

# Appendice VII

## Uso del generatore grafico VI.OR

a cura di Daniele Lenzi

### 1 Il generatore VIR#

Nella versione per commodore 64 si è utilizzato un generatore grafico che consente l'utilizzo di istruzioni evolute in BASIC. Questo perché il sistema operativo del C64 non dispone di routine veloci in linguaggio macchina per la gestione della pagina grafica.

Il programma grafico VIR# che il lettore può trovare all'inizio della cassetta fornita con il libro deve essere caricato all'accensione del calcolatore in quanto tutti i programmi presentati ne fanno uso.

VIR# consente di lavorare sia con il modo grafico monocoloro (HGR) sia con quello multicolore (MGR) in cui si possono definire tre colori scelti tra i sedici disponibili più il colore di fondo. In entrambi i casi il punto è disegnato su una finestra rappresentante il quarto quadrante di un piano cartesiano con un range di 320 per 200 punti, tuttavia nel modo MGR la risoluzione orizzontale si dimezza. Grazie al generatore si può inoltre scrivere testi in pagina grafica in qualsiasi posizione, si possono gestire in modo semplice gli SPRITE e costruire forme definite in apposite tabelle (Istruzione DRAW) e risulta possibile salvare su nastro o disco la pagina grafica appena creata per poterla successivamente stampare attraverso una routine di hardcopy che però non è fornita in questo programma.

VIR# utilizza per disegnare l'ultima pagina grafica cioè quella allocata da \$E000 a \$FFFF e per i colori l'area compresa tra \$CB00 e \$CFFF. È possibile però scambiare il contenuto dell'ultima pagina grafica con quella allocata da \$A000 a \$BFFF. Il generatore grafico può essere azionato in caso di reset con SYS 35799 o molto più semplicemente con RUN/STOP + RESTORE. Se VIR# è utilizzato in modo diverso da questo contesto è consigliabile dare un NEW subito dopo il caricamento dello stesso. Si forniscono ora l'elenco di tutte le istruzioni aggiunte dal generatore grafico che come si può vedere devono essere precedute dal simbolo (@).

## 1.1 Comandi per il modo testo

**@HTAB N** Tabulazione orizzontale di  $N$  colonne ( $0 \leq N \leq 39$ )

**@VTAB N** Tabulazione verticale di  $N$  righe ( $0 \leq N \leq 24$ )

**@P N** Pausa di  $N$  decimi di secondo ( $1 \leq N \leq 2550$ ) es **@P 20**: pausa di 2 secondi

## 1.2 Comandi per Modo grafico

**@HGR A,B** Passa al modo grafico ad alta risoluzione monocoloro definendo tra i sedici colori disponibili  $A$  come colore di fondo e  $B$  come colore del punto disegnato.  $A$  e  $B$  sono numeri compresi tra 0 e 15.

**@MGR A,B,C,D** Passa al modo grafico ad alta risoluzione multicoloro che consente l'utilizzo di tre colori  $B, C, D$  per i grafici ed  $A$  per il fondo.

**@TEXT B,G,C** Torna al modo testo definendo  $B$  come colore del bordo  $G$  come colore di fondo e  $C$  come colore del cursore. Tavole di corrispondenza tra numeri e colori relativi si trovano nel manuale del calcolatore.

**@CLEAR** Pulisce la pagina grafica utente (\$E000 – \$FFFF)

**@HCOL N** Definisce il colore da utilizzare il base alla posizione con cui sono stati scelti con il comando HGR o MGR. Con riferimento a MGR se  $N$  vale:

0 il colore scelto è  $A$  (Cancellazione)

1 il colore scelto è  $B$

2 il colore scelto è  $C$

3 il colore scelto è  $D$

Nel modo HGR sono validi solo i valori 0 (Cancellazione) e 1 (Scrittura), in ogni caso se  $N = 4$  si disegna il complementare di ciò che si disegna sullo schermo (EOR). Per uscire dal modo EOR è necessario ridefinire HGR.

**@HPLOT X,Y** Traccia il punto di coordinate  $X;Y$  del colore definito con HCOL. La visualizzazione sullo schermo è relativa al quarto quadrante del piano cartesiano con  $0 \leq X \leq 319$  e  $0 \leq Y \leq 199$ , tuttavia essendo l'algebra del punto calcolata con due byte si possono inserire in  $X$  e  $Y$  valori compresi tra  $-32767$  e  $32768$ . esempio:

10 @CLEAR (Pulisce la pagina grafica)  
20 @HGR 0,1 : @HCOL 1 (Grafica e colori)  
30 @HPLOT 100,100 (Traccia il punto 100,100)  
40 @P 30 (Aspetta tre secondi e termina)

Nell'esempio di traccia un punto di coordinate 100,100 di colore bianco su sfondo nero.

**@HPLOT X1,Y1 TO X2,Y2 TO ... Xn,Yn** Traccia la spezzata che unisce gli  $n$  punti specificati

**@HPLOT TO X,Y** Unisce le estremità dell'ultima linea tracciata con il punto  $X,Y$  specificato.

**@T** Effettua lo scambio della pagina grafica. Tutte le istruzioni **VIR#** agiscono sulla pagina grafica ( $\$E000 - \$FFFF$ ), ma se ne può disporre di un'altra utilizzabile come buffer d'immagine ( $\$A000 - \$BFFF$ ). L'istruzione considerata effettua lo scambio tra queste due pagine.

**@SPRITE N,S,F,C,X,Y** Se  $S = 1$  accende lo sprite  $N$  ( $0 \leq N \leq 7$ ) alle coordinate  $X,Y$  ( $0 \leq X \leq 511, 0 \leq Y \leq 255$ ) del colore  $C$  e della forma  $F$ . Se  $S = 0$  spegne lo sprite. Se  $IB$  è la locazione di inizio del banco di memoria in cui si sta lavorando  $F$  è un numero tale che  $F \times 64 + IB$  fornisce la posizione iniziale dove sono memorizzati i 64 dati che costituiscono uno sprite, per cui lo sprite  $N$  assumerà la forma definita in questi dati. Con **VIR#** nel modo testo  $IB = 0$  mentre in pagina grafica  $IB = 49152$ . Questo è il motivo per cui gli sprite sono memorizzati a partire da quest'ultima locazione che consente di poterli utilizzare in pagina grafica. Per maggiori delucidazioni si può osservare il programma 3, ma per la costruzione degli sprite si rimanda al manuale del calcolatore.

**@SMOVE N,X,Y** Sposta lo sprite  $N$  alle coordinate  $X,Y$  l'operazione avviene con lo spegnimento dello sprite durante la traslazione. Ciò risulta di maggior effetto quando gli spostamenti sono notevoli. È da notare che lo schermo virtuale delle animazioni (Sprite) ha un'origine diversa rispetto alle coordinate della pagina grafica.

**@DRAW N,X,Y** Traccia la forma  $N$  alle coordinate  $X,Y$ . La forma è definita con una sequenza di traslazioni ortogonali unitarie così codificate:  
0 ↓ OR 4 Traslazione senza tracciamento  
1 → OR 4 Traslazione senza tracciamento  
2 ↑ OR 4 Traslazione senza tracciamento  
3 ← OR 4 Traslazione senza tracciamento

La sequenza di questi codici definisce la forma da tracciare. Ogni forma è separata dall'altra dal codice 128. L'insieme di tutte le forme costituisce la shape table (tabella di forma) la cui fine è specificata dal codice 255. L'inizio della shape table deve essere comunicata attraverso il puntatore 736 – 737.  $N$  fa riferimento al numero di ordine delle forme.

**AVVERTENZA:** Questa routine è specifica per l'uso in questa sede. La sua generalizzazione può portare ad errori cumulativi di calcolo, e conseguente tracciamento, quando la forma sia costituita da un numero elevato di traslazioni. Questo perché il calcolo delle componenti di rotazione è effettuato con una risoluzione di soli 9 bit.

**@SCALA  $SX,SY$**  Definisce la scala con cui l'istruzione DRAW traccierà la forma.  $SX,SY$  sono i fattori di scala orizzontali e verticali e sono numeri compresi tra 2 e 255.

**@ROT  $R$**  Definisce l'angolo di rotazione per la funzione draw. La rotazione  $R$  è espressa in  $360/256$  gradi ed i suoi limiti sono compresi tra 0 e 255.

**@S'Stringa', $D$**  Salva la pagina grafica utente ( $\$E000 - \$FFF$ ) sul dispositivo  $D$  specificato. Il salvataggio su nastro impiega a velocità standard circa 3 minuti, mentre con l'ALLFAST# (vedi oltre) il tempo si riduce a circa 20 secondi.

**@L'Stringa', $D$**  Carica una pagina grafica precedentemente salvata con S dal dispositivo  $D$  specificato. Valgono le stesse considerazioni di carattere temporale.

**@TEST 'Stringa', $X,Y,M,C$**  Stampa il testo 'Stringa' in pagina grafica. Scrive la stringa specificata a partire dalle coordinate  $X,Y$  del colore  $C$  ( $C$  è indipendente dalla definizione fatta con HGR e HCOL, e può essere un numero compreso tra 0 e 15).  $M$  è un flag che se vale 0 stampa nel set MAIUSCOLO-GRAFICO, se vale 1 nel set minuscolo-maiuscolo. Con POKE 36465,176 la stampa è effettuata in OR (Il testo si sovrappone ai grafici). Con POKE 36465,207 si ritorna al modo di stampa normale.

## 2 Uso dell'ALLFAST#

L'ALLFAST# è una routine in linguaggio macchina contenuta in VIR# che accelera ogni I/O su nastro. Dopo il caricamento di VIR#, ALLFAST# è già in funzione e da quel momento ogni LOAD, SAVE, VERIFY, OPEN,

PRINT#, GET# e INPUT# riferito al dispositivo 1, sarà notevolmente accelerato. L'ALLFAST# è una routine perfettamente standard nel senso che funziona in modo identico allo standard originale; ciò rende la routine "trasparente", ma soprattutto consente di utilizzare programmi che utilizzavano il registratore senza che debbano essere modificati per adattarli alla velocità dell'ALLFAST# (Questo in BASIC, ma anche in linguaggio macchina se esso utilizza per l'I/O le routine del kernel). I programmi forniti con il libro sono velocizzati con tale routine per cui per il loro caricamento è necessario avere già in memoria l'ALLFAST#, tuttavia si possono fare modifiche a programmi proposti e quindi salvarli su un'altra cassetta sia a velocità standard che a velocità elevata. Infatti battendo #0 e quindi return si disattiva l'ALLFAST# e, da quel momento in poi la gestione dell'I/O su nastro è restituita alla velocità standard. Per tornare all'ALLFAST# battete # e quindi return.

Naturalmente l'ALLFAST# accelera anche i due comandi già visti di salvataggio e caricamento della pagina grafica.

Il programma VIR# è allocato dalle locazioni 32768 a 40959; questa è una informazione utile nel caso lo si voglia registrare nuovamente su cassetta.

Gli sprite sono invece stati allocati tra le locazioni 49152 e 51400. L'ordine di registrazione da seguire per il programma principale è il seguente:

VIR# (Da registrare a velocità standard)

HELLO (Questo e i seguenti a velocità dell'ALLFAST# se si vuole)

SPRITE

HELLO 1

MECC 1

MECC 2

MECC 3

MECC 4