

Parliamo un po' di mouse...

A partire da questa puntata ci occuperemo di un argomento nuovo e che speriamo risulti utile ai programmatori: analizzeremo il «mouse» dal punto di vista della sua gestione software, gestione che può avvenire in praticamente qualunque linguaggio, dall'Assembler al Basic e dal C al Pascal...

A cosa serve e come è fatto un «mouse»

Il mouse, come i lettori ben sapranno, è un particolare dispositivo di Input collegabile al nostro computer: dotando quest'ultimo di un apposito driver e facendo girare un programma che per l'appunto gestisce il mouse (ad esempio un programma di grafica), abbiamo a disposizione uno strumento che ci permette di spostare a nostro piacimento il cursore nello schermo, in alternativa ai tasti «freccia», muovendo semplicemente il mouse sul tavolo. In particolare all'interno del mouse stesso c'è una pallina che ruota non appena facciamo scivolare il mouse sul tavolo e per mezzo di appositi dispositivi il moto di questa pallina viene trasformato nella corrispondente variazione di due coordinate (una «x» ed una «y»), che vengono incessantemente inviate al computer per mezzo di una porta seriale.

Il driver presente sul PC (e fornito in dotazione di ogni mouse in commercio) provvede istante per istante a leggere i dati provenienti dal mouse (in generale appunto le due coordinate x ed y nonché lo stato dei due o tre pulsanti

posti sul mouse) ed a renderli disponibili in maniera codificata ai programmi che vogliono (e sanno...) usarli.

Abbiamo parlato di driver del mouse, che in genere si può trovare sotto forma di due programmi: il primo si chiama quasi sempre «MOUSE.COM» e perciò è direttamente eseguibile in qualunque momento sotto DOS mentre il secondo, se presente, si chiama «MOUSE.SYS» ed è un vero e proprio «device driver» e come tale può essere avviato solo al bootstrap ponendo il comando:

```
device = mouse.sys
```

all'interno del file «CONFIG.SYS» e con eventuali parametri aggiuntivi, quali ad esempio il numero della porta seriale a cui è collegato il mouse.

È questo driver appunto, che si preoccupa in maniera «trasparente» (e perciò invisibile all'utente) di gestire i byte ricevuti dalla porta seriale (in genere la «COM1:» e con una velocità di 1200 Baud), lavorando «sotto interrupt» e cioè essendo interpellato solamente quando è disponibile un byte da parte della porta seriale e viceversa lasciando per il restante tempo il controllo al sistema operativo e/o al programma che è in corso di esecuzione.

Abbiamo detto pure che il driver effettua certe operazioni e rende disponibili dei dati: tali dati possono essere gestiti a loro volta per mezzo di un apposito dispositivo software, l'INT 33H, alla stessa maniera di come, ad esempio, si può gestire il video a partire dall'INT 10H e cioè ponendo in registri ben definiti dei valori opportuni ed ottenendo viceversa le «risposte» come contenuti di altri registri ben prestabiliti. A pensarci bene questo è un metodo assai ben collaudato ed efficace, che consente di effettuare una serie anche molto grande di possibili operazioni al variare di un valore posto nel registro AH ed in particolare funzione del contenuto di altri registri del microprocessore.

Prima però di parlare del mouse dal punto di vista software, vediamo di dire qualcosa sul «mondo» in cui si muove il cursore a seguito di spostamenti del mouse stesso.

La pagina del video

Come è ben noto, il mouse lavora sia

AX = 0H	Mouse Reset
AX = 1H	Cursor Enable
AX = 2H	Cursor Disable
AX = 3H	Get Mouse Position and Button Status
AX = 4H	Set Mouse Position
AX = 5H	Get Button Press Information
AX = 6H	Get Button Release Information
AX = 7H	Set min & max horizontal position
AX = 8H	Set min & max vertical position
AX = 9H	Set Graphic Cursor Block
AX = 0AH	Set Text Cursor
AX = 0BH	Read Motion Counters
AX = 0CH	Set User-defined subroutine
AX = 0DH	Enable Light Pen Emulation
AX = 0EH	Disable Light Pen Emulation
AX = 0FH	Set Mickey/Pixel Ratio
AX = 10H	Window Conditional Off
AX = 12H	Set Large Graphic Cursor
AX = 13H	Set Speed Threshold

Elenco delle funzioni a disposizione per la gestione di un mouse per mezzo dell'INT 33H: ad eccezione della funzione 12H, si tratta di funzioni standard alle quali occasionalmente se ne affiancano di nuove.

sotto grafica che in semplice modo alfanumerico, anche se siamo in pratica più abituati a vedere programmi (da Windows al DrHalo, dall'AutoCAD al PaintBrush, tanto per citare alcuni esempi) nei quali il cursore, a seconda delle circostanze, assume aspetti ben differenti.

Nei programmi citati si hanno ad esempio i seguenti tipi di cursore (in realtà in alcuni casi ce ne sono altri ancora, ma non vogliamo elencare tutte le possibili variazioni):

— sotto Windows il cursore è generalmente una freccia bianca, ma che diventa subito una clessidra laddove l'operatore deve attendere, oppure una croce (nell'«Otello»), oppure ancora una barretta verticale;

— DrHalo invece è un po' più spartano, ma essenziale, presentando una piccola crocetta, che può in genere diventare un rettangolo «elastico» oppure un cerchio o un'ellisse altrettanto «estendibili» a seconda della funzione che si desidera svolgere;

— per AutoCAD praticamente vale quanto detto per il DrHalo, salvo il fatto che le possibilità sono ancora più estese;

— PaintBrush infine già all'interno di un'unica schermata presenta un cursore variabile tra freccette che puntano verso l'alto e verso sinistra o in diagonale e tra i tipi di cursore gestibili nella pagina grafica vera e propria, che ad esempio possono essere cerchietti o quadratini pieni, di ampiezza prestabilita.

Altro fatto importante è che quasi tutti questi tipi di cursore sono dotati di «colore», che cambia a seconda del colore dello sfondo in cui si trova a transitare il cursore stesso, in modo tale da renderlo sempre visibile (anche se in alcuni casi possono capitare dei colori non molto contrastati).

Inoltre il tutto avviene indifferentemente dal tipo di scheda grafica utilizzata, anche perché in genere tutti i programmi di questo tipo sono installabili in un gran numero di schede ed il mouse, salvo rarissime eccezioni, si comporta fedelmente di conseguenza.

Questo per quel che riguarda la grafica: anche in modo testo il cursore, e quindi il mouse, può fare la parte del leone e come esempio di questo citiamo su tutti il QuickBasic (confidenzialmente «QB») dotato di un bel cursore rosso lampeggiante che convive tranquillamente con quello ordinario (una sottolineatura bianca lampeggiante), salva fatta la possibilità in ogni istante di metterli «in passo», cioè ad esempio per spostare velocemente il punto in cui si vogliono effettuare correzioni al listato del programma.

Tutto quanto detto finora, salvo le solite rarissime eccezioni, è gestibile via software per mezzo di apposite chiama-

te all'INT 33H (attenzione, è proprio 33 esadecimale corrispondente al valore 51 decimale, che NON verrà più citato nel seguito): è quanto ci proponiamo di fare, corredando il tutto con piccoli esempi «multi-linguaggi».

Apriamo una parentesi. I linguaggi di cui sopra saranno l'Assembler, il Turbo Pascal ed il Basic (del BasicA e non del QB visto che quest'ultimo risulta un po' meno interattivo da questo punto di vista) e non il «C» per un'atavica idiosincrasia del redattore verso tale «linguaggio» a tutto favore del meraviglioso Turbo Pascal. Ciò detto chiudiamo la parentesi.

Parlavamo dunque dei colori e dei differenti tipi di schede grafiche: i lettori ben sanno che ogni tipo di scheda ha una sua particolare risoluzione in termini di pixel rappresentabili, nonché di colori associabili ai singoli pixel.

Senza voler fare un trattato completo sui vari modi video, per i quali rimandiamo a vecchi e a prossimi articoli (è una promessa... n.d.r.) possiamo riassumere nella tabella A le principali schede in circolazione con a fianco riportate una o più possibili risoluzioni.

Oltre alle ben note schede abbiamo aggiunto una generica «Extended VGA» (di qualsiasi marca e modello), capace di prestazioni notevolissime: il modo testo «100 x 75» in particolare consente di leggere su schermo il contenuto completo di una cartella dattiloscritta in formato A4 (a 66 righe), nonché il formato a 72 righe, tipico dei moduli continui per stampanti! Per non parlare poi del modo grafico «1024 x 768» che però richiede dei monitor ben al di là delle capacità d'acquisto di un utente medio...

Qualunque sia il modo video prescelto, tanto in grafica quanto in modo testo, il mouse «vede» sempre un rettangolo le cui coordinate vanno da (0,0), al solito in alto a sinistra, al punto avente come x ed y il massimo valore rag-

giungibile in risoluzione orizzontale e verticale: come dire che comunque i valori «x» ed «y» forniti istante per istante dal driver del mouse si riferiscono alla situazione «grafica», anche se si è in modo testo.

Per chiarire meglio facciamo un esempio, quello più «normale»: vogliamo il cursore in una pagina «80 x 25» su di una CGA. In tal caso il mouse vedrà sempre una «x» compresa tra 0 e 639, per cui il cursore sarà sulla colonna data da «x / 8»; viceversa la «y» varierà tra 0 e 199 ed anche la linea in cui si trova il cursore sarà data da «y / 8»: questo perché il cursore ed i generici caratteri nella CGA sono ampi «8x8» pixel.

In generale, specie dalle EGA in su, in cui ci possono essere più font di caratteri di ampiezze (in termini di pixel) differenti, bisognerà dividere la «x» per l'ampiezza orizzontale del singolo carattere espressa in pixel ed analogamente fare per la «y», comunque il tutto è veramente semplice ed al limite si può procedere per tentativi...

Altro fatto di cui si deve tener conto è che già nella CGA in modalità «4 colori» non c'è più la corrispondenza uno a uno tra bit di locazioni di memoria e pixel: in questo caso ad esempio ad ogni pixel, che può assumere 4 colori, corrispondono effettivamente 2 bit affiancati del byte corrispondente nella memoria video: peggio si ha nell'EGA e nella VGA in cui ad ogni pixel corrispondono bit di byte appartenenti a più «piani» di memoria per cui la corrispondenza diventa ancora più complessa. Meglio di tutti si comporta la VGA nel modo a 256 colori (però con risoluzione 320x200, purtroppo...) nel qual caso ad ogni pixel corrisponde un byte della memoria, byte che come noto consente di rappresen-

scheda video	risoluzioni possibili	
	testo	grafica
CGA	40 x 25	320 x 200
	80 x 25	640 x 200
Hercules	80 x 25	720 x 348
EGA	40 x 25	320 x 200
	80 x 25	640 x 200
	80 x 43	640 x 350
VGA MCGA	40 x 25	320 x 200
	80 x 25	640 x 200
	80 x 43	640 x 350
	80 x 50	640 x 480
Extended VGA	40 x 25	320 x 200
	80 x 25	640 x 200
	80 x 43	640 x 350
	80 x 50	640 x 480
	132 x 25	1024 x 768
	132 x 60	800 x 600
	100 x 75	

Tabella A

```

100 DEF SEG = 0
110 MOUSESEG = PEEK(51 * 4 + 3) * 256 + PEEK(51 * 4 + 2)
120 MOUSE = PEEK(51 * 4 + 1) * 256 + PEEK(51) + 2
130 DEF SFG = MOUSESEG

```

Figura 2

```

uses dos;

var reg : registers;

procedure mouse(ax : word);
begin
  reg.AX := ax;
  intr($33,reg);
end;

begin
  mouse(0); {vedremo che serve ad inizializzare il mouse}
  if reg.AX <> $ffff
  then halt(1); {AX = -1 significa che tutto e' ok}
  {...}
end.

```

Figura 1

tare 256 valori, appunto il colore del pixel corrispondente (a proposito: il «pixel» in questo caso ha delle dimensioni veramente notevoli, rispetto a quelle del modo grafico 640x480, viceversa gradevolmente «piccole»...).

Veniamo dunque al software e all'INT 33H

Prima di buttarci a capofitto nell'analisi delle varie funzioni a disposizione dell'utente (ricordiamoci di caricare il driver prima di compiere qualunque esperimento...), diciamo subito che la «standardizzazione de facto» in merito ai mouse è quella imposta dalla Microsoft, la quale ha per prima prestabilito le funzioni gestibili tramite l'INT 33H.

Lo standard è nato dal fatto che tutti i produttori di mouse, per proporre un mouse compatibile, devono fornire un set di funzioni completamente compatibile con quello della Microsoft e cioè comprendente almeno tutte le funzioni previste dallo standard: infatti, siccome l'appetito vien mangiando (come è successo nel caso delle schede «super-VGA»), di solito i mouse via via introdotti in commercio possiedono caratteristiche e funzioni sempre più evolute e perciò tutte quelle che sono funzioni aggiuntive sono più che benvenute, ma poi ovviamente non bisogna pretendere che un programma che le gestisca in pieno possa essere poi «compatibile verso il basso». Dato a Cesare quel che è di Cesare, passiamo all'analisi delle possibili funzioni «base» dell'INT 33H: faremo sempre riferimento alla tabella pubblicata nella prima pagina dell'articolo che vuole solamente essere un minimo di riferimento, lasciando al testo le spiegazioni in merito.

La chiamata delle funzioni

Prima ancora di iniziare l'analisi delle singole funzioni, che demandiamo alla prossima puntata, vediamo ora, come promesso, come si effettua la chiamata alle funzioni nei tre linguaggi citati in precedenza. Per quanto riguarda l'Assembler, che è e sarà implicitamente sottinteso soprattutto per quel che riguarda i registri di «interfacciamento» con le routine, prima di ogni chiamata bisogna inizializzare opportunamente i registri necessari, compreso il registro AX che dovrà contenere il numero della funzione stessa dopodiché non resta altro che inserire la chiamata all'interrupt.

In generale dunque avremo dei frammenti di programma di questo tipo:

```

...
MOV AX,numerofunzione
MOV BX, ...      \
MOV CX, ...      > solo se necessario
MOV DX, ...      /
...
INT 33H          altri registri laddove servano

```

Per quanto riguarda il Turbo Pascal il tutto è molto semplice e l'effettiva implementazione dipende esclusivamente dai gusti personali: il metodo usato dal redattore della rubrica prevede la creazione di una «procedura» alla quale si passa il numero della funzione, demandando ad una variabile globale «reg» (di tipo «registers», tipo automaticamente definito per mezzo della clausola «uses dos» in testa al programma), i cui record sono proprio i registri che dobbiamo inizializzare e/o leggere.

In particolare suggeriamo un'implementazione come è rappresentato in figura 1.

Questa implementazione, alquanto

scarna, lo riconosciamo, ha il vantaggio di tenere sempre sotto controllo i registri tramite la variabile «reg» oltre a chiamare la routine di gestione del mouse con un nome inequivocabile...

Comunque vediamo di buon occhio anche una versione della procedura in cui come parametri passiamo anche i registri, ma tutto sommato è una fatica inutile, in quanto già esiste la variabile globale «reg»: eppoi una chiamata del tipo:

```
mouse (8,0,0,100);
```

oltre che illeggibile (chi si ricorda dopo un po' che il terzo parametro è CX, ecc...?!) è molto «Basica» (aggettivo derivato dal Basic), il che non depone certo a favore del Pascal.

Invece in Basic bisogna inventarsi un bel po' di roba:

— innanzitutto bisogna calcolarsi l'indirizzo della routine posta in memoria e che realizza il driver del mouse e questo si fa con le istruzioni riportate in figura 2;

— poi si può chiamare la routine passando (e ricevendo) come parametri quattro valori interi rappresentanti i valori dei registri AX, BX, CX, e DX, grazie alle variabili M1%, M2%, M3% ed M4% (ma si possono scegliere altri nomi!!!):

```
398 M1% = numerofunzione: 'chi si ricorda che è AX?!
```

```
399 M4% = qualcosa: 'che registro sarà?!
```

```
400 CALL MOUSE(M1%,M2%,M3%,M4%)
```

senza contare che così non si possono chiamare funzioni che utilizzano altri registri: vedremo infatti che in alcuni casi serviranno anche i registri ES, SI e DI, con l'obbligo dunque di crearsi delle routine in linguaggio Assembler, a meno di non buttare tutto a mare e passare a linguaggi più seri e potenti.

Con questo abbiamo dunque terminato questa prima parte: dalla prossima inizieremo l'analisi «a tappeto» delle funzioni, alcune delle quali molto interessanti.

Il tutto in modo veramente semplice, come sarà facile constatare...

Byte Line



STAMPANTI

Dela Printer, 180 cps, 80 col.	L. 499.000
NEC P6 Plus 24 aghi	L. 1.298.000
NEC P7 Plus 24 aghi	L. 1.698.000
NEC P2200 24 aghi	L. 649.000
Citizen 120 D	L. 298.000
Citizen MSP-15 E	L. 549.000
Citizen HQP-40	L. 949.000
Citizen 180 E	L. 349.000
Star LC-10	L. 399.000
Star LC-10 color	L. 499.000
Star LC-24 1024 aghi	L. 649.000
Cavo IBM - Centronics	L. 13.900
Epson LQ-500	L. 649.000

COMPUTER

XT compatibile 10 MHz	da L. 690.000
AT compatibile 12 MHz	da L. 1.190.000
AT completo 512K HD 20Mb	L. 1.999.000

ACCESSORI

Handscanner 105 mm	L. 399.000
IBM- Mouse	L. 79.000
GENOA SuperEGA Hires	L. 498.000
Genoa Super VGA 5200	L. 699.000
Modem 1200H interno	L. 178.000
Modem 1200C esterno	L. 239.000
Fax Murata M-1	L. 1.390.000
ordine minimo 100 dischetti 100% Errorfree	
Dela Disk 5.25" 2D	L. 840
Dela Disk 3.5" 2DD	L. 2.100
No Name 5.25" 2D	L. 690
No Name 3.5" 2DD	L. 1.890
No Name 5.25" 2HD 1.2 Mbyte	L. 2.100
Diskbox per 100 Floppy 5,25"	L. 14.900
Diskbox per 50 Floppy 3,5"	L. 14.900

I PREZZI SI INTENDONO
AL NETTO DI I.V.A.

HARD DISK

Seagate File Card	L. 799.000
ST225 21,4 MB	L. 349.000
ST251 42,8 MB	L. 678.000
ST250 40 MB RLL incl. contr	L. 699.000
AMIGOS 20 MB Hard-Disk per	
AMIGA 500 o AMIGA 1000	L. 999.000

Prezzi suscettibili alla variazione del dollaro!!!

MONITOR

Flatscreen Dual	
Frequency Invers	L. 238.000
NEC Multisync II	L. 1.098.000
Mitsubishi Multisync	
EUM 1481 A	L. 998.000
Cavo Mitsubishi - VGA	L. 39.000

VENDITA PER CORRISPONDENZA

Byte Line

Via Lorenzo il Magnifico, 148
00162 Roma - Tel. (06) 42.70.418