

Personal Computer
mz-700

**MANUALE
DELL'UTENTE**



SHARP

This apparatus complies with requirements of EEC directive 76/889/EEC.

Das Gerät stimmt mit den Bedingungen der EG-Richtlinien 76/889/EWG überein.

Cet appareil répond aux spécifications de la directive CCE 76/889/CCE.

Dit apparaat voldoet aan de vereisten van EEG-reglementen 76/889/EEG.

Apparatet opfylder kravene i EF direktivet 76/889/EF.

Quest'apparecchio è stato prodotto in conformità alle direttive CEE 76/889/CEE.

Personal Computer
mz-700

Manuale Dell'utente

AVVERTENZA

Questo manuale è stato scritto per i computer personali della serie MZ-700 e per l'interprete BASIC che é adattato al MZ-700.

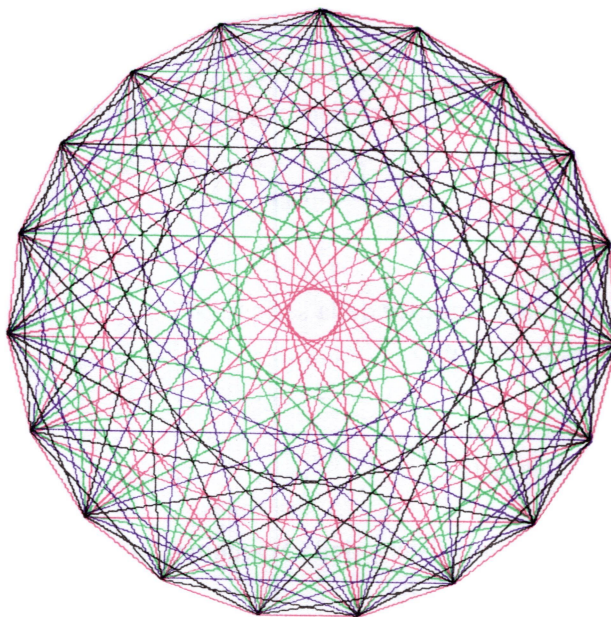
- (1) L'intero sistema software dei computer della serie MZ-700 é impacchettato in struttura software (cassette, ecc.) in formato "file". Il contenuto dell'intero sistema software e il materiale presentatò in questo manuale sono soggetti a modifiche senza preavviso per motivi di miglioramento del prodotto e altre ragioni. Occorre inoltre fare attenzione a che il numero di versione del "file" del sistema software usato corrisponda a quello indicato in questo manuale.
- (2) L'intero sistema per il computer personale della serie sharp MZ-700 è stato sviluppato dalla Società Sharp e quindi tutti i diritti su tale software sono riservati. E' proibita la riproduzione del sistema software e del contenuto di questo libro.
- (3) Il computer e il contenuto di questo manuale sono stati rigorosamente controllati per completezza e correttezza; tuttavia se doveste andare incontro a dei problemi durante l'uso o avere delle domande a cui questo manuale non può rispondere, non esitate nel consultare l'addetto Sharp per l'assistenza. A scanso di equivoci, teniamo a precisare che la Società Sharp e i suoi rappresentanti non assumeranno responsabilità per smarrimenti o danni occorsi durante il funzionamento o l'uso di questo apparecchio.

Prefazione

Congratulazioni per l'acquisto di un computer personale della serie Sharp MZ-700. Prima di usare il computer, leggete e accertatevi di aver compreso le procedure d'uso che sono descritte in questo manuale. Le caratteristiche dell'apparecchio e le procedure d'uso sono descritte nei Capitoli 1 e 3, perciò leggete per prima questi capitoli.

L'intero software dei computer della serie MZ-700 è distribuito nelle cassette.

La cassetta di cui è provvista il computer contiene il BASIC 1Z-013B, un interprete BASIC ad alto livello che consente la programmazione nel linguaggio BASIC e rende possibile utilizzare a pieno le capacità del MZ-700. L'interprete BASIC 1Z-013B e le procedure per l'uso sono descritti in questo manuale nei particolari.



DISEGNO OTTENUTO PER MEZZO DELLA STAMPANTE "PLOTTER" A COLORI.

MZ-700 Manuale Dell'Utente

INDICE

Capitolo 1 Il Mondo del Computer Personale della Serie MZ-700

1. 1	Caratteristiche della serie MZ-700	8
1. 2	Uso di questo manuale	10
1. 3	Introduzione ai calcolatori	11

Capitolo 2 BASIC

2. 1	Introduzione alla programmazione in BASIC	14
2. 2	Profilo del BASIC	19
2. 3	Comandi e istruzioni BASIC frequentemente usati	26
2. 4	Funzioni incorporate	70
2. 5	Funzioni di stringa	75
2. 6	Istruzioni per la visualizzazione a colori	79
2. 7	Comandi per la stampante-plotter a colori	82
2. 8	Istruzioni di controllo dei programmi linguaggio macchina	91
2. 9	Istruzioni I/O	95
2. 10	Altre istruzioni	96

Capitolo 3 Uso dell'MZ-700

3. 1	Aspetto dei personal computer della serie MZ-700	104
3. 2	Collegamento all'unità video	106
3. 3	Registratore dati	108
3. 4	Stampante-plotter a colori	110
3. 5	Funzionamento dei tasti	114

Capitolo 4 Hardware (Componenti fisiche della Macchina)

4. 1	Diagramma del sistema MZ-700	122
4. 2	Configurazione di memoria	123
4. 3	I/E nella mappa della memoria (\$E000 ~ \$E008)	130
4. 4	Sistema di segnali della V-RAM colori	133
4. 5	Schemi circuitali dell'MZ-700	134

Capitolo 5 Comandi e Sottoroutine Monitor

5. 1	Comandi monitor	146
5. 2	Funzioni ed uso dei comandi monitor	147
5. 3	Sottoroutine monitor	151

APPENDICI

A. 1	Tadella codici	156
A. 2	Specifiche dei calcolatori serie MZ-700	161
A. 3	Lista messaggi errore BASIC	163
A. 4	Gruppo di istruzioni Z80A	164
A. 5	Lista di assemblaggio programma monitor	168
A. 6	Codici di controllo della stampante-plotter a colori	202
A. 7	Note relative al funzionamento	205

INDICE

[COMANDI BASIC] () è formato abbreviato.

A

ABS	70
ASC	77
ATN	70
AUTO	(A.) 29
AXIS	(AX.) 89

B

BYE	(B.) 33
-----------	--------------

C

CHRS	77
CIRCLE	(CI.) 90
CLOSE	(CLO.) 67
CLR	58
COLOR	(COL.) 79
CONSOLE	(CONS.) 98
CONT	(C.) 32
COS	70
CURSOR	(CU.) 59

D

DEF FN	54
DEF KEY	55
DELETE	(D.) 29
DIM	54

E

END	(E.) 57
EXP	70

F

FOR~NEXT	(F. ~N.) 45
----------------	------------------

G

GET	41
GOSUB	
~RETURN	(GOS.~RET.) 47
GOTO	(G.) 46
GPRINT	(GP.) 88

H

HSET	(H.) 88
------------	--------------

I

IF ERL	97
IF ERN	96
IF~GOSUB	(IF~GOS.) 51
IF~GOTO	(IF~G.) 51
IF~THEN	(IF~TH.) 48
INP	95
INPUT	(I.) 40
INPUT/T	(I./T) 67
INT	70

J

K

KEY LIST	(K. L.) 33
----------------	-----------------

L

LEFT\$	76
LEN	75
LET	34
LIMIT	(LIM.) 92
LINE	85
LIST	(L.) 30
LIST/P	(L. /P) 84
LN	70
LOAD	(LO.) 26
LOG	70

M

MERGE	(ME.) 30
MID\$	76
MODE GR	(M. GR) 83
MODE TL	(M. TL) 83
MODE TN	(M. TN) 83
MODE TS	(M. TS) 83
MOVE	87
MUSIC	(MU.) 64

N

NEW 30

O

ON ERROR

GOTO (ON ERR. G.) 96

ON~GOSUB (ON~GOS.) 53

ON~GOTO (ON~G.) 52

OUT 95

P

PAGE 84

PAI 70

PCOLOR (PC.) 83

PEEK 93

PHOME (PH.) 87

PLOT OFF (PL. OFF) 98

PLOT ON (PL. ON) 98

POKE 92

PRINT (?) 35

PRINT USING (?USI.) 36

PRINT/P (?/P) 84

PRINT/T (?/T) 67

PRINT [a, β] (? [a, β]) 80**Q****R**

RAD 70

READ~DATA (REA. ~DA.) 42

REM 56

RENUM (REN.) 31

RESET 61

RESTORE (RES.) 44

RESUME (RESU.) 97

RIGHT\$ 76

RLINE (RL.) 86

RMOVE (RM.) 87

RND 71

ROPEN (RO.) 67

RUN (R.) 32

S

SAVE (SA.) 27

SET 61

SGN 70

SIN 70

SIZE 97

SKIP 84

SPC 61

SQR 70

STOP (S.) 57

STR\$ 78

T

TAB 60

TAN 70

TEMPO (TEM.) 66

TEST (TE.) 84

TI\$ 58

U

USR (U.) 93

V

VAL 78

VERIFY (V.) 28

W

WOPEN (W.) 67

X**Y****Z**

IL MONDO DEL COMPUTER PERSONALE DELLA SERIE MZ-700

Capitolo 1



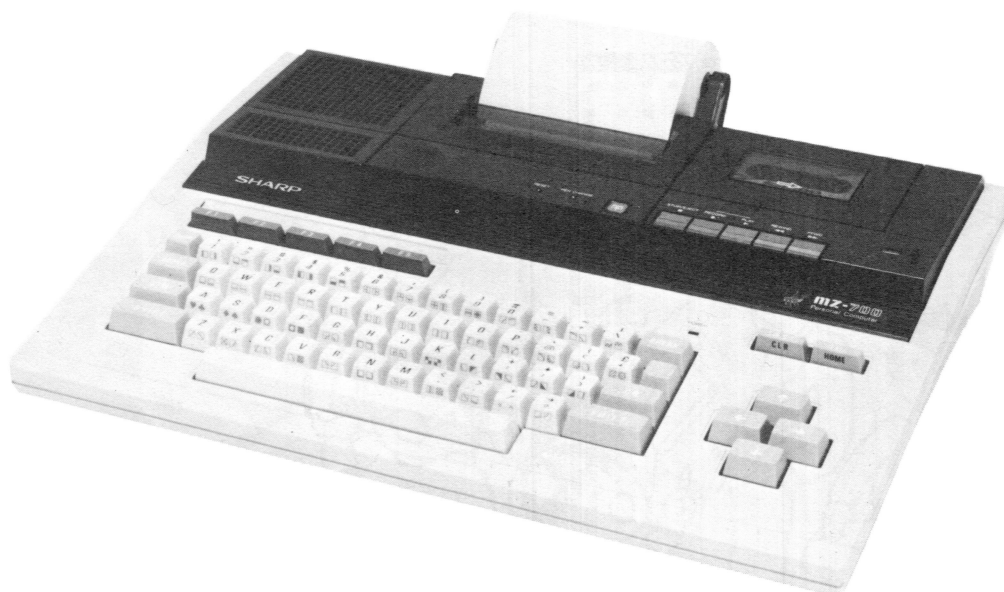
1.1 Caratteristiche della serie MZ-700

Nello spazio di pochi decenni, i calcolatori hanno subito una trasformazione incredibile; dai mostri giganteschi dal peso di parecchi milioni di chilogrammi che erano, sono divenuti delle apparecchiature compatte e poco costose che possono essere usate pressoché da chiunque. La serie Sharp MZ-700 è una rappresentanza di queste macchine. I calcolatori non sono più di dominio esclusivo di poche persone speciali; il mondo dell'informatica è aperto a tutti ed ora chiunque può impadronirsi della tecnica dell'informatica in modo autodidattico.

Le persone comunicano mediante parole ed espressioni.

Viceversa i calcolatori della serie Sharp MZ-700 comunicano attraverso un normale apparecchio televisivo o una stampante speciale. Si può usare qualsiasi televisione, sia a colori che in bianco e nero oppure si possono usare degli schermi video speciali se si vuole una maggiore risoluzione e nitidezza; vi sorprenderà la bellezza ottenuta con questi schermi.

E' possibile collegare un registratore a cassette ai calcolatori della serie Sharp MZ-700 per registrare i programmi usati per comandare il calcolatore. Quando si desidera una registrazione stampata di tali programmi o dei risultati dell'elaborazione del calcolatore, è possibile farlo su di una stampante-plotter quadricromatica, compatta, di design elegante.



MZ-731

Nota: Nel corso di questo manuale, il termine “MZ-700” verrà usato per indicare uno qualsiasi dei calcolatori della serie MZ-700 (MZ-711, MZ-721 ed MZ-731).



MZ-721



MZ-711

1.2 Uso di questo manuale

Prima che cominciamo a parlare della programmazione, perché non cerchiamo di giocare un po' con l'MZ-700? Siamo certi che è quello che volete fare invece di aspettare sino ad aver letto questo libro. Per far ciò, leggete "Uso dell'MZ-700" nel capitolo 3 (dovete leggere solo le parti che si riferiscono al modello che state usando). Collegate l'MZ-700 ad una televisione, leggete la spiegazione delle procedure per l'uso della tastiera, ed imparate quali caratteri vengono ottenuti con la pressione di ciascun tasto.

Se usate per la prima volta l'MZ-700, leggete i capitoli 1 e 2 nell'ordine. Prima, può darsi che troviate difficile afferrare il significato dei diversi comandi del linguaggio di programmazione BASIC; tuttavia, anche se non capite le spiegazioni, seguite gli esempi che vengono dati. In questo modo, svilupperete gradualmente l'idea di che cos'è il BASIC.

Potete saltare le parti del capitolo 2 che cominciano con il punto 2. 8 "Istruzioni di Controllo dei Programmi Linguaggio Macchina"; tuttavia, queste sezioni si riveleranno utili quando avrete imparato completamente la programmazione con il BASIC, o se desiderate conoscere meglio il funzionamento interno dell'elaboratore.

Se avete usato l'MZ-80K, noterete che i comandi e le istruzioni del BASIC per l'MZ-700, vengono usati nello stesso modo della famiglia SP-5025, per cui l'MZ-700, può essere usato praticamente nello stesso modo dell'MZ-80K. La principale differenza tra i due consiste nelle istruzioni per i colori (applicabili sia allo schermo televisivo che alla stampante-plotter a colori) che sono state aggiunte; tuttavia, vi risulterà facile impararle leggendo il punto 2. 6 "Istruzioni per la visualizzazione a colori" ed il punto 2. 7 "Comandi per la stampante-plotter a colori". Fatto ciò, sarete subito affascinati dalla potenza del BASIC espanso.

Questo manuale tratta anche "Uso dell'MZ-700" (capitolo 3), una sezione di consultazione dal titolo "Hardware" (componenti fisiche della macchina) (capitolo 4), "Comandi e Sottoroutine Monitor" (capitolo 5), e delle appendici con altre informazioni.

Ora proseguiamo per imparare tutto il possibile sul l'MZ-700. Ci auguriamo che questo manuale si riveli utile.

1.3 Introduzione ai calcolatori

1.3.1 Cos'è il BASIC?

Le persone usano dei linguaggi per comunicare tra di loro, ed anche per comunicare con i calcolatori, si usano dei linguaggi particolari. Il BASIC è uno di questi linguaggi.

Beginner's All-purpose Symbolic Instruction Code

Così come gli esseri umani usano delle lingue come l'inglese, il francese, il tedesco ed il giapponese per comunicare, anche per comunicare con i calcolatori esistono molte lingue diverse. Tra queste vi sono il BASIC, il FORTRAN, il COBOL ed il PASCAL. Il BASIC è il linguaggio la cui struttura è più vicina a quella delle lingue usate dalle persone.





1.3.2 Caricamento del BASIC nell'MZ-700

Il linguaggio BASIC deve essere caricato nell'MZ-700 prima che questo possa essere usato per qualsiasi lavoro. Nella confezione dell'MZ-700 è stato incluso un nastro a cassetta contenente questo linguaggio. "Insegniamo" ora il linguaggio al calcolatore; le procedure per questa operazione vengono descritte qui di seguito. (La spiegazione parte dal presupposto che usiate un MZ-731; tuttavia, le procedure sono fondamentalmente uguali per tutti i calcolatori della serie MZ-700).


- (1) Collegate il video come descritto a pagina 106.
- (2) Accendete la macchina mediante l'interruttore di alimentazione posto sul retro del calcolatore.
- (3) Sullo schermo vengono visualizzati i caratteri seguenti e compare un quadratino lampeggiante. Questo quadratino viene chiamato cursore.


```

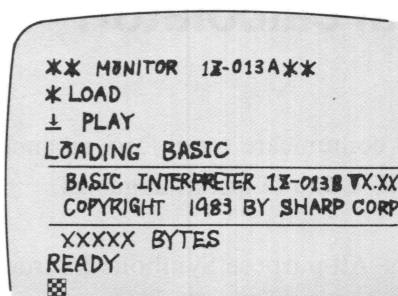
**  MON I T O R 1 Z - Ø 1 3 A  **
*
  |
  | Cursore
```

- (4) Inserite la cassetta contenente il linguaggio BASIC nel registratore del calcolatore.
- (5) Battete la parola  e premete il tasto ^{*1}. Dopo aver fatto ciò, sullo schermo compare il messaggio  PLAY.
- (6) Premete il tasto  del registratore; la cassetta comincia a girare ed ha inizio il caricamento del BASIC.
- (7) Dopo aver completato il caricamento, viene visualizzato il messaggio READY ed il cursore comincia nuovamente a lampeggiare.

Nota:

*1  . . . Questa è l'istruzione per caricare i programmi o i dati dalla cassetta.

*2  Questo è il tasto del ritorno a capo, e viene essenzialmente utilizzato per indicare il completamento dell'immissione di un'istruzione.



Questo completa il caricamento del programma BASIC. Potete “parlare” al calcolatore usando il BASIC, ed il calcolatore vi risponderà.

1.3.3 Cerchiamo di eseguire un programma

Il caricamento del BASIC nel calcolatore non provoca nessuna azione da parte di questo; è necessario dargli prima delle istruzioni in BASIC su ciò che deve fare. Benché le spiegazioni delle istruzioni del BASIC vengano date solo in seguito, procediamo e cerchiamo di eseguire un programma in BASIC ora.

Togliete la cassetta dal registratore e capovolgetela in modo che il lato “B” sia verso l’alto. Su questo lato della cassetta è registrato un programma campione. Usando le procedure seguenti, caricate questo programma nel calcolatore ed eseguitelo.

- (1) Dopo aver capovolto la cassetta ed averla ricaricata nel registratore, premete il tasto REWIND per riavvolgerla. Quindi, battete **LOAD** e premete il tasto **CR**, quando viene visualizzato il messaggio **PLAY**, premete il tasto **PLAY** del registratore. Questo dà inizio al caricamento del programma campione.
- (2) Quando il caricamento è completo, la cassetta si ferma, sullo schermo viene visualizzato il messaggio **READY**, ed il cursore ricomincia a lampeggiare.
- (3) Ora che il programma è stato caricato nella memoria del calcolatore, cercate di eseguirlo. Questo si fa battendo **RUN** e premendo il tasto **CR**.
- (4) Diamo ora un’occhiata al programma. Tenete premuto il tasto **SHIFT** e premete il tasto **BREAK**. Questo arresta l’esecuzione del programma e visualizza le parole **BREAK** e **READY**. Quindi il cursore ricomincia a lampeggiare.
- (5) Battete **LIST** e premete il tasto **CR**. Questo elenca le righe del programma sullo schermo una dopo l’altra. (L’emissione della lista può essere arrestata temporaneamente in qualsiasi momento premendo la barra spaziatrice).
- (6) Se volete riprendere l’esecuzione del programma, battete nuovamente **RUN** e premete il tasto **CR**.
- (7) Se volete eseguire un programma diverso, inserite la cassetta contenente il programma nel registratore, caricate (LOAD) il programma, quindi eseguitelo (RUN). Il programma precedente viene automaticamente cancellato dalla memoria quando viene caricato quello nuovo, per cui il calcolatore contiene soltanto il linguaggio BASIC e l’ultimo programma caricato.



2.1 Introduzione alla programmazione in BASIC

2.1.1 Modo diretto

Ora che avete fatto alcune immissioni dalla tastiera dell'MZ-700, siete arrivati al punto in cui siete pronti a cominciare ad imparare come si effettua la programmazione. Prima di cominciare tuttavia, cercate di usare l'MZ-700 come una qualsiasi calcolatrice tascabile. (Questa operazione si chiama uso dell'MZ-700 nel "modo diretto"). Battete quanto segue, esattamente come fareste con una calcolatrice tascabile.

```
4+9=CR
```

Come potete vedere, il calcolatore non fa niente quando gli viene presentato un problema in questa forma; da questo punto di vista il calcolatore ed una calcolatrice normale sono totalmente diversi, e le istruzioni devono essere immesse in una forma che possa essere capita dal calcolatore (cioè nella forma prescritta dal linguaggio BASIC). Battete ora il problema come illustrato qui di seguito.

```
PRINT 4+9CR
```

Se avete fatto tutto correttamente, verrà visualizzata la risposta "13" e lo schermo comparirà come illustrato qui di seguito.

```
READY  
PRINT 4+9  
13  
READY
```

PRINT è un'istruzione che dice al calcolatore di visualizzare qualcosa sullo schermo. Qui, al calcolatore viene data l'istruzione di visualizzare il risultato della somma di 4 + 9.

Cerchiamo ora di fare qualche calcolo un po' più complesso.

Con il BASIC per l'MZ-700, gli operatori (simboli) per le operazioni aritmetiche fondamentali sono i seguenti:

Addizione:	+	
Sottrazione:	-	
Moltiplicazione:	*	(asterisco)
Divisione:	/	(barra)
Potenza:	↑	

Quando i simboli come "*", "+", e "↑" vengono mescolati insieme in un'unica espressione aritmetica, l'ordine in cui vengono eseguiti i calcoli indicati dai simboli, viene chiamato la loro priorità.

Esattamente come nell'algebra normale, le operazioni possono essere messe tra parentesi, ed in tal modo vengono eseguite per prime le operazioni incluse nelle parentesi più interne. In una serie di parentesi, il calcolo dell'esponente viene eseguito per primo, seguito dalla moltiplicazione e dalla divisione (che hanno un'uguale priorità e che pertanto vengono eseguite nell'ordine in cui vengono incontrate nell'espressione, da sinistra verso destra), e quindi dalla somma e dalla sottrazione.

Per esempio, per calcolare il risultato dell'espressione $3 \times 6 \times (6 + 3 \times (9 - 2 \times (4 - 2) + 1))$, immettete quanto segue.

```
PRINT 3*6*(6+3*(9-2*(4-2)+1))CR
```

Cercate ora di usare il calcolatore per svolgere una serie di altri problemi aritmetici.

[ESERCIZIO]

1. $\frac{6+4}{6-4}$

2. $3 \times (5+9 \times (9-2) - \frac{6}{4-2}) + 5$

3. $(3+4) \times (5+6)$

4. $\frac{10+20}{6} \times (2+3)$

5. $\frac{10+20}{6 \times (2+3)}$

[SOLUZIONE]

PRINT (6+4)/(6-4)

5

PRINT 3*(5+9*(9-2)-6/(4-2))+5

200

PRINT (3+4)*(5+6)

77

PRINT (10+20)/6*(2+3)

25

PRINT (10+20)/(6*(2+3))

1

Dopo aver svolto gli esercizi, battete `[?][5][*][8]` e premete il tasto `[CR]`; viene visualizzata la risposta "40". Il motivo di ciò é che il BASIC interpreta il punto interrogativo nello stesso modo dell'istruzione PRINT. Ricordatevene per vostra comodità in quanto é una forma abbreviata dell'istruzione PRINT.

Immettete ora quanto segue. (Le virgolette vengono immesse tenendo premuto il tasto `[SHIFT]` e premendo il tasto `[2]`).

`PRINT"4+9="CR`

Come potete vedere, i caratteri tra virgolette vengono visualizzati sullo schermo ma non viene visualizzata la risposta. Immettete ora quanto segue.

`PRINT"ABCDEFGH"CR`

Questo provoca la visualizzazione sullo schermo di ABCDEFGH.

In altre parole, usando l'istruzione PRINT insieme alle virgolette si dice all'MZ-700 di visualizzare i caratteri sullo schermo esattamente come sono specificati tra virgolette. I caratteri tra virgolette vengono chiamati "stringa di caratteri" o "stringa".

Proseguite ora immettendo quanto segue.

`PRINT"4+9=":4+9CR`

Questo provoca la visualizzazione sullo schermo di quanto segue.

4+9=_ 1 3. (Il simbolo "_" indica uno spazio. Effettivamente non viene visualizzato niente sullo schermo TV nella posizione indicata da questo simbolo).

In altre parole, le istruzioni suddette dicono al calcolatore di visualizzare sia la stringa di caratteri "4 + 9 =" che il risultato dell'espressione aritmetica "4 + 9 =". Immettete ora quanto segue.

`PRINT"4+9="."4+9CR`

Dopo aver battuto questa immissione, sullo schermo dovrebbe essere visualizzato quanto segue.

4+9=_ _ _ _ _ 1 3

Il motivo per cui lo schermo appare diverso questa volta é che l'istruzione PRINT visualizza le informazioni (stringhe di caratteri o i risultati di espressioni aritmetiche) in modo diverso a seconda che siano separate l'una dall'altra da dei punti e virgola o da delle virgole.

Punto e virgola (;) . . . Dice al calcolatore di visualizzare i dati immediatamente adiacenti l'uno all'altro.

Virgola (,) Dice al calcolatore di visualizzare il dato nella posizione che si trova a dieci spazi (colonne) dall'inizio della riga di visualizzazione.

Se avete l'MZ-731 (o una stampante-plotter separata), attaccate ora i caratteri 「/P」 alla fine della parola PRINT.

PRINT/P"4+9=";4+9CR

Questa volta non compare niente sullo schermo, ma lo stesso risultato viene stampato sulla stampante-plotter. In altre parole, i simboli 「/P」 commutano l'emissione dallo schermo alla stampante-plotter.

Questo completa la spiegazione delle procedure per usare l'MZ-700 come fareste con una calcolatrice tascabile.

Nota: PRINT "5 + 8 = "; 5 + 8 visualizza 5 + 8 = 13, mentre PRINT "5 - 8 = "; 5 - 8 visualizza 5 - 8 = -3. Il motivo é che viene sempre riservato uno spazio per un simbolo che indica se il risultato é positivo o negativo, ma il simbolo viene visualizzato in tale spazio solo quando il risultato é negativo.

2.1.2 Programmazione

Cerchiamo di eseguire un programma semplice. Tuttavia come prima cosa, accertiamoci che l'area della memoria del calcolatore usata per memorizzare i programmi sia completamente vuota. Fatelo battendo NEW e premendo il tasto [CR]. (Questa istruzione verrà spiegata più dettagliatamente in seguito; vedere pagina 30.)

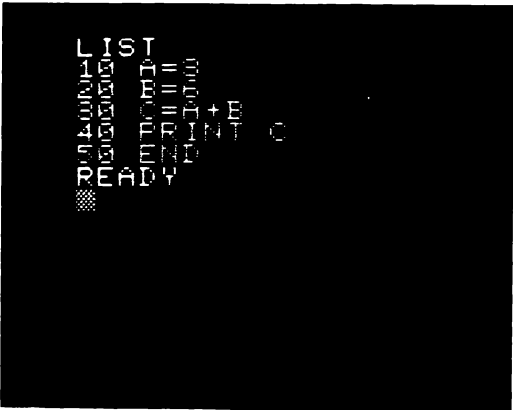
Battete il programma seguente esattamente come indicato.

- 1Ø A=3[CR] Assegna ad A il valore 3
- 2Ø B=6[CR] Assegna a B il valore 6
- 3Ø C=A+B[CR] Assegna a C il risultato di A + B
- 4Ø ? C[CR] Visualizza il valore assegnato a C.
- 5Ø END[CR] Istruzione che indica la fine del programma.

I numeri 10, 20, 30, e così via all'estremità sinistra di ogni riga vengono chiamati i numeri di riga del programma, o semplicemente i numeri di riga; questi numeri indicano l'ordine in cui le istruzioni devono essere eseguite dal calcolatore. Le istruzioni della riga con il numero più basso vengono eseguite per prime, seguite da quelle della riga con il numero immediatamente superiore, e così via. I numeri di riga devono essere dei numeri interi compresi tra 1 e 65535.

In questo programma si sarebbero potuti usare i numeri di riga 1, 2, 3 ecc. invece di 10, 20, 30. Tuttavia, é pratica comune assegnare dei numeri di riga in incrementi di 10 per consentire di inserire in seguito altre righe.

Controlliamo ora se le righe sono state immesse correttamente. Battete LIST e premete il tasto [CR]. In questo modo si ottiene una visualizzazione della lista delle righe del programma. Notate che il punto interrogativo immesso all'inizio della riga 40 é stato convertito in PRINT, la forma completa del comando per la visualizzazione dei dati sullo schermo.



Eseguiamo ora il programma.

`RUN` `CR`

Immettete RUN e premete il tasto `CR`; il risultato viene visualizzato sulla riga 9 dello schermo.

Spiegheremo ora le procedure per apportare dei cambiamenti ai programmi. Come prima cosa, cambiamo l'istruzione della riga 20 da $B = 6$ a $B = 8$. Battete LIST 20 e premete il tasto `CR`; in questo modo viene visualizzata sullo schermo solo la riga 20 del programma. Quindi, usate i tasti di controllo del cursore (i tasti alla destra della tastiera che sono contrassegnati con delle frecce) per spostare il cursore al numero "6", premete quindi il tasto `8` ed il tasto `CR` in successione per effettuare il cambiamento. **Notate che il cambiamento non è completato sino a che non viene premuto il tasto `CR`.**

Battete ora LIST e premete nuovamente il tasto `CR` per confermare che il cambiamento deve essere effettuato.

Quindi, cambiate la riga 30 del programma in $C = 30 * A + B$.

Usando i tasti di controllo del cursore, spostate il cursore in modo che venga posizionato sopra alla "A" della riga 30, quindi premete tre volte in successione il tasto `INST`. Questo sposta "A + B" di tre spazi a destra.

```
C = _ _ _ A + B
      ↑
    Posizione del cursore
```

Battete ora `30` `*` e premete il tasto `CR` per completare l'inserimento. Chiedete la lista LIST del programma per avere la conferma che il cambiamento sia stato fatto correttamente.

Cambiate ora nuovamente la riga 30 in modo che diventi " $C = 30 * A$ " invece di " $C = 30 * A + B$ ". Fatelo spostando il cursore alla posizione immediatamente a destra di B e premendo due volte il tasto `DEL`; questo cancella "+B". Premete il tasto `CR` per completare il cambiamento.

Chiedete ora la lista (LIST) del programma che deve comparire come illustrato qui di seguito.

```
1Ø A=3
2Ø B=8
3Ø C=3Ø*A
4Ø PRINT C
5Ø END
```

Per cancellare un'intera riga da un programma, basta immettere il numero di tale riga e premere il tasto `CR`; cancellate in questo modo la riga 20, quindi chiedete la lista (LIST) del programma per avere la conferma che la riga sia stata cancellata.

Potremmo inserire l'istruzione "?A" tra le righe 30 e 40, battendo `35` `_` `?A` e premendo il tasto `CR`. Fatelo, quindi chiedete la lista (LIST) del programma per avere la conferma che la riga sia stata aggiunta. Cancellate ora la riga 35 immettendo 35 e premendo il tasto `CR`.

Il procedimento di cambiamento o inserzione di righe in un programma fatto in questo modo, viene chiamato revisione (editing), e il programma che risulta da questo procedimento viene chiamato testo BASIC. Ogni riga del programma può comprendere un massimo di 255 caratteri, incluso il numero di riga, ma la lunghezza massima viene ridotta di quattro caratteri se si usa il punto interrogativo per indicare l'istruzione PRINT. Inoltre, la lunghezza massima della riga viene ridotta di altri due caratteri, se si usano le lettere minuscole.

A questo punto, il programma contenuto nella memoria del calcolatore è il seguente.

```
1Ø A=3
3Ø C=3Ø*A
4Ø PRINT C
5Ø END
```

Useremo ora questo programma per spiegare le procedure per registrare i programmi su cassetta. Preparate una cassetta vergine (una cassetta sulla quale non sia stato registrato niente) ed inseritela nel registratore, quindi battete quanto segue dalla tastiera.

SAVE "CALCULATION" J

Qui, "CALCULATION" é il nome che deve essere registrato sulla cassetta per identificare il programma. E' possibile assegnare un nome qualsiasi purché non sia più lungo di 16 caratteri.

Nota: Il simbolo J nell'esempio precedente rappresenta il tasto CR.

Quando viene premuto il tasto CR, sullo schermo viene visualizzato "⏏ RECORD. PLAY". Premendo a questo punto il tasto RECORD sul registratore, si registra il programma sulla cassetta.

Il nome assegnato al programma viene chiamato il suo nome di file (archivio). Non é assolutamente necessario specificare un nome di file, ma dal punto di vista della gestione dei file é buona idea assegnare un nome. Naturalmente, il nome di file viene registrato sul nastro insieme al programma.

Quando la registrazione é terminata, viene visualizzato READY per indicare che il calcolatore ha finito. Premete ora il tasto STOP sul registratore e riavvolgete il nastro.

Il programma é ancora presente nella memoria del calcolatore dopo che la registrazione é terminata, pertanto battete NEW J per cancellarlo (immettete LIST J per confermare che il programma sia stato cancellato). Usiamo ora l'istruzione LOAD per caricare nuovamente il programma in memoria dalla cassetta, come é descritto a pagina 12.

Quando una cassetta contiene molti programmi, quello che deve essere caricato può essere identificato specificando il nome di file del programma unitamente all'istruzione LOAD come segue.

LOAD "CALCULATION" J

Specificando il nome di file in questo modo, si dice al calcolatore di ignorare tutti i programmi su nastro diversi da quello con il nome specificato. Se non viene specificato il nome di file (cioé se si immette soltanto LOAD J), il calcolatore carica il primo programma che incontra.

Nota: Quando usate un registratore diverso dal registratore dati incorporato nell'MZ-731 e MZ-721 leggete le istruzioni di pagina 109 prima di tentare di registrare o di caricare i programmi.

Il comando LIST precedentemente illustrato può essere usato in molti modi diversi. Per esempio, durante la revisione, é possibile usare LIST 20 J per visualizzare soltanto la riga 20 di un programma. E' possibile avere la lista di un intero programma immettendo LIST J. Gli altri usi dell'istruzione sono i seguenti.

LIST -30 CR	Elenca tutte le righe del programma sino alla riga 30.
LIST 30- CR	Elenca tutte le righe dalla riga 30 alla fine del programma.
LIST 30-50 CR	Elenca tutte le righe dalla riga 30 alla riga 50.
LIST 30 CR	Elenca la riga 30.

Quando effettuate la revisione dei programmi elencando le singole righe con l'istruzione LIST, premete il tasto CLR (il tasto INST) unitamente al tasto SHIFT quando lo schermo diventa troppo affollato. Questo annulla l'intero schermo e sposta il cursore nell'angolo superiore di sinistra. (Questo non influenza il programma in memoria). Quindi, immettete nuovamente LIST < numero di riga > J per elencare la riga che deve essere rivista.

2.2 Profilo del BASIC

2.2.1 Costanti

La costante é un numero o una stringa di caratteri che é scritta in un programma e che viene usata da tale programma quando viene eseguito. I tipi di costanti includono le costanti numeriche, le costanti di stringa (caratteri), e le costanti di sistema. Queste costanti vengono spiegate qui di seguito.

Costanti numeriche

La costante numerica é un numero che ha un massimo di 8 cifre significative. L'esponente di tali costanti deve essere compreso tra 10^{-38} e 10^{38} (la gamma massima é da 1.548437E-38 a 1.7014118E + 38).

Esempi:

-123. 4

Ø. 789

3748. Ø

3. 7E+12.....3. 7×10¹²

7. 65E-9.....7. 65×10⁻⁹ } E indica l'esponente

14. 8E9.....14. 8×10⁹ }

Numeri esadecimali: I numeri possono essere specificati in formato esadecimale soltanto per l'indirizzamento diretto della memoria con le istruzioni LIMIT, POKE, PEEK e USR (vedere pagine 92 e 93), e sono rappresentati come quattro posizioni precedute da un simbolo di dollaro (\$).

Esempi:

LIMIT \$BFFF

USR (\$CØØØ, X\$) X\$ rappresenta la variabile di stringa.

Costanti di stringa

Le costanti di stringa sono delle lettere e dei simboli tra virgolette che vengono inclusi nei programmi per consentire l'emissione allo schermo video o alla stampante dei titoli o dei messaggi. I caratteri "4+9" che compaiono a pagina 15 sono una costante di caratteri e non una costante numerica. Con il BASIC, una costante di stringa può essere costituita da un massimo di 255 caratteri. (Le virgolette non possono essere incluse in una costante di stringa.)

Esempi:

" ABCDEFG "

" 1234567891Ø "

DATA ABCDEFG..... Le virgolette non sono necessarie quando le costanti di stringa vengono specificate in un'istruzione DATA; tuttavia possono essere usate se lo si desidera.

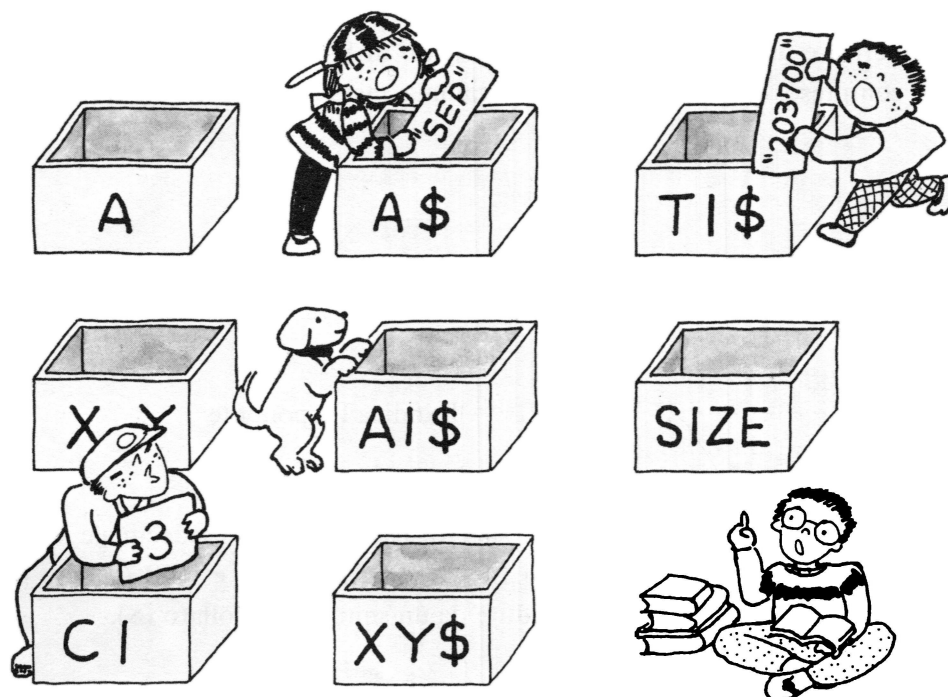
2.2.2 Variabili

Il termine “variabili” ha un significato diverso nel BASIC rispetto a quando viene usato nelle espressioni algebriche. In parole povere, le variabili del BASIC sono delle “scatole” nella memoria per la memorizzazione di numeri e di caratteri (stringhe di caratteri). I tipi di variabili usati nel BASIC includono le variabili numeriche, le variabili di stringa, e le variabili di sistema.

Variabili numeriche

Variabili di stringa

Variabili di sistema



Variabili numeriche

Nelle variabili numeriche possono essere memorizzati solo i dati numerici.

E' necessario assegnare dei nomi a queste variabili conformemente alle regole seguenti.

- Il nome della variabile può essere costituito da un numero qualsiasi di caratteri, ma soltanto i primi due caratteri vengono effettivamente usati dal programma interprete del BASIC per identificare la variabile. Inoltre, il primo carattere del nome della variabile deve essere una lettera (A – Z); per le variabili successive é possibile usare sia delle lettere che dei numeri.
- Non é possibile usare i nomi dei comandi e delle specifiche del BASIC come nomi per le variabili.

Nomi corretti delle variabili: ABC, XY, ABCD, A12345
(ABC e ABCD vengono considerate la stessa variabile).

Nomi errati delle variabili: PRINT (PRINT é una istruzione del BASIC)
C@ (i nomi delle variabili non possono includere i caratteri speciali).

Esempio:

1Ø A=5 Memorizza 5 nella variabile A.

2Ø PRINT A..... Visualizza il valore memorizzato nella variabile A.

Variabili di stringa

Le variabili di stringa sono delle variabili che vengono usate per memorizzare le stringhe di caratteri. I nomi assegnati alle variabili di stringa devono osservare le stesse regole dei nomi assegnati alle variabili numeriche; è però attaccato un simbolo di dollaro (\$) alla fine dei nomi delle variabili di stringa per differenziarle dagli altri tipi di variabili.

Le variabili di stringa possono essere usate per memorizzare un massimo di 255 caratteri. Tali variabili sono vuote sino a che non vengono assegnati loro i dati di stringa.

L'unico operatore che può essere usato nelle espressioni che includono più di una variabile di stringa è il segno "+".

Esempio:

- 1Ø A\$= " ABCD " Sostituisce la stringa di caratteri ABCD nella variabile di stringa A\$.
- 2Ø B\$= " XYZ " Sostituisce la stringa di caratteri XYZ nella variabile di stringa B\$.
- 3Ø C\$=A\$+B\$ Sostituisce la somma delle variabili di stringa A\$ e B\$ (ABCD XYZ) nella variabile di stringa C\$.
- 4Ø PRINT C\$ Visualizza il contenuto della variabile di stringa C\$.

Variabili di sistema

Le variabili di sistema contengono dei valori che vengono cambiati automaticamente dai programmi interpreti del BASIC. Le variabili di sistema sono l'ampiezza (la variabile che indica la quantità di area libera BASIC) e TI\$ (una variabile di 6 cifre che contiene il valore dell'orologio 24 ore del sistema).

Esempi:

- 1Ø TI\$= " Ø135ØØ " Questa istruzione assegna il valore corrispondente a 1 : 35 : 00 alla variabile del sistema TI\$ e regola l'orologio del sistema a tale ora.
- 2Ø PRINT TI\$. L'esecuzione di questa istruzione visualizza l'ora attuale dell'orologio del sistema (ora nelle 24 ore).

Formato video:

- 132819 Indica che l'ora è 13:28:19.

- PRINT SIZE J Questo visualizza la quantità attuale di spazio libero nella memoria del calcolatore (in altre parole, la quantità di spazio disponibile per le ulteriori righe di programma). Il valore indicato da questa variabile diminuisce ogni volta che si immette una riga di programma.

2.2.3 Matrici

Potete pensare alle matrici come a degli scaffali all'interno della memoria del calcolatore che contengono delle file di scatole, ciascuna delle quali rappresenta una variabile. Le scatole su questi scaffali sono disposte in una sequenza ordinata e sono identificate da numeri; questi numeri vengono chiamati sottoscritti, in quanto sono degli indici sottoscritti al nome che identifica l'intero gruppo di scatole.

Tali scaffali di scatole vengono allestiti eseguendo semplicemente un'istruzione che dichiara che essi esistono; questo si chiama fare una dichiarazione di matrice. La dichiarazione di matrice specifica il numero di scatole che devono essere incluse in ogni gruppo di scaffali (cioè l'ampiezza degli scaffali) ed il modo in cui devono essere disposte.

Le scatole contenute in ogni scaffale possono essere disposte in sequenze che hanno un numero qualsiasi di dimensioni. Pertanto, una matrice monodimensionale può essere considerata un singolo scaffale che contiene una fila di scatole; una matrice bidimensionale può essere considerata una pila di scaffali, ciascuno dei quali contiene una fila di scatole; e così via. Queste scatole, o variabili, vengono chiamate elementi di matrice.

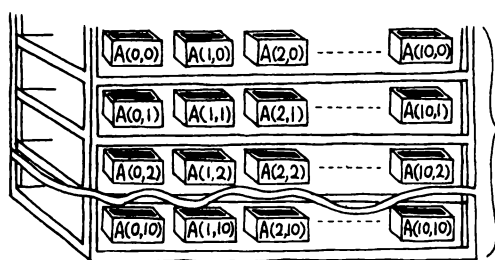
Il numero di sottoscritti usati per identificare ciascuno degli elementi di matrice, corrisponde al numero di dimensioni di tale matrice. Per esempio, ciascuno degli elementi di una matrice monodimensionale è identificato da un unico sottoscritto che indica la posizione della scatola nella fila; ciascuno degli elementi di una matrice bidimensionale è identificato da due sottoscritti, uno che identifica la fila della scatola ed uno che indica la posizione della scatola all'interno di tale fila; e così via. I numeri che vengono usati come sottoscritti partono da zero ed hanno un valore massimo che è determinato dall'ampiezza di ciascuna delle dimensioni della matrice (cioè, il numero di scatole di ciascuna fila, ecc.)

L'ampiezza massima di una matrice è limitata dalla quantità di spazio libero disponibile nella memoria del calcolatore (cioè dall'ampiezza del programma, dal numero di dati che devono essere memorizzati nella matrice, e così via). La sintassi del BASIC non pone alcuna limitazione al numero di dimensioni che si possono usare per le matrici, ma in pratica il numero di dimensioni è limitato dalla quantità di spazio di memoria libero disponibile per la memorizzazione delle variabili di matrice.

La matrice deve essere dichiarata prima di poter memorizzare i valori in uno dei suoi elementi.



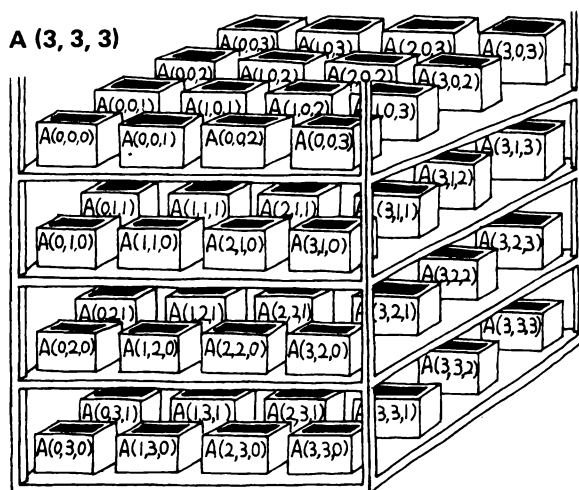
Una matrice monodimensionale costituita da 101 elementi.



A (10, 10)

Una matrice bidimensionale costituita da 11 x 11 elementi.

Una matrice tridimensionale costituita da 4 x 4 x 4 elementi.



Vediamo,
4 x 4 x 4 ...
fa 64
elementi.

Le variabili che costituiscono una matrice sono chiamate gli elementi di matrice.

Esempio 1:

10 DIM A(5)..... Dichiarare una matrice numerica monodimensionale A con 6 elementi.

20 DIM X\$(8)..... Dichiarare una matrice di stringa monodimensionale X\$ con 9 elementi.

10 DIM A(5), X\$(8)..... Svolge la stessa funzione delle righe 10 e 20 precedenti.

Esempio 2:

10 DIM B(5, 5)..... Dichiarare una matrice numerica bidimensionale B con 6 x 6 elementi.

20 DIM Y\$(5, 8)..... Dichiarare una matrice di stringa bidimensionale Y\$ con 6 x 9 elementi.

10 DIM B(5, 5), Y\$(5, 8), A(5), X\$(8)..... Dichiarare due matrici numeriche e due matrici di stringhe.

Esempio 3:

10 DIM C(3, 3, 3)..... Dichiarare una matrice tridimensionale C con 4 x 4 x 4 elementi.

Nota: Si devono usare dei nomi diversi per ogni matrice che viene dichiarata; per esempio, l'istruzione DIM A(5), A(6) non è una dichiarazione di matrice legale.

Eseguite il programma riportato qui di seguito e controllate i risultati ottenuti.

```
10 DIM A(2), B$(2)
20 A(0)=26
30 A(1)=9
40 A(2)=-100
50 B$(0)= " ABC "
60 B$(1)= " XYZ "
70 B$(2)= " MZ-700 "
80 PRINT A(1)
90 PRINT B$(2)
100 PRINT A(2)
110 PRINT B$(0)+B$(1)
120 PRINT A(0)
```

Nota: Le singole variabili all'interno di una matrice, quali A(5) ed X\$(8), vengono chiamate gli elementi di una matrice. Le costanti numeriche, le variabili numeriche e le matrici numeriche vengono chiamate collettivamente delle espressioni numeriche, mentre le costanti di stringa, le variabili di stringa e le schiere di stringa vengono chiamate collettivamente espressioni di stringa.

2.2.4 Operazioni in BASIC

Nel BASIC, le operazioni aritmetiche assumono una forma leggermente diversa da quanto avviene nell'aritmetica normale. I diversi operatori aritmetici usati nel BASIC vengono illustrati nella tabella seguente. La priorità di questi operatori quando vengono usati insieme nell'ambito di un'unica espressione (la sequenza in cui vengono eseguite le diverse operazioni aritmetiche) è quella indicata dai numeri della colonna di sinistra della tabella; **tuttavia, gli operatori all'interno delle parentesi hanno sempre la priorità massima.**

Operazioni aritmetiche

	Operatore	Operazione	Formato
1	\uparrow	Potenza	$X \uparrow Y$ (indica X^Y ; cioè, X alla Y-esima potenza).
2	$-$	Negazione	$-X$
3	$*$, $/$	Moltiplicazione, divisione	$X * Y$ (X per Y), X/Y ($\frac{X}{Y}$; cioè, X diviso per Y)
4	$+$, $-$	Plus, meno	$X + Y$ (X più Y), $X - Y$ (X meno Y)



Esempio 1:

10 A=3*8/4..... Quando in una espressione aritmetica viene usata una serie di operatori con la stessa priorità, i calcoli vengono eseguiti da sinistra verso destra; pertanto, il risultato dell'espressione a sinistra è 6.

Esempio 2:

10 A=60-6*8+2..... Il risultato è 14.
20 B=(60-6)*8+2.. Il risultato è 434.

Esempio 3:

10 A=2↑3..... Assegna ad A il valore 2 alla terza potenza; il risultato è 8.

Operazioni di stringa

Le operazioni di stringa vengono usate per creare nuove stringhe di caratteri mediante concatenazione (collegamento) di due o più stringhe più brevi. L'unico operatore che si può usare nelle operazioni di stringa è il segno "+".

Esempio:

PRINT "ABC" + "DEF" J → Visualizza la stringa di caratteri "ABCDEF".

2.2.5 Impostazioni iniziali

Le impostazioni iniziali fatte quando viene avviato il BASIC 1Z-013B, sono come viene descritto qui di seguito.

■ Tastiera

- 1) Modo operativo: normale (alfanumerico)
- 2) Tasti funzionali definibili

F1 :	"RUN" +CHR\$(13)	SHIFT + F1 :	"CHR\$("
F2 :	"LIST"	SHIFT + F2 :	"DEF KEY ("
F3 :	"AUTO"	SHIFT + F3 :	"CONT"
F4 :	"RENUM"	SHIFT + F4 :	"SAVE"
F5 :	"COLOR"	SHIFT + F5 :	"LOAD"

Nota: Nella definizione del tasto funzionale F1 è incluso un codice di ritorno a capo.

■ Orologio incorporato

Il valore iniziale dato alla variabile del sistema TI\$ è "000000".

■ Funzione musicale

- 1) Tempo dell'esecuzione musicale: 4 (moderato, circa velocità media)
- 2) Durata della nota: 5 (nota di un quarto J)

■ Tasti di comando e caratteri di comando

I tasti di comando sono i tasti che eseguono delle funzioni speciali quando vengono premuti unitamente al tasto **CTRL**. Le funzioni di questi tasti ed i loro corrispondenti codici ASCII vengono illustrati nella tabella seguente. I codici che possono essere emessi con l'istruzione PRINT per ottenere lo stesso risultato ottenuto premendo il corrispondente tasto di comando sono contrassegnati da asterischi.

[Codici di comando]

CTRL +	Codice ASCII (decimale)	Funzione
E	5	Sceglie il modo di immissione in lettere minuscole per i caratteri alfanumerici.
F	6	Sceglie il modo di immissione in lettere maiuscole per i caratteri alfanumerici.
M	13	Ritorno a capo (CR).
P	16	Uguale al tasto DEL .
Q	17	Sposta il cursore in basso di una riga (↓).
R	18	Sposta il cursore in alto di una riga (↑).
S	19	Sposta il cursore a destra di una colonna (carattere) (→).
T	20	Sposta il cursore a sinistra di una colonna (←).
U	21	Sposta il cursore alla posizione di riposo (HOME).
V	22	Riporta lo schermo al colore di sfondo (CLR).
W	23	Mette il calcolatore nel modo operativo di immissione caratteri grafici (GRAPH).
X	24	Inserisce uno spazio (INST).
Y	25	Mette il calcolatore nel modo operativo di immissione di dati alfanumerici.

■ Altro

Il limite inferiore dell'area di testo BASIC è stabilito all'indirizzo \$FEFF (questo corrisponde all'esecuzione di LIMIT MAX).

Per le impostazioni iniziali della stampante, si veda la voce stampante.

2.3 Comandi e istruzioni BASIC frequentemente usati

2.3.1 Istruzioni di immissione/emissione dei file programmi

2.3.1.1 LOAD (formato abbreviato: LO.)

Formato

LOAD or LOAD " nome file "

Funzione

Questo comando carica il file dei testi in BASIC specificati o un file di linguaggio macchina che deve essere collegato ad un programma BASIC dalla cassetta.

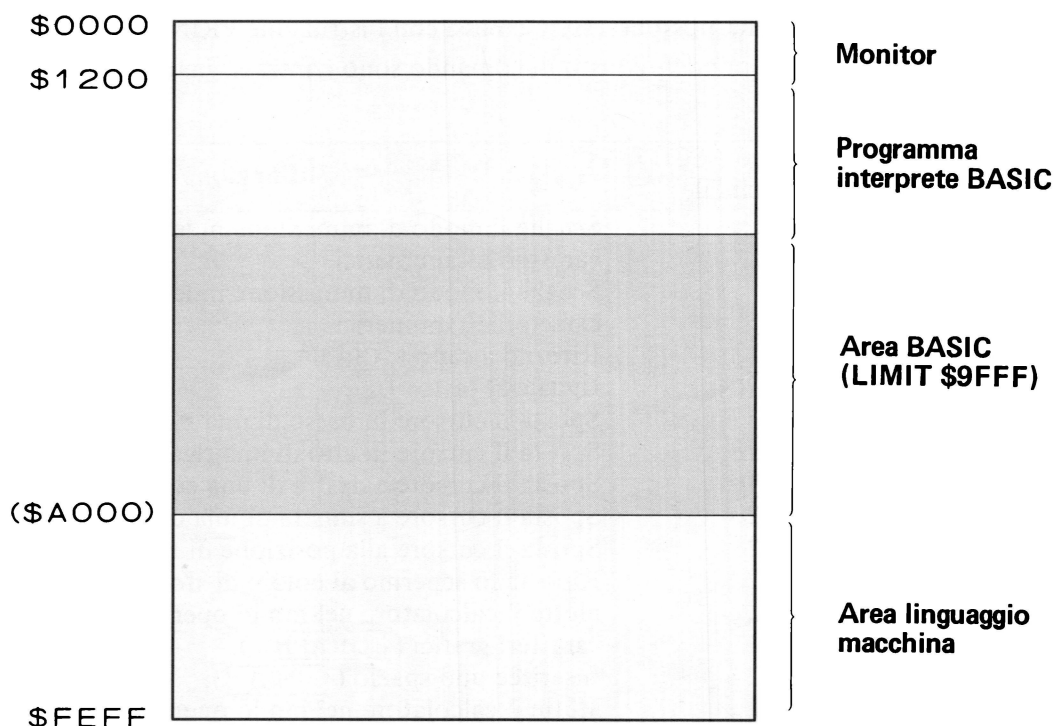
(Vedere pagine 12 e 18).

Con questo comando é possibile caricare soltanto i file di testi in BASIC e i programmi di linguaggio macchina. Quando il file da caricare é un file di testi BASIC, il programma attuale viene cancellato dall'area di testo BASIC quando viene caricato il nuovo programma.

Nota

Quando caricate una routine di linguaggio macchina che deve essere collegata ad un programma BASIC, é necessario eseguire l'istruzione LIMIT per riservare un'area di programma per il linguaggio macchina in memoria. Inoltre, il file dei programmi linguaggio macchina applicabile viene eseguito non appena il caricamento é stato completato, se l'indirizzo di caricamento é all'interno di tale area. (In questo caso, il testo BASIC non viene cancellato).

Il comando LOAD può essere utilizzato all'interno di un programma per caricare un file di programmi linguaggio macchina.



Nota: Il limite inferiore dell'area BASIC si sposta a seconda dell'ampiezza del testo di programma caricato.

2. 3. 1. 2 SAVE (formato abbreviato: SA.)

Formato

SAVE o SAVE " nome file "

Funzione

Questo comando assegna un nome di file al programma BASIC nella memoria del calcolatore e lo memorizza sul nastro.



Nota

Questo comando memorizza soltanto il testo del programma BASIC (cioè, il testo del programma visualizzato eseguendo il comando LIST); non memorizza alcun programma di linguaggio macchina nell'area linguaggi macchina.

Il nome di file specificato viene registrato sul nastro insieme al file dei testi BASIC; specificate il nome desiderato usando un massimo di 16 caratteri. Se non viene specificato alcun nome per il file, il programma viene registrato senza un nome di file; notate tuttavia che questo può rendere difficoltosa la gestione dei file se in unico nastro vengono registrati più di un programma.

2. 3. 1. 3 VERIFY (formato abbreviato: V.)

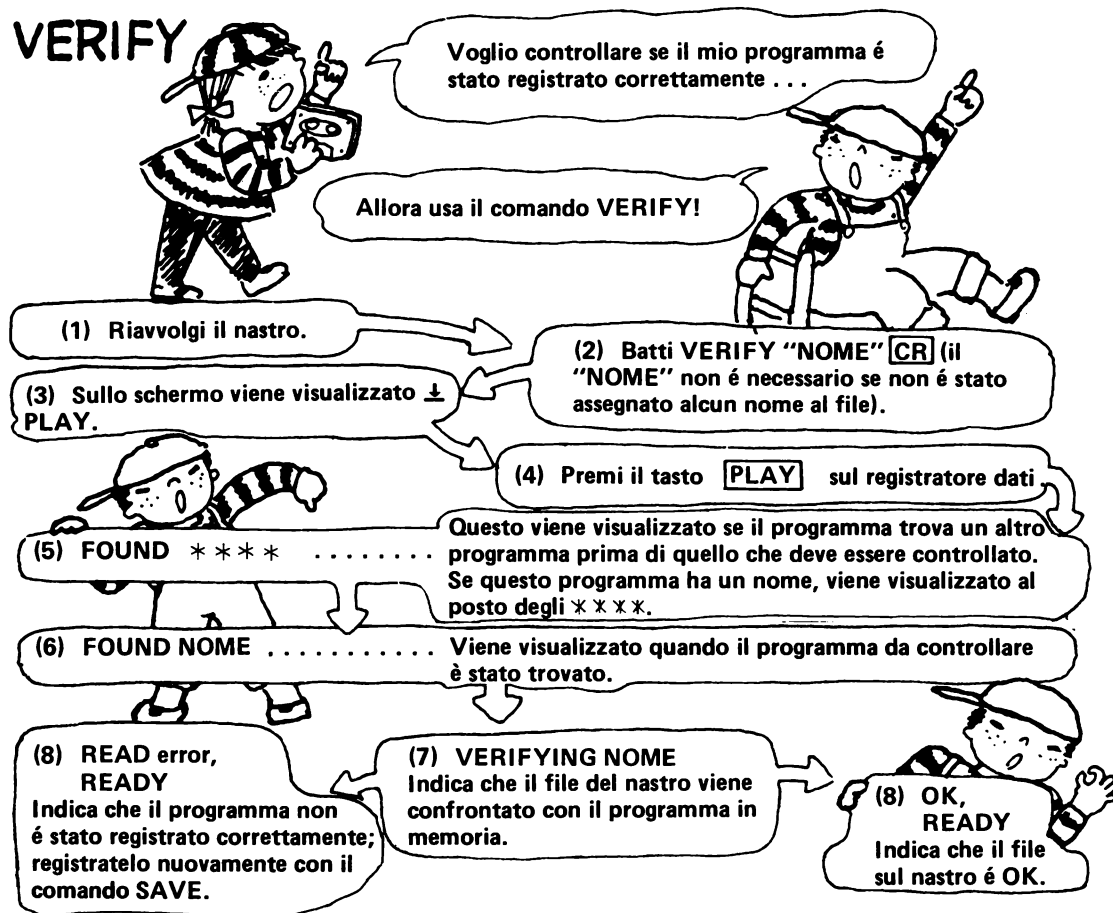
Esempio

VERIFY o VERIFY "nome file"

Funzione

Questo comando viene usato per avere la conferma che i programmi sono stati correttamente registrati su nastro mediante il comando SAVE. Questo viene fatto leggendo il nastro e confrontando il programma letto con il programma contenuto in memoria. Se i due programmi sono uguali, viene visualizzato "OK"; se sono diversi, viene visualizzato "READ error". In quest'ultimo caso, memorizzate nuovamente il programma.

VERIFY



2.3.2 Comandi di revisione testo

2.3.2.1 AUTO (formato abbreviato: A.)

Formato

AUTO o AUTO LS, n

LS ... Numero della riga iniziale

n..... Incremento dei numeri di riga

Funzione

Questo comando genera automaticamente i numeri delle righe dei programmi durante l'immissione delle istruzioni di programma BASIC.

Esempi

(Esempio 1)

AUTO ↵

10..... ↵

20..... ↵

30..... ↵

(Esempio 2)

AUTO 300, 5 ↵

300..... ↵

305..... ↵

310..... ↵

Genera automaticamente i numeri delle righe del programma con un incremento di 5, partendo dalla riga 300.

(Esempio 3)

AUTO 100 ↵

100..... ↵

110..... ↵

120..... ↵

} Genera i numeri delle righe del programma con un incremento di 10, partendo dalla riga 100.

(Esempio 4)

AUTO, 20 ↵

10..... ↵

30..... ↵

50..... ↵

} Genera i numeri delle righe del programma con un incremento di 20, partendo dalla riga 10.

Nota: Si pone termine al comando AUTO premendo

SHIFT

 e

BREAK

.

2.3.2.2 DELETE (formato abbreviato: D.)

Formato

DELETE LS—LE Cancella le righe del programma da LS a LE.

DELETE —LE Cancella tutte le righe del programma dall'inizio del programma fino alla riga LE.

DELETE LS— Cancella tutte le righe del programma dalla riga LS fino alla fine del programma.

DELETE LS Cancella la riga LS.

Esempi

(Esempio 1)

DELETE 150—350 ↵ ... Cancella tutte le righe del programma dalla 150 alla 350.

(Esempio 2)

DELETE —100 ↵ Cancella tutte le righe del programma sino alla riga 100.

(Esempio 3)

DELETE 400— ↵ Cancella tutte le righe del programma dalla 400 alla fine del programma.

2. 3. 2. 3 LIST (formato abbreviato: L.)

Formato	<div><div>LIST</div><div>LIST Ls-Le</div><div>LIST Ls-</div><div>LIST -Le</div></div>	} Ls indica il numero della riga di inizio mentre Le indica il numero della riga finale.

Funzione Questo comando elenca tutte o parte delle righe di programma contenute in un'area di testo BASIC sullo schermo video.

LIST ↵ Elenca l'intero programma.
LIST -30 ↵ Elenca tutte le righe del programma sino alla riga 30.
LIST 30- ↵ Elenca tutte le righe del programma dalla riga 30 alla fine.
LIST 30-50 ↵ Elenca tutte le righe del programma dalla riga 30 alla riga 50.
LIST 30 ↵ Elenca la riga 30 del programma.

L'emissione della lista del programma allo schermo video può essere interrotta temporaneamente premendo la barra spaziatrice; la produzione della lista viene quindi ripresa quando si lascia andare la barra spaziatrice. Per porre termine all'emissione della lista, premete il tasto **BREAK** insieme al tasto **SHIFT**.

2. 3. 2. 4 LIST/P (fromato abbreviato: L./P)

Formato	LIST/P <Ls-Le>
	Ls ... Numero della riga iniziale
	Le ... Numero della riga finale

Funzione Questo comando stampa la lista di tutto o parte del programma nell'area di testo BASIC sulla stampante. La gamma delle righe di programma da listare viene specificata nello stesso modo come con il comando LIST precedentemente descritto.

Nota: Le parentesi ad angolo <...> indicano che il dato racchiuso tra parentesi é facoltativo.

2. 3. 2. 5 MERGE (formato abbreviato: ME.)

Esempio MERGE o MERGE " nome file "

Funzione Il comando MERGE viene usato per leggere un programma dalla cassetta. Quando un programma viene letto usando questo comando, esso viene attaccato al programma in memoria. Se si omette il "nome file", il calcolatore legge il primo file incontrato sulla cassetta.

Se dei numeri di riga del programma letto sono uguali a quelli del programma in memoria, le righe corrispondenti del programma in memoria vengono sostituite con le righe del programma letto.

2. 3. 2. 6 NEW

Formato NEW

Funzione Il comando NEW cancella l'area di testo BASIC e annulla tutte le variabili. Eseguite questo comando quando desiderate annullare il programma in memoria prima di immettere un altro programma. Questo comando non cancella l'area di linguaggio macchina riservata dall'istruzione LIMIT.

Poiché l'area di testo BASIC viene automaticamente cancellata dal comando LOAD, non é necessario eseguire questo comando prima di caricare un programma BASIC dalla cassetta.

2. 3. 2. 7 RENUM (formato abbreviato: REN.)

Formato

RENUM		}	Ln	Nuovo numero di riga	
RENUM	Ln		Lo	Vecchio numero di riga
RENUM	Ln, Lo, n		n	Incremento	

Funzione

Questo comando rinumerava le righe di un programma BASIC. Quando viene eseguito questo comando, vengono anche riassegnati i numeri di riga indicati nelle istruzioni di salto quali GOTO, GOSUB, ON ~ GOTO, e ON ~ GOSUB.

RENUM Rinumerava le righe del programma attuale nella memoria per cui partono da 10 e sono incrementate in unità di 10.

RENUM 100 Rinumerava le righe del programma attuale nella memoria per cui partono da 100 e vengono incrementate in unità di 10.

RENUM 100, 50, 20 Rinumerava le righe del programma attuale in memoria cominciando dal numero di riga 50; il numero di riga 50 viene rinumerato a 100, e i numeri di riga successivi vengono incrementati in unità di 20.

Esempio

L'esempio seguente illustra il risultato dell'esecuzione di RENUM 100, 50, 20 per un programma campione.

(Prima della rinumerazione)		(Dopo la rinumerazione)
50 A=1	}	100 A=1
60 A=A+1		120 A=A+1
70 PRINT A		140 PRINT A
100 GOTO 60		160 GOTO 120

Nota

Quando specificate i nuovi ed i vecchi numeri di riga, il nuovo numero di riga specificato deve essere superiore al vecchio numero di riga. Notate che si avrà un errore se l'esecuzione di questo comando porta alla generazione di un numero di riga superiore a 65535.

2.3.3 Comandi di controllo

2.3.3.1 RUN (formato abbreviato: R.)

Formato

RUN o RUN Ls

Ls Numero di righe iniziale

Funzione

Questo comando esegue il programma attuale nell'area di testo BASIC.

Se il programma deve essere eseguito partendo dalla prima riga di programma, basta immettere RUN e premere il tasto **[CR]**. Se l'esecuzione deve cominciare da una riga diversa da quella con il numero minore, battete RUN Ls (dove Ls è il numero di riga da cui deve cominciare l'esecuzione) e premete il tasto **[CR]**.

Quando viene eseguito questo comando, il programma interprete BASIC annulla tutte le variabili e le matrici prima di passare il controllo al programma BASIC.

2.3.3.2 CONT (formato abbreviato: C.)

Formato

CONT

Funzione

Il comando CONT viene usato per riprendere l'esecuzione di un programma che è stata interrotta premendo **[SHIFT] + [BREAK]** o da un'istruzione STOP del programma. Questo comando può essere usato anche per continuare l'esecuzione di un programma che è stato interrotto da un'istruzione END; tuttavia, in questo caso occorre fare attenzione a che le righe che seguono l'istruzione END non siano le righe di una sottoroutine.

Nella tabella seguente vengono dati degli esempi di situazioni in cui il comando CONT può e non può essere usato.

Continuazione programma possibile	Continuazione programma impossibile
<ul style="list-style-type: none">• Esecuzione programma arrestata premendo [SHIFT] + [BREAK].• Esecuzione programma arrestata da un comando STOP.• Esecuzione programma arrestata premendo [SHIFT] + [BREAK] mentre il programma era in attesa di un'immissione per un'istruzione INPUT.	<ul style="list-style-type: none">• Prima che sia stato eseguito un programma RUN.• Viene visualizzato "READY" a causa di un errore verificatosi durante l'esecuzione del programma.• Funzionamento della cassetta interrotto premendo [SHIFT] + [BREAK].• Esecuzione programma arrestata durante l'esecuzione di un'istruzione MUSIC.• Esecuzione programma arrestata e "READY" visualizzato dopo l'esecuzione di un'istruzione END.

2. 3. 3. 3 **BYE** (formato abbreviato: **B.**)

Formato

BYE

Funzione

Questo comando riporta il controllo del calcolatore dal programma interprete BASIC IZ-013B al programma monitor nella RAM. (I comandi monitor vengono spiegati partendo da pagina 99).

2. 3. 3. 4 **KEY LIST** (formato abbreviato: **K. L.**)

Formato

KEY LIST

Funzione

Questo comando visualizza una lista delle stringhe di caratteri assegnate ai tasti funzionali definibili.

```
KEY LIST
DEF KEY (1) = " RUN " + CHR$ (13)
DEF KEY (2) = " L I S T "
DEF KEY (3) = " AUTO "
DEF KEY (4) = " RENUM "
DEF KEY (5) = " COLOR "
DEF KEY (6) = " CHR$ ( "
DEF KEY (7) = " DEF KEY ( "
DEF KEY (8) = " CONT "
DEF KEY (9) = " SAVE "
DEF KEY (10) = " LOAD "
READY
```



2.3.4 Istruzione di assegnazione

LET

Formato

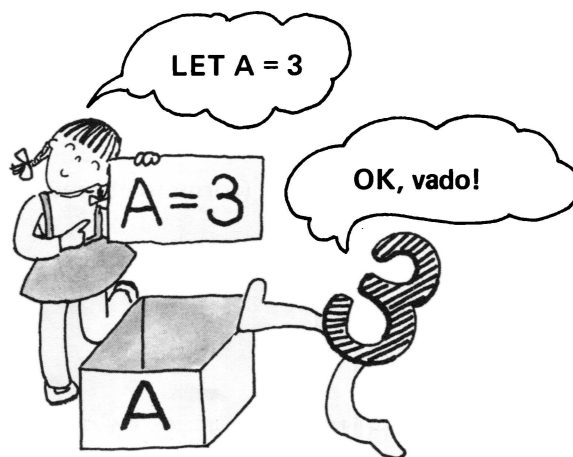
LET $v = e$ o $v = e$

v ... Variabile o elemento di matrice numerici, oppure variabile o elemento di matrice di stringa.

e ... Espressione numerica (costituita da uno o più costanti, variabili, o elementi di matrice) o espressione di stringa (costituita da una o più costanti, variabili o elementi di matrice).

Funzione

Questa istruzione assegna il valore (numerico o di stringa) specificato da e alla variabile o elemento di matrice specificato da v . Come illustrato nell'esempio seguente, LET può essere tralasciato.



Esempio

10 A=10

20 B=20

30 A=A+B

40 PRINT A

50 END

10 LET A=10

20 LET B=20

30 LET A=A+B

40 PRINT A

50 END

RUN ↓ I due programmi precedenti producono
30 esattamente lo stesso risultato.

I seguenti sono degli esempi di uso non corretto dell'istruzione LET.

20 A\$=A+B... Invalida in quanto vengono specificati tipi diversi di variabili (di stringa e numerica) ai lati del segno “=”.

20 LOG (LK) =LK+1... Invalida in quanto il lato sinistro dell'istruzione non è una variabile o un elemento di matrice numerico.

2.3.5 Istruzioni di immissione/emissione

Le istruzioni di immissione/emissione sono i mezzi con cui i dati vengono presentati al calcolatore per l'elaborazione, e con cui i risultati dell'elaborazione vengono emessi allo schermo o alla stampante.

2.3.5.1 PRINT

Formato

```

{ PRINT } { variabile } < { ; } { variabile } >
{ ? } { constant } { , } { constant } .....
{ expression } { expression }

```

Funzione

Questa istruzione emette i valori delle variabili, delle costanti, delle stringhe di caratteri o delle espressioni allo schermo. I valori vengono visualizzati partendo dalla posizione attuale del cursore sullo schermo. (Per far scendere il cursore di una riga sullo schermo, eseguite l'istruzione PRINT senza specificare alcuna variabile, costante o espressione).

Per semplificare l'immissione da tastiera di quest'istruzione, è possibile battere un punto interrogativo (?) invece della parola PRINT.

I dati numerici vengono visualizzati da questa istruzione in uno di due formati: formato numeri reali o formato esponenziale.

Formato numeri reali

I valori numerici nella gamma compresa tra 1×10^{-8} e 1×10^8 vengono visualizzati nel formato dei numeri reali.

Esempio

```

-1.9999
63598757
0.00000001 ..... 1 x 10-8
99999999

```

Formato esponenziale

I numeri che non possono essere visualizzati nel formato dei numeri reali vengono visualizzati nel formato esponenziale.

Esempio

```







-.31415E+9 ..... -0.31415 x 109
.513606E-20 ..... 0.513606 x 10-20

```

A lato dell'esponente (il numero che segue la "E") di un numero visualizzato in formato esponenziale, viene sempre visualizzato un segno più (+) o meno (-).

Qui di seguito vengono illustrati alcuni metodi speciali per usare l'istruzione PRINT.

```

PRINT "  " Cancella l'intero schermo e porta il cursore alla posizione di sede
                                     (l'angolo superiore di sinistra dello schermo).
PRINT "  " Sposta il cursore alla posizione di sede senza annullare lo schermo.
PRINT "  " Sposta il cursore di una colonna a destra.
PRINT "  " Sposta il cursore di una colonna a sinistra.
PRINT "  " Sposta il cursore di una riga verso l'alto.
PRINT "  " Sposta il cursore di una riga in basso.

```

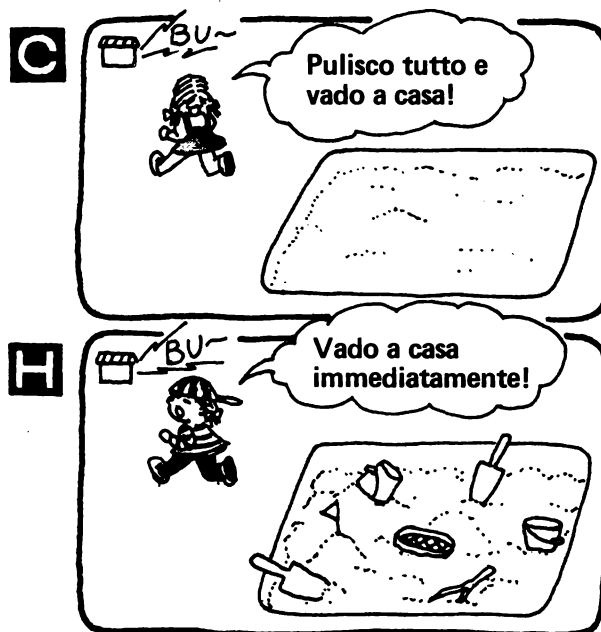

PRINT "C11111A" ... Annulla lo schermo, quindi visualizza il carattere "A" all'inizio della sesta riga dall'alto.

Le procedure per immettere questi comandi vengono descritte nella pagina seguente.

Nota: Le barre verticali nella descrizione del formato indicano che è possibile scegliere uno qualsiasi dei dati inclusi.

Per immettere i caratteri speciali per il controllo del cursore, premete il tasto **GRAPH**; questo mette il BASIC nel modo operativo di immissione dei caratteri grafici e cambia la forma del cursore in "■". Quindi immettete i caratteri come segue.

- Premete il tasto **CLR** .
- Premete il tasto **HOME** .
- Premete il tasto **→** .
- Premete il tasto **←** .
- Premete il tasto **↑** .
- Premete il tasto **↓** .



Dopo aver immesso un carattere speciale, premete il tasto **ALPHA** per tornare dal modo di immissione dei caratteri grafici al modo di immissione dei caratteri alfanumerici.

2. 3. 5. 2 PRINT USING (formato abbreviato: ?USI.)

Formato

PRINT USING " stringa formato "; variabile < { ; } variabile ... >

Funzione

Questa istruzione visualizza i dati sullo schermo in un formato specifico. La specifica di formato è costituita da un carattere o da una stringa di caratteri tra virgolette, e viene specificato immediatamente dopo la parola USING come segue.

(1) Stringhe di specifica di formato per i valori numerici

(a) #

Il segno di numero viene usato per specificare il numero massimo di posizioni da visualizzare. Se il numero delle posizioni del numero visualizzato è inferiore al numero di segni # specificati nella "stringa formato", i numeri vengono giustificati a destra nel campo definito da tale stringa.

(Esempio:)

10 A = 123

20 PRINT USING "####"; A

RUN ↵

123

(b) .

E' possibile includere un punto in una stringa di formato costituita dai segni # per specificare la posizione in cui deve essere visualizzato il punto decimale. Il numero di segni # alla destra del punto decimale specifica il numero di posizioni decimali da visualizzare.

(Esempio:)

10 A = 12.345 : B = 6.789

20 PRINT USING "###.##" ; A

30 PRINT USING "###.##" ; B

RUN ↵

└ 12.34

└└ 6.79

(c) ,

E' possibile includere anche le virgole nella "stringa formato" per indicare le posizioni in cui devono essere visualizzate le virgole. I numeri vengono giustificati a destra nello stesso modo di quando si usano i segni da soli.

(Esempio:)

10 A = 6345123 : B = 987324

20 PRINT USING "#,###,###" ; A

30 PRINT USING "#,###,###" ; B

RUN ↵

6,345, 123

└└ 987,324

(d) + e -

E' possibile includere un segno più (+) o meno (-) alla fine della "stringa formato" per specificare che deve essere visualizzato in tale posizione il segno del numero invece di uno spazio. Per esempio, PRINT USING "####+" provoca la visualizzazione del segno immediatamente dopo il numero. (PRINT USING "####-" provoca la visualizzazione di un segno meno dopo il numero se il numero è negativo, se il numero è positivo, in tale posizione viene visualizzato soltanto uno spazio).

Inoltre, è possibile specificare un segno più all'inizio di una stringa formato per indicare che il segno del numero deve essere visualizzato in tale posizione indipendentemente dal fatto che sia positivo o negativo.

(Esempio:)

PRINT USING "####+" ; -13

└└ 13-

PRINT USING "+####" ; 25

└└ +25

(Nota:)

Sebbene venga visualizzato un segno meno se è stato specificato all'inizio della stringa formato, questo non ha alcuna relazione con il segno del numero.

(e) **

Se si specifica una coppia di asterischi all'inizio della stringa formato, si indica che devono essere visualizzati degli asterischi nelle posizioni degli zeri iniziali.

(Esempio:)

```
10 A = 1234
```

```
20 PRINT USING " **#### " ; A
```

```
RUN ↵
```

```
**1234
```

(f) ££

Se si specifica una coppia di segni di sterlina all'inizio della stringa formato, si indica che deve essere visualizzato un segno di sterlina nella posizione immediatamente a sinistra del numero.

(Esempio:)

```
10 A = 123
```

```
20 PRINT USING " ££#### " ; A
```

```
RUN ↵
```

```
££ 123
```

(g) \$\$

Se si specifica una coppia di segni di dollaro all'inizio della stringa formato, si indica che deve essere visualizzato un segno di dollaro nella posizione immediatamente a sinistra del numero.

(h) ↑↑↑↑

E' possibile includere alla fine di una stringa formato quattro operatori esponenziali per controllare la visualizzazione dei numeri nel formato esponenziale.

(Esempio:)

```
10 A = 51123
```

```
20 PRINT USING " ##.### ↑↑↑↑ " ; A
```

```
RUN ↵
```

```
 5.112E+04
```

In questo caso, il primo segno di numero è riservato per la visualizzazione del segno del numero.

(i) Lista estesa di operandi

E' possibile specificare una lista di variabili a seguito di un'unica istruzione PRINT USING separandole l'una dall'altra con delle virgole o dei punti e virgola. Così facendo, il formato specificato in "stringa formato" viene usato per visualizzare tutti i valori risultanti.

(Esempio:)

```
10 A = 5.3 : B = 6.9 : C = 7.123
```

```
20 PRINT USING " ##.### " ; A, B, C
```

```
RUN ↵
```

```
 5.300 6.900 7.123
```

(2) Specifica di formato per i valori di stringa

(a) !

Quando i valori che vengono visualizzati sono delle stringhe di caratteri, specificando un punto esclamativo nella "stringa formato" si provoca la visualizzazione soltanto del primo carattere della stringa specificata.

(Esempio:)

```
10 A$ = "CDE"
20 PRINT USING "!" ; A$
RUN
C
```

(b) & _ _ _ _ &

Se si specifica "& _ _ _ _ &" nella stringa formato, si provoca la visualizzazione dei primi 2 + n caratteri delle espressioni di stringa specificate (dove n è il numero di spazi tra le due commerciali). Se vengono specificati meno di 2 + n caratteri in un'espressione di stringa, i caratteri vengono giustificati a sinistra nel campo definito da "& _ _ _ _ &".

(Esempio:)

```
10 A$ = "ABCDEFGH"
20 PRINT USING "& _ _ _ _ &" ; A$
RUN
ABCDEF
10 A$ = "XY"
20 PRINT USING "& _ _ _ _ &" ; A$
RUN
XY
```

(3) Funzione di emissione delle costanti di stringa

Quando nella stringa formato di un'istruzione PRINT USING vengono inclusi dei caratteri diversi da quelli descritti in precedenza, questi caratteri vengono visualizzati insieme al valore specificato dopo il punto e virgola.

(Esempio:)

```
10 A = 123
20 PRINT USING "DATA####" ; A
RUN
DATA 123
```

(4) Separazione di USING

Generalmente, PRINT e USING vengono specificati l'uno adiacente all'altro; tuttavia è possibile usarli separatamente nella stessa istruzione.

(Esempio:)

```
10 A = -12 : B = 14 : C = 12
20 PRINT A ; B ; USING "####" ; C
      Funzione normale PRINT  Funzione USING
RUN
-12 14 12
```

2. 3. 5. 3 INPUT (formato abbreviato: I.)

Esempio

$\text{INPUT} \left\{ \begin{array}{l} \text{variabile numerica} \\ \text{variabile di stringa} \\ \text{elemento di matrice} \end{array} \right\} \dots \text{INPUT} \text{ "stringa di caratteri " ; } \left\{ \begin{array}{l} \text{variabile numerica} \\ \text{variabile di stringa} \\ \text{elemento di matrice} \end{array} \right\} \dots$

```
INPUT A
INPUT B$
INPUT X(5)
```

```
INPUT " DATA  A=" ; A
INPUT " YES  OR  NO " ; B$
INPUT " KEY   IN " ; X (5)
```

Funzione

INPUT è una delle istruzioni che vengono usate per immettere dei valori da assegnare alle variabili durante l'esecuzione del programma. L'esecuzione del programma fa una pausa quando si incontra un'istruzione INPUT per consentire di battere i valori dalla tastiera. Dopo che l'immissione è stata completata, i valori vengono sostituiti nelle variabili specificate premendo il tasto **[CR]**, quindi l'esecuzione del programma riprende.

(Esempio:)

```
10 INPUT A, B
20 C=A+B
30 PRINT C
40 END
```

Quando si esegue il programma suddetto, viene visualizzato un punto interrogativo ed il cursore lampeggia per indicare che il calcolatore aspetta l'immissione di dati; immettete un qualsiasi numero arbitrario, quindi premete il tasto **[CR]**. Questo assegna il valore immesso alla variabile A. Dopo aver fatto ciò, il punto interrogativo viene nuovamente visualizzato. Il motivo è che nell'istruzione INPUT alla riga 10 sono state specificate due variabili (A e B), ma è stato immesso un unico valore (quello che viene sostituito nella variabile A). Immettete un altro numero arbitrario e premete nuovamente il tasto **[CR]**; questo sostituisce il secondo valore immesso nella variabile B e provoca la continuazione dell'esecuzione alla riga successive sommano i valori di A e B, sostituiscono il risultato in C, quindi visualizzano il contenuto di C.

Poiché le variabili usate in questo esempio sono delle variabili numeriche, il calcolatore visualizza il messaggio **ILLEGAL DATA ERROR** se viene fatto il tentativo di immettere dei caratteri non numerici. Il punto interrogativo viene quindi nuovamente visualizzato per chiedere all'utente di immettere nuovamente un valore legale (un valore il cui tipo sia uguale a quello della variabile o dell'elemento di matrice nel quale deve essere sostituito). Accertatevi di immettere dei dati il cui tipo corrisponde a quello della variabile o delle variabili specificate nell'istruzione INPUT.

Durante l'esecuzione del programma, può essere difficile ricordare quali dati devono essere immessi quando viene visualizzato il punto interrogativo: pertanto, vengono generalmente incluse nelle istruzioni INPUT delle stringhe di messaggi che devono essere visualizzate sullo schermo come promemoria. Questo viene fatto come mostrato nell'esempio di programma seguente.

```

10 INPUT "A=" ; A
20 INPUT "B=" ; B
30 PRINT "A+B=" ; A+B
40 PRINT "A-B=" ; A-B
50 PRINT "A*B=" ; A*B
60 PRINT "A/B=" ; A/B
70 END

```

Eseguite il programma precedentemente illustrato. L'inclusione delle stringhe di caratteri nelle istruzioni PRINT ed INPUT dà una chiara indicazione del funzionamento del programma.

I programmi pratici del calcolatore sono costituiti da combinazioni di sequenze simili a quelle mostrate. Combinando i comandi, le istruzioni e le sequenze in modi diversi, scoprirete presto che esistono molti metodi diversi per ottenere un risultato desiderato.

2. 3. 5. 4 GET

Esempio

GET v

v Variabile o elemento di matrice numerica, o variabile o elemento di matrice.

Funzione

Quando si incontra questa istruzione durante l'esecuzione del programma, il programma interprete BASIC controlla se è premuto un tasto della tastiera e, in tal caso, assegna il valore corrispondente alla variabile specificata in v. Mentre l'istruzione, INPUT chiede l'immissione di dati ed aspetta sino a che tali dati sono stati immessi prima di riprendere l'esecuzione, l'istruzione GET continua l'esecuzione indipendentemente dal fatto che venga premuto un tasto.

Sebbene i dati vengano sostituiti nella variabile v dall'istruzione GET se vengono premuti dei tasti quando viene eseguita l'istruzione, la variabile verrà lasciata vuota (0 per una variabile numerica o nullo per una variabile di stringa) se non viene premuto alcun tasto.

Con le variabili numeriche, questa istruzione consente l'immissione di un'unica cifra (da 0 a 9); con le variabili di stringa, consente l'immissione di un unico carattere.

Questa istruzione può essere estremamente utile quando volete immettere dei dati senza premere il tasto **CR**, come spesso avviene con i programmi dei giochi.

(Esempio:)

```

10 PRINT "NEXT GO? (Y OR N) "
20 GET A$
30 IF A$="Y" THEN 50... Nell'esempio precedente, l'esecuzione
                           salta dalla riga 30 alla riga 50 se il
                           valore della variabile A$ è "Y".

40 GOTO 20..... La riga 40 trasferisce incondizionata-
50 PRINT "PROGRAM END " mente l'esecuzione alla riga 20.
60 END

```

Questo programma visualizza il messaggio "NEXT GO? (Y OR N)" e aspetta l'immissione. Quando viene premuto il tasto Y, l'esecuzione passa alla linea 50 ed il programma termina. Sino a tale momento tuttavia, l'esecuzione itera ripetutamente tra le righe 20 e 40. Cancellate ora le righe 30 e 40 ed eseguite nuovamente il programma. Come potete vedere, l'esecuzione viene completata immediatamente indipendentemente dal fatto che sia stato premuto o meno un tasto.

Nota: Quando le istruzioni GET vengono eseguite in successione, deve essere inclusa una routine tra di loro per garantire che ciascuna di esse venga completata prima di passare a quella successiva. Il motivo è che la vibrazione del tasto (vibrazione dei contatti degli interruttori dei tasti) può risultare nell'esecuzione simultanea di due istruzioni GET.

2. 3. 5. 5 READ ~ DATA (formato abbreviato: REA. ~ DA.)

Formato	READ $\left\{ \begin{array}{l} \text{variabile numerica} \\ \text{variabile di stringa} \\ \text{elemento di matrice} \end{array} \right\}, < \left\{ \begin{array}{l} \text{variabile numerica} \\ \text{variabile di stringa} \\ \text{elemento di matrice} \end{array} \right\}, \dots\dots\dots >$ DATA $\left\{ \begin{array}{l} \text{costante numerica} \\ \text{costante di stringa} \end{array} \right\}, < \left\{ \begin{array}{l} \text{costante numerica} \\ \text{costante di stringa} \end{array} \right\}, \dots\dots\dots >$
----------------	--

Funzione	<p>Come le istruzioni INPUT e GET, l'istruzione READ viene usata per presentare i dati al calcolatore per l'elaborazione. Tuttavia, diversamente dall'istruzione INPUT e GET, i dati non vengono immessi dalla tastiera, bensì sono memorizzati nel programma stesso nelle istruzioni DATA. Più specificatamente, la funzione dell'istruzione READ è di leggere i dati successivi nelle variabili da una lista di valori che segue un'istruzione DATA. In questo caso, deve esistere una corrispondenza uno ad uno tra le variabili delle istruzioni READ ed i dati specificati nelle istruzioni DATA.</p>
-----------------	--

Esempi	<p>(Esempio 1)</p> <pre> 1Ø READ A, B, C, D 2Ø PRINT A;B;C;D 3Ø END 4Ø DATA 1Ø, 1ØØ, 5Ø, 6Ø RUN ↵ </pre>
---------------	--

1Ø 1ØØ 5Ø 6Ø In questo esempio, i valori specificati nell'istruzione DATA vengono letti nelle variabili A, B, C e D dall'istruzione READ, quindi vengono visualizzati i valori di tali variabili.

(Esempio 2)

```
10 READ X$, A1, Z$
20 PRINT X$; A1; Z$
30 END
40 DATA A, 1, C .....
```

Come mostrato nell'esempio seguente, i dati di stringa inclusi nelle istruzioni DATA non devono essere racchiusi tra virgolette.

RUN ↵

A_1 C

..... L'istruzione READ di questo esempio prende i dati successivi dalla lista specificata nell'istruzione DATA, quindi sostituisce ogni dato nella variabile corrispondente della lista che segue l'istruzione READ.

(Esempio 3)

```
10 DIM A (2)
20 READ A (0), A (1), A (2)
30 PRINT A (0); A (1); A (2)
40 END
50 DATA 3, 4, 5
```

RUN ↵

3 4 5

..... L'istruzione READ di questo programma sostituisce i valori numerici che seguono l'istruzione DATA negli elementi di matrice A(0), A(1) ed A(2), quindi l'istruzione PRINT sulla riga 30 visualizza i valori di tali elementi di matrice.

(Esempio 4)

```
10 READ A
20 READ B
30 DATA X
```

..... L'esempio precedente non è corretto in quanto (1) è specificata una variabile numerica dall'istruzione READ sulla riga 10, ma il valore specificato dopo l'istruzione DATA è un valore di stringa, e (2) non esistono dati che possono essere letti dall'istruzione READ sulla riga 20.

2. 3. 5. 6 RESTORE (formato abbreviato: RES.)

Formato

RESTORE o RESTORE Ln

Funzione

Quando vengono eseguite le istruzioni READ, viene incrementato un cursore gestito dal programma interprete BASIC per seguire il dato successivo che deve essere letto dalle istruzioni DATA. L'istruzione RESTORE ripristina questo cursore a: (1) l'inizio della prima istruzione DATA del programma o (2) l'inizio dell'istruzione DATA di una riga specificata.

Esempio

```
10 DATA 1, 2, 3
20 DATA "AA", "BB"
30 READ X, Y
40 READ Z, V$
.....
100 RESTORE
110 READ A, B, C, D$, E$
.....
200 READ I, J
210 RESTORE
220 READ M, N
230 RESTORE 260
240 READ O, P
250 DATA 1, 2, 3, 4
260 DATA -1, -2, -3, -4
```

Si ha un errore se il numero specificato in Ln è il numero di una riga che non esiste.

```
10 X=33*RND(1)
20 FOR A=1 TO 5
30 READ M$
40 PRINT TAB(0); "◆"; TAB(X); M$;
50 PRINT TAB(37); "◆"
60 NEXT A
70 Y=10*RND(1)
80 FOR A=1 TO Y
90 PRINT TAB(0); "◆";
100 PRINT TAB(37); "◆":NEXT
110 RESTORE:GOTO 10
120 DATA " ■■■", " ●■■■■●"
130 DATA " ■■■", " ●■■■■●"
140 DATA " ■■■"
```

Questa funzione crea dei numeri casuali (vedere pagina 71).

Nota: Vedere pagina 60 per la funzione TAB e pagina 45 per l'istruzione FOR ~ NEXT.

2.3.6 Istruzioni di iterazione e di salto

2.3.6.1 FOR ~ NEXT (formato abbreviato: F. ~ N.)

Esempio

```
FOR cv = iv TO fv < STEP sv >
```

```
.....  
NEXT < cv >
```

cv Variabile di controllo; una variabile o un elemento di matrice numerici.

iv Valore iniziale; un'espressione numerica.

fv Valore finale; un'espressione numerica.

sv Incremento o valore di passo; un'espressione numerica (se è omessa, si assume 1).

Funzione

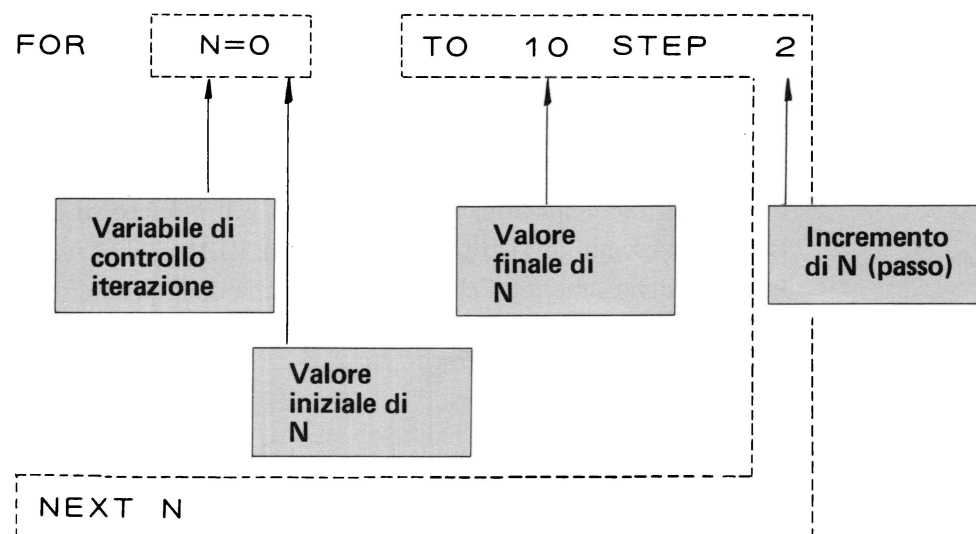
Questa istruzione ripete per un certo numero di volte le istruzioni tra FOR e NEXT.

```
10 A=0  
20 FOR N=0 TO 10 STEP 2  
30 A=A+1  
40 PRINT "N=" ; N,  
50 PRINT "A=" ; A  
60 NEXT N
```

- (1) Nel programma precedente, viene assegnato 0 ad N come valore iniziale.
- (2) Quindi, vengono eseguite le righe 20 ~ 50 e vengono visualizzati i valori delle variabili A ed N.
- (3) Nella riga 60, il valore di N viene aumentato di 2, dopodiché il programma interprete BASIC controlla per vedere se N è superiore a 10, il valore finale. Se non è superiore a 10, le righe dopo la riga 20 vengono ripetute.

Quando il valore di N supera 10, l'esecuzione lascia l'iterazione e vengono eseguite le istruzioni successive (sulle righe dopo la riga 60). Il programma precedente ripete l'iterazione 6 volte.

Se non si inserisce < STEP sv > nella specifica dell'istruzione, il valore di N viene aumentato di 1 ogni volta che l'iterazione viene ripetuta. Nel caso del programma precedente, l'omissione di < STEP sv > porta ad 11 ripetizioni dell'iterazione.



Le iterazioni FOR ~ NEXT possono essere annidate all'interno di altre iterazioni FOR ~ NEXT. In tal caso, le iterazioni più interne devono essere totalmente incluse in quelle più esterne. Inoltre, si devono usare delle variabili di controllo separate per ogni iterazione.

Formato

```

10 FOR X=1 TO 9
20 FOR Y=1 TO 9
30 PRINT X*Y;
40 NEXT Y
50 PRINT
60 NEXT X
70 END

```

Iterazione interna
Iterazione esterna

```

FOR A=1 TO 3
FOR B=1 TO 5
FOR C=1 TO 7
.....
NEXT C
NEXT B
NEXT A

```

} NEXT C, B, A

Quando le iterazioni C, B ed A terminano tutte nello stesso punto come nell'esempio precedente, si può usare un'istruzione NEXT per indicare la fine di tutte le iterazioni.

Esempio errato:

```

FOR J=1 TO 10
FOR J=K TO K+S
NEXT J

```

```

FOR I=1 TO 10
FOR J=K TO K+5
NEXT I
NEXT J

```

✗ In ogni iterazione devono essere usate delle variabili di controllo diverse.

✗ Le iterazioni non possono incrociarsi.

Nota

La sintassi del BASIC non limita il numero di livelli ai quali è possibile annidare le iterazioni; tuttavia, è necessario dello spazio per memorizzare gli indirizzi di ritorno per ogni livello, cosicché il numero di livelli è limitato dalla quantità di spazio libero disponibile. L'istruzione CLR (vedere pagina 58) non può essere usata all'interno di un'iterazione FOR ~ NEXT.

2. 3. 6. 2 GOTO (formato abbreviato: G.)

Formato

GOTO Ln

Ln: Numero della riga di destinazione

Funzione

Quest'istruzione trasferisce incondizionatamente l'esecuzione del programma al numero di righe specificato in Ln. Se Ln è il numero di una riga che contiene delle istruzioni eseguibili (istruzioni diverse da REM o DATA), l'esecuzione riprende da tale riga; diversamente, l'esecuzione riprende dalla prima istruzione eseguibile dopo il numero di riga Ln.

Esempio

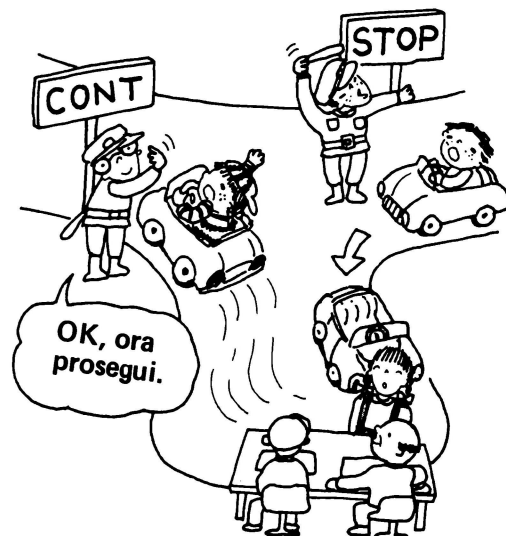
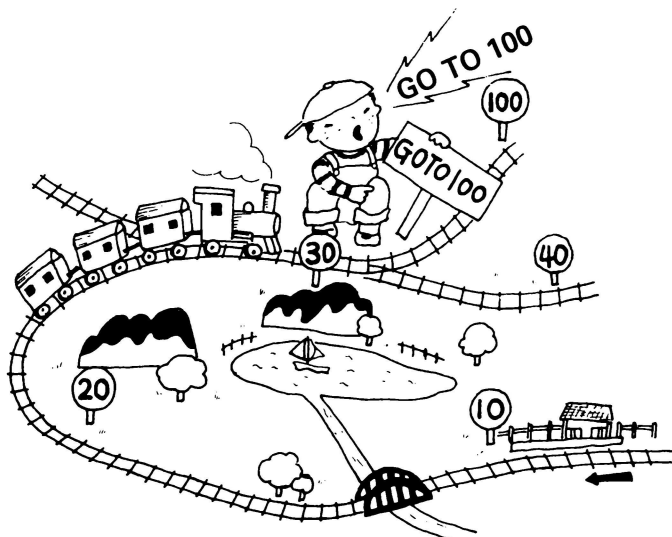
```

10 N=1
20 PRINT N
30 N=N+1
40 GOTO 20
50 END

```

..... Trasferisce l'esecuzione del programma alla riga 20.

Poiché l'esecuzione del programma precedentemente mostrato continua indefinitamente, fermatela premendo contemporaneamente i tasti **SHIFT** e **BREAK** (questo può essere fatto in qualsiasi momento per arrestare l'esecuzione di un programma BASIC). Per riprendere l'esecuzione, eseguite il comando **CONT** .



Nota

Il numero di riga specificato in un'istruzione **GOTO** non può essere quello di una riga inclusa in un'iterazione **FOR ~ NEXT**.

2. 3. 6. 3 **GOSUB ~ RETURN** (formato abbreviato: **GOS. ~ RET.**)

Formato

GOSUB Ln

RETURN

Ln ... Numero della riga di destinazione

Funzione

L'istruzione **GOSUB** trasferisce incondizionatamente l'esecuzione del programma ad una sottoroutine BASIC che comincia al numero di riga specificato in **Ln**; dopo che l'esecuzione della sottoroutine è stata completata, l'esecuzione ritorna all'istruzione che segue **GOSUB** quando viene eseguita un'istruzione **RETURN**.

Le istruzioni **GOSUB ~ RETURN** vengono usate frequentemente quando è necessaria la stessa elaborazione in parecchi punti diversi di un programma. In tali casi, viene inclusa in un certo punto del programma una sottoroutine che esegue queste elaborazioni, e l'esecuzione viene fatta saltare a questa sottoroutine nei punti appropriati mediante l'istruzione **GOSUB**. Dopo che è stata completata l'elaborazione richiesta, l'esecuzione viene riportata alla routine principale mediante l'istruzione **RETURN**.

Esempio

```

100 X=10
110 GOSUB 200
120 PRINT X
130 END
200 X=X*2
210 RETURN

```

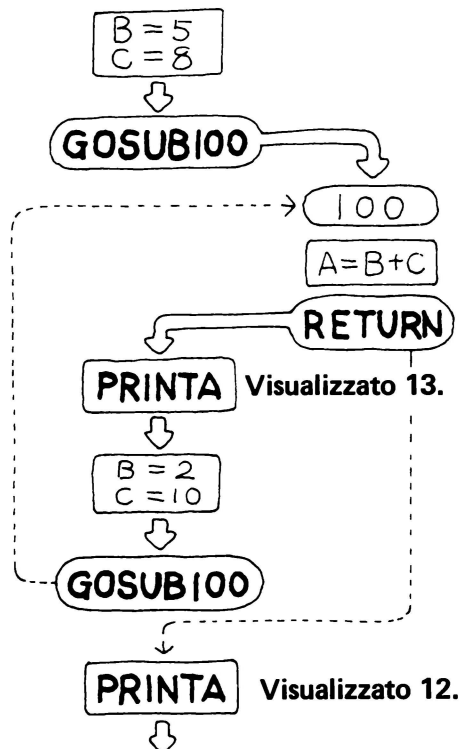
Nota

La sintassi del BASIC non impone alcun limite all'entità del possibile annidamento delle sottoroutine (cioé, al numero di livelli di chiamate di sottoroutine che possono essere fatte da altre sottoroutine); tuttavia, in pratica la limitazione viene imposta dalla quantità di spazio libero in memoria disponibile per memorizzare indirizzi di ritorno.

```

10 B=5
20 C=8
30 GOSUB 100
40 PRINT A
50 B=2
60 C=10
70 GOSUB 100
80 PRINT A
90 END
100 A=B+C
110 RETURN

```



2. 3. 6. 4 IF ~ THEN (formato abbreviato: IF ~ TH.)

Funzione

IF e THEN Ln

IF e THEN, istruzione

e: Espressione di confronto o espressione logica

Ln: Numero della riga di destinazione

Funzione

Le istruzioni IF ~ THEN vengono usate per controllare il salto dell'esecuzione del programma conformemente al risultato di un'espressione logica o di confronto. Quando il risultato di tale espressione è vero, vengono eseguite le istruzioni che seguono THEN. Se dopo THEN viene specificato un numero di riga, l'esecuzione del programma salta a tale riga del programma se il risultato dell'espressione è vero. Se il risultato dell'espressione logica o di confronto è falso, l'esecuzione continua con la riga del programma successiva a quella contenente l'istruzione IF ~ THEN.

IF

Condizione

THEN

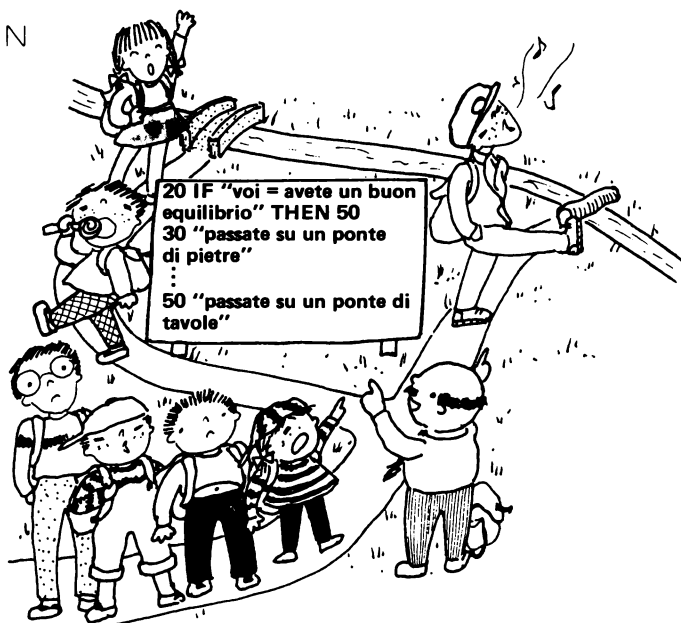
Istruzione o numero di riga

Esempio

```

IF.....THEN 100
IF.....THEN GOTO o IF.....GOTO
IF.....THEN PRINT o IF.....THEN ?
IF.....THEN A=5*7 assegnazione
IF.....THEN I=10: J=50
IF.....THEN INPUT
IF.....THEN READ
IF.....THEN GOSUB
IF.....THEN RETURN
IF.....THEN STOP
IF.....THEN END

```



Esempi di espressioni logiche e di confronto

	Operatore	Applicazione campione	Spiegazione
Espressioni di confronto	=	IF A=X THEN...	Se il valore della variabile numerica A è uguale al valore di X, esegui le istruzioni che seguono THEN.
		IF A\$="XYZ" THEN...	Se il contenuto della variabile di stringa A\$ è uguale al valore di "XYZ", esegui le istruzioni che seguono THEN.
	>	IF A>X THEN...	Se il valore della variabile A è superiore ad X, esegui le istruzioni che seguono THEN.
	<	IF A<X THEN...	Se il valore della variabile A è inferiore ad X, esegui le istruzioni che seguono THEN.
	<> or ><	IF A<>X THEN...	Se il valore della variabile A non è uguale ad X, esegui le istruzioni che seguono THEN.
	>= or =>	IF A>=X THEN...	Se il valore della variabile A è superiore o uguale ad X, esegui le istruzioni che seguono THEN.
Espressioni logiche	<= or =<	IF A<=X THEN...	Se il valore della variabile A è inferiore o uguale ad X, esegui le istruzioni che seguono THEN.
	*	IF(A>X)*(B>Y) THEN...	Se il valore della variabile A è superiore ad X ed il valore della variabile B è superiore ad Y, esegui le istruzioni che seguono THEN.
	+	IF(A>X)+(B>Y) THEN...	Se il valore della variabile A è superiore ad X oppure il valore della variabile B è superiore ad Y, esegui le istruzioni che seguono THEN.

Nota

Precauzioni nel confronto dei valori numerici. Con il BASIC 1Z-013B, i valori numerici sono rappresentati internamente in rappresentazione binaria con virgola mobile; poiché tali valori devono essere convertiti in altre forme per l'elaborazione o la visualizzazione esterna (per esempio in formato decimale con l'istruzione PRINT), può verificarsi una certa quantità di errori di conversione.

Per esempio, quando viene valutata un'espressione aritmetica il cui risultato matematico è un numero intero, può non essere ritornato un valore intero al completamento dell'operazione se vengono trattati dei valori diversi dai numeri interi quando si effettuano i calcoli. Fate particolare attenzione a tenere conto di questo fatto quando valutate le espressioni di confronto usando "=".

Questa necessità è illustrata dal programma campione seguente, che ritorna FALSE dopo aver provato l'uguaglianza tra 1 e $1/100 \times 100$.

```
10 A=1/100*100
20 IF A=1 THEN PRINT "TRUE":GOTO 40
30 PRINT "FALSE"
40 PRINT "A=";A
50 END
```

```
RUN
FALSE
A=1
```

Il fatto che venga visualizzato sia "FALSE" che "A = 1" come risultato di questo programma, dimostra che la rappresentazione esterna dei numeri può differire dalla rappresentazione interna dei numeri.

Pertanto, un metodo migliore per controllare l'uguaglianza nell'esempio di programma precedente è il seguente.

```
20 IF ABS(A-1) < .1E-8 THEN PRINT "TRUE":
   GOTO 40
```

2. 3. 6. 5 IF ~ GOTO (formato abbreviato: IF ~ G.)

Formato

IF e GOTO Lr

e: Espressione di confronto o espressione logica

Lr: Numero della riga di destinazione.

Funzione

Questa sequenza di istruzioni valuta la condizione definita dall'espressione di confronto o logica e, quindi salta al numero di riga specificato in Lr se la condizione è soddisfatta. Come con la sequenza IF ~ THEN, IF ~ GOTO viene usata per il salto condizionato; quando la condizione specificata è soddisfatta, l'esecuzione del programma salta al numero di riga specificato in Lr. Se la condizione non è soddisfatta, l'esecuzione continua con la riga successiva del programma. (Eventuali istruzioni dopo IF ~ GOTO sulla stessa riga di programma verranno ignorate).

Formato

```
10 G=0:N=0
20 INPUT "GRADE=" ; X
30 IF X=999 GOTO 100
40 T=T+X:N=N+1
50 GOTO 20
100 PRINT "-----"
110 PRINT "TOTAL:" ; T
120 PRINT "NO. PEOPLE:" ; N
130 PRINT "AVERAGE:" ; T/N
140 END
```

2. 3. 6. 6 IF ~ GOSUB (formato abbreviato: IF ~ GOS.)

Formato

IF e GOSUB Lr

e: Espressione di confronto o espressione logica

Lr: Numero della riga di destinazione

Funzione

Questa istruzione valuta la condizione definita dall'espressione di confronto o logica e, quindi, se la condizione è soddisfatta, salta alla sottoroutine che comincia al numero di riga specificato in Lr. Al completamento della sottoroutine, l'esecuzione ritorna alla prima istruzione eseguibile dopo l'istruzione di richiamo IF ~ GOSUB; pertanto, se diverse istruzioni sono incluse nella riga con l'istruzione IF ~ GOSUB, l'esecuzione ritorna alla prima istruzione che segue IF ~ GOSUB.

Esempio

```
10 INPUT " X= " ; X
20 IF X>0 GOSUB 100:PRINT "X<0"
30 IF X=0 GOSUB 200:PRINT "X=0"
40 IF X>0 GOSUB 300:PRINT "X>0"
50 PRINT "-----"
60 GOTO 10
100 PRINT " * PROGRAM LINE 100 " : RETURN
200 PRINT " * PROGRAM LINE 200 " : RETURN
300 PRINT " * PROGRAM LINE 300 " : RETURN
```


2. 3. 6. 7. ON ~ GOTO (formato abbreviato: ON ~ G.)

Formato

ON e GOTO $Lr_1 < , Lr_2, Lr_3, \dots, Lr_i >$

e ... Variabile, elemento di matrice o espressione numerici.

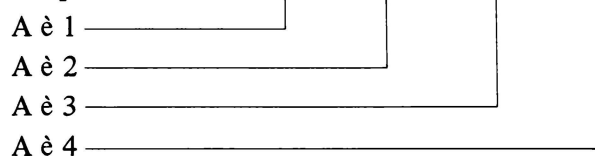
Lr_i . Lista dei numeri delle righe di destinazione

Funzione

Questa istruzione salta l'esecuzione ad uno dei numeri di riga che seguono GOTO, a seconda del valore di e. Il valore di e indica quali dei numeri di riga che seguono GOTO deve essere usato per il salto; in altre parole, se e è 1, l'esecuzione salta al primo numero di riga della lista; se e è 2, l'esecuzione salta al secondo numero di riga della lista; e così via. Per esempio:

```
100 ON A GOTO 200, 300, 400, 500
```

Destinazione quando



Esempio

```
10 INPUT "NUMBER" ; A  
20 ON A GOTO 50, 60, 70  
50 PRINT "XXX" : GOTO 10  
60 PRINT "YYY" : GOTO 10  
70 PRINT "ZZZ" : GOTO 10
```

RUN

NUMBER ? 1

XXX

NUMBER ? 2

YYY

NUMBER ? ❏

Se viene specificato un numero decimale come 1, 2, la parte decimale viene troncata prima della valutazione dell'istruzione.

Nota

Quando il valore di e in un'istruzione ON ~ GOTO è superiore al numero dei numeri di riga specificati dopo GOTO, l'esecuzione continua con la riga successiva del programma.

Questo vale anche se il valore di e è inferiore ad 1 o negativo.

Inoltre, se il valore di e è un numero non intero, la parte decimale viene troncata per ottenere un valore intero prima della valutazione dell'istruzione.

2. 3. 6. 8 ON ~ GOSUB (formato abbreviato: ON ~ GOS.)

Formato

ON e GOSUB Lr₁ < , Lr₂, Lr₃, , Lr_i >

e . . . Variabile, elemento di matrice o espressione numerici

Lr_i . Numeri delle righe di destinazione

Funzione

Questa istruzione fa saltare l'esecuzione alla sottoroutine che comincia ad uno dei numeri di riga che seguono GOSUB, a seconda del valore di e. Il funzionamento di questa istruzione è fondamentalmente uguale a quello della funzione ON ~ GOTO, ma tutti i salti sono fatti alle sottoroutine. Al ritorno dalla sottoroutine, l'esecuzione riprende dalla prima istruzione eseguibile dopo l'istruzione ON ~ GOSUB che ha effettuato la chiamata.

Esempio

Proviamo ad usare l'istruzione ON ~ GOSUB in un programma di calendario. Il punto più importante da notare nel programma seguente è che viene fatta una chiamata di sottoroutine alla riga 180, anche se la riga 180 stessa fa parte di una sottoroutine (dalla riga 170 alla riga 190) che è chiamata dalla riga 90. Le sottoroutine possono essere annidate a molti livelli in questo modo.

```
10 A$=" ENGL " :B$=" MATH " :C$=" FREN "
20 D$=" SCI " :E$=" MUS " :F$=" GYM "
30 G$=" HIST " :H$=" ART " :I$=" GEOG "
40 J$=" BUS " :K$=" H RM " :PRINT "■"
50 INPUT "WHAT DAY? ";X$
60 FOR Z=1 TO 7:Y$=MID$("SUNMONTUEWEDTHU
   FRISAT", 1+3*(Z-1), 3) : IF Y$=X$ THEN X=Z
70 NEXT Z
80 FOR Y=0 TO 4:PRINT TAB(5+6*Y);Y+1;
90 NEXT Y:PRINT
100 ON X GOSUB 180,120,130,140,150,160,170
110 PRINT:GOTO 50
120 PRINT"MON " ;A$;B$;D$;G$;K$:RETURN
130 PRINT"TUE " ;B$;E$;H$;H$;D$:RETURN
140 PRINT"WED " ;C$;C$;I$;A$;F$:RETURN
150 PRINT"THU " ;B$;D$;F$;G$;E$:RETURN
160 PRINT"FRI " ;A$;D$;I$;C$;C$:RETURN
170 PRINT"SAT " ;B$;G$;D$;K$:RETURN
180 FOR Y=1 TO 6
190 ON Y GOSUB 120,130,140,150,160,170
200 PRINT:NEXT Y
210 RETURN
```

2.3.7 Istruzioni di definizione

2.3.7.1 DIM

Formato

DIM $a_1(i_1) <, a_2(i_2), \dots, a_i(im) >$

DIM $b_1(i_1, j_1) <, b_2(i_2, j_2), \dots, b_i(in, jn) >$

a_i Nome della matrice unidimensionale (lista)

b_i Nome della matrice bidimensionale (tabella)

im, in, jn Dimensioni

Funzione

Questa istruzione viene usata per dichiarare (definire) le matrici con 1 ~ 4 dimensioni e per riservare dello spazio in memoria per il numero di dimensioni dichiarate (DIM: dimensione). E' possibile specificare per il nome della matrice un massimo di 2 caratteri, e si possono specificare dei subscripti di qualsiasi valore per definire l'ampiezza delle dimensioni; tuttavia, il numero delle dimensioni che si possono usare è limitato in pratica dalla quantità di memoria libera disponibile.

Esempi

(Esempio 1)

```
10 DIM A (100)
20 FOR J=0 TO 100
30 READ A (J)
40 NEXT J
50 DATA 5, 30, 12, .....
```

(Esempio 2)

```
10 DIM A$ (1), B$ (1), C$ (1)
20 FOR J=0 TO 1 : READ A$ (J), B$ (J)
30 C$ (J) = A$ (J) + " " + B$ (J)
40 PRINT A$ (J), B$ (J), C$ (J)
50 NEXT J
60 END
70 DATA YOUNG, GIRL, WHITE, ROSE
```

Nota

L'esecuzione dell'istruzione DIM fissa i valori di tutti gli elementi delle matrici dichiarate a 0 (per le matrici numeriche) o nullo (per le matrici di stringa). Pertanto, questa istruzione deve essere eseguita prima di assegnare i valori alle matrici.

Tutte le dichiarazioni di matrice vengono annullate dall'esecuzione di un'istruzione CLR (vedere pagina 58) e di un'istruzione NEW (vedere pagina 30).

2.3.7.2 DEF FN

Formato

DEF FN $f(x) = e$

f .. Nome assegnato alla funzione che viene definita (una lettera maiuscola da A ~ Z)

x .. Argomento (nome variabile)

e .. Espressione numerica (costante, variabile, elemento di matrice o funzione) o funzione dell'utente precedentemente definita.

Funzione

L'istruzione DEF FN viene usata per definire la funzione utente FN $f(x)$. Tali funzioni sono costituite da combinazioni di funzioni intrinseche nel BASIC.

Esempio

DEF FNA (X) = 2 * X ↑ 2 + 3 * X + 1 ... 2X² + 3X + 1 come FNA (X).
 DEF FNE (V) = 1 / 2 * M * V ↑ 2 Definisce 1/2MV² come FNE (V).

10 DEF FNB (X) = TAN (X - PAI (1) / 6)

20 DEF FND (X) = FNB (X) / C + X. Definisce la funzione FNB usando la funzione definita nella riga 10.

(Definizioni non corrette)

10 DEF FNK (X) = SIN (X / 3 + PAI (1) / 4), FNL (X) = EXP (-X ↑ 2 / K)
 E' possibile definire una sola funzione utente mediante una singola istruzione DEF FN.

Trovate l'energia cinetica di una massa di 5.5 quando le vengono impartite delle accelerazioni iniziali di 3.5, 3.5 x 2, e 3.5 x 3.

```
10 DEF FNE (V) = 1 / 2 * M * V ↑ 2
20 M = 5.5 : V = 3.5
30 PRINT FNE (V), FNE (V * 2), FNE (V * 3)
40 END
```

Nota

Tutte le definizioni di funzione utente vengono annullate quando viene eseguita l'istruzione CLR e l'istruzione NEW.

2. 3. 7. 3 DEF KEY**Formato**

DEF KEY (k) = S\$

k Numero di tasto funzionale definibile (1 ~ 10)

S\$ Stringa di caratteri (fino a 15 caratteri).

Funzione

Le stringhe di caratteri possono essere assegnate ad uno qualsiasi dei 10 tasti funzionali, per consentire l'immissione delle stringhe in qualsiasi momento solo premendo un unico tasto. Questa istruzione viene usata per definire tali stringhe ed assegnarle ai tasti funzionali definibili. I numeri dei tasti funzionali 1 ~ 5, vengono immessi premendo il tasto corrispondente nell'angolo superiore di sinistra della tastiera; i tasti 6 ~ 10 vengono immessi premendo il tasto **SHIFT** insieme al tasto corrispondente. Il numero del tasto funzionale (1 ~ 10) viene specificato in k, e la stringa (o il comando) che deve essere assegnata al tasto viene specificata esattamente come deve essere immessa in S\$. L'esecuzione dell'istruzione DEF KEY annulla la definizione precedente del tasto funzionale definibile.

Non è possibile specificare alcuna altra istruzione dopo l'istruzione DEF KEY sulla stessa riga.

(Esempio:)

10 DEF KEY (1) = " INPUT " Definisce il tasto **F1** come INPUT

20 DEF KEY (2) = " RUN " + CHR\$ (13) . . Definisce il tasto **F2** come RUN J .

Note: CHR\$ (13) indica il codice ASCII di **CR**, e la sua specificazione insieme alla stringa assegnata ad un tasto funzionale definibile, ha lo stesso effetto che si ottiene premendo il tasto **CR**. (Vedere la descrizione della funzione CHR\$ a pagina 77 e la tabella dei codici ASCII a pagina 156.

2.3.8 Frase di commento e comandi di controllo

2.3.8.1 REM

Formato

REM r

r Commento del programmatore

Funzione

REM è un'istruzione non eseguibile che è specificata in una riga di programma per ottenere che il programma interprete BASIC ignori il resto di tale riga. Poiché le istruzioni REM non sono eseguibili, possono essere inclusi in qualsiasi punto del programma senza influenzare i risultati dell'esecuzione. Le istruzioni REM vengono generalmente usate per facilitare la lettura di un programma o per aggiungere delle note esplicative ad un programma. L'apostrofo (') è la forma abbreviata dell'istruzione REM, e specificando un apostrofo all'inizio di una riga di programma si ottiene lo stesso effetto ottenuto specificando REM.

Righe di programma con istruzioni multiple

Quando in una singola riga di programma è inclusa più di un'istruzione, ogni istruzione deve essere separata da quella precedente da un due punti (:). Il funzionamento del programma interprete BASIC è generalmente uguale in tali casi come quando le stesse istruzioni vengono specificate su righe diverse.

Per esempio, i due programmi seguenti producono esattamente lo stesso risultato.

10 A=5	}	→	10 A=5:B=8:C=A*B:PRINT C
20 B=8			
30 C=A*B			
40 PRINT C			

Tuttavia non è possibile includere le istruzioni eseguibili sulla stessa riga di un'istruzione REM a meno che l'istruzione REM sia l'ultima istruzione di tale riga. Pertanto, i programmi che includono delle righe come quella seguente non funzionano correttamente.

```
10 REM PRODOTTO DI A E B:A=5:B=8:C=A*B:PRINT C
```

Nei casi come questo, usate la forma seguente.

```
10 REM PRODOTTO DI A E B
20 A=5:B=8:C=A*B:PRINT C
```

Notate inoltre che il funzionamento del programma può differire quando si usano delle righe di istruzioni multiple come mostrato qui di seguito.

10 INPUT A	}	Questo programma visualizza 1 se il valore immesso nella riga 10 è superiore o uguale a 100, e 0 se il valore immesso è inferiore a 100.
20 B=0		
30 IF 99<A THEN B=1		
40 PRIN B		
50 END		

```
10 INPUT A:B=0:IF 99<A THEN B=1:PRINT B
20 END
```

Questo programma visualizza 1 se il valore immesso è superiore o uguale a 100, ma non visualizza niente se il valore immesso è inferiore a 100. Il motivo è che le istruzioni che seguono THEN nella riga 10 non vengono eseguite se la condizione IF non è soddisfatta.

2. 3. 8. 2 STOP (formato abbreviato: S.)

Formato

STOP

Funzione

Arresta temporaneamente l'esecuzione del programma, visualizza BREAK, e READY, quindi attende l'immissione dei comandi da eseguire nel modo diretto.

L'istruzione STOP viene usata per interrompere temporaneamente l'esecuzione del programma, e può essere inserita in tutti i punti e le posizioni del programma necessari. Poiché l'esecuzione del programma viene interrotta solo temporaneamente, è possibile usare l'istruzione PRINT nel modo diretto per controllare i valori memorizzati nelle variabili, dopodiché l'esecuzione può essere ripresa immettendo CONT.

Esempio

```
10 READ A, B
20 X=A*B
30 STOP
40 Y=A/B
50 PRINT X, Y
60 DATA 15, 5
70 END
RUN
BREAK IN 30
```

Nota

Diversamente dall'istruzione END, nessun file viene chiuso dall'istruzione STOP. (Vedere pagina 67 relativa alle procedure per l'apertura e la chiusura dei file).

2. 3. 8. 3 END (formato abbreviato: E.)

Formato

END

Funzione

L'istruzione END pone termine all'esecuzione del programma e riporta il programma interprete BASIC al modo comando per l'immissione dei comandi in modo diretto. Quando questa istruzione viene eseguita, viene visualizzato READY per indicare che il programma interprete BASIC è pronto. Dopo che è stata eseguita l'istruzione END, non è possibile riprendere l'esecuzione eseguendo il comando CONT anche se vi sono delle istruzioni esecutive sulle righe di programma che seguono l'istruzione END.

Nota

Tutti i file aperti vengono chiusi quando viene eseguita l'istruzione END. (Vedere pagina 67 per quanto concerne le procedure per aprire e chiudere i file).

Differenze tra le istruzioni STOP ed END

	Schermo video	File	Ripresa dell'esecuzione
STOP	BREAK IN xxxx READY	I file aperti non vengono chiusi	L'esecuzione può essere ripresa eseguendo CONT.
END	READY	I file aperti vengono chiusi	L'esecuzione non può essere ripresa.

2. 3. 8. 4 CLR

Formato

Funzione

CLR

Il comando CLR annulla tutte le variabili e tutte le definizioni di matrice. Tutte le variabili numeriche vengono azzerate, e le stringhe nulle (" ") vengono messe in tutte le variabili di stringa; le matrici vengono eliminate interamente annullando tutte le istruzioni DIM precedentemente eseguite. Pertanto, le istruzioni DIM devono essere eseguite per definire nuovamente le dimensioni delle matrici necessarie prima che queste possano essere nuovamente usate.

Il comando CLR annulla anche tutte le definizioni di funzione fatte con l'istruzione DEF FN; pertanto, è necessario anche eseguire nuovamente le istruzioni DEF FN per ridefinire tali funzioni prima che possano essere nuovamente usate.

Nota

Non è possibile includere le istruzioni CLR in una iterazione FOR ~ NEXT o in una iterazione FOR ~ NEXT o in una sottoroutine BASIC.

2. 3. 8. 5 TI\$

Formato

Funzione

TI\$ "hh mm ss"

TI\$ è il nome della variabile di stringa del sistema che contiene l'ora dell'orologio incorporato nel calcolatore. Questa variabile incorporata viene incrementata automaticamente una volta al secondo, e la stringa di 6 caratteri contenuta in questa variabile indica l'ora, il minuto ed il secondo e per ciascuna di queste voci vengono usati due caratteri. Per esempio, se la stringa contenuta in TI\$ è "092035", l'ora è 9:20:35. La variabile TI\$ viene fissata automaticamente a 00:00:00 quando il BASIC viene caricato nel calcolatore. Per determinare l'ora attuale del giorno, usate l'istruzione di assegnazione stringa. Per esempio, l'orologio può essere regolato sulle 19:00:00 eseguendo quanto segue.

TI\$ = " 190000 "

L'orologio viene regolato su 19:00:00 e quindi riavviato automaticamente quando viene premuto il tasto CR.

Le cifre specificate per l'ora devono essere comprese nella gamma tra 00 e 23, mentre quelle specificate per i minuti e per i secondi devono essere entrambe nella gamma compresa tra 00 e 59.

Esempio

Il programma seguente visualizza l'ora locale attuale in diverse città del mondo.

```

10 PRINT "■"
20 DIM C$(10), D(10), E(10), T$(10)
30 FOR I=1 TO 10:READ C$(I), D(I):NEXT I
40 PRINT "ENTER NEW YORK TIME (HOUR, MINUT
    E, SECOND) "
50 INPUT B$:TI$=B$:PRINT "■"
60 PRINT "■":T$(1)=TI$
70 FOR I=1 TO 10
80 E(I)=VAL(LEFT$(T$(1), 2))+D(I)
90 IF E(I)=24 THEN E(I)=0
100 IF E(I)<0 THEN E(I)=24+E(I)
110 T$(1)=STR$(E(I))+RIGHT$(T$(1), 4)
120 IF LEN(T$(1))=5 THEN T$(1)="0"+T$(1)
130 PRINT C$(I);TAB(15);LEFT$(T$(1), 2);
140 PRINT": ";MID$(T$(1), 3, 2);": ";RIGHT$(
    T$(1), 2);
150 NEXT I:GOTO 60
160 DATA NEW YORK, 0, MOSCOW, 8, RIO DE JANE
    IRO, 2
170 DATA SYDNEY, 15, HONOLULU, -5, LONDON, 5,
    CAIRO, 7
180 DATA TOKYO, 14, SAN FRANCISCO, -3, PARIS
    , 6

```

Nota

La variabile TI\$ non può essere specificata in un'istruzione INPUT. Inoltre, dopo che l'ora passa da 23:59:59 a 00:00:00, non viene visualizzata l'ora "00:00:01".

2.3.8.6 CURSOR (formato abbreviato: CU.)

Formato

CURSOR x, y

x ... Coordinata X (0 ~ 39)

y ... Coordinata Y (0 ~ 24)

Funzione

Questo comando viene usato per spostare il cursore su di una posizione specificata sullo schermo televisivo, e può essere usato insieme alle istruzioni PRINT e INPUT per visualizzare i caratteri in qualsiasi posizione desiderata.

Nel sistema di coordinate schermo usato, le colonne dello schermo sono numerate da sinistra verso destra, partendo da 0 sul lato sinistro e terminando con 39 sul lato destro; le righe dello schermo sono numerate dall'alto al basso, dove 0 indica la riga superiore dello schermo e 24 indica la riga inferiore. Pertanto, il cursore può essere spostato su qualsiasi posizione desiderata nella gamma da (0, 0), che indica l'angolo superiore di sinistra dello schermo, a (39, 24) che indica l'angolo inferiore di destra.

Esempio

Il programma seguente sposta un asterisco (*) sullo schermo quando vengono premuti i tasti di controllo del cursore.

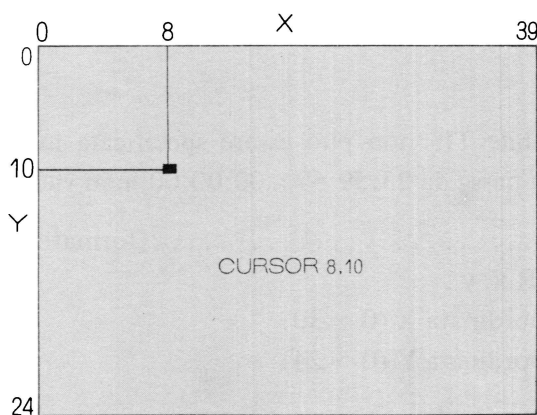
```

10 X=0:Y=0
15 PRINT " "
20 CURSOR X,Y:PRINT "*";
30 GET A$:IF A$=" " THEN 30
40 CURSOR X,Y:PRINT " ";
50 IF A$="↑" THEN Y=Y-1 :REM "UP"
60 IF A$="↓" THEN Y=Y+1 :REM "DOWN"
70 IF A$="←" THEN X=X-1 :REM "LEFT"
80 IF A$="→" THEN X=X+1 :REM "RIGHT"
90 IF X<0 THEN X=0
100 IF Y<0 THEN Y=0
110 IF X>38 THEN X=38
120 IF Y>24 THEN Y=24
150 GOTO 20

```

Nota

Se il valore specificato per X o per Y è diverso da un numero intero, viene convertito in un numero intero troncando la parte decimale prima di spostare il cursore. Altri metodi per spostare il cursore che vengono usati insieme all'istruzione PRINT, includono le funzioni TAB e SPC. (Vedere pagina 61 per la descrizione della funzione SPC).

**2. 3. 8. 7 TAB****Formato****TAB (x)**

x ... Espressione numerica

Funzione

La funzione TAB viene usata insieme all'istruzione PRINT per spostare il cursore sulla posizione del carattere che si trova ad x + 1 posizioni dal lato sinistro dello schermo. (Questa operazione viene chiamata tabulazione per spazi).

Esempio

```

PRINT TAB (5) ; " XYZ " ; TAB (10) ; " ABC "
0 1 2 3 4 5 6 7 8 9 0 1 2
_ _ _ _ _ XYZ _ _ ABC

```

Non effettivamente visualizzato

Nota

La tabulazione può essere usata soltanto per spostare il cursore a destra; pertanto, non avviene niente se questa funzione viene usata unitamente all'istruzione **PRINT** quando il cursore è già alla destra della posizione del carattere specificata in (x).

(Esempio:)

```
PRINT TAB (5) ; " XYZ " ; TAB (5) ; " ABC "
```

```
0 1 2 3 4 5 6 7 8 9 0
```

```
    _ _ _ _ _ XYZ ABC
```

2. 3. 8. 8 SPC**Formato**

SPC (n)

n ... Espressione numerica

Funzione

Usata insieme all'istruzione **PRINT**, questa funzione produce una stringa di n spazi e pertanto muove il cursore di n posizioni di carattere a destra della sua posizione attuale.

Esempi

(Esempio 1)

```
PRINT SPC (5) ; " ABC "
```

```
0 1 2 3 4 5 6 7
```

```
    _ _ _ _ _ ABC
```

(Esempio 2)

L'esempio seguente illustra la differenza tra le funzioni **TAB** ed **SPC**.

```
1 Ø ? TAB (2) ; " ABC " ; TAB (6) ; " DEF "
```

```
2 Ø ? SPC (2) ; " ABC " ; SPC (6) ; " DEF "
```

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3
```

```
_ _ ABC _ DEF _ _ _ _ _
```

```
_ _ ABC _ _ _ _ _ DEF
```

2. 3. 8. 9 SET, RESET

Queste istruzioni vengono usate per attivare o disattivare i puntini in una posizione specifica dello schermo.

Formato	Funzione	Gamma delle coordinate X/Y
SET X, Y <, C > X ... Espressione numerica che specifica la coordinata X. Y ... Espressione numerica che specifica la coordinata Y. C ... Codice colore (0 ~ 7).	Accendet i puntini sulle coordinate del video specificate da X e Y. (SET)	$0 \leq X \leq 79$ $0 \leq Y \leq 49$
RESET X, Y X ... Espressione numerica che specifica la coordinata X. Y ... Espressione numerica che specifica la coordinata Y.	Pisattira i puntini in corrispondenza delle coordinate dello schermo specificate da X ed Y. (RESET)	$0 \leq X \leq 79$ $0 \leq Y \leq 49$

Quando si specifica un codice di colore, il colore dei puntini visualizzati dall'istruzione SET è il seguente.

- (0) Nero
- (1) Blu
- (2) Rosso
- (3) Porpora
- (4) Verde
- (5) Azzurro
- (6) Giallo
- (7) Bianco

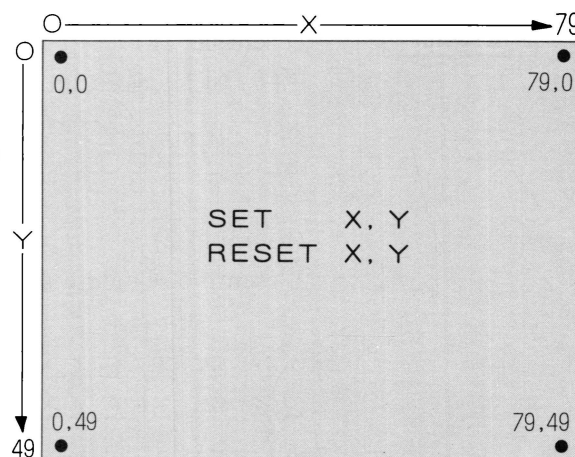
Poiché l'istruzione SET attiva simultaneamente quattro punti, se si cambia il colore di uno dei punti di tale gruppo di quattro, cambia il colore anche degli altri punti.

Le istruzioni SET e RESET possono essere usate per produrre un'ampia varietà di effetti interessanti; alcuni esempi vengono mostrati qui di seguito.

1. Attivazione di un punto sullo schermo.

```
10 PRINT "■"  
20 X=79:Y=49  
30 SET X,Y,2  
40 RESET X,Y  
50 GOTO 30
```

← Attiva i punti.
← Disattiva i punti.



2. Colorazione di bianco dell'intero schermo.

```
10 PRINT "■"  
20 FOR X=0 TO 79  
30 FOR Y=0 TO 49  
40 SET X,Y,7  
50 NEXT Y,X  
60 GOTO 10
```

3. Tracciare un rettangolo attorno ai margini dello schermo.

```
10 PRINT "■"  
20 FOR X=0 TO 79  
30 SET X,0  
40 SET X,49  
50 NEXT X  
60 FOR Y=0 TO 49  
70 SET 0,Y  
80 SET 79,Y  
90 NEXT Y  
100 GOTO 100
```

-
4. Un programma che simula le increspature prodotte da una pietra gettata in uno stagno.

```
10 X=40:Y=25
20 DEF FNY (Z) =SQR (R*R-Z*Z)
30 PRINT "■":SET X, Y
40 R=R+5
50 FOR Z=0 TO R
60 T=FNY (Z)
70 SET X+Z, Y+T
80 SET X+Z, Y-T
90 SET X-Z, Y+T
100 SET X-Z, Y-T
110 NEXT Z
120 IF R<>25 THEN 40
130 GOTO 130
```

5. Un programma che simula una palla che rimbalza sulle quattro pareti.

```
10 PRINT "■"
20 FOR X=0 TO 79
30 SET X, 0:SET X, 49
40 NEXT X
50 FOR Y=0 TO 49
60 SET 0, Y:SET 79, Y
70 NEXT Y
80 X=79*RND (1):Y=49*RND (1)
90 A=1:B=1
100 SET X, Y
110 IF X<2 GOSUB 200
120 IF X>78 GOSUB 200
130 IF Y<2 GOSUB 250
140 IF Y>48 GOSUB 250
150 RESET X, Y
160 X=X+A:Y=Y+B:GOTO 100
200 A=-A:MUSIC "A0": RETURN
250 B=-B:MUSIC "A0": RETURN
```

Nota

Per quanto concerne il comando JOY, vedere il manuale delle istruzioni per il circuito di barra.

2.3.9 Istruzioni di controllo della musica

Questa sezione tratta le istruzioni MUSIC e TEMPO che vengono usate per controllare l'esecuzione della musica da parte del calcolatore. Come è implicito nel suo stesso nome, l'istruzione TEMPO specifica la velocità a cui viene eseguita la musica. Le note (incluse le minime e le ottave superiori ed inferiori) nonché la durata delle note prodotte sono controllate dall'istruzione MUSIC.

Tempo:	Specificato con TEMPO come variabile o costante numerica con un valore da 1 (lento) a 7 (veloce).
Melodia:	Specificata con MUSIC come una variabile di stringa costituita da una raccolta di note.
Specificazione della nota:	<div>ottava</div> <div># (diesis)</div> <div>nome nota</div> <div>durata</div>

2.3.9.1 MUSIC (formato abbreviato: MU.)

Formato

MUSIC X\$

X\$... Stringa di dati

Esegue automaticamente la musica.

Discussione

Questa istruzione emette all'altoparlante la melodia o gli effetti sonori specificati dalla stringa di caratteri o dalla variabile di stringa del suo argomento. La velocità con cui viene eseguita questa melodia è quella specificata con l'istruzione TEMPO (vedere pagina 66).

Il formato della specifica di ciascuna nota è il seguente:

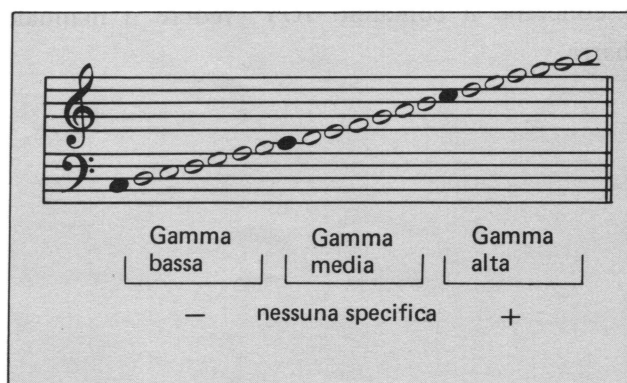
< specifica ottava > < # (diesis) > nome nota < durata >

I segni più o meno vengono usati per specificare l'ottava. Se non viene specificato niente, si assume la gamma mediana. Le tre gamme di suoni che il calcolatore è in grado di produrre sono illustrate nella figura seguente. Per esempio, le note C ("do") indicate dai puntini neri sottostanti sono differenziate l'una dall'altra dalla specifica di ottava.

C basso -C

C medio C

C alto +C

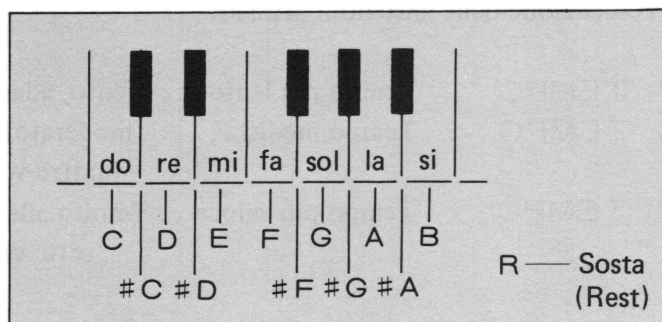


Specifica delle note

I simboli usati per specificare le note all'interno di ogni gamma sono i seguenti:

CDEFGAB # R

La relazione tra la scala musicale (do, re, mi, fa, sol, la, si, do) e questi simboli viene illustrata qui di seguito. Il simbolo di diesis (#) viene usato per specificare le minime. Gli intervalli silenti vengono specificati con "R".



Specifica della durata

La specifica della durata determina la lunghezza della nota specificata. Le durate da 1/32 all'unità vengono specificate come numeri da 0 a 9. (Quando si specifica R, questo determina la lunghezza dell'intervallo silente).

Pausa di 1/32	Pausa di 1/16	Pausa di 1/16 puntata	Pausa di 1/8	Pausa di 1/8 puntata	Pausa di 1/4	Pausa di 1/4 puntata	Pausa di 1/2	Pausa di 1/2 puntata	Pausa intera
Nota di 1/32	Nota di 1/16	Nota di 1/16 puntata	Nota di 1/8	Nota di 1/8 puntata	Nota di 1/4	Nota di 1/4 puntata	Nota di 1/2	Nota di 1/2 puntata	Nota intera
0	1	2	3	4	5	6	7	8	9

Quando delle note successive hanno la stessa durata, è possibile omettere la specifica della seconda nota e da quelle successive. Se non viene specificata alcuna durata, per la prima nota, si assumono le note di 1/4.

Volume del suono

Il volume del suono prodotto non può essere controllato dal programma, ma può essere regolato con il controllo del volume esterno del calcolatore.

Esempio

Assegnamo una stringa ad SR\$ per suonare il tema dall'inizio della serenata in re maggiore di Beethoven (Opus 25).

```
SR$ = " + A3 + #F1 + A + B3A + D + #F1A + D3A + D  
      + #F1A + D3 + #F1A + D + E + #F + G + A3R "
```



2. 3. 9. 2 TEMPO (formato abbreviato: TEM.)

Formato

TEMPO x

x Espressione numerica (1 ~ 7)

Funzione

Questa istruzione determina il tempo al quale viene suonata la musica mediante l'istruzione MUSIC. Se questa istruzione non viene eseguita, si assume il TEMPO 4 per l'esecuzione delle istruzioni MUSIC.

30 TEMPO 1 Tempo più lento ... (lento, adagio)

30 TEMPO 4 Tempo medio (moderato);

quattro volte più veloce del TEMPO 1.

30 TEMPO 7 Tempo più veloce .. (molto allegro, presto);

sette volte più veloce del TEMPO 1.

Esempio

10 REM MAZURCA DI CHOPIN

20 MM\$ = "A3" : M1\$ = "A5+#C3+D+E+#F+G+#FO+G+#
F4+E3+D+#CB"

30 M2\$ = "A3+D2RO+D1+E2+D+#C3B+#C7+#C3"

40 M3\$ = "A3+#C2RO+#C1+D2+#CB3A+D7+D3"

50 TEMPO 3

60 MUSIC MM\$, M1\$, M2\$, M1\$, M3\$, M1\$, M2\$, M1
\$, M3\$

70 END



2.3.10 Comandi di immissione/emissione dei file di dati

Sebbene si possano usare i comandi SAVE e LOAD, per scrivere o leggere il testo del programma, vengono usati altri comandi per registrare o leggere i diversi tipi di dati trattati dai programmi. Questi comandi vengono descritti qui di seguito.

	Formato	Funzione
WOPEN (abbreviato W.)	WOPEN < file name >	Aprire un file di dati sulla cassetta prima di scrivervi i dati. Questo comando assegna anche un nome al file di dati.
PRINT/T (abbreviato ?/T)	PRINT/T d_1 < , d_2 , d_3 , ... d_n > d_n ... dati numerici o dati di stringa	Scrivere i dati sulla cassetta nello stesso formato in cui vengono visualizzati mediante l'istruzione PRINT.
ROPEN (abbreviato RO.)	ROPEN < file name >	Cerca il file di dati sulla cassetta con il nome specificato ed apre tale file per preparare la lettura dei dati dallo stesso.
INPUT/T (abbreviato I./T)	INPUT/T v_1 < , v_2 , v_3 , ... v_n > v_n ... dati numerici o dati di stringa	Usato per immettere i dati da un file nella cassetta e passarli al programma (in un modo simile a quello con cui si usa l'istruzione INPUT per immettere i dati dalla tastiera).
CLOSE (abbreviato CLO.)	CLOSE	Istruzione che chiude i file di dati nella cassetta dopo che è stata completata la scrittura o la lettura.

Diversamente dai comandi LOAD e SAVE, l'esecuzione delle istruzioni WOPEN e ROPEN non visualizza alcun messaggio. Se desiderate la visualizzazione di un messaggio, usate l'istruzione PRINT per definirne uno nel programma.

Nota: Quando si usa un normale registratore a cassette, non può essere possibile registrare i file di dati anche se non si incontrano problemi nella memorizzazione o nella lettura dei programmi con i comandi SAVE e LOAD.

(Esempio 1)

Il programma seguente scrive i numeri da 1 a 99 su cassetta.

```
10 WOPEN "DATA "  
20 FOR X=1 TO 99  
30 PRINT/T X  
40 NEXT X  
50 CLOSE  
60 END
```

(Esempio 2)

Il programma seguente legge i dati dal file di dati preparato nell'esempio 1 precedente. Prima di eseguire questo programma, accertatevi di riavvolgere il nastro della cassetta.

```
10 OPEN "DATA "  
20 FOR X=1 TO 99  
30 INPUT/T A  
40 PRINT A  
50 NEXT X  
60 CLOSE  
70 END
```

(Esempio 3)

Il programma seguente crea un file di dati costituito da dati di stringa.

```
10 DIM N$(5)  
20 N$(1) = "BACH "  
30 N$(2) = "MOZART "  
40 N$(3) = "BEETHOVEN "  
50 N$(4) = "CHOPIN "  
60 N$(5) = "BRAHMS "  
70 WOPEN "GREAT MUSICIAN "  
80 FOR J=1 TO 5  
90 PRINT/T N$(J)  
100 NEXT J  
110 CLOSE  
120 END
```

(Esempio 4)

Il programma seguente legge i dati di stringa dal file creato nell'esempio 3. Prima di eseguire questo programma, accertatevi di riavvolgere il nastro.

```
200 DIM M$(5)  
210 OPEN "GREAT MUSICIAN "  
220 FOR K=1 TO 5  
230 INPUT/T M$(K)  
240 PRINT M$(K)  
250 NEXT K  
260 CLOSE  
270 END
```

E' possibile anche creare dei file di dati che includono sia dei dati numerici che di stringa. Tuttavia, poiché si verificherà un errore se il tipo di dati letto non corrisponde al tipo di variabile specificata nell'istruzione INPUT/T, è generalmente meglio limitare i file ad un tipo di dati o all'altro.

Nota: E' possibile omettere il nome del file quando si apre un file sequenziale con l'istruzione WOPEN.

Tuttavia, è probabile che questo porti a degli errori se molti file sono inclusi nello stesso nastro; pertanto, si consiglia di prendere l'abitudine di assegnare dei nomi ai file di dati sequenziali.

Il programma seguente registra i voti di uno studente in inglese, francese, scienze e matematica in un file di dati sequenziale su cassetta.

```

10 INPUT "ENTER NO. OF STUDENTS " ; N
20 DIM N$ (N) , K (N) , E (N)
30 DIM R (N) , S (N)
40 A$ = "GRADE IS "
50 FOR X=1 TO N
60 PRINT:PRINT "STUDENT NO. " ; X
70 INPUT "ENTER STUDENT NAME: " ; N$ (X)
80 PRINT "ENG " ; A$ ; : INPUT K (X)
90 PRINT "FREN" ; A$ ; : INPUT E (X)
100 PRINT "SCI " ; A$ ; : INPUT R (X)
110 PRINT "MATH" ; A$ ; : INPUT S (X)
120 NEXT X
130 WOPEN "GRADES " ← Apre il file di dati "GRADES" per l'emissione su nastro.
140 PRINT/T N ← Scrive il numero degli studenti della classe nel file.
150 FOR X=1 TO N
160 PRINT/T N$ (X) , K (X) , E (X) , R (X) , S (X) ← Scrive i voti nel file.
170 NEXT X
180 CLOSE ← Chiude il file su cassetta.
190 END

```

Il programma seguente legge i dati dei voti scritti nel file su cassetta dal programma illustrato in precedenza, quindi calcola e visualizza la media dei voti di ogni studente e la media di classe per ciascuna delle varie materie.

```

10 ROPEN "GRADES " ← Apre il file su cassetta "GRADES" per l'immissione.
20 INPUT/T N ← Legge il numero degli studenti della classe.
30 DIM N$ (N) , K (N) , E (N)
40 DIM R (N) , S (N)
50 FOR X=1 TO N
60 INPUT/T N$ (X) , K (X) ← Legge i nomi degli studenti ed i voti per l'inglese.
70 INPUT/T E (X) , R (X) , S (X) ← Legge i voti per il francese, scienze e matematica.
80 NEXT X
90 CLOSE ← Chiude il file.
100 PRINT TAB (10) ; "ENG " ;
110 PRINT TAB (15) ; "FREN" ;
120 PRINT TAB (20) ; "SCI " ;
130 PRINT TAB (25) ; "MATH"
140 FOR X=1 TO N
150 PRINT N$ (X) ; TAB (10) ; K (X) ;
160 PRINT TAB (15) ; E (X) ;
170 PRINT TAB (20) ; R (X) ;
180 PRINT TAB (25) ; S (X) ;
190 PRINT TAB (30) ; (K (X) + E (X) + R (X) + S (X) ) / 4
200 K (0) = K (0) + K (X) : E (0) = E (0) + E (X)
210 R (0) = R (0) + R (X) : S (0) = S (0) + S (X)
220 NEXT X
230 PRINT TAB (10) ; K (0) / N ; TAB (15) ; E (0) / N ;
240 PRINT TAB (20) ; R (0) / N ; TAB (25) ; S (0) / N
250 END

```

2.4 Funzioni incorporate

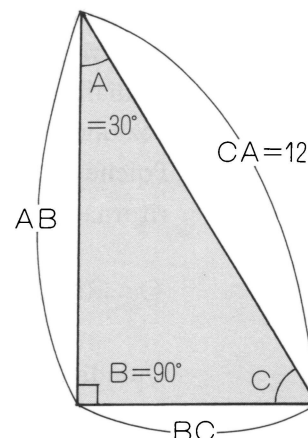
Funzione	Simbolo BASIC	Esempio	Descrizione
Valore assoluto	ABS (X)	A = ABS (X)	Assegna il valore assoluto della variabile X alla variabile A. Esempi: A = ABS (2. 9) → A = 2.9 A = ABS (-5. 5) → A = 5.5
Segno	SGN (X)	A = SGN (X)	Assegna il segno numerico della variabile X alla variabile A. Se il valore di X è negativo, ad A viene assegnato -1; se X è 0, ad A viene assegnato 0; e se X è positivo, ad A viene assegnato 1. $A = \begin{cases} 1 & (X > 0) \\ 0 & (X = 0) \\ -1 & (X < 0) \end{cases}$ Esempio: Alla variabile A viene assegnato 1 quando si esegue A = SGN (0.4).
Conversione numero intero	INT (X)	A = INT (X)	Assegna ad A il più alto valore in numero intero che è inferiore o uguale al valore della variabile X. Esempi: A = INT (3. 87) → A = 3 A = INT (0. 6) → A = 0 A = INT (-3. 87) → A = -4
Funzioni trigonometriche	SIN (X)	A = SIN (X) A=SIN (30*PAI(1)/180)	Assegna il seno di X (dove X è in radianti) alla variabile A. Se il valore di X è in gradi, deve essere convertito in radianti prima di usare questa funzione per ottenere il seno. Poiché un grado è uguale a $\pi/180$ radianti, il valore in radianti viene ottenuto moltiplicando il numero dei gradi per PAI(1)/180. Per esempio: 30e = 30 * PAI(1)/180 radianti. Altrettanto vale per le funzioni COS, TAN e ATN.
	COS (X)	A = COS (X) A=COS (200*PAI(1)/180)	Assegna il coseno di X (dove X è in radianti) alla variabile A.
	TAN (X)	A = TAN (X) A=TAN(Y*PAI(1)/180)	Assegna la tangente di X (dove X è in radianti) alla variabile A.
	ATN (X)	A = ATN (X) A=180/PAI(1)*ATN(X)	Assegna l'arcotangente in radianti di X ($\tan^{-1} X$) alla variabile A. Il valore ritornato sarà nella gamma compresa tra $-\pi/2$ e $\pi/2$.
Radice quadrata	SQR (X)	A = SQR (X)	Calcola la radice quadrata di X ed assegna il risultato alla variabile A. X deve essere un numero positivo o 0.
Potenza	EXP (X)	A = EXP (X)	Calcola il valore di e^x ed assegna il risultato alla variabile A.
Logaritmo decimale	LOG (X)	A = LOG (X)	Calcola il logaritmo decimale di X ($\log_{10} X$) ed assegna il risultato alla variabile A.
Logaritmo naturale	LN (X)	A = LN (X)	Calcola il logaritmo naturale di X ($\log_e X$) ed assegna il risultato alla variabile A.
Rapporto della circonferenza rispetto al diametro.	PAI (X)	A = PAI (X)	Assegna il valore alla variabile A che è X volte il valore di PI.
Radianti	RAD (X)	A = RAD (X)	Converte il valore di X (dove X è in gradi) in radianti ed assegna il risultato alla variabile A.

Esempi di uso delle funzioni incorporate

(Esempio 1)

Risolviamo i diversi elementi di un triangolo con un programma BASIC.

L'angolo A del triangolo mostrato nella figura a destra è di 30° , l'angolo B è un angolo retto, ed il lato CA è lungo 12. Il programma seguente trova tutti gli angoli del triangolo, la lunghezza dei suoi lati, e l'area totale.



```
10 A=30 : B=90 : CA=12
20 AB=CA*COS (A*PI/180)
30 BC=CA*SIN (A*PI/180)
40 S=AB*BC/2
50 C=180-A-B
60 PRINT "AB=" ; AB, "BC=" ; BC, "CA=" ; CA
70 PRINT "AREAS=" ; S
80 PRINT "A=" ; A, "B=" ; B, "C=" ; C
90 END
```

(Esempio 2)

Cambiamo ora la riga 50 del programma per usare ATN, la funzione per trovare l'arcotangente di un numero, cioè per trovare l'angolo C conoscendo i lati AB e BC.

```
10 A=30 : B=90 : CA=12
20 AB=CA*COS (A*PI/180)
30 BC=CA*SIN (A*PI/180)
40 S=AB*BC/2
50 C=ATN (AB/BC) *180/PI
60 PRINT "AB=" ; AB, "BC=" ; BC, "CA=" ; CA
70 PRINT "AREAS=" ; S
80 PRINT "A=" ; A, "B=" ; B, "C=" ; C
90 END
```

Funzione RND

Formato

RND (X)

X . . . Fspresione numerica

Funzione

La funzione RND rimanda un numero pseudocasuale nella gamma tra 0,00000001 e 0,99999999.

Quando X è superiore a 0, il numero casuale ritornato è quello che segue quello precedentemente generato dal programma interprete BASIC in una data serie di numeri pseudocasuali.

Quando $X \leq 0$, il generatore di numeri pseudocasuali del programma interpreti BASIC viene nuovamente inizializzato per avviare una nuova serie, ed il numero pseudocasuale ritornato è il primo di tale serie. Tale reinizializzazione della serie di numeri pseudocasuali può essere utilizzata per consentire le simulazioni basate sui numeri casuali da riprodurre.

La funzione RND viene spesso usata nei programmi dei giochi per produrre dei numeri imprevedibili, come nei giochi di fortuna. Usiamo la funzione RND per studiare la percentuale di uscita di ciascuna delle sei facce di un dado simulando l'azione di gettarlo per un dato numero di volte.

Poiché le facce di ogni dado sono numerate da 1 a 6, dobbiamo moltiplicare il valore ritornato dalla funzione RND per 6.

$$0 < \text{RND}(1) < 1 \xrightarrow{\times 6} 0 < 6 * \text{RND}(1) < 6$$

Quindi dobbiamo usare la funzione INT per convertire il valore ottenuto in un numero intero.

$$\text{INT}(6 * \text{RND}(1)) \rightarrow 0, 1, 2, 3, 4, 5$$

Il risultato è un numero intero compreso tra 0 e 5; ora viene aggiunto 1 per ottenere i numeri che corrispondono al numero di puntini su ciascuna delle sei facce di un dado.

$$\text{INT}(6 * \text{RND}(1)) + 1 \rightarrow 1, 2, 3, 4, 5, 6$$

Questa sequenza viene eseguita un numero di volte specificato per ciascun lancio del dado. Incorporiamo ora la sequenza in un programma e controlliamo i risultati.

Esempio

```

10 PRINT "QUANTE VOLTE VU
   OI LANCIARE IL DADO? ";
20 INPUT N
30 FOR J=1 TO N
40 R=INT(6*RND(1))+1
50 IF R=1 THEN N1=N1+1
60 IF R=2 THEN N2=N2+1
70 IF R=3 THEN N3=N3+1
80 IF R=4 THEN N4=N4+1
90 IF R=5 THEN N5=N5+1
100 IF R=6 THEN N6=N6+1
110 NEXT J
120 P1=N1/N:P2=N2/N:P3=N3/N
130 P4=N4/N:P5=N5/N:P6=N6/N
140 PRINT P1, P2, P3, P4, P5, P6
150 END

```

La funzione RND genera dei numeri compresi nella gamma da 0,00000001 a 0,99999999.

Ecco come ottenere dei numeri interi casuali nella gamma compresa tra 6 e 8!

$$\text{INT}(3 * \text{RND}(1)) + 6 = 6, 7, 8$$

Come mai? Se il dado viene lanciato un numero di volte sufficiente, la percentuale di uscita di ogni numero dovrebbe essere pressoché uguale. Da un punto di vista matematico, ogni numero dovrebbe uscire mediamente una volta ogni sei lanci, cioè circa il 16,7% delle volte. Ci si avvicina sempre di più a questo ideale matematico aumentando il numero dei lanci.

Esempio

Usiamo ora la funzione RND in un programma che mette a prova la vostra capacità di trovare l'area di un triangolo di dimensione casuale. Qui la funzione RND viene usata per determinare la lunghezza di ciascuno dei tre lati del triangolo, quindi calcolate da voi l'area del triangolo e presentate la vostra risposta al calcolatore per il controllo.

```
10 DIM A (3) , L$ (4)
20 FOR J=1 TO 4
30 READ L$ (J) : NEXT J
40 FOR J=1 TO 3
50 A (J) = INT (20*RND (1) ) + 1
60 NEXT J
70 IF A (1) >= A (2) + A (3) GOTO 40
80 IF A (2) >= A (1) + A (3) GOTO 40
90 IF A (3) >= A (1) + A (2) GOTO 40
100 W = (A (1) + A (2) + A (3) ) / 2
110 T = W : FOR J=1 TO 3
120 T = T * (W - A (J) ) : NEXT J
130 SS = SQR (T) : S = INT (SS)
140 IF SS - S > 0.5 THEN S = S + 1
150 PRINT "C1111"
160 PRINT "TROVA L'AREA DEL
TRIANGOLO SEGUENTE"
170 PRINT "ARROTONDA LA RISPOSTA
AL NUMERO INTERO PIU' VICINO"
180 PRINT
190 PRINT TAB (8) ; "A "
200 PRINT TAB (8) ; "□□" ; TAB (15) ; L$ (1)
; A (1)
210 PRINT TAB (7) ; "□ □" ; TAB (15) ; L$ (2)
; A (2)
220 PRINT TAB (6) ; "□ □" ; TAB (15) ; L$ (3)
; A (3)
230 PRINT TAB (5) ; "□ □"
240 PRINT TAB (3) ; "B□ □C"
250 PRINT TAB (4) ; "□□□□□□□□"
260 PRINT "1111"
270 PRINT TAB (3) ; L$ (4) ;
280 INPUT Y
290 IF Y = S THEN PRINT "OK!!" : GOTO 40
300 IF Y < S THEN PRINT "TROPPO PICCOL
A!" : GOTO 320
310 PRINT "TROPPO GRANDE!"
320 PRINT "↑↑" ;
330 PRINT "↑" ;
340 GOTO 270
350 DATA LUNGHEZZA LATO AB=, LUNGHEZZA
LATO BC=
360 DATA LUNGHEZZA LATO CA=, L'AREA S
DEL TRIANGOLO ABC E'
```

Nota

Notate che specificando un valore per X inferiore o uguale a 0, si otterrà sempre lo stesso numero per un dato valore di X . Il motivo è che specificando 0 o un numero negativo, si riinizializza il generatore di numeri pseudocasuali all'inizio della serie di numeri casuali.

2.5 Funzioni di stringa

2.5.1 LEN

Formato

LEN (X\$)

X\$. . . Espressione di stringa

Funzione

Questa funzione ritorna il numero di caratteri incluso nell'espressione di stringa rappresentata da X\$. Questo valore include gli spazi che non sono visualizzati sullo schermo e gli eventuali caratteri di controllo della stringa, nonché le lettere, i numeri ed i simboli.

Esempio

(Esempio 1)

```
10 A$="ABCDEFGG"  
20 PRINT LEN (A$)
```

```
RUN  
7
```

(Esempio 2) Il seguente programma usa la funzione LEN per tracciare dei quadrati sullo schermo.

```
10 ? "■" : ? "IMMETTERE 3 O PIU' ASTERISCHI "  
20 INPUT A$  
30 FOR I=1 TO LEN (A$) -2  
40 PRINT TAB (2) ; "*" ; SPC (LEN (A$) -2) ; "*" ;  
50 NEXT I  
60 PRINT TAB (2) ; A$ : GOTO 20
```

(Esempio 3) La funzione LEN può essere usata anche per produrre una "parata" di caratteri come illustrato qui di seguito.

```
10 S$="SHARP BASIC"  
20 FOR I=1 TO LEN (S$)  
30 ? RIGHT$ (S$, I)  
40 NEXT I  
50 END  
RUN  
C  
I C  
S I C  
:  
:  
:  
SHARP BASIC
```

(Esempio 4)

```
PRINT LEN (STR$ (PAI (1)))  
9
```

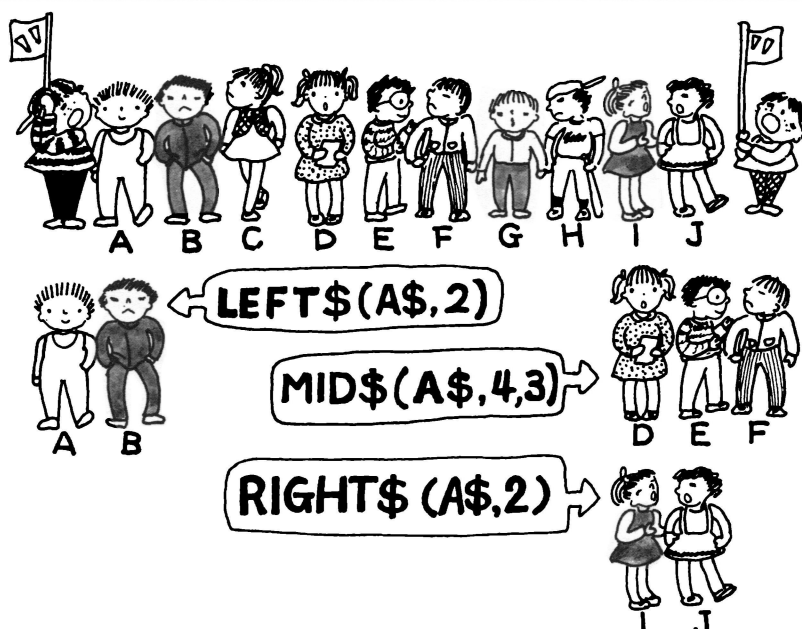
PAI (1), la funzione che ritorna il valore del rapporto della circonferenza di un cerchio rispetto al suo diametro, contiene la costante di 8 cifre 3,1415927 (il valore approssimativo di PI). Quando la lunghezza della stringa di caratteri prodotta convertendo questa costante con la funzione STR\$ viene valutata con la funzione LEN, viene ritornata una lunghezza di stringa totale di 9.

La lunghezza ritornata dalla funzione LEN non corrisponde al numero di caratteri della stringa che è l'argomento della funzione se la stringa include delle lettere minuscole.

2.5.2 LEFT\$, MID\$, e RIGHT\$

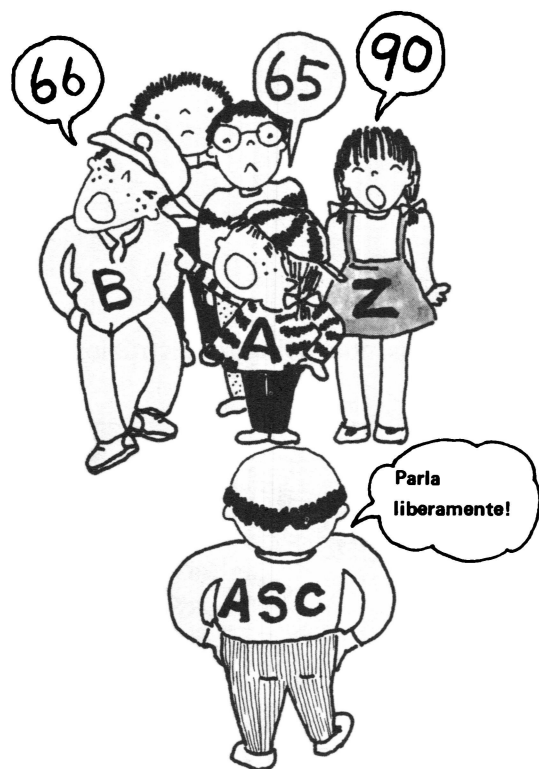
Le funzioni LEFT\$, MID\$ e RIGHT\$ vengono usate per estrarre le stringhe di caratteri dall'estremità sinistra, dall'estremità destra o dalla metà di un'espressione di caratteri.

Formato X\$: Espressione di stringa m e n: Espressioni numeriche	Funzione	Esempio (quando A\$ = "ABCDEFGG")	Commenti
LEFT\$ (X\$, n)	Ritorna la stringa di caratteri costituita dagli n caratteri che costituiscono l'estremità sinistra dell'espressione di stringa X\$.	B\$ = LEFT\$ (A\$, 2) B\$ ← AB CDEFG Sostituisce 2 caratteri dall'estremità di sinistra della variabile di stringa A\$ nella variabile di stringa B\$. Pertanto, B\$ = "AB".	$0 \leq n \leq 255$
MID\$ (X\$, m, n)	Ritorna la stringa di caratteri costituita dagli n caratteri che costituiscono gli n caratteri partendo dall'm-esimo carattere nelle espressioni di stringhe X\$.	B\$ = MID\$ (A\$, 3, 3) B\$ ← AB CDE FG Sostituisce i tre caratteri partendo dal terzo carattere nella variabile di stringa A\$ nella variabile di stringa B\$.	$1 \leq m \leq 255$ $0 \leq n \leq 255$
RIGHT\$ (X\$, n)	Ritorna la stringa di caratteri costituita dagli n caratteri che costituiscono l'estremità destra dell'espressione di stringa X\$.	B\$ = RIGHT\$ (A\$, 2) B\$ ← ABCDE FG Sostituisce due caratteri dall'estremità destra della variabile di stringa A\$ nella variabile di stringa B\$. Pertanto, B\$ = "FG".	$0 \leq n \leq 255$



2.5.3 ASC e CHR\$

Formato	Funzione	Esempio
ASC (x\$) x\$: Espressione di stringa	Ritorna il codice ASCII per il primo carattere nell'espressione di stringa X\$.	<p>X = ASC (" A ") Sostituisce 65 (il codice ASCII per la lettera A) nella variabile X.</p> <p>Y=ASC (" S HARP ")</p> <p>Sostituisce 83 (il codice ASCII per S, la prima lettera della stringa "SHARP") nella variabile X.</p>
CHR\$ (x) x: Espressione numerica	Ritorna la lettera il cui codice ASCII corrisponde al valore dell'espressione numerica X. (Non viene ritornato alcun carattere se il valore specificato per X è inferiore a 33; pertanto, si deve usare PRINT " _ " o PRINT SPC (1) per ottenere degli spazi, invece di CHR\$ (32)).	<p>A\$ = CHR\$ (65) Assegna A, la lettera corrispondente al codice ASCII 65, alla variabile di stringa A\$. Questa funzione può essere usata per visualizzare i caratteri che non possono essere immessi dalla tastiera come segue.</p> <p>PRINT CHR\$ (107) J Questo visualizza i caratteri grafici ☒.</p>



Nota: Il codice ASCII è un sistema di codici standard che viene frequentemente usato con i calcolatori. Questo codice utilizza dei numeri di 8 bit per rappresentare le lettere dell'alfabeto, i numeri ed i simboli quali il segno di dollari ed il punto interrogativo. La tavola completa dei codici viene data a pagina 156.

2.5.4 VAL e STR\$

Formato	Funzione	Esempio
STR\$ (x) x: Espressione numerica	Ritorna una stringa di caratteri ASCII che rappresenta il valore dell'espressione numerica X.	<p>A\$ = STR\$ (-12) Sostituisce la stringa di caratteri "-12" nella variabile di stringa A\$.</p> <p>B\$ = STR\$ (70 * 33) Sostituisce la stringa di caratteri " 2310" nella variabile di stringa B\$.</p> <p>C\$ = STR\$ (1200000 * 5000) Sostituisce la stringa di caratteri " 6E + 09" nella variabile di stringa C\$.</p> <p>Nota: I valori numerici positivi vengono visualizzati con uno spazio iniziale per indicare che è stato ommesso il segno più (+). Tuttavia, questo spazio non è incluso nella stringa di caratteri ritornata dalla funzione STR\$.</p>
VAL (x\$) x\$: Espressione di stringa	Converte una rappresentazione di caratteri ASCII di un valore numerico in un valore numerico. Questo è complementare alla funzione STR\$.	<p>A = VAL (" 123 ") Converte la stringa di caratteri " 123 " nel numero 123 e lo assegna alla variabile numerica A.</p>

Il programma campione seguente illustra l'uso di alcune delle funzioni trattate in precedenza per visualizzare dei valori numerici in formato tabulare (con i punti decimali allineati).

```

1. 23456
12. 3456
1Ø
1
1234

```

Se i valori letti dall'istruzione DATA venissero visualizzati usando soltanto l'istruzione PRINT, il risultato sarebbe il seguente.

```

1Ø FOR X=1 TO 5
2Ø READ A
3Ø L=5-LEN (STR$ (INT(A)))
4Ø PRINT TAB (L) ;A
5Ø NEXT:END
6Ø DATA 1. 23456, 12. 3456
7Ø DATA 123. 456, 1234. 56
8Ø DATA 12345. 6

```

```

1. 23456
12. 3456
123. 456
1234. 56
12345. 6

```



2.6 Istruzioni per la visualizzazione a colori

Una delle più importanti caratteristiche dell'MZ-700 è che consente la visualizzazione dei caratteri e dei grafici usando un massimo di 8 colori.

2.6.1 COLOR (formato abbreviato: COL.)

Formato

COLOR x, y, c <, b >

x Coordinata X (0 ~ 39)

y Coordinata Y (0 ~ 24)

c Istruzione per il colore dei caratteri (0 ~ 7).

b Istruzione per il colore dello sfondo (0 ~ 7).

Funzione

Questa istruzione viene usata per determinare i colori di primo piano e di sfondo per i caratteri in una posizione specifica dello schermo. E' possibile specificare un massimo di 8 colori diversi per il primo piano (c) o lo sfondo (b) come mostrato nella tabella seguente.

N° colore	Colore
0	Nero
1	Blu
2	Rosso
3	Porpora
4	Verde
5	Azzurro
6	Giallo
7	bianco

Esempio

- (1) Cambiamento del colore di sfondo dell'intero schermo

COLOR . . . 2 (Cambia il colore di sfondo usato per la visualizzazione dei caratteri in rosso).

- (2) Cambiamento del colore di primo piano dell'intero schermo (il colore usato per la visualizzazione di tutti i caratteri)

COLOR . . 3 (Cambia il colore usato per la visualizzazione di tutti i caratteri in porpora).

- (3) Cambiamento dei colori sia dello sfondo che del primo piano per l'intero schermo

COLOR . . 1, Ø (Cambia il colore usato per la visualizzazione di tutti i caratteri in blu e cambia lo sfondo usato per la visualizzazione dei caratteri in nero).

- (4) Cambiamento del colore dello sfondo in una posizione specifica dello schermo

COLOR 2, 2, . 4 (Cambia il colore dello sfondo in corrispondenza delle coordinate 2, 2 in verde).

- (5) Cambiamento del colore di primo piano in una posizione specifica dello schermo
 COLOR 3, 2, 7 (Cambia il colore di primo piano in una posizione specifica dello schermo)
- (6) Cambiamento del colore sia del primo piano che dello sfondo in una posizione specifica dello schermo
 COLOR 4, 2, 4, 2 (Cambia il colore del primo piano in corrispondenza delle coordinate 4, 2 in verde e cambia il colore dello sfondo in tale posizione in rosso).

2.6.2 Aggiunta delle specifiche di colore all'istruzione PRINT

Formato

$\left\{ \begin{array}{c} \text{PRINT} \\ ? \end{array} \right\} [f, b] \left\{ \begin{array}{c} \text{variabile} \\ \text{costante} \\ \text{espressione} \end{array} \right\} < \left\{ \begin{array}{c} ; \\ , \end{array} \right\} \left\{ \begin{array}{c} \text{variabile} \\ \text{costante} \\ \text{espressione} \end{array} \right\} \left\{ \begin{array}{c} ; \\ , \end{array} \right\} \dots >$

o

$\left\{ \begin{array}{c} \text{PRINT} \end{array} \right\} [f, b] \text{ USING "stringa formato" ; } \left\{ \begin{array}{c} \text{variabile} \\ \text{costante} \\ \text{espressione} \end{array} \right\} < \left\{ \begin{array}{c} ; \\ , \end{array} \right\} \left\{ \begin{array}{c} \text{variabile} \\ \text{costante} \\ \text{espressione} \end{array} \right\} \dots >$

f ... Specifica del primo piano (colore dei caratteri) (un numero tra 0 e 7)

b ... Specifica del colore dello sfondo dei caratteri (un numero da 0 e 7)

Funzione

Aggiungendo le specifiche di colore alle istruzioni PRINT e PRINT USING descritte a pagine 35 e 36, è possibile visualizzare i caratteri in diversi colori. Nel formato precedente, f indica il colore del primo piano dei caratteri mentre b indica il colore dello sfondo dei caratteri. Se si specifica soltanto il colore del primo piano, viene usato il colore di sfondo attuale per la visualizzazione dei caratteri; ciò viene fatto specificando il colore del primo piano, seguito da una virgola. Se si specifica soltanto il colore dello sfondo, viene usato il colore attuale del primo piano per la visualizzazione dei caratteri; in questo caso, una virgola deve precedere la specifica del colore dello sfondo.

Esempio

(Esempio 1)

PRINT (6, 5) "ABCDE " Visualizza le lettere "ABCDE" in giallo su sfondo azzurro.

PRINT (, 4) "FGHIJ " Visualizza le lettere "FGHIJ" in giallo su sfondo verde.

PRINT (7,) "VWXYZ " Visualizza le lettere "VWXYZ" in verde su sfondo bianco.

(Esempio 2) Aggiungiamo il colore al programma delle corse automobilistiche illustrato a pagina 44.

```
10 PRINT (, 1) "■"
20 Q=INT (5*RND (1)) +2 : X=33*RND (1)
30 FOR A=1 TO 5
40 READ M$
50 PRINT TAB (0) ; "◆" ; TAB (X) ;
60 PRINT (Q, 1) M$ ;
70 PRINT (7, 1) TAB (37) ; "◆"
80 NEXT A
90 Y=10*RND (1)
100 FOR A=1 TO Y
110 PRINT TAB (0) ; "◆" ;
120 PRINT TAB (37) ; "◆" : NEXT
130 RESTORE : GOTO 20
140 DATA " ■●■ ", " ●■■■■■ "
150 DATA " ■■■ ", " ●■■■■■ "
160 DATA " ■■■ "

```

Con le normali istruzioni PRINT (quelle senza le specifiche di colore), i colori di primo piano e di sfondo usati per la visualizzazione dei caratteri sono quelli che sono stati specificati con l'ultima istruzione COLOR.

2.7 Comandi per la stampante-plotter a colori

I comandi per la stampante-plotter a colori descritti qui di seguito possono essere usati con l'MZ-731 o, quando è collegato alla stampante-plotter a colori MZ-1P01, con MZ-711 o l'MZ-721. La stampante-plotter a colori può essere usata in due modi: il modo testo (per la stampa di liste di programmi, risultati di calcoli o altri dati costituiti da caratteri), o il modo grafico (per disegnare figure e grafici). Inoltre, è possibile usare uno qualsiasi di quattro colori (nero, blu, verde o rosso) per la stampa dei caratteri e dei grafici. Questa capacità è particolarmente utile quando si usa la stampante nel modo grafico.

2.7.1 Informazioni generali sulla stampante-plotter a colori

- (1) La stampante-plotter a colori funziona in due modi: il **modo testo** (per la stampa dei risultati di calcoli, liste di programmi, ed altri dati costituiti da caratteri) ed il **modo grafico** (usato per disegnare figure e grafici). La stampante funziona in un solo modo per volta. (I comandi per la stampante grafica vengono ignorati mentre la stampante è nel modo testo, e viceversa).
- (2) I parametri della stampante vengono ripristinati quando la stampante viene commutata dal modo operativo grafici al modo operativo testo. (In altre parole, vengono riinizializzate le impostazioni delle coordinate x ed y delle penne).
- (3) La stampante funziona con l'alimentazione fornita dall'unità principale dell'MZ-700 e non è dotata di un interruttore di alimentazione separato.
- (4) Vengono usati i seguenti interruttori per controllare il funzionamento della stampante.
 - a. Interruttore di alimentazione Fa avanzare la carta.
 - b. Interruttore di ripristino Ripristina (riinizializza) la stampante.
 - c. Interruttore di cambiamento penna Usato quando si sostituiscono le penne nella stampante.
- (5) I colori delle penne sono quattro: nero, blu, verde e rosso.
- (6) Quando la stampante viene usata nel modo operativo testo, è possibile stampare i caratteri in una di tre diverse misure. La misura più grande consente la stampa di un massimo di 26 caratteri per riga, la misura media consente la stampa di un massimo di 40 caratteri per riga mentre la misura più piccola consente la stampa di un massimo di 80 caratteri per riga.

I caratteri che possono essere stampati usando la stampante nel modo operativo testo vengono indicati qui di seguito.

Mentre la stampante è in questo modo operativo, non possono essere prodotti altri simboli, lettere o caratteri grafici.

Nella maggior parte dei casi, i codici ASCII esadecimali verranno stampati in un colore diverso se si tenta di stampare i caratteri grafici con l'istruzione PRINT/P o il comando LIST/T.

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIH  
IJKLMNOPQRSTUVWXYZ[\]^_`~tgh bxdhpcq  
azwsu' Ÿkfv üßjn ūm' }^o!äöä y<  "π□ →▣  
▣▣▣▣↓_!"#$%&'()*+,-./0123456789:;<=>?@A  
BCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`~tgh  
bxdhpcqazwsu' Ÿkfv üßjn ūm' }^o!äöä y<  "  
π□ →▣▣▣▣▣▣↓_!"#$%&'()*+,-./0123456789:  
;<=>
```

2.7.2 Impostazioni iniziali della stampante

Le impostazioni iniziali della stampante in atto quando viene avviato il programma interprete BASIC 1Z-013B, sono le seguenti:

- (1) Colore della penna: Nero
- (2) Posizione della penna: Lato sinistro del carrello. (Prima riga di una pagina)
- (3) Modo operativo: Modo operativo testo
- (4) Dimensione di stampa: 40 caratteri/riga (dimensione standard)
60 righe/pagina

2.7.3 Comandi di specifica del modo operativo

Questi comandi vengono usati per mettere la stampante nel modo operativo testo per la stampa di lettere e di testi numeici. Questo è il modo in atto quando la macchina viene accesa; la dimensione iniziale del carattere è di 40 caratteri/riga.

(1) **MODE TN** (formato abbreviato: **M. TN**)

Questo comando riporta la stampante al modo operativo testo dal modo grafico ed imposta la dimensione dei caratteri a 40 caratteri/riga.

(2) **MODE TL** (formato abbreviato: **M. TL**)

Questo comando riporta la stampante al modo operativo testo dal modo grafico ed imposta la dimensione dei caratteri a 26 caratteri/riga.

(3) **MODO TS** (formato abbreviato: **M. TS**)

Questo comando riporta la stampante al modo operativo testo dal modo grafico ed imposta la dimensione dei caratteri ad 80 caratteri/riga.

*** CHARACTER MODE ***

Modo 80 caratteri

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Modo 40 caratteri

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Modo 26 caratteri

ABCDEFGHIJKLMNOPQRSTUVWXYZ

(4) **MODE GR** (formato abbreviato **M. GR**)

Il comando **MODE GR** viene usato per commutare la stampante dal modo operativo testo al modo operativo grafico per la stampa di diagrammi e grafici. Quando si passa a questo modo operativo, è necessario che il programma BASIC venga eseguito per prendere nota della dimensione di carattere in uso immediatamente prima del cambiamento di modo. Questo serve per tornare al modo operativo testo quando si preme il tasto **BREAK** o si incontra un comando **STOP**.

2.7.4 Comandi per la selezione del colore della penna

PCOLOR n	{	n : 0	nero (formato abbreviato: PC.)
		n : 1	blu	
		n : 2	verde	
		n : 3	rosso	

Questo comando specifica il colore da usare per la stampa dei caratteri o dei grafici. n è un numero da 0 a 3, dove 0 corrisponde al nero, 1 al blu, 2 al verde e 3 al rosso.

Nel modo testo, quando si esegue **PCOLOR** nel modo testo ogni stato è nello stato iniziale tranne la penna a colori.

Per mantenere lo stato corrente eseguite **/PRINT/P CHR\$(29)** prossimo colore.

Questo comando può essere immesso sia nel modo operativo testo che nel modo operativo grafici.

2.7.5 Comandi per il modo operativo testo

2.7.5.1 TEST (formato abbreviato: TE.)

Formato

TEST

Funzione

Con questo comando si ottiene che la stampante stampi dei quadrati in ciascuno dei quattro diversi colori per controllare la specifica del colore, la quantità di inchiostro della penna, e così via. (E' utilizzabile soltanto nel modo testo).



0



1



2



3

.... Valore di n in PCOLOR n
(Nero) (Blu) (Verde) (Rosso)

2.7.5.2 SKIP

Formato

SKIP n

n ... Un numero compreso nella gamma da -20 a 20

Funzione

Questo comando viene usato per trascinare la carta. La carta viene trascinata di n righe nella direzione di avanzamento quando il valore di n è positivo; se il valore specificato per n è negativo, la carta viene trascinata di n righe in direzione inversa. Si noti che si avrà "Printer mode error" se questo comando viene eseguito mentre la stampante è nel modo operativo grafici.

2.7.5.3 PAGE

Formato

PAGE n

n ... Un numero intero nella gamma $1 \leq n \leq 72$

Funzione

Questo comando specifica il numero di righe per pagina. (Eseguibile solo nel modo operativo testo).

2.7.5.4 LIST/P (formato abbreviato: L./P)

Formato

LIST/P o LIST/P <LS-Le>

LS Numero della riga iniziale

Le Numero della riga finale

Funzione

Questo comando stampa la lista di tutte o parte delle righe di programma in memoria sulla stampante. Vedere la spiegazione del comando LIST a pagina 30, per la spiegazione delle procedure per specificare la gamma di righe da stampare. Si noti che, quando nella lista del programma sono inclusi dei caratteri grafici, la maggior parte di essi verrà stampata in un colore diverso come codici ASCII esadecimali. Vedere pagina 156 per i codici ASCII della stampante.

Questo comando può essere eseguito soltanto nel modo operativo testo.

2.7.5.5 PRINT/P (formato abbreviato: ?/P)

Formato

PRINT/P <I₁, d₁, I₂, d₂ ... I_n, d_n>

I_n ... Lista di emissione (espressioni numeriche o di stringa)

d_n ... Delimitatore

Funzione

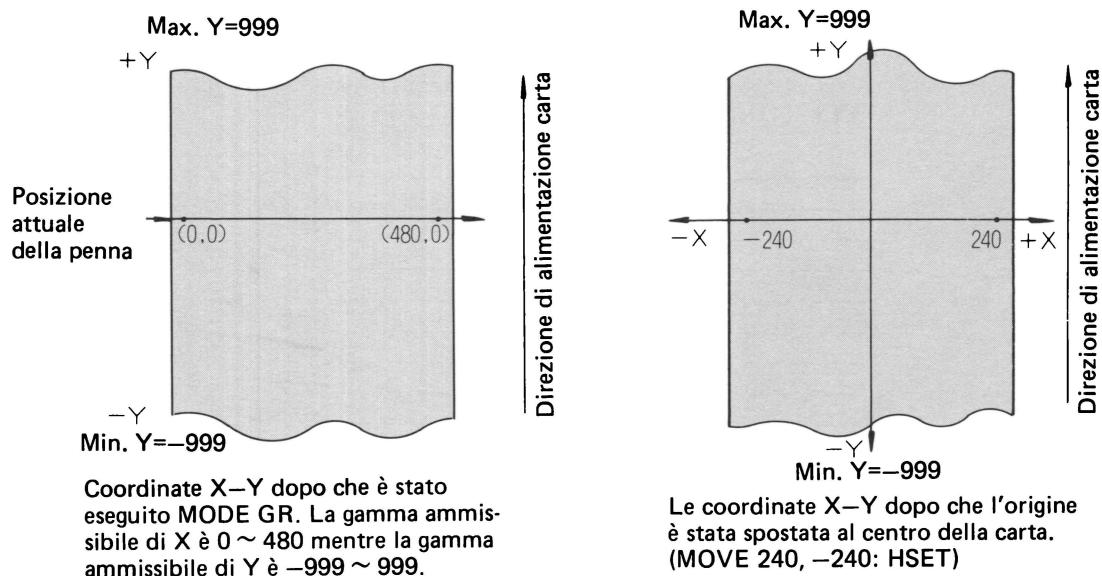
Questo comando emette i dati nella lista di emissione alla stampante. Per i dettagli sull'uso di questo comando, vedere la descrizione del comando PRINT a pagina 35. Vedere le pagina 82 e per la stampa dei caratteri grafici.

2.7.5.6 PRINT/P USING (formato abbreviato: ?/P USI)

Salvo il fatto che l'emissione è diretta alla stampante, è uguale all'istruzione PRINT USING descritta a pagina 36.

2.7.6 Istruzioni per il modo operativo grafico

Le istruzioni per il modo operativo grafico diventano valide dopo che è stata eseguita l'istruzione **MODE GR**. Quando viene eseguita questa istruzione, la posizione attuale della penna viene fissata all'origine ($X = 0, Y = 0$). Tuttavia, l'origine può essere stabilita in qualsiasi posizione. Fate attenzione a non specificare una posizione fuori dall'area di stampa, in quanto questo può danneggiare la penna o provocare altri problemi.



Nota: Vedere pagina 88 per l'istruzione **HSET**.

2.7.6.1 LINE

Formato

LINE $x_1, y_1 <, x_2, y_2, \dots, x_i, y_i >$ or

LINE %n, $x_1, y_1 <, x_2, y_2, \dots, x_i, y_i >$

n Numero intero compreso tra 1 e 16

x_i Numero che indica la coordinata X ($x_i = -480 \sim 480$; il limite varia a seconda della posizione attuale della penna).

y_i Numero che indica la coordinata Y ($y_i = -999 \sim 999$)

Funzione

Questa istruzione traccia una riga dalla posizione attuale della penna alla posizione (x_1, y_1) quindi traccia una riga da (x_1, y_1) a (x_2, y_2) e così via. n specifica il tipo di riga tracciata come illustrato qui di seguito.

n = 1 : linea intera

n = 2 ~ 16 : linea tratteggiata

Se si traslascia %, viene assunto il valore precedente di n. Il valore iniziale di n è 1 (linea intera).

Esempio

(Esempio 1) Il programma seguente traccia un quadrato il cui lato è lungo 240 unità.

10 **MODE GR** Commuta al modo operativo grafici.

20 **LINE 240, 0** Traccia una riga dall'origine al centro della carta.

30 **LINE 240, -240**

40 **LINE 0, -240**

50 **LINE 0, 0** Traccia una riga all'origine.

60 **MODE TN** Ritorna al modo operativo testo.

(Esempio 2) Il programma seguente traccia lo stesso quadrato dell'esempio precedente.

10 **MODE GR**

20 **LINE 240, 0, 240, -240, 0, -240, 0, 0**

30 **MODE TN**

(Esempio 3) Il programma seguente traccia un rettangolo avente un lato lungo 240 unità.

```
1Ø MODE GR
2Ø SQ=INT(12Ø*SQR(3))
3Ø LINE %2, 24Ø, Ø, 12Ø, -SQ, Ø, Ø
4Ø MODE TN
```

Le righe indicate con n sono come segue.

```
*** LINE 1-16 ***
_____ N=1
----- N=2
----- N=3
----- N=4
----- N=5
----- N=6
----- N=7
----- N=8
----- N=9
----- N=10
----- N=11
----- N=12
----- N=13
----- N=14
----- N=15
----- N=16
```

2.7.6.2 RINE (formato abbreviato: RL.)

Formato

RLINE $x_1, y_1 < , x_2, y_2, \dots, x_i, y_i \dots >$
RLINE $\%n, x_1, y_1, < , x_2, y_2, \dots, x_i, y_i \dots >$

n Numero intero da 1 a 16

x_i Numero che indica la coordinata X (-480 ~ 480)

y_i Numero che indica la coordinata Y (-999 ~ 999)

Funzione

Questa istruzione traccia una riga dalla posizione attuale della penna alla posizione indicata dalla coordinata relativa x_1, y_1 , quindi traccia una riga a tale punto dalla posizione indicata dalle coordinate relative x_2, y_2 , e così via. n è uguale come per l'istruzione LINE.

Esempio

Questo programma traccia lo stesso rettangolo dell'esempio 3 precedente.

```
1Ø MODE GR
2Ø SQ=INT(12Ø*SQR(3))
3Ø RLINE %1, 24Ø, Ø, -12Ø, -SQ, -12Ø, SQ
4Ø MODE TN
```

Posizione iniziale della penna

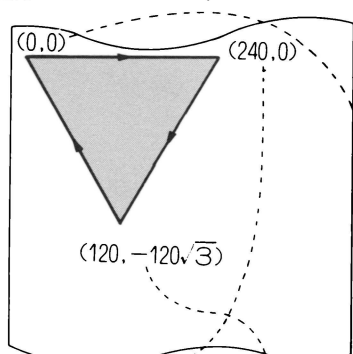


Figura tracciata da LINE

Posizione iniziale della penna

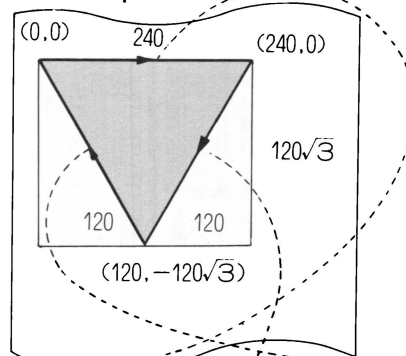


Figura tracciata da RLINE

2. 7. 6. 3 MOVE

Formato

MOVE x, y

x Numero intero che indica la coordinata X (−480 ~ 480)

y Numero intero che indica la coordinata Y (−999 ~ 999)

Funzione

Questa istruzione alza la penna e la sposta alla posizione specificata (x, y).

Esempio

Il programma seguente traccia una croce con una lunghezza di lato di 480 unità.

1Ø MODE GR

2Ø LINE 48Ø, Ø

3Ø MOVE 24Ø, 24Ø..... Alza la penna a (480, 0) e la sposta a (240, 240).

4Ø LINE 24Ø, −24Ø

5Ø MODE TN

Accertatevi di far avanzare la carta prima di eseguire questo programma.

2. 7. 6. 4 RMOVE (farmato abbreviato: **RM.**)

Formato

RMOVE x, y

x Numero intero che indica la coordinata relativa X (−480 ~ 480)

y Numero intero che indica la coordinata relativa Y (−999 ~ 999)

Funzione

Questa istruzione alza la penna e la sposta alla posizione indicata dalla **coordinata relativa** ($\Delta X, \Delta Y$).

Esempio

Il programma seguente traccia la stessa croce dell'esempio fatto per l'istruzione **MOVE**.

1Ø MODE GR

2Ø LINE 48Ø, Ø

3Ø RMOVE −24Ø, 24Ø..... Alza la penna a (480, 0), quindi la sposta di −240 unità nella direzione X e di 240 unità nella direzione Y.

4Ø LINE 24Ø, −24Ø

5Ø MODE TN

Acceratevi di far avanzare la carta prima di eseguire questo programma.

2. 7. 6. 5 PHOME (formato abbreviato: **RH.**)

Formato

PHOME

Funzione

Questa istruzione riporta la penna all'origine.

Esempio

L'esempio seguente traccia la stessa croce in rosso dell'esempio dato per l'istruzione **MOVE**.

1Ø MODE GR

2Ø LINE 48Ø, Ø :MOVE 24Ø, 24Ø

3Ø LINE 24Ø, −24Ø

4Ø PHOME Riporta la penna all'origine.

5Ø PCOLOR 3

6Ø LINE Ø, 24Ø, 48Ø, 24Ø, 48Ø, −24Ø, Ø, −24Ø, Ø, Ø

7Ø MODE TN

2. 7. 6. 6 HSET (formato abbreviato: H.)

Formato
Funzione

HSET

Questa istruzione determina la posizione attuale della penna come nuova origine. Con questa funzione, l'origine può essere fissata nella posizione più appropriata per disegnare le figure. Frequentemente si esegue un'istruzione MOVE prima di eseguire questo comando.

Esempio

```
1Ø MODE GR
2Ø MOVE 24Ø, -24Ø
3Ø HSET ..... Fissa la nuova origine.
4Ø FOR I=1 TO 36Ø STEP 3Ø
5Ø LINE 24Ø*COS (PAI(1)*I/18Ø),24Ø*SIN (PAI(1)*I/18Ø)
6Ø PHOME
7Ø NEXT
8Ø MODE TN
```

2. 7. 6. 7 GPRINT (formato abbreviato GP.)

Formato:

GPRINT [n, @], x\$

GPRINTx\$

n Numero intero che indica la dimensione del carattere (0 ~ 63)

@ Numero intero che indica la direzione in cui vengono stampate le righe di caratteri. (@ = 0 ~ 3)

x\$ Carattere

Funzione:

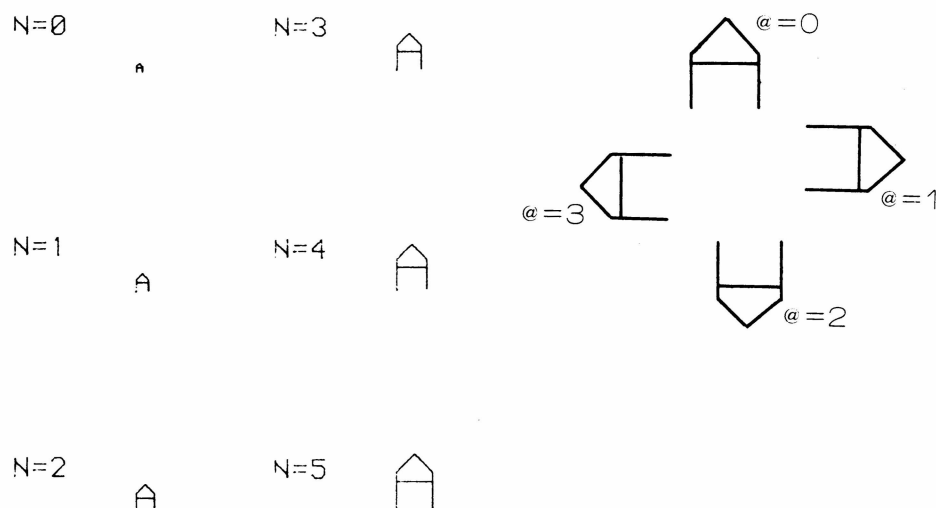
Questa istruzione stampa il carattere specificato usando la dimensione e la direzione specificate. E' possibile stampare 80 caratteri su ogni riga quando n = 0; è possibile stampare 40 caratteri per ogni riga quando n = 1; ed è possibile stampare 26 caratteri per ogni riga quando n = 2.

Quando si omettono n e @, si assumono le impostazioni precedenti. I loro valori iniziali sono n = 1 e @ = 0.

Esempio:

```
1Ø MODE GR
2Ø GPRINT " A " ..... Stampa "A" nel modo operativo grafico.
3Ø GPRINT [2, 2], " A ". Stampa una "A" capovolta nel modo 26
caratteri/riga.
```

Le figure seguenti illustrano diversi esempi di stampati.



2. 7. 6. 8 **AXIS** (formato abbreviato: **AX.**)

Formato: **AXIS** x, p, r

x Numero intero che specifica l'asse tracciato (0 o 1)

p Numero intero che specifica il passo di scala (-999 ~ 999)

r Numero intero che specifica il numero di ripetizioni (1 ~ 255)

Funzione: Questa istruzione traccia l'asse delle X quando x = 1 e l'asse delle Y quando x = 0.
Il numero di segni di scala sono specificati in r e tracciati con un passo p.

Esempio: L'esempio seguente traccia gli assi delle X e delle Y con i segni di scala da -240 a 240 ad intervalli di 10 unità.

1Ø MODE GR..... Commuta la stampante al modo operativo grafico.

2Ø MOVE 24Ø, 5

3Ø GPRINT [1, Ø], " A "

4Ø MOVE 24Ø, Ø..... Alza la penna e la sposta alla posizione A (240, 0)

5Ø AXIS Ø, -1Ø, 48..... Traccia l'asse delle Y dalla posizione A alla posizione B includendo i segni di scala ad intervalli di 10 unità.

6Ø MOVE 24Ø, -5ØØ

7Ø GPRINT [1, Ø], " B "

8Ø MOVE Ø, -24Ø..... Alza la penna e la sposta alla posizione C (0, -240)

9Ø GPRINT [1, Ø], " C "

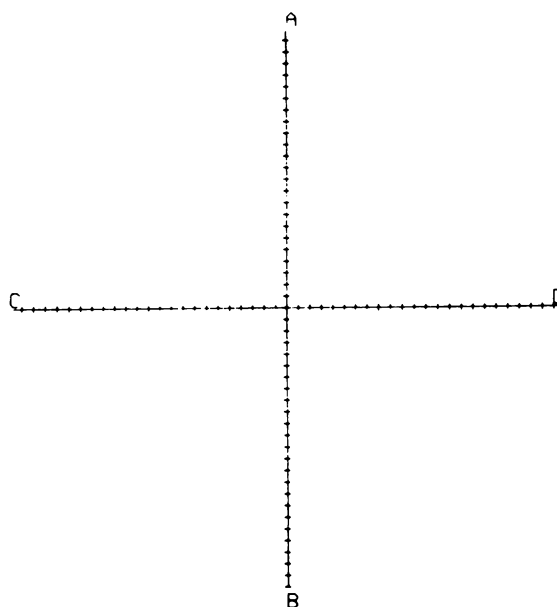
1ØØ MOVE Ø, -24Ø

11Ø AXIS 1, 1Ø, 48..... Traccia l'asse delle X dalla posizione C alla posizione D includendo i segni di scala ad intervalli di 10 unità.

12Ø MOVE 47Ø, -24Ø

13Ø GPRINT [1, Ø], " D "

14Ø MODE TN



Le coordinate possono essere usate come le normali coordinate cartesiane dopo aver impostato il punto di intersezione dell'asse delle X e dell'asse delle Y come nuova origine. (X = -240 ~ 240, y = -240 ~ 240)

2. 7. 6. 9 CIRCLE (formato abbreviato CI.)

Formato

CIRCLE x, y, r, s, e, d

x, y Posizione del centro (-999 ~ 999)

r Raggio (0 ~ 999)

s Angolo iniziale (in gradi)

e Angolo finale (in gradi)

d Angolo di traslazione (in gradi)

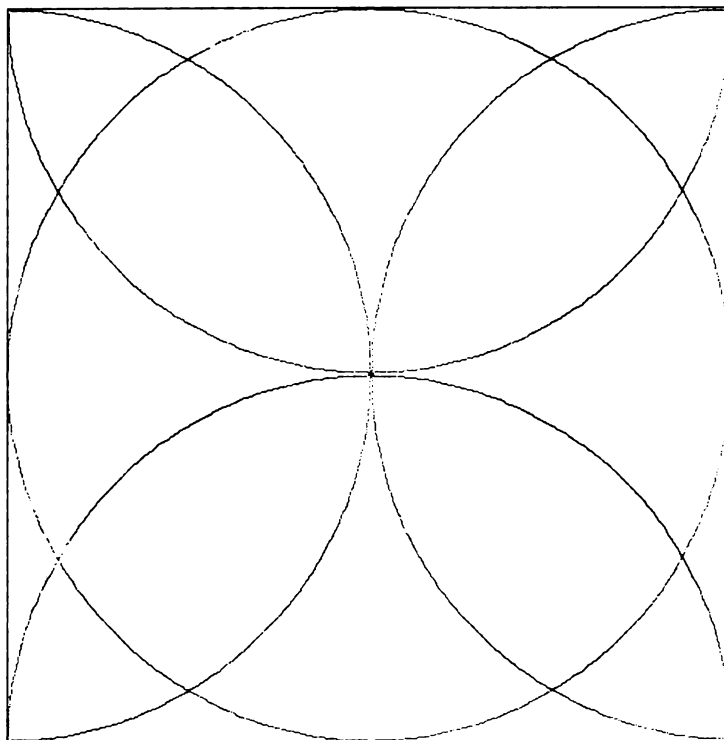
Funzione

Questa istruzione traccia un cerchio o un arco con un raggio r ed un passo d nella posizione (x, y), partendo dall'angolo s e terminando all'angolo e. Viene tracciato un cerchio completo quando s = 0, e = 360 e d = 0.2.

Di fatto questa istruzione traccia un poligono; pertanto d deve essere il più possibile piccolo per tracciare una figura regolare. s deve essere inferiore ad e. Quando d = 0, vengono tracciate le linee che collegano il centro al punto di inizio ed il centro al punto finale.

Esempio

```
1Ø MODE GR
2Ø LINE 48Ø, Ø, 48Ø, -48Ø, Ø, -48Ø, Ø, Ø
3Ø MOVE 24Ø, -24Ø
4Ø HSET
5Ø CIRCLE Ø, Ø, 24Ø, Ø, 36Ø, Ø, 2
6Ø CIRCLE 24Ø, Ø, 24Ø, 9Ø, 27Ø, Ø, 2
7Ø CIRCLE Ø, 24Ø, 24Ø, 18Ø, 36Ø, Ø, 2
8Ø CIRCLE -24Ø, Ø, 24Ø, 27Ø, 45Ø, Ø, 2
9Ø CIRCLE Ø, -24Ø, 24Ø, Ø, 18Ø, Ø, 2
1ØØ MODE TN
```



2.8 Istruzioni di controllo dei programmi linguaggio macchina

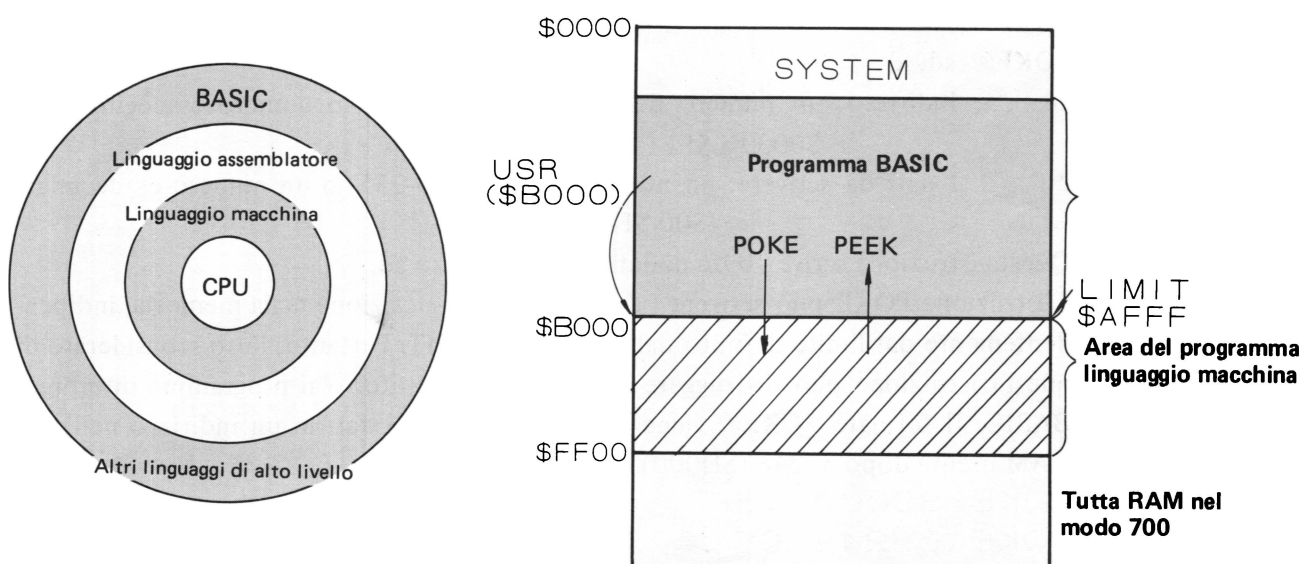
Il programma interprete MZ-700 BASIC supporta parecchie istruzioni di controllo dei programmi linguaggio macchina. Con queste istruzioni, i programmi di linguaggio macchina possono essere collegati ad un programma BASIC.

I linguaggi di programmazione dei calcolatori formano una struttura gerarchica come illustrato qui di seguito. I linguaggi di alto livello come il BASIC eseguono automaticamente il lavoro richiesto quando si usano dei linguaggi di livello inferiore come il linguaggio assembler. Sebbene i linguaggi di livello alto siano convenienti e facili da usare, essi non possono però controllare direttamente il CPU. Il linguaggio di livello più basso (linguaggio macchina) controlla direttamente il CPU e consente un'elevata velocità di elaborazione, ma richiede un'abilità notevole per codificare i programmi lunghi.

Le specifiche di controllo del programma linguaggio macchina consentono delle tecniche di programmazione sofisticate che rendono possibile l'utilizzazione dei vantaggi sia del BASIC che del linguaggio macchina.

I programmi linguaggio macchina possono essere generati e caricati nell'area del programma linguaggio macchina (riservata con l'istruzione BASIC LIMIT) usando il programma monitor o il programma assembler e il programma caricatore. Tali programmi linguaggio macchina possono essere richiamati dai programmi BASIC con la funzione `USR ()`. I programmi linguaggio macchina possono inoltre essere caricati in memoria usando un programma BASIC che utilizza l'istruzione `POKE` per scrivere ogni passo in codice macchina. Il programma linguaggio macchina risultante può quindi essere richiamato dai programmi BASIC con la funzione `USR ()`.

La mappa di memoria al fondo destro trocchia il concetto d'accesso dati con `POKE` e `PEEK`, e di programmi linguaggio macchina con `USR ()`.



2.8.1 LIMIT formato abbreviato LIM.

Formato

LIMIT ad

ad Indirizzo; un numero decimale da 0 a 65279 o un numero esadecimale da 4 posizioni da \$0000 a \$FEFF.

Funzione

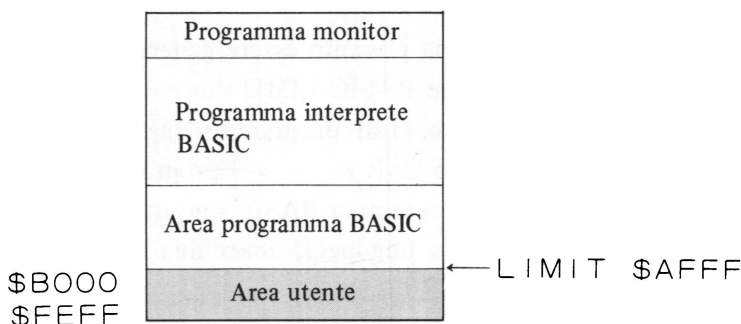
Questa istruzione limita l'area di memoria che può essere usata dal programma interprete BASIC. ad indica che il limite superiore dell'area BASIC e l'area dall'indirizzo seguente (ad + 1) a \$FEFF (65279) possono essere usati per i programmi di linguaggio macchina o i dati speciali.

Esempio

LIMIT \$AFFF

Limita l'area del programma BASIC a \$AFFF.

Nota: L'area da \$FF00 a \$FFFF viene usata dal programma monitor come area di lavoro, pertanto non può essere usata come area utente. L'istruzione **LIMIT** deve essere usata all'inizio di un programma BASIC.



Usate **LIMIT MAX** per annullare il limite definito da **LIMIT** ad.

2.8.2 POKE

Formato

POKE ad, d

POKE@ ad, d

ad Indirizzo; un numero decimale da 0 a 65535 o numero esadecimale da 4 posizioni da \$0000 a \$FFFF.

d I dati da scrivere; un numero decimale (0-255) o un numero esadecimale da 2 posizioni da \$00 a \$FF.

Funzione

Questa istruzione scrive i byte di dati d nell'indirizzo ad.

L'istruzione **POKE** può scrivere i dati in qualsiasi posizione nella memoria, indipendentemente dal limite definito dall'istruzione **LIMIT**. Pertanto, l'uso sconsiderato di questa istruzione può distruggere il programma monitor o il programma interprete BASIC. Il formato **POKE@** viene usato per scrivere i dati in un indirizzo nell'area RAM utente dopo 53248 (\$D000). (Vedere pagina 125).

Esempio

POKE \$D000, \$5F

POKE 53248, 95

Le due istruzioni precedenti svolgono la stessa funzione.

Nota: Un'istruzione **POKE** che specifica un indirizzo dopo \$D000 scrive i dati nell'area RAM del video.

2.8.3 PEEK

Formato

PEEK (ad)

PEEK@ (ad)

Esempio

ad Indirizzo in notazione decimale o esadecimale (0-65535 o \$0000-\$FFFF)
Questa funzione ritorna il contenuto dell'indirizzo specificato come numero decimale da 0 a 255. Usate il formato PEEK@ per svolgere la funzione PEEK in un'area RAM utente dopo 53248 (\$D000).

Funzione

Il programma seguente visualizza i dati memorizzati nell'area da 40960 (\$A000) a 40975 (\$A00F).

```
1Ø FOR AD= 4Ø96Ø TO 4Ø975
2Ø ? PEEK (AD)
3Ø NEXT AD
```

2.8.4 USR (formato abbreviato U)

Formato

USR (ad) oppure

USR (ad, x\$)

ad Indirizzo (decimale o esadecimale da 4 cifre)

x\$ Dati di stringa

Funzione

Questa è una funzione speciale che trasferisce il controllo ad un programma linguaggio macchina che comincia ad un indirizzo specificato. Come con CALL ad, il controllo viene ritornato all'istruzione che segue la funzione USR se il programma linguaggio macchina include un'istruzione di ritorno (RET o RET cc).

Quando si specifica x\$, l'indirizzo iniziale dell'area di memoria contenente x\$ viene caricato nel registro DE, quindi la lunghezza di x\$ viene caricata nel registro BC prima di richiamare il programma linguaggio macchina. Questo consente ad un programma BASIC di passare i dati di stringa ad un programma di linguaggio macchina.

2.8.5 Preparazione programmi linguaggio macchina

In questa sezione viene presentato come esempio un programma linguaggio macchina che riempie l'intero schermo con i caratteri supportati dall'MZ-700. Il programma BASIC seguente carica tale programma macchina in memoria e lo richiama.

```
10 LIMIT $BFFF .....Limita l'area BASIC a $BFFF.
20 GOSUB 50
30 USR($C000) .....Richiama il programma linguaggio macchina.
40 END
50 FOR I =49152 TO 49181
60 READ M .....Legge i dati per il programma linguaggio macchina dalle
70 POKE I,M .....istruzioni DATA e le scrive nell'area del linguaggio
80 NEXT I .....macchina.
90 RETURN

100 DATA 197:REM      PUSH BC .....Inizio dei dati per il programma linguaggio macchina.
110 DATA 213:REM      PUSH DE
120 DATA 229:REM      PUSH HL
130 DATA 22,0:REM      LD D,0
140 DATA 33,0,208:REM  LD HL,D000H
150 DATA 1,232,3:REM  LD BC,1000
160 DATA 243:REM      DI
170 DATA 211,227:REM  OUT (E3H),A .....Commuta il blocco di memoria alla RAM video. (Vedere
180 DATA 114:REM      STO:LD (HL),D .....pagina 157).
190 DATA 35:REM      INC HL .....Determina un codice di visualizzazione per la RAM video.
200 DATA 20:REM      INC D
210 DATA 11:REM      DEC BC
220 DATA 120:REM      LD A,B
230 DATA 177:REM      OR C
240 DATA 194,14,192:REM JP NZ,STO
250 DATA 211,225:REM  OUT (E1H),A .....Commuta il blocco di memoria a RAM. (Vedere pagina
260 DATA 251:REM      EI .....127).
270 DATA 225:REM      POP HL
280 DATA 209:REM      POP DE
290 DATA 193:REM      POP BC
300 DATA 201:REM      RET .....Ritorna al programma BASIC.
```

Se il programma di linguaggio macchina è stato generato con il programma monitor o memorizzato su cassetta con il nome di file DISPLAYCODE, usate il programma seguente per richiamare il programma linguaggio macchina.

```
110 LIMIT $BFFF
110 LOAD "DISPLAYCODE"
120 USR($C000)
```

2.9 Istruzioni I/O

Tutte le unità esterne (inclusi i moduli a floppy disc) sono collegate all'MZ-700 tramite una piastra di interfaccia facoltativa. La piastra di interfaccia universale facoltativa consente all'utente di collegare delle unità esterne quali un plotter X-Y, un perforatore di banda, ed un sintetizzatore musicale all'MZ-700.

Sulla piastra di interfaccia universale è previsto un interruttore di selezione di indirizzo di porta per consentire l'assegnamento di un qualsiasi indirizzo di porta da 0 a 239 (00H-EFH) a qualsiasi unità. Gli indirizzi 240-255 sono riservati alle unità periferiche facoltative fornite dalla Sharp.

Le istruzioni INP ed OUT consentono all'utente di trasferire i dati dalle/alle unità esterne attraverso la scheda di immissione/emissione universale, facoltativa. Il formato di queste istruzioni è il seguente.

INP #P, D Legge i dati da 8 bit dalla porta P, li converte in un numero decimale e li assegna alla variabile D.

OUT #P, D Converte un numero decimale nella variabile D in formato binario e lo emette alla porta D.

Queste istruzioni, ampliano notevolmente la gamma di applicazioni dei calcolatori della serie MZ-700.

2.10 Altre istruzioni

2.10.1 ON ERROR GOTO (formato abbreviato: ON ERR. G.)

Formato	ON ERROR GOTO Lr Lr . . . Numero della riga di destinazione (punto di ingresso della routine di elaborazione errori)
Funzione	Questa istruzione provoca il salto dell'esecuzione al numero di riga Lr se si verifica un errore. Le istruzioni IF ERN e IF ERL vengono usate in una routine di interruzione partendo da tale riga per controllare l'elaborazione successiva a seconda del tipo di errore e del numero di riga in cui l'errore si è verificato. L'inclusione di un'istruzione RESUME alla fine della routine di elaborazione errori rende possibile riportare l'esecuzione alla riga in cui si è verificato l'errore. L'esecuzione di un'istruzione ON ERROR GOTO annulla il numero di riga di interruzione per errore definito dall'istruzione ON ERROR GOTO. La definizione del numero di riga di interruzione per errore viene annullata anche eseguendo un'istruzione CLR.

2.10.2 IF ERN

Formato	IF (espressione di confronto utilizzando ERN) THEN Lr IF (espressione di confronto utilizzando ERN) THEN (istruzione) IF (espressione di confronto utilizzando ERN) GOTO Lr Lr . . . Numero della riga di destinazione.
Funzione	Questa istruzione fa saltare l'esecuzione alla routine (interruzione) di elaborazione degli errori cominciando dalla riga Lr oppure esegue l'istruzione che segue THEN quando il risultato della < espressione di confronto utilizzando ERN > è vero. ERN è una funzione speciale che ritorna un numero corrispondente all'errore verificatosi. Vedere pagina 163 per i numeri errati.
Esempio	Quanto segue illustra una routine di elaborazione degli errori che comincia alla riga 1000, la quale provoca il salto dell'esecuzione alla riga 1200 se il numero di errore è 5. 1000 ON ERROR GOTO 1000 Dichiaro il numero di riga della routine di elaborazione degli errori. 1000 IF ERN=5 THEN 1200 Salta a 1200 se si è verificato un errore di superamento della capacità della stringa.

2.10.3 IF ERL

Formato

IF (espressione di confronto utilizzando ERL) THEN Lr
IF (espressione di confronto utilizzando ERL) THEN istruzione
IF (espressione di confronto utilizzando ERL) GOTO Lr

Funzione

Questa istruzione provoca il salto dell'esecuzione alla routine con inizio alla riga Lr oppure esegue l'istruzione che segue THEN quando il risultato della < espressione di confronto utilizzando ERL > è vero.

ERL è una funzione speciale che ritorna il numero di riga in cui si è verificato un errore.

Esempio

L'istruzione seguente provoca il salto dell'esecuzione alla riga 1300 se si è verificato un errore alla riga 250.

```
1010 IF ERL = 250 THEN 1300
```

L'istruzione seguente ritorna il controllo alla riga 520 nella routine principale se il numero di errore è 43 ed il numero della riga dell'errore è diverso da 450.

```
1020 IF (ERN = 43) * (ERL < > 450) THEN RESUME 520
```

2.10.4 RESUME (formato abbreviato: RESU.)

Formato

RESUME < NEXT >

RESUME Lr

Lr . . . numero di riga o 0

Funzione

Questa istruzione restituisce il controllo alla routine principale da una routine di elaborazione degli errori.

Discussione

Il sistema memorizza il numero della riga in cui si è verificato l'errore e riporta l'esecuzione del programma su quella riga oppure su un'altra riga indicata dopo che è stato corretto l'errore.

L'istruzione RESUME potrebbe essere usata in una delle quattro forme seguenti:

RESUME..... Ritorna alla riga di errore.

RESUME NEXT..... Ritorna alla riga successiva alla riga di errore.

RESUME Lr..... Ritorna alla riga Lr.

RESUME Ø..... Ritorna all'inizio della routine principale.

Se si incontra l'istruzione RESUME quando non si è verificato nessun errore, avviene l'errore 21 (RESUME ERROR).

Se non è possibile eseguire l'istruzione RESUME, si verifica l'errore 20 (CAN'T RESUME ERROR).

2.10.5 SIZE

Formato

PRINT SIZE

Funzione

Questa è una funzione speciale che ritorna il numero di byte in memoria che possono essere usati per la memorizzazione dei programmi BASIC.

Per esempio, PRINT SIZE visualizza il numero di byte liberi di area della memoria.

2.10.6 PLOT ON

Formato

PLOT ON

Funzione

Questa istruzione rende possibile l'uso della stampante-plotter a colori come unità video. Pertanto, l'MZ-700 può essere usato senza uno schermo video esterno.

Questa istruzione funziona soltanto quando è installata la stampante-plotter a colori ed è stata precedentemente eseguita l'istruzione **MODE TN**.

Esempio

PLOT ON

Nota

Viene stampato un punto “.” per rappresentare gli eventuali caratteri che non sono memorizzati nel generatore di caratteri della stampante-plotter a colori (vedere pagina 160). I tasti **INST**, **DEL** e “←” vengono disabilitati eseguendo questa istruzione. E' possibile usare **CTRL** + **G** per cambiare la penna.

2.10.7 PLOT OFF

Formato

PLOT OFF

Funzione

Questa istruzione annulla il modo operativo **PLOT ON** della stampante-plotter.

Esempio

PLOT OFF

2.10.8 CONSOLE (formato abbreviato: **CONS.**)

Formato

CONSOLE < Is, In, < Cs, Cn > >

Is : Riga iniziale dell'area di scorrimento

In : Numero di righe nell'ambito
dell'area di scorrimento

Cs : Colonna iniziale dell'area di
scorrimento

Cn : Numero di colonne all'interno
dell'area di scorrimento

Esempio

CONSOLE Ø, 25, Ø, 4Ø

CONSOLE 5, 15

CONSOLE Ø, 25, 5, 3Ø

CONSOLE Ø, 1Ø, Ø, 1Ø

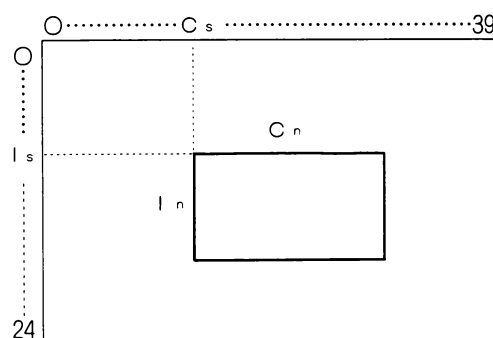
CONSOLE

Funzione

Questa istruzione specifica l'ampiezza dell'area di scorrimento; cioè l'area che viene annullata da **PRINT** "■".

Il primo esempio specifica l'intero schermo come area di scorrimento. Il secondo specifica l'area compresa tra le righe 5 e 15 come area di scorrimento. Il terzo specifica l'area tra le colonne 5 e 30 come area di scorrimento. Il quarto specifica le posizioni 10 x 10 nell'angolo superiore di sinistra dello schermo come area di scorrimento.

Questa istruzione è utile per escludere i margini di sinistra e/o di destra dell'immagine dall'area di visualizzazione; quando questi sono nascosti dietro ai bordi dello schermo. l'ultimo esempio non specifica l'area di scorrimento. Quando l'area di scorrimento non è specificata, è possibile far scorrere lo schermo verso l'alto o verso il basso. Tuttavia, questo rende più difficile eseguire la revisione dello schermo in quanto i valori di Cn ed In diventano più piccoli.



2.11 Funzione monitor

La sezione IOCS del programma interprete BASIC include il programma monitor per facilitare l'immissione dei programmi linguaggio macchina. Questo programma monitor usa l'area stack (impilamento) da FF00H a FFFFH come area stack.

Questo programma monitor include il programma revisore dello schermo simile a quello del BASIC consentendo di cambiare il contenuto di qualsiasi indirizzo nell'ambito dell'area RAM 64K come descritto qui di seguito.

2.11.1 Formato di revisione

: indirizzo = dati dati dati
: (due punti) ... Indica che è possibile rivedere la riga successiva.
indirizzo ... Indica l'indirizzo iniziale dell'area di memoria di cui si può cambiare il contenuto (cifre esadecimali 4)
= ... Separa i dati dall'indirizzo.
dati ... Numero esadecimale di due posizioni o un punto e virgola “ ; ” più il carattere che è scritto nell'indirizzo specificato. Uno spazio in bianco viene usato per separare i dati adiacenti.

2.11.2 Comando di commutazione stampante (comando P)

Formato * P

Questo comando commuta l'emissione dati con il comando D o F tra la stampante ed il video. Se la stampante non è collegata al calcolatore, viene visualizzato il messaggio “ERR?” ed il calcolatore rimane in attesa di un altro comando. Controllate il collegamento della stampante o eseguite nuovamente il comando P per commutare l'unità di emissione al video.

2.11.3 Comando di riversamento (Comando D)

Formato * D < indirizzo iniziale < _ indirizzo finale > >

Questo comando riversa il contenuto della memoria dall'indirizzo iniziale all'indirizzo finale. Se l'indirizzo finale è omesso, viene riversato il contenuto del blocco di 128 byte che comincia all'indirizzo specificato. Se entrambi gli indirizzi sono omessi, riversa il contenuto del blocco di 128 byte successivo al blocco di memoria precedentemente riversato. Il formato in cui i dati vengono riversati è il seguente.

: HHHH=HH_HH_HH HH HH HH HH HH /ABCDE. G.
↑
Indirizzo di partenza 8 byte (Codice esadecimale) 8 byte (Caratteri)

Il contenuto di qualsiasi posizione può essere cambiato spostando il cursore al byte corrispondente, immettendo i nuovi dati e premendo il tasto CR .

Nota: I codici di controllo vengono visualizzati come un punto (.) nel campo dati dei caratteri. Premendo

il tasto BREAK si arresta l'operazione di riversamento, e premendo simultaneamente i tasti

SHIFT e BREAK si ritorna il monitor al modo di immissione comandi.

2.11.4 Comando di impostazione della memoria (comando M)

Formato * M [indirizzo iniziale]

Questo comando viene usato per cambiare il contenuto della memoria. Se l'indirizzo iniziale è omissso, si assume l'indirizzo attualmente indicato dal contatore del programma. Premete insieme i tasti SHIFT e BREAK per porre termine a questo comando. Quando questo comando viene immesso, l'indirizzo iniziale del blocco di memoria ed il suo contenuto vengono visualizzati nel formato di revisione precedentemente descritto ed il cursore viene spostato sui dati da cambiare. Immettete i nuovi dati e premete il tasto CR. vengono quindi visualizzati l'indirizzo successivo ed il suo contenuto.

2.11.5 Comando di ricerca (comando F)

Formato * F [indirizzo iniziale] _ [indirizzo finale] _ [dati] _ [dati] _

Questo comando ricerca la stringa di dati specificata nell'area di memoria dall'indirizzo iniziale all'indirizzo finale. Quando la stringa viene trovata, l'indirizzo della stringa ed il suo contenuto vengono riversati sullo schermo. Questo comando viene terminato premendo simultaneamente i tasti SHIFT e BREAK.

2.11.6 Chiamata di sottoroutine (comando G)

Formato * G [call address]

Questo comando richiama la sottoroutine che comincia all'indirizzo specificato. L'indicatore di stack si trova a FFFE_H.

2.11.7 Comando di trasferimento (comando T)

Formato * T [indirizzo iniziale] _ [indirizzo finale] _ [indirizzo di destinazione]

Questo comando trasferisce il contenuto della memoria tra l'indirizzo iniziale e l'indirizzo finale all'area di memoria che comincia in corrispondenza dell'indirizzo di destinazione.

2.11.8 Comandi di memorizzazione (salvataggio) (comando S)

Formato * S [indirizzo iniziale] _ [indirizzo finale] _ [indirizzo di esecuzione] : [nome file]

Questo comando memorizza (salva) il contenuto della memoria tra l'indirizzo iniziale e l'indirizzo finale sulla cassetta con il nome di file specificato.

2.11.9 Comando di caricamento (comando L)

Formato * L < indirizzo di caricamento > < : nome file >

Questo comando carica il file specificato in memoria partendo dall'indirizzo di caricamento. Se l'indirizzo di caricamento è omissso, viene assunto come indirizzo di caricamento l'indirizzo di esecuzione contenuto nel file. Se il nome del file viene omissso, viene caricato il primo file del nastro. Viene visualizzato il messaggio "ERR?" se viene rilevato un errore di totale di controllo oppure se viene premuto il tasto BREAK durante l'esecuzione, quindi il monitor ritorna al modo di immissione comandi. Il modo di immissione comandi viene immesso quando l'esecuzione è completata.

2.11.10 Comando di verifica (comando V)

Formato

 * V < nome file >

Questo comando legge il file specificato dalla cassetta e lo confronta con il contenuto della memoria. Questo consente di avere la conferma che un programma sia stato correttamente registrato con il comando SAVE. Se si riscontrano delle differenze tra i dati letti dal nastro e quelli contenuti in memoria, viene visualizzato il messaggio "ERR?".

2.11.11 Comando di ritorno (comando R)

Formato

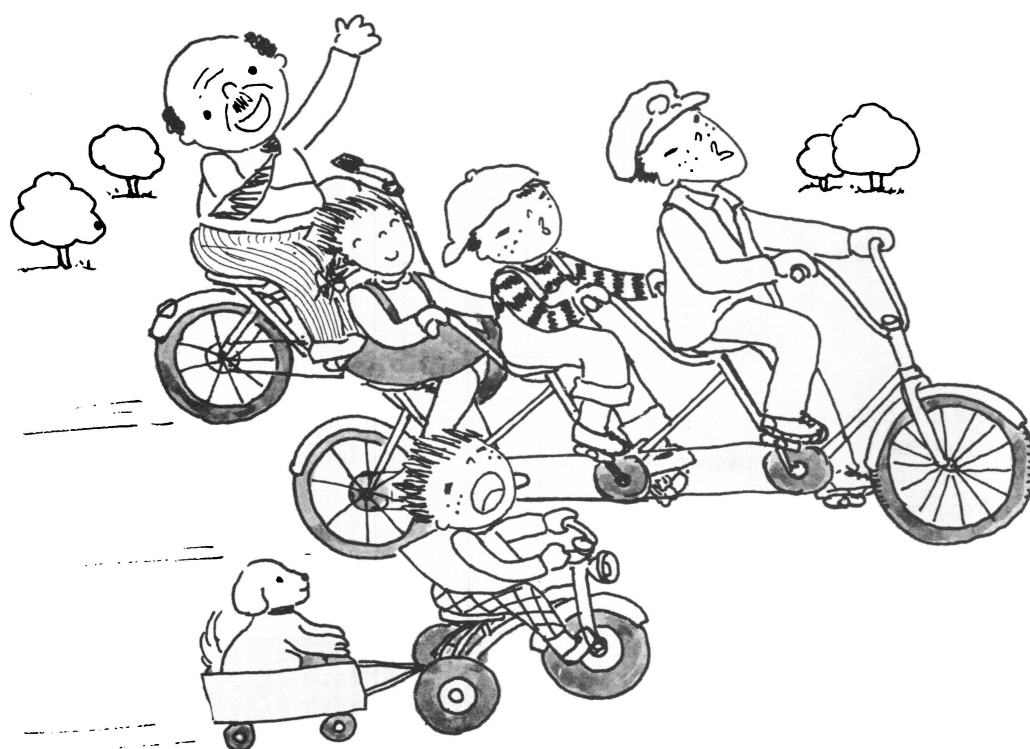
 * R

Questo comando ritorna il controllo al programma del sistema che ha richiamato il programma monitor e ripristina il registro SP (indicatore stack) ed HL ai valori che questi contenevano quando è stato richiamato il programma monitor. L'esecuzione viene ripresa quando viene eseguito il comando che segue BYE.

Questo comando non può ritornare il controllo se il monitor è stato richiamato da un programma del sistema il cui indicatore di stack è tra FF00H e FFFFH, oppure se l'indicatore di stack non contiene un indirizzo di ritorno. In questi casi, usate il comando G per richiamare il punto di ingresso di inizio caldo.

Uso dell 'MZ-700

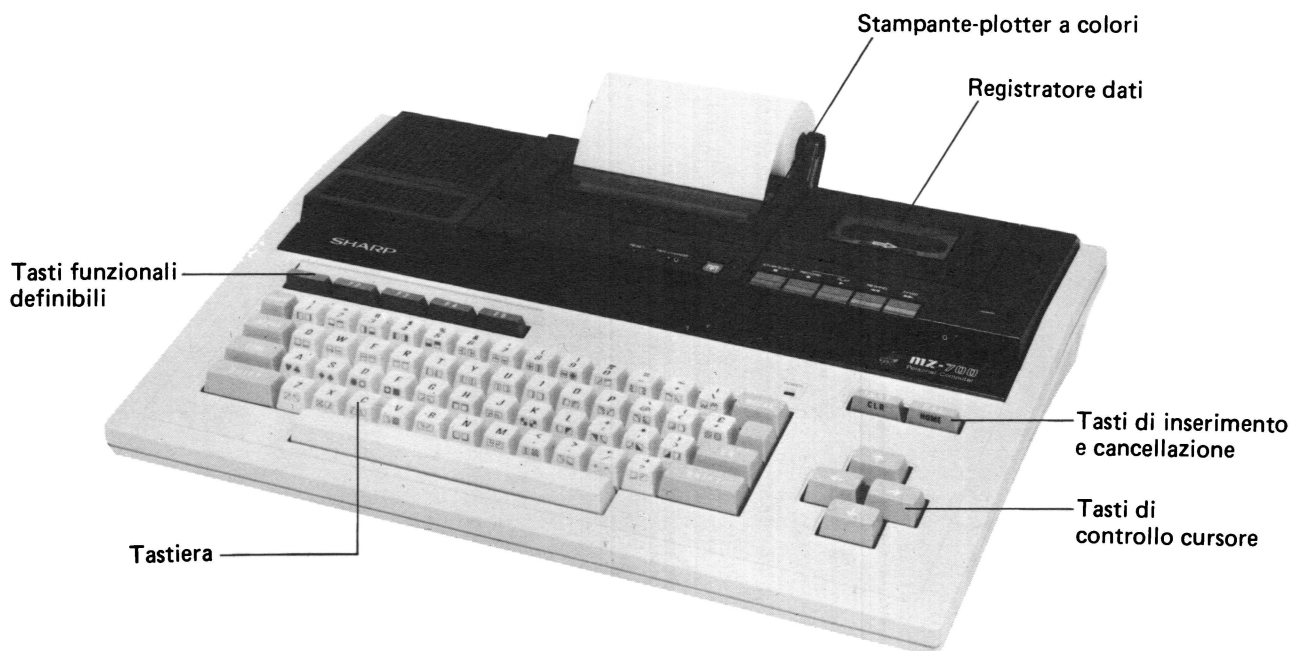
Capitolo 3



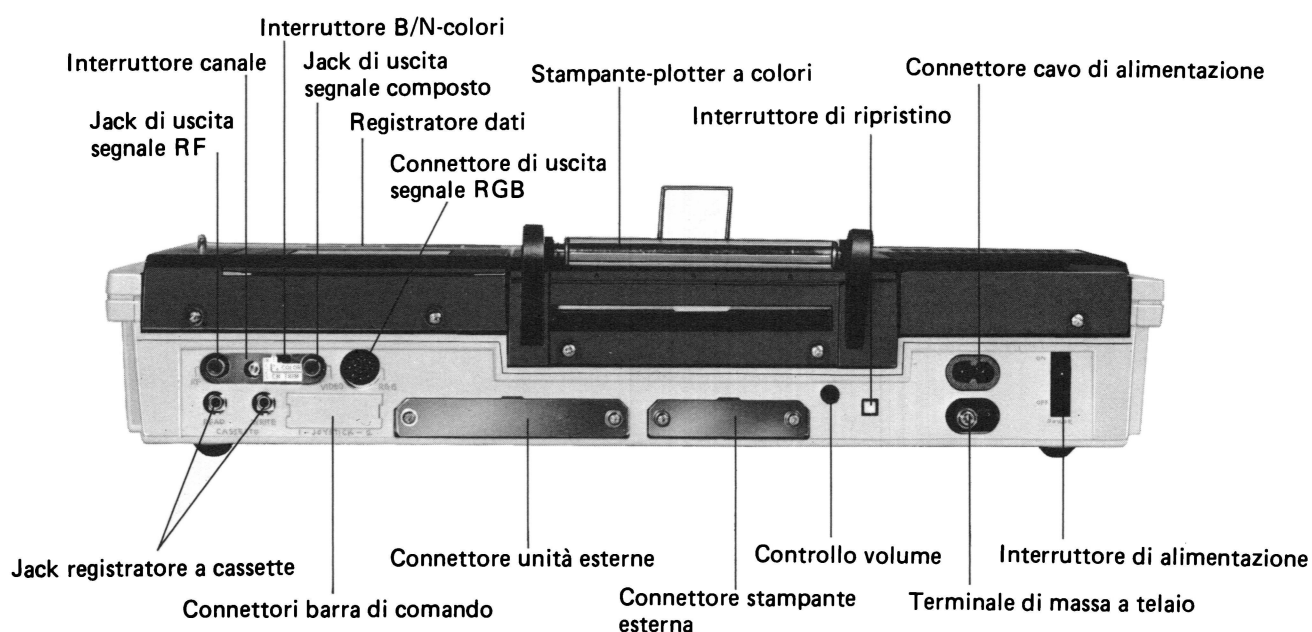
3. 1 Aspetto dei personal computer della serie MZ-700

3.1.1 MZ-731

■ Vista anteriore

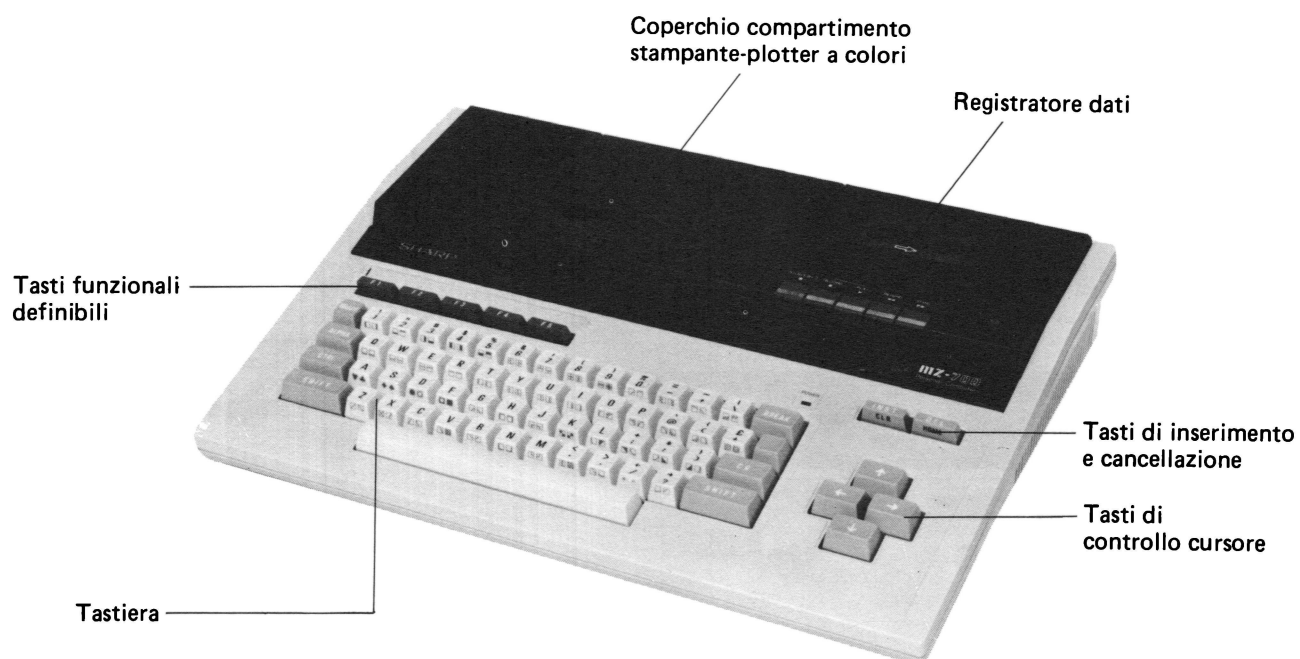


■ Vista posteriore



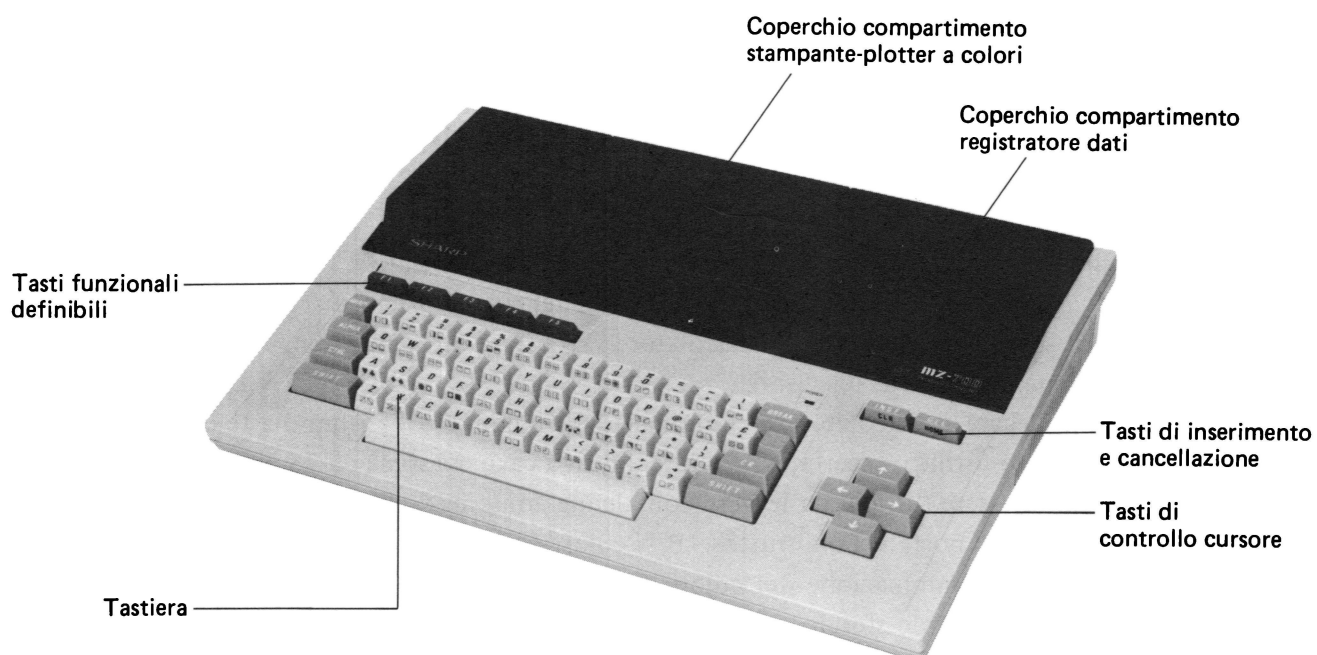
3.1.2 MZ-721

■ Vista anteriore



3.1.3 MZ-711

■ Vista anteriore

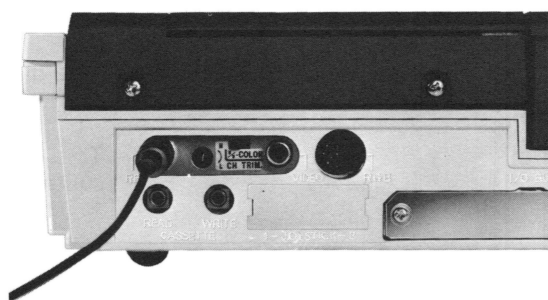


3.2 Collegamento all'unità video

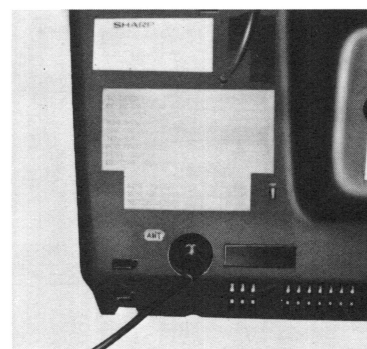
Accertatevi di spegnere sia il calcolatore che l'unità video prima di collegarli.
Quando allacciate lo MZ-700 all'apparecchio TV dovete staccare l'antenna.

3.2.1 Collegamento del televisore all'MZ-700

Staccate l'alimentatore dell'antenna dai terminali dell'antenna UHF del televisore. Inserite la spina del cavo apposito nel jack di uscita segnale RF sul pannello posteriore del calcolatore e collegate i capocorda ad Y sull'altra estremità del cavo ai terminali di antenna UHF 75-ohm sul televisore.



Vista posteriore del MZ-700

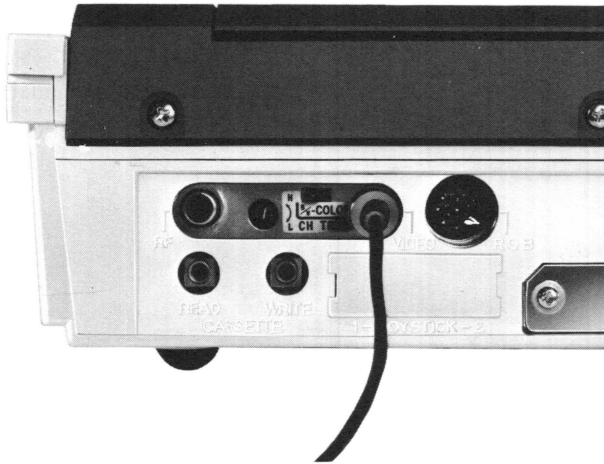


Vista posteriore della TV domestica

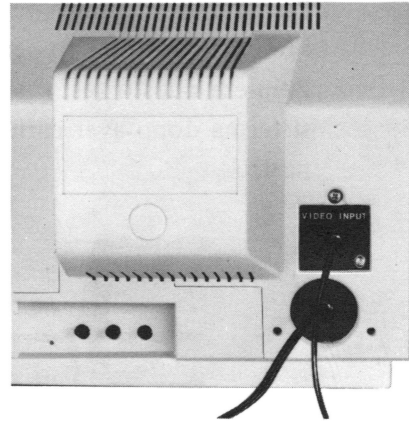
- Regolate i controlli (sintonizzazione, controllo colore, ecc.) del televisore alle condizioni ottimali prima di collegarlo al calcolatore.
- Notate che il colore e la qualità delle immagini visualizzate saranno di qualità inferiore con un televisore che non con un'unità monitor a colori speciale. Notate inoltre che le immagini possono essere di colore errato o non essere a colori se il televisore non è regolato correttamente.
- Parte dello schermo può essere omessa se le frequenze di esplorazione in verticale ed in orizzontale del televisore non corrispondono a quelle del calcolatore. Questo non è un problema del calcolatore; rivolgetevi al vostro rivenditore del televisore.
- Parte dello schermo non può essere visibile se l'immagine non è centrata.
- Accertatevi di togliere l'alimentatore dell'antenna dal televisore prima di collegarlo al calcolatore; diversamente, il segnale proveniente dal calcolatore verrà irradiato dall'antenna del televisore, interferendo possibilmente con altri televisori.
- Accertatevi di collegare il calcolatore ai terminali dell'antenna da 75 ohm del televisore. Se non è possibile usare il cavo fornito, accertatevi di usare un cavo **cavo coassiale da 75 ohm**.
- I caratteri possono essere difficili da leggere con alcune combinazioni di colori di primo piano e di sfondo. In questi casi, portate l'interruttore B/N-colori in posizione B/N per ottenere un maggior contrasto. La miglior combinazione dei colori di primo piano e di sfondo è il bianco per il primo piano ed il nero o il blu per lo sfondo.
- Non è incluso alcun segnale audio nel segnale RF alimentato al televisore, per cui l'altoparlante del televisore non può emettere alcun suono.

3.2.2 Collegamento dello schermo verde da 12 pollici MZ-1D04 al calcolatore

Usate il cavo incluso nello schermo verde MZ-1D04 per collegarlo al calcolatore. Collegate un'estremità del cavo nel jack del segnale composto sul pannello posteriore del calcolatore, inserite l'altra estremità nel jack di ingresso sull'MZ-1D04, quindi mettete l'interruttore B/N-colori in posizione B/N.



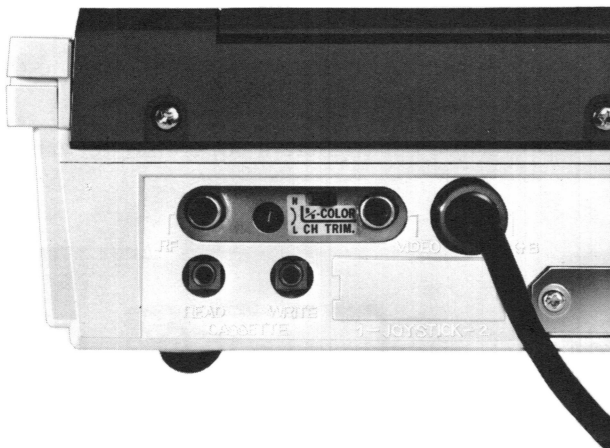
Pannello posteriore del calcolatore serie MZ-700



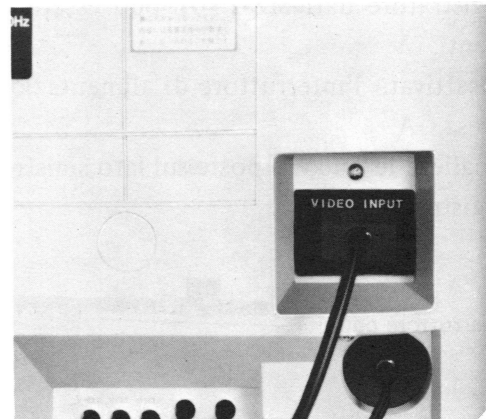
Pannello posteriore dell'MZ-1D04.

3.2.3 Collegamento dello schermo a colori da 14 pollici MZ-1D05 al calcolatore

Usate il cavo incluso nello schermo a colori MZ-1D05 per collegarlo al calcolatore. Inserite il connettore DIN del cavo nel connettore di uscita del segnale RGB sull'MZ-700 ed.

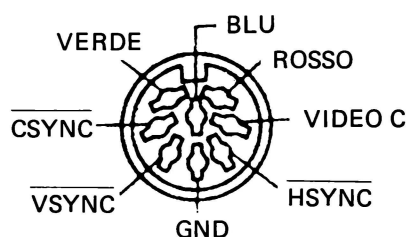


Pannello posteriore del calcolatore serie MZ-700



Pannello posteriore dell'MZ-1D05

Le assegnazioni del pin del connettore di uscita del segnale RGB dell'MZ-700 sono come illustrato qui di seguito.



Connettore DIN di uscita del segnale RGB
(visto dal lato posteriore)

3.3 Registratore dati

■ Registratore dati incorporato nell'MZ-731 ed MZ-721

Il registratore dati incorporato può essere usato come un normale registratore a cassette.

RECORD

Premete questo tasto per registrare i programmi ed i dati.

PLAY

Premete questo tasto per caricare i programmi ed i dati.

REWIND

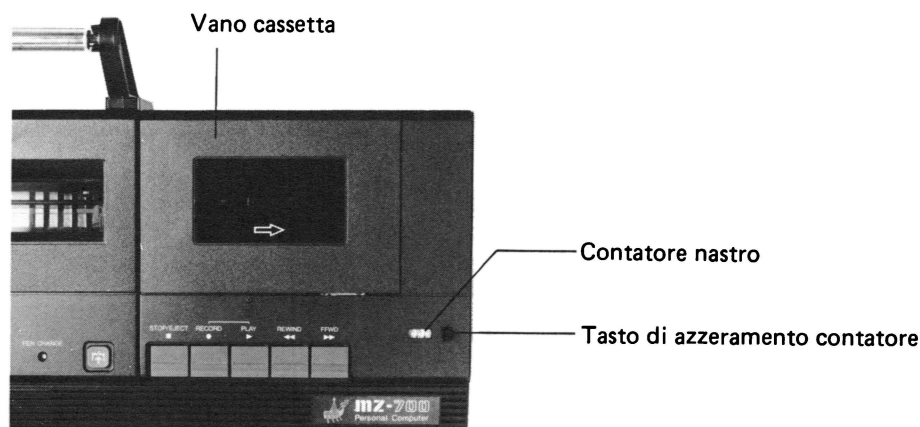
Premete questo tasto per riavvolgere il nastro.

FFWD

Premete questo tasto per far avanzare velocemente il nastro.

STOP/EJECT

Premete questo tasto per arrestare il nastro, per sbloccare altri tasti quando il nastro si ferma dopo aver caricato o registrato i programmi o i dati, oppure per espellere il nastro.



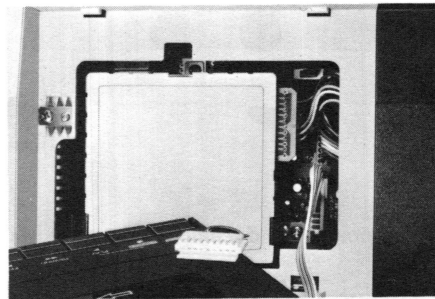
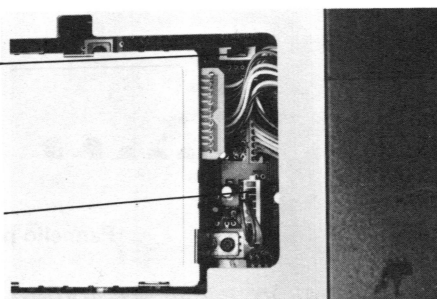
■ MZ-1T01

Il registratore dati MZ-1T01 può essere installato nell'MZ-711. Le procedure di installazione sono le seguenti.

1. Disattivate l'interruttore di alimentazione del calcolatore e disinserite il cavo di alimentazione dalla presa CA.
2. Togliete le due viti poste sul lato sinistro del pannello posteriore per togliere il coperchio del vano del registratore.

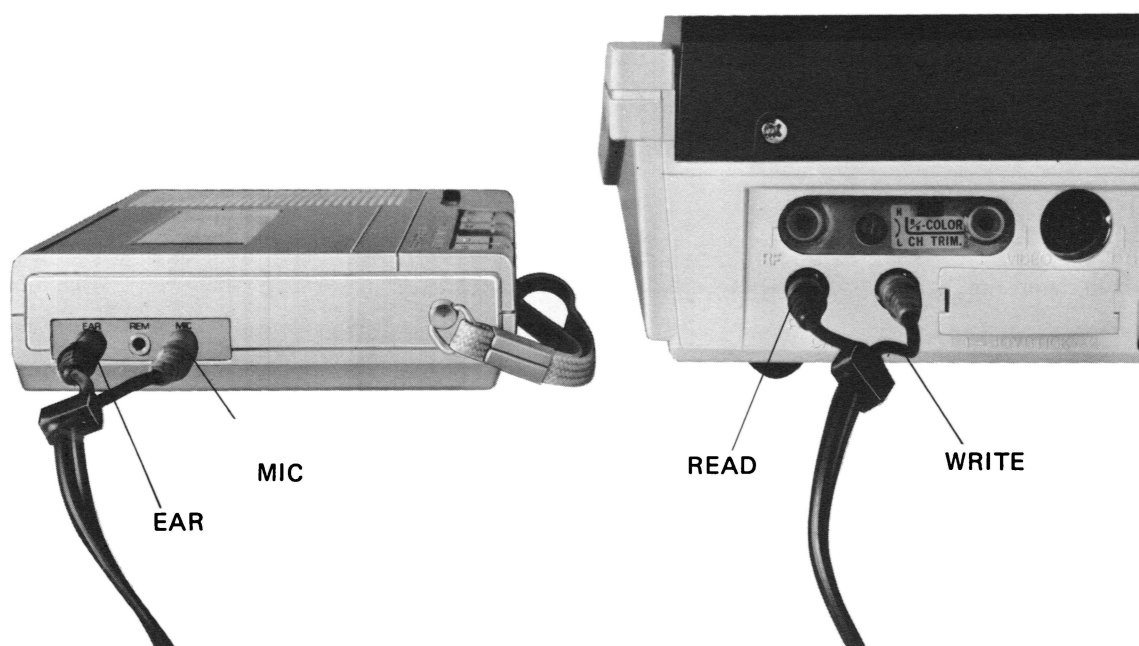
Interruttore polarità

Connettore congiunto



3. Togliete il coperchio del connettore congiunto.
4. Inserite il connettore dell'MZ-1T01 sul connettore a 9 pin posto sul lato posteriore sinistro del vano registratore dell'MZ-711.
5. Posizionate il registratore dati nel vano apposito e fissatelo in posizione con le due viti. Durante questa operazione, fate attenzione ad evitare di pizzicare il cavo del connettore tra il registratore dati ed il calcolatore (diversamente non è possibile serrare le viti).

■ Normale registratore a cassette

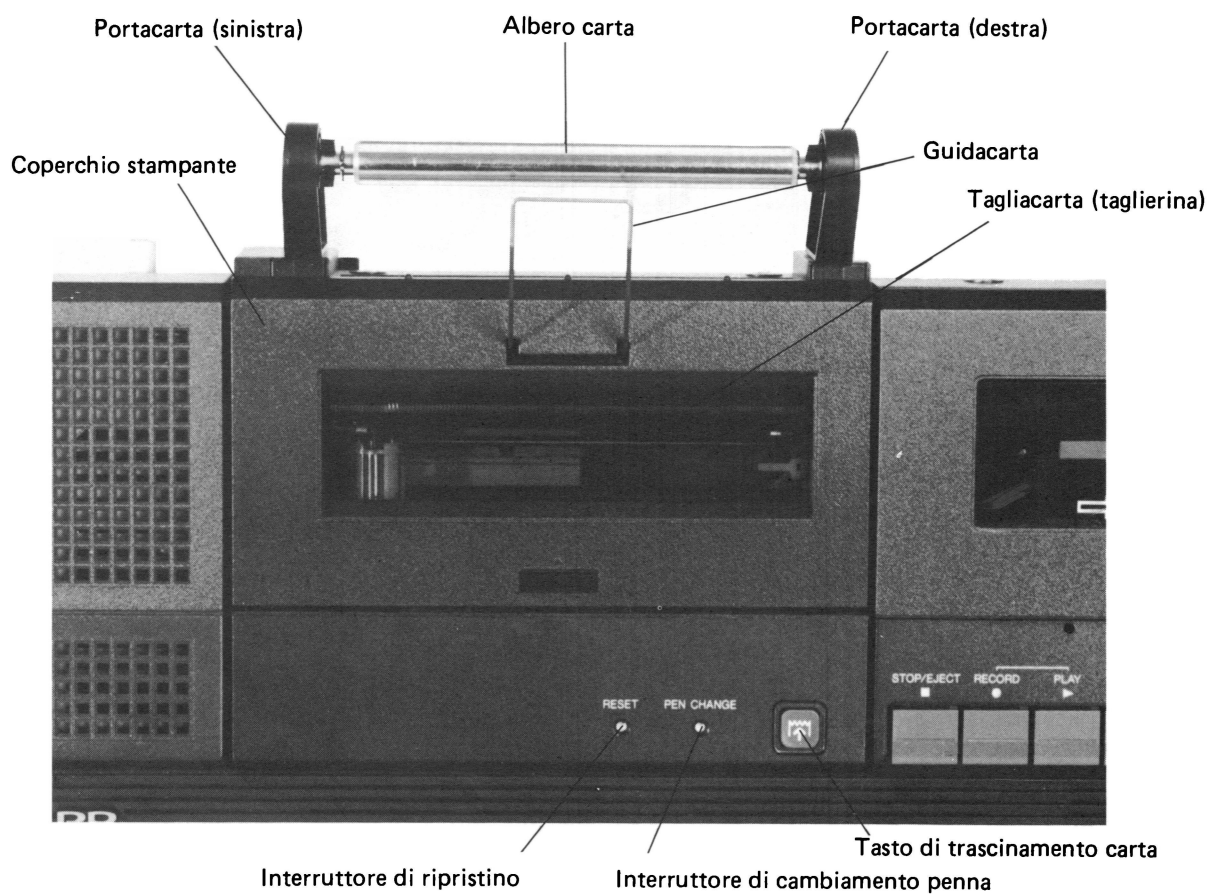


Usando i cavi audio commercialmente disponibili con minispine da 3,5 mm, collegate il jack WRITE del calcolatore al jack MIC del registratore a cassette e collegate il jack READ del calcolatore al jack EXT SP o EAR del registratore a cassette.

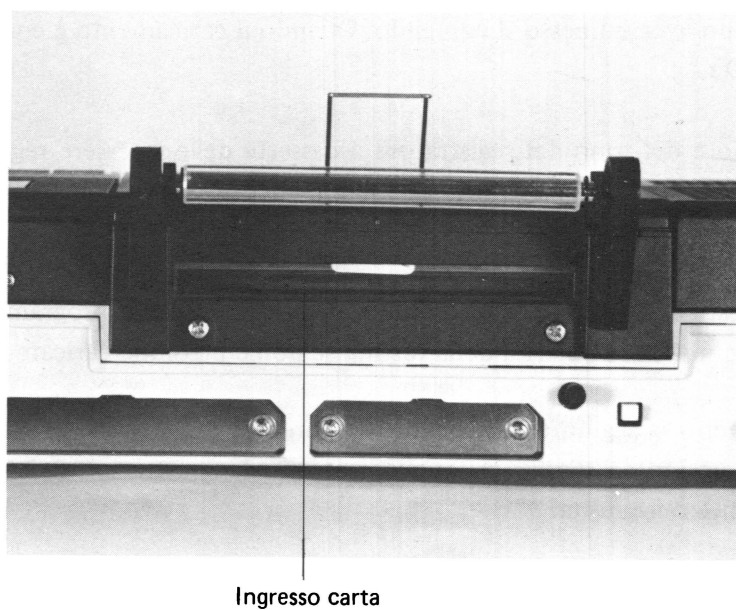
Fate attenzione a quanto segue quando usate un normale registratore a cassette.

- (1) Il messaggio " \perp RECORD. PLAY " non compare quando viene immesso un comando SAVE. Accertatevi di premere il tasto RECORD sul registratore prima di immettere questo comando. Premete il tasto STOP per fermare il registratore dopo che è stato visualizzato il messaggio "READY". Senza la pressione del tasto STOP, il registratore non si ferma.
- (2) Il messaggio " \perp PLAY " non compare quando viene immesso un comando LOAD. Accertatevi di avviare il nastro dopo aver immesso il comando. Quando il caricamento è completo, viene visualizzato il messaggio "READY".
- (3) I controlli del livello e del tono del registratore a cassette devono essere regolati ai livelli appropriati. Alcuni registratori a cassette (per esempio quelli con il controllo automatico del livello) non possono essere utilizzabili. In questi casi, acquistate l'MZ-1T01.
- (4) La polarità della testina può rendere impossibile il caricamento dei programmi forniti con il calcolatore. Cercate di commutare la polarità della testina se non è possibile caricare i programmi.
- (5) In caso di trasferimenti o aggiunte, usate il registratore utilizzato per la registrazione. Se il registratore del trasferimento o aggiunta è diverso da quello usato per la registrazione, risulta impossibile effettuare qualsiasi trasferimento o aggiunta.
- (6) I dati scritti usando un normale registratore a cassette possono non essere leggibili con il registratore dati. Pertanto si consiglia l'uso dell'MZ-1T01.

3.4 Stampante-plotter a colori




Stampante-plotter (vista dall'alto)



Stampante-plotter (vista dal lato posteriore)

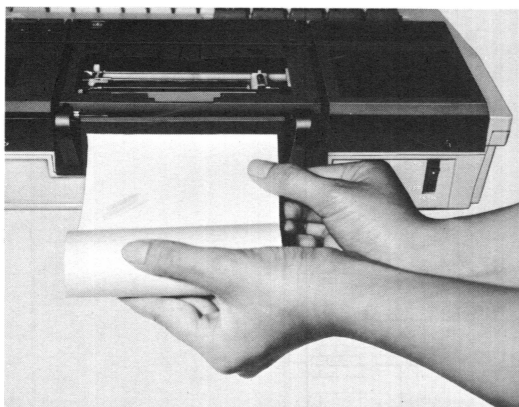
■ Caricamento della bobina di carta

1. Togliete il coperchio della stampante.
 2. Tagliate l'estremità della bobina di carta ed inseritela nell'ingresso della carta. (Fate attenzione a non piegare o stropicciare l'estremità della carta durante questa operazione).
 3. Attivate l'interruttore di alimentazione dell'MZ-731 e premete il tasto  (trascinamento carta) per trascinare la carta sino a che il margine superiore della carta si trova a 3-5 cm sopra all'uscita.
 4. Inserite l'albero nella bobina e montatelo sui portacarta.
 5. Installate il coperchio della stampante in modo che l'estremità della carta esca attraverso la taglierina.
- Per togliere il rullo dalla stampante per la sostituzione, tagliate la carta in corrispondenza dell'ingresso e premete il tasto di trascinamento della carta.

- Le bobine di carta per i calcolatori della serie MZ-700 sono disponibili presso qualsiasi rivenditore Sharp. Non usate carta diversa da quella specificata.

La lunghezza della carta è di 23-25 metri, ed il diametro massimo della bobina che può essere caricata è di 50 mm. La carta non avanza correttamente se si usa una bobina di diametro superiore, il che porta ad una scarsa qualità di stampa.

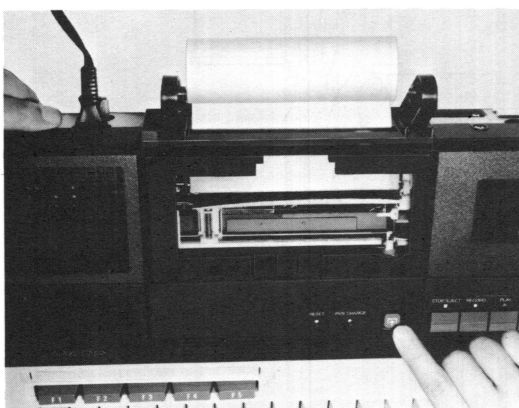
Procedure per il caricamento della bobina di carta



(A) Inserite la carta nell'ingresso apposito.



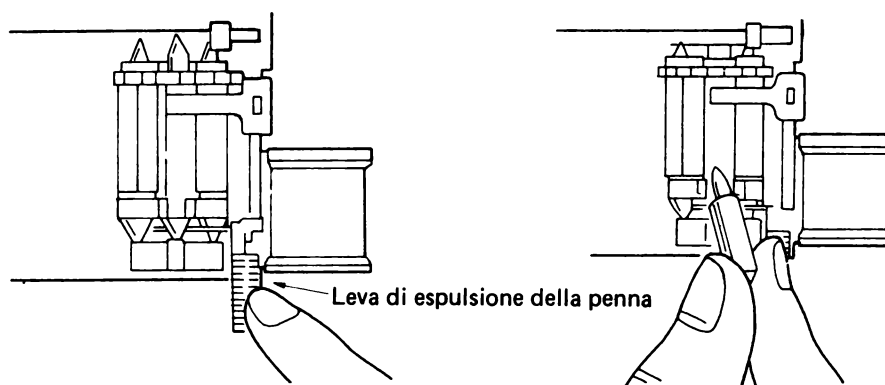
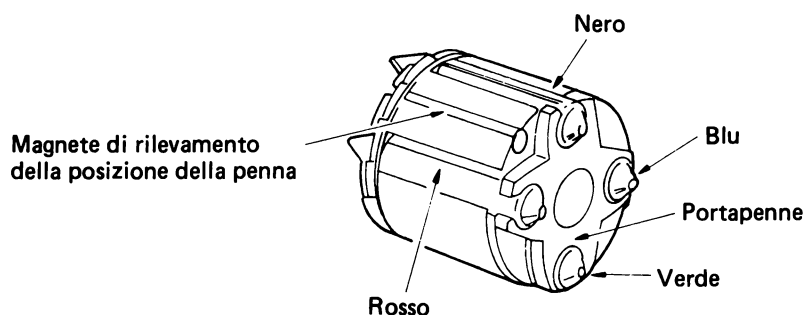
(C) Rimontate il coperchio della stampante.



(B) Premete il tasto di trascinamento della carta per fare avanzare la carta.

■ Installazione/sostituzione delle penne

1. Togliete il coperchio della stampante e premete l'interruttore PEN CHANGE con una biro o simili; questo provoca lo spostamento alla destra della stampante del portapenne per consentire la sostituzione della stessa.
2. Premete la leva di espulsione della penna per espellere la penna che è al vertice del portapenne. Durante questa operazione, appoggiate leggermente il dito sulla punta della penna mentre spingete la leva di espulsione per impedire che la penna cada all'interno della stampante.
3. Inserite una nuova penna.
4. Premete nuovamente l'interruttore PEN CHANGE per portare un'altra penna alla punta del portapenne.
5. Sostituite tutte e quattro le penne (nero, blu, verde e rosso) nello stesso modo. Quando avete finito, premete il tasto RESET per preparare la stampante a stampare con la penna nera. Eseguite il comando BASIC TEST per avere la conferma che tutti i colori vengano stampati correttamente.

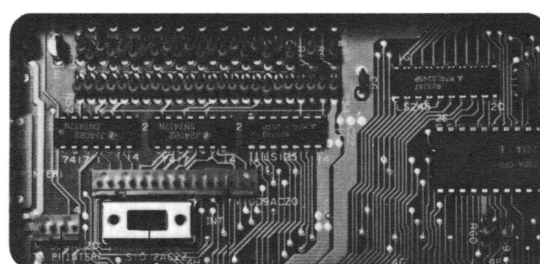


- I ricambi delle penne della stampante (penne a sfera), si possono acquistare presso il fornitore da cui la stampante è stata acquistata.

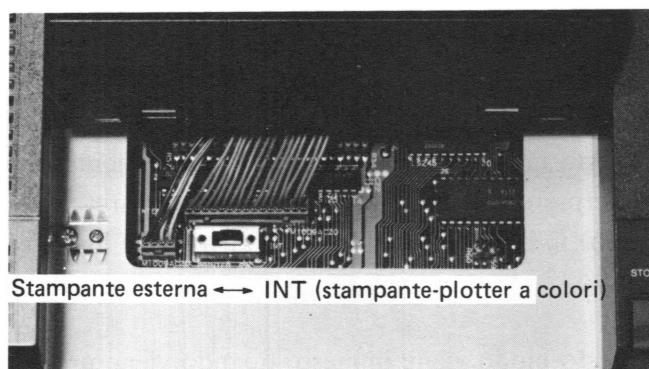
- EA-850B (nero; 4 penne)
- EA-850C (nero, blu, verde, rosso; 4 penne, una per ciascun colore)

■ Installazione della stampante a colori MZ-1P01 (per i modelli diversi dall'MZ-731)

1. Disattivate l'interruttore di alimentazione del calcolatore e disinserite il cavo di alimentazione.
2. Togliete le due viti poste al centro del pannello posteriore per togliere il coperchio del vano stampante.
3. Accertate che l'interruttore della stampante sulla piastra sia in posizione INT.
4. Inserite il connettore della stampante nel connettore corrispondente sulla piastra, quindi posizionate la stampante nel vano apposito e fissatela in posizione con le due viti. Nel corso di questa operazione, fate attenzione a non pizzicare il cavo connettore tra il registratore dati ed il calcolatore (diversamente, le viti non possono essere serrate).



Connettore stampante
Interruttore stampante
Connettore di alimentazione



Connessione della stampante-plotter a colori all'MZ-700.

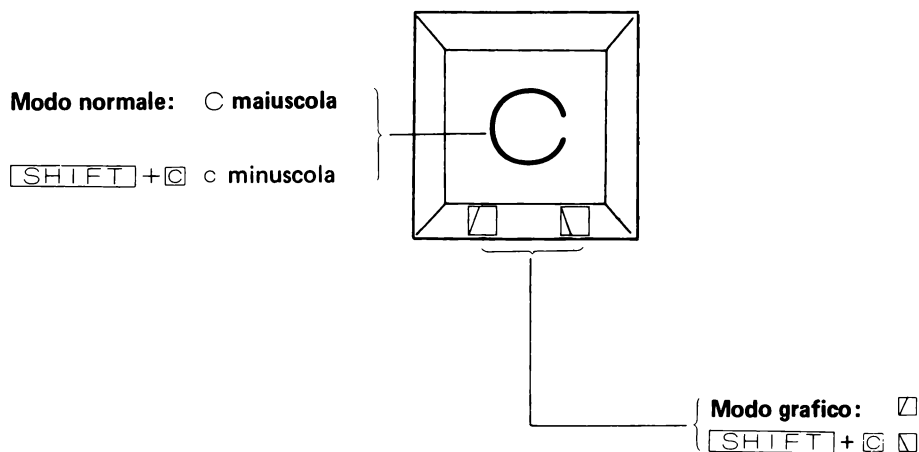
■ Connessione di una stampante esterna (MZ-80P5(K))

La stampante MZ-80P5(K) per i calcolatori della serie MZ-80K può essere collegata al connettore della stampante esterna dell'MZ-700 (vedere pagina 104) senza alcuna scheda speciale di interfaccia. Usate un cavo di connessione facoltativo per effettuare la connessione.

Quando usate una stampante esterna, l'interruttore della stampante sulla piastra deve essere messo in posizione stampante esterna. Pertanto, la stampante-plotter a colori e la stampante esterna non possono essere usate simultaneamente.

Notate che se viene eseguito un programma con una stampante esterna che include delle istruzioni di controllo della stampante-plotter a colori, verranno stampati dei caratteri senza significato (i codici di controllo della stampante-plotter).

Per esempio, i caratteri immessi mediante il tasto C nei diversi modi di immissione sono i seguenti.



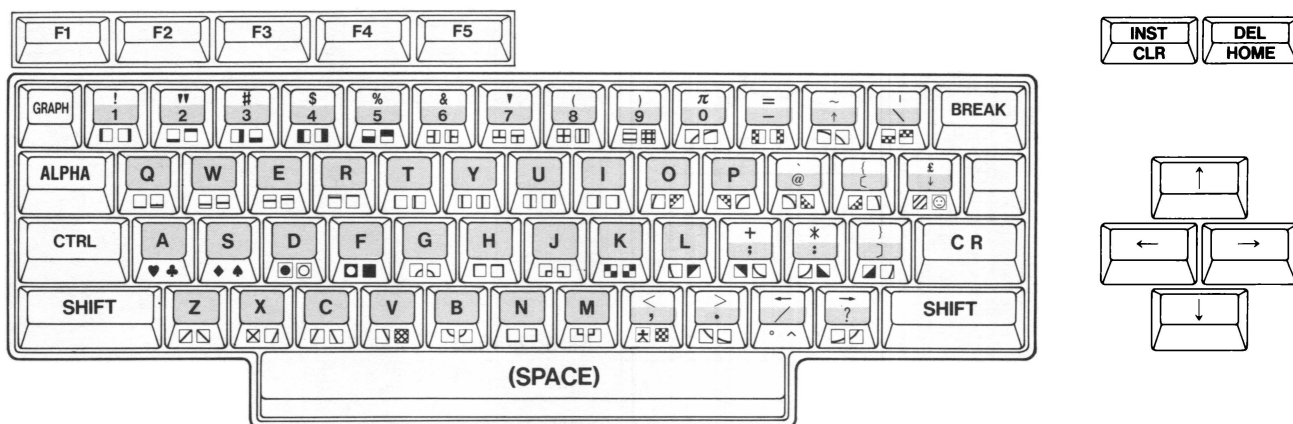
I tasti speciali vengono spiegati qui di seguito.

SHIFT	Premendo questo tasto si consente l'immissione dei caratteri nella posizione di spostamento. Per i caratteri alfabetici, i caratteri nella posizione di spostamento sono le lettere minuscole; per i tasti diversi dai tasti alfabetici, i caratteri nella posizione di spostamento sono quelli indicati nella parte superiore del tasto. Nel modo GRAPH, viene inserito il pattern grafico indicato sulla parte frontale di ogni chiave.
C R	Premendo questo tasto si immette un codice CR (ritorno a capo), che pone termine alla riga e sposta il cursore all'inizio della riga successiva.
BREAK	Premendo questo tasto si immette un codice BREAK. Premendolo insieme al tasto SHIFT si arresta l'esecuzione di un programma o il funzionamento del registratore dati.
GRAPH	Premendo questo tasto si cambia il modo di immissione da normale a grafico per l'immissione degli schemi grafici mostrati sui tasti.
ALPHA	Premendo questo tasto si cambia il modo di immissione da grafico a normale.

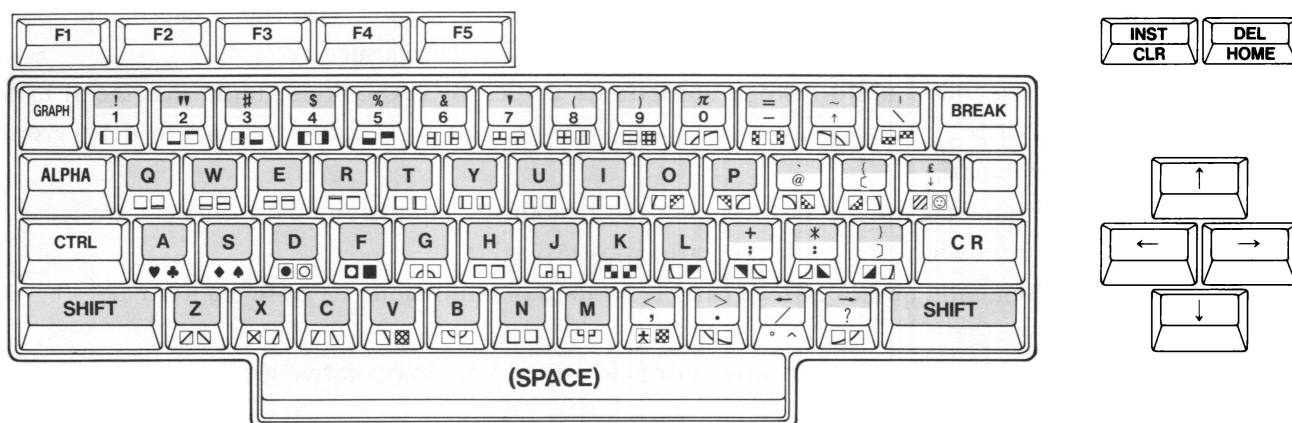
Il simbolo del cursore è ☒ nel modo operativo normale e ☐ nel modo operativo grafico.

(1) Modo operativo normale (modo operativo alfanumerico)

I caratteri immessi da ciascun tasto nel modo operativo normale sono quelli indicati nelle aree retinate della figura seguente.

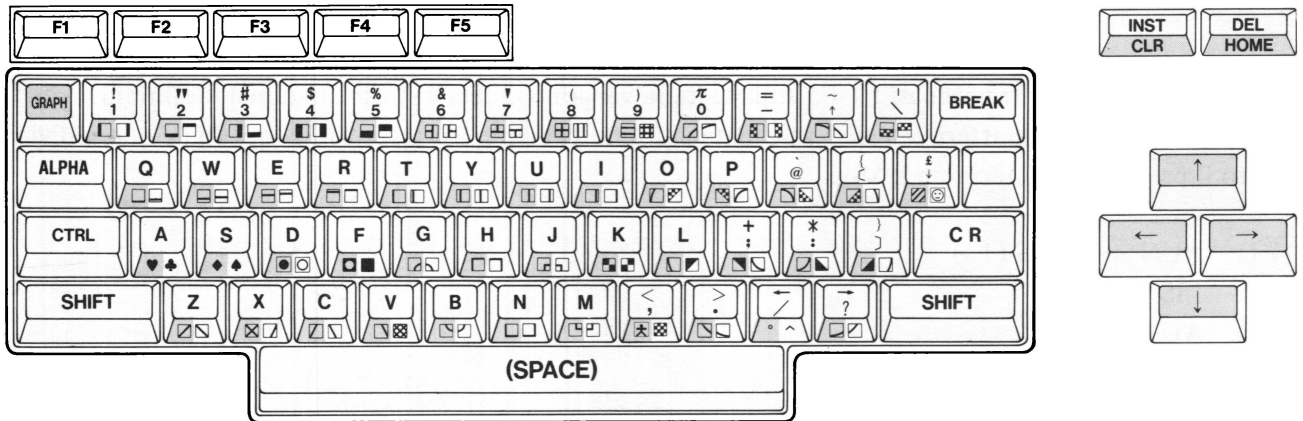


Quando si preme il tasto **SHIFT** insieme agli altri tasti, si immettono le lettere in minuscolo (o gli altri simboli indicati dalle aree retinate nella figura seguente).

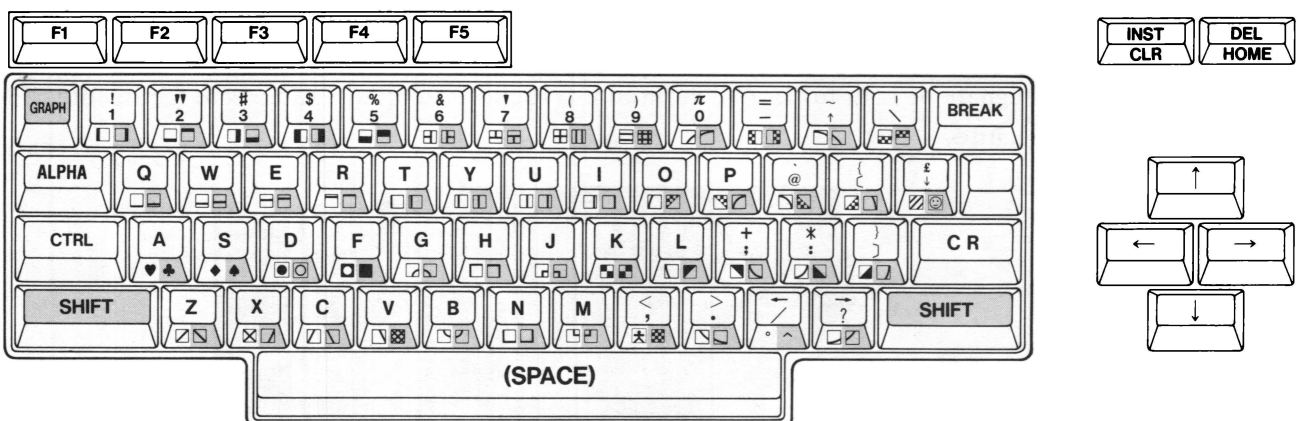




(2) Modo operativo grafico

Premando il tasto **GRAPH** si mette il calcolatore nel modo operativo di immissione grafici. I caratteri immessi da ciascun tasto nel modo operativo grafico sono quelli indicati dalla aree retinate nella figura seguente. In questo modo operativo, premendo uno qualsiasi dei tasti di controllo del cursore, il tasto **INST/CLR** o il tasto **DEL/HOME**, si immettono i simboli rispettivamente **↑** , **←** , **→** , **↓** , **C** , o **H** .



Quando vengono premuti insieme con il tasto SHIFT altri tasti, vengono inseriti simboli indicati dalle aree del video qui sotto.



Il simbolo del cursore è  nel modo operativo grafico. Per tornare al modo operativo normale, premete il tasto  .

3.5.2 Tasti funzionali definibili



Tasti funzionali definibili

I cinque tasti blu contrassegnati F1-F5 sulla tastiera vengono chiamati tasti funzionali definibili.

Alcune stringhe di caratteri vengono assegnate automaticamente a questi tasti come segue quando viene attivato il programma interprete BASIC.

F1: " RUN " + CHR\$ (13)

F2: " LIST "

F3: " AUTO "

F4: " RENUM "

F5: " COLOR "

SHIFT + F1: " CHR\$ ("

SHIFT + F2: " DEF KEY ("

SHIFT + F3: " CONT "

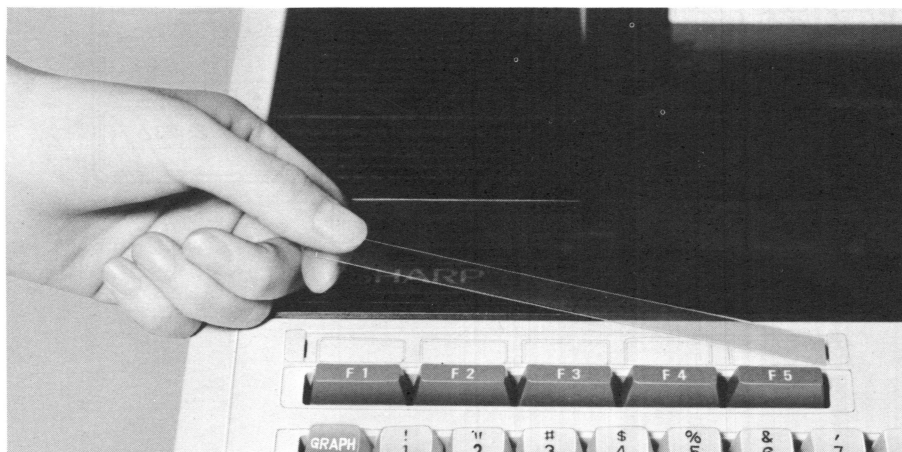
SHIFT + F4: " SAVE "

SHIFT + F5: " LOAD "

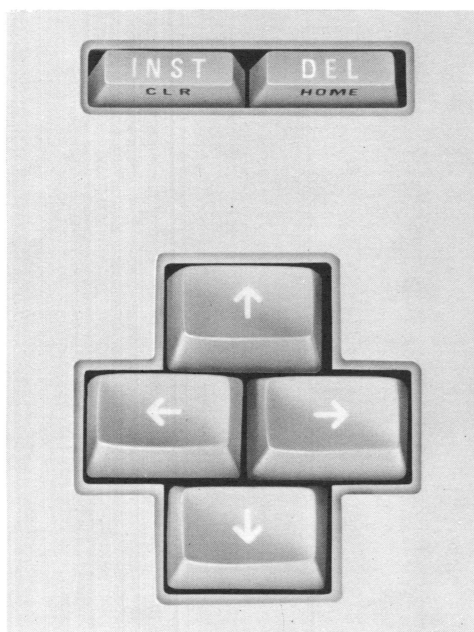
Quando si preme uno di questi tasti, viene immessa la stringa di caratteri assegnata a tale tasto; pertanto, le istruzioni che vengono usate frequentemente possono essere immesse premendo un solo tasto. La stringa di caratteri assegnata a ciascuno dei tasti funzionali definibili può essere cambiata dall'istruzione DEF KEY. (vedere pagina 55, istruzione DEF KEY).

■ Etichette dei tasti funzionali definibili

E' possibile applicare sotto alla copertura trasparente posta sopra ai tasti, delle etichette che indicano le stringhe di caratteri assegnate ai tasti funzionali definibili. Il foglio trasparente può essere facilmente tolto come illustrato qui di seguito.





3.5.3 Tasti di controllo del cursore e tasti di inserimento e di cancellazione



Tasti di controllo del cursore e tasti di inserimento e di cancellazione

I tasti di controllo del cursore sono i quattro tasti gialli alla destra della tastiera che sono contrassegnati con delle frecce.

Premendo questi tasti si sposta il cursore di una posizione nella direzione indicata dalla freccia. Questi tasti vengono usati quando si effettua la revisione dei programmi.



I tasti  e  hanno le funzioni seguenti.





Inserisce uno spazio nella posizione del cursore e sposta tutti i caratteri successivi a destra di una posizione. INST: inserimento.



Cancella il carattere alla sinistra del cursore e sposta tutti i caratteri successivi a destra di una posizione. DEL: cancellare.

 + 

Annulla l'intero schermo e riporta il cursore nell'angolo superiore di sinistra dello schermo. Questo tasto non interessa il programma in memoria. CLR: annullamento.

 + 

Riporta il cursore all'angolo superiore di sinistra dello schermo (non influenza i caratteri visualizzati).

Vedere pagine 16 e 17.

Hardware (Componenti fisiche della Macchina)

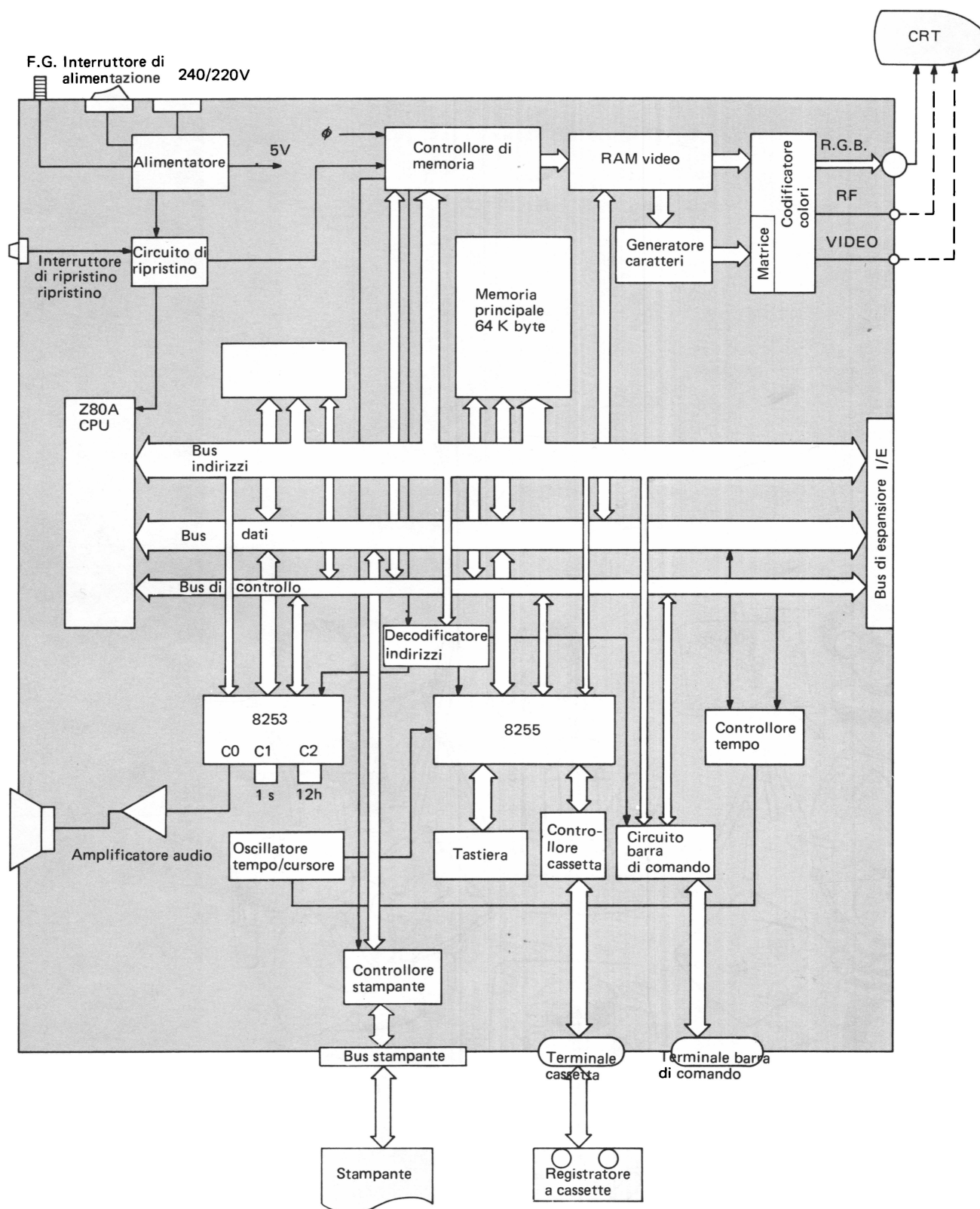
Capitolo 4



Nota: Il contenuto di questo capitolo viene dato a solo scopo consultativo, e la Sharp non si assume alcuna responsabilità di rispondere ad eventuali domande sul suo contenuto.

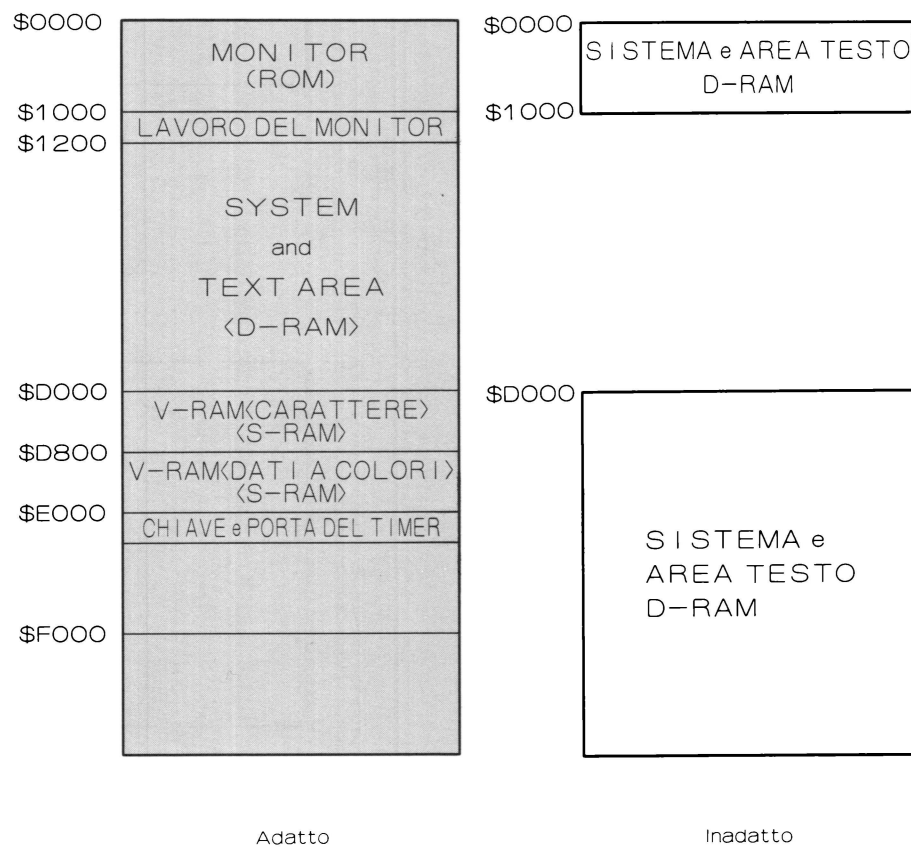
4.1 Diagramma del sistema MZ-700

La figura seguente illustra la configurazione di sistema dei calcolatori della serie MZ-700.



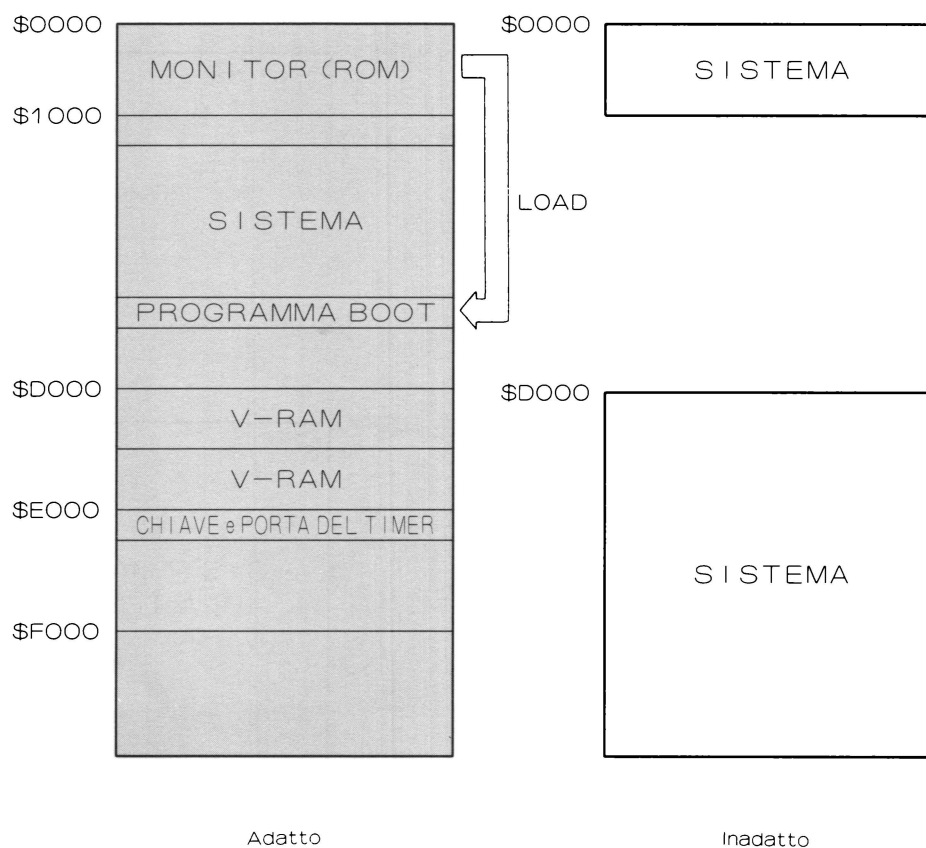
4.2 Configurazione di memoria

4.2.1 Mappa della memoria all'accensione (modo 80K)



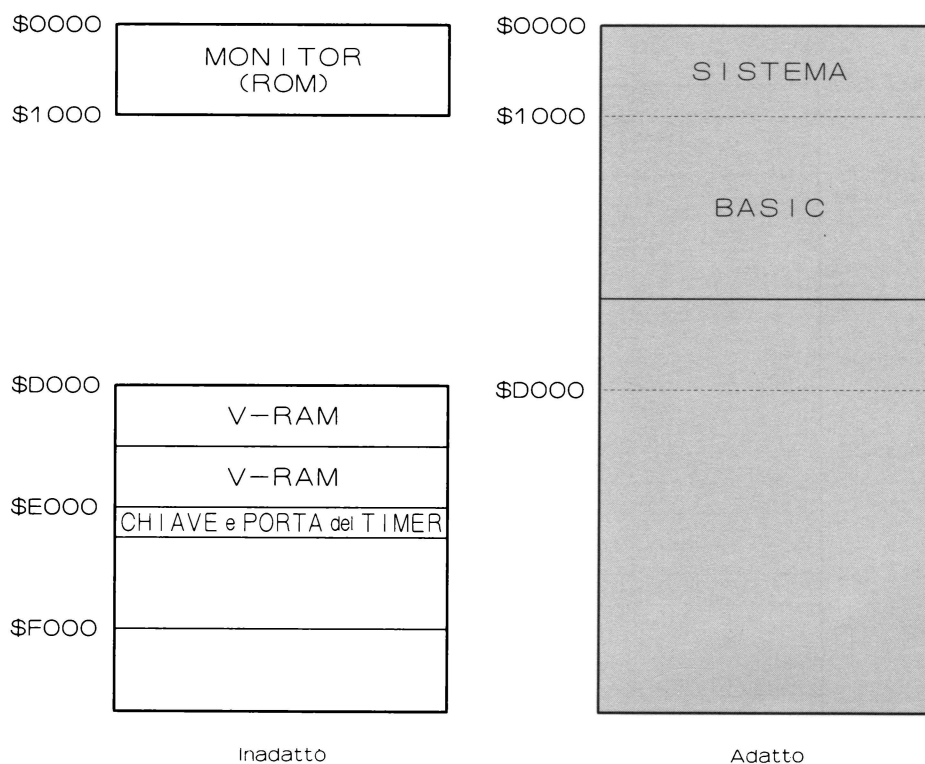
- La mappa della memoria è come quella precedentemente mostrata, immediatamente dopo che la macchina è stata accesa. (Il contenuto dell'area V-RAM da \$D000 a \$DFFF non è uguale a quello dell'MZ-80K).
- Il punto di immissione della ROM monitor è uguale a quello dell'MZ-80K.

4.2.2 Mappa della memoria in fase di caricamento del programma di sistema (BASIC)



- Quando viene immesso il comando **LOAD monitor**, il programma di lancio iniziale (bootstrap) viene caricato nell'area RAM del sistema dalla ROM ed il controllo viene trasferito a tale programma.
- Il programma di lancio iniziale carica il programma interprete BASIC nell'area del sistema dalla cassetta.

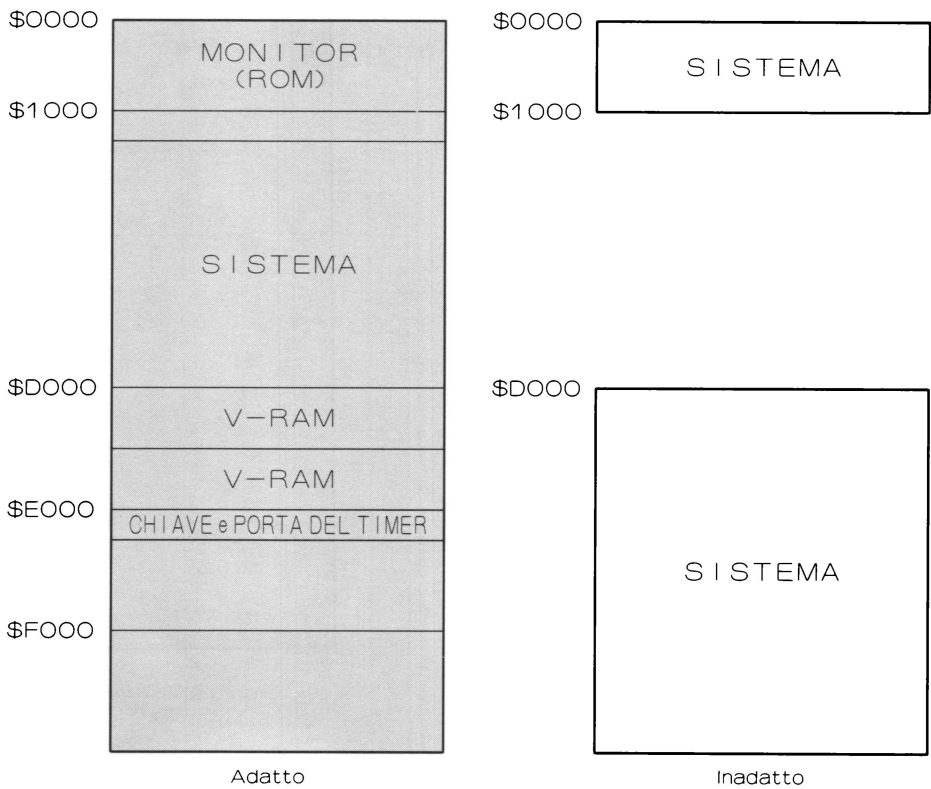
4.2.3 Mappa della memoria dopo che è stato caricato il programma interprete BASIC (modo MZ-700)



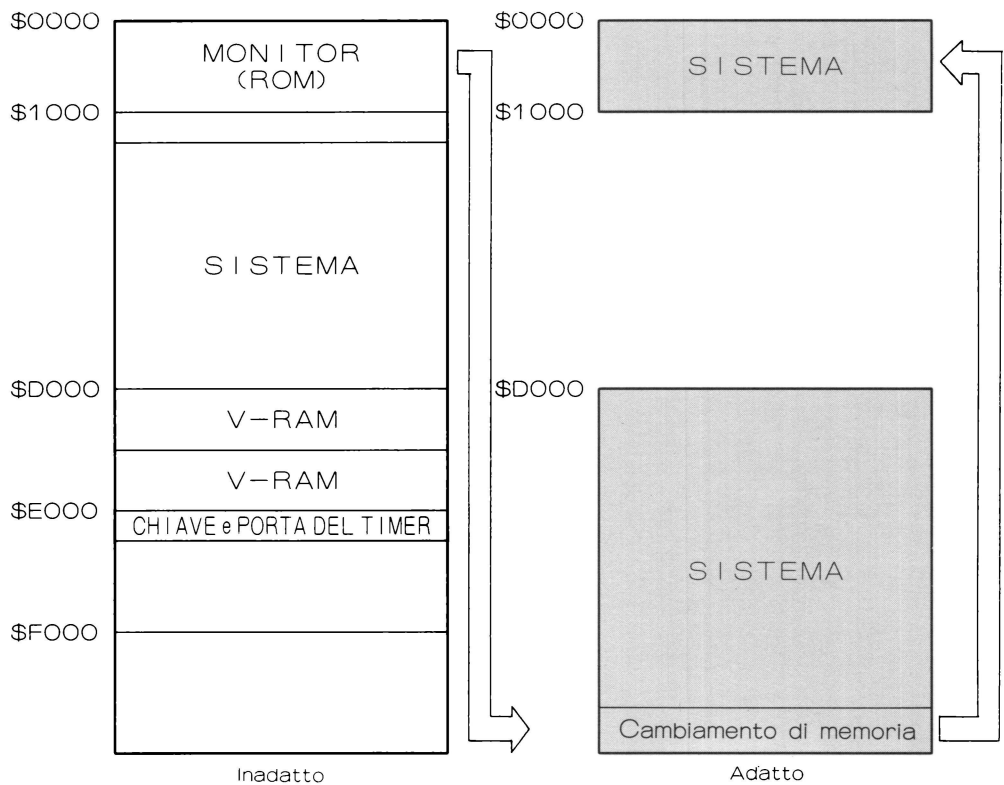
- La mappa della memoria è come mostrato qui di sopra dopo che è stato caricato il programma interprete BASIC.
- Viene eseguita la commutazione di banco per accedere alla V-RAM o all'area KEY e TIMER PORT.

4.2.4 Mappa della memoria dopo il ripristino manuale

La mappa della memoria è come mostrato qui di seguito dopo che è stato premuto l'interruttore di ripristino sul pannello posteriore.



Dopo aver premuto l'interruttore di ripristino unitamente al tasto **CTRL** , la mappa della memoria è come illustrato qui di seguito.



- Quando l'interruttore di ripristino viene premuto insieme al tasto **CTRL** , alla RAM vengono assegnati gli indirizzi \$0000-\$0FFF e da \$D000 a \$FFFF.
- Quando viene immesso il comando # dopo che è stato premuto l'interruttore di ripristino, il calcolatore funziona nello stesso modo di quando è stato premuto l'interruttore di ripristino insieme al tasto **CTRL** .

4.2.5 Commutazione banchi

- a) I blocchi di memoria possono essere scelti emettendo i dati alle porte I/E come illustrato qui di seguito.

ALTERNAMENTI

PORTA I/O	\$0000 ~ \$0FFF	\$D000 ~ \$FFFF
\$ E0	AREA SISTEMA (D-RAM)	
\$ E1		AREA SISTEMA (D-RAM)
\$ E2	MONITOR (ROM)	
\$ E3		V-RAM, CHIAVE, TIMER
\$ E4	MONITOR (ROM)	V-RAM, CHIAVE, TIMER
\$ E5		Inibisce
\$ E6		Ritorno alla condizione iniziale, dove viene inibito da \$E5.

Nota: L'emissione dei dati alla porta I/E \$E4 svolge la stessa funzione ottenuta con la pressione dell'interruttore di ripristino.

- b) Esempi:

OUT (\$E0), A

Assegna alla RAM gli indirizzi da \$0000 a \$0FFFF, ma non cambia l'indirizzo di esecuzione. Il contenuto della variabile A non ha effetto sul risultato.

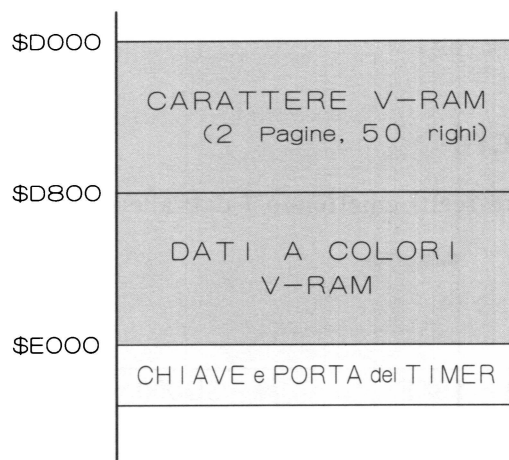
OUT (\$E4), A

Inizializza la memoria allo stato in atto immediatamente dopo a che è stata attivata l'alimentazione.

Nota: Poiché il contatore di programma non viene spostato dall'istruzione OUT, si deve fare attenzione quando si commutano i blocchi di memoria se il contatore di programma si trova nell'area da \$0000 a \$0FFFF o da \$D000 a \$FFFF.

4.2.6 Mappa della memoria quando si accede alla V-RAM

i) V-RAM (RAM video) Mappa della memoria



ii) Corrispondenza tra l'indirizzo V-RAM e la posizione sullo schermo

L'MZ-700 ha un'area V-RAM di 2K byte, ma soltanto 1K byte di tale area può essere visualizzato sullo schermo nello stesso momento. L'area visualizzata può essere cambiata facendo scorrere lo schermo.

a) Area visualizzata immediatamente dopo il ripristino (o l'accensione):

D000
1

← Indirizzo
 ← Numero byte

	1	2	3	39	40	
1	D000* 1	D001 2	D002 3	D026 39	D027 40	<div style="border: 1px solid black; width: 15px; height: 15px; display: inline-block;"></div> Colonna
2	D028 41	D029 42	D02A 43	D04E 79	D04F 80	
25	D3C0 961	D3C1 962	D3C2 963	D3E6 999	D3E7 1000	

Riga

b) Area visualizzata dopo che lo schermo è stato fatto scorrere verso l'alto di una riga dalla fine della V-RAM:

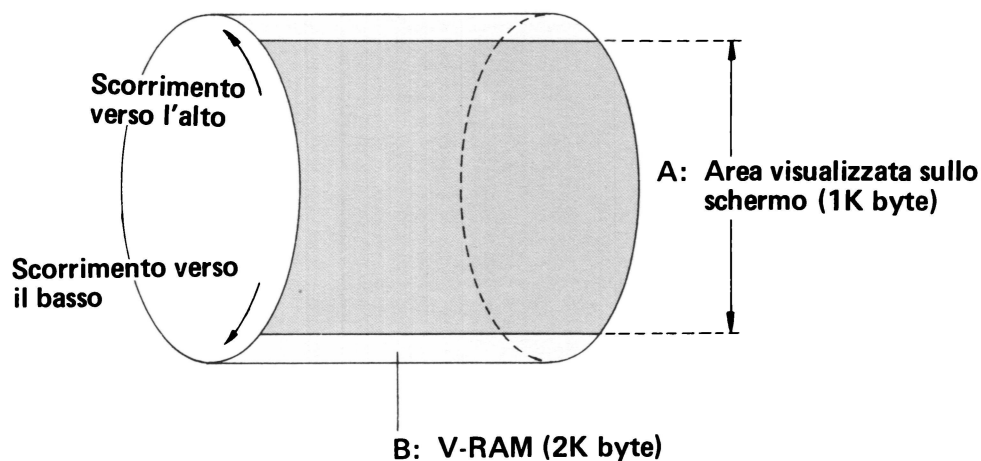
	1	2	3					39	40
1	D000 1041	D001 1042	D002 1043					D026 1079	D027 1080
2	D028 1081	D029 1082	D02A 1083					D04E 1119	D04F 1120
24	D398 1961	D399 1962	D39A 1963					D3BE 1999	D3BF 2000
25	D3C0 1	D3C1 2	D3C2 3					D3E6 39	D3E7 40

Nota: La riga costituita dai byte 1-40 viene indirizzata ciclicamente a quella costituita dai byte 1961-2000 come illustrato in precedenza.

iii) Scorrimento verso l'alto e verso il basso

a) Lo schermo viene fatto scorrere verso l'alto premendo insieme il tasto **SHIFT** e il tasto **↑**, e viene fatto scorrere verso il basso premendo insieme il tasto **SHIFT** ed il tasto **↓**.

b) Scorrimento verso l'alto e verso il basso



- Durante lo scorrimento, l'area visualizzata sullo schermo si sposta attraverso l'area V-RAM di 2K byte come mostrato in precedenza.
- La fine dell'area V-RAM viene indirizzata ciclicamente all'inizio della V-RAM come illustrato in precedenza.
- Durante lo scorrimento il cursore non si sposta sullo schermo.

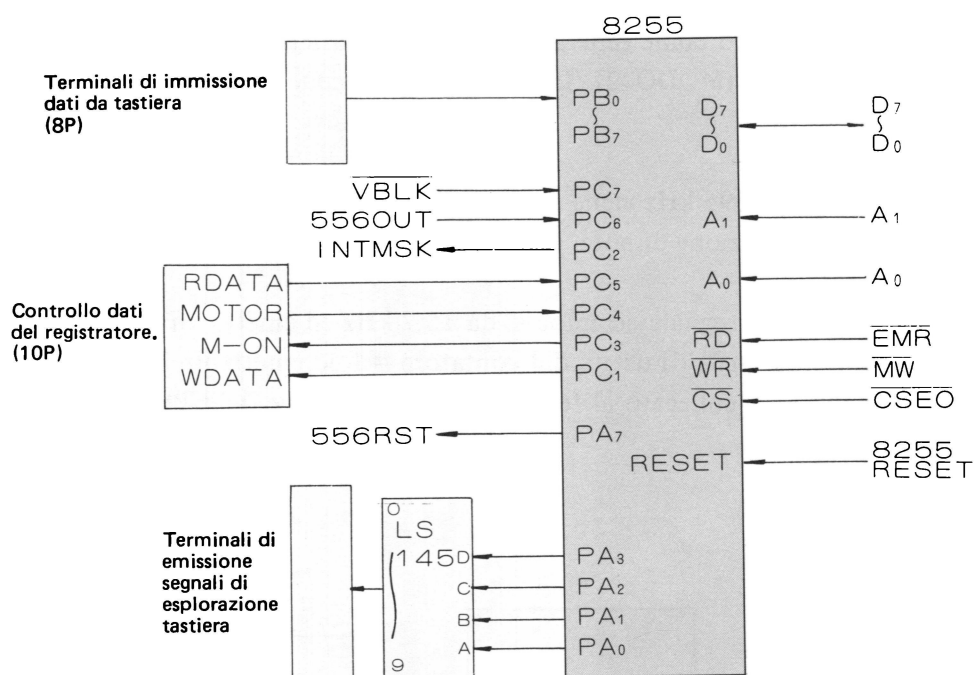
4.3 I/E nella mappa della memoria (\$E000-\$E008)

Gli indirizzi \$E000-\$E008 vengono assegnati all'interfaccia periferica programmabile 8255, al temporizzatore di intervalli programmabile 8253 e agli altri circuiti integrati di controllo di I/E in modo che è possibile accedere alle diverse unità di I/E (incluse le funzioni musicali usando il contatore # 0 dell'8253) nello stesso modo della memoria. La tabella di I/E nella mappa della memoria viene illustrata qui di seguito.

Indirizzo memoria CPU	Controllore	Funzionamento
\$E000 \$E001 \$E002 \$E003	8255	P _A : Emissione P _B : Immissione P _C : Controllo immissione ed emissione mediante impostazione bit Controllo modo operativo
\$E004 \$E005 \$E006 \$E007	8255	C ₀ : Modo 3 (generatore di onda quadra) C ₁ : Modo 2 (generatore di cadenza) C ₂ : Modo 0 (contatore terminale) Controllo modo
\$E008	LS367, etc.	Tempo, barra di comando ed immissione HBLNK

4.3.1 Sistema segnali dell'8255

L'8255 emette segnali di esplorazione tastiera, immette dati da tastiera, e controlla la temporizzazione del registratore a cassette e del lampeggiamento del cursore.



Porta	Terminale	I/O	Stato attivo	Descrizione del controllo	Nome del segnale
PA (\$E000)	PA ₀	OUT	H	Segnali di esplorazione tastiera	556 RST
	PA ₁		H		
	PA ₂		H		
	PA ₃		H		
	PA ₇		L	Ripristina il temporizzatore di lampeggiamento del cursore.	
PB (\$E001)	PB ₀	IN	L	Segnali di immissione dati da tastiera	
	PB ₁		L		
	PB ₂		L		
	PB ₃		L		
	PB ₄		L		
	PB ₅		L		
	PB ₆		L		
	PB ₇		L		
PC* (\$E002)	PC ₁	OUT	—	Scrittura dati su cassetta	WDATA
	PC ₂	OUT	L	Inibisce le interruzioni dell'orologio	INTMSK
	PC ₃	OUT		Segnale di azionamento motore	M-ON
	PC ₄	IN	H	Indica che il motore è attivato	MOTOR
	PC ₅	IN	—	Letture dati da cassetta	RDATA
	PC ₆	IN	—	Segnale di immissione temporizzatore lampeggiamento cursore	556 OUT
	PC ₇	IN	—	Segnale di soppressione in verticale	VLK

* Ogni bit di dati emesso può essere fissato o ripristinato indipendentemente.

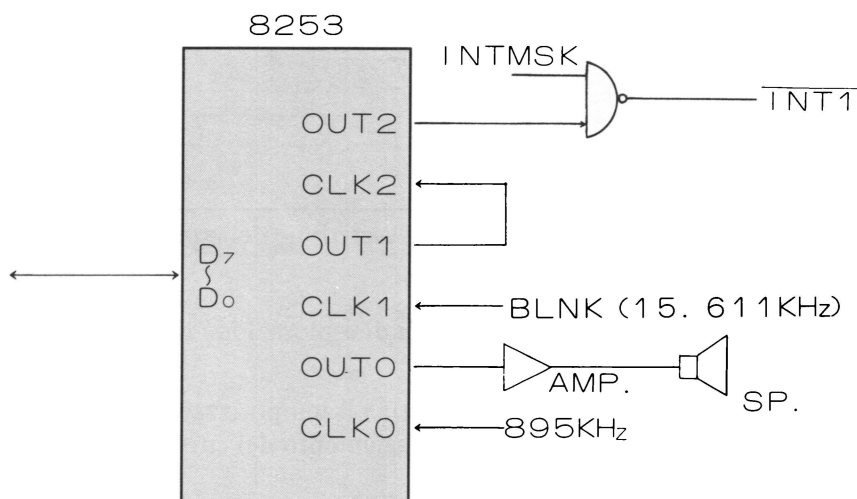
4.3.2 Sistema di segnali dell'8253

L'8253 include tre contatori # 0, # 1, e # 2. Il contatore # 0 viene usato per la generazione dei suoni mentre i contatori # 1 e # 2 vengono usati per l'orologio incorporato.

Il contatore # 0 viene usato come generatore di onda quadra (MODO 3) ed il contatore # 1 viene usato come generatore di cadenza (MODO 2). Il contatore # 2 viene usato per l'interruzione del conteggio a terminale (MODO 0).

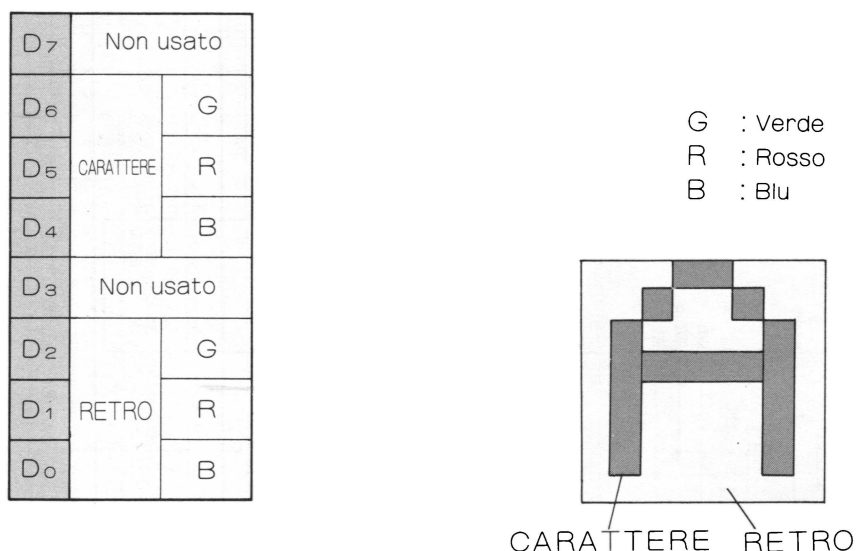
Un segnale ad impulsi da 895 kHz viene applicato al contatore # 0, il quale divide la frequenza al valore specificato secondo l'informazione di nota. Questo segnale diviso viene emesso al generatore acustico.

Il contatore # 1 conta un segnale ad impulsi da 15,7 kHz ed emette un impulso a OUT1 ogni secondo. Il contatore # 2 conta il segnale di uscita dal contatore # 1 ed emette un impulso di alto livello ad OUT2 ogni 12 ore. Poiché OUT2 è collegato al terminale di interruzione del CPU, il CPU elabora l'interruzione ogni 12 ore.

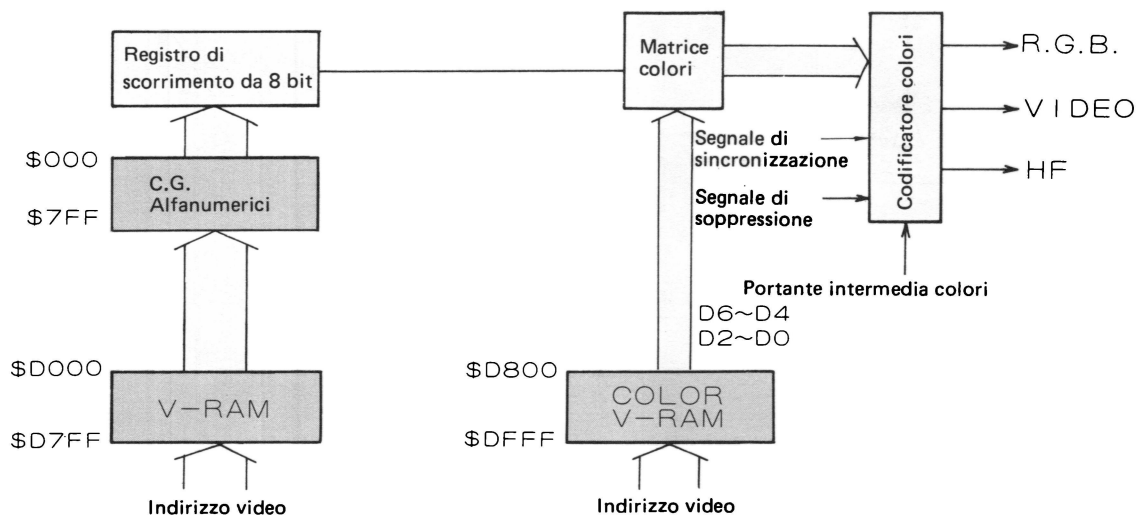


4.4 Sistema di segnali della V-RAM colori

Le informazioni sui colori dell'MZ-700 sono controllate in unità di caratteri; cioè viene assegnata una tabella di informazioni sui colori da 1 byte ad ogni carattere visualizzato sullo schermo. La tabella di informazione sui colori viene illustrata nella figura seguente.



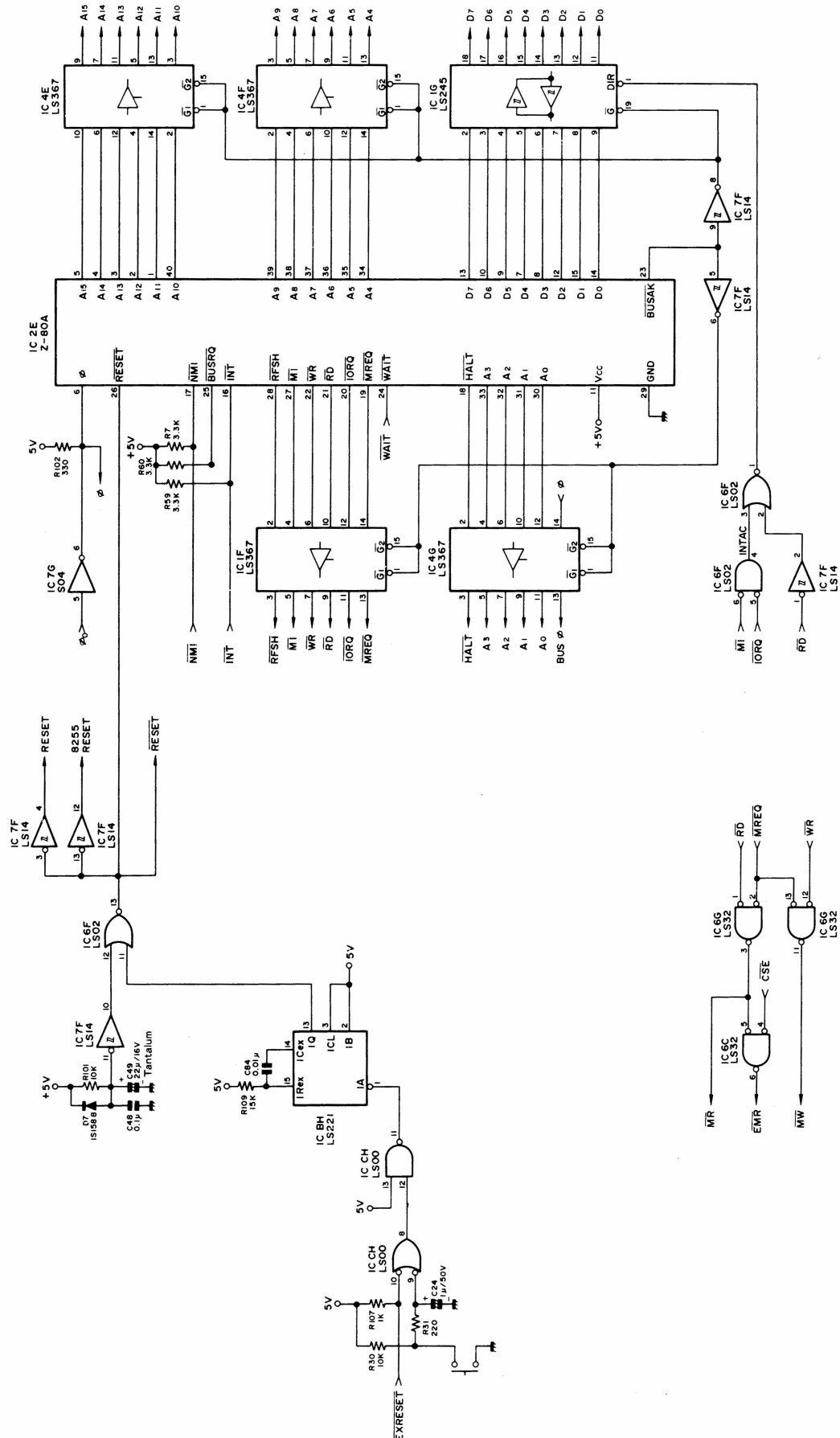
Si accede alle tabelle di informazioni sui colori come segue.



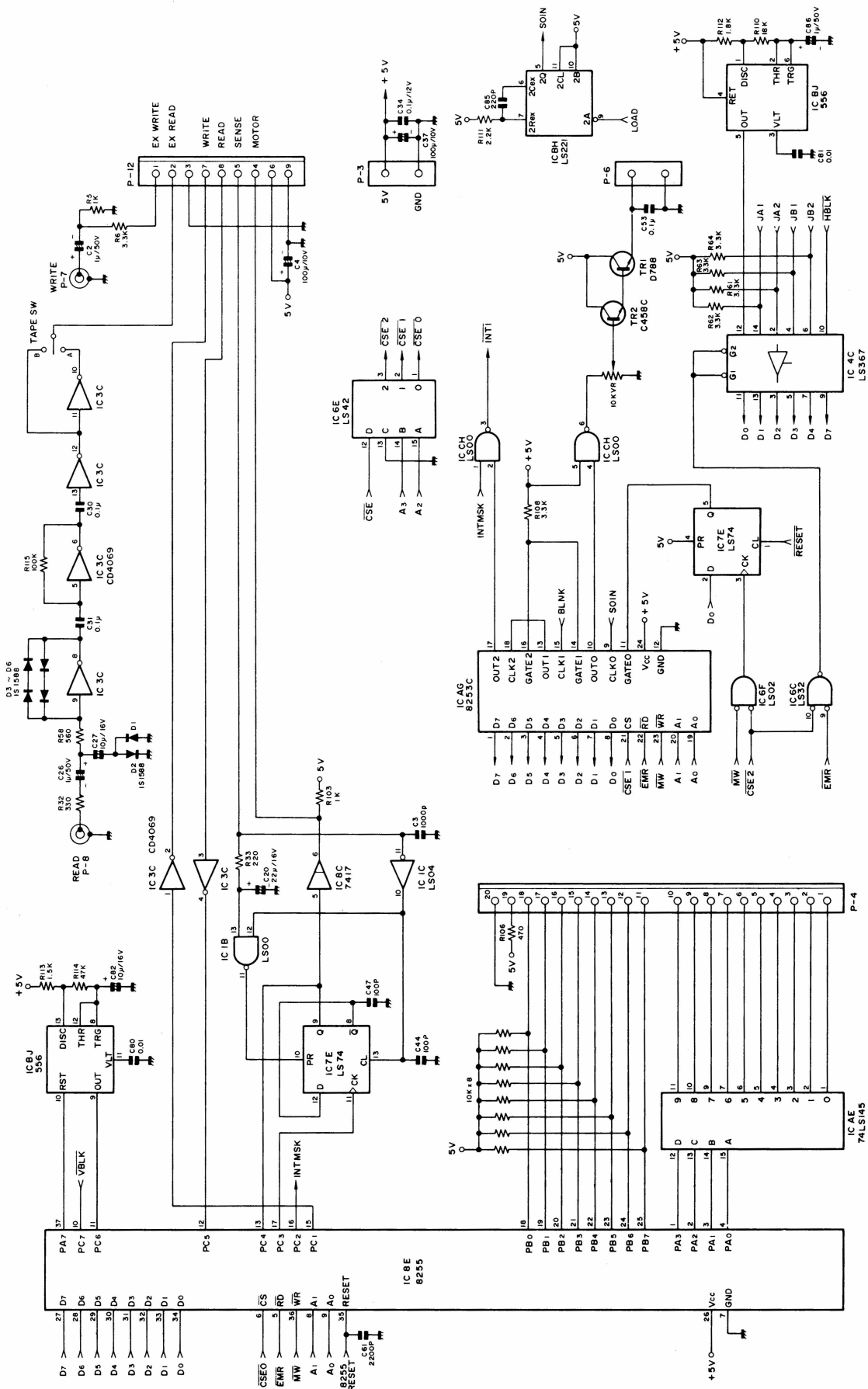
I caratteri visualizzati vengono memorizzati agli indirizzi \$D000-\$D7FF della V-RAM, e le tabelle di informazioni sui colori vengono memorizzate agli indirizzi \$D800-\$DFFF della V-RAM.

4.5 Schemi circuitali dell'MZ-700

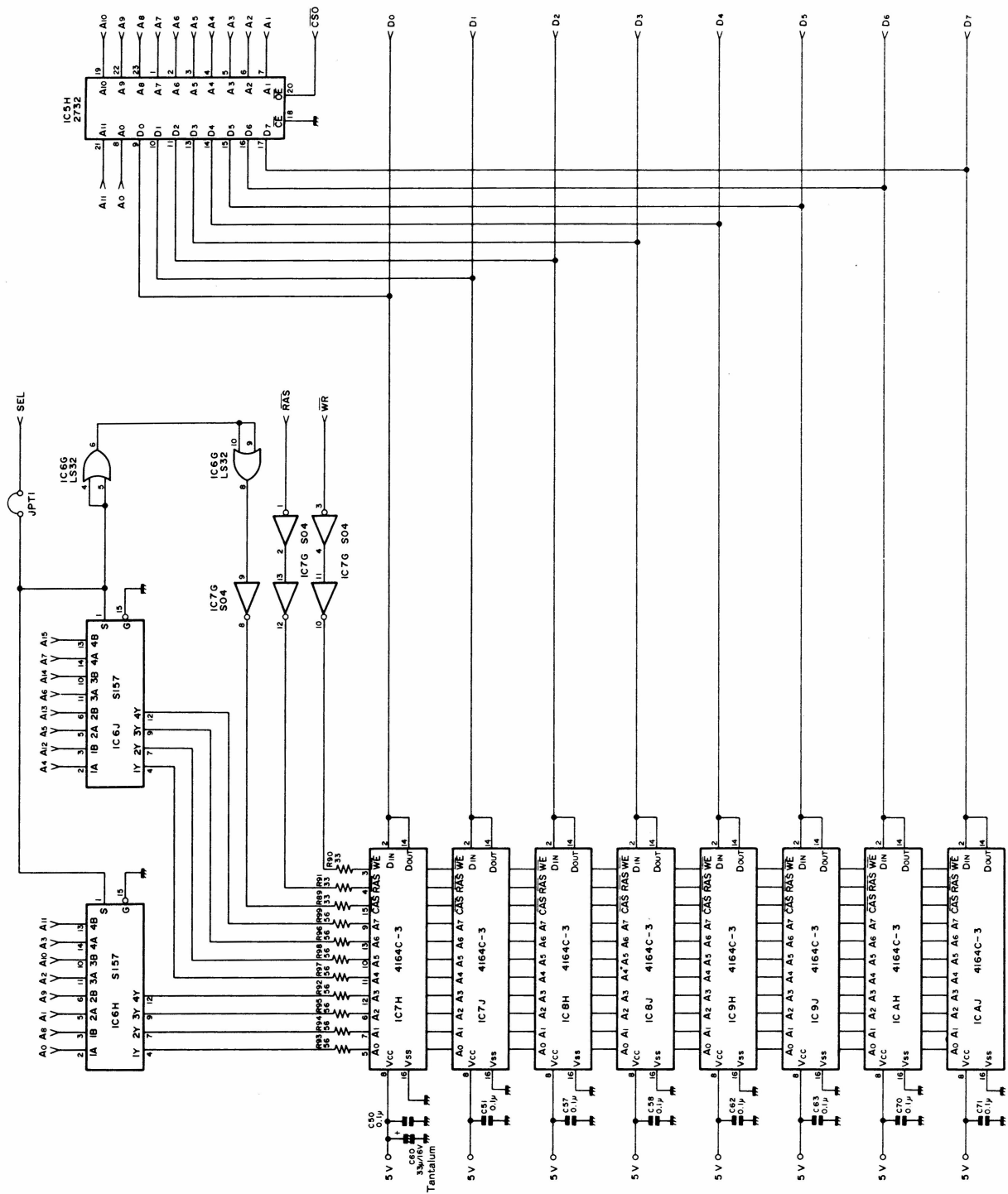
[Circuito piastra CPU (1)]



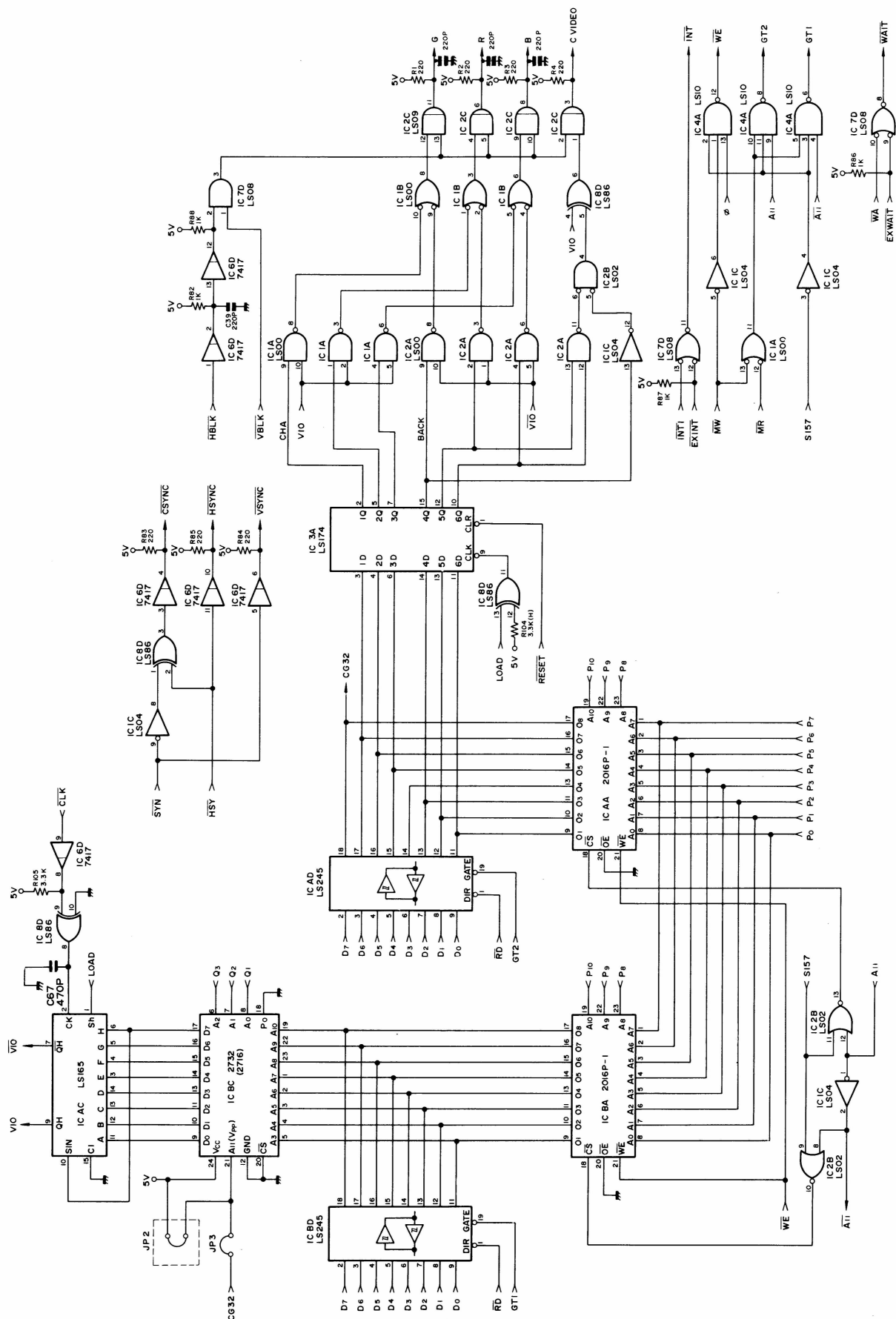
[Circuito piastra CPU (2)]



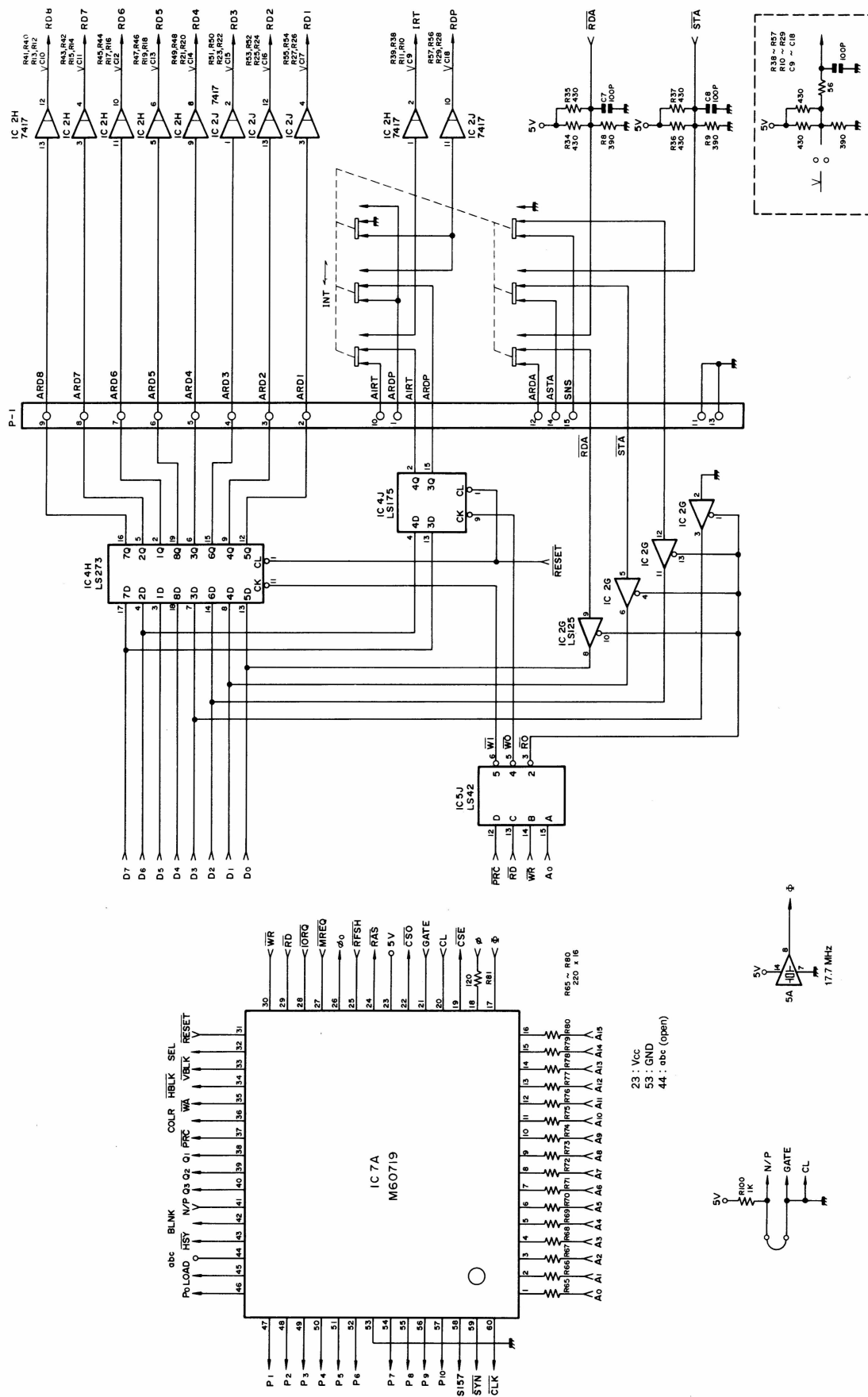
[Circuito piastra CPU (3)]



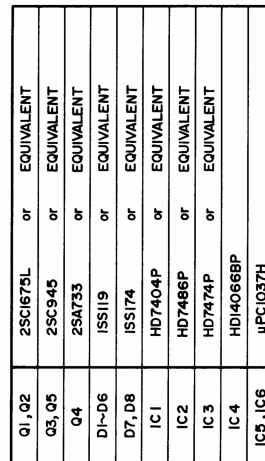
[Circuito piastra CPU (4)]



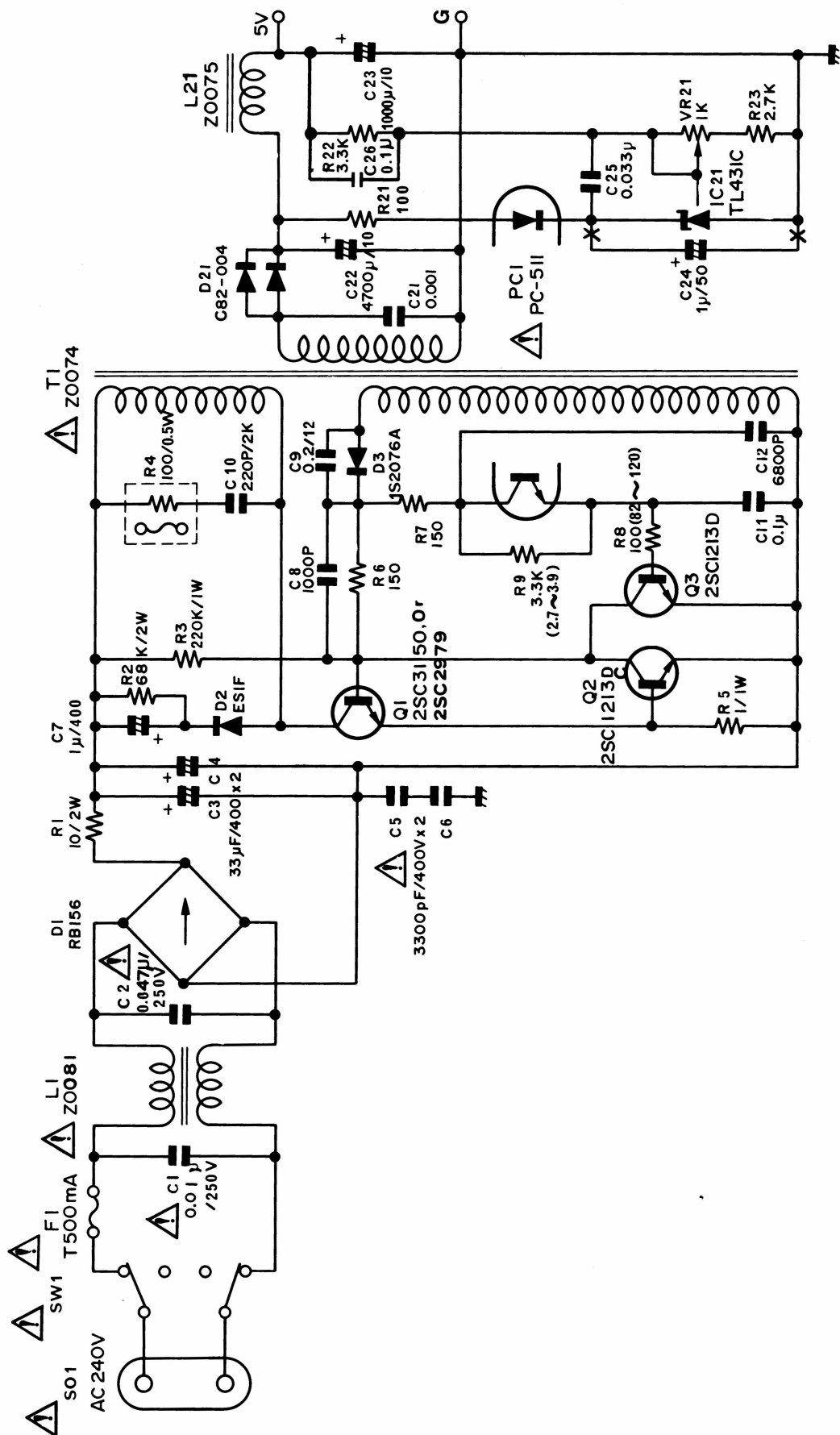
[Circuito piastra CPU (5)]



[Circuito codificatore colori)]



[Alimentatore]

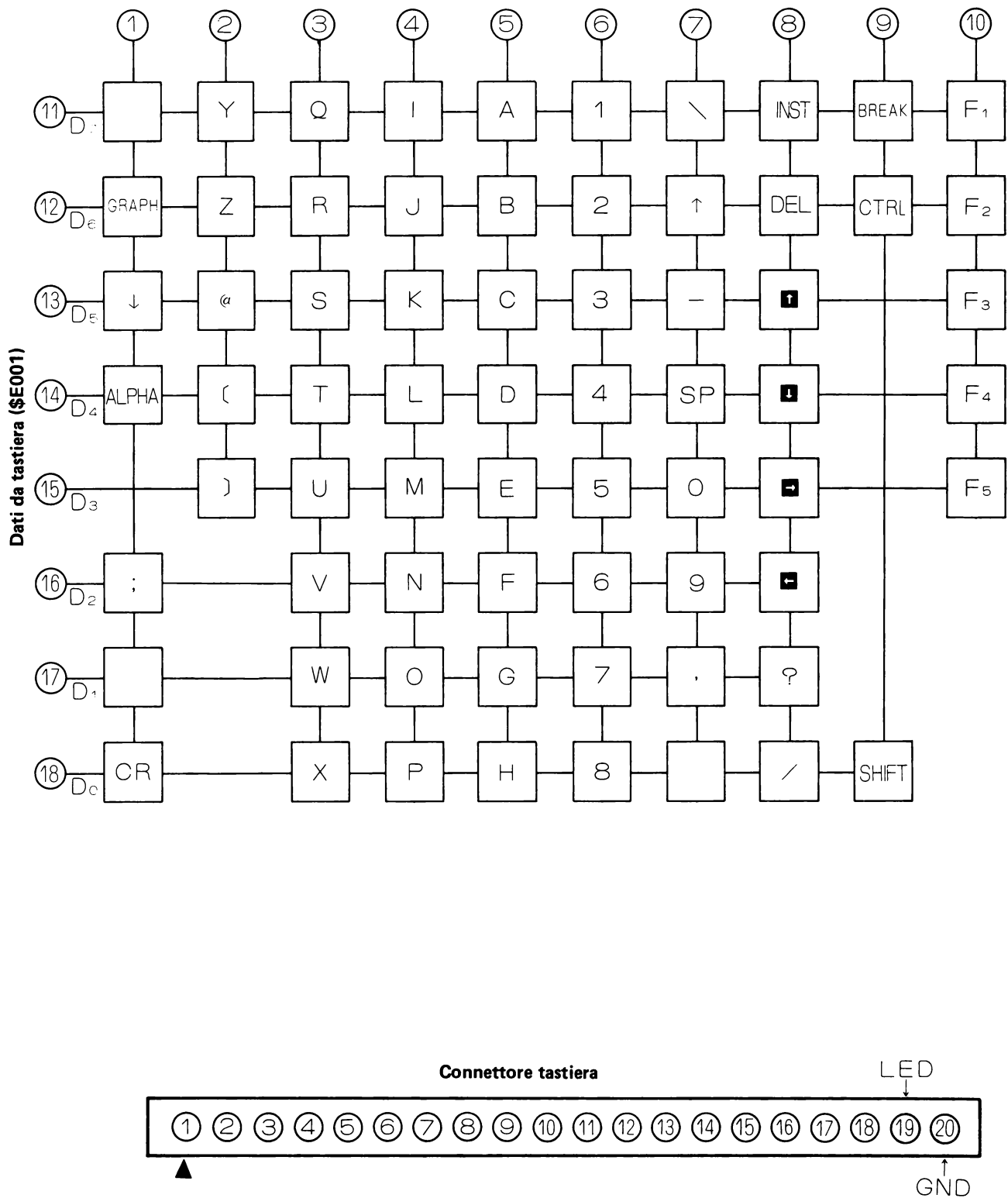


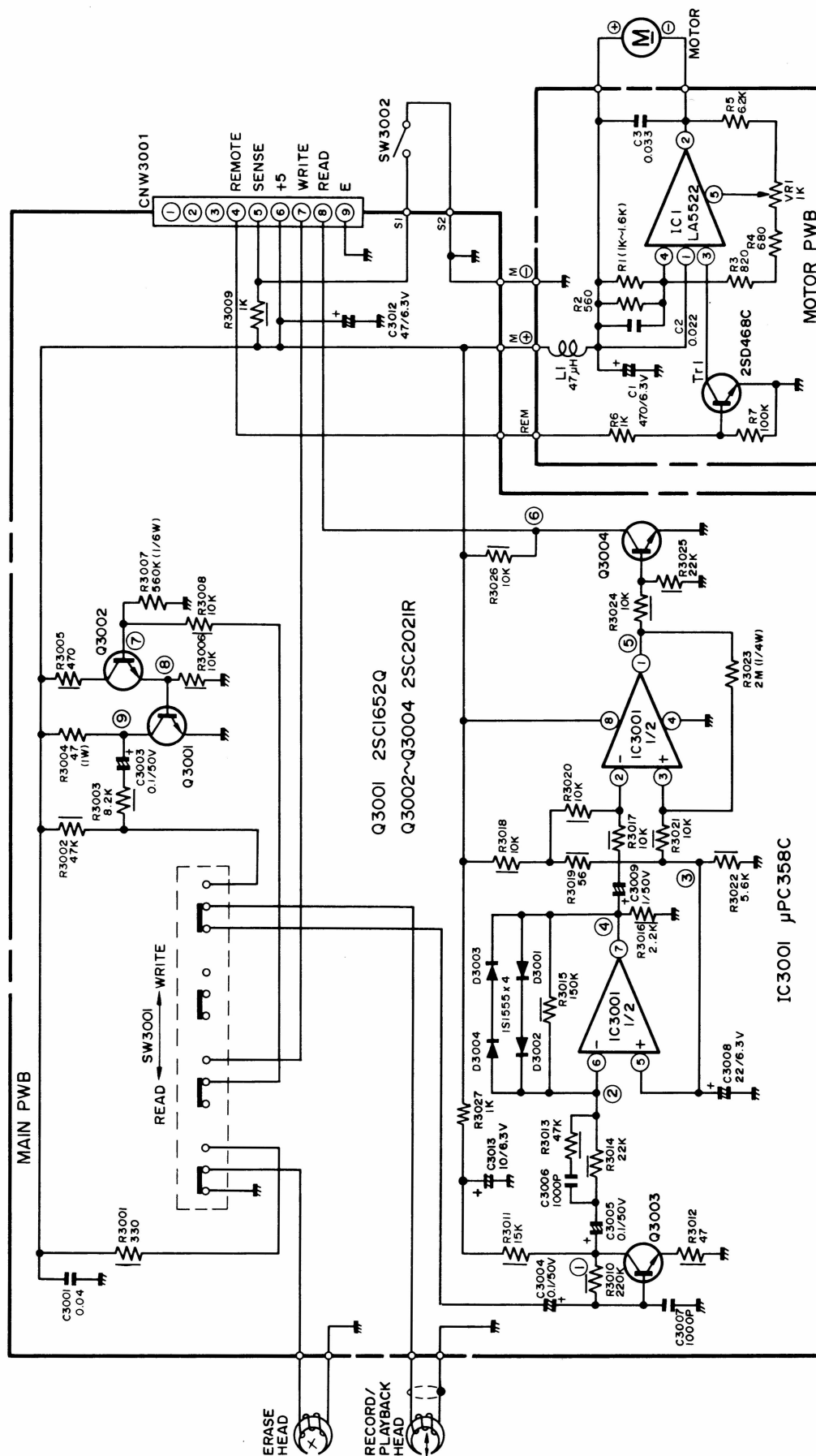
[Configurazione terminale piastra CPU]

P-1	1	ARDP
	2	ARD1
	3	ARD2
	4	ARD3
	5	ARD4
P-5	1	+5V
	2	+5V
	3	GND
	4	GND
P-10	1	RDP
	3	RD1
	5	RD2
	7	RD3
	9	RD4
P-11	15	RD7
	17	RD8
	19	IRT
	21	RD
	23	STA
P-13	1	5V
	2	VBLK
	3	JA1
	4	JA2
	5	GND
P-14	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-9	1	GND
	2	C SYNC
	3	C VIDEO
	4	H SYNC
	5	V SYNC
P-12	6	GND
	7	+5V
	8	G
	9	B
	10	R
P-15	11	COLR
	12	GND
P-16	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-17	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-18	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-19	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-20	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-21	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-22	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-23	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-24	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-25	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-26	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-27	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-28	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-29	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-30	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-31	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-32	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-33	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-34	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-35	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-36	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-37	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-38	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-39	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-40	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-41	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-42	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-43	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-44	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-45	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-46	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-47	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-48	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-49	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-50	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-51	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-52	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-53	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-54	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-55	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-56	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-57	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-58	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-59	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-60	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-61	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-62	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-63	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-64	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-65	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-66	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-67	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-68	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-69	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-70	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-71	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-72	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-73	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-74	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-75	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-76	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-77	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-78	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-79	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-80	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-81	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-82	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-83	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-84	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-85	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-86	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-87	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-88	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-89	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-90	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-91	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-92	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-93	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-94	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-95	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-96	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-97	1	5V
	2	VBLK
	3	JB1
	4	JB2
	5	GND
P-98	1	5V
	2	VBLK</

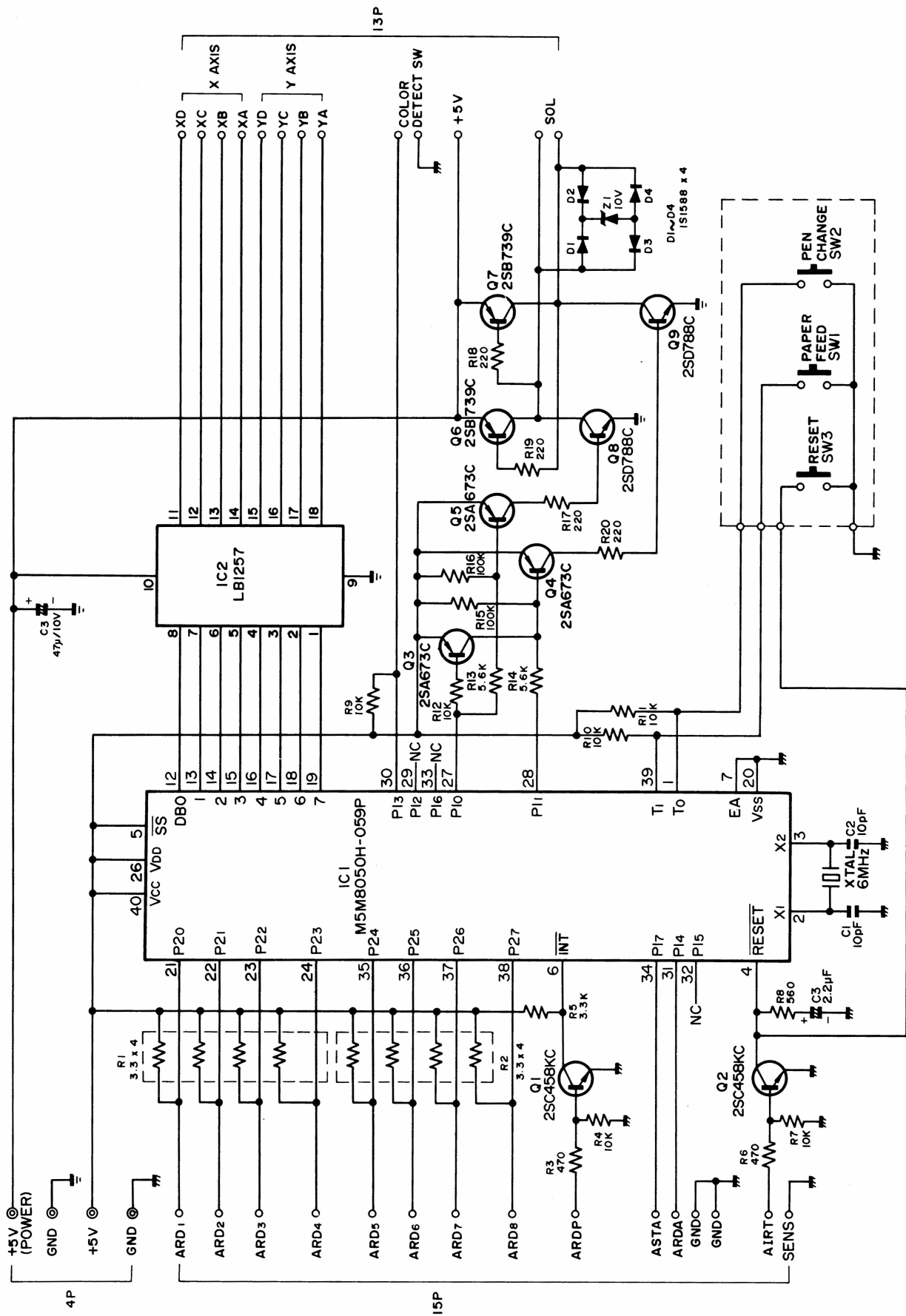
[Circuito matrice tastiera]

L'8255 emette il segnale di esplorazione tastiera dalla porta PA alla tastiera e legge i dati immessi da tastiera dalla porta PB. La figura seguente illustra la matrice dei tasti.





[Circuito stampante-plotter a colori]



Comandi e Sottoroutine Monitor

Capitolo 5



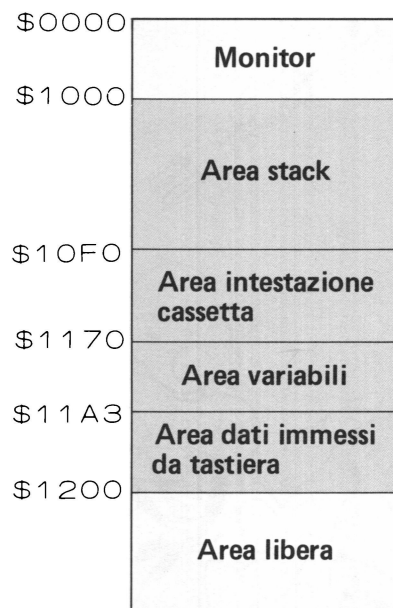
5.1 Comandi monitor

Il programma monitor inizia immediatamente dopo che l'alimentazione è stata attivata ed attende l'immissione di un comando monitor. I comandi monitor vengono elencati qui di seguito. In questo capitolo, **[CR]** indica che deve essere premuto il tasto di ritorno a capo.

- Comando L Carica i file su cassetta in memoria.
- Comando P Emette la stringa di caratteri specificata alla stampante. (Stampa)
- Comando M Cambia il contenuto della memoria. (Correzione memoria)
- Comando J Trasferisce il controllo all'indirizzo specificato. (Salto)
- Comando S Memorizza il contenuto del blocco di memoria specificato su cassetta. (Memorizzazione)
- Comando V Confronta il contenuto del nastro con il contenuto della memoria.
- Comando # Trasferisce il controllo all'area RAM.
- Comando B Provoca un segnale acustico ogni volta che viene premuto un tasto. Eseguendo nuovamente questo comando, si interrompe la segnalazione.

■ Configurazione dell'area di lavoro del monitor

Viene mostrata qui di seguito la configurazione dell'area di lavoro monitor da \$1000 a \$11FF.



Nota: Il monitor ROM descritto in questo capitolo non è uguale alla funzione monitor del programma interprete BASIC.

5.2 Funzioni ed uso dei comandi monitor

Questa sezione descrive le funzioni e l'uso degli otto comandi monitor.

- I comandi vengono eseguiti quando viene premuto il tasto `CR`. I caratteri devono essere immessi nell'ordine corretto. Se in una stringa di comando vengono inclusi dei caratteri illeciti (per esempio spazi), il monitor respinge il comando.
- Tutti i dati numerici devono essere immessi in formato esadecimale e tutti i dati vengono visualizzati in formato esadecimale. Pertanto, i dati da 1 byte vengono rappresentati con due posizioni esadecimali mentre i dati da 2 byte vengono rappresentati con 4 posizioni esadecimali.
Per esempio, il numero decimale 21 viene visualizzato come 15 ed il numero decimale 10 deve essere battuto come 0A. La posizione superiore (" 0 ") non può essere omessa.
- Se il numero di caratteri battuti come operando supera il numero specificato, i caratteri in eccesso vengono scartati.
- Ogni comando può accedere a qualsiasi posizione della memoria. Pertanto, il programma monitor può essere cambiato se i comandi vengono usati con incuria. **Poiché questo può produrre delle perdite di controllo sul sistema, si faccia attenzione ad evitare di cambiare il contenuto del programma monitor.**

5.2.1 Comando L

Formato
Funzione

L

Questo comando carica in memoria il primo file linguaggio macchina incontrato sulla cassetta. Dopo aver immesso il comando L, lo schermo cambia come segue.

*L J

↓ PLAY

Premete il tasto `PLAY` del registratore dati. Quando si trova un programma linguaggio macchina, viene visualizzato il messaggio "LOADING nome programma". Per esempio, durante il caricamento del programma interprete BASIC, viene visualizzato il messaggio seguente.

LOADING BASIC

5.2.2 Comando P (P : stampante)

Formato

Questo comando viene usato come segue per controllare la stampante.

*PABC J

Stampa le lettere "ABC".

*P&T J

Stampa la configurazione di prova.

*P&S J

Imposta la larghezza di riga (misura dei caratteri) ad 80 caratteri/riga

*P&L J

Imposta la larghezza di riga (misura dei caratteri) ad 40 caratteri/riga

*P&G J

Commuta la stampante al modo operativo grafico.

*P&C J

Cambia il colore della penna.

5.2.3 Comando M (M : modifica memoria)

Formato

M h h h h

h h h h indirizzo iniziale

Funzione

Questo comando viene usato per cambiare il contenuto della memoria di 1 byte per volta, partendo dall'indirizzo specificato.

*MC0000 J

C000 00 FF

C001 00 FF

C002 00 FF

C003 00 FF

C004 00 [SHIFT] + [BREAK]

*MC010 J

C010 00 88

C011 00 88

C012 00 88

C013 00 88

C014 00 [SHIFT] + [BREAK]

*

Per porre termine al comando M, premete simultaneamente i tasti [SHIFT] e [BREAK] .

5.2.4 Comando J (J : salto)

Formato	J h h h h h h h h indirizzo di destinazione
Funzione	Questo comando trasferisce il controllo all'indirizzo specificato; cioè imposta l'indirizzo specificato nel contatore del programma.

*J1200J Salta all'indirizzo \$1200.

5.2.5 Comando S (S : Memorizzazione)

Formato	S h h h h h' h' h' h' h" h" h" h" h h h h indirizzo iniziale h' h' h' h' indirizzo finale h" h" h" h" indirizzo di esecuzione
Funzione	Durante l'esecuzione, questo comando chiede l'immissione di un nome di file, quindi memorizza il contenuto della memoria da h h h h a h' h' h' h' sulla cassetta con il nome di file specificato. Presupponendo che un programma di linguaggio macchina nell'area da \$6000 a \$60A3 il cui indirizzo di esecuzione è a \$6050 debba essere memorizzato con il nome di file "MFILE", il comando viene immesso come segue.

*S600060A36050J
FILENAME? MFILEJ
⏏ RECORD.PLAY .

Accertatevi che sia caricata nel registratore dati una cassetta vergine e premete il tasto **RECORD**.

Se la linguetta di protezione scrittura della cassetta è stata tolta, non è possibile premere il tasto **RECORD**. Sostituirla con un'altra cassetta.

Questo programma può essere usato soltanto per memorizzare i programmi di linguaggio macchina.

WRITING MFILE
OK!



Nota: Per abortire la registrazione, tenete premuti i tasti **SHIFT** e **BREAK** sino a che compare il messaggio " * ".

5.2.6 Comando V (V : Verifica)

Formato	V
Funzione	Confronta un file di programma linguaggio macchina memorizzato su cassetta usando il comando S con il programma originale in memoria.
	 *V J ⏏ PLAY OK

Premete il tasto **PLAY** per leggere il file su cassetta quando viene visualizzato il messaggio " ⏏ PLAY ". Il messaggio "OK" viene visualizzato quando il contenuto del file su cassetta corrisponde a quello del programma originale; diversamente viene visualizzato il messaggio "Check sum error" .

Si consiglia di eseguire questo comando immediatamente dopo aver registrato un programma con il comando S.

5.2.7 Comando

Formato	#
Funzione	*# J
	Dopo aver premuto l'interruttore RESET, l'esecuzione di questo comando si produce lo stesso effetto ottenuto premendo contemporaneamente l'interruttore RESET ed il tasto CTRL .

5.2.8 Comando B (B : Campanello)

Formato	B
Funzione	*B J
	L'esecuzione di questo comando provoca il suono del campanello ogni volta che viene premuto un tasto. Eseguendolo nuovamente il campanello viene disattivato.

5.3 Sottoroutine monitor

Le sottoroutine seguenti sono previste per il **monitor 1Z-013A**. Ogni nome di sottoroutine rappresenta simbolicamente la funzione della sottoroutine corrispondente. Queste sottoroutine possono essere richiamate dai programmi utenti.

I registri memorizzati sono i registri i cui contenuti vengono ripristinati quando il controllo viene ritornato al programma chiamante. Il contenuto degli altri registri viene cambiato mediante l'esecuzione della sottoroutine.

Nome e punto di immissione (esadecimale)	Funzione	Registro memorizzato
CALL LETNL (0006)	Sposta il cursore all'inizio della riga successiva.	Diverso da AF
CALL PRINTS (000C)	Visualizza uno spazio nella posizione del cursore	Diverso da AF
CALL PRINTS (0012)	Visualizza il carattere corrispondente al codice ASCII memorizzato in ACC nella posizione del cursore. Vedere appendice A. 1 per i codice ASCII. Non viene visualizzato nessun carattere quando si immette il codice 0D (ritorno a capo) o 11-16 (codice di controllo cursore), ma la funzione corrispondente viene eseguita (un ritorno a capo per 0D ed il movimento del cursore per 11-16).	Diverso da AF.
CALL MSG (0015)	Visualizza un messaggio, partendo dalla posizione del cursore. L'indirizzo iniziale dell'area in cui il messaggio è memorizzato deve essere definito nel registro DE prima di chiamare questa sottoroutine, ed il messaggio deve terminare con un codice di ritorno a capo (0D). Il ritorno a capo non viene eseguito. Il cursore viene spostato se nel messaggio è incluso un codice di controllo cursore (11-16).	Tutti i registri
CALL BELL (003E)	Suona brevemente una A alta (circa 880 Hz).	Diverso da AF
CALL MELDY (0030)	Suona la musica secondo i dati musicali memorizzati nell'area di memoria partendo dall'indirizzo indicato nel registro DE. I dati musicali devono avere lo stesso formato dell'istruzione MUSIC del BASIC e devono terminare con 0D o C8. Quando l'esecuzione è completata, il controllo viene restituito al programma chiamante con il flag C fissato a 0; quando l'esecuzione viene interrotta con il tasto BREAK , il controllo viene ritornato con il flag C fissato ad 1.	Diverso da AF
CALL XTEMP (0041)	Determina il tempo musicale secondo i dati di tempo memorizzati nell'accumulatore (ACC). ACC ← 01 velocità minima ACC ← 04 velocità media ACC ← 07 velocità massima Si noti che i dati nell'accumulatore non sono il codice ASCII corrispondente ad 1-7 bensì il codice binario per il numero applicabile.	Tutti i registri
CALL MSTA (0044)	Genera un suono continuo della frequenza specificata. La frequenza viene data dall'equazione seguente. $\text{freq.} = 895 \text{ kHz}/\text{nn}'$ Qui nn' è un numero di 2 byte memorizzato negli indirizzi 11A1 e 11A2 (n in 11A2 ed n' in 11A1).	BC e DE

Nome e punto di immissione (esadecimale)	Funzione	Registro memorizzato	
CALL MSTP (0047)	Arresta il suono generato con la sottoroutine CALL MSTA.	Diverso da AF.	
CALL TIMST (0033)	Regola ed avvia l'orologio incorporato. I registri devono essere impostati come segue prima di richiamare questa routine. ACC ← 0 (AM), ACC ← 1 (PM) DE: ← Numero esadecimale di 4 cifre che rappresenta l'ora in secondi. ex) 1:00:00 = 3600 (secondi) = 0000111000010000 = 0E10	Diverso da AF.	
CALL TIMRD (003B)	Legge l'orologio incorporato e ritorna l'ora come segue. ACC ← 0 (AM), ACC ← 1 (PM) DE ← Numero esadecimale di 4 cifre che rappresenta l'ora in secondi.	Diverso da AF e DE.	
CALL BRKEY (001E)	Controlla se i tasti SHIFT e BREAK sono entrambi premuti. Il flag Z viene fissato quando vengono premuti simultaneamente; diversamente viene azzerato.	Diverso da AF.	
CALL GETL (0003)	Legge una riga di dati dalla tastiera e la memorizza nell'area di memoria partendo dall'indirizzo indicato nel registro DE. Questa routine arresta la lettura dei dati quando viene premuto il tasto RETURN, quindi attacca un codice di ritorno a capo (0D) alla fine dei dati letti. E' possibile immettere un massimo di 80 caratteri (incluso il codice di ritorno a capo) in una riga. I caratteri immessi da tastiera vengono rimandati in eco allo schermo, ed i codici di controllo del cursore possono essere inclusi nella riga. Quando vengono premuti simultaneamente i tasti SHIFT e BREAK, viene memorizzato un codice BREAK nell'indirizzo indicato nel registro DE e un codice di ritorno a capo viene memorizzato nell'indirizzo successivo.	Tutti i registri.	
CALL GETKY (001B)	Legge un codice di carattere (ASCII) dalla tastiera. Se non viene premuto alcun tasto, il controllo viene rimandato al programma chiamante con 00 impostato ACC. Non è previsto niente per evitare gli errori di lettura dovuti alla vibrazione dei tasti, ed i caratteri immessi non vengono rimandati in eco allo schermo. Quando viene premuto uno dei tasti speciali (per esempio DEL o CR), questa sottoroutine ritorna un codice ad ACC che è diverso dal codice corrispondente ASCII come mostrato qui di seguito. Qui, i codici di visualizzazione vengono usati per indirizzare i caratteri memorizzati nel generatore di caratteri e sono diversi dai codici ASCII.	Diverso da AF.	
Tasto speciale letto con GETKEY.	Tasto speciale.	Codice stabilito in ACC.	Codice di visualizzazione.
	DEL	60	C7
	INST	61	C8
	ALPHA	62	C9
	BREAK	64	CB
	CR	66	CD
	↓	11	C1
	↑	12	C2
	→	13	C3
	←	14	C4
	HOME	15	C5
	CLR	16	C6

APPENDICI



A.1 Tadella codici

■ Tabella codici ASCII

MSD è un'abbreviazione che sta per posizione più significativa, e rappresenta i quattro bit superiori di ogni codice, LSD è un'abbreviazione che sta per posizione meno significativa e rappresenta i quattro bit inferiori di ogni codice. I codici 11_H-16_H sono i codici di controllo del cursore. Per esempio, l'esecuzione di CALL PRNT (una sottoroutine monitor) con 15_H impostato in ACC ritorna il cursore alla posizione di sede, cioè nell'angolo superiore di sinistra. (Non viene visualizzato "▣").

MSD \ LSD		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000			SP	O	@	P	☛	☒	}	—	q	n		☐	—	☐
1	0001		↓	!	I	A	Q	H	☒	+	☒	a	/	■	☐	♠	●
2	0010		↑	"	2	B	R	I	☒	☐	e	z	Ü	■	☐		☐
3	0011		→	#	3	C	S	♠	☒	☐	`	w	m	☐	☐	—	♥
4	0100		←	\$	4	D	T	✚	☒	☐	~	s	☐	☐	☐	☐	☐
5	0101		H	%	5	E	U	✚	☒	☐	☒	u	☐	☐	☐	☐	☐
6	0110		☐	&	6	F	V	¥	☒	☐	t	i	☐	→	☐	—	☒
7	0111			'	7	G	W	☐	☒	☐	g	=	o	☐	☐	☐	○
8	1000			(8	H	X	☐	☒	☐	h	ö	l	■	☐	☐	♣
9	1001)	9	I	Y	☐	☐	☐	k	Ä	☐	☐	☐	☐	☐
A	1010			*	:	J	Z	☐	☐	☐	b	f	ö	☐	☐	☐	♦
B	1011			+	;	K	☐	☐	°	^	x	v	ä	☐	☐	☐	£
C	1100			,	<	L	☐	☐	☐	☐	d	☐	☐	☐	☐	☐	↓
D	1101	CR		—	=	M	☐	☐	☐	☐	r	ü	y	☐	☐	☐	☐
E	1110			.	>	N	↑	☐	☐	☐	p	β	☐	☐	☐	☐	☐
F	1111			/	?	O	←	☐	☐	☐	c	j	☐	☐	☐	☐	π

■ Tabella codici schermo

I codici schermo vengono usati per indirizzare le configurazioni di caratteri memorizzate nel generatore di caratteri. Questi codici devono essere trasferiti alla RAM video per visualizzare i caratteri.

Le sottoroutine monitor PRNT (0012_H) ed MSG (0015_H) convertono i codici ASCII in codici schermo e li trasferiscono alla posizione nella V-RAM indicata per il cursore.

I codici C1_H-C6_H servono per controllare il cursore.

MSD LSD		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	SP	P	O	▢	}	↑	π	▢		p	▢	▢	↓	▢	▢	SP
1	0001	A	Q	I	▢	♠	<	!	▢	a	q	▢	▢	↓	▢	▢	▢
2	0010	B	R	2	▢	▢	▢	"	▢	b	r	▢	▢	↑	▢	▢	▢
3	0011	C	S	3	▢	♥	#	▢	▢	c	s	▢	▢	→	▢	▢	▢
4	0100	D	T	4	▢	♦	▢	\$	▢	d	t	▢	▢	←	▢	▢	▢
5	0101	E	U	5	▢	←	@	%	▢	e	u	~	▢	H	▢	▢	▢
6	0110	F	V	6	▢	♣	▢	&	▢	f	v	▢	▢	▢	▢	▢	▢
7	0111	G	W	7	▢	●	>	'	▢	g	w	▢	▢	▢	▢	▢	▢
8	1000	H	X	8	▢	○	↓	(▢	h	x	▢	▢	H	▢	▢	▢
9	1001	I	Y	9	▢	?	▢)	▢	i	y	▢	▢	I	▢	▢	▢
A	1010	J	Z	▢	▢	○	→	+	▢	j	z	β	▢	▢	▢	▢	▢
B	1011	K	£	=	▢	▢	▢	*	▢	k	ä	ü	▢	▢	▢	▢	▢
C	1100	L	▢	;	▢	▢	▢	▢	▢	l	▢	ö	▢	▢	▢	▢	▢
D	1101	M	▢	▢	▢	▢	▢	▢	▢	m	▢	Ü	▢	▢	▢	▢	▢
E	1110	N	▢	▢	▢	▢	▢	▢	▢	n	▢	Ä	▢	▢	▢	▢	▢
F	1111	O	▢	,	▢	▢	▢	▢	▢	o	▢	Ö	▢	▢	▢	▢	▢

I pattern dei caratteri della pagina precedente sono contenuti nei byte 2K che costituiscono la prima metà del CG-ROM. I pattern dei caratteri della seconda metà del CG-ROM sono riportati nella pagina successiva. Tuttavia, i pattern dei caratteri del seconde byte 2K del CG-ROM non sono sostenuti dal BASIC e non possono essere comandati direttamente dalla tastiera. Sebbene essi possano essere evidenziati usando l'istruzione POKE come mostrato nell'esempio di cui sotto, non possono essere emessi e inviati ad alcuna stampante (sia interna che esterna).

< Esempi >

- (1) Il seguente esempio di programma mette in mostra pattern di caratteri della seconda metà del CG-ROM sul video CRT.

```

10 COLOR, , 7, 0
20 PRINT " 3 " ;
30 FOR J=55296 TO 56296 —→ 55296 = $D800
40 POKE J, 240 —→ Specifica la seconda metà del tyte
                    2K del CG-ROM. 240 = $F0
50 NEXT J
60 A=53248 : I=0 : H=0 —→ 53248 = $D000
70 POKE A, I
80 A=A+2
90 I=I+1 : IF I=256 THEN GOTO 120
100 H=H+1 : IF H=20 THEN A=A+40 : H=0
110 GOTO 70
120 GOTO 120

```

- (2) L'esempio di cui sotto illustra l'uso di linguaggio meccanico per visualizzare pattern di caratteri della seconda metà del CG-ROM sul video CRT.

LD HL, D000H	DISP : XOR A
CALL DISP	LD B, 00H
LD HL, D208H	DISP2 : LD (HL), A
CALL DISP	INC HL
LD A, F1H	INC A
LD HL, DA08H	DEC B
LD DE, DA09H	JP NZ, DISP2
LD BC, 00FFH	RET
LD (HL), A	
LDIR	
END	

■ Visualizza la tavola dei codici (seconda metà dei byte 2K).

MSD LSD		0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0	0 0 0 0																
1	0 0 0 1																
2	0 0 1 0																
3	0 0 1 1																
4	0 1 0 0																
5	0 1 0 1																
6	0 1 1 0																
7	0 1 1 1																
8	1 0 0 0																
9	1 0 0 1																
A	1 0 1 0																
B	1 0 1 1																
C	1 1 0 0																
D	1 1 0 1																
E	1 1 1 0																
F	1 1 1 1																

■ **Tabella dei codici ASCII per la stampante-plotter a colori.**

I caratteri grafici diversi da quelli precedentemente indicati non possono essere stampati, ma il codice esadecimale corrispondente viene stampato in un diverso colore di penna.

MSD LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	Ø	@	P			}		q	n				
1		↓	∇	1	A	Q					a					
2		↑	"	2	B	R				e	Z	ū				
3		→	#	3	C	S				`	w	m				
4		←	\$	4	D	T				~	s					
5		H	%	5	E	U					u					
6		Ⓢ	&	6	F	V				t	i		→			
7			'	7	G	W				g		o		=		
8			[8	H	X				h	ō	!				
9]	9	I	Y					k	ā				
A			*	:	J	Z				b	f	ö				
B			+	;	K	[°	^	x	v	ā				£
C			,	<	L	\				d						↓
D			-	=	M]				^	ü	y				
E			.	>	N	↑				p	ß	{				
F			/	?	O	←				c	j		=			π

A.2 Specifiche dei calcolatori serie MZ-700

A.2.1 MZ-700

CPU:	SHARP LH0080A (Z80A)
Orologio (clock):	3,5 MHz
Memoria:	ROM 4K byte (ROM) 2K byte (generatore di caratteri) RAM 64K byte (area programma) 4K byte (RAM video)
Emissione video:	Sistema PAL Segnale RGB Segnale composto (B/N) Segnale RF (UHF 36 ± 3 CH, B/N)
Formato schermo:	40 caratteri per 25 righe Matrice caratteri 8 x 8 punti
Colori:	8 colori per i caratteri 8 colori per lo sfondo
Funzione musicale:	incorporata (emissione massima 500 mW)
Orologio (clock):	incorporato (orologio 24 ore, nessuna riserva)
Tasti:	69 tasti Standard ASCII Tasti funzionali definibili, tasti di controllo del cursore
Funzione di revisione:	editore schermo (Controllo cursore, sede, annullamento, inserimento e cancellazione)
Temperatura:	Di esercizio 0 ~ 35°C Di conservazione -20 ~ 60°C
Umidità:	Di esercizio 85% o meno Di conservazione 85% o meno
Dimensioni:	MZ-731; 400 (L) x 305 (P) x 102 (H) mm MZ-721; 440 (L) x 305 (P) x 86 (H) mm MZ-711; 440 (L) x 305 (P) x 86 (H) mm
Peso:	MZ-731; 4,6 kg MZ-721; 4,0 kg MZ-711; 3,6 kg
Accessori:	Nastro a cassetta (BASIC (lato A)) Programmi applicativi (lato B) Manuale Dell'utente, etichette funzioni, cavo di alimentazione, cavo di connessione televisore Gli attacchi per la stampante-plotter a colori vengono elencati in seguito.

A.2.2 Specifiche piastre CPU

CPU:	LH0080A (Z80A)	1	
PPI:	8255	1	
PIT:	8253	1	
Controllore di memoria			
(CRTC):	M60719	1	
ROM:	monitor	4K byte ROM	1
	Generatore di caratteri	4K byte ROM	1
RAM:	64K bit D-RAM	8	
	2K byte S-RAM	2	
Bus I/E:	Bus I/E espansione	1	
	Bus I/E stampante	2 (Non possono essere usati	
	Terminale READ/WRITE cassetta	2 contemporaneamente)	
	Terminale barra di comando	2	

A.2.3 Specifiche stampante-plotter a colori

Sistema di stampa:	4 colori selezionabili usando panne a sfera
Colori:	1. nero; 2. blu; 3. verde; 4. rosso.
Velocità di stampa:	10 caratteri/secondo in media quando si stampa con i caratteri di dimensione più piccola.
Larghezza di riga:	80 colonne, 40 colonne o 26 colonne (scelta dal software)
Numero di caratteri:	115 (inclusi i caratteri ASCII)
Risoluzione:	0,2 mm
Accessori:	bobina di carta (1), penna a sfera (nero, blu, verde, rosso) portacarta (sinistra e destra) albero bobina (1) guidacarta (1)

A.2.4 Specifiche registratore dati

Tipo:	Meccanismo a cassetta compatto standard IEC.
Sistema di	
registrazione/lettura:	2 piste, 1 canale monofonico
Velocità nominale:	4,8 cm/s \pm 3,5%
Tipo di interruttore	
di controllo:	Tipo piano
Interruttore di	
controllo:	Tasti PLAY, FF, REW, STOP/EJECT e REC e pulsante di azionamento contatore
Metodo di	
trasferimento dati:	Metodo Sharp PWM
Velocità di	
trasferimento dati:	1200 bps (typ.)
Nastro:	Nastro a cassetta audio normale

A.2.5 Specifiche dell'alimentatore

(Fornisce l'alimentazione alla stampante-plotter a colori ed al registratore dati, nonché all'unità principale).

Ingresso:	240/220 V \pm 10%, 50/60 Hz, 20 W
Uscita:	5 V

A.3 Lista messaggi errore BASIC

Il programma interprete BASIC visualizza un messaggio errori in uno dei formati seguenti quando si verifica un errore durante l'uso.

- | | |
|--|--------------------------|
| 1. < tipo di errore > ERROR | (Errore modo diretto) |
| 2. < tipo di errore > ERROR IN numero riga | (Errore modo esecuzione) |

I messaggi di errore nel formato 1 vengono emessi quando viene rilevato un errore durante l'esecuzione di un comando diretto o l'immissione di un programma. I messaggi di errore nel formato 2 vengono emessi quando viene rilevato un errore durante l'esecuzione di un programma.

I messaggi di errore che possono essere visualizzati vengono indicati qui di seguito.

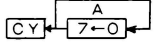
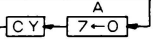
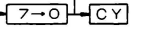
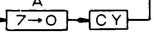
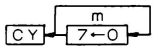
SYNTAX

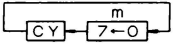
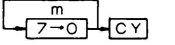
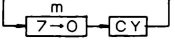
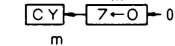
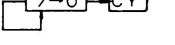
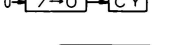
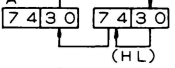
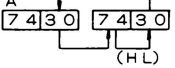
Numero errore	Messaggio visualizzato	Descrizione
1	Syntax error	Errore sintassi
2	Over flow error	I dati numerici usati esulano dalla gamma specificata, oppure si è verificato un superamento di capacità (overflow).
3	Illegal data error	E' stata usata una costante o una variabile illecita.
5	String length error	La lunghezza della stringa supera 255 caratteri.
6	Memory capacity error	La capacità della memoria è insufficiente.
7	Array def. error	Si è tentato di ridefinire una matrice in un'ampiezza superiore a quella precedentemente definita.
8	Linlength error	La riga è troppo lunga.
10	GOSUB nesting error	Il numero di livelli dell'annidamento GOSUB ha superato il limite determinato dallo spazio di memoria utilizzabile.
11	FOR~NEXT error	Il numero di livelli delle iterazioni FOR~NEXT ha superato il limite determinato dall'area di memoria utilizzabile.
12	DEF FN nesting error	Il numero di livelli dell'annidamento DEF FN ha superato il limite.
13	NEXT error	E' stato usato NEXT senza un FOR corrispondente.
14	RETURN error	E' stato usato RETURN senza un GOSUB corrispondente.
15	Un def. function error	E' stata chiamata una funzione non definita.
16	Un def. line num. error	E' stato indicato un numero di riga non utilizzato.
17	Can't continue	Il comando CONT non può essere eseguito.
18	Memory protection	E' stato fatto il tentativo di scrivere dei dati nell'area di controllo BASIC.
19	Instruction error	I comandi e le istruzioni nel modo diretto sono mescolati.
20	Can't RESUME error	Non è possibile eseguire RESUME.
21	RESUME error	E' stato fatto il tentativo di eseguire RESUME quando non si era verificato alcun errore.
24	READ error	E' stata usata READ senza un'istruzione DATA corrispondente.
43	Already open error	E' stata emessa un'istruzione OPEN ad un file già aperto.
63	Out of file error	Fuori file durante la lettura del file.
65	Printer is not ready	La stampante non è collegata.
68	Printer mode error	Errore nel modo stampante-plotter a colori.
70	Check sum error	Errore totale di controllo (durante la lettura del nastro)

A.4 Gruppo di istruzioni Z80A

Viene dato qui di seguito per riferimento un sommario delle istruzioni Z80A.

Mnemonica	Operazione simbolica	Codice operativo	Mnemonica	Operazione simbolica	Codice operativo
Gruppo di caricamento 8 bit			LD HL, (nn)	$H \leftarrow (nn+1)$ $L \leftarrow (nn)$	00 101 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$
LD r, r'	$r \leftarrow r'$	01 r r'	LD dd, (nn)	$dd_H \leftarrow (nn+1)$ $dd_L \leftarrow (nn)$	11 101 101 01 dd1 011 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$
LD r, n	$r \leftarrow n$	00 r 110 $\leftarrow n \rightarrow$	LD IX, (nn)	$IX_H \leftarrow (nn+1)$ $IX_L \leftarrow (nn)$	11 011 101 00 101 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$
LD r, (HL)	$r \leftarrow (HL)$	01 r 110	LD IY, (nn)	$IY_H \leftarrow (nn+1)$ $IY_L \leftarrow (nn)$	11 111 101 00 101 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$
LD r, (IX+d)	$r \leftarrow (IX+d)$	11 011 101 01 r 110 $\leftarrow d \rightarrow$	LD (nn), HL	$(nn+1) \leftarrow H$ $(nn) \leftarrow L$	00 100 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$
LD r, (IY+d)	$r \leftarrow (IY+d)$	11 111 101 01 r 110 $\leftarrow d \rightarrow$	LD (nn), dd	$(nn+1) \leftarrow dd_H$ $(nn) \leftarrow dd_L$	11 101 101 01 dd0 011 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$
LD (HL), r	$(HL) \leftarrow r$	01 110 r	LD (nn), IX	$(nn+1) \leftarrow IX_H$ $(nn) \leftarrow IX_L$	11 011 101 00 100 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$
LD (IX+d), r	$(IX+d) \leftarrow r$	11 011 101 01 110 r $\leftarrow d \rightarrow$	LD (nn), IY	$(nn+1) \leftarrow IY_H$ $(nn) \leftarrow IY_L$	11 111 101 00 100 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$
LD (IY+d), r	$(IY+d) \leftarrow r$	11 111 101 01 110 r $\leftarrow d \rightarrow$	LD SP, HL	$SP \leftarrow HL$	11 111 001
LD (HL), n	$(HL) \leftarrow n$	00 110 110 $\leftarrow n \rightarrow$	LD SP, IX	$SP \leftarrow IX$	11 011 101 11 111 001
LD (IX+d), n	$(IX+d) \leftarrow n$	11 011 101 00 110 110 $\leftarrow d \rightarrow$ $\leftarrow n \rightarrow$	LD SP, IY	$SP \leftarrow IY$	11 111 101 11 111 001
LD (IY+d), n	$(IY+d) \leftarrow n$	11 111 101 00 110 110 $\leftarrow d \rightarrow$ $\leftarrow n \rightarrow$	PUSH qq	$(SP-2) \leftarrow qq_L$ $(SP-1) \leftarrow qq_H$	11 qq0 101
LD A, (BC)	$A \leftarrow (BC)$	00 001 010	PUSH IX	$(SP-2) \leftarrow IX_L$ $(SP-1) \leftarrow IX_H$	11 011 101 11 100 101
LD A, (DE)	$A \leftarrow (DE)$	00 011 010	PUSH IY	$(SP-2) \leftarrow IY_L$ $(SP-1) \leftarrow IY_H$	11 111 101 11 100 101
LD A, (nn)	$A \leftarrow (nn)$	00 111 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$	POP qq	$qq_H \leftarrow (SP+1)$ $qq_L \leftarrow (SP)$	11 qq0 001
LD (BC), A	$(BC) \leftarrow A$	00 000 010	POP IX	$IX_H \leftarrow (SP+1)$ $IX_L \leftarrow (SP)$	11 011 101 11 100 001
LD (DE), A	$(DE) \leftarrow A$	00 010 010	POP IY	$IY_H \leftarrow (SP+1)$ $IY_L \leftarrow (SP)$	11 111 101 11 100 001
LD (nn), A	$(nn) \leftarrow A$	00 110 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$			
LD A, I	$A \leftarrow I$	11 101 101 01 010 111			
LD A, R	$A \leftarrow R$	11 101 101 01 011 111			
LD I, A	$I \leftarrow A$	11 101 101 01 000 111			
LD R, A	$R \leftarrow A$	11 101 101 01 001 111			
Gruppo di caricamento 16 bit			Scambio gruppo e blocco trasferimento e ricerca gruppo		
LD dd, nn	$dd \leftarrow nn$	00 dd0 001 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$	EX DE, HL	$DE \leftrightarrow HL$	11 101 011
LD IX, nn	$IX \leftarrow nn$	11 011 101 00 100 001 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$	EX AF, AF'	$AF \leftrightarrow AF'$	00 001 000
LD IY, nn	$IY \leftarrow nn$	11 111 101 00 100 001 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$	EXX	$(BC) \leftrightarrow (BC')$ $(DE) \leftrightarrow (DE')$ $(HL) \leftrightarrow (HL')$	11 011 001
			EX (SP), HL	$H \leftrightarrow (SP+1)$ $L \leftrightarrow (SP)$	11 100 011
			EX (SP), IX	$IX_H \leftrightarrow (SP+1)$ $IX_L \leftrightarrow (SP)$	11 011 101 11 100 011
			EX (SP), IY	$IY_H \leftrightarrow (SP+1)$ $IY_L \leftrightarrow (SP)$	11 111 101 11 100 011

Mnemonica	Operazione simbolica	Codice operativo	Mnemonica	Operazione simbolica	Codice operativo
LDI	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC-1	11 101 101 10 100 000	DEC m	m←m-1	← d → 101
LDIR	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC-1	11 101 101 10 110 000	Gruppo aritmetico e di controllo per scopi generali		
LDD	Ripetere sino a che BC=0 (DE)←(HL) DE←DE-1 HL←HL-1 BC←BC-1	11 101 101 10 101 000	DAA	Regolazione decimale sul contenuto di A dopo l'addizione o la sottrazione	00 100 111
LDDR	(DE)←(HL) DE←DE-1 HL←HL-1 BC←BC-1	11 101 101 10 111 000	CPL	$A \leftarrow \overline{A}$	00 101 111
CPI	Ripetere sino a che BC=0 A-(HL) HL←HL+1 BC←BC-1	11 101 101 10 100 001	NEG	$A \leftarrow \overline{A} + 1$	11 101 101
CPIR	A-(HL) HL←HL+1 BC←BC-1	11 101 101 10 110 001	CCF	$CY \leftarrow \overline{CY}$	01 000 100
CPD	Ripetere sino a che A=(HL) o BC=0 A-(HL) HL←HL-1 BC←BC-1	11 101 101 10 101 001	SCF	$CY \leftarrow 1$	00 111 111
CPDR	A-(HL) HL←HL-1 BC←BC-1	11 101 001 10 111 001	NOF	Nessuna operazione, ma PC viene incrementato.	00 110 111
Gruppo logico ed aritmetico da 8 bit			HALT	Arresto del CPU	01 110 110
ADD A, r	$A \leftarrow A + r$	10 000 r	DI	IFF←0	11 110 011
AD A, n	$A \leftarrow A + n$	11 000 110 ← n →	EI	IFF←1	11 111 011
ADD A, (HL)	$A \leftarrow A + (HL)$	10 000 110	IM0	Impostare modo interruzione 0	11 101 101
ADD A, (IX+d)	$A \leftarrow A, (IX+d)$	11 011 101 10 000 110 ← d →	IM1	Impostare modo interruzione 1	11 101 101
ADD A, (IY+d)	$A \leftarrow A + (IY+d)$	11 111 101 10 000 110 ← d →	IM2	Impostare modo interruzione 2	01 010 110 11 101 101 01 011 110
ADC A, s	$A \leftarrow A + s + CY$	001	Gruppo aritmetico da 16 bit		
SUB s	$A \leftarrow A - s$	010	ADD HL, ss	$HL \leftarrow HL + ss$	00 ss1 001
SBC A, s	$A \leftarrow A - s - CY$	011	ADC HL, ss	$HL \leftarrow HL + ss + CY$	11 101 101
AND s	$A \leftarrow A \wedge s$	100	SBC HL, ss	$HL \leftarrow HL - ss - CY$	01 ss1 010 11 101 101 01 ss0 010
OR s	$A \leftarrow A \vee s$	110	ADD IX, pp	$IX \leftarrow IX + pp$	11 011 101 00 pp1 001
XOR s	$A \leftarrow A \oplus s$	101	ADD IY, rr	$IY \leftarrow IY + rr$	11 111 101 00 rr1 001
CP s	$A - s$	111	INC ss	$ss \leftarrow ss + 1$	00 ss0 011
INC r	$r \leftarrow r + 1$	00 r 100	INC IX	$IX \leftarrow IX + 1$	11 011 101 00 100 011
INC (HL)	$(HL) \leftarrow (HL) + 1$	00 110 100	INC IY	$IY \leftarrow IY + 1$	11 111 101 00 100 011
INC (IX+d)	(IX+d) ←(IX+d)+1	11 011 101 00 110 100 ← d →	DEC ss	$ss \leftarrow ss - 1$	00 ss1 011
INC (IY+d)	(IY+d) ←(IY+d)+1	11 111 101 00 110 100 ← d →	DEC IX	$IX \leftarrow IX - 1$	11 011 101 00 101 011
			DEC IY	$IY \leftarrow IY - 1$	11 111 101 00 101 011
			Rotazione e spostamento gruppo		
			RLCA		00 000 111
			RLA		00 010 111
			RRCA		00 001 111
			RRA		00 011 111
			RLC r		11 001 011
			RLC (HL)		00 000 r 11 001 011 00 000 110

Mnemonica	Operazione simbolica	Codice operativo	Mnemonica	Operazione simbolica	Codice operativo
RLC (IX+d)		11 011 101 11 001 011 ← d → 00 000 110	Salto gruppo		
RLC (IY+d)		11 111 011 11 001 011 ← d → 00 000 110	JP nn	PC←nn	11 000 011 ← n → ← n → 11 cc 010 ← n → ← n →
RL m		010	JP cc, nn	Se la condizione cc è vera, PC←-nn; diversamente, continuare.	
RRC m		001	JR e	PC←PC+e	00 011 000 ← e-2 →
RR m		011	JR C, e	Se C=0, continuare. Se C=1, PC←PC+e.	00 111 000 ← e-2 →
SLA m		100	JR Z, e	Se Z=0, continuare. Se C=1, PC←PC+e,	00 101 000 ← e-2 →
SRA m		101	JR NC, e	Se C=1, continuare. Se C=0, PC←PC+e.	00 110 000 ← e-2 →
SRL m		111	JR NZ, e	Se Z=1, continuare. Se Z=0, PC←PC+e.	00 100 000 ← e-2 →
RLD		11 101 101 01 101 111	JP (HL)	PC←HL	11 101 001
RRD		11 101 101 01 100 111	JP (IX)	PC←IX	11 011 101 11 101 001 11 111 101 11 101 001
Impostazione bit, ripristino e prova gruppo			JP (IY)	PC←IY	00 010 000 ← e-2 →
BIT b, r	$Z \leftarrow \overline{rb}$	11 001 011 01 b r	DJNZ e	Se B=0, continuare. Altrimenti PC←PC+e. PC←PC+e	
BIT b, (HL)	$Z \leftarrow \overline{(HL)b}$	11 011 011 01 b 110	Chiamata e ritorno gruppo		
BIT b, (IX+d)	$Z \leftarrow \overline{(IX+d)b}$	11 011 101 11 001 011 ← d → 01 b 110	CALL nn	(SP-1)←PC _H (SP-2)←PC _L PC←nn	11 001 101 ← n → ← n →
BIT b, (IY+d)	$Z \leftarrow \overline{(IY+d)b}$	11 111 101 11 001 011 ← d → 01 b 110	CALL cc, nn	Se la condizione cc è falsa, continuare; diversamente come CALL nn.	11 cc 100 ← n → ← n →
SET b, r	rb←1	11 001 011 11 b r	RET	PC _L ←(SP) PC _H ←(SP+1)	11 001 001
SET b, (HL)	(HL)b←1	11 001 011 11 b 110	RET cc	Se la condizione cc è falsa, continuare; diversamente come RET.	11 cc 000
SET b, (IX+d)	(IX+d)b←1	11 001 101 11 001 011 ← d → 11 b 110	RETI	Ritorno da interruzione	11 101 101 01 001 101
SET b, (IY+d)	(IY+d)b←1	11 111 101 11 001 011 ← d → 11 b 110	RETN	Ritorno da NMI.	11 101 101 01 000 101
RES b, m	mb←0	11 b 110 10	RST p	(SP-1)←PC _H (SP-2)←PC _L PC _H ←0 PC _L ←p	11 t 111

Mnemonica	Operazione simbolica	Codice operativo	Mnemonica	Operazione simbolica	Codice operativo
Immissione ed emissione gruppo					
IN A, (n)	$A \leftarrow (n)$	11 011 011 $\leftarrow n \rightarrow$	OUT (n), A	$(n) \leftarrow A$	11 010 011 $\leftarrow n \rightarrow$
IN r, (C)	$r \leftarrow (C)$	11 101 101 01 r 000	OUT (C), r	$(C) \leftarrow r$	11 101 101 01 r 001
INI	$(HL) \leftarrow (C)$ $B \leftarrow B - 1$ $HL \leftarrow HL + 1$	11 101 101 10 100 010	OUTI	$(C) \leftarrow (HL)$ $B \leftarrow B - 1$ $HL \leftarrow HL + 1$	11 101 101 10 100 011
INIR	$(HL) \leftarrow (C)$ $B \leftarrow B - 1$ $HL \leftarrow HL + 1$ Ripetere sino a che B=0	11 101 101 10 110 010	OTIR	$(C) \leftarrow (HL)$ $B \leftarrow B - 1$ $HL \leftarrow HL + 1$ Ripetere sino a che B=0	11 101 101 10 110 011
IND	$(HL) \leftarrow (C)$ $B \leftarrow B - 1$ $HL \leftarrow HL - 1$	11 101 101 10 101 010	OUTD	$(C) \leftarrow (HL)$ $B \leftarrow B - 1$ $HL \leftarrow HL - 1$	11 101 101 10 101 011
INDR	$(HL) \leftarrow (C)$ $B \leftarrow B - 1$ $HL \leftarrow HL - 1$ Ripetere sino a che B=0	11 101 101 10 111 010	OTDR	$(C) \leftarrow (HL)$ $B \leftarrow B - 1$ $HL \leftarrow HL - 1$ Ripetere sino a che B=0	11 101 101 10 111 011

Nota: Il significato dei simboli usati nella tabella precedente è il seguente.

r, r'	Registro	dd, ss	Coppia registri	qq	Coppia registri	pp	Coppia registri
000	B	00	BC	00	BC	00	BC
001	C	01	DE	01	DE	01	DE
010	D	10	HL	10	HL	10	IX
011	E	11	SP	11	AF	11	SP
100	H						
101	L						
111	A						

rr	Coppia registri	b	Impostazione bit	cc	Condizione	t	p
00	BC	000	0	000	NZ non zero	000	00H
01	DE	001	1	001	Z zero	001	08H
10	IX	010	2	010	NC non carry	010	10H
11	SP	011	3	011	C carry	011	18H
		100	4	100	PO parity odd	100	20H
\wedge : Operazione AND		101	5	101	PE parity even	101	28H
\vee : Operazione OR		110	6	110	P sign positive	110	30H
\oplus : Operazione OR esclusivo		111	7	111	M sign negative	111	38H

s: r, n, (HL), (IX + d), (IY + d)

CY: Riporto flip-flop

(Coppia registri)_H: 8 bit superiori della coppia di registri

m : r, (HL), (IX + d), (IY + d)

mb : Bit b o posizione m

(Coppia registri)_L: 8 bit inferiori della coppia di registri

Per i codici operativi ADC, SUB, SBC, AND, OR, XOR e CP, i bit in () sostituiscono (000) nel gruppo ADD.

Per il codice operativo DEC, (101) sostituisce (100) nel gruppo INC.

Operazioni simili valgono per i codici operativi del gruppo di rotazione e di spostamento (SHIFT) e del gruppo di impostazione, ripristino e prova bit.

A.5 Lista di assemblaggio programma monitor

Nelle pagine seguenti viene data una lista di assemblaggio del monitor 1Z-013A.

Questa lista di assemblaggio è stata prodotta con l'assemblatore Z80 contenuto nel sistema operativo a Floppy DOS. Il significato dei simboli della lista è il seguente.

Indirizzo relativo	Messaggio assemblatore		Mnemonico (codice operativo)		Commento
	Codice oggetti rilocabili	Identificazione	Operando		
20	02A7 13		INC DE		
21	02A8 13		INC DE		
22	02A9 13		INC DE		
23	02AA C9		RET		
24	02AB	;			
25	02AB	;			
26	02AB	;			
27	02AB	; ORG 02ABH ; MLDST			
28	02AB	;			
29	02AB	; MELODY START & STOP			
30	02AB	;			
31	02AB	MLDST: ENT			
32	02AB 2AA111		LD HL, (RATIO)		
33	02AE 7C		LD A, H		
34	02AF B7		OR A		
35	02B0 280C		JR Z, MLDSP		
36	02B2 D5		PUSH DE		
37	02B3 EB		EX DE, HL		
38	02B4 2104E0		LD HL, CONTO		
39	02B7 73		LD (HL), E		
40	02B8 72		LD (HL), D		
41	02B9 3E01		LD A, 1		
42	02BB D1		POP DE		
43	02BC 1806		JR MLDS1		
44	02BE	;			
45	02BE	MLDSP: ENT			
46	02BE 3E36		LD A, 36H		; MODE SET (8253 C0)
47	02C0 3207E0		LD (CONTF), A		; E007H
48	02C3 AF		XOR A		
49	02C4 3208E0	MLDS1: LD	(SUNDG), A		; E008H
50	02C7 C9		RET		; TEHRD RESET

Poiché l'indirizzo iniziale del monitor 1Z-013A è impostato a \$0000, gli indirizzi rilocabili ed i codici oggetto nella lista di assemblaggio possono essere assunti rispettivamente come indirizzi assoluti e come codice oggetto.

Questa lista di assemblaggio viene fornita solo per riferimento e la Sharp Corporation non assume alcune responsabilità per la risposta ad eventuali domande.

Si noti che questo monitor differisce dal programma monitor incluso nel programma interprete BASIC.

```

** Z80 ASSEMBLER SB-7201 <12-013A> PAGE 01 04.07.83
01 0000 ;
02 0000 ;
03 0000 ; MONITOR PROGRAM 12-013A
04 0000 ;
05 0000 ; (M2-700) FOR PAL
06 0000 ;
07 0000 ; REV. B3.4.7
08 0000 ;
09 0000 ;
10 0000 MONIT: ENT
11 0000 C34A00 GETL: ENT
12 0003 ENT
13 0003 C3E607 J?GETL
14 0006 ENT
15 0006 C30E09 J?LTNL
16 0009 ENT
17 0009 C31B09 J?NL
18 000C ENT
19 000C C32009 J?PRTS
20 000F ENT
21 000F C32409 J?PRTT
22 0012 ENT
23 0012 C33509 J?PRNT
24 0015 ENT
25 0015 C39308 MSG: ENT
)
26 0018 MSGX: ENT
27 001B C3A108 ENT
28 001B GETKY: ENT
29 001B C3BD08 J?GET
30 001E BRKEY: ENT
31 001E C3320A J?BRK
32 0021 WRINF: ENT
33 0021 C33604 J?WRI
34 0024 WRDAT: ENT
35 0024 C37504 J?WRD
36 0027 RDINF: ENT
37 0027 C3DB04 J?RDI
38 002A RDDAT: ENT
39 002A C3FB04 J?RDD
40 002D VERFY: ENT
41 002D C38B05 J?VRFY
42 0030 MELDY: ENT
43 0030 C3C701 J?MLDY
44 0033 TIMST: ENT
45 0033 C30B03 J?TMST
46 0036 NOP
47 0037 NOP
48 003B TIMRD: ENT
49 003B C33B10 J?TMRD
50 003B C35B03 J?BELL
51 003E XTEMP: ENT
52 003E C37705 J?BEL
53 0041 J?TEMP
54 0041 C3E502 J?TEMP
55 0044 MSTA: ENT
56 0044 C3AB02 J?MLDST
57 0047 MSTP: ENT
58 0047 C3BE02 J?MLDSP
59 004A ;
60 004A ;

** Z80 ASSEMBLER SB-7201 <12-013A> PAGE 02 04.07.83
01 004A ;
02 004A LD SP,SP
03 004A 31F010 LD IM,1
04 004D ED56 CALL ?MODE
05 004F CD3E07 CALL ?BRK
06 0052 CD320A J?NC,STO
07 0055 3019 CP 20H
08 0057 FE20 J?NZ,STO
09 0059 2015 J?NZ,STO
10 005B ENT
11 005B D3E1 (E1H),A
12 005D 110FF DE,FFFOH
13 0060 21B00 HL,$MCP
14 0063 010500 BC,05
15 0066 ED80 LDIR
16 006B C3FOFF JP FFFOH
17 006B ;
18 006B $MCP: ENT
19 006B D3E0 DEFW
20 006D C300 DEFW
21 006F 00 DEFW
22 0070 ;
23 0070 ENT
24 0070 06FF LD B,FFH
25 0072 21F110 LD HL,NAME
26 0075 CDD80F ?CLER
27 007B 3E16 LD A,16H
28 007A CD1200 PRNT
29 007D 3E71 LD A,71H
30 007F 2100D8 HL,DB00H
31 0082 CDD509 ?CLRBL
32 0085 21BD03 LD HL,TIMIN
33 008B 3EC3 LD A,C3H
34 008A 323B10 LD (103BH),A
35 008D 223910 LD (1039H),HL
36 0090 3E04 LD A,04
37 0092 329E11 LD (TEMPW),A
38 0095 CDBE02 MLDSP
39 009B CD0900 NL
40 009B 11E706 LD DE,MSG?3
41 009E DF RST 3
42 009F CD7705 CALL ?BEL
43 00A2 SS: ENT
44 00A2 3E01 LD A,01H
45 00A4 329D11 LD (SWRK),A
46 00A7 2100E8 LD HL,EB00H
47 00AA 77 LD (HL),A
48 00AB 1B55 JR FD2
49 00AD ENT
50 00AD CD0900 CALL NL
51 00B0 3E2A LD A,2AH
52 00B2 CD1200 PRNT
53 00B5 11A311 LD DE,BUFER
54 00B8 CD0300 GETL
55 00BB 1A LD A,(DE)
56 00BC 13 DE
57 00BD FE0D CP ODH
58 00BF 2BEC J?Z,ST1
59 00C1 FE4A CP J?
60 00C3 2B2E JR Z,GOTO

```

```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 03          04.07.83
01 00C5 FE4C      CP      'L'
02 00C7 2B4B      JR      Z,LOAD
03 00C9 FE46      CP      'F'
04 00CB 2B32      JR      Z,FD
05 00CD FE42      CP      'B'
06 00CF 2B26      JR      Z,SG
07 00D1 FE23      CP      '#'
08 00D3 2B86      JR      Z,CNYO
09 00D5 FES0      CP      'P'
10 00D7 2B7C      JR      Z,PTST
11 00D9 FE4D      CP      'M'
12 00DB CAAB07    JP      Z,MOR
13 00DE FES3      CP      'S'
14 00E0 CASE0F    JP      Z,SAVE
15 00E3 FES6      CP      'V'
16 00E5 CACB0F    JP      Z,VRFY
17 00E8 FE44      CP      'D'
18 00EA CA290D    JP      Z,DUMP
19 00ED
20 00ED
21 00ED          DEFS    +4
22 00F1
23 00F1 1BC8      JR      ST2
24 00F3
25 00F3          JUMP    COMMAND
26 00F3
27 00F3 CD3D01    GOTO:   CALL  HEXIY
28 00F6 E9        JP      (HL)
29 00F7
30 00F7          KEY    SOUND ON OFF
31 00F7
32 00F7 3A9D11    SG:      LD      A,(SWRK)
33 00FA 1F        RRA
34 00FB 3F        CCF
35 00FC 17        RLA
36 00FD 1BA5      JR      SS+2
37 00FF
38 00FF          FLOPPY
39 00FF
40 00FF 2100F0    FD:      LD      HL,F000H
41 0102 7E        FD2:   LD      A,(HL)
42 0103 B7        OR      A
43 0104 20A7      JR      NZ,ST1
44 0106 E9        FD1:   JP      (HL)
45 0107
46 0107
47 0107          ERROR (LOADING)
48 0107
49 0107          ?ER:   ENT      02H
50 0107 FE02      CP      Z,ST1
51 0109 2BA2      LD      DE,MSGE1
52 010B 114701    RST      3
53 010E DF        JR      ST1
54 010F 1B9C
55 0111
56 0111          LOAD    COMMAND
57 0111
58 0111          LOAD:   CALL  ?RDI
59 0111 CDB004    JR      C,?ER
60 0114 3BF1

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 04          04.07.83
01 0116 CD0900    LOAD:   CALL  NL
02 0119 11A009    LD      DE,MSG?2
03 011C DF        RST      3
04 011D 11F110    LD      DE,NAME
05 0120 DF        RST      3
06 0121 CDFB04    CALL  C,?RDR
07 0124 3BE1      JR      HL,(EXADR)
08 0126 2A0611    LD      A,H
09 0129 7C        CP      12H
10 012A FE12      JR      C,LOAD-2
11 012C 3BE1      JP      (HL)
12 012E E9
13 012F
14 012F
15 012F
16 012F          GETLINE AND BREAK IN CHECK
17 012F
18 012F          EXIT BREAK IN THEN JUMP (ST1)
19 012F          ACC=TOP OF LINE DATA
20 012F
21 012F          BGETL:  ENT      (SP),HL
22 012F E3        EX      BC
23 0130 C1        POP     DE,BUFER
24 0131 11A311    LD      GETL
25 0134 CD0300    LD      A,(DE)
26 0137 1A        CP      1BH
27 0138 FE1B      JR      Z,LOAD-2
28 013A 2BD3      JP      (HL)
29 013C E9
30 013D
31 013D
32 013D          ASCII TO HEX CONVERT
33 013D          INPUT (DE)=ASCII
34 013D          CY=1 THEN JUMP (ST1)
35 013D          HEXIY:  ENT      (SP),IY
36 013D FDE3      EX      AF
37 013F F1        POP     HL,HEX
38 0140 CD1004    JR      C,LOAD-2
39 0143 3BCA      JP      (IY)
40 0145 FDE9
41 0147
42 0147
43 0147          MSGE1:  ENT      DEFB    0DH
44 0147
45 0147 434B4543    PLOTTER PRINTER TEST COMMAND
46 014B 4B205355    (DPG23)
47 014F 4D204552    %=CONTROL COMMANDS GROUP
48 0153 2E        C=PEN CHENGE
49 0154 0D        G=GRAPH MODE
50 0155          S=80 CHA. IN 1 LINE
51 0155          L=40 CHA. IN 1 LINE
52 0155          T=PLOTTER TEST
53 0155          IN (DE)=PRINT DATA
54 0155
55 0155
56 0155
57 0155
58 0155
59 0155
60 0155

```

```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 05 04.07.83
01 0155 ; PTEST: ENT
02 0156 LD A,(DE)
03 0157 CP &'
04 0158 JZ PTST1
05 0159 INC DE
06 015A LD A,(DE)
07 015B CP &'
08 015C JZ PTST1
09 015D LD A,(DE)
10 015E CP &'
11 015F JZ PTST1
12 0160 INC DE
13 0161 LD A,(DE)
14 0162 CP &'
15 0163 JZ PTST1
16 0164 LD A,(DE)
17 0165 CP &'
18 0166 JZ PTST1
19 0167 LD A,(DE)
20 0168 CP &'
21 0169 JZ PTST1
22 016A LD A,(DE)
23 016B CP &'
24 016C JZ PTST1
25 016D LD A,(DE)
26 016E CP &'
27 016F JZ PTST1
28 0170 LD A,(DE)
29 0171 CP &'
30 0172 JZ PTST1
31 0173 LD A,(DE)
32 0174 CP &'
33 0175 JZ PTST1
34 0176 LD A,(DE)
35 0177 CP &'
36 0178 JZ PTST1
37 0179 LD A,(DE)
38 017A CP &'
39 017B JZ PTST1
40 017C LD A,(DE)
41 017D CP &'
42 017E JZ PTST1
43 017F LD A,(DE)
44 0180 CP &'
45 0181 JZ PTST1
46 0182 LD A,(DE)
47 0183 CP &'
48 0184 JZ PTST1
49 0185 LD A,(DE)
50 0186 CP &'
51 0187 JZ PTST1
52 0188 LD A,(DE)
53 0189 CP &'
54 018A JZ PTST1
55 018B LD A,(DE)
56 018C CP &'
57 018D JZ PTST1
58 018E LD A,(DE)
59 018F CP &'
60 0190 JZ PTST1

; 80 IN 1 LINE
; 80 IN 1 LINE
; PEN CHANGE
; GRAPH MODE
; TEST
; PLOT MESSAGE
; 01-09-09-08-0D
; 01-09-09-09-0D
; TEST PATTERN
; GRAPH CODE
; 1 CHANGE CODE (TEXT MO

PTST1: CALL MSG
JP ST1

.LPT: LD DE,LLPT
JR PTST1

..LPT: LD DE,SLPT
JR PTST1

PTRN: LD A,04H
PLOT+2

PLOT: LD A,02H
CALL LPRINT
JR PTSTO

PEN: LD A,1DH
PLOT+2

; 1CHA. PRINT TO $LPT
; IN: ACC PRINT DATA
LPRINT: LD C,0
LD B,A
CALL RDA
LD A,B
OUT (FEH),A
LD A,BOH
OUT (FEH),A
LD C,01H
CALL RDA
XOR A
OUT (FEH),A
RET

; $LPT MSG.
; IN: DE DATA LOW ADR.
; ODH MSG. END
;

```

```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 06 04.07.83
01 01A5 D5
02 01A6 C5
03 01A7 F5
04 01A8 1A
05 01A9 CDBF01
06 01AC 1A
07 01AD 13
08 01AE FE0D
09 01B0 20F6
10 01B2 F1
11 01B3 C1
12 01B4 D1
13 01B5 C9
14 01B6
15 01B6
16 01B6
17 01B6
18 01B6
19 01B6 DBFE
20 01B6 DBFE
21 01B8 E60D
22 01BA B9
23 01BB C8
24 01BC CD1E00
25 01BF 20F5
26 01C1 31F010
27 01C4 C3AD00
28 01C7
29 01C7
30 01C7
31 01C7
32 01C7
33 01C7
34 01C7
35 01C7
36 01C7
37 01C7
38 01C7
39 01C7 C5
40 01C8 D5
41 01C9 E5
42 01CA 3E02
43 01CC 32A011
44 01CF 0601
45 01D1 1A
46 01D2 FE0D
47 01D4 283B
48 01D6 FECB
49 01D8 2837
50 01DA FEDF
51 01DC 2827
52 01DE FE2D
53 01E0 2823
54 01E2 FE2B
55 01E4 2827
56 01E6 FED7
57 01E8 2823
58 01EA FE23
59 01EC 216C02
60 01EF 2004

MSG: PUSH BC
PUSH AF
PUSH AF
LD LPRINT
LD A,(DE)
INC DE
CP ODH
JR NZ,PMSG1
POP AF
POP BC
POP DE
RET

RDA CHECK
; RDA
; IN: C RDA CODE
; BRKEY IN TO MONITOR RETURN
RDA: IN A,(FE)
AND ODH
CP Z
RET Z
CALL BRKEY
JR NZ,RDA
LD SP,SP
JP ST1

; ORG 01C7H
; MELODY
; DE=DATA LOW ADR.
; EXIT CF=1 BREAK
; CF=0 OK
?MLDY: ENT
PUSH BC
PUSH DE
PUSH HL
LD A,02H
LD (OCTV),A
LD B,01
MLD1: LD A,(DE)
CP ODH
JR Z,MLD4
CP CBH
JR Z,MLD4
CP CFH
JR Z,MLD2
CP 2DH
JR Z,MLD2
CP 2BH
JR Z,MLD3
CP D7H
JR Z,MLD3
CP 23H
LD HL,MTBL
NZ,+6

; CR
; END MARK
; UNDER OCTAVE
;
;
; UPPER OCTAVE
; "H" HANDON

```



```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 07                                04.07.83
01 01F1 21B402 LD HL,M#TBL
02 01F4 13 INC DE
03 01F5 CD1C02 CALL ONPU
04 01F8 3BD7 JR C,MLD1
05 01FA CDCB02 CALL RYTHM
06 01FD 3B15 JR C,MLDS
07 01FF CDAB02 CALL MLDST
08 0202 41 LD B,C
09 0203 1BCC JR MLD1
10 0205 3E03 LD A,+3
11 0207 32A011 LD DE (OCTV),A
12 020A 13 INC DE
13 020B 1BC4 JR MLD1
14 020D 3E01 LD A,1
15 020F 1BF6 JR MLD2+2
16 0211 CDCB02 CALL RYTHM
17 0214 F5 PUSH AF
18 0215 CDBE02 CALL MLDSP
19 0218 F1 POP AF
20 0219 C39B06 JP RET3
; ONPU TO RATIO CONV
21 021C
22 021C
23 021C
24 021C ; EXIT (RATIO)=RATIO VALUE
25 021C ; C=ONTYO*TEMPO
26 021C
27 021C ONPU: ENT
28 021C C5 PUSH BC
29 021F 060B LD B,B
30 0221 1A LD A,(DE)
31 0220 BE CP (HL)
32 0221 2B09 JR Z,ONP2
33 0223 23 INC HL
34 0224 23 INC HL
35 0225 23 INC HL
36 0226 10FB DJNZ -6
37 0228 37 SCF
38 0229 13 INC DE
39 022A C1 POP BC
40 022B C9 RET
41 022C 23 ONP2: INC HL
42 022D D5 PUSH DE
43 022E 5E LD E,(HL)
44 022F 23 INC HL
45 0230 56 LD D,(HL)
46 0231 EB EX DE,HL
47 0232 7C LD A,H
48 0233 B7 OR A
49 0234 2B09 JR Z,+11
50 0236 3AA011 LD A,(OCTV)
51 0239 3D DEC A
52 023A 2B03 JR Z,+5
53 023C 29 ADD HL,HL
54 023D 1BFA JR -4
55 023F 22A011 LD (RATIO),HL
56 0242 21A011 LD HL,OCTV
57 0245 3602 LD (HL),2
58 0247 2B DEC HL
59 0248 D1 POP DE
60 0249 13 INC DE

; 11A0H OCTAVE WORK
; 11A1H ONPU RATIO

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 08                                04.07.83
01 024A 1A LD A,(DE)
02 024B 47 LD B,A
03 024C E6F0 AND FOH
04 024E FE30 CP 30H
05 0250 2B03 JR Z,+5
06 0252 7E LD A,(HL)
07 0253 1B05 JR +7
08 0255 13 INC DE
09 0256 7B LD A,B
10 0257 E60F AND OFH
11 0259 77 LD (HL),A
12 025A 219C02 LD HL,OFTBL
13 025D B5 ADD A,L
14 025E 6F LD L,A
15 025F 4E LD C,(HL)
16 0260 3A9E11 LD A,(TEMPW)
17 0263 47 LD B,A
18 0264 AF XOR A
19 0265 B1 ONP3: ADD A,C
20 0266 10FD DJNZ -1
21 026B C1 POP BC
22 0269 4F LD C,A
23 026A AF XOR A
24 026B C9 RET
25 026C
26 026C ;
27 026C ; MTBL:
28 026C 43 DEFH 43H
29 026D 460B DEFH 0B46H
30 026F 44 DEFH 44H
31 0270 5F07 DEFH 075FH
32 0272 45 DEFH 45H
33 0273 9106 DEFH 0691H
34 0275 46 DEFH 46H
35 0276 3306 DEFH 0633H
36 0278 47 DEFH 47H
37 0279 B605 DEFH 0586H
38 027B 41 DEFH 41H
39 027C EC04 DEFH 04ECH
40 027E 42 DEFH 42H
41 027F 6404 DEFH 0464H
42 0281 52 DEFH 52H
43 0282 0000 DEFH 0
44 0284 MTBL: ENT
45 0284 43 DEFH 43H
46 0285 CF07 DEFH 07CFH
47 0287 44 DEFH 44H
48 0288 F506 DEFH 06F5H
49 028A 45 DEFH 45H
50 028B 3306 DEFH 0633H
51 028D 46 DEFH 46H
52 028E DA05 DEFH 05DAH
53 0290 47 DEFH 47H
54 0291 3705 DEFH 0537H
55 0293 41 DEFH 41H
56 0294 A504 DEFH 04A5H
57 0296 42 DEFH 42H
58 0297 2304 DEFH 0423H
59 0299 52 DEFH 52H
60 029A

```

```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 09                                04.07.83
01 029A 0000                                LD (HL),FBH
02 029C 01                                INC HL
03 029C 01                                LD A,(HL)
04 029D 02                                AND B1H
05 029E 03                                JR NZ,+4
06 029F 04                                SCF
07 02A0 06                                RET
08 02A1 08                                LD A,(TEMP)
09 02A2 0C                                RRCA
10 02A3 10                                JR C,-4
11 02A4 18                                LD A,(TEMP)
12 02A5 20                                RRCA
13 02A6                                JR NC,-4
14 02A6                                DJNZ -12
15 02A6                                XOR A
16 02A6                                RET
17 02A6                                ;
18 02A6                                ;
19 02A6 13                                ; TEMPO SET
20 02A7 13                                INC
21 02AB 13                                ; ACC=VALUE (1-7)
22 02A9 13                                ;
23 02AA C9                                ?TEMP: ENT
24 02AB                                PUSH AF
25 02AB                                PUSH BC
26 02AB                                AND OFH
27 02AB                                LD B,A
28 02AB                                LD A,B
29 02AB                                SUB B
30 02AB                                LD (TEMPW),A
31 02AB                                POP BC
32 02AB 2AA111                            LD A,1
33 02AE 7C                                POP AF
34 02AF B7                                RET
35 02B0 2B0C                            CRT MANAGMENT
36 02B2 D5                                ;
37 02B3 EB                                ;
38 02B4 2104E0                            EXIT HL:DSPLY H=Y,L=X
39 02B7 73                                DE:MANG ADR. (ON DSPXY)
40 02B8 72                                A:MANG DATA
41 02B9 3E01                                CY:MANG=1
42 02BB D1                                ;
43 02BC 1B06                            ; MANG: ENT
44 02BE                                LD HL,MANG
45 02BE                                LD A,(1172H)
46 02BE 3E36                            ADD A,L
47 02C0 3207E0                            LD L,A
48 02C3 AF                                LD A,(HL)
49 02C4 320BE0                            INC HL
50 02C7 C9                                RL (HL)
51 02C8                                OR (HL)
52 02C8                                RR (HL)
53 02C8                                RRCA
54 02C8                                EX DE,HL
55 02C8                                LD HL,(DSPXY)
56 02C8                                RET
57 02C8                                ;
58 02C8                                ;
59 02C8                                ; ORG 030BH
60 02C8 2100E0                            ;

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 10                                04.07.83
01 02CB 36FB                                LD
02 02CD 23                                INC
03 02CE 7E                                LD A,(HL)
04 02CF E6B1                            AND B1H
05 02D1 2002                            JR NZ,+4
06 02D3 37                                SCF
07 02D4 C9                                RET
08 02D5 3A0BE0                            LD A,(TEMP)
09 02D8 0F                                RRCA
10 02D9 3BFA                            JR C,-4
11 02DB 3A0BE0                            LD A,(TEMP)
12 02DE 0F                                RRCA
13 02DF 30FA                            JR NC,-4
14 02E1 10F2                            DJNZ -12
15 02E3 AF                                XOR A
16 02E4 C9                                RET
17 02E5                                ;
18 02E5                                ;
19 02E5                                ; TEMPO SET
20 02E5                                INC
21 02E5                                ; ACC=VALUE (1-7)
22 02E5                                ;
23 02E5                                ?TEMP: ENT
24 02E5 F5                                PUSH AF
25 02E6 C5                                PUSH BC
26 02E7 E60F                            AND OFH
27 02E9 47                                LD B,A
28 02EA 3E08                            LD A,B
29 02EC 90                                SUB B
30 02ED 329E11                            LD (TEMPW),A
31 02F0 C1                                POP BC
32 02F1 F1                                POP AF
33 02F2 C9                                RET
34 02F3                                ;
35 02F3                                ; CRT MANAGMENT
36 02F3                                ;
37 02F3                                ;
38 02F3                                ; EXIT HL:DSPLY H=Y,L=X
39 02F3                                DE:MANG ADR. (ON DSPXY)
40 02F3                                A:MANG DATA
41 02F3                                CY:MANG=1
42 02F3                                ; MANG: ENT
43 02F3 217311                            LD HL,MANG
44 02F6 3A7211                            LD A,(1172H)
45 02F9 85                                ADD A,L
46 02FA 6F                                LD L,A
47 02FB 7E                                LD A,(HL)
48 02FC 23                                INC HL
49 02FD CB16                            RL (HL)
50 02FF B6                                OR (HL)
51 0300 CB1E                            RR (HL)
52 0302 0F                                RRCA
53 0303 EB                                EX DE,HL
54 0304 2A7111                            LD HL,(DSPXY)
55 0307 C9                                RET
56 0308                                ;
57 0308                                ;
58 0308                                ; ORG 030BH
59 0308                                ;
60 0308                                ;

```

```

** Z80 ASSEMBLER SB-7201 <12-013A> PAGE 11 04.07.83
01 0308 ; TIME SET
02 0308 ;
03 0308 ; ACC=0 : AM
04 0308 ; =1 : PM
05 0308 ; DE=SEC: BINARY
06 0308 ;
07 0308 ?TMST: ENT
08 0308 F3 DI
09 0309 C5 PUSH BC
10 030A D5 PUSH DE
11 030B E5 PUSH HL
12 030C 329B11 LD (AMPM),A
13 030F 3EFO LD A,FOH
14 0311 329C11 LD (TIMEG),A
15 0314 21COAB LD HL,ABCOH
16 0317 AF XOR A
17 0318 ED52 SBC HL,DE
18 031A E5 TA
19 031B 00 PUSH HL
20 031C EB EX
21 031D 2107E0 LD HL,CONTF
22 0320 3674 LD (HL),74H
23 0322 36B0 LD (HL),BOH
24 0324 2B DEC HL
25 0325 73 LD (HL),E
26 0326 72 LD (HL),D
27 0327 2B DEC HL
28 0328 360A LD (HL),OAH
29 032A 3600 LD (HL),0
30 032C 23 INC HL
31 032D 23 INC HL
32 032E 3680 LD (HL),80H
33 0330 2B DEC HL
34 0331 4E LD C,(HL)
35 0332 7E LD A,(HL)
36 0333 BA CP D
37 0334 20FB JR NZ,?TMS1
38 0336 79 LD A,C
39 0337 BB CP E
40 0338 20F7 JR NZ,?TMS1
41 033A 2B DEC HL
42 033B 00 NOP
43 033C 00 NOP
44 033D 00 NOP
45 033E 36FB LD (HL),FBH
46 0340 363C LD (HL),3CH
47 0342 23 INC HL
48 0343 D1 POP DE
49 0344 4E LD C,(HL)
50 0345 7E LD A,(HL)
51 0346 BA CP D
52 0347 20FB JR NZ,?TMS2
53 0349 79 LD A,C
54 034A BB CP E
55 034B 20F7 JR NZ,?TMS2
56 034D E1 POP HL
57 034E D1 POP DE
58 034F C1 POP BC
59 0350 FB EI
60 0351 C9 RET

; TIME SET
;
; ACC=0 : AM
; =1 : PM
; DE=SEC: BINARY
;
; TMST: ENT
; DI
; PUSH BC
; PUSH DE
; PUSH HL
; LD (AMPM),A
; LD A,FOH
; LD (TIMEG),A
; LD HL,ABCOH
; XOR A
; SBC HL,DE
; TA
; PUSH HL
; EX
; LD HL,CONTF
; LD (HL),74H
; LD (HL),BOH
; DEC HL
; LD (HL),E
; LD (HL),D
; DEC HL
; LD (HL),OAH
; LD (HL),0
; INC HL
; INC HL
; LD (HL),80H
; DEC HL
; LD C,(HL)
; LD A,(HL)
; CP D
; JR NZ,?TMS1
; LD A,C
; CP E
; JR NZ,?TMS1
; DEC HL
; NOP
; NOP
; NOP
; LD (HL),FBH
; LD (HL),3CH
; INC HL
; POP DE
; LD C,(HL)
; LD A,(HL)
; CP D
; JR NZ,?TMS2
; LD A,C
; CP E
; JR NZ,?TMS2
; POP HL
; POP DE
; POP BC
; EI
; RET

** Z80 ASSEMBLER SB-7201 <12-013A> PAGE 12 04.07.83
01 0352 ;
02 0352 ; BELL DATA
03 0352 ;
04 0352 ;
05 0352 ?BELD: ENT
06 0352 D7H DEFB
07 0353 4130 DEFB 'A0'
08 0355 0DH DEFB 0DH
09 0356 ;
10 0356 ;
11 0356 ;
12 0356 ; DEFS +2
13 0358 ; ORG 035BH
14 0358 ;
15 0358 ; TIME READ
16 0358 ;
17 0358 ; EXIT ACC=0 :AM
18 0358 ; =1 :PM
19 0358 ; DE=SEC. BINARY
20 0358 ;
21 0358 ?TMRD: ENT
22 0358 E5 PUSH HL
23 0359 2107E0 LD HL,CONTF
24 035C 3680 LD (HL),80H
25 035E 2B DEC HL
26 035F F3 DI
27 0360 5E LD E,(HL)
28 0361 56 LD D,(HL)
29 0362 FB EI
30 0363 7B LD D
31 0364 B2 OR D
32 0365 280E JR Z,?TMR1
33 0367 AF XOR A
34 0368 21COAB LD HL,ABCOH
35 036B ED52 SBC HL,DE
36 036D 3810 JR C,?TMR2
37 036F EB EX DE,HL
38 0370 3A9B11 LD A,(AMPH)
39 0373 E1 POP HL
40 0374 C9 RET
41 0375 11COAB ?TMR1: LD DE,ABCOH
42 0378 3A9B11 LD A,(AMPH)
43 037B EE01 XOR 1
44 037D E1 POP HL
45 037E C9 RET
46 037F F3 DI
47 0380 2106E0 ?TMR2: LD HL,CONTF
48 0383 7E LD A,(HL)
49 0384 2F CPL
50 0385 5F LD E,A
51 0386 7E LD A,(HL)
52 0387 2F CPL
53 0388 57 LD D,A
54 0389 FB EI
55 038A 13 INC DE
56 038B 18EB JR ?TMR1+3
57 038D ;
58 038D ; TIME INTERRUPT
59 038D ;
60 038D ; TIMIN: ENT

```

```

01 038D F5      PUSH AF
02 038E C5      PUSH BC
03 038F D5      PUSH DE
04 0390 E5      HL,AMPM
05 0391 219B11 LD A,(HL)
06 0394 7E      XOR 1
07 0395 EE01    LD (HL),A
08 0397 77      LD HL,CONTF
09 0398 2107E0 LD (HL),80H
10 039B 3680    DEC HL
11 039D 2B      PUSH HL
12 039E E5      LD E,(HL)
13 039F 5E      LD D,(HL)
14 03A0 56      LD HL,ABCOH
15 03A1 21C0AB LD HL,DE
16 03A4 19      ADD HL,DE
17 03A5 2B      DEC HL
18 03A6 2B      EX DE,HL
19 03A7 EB      POP HL
20 03A8 E1      LD (HL),E
21 03A9 73      LD (HL),D
22 03AA 72      POP HL
23 03AB E1      POP DE
24 03AC D1      POP BC
25 03AD C1      POP AF
26 03AE F1      EI
27 03AF FB      RET
28 03B0 C9      ;
29 03B1          ; SPACE PRINT AND DISP ACC
30 03B1          ;
31 03B1          ;
32 03B1          ; INPUT:HL=DISP. ADR.
33 03B1          ;
34 03B1          SPHEX: ENT CALL ?PTS
35 03B1 CD2009  LD A,(HL)
36 03B4 7E      CALL PRTHX
37 03B5 CDC303 LD A,(HL)
38 03B8 7E      RET
39 03B9 C9      ;
40 03BA          ;
41 03BA          ;
42 03BA          ; ORG 03BAH
43 03BA          ;
44 03BA          ; (ASCII PRINT) FOR HL
45 03BA          ;
46 03BA          PRTHL: ENT
47 03BA 7C      LD A,H
48 03BA 7C      CALL PRTHX
49 03BB CDC303 LD A,L
50 03BE 7D      JR PRTHX
51 03BF 1B02    ;
52 03C1          ;
53 03C1          ; ORG 03C3H:PRTHX
54 03C3          ;
55 03C3          ; (ASCII PRINT) FOR ACC
56 03C3          ;
57 03C3          PRTHX: ENT
58 03C3          PUSH AF
59 03C3 F5      RRCA
60 03C4 0F      RRCA
01 03C5 0F      RRCA
02 03C6 0F      RRCA
03 03C7 0F      RRCA
04 03C8 CDDA03  CALL ASC
05 03CB CD1200  CALL PRNT
06 03CE F1      POP AF
07 03CF CDDA03  CALL ASC
08 03D2 C31200  JP PRNT
09 03D5          ;
10 03D5          ;
11 03D5          ;
12 03D5          ; 80 CHA. 1 LINE CODE (DATA)
13 03D5          ;
14 03D5          ;
15 03D5          ; SLPT: ENT
16 03D5 01      DEFB 01H
17 03D6 09      DEFB 09H
18 03D7 09      DEFB 09H
19 03D8 09      DEFB 09H
20 03D9 0D      DEFB 0DH
21 03DA          ;
22 03DA          ; ORG 03DAH:ASC
23 03DA          ;
24 03DA          ; HEXADECIIMAL TO ASCII
25 03DA          ; IN: ACC (D3-D0)=HEXADECIIMAL
26 03DA          ; EXIT: ACC = ASCII
27 03DA          ;
28 03DA          ; ASC: ENT
29 03DA E60F    AND 0FH
30 03DC FE0A    CP 0AH
31 03DE 3B02    JR C,NOADD
32 03E0 C607    ADD A,7
33 03E2          NOADD: ENT
34 03E2 C630    ADD A,30H
35 03E4 C9      RET
36 03E5          ;
37 03E5          ; ASCII TO HEXADECIIMAL
38 03E5          ; IN: ACC = ASCII
39 03E5          ; EXIT: ACC = HEXADECIIMAL
40 03E5          ; CY = 1 ERROR
41 03E5          ;
42 03E5          ; HEXJ: ENT
43 03E5 D630    SUB 30H
44 03E7 DB      RET C
45 03E8 FE0A    CP 0AH
46 03EA 3F      CCF
47 03EB D0      RET NC
48 03EC D607    SUB 7
49 03EE FE10    CP 10H
50 03F0 3F      CCF
51 03F1 DB      RET C
52 03F2 FE0A    CP 0AH
53 03F4 C9      RET
54 03F5          ;
55 03F5          ;
56 03F5          ; ORG 03F9H:HEX
57 03F9          ; HEX: ENT
58 03F9 18EA    JR HEXJ
59 03F9 18EA    JR HEXJ
60 03FB          ;

```

04.07.83

PAGE 16

04.07.93

-177

```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 19 04.07.83
01 050E ; IN BC=SIZE
02 050E ; DE=LOAD ADR.
03 050E ;
04 050E ; EXIT ACC=0 : OK CF=0
05 050E ; =1 : ER =1
06 050E ; =2 : BREAK=1
07 050E ;
08 050E RTAPE: ENT DE
09 050E D5 PUSH BC
10 050F CS PUSH HL
11 0510 E5 LD H,2
12 0511 2602 LD H,2
13 0513 RTP1: ENT BC,KEYPB
14 0513 0101E0 LD DE,CSTR
15 0516 1102E0 LD
16 0519 RTP2: ENT
17 0519 CD0106 CALL
18 051C 3854 JR C,RTP6
19 051E CD440A CALL DLY3
20 0521 1A LD A,(DE)
21 0522 E620 AND 20H
22 0524 CA1905 JP Z,RTP2
23 0527 54 LD D,H
24 052B 210000 LD HL,0
25 052B 229711 LD HL,(SUMDT),HL
26 052E E1 POP HL
27 052F C1 POP BC
28 0530 C5 PUSH BC
29 0531 E5 PUSH HL
30 0532 RTP3: ENT
31 0532 CD2406 CALL
32 0535 383B JR C,RTP6
33 0537 77 LD HL,(HL),A
34 053B 23 INC HL
35 0539 0B DEC BC
36 053A 7B LD A,B
37 053B B1 OR C
38 053C 20F4 JR NZ,RTP3
39 053E 2A9711 LD HL,(SUMDT)
40 0541 CD2406 CALL RBYTE
41 0544 382C JR C,RTP6
42 0546 5F LD E,A
43 0547 CD2406 CALL RBYTE
44 054A 3826 JR C,RTP6
45 054C BD CP L
46 054D 2016 JR NZ,RTP5
47 054F 7B LD A,E
48 0550 BC CP H
49 0551 2012 JR NZ,RTP5
50 0553 RTPB: ENT
51 0553 AF XOR A
52 0554 RTP4: ENT
53 0554 RET2: ENT
54 0554 E1 HL
55 0555 C1 POP BC
56 0556 D1 POP DE
57 0557 CD0007 CALL MSTOP
58 055A F5 AF
59 055B 3A9C11 LD A,(TIMEG)
60 055E FEF0 CP F0H

; INT. CHECK
01 0560 2001
02 0562 FB JR NZ,+3
03 0563 F1 POP AF
04 0564 C9 RET
05 0565 ;
06 0565 RTP5: ENT
07 0565 15 DEC D
08 0566 2806 JR Z,RTP7
09 0568 62 LD H,D
10 0569 CDE20F GAPCK
11 056C 1BA5 RTP1: JR
12 056E ENT
13 056E 3E01 LD A,1
14 0570 1802 RTP9: JR
15 0572 ENT
16 0572 3E02 LD A,2
17 0574 RTP9: ENT
18 0574 37 SCF
19 0575 1BDD JR RTP4
20 0577 ;
21 0577 ; BELL
22 0577 ;
23 0577 ;
24 0577 ?BEL: ENT
25 0577 D5 PUSH DE
26 0578 115203 LD DE,?BELD
27 057B F7 RST 6
28 057C D1 POP DE
29 057D C9 RET
30 057E ;
31 057E ; FLASING AND KEYIN
32 057E ; EXIT:ACC INPUT KEY DATA(DSP.CODE)
33 057E ; H=FOH THEN NO KEYIN(Z FLG.)
34 057E ;
35 057E FLKEY: ENT
36 057E CDEF09 CALL ?FLAS
37 0581 CDCA0B CALL ?KEY
38 0584 FEF0 CP F0H
39 0586 C9 RET
40 0587 ;
41 0587 ;
42 0587 ;
43 0587 ;
44 0587 ;
45 0587 ;
46 0588 ; ORG 058BH
47 0588 ;
48 0588 ;
49 0588 ; VERIFY (FROM $DMT)
50 0588 ;
51 0588 ; EXIT ACC =0 : OK CF=0
52 0588 ; =1 : ER CF=1
53 0588 ; =2 : BREAK CF=1
54 0588 ;
55 0588 ?VRFY: ENT
56 0588 F3 DI
57 0589 D5 PUSH DE
58 058A C5 BC
59 058B E5 PUSH HL
60 058C ED4B0211 LD BC,(SIZE)

```

04.07.83

PAGE 20

```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 21                                04.07.83
01 0590 2A0411      LD HL,(DTADR)
02 0593 16D2      LD D,D2H
03 0595 1E53      LD E,53H
04 0597 7B      LD A,B
05 0598 B1      OR C
06 0599 28B9      JR Z,RTP4
07 059B CD1A07      CALL CKSUM
08 059E CD9F06      CALL MOTOR
09 05A1 3BCF      JR C,RTP6
10 05A3 CDB06      CALL TMRK
11 05A6 3BCA      JR C,RTP6
12 05AB CDAD05      CALL TVRFY
13 05AB 18A7      JR RTP4
14 05AD
15 05AD
16 05AD
17 05AD
18 05AD
19 05AD
20 05AD
21 05AD
22 05AD
23 05AD
24 05AD
25 05AD
26 05AD
27 05AD D5      TVRFY: ENT DE
28 05AE C5      PUSH BC
29 05AF E5      PUSH HL
30 05B0 2602      LD H,2
31 05B2      ENT
32 05B2 0101E0      LD BC,KEYPB
33 05B5 1102E0      LD DE,CSTR
34 05B8      ENT
35 05BB CD0106      CALL EDGE
36 05BB DA7205      JP C,RTP6
37 05BE CD4A0A      CALL DLY3
38 05C1 1A      LD A,(DE)
39 05C2 E620      AND 20H
40 05C4 CAB805      JP Z,TVF2
41 05C7 54      LD ,H
42 05C8 E1      HL
43 05C9 C1      POP BC
44 05CA C5      PUSH BC
45 05CB E5      PUSH HL
46 05CC      ENT
47 05CC CD2406      CALL RBYTE
48 05CF 3BA1      JR C,RTP6
49 05D1 BE      CP (HL)
50 05D2 209A      JR NZ,RTP7
51 05D4 23      INC HL
52 05D5 0B      DEC BC
53 05D6 7B      LD A,B
54 05D7 B1      OR C
55 05D8 20F2      JR NZ,TVF3
56 05DA 2A9911      LD HL,(CSMDT)
57 05DD CD2406      CALL RBYTE
58 05E0 BC      CP H
59 05E1 20B8      JR NZ,RTP7
60 05E3 CD2406      CALL RBYTE

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 22                                04.07.83
01 05E6 BD      CP L
02 05E7 20B5      JR NZ,RTP7
03 05E9 15      DEC D
04 05EA CA5305      JP Z,RTP8
05 05ED 62      LD H,D
06 05EE 18C2      JR TVF1
07 05F0      ; FLASHING DATA LOAD
08 05F0      ;
09 05F0      ;
10 05F0      ?LOAD: ENT
11 05F0 F5      PUSH AF
12 05F1 3ABE11      LD A,(FLASH)
13 05F4 CDB10F      CALL ?PONT
14 05F7 77      LD (HL),A
15 05F8 F1      POP AF
16 05F9 C9      RET
17 05FA
18 05FA
19 05FA
20 05FA
21 05FA
22 05FA CD0900      NLPHL: ENT NL
23 05FD CDBA03      CALL PRTHL
24 0600 C9      RET
25 0601
26 0601
27 0601
28 0601
29 0601
30 0601
31 0601
32 0601
33 0601
34 0601
35 0601
36 0601
37 0601 3EFB      EDGE: ENT
38 0603 3200E0      LD A,FBH
39 0606 00      LD (KEYPA),A
40 0607      NOP
41 0607 0A      ENT
42 0608 E6B1      LD A,(BC)
43 060A 2002      AND B1H
44 060C 37      JR NZ,+4
45 060D C9      SCF
46 060E 1A      RET
47 060F E620      LD A,(DE)
48 0611 20F4      AND 20H
49 0613      JR NZ,EDG1
50 0613 0A      ENT
51 0614 E6B1      LD A,(BC)
52 0616 2002      AND B1H
53 0618 37      JR NZ,+4
54 0619 C9      SCF
55 061A 1A      RET
56 061B E620      LD A,(DE)
57 061D 2BF4      AND 20H
58 061F C9      JR Z,EDG2
59 0620      RET
60 0620      ;

```



```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 23          04.07.83
01 0620      DEFS +4
02 0624      :ORG 0624H:RBYTE
03 0624      ;
04 0624      ;
05 0624      ; 1 BYTE READ
06 0624      ;
07 0624      : EXIT SUMDT=STORE
08 0624      : CF=1 : BREAK
09 0624      ;
10 0624      : CF=0 : DATA=ACC
11 0624      ;
12 0624      RBYTE: ENT
13 0624      PUSH BC
14 0625      PUSH DE
15 0626      PUSH HL
16 0627      LD HL,0800H
17 0628      LD BC,KEYPB
18 062D      LD DE,CSTR
19 0630      ENT
20 0630      CALL
21 0633      JP C,RBY3
22 0636      CALL DLY3
23 0639      LD A,(DE)
24 063A      AND 20H
25 063C      JP Z,RBY2
26 063F      PUSH HL
27 0640      LD HL,(SUMDT)
28 0643      INC HL
29 0644      LD HL,(SUMDT),HL
30 0647      POP HL
31 0648      SCF
32 0649      ENT
33 0649      LD A,L
34 064A      RLA
35 064B      LD L,A
36 064C      DEC H
37 064D      JP NZ,RBY1
38 0650      CALL EDGE
39 0653      LD A,L
40 0654      ENT
41 0654      POP HL
42 0655      POP DE
43 0656      POP BC
44 0657      RET
45 0658      ;
46 0658      ; TAPE MARK DETECT
47 0658      ;
48 0658      ; E=0L0 : INFORMATION
49 0658      ; =0S0 : DATA
50 0658      ; EXIT CF=0 :OK
51 0658      ; =1 :BREAK
52 0658      ;
53 0658      ; DEFS +3
54 0658      ;
55 0658      ; TMARK: ENT
56 0658      ;
57 0658      ; :ORG 065BH
58 0658      ;
59 0658      ; CALL GAPCK
60 0658      CDE20F

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 24          04.07.83
01 065E      PUSH BC
02 065F      DE
03 0660      PUSH HL
04 0661      LD HL,282BH
05 0664      LD A,E
06 0665      CP CCH
07 0667      JR Z,+5
08 0669      LD HL,1414H
09 066C      LD HL,(TMCNT),HL
10 066F      LD BC,KEYPB
11 0672      LD DE,CSTR
12 0675      TM1: ENT
13 0675      LD HL,(TMCNT)
14 0678      TM2: ENT
15 0678      CALL
16 067B      LD HL,0800H
17 067D      LD BC,KEYPB
18 0680      LD DE,CSTR
19 0681      E620
20 0683      28F0
21 0685      25
22 0686      20F0
23 0688      20F0
24 068B      CD0106
25 068B      380E
26 068D      CD4A0A
27 0690      1A
28 0691      E620
29 0693      20E0
30 0695      2D
31 0696      20F0
32 0698      CD0106
33 0698      ENT
34 069B      ENT
35 069B      E1
36 069C      D1
37 069D      C1
38 069E      C9
39 069F      ;
40 069F      ; MOTOR ON
41 069F      ;
42 069F      ; IN D=0M0 :WRITE
43 069F      ; =0R0 :READ
44 069F      ; EXIT CF=0 :OK
45 069F      ; =1 :BREAK
46 069F      ; MOTOR: ENT
47 069F      PUSH BC
48 069F      C5
49 06A0      D5
50 06A1      E5
51 06A2      060A
52 06A4      3A02E0
53 06A4      LD 10H
54 06A7      E610
55 06A9      280E
56 06AB      JR Z,MOT4
57 06AB      06FF
58 06AD      CD9609
59 06B0      1802
60 06B2      18E8

; 2 SEC DELAY
; 7 MSEC DELAY
; MOTOR ENTRY ADJUST
; ORG 06B2H

```

```

01 070A 280B      JR      Z,MST3
02 070C 3E06      LD      A,06H
03 070E 3203E0    LD      (CSTPT),A
04 0711 3C        INC      A
05 0712 3203E0    LD      (CSTPT),A
06 0715 10EE      DJNZ     MST1
07 0717 C3E60E    JP      ?RSTR1
08 071A          ;
09 071A          ;
10 071A          ;
11 071A          ;
12 071A          ;
13 071A          ;
14 071A          ; CHECK SUM
15 071A          ;
16 071A          ; IN BC=SIZE
17 071A          ; HL=DATA ADR.
18 071A          ; EXIT SUMDT=STORE
19 071A          ; CSMDT=STORE
20 071A          ;
21 071A          ; CKSUM: ENT
22 071A C5        PUSH     BC
23 071B D5        PUSH     DE
24 071C E5        PUSH     HL
25 071D 110000    LD      DE,0
26 0720          LD      A,B
27 0721 B1        OR       C
28 0722 200B      JR      NZ,CKS2
29 0724 EB        EX      DE,HL
30 0725 229711    LD      (SUMDT),HL
31 0728 E1        LD      (CSMDT),HL
32 072B E1        POP      HL
33 072C D1        POP      DE
34 072D C1        POP      BC
35 072E C9        RET
36 072F          LD      A,(HL)
37 0730 C5        PUSH     BC
38 0731 060B      LD      B,+B
39 0733          LD      A,(HL)
40 0734 3001      JR      NC,+3
41 0735          INC      DE
42 0736 13        DJNZ     CKS3
43 0737 10FA      POP      BC
44 0738 23        INC      HL
45 0739 0B        DEC      BC
46 073A          JR      CKS1
47 073B          ;
48 073C 1BE2      ; MODE SET OF KEYPORT
49 073D          ;
50 073E          ;
51 073F          ;
52 0740          ;
53 0741          ;
54 0742          ;
55 0743          ;
56 0744          ;
57 0745          ;
58 0746          ;
59 0747          ;
60 0748          ;
61 0749          ;
62 074A          ;
63 074B          ;
64 074C          ;
65 074D          ;
66 074E          ;
67 074F          ;
68 0750          ;
69 0751          ;
70 0752          ;
71 0753          ;
72 0754          ;
73 0755          ;
74 0756          ;
75 0757          ;
76 0758          ;
77 0759          ;
78 075A          ;
79 075B          ;
80 075C          ;
81 075D          ;
82 075E          ;
83 075F          ;
84 0760          ;
85 0761          ;
86 0762          ;
87 0763          ;
88 0764          ;
89 0765          ;
90 0766          ;
91 0767          ;
92 0768          ;
93 0769          ;
94 076A          ;
95 076B          ;
96 076C          ;
97 076D          ;
98 076E          ;
99 076F          ;
100 0770          ;
101 0771          ;
102 0772          ;
103 0773          ;
104 0774          ;
105 0775          ;
106 0776          ;
107 0777          ;
108 0778          ;
109 0779          ;
110 077A          ;
111 077B          ;
112 077C          ;
113 077D          ;
114 077E          ;
115 077F          ;
116 0780          ;
117 0781          ;
118 0782          ;
119 0783          ;
120 0784          ;
121 0785          ;
122 0786          ;
123 0787          ;
124 0788          ;
125 0789          ;
126 078A          ;
127 078B          ;
128 078C          ;
129 078D          ;
130 078E          ;
131 078F          ;
132 0790          ;
133 0791          ;
134 0792          ;
135 0793          ;
136 0794          ;
137 0795          ;
138 0796          ;
139 0797          ;
140 0798          ;
141 0799          ;
142 079A          ;
143 079B          ;
144 079C          ;
145 079D          ;
146 079E          ;
147 079F          ;
148 07A0          ;
149 07A1          ;
150 07A2          ;
151 07A3          ;
152 07A4          ;
153 07A5          ;
154 07A6          ;
155 07A7          ;
156 07A8          ;
157 07A9          ;
158 07AA          ;
159 07AB          ;
160 07AC          ;
161 07AD          ;
162 07AE          ;
163 07AF          ;
164 07B0          ;
165 07B1          ;
166 07B2          ;
167 07B3          ;
168 07B4          ;
169 07B5          ;
170 07B6          ;
171 07B7          ;
172 07B8          ;
173 07B9          ;
174 07BA          ;
175 07BB          ;
176 07BC          ;
177 07BD          ;
178 07BE          ;
179 07BF          ;
180 07C0          ;
181 07C1          ;
182 07C2          ;
183 07C3          ;
184 07C4          ;
185 07C5          ;
186 07C6          ;
187 07C7          ;
188 07C8          ;
189 07C9          ;
190 07CA          ;
191 07CB          ;
192 07CC          ;
193 07CD          ;
194 07CE          ;
195 07CF          ;
196 07D0          ;
197 07D1          ;
198 07D2          ;
199 07D3          ;
200 07D4          ;
201 07D5          ;
202 07D6          ;
203 07D7          ;
204 07D8          ;
205 07D9          ;
206 07DA          ;
207 07DB          ;
208 07DC          ;
209 07DD          ;
210 07DE          ;
211 07DF          ;
212 07E0          ;
213 07E1          ;
214 07E2          ;
215 07E3          ;
216 07E4          ;
217 07E5          ;
218 07E6          ;
219 07E7          ;
220 07E8          ;
221 07E9          ;
222 07EA          ;
223 07EB          ;
224 07EC          ;
225 07ED          ;
226 07EE          ;
227 07EF          ;
228 07F0          ;
229 07F1          ;
230 07F2          ;
231 07F3          ;
232 07F4          ;
233 07F5          ;
234 07F6          ;
235 07F7          ;
236 07F8          ;
237 07F9          ;
238 07FA          ;
239 07FB          ;
240 07FC          ;
241 07FD          ;
242 07FE          ;
243 07FF          ;
244 0800          ;
245 0801          ;
246 0802          ;
247 0803          ;
248 0804          ;
249 0805          ;
250 0806          ;
251 0807          ;
252 0808          ;
253 0809          ;
254 080A          ;
255 080B          ;
256 080C          ;
257 080D          ;
258 080E          ;
259 080F          ;
260 0810          ;
261 0811          ;
262 0812          ;
263 0813          ;
264 0814          ;
265 0815          ;
266 0816          ;
267 0817          ;
268 0818          ;
269 0819          ;
270 081A          ;
271 081B          ;
272 081C          ;
273 081D          ;
274 081E          ;
275 081F          ;
276 0820          ;
277 0821          ;
278 0822          ;
279 0823          ;
280 0824          ;
281 0825          ;
282 0826          ;
283 0827          ;
284 0828          ;
285 0829          ;
286 082A          ;
287 082B          ;
288 082C          ;
289 082D          ;
290 082E          ;
291 082F          ;
292 0830          ;
293 0831          ;
294 0832          ;
295 0833          ;
296 0834          ;
297 0835          ;
298 0836          ;
299 0837          ;
300 0838          ;
301 0839          ;
302 083A          ;
303 083B          ;
304 083C          ;
305 083D          ;
306 083E          ;
307 083F          ;
308 0840          ;
309 0841          ;
310 0842          ;
311 0843          ;
312 0844          ;
313 0845          ;
314 0846          ;
315 0847          ;
316 0848          ;
317 0849          ;
318 084A          ;
319 084B          ;
320 084C          ;
321 084D          ;
322 084E          ;
323 084F          ;
324 0850          ;
325 0851          ;
326 0852          ;
327 0853          ;
328 0854          ;
329 0855          ;
330 0856          ;
331 0857          ;
332 0858          ;
333 0859          ;
334 085A          ;
335 085B          ;
336 085C          ;
337 085D          ;
338 085E          ;
339 085F          ;
340 0860          ;
341 0861          ;
342 0862          ;
343 0863          ;
344 0864          ;
345 0865          ;
346 0866          ;
347 0867          ;
348 0868          ;
349 0869          ;
350 086A          ;
351 086B          ;
352 086C          ;
353 086D          ;
354 086E          ;
355 086F          ;
356 0870          ;
357 0871          ;
358 0872          ;
359 0873          ;
360 0874          ;
361 0875          ;
362 0876          ;
363 0877          ;
364 0878          ;
365 0879          ;
366 087A          ;
367 087B          ;
368 087C          ;
369 087D          ;
370 087E          ;
371 087F          ;
372 0880          ;
373 0881          ;
374 0882          ;
375 0883          ;
376 0884          ;
377 0885          ;
378 0886          ;
379 0887          ;
380 0888          ;
381 0889          ;
382 088A          ;
383 088B          ;
384 088C          ;
385 088D          ;
386 088E          ;
387 088F          ;
388 0890          ;
389 0891          ;
390 0892          ;
391 0893          ;
392 0894          ;
393 0895          ;
394 0896          ;
395 0897          ;
396 0898          ;
397 0899          ;
398 089A          ;
399 089B          ;
400 089C          ;
401 089D          ;
402 089E          ;
403 089F          ;
404 08A0          ;
405 08A1          ;
406 08A2          ;
407 08A3          ;
408 08A4          ;
409 08A5          ;
410 08A6          ;
411 08A7          ;
412 08A8          ;
413 08A9          ;
414 08AA          ;
415 08AB          ;
416 08AC          ;
417 08AD          ;
418 08AE          ;
419 08AF          ;
420 08B0          ;
421 08B1          ;
422 08B2          ;
423 08B3          ;
424 08B4          ;
425 08B5          ;
426 08B6          ;
427 08B7          ;
428 08B8          ;
429 08B9          ;
430 08BA          ;
431 08BB          ;
432 08BC          ;
433 08BD          ;
434 08BE          ;
435 08BF          ;
436 08C0          ;
437 08C1          ;
438 08C2          ;
439 08C3          ;
440 08C4          ;
441 08C5          ;
442 08C6          ;
443 08C7          ;
444 08C8          ;
445 08C9          ;
446 08CA          ;
447 08CB          ;
448 08CC          ;
449 08CD          ;
450 08CE          ;
451 08CF          ;
452 08D0          ;
453 08D1          ;
454 08D2          ;
455 08D3          ;
456 08D4          ;
457 08D5          ;
458 08D6          ;
459 08D7          ;
460 08D8          ;
461 08D9          ;
462 08DA          ;
463 08DB          ;
464 08DC          ;
465 08DD          ;
466 08DE          ;
467 08DF          ;
468 08E0          ;
469 08E1          ;
470 08E2          ;
471 08E3          ;
472 08E4          ;
473 08E5          ;
474 08E6          ;
475 08E7          ;
476 08E8          ;
477 08E9          ;
478 08EA          ;
479 08EB          ;
480 08EC          ;
481 08ED          ;
482 08EE          ;
483 08EF          ;
484 08F0          ;
485 08F1          ;
486 08F2          ;
487 08F3          ;
488 08F4          ;
489 08F5          ;
490 08F6          ;
491 08F7          ;
492 08F8          ;
493 08F9          ;
494 08FA          ;
495 08FB          ;
496 08FC          ;
497 08FD          ;
498 08FE          ;
499 08FF          ;
500 0900          ;
501 0901          ;
502 0902          ;
503 0903          ;
504 0904          ;
505 0905          ;
506 0906          ;
507 0907          ;
508 0908          ;
509 0909          ;
510 090A          ;
511 090B          ;
512 090C          ;
513 090D          ;
514 090E          ;
515 090F          ;
516 0910          ;
517 0911          ;
518 0912          ;
519 0913          ;
520 0914          ;
521 0915          ;
522 0916          ;
523 0917          ;
524 0918          ;
525 0919          ;
526 091A          ;
527 091B          ;
528 091C          ;
529 091D          ;
530 091E          ;
531 091F          ;
532 0920          ;
533 0921          ;
534 0922          ;
535 0923          ;
536 0924          ;
537 0925          ;
538 0926          ;
539 0927          ;
540 0928          ;
541 0929          ;
542 092A          ;
543 092B          ;
544 092C          ;
545 092D          ;
546 092E          ;
547 092F          ;
548 0930          ;
549 0931          ;
550 0932          ;
551 0933          ;
552 0934          ;
553 0935          ;
554 0936          ;
555 0937          ;
556 0938          ;
557 0939          ;
558 093A          ;
559 093B          ;
560 093C          ;
561 093D          ;
562 093E          ;
563 093F          ;
564 0940          ;
565 0941          ;
566 0942          ;
567 0943          ;
568 0944          ;
569 0945          ;
570 0946          ;
571 0947          ;
572 0948          ;
573 0949          ;
574 094A          ;
575 094B          ;
576 094C          ;
577 094D          ;
578 094E          ;
579 094F          ;
580 0950          ;
581 0951          ;
582 0952          ;
583 0953          ;
584 0954          ;
585 0955          ;
586 0956          ;
587 0957          ;
588 0958          ;
589 0959          ;
590 095A          ;
591 095B          ;
592 095C          ;
593 095D          ;
594 095E          ;
595 095F          ;
596 0960          ;
597 0961          ;
598 0962          ;
599 0963          ;
600 0964          ;
601 0965          ;
602 0966          ;
603 0967          ;
604 0968          ;
605 0969          ;
606 096A          ;
607 096B          ;
608 096C          ;
609 096D          ;
610 096E          ;
611 096F          ;
612 0970          ;
613 0971          ;
614 0972          ;
615 0973          ;
616 0974          ;
617 0975          ;
618 0976          ;
619 0977          ;
620 0978          ;
621 0979          ;
622 097A          ;
623 097B          ;
624 097C          ;
625 097D          ;
626 097E          ;
627 097F          ;
628 0980          ;
629 0981          ;
630 0982          ;
631 0983          ;
632 0984          ;
633 0985          ;
634 0986          ;
635 0987          ;
636 0988          ;
637 0989          ;
638 098A          ;
639 098B          ;
640 098C          ;
641 098D          ;
642 098E          ;
643 098F          ;
644 0990          ;
645 0991          ;
646 0992          ;
647 0993          ;
648 0994          ;
649 0995          ;
650 0996          ;
651 0997          ;
652 0998          ;
653 0999          ;
654 099A          ;
655 099B          ;
656 099C          ;
657 099D          ;
658 099E          ;
659 099F          ;
660 09A0          ;
661 09A1          ;
662 09A2          ;
663 09A3          ;
664 09A4          ;
665 09A5          ;
666 09A6          ;
667 09A7          ;
668 09A8          ;
669 09A9          ;
670 09AA          ;
671 09AB          ;
672 09AC          ;
673 09AD          ;
674 09AE          ;
675 09AF          ;
676 09B0          ;
677 09B1          ;
678 09B2          ;
679 09B3          ;
680 09B4          ;
681 09B5          ;
682 09B6          ;
683 09B7          ;
684 09B8          ;
685 09B9          ;
686 09BA          ;
687 09BB          ;
688 09BC          ;
689 09BD          ;
690 09BE          ;
691 09BF          ;
692 09C0          ;
693 09C1          ;
694 09C2          ;
695 09C3          ;
696 09C4          ;
697 09C5          ;
698 09C6          ;
699 09C7          ;
700 09C8          ;
701 09C9          ;
702 09CA          ;
703 09CB          ;
704 09CC          ;
705 09CD          ;
706 09CE          ;
707 09CF          ;
708 09D0          ;
709 09D1          ;
710 09D2          ;
711 09D3          ;
712 09D4          ;
713 09D5          ;
714 09D6          ;
715 09D7          ;
716 09D8          ;
717 09D9          ;
718 09DA          ;
719 09DB          ;
720 09DC          ;
721 09DD          ;
722 09DE          ;
723 09DF          ;
724 09E0          ;
725 09E1          ;
726 09E2          ;
727 09E3          ;
728 09E4          ;
729 09E5          ;
730 09E6          ;
731 09E7          ;
732 09E8          ;
733 09E9          ;
734 09EA          ;
735 09EB          ;
736 09EC          ;
737 09ED          ;
738 09EE          ;
739 09EF          ;
740 09F0          ;
741 09F1          ;
742 09F2          ;
743 09F3          ;
744 09F4          ;
745 09F5          ;
746 09F6          ;
747 09F7          ;
748 09F8          ;
749 09F9          ;
750 09FA          ;
751 09FB          ;
752 09FC          ;
753 09FD          ;
754 09FE          ;
755 09FF          ;
756 0A00          ;
757 0A01          ;
758 0A02          ;
759 0A03          ;
760 0A04          ;
761 0A05          ;
762 0A06          ;
763 0A07          ;
764 0A08          ;
765 0A09          ;
766 0A0A          ;
767 0A0B          ;
768 0A0C          ;
769 0A0D          ;
770 0A0E          ;
771 0A0F          ;
772 0A10          ;
773 0A11          ;
774 0A12          ;
775 0A13          ;
776 0A14          ;
777 0A15          ;
778 0A16          ;
779 0A17          ;
780 0A18          ;
781 0A19          ;
782 0A1A          ;
783 0A1B          ;
784 0A1C          ;
785 0A1D          ;
786 0A1E          ;
787 0A1F          ;
788 0A20          ;
789 0A21          ;
790 0A22          ;
791 0A23          ;
792 0A24          ;
793 0A25          ;
794 0A26          ;
795 0A27          ;
796 0A28          ;
797 0A29          ;
798 0A2A          ;
799 0A2B          ;
800 0A2C          ;
801 0A2D          ;
802 0A2E          ;
803 0A2F          ;
804 0A30          ;
805 0A31          ;
806 0A32          ;
807 0A33          ;
808 0A34          ;
809 0A35          ;
810 0A36          ;
811 0A37          ;
812 0A38          ;
813 0A39          ;
814 0A3A          ;
815 0A3B          ;
816 0A3C          ;
817 0A3D          ;
818 0A3E          ;
819 0A3F          ;
820 0A40          ;
821 0A41          ;
822 0A42          ;
823 0A43          ;
824 0A44          ;
825 0A45          ;
826 0A46          ;
827 0A47          ;
828 0A48          ;
829 0A49          ;
830 0A4A          ;
831 0A4B          ;
832 0A4C          ;
833 0A4D          ;
834 0A4E          ;
835 0A4F          ;
836 0A50          ;
837 0A51          ;
838 0A52          ;
839 0A53          ;
840 0A54          ;
841 0A55          ;
842 0A56          ;
843 0A57          ;
844 0A58          ;
845 0A59          ;
846 0A5A          ;
847 0A5B          ;
848 0A5C          ;
849 0A5D          ;
850 0A5E          ;
851 0A5F          ;
852 0A60          ;
853 0A61          ;
854 0A62          ;
855 0A63          ;
856 0A64          ;
857 0A65          ;
858 0A66          ;
859 0A67          ;
860 0A68          ;
861 0A69          ;
862 0A6A          ;
863 0A6B          ;
864 0A6C          ;
865 0A6D          ;
866 0A6E          ;
867 0A6F          ;
868 0A70          ;
869 0A71          ;
870 0A72          ;
871 0A73          ;
872 0A74          ;
873 0A75          ;
874 0A76          ;
875 0A77          ;
876 0A78          ;
877 0A79          ;
878 0A7A          ;
879 0A7B          ;
880 0A7C          ;
881 0A7D          ;
882 0A7E          ;
883 0A7F          ;
884 0A80          ;
885 0A81          ;
886 0A82          ;
887 0A83          ;
888 0A84          ;
889 0A85          ;
890 0A86          ;
891 0A87          ;
892 0A88          ;
893 0A89          ;
894 0A8A          ;
895 0A8B          ;
896 0A8C          ;
897 0A8D          ;
898 0A8E          ;
899 0A8F          ;
900 0A90          ;
901 0A91          ;
902 0A92          ;
903 0A93          ;
904 0A94          ;
905 0A95          ;
906 0A96          ;
907 0A97          ;
908 0A98          ;
909 0A99          ;
910 0A9A          ;
911 0A9B          ;
912 0A9C          ;
913 0A9D          ;
914 0A9E          ;
915 0A9F          ;
916 0AA0          ;
917 0AA1          ;
918 0AA2          ;
919 0AA3          ;
920 0AA4          ;
921 0AA5          ;
922 0AA6          ;
923 0AA7          ;
924 0AA8          ;
925 0AA9          ;
926 0AAA          ;
927 0AAB          ;
928 0AAC          ;
929 0AAD          ;
930 0AAE          ;
931 0AAF          ;
932 0AB0          ;
933 0AB1          ;
934 0AB2          ;
935 0AB3          ;
936 0AB4          ;
937 0AB5          ;
938 0AB6          ;
939 0AB7          ;
940 0AB8          ;
941 0AB9          ;
942 0ABA          ;
943 0ABB          ;
944 0ABC          ;
945 0ABD          ;
946 0ABE          ;
947 0ABF          ;
948 0AC0          ;
949 0AC1          ;
950 0AC2          ;
951 0AC3          ;
952 0AC4          ;
953 0AC5          ;
954 0AC6          ;
955 0AC7          ;
956 0AC8          ;
957 0AC9          ;
958 0ACA          ;
959 0ACB          ;
960 0ACC          ;
961 0ACD          ;
962 0ACE          ;
963 0ACF          ;
964 0AD0          ;
965 0AD1          ;
966 0AD2          ;
967 0AD3          ;
968 0AD4          ;
969 0AD5          ;
970 0AD6          ;
971 0AD7          ;
972 0AD8          ;
973 0AD9          ;
974 0ADA          ;
975 0ADB          ;
976 0ADC          ;
977 0ADE          ;
978 0ADF          ;
979 0AE0          ;
980 0AE1          ;
981 0AE2          ;
982 0AE3          ;
983 0AE4          ;
984 0AE5          ;
985 0AE6          ;
986 0AE7          ;
987 0AE8          ;
988 0AE9          ;
989 0AEA          ;
990 0AEB          ;
991 0AEC          ;
992 0AED          ;
993 0AEE          ;
994 0AEF          ;
995 0AF0          ;
996 0AF1          ;
997 0AF2          ;
998 0AF3          ;
999 0AF4          ;
1000 0AF5          ;
1001 0AF6          ;
1002 0AF7          ;
1003 0AF8          ;
1004 0AF9          ;
1005 0AFA          ;
1006 0AFB          ;
1007 0AFC          ;
1008 0AFD          ;
1009 0AFE          ;
1010 0AFF          ;
1011 0B00          ;
1012 0B01          ;
1013 0B02          ;
1014 0B03          ;
1015 0B04          ;
1016 0B05          ;
1017 0B06          ;
1018 0B07          ;
1019 0B08          ;
1020 0B09          ;
1021 0B0A          ;
1022 0B0B          ;
1023 0B0C          ;
1024 0B0D          ;
1025 0B0E          ;
1026 0B0F          ;
1027 0B10          ;
1028 0B11          ;
1029 0B12          ;
1030 0B13          ;
1031 0B14          ;
1032 0B15          ;
1033 0B16          ;
1034 0B17          ;
1035 0B18          ;
1036 0B19          ;
1037 0B1A          ;
1038 0B1B          ;
1039 0B1C          ;
1040 0B1D          ;
1041 0B1E          ;
1042 0B1F          ;
1043 0B20          ;
1044 0B21          ;
1045 0B22          ;
1046 0B23          ;
1047 0B24          ;
1048 0B25          ;
1049 0B26          ;
1050 0B27          ;
1051 0B28          ;
1052 0B29          ;
1053 0B2A          ;
1054 0B2B          ;
1055 0B2C          ;
1056 0B2D          ;
1057 0B2E          ;
1058 0B2F          ;
1059 0B30          ;
1060 0B31          ;
1061 0B32          ;
1062 0B33          ;
1063 0B34          ;
1064 0B35          ;
1065 0B36          ;
1066 0B37          ;
1067 0B38          ;
1068 0B39          ;
1069 0B3A          ;
1070 0B3B          ;
1071 0B3C          ;
1072 0B3D          ;
1073 0B3E          ;
1074 0B3F          ;
1075 0B40          ;
1076 0B41          ;
1077 0B42          ;
1078 0B43          ;
1079 0B44          ;
1080 0B45          ;
1081 0B46          ;
1082 0B47          ;
1083 0B48          ;
1084 0B49          ;
1085 0B4A          ;
1086 0B4B          ;
1087 0B4C          ;
1088 0B4D          ;
1089 0B4E          ;
1090 0B4F          ;
1091 0B50          ;
1092 0B51          ;
1093 0B52          ;
1094 0B53          ;
1095 0B54          ;
1096 0B55          ;
1097 0B56          ;
1098 0B57          ;
1099 0B58          ;
1100 0B59          ;
1101 0B5A          ;
1102 0B5B          ;
1103 0B5C          ;
1104 0B5D          ;
1105 0B5E          ;
1106 0B5F          ;
1107 0B60          ;
1108 0B61          ;
1109 0B62          ;
1110 0B63          ;

```



```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 29          04.07.83
01 07E6 ; (END =CR )
02 07E6 ;
03 07E6 ;
04 07E6 ?GETL: ENT AF
05 07E6 F5 PUSH BC
06 07E7 C5 PUSH HL
07 07E8 E5 PUSH DE
08 07E9 D5 ENT
09 07EA CDB309 CALL ??KEY
10 07EB CDB309 CALL ??KEY
11 07ED F5 ENT
12 07ED F5 ENT
13 07EE 47 LD B,A
14 07EF 3A9D11 RRCA A,(SWRK)
15 07F2 0F LD A,B
16 07F3 D47705 CALL NC,?BEL
17 07F6 78 LD A,B
18 07F7 217011 LD HL,KANAF
19 07FA E6F0 AND F0H
20 07FC FEC0 CP COH
21 07FE D1 POP DE
22 07FF 78 LD A,B
23 0800 2016 JR NZ,GETL2
24 0802 FEDD CP CDH
25 0804 2855 JR Z,GETL3
26 0806 FEDB CP CBH
27 0808 CA2208 JP Z,GETLC
28 080B FECF CP CFH
29 080D 2B09 JR Z,GETL2
30 080F FEC7 CP C7H
31 0811 300A JR NC,GETL5
32 0813 CB1B RR E
33 0815 78 LD A,B
34 0816 3005 JR NC,GETL5
35 0818 ENT
36 081B CDB50D CALL ?DSP
37 081B 1BCD JR GETL1
38 081D ENT
39 081D CDDC0D ?DPC: CALL GETL1
40 0820 1BCB JR
41 0822 ;
42 0822 ; BREAK IN
43 0822 ;
44 0822 E1 GETLC: POP HL
45 0823 E5 HL
46 0824 361B LD (HL),1BH
47 0826 23 INC HL
48 0827 360D LD (HL),0DH
49 0829 1853 JR GETLR
50 082B ;
51 082B ; GETLA
52 082B 0F GETLA: RRCA
53 082C 3037 JR NC,GETL6
54 082E 1833 JR GETLB
55 0830 ;
56 0830 ;
57 0830 ;
58 0830 ; DELAY 7M SEC AND SWEP
59 0830 ;
60 0830 CD9609 DSWEP: CALL DLY12

; ENTRY KEY
; IN KEY DATA SAVE
; BELL WORK
; ENTRY BELL
; KANA & GRAPH FLAG
; Ereg=FLAGreg
; CR
; BREAK
; NIKO MARK WH.
; CRT EDITION
; CY ?

; CRT CONTROL
; BREAK CODE
; CY+D7

01 0833 CD500A
02 0836 C9 RET
03 0837 ;
04 0837 ;
05 0837 DEFS 36
06 085B ;
07 085B ;
08 085B ;
09 085B ; ORG 085BH;GETL3
10 085B ;
11 085B CDF302 GETL3: CALL .MANG
12 085E 062B LD B,40
13 0860 30C9 JR NC,GETLA
14 0862 25 DEC H
15 0863 0650 GETLB: LD B,80
16 0865 2E00 GETL6: LD L,0
17 0867 CDB40F CALL ?PNT1
18 086A D1 POP DE
19 086B D5 PUSH DE
20 086C 7E LD A,(HL)
21 086D CDBCE0B GETLZ: LD ?DACN
22 0870 12 LD (DE),A
23 0871 23 INC HL
24 0872 13 DE
25 0873 10F7 DJNZ GETLZ
26 0875 EB EX DE,HL
27 0876 360D GETLU: LD (HL),0DH
28 0878 2B JR DEC HL
29 0879 7E LD A,(HL)
30 087A FE20 CP 20H
31 087C ;
32 087C ; CR AND NEW LINE
33 087C ;
34 087C ; JR Z,GETLU
35 087C 2BF8 ;
36 087E ; NEW LINE RETURN
37 087E ;
38 087E ;
39 087E CDDCE09 GETLR: CALL ?LTNL
40 0881 D1 POP DE
41 0882 E1 POP HL
42 0883 C1 POP BC
43 0884 F1 POP AF
44 0885 C9 RET
45 0886 ;
46 0886 ;
47 0886 ;
48 0886 ; ORG 0893H
49 0893 ; DEFS +13
50 0893 ; MESSAGE PRINT
51 0893 ;
52 0893 ; DE PRINT DATA LOW ADR.
53 0893 ; END=CR
54 0893 ;
55 0893 ;
56 0893 ; MSG: ENT
57 0893 F5 PUSH AF
58 0894 C5 PUSH BC
59 0895 D5 PUSH DE
60 0896 1A LD A,(DE)
MSG1:

```

```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 31          04.07.83
01 0897 FE0D      CP      ODH
02 0899 2B0C      JR      Z,MSGX2
03 089B CD3509    CALL    ?PRINT
04 089E 13        INC     DE
05 089F 1BF5      JR      MSG1
06 0BA1          ;
07 0BA1          ;ORG 0BA1H
08 0BA1          ;
09 0BA1          ; ALL PRINT MESSAGE
10 0BA1          ;
11 0BA1          ;
12 0BA1          ?MSGX: ENT
13 0BA1 F5        PUSH    AF
14 0BA2 C5        PUSH    BC
15 0BA3 D5        PUSH    DE
16 0BA4 1A        LD      A,(DE)
17 0BA5 FE0D      CP      ODH
18 0BA7 CAE60E    JP      Z,?RSTR1
19 0BA8 CDB90B    CALL    ?ADCN
20 0BAD CD6C09    CALL    PRINT3
21 0BB0 13        INC     DE
22 0BB1 1BF1      JR      MSGX1
23 0BB3          ;
24 0BB3          ; TOP OF KEYTBLs
25 0BB3          ;
26 0BB3 112A0C    ?KYSM: LD      DE,KTBLs
27 0BB6 1B42      JR      ?KY5
28 0BB8          ;
29 0BB8          ; BREAK CODE IN
30 0BB8          ;
31 0BB8 3ECB      #BRK: LD      A,CBH
32 0BB8 B7        OR      A
33 0BB8 1B19      JR      ?KY1
34 0BBD          ;
35 0BBD          ;ORG 0BBDH
36 0BBD          ;
37 0BBD          ; GETKEY
38 0BBD          ;
39 0BBD          ; NOT ECHO BACK
40 0BBD          ;
41 0BBD          ;
42 0BBD          ; EXIT:ACC=ASCII CODE
43 0BBD          ;
44 0BBD          ?GET: ENT
45 0BBD CDCA0B    CALL    ?KEY
46 0BC0 D6F0      SUB     F0H
47 0BC2 C8        RET     Z
48 0BC3 C6F0      ADD     A,F0H
49 0BC5 C3CE0B    JP      ?DACN
50 0BCB          ;
51 0BCB          ;
52 0BCB          ; DEFS +2
53 0BCA          ;
54 0BCA          ;
55 0BCA          ;
56 0BCA          ;ORG 0BCAH: ?KEY
57 0BCA          ;
58 0BCA          ; 1KEY INPUT
59 0BCA          ; IN      B = KEY MODE(SHIFT,CTRL,BREAK)
60 0BCA          ;

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 32          04.07.83
01 0BCA          ; C = KEY DATA (COLUMN & ROW)
02 0BCA          ; ACC=DISPLAY CODE
03 0BCA          ; EXIT
04 0BCA          ; IF NO KEY ACC=F0H
05 0BCA          ; IF CY=1 THEN ATTRIBUTE ON
06 0BCA          ; (SMALL,HIRAKANA)
07 0BCA          ; ?KEY: ENT
08 0BCA C5        PUSH    BC
09 0BCB D5        PUSH    DE
10 0BCB E5        PUSH    HL
11 0BCD CD300B    CALL    DSMWP
12 0BD0 78        LD      A,B
13 0BD1 07        RLCA
14 0BD2 3B06      JR      C,?KY2
15 0BD4 3EFO      LD      A,F0H
16 0BD6          ;
17 0BD6 E1        ENT     HL
18 0BD7 D1        POP     DE
19 0BD8 C1        POP     BC
20 0BD9 C9        RET
21 0BDA          ;
22 0BDA          ; ?KY2: ENT
23 0BDA 11EA0B    LD      DE,KTBL
24 0BD0 78        LD      A,B
25 0BDE FE8B      CP      8BH
26 0BE0 2B06      JR      Z,#BRK
27 0BE2 2B00      LD      H,0
28 0BE4 69        LD      L,C
29 0BE5 CB6F      BIT     5,A
30 0BE7 200E      JR      NZ,?KYS-3
31 0BE9 3A7011    LD      A,(KANAF)
32 0BEC 0F        RRCA
33 0BED DAFE0B    JP      C,?KYGRP
34 0BF0 78        LD      A,B
35 0BF1 17        RLA
36 0BF2 17        RLA
37 0BF3 3BBE      JR      ?KYS
38 0BF5 1B03      LD      DE,KTBLC
39 0BF7 11AA0C    LD      HL,DE
40 0BFA 19        ADD     HL,DE
41 0BFA 19        ADD     HL,DE
42 0BF8 7E        LD      A,(HL)
43 0BF8 7E        LD      ?KY1
44 0BFC 1BD8      JR      4,B
45 0BFE CB70      BIT     Z,?KYGRS
46 0BFE CB70      JR      DE,KTBLG
47 0900 2B07      LD      HL,DE
48 0902 11E90C    LD      ADD
49 0905 19        SCF
50 0906 37        JR      ?KYS5
51 0907 1BF2      ;
52 0909          ; ?KYGRS: LD
53 0909 116A0C    LD      DE,KTBLGS
54 090C 1BEC      JR      ?KYS
55 090E          ;
56 090E          ;
57 090E          ;
58 090E          ;
59 090E          ; ORG 090EH
60 090E          ;

```

```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 33 04.07.83
01 090E ; NEWLINE
02 090E ;
03 090E ;
04 090E ?LTNL: ENT
05 090E XOR A
06 090F LD (DPRNT),A
07 0912 LD A,CDH
08 0914 JR PRNT5
09 0916 DEFS +2
10 0918 ;ORG 0918H
11 0918 ;
12 0918 ?NL: ENT
13 091B LD A,(DPRNT)
14 091B OR A
15 091C RET Z
16 091D JR ?LTNL
17 091F DEFS +1
18 0920 ;ORG 0920H
19 0920 ;
20 0920 ; PRINT SPACE
21 0920 ;
22 0920 ?PRTS: ENT
23 0920 LD A,20H
24 0922 JR ?PRNT
25 0924 ;
26 0924 ; PRINT TAB
27 0924 ;
28 0924 ?PRTT: ENT
29 0924 CALL PRNTS
30 0927 LD A,(DPRNT)
31 092A OR A
32 092B RET Z
33 092C SUB +10
34 092E JR C,-10
35 0930 JR NZ,-4
36 0932 DEFS +3
37 0935 ;ORG 0935H
38 0935 ;
39 0935 ; PRINT
40 0935 ;
41 0935 ; IN ACC. = PRINT DATA (ASCII)
42 0935 ;
43 0935 ?PRNT: ENT
44 0935 CP ODH
45 0937 JR Z,?LTNL
46 0939 PUSH BC
47 093A LD C,A
48 093B LD B,A
49 093C CALL ?PRT
50 093F LD A,B
51 0940 POP BC
52 0941 RET
53 0942 ;
54 0942 ; MSGOK: ENT
55 0942 DEFM 'OK!'
56 0942 DEF B ODH
57 0945 ;ORG 0945H
58 0946 ;
59 0946 ; PRINT ROUTINE
60 0946 ;

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 34 04.07.83
01 0946 ; 1 CHA.
02 0946 ; INPUT:C=ASCII DATA (?DSP+?DPCT)
03 0946 ;
04 0946 ?PRT: ENT
05 0946 LD A,C
06 0947 CALL ?ADCN
07 094A LD C,A
08 094B CP FOH
09 094D RET Z
10 094E AND FOH
11 0950 CP COH
12 0952 LD A,C
13 0953 JR NZ,PRNT3
14 0955 CP C7H
15 0957 JR NC,PRNT3
16 0959 PRNTS: ENT
17 0959 CALL ?DPCT
18 095C CP C3H
19 095E JR Z,PRNT4
20 0960 CP C5H
21 0962 JR Z,PRNT2
22 0964 CP C6H
23 0966 NZ
24 0967 PRNT2: XOR A
25 096B LD (DPRNT),A
26 096B C9
27 096C RET
28 096C PRNT3: ENT
29 096F CALL ?DSP
30 0972 LD A,(DPRNT)
31 0973 INC A
32 0975 CP +80
33 0977 JR C,PRNT2+1
34 0979 SUB +80
35 097B JR PRNT2+1
36 097B ;
37 097B ;
38 097B ;
39 097B ; FLASSING BYPASS 1
40 097B ;
41 097B ;
42 097B FLAS1: ENT
43 097B LD A,(FLASH)
44 097E JR FLAS2
45 0980 ;
46 0980 ; BREAK SUBROUTINE BYPASS 1
47 0980 ;
48 0980 ;
49 0980 ; CTRL OR NOT KEY
50 0980 ?BRK2: ENT
51 0980 BIT 5,A
52 0982 JR Z,?BRK3
53 0984 OR A
54 0985 RET
55 0986 ;
56 0986 ?BRK3: LD A,20H
57 098B OR A
58 0989 SCF
59 098A RET
60 098B ;

; NOT OR CTRL
; CTRL
; NOTKEY A=7FH
; CTRL D5=1
; ZERO FLG. CLR

```

```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 35          04.07.83
01 098B 46494C45 MSGSV: ENT DEFN 'FILENAME?'
02 098B 46494C45
03 098F 4E414D45
04 0993 3F20 DEFN ODH
05 0995 OD
06 0996 DLY 7 MSEC
07 0996
08 0996
09 0996 DLY12: ENT PUSH BC
10 0996 C5 LD B,15H
11 0997 0615 CALL DLY3
12 0999 CD4A0A DJNZ -3
13 099C 10FB POP BC
14 099E C1 RET
15 099F C9
16 09A0
17 09A0
18 09A0
19 09A0
20 09A0 MSG72: ENT DEFN 'LOADING'
21 09A0
22 09A0 4C4F4144 DEFN ODH
23 09A4 494E4720
24 09AB OD
25 09A9
26 09A9
27 09A9
28 09A9
29 09A9
30 09A9 DLY4: ENT LD A,59H
31 09A9 3E59 DEC A
32 09AB 3D JP NZ,-1
33 09AC C2AB09 RET
34 09AF C9
35 09B0
36 09B0
37 09B0
38 09B3
39 09B3
40 09B3
41 09B3
42 09B3
43 09B3
44 09B3
45 09B3
46 09B3
47 09B3
48 09B3
49 09B3
50 09B3 E5
51 09B4 CD920B CALL ?SAVE
52 09B7 ENT
53 09B7 CD7E05 CALL FLKEY
54 09BA 20FB JR NZ,KSL1
55 09BC ENT
56 09BC CD7E05 CALL FLKEY
57 09BF 29FB JR Z,KSL2
58 09C1 67 LD H,A
59 09C2 CD9609 CALL DLY12
60 09C5 CDCA0B CALL ?KEY

; KEY
; KEY IN THEN JUMP
; NOT KEY IN THEN JUMP
; DELAY CHATTER
; DELAY 18E2

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 36          04.07.83
01 09C8 F5 PUSH AF
02 09C9 BC CP H
03 09CA E1 POP HL
04 09CB 20EF JR NZ,KSL2
05 09CD E5 PUSH HL
06 09CE F1 POP AF
07 09CF CDF005 CALL ?LOAD
08 09D2 E1 POP HL
09 09D3 C9 RET
10 09D4
11 09D4
12 09D4
13 09D4
14 09D4
15 09D4 AF A
16 09D5 ENT
17 09D5 010008 LD BC,0800H
18 09D8 ENT
19 09D8 D5 PUSH DE
20 09D9 57 LD D,A
21 09DA ENT
22 09DA 72 LD (HL),D
23 09DB 23 INC HL
24 09DC 08 DEC BC
25 09DD 78 LD A,B
26 09DE B1 OR C
27 09DF 20F9 JR NZ,CLEAR1
28 09E1 D1 POP DE
29 09E2 C9 RET
30 09E3
31 09E3
32 09E3
33 09E3
34 09E3
35 09E3
36 09E3
37 09E3 F5 ENT
38 09E4 E5 PUSH HL
39 09E5 3A02E0 LD A,(KEYFC)
40 09E8 07 RLCA
41 09E9 07 RLCA
42 09EA 38BF JR C,FLAS1
43 09EC 3A9211 LD A,(FLSDT)
44 09EF ENT
45 09EF CDB10F CALL ?PONT
46 09F2 77 LD (HL),A
47 09F3 ENT
48 09F3 E1 POP HL
49 09F4 F1 POP AF
50 09F5 C9 RET
51 09F6
52 09F6
53 09F6
54 09F6
55 09FF
56 09FF
57 09FF
58 09FF
59 09FF
60 09FF 18E2 JR ?FLS

; CY FLG.
; BC = CLR BYTE SIZE
; A = CLR DATA
; IN KEY DATA
; FLASHING DATA LOAD
; DISPLAY POSITION

```

-187


```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 39          04.07.83
01 0A6F FEF8          CP      FBH
02 0A71 2BF3          JR      Z,SNEP0
03 0A73          SWEP9:  ENT    B,D
04 0A74 42          LD      HL, D
05 0A75 E1          POP     DE
06 0A76 D1          POP     DE
07 0A77 C9          RET
08 0A77          ;
09 0A77          SWEP3:  ENT    A,(KEYPB)
10 0A77 3A01E0       LD      CPL
11 0A7A 2F          OR      A
12 0A7B B7          JR      Z,SNEP0
13 0A7C 2BE8       LD      E,A
14 0A7E 5F          LD      ENT
15 0A7F          SWEP2:  ENT    H,B
16 0A7F 260B       LD      A,B
17 0A81 78          LD      OFH
18 0A82 E60F       AND     RLCA
19 0A84 07          RLCA
20 0A85 07          RLCA
21 0A86 07          LD      C,A
22 0A87 4F          LD      A,E
23 0A88 7B          DEC     H
24 0A89 25          RRCA
25 0A8A 0F          JR      NC,-2
26 0A8B 30FC       JR      A,H
27 0A8D 7C          ADD     C,A
28 0A8E 81          LD      C,A
29 0A8F 4F          JR      SNEP01
30 0A90 1BD2
31 0A92          ;
32 0A92          ;
33 0A92          ;
34 0A92          ;
35 0A92          ;
36 0A92          ;
37 0A92 F0          DEFBL:  ; 00 - OF ;
38 0A93 F0          DEFBL  F0H
39 0A94 F0          DEFBL  F0H
40 0A95 F3          DEFBL  F3H
41 0A96 F0          DEFBL  F0H
42 0A97 F5          DEFBL  F5H
43 0A98 F0          DEFBL  F0H
44 0A99 F0          DEFBL  F0H
45 0A9A F0          DEFBL  F0H
46 0A9B F0          DEFBL  F0H
47 0A9C F0          DEFBL  F0H
48 0A9D F0          DEFBL  F0H
49 0A9E F0          DEFBL  F0H
50 0A9F F0          DEFBL  F0H
51 0AA0 F0          DEFBL  F0H
52 0AA1 F0          DEFBL  F0H
53 0AA2          ; 10 - 1F ;
54 0AA2 F0          DEFBL  F0H
55 0AA3 C1          DEFBL  C1H
56 0AA4 C2          DEFBL  C2H
57 0AA5 C3          DEFBL  C3H
58 0AA6 C4          DEFBL  C4H
59 0AA7 C5          DEFBL  C5H
60 0AAB C6          DEFBL  C6H

; BREAK KEY ROW
; 00 - OF ;
; 01 - 1F ;
; 02 - 2F ;
; 03 - 3F ;
; 04 - 4F ;
; 05 - 5F ;
; 06 - 6F ;
; 07 - 7F ;
; 08 - 8F ;
; 09 - 9F ;
; 0A - AF ;
; 0B - BF ;
; 0C - CF ;
; 0D - DF ;
; 0E - EF ;
; 0F - FF ;
; 10 - 1F ;
; 11 - 2F ;
; 12 - 3F ;
; 13 - 4F ;
; 14 - 5F ;
; 15 - 6F ;
; 16 - 7F ;
; 17 - 8F ;
; 18 - 9F ;
; 19 - AF ;
; 1A - BF ;
; 1B - CF ;
; 1C - DF ;
; 1D - EF ;
; 1E - FF ;
; 1F - 00 ;
; 20 - 2F ;
; 21 - 3F ;
; 22 - 4F ;
; 23 - 5F ;
; 24 - 6F ;
; 25 - 7F ;
; 26 - 8F ;
; 27 - 9F ;
; 28 - AF ;
; 29 - BF ;
; 2A - CF ;
; 2B - DF ;
; 2C - EF ;
; 2D - FF ;
; 2E - 00 ;
; 2F - 01 ;
; 30 - 02 ;
; 31 - 03 ;
; 32 - 04 ;
; 33 - 05 ;
; 34 - 06 ;
; 35 - 07 ;
; 36 - 08 ;
; 37 - 09 ;
; 38 - 0A ;
; 39 - 0B ;
; 40 - 0C ;
; 41 - 0D ;
; 42 - 0E ;
; 43 - 0F ;
; 44 - 10 ;
; 45 - 11 ;
; 46 - 12 ;
; 47 - 13 ;
; 48 - 14 ;
; 49 - 15 ;
; 50 - 16 ;
; 51 - 17 ;
; 52 - 18 ;
; 53 - 19 ;
; 54 - 1A ;
; 55 - 1B ;
; 56 - 1C ;
; 57 - 1D ;
; 58 - 1E ;
; 59 - 1F ;
; 60 - 20 ;
; 61 - 21 ;
; 62 - 22 ;
; 63 - 23 ;
; 64 - 24 ;
; 65 - 25 ;
; 66 - 26 ;
; 67 - 27 ;
; 68 - 28 ;
; 69 - 29 ;
; 70 - 2A ;
; 71 - 2B ;
; 72 - 2C ;
; 73 - 2D ;
; 74 - 2E ;
; 75 - 2F ;
; 76 - 30 ;
; 77 - 31 ;
; 78 - 32 ;
; 79 - 33 ;
; 80 - 34 ;
; 81 - 35 ;
; 82 - 36 ;
; 83 - 37 ;
; 84 - 38 ;
; 85 - 39 ;
; 86 - 3A ;
; 87 - 3B ;
; 88 - 3C ;
; 89 - 3D ;
; 90 - 3E ;
; 91 - 3F ;
; 92 - 40 ;
; 93 - 41 ;
; 94 - 42 ;
; 95 - 43 ;
; 96 - 44 ;
; 97 - 45 ;
; 98 - 46 ;
; 99 - 47 ;
; 100 - 48 ;
; 101 - 49 ;
; 102 - 4A ;
; 103 - 4B ;
; 104 - 4C ;
; 105 - 4D ;
; 106 - 4E ;
; 107 - 4F ;
; 108 - 50 ;
; 109 - 51 ;
; 110 - 52 ;
; 111 - 53 ;
; 112 - 54 ;
; 113 - 55 ;
; 114 - 56 ;
; 115 - 57 ;
; 116 - 58 ;
; 117 - 59 ;
; 118 - 5A ;
; 119 - 5B ;
; 120 - 5C ;
; 121 - 5D ;
; 122 - 5E ;
; 123 - 5F ;
; 124 - 60 ;
; 125 - 61 ;
; 126 - 62 ;
; 127 - 63 ;
; 128 - 64 ;
; 129 - 65 ;
; 130 - 66 ;
; 131 - 67 ;
; 132 - 68 ;
; 133 - 69 ;
; 134 - 6A ;
; 135 - 6B ;
; 136 - 6C ;
; 137 - 6D ;
; 138 - 6E ;
; 139 - 6F ;
; 140 - 70 ;
; 141 - 71 ;
; 142 - 72 ;
; 143 - 73 ;
; 144 - 74 ;
; 145 - 75 ;
; 146 - 76 ;
; 147 - 77 ;
; 148 - 78 ;
; 149 - 79 ;
; 150 - 7A ;
; 151 - 7B ;
; 152 - 7C ;
; 153 - 7D ;
; 154 - 7E ;
; 155 - 7F ;
; 156 - 80 ;
; 157 - 81 ;
; 158 - 82 ;
; 159 - 83 ;
; 160 - 84 ;
; 161 - 85 ;
; 162 - 86 ;
; 163 - 87 ;
; 164 - 88 ;
; 165 - 89 ;
; 166 - 8A ;
; 167 - 8B ;
; 168 - 8C ;
; 169 - 8D ;
; 170 - 8E ;
; 171 - 8F ;
; 172 - 90 ;
; 173 - 91 ;
; 174 - 92 ;
; 175 - 93 ;
; 176 - 94 ;
; 177 - 95 ;
; 178 - 96 ;
; 179 - 97 ;
; 180 - 98 ;
; 181 - 99 ;
; 182 - 9A ;
; 183 - 9B ;
; 184 - 9C ;
; 185 - 9D ;
; 186 - 9E ;
; 187 - 9F ;
; 188 - A0 ;
; 189 - A1 ;
; 190 - A2 ;
; 191 - A3 ;
; 192 - A4 ;
; 193 - A5 ;
; 194 - A6 ;
; 195 - A7 ;
; 196 - A8 ;
; 197 - A9 ;
; 198 - AA ;
; 199 - AB ;
; 200 - AC ;
; 201 - AD ;
; 202 - AE ;
; 203 - AF ;
; 204 - B0 ;
; 205 - B1 ;
; 206 - B2 ;
; 207 - B3 ;
; 208 - B4 ;
; 209 - B5 ;
; 210 - B6 ;
; 211 - B7 ;
; 212 - B8 ;
; 213 - B9 ;
; 214 - BA ;
; 215 - BB ;
; 216 - BC ;
; 217 - BD ;
; 218 - BE ;
; 219 - BF ;
; 220 - C0 ;
; 221 - C1 ;
; 222 - C2 ;
; 223 - C3 ;
; 224 - C4 ;
; 225 - C5 ;
; 226 - C6 ;
; 227 - C7 ;
; 228 - C8 ;
; 229 - C9 ;
; 230 - CA ;
; 231 - CB ;
; 232 - CC ;
; 233 - CD ;
; 234 - CE ;
; 235 - CF ;
; 236 - D0 ;
; 237 - D1 ;
; 238 - D2 ;
; 239 - D3 ;
; 240 - D4 ;
; 241 - D5 ;
; 242 - D6 ;
; 243 - D7 ;
; 244 - D8 ;
; 245 - D9 ;
; 246 - DA ;
; 247 - DB ;
; 248 - DC ;
; 249 - DD ;
; 250 - DE ;
; 251 - DF ;
; 252 - E0 ;
; 253 - E1 ;
; 254 - E2 ;
; 255 - E3 ;
; 256 - E4 ;
; 257 - E5 ;
; 258 - E6 ;
; 259 - E7 ;
; 260 - E8 ;
; 261 - E9 ;
; 262 - EA ;
; 263 - EB ;
; 264 - EC ;
; 265 - ED ;
; 266 - EE ;
; 267 - EF ;
; 268 - F0 ;
; 269 - F1 ;
; 270 - F2 ;
; 271 - F3 ;
; 272 - F4 ;
; 273 - F5 ;
; 274 - F6 ;
; 275 - F7 ;
; 276 - F8 ;
; 277 - F9 ;
; 278 - FA ;
; 279 - FB ;
; 280 - FC ;
; 281 - FD ;
; 282 - FE ;
; 283 - FF ;
; 284 - 00 ;
; 285 - 01 ;
; 286 - 02 ;
; 287 - 03 ;
; 288 - 04 ;
; 289 - 05 ;
; 290 - 06 ;
; 291 - 07 ;
; 292 - 08 ;
; 293 - 09 ;
; 294 - 0A ;
; 295 - 0B ;
; 296 - 0C ;
; 297 - 0D ;
; 298 - 0E ;
; 299 - 0F ;
; 300 - 10 ;
; 301 - 11 ;
; 302 - 12 ;
; 303 - 13 ;
; 304 - 14 ;
; 305 - 15 ;
; 306 - 16 ;
; 307 - 17 ;
; 308 - 18 ;
; 309 - 19 ;
; 310 - 1A ;
; 311 - 1B ;
; 312 - 1C ;
; 313 - 1D ;
; 314 - 1E ;
; 315 - 1F ;
; 316 - 20 ;
; 317 - 21 ;
; 318 - 22 ;
; 319 - 23 ;
; 320 - 24 ;
; 321 - 25 ;
; 322 - 26 ;
; 323 - 27 ;
; 324 - 28 ;
; 325 - 29 ;
; 326 - 2A ;
; 327 - 2B ;
; 328 - 2C ;
; 329 - 2D ;
; 330 - 2E ;
; 331 - 2F ;
; 332 - 30 ;
; 333 - 31 ;
; 334 - 32 ;
; 335 - 33 ;
; 336 - 34 ;
; 337 - 35 ;
; 338 - 36 ;
; 339 - 37 ;
; 340 - 38 ;
; 341 - 39 ;
; 342 - 3A ;
; 343 - 3B ;
; 344 - 3C ;
; 345 - 3D ;
; 346 - 3E ;
; 347 - 3F ;
; 348 - 40 ;
; 349 - 41 ;
; 350 - 42 ;
; 351 - 43 ;
; 352 - 44 ;
; 353 - 45 ;
; 354 - 46 ;
; 355 - 47 ;
; 356 - 48 ;
; 357 - 49 ;
; 358 - 4A ;
; 359 - 4B ;
; 360 - 4C ;
; 361 - 4D ;
; 362 - 4E ;
; 363 - 4F ;
; 364 - 50 ;
; 365 - 51 ;
; 366 - 52 ;
; 367 - 53 ;
; 368 - 54 ;
; 369 - 55 ;
; 370 - 56 ;
; 371 - 57 ;
; 372 - 58 ;
; 373 - 59 ;
; 374 - 5A ;
; 375 - 5B ;
; 376 - 5C ;
; 377 - 5D ;
; 378 - 5E ;
; 379 - 5F ;
; 380 - 60 ;
; 381 - 61 ;
; 382 - 62 ;
; 383 - 63 ;
; 384 - 64 ;
; 385 - 65 ;
; 386 - 66 ;
; 387 - 67 ;
; 388 - 68 ;
; 389 - 69 ;
; 390 - 6A ;
; 391 - 6B ;
; 392 - 6C ;
; 393 - 6D ;
; 394 - 6E ;
; 395 - 6F ;
; 396 - 70 ;
; 397 - 71 ;
; 398 - 72 ;
; 399 - 73 ;
; 400 - 74 ;
; 401 - 75 ;
; 402 - 76 ;
; 403 - 77 ;
; 404 - 78 ;
; 405 - 79 ;
; 406 - 7A ;
; 407 - 7B ;
; 408 - 7C ;
; 409 - 7D ;
; 410 - 7E ;
; 411 - 7F ;
; 412 - 80 ;
; 413 - 81 ;
; 414 - 82 ;
; 415 - 83 ;
; 416 - 84 ;
; 417 - 85 ;
; 418 - 86 ;
; 419 - 87 ;
; 420 - 88 ;
; 421 - 89 ;
; 422 - 8A ;
; 423 - 8B ;
; 424 - 8C ;
; 425 - 8D ;
; 426 - 8E ;
; 427 - 8F ;
; 428 - 90 ;
; 429 - 91 ;
; 430 - 92 ;
; 431 - 93 ;
; 432 - 94 ;
; 433 - 95 ;
; 434 - 96 ;
; 435 - 97 ;
; 436 - 98 ;
; 437 - 99 ;
; 438 - 9A ;
; 439 - 9B ;
; 440 - 9C ;
; 441 - 9D ;
; 442 - 9E ;
; 443 - 9F ;
; 444 - A0 ;
; 445 - A1 ;
; 446 - A2 ;
; 447 - A3 ;
; 448 - A4 ;
; 449 - A5 ;
; 450 - A6 ;
; 451 - A7 ;
; 452 - A8 ;
; 453 - A9 ;
; 454 - AA ;
; 455 - AB ;
; 456 - AC ;
; 457 - AD ;
; 458 - AE ;
; 459 - AF ;
; 460 - B0 ;
; 461 - B1 ;
; 462 - B2 ;
; 463 - B3 ;
; 464 - B4 ;
; 465 - B5 ;
; 466 - B6 ;
; 467 - B7 ;
; 468 - B8 ;
; 469 - B9 ;
; 470 - BA ;
; 471 - BB ;
; 472 - BC ;
; 473 - BD ;
; 474 - BE ;
; 475 - BF ;
; 476 - C0 ;
; 477 - C1 ;
; 478 - C2 ;
; 479 - C3 ;
; 480 - C4 ;
; 481 - C5 ;
; 482 - C6 ;
; 483 - C7 ;
; 484 - C8 ;
; 485 - C9 ;
; 486 - CA ;
; 487 - CB ;
; 488 - CC ;
; 489 - CD ;
; 490 - CE ;
; 491 - CF ;
; 492 - D0 ;
; 493 - D1 ;
; 494 - D2 ;
; 495 - D3 ;
; 496 - D4 ;
; 497 - D5 ;
; 498 - D6 ;
; 499 - D7 ;
; 500 - D8 ;
; 501 - D9 ;
; 502 - DA ;
; 503 - DB ;
; 504 - DC ;
; 505 - DD ;
; 506 - DE ;
; 507 - DF ;
; 508 - E0 ;
; 509 - E1 ;
; 510 - E2 ;
; 511 - E3 ;
; 512 - E4 ;
; 513 - E5 ;
; 514 - E6 ;
; 515 - E7 ;
; 516 - E8 ;
; 517 - E9 ;
; 518 - EA ;
; 519 - EB ;
; 520 - EC ;
; 521 - ED ;
; 522 - EE ;
; 523 - EF ;
; 524 - F0 ;
; 525 - F1 ;
; 526 - F2 ;
; 527 - F3 ;
; 528 - F4 ;
; 529 - F5 ;
; 530 - F6 ;
; 531 - F7 ;
; 532 - F8 ;
; 533 - F9 ;
; 534 - FA ;
; 535 - FB ;
; 536 - FC ;
; 537 - FD ;
; 538 - FE ;
; 539 - FF ;
; 540 - 00 ;
; 541 - 01 ;
; 542 - 02 ;
; 543 - 03 ;
; 544 - 04 ;
; 545 - 05 ;
; 546 - 06 ;
; 547 - 07 ;
; 548 - 08 ;
; 549 - 09 ;
; 550 - 0A ;
; 551 - 0B ;
; 552 - 0C ;
; 553 - 0D ;
; 554 - 0E ;
; 555 - 0F ;
; 556 - 10 ;
; 557 - 11 ;
; 558 - 12 ;
; 559 - 13 ;
; 560 - 14 ;
; 561 - 15 ;
; 562 - 16 ;
; 563 - 17 ;
; 564 - 18 ;
; 565 - 19 ;
; 566 - 1A ;
; 567 - 1B ;
; 568 - 1C ;
; 569 - 1D ;
; 570 - 1E ;
; 571 - 1F ;
; 572 - 20 ;
; 573 - 21 ;
; 574 - 22 ;
; 575 - 23 ;
; 576 - 24 ;
; 577 - 25 ;
; 578 - 26 ;
; 579 - 27 ;
; 580 - 28 ;
; 581 - 29 ;
; 582 - 2A ;
; 583 - 2B ;
; 584 - 2C ;
; 585 - 2D ;
; 586 - 2E ;
; 587 - 2F ;
; 588 - 30 ;
; 589 - 31 ;
; 590 - 32 ;
; 591 - 33 ;
; 592 - 34 ;
; 593 - 35 ;
; 594 - 36 ;
; 595 - 37 ;
; 596 - 38 ;
; 597 - 39 ;
; 598 - 3A ;
; 599 - 3B ;
; 600 - 3C ;
; 601 - 3D ;
; 602 - 3E ;
; 603 - 3F ;
; 604 - 40 ;
; 605 - 41 ;
; 606 - 42 ;
; 607 - 43 ;
; 608 - 44 ;
; 609 - 45 ;
; 610 - 46 ;
; 611 - 47 ;
; 612 - 48 ;
; 613 - 49 ;
; 614 - 4A ;
; 615 - 4B ;
; 616 - 4C ;
; 617 - 4D ;
; 618 - 4E ;
; 619 - 4F ;
; 620 - 50 ;
; 621 - 51 ;
; 622 - 52 ;
; 623 - 53 ;
; 624 - 54 ;
; 625 - 55 ;
; 626 - 56 ;
; 627 - 57 ;
; 628 - 58 ;
; 629 - 59 ;
; 630 - 5A ;
; 631 - 5B ;
; 632 - 5C ;
; 633 - 5D ;
; 634 - 5E ;
; 635 - 5F ;
; 636 - 60 ;
; 637 - 61 ;
; 638 - 62 ;
; 639 - 63 ;
; 640 - 64 ;
; 641 - 65 ;
; 642 - 66 ;
; 643 - 67 ;
; 644 - 68 ;
; 645 - 69 ;
; 646 - 6A ;
; 647 - 6B ;
; 648 - 6C ;
; 649 - 6D ;
; 650 - 6E ;
; 651 - 6F ;
; 652 - 70 ;
; 653 - 71 ;
; 654 - 72 ;
; 655 - 73 ;
; 656 - 74 ;
; 657 - 75 ;
; 658 - 76 ;
; 659 - 77 ;
; 660 - 78 ;
; 661 - 79 ;
; 662 - 7A ;
; 663 - 7B ;
; 664 - 7C ;
; 665 - 7D ;
; 666 - 7E ;
; 667 - 7F ;
; 668 - 80 ;
; 669 - 81 ;
; 670 - 82 ;
; 671 - 83 ;
; 672 - 84 ;
; 673 - 85 ;
; 674 - 86 ;
; 675 - 87 ;
; 676 - 88 ;
; 677 - 89 ;
; 678 - 8A ;
; 679 - 8B ;
; 680 - 8C ;
; 681 - 8D ;
; 682 - 8E ;
; 683 - 8F ;
; 684 - 90 ;
; 685 - 91 ;
; 686 - 92 ;
; 687 - 93 ;
; 688 - 94 ;
; 689 - 95 ;
; 690 - 96 ;
; 691 - 97 ;
; 692 - 98 ;
; 693 - 99 ;
; 694 - 9A ;
; 695 - 9B ;
; 696 - 9C ;
; 697 - 9D ;
; 698 - 9E ;
; 699 - 9F ;
; 700 - A0 ;
; 701 - A1 ;
; 702 - A2 ;
; 703 - A3 ;
; 704 - A4 ;
; 705 - A5 ;
; 706 - A6 ;
; 707 - A7 ;
; 708 - A8 ;
; 709 - A9 ;
; 710 - AA ;
; 711 - AB ;
; 712 - AC ;
; 713 - AD ;
; 714 - AE ;
; 715 - AF ;
; 716 - B0 ;
; 717 - B1 ;
; 718 - B2 ;
; 719 - B3 ;
; 720 - B4 ;
; 721 - B5 ;
; 722 - B6 ;
; 723 - B7 ;
; 724 - B8 ;
; 725 - B9 ;
; 726 - BA ;
; 727 - BB ;
; 728 - BC ;
; 729 - BD ;
; 730 - BE ;
; 731 - BF ;
; 732 - C0 ;
; 733 - C1 ;
; 734 - C2 ;
; 735 - C3 ;
; 736 - C4 ;
; 737 - C5 ;
; 738 - C6 ;
; 739 - C7 ;
; 740 - C8 ;
; 741 - C9 ;
; 742 - CA ;
; 743 - CB ;
; 744 - CC ;
; 745 - CD ;
; 746 - CE ;
; 747 - CF ;
; 748 - D0 ;
; 749 - D1 ;
; 750 - D2 ;
; 751 - D3 ;
; 752 - D4 ;
; 753 - D5 ;
; 754 - D6 ;
; 755 - D7 ;
; 756 - D8 ;
; 757 - D9 ;
; 758 - DA ;
; 759 - DB ;
; 760 - DC ;
; 761 - DD ;
; 762 - DE ;
; 763 - DF ;
; 764 - E0 ;
; 765 - E1 ;
; 766 - E2 ;
; 767 - E3 ;
; 768 - E4 ;
; 769 - E5 ;
; 770 - E6 ;
; 771 - E7 ;
; 772 - E8 ;
; 773 - E9 ;
; 774 - EA ;
; 775 - EB ;
; 776 - EC ;
; 777 - ED ;
; 778 - EE ;
; 779 - EF ;
; 780 - F0 ;
; 781 - F1 ;
; 782 - F2 ;
; 783 - F3 ;
; 784 - F4 ;
; 785 - F5 ;
; 786 - F6 ;
; 787 - F7 ;
; 788 - F8 ;
; 789 - F9 ;
; 790 - FA ;
; 791 - FB ;
; 792 - FC ;
; 793 - FD ;
; 794 - FE ;
; 795 - FF ;
; 796 - 00 ;
; 797 - 01 ;
; 798 - 02 ;
; 799 - 03 ;
; 800 - 04 ;
; 801 - 05 ;
; 802 - 06 ;
; 803 - 07 ;
; 804 - 08 ;
; 805 - 09 ;
; 806 - 0A ;
; 807 - 0B ;
; 808 - 0C ;
; 809 - 0D ;
; 810 - 0E ;
; 811 - 0F ;
; 812 - 10 ;
; 813 - 11 ;
; 814 - 12 ;
; 815 - 13 ;
; 816 - 14 ;
; 817 - 15 ;
; 818 - 16 ;
; 819 - 17 ;
; 820 - 18 ;
; 821 - 19 ;
; 822 - 1A ;
; 823 - 1B ;
; 824 - 1C ;
; 825 - 1D ;
; 826 - 1E ;
; 827 - 1F ;
; 828 - 20 ;
; 829 - 21 ;
; 830 - 22 ;
; 831 - 23 ;
; 832 - 24 ;
; 833 - 25 ;
; 834 - 26 ;
; 835 - 27 ;
; 836 - 28 ;
; 837 - 29 ;
; 838 - 2A ;
; 839 - 2B ;
; 840 - 2C ;
; 841 - 2D ;
; 842 - 2E ;
; 843 - 2F ;
; 844 - 30 ;
; 845 - 31 ;
; 846 - 32 ;
; 847 - 33 ;
; 848 - 34 ;
; 849 - 35 ;
; 850 - 36 ;
; 851 - 37 ;
; 852 - 38 ;
; 853 - 39 ;
; 854 - 3A ;
; 855 - 3B ;
; 856 - 3C ;
; 857 - 3D ;
; 858 - 3E ;
; 859 - 3F ;
; 860 - 40 ;
; 861 - 41 ;
; 862 - 42 ;
; 863 - 43 ;
; 864 - 44 ;
; 865 - 45 ;
; 866 - 46 ;
; 867 - 47 ;
; 868 - 48 ;
; 869 - 49 ;
; 870 - 4A ;
; 871 - 4B ;
; 872 - 4C ;
; 873 - 4D ;
; 874 - 4E ;
; 875 - 4F ;
; 876 - 50 ;
; 877 - 51 ;
; 878 - 52 ;
; 879 - 53 ;
; 880 - 54 ;
; 881 - 55 ;
; 882 - 56 ;
; 883 - 57 ;
; 884 - 58 ;
; 885 - 59 ;
; 886 - 5A ;
; 887 - 5B ;
; 888 - 5C ;
; 889 - 5D ;
; 890 - 5E ;
; 891 - 5F ;
; 892 - 60 ;
; 893 - 61 ;
; 894 - 62 ;
; 895 - 63 ;
; 896 - 64 ;
; 897 - 65 ;
; 898 - 66 ;
; 899 - 67 ;
; 900 - 68 ;
; 901 - 69 ;
; 902 - 6A ;
; 903 - 6B ;
; 904 - 6C ;
; 905 - 6D ;
; 906 - 6E ;
; 907 - 6F ;
; 908 - 70 ;
; 909 - 71 ;
; 910 - 72 ;
; 911 - 73 ;
; 912 - 74 ;
; 913 - 75 ;
; 914 - 76 ;
; 915 - 77 ;
; 916 - 78 ;
; 917 - 79 ;
; 918 - 7A ;
; 919 - 7B ;
; 920 - 7C ;
; 921 - 7D ;
; 922 - 7E ;
; 923 - 7F ;
; 924 - 80 ;
; 925 - 81 ;
; 926 - 82 ;
; 927 - 83 ;
; 928 - 84 ;
; 929 - 85 ;
; 930 - 86 ;
; 931 - 87 ;
; 932 - 88 ;
; 933 - 89 ;
; 934 - 8A ;
; 935 - 8B ;
; 936 - 8C ;
; 937 - 8D ;
; 938 - 8E ;
; 939 - 8F ;
; 940 - 90 ;
; 941 - 91 ;
; 942 - 92 ;
; 943 - 93 ;
; 944 - 94 ;
; 945 - 95 ;
; 946 - 96 ;
; 947 - 97 ;
; 948 - 98 ;
; 949 - 99 ;
; 950 - 9A ;
; 951 - 9B ;
; 952 - 9C ;
; 953 - 9D ;
; 954 - 9E ;
; 955 - 9F ;
; 956 - A0 ;
; 957 - A1 ;
; 958 - A2 ;
; 959 - A3 ;
; 960 - A4 ;
; 961 - A5 ;
; 962 - A6 ;
; 963 - A7 ;
; 964 - A8 ;
; 965 - A9 ;
; 966 - AA ;
; 967 - AB ;
; 968 - AC ;
; 969 - AD ;
; 970 - AE ;
; 971 - AF ;
; 972 - B0 ;
; 973 - B1 ;
; 974 - B2 ;
; 975 - B3 ;
; 976 - B4 ;
; 977 - B5 ;
; 978 - B6 ;
; 979 - B7 ;
; 980 - B8 ;
; 981 - B9 ;
; 982 - BA ;
; 983 - BB ;
; 984 - BC ;
; 985 - BD ;
; 986 - BE ;
; 987 - BF ;
; 988 - C0 ;
; 989 - C1 ;
; 990 - C2 ;
; 991 - C3 ;
; 992 - C4 ;
; 993 - C5 ;
; 994 - C6 ;
; 995 - C7 ;
; 996 - C8 ;
; 997 - C9 ;
; 998 - CA ;
; 999 - CB ;
; 1000 - CC ;
; 1001 - CD ;
; 1002 - CE ;
; 1003 - CF ;
; 1004 - D0 ;
; 1005 - D1 ;
; 1006 - D2 ;
; 1007 - D3 ;
; 1008 - D4 ;
; 1009 - D5 ;
; 1010 - D6 ;
; 1011 - D7 ;
; 1012 - D8 ;
; 1013 - D9 ;
; 1014 - DA ;
; 1015 - DB ;
; 1016 - DC ;
; 1017 - DD ;
; 1018 - DE ;
; 1019 - DF ;
; 1020 - E0 ;
; 1021 - E1 ;
; 1022 - E2 ;
; 1023 - E3 ;
; 1024 - E4 ;
; 1025 - E5 ;
; 1026 - E6 ;
; 1027 - E7 ;
; 1028 - E8 ;
; 1029 - E9 ;
; 1030 - EA ;
; 1031 - EB ;
; 1032 - EC ;
; 1033 - ED ;
; 1034 - EE ;
; 1035 - EF ;
; 1036 - F0 ;
; 1037 - F1 ;
; 1038 - F2 ;
; 1039 - F3 ;
; 1040 - F4 ;
; 1041 - F5 ;
; 1042 - F6 ;
; 1043 - F7 ;
; 1044 - F8 ;
; 1045 - F9 ;
; 1046 - FA ;
; 1047 - FB ;
; 1048 - FC ;
; 1049 - FD ;
; 1050 - FE ;
; 1051 - FF ;
; 1052 - 00 ;
; 1053 - 01 ;
; 1054 - 02 ;
; 1055 - 03 ;
; 1056 - 04 ;
; 1057 - 05 ;
; 1058 - 06 ;
; 1059 - 07 ;
; 1060 - 08 ;
; 1061 - 09 ;
; 1062 - 0A ;
; 1063 - 0B ;
; 1064 - 0C ;
; 1065 - 0D ;
; 1066 - 0E ;
; 1067 - 0F ;
;
```


04.07.83	04.07.83	04.07.83	04.07.83
01 ORCE	01 OBFC 13	DEFB 13H	DEFB 13H
02 ORCE	02 OBFD 14	DEFB 14H	DEFB 14H
03 ORCE	03 OBFE 15	DEFB 15H	DEFB 15H
04 ORCE	04 OBFF 16	DEFB 16H	DEFB 16H
05 ORCE	05 OC00 17	DEFB 17H	DEFB 17H
06 ORCE	06 OG01 18	DEFB 18H	DEFB 18H
07 ORCE	07 OC02 19	DEFB 19H	DEFB 19H
08 ORCE	08 OC03 0A	DEFB 0AH	DEFB 0AH
09 ORCE	09 OC04 0B	DEFB 0BH	DEFB 0BH
10 ORCE	10 OC05 0C	DEFB 0CH	DEFB 0CH
11 ORCE	11 OC06 0D	DEFB 0DH	DEFB 0DH
12 ORCE	12 OC07 0E	DEFB 0EH	DEFB 0EH
13 ORCE	13 OC08 0F	DEFB 0FH	DEFB 0FH
14 ORCE	14 OC09 10	DEFB 10H	DEFB 10H
15 ORCE	15 OC0A 01	DEFB 01H	DEFB 01H
16 ORCE	16 OC0B 02	DEFB 02H	DEFB 02H
17 ORCE	17 OC0C 03	DEFB 03H	DEFB 03H
18 ORCE	18 OC0D 04	DEFB 04H	DEFB 04H
19 ORCE	19 OC0E 05	DEFB 05H	DEFB 05H
20 ORCE	20 OC0F 06	DEFB 06H	DEFB 06H
21 ORCE	21 OC10 07	DEFB 07H	DEFB 07H
22 ORCE	22 OC11 08	DEFB 08H	DEFB 08H
23 ORCE	23 OC12 21	DEFB 21H	DEFB 21H
24 ORCE	24 OC13 22	DEFB 22H	DEFB 22H
25 ORCE	25 OC14 23	DEFB 23H	DEFB 23H
26 ORCE	26 OC15 24	DEFB 24H	DEFB 24H
27 ORCE	27 OC16 25	DEFB 25H	DEFB 25H
28 ORCE	28 OC17 26	DEFB 26H	DEFB 26H
29 ORCE	29 OC18 27	DEFB 27H	DEFB 27H
30 ORCE	30 OC19 28	DEFB 28H	DEFB 28H
31 ORCE	31 OC1A 59	DEFB 59H	DEFB 59H
32 ORCE	32 OC1B 50	DEFB 50H	DEFB 50H
33 ORCE	33 OC1C 2A	DEFB 2AH	DEFB 2AH
34 ORCE	34 OC1D 00	DEFB 00H	DEFB 00H
35 ORCE	35 OC1E 20	DEFB 20H	DEFB 20H
36 ORCE	36 OC1F 29	DEFB 29H	DEFB 29H
37 ORCE	37 OC20 2F	DEFB 2FH	DEFB 2FH
38 ORCE	38 OC21 2E	DEFB 2EH	DEFB 2EH
39 ORCE	39 OC22 C8	DEFB C8H	DEFB C8H
40 ORCE	40 OC23 C7	DEFB C7H	DEFB C7H
41 ORCE	41 OC24 C2	DEFB C2H	DEFB C2H
42 ORCE	42 OC25 C1	DEFB C1H	DEFB C1H
43 ORCE	43 OC26 C3	DEFB C3H	DEFB C3H
44 ORCE	44 OC27 C4	DEFB C4H	DEFB C4H
45 ORCE	45 OC28 49	DEFB 49H	DEFB 49H
46 ORCE	46 OC29 2D	DEFB 2DH	DEFB 2DH
47 ORCE	47 OC2A	DEFB 2DH	DEFB 2DH
48 ORCE	48 OC2B	DEFB 2DH	DEFB 2DH
49 ORCE	49 OC2C	DEFB 2DH	DEFB 2DH
50 ORCE	50 OC2D	DEFB 2DH	DEFB 2DH
51 ORCE	51 OC2E	DEFB 2DH	DEFB 2DH
52 ORCE	52 OC2F	DEFB 2DH	DEFB 2DH
53 ORCE	53 OC30	DEFB 2DH	DEFB 2DH
54 ORCE	54 OC31	DEFB 2DH	DEFB 2DH
55 ORCE	55 OC32	DEFB 2DH	DEFB 2DH
56 ORCE	56 OC33	DEFB 2DH	DEFB 2DH
57 ORCE	57 OC34	DEFB 2DH	DEFB 2DH
58 ORCE	58 OC35	DEFB 2DH	DEFB 2DH
59 ORCE	59 OC36	DEFB 2DH	DEFB 2DH
60 ORCE	60 OC37	DEFB 2DH	DEFB 2DH
61 ORCE	61 OC38	DEFB 2DH	DEFB 2DH
62 ORCE	62 OC39	DEFB 2DH	DEFB 2DH
63 ORCE	63 OC3A	DEFB 2DH	DEFB 2DH
64 ORCE	64 OC3B	DEFB 2DH	DEFB 2DH
65 ORCE	65 OC3C	DEFB 2DH	DEFB 2DH
66 ORCE	66 OC3D	DEFB 2DH	DEFB 2DH
67 ORCE	67 OC3E	DEFB 2DH	DEFB 2DH
68 ORCE	68 OC3F	DEFB 2DH	DEFB 2DH
69 ORCE	69 OC40	DEFB 2DH	DEFB 2DH
70 ORCE	70 OC41	DEFB 2DH	DEFB 2DH
71 ORCE	71 OC42	DEFB 2DH	DEFB 2DH
72 ORCE	72 OC43	DEFB 2DH	DEFB 2DH
73 ORCE	73 OC44	DEFB 2DH	DEFB 2DH
74 ORCE	74 OC45	DEFB 2DH	DEFB 2DH
75 ORCE	75 OC46	DEFB 2DH	DEFB 2DH
76 ORCE	76 OC47	DEFB 2DH	DEFB 2DH
77 ORCE	77 OC48	DEFB	

**	Z80	ASSEMBLER	SB-7201	<17-013A>	PAGE	47	04.07.83	**	Z80	ASSEMBLER	SB-7201	<17-013A>	PAGE	48	04.07.83
01	0C2E	F0		DEFB	FOH		; NO	01	0C63	C5		DEFB	C5H		; HOME
02	0C2F	6A		DEFB	6AH		; +	02	0C64	C2		DEFB	C2H		; CURSOR UP
03	0C30	6B		DEFB	6BH		; *	03	0C65	C1		DEFB	C1H		; CURSOR DOWN
04	0C31	CD		DEFB	CDH		; CR	04	0C66	C3		DEFB	C3H		; CURSOR RIGHT
05	0C32		; S1	0B-0F				05	0C67	C4		DEFB	C4H		; CURSOR LEFT
06	0C32	99		DEFB	99H		; y	06	0C68	5A		DEFB	5AH		; ↵
07	0C33	9A		DEFB	9AH		; z	07	0C69	45		DEFB	45H		; ↵
08	0C34	A4		DEFB	A4H		; <	08	0C6A						
09	0C35	BC		DEFB	BCH		; {	09	0C6A			GRAPHIC			
10	0C36	40		DEFB	40H		; }	10	0C6A						
11	0C37	F0		DEFB	F0H		; NULL	11	0C6A			KTBLGS: ENT			
12	0C38	F0		DEFB	F0H		; NULL	12	0C6A			ISO 00-07			
13	0C39	F0		DEFB	F0H		; NULL	13	0C6A	BF		DEFB	BFH		; SPARE
14	0C3A			DEFB				14	0C6B	F0		DEFB	FOH		; GRAPH BUT NULL
15	0C3A	91		DEFB	91H		; q	15	0C6C	E5		DEFB	E5H		; #↑
16	0C3B	92		DEFB	92H		; r	16	0C6D	C9		DEFB	C9H		; ALPHA
17	0C3C	93		DEFB	93H		; s	17	0C6E	F0		DEFB	F0H		; NO
18	0C3D	94		DEFB	94H		; t	18	0C6F	42		DEFB	42H		; #;
19	0C3E	95		DEFB	95H		; u	19	0C70	B6		DEFB	B6H		; #:
20	0C3F	96		DEFB	96H		; v	20	0C71	CD		DEFB	CDH		; CR
21	0C40	97		DEFB	97H		; w	21	0C72			0B-0F			
22	0C41	98		DEFB	98H		; x	22	0C72	75		DEFB	75H		; #Y
23	0C42			DEFB				23	0C73	76		DEFB	76H		; #Z
24	0C42	89		DEFB	89H		; i	24	0C74	B2		DEFB	B2H		; #@
25	0C43	8A		DEFB	8AH		; j	25	0C75	D8		DEFB	D8H		; #!C
26	0C44	8B		DEFB	8BH		; k	26	0C76	4E		DEFB	4EH		; #!
27	0C45	8C		DEFB	8CH		; l	27	0C77	F0		DEFB	F0H		; #NULL-
28	0C46	8D		DEFB	8DH		; m	28	0C78	F0		DEFB	FOH		; #NULL-
29	0C47	8E		DEFB	8EH		; n	29	0C79	F0		DEFB	FOH		; #NULL
30	0C48	8F		DEFB	8FH		; o	30	0C7A			10-17			
31	0C49	90		DEFB	90H		; p	31	0C7A	3C		DEFB	3CH		; #Q
32	0C4A			DEFB				32	0C7B	30		DEFB	30H		; #R
33	0C4A	81		DEFB	81H		; a	33	0C7C	44		DEFB	44H		; #S
34	0C4B	82		DEFB	82H		; b	34	0C7D	71		DEFB	71H		; #T
35	0C4C	83		DEFB	83H		; c	35	0C7E	79		DEFB	79H		; #U
36	0C4D	84		DEFB	84H		; d	36	0C7F	DA		DEFB	DAH		; #V
37	0C4E	85		DEFB	85H		; e	37	0C80	38		DEFB	38H		; #W
38	0C4F	86		DEFB	86H		; f	38	0C81	6D		DEFB	6DH		; #X
39	0C50	87		DEFB	87H		; g	39	0C82			1B-1F			
40	0C51	88		DEFB	88H		; h	40	0C82	7D		DEFB	7DH		; #I
41	0C52			DEFB				41	0C83	5C		DEFB	5CH		; #J
42	0C52	61		DEFB	61H		; i	42	0C84	5B		DEFB	5BH		; #K
43	0C53	62		DEFB	62H		; "	43	0C85	B4		DEFB	B4H		; #L
44	0C54	63		DEFB	63H		; #	44	0C86	1C		DEFB	1CH		; #M
45	0C55	64		DEFB	64H		; \$	45	0C87	32		DEFB	32H		; #N
46	0C56	65		DEFB	65H		; %	46	0C88	B0		DEFB	B0H		; #O
47	0C57	66		DEFB	66H		; &	47	0C89	D6		DEFB	D6H		; #P
48	0C58	67		DEFB	67H		; '	48	0C8A			20-27			
49	0C59	68		DEFB	68H		; (49	0C8A	53		DEFB	53H		; #A
50	0C5A			DEFB				50	0C8B	6F		DEFB	6FH		; #B
51	0C5A	80		DEFB	80H		; \	51	0C8C	DE		DEFB	DEH		; #C
52	0C5B	A5		DEFB	A5H		; /	52	0C8D	47		DEFB	47H		; #D
53	0C5C	2B		DEFB	2BH		; POND MARK	53	0C8E	34		DEFB	34H		; #E
54	0C5D	00		DEFB	00H		; YEN	54	0C8F	4A		DEFB	4AH		; #F
55	0C5E	60		DEFB	60H		; SPACE	55	0C90	4B		DEFB	4BH		; #G
56	0C5F	69		DEFB	69H		; π	56	0C91	72		DEFB	72H		; #H
57	0C60	51		DEFB	51H		;)	57	0C92			2B-2F			
58	0C61	57		DEFB	57H		; >	58	0C92	37		DEFB	37H		; #1
59	0C62			DEFB			; <	59	0C93	3E		DEFB	3EH		; #2
60	0C62	C6		DEFB	C6H		; CLR	60	0C94	7F		DEFB	7FH		; #3

```

** Z80 ASSEMBLER SB-7201 <1Z-013A> PAGE 49 04.07.83
01 0C95 7B DEF8 7BH ; #4
02 0C96 3A DEF8 3AH ; #5
03 0C97 5E DEF8 5EH ; #6
04 0C98 1F DEF8 1FH ; #7
05 0C99 BD DEF8 BDH ; #8
06 0C9A D4 DEF8 D4H ; #YEN
07 0C9B 9E DEF8 9EH ; #+
08 0C9C D2 DEF8 D2H ; #-
09 0C9D 00 DEF8 00H ; SPACE
10 0C9E 9C DEF8 9CH ; #0
11 0C9F A1 DEF8 A1H ; #9
12 0CA0 CA DEF8 CAH ; #,
13 0CA1 B8 DEF8 B8H ; #.
14 0CA2 C8 DEF8 C8H ; INST
15 0CA3 C7 DEF8 C7H ; DEL.
16 0CA4 C2 DEF8 C2H ; CURSOR UP
17 0CA5 C1 DEF8 C1H ; CURSOR DOWN
18 0CA6 C3 DEF8 C3H ; CURSOR RIGHT
19 0CA7 C4 DEF8 C4H ; CURSOR LEFT
20 0CA8 BA DEF8 BAH ; #?
21 0CA9 DB DEF8 DBH ; #/
22 0CAA ;
23 0CAA ; CONTROL CODE
24 0CAA ;
25 0CAA ;
26 0CAA ;
27 0CAA ;
28 0CAA 00-07N KTBLG: ENT
29 0CAB F0 DEF8 F0H ;S0
30 0CAC F0 DEF8 F0H
31 0CAC F0 DEF8 F0H
32 0CAD F0 DEF8 F0H
33 0CAE F0 DEF8 F0H
34 0CAF F0 DEF8 F0H
35 0CB0 F0 DEF8 F0H
36 0CB1 F0 DEF8 F0H
37 0CB2 F0 DEF8 F0H
38 0CB2 F0 DEF8 F0H
39 0CB3 5A DEF8 5AH ; #Y E3 (CHECKER)
40 0CB4 F0 DEF8 F0H ; #Z E4
41 0CB5 F0 DEF8 F0H ; #Q
42 0CB6 F0 DEF8 F0H ; #L E5
43 0CB7 F0 DEF8 F0H ; #J E7
44 0CB8 F0 DEF8 F0H
45 0CB9 F0 DEF8 F0H
46 0CBA C1 DEF8 C1H
47 0CBA C2 DEF8 C2H
48 0CBC C3 DEF8 C3H
49 0CBD C4 DEF8 C4H
50 0CBE C5 DEF8 C5H
51 0CBF C6 DEF8 C6H
52 0CC0 F0 DEF8 F0H ; #V E1
53 0CC1 F0 DEF8 F0H ; #W E2
54 0CC2 F0 DEF8 F0H ; #X
55 0CC2 F0 DEF8 F0H
56 0CC2 F0 DEF8 F0H ; #I F9
57 0CC3 F0 DEF8 F0H ; #J FA
58 0CC4 F0 DEF8 F0H ; #K FB
59 0CC5 F0 DEF8 F0H ; #L FC
60 0CC6 F0 DEF8 F0H ; #M FD

** Z80 ASSEMBLER SB-7201 <1Z-013A> PAGE 50 04.07.83
01 0CC7 F0 DEF8 F0H ; #N FE
02 0CC8 F0 DEF8 F0H ; #O FF
03 0CC9 F0 DEF8 F0H ; #P EO
04 0CCA 20-27 DEF8 20-27 ;S4
05 0CCA F0 DEF8 F0H
06 0CCB F0 DEF8 F0H
07 0CCC F0 DEF8 F0H
08 0CCD F0 DEF8 F0H
09 0CCE F0 DEF8 F0H
10 0CCF F0 DEF8 F0H
11 0CD0 F0 DEF8 F0H ; #F F6
12 0CD1 F0 DEF8 F0H ; #G F7
13 0CD2 28-2F DEF8 28-2F ;S5
14 0CD2 F0 DEF8 F0H
15 0CD3 F0 DEF8 F0H
16 0CD4 F0 DEF8 F0H
17 0CD5 F0 DEF8 F0H
18 0CD6 F0 DEF8 F0H
19 0CD7 F0 DEF8 F0H
20 0CD8 F0 DEF8 F0H
21 0CD9 F0 DEF8 F0H
22 0CDA 30-37 DEF8 30-37 ;S6
23 0CDA F0 DEF8 F0H
24 0CDB F0 DEF8 F0H
25 0CDC F0 DEF8 F0H
26 0CDD F0 DEF8 F0H
27 0CDE F0 DEF8 F0H
28 0CDF F0 DEF8 F0H
29 0CEO F0 DEF8 F0H ;S7
30 0CE1 38-3F DEF8 38-3F
31 0CE1 F0 DEF8 F0H
32 0CE2 F0 DEF8 F0H
33 0CE3 F0 DEF8 F0H
34 0CE4 F0 DEF8 F0H
35 0CE5 F0 DEF8 F0H
36 0CE6 F0 DEF8 F0H
37 0CE7 F0 DEF8 F0H
38 0CE8 F0 DEF8 F0H
39 0CE9 ;
40 0CE9 ; KANA
41 0CE9 ;
42 0CE9 ;
43 0CE9 00-07 KTBLG: ENT
44 0CE9 BF DEF8 BFH ;S0
45 0CEA F0 DEF8 F0H
46 0CEB CF DEF8 CFH
47 0CEC C9 DEF8 C9H
48 0CED F0 DEF8 F0H
49 0CEE B5 DEF8 B5H
50 0CEF 4D DEF8 4DH
51 0CF0 CD DEF8 CDH
52 0CF1 0B-0F DEF8 0B-0F ;S1
53 0CF1 35 DEF8 35H
54 0CF2 77 DEF8 77H
55 0CF3 D7 DEF8 D7H
56 0CF4 B3 DEF8 B3H
57 0CF5 B7 DEF8 B7H
58 0CF6 F0 DEF8 F0H
59 0CF7 F0 DEF8 F0H
60 0CF8 F0 DEF8 F0H

```

** Z80 ASSEMBLER SB-7201	<17-013A>	PAGE 51	04.07.83	** Z80 ASSEMBLER SB-7201	<17-013A>	PAGE 52	04.07.83
01 0CF9	10-17	\$S2		01 0D2C CDA602	.4DE		
02 0CF9 7C	DEFB 7CH		; KA	02 0D2F E5	PUSH HL		
03 0CFA 70	DEFB 70H		; KE	03 0D30 CD1004	CALL HLHEX		; END ADR.
04 0CFB 41	DEFB 41H		; SH1	04 0D33 D1	POP DE		
05 0CFC 31	DEFB 31H		; K0	05 0D34 3B52	JR C,DUM1		; DATA ER. THEN
06 0CFD 39	DEFB 39H		; H1	06 0D36 EB	EX DE,H1		
07 0CFE A6	DEFB A6H		; TE	07 0D37 0608	LD B,OBH		; DISP 8BYTES
08 0CF7 78	DEFB 78H		; KI	08 0D39 OE17	LD LD		; CHA. PRINT BIAS
09 0D00 DD	DEFB DDH		; CH1	09 0D3B CDFAO5	CALL NPLH		; NEWLINE PRINT
10 0D01	18-1F	\$S3		10 0D3E CDB103	CALL SPHEX		; SP. PRT.+ACC PRT.
11 0D01 3D	DEFB 3DH		; FU	11 0D41 23	INC HL		
12 0D02 5D	DEFB 5DH		; M1	12 0D42 F5	PUSH AF		
13 0D03 6C	DEFB 6CH		; MU	13 0D43 3A7111	LD A,(DSPXY)		; DISPLAY POINT
14 0D04 56	DEFB 56H		; ME	14 0D46 B1	ADD A,C		
15 0D05 1D	DEFB 1DH		; RH1	15 0D47 327111	LD (DSPXY),A		; X AXIS.=X+Creg
16 0D06 33	DEFB 33H		; RA	16 0D4A F1	POP AF		
17 0D07 D5	DEFB D5H		; HE	17 0D4B FE20	CP 20H		
18 0D08 B1	DEFB B1H		; HO	18 0D4D 3002	JR NC,+4		
19 0D09	20-27	\$S4		19 0D4F 3E2E	LD A,2EH		; .
20 0D09 46	DEFB 46H		; SA	20 0D51 CDB90B	?ADCN		; ASCII TO DSPLAY CODE
21 0D0A 6E	DEFB 6EH		; TO	21 0D54 CD6C09	CALL PRNT3		
22 0D0B D9	DEFB D9H		; THU	22 0D57 3A7111	LD A,(DSPXY)		
23 0D0C 48	DEFB 48H		; SU	23 0D5A OC	INC C		
24 0D0D 74	DEFB 74H		; KU	24 0D5B 91	SUB C		
25 0D0E 43	DEFB 43H		; SE	25 0D5C 327111	LD (DSPXY),A		
26 0D0F 4C	DEFB 4CH		; SO	26 0D5F OD	DEC C		; ASCII DSP POSITION
27 0D10 73	DEFB 73H		; MA	27 0D60 OD	DEC C		
28 0D11	28-2F	\$S5		28 0D61 OD	DEC C		
29 0D11 3F	DEFB 3FH		; A	29 0D62 E5	PUSH HL		
30 0D12 36	DEFB 36H		; I	30 0D63 ED52	HL,DE		
31 0D13 7E	DEFB 7EH		; U	31 0D65 E1	POP HL		
32 0D14 3B	DEFB 3BH		; E	32 0D66 2B1D	JR Z,DUM1-3		
33 0D15 7A	DEFB 7AH		; O	33 0D68 3EFB	A,FBI		
34 0D16 1E	DEFB 1EH		; NA	34 0D6A 3200E0	LD (KEYPA),A		
35 0D17 5F	DEFB 5FH		; NI	35 0D6D 00	NOP		
36 0D18 A2	DEFB A2H		; NU	36 0D6E 3A01E0	A,(KEYPB)		
37 0D19	30-37	\$S6		37 0D71 FEFE	CP FEH		; SHIFT KEY ?
38 0D19 D3	DEFB D3H		; YO	38 0D73 2003	JR NZ,+5		
39 0D1A 9F	DEFB 9FH		; YU	39 0D75 CDA60D	?BLNK		; 64MSEC DELAY
40 0D1B D1	DEFB D1H		; YA	40 0D7B 10C4	DJNZ DUM2		
41 0D1C 00	DEFB 00H		; SPACE	41 0D7A CDCA0B	CALL ?KEY		; STOP DISP
42 0D1D 9D	DEFB 9DH		; NO	42 0D7D B7	OR A		
43 0D1E A3	DEFB A3H		; NE	43 0D7E 2BFA	JR Z,-4		; SPACE KEY THEN STOP
44 0D1F D0	DEFB D0H		; RU	44 0D80 CD320A	CALL ?BRK		; BREAK IN ?
45 0D20 B9	DEFB B9H		; RE	45 0D83 20B2	JR NZ,DUM3		
46 0D21	38-3F	\$S7		46 0D85 C3AD00	JP ST1		; COMMAND IN !
47 0D21 C6	DEFB C6H		; ?CLR @	47 0D8B 21A000	LD HL,160		; 20*B BYTE
48 0D22 C5	DEFB C5H		; ?HOME @	48 0D8B 19	ADD HL,DE		
49 0D23 C2	DEFB C2H		; ?CURSOR UP	49 0DBC 1BA8	JR DUM3-1		
50 0D24 C1	DEFB C1H		; ?CURSOR DOWN	50 DBE			
51 0D25 C3	DEFB C3H		; ?CURSOR RIGHT	51 DBE			
52 0D26 C4	DEFB C4H		; ?CURSOR LEFT	52 DBE			
53 0D27 BB	DEFB BBH		; DASH	53 DBE			
54 0D28 BE	DEFB BEH		; R0	54 DBE			

```

** Z80 ASSEMBLER SB-7201 <12-013A> PAGE 53                                04.07.83
01 0DA6 ; V-BLANK CHECK ;
02 0DA6 ;
03 0DA6 ?BLNK: ENT AF
04 0DA6 F5 PUSH AF
05 0DA7 3A02E0 LD A,(KEYFC)
06 0DA8 30FA RLC A
07 0DA9 30FA JR NC,-4
08 0DAD 3A02E0 LD A,(KEYFC)
09 0DB0 07 RLC A
10 0DB1 3BFA JR C,-4
11 0DB3 F1 POP AF
12 0DB4 C9 RET
13 0DB5 ;
14 0DB5 ;ORG 0DB5H;?DSP
15 0DB5 ;
16 0DB5 ;
17 0DB5 ;
18 0DB5 ; DISPLAY ON POINTER ;
19 0DB5 ;
20 0DB5 ; ACC = DISPLAY CODE
21 0DB5 ; EXCEPT FOH
22 0DB5 ;
23 0DB5 ?DSP: ENT
24 0DB5 F5 PUSH AF
25 0DB6 C5 PUSH BC
26 0DB7 D5 PUSH DE
27 0DB8 E5 PUSH HL
28 0DB9 ENT
29 0DB9 CDB10F CALL ?POINT
30 0DB9 LD (HL),A
31 0DBD 2A7111 LD HL,(DSPXY)
32 0DC0 7D LD A,L
33 0DC1 FE27 CP +39
34 0DC3 200B JR NZ,DSP04
35 0DC5 CDF302 CALL -MANG
36 0DCB 3B06 JR C,DSP04
37 0DCA EB EX DE,HL
38 0DCB 3601 LD (HL),+1
39 0DCD 23 INC HL
40 0DCE 3600 LD (HL),0
41 0DD0 ENT
42 0DD0 3EC3 LD A,C3H
43 0DD2 1B0C JR ?DPCT+4
44 0DD4 ;
45 0DD4 ;
46 0DD4 ;
47 0DD4 ; GRAPHIC STATUS CHECK
48 0DD4 ;
49 0DD4 ;
50 0DD4 3A7011 GRSTAS: LD A,(KANAF)
51 0DD7 FE01 CP 01H
52 0DD9 3ECA LD A,CAH
53 0DDB C9 RET
54 0DDC ;
55 0DDC ;
56 0DDC ;
57 0DDC ;
58 0DDC ;
59 0DDC ;
60 0DDC ;ORG 0DDCH;?DPCT

** Z80 ASSEMBLER SB-7201 <12-013A> PAGE 54                                04.07.83
01 0DDC ;
02 0DDC ;
03 0DDC ; DISPLAY CONTROL ;
04 0DDC ;
05 0DDC ; ACC = CONTROL CODE
06 0DDC ;
07 0DDC ?DPCT: ENT
08 0DDC F5 PUSH AF
09 0DDC C5 PUSH BC
10 0DDC D5 PUSH DE
11 0DDF E5 PUSH HL
12 0DE0 47 LD B,A
13 0DE1 E6F0 AND FOH
14 0DE3 FEC0 CP COH
15 0DE5 201B JR NZ,CURSS
16 0DE7 A8 XOR B
17 0DE8 07 RLC A
18 0DE9 4F LD C,A
19 0DEA 0600 LD B,+0
20 0DEC 21AA0E LD HL,CTBL
21 0DEF 09 ADD HL,BC
22 0DF0 5E LD E,(HL)
23 0DF1 23 INC HL
24 0DF2 56 LD D,(HL)
25 0DF3 2A7111 LD HL,(DSPXY)
26 0DF4 EB EX DE,HL
27 0DF7 E9 JP (HL)
28 0DF8 ;
29 0DF8 ;
30 0DF8 ;
31 0DF8 ; CURSD: ENT
32 0DF8 EB EX DE,HL
33 0DF9 7C LD A,H
34 0DFA FE18 CP +24
35 0DFC 2B25 JR Z,CURS4
36 0DFE 24 INC H
37 0DFF ENT
38 0DFF ;
39 0DFF ;
40 0DFF ;
41 0DFF ENT
42 0DFF 227111 LD (DSPXY),HL
43 0E02 C3E50E ?RSTR
44 0E05 ;
45 0E05 ;
46 0E05 ;
47 0E05 ;
48 0E07 B7 OR A
49 0E08 2BF8 JR Z,CURSS
50 0E0A 25 DEC H
51 0E0B ENT
52 0E0B 1BF2 CURSU1: ENT
53 0E0D JR CURS3
54 0E0D EB EX DE,HL
55 0E0E 7D LD A,L
56 0E0F FE27 CP +39
57 0E11 3003 JR NC,CURS2
58 0E13 2C LD L
59 0E14 1BE9 INC L
60 0E16 JR CURS3
ENT
CURS2: ENT

```



```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 55 04.07.83
01 0E16 2E00 LD L,+0
02 0E18 24 INC H
03 0E19 7C LD A,H
04 0E1A FE19 CP +25
05 0E1C 3BE1 JR C,CURS1
06 0E1E 2618 LD H,+24
07 0E20 227111 LD (DSPXY),HL
CURS4: ENT
08 0E23 1848 JR SCROL
09 0E25 ;
CURSL: ENT
10 0E25 EB ; LD HL,(DSPXY)
11 0E25 7D LD A,L
12 0E26 7D LD A,L
13 0E27 B7 OR A
14 0E28 2B03 JR Z,+5
15 0E2A 2D DEC L
16 0E2B 1BD2 JR CURS3
17 0E2D 2E27 LD L,+39
18 0E2F 25 DEC H
19 0E30 F20B0E JP P,CURSUI1
20 0E33 2600 LD H,0
21 0E35 227111 LD (DSPXY),HL
22 0E38 1BC8 JR CURS
;
CLRS: ENT
23 0E3A 217311 LD HL,MANG
24 0E3A 061B LD ?CLR
25 0E3F CDD80F CALL HL,DOOH
26 0E42 2100D0 LD #CLR0B
27 0E45 CDD409 LD A,71H
28 0E48 3E71 CALL #CLR8
29 0E4A CDD509 ENT
HOME: ENT
30 0E4D 210000 LD HL,0
31 0E50 1BAD JR CURS3
;
DEFS +8
;
CR
;
CR: ENT
40 0E5A CDF302 CALL RRCA
41 0E5D 0F JR L,0
42 0E5E 30B6 LD H
43 0E60 2E00 INC +24
44 0E62 24 CP Z,CR1
45 0E63 FE18 JR H
46 0E65 2B03 INC CURS1
47 0E67 24 JR (DSPXY),HL
48 0E68 1895 ENT
49 0E6A 227111 LD
50 0E6D ; SCROL
51 0E6D 01C003 SCROL: ENT
52 0E6D 1100D0 LD DE,SCRN
53 0E6D 212BD0 LD HL,SCRN+40
54 0E70 0E73 PUSH BC
55 0E73 C5

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 56 04.07.83
01 0E77 EDB0 LDIR
02 0E79 C1 POP BC
03 0E7A D5 DE
04 0E7B 1100D8 LD DE,SCRN+800H
05 0E7E 212BD8 LD HL,SCRN+82BH
06 0E81 EDB0 LDIR
07 0E83 062B LD B,40
08 0E85 EB DE,HL
09 0E86 3E71 LD A,71H
10 0E8B CDDDOF CALL ?DINT
11 0E8B E1 POP HL
12 0E8C 062B LD B,40
13 0E8E CDD80F ?CLR
14 0E91 011A00 BC,26
15 0E94 117311 LD DE,MANG
16 0E97 217411 LD HL,MANG+1
17 0E9A EDB0 LDIR
18 0E9C 3600 LD (HL),0
19 0E9E 3A7311 LD A,(MANG)
20 0EA1 B7 OR A
21 0EA2 2B41 JR Z,?RSTR
22 0EA4 217211 LD HL,DSPXY+1
23 0EA7 35 DEC (HL)
24 0EAB 1BC3 JR SCROL
;
CONTROL CODE TABLE
;
CTBL: ENT
25 0EAA 6D0E DEFN SCRL
26 0EAA F80D DEFN CURSD
27 0EAA 050E DEFN CURSU
28 0EAA 0D0E DEFN CURSR
29 0EAA 4D0E DEFN HOME
30 0EAA 3A0E DEFN CLRS
31 0EAA F80E DEFN DEL
32 0EAA E10E DEFN INST
33 0EAA E00E DEFN ALPHA
34 0EAA E00E DEFN KANA
35 0EAA E00E DEFN ?RSTR
36 0EAA E00E DEFN ?RSTR
37 0EAA E00E DEFN ?RSTR
38 0EAA E00E DEFN ?RSTR
39 0EAA E00E DEFN ?RSTR
40 0EAA E00E DEFN ?RSTR
41 0EAA E00E DEFN ?RSTR
42 0EAA E00E DEFN ?RSTR
43 0EAA E00E DEFN ?RSTR
44 0EAA E00E DEFN ?RSTR
45 0EAA E00E DEFN ?RSTR
46 0EAA E00E DEFN ?RSTR
47 0EAA E00E DEFN ?RSTR
48 0EAA E00E DEFN ?RSTR
49 0EAA E00E DEFN ?RSTR
50 0EAA E00E DEFN ?RSTR
51 0EAA E00E DEFN ?RSTR
52 0EAA E00E DEFN ?RSTR
53 0EAA E00E DEFN ?RSTR
54 0EAA E00E DEFN ?RSTR
55 0EAA E00E DEFN ?RSTR
56 0EAA E00E DEFN ?RSTR
57 0EAA E00E DEFN ?RSTR
58 0EAA E00E DEFN ?RSTR
59 0EAA E00E DEFN ?RSTR
60 0EAA E00E DEFN ?RSTR

INST: SET 3,H
INST2: LD A,(HL)
INC HL
LD (HL),A
DEC HL
RES 3,H
LD A,C
OR B
JR NZ,INST2
EX DE,HL

```

```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 57                                04.07.83
01 0ED9 3600 LD (HL),0
02 0EDB CBDC SET 3,H
03 0EDD 3671 LD (HL),71H
04 0EDF 1804 JR ?RSTR
;
05 0EE1
06 0EE1
07 0EE1
08 0EE1
09 0EE1 ;ORG 0EE1H:ALPHA
10 0EE1
11 0EE1 ALPHA: ENT
12 0EE1 XOR A
13 0EE2 ALPHA1: ENT
14 0EE2 327011 LD (KANAF),A
15 0EE5
16 0EE5
17 0EE5 ; RESTORE ;
18 0EE5
19 0EE5 ?RSTR: ENT
20 0EE5 E1 POP HL
21 0EE6 ?RSTR1: ENT
22 0EE6 D1 POP DE
23 0EE7 C1 POP BC
24 0EE8 F1 POP AF
25 0EE9 C9 RET
26 0EEA
27 0EEA ; MONITOR WORK AREA ;
28 0EEA
29 D000 P
30 E003 P
31 0EEA
32 0EEA
33 0EEA ; DEFS +4
34 0EEA ;ORG 0EEEH:KANA
35 0EEE
36 0EEE
37 0EEE KANA: ENT
38 0EEE CDD40D CALL GRSTAS
39 0EF1 CAB90D JP Z,DSP01
;
40 0EF4 3E01 LD A,+1
41 0EF6 1BEA JR ALPH1
42 0EF8
43 0EF8
44 0EF8 DEL: ENT
45 0EF8 EB EX DE,HL
46 0EF9 7C LD A,H
47 0EFA B5 OR L
48 0EFB 2BE8 JR Z,?RSTR
49 0EFD 7D LD A,L
50 0EFE B7 OR A
51 0EFF 200D JR NZ,DEL1
52 0F01 CDF302 CALL .MANG
53 0F04 3B0B JR C,DEL1
54 0F06 CDB10F CALL ?PONT
55 0F09 2B DEC HL
56 0F0A 3600 LD (HL),+0
57 0F0C 1B25 JR INST-5
58 0F0E ENT
59 0F0E CDF302 CALL RRCA
60 0F11 OF

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 58                                04.07.83
01 0F12 5E2B LD A,40
02 0F14 3001 JR NC,+3
03 0F16 07 RLCA
04 0F17 95 SUB L
05 0F18 47 LD B,A
06 0F19 CDB10F CALL ?PONT
07 0F1C 7E LD A,(HL)
08 0F1D 2B DEC HL
09 0F1E 77 LD (HL),A
10 0F1F 23 INC HL
11 0F20 CBDC SET 3,H
12 0F22 7E LD A,(HL)
13 0F23 2B DEC HL
14 0F24 77 LD (HL),A
15 0F25 CB9C RES 3,H
16 0F27 23 INC HL
17 0F28 23 INC HL
18 0F29 10F1 DJNZ DEL2
19 0F2B 2B DEC HL
20 0F2C 3600 LD (HL),0
21 0F2E CBDC SET 3,H
22 0F30 217100 LD HL,71H
23 0F33 3ED4 LD A,C4H
24 0F35 C3E00D JP ?DPT+4
25 0F3B
26 0F3B ; INST:
27 0F3B CDF302 CALL .MANG
28 0F3B 0F RRCA
29 0F3C 2E27 LD L,+39
30 0F3E 7D LD A,L
31 0F3F 3001 JR NC,+3
32 0F41 24 INC H
33 0F42 CDB40F CALL ?PNT1
34 0F45 E5 PUSH HL
35 0F46 2A7111 LD HL,(DSPXY)
36 0F49 3002 JR NC,+4
37 0F4B 3E4F LD A,+79
38 0F4D 95 SUB L
39 0F4E 0600 LD B,0
40 0F50 4F LD C,A
41 0F51 D1 POP DE
42 0F52 2B91 JR Z,?RSTR
43 0F54 1A LD A,(DE)
44 0F55 B7 OR A
45 0F56 20BD NZ,?RSTR
46 0F58 62 LD H,D
47 0F59 6B LD L,E
48 0F5A 2B LD HL
49 0F5B C3CA0E DEC INST2
50 0F5E
51 0F5E ;
52 0F5E ; PROGRAM SAVE
53 0F5E ;
54 0F5E ; CMD. 'S'
55 0F5E ;
56 0F5E ; SAVE:
57 0F5E CDD3D01 ENT
58 0F61 220411 CALL HEXIY
59 0F64 44 LD (DTADR),HL
60 0F65 4D LD B,H
; C,L
; START ADR. BUFFER
; DATA ADR. BUFFER

```

```

** Z80 ASSEMBLER SB-7201 <12-013A> PAGE 59 04.07.83
01 OF66 CDA602 CALL .4DE
02 OF69 CD3D01 CALL HEX1Y
03 OF6C ED42 SBC HL,BC
04 OF6E 23 INC HL
05 OF6F 220211 LD (SIZE),HL
06 OF72 CDA602 CALL .4DE
07 OF75 CD3D01 CALL HEX1Y
08 OF78 220611 LD (EXADR),HL
09 OF7B CD0900 CALL NL
10 OF7E 11B09 LD DE,MSGSV
11 OF81 DF RST 3
12 OF82 CD2F01 CALL BGTEL
13 OF85 CDA602 CALL .4DE
14 OF88 CDA602 CALL .4DE
15 OF8B 21F110 LD HL,NAME
16 OF8E 13 SAV1: ENT DE
17 OF8F 1A LD A,(DE)
18 OF90 77 LD (HL),A
19 OF91 23 INC HL
20 OF92 FE0D CP ODH
21 OF94 20F8 JR NZ,SAV1
22 OF96 3E01 LD A,01H
23 OF98 32F010 LD (ATTRB),A
24 OF9B CD3604 CALL ?MRI
25 OF9E DA0701 JP C,?ER
26 OFA1 CD7504 CALL ?MRD
27 OFA4 DA0701 JP C,?ER
28 OFA7 CD0900 CALL NL
29 OFAA 114209 LD DE,MSGOK
30 OFAD DF RST 3
31 OFAE C3AD00 JP ST1
32 OFB1 ;
33 OFB1 ;
34 OFB1 ;
35 OFB1 ;
36 OFB1 ;
37 OFB1 ;
38 OFB1 ;
39 OFB1 ;
40 OFB1 ;
41 OFB1 ;
42 OFB1 ;
43 OFB1 ;
44 OFB1 ;
45 OFB1 2A7111 LD HL,(DSPXY)
46 OFB4 ;
47 OFB4 ;
48 OFB4 ;
49 OFB4 ;
50 OFB4 F5 ENT AF
51 OFB5 C5 PUSH BC
52 OFB6 D5 PUSH DE
53 OFB7 E5 PUSH HL
54 OFB8 C1 POP BC
55 OFB9 112B00 LD DE,002BH
56 OFBC 21B0CF LD HL,SCRN-40
57 OFBF ENT
58 OFBF 19 ADD HL,DE
59 OFC0 05 DEC B
60 OFC1 F2BF0F JP P,-2

** Z80 ASSEMBLER SB-7201 <12-013A> PAGE 60 04.07.83
01 OFC4 0600 LD B,+0
02 OFC6 09 LD HL,BC
03 OFC7 D1 POP DE
04 OFC8 C1 POP BC
05 OFC9 F1 POP AF
06 OFCA C9 RET
07 OFCB ;
08 OFCB ;
09 OFCB ;
10 OFCB ;
11 OFCB ;
12 OFCB ;
13 OFCB CD8B05 VRFY: ENT CALL
14 OFCE DA0701 JP C,?ER
15 OFD1 114209 LD DE,MSGOK
16 OFD4 DF RST 3
17 OFD5 C3AD00 JP ST1
18 OFD8 ;
19 OFD8 ;
20 OFD8 ;
21 OFD8 ;
22 OFD8 ;
23 OFD8 ;
24 OFD8 ;
25 OFD8 ;
26 OFD8 ;
27 OFD8 ;
28 OFD8 ;
29 OFD8 AF XOR A
30 OFD9 1B02 JR +4
31 OFDB ?CLRFF: ENT LD A,FFH
32 OFDB 3EFF LD (HL),A
33 OFDD 77 LD HL
34 OFDE 23 INC HL
35 OFDE 23 DJNZ -2
36 OFDF 10FC RET
37 OFE1 C9 ;
38 OFE2 ;
39 OFE2 ;
40 OFE2 ;
41 OFE2 ;
42 OFE2 ;
43 OFE2 C5 GAPCK: ENT PUSH BC
44 OFE3 D5 PUSH DE
45 OFE4 E5 PUSH HL
46 OFE5 0101E0 LD BC,KEYPB
47 OFE8 1102E0 LD DE,CSTR
48 OFEB ;
49 OFEB 2664 LD H,100
50 OFED ;
51 OFED CD0106 LD EDGE
52 OFE0 3B0B JR C,GAPCK3
53 OFE2 CD4A0A CALL DLY3
54 OFE5 1A LD A,(DE)
55 OFE6 E620 AND 20H
56 OFE8 20F1 JR NZ,GAPCK1
57 OFFA 25 DEC H
58 OFFB 20F0 JR NZ,GAPCK2
59 OFFD ;
60 OFFD C39B06 GAPCK3: ENT JP RET3

; CALL DLY2*3

```

```

** Z80 ASSEMBLER SB-7201 <1Z-013A> PAGE 61          04.07.83
01 1000      SKP      H
** Z80 ASSEMBLER SB-7201 <1Z-013A> PAGE 62          04.07.83
01 1000      ;
02 1000      ;
03 1000      ; MONITOR WORK AREA ;
04 1000      ; (M2-700) ;
05 1000      ;
06 1000      ;
07 1000      ;
08 10F0      ORG 10F0H
09 10F0      SP: ENT
10 10F0      IBUFE: ENT
11 10F0      ATRB: ENT
12 10F0      DEFS +1
13 10F1      NAME: ENT
14 10F1      DEFS +17
15 1102      SIZE: ENT
16 1102      DEFS +2
17 1104      DTADR: ENT
18 1104      DEFS +2
19 1106      EXADR: ENT
20 1106      DEFS +2
21 1108      COMNT: ENT
22 1108      DEFS 104
23 1170      KANAF: ENT
24 1170      DEFS +1
25 1171      DSPXY: ENT
26 1171      DEFS +2
27 1173      MANG: ENT
28 1173      DEFS +27
29 118E      FLASH: ENT
30 118E      DEFS +1
31 118F      FLPST: ENT
32 118F      DEFS +2
33 1191      FLSST: ENT
34 1191      DEFS +1
35 1192      FLSDT: ENT
36 1192      DEFS +1
37 1193      STRGF: ENT
38 1193      DEFS +1
39 1194      DPRNT: ENT
40 1194      DEFS +1
41 1195      TMCNT: ENT
42 1195      DEFS +2
43 1197      SUMDT: ENT
44 1197      DEFS +2
45 1199      CSMDT: ENT
46 1199      DEFS +2
47 119B      AMPN: ENT
48 119B      DEFS +1
49 119C      TIMEG: ENT
50 119C      DEFS +1
51 119D      SWRK: ENT
52 119D      DEFS +1
53 119E      TEMPW: ENT
54 119E      DEFS +1
55 119F      ONTYO: ENT
56 119F      DEFS +1
57 11A0      OCTV: ENT
58 11A0      DEFS +1
59 11A1      RATIO: ENT
60 11A1      DEFS +2
; TAPE BUFFER(128B)
; ATTRIBUTE
; FILE NAME
; BYTE SIZE
; DATA ADR
; EXECUTION ADR
; COMMENT
; KANA FLAG
; DISPLAY CO-ORDINATES
; COLOUMN MANAGEMENT
; FLASHING DATA
; FLASHING POSITION
; FLASHING STATUS
; CURSOR DATA
; STRING FLAG
; TAB COUNTER
; TAPE MARK COUNTER
; CHECK SUM DATA
; FOR COMPARE SUM DATA
; AMPM DATA
; TIME FLAG
; KEY SOUND FLAG
; TEMPO WORK
; ONTYO WORK
; OCTAVE WORK
; ONPU RATIO

```

```

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 63                                04.07.83
; GET LINE BUFFER
01 11A3          BUFR:  ENT      DEFS      +81
02 11A3          ;
03 11F4          ;
04 11F4          ;
05 11F4          ;
06 11F4          ;
07 11F4          ;
08 E000          KEYPA: EQU      E000H
09 E001          KEYPB: EQU      E001H
10 E002          KEYPC: EQU      E002H
11 E003          KEYPD: EQU      E003H
12 E002          CSTR:  EQU      E002H
13 E003          CSTPT: EQU      E003H
14 E004          CONTO: EQU      E004H
15 E005          CONT1: EQU      E005H
16 E006          CONT2: EQU      E006H
17 E007          CONTF: EQU      E007H
18 E008          SUNDG: EQU      E008H
19 E008          TEMP:  EQU      E008H
20 11F4          ;
21 11F4          END

** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 64                                04.07.83
#BRK 08BB      #CLR08 09D4      #CLR8  09D5      $MCP  006B      ..LPT  017B
.4DE 02A6      .LPT  0176      .MANG  02F3      2HE1      0434      2HEX  041F
?KEY 09B3      ?ADCN 08B9      ?BEL  0577      ?BELD  0352      ?BLNK  0DA6
?BRK 0A32      ?BRK1 0A48      ?BRK2  09B0      ?BRK3  09B6      ?CLER  0FDB
?CLRRF 0AFB      ?DACH 08CE      ?DINT  0FDD      ?DPCF  0DDC      ?DSP  0DB5
?ER  0107      ?FLAS 09FF      ?FLS  09E3      ?GET  08BD      ?GETL  07E6
?KEY 08CA      ?KY1  0BD6      ?KY2  08DA      ?KY5  08FA      ?KY55  08FB
?KYGRP 08FE      ?KYGRS 0909      ?KYSM  08B3      ?LOAD  05F0      ?LNL  090E
?MLDY 01C7      ?MODE 073E      ?MSG  0893      ?MSGX  08A1      ?NL  091B
?PNT1 0FB4      ?PNT2 0FBF      ?PNT  0FB1      ?PRNT  0935      ?PRT  0946
?PRTS 0920      ?PRTT 0924      ?RDD  04FB      ?RDI  04DB      ?RSTR  0EE5
?RSTR1 0EE6      ?SAVE 0892      ?SWEP  0A50      ?TEMP  02E5      ?TMR1  0375
?TMR2 037F      ?TMRD 035B      ?TMS1  0331      ?TMS2  0344      ?TMS  0308
?VRFY 058B      ?WRD  0475      ?WRI  0436      ?ALPH1 0EE2      ?ALPHA 0EE1
AMPM  119B      ASC  03DA      ATBL  0A92      ATTR  10F0      AUTO3  07ED
BELL  003E      BGTEL 012F      BRKEY 001E      BUFR  071A      CKS1  0720
CKS2 072F      CKS3  0733      CKSUM  071A      CLEAR  09DB      CLEAR1 09DA
CLRS  0E3A      CHYO  005B      COMNT  110B      CONTO  0E04      CONT1  0E05
CONTF 0E06      CONTF 0E07      CR  0E5A      CR1  0E6A      CSMDT  1199
CSTPT 0E03      CSTR  0E02      CTBL  0EAA      CURS1  00FF      CURS2  0E16
CURS3 0E0F      CURS4  0E23      CURS5  0E02      CURSD  0DFB      CURSL  0E25
CURSR 0E0D      CURSU  0E05      CURSU1 0E0B      DACN1  0BE3      DACN2  0BDF
DACN3 0BE0      DEL  0E0E      DEL1  0F0E      DEL2  0F1C      DLY1  0759
DLY12 0996      DLY2  0760      DSPXY  1171      DSWEP  09A9      DFRNT  1194
DSF01 0DB9      DSFO4 0DD0      DXPY  1171      DSWEP  09A9      DTADR  1104
DUM1  0DB8      DUM2  0D3E      DUM3  0D37      DUMP  0D29      EDG1  0607
EDG2  0613      EDGE  0601      EXADR  1106      FD  00FF      FD1  0106
FD2  0102      FLAS1 097B      FLAS2  09F7      FLAS3  09F3      FLASH  11BE
FLKEY 057E      FLFST 11BF      FLSDT  1192      FLSST  1191      GAP  077A
GAP1  078E      GAP2  0796      GAP3  079C      GAPCK  0FE2      GAPCK1 0FEB
GAPCK2 0FED      GAPCK3 0FFD      GETKY  001B      GETL  0003      GETL1  07EA
GETLB  0818      GETL3  085B      GETL5  081D      GETLA  0865      GETLA  082B
GETL8  0863      GETLC  0822      GETLR  087E      GETLU  0876      GETLZ  086C
GOTO  00F3      GRSTAS 0DD4      HEX  03F9      HEX1Y  013D      HEXJ  03E5
HL1  041D      HLHEX  0410      HOME  0E4D      IBUFE  10F0      INST  0F38
INST2  0ECA      KANA  0EEE      KANAF  1170      KANST  0E03      KEYPA  0E00
KEYPB  0E01      KEYPC  0E02      KEYPF  0E03      KSL1  09B7      KSL2  09BC
KTBL  0BEA      KTBLC  0CAA      KTBLG  0CE9      KTBLGS  0C6A      KTBL5  0C2A
LETNL 0006      LLPT  0470      LDAO  0116      LOAD  0111      LONG  0A1A
LPRNT 01BF      M#TBL 0284      MANG  1173      MCR  07AB      MCR1  07AB
MCR2  07D4      MCR3  07D7      MELDY  0030      MLD1  01D1      MLD2  0205
MLD3  020D      MLD4  0211      MLD5  0214      MLD51  02C4      MLDSP  02BE
MLDST 02AB      MONIT 0000      MOT1  06A4      MOT2  06A8      MOT4  06B9
MOT5  06D8      MOT7  06B7      MOT8  06D0      MOT9  06D7      MOTOR  069F
MSG  0015      MSG#1  03FB      MSG#2  03FD      MSG#3  0402      MSG#7  0467
MSG1  0896      MSG#2  09A0      MSG#3  06E7      MSG#4  0942      MSG#K  0942
MSGSV 09BB      MSGX  001B      MSGX1  08A4      MSGX2  08A7      MST1  0705
MST2  070C      MST3  0717      MSTA  0044      MSTOP  0700      MSTP  0047
MTBL  026C      NAME  10F1      NL  0009      NLPHL  05FA      NOADD  03E2
OCTV  11A0      ONP1  021F      ONP2  022C      ONP3  0285      ONPU  021C
ONTYD  119F      OPTBL  029C      PEN  01BB      PLOT  0184      PMSG  01A5
PMSG1  01AB      PRNT  0012      PRNT2  0967      PRNT3  096C      PRNT4  096F
PMSG5 0959      PRNTS  000C      PRNTT  000F      PRTHL  03BA      PRTHX  03C3
PTEST 0155      PTRN  0180      PTST0  015A      PTST1  0170      RATIO  11A1
RBY1  0630      RBY2  0649      RBY3  0654      RBYTE  0624      RD1  04E6
RDA  01B6      RDATA  002A      RDINF  0027      RET1  04D2      RET2  0554
RET3  069B      RTAPE  050E      RTP1  0513      RTP2  0519      RTP3  0532
RTP4  0554      RTP5  0565      RTP6  0572      RTP7  056E      RTP8  0553
RTP9  0574      RYTHM  02CB      SAV1  0FBE      SAVE  0F5E      SCRN  0000
SCROL 0E6D      SG  00F7      SHORT  0A01      SIZE  1102      SLPT  03D5

```

```

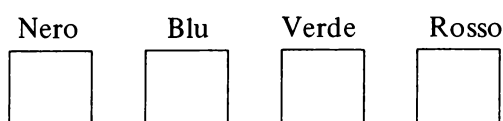
** Z80 ASSEMBLER SB-7201 <17-013A> PAGE 65 04.07.83
SP 10F0 SPHEX 03B1 SS 00A2 ST0 0070 ST1 00AD
ST2 00BB START 004A STRGF 1193 SUMDT 1197 SUNDG E008
SV0 0BA2 SV1 0BB5 SWEP0 0A66 SWEP01 0A64 SWEP2 0A7F
SWEP3 0A77 SWEP6 0A5F SWEP9 0A73 SWRK 119D TEMP E00B
TEMPW 119E TIMFG 119C TIMIN 03BD TIMRD 003B TIMST 0033
TM1 0675 TM2 067B TM3 068B TM4 069B TMRK 065B
TMCNT 1195 TVF1 05B2 TVF2 05B8 TVF3 05CC TVRFY 05AD
VERFY 002D VBOFF 0747 VRFY 0FCB VRNS 0BC5 WBY1 076D
WBYTE 0767 WRDAT 0024 WR11 0444 WR12 045E WR13 0464
WRINF 0021 WTAP1 0494 WTAP2 04A5 WTAP3 04D2 WTAPE 04BA
XTMP 0041

```

A.6 Codici di controllo della stampante-plotter a colori

A.6.1 Codici di controllo usati nel modo operativo testo

- Codice testo (\$01)
Imposta la stampante nel modo operativo testo.
- Codice grafico (\$02) Uguale all'istruzione BASIC MODE GR
Imposta la stampante nel modo operativo grafico.
- Allineamento (\$03) Uguale all'istruzione BASIC SKIP-1
Sposta la carta di una riga in direzione contraria. Il contatore di righe viene diminuito di 1.
- Prova penna (\$04) Uguale all'istruzione BASIC TEST
Scriva gli schemi seguenti per cominciare a far fluire l'inchiostro dalle penne, quindi imposta scala = 1 (40 caratteri/riga), colore = 0.
Scala = 1 (40 caratteri/riga), colore = 0



- Scala di riduzione (\$09) + (\$09) + (\$09)
Riduce la scala da 1 a 0 (80 caratteri/riga).
- Annullamento riduzione (\$09) + (\$09) + (\$0B)
Ingrandisce la scala da 0 a 1 (40 caratteri/riga).
- Impostazione contatore righe (\$09) + (\$09) + (ASCII)₂ + (ASCII)₁ + (ASCII)₀ + \$0D)
..... Uguale all'istruzione BASIC PAGE.
Specifica il numero di righe per pagina come indicato da 3 byte del codice ASCII. Il numero massimo di righe per pagina è 255. Impostazione a 66 quando l'alimentazione viene attivata o quando il sistema viene ripristinato.
- Avanzamento righe (\$0A) Uguale all'istruzione BASIC SKIP 1
Sposta la carta di una riga in direzione di avanzamento. Il contatore di righe viene incrementato di 1.
- Ingrandimento scala (\$0B)
Ingrandisce la scala da 1 a 2 (26 caratteri/riga)
- Annullamento ingrandimento (\$0c)
Riduce la scala da 2 a 1.
- Ritorno a capo (\$0D)
Sposta il carrello alla sinistra dell'area di stampa.
- Ritorno unitario (\$0E)
Sposta il carrello a sinistra di un colonna. Questo codice viene ignorato quando il carrello si trova alla sinistra dell'area di stampa.
- Trascinamento moduli (\$0F)
Sposta la carta all'inizio della pagina successiva e ripristina il contatore di righe a 0.
- Prossimo colore (\$1D)
Fa diventare la penna del colore successivo.

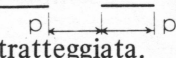
A.6.2 Scala caratteri

- La scala caratteri viene automaticamente impostata ad 1 (40 caratteri/riga) quando la macchina viene accesa. In seguito, può essere modificata mediante i codici ed i comandi di controllo.
- Nel modo operativo grafico, la scala può essere cambiata nella gamma da 0 a 63.
- La scala viene impostata ad 1 quando il modo operativo viene commutato da grafico a testo.

A.6.3 Comandi del modo operativo grafico

A. 6. 3. 1 Tipo comandi

Nel modo operativo grafico, la stampante può essere controllata emettendo i comandi seguenti alla stampante. Le parole tra parentesi sono le istruzioni BASIC che hanno le stesse funzioni dei comandi nel modo operativo grafico.

Nome comando	Formato	Funzione
LINE TYPE	Lp (p = 0 ~ 15)	Specifica il tipo di riga (intera o tratteggiata) ed il passo dei punti.  p = 0: linea intera, p = 1 ~ 15: linea tratteggiata.
ALL INITIALIZE	A	Imposta la stampante nel modo operativo testo.
HOME (PHOME)	H	Alza la penna e la riporta all'origine (posizione di sede).
INITIALIZE (HSET)	I	Imposta la posizione attuale della penna come origine (x = 0, y = 0)
DRAW (LINE)	Dx, y, . . . xn, yn (-999 ≤ x, y ≤ 999)	Traccia le linee dalla posizione attuale della penna alle coordinate (x ₁ , y ₁), quindi alle coordinate (x ₂ , y ₂) e così via.
RELATIVE DRAW (RLINE)	JΔx, Δy . . . Δxn, Δyn (-999 ≤ Δx, Δy ≤ 999)	Traccia le linee dalla posizione attuale della penna alle coordinate relative (Δx ₁ , Δy ₁), quindi alle coordinate relative Δx ₂ , Δy ₂) e così via.
MOVE (MOVE)	Mx, y (-999 ≤ x, y ≤ 999)	Alza la penna e la sposta alle coordinate (x, y).
RELATIVE MOVE (RMOVE)	RΔx, Δy (-999 ≤ Δx, Δy ≤ 999)	Alza la penna e la sposta alle coordinate relative (Δx, Δy).
COLOR CHANGE (PCOLOR)	Cn (n = 0 ~ 3)	Cambia il colore della penna ad n.
SCALE SET	Sn (n = 0 ~ 63)	Specifica la scala dei caratteri.
ALPHA ROTATE	Qn (n = 0 ~ 3)	Specifica la direzione in cui vengono stampati i caratteri.
PRINT	Pc ₁ c ₂ c ₃ . . . cn (n = ∞)	Stampa i caratteri.
AXIS (AXIS)	Xp, q, r (p = 0 ~ 1) (q = -999 to 999) (r = 1 ~ 255)	Traccia l'asse delle X quando p = 1 e l'asse delle Y quando p = 0. specifica il passo della scala ed r specifica il numero di segni sulla scala da tracciare.

A. 6. 3. 2 Formato dei comandi

Sono previsti 5 tipi di formati dei comandi come mostrato qui di seguito.

1. Solo carattere di comando (senza parametri)
"A", "H", "I"
2. Carattere di comando più un parametro
"L", "C", "S", "Q"
3. Carattere di comando più coppie di parametri
"D", "J", "M", "R"
", " viene usata per separare i parametri ed il codice CR viene usato per porre termine alla lista dei parametri.
4. Comando più stringa di caratteri
"P"
La stringa di caratteri viene terminata con un codice CR.
5. Comando più tre parametri
"X"
", " viene usata per separare i parametri.

A.6.3.3 Specifica dei parametri

1. Gli spazi iniziali vengono ignorati.
2. Qualsiasi numero preceduto da " - " viene trattato come un numero negativo.
3. Se il numero delle posizioni di un numero è superiore a 3, sono valide solo le tre posizioni inferiori.
4. Ogni parametro viene terminato con " , " o un codice CR. Se in un parametro sono inclusi dei caratteri diversi dai numeri, i caratteri successivi vengono ignorati sino a che non viene rilevata una virgola o un codice CR.

Esempio D _ _ - 1 3 5 . 2 1 ,
 ↑ ↑
 Ignorato

A. 6. 3. 4 Formato abbreviato

1. Qualsiasi comando può essere seguito da un comando di un solo carattere senza immettere un codice CR.
Esempio: " HD100, 200 " CR è valido ed è uguale a " H " CR ' D100, 200 " CR.
2. Qualsiasi comando può essere seguito da un comando con un solo parametro separandoli con una virgola " , ".
Esempio: " L0, S1, Q0, C1, D100, 200 " CR è valido.
3. Un comando con coppie di parametri deve essere terminato con un codice CR.

A. 6. 3. 5 Cambiamento dati dovuto alla commutazione di modo operativo

I dati seguenti cambiano quando la stampante viene commutata dal modo operativo grafico al modo operativo testo.

- Coordinate X ed Y
Y viene impostata a 0 e l'origine viene posta alla sinistra dell'area di stampa.
- Direzione dei caratteri
Q viene impostato a 0.
- Scala caratteri
La scala caratteri viene impostata ad 1.
- L'impostazione del tipo di riga non viene toccata.

A.7 Note relative al funzionamento

■ Registratore dati

- Sebbene il registratore dati dell'MZ-700 sia altamente affidabile, la testina di lettura/scrittura si usura dopo un uso prolungato. Inoltre le particelle magnetiche e la polvere si accumulano sulla testina degradando le prestazioni di lettura/scrittura. Pertanto, la testina deve essere pulita periodicamente o sostituita quando è usurata.

1. Per pulire la testina, aprite il vano della cassetta, premete il tasto **PLAY** e spazzolate la testina ed il cilindretto di presa usando un cotton fioc. Se sono molto sporchi, imbevete il cotton fioc di alcol.
2. Quando la testina è usurata, rivolgetevi al vostro rivenditore. Non tentate di sostituirla da soli.

■ Nastro a cassetta

- Con l'MZ-700 è possibile usare qualsiasi nastro a cassetta commercialmente disponibile. Tuttavia si consiglia di usare dei nastri a cassetta di qualità prodotti da fabbricanti affidabili.

- Usate nastri di tipo normale.
- Evitate di usare nastri a cassetta di tipo C-120.
- Si consiglia l'uso di nastri a cassetta C-60 o più brevi.
- Accertatevi di eliminare eventuali laschi del nastro servendovi di una matita o simili come mostrato a destra, prima di caricare il nastro a cassetta; diversamente il nastro può spezzarsi o avvolgersi attorno al cilindretto di presa.

- **Protezione dei programmi/dati dalla cancellazione accidentale**

Il registratore dati dell'MZ-700 è dotato di una funzione di protezione scrittura che funziona nello stesso modo dei normali registratori a cassette.

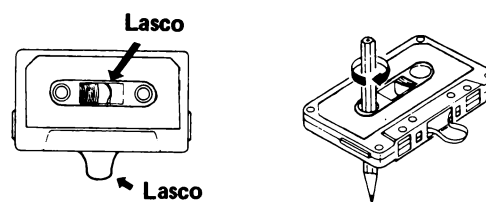
Per impedire che i dati vengano cancellati accidentalmente, togliete la linguetta di bloccaggio registrazione dalla cassetta con un cacciavite o simile. Questo rende impossibile premere il tasto **RECORD** impedendo la cancellazione di dati di valore.

■ Altro

- Vedere pagina 109 per i registratori a cassette commercialmente disponibili.

■ Unità video

Quando si usa un'unità video diversa da quella specificata per l'MZ-700, lo schermo deve essere regolato. Vedere pagina 106.



Togliete la linguetta di bloccaggio registrazione con un cacciavite



■ Stampante-plotter a colori

- Non ruotate il tamburo delle penne in direzione inversa quando sostituite le penne.
- Accertatevi di togliere le penne dal tamburo, copritele con i cappucci e conservatele in una scatola per impedire che si secchino quando la stampante non viene usata per un periodo di tempo prolungato.
- Ci vuole un po' di tempo perché l'inchiostro sulla carta asciughi. (L'inchiostro è idrosolubile).
- Non strappate la carta quando il coperchio della stampante è stato tolto. Tenete abbassato il portacarta quando strappate il foglio.
- Non toccate il meccanismo interno quando sostituite le penne. In caso contrario è possibile danneggiare la stampante.
- La stampante-plotter a colori genera per un momento un suono quando viene accesa. Questo non è un problema.
- Le lettere stampate nel modo 80 caratteri/riga possono essere difficili da leggere. In questo caso, usate il modo 40 caratteri/riga.
- Nel modo grafico, le righe stampate ripetutamente possono diventare confuse. Questo è più probabile che avvenga quando viene stampata ripetutamente una linea tratteggiata, a causa delle caratteristiche della penna a sfera, questo è inevitabile.

■ Note sul software

- Ci vogliono circa 3 minuti per caricare il programma interprete BASIC.
- Si deve usare l'interruttore di ripristino sul pannello posteriore nei casi seguenti. (Vedere 3. 1. 1).
Per arrestare l'esecuzione di un programma BASIC durante l'esecuzione normale o quando il programma entra in un'iterazione infinita.
Per tornare al programma, usate il comando #. Tuttavia, il programma o l'hardware devono essere controllati se il programma entra in iterazione.

■ Errore di calcolo BASIC

- Il BASIC converte i valori decimali in valori binari a virgola mobile prima di eseguire i calcoli, quindi converte i risultati dei calcoli binari in numeri decimali per la visualizzazione. Questo può portare ad una certa quantità di errore.

(Esempio:)

```
PRINT 817.3-810.4  
6.899999          . . . . . Il risultato corretto è 6.9.
```

- Vengono fatte delle approssimazioni durante il calcolo delle funzioni e delle potenze.
- Si deve tener conto di quanto sopra quando si usano le istruzioni IF.

(Esempio:)

```
10 A=1/100*100  
20 IF A=1 THEN PRINT "TRUE" :GOTO 40  
30 PRINT "FALSE"  
40 PRINT "A=" ; A  
50 END  
RUN  
FALSE  
A=1
```

Sebbene il risultato pratico dell'equazione della riga 10 sia 1, questo programma stampa FALSE a causa dell'errore dovuto alla conversione.

■ Note sulla manipolazione

● Interruttore di alimentazione

L'interruttore di alimentazione non deve essere toccato per almeno 10 secondi dopo che è stato attivato o disattivato. Questo è necessario per garantire il funzionamento corretto del calcolatore. Non staccate dalla presa il cavo di alimentazione quando l'interruttore di alimentazione è attivato; diversamente possono verificarsi degli inconvenienti.

● Cavo di alimentazione

Evitate di mettere degli oggetti pesanti come le scrivanie sopra al cavo di alimentazione. Questo può danneggiare il cavo di alimentazione risultando possibilmente in un incidente grave. Accertatevi di afferrare il cavo per la spina quando lo staccate dalla presa.

● Tensione di alimentazione

La tensione di alimentazione è di 240/220 VCA. Il calcolatore non può funzionare correttamente se la tensione è troppo alta o troppo bassa. Rivolgetevi al vostro rivenditore nel caso si presenti questo problema.

● Ventilazione

Sono previste molte aperture di sfiato nella cassa per prevenire il surriscaldamento. Mettete il calcolatore in un posto ben ventilato e non copritelo con un panno. Non mettete alcun oggetto alla sinistra del calcolatore, perché questo è il lato in cui si trovano le aperture di sfiato per l'alimentatore.

● Umidità e polvere

Non usate il calcolatore in luoghi umidi o polverosi.

● Temperatura

Non mettete il calcolatore vicino a termosifoni o in luoghi in cui può essere esposto alla luce solare diretta; se non viene rispettata questa precauzione, i componenti del calcolatore possono essere danneggiati.

● Acqua e materiali estranei

L'acqua ed altri materiali estranei (per esempio spilli) che entrano nel calcolatore lo danneggiano. Staccate dalla presa immediatamente il cavo di alimentazione e contattate il rivenditore nel caso si verifichi uno di questi incidenti.

● Urti violenti

Evitate di sottoporre il calcolatore ad urti violenti; questi danneggiano il calcolatore permanentemente.

● Guasti

Arrestate immediatamente il funzionamento e contattate il rivenditore se notate delle anomalie.

● Non uso prolungato

Accertatevi di staccare dalla presa il cavo di alimentazione se il calcolatore non viene usato per un periodo di tempo prolungato.

- **Collegamento di unità periferiche**

Usate soltanto parti e componenti della Sharp quando collegate eventuali unità periferiche; diversamente il calcolatore può essere danneggiato.

- **Sporcizia**

Sfregate la cassa con un panno morbido imbevuto d'acqua o di detergente quando si sporca. Per evitare lo scolorimento della cassa non usate fluidi volatili quali benzene.

- **Rumore**

Si consiglia di usare un filtro di linea quando il calcolatore viene usato in un luogo in cui possono essere presenti segnali di rumore di alto livello nell'alimentazione CA. (Il vostro rivenditore Sharp vi può fornire un filtro di linea). Allontanate il più possibile i cavi dei segnali dal cavo di alimentazione e dagli altri apparecchi elettrici.

- **Interferenza RF**

Può verificarsi un'interferenza con la ricezione televisori o radio a causa del segnale RF generato dal calcolatore se questo viene usato vicino ad un televisore o ad una radio.

I televisori generano un forte campo magnetico che può risultare in un funzionamento non corretto del calcolatore. Se questo avviene, spostate il televisore di almeno 2-3 metri dal calcolatore.

Copia/Correzione Dell'Interprete Basic Per L'MZ-700

A. Seguite la procedura seguente per copiare il nastro contenente l'interprete BASIC.

- 1) Accendete l'MZ-700 (→ controllo al monitor)
- 2) Modificate il contenuto della memoria con il comando "M" del monitor nel seguente modo:

```
*MCF00
CF00  FF → CD
CF01  00 → 27
CF02  FF → 00
CF03  00 → 38
CF04  FF → 03
CF05  00 → CD
CF06  FF → 2A
CF07  00 → 00
CF08  FF → DA
CF09  00 → FE
CF0A  FF → 00
CF0B  00 → C3
CF0C  FF → AD
CF0D  00 → 00
CF0E  FF → CD
CF0F  00 → 27
CF10  FF → 00
CF11  00 → 38
CF12  FF → F5
CF13  00 → C3
CF14  FF → CB
CF15  00 → 0F
```

Terminate premendo contemporaneamente i tasti SHIFT e BREAK.

N.B.: Non sempre il contenuto della memoria da CF00 a CF15 è come mostrato sopra.

- 3) Inserite la cassetta che deve essere letta (dalla quale fare la copia) nel registratore.
- 4) Date il comando "J" (di salto) nel seguente modo:

```
* JCF00 CR
⌞ PLAY
```

N.B.: Se un tasto del registratore è già premuto la scritta "PLAY" non appare.

- 5) Confermate il comando "⌞PLAY" sopra citato premendo il relativo tasto ed in questo modo il contenuto del nastro viene caricato. In questo caso non viene fornita alcuna informazione (tipo nome del file o altro).
Quando si verifica un errore ripartite dal punto 1).
- 6) Inserite nel registratore la cassetta su cui desiderate registrare il BASIC e riavvolgete il nastro.

7) Date il seguente comando:

* J1108

8) Vengono, ora, visualizzate le seguenti informazioni:

S-BASICEX SAVER xx ☐ xx ☒
HIT ANY KEY?

9) Premete un tasto qualsiasi:

⌞ Record Play

Il tasto avrebbe dovuto essere stato premuto precedentemente.

10) Premete i tasti e . La copia inizia e le seguenti indicazioni appaiono:

WRITING S-BASIC

Nel caso dell'MZ-711 il punto 9) deve essere effettuato dopo che il registratore esterno è in condizione di operare.

11) Un suono segnala la fine della copia.

12) Il ritorno al monitor avviene premendo il tasto di sulla parte posteriore del computer.

13) Riavvolgete il nastro e premete il tasto .

14) Date il seguente comando:

* JCF0E

⌞ PLAY

15) Premete il tasto del registratore e la funzione di verifica ha inizio. Se la verifica è positiva appare l'indicazione "OK!" senza altre informazioni. Quando si verifica un errore, ripartite dal punto 4).

16) Al fine di proteggere dalla scrittura la nuova cassetta rimuove te l'apposito pezzeto di plastica.

B. Per modificare l'interprete BASIC è necessaria la seguente procedura.

a) Operate come per la copia dal punto 1) al punto 5).

b) Con il comando "M" richiamate l'indirizzo da modificare.

ES. 8A in 1234H modificato in 7A.

		Premete
*M		1234
1234	8A	7A <input type="text" value="CR"/>
1235	8A	<input type="text" value="SHIFT"/> + <input type="text" value="BREAK"/>
*		

C. Riprendete l'operazione di copia dal punto 6) in avanti come descritto precedentemente.

SHARP CORPORATION
OSAKA, JAPAN

Printed in Japan
Gedruckt in Japan
Imprime' au Japon
Stampato in Giappone

3J 0.502-I
TINSI1092ACZZ

IN-700 MANUALE DELL'UTENTE SHARP