VIC da zero +64

di Tommaso Pantuso

Ritorna questo mese, come promesso, Vic da zero + 64.

La forma sotto cui proponiamo queste pagine è comunque un po' diversa da quella consueta. Visto infatti l'interesse suscitato nei lettori dagli argomenti trattati nel corso degli oltre due anni di vita di Vic da zero + 64, e viste le richieste di partecipazione dei lettori, abbiamo deciso di aprire la rubrica anche agli interventi esterni che rientrano, per così dire, nel suo spirito e stile. Se volete quindi intervenire con i vostri articoli, mettetevi in contatto con noi.

Oggi vi proponiamo, per cominciare, un articolo di Alessandro Guida e Gianni Itta, su un argomento che riteniamo di largo interesse, riguardante le problematiche legate al modo in cui velocizzare le operazioni con il registratore a cassette del 64 e capirne a fondo il funzionamento.

Per ragioni di spazio siamo stati costretti a dividere in due parti l'argomento, che sarà concluso nel prossimo numero.

Flashtape per C64 *di Alessandro Guida e Gianni Itta*

Prima parte

Indubbiamente il registratore è la periferica più diffusa del Commodore 64, grazie anche alla sua semplicità d'uso e affidabilità. Purtroppo a queste doti fa riscontro una lentezza a volte davvero insopportabile. Basti pensare che un programma lungo 26K (105 blocchi di un disco) richiede ben 9 minuti per essere registrato o letto da nastro.

Poiché è comune incontrare programmi (specie i giochi) lunghi anche il doppio, ecco che i 20 minuti necessari diventano un tempo inaccettabile.

Con il programma che presentiamo ci è possibile ridurre di circa nove volte tali tempi, ma soprattutto ci è fornita l'occasione per capire il funzionamento della periferica-registratore e il perché di tanta lentezza.

Per necessità di cose supporremo che il lettore abbia una, sia pur minima, conoscenza del linguaggio macchina nel C64.

Il flashtape

Il nome (opera di Gianni) dato a questo programma serve a sottolineare le caratteristiche «velocistiche» impresse al nastro. Abbiamo comparato il nostro velocizzatore con quelli comunemente in commercio, e ne abbiamo ricavato la tabella 1. Ma la velocità non sarà l'unico pregio. Ecco le altre caratteristiche che si vogliono ottenere:

1) Il programma registrato in formato «flashtape» deve essere caricabile direttamente dal nastro senza richiedere la presenza nella memoria del computer di particolari routine lette in precedenza.

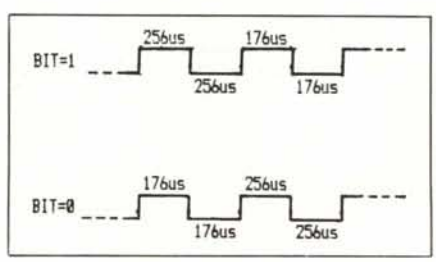


Figura 1a - Codifica dei valori 1 e 0 su nastro da parte del S.O.

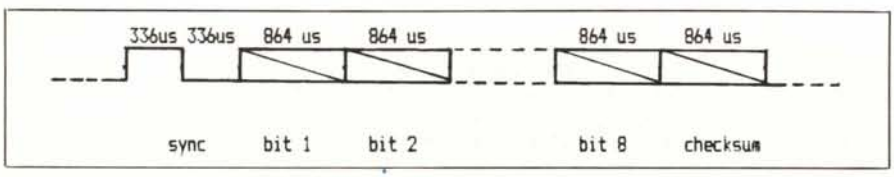


Figura 1b - Codifica di un byte su nastro.

2) Deve essere possibile registrare i programmi con possibilità di autostart a termine load (sia con RUN che con SYS). In altre parole, quando si rileggerà il programma dal nastro non sarà necessario dare il comando RUN o SYS perché il programma andrà in esecuzione da solo.

3) Deve essere possibile registrare anche la parte di RAM che va da $A000 a $CFFF, permettendo così il salvataggio dei programmi che occupano tutta la RAM (i famosi 202 blocchi su disco!).

Modo	Tempo (min.)
Flashtape	1.06
Disk Drive	1.10
Turbo Supervic	1.20
Connection	1.25
Turbo Tape	2.00
Load normale	8.45

Tabella 1 - Tempi di caricamento per un programma test da 26K (105 blocchi disco).

La registrazione su nastro

Il registratore Commodore è per molti versi simile ad un normalissimo registratore audio. Vi è una parte meccanica che fa scorrere il nastro ed una testina che legge o scrive i segnali su di una cassetta.

Pochi componenti elettronici si occupano di adeguare i livelli in ingresso/uscita del C64 a quelli della testina: registrare un programma su nastro significherà inviare al nastro tutti i byte un bit dopo l'altro. È, quindi, necessario trovare un modo per distinguere i bit «1» da quelli «0», e per riconoscere l'inizio della sequenza di bit che compongono il programma; potrebbe, infatti, capitare di posizionarsi con il nastro a metà di una registrazione precedente, ed avere una lettura errata.

La Commodore per codificare i dati su nastro utilizza tre tipi di segnali ad onda quadra (la parte positiva e quella negativa del segnale hanno la stessa durata) di frequenza diversa. L'impulso breve ha una durata di circa 176 microsecondi, l'impulso medio dura 256 micros. e quello lungo ben 336 micros. Un bit 1 viene codificato con un segnale medio ed uno corto, il contrario per un bit di valore 0, pertanto, la durata di un singolo bit su nastro, è costante e pari a 864 microsecondi (vedi figura 1a).

Il singolo byte viene codificato in-

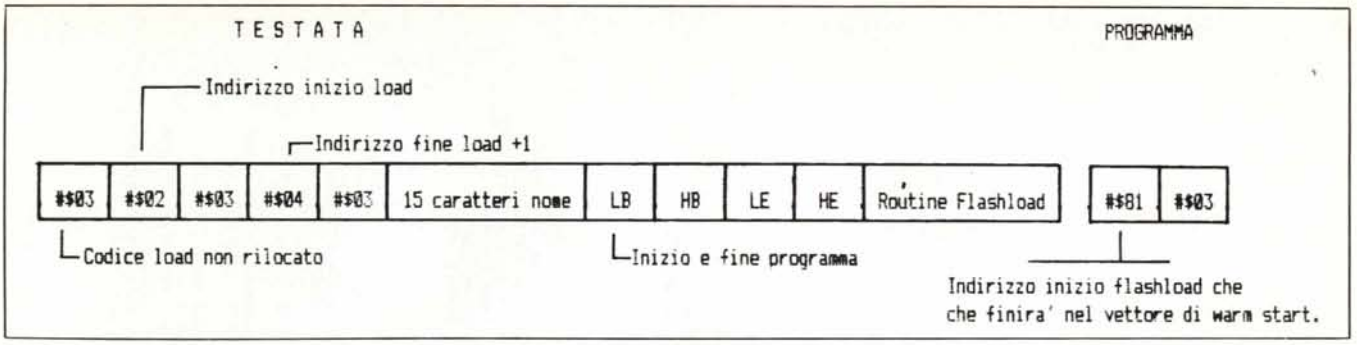


Figura 3 - Formato testata e programma in una registrazione con Flashtape. I due byte-programma sono seguiti dal programma vero in formato veloce.

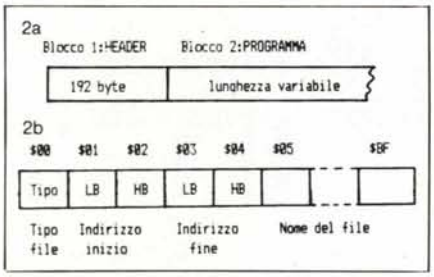


Figura 2a - Formato registrazione programmi.
Figura 2b - Formato header registrazione.

viando un bit di sincronismo, gli 8 bit della parola-dati ed un bit finale di parità per il controllo dell'errore. Il bit di sincronismo è formato da un segnale lungo ed uno medio, mentre il bit di parità è codificato come un bit normale. Lo scopo del bit di sincronismo è quello di separare un byte dall'altro, e di indicare l'inizio dei segnali che determinano lo stato degli 8 bit.

Con semplici calcoli possiamo verificare che un byte richiede 8448 microsecondi, cioè circa 9 millisecondi.

Per registrare 26624 byte (26K) avremo bisogno di:

0.009 × 26624 = 239.6 secondi.

A questi vanno aggiunti 10 secondi di impulsi brevi che vengono preposti ai dati veri e propri. Questo treno di impulsi iniziale svolge un duplice compito: permettere al motore del nastro di raggiungere la giusta velocità e calcolare la durata degli impulsi per compensare eventuali variazioni tra la velocità di scrittura e quella di lettura. Ma pur tenendo conto di questi altri 10 secondi il totale ci dà circa 249 secondi (4.15 minuti), ben lontani dagli 8.45 minuti effettivamente necessari.

Questa discrepanza si spiega con il fatto che i dati vengono registrati due volte per fare in modo che dati errati sfuggiti al controllo del bit di parità vengano rintracciati ugualmente.

Le routine Commodore di scrittura su nastro prevedono la registrazione dei dati raggruppati in blocchi. Ogni blocco viene inviato al nastro secondo lo standard visto prima.

Il primo blocco, detto «HEADER» o testata, contiene i dati identificativi del file (nome, tipo, ecc.) ed è lungo sempre 192 byte. L'header viene costruito (o caricato in caso di load) nel buffer cassetta posto a partire dal $033c.

Nel caso dei programmi segue il secondo blocco che contiene tutto il programma (vedi fig. 2).

L'header viene riconosciuto dal computer grazie ai dieci secondi di impulsi brevi che lo precedono; tra un blocco e l'altro, invece, vi è una pausa di circa 2 secondi.

I file dati, diversamente dai programmi, vengono suddivisi in tanti blocchi da 192 caratteri.

L'header di una registrazione

Diamo un'occhiata da vicino alla testata delle registrazioni. Il primo byte indica il tipo di file seguente: il codice #$01 indica un programma, mentre #$03 ha invece un significato particolare: esso specifica pure un programma, ma questo verrà caricato senza rilocarlo all'inizio della RAM-Basic ($0801) anche se si è dato solo il comando LOAD.

Ciò vuol dire che il programma verrà comunque posto dove specificano i due byte seguenti nell'header. Infatti, il secondo e terzo byte della testata contengono l'indirizzo da cui cominciare a memorizzare i dati in arrivo dal registratore. Usando il codice $03 seguito dall'indirizzo del vettore di warm-start, si può ottenere una procedura di autostart poiché, indipendentemente dal comando di LOAD dato, il 64 comincerà a caricare dalle locazioni che contengono tale vettore modificandolo. Al termine della lettura il computer farà riferimento proprio a questo vettore per ritornare al Basic con il familiare READY, ed avendolo modificato sarà forzato a saltare all'indirizzo che vi avremo sostituito.

I byte 3 e 4 dell'header contengono il numero della locazione successiva all'ultima che deve essere caricata dal nastro e, infine, dal byte 5 al 192 è spazio riservato al nome del file. Ci sarebbe, quindi, posto per un nome di ben 187 caratteri, ma in realtà, all'atto del caricamento di un programma, ne vengono mostrati solo 16, rendendo superflui tutti gli altri. Teniamo presente, quindi che, in pratica, possiamo disporre di gran parte dell'header per memorizzarci altri dati, come vedremo tra poco.

L'allocazione delle nuove routine

In generale il nostro velocizzatore si comporrà di due routine principali: Flashsave e Flashload. La parte che gestirà il save la allocheremo in $E000 in modo che abilitando tutta la RAM (azzerando il bit 1 del registro $01) avremo disponibile sia le nostre routine sia la RAM da $A000 in poi che normalmente non è registrabile perché mascherata dalla ROM del Basic.

La routine di Flashload la incorporeremo invece nell'header dei programmi da registrare che verrà regi-

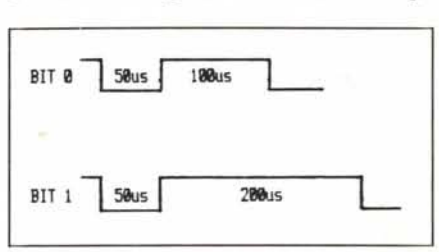


Figura 4 - Codifica dei bit 0 e 1 da parte del Flashtape.

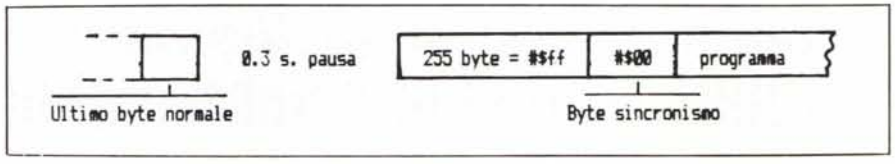


Figura 5 - Composizione parte sincronismo per il flashtape.

```
00002  0000                ;
00003  0000                      SRFLAG=$02E1     ;SYS=80.RUN=01.READY=00
00004  0000                      SYSADR=$02E2
00005  0000                      NAMLEN=$B7       ;LUNGHEZZA NOME
00006  0000                      NAMADR=$BB       ;PUNTATORE NOME
00007  0000                ;
00008  0000                      CHRGET=$0073
00009  0000                      CHRGOT=$0079
00010  0000                ;
00011  0000                ;INIZIO ROUTINE DI SAVE IN FORMATO FLASH
00012  0000                ;
00013  0000                ;REGISTRAZIONE TESTATA CONTENENTE IL FLASHLOAD
00014  0000                ;
00015  0000                      *=$E000
00016  E000                ;
00017  E000                ;
00018  E000                ;CONTROLLO PARAMETRI SAVE
00019  E000                ;
00020  E000  A9 00         BEGIN  LDA #$00
00021  E002  8D E1 02             STA SRFLAG       ;DEFAULT READY
00022  E005  A9 2A                LDA #$2A
00023  E007  8D E2 02             STA SYSADR       ;PREPARA INDIRIZZO PER
00024  E00A  A9 A5                LDA #$A5         ; RELINK E WARM START.
00025  E00C  8D E3 02             STA SYSADR+1
00026  E00F  20 79 00             JSR CHRGOT       ;SE NON VI SONO ALTRI
00027  E012  AA                   TAX              ; MANCANO PARAMETRI
00028  E013  F0 57                BEQ HEADER
00029  E015  C9 3A                CMP #':          ;PRIMO CARATTERE = ': ?
00030  E017  D0 30                BNE ERRLK        ;NO-SALTA A SINTAX ERROR
00031  E019  20 73 00             JSR CHRGET       ;DOPO I DUE PUNTI DEVE
00032  E01C  F0 2B                BEQ ERRLK        ; SEGUIRE IL NOME
00033  E01E  A0 00                LDY #$00
00034  E020  84 B7                STY NAMLEN       ;AZZERA NAMLEN
00035  E022  C9 2C                CMP #'.          ;SALTA IL NOME
00036  E024  F0 26                BEQ SRMODE
00037  E026  A5 7A                LDA $7A          ;AGGIORNA PUNTATORI
00038  E028  85 BB                STA NAMADR       ; ALL'INIZIO DEL NOME
00039  E02A  A5 7B                LDA $7B
00040  E02C  85 BC                STA NAMADR+1
00041  E02E  20 73 00      LENGHT JSR CHRGET       ;LEGGE FILENAME
00042  E031  C9 00                CMP #$00
00043  E033  F0 04                BEQ DONE
00044  E035  C9 2C                CMP #'.
00045  E037  D0 F5                BNE LENGHT
00046  E039  38            DONE   SEC
00047  E03A  A5 7A                LDA $7A
00048  E03C  E5 BB                SBC NAMADR       ;CALCOLA LUNGHEZZA NOME
00049  E03E  85 B7                STA NAMLEN
00050  E040  20 79 00             JSR CHRGOT
00051  E043  C9 00                CMP #$00         ;FINE LINEA BASIC ?
00052  E045  F0 25                BEQ HEADER       ;SI
00053  E047  D0 03                BNE SRMODE       ;NO-ANALIZZA MODO
00054  E049  4C C3 02      ERRLK  JMP ERROR
00055  E04C                ;
00056  E04C                ;ANALIZZA RICHIESTA DI READY/SYS/RUN
00057  E04C                ;
00058  E04C  20 73 00      SRMODE JSR CHRGET
00059  E04F  C9 53                CMP #'S
00060  E051  F0 07                BEQ SYS
00061  E053  C9 52                CMP #'R
00062  E055  D0 F2                BNE ERRLK
00063  E057  A9 01                LDA #$01         ;FLAG X RUN
00064  E059  2C                   .BYTE $2C        ;SALTA ISTRUZIONE SUCC.
00065  E05A  A9 80         SYS    LDA #$80         ;FLAG X SYS
00066  E05C  8D E1 02             STA SRFLAG
00067  E05F  20 CC 02             JSR VALGET       ;LEGGE INDIRIZZO 'SYS'
00068  E062  A5 14                LDA $14          ;IN $14.$15
00069  E064  8D E2 02             STA SYSADR
00070  E067  A5 15                LDA $15
00071  E069  8D E3 02             STA SYSADR+1
00072  E06C                ;
00073  E06C                ;INIZIO REGISTRAZIONE IN FORMATO NORMALE
00074  E06C                ; DELL'HEADER CONTENTE LA ROUTINE DI LOAD
00075  E06C                ;
00076  E06C  20 38 F8      HEADER JSR $F838        ;GESTIONE MESSAGGI E
00077  E06F                                        ; E TASTO PLAY
```



```
00154  E0F7  02                   .BYTE $02,$03    ;INDIRIZZO INIZIO LOAD
00154  E0F8  03
00155  E0F9  04                   .BYTE $04,$03    ;INDIRIZZO DI FINE LOAD
00155  E0FA  03
00156  E0FB                ;
00157  E0FB  81 03         LOADAD .WORD LOAD       ;INDIRIZZO PARTENZA
00158  E0FD                                        ; ROUTINE LOAD VELOCE.
00159  E0FD                ;
00160  E0FD                ;ENTRY POINT SAVE VELOCE
00161  E0FD                ;
00162  E0FD  78            FLSAVE SEI              ;DISABILITA IRQ
00163  E0FE  A9 7F                LDA #$7F         ;DISABILITA NMI
00164  E100  8D 0D DD             STA $DD0D
00165  E103  AD 11 D0             LDA $D011        ;DISABILITA VIDEO
00166  E106  29 EF                AND #$EF
00167  E108  8D 11 D0             STA $D011
00168  E10B  A5 01                LDA $01          ;AVVIA MOTORE
00169  E10D  29 1F                AND #$1F
00170  E10F  09 08                ORA #$08         ;CON SEGNALE ON
00171  E111  85 01                STA $01
00172  E113  20 85 E1             JSR ADRLIM       ;LEGGE LIMITI MEMORIA
00173  E116                                        ; DA REGISTRARE.
00174  E116                ;
00175  E116                ;INVIA 255 BYTE #FF PER LASCIARE REGOLARIZZARE LA
00176  E116                ;VELOCITA' DEL NASTRO
00177  E116                ;
00178  E116  A0 FF                LDY #$FF         ;NUMERO BYTE
00179  E118  A9 FF         HEAD   LDA #$FF         ;BYTE DA INVIARE
00180  E11A  20 4A E1             JSR BYTOUT       ;INVIA BYTE
00181  E11D  88                   DEY              ;SE NON HA FINITO
00182  E11E  D0 F8                BNE HEAD
00183  E120                ;
00184  E120                ;INVIA BYTE DI SINCRONISMO (#$00)
00185  E120                ;
00186  E120  A0 07                LDY #$07
00187  E122  18            SYNCBY CLC              ;BIT = 0
00188  E123  20 5B E1             JSR BITOUT       ;INVIA IL BIT
00189  E126  88                   DEY
00190  E127  10 F9                BPL SYNCBY
00191  E129                ;
00192  E129                ;REGISTRAZIONE MEMORIA
00193  E129                ;
00194  E129  A0 00         NEXTBY LDY #$00
00195  E12B  B1 AC                LDA ($AC),Y      ;LEGGE CARATTERE
00196  E12D  20 4A E1             JSR BYTOUT       ;E LO INVIA SU NASTRO
00197  E130  20 6B 03             JSR BUMPAD       ;AGGIORNA INDIRIZZO
00198  E133  90 F4                BCC NEXTBY       ;SE NON HA FINITO RIPETE
00199  E135  20 8F E1             JSR DELAY        ;ATTENDE 0.3 SEC
00200  E138  A5 01                LDA $01          ;FERMA MOTORE
00201  E13A  29 F7                AND #$F7         ;E SEGNALE OFF
00202  E13C  09 20                ORA #$20
00203  E13E  85 01                STA $01
00204  E140  58                   CLI
00205  E141  18                   CLC
00206  E142  A9 00                LDA #$00
00207  E144  8D A0 02             STA $02A0
00208  E147  4C B9 02             JMP SAVEND       ;RIABILITA SCHERMO
00209  E14A                                        ; E TERMINA
00210  E14A                ;
00211  E14A                ;ROUTINE DI OUTPUT DI UN BYTE
00212  E14A                ;
00213  E14A  85 A4         BYTOUT STA $A4          ;CONSERVA BYTE DA TRASM.
00214  E14C  A9 08                LDA #$08         ;8-BIT(+1)DA INVIARE
00215  E14E  85 A3                STA $A3
00216  E150  18                   CLC              ;PRIMO BIT DI SINCRON.
00217  E151  20 5B E1      NEXTBI JSR BITOUT       ;INVIA BIT
00218  E154  06 A4                ASL $A4          ;PORTA NEL CARRY IL BIT
00219  E156                                        ; SUCCESSIVO
00220  E156  C6 A3                DEC $A3          ;CONTROLLA SE HA FINITO
00221  E158  10 F7                BPL NEXTBI       ;ANCORA
00222  E15A  60                   RTS
00223  E15B                ;
00224  E15B                ;ROUTINE DI INVIO DI UN BIT AL NASTRO
00225  E15B                ;
00226  E15B  A9 64         BITOUT LDA #$64         ;IMPOSTA TIMER-L
00227  E15D  8D 06 DD             STA $DD06        ;SU 100 MICROS.
```

```
00078  E06F  20 8F F6        JSR $F68F       ;MESSAGGIO SAVING ....      2
00079  E072                  ;
00080  E072  A9 3C           LDA #$3C        ;CARICA $033C NEL
00081  E074  85 B2           STA $B2         ; NEL PUNTATORE $B2.$B3
00082  E076  A9 03           LDA #$03
00083  E078  85 B3           STA $B3
00084  E07A                  ;
00085  E07A  A0 BF           LDY #$BF        ;RIPULISCE BUFFER
00086  E07C  A9 20           LDA #$20
00087  E07E  91 B2     CLEAR STA ($B2),Y
00088  E080  88              DEY
00089  E081  D0 FB           BNE CLEAR
00090  E083                  ;
00091  E083  A2 00           LDX #$00
00092  E085  BD F6 E0  HCODE LDA CODE.X      ;COPIA I CODICI INIZIALI
00093  E088  91 B2           STA ($B2),Y     ; DELL'HEADER NECESSARI
00094  E08A  E8              INX             ; PER L'AUTOSTART
00095  E08B  C8              INY
00096  E08C  C0 05           CPY #$05
00097  E08E  D0 F5           BNE HCODE
00098  E090  84 9F           STY $9F         ;SALVA PUNTATORE BUFFER
00099  E092  A0 00           LDY #$00
00100  E094  84 9E           STY $9E         ;PUNTATORE NOME
00101  E096  A4 9E     NAMSET LDY $9E        ;COPIA IL NOME DEL FILE
00102  E098  C4 B7           CPY NAMLEN      ; NEL BUFFER DI CASSETTA
00103  E09A  F0 0C           BEQ NAMEND      ;FINE NOME
00104  E09C  B1 BB           LDA (NAMADR),Y
00105  E09E  A4 9F           LDY $9F
00106  E0A0  91 B2           STA ($B2),Y
00107  E0A2  E6 9E           INC $9E
00108  E0A4  E6 9F           INC $9F
00109  E0A6  D0 EE           BNE NAMSET
00110  E0A8                  ;
00111  E0A8                  ;COPIA ROUTINE DI FLASHLOAD NEL BUFFER CASSETTA
00112  E0A8                  ; I PRIMI 4 BYTE SONO L'INDIRIZZO DI PARTENZA E FINE
00113  E0A8                  ; DEL PROGRAMMA DA REGISTRARE.
00114  E0A8                  ;
00115  E0A8  A0 03     NAMEND LDY #$03
00116  E0AA  B9 2B 00  IOADR  LDA $002B,Y     ;COPIA INDIRIZZI INIZIO
00117  E0AD  99 50 03         STA $0350,Y     ; FINE IN $0350 E $0352
00118  E0B0  88               DEY
00119  E0B1  10 F7            BPL IOADR
00120  E0B3                   ;
00121  E0B3  A0 A8            LDY #$A8
00122  E0B5  B9 97 E1  FLCODE LDA COPY,Y      ;COPIA ROUTINE DI LOAD
00123  E0B8  99 53 03         STA $0353,Y
00124  E0BB  88               DEY
00125  E0BC  D0 F7            BNE FLCODE
00126  E0BE  AD E1 02         LDA SRFLAG      ;AGGIORNA TERMINE ROUT.
00127  E0C1  C9 01            CMP #$01        ; IN BASE AL FLAG DI
00128  E0C3  F0 11            BEQ IOBUF       ; SYS/RUN/READY
00129  E0C5  A9 4C            LDA #$4C
00130  E0C7  8D F0 03         STA FINE
00131  E0CA  AD E2 02         LDA SYSADR
00132  E0CD  8D F1 03         STA FINE+1
00133  E0D0  AD E3 02         LDA SYSADR+1
00134  E0D3  8D F2 03         STA FINE+2
00135  E0D6  20 D7 F7  IOBUF  JSR $F7D7       ;IMPOSTA INDIRIZZO BUF.
00136  E0D9  A9 69            LDA #$69        ;105 SYNC
00137  E0DB  85 AB            STA $AB
00138  E0DD  20 6B F8         JSR $F86B       ;SCRIVE BLOCCO HEADER
00139  E0E0                   ;
00140  E0E0  A9 FB            LDA #<LOADAD    ;AGGIORNA INDIRIZZI I/O
00141  E0E2  85 C1            STA $C1         ;PER PUNTARE AI DUE BYTE
00142  E0E4  A9 E0            LDA #>LOADAD    ;CHE PROVOCANO L'AUTOST.
00143  E0E6  85 C2            STA $C2         ;DELLA ROUTINE DI LOAD
00144  E0E8  A9 FD            LDA #<LOADAD+2
00145  E0EA  85 AE            STA $AE         ;LBYTE INDIRIZZO FINE
00146  E0EC  A9 E0            LDA #>LOADAD
00147  E0EE  85 AF            STA $AF         ;HBYTE
00148  E0F0                   ;
00149  E0F0  20 67 F8         JSR $F867       ;REGISTRA I DUE BYTE
00150  E0F3  4C FD E0         JMP FLSAVE
00151  E0F6                   ;
00152  E0F6  03        CODE  .BYTE 03        ;CODICE PER LOAD NON
00153  E0F7                                  ; NON RILOCATO
```

```
00228  E160  A9 00           LDA #$00        ;IMPOSTA TIMER-H          4
00229  E162  8D 07 DD        STA $DD07
00230  E165  A9 19           LDA #$19        ;AVVIA TIMER
00231  E167  8D 0F DD        STA $DD0F
00232  E16A  A5 01           LDA $01
00233  E16C  49 08           EOR #$08        ;SEGNALE OFF
00234  E16E  85 01           STA $01
00235  E170  A9 02           LDA #$02        ;ATTENDE TIMER
00236  E172  2C 0D DD  NREADY BIT $DD0D
00237  E175  F0 FB           BEQ NREADY
00238  E177  90 05           BCC ZERO        ;SE BIT DA INVIARE = 0
00239  E179                                  ; SALTA DELTA AGGIUNTIVO
00240  E179  A2 14           LDX #$14        ;RITARDO DI 100 MICROS.
00241  E17B  CA        LOOP  DEX
00242  E17C  D0 FD           BNE LOOP
00243  E17E  A5 01     ZERO  LDA $01         ;SEGNALE ON DI NUOVO
00244  E180  49 08           EOR #$08
00245  E182  85 01           STA $01
00246  E184  60              RTS
00247  E185  A2 03     ADRLIM LDX #$03       ;RILEVA I LIMITI DEL
00248  E187  BD 50 03  L2     LDA $0350.X     ; PROGRAMMA DA REGISTR.
00249  E18A  95 AC           STA $AC.X       ; DALLA TESTATA
00250  E18C  CA              DEX
00251  E18D  10 F8           BPL L2
00252  E18F  A0 00     DELAY LDY #$00        ;ATTENDE CIRCA 0.3 SEC.
00253  E191  88        L3    DEY
00254  E192  D0 FD           BNE L3
00255  E194  CA              DEX
00256  E195  D0 F8           BNE DELAY
00257  E197  60        COPY  RTS
00258  E198                  ;
00259  E198                  ;ROUTINE DI LOAD VELOCE MEMORIZZATA NELLA TESTATA
00260  E198                  ; (HEADER) DEL PROGRAMMA. NEL BUFFER DI CASSETTA
00261  E198                  ; ($033C).
00262  E198                  ;
00263  E198                        *=$0354
00264  0354                  ;
00265  0354                  ;ROUTINE DI BITIN (INPUT DI UN BIT)
00266  0354                  ;
00267  0354  A9 10     BITIN LDA #$10
00268  0356  2C 0D DC  WAIT  BIT $DC0D       ;ATTENDE SEGNALE DI SYNC
00269  0359  F0 FB           BEQ WAIT
00270  035B  AD 0D DD        LDA $DD0D       ;RILEVA STATO TIMER
00271  035E  A2 00           LDX #$00
00272  0360  8E 07 DD        STX $DD07       ;RIIMPOSTA TIMER
00273  0363  A2 19           LDX #$19
00274  0365  8E 0F DD        STX $DD0F       ;RIAVVIA TIMER
00275  0368  4A              LSR A
00276  0369  4A              LSR A           ;RIPORTA BIT NEL CARRY
00277  036A  60              RTS
00278  036B                  ;
00279  036B                  ;ROUTINE INCREMENTO INDIRIZZO LOAD
00280  036B                  ;
00281  036B  E6 AC     BUMPAD INC $AC
00282  036D  D0 02           BNE L4
00283  036F  E6 AD           INC $AD
00284  0371  A6 AC     L4    LDX $AC
00285  0373  A4 AD           LDY $AD         ;SE FINE CARRY=1
00286  0375  E4 AE           CPX $AE
00287  0377  D0 06           BNE NOC
00288  0379  C4 AF           CPY $AF
00289  037B  D0 02           BNE NOC
00290  037D  38              SEC
00291  037E  24              .BYTE $24       ;SALTA ISTRUZIONE SUCCESS.
00292  037F  18        NOC   CLC
00293  0380  60              RTS
00294  0381                  ;
00295  0381                  ;ENTRY POINT ROUTINE LOAD VELOCE
00296  0381                  ;
00297  0381  78        LOAD  SEI
00298  0382  AD 11 D0        LDA $D011       ;DISABILITA VIDEO
00299  0385  29 EF           AND #$EF
00300  0387  8D 11 D0        STA $D011
00301  038A  A5 01           LDA $01         ;AVVIA MOTORE
00302  038C  29 1F           AND #$1F
00303  038E  85 01           STA $01
```

(continua a pagina 128)

(segue da pagina 127)

```
00304  0390  A2 03          LDX #$03        :RILEVA I LIMITI DEL
00305  0392                                 : PROGRAMMA DA CARICARE
00306  0392  BD 50 03  L9    LDA $0350.X     :DALLA TESTATA
00307  0395  95 AC          STA $AC.X
00308  0397  CA             DEX
00309  0398  10 F8          BPL L9
00310  039A  A0 00          LDY #$00        :ATTENDE CIRCA 0.3 SEC.
00311  039C  88       L10   DEY
00312  039D  D0 FD          BNE L10
00313  039F  CA             DEX
00314  03A0  D0 FA          BNE L10
00315  03A2                 ;
00316  03A2                 :RICEVE 255 SEGNALI PER PERMETTERE LA STABILIZZAZIONE
00317  03A2                 :DELLA VELOCITA' DEL NASTRO
00318  03A2                 ;
00319  03A2  A0 FF          LDY #$FF
00320  03A4  A9 10    L6    LDA #$10
00321  03A6  2C 0D DC WAIT2 BIT $DC0D       :RILEVA SEGNALE
00322  03A9  F0 FB          BEQ WAIT2
00323  03AB  88             DEY
00324  03AC  D0 F6          BNE L6
00325  03AE                 ;
00326  03AE                 :RILEVA BYTE DI SINCRONIZZAZIONE (#$00)
00327  03AE                 ;
00328  03AE  A9 BA          LDA #$BA        :IMPOSTA TIMER-L
00329  03B0  8D 06 DD       STA $DD06       :SU 186 MICROS.
00330  03B3  A9 01    NOSYNC LDA #$01       :FLAG SYNC NON RICEVUTO
00331  03B5  48       SYNC  PHA
00332  03B6  20 54 03       JSR BITIN       :RICEVE BIT
00333  03B9  68             PLA             :RECUPERA BYTE IN
00334  03BA                                 : RICEZIONE
00335  03BA  B0 F7          BCS NOSYNC      :SE BIT=1 NON E' IL
00336  03BC                                 : BYTE DI SINCRONISMO
00337  03BC  2A             ROL A           :CONSERVA BIT RICEVUTO
00338  03BD  C9 00          CMP #$00        :SE BYTE=$00 FINE
00339  03BF  D0 F4          BNE SYNC
00340  03C1                 ;
00341  03C1                 :LETTURA DATI DAL NASTRO
00342  03C1                 ;
00343  03C1  A9 08    BYTEIN LDA #$08       :8 BIT (+1) DA RICEVERE
00344  03C3  85 A3          STA $A3
00345  03C5  20 54 03 L1    JSR BITIN       :RICEVE BIT
00346  03C8  26 A4          ROL $A4         :BIT IN $A4
00347  03CA  C6 A3          DEC $A3         :-1 BIT DA RICEVERE
00348  03CC  10 F7          BPL L1
00349  03CE  A5 A4          LDA $A4         :BYTE RIC. IN (A)
00350  03D0
00351  03D0  A0 00          LDY #$00
00352  03D2  91 AC          STA ($AC).Y     :MEMORIZZA BYTE
00353  03D4  20 6B 03       JSR BUMPAD      :AGGIORNA INDIRIZZO
00354  03D7  90 E8          BCC BYTEIN      :RIPETE SE C=0
00355  03D9  A9 83          LDA #$83
00356  03DB  8D 02 03       STA $0302
00357  03DE  A9 A4          LDA #$A4
00358  03E0  8D 03 03       STA $0303
00359  03E3  A9 00          LDA #$00
00360  03E5  8D A0 02       STA $02A0
00361  03E8  20 93 FC       JSR $FC93       :FERMA MOTORE E ABILITA
00362  03EB                                 : SCHERMO.
00363  03EB  58             CLI
00364  03EC  86 2D          STX $2D         :AGGIORNA FINE PROGRAMMA
00365  03EE  84 2E          STY $2E
00366  03F0  A9 A7    FINE  LDA #$A7        :FINE PREDISPOSTA
00367  03F2  48             PHA             : PER IL RUN AUTOMATICO
00368  03F3  A9 E9          LDA #$E9
00369  03F5  48             PHA
00370  03F6  A9 00          LDA #$00
00371  03F8  4C 71 A8       JMP $A871
00372  03FB                 ;
00373  03FB                 :INIZIO ROUTINE SAVE VELOCE IN $02A7
00374  03FB                 ;
00375  03FB                 *=$02A7
00376  02A7                 ;
00377  02A7  A5 BA    SAVEIN LDA $BA        :SE PERIFERICA DIVERSA
00378  02A9  C9 01          CMP #$01        :DA 1 (REGISTRATORE)
00379  02AB  F0 03          BEQ OK          :VA AL LOAD NORMALE
00380  02AD  4C ED F5       JMP $F5ED
00381  02B0  A5 01    OK    LDA $01         :ABILITA RAM IN $E000
00382  02B2  29 FD          AND #$FD
00383  02B4  85 01          STA $01
00384  02B6  4C 00 E0       JMP BEGIN       :CONTROLLO SINTASSI E
00385  02B9                                 : INIZIO SAVE.
00386  02B9  A5 01    SAVEND LDA $01        :RIABILITA ROM IN $E00
00387  02BB  09 02          ORA #$02
00388  02BD  85 01          STA $01
00389  02BF  18             CLC
00390  02C0  4C 93 FC       JMP $FC93       :RIABILITA SCHERMO E
00391  02C3                                 : TORNA AL BASIC
00392  02C3                 ;
00393  02C3  A5 01    ERROR LDA $01
00394  02C5  09 02          ORA #$02
00395  02C7  85 01          STA $01
00396  02C9  4C 08 AF       JMP $AF08       :SYNTAX ERROR
00397  02CC  A5 01    VALGET LDA $01
00398  02CE  09 02          ORA #$02
00399  02D0  85 01          STA $01
00400  02D2  20 73 00       JSR CHRGET
00401  02D5  20 6B A9       JSR $A96B
00402  02D8  A5 01          LDA $01
00403  02DA  29 FD          AND #$FD
00404  02DC  85 01          STA $01
00405  02DE  60             RTS
```

Listato 1 - Prima parte del programma Flashtape. La seconda parte ed i commenti dettagliati sulla struttura globale del programma li troverete sul prossimo numero.

strato in forma standard per permetterne la ricerca da parte del Commodore senza l'ausilio di alcuna routine particolare. Avremo, allora un header normale (contenente il flashload), seguito da un mini-programma fatto di soli due byte, necessari a lanciare l'esecuzione del flashload, a cui seguirà il programma in formato veloce.

La figura 3 dovrebbe illustrare meglio il concetto.

I due byte contenenti l'indirizzo di inizio della routine di flashload verranno caricati nel vettore di warm start per effetto dei primi 5 byte dell'header cosicché, il microprocessore, invece di tornare al Basic, passerà all'esecuzione della nostra routine che si occuperà di caricare il programma registrato di seguito in formato flash.

Il nuovo standard di registrazione

Al contrario dello standard Commodore che per i singoli bit prevede segnali di lunghezza totale costante, noi utilizzeremo segnali di durata variabile e tramite il controllo della parte positiva del segnale, stabiliremo la presenza di bit 0 o 1.

Ogni bit è misurato a partire dal termine del precedente. Vi è un livello basso di circa 50 microsecondi, seguito da un livello alto della durata di 100 microsecondi per un bit 0 e di 200 per un bit 1; ogni byte è composto da nove bit, di cui, il primo, serve solo a dare il sincronismo ai seguenti. È facile immaginare come questo sistema di registrazione dei dati sia molto sensibile alle variazioni di velocità. Poiché dopo aver caricato i due byte che seguono l'header il nastro ha un breve arresto, bisognerà fare in modo che il motorino raggiunga nuovamente la velocità esatta prima di ricominciare la lettura. Inoltre, non avendo un segnale particolare che indichi l'inizio di un byte (ricordiamo che il Kernal usa a tale scopo un impulso lungo diverso da quelli che codificano i singoli bit) dobbiamo essere sicuri di sincronizzarci con il primo bit in assoluto, altrimenti si corre il rischio di leggere tutti i dati shiftati di uno o più bit.

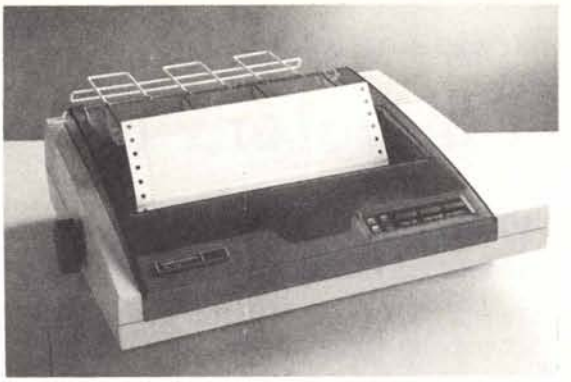
A tale scopo abbiamo fatto precedere il programma da una pausa di circa 0.3 secondi, seguita da una sequenza di 255 byte di valore #$ff (tutti bit = 1) con un byte finale di sincronismo di valore #$00 (8 bit = 0). Poiché il valore 1 ha una durata su nastro superiore a quella del valore 0, anche se inizialmente il nastro non ha ancora raggiunto la giusta velocità non è possibile che un 1 venga scambiato per 0. Pertanto, la routine di flashload resta in attesa della sequenza di 8 bit a 0 per iniziare la lettura del programma dal primo bit seguente. MC

MASTER BIT

MI.PE.CO. VENDITA PER CORRISPONDENZA

INTERFACCIA PARLANTE CURRAH L. 75.000

Scrivete le parole da pronunciare "Lei" le leggerà:
LET S$ = "sAlve" enter
sentirete la parola salve dall'altoparlante del T.V.
Molti programmi prevedono già il suo uso (Birds and the Bees, Lunar jet man, maziacs, VOICE CHESS ecc.
Compreso nel prezzo manuale completo in italiano più un programma compilatore per farle pronunciare in italiano qualsiasi parola richiesta.
Parla attraverso il televisore con chiara voce sintetica.



MANNESMANN MT 80+

L. 599.000

80 col. - 100 cps - interfaccia Centronics - foglio singolo e modulo continuo - bidirezionale.

QL SINCLAIR 128K 599.000

Tutto compreso 6 mesi di garanzia

CPU MOROTOLA 68008 da 32 BIT e 2 microdrive. Ultima versione con nuovi programmi, alimentatore, manuale in inglese, manuale in italiano, 4 cartucce con i 4 programmi gestionali + 1 cartuccia con giochi originali (PIRATE, ZETA, PED, GUN, SREAKOUT, HUNT) e in regalo un ottimo copiatore per mdv e floppy di Massimo Rossi

SPECTRUM 48K PLUS 299.000

Tutto compreso 6 mesi di garanzia

con lo SPECTRUM plus **manuale in italiano** *e in regalo 5 programmi in italiano (conto corrente, grafica, funzioni, bioritmi, esapedone + il Supercopiatore di Massimo Rossi)*

QL 512 K 949.000
Espansione da 512K montata internamente, non necessita di alimentazione supplementare e lascia il connettore libero per altre periferiche.

nuovo SPECTRUM 48K + ... 299.000
manuale in italiano, cavetti alimentatore, cassette dimostrative e oltre 50.000 lire di software originale e in italiano.

STAMPANTE ALPHACOM 32 ..... 149.000
per Spectrum e ZX 81 istruzioni in italiano 2 rulli di carta in regalo.

10 RULLI di carta termica 29.000

MANNESMANN TALLY tutti i modelli

MT 80+ 599.000
foglio singolo e continuo, interfaccia Centronics, 100 cps vari set di caratteri - Bidirezionali.

MT85 ..... 899.000
interfaccia Centronics o seriale a scelta 180 cps 80/136 col. foglio singolo e continuo.

DISCHI 3"½ 13.000
DISCHI 3"½ 10 pezzi 100.000
garantiti doppia faccia e doppia densità

INTERFACCIA PER JOYSTICK UNA PRESA 29.000
tipo Kempston, per tutti i joystick stand. 9 PIN D.

INTERFACCIA PER JOYSTICK DUE PRESE 39.000
tipo Kempston, per tutti i Joystick stand. 9 PIN D.

JOYSTICK STANDARD 9 PIN D 15.000

CONVERTITORE 99.000
Da RS232 a Centronics per interfaccia 1 o per QL **cavi e connettori speciali compresi.**

INTERFACCIA CENTRONICS SPECTRUM 99.000
senza software tutto su Rom compreso il copy

8 CARTUCCE x MICRODRIVE 49.000

TRISLOT 27.000
presa tripla per connettore Spectrum

MANUALE IN ITALIANO x SPECTRUM 16.000
«Come usare il tuo Spectrum»

ROM «JS» NUOVO TIPO (256K + 128K) . 99.000
trasforma il tuo QL in un «JS».

INTERFACCIA PARLANTE CURRAH 75.000
manuale completo in italiano.

ESPANSIONE + 32K x SPECTRUM 59.000
issue 2 o 3 specificare, facilissima da montare, istruzioni dettagliate in italiano con fotografie, porta il VS. Spectrum da 16 K a 48 K. Montaggio gratis.

TASTIERA DELLO SPECTRUM PLUS 85.000
Kit per trasformare lo Spectrum normale in PLUS

DISK DRIVE 3"½ x INTERF. x QL 619.000
Oltre 700K formattati

DISK DRIVE 3"½ INTERF. x SPECTRUM 519.000
Oltre 700K formattati

KIT DI ESPANSIONE x QL A 512 ..... 249.000
Si monta all'interno del QL, si consiglia l'assistenza di un tecnico specializzato.

ESPANSIONE DEL VOSTRO QL A 512K 349.000
Montata all'interno del Vostro QL e collaudata con garanzia di 3 mesi spedite il Compûter solo dopo aver avuto un contatto telefonico.

TOOLKIT II x QL SU ROM 89.000

MI.PE.CO. VENDITA PER CORRISPONDENZA

PARTI DI RICAMBIO PER SPECTRUM E QL

GARANZIA 48H: oltre la normale Garanzia di 6 mesi per i Computer e di 3 mesi per gli accessori, la MI.PE.CO. si impegna a sostituire tutto il materiale trovato malfunzionante, entro 48 ore dal ricevimento.

AVVERTENZE - tutti i prezzi sono comprensivi di IVA e spese postali - per ordini inferiori alle 50.000 lire aggiungere L. 5.000 per contributo spese di spedizione — pagamento contrassegno al ricevimento del pacco — è gradito un contatto telefonico — **sconti quantità.**
Listino prezzi aggiornato anche su richiesta telefonica.

MI.PE.CO. Cas. Postale 3016 - 00121 ROMA (OSTIA)

ORDINI TELEFONICI (ore 8.30/19.30): 06/5611251

NELLE TUE MANI

tutta la potenza di una grande stampante

P-40 ideale per home e personal computer

Questa è Epson P-40, la stampante termica ultracompatta, quasi tascabile, la compagna ideale per il tuo personal computer a casa, a scuola e anche nel lavoro.

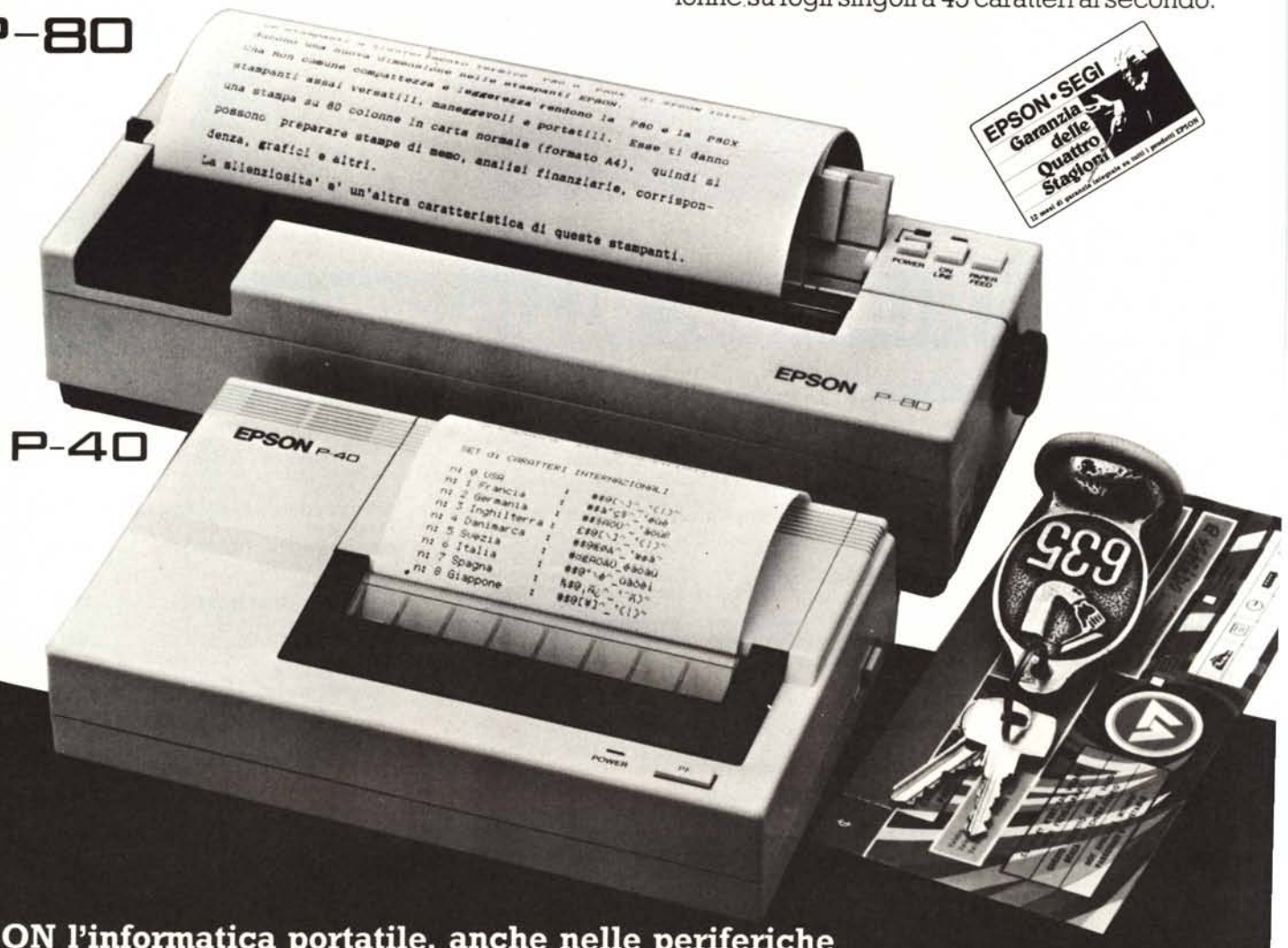
Piccola, robusta, progettata per lavorare a lungo e realizzata con la proverbiale qualità Epson, la P-40 funziona con batterie ricaricabili e stampa grafici e testi su 20, 40 o 80 colonne (modo compresso) a 45 caratteri al secondo.

Regala Epson P-40 al tuo personal. Con la piccola Epson il tuo personal diventa grande!

P-80 e P-80X la qualità di stampa professionale

Con la nuova P-80 e il tuo personal computer hai la stessa qualità delle stampanti a matrice da tavolo a 80 colonne per produrre prospetti proposte d'acquisto, tabelle o listini di elevata qualità su carta termica o su carta comune. Se poi desideri una qualità di stampa virtualmente indistinguibile da quella delle macchine da scrivere, scegli P-80X, con i suoi 24 "aghi" capaci di produrre caratteri pieni e netti, autorevoli, per la tua corrispondenza più importante. Quando vuoi, dove vuoi.

P-80 e P-80X stampano su 40, 80 e 136 colonne su fogli singoli a 45 caratteri al secondo.



studio martinetti

EPSON l'informatica portatile, anche nelle periferiche

EPSON SEGI

EPSON-SEGI S.p.A.
20124 Milano - Via Timavo, 12 - Tel. 02-6709136-7-8-9-0
40121 Bologna - Via Pietramellara, 6S A/B - Tel. 051-273686
35128 Padova - Via Pellizzo, 23/9 int. 4/U - Tel. 049-8070870
00199 Roma - Via Asmara, 58 - Tel. 06-8395766-8394458