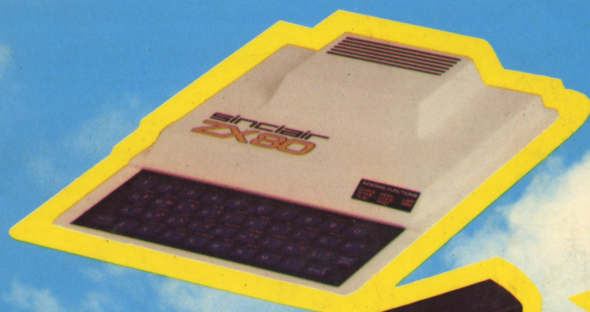


66 PROGRAMMI PER ZX81 E ZX80 CON NUOVA ROM + HARDWARE



GRUPPO
EDITORIALE
JACKSON

GAETANO
MARANO

66 PROGRAMMI
PER ZX81
E ZX80 CON NUOVA ROM
+ HARDWARE

di

Gaetano Marano



Gruppo
Editoriale
Jackson
Via Rosellini, 12
20124 Milano

° Copyright per l'edizione originale Gruppo Editoriale Jackson 1982

Il Gruppo Editoriale Jackson ringrazia per il prezioso lavoro svolto nella stesura dell'edizione italiana la signora Francesca di Fiore e l'ing. Roberto Pancaldi.

Tutti i diritti sono riservati. Stampato in Italia. Nessuna parte di questo libro può essere riprodotta, memorizzata in sistemi di archivio, o trasmessa in qualsiasi forma o mezzo, elettronico, meccanico, fotocopia, registrazione o altri senza la preventiva autorizzazione scritta dell'editore.

Stampato in Italia da:
S.p.A. Alberto Matarelli - Milano - Stabilimento Grafico

SOMMARIO

| | |
|---|------------|
| PREFAZIONE | VII |
| NOTE IMPORTANTI SUI PROGRAMMI E SULLA RAM UTILIZZATA | 1 |
| AVVISATORE ACUSTICO PER TASTIERA ED ALIMENTATORE TAMPONE PER ZX80 E ZX81 | 3 |
| Avvisatore Acustico | 3 |
| Alimentatore tampone | 5 |
| GRAFICA E DISEGNI | 9 |
| 1 Lavagna grafica | 9 |
| 2 Lavagna alfanumerica e grafica | 12 |
| 3 Roditore | 14 |
| 4 Roditore -2- con PLOT-UNPLOT | 16 |
| 5 Programma grafico | 16 |
| 6 Programma universale per PRINT AT | 18 |
| 7 Programma universale per PLOTTAGGIO | 20 |
| 8 Plottaggio QUADRATI | 21 |
| 9 Plottaggio CERCHI | 23 |
| 10 Plottaggio ELLISSI | 25 |
| Disegni | 28 |
| 11 Cielo stellato | 28 |
| 12 NEW YORK | 28 |
| 13 Montagne | 31 |
| PROGRAMMI VARI | 33 |
| 14 Progettazione di Multivibratori Monostabili con 555 | 33 |
| 15 Contapezzi | 35 |
| 16 Scritte rotanti | 40 |
| 17 Renumber | 43 |
| 18 Lettore Memoria | 46 |
| 19 Programma caricamento routines in linguaggio macchina contenute in stringa | 47 |
| 20 Programma caricamento routines in linguaggio macchina contenute in 1 REM | 49 |
| Conversioni numeriche | 50 |
| 21 Conversione da binario a decimale | 51 |
| 22 Conversione da binario a esadecimale | 51 |
| 23 Conversione da esadecimale a decimale | 52 |

| | |
|--|------------|
| 24 Conversione da esadecimale a binario | 53 |
| 25 Conversione da decimale a esadecimale | 54 |
| 26 Conversione da decimale a binario | 55 |
| 27 Programma di riunione delle routines di conversione numerica | 56 |
| ANIMAZIONI | 61 |
| 28 Countdown e lancio missile | 61 |
| 29 Caleidoscopio | 63 |
| 30 Rombospirale | 68 |
| 31 Animazione esplosione (con sonoro) | 70 |
| 32 Programma universale animazione con PRINT AT | 71 |
| 33 Programma universale animazione con PLOT UNPLOT | 74 |
| PROGRAMMI PER INTERFACCIA | 77 |
| Circuito di interfaccia per ZX80 e ZX81 | 77 |
| Programmi | 84 |
| 34 Programma universale per circuito di Interfaccia | 84 |
| 35 Luci rotanti | 88 |
| 36 Luci casuali | 89 |
| 37 Luci a riempimento e svuotamento | 90 |
| 38 Punto luminoso mobile | 91 |
| 39 Tasto automatico per telegrafia | 91 |
| MUSICA | 95 |
| Scheda musicale | 95 |
| Programmi | 101 |
| 40 Organo a 4 ottave | 101 |
| 41 Metronomo | 102 |
| 42 Scale musicali | 103 |
| 43 Generatore di musica casuale - 50 note | 104 |
| EFFETTI SONORI | 107 |
| 44 Sirena a 2 toni | 107 |
| 45 Sirena Americana | 107 |
| 46 Sirena multitono | 108 |
| 47 Mitra | 109 |
| 48 Campanella arrivo treni in stazione | 109 |
| 49 Segnale telefonico di linea occupata | 109 |
| 50 Segnale telefonico di linea libera | 110 |
| 51 Squillo del telefono | 110 |
| 52 Orologio | 110 |
| 53 Grillo | 111 |

| | |
|--------------------------------------|------------|
| 54 Segnale orario | 111 |
| 55 Sveglia | 112 |
| 56 Din Don | 112 |
| 57 Generatore di suoni casuali | 112 |
| GIOCHI | 115 |
| 58 Miniroulette (con sonoro) | 115 |
| 59 Campo minato (con sonoro) | 117 |
| 60 1 — 40 Dadi | 121 |
| 61 Tavola di numeri e lettere | 123 |
| 62 Tombola | 125 |
| 63 Metal Detector (con sonoro) | 127 |
| 64 Segnatempo (con sonoro) | 129 |
| 65 Lotto | 130 |
| 66 Totocalcio | 133 |

PREFAZIONE

Questo libro contiene 66 programmi di vario genere, scritti per girare sul computer Sinclair ZX81, computer che, per le sue eccezionali qualità unite ad un modestissimo prezzo, ha già conquistato centinaia di migliaia di persone in tutto il mondo riuscendo a superare già, secondo gli ultimi dati forniti dalla Sinclair, in numero di pezzi, le vendite di qualsiasi altro modello di computer, singolarmente preso, costruito sino ad oggi.

LO ZX81 viene utilizzato anche troppo per realizzare dei giochi. Ho pensato, perciò, di offrire in questo libro una maggiore varietà di programmi cercando di sfruttare nella gran parte di essi soprattutto le capacità grafiche dello ZX81 e, comunque, senza dimenticare del tutto i giochi che sono, naturalmente, presenti nella parte finale di questo libro con 9 programmi.

Oltre alle normali possibilità dello ZX81, il lettore troverà alcuni circuiti molto semplici e poco costosi che consentono di aggiungere nuove possibilità allo ZX81. Oltre ad un utilissimo avvisatore acustico per la tastiera (realizzabile con meno di 3.000 lire) e ad un alimentatore con accumulatore (per conservare i dati della memoria e usare il computer anche durante le interruzioni della corrente di rete), tali circuiti sono costituiti soprattutto da un circuito di INTERFACCIA che costa in componenti meno di 5000 lire e che permette di collegare lo ZX81 a qualsiasi dispositivo esterno e da un circuito MUSICALE realizzabile con circa 10000 lire e che fornisce tramite semplici istruzioni in BASIC, fino a 50 note su 4 ottave.

Di questi due ultimi circuiti, le cui applicazioni sono infinite, non vengono forniti solo gli schemi e le informazioni per costruirli, ma anche numerosi programmi pratici.

In particolare la scheda musicale è utilizzata per vari effetti sonori, per realizzare un organo a 4 ottave che usa i tasti dello ZX81 e per dare il sonoro a vari programmi tra cui naturalmente alcuni dei giochi che diventano così probabilmente gli unici giochi con effetti sonori disponibili a tutt'oggi per i computers Sinclair.

Un'altra novità di questo libro è l'uso in alcuni programmi (come per esempio l'organo elettronico) di speciali tastiere disegnate da sovrapporre alla tastiera sensitiva degli ZX per assegnare ai tasti alcune funzioni particolari.

Come indicato dal titolo, anche chi ha il vecchio ZX80 con la nuova ROM, può utilizzare molti dei programmi di questo libro compresi i programmi che usano i circuiti di interfaccia e musicale, circuiti che naturalmente possono essere collegati anche allo ZX80.

Gaetano Marano

NOTE IMPORTANTI SUI PROGRAMMI E SULLA RAM UTILIZZATA

Tutti i programmi possono girare sullo ZX81 ed una gran parte anche sullo ZX80 con la nuova ROM.

A fianco del titolo di ciascun programma viene indicato se il programma stesso funziona in FAST e/o SLOW. Se c'è scritto FAST il programma può girare su ZX80 e su ZX81, naturalmente in modo FAST. Se c'è scritto SLOW il programma può girare solo sullo ZX81. Se, infine, c'è scritto FAST o SLOW il programma può girare sia su ZX81 che su ZX80, però nel caso dello ZX80 si hanno alcune limitazioni, come per esempio in un disegno, l'impossibilità di vederlo formarsi, in sostanza, quindi, si ha la normale elaborazione ma non l'animazione grafica.

Per quanto riguarda il fabbisogno di memoria, solo per alcuni programmi, come per esempio gli effetti sonori, è sufficiente il singolo KBYTE e, comunque, nessuno richiede più di 4 KBYTES, per cui possono andare bene sia gli ZX espansi con la cartuccia RAM da 16K che con la cartuccia RAM da 3K. Non solo, ma in alcuni

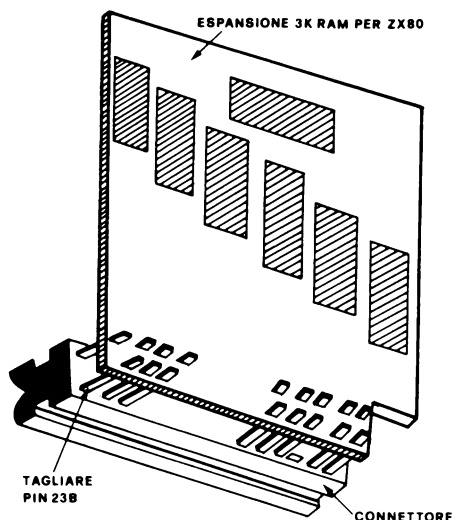


Figura 1

programmi che hanno delle routines in linguaggio macchina, le routines stesse sono state inserite entro i primi 4K di RAM in modo che tali programmi possano essere usati, senza modifiche, sia da chi dispone di RAM da 4K che da chi dispone di RAM da 16K.

Infine, se avete solo lo ZX81, ma non l'espansione RAM da 16K e volete usare la vecchia espansione da 3K, questo è possibile ma occorre effettuare sulla cartuccia da 3K una piccola modifica interrompendo e piegando in modo che non faccia contatto, il collegamento tra il connettore ed il circuito stampato, corrispondente a Pin 23B del connettore stesso, come visibile in fig. 1.

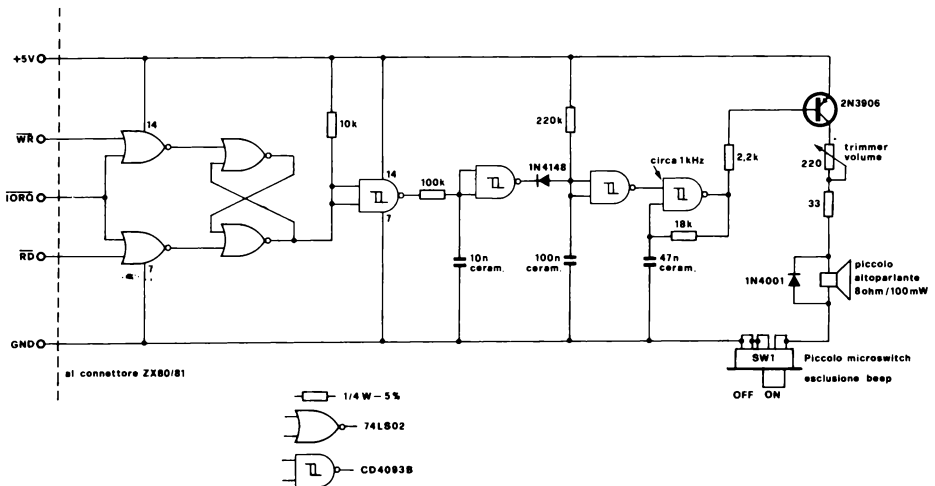
AVVISATORE ACUSTICO PER TASTIERA ED ALIMENTATORE TAMPONE PER ZX80 E ZX81

Avvisatore Acustico

L'avvisatore acustico per la tastiera sensibile degli ZX80/81 produce un breve suono ogni volta che viene premuto uno dei tasti.

L'utilità è evidente poiché questo semplice circuito (che può essere realizzato con 3-4.000 lire) permette di inserire programmi anche molto lunghi in un tempo molto più breve e senza dovere guardare il video ogni volta che si preme un tasto per verificare l'avvenuto inserimento della lettera, del simbolo o della Keyword richiesta.

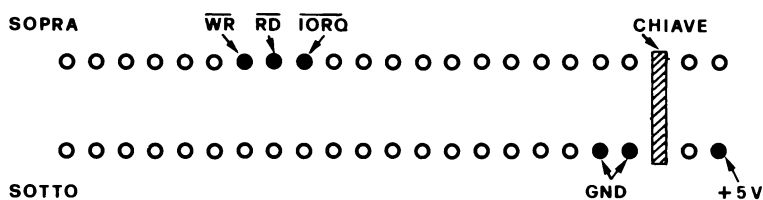
Il circuito è in fig. 2 ed utilizza come semiconduttori due integrati ed un transistor.



Tutti i componenti possono essere montati su un pezzettino di circuito stampato forato facendo, naturalmente, attenzione a collegare esattamente i vari componenti, tra questi soprattutto i due diodi che occorre verificare che siano collegati come dallo schema, prima di dare corrente.

Il circuito dell'avvisatore acustico va collegato tramite 5 fili (come visibile sempre in fig. 2) o all'interno del computer o, preferibilmente, al connettore dell'espansione di memoria tramite 5 spezzi di filo sottile ad un capo di diverso colore, come per esempio il filo per WIRE-WRAP.

In fig. 3 è visibile il connettore della RAM, per ZX80/81, visto da dietro e con indicati i 5 Pin ai quali vanno saldati i 5 fili dell'avvisatore acustico (+5V, GND, WR, RD, IORQ).



Connettore RAM 3K/16K per ZX80/81 visto da dietro

Figura 3

Nel collegare i fili al connettore fare molta ATTENZIONE a saldare i fili stessi ai Pin indicati e non ad altri e, soprattutto, controllare di non avere cortocircuitato durante la saldatura qualcuno dei Pin adiacenti, SPECIALMENTE i Pin che portano i +5V e i +9V.

TENETE PRESENTE CHE UN'ERRATA COSTRUZIONE DEL CIRCUITO O UNA ERRATA SALDATURA DEI FILI AL CONNETTORE PUO' CAUSARE DANNI AL COMPUTER.

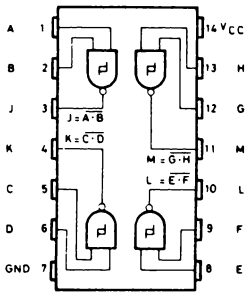
Per cui se vi occupate di programmazione ma non di circuiti integrati, è preferibile che facciate realizzare o controllare il tutto da un tecnico che abbia già esperienza di montaggi e saldature.

In fig. 4 vi sono le configurazioni degli integrati e del transistor usati.

Una cosa molto importante da dire prima di terminare questa parte è che l'avvisatore acustico genera un suono alla pressione dei tasti dello ZX81 solo quando lo ZX81 funziona nel modo FAST; questa non è una limitazione perchè, per inserire velocemente dei programmi sullo ZX81 si deve, comunque operare in FAST, poichè come avete avuto modo di verificare l'inserimento di nuove linee con il computer in SLOW è già di per sè molto lento.

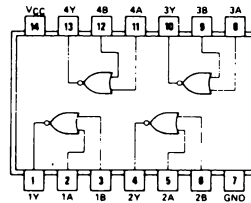
L'avvisatore acustico dispone anche di un piccolo interruttore che serve per disinserire il suono quando non è necessario, come per esempio quando è in funzione anche la scheda musicale che verrà esposta più avanti.

COS/MOS QUAD 2-INPUT NAND SCHMITT TRIGGERS



CD4093B

QUADRUPLE 2-INPUT POSITIVE-NOR GATES



74LS02

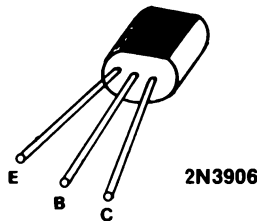


Figura 4

Alimentatore tampone

Questo alimentatore oltre alla funzione secondaria di fornire una tensione maggiormente stabilizzata, ha lo scopo principale di preservare i dati scritti nella RAM durante le interruzioni della tensione di rete e questo per un tempo più che sufficiente a conservare il programma (che a volte può avere richiesto molto tempo per essere scritto) su cassetta e, nella maggior parte dei casi, anche più che sufficiente ad attendere il ritorno della luce.

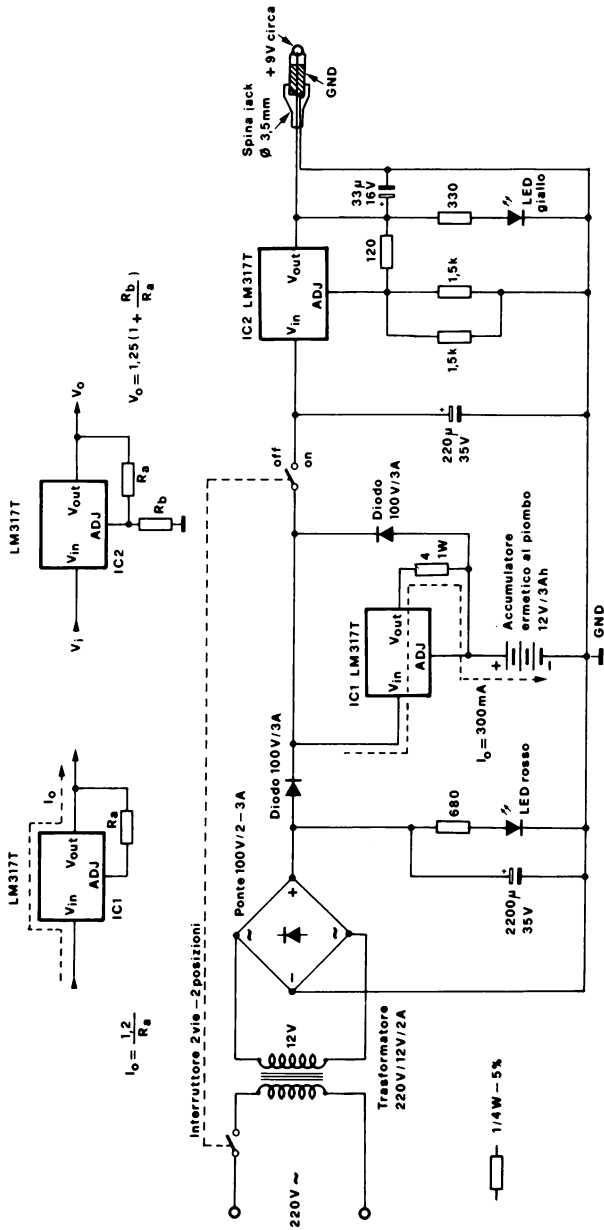


Figura 5

L'utilizzo ideale dell'alimentatore tampone è insieme ad un piccolo televisore anch'esso fornito di alimentazione autonoma a batteria.

Il circuito dell'alimentatore tampone è in fig. 5. La tensione necessaria durante i periodi di blackout è fornita da un accumulatore ermetico al piombo da 12V 3Ah (B₁).

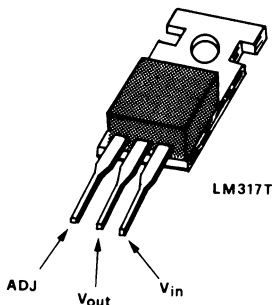


Figura 6

La configurazione dei due LM317T, utilizzati nel circuito, è in fig. 6; di questi due integrati uno (IC₁) viene utilizzato per ricaricare l'accumulatore e l'altro (IC₂) per stabilizzare la tensione ai circa 9V necessari per il computer.

La fig. 7 mostra una possibile realizzazione pratica dell'alimentatore. Anche per questo circuito vale quanto detto per l'avvisatore acustico circa la necessità di fare attenzione ed effettuare tutti i collegamenti nel modo giusto.

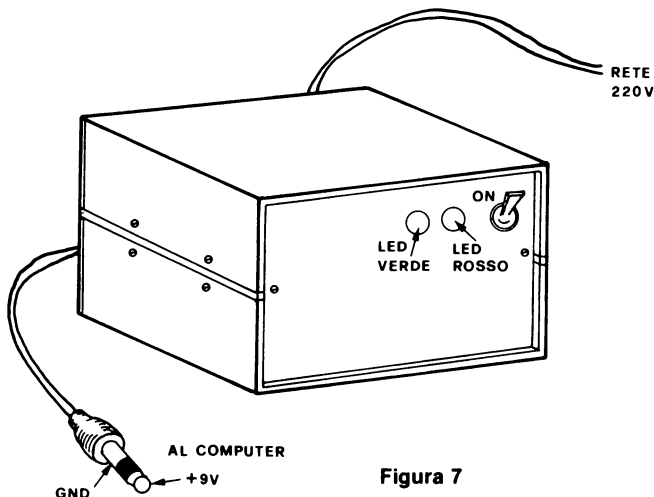


Figura 7

I due LM317T vanno montati su due piccoli dissipatori oppure sul retro del mobile metallico, in questo caso però isolandoli tra loro con delle miche e verificando tale isolamento con l'ohmmetro PRIMA di dare corrente all'apparecchio.

In particolare è MOLTO IMPORTANTE CONTROLLARE, PRIMA DI COL-

LEGARE L'ALIMENTATORE AL COMPUTER, che la polarità del JACK di uscita sia esattamente uguale a quella dell'alimentatore Sinclair con, quindi, il positivo (+9V) sulla punta, ed inoltre controllare che la tensione di uscita non sia superiore a 11-12V.



Figura 9

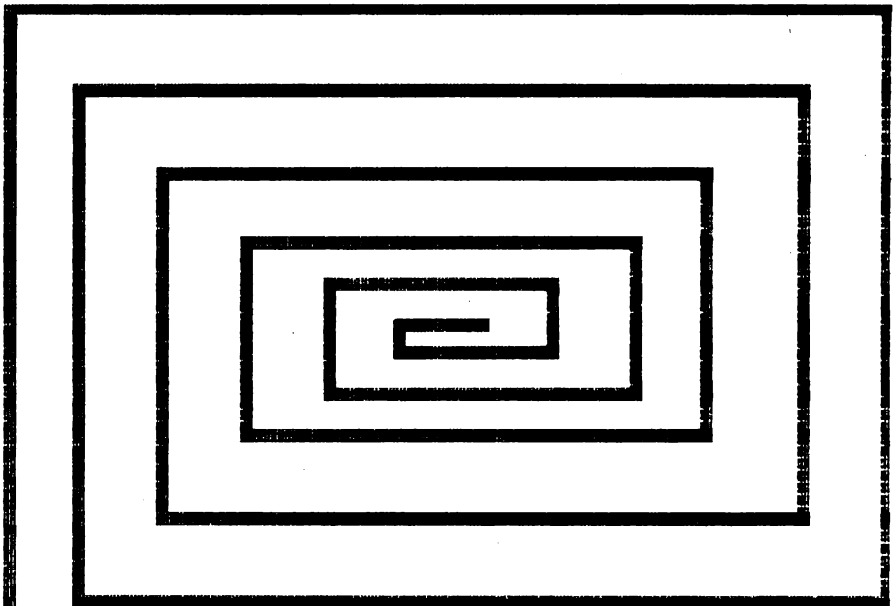


Figura 10

All'inizio, subito dopo RUN e NEWLINE appare un piccolo quadratino nero in basso a sinistra dello schermo ed il programma è predisposto per disegnare. Premendo uno degli 8 tasti rossi il quadratino si sposta nella direzione indicata sul tasto premuto lasciando dove passa una traccia.

Se si vuole spostare il quadratino per andare a disegnare in un'altra parte dello schermo o se si vuole effettuare una modifica al disegno già fatto, si può premere il tasto verde "MODIFICHE E SPOSTAMENTI" e quindi i tasti rossi necessari per spostarsi.

Una volta effettuata la modifica o lo spostamento si può ritornare a disegnare premendo il tasto verde "DISEGNO" e quindi i tasti rossi.

Il tasto verde "SCROLL" permette di spostare tutto il disegno fatto fino a quel

```

100 RAM LAVAGNA
200 RAM ALFANUMERICA
300 RAM E GRAFICA
1000 SLOW
1100 LET C=0
1200 LET T=0
1300 LET X=0
1400 LET Y=0
1450 LET U=0
1500 LET X=CODE INKEY$
1600 IF X<64 AND K=115 THEN LET
C=C+1
1700 IF X=65 THEN LET U=0
1800 IF X=66 THEN LET X=128
1900 IF X=67 THEN LET U=0
2000 IF X>0 AND X<64 THEN LET X=
X-1
2100 IF C<21 AND K=115 THEN LET
C=C+1
2200 IF C>0 AND K=114 THEN LET C
=C-1
2300 IF L<21 AND K=113 THEN LET
L=L+1
2400 IF L>0 AND K=112 THEN LET L
=L-1
2500 IF X=117 THEN SCROLL
2600 IF X=123 THEN LET T=0
2700 IF X=119 THEN LET T=1
2800 IF U=0 THEN PRINT AT L,C;CH
(X+128)
2900 IF U=1 THEN PRINT AT L,C;CH
(X+128)
3000 IF T=1 THEN PRINT AT L,C;"
3100 GOTO 150

```

Figura 11

momento verso l'alto. Infine il tasto verde "STOP" blocca l'esecuzione del programma.

Le figg. 9 e 10 mostrano due dei molti tipi di disegni realizzabili con il programma.

Lavagna alfanumerica e grafica

SLOW

Questo programma (fig. 11), a differenza di quello precedente, usa per disegnare l'istruzione PRINT AT e ciò permette di inserire nel disegno tutti i simboli grafici, le lettere, i numeri, ecc. anche in campo inverso.

Per i vari comandi viene utilizzata la tastiera N. 2 che, a differenza della tastiera utilizzata nel programma precedente, non viene sovrapposta alla tastiera dello ZX81 ma va sistemata al di sopra della prima fila di tasti (1...0); questo perchè i tasti vanno lasciati liberi per inserire i caratteri con cui realizzare i vari disegni.

Con il programma in funzione i vari comandi della tastiera speciale vanno azionati premendo contemporaneamente il tasto SHIFT ed uno dei tasti della prima fila. Premendo uno qualsiasi dei tasti (senza SHIFT) appare sul video il carattere o numero corrispondente, se si vuole lo stesso carattere ma inverso occorre premere SHIFT e il tasto 9 escluso lo spazio inverso che si ottiene premendo SHIFT e il tasto 2; per ritornare ai caratteri normali si preme SHIFT ed il tasto 3.

Una volta scelto il carattere si può iniziare a disegnare premendo SHIFT e i tasti 5, 6, 7, 8, oppure lo si può spostare usando gli stessi tasti ma premendo prima SHIFT

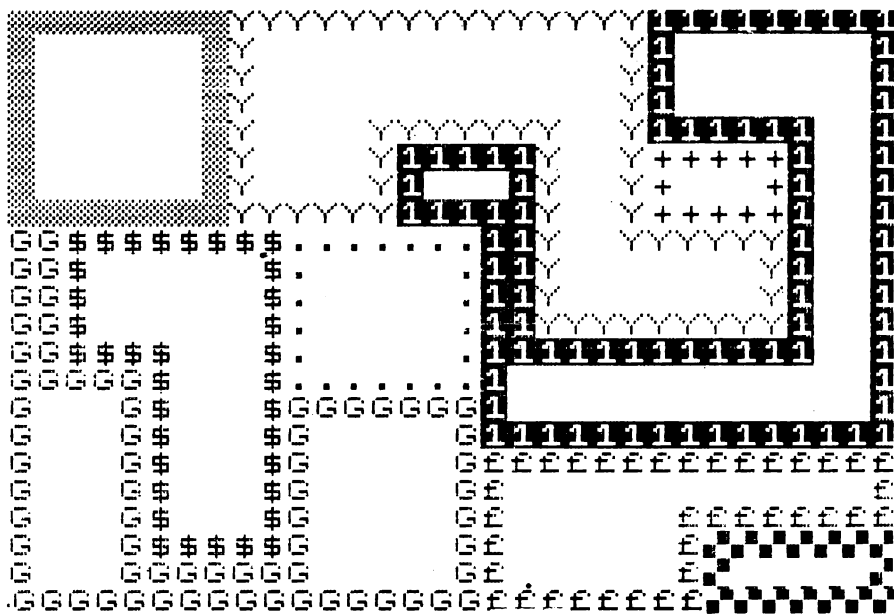


Figura 12

```

10 REM - RODITORE
100 SLOW
110 FOR A=1 TO 88
120 PRINT " ";
130 NEXT A
140 PRINT
150 LET L=21
160 LET C=0
170 PRINT AT L,C:"."
180 FOR T=1 TO 3
190 NEXT T
200 PRINT AT L,C:" "
210 IF C>0 AND INKEY$="5" THEN
LET C=C-1
220 IF L<21 AND INKEY$="6" THEN
LET L=L+1
230 IF L>0 AND INKEY$="7" THEN
LET L=L-1
240 IF C<31 AND INKEY$="8" THEN
LET C=C+1
250 GOTO 170

```

Figura 13

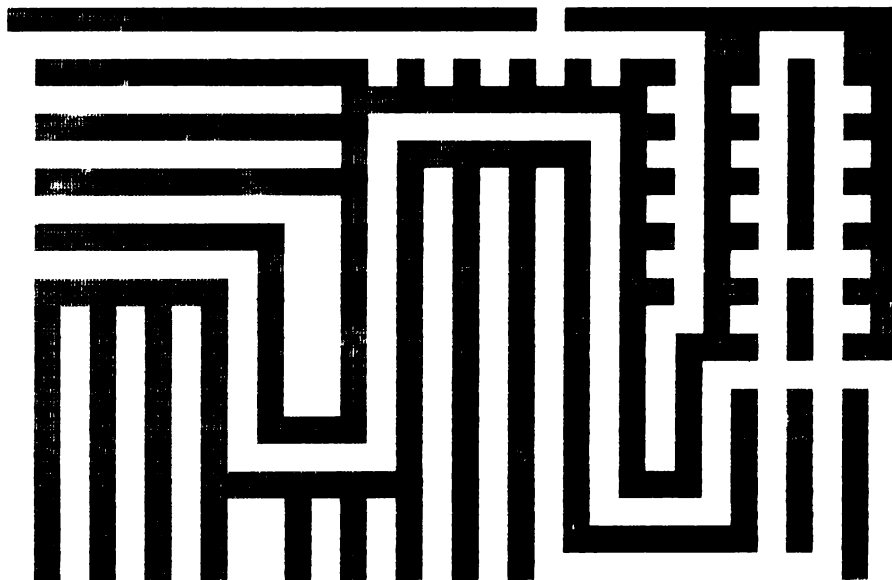


Figura 14

ed il tasto 0, tasto che si deve premere anche per effettuare delle modifiche al disegno.

Effettuata la modifica o lo spostamento, per ritornare alla scrittura si deve premere SHIFT ed il tasto 4.

La fig. 12 mostra un esempio di disegno realizzato con il programma appena esposto.

Roditore

SLOW

Questo programma grafico (fig. 13) inizia riempiendo lo schermo di spazi inversi o di qualsiasi altro carattere, numero o simbolo grafico.

Dopo di ciò appare nell'angolo in basso a sinistra un punto lampeggiante. Tale punto può essere spostato nelle quattro direzioni tramite i tasti 5, 6, 7, 8.

Dove passa il punto "mangia" una parte del rettangolo di caratteri o spazi inversi formatosi all'inizio del programma.

```
120 PRINT "████████████████████";
```

Figura 15

Un esempio di disegno realizzabile con il programma "Roditore" è una specie di labirinto visibile in fig. 14. La linea 120 contiene tra gli apici 8 spazi inversi che con le linee 110 e 130 producono sullo schermo un rettangolo nero.

```
120 PRINT "████████████████";
```

Figura 16

Per avere un diverso fondo la linea 120 può essere modificata come visibile nelle figure 15 - 16 - 17. In queste nuove linee i caratteri tra gli apici sono sempre 8 però non si tratta più di spazi inversi ma del carattere grafico del tasto A (fig. 15) o del

```
120 PRINT "A.A.A.A.A.A.A.A";
```

Figura 17

simbolo inverso del \$ (fig. 16) o ancora di una serie di punti inversi (fig. 17). Nel listato di fig. 13 si possono eliminare, eventualmente, le istruzioni 180 e 190 per rendere più veloce il movimento del punto.

Il programma può essere fermato con il tasto BREAK.


```

1000 REM      RODITORE      -2-
1100 REM      CON  PLOT-UNPLOT
1100 SCOW
1110 FOR A=1 TO 88
1120 PRINT " ";
1130 NEXT A
1140 PRINT
1150 LET X=0
1160 LET Y=0
1165 LET K=0
1170 IF K=0 THEN PLOT X,Y
1180 UNPLOT X,Y
1190 IF K=1 THEN PLOT X,Y
1200 IF INKEY$="S" THEN LET K=1
1210 IF INKEY$="M" THEN LET K=0
1220 IF X>0 AND INKEY$="S" THEN
LET X=X-1
1230 IF Y<43 AND INKEY$="7" THEN
LET Y=Y+1
1240 IF Y>0 AND INKEY$="6" THEN
LET Y=Y-1
1250 IF X<63 AND INKEY$="8" THEN
LET X=X+1
1260 GOTO 1170

```

Figura 18

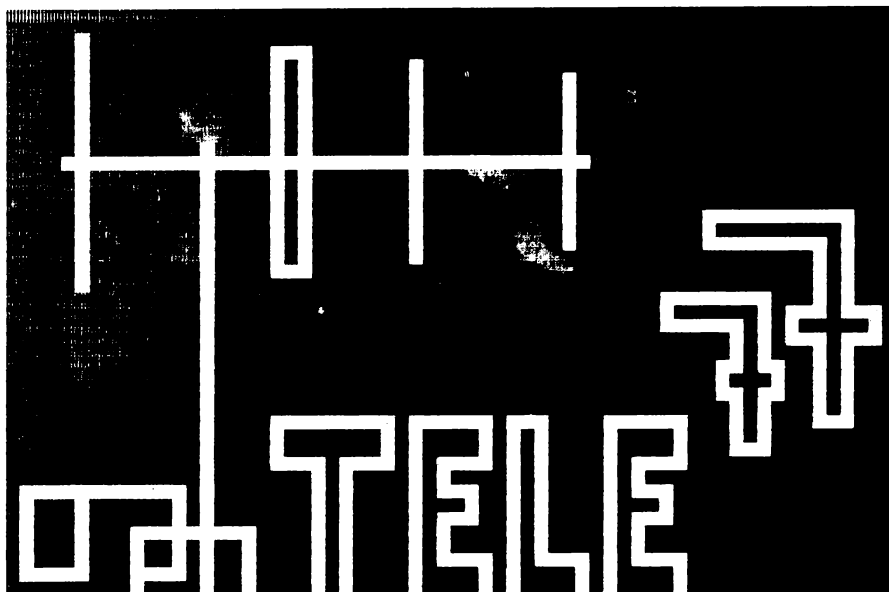


Figura 19

Questo secondo programma di Roditore grafico (fig. 18) è simile al precedente con la differenza che usa le istruzioni PLOT e UNPLOT, il che permette di realizzare dei disegni con lo stesso sistema ma con una definizione quattro volte maggiore, inoltre è anche possibile modificare un disegno già fatto o correggere eventuali errori. L'istruzione 120 contiene tra gli apici 8 spazi inversi, tale istruzione assieme alla 110 e 130 disegnano un grosso rettangolo nero sullo schermo, dopodichè il quadratino in basso a sinistra di tale rettangolo lampeggia.

Tramite i tasti 5, 6, 7, 8 si può disegnare spostando il quadratino lampeggiante nelle quattro direzioni.

Se si vuole modificare un disegno o si vuole spostare il quadratino lampeggiante in una zona del rettangolo scuro, si può premere prima il tasto "S" (Scrivo) premendo poi il tasto "M" (Mangia) per tornare a disegnare.

La fig. 19 mostra un esempio di disegno realizzato con questo programma.

Programma grafico

FAST
O SLOW

Il programma di fig. 20 può essere utilizzato per tracciare il grafico di una qualsiasi funzione matematica.

```

10  REM  PROGRAMMA GRAFICO
20  REM  -----
77  PRINT TAB 4; "ATTENDERE 2 MI
NUTI"
88  PAUSE 200
99  CLS
100 FAST
110 DIM A(64)
120 FOR K=1 TO 64
130 LET A(K)=K*K/110
140 NEXT K
150 SLOW
160 FOR B=0 TO 63
170 GOSUB 210
180 NEXT B
190 FAST
2000 STOP
2010 FOR E=B TO 63
2020 PLOT E,A(B+1)
2030 NEXT E
240 RETURN

```

Figura 20

La fig. 21, per esempio, mostra il grafico relativo alla funzione matematica presente nella linea 130 del programma di fig. 20 ($K \cdot K / 110$). Occorre tenere presente che per disegnare tale grafico sullo schermo il computer ha bisogno di circa 2 minuti di tempo.

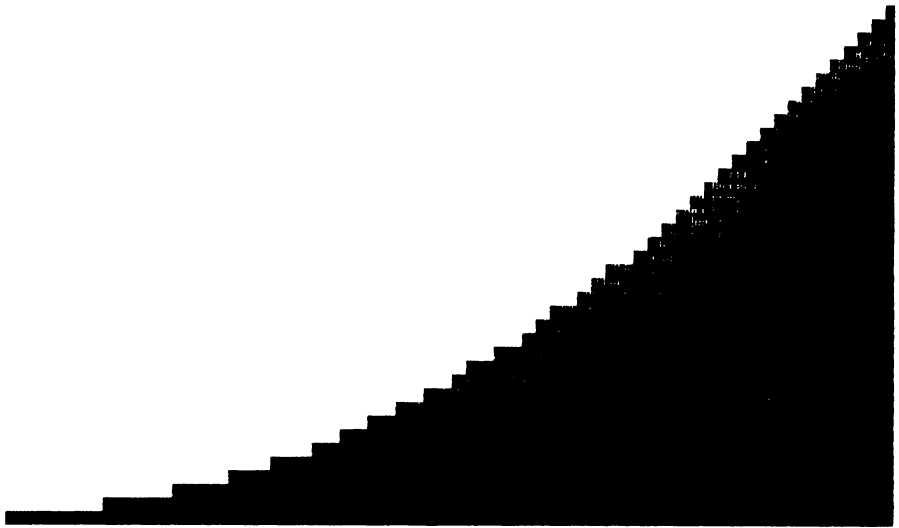


Figura 21

Se si vuole disegnare solo la curva della funzione matematica senza la parte scura di riempimento ed anche con una maggiore velocità di esecuzione, si può modificare il programma di fig. 20 sostituendo la linea 170 ~~GOSUB 210~~ con la linea 170 indicata in fig. 22.

```
170 PLOT B, A(B+1)
```

Figura 22

```
130 LET A(K) = RND * (K / 1.5)
```

Figura 23

```
130 LET A(K) = TAN K * 5
```

Figura 24

```
130 LET A(K) = LN K * 10
```

Figura 25

Nelle figure 23, 24, 25 vengono date altre tre versioni della linea 130 (da inserire nel programma di fig. 20) con tre diverse funzioni matematiche per ottenere altrettante diverse curve sullo schermo.

Programma universale per PRINT AT

**FAST
O SLOW**

Molto spesso all'inizio di un programma è necessario per prima cosa inserire in varie parti dello schermo tutta una serie di lettere, numeri e simboli, grafici e non, per formare delle frasi e/o dei disegni.

Questa operazione può richiedere anche molte decine di linee di programma contenenti l'istruzione PRINT AT.

```

10 REM   PROGRAMMA UNIVERSALE
20 REM   PER PRINT AT
30 REM   -IN FAST O SLOW-
100 SLOW
110 REM   CODICI LINEE IN L$
120 LET L$="00103333455566344444
SSSSS777888999AAAABBBBCCCCDD"
130 REM   CODICI COLONNE IN C$
140 LET C$="010KJLMLKKKLMTPQRST
PQRST5674563693456789369468567"
150 REM   CARATTERI IN X$
160 LET X$="01TESTST APRINT"
170 IF LEN L$ <> LEN C$ OR LEN C$
<> LEN X$ THEN GOTO 220
180 FOR P=1 TO LEN X$
190 PRINT AT CODE L$(P TO )-28,
CODE C$(P TO )-28;X$(P TO P)
200 NEXT P
210 STOP
220 PRINT AT 0,10;" ERRORE "
230 PROCSE 100
240 PROCST
250 LIST

```

Figura 26

Se però si stabilisce prima con esattezza quali caratteri vanno inseriti e per ciascuno in quale linea e colonna, tutta questa fase può essere svolta (in FAST o SLOW) dal programma di fig. 26.

I caratteri da inserire sono nella variabile X\$ mentre le linee e le colonne in cui inserire tali caratteri sono contenute in forma codificata nelle variabili L\$ (linea) e C\$ (colonna).

Programma universale per PLOTTAGGIO

**FAST
O SLOW**

Come nel caso precedente per PRINT AT, anche una lunga serie di linee di programma con PLOT può essere condensata in un programma più breve come è visibile in fig. 28.

In tale programma le coordinate X e Y di ciascuno punto da plottare sono inserite

```

1000 REM PROGRAMMA UNIVERSALE
1100 REM PER PLOTTAGGIO
1200 REM - IN FAST O SLOW -
1300 REM
1400 REM COORDINATE ORIZZ. IN X#
1500 REM X# = 13 14 15 16 17 18 1
1600 REM PLOT 00 00 00 00 04 04 04 04
1700 REM 00 00 00 00 00 00 00 00 00 00 00 00
1800 REM 00 00 00 00 00 00 00 00 00 00 00 00
1900 REM 00 00 00 00 00 00 00 00 00 00 00 00
2000 REM 00 00 00 00 00 00 00 00 00 00 00 00
2100 REM COORDINATE VERT. IN Y#
2200 REM Y# = 05 06 05 05 05 05 05
2300 REM PLOT 00 00 00 00 00 00 00 00 00 00 00 00
2400 REM 00 00 00 00 00 00 00 00 00 00 00 00
2500 REM 00 00 00 00 00 00 00 00 00 00 00 00
2600 REM 00 00 00 00 00 00 00 00 00 00 00 00
2700 REM 00 00 00 00 00 00 00 00 00 00 00 00
2800 REM 00 00 00 00 00 00 00 00 00 00 00 00
2900 REM 00 00 00 00 00 00 00 00 00 00 00 00
3000 REM 00 00 00 00 00 00 00 00 00 00 00 00
3100 REM 00 00 00 00 00 00 00 00 00 00 00 00
3200 REM 00 00 00 00 00 00 00 00 00 00 00 00
3300 REM 00 00 00 00 00 00 00 00 00 00 00 00
3400 REM 00 00 00 00 00 00 00 00 00 00 00 00
3500 REM 00 00 00 00 00 00 00 00 00 00 00 00
3600 REM 00 00 00 00 00 00 00 00 00 00 00 00
3700 REM 00 00 00 00 00 00 00 00 00 00 00 00
3800 REM 00 00 00 00 00 00 00 00 00 00 00 00
3900 REM 00 00 00 00 00 00 00 00 00 00 00 00
4000 REM 00 00 00 00 00 00 00 00 00 00 00 00
4100 REM 00 00 00 00 00 00 00 00 00 00 00 00
4200 REM 00 00 00 00 00 00 00 00 00 00 00 00
4300 REM 00 00 00 00 00 00 00 00 00 00 00 00
4400 REM 00 00 00 00 00 00 00 00 00 00 00 00
4500 REM 00 00 00 00 00 00 00 00 00 00 00 00
4600 REM 00 00 00 00 00 00 00 00 00 00 00 00
4700 REM 00 00 00 00 00 00 00 00 00 00 00 00
4800 REM 00 00 00 00 00 00 00 00 00 00 00 00
4900 REM 00 00 00 00 00 00 00 00 00 00 00 00
5000 REM 00 00 00 00 00 00 00 00 00 00 00 00
5100 REM 00 00 00 00 00 00 00 00 00 00 00 00
5200 REM 00 00 00 00 00 00 00 00 00 00 00 00
5300 REM 00 00 00 00 00 00 00 00 00 00 00 00
5400 REM 00 00 00 00 00 00 00 00 00 00 00 00
5500 REM 00 00 00 00 00 00 00 00 00 00 00 00
5600 REM 00 00 00 00 00 00 00 00 00 00 00 00
5700 REM 00 00 00 00 00 00 00 00 00 00 00 00
5800 REM 00 00 00 00 00 00 00 00 00 00 00 00
5900 REM 00 00 00 00 00 00 00 00 00 00 00 00
6000 REM 00 00 00 00 00 00 00 00 00 00 00 00
6100 REM 00 00 00 00 00 00 00 00 00 00 00 00
6200 REM 00 00 00 00 00 00 00 00 00 00 00 00
6300 REM 00 00 00 00 00 00 00 00 00 00 00 00
6400 REM 00 00 00 00 00 00 00 00 00 00 00 00
6500 REM 00 00 00 00 00 00 00 00 00 00 00 00
6600 REM 00 00 00 00 00 00 00 00 00 00 00 00
6700 REM 00 00 00 00 00 00 00 00 00 00 00 00
6800 REM 00 00 00 00 00 00 00 00 00 00 00 00
6900 REM 00 00 00 00 00 00 00 00 00 00 00 00
7000 REM 00 00 00 00 00 00 00 00 00 00 00 00
7100 REM 00 00 00 00 00 00 00 00 00 00 00 00
7200 REM 00 00 00 00 00 00 00 00 00 00 00 00
7300 REM 00 00 00 00 00 00 00 00 00 00 00 00
7400 REM 00 00 00 00 00 00 00 00 00 00 00 00
7500 REM 00 00 00 00 00 00 00 00 00 00 00 00
7600 REM 00 00 00 00 00 00 00 00 00 00 00 00
7700 REM 00 00 00 00 00 00 00 00 00 00 00 00
7800 REM 00 00 00 00 00 00 00 00 00 00 00 00
7900 REM 00 00 00 00 00 00 00 00 00 00 00 00
8000 REM 00 00 00 00 00 00 00 00 00 00 00 00
8100 REM 00 00 00 00 00 00 00 00 00 00 00 00
8200 REM 00 00 00 00 00 00 00 00 00 00 00 00
8300 REM 00 00 00 00 00 00 00 00 00 00 00 00
8400 REM 00 00 00 00 00 00 00 00 00 00 00 00
8500 REM 00 00 00 00 00 00 00 00 00 00 00 00
8600 REM 00 00 00 00 00 00 00 00 00 00 00 00
8700 REM 00 00 00 00 00 00 00 00 00 00 00 00
8800 REM 00 00 00 00 00 00 00 00 00 00 00 00
8900 REM 00 00 00 00 00 00 00 00 00 00 00 00
9000 REM 00 00 00 00 00 00 00 00 00 00 00 00
9100 REM 00 00 00 00 00 00 00 00 00 00 00 00
9200 REM 00 00 00 00 00 00 00 00 00 00 00 00
9300 REM 00 00 00 00 00 00 00 00 00 00 00 00
9400 REM 00 00 00 00 00 00 00 00 00 00 00 00
9500 REM 00 00 00 00 00 00 00 00 00 00 00 00
9600 REM 00 00 00 00 00 00 00 00 00 00 00 00
9700 REM 00 00 00 00 00 00 00 00 00 00 00 00
9800 REM 00 00 00 00 00 00 00 00 00 00 00 00
9900 REM 00 00 00 00 00 00 00 00 00 00 00 00
10000 REM 00 00 00 00 00 00 00 00 00 00 00 00

```

Figura 28

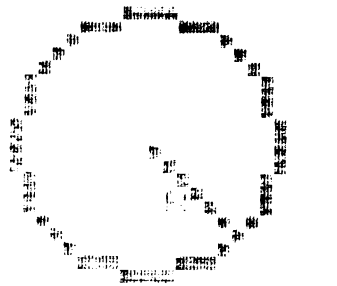


Figura 29

rispettivamente nelle variabili X\$ e Y\$. Come è visibile tra un numero e l'altro va inserito uno spazio.

Naturalmente le coordinate vanno comprese tra 00 e 63 in X\$ e tra 00 e 43 in Y\$; i numeri inferiori a 10 vanno inseriti con uno 0 davanti (per es. il numero 7 va inserito come 07).

Le coordinate inserite come esempio nel programma di fig. 28 danno il disegno di fig. 29; la lettera R viene aggiunta a disegno ultimato dall'istruzione contenuta nella linea 300. Realizzare lo stesso disegno senza questo programma avrebbe richiesto 58 linee di programma con PLOT.

Plottaggio QUADRATI

FAST
O SLOW

Quando occorre disegnare varie figure quadrate si può utilizzare il programma di fig. 30 che disegna quadrati di qualsiasi dimensione inserendo le coordinate X e Y

```

1000 PRINT "INSERIRE X"
2000 INPUT X
3000 CLS
4000 PRINT "INSERIRE Y"
5000 INPUT Y
6000 CLS
7000 PRINT "INSERIRE L"
8000 INPUT L
9000 GOSUB 4000
9999 STOP
9998 REM -----

4000 REM PLOTTAGGIO QUADRATI
4010 SLOW
4020 FOR A=X TO X+L
4030 IF A<0 OR A>63 THEN GOTO 40
4040 IF Y>=0 AND Y<=43 THEN PLOT
4050 IF Y+L>=0 AND Y+L<=43 THEN
PLOT A,Y+L
4060 NEXT A
4070 FOR A=Y TO Y+L
4080 IF A<0 OR A>43 THEN GOTO 41
4090 IF X>=0 AND X<=63 THEN PLOT
X+L,A
4100 NEXT A
4110 RETURN

```

Figura 30

(anche negative) dell'angolo in basso a sinistra del quadrato stesso e naturalmente il lato "L" (vedere fig. 31).

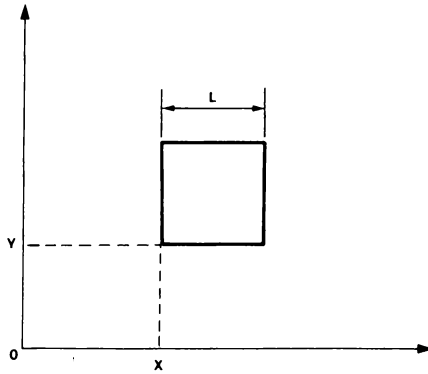


Figura 31

```

000010 PLOT. CASUALE QUADRATI
000020 .....
000030 RND
000040 FOR
000050 DO = 1 TO 7
000060 LET X = INT (50*RND)
000070 LET Y = INT (25*RND)
000080 LET L = INT (30*RND)
000090 GOTO 10
000100 B 4000
000110 NEXT L
000120 B
000130 P
-----

```

Figura 32

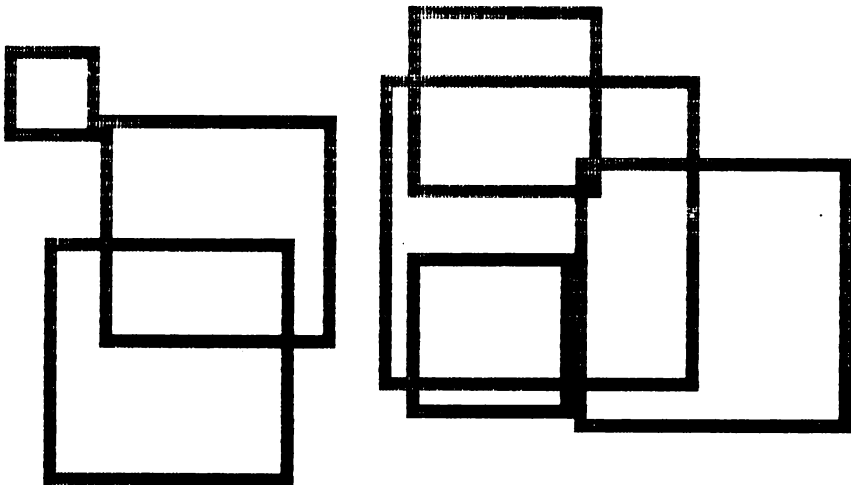


Figura 33

Come esempio si possono sostituire le istruzioni da 10 a 999 in fig. 30 con le nuove istruzioni espote in fig. 32 per produrre sullo schermo il disegno di 7 quadrati con posizione e grandezza casuali (fig. 33).

Plottaggio CERCHI

**FAST
O SLOW**

È possibile disegnare anche dei cerchi usando il programma di fig. 34. In tale programma occorre inserire all'inizio le coordinate X e Y (anche negative) del

```

10 PRINT "INSERIRE X"
20 INPUT X
30 CLS
40 PRINT "INSERIRE Y"
50 INPUT Y
60 CLS
70 PRINT "INSERIRE R"
80 INPUT R
90 CLS
1000 GOSUB 5000
1100 STOP
1200 RETURN

5000 REM PLOTTAGGIO CERCHI
5010 SLOW
5020 FOR C=0 TO 360 STEP 1+INT
5030 NEXT C
5040 LET X=C*PI/180
5050 LET Y=R*SIN X
5060 LET B=R*COS X+Y
5070 IF B >=0 AND B <=90 AND Y >=0
5080 THEN PLOT ,A,B
5090 NEXT C
5100 RETURN

```

Figura 34

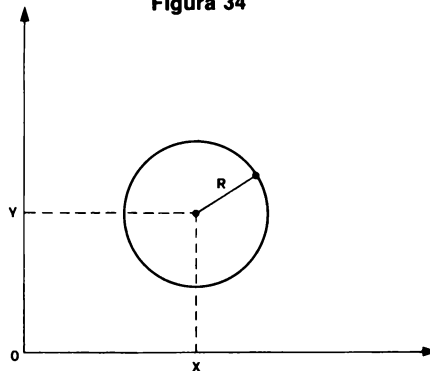


Figura 35

```

10 REM CERCHI CONCENTRICI
20 REM
30 LET X=0
40 LET Y=0
50 FOR R=5 TO 30 STEP 5
60 GOSUB 5000
70 NEXT R
100 STOP
999 REM -----

```

Figura 36

centro ed il raggio "R" del cerchio da disegnare (vedere fig. 35). Anche in questo caso come esempio vengono date in fig. 36 alcune nuove istruzioni da inserire nel programma di fig. 34 in sostituzione delle prime 12 istruzioni (da 10 a 999) per ottenere una serie di cerchi concentrici (fig. 37).

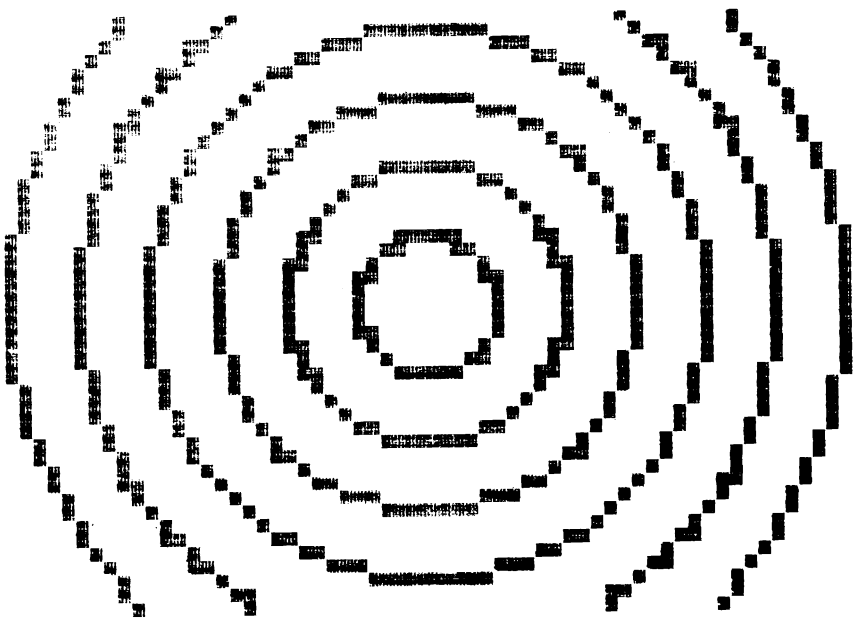


Figura 37

Plottaggio ELLISSI

FAST
O SLOW

Per terminare questa breve serie di figure geometriche da disegnare con il computer viene dato anche un programma per disegnare le ellissi (fig. 38), a partire dalle

```

0000 PRINT "INSERIRE [X]"
0010 INPUT X
0020 PRINT "INSERIRE [Y]"
0030 INPUT Y
0040 PRINT "INSERIRE [RMA]"
0050 INPUT RMA
0060 PRINT "INSERIRE [RMI]"
0070 INPUT RMI
0080 GOTO 0000
0090 END
-----
0100 REM PLOTTAGGIO [ELLISSI]
0110 GOTO 0120
0120 IF RMI>RMA THEN LET S=RMA
0130 IF RMI<RMA THEN LET S=RMI
0140 FOR C=0 TO 360 STEP 1+INT (
0150 )
0160 LET K=C*PI/180
0170 LET A=(RMA-0.5)*(1+COS K)-R
0180 LET B=(RMI-0.5)*(1+SIN K)-R
0190 IF A>=0 AND A<=80 AND B>=0
0200 B<=40 THEN PLOT A,B
0210 NEXT C.
0220 RETURN
    
```

Figura 38

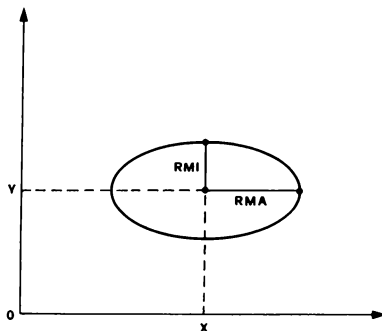


Figura 39

coordinate X e Y (anche negative) del centro e dei valori del raggio maggiore (RMA) e dal raggio minore (RMI) come visibile in fig. 39. In figura 40 e figura 41 sono mostrati due tipi diversi di ellissi con indicati in alto i valori inseriti per ottenerle.

X=32 **Y=22** **RMA=30** **RMI=15**

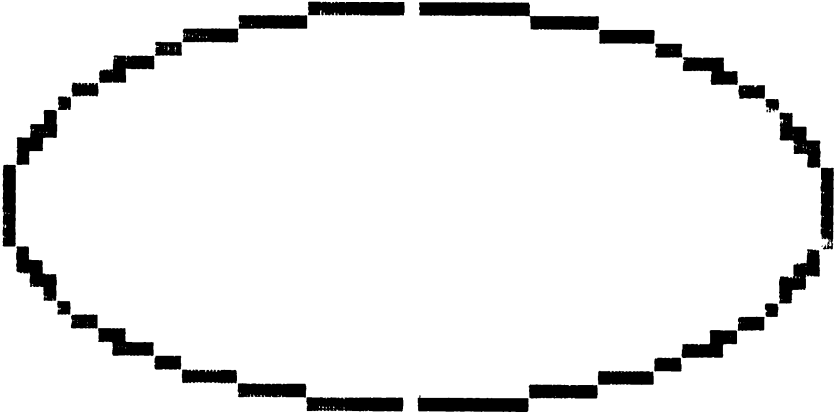


Figura 40

X=30 **Y=20** **RMA=10** **RMI=20**

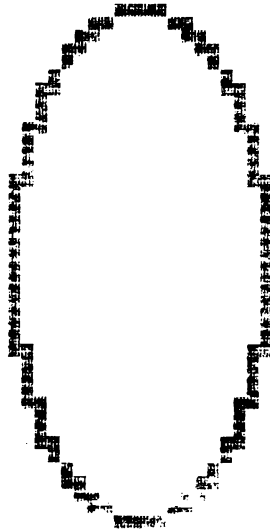


Figura 41

L'esempio fornito per questo programma è particolarmente bello da realizzare, infatti inserendo le istruzioni di fig. 42 nel listato di fig. 38 si ottiene il disegno di un cilindro Tridimensionale (fig. 43), e chi dispone di uno ZX81 (che ha la funzione SLOW) potrà anche vedere il cilindro mentre si forma.

Cilindri di altra forma possono essere disegnati usando altri valori per X, Y, RMA, RMI, H (altezza).

Disegni

I prossimi tre programmi mostrano le capacità del computer nell'eseguire dei disegni. Tali disegni possono essere utili in altri programmi oppure essere usati come esempio dimostrativo fine a se stesso.

Tutti e tre i programmi possono funzionare anche in FAST però in SLOW sono molto meglio perchè si vedono i disegni mentre si formano.

Cielo stellato

FAST
O SLOW

Una particolarità del programma (fig. 44) per disegnare un cielo stellato è che il numero delle stelle, piccole e grandi, è maggiore in alto e diminuisce in basso verso l'orizzonte (fig. 45) così come nella realtà.

NEW YORK

FAST
O SLOW

Con poche linee di programma (fig. 46) il computer è anche in grado di disegnare in modo casuale il panorama di una città come New York con i suoi grattacieli (fig. 47).

```

100000 REM CIELO STELLATO
110000 REM -----
120000 SLOW
130000 CLAND
140000 FOR L=1 TO 28
150000 FOR C=0 TO 31
160000 LET S=1+INT ((9+L*2)*RND)
170000 IF S=2 THEN PRINT AT L-1,C; " "
180000 IF S=4 THEN PRINT AT L-1,C; " "
190000 NEXT C
200000 NEXT L

```

Figura 44

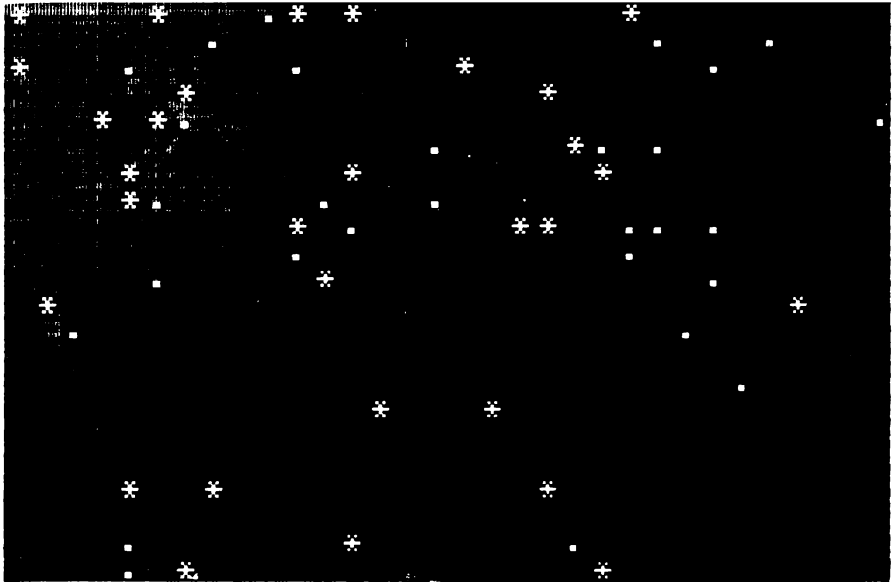


Figura 45

```

100 REM      NEW YORK
110 REM      -----
120 REM
130 LET X=0
140 LET H=2+INT (14*RND)
150 FOR L=1 TO 2+INT (2*RND)
160 FOR Y=0 TO H
170 PLOT X,Y
180 IF Y>0 AND Y<H AND (4*RND)<3 THEN UNPLOT X,Y
190 NEXT Y
200 LET X=X+1
210 IF X=64 THEN GOTO 220
220 NEXT L
230 GOTO 130

```

Figura 46



Figura 47

Se poi si aggiungono al programma di fig. 46 le linee 77, 88, 99 (fig. 48), le linee 220, 230, 240 (fig. 49) e si sostituiscono le linee 160, 165, con le nuove due linee di fig. 50 si può anche ottenere lo stesso panorama di New York ma di notte (fig. 51).

```
77 FOR P=1 TO 88  
88 PRINT "██████████";  
99 NEXT P
```

Figura 48

```
220 FOR T=0 TO 63  
230 PLOT T,0  
240 NEXT T
```

Figura 49

```
160 UNPLOT X,Y  
165 IF Y>0 AND Y<H AND (4*RND) >  
3 THEN PLOT X,Y
```

Figura 50

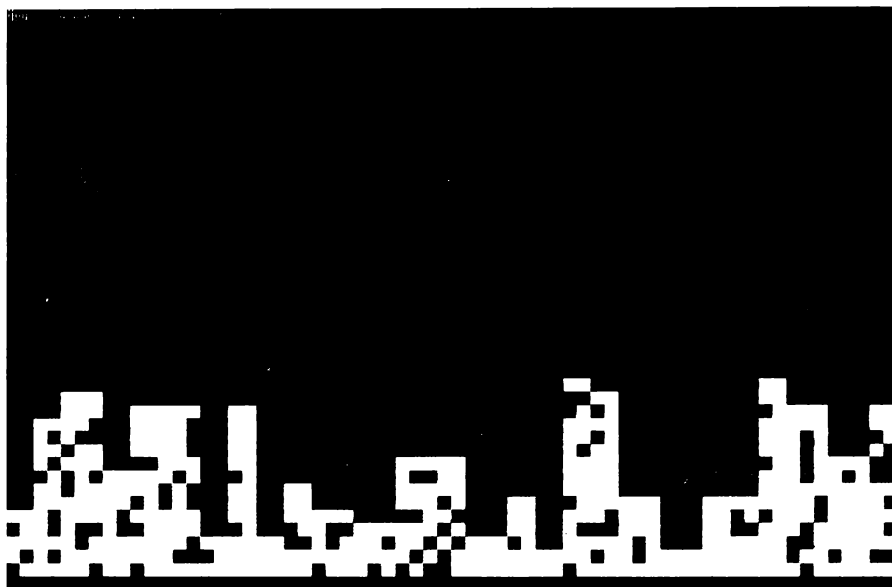


Figura 51

L'ultimo programma grafico (fig. 52) disegna in modo casuale delle montagne (fig. 53) alle quali può essere aggiunto anche un orizzonte inserendo nel programma le linee 270, 280, 290 (fig. 54).

```

10  REM  MONTAGNE
20  REM  -----
100  SLOW
110  LET  A=5
120  LET  M=1
130  LET  X=0
135  RAND
140  LET  E=1+INT (8*RND)
150  FOR  H=1  TO  E
160  IF  M=1  THEN  FOR  Y=0  TO  ABS
(A+H)
170  IF  M=-1  THEN  FOR  Y=0  TO  ABS
(A-H)
180  IF  M=1  AND  A+H>=0  OR  M=-1  A
ND  A-H>=0  THEN  PLOT  X,Y
190  NEXT  Y
200  LET  X=X+1
210  IF  X=64  THEN  GOTO  270
220  NEXT  H
230  IF  M=1  THEN  LET  A=A+E
240  IF  M=-1  THEN  LET  A=A-E
250  LET  M=M*(-1)
260  GOTO  140

```

Figura 52

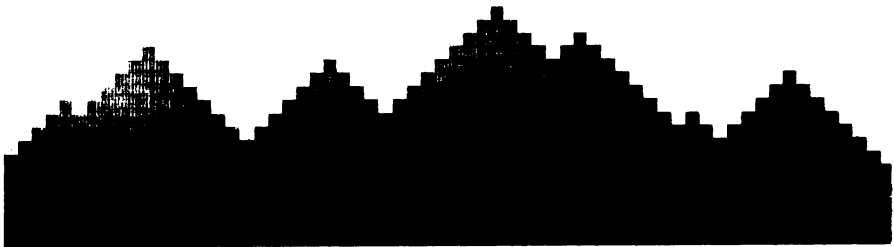


Figura 53

```

270  FOR  T=0  TO  31
280  PRINT  AT  21,T;  "███"
290  NEXT  T

```

Figura 54

È anche possibile, inoltre, aggiungendo le linee 77, 88, 99 (fig. 55) e sostituendo la linea 180 con la nuova linea di fig. 56, ottenere delle montagne chiare di notte.

```
77 FOR P=1 TO 88  
88 PRINT " ██████████ ";  
99 NEXT P
```

Figura 55

```
180 IF M=1 AND A+H>=0 OR M=-1 AND  
A-H>=0 THEN UNPLOT X,Y
```

Figura 56

PROGRAMMI VARI

Progettazione di Multivibratori Monostabili con 555

FAST
O SLOW

Una delle applicazioni più interessanti del computer è nella progettazione di circuiti elettronici. Un esempio pratico è il programma di fig. 57 che permette di calcolare i valori di un Multivibratore Monostabile con il timer 555 (fig. 58).

```
10 REM PROGETTAZIONE DI
20 REM MULTIVIBRATORE
30 REM MONOSTABILE CON 555
100 SLOW
110 PRINT "INSERIRE RA ( 1--100
00 KOHM ):"
115 PRINT "E PREMERE NEWLINE"
120 INPUT RA
130 IF RA=0 THEN GOTO 150
140 IF RA>10000 OR RA<1 THEN GO
TO 120
150 CLS
160 PRINT "INSERIRE C ( 0.001--
100 UF ):"
165 PRINT "E PREMERE NEWLINE"
170 INPUT C
180 IF C=0 THEN GOTO 200
190 IF C>100 OR C<0.001 THEN GO
TO 170
200 CLS
210 PRINT "INSERIRE T ( 0.01--1
000000 PERIODO ):"
215 PRINT "E PREMERE NEWLINE"
220 INPUT T
230 IF T=0 THEN GOTO 250
240 IF T>1000000 OR T<0.01 THEN
GOTO 220
250 CLS
260 IF RA=0 AND C=0 AND T=0 THE
N GOTO _110
```

Figura 57 (Continua)

```

270 IF RA>0 AND C>0 AND T>0 THE
N GOTO 110
280 IF RA=0 AND C=0 OR RA=0 AND
T=0 OR C=0 AND T=0 THEN GOTO 11
0
290 IF RA>0 AND C>0 THEN PRINT
300 " " ; TAB 6; RA; " KOHM"; TAB 1; TAB
0 " " ; TAB 6; C; " UF"; TAB 1; TAB
0 " " ; TAB 6; 1.1*RA*C; " MSEC."
300 IF RA>0 AND T>0 THEN PRINT
310 " " ; TAB 6; RA; " KOHM"; TAB 1; TAB
0 " " ; TAB 6; T/(1.1*RA); " UF"; T
0 " " ; TAB 6; T; " MSEC."
310 IF C>0 AND T>0 THEN PRINT "
320 RA"; TAB 6; T/(1.1*C); " KOHM"; TAB
1; TAB 0; " " ; TAB 6; C; " UF"; TAB
1; TAB 0; " " ; TAB 6; T; " MSEC."
320 PRINT AT 9,0; "PREMERE NEWLI
N M"
330 PRINT "PER ALTRI VALORI"
340 INPUT X$
350 CLS
360 GOTO 110

```

Figura 57 (Fine)

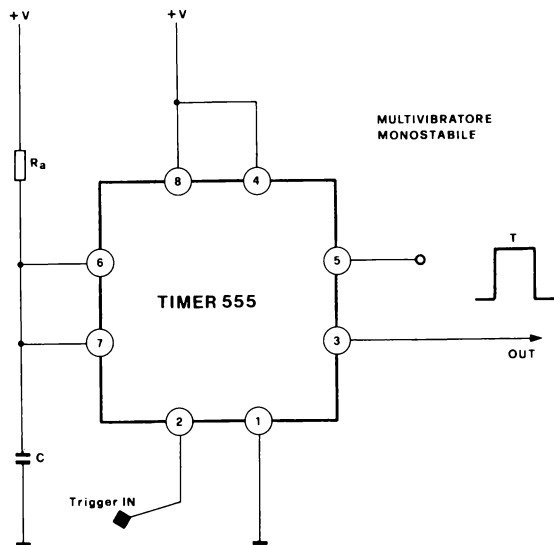


Figura 58

Facendo girare il programma usciranno in sequenza le tre scritte di figg. 59, 60, 61 che chiedono l'inserimento dei tre valori. Se ad esempio si vuole conoscere il valore del tempo T a partire dalla resistenza RA e dal condensatore C si dovrà inserire

```
INSERIRE RA ( 1--10000 KOHM )  
E PREMERE NEWLINE
```

Figura 59

```
INSERIRE C ( 0.001--100 UF )  
E PREMERE NEWLINE
```

Figura 60

```
INSERIRE T ( 0.01--100000 MSEC. )  
E PREMERE NEWLINE
```

Figura 61

```
RA      820    KOHM  
C       0.1    UF  
T       90.2   MSEC.
```

```
PREMERE NEWLINE  
PER ALTRI VALORI
```

Figura 62

prima RA e poi C mentre per T (fig. 61) si dovrà inserire il valore 0, in modo simile si dovrà operare se si vuole ricavare RA da C e T o C da RA e T. Un tipico risultato fornito dal programma è visibile in fig. 62.

Contapezzi

SLOW

Un uso insolito dello ZX81 può essere quello di contapezzi (fig. 63) capace di contare da 0 a 99.999.999 pezzi.

Facendo girare il programma esce sul video la scritta di fig. 64. Ogni volta che si preme NEWLINE il numero viene incrementato di uno. Oltre a ciò si può inserire anche un numero qualsiasi da aggiungere (per es. 337115) che premendo NEWLI-

```

100 REM CONTAPEZZI DA 0
1000 REM A 99.999.999 PEZZI
1100 SLOW
1110 PRINT "PREMERE NEWLINE PER
1 1000 PRINT NO:"
1200 PRINT "0 INTRODURRE N° PEZZI
IN FUCO IN:"
140 PRINT "MENO (CON -) E PREME
RE
150 PRINT AT 10,0; "■■■■■
160 PRINT AT 11,0; "■■■■■
170 PRINT AT 12,0; "■■■■■

190 LET N=0
200 LET C=0
210 IN PUT C
220 IF N#=# THEN LET N#="1"
230 IF CODE N#<>22 THEN FOR A=1
TO 240 LET N#
240 IF CODE N#=22 THEN FOR A=2
TO 250 LET N#
250 IF CODE N#(A TO )<28 OR COD
E N#(A TO )>37 THEN GOTO 600
260 NEXT A
270 IF (N+VAL N#)>(1E8-1) OR (N
+VAL N#)<0 THEN GOTO 600
280 LET N=N+VAL N#
290 DIM B$(1,0-LEN STR$ N)
300 LET A#=STR$ N+B$(1)
310 FOR P=1 TO LEN A#-1
320 LET C=P*2+14
330 LET J=CODE A#(P TO )
340 IF J=0 THEN GOSUB 1997
350 IF J>0 THEN GOSUB (J-26)*10
360 NEXT P
370 GOTO 210
380 FOR E=1 TO 3
390 PRINT AT 15,5; "NUMERO ERRAT
O O Eccessivo"
400 LET T="TAN 0
410 PRINT AT 15,5; "

```

Figura 63 (Continua)

NE viene sommato al numero presente sullo schermo, oppure si può inserire un numero preceduto dal segno “-” e che premendo NEWLINE viene sottratto dal numero in quel momento presente sullo schermo. Se si cerca di sommare o di sottrarre un numero che farebbe scendere il numero presente sullo schermo sotto lo

PREMERE NEWLINE PER 1 SOLO PEZZO
O INTRODURRE N° PEZZI IN PIU' O IN
MENO (CON -) E PREMERE NEWLINE

N° PEZZI

NUMERO ERRATO O ECCESSIVO

Figura 64

PREMERE NEWLINE PER 1 SOLO PEZZO
O INTRODURRE N° PEZZI IN PIU' O IN
MENO (CON -) E PREMERE NEWLINE

N° PEZZI 47 158302

Figura 65

0 o lo farebbe andare oltre il numero massimo di pezzi, il programma segnala l'errore nel modo indicato sempre in fig. 64. Un esempio d'uso dei contapezzi è in fig. 65.

La parte del programma che scrive i numeri di grandi dimensioni può essere utile da inserire anche in altri programmi.

Del contapezzi viene data anche una versione più breve e più rapida da scrivere (fig. 66) che però non usa i numeri grandi ma presenta sul video i numeri di dimensioni normali (fig. 67).


```

1000 REM CONTAREZZI DA 0
1001 REM A 99.999.999 PEZZI
1100 STOP
1110 PRINT "PREMERE NEWLINE PER
1 1900 LO PRINT NO"
1200 PRINT "O INTRODURRE N° PEZZI
IN PIU' O IN"
140 PRINT "O INTRODURRE N° PEZZI
150 PRINT "MENO (CON -) E PREME
RE NEWLINE"
160 PRINT AT 10,2;"N° PEZZI"
190 LET N=0
210 INPUT N$
2200 IF N$="# THEN LET N$="1"
2300 IF CODE N$ <> 22 THEN FOR A=1
TO 240 LET N$
240 IF CODE N$=22 THEN FOR A=2
TO 250 LET N$
250 IF CODE N$(A TO ) < 28 OR COD
E N$(A TO ) > 37 THEN GOTO 600
260 NEXT A
270 IF (N+VAL N$) > (1E8-1) OR (N
+VAL N$) < 0 THEN GOTO 600
280 LET N=N+VAL N$
290 DIM B$(1,9)-LEN STR$ N)
300 LET B$=STR$ N+B$(1)
310 PRINT AT 10,10;A$
320 GOTO 210
330 FOR E=1 TO 3
340 PRINT AT 15,0;"INSERITO N°
PEZZI TO ACCRESCERE"
350 LET T=TAN 2
360 PRINT AT 15,0;"
640 LET T=TAN 2
650 NEXT E
660 GOTO 210

```

Figura 66

PREMERE NEWLINE PER 1 SOLO PEZZO
O INTRODURRE N° PEZZI IN PIU' O IN
MENO (CON -) E PREMERE NEWLINE
N° PEZZI 07654321

Figura 67

Una delle applicazioni del programma di "Scritte rotanti" (fig. 68) può essere nella pubblicità e nell'informazione in genere.

```

10 REM SCRITTE ROTANTI
100 SLOW
110 PRINT "INSERIRE FRASE"
120 INPUT A$
130 CLS
140 PRINT "INSERIRE VELOCITA" (
150 )
160 INPUT T
170 CLS
175 PRINT AT 3,0;"
180 PRINT AT 7,0;"
190 IF LEN A# < 32 THEN DIM H$(32
LEN A# + 10)
195 IF LEN A# > 31 THEN DIM H$(10
200 LET B# = A# + I#
210 LET C# = B#
220 PRINT AT 5,0;C$(1 TO 32)
230 LET C# = C$(2 TO )
240 IF LEN C# < 32 THEN LET C# = C#
+
250 FOR D=1 TO T
260 NEXT D
270 GOTO 200

```

Figura 68

QUESTA E' UNA SCRITTA ROTANTE .

Figura 69

NA SCRITTA ROTANTE. QUESTA E' UN

Figura 70

Dopo avere premuto RUN e NEWLINE si inserisce prima la frase anche di molti caratteri e poi la velocità di rotazione. Dopodichè i caratteri, i numeri e i simboli della frase inserita iniziano a ruotare da destra a sinistra. Le figure 69 e 70, 71 e 72, 73 e 74 mostrano tre esempi di scritte rotanti ciascuno dei

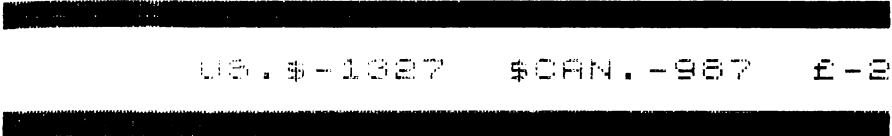


Figura 71

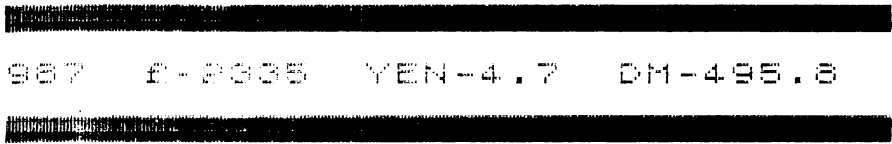


Figura 72

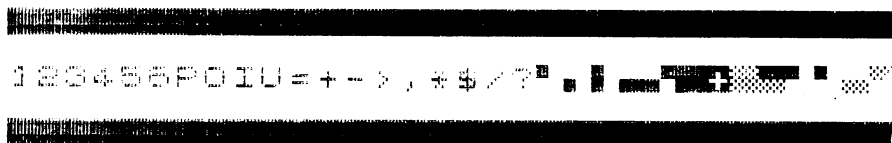


Figura 73



Figura 74

```
265 IF INKEY$="P" THEN PAUSE 200
```

Figura 75

```
265 IF INKEY$="P" THEN GOTO 280
270 GOTO 220
280 INPUT P$
290 GOTO 220
```

Figura 76

quali visto in due differenti momenti, comunque il modo migliore per rendersi conto del funzionamento è quello di provare il programma in pratica. Aggiungendo la linea 265 (fig. 75) si può rallentare la rotazione della frase di 4 secondi ad ogni carattere (PAUSE 200) se si tiene premuto il tasto "P".

```
170 PRINT AT 4,0;"
180 PRINT AT 6,0;"
```

Figura 77

```
2001 REM LINEE 204 >>> 209
2002 REM ROUTINE INVERSIONE
2003 REM CARATTERI DELLA FRASE
2004 DIM I$(1,LEN B$)
2005 FOR I=1 TO LEN B$
2006 IF CODE B$(I TO I) < 64 THEN
LET I$(1,I)=CHR$(CODE B$(I TO I)
)+128)
2007 IF CODE B$(I TO I) > 127 THEN
LET I$(1,I)=CHR$(CODE B$(I TO
I))
2008 NEXT I
2009 LET B$=I$(1)
```

Figura 78

A\$ = 1234567890POIUYTREWQASDFGHJKL.MNBUCXZ, > < * / ?) : = + - (\$ % & ' " \$ % & ' " \$ % & ' "

Figura 79

67890POIUYTREWQASDFGHJKL.MNBUCXZ

Figura 80

È anche possibile, inserendo le linee 265, 280 e 290 di fig. 76, fare in modo di bloccare la rotazione della frase ogni volta che si preme il tasto "P" e fino a che non si preme NEWLINE.

Un'altra possibilità ancora è quella di avere una frase in caratteri inversi anche inserendo dei caratteri normali, per ottenere ciò occorre modificare le linee 170 e 180 nel modo indicato in fig. 77 ed aggiungere al programma le nove linee di fig. 78, dopodichè inserendo, per esempio, la frase di fig. 79 si ottiene una scritta rotante come quella di fig. 80.

Un programma di Renumber è molto utile durante la programmazione poichè permette di riordinare automaticamente tutti i numeri di linea.

La versione di fig. 81 realizza questo riordino esclusi i numeri dopo GOTO e

```

000000 STOP
000100 REM RENUMBER
000200 FAST
000300 PRINT "INSERIRE N° LINEA INI
000400 RENUMBER"
000500 INPUT NIR
000600 CLS
000700 PRINT "INSERIRE NUOVO N° LIN
000800 INIZIALE"
000900 INPUT NLI
001000 CLS
001100 PRINT "INSERIRE DISTANZA TR
001200 LINEE"
001300 INPUT DTL
001400 CLS
001500 LET IM=10500
001600 LET NX=PEEK (IM+1)*256+PEEK
(CHIM+0)
001700 IF NX<NIR THEN GOTO 0110
001800 IF NX=0000 THEN GOTO 0150
001900 DOXA (CHIM+1),INT (NLI/256)
002000 DOXA (CHIM+0),NLI-256*INT (NL
H/256)
002100 LET NLI=NLI+DTL
002200 LET IM=IM+5
002300 IF PEEK IM=118 THEN GOTO 00
00
002400 LET IM=IM+1
002500 GOTO 0100
002600 LIST
    
```

Figura 81

INSERIRE NUOVO N° LINEA INIZIALE

Figura 82

INSERIRE N° LINEA INIZIO RENUMBER

Figura 83

INSERIRE DISTANZA TRA LINEE

Figura 84


```

1100  RRRR  PROGRAMMA PROVA
1110  RRRR  PER RENUMBER
1200  RRRR  LINEE 1---7777
1400  RRRR  - - - - -
1500  RRRR  R = 2
1700  RRRR  . . . . .
1800  RRRR  . . . . .
1900  RRRR  O 1007
2000  RRRR  . . . . .
2100  RRRR  . . . . .
2200  RRRR  . . . . .
2300  RRRR  U 4
2400  RRRR  . . . . .
2500  RRRR  . . . . .
2600  RRRR  . . . . .
2700  RRRR  . . . . .
2800  RRRR  . . . . .
2900  RRRR  . . . . .
3000  RRRR  X
3100  RRRR  FINE PROGRAMMA PROVA

```

Figura 87

GOSUB che vanno modificati singolarmente, in tal modo si ha un programma di Renumber più facile da inserire e togliere quando occorre.

Il programma va fatto girare con un RUN 9010 o con un GOTO 9010. Dopo di ciò si vedranno apparire sul video per prima la scritta di fig. 82 che chiede di inserire il numero da cui iniziare la rinumerazione (per es. 3), poi la scritta di fig. 83 che chiede di inserire il nuovo numero di linea iniziale (per es. 100) e infine la scritta di fig. 84 che chiede di inserire la distanza desiderata tra le linee (per es. 10).

La fig. 85 mostra come esempio un programma che ha proprio bisogno di essere riordinato, mentre le figure 86 e 87 mostrano due differenti esempi di rinumerazione del programma di fig. 85 ottenute con il programma di Renumber appena descritto.

Letture Memoria

FAST O SLOW

Un'altra utility interessante è il lettore della memoria del computer (fig. 88) con il quale si può leggere il contenuto di una parte qualsiasi della ROM o della RAM a partire dall'indirizzo introdotto all'inizio (fig. 89).

La fig. 90 mostra un esempio di lettura della RAM ed in particolare proprio di una zona nella quale è scritto il programma di fig. 88.

```

10 REM LETTORE MEMORIA
100 SLOW
110 PRINT "INSERIRE INDIRIZZO D
I PARTENZA"
120 PRINT "( 0--85535 ) E PREME
RE NEWLINE"
130 INPUT A
140 IF A<0 OR A>85535 THEN GOTO
150
160 CLS
170 PRINT " INDIRIZZO  CONTENUT
O  CARATTERE"
180 PRINT "-----"
190 FOR L=0 TO 21
200 PRINT AT L,0;A+L-2;TAB 15;P
EEK (A+L-2)
210 IF PEEK (A+L-2)<>118 THEN P
RINT AT L,04;CHR$ PEEK (A+L-2)
220 NEXT L
230 COPY
240 OUTPUT X$
250 GOTO 110

```

Figura 88

INSERIRE INDIRIZZO DI PARTENZA
(0--85535) E PREMERE NEWLINE

Figura 89

```

INDIRIZZO  CONTENUTO  CARATTERE
-----
10 000010 0000  REM
11 000011 0000
12 000012 0000
13 000013 0000
14 000014 0000
15 000015 0000
16 000016 0000
17 000017 0000
18 000018 0000
19 000019 0000
20 000020 0000
21 000021 0000
22 000022 0000
23 000023 0000
24 000024 0000
25 000025 0000
26 000026 0000
27 000027 0000
28 000028 0000
29 000029 0000
30 000030 0000
31 000031 0000
32 000032 0000
33 000033 0000
34 000034 0000
35 000035 0000
36 000036 0000
37 000037 0000
38 000038 0000
39 000039 0000
40 000040 0000
41 000041 0000
42 000042 0000
43 000043 0000
44 000044 0000
45 000045 0000
46 000046 0000
47 000047 0000
48 000048 0000
49 000049 0000
50 000050 0000
51 000051 0000
52 000052 0000
53 000053 0000
54 000054 0000
55 000055 0000
56 000056 0000
57 000057 0000
58 000058 0000
59 000059 0000
60 000060 0000
61 000061 0000
62 000062 0000
63 000063 0000
64 000064 0000
65 000065 0000
66 000066 0000
67 000067 0000
68 000068 0000
69 000069 0000
70 000070 0000
71 000071 0000
72 000072 0000
73 000073 0000
74 000074 0000
75 000075 0000
76 000076 0000
77 000077 0000
78 000078 0000
79 000079 0000
80 000080 0000
81 000081 0000
82 000082 0000
83 000083 0000
84 000084 0000
85 000085 0000
86 000086 0000
87 000087 0000
88 000088 0000
89 000089 0000
90 000090 0000
91 000091 0000
92 000092 0000
93 000093 0000
94 000094 0000
95 000095 0000
96 000096 0000
97 000097 0000
98 000098 0000
99 000099 0000
100 000100 0000

```

Figura 90

Il programma ha un'istruzione (220) che stampa il pezzo di memoria appena esaminato, dopodichè premendo NEWLINE si può introdurre un nuovo indirizzo di partenza.

**Programma caricamento routines
in linguaggio macchina
contenute in stringa**

FAST

Se non si hanno limitazioni di memoria il modo più chiaro e più comodo per inserire in un programma delle routines in linguaggio macchina è quello di caricare le varie istruzioni decimali (con uno spazio tra l'una e l'altra) in una stringa (per es. A\$) e di utilizzare il programma di fig. 91 (istruzioni da 2000 a 2090), programma utilizzato in pratica anche in due delle animazioni del prossimo capitolo. Le linee da 1 a 80 in fig. 91 servono per mostrare il funzionamento della routine in linguaggio macchina contenuta in A\$ (linea 2020).

```

1 REM PROGRAMMA DI PROVA
10 GOSUB 2000
20 PRINT "INSERIRE N ( 1--255
) "
30 INPUT N
40 CLR
45 POKE 16507,N
50 PRINT USR 20205
60 FAST
70 STOP
10000 REM ROUTINE CARICAMENTO
10001 REM PROGRAMMI IN
10002 REM LINGUAGGIO MACCHINA
10003 REM A PARTIRE DA
10004 REM ISTRUZIONI DECIMALI
20000 REM
20005 LET I$="040 100 064 041 041
20010 GY7 20205
20020 LET A$=20205
20030 FOR L=1 TO LEN A$ STEP 4
20040 POKE 31,VAL A$(L TO L+3)
20050 LET A=A+1
20060 NEXT L
20070 STOP
20080 RETURN

```

Figura 91

```

00005 040 44 LD HL,(16507)
00010 120 44
00015 054 44
00020 041 44 ADD HL,HL
00025 041 44 ADD HL,HL
00030 020 44 LD B,H
00035 077 44 LD C,L
00040 001 44 RETURN

```

Figura 92

```
INSERIRE N ( 1--255 )
```

Figura 93

```

00010 REM MODIFICA --1--
00020 LET A#="0401200640410410600
00030 1:"
00040 LET A=00005
00050 FOR L=1 TO LEN A# STEP 3
00060 POKE A,VAL A$(L TO L+2)
00070 LET A=A+1
00080 NEXT L

```

Figura 94

```

00010 REM MODIFICA --2--
00020 INPUT A#
00030 LET A=00225
00040 FOR L=1 TO LEN A# STEP 3

```

Figura 95

Tale routine di esempio (fig. 92) serve solo a moltiplicare per quattro un numero inserito all'inizio del programma (fig. 93) e a presentare il risultato sul video. Modificando alcune linee del programma nel modo indicato in fig. 94, si possono eliminare gli spazi tra le istruzioni contenute nella stringa (linea 2020) ogni istruzione, però, deve essere sempre formata da tre numeri anche quando è inferiore a 100 (per es. 2 va scritto 002).

È anche possibile sostituire la stringa con le istruzioni della linea 2020 con un INPUT A\$ (fig. 95) in modo da potere inserire ogni volta delle routines da provare. Oltre che in decimali è possibile caricare le istruzioni nella stringa anche in esadecimale utilizzando il programma di fig. 96 e le istruzioni esadecimali di esempio contenute in A\$ sono sempre quelle di fig. 92. Un'ulteriore compattazione della stringa può essere ottenuta eliminando gli spazi tra un'istruzione e l'altra, utilizzando l'ultima modifica esposta in fig. 97.

Per convertire delle istruzioni decimali in esadecimali, e viceversa, si possono usare gli appositi programmi per conversioni di numeri in base diversa che sono presentati più avanti in questo stesso capitolo.

```

1000 REM ROUTINE CARICAMENTO
1001 REM PROGRAMMI IN
1002 REM LINGUAGGIO MACCHINA
1003 REM A PARTIRE DA
1004 REM ISTRUZIONI ESADECIMALI
1005 REM MEMORIZZATE IN A$
2000 FAST
2020 LET A$="2A 7B 40 29 29 44 4
2030"
2030 LET A=20225
2040 FOR L=1 TO LEN A$ STEP 3
2050 POKE A,(CODE A$(L TO )-28)*
2060+(CODE A$(L+1 TO )-28)
2070 LET A=A+1
2080 NEXT L

```

Figura 96

```

2015 REM MODIFICA --3--
2020 LET A$="2A7B402929444DC9"
2030 LET A=20225
2040 FOR L=1 TO LEN A$ STEP 2

```

Figura 97

Programma caricamento routines in linguaggio macchina contenute in 1 REM

FAST

Un sistema già noto per inserire delle istruzioni in linguaggio macchina in un programma è quello di scrivere tali istruzioni (in esadecimale) nell'istruzione 1 REM posta all'inizio del programma.

Rispetto a programmi simili già esistenti la versione di fig. 98 (linee da 2000 a 2090 più linee 1 e 2) permette di inserire nella istruzione 1 REM un numero qualsiasi di

```

1 REM 2A7B402929444DC9
0000 GOSUB 2000
) " 00 PRINT "INSERIRE N ( 1--255
00 INPUT N
0000 CCLS
0040 POKB 16507,N
0000 PRINT USR 00205
0000 TRIST
0000 STOP
10000 REM ROUTINE CARICAMENTO
10001 REM PROGRAMMI IN
10002 REM LINGUAGGIO MACCHINA
10003 REM A PARTIRE DA
10004 REM ISTRUZIONI ESADECIMALI
10005 REM MEMORIZZATE IN <1 REM>
20000 TRIST
20001 LET S = 16511+PEEK 16511+256*
20002 LET L = S
20003 LET D = 200205
20004 POKB L,16514 TO S STEP 2
20005 POKB D,((PEEK L)-28)*16+((P
20006 (L+1))-28)
20007 LET D = D+1
20008 NEXT L
20009 GOTO 20000
20010 RETURN

```

Figura 98

istruzioni esadecimali senza modificare il programma il quale controlla automaticamente la lunghezza del contenuto di 1 REM con l'istruzione contenuta nella linea 2020.

Se la routine in linguaggio macchina è più lunga di 200-220 Bytes o se si usa la RAM da 16K l'istruzione 2030 va modificata in conseguenza.

Conversioni numeriche

I prossimi sei programmi eseguono le conversioni tra i tre principali tipi di numerazione utilizzati dai programmatori, le numerazioni binaria, decimale e esadecimale. Di ciascun programma viene dato anche un esempio pratico.

Le applicazioni dei programmi di conversione possono essere varie, per esempio la conversione da decimale a esadecimale può essere utile per convertire una serie di

istruzioni decimali in linguaggio macchina nei corrispondenti valori esadecimali da inserire in un programma.

Tenete presente che la matematica del computer ha una precisione di 8-9 Digit per cui se il numero binario, decimale o esadecimale inserito è troppo grande il numero ottenuto dalla conversione può risultare arrotondato.

In tutti e sei i programmi i numeri da convertire vanno inseriti dal Bit più significativo (MSB) al Bit meno significativo (LSB).

Tutti i programmi hanno un'istruzione finale per la stampa del risultato.

**Conversione da binario
a decimale**

FAST O SLOW

Programma in fig. 99. Esempio in fig. 100

```

10 INPUT A$
1000 REM CONVERSIONE DA
1010 REM BINARIO A DECIMALE
1020 FAST
1030 LET N=0
1040 LET K=0
1050 FOR C=LEN A$-1 TO 0 STEP -1
1060 LET K=K+1
1070 LET N=N+2**C*VAL A$(K TO K)
1080 NEXT C
1090 CLS
1100 PRINT "N° BINARIO:"
1110 PRINT A$
1120 PRINT
1130 PRINT "= DECIMALE ";N
1140 COPY

```

Figura 99

```

N° BINARIO:
1100011011011101
= DECIMALE 50909

```

Figura 100

**Conversione da binario
a esadecimale**

FAST O SLOW

Programma in fig. 101. Esempio in fig. 102

```

1510 INPUT A$
1520 REM CONVERSIONE DA
1530 REM BINARIO A ESADECIMALE
1540 FAST
1550 LET S=0
1560 LET L=LEN A$/4-INT (LEN A$/
4)
1570 IF L=0 THEN GOTO 1620
1580 LET S=1
1590 IF L<0.25 THEN LET S=S+2
1600 IF L>0.25 AND L<0.51 THEN L
ET S=S+1
1610 DIM S$(1,S)
1620 LET S$(1)="000"
1630 LET A$=S$(1)+A$
1640 LET H=LEN A$/4
1650 DIM N(H)
1660 FOR P=1 TO H
1670 LET N(P)=0
1680 FOR C=0 TO 3
1690 LET N(P)=N(P)+2**C*VAL A$(P
+4-1)
1700 NEXT C
1710 NEXT P
1720 CLS
1730 PRINT "N° BINARIO:"
1740 IF S=0 THEN PRINT A$
1750 IF S>0 THEN PRINT A$(S+1 TO
)
1760 PRINT
1770 PRINT "= ESADECIMALE      ";
1780 FOR P=1 TO H
1790 PRINT CHR$(N(P)+28);
1800 NEXT P
1810 PRINT
1820 COPY

```

Figura 101

```

N° BINARIO:
1100011011011101

= ESADECIMALE      C6DD

```

Figura 102

Conversione da esadecimale
a decimale

FAST O SLOW

Programma in fig. 103. Esempio in fig. 104

```

000010 INPUT A$
000020 REM CONVERSIONE DA
000030 REM ESADECIMALE A DECIMALE
000040 FAST
000050 LET Z=0
000060 FOR J=0 TO LEN A$-1
000070 LET Z=Z+((CODE A$(LEN A$-J
000080 )-20)*10**J)**J)
000090 NEXT J
000100 PRINT "N° ESADECIMALE: "
000110 PRINT
000120 PRINT "= DECIMALE ";Z
000130 COPY

```

Figura 103

```

N° ESADECIMALE: C6DD
= DECIMALE 50909

```

Figura 104

Conversione da esadecimale
a binario

FAST O SLOW

Programma in fig. 105. Esempio in fig. 106

```

000010 INPUT A$
000020 REM CONVERSIONE DA
000030 REM ESADECIMALE A BINARIO
000040 FAST
000050 DIM N$(1,LEN A$*4)
000060 LET B$="0000 0001 0010 0011
000070 0100 0101 0110 0111 1000 1001 1
000080 010 1011 1100 1101 1110 1111"
000090 FOR E=1 TO LEN A$
000100 LET C=CODE A$(E TO )-27
000110 LET N$(1,E*4-3 TO E*4)=B$(C
000120 *55-4 TO C*55-1)
000130 NEXT E
000140 FOR Z=1 TO LEN N$(1)
000150 LET R$=N$(1,Z TO )

```

Figura 105 (Continua)

```

00010 IF N$(1,Z)="1" THEN GOTO 26
00020 NEXT Z
00030 PRINT "N" ESADDECIMALE:
00040 PRINT
00050 PRINT "= BINARIO ( MSB >>>
00060 PRINT R$
00070 COPY

```

Figura 105 (Fine)

```

N ESADDECIMALE: 06DD
= BINARIO ( MSB >>> LSB )
1100011011011101

```

Figura 106

Conversione da decimale
a esaddecimale

FAST O SLOW

Programma in fig. 107. Esempio in fig. 108

```

00010 INPUT N
00020 REM CONVERSIONE DA
00030 REM DECIMALE A ESADDECIMALE
00040 PRINT
00050 LET E=0
00060 IF 15*16**E>N THEN GOTO 307
00070 LET E=E+1
00080 GOTO 03040
00090 LET H#=N/16
00100 LET U=N
00110 FOR Q=E TO 0 STEP -1
00120 FOR F=15 TO 0 STEP -1
00130 LET X=U/16**Q
00140 IF J-K>=0 THEN LET H# = H# + CH
00150 H# = (H# + F)
00160 IF U - X >= 0 THEN GOTO 0150
00170 NEXT F

```

Figura 107 (Continua)


```

0100 LET J=J-K
01100 NEXT C
01170 IF LEN H# > 1 AND CODE H# = 28
01180 LET H# = H# (2 TO 1)
012000 CLR
012100 PRINT "N# DECIMALE: "; N
012200 PRINT
012310 PRINT " = ESADECIMALE "; H
01#
01240 COPY

```

Figura 107 (Fine)

```

N# DECIMALE:      50909
= ESADECIMALE    C6DD

```

Figura 108

Conversione da decimale a binario

FAST O SLOW

Programma in fig. 109. Esempio in fig. 110

```

010 INPUT N
01500 REM CONVERSIONE DA
01510 REM DECIMALE A BINARIO
01520 FAST
01530 LET E=0
01540 IF N#*E > N THEN GOTO 01570
01550 LET E=E+1
015600 GOTO 01540
015700 LET B$=""
015800 LET LEN=0
015900 FOR C=LEN-1 TO 0 STEP -1
015910 LET K=C+1
016000 IF J-K >= 0 THEN LET B$=B$+"1"
016100 IF J-K < 0 THEN LET B$=B$+"0"
016200 IF J-K >= 0 THEN LET J=J-K
016300 NEXT C
016400 CLR
016500 PRINT "N# DECIMALE: "; N
016600 PRINT
016700 PRINT " = BINARIO. ( MSB >>>
016800 PRINT B$
016900 COPY

```

Figura 109

N° DECIMALE: 50909

= BINARIO (MSB >>> LSB)
1100011011011101

Figura 110

**Programma di riunione delle
routines di conversione numerica**

SLOW

I sei programmi di conversione numerica appena descritti possono essere riuniti in una sorta di calcolatrice per Programmatori togliendo da tutti e sei i programmi le linee 10 ed aggiungendoli al programma di fig. 111.

```
10 ROM CALCOLATRICE
20 ROM PER PROGRAMMATORI
30 PROM
40 PRINT "PREMERE PER INSE
50 PRINT "-----
60 PRINT TAB 5;"0" BINARI
70 PRINT TAB 5;"D" DECIMA
80 PRINT TAB 5;"E" ESADEC
90 IF INKEY#="B" THEN GOTO 200
100 IF INKEY#="O" THEN GOTO 300
110 IF INKEY#="E" THEN GOTO 500
120 GOTO 100
130 GOSUB 1000
140 PRINT "INSERIRE N° IN BINAR
150 PRINT "( MSB >>> LSB )E PRE
160 INPUT N$
170 FOR A=1 TO LEN A$
180 IF CODE A$(A TO )>99 OR COD
190 NEXT A
200 PRINT "PREMERE: PER CONU
210 PRINT "-----
```

Figura 111 (Continua)

```

000 PRINT TAB 5;"D"          DECIMAL
M 010 PRINT TAB 5;"E"          ESADECI
D 020 IF INKEY$="D" THEN GOTO 100
0 030 IF INKEY$="E" THEN GOTO 150
0 040 GOTO 320
0050 OLS
0060 PRINT "INSERIRE N° DECIMALE
0070 PRINT "E PREMERE NEWLINE"
0080 INPUT A$
0090 FOR A=1 TO LEN A$
E 400 IF CODE A$(A TO )>37 OR COD
A A$(A TO )<28 THEN GOTO 380
410 NEXT A
415 LET N=VAL A$
4000 OLS
E 430 PRINT "PREMERE: PER CONU
RTIRE IN"
440 PRINT "-----"
00450 PRINT TAB 5;"B"          BINARIO
00460 PRINT TAB 5;"E"          ESADECI
M 470 IF INKEY$="B" THEN GOTO 350
0 480 IF INKEY$="E" THEN GOTO 300
0 490 GOTO 470
0500 OLS
D 0510 PRINT "INSERIRE N° ESADECIM
D 0520 PRINT "( 0 >> F ) E PREMERE
NEWLINE"
0530 INPUT A$
0540 FOR A=1 TO LEN A$
E 0550 IF CODE A$(A TO )>43 OR COD
A A$(A TO )<28 THEN GOTO 530
0560 NEXT A
0570 OLS
E 0580 PRINT "PREMERE: PER CONU
RTIRE IN"
0590 PRINT "-----"

```

Figura 111 (Continua)

```

000 PRINT TAB 5;"B"          BINARIO
010 PRINT TAB 5;"D"          DECIMALE
020 IF INKEY#="B" THEN GOTO 250
030 IF INKEY#="D" THEN GOTO 200
040 GOTO 000
050 PRINT AT 10,0;"PREMERE NEWLINE"
060 PRINT AT 11,0;"UN ALTRA CONVERSIONE"
070 INPUT X$
080 CLS
090 GOTO 100
100 GOTO 000
110 GOTO 000
120 GOTO 000
130 GOTO 000
140 GOTO 000
150 GOTO 000
160 GOTO 000
170 GOTO 000

```

Figura 111 (Fine)

```

PREMERE:          PER INSERIRE N°
-----
B                BINARIO
D                DECIMALE
E                ESADECIMALE

```

Figura 112

```

INSERIRE N° IN BINARIO ( 1/0 )
( MSB >>> LSB ) E PREMERE NEWLINE

```

Figura 113

```

INSERIRE N° DECIMALE
E PREMERE NEWLINE

```

Figura 114

```

INSERIRE N° ESADECIMALE
( 0 >>> F ) E PREMERE NEWLINE

```

Figura 115

Dopo di ciò, dando il RUN, il programma chiede all'inizio che tipo di numero si vuole convertire (fig. 112).

A seconda del tasto premuto uscirà sul video una delle tre richieste di inserimento di un numero visibile nelle figure 113, 114 e 115 a cui seguirà una delle corrispondenti tabelle delle figure 116, 117 e 118 che chiede in quale codice deve essere convertito il numero inserito.

```
PREMERE :      PER CONVERTIRE IN
-----
      D      DECIMALE
      E      ESADECIMALE
```

Figura 116

```
PREMERE :      PER CONVERTIRE IN
-----
      B      BINARIO
      E      ESADECIMALE
```

Figura 117

```
PREMERE :      PER CONVERTIRE IN
-----
      B      BINARIO
      D      DECIMALE
```

Figura 118

```
PREMERE NEWLINE PER
UN'ALTRA CONVERSIONE
```

Figura 119

A seconda della conversione effettuata il risultato sul video sarà uno di quelli già visti come esempio nei sei programmi precedenti (figure 100, 102, 104, 106, 108, 110) insieme alla scritta di fig. 119 che chiede se si vuole effettuare un'altra conversione.

Nelle conversioni da decimale a binario e da decimale a esadecimale le cifre 67.092.480 e 67.108.864 sono rispettivamente i due numeri massimi da inserire oltre i quali il programma può sbagliare o arrotondare.

ANIMAZIONI

Countdown e lancio missile

SLOW

Il primo programma di animazione (fig. 120) disegna sullo schermo un cielo notturno con una rampa di lancio ed un missile pronto a partire (fig. 121). Terminato il conteggio alla rovescia che dura 10 secondi il missile parte (fig. 122) per poi sparire nella parte alta dello schermo.

Per ottenere un movimento rapido del missile e per non vedere il disegno dello stesso formarsi ad ogni spostamento è stato necessario utilizzare una breve routine in linguaggio macchina contenuta nella stringa A\$ (linea 130) e caricata nella RAM all'inizio del programma. La routine viene descritta più dettagliatamente qui di seguito:

| Indirizzo RAM | Codice decimale | Mnemonico |
|------------------|--------------------|-----------------------------|
| 20224 | 001 | LD BC, nn |
| 20225 | 000 | } Numero caricato dal BASIC |
| 20226 | 000 | |
| 20227 | 017 | |
| 20228 | 033 | LD DE, 33 |
| 20229 | 000 | } LD HL, (16396) |
| 20330 | 042 | |
| 20331 | 012 | |
| 20332 | 064 | } ADD, HL, BC |
| 20333 | 009 | |
| 20334 | 054 | } LD (HL), 5 |
| 20335 | 005 | |
| 20336 | 025 | ADD HL, DE |
| 20337 | 054 | } LD (HL), 5 |
| 20338 | 005 | |
| 20339 | 025 | ADD HL, DE |
| 20340 | 054 | } LD (HL), 6 |
| 20341 | 006 | |
| 20342 | 035 | INC HL |
| 20343 | 054 | } LD (HL), 129 |
| 20344 | 129 | |
| 20345 | 043 | DEC HL |
| 20346 | 025 | ADD HL, DE |
| 20347 | 054 | } LD (HL), 128 |
| 20348 | 128 | |
| 20349 | 035 | INC HL |
| 20350 | 054 | } LD (HL), 128 |
| 20351 | 128 | |
| 20352 | 201 | RETURN |


```

450 POKE 20226,INT (H/256)
470 LET Z=USR 20224
480 NEXT H
490 PRINT AT 0,18:" "
500 PRINT AT 1,18:" "
510 PRINT AT 2,18:" "

```

Figura 120 (Fine)

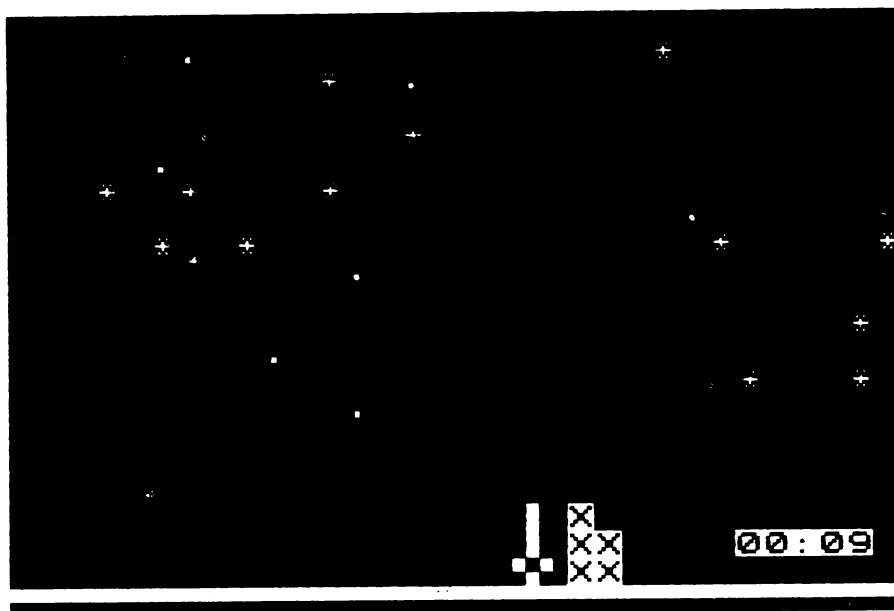


Figura 121

CaLeIdOsCoPiO

SLOW

In un caleidoscopio diviso in 4 zone un oggetto visibile in una zona deve apparire immediatamente e simmetricamente nelle altre tre.

Per questo scopo la grafica del BASIC non è sufficientemente veloce, per cui anche in questo secondo programma di animazione (fig. 123) la maggiore velocità nel disegnare sullo schermo è ottenuta con la routine in linguaggio macchina contenuta nella stringa A\$ (linea 300).

La fig. 124 riporta un esempio di disegno prodotto dal programma, disegno che si modifica continuamente e velocemente sullo schermo.

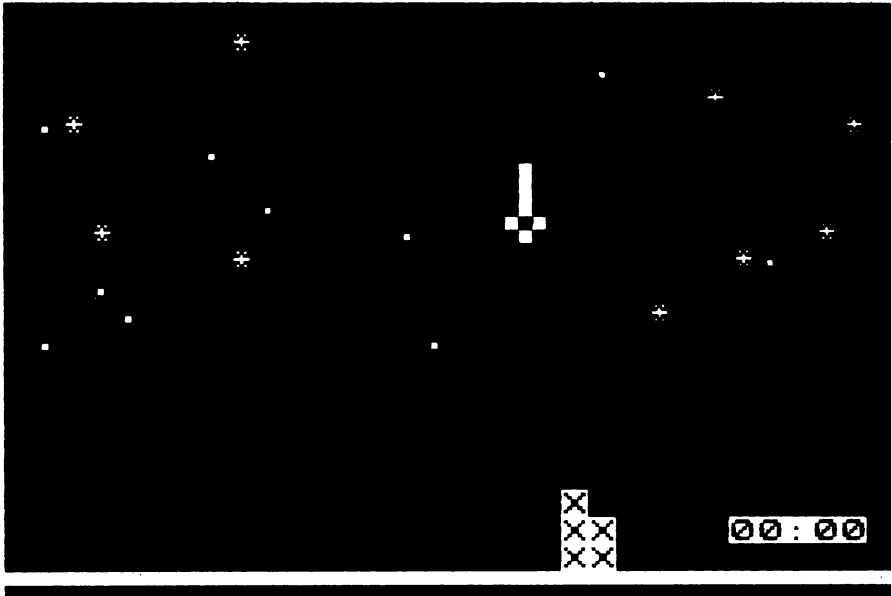


Figura 122

Così com'è il programma disegna sullo schermo in media un terzo di caratteri normali, un terzo di caratteri inversi ed un terzo di spazi bianchi, modificando le linee 180, 190, 200 come in fig. 125, il programma disegnerà sullo schermo solo caratteri grafici, diretti e inversi e spazi bianchi (fig. 126), che cambiano di continuo.

Il programma può essere fermato premendo il tasto S.

Qui di seguito viene esposta più in dettaglio la routine in linguaggio macchina utilizzata dal programma:

| Indirizzo RAM | Codice decimale | Mnemonico |
|------------------|--------------------|--------------------|
| 20230 | 022 | } — LD D, n (n=X) |
| 20231 | 000 | |
| 20232 | 042 | |
| 20233 | 012 | } — LD HL, (16396) |
| 20234 | 064 | |
| 20235 | 237 | } — LD BC, (20225) |
| 20236 | 075 | |
| 20237 | 001 | |
| 20238 | 079 | } — ADD HL, BC |
| 20239 | 009 | |
| 20240 | 237 | } — LD BC, (20227) |
| 20241 | 075 | |
| 20242 | 003 | |
| 20243 | 079 | |

| Indirizzo RAM | Codice decimale | Mnemonico |
|------------------|--------------------|--------------------|
| 20244 | 237 | } — SBC HL, BC |
| 20245 | 066 | |
| 20246 | 205 | } — CALL 20297 |
| 20247 | 073 | |
| 20248 | 079 | |
| 20249 | 237 | } — LD BC, (20227) |
| 20250 | 075 | |
| 20251 | 003 | |
| 20252 | 079 | } — ADD HL, BC |
| 20253 | 009 | |
| 20254 | 009 | — ADD HL, BC |
| 20255 | 205 | } — CALL 20297 |
| 20256 | 073 | |
| 20257 | 079 | |
| 20258 | 033 | } — LD HL, 692 |
| 20259 | 180 | |
| 20260 | 002 | |
| 20261 | 237 | } — LD BC, (20225) |
| 20262 | 075 | |
| 20263 | 001 | |
| 20264 | 079 | } — SBC HL, BC |
| 20265 | 237 | |
| 20266 | 066 | } — LD (20225), HL |
| 20267 | 034 | |
| 20268 | 001 | |
| 20269 | 079 | } — LD HL, (16396) |
| 20270 | 042 | |
| 20271 | 012 | |
| 20272 | 064 | } — LD BC, (20225) |
| 20273 | 237 | |
| 20274 | 075 | |
| 20275 | 001 | } — ADD HL, BC |
| 20276 | 079 | |
| 20277 | 009 | } — LD BC, (20227) |
| 20278 | 237 | |
| 20279 | 075 | |
| 20280 | 003 | } — SBC HL, BC |
| 20281 | 079 | |
| 20282 | 237 | |
| 20283 | 066 | } — CALL 20297 |
| 20284 | 205 | |
| 20285 | 073 | |
| 20286 | 079 | } — LD BC (20227) |
| 20287 | 237 | |
| 20288 | 075 | |
| 20289 | 003 | } — ADD HL, BC |
| 20290 | 079 | |
| 20291 | 009 | — ADD HL, BC |
| 20292 | 009 | — ADD HL, BC |

| Indirizzo RAM | Codice decimale | Mnemonico |
|---------------|-----------------|-----------|
|---------------|-----------------|-----------|

| | | | |
|------------|-------|----------------|----------------|
| | 20293 | 205 | } CALL 20297 |
| | 20294 | 073 | |
| | 20295 | 079 | |
| | 20296 | 201 | — RETURN |
| subroutine | 20297 | 114 | — LD (HL), D |
| | 20298 | 035 | — INC HL |
| | 20299 | 114 | — LD (HL), D |
| | 20300 | 001 | } — LD BC, 32 |
| | 20301 | 032 | |
| | 20302 | 000 | } — ADD HL, BC |
| | 20303 | 009 | |
| | 20304 | 114 | — LD (HL), D |
| | 20305 | 035 | — INC HL |
| | 20306 | 114 | — LD (HL), D |
| | 20307 | 001 | } — LD BC, 34 |
| | 20308 | 034 | |
| 20309 | 000 | } — SBC HL, BC | |
| 20310 | 237 | | |
| 20311 | 066 | } — RETURN | |
| 20312 | 201 | | |

```

100 REM  CALENDOSCOPIO
200 REM  -----
1000 GOSUB 270
1100 SLOW
1200 RAND
1300 LET T=49+66*INT (5*RND)
1400 LET P=1+2*INT (5*RND)
1500 POKE 20225,T-(INT (T/256))*
256
1600 POKE 20226,INT (T/256)
1700 POKE 20227,P
1800 LET X=INT (192*RND)
1900 IF X>83 THEN LET X=X+64
2000 IF X>191 THEN LET X=0
2100 POKE 20231,X
2200 LET M=USR 20230
2300 FOR R=1 TO 1
2400 NEXT R
2500 IF INKEY#="S" THEN GOTO 370
2600 GOTO 130
2700 PRINT
2800 POKE 20228,0
2900 LET H#="000 000 040 010 064
3000 075 001 070 009 037 070 000

```

Figura 123 (Continua)

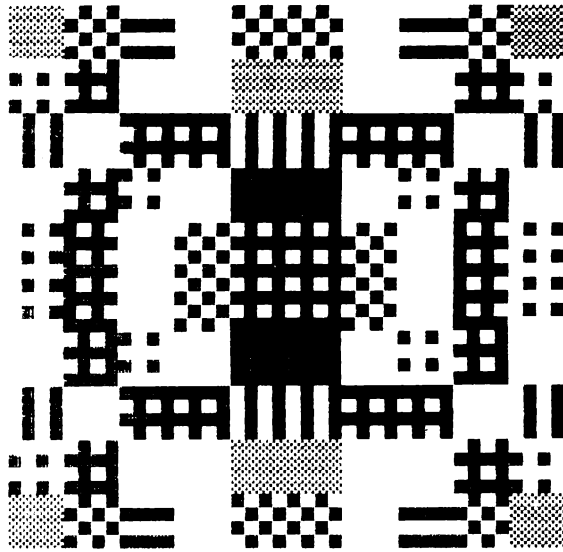


Figura 126

RoMbOsPiRaLe

SLOW

Anche senza ricorrere a delle apposite routines in linguaggio macchina, si possono ottenere delle buone animazioni. Una di queste è ottenuta dal programma Rombo-spirale (fig. 127).

```

 10  REM  ROMBOSPIRALE
100  SLOW
110  LET  C=1
120  GOSUB 370
130  FOR  P=1 TO 11
140  PRINT AT 12-P, P+S; CHR$ X
150  NEXT  P
160  FOR  N=1 TO 11 STEP 2
170  GOSUB 370
180  FOR  P=1 TO 12-N
190  PRINT AT -1+P+N, 15+P; CHR$ X
200  NEXT  P
210  GOSUB 370
220  FOR  P=1 TO 12-N
230  PRINT AT 10+P, 28-P-N; CHR$ X

```

Figura 127 (Continua)

```

00000040 ZNEXT P
00000000 GOST U P
00000000 PPRINT TO = 0 70
00000000 PPRINT TO AT TO 00 - P - N , 17 - P ; CHR$ X
00000000 ZNEXT P
00000000 GOST U P
00000000 PPRINT TO = 0 70
00000000 PPRINT TO AT TO 10 - P , 0 + P + N ; CHR$ X
00000000 ZNEXT P
00000000 ZNEXT P
00000000 LET C = C * (- 1)
00000000 GOTO X = 0
00000000 GOTO X = 100
00000000 IF C = 1 THEN LET X = 1 + INT (12
7 + INT (
00000000 IF X > 60 THEN LET X = X + 64
00000000 RETURN

```

Figura 127 (Fine)

Tale programma disegna e cancella alternativamente in continuazione, una spirale a forma di rombo formata da caratteri scelti casualmente.

La spirale viene disegnata o cancellata partendo dall'esterno verso l'interno. La spirale appena completata appare, per esempio, come in fig. 128.

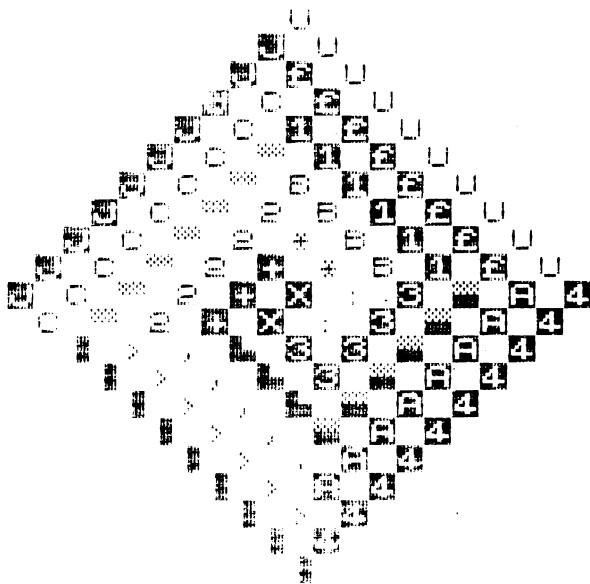


Figura 128

Un'altra animazione unicamente in BASIC è l'esplosione di un oggetto colpito da un proiettile creato dal programma di fig. 129.

```

10  REM ANIMAZIONE ESPLOSIONE
100  GOTO 110
110  PRINT
120  PRINT
130  PRINT
140  PRINT
150  PRINT
160  PRINT
170  PRINT
180  PRINT
190  PRINT
200  PRINT
210  PRINT
220  PRINT
230  PRINT
240  PRINT
250  PRINT
260  PRINT
270  PRINT
280  PRINT
290  PRINT
300  PRINT
310  PRINT
320  PRINT
330  PRINT
340  PRINT
350  PRINT
360  PRINT
370  PRINT
380  PRINT
390  PRINT
400  PRINT
410  PRINT
420  PRINT
430  PRINT
440  PRINT
450  PRINT
460  PRINT
470  PRINT
480  PRINT
490  PRINT
500  PRINT
510  PRINT
520  PRINT
530  PRINT
540  PRINT
550  PRINT
560  PRINT
570  PRINT
580  PRINT
590  PRINT
600  PRINT
610  PRINT
620  PRINT
630  PRINT
640  PRINT
650  PRINT
660  PRINT
670  PRINT
680  PRINT
690  PRINT
700  PRINT
710  PRINT
720  PRINT
730  PRINT
740  PRINT
750  PRINT
760  PRINT
770  PRINT
780  PRINT
790  PRINT
800  PRINT
810  PRINT
820  PRINT
830  PRINT
840  PRINT
850  PRINT
860  PRINT
870  PRINT
880  PRINT
890  PRINT
900  PRINT
910  PRINT
920  PRINT
930  PRINT
940  PRINT
950  PRINT
960  PRINT
970  PRINT
980  PRINT
990  PRINT
1000 PRINT

```

Figura 129

Le figure da 130 a 136 mostrano in sequenza alcune fasi dell'esplosione. Il programma ha anche quattro istruzioni (135, 195, 295, 385) per avere anche dei suoni una

volta che sarà collegata la scheda musicale che è presentata in dettaglio nel capitolo sulla musica.



Figura 130



Figura 131



Figura 132

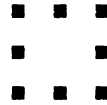


Figura 133

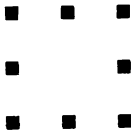


Figura 134

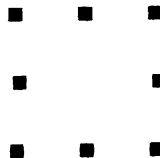


Figura 135

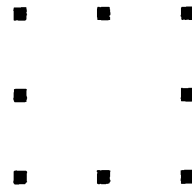


Figura 136

Programmazione universale animazione con PRINT AT

SLOW

Una complessa animazione può richiedere decine di linee di programma con PRINT AT.

Come già visto nel capitolo sulla Grafica, una lunga serie di linee con PRINT AT può essere condensata in un programma più breve.

Il programma universale di animazione di fig. 137 è in gran parte uguale al programma universale per PRINT AT di fig. 26 con l'aggiunta, però di alcune linee per aggiungere la funzione del Tempo.

Anche nel programma di fig. 137 i codici per le coordinate delle linee (in L\$) e delle colonne (in C\$) sono quelli elencati nella tabella 1. I caratteri sono sempre nella stringa X\$ mentre i Tempi tra l'esecuzione di un PRINT AT e il successivo sono contenuti nella stringa T\$ e codificati secondo la tabella 2; tali tempi sono in una scala arbitraria, vale a dire £ non corrisponde ad un secondo ma solo al tempo più breve, e allo stesso modo X rappresenta solo il tempo più lungo.

Se si ha bisogno di un disegno complesso insieme ad una lunga animazione si può

senso opposto. La linea 160 può essere modificata come indicato in fig. 138 (con 29 spazi inversi alternati a 29 spazi normali) per fare in modo che l'oggetto in movimento abbia sempre la stessa forma.

Programma universale animazione con PLOT-UNPLOT

SLOW

Se si vuole una risoluzione quattro volte maggiore e se non è necessario inserire dei caratteri, si può realizzare un'animazione con PLOT e UNPLOT utilizzando il programma di fig. 139.

Tutte le istruzioni di animazione sono inserite in un'unica stringa (C\$) ed ogni istruzione è costituita da sei caratteri che formano due numeri da due caratteri separati da un segno + oppure - e seguiti da uno spazio che separa un'istruzione dall'altra. Il primo ed il secondo numero di ogni istruzione rappresentano rispettivamente le coordinate X e Y del pixel da plottare (se tra i numeri c'è un +) o da unplottare (se tra i numeri c'è un -).

```

100 REM PROGRAMMA UNIVERSALE
200 REM ANIMAZIONE CON PLOT
300 REM UNPLOT E IN SLOW
1100 REM COORDINATE IN C$ :
1200 REM 1° VALORE - COORD. <X>
1300 REM 2° VALORE - COORD. <Y>
1400 REM AL CENTRO SCELTA MODO
1500 REM + PLOT - UNPLOT
1600 LET C$ = "+00+00 07+02 08+02 1
00+00 10+00 02 14+02 16+02 18+02 19+
00+00 10-00-00 00+01 00-00-00 00+00 00+00
100-00 10+00 10-00 11+00 11-00 1
1+00 11+01 11-01 12+01 12-01 13+
01 13-01 13+02 13+03 13-03 14+03
14-03 15+03 15-03 15+02 15+01 1
05 01 16+01 16-01 17+01 17-01 17+
00 17+00 17-00 17+04
1700 FOR P=1 TO LEN C$ STEP 6
1800 IF CODE C$(P+5) TO )=21 THEN
PLOT VAL C$(P TO P+1),VAL C$(P+
0 TO P+4)
1900 IF CODE C$(P+2) TO )=22 THEN
UNPLOT VAL C$(P TO P+1),VAL C$(
P+3 TO P+4)
200 NEXT P

```

Figura 139

Anche in questo caso la stringa con le istruzioni contiene un esempio di animazione che non ha un'utilità pratica ma che mostra il funzionamento del programma.



Figura 140



Figura 141



Figura 142

L'animazione (di cui nelle figure 140, 141, 142 si può vedere l'inizio, una fase intermedia e la fine) rappresenta una sorta di muro con dei fori che vengono attraversati e riempiti da un pixel in movimento.

PROGRAMMI PER INTERFACCIA

Circuito di interfaccia per ZX80-ZX81

Con solo due integrati (due comuni TTL-LS) e un costo totale inferiore a 5000 lire si può realizzare un circuito di Interfaccia ad 8 uscite che permette di collegare uno ZX81 o un ZX80 al mondo esterno consentendo di realizzare numerosi programmi di automazione, di controllo, di robotica, ecc. altrimenti non possibili con il solo computer.

Oltre a ciò il circuito di Interfaccia è indispensabile per pilotare il circuito musicale esposto nel prossimo capitolo, così come altri circuiti realizzati da ciascun possessore di un ZX.

IMPORTANTE - L'interfaccia va collegata tramite dei fili direttamente al circuito stampato dello ZX.

TENERE PRESENTE CHE UNA ERRATA COSTRUZIONE DEL CIRCUITO O UN ERRATO COLLEGAMENTO DEI FILI AL CIRCUITO STAMPATO PUO' CAUSARE DANNI AL COMPUTER, per cui se vi occupate di programmazione ma non di circuiti elettronici, è preferibile che facciate realizzare o controllare il tutto ad un Tecnico che abbia già esperienza di montaggi e saldature.

Lo schema del circuito di Interfaccia è in fig. 143 ed usa, come già detto, due integrati: un 74LS373 che contiene 8 memorie LATCH ed un 74LS27 che contiene tre NOR a tre ingressi utilizzati per la decodifica dell'indirizzo di memoria.

Le configurazioni dei due integrati sono in fig. 144 e 145.

I quindici ingressi del circuito di Interfaccia vanno collegati, tramite quindici spezzoni di filo sottile ad un capo, a quindici dei quaranta piedini dello Z80. La saldatura va effettuata sul circuito stampato dal lato delle saldature. In figura 146 è mostrato lo Z80 **visto da sotto** con indicati i quindici piedini ai quali vanno saldati i fili provenienti dall'Interfaccia. Per i collegamenti tra computer e Interfaccia il filo più adatto può essere quello per WIRE-WRAP che è appunto molto sottile e ad un capo.

Prima di richiudere il computer controllare di avere collegato i fili esattamente e che i fili stessi siano saldati bene e non si stacchino.

Se avete usato dei fili abbastanza sottili potete farli uscire all'esterno tramite uno dei fori già esistenti sui due ZX.

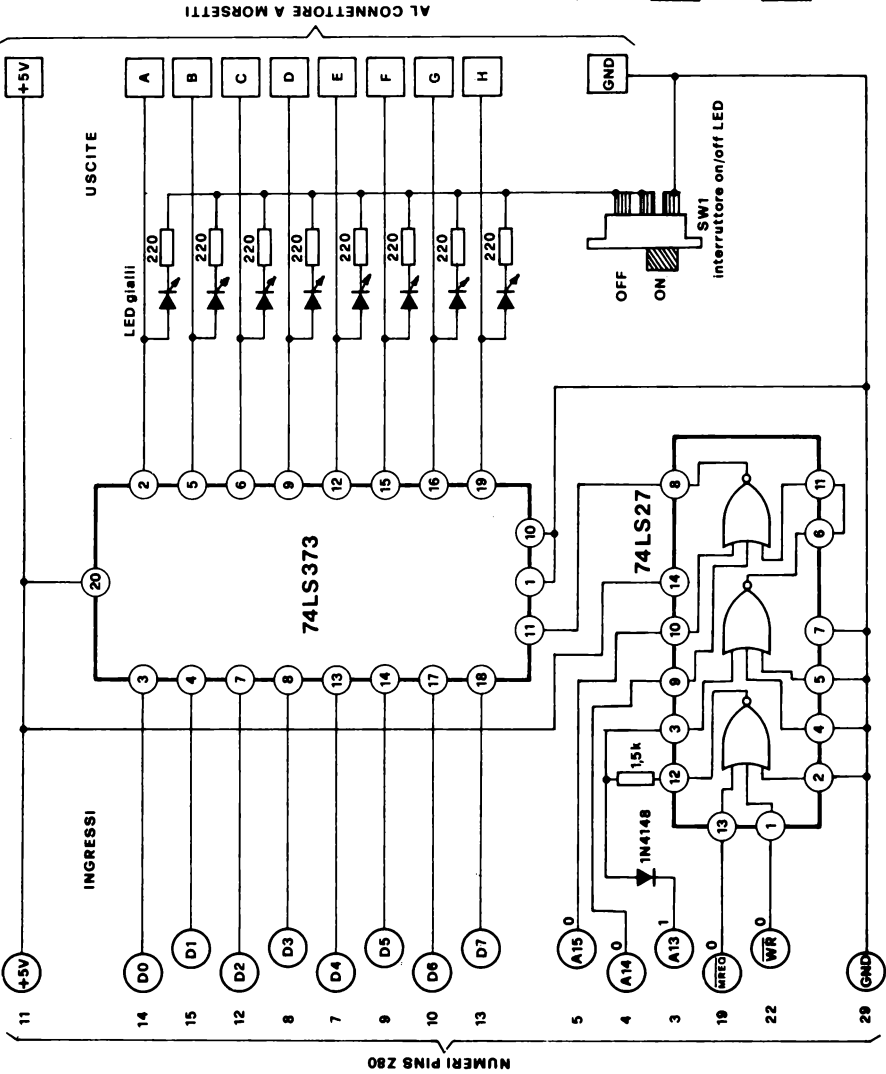
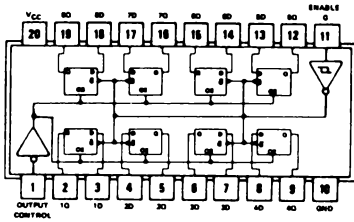


Figura 143

OCTAL D-TYPE LATCHES

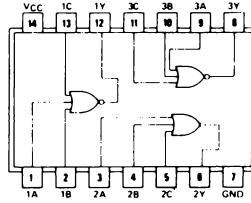
3 STATE OUTPUTS
COMMON OUTPUT CONTROL
COMMON ENABLE



74LS373

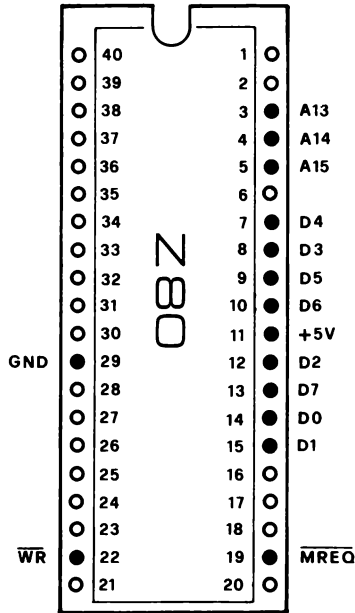
Figura 144

TRIPLE 3-INPUT
POSITIVE-NOR GATES



74LS27

Figura 145



C.I. VISTO DA SOTTO IL
CIRCUITO STAMPATO
(LATO SALDATURE)

Figura 146

Le dieci uscite dell'Interfaccia (A.....H, +5V, GND) vanno collegate ad un connettore del tipo visibile in fig. 147 al quale potranno essere collegati i circuiti esterni come ad esempio la scheda musicale.

Il consumo del circuito senza carico e con i LED non inseriti è di 22 mA, con i LED inseriti e accesi può arrivare fino a 50 mA.

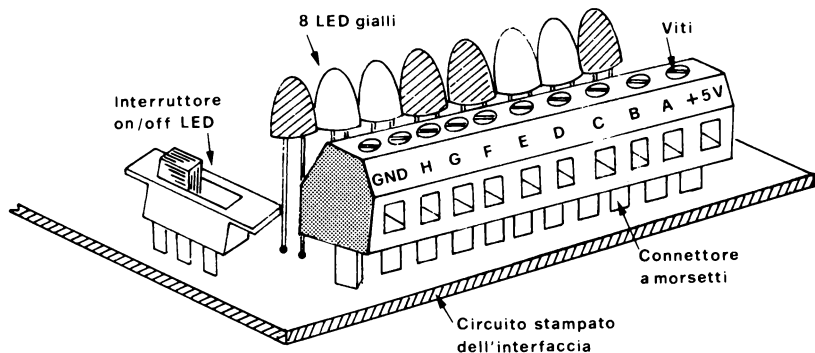


Figura 147

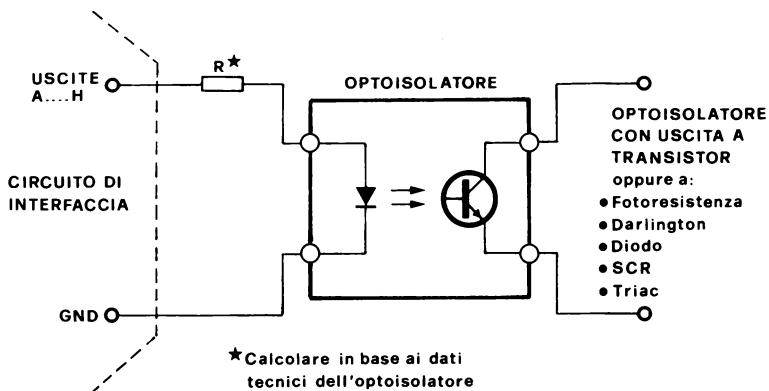


Figura 148

Il funzionamento della scheda può essere verificato, con l'interruttore per l'inserimento dei LED in posizione ON, eseguendo il seguente programma:

1Ø FAST (solo su ZX-81)
 2Ø POKE 8888,Ø
 3Ø PAUSE 25
 4Ø POKE 8888,255
 5Ø PAUSE 25
 6Ø GOTO 2Ø

Dopo avere premuto RUN e NEWLINE gli otto LED devono accendersi e spegnersi contemporaneamente al ritmo di una volta al secondo.

Se qualcuno dei LED non si accende o rimane sempre acceso ricontrollare innanzitutto le linee 20 e 40 del programma, se tali linee sono corrette allora la causa può essere dovuta ad un LED difettoso, ad una interruzione, ad un corto circuito,

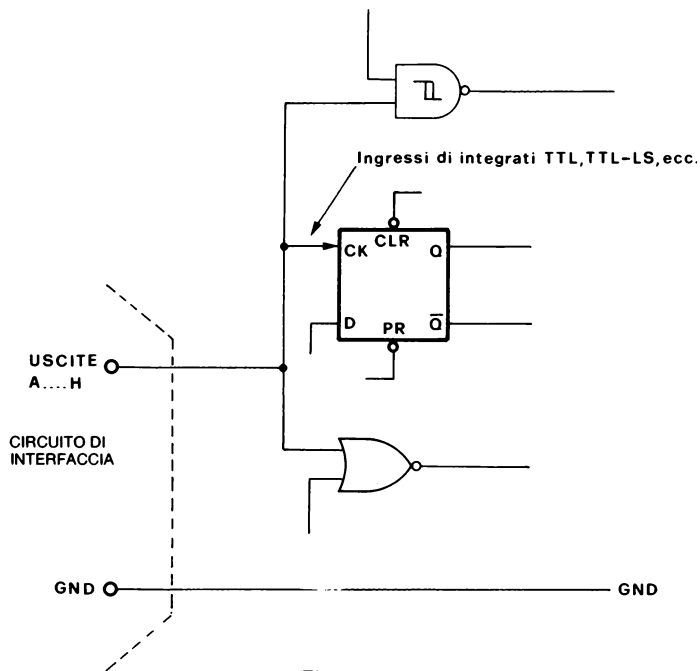
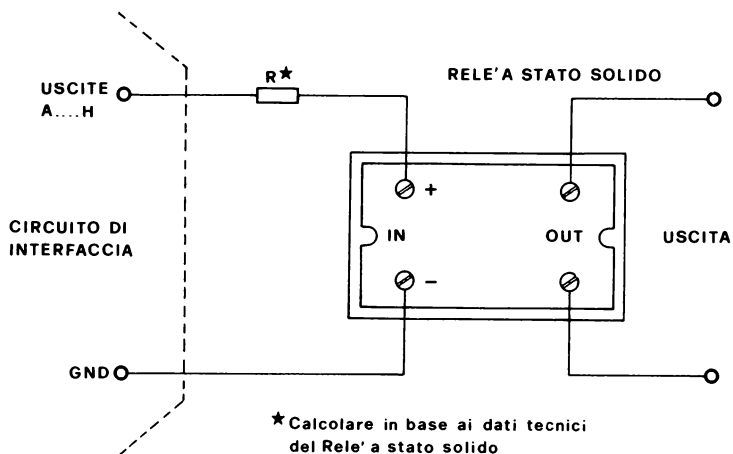


Figura 149



* Calcolare in base ai dati tecnici del Relè a stato solido

Figura 150

all'errato collegamento tra computer e Interfaccia o anche a un integrato difettoso. A seconda del programma utilizzato e dei circuiti collegati alle otto uscite, il circuito di Interfaccia può essere usato in moltissime applicazioni.

Alle otto uscite TTL A....H possono essere collegati vari tipi di circuiti. Si possono collegare, per esempio, degli optoisolatori (fig. 148) con ingresso a LED e uscita a fotoresistenza oppure a fototransistor, a fotodarlington, a fotodiodo, a fotoSCR o a fotoTRIAC. L'uso di optoisolatori consente di isolare il computer e l'Interfaccia dall'apparecchio che devono pilotare.

Naturalmente alle otto uscite dell'Interfaccia si possono collegare gli ingressi di circuiti logici TTL, TTL-LS, TTL-S, ecc. (fig. 149) il numero di ingressi collegabili a ciascuna uscita dipende dal tipo di logica usata. Se con l'Interfaccia si vogliono pilotare carichi ad alta tensione e/o ad alta corrente, si possono collegare ad essa

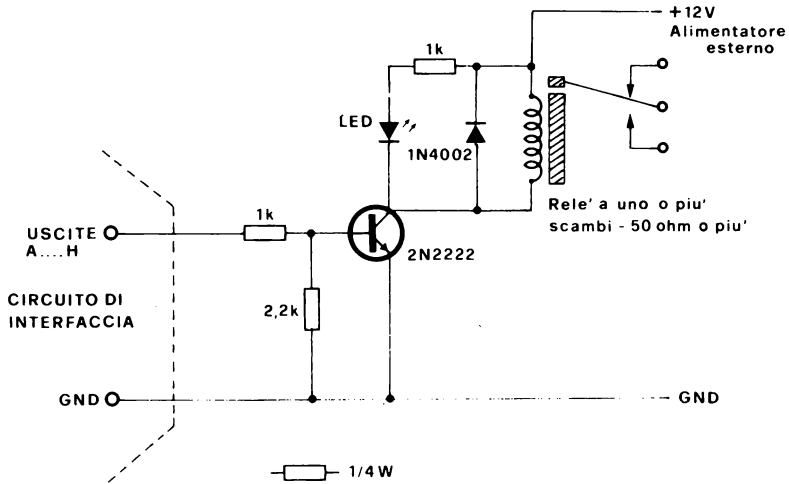


Figura 151

dei Relè a Stato Solido (fig. 150) oppure dei normali Relè Elettromagnetici usando per esempio il circuito di fig. 151. Poichè i Relè assorbono molta corrente occorre alimentarli con un alimentatore esterno e non tramite il computer. Sempre nel circuito di fig. 151 si possono utilizzare Relè funzionanti a tensioni diverse da 12V, oppure usare carichi diversi da un Relè, così come anche il transistor può essere sostituito con altri tipi oppure con un darlington o un VMOS.

Infine, l'interfaccia è indispensabile per pilotare la scheda musicale esposta nel prossimo capitolo.

Nel collegare l'Interfaccia ad un qualsiasi carico occorre tenere presente che il circuito è alimentato dal computer e che, perciò, occorre limitare la corrente assorbita dal suo alimentatore.

Dalle uscite A....H e +5V insieme, si può assorbire un totale di 100-150 mA al massimo (di cui 10-15 mA massimo da ciascuna delle otto uscite TTL e il resto

dall'uscita +5V): tale corrente totale può arrivare anche a 200-250 mA se si usa l'alimentatore Sinclair più grande, alimentatore che è comunque indispensabile se, oltre all'Interfaccia, vengono collegati al computer la stampante e/o l'espansione RAM. Tenete presente che un eccessivo assorbimento di corrente surriscalda l'integrato regolatore a 5V e di conseguenza anche l'interno del computer. L'interfaccia viene comandata dal computer tramite una semplice istruzione in BASIC (POKE). La POKE può essere compresa tra POKE 8192,n e POKE 16383,n si suggerisce, comunque, di usare o la POKE 8888,n oppure la POKE 9999,n in

Tabella 3

| POKE 8888, | USCITA ATTIVA ("1" logico TTL) | |
|------------|-----------------------------------|------|
| 1 | A | (D0) |
| 2 | B | (D1) |
| 4 | C | (D2) |
| 8 | D | (D3) |
| 16 | E | (D4) |
| 32 | F | (D5) |
| 64 | G | (D6) |
| 128 | H | (D7) |

POKE: *minimo* 8888,0 *massimo* 8888,255

con POKE 8888,0 Tutte le uscite DISATTIVATE ("0" logico TTL)

esempio: POKE 8888,75

USCITE ATTIVE: A, B, D, G

$$(1+2+8+64 = 75)$$

quanto sono più facili da ricordare e usano solo quattro numeri (in questo e nei prossimi capitoli viene usata la POKE 8888 nei programmi che usano solo l'Interfaccia, e la POKE 9999 nei programmi che usano la scheda musicale, questo per indicare quando la POKE viene usata per produrre un suono). Nella POKE il numero "n" va compreso tra 0 e 255, con "n" uguale a 0 tutte e otto le uscite sono allo "0" logico, con "n" uguale a 255 tutte e otto le uscite sono all'"1" logico. Le combinazioni richieste possono essere ottenute tramite la Tabella 3; per esempio, se si vogliono attivare le uscite B, E, F, H, occorre sommare i corrispondenti numeri della Tabella 3 ($2+16+32+128 = 178$) ed effettuare l'istruzione POKE 8888,178. Allo stesso modo si può ottenere qualsiasi combinazione di uscite attive e non attive.

Nei programmi in cui occorra una maggiore velocità l'Interfaccia può essere comandata anche da un'istruzione in Linguaggio Macchina, anche in questo caso scrivendo un numero (n) in uno qualsiasi degli indirizzi compresi tra 8192 e 16383.

Programmi

L'uso dell'Interfaccia è legato soprattutto alle applicazioni ideate da ciascun utilizzatore. In sostanza ognuno potrà collegare il computer tramite l'Interfaccia a delle attrezzature, dei servomeccanismi, degli strumenti, ecc. che utilizza nel proprio lavoro o nel proprio hobby, questo allo scopo di rendere automatici o più veloci alcuni processi o di avere nuove funzioni e applicazioni rese possibili dalla potenza e dalla velocità elaborativa del computer.

Le applicazioni esposte qui di seguito comprendono un programmatore universale utilizzabile in molti casi pratici, alcuni programmi di esempio per produrre degli effetti di luce ed infine un tasto elettronico per telegrafia.

Programma universale per circuito di Interfaccia

FAST

Il programmatore universale (fig. 152) può eseguire una serie di istruzioni, alternate a delle pause di durata prestabilita.

La stringa D\$ (linea 110) contiene in forma decimale le istruzioni sperate da uno

```

10  REM  PROGRAMMATORE
20  REM  UNIVERSALE PER
30  REM  CIRCUITO DI INTERFACCIA
100 FAST
102 REM  D$ CONTIENE I CODICI
104 REM  PER PILOTARE I
106 REM  DISPOSITIVI COLLEGATI
108 REM  ALL'INTERFACCIA
110 LET  D$="054 155 200 015 014
100 108 011 255 054 200 038"
112 REM  P$ CONTIENE LE PAUSE
114 REM  DA EFFETTUARE TRA UNA
116 REM  POKE E LA SUCCESSIVA
120 LET  P$="100 100 150 500 050
010 000 100 030 200 100 225"
130 FOR  A=1 TO LEN D$ STEP 4
140 POKE 8888,VAL D$(A TO A+2)
150 PAUSE VAL P$(A TO A+2)
160 NEXT A
190 GOTO 130

```

Figura 152

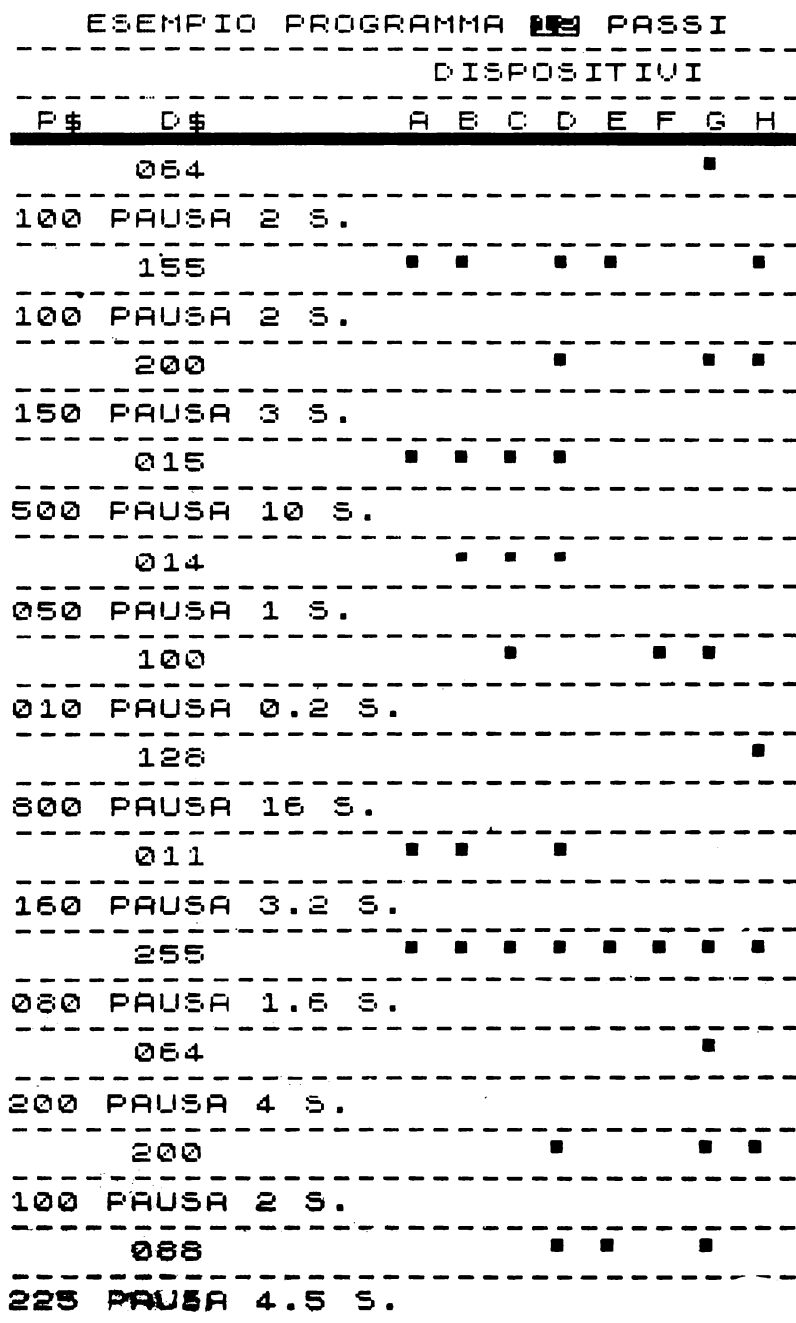


Figura 153

spazio; il numero decimale di ogni istruzione (che deve essere sempre di tre cifre) può essere ricavato utilizzando la Tabella 3 in base ai dispositivi (collegati alle uscite A...H) che l'istruzione stessa deve attivare o non attivare in quel dato momento.

```

145 LET T=VAL P$(A TO A+2)
150 FOR M=1 TO 50
160 PAUSE T
170 NEXT M

```

Figura 154

```

145 LET T=VAL P$(A TO A+2)
150 FOR M=1 TO 3000
160 PAUSE T
170 NEXT M

```

Figura 155

```

10 REM LUCI ROTANTI
100 FAST
110 LET T=1
120 POKE 8088,1
130 PAUSE T
140 POKE 8088,2
150 PAUSE T
160 POKE 8088,4
170 PAUSE T
180 IF INKEY$="6" THEN LET T=T+
1
190 IF T>1 AND INKEY$="7" THEN
LET T=T-1
200 GOTO 120

```

Figura 156

La stringa P\$ (linea 120) contiene, invece, le pause che il computer deve inserire tra l'esecuzione di una istruzione e la successiva.

Il programma di fig. 152 contiene nelle stringhe D\$ e P\$ un esempio di programmazione lungo 12 passi, esempio esposto più chiaramente in fig. 153.

La prima istruzione di D\$ è 064 che attiva il dispositivo collegato all'uscita G dell'Interfaccia; la prima istruzione di P\$ è 100 che introduce una pausa di 2 secondi, dopodichè viene eseguita la seconda istruzione di D\$ (155) che attiva altri dispositivi, seguita da un'altra Pausa e così via.

La sequenza appena esposta può essere, ad esempio, quella di un processo di automazione.

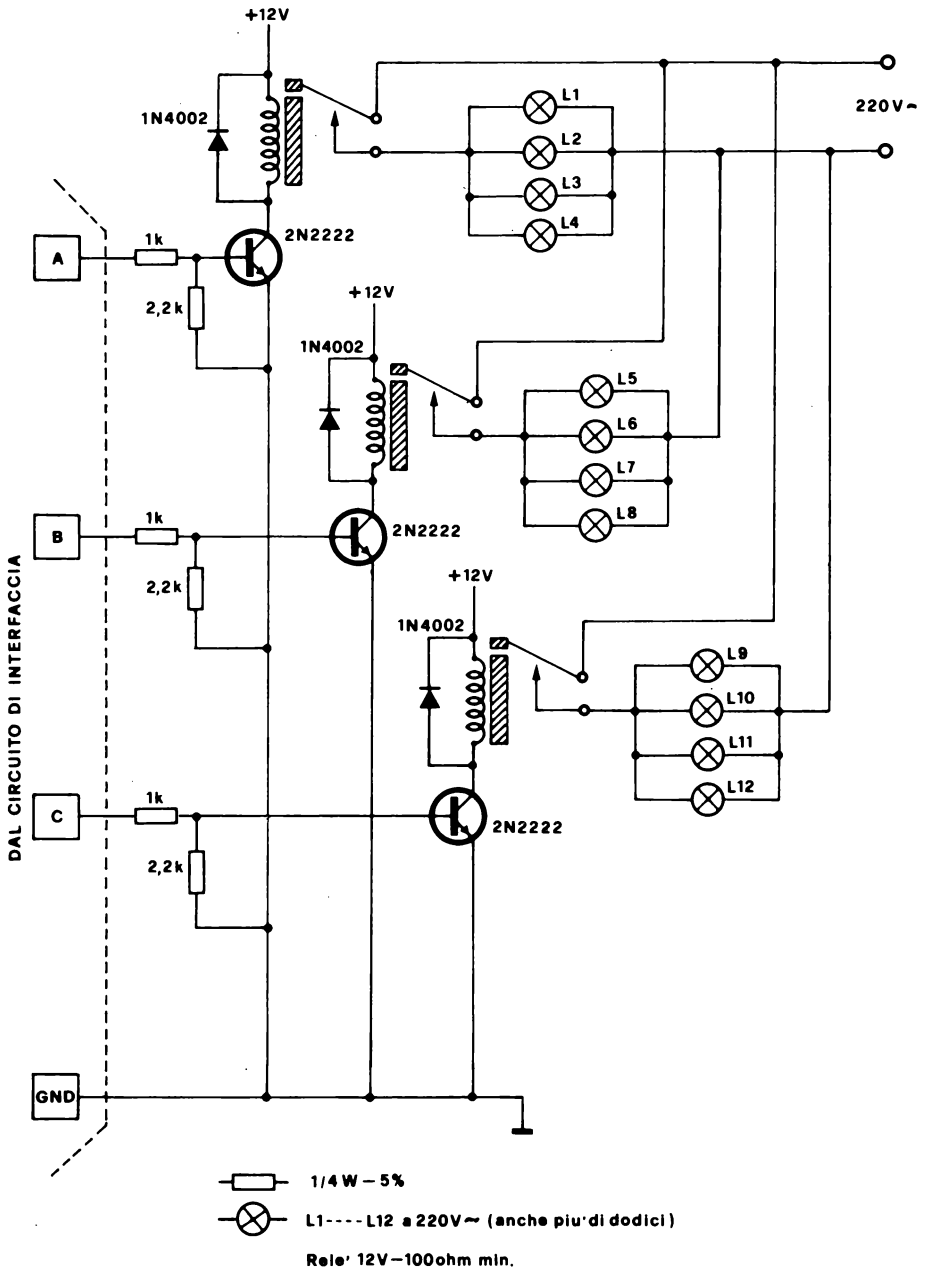


Figura 157

Senza collegare dei dispositivi alle uscite dell'Interfaccia, il funzionamento del programma può essere messo in evidenza dall'accendersi e spegnersi degli otto LED.

Naturalmente la sequenza può esser costituita da un numero maggiore o minore delle dodici istruzioni dell'esempio purchè il numero di istruzioni contenute in D\$ e in P\$ sia uguale, vale a dire che le due stringhe abbiano la stessa lunghezza.

I numeri delle istruzioni di pausa contenute nella stringa P\$ possono essere compresi tra 000 e 999, con una pausa massima, quindi, di circa 20 secondi. Inserendo nel programma le linee di fig. 154 i numeri contenuti in P\$ non saranno più cinquantesimi di secondo ma secondi (da 000 a 999). Infine, inserendo nel programma le linee di fig. 155 i numeri contenuti in P\$ corrisponderanno a minuti.

Tenete presente che i computer Sinclair non usano come clock un quarzo ma un filtro ceramico che ha una precisione inferiore, per cui anche i minuti o i secondi o i cinquantesimi di secondo delle istruzioni di pausa contenute in P\$ avranno una precisione inferiore a sistemi quarzati.

Luci rotanti

FAST

Il programma di luci rotanti (fig. 156) è il primo di quattro effetti di luce che utilizzano il circuito di Interfaccia. Per usare il programma occorre collegare a tre delle uscite dell'Interfaccia (A, B, C) tre relè (fig. 157), con alimentatore separato dal computer, ciascuno dei quali pilota l'accensione di due o più lampadine anch'esse alimentate esternamente e disposte come in fig. 158.

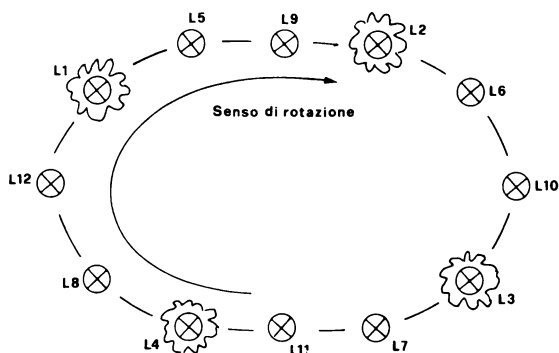


Figura 158

La velocità di rotazione che parte con il massimo può essere diminuita tenendo premuto il tasto "6" e di nuovo aumentata tenendo premuto il tasto "7".

Il senso di rotazione può essere invertito senza toccare la sistemazione delle lampadine semplicemente spostando il contenuto della linea 160 nella linea 120 ed il contenuto della linea 120 nella linea 160.

Il programma di fig. 159 comanda l'accensione e lo spegnimento casuale di otto lampadine, collegate alle otto uscite dell'Interfaccia tramite otto circuiti come quello di fig. 160.

```

10 REM LUCI CASUALI
100 FAST
110 LET T=1
120 RAND
130 LET N=INT (256*RND)
140 POKE 8888,N
150 IF INKEY$="6" THEN LET T=T+
1
160 IF T>1 AND INKEY$="7" THEN
LET T=T-1
170 PAUSE T
180 GOTO 130
    
```

Figura 159

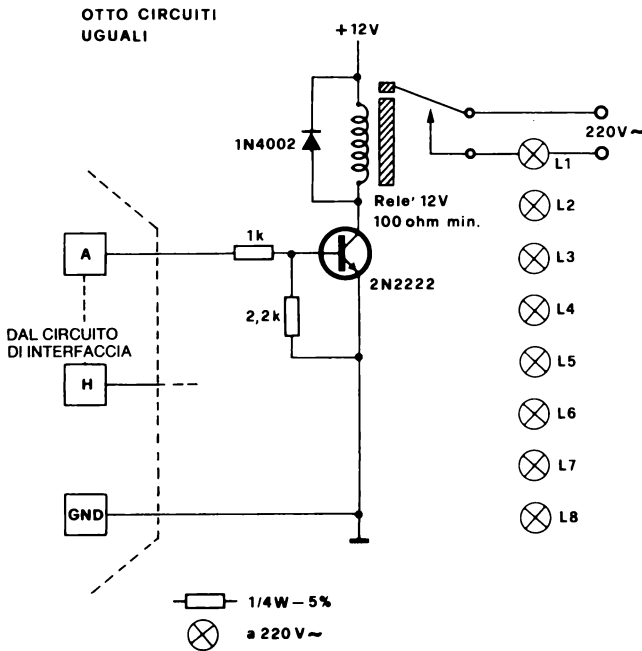


Figura 160

Come nel programma precedente, anche in questo caso, la velocità di cambiamento si può diminuire premendo il tasto "6" e di nuovo aumentare premendo il tasto "7", tale velocità si può, inoltre, rendere casuale modificando il programma come in fig. 161.

```

10 REM LUCI CASUALI
20 REM CON RITARDO CASUALE
100 FAST
110 LET T=1
120 RAND
130 LET N=INT (255*RND)
140 POKE 8888,N
150 PAUSE INT (10*RND)
160 GOTO 130

```

Figura 161

Senza collegare alle uscite i relè e le lampadine il funzionamento di questo e dei prossimi due programmi può essere messo in evidenza anche dai LED dell'Interfaccia.

Luci a riempimento e svuotamento

FAST

Sempre con 8 lampadine alle otto uscite dell'Interfaccia si può ottenere un effetto di riempimento e svuotamento graduale della linea di luci facendo girare il programma di fig. 162.

```

10 REM LUCI A RIEMPIMENTO
20 REM E SVUOTAMENTO
100 FAST
110 LET T=1
120 LET M=1
130 IF M=1 THEN FOR A=0 TO 8
140 IF M=-1 THEN FOR A=8 TO 0 S
TEP -1
150 POKE 8888,2*A-1
160 IF INKEY$="6" THEN LET T=T+
1
170 IF T>1 AND INKEY$="7" THEN
LET T=T-1
180 FOR D=1 TO T
190 NEXT D
200 NEXT A
210 LET M=M*-1
220 GOTO 130

```

Figura 162

Anche qui la velocità può essere modificata premendo i tasti 6 e 7. Si può anche ottenere un effetto di solo svuotamento modificando le linee 130 e 210 nel modo indicato in fig. 163, oppure un effetto di solo riempimento modificando le stesse linee come indicato in fig. 164.

```

130 FOR A=8 TO 0 STEP -1

210 POKE 8888,255

```

Figura 163

```

130 FOR A=0 TO 8

210 POKE 8888,0

```

Figura 164

Punto luminoso mobile

FAST

L'ultimo programma per effetti di luce (fig. 165) fa muovere un punto luminoso lungo la linea di otto lampadine collegate alle otto uscite dell'Interfaccia. Di nuovo i tasti 6 e 7 regolano la velocità di movimento.

```

10 REM PUNTO LUMINOSO MOBILE
100 FAST
110 LET T=1
120 FOR A=0 TO 7
130 POKE 8888,2**A
140 IF INKEY$="6" THEN LET T=T+
1
150 IF T>1 AND INKEY$="7" THEN
LET T=T-1
160 FOR D=1 TO T
170 NEXT D
180 NEXT A
190 POKE 8888,0
200 GOTO 120

```

Figura 165

Tasto automatico per telegrafia

FAST

Sempre utilizzando l'Interfaccia si può realizzare un Tasto automatico per telegrafia da 3-30 parole/minuto facendo girare il programma di fig. 166.

```

1000 REM TASTO AUTOMATICO
1100 REM PER TELEGRAFIA
1200 REM 3-30 PAROLE/MINUTO
1300 REM
1400 REM <C> = .....
1500 REM <B> = .....
1600 REM
1700 PRINT
1800 PRINT "INSERIRE VELOCITA'"
1900 PRINT "1-9 PUNTI AL SECONDO"
2000 PRINT "E PREMERE NEWLINE"
2100 INPUT P
2200 LET P=INT P
2300 IF P<1 OR P>9 THEN GOTO 140
2400 LET T=INT (25/P)
2500 LET TT=0*T
2600 IF INKEY$="C" THEN GOTO 2800
2700 IF INKEY$="B" THEN GOTO 2800
2800 PRINT " "
2900 PRINT " "
3000 PRINT " "
3100 PRINT " "
3200 PRINT " "
3300 PRINT " "
3400 PRINT " "
3500 PRINT " "
3600 PRINT " "
3700 PRINT " "
3800 PRINT " "
3900 PRINT " "
4000 PRINT " "
4100 PRINT " "
4200 PRINT " "
4300 PRINT " "
4400 PRINT " "
4500 PRINT " "
4600 PRINT " "
4700 PRINT " "
4800 PRINT " "
4900 PRINT " "
5000 PRINT " "
5100 PRINT " "
5200 PRINT " "
5300 PRINT " "
5400 PRINT " "
5500 PRINT " "
5600 PRINT " "
5700 PRINT " "
5800 PRINT " "
5900 PRINT " "
6000 PRINT " "
6100 PRINT " "
6200 PRINT " "
6300 PRINT " "
6400 PRINT " "
6500 PRINT " "
6600 PRINT " "
6700 PRINT " "
6800 PRINT " "
6900 PRINT " "
7000 PRINT " "
7100 PRINT " "
7200 PRINT " "
7300 PRINT " "
7400 PRINT " "
7500 PRINT " "
7600 PRINT " "
7700 PRINT " "
7800 PRINT " "
7900 PRINT " "
8000 PRINT " "
8100 PRINT " "
8200 PRINT " "
8300 PRINT " "
8400 PRINT " "
8500 PRINT " "
8600 PRINT " "
8700 PRINT " "
8800 PRINT " "
8900 PRINT " "
9000 PRINT " "
9100 PRINT " "
9200 PRINT " "
9300 PRINT " "
9400 PRINT " "
9500 PRINT " "
9600 PRINT " "
9700 PRINT " "
9800 PRINT " "
9900 PRINT " "

```

Figura 166

All'inizio occorre scegliere una delle nove velocità (fig. 167) inserendo il numero di punti al secondo (da 1 a 9) quindi, premuto NEWLINE, il computer produrrà una serie di punti ogni volta che si preme il Tasto "C" e una serie di linee ogni volta che si preme il Tasto "B"; i punti e le linee prodotti sono autocompletanti e la durata è in proporzione tra loro e in rapporto alla velocità scelta.

```

INSERIRE VELOCITA'
1-9 PUNTI AL SECONDO
E PREMERE NEWLINE

```

Figura 167

Come si è detto si possono inserire nove velocità, togliendo però completamente la linea 150 si potranno inserire velocità intermedie come 1.5, 7.3, 4.4 punti al secondo, ecc.

Per utilizzare praticamente il programma del Tasto per telegrafia occorre collegare il circuito di fig. 168 tra l'uscita H dell'Interfaccia e l'ingresso CW del trasmettitore. Come noterete alle linee 230, 250, 280 e 300 è stata usata, invece della POKE 8888 la POKE 9999 che, come già detto all'inizio di questo capitolo, indica la generazione

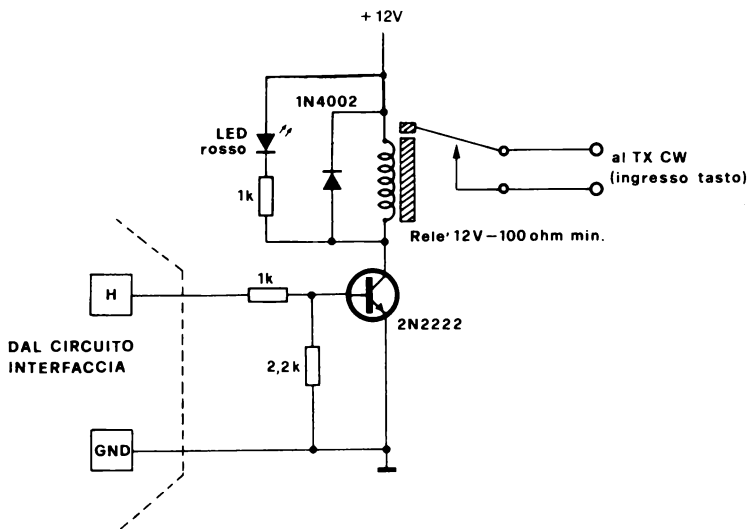


Figura 168

di un suono, questo perchè, collegando all'Interfaccia il circuito musicale descritto nel prossimo capitolo, si può dotare il Tasto per telegrafia di un monitor sonoro senza modificare il programma.

MUSICA

Scheda musicale

Con una spesa inferiore alle 10000 lire si può collegare ai computers ZX80 e ZX81 (tramite il circuito di Interfaccia già esposto nel capitolo precedente) una scheda musicale capace di produrre 50 note musicali su 4 ottave tramite delle semplici istruzioni in BASIC.

Lo scopo principale è quello di permettere al computer di produrre della musica, però tale scheda si presta a molti altri usi come per esempio sonorizzare dei giochi, produrre dei segnali di allarme, e, comunque, per inserire nei programmi delle note acustiche ovunque può essere utile.

Il circuito (fig. 169) usa tre comuni integrati TTL-LS più un timer 555, un transistor 2N2222 e pochi altri componenti passivi.

Le configurazioni dei quattro integrati e del transistor sono in fig. 170. Il circuito può essere montato su un pezzo di circuito stampato già forato.

È essenziale che le resistenze siano tutte al 5% (1/4 W), che il condensatore da 4700 pF sia del tipo poliestere al 5 - 10 % massimo che i TTL usati siano della serie LS ed infine che i Trimmers siano di buona qualità e possibilmente del tipo incapsulato (come i Trimmer neri e bianchi della PHIER) in modo che la taratura non venga modificata da urti accidentali.

I dieci ingressi (+5V, A.....H, GND) della scheda musicale così realizzata vanno collegati tramite dieci spezzoni di filo di tre colori ai corrispondenti dieci morsetti di uscita del circuito di Interfaccia (fig. 171), fare in particolare attenzione a non invertire tra loro i collegamenti che vanno alle uscite + 5V e GND.

La corrente media che la scheda musicale assorbe dall'alimentatore del computer va dai 30 mA al minimo volume ai 65 mA al massimo volume. Il consumo può essere ridotto e la potenza sonora può essere aumentata togliendo il transistor e l'altoparlante e collegando l'uscita della scheda musicale ad un amplificatore audio più potente ed alimentato esternamente (fig. 172).

La taratura delle 50 frequenze della scheda musicale va effettuata regolando solo i sei trimmers (tm1...tm6) in modo da ottenere determinate frequenze di uscita, il settimo trimmer (tm7) è usato per la regolazione del Volume, vi è anche un interruttore a slitta (SW1) che serve per l'esclusione del suono. Per effettuare la taratura occorre collegare un frequenzimetro digitale all'uscita "DFM" della scheda musicale (vedere fig. 169) e far girare sul computer il seguente programma:

DAL CIRCUITO DI INTERFACCIA

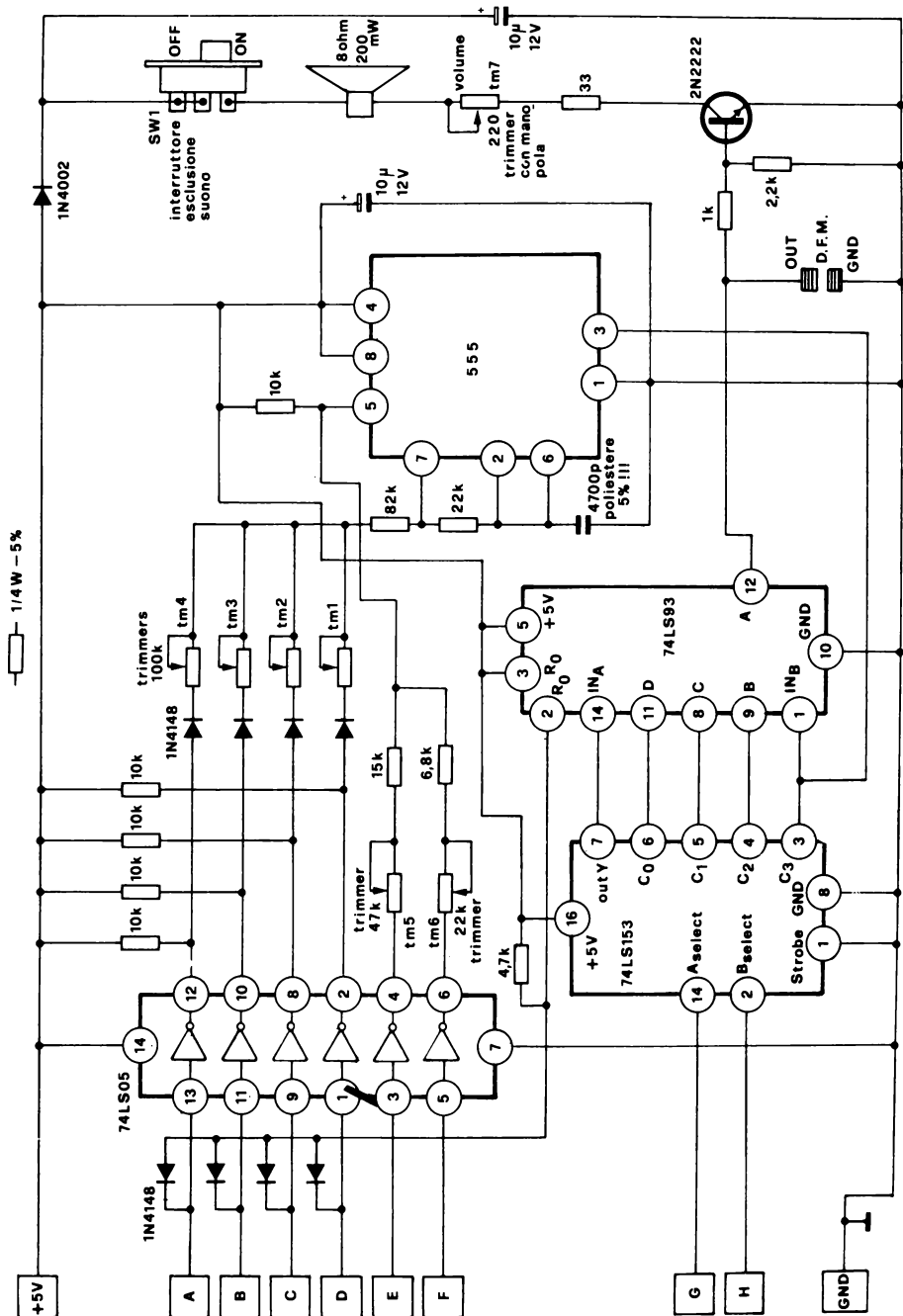
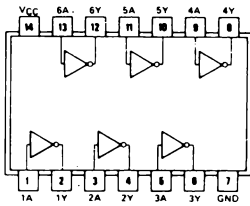


Figura 169

5 REM ROUTINE TARATURA
 10 FAST
 20 PRINT "INSERIRE CODICE NOTA"
 30 INPUT N
 40 CLS
 50 POKE 9999,N
 60 GOTO 20

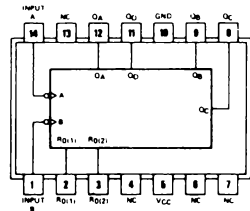
programma necessario per evitare di scrivere ad ogni nota l'istruzione POKE 9999,n.

HEX INVERTERS WITH OPEN-COLLECTOR OUTPUTS



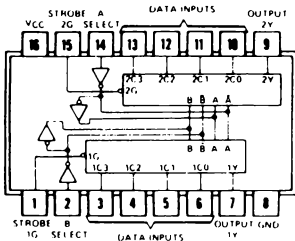
74LS05

DIVIDE-BY-TWO AND DIVIDE-BY-EIGHT

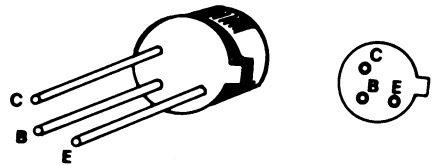


74LS93

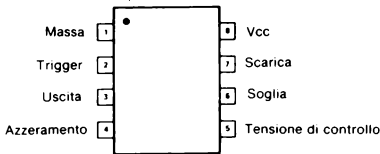
DUAL 4-LINE TO 1-LINE DATA SELECTORS/MULTIPLEXERS



74LS153



2N2222



555

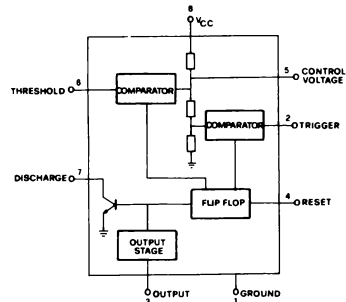


Figura 170

Dopo aver premuto RUN e NEWLINE, le sei tarature vanno effettuate secondo l'ordine indicato nella Tabella 4, introducendo per primo il codice 199 e regolando il trimmer tm1 in modo da avere in uscita una frequenza il più possibile vicina a 523.25 Hz, quindi proseguendo con l'introduzione degli altri cinque codici e con la

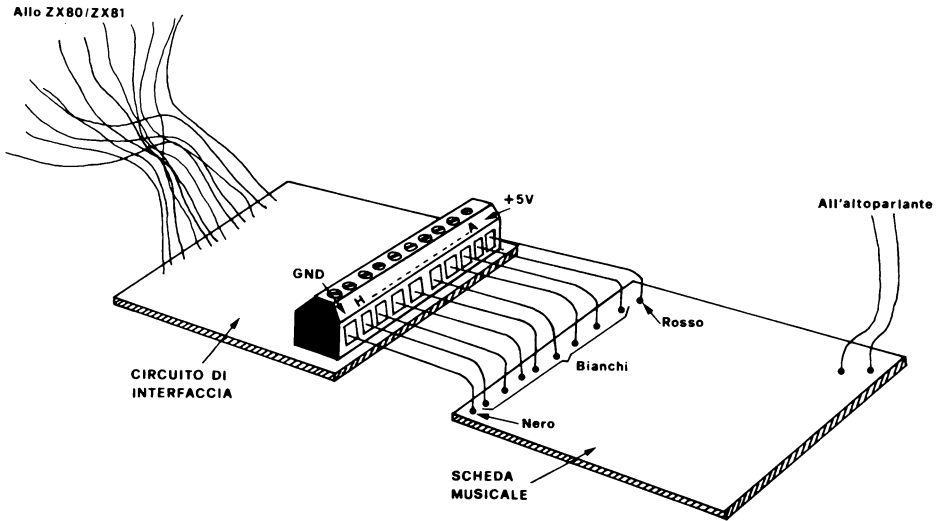


Figura 171

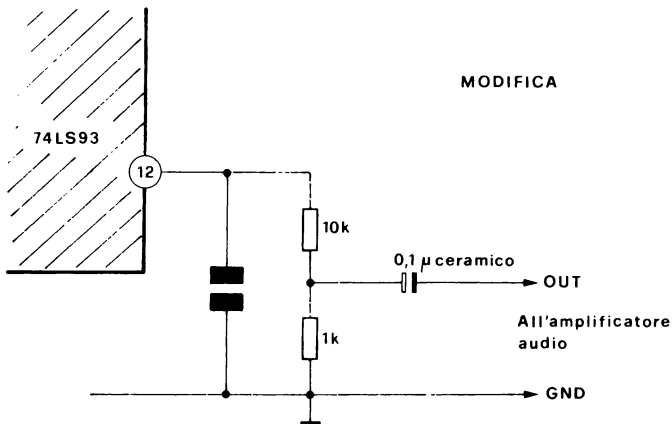


Figura 172

Tabella 4

| ORDINE DI TARATURA | POKE 9999, (codice nota) | trimmer da regolare | Frequenza da approssimare (Hz) |
|------------------------------|--------------------------|---------------------|--------------------------------|
| 1 ^a | 199 | tm 1 | 523,25 |
| 2 ^a | 203 | tm 2 | 554,37 |
| 3 ^a | 205 | tm 3 | 587,33 |
| 4 ^a | 206 | tm 4 | 622,25 |
| 5 ^a | 219 | tm 5 | 698,46 |
| 6 ^a | 235 | tm 6 | 880,00 |
| <i>tm 7 - Trimmer VOLUME</i> | | | |

regolazione dei corrispondenti trimmers in modo da ottenere le frequenze indicate nella tabella. Nell'effettuare la taratura fare attenzione ad inserire esattamente i sei codici e a non sbagliare nello scegliere ogni volta il trimmer da regolare, un'altra cosa importante è quella di regolare i trimmers con un cacciavite senza urtare gli altri trimmers già tarati e soprattutto senza toccare con le mani i trimmers né altri

Tabella 5

| FREQUENZA DELLA SCHEDA MUSICALE 4 OTTAVE - 50 NOTE | | | |
|--|-------------------------|--------------------------|--------------------|
| NOTA | FREQUENZA STANDARD (Hz) | CODICE NOTA (POKE 9999,) | CODICE ESADECIMALE |
| 1 DO | 65,406 | 7 | 07 |
| 2 DO# | 69,296 | 11 | 0B |
| 3 RE | 73,416 | 13 | 0D |
| 4 RE# | 77,782 | 14 | 0E |
| PAUSA | 0 | 15 | 0F |
| 5 MI | 82,407 | 23 | 17 |
| 6 FA | 87,307 | 27 | 1B |
| 7 FA# | 92,499 | 29 | 1D |
| 8 SOL | 97,999 | 30 | 1E |
| 9 SOL# | 103,83 | 39 | 27 |
| 10 LA | 110,00 | 43 | 2B |
| 11 LA# | 116,54 | 45 | 2D |
| 12 SI | 123,47 | 46 | 2E |
| 13 DO | 130,81 | 71 | 47 |
| 14 DO# | 138,59 | 75 | 4B |

Tabella 5 (Continua)

| FREQUENZA DELLA SCHEDA MUSICALE 4 OTTAVE - 50 NOTE | | | | |
|--|------|----------------------------|-----------------------------|-----------------------|
| | NOTA | FREQUENZA STANDARD (Hz) | CODICE NOTA (POKE 9999,) | CODICE ESADECIMALE |
| 15 | RE | 146,83 | 77 | 4D |
| 16 | RE# | 155,56 | 78 | 4E |
| 17 | MI | 164,81 | 87 | 57 |
| 18 | FA | 174,61 | 91 | 5B |
| 19 | FA# | 185,00 | 93 | 5D |
| 20 | SOL | 196,00 | 94 | 5E |
| 21 | SOL# | 207,65 | 103 | 67 |
| 22 | LA | 220,00 | 107 | 6B |
| 23 | LA# | 223,08 | 109 | 6D |
| 24 | SI | 246,94 | 110 | 6E |
| 25 | DO | 261,63 | 135 | 87 |
| 26 | DO# | 277,18 | 139 | 8B |
| 27 | RE | 293,66 | 141 | 8D |
| 28 | RE# | 311,13 | 142 | 8E |
| 29 | MI | 329,63 | 151 | 97 |
| 30 | FA | 349,23 | 155 | 9B |
| 31 | FA# | 369,99 | 157 | 9D |
| 32 | SOL | 392,00 | 158 | 9E |
| 33 | SOL# | 415,30 | 167 | A7 |
| 34 | LA | 440,00 | 171 | AB |
| 35 | LA# | 466,16 | 173 | AD |
| 36 | SI | 493,88 | 174 | AE |
| 37 | DO | 523,25 | 199 | C7 |
| 38 | DO# | 554,37 | 203 | CB |
| 39 | RE | 587,33 | 205 | CD |
| 40 | RE# | 622,25 | 206 | CE |
| 41 | MI | 659,26 | 215 | D7 |
| 42 | FA | 698,46 | 219 | DB |
| 43 | FA# | 739,99 | 221 | DD |
| 44 | SOL | 783,99 | 222 | DE |
| 45 | SOL# | 830,61 | 231 | E7 |
| 46 | LA | 880,00 | 235 | EB |
| 47 | LA# | 932,33 | 273 | ED |
| 48 | SI | 987,77 | 238 | EE |
| 49 | DO | 1046,0 | 229 | E5 |
| 50 | DO# | 1108,7 | 236 | EC |

fili o contatti scoperti o piedini degli integrati, questo perchè la resistenza delle dita potrebbe falsare la lettura sul frequenzimetro e di conseguenza la precisione di tutta la taratura. Alla fine della taratura è bene ricontrollare le sei frequenze reinserendo i sei codici della Tabella 4 sempre utilizzando la ROUTINE TARATURA già esposta, routine utile anche successivamente per ascoltare le 50 note che la scheda può produrre.

Le 50 note insieme alle loro frequenze standard e alle corrispondenti POKEs da effettuare per ottenerle sono esposte nella tabella 5, nella stessa Tabella è indicata anche la POKE (9999,15) da effettuare per avere una PAUSA, vale a dire nessuna nota o frequenza emessa.

Se la taratura è stata effettuata con cura la precisione delle 50 note dovrebbe essere migliore dello 0.5 — 1% rispetto ai valori standard indicati nella Tabella 5, il circuito utilizzato per la scheda musicale, oltre alla precisione, ha anche un'ottima stabilità sebbene non sia quarzato ma a resistenza e condensatore.

Sempre nella Tabella 5 sono indicati anche i codici esadecimali di tutte le note e della Pausa, codici utili per rendere più compatte all'interno dei programmi le stringhe contenenti lunghe sequenze sonore o musicali.

Programmi

La scheda musicale è senz'altro il circuito più interessante in questo libro per cui ad esso vengono dedicati 4 programmi in questo capitolo sulla musica e ben 14 programmi di effetti sonori nel prossimo capitolo.

Oltre che per la musica e per produrre effetti sonori la scheda musicale può essere usata in molti altri programmi ovunque possa essere necessario o utile aggiungere dei suoni, in questo stesso libro vi sono alcuni esempi, di tali applicazioni nel capitolo prima di questo, dove la scheda è usata come monitor sonoro del Tasto per telegrafia, e nell'ultimo capitolo dove la scheda permette di realizzare dei giochi con anche il sonoro.

Organo a 4 ottave

FAST

Il programma musicale più interessante e certamente quello di fig. 173 che trasforma il computer in un organo elettronico a 4 ottave/49 note da DO 65.4Hz a DO 1046.0 Hz.

L'organo usa la tastiera speciale N. 3 posta alla fine del libro e che va ritagliata e sovrapposta alla tastiera dello ZX. Poichè il computer ha solo 40 tasti, mentre ne occorrerebbero 49, vengono usati solo 25 tasti i quali suonano le 2 ottave più basse premendoli da soli e le 2 ottave più alte premendoli assieme al tasto SHIFT che nella tastiera speciale corrisponde al tasto rosso.

I codici delle POKEs effettuate dal programma, a seconda del tasto premuto e della nota da emettere, sono nella stringa A\$ (linea 110). Se la lunghezza (LEN) della

```

1000 REM ORGANO ELETTRONICO
1100 REM 4 OTTAVE - 26 TASTI
1200 FAST
1300 LET A$=""
1400 FOR I=0 TO 25
1500   FOR J=0 TO 25
1600     POKE 1000+I*26+J, VAL A$(I+J)
1700   NEXT J
1800 NEXT I
1900 PRINT A$
2000 POKE 9999, VAL A$(A-3 TO A-1)
2100 GOTO 1200

```

Figura 173

stringa A\$ è diversa da 271 il programma non funziona, per controllare di avere inserito tutti i codici e gli spazi è sufficiente inserire nel programma la linea 111 (fig. 174), quindi far girare il programma e fermarlo con il tasto BREAK; una volta

```

111 PRINT LEN A$

```

Figura 174

verificato sul video che la lunghezza di A\$ è esatta la linea 111 può essere tolta ed il programma utilizzato regolarmente con la tastiera speciale.

Metronomo

FAST

Il programma di fig. 175 produce le battute tipiche di un Metronomo il cui numero di battute al minuto (da 60 a 300) può essere predeterminato inserendolo all'inizio (fig. 176).

Il programma può essere fermato usato il tasto BREAK.

La nota emessa dal metronomo può essere cambiata sostituendo il codice 71 della POKE (linea 200) con un altro dei codici della Tabella 5.


```

100 REM METRONOMO
110 SLOW
120 PRINT "INSERIRE N° BATTUTE"
130 PRINT "AL MINUTO ( 60--300"
140 PRINT "E PREMERE NEWLINE"
150 INPUT B
160 IF B<60 OR B>300 THEN GOTO
170 CLS
180 FAST
190 LET T=30000/B-1
200 POKE 9999,71
210 PAUSE 1
220 POKE 9999,15
230 PAUSE T
240 GOTO 200

```

Figura 175

```

INSERIRE N° BATTUTE
AL MINUTO ( 60--300 )
E PREMERE NEWLINE

```

Figura 176

Scale Musicali

FAST

Il programma di fig. 177 esegue alternativamente a salire e a scendere la scala delle 50 note prodotte dalla scheda musicale.

```

10000 REM SCALE MUSICALI
10100 FAST
10200 LET A$="007 011 013 014 023
027 030 030 043 045 046 071
075 077 078 087 091 093 094 103
107 109 110 135 139 141 142 151
155 157 158 167 171 173 174 199
20000 2005 2006 2015 2019 221 222 231
20000 FOR T=1 TO 2000 STEP 4
10400 POKE 9999,VAL A$(F TO F+2)
10500 PAUSE 10
10600 NEXT F

```

Figura 177 (Continua

```

1070 POKE 9999,15
1080 PAUSE 200
1090 FOR F=2000 TO 1 STEP -4
1100 POKE 9999,VAL A$(F-3 TO F)
1110 PAUSE 10
1120 NEXT F
1130 POKE 9999,15
1140 PAUSE 200
1150 GOTO 1030

```

Figura 177 (Fine)

La durata di ciascuna nota può essere aumentata o diminuita aumentando o diminuendo le pause alle linee 1080 e 1110. È anche possibile avere scale solo a salire togliendo le linee da 1090 a 1140, o scale solo a scendere togliendo le linee da 1030 a 1080.

Nella stringa A\$ (linea 1090) è importante che dopo l'ultimo codice (236) e prima degli apici vi sia uno spazio.

Generatore di musica casuale - 50 note

FAST

L'ultimo programma (fig. 178) di questo capitolo sulla musica produce una sorta di musica casuale formata da combinazioni delle 50 note.

```

1000 REM GENERATORE DI MUSICA
1005 REM CASUALE - 50 NOTE
1010 FAST
1020 LET T=10
1030 LET A$=""
0007 0011 0013 0014 0023
0027 0029 0030 0037 0043 0045 0046 0071
0075 0077 0078 0087 0091 0093 0094 103
107 109 110 135 139 141 142 151
155 157 158 167 171 173 174 199
203 205 206 215 219 221 222 231
235 237 238 249 256 : 21 22 231
1040 PRINT
1050 FOR N=1 TO 1000
1060 LET F=(1+INT (RAND*50)) *4
1070 POKE 9999,VAL A$(F-3 TO F)
1080 PAUSE T
1090 IF INKEY$="7" AND T>1 THEN
LET T=T-1
1100 IF INKEY$="6" THEN LET T=T+
1
1110 NEXT N
1120 POKE 9999,15

```

Figura 178

La velocità dell'esecuzione musicale può essere aumentata premendo il tasto "7" e diminuita premendo il tasto "6".

Il programma produce una sequenza di mille note, tale sequenza può essere continua togliendo la linea ~~1050~~ e sostituendo la linea ~~1110~~ con la linea:

~~1110~~ GOTO ~~1160~~

Nella stringa A\$ (linea ~~1030~~) è importante che dopo l'ultimo codice (236) e prima degli apici vi sia uno spazio.

EFFETTI SONORI

I programmi di effetti sonori funzionano tutti nel modo FAST e possono essere utilizzati sia da soli che come subroutines di altri programmi. Se avete realizzato e collegato al computer l'Avvisatore acustico per tastiera presentano all'inizio di questo libro, questo va disinserito tramite l'apposito interruttore quando viene fatto girare uno dei programmi di effetti sonori da solo o inserito in altri programmi.

Sirena a 2 toni (fig. 179)

FAST

```
1000 REM SIRENA A 2 TONI
1010 FAST
1020 FOR D=1 TO 30
1030 POKE 9999,141
1040 PAUSE 25
1050 POKE 9999,158
1060 PAUSE 25
1070 NEXT D
1080 POKE 9999,15
```

Figura 179

Sirena americana (fig. 180)

FAST

```
1000 REM SIRENA AMERICANA
1010 FAST
1020 FOR D=1 TO 90
1030 POKE 9999,199
1035 PAUSE 1
1040 POKE 9999,203
1045 PAUSE 1
1050 POKE 9999,205
1055 PAUSE 1
1060 POKE 9999,206
1065 PAUSE 1
1070 POKE 9999,215
1075 PAUSE 1
1080 POKE 9999,219
```

Figura 180 (Continua)

```

1000 POKK M 1
1005 POKK M 9999 , 221
1010 POKK M 1
1100 POKK M 9999 , 222
1105 POKK M 1
1110 POKK M 9999 , 221
1115 POKK M 1
1120 POKK M 9999 , 219
1125 POKK M 1
1130 POKK M 9999 , 215
1135 POKK M 1
1140 POKK M 9999 , 206
1145 POKK M 1
1150 POKK M 9999 , 205
1155 POKK M 1
1160 POKK M 9999 , 203
1165 POKK M 1
1170 NEXT D
1180 POKK M 9999 , 15

```

Figura 180 (Fine)

Sirena multitono (fig. 181)

FAST

```

1000 REM SIRENA MULTITONO
1010 FAST
1020 FOR D=1 TO 30
1030 LET A$="091 103 141 199 "
1040 FOR F=1 TO LEN A$ STEP 4
1050 POKK M 9999,VAL A$(F TO F+2)
1060 PAUSE S
1070 NEXT F
1080 FOR F=LEN A$ TO 1 STEP -4
1090 POKK M 9999,VAL A$(F-3 TO F)
1100 PAUSE S
1110 NEXT F
1120 NEXT D
1130 POKK M 9999 , 15

```

Figura 181

In questa sirena il numero dei toni diversi dipende dal numero di codici contenuti nella stringa A\$ (linea 1030). Nell'esempio di fig. 181 vi sono 4 codici di note per cui la sirena è a 4 toni, dopo ogni codice di nota va sempre inserito uno spazio compreso anche l'ultimo codice (nell'esempio, dopo il codice 199).

Altre modifiche possono riguardare le linee 1060 e 1100 che stabiliscono la durata di ogni nota e la linea 1020 che può essere modificata per aumentare o diminuire la durata del suono.

```

1000 REM      MITRA
1010 FAST
1015 RAND
1020 FOR N=1 TO 10
1030 FOR C=1 TO 3+INT (RND*10)
1040 POKE 9999,30
1050 FOR T=1 TO 4
1060 NEXT T
1070 POKE 9999,15
1080 FOR T=1 TO 4
1090 NEXT T
1100 NEXT C
1110 PAUSE 20+INT (RND*30)
1120 NEXT N

```

Figura 182

Campanella arrivo treni in stazione (fig. 183)

FAST

```

1000 REM      CAMPANELLA ARRIVO
1005 REM      TRENI IN STAZIONE
1010 FAST
1020 FOR N=1 TO 2000
1030 POKE 9999,2000
1040 FOR T=1 TO 12
1050 NEXT T
1060 POKE 9999,15
1070 FOR T=1 TO 1
1080 NEXT T
1090 NEXT N

```

Figura 183

Segnale telefonico di linea occupata (fig. 184)

FAST

```

1000 REM      SEGNALE TELEFONICO
1005 REM      DI LINEA OCCUPATA
1010 FAST
1020 FOR N=1 TO 30
1030 POKE 9999,215
1040 PAUSE 10
1050 POKE 9999,15
1060 PAUSE 10
1070 NEXT N

```

Figura 184


```

1000 REM GRILLO
1010 FAST
1020 FOR N=1 TO 50
1030 FOR K=1 TO 50
1040 POKE 9999,240
1050 FOR T=1 TO 2
1060 NEXT T
1070 POKE 9999,15
1080 FOR T=1 TO 2
1090 NEXT T
1100 NEXT K
1110 PAUSE 25
1120 NEXT N

```

Figura 188

Sostituendo nel programma le linee 1030 e 1110 con quelle di fig. 189 si può ottenere un effetto simile a più Grilli.

```

1000 REM GRILLI
1030 FOR K=1 TO 3
1110 PAUSE 6

```

Figura 189

Il codice 240 usato nella POKE della linea 1040 non è uno dei codici nota della Tabella 5 ma è il codice che fa produrre alla scheda musicale la frequenza più alta (circa 1330 Hz).

```

1000 REM SEGNALE ORARIO
1010 FAST
1020 FOR A=1 TO 5
1030 POKE 9999,238
1040 PAUSE 4
1050 POKE 9999,15
1060 PAUSE 46
1070 NEXT A
1080 PAUSE 50
1090 POKE 9999,238
1100 PAUSE 4
1110 POKE 9999,15

```

Figura 190

Sveglia (fig. 191)

FAST

```

1000 REM SVEGLIA
1010 FAST
1020 FOR N=1 TO 2000
1030 POKE 9999,2000
1040 POKE 9999,2031
1050 POKE 9999,2035
1060 FOR T=1 TO 4
1070 NEXT T
1080 POKE 9999,15
1090 LET X=1
1100 NEXT N

```

Figura 191

Din - Don (fig. 192)

FAST

```

1000 REM DIN-DON
1010 FAST
1020 FOR N=1 TO 10
1030 POKE 9999,238
1040 PAUSE 200
1050 POKE 9999,15
1060 PAUSE 200
1070 POKE 9999,222
1080 PAUSE 200
1090 POKE 9999,15
1100 PAUSE 50
1110 NEXT N

```

Figura 192

Generatore di suoni casuali (fig. 193)

FAST

```

1000 REM GENERATORE DI SUONI
1005 REM CASUALI
1010 FAST
1020 PRINT "INSERIRE RITARDO TRA
SUONO E "
1030 PRINT "SUONO( 1--20 ) E PRE
MERE NEWLINE "
1040 INPUT R
1050 IF R<1 OR R>20 THEN GOTO 10
40

```

Figura 193 (Continua)

```
1060 CLS
1070 RAND
1080 FOR N=1 TO 1000
1090 POKE 9999,RND*255
1100 FOR T=1 TO R
1110 NEXT T
1120 NEXT N
```

Fig. 193 (Fine)

GIOCHI

Prima di far girare il programma giochi che usano la scheda musicale per produrre dei suoni, occorre disinserire l'Avvisatore Acustico per tastiera tramite l'apposito interruttore.

Miniroulette (con sonoro)

SLOW

Il primo gioco (fig. 194) è una Roulette con solo 24 posizioni.

```
10 REM MINI ROULETTE
100 SLOW
105 RAND
110 PRINT AT 0,19;"MINI ROULETT
F:
120 FOR A=1 TO 24 STEP 2
130 PRINT AT 10-10*COS (A/12*PI
),10+10*SIN (A/12*PI);CHR$(A+27
)
140 PRINT AT 10-10*COS ((A+1)/1
2*PI),10+10*SIN ((A+1)/12*PI);CH
R$(A+156)
150 NEXT A
160 LET CX=0
170 LET CY=0
180 PRINT AT 20,17;"PREMERE NEW
L INIZIARE"
F:
190 PRINT AT 21,17;"PER INIZIAR
E
200 INPUT A#
210 UNPLOT CX,CY
220 PRINT AT 20,17;"
230 PRINT AT 21,17;"
240 LET N=25+INT (RAND*25)
250 FOR H=1 TO N
260 LET F=H/12*PI
270 LET CX=21+10*SIN F
```

Figura 194 (Continua)

```

280 LET CY=22.5+17*COS F
290 PLOT CX,CY
300 POKE 9999,199
310 FOR T=1 TO H/3
320 NEXT T
330 POKE 9999,15
340 IF H=N THEN GOTO 180
350 UNPLOT CX,CY
360 NEXT H

```

Figura 194 (Fine)

La Roulette appare sul video come in fig. 195. Premendo NEWLINE la pallina (il quadratino nero all'interno del cerchio) inizia a ruotare in senso orario fermandosi casualmente in una delle 24 posizioni. Durante la rotazione della pallina viene emesso un suono prodotto dalle linee 300 e 330 del programma.

Il programma può essere fermato durante la rotazione della pallina usando il tasto BREAK.

note: La linea 220 contiene tra gli apici 15 spazi, la linea 230 contiene tra gli apici 12 spazi.



Figura 195

Questo programma (fig. 196) simula un campo in cui vi sono 7 mine nascoste casualmente. Il giocatore deve spostare usando i tasti 5, 6, 7, 8, il punto lampeggiante dalla posizione di partenza nell'angolo in basso a sinistra alla posizione di arrivo nell'angolo in alto a destra senza incappare in una delle mine nascoste. Per attraversare il campo minato senza danni il giocatore può contare sul proprio intuito e sulla propria fortuna, se vi riesce appare sul video la scritta "SALVOOO!!!"

```

1000 REM CAMPO MINATO
1010 REM -----
1020 GOTO 1
1030 FOR A = 0 TO 15
1040 FOR B = 1 TO 88
1050 PRINT A " ";
1060 NEXT B
1070 PRINT A
1080 LET C = 1
1090 PRINT C
1100 PRINT AT 0,0;"0"
1110 PRINT AT 7,0;"0"
1120 PRINT D
1130 FOR D = 1 TO 7
1140 LET T = INT (300*RAND)
1150 LET F = INT (200*RAND)
1160 IF T < 10 OR F > 14 THEN GOTO 2
1170
1180 LET M(A,1) = F
1190 LET M(A,2) = G
1200 NEXT A
1210 PRINT AT L,0;" "
1220 PRINT AT 0,17;" "
1230 PRINT AT 0,0;" "
1240 LET X = L
1250 LET Y = 0
1260 IF C > 0 AND INKEY$ = "5" THEN
1270 LET C = C - 1
1280 IF L < 21 AND INKEY$ = "6" THEN
1290 LET L = L + 1
1300 IF L > 0 AND INKEY$ = "7" THEN
1310 LET L = L - 1
1320 IF C < 31 AND INKEY$ = "8" THEN
1330 LET C = C + 1
1340 IF X <> L OR Y <> C THEN GOTO 4
1350
1360

```

Figura 196 (Continua)

```

000000 GOTO 4000
400000 FOR I=1 TO 7
401000 IF M(A,1)=L AND M(A,2)=0 TH
E NZ GOTO 4000
402000 NEXT I
403000 IF L=0 AND C=31 THEN GOTO 5
000000 GOTO 4000
404000 FOR I=1 TO 40
405000 PRINT M(I,I)
406000 NEXT I
407000 FOR I=1 TO 40
408000 PRINT M(I,I)
409000 NEXT I
410000 PRINT AT L,C: "*"
411000 PRINT AT 0,31: "*"
412000 PRINT AT 0,0: "*"
SALV000 ///

```

Figura 196 (Fine)

```

240 PRINT AT F,G: "*"

```

Figura 197

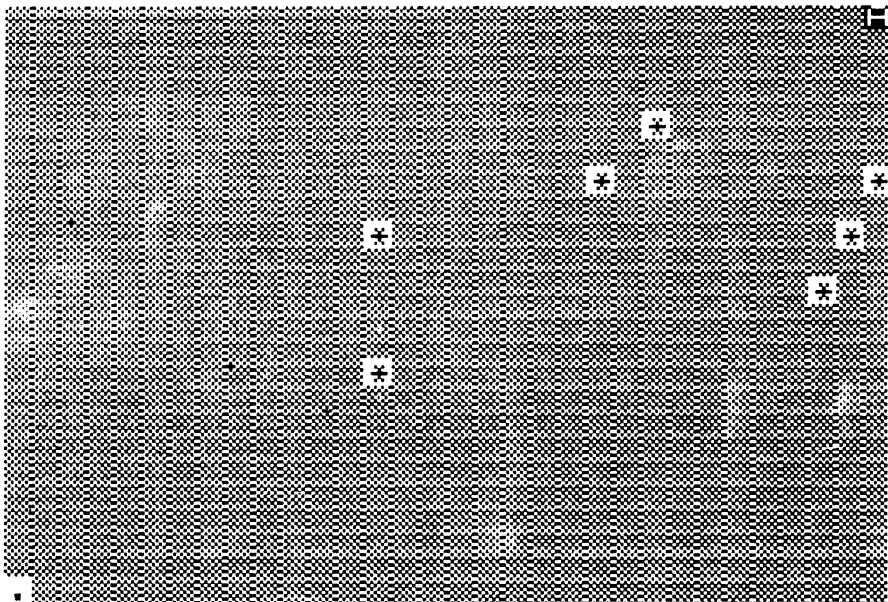


Figura 198

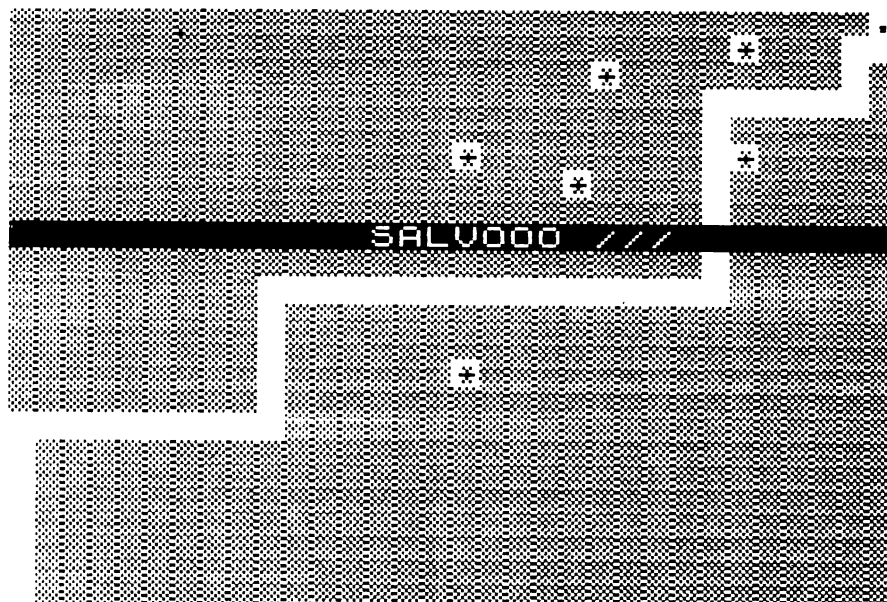


Figura 199

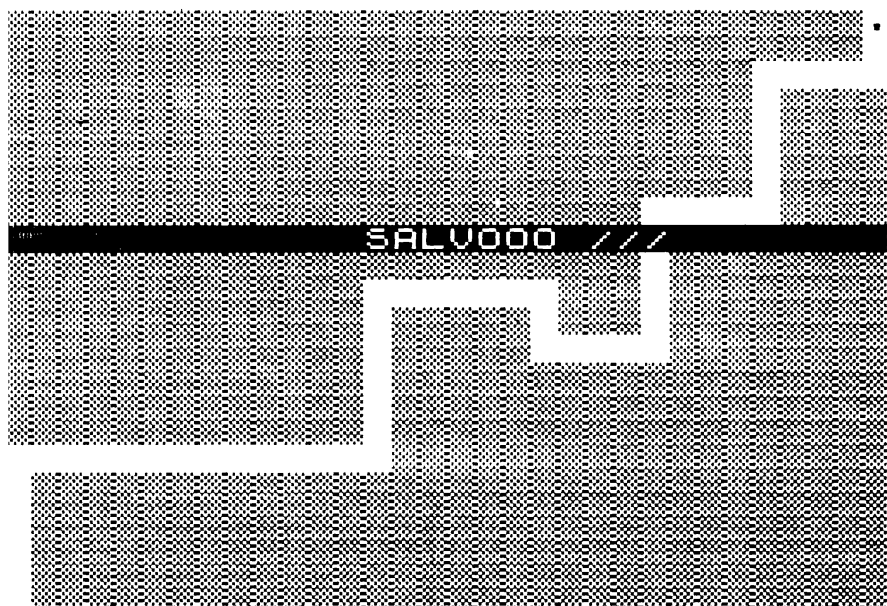


Figura 200

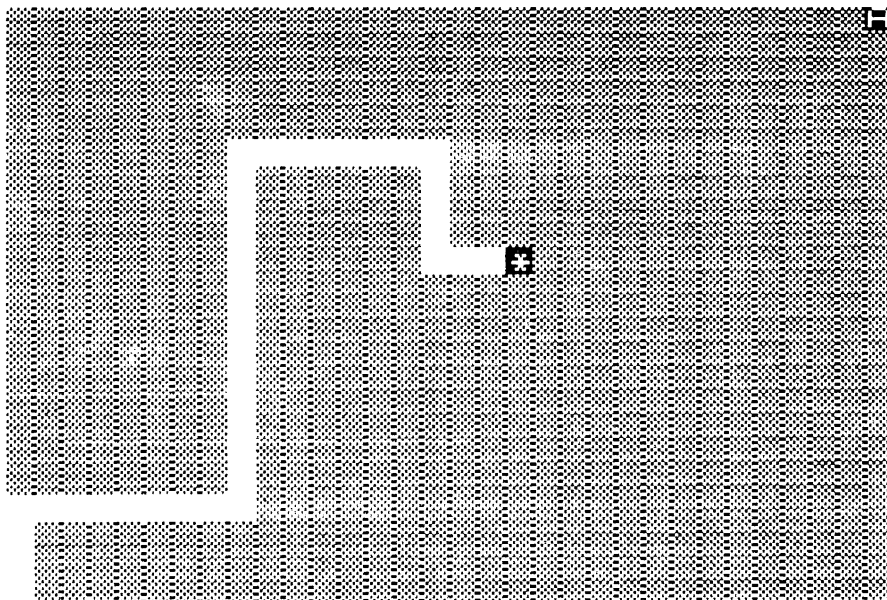


Figura 201

Aggiungendo la linea 245 (fig. 197) e facendo girare il programma si può avere una idea di come vengono disposte le mine sul campo (fig. 198). Se si sposta il punto lampeggiante dalla partenza all'arrivo evitando le mine come già detto appare la scritta "SALVOOO" (fig. 199).

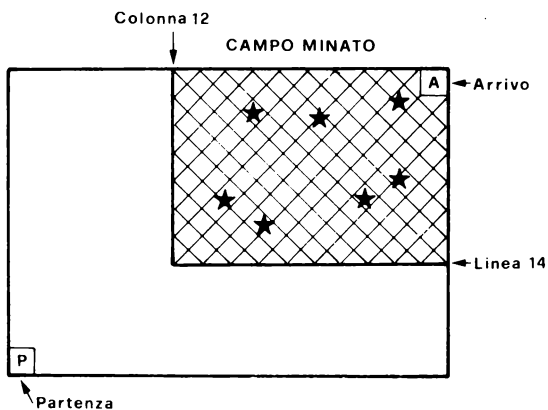


Figura 202

Naturalmente durante il gioco le mine devono essere nascoste per cui si deve prima togliere dal programma la linea 245.

Ora con le mine nascoste può capitare di terminare il gioco senza danno (fig. 200) oppure può succedere prima dell'arrivo di capitare su una delle mine (fig. 201). Se attraversando il campo si finisce su una mina il computer simula con il suono e con l'immagine una sorta di esplosione.

Il numero delle mine può essere aumentato o diminuito per rendere più difficile o più facile il gioco. Se per esempio si vogliono nascondere 12 mine invece di 7 occorre modificare il contenuto delle linee 190, 210 e 400 sostituendo, all'interno delle linee stesse, tutti i 7 con dei 12.

Così com'è, il programma nasconde le mine in una zona limitata prossima alla posizione di arrivo (fig. 202), se però si modifica il programma in modo da nascondere più mine (per esempio 15-20 mine) si può fare in modo che tali mine vengano sparse per tutto il campo e non in una zona limitata, togliendo dal programma la linea 240.

nota: La linea 120 contiene tra gli apici 8 caratteri grafici ottenuti premendo (con il computer in modo GRAPHIC) insieme al tasto SHIFT i tasti A H A H A H A H.

1—40 Dadi

**FAST
O SLOW**

Sfruttando al massimo la risoluzione grafica dello ZX81 si possono fare apparire sullo schermo fino a quaranta Dadi, usando il programma di fig. 203. Premendo RUN e NEWLINE il programma chiede innanzitutto quanti dadi devono uscire (fig. 204) quindi li disegna sullo schermo indicando alla fine il numero di Dadi ed il loro punteggio totale (fig. 205).

note: Le linee da 230 a 330 contengono tra gli apici 4 caratteri grafici ciascuna, ottenuti con il computer in modo GRAPHIC e premendo SHIFT. Qui di seguito viene data una lista dei tasti ai quali corrispondono i caratteri grafici da inserire tra gli apici nelle varie linee, in questa lista gli spazi inversi sono indicati con SPACE (da premersi sempre in modo GRAPHIC ma senza SHIFT):

| | |
|-----|------------------------|
| 23Ø | 3, 6, 6, 6 |
| 24Ø | 8, SPACE, SPACE, SPACE |
| 25Ø | 8, Q, SPACE, SPACE |
| 26Ø | 8, Q, SPACE, Q |
| 27Ø | 8, Q, Q, Q |
| 28Ø | 8, SPACE, Q, SPACE |
| 29Ø | 8, SPACE, SPACE, SPACE |
| 30Ø | 8, SPACE, SPACE, SPACE |
| 31Ø | 8, SPACE, SPACE, Q |
| 32Ø | 8, Q, SPACE, Q |
| 33Ø | 8, Q, Q, Q, |

```

100 REM      1--40 DADI
1100 SLOW
1110 PRINT "INSERIRE N° DADI ( 1-
-40 )
120 PRINT "E PREMERE NEWLINE"
130 INPUT N
140 IF N<1 OR N>40 THEN GOTO 13
0
150 CLS
160 LET K=0
170 LET Y=0
180 RAND
190 FOR L=0 TO 19 STEP 4
2000 FOR C=0 TO 31 STEP 4
2010 LET Y=Y+1
2030 LET X=1+INT (RAND*6)
2040 PRINT AT L,C;"███"
"███" IF X=1 THEN PRINT AT L+1,C;
250 IF X=2 OR X=3 THEN PRINT AT
L+1,C;"███"
260 IF X=4 OR X=5 THEN PRINT AT
L+1,C;"███"
"███" IF X=6 THEN PRINT AT L+1,C;
R 280 IF X=1 OR X=3 OR X=5 THEN P
R 290 IF X=2 OR X=4 OR X=6 THEN P
R 300 IF X=1 THEN PRINT AT L+3,C;
"███"
310 IF X=2 OR X=3 THEN PRINT AT
L+3,C;"███"
320 IF X=4 OR X=5 THEN PRINT AT
L+3,C;"███"
"███" IF X=6 THEN PRINT AT L+3,C;
340 LET K=K+X
350 IF Y=N THEN GOTO 380
360 NEXT C
370 NEXT L
380 PRINT
390 PRINT "      ";N;" DADI";"      P
UNTEGGIO = ";K

```

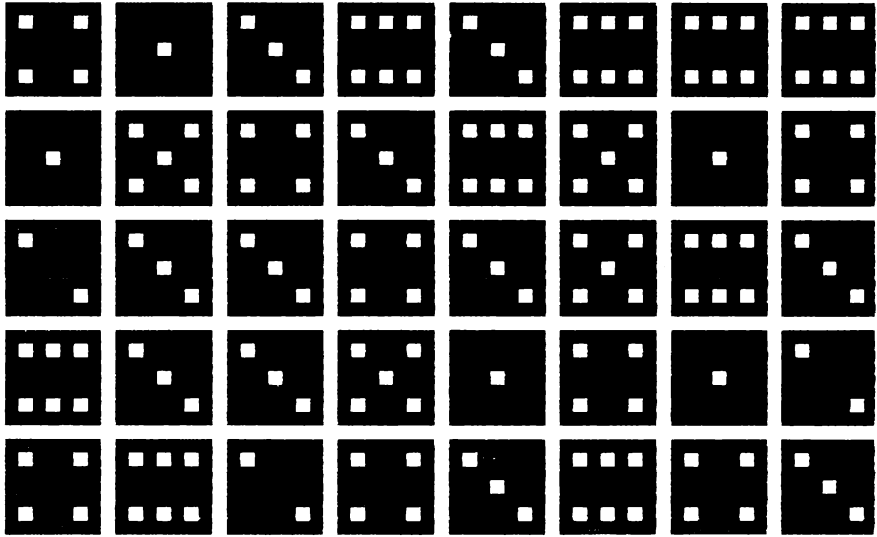
Figura 203

```

INSERIRE N° DADI ( 1--40 )
E PREMERE NEWLINE

```

Figura 204



40 DADI PUNTEGGIO = 150

Figura 205

Tavola di numeri e lettere

SLOW

La prima parte del programma di fig. 206 disegna sul video una tavola di 16 caselle di cui 15 occupate dai numeri da 0 a 9 più le lettere da A ad E. Subito dopo, con il

```

100 REM TAVOLA DI NUMERI
200 REM E LETTERE
300 SLOW
400 DIM D (4, 4)
500 LET D## (1) = "1234"
600 LET D## (2) = "5678"
700 LET D## (3) = "90AB"
800 LET D## (4) = "CDE"
900 PRINT D; 0, 10;
1000 PRINT D; 4, 10;
1100 PRINT D; 5, 10;
1200 PRINT D; 7, 10;
1300 PRINT D; 8, 10;

```

Figura 206 (Continua)

```

2000 FAST
2010 LET L=4
2020 LET C=4
2030 FOR M=1 TO 500
2040 LET X=1+INT (4*RND)
2050 IF C<4 AND X=1 THEN GOSUB 3
3000 IF L>1 AND X=2 THEN GOSUB 4
4000 IF L<4 AND X=3 THEN GOSUB 5
5000 IF C>1 AND X=4 THEN GOSUB 5
6000
3010 NEXT M
3020 SLOW
3030 IF C<4 AND INKEY$="5" THEN
GOSUB 3040
3040 IF L>1 AND INKEY$="6" THEN
GOSUB 4040
3050 IF L<4 AND INKEY$="7" THEN
GOSUB 5040
3060 IF C>1 AND INKEY$="8" THEN
GOSUB 6040
3070 GOTO 300
3080 PRINT AT L+3,C+13;" "
3090 LET A$(L,C)=A$(L,C+1)
4000 LET A$(L,C+1)=" "
4010 PRINT AT L+3,C+12;A$(L,C)
4020 LET C=C+1
4030 RETURN
4040 PRINT AT L+2,C+12;" "
4050 LET A$(L,C)=A$(L-1,C)
4060 LET A$(L-1,C)=" "
4070 PRINT AT L+3,C+12;A$(L,C)
4080 LET L=L-1
4090 RETURN
5000 PRINT AT L+4,C+12;" "
5010 LET A$(L,C)=A$(L+1,C)
5020 LET A$(L+1,C)=" "
5030 PRINT AT L+3,C+12;A$(L,C)
5040 LET L=L+1
5050 RETURN
5060 PRINT AT L+3,C+11;" "
5070 LET A$(L,C)=A$(L,C-1)
5080 LET A$(L,C-1)=" "
5090 PRINT AT L+3,C+12;A$(L,C)
6000 LET C=C-1
6010 RETURN

```

Figura 206 (Continua)

computer in FAST, le 15 caselle con le lettere e i numeri vengono mescolate per circa 45 secondi per poi apparire sullo schermo come in fig. 207.

A questo punto, usando i tasti 5, 6, 7, 8 si possono spostare le varie caselle con i numeri e le lettere per cercare di ritornare alla disposizione iniziale (fig. 208).

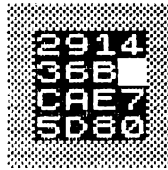


Figura 207

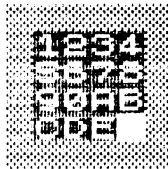


Figura 208

Se all'inizio del programma si vuole vedere in che modo il computer mescola le varie caselle si può sostituire la linea 220 FAST con la linea 220 SLOW, in questo modo però tale fase di mescolamento durerà circa tre minuti.

Tombola

SLOW

Il programma della tombola (fig. 209) per prima cosa disegna in nero sul video le 6 cartelle del tabellone e la scritta "Tombola".

```

10 REM      TOMBOLA
100 SLOW
110 PRINT AT 0,0;"TOMBOLA"
120 PRINT AT 1,0;"1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100"
130 FOR L=3 TO 21
140 IF NOT L=8 AND NOT L=9 AND NOT L=15 AND NOT L=16 THEN PRINT AT L,0;" "
150 NEXT L

```

Figura 209 (Continua)

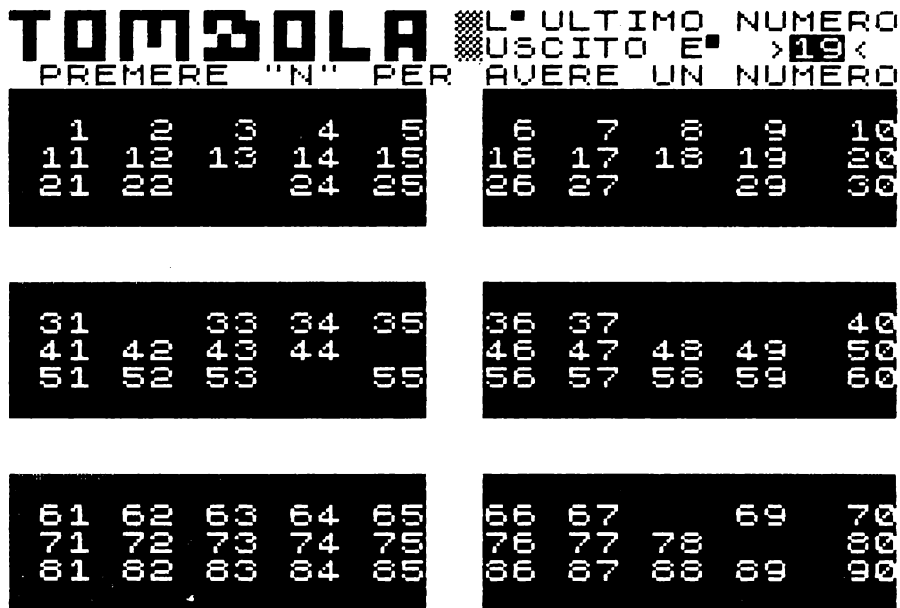


Figura 210

Ora ogni volta che si preme il tasto "N" il programma estrae uno dei numeri da 1 a 90 che non sia già uscito e lo sistema nella sua cartella, inoltre il numero appena uscito lampeggia nell'apposito riquadro nero posto in alto a destra.

È molto importante premere ogni volta il tasto "N" giusto il tempo necessario perchè nel piccolo riquadro lampeggiante appaia un nuovo numero.

L'aspetto del tabellone dopo l'uscita di molti numeri può essere come in fig. 210.

Se si è in molti a giocare si possono collegare più televisori all'uscita del computer e sistemarli in vari punti del tavolo.

Il programma può essere fermato usando il tasto BREAK.

Metal Detector (con sonoro)

SLOW

Il programma (fig. 211) disegna un campo e vi nasconde il tesoro in un punto scelto casualmente dal computer.

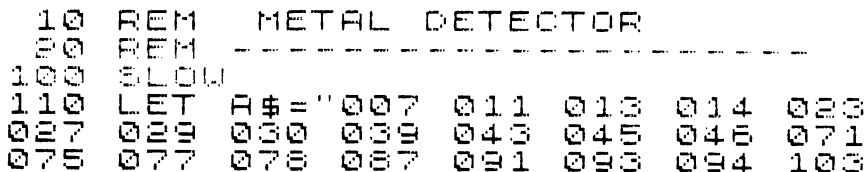


Figura 211 (Continua)

```

107 109 110 135 139 141 142 151
195 197 198 167 171 173 174 199
200 200"
RND) LET X=INT (15*RND)+INT (18*
100 LET Y=INT (12*RND)+INT (11*
RND)
140 IF X<8 AND Y>15 THEN GOTO 1
20
150 LET L=21
160 LET C=0
170 FOR S=1 TO 68
180 PRINT "███";
190 NEXT S
200 PRINT
210 LET M=0
220 LET T=15
230 PRINT "DT L,C)";
240 PRINT "DT L,15";
250 PRINT "DT L,C)";
260 IF C>0 AND INKEY$="5" THEN
L 270 C=C-1
280 IF L<21 AND INKEY$="6" THEN
LET L=L+1
300 IF L>0 AND INKEY$="7" THEN
L 310 L=L-1
320 IF C<31 AND INKEY$="8" THEN
LET C=C+1
330 IF CODE INKEY$>32 AND CODE
INKEY$<37 THEN GOTO 370
340 IF L=Y AND C=X THEN GOTO 34
0
350 POKe 0000,F
360 GOTO 390
370 PRINT "DT Y,X)";
380 POKe 0000,F,X);
0 390 PRINT "DT 0,0)"; "TESORO TROVAT
O 400 PRINT "███";
410 POKe 0000,15
420 STOP
430 LET M=M+1
440 LET J=53-ABS (Y-L)-ABS (C-X)
)
450 IF J>13 THEN LET J=J-14
460 LET P=VAL A$(J*4-3 TO J*4-1
)
470 GOTO 310

```

Figura 211 (Fine)

TESORO TROVATO IN 54 MOSSE



Figura 212

Il giocatore dovrà spostare, tramite i tasti 5, 6, 7, 8, il punto lampeggiante in basso a sinistra fino al punto dove è nascosto il tesoro.

Come in un vero Rivelatore di Metalli la frequenza del suono emessa dal computer aumenterà diventando sempre più acuta quando il punto lampeggiante si avvicina al tesoro nascosto mentre diminuirà se il punto lampeggiante viene spostato nella direzione sbagliata.

Appena il punto lampeggiante passerà sul tesoro apparirà il simbolo del dollaro ed il computer indicherà con una scritta in alto sul video in quanti spostamenti il giocatore è riuscito a trovarlo (fig. 212).

Segnatempo (con sonoro)

FAST

Questo programma (fig. 213) non è propriamente un gioco ma può essere utilizzato in molti giochi con limite di tempo per scandire il passare dei secondi.

```
100 REM  SEGNATEMPO
110 FAST
120 PRINT "INSERIRE INTERVALLO
TRA"
130 PRINT "I SEGNALI ( 1--30 S
ED. ) "
```

Figura 213 (Continua)

```

140 INPUT S
150 IF S<1 OR S>30 THEN GOTO 14
160 LET T=S*50-2
170 POKE 00000,7
180 POKE 00000,15
190 POKE 00000,222
200 POKE 00000,15
240 GOTO 100

```

Figura 213 (Fine)

Il programma emette un suono basso per un secondo, appena tale suono sparisce il computer comincia ad alternare delle pause e dei brevi BIP sonori.

Il tempo che intercorre tra un BIP e l'altro può essere scelto tra 1 e 30 secondi e va introdotto all'inizio quando il programma lo richiede.

Lotto

SLOW

Il programma di fig. 214 estrae dei numeri casuali da giocare al Lotto.

Per prima cosa il programma chiede su quali Ruote si vuole effettuare la Giocata

```

10 REM *****
20 INPUT "SCELTA" S
30 FOR I=1 TO 10
40 LET A$(I)=RND*100
50 PRINT I;"RUOTA" A$(I)
60 NEXT I
70 PRINT "SCELTA" S
80 PRINT "I TASTI RELATIVI ALL"
90 PRINT "CHITTA"
100 FOR L=5 TO 14
110 PRINT AT L,L;CHR$(L+151);"
120 :A$(L-4)

```

Figura 214 (Continua)

```

0000 NEXT L
0000 PRINT L
T 0000 CONTINUE AT 18,3;"PREMERE <C>"
0000 DIM INKEY$(1,10)
0000 LET N=#(1,10)
#(1,N+1)=0 AND INKEY#=-98
0000 IF N>=0 AND N<10 THEN LET H
0000 IF N>=0 AND N<10 THEN PRINT
0010 IF N+5=10; "4" THEN GOTO 330
0000 GOT O 000
0000 CLS
0040 PRINT "SCEGLIERE LA GIOCATTA"
:
0000 PRINT
0000 PRINT
0000 PRINT "          INSERIRE          PE"
P 0000 PRINT "-----"
- - - - -
0000 PRINT
0400 PRINT TAB 7;"1";TAB 18;"NUM"
E RI SINGOLI"
0410 PRINT
0420 PRINT TAB 7;"2";TAB 18;"AMB"
H "
0430 PRINT
0440 PRINT TAB 7;"3";TAB 18;"TER"
N H "
0450 PRINT
0460 PRINT TAB 7;"4";TAB 18;"QUA"
T M P Z E "
0470 PRINT
0480 PRINT TAB 7;"5";TAB 18;"CIN"
Q U C H Z E "
0490 PRINT AT 19,5;"POI PREMERE"
N E U C L I N E "
0500 INPUT G
0510 IF G<1 OR G>5 THEN GOTO 500
0520 CLS
0530 DIM X(5)
0540 LET P=0
0550 FOR T=1 TO 10
0560 IF H$(1,T)="*" THEN GOSUB 6
0570 NEXT T
0580 STOP
0590 FOR K=1 TO 5

```

Figura 214 (Continua)

```

610 LET X(K)=1+INT (90*RND)
620 NEXT K
630 IF X(1)=X(2) OR X(1)=X(3) OR
R X(1)=X(4) OR X(1)=X(5) OR X(2)
=X(3) OR X(2)=X(4) OR X(2)=X(5)
OR X(3)=X(4) OR X(3)=X(5) OR X(4
)=X(5) THEN GOTO 600
640 PRINT AT P,4;A$(T)
650 FOR D=1 TO 6
660 PRINT AT P,14+D*3;X(D)
670 NEXT D
680 LET P=P+2
690 RETURN

```

Figura 214 (Fine)

SCEGLIERE LE RUOTE PREMENDO
I TASTI RELATIVI ALLE CITTÀ.



PREMERE <0> PER CONTINUARE

Figura 215

(fig. 215), la scelta delle città viene effettuata premendo i corrispondenti numeri da 1 a 0 (la prima fila di tasti del computer), quindi si preme il tasto "C" per passare alla scelta della Giocata (fig. 216). Tale scelta viene effettuata inserendo un numero da 1 a 5 (per avere un numero singolo, un ambo, ecc.) e premendo poi NEWLINE. A questo punto appaiono sullo schermo le città con a fianco i numeri scelti casualmente dal computer.

La fig. 217 mostra un esempio di risultato ottenuto selezionando tutte le Ruote e scegliendo la cinquina come Giocata.

SCEGLIERE LA GIOCATATA

| INSERIRE | PER AVERE |
|----------|----------------|
| 1 | NUMERI SINGOLI |
| 2 | AMBI |
| 3 | TERNI |
| 4 | QUATERNE |
| 5 | CINQUINE |

POI PREMERE NEWLINE

Figura 216

| | | | | | |
|-----------------|----|----|----|----|----|
| BARI | 1 | 6 | 40 | 72 | 25 |
| CAGLIARI | 18 | 19 | 13 | 63 | 7 |
| FIRENZE | 50 | 65 | 54 | 51 | 70 |
| GENOVA | 47 | 56 | 66 | 62 | 3 |
| MILANO | 71 | 14 | 32 | 16 | 68 |
| NAPOLI | 68 | 52 | 79 | 23 | 7 |
| PALERMO | 70 | 17 | 51 | 38 | 55 |
| ROMA | 63 | 90 | 60 | 31 | 25 |
| TORINO | 55 | 47 | 35 | 57 | 33 |
| VENEZIA | 36 | 36 | 62 | 51 | 33 |

Figura 217

Totocalcio

FAST O
SLOW

L'ultimo programma di questo libro (fig. 218) visualizza sullo schermo e stampa alla fine sulla stampante delle colonne di risultati da giocare al Totocalcio. Il

```

100010 REM TOTOALCALCIO
100020 REM GENERAZIONE DI
100030 REM COLONNE CASUALI
100040 SLOW
100050 PRINT "QUANTE COLONNE ? (
100060 )"
100070 INPUT N
100080 IF N<1 OR N>16 THEN GOTO 12
0
100090 CLS
100100 PRINT "INSERIRE PERCENTUALE
DI USCITA"
100110 PRINT "DEGLI <1> IN % (
100120 )"
100130 INPUT P1
100140 IF P1<0 OR P1>100 THEN GOTO
100150
100160 CLS
100170 PRINT "INSERIRE PERCENTUALE
DI USCITA"
100180 PRINT "DEGLI <X> IN % (
100190 )"
100200 INPUT PX
100210 IF PX<0 OR PX>(100-P1) THEN
GOTO 0
100220 CLS
100230 PRINT " 1 - ";P1;" % X = "
100240 PRINT " 2 - ";100-(P1+PX);" % "
100250 PRINT
100260 PRINT
100270 FOR L=1 TO 13
100280 FOR C=1 TO N
100290 LET X=1+INT (100*RND)
100300 IF X<P1+1 THEN PRINT "1 ";
100310 IF X>P1 AND X<(P1+PX+1) THE
PRINT "X";
100320 IF X>(P1+PX) THEN PRINT "2
100330
100340 NEXT C
100350 PRINT
100360 IF L=3 OR L=6 OR L=9 THEN P
PRINT
100370 NEXT L
100380 COPY

```

Figura 218

numero di colonne desiderato (che può essere da 1 a 16) va introdotto all'inizio del programma (fig. 219).

La scelta dei risultati tra 1, X e 2 è parzialmente casuale poiché il programma prevede all'inizio l'introduzione della percentuale di uscita degli "1" (fig. 220) e degli "X" (fig. 221) che si ritiene più probabile, se la percentuale totale degli "1" e

```
QUANTE COLONNE ? ( 1--16 )
```

Figura 219

```
INSERIRE PERCENTUALE DI USCITA  
DEGLI <1> IN ■/■ ( 0--100 )
```

Figura 220

```
INSERIRE PERCENTUALE DI USCITA  
DEGLI <X> IN ■/■ ( 0--50 )
```

Figura 221

```
1 ; 50 ■/■      X ; 30 ■/■      2 ; 20 ■/■

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
X X X X X X X X X X X X X X X X
2 2 1 X 1 1 X 2 1 X 1 1 2 1 2 1

X X 1 1 1 X 1 X X X 1 X 2 1 1 X
1 X 1 2 2 X 2 1 X X X X X 1 X 1 1
2 1 1 X 1 X X X 1 X X 1 X 2 X X
2 X X 2 1 X 1 X X 2 X 1 1 2 1 1

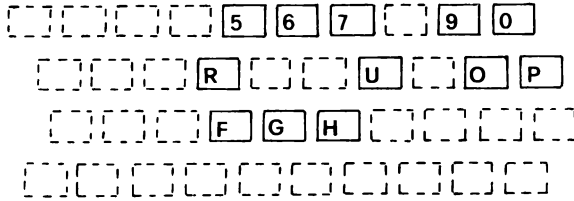
1 1 1 1 1 1 1 1 1 1 1 1 1 1 X X X
1 1 1 1 X 1 X 1 X X X 1 1 1 1 1
2 1 X X 1 X X 1 X X 2 X 1 X X X
X 1 2 2 X X X X X 2 X 1 X X 1 1 1
```

Figura 222

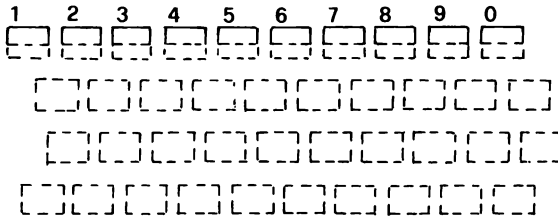
degli "X" e inferiore al 100% la parte rimanente rappresenta la percentuale di uscita del "2".

La fig. 222 mostra un esempio di risultato con 16 colonne e con in alto le percentuali di 1, X, 2, introdotte.

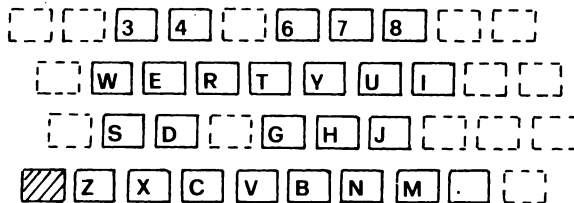
Dove collocare le tastiere allegate



Corrispondenza tra la Tastiera 1 e i tasti dello ZX81



Corrispondenza tra la Tastiera 2 e i tasti dello ZX81



Corrispondenza tra la Tastiera 3 e i tasti dello ZX81

Chi possiede uno ZX81 o anche uno ZX80 con nuova ROM, troverà senz'altro in questo libro molte cose utili.

Innanzitutto 66 programmi che coprono varie applicazioni e le cui principali caratteristiche sono soprattutto il largo uso delle capacità grafiche e di animazione dello ZX81 e (molto importante) la sicurezza di funzionamento dei programmi non essendo stati essi ricopiati dal video ma stampati direttamente dal computer (subito dopo avere girato con successo) con una speciale stampante ad alta qualità, costruita appositamente per questo libro.

Vi sono anche alcuni circuiti collegabili ai due ZX: un alimentatore tampone, un avvisatore acustico per la tastiera e, soprattutto, un'interfaccia per collegare gli ZX ad infinite apparecchiature esterne, oltre ad una scheda musicale che genera ben 50 note su 4 ottave.

E il meglio, in particolare degli ultimi tre circuiti detti, è che possono essere costruiti con POCHE MIGLIAIA di lire ciascuno!

Non solo ma vengono dati, per l'interfaccia e la scheda musicale, anche numerosi programmi applicativi tra cui, specialmente per quest'ultima, molti effetti sonori ed alcuni giochi che sono, probabilmente, gli unici programmi con sonoro disponibili a tutt'oggi per i computers Sinclair.

Ed ancora, altra novità, troverete alcune tastiere disegnate da sovrapporre alla tastiera del Sinclair in alcune applicazioni speciali.

Dello stesso autore:

GLI AMPLIFICATORI DI NORTON QUADRUPLI LM3900 & LM359 con esperimenti.



GAETANO MAFRANO

66

PROGRAMMI PER

ZX81 E ZX80

66

CON UN ROM +

HARDWARE

GRUPPO EDITORIALE JACKSON

