

Gestione delle interruzioni

di Andrea de Prisco

Alla scoperta del Commodore 64, dopo la grafica, il suono, il disco, il sistema operativo e l'ADP Basic, parleremo di interrupt: perché avvengono e come si possono manipolare per far svolgere nuove funzioni al nostro super smanetto-computer.

Prima parte

Gli Interrupt del CIA 6526

Questo mese parleremo delle interruzioni al microprocessore da parte del CIA 6526; il mese prossimo vedremo quelle relative al video interface chip 6567. Alla fine di questo ciclo avremo aggiunto al nostro 64 i caratteri lampeggianti, la visualizzazione continua dell'orologio, una sveglia programmabile, la visualizzazione di 16 sprite contemporaneamente su video, nonché la possibilità di aprire finestre testografica ad alta risoluzione.

Il tutto s'intende senza paralizzare il funzionamento della macchina: potremo comodamente editare un programma mentre in un angolo dello schermo, secondo dopo secondo, l'orologio scandirà il tempo.

Per ben descrivere queste nuove cose per il 64, dovremo un po' parlare dell'elaborazione parallela, di come sia possibile far girare contemporaneamente (o quasi) due programmi con un solo processore, o se preferite all'interno dello stesso computer. Il "quasi" tra parentesi indica che il trucco c'è, ma non si vede. Di fatto un processore può elaborare un solo programma: è con tutta una serie di piccoli artifici che si può simulare una multiprogrammazione. Si può ad esempio far girare per un intervallo di tempo un programma, per un successivo intervallo un altro, poi un altro e ciclicamente ripetere il tutto, continuamente. Il processore, per così dire, spartisce il tempo, dedicandosi un po' per ogni programma in memoria. Se immaginiamo intervallini di tempo piccolissimi, dell'ordine di qualche millesimo di secondo, l'approssimazione sarà eccellente: a tutti gli effetti il nostro sistema apparirà come una macchina capace di eseguire parallelamente più programmi. Per fare un esempio, immaginiamo di scrivere due programmi grafici: il primo disegna sulla metà sinistra dello schermo un quadrato, l'altro un cerchio nel semischermo opposto. Senza programmazione parallela, qualsiasi computer disegnerà prima l'uno e poi l'altro. Con il sistema sopra descritto, disegnerà un pezzo di quadrato, un pezzo di cerchio, poi un pezzo di quadrato, un pezzo di cerchio, fino al completamento di ambedue le forme. Anche così il risultato può essere deludente, non c'è molta differenza fra i due

modi descritti. Il passo importante è velocizzare quanto più possibile questa sincronia: se il tutto avviene a intervalli di un millesimo di secondo o poco più vedremo

cerchio e quadrato formarsi contemporaneamente sul video. Anzi non sarà possibile dall'esterno capire se sono due processori a eseguire due programmi, ognuno per

PROGRAMMA 1

```

10 FOR I=2049 TO 2302:READ I:I:POKE I,I:NEXT
30 REM *****
40 REM *
50 REM *   O R O L O G I O   *
60 REM *
70 REM *   -----   *
80 REM *   (C) 1985 ADP-SOFTWARE   *
90 REM *   -----   *
91 REM *
92 REM *   POKE44,9:POKE2304,0:NEW   *
93 REM *   PRIMA DI BATTERE O   *
94 REM *   CARICARE QUESTO   *
95 REM *   PROGRAMMA   *
96 REM *
98 REM *****
1000 DATA 120,169,0,141,0,9,169,9,133,44,169,8,141,2
1,3,169,56,141,20,3,165,160
1010 DATA 133,170,165,161,133,171,165,162,133,172,32
,154,8,32,154,8,134,251,32
1020 DATA 154,8,134,252,165,172,133,253,169,0,133,25
4,88,96,230,254,165,254,201
1030 DATA 60,240,3,76,49,234,162,0,134,254,230,251,1
65,251,201,60,208,22,134,251
1040 DATA 230,252,165,252,201,60,208,12,134,252,230,
253,165,253,201,24,208,2,134
1050 DATA 253,160,0,165,253,32,213,8,169,186,153,0,4
,173,134,2,153,0,216,200,165
1060 DATA 252,32,213,8,169,186,153,0,4,173,134,2,153
,0,216,200,165,251,32,213
1070 DATA 8,76,49,234,234,234,234,234,234,234,234,23
4,234,216,162,255,134,255
1080 DATA 134,254,134,253,230,255,208,6,230,254,208,
2,230,253,56,165,172,233,60
1090 DATA 133,172,165,171,233,0,133,171,165,170,233,
0,133,170,176,225,24,165,172
1100 DATA 105,60,170,165,255,133,172,165,254,133,171
,165,253,133,170,96,162,255
1110 DATA 56,232,201,10,48,4,233,10,176,247,72,138,2
4,105,176,153,0,4,173,134
1120 DATA 2,153,0,216,200,104,24,105,176,153,0,4,173
,134,2,153,0,216,200,96

```

```

0001 70 SEI
0002 A9 00 LDA #000
0004 0D 00 09 STA #0900
0007 A9 09 LDA #009
0009 05 2C STA #2C
000B A9 08 LDA #008
000D 0D 15 03 STA #0315
0010 A9 38 LDA #038
0012 0D 14 03 STA #0314
0015 A5 A0 LDA #A0
0017 05 AA STA #AA
0019 A5 A1 LDA #A1
001B 05 AB STA #AB
001D A5 A2 LDA #A2
001F 05 AC STA #AC
0021 20 9A 08 JSR #089A
0024 20 9A 08 JSR #089A
0027 06 FB STX #FB
0029 20 9A 08 JSR #089A
002C 06 FC STX #FC
002E A5 AC LDA #AC
0030 05 FD STA #FD
0032 A9 00 LDA #000
0034 05 FE STA #FE
0036 58 CLI
0037 60 RTS
0038 E6 FE INC #FE
003A A5 FE LDA #FE
003C C9 3C CMP #3C
003E F0 03 BEQ #0943
0040 4C 31 EA JMP #EA31
0043 A2 00 LDX #000
0045 06 FE STX #FE
0047 E6 FB INC #FB
0049 A5 FB LDA #FB
004B C9 3C CMP #3C
004D 08 16 BNE #0865
004F 06 FB STX #FB
0051 E6 FC INC #FC
0053 A5 FC LDA #FC
0055 C9 3C CMP #3C
0057 D0 0C BNE #0865
0059 06 FC STX #FC
005B E6 FD INC #FD
005D A5 FD LDA #FD
005F C9 18 CMP #18
0061 D0 02 BNE #0865
0063 06 FD STX #FD
0065 A0 00 LDY #000
0067 A5 FD LDA #FD
0069 20 D5 08 JSR #08D5
006B A9 BA LDA #08BA
006E 99 00 04 STA #0400,Y
0071 AD 06 02 LDA #0206
0074 99 00 D8 STA #D800,Y
0077 C8 INY
0078 A5 FC LDA #FC
007A 20 D5 08 JSR #08D5
007D A9 BA LDA #08BA
007F 99 00 04 STA #0400,Y
0082 AD 06 02 LDA #0206
0085 99 00 D8 STA #D800,Y
0088 C8 INY
0089 A5 FB LDA #FB
008B 20 D5 08 JSR #08D5
008E 4C 31 EA JMP #EA31

```

Listato 1- Implementazione dell'orologio.

```

0001 70 SEI
0002 A9 00 LDA #000
0004 0D 00 09 STA #0900
0007 A9 09 LDA #009
0009 05 2C STA #2C
000B A9 08 LDA #008
000D 0D 15 03 STA #0315
0010 A9 17 LDA #017
0012 0D 14 03 STA #0314
0015 58 CLI
0016 60 RTS
0017 A5 A2 LDA #A2
0019 29 20 AND #20
001B C5 FF CMP #FF
001D 08 03 BNE #0822
001F 4C 31 EA JMP #EA31
0022 05 FF STA #FF
0024 0A ASL
0025 0A ASL
0026 05 FE STA #FE
0028 49 FF EOR #FF
002A 49 00 EOR #000
002C 05 FD STA #FD
002E A2 00 LDX #000,X
0030 BD 00 D8 LDA #D800,X
0033 29 04 AND #04
0035 D8 0A BNE #D8A1
0037 BD 00 04 LDA #0400,X
0039 05 FE ORA #FE
003C 25 FD AND #FD
003E 9D 00 04 STA #0400,X
0041 BD 00 D8 LDA #D800,X
0044 29 04 AND #04
0046 D8 0A BNE #D852
0048 BD 00 05 LDA #0500,X
004B 05 FE ORA #FE
004D 25 FD AND #FD
004F 9D 00 05 STA #0500,X
0052 BD 00 DA LDA #DA00,X
0055 29 04 AND #04
0057 D8 0A BNE #D863
0059 BD 00 06 LDA #0600,X
005C 05 FE ORA #FE
005E 25 FD AND #FD
0060 9D 00 06 STA #0600,X
0063 BD 00 D8 LDA #D800,X
0066 29 04 AND #04
0068 D8 0A BNE #D874
006A BD 00 07 LDA #0700,X
006D 05 FE ORA #FE
006F 25 FD AND #FD
0071 9D 00 07 STA #0700,X
0074 E8 INX
0075 D0 B9 BNE #D0B9
0077 4C 31 EA JMP #EA31

```

Listato 2- Implementazione caratteri Flashing.

```

0001 70 SEI
0002 A9 00 LDA #000
0004 0D 00 0A STA #0A00
0007 A9 0A LDA #00A
0009 05 2C STA #2C
000B A5 2E LDA #2E
000D 05 FF STA #FF
000F A5 2D LDA #2D
0011 05 FE STA #FE
0013 A0 00 LDY #000

```

```

0015 B1 FE LDA (#FE),Y
0017 C9 41 CMP #41
0019 D0 07 BNE #D022
001B C8 INY
001C B1 FE LDA (#FE),Y
001E C9 CC CMP #CC
0020 F0 21 BEQ #F043
0022 A5 FE LDA #FE
0024 C5 2F CMP #2F
0026 D0 06 BNE #D02E
0028 A5 FF LDA #FF
002A C5 30 CMP #30
002C F0 10 BEQ #F03E
002E 18 CLC
002F D8 CLD
0030 A5 FE LDA #FE
0032 69 07 ADC #07
0034 85 FE STA #FE
0036 A5 FF LDA #FF
0038 69 00 ADC #00
003A 85 FF STA #FF
003C 90 D5 BCC #0813
003E A2 16 LDX #16
0040 4C 37 A4 JMP #A437
0043 C8 INY
0044 B1 FE LDA (#FE),Y
0046 C9 04 CMP #04
0048 D0 F4 BNE #D08E
004A C8 INY
004B B1 FE LDA (#FE),Y
004D 85 FB STA #FB
004F C8 INY
0050 B1 FE LDA (#FE),Y
0052 85 FC STA #FC
0054 A0 00 LDY #000
0056 B1 FB LDA (#FB),Y
0058 29 0F AND #0F
005A 85 AC STA #AC
005C C8 INY
005D B1 FB LDA (#FB),Y
005F 29 0F AND #0F
0061 85 AA STA #AA
0063 A5 AC LDA #AC
0065 0A ASL
0066 0A ASL
0067 0A ASL
0068 18 CLC
0069 65 AC ADC #AC
006B 65 AC ADC #AC
006D 65 AA ADC #AA
006F 65 AA STA #AA
0071 C8 INY
0072 B1 FB LDA (#FB),Y
0074 29 0F AND #0F
0076 85 AC STA #AC
0078 C8 INY
0079 B1 FB LDA (#FB),Y
007B 29 0F AND #0F
007D 85 AB STA #AB
007F A5 AC LDA #AC
0081 0A ASL
0082 0A ASL
0083 0A ASL
0084 18 CLC
0085 65 AC ADC #AC
0087 65 AC ADC #AC
0089 65 AB ADC #AB
008B 85 AB STA #AB

```

```

008D A0 00 LDY #000
008F 04 FF STY #FF
0091 04 FE STY #FE
0093 04 FD STY #FD
0095 A6 AA LDX #AA
0097 F0 16 BEQ #F0AF
0099 18 CLC
009A A5 FF LDA #FF
009C 69 C0 ADC #C0
009E 85 FF STA #FF
00A0 A5 FE LDA #FE
00A2 69 48 ADC #48
00A4 85 FE STA #FE
00A6 A5 FD LDA #FD
00A8 69 03 ADC #03
00AA 85 FD STA #FD
00AC CA DEX
00AD C0 EA BNE #0899
00AF A6 AB LDX #AB
00B1 F0 15 BEQ #F0C8
00B3 18 CLC
00B4 A5 FF LDA #FF
00B6 69 10 ADC #10
00B8 85 FF STA #FF
00BA A5 FE LDA #FE
00BC 69 0E ADC #0E
00BE 85 FE STA #FE
00C0 A5 FD LDA #FD
00C2 69 00 ADC #00
00C4 85 FD STA #FD
00C6 CA DEX
00C7 D0 EA BNE #08B3
00C9 78 SEI
00CA A9 08 LDA #08
00CC 8D 15 03 STA #0315
00CE A9 D6 LDA #D6
00D0 8D 14 03 STA #0314
00D2 58 CLI
00D3 60 RTS
00D4 A5 A2 LDA #A2
00D6 C5 FF CMP #FF
00D8 D0 0C BNE #D0E8
00DA A5 A1 LDA #A1
00DC C5 FE CMP #FE
00DE D0 06 BNE #D0E8
00E0 A5 FD LDA #FD
00E2 C5 A0 CMP #A0
00E4 F0 03 BEQ #F03E
00E6 4C 31 EA JMP #EA31
00E8 A2 18 LDX #18
00EA A9 00 LDA #000
00EC F0 00 D4 STA #D400,X
00EE CA DEX
00EF 10 FA BPL #0EF
00F1 A9 0F LDA #0F
00F3 8D 18 D4 STA #D418
00F5 A9 BE LDA #BE
00F7 8D 05 D4 STA #D405
00F9 A9 59 LDA #59
00FB 8D 06 D4 STA #D406
00FD A9 DC LDA #DC
00FF 8D 01 D4 STA #D401
0101 A9 21 LDA #21
0103 8D 04 D4 STA #D404
0105 4C 31 EA JMP #EA31

```

Listato 3- Implementazione della sveglia programmabile.

conto proprio, o un solo processore che spartisce equamente il suo tempo macchina.

Anche il 64 è una macchina multiprocessore, per l'esattezza bi-processo. Sin dal momento dell'accensione girano contemporaneamente due programmi. Anche in questo caso il "contemporaneamente" va preso con le dovute cautele: diciamo, come sempre, che per qualche attimo fa una cosa, in altri momenti ne fa un'altra. Ciò che normalmente fa è eseguire programmi Basic o più semplicemente attendere comandi mostrando un bel cursore lampeggiante.

Straordinariamente (ogni sessantesimo di secondo) molla tutto e esegue alcune funzioni. Tra queste la famosissima scansione della tastiera per sapere se è stato premuto qualche tasto, e l'aggiornamento dell'orologio interno. Dopo un altro sessantesimo di secondo, rimolla tutto e riesegue la stessa sequenza di istruzioni. Che quanto detto corrisponde a realtà lo si può toccare con mano, semplicemente considerando la variabile TIS: qualsiasi cosa stiamo facendo c'è sempre qualcosa che provvede a incrementarla, segnando sempre l'orario esatto (o quasi).

Un altro "quasi" tra parentesi: cosa c'è? Scendiamo un altro po' nei dettagli. Dicevamo che ogni sessantesimo di secondo il microprocessore interrompe le operazioni in corso per eseguire la scansione della tastiera e altro. Interrompe è il termine più esatto: è proprio una interruzione che arriva dall'esterno ad avvertire il 6510 che un altro 60-esimo di secondo è passato. Uno dei due processori di I/O, un 6526, ha un contatto diretto con il microprocessore: un vero e proprio filo elettrico collega due piedini dei due integrati. Il 6526 manda un impulso elettrico ogni 60-esimo di secondo

al piedino IRQ del 6510. IRQ sta per Interrupt ReQuest, in linguaggio Made in Italy: Richiesta di Interruzione. Non sempre però l'interruzione è accolta: se il processore sta facendo qualcosa di più importante preferisce non essere disturbato. Questo capita, ad esempio, durante le operazioni di Input/Output con le periferiche. Per questo motivo, se si usa il disco, l'aggiornamento dell'orologio interno non è assicurato ad ogni 60-esimo di secondo e lo stesso tende a perdere terreno rispetto all'ora esatta.

Quando l'impulso (l'interrupt) è accolto, il processore salva in apposite locazioni lo stato del processo e esegue il cosiddetto programma di manipolazione delle interruzioni. Lo stato del processo è una coppia di informazioni: una è il Program Counter, l'indirizzo della prossima istruzione da eseguire e l'altra è la Processor Status Word, un particolare registro che conserva varie informazioni tra cui l'esito dell'ultima operazione e altro. Entrambe servono per ripristinare lo stato interno al ritorno dal programma di manipolazione delle interruzioni.

L'indirizzo di questo è memorizzato nelle due ultime locazioni della memoria del 64: la \$FFFE e la \$FFFF e contengono il valore esadecimale \$FF48.

Sappiamo dunque che ogni 60-esimo di secondo il 6510 salta a \$FF48: vediamo cosa succede. Curiosando a quell'indirizzo, troviamo la seguente serie di istruzioni in linguaggio macchina:

```

$FF48 PHA
$FF49 TXA
$FF4A PHA
$FF4B TYA
$FF4C PHA
$FF4D TSX
$FF4E LDA $0104,X
$FF51 AND #$10
$FF53 BEQ $FF58
$FF55 JMP ($0316)
$FF58 JMP ($0314)

```

Le prime 5 istruzioni provvedono a salvare in una apposita zona di memoria detta Stack il contenuto dei tre registri del 6510. Le successive tre istruzioni servono per interrogare la Processor Status Word salvata anch'essa nello Stack al momento dell'interruzione. L'informazione che si vuole conoscere è se l'interrupt proviene dall'esterno (è scoccato un altro 60-esimo di secondo) o è semplicemente dovuto all'esecuzione dell'istruzione BRK (Break) che provoca un interrupt interno. Il BEQ di \$FF53 seleziona le due possibilità: a noi serve il secondo JMP, ci troviamo nel caso di interruzione dall'esterno. L'istruzione JMP corrisponde a un GOTO e ha due formati: diretto e indiretto. Nel primo caso il salto è normale: l'indirizzo che segue la JMP specifica dove si vuole far continuare l'esecuzione. La JMP indiretta si specifica con un indirizzo tra parentesi: nel nostro

PROGRAMMA 2

```

10 FOR I=2049 TO 2169:READ I:POKE I,I:NEXT
20 SYS2049
30 REM *****
40 REM *
50 REM *      CARATTERI FLASHING      *
60 REM *
70 REM *      -----                  *
80 REM *      (C) 1985 ADP-SOFTWARE    *
90 REM *      -----                  *
91 REM *
92 REM *      POKE44,9:POKE2304,0:NEW  *
93 REM *      PRIMA DI BATTERE 0      *
94 REM *      CARICARE QUESTO         *
95 REM *      PROGRAMMA                *
96 REM *
98 REM *****
1000 DATA 120,169,0,141,0,9,169,9,133,44,169,8,141,2
1,3,169,23,141,20,3,88,96
1010 DATA 165,162,41,32,197,255,208,3,76,49,234,133,
255,10,10,133,254,73,255,73
1020 DATA 128,133,253,162,0,189,0,216,41,4,208,10,18
9,0,4,5,254,37,253,157,0,4
1030 DATA 189,0,217,41,4,208,10,189,0,5,5,254,37,253
,157,0,5,189,0,218,41,4,208
1040 DATA 10,189,0,6,5,254,37,253,157,0,6,189,0,219,
41,4,208,10,189,0,7,5,254
1050 DATA 37,253,157,0,7,232,208,185,76,49,234

```

caso si deve saltare all'indirizzo contenuto nella cella \$0314 e seguente (ci vogliono 2 byte per specificare un indirizzo). Cambiando il contenuto di \$0314 e \$0315 è possibile manipolare le interruzioni in modo diverso. Si può ad esempio inserire un nuovo pezzo di programma, a capo di quello per la manipolazione dell'interruzione (locato a EA31), che sarà eseguito

ogni 60-esimo di secondo. Inserirò in \$0314 e \$0315 l'indirizzo della nostra routine e al termine di questa un JMP \$EA31 consentirà l'esecuzione del programma di manipolazione dell'interruzione vero e proprio.

Tre Programmi

Come prima applicazione di questo metodo, è stato preparato il programma "Orologio" che permette la visualizzazione continua della variabile TI\$ nell'angolo in alto a sinistra del video. Per questo e per gli altri, prima di caricarli in memoria è necessario spostare il puntatore di inizio programmi Basic, come indicato in testa ad ogni listato, per fare spazio alla routine in linguaggio macchina.

Dopo aver dato run al primo programma, per far partire l'orologio, sempre dopo aver inizializzato TI\$, è sufficiente digitare SYS 2049.



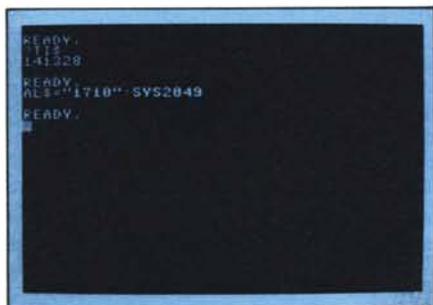
In alto a sinistra il 64 visualizza continuamente l'ora.



Una volta caricato e fatto partire il programma, risulteranno lampeggianti tutti i caratteri visualizzati nei colori Nero, Bianco, Rosso, Ciano, Arancio, Marrone, Rosso Chiaro e Grigio Scuro.



Il programma orologio è assai semplice: la prima operazione che compie è di inizializzare un proprio orologio interno a partire dalla variabile numerica TI locata nelle celle 160-162 (decimale). Con qualche semplice divisione intera a partire da questa, inserisce nelle locazioni 251-253 i corrispondenti secondi, minuti e ore. Una volta ogni 60 chiamate di questo programma, incrementa la locazione 251 (i secondi). Ogni 60 incrementi di 251, incrementa 252 (i minuti) e ogni 60 incrementi di 252, incrementa 253 (le ore). La visualizzazione avviene ogni secondo, quindi se si cancella il video con Shift + CLR/Home dopo pochi attimi si rivedrà comparire l'orario. Quando siamo stufo, è sufficiente battere Run/Stop e Restore per veder scomparire l'orario (SYS 2049 per riattivarlo). Questo perché la sequenza Run/Stop e Restore resetta vari puntatori interni, tra cui l'indirizzo del



L'orologio interno segna le 14 e 13. La sveglia suonerà alle 17 e 10.

programma di manipolazione dell'interruzioni.

Il secondo programma implementa via software i caratteri lampeggianti sul 64. Una volta caricato e fatto partire con SYS 2049 risulteranno lampeggianti tutti i caratteri visualizzati nei colori Nero, Bianco, Rosso, Ciano, Arancio, Marrone, Rosso Chiaro, e Grigio Scuro, in altre parole i colori ottenibili con la pressione dei tasti 1-4, preceduti da Control o dal logo Commodore. Si consiglia di far lampeggiare scritte non in modo diretto (selezionando il colore e digitando il testo), ma sottoforma di istruzione PRINT, racchiudendo tra apici il testo che si vuol fare lampeggiare, inserendo come primo carattere un colore Flashing e come ultimo un colore a scelta tra gli altri 8. In questo modo si evita che porzioni di schermo (arbitrariamente colorate dal 64) inizino a lampeggiare contro le nostre aspettative.

Per quanto riguarda il funzionamento, basta notare che i codici dei colori lampeggianti hanno tutti il bit 2 a zero, (vedi tabella).

Il bit 2 è il terzo a cominciare da destra.

Ogni 32 sessantesimi di secondo (circa 1/2 s.) la nostra routine "parallela" scandisce tutta la pagina colore e dove trova un codice col bit 2 a zero, nella corrispondente posizione in pagina video, settando o resettando il bit 7, converte il carattere da normale a Reverse o viceversa.

PROGRAMMA 3

```

10 FOR I=2049 TO 2320:READ I:I:POKE I,I:NEXT
30 REM *****
40 REM *
50 REM *      S V E G L I A      *
60 REM *
70 REM *      -----      *
80 REM *      (C) 1985 ADP-SOFTWARE      *
90 REM *      -----      *
91 REM *
92 REM *      POKE44,10:POKE2560,0:NEW      *
93 REM *      PRIMA DI BATTERE O      *
94 REM *      CARICARE QUESTO      *
95 REM *      PROGRAMMA      *
96 REM *
98 REM *****
1000 DATA120,169,0,141,0,10,169,10,133,44,165,46,13
3,255,165,45,133,254,160,0
1010 DATA177,254,201,65,208,7,200,177,254,201,204,2
40,33,165,254,197,47,208,6
1020 DATA165,255,197,48,240,16,24,216,165,254,105,7
,133,254,165,255,105,0,133
1030 DATA255,144,213,162,22,76,55,164,200,177,254,2
01,4,208,244,200,177,254,133
1040 DATA251,200,177,254,133,252,160,0,177,251,41,1
5,133,172,200,177,251,41,15
1050 DATA133,170,165,172,10,10,24,101,172,101,17
2,101,170,133,170,200,177
1060 DATA251,41,15,133,172,200,177,251,41,15,133,17
1,165,172,10,10,10,24,101
1070 DATA172,101,172,101,171,133,171,160,0,132,255,
132,254,132,253,166,170,240
1080 DATA22,24,165,255,105,192,133,255,165,254,105,
75,133,254,165,253,105,3,133
1090 DATA253,202,208,234,166,171,240,21,24,165,255,
105,16,133,255,165,254,105
1100 DATA14,133,254,165,253,105,0,133,253,202,208,2
34,120,169,8,141,21,3,169
1110 DATA214,141,20,3,88,96,165,162,197,255,208,12,
165,161,197,254,208,6,165
1120 DATA253,197,160,240,3,76,49,234,162,24,169,0,1
57,0,212,202,16,250,169,15
1130 DATA141,24,212,169,190,141,5,212,169,89,141,6,
212,169,220,141,1,212,169
1140 DATA33,141,4,212,76,49,234
    
```

Colore	Decimale	Binario
Nero	0	0000
Bianco	1	0001
Rosso	2	0010
Ciano	3	0011
Porpora	4	0100
Verde	5	0101
Blu	6	0110
Giallo	7	0111
Arancio	8	1000
Marrone	9	1001
Rosso C.	10	1010
Grigio S.	11	1011
Grigio M.	12	1100
Verde C.	13	1101
Blu C.	14	1110
Grigio C.	15	1111

Il terzo dei tre programmi riportati realizza una vera e propria sveglia programmabile sul 64. Anche in questo caso, dopo aver caricato il programma, bisogna inizializzare TI\$ per informare il computer circa l'orario corrente. Prima del fatidico SYS 2049 bisogna indicare nella variabile ALS l'orario di sveglia. A differenza di TI\$, ALS è in formato 4 cifre: ore e minuti. Quando TI\$ raggiunge ALS si avvertirà un segnale dall'altoparlante del vostro TV. Potrebbe essere utile per ricordarsi gli appuntamenti mentre si lavora al computer: ai softwaristi (ogni riferimento al sottoscritto è puramente "casuale") capita di immergersi talmente nel problema da non avvertire più il passare del tempo! **MC**