



Grafica 640x200

prima parte

■ Dedicheremo alcuni articoli di 128 da zero a un bit del Commodore 128. Precisamente al bit 7 del registro 25 del video display chip 8563 che permette di selezionare il modo grafico 640x200, altrimenti non disponibile. ■

Il vizio

Mamma Commodore certamente non è nuova per queste cose: poteva mai fare un computer completo di tutto?

A dire il vero una volta, col Plus 4 un tentativo l'aveva fatto: è stato un vero fallimento, solo pochi esemplari venduti prima di uscire rapidamente di produzione. Sarà forse perché l'utente medio di computer Commodore smanettomane per antonomasia (eccezion fatta per quelli che comprano il 64 solo per giocare). E a dire il vero, a «noi» smanettomani, i computer chiusi, non aperti cioè a nuove esperienze hard e soft, fanno un po' schifo. Il 128 non fa eccezioni: grafica 640x200 solo se sei in grado di farla saltare fuori. Il bello è... senza che il costruttore mi dica come fare (su quel maledetto «manuale» fornito con la macchina).

Fortunatamente un po' di documentazione non ufficiale siamo riusciti a

trovarla (naturalmente oltre oceano) e il modo grafico ad altissima risoluzione salta fuori settando un opportuno bit dell'8563. Poi però bisogna fare un po' di routine in linguaggio macchina per pulire il video grafico, per plottare punti, linee, rettangoli, cerchi, riempire aree, scrivere caratteri, e tanti altri eccetera-eccetera limitati solo dalla fantasia di chi programma.

Questo mese vedremo come attivare la pagina grafica, pulire il video e plottare i punti: il minimo indispensabile per fare qualche applicazione. Infine presenteremo un programma MATH PACK (si è proprio il solito, ndr) che permette di studiare funzioni matematiche reali di variabile reale (quasi, viste le limitazioni intrinseche dei computer).

Grafica 640x200

Per passare al modo grafico di altissima risoluzione basta scrivere il valo-

re \$ 87 (decimale 135) nel registro 25 del VDC. \$ 87 per settare il bit 7 di tale registro e scorrere la pagina video tutta a destra in modo da non mostrare quell'orribile brulichio di pixel (tavanata galattica) presente ad esempio nell'apertura di nobili programmi come Superscript e Superbase ad opera della Precision Software.

I colori disponibili in tale grafica sono 16 anche se visualizzabili solo uno alla volta, uguale per tutti i pixel in campo. A tale scopo, i 4 bit più significativi del registro 26 permettono di scegliere il colore della grafica, nel modo già visto precedentemente per i caratteri e per il colore di fondo, ovvero specificando quali componenti cromatiche (blu, rosso, verde) vogliamo e se desideriamo l'intensità doppia.

In definitiva, per selezionare la pagina grafica è sufficiente la linea basic:

POKE DEC ("D600"), 25:POKE DEC ("D601"), 135

Figura 1

```

10 POKE DEC("D600"),25:POKE DEC("D601"),135
20 POKE DEC("D600"),18:POKE DEC("D601"),0
30 POKE DEC("D600"),19:POKE DEC("D601"),0
40 FOR I=1 TO 16000
50 SYS 52682,0
60 NEXT

```

Figura 3

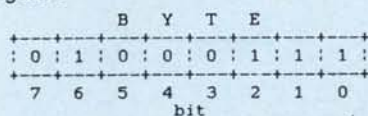
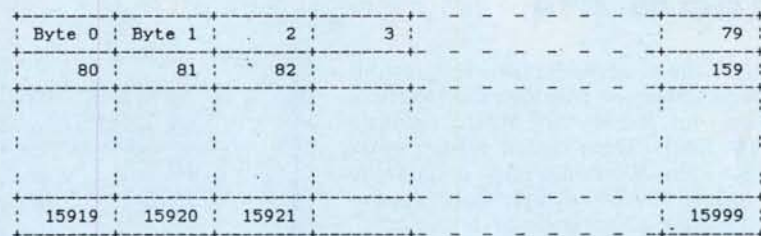


Figura 2



Ricordiamo che le due poke devono essere date sulla stessa linea, pena il non riconoscimento del comando, da parte dell'8563. Per tornare al modo testo (se ancora non abbiamo modificato la video RAM) basta eseguire la linea:

```
POKE DEC ("D600"), 25:POKE DEC ("D601"), 71
```

Chi avrà provato ad accedere all'altissima risoluzione con le poke viste prima, avrà certamente notato lo schermo pieno di pixel accesi un po' dappertutto. Non sono buttati lì a caso, ma quello che vediamo è, sotto forma di bit, tutto il contenuto della vi-

deo ram, compreso il generatore dei caratteri, la pagina testo e la pagina colore.

Perché mi dice ciò? Risposta: per il semplice fatto che ripulendo la pagina video, ovvero resettando tutti i bit in modo da avere un campo assolutamente vergine per disegnare in altra risoluzione, perderemo anche il generatore dei caratteri che dovrà essere ricaricato se vogliamo tornare al modo testo 80 colonne.

In figura 1 è mostrato il listatino Basic per ripulire la video RAM: lo abbiamo riportato solo per completezza, da Basic l'operazione è così lenta che viene voglia di spegnere il computer e

mettersi a piangere. Ovviamente in queste pagine è presente anche la versione in linguaggio macchina (che, di contro, fa solo rattristire un po').

Per ricaricare in RAM il generatore dei caratteri, ripetiamo, se per disegnare abbiamo ripulito la memoria, è disponibile una apposita routine di sistema operativo locata all'indirizzo decimale 52748 del banco 15. Ovvero, per tornare al modo testo, oltre alle due POKE sopra indicate, sarà necessario effettuare anche:

```
SYS 52748
```

Se ci troviamo in linguaggio macchina, effettueremo (oltre a ripristinare il registro 25):

```
JSR $CEOC
```

essendo \$CEOC esattamente 52748 in esadecimale. Semplice no?

La pagina grafica

Dicevamo che ad ogni pixel del video ad altissima risoluzione corrisponde un bit della video RAM. Nella fattispecie, se tale bit è a 1 vedremo un pixel acceso nella posizione corrispondente nello schermo, se tale è a 0 il corrispondente pixel sarà spento. Corrispondente come da figura 2: abbiamo che la prima linea di pixel è immagine dei primi 80 byte la seconda linea dei secondi e così via. Nell'ambito di ogni byte bisognerà settare il bit opportuno, tenuto conto che il bit 7 di un byte è quello alla sua estrema sinistra e il bit 0 è quello alla sua destra, come mostrato in figura 3.

Detto questo, proviamo ad accendere il pixel di coordinate (200,100). Come di consueto l'origine (0,0) è posta in alto a sinistra quindi il range di valori per la X è compreso tra 0 e 639, quello della Y tra 0 e 199.

Per il momento dimentichiamo le complicazioni indotte dall'8563 e la sua memoria indirettamente accessibile e riduciamo il problema al calcolo del byte da modificare (all'interno dei

Figura 4

```

01300 A2 19 LDX #19
01302 A9 67 LDA #87
01304 4C CC CD JMP $CDCC

```

Figura 5

```

01307 A2 19 LDX #19
01309 A9 67 LDA #87
0130B 20 CC CD JSR $CDCC
0130E 20 CC CE JSR $CEOC
01311 A9 93 LDA #93
01313 4C D2 FF JMP $FFD2

```

Figura 6

```

01316 A2 12 LDX #12
01318 A9 00 LDA #00
0131A 20 CC CD JSR $CDCC
0131D A2 13 LDX #13
0131F 20 CC CD JSR $CDCC
01322 AA TAX
01323 A0 40 LDY #40
01325 A9 1F LDA #1F
01327 8D 00 D6 STA $D600
0132A 2C 00 D6 BIT $D600
0132D 10 FB BPL $132A
0132F A9 00 LDA #00
01331 8D 01 D6 STA $D601
01334 CA DEX
01335 D0 EE BNE $1325
01337 88 DEY
01338 D0 EB BNE $1325
0133A 60 RTS

```

Figura 7

```

01340 8D 00 0B STA $0B00
01343 8E 01 0B STX $0B01
01346 8C 02 0B STY $0B02
01349 AD 00 0B LDA $0B00
0134C 38 SEC
0134D E9 80 SBC #80
0134F AD 01 0B LDA $0B01
01352 E9 02 SBC #02
01354 B0 E4 BCS $133A
01356 38 SEC
01357 AD 02 0B LDA $0B02
0135A E9 C8 SBC #C8
0135C B0 DC BCS $133A
0135E 38 SEC
0135F AD 00 0B LDA $0B00
01362 E9 00 SBC #00
01364 AD 01 0B LDA $0B01
01367 E9 00 SBC #00
01369 90 CF BCC $133A
0136B AD 02 0B LDA $0B02
0136E E9 00 SBC #00
01370 90 C8 BCC $133A
01372 AD 02 0B LDA $0B02
01375 8D 03 0B STA $0B03
01378 A2 00 LDX #00
0137A 8E 04 0B STX $0B04
0137D 0E 03 0B ASL $0B03
01380 2E 04 0B ROL $0B04
01383 0E 03 0B ASL $0B03
01386 2E 04 0B ROL $0B04
01389 0E 03 0B ASL $0B03
0138C 2E 04 0B ROL $0B04
0138F 0E 03 0B ASL $0B03
01392 2E 04 0B ROL $0B04
01395 AD 03 0B LDA $0B03
01398 8D 05 0B STA $0B05
0139B AD 04 0B LDA $0B04
0139E 8D 06 0B STA $0B06
013A1 0E 05 0B ASL $0B05
013A4 2E 06 0B ROL $0B06
013A7 0E 05 0B ASL $0B05
013AA 2E 06 0B ROL $0B06
013AD 18 CLC
013AE AD 03 0B LDA $0B03
013B1 6D 05 0B ADC $0B05
013B4 8D 05 0B STA $0B05
013B7 AD 04 0B LDA $0B04
013BA 6D 06 0B ADC $0B06
013BD 8D 06 0B STA $0B06
013C0 AD 00 0B LDA $0B00
013C3 8D 03 0B STA $0B03
013C6 AD 01 0B LDA $0B01
013C9 8D 04 0B STA $0B04
013CC 4E 04 0B LSR $0B04
013CF 6E 03 0B ROR $0B03
013D2 4E 04 0B LSR $0B04
013D5 6E 03 0B ROR $0B03
013D8 4E 04 0B LSR $0B04
013DB 6E 03 0B ROR $0B03
013DE 18 CLC
013DF AD 05 0B LDA $0B05
013E2 6D 03 0B ADC $0B03
013E5 8D 05 0B STA $0B05
013E8 AD 06 0B LDA $0B06
013EB 6D 04 0B ADC $0B04
013EE 8D 06 0B STA $0B06
013F1 AD 00 0B LDA $0B00
013F4 29 07 AND #07
013F6 AA TAX
013F7 E8 INX
013F8 38 SEC
013F9 A9 00 LDA #00
013DF AD 05 0B LDA $0B05
013E2 6D 03 0B ADC $0B03
013E5 8D 05 0B STA $0B05
013E8 AD 06 0B LDA $0B06
013EB 6D 04 0B ADC $0B04
013EE 8D 06 0B STA $0B06
013F1 AD 00 0B LDA $0B00
013F4 29 07 AND #07
013F6 AA TAX
013F7 E8 INX
013F8 38 SEC
013F9 A9 00 LDA #00
013FB 6A ROR
013FC CA DEX
013FD D0 FC BNE $13FB
013FF 8D 07 0B STA $0B07
01402 A2 12 LDX #12
01404 AD 06 0B LDA $0B06
01407 20 CC CD JSR $CDCC
0140A A2 13 LDX #13
0140C AD 05 0B LDA $0B05
0140F 20 CC CD JSR $CDCC
01412 20 D8 CD JSR $CDDB
01415 0D 07 0B ORA $0B07
01418 8D 07 0B STA $0B07
0141B A2 12 LDX #12
0141D AD 06 0B LDA $0B06
01420 20 CC CD JSR $CDCC
01423 A2 13 LDX #13
01425 AD 05 0B LDA $0B05
01428 20 CC CD JSR $CDCC
0142B AD 07 0B LDA $0B07
0142E 20 CA CD JSR $CDCA
01431 60 RTS

```


128 da zero

16000 che ci necessitano) e in questo bit da settare, in funzione del pixel cui siamo interessati (nel nostro esempio (200, 100)). Dato che il nostro pixel giace sulla 101-esima linea certamente le prime 100 file di byte non saranno interessate all'accensione di tale pixel. Nell'ambito della linea 101, dato che ogni 8 pixel un byte, il byte interessato sarà dato dal dividere l'ascissa per 8: 200 diviso 8 fa 25. In tale byte (dato da $100 \cdot 80 + 25$) setteremo il bit 7-RESTO ($X/8$), ovvero se $X/8$ dà come resto 0 (il nostro caso), setteremo il bit 7, se $X/8$ dà come resto 1, il bit 6 e così via.

Le routine in L.M.

In figura 4 è mostrata la «routine» in linguaggio macchina per passare al modo grafica. Essa sfrutta una routine del sistema operativo, posta a \$ CDCC (prendete nota) del banco 15 che permette di mandare il valore posto nel registro A, al registro del VDC indicato nel registro X. Nel caso nostro metteremo in X il valore \$ 19 (25 in esadecimale, il registro che permette di accedere al modo grafico) e nel registro A il valore \$ 87, come già detto. Un salto a \$ CDCC e la pagina grafica è servita. Da tastiera o da Basic, per accedere alla routine di figura 4, che se siamo interessati la digiteremo

Nota

I codici di controllo nei listati sono riportati in forma «esplicita», in conseguenza dell'impiego della stampante Star NL-10 e relativa interfaccia per Commodore. Ovviamente, nella digitazione del programma è necessario usare i consueti tasti che corrispondono alle indicazioni fra parentesi: ad esempio cursore destro per (RGHT), CTRL-3 per (RED) eccetera.

(CLR)	=	⏏	(YEL)	=	⏏
(HOME)	=	⏏	(RVS)	=	⏏
(DOWN)	=	⏏	(OFF)	=	⏏
(UP)	=	⏏	(ORNG)	=	⏏
(RGHT)	=	⏏	(BRN)	=	⏏
(LEFT)	=	⏏	(LRED)	=	⏏
(BLK)	=	⏏	(GRY1)	=	⏏
(WHT)	=	⏏	(GRY2)	=	⏏
(RED)	=	⏏	(LGRN)	=	⏏
(CYN)	=	⏏	(LBLU)	=	⏏
(PUR)	=	⏏	(GRY3)	=	⏏
(GRN)	=	⏏	(SWLC)	=	⏏
(BLU)	=	⏏			

```

100 GOSUB1660:FAST:GOTO160
110 REM *****
120 REM * SUBROUTINE PLOT *
130 REM *****
140 SYS4928, X AND 255, X/256, Y
150 RETURN
160 PRINT"(CLR)":WINDOW 15,1,79,24
170 PRINT"(LGRN) (DOWN)
180 PRINT"----- (RVS) (GRY1) ** MATH PACK 128 ** (LGRN) (OFF)
190 PRINT"
200 PRINT"
210 PRINT" (YEL) 1. (LRED) INPUT NUOVA FUNZIONE (LGRN)
220 PRINT"
230 PRINT" (YEL) 2. (LRED) INPUT NUOVO INTERVALLO (LGRN)
240 PRINT"
250 PRINT" (YEL) 3. (LRED) OUTPUT GRAFICO Y=F(X) (LGRN)
260 PRINT"
270 PRINT" (YEL) 4. (LRED) INTEGRALE DEFINITO (LGRN)
280 PRINT"
290 PRINT" (YEL) 5. (LRED) DERIVATA PRIMA E SECONDA (LGRN)
300 PRINT"
310 PRINT" (YEL) 6. (LRED) RICERCA MASSIMI E MINIMI (LGRN)
320 PRINT"
330 PRINT" (YEL) 7. (LRED) RICERCA SOLUZIONI REALI (LGRN)
340 PRINT"
350 PRINT" (YEL) 8. (LRED) QUIT (LGRN)
360 PRINT"
370 PRINT"
380 PRINT" (GRY1) (RVS) (C) 1987 ADP-SOFTWARE (LGRN) (OFF)
390 PRINT" (HOME) (HOME) "
400 GET$:IFZ$<"1"ORZ$>"8"THEN400
410 ONVAL(Z$)GOTO460,650,720,1230,1130,1490,1320,420
420 PRINT"(HOME) (HOME) (CLR)":END
430 REM *****
440 REM * NUOVA FUNZIONE *
450 REM *****
460 WINDOW19,6,54,20
470 PRINT"(CLR) (DOWN) ($15) (LRED) (RVS) NUOVA FUNZIONE (OFF) (DOWN) "
480 INPUT"(DOWN) (CYN) F(X) =":Z$:PRINT"(DOWN) (BLK) 530DEFFNY(X) =":Z$:PRINT"RUN 220 (HOME) (DOWN) (DOWN) (DOWN) "
490 POKE208,2:POKE842,13:POKE843,13:END
500 REM *****
510 REM *****
520 REM *****
530 REM *****
540 REM *****
550 REM *****
560 REM *****
570 REM *****
580 REM *****
590 REM *****
600 REM *****
610 REM *****
620 REM *****
630 REM *****
640 REM *****
650 WINDOW19,6,54,20
660 PRINT"(CLR) (DOWN) ($15) (LRED) (RVS) NUOVO INTERVALLO (OFF) (DOWN) "
670 GOSUB1070:X1=T1:X2=T2:GOTO160
680 DEFFNY(X)=SIN(X)+COS(2*X)*SIN(X/3)+SIN(COS(X))/2
690 DEFFNY1(X)=(FNY(X+1E-4)-FNY(X-1E-4))/2E-4
700 DEFFNY2(X)=(FNY1(X+1E-4)-FNY1(X-1E-4))/2E-4
710 DEFFNAR(Y)=INT(Y*1E4+.5)/1E4:RETURN
720 WINDOW19,6,54,20
730 PRINT"(CLR) (DOWN) ($15) (LRED) (RVS) OUTPUT GRAFICO (OFF) (DOWN) "
740 IFX1<X2THENGOSUB1070:X1=T1:X2=T2
750 Z$="SI":FL=0:GOSUB680:INPUT"(CYN) ASSI CARTESIANI ":Z$:IFZ$="SI"THENFL=1
760 Y1=0:Y2=0
770 Z$="SI":INPUT"AUTO SCALATURA ":Z$:IFZ$="SI"THEN850
780 INPUT"YMIN YMAX":Y1,Y2
790 IFY1<Y2THENT=Y1:Y1=Y2:Y2=T
800 Y1=-Y1:Y2=-Y2
810 SYS4864:SYS4886:GOTO920
820 REM *****
830 REM * GRAFICO DI F(X) *
840 REM *****
850 PRINT"(CLR)":Y1=1E30:Y2=-1E30
860 SYS4864:SYS4886
870 FORX=X1TOX2STEP(X2-X1)/90
880 Y=-FNY(X)
890 IFY<Y1THENT=Y1:Y=Y1
900 IFY>Y2THENT=Y2:Y=Y2
910 NEXT
920 KY=195/(Y2-Y1)
930 KX=639/(X2-X1)
940 IFFL=0THEN1010
950 IFNOT(Y1<0ANDY2>0)THEN 980
960 Y=-Y1*KY:FORX=0TO639
970 GOSUB140:NEXT
980 IFNOT(X1<0ANDX2>0)THEN 1010
990 X=-X1*KX:FORX=0TO199
1000 GOSUB140:NEXT
1010 FORX=X1TOX2STEP(X2-X1)/640
1020 X=(X0-X1)*KX:Y=(-FNY(X0)-Y1)*KY+3
1030 GOSUB140
1040 NEXT
1050 GETZ$:IFZ$="1"THEN1050
1060 SYS4871:PRINT"(HOME) (HOME) (CLR)":GOTO160
1070 T1=0:T2=0:INPUT"(CYN) INTERVALLO":T1,T2:IFT1>T2THENT=T1:T1=T2:T2=T
1080 IFT1<T2THENT1=T1:T1=T2:T2=T
1090 RETURN
1100 REM *****
1110 REM * DERIVATA *
1120 REM *****
1130 WINDOW19,6,54,20

```



```

1140 PRINT"(CLR) (DOWN) ($15) (LRED) (RVS) DERIVATA (OFF) (DOWN)";GOSUB680
1150 Z$="":INPUT"(HOME) (DOWN) (DOWN) (DOWN) (CYN) X=";Z$:IFZ$="" THEN160
1160 XX=VAL(Z$):Y=FNXY(X):Y=FNAR(Y):PRINT"(DOWN) F (X) =";Y;"(LEFT)
1170 Y=FNXY1(X):Y=FNAR(Y):PRINT"(DOWN) F' (X) =";Y;"(LEFT)
1180 Y=FNXY2(X):Y=FNAR(Y):PRINT"(DOWN) F'' (X) =";Y;"(LEFT)
1190 GOTO1150
1200 REM *****
1210 REM * INTEGRAZIONE *
1220 REM *****
1230 WINDOW19,6,54,20
1240 PRINT"(CLR) (DOWN) ($15) (LRED) (RVS) INTEGRALE DEFINITO (OFF) (DOWN)";GOSUB1070
:INPUT"(DOWN) (CYN) N. SUDDIVISIONI";C:GOSUB680
1250 J1=FNXY(T1):S=(T2-T1)/C:A=0:FORXX=T1+STOT2STEPS:J2=FNXY(X):A=A+S*(J2+J1)/2
1260 J1=J2:NEXT A=FNAR(A):PRINT"(DOWN) (DOWN) (LEFT) (DOWN) (LEFT) (UP) F(X) DX=";A
1270 GETZ$:IFZ$="" THEN1270
1280 GOTO160
1290 REM *****
1300 REM * SOLUZIONI REALI *
1310 REM *****
1320 WINDOW19,6,54,20
1330 PRINT"(CLR) (DOWN) ($15) (LRED) (RVS) SOLUZIONI REALI (OFF) (DOWN)";GOSUB1070:GO
SUB680:FR=0
1340 PRINT"(DOWN) (RVS) (YEL) ASCISSA (OFF) (CYN) (DOWN) "
1350 R=(T2-T1)/100
1360 J1=SGN(FNXY1(T1)):S=R
1370 T1=T1+S:J3=FNXY(T1):J2=-SGN(J3)*(ABS(J3)>1E-8):IFJ1<>J2THEN1400
1380 IFT1<T2THEN1370
1390 PRINT"(RVS) (DOWN) (YEL) STOP ! (OFF) (CYN) ";GOTO1270
1400 IFFR=OTHENSS=S:FR=1
1410 IFJ1=OTHENZ-T1=S:GOTO1440
1420 IFJ2=OTHENZ-T1=GOTO1440
1430 T1=T1-S:S=S/2:J1=SGN(FNXY1(T1)):GOTO1370
1440 Z=FNAR(Z):PRINTZ:FR=0:T1=T1+R:GOTO1360
1450 FORU=ITOLEN(A$):POKE50683+U,ASC(MID$(A$,U,1)):NEXT:POKE50683+U,35:GOTO980
1460 REM *****
1470 REM * MAX & MIN *
1480 REM *****
1490 WINDOW19,6,54,20
1500 PRINT"(CLR) (DOWN) ($15) (LRED) (RVS) MAX & MIN (OFF) (DOWN)";GOSUB1070:GOSUB680
:FR=0
1510 PRINT"(DOWN) (RVS) (YEL) ASCISSA *** ORDINATA (CYN) (OFF) (DOWN) "
1520 R=(T2-T1)/50
1530 J1=SGN(FNXY1(T1)):S=R
1540 T1=T1+S:J3=FNXY1(T1):J2=-SGN(J3)*(ABS(J3)>1E-20):IFJ1<>J2OR$1E-15THEN1570
1550 IFT1<T2THEN1540
1560 GOTO1390
1570 IFFR=OTHENSS=S:FR=1
1580 IFJ1=OTHENZ-T1=S:GOTO1610
1590 IFJ2=OTHENZ-T1=GOTO1610
1600 T1=T1-S:S=S/2:J1=SGN(FNXY1(T1)):GOTO1540
1610 Z=FNAR(Z):T=FNAR(FNXY(Z)):Z$=LEFT$( " ",4-LEN(STR$(INT(T))))+STR$(T)
1620 J3=FNAR(FNXY2(Z))
1630 IFSGN(J3)=1THENPRINTZ$,"(LEFT) (LEFT) MIN";Z$
1640 IFSGN(J3)=-1THENPRINTZ$,"(LEFT) (LEFT) MAX";Z$
1650 FR=0:T1=T1+R:GOTO1530
1660 PRINT"(CLR) (DOWN) (DOWN) (DOWN) (DOWN) "
1670 PRINT"(PUR)
1680 PRINT"(PUR)
1690 PRINT"(PUR)
1700 PRINT"(PUR)
1710 PRINT"(PUR)
1720 PRINT"(PUR)
1730 PRINT"(PUR)
1740 PRINT"(PUR)
1750 PRINT"(LBLU)
1760 PRINT"(LBLU)
1770 PRINT"(LBLU)
1780 PRINT"(LBLU)
1790 PRINT"(LBLU)
1800 PRINT"(LBLU)
1810 PRINT"(LBLU)
1820 PRINT"(CYN) (DOWN) (DOWN) (RVS) (C) 1987 ADP-SOFTWARE
(OFF) "
1830 FORK=4864TO5169:READKK:POKEK,KK:NEXT
1840 RETURN
1850 DATA162,25,169,135,76,204,205,162,25,169,71,32,204,205,32,12,206,169,147
1860 DATA76,210,255,162,18,169,0,32,204,205,162,19,32,204,205,170,160,64,169
1870 DATA31,141,0,214,44,0,214,16,251,169,0,141,1,214,202,208,238,136,208,235
1880 DATA96,234,234,234,234,234,141,0,11,142,1,11,140,2,11,173,0,11,56,233,128
1890 DATA173,1,11,233,0,144,207,173,2,11,233,0,144,200,173,2,11,141,3,11,162
1900 DATA10,142,4,11,14,3,11,46,4,11,14,3,11,46,4,11,14,3,11,46,4,11,14,3,11,46
1910 DATA4,11,173,3,11,141,5,11,173,4,11,141,6,11,14,5,11,46,6,11,14,5,11,46
1920 DATA6,11,24,173,3,11,109,5,11,141,5,11,173,4,11,109,6,11,141,6,11,173,0
1930 DATA11,141,3,11,173,1,11,141,4,11,78,4,11,110,3,11,78,4,11,110,3,11,78,4
1940 DATA11,110,3,11,24,173,5,11,109,3,11,141,5,11,173,6,11,109,4,11,141,6,11
1950 DATA173,0,11,41,7,170,232,56,169,0,106,202,208,252,141,7,11,162,18,173,6
1960 DATA11,32,204,205,162,19,173,5,11,32,204,205,32,216,205,13,7,11,141,7,11
1970 DATA162,18,173,6,11,32,204,205,162,19,173,5,11,32,204,205,173,7,11,32,202
1980 DATA205,96
1990 DATA205,96

```

tramite il monitor interno al 128, basterà il comando SYS 4864.

In figura 5 è mostrata la routine che permette di ritornare al modo testo, ovviamente perdendo il contenuto della pagina grafica. Si manda al registro \$ 19 il valore \$ 47 e con la subroutine del S.O. locata all'indirizzo \$ CEC si immettono i caratteri nel generatore dei caratteri nella video RAM. Prima di finire, viene ripulito il video mandando un codice ascii \$ 93 (per gli amici «cuoricino») alla routine \$ FFD2 che lo «printa» sullo schermo. Da Basic digiteremo, per tornare al modo caratteri 80 colonne, SYS 4871.

La routine HCLEAR di figura 6 pulisce il video grafico riempiendo di 0 tutta la RAM dell'integrato video. Per fare ciò si avvale dei due registri \$ 12 e \$ 13 per puntare il byte interessato e del registro \$ 1F per spedire lo zero purificatore. Ricordiamo che i puntatori \$ 12 e \$ 13 si autoincrementano man mano che mandiamo dati tramite \$ 1F: per questo motivo è sufficiente solo inizializzarli al primo byte e non curarci più di loro. Ciò che faremo è solo di mandare 16k zeri l'uno dopo l'altro.

Infine, in figura 7 è mostrata la routine che plotta il punto dato. Essendo molto lunga, a causa dei calcoli che bisogna compiere per trovare byte e bit cercato, non la commenteremo nei minimi dettagli. Tra le cose che desideriamo dirvi, il fatto che la routine controlla se il punto passato è o no all'interno del range plottabile, in modo da scartare richieste fuori campo. Il controllo per pixel di coordinata negativa, pur non essendo necessaria, è stato messo in previsione di future espansioni della routine, come la possibilità di CLIP-ing di aree di video. Ritorniamo in seguito sull'argomento.

Per l'interfacciamento da Basic o da tastiera, è necessario passare alla routine nel registro A la parte bassa dell'ascissa, nel registro X la parte alta (dato che l'ascissa può assumere valori fino a 639, maggiori quindi di 255) e nel registro Y l'ordinata del punto. In pratica, per plottare il punto di coordinate (X1, Y1) scriveremo:

```
SYS 4228,X1 AND 255,X1/256,Y1
```

essendo X1 AND 255 la parte bassa e X1/256 la parte alta di X1.

Math Pack

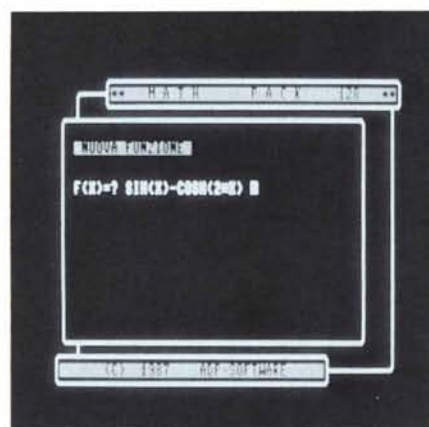
Il programma listato in queste pagine, è il glorioso Math Pack che i lettori più anziani, soprattutto se precedentemente vichinghi o post-vichinghi, dovrebbero già conoscere. Ne è stata in-



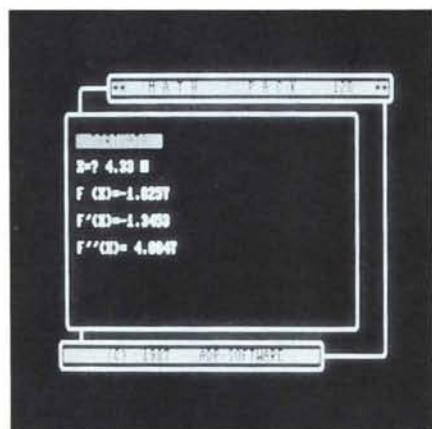
Menu del programma Math Pack.



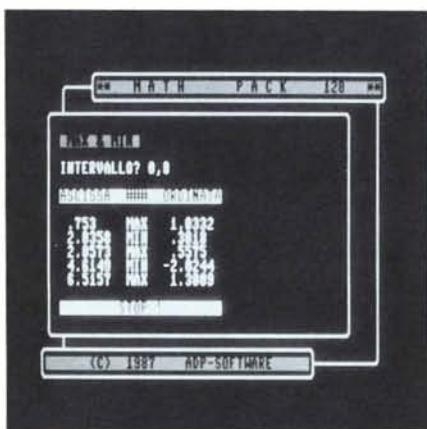
Output grafico.



Inserimento funzione.



Derivata prima e seconda.



Ricerca massimi e minimi.

fatti già presentata versione per il glorioso VIC-20, successivamente modificata per la grafica del 64 e ora per quella del 128.

Il programma è sostanzialmente lo stesso: solo poche modifiche sono state fatte, alcune come da richiesta di lettori, altre indotte dalla particolare architettura dello strano Commodore 128. Ad esempio, non è più possibile tornare al grafico precedentemente tracciato, dato che quando visualizziamo testi ad 80 colonne, necessariamente dobbiamo caricare nella video RAM il generatore di caratteri (calpestando pezzi di disegno).

La miglioria apportata consiste nella possibilità di sganciare l'autoscaling e selezionare così, manualmente, la porzione di piano cartesiano che desideriamo vedere. Nella precedente versione, infatti, se disegnavamo una funzione, questa era plottata in modo che il suo massimo assoluto (nell'intervallo) toccava il limite superiore dello schermo e il suo minimo quello inferiore. Se da un lato tale sistema permetteva un'agevole visualizzazione di funzioni di cui non conoscevamo l'an-

damento e quindi in quale parte di piano capitava, nel caso di funzioni asintotiche il tutto si risolveva in una ingiusta compressione di tutto il grafico, tanto da comprometterne la leggibilità.

Dando il RUN al programma, dopo pochi secondi appare il menu composto da 8 opzioni. L'ultima, di Quit, come è facile supporre serve per uscire dal programma.

Per inserire una funzione si seleziona l'opzione 1 e si digita una qualunque espressione in cui sia presente la variabile X. Si noti che la sintassi è quella normale del Basic quindi sono necessarie tutte le parentesi che il Basic richiede. Non scriveremo $\sin x$ ma $\text{SIN}(X)$, asterisco per moltiplicare, freccetta in su per l'elevazione a potenza.

L'opzione 2 permette di inserire l'intervallo di ascissa di cui desideriamo il tracciamento del grafico. Tale opzione si usa quando vogliamo inserire un secondo intervallo di tracciamento: il primo è richiesto anche con l'opzione 3 che serve appunto per far tracciare il grafico. In questo è chiesto se si desi-

derano o meno gli assi cartesiani e se vogliamo l'auto-scaling. Tutte le risposte per default sono pari a SÌ e per negarne una si indica chiaramente NO.

Le ultime 4 opzioni permettono rispettivamente di calcolare l'integrale definito su un intervallo, la derivata prima e seconda nonché il valore stesso della funzione in un determinato punto, la ricerca dei massimi e dei minimi e delle soluzioni reali.

Quando richiesto si dovrà inserire un intervallo e nel caso dell'integrale definito anche il numero delle suddivisioni. Valori alti (alcune centinaia) permettono un calcolo più accurato con tempi di attesa lunghi, mentre con valori bassi abbiamo esattamente il contrario: tempi di risposta veloci ma bassa precisione del risultato. Generalmente 100 dovrebbe rappresentare un buon compromesso velocità-precisione. Buon divertimento.

Inviare i vostri contributi

Chiunque voglia contribuire alla rubrica «128 da zero» proponendo argomentazioni, trucchetti o semplicemente notizie riguardanti il Commodore 128 non deve far altro che inviare il proprio contributo in redazione, possibilmente indicando sulla busta «128 da zero».

I lavori più interessanti saranno ospitati su queste pagine ovviamente ricompensando gli autori a seconda sia delle dimensioni che dell'importanza degli elaborati stessi. Non dimenticate (a tal proposito) di indicare sempre chiaro il vostro nome, indirizzo, codice fiscale. Anche il vostro numero di telefono potrebbe esserci utile per potervi eventualmente contattare prima della pubblicazione per chiarimenti. Buon lavoro.