

# I SEGRETI DEL TI-99/4A a cura di Giuseppe Merlina

*Questa volta parleremo del set di istruzioni grafiche implementate del Basic del TI 99.*

*Avendo notato che molte delle lettere giunte in redazione lamentano la mancanza di chiarezza del manuale fornito a corredo della consolle, abbiamo deciso di dedicare alcune puntate di questa rubrica all'esposizione in modo più organico e soprattutto privo di errori (numerose nel testo in questione) delle istruzioni che possono dare problemi di interpretazione. Analizzeremo in questa puntata anche i comandi grafici disponibili nel modulo SSS Extended Basic in modo da darvi gli elementi necessari per farvi una chiara idea sulla opportunità o meno di espandere la vostra consolle con questo linguaggio esteso; tenete presente però che le possibilità aggiuntive non si limitano alla gestione delle capacità grafiche.*

*Prima di cominciare dobbiamo rettificare (in parte) quanto scritto su MC n. 20 a riguardo degli operatori logici OR e AND.*

## OR, AND e NOT

Il TI 99 assegna ad una espressione relazionale falsa lo 0 (e fin qui ci siamo anche se per un errore tipografico nell'articolo suddetto lo 0 è diventato una o), ma -1 ad una vera (e non 1). Alcuni linguaggi Basic utilizzano l'unità positiva come indicatore di condizione verificata, quello implementato nel Texas invece preferisce quella negativa; quindi se provate a digitare PRINT 1 < 2 otterrete il valore -1, se invece scrivete PRINT 1 > 2 otterrete il valore 0.

Il lettore Andrea Capasso di Roma ci ha segnalato ultimamente il modo di aggiungere anche la funzione NOT e pertanto riteniamo opportuno riepilogare brevemente tutta la questione.

Il TI-Basic non permette le operazioni logiche OR, AND e NOT e quindi non è possibile inserire più di un confronto sulla stessa linea di programma con conseguenze appesantimento di quest'ultimo. Se però noi racchiudiamo i singoli confronti tra parentesi, potremo usare gli operatori matematici \*e + al posto di quelli logici AND e OR. Questo perché ad ogni sezione del confronto multiplo (composta da una coppia di parentesi e dalla espressione relazionale in essa contenuta) viene assegnato il valore 0 in caso di esito negativo del confronto e -1 in caso positivo. Esiste però un problema che, come scrive Capasso, potrebbe portare a dei risultati che farebbero rivoltare nella tomba il povero Boole; finché all'interno di un confronto multiplo impieghiamo solo AND (\*) o OR (+) tutto fila liscio, ma quando capita di dover usare i due operatori contemporaneamente esiste la possibilità di ottenere risultati completamente sballati. Ad esempio l'espressione (1 < 10) + (2 < 10) \* (3 < 10) è evidentemente vera, mentre invece il risultato sarà 0 (falso) poiché (-1) + (-1) \* (-1) = (-1) + (+1) = 0; fate quindi attenzione. In realtà esisterebbe più di un modo per portare le cose alla normalità, ma francamente pensiamo che alla fine le complicazioni sarebbero maggiori dei vantaggi e quindi in casi dubbi consigliamo di evitare l'uso del confronto multiplo sulla stessa riga.

Infine, per ottenere la funzione NOT, sarà sufficiente implementarla tramite l'istruzione DEF NOT(X) = X = 0 che, sebbene appaia alquanto bizzarra, in realtà assolve egregiamente il suo compito; notare che il secondo segno di uguaglianza è un operatore relazionale e non di assegnazione.

## Il TI 99 e la grafica del Basic

Il TI 99 divide lo schermo del vostro televisore (o monitor) in 49.152 punti disposti in modo da formare un rettangolo composto da una matrice di 256 x 192. A sua volta questa matrice viene suddivisa in gruppi di 64 Pixel, corrispondenti ad un carattere di 8 x 8 punti; ne consegue il fatto che avremo la possibilità di visualizzare contemporaneamente 768 caratteri disposti in un

reticolato fittizio di 32 colonne per 24 righe. Tali caratteri vengono identificati da un numero compreso tra 0 e 255 corrispondente al codice ASCII (American Standard Code for Interchange Information) ad essi attribuito, ma in pratica quelli realmente utilizzabili sono 128 dal momento che solo i codici da 32 a 159 possono essere definiti dall'utilizzatore.

I codici da 32 a 126 sono quelli standard rappresentanti i numeri, le lettere maiuscole e minuscole nonché i segni speciali; tali caratteri possono solo essere ridefiniti all'interno di un programma tramite l'istruzione CALL CHAR, mentre al suo termine vengono automaticamente riportati al loro Pattern originale.

I codici da 127 a 159 invece possono essere ridefiniti anche direttamente da tastiera e conservano le loro caratteristiche fino a quando non si decide di apportar loro delle variazioni o di spegnere la consolle.

Una informazione che non viene riportata da nessuna parte e che abbiamo scoperto casualmente è quella riguardante i caratteri 128 e 129 (ottenibili tramite CTRL(.) e CTRL(A) rispettivamente); se all'interno del vostro programma è previsto l'impiego del registratore a cassette, evitate accuratamente di usarli dal momento che l'interfaccia registratore li utilizza per il suo funzionamento e che quindi ne cambia i valori ad ogni operazione di I/O da nastro. Per prova accendete il TI 99 (o, se già acceso, resettatelo tramite QUIT) e digitate CTRL(.) e CTRL(A), quindi premete ENTER; sul video apparirà BAD NAME, non fateci caso e battete OLD CS1; in corrispondenza delle posizioni dello schermo nelle quali abbiamo inserito i codici 128 e 129 saranno visualizzati dei simboli strani che se a noi non dicono niente, per l'interfaccia registratore hanno invece un ben preciso significato.

Da notare inoltre che, contrariamente a quanto specificato sul manuale, il codice ASCII 127 (FCTN(V)) non viene resettato a programma ultimato. Come avrete visto è possibile anche ottenere i caratteri 127 e 128 sul video tramite una pressione simultanea del tasto CTRL e FCTN con un altro; ci siamo resi conto che l'argomento "uso della tastiera" è alquanto ingarbugliato sul manuale, per non parlare della istruzione CALL KEY che, a causa delle numerose varianti ottenibili, è forse quella più difficile da interpretare all'interno del TI-Basic; la prossima puntata di "I segreti del TI 99/4A" sarà pertanto dedicata interamente a tale argomento.

## L'istruzione CALL CHAR

Il formato è: CALL CHAR (A,A\$) dove A rappresenta il nome di una variabile contenente un numero compreso tra 32 e 159, oppure direttamente tale valore; A\$ invece è il nome di una variabile di stringa precedentemente definita con una serie di codici esadecimali (16) indicanti, tramite una codificazione, quali punti del carattere devono essere accesi e quali spenti, oppure direttamente tale stringa esadecimale racchiusa tra virgolette.

È il caso di soffermarsi un attimo sul concetto di punto acceso o spento poiché tale definizione è errata da un punto di vista formale. Normalmente indichiamo come acceso un punto luminoso, ossia più chiaro di quelli che gli sono attorno, ma in realtà nel dichiarare la CALL CHAR noi potremo ottenere dei punti accesi che otticamente sono più scuri degli altri dello stesso carattere. Sarà quindi più corretto dire che i Pixel indicati come ON assumeranno il colore di FOREGROUND dell'eventuale CALL COLOR ad essi relativo, mentre quelli OFF il colore di BACKGROUND, tenendo presente che quest'ultimo può ovviamente, essere più chiaro del precedente.

La funzione dell'istruzione CALL CHAR è quella di programmare l'aspetto grafico di un determinato carattere e quindi di una porzione dello schermo composta da 64 punti; volendo variare uno solo di questi Pixel, dovremo obbligatoriamente ridefinire gli altri 63 anche se non avremo apportato loro nessuna modifica.

## Le istruzioni CALL VCHAR e CALL HCHAR

L'istruzione CALL CHAR non provoca nessun effetto immediato sullo schermo in quanto è un comando di assegnazione e non di esecuzione. Per visualizzare il nuovo carattere definito oppure quello assegnato "d'ufficio" dal TI-Basic (lettere, numeri, segni speciali), occorre impiegare l'istruzione CALL VCHAR o la sua equivalente HCHAR che si differenzia dalla prima per il fatto che, in caso di uso del parametro di ripetizione, otterremo un avanzamento in senso orizzontale anziché verticale. Il formato è: CALL VCHAR (A, B, C, D) dove A e B rappresentano rispettivamente le variabili contenenti il numero di riga e di colonna in cui si vuole visualizzare il carattere indicato nella variabile C per un numero di volte pari a D in senso verticale o orizzontale (CALL HCHAR); quest'ultimo è un parametro facoltativo e non è necessario nel caso si voglia mostrare un solo carattere. Ovviamente al posto delle variabili si possono inserire direttamente i valori numerici relativi.

```

10 REM - INDIRIZZAMENTO SINGOLO PUNTO -
20 CALL CLEAR
30 OPTION BASE 1
40 DIM MAT(8,8),SCHERMO$(121),BIN$(16)
50 ESA$="0123456789ABCDEF"
60 FOR J=1 TO 121
70 SCHERMO$(J)="0000000000000000"
80 CALL CHAR(J+32,"0")
90 NEXT J
100 K=33
110 FOR R=8 TO 18
120 FOR C=11 TO 21
130 CALL HCHAR(R,C,K,1)
140 K=K+1
150 NEXT C
160 NEXT R
170 FOR J=1 TO 16
180 READ BIN$(J)
190 NEXT J
200 DATA 0000,0001,0010,0011,0100,0101,0110,0111
210 DATA 1000,1001,1010,1011,1100,1101,1110,1111
220 X=1
230 Y=1
240 C=INT((X+7)/8)
250 IX=X-((C-1)*8)
260 R=INT((Y+7)/8)
270 IY=Y-((R-1)*8)
280 K=32+C+11*(R-1)
290 FOR J=1 TO 16
300 L=POS(ESA$,SEG$(SCHERMO$(K-32),J,1),1)
310 FOR M=1 TO 4
320 E=INT((J+1)/2)
330 F=M+4*(J-(2*INT((J-1)/2))-1)
340 MAT(E,F)=VAL(SEG$(BIN$(L),M,1))
350 NEXT M
360 NEXT J
370 MAT(IY,IX)=1
380 A$=""
390 FOR RI=1 TO 8
400 FOR CO=1 TO 5 STEP 4
410 A$=A$&SEG$(ESA$, (MAT(RI,CO)*8+MAT(RI,CO+1)
*4+MAT(RI,CO+2)*2+MAT(RI,CO+3)+1),1)
420 NEXT CO
430 NEXT RI
440 SCHERMO$(K-32)=A$
450 CALL CHAR(K,A$)
460 X=X+1
470 Y=Y+1
480 IF X>88 THEN 500
490 GOTO 240
500 END

```

A pagina 77 del manuale in nostro possesso (non sappiamo però se ne siano state fatte ristampe più aggiornate) si dichiara che: "Poiché le colonne 1, 2, 31 e 32 non si vedono sul video, potete usare solo numeri di colonna da 3 a 30", il che non corrisponde, come è facilmente verificabile, a verità dal momento che sono perfettamente visualizzate tutte le 32 colonne. Per coloro che hanno notato uno spostamento a sinistra del quadro (presente anche nell'utilizzazione di un monitor) diremo che invece non c'è nulla da fare a causa dei diversi sincronismi video americani ed europei e che non è colpa di una cattiva taratura del modulatore Pal; sconsigliamo vivamente di tentare di portare le cose alla normalità operando sui comandi del TV: otterreste solo una deformazione dell'immagine.

## L'istruzione CALL GCHAR

Il formato è: CALL GCHAR (A, B, C) dove A e B sono le variabili destinate a contenere i valori delle coordinate (riga e colonna) dello schermo nella cui posizione si vuole sapere quale carattere sia attualmente visualizzato; nella variabile C infatti avremo il codice del carattere ASCII corrispondente alla riga A, colonna B.

Alcuni lettori ci hanno scritto chiedendo quale possa essere la reale utilità di questa istruzione; la risposta è semplice: enorme, specialmente nei programmi di giochi in cui spesso occorre poter controllare la presenza o meno di un determinato carattere (per esempio quello rappresentante un proiettile) in una determinata posizione dello schermo (per es. il bersaglio). Inoltre, se essa non fosse stata presente, non sarebbe stato possibile realizzare la subroutine ACCEPT AT (vedi MC n. 20) e neanche il programma presentato questo mese nella rubrica software TI 99/4A se non complicando notevolmente le cose. In effetti dovete immaginare lo schermo come una matrice bidimensionale di  $24 \times 32$  elementi, ognuno dei quali contiene il codice del carattere rappresentato nella posizione corrispondente; l'istruzione CALL GCHAR vi permette di conoscere tale codice in ogni momento del programma.

## Le istruzioni CALL COLOR e CALL SCREEN

Nel TI 99 abbiamo a disposizione 16 colori diversi con i quali è possibile ottenere numerose combinazioni di accoppiamento tra disegni e sfondi. L'istruzione CALL SCREEN, seguita da un numero da 1 a 16 posto tra parentesi, permette di specificare quale deve essere il colore (indicato da tale numero) assegnato allo schermo nei punti in cui esso risulta vuoto, ossia dove non è presente nessun carattere. Normalmente il video assume il colore viola (codice 8), mentre durante l'esecuzione di un programma prende il verde chiaro (codice 4).

Il formato dell'istruzione CALL COLOR invece è: CALL COLOR (A,B,C), dove A rappresenta il numero dell'insieme di caratteri da ridefinire dal punto di vista cromatico con la tinta di FOREGROUND indicata da B e di BACKGROUND indicata da C.

I codici ASCII sono divisi in 16 gruppi di 8 caratteri ciascuno, perciò se vorremo cambiare l'assegnazione dei colori di un determinato carattere, automaticamente dovremo influenzare nello stesso modo gli altri appartenenti allo stesso insieme.

## Indirizzamento del singolo punto

Come vi abbiamo preannunciato la volta scorsa, esiste il modo di indirizzare il singolo Pixel sul video tramite il TI-Basic. Vi presentiamo pertanto il listato di un piccolo programma dimostrativo, precisando però che il suo interesse è, a nostro avviso, più accademico che pratico dal momento che il gran numero di

passaggi e di calcoli necessari per il suo funzionamento porta ad un allungamento dei tempi di tracciatura a livelli a dir poco esasperanti. Il principio di funzionamento del programma presentato è alquanto contorto, proveremo comunque a spiegarlo nel miglior modo possibile; non è escluso inoltre che è presumibilmente possibile ottimizzarlo. Se qualcuno di voi lettori è in grado di farlo ci tenga al corrente, tenendo presente però che devono essere soddisfatti i seguenti requisiti:

- 1) I punti da accendere devono essere individuati da coordinate (X,Y).
- 2) I punti devono essere accessibili senza soluzione di continuità, ossia uno di seguito all'altro, senza salti o interruzioni.
- 3) Il plottaggio di un punto all'interno di un carattere non deve distruggere o falsare l'informazione che vi era precedentemente memorizzata.

Esaminiamo il modo in cui noi abbiamo risolto il problema, anche se inevitabilmente con alcune limitazioni pratiche.

Dal momento che in TI-Basic si hanno a disposizione un massimo di 128 codici ridefinibili, non è stato possibile rendere grafico tutto lo schermo, ma solo una sua parte costituita da 121 caratteri disposti in 11 righe e in 11 colonne in modo da formare un quadrato. Le linee da 100 a 160 si occupano di visualizzare sullo schermo a partire dalla riga 8, colonna 11 i codici da 33 a 153 disposti in righe consecutive. All'inizio del programma tali codici vengono definiti come vuoti dalla linea 80 in modo da renderli disponibili per costituire l'area grafica ad alta risoluzione. La riga 40 dimensiona una matrice a due vettori; la prima costituisce una zona di lavoro nella quale ogni carattere da modificare verrà prima trasferito e quindi variato in base alle coordinate del punto da accendere; il vettore alfanumerico SCHERMO\$ contiene invece le stringhe esadecimali di definizione della forma di ogni carattere nella posizione corrispondente dello schermo, mentre BIN\$ i 16 valori binari corrispondenti ai codici esadecimali da 0 a F occorrenti per le operazioni di conversione della matrice dei caratteri. I valori delle coordinate X e Y possono variare da 1 a 88, e l'origine (X = 1, Y = 1) è disposta nell'angolo in alto a sinistra del campo grafico. Vediamo in pratica che cosa accade nel caso in cui, per esempio, si voglia accendere il punto di coordinate X = 30, Y = 15.

La linea 240 trasferisce in C il numero della colonna corrispondente alla coordinata della X, la 250 in IX il valore orizzontale del punto all'interno della matrice del carattere; le linee 260 e 270 svolgono la stessa funzione, ma in senso verticale; la linea 280 assegna a K il codice ASCII del carattere relativo alla posizione trovata; nel nostro caso avremo C = 4, IX = 6, R = 2, IX = 7, K = 47. Le righe da 290 a 360 caricano nella matrice MAT i codici binari corrispondenti alla stringa di definizioni esadecimale della forma del carattere all'interno del quale viene a trovarsi il punto da accendere; questa procedura si rende necessaria per salvare l'eventuale contenuto del codice senza distruggere l'informazione precedente, ma aggiungendovi solo il segno 1 (Pixel ON) nelle coordinate IY, IX (linea 370).

In seguito (righe 380-450) il contenuto della matrice di transito così modificata viene riconvertita nella stringa esadecimale di definizione della forma del carattere; tale stringa viene salvata nel vettore SCHERMO\$ per essere disponibile come base di partenza per una eventuale nuova modifica e assegnata al carattere K che sarà visualizzato sullo schermo nella sua nuova forma (450). Da notare che non si è reso necessario l'uso dell'istruzione CALL HCHAR, poiché i codici sono già visibili sul video e cambiano aspetto automaticamente ad ogni nuova assegnazione.

Il programma presentato, a titolo dimostrativo, disegna una linea obliqua composta da 88 punti e con origine nel punto (1,1) del quadrato grafico. Le istruzioni da modificare per ottenere

altri tracciati sono: 220 - 230 - 460 - 470 - 480 - 490.

Chi volesse implementare tale routine usando l'Extended Basic, può semplificarla notevolmente poiché usando l'istruzione CALL CHARPAT (vedremo dopo di che si tratta), il vettore SCHERMO\$ e le operazioni ad esso relative non sono più necessari.

## Le istruzioni grafiche dell'Extended Basic

Il Basic Esteso contiene un set di istruzioni grafiche notevolmente più potente del TI-Basic; sebbene continui a perdurare (per motivi più che altro hardware accennati nella precedente puntata) l'impossibilità di indirizzare il singolo Pixel, le capacità aggiuntive nel campo della grafica per i giochi sono notevoli.

Tutte le istruzioni già viste nel TI-Basic sono presenti nell'Extended, l'unica variante è che i codici dei caratteri ridefinibili vanno da 32 a 143, ossia se ne hanno a disposizione 16 di meno.

L'istruzione CALL CHARPAT (di cui abbiamo accennato prima) permette di ottenere immediatamente la stringa esadecimale di definizione del singolo carattere; ecco perché, nel programma di indirizzamento del singolo punto, non si rende più necessaria la sua memorizzazione all'interno del vettore SCHERMO\$.

Il grosso vantaggio dell'Extended è di poter impiegare fino a 28 Sprite in movimento continuo sullo schermo e di poterne definire: la forma, il colore, la grandezza (singola o doppia), la direzione e la velocità del movimento. Per chi non lo sapesse, uno Sprite è un carattere grafico che si muove sullo schermo in modo continuo con direzione e velocità variabili a seconda dei suoi parametri di definizione; può assumere uno qualsiasi dei 16 colori disponibili ed è completamente autonomo, una volta creato, rispetto all'esecuzione del programma.

L'istruzione CALL COINC permette di rilevare la sovrapposizione sullo schermo di due Sprite entro una zona di tolleranza specificata da un apposito parametro oppure la coincidenza di uno Sprite con un determinato punto del video. Per mezzo dell'istruzione CALL DISTANCE invece si ottiene la distanza (elevata al quadrato) che separa uno Sprite da un altro o da un punto dello schermo. Il posizionamento degli Sprite avviene al momento della loro creazione con la CALL SPRITE tramite i parametri di riga e di colonna (da 1 a 192 e da 1 a 256) oppure successivamente tramite la CALL LOCATE; la loro posizione attuale può essere rilevata in ogni momento durante l'esecuzione del programma tramite la CALL POSITION.

Per concludere questa rapida carrellata sulle istruzioni grafiche potenziate del Basic Esteso parleremo della CALL MAGNIFY che, come intuibile dal nome, permette, tramite le sue quattro opzioni possibili, di ottenere: 1) lo Sprite nella sua rappresentazione normale di un carattere; 2) lo Sprite di dimensioni doppie, ossia lo stesso Pattern occupante quattro caratteri posti a quadrato anziché uno; 3) lo Sprite di quattro caratteri, ma ognuno definito in modo diverso; 4) come l'opzione 3 ma di dimensioni doppie, ossia lo Sprite occupa 16 caratteri disposti  $4 \times 4$ .

Bisogna rilevare che, per mezzo di tali istruzioni e con un po' di bravura è possibile creare dei programmi di giochi che ben poco hanno da invidiare a quelli commerciali pur senza andarsi ad impelagare nelle difficoltà del linguaggio Assembler. A questo proposito siamo incerti se sia il caso o meno di pubblicare (nella rubrica di software) dei programmi in Extended poiché non vorremmo che, per accontentare i possessori di tale linguaggio, scontentassimo invece chi ne è privo. Gradiremmo pertanto che foste voi, tramite le vostre richieste ed i vostri contributi, ad indicarci la strada da seguire.

# I NTERNATIONAL C OMPUTER S YSTEMS

Uffici di Roma: Via della Balduina, 85-89 - Tel. 34.81.85 - 34.92.760-660 - Telex 611091 CRM C Stabilimento: Via Nettunense, 49 - 00042 Anzio - Tel. 98.46.206

In Italia come in tutto il mondo la gamma dei nostri elaboratori sta ricevendo l'adesione degli esperti di informatica e degli utilizzatori. Per ragioni che sono le più valide: rigore tecnologico, fabbricazione professionale e sforzo costante di creare degli autentici sistemi di informatica al costo più basso. La International Computer Systems garantisce la distribuzione dei prodotti migliori direttamente dagli stabilimenti produttivi situati in Giappone, Irlanda, Italia.

## M23 mark III - M23 mark V

**Piccolo. Leggero. Potente.  
Si impara a programmarlo in tre giorni!**

Configurazioni a scelta con floppy da 5 o da 8 pollici monitor a fosfori verdi o a colori (RGB) da 14 pollici.  
Scheda grafica a colori opzionale.

### Unità centrale

Un microprocessore ZILOG Z 80A con un clock a 4 MHz gestisce le risorse del sistema.

Un 2° micro APU effettua tutti i calcoli matematici.

Una memoria RAM da 128 Kbytes è a disposizione utente.

Due interfacce seriali RS232 programmabili e un'interfaccia parallela permettono il collegamento con l'esterno.

Questo insieme dà all'unità centrale la potenza richiesta per una larga gamma di applicazioni.

### Unità minifloppy

Due minifloppy da 5" (328 Kbytes ciascuno), semplice faccia, doppia densità, gestiti da un'interfaccia interna DMA (accesso diretto memoria).

### Unità floppy 8"

Due Driver doppia faccia, doppia densità (1,1 MB ciascuno), con possibilità di formattazione in tutti i formati IBM.

### Tastiera

Un blocco alfanumerico standard con maiuscole e minuscole.

Un blocco numerico separato con i comandi del cursore.

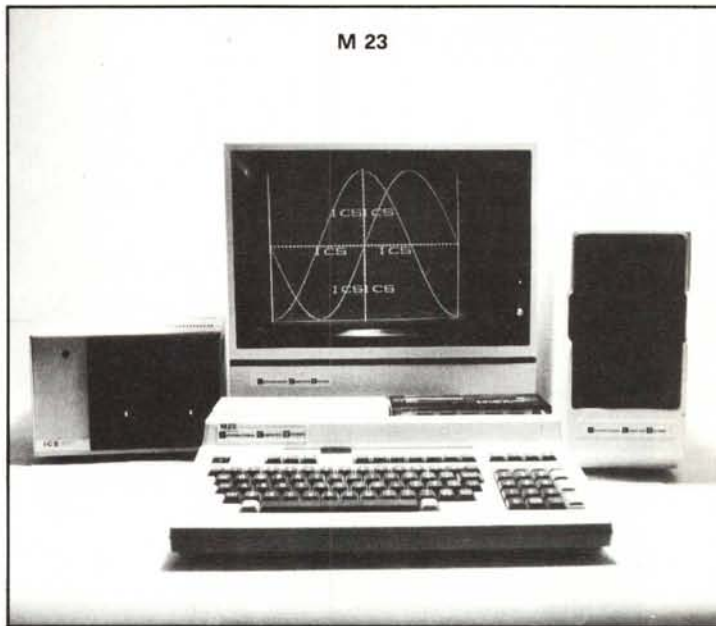
Un blocco di 14 funzioni programmabili.

Le sue numerose funzioni permettono una grande flessibilità di utilizzo.

### Schermo

25 righe per 80 colonne maiuscole e minuscole in visione normale o "negativa".

32 caratteri semigrafici permettono la costruzione di tabelle o di grafici.



## SYSTEM SOFTWARE

● Relocatable assembler ● Editor ● Debugger ● Relocatable loader ● Library file editor

● Subroutines in Assembler possono essere richiamate all'interno di programmi in BASIC o in Fortran ● EBASIC - Interprete esteso occupa circa 32 Kbytes ● CBASIC - Compilatore compatibile con Ebasic consente di aumentare di 5/6 volte la velocità di esecuzione ● MBASIC - A doppia precisione (13 cifre) per calcoli tecnici e matriciali ● TBASIC - Per trasmissione dati e collegamento con altri computers ● FORTRAN IV - Per calcoli tecnico-scientifici ● COBOL - Corrispondente a livello ANSI 74 ● UCSD PASCAL ● L'SGL è un linguaggio grafico che permette, eventualmente anche con monitor a colori, di eseguire disegni estremamente complessi utilizzando la libreria BASIC con delle subroutines per le funzioni più comuni.

Vasta scelta di software applicativo gestionale-scientifico

PIPS, un linguaggio facile da imparare, sfrutta al massimo le capacità della macchina

Il PIPS, software unico, sviluppato per uso gestionale, è molto più vicino alla mente umana dell'Assembler, del Fortran, del Basic. Il PIPS permette a tutti di usare un potente computer con facilità. Il PIPS lavora utilizzando oltre 100 comandi. La gestione dei dati avviene tramite la semplice selezione di questi comandi. Per ricercare dei dati si imposta il comando CS. Per sortare si imposta SORT. Per funzioni grafiche si imposta GR. E così via. Vari programmi e funzioni possono essere ottenute a seconda dell'ordine con cui si selezionano i comandi. Il PIPS elimina la necessità di programmi specializzati. Alcuni tipi di lavoro richiedono soltanto di digitare i comandi nel loro ordine, per ottenere i risultati richiesti!

## M 243 - M 343 Una famiglia di micro da 8 e da 16 bit multiutente con multiprogrammazione

L'M 243 e l'M 343 sono il culmine di anni di esperienza combinati con la più sofisticata tecnologia. Sono microcomputers completamente nuovi che si adattano perfettamente ai più disparati tipi di applicazioni. Offrono possibilità di ampliamento in memoria centrale con schede; in memoria di massa con dischi floppy da 5" e da 8" e dischi rigidi Winchester. Oltre ad avere inserite interfacce di qualsiasi tipo e a poter essere utilizzati come terminali intelligenti di computers più potenti, sono dotati di uno schermo completamente grafico ad altissima definizione anche a colori e permettono la gestione di più posti dilavoro in multi-programmazione.

### Unità Centrale

Un microprocessore a 8 bit Z80A gestisce le risorse del sistema nel M 243.

Un microprocessore a 16 bit 8086 è invece utilizzato nel modello M 343.

Un 2° processore logico effettua tutte le operazioni logiche sui numeri fino a 32 bit in virgole flottanti.

Un counter/timer programmabile da software controlla la successione delle operazioni.

Un orologio in tempo reale, con batteria tampone, fornisce la data e l'ora e permette di avviare, tra l'altro, dei programmi ad ore prestabilite.

Una memoria RAM da 192 Kbytes a 1 Mbytes è a disposizione utente. Tale memoria consente la presenza di più posti lavorocompleti in multiprogrammazione.

Quattro canali seriali RS232 programmabili da 50 a 19.200 Baud e un canale parallelo permettono il collegamento con l'esterno.



## M5 - Home Computer Il micro più piccolo della nostra famiglia

Si collega al televisore a colori di casa ed ad un registratore a cassette

### Unità centrale

Z 80A - RAM 4 k + 16 k video RAM espandibile con cassetta fino ad altri 32 k.

Uscita per stampante parallela.

Uscita per TV color.

Uscita per monitor e altoparlante.

Optional n. 2 Joypads per video game.

Tastiera con 52 tasti a 4 funzioni (maiuscoli, minuscoli, istruzioni basic e semigrafica).

Cassetta elettroniche con basic, pips e vasta scelta di video games.

Visitateci al prossimo SMAU (Milano)

Pad. 12 - Stand A18



## INSTALLAZIONE IN TUTTA ITALIA CON LE SEGUENTI PROCEDURE

- Contabilità generale magazzino fatturazione.
- Contabilità generale e semplificata per commercialisti.
- Contabilità generale a booking per Agenzie di Viaggi.
- Trattamento testi e mailing list merge universale.
- Contabilità finanziaria per scuole ed enti pubblici.
- Paghe e stipendi per scuole.
- Gestione magazzini componenti o ricambi.
- Gestione biblioteche.
- Gestione iscritti ordini professionali.
- Calcolo strutture per zone sismiche.
- Gestione laboratori di analisi cliniche.

STAMPANTI 80-132-220 COLONNE ANCHE GRAFICHE A MATRICE DI 9 AGHI ED A MARGHERITA.

PLOTTER A 8 COLORI.  
CONVERTITORI ANALOGICI/DIGITALI E D/A.

Cercansi distributori per zone libere



# A CIASCUNO IL SUO

**HP85** PERSONAL COMPUTER INTEGRATO PORTATILE  
PER APPLICAZIONI TECNICO SCIENTIFICHE

**HP86** COMPUTER MODULARE AD ALTE PRESTAZIONI  
PER APPL. SCIENTIFICHE E GESTIONALI

**HP9816** PERSONAL COMPUTER A 16/32 BIT PER APPL.  
AD ALTA VELOCITA' DI ELABORAZIONE

**HP120/  
125** COMPUTER PER UFFICIO PER APPLICAZIONI  
MANAGERIALI E DI WORD PROCESSING

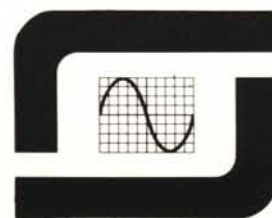


Ogni applicazione richiede prestazioni diverse. L'esperienza SILVERSTAR vi aiuta a scegliere la soluzione più adatta alle vostre esigenze, abbinando l'alta qualità dei computer HP ad una serie completa di programmi applicativi.

	HP 85	HP 86	HP 125	HP 9816
Mem. RAM fino a	32 K	576 K	64 K	768 K
Sistema operat./Linguaggi	Basic HP	Basic HP CPM/Pascal	CPM	Basic/Pascal HPL
Video	alfanumer./grafico	alfanum./grafico (esterno)	alfanumerico	alfanumer./grafico
Interfacce interne	—	Parall. Centronics	HP IB+2 RS232C	HP IB, RS232
Periferiche interne	Printer e cart. magnetica	—	—	—

*Se siete interessati ai personal computer HP compilate e spediteci questo tagliando.*

Cognome .....  
 Nome .....  
 Qualifica ..... Azienda .....  
 Via .....  
 CAP ..... Città .....  
 Telefono .....



**silverstar**  
componenti e sistemi

Sede: 20146 Milano - Via dei Gracchi, 20 - Tel. (02) 4996 (12 linee) - Telex 332189  
 40122 Bologna - Via del Porto, 30 - Tel. (051) 522231  
 00198 Roma - Via Paisiello, 30 - Tel. (06) 8448841 (5 linee) - Telex 610511  
 10139 Torino - P.za Adriano, 9 - Tel. (011) 443275/6 - 442321 - Telex 220181