



a cura di Maurizio Mauri

# Il Video

prima parte

di Sergio e Dario Neddi

■ Cominciamo finalmente a parlare del video, che è una delle cose più appariscenti dell'MSX. Per ora tratteremo del video dell'MSX1; in seguito ci ripromettiamo di approfondire anche quello dell'MSX2.

In questo primo articolo dedicato al video descriveremo le caratteristiche generali del processore video, rimandando una trattazione più completa dei vari SCREEN ed esempi pratici ai prossimi articoli. ■

Il TMS9918 o VDP (Video Display Processor) è senz'altro il chip più interessante di un computer MSX. Questo componente (od uno equivalente) gestisce da solo la VRAM (RAM video) di 16K, la grafica, il colore e gli sprite.

Ricordiamo brevemente le caratteristiche grafiche: il video dell'MSX permette l'uso di 4 schermi (SCREEN), come ormai tutti sanno. I primi 2 (lo 0 e l'1) sono di testo e gli altri 2 (il 2 e il 3) sono invece grafici.

Vediamo ora le caratteristiche salienti di questi schermi.

## Lo Screen 0

Lo SCREEN 0 è il più comunemente usato dei 2 modi testo ed è forse il meno interessante per chi fa giochi in quanto non permette gli sprite e tutti i caratteri devono obbligatoriamente avere lo stesso colore. I caratteri visualizzati sono formati a partire da una matrice di 6 punti orizzontali ed 8 punti verticali. Questa particolarità «taglia» alcuni caratteri del set MSX che sono più larghi, come ad esempio la classica faccina sorridente.

Come controparte offre la possibilità di avere 40 colonne che, anche se non sono molte (per usi più profes-

sionali occorrerebbero almeno 80 come nell'MSX2) sono già qualcosa. Le righe ammesse sono 24.

## Lo Screen 1

È uno schermo di testo con 32 colonne e 24 righe. Permette gli sprite e si possono avere caratteri di colore diverso: al loro codice ASCII è infatti possibile associare un diverso colore per il carattere e per il suo sfondo, con la limitazione che i byte di colore sono solo 32 e che quindi i colori sono validi per gruppi di 8 caratteri (infatti  $32 \times 8$  fa 256, cioè si coprono tutti i 256 codici visualizzabili dello schermo).

Queste caratteristiche rendono facile la realizzazione di giochi, ed infatti molti videogiochi commerciali lavorano in SCREEN 1.

## Lo Screen 2

Questo schermo permette di avere la grafica ad alta risoluzione e quindi è il più usato dei modi grafici. I pixel indirizzabili sono 256 per 192 con alcune limitazioni per i colori, che non possono essere più di 2 per ogni striscia orizzontale di 8 pixel. Possono co-

munque venire usati tutti i 16 colori che permette il computer e pure gli sprite.

## Lo Screen 3

È meno usato degli altri schermi in quanto la risoluzione è scarsa ( $64 \times 48$  pixel) però è immune da limitazioni sull'uso del colore: ogni pixel infatti può venire colorato con qualsiasi colore. Ciò nonostante non viene quasi mai usato se non per farvi apparire delle scritte giganti, utili per titoli e schermate introduttive.

## Gli Sprite

Come si sa il VDP è in grado di gestire 256 sprite 8 per 8 pixel normali od ingranditi, oppure 64 sprite 16 per 16 normali od ingranditi.

Sullo schermo possono però comparire al massimo 32 sprite, e non più di 4 sulla stessa riga.

Gli sprite sono monocolori; per farli multicolori è necessario sovrapporre diversi di differenti colori.

Gli sprite giacciono su 32 differenti piani. Lo sprite che ha priorità più elevata (il numero 0) può sovrapporsi a tutti gli altri, quello che ha priorità inferiore (il numero 31) può venire sovrapposto da tutti gli altri.

In VRAM vi sono 2 aree di memoria che riguardano gli sprite chiamate Sprite Descriptor Table e Sprite Attribute List. La prima è in pratica il generatore di caratteri degli sprite ed è lunga 2048 byte (8 per 256). La seconda definisce la posizione e il colore di ognuno dei 32 sprite ed è lunga 128 byte (4 byte per ogni sprite). Questi 4 byte hanno il seguente significato: il primo byte è la posizione verticale dello sprite e inizia da 255 (prima riga in alto dello schermo) seguito da 0 fino a 190; valori compresi fra 191 e 254 in-

dividano sullo schermo posizioni al di sotto dell'ultima riga e quindi nascondono lo sprite. Tuttavia il valore ODOH (208) ha l'effetto di rendere indefiniti tutti gli sprite con priorità inferiore.

Il secondo byte è la posizione orizzontale dello sprite da 0 a 255.

Il terzo byte definisce il carattere usato dallo sprite, e corrisponde al carattere definito nella Sprite Descriptor Table. Nel quarto byte i quattro bit meno significativi (da 0 a 3) definiscono il colore dello sprite (da 0 a 15) mentre il bit 7 se è 1 fa sì che la posizione dello sprite sia shiftata di 32 pixel a sinistra. Quest'ultima cosa è quella che permette agli sprite di apparire gradualmente anche nella parte sinistra dello schermo (come si fa da Basic quando si danno posizioni orizzontali dello sprite comprese fra -32 e -1).

## II VDP

Il VDP ha 8 registri interni a sola scrittura (cioè non si può leggere ciò

che è stato scritto) ed un registro a sola lettura. Da Basic è possibile leggere tutti i registri perché il Basic stesso (o meglio il BIOS) se ne fa una copia in RAM (da &HF3DF a &HF3E6). Il registro di stato viene letto dalla routine di interrupt e memorizzato nella locazione &HF3E6.

Questi registri servono per controllare tutte le funzioni del VDP ed i vari puntatori alle zone di VRAM dedicate allo schermo, agli sprite, ai caratteri, ecc.

I registri interni del VDP non sono indirizzabili direttamente con istruzioni di I/O, ma, come vedremo, sono accessibili tramite altri registri che hanno lo scopo di mettere il VDP e la VRAM in contatto con la CPU.

A proposito di VRAM, l'MSX ne ha 16K che, pur se non sono molti come i 64 o i 128K dell'MSX2 sono pur sempre una quantità rispettabile.

La VRAM contiene un sacco di cose: per esempio oltre alla mappa dello schermo, la mappa dei colori, le tabelle di definizione e gli attributi degli sprite, contiene anche il generatore di

caratteri alfanumerici e grafici che viene trasferito in VRAM dalla ROM all'accensione (e ad ogni comando SCREEN).

Il generatore di caratteri, (cioè la descrizione punto per punto di come è costituito un carattere) essendo in VRAM può venire modificato a piacere per creare caratteri personalizzati.

L'indirizzo del generatore di caratteri in ROM è memorizzato nelle locazioni 0004H e 0005H della ROM (1BBFH nei computer da noi esaminati) e si estende esattamente per 2K (infatti ogni carattere per venire definito necessita di 8 byte, essendo costituito da 8 per 8 punti; i caratteri MSX sono 256 e 256 per 8 = 2048, cioè 2K).

Un generatore di caratteri personalizzato lo possiamo costruire anche in RAM (controllata dalla CPU), così non corriamo il rischio che il nostro set di caratteri in VRAM sia modificato ad ogni istruzione SCREEN: basta mettere in 0F920H l'indirizzo del nostro generatore di caratteri e in 0F91FH lo «slot ID» (lo vedremo nelle prossime puntate) cioè lo slot dove si trova il generatore di caratteri.

### TABELLA REGISTRI VDP

#### VDP (0):

i bit da 7 a 2 sono riservati. Devono essere 000000.

#### Bit 1:

porta il VDP in bit-map mode.

#### Bit 0:

abilitazione/disabilitazione video esterno.

Se 1 abilita ingresso video esterno.

#### VDP (1):

#### Bit 7:

seleziona l'ampiezza della VRAM. Se posto a 0 il VDP si predispose a gestire una VRAM di 4K, se posto a 1 una VRAM di 16K.

#### Bit 6:

abilitazione/disabilitazione del video. Se viene posto a 0 sparisce lo schermo, cioè tutto lo schermo prende uniformemente il colore del bordo e l'immagine riappare soltanto rimettendo questo bit a 1.

#### Bit 5:

abilitazione/disabilitazione dell'interrupt. Ponendo questo bit a 0 vengono disabilitati gli interrupt generati dal VDP e quindi vengono a bloccarsi tutte le funzioni associate a questo interrupt (come ad esempio la scansione della tastiera).

#### Bit 4:

se posto a 1 il VDP passa in modo testo (SCREEN 0).

#### Bit 3:

se posto a 1 il VDP passa in modo multi-color (SCREEN 3).

#### Bit 2:

riservato: deve essere 0.

#### Bit 1:

misura degli sprite. Se posto a 0 la misura è 8x8, se è 1 la misura è 16x16.

#### Bit 0:

Dimensione degli sprite. Se posto a 0 la dimensione è normale, se posto a 1 la dimensione è doppia.

#### VDP (2):

definisce la base della Screen Image Table, cioè la zona di VRAM adibita a memoria di schermo.

L'indirizzo in VRAM è il valore di questo registro moltiplicato 0400H, cioè una variazione di una unità equivale ad una variazione di 0400H (1K) della base.

#### VDP (3):

definisce la base della Color Table, cioè la zona di VRAM adibita a mappa dei colori.

L'indirizzo in VRAM è il valore di questo registro moltiplicato 0040H, cioè ogni unità sposta la mappa di 64 byte.

#### VDP (4):

definisce l'indirizzo per la Pattern Descriptor Table. Questo è il valore del registro per 0800H, cioè è possibile definire l'indirizzo per passi di 2K. Si tratta del generatore di caratteri.

#### VDP (5):

definisce l'indirizzo per la Sprite Attribute List.

L'indirizzo si calcola moltiplicando il valore del registro per 0080H, cioè l'indirizzo è definibile per passi di 128 byte.

#### VDP (6):

definisce la base per la Sprite Descriptor Table.

L'indirizzo si calcola moltiplicando il valore del registro per 0080H, cioè l'indirizzo è definibile per passi di 2K.

#### VDP (7):

Il nybble più significativo definisce il colore dei caratteri in SCREEN 0. Il nybble meno significativo definisce il colore dello sfondo in SCREEN 0 e il colore del bordo in tutti gli SCREEN.

## Significato dei registri VDP

Citiamo il significato dei vari registri, che abbiamo tratto dal manuale EDITOR/ASSEMBLER del TI 99/4A, computer che possiede lo stesso integrato video.

N.B. Se andate a consultare questo manuale tenete conto della numerazione dei vari bit: infatti normalmente i vari bit che compongono un byte vengono numerati in ordine decrescente da sinistra a destra, per cui il bit più significativo è il 7 e quello meno significativo è il bit 0. Per la Texas Instruments invece l'ordine è inverso, per cui il bit più significativo è il bit 0 e quello meno significativo è il bit 7. Occorre quindi «tradurre» l'ordine dei bit per evitare confusione (ed infatti abbiamo fatto la traduzione nella tabella pubblicata a fianco).

In SCREEN 2 i VDP (3) e VDP (4) fanno eccezione: sia la mappa dei colori che il generatore di caratteri sono lunghi 6K e possono iniziare solo agli indirizzi 0 e 2000H. Scrivendo nel VDP (3) i valori 07FH o 0FFH si definisce la base della Color Table rispettivamente a 0 o a 2000H; analogamente se si vuole l'inizio del generatore di caratteri a 0 o a 2000H bisogna scrivere i valori 3 o 7 nel VDP (4).

Infine se si vuole entrare in SCREEN 1 i 3 bit di modo (cioè il bit 1 del VDP (0) e i bit 3 e 4 del VDP (4)) debbono essere contemporaneamente 0.

Oltre a questi registri di sola scrittura ce n'è uno a sola lettura: si tratta dello Status Register ed è accessibile



da Basic tramite la variabile VDP (8). Vediamo il significato dei vari bit che la compongono.

**VDP (8):****Bit 7:**

flag di interrupt: se è 1 significa che vi è una richiesta di interrupt; in questo caso leggendo il registro di stato questo bit viene resettato, annullando in sostanza la richiesta di interrupt.

**Bit 6:**

flag del quinto sprite: se è 1 significa che vi è un quinto sprite allineato.

**Bit 5:**

flag di collisione: se è 1 significa che ci sono degli sprite in collisione.

**Bit da 0 a 4:**

contengono il numero del quinto sprite allineato.

### Contenuto di Default dei registri VDP (in HEX)

	SCREEN 0	SCREEN 1	SCREEN 2	SCREEN 3
VDP (0):	00	00	02	00
VDP (1):	F0	E0	E0	E8
VDP (2):	00	06	06	02
VDP (3):	00	80	FF	00
VDP (4):	01	00	03	00
VDP (5):	00	36	36	36
VDP (6):	00	07	07	07
VDP (7):	F4	04	04	04

L'istruzione BASE (N) del Basic fornisce l'indirizzo iniziale delle tabelle del VDP, cioè si tratta dei valori che, opportunamente elaborati dal sistema, vanno a programmare i registri VDP (da 2 a 6) in modo da farlo puntare agli indirizzi corretti in VRAM nei vari modi grafici; il parametro «N» deve essere compreso tra 0 e 19 (decimale).

Ecco gli indirizzi iniziali delle tabelle del VDP (è in definitiva una mappa della VRAM):

SCREEN	Schermo	Colori	Gen.Car.	Attr.Spr.	Prof.Spr.
0	0	—	2048	—	—
1	6144	8192	0	6912	14336
2	6144	8192	0	6912	14336
3	2048	0	—	6912	14336

Per chiarezza abbiamo riportato le voci Schermo, Colori, Gen.Car., Attr.Spr. e Prof.Spr. invece delle originali Screen Image Table, Color Table, Pattern Descriptor Table, Sprite Attribute List e Sprite Descriptor Table.

### Il VDP in L.M.

Il VDP si presenta alla CPU come un insieme di 4 registri: uno che per-

mette di indirizzare la VRAM oppure i registri interni, altri 2 per la lettura/scrittura dei dati da e verso la VRAM e i registri interni, ed il registro di stato che viene letto dalla CPU ogni 50esimo di secondo. Sugli MSX i due registri di lettura/scrittura coincidono nella porta 98H (data port) e gli altri due coincidono nella porta 99H (command port).

Per accedere direttamente alla VRAM (per quanto sia più semplice accedervi tramite le routine del BIOS) bisogna tener conto che gli indirizzi VRAM possono assumere valori compresi tra 0 e 16383 e che il registro di indirizzamento del VDP è ad 8 bit. Dovremo quindi inviare l'indirizzo con due operazioni di I/O, col sistema basso-alto.

Per scrivere in una cella di VRAM bisogna innanzitutto disabilitare l'interrupt, poi inviare il byte meno significativo all'indirizzo 99H, poi settare il bit 6 del byte più significativo e mandarlo sempre in uscita all'indirizzo 99H, quindi inviare il dato che si vuole scrivere nella VRAM all'indirizzo 98H.

Se si vuole invece leggere dalla VRAM l'operazione è analoga alla precedente, ma invece di settare il bit 6 all'atto dell'invio della parte più significativa dell'indirizzo lo si resetta e poi si può andare a leggere il dato dal registro mappato a 98H. Questo bit è evidentemente il flag di lettura/scrittura.

Da notare che il registro di indirizzamento del VDP è autoincrementante, cioè una volta inviato il primo dato questo va ad indirizzare il byte successivo; si possono quindi scrivere (o leggere) più byte consecutivi senza dover riscrivere di volta in volta l'indirizzo.

Per accedere direttamente ai registri del VDP si usa solo la porta 99H; prima si invia il dato da scrivere, poi si invia il numero del registro con il bit 7 posto a 1.

Infine per leggere il registro di stato basta semplicemente leggere la porta 99H.

### Esempio di scrittura in VRAM

Supponiamo di voler scrivere il byte OFFH nella cella di VRAM di indirizzo 0400H, contenuto nella coppia di registri HL:

```
LD HL,0400H ;indirizzo
DI
LD A,L ;invia il byte
OUT (99H),A ;meno significativo
LD A,H
OR 40H ;setta il bit 6
OUT (99H),A ;byte più significativo
EI
LD A,OFFH
OUT (98H),A ;invia il dato
```

Naturalmente questo è solo un esempio per chiarire il funzionamento del VDP; per scrivere (e leggere) nella VRAM è molto più pratico usare le routine del BIOS.

Chi invece sta programmando un videogioco ed ha bisogno quindi di una elevatissima velocità di visualizzazione, non potendo fare a meno di accedere direttamente al VDP tenga presente che gli indirizzi di I/O (98H e 99H) potrebbero non essere validi su tutti gli MSX.

Infatti la locazione 0006H della ROM contiene il numero della porta per la lettura del dato dal processore video e la locazione 0007H della ROM contiene il numero della porta per la scrittura del dato nel processore video. In tutti i computer attualmente in commercio è sempre 98H per entrambe le locazioni, ma il fatto che ci sia tutta questa roba in ROM dimostra che su alcune macchine potrebbero aversi delle sorprese.

A tal proposito citiamo il fatto che nello standard MSX è prevista una espansione capace di trasformare un MSX1 in MSX2; questo adattatore prevede per I/O con il nuovo processore video, il 9938, l'uso delle porte 88H, 89H, 8AH e 8BH dato che le 98H e 99H sono già occupate.

Ora riprendiamo in esame la routine presentata più sopra e vediamo come risulta semplificabile sfruttando il BIOS:

```
LD HL,0400H
LD A,OFFH
CALL 004DH
```

dove la routine che inizia all'indirizzo esadecimale 004DH effettua appunto la scrittura.

Come ulteriore esempio riportiamo una routine per cambiare il colore del bordo (non in SCREEN 0) in rosso (colore 6):

```
LD A,6 ;dato
OUT (99H),A
LD A,7 ;registro
OR 80H ;setta il bit 7
OUT (99H),A
```

semplificabile anch'essa usando le routine del BIOS:

```
LD BC,67H
CALL 0047H
```

Ambedue le routine sono equivalenti all'istruzione Basic: VDP (7) = 6.

La prossima volta vedremo in dettaglio le routine che agiscono sul VDP con qualche esempio pratico.