181,132,250,166,180,134,254,166,181,13
12496 DATA 255, 32, 0, 48,198,180,208,239,198,181, 16,235, 96,164,250,166
12512 DATA 254,134,180,166,255,134,181,132,250,166,180,134,254,166,181,13
12528 DATA 255, 32, 0, 48,136,208,240, 96
```

HIRES 512x256

Valgono le stesse considerazioni di cui sopra.

```
40 FAST TRAP%60 BANK 15
50 REM"_____
60 REM"**************** COLORS ****************
70 REM"-----
80 POKE DEC("D600"),DEC("1A")
90 POKE DEC("D601"),DEC("81")
100 REM"_____
110 REM"************** HI-RES ON ***************
120 REM"-----
130 POKE DEC("D600"),DEC("19")
140 POKE DEC("D601"),DEC("87")
150 REM"_____
160 REM"********* # RIGHE DISPLAY ************
170 REM"-----
180 POKE DEC("D600"),DEC("06")
190 POKE DEC("D601"),32
200 REM"_____
210 REM"********* # COLONNE DISPLAY **********
220 REM"-----
230 POKE DEC("D600"),DEC("01")
240 POKE DEC("D601"),64
250 REM"_____
260 REM"** CENTRA LA PAGINA IN ORIZZONTALE ***
270 REM"-----
280 POKE DEC("D600"),DEC("02")
290 POKE DEC("D601"),92
300 REM"_____
310 REM"**** CENTRA LA PAGINA IN VERTICALE ***
320 REM"-----
330 POKE DEC("D600"),DEC("05")
340 POKE DEC("D601"),35
350 REM"_____
360 REM"************ CLEAR HI-RES ************
370 REM"-----
380 SYS 12408 REM" HIRES (64*8=512)X(32*8=256)
390 REM"_____
400 REM"************ DRAW X/Y AXIS ************
410 REM"-----
420 FOR Y=0 TO 255 STEP 31.875
430   POKE 250,Y:POKE255,511/256:POKE254,511AND255:SYS12441
440 NEXT
450 FOR X=0 TO 511 STEP 63.875
460   POKE 250,255:POKE255,X/256:POKE254,XAND255:SYS12473
470 NEXT
480 REM"_____
490 REM"********* PLOT FUNCTION **************
500 REM"-----
510 FOR X=0 TO 511
520   Y=SIN(X/80)*120+120
530   POKE 250,Y:POKE255,X/256:POKE254,XAND255:SYS12288
540 NEXT
550 GETLEVA1
560 REM"_____
570 REM"************** ALL O.K. ***************
580 REM"-----
590 POKE DEC("D600"),DEC("19")
600 POKE DEC("D601"),7
610 POKE DEC("D600"),DEC("01")
620 POKE DEC("D601"),80
630 POKE DEC("D600"),DEC("06")
640 POKE DEC("D601"),25
650 POKE DEC("D600"),DEC("07")
660 POKE DEC("D601"),32
670 POKE DEC("D600"),DEC("02")
680 POKE DEC("D601"),100
690 PRINT"3" SYS52748
```

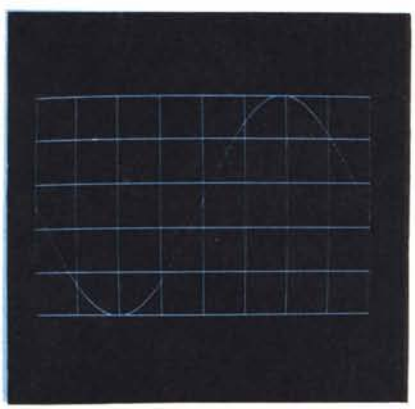
PLOT 512x256

Simile alla precedente e cosi formata:

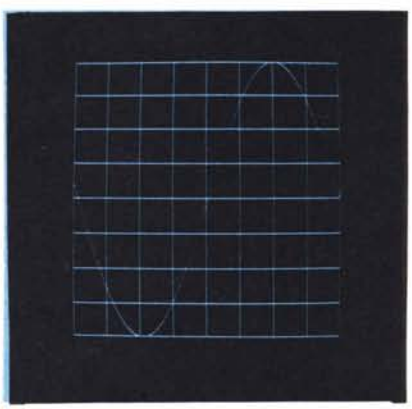
Routine	Parametri	SYS	Descrizione
PLOT	0<= X <=511 0<= Y <=255	12288	Poke 250 , Y Poke 254 , X and 255 Poke 255 , X / 256
CLEAR		12408	Cancella la pagina.
ORIZZON.	Come per il Plot	12441	Traccia una linea orizzontale di lunghezza X partendo dal punto (0,Y).
VERTIC.	Come per il Plot	12473	Traccia una linea verticale di lunghezza Y partendo dal punto (X,0).

```
10 REM ------------------------ PLOT 512X256 ------------------------
20
30 FOR I=12288TO12499:READA:POKEI,A:NEXT
12288 DATA 166,250,162, 0,134,251, 6,250, 38,251, 6,250, 38,251, 6,250
12304 DATA  38,251, 6,250, 38,251, 6,250, 38,251, 6,250, 38,251,165,254
12320 DATA  41, 7,133,252, 70,255,102,254, 70,255,102,254, 70,255,102,254
12336 DATA  24,165,250,101,254,133,250,165,251,101,255,133,251, 96, 80, 48
12352 DATA  32,216,205,166,252, 29,111, 48, 32, 80, 48, 32,202,205, 96,234
12368 DATA 162, 18,142, 0,214, 44, 0,214, 16,251,166,251,142, 1,214,142
12384 DATA  19,142, 0,214, 44, 0,214, 16,251,166,250,142, 1,214, 96,138
12400 DATA  64, 32, 16, 8, 4, 2, 1, 0,162, 0,134,250,134,251, 32, 80
12416 DATA  48,160, 64,169, 31,141, 0,214, 44, 0,214, 16,251,169, 0,141
12432 DATA  1,214,232,208,238,136,208,235, 96,164,250,166,254,134,180,166
12448 DATA 255,134,181,132,250,166,180,134,254,166,181,134,255, 32, 0, 48
12464 DATA 198,180,208,239,198,181, 16,235, 96,164,250,166,254,134,180,166
12480 DATA 255,134,181,132,250,166,180,134,254,166,181,134,255, 32, 0, 48
12496 DATA 136,208,240, 96
```

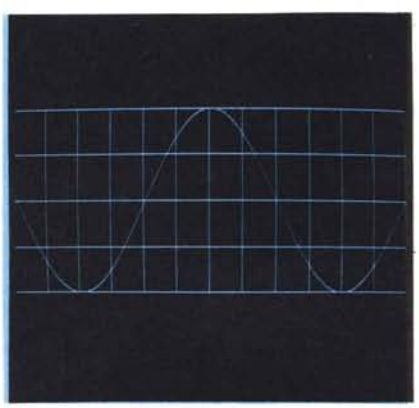
ESEMPI DI GRAFICA IN ALTA RISOLUZIONE PER IL C-128



Hires 640× 2.



Hires 512× 256.



Hires 768× 168.

MC