



8563: ultimo atto

■ L'articolo di questo mese di 128 da zero si divide in tre parti. Nella prima parte concluderemo il nostro discorso sull'integrato video 8563 mostrandovi come tale processore video manipola i byte della sua video ram. La seconda parte ospiterà un tool grafico inviatoci da un lettore che, a dispetto delle routine presentate nei due numeri scorsi, permette di compiere molte operazioni prima non disponibili, infine con altri contributi pervenuti proprio grazie al nostro annuncio di collaborazione, inizieremo un discorso nuovo sul sistema operativo del 128. ■

24, 30, 32 e 33

L'ultima scoperta circa l'8563 riguarda la possibilità che ha questo di spostare blocchi di byte all'interno della sua memoria privata. Ed era proprio auspicabile che ciò fosse stato possibile, non per altro, per permettere una qualsiasi applicazione a 80 colonne (specie word processing) senza far innervosire troppo l'utente davanti a lentissimi scroll a causa dell'indirizzamento bi-indiretto di questi benedetti 16 K. Ovvero noi diciamo all'integrato cosa spostare e «lui» tra le righe ci accontenta velocissimamente. Specialmente considerato il tempo che impiegheremo noi a farlo dall'esterno.

Per definire quale blocco di byte della video ram desideriamo trasferire in un'altra qualsiasi posizione sempre nella video ram, utilizzeremo i registri indicati a capo di questo paragrafo. Per comunicare al processore video la destinazione di questo blocco, sarà sufficiente usare i registri 18 e 19, già utilizzati nelle puntate precedenti essendo questi il puntatore ufficiale al byte interessato.

Indicheremo nei registri 32 e 33 rispettivamente la parte alta e la parte bassa dell'indirizzo del byte iniziale

del nostro blocco, nel registro 30 il numero di byte da trasferire, dopo aver opportunamente settato il bit 7 del registro 24 detto appunto bit copy. Si noti che la massima quantità di byte che riusciremo a far trasferire in un colpo solo è limitata a 256. Ciò si ottiene indicando nel registro 30 il valore 0, successore di 255 nell'aritmetica a 8 bit. Ovviamente per tutte le altre quantità comprese tra 1 e 255 basterà inserire tale valore per essere accontentati.

Facciamo un esempio: immaginiamo di dover trasferire 200 (decimale) byte a partire dall'indirizzo \$100 all'indirizzo \$400. Abbiamo scelto questi valori in modo da osservare sul video ciò che accade. Scriveremo un piccolo programma Basic per settare i vari registri. Per prima cosa definiamo la destinazione, mettendo \$400, suddiviso in parte alta e parte bassa nei registri 18 e 19:

```
10 POKE DEC("D600"),18:POKE
DEC("D601"),4
20 POKE DEC ("D600"),19:POKE
DEC("D601"),0
```

Attenzione, ogni coppia di poke su un'unica linea.

Successivamente definiamo il primo byte del blocco da trasferire indicando \$100 nei registri 32 e 33:

```
30 POKE DEC("D600"),32:POKE
DEC("D601"),1
40 POKE DEC ("D600"),33:POKE
DEC("D601"),0
```

Segue il settaggio del bit copy:

```
50 POKE DEC("D600"),24: A=PEEK
(DEC("D601"))
60 POKE DEC ("D600"),24:POKE
DEC("D601"),A OR 128
```

Infine indichiamo nel registro 30 il numero di byte da copiare:

```
70 POKE DEC ("D600"),30:POKE
DEC("D601"),200
```

Detto fatto: basta solo dare RUN. Ovviamente riempiate prima il video di caratteri (non tutti uguali) in modo da notarne lo spostamento.

Ultime curiosità

Oltre alla possibilità di spostare blocchi di video ram, il chip 8563 permette di gestire la penna ottica, di visualizzare più di 25 righe di testo, ca-

ratteri più larghi o più stretti di 8 pixel ed altre "cosucce" simili.

Ad esempio pare che il bit 5 di SD600 indichi se il pannello elettronico del video sta componendo la schermata o sia in fase di ritorno per un nuovo quadro. In questo modo si potrebbero realizzare routine sincronizzate col quadro, a mo' di come si faceva per il 64, per costruire utility interessanti. Ancora aperto resta, per citarne uno, il problema dello scrolling fine sulle 80 colonne. Dal momento che è mia profonda convinzione che l'utente medio del Commodore 128, e la qualità del software dei lettori che arriva in redazione lo testimonia abbondantemente, è smantettomane con tanto di pedigree, se qualcuno fosse interessato ad approfondire questi studi e perviene a risultati interessanti, non manchi di comunicarci. Non tanto per noi, inutile dirlo, ma per tutti gli altri 128-isti sicuramente assetati di notizie di questo genere.

Master Graphic 128

di Luigi Magni - Perugia

Il tool grafico inviatoci dal lettore di Perugia, come detto nell'apertura, oltre a implementare la grafica 640x200 con la possibilità di disegnare punti e linee, permette alcune operazioni in più come l'HardCopy su stampante 803 (il tool del sottoscritto, molto vigliaccamente, stampava solo su STAR N1-10), la possibilità di salvare e caricare immagini da disco, nonché permette di passare indistintamente a pagina grafica a testo e viceversa, salvando momentaneamente la pagina grafica nel banco 1 della normale ram del 128. Esso è listato in esadecimale in questa pagina e può essere facilmente ricopiato in memoria utilizzando il monitor di linguaggio macchina residente nel 128.

Anche questa volta i vari comandi sono impartiti tramite normali SYS e se bisogna passare dei parametri si utilizza il formato:

SYS indirizzo, A, X, Y

dove A, X, Y sono i valori da introdurre negli anonimi registri. Il tool grafico Master Graphic 128 mette a disposizione in tutto 9 comandi suddivisi, per comodità in 4 gruppi. Comincia-

Utility:

Il primo gruppo di comandi si compone di tre routine atte ad attivare la grafica ad alta risoluzione. Abbiamo la possibilità di passare al modo grafico mostrando una pagina *ripulita*, passare dalla grafica al modo testo e, molto interessante, passare dal testo ad un grafico precedentemente tracciato. Come già spiegato nei numeri scorsi, per fare questo è necessario trasferire

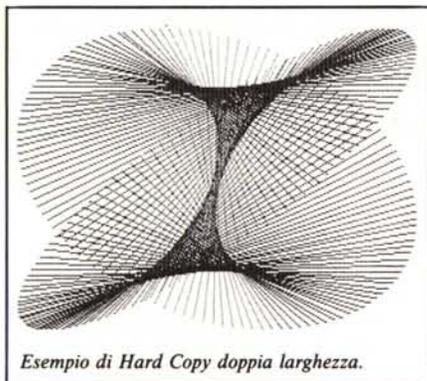
```

>01300 20 D2 14 A9 B0 BD FA 12 85 FD 85 FB A9 3E BD FB
>01310 12 85 FE 85 FC A9 00 BD FC 12 8D FD 12 8D FE 12
>01320 A9 04 AA A0 FF 20 BA FF A9 00 20 BD FF 20 C0 FF
>01330 A2 04 20 C9 FF A9 08 20 D2 FF A9 0D 20 D2 FF A9
>01340 C8 BD FF 12 38 A5 FD E9 50 85 FD B0 02 C6 FE A4
>01350 FD A5 FE 20 19 14 BD FD 12 38 A5 FB E9 50 85 FB
>01360 B0 02 C6 FC A4 FB A5 FC 20 19 14 AE FC 12 E0 00
>01370 F0 08 4A 6E FD 12 CA 4C 6E 13 AD FD 12 4A A2 08
>01380 E0 00 F0 08 0A 6E FE 12 CA 4C 80 13 6E FE 12 AD
>01390 FE 12 EA EA 09 80 20 D2 FF CE FF 12 AD FF 12 D0
>013A0 A3 A9 0D 20 D2 FF EE FC 12 AD FC 12 C9 08 D0 08
>013B0 A9 00 BD FC 12 4C CF 13 AD FB 12 85 FC 85 FE AD
>013C0 FA 12 85 FB EE FA 12 AD FA 12 85 FD 4C DD 13 AD
>013D0 FB 12 85 FE 85 FC AD FA 12 85 FD 85 FB AD FA 12
>013E0 C9 D0 F0 03 4C 3F 13 4C 26 14 A9 0D 20 D2 FF A9
>013F0 0F 20 D2 FF A9 04 20 C3 FF 20 CC FF A9 EA BD 92
>01400 13 BD 93 13 A9 D0 8D E1 13 A9 4C 8D E7 13 A9 26
>01410 8D E8 13 A9 14 8D E9 13 60 A2 12 20 CC CD 98 E8
>01420 20 CC CD 4C 8D CD A9 29 8D 92 13 A9 07 8D 93 13
>01430 A9 D1 8D E1 13 A9 EA 8D E7 13 BD E8 13 8D E9 13
>01440 4C 3F 13 20 D2 FF 20 D2 FF 60 A9 43 8D 97 13 A9
>01450 14 8D 98 13 20 00 13 A9 D2 8D 97 13 A9 FF 8D 98
>01460 13 60 20 64 15 F0 23 A9 E0 85 FE A9 00 85 FD 85
>01470 FB A9 D0 85 FC A9 FB 8D B9 02 04 15 20 11 15
>01480 A2 19 A9 87 20 CC CD 20 2C CD A9 3F 85 FC A0 00
>01490 B4 FD 20 04 15 A9 00 20 CA CD 88 D0 FA C6 FC D0
>014A0 F6 60 20 64 15 D0 FA A9 FF 85 FE A9 20 85 FD A9
>014B0 E0 85 FC A9 00 85 FB A9 FB 8D B9 02 04 15 20
>014C0 11 15 A2 19 A9 47 20 CC CD 20 27 15 20 0C CE 4C
>014D0 2C CD 20 64 15 F0 CA 20 27 15 A2 19 A9 87 20 CC
>014E0 CD A9 20 85 FD A9 FF 85 FE 20 04 15 A0 00 A2 7F
>014F0 20 A2 02 20 CA CD CD D0 F5 E6 FC A5 FC C5 FE CD
>01500 ED 4C 2C CD A2 12 A5 FD 20 CC CD E8 A9 00 4C CC
>01510 CD A0 00 20 DB CD A2 7F 20 AF 02 C8 D0 F5 E6 FC
>01520 A5 FC C5 FE D0 ED 60 A9 D0 85 FC A9 00 85 FD 85
>01530 FB 20 04 15 A9 FB 8D AA 02 8D B9 02 A0 00 A2 7F
>01540 20 A2 02 48 20 DB CD A2 7F 20 AF 02 20 04 15 68
>01550 20 CA CD EE 0D 15 CB D0 E5 E6 FD E6 FC A5 FC C9
>01560 E0 D0 DB 60 A2 19 20 DA CD C9 87 60 DB 85 62 86
>01570 61 84 63 A5 FF 85 65 A5 FE 85 64 A5 FA 85 66 A9
>01580 E6 8D 5B 16 8D 63 16 8D 92 16 8D 9E 16 8D C3 16
>01590 8D CB 16 A9 61 8D 02 16 A9 64 8D 04 16 A9 00 8D
>015A0 60 16 8D CB 16 85 6D 85 6C 8D A5 62 E5 65 80 25
>015B0 A9 C6 8D 5B 16 8D 63 16 8D C3 16 8D CB 16 A9 64
>015C0 8D 02 16 A9 61 8D 04 16 A9 FF 8D 60 16 8D CB 16
>015D0 38 A5 65 E5 62 85 68 D0 27 38 A5 61 E5 64 B0 18
>015E0 A9 C6 8D 5B 16 8D 63 16 8D C3 16 8D CB 16 A9 64
>015F0 8D 02 16 A9 61 8D 04 16 A9 FF 8D 60 16 8D CB 16
>01600 38 A5 65 E5 62 85 68 D0 27 38 A5 61 E5 64 B0 18
>01610 38 A5 61 E5 64 B0 02 C6 68 85 67 38 A5 63 E5 66
>01620 B0 0D A9 C6 8D 92 16 8D 9E 16 38 A5 66 E5 63 85
>01630 69 A5 68 D0 12 38 A5 67 E5 69 B0 08 A9 00 85 68
>01640 A5 69 85 6A 4C 45 16 A5 68 85 68 A5 67 85 6A A9
>01650 FF 85 68 E6 69 20 E5 16 A5 68 C9 FF D0 49 A5 65
>01660 C5 62 D0 07 A5 64 C5 61 D0 01 60 E6 64 A5 64 C9
>01670 00 D0 02 E6 65 18 A5 6C 65 69 85 6C 90 02 E6 6D
>01680 38 A5 6D E5 68 AA 90 55 F0 08 38 A5 6C E5 6A B0
>01690 0D CA 4C 8E 16 38 A5 6C E5 6A 90 41 F0 3F 85 6C
>016A0 86 6D C6 66 4C CD 16 A5 66 C5 63 D0 01 60 C6 66
>016B0 18 A5 6C 65 67 85 6C 90 02 E6 6D 38 A5 6C E5 6A
>016C0 0F C6 6D A6 6D E0 FF D0 07 A9 00 85 6D 4C CD
>016D0 16 85 6C E6 64 A5 64 C9 00 D0 02 E6 65 A5 66
>016E0 64 A4 66 20 DF 16 4C 46 16 A5 62 A6 61 A4 63 85
>016F0 FF 86 FE 84 FA A6 FA 86 FC A2 00 86 FB 86 FD 06
>01700 FA 26 FB 06 FA 26 FB 06 FA 26 FB 06 FA 26 FD 06
>01710 FC 26 FD 06 FC 26 FD 18 A5 65 FC 85 FA 06
>01720 65 FD 85 FB A5 FE 29 07 85 F9 46 FF 66 FE 46 FF
>01730 66 FE 46 FF 66 FE 18 A5 FA 65 FE 85 FA A5 FB 65
>01740 FF 85 FB 20 58 17 20 DB CD A6 F9 A4 0C FF F0
>01750 32 1D 7A 17 20 5B 17 20 CA CD 60 A2 12 BE 00 D6
>01760 2C 00 D6 10 FB A6 FB 8E 01 D6 A2 13 8E 00 D6 2C
>01770 00 D6 10 FB A6 FA 8E 01 D6 60 80 40 20 10 08 04
>01780 02 01 00 3D 89 17 4C 54 17 7F BF DF EF F7 FB FD
>01790 FE 20 1E 18 20 D2 14 A6 61 20 C9 FF A9 8B 85 FC
>017A0 A9 00 85 FD 85 FE 85 FF 20 04 15 20 DB CD 48 4A
>017B0 4A 4A 4A 20 32 18 68 29 0F 20 32 18 E6 FE D0 02
>017C0 E6 FF A5 FF C9 3F D0 E3 A5 FE C9 FF D0 D0 F0 45
>017D0 20 1E 18 20 62 14 A6 61 20 C6 FF 20 CF FF 85 FA
>017E0 20 CF FF 85 FB A9 08 85 FC A9 00 85 FD 85 FE 85
>017F0 FF 20 04 15 20 51 18 0A 0A 0A 48 20 51 18 68
>01800 05 FD 20 CA CD E6 FE D0 02 E6 FF A5 90 F0 E5 A5
>01810 FE C9 FF D0 DF 20 CC FF A5 61 20 C3 FF 60 48 98
>01820 85 61 20 BA FF 68 A2 AE A0 18 20 BD FF 20 C0 FF
>01830 EA 60 8B 89 9E 18 AA 89 7E 18 04 FA 0C 26 FA 8B
>01840 D0 09 48 A5 20 D2 FF 0E 03 A5 FE C9 FF D0 D0 8E 84 FC
>01850 60 A2 0F A5 FA 3D BE 18 DD 7E 18 F0 03 CA D0 F3
>01860 B6 FD 8D 9E 18 AA 4C 06 FB 26 FA 8B D0 07 20
>01870 CF FF 85 FB A0 08 CA D0 EF 84 FC A5 FD 69 00 A0
>01880 A8 B0 88 C0 C8 F0 D0 DB E0 FA E8 FB FC 80 80 FB
>01890 FB FB
>018A0 05 05 05 05 05 06 05 05 05 06 05 06 03 FF 00
READY.

```

Listato esadecimale del Tool Master Graphic 128.

la pagina grafica in una zona di memoria ram estranea ai 16 K dell'8563 dato che in questi si dovrà ricopiare il set di caratteri e saranno utilizzati 4 k per il video e la ram attribuiti. Per la precisione non vengono trasferiti tutti i 16 k ma solo 12, i primi 4 e gli ultimi 8, dato che i 4 k lasciati non vengono toccati quando si è in modo testo. Il rovescio della medaglia è che rubiamo 12 k al Basic, nella fattispecie allo spazio per le variabili. Restano comunque disponibili circa 40 k che, come è facile riconoscere, sono più che sufficienti nella maggioranza dei casi.



Esempio di Hard Copy doppia larghezza.

Le SYS da chiamare sono: per passare in pagina grafica cancellando:

SYS 5218

per tornare al modo testo:

SYS 5282

per tornare in HI-RES visualizzando la pagina grafica precedente:

SYS 5330

Plot & draw:

Come lo stesso lettore dichiara, riguardo al plot c'è poco da dire essendo praticamente la stessa routine pubblicata nel numero di aprile 86 di MCmicrocomputer. Le uniche modifiche riguardano la possibilità di cancellare un punto precedentemente tracciato. Per quanto riguarda il draw pare che il lettore abbia fatto i salti mortali per implementarlo non disponendo degli indirizzi delle routine per le divisioni presenti nel Basic del 128. Salti così mortali che, parole sempre dell'autore, il disassemblato risulta poco comprensibile per il fatto che il programma (orrore!) si automodifica nel corso dell'elaborazione a seconda di determinate condizioni. Leggi: un programma siffatto non potrà mai essere messo su rom, scusate se è molto. Passando agli indirizzi, per plottare un punto di coordinate (X1, Y1) scriveremo:

SYS 5855, X1/256, X1 AND 255, Y1

```

10 PRINT (CLR) " : IFPEEK(4864) <> 32 THEN PRINT "CARICARE PRIMA IL PROGRAMMA MATER 6
40-200" : END
20 CLR: BANK 15: FAST: TRAP 100: POKE 54784, 26: POKE 54785, 96
30 POKE 12, 0: Z=1: H=0: Z1=0: H1=1
40 PRINT (RVS) DEMO HI-RES ROUTINES 640*200 BY LUIGI MAGNI (FG)
(OFF) "
50 PRINT (DOWN) (DOWN) (DOWN) (RVS) 1 (OFF) CARICA SCHERMATA: PRINT (DOWN) (RVS) 2 (O
FF) SALVA SCHERMATA: PRINT (DOWN) (RVS) 3 (OFF) STAMPA SCHERMATA: PRINT (DOWN) (
RVS) 4 (OFF) VISUALIZZA HI-RES"
60 PRINT (DOWN) (RVS) X (OFF) FINE: CHAR, 18, 24, "QUALSIASI ALTRO TASTO PER IL DEMO
AUTOMATICO"
70 GETKEYA$: IFA#="1" THEN 210: ELSE IFA#="2" THEN 160: ELSE IFA#="3" THEN 120
75 IFA#="4" THEN 260: ELSE IFA#="X" THEN END: ELSE SYS 5218
80 FOR I=0 TO 2 * STEP .05: X=320-SIN(I+Z) * 319: Y=100-COS(I+H) * 99: Y1=100-SIN(I+Z1) * 99: X
1=320-COS(I+H1) * 319: GOSUB 110: NEXT
90 Z=RND(1) * 2 * PI: H=RND(2) * 2 * PI: Z1=RND(3) * 2 * PI: H1=RND(4) * 2 * PI: POKE 208, 0: GETKEYA$
100 SYS 5282: GOTO 70
110 A=X/256: B=X AND 255: A1=X1/256: B1=X1 AND 255: POKE 250, Y: POKE 254, B: POKE 255, A: SYS 548
4, A1, B1, Y1: RETURN
120 CHAR, 30, 16, ""
130 INPUT (RVS) N (OFF) ORMALE O (RVS) D (OFF) OPPIA " : ST: SYS 50341, 16: IF ST <> "N" AND S
T <> "D" THEN 70
140 IF ST="N" THEN SYS 54864: ELSE SYS 5194
150 GOTO 90
160 CHAR, 30, 16, ""
170 NMS="" : INPUT "NOME=" : NMS: SYS 50341, 16: IF NMS="" THEN 70
180 NMS="H/" + LEFT$(NMS, 14) + ", P, W" : L=LEN(NMS) : FOR I=1 TO L: POKE 6317+I, ASC(MID$(NMS, I
, 1)) : NEXT I
190 SYS 6033, L, B, 2: IF DS > 1 THEN SYS 5282: CHAR, 30, 18, DS, 1: DCLEAR: SLEEP 3: SYS 50341, 16:
GOTO 160
200 DCLEAR: GOTO 90
210 CHAR, 30, 16, ""
220 NMS="" : INPUT "NOME=" : NMS: SYS 50341, 16: IF NMS="" THEN 70
225 IF NMS="" THEN WINDOW 50, 2, 79, 23, 1: CATALOG "H/" : PRINT (DOWN) PREMI UN TASTO: GET
KEYA$: PRINT (CLR) (HOME) (HOME) : GOTO 210
230 NMS="H/" + LEFT$(NMS, 14) + ", L=LEN(NMS) : FOR I=1 TO L: POKE 6317+I, ASC(MID$(NMS, I, 1)) : NE
XT I
240 SYS 6096, L, B, 2: IF DS > 1 THEN SYS 5282: CHAR, 30, 18, DS, 1: DCLEAR: SLEEP 3: SYS 50341, 16:
GOTO 210
250 DCLEAR: GOTO 90
260 SYS 5330: GOTO 90

```

Listato Basic: demo del Master Graphic.

dove, come noto, X1/256 è la parte alta (gli 8 bit più significativi) di X1 e X1 AND 255 è la parte bassa.

Per tracciare una linea dal punto (X1, Y1) al punto (X2, Y2) scriveremo la sequenza

```

POKE 255, X1/256
POKE 254, X1 AND 255
POKE 250, Y1
SYS 5484, X2/256, X2 AND 255, Y2

```

Infine per cancellare punti invece che disegnarli basta introdurre il valore 255 nella cella di memoria 12 (POKE 12, 255) mentre qualsiasi altro valore introdotto nella stessa locazione ripristinerà il modo scrittura.

Stampa:

Per le due routine di stampa diremo solo che questa avviene in senso verticale ed è disponibile la stampa singola o doppia, nel primo caso stamperemo un punto per ogni pixel ma avremo un'immagine compressa, nel secondo caso le proporzioni saranno ristabilite ma per ogni pixel acceso stamperemo due punti affiancati sulla carta. Considerato inoltre che i pixel del 128 sono manifestamente oblungi (640 x 200 parla da solo) la seconda soluzione appare la migliore. Le SYS: per la stampa singola (immagine compressa):

SYS 4864

per la stampa doppia (immagine proporzionata):

SYS 5194

Hi-Res Compaer:

Il tool di routine grafiche Master Graphic 128 si chiude con due utility per salvare e caricare schermate in alta risoluzione. Esse utilizzano una non meglio specificata tecnica a nibble (semi-byte) con la quale se salviamo immagini con grosse aree vuote o piene risparmiamo spazio su disco oltre che guadagnare in velocità di trasferimento.

Per salvare o caricare una pagina grafica è necessario inserire (a colpi di POKE) nelle locazioni a partire dalla numero 6318 i codici ascii dei caratteri che compongono il nome del file suffissati dalla stringa «P,W» se si tratta di un salvataggio. Fatto ciò si esegue per il save:

SYS 6033, L, P, C

per il load:

SYS 6096, L, P, C

dove, in tutt'e due i casi, L è la lunghezza del nome (compreso l'eventuale suffisso) P è la periferica (8 per il disco 1 per il nastro), C il canale di I/O utilizzato (per il disco qualsiasi numero compreso tra 2 e 14).

Le routine del Kernal

di Adriano Asnaghi - Mestre (VE)

Il primo «vero» contributo a 128 da zero arrivato da un lettore riguarda alcune routine del Kernal del 128. Come è noto tali routine si usano nella programmazione in linguaggio macchina per sfruttare da quel livello tutte le risorse del computer quali video, stampante, disco, tastiera, porte ecc.

Suddivideremo il materiale pervenuto in alcune sezioni che pubblicheremo una alla volta, per qualche numero, come appendice alle varie puntate. Se intanto arriva in redazione dell'altro sempre in merito al Kernal, prolungheremo ad oltranza tali appuntamenti.

Prima di passare la parola, è necessaria una leggera tiratina d'orecchie all'autore che in alcuni punti è stato impreciso (mi sono permesso di aggiungere alcune cose) e proprio nella prima routine ha preso un piccolo abbaglio (dovevo intervenire). Questo agli altri lettori non interessa, l'Asnaghi potrà controllare col suo testo alla mano dove sono state effettuate le modifiche.

In bocca al lupo!

Le KERNAL di EDIT e gestione video

All'indirizzo \$C000 del banco 15 c'è la ROM per la gestione dell'EDIT. Alcune delle routine presenti possono essere proficuamente utilizzate in L.M. per svolgere funzioni utente. La descrizione di alcune di queste è riportata di seguito.

1) Definizione di una FUNCTION KEY

La routine che definisce una Function Key è locata a \$FF65 con un salto a \$C021. I parametri che devono essere passati alla routine sono:

Accumulatore - indica l'indirizzo in pagina zero di tre byte contigui contenenti rispettivamente l'indirizzo basso e alto dell'area di memoria dove è memorizzato il testo da inserire nella Key e il numero del banco.

Reg. X - indica il numero di Key da ridefinire (1-10).

Reg. Y - contiene la lunghezza del testo.

I valori di default presenti nella inizializzazione sono locati a \$CEA8.

2) Leggere o modificare la posizione corrente del CURSORE

La routine è locata all'indirizzo \$FFF0 con un salto a \$C018 e poi a \$CC6A. I parametri che devono essere passati alla routine sono:

— per leggere la posizione del cursore nella finestra in corso

Carry - posto a 1

essa ritorna in valori in

Reg. X - colonna

Reg. Y - riga

— per modificare la posizione del cursore nella finestra in corso

Carry - posto a 0

Reg. X - colonna

Reg. Y - riga

Questa routine può essere utilizzata anche in Basic con il seguente richiamo

```
SYS DEC("CC6C"),,riga,colonna
```

l'entry point è \$CC6C perché a \$CC6A c'è il test del carry ed il relativo salto se il carry è a 1.

3) Esegue l'output di un TESTO (utilizzabile in L.M.)

La routine è locata all'indirizzo \$FF7D con un salto a \$FA17. Questa routine permette di trasmettere, sul device corrente di output, un testo generico. Il testo deve essere specificato immediatamente dopo la chiamata della routine e deve terminare con un \$00. Il controllo del flusso di programma è dato all'istruzione seguente il marker di fine (lo \$00 appunto!). Nessun parametro deve essere passato alla routine.

La routine salva anche tutti i registri meno lo STATUS.

Il seguente è un esempio di utilizzo:

```
ENTRY JSR $C142 ; pulisce lo schermo
LDY #03 ; riga 4
LDX #0A ; colonna 11
CLC ; carry=0 per SET
JSR $FF65 ; sposta il cursore
JSR $FF7D ; visualizza il testo seguente
.DC 'PROVA KERNAL ',#00,#00
RTS ; rientra
```

4) Definizione di una WINDOW

La routine è locata all'indirizzo \$CA14 e \$CA16.

Il primo indirizzo di entrata definisce l'angolo alto-sinistro della Window ed il secondo definisce l'angolo basso-destro della stessa.

Le posizioni degli angoli dipendono dalle coordinate correnti del cursore. Ad esempio:

```
ENTRY LDY #03 ; riga 4
STY #EB ; colonna 11
LDX #0A ; angolo alto window
LDY #14 ; riga 21
STY #EB ; colonna 31
LDX #1D ; angolo basso window
STX #EC ; azzera window
JSR $CA16 ; azzera window
RTS ; rientra
```

Per ritornare al modo normale, cioè avere una window a pieno schermo basterà richiamare la routine locata a \$CA24. L'azzeramento della window può essere eseguito con la routine a \$C142 che posiziona anche il cursore nell'angolo alto a sinistra.

Lo spostamento del cursore nella window può essere eseguito mediante le seguenti routine:

Cursore Giù - \$C85A

Cursore Su - \$C867

Cursore a Sinistra - \$C875

Cursore a Destra - \$C884

L'esecuzione di una TAB avviene all'indirizzo \$C94F, mentre per togliere tutte le posizioni di tabulazione si deve richiamare la routine a \$C980.

È poi possibile ridefinire le nuove posizioni di tabulazione mediante la routine a \$C963, ponendo nel Reg. Y il valore della colonna dove mettere il marker di tabulazione.

5) Lo scrolling del video

Lo scrolling del contenuto di una finestra avviene tramite due routine, la prima che esegue lo SCROLL-UP e la seconda lo SCROLL-DOWN, locate rispettivamente agli indirizzi \$CABC e \$CACA.

6) Le routine che eseguono la sequenza <ESC> carattere

L'indirizzo di queste routine può essere rilevato all'indirizzo \$C9DE.

In questa zona di memoria c'è un vettore contenente gli indirizzi (il valore presente è l'indirizzo effettivo meno 1 perché le routine del kernal eseguono il richiamo via RTS) delle routine ordinate secondo il carattere abbinato con il codice di escape (@, A, B, ecc.). Infatti, per attivare il Beep occorre premere la sequenza <ESC>G, e quindi l'indirizzo della routine relativa si trova a

\$C9DE + (ASC("G") - ASC("@")) * 2,

cioè a \$C9EC che contiene l'indirizzo \$CB36. Questo indirizzo va aggiustato di 1 fornendo \$CB37.

Infine, un JSR a questo indirizzo attiverà il Beep da L.M.

Per gli altri codici il procedimento ora descritto fornisce i relativi indirizzi. Occorre tenere presente che tutte queste routine utilizzano la posizione corrente del cursore.