

## Sprite 6499

### Programma per l'adattatore telematico 6499 del C64

di Flavio Palermo - Ventimiglia

Ogni giorno, quando ci capita di dover utilizzare il Servizio Videotel della SIP, per poterne trarre tutte le informazioni che ci possono aiutare nel nostro lavoro, o che ci aggiornano su determinati settori della nostra società, compiamo dei gesti molto semplici, quasi banali, come premere alcuni tasti prestabiliti della tastiera di un computer, o di un qualsiasi terminale. Bene, a me è venuto il desiderio di conoscere ed arrivare a capire che cosa avviene dentro l'adattatore telematico 6499 del C64.

Alla fine di tutte le mie ricerche, sono riuscito a trovare i punti focali, sui quali intervenire, per aggiungere qualche possibilità in più al programma di comunicazione con il VDT: in particolare, nella versione che qui vi presento, ho realizzato uno sprite a forma di freccia, che si muove sullo schermo, mentre siamo collegati con il VDT, dandoci la possibilità di selezionare i vari argomenti, senza utilizzare la tastiera.

Per introdurre in memoria il programma, ho usato un caricatore Basic. Dopo aver digitato il listato del caricatore (fate molta attenzione a trascrivere le linee DATA in modo corretto, perché basta un solo codice sbagliato, per provocare il blocco del sistema), lo registrate su disco, o su nastro, e poi, subito dopo, digitate in modo diretto:

```
POKE 8192,0: POKE 44,32: NEW
```

Quindi, caricate il programma caricatore, e date il RUN.

Dopo alcuni secondi, vi verrà chiesto se volete utilizzare il disco oppure il nastro, dove verrà registrato il programma vero e proprio.

Per utilizzare quest'ultimo, si deve:  
1) accendere il C64, con l'adattatore telematico inserito; 2) tornare al Basic per mezzo del tasto F8; 3) caricare e far eseguire il programma Sprite.

Se tutto funziona correttamente, vi troverete nel menu principale del 6499.

*È disponibile, presso la redazione, il disco con i programmi pubblicati in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 247.*

A questo punto, premete il tasto F5, per selezionare il menu Videotex. Avrete sicuramente notato che, in alto a sinistra dello schermo, è apparso uno sprite a forma di freccia, ma attenzione: fin quando non sarete collegati con il VDT, lo sprite non viene attivato.

Dopo aver chiamato il 165, per mezzo dell'agenda su disco o direttamente sulla tastiera, potete utilizzare lo sprite, per mezzo del joystick in porta 2, e cliccare con il tasto fire gli argomenti che vi interessano. Vi è possibile modificare la velocità dello sprite, tenendo premuto il tasto fire e posizionando la leva del joystick in una delle quattro posizioni.

Cerchiamo, ora, di descrivere, in linea generale, la struttura ed il funzionamento del 6499.

È noto a tutti che il 6499 è un modem con software di comunicazione, incorporato in una ROM da 32K. In realtà quest'ultima, però, è composta da due ROM sovrapposte, che chiameremo ROM1 e ROM2, mappate dalla locazione \$ 8000 alla locazione \$ BFFF.

Per selezionare la ROM1 oppure la ROM2, occorre, rispettivamente, settare o resettare il BIT3 della locazione I/O \$DE01:

Es.:

```
seleziona ROM1
LDA #$3C          00111100
STA $DE01
seleziona ROM2
LDA #$34          00110100
STA $DE01
```

Mentre, per attivare o disattivare la ROM selezionata in precedenza, bisogna rispettivamente settare o resettare il BIT3 della locazione I/O \$DE03:

Es.:

```
attiva ROM selezionata
LDA #$3E          00111110
STA $DE03

disattiva la ROM selezionata
LDA #$36          00110110
STA $DE03
```

All'accensione del C64, con l'adattatore telematico 6499, viene attivata la ROM1, dove sono abilitate tutte le funzioni del menu principale, ad esclusione del tasto F5. Quando si seleziona quest'ultimo, viene attivata la ROM2, che ci permette di accedere al menu Videotex, in cui sono attivate le seguenti funzioni:

- \* F4: modo disco;
- \* F5: visualizza frame;
- \* F7: editor mailbox,

mentre quando si selezionano le funzioni:

- \* F1: chiamata manuale;
- \* F3: chiamata da file;
- \* F8: ritorno al menu principale,

avviene una commutazione dalla ROM2 alla ROM1.

Prendendo in esame la routine di commutazione del 6499, da ROM1 a ROM2, e quella opposta, osserviamo che esse sono identiche, e si trovano sia sulla ROM1 che sulla ROM2: la prima è allocata dalla locazione \$800D, e la seconda dalla locazione \$802A.

Es.:

routine di commutazione da ROM1 a ROM2

```
800D LDA $DE01
8010 EOR #$08
8012 STA $DE01
```

routine di commutazione da ROM2 a ROM1

```
802A LDA $DE01
802D EOR #$08
802F STA $DE01
```

#### Nota

I codici di controllo nei listati sono riportati in forma «esplicita», in conseguenza dell'impiego della stampante Star NL-10 e relativa interfaccia per Commodore. Ovviamente, nella digitazione del programma è necessario usare i consueti tasti che corrispondono alle indicazioni fra parentesi: ad esempio cursore destro per (RGHT), CTRL-3 per (RED) eccetera.

(CLR)	=		(YEL)	=	
(HOME)	=		(RVS)	=	
(DOWN)	=		(OFF)	=	
(UP)	=		(ORNG)	=	
(RGHT)	=		(BRN)	=	
(LEFT)	=		(LRED)	=	
(BLK)	=		(GRY1)	=	
(WHT)	=		(GRY2)	=	
(RED)	=		(LGRN)	=	
(CYN)	=		(LBLU)	=	
(PUR)	=		(GRY3)	=	
(GRN)	=		(SWLC)	=	
(BLU)	=				

La prima routine viene chiamata quando giunge a noi il segnale VDT (dopo la chiamata da file o manuale), oppure, subito dopo la pressione del tasto F5 dal menu principale, per accedere al menu Videotex.

La seconda routine, invece, viene chiamata quando dal menu Videotex si premono F1, F3, F8.

Chiaramente, se noi dal menu principale, selezioniamo, tramite F5, il menu Videotex, e poi premiamo F1 o F3, avvengono tre commutazioni:

I commutazione: da ROM1 a ROM2 (dopo aver premuto F5 dal menu principale);

II commutazione: da ROM2 a ROM1 (per comporre il 165 manualmente o da file);

III commutazione: da ROM1 a ROM2 (quando giunge il segnale VDT).

Se vogliamo inserire una routine, che giri contemporaneamente al PRG di comunicazione, dobbiamo effettuare alcune modifiche a quest'ultimo, che deve essere trasferito in RAM.

Precedentemente, abbiamo detto che la ROM del 6499 è composta da ROM1+ROM2, e che la locazione \$DE01 ci permette la commutazione da una ROM all'altra.

Dovendo trasferire la ROM del 6499 nella RAM, sarebbe comodo avere a disposizione una locazione di I/O, che ci permetta la commutazione di due RAM. Ma questa è un'utopia, perché nel C64 esiste un solo banco di RAM da 16K, allocato da \$8000 a \$BFFF sotto la ROM del 6499. Perciò, ho dovuto ideare un sistema che mi permettesse di effettuare una commutazione da ROM1/ROM2, sempre sullo stesso banco di RAM.

Una volta che la ROM, attiva in quell'istante, viene copiata sulla RAM sottostante, per mezzo della subroutine, alla locazione \$7FA3, tramite i sottoprogrammi alle locazioni \$7E07, \$7E1B, \$7E3E, viene effettuata una modifica del PRG originale agli indirizzi \$800D e \$802A.

Es.:

originale ROM		modifica RAM	
800D LDA	\$DE01	800D	NOP
8010 EOR	#\$08	800E	NOP
8012 STA	\$DE01	800F	NOP
		8010	NOP
		8011	NOP
		8012	JSR\$
			indirizzo

In questo modo, quando il PRG do-

vrebbe compiere una commutazione, viene deviato agli indirizzi \$7DB8, \$7DD8, \$7DE7, i quali provvedono a:

- 1) attivare la ROM del Basic e del 6499;
- 2) eseguire la commutazione da ROM1 a ROM2 o viceversa;
- 3) copiare la ROM attivata nella RAM sottostante;
- 4) disattivare la ROM del 6499 e la ROM del Basic;
- 5) effettuare le modifiche opportune.

Quello che mi ha lasciato perplesso è che, se si cerca di attivare la ROM del 6499, senza prima attivare la ROM del Basic, tramite la locazione \$00, non si ottiene il risultato voluto. Oltre ai sottoprogrammi di modifica per le commutazioni, già illustrati, esiste un'altra modifica (\$7E2F), che permette la deviazione dal PRG principale della ROM2 sul PRG di gestione dello sprite. E cioè:

originale ROM2		modifica RAM
AF19 LDY	\$C6	A519 JMP \$7E55
A51B BEQ	\$A52F	
A51D		prosegue programma

#### Sprite 6499.

```

1 DATA 011,008,010,000,158,050,048,054
2 DATA 050,000,000,000,145,169,076,133
3 DATA 247,169,008,133,248,169,125,133
4 DATA 250,169,168,133,249,160,000,177
5 DATA 247,145,249,230,247,208,002,230
6 DATA 248,230,249,208,002,230,250,165
7 DATA 247,201,162,165,248,233,010,144
8 DATA 230,076,168,125,234,015,009,022
9 DATA 001,012,006,032,015,013,018,005
10 DATA 012,001,016,032,248,125,032,163
11 DATA 127,032,062,126,206,070,128,108
12 DATA 252,255,234,152,072,032,246,125
13 DATA 032,001,126,032,163,127,032,007
14 DATA 126,032,027,126,032,047,126,173
15 DATA 067,003,208,003,032,196,127,104
16 DATA 168,096,234,152,072,032,246,125
17 DATA 032,163,127,032,062,126,104,168
18 DATA 096,234,152,072,032,246,125,032
19 DATA 163,127,032,062,126,104,168,096
20 DATA 234,230,001,169,062,141,003,222
21 DATA 169,060,208,002,169,052,141,001
22 DATA 222,096,160,008,185,018,126,153
23 DATA 041,128,136,208,247,096,234,234
24 DATA 234,234,234,032,231,125,160,008
25 DATA 185,038,126,153,012,138,136,208
26 DATA 247,096,234,234,234,234,234,032
27 DATA 216,125,160,003,185,058,126,153
28 DATA 024,165,136,208,247,096,076,085
29 DATA 126,160,008,185,076,126,153,012
30 DATA 128,153,041,128,136,208,244,096
31 DATA 234,234,234,234,234,032,184,125
32 DATA 164,198,240,003,076,029,165,173
33 DATA 000,220,044,103,126,308,005,076
34 DATA 008,127,016,234,201,119,240,049
35 DATA 201,123,240,084,201,125,240,007
36 DATA 201,126,240,021,076,047,165,024
37 DATA 173,001,208,105,001,201,245,206
38 DATA 002,169,000,141,001,208,076,247
39 DATA 126,056,173,001,208,233,001,208
40 DATA 002,169,245,141,001,208,076,247
41 DATA 126,024,173,000,208,105,001,141
42 DATA 000,208,173,016,208,105,000,141
43 DATA 016,208,208,003,076,247,126,173
44 DATA 000,208,201,085,208,246,169,000
45 DATA 141,000,208,141,016,208,240,236
46 DATA 056,173,000,208,233,001,141,000
47 DATA 208,173,016,208,233,000,141,016
48 DATA 208,208,003,076,247,126,173,000
49 DATA 208,201,255,208,005,169,085,141
50 DATA 000,208,201,000,208,247,169,255

```

```

51 DATA 141,000,208,169,000,141,016,208
52 DATA 240,225,169,008,160,255,136,208
53 DATA 253,056,233,001,208,246,234,234
54 DATA 076,047,165,201,111,240,010,041
55 DATA 015,073,015,141,248,126,076,047
56 DATA 165,169,000,133,249,133,250,133
57 DATA 247,133,248,056,173,001,208,233
58 DATA 050,074,074,074,133,247,160,000
59 DATA 024,165,249,101,247,133,249,165
60 DATA 250,105,000,133,250,200,192,040
61 DATA 208,238,234,234,056,173,000,208
62 DATA 233,024,133,247,173,016,208,041
63 DATA 001,233,000,133,248,160,000,070
64 DATA 248,102,247,200,192,003,208,247
65 DATA 234,024,165,247,101,249,133,247
66 DATA 165,248,101,250,133,248,234,024
67 DATA 169,080,101,247,133,247,169,023
68 DATA 181,248,133,248,234,160,000,177
69 DATA 247,201,048,048,006,201,058,016
70 DATA 016,048,007,201,031,208,010,024
71 DATA 105,061,141,119,002,169,001,133
72 DATA 198,169,128,024,105,001,208,251
73 DATA 200,208,246,076,085,126,169,000
74 DATA 133,247,169,128,133,248,160,000
75 DATA 177,247,145,247,200,208,249,230
76 DATA 248,165,248,201,192,208,241,169
77 DATA 054,141,003,222,133,001,096,160
78 DATA 000,185,230,127,153,064,003,200
79 DATA 192,025,234,208,244,140,000,208
80 DATA 238,021,208,169,013,141,248,007
81 DATA 169,050,141,001,208,234,234,234
82 DATA 096,000,000,000,254,000,000,252
83 DATA 000,000,248,000,000,252,000,000
84 DATA 238,000,000,199,000,000,131,128
85 DATA 000,001
86 PRINTCHR$(147);:AS=CHR$(17)+CHR$(17)
87 IFPEEK(44)=32THEN89
88 PRINTAS:"POKE44,32:POKE8192,0:NEW":PRINTAS:AS:"R1:ARICA:
L PRG":GOTO98
89 PRINTTAB(12):"ATTENDERE PREGO"
90 PORT=0TO673:READA:B=B+A:POKE2049+I,A:NEXT
91 IF B<>87752THENPRINTAS:"ERRORE NELLE ISTRUZIONI: DATA":END
92 PRINTCHR$(147):TAB(10):"DISCO O NASTRO (D/N)"
93 GETC$:IFC$>"D"ANDC$<"N"THEN 93
94 IF C$="D"THEN D=8:GOTO96
95 D=1
96 PRINTCHR$(147);:AS:"POKE43,1:POKE44,6:POKE45,16:POKE46,10"
97 PRINTAS:"SAVE":CHR$(34):"SPRITE 6499":CHR$(34):",":D
98 POKE198,2:POKE631,13:POKE632,13:PRINTCHR$(147);

```

Con questa modifica, il PRG principale, invece di controllare il numero di caratteri del buffer della tastiera (\$C6), viene deviato all'indirizzo \$7E55, dove inizia la routine sprite.

Chiaramente, l'istruzione di controllo del buffer, si trova proprio alla locazione \$7E55.

Se nel buffer è presente un carattere, il PRG sprite provvede a restituire il controllo al PRG principale, tramite un JMP \$A51D; viceversa, se nel buffer della tastiera non è presente nessun carattere, viene controllata la porta 2 del joystick, per rilevare la posizione di quest'ultimo.

Le routine che permettono la gestione dello sprite, sono le seguenti:

\$7E9E - \$7EC4: movimento verso destra  
 \$7EC5 - \$7EF6: movimento verso sinistra  
 \$7E8E - \$7E9D: movimento verso l'alto  
 \$7E7C - \$7E8D: movimento verso il basso  
 \$7EF7 - \$7F07: routine di ritardo dello sprite  
 \$7F08 - \$7F15: modifica velocità sprite.

Il sottoprogramma che invia al VDT il

carattere sotto lo sprite, quando si esprime il tasto fire del joystick, invece, è allocato dalla locazione \$7F16 - \$7FA2. Quest'ultimo esegue la trasformazione delle coordinate dello sprite in un indirizzo della memoria, che corrisponde al carattere sotto lo sprite, ed invia il carattere al buffer della tastiera, proprio come se avessimo premuto il tasto corrispondente.

La formula che ci permette di individuare l'indirizzo del carattere sotto lo sprite, è la seguente:

$$I = M + 40 \times \text{INT}(Y/8) + \text{INT}(X/8)$$

dove X e Y sono le coordinate dello sprite, ed M è l'indirizzo \$1750, che rappresenta la prima locazione della memoria in cui viene memorizzata la pagina VDT, prima di essere elaborata, e visualizzata sullo schermo.

Passiamo, ora, ad analizzare le routine che eseguono questa formula:

\$7F16 - \$7F1F: azzera puntatori F7, F8, F9, FA;  
 \$7F20 - \$7F2A: esegue INT(Y/8);  
 \$7F2B - \$7F40: somma 40 volte il valore ottenuto dalla precedente:  $40 \times \text{INT}(Y/8)$ ;  
 \$7F41 - \$7F5D: esegue INT(X/8);  
 \$7F5E - \$7F6B: somma le ultime due, ottenendo  $40 \times \text{INT}(Y/8) + \text{INT}(X/8)$

\$7F6C - \$7F79: somma il valore \$1750 alla precedente;

\$7F7A - \$7F91: preleva il carattere all'indirizzo specificato dalla precedente, controlla che sia compreso tra «1» e «9» o che sia il carattere «#» e lo invia al buffer;

\$7F92 - \$7FA0: routine di ritardo e salto alla \$7E55, da dove si avrà il ritorno al programma della ROM.

Come potete vedere, questo programma, pur essendo già di se stesso abbastanza complesso, può certamente servire da supporto, per lo sviluppo di altre utility, abbinate all'adattatore telematico 6499. E quindi, sù, datevi da fare anche un po' voi, adesso!!!

#### Nota:

Per utilizzare il programma occorre:

- 1) Accendere il C64 con la cartuccia del 6499 inserita;
- 2) tornare al Basic del C64 per mezzo del tasto F8;
- 3) caricare dal disco il programma e farlo eseguire.

N.d.r. È disponibile in redazione il dischetto contenente anche il file-agenda dei numeri Videotel di tutta Italia.

## Trace 64

di Andrea Agostini - Bologna

Il programma, che gira sul C64, svolge la funzione di un'istruzione purtroppo assente nel Basic 2.0 del 64. Il suo compito è di visualizzare in tempo reale il numero della linea di ciascuna istruzione eseguita in un programma Basic, dopo il RUN.

La routine si attiva con SYS49178 (\$C01A) e si disattiva con SYS49189 (\$C025). Il funzionamento è semplice: alle locazioni \$308-309 si trova un vettore che punta alla routine di esecuzione delle istruzioni Basic, e tutte le volte

che deve esserne eseguita una l'interprete salta all'indirizzo contenuto in tale vettore. Volendo eseguire una procedura (in questo caso stampare un numero di linea) ogni volta che viene eseguita un'istruzione, si deve mettere nel suddetto vettore l'indirizzo iniziale della routine, in modo che anziché eseguire il comando l'interprete passi il controllo alla routine in questione. Se occorrerà anche eseguire i comandi Basic (come

in questo caso), basterà concludere la routine con un salto incondizionato ad A7E4, l'indirizzo al quale punta normalmente il vettore.

#### Analisi del disassemblato

**C000-C005:** legge il contenuto di \$3A e, se è uguale a \$FF e cioè se il comando è stato inserito in modo diretto, salta a C017.

**C006-C016:** stampa il numero di linea (contenuto in \$39-3A) racchiuso fra parentesi quadre. La routine in BDCD serve a stampare un numero compreso fra 0 e 65535 il cui byte basso sia contenuto nel registro X e il byte alto nell'accumulatore.

**C017:** in A7E4 è locata la normale routine di esecuzione delle istruzioni Basic: dopo avere stampato il numero di linea, il programma vi salta per eseguire i comandi.

**C01A-C024:** questa routine serve semplicemente ad attivare il Trace, facendo puntare il vettore in 308-309, che contiene l'indirizzo di partenza della routine di esecuzione delle istruzioni, a C000, indirizzo di partenza della routine di Trace.

**C025-C02F:** staccata anch'essa dal resto del programma, questa routine disattiva il Trace, facendo puntare il vettore di cui sopra ad A7E4, e cioè ripristinandone i valori normali.

```
10 I=49152
20 READA:IFA<0THENEND
30 POKEI,A:I=I+1:GOTO20
40 DATA165,058,201,255,240,017,166
50 DATA057,169,091,032,210,255,165
60 DATA058,032,205,189,169,093,032
70 DATA210,255,076,228,167,169,000
80 DATA141,008,003,169,192,141,009
90 DATA003,096,169,228,141,008,003
100 DATA169,167,141,009,003,096,208
110 DATA-1
```

Listato del programma Trace in formato Data Marker 64 (vedi MC n. 79, novembre 1988, pagg. 257-258).

```
*
** C000 A5 3A LDA #3A
** C002 C9 FF CMP #$FF
** C004 F0 11 BEQ #C017
** C006 A6 39 LDX #39
** C008 A9 5B LDA #$5B
** C00A 20 D2 FF JSR $FFD2
** C00D A5 3A LDA #3A
** C00F 20 CD ED JSR $BDCD
** C012 A9 5D LDA #$5D
** C014 20 D2 FF JSR $FFD2
** C017 4C E4 A7 JMP #A7E4
** C01A A9 00 LDA #00
** C01C 8D 08 03 STA #0308
** C01F A9 C0 LDA #$C0
** C021 8D 09 03 STA #0309
** C024 60 RTS
** C025 A9 E4 LDA #$E4
** C027 8D 08 03 STA #0308
** C02A A9 A7 LDA #$A7
** C02C 8D 09 03 STA #0309
** C02F 60 RTS
*
```

Disassemblato del programma Trace.



# Elenco del software disponibile su cassetta o minifloppy

Per ovviare alle difficoltà incontrate da molti lettori nella digitazione dei listati pubblicati nelle varie rubriche di software sulla rivista, MCmicrocomputer mette a disposizione i programmi più significativi direttamente su supporto magnetico. Riepiloghiamo qui sotto i programmi disponibili per le varie macchine, ricordando che i titoli non sono previsti per computer diversi da quelli indicati. Il numero della rivista su cui viene descritto ciascun programma è riportato nell'apposita colonna; consigliamo gli interessati di procurarsi i relativi numeri arretrati, eventualmente rivolgendosi al nostro Servizio Arretrati utilizzando il tagliando pubblicato in fondo alla rivista.

Per l'ordinazione inviare l'importo (a mezzo assegno, c/c o vaglia postale) alla Technimedia srl, Via Carlo Perrier 9, 00157 Roma.

Codice	Titolo Programma	MC n.	Prezzo
<b>APPLE II</b>			
DA2/06	Miniset + Levados	37	15000
DA2/07	27 programmi grafici	38	30000
DA2/08	Adventure Editor	38	15000
DA2/09	Animazione Funzioni	42	15000
DA2/12	Routine Grafiche Esiose	44	15000
DA2/13	Scroll 300 linee	46	15000
DA2/14	Assembler in Basic	50	15000
DA2/15	G-Basic II	53	15000
DA2/16	Disk Editor	54	15000
DA2/17	Latino	57	15000
DA2/18	Battaglia	61	15000
DA2/19	Catago	64	15000
DA2/20	Apple Puzzle II	65	15000
DA2/21	Precisione Multibla	68	15000
DA2/22	Sistema 2 + Tolo 5.3 IIGS	68	15000
DA2/23	Operazione Apokalypsis	71	30000
DA2/24	Classifiche di Formula I	72	15000
DA2/25	Programmabile RPN	73	15000
DA2/26	Supercircle + Polygonale	74	15000
DA2/27	Hard Copy Oki 83/A	76	15000
DA2/28	ProDOS Utility	77	15000
DA2/29	Modulo Base	78	15000
DA2/30	List db	79	15000
DA2/31	Bioritmi	80	15000
<b>COMMODORE/AMIGA</b>			
DAM/01	F-15	63	15000
DAM/02	Gest. liste programmi	64	15000
DAM/03	Studio di Funzioni	66	15000
DAM/04	Math Pack	68	15000
DAM/05	Redcode & Mars (Core Wars)	68	15000
DAM/06	Life	69	15000
DAM/07	Rubrica Telefonica	70	15000
DAM/08	Piramidi	70	15000
DAM/09	Regolazione dei colori	71	15000
DAM/10	Analitica	71	15000
DAM/11	Grafici	72	15000
DAM/12	Traduttore	73	15000
DAM/13	La Borsa	74	15000
DAM/14	DMA Music Compiler	74	15000
DAM/15	Poker	76	15000
DAM/16	Programmi per il Copper	79	15000
DAM/17	Mandelbrot mania	81	15000
<b>MS-DOS</b>			
DMS/01	Plotter + Morse	67	15000
DMS/02	Melole + Spawn	68	15000
DMS/03	Pretty + Scritte scorrevoli + Compute	69	15000
DMS/04	Emulatore CGA per Hercules	70	15000
DMS/05	Turbo Directory	71	15000
DMS/06	Math Tool S	72	15000
DMS/07	Bioritmi + Routine	72	15000
DMS/08	Salvavideo + Scritte scorrev. + PG151	73	15000
DMS/09	Optimizer + Indenter dBase III	74	15000
DMS/10	Joystick Controller	75	15000
DMS/11	BootSlow + SlowDown + Turbo Utility	75	15000
DMS/12	Redcode & Mars (Core Wars)	76	15000
DMS/13	Gestione Errori Critici Disco + PosCur	77	15000
DMS/14	Finesse & Desk	78	15000
DMS/15	General Manager	78	15000
DMS/16	Tool 05	79	15000
DMS/17	PullDown Menu + Retrace	80	15000
DMS/18	Rehe	81	15000
DMS/19	La spada di Krall	82	15000
DMS/20	Regressione	82	15000
DMS/21	Tesseract + Charset Editor	83	15000
DMS/22	Sega-File + Installatore	84	15000
<b>ATARI ST</b>			
DST/01	Virus Killer	74	15000
DST/02	Mandelbrot + Proiez. Ort. + Bilancio	78	15000
DST/03	Diagrammi di Henon	81	15000
DST/04	Paroliamo	84	15000
DST/05	Enalitto	85	15000
<b>COMMODORE 128</b>			
D28/01	MMCalc	53	15000
D28/02	Hardcopy 128	55	15000
D28/03	Sheet II	57	15000
D28/04	Star Quest	58	15000

Codice	Titolo Programma	MC n.	Prezzo
D28/05	Family Budget	60	15000
D28/06	La Casa Sirogata	61	15000
D28/07	Strutture 80/33	63	15000
D28/08	Bas 80 V. 2.0a	64	15000
D28/09	Paint 80 1.0	65	15000
D28/10	Bas 80 V. 2.11	66	15000
D28/11	Calendario Perpetuo + Montecarlo	67	15000
D28/12	Disegna Circuiti	68	15000
D28/13	Mark's Data Base	70	15000
D28/14	Label Disk + Disk Editor + Dem DCS	71	15000
D28/15	PullDown 128HR + Menu + Drawer	72	15000
D28/16	Prospettive	73	15000
D28/17	Char 80 V. 1.0	74	15000
D28/18	Italia 128	75	15000
D28/19	Super Sprite	77	15000
D28/20	Othello	80	15000
D28/21	Expert System Shell 128	81	15000
D28/22	Kit di programmazione S.O.G.A.R. 128	82	15000
D28/23	Caratteri Programmabili	83	15000
D28/24	Brush & Mouse	84	15000
<b>COMMODORE 64</b>			
D64/11	Anno Domini	57	15000
D64/12	The Disk Editor	54/6/7	15000
D64/13	Boz's Adventure	57	15000
D64/14	Link-64	57	15000
D64/15	New Char 2.2	58	15000
D64/16	Music 64	59	15000
D64/17	TRX-MEM	59	15000
D64/18	WOS + WBasic	60	15000
D64/19	Strange Basic + D'ocula	63	15000
D64/20	File Rescue	64	15000
D64/21	La Casa	64	15000
D64/22	Digital Voice	65	15000
D64/23	Vita 3D	65	15000
D64/24	Corso di Linguistica	66	15000
D64/25	Archipiplus	66	15000
D64/26	Math Pack Plus	66	15000
D64/27	Scroll + Multitask + Classifica	67	15000
D64/28	Calend. Perpetuo + Effetto Telecamera	68	15000
D64/29	Listing Plus + Utility Data	69	15000
D64/30	Trocchi e Routine per programmatori	71	15000
D64/32	Flow-Chart + Flower's Love	73	15000
D64/33	Sprite Editor	76	15000
D64/34	Portfolio 64 + Elimina. bordi schermo	77	15000
D64/35	Alfabeto Morse + Locate + Menu/Driver	78	15000
D64/36	Schedario Gare	80	15000
D64/37	Intonatore	81	15000
D64/38	Gendata 64	82	15000
D64/39	La mappa del Domino	83	15000
D64/40	Block Utility	84	15000
D64/41	Sprite	85	15000
<b>MSX</b>			
DMX/01	Tolo 13	60	15000
DMX/02	Painter	62	15000
DMX/03	MSX Bank	63	15000
DMX/04	Grafica 3D + Hard Copy	65	15000
DMX/05	Easy Disk	66	15000
DMX/06	Classifiche	67	15000
DMX/07	Magic Paint	67	15000
DMX/08	Autogest	68	15000
DMX/09	Compilatore v. 1.01	69	15000
DMX/10	Diskmap	70	15000
DMX/11	Mini dBase MSX	71	15000
DMX/12	Grafica in Turbo Pascal	72	15000
DMX/13	Math Pack Plus 3.20	73	15000
DMX/14	RS2CAD	75	15000
DMX/15	Simple Desk	76	15000
DMX/16	The MSX2 Super Print	77	15000
DMX/17	Grafica in Turbo Pascal (Graph 1&2)	77	15000
DMX/18	Hard Copy	78	15000
DMX/19	HEXCOMP	79	15000
DMX/20	Utilities in Turbo Pascal	80	15000
DMX/21	dBase MSX Plus	81	15000
DMX/22	Turbo Pascal Turtle Graphics	82	15000
DMX/23	PurChar + 4010 S	84	15000
DMX/24	Analysis + Lister Basic	85	15000

Nota:  
l'iniziale del codice è C per le cassette, D per i floppy.

# FLOPPERIA

SRL

Viale Monte Nero, 31  
20135 Milano

Tel. (02) 55.18.04.84 ric. aut.  
Fax (02) 55.18.81.05 (24 ore)

Abbiamo tutto ciò che vi serve.