

## PIKPOK

di Roberto Marassi - Pistoia

PIKPOK è un «monitor istantaneo», in linguaggio macchina, che vi consentirà di leggere in tempo reale tutti i registri di memoria del C-64, e di modificarli con la semplice pressione di un tasto. È formato da due routine:

[PIK] (allocata nei registri 830-922). Effettua la lettura e il display di un'intera pagina di memoria (256 byte) in testa allo schermo. L'inserimento di questa routine nell'interrupt IRQ, con aggiornamento del display 60 volte al secondo, avrebbe comportato un eccessivo rallentamento delle altre funzioni del computer (compresa la stessa routine [POK]). Ho preferito quindi ricorrere all'interrupt non mascherabile (NMI), regolando il timer del CIA2 in modo da generare

un interrupt 10 volte al secondo, frequenza più che sufficiente per un display in tempo reale. Naturalmente, dato che i timer del CIA2 vengono usati anche per la temporizzazione delle operazioni di input/output, sarà necessario disattivare PIKPOK con SHIFT-RETURN (v. dopo) o con SYS 836 in modo diretto, prima di eseguire operazioni di lettura/scrittura (disco, registratore, stampante ecc.).

[POK] (allocata nei registri 50694-58471). Cambia la pagina di memoria sullo schermo, visualizza in decimale o esadecimale i valori dei registri, del loro contenuto e degli indirizzi a doppio byte, e ne permette la modifica istantanea.

Digitate il programma, salvatelo su supporto magnetico e date il RUN. Se i DATA sono stati introdotti correttamente, premendo un tasto dopo la schermata iniziale avrete sullo schermo la pagina

«zero» (se non dovesse comparire, premete RUN-STOP/RESTORE, digitate SYS 836 e ridate il RUN). Potete osservare come alcuni registri si modificano continuamente (sono quelli dell'orologio interno TI del C-64), altri si modificano in seguito alla pressione dei tasti, ecc. Si può dedurre il valore dei singoli byte dal loro «codice schermo», tenendo presente che il display è nel modo maiuscolo/minuscolo: così, ad esempio, una «h» minuscola corrisponde al valore 8, una «H» maiuscola al valore 72, e gli stessi caratteri in «reverse» ai valori 136 e 200 rispettivamente.

Vediamo ora le possibilità offerte dalla routine [POK] che è stata attivata:

Cambio pagina - Il tasto-funzione [f7] aumenta di 1 il numero della pagina visualizzata; lo stesso tasto shiftato ([f8]) lo diminuisce di 1.

Il puntatore - Il carattere visualizzato

```

100 rem" ** pikpok - by r.morassi **
110 :
120 poke53281,6
130 print"(clr)(swlc)(wht) ** PIKPOK Instant Monitor **"
140 print"(down)(cyn) (rvs)Routine [PIK](off) - Visualizza, in tempo"
150 print" reale, il contenuto di una pagina di"
160 print" memoria, sotto forma di codici-schermo"
170 print" (nel set minuscolo)."
180 print" (rvs)Routine [POK](off) -Impostando un carattere"
190 print" sulla tastiera, il suo codice-schermo"
200 print" viene trasferito nel registro sotto il"
210 print" puntatore ((blk)nero)(cyn). I tasti CURSOR spo-"
220 print" stano il puntatore. (rvs)F1/F2(off) Stampa (#dec
230 print" /$hex) address 16 bit (l.b. puntato) -
240 print" (rvs)F3/F4(off) Stampa (#dec/$hex) valore byte -
250 print" (rvs)F5/F6(off) Stampa (#dec/$hex) n. registro -
260 print" (rvs)F7/F8(off) Aumenta/diminuisce n. pagina -
270 print" Disabilitazioni: (rvs)RETURN(off) Solo [POK]
280 print" (rvs)SHIFT/RETURN(off) [PIK] + [POK]
290 print"(down)(wht) ** Comandi in modo diretto **"
300 print" (rvs)(cyn)SYS830:POKEB28,N(off) Abilita [PIK] a pag. N"
310 print"(rvs)SYS945,N(off) Abilita [PIK]+[POK] a pagina N"
320 print"(rvs)SYS954(off) Abilita [POK] se PIK gia' acceso"
330 print"(rvs)SYS836(off) Disabilita PIK"
340 print"(rvs)POKE1,53 (55)(off) Legge S.O. in RAM (ROM)"
350 print$pc(8) (yel)Attendere, prego :(up)"
360 gosub590
370 :
380 printchr$(18+128*((peek(162)/16)and1))$pc(9)*** Premi un tasto ***(up)"
390 geta$:ifa$=""then380
400 :
410 print"(clr)(wht)"tab(255)spc(115)"Tasti di controllo [POK]"
420 print" (F1/F2) Address (dec/hex) ↵
430 print" (F3/F4) Valore byte (dec/hex)+→
440 print" (F5/F6) N. registro (dec/hex) |
450 print" (F7/F8) N. pagina (decimale) ↓
460 print" F7/F8 Aumenta/diminuisce n. pagina"
470 print" Tasti CURSOR: spostano il puntatore"
480 print" Disabilita: (rvs)RETURN(off) Solo [POK] -
490 print" (rvs)SHIFT/RETURN(off) [PIK] + [POK]
500 print"(down)(wht) ** Comandi in modo diretto **"
510 print" (rvs)(cyn)SYS830:POKEB28,N(off) Abilita [PIK] a pag. N"
520 print"(rvs)SYS945,N(off) Abilita [PIK]+[POK] a pagina N"
530 print"(rvs)SYS954(off) Abilita [POK] se PIK gia' acceso"
540 print"(rvs)SYS836(off) Disabilita PIK"
550 print"(rvs)POKE1,53 (55)(off) Legge S.O. in RAM (ROM) (home) (yel)"
560 poke650,128:sys51170:sys830:sys50859
570 end
580 :
590 ifpeek(50763)=80andpeek(50764)=79andpeek(50765)=75thenreturn
600 a=0:forx=0to149:ready:a=a+y:poke830+x,y:next
610 ifa<>17360thenprint"errore nel primo blocco data":stop
620 a=0:forx=0to498:ready:a=a+y:poke50694+x,y:next
630 ifa<>58471thenprint"errore nel secondo blocco data":stop
640 return
650 :
660 rem * modifica vettori interrupt *
670 :

```

```

680 data 120,032,021,198,088,096,120,032
690 data 006,198,169,071,141,024,003,169
700 data 254,141,025,003,169,000,141,138
710 data 002,088,076,129,255
720 :
730 rem" ** routine [pik] **
740 :
750 data 072,138,072
760 data 152,072,173,060,003,133,252,169
770 data 000,133,251,133,253,170,168,177
780 data 251,157,000,004,236,052,003,208
790 data 024,201,032,240,004,201,096,208
800 data 005,009,128,157,000,004,169,000
810 data 157,000,216,165,253,240,011,208
820 data 020,165,253,208,011,169,007,157
830 data 000,216,169,010,133,253,208,005
840 data 169,003,157,000,216,169,022,141
850 data 024,208,198,253,232,200,208,191
860 data 076,081,254,032,062,003,032,241
870 data 183,142,060,003,169,000,141,054
880 data 003,169,024,141,053,003,169,128
890 data 141,138,002,024,162,024,160,000
900 data 032,240,255,076,181,198
910 :
920 rem" ** resetta timer NMI **
930 :
940 data 169,127,141,013,221,169,008,141
950 data 014,221,141,015,221,096,000
960 :
970 rem" ** inserisce wedge NMI **
980 :
990 data 162,190,142,005,221,162,255,142,004
1000 data 221,232,142,007,221,142,060,003
1010 data 169,001,141,006,221,169,017,141
1020 data 014,221,169,081,141,015,221,169
1030 data 127,141,013,221,169,130,141,013
1040 data 221,169,091,141,024,003,169,003
1050 data 141,025,003,096,000
1060 :
1070 rem" ** password p-o-k **
1080 :
1090 data 080,079,075
1100 :
1110 rem" ** routine [pok] **
1120 :
1130 data 160,000,174,052,003,134,254,174
1140 data 060,003,134,255,177,254,141,055
1150 data 003,096,000,000,165,254,170,041
1160 data 015,032,159,198,141,057,003,138
1170 data 032,155,198,141,056,003,165,255
1180 data 170,041,015,032,159,198,141,059
1190 data 003,138,032,155,198,141,058,003
1200 data 169,036,032,210,255,162,000,189
1210 data 056,003,032,210,255,232,224,004
1220 data 208,245,076,200,198,074,074,074
1230 data 074,201,010,144,004,024,105,087
1240 data 096,024,105,048,096,169,011,141

```



Tasto	Normale	Reverse	Tasto	Normale	Reverse	Tasto	Normale	Reverse	Tasto	Normale	Reverse
@	0	128	SPC	32	160	SHFT *	64	192	SHFT SPC	96	224
A	1	129	! (SHFT 1)	33	161	SHFT A	65	193	CMDR K	97	225
B	2	130	" (SHFT 2)	34	162	SHFT B	66	194	CMDR I	98	226
C	3	131	# (SHFT 3)	35	163	SHFT C	67	195	CMDR T	99	227
D	4	132	\$ (SHFT 4)	36	164	SHFT D	68	196	CMDR @	100	228
E	5	133	% (SHFT 5)	37	165	SHFT E	69	197	CMDR G	101	229
F	6	134	& (SHFT 6)	38	166	SHFT F	70	198	CMDR +	102	230
G	7	135	' (SHFT 7)	39	167	SHFT G	71	199	CMDR M	103	231
H	8	136	( (SHFT 8)	40	168	SHFT H	72	200	CMDR £	104	232
I	9	137	) (SHFT 9)	41	169	SHFT I	73	201	SHFT £	105	233
J	10	138	*	42	170	SHFT J	74	202	CMDR N	106	234
K	11	139	+	43	171	SHFT K	75	203	CMDR Q	107	235
L	12	140	,	44	172	SHFT L	76	204	CMDR D	108	236
M	13	141	-	45	173	SHFT M	77	205	CMDR Z	109	237
N	14	142	.	46	174	SHFT N	78	206	CMDR S	110	238
O	15	143	/	47	175	SHFT O	79	207	CMDR P	111	239
P	16	144	0	48	176	SHFT P	80	208	CMDR A	112	240
Q	17	145	1	49	177	SHFT Q	81	209	CMDR E	113	241
R	18	146	2	50	178	SHFT R	82	210	CMDR R	114	242
S	19	147	3	51	179	SHFT S	83	211	CMDR W	115	243
T	20	148	4	52	180	SHFT T	84	212	CMDR H	116	244
U	21	149	5	53	181	SHFT U	85	213	CMDR J	117	245
V	22	150	6	54	182	SHFT V	86	214	CMDR L	118	246
W	23	151	7	55	183	SHFT W	87	215	CMDR Y	119	247
X	24	152	8	56	184	SHFT X	88	216	CMDR U	120	248
Y	25	153	9	57	185	SHFT Y	89	217	CMDR O	121	249
Z	26	154	:	58	186	SHFT Z	90	218	SHFT @	122	250
[ (SHFT :)	27	155	;	59	187	SHFT +	91	219	CMDR F	123	251
£	28	156	< (SHFT .)	60	188	CMDR -	92	220	CMDR C	124	252
l (SHFT ;)	29	157	=	61	189	SHFT -	93	221	CMDR X	125	253
l	30	158	> (SHFT .)	62	190	SHFT †	94	222	CMDR V	126	254
†	31	159	? (SHFT /)	63	191	CMDR *	95	223	CMDR B	127	255

Tabella di consultazione per sapere la combinazione dei tasti da usare per modificare i valori in memoria con PIKPOK.

in nero, inizialmente nell'angolo in alto a sinistra, è il «puntatore» usato da [POK]. Lo si può spostare lungo la pagi-

```

1250 data 053,003,169,032,141,054,003,162
1260 data 255,142,052,003,174,060,003,142
1270 data 055,003,169,000,174,055,003,032
1280 data 205,189,162,004,169,032,032,210
1290 data 255,202,208,248,024,174,053,003
1300 data 172,054,003,032,240,255,238,052
1310 data 003,032,228,255,240,251,048,082
1320 data 201,032,144,033,201,096,144,004
1330 data 041,223,208,002,041,063,166,199
1340 data 240,002,009,128,160,000,174,052
1350 data 003,134,254,174,060,003,134,255
1360 data 145,254,076,108,199,201,017,208
1370 data 012,173,052,003,024,105,040,141
1380 data 052,003,076,223,198,201,018,208
1390 data 002,133,199,201,013,208,009,169
1400 data 000,141,052,003,076,084,003,234
1410 data 201,029,208,173,238,052,003,076
1420 data 223,198,041,127,201,127,208,002
1430 data 169,094,201,032,144,005,009,064
1440 data 076,244,198,201,017,208,012,173
1450 data 052,003,056,233,040,141,052,003
1460 data 076,223,198,201,029,208,003,206
1470 data 052,003,201,018,208,004,162,000
1480 data 134,199,201,006,208,006,032,078
1490 data 198,076,192,198,201,005,208,009
1500 data 032,078,198,200,177,254,076,194
1510 data 198,201,007,208,009,173,060,003
1520 data 174,052,003,076,197,198,201,010
1530 data 208,021,160,000,173,052,003,133
1540 data 254,173,060,003,133,255,177,254
1550 data 133,255,132,254,076,098,198,201
1560 data 011,208,009,173,052,003,172,060
1570 data 003,076,158,199,201,009,208,013
1580 data 032,078,198,200,177,254,168,173
1590 data 055,003,076,158,199,201,008,208
1600 data 006,238,060,003,076,181,198,201
1610 data 012,208,006,206,060,003,076,181
1620 data 198,201,013,208,003,076,068,003
1630 data 076,223,198,000
1640 :
1650 rem" ** copia s.o. da ROM in RAM **
1660 :
1670 data 120,169,160,133,253
1680 data 169,000,133,252,168,177,252,145
1690 data 252,200,208,249,230,253,208,245
1700 data 088,096
ready.

```

È disponibile, presso la redazione, il disco con il programma presentato in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 279.

na di memoria usando i tasti-cursore.

Lettura di un indirizzo - Posizionando il puntatore sul low-byte di un indirizzo a 16 bit, e premendo [f1], l'indirizzo viene visualizzato in forma decimale. Questo equivale all'istruzione Basic:

```
PRINT (PEEK(x)+256*PEEK(x+1))
```

Esempio: i registri 43-44 puntano alla locazione di inizio del programma Basic. Portando il puntatore sul registro 43 e premendo [f1] si ottiene il valore 2049, cioè (1+256\*8). La stessa operazione sui registri 45-46 vi darà l'indirizzo a cui termina il programma Basic.

Valore del byte - Il valore del byte sotto il puntatore viene visualizzato premendo [f3].

Numero del registro - Il numero del registro sotto il puntatore viene visualizzato premendo [f5].

Esadecimale - Tutti i valori (indirizzo, byte, registro) verranno visualizzati in forma esadecimale anziché decimale, premendo i rispettivi tasti shiftati (cioè [f2], [f4], e [f6] rispettivamente).

Modifica dei valori in memoria - Se si preme un carattere stampabile, eventualmente preceduto da «rvs-on» o «rvs-off» se necessario, il codice-schermo dello stesso viene trasferito nel registro che si trova in quel momento sotto il puntatore. Se si tratta di una pagina di ROM, il valore verrà trasferito nella RAM sottostante. Esempio: spostatevi alla pagina 8 dove inizia il programma Basic, e modificate alcuni byte. Il programma verrà effettivamente modificato, come potrete verificare con un ritorno al modo diretto e successivo LIST (v. dopo).

Uscita dalla routine [POK] - Premendo il tasto RETURN viene disattivata la sola routine [POK], e cancellato lo schermo: la routine [PIK] è sempre attiva, e continua a visualizzare la pagina di memoria

in tempo reale. Riportate il cursore in vista, e provate a dare delle istruzioni Basic per vederne l'effetto sui vari registri di memoria visualizzati da [PIK]. Allo stesso modo potete vedere come vengono gestite le variabili, le stringhe, le matrici ecc.

Uscita completa dalla routine [PIK-POK] - Premendo SHIFTRETURN anziché RETURN, entrambe le routine [PIK]+[POK] vengono disattivate, lo schermo si cancella, e i timer del CIA2 vengono fermati e resettati.

Una volta tornati in modo diretto si può dare il NEW della parte Basic del programma e scrivere o caricare da un supporto magnetico un altro programma (in quest'ultimo caso, come detto, è necessario resettare i timer con SYS 836). [PIKPOK] può essere riattivato in qualunque momento con i seguenti comandi:

se [PIK] è disattivato:

SYS 830: riattiva [PIK] a pagina zero; può essere seguito da POKE 828,N (N=-255) che cambia il numero N della pagina visualizzata.

SYS 945,N: attiva [PIK]+[POK] a pagina N, se [PIK] è attivato:

SYS 954: attiva [POK] senza cambiare il numero della pagina. Notate che i numeri visualizzati da [POK] compaiono ora nell'angolo in basso a sinistra.

SYS 945,N: attiva [POK] a pagina N.

SYS 836: disattiva [PIK], resetta i timer del CIA2 e cancella lo schermo.

### Modifiche del sistema operativo (Interpreter + Kernal)

Le pagine 160-191 (\$A000-\$BFFF) e 224-255 (\$E000-\$FFFF) contengono, rispettivamente, l'Interpreter e il Kernal del sistema operativo. Essendo in ROM, possono essere lette da [PIK] ma



non modificate. Tuttavia, all'inizio del programma, l'intero S.O. viene copiato nella RAM sottostante e questo ne consente la modifica purché si ordini al computer di leggere le relative pagine in RAM anziché in ROM. In modo diretto, ciò si ottiene con POKE 1,53. Usando invece [PIKPOK] a pag. zero, si sposta il puntatore sul registro 1 che mostra il carattere «7» (valore byte = 55) e lo si sostituisce col carattere «5» (valore byte = 53). Torniamo ora in modo diretto con RETURN, e richiamiamo la pagina 163 con SYS 945,163. Questa contiene una parte dei messaggi Basic che essendo ora in RAM, si possono modificare. Ad esempio, la scritta READY si può sostituire con una altra di 6 caratteri, come PRONTO (attenzione: i caratteri alfabetici vanno introdotti in maiuscolo, cioè premendo SHIFT!). Tornando in modo diretto, il messaggio sarà PRONTO e non più READY. Nello stesso modo si possono sostituire le parole chiave Basic (pagg. 160-161) con altre della stessa lunghezza, ricordando di scrivere in «reverse» l'ultimo carattere: il S.O. accetterà le nuove parole e darà ?SYNTAX ERROR su quelle vecchie. I più esperti potranno naturalmente modificare il S.O. in maniera più sostanziale, ad esempio sostituendo i comandi Basic con altri personalizzati, previa modifica della tabella degli indirizzi, ecc.

Premendo RUN-STOP+RESTORE, il registro 1 viene resettato al valore di 55 e quindi il S.O. viene letto nuovamente nella ROM. Per mantenere il S.O. in RAM, con le vostre eventuali modifiche, digitate (finché è in RAM) il seguente comando: POKE 64982,53. A questo punto, anche il RUN-STOP+RESTORE non avrà effetto sulla configurazione di memoria, e per rileggere nuovamente in ROM dovrete modificare direttamente il registro 1 con POKE 1,55.

Riportiamo in queste pagine il disassemblato, commentato, di entrambe le routine. La tabella di consultazione potrà servire per sapere subito la combinazione di tasti da usare per modificare i valori in memoria con PIKPOK.

(Ndr) In alcuni casi, per un uso più agevole del programma, conviene sostituire la prima istruzione della linea 560 con POKE 650,127. Infatti, in certi momenti il cursore è poco controllabile e diventa difficile posizionarlo sul carattere desiderato.

Ancora, ogni volta che si tocca un tasto funzione il cursore si sposta nella locazione successiva quindi, come forse è poco chiaro dall'articolo, se vogliamo ad esempio conoscere il contenuto del byte N, prima di premere F3 dobbiamo posizionarci sul byte N-1.

```

Routine [PIK] disassemblata
----- (SYS 830) -----
..033e 78 sei jrj $c615 ; disabilita IRQ
..033f 20 15 c6 jrj $c615 ; subroutine inserzione wedge NMI
..0342 58 cli ; riabilita IRQ
..0343 60 rts ; esce
----- (SYS 836) -----
..0344 78 sei jrj $c606 ; disabilita IRQ
..0345 20 06 c6 jrj $c606 ; subroutine reset registri CIA2
..0348 a9 47 lda #$47 ; ripristina
..034a 8d 18 03 sta $0318 ; il puntatore
..034d a9 fe lda $ffe ; di NMI
..034f 8d 19 03 sta $0319 ; a $fe47
..0352 a9 00 lda #00 ; azzerà il flag $02a8
..0354 8d 8a 02 sta $02a8 ; (annulla ripetizione tasti)
..0357 58 cli ; riabilita IRQ
..0358 4c 81 ff jmp $ff81 ; inzializza lo schermo ed esce
----- (inizio wedge) -----
..035b 48 pha ; salva
..035c 8a txa ; sullo stack
..035d 48 pha ; i registri
..035e 98 tya ; A,X,Y
..035f 48 pha ; del 6502
..0360 ad 3c 03 lda $033c ; copia il n. pagina
..0363 85 fc sta $fc ; in $fc
..0365 a9 00 lda #00 ; azzerà
..0367 85 fb sta $fb ; e $fd
..0369 85 fd sta $fd ; ed i registri
..036b aa tax ; X ed Y
..036c a8 tay ; legge byte puntato da $fb-fc
..036d b1 fb lda ($fb),y ; lo copia nello schermo
..036f 9d 00 04 sta $0400,x ; X e <> dal valore in $0334 ?
..0372 ec 34 03 cpx $0334 ; si' : salta a controllo contatore
..0375 d0 18 bne $038f ; no - A e' uguale a "spazio" ?
..0377 c9 20 cmp #20 ; si' : salta
..0379 f9 04 beq $037f ; no - e' uguale a "spazio shiftato" ?
..037b c9 60 cmp #60 ; no - salta
..037d 00 05 bne $0384 ; si' : mette in "reverse"
..037f d0 80 ora #80 ; e copia nello schermo
..0381 9d 00 04 sta $0400,x ;
..0384 a9 00 lda #00 ; colora in nero
----- (SYS 830) -----
..c644 a9 03 lda #03 ; (lo-byte
..c646 8d 19 03 sta $0319 ; e hi-byte)
..c649 60 rts ; esce
..c64a 00 brk ;
----- (SYS 830) -----
..c64b 50 4f 4b ; stringa password P-O-K
----- (SYS 830) -----
..c64e a0 00 ldy #00 ; copia contatori
..c650 ae 34 03 ldx $0334 ; da $0334 (pagina)
..c653 86 fe stx $fe ; a $fe
..c655 ae 3c 03 ldx $033c ; e da $033c (posizione)
..c658 86 ff stx $ff ; a $ff
..c65a b1 fe lda ($fe),y ; legge il byte puntato da $fe-ff
..c65c 8d 37 03 sta $0337 ; in $0337
..c65f 60 rts ; esce
..c660 00 brk ;
..c661 00 brk ;
----- (SYS 830) -----
..c662 a5 fe lda $fe ; carica valore da $fe (lo-byte)
..c664 aa tax ; in A ed in X
..c665 29 0f and #0f ; isola lo-nibble
..c667 20 9f c6 jsr $c69f ; converte in ASCII esadecimale
..c66a 8d 39 03 sta $0339 ; deposita in $0339
..c66d 8a txa ; recupera valore
..c66e 20 9b c6 jsr $c69b ; divide per 16 e converte in ASCII
..c671 8d 38 03 sta $0338 ; deposita in $0338
..c674 a5 ff lda $ff ; carica valore da $ff (hi-byte)
..c676 aa tax ; in A ed in X
..c677 29 0f and #0f ; isola lo-nibble
..c679 20 9f c6 jsr $c69f ; converte in ASCII esadecimale
..c67c 8d 3b 03 sta $033b ; deposita in $033b
..c67f 8a txa ; recupera valore
..c680 20 9b c6 jsr $c69b ; divide per 16 e converte in ASCII
..c683 8d 3a 03 sta $033a ; deposita in $033a
..c686 a9 24 lda #$24 ; carica codice ASCII di "$"
..c688 20 d2 ff jsr $ffd2 ; lo stampa sullo schermo
..c68b a2 00 ldx #00 ; stampa in successione
..c68d bd 38 03 lda $0338,x ; i quattro valori
..c690 20 d2 ff jsr $ffd2 ; contenuti
..c693 e8 inx ; in $0338-033b
..c694 e0 04 cpx #04 ;
..c696 d0 f5 bne $c68d ;
..c698 4c c8 c6 jmp $c6c8 ; prosegue con stampa spazi

```

```

Routine [PIK] disassemblata
----- (SYS 830) -----
..033e 78 sei jrj $c615 ; disabilita IRQ
..033f 20 15 c6 jrj $c615 ; subroutine inserzione wedge NMI
..0342 58 cli ; riabilita IRQ
..0343 60 rts ; esce
----- (SYS 836) -----
..0344 78 sei jrj $c606 ; disabilita IRQ
..0345 20 06 c6 jrj $c606 ; subroutine reset registri CIA2
..0348 a9 47 lda #$47 ; ripristina
..034a 8d 18 03 sta $0318 ; il puntatore
..034d a9 fe lda $ffe ; di NMI
..034f 8d 19 03 sta $0319 ; a $fe47
..0352 a9 00 lda #00 ; azzerà il flag $02a8
..0354 8d 8a 02 sta $02a8 ; (annulla ripetizione tasti)
..0357 58 cli ; riabilita IRQ
..0358 4c 81 ff jmp $ff81 ; inzializza lo schermo ed esce
----- (inizio wedge) -----
..035b 48 pha ; salva
..035c 8a txa ; sullo stack
..035d 48 pha ; i registri
..035e 98 tya ; A,X,Y
..035f 48 pha ; del 6502
..0360 ad 3c 03 lda $033c ; copia il n. pagina
..0363 85 fc sta $fc ; in $fc
..0365 a9 00 lda #00 ; azzerà
..0367 85 fb sta $fb ; e $fd
..0369 85 fd sta $fd ; ed i registri
..036b aa tax ; X ed Y
..036c a8 tay ; legge byte puntato da $fb-fc
..036d b1 fb lda ($fb),y ; lo copia nello schermo
..036f 9d 00 04 sta $0400,x ; X e <> dal valore in $0334 ?
..0372 ec 34 03 cpx $0334 ; si' : salta a controllo contatore
..0375 d0 18 bne $038f ; no - A e' uguale a "spazio" ?
..0377 c9 20 cmp #20 ; si' : salta
..0379 f9 04 beq $037f ; no - e' uguale a "spazio shiftato" ?
..037b c9 60 cmp #60 ; no - salta
..037d 00 05 bne $0384 ; si' : mette in "reverse"
..037f d0 80 ora #80 ; e copia nello schermo
..0381 9d 00 04 sta $0400,x ;
..0384 a9 00 lda #00 ; colora in nero

```





(segue da pag. 277)

```

..c727 8d 34 03 sta $0334      ; posizione del puntatore
..c72a 4c 54 03 jmp $0354      ; ed esce dalla routine POK
..c72d ea nop                    ;
..c72e c9 1d cmp #81d           ;
..c730 d0 ad bne $c6df          ;
..c732 ee 34 03 inc $0334      ; si: incrementa di 1 il puntatore
..c735 4c df c6 jmp $c6df      ; e torna in attesa
-----
..c738 29 7f and #87f         ; (valore > 128) - resetta bit 7
..c73a c9 7f cmp #87f         ;
..c73c d0 02 bne $c740         ; e: #127? (in origine: #255)
..c73e a9 5e lda #85e          ; si: carica # 94
..c740 c9 20 cmp #820         ; e: < #32?
..c742 09 05 bcc $c749         ; si: prosegue controllo
..c744 09 40 ora #840          ; no: setta bit 6
..c746 4c f4 c6 jmp $c6f4      ; e prosegue da $c6f4
..c749 c9 11 cmp #811         ;
..c74b d0 0c bne $c759         ; no: prosegue controllo
..c74d ad 34 03 lda $0334      ; si:
..c750 38 sec                    ; decrementa
..c751 e9 28 sbc #828          ; di 40
..c753 8d 34 03 sta $0334      ; la posizione del puntatore
..c756 4c df c6 jmp $c6df      ; e torna in attesa
..c759 c9 1d cmp #81d         ; e: "cursor left"?
..c75b d0 03 bne $c760         ; no: prosegue controllo
..c75d ce 34 03 dec $0334      ; si: decrementa il puntatore di 1
..c760 c9 12 cmp #812         ; e: "reverse off"?
..c762 d0 04 bne $c768         ; no: prosegue controllo
..c764 a2 00 ldx #800          ; si: azzerà
..c766 86 c7 stx $c7           ; il flag "reverse"
..c768 c9 06 cmp #806         ; e: "F3"?
..c76a d0 06 bne $c772         ; no: prosegue controllo
..c76c 20 4e c6 jsr $c64e      ; si: copia valore byte in $0337
..c76f 4c c0 c6 jmp $c6c0      ; e lo stampa in decimale
..c772 c9 05 bne $c77f         ; e: "F1"?
..c774 d0 09 bne $c77f         ; no: prosegue controllo
..c776 20 4e c6 jsr $c64e      ; si: copia valore byte in $0337
..c779 c8 iny                    ; carica in A
..c77a b1 fe lda ($fe),y        ; il valore successivo (hi-byte)
..c77c 4c c2 c6 jmp $c6c2      ; e stampa address in decimale
..c77f c9 07 cmp #807         ; e: "F5"?
..c781 d0 09 bne $c78c         ; no: prosegue controllo
..c783 ad 3c 03 lda $033c      ; si: carica n. pagina in A (hi-byte)
..c786 ae 34 03 ldx $0334      ; e posizione puntatore in X (lo-byte)
..c789 4c c5 c6 jmp $c6c5      ; stampa valore registro (decimale)
..c78c c9 0a cmp #80a         ; e: "F4"?
..c78e d0 15 bne $c7a5         ; no: prosegue controllo
..c790 a0 00 ldy #800          ; si: carica n. pagina
..c792 ad 34 03 lda $0334      ; in $fe
..c795 85 fe sta $fe          ; e posizione puntatore
..c797 ad 3c 03 lda $033c      ; in $ff
..c79a 85 ff sta $ff          ; carica valore byte
..c79e 85 ff sta $ff          ; lo copia in $ff
..c7a0 84 fe sty $fe          ; azzerà $fe
..c7a2 4c 62 c6 jmp $c662      ; stampa valore in esadecimale
;
; e' "F6"?
; no: prosegue controllo
; si: carica n. pagina in A
; e posizione puntatore in X
; stampa registro in esadecimale
; e "F2"?
; no: prosegue controllo
; si: copia byte in $0337
; legge
; il byte successivo
; copia in Y
; copia valore byte in A
; stampa address in esadecimale
; e "F7"?
; no: prosegue controllo
; si: incrementa n. pagina di 1
; e stampa nuovo valore
; e "F8"?
; no: prosegue controllo
; si: decrementa n. pagina di 1
; e stampa nuovo valore
; e "shift-return"?
; no: torna in attesa
; si: disattiva il wedge ed esce
; (se no, torna in attesa)
;
;----- copia s.o. da ROM in RAM (SYS 51170)
;
; disabilita IRQ
; carica l'indirizzo
; $a000
; in $fc (lo-byte)
; e $fd (hi-byte)
;
; legge byte da ROM
; e lo copia nella RAM sottostante
; aumenta indice
; ripete per 256 volte
; aumenta hi-byte
; inc $fd
; ripete fino a $ffff
; riabilita IRQ
; ed esce
;
;-----
..c7e2 78 sei                    ;
..c7e3 a9 a0 lda #8a0          ;
..c7e5 85 fd sta $fd          ;
..c7e7 a9 00 lda #800          ;
..c7e9 85 fc sta $fc          ;
..c7eb a8 tay                    ;
..c7ec b1 fc lda ($fc),y        ;
..c7ee 91 fc sta ($fc),y        ; e lo copia nella RAM sottostante
..c7f0 c8 iny                    ;
..c7f1 d0 f9 bne $c7f3         ; ripete per 256 volte
..c7f3 e6 fd inc $fd          ;
..c7f5 d0 f5 bne $c7f3         ;
..c7f7 58 cli                    ;
..c7f8 60 rts                    ;

```





# 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>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	78	15000
DAM/16	Programmi per il Copper	79	15000
DAM/17	Mandelbrot mania	81	15000
DAM/18	SF-Search File	86	15000
DAM/19	AMiprac	87	15000
DAM/20	Click	93	15000
DAM/21	Editor di Videogame	95	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 5	72	15000
DMS/07	Bortini + 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	76	15000
DMS/12	Redcode & Mars (Core Wars)	76	15000
DMS/13	Gestione Errori Critici Disco + PosCur	77	15000
DMS/14	Finestre & Desk	78	15000
DMS/15	General Manager	78	15000
DMS/16	Tool 05	79	15000
DMS/17	Pulldown Menu + Retrace	80	15000
DMS/18	Righe	81	15000
DMS/19	La spada di Krail	82	15000
DMS/20	Regression	82	15000
DMS/21	Teasparc + Charpsit Editor	83	15000
DMS/22	Sega-File + Installatore	84	15000
DMS/23	Shape Tool.kit	87	15000
DMS/24	Starfox	88	15000
DMS/25	Inline in Turbo C + Svuota Keyboard Buffer	89	15000
DMS/26	Sistemi parametrici + GIFDIR 1.0	90	15000
DMS/27	QuickMatrix + Stampa Virtuale	91	15000
DMS/28	Whereis + Bold.C + Setmenu	92	15000
DMS/29	Bobo + MGA 100	93	15000
DMS/30	War+LS (2 dischi)	94	20000
DMS/31	Warbot Arona	95	15000
DMS/32	Cmouse + Sp-Build + Tratt.Lsp	96	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	Paroliario	84	15000
DST/05	Enalotto	85	15000
DST/06	Cariche	83	15000
DST/07	Flow-Chart	87	15000
DST/08	Funz Manual	89	15000
DST/09	RedCode	88	15000
DST/10	Pattern Generator 2.1	92	15000
DST/11	ST Hopalong + Moto casuale	93	15000
DST/12	Speezy Mail	94	15000
DST/13	Mead	94	15000
DST/14	Designer	95	15000
DST/15	Pro-Function 1.0	96	15000
<b>COMMODORE 128</b>			
D28/01	MMCalc	53	15000
D28/02	Hardcopy 128	55	15000
D28/03	Sheet It	57	15000
D28/04	Star Quest	58	15000
D28/05	Family Budget	60	15000
D28/06	La Casa Stregata	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 DOS	71	15000

Codice	Titolo Programma	MC n.	Prezzo
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
D28/25	Astrolabio	89	15000
D28/26	Sort Vettore Stringa + Input Stringa	90	15000
D28/27	Piano Cartesiano + Calendario	91	15000
D28/28	Bingol 90	92	15000
D28/29	Kaala View + Clock 128 + File Data Maker	93	15000
D28/30	Curve Polar	94	15000
D28/31	Biblio 128	95	15000
<b>COMMODORE 64</b>			
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 + Dracula	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	Archipius	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	Linko Plus + Utility Data	69	15000
D64/31	Trucchi e Routine per programmatori	71	15000
D64/32	Flow-Chart + Flower's Love	73	15000
D64/33	Sprite Editor	76	15000
D64/34	Portfollio 64 + Elim. bordi schermo	77	15000
D64/37	Intonatore	81	15000
D64/38	Giandata 64	82	15000
D64/40	Block Utility	84	15000
D64/41	Sprite 6499	85	15000
D64/42	Mot Mot Labyrinth + Tool 64	86	15000
D64/43	Titolatore 64	88	15000
D64/44	Editor Musicale	89	15000
D64/45	Dump Variabili	90	15000
D64/46	FBasic V1.0 + Tod-Basic	91	15000
D64/47	Le sedici carte + Stampa Directory	92	15000
D64/48	REL-Arch 64 Statistica 3.0	94	15000
D64/49	MegaGame 64	95	15000
D64/50	PkPok	96	15000
<b>MSX</b>			
DMX/01	Toto 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	RGB CAD	75	15000
DMX/15	Simple Desk	76	15000
DMX/16	The MSX2 Super Print	77	15000
DMX/17	Grafica in Turbo Pascal	77	15000
DMX/18	Hard Copy	78	15000
DMX/19	HEXDUMP	79	15000
DMX/20	Utilities in Turbo Pascal	80	15000
DMX/21	dBase MSX Plus	81	15000
DMX/23	PutChar + 4010 S	84	15000
DMX/24	Analysis + Lister Basic	85	15000
DMX/25	Alliment MSX	86	15000
DMX/26	SortDir + Octogonius + 3D Mover	87	15000
DMX/27	Tool/Extended Basic + Bindata.Bas	88	15000
DMX/28	Chain & Window + Chain Utility	90	15000
DMX/29	Latino Pack	91	15000

Nota:  
Il supporto è rappresentato da dischi da 5.25" e 3.5" in relazione alle caratteristiche dei disk drive presenti sul computer in oggetto.

## **COMPUTER GRAFICA**

### **PROFESSIONALE E SEMIPROFESSIONALE**

Fornendo qualificata assistenza tecnica pre e post-vendita

- ✦ **Amiga 500, 1000, 2000**
- ✦ **Espansioni e periferiche con tecnologia avanzata a prezzi competitivi**
- ✦ **Nuove unità centrali Motorola 68020 e 68030 con velocità fino a 25 MHz che migliorano le prestazioni di 4-6 volte**
- ✦ **DIGITALIZZATORI:**
  - FRAMER in tempo reale a colori**
  - VD-AMIGA in tempo reale in B/N**
  - VIDEON 2 a colori con filtratura elettronica**
- ✦ **GENLOCK:**
  - ECR - VIDEOCOMP - NERIKI - MAGNI**
- ✦ **Stampanti a colori a getto d'inchiostro XEROX 4020**
- ✦ **SOFTWARE ORIGINALE per paint, animazioni, titoli, modellazione solida, effetti speciali, editoria elettronica**

**INFORMATICA ITALIA** S.r.l.

C.so Re Umberto, 128

10128 TORINO Tel. 011/501647