

Sinclair COMPUTER

Mensile per gli utenti di ZX 81 e Spectrum

#2

L. 2.500

Super monitor

Slalom

Prova: GP 100 A e Alphacom 32

Trasferimento pagina video

Studio d'una funzione



SOFTWARE !!

DISTRIBUITO DA

REBIT COMPUTER

sinclair

PSION 



QUICKSILVA

© RABBIT SOFTWARE



SUPERSOFT



A&F software

ANIROG



ATARISOFT

Audiogenic



Datasoft®



HISOFT PASCAL
DEVPAC



HesWare



HEWSON CULSANT



INTERCEPTOR
MICRO'S ©

LLAMASOFT!!

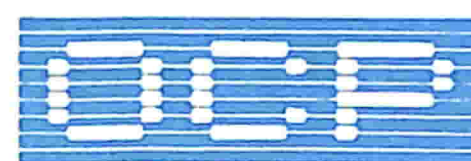
MIKRO-GEN



MELBOURNE

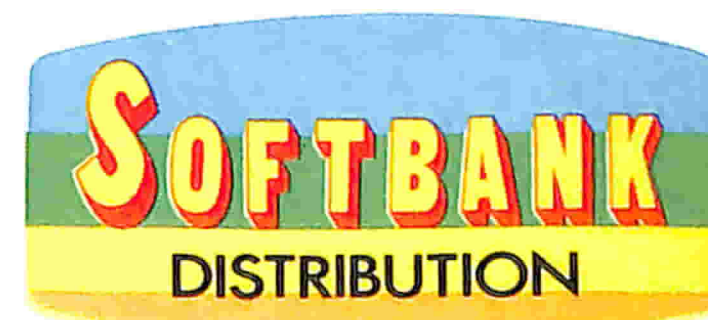


PICTURESQUE



**UNA VASTA
GAMMA
DI PROGRAMMI,
DI GIOCHI
E DI UTILITÀ**

DISTRIBUITI DA:

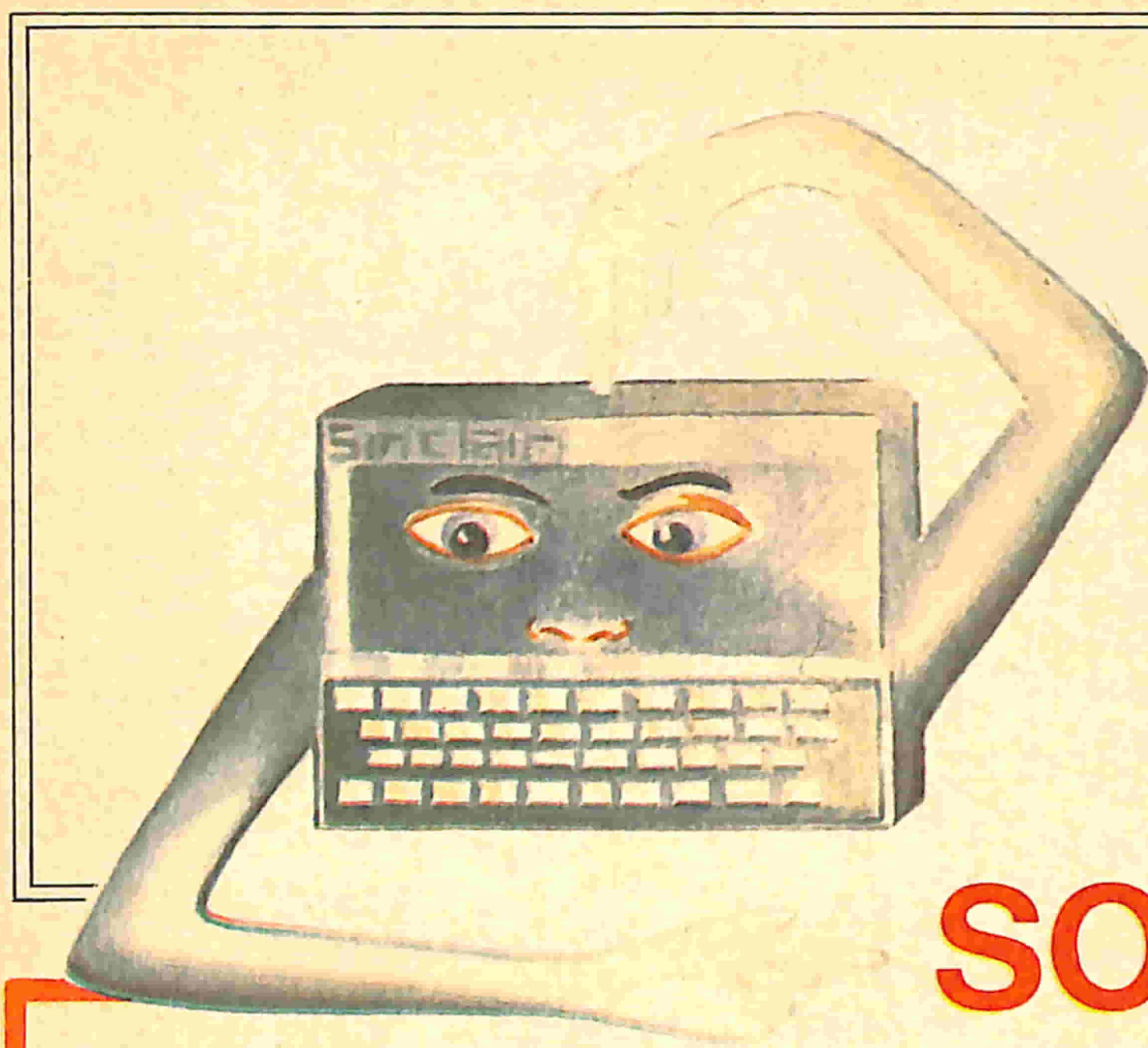


DIVISION OF G.B.C.



Magazzino

Sinclair COMPUTER



SOMMARIO

- | | | |
|--|--|--|
| 4 Sinclairamente vostro | 20 Slalom | 37 La ROM dello ZX81 (seconda parte) |
| 6 Prova su strada: stampanti Alphacom 32 e GP 100 A | 22 Studio d'una funzione | 45 Sinclair-Parade |
| 10 Sotto il lenzuolo dello Spectrum (seconda parte) | 29 Super monitor | 47 Sinclairama: QL, un computer che farà strada |
| 14 La strada della dannazione | 31 Trasferimento della pagina video | |
| 17 Almanacchi, almanacchi nuovi | 33 Dump, o la bella variabilina | |

DIRETTORE RESPONSABILE
Agostina Ronchetti

COORDINAMENTO TECNICO
Mauro Soldavini

REDAZIONE
Lorenzo Bossi
Franco Raspa

GRAFICA, IMPAGINAZIONE E DISEGNI
Renato Caruso
Francesco Amatori

FOTO
Franco Vignati

DIFFUSIONE E ABBONAMENTI
Marina

DIREZIONE, REDAZIONE

Via Cristoforo Colombo, 49
20090 Trezzano/Milano - Tel. 02/8466675

Autorizzazione del Tribunale di Milano
n. 522 del 12.11.83

STAMPA:
Pizzorni (Cremona)

Concessionario esclusivo per la diffusione:
MEPE spa - Via G. Carcano, 32 - Milano

Spedizione in abbonamento postale
Gruppo III/70

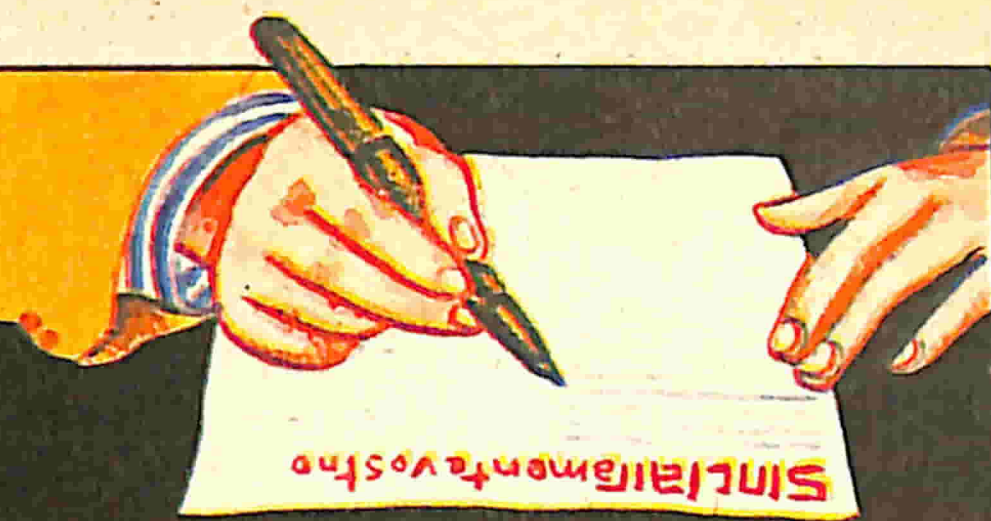
Prezzo della rivista L. 2.500
Numero arretrato L. 5.000

Abbonamento annuo L. 20.000.

I versamenti vanno indirizzati a: Sinclair Computer - V.le Famagosta, 75 - 20142 Milano - mediante emissione di assegno bancario, cartolina-vaglia o utilizzando il c/c postale n.ro 3042609

Per i cambi di indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500 anche in francobolli.

TUTTI I DIRITTI DI RIPRODUZIONE
O TRADUZIONE DEGLI ARTICOLI
PUBBLICATI SONO RISERVATI.



LE lettere in risposta al primo numero della rivista sono arrivate velocissime, tanto che possiamo iniziare a rispondere già in questo secondo numero. La cosa ci ha fatto naturalmente molto piacere: segno che la rivista era attesa.

Non sono mancate alcune critiche negative, ma soprattutto costruttive, con suggerimenti su che cosa cambiare e con i desideri dei lettori: è quello che aspettiamo. Continuate a scrivere e fateci sapere che cosa vorreste vedere pubblicato; anzi, mandatecelo.

Riguardo alla qualità dei listati, abbiamo provveduto subito, e già tutti i programmi di questo numero sono eseguiti con una stampante migliore (di cui pubblichiamo anche la prova), che dovrebbe aver risolto i problemi di leggibilità; per i prossimi numeri contiamo di migliorare ancora.

Una risposta a tutti coloro che chiedono più spazio per lo Spectrum: le pagine dedicate al software saranno di più e meglio curate (con la vostra collaborazione), ma tenete in considerazione che la prima generazione di Sinclairisti - quelli con lo ZX81 - è ben lontana dal fare la fine dei dinosauri, e non possiamo tirarci addosso le sue giuste ire trascurandola.

Per le trasposizioni da ZX81 a Spectrum di un programma, stiamo studiando come fare perchè sia facilmente comprensibile senza raddoppiare i listati; l'operazione inversa non è quasi mai possibile senza radicali rifacimenti. Se però farete qualcuna di queste traduzioni per programmi che avremo pubblicato, inviatecela.

Didattica e telescopi

Desidero conoscere di esperienze didattiche di utilizzazione dello Spectrum, con particolare riferimento alle applicazioni linguistiche. (Bruno Manfredi - Torino)

Vorrei chiedere se esiste un programma per lo Spectrum che possa tradurre l'inglese in italiano e uno che dia le coordinate celesti per puntare il telescopio sulle costellazioni principali e sui pianeti, nei prossimi anni. (Guglielmo Di Filippo - Isola G.S. TE).

- Alcuni programmi di tipo didattico esistono e riguardano soprattutto materie tecniche, per esempio, risoluzione di problemi di matematica o fisica, anche se dobbiamo rilevare che finora questo genere di applicazioni è volto a fornire soluzioni pronte, piuttosto che supportare un vero training.

Per "applicazioni linguistiche" creiamo che si intenda la stessa cosa della seconda lettera - ausilio alla traduzione - e la risposta è negativa: per tradurre da una lingua a un'altra occorre dotare il computer di un vocabolario più o meno completo (per non parlare di idiomi, gerghi di settore, frasi fatte, etc.), cosa che richiede Megabytes a pacchi.

Infine per i problemi astronomici, sappiamo che questi programmi esistono per altri computer; se qualche lettore ne ha sviluppati per i Sinclair, è invitato a contattare la redazione. Nel frattempo, possiamo solo indicare un libro ottimo, ma purtroppo non facile da trovare: Jean Meeus "Astronomical formulae for calculators", edito a Bruxelles nel 1979.

Un connettore difficile

Vorrei un consiglio su come potermi procurare dei connettori per lo Spectrum. Mi rivolgo a voi essendo risultata vana ogni ricerca. (Alberto Galanti - Vicenza).

- Il connettore posteriore dello Spectrum (come quello dello ZX81) è ricavato direttamente sulla scheda e non ha nessuna caratteristica strana: occorre un connettore femmina per circuito stampato a doppia fila, da 28+28 contatti (23+23 per lo ZX81). Qualsiasi negozio di componenti elettronici ne dispone. In genere sono più lunghi e vanno tagliati su misura.

La linea zero

Ho trovato in alcuni programmi l'istruzione n.0, che risulta anche non cancellabile con "0 < ENTER >". Come si fa a cancellarla? Come si fa a introdurla in un programma senza che compaia "Nonsense in BASIC". (Carlo Di Ciolo - Pisa)

- Una linea di programma con numero zero viene introdotta proprio per preservare accidentali cancellazioni, e contiene nella maggior parte dei casi, un'istruzione REM, con un testo (Copyright, nome dell'autore, istruzioni particolari...) o con una routine in linguaggio macchina.

Per ottenerla non si può scrivere direttamente "0 REM...", procedura rifiutata dal sistema. Occorre conoscere l'indirizzo di inizio dell'area basic, e più precisamente la locazione che contiene il byte basso del primo numero

Per questa rubrica scrivi a:
Sinclair Computer
c/o Sigma Informatica
v. Cristoforo Colombo, 49
20090 Trezzano
Milano

di linea: queste locazioni sono: 16510 per lo ZX81 e 23756 per lo Spectrum. Si scrive la linea di programma con un numero qualsiasi (inferiore a 255), quindi si esegue POKE n,0 usando il numero relativo al computer che si ha davanti. Per cancellarla si ripete POKE n,x - dove x è un numero inferiore a quello della seconda linea.

Cobol?

Desidererei sapere se esiste in commercio un compilatore o interprete del linguaggio Cobol per lo Spectrum e dove posso trovare la relativa cassetta nella mia provincia. (Stefano Dal Lago - Valdarno VI)

- Il Cobol (COMmon BUiness Oriented Language) è un linguaggio risalente al 1959 e destinato ad applicazioni commerciali sui grandi calcolatori. Non ci risulta che abbia mai avuto trasposizioni su hobby computer, operazione non impossibile ma certo ardua: la colloquialità - uno dei pregi del Cobol - rende un compilatore piuttosto ingombrante.

Espansioni RAM

Avendo uno Spectrum da 16K, ho intenzione di espanderlo a 48K, e vorrei sapere: 1/ tra le espansioni in commercio qual è la più consigliata e i prezzi; 2/ se gli integrati vanno semplicemente montati; 3/ se vi è anche un supporto software. (Leonardo Colucci - Napoli)

- Sulla scheda dello Spectrum vi sono 11 zoccoli vuoti, in cui si inseriscono i vari chip secondo le indicazioni della casa madre: non occorre nessun software. Riguardo ai chip di RAM, può cambiare la marca, ma non la sostanza: occorrono memorie da 32K. Si possono però montare anche quelle da 64K (con una di queste sigle: 4164, 4864, 6665, o equivalenti), più facili da trovare. La memoria in più può essere raggiunta con un trucchetto di cui parleremo nel prossimo numero.

Ho uno ZX81 16K. Molti listati, denunciati per 1K, non li accetta per mancanza di memoria. Quando inserisco la 16K lo schermo diventa tutto ondulante, rendendo impossibile la lettura. Il programma però lo accetta, perchè si vedono macchie nere che si spostano in alto dando il N/L. Ho provato alimentatori da 6A, ma il difetto rimane. (Ettore Bosio - Asti)

- Dando (speriamo!) per acquisito che si inserisce prima l'espansione e poi l'alimentazione, il malfunzionamento potrebbe risiedere nella circuitazione della RAM, ma a distanza non possiamo dire di più. Provi a sostituire la 16K (chiedendola a qualche collega).

Tutti quei numeri

Sono un neofita non troppo giovane (40 anni) dello Spectrum e fra le tante cose che non ho capito si è aggiunta la pagina che avete dedicato alla ROM dello ZX81: tutti quei numeri non mi dicono nulla. Ho tentato con un solo FOR/NEXT di far muovere nei due sensi orizzontali un carattere ma non ci sono riuscito; o meglio se schiaccio il carattere che dovrebbe invertire, il carattere continua fino alla fine e poi si inverte. (Dino Busso - Somma L. VA)

- A più d'uno i listati della ROM ZX81 saranno sembrati solo file di numeri;

ma abbiamo avuto delle richieste e dobbiamo cercare di dare qualcosa sia ai neofiti che agli esperti. Precisiamo comunque che, essendo tutto il disassemblato piuttosto lungo, qualche testo esplicativo verrà pubblicato dopo le prime puntate.

Per l'altro problema, le indicazioni sono un po' vaghe per poterla aiutare: di che programma si tratta? Ci mandi un listato.

Protezioni e copiatore

Ho caricato il programma per leggere l'header pubblicato su Sinclair Computer n. 1. Ho però potuto notare che alcuni programmi hanno l'autostart anche essendo scritti tutti in linguaggio macchina. In questo modo non posso dare il comando SAVE per copiarli, in quando immuni al BREAK. Come possa fare? (Fabrizio B. Cordero - Baldissero C. TO)

- Anche se molto diplomaticamente, il lettore ci sta chiedendo come proteggere un programma, per farne una copia. Tecnicamente la protezione perfetta non è ancora stata inventata, ma non tutte le cassette hanno gli stessi accorgimenti: bisogna andare un po' per tentativi e non c'è un metodo univoco. A parte ciò, ci sono evidenti ragioni di mercato che ci impedirebbero di rispondere... almeno in questa sede.

Lo ZX81 e i "Data"

E' possibile utilizzare i "Data" sullo ZX81? (Marco Seidita - Rivolta TO)

- Il basic dello ZX81 non prevede le istruzioni DATA, READ, RESTORE. Per leggere da programma una serie di

```
10 REM ** DATA **
20 LET A$="001002003004005006007008009010"
30 REM ** READ **
40 FOR N=1 TO 30 STEP 3
50 LET X=VAL A$(N TO N+2)
60 PRINT X
70 NEXT N
```

numeri o di stringhe sono stati suggeriti vari artifici: l'esempio del listato è uno dei più usati. Si scrivono i numeri, in questo caso gli interi da 1 a 10, in una stringa, aggiungendo degli zeri in modo che siano tutti della stessa lunghezza; ogni volta che serve un dato, il programma legge una sezione di stringa con la linea xy, che simula un READ.

“PROVE SU STRADA”

due Sinclair, due stampanti, interfacce, cavi e software

di Maubat

Già dagli ormai storici tempi dello ZX80 i swaristi più esigenti avevano rivolto accorati appelli ai “fratelli” hardwaristi: si era manifestato subito il desiderio di mettere i computer Sinclair in grado di spedire efficacemente treni di bit a una stampante degna di questo nome.

Intendiamoci, onore al merito della ZX-Printer: a quel prezzo nessuno pretende di più. Ma le prestazioni già dello ZX81, e a maggior ragione dello Spectrum, meritano di meglio.

Un meglio che oggi c'è: il mercato è cresciuto in tutte le direzioni e gli accessori, hard e soft, occhieggiati e invidiati per mesi sulle riviste anglosassoni, sono finalmente cose normali anche da noi (pecche di distribuzione a parte: spesso gli amici sinclairisti del Sud sono costretti ad affidarsi alle lente vie postali del Nord).

Come che sia, tra mercato ufficiale, quasi ufficiale, ufficioso, pirata, clandestino, replicante e alieno, il materiale gira, e questo è sicuramente positivo.

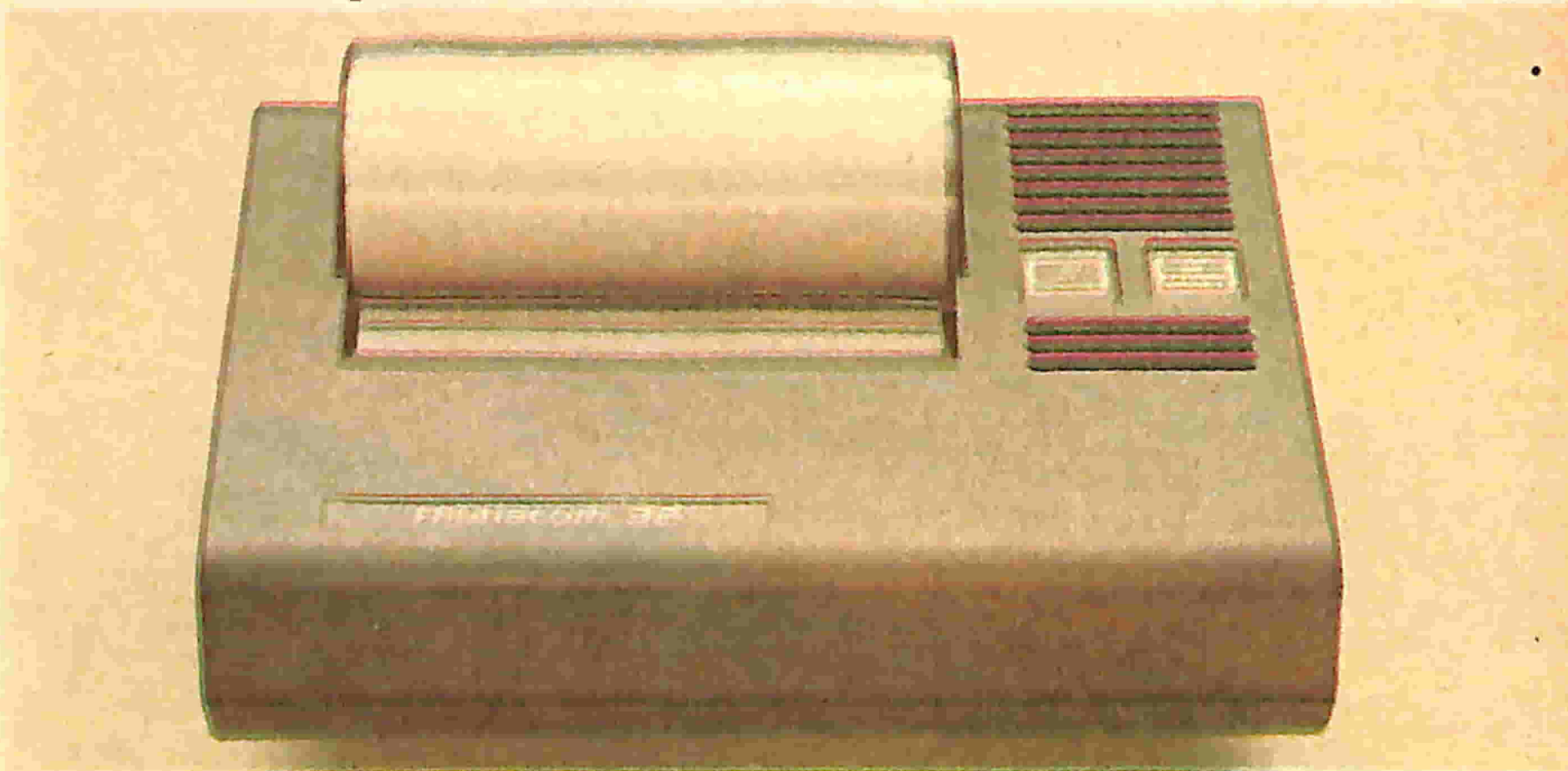
Due stampanti, due filosofie

L'output per la stampa nei

Sinclair non risponde a nessuno standard: per mettere i computer in grado di gestire una stampante ci sono due possibilità: costruire la stampante stessa in adeguamento agli ZX (se la montagna non va a Maometto...), oppure prendere una stampante di serie e studiare l'interfaccia (ed il software) necessaria alle conversioni. Delle due stampanti provate una,

ta progettata espressamente per i prodotti Sinclair, ai quali si ispira vagamente anche nello stile: molto compatta (solo 20x15 cm), di materiale plastico nero e ruvido, simile a quello che carrozza lo Spectrum.

Non necessita di interfaccia: il connettore si inserisce direttamente sulla porta di espansione (ahimè, un altro cavo corto 15



L'Alphacom 32 si presenta così.

l'Alphacom 32, ha seguito la prima strada; l'altra è una Seikosha GP-100A, strettamente di serie, con un'adeguato interfacciamento.

Alphacom 32

Di produzione, manco a dirlo, orientale (Hong Kong), anche se di casa madre californiana, è sta-

cm scarsi!).

A differenza della ZX-Printer, però, l'Alphacom 32 ha un suo alimentatore, richiedendo ben 24 V di tensione e un discreto assorbimento (ca. 1A). Onde evitare errori di collegamento, che potrebbero risultare fatali per il computer, l'alimentatore della stampante è saggiamente dotato di un jack femmina. A proposito

della ZX-Printer, sono in molti coloro che lamentano lo scadente funzionamento dell'accessorio con lo Spectrum, soprattutto a 48K: si tratta, principalmente, di un problema di sottoalimentazione: lo ZX-Power Supply, nella versione più potente, riesce a soddisfare una richiesta di 1.2 - 1.3A, che per Spectrum 48K + Printer non sono sufficienti.

L'Alphacom 32, appunto, non ha di questi problemi.

Una volta collegata funziona esattamente come la ZX-Printer, rispondendo alle tre funzioni LPRINT, LIST e COPY. Sulla sinistra del rotolo di carta vi sono due pulsanti: "OFF" e "ON/ADVANCE". Questo va tenuto premuto qualche istante all'accensione per abilitare la stampa; serve poi per l'inserimento del rotolo nuovo di carta (per l'avanzamento è più semplice tirare).

Durante la stampa comandata da programma, OFF funziona come un BREAK: disabilita la stampante - per riattivare si preme ON -, ma senza spegnerla. In-

La lunghezza del cavo non permette di allontanare troppo l'Alphacom dallo Spectrum.



Il connettore dell'Alphacom consente l'inserimento di altre periferiche.

fatti il dissipatore continua a scaldare; dopo un po' sentirete un discreto calore uscire dalle feritoie, ma non preoccupatevi: tutti i componenti sensibili sono lontani, e potete tenerla accesa a lungo senza rischi.

La carta è termica bianca, con stampa in blu o in nero (dipende dalla marca che trovate). I rotoli da 20-25 metri trovano posto in un alloggiamento del coperchio, a sua volta protetto da un elegante plasticone fumée. In questo formato (4 pollici e 1/4, cioè 105-110 mm) si trovano anche

rotoli più grossi, di circa 100 metri (40 piani di nitidezza...), che possono risultare più convenienti se siete bravi a costruirvi un trabiccolo portacarta esterno.

Vediamo infine come stampa, questa scatoletta: è stata una gradevole sorpresa, perchè la qualità di scrittura è ottima, molto nitida e costante nella resa. Per la cronaca, quasi tutti i listati della rivista vengono ora prodotti con l'Alphacom 32, e siamo certi che non ci maledirete più per la scarsa leggibilità. Viene mantenuto il formato di stampa in 32 colonne, e anche il metodo è più vicino a quello della ZX-Printer che di una tradizionale.

Sebbene, infatti, a prima vista faccia pensare a una matrice a punti, non si vede nessuna testina, e la scrittura è ottenuta con "punti caldi" che si muovono a zig-zag; invece di avere due soli aghi che attraversano tutto il foglio, i punti sono 16 (una hard-copy di schermo nero mostra 16 colonne di riempimento), su una barra mossa trasversalmente da



una camma. Poichè ogni punto deve fare poca strada, si può avere una velocità nettamente superiore a quella della ZX-Printer.

Tutto OK, allora? Stampa nitida, prezzo ragionevole anche per lo ZX81, buona velocità, silenziosità... volendo cercare, un difettino si trova, non grave: l'hard-copy, ottimo come resa grafica, non è sincronizzato bene e la carta avanza troppo velocemente rispetto alla stampa: così una funzione CIRCLE dello Spectrum viene resa con un grazioso ovale. Ma pazienza. Promossa ugualmente con 8 e 1/2.

Seikosha GP-100A + Interfaccia

Della Seikosha in sé non abbiamo niente di nuovo da dire: è una delle stampanti economiche più vendute, con un ottimo rap-
vate sono: la MEMOPAK I/F (Centronics Type Parallel Printer Interface) della Memotech, per lo ZX81, e due schede di produzione artigianale per lo Spectrum, sostanzialmente uguali nello schema, ambedue accompagnate da una cassetta di software.

La Seikosha è troppo nota agli utenti perchè si renda necessaria una presentazione



La confezione viene fornita con l'alimentatore e un rotolo di carta speciale.

La Memotech, inglese di Oxford, è ben nota nell'ambiente Sinclair, avendo in catalogo vari accessori (RAM; grafica Hi-Res, porto prezzo/prestazioni, ed è stata recensita da tutta la stampa specializzata e non. Il prezzo di listino attuale è circa 550.000 lire (+ IVA), il che non ne giustifica l'acquisto, forse, per lo ZX81, ma la rende interessante per lo Spectrum, soprattutto se lo si usa molto per output scritti.

Le schede di interfaccia protastiere...), disegnati apposta per accompagnarsi alla scatola dello ZX81 (tutti rigorosamente neri).

Il manuale d'uso è molto preciso, diremmo che è il classico "a prova d'imbecille" (meglio così, ovviamente, seppure di un imbecille che deve sapere l'inglese), strutturato com'è a domande e risposte. La scheda non prevede software speciale: si collega e si usa. Qui abbiamo un appunto da fare: tanta raffinatezza di confezione, ma il connettore d'uscita non ha la slitta-guida per la spina del cavo tipo Centronics. In fondo, ci sono sole due possibilità, una giusta e una sbagliata.

Funziona bene per il LLIST e per LPRINT; una serie di controlli ordinari della stampante, come il *line-feed* o la stampa a doppia larghezza, si ottengono via software con LPRINT CHR\$ seguito da codici appropriati (indicati nel manuale).

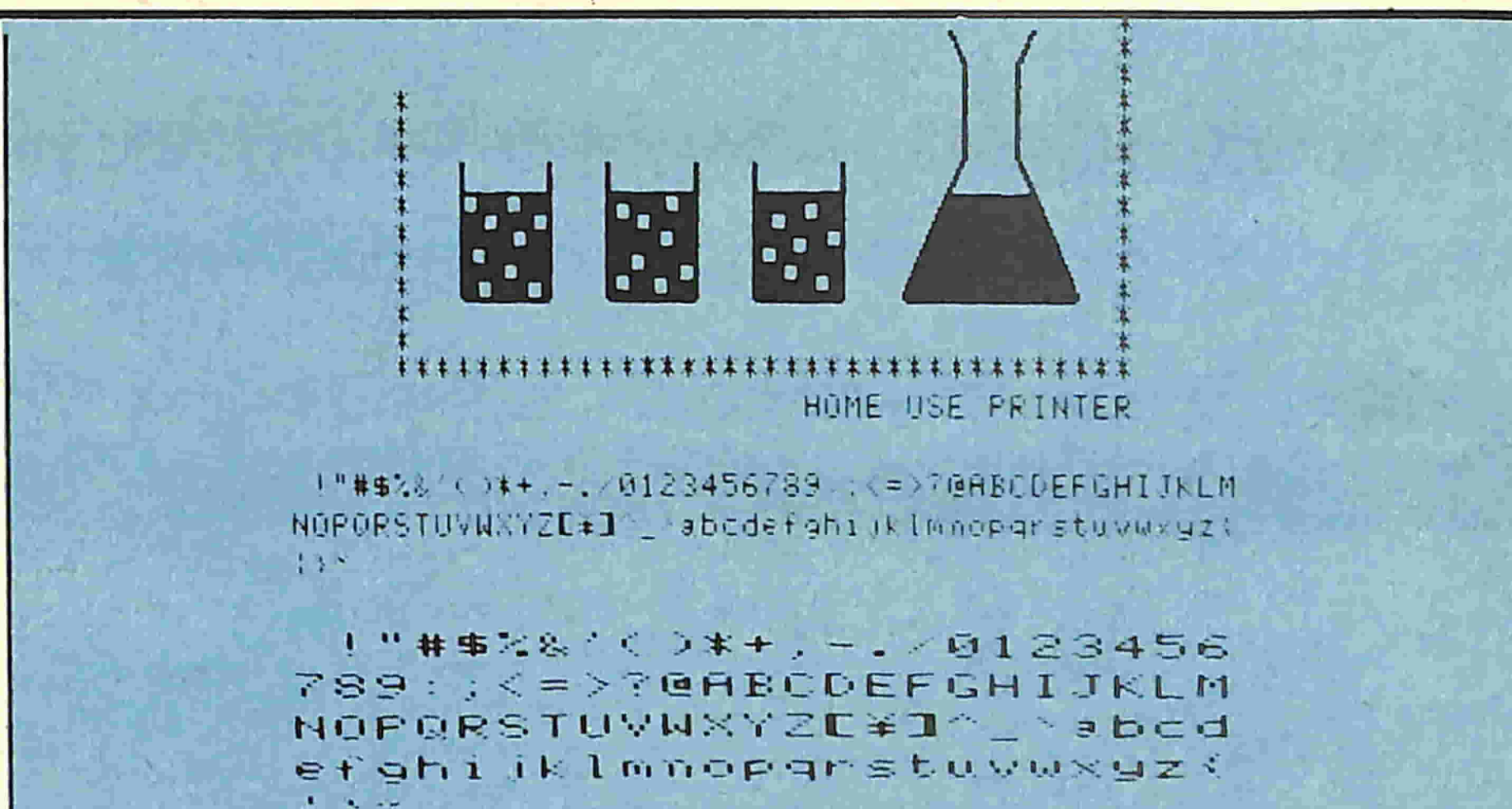
Se si vogliono scrivere testi, si possono fare cose interessanti: la conversione ASCII riconosce i caratteri in campo inverso come minuscole, e come tali li stampa. Caratteri speciali si possono otte-

nere variando le combinazioni dei primi due microswitch sul retro della Seikosha, che selezionano i diversi set alfabetici nazionali. Le istruzioni dell'interfaccia non fanno cenno agli switch, e il funzionamento sembra uguale in qualsiasi combinazione.

Dove la Memotech imbrogia un po' le carte è sulla funzione COPY, lasciando intendere che può essere usata normalmente: essendo la GP-100A una graphic printer, si penserebbe di poter fare l'hard copy dello schermo, mentre ciò è vero solo per caratteri alfanumerici. Su caratteri grafici o pixel plottati il COPY risponde con errore C (abbiamo provato, per scrupolo, con tutte le posizioni di switch). Se occorre inviare prima qualche istruzione, non viene scritto.

Comunque, 8 e 1/2 alla Seikosha e 7 alla Memotech. Le interfacce provate per lo Spectrum sono sostanzialmente uguali: schema simile, stessi IC. Una, siglata "B&V Interface", è vestita con un contenitore plastico bianco che sembra tagliato con strumenti da boscaiolo. L'altra, della "Nuova NEWEL", addirittura senza protezione, rivela però più cura nella costruzione: piste ordinate, integrati "zoccolati" e non saldati come nella prima.

In pratica, a parte le considerazioni estetiche, le due schede si sono rivelate uguali anche nel software richiesto e nei risultati. Ora non veniteci a chiedere chi dei due ha copiato, forse l'han fatto tutti e due da un terzo.



Esempio di stampa con la Seikosha GP 100A.

La cassetta contiene un breve programma in L/M, che va caricato in posti diversi della RAM, a seconda che abbiate il 16K o il 48K. Le poche righe di Basic all'inizio fungono appunto da loader, oltre a contenere la pubblicità. La dislocazione del L/M è prevista il più in alto possibile (col 48K parte a 64515), per poterla utilizzare con qualsiasi programma, anche molto lungo. Problemi possono sorgere con giochi che hanno anch'essi routine L/M piazzate là in cima.

L'uso della Seikosha con lo Spectrum è immediato e dà ottimi risultati. Premesso che bisogna porre gli switch 1 e 3 su ON (2 e 4 su OFF), cosa che non trovate scritta da nessuna parte neanche questa volta, si carica il programmino di gestione, si cancella il Basic con un NEW, e si può caricare il software che ci interessa. Dopo ogni NEW, RUN e CLEAR occorre una reinizializzazione (inseribile anche nel programma) con RANDOMIZE USR 64973. A questo punto LPRINT

e LLIST sono normalmente operative (ad esclusione di LPRINT AT), e stampano su tutte le 80 colonne. Per avere un COPY del video, invece, si chiama RAND USR 65044, che riproduce qualsiasi simbolo grafico o carattere, sulla larghezza di 32 colonne, ben proporzionato (il quadrato resta quadrato). Con RAND USR 65047 si ottiene un bellissimo COPY a tutta pagina; anzi, un po' di più, perchè riproduce le prime 30 colonne, perdendo le ultime due a destra.

Se per errore cercate di fare un COPY da tastiera, si perde il controllo del sistema: di solito è riacquistabile spegnendo la stampante, senza dover ricaricare.

Questa possibilità dell'Hard-Copy grafico ad alta risoluzione è una delle caratteristiche che possono rendere particolarmente felice il matrimonio Spectrum/-Seikosha, considerato anche il costo di questa stampante. Diamo ancora 8 e 1/2 alla Seikosha e 7 e 1/2 all'interfaccia, per le carenze nelle istruzioni.

Viaggio nello Spectrum

Seconda parte



LO Z80A è il microprocessore dello Spectrum. Ha 40 pin ed è siglato IC2 nello schema generale del computer. La CPU svolge molte funzioni fondamentali:

- legge dati dalla memoria;
- scrive dati nella memoria;
- legge dati da un dispositivo di I/O;
- scrive dati in un dispositivo di I/O;
- svolge operazioni aritmetiche e logiche sui dati.

Quello che il microprocessore fa in un dato istante dipende dalle istruzioni che riceve dalla memoria. Uno Z80A riconosce 158 diversi tipi di istruzione, e la sua programmazione in codice macchina è una materia a sè. Forse avete già sentito parlare dello Z80 e di computer che lavorano con questo microprocessore. E' identico allo Z80A (ne è, in un certo modo, il padre): la sola differenza è la velocità di lavoro, passata da 2.5 MHz a 4 MHz. L'ultimo nato della famiglia è lo Z80B, che lavora a 6 MHz.

Come abbiamo già detto, lo Z80A è un microprocessore a 8 bit: effettua i trasferimenti di dati a 8 bit per volta. I primi microprocessori erano dispositivi a 4 bit, oggi se ne costruiscono anche a 16 e a 32 bit. Questi sono molto più veloci di quelli a 8 bit, potendo trasferire il doppio o il quadruplo di informazioni nello stesso tempo. Tuttavia, essendo molto più costosi, non vengono usati, per ora, negli home-computers, dove domina incontrastato l'8 bit.

Nello Spectrum, il segnale di clock dello Z80A oscilla a 3.5 MHz (eccettuato quando è in collegamento con la memoria video).

Tutte le operazioni della CPU sono in sincronia con questo segnale di clock; le transizioni 0-1 e 1-0 devono avvenire molto velocemente per essere sicuri che l'inizio di ogni ciclo in tutte le operazioni avvenga esattamente nello stesso istante. Per questo è necessaria la presenza di TR3, con le resistenze e i diodi che lo circondano. Il segnale di clock a 3.5 MHz dalla ULA non esegue con sufficiente velocità le transazioni tra gli stati logici 0 e 1.

Tutti i circuiti interni alla CPU possono iniziare

a lavorare a tensioni leggermente diverse nell'intervallo +0.8V / +2V. Se il tempo per il segnale di clock per andare da +0.8V a +2V è significativo (in nanosecondi), parti diverse delle circuitazioni interne possono attivarsi in istanti diversi. E se si verificano differenze negli istanti di attivazione, lo Z80A, può avere malfunzionamenti. Il TR3 assicura che la transazione al pin di clock della CPU avvenga con sufficiente rapidità.

Descrizione dei pin dello Z80A

(Nota: le sigle sopralineate, come $\overline{RD}$, stanno a significare attività a segnale basso, cioè l'operazione avviene con $\overline{RD}=0$; normalmente, 1=vero e 0=falso, ma per i segnali di controllo con sigla sopralineata 1=falso e 0=vero).

(Da 30 a 40 e da 01 a 05) A0/A15: bus indirizzi. Output a tre stati (cioè 0, 1 e floating). Quando sta cambiando, un altro dispositivo provvede al controllo del bus indirizzi al posto della CPU. Il bus dispone di $2^{16} = 65536$ diversi indirizzi per gli scambi di dati in memoria, oppure di $2^8 = 256$ indirizzi per i dispositivi I/O (durante le operazioni di I/O si usano le otto linee di indirizzo più basse). La CPU può anche rinfrescare la memoria (vedi il pin di refresh) fornendo indirizzi validi di refresh sui 7 bits più bassi del bus indirizzi (06): clock.

(07/10 e 12/15) D0/D7: bus dati. Queste otto linee bidirezionali a tre stati sono usate per i trasferimenti di dati in input e in output. I trasferimenti avvengono tra la memoria o dispositivi periferici e lo Z80A.

(11): +5 Volt.

(16) $\overline{INT}$: ingresso richiesta di interrupt, attivo basso. Questo segnale può essere generato da dispositivi esterni, per far sì che la CPU esegua un particolare programma in codice macchina, che si trova da qualche parte nella memoria. Se il flip-flop interno, abilitatore dell'interrupt controllato via software, è attivato, la CPU accetta e riconosce l'interrupt. Un flip-flop è equivalente a un bit di memo-

ria. La CPU usa alcuni bit della sua memoria interna per ricordare se deve o meno accettare gli interrupt provenienti da altri dispositivi. Per ulteriori dettagli sull'uso di questi interrupt, consultate un manuale dedicato allo Z80.

(17) $\overline{\text{NMI}}$: input di interrupt non mascherabile. Opera sul connettore (going-edge) negativo del segnale interrompente. $\overline{\text{NMI}}$ è sempre accettato dalla CPU alla fine dell'istruzione corrente. Forza la CPU ad eseguire il programma in memoria, iniziando al-

l'indirizzo 102 decimale. (66 hex).

(18) $\overline{\text{HALT}}$: output, attivo basso. Indica che la CPU ha eseguito un'istruzione di $\overline{\text{HALT}}$ nel software. Attende un interrupt da un altro dispositivo prima di riprendere le operazioni.

(19) $\overline{\text{MREQ}}$: output a tre stati di interrogazione della memoria, attivo basso. Questo segnale indica alla memoria che il bus indirizzi ora contiene un indirizzo valido per un'operazione di lettura o scrittura. Viene richiesto per distinguere tra un'opera-

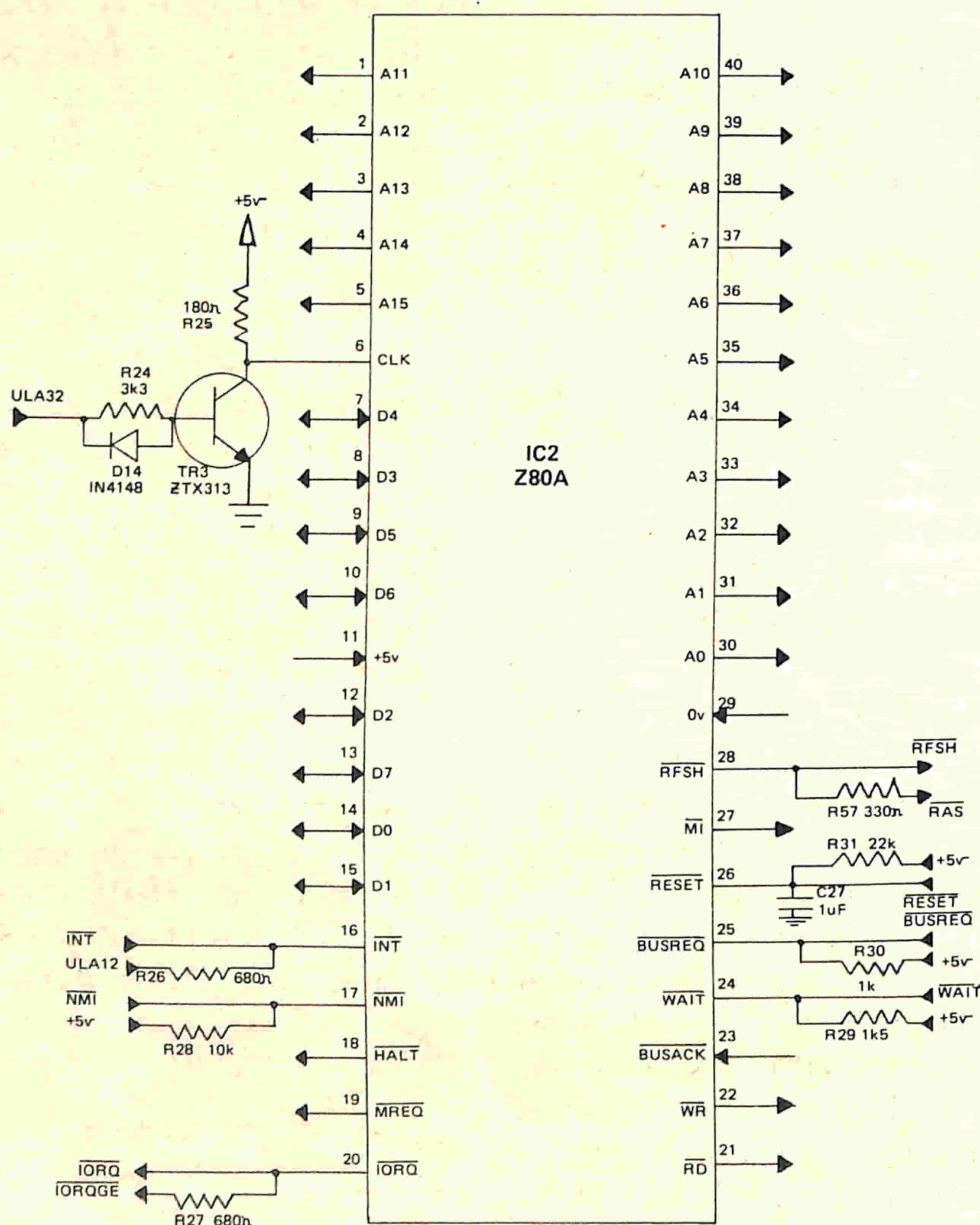


Fig. 1. Lo Z80A.

zione di memoria e una di I/O. Per esempio. MREQ, RD e ROMCS devono essere tutti attivi prima di poter effettuare una lettura dalla ROM del Basic.

(20) $\overline{\text{IORQ}}$: interrogazione di I/O; output a tre stati, attivo basso. Indica che la metà inferiore del bus indirizzi contiene un indirizzo valido di I/O per un'operazione di lettura o scrittura. Al segnale viene richiesto di distinguere tra un'operazione di memoria e una di input/output. Da notare che $\overline{\text{IORQ}}$ e $\overline{\text{MREQ}}$ non possono mai essere attivi contemporaneamente.

(21) $\overline{\text{RD}}$: lettura; output a tre stati, attivo basso. Indica che la CPU vuole leggere dalla memoria o da un dispositivo di I/O. La periferica indirizzata userà il segnale per inviare i dati interessati sul bus dati della CPU.

(22) $\overline{\text{WR}}$: scrittura; output a tre stati, attivo basso. Indica che il bus dati della CPU contiene dati da immagazzinare nelle locazioni indirizzate, oppure output destinati alla porta I/O selezionata. La memoria userà questo segnale per immagazzinare i dati provenienti dal bus dati.

(23) $\overline{\text{BUSACK}}$: output di riconoscimento del bus, attivo basso. Indica alla periferica coinvolta che ora può prendere il pieno controllo del bus della CPU.

(24) $\overline{\text{WAIT}}$: input, attivo basso. Usato dalla memoria lenta o dai dispositivi di I/O per dire alla CPU che non sono ancora pronti per il trasferimento dei dati. La CPU resta senza far niente finché la periferica lenta non indica che è pronta.

(25) $\overline{\text{BUSREQ}}$: input di interrogazione del bus, attivo basso. Usato da un'unità periferica per richiedere l'accesso al bus indirizzi della CPU, al bus dati e ai segnali di controllo dell'output a tre stati per i suoi usi propri. La CPU riprende il controllo di tutti i suoi bus non appena il ciclo corrente è terminato; per indicare all'unità periferica che l'ha richiesto che il controllo può essere preso, la CPU setta "basso" il pin $\overline{\text{BUSACK}}$ (v.).

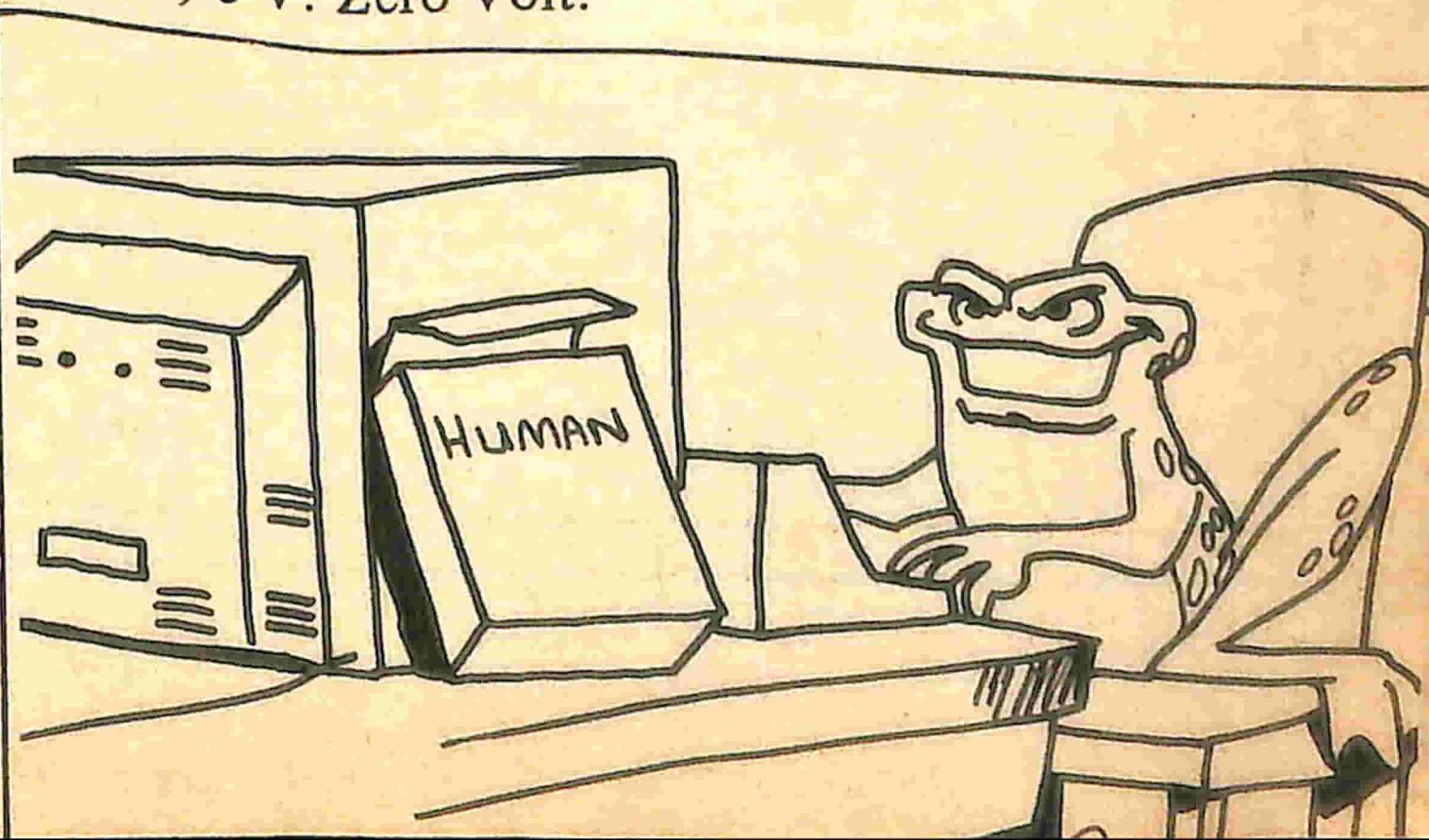
(26) $\overline{\text{RESET}}$: input, attivo basso. Questo segnale forza la CPU per portarla in stato di reset. Nello

Spectrum, quando si inserisce l'alimentazione, il pin di reset viene mantenuto basso da C27, finché questo non viene caricato attraverso R31. Ciò per consentire il resto del computer di raggiungere lo stato operativo, prima che la CPU inizi il lancio del programma in memoria ROM dall'indirizzo 0000. In seguito vedremo come applicare allo Spectrum un pulsante di reset.

(27) $\overline{\text{M1}}$: ciclo di macchina uno; output, attivo basso. Il segnale significa che la CPU sta in quell'istante eseguendo il codice operativo (Op-code) per la prossima istruzione presente in memoria. M1 inoltre interviene con $\overline{\text{IORQ}}$ per indicare un ciclo di riconoscimento di interrupt.

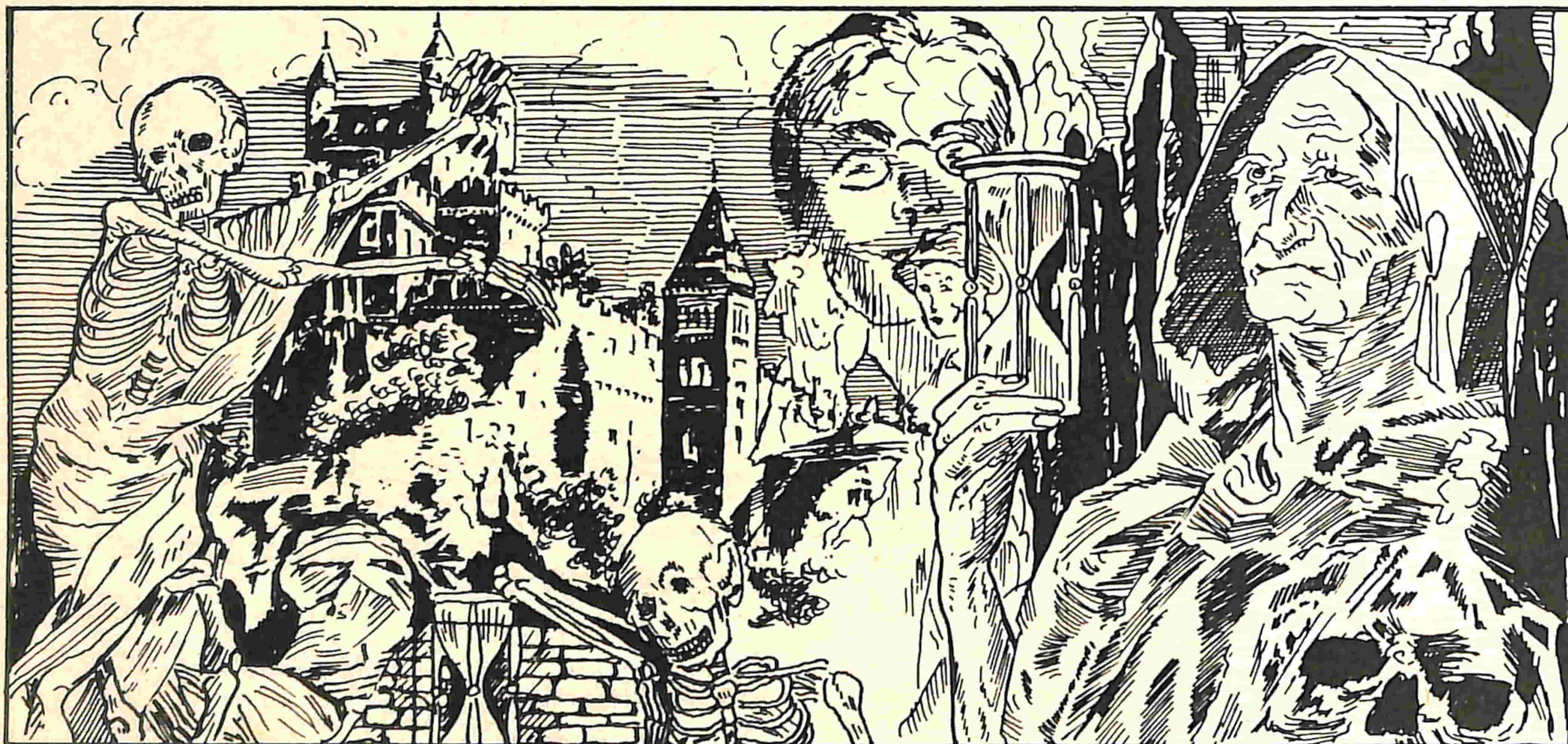
(28) $\overline{\text{RFSH}}$: refresh, output di rinfresco, attivo basso. Indica che i 7 bit più bassi del bus indirizzi contengono un indirizzo di refresh per le memorie dinamiche. L'operazione è necessaria ai chip di memoria dinamica perchè non "dimentichino". Se il refresh non viene eseguito almeno una volta ogni due millisecondi, c'è il pericolo di perdere la memoria. Quando il segnale è presente, le otto linee più alte del bus indirizzi riportano i contenuti del registro I della CPU. Ciò conduce ad uno stato di fault nell'hardware dello Spectrum. Settando il registro I a un qualsiasi valore tra 64 e 127, avvengono cose interessanti nella RAM video. A15 e A14 dalla CPU stanno selezionando i 16k di RAM su cui normalmente la ULA ha la priorità. In questo caso, sebbene l'indirizzo relativo si raggiunga con MREQ attivo, non si invia nessun segnale RD nè WR, poiché si tratta di un refresh. Questa combinazione di segnali sembra confondere la ULA, cosicché essa non ferma correttamente il clock della CPU.

(29) 0 V: Zero Volt.



“LA STRADA DELLA DANNAZIONE”

tra zombiës, muri vaganti e guardiani corrotti, una spiritosa variante sul labirinto a punteggio



```

1 REM : : : : : : : : : : : : : : : : : : : :
2 REM : : : : LA STRADA DELLA : : : :
3 REM : : : : DANNAZIONE : : : :
4 REM : (WHICH MICRO?/10.83) : : : :
5 REM : : : : VERS. IT. BY MAUBAT/83 : : : :
6 GOSUB 6600
10 CLS
15 LET A=730
20 LET S=0
25 LET DFS=PEEK 16396+256*PEEK
16397
30 LET B=0
35 LET KPS=DFS+576
40 LET OPM=DFS+50
50 LET NPM=OPM
60 LET TIM=-1
70 LET T1=8
75 LET K1=20217
80 LET KM=0
200 GOSUB 6000

```

```

290 IF T1=0 THEN LET T1=8
291 LET T1=T1-1
292 IF T1=0 THEN GOTO 4000
295 IF TIM=0 THEN GOTO 4400
297 LET TIM=TIM-1
298 IF INT (RND*16)=10 AND KM=1
THEN GOTO 2000
300 GOTO 290+710*(INKEY$="7")+7
30*(INKEY$="6")+750*(INKEY$="5")
+770*(INKEY$="8")+790*(INKEY$="0")
1000 LET NPM=OPM-33
1010 GOTO 1200
1020 LET NPM=OPM+33
1030 GOTO 1200
1040 LET NPM=OPM-1
1050 GOTO 1200
1060 LET NPM=OPM+1
1070 GOTO 1200
1080 GOTO 8000
1200 IF PEEK NPM=126 THEN GOTO 2

```



```

90
1210 IF PEEK NPM=0 THEN GOTO 140
0
1220 IF PEEK NPM=27 THEN GOTO 13
50
1230 IF PEEK NPM=8 THEN GOTO 160
0
1240 IF PEEK NPM=28 THEN GOTO 17
00
1250 IF PEEK NPM=134 THEN GOTO 1
850
1260 IF PEEK NPM=6 THEN GOTO 190
0
1350 LET S=S+5
1400 POKE OPM,0
1410 POKE NPM,23
1420 LET OPM=NPM
1430 GOTO 290
1500 POKE OPM,8
1510 POKE NPM,23
1520 LET OPM=NPM
1530 GOTO 290
1700 LET B=B+100
1710 POKE OPM,6
1715 POKE NPM,23
1720 LET OPM=NPM
1721 LET NPM=NPM+33
1722 POKE OPM,128
1723 POKE NPM,23
1724 LET OPM=NPM
1725 GOTO 290
1730 REM : : : AREA DELLA MORTE
1810 GOTO 290
1850 POKE OPM,0
1855 POKE NPM,23
1860 LET OPM=NPM
1865 LET NPM=OPM-33
1870 POKE OPM,6
1880 POKE NPM,23
1885 LET OPM=NPM
1888 LET KM=1
1890 GOTO 290
1900 POKE OPM,6
1910 POKE NPM,23
1920 LET OPM=NPM
1930 LET NPM=OPM+33
1940 POKE OPM,134
1950 POKE NPM,23
1955 LET OPM=NPM
1958 LET KM=0
1960 GOTO 290
2000 LET RUN=INT (RND*14)
2010 FOR X=1 TO RUN
2015 LET K1=KPS-(X*33)
2020 IF PEEK K1=23 THEN GOTO 210
0
2030 POKE K1+33,6
2040 POKE K1,21
2050 NEXT X
2055 POKE K1,8
2060 GOTO 290
2100 FOR X=128 TO 138
2110 POKE NPM,X
2120 NEXT X
2130 GOTO 3000
2480 REM : : : AREA DELLA MORTE
2540 GOTO 290

```

```

3000 CLS
3015 PRINT " SEI STATO ASSASSIN
ATO DA UNO ZOMBIE INFURIATO."
3017 PRINT " PERD" SEI MORTO
CON "S+B;" CREDITI."
3020 PRINT " NON CHE TI SERVA
NO A MOLTO, ORA. NON SEI NEA
NCHE NELLA SI- TUAZIONE MIGLIOR
E PER FARE TESTAMENTO."
3030 GOTO 8150
4000 IF INT (RND*15)=5 THEN GOTO
4500
4010 LET BPS=DFS+(INT (RND*22)+1
)*32+INT (RND*33)
4015 IF PEEK BPS=118 THEN GOTO 2
90
4016 IF PEEK BPS=8 THEN GOTO 290
4017 IF PEEK BPS=23 THEN GOTO 29
0
4018 IF PEEK BPS=128 THEN GOTO 2
90
4019 IF PEEK BPS=134 THEN GOTO 2
90
4020 IF PEEK BPS=6 THEN GOTO 290
4030 POKE BPS,128
4050 GOTO 290
4400 LET TIM=30
4405 POKE DFS+118,128
4410 GOTO 290
4500 LET TIM=30
4510 POKE DFS+118,28
4515 IF S+B>1300 THEN GOTO 8500
4520 GOTO 290
6000 PRINT AT 0,0; "
6010 PRINT "
6020 PRINT "
6030 PRINT "
6040 PRINT "
6050 PRINT "
6060 PRINT "
6070 PRINT "
6080 PRINT "
6090 PRINT "
6100 PRINT "
6110 PRINT "
6120 PRINT "
6130 PRINT "
6140 PRINT "
6150 PRINT "
6160 PRINT "

```

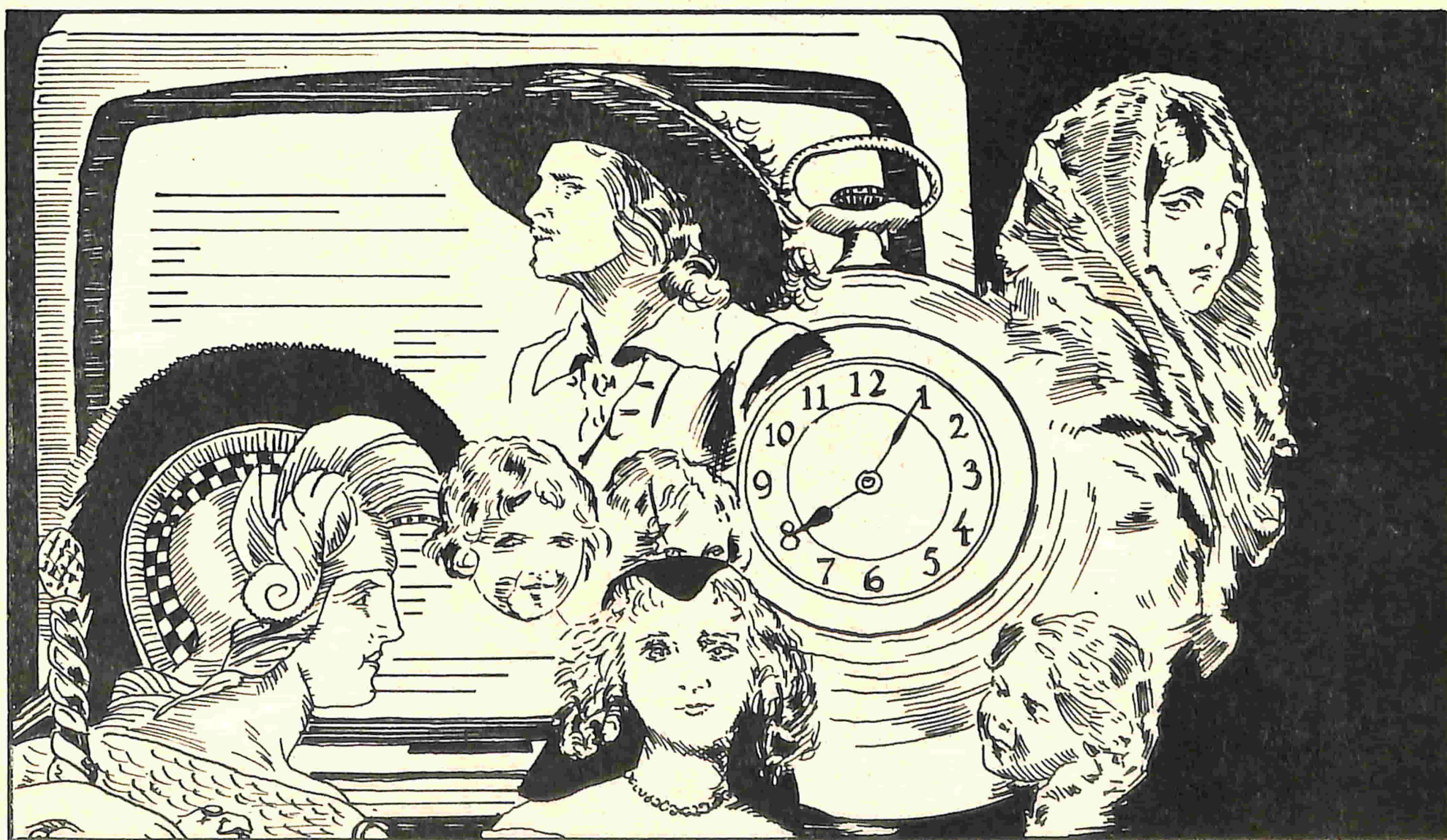

“ALMANACCHI, ALMANACCHI NUOVI...”

il nostro ZX81, facendo sfoggio di erudizione leopardiana, s'è messo a produrre almanacchi per l'anno nuovo e anche per gli anni vecchi

CON questo “GREGKAL”, cioè software per calendario gregoriano, potete stampare il taccuino di un qualsiasi anno posteriore al 1583. Questa data, per chi non lo sapesse, è quella di introduzione

del calendario che tuttora usiamo, e che prese il nome del papa che lo promulgò. Per avere anche anni precedenti, occorrerebbe qualche complicazione in più, perchè nell'ottobre 1583 dovet-

tero saltare dieci giorni, passando dal 04 al 15, a causa di un errore nel calendario fino allora in uso; errore protrattosi per circa 1600 anni, visto che il metodo risaliva agli astronomi di Giulio Cesare.



Potete iniziare a stampare da un mese qualsiasi, e per il numero di mesi che volete - entro l'anno scelto. Il programma non è molto veloce, non per i calcoli, che durano pochi secondi, ma a causa della routine di ingrandimento dei caratteri, decisamente di scarsa efficienza.

Note tecniche

Il Julian Day, o giorno giuliano, non ha niente a che vedere con il calendario giuliano: è un metodo di conteggio "assoluto" del tempo, che numera i giorni

astronomici a partire da circa 2.500.000 giorni fa: è molto usato, per la sua praticità, nei calcoli astronomici. Lo ritroverete nel programma per lo Spectrum per trovare l'Ascendente, in'altra parte di questo stesso numero della rivista.

La routine che cerca gli anni bisestili abbisogna forse di una spiegazione: anzitutto notate come con una sola istruzione, con l'uso degli operatori logici, sia possibile scegliere tra ben quattro situazioni diverse. Nell'attuale calendario sono bisestili, come sapete, gli anni divisibili per 4, pe-

rò: gli anni secolari (che terminano con due zeri: 1700, 1800, etc.), sono bisestili solo se divisibili per 400. Quindi hanno febbraio di 29 giorni solo il 1600, il 2000, il 2400...

Per esempio...

... per incollare sull'agenda qualcosa di diverso, per scoprire in che giorno siete nati, per trovare che la presa della Bastiglia avvenne di martedì, per fare agli amici un regalo ottimista: un chilometrico calendario da qui al 2023.

```

10 REM
12 REM
13 REM   CALENDARIO
14 REM   GREGORIANO
15 REM
16 REM
17 REM   BY MAUBERT/83 - - - -
18 REM
19 REM ---INIZIALIZZ. E INPUT-
20 GOSUB 5000
30 SLOW
40 PRINT " * * * * STAMPA CA
LENDARIO * * *"
50 PRINT
60 PRINT "DA DOVE COMINCIAMO ?
70 PRINT
80 PRINT " ** ANNO (>1583): ";TA
B 25;
90 INPUT A
100 PRINT A
110 PRINT
120 PRINT " ** MESE ";TAB 25;
130 INPUT M
140 PRINT M
150 PRINT
160 PRINT " ** QUANTI MESI ? ";T
AB 25;
170 INPUT N
180 PRINT N
190 PAUSE 50
200 POKE 16437,255
203 REM ---CALCOLO JULIAN DAY--
204 REM ---E GIORNO D.SETTIMANA
210 FAST
230 LET A1=A-(1 AND M<=2)
240 LET M1=M+(12 AND M<=2)
250 LET B=INT (A1/100)

```

```

260 LET B=2-B+INT (B/4)
270 LET JD=INT (365.25*A1)+INT
(30.6001*(M1+1))+1720997+B
280 LET UD=JD-INT (JD/7)*7
290 LET UD=UD+(7 AND UD=0)
295 REM ---STAMPA INGRAND:ANNO-
360 LET A$=STR$ A
370 LET E$(13)="2300"+STR$ CODE
A$(1)+STR$ CODE A$(2)+STR$ CODE
A$(3)+STR$ CODE A$(4)+"002323"
380 LET P=13
390 LET Q=43
400 CLS
410 GOSUB 5500
415 REM ---STAMPA INGRAND.MESE-
420 LET Q=28
430 FOR P=M TO M+N-1
440 GOSUB 5500
443 REM ---ST.GIORNI D.SETTIM.-
445 FOR I=1 TO 7
450 PRINT AT 14+I,0;F$(I*9-8 TO
I*9)
455 NEXT I
457 REM ---RIC.ANNI BISESTILI--
460 GOSUB 5000
463 REM ---NUM.GIORNI D.MESE---
465 LET L=31
470 IF P=4 OR P=6 OR P=9 OR P=1
1 THEN LET L=30
480 IF P=2 THEN LET L=28+L1
483 REM ---STAMPA DATE-----
485 LET J=0
490 FOR I=1 TO L
500 PRINT AT 14+UD,11+J;" " AND
I<10;I
510 LET UD=UD+(1 AND UD<7)-(6 A
ND UD=7)

```



```

520 LET J=J+(3 AND WD=1)
530 NEXT I
540 REM ---INVERSE DOMENICHE---
550 LET K=255*PEEK 16397+PEEK 1
6396+703
560 FOR I=K TO K+22
570 POKE I,PEEK I+128
580 NEXT I
583 REM ---HARD COPY-----
585 COPY
587 CLS
590 NEXT P
600 GOTO 30
4997 REM ---CALC. ANNI BISESTILI
5000 LET L1=1*((A-INT (A/100))*10
0=0 AND A-INT (A/400)*400=0) OR
(A-INT (A/4)*4=0 AND A-INT (A/10
0)*100<>0))
5020 RETURN
6497 REM ---INGRANDIM. CARATTERI-
6500 FOR H=1 TO 9
6510 LET Z=VAL E$(P,H*2-1 TO H*2
)
7000 FOR I=0 TO 7
7010 LET T=PEEK (7680+8*Z+I)
7020 FOR J=7 TO 0 STEP -1
7030 LET U=T-INT (T/2)*2
7040 IF U<>0 THEN PLOT J+7*(H-1)
,0-I
7050 LET T=INT (T/2)
7060 NEXT J
7070 NEXT I
7080 NEXT H
7100 RETURN
7997 REM ---DATI-----
8000 CLEAR
8010 DIM E$(13,18)
8020 LET C$="44425151384652000004
34239395538465200503855635200000
000385355464942000000050384444465

```

```

2000000044465844515200000004958444
946520000000384452565752000000564
25757425039554252575752395542000
05152594250395542004146404250395
54200"
8030 FOR F=1 TO 12
8040 LET E$(F)=C$(F*18-17 TO F*1
8)
8050 NEXT F
8070 LET F$="LUNEDI MARTEDI M
ERCOLEDIGIOVEDI VENERDI SABATO
DOMENICA"
8110 RETURN
9000 CLEAR
9020 SAVE "GREGKA"
9030 RUN

```

* 1884 **

GENNAIO

LUNEDI		7	14	21	28
MARTEDI	1	8	15	22	29
MERCOLEDI	2	9	16	23	30
GIOVEDI	3	10	17	24	31
VENERDI	4	11	18	25	
SABATO	5	12	19	26	
DOMENICA	6	13	20	27	

* 1789 ** * 1984 **

LUGLIO

LUNEDI		6	13	20	27
MARTEDI		7	14	21	28
MERCOLEDI	1	8	15	22	29
GIOVEDI	2	9	16	23	30
VENERDI	3	10	17	24	31
SABATO	4	11	18	25	
DOMENICA	5	12	19	26	

FEBBRAIO

LUNEDI		6	13	20	27
MARTEDI		7	14	21	28
MERCOLEDI	1	8	15	22	29
GIOVEDI	2	9	16	23	30
VENERDI	3	10	17	24	
SABATO	4	11	18	25	
DOMENICA	5	12	19	26	

“SLALOM”

a volte, le cose semplici sono quelle che funzionano meglio

UN gioco di abilità su una pista tortuosa quanto basta; voi siete il piccolo asterisco in basso sullo schermo, e potete muovervi a destra e a sinistra, usando rispettivamente i tasti M e N.

La pista vi viene incontro, dall'alto in basso: dovete “solo” imboccarla e seguire le curve.

Quando andate a sbattere, e prima o poi capita a tutti, vi viene comunicato il tempo, cioè quanto siete riusciti a sopravvivere.

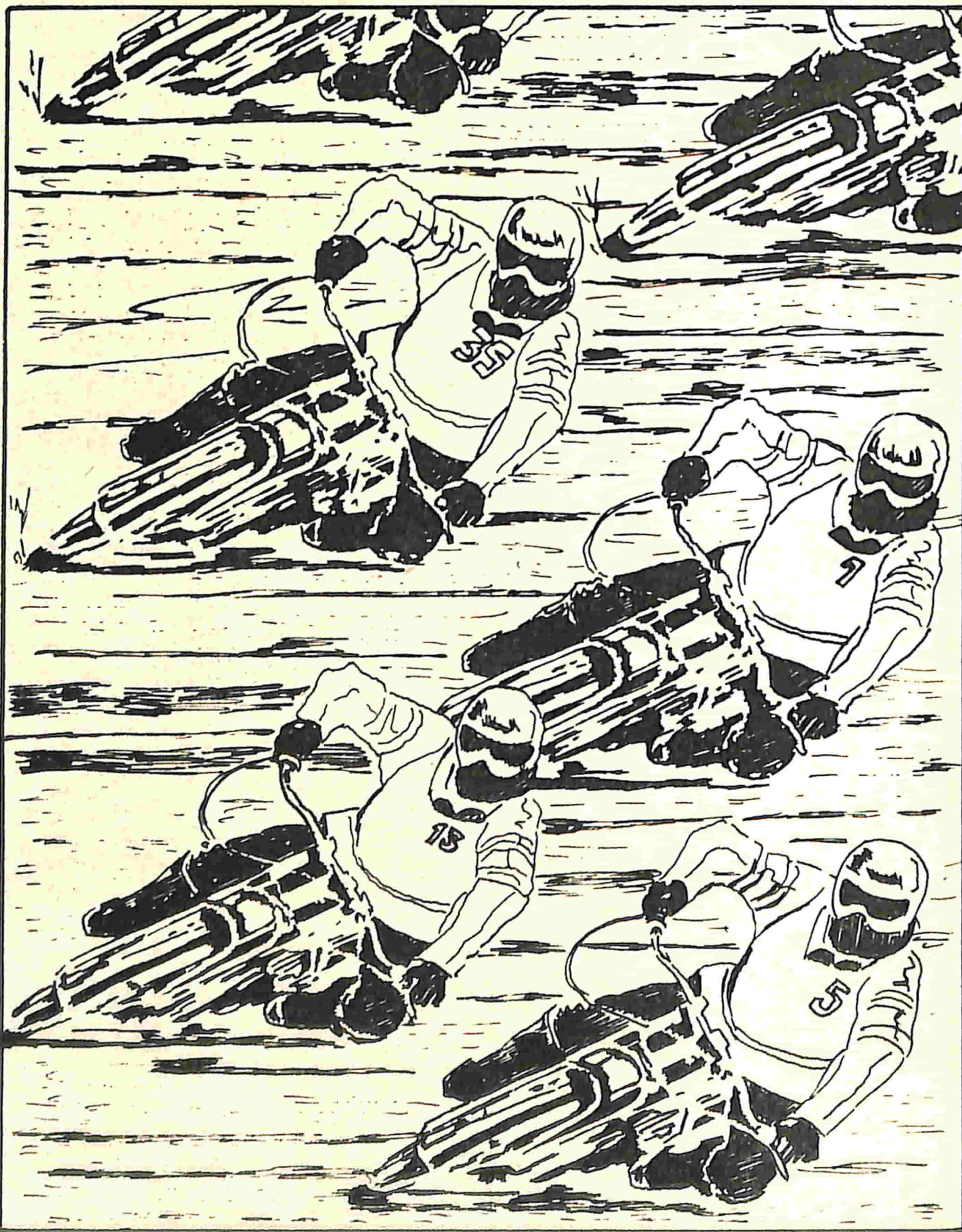
Il gioco, girando sullo ZX81, non è molto vestito, e la grafica è meno che essenziale, ma - facciamo una considerazione generale - programmi brevi e “minimali” come questo li scegliamo in genere perchè contengono routines interessanti: se il software pubblicato sui periodici non fosse anche riciclabile, servirebbe a poco, no?

In questo caso trovate la routine in L/M che simula il movimento, basilare in molte situazioni di gioco. Il listato inizia con una REM di 73 caratteri, in cui caricare il L/M, che è riprodotto insieme alle istruzioni disassemblate. Se volete collocare la routine in altre aree della memoria,

dovete fare attenzione agli indirizzi assoluti e ai salti.

Il programma si lancia con

RUN 270; dopo un incidente, per ripartire premere un tasto diverso da quelli di guida. ■




```

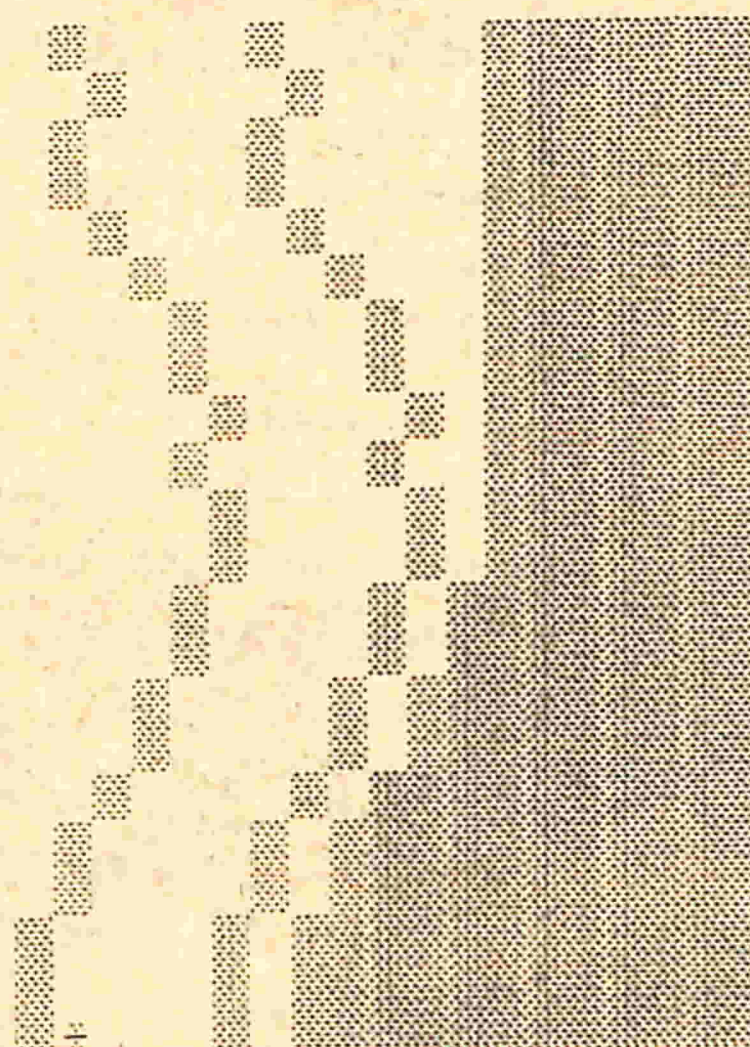
1 REM SGN ?USRND RETURN 4-UR
RND RETURN LOAD 4-Y COPY / RETU
RN RUN 4-Y=5RND=?EERND=5RND
5 ??E=LRND=?E= GOSUB 55SGN ? :
0* : TAN .....0
3 REM
5 REM SLALOM
7 REM ZX-USER CLUB/D
9 REM
10 LET B=A+INT (RND*3)-1
20 IF B>1 AND B<21 THEN LET A=
B
30 IF USR 16516<>0 THEN GOTO 2
00
40 PRINT AT 0,A;" "
50 GOTO 0
100 LET A=9
105 POKE 16574,A
110 POKE 16436,255
115 POKE 16437,255
120 LET B=694+PEEK 16396+255*PE
EK 16397

```

```

125 POKE 16571,B-255*INT (B/255
)
130 POKE 16572,INT (B/255)
135 GOTO 0
200 LET A=INT ((65535-PEEK 1643
6-255*PEEK 16437)/50)
210 PRINT AT 0,0;"TEMPO : ";IN
T (A/60);" MIN ";A-INT (A/60)*60
;" SEC"
220 FOR A=0 TO 70
230 PRINT AT 21,PEEK 16574;" "
240 PRINT AT 21,PEEK 16574;"*"
250 NEXT A
260 IF INKEY$="" THEN GOTO 260
270 CLS
280 PRINT " ";
" ";AT 0,USR 16516;" "
290 RUN 100
9000 CLEAR
9010 SAVE "SLALO"
9020 STOP

```



```

4082 00 NOP
4083 00 NOP
4084 0A 25 40 LD A,(4025)
4087 7F 7F CP 7F
4089 20 16 JR NZ,40A1
408B 0A 26 40 LD A,(4026)
408E 7F 7F CP 7F
4090 20 04 JR NZ,4096
4093 0E FF LD A,FF
4094 13 06 JR 409C
4096 7F F7 CP F7
4098 20 07 JR NZ,40A1
409A 3E 01 LD A,01
409C 21 BE 40 LD HL,40BE
409F 86 ADD A,(HL)
40A0 77 LD (HL),A
40A1 2A 0C 40 LD HL,(400C)
40A4 01 B6 02 LD BC,02B6
40A7 09 ADD HL,BC

```

```

40A8 22 82 40 LD (4082),HL
40AB 01 21 00 LD BC,0021
40AE 09 ADD HL,BC
40AF 54 LD D,H
40B0 5D LD E,L
40B1 2A 82 40 LD HL,(4082)
40B4 01 B5 02 LD BC,02B5
40B7 ED B8 LDDR
40B9 00 NOP
40BA 21 D1 46 LD HL,46C9
40BD 0E 0A LD C,0D
40BF 09 ADD HL,BC
40C0 AF XOR A
40C1 BE CP (HL)
40C2 C0 RET NZ
40C3 35 17 LD (HL),17
40C5 0E 00 LD C,00
40C7 C9 RET

```


“STUDIO DI FUNZIONE”

dopo l'assaggio del primo numero, un programma, applicativo di matematica per le ultime classi delle scuole superiori

di Stefano Agresti

I grafici sono entrati a far parte della nostra vita in maniera prepotente, non possiamo evitarli, tanto che ce li ritroviamo sotto gli occhi riarrangiati, rivisti, rinnovati, nei giornali, nella pubblicità, come se alcuni argomenti assumessero un aspetto più importante nel momento in cui c'è un grafico a loro sostegno. Niente da obiettare comunque al loro ingresso in vari campi, più o meno tecnici, anche se in effetti, agli occhi assonnati del sig. Rossi che guarda distrattamente la TV essi possono sembrare voli pazzeschi di zanzare traccianti. E' certo però, che un dentifricio che si presenti con la curva che ne indichi l'assorbimento da parte delle gengive ha il suo fascino.

Il programma presentato è in grado di produrre grafici di funzioni matematiche dagli andamenti più disparati, ovviamente che abbiano un campo di esistenza nei numeri reali. Oltre alla curva, dà informazioni sulla simmetria della $f(x)$, la tabulazione in un intervallo scelto, l'identificazione di intervalli in cui cambia di segno, il calcolo degli zeri.

Premesse

Alcune considerazioni, prima di mettere le mani sulla tastiera: 1/ il programma non è in grado di stabilire l'insieme di definizione della funzione, che va quindi cercato a priori per scegliere l'intervallo di studio; se si cerca, per esempio, di tracciare la funzione

$$y = \ln x$$

tra -10 e +10, ovviamente, non essendo definito il logaritmo di un numero negativo, si ottiene il messaggio "Invalid Argument". Quindi attenzione con LN, arcsin, arccos, radici n -esime.

2/ Lo Spectrum non è in grado di valutare espressioni del tipo x^n con x minore di zero: l'ostacolo va aggirato cambiando forma all'espressione, in modo che l'argomento sia sempre positivo; per far ciò, bisogna distinguere tra n pari e n dispari:

$$\text{con } n \text{ pari } f(x)^n = \text{SGN } f(x) = \text{ABS } f(x)^n$$

$$\text{con } n \text{ dispari } f(x)^n = \text{ABS } f(x)^n$$

Per i radicali è la stessa cosa: nel caso di n pari sono definiti solo per argomenti maggiori o uguali a zero, ma per n dispari occorre procedere come sopra:

$$(fx)^{1/3} = \text{SGN } (fx) = \text{ABS } (fx)^{1/3}.$$

Tenedo conto di ciò, questo programma è in grado di tracciare qualunque grafico. C'è da tenere presente che nel calcolo si può andare a pescare un x in cui $f(x)$ non è definita, caso tipico un asintoto verticale. Per evitare il blocco del programma, è stato aggiunto ad ogni x un valore casuale, che modifica la nona cifra decimale e quindi evita l'overflow.

La breve subroutine in linguaggio macchina serve per trasferire uno schermo in memoria e per poterlo richiamare (v. listato). Questa routine è caricata tramite la stringa $a\$,$ contenente le istruzioni in decimali: il caricamento avviene ad ogni RUN, ma una volta fatto potete salvare la routine con SAVE "movevideo" CODE 58400,24 per poterla riutilizzare qui o altrove. RAND.USR 58400 sposta l'immagine dal video alla memoria, RAND.USR 58412 esegue l'operazione inversa, il tutto grazie all'istruzione LDIR, che serve appunto a trasferire blocchi di dati; vengono spostati i contenuti di 6912

locazioni a partire da 16384 (inizio display-file) e da 58432 (scelta in modo da terminare il più possibile vicino al RAMTOP).

Tutto ciò porta a un notevole ingombro di memoria, per cui non è possibile far girare questo programma sullo Spectrum 16K.

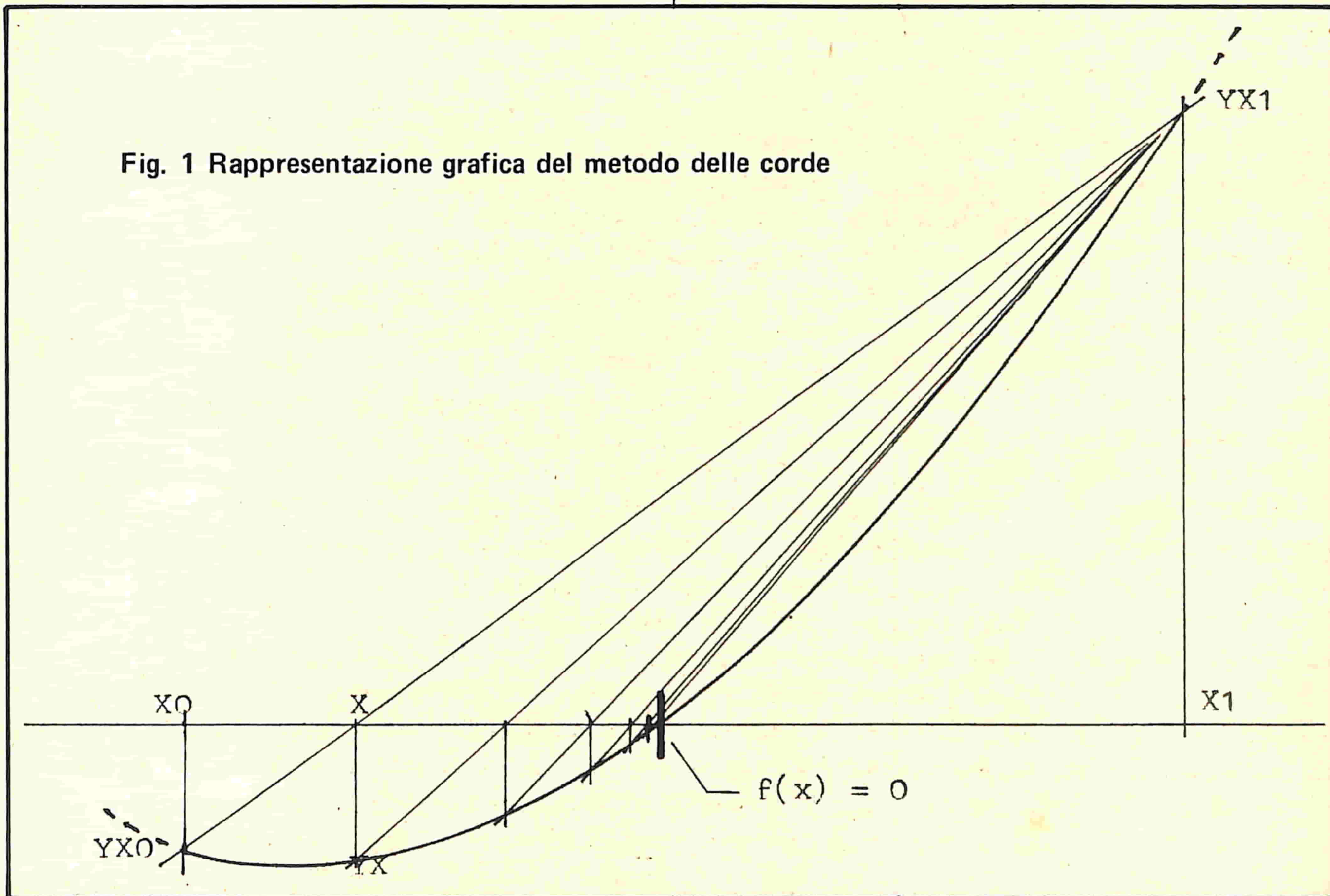
Calcolo degli zeri e metodo delle corde

Per il calcolo degli zeri della funzione è stato usato il metodo di bisezione, seguito dal metodo delle corde modificato. Qualun-

que metodo si usi, è necessario fornire un intervallo di partenza in cui la funzione cambi segno una a una sola volta e in cui vi sia uno e un solo zero.

L'individuazione di questi intervalli è già stata fatta precedentemente, però attenzione: non

Fig. 1 Rappresentazione grafica del metodo delle corde



andate a cercare uno zero nel punto in cui vi è un cambiamento di segno causato da un asintoto verticale dispari; occhio quindi al grafico.

Gli intervalli sono rilevati direttamente, durante il tracciamento della curva, e memorizzati in un vettore $z(n)$.

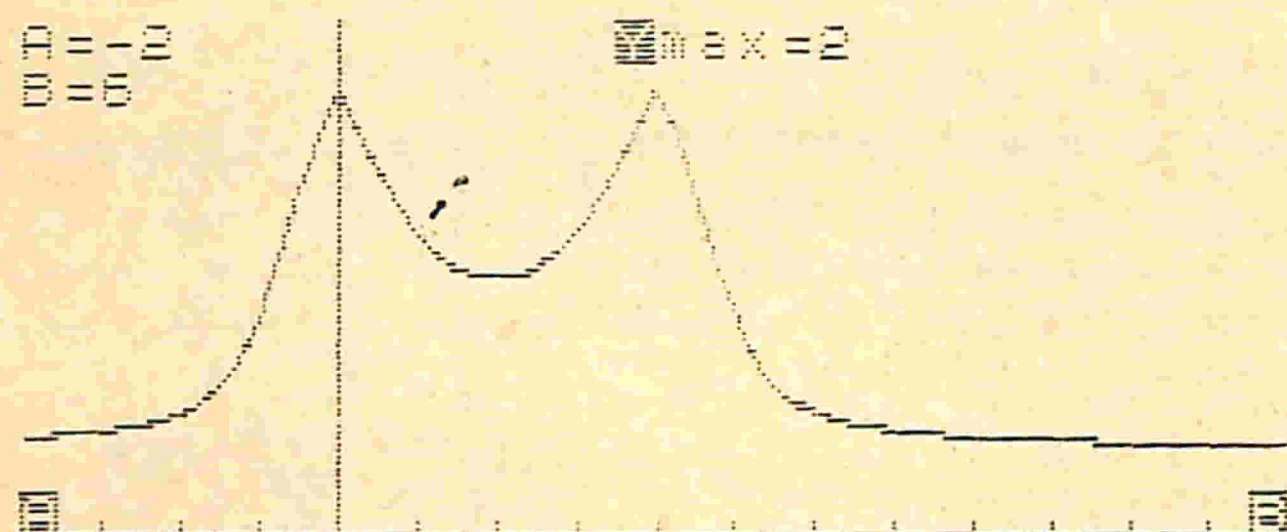
La bisezione consiste nel rendere l'intervallo in cui si trova la

soluzione il più piccolo possibile, fino al limite della precisione di macchina, che per lo Spectrum è dell'ordine di $E-10$. L'algoritmo si basa sul dividere continuamente a metà l'intervallo, in maniera opportuna, e lo si può capire in modo diretto, seguendo il flusso dalla linea 4508 a 4518.

Fatto ciò, si passa al metodo delle corde (fig. 1). Si sostituisce

alla curva la retta che unisce (X_0, YX_0) con (X_1, YX_1) . Determinato X , ci si pone di nuovo nel caso di un intervallo che contiene lo zero, cioè si prende $X_0 = X$ e $YX_0 = YX$ se $SIGN(YX_0) = SIGN(YX)$, oppure $X_1 = X$, $YX_1 = YX$ se $SIGN(YX_1) = SIGN(YX)$ e si ripete il tracciamento della retta; si continua così fino a che YX non è sotto una certa

A=-2
B=6



$f(x) = .5 \cdot \text{ATN} (\text{ABS} (x \cdot x - 2 \cdot x) - 1)$

Se c'è overflow dai GO TO 3000
TABULAZIONE:

X	f(x)
-2	0.37141416
-1.2	0.42555585
-0.4	1.0280987
0.4	1.2706402
1.2	1.0280986
2	1.7235679
2.8	0.53881658
3.6	0.38857806
4.4	0.36184084
5.2	0.3518556
6	0.34691511

3 GO TO 100: REM una volta caricato il programma si salva con
GO TO 5

4 REM **SAVE AND GO ROUTINE**
5 SAVE "fxstudy" LINE 5000: 6

TOP

19 REM **FXSTUDY**
20 REM
21 REM di AGRESTI STEFANO
22 REM spectrum 48 K

23 REM
29 REM modifica del 13/10/1983

30 REM **DEF. PARAMETRI PRINCIP.**

34 REM x=var. indep.
A,B=estremi dell'interv.in esame

Ymax=ordinata max rappresentab.
y\$=contiene f(x)
eps=precisione di macchina
zeri=1 f(x) ha zeri (0 non li ha)
tsc=1 già eseguita routine di tracciamento cambi segno (0 non eseg.)
break=1 segnala interr. grafico
fatto1=1 " fatto1 scelta A,B
ind=0 f(x)=pari; ind=1 dispari
scz=1 eseg.sub.cercazeri (0 no)
36 REM c\$=stringa ricorrente
ad= ampiezza divisioni asse X
nd= numero divisioni asse X
rnd=e'usato per modif. la nona cifra decim.in modo da evitare overflow, nel grafico, per asintoti
xn,yn=valori da plottare legati alla risoluzione video
A\$,B\$=memorizzano risposte (s/n)
pippo=aggiusta divisione asse X
tol=tolleranza int.min bisezione
ytol= " " y min in cercazeri
passo=passo di tabulazione
37 REM x0,x1=interv.di ricerca di zeri per bisezione e corde
yx0,yx1=corrispondenti val. f(x)
DEFINIZIONE VETTORI:
Z(n)=contiene gli intervalli in cui f(x) cambia segno

100 REM **inizio programma**
150 REM
190 POKE 23609,50: REM beep!
200 BORDER 1: INK 7: PAPER 1: B
RIGHT 0: CLS
250 PRINT AT 0,0;"Con questo programma e' possibile studiare automaticamente una qualunque f(x); il grafico puo'essere espanso o compresso modificando (A,B) e Ymax (dimensioni del grafico)"

252 PRINT "E' da notare che l'intervallo (A,B), deve essere specificato con attenzione considerando il campo di esistenza della f(x)."

254 PRINT "Inoltre: una espressione elevata a potenza dispari del tipo $[f(x)]^n$ va scritta come: $\text{SGN } f(x) \cdot \text{ABS } f(x)^n$; se invece n e' pari allora scrivi: $\text{ABS } f(x)^n$; (per x^2 basta porre: $x \cdot x$)"

256 PRINT "Lo stesso discorso si applica ad argomenti di radici"

257 PRINT "Se il programma dovesse bloccarsi per un qualsiasi motivo dai GO TO 300"

258 PRINT "Esempio di f(x): $x \cdot \sin(3 \cdot x)$ "

270 REM **Def. f(x), A,B,Ymax**

280 INPUT AT 0,0: PAPER 3;"SCRIVI ORA L'ESPRESSIONE DI f(x)"; LINE y\$: IF y\$="" THEN GO TO 270

290 LET A=-10: LET B=-A: LET Ymax=10


```

299 CLS
310 PRINT AT 5,0; FLASH 1; PAPE
R 2; INK 7;"f(x)=";y$
320 LET fatto1=0; PRINT AT 12,0
;"Disegno questa funzione nell'i
n-tervallo:";A;"A=";A;"B=";B;
"con";Ymax;"Ymax";
330 GO TO 575
350 REM 0.K. + eps
362 PRINT AT 11,13;"0.K."
363 PRINT AT 12,13;"0.K."
364 PRINT AT 13,13;"0.K."
366 LET eps=1; REM Calcolo dell
a precisione di macchina: eps=
1.1615322E-10
367 LET eps=eps/2; IF (1+eps<>1
) THEN GO TO 367
368 BEEP 2.5,30
370 REM inizializzazione
375 RANDOMIZE; LET zer1=0; LET
tcs=0; DIM z(255); REM azera z
380 LET rnd=(INT (RND*100)+1)*
eps*2; REM evita asintoti vert.
390 REM generazione grafico f(x)
395 INK 0; PAPER 7; CLS; LET c
$="Premi i per interrompere"; PR
INT PAPER 3; INK 7;c$;AT 20,0; P
APER 2;"REM:GOTO 300"
397 POKE 23560,0
400 FOR n=1 TO 255
410 LET x=A+(B-A)*n/255
415 LET x=x+rnd-(2*rnd AND x>=B
); LET yx=VAL y$. REM valore di
y in: x+random
418 LET yn=88+yx*87/Ymax
420 IF n=1 THEN GO TO 490
430 IF ynm1<0 OR ynm1>175 THEN
GO TO 450
435 PLOT n-1,ynm1; IF yn>=0 AND
yn<=175 THEN DRAW 1,yn-ynm1
440 IF yn>175 THEN DRAW 1,175-y
nm1
445 IF yn<0 THEN DRAW 1,-ynm1
450 IF yn<0 OR yn>175 THEN GO T
O 480
455 PLOT n,yn; IF ynm1>175 THEN
DRAW -1,175-yn
460 IF ynm1<0 THEN DRAW -1,-yn
480 IF ynm1*yn<0 AND ABS (yn-yn
m1)>175 THEN: FOR j=0 TO 12: PL
OT n,13+j; DRAW INK 2,0,8; NEXT
j; REM traccia asintoti dispari
485 IF yx*yxnm1=0 THEN LET zer
i=1; LET z(n-1)=xnm1; LET z(n)=x
; REM f(x) cambia segno?
490 LET ynm1=yn; LET yxnm1=yx
492 LET xnm1=x
495 LET break=0; LET a$=CHR$ PE
EK 23560; IF a$="i" THEN LET bre
ak=1; GO TO 575
500 NEXT n
510 REM tracciamento assi
515 PRINT AT 0,0;: LET asse=0
520 PLOT 0,88; DRAW 255,0
530 LET nd=B-A; IF A*B<=0 THEN
LET asse=-A/nd*255; PLOT asse,0;
DRAW 0,175

```

```

531 IF nd<5 THEN LET nd=nd*10:
GO TO 531
532 IF nd>60 THEN LET nd=nd/10:
GO TO 532
534 IF nd<10 THEN LET nd=nd*2:
GO TO 534
535 IF nd>30 THEN LET nd=nd/2:
GO TO 535
538 LET ad=255/nd
539 LET pippo=asse/ad; LET pipp
o=(INT pippo-pippo)*ad
540 PRINT AT 20,0;AT 20,0;"f(x)
=";y$;AT 10,0;"@";AT 10,31;"@";
AT 0,15;"Ymax"
549 FOR n=0 TO nd: PLOT ad*n+pi
ppo,88; DRAW 0,2; NEXT n
550 BEEP 1,10
551 PRINT OVER 1;AT 0,0;"A=";A;
AT 1,0;"B=";B;AT 0,19;"=";Ymax
555 REM traccia cambi di segno
560 IF zer1=0 OR tcs=1 THEN GO
TO 575
565 FOR n=1 TO 255
568 IF z(n)<>0 THEN LET n=n+1:
PLOT INK 2,n,88; DRAW INK 2,0,10
570 NEXT n; LET tcs=1
575 REM Modifica (A,B) ,Ymax
580 IF fatto1=1 THEN: INPUT AT
0,0;"Vuoi cambiare f(x)?(enter
s/n)"; LINE a$: IF a$="s" THEN G
O TO 200
585 INPUT AT 0,0;"Vuoi modifica
re(A,B)?(enter:s/n)"; LINE A$: I
F A$="s" THEN INPUT "A=";A;"B=";
B; PRINT AT 15,0;"A=";A;"B=";B;
590 INPUT AT 0,0;"Vuoi modifica
re Ymax?(enter:s/n)"; LINE B$: I
F B$="s" THEN INPUT "Ymax=";Ymax
; PRINT AT 19,0;"Ymax=";Ymax;
592 IF A$="s" OR B$="s" OR fatt
o1=0 THEN LET fatto1=1; GO TO 36
0
595 IF break=1 THEN PRINT AT 0,
0; PAPER 6;"Ma allora che vuoi?!
. Fammi continuare!"; BEEP .5
,30; BEEP 3,7; PRINT AT 0,0; PAP
ER 3; INK 7;c$;: PRINT "
"; GO TO 500
1050 INPUT AT 0,0;"Vuoi una stam
pa ? (enter s/n)"; LINE a$: IF
a$="s" THEN COPY
1052 RANDOMIZE USR 56400; REM sa
lvavideo
1060 REM Valutazione pari, disp.
1065 LET ind=-1; IF A>=0 THEN GO
TO 1100
1070 LET x=A+RND*ABS (B-A); LET
yx=VAL y$
1071 LET x=-x; LET ymx=VAL y$
1080 IF yx=-ymx THEN LET ind=1
1085 IF yx=ymx THEN LET ind=0
1100 REM Menu
1110 POKE 23560,0; CLS; PRINT P
APER 1; INK 7;"Quello che posso
fare per te ad-esso e:";: PRIN
T "1)Tabulare la f(x)
nel-
l'intervallo scelt

```


“TRASFERIMENTO D'UNA PAGINA VIDEO”

Memorizzare e richiamare una pagina di video è un'operazione molto frequente, non solo nei giochi. Se dovessimo eseguirla in basic, richiederebbe un tempo eccessivo, specialmente con lo Spectrum, dove il display file occupa quasi 7K; l'autore di "FXSTUDY" propone una soluzione in 1/m per lo Spectrum, che può essere ulteriormente abbreviata, evitando di ripetere le ultime tre istruzioni, sostituendole con un salto relativo JR(e); le locazioni da cambiare sono:

```
32018 18 F2 JR 32006
32020 00 NOP
```

Per lo ZX81, occorre sostituire un paio di istruzioni, perchè il display-file non ha un indirizzo fisso. La routine è dislocata, come al solito, in una REM in apertura di programma; l'indirizzo di inizio della memoria video viene cercato ad ogni chiamata nella variabile di sistema D-FILE (16396/16397). Nell'esempio la pagina di schermo viene memorizzata a partire da 7000 hex (28672 dec), ma potete metterla altrove, così come potete memo-

rizzare più di una pagina. Consigliabile sempre abbassare il RAM-TOP e archiviare sopra questo limite.

Per chi è alle prime armi con il linguaggio macchina, diamo alcune spiegazioni sull'uso e il funzionamento della routine: nel registro HL si pone l'indirizzo di inizio della memoria video; nel reg.DE quello da cui si vuole iniziare la memorizzazione. Il numero posto in BC, pari alla lunghezza in bytes della memoria video, funge da contatore. La funzione LDIR ricopia il contenuto dell'indirizzo che si trova in HL all'indirizzo che si trova in DE,

quindi decrementa di uno HL, DE e contatore in BC e ripete l'operazione finchè BC non è zero. Per richiamare al video i dati archiviati, si esegue esattamente la stessa operazione, scambiando di posto tra loro gli indirizzi in HL e DE.

Gli indirizzi di chiamata sono: 16514 (archivia) e 16526 (richiama) per lo ZX81; (nn) e (nn+12) per lo Spectrum, con nn=indirizzo da cui iniziate a caricare i codici. Per questa operazione vi rimandiamo ai manuali dei rispettivi computer, che contengono entrambi esempi di hex-loader (caricatore di esadecimale).

```
16514 2A 00 40 LD HL,(16396)
16517 11 00 70 LD DE,28672
16520 01 18 03 LD BC,792
16523 ED B0 LDIR
16525 C9 RET
16526 21 00 70 LD HL,28672
16529 ED 5B 0C 40 LD DE,(16396)
16533 18 F1 JR 16520
```

Versione ZX81
List. 1

```
32000 21 00 40 LD HL,16384
32003 11 40 E4 LD DE,33000
32006 01 00 1B LD BC,6912
32009 ED B0 LDIR
32011 C9 RET
32012 21 40 E4 LD HL,33000
32015 11 00 40 LD DE,16384
32018 18 F2 JR 32006
```

Versione Spectrum
List. 2


```

0"
1111 PRINT ,,"      2)Valutare gli
   intervalli      all'interno d
   ei quali        la f(x) cambi
   a segno         (e ricerca di
   zeri)"
1112 PRINT ,,"      3)Farti rivede
   re il grafico   e modificare
   A,B,Ymax,f(x)"
1120 PRINT AT 21,0; PAPER 3; INK
   7;"Scegli e premi uno dei numer
   i."
1121 IF ind=1 THEN PRINT AT 18,0
; "NOTA la f(x) e' simmetrica ri
s-petto all'origine"
1122 IF ind=0 THEN PRINT AT 18,0
; "NOTA la f(x) e' simmetrica ri
s-petto all'asse y"
1130 LET a$=CHR$ PEEK 23560: IF
a$="1" THEN GO SUB 3000
1131 IF a$="2" THEN GO SUB 4000
1132 IF a$="3" THEN CLS : RANDOM
IZEUSR 58412: GO TO 555: REM lo
advideo
1133 GO TO 1130
3000 REM Tab. grafico
3005 CLS : PRINT PAPER 2; INK 7;
"Se c'e' overflow dai " FLASH
1;"GO TO 3000": FLASH 0;" TABUL
AZIONE:" , , TAB 4;"X";TAB 18;"f(x)"
"
3007 LET passo=(B-A)/10
3010 FOR v=A TO B STEP passo
3020 LET rnd=(INT (RAND*100)+1)*e
ps#2: LET x=v+rnd-(2*rnd AND v>=
B): PRINT TAB 3;" "v;TAB 18;VAL
y$
3030 NEXT v
3040 INPUT "enter un ☐ che ti in
teressa, ☐ per uscire, oppure ☐
per stampare " : LINE a$: IF a
$="q" THEN GO TO 1100
3045 IF a$="p" THEN COPY : GO TO
3040
3048 IF a$="" OR a$>"9" AND a$<>
"q" THEN GO TO 3040
3050 LET x=VAL a$: PRINT TAB 4;x
;TAB 18;VAL y$: GO TO 3040
4000 REM Valut. cambi segno
4004 CLS : LET scz=0: PRINT PAPE
R 2; INK 7;"Intervalli entro i q
uali si ve-rifica un cambiament
o di segno per f(x)" , , "( gia' i
ndicati nel grafico )"
4005 IF zeri=0 THEN PRINT AT 10,
0; PAPER 3; INK 7;"NESSUN CAMBIA
MENTO DI SEGNO E' RILEVATO IN Q
UESTO INTERVALLO": BEEP 4,15: G
O TO 1100
4010 FOR n=1 TO 255
4012 IF z(n)<>0 THEN PRINT , , TAB
10;z(n): LET n=n+1: PRINT TAB 1
0;z(n)
4020 NEXT n: INPUT AT 0,0;"Vuoi
una copia (enter s/n)": LINE a$:
IF a$="s" THEN COPY
4028 INPUT AT 0,0;"Calcolo uno z

```

```

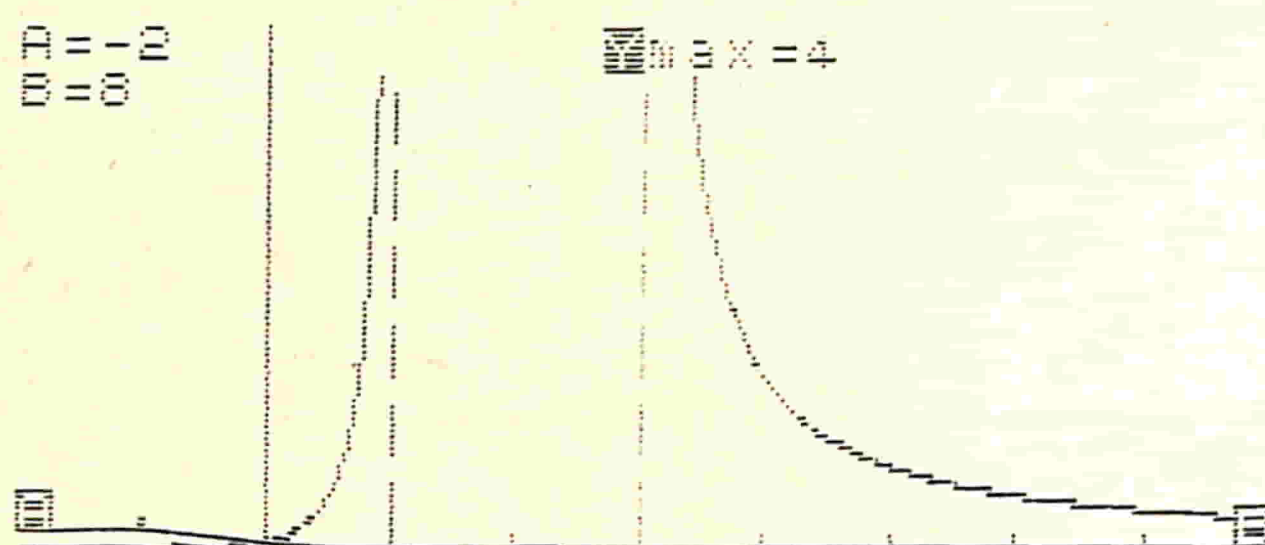
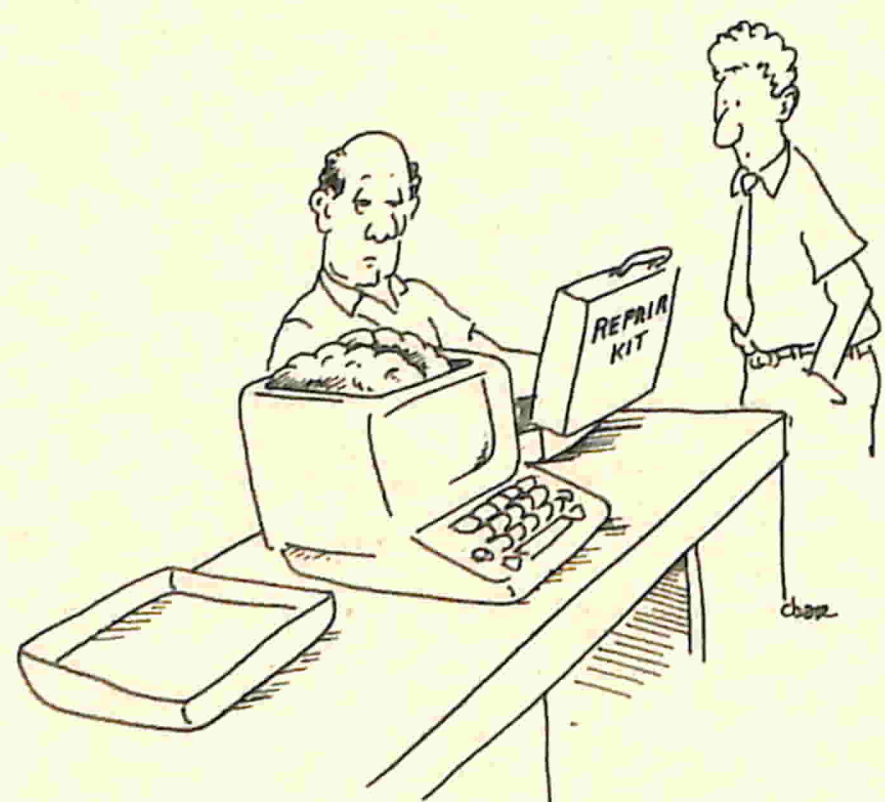
ero di f(x) ?      (enter s/n) "
; LINE a$: IF a$<>"s" THEN GO TO
1100
4030 IF scz=0 THEN PRINT INK 7;
PAPER 3;"Enter un intervallo x0
< x1 in cui vi e' uno e un solo
zero per la f(x), non introdurre
interval-li in cui vi sono asin
toti ver-ticali, se cio' accade=>
": FLASH 1;"GOTO 4030": PRINT IN
VERSE 1;c$;" L'esec."
4040 INPUT "x0=";x0,"x1=";x1
4050 GO SUB 4500
4060 PRINT , , "La f(x) ha uno zer
o nel punto: " : INVERSE 1;zero;
INVERSE 0;" If(x)=";yx;"1"
4065 LPRINT AT 0,0;"f(x)=0 per x
=" ;zero , , , , ,
4070 GO TO 4025
4500 REM Subr. cerca zero
4502 LET tol=eps#2*(ABS x1+1)
4505 LET x=x0: LET yx0=VAL y$: L
ET x=x1: LET yx1=VAL y$: LET yto
l=(1+ABS (yx1-yx0))*tol
4508 REM Bisezione
4509 LET xp=x0
4510 LET x=(x0+x1)/2: LET yx=VAL
y$: IF ABS (x1-x0)<=tol OR x=xp
THEN GO TO 4520
4517 LET xp=x: IF yx*yx1>0 THEN
LET x1=x: LET yx1=yx: GO TO 4510
4518 LET x0=x: LET yx0=yx: GO TO
4510
4519 REM Carde modificato
4520 LET yxp=yx0
4530 IF ABS yx<=ytol*1000 AND yx
=yxp THEN LET zero=x: LET scz=1:
RETURN
4540 IF yx0<>yx1 THEN LET x=(yx1
*yx0-yx0*x1)/(yx1-yx0): LET yx=VA
L y$
4548 LET a$=CHR$ PEEK 23560: IF
a$="i" THEN GO TO 4610
4550 IF yx*yx0>0 THEN LET x0=x:
LET yx0=yx: IF yx*yxp>0 THEN LET
yx1=yx1/2: GO TO 4600
4560 LET x1=x: LET yx1=yx: IF yx
*yxp>0 THEN LET yx0=yx0/2
4600 LET yxp=yx: GO TO 4530
4610 BEEP .5,10
4615 PRINT , , INVERSE 1;"Non rie
sco a trovare nessun zero in que
sto intervallo: il valore a cui
sono arrivato e' " : PRINT "x=";x
; " f(x)=";yx
4620 LET scz=0: GO TO 4025
5000 REM Routine LM movevideo
5010 CLEAR 58399: LET a$="033000
064017064228001000002723717620103
3064228017000006400100000272371762
01"
5020 FOR n=0 TO 23: POKE 58400+n
,VAL a$(3*n+1 TO 3*n+3): NEXT n
5030 GO TO 100
5040 REM si puo' salvare questa
routine in LM con SAVE "savevide
o" CODE 58400,24

```


tolleranza e rimane costante, per la macchina, a ogni iterazione (cioè finchè non abbiamo raggiunto i limiti di precisione del computer). Il metodo delle corde modificato consiste semplicemente nell'unire i punti (X0, YX0), e (X1, YX1/2) (oppure X0, YX0/2) e (X1, YX1): in questo modo la convergenza è più rapida. Se non doveste arrivare a una soluzione, provate a dare un intervallo iniziale più grande (questione di tolleranze).

Quando vi sarete stancati di tracciare le curve più strane, sarebbe una buona idea completare il programma con un metodo per il calcolo approssimato degli integrali (Simpson o regola dei trapezi), così potreste calcolare la quantità complessiva di dentifricio assorbito nei cinque minuti in cui vi lavate i denti. ■

(*Certo che è una buona idea. Caro Agresti. Stiamo aspettando - N.d.r.).



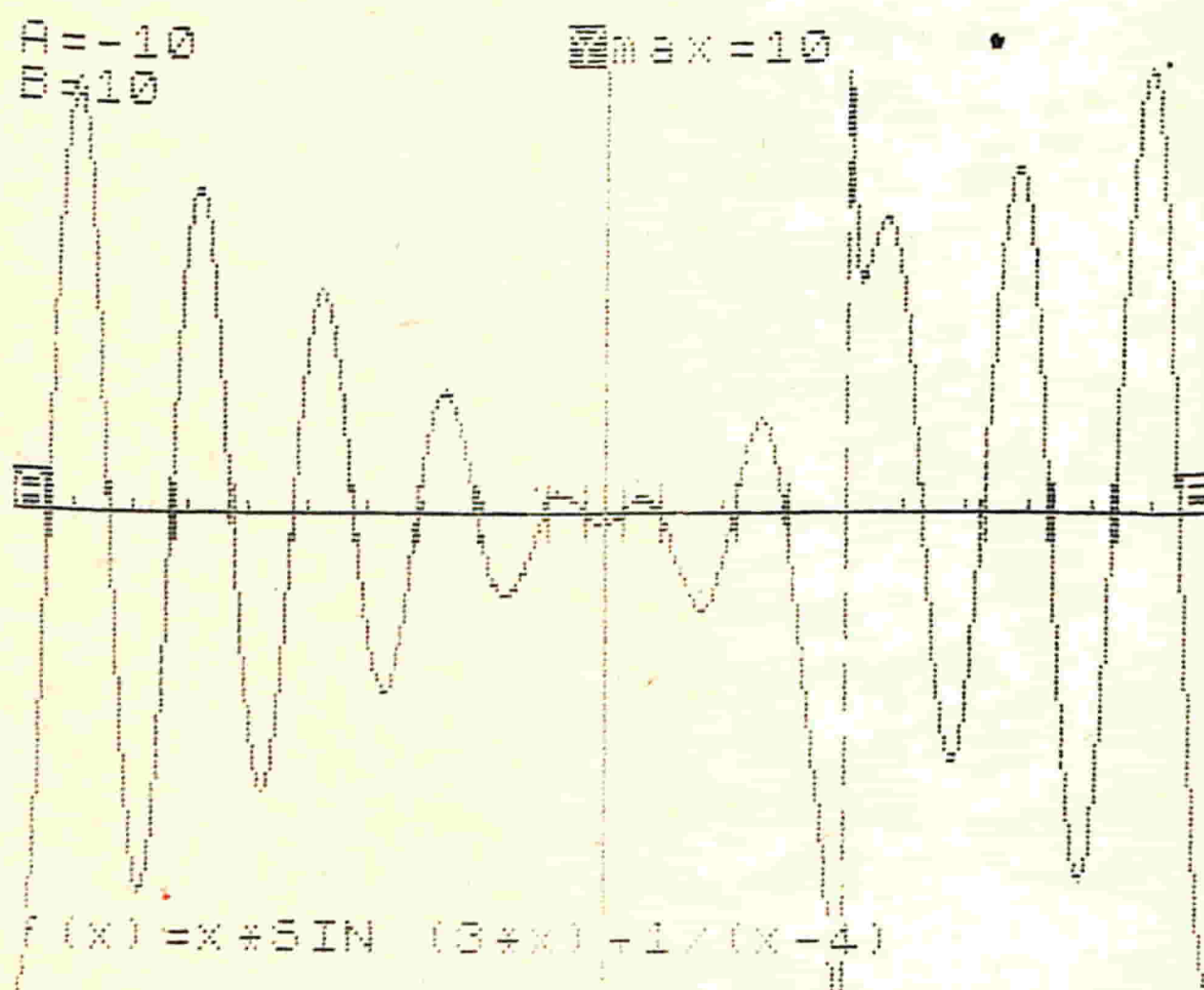
$$f(x) = x / (x^2 - 4x + 3)$$

Intervalli entro i quali si verifica un cambiamento di segno per f(x)
(già indicati nel grafico)

-0.030215682
4.6555129E-9

0.98039216
1.0198078

2.9803922
3.0198078



$$f(x) = x * \sin(3*x) + 1 / (x-4)$$

$$f(x) = 0 \text{ per } x = 3.3684612$$

"SUPER MONITOR"

una utility per leggere e scrivere facilmente nella memoria dello Spectrum

di Massimo Rossi

QUESTO programma è nato dall'esigenza di avere un monitor esadecimale che permettesse di leggere rapidamente il contenuto di una serie di locazioni di memoria, e che desse anche la possibilità all'utilizzatore di scrivere nelle stesse con facilità.

Infatti, mentre in commercio esistono vari tipi di programmi che assemblano, non è facile trovarne qualcuno che permetta di scrivere direttamente nelle celle di memoria che ci interessano. Questo programma è stato realizzato per fornire uno strumento duttile a chi voglia cimentarsi a copiare o a scrivere programmi in linguaggio macchina. Offre parecchie opzioni di indubbia utilità pratica, quali un convertitore decimale-esadecimale (e viceversa), ed altre che vedremo in seguito.

Tutti gli indirizzi in input possono essere dati sia in decimale, che in esadecimale (questi ultimi devono essere a lettere maiuscole e preceduti dal segno \$); il display è suddiviso in tre colonne:

- l'indirizzo del primo byte della riga (zona a fondo azzurro);
- i contenuti dei sei byte, in successione orizzontale (zona a fon-

```

0) Super Monitor
   ©1984 Massimo Rossi

5 LET pa=.9: LET fl=0: LET at
tr=7: LET t$="": LET x=0: LET y=
0: LET e$="0123456789ABCDEF"
80 CLS: GO SUB 6110
90 GO SUB 7000: LET attr=7
92 IF x<4 THEN LET attr=5: GO
TO 95
93 IF x>24 THEN LET attr=4
95 LET xx=x: LET yy=y
100 IF CODE INKEY$=11 AND y>0 T
HEN LET y=y-1: GO SUB 6100
110 IF CODE INKEY$=10 AND y<21
THEN LET y=y+1: GO SUB 6100
120 IF CODE INKEY$=8 AND x>0 TH
EN LET x=x-1: GO SUB 6100
130 IF CODE INKEY$=9 AND x<30 T
HEN LET x=x+1: GO SUB 6100
137 IF CODE INKEY$=226 THEN INP
UT "TRANSFER Start":a$: GO SUB
3000: LET ss=num: INPUT "TRANSFE
R End":a$: GO SUB 3000: LET se=n
um: INPUT "Nuovo Indirizzo":a$:
GO SUB 3000: LET ni=num: FOR i=s
s TO se: POKE i-ss+ni,PEEK (i)
NEXT i
140 IF CODE INKEY$=172 THEN INP
UT "Nuovo Start":a$: GO SUB 300
0: LET ns=num: GO SUB 4000: GO S
UB 6100
142 IF CODE INKEY$=195 THEN COP
Y: LET ch=0: FOR g=ns TO ns+132
: LET ch=ch+PEEK (g): NEXT g: LP
RINT "CHECK = ":ch

```

do bianco);

- il significato del contenuto dei sei byte in codice ASCII (zona a fondo verde).

Al momento del RUN, apparirà sullo schermo un quadratino luminoso: il cursore. Può essere

spostato nelle quattro direzioni premendo CAPS SHIFT insieme al tasto della prima fila (5,6,7,8) la cui freccia corrisponde alla direzione desiderata.

A questo punto possiamo decidere quali byte desideriamo


```

150 LET code=CODE INKEY$: IF co
de>=32 AND code<=164 AND x>=24 T
HEN POKE 23658,0: PRINT BRIGHT 1
;AT y,x;INKEY$: PAUSE pa: POKE 2
3658,8: IF x<30 THEN LET x=x+1:
GO SUB 6100
155 LET code=CODE INKEY$: IF co
de>=48 AND code<=57 AND x<=24 OR
code>=97 AND code<=102 OR code=
32 THEN POKE 23658,8: PRINT BRIG
HT 1; PAPER attr;AT y,x;INKEY$:
PAUSE pa: POKE 23658,0: IF x<30
THEN LET x=x+1: GO SUB 6100
160 IF CODE INKEY#=13 THEN GO S
UB 3000
170 IF CODE INKEY#=204 THEN LET
ns=ns+132: GO SUB 4000
180 IF CODE INKEY#=205 THEN LET
ns=ns-132: GO SUB 4000
190 IF CODE INKEY#=198 THEN INP
UT "SAVE Start ";a$: GO SUB 3000
: LET ss=num: INPUT "SAVE End";a
$: GO SUB 3000: LET se=num: INPU
T "TITOLO";a$: SAVE a$CODE ss,se
-ss
200 IF CODE INKEY#=197 THEN INP
UT "Numero da Convertire";a$: GO
SUB 3500
500 GO TO 90
2999 REM input esad dec
3000 IF a$(1 TO 1)="#" THEN GO T
O 3020
3010 LET num=VAL a$: RETURN
3020 LET n#=a$(2 TO 5): GO SUB 6
040: LET num=dec: RETURN
3499 REM conversione
3500 IF a$(1 TO 1)="#" THEN GO T
O 3520
3510 LET nr=VAL a$: GO SUB 5050:
OPEN #5,"k": PRINT #5;r$: PAUSE
0: CLOSE #5: GO TO 90
3520 LET n#=a$(2 TO 5): GO SUB 6
040: OPEN #5,"k": PRINT #5;dec:
PAUSE 0: CLOSE #5: GO TO 90
4000 REM riempie pagina
4010 CLS : FOR v=0 TO 21
4020 LET nr=ns+v*6: GO SUB 5045:
PRINT PAPER 5;AT v,0;r$
4030 FOR o=6 TO 21 STEP 3
4040 LET dec=PEEK ((ns+v*6)+(o/3
)-2): LET t=dec: GO SUB 5000: PR
INT AT v,o;r$
4045 IF t>164 OR t<32 THEN LET t
=63
4047 LET t#=t#+CHR$ t
4050 NEXT o
4055 PRINT PAPER 4;AT v,25;t$: L
ET t#=""
4057 IF fl=1 THEN LET fl=0: GO S
UB 6110: GO TO 90
4060 NEXT v
4070 RETURN
5000 REM dec esad
5010 LET d=INT (dec/16): IF d<1
THEN LET d=0
5020 LET dd=dec-16*d

```

```

5030 LET r#=e$(d+1 TO d+1)+e$(dd
+1 TO dd+1)
5040 RETURN
5045 REM dec esad big
5050 LET hi=INT (nr/256): LET lo
=nr-256*hi
5060 LET dec=lo: GO SUB 5000: LE
T l#=r$: LET dec=hi: GO SUB 5000
: LET r#=r#+l$
5070 RETURN
6000 REM esad dec
6010 LET a#=r$(2 TO 2): FOR u=1
TO 16: IF a#=e$(u TO u) THEN GO
TO 6020
6015 NEXT u
6020 LET a#=r$(1 TO 1): FOR d=1
TO 16: LET q#=e$(d TO d): IF a#=
q$ THEN GO TO 6030
6025 NEXT d
6030 LET dec=(d-1)*16+u-1: RETUR
N
6040 REM esad dec big
6050 LET r#=n$(1 TO 2): GO SUB 6
000: LET h=dec: LET r#=n$(3 TO 4
): GO SUB 6000: LET l=dec
6070 LET dec=h*256+l: RETURN
6099 REM cursore
6100 PRINT AT yy,xx: BRIGHT 0; P
APER attr;SCREEN$ (yy,xx);
6110 PRINT AT y,x: BRIGHT 1;SCRE
EN$ (y,x);
6120 RETURN
7000 REM print comandi
7010 OPEN #5,"k": PRINT #5;AT 0,
0: PAPER 2;; INK 7;"I=INPUT/S=CO
PY/F=AVANTI/D=INDIE-TRO/Y=SAVE/A
=SHIFT/U=CONVERSIONE"
7020 RETURN
7999 REM enter numeri
8000 BEEP .5,5
8010 LET n#=""
8015 FOR i=0 TO 3: LET a#=SCREEN
$ (y,i): LET n#=n#+a$
8017 NEXT i: GO SUB 6050 LET de
=dec
8019 IF x>=24 THEN GO TO 8500
8020 FOR o=6 TO 21 STEP 3
8030 LET r#=SCREEN$ (y,o): LET y
#=SCREEN$ (y,o+1): LET r#=r#+y$:
GO SUB 6000
8040 POKE de+(o/3)-2,dec
8050 NEXT o
8060 GO TO 8520
8499 REM enter lettere
8500 FOR i=25 TO 30
8505 LET po=CODE SCREEN$ (y,i):
IF po=63 THEN NEXT i
8510 POKE de+i-25,po
8520 NEXT i: LET v=y: LET fl=1:
IF y<21 AND x<=24 THEN LET y=y+1
: LET x=6: GO SUB 6100
8525 IF y<21 AND x>=24 THEN LET
y=y+1: LET x=25: GO SUB 6100
8530 GO SUB 7000: GO TO 4030
9999 CLS : PRINT CODE INKEY$: GO
TO 9999

```


esplorare: una volta stabilito l'indirizzo di inizio, premiamo SYMBOL SHIFT e "I" (input), e vedremo apparire sullo schermo la scritta:

Nuovo Indirizzo:
ora dovremo scrivere l'indirizzo di partenza in decimale, oppure in esadecimale (ricordando, in questo caso, di premere il simbolo "\$" e di scrivere le lettere in maiuscolo - ad es: \$5B0A).

Una volta premuto ENTER, lo schermo si riempirà di caratteri, e il cursore sarà libero di muoversi nelle quattro direzioni.

Come scrivere nella memoria

Se vogliamo cambiare il contenuto di qualche locazione, non dobbiamo far altro che portare il cursore sopra al byte desiderato, e premere il tasto corrispondente al numero o alla lettera (anche qui maiuscola) che vogliamo scrivere.

Terminata l'elaborazione di ogni riga, dovremo semplicemente premere ENTER, dopo aver posizionato il cursore nella zona bianca, se abbiamo corretto dei byte, o nella zona verde, se abbiamo modificato delle scritte.

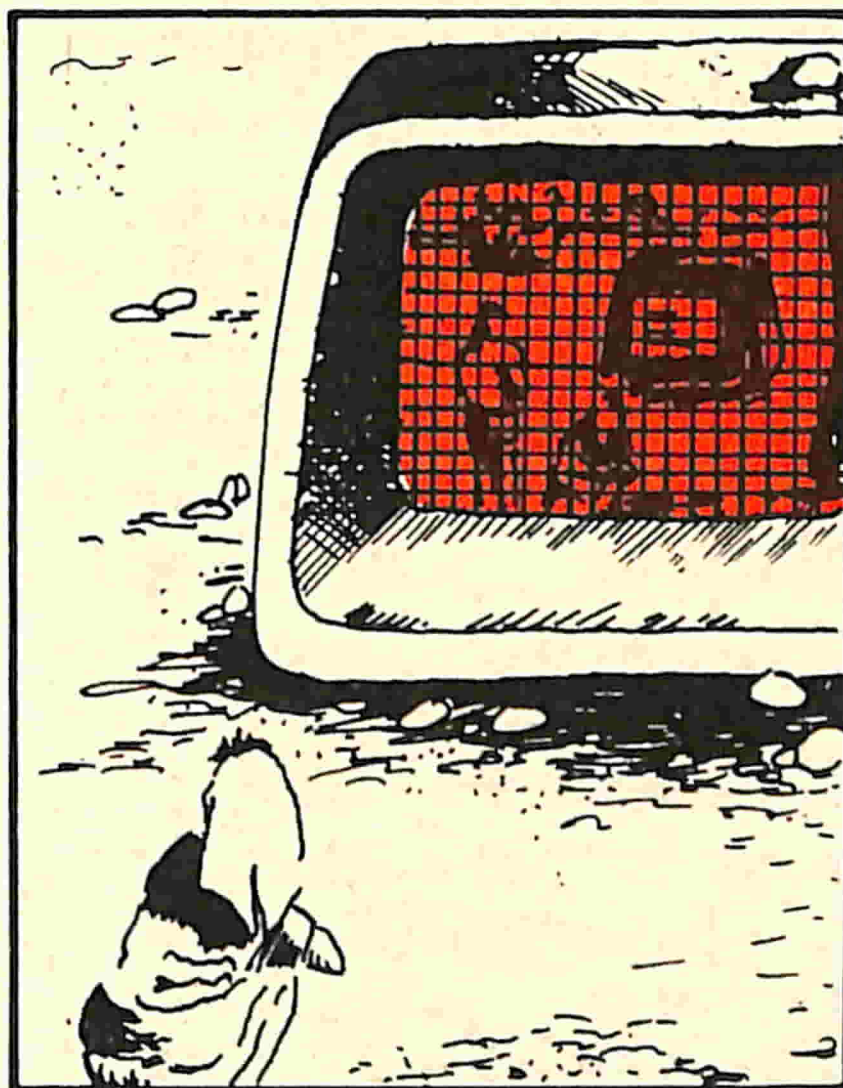
Vi consigliamo di premere bene i tasti, poichè, essendo il programma in basic, lo Spectrum ha un tempo di reazione abbastanza lungo. Possiamo quindi:

a/ cambiare la locazione delle celle di memoria raffigurate, modificando l'indirizzo scritto all'inizio della riga (in campo azzurro);

b/ cambiare i contenuti delle cel-

le di memoria stesse, come spiegato sopra;

c/ cambiare delle scritte, portando il cursore sopra alla zona verde (quella più a destra), e scrivendo direttamente le lettere nella posizione corrispondente alle locazioni che ci interessano.



Quest'ultima possibilità è di grande utilità per chi desidera modificare o tradurre i messaggi che appaiono in un programma in linguaggio macchina: basterà visionare il programma stesso, finchè non appariranno le scritte, e quindi inserire le variazioni, mantenendo il cursore nella zona verde, e premendo ENTER alla fine di ogni riga.

Il cursore riapparirà sulla riga successiva, all'inizio della zona in cui si trovava al momento dell'ENTER, e la riga stessa verrà aggiornata con nuovi valori inseriti.

Quando il cursore si trova nella zona bianca e blu, non è possibile usare altro che le lettere MAIUSCOLE che compongono il codice esadecimale, ed è perciò

inutile cercare di premere altri tasti, all'infuori di: ABCDEF0-123456789 e SPACE (quest'ultimo serve per cancellare o saltare da un byte al successivo).

Comandi speciali

Il programma è dotato di una serie di comandi speciali, atti a rendere l'uso della memoria dello Spectrum e le notazioni esadecimali più facili:

- *Symbol Shift + "I"*. Si richiede l'indirizzo di memoria da cui parte la rappresentazione sullo schermo. Esso può venir fornito in termini decimali, senza alcun prefisso, oppure in esadecimale, preceduto da "\$" e scritto a lettere maiuscole.

- *Symbol Shift + "F"*. Ogni pagina di schermo rappresenta 132 locazioni di memoria; se vogliamo proseguire da dove si interrompe la pagina, con questo comando possiamo vedere i successivi 132 bytes.

- *Symbol Shift + "D"*. Questa funzione è l'opposto di quella precedente, cioè mostra i 132 bytes precedenti a quelli rappresentati.

- *Symbol Shift + "U"*. Questo comando permette di convertire un numero esadecimale in un numero decimale, e viceversa: basta immettere il numero esadecimale preceduto da "\$", con lettere maiuscole, e di formato quattro cifre (cioè AB è 00AB), per avere il suo corrispondente decimale; per la funzione invece, si immette il numero senza prefisso.

• *Symbol Shift* + "A". Con questa funzione si trasferiscono interi blocchi di memoria, a patto, naturalmente, che la distanza a cui si trasferiscono sia maggiore del numero stesso dei bytes (a questa limitazione si può ovviare spostando temporaneamente il blocco in un'area diversa, dalla quale si può raggiungere la zona desiderata senza creare sovrapposizioni). Anche qui, i dati potranno essere forniti indifferentemente nelle due notazioni.

• *Symbol Shift* + "Y". Si possono salvare su nastro blocchi di bytes, di cui basta specificare l'indirizzo iniziale e quello finale, non si deve calcolare la lunghezza del blocco: a questo, ci pensa Super Monitor.

• *Symbol Shift* + "S". Esegue la copia diretta alla stampante della pagina rappresentata. Poichè pensiamo, in futuro, di presentare alcune routines in assembler, o dei set di caratteri speciali, abbiamo fornito la stampa di una funzione di controllo, chiamata check. Super Monitor, alla fine della stampa, esegue la somma dei bytes mostrati e la scrive. Il lettore che dovesse copiare una routine rappresentata in tal modo, potrebbe a sua volta eseguire la stampa, e controllare se il suo check corrisponde a quello dell'originale, individuando così rapidamente eventuali errori.

Remarks

5 - 90 - setta le variabili; *pa* è il tempo di ritardo alla lettura dei

testi: se si desidera un cursore più o meno rapido, si può modificare questo valore;

92 - 130 - muove il cursore nelle quattro direzioni, con gli opportuni stop;

137 - 142 - accettano i comandi speciali;

150 - 155 - scrivono al premere dei tasti; sono inclusi i controlli anti-errore per limitare l'accettazione alle lettere ABCDEF e alle cifre;

160 - 500 - accetta altri comandi speciali;

3000-3500 - accetta, ad ogni input, se si tratta di un numero esadecimale, e ne accetta il valore;

3500-3520 - richiama le routines opportune per l'esecuzione di una conversione di un numero;

4000-5000 - riempie la pagina, convertendo i dati dalla memo-

ria, e mostrandoli sullo schermo, nelle varie zone colorate;

5000-5040 - conversione decimale - esadecimale di numeri fino a 256;

5050-5070 - conversione come sopra, ma per numeri fino a 65536;

6000-6030 - conversione esadecimale-decimale di numeri fino a FF;

6050-6070 - conversione, come sopra, ma per numeri fino a FFFF;

6100-6120 - stampa e rileva il cursore;

7000-7020 - stampa le istruzioni a fondo pagina;

8000-8060 - accetta la riga su cui si trova il cursore nella zona bianca;

8500-8530 - come sopra, ma per la zona verde (caratteri). ■

```

0140 43 49 52 43 4C C5 CIRC?
0153 49 4E CB 50 41 50 IN?PAP
0159 45 D2 46 4C 41 53 E?ELAS
015F CB 42 5E 49 47 48 ?BRIGH
0165 D4 49 4E 56 45 52 ?INVER
016B 53 C5 4F 56 45 D2 S?QUE?
0171 4F 55 D4 4C 50 52 OUTLPR
0177 49 4E D4 4C 4C 49 IN?LLI
017D 53 D4 53 54 4F D0 S?STD?
0183 52 45 41 C4 44 41 REA?DA
0189 54 C1 52 46 53 54 T?REST
018F 4F 52 C5 4E 45 D7 OR?NE?
0195 42 4F 52 44 45 D2 BORDE?
019B 43 4F 4E 54 48 4E CONTIN
01A1 55 C5 44 49 CD 52 U?DI?R
01A7 45 CD 46 4F D2 47 E?FO?G
01AD 4F 20 54 CF 47 4F O T?GO
01B3 20 53 55 C2 49 4E SU?IN
01B9 50 55 D4 4C 4F 41 PU?LOA
01BF C4 4C 49 53 D4 4C ?LIS?L
01C5 45 D4 50 41 55 53 E?PAUS
01CB C5 4E 45 52 D4 52 ?NEXT?
CHECK = 13667

```


"DUMP, O LA BELLA VARIABILINA"

un utilissimo accessorio per i programmatori esigenti

di Gianluca Carri

DUMP è un programma di utilità in linguaggio macchina che consente di visualizzare sullo schermo i nomi di tutte le variabili in memoria con il relativo valore numerico o alfanumerico. Disporre di questo accessorio si rivelerà molto utile in fase di stesura e correzione di programmi complessi: eviterete di incorrere in errore riutilizzando una variabile già definita.

Il programma segnala i seguenti tipi di variabili:

a/ numeriche, con nome lungo a piacere: fornisce nome e valore in floating point;

b/ alfanumeriche: fornisce nome e stringa;

c/ di ciclo FOR/NEXT: fornisce il nome contrassegnato da un asterisco e il valore corrente;

d/ array numerici e alfanumerici: fornisce i valori delle dimensioni; per limitare l'uscita video, non vengono listati tutti i singoli elementi.

Il programma, adatto anche allo Spectrum 16K, inizia all'indirizzo 32000 (7D00H). Potendo essere richiamato anche da pro-

gramma, può stampare l'andamento di una serie di variabili in tutti i punti critici; quindi un aiuto nel debugging molto più com-

pleto e veloce dell'uso di noiose PRINT. Per verificare il corretto caricamento, provate a far girare questo ciclo:

(list n° 1)

```
5 LET a$=""
10 FOR a=1 TO 3
20 LET n=a*PI
30 LET a$=a$+STR$ INT n
40 RANDOMIZE USR 32000: PAUSE 0
50 NEXT a
```

dovreste ottenere quanto segue (PAUSE 0 ferma l'esecuzione ad ogni passaggio, finchè non si preme un tasto):

(list n° 2)

```
a*:1
n:3.1415927
a$:3
```

```
a*:2
n:6.2831853
a$:36
```

```
a*:3
n:9.424778
a$:369
```


La variabile *a* è seguita da un asterisco, ad indicare che si tratta del controllo di un ciclo FOR/-

NEXT. In certi casi si possono avere risultati ambigui; per esempio con

(list n° 3)

```
10 LET a$=""
20 LET b$=""
30 LET c$=""
40 RANDOMIZE USR 32000
```

pur avendo tre stringhe diverse l'uscita di DUMP sarà questa:

(list n° 4)

```
a$:
b$:
c$:
```

L'inconveniente si supera con l'istruzione di INVERSE:

(list n° 5)

```
10 LET a$=""
20 LET b$=""
30 LET c$=""
40 INVERSE 1: RANDOMIZE USR 32
000: INVERSE 0
```

```
a$:
b$:
c$:
```

Note al listato

100 - Scrive sullo schermo TV.

110-120 - HL è il puntatore all'interno dell'area variabili.

130-180 - Controlla se l'area variabili è finita, e se sì, ritorna al Basic.

200-410 - Decodifica il tipo di variabile e chiama la routine corrispondente.

440-590 - La variabile è una stringa: visualizza nome e contenuto.

620-660 - Variabile numerica con nome formato da una sola lettera.

690-830 - Variabile numerica con nome formato da più di una lettera.

870-1240 - Visualizza il valore delle dimensioni dell'array puntato da HL.

1270-1350 - Variabile di controllo FOR.. NEXT.

1440-1490 - Visualizza il nome della variabile.

1520-1690 - Visualizza il valore assunto dalla variabile numerica.

Di seguito è listato il programma Basic necessario per caricare alle giuste locazioni di memoria i codici del programma in linguaggio macchina. Il programma, una volta immesso, va fatto girare con RUN. Al messaggio "start tape, then press any key" si avvierà il registratore in posizione REC per salvare il programma DUMP. Se in un secondo tempo lo vorrete ricaricare, impostate:

CLEAR 31999: LOAD ""CODE

Una volta salvato su cassetta

si avrà un NEW che cancella il programma Basic di caricamento ma che lascia intatto DUMP allocato sopra a RAMTOP. Provate con:

```
LET abcd=-678: RAN DOMIZE
USR 32000
```

Se non avete sbagliato a battere il programma, dovrete avere sullo schermo:

abcd: -678

Nota. Se desiderate rilocare il programma a indirizzi più elevati (nel caso possediate uno Spectrum 48K), in modo da permettere l'uso di DUMP con programmi molto lunghi, cambiate tutti i numeri "125" che compaiono nelle linee DATA del programma

Basic in altrettanti "253".

Per fare eseguire il programma sarà necessario RAN DOMIZE USR 64768. Tutti i CLEAR prima menzionati saranno "CLEAR 64767"; la linea 20 del listato Basic diverrà: FOR a = 64768 TO 64994 mentre la linea 50 sarà: SAVE "DUMP" CODE 64768,226.

```
1 DATA 253,2003,0002,134,042,07
5,092,043,035,062,013,215,126,25
4,128,200,203,127,032,014,203,11
1,032,005,205,066,125,024,235,20
5,087,125,024,230,203,119,032,01
4,203,111,032,005,205,121,125,02
4,217,205,097,125,024,212,203,11
1,032,005,205,188,125,024,203,20
5,171,125,024,198,205,196,125,06
2,036,215,062,058,215,078,035,07
0,120,177,200,035,011,126,215,02
4,247,205,196,125,062,058,215,20
5,204,125,201,205,196,125,043
2 DATA 035,126,203,127,032,00
3,215,024,247,203,191,215,062,05
0,215,035,205,204,125,201,205,19
0,125,062,040,215,094,035,066,03
5,213,126,245,070,035,197,078,03
5,070,200,205,027,020,225,002,04
4,215,193,016,240,062,000,215,06
2,041,215,241,200,135,060,235,07
0,000,000,167,237,060,235,025,20
1,205,196,125,062,042,215,062,05
0,215,205,204,125,017,013,000,02
5,201,205,196,125,062,036,215,02
4,184,126,230,031,198,096,215,03
5,201,126,035,213,197,094,035,08
6,035,078,035,070,229,205,182,04
2,205,050,032,225,193,209,201
```

```
10 CLEAR 31999: RESTORE
20 FOR a=32000 TO 32225
30 READ b: POKE a,b
40 NEXT a
50 SAVE "DUMP" CODE 32000,226
```

Il programma DUMP in codici decimali e la routine che lo inserisce oltre il RAMTOP.

```
d(7)
e(7,2)
f#(7,19)
i#:6.5449847
j#:0
x:19
h:1.8833333
a:3.1673607
b:.00022711
c:2.86234E-8
d:36525
t:0.84015273
p:6.862065
ts:3.6816655
e:0.40951058
q1:10
as:213.16736
c#:3:10
a#:Ge
```

Un lancio di DUMP con un programma ricco di variabili ne evidenzia l'utilità.

DUMP disassemblato

0010 : **** "DUMP"	7D09 3E0D	0140	LD A,0DH
0020 :	7D0B D7	0150	RST 10H
0030 : "lc) 1983"	7D0C 7E	0160	LD A,(HL)
0040 : "Gianluca"	7D0D FE80	0170	CP 80H
0050 : "Carri"	7D0F C8	0180	RET Z
0060 :		0190 :	
0070 :		0200	BIT 7,A
7D08 5C4B	ORG 32000	0210	JR NZ,NO-7
	EQU 23627	0220	BIT 5,A
0090		0230	JR NZ,NO-5
7D00 FDCB0286	RES 0,(IY+2)	0240	CALL STRNG
7D04 2A4B5C	LD HL,(IY+2)	0250	JR LOOP
7D07 2B	DEC HL	0260 NO-5	CALL NUM1
7D08 23	LOOP INC HL	0270	JR LOOP

7D22	CB77	0280	NO-7	BIT	5.A	7D8A	23	1000	INC	HL
7D24	200E	0290		JR	NZ.NO-6	7D8B	46	1010	LD	B.(HL)
7D26	CB6F	0300		BIT	5.A	7D8C	E5	1020	PUSH	HL
7D28	2005	0310		JR	NZ.NARR	7D8D	CD1B1A	1030	CALL	1A1BH
7D2A	CD797D	0320		CALL	ARRAY	7D90	E1	1040	POP	HL
7D2D	18D9	0330		JR	LOOP	7D91	3E2C	1050	LD	A."."
7D2F	CD617D	0340	NARR	CALL	NUM2	7D93	D7	1060	RST	10H
7D32	18D4	0350		JR	LOOP	7D94	C1	1070	POP	BC
7D34	CB6F	0360	NO-6	BIT	5.A	7D95	10F0	1080	DJNZ	NEXTD
7D36	2005	0370		JR	NZ.NARRC	7D97	3E08	1090	LD	A.8
7D38	CD6C7D	0380		CALL	ARRCH	7D99	D7	1100	RST	10H
7D3B	18CB	0390		JR	LOOP	7D9A	3E29	1110	LD	A."."
7D3D	CDAB7D	0400	NARRC	CALL	FNEXT	7D9C	D7	1120	RST	10H
7D40	18CB	0410		JR	LOOP	7D9D	F1	1130	POP	AF
		0420	:			7D9E	D1	1140	POP	DE
		0430	:			7D9F	87	1150	ADD	A
7D42	CDC47D	0440	STRNG	CALL	NAME	7DA0	3C	1160	INC	A
7D45	3E24	0450		LD	A."\$"	7DA1	EB	1170	EX	DE.HL
7D47	D7	0460		RST	10H	7DA2	4F	1180	LD	C.A
7D48	3E3A	0470		LD	A."."	7DA3	0500	1190	LD	B.0
7D4A	D7	0480		RST	10H	7DA5	A7	1200	AND	A
7D4B	4E	0490		LD	C.(HL)	7DA6	ED42	1210	SBC	HL.BC
7D4C	23	0500		INC	HL	7DA8	EB	1220	EX	DE.HL
7D4D	46	0510		LD	B.(HL)	7DA9	19	1230	ADD	HL.DE
7D4E	78	0520	TEXT	LD	A.B	7DAA	C9	1240	RET	
7D4F	B1	0530		OR	C			1250	:	
7D50	C8	0540		RET	Z			1260	:	
7D51	23	0550		INC	HL	7DAB	CDC47D	1270	FNEXT	CALL NAME
7D52	0B	0560		DEC	BC	7DAE	3E2A	1280	LD	A."\$"
7D53	7E	0570		LD	A.(HL)	7DB0	D7	1290	RST	10H
7D54	D7	0580		RST	10H	7DB1	3E3A	1300	LD	A."."
7D55	18F7	0590		JR	TEXT	7DB3	D7	1310	RST	10H
		0600	:			7DB4	CDC47D	1320	CALL	VALUE
		0610	:			7DB7	110D00	1330	LD	DE.13
7D57	CDC47D	0620	NUM1	CALL	NAME	7DBA	19	1340	ADD	HL.DE
7D5A	3E3A	0630		LD	A."."	7DBB	C9	1350	RET	
7D5C	D7	0640		RST	10H			1360	:	
7D5D	CDC47D	0650		CALL	VALUE			1370	:	
7D60	C9	0660		RET		7DBC	CDC47D	1380	ARRCH	CALL NAME
		0670	:			7DBF	3E24	1390	LD	A."\$"
		0680	:			7DC1	D7	1400	RST	10H
7D61	CDC47D	0690	NUM2	CALL	NAME	7DC2	18B8	1410	JR	DIM
7D64	2B	0700		DEC	HL			1420	:	
7D65	23	0710	OCHAR	INC	HL			1430	:	
7D66	7E	0720		LD	A.(HL)	7DC4	7E	1440	NAME	LD A.(HL)
7D67	CB7F	0730		BIT	7.A	7DC5	E61F	1450	AND	1FH
7D69	2003	0740		JR	NZ.LCHAR	7DC7	C660	1460	ADD	60H
7D6B	D7	0750		RST	10H	7DC9	D7	1470	RST	10H
7D6C	18F7	0760		JR	OCHAR	7DCA	23	1480	INC	HL
7D6E	C8BF	0770	LCHAR	RES	7.A	7DCB	C9	1490	RET	
7D70	D7	0780		RST	10H			1500	:	
7D71	3E3A	0790		LD	A."."			1510	:	
7D73	D7	0800		RST	10H	7DCC	7E	1520	VALUE	LD A.(HL)
7D74	23	0810		INC	HL	7DCD	23	1530	INC	HL
7D75	CDC47D	0820		CALL	VALUE	7DCE	D5	1540	PUSH	DE
7D78	C9	0830		RET		7DCF	C5	1550	PUSH	BC
		0840	:			7DD0	5E	1560	LD	E.(HL)
		0850	:			7DD1	23	1570	INC	HL
7D79	CDC47D	0860	ARRAY	CALL	NAME	7DD2	56	1580	LD	D.(HL)
7D7C	3E28	0870	DIM	LD	A."("	7DD3	23	1590	INC	HL
7D7E	D7	0880		RST	10H	7DD4	4E	1600	LD	C.(HL)
7D7F	5E	0890		LD	E.(HL)	7DD5	23	1610	INC	HL
7D80	23	0900		INC	HL	7DD6	46	1620	LD	B.(HL)
7D81	56	0910		LD	D.(HL)	7DD7	E5	1630	PUSH	HL
7D82	23	0920		INC	HL	7DD8	CDB62A	1640	CALL	2AB6H
7D83	D5	0930		PUSH	DE	7DD9	CD3220	1650	CALL	2032H
7D84	7E	0940		LD	A.(HL)	7DDE	E1	1660	POP	HL
7D85	F5	0950		PUSH	AF	7DDF	C1	1670	POP	BC
7D86	46	0960		LD	B.(HL)	7DE0	D1	1680	POP	DE
7D87	23	0970	NEXTD	INC	HL	7DE1	C9	1690	RET	
7D88	C5	0980		PUSH	BC			1700		
7D89	4E	0990		LD	C.(HL)			1710	END	

ESCLUSIVO

a cura di Bianca Orlando

LA ROM dello ZX81

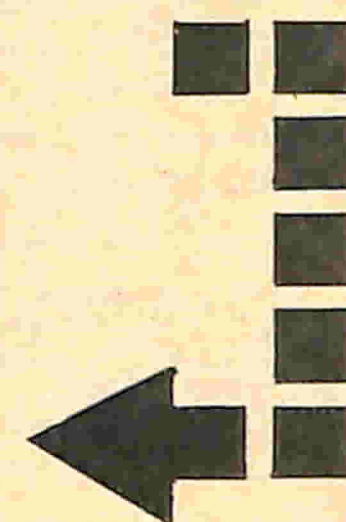
(Seconda parte)

FAST 1

Il generatore di NMI (interrupt non mascherabile) viene escluso con un'istruzione di OUT. Quindi si resetta il bit 7 di CDFLAG.

```
02E7 FD CB 3B 7E BIT 7, (HL)
02EB CB          RET Z
02ED 7B          HALT
02ED D3 FD      OUT (FD), A
02EF FD CB 3B 5E RES 7, (HL)
02F3 CB          RET
```

Subroutine FAST
resetta il bit 7 di CDFLAG
se è attivato lo SLOW
Disattiva il gener. NMI
Resetta il bit 7.

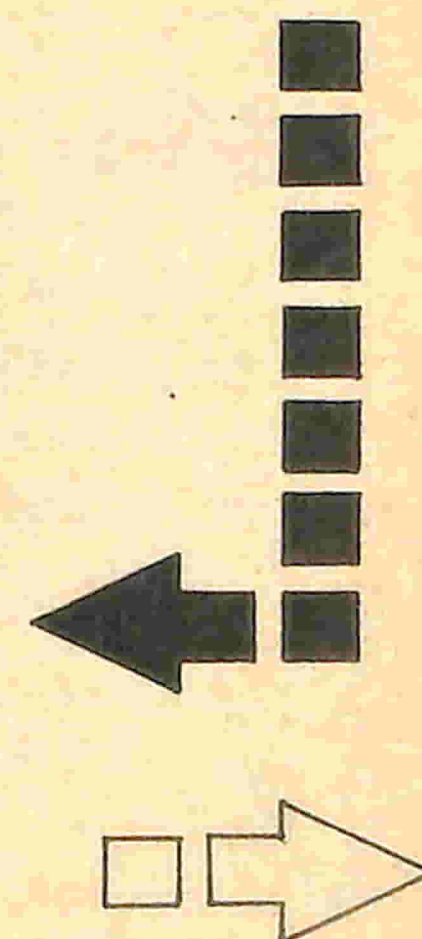


SAVE

Si inizia chiamando la subroutine NAME, per avere dalla coppia di registri HL, l'inizio del nome del programma. Dopo una pausa di 5 secondi, invia i dati del nome del programma e quindi i dati del programma vero e proprio con le variabili, mediante istruzioni di OUT, alla sezione di output.

```
02F6 CD AB 03 CALL 03AB
02F8 CB 7D 00 OR C, 02F4
02FA EB 7B EX DE, HL
02FC 01 0B LD DE, 120B
02FE 01 4B CALL 0F4B
0300 00 00 JP NC, 030B
0302 01 00 LD DE, 0304
0304 01 00 LD A, D
0306 00 00 OR E, A
0308 00 00 JP NZ, 02FF
030A CD 1E 03 CALL 031E
030C CB 7E BIT 7, (HL)
```

SAVE di un carattere del nome
Se è l'ultimo carattere,
salta al byte successivo
Memorizza il nome
Memorizza le variabili di sistema
(incrementa di un byte)
(chiama UPDATE)
incrementa al byte successivo
Setta il bit di start
Carica nel registro E
Già ruotato 8 volte?
A=00 oppure FF

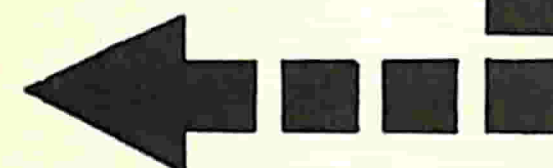



```

0010 20 INC HL
0011 20 JR Z,030B
0012 21 LD HL,4009
0013 CD 40 CALL 031E
0014 CD 01 CALL 01FC
0015 18 JR 0316
0016 5F LD E,(HL)
0017 37 SCF
0018 0B RL E
0019 13 RET Z
0020 08 SBC A,A
0021 05 AND 05
0022 04 ADD A,04
0023 4F LD C,A
0024 D3 OUT (FF),A
0025 06 LD B,23
0026 10 DJNZ 032D
0027 CD 40 CALL 0F40
0028 00 JR NC,03A8
0029 06 LD B,1E
0030 10 DJNZ 0336
0031 0D DEC C
0032 20 JR NZ,0329
0033 A7 AND A
0034 10 DJNZ 0336
0035 18 JR 0320

```

A=00 oppure 05
 A=00 oppure 09
 Carica nel registro C
 All'output EAR
 Valore del timer
 Timer
 Controllo del tasto BREAK
 Se premuto, mess. errore D
 Valore del timer
 Timer
 Azzerà il contatore di bit
 Legge il test di 8 bit
 Timer
 Legge il bit successivo



LOAD

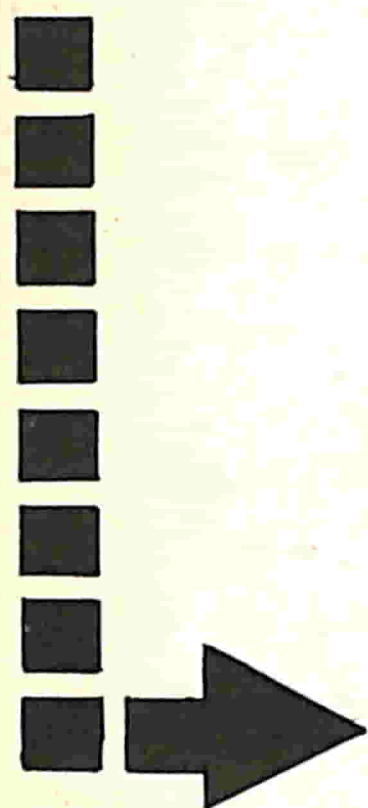
Chiama la subroutine NAME per caricare in HL l'inizio del nome del programma. I dati letti vengono confrontati con il nome del programma richiesto e caricati in RAM in caso di riconoscimento.

```

0340 CD A8 03 CALL 03A8
0343 CB 12 RL D
0345 CB 0A RRC D
0347 CD 40 03 CALL 034C
034A 18 FB JR 0347
034C 0E 01 LD C,01
034E 06 00 LD B,00
0350 3E 7F LD A,7F
0352 0B FE IN A,(FE)
0354 D3 FF OUT (FF),A
0356 1F RRA
0357 30 49 JR NC,03A2
0359 17 RLA
035A 17 RLA
035B 38 28 JR C,0385
035D 10 F1 DJNZ 0350
035F 7F POP AF
0360 BA CP D
0361 D2 E5 03 JP NC,03E5
0364 02 LD H,D
0365 0B LD L,E
0368 CD 40 03 CALL 034C
036B CB 7A BIT 7,D
036D 70 LD A,C
036E 20 03 JR NZ,0371
036F BE CP (HL)
0371 20 D6 JR NZ,0347
0372 17 INC HL
0373 17 RLA

```

LOAD, controllo del nome
 (lettura del nastro)
 Attesa del primo segnale
 Controllo del tasto BREAK
 Impulsi video
 Salto in caso di BREAK
 Pone un byte in C,
 se il flag C è settato
 Nuovo start, se necessario,
 Porta in HL l'indirizzo
 del nome del programma
 Legge un byte del nome
 del programma
 Confronta con il nome richiesto
 Ritorna se non corrisponde
 Carattere success. del nome
 Controlla se è l'ultimo
 Se no, byte successivo
 Incrementa E-LINE di 256
 Indirizzo di start del caricam.
 Carica il primo byte in 4009hex
 (Legge un byte)
 Lo trasferisce in RAM
 (UPDATE)
 Si posiziona sul byte success.
 Lettura dei bit



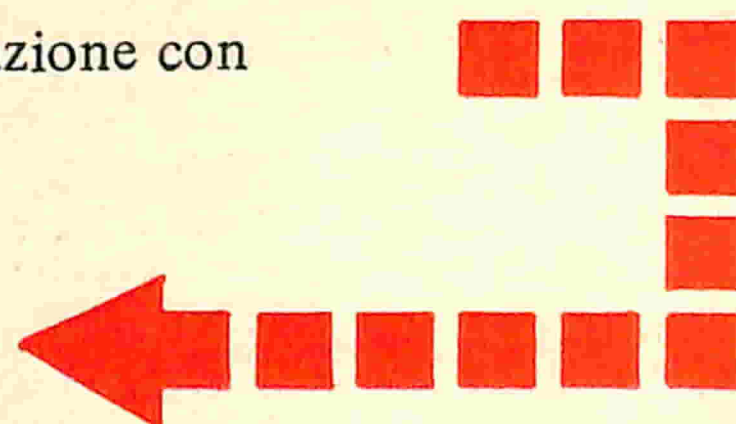
0373	30	F1		JR NC,0365
0375	FD	34	15	INC (IY+15)
0378	21	09	40	LD HL,4009
037B	50			LD D,B
037C	CD	4C	03	CALL 034C
037F	71			LD (HL),C
0380	CD	FC	01	CALL 01FC
0383	18	F8		JR 037B
0385	D5			PUSH DE
0388	1E	94		LD E,94
0388	06	1A		LD B,1A
038A	1D			DEC E
038B	DB	FE		IN A,(FE)
038D	17			RLA
038E	CB	7B		BIT 7,E
0390	7B			LD A,E
0391	38	F5		JR C,0388
0393	10	F5		DJNZ 038A
0395	D1			POP DE
0396	20	04		JR NZ,039C
0398	FE	56		CP 56
039A	30	B2		JR NC,034E
039C	3F			CCF
039D	CB	11		RL C
039F	30	AD		JR NC,034E
03A1	C9			RET
03A2	7A			LD A,D
03A3	A7			AND A
03A4	28	BB		JR Z,0361

Timer
Interroga l'ingresso MIC
Pronto con un bit
Legge il bit success.
se non è ancora pronto
Legge il bit success.
torna a 0369 o 037F
Nuovo start se D=0
altrimenti codice err. D

MESSAGGIO DI ERRORE D

Invia allo schermo il messaggio di errore D, in caso di interruzione con BREAK.

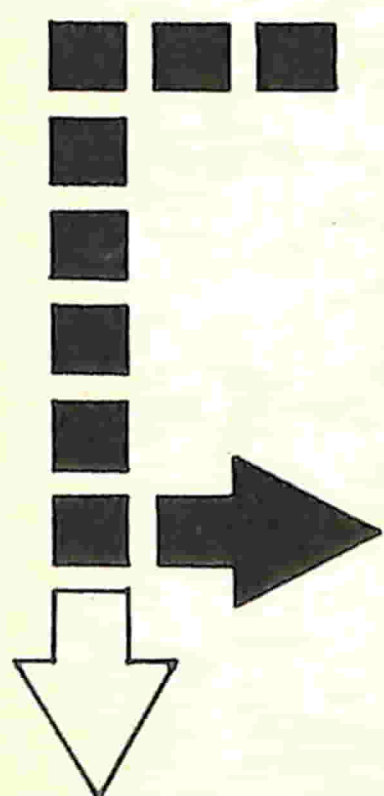
03A6	CF		RST 8
03A7	0C		DB

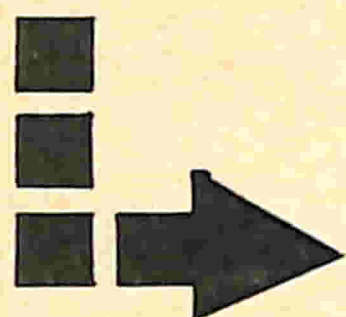


NAME

Questa subroutine viene utilizzata da LOAD e da SAVE. Nella coppia HL si carica il puntatore del nome di programma. Quindi viene invertito l'ultimo carattere e attivato il modo FAST. Attraverso il carattere in campo inverso lo ZX81 riconosce, al momento della rilettura, la fine del nome del programma. Per questo motivo non è consigliabile utilizzare caratteri in campo inverso nell'assegnazione di un nome al programma.

03A8	CD	55	0F	CALL 0F55	Definisce il nome per LOAD/SAVE
03AB	3A	01	40	LD A,(4001)	Interroga i flags, event. settati
03AE	87			ADD A,A	con 0F55, e li controlla
03AF	FA	9A	0D	JP M,0D9A	Messaggio di errore C
03B2	E1			POP HL	Attiva il FAST
03B3	D0			PET NC	





```
03B4 E5
03B5 CD E7 02
03B8 CD F8 13
03BB 02
03BC 0B
03BD 0D
03BE 78
03BF 00
03C0 CB FE
03C2 C9
```

```
PUSH HL
CALL 02E7
CALL 13F8
LD H,D
LD L,E
DEC C
RET M
ADD HL,BC
SET 7,(HL)
RET
```

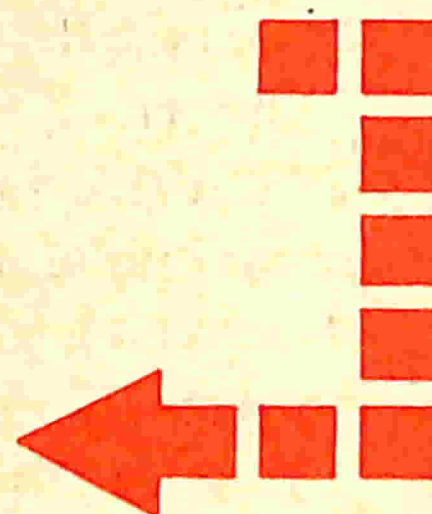
Prende l'indirizzo del nome d. progr., lo carica in HL
In BC la lunghezza d. stringa
Poichè si conta da zero, sottrae uno.

NEW

Come prima cosa attiva il FAST e pone la coppia BC su RAMTOP-1. Poi esegue la subroutine del test di memoria (v.). Se si innalza il RAMTOP, con una POKE seguita da NEW, il test di memoria può controllare la RAM fino a 48K (+16k di ROM = 64k, ovviamente; anche se 8k di ROM non sono usati).

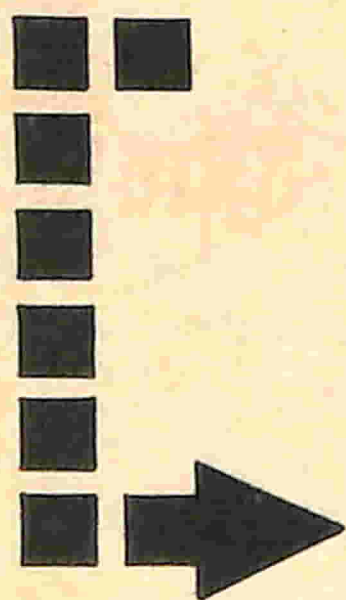
```
03C3 CD E7 02 CALL 02E7
03C6 ED 4B 04 40 LD BC,(4004)
03CA 0B DEC BC
```

NEW, attiva il FAST
RAMTOP in BC e decrementa di uno



TEST DI MEMORIA

Partendo dal valore che è stato posto in BC (cioè il RAMTOP) si scrive all'indietro nella memoria il valore 02 in tutte le locazioni, fino a 4000hex. Quindi si controlla la memoria da 4000hex in avanti, finchè si trovano degli 02 nelle locazioni. Con ciò il sistema verifica quant'è grande la memoria. Il valore finale viene trascritto nuovamente in RAMTOP.



```
03CB 60
03CC 69
03CD 3E 3F
03CF 38 02
03D1 2B
03D2 BC
03D3 20 FA
03D5 A7
03D6 ED 42
03D8 09
03D9 23
03DA 00 05
03DC 35
03DD 20 03
03DF 35
03E0 20 F3
03E2 22 04 40
```

```
LD H,B
LD L,C
LD A,3F
LD (HL),02
DEC HL
CP H
JR NZ,03CF
AND A
SBC HL,BC
ADD HL,BC
INC HL
JR NC,03E2
DEC (HL)
JR Z,03E2
DEC (HL)
JR Z,03D5
LD (4004),HL
```

Test di memoria
In HL l'indirizzo di partenza
Carica 02 in tutta la memoria
proced. all'indietro
Quando arriva a 3FFF
si interrompe
Legge da 4000 in avanti
finchè non trova più 02
Punta in HL sull'indir. success.
Setta il RAMTOP
Decrementa la locaz. di mem.
Setta il RT
Decr. loc. mem.
Locaz. mem. OK
Setta il RT

INIT

Questa sezione di programma viene utilizzata dallo ZX81 quando viene acceso e dopo ogni NEW.

- 1) Scrive 3E nella locazione più alta (che è 7FFF (= 32767) con l'espansione di 16K);
- 2) il registro SP punta la cella di memoria successiva;
- 3) ERR-SP punta la cella di memoria successiva;
- 4) il registro I contiene il valore 1E - questo registro indica l'inizio del generatore di caratteri, il cui indirizzo è appunto 1E00 (7680dec);
- 5) seleziona il modo di interrupt 1;
- 6) pone ERR-NR nella coppia IY;
- 7) prepara il modo SLOW;
- 8) D-FILE indica la fine del testo del programma - se non c'è nessun programma, punta l'indirizzo 16509dec;
- 9) produce uno schermo minimale (solo per la versione 1k): ogni linea consiste di un solo carattere, un NEWLINE;
- 10) si predispone la VARS del sistema; punta la prima locazione di memoria dopo la memoria di schermo;
- 11) chiama CLEAR, e con ciò cancella tutte le variabili;
- 12) pone il cursore nella linea di lavoro;
- 13) infine esegue la subroutine che attiva il modo SLOW.

03E5	2A	04	40	LD HL, (4004)	Pone 3E nella loc. di mem.
03E8	2B			DEC HL	più alta come marcatore
03E9	3E	3E		LD (HL), 3E	Lo stackpointer indica
03EB	2B			DEC HL	la prossima locaz.
03EC	7D			LD SP, HL	Il puntatore ERR-SP
03ED	2B			DEC HL	indica 2 bytes sotto
03EE	2B			DEC HL	I punta il gener. di caratt.
03EF	22	02	40	LD (4002), HL	Inserisce il modo interrupt 1
03F2	3E	1E		LD A, 1E	In IY il valore di ERR-NR
03F4	ED	47		LD I, A	Prepara CDFLAG per lo SLOW
03F6	ED	56		IM 1	D-FILE punta la fine d. progr.
03F8	7D	21	00 40	LD IY, 4000	Produce uno schermo minimale
03FC	7D	36	3E 40	LD (IY+3E), 40	di 25 NEWLINE
0400	21	7D	40	LD HL, 407D	Definisce VARS
0403	22	0C	40	LD (400C), HL	Chiama CLEAR
0406	06	19		LD B, 19	Disloca il cursore
0408	3E	76		LD (HL), 76	Attiva lo SLOW
040A	23			INC HL	
040B	10	FB		DJNZ 040B	
040D	22	10	40	LD (4010), HL	
0410	CD	9A	14	CALL 149A	
0413	CD	AD	14	CALL 14AD	
0416	CD	07	02	CALL 0207	



LIST 1

Mostra sullo schermo una parte del programma. Per esempio, LIST 50 punta il programma dalla linea 50. L'area di lavoro non viene cambiata, nè vi si scrive qualcosa. Questo avviene anche quando il cursore è mosso verso il basso. Il programma è mostrato da S-TOP.



```

0419 CD 2A 0A CALL 0A2A
041C 2A 0A 40 LD HL,(400A)
041F ED 5B 23 40 LD DE,(4023)
0423 A7 AND A
0424 ED 52 SBC HL,DE
0426 EB EX DE,HL
0427 30 04 JR NC,042D
0429 19 ADD HL,DE
042A 22 23 40 LD(4023),HL
042D CD D8 09 CALL 09D8
0430 28 01 JR Z,0433
0432 EB EX DE,HL
0433 CD 3E 07 CALL 073E
0436 FD 35 1E DEC(IY+1E)
0439 20 37 JR NZ,0472
043B 2A 0A 40 LD HL,(400A)
043E CD D8 09 CALL 09D8
0441 2A 16 40 LD HL,(4016)
0444 37 SCF
0445 ED 52 SBC HL,DE
0447 21 23 40 LD HL,4023
044A 30 0B JR NC,0457
044C EB EX DE,HL
044D 7E LD A,(HL)
044E 23 INC HL
044F ED A0 LDI
0451 12 LD(DE),A
0452 18 C5 JR 0419
0454 21 0A 40 LD HL,400A
0457 3E LD E,(HL)
0458 23 INC HL
0459 36 LD D,(HL)
045A EB PUSH HL
045B EB EX DE,HL
045C 23 INC HL
045D CD D8 09 CALL 09D8
0460 CD BB 05 CALL 05BB
0463 E1 POP HL
0464 FD CB 2D 6E BIT 5,(HL)
0468 20 08 JR NZ,0472
046A 72 LD(HL),D
046B 2B DEC HL
046C 73 LD(HL),E
046D 16 AA JR 0419

```

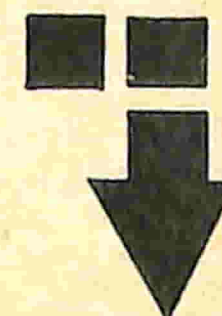
RICHIAMO DELLA LINEA DI LAVORO

Interessa la parte inferiore dello schermo, in cui viene copiata la linea di lavoro. La rappresentazione interna delle cifre, che si trova dopo la rappresentazione alfabetica, non viene copiata, poichè ciò sarebbe di disturbo per la rappresentazione di una linea di programma.

```

046F CD AD 14 CALL 14AD
0472 2A 14 40 LD HL,(4014)

```



Fiera di Milano

14-18 Aprile

1984



**quando il computer
sa fare qualcosa di più**

È bello sapere che ognuno di noi può contare in ogni momento su un amico fidato, tanto serio e preciso sul lavoro, quanto versatile e disponibile fuori dall'ufficio. Capace, tra l'altro, di fotografare, disegnare, farti l'oroscopo o i bioritmi, prescriverti la dieta, scrivere la tua musica, aiutarti nello studio e sempre pronto per una partita a scacchi.

Il computer, oggi, è anche questo e tante altre cose.

14-18 Aprile 1984. Cinque giorni per presentare al grande pubblico tutto quello che di nuovo e particolare si può fare con il computer nel campo del lavoro e dell'hobby.

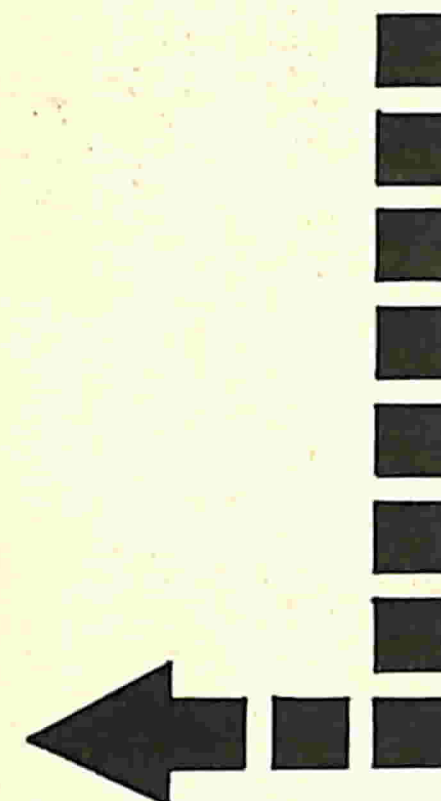
COMPUTER SHOW è un'iniziativa del Salone dell'Informatica
Informazioni e adesioni:

Segreteria: 20139 Milano - Via Marochetti, 27 - tel. (02) 53.98.267 - 56.93.973

```

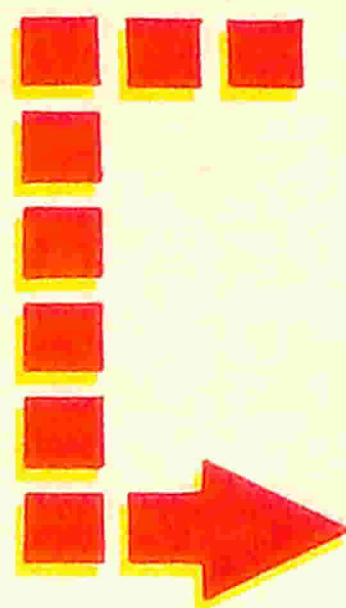
0475 7E      LD A,(HL)
0476 FE 7E   CP 7E
0478 20 08   JR NZ,0482
047A 01 06 00 LD BC,0006
047D CD 60 0A CALL 0A60
0480 18 FC   JR 0475
0482 FE 76   CP 76
0484 23     INC HL
0485 20 EE   JR NZ,0475
0487 CD 37 05 CALL 0537
048A CD 1F 0A CALL 0A1F
048D 2A 14 40 LD HL,(4014)
0490 FD 36 00 FF LD(IY+00),FF
0494 CD 66 07 CALL 0766
0497 FD CB 00 7E BIT 7,(HL)
049B 20 24   JR NZ,04C1
049D 3A 22 40 LD A,(4022)
04A0 FE 18   CP 18
04A2 30 1D   JR NC,04C1
04A4 3C     INC A
04A5 32 22 40 LD(4022),A
04A8 47     LD B,A
04AB 0E 01   LD C,01
04AD CD 18 09 CALL 0918
04AE 54     LD D,H
04AF 5D     LD E,L
04B0 7E     LD A,(HL)
04B1 2B     DEC HL
04B2 BE     CP(HL)
04B3 20 FC   JR NZ,04B1
04B5 23     INC HL
04B6 FE     EX DE,HL
04B7 3A 05 40 LD A,(4005)
04BA FE 4D   CP 4D
04BC D0 5D 0A CALL C,0A5D
04BF 18 C9   JR 048A

```



ATTESA DI UN TASTO

Il puntatore di errore è settato a zero e produce uno schermo, finchè non si preme un tasto.



```

04C1 21 00 00 LD HL,0000
04C4 22 18 40 LD(4018),HL
04C7 21 3B 40 LD HL,403B
04CA CB 7E   BIT 7,(HL)
04CC CC 29 02 CALL Z,0229
04CF CB 40   BIT 0,(HL)
04D1 28 FC   JR Z,04CF
04D3 ED 4B 25 40 LD BC,(4025)
04D7 CD 4B 07 CALL 0F4B
04DA CD BD 07 CALL 07BD
04DD 30 93   JR NC,0472

```




TOP 10

CLASSIFICA DELLE PIU' VENDUTE DEL MESE

Arcade games

Titolo - (Produttore) - [Posizione precedente]

- 1 Flight Simulation (*Psion*) [1]
- 2 3D Tanx (*DK Tronics*) [4]
- 3 Horace goes skiing (*Psion / Melbourne*) [10]
- 4 Manic Miner (*Bug Byte*) [3]
- 5 Chequered Flag (*Psion*) [2]
- 6 Frogger (*Rabbit*)
- 7 Hungry Horace (*Psion / Melbourne*)
- 8 Aquarius (*Bug / Byte*) [9]
- 9 Maziacs (*DK' Tronics*) [6]
- 10 Monster in Hell [8]

Giochi di strategia

Titolo - (Produttore) - [Posizione precedente]

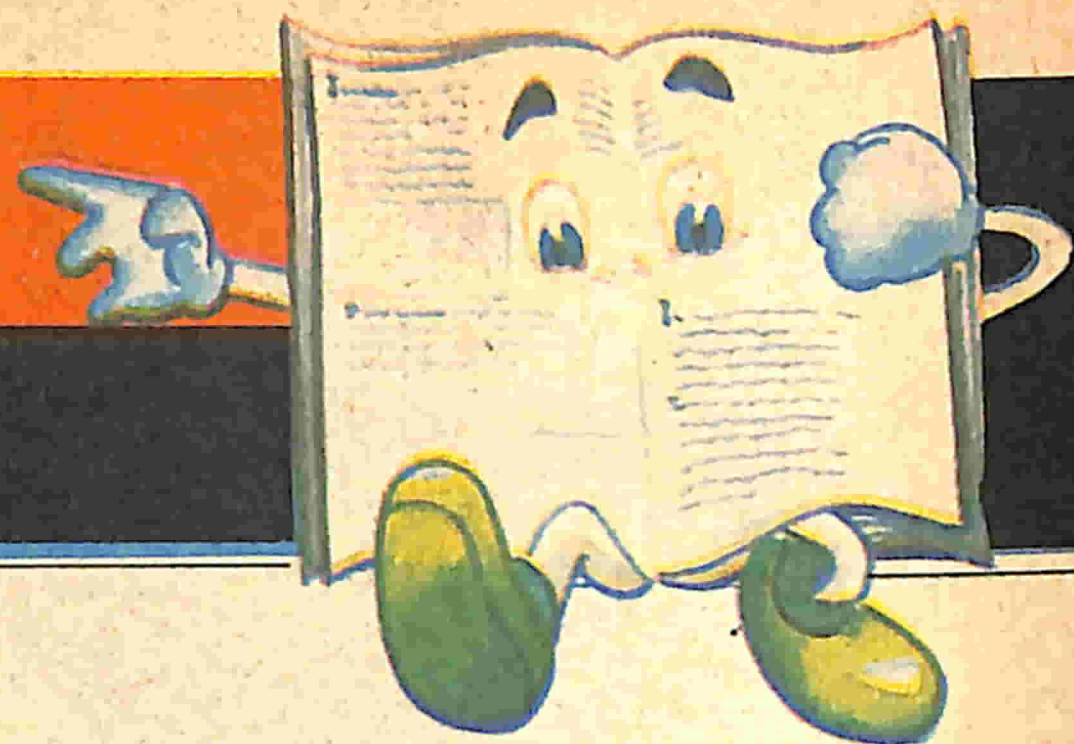
- 1 Chess
(*Psion / Rebit*) [1]
- 2 16K Superchess
(*CP Software*) [3]
- 3 Dictator
(*DK' Tronics*)
- 4 ZX Daught
(*CP Software*) [5]
- 5 Cyrus in Chess
(*Sinclair*)

Utility e home

Titolo - (Produttore) - [Posizione precedente]

- 1 VU - File (*Psion / Rebit*) [1]
- 2 Cass. Dimostrativa Horizont (*Rebit*)
- 3 Piccolo magazzino (*Rebit*)
- 4 Vu - 3D (*Psion / Rebit*)
- 5 Conto corrente (*Rebit*)

Per questa rubrica scrivi a:
Sinclair Computer
c/o Sigma Informatica
V. Cristoforo Colombo, 49
20090 Trezzano
Milano



QL, tanta strada

In un'atmosfera molto mondana, presso la sala congressi dell'Hotel Michelangelo, a un passo dalla Stazione Centrale, si è tenuta il 20 febbraio scorso la conferenza stampa di presentazione del nuovo personal computer della Sinclair.

Dopo i convenevoli di rito, Mr. Charles Cotton, Overseas Business Manager della Sinclair, ha illustrato il nuovo prodotto e la filosofia che lo ha ispirato. Tecnici della PSION hanno dato alcune dimostrazioni pratiche. La Psion è una ditta già nota per aver sviluppato il software di base dello Spectrum, ed è l'artefice

anche del sistema operativo del QL.

Con un rapporto prezzo/prestazioni intenzionato a sconvolgere il mercato, il QL vuole proporsi ad un pubblico vasto e diversificato, dall'hobbysta esigente al professionista. Un microprocessore a 32 bit offre una potenza di calcolo che trova confronti solo su computer di costo molto superiore: i Motorola della famiglia 68000 sono montati, per esempio, dalla Hewlett-Packard 9816 e dal nuovo Apple McIntosh.

Esteticamente, il QL si presenta molto sobrio: il nero-sin-

clair sta diventando uno stile. Un contenitore liscio, rettangolare, un po' spigoloso forse, che ha fatto dire a molti, alla conferenza stampa, "Però è bello!"; tasti intarsiati in bianco, nessuna fascia o scritta in colore.

La tastiera, standard QWERTY, ha avuto un coro di "finalmente!", e in effetti sembra, finalmente, all'altezza della situazione. Alla sua sinistra un connettore (nascosto) per inserire l'espansione RAM da 512K (un 32 bit consente di avere memoria praticamente senza limiti); a destra, in un contenitore che fa corpo unico con la tastiera, due mi-



HARDWARE DIRECTORY

EDIZIONE SPECIALE
COMPUTERS



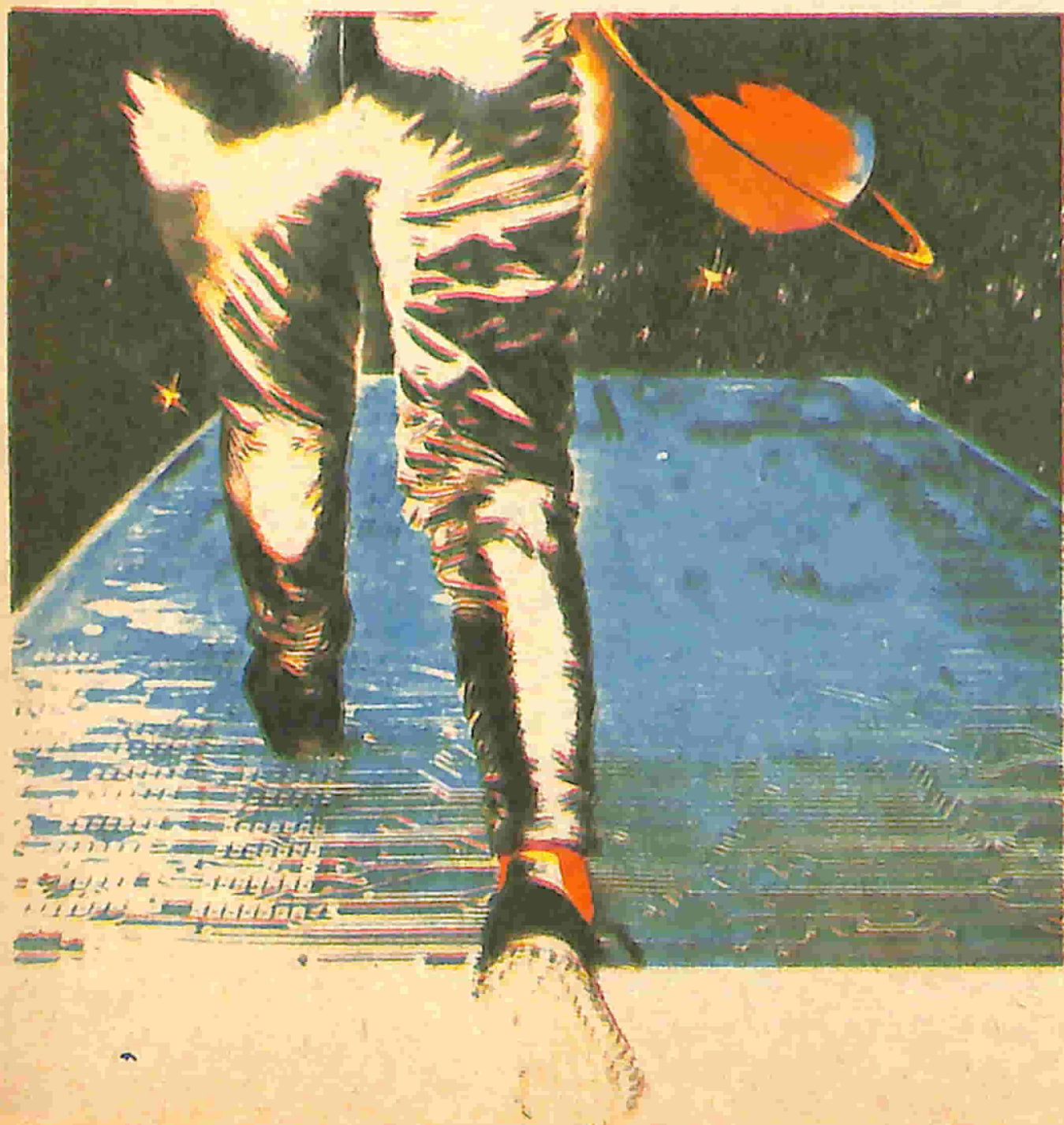
30
Hobby
& Home
Computer

350
mini
e
micro

150
Mainframe

Computer

PROGRAMMARE
PRESTO E BENE



In omaggio
il volume

crodrive, uguali nella meccanica a quelli dello Spectrum (ma non compatibili come sistema operativo); sul retro, ingressi e uscite per tutto quello che potete desiderare, dal joystick alla banca dati.

A proposito dei microdrive, abbiamo chiesto a Mr. Cotton se non ritiene che ci sia una certa sproporzione tra la RAM di oltre mezzo Mbyte e l'unità di memoria di massa di "soli" 100K; con uno slicing molto britannico sulla domanda, Mr. Cotton ha rivelato che sarà disponibile l'interfaccia per disco rigido Winchester (sì, ma il prezzo?). La politica di Clive Sinclair finora è stata vincente, ma a tanti sinclairisti piacerebbe tanto un biicamente tradizionale floppy. Ai posteri.

Il QL verrà consegnato all'utente con 4 programmi applicativi che gli danno un tono molto professionale: un word processor, di cui si dicono meraviglie, un o "spreadsheet", tabellone



elettronico per calcoli (tipo Visi-calc), un archivio e un programma per grafica commerciale (istogrammi, curve, torte). Visto che ci sono le prese per i joystick, regalare anche un giochino non sarebbe stata una cattiva idea... Questo pacchetto di software

verrà "nazionalizzato" in diverse lingue (francese, italiano, tedesco), ed anche la tastiera verrà offerta in più versioni.

Stando alle dichiarazioni dell'importatore, la distribuzione regolare in Italia dovrebbe iniziare a fine autunno.

Sinclair QL - Scheda -

Microprocessore: Motorola 68000 (32 bit, CPU), Intel 8049 (tastiera, effetti sonori, controllo RS232C), due chip di controllo di progetto originale Sinclair.

Sistema operativo: QDOS, originale Sinclair, monoutente, multitasking (possibilità di lavorare simultaneamente con più programmi).

Memoria: ROM di base non dichiarata; ROM estendibile con cartucce di 32K; RAM fissa di 128K; espansione RAM di 512K; memoria video non dichiarata.

L i n g u a g g i: SuperBASIC Sinclair versione potenziata dal basic dello Spectrum; in corso di sviluppo: compilatore Pascal, Assembler 68000.

Tastiera: QWERTY a 65 tasti; 5 tasti funzione; previste tastiere nazionali.

Display: uscita UHF per tv color o b/n, formato 40/60 colonne; uscita RGB monitor alta risoluzione, a colori o monocromatico, formato 85 colonne, pixel 512x256 (4 colori) o 256x256 (8 colori).

I/O: presenti 2 interfacce seriali

RS232C, 2 prese per joystick o comandi cursore (mouse), 2 prese per collegamenti in rete con altri QL o con Spectrum, connettore per cartucce ROM; interfacce esterne in programma: parallela per stampante, IEEE-488, I/O per disco rigido Winchester.

Memoria di massa: due microdrive incorporati, cartuccia a nastro di circa 100K (non formattati).

A l i m e n t a z i o n e: trasformatore di tensione esterno.

Prezzo indicativo: Lit. 1.300.000 in versione minima.

**Se vuoi
abbonarti**

Registrate il mio abbonamento annuale a Sinclair Computer.

☐ Ho versato oggi stesso il canone di Lire 20.000 a mezzo c/c postale n° 30426209 intestato a:
Sinclair Computer - V.le Famagosta, 75 - 20142 Milano

☐ Accludo assegno per lire 20.000 banca n° a favore di
Sinclair Computer

Il mio computer è: uno ZX81 ☐ uno Spectrum ☐ altro (specificare)

Ho ☐ / non ho ☐ la stampante, ma voglio ☐ comprarla.

Preferisco programmi di gioco ☐ , didattici ☐ , d'utilità ☐ , altro

Nome

Via n° CAP. Città

Tel. Orario

**Se vuoi
collaborare**

Registrate fra i collaboratori regolari di Sinclair Computer.

A titolo di prova vi invio un articolo e la cassetta col programma "....."
....." di cui vi garantisco l'assoluta originalità autorizzandovene la pubblicazione.

Per il compenso ☐ telefonatemi al ☐ scrivete mi all'indirizzo sottoindicato.

Nome

Via n° CAP. Città

Tel. Orario

**Se vuoi
un consiglio
o consigliarci**

HELP

Nome

Via n° CAP. Città

Tel. Orario

**Se vuoi
votare
il software**

Questo mese ho comprato i seguenti programmi e li ho giudicati così:

Titolo	Giudizio d'utente (mettere una x nella casella voluta)			
	Ottimo	Buono	Mediocre	Deludente

Nome

Via n° CAP. Città

Tel. Orario

Facoltativo

Da inviare in busta chiusa a:

**Sinclair
Computer**
v.le Famagosta, 75
20142 Milano

***Si, voglio
abbonarmi***

Da inviare in busta chiusa a:

Sinclair Computer
c/o Sigma Informatica
v. Cristoforo Colombo, 49
20090 Trezzano S.N. / Milano

***Si, voglio
collaborare***

Da inviare in busta chiusa a:

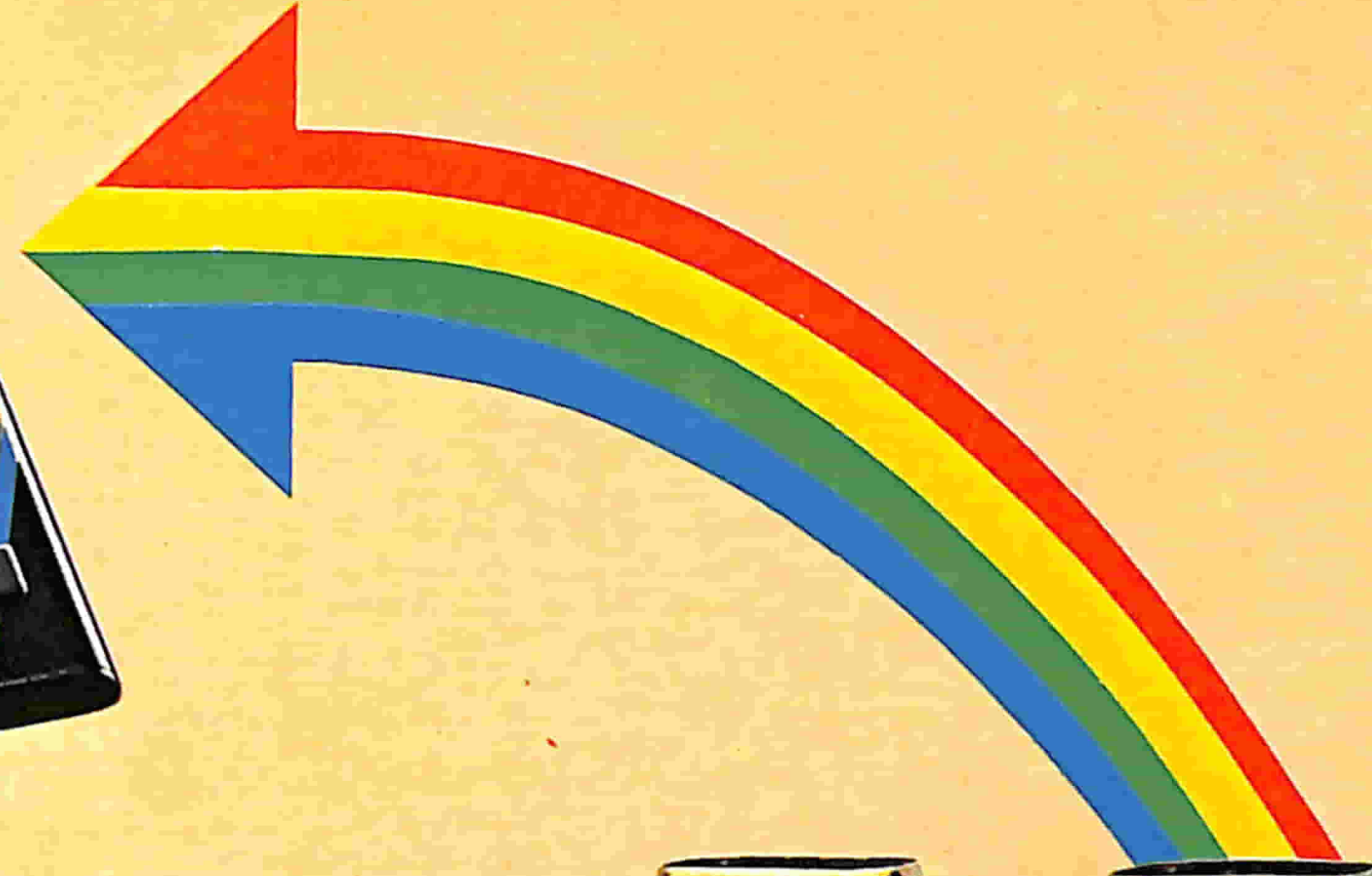
Sinclair Computer
c/o Sigma Informatica
v. Cristoforo Colombo, 49
20090 Trezzano S.N. / Milano

***Si, chiedo
consiglio***

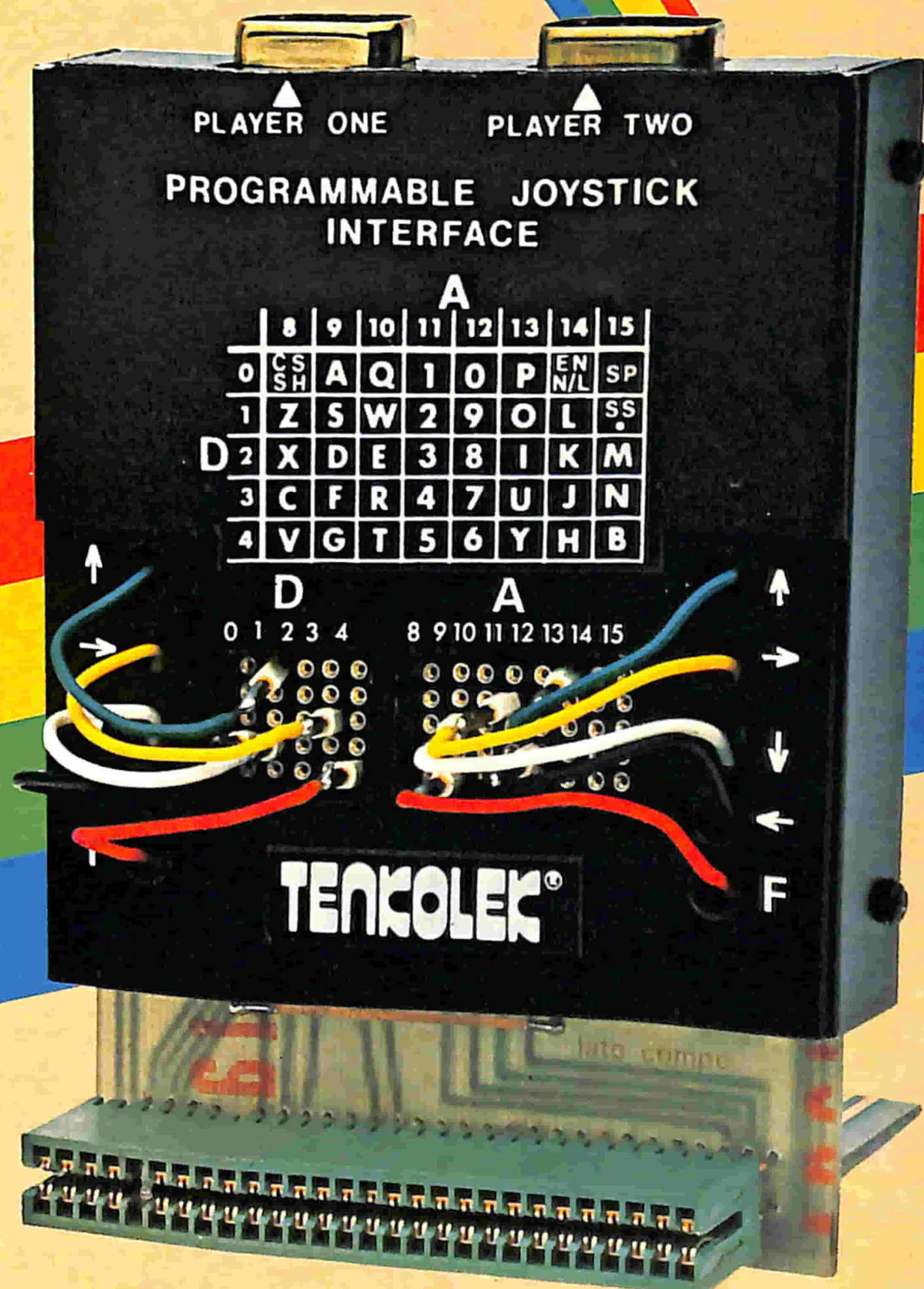
Da inviare in busta chiusa a:

Sinclair Computer
c/o Sigma Informatica
v. Cristoforo Colombo, 49
20090 Trezzano S.N. / Milano

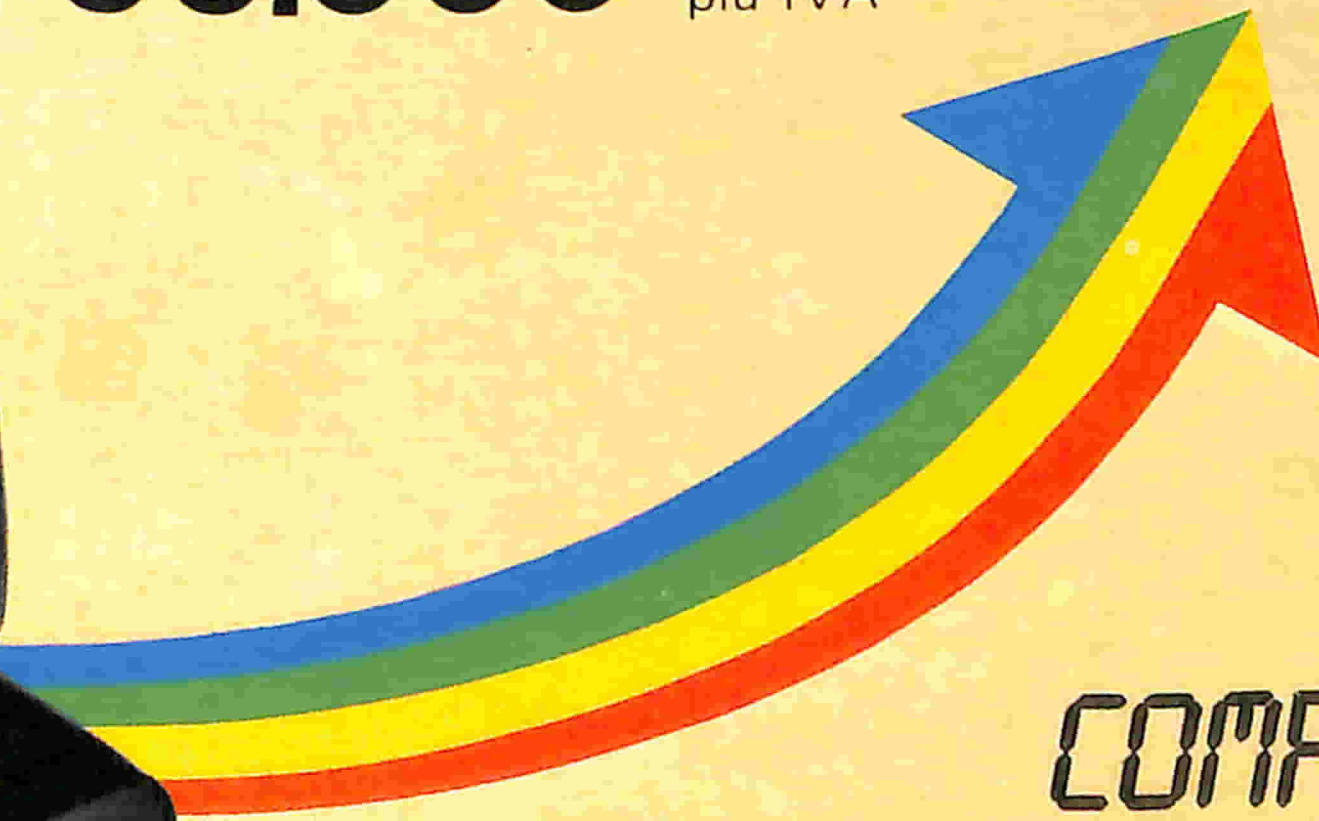
***Si, voglio
votare***



**PROGRAMMABLE
JOYSTICK
INTERFACE
ZX Spectrum**



L. 99.000 più IVA



**ADD ACTION
TO YOUR
COMPUTER GAMES!!**

TENKOLEK®

DISTRIBUITO DA





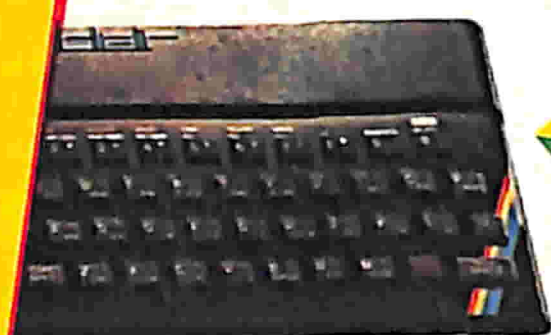
ce l'hai la supergaranzia?

La Rebit Computer, distributrice per l'Italia dei prodotti SINCLAIR, ha messo a punto una nuova **supergaranzia** che ti darà i seguenti vantaggi:

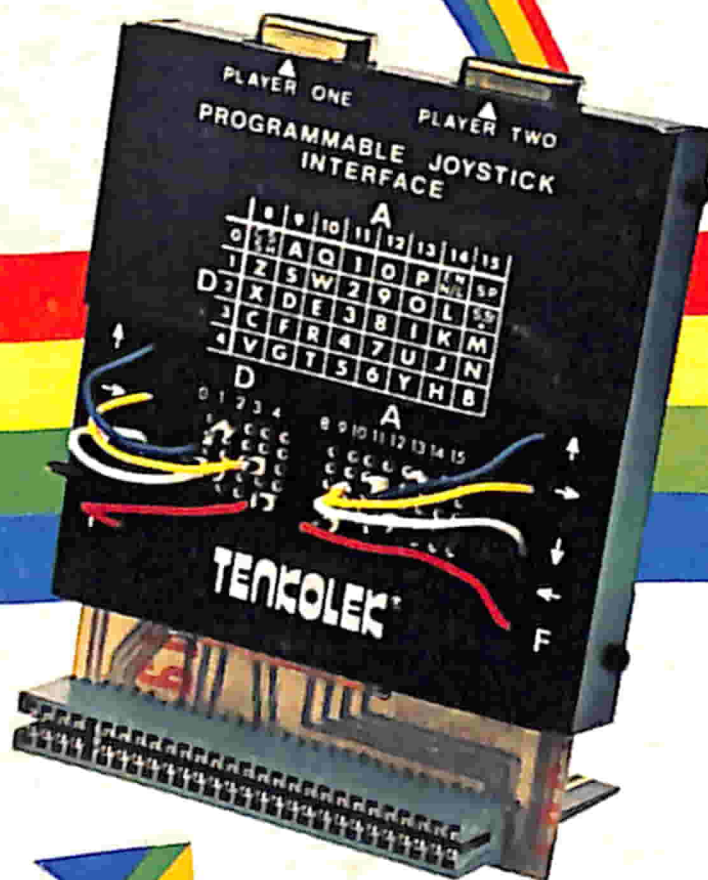
- 1° Prezzo ridotto nell'acquisto dell'interfaccia programmabile.
- 2° Tessera sconto sull'acquisto dei programmi.
- 3° Tariffa ridotta per l'abbonamento a "Sperimentare con il Computer"
- 4° Libro sulle interfacce e sui microdrives.

Un risparmio di oltre 70.000 lire.

NON PERDERE QUESTA OCCASIONE
al prezzo ECCEZIONALE
di **L. 49.000 + IVA**
anzichè
L. 99.000 + IVA



**PROGRAMMABLE
JOYSTICK
INTERFACE
ZX Spectrum**



**ADD ACTION
TO YOUR
COMPUTER GAMES !!
TENKOLEK®**

**Inoltre riceverete
in OMAGGIO
direttamente a casa,
l'opuscolo in italiano :
SINCLAIR ZX Interfaccia 1
ZX Microdrive
del valore di L. 10.000**

