

SPRING
SOFTWARE

MSX

COMPUTER MAGAZINE

N.7 MAR/APR 1986

Sped in abb. post. Gr. III L. 9.000

10

PROGRAMMI
SU CASSETTA

LINGUAGGIO
MACCHINA
SPECIALE
SUPER
VIDEO



MSX Computer Magazine è edita da Arcadia srl,
C.so Vitt. Emanuele 15, Milano.
Tel. 02/706329 (solo giovedì h. 15-18).
Una copia L. 9.000.
Fotocomposizione: Composit.
Stampa: Garzanti,
Milano. Distribuzione: SO.DI.P. Angelo
Patuzzi srl, Via Zuretti 25, Milano.
Registrato Trib. Milano N. 52 del 2/2/85.
Resp. Sira Rocchi.
Sped. in abb. post. Gr. III/70.
MSX is a trademark of MicroSoft Co.
Manoscritti, disegni, fotografie
e programmi inviati non si
restituiscono anche se non pubblicati.

IN QUESTO NUMERO

GLI ARTICOLI

- MSX TAPE SOFT
- SUPER SCREEN 2
- LINGUAGGIO MACCHINA

- SUPER TRIS PROGRAMMI
- TURBO DRAW
- NOVITÀ MSX 2

I PROGRAMMI

- BOMBER
- CYCLE
- NAVY
- MUSICOMPOSER
- LEOPARD II

- NETWORK
- BIS
- HALLEY
- DUPLICATORE
- BOMB SHELL

MSX TAPE SOFT

I programmi che vi presentiamo in questo settimo numero di MSX Computer Magazine sono tutti compatibili con qualsiasi sistema MSX. Ecco per voi ben 10 programmi.

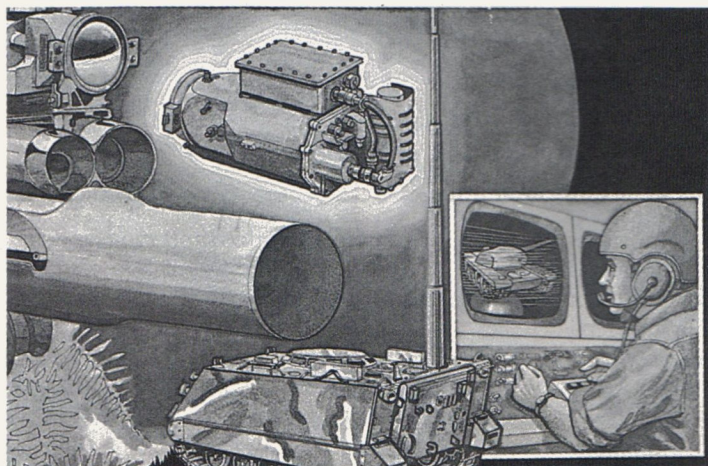
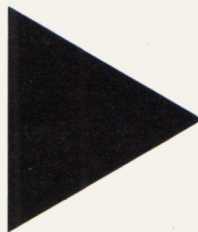
Ricordate di collegare la spina del controllo motore alla presa REM del vostro registratore, se quest'ultimo la



possiede. Assicuratevi che la spina rossa sia collegata alla presa MIC del registratore e la spina nera alla presa EAR. Se il vostro mangiacassette non possiede la presa REM, fate particolare attenzione a quando un programma è stato caricato o deve essere caricato, affinché il nastro scorra per il giusto tempo. Appena vedete appa-

1

BOMBER
("BOMBER")
di M. Belardi



Vi trovate in una base militare ed il vostro compito è quello di difendere dall'assalto di aerei suicida, bombe, missili e carri armati, 15 radar con l'aiuto di un cannone a due canne che non può essere colpito.

Il gioco termina quando tutti i radar sono stati distrutti.

Gli attacchi ai radar vengono effettuati casualmente: da destra e da sinistra per quanto riguarda le bombe d'aereo e i carri armati; da dietro le montagne per i missili e dall'alto per l'aereo suicida. L'arrivo di quest'ultimo, dato che cade in picchiata molto velocemente,

è preannunciato dal rumore del suo motore.

Si vince un radar ogni 500 punti: i vari punteggi sono indicati nel programma. Il punteggio record viene aggiornato in tempo reale durante il gioco.

Vi è una routine in linguaggio macchina per controllare lo spostamento del cannone, lo sparo dei proiettili, il suono ed il cambio degli sprite dei proiettili e del cannone per simulare l'allontanarsi dei colpi ed il rinculo del cannone.

Il programma è lungo circa 6 Kbyte; per caricarlo digitare: CLOAD "BOMBER".

rire sul video, dopo un comando di caricamento, la scritta OK, spegnete il registratore.

ATTENZIONE: la presentazione va caricata con il comando: BLOAD "MSXPR",R per gli altri programmi, invece, segui-



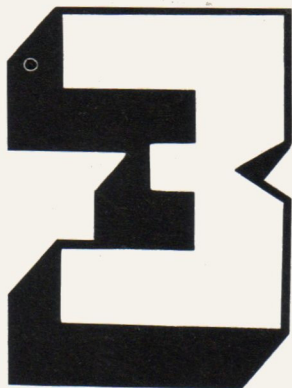
re attentamente le istruzioni riportate nei riquadri.

Nella cassetta allegata a questo fascicolo troverete sul lato A:

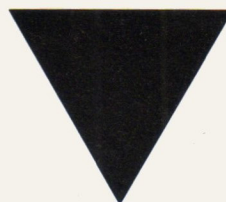
Presentazione, Bomber, Cycle, Navy, Music composer, Leopard II.

Sul lato B:

Network, Bis, Halley, Duplicatore, Bomb Shell.



NAVY
("NAVY")



di E. Dassi

La classica battaglia navale giocata, anziché sulla carta, direttamente sullo schermo contro il computer.

Dopo aver inserito il tempo di pausa tra il vostro tiro e quello del computer, disponete i sei pezzi a vostra disposizione: un incrociatore pesante, tre incrociatori leggeri e due sottomarini con i quali dovrete distruggere la flotta navale dell'MSX.

Posizionate sulla mappa le navi ed i sottomarini indicando per prima la

coordinata orizzontale, quindi quella verticale. Successivamente sparate i vostri colpi. Per vedere le condizioni della vostra flotta premere il tasto SPACE.

Il programma è ampiamente commentato in ogni sua procedura; per modificare il disegno delle navi e dei sottomarini osservare le linee 32100-32440.

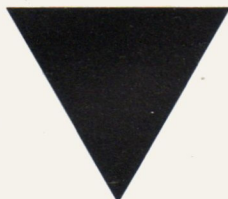
Per caricare il programma digitare:

CLOAD "NAVY"



CYCLE
("CYCLE")

di R. La Rocca



In sella ad una veloce bicicletta rossa state partecipando ad una gara ciclistica. La corsa richiede diverse attenzioni: mantenere una certa velocità e non uscire di strada.

In alto sullo schermo vi sono cinque indicatori di marcia: quello della velocità, quello della pendenza del terreno, quello dell'energia, quello della distanza percorsa e infine quello del tempo rimanente.

La velocità di marcia può essere aumentata premendo il tasto del cursore-destra oppure diminuita usando il tasto del cursore-sinistra. Fate attenzione a non superare i 40 km/h oppure a scendere sotto ai 10 km/h pena la fine della gara. Allo stesso modo tenete sotto controllo l'indicatore dell'energia perché se dovesse finire dovrete abbandonare la gara. L'energia cala con il passare del tempo, con il contatto con i

bordi della strada, quando aumenta la velocità ed infine in presenza di tracciati in salita. La gara può interrompersi anche per lo scadere del tempo.

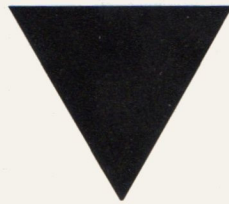
Ovviamente se terrete conto di tutte queste cose non è detto che vincerete la corsa perché dovrete sempre vedervela con gli altri tre concorrenti. Prima d'iniziare la «pedalata» scegliete il percorso da farsi premendo il tasto corrispondente.

Il programma sfrutta diverse routines in linguaggio macchina: lo scroll del video, il controllo della tastiera, il movimento degli sprites, ecc. Per chi volesse velocizzare maggiormente il gioco basta variare i valori della POKE 59999 mentre per velocizzare lo scroll basta cambiare il valore dell'istruzione ON INTERVAL GOSUB.

Per caricare il gioco digitare: CLOAD "CYCLE"

4

MUSIC COMPOSER ("MUSCOM")



di T. Leddi



Volete sentire suonare il vostro MSX come un vero e proprio musicista? Semplice, procuratevi uno spartito musicale e trascrivete le note così come le vedete disegnate sulla carta anche sul computer ed il gioco è fatto!

Questo programma permette di ascoltare fino a 5 pagine di musica a 3 voci. Per scrivere le note basta trovare la posizione corretta muovendo il cursore con i tasti cursore e poi premere la barra spaziatrice. Premendo il tasto «Z» si disegna, nella posizione puntata dal cursore, una pausa.

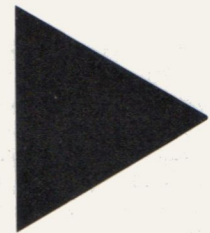
Non è obbligatorio usare tutte e tre le voci, però per un corretto uso bisogna, nelle voci utilizzate, non lasciare nessun spazio vuoto (cioè senza nota o pausa) tra l'inizio del pezzo e l'ultima nota scritta. Per muoversi alla pagina successiva bisogna premere il tasto SPACE quando il cursore si trova tutto a destra. Per visualizzare una pagina qualsiasi si deve portare il cursore tutto a sinistra, premere il tasto SPACE ed inserire il numero della pagina da visualizzare. Con la stessa tecnica portando il cursore nelle due zone colorate e poi usando la barra spaziatrice è possibile in quella di sinistra ascoltare la musica ed in quella di destra cancellare ogni cosa scritta per ricominciare da capo. Si possono scrivere solo note da 1/4, però mettendo vicine delle note di uguale altezza queste vengono «legate». Si ottengono così durate da 1/2, 3/4, 1/1, ecc.

Per chi volesse conservare o utilizzare le musiche per un proprio programma, si può richiedere il contenuto delle stringhe delle tre voci interrompendo il programma (CTRL+STOP) e digitando: PRINT P1\$,P2\$,P3\$. Per variare il tempo d'esecuzione della musica è sufficiente sostituire ai 255 presenti alla linea 7030 il valore desiderato.

Per caricare il programma digitare: CLOAD "MUSCOM"

5

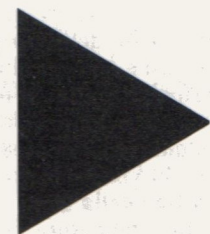
LEOPARD II ("LEOPAR")



di D. Montresor

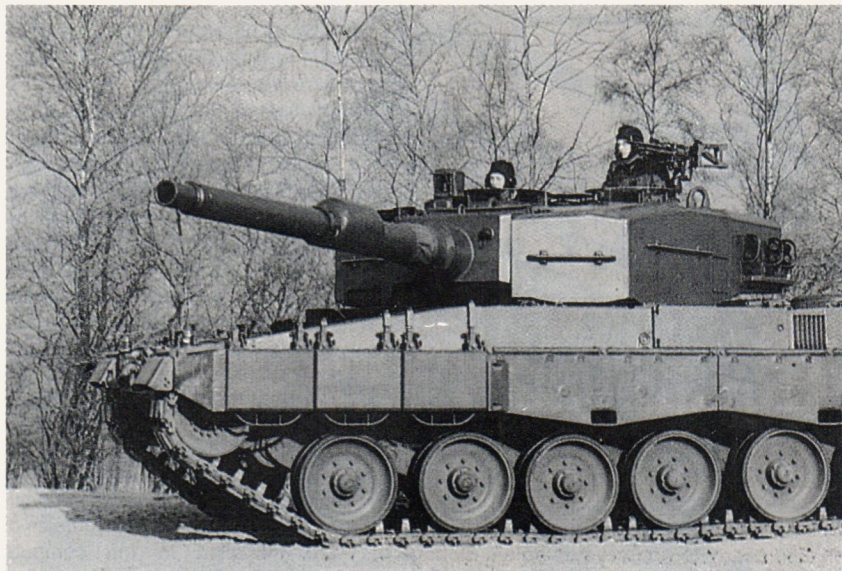
6

NETWORK ("NETWOR")



di N. Paggini

Il compito che ci siamo prefissi col nostro anfibio è quello di distruggere i reattori di diverse centrali nucleari, per far questo dobbiamo attraversare un territorio paludoso cosparsi di muri anticarro e campi minati. I muri possono essere abbattuti a cannonate mentre le mine ovviamente vanno evitate. A sinistra dello schermo sono visualizzate le riserve a nostra disposizione (anfibi di scorta), le munizioni ed il carburante disponibili, lungo il tragitto sono disseminati depositi che saranno utili in caso che le scorte siano state consumate (basta passarci sopra). In acqua si consuma il doppio della benzina rispetto alla marcia su terreno, inoltre ogni volta che ci si ferma a sparare si consuma anche in questo caso benzina per ripartire. La parte sensibile dell'anfibio è quella anteriore quindi con un po' di pratica si riescono ad evitare gli ostacoli muovendosi in diagonale. Distrutto il nucleo del reattore, evitando le colonne di energia ed i colpi che piovono dall'alto, si ha una riserva in più, un bonus di 2000 punti e si prosegue con la differenza che i campi minati iniziano subito all'inizio del per-



corso. Le mine d'acqua iniziano a 4000 punti (linea programma 3560) mentre quelle di terra a 8000 punti (linea 3580). Volendo aumentare i leopard di riserva modificare la linea 140.

Il gioco è salvato in tre blocchi: il primo è in Basic ("LEOPAR"), il se-

condo e il terzo in codice macchina ("CARATT", "PERCOR"). Per caricare digitare CLOAD "LEOPAR", appena appare la scritta "OK" fermare il registratore, digitare RUN+ENTER quindi far ripartire il registratore per il caricamento automatico del secondo e terzo blocco.

Network è la versione del famoso Monopoli trasportata nel mondo della televisione. Si può giocare da 2 a 8 giocatori ed ognuno rappresenta un canale televisivo. Lo scopo del gioco è fare in modo di guadagnare il più possibile facendo capitare i giocatori avversari sulle proprie caselle (trasmissioni). Il percorso è di 40 caselle al termine delle quali ogni giocatore riceverà un bonus di 200 milioni. Quando un giocatore acquista tutte le trasmissioni di una determinata fascia (ad esempio: FASCIA SERALE, FILM, TRASMISSIONI SPORTIVE, rappresentate da colori diversi) può allora mettere degli SPOT (fino ad un massimo di 5) su ogni trasmissione che faranno naturalmente aumentare il pedaggio che gli avversari dovranno pagare. Esistono poi accordi con reti estere (le STAZIONI nel Monopoli) e noleggio di satelliti e stazioni riceventi (ENERGIA ELETTRICA e ACQUA POTABILE nel Monopoli). Ogni giocatore all'inizio del suo turno lancia i dadi premendo la barra spaziatrice (l'uso della barra spaziatrice consente di proseguire col programma dopo che si è finito di guardare una schermata). Se egli capitasse in una casella vuota, può acquistar-

la pagando l'importo corrispettivo. Se non volesse acquistarla allora si fa un'asta tra gli altri giocatori e la casella viene venduta al miglior offerente. Se un giocatore capita su una casella di un altro egli paga una somma proporzionale al numero degli spots sulla casella. Se invece un giocatore capita su una propria casella può:

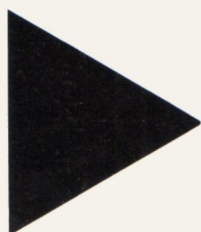
comprare degli spots (solo se possiede tutto il gruppo di trasmissioni), oppure vendere ad un altro giocatore o sospendere la trasmissione ricevendo dal banco del denaro ma perdendo la proprietà della trasmissione.

Per caricare Network digitare CLOAD "NETWOR".





BIS
("BIS")



di P. Gazzarri

Ecce un programma per due giocatori che hanno voglia di misurare la loro memoria.

Chi possiede il turno di gioco, sceglie una coppia fra le 35 del tabellone. In due tentativi bisogna inserire i numeri delle carte da scoprire e trovare due simboli uguali vincendo così un punto e mantenendo il turno; in caso contrario le figure delle caselle verranno coperte e il turno passerà all'avversario. Se verrà scoperta la figura con la lettera «J» (Jolly) si vin-

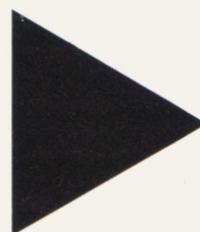


ceranno 5 punti ed una coppia di simboli uguali, mantenendo sempre la mano. In alto sullo schermo appariranno i nomi dei giocatori ed i loro punteggi. Il giocatore che possiede il turno ha il nome sottolineato.

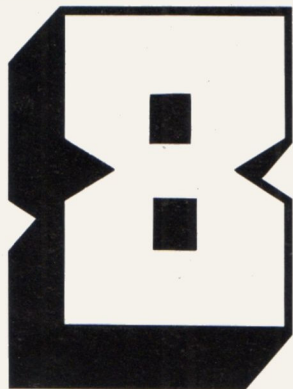
La grafica del programma è stata realizzata ridefinendo i caratteri in SCREEN 1 ed assegnando nuovi colori. Per caricare il gioco digitare CLOAD "BIS".



DUPLICATORE
("DUPLI")



di A. Tarricone



HALLEY
("HALLEY")

di M. Orfei

Dal giorno del suo primo avvistamento, avvenuto il 16 ottobre 1982 con il grande telescopio da 5 metri di Monte Palomar, la cometa di Halley ha percorso ben un miliardo e cinquecento milioni di km, aumentando di oltre due milioni di volte la propria luminosità apparente. In questo periodo la cometa è facilmente avvistabile, anche con un normale binocolo «10x50», però bisogna saperla individuare. Questo programma vi

aiuterà certamente nella vostra «caccia» alla stella di Halley. È possibile avere visualizzata sullo schermo l'intera traiettoria planetaria della cometa, quella ravvicinata, la sua posizione giorno per giorno e così via.

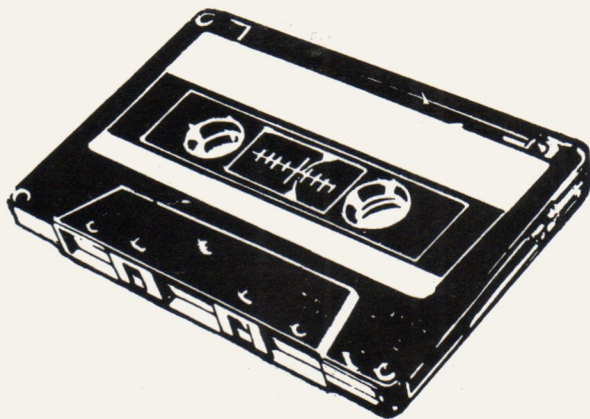
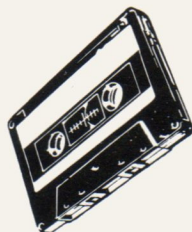
Per scegliere l'opzione basta premere il tasto corrispondente e... guardare un pezzo di cielo nel proprio televisore!

Per caricare il programma digitare: CLOAD "HALLEY"



Quando vogliamo duplicare un programma Basic su un'altra cassetta le operazioni da farsi sono molto semplici: con il comando CLOAD si carica nel computer il programma e con CSAVE lo si memorizza sulla nuova cassetta. Se invece dobbiamo duplicare un programma salvato con il comando BSAVE non è possibile farne una sua copia se non si conoscono gli indirizzi d'inizio, fine e start del comando BSAVE. Questo programma, il cui uso deve essere strettamente personale, consente di ricavare i dati necessari per copiare i programmi registrati con il comando BSAVE.

Una volta dato il RUN al programma aspettate fino a che le istruzioni scompaiono dal video, dopodiché inserite la cassetta posizionando il nastro all'inizio del programma da copiare e premete il tasto PLAY del registratore. Dopo pochi secondi appariranno sullo schermo le informazioni riguardanti il programma: nome, indirizzo d'inizio, indirizzo di fine e indirizzo di start. Vi conviene annotare tali dati se volete procedere voi



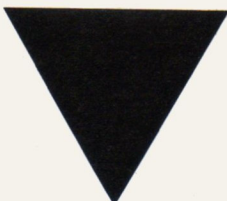
stessi alla duplicazione del programma, altrimenti seguite le istruzioni che appariranno sul video.

La routine in linguaggio macchina che legge l'header (l'insieme dei dati

riguardanti il tipo di programma, il suo nome, ecc.) risiede alle linee 530-560.

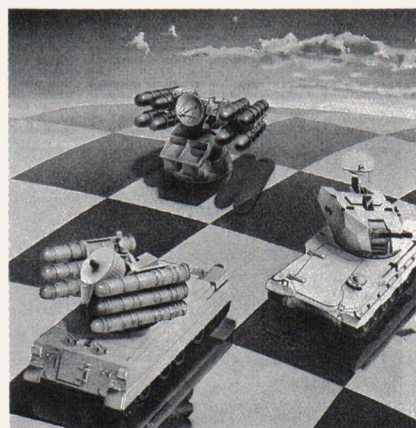
Per caricare il programma digitare CLOAD "DUPLI".

10



BOMB SHELL ("BOMB")

adatt. da G. Riccobono



All'inizio del gioco vedrete apparire sullo schermo dei muri, delle bombe azzurre, dei caratteri simili a dei quadratini bianchi, una lettera «S» e una lettera «G» entrambe gialle.

Sopra la lettera «S» compare un piccolo omino rosso che voi dovete muovere sullo schermo usando i tasti cursore. Premendo la barra spaziatrice il vostro omino sparerà. Scopo del gioco è di distruggere, sparando il vostro proiettile, tutte le bombe azzurre presenti sullo schermo. Attenzione però!, dai quadratini bianchi escono

degli omini verdi che vi inseguono. Se uno di loro vi raggiunge vi uccide col suo tocco mortale. Attenzione anche alle bombe: dopo essere esplose emanano un gas tossico velenosissimo il cui contatto è letale! I vostri proiettili non possono, ovviamente oltrepassare i muri ma il fumo sì. Una volta che avete fatto esplodere tutte le bombe dovete passare sopra la lettera «G» per passare al livello di gioco successivo. Le bombe sono numerate da 1 a 3. Quelle da «1» producono poco gas, quelle da «2» una quantità media e quella da «3» tanto gas. Il program-

ma salta i livelli 8, 11, 20, 23, 32 e 35.

Il programma è salvato in due blocchi ("BOMB" in Basic e "LM.OBJ" in 1/m). Per caricare il gioco digitare:

CLOAD "BOMB"

poi una volta apparsa la scritta "OK" fermate il registratore e date RUN+ENTER. Apparirà la scritta "BOMB SHELL" e più in basso «schiaia barra spaziatrice». Premete la barra di spazio e il tasto PLAY del registratore per il caricamento del linguaggio macchina.

SUPER SCREEN 2

VELOCIZZIAMO LA STAMPA ED INTRODUCIAMO GLI UDG ANCHE NELLO SCHERMO GRAFICO

di MARGIE TORNABUONI

Sappiamo che le istruzioni di invio caratteri sul video sono, nell'MSX Basic, veloci e complete (PRINT, LOCATE, SPC, TAB e così via) ad eccezione della stampa in schermo grafico. Infatti per poter scrivere in SCREEN 2 non possiamo eseguire una semplice istruzione di PRINT ma dobbiamo associare quest'ultima ad un particolare canale aperto precedentemente con l'istruzione OPEN "GRP:" FOR OUTPUT AS #n. Inoltre il posizionamento di stampa non è più effettuato con l'istruzione LOCATE ma puntando la stampa direttamente in riferimento alle coordinate grafiche. Cioè dobbiamo precedere l'istruzione PRINT #n, "stringa" da un'istruzione del tipo PSET, PRESET ecc. Il vantaggio di questo indirizzamento di stampa è quello di poter posizionare i caratteri in qualunque punto del video senza limitarsi alle 24 righe concesse nei modi di testo.

Il programma che stiamo per presentare vuole ribaltare la situazione, cioè aumentare notevolmente la velocità e limitare il posizionamento di stampa in screen 2. La ragione di questa scelta risiede nel fatto che solitamente si utilizza lo screen 2 per eseguire giochi o grafici in cui il tempo di elaborazione determina molto il risultato finale; ora associando all'elaborazione anche una stampa (per esempio il punteggio in un gioco, le coordinate grafiche in un programma per disegnare) il tempo d'esecuzione del programma si riduce notevolmente causando degli effetti spiacevoli.

Noi abbiamo risolto il problema scrivendo una subroutine in linguaggio macchina. Il risultato è stato eccellente: velocità di stampa quattro volte superiore a quella in screen 0 e possibilità di ridefinire i caratteri in screen 2.

Vediamo ora come utilizzare il programma.

Per chi avesse un assembler può

usufruire dei due listati assembler, ma attenzione; il primo va caricato ed eseguito immediatamente per le ragioni che spiegheremo tra poco, mentre il secondo viene lanciato solo quando si vuole stampare nello schermo grafico. Per chi invece non disponesse di un programma assembler dovrà copiare il listato Basic.

Il programma è costituito da due

IL PROGRAMMA DI STAMPA

```

10 CLEAR 200,60000!
20 GOSUB 60100/CARICAMENTO L/M
30 'ESEMPIO
40 POKE &HEAE4,10/COORDINATA RIGA
50 POKE &HEAE5,3/COORDINATA COLONNA
60 S$="MSX COMPUTER MAGAZINE"
70 SCREEN 2
80 S$=USR(S$)
90 GOTO 90
60000 /*****
60010 /* SUBROUTINE SUPER SCREEN 2 *
60020 /* by *
60030 /* *
60040 /* ORIGINAL SOFT. *
60050 /*****
60060 /
60070 /**** COPIA IN RAM I ****
60080 /**** PROFILI CARATTERI****
60090 /

```




programmi assembler; il primo, linee 60100-60120 (listato Basic), viene caricato in memoria ed eseguito subito perché serve a trasportare la tabella profili dei caratteri in screen 1 nella memoria RAM a partire dall'indirizzo 60280 fino all'indirizzo 62328, il secondo programma, che viene caricato dall'indirizzo 60001 all'indirizzo 60136, è la subroutine vera e propria

che va a sovrapporsi alla precedente già utilizzata. Ecco perché coloro che utilizzano un assembler devono prima caricare il programma assembler numero 1, eseguirlo e poi caricare il programma assembler numero 2. Comunque consigliamo a tutti di caricare il listato Basic proprio perché esegue direttamente ogni procedura sopra detta e contiene anche un

```

60100 SCREEN 1
60110 DATA 01,00,08,11,78,EB,21,00,00,CD
,59,00,C9
60120 RESTORE 60110:FOR N=60001! TO 6001
3!:READ A#:POKE N,VAL("&H"+A#):NEXT:DEFU
SR=60001!:A=USR(0):SCREEN 0
60130 /
60140 **** CARICAMENTO L/M ****
60150 /
60160 DATA 1A,47,AF,B8,C8,13,1A,32,E6,EA
,13,1A,32,E7,EA,C5,CD,A8,EA,CD,BC,EA,C1,
ED,5B,E6,EA,1A,13,ED,53,E6,EA
60170 DATA C5,E5,01,08,00,54,5D,CD,CF,EA
,CD,5C,00,E1,E5,11,00,20,19,01,08,00,3A,
E8,EA,CD,56,00,E1,06,08,23,10,FD
60180 DATA C1,10,D1,C9,21,E9,F3,46,CB,20
,CB,20,CB,20,CB,20,3A,EA,F3,B0,32,E8,EA,
C9,21,E4,EA,46,60,E5,21,E5,EA,46
60190 DATA E1,CB,20,CB,20,CB,20,68,C9,C5
,D5,21,78,EB,11,00,00,5F,06,03,CB,23,CB,
12,10,FA,19,D1,C1,C9,00,00,00,00,00
60200 RESTORE 60160:FOR N=60001! TO 6013
6!:READ A#:POKE N,VAL("&H"+A#):NEXT:DEFU
SR=60001!:RETURN

```

LE ROUTINE

1

```

1 ; ROUTINE N.1
2 ; PER LA COPIA PROF
  ILI-CARATTERI
3 ;
4   ORG 60001
5   LD BC,2048
6   LD DE,60280
7   LD HL,0
8   CALL 0059H
9   RET
10  END

```

2

```

1 ; ROUTINE N.2
2 ; STAMPA VELOCE IN SCREEN 2
3 ;
4   ORG 60001
5   LD A,(DE)
6 ; B=LEN STRINGA
7   LD B,A
8   XOR A
9   CP B
10 ; RITORNA SE LA
11 ;STRINGA E' NULLA
12  RET Z
13 ; (START)=INIZIO STRINGA
14  INC DE
15  LD A,(DE)
16  LD (START),A
17  INC DE
18  LD A,(DE)
19  LD (START+1),A
20  PUSH BC
21 ; CALCOLA CODICE DI
22 ; INK E PAPER
23  CALL ATTR
24 ; CALCOLA INDIRIZZO
25 ; DI INIZIO STAMPA
26  CALL CALCI
27  POP BC
28 ; INIZIA LA PARTE
29 ; PRINCIPALE
30 MAIN:      LD DE,(START)
31 ; A=CARATTERE DA STAMPARE
32  LD A,(DE)
33 ; AGGIORNA IL PUNTAIORE
34 ; DI STRINGA
35  INC DE
36  LD (START),DE
37  PUSH BC
38  PUSH HL
39 ; N. DATI DA TRASFERIRE
40  LD BC,8
41 ; DE=INDIRIZZO PROFILI
42 ; IN RAM
43  LD D,H
44  LD E,L
45 ; CHIAMATA TAB: RESTITUISCE
46 ; HL=INDIRIZZO IN RAM PROFILO
47 ; CARATTERE
48  CALL TAB
49 ; STAMPA IL CARATTERE
50  CALL 005CH
51  POP HL
52  PUSH HL
53  LD DE,8192
54 ; HL=INDIRIZZO TABELLA COLORI
55  ADD HL,DE
56 ; N. DATI DA TRASFERIRE
57  LD BC,8
58 ; A=CODICE COLORE
59  LD A,(ATT)
60 ; COLORA IL CARATTERE
61  CALL 0056H
62  POP HL
63  LD B,8

```


SOFTWARE

```
64 AGGIORNA: INC HL
65 : HL=PROSSIMO INDIRIZZO
66 : DI STAMPA
67 : DJNZ AGGIORNA
68 : POF BC
69 : DJNZ MAIN
70 : RET
71 :
72 : CALCOLO CODICE COLORE
73 :
74 ATTR: LD HL,0F3E9H
75 : B=CODICE INK
76 : LD B,(HL)
77 : INK NEL NIBBLE SUPERIORE
78 : SLA B
79 : SLA B
80 : SLA B
81 : SLA B
82 : A=CODICE PAPER
83 : LD A,(0F3EAH)
84 : (ATT)=BYTE COLORE
85 : OR B
86 : LD (ATT),A
87 : RET
88 :
89 : CALCOLO INDIRIZ. PRO
    FILI VRAM
90 :
91 CALC1: LD HL,RIGA
92 : LD B,(HL)
93 : LD H,B
94 : PUSH HL
95 : LD HL,COLONNA
96 : LD B,(HL)
97 : POP HL
98 : SLA B
99 : SLA B
100 : SLA B
101 : B=B*8 INDIRIZZO COLONNA
102 : LD L,B
103 : RET
104 :
105 : INPUT A=CARATTERE
106 : OUTPUT HL=IND. PROFILO
    CARATTERE
107 :
108 TAB: PUSH BC
109 : PUSH DE
110 : LD HL,60280
111 : HL=BASE PROFILI
112 : LD DE,0
113 : LD E,A
114 : LD B,3
115 TAB2: SLA E
116 : RL D
117 : DJNZ TAB2
118 : DE=A*B
119 : ADD HL,DE
120 : HL=START PROFILO CARATTERE
121 : POP DE
122 : POP BC
123 : RET
124 :
125 : VARIABILI
126 :
127 RIGA: DEFB 0
128 COLONNA: DEFB 0
129 START: DEFW 00
130 ATT: DEFB 0
131 : END
```

esempio di utilizzo della routine. Infatti osservando le linee del listato dalla 40 alla 80 si può capire chiaramente quali sono i parametri necessari per la stampa veloce in screen 2.

LA POSIZIONE DI STAMPA

Prima di tutto occorre definire lo start della subroutine in linguaggio macchina (DEFUSR=60001, linea 60200), dopodiché inserire nella locazione di memoria &HEAE4 e &HEAES le coordinate di riga e colonna relative all'inizio stampa, proprio come in screen 1. Il valore inseribile all'indirizzo &HEAE4 (coordinata riga) varia tra 0 e 23 mentre quello all'indirizzo &HEAE5 (coordinata colonna) varia tra 0 e 31; se tali locazioni di memoria non vengono modificate mantengono l'ultimo valore assegnato loro; inizialmente le coordinate sono a 0,0 riferite cioè all'angolo superiore sinistro dello schermo. Dopo aver definito la posizione di stampa dobbiamo decidere cosa stampare inserendo ciò che dobbiamo scrivere in una variabile alfanumerica (nel listato, linea 60, è la variabile S\$), se dobbiamo stampare un numero lo assegnamo ad una stringa con la funzione STR\$(numero). Finalmente, dopo essere entrati in screen 2, possiamo richiamare la subroutine in 1/m ricordandoci di adoperarla come alla linea 80, cioè il parametro di USR deve essere la stringa da stampare, diciamo stringa e non variabile stringa perché la linea 80 potrebbe essere anche:

```
80 S$=USR("MSX
COMPUTER MAGAZINE")
```

cioè possiamo passare direttamente i caratteri da stampare nella funzione USR senza doverli assegnare ad una variabile. Sta a voi utilizzare la forma che vi è più comoda. Osservate i risultati e confrontateli con quelli precedenti offerti dall'MSX Basic, inutile fare delle considerazioni, non

c'è paragone in termini di velocità!

Inizialmente abbiamo accennato al fatto che con questa subroutine è possibile stampare anche caratteri ridefiniti dall'utente, infatti con questa utility la stampa in screen 2 si avvale di una tabella profili caratteri non più in ROM ma in RAM (trasportata dalla prima routine in 1/m). A questo punto appare assai semplice il modo di ridefinire un carattere. Supponiamo per esempio di voler definire il carattere spazio in modo inverso, cioè un quadretto pieno. Sapendo che il suo codice è 32 e che per ottenere la forma di un quadretto pieno bisogna inserire in ogni byte del carattere il codice 255, il programmino che esegue tale operazione è il seguente:

```
10 START=60280+32*8
20 FOR N=START TO START+7
30 POKE N,255
40 NEXT N
```

Sapendo che START è la posizione d'inizio in RAM del carattere spazio, si esegue un ciclo da START a START+7 che modifica il valore di queste locazioni ottenendo così uno spazio inverso.

In chiusura di questo articolo vogliamo soddisfare anche i patiti di assembler che oltre ai risultati della subroutine vorranno anche sapere come sono stati raggiunti.

La routine numero 1 è stata descritta in precedenza e quindi passiamo direttamente a commentare il listato assembler numero 2.

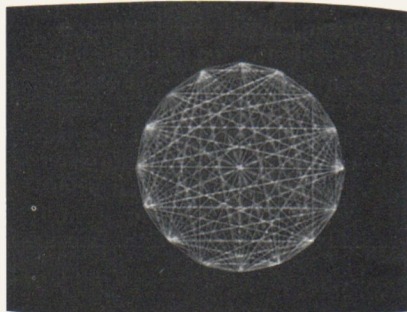
IL SECONDO LISTATO

La prima istruzione che troviamo è una LD A,(DE); dobbiamo a questo punto spiegare che quando si utilizza la funzione USR con parametri alfanumerici il registro DE, al momento della chiamata USR, punta all'indirizzo in memoria contenente i dati relativi al parametro alfanumerico passato con USR. Il primo byte contiene

TEMA UNA RAGAZZA AL MARE

SVOLGIMENTO UN CENTOMILA

la lunghezza della stringa, il secondo e il terzo byte l'indirizzo dove è memorizzata la stringa. A questo punto con l'istruzione LD A,(DE) si consi-



dera la lunghezza della stringa e si esegue subito un ritorno se la stringa in esame è nulla. Altrimenti si recupera l'indirizzo d'inizio stringa e lo si memorizza nella locazione START.

I COLORI DELLA STAMPA

Successivamente si chiama la subroutine ATTR (linea 23) la quale pone nella locazione ATT il codice per il colore della stampa. Quest'ultimo è sempre quello utilizzato come colori generali d'inchiostro e carta dello screen 2. Calcolato il colore di stampa il programma chiama, alla linea 26, la subroutine CALC1 la quale calcola, sulla base dei valori di RIGA e COLONNA, l'inizio stampa in screen 2. Dalla linea 30 alla linea 70 risiede il programma principale che viene eseguito tante volte quanti sono i caratteri da stampare, cioè la lunghezza della stringa da inviare allo screen 2. All'interno di questa parte di programma viene fatto avanzare il puntatore di stringa (linea 35), calcolato (linea 48) l'indirizzo d'inizio profilo carattere, stampato (linea 50) e colorato (linea 61). Viene incrementato il puntatore di stampa in screen 2 e ripetute le operazioni sopra viste tante volte quanti sono i caratteri da stampare.



MARISA MODEL

Hai un computer e sai disegnare sul video. Prova a svolgere il tema proposto: una ragazza al mare, o in piscina. Con contorno magari di ombrelloni coloratissimi, del blu dell'acqua, di bikini o di monokini di moda, e di tutto quant'altro vuoi. Inviaci lo screen. Se è bello lo utilizziamo noi per la copertina (facciamo noi la foto a meno che tu non voglia già inviare una diapositiva a colori) e ti paghiamo centomila lire. Naturalmente il tuo nome viene citato come autore del disegno. La cassetta o la foto deve essere inviata in redazione (MSX Computer Magazine, Vitt. Emanuele 15, 20122 Milano), la modella può rimanere con voi (ci contentiamo dell'immagine). Il materiale inviato non potrà essere restituito se non accompagnato da indirizzo e affrancatura (visto che costi postali?!). Coraggio: le centomila son pronte per un disegno graficomputerizzato che si faccia guardare!

UNA MAGLIETTA IN REGALO!

a chi si abbona a

**MSX
COMPUTER
MAGAZINE**



sei magnifiche cassette di programmi di gioco e di utilità, sempre più belle e ricche!



il prezzo dell'abbonamento (Lire 50 mila) è bloccato per sei numeri e non ti verranno quindi richiesti aumenti (già subito intanto risparmi 4 mila lire)!



avrà subito, direttamente a casa, un'elegante maglietta (realizzata con le riviste consorelle Elettronica 2000 e Load'n'Run) assolutamente gratis!

**ABBONATI
OGGI
STESSO**

Basta inviare un vaglia ordinario (quello rosa, da richiedere in un qualunque ufficio postale) di lire 50 mila.

Indica esattamente da quale fascicolo desideri l'abbonamento ed i tuoi dati chiari e precisi. Indirizza a MSX Computer Magazine, C.so Vitt. Emanuele, 15 - 20122 Milano.



IL LINGUAGGIO MACCHINA

COME PROGRAMMARE IN LINGUAGGIO MACCHINA. I CODICI ISTRUZIONE DEL MICROPROCESSORE Z80 A

(2ª PUNTATA)

Nella scorsa puntata abbiamo parlato della struttura dello Z80, dei suoi registri e presentato una parte del set delle istruzioni di caricamento. Questa volta concluderemo il discorso sulle LOAD ad 8 e 16 bit per poi passare a descrivere i flags e le operazioni aritmetiche.

Tra gli indirizzamenti indiretti abbiamo visto solo, per quelli a 8 bit, LD A, (nn) e, per quelli a 16 bit, LD cr, (nn), dove cr sta ad indicare una coppia di registri (BC, DE, HL, IX, IY ed SP). È chiaro che lo Z80 non offre soltanto i caricamenti da memoria a registro ma anche l'inverso. Così per i trasferimenti a 16 bit vale la forma sintattica LD (nn), cr il cui significato è identico a LD cr, (nn), con la differenza che si sono scambiati gli operandi. Va precisato che in (nn) andrà il byte meno significativo di cr e in (nn+1) quello più significativo. Così per esempio LD (30000), HL pone L in (30000) e H in (30001).

Per gli 8 bit vale l'istruzione LD (nn), A oppure LD (cr), A dove nn è un indirizzo di memoria compreso tra 0 e 65535. Sembra che quindi che tra gli 8 bit l'unico registro privilegiato sia A, questo è vero ma solo in parte, infatti anche gli altri registri possono caricare i loro valori in memoria o dalla memoria, purché si rispetti la seguente sintassi:

LD r, (ss) e LD (ss), r

dove r è un qualunque registro ad 8 bit e ss è il registro HL, IX o IY. Per

questi ultimi due registri a 16 bit dobbiamo aprire una parentesi sul loro utilizzo per indirizzare una locazione di memoria. IX e IY devono essere usati insieme ad un byte di displacement (spiazzamento). Il byte di displacement è un numero che può avere valori positivi o negativi compresi tra +127 e -128. Quindi la forma corretta dell'utilizzo di IX e IY è: LD (IX+d), r o LD r, (IX+d) e LD (IY+d), r o LD r, (IY+d)

dove d è un valore compreso tra +127 e -128. Questo tipo d'indirizzamento viene detto indicizzato perché utilizza appunto i registri indice.

Dopo la descrizione dell'indirizzamento con IX e IY, torniamo a parlare delle istruzioni di caricamento. Lo Z80 offre non solo la possibilità di memorizzare un registro in una locazione di memoria ma anche direttamente un numero, purché questo sia ad un byte. L'istruzione generale è la seguente: LD (ss), n; dove ss è HL, IX+d o IY+d e n un valore compreso tra 0 e 255.

Chiudiamo l'argomento delle istruzioni di caricamento descrivendo i due comandi relativi ai 16 bit: PUSH e POP.

Per comprendere queste due istruzioni bisogna rifarsi al concetto di stack, già visto la scorsa volta ma che riprenderemo in questa sede. Per stack (pila) s'intende un sistema di memorizzazione dati. Lo stack è organizzato in modo tale che ogni nuovo dato memorizzato in esso si colloca in "cima" agli altri. Il puntatore allo stack è il registro SP e quando lo stack è vuoto SP ha un indirizzo di memoria alto; invece man mano che lo stack si riempie di dati il registro SP diminuisce il suo valore. In altre parole la memoria dello stack si estende dall'alto verso il basso. Nella pila è possibile memorizzare solo registri a 16 bit compresi IX e IY ed escludendo, ovviamente, SP. L'istru-



COMPUTER MATES

ASSEMBLER

zione PUSH cr decrementa di 2 il valore di SP e poi immette all'indirizzo (SP) il valore del registro cr. POP cr invece carica nel registro cr il contenuto dell'indirizzo (SP) e poi incrementa di 2 SP aggiornando lo stack. Da quanto detto appare chiara la regola che l'ultimo dato immesso è anche il primo ad uscire. Prima di proseguire dobbiamo spiegare il concetto di rappresentazione in "complemento a due". Questo termine verrà usato più volte ed è quindi necessario darne una spiegazione.

La rappresentazione in complemento a due è un modo di codificare gli interi negativi e positivi. La normale rappresentazione binaria con tre bit ci permette di rappresentare i numeri decimali da 0 a 7. La codifica in complemento a due con tre bit ci

permette di rappresentare gli interi tra -4 e +3 come mostra la seguente tabella:

Numero decimale	Complemento a due
3	011
2	010
1	001
0	000
-1	111
-2	110
-3	101
-4	100

Cioè i numeri positivi hanno tutti i codici in complemento a due con il primo bit uguale a 0, mentre i codici dei numeri negativi iniziano con 1. Avendo a disposizione una rappresentazione ad n bit, il suo complemento a due codifica i numeri da $-2^{(n-1)}$ a $+2^{(n-1)}-1$. Quindi in un registro

a 8 bit possiamo rappresentare numeri da -128 a +127, mentre in un registro a 16 bit da -32768 a +32767. Per ottenere il complemento a due di un numero è sufficiente modificare tutti gli 1 del numero in 0 e tutti gli 0 in 1, dopodiché sommare 1 alla nuova rappresentazione. Supponiamo quindi di avere una rappresentazione a 8 bit del numero 01000000 (valore decimale 64) e di voler trovare il suo complemento a due. Per prima cosa cambiamo gli 1 in 0 e gli 0 in 1 ed otteniamo 10111111. A questo nuovo numero sommiamo 1, cioè 00000001, e il nuovo risultato è 11000000, che in complemento a due rappresenta -64. Precedentemente abbiamo detto che se il primo bit è 0 il numero è positivo ed il suo valore è la reale rappresentazione binaria. Se invece il numero inizia con 1 vuol dire che è negativo e per sapere il suo valore bisogna eseguire la seguente formuletta: $v-2^n$, dove v è il valore reale del numero (cioè senza considerare il complemento a due) ed n è il numero di bit della rappresentazione. Facciamo subito un esempio: abbiamo il numero 11101001 e vogliamo sapere il suo valore in complemento a due. Il valore binario di 11101001 è in decimale 233. Il numero di bit è 8 quindi il suo valore è $233-2^8$, cioè -23.

Sperando di essere stati sufficientemente chiari passiamo alla descrizione del registro F dei flags, argomento anch'esso assai importante per poter comprendere successivamente i risultati delle istruzioni aritmetiche e di manipolazione dei bit.

Il registro F è ad 8 bit ma per comprendere i suoi differenti significati bisogna analizzare alcuni suoi bit, precisamente sei:

carry (C): questo bit viene settato, cioè posto a 1, quando durante un'operazione aritmetica c'è un riporto del bit più significativo del risultato; zero (Z): questo bit è 1 se il risultato di una operazione è zero; segno (S): è semplicemente la copia

1		ORG	62000
2	F230	LD	A, (LOC)
3	F233	LD	B, A
4	F234	LD	A, (LOC+1)
5	F237	ADD	A, B
6	F238	LD	(LOC+2), A
7	F23B	INC	B
8	F23C	LD	A, B
9	F23D	LD	(LOC+3), A
10	F240	LD	HL, (LOC+4)
11	F243	LD	DE, (LOC+6)
12	F247	ADD	HL, DE
13	F248	LD	(LOC+8), HL
14	F24B	RET	
15	F24C	DEFB	0
16	F24D	DEFB	0
17	F24E	DEFB	0
18	F24F	DEFB	0
19	F250	DEFW	00
20	F252	DEFW	00
21	F254	DEFW	00
22		END	

Un programmino d'esempio per quanto descritto in questa seconda lezione.

del bit più significativo del risultato di un'operazione. S indica se il risultato, in complemento a due, è positivo (0) o negativo (1);

parità/overflow (P/V): il bit P/V assume risultati differenti a seconda del tipo d'istruzione eseguita. Per le istruzioni aritmetiche segnala l'overflow tra numeri in complemento a due. Precisamente è 1 se la somma di due numeri positivi genera un numero negativo oppure se la somma di due numeri negativi genera un numero positivo. Durante le operazioni di manipolazione dei bit il flag P/V è 1 ad indicare che la parità (il numero di bit a 1 in un byte) è pari ed è 0 se la parità è dispari;

sottrazione (N): bit dedicato esclusivamente alla logica BCD;
half-carry (H): bit dedicato esclusivamente alla logica BCD.

Abbiamo analizzato i diversi flag. Si parlerà ancora di loro durante la presentazione di nuove istruzioni.

Le istruzioni aritmetiche dello Z80 comprendono operandi a 16 e a 8 bit. Incominciamo ad analizzare le operazioni dedicate ai 16 bit.

Vi sono quattro tipi di operazioni: incremento, decremento, somma e sottrazione. L'istruzione d'incremento è la seguente: INC cr, dove cr è una coppia di registri (BC, DE, HL, IX, IY e SP). Il suo effetto è quello di incrementare di 1 il valore di cr. L'istruzione di decremento è DEC cr ed il suo effetto è quello di decrementare di 1 il valore di cr. Sia INC che DEC non modificano alcun flag.

L'operazione di somma a 16 bit prevede la seguente forma:

ADD ss,cr

dove ss può essere il registro HL, IX o IY e cr è una coppia di registri. È da notare che è valida l'istruzione ADD HL,HL mentre non lo sono ADD IX,IY, ADD IY,IX, ADD HL,IX e ADD HL,IY. L'istruzione generale ADD ss,cr non modifica alcun flag eccetto N che lo pone sempre a 0. Per modificare i flag dopo un'operazione

```

40 CLEAR 200,&HF22F
50 'SEGUE DATI L/M IN ESADECIMALE
60 DATA 3A,4C,F2:' LD A,(F24C)
70 DATA 47:' LD B,A
80 DATA 3A,4D,F2:' LD A,(F24D)
90 DATA 80:' ADD A,B
100 DATA 32,4E,F2:' LD (F24E),A
110 DATA 04:' INC B
120 DATA 78:' LD A,B
130 DATA 32,4F,F2:' LD (F24F),A
140 DATA 2A,50,F2:' LD HL,(F250)
150 DATA ED,5B,52,F2:' LD DE,(F252)
160 DATA 19:' ADD HL,DE
170 DATA 22,54,F2:' LD (F254),HL
180 DATA 09:' RET
190 DATA 00:' (F24C)
200 DATA 00:' (F24D)
210 DATA 00:' (F24E)
220 DATA 00:' (F24F)
230 DATA 00,00:' (F250-F251)
240 DATA 00,00:' (F252-F253)
250 DATA 00,00:' (F254-F255)
260 'CARICAMENTO DATI L/M IN MEMORIA
270 FOR N=&HF230 TO &HF255
280 READ A$:POKE N,VAL("&H"+A$)
290 NEXT
300 DEFUSR=&HF230 'START L/M
310 CLS:INPUT "A <0-255>";A%
320 IF A%>255 OR A%<0 THEN 310
330 POKE &HF24D,A%
340 INPUT "B <0-255>";B%
350 IF B%>255 OR B%<0 THEN 340
360 POKE &HF24C,B%
370 INPUT "HL <0-65535>";HL
380 HL=INT(HL)
390 IF HL>65535! OR HL<0 THEN 370
400 POKE &HF250,HL-INT(HL/256)*256
410 POKE &HF251,INT(HL/256)
420 INPUT "DE <0-65535>";DE
430 DE=INT(DE)
440 IF DE>65535! OR DE<0 THEN 420
450 POKE &HF252,DE-INT(DE/256)*256
460 POKE &HF253,INT(DE/256)
470 A=USR(0)'ESECUZIONE L/M
480 PRINT "IL RISULTATO DI ADD A,B";
490 PRINT " E':";PEEK(&HF24E)
500 PRINT "IL RISULTATO DI INC B";
510 PRINT " E':";PEEK(&HF24F)
520 PRINT "IL RISULTATO DI ADD HL,DE";
530 R=PEEK(&HF254)+256*PEEK(&HF255)
540 PRINT " E':";R
550 PRINT "VUOI ESEGUIRE NUOVAMENTE"
560 PRINT " IL PROGRAMMA <S/N>";
570 INPUT R$
580 IF R$="S" OR R$="s" THEN 310
590 END

```

Analizzate l'assembler e poi caricate il Basic: comprenderete maggiormente le istruzioni descritte.

ASSEMBLER

di somma a 16 bit bisogna utilizzare l'istruzione ADC HL,ss (ss è uno dei seguenti registri: BC,DE,HL e SP) la quale somma al registro HL il valore del registro ss più il valore del flag C, cioè 1 o 0. I flag modificati da questa istruzione sono: S,Z,C e N; quest'ultimo posto sempre a 0. Infine, tra le istruzioni aritmetiche a 16 bit, troviamo l'istruzione di sottrazione SBC HL,ss per la quale vale lo stesso discorso di ADC HL,ss, con la differenza che anziché sommare si sottrae.

Ed ora veniamo alle istruzioni aritmetiche ad 8 bit. Anch'esse, come

un qualsiasi registro ad 8 bit, (HL), (IX+d) o (IY+d). La sua forma sintattica è ADD A,s, dove s sono gli operandi appena citati. Esiste poi la possibilità di sommare all'accumulatore anche il valore del flag di carry con l'istruzione ADC A,s.

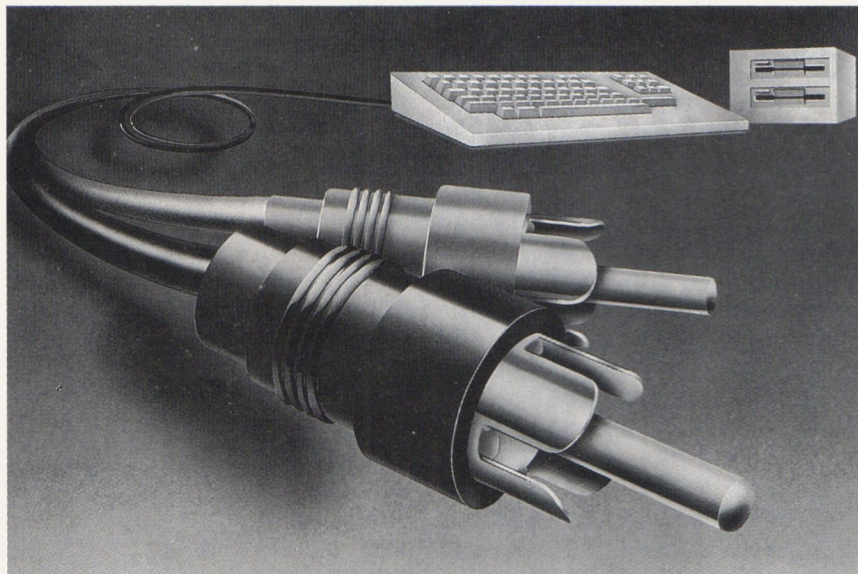
Anche per la sottrazione esistono due istruzioni: la prima sottrae ad A il valore di s ed è SUB s, la seconda sottrae ad A oltre ad s anche il valore di C ed è SBC A,s.

Nelle operazioni di INC e DEC a 8 bit tutti i flag, eccetto C che rimane invariato, vengono modificati. Nelle

operazione ADD A,200. Dopo quest'ultima operazione in A avremo il valore 4 e nel flag C il valore 1. Infatti $200+60$ è superiore a 255 (il massimo valore rappresentabile con 8 bit) e quindi C sta ad indicare che si è superati 255. Cioè rappresenta il bit di riporto. Il flag di overflow P/V si comporta allo stesso modo di C con la sola differenza che riferisce ai calcoli in complemento a due. Riprendiamo le operazioni appena viste. LD A,200 in complemento a due sta a significare carica in A il valore -56 ($256-200$). ADD A,60 quindi si traduce come $-56+60$, risultato 4. Dopo queste operazioni il flag P/V è a 0 ad indicare che l'operazione avvenuta è corretta, cioè è rimasta nel range consentito da -128 a $+127$ (per gli 8 bit).

Considerando sempre operazioni in complemento a due proviamo ad eseguire LD A,200 (200 sta per -56) e ADD A,156 (156 sta ad indicare -100). Il risultato in A sarà 100, il flag C sarà settato e anche il flag P/V ad indicare che l'operazione in complemento a due è andata in overflow, infatti $-56+(-100)=-156$ che è fuori dai numeri negativi rappresentabili in un byte.

A conclusione di questa seconda puntata vogliamo proporvi un programmino in Assembler (listato 1), e la sua codifica in codici macchina per l'MSX (listato 2), che utilizza le

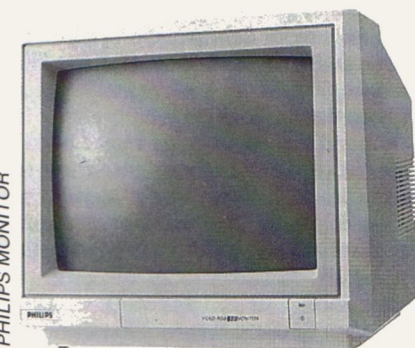


quelle a 16 bit, sono quattro: incremento, decremento, somma e sottrazione. L'istruzione d'incremento è identica a quella a 16 bit: INC s, dove s è un qualsiasi registro a 8 bit, (HL), (IX+d) o (IY+d). Quindi l'istruzione INC a 8 bit può anche incrementare il contenuto di una locazione di memoria. Allo stesso modo opera l'istruzione di decremento: DEC s.

L'istruzione di somma agisce, come primo operando, solo sul registro A ma in compenso come secondo operando può avere un numero a un byte,

istruzioni di addizione e sottrazione anche il flag di carry viene modificato.

Va ora fatta una breve descrizione sul comportamento dei flag di carry e overflow durante le operazioni aritmetiche. Tali considerazioni sono valide sia per le operazioni a 8 bit che per quelle a 16 bit. Il flag di carry dopo un'operazione di somma, per esempio, sarà a 1 se il risultato è stato maggiore di un valore a 8 bit o 16 bit. Consideriamo per esempio di avere in A il valore 60 e di eseguire l'ope-



istruzioni fino adesso imparate. Il suo scopo è quello di caricare due numeri ad 8 bit nei registri A e B, eseguire la loro somma, porre in memoria il risultato e il valore di B incrementato di 1. Allo stesso modo è realizzata una somma 16 bit con i registri HL e DE ed il risultato è posto in memoria.

Analizzate il listato Assembler e poi caricate il listato Basic e fatelo girare, esso vi aiuterà a comprendere maggiormente le istruzioni descritte in questa puntata.



MSX BANK

FINO AD UN MILIONE PER UN PROGRAMMA TUO, CHE SIA ORIGINALE

Se hai programmi originali, che siano proprio pensati e fatti da te, di giochi e utilità, mandaceli in visione. Quelli pubblicati verranno compensati e firmati con il vostro nome. S'intende che se realizzerai qualcosa di davvero favoloso potrai guadagnare addirittura 1 milione di lire! Registra un solo programma per ogni cassetta, completo di istruzioni scritte, ed invialo a
MSX Computer Magazine, c.so Vittorio Emanuele 15, 20122 Milano.

ATTENZIONE: I programmi inviati debbono essere assolutamente inediti, a noi ceduti in esclusiva. L'editore si riserva di rifiutare programmi anche già accettati e di non compensare programmi già pubblicati qualora si verificasse che gli stessi programmi (anche quando con titoli diversi) siano stati ceduti o pubblicati da altri.



3 PROGRAMMI 3

TRE ROUTINES GRAFICHE PER ABBELLIRE, COLORARE E POTENZIARE I VOSTRI PROGRAMMI. DAI CARATTERI PIÙ GRANDI AL CURSORE LAMPEGGIANTE ALL'INVERSE COLOR

Sappiamo che non sempre i programmatori hanno bisogno di complessi programmi ma anzi, i veri "smanettoni" del computer sono sempre alla ricerca di piccole e potenti utility, possibilmente in linguaggio macchina, da inserire nei propri mega programmi così da renderli sempre più funzionali e belli. Quello che stiamo per proporvi in questo articolo sono tre routine in linguaggio macchina adatte proprio a quelle persone

appena citate. Non ci resta quindi che passare alla descrizione di ogni programma.

INGRANDITORE DI CARATTERI

Il programma del listato 1 vi permette di visualizzare, solo in SCREEN 1, i caratteri otto volte più grandi della loro dimensione normale. Il programma è strutturato nel seguente modo: dalla linea 10 alla linea

40 inizializzazione, dalla linea 50 alla linea 220 dimostrazione e dalla linea 1000 alla linea 1230 la subroutine di caricamento in memoria del linguaggio macchina. Sarà quest'ultima parte del listato che dovrà essere inserita nei programmi che vorranno utilizzare la routine di ingrandimento caratteri.

Per far funzionare la parte in linguaggio macchina bisogna prima di tutto settare alcune locazioni di me-

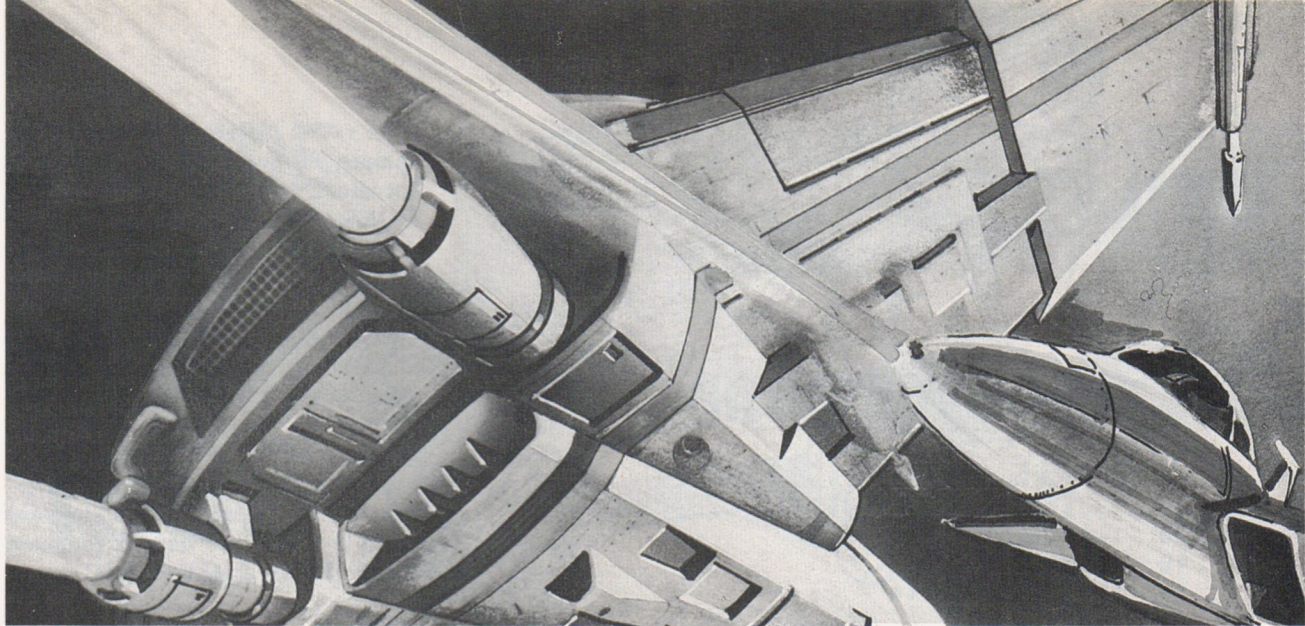
INGRANDITORE DI CARATTERI

```

10 SCREEN 1
20 CLEAR 200,56999!
30 GOSUB 1000
40 DEFUSR=57000!
50 /*****
60 / DIMOSTRAZIONE
70 /*****
80 POKE 62030!,5
90 POKE 62029!,3
100 POKE 62031!,77
110 A=USR(0)
120 POKE 62029!,13
130 POKE 62031!,83
140 A=USR(0)
150 POKE 62029!,23
160 POKE 62031!,88
170 POKE 62026!,15
180 A=USR(0)
190 POKE 62028!,5
200 DEFUSR=56900!
210 A=USR(0)
220 END
1000 /*****
1010 / DATI L/M
1020 /*****
1030 DATA 21,F0,07,01,08,00,3E,FF
1040 DATA CD,56,00,01,08,00,3A,4F
1050 DATA F2,3C,21,00,00,3D,CA,C5
1060 DATA DE,09,C3,ED,DE,11,5B,F2
1070 DATA CD,59,00,3A,4E,F2,01,20
1080 DATA 00,21,00,00,3D,CA,DC,DE
1090 DATA 09,C3,D4,DE,11,00,00,3A
1100 DATA 4D,F2,5F,19,ED,5B,22,F9
1110 DATA 19,11,5B,F2,3E,08,F5,1A
1120 DATA 47,3E,FE,CB,78,C4,4D,00
1130 DATA 23,3E,FE,CB,70,C4,4D,00
1140 DATA 23,3E,FE,CB,68,C4,4D,00
1150 DATA 23,3E,FE,CB,60,C4,4D,00
1160 DATA 23,3E,FE,CB,58,C4,4D,00
1170 DATA 23,3E,FE,CB,50,C4,4D,00
1180 DATA 23,3E,FE,CB,48,C4,4D,00
1190 DATA 23,3E,FE,CB,40,C4,4D,00
1200 DATA 13,01,19,00,09,F1,3D,C2
1210 DATA EE,DE,C9,0E
1220 FOR N=57000! TO 57147!:READ A$:POKE
N,VAL("&H"+A$):NEXT N
1230 RETURN

```

Listato 1: permette di visualizzare i caratteri otto volte più grandi



moria aventi il seguente significato:
 62029 = il numero della colonna a partire dalla quale viene stampato il carattere. Questo numero deve essere compreso tra 1 e 25.

62030 = il numero della riga a partire dalla quale viene visualizzato il carattere. Questo numero deve essere compreso tra 1 e 17.

62031 = il valore ASCII del carattere da stampare.

Nella dimostrazione le linee 100,

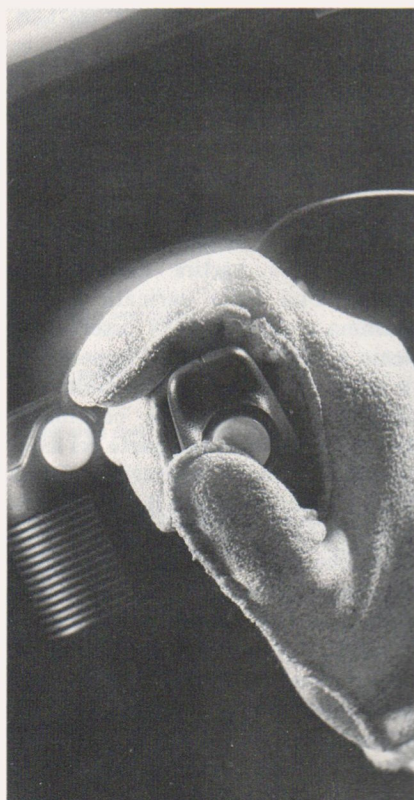
130 e 160 modificano la locazione 62031 rispettivamente con i valori ASCII corrispondenti ai caratteri "M", "S" e "X".

Il linguaggio macchina si colloca dall'indirizzo 57000 all'indirizzo 57147.

FLASH CURSOR

Nei calcolatori MSX il cursore appare come un quadretto bianco e

spesso, quando il video è pieno di caratteri, è difficile riconoscerlo. Poter individuare immediatamente la posizione del cursore vuol dire agevolare di molto la fase di editing e di debugging. Affinché il cursore appaia subito all'occhio del programmatore esso deve lampeggiare. Il programma del listato 2 trasforma il classico cursore dell'MSX in uno lampeggiante. Molti si chiederebbero come questo sia possibile, visto che le routine di gestione



FLASH CURSOR

```

10 CLEAR 200,55743!
20 GOSUB 1000
30 DEFUSR=55744!
40 A=USR(0)
1000 /*****
1010 / DATI L/M
1020 /*****
1030 DATA 11,CD,D9,21,02,FD,3E,C3
1040 DATA 77,23,73,23,72,2A,DC,F3
1050 DATA CD,F2,0B,CD,4A,00,57,3A
1060 DATA A8,FC,B7,20,04,3E,DB,18
1070 DATA 02,3E,FA,CD,4D,00,CD,F3
1080 DATA D9,7A,CD,4D,00,CD,F3,D9
1090 DATA 28,E5,C9,01,00,0F,CD,9C
1100 DATA 00,C0,0B,78,B1,20,F7,C9
1110 FOR N=55744! TO 55807!:READ A#:POKE
    N,VAL("&H"+A#):NEXT N
1120 RETURN
  
```

Listato 2: il cursore finalmente si evidenzia lampeggiando.

UTILITY

INVERSE COLOR

```
10 CLEAR 200,56699!
20 GOSUB 1000:DEFUSR=56700!
30 /*****
40 / DIMOSTRAZIONE
50 /*****
60 SCREEN 1
70 X=0
80 FOR A=6144 TO 6911
90 VPOKE A,X
100 X=X+1
110 IF X=256 THEN X=0
120 NEXT A
130 X=17
140 FOR A=0 TO 31
150 VPOKE 8192+A,X
160 X=X+3
170 NEXT A
180 FOR X=1 TO 1000:NEXT X
190 A=USR(0)
200 FOR X=1 TO 300:NEXT X
210 GOTO 190
1000 /*****
1010 / DATI L/M
1020 /*****
1030 DATA 21,00,20,3E,20,F5,CD,4A
1040 DATA 00,E6,0F,47,CD,4A,00,CB
1050 DATA 0F,CB,0F,CB,0F,CB,0F,E6
1060 DATA 0F,CB,00,CB,00,CB,00,CB
1070 DATA 00,B0,CD,4D,00,23,F1,3D
1080 DATA C2,81,DD,C9,0A
1090 FOR N=56700! TO 56744!:READ A$:POKE
N,VAL("&H"+A$):NEXT N
1100 RETURN
```

Listato 3: consente di scambiare in SCREEN il colore della carta con quello dell'inchiostro.

del cursore sono in ROM e quindi non modificabili. Per risolvere il problema si è utilizzata la tecnica degli "hooks". Gli hooks sono delle zone di memoria situate nella parte alta della RAM di cui se ne serve il sistema per eventuali espansioni quali, per esempio, il disco o l'interfaccia seriale. Cioè il sistema prima di eseguire alcune routine presenti in ROM, controlla che nella zona di memoria dedicata agli hooks non vi sia qualche salto a programmi di estensione del

sistema, in questo caso il nuovo programma è quello di gestione del cursore lampeggiante. Ma degli hooks avremo modo di parlarne più dettagliatamente in futuro, invece ora vediamo come utilizzare questa routine. Caricate il programma e date il RUN. Sia che siate in SCREEN 0 che in SCREEN 1 il cursore ora lampeggerà.

Provate a selezionare il modo insert e il cursore ora avrà la forma di un trattino lampeggiante.

Se volete variare la frequenza del lampeggio modificate il valore della locazione &HD9F5 con l'istruzione POKE &HD9F5,n con n compreso tra 1 e 255. Potete anche modificare la forma del cursore sia nel modo normale che nel modo insert variando i contenuti delle locazioni &HD9DE e &HD9E2 con un valore compreso tra 1 e 255. Per ritornare ai cursori iniziali digitare POKE &HD9DE,&HFA e POKE &HD9E2,&HDA.

Per disabilitare il cursore lampeggiante e ripristinare quello standard MSX è sufficiente dare il comando diretto POKE &HFDC2,&HC9.

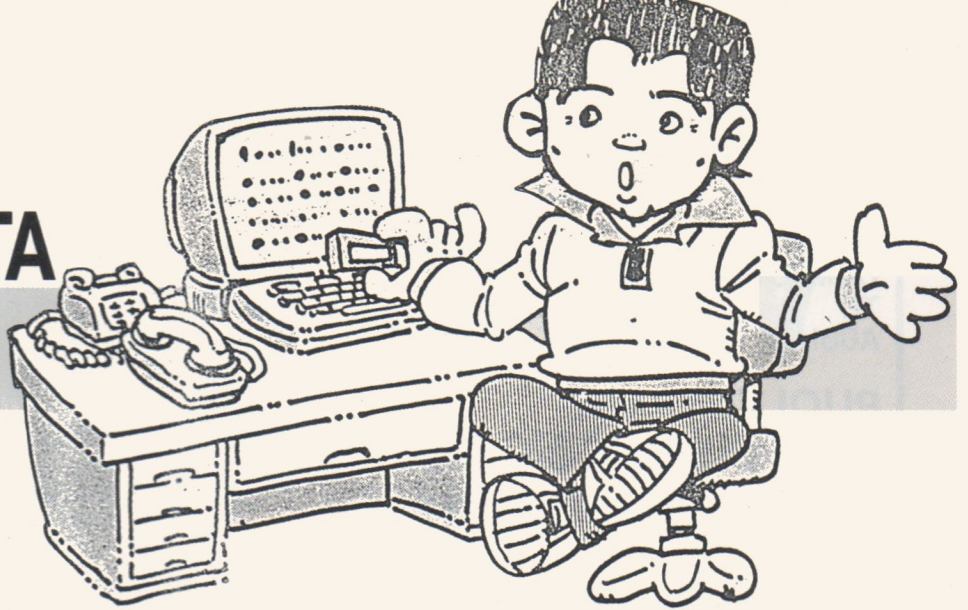
INVERSE COLOR

In molti home e personal computer esiste una particolare istruzione che consente di far lampeggiare e scrivere in inverse i messaggi inviati al video con il classico comando PRINT. L'effetto risultante è quello di evidenziare particolari messaggi o addirittura creare effetti particolari in programmi di gioco. Il programma del listato 3 consente di scambiare, solo in SCREEN 1, il colore della carta con quello dell'inchiostro così da ottenere una specie di funzione "inverse" e, se la routine è attivata di frequente, anche di "flash".

Il programma è strutturato nel seguente modo: linee 10-20 inizializzazione, linee 30-210 dimostrazione e alle linee 1000-1100 subroutine di caricamento in memoria dei codici in linguaggio macchina.

La routine è lunga solo 45 bytes e si colloca a partire dall'indirizzo 56700. Una volta caricato il programma e dato il RUN osservate il demo e subito capirete l'importanza di questa routine e le sue molteplici applicazioni. Prima di concludere diamo un'occhiata alla parte del listato dedicata alla dimostrazione. Dalla linea 80 alla linea 120 il ciclo a conteggio FOR A-NEXT riempie l'intero video con tutti i caratteri disponibili sui computer MSX. Successivamente dalla linea 140 alla linea 170 vengono colorati, a gruppi di otto, i diversi caratteri visualizzati. Infine viene eseguita ripetutamente la linea 190, che attiva la routine in linguaggio macchina facendo lampeggiare l'intero schermo a più colori.

GRANDE INCHIESTA MSX 1986



Abbiamo bisogno anche del tuo consiglio e del tuo parere per migliorare MSX Computer Magazine e seguitare a farne una rivista che corrisponda sempre e sempre di più alle tue aspettative ed ai tuoi desideri. Compila con sincerità il questionario e spediscilo (corredato del tuo nome ed indirizzo solo se lo vuoi) a MSX Computer Magazine, c.so Vitt. Emanuele 15, 20122 Milano. Grazie!

Ciao, quanti anni hai?
.....

Dovi vivi?

- nord centro sud
 città provincia

Marca e tipo del tuo computer MSX:
.....

Sei uno studente o lavori?
.....

Da quanto tempo leggi MSX Computer Magazine?
.....

Lo compri sempre?

- sì no

Se no, perché?
.....

Quali altre riviste del settore leggi?
.....

Secondo te, MSX Computer Magazine tratta in modo soddisfacente tutti i temi relativi agli MSX?

- sì no

Quali argomenti ti interessano di più?
.....
.....

Quale giudizio complessivo dai dei programmi sulla cassetta di MSX Computer Magazine?

- discreto buono
 ottimo insufficiente

Qual è il genere di programma che preferiresti trovare sulla cassetta?

- giochi utility
 educational avventure

Hai mai provato ad "inventare" un programma tutto tuo?

- sì no

Indica tre articoli che vorresti veder presto pubblicati sulla rivista
.....
.....

Indica tre programmi che vorresti veder pubblicati presto sulla nostra cassetta
.....
.....

Quali sono gli articoli pubblicati che ti sono piaciuti di più?
.....
.....

Quali sono i programmi già pubblicati che ti sono piaciuti di più?
.....
.....

Cosa manca secondo te su MSX Computer Magazine?
.....
.....

Ti piace partecipare ai concorsi con premi?

- sì no

Hai altri hobby oltre al computer? Quali?
.....
.....

Qual è la tua critica più feroce ad MSX Computer Magazine?
.....
.....

E il miglior complimento?
.....
.....

Sei abbonato?

- sì no

Se no, come mai?
.....
.....

Se tu dovessi dare un voto di merito da 0 a 10 ad MSX Computer Magazine, che voto daresti?
.....
.....

**SCRIVI QUI
IL TUO NOME**

(solo se vuoi)

NOME..... COGNOME.....

VIA..... CITTÀ.....

QUALUNQUE
COMPUTER
TU
ABBIA...

PUOI ENTRARE
GRATIS nel
MODEM CLUB!



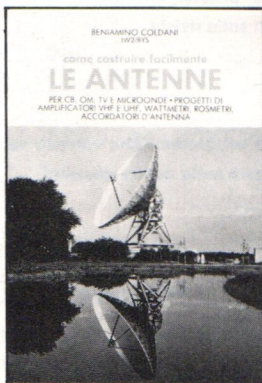
Basta telefonare (sempre disponibili
24 ore su 24) allo (02) 70.68.57.
Maggiori informazioni
telefonando allo (02) 70.63.29
solo il giovedì dalle 15 alle 18.



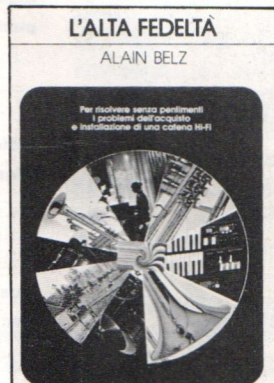
PER LA TUA BIBLIOTECA TECNICA



Conoscere l'Elettronica
Tutta l'elettronica digitale,
semplicemente, con
esperimenti e montaggi.
Lire 8.000



Le Antenne
Dedicato agli appassionati
dell'alta frequenza: come
costruire i vari tipi di
antenna, a casa propria.
Lire 6.000



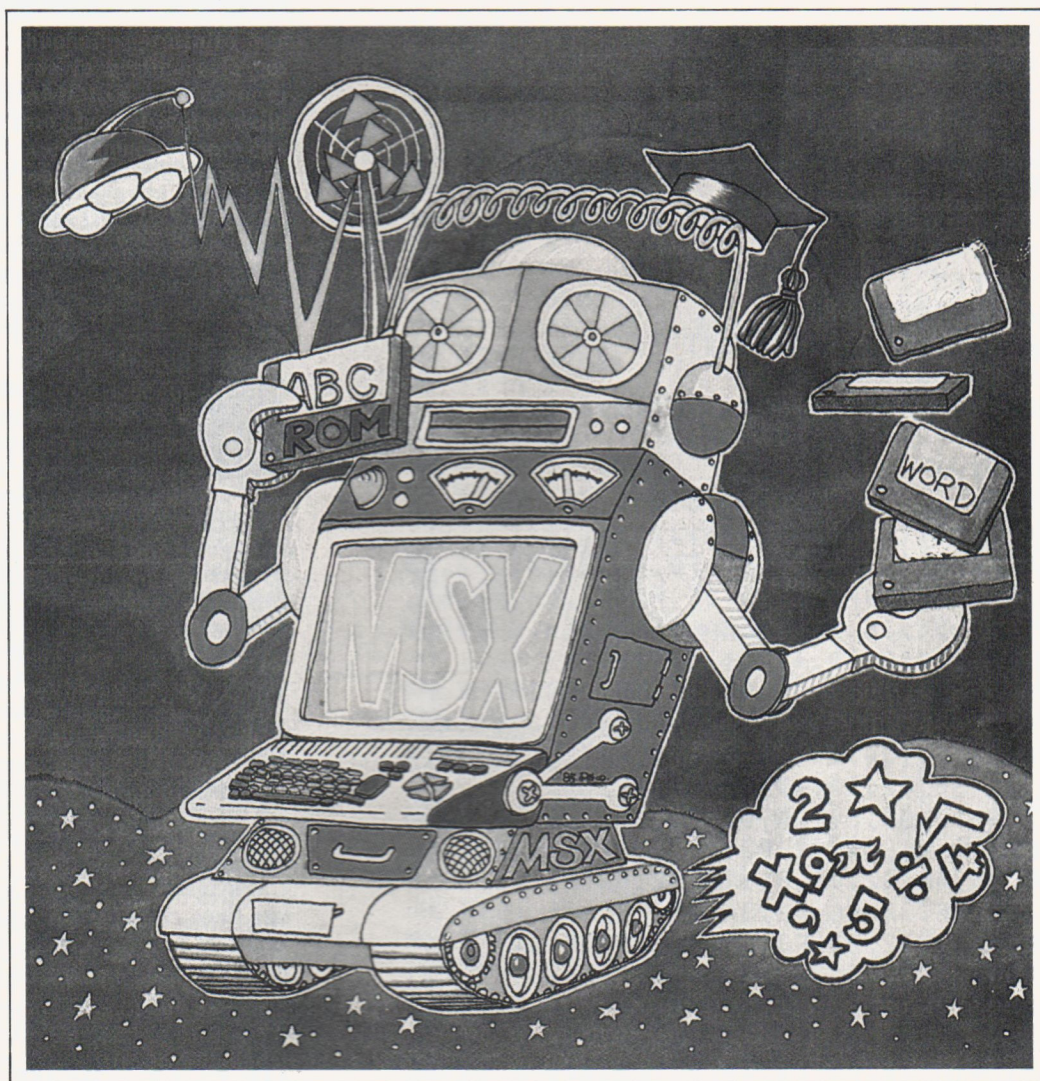
Alta Fedeltà
Per risolvere senza pentimenti
i problemi dell'acquisto
e dell'installazione di una
catena hi-fi.
Lire 3.000

Per ogni ordine inviare vaglia postale ordinario a Electronica 2000, C.so Vitt. Emanuele 15, 20122 Milano.

G R A P H I C

TURBO DRAW

VELOCIZZIAMO L'MSX BASIC PER DISEGNARE PUNTI E LINEE CON DUE MINIPROGRAMMI IN ASSEMBLER



Le routine che stiamo per presentarvi servono per disegnare punti e linee in un tempo molto minore rispetto alle equivalenti istruzioni Basic PSET e LINE.

Sappiamo che l'MSX-Basic offre differenti istruzioni grafiche e tutte

ad esecuzione rapida, ma spesso si richiede nei programmi di disegnare delle figure in tempi assai ridotti tali che nemmeno l'MSX-Basic riesce a soddisfare. Ecco che allora abbiamo realizzato un piccolo programma in linguaggio macchina dalle grandi

prestazioni: disegnare punti e linee.

Per tracciare un semplice punto il programma si serve soltanto di due parametri: coordinata X e coordinata Y.

Per tracciare una linea, invece, occorrono quattro parametri: le coordi-

LISTATO BASIC

```

1 CLEAR 200,&HF22F
20 GOSUB 60000 'CARICA L/M
30 Y=&HF305:X=Y+1:Y2=X+1:X2=Y2+1 '
   X,Y,X2,Y2 SONO LE LOCAZIONI DI
   MEMORIA DELLE COORDINATE DEL PUNTO
   O DELLA LINEA DA TRACCIARE
40 '*** ESEMPIO ***
50 SCREEN 0
60 PRINT "DISEGNO UN PUNTO DI":PRINT"COO
RDINATE (100,100)":FOR N=1 TO 5000:NEXT
70 POKE X,100:POKE Y,100
80 SCREEN 2
90 A%=USR(0)
100 FOR N=1 TO 5000:NEXT
110 SCREEN 0
120 PRINT "DISEGNO UNA LINEA DAL PUNTO (
100,20) AL PUNTO (20,67)":FOR N=1 TO 500
0:NEXT
130 POKE X,100:POKE Y,20:POKE X2,20:POKE
Y2,67
140 SCREEN 2
150 A%=USR1(0)
160 FOR N=1 TO 5000:NEXT
170 GOTO 50
60000 '*****
60001 '*          TURBO-DRAW          *
60002 '*              by              *
60003 '*          ORIGINAL SOFT.     *
60004 '*****
60005 DATA 3A,06,F3,E6,F8,67,3A,05,F3,CB
,3F,CB,3F,CB,3F,57,3A,05,F3,F5,E6,F8,6F,
F1,95,6F,5C,26,00,19,3A,06,F3,E6,07,47,0
4,3E,01,0F,10,FD,47,CD,4A,00
60006 DATA B0,CD,4D,00,11,00,20,19,3E,F4
,CD,4D,00,C9,CD,30,F2,DD,21,09,F3,11,01,
01,2A,05,F3,3A,08,F3,BC,30,02,15,15,3A,0
7,F3,BD,30,02,1D,1D,62,6B,22
60007 DATA 09,F3,2A,05,F3,3A,08,F3,94,30
,02,ED,44,47,3A,07,F3,95,30,02,ED,44,4F,
B8,30,0D,AF,32,0B,F3,7A,32,0C,F3,60,51,5
8,18,0D,AF,32,0C,F3,3A,09,F3
60008 DATA 32,0B,F3,61,50,59,7C,CB,3F,6F
,7D,82,38,03,BB,38,16,93,6F,3A,06,F3,DD,
86,01,32,06,F3,3A,05,F3,DD,86,00,32,05,F
3,18,13,6F,3A,06,F3,DD,86,03
60009 DATA 32,06,F3,3A,05,F3,DD,86,02,32
,05,F3,C5,E5,D5,DD,E5,CD,30,F2,DD,E1,D1,
E1,C1,25,C8,18,BF,00,00,00,00,00,00,00,0
0
60010 RESTORE 60005:FOR N=&HF230 TO &HF3
0C:READ A$:POKE N,VAL("&H"+A$):NEXT:DEFU
SR=&HF230:DEFUSR1=&HF26C
60011 RETURN

```



nate del punto di partenza e le coordinate del punto d'arrivo. Sia per la linea che per il punto si aggiunge un altro parametro, il colore, del quale parleremo più avanti.

Tutti i parametri sopra elencati devono essere memorizzati in particolari locazioni di memoria prima di eseguire la chiamata alla routine esecutiva. I valori delle locazioni di memoria sono:

&HF306= coordinata X
 &HF305= coordinata Y
 &HF308= coordinata X2
 &HF307= coordinata Y2
 &HF267= codice colore

Supponiamo ora di aver assegnato ad USR la chiamata alla routine di tracciatura punto e ad USR1 quella di disegno di una linea. Volendo disegnare il punto (20,80) dovremo scrivere le seguenti linee di programma:

```

100 POKE &HF306,20:POKE &HF305,80
110 SCREEN 2
120 A%=USR (0)

```

..... resto del programma

Cioè bisogna prima settare le coordinate del punto da tracciare e poi eseguire la chiamata.

Proviamo ora invece a disegnare una linea che parta dal punto (10,4) e arrivi al punto (23,170). Le linee di programma che eseguono questa operazione sono:

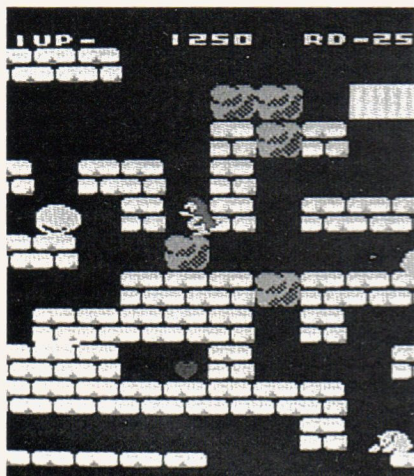
```

100 POKE &HF306,10:POKE &HF305,4
110 POKE &HF308,23:POKE &HF307,170
120 SCREEN 2
130 A%=USR1 (0)

```

..... resto del programma

Così come per il punto, anche per la linea bisogna definire i suoi parametri prima di accedere, con USR1, alla routine esecutiva. Particolare at-



tenzione deve essere fatta quando si definisce il punto di partenza e di arrivo della linea, questi non devono coincidere ma devono essere obbligatoriamente diversi. In altre parole non tentate mai di disegnare un punto con una linea! Il valore della X, sia per punti che per linee, dovrà essere compreso tra 0 e 255, mentre quello della Y tra 0 e 191. Cioè le routine non riescono a tracciare punti o linee fuoriuscenti dallo schermo, operazione invece ammessa dall'MSX-Basic.

Per quanto riguarda il colore del disegno, il programma inizialmente utilizza come colore della carta il blu (codice 4) e come colore d'inchiostro il bianco (codice 15). Comunque è possibile disegnare con qualsiasi colore; infatti basta calcolare il codice inchiostro-carta e memorizzarlo all'indirizzo &HF267 prima d'eseguire la chiamata alla routine. Il codice colore si calcola nel seguente modo:
 codice colore = codice inchiostro * 16 + codice carta.

Per esempio, volendo disegnare una linea di colore giallo chiaro (codice 11) su sfondo verde (codice 7), il codice colore risultante sarà 183 (11*16+7) e dovrà essere inserito nella routine con la seguente istruzione:

POKE &HF267,183

Per caricare il programma in linguaggio macchina utilizzate il listato Basic oppure, se disponete di un assembler, il listato Assembler. Consigliamo comunque a tutti di analizzare il listato Basic perché contiene dalla linea 50 alla linea 170 una dimostrazione di utilizzo della routine. Una volta capito come funziona l'insieme potete cancellare le linee dalla 50 alla 170 e inserire il vostro programma. È importante non tralasciare la linea 1 che riserva spazio in me-

IN ASSEMBLER

```

1
2 F230 3A06F3   PLOT:
3 F233 E6F8
4 F235 67
5 F236 3A05F3
6 F239 CB3F
7 F23B CB3F
8 F23D CB3F
9 F23F 57
10 F240 3A05F3
11 F243 F5
12 F244 E6F8
13 F246 6F
14 F247 F1
15 F248 95
16 F249 6F
17 F24A 5C
18 F24B 2600
19 F24D 19
20 F24E 3A06F3
21 F251 E607
22 F253 47
23 F254 04
24 F255 3E01
25 F257 0F   ROTATE:
26 F258 10FD   DJNZ ROTATE
27 F25A 47     LD B,A
28 F25B CD4A00  CALL 004AH
29 F25E B0     OR B
30 F25F CD4D00  CALL 004DH
31 F262 110020 LD DE,8192
32 F265 19     ADD HL,DE
33 F266 3EF4   LD A,244
34 F268 CD4D00  CALL 004DH
35 F26B C9     RET
36 F26C CD30F2  DRAW:
37 F26F DD2109F3 LD IX,D1E1
38 F273 110101 LD DE,0101H
39 F276 2A05F3 LD HL,(Y)
40 F279 3A08F3 LD A,(X2)
41 F27C BC     CP H
42 F27D 3002   JR NC,L1
43 F27F 15     DEC D
44 F280 15     DEC D
45 F281 3A07F3 L1: LD A,(Y2)
46 F284 BD     CP L
47 F285 3002   JR NC,L2
48 F287 1D     DEC E
49 F288 1D     DEC E
50 F289 62     L2: LD H,D
51 F28A 6B     LD L,E
52 F28B 2209F3 LD (D1E1),HL
53 F28E 2A05F3 LD HL,(Y)
54 F291 3A08F3 LD A,(X2)
55 F294 94     SUB H
56 F295 3002   JR NC,L3
57 F297 ED44   NEG
58 F299 47     L3: LD B,A
59 F29A 3A07F3 LD A,(Y2)
60 F29D 95     SUB L

```

SEGUE →

61	F29E	3002		JR	NC, L4
62	F2A0	ED44		NEG	
63	F2A2	4F	L4:	LD	C, A
64	F2A3	B8		CP	B
65	F2A4	300D		JR	NC, L5
66	F2A6	AF		XOR	A
67	F2A7	320BF3		LD	(E2), A
68	F2AA	7A		LD	A, D
69	F2AB	320CF3		LD	(D2), A
70	F2AE	60		LD	H, B
71	F2AF	51		LD	D, C
72	F2B0	58		LD	E, B
73	F2B1	180D		JR	L6
74	F2B3	AF	L5:	XOR	A
75	F2B4	320CF3		LD	(D2), A
76	F2B7	3A09F3		LD	A, (D1E1)
77	F2BA	320BF3		LD	(E2), A
78	F2BD	61		LD	H, C
79	F2BE	50		LD	D, B
80	F2BF	59		LD	E, C
81	F2C0	7C	L6:	LD	A, H
82	F2C1	CB3F		SRL	A
83	F2C3	6F		LD	L, A
84	F2C4	7D	LOOP:	LD	A, L
85	F2C5	82		ADD	A, D
86	F2C6	3803		JR	C, L7
87	F2C8	BB		CP	E
88	F2C9	3816		JR	C, L8
89	F2CB	93	L7:	SUB	E
90	F2CC	6F		LD	L, A
91	F2CD	3A06F3		LD	A, (X)
92	F2D0	DD8601		ADD	A, (IX+1)
93	F2D3	3206F3		LD	(X), A
94	F2D6	3A05F3		LD	A, (Y)
95	F2D9	DD8600		ADD	A, (IX+0)
96	F2DC	3205F3		LD	(Y), A
97	F2DF	1813		JR	DIS
98	F2E1	6F	L8:	LD	L, A
99	F2E2	3A06F3		LD	A, (X)
100	F2E5	DD8603		ADD	A, (IX+3)
101	F2E8	3206F3		LD	(X), A
102	F2EB	3A05F3		LD	A, (Y)
103	F2EE	DD8602		ADD	A, (IX+2)
104	F2F1	3205F3		LD	(Y), A
105	F2F4	C5	DIS:	PUSH	BC
106	F2F5	E5		PUSH	HL
107	F2F6	D5		PUSH	DE
108	F2F7	DDE5		PUSH	IX
109	F2F9	CD30F2		CALL	PLOT
110	F2FC	DDE1		POP	IX
111	F2FE	D1		POP	DE
112	F2FF	E1		POP	HL
113	F300	C1		POP	BC
114	F301	25		DEC	H
115	F302	C8		RET	Z
116	F303	18BF		JR	LOOP
117	F305	00	Y:	DEFB	0
118	F306	00	X:	DEFB	0
119	F307	00	Y2:	DEFB	0
120	F308	00	X2:	DEFB	0
121	F309	0000	D1E1:	DEFW	00
122	F30B	00	E2:	DEFB	0
123	F30C	00	D2:	DEFB	0
124				END	

moria, a partire dalla locazione 62000, per i codici macchina. Prima di passare alla descrizione del listato Assembler ricordiamo che la routine opera esclusivamente in SCREEN 2, quindi non cercate di utilizzarla in altri modi grafici.

Il listato Assembler, come già detto prima, si suddivide in due parti: quella dedicata alla tracciatura di un punto e quella relativa al disegno di una linea.

Incominciamo ad analizzare la prima parte, identificata dall'etichetta PLOT. Conosciute le coordinate del punto, bisogna calcolare la locazione di memoria a cui appartiene. Questa operazione è eseguita dalla linea 2 alla linea 19 restituendo in HL il risultato. Successivamente, linee 20-27, si calcola, in base alla coordinata X, il bit da settare nella locazione di memoria. Infine, linee 28-35, si prelevano i "punti" già settati in quella locazione di memoria, si somma il nuovo punto e si ripristina il nuovo dato in memoria aggiornando anche i colori.

La seconda parte del programma, quella dedicata al disegno di una linea e identificata dall'etichetta DRAW, è più lunga e complessa della prima. Bisogna innanzitutto dire che una linea è un insieme di punti e quindi questa routine calcola i diversi punti appartenenti alla linea e li disegna utilizzando la routine PLOT. Inizialmente il programma DRAW traccia il punto iniziale della retta, successivamente stabilisce quale direzione assume la linea: alto, basso, destra o sinistra. Questa operazione viene svolta dalla linea 37 alla linea 52; il risultato in D è -1 se la linea procede verso sinistra, 1 se verso destra. In E il valore 1 indica la direzione della linea verso il basso, -1 verso l'alto. Successivamente si calcola la distanza assoluta tra le coordinate X e Y del punto di partenza e di arrivo della retta e sulla base di questi dati si decide punto per punto quale step eseguire: orizzontale (destra o sinistra) o verticale (su o giù). Ogni punto calcolato viene disegnato utilizzando la routine PLOT.

Volendo velocizzare al massimo il disegno di una figura è consigliabile memorizzare tutte le coordinate delle rette che la compongono e poi, con un semplice programmino in L/M, trasferibile a coppie nelle locazioni X,Y,X2 e Y2 e richiamare più volte DRAW. Provate a sviluppare quanto appena detto, il risultato sarà di sicuro effetto!

ECCEZIONALE NOVITÀ

IN TUTTE LE EDICOLE

TUTTO QUEL CHE SERVE PER ENTRARE NEL FAVOLOSO MONDO DELLA COMPUTER-COMUNICAZIONE VIA TELEFONO

Hai un computer e un telefono? Questa rivista è fatta apposta per te! Tutto l'hardware e il software (su cassetta!) per comunicare via computer e telefono in Italia e in tutto il mondo. Potrai metterti in contatto con altri computer, trasmettere e ricevere posta, catturare programmi, trovare i mille amici (gli hackers!) del Modem Club International con cui scambiare programmi e informazioni... più una banca dati diretta a disposizione 24 ore... più altro ancora...

la prima rivista per computer via telefono

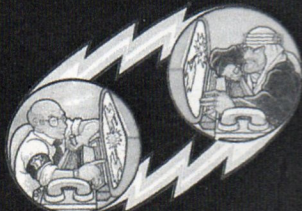
MODEM

COMPUTER MAGAZINE

N. 1 GENNAIO 1986

Sped. in abb. post. gr. III - L. 9.000

PIÙ
I NUMERI
DI TELEFONO
PER PORSI
IN
CONTATTO



CON IL
FASCICOLO,
GRATIS
IL SOFTWARE
SU
CASSETTA

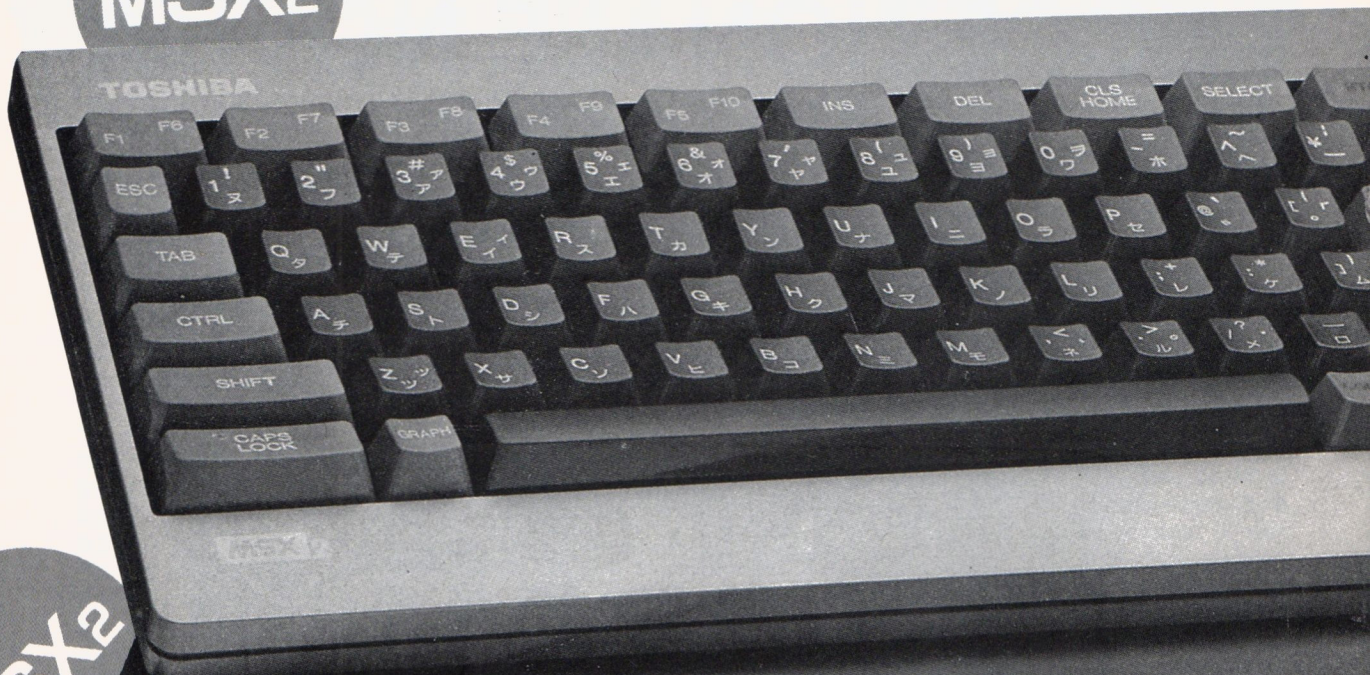
COSTRUISCI IL TUO MODEM!
MODEM CLUB
DATA HACKERS NEWS
MICROMARKET LETTORI
LE INTERFACCE UTILI
MODEM MERCATO

da casa
**BANCA
DATI**

con il software
**IN CASSETTA
OMAGGIO!**

Questo fascicolo può anche essere richiesto
con vaglia postale di Lire 10.000 ad Arcadia, c.so Vittorio Emanuele 15, 20122 Milano

MSX2



MSX2



Nell'articolo "MSX quante novità" apparso nel numero 5 di MSX Computer Magazine, sono state descritte le future innovazioni in termini di software e di hardware del mondo MSX. Tra queste novità la più interessante è stata ed è tuttora la prossima immissione sul mercato dei sistemi MSX della seconda generazione. Ora, dopo avere avuto l'occasione di poter provare una di queste nuove macchine, siamo in grado di darvi una maggiore descrizione su quelle che sono le migliorie rispetto agli MSX 1.

Il Toshiba HX-23, che nella versione europea sarà denominato HX-33 (quello della nostra prova era una versione giapponese), assomiglia co-

me estetica al fratello minore HX-22; però un piccolo 2 a lato della scritta "MSX", in alto a destra, ne conferma immediatamente la diversità dal secondo. Continuando l'osservazione ci si accorge che l'HX-23 ha notevoli capacità d'espansione. Infatti sul retro del computer vi sono differenti prese riguardanti la comunicazione, il video ed il suono. Per quanto riguarda l'interfacciamento con altre periferiche troviamo il connettore Amphenol per il collegamento della stampante parallela. Questa uscita è presente in tutti gli MSX 2 mentre non lo è per gli MSX 1. A lato dell'uscita parallela vi è l'interfaccia RS-232. Numero-se le possibilità di visualizzare l'immagine inviata dal computer. Tro-

viamo infatti l'uscita RF per il televisore e le uscite per monitor RGB e video composito. Una caratteristica particolare di questo MSX 2 è quella di avere l'uscita sonora separata su due prese, più precisamente su quella di sinistra esce il secondo canale, su quella di destra il terzo mentre su entrambe il primo. Un interruttore posto a lato delle uscite sonore consente di modificare l'uscita da stereo a mono. Sul lato destro del computer troviamo invece due prese per i joystick e quella per il registratore. Gli slot per le espansioni sono due.

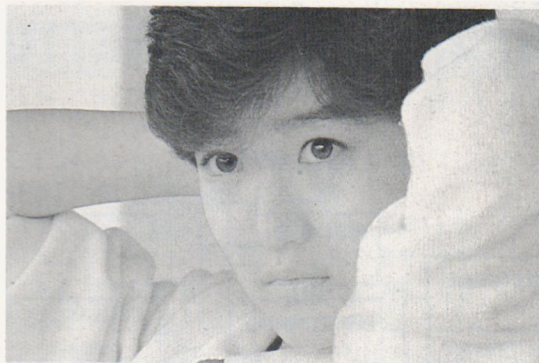
Passiamo ora a distinguere le diversità hardware tra MSX 1 ed MSX 2. Entrambi i computer hanno come microprocessore lo Z80, come gene-

H A R D W A R E

TOSHIBA MSX 2

**PIÙ GRAFICA, PIÙ MEMORIA
PIÙ COMANDI!**

di E. DASSI



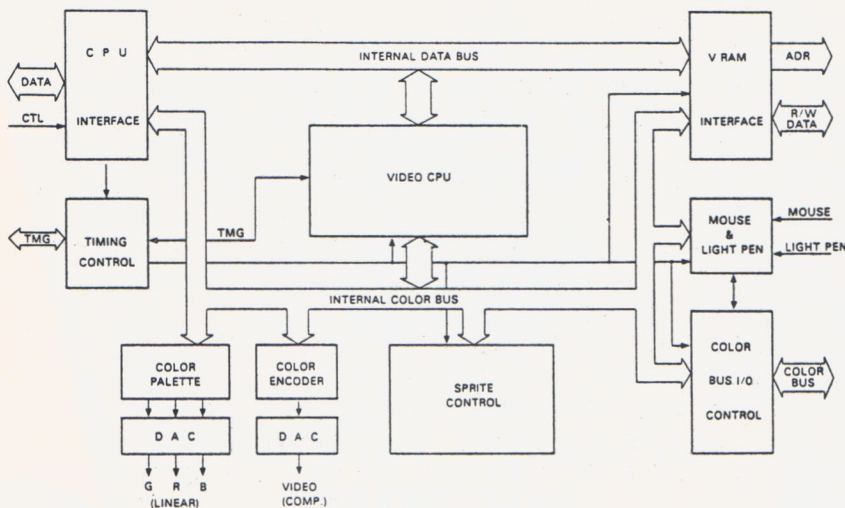
ratore sonoro l'AY-3-8910 (che nell'MSX 2 potrà essere interfacciato con il chip opzionale MSX Audio: un vero e proprio sintetizzatore FM a nove voci!) mentre per quanto riguarda il processore video il primo si basa sul noto TMS 9918A ed il secondo sul V9938: un TMS 9918A espanso a 64 piedini con notevoli capacità grafiche in più rispetto al fratello minore. Questo nuovo integrato necessita, per lo standard MSX 2, di 64 Kb di memoria video ma il Toshiba HX-23 ne ha ben 128 Kb offrendo così le massime prestazioni grafiche di un MSX della seconda generazione. Anche la ROM è aumentata passando da 32 a 48 Kb. L'aumento della ROM è stato necessario per poter inserire i nuovi

comandi dei quali parleremo più avanti. Infine la memoria RAM è di 64 Kb, quantità minima per un MSX 2, che può essere espansa, con un chip opzionale, fino a 4 Mb. In più tutti gli MSX 2 hanno un orologio interno completo di batteria di back-up così, anche quando il computer è spento, l'ora non viene persa. Vi sono altre due opzioni hardware: la prima è un digitalizzatore di immagini captate con una telecamera, la seconda il superimpose, un dispositivo in grado di sovrapporre immagini computerizzate a quelle video-registate.

L'aumento della ROM a 48 Kb è stato necessario, come detto prima, per poter aggiungere nuovi comandi Basic in grado di sfruttare le nuove

risorse hardware. Iniziamo con la grafica che è la sezione più evoluta di questi nuovi MSX. Ora il comando SCREEN può selezionare molti più formati del display rispetto all'MSX 1, precisamente nove. Con il comando SCREEN o si seleziona il primo modo testo uguale a quello degli MSX 1 (40 colonne per 24 righe) ma utilizzando il comando WIDTH, con operando superiore a 40, automaticamente il display visualizza 80 colonne per 24 o 26 linee, con la possibilità di utilizzare 4 colori scelti da una tavolozza di 512 tonalità. Il modo testo 2 (SCREEN 1) è uguale a quello degli MSX 1 così come il modo grafico 1 (SCREEN 2) e 2 (SCREEN 3). Con SCREEN 4 si incominciano ad utiliz-

PROCESSORE VIDEO



zare le nuove prestazioni grafiche degli MSX 2. In questo modo il display visualizza ancora 256x192 pixels, i colori sono ancora 16 però gli sprite visualizzabili su di una linea non sono

più 4 ma 8. Inoltre ogni linea di ogni sprite può avere un proprio colore. Il comando SCREEN 5 non comporta notevoli differenze rispetto al primo tranne che i 16 colori utilizzabili sono

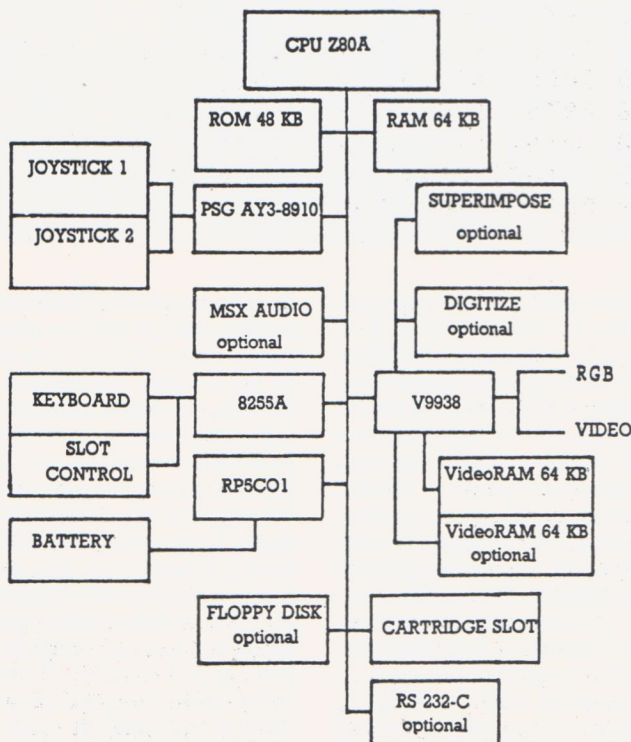
scelti da una tavolozza di 512 e che la risoluzione è di 256x212 pixels. Il successivo display grafico (SCREEN 6) ha una risoluzione di 512x212 pixels ed i colori sono ora soltanto 4, scelti sempre da una palette di 512. Gli ultimi due formati del display (SCREEN 7 e SCREEN 8) sono attivabili solo se la VRAM è di 128 Kb. Con SCREEN 7 la risoluzione è sempre di 512x212 pixels ma i colori disponibili contemporaneamente sono 16. Con SCREEN 8, invece, la risoluzione si riduce a 256x212 pixels ma i colori visualizzabili sono ben 256 consentendo di realizzare disegni artistici davvero eccezionali. Vi è da notare che in questi nuovi display grafici il colore di ogni punto è indipendente da quello degli altri, mentre negli MSX 1 potevano esserci solo 2 colori ogni otto pixels orizzontali. Inoltre attivando in linguaggio macchina una speciale funzione del processore video, è possibile raddoppiare la risoluzione verticale portando così la grafica a 512x424 pixel!

Sempre a riguardo della grafica il Basic offre nuovi comandi: SET PAGE e COPY. Il primo seleziona su video una delle diverse (memoria permettendo) pagine grafiche, mentre il secondo attiva una particolare funzione del V9938 capace di trasferire zone di VRAM in altre o di VRAM ad array e viceversa. Il risultato è un insieme di effetti grafici veramente spettacolari e velocissimi.

Tra le nuove istruzioni troviamo GET TIME e SET TIME per avere in tempo reale l'ora e per aggiornarla; GET DATE e SET DATE per ottenere la data e per settarla. Nella memoria dell'orologio è possibile memorizzare costantemente una password (SET PASSWORD) per l'eccesso riservato al sistema, la scritta iniziale, i colori e così via.

Come abbiamo detto la RAM minima per un MSX 2 è di 64 Kb ma per il Basic ne sono riservati i soliti 32 Kb, che poi si riducono a 28815 bytes per la programmazione. Gli altri 32 Kb sono gestiti come RAMDISK, cioè vengono utilizzati come supporto di memoria di massa ad altissima velocità. È chiaro che una volta tolta l'alimentazione al computer i dati della RAMDISK spariranno, quindi a fine lavoro dovranno essere trasferiti su disco o cassetta, però la gestione di questa memoria si rivela oltre che veloce anche facile. Infatti per ricevere o trasferire i dati alla RAMDISK vanno utilizzati gli stessi comandi per il registratore e per il disco: SA-

MSX2, SUA ARCHITETTURA



HAI TUTTI I NUMERI ARRETRATI?

Di questo fascicolo (N. 1) sono disponibili la sola cassetta e le fotocopie delle istruzioni dei programmi.



Di questo fascicolo (N. 2) sono disponibili la sola cassetta e le fotocopie delle istruzioni dei programmi.



**PUOI RICEVERLI
DIRETTAMENTE
A CASA!**

Basta inviare vaglia postale ordinario di lire 10.000 specificando sul vaglia stesso quale fascicolo desideri ed i tuoi dati chiari e completi. Spedisci ad Arcadia s.r.l., c.so Vitt. Emanuele 15, 20122 Milano. Spedizioni rapide!

*Dove non arriva il libro di testo arrivano
le «Garzantine»*

*le piccole-grandi enciclopedie che non fanno perdere tempo
pratiche, economiche, in un solo volume*

- pronte nelle risposte • essenziali e esaurienti nelle definizioni
- chiare nelle spiegazioni • esperte nelle difficoltà



una per ogni materia

La Nuova Enciclopedia della Letteratura · La Nuova Enciclopedia Geografica · La Nuova Enciclopedia della Musica · Enciclopedia di Filosofia e epistemologia · logica formale · linguistica · psicologia · psicoanalisi · pedagogia · antropologia culturale · teologia · religioni · sociologia · Enciclopedia del Diritto e dell'Economia e di scienza delle finanze · statistica · matematica per le scienze sociali · informatica · marketing · management · contabilità aziendale · matematica finanziaria e attuariale · tecnica bancaria · borsa

per consultazioni lampo La Nuova Enciclopedia Universale

oggi nuovi con migliaia di parole nuove Il Nuovo Dizionario Italiano Il Nuovo Dizionario Inglese Il Nuovo Dizionario Francese

GARZANTI

ARCHIVIO SU RAMDISK

VE, LOAD, OPEN, CLOSE, ecc. L'unica differenza è che ora l'identificatore del supporto di massa non è più "CAS" oppure "A" o "B" ma "MEM". Vi sono però alcuni comandi aggiuntivi che consentono di eseguire altre operazioni sulla RAMDISK. CALL MEMINI formatta la RAMDISK cancellando ogni cosa precedentemente memorizzata. CALL MFILES da' una lista dei nomi dei files memorizzati nella RAMDISK e la quantità di memoria ancora libera. CALL MKILL consente di cancellare un file e CALL MNAME di assegnare ad un file già esistente un nuovo nome.

Al fine di evidenziare le prestazioni della RAMDISK e quelle grafiche abbiamo voluto pubblicare un programmino che visualizza su 80 colonne un mini archivio gestito con la RAMDISK. Il programma è un adattamento del listato apparso nell'articolo "L'archivio elettronico" nel primo numero di MSX Computer Magazine. La prima istruzione è CALL MEMINI la quale inizializza l'archivio. Le successive operazioni di gestione archivio utilizzano nel comando OPEN l'identificatore MEM.

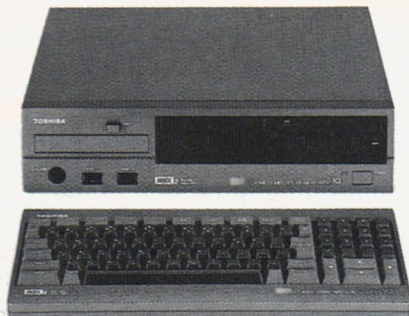
Concludiamo questa prova informandovi che la Toshiba affiancherà all'HX-23 un altro MSX 2: l'HX-34, un computer dal design più professionale in quanto avrà la tastiera, completa di tastierino numerico, separata dalla consolle la quale conterrà un disk drive 3.5 pollici da 720 Kb formattati con la possibilità dell'inserimento di un secondo disco; due slot per le espansioni e nella tastiera un tasto speciale per l'hard copy del video su stampante. Il sistema riconosce la pressione di questo tasto in ogni momento, quindi è possibile trasferire su carta immagini di videogiochi, videocomunicazioni, ecc.

Ma ritorniamo all'HX-23 che, come detto all'inizio, nella versione europea sarà denominato HX-33. Melchioni non ha ancora precisato la data di commercializzazione anche se si prevede il periodo giugno-settembre '86. Per quanto riguarda il prezzo l'HX-23 dovrebbe costare, IVA compresa, sotto al milione.

La nostra impressione sulla macchina ed in generale sugli MSX 2 è stata molto buona, soprattutto per quanto riguarda la grafica e la gestione della RAMDISK. Lo è ancor di più se pensiamo che il discorso dello standard MSX continua non solo con la seconda generazione ma anche tra questa e la precedente.

```
5 CALL MEMINI
10 SCREEN 0:WIDTH 80:MAXFILES=1
20 CLS:LOCATE 20,2:PRINT "MENU"
30 ON KEY GOSUB 110,400,700
40 LOCATE 15,5:PRINT "F1=IMMETTI INDIRIZZ
20"
50 LOCATE 15,7:PRINT "F2=SCRIVI INDIRIZZ
0"
60 LOCATE 15,9:PRINT "F3=FINE PROGRAMMA"
70 KEY (1) ON:KEY (2) ON:KEY (3) ON
80 GOTO 80
110 OPEN "MEM:ADDRESS" FOR OUTPUT AS#1
115 CLS
120 LOCATE 10,5:INPUT "NOME";A$
130 LOCATE 10,7:INPUT "INDIRIZZO";B$
140 LOCATE 10,9:INPUT "CITTA\ ";C$
150 LOCATE 10,11:INPUT "TELEFONO";D$
160 LOCATE 10,20:INPUT "DATI CORRETTI (S
/N)";F$
170 IF F$="N" OR F$="n" THEN 115
180 PRINT #1,A$;",";B$;",";C$;",";D$
190 LOCATE 10,22:INPUT "ALTRO INDIRIZZO
(S/N)";F$
200 IF F$="n" OR F$="N" THEN CLOSE#1:RET
URN 20
210 GOTO 115
400 OPEN "MEM:ADDRESS" FOR INPUT AS#1
415 CLS
420 IF EOF(1) THEN CLOSE#1:RETURN 20
430 INPUT #1,A$,B$,C$,D$
440 PRINT "NOME      ";A$:PRINT
450 PRINT "INDIRIZZO:";B$:PRINT
460 PRINT "CITTA\    ";C$:PRINT
470 PRINT "TELEFONO  ";D$
480 LOCATE 10,22:INPUT "VUOI UN ALTRO IN
DIRIZZO (S/N)";F$
490 IF F$="N" OR F$="n" THEN CLOSE#1:RET
URN 20
500 GOTO 415
700 CLS:PRINT "FINE PROGRAMMA"
710 END
```

Programma archivio 80 colonne.
In basso l'HX-33, a sinistra l'HX-34.



RIVISTA PROGRAMMI PER MSX

MSX Computer Magazine

7



Per caricare digitare CLOAD

SOMMARIO

LATO A:

- BOMBER
- CYCLE
- NAVY
- MUSICOMPOSER
- LEOPARD II

LATO B:

- NETWORK
- BIS
- HALLEY
- DUPLICATORE
- BOMB SHELL

QUESTA CASSETTA
È DI

NOME _____

COGNOME _____

VIA _____

N. _____

CITTA _____

MSX COMPUTER MAGAZINE



NOVITÀ

UN LOOK
COLORATO
PER LA TUA
CASSETTA



RITAGLIA
LUNGO
IL BORDO
SEGNATO
IN NERO
E
PIEGA
SEGUENDO
IL
TRATTEGGIO
INDICATO

PERSONALIZZA
LA
CASSETTA
CON IL
TUO NOME

MSX COMPUTER
MAGAZINE
PER LA TUA
SOFT-TECA