

software

APPLE



Due per due uguale... due

Due programmi per due lettori, o due lettori per due programmi? Se due lettori mandano un programma simile e un lettore manda lo stesso programma due volte, quanti programmi ci sono in tutto? Semplice: sempre due! Perché allora pubblicare due programmi uguali, o due volte lo stesso programma? Perché in fondo sono tutti e due interessanti, vuoi per le soluzioni adottate, vuoi perché girano su macchine diverse (sfruttandone appieno le capacità) vuoi perché sono arrivati in redazione nello stesso momento e non mi sembra giusto (essendo tutti e due fatti bene) penalizzare un lettore con un eventuale sorteggio.

Il primo programma (o dovrei dire i primi due) riguarda un orologio in tempo reale sul IIc realizzato via software, gli altri due sono una nuova versione ProDOS di un vecchio programma, sempre per il IIc.

Soft clock

di Flavio Vicentini - Verona

L'idea di scrivere un programma orologio mi ha sempre allettato, forse perché nel buon Apple IIc non esiste la possibilità di installare un orologio (per quanto riguarda l'Italia). Ad ogni modo con un minimo di fantasia è possibile collegare alla nuova mela qualunque periferica ed è possibile far girare programmi di qualunque tipo, per chi non lo sapesse si possono anche far girare contemporaneamente due o tre programmi e senza bisogno di nessuna modifica, ma semplicemente sfruttando l'inesauribile 65C02. L'orologio che ho realizzato non intende essere un'alternativa ad una vera clock card, ma può senz'altro essere molto utile e versatile, soprattutto per coloro che non possono fare a meno di avere sempre e comunque l'ora sotto al naso. Le principali caratteristiche di questo clock sono:

- 1) il programma orologio gira contemporaneamente al sistema operativo ed alla maggior parte dei programmi;
- 2) l'ora è sempre presente nel vertice alto a destra dello schermo indipendentemente dal programma che sta girando e dallo scroll del video;
- 3) la sua precisione è di un cinquantesimo di secondo;
- 4) per come è stato concepito l'orologio effettua una misurazione del tempo che non può sbagliare se non di frazioni di tempo di durata irrilevante.

Come tutte le cose il programma ha anche dei buchi, non molti comunque. Uno di questi, e forse l'unico, è che il suo funzionamento si interrompe con l'attivazione dell'unità disco.

Veniamo ora alle caratteristiche tecniche.

Il primo problema che mi si era presentato era quello di poter scrivere una routine che funzioni contemporaneamente ai normali programmi e al sistema operativo; l'unica tecnica di cui io ero a conoscenza si chiama tecnica ad interrupt. Due parole sull'interrupt.

L'interrupt è un segnale che può essere inviato ad un apposito piedino

Soft clock

```

5 PRINT CHR$(4);"BLOAD SOFT.CLOCK"
10 HOME
30 PRINT "ProDOS SOFT CLOCK by Vicentini Flavio 6 apr 1986"
40 VTAB 10: HTAB 2: INPUT "Introdurre ora (oommss) ";OUR
45 IF OUR > 235959 OR OUR < 000000 THEN 10
50 OO = INT (OUR / 100000):OUR = OUR - OO * 100000
60 POKE 798,OO
70 OO = INT (OUR / 10000):OUR = OUR - OO * 10000
80 POKE 797,OO
90 OO = INT (OUR / 1000):OUR = OUR - OO * 1000
100 POKE 796,OO
110 OO = INT (OUR / 100):OUR = OUR - OO * 100
120 POKE 795,OO
130 OO = INT (OUR / 10):OUR = OUR - OO * 10
140 POKE 794,OO
150 POKE 793,OUR
152 NORMAL
155 HOME
160 POKE 34,1
170 CALL 768
    
```

Listato Basic del programma di settaggio dell'orologio.

Soft clock

```

0300- A9 20 8D FE 03 A9 03 8D
0308- FF 03 A9 50 8D 18 03 A9
0310- 09 20 3D C4 60 00 00 00
0318- 2E 08 01 01 01 0A 02 00
0320- CE 18 03 AD 18 03 C9 00
0328- F0 01 40 EE 19 03 AD 19
0330- 03 C9 0A F0 03 4C 5F 03
0338- A9 00 8D 19 03 EE 1A 03
0340- AD 1A 03 C9 06 F0 03 4C
0348- 9F 03 A9 00 8D 1A 03 EE
0350- 1B 03 AD 1B 03 C9 0A F0
0358- 03 4C 9F 03 A9 00 8D 1B
0360- 03 EE 1C 03 AD 1C 03 C9
0368- 06 F0 03 4C 9F 03 A9 00
0370- 8D 1C 03 EE 1D 03 AD 1E
0378- 03 C9 02 F0 10 AD 1D 03
0380- C9 0A F0 03 4C 9F 03 EE
0388- 1E 03 4C 9F 03 AD 1D 03
0390- C9 04 F0 03 4C 9F 03 A9
0398- 00 8D 1D 03 8D 1E 03 A9
03A0- 32 8D 18 03 A9 01 8D 22
03A8- 00 AD 19 03 69 B0 8D 27
03B0- 04 AD 1A 03 69 B0 8D 26
03B8- 04 A9 AD 8D 25 04 8D 22
03C0- 04 AD 1B 03 69 B0 8D 24
03C8- 04 AD 1C 03 69 B0 8D 23
03D0- 04 AD 1D 03 69 B0 8D 21
03D8- 04 AD 1E 03 69 B0 8D 20
03E0- 04 40 00
    
```

Dump esadecimale del programma che gestisce l'orologio.

della CPU e che provoca l'interruzione del lavoro in corso per andare ad eseguire una routine il cui indirizzo di partenza è memorizzato in due apposite locazioni di memoria.

Il IIc è in grado di ricevere interrupt praticamente da ogni ingresso di cui è provvisto e cioè dal mouse, dalle linee seriali dalla porta per il disco, dai tasti e anche dal suono interno.

È proprio di quest'ultimo ingresso che noi faremo uso.

Questa fonte di interruzione si chiama Video Blank e il suo utilizzo permette di sincronizzare le operazioni di scrittura sul video nel momento in cui il pennello inizia la scansione del video.

Questo segnale nel IIc è stato collegato al sistema dell'interrupt, in questo modo ogni volta che si effettua il refresh del video si genera un interrupt.

Tale refresh viene effettuato ogni cinquantesimo di secondo, quindi è sufficiente fare in modo che ad ogni interrupt si vada ad eseguire un'apposita routine di aggiornamento orologio.

Per fare questo sarà sufficiente porre nelle locazioni 3FE e 3FF rispettivamente la parte bassa e la parte alta dell'indirizzo della routine orologio, e richiamare la routine XSETMOU (sita in locazione C43D) passandole come parametro un valore numerico che indica il tipo di interrupt che si vuole abilitare. Tale numero è il 9 ed andrà posto nell'accumulatore prima di chiamare XSETMOU.

A questo punto non resta che spiegare il funzionamento del programma vero e proprio, tra l'altro molto semplice.

Il programma soft clock ogni volta che viene chiamato incrementa una variabile, quando tale valore sarà arrivato a cinquanta, cioè trascorso un secondo, si effettua l'aggiornamento dell'orologio in memoria e si corregge l'ora sul video, inoltre viene modificata la dimensione della finestra video in modo che l'eventuale scroll del video non faccia sparire l'ora dall'angolo alto a destra dello schermo.

Il programma basic serve per inizializzare l'orologio, cioè porre i valori iniziali dell'ora nelle apposite locazioni e attivare la routine di interrupt. Per eseguire il programma bisognerà digitare:

```
RUN INIT.CLOCK
```

inserire poi le ore, minuti e secondi come se fossero un solo numero, es. 122332 significa ore 12 minuti 23 secondi 32.

L'orologio funziona sia in 40 che 80 colonne senza nessun problema.

Ringrazio MCmicrocomputer e Valter Di Dio per l'articolo di febbraio che senza dubbio mi ha ispirato e stimolato nella realizzazione dell'orologio.

Come si usa

Introdurre disco ProDOS contenente i file «init.clock» e «clock».

Digitare ora:

```
RUN INIT.CLOCK
```

appena appare la videata per l'inserimento dell'ora digitare l'ora come se fosse un unico numero di 6 cifre: es. 150000 significa ore 3 PM, minuti 0, secondi 0.

Orologio in tempo reale

*300,34C

```
0300- A9 10 8D FE 03 A9 03 8D
0308- FF 03 A9 09 20 3D C4 60
0310- EE FB 03 AD FB 03 C9 32
0318- D0 32 A9 00 8D FB 03 EE
0320- FC 03 AD FC 03 C9 3C D0
0328- 23 A9 00 8D FC 03 EE 92
0330- BF AD 92 BF C9 3C D0 14
0338- A9 00 8D 92 BF EE 93 BF
0340- AD 93 BF C9 18 D0 05 A9
0348- 00 8D 93 BF 40
```

▲ Routine che aggiorna l'ora di sistema del ProDOS.
Routine che aggiorna l'ora sul video. ▶

*300,379

```
0300- A9 10 8D FE 03 A9 03 8D
0308- FF 03 A9 09 20 3D C4 60
0310- EE FB 03 AD FB 03 C9 32
0318- D0 5F A9 00 8D FB 03 EE
0320- 07 04 AD 07 04 C9 BA D0
0328- 50 A9 80 8D 07 04 EE 06
0330- 04 AD 06 04 C9 B6 D0 41
0338- A9 80 8D 06 04 EE 04 04
0340- AD 04 04 C9 BA D0 32 A9
0348- 80 8D 04 04 EE 03 04 AD
0350- 03 04 C9 B6 D0 23 A9 80
0358- 8D 03 04 EE 01 04 AD 01
0360- 04 C9 BA D0 14 A9 80 8D
0368- 01 04 EE 00 04 AD 00 04
0370- C9 BA D0 05 A9 80 8D 00
0378- 04 40
```

Per disattivare l'orologio premere CONTROL-RESET. Per far ripartire l'orologio scrivere CALL 768. Per cambiare ora eseguire INIT.CLOCK.

Locazioni di memoria utili:

- 0318 contiene la frazione di secondo attuale, espressa in cinquantesimi di secondo
- 0319 contiene i secondi
- 031A contiene le decine di secondi
- 031B contiene i minuti
- 031C contiene le decine di minuti
- 031D contiene le ore
- 031E contiene le decine di ore

Orologio in tempo reale

di Michele Benvegnù
Cannaregio (VE)

Raccolgo volentieri l'invito, lanciato da Valter Di Dio sul numero 49 di MCmicrocomputer a pagina 117, di apportare utili migliorie alla routine

«Orologio in tempo reale» che sfrutta gli interrupt di Apple IIc per aggiornare l'ora indipendentemente dall'esecuzione del programma in corso.

L'idea è stata quella di utilizzare tale routine per tenere costantemente aggiornata l'ora di sistema. È necessario rammentare che ProDOS conserva nel suo interno la data e l'ora di sistema, che vengono utilizzate al momento di salvare i file su disco per datarli correttamente sul catalogo.

A tale scopo ProDOS utilizza 4 locazioni di memoria, dalla 49040 alla 49043 (\$BF90-\$BF93): le ore occupano la loc. 49043, i minuti la loc. 49042. La data è compattata in due soli byte, e precisamente l'anno occupa i 7 bit più significativi della loc. 49041, il mese occupa il bit meno significativo della loc. 49041 e i 3 bit più significativi della loc. 49040, il giorno occupa i 5 bit meno significativi della loc. 49040.

Così è possibile settare data e ora di sistema con la seguente linea:

```
POKE 49043, ORE: POKE 49042, MINUTI:
POKE 49041, ANNO * 2 + (MESE > 7): POKE 49040, ((MESE - 8) * 32 + GIORNO) * (MESE > 7) + (MESE * 32 + GIORNO) * (MESE < 8)
```

Viceversa è possibile leggere data e ora di sistema con la seguente linea:

```
GIORNO = PEEK (49040) - INT (PEEK (49040) / 32) * 32: ANNO = INT (PEEK (49041) / 2): MESE = (PEEK (49041) - ANNO * 2) * 8 + INT (PEEK (49040) / 32): MINUTI = PEEK (49042): ORE = PEEK (49043)
```

Tasto mela-vuota : Tasto mela-vuota : Tasto mela-piena

A = INIT	A = CREATE	A = CONT
B = CATALOG	B = CATALOG	B = NORMAL
C = MON	C = CAT	C = CALL
D = DELETE	D = DELETE	D = DATA
E = EXEC	E = EXEC	E = END
F = NOMGN	F = FLUSH	F = FOR
G = MAXFILES	G = BYE	G = GOTO
H = INT	H = FRE	H = HOME
I = BRUN	I = BRUN	I = INPUT
J = RUN	J = RUN	J = INVERSE
K = READ	K = READ	K = GOSUB
L = LOAD	L = LOAD	L = LIST
M = BLOAD	M = BLOAD	M = LEFT\$
N = CLOSE	N = CLOSE	N = NEXT
O = OPEN	O = OPEN	O = RESTORE
P = PRE	P = PREFIX	P = PRINT
Q = RENAME	Q = RENAME	Q = RETURN
R = INE	R = RESTORE	R = RIGHT\$
S = SAVE	S = SAVE	S = SPEED=
T = BSAVE	T = BSAVE	T = THEN
U = FP	U = STORE	U = MID\$
V = POSITION	V = POSITION	V = TEXT
W = WRITE	W = WRITE	W = FLASH
X = LOCK	X = LOCK	X = STR\$
Y = UNLOCK	Y = UNLOCK	Y = VAL
Z = APPEND	Z = CHAIN	Z = CHR\$

Tasti ridefiniti

Tasto originale :	Sostituito con :
DELETE	=
*	*
!	!
ò	!
TAB	!
FRECCIA GIU	(
FRECCIA SU)
(meno) -	(sott.)
(sott.) -	(meno)

Comandi abbinati ai tasti mela. La prima colonna è DOS, la seconda ProDOS e la terza del Basic.

tesimi di secondo: dopo 50 cicli, viene resettata e si incrementa la successiva \$3FC. Questa conta i secondi, e viene resettata dopo 60 volte, incrementando la loc. \$BF92, che è quella che ProDOS utilizza per contenere i minuti; dopo 60 minuti, si resetta la loc. \$BF92 e si incrementa la BF93, contenente le ore.

2000:	2000	7	ORG	\$2000
2000:A2 00	8	LDX	\$0	
2002:BD 50 20	9 01	LDA	\$2050,X	
2005:9D 00 03	10	STA	\$300,X	
2008:E8	11	INX		
2009:E0 50	12	CPX	\$F50	
200B:9D F5 2002	13	BCC	Q1	
200D:A2 00	14	LDX	\$0	
200F:2C 81 C0	15	BIT	\$C081	
2012:2C 81 C0	16	BIT	\$C081	
2015:8D 00 21	17 02	LDA	\$2100,X	
2018:9D 00 D4	18 FIL	STA	\$D400,X	
2018:E8	19	INX		
201C:D0 F7 2015 20	20	BNE	Q2	
201E:EE 1A 20	21	INC	FIL+2	
2021:EE 17 20	22	INC	Q2+2	
2024:A0 17 20	23	LDA	Q2+2	
2027:C9 23	24	CHP	\$23	
2029:19 EA 2015 25	25	BCC	Q2	
202B:A9 AC	26	LDA	\$9AC	
202D:8D CB 9A	27	STA	\$9ACB	
2030:A9 00	28	LDA	\$0	
2032:8D C9 9A	29	STA	\$9AC9	
2035:A9 03	30	LDA	\$3	
2037:8D CA 9A	31	STA	\$9ACA	
203A:2C BA C0	32	BIT	\$C0BA	
203D:A0	33	RTS		

Una volta settata, l'ora di sistema rimane costante: purtroppo non è possibile inserire schede orologio in Apple IIc: a ciò è però possibile supplire con una breve routine in L.M., che, sfruttando gli interrupt generati ogni cinquantesimo di secondo dal segnale di video blank, scandisce lo scorrere del

tempo, senza interferire con il programma in esecuzione, dal momento della sua attivazione sino al reset del sistema, esclusi i periodi di accesso al disco.

A tale scopo la routine pubblicata sul numero 49 di MC è stata così modificata: la loc. \$3FB conta i cinquan-

SOURCE FILE EDI =MELA

```

0000: 1 *****
0000: 2 *
0000: 3 * Ridefinizione della tastiera *
0000: 4 ** associazione dei comandi ProDOS al tasto melavuota *
0000: 5 ** associazione dei comandi APPLESOFT al tasto melapiena *
0000: 6 *
0000: 7 * by Antonio Guglielmino Catania (c) 5/7/1986 *
0000: 8 *
0000: 9 *****
0000: 10 *
0000: 0300 11 HERE1 EQU $300
0000: 0400 12 HERE2 EQU $D400
0000: 0004 13 LEN EQU $4
0000: 0200 14 BUFF EQU $200
----- NEXT OBJECT FILE NAME IS /RAM/MELA
0300: 0300 15 ORG HERE1
0300:20 1D FD 14 MAIN1 JSR $FD10 ;Keinput
0303:2C 83 C0 17 BIT $C0B3 ;Attiva la language card in
0306:2C 83 C0 18 BIT $C0B3 ;Letture-Scrittura
0309:2C 41 C0 19 BIT $C061 ;Test mela-vuota
030C:30 17 8225 20 BHI SYSTEM ;Test mela-piena
030E:2C 42 C9 21 BIT $C062
0311:30 09 031C 22 BHI BASIC
0313:20 00 04 23 JSR HERE2
0316:2C 8A C0 24 BIT $C0BA ;Attiva la ROM
0319:4C CB 9A 25 JMP $9ACB
031C:20 3A D4 26 BASIC JSR $A51C1
031F:2C 8A C0 27 BIT $C0BA
0322:4C 00 03 28 JMP HERE1
0325:20 14 04 29 SYSTEM JSR SYSTEM1
0328:2C 8A C0 30 BIT $C0BA
032B:4C 00 03 31 JMP HERE1
032E:2C 8A C0 32 VIDOUT JSR $A51C1
0331:20 F0 FD 33 JSR $FD00 ;Stampa un carattere
0334:2C 83 C0 34 BIT $C0B3
0337:2C 83 C0 35 BIT $C0B3 ;e la riattiva
033A:60 36 RTS
033E:8F 00 D0 37 LEGD1 BIT $C0BA ;Disattiva la L.C.
0341:2C 83 C0 38 LDA $D000,Y ;Legge un carattere dalla ROM
0344:2C 83 C0 39 BIT $C0B3 ;(dalla tavola APPLESOFT)
0347:60 40 RTS ;e la riattiva
----- NEXT OBJECT FILE NAME IS /RAM/MELB
0400: 0400 42 ORG HERE2
0400:DA 43 MAIN2 DFB $DA ;$C0C2 -> PHX
0401:A2 00 44 LDX $0
0403:00 11 D5 45 CCI CMP TAST,X ;Vede se è un tasto ridefinito
0406:F0 07 D40F 46 BEQ OK
0408:EB 47 INX
0409:EB 0B 48 CPX $11
040B:98 F6 0403 49 BEC CCI
040D:FA 50 DFB $FA ;$C0C2 -> PLX
040E:60 51 RTS
040F:BD 1B D5 52 OK LDA RIDEFI,X
0412:80 F7 53 DFB $F7 ;$C0C2 -> BRA
0414:5A 54 SYSTEM1 DFB $5A ;$C0C2 -> PHY
0415:A0 00 55 LDY $0
0417:DF 75 D4 56 CI CMP TAST1,Y
041A:F0 07 D423 57 BEQ TROWATO ;con la mela-vuota
041C:CB 58 INY
041D:C0 18 59 CPY $27
041F:90 F6 D417 60 BCC C1
0421:7A 61 DFB $7A ;$C0C2 -> PLY
0422:A0 62 RTS
0423:8F 0F D4 63 TROWATO LDA LUNGH,Y ;Sostituisco con la parola abbinata
0426:85 04 64 STA LBN
0428:8F C3 D4 65 LDA BYLOW,Y

```

```

D42B:8D 31 D4 66 STA LETTURA+1
D42E:A0 00 67 LDY $0
D430:89 00 88 68 LETTURA LDA $8800,Y ;Legge un carattere dalla tavola
D433:20 64 D4 69 JSR ROUTINE ;dei comandi BASIC.
D436:90 FB D430 70 BCC LETTURA
D438:7A 71 DFB $7A ;$C0C2 -> PLY
D439:60 72 RTS
D43A:5A 73 BASIC1 DFB $5A ;$C0C2 -> PHY
D43B:A0 00 74 LDY $0
D43D:DF 75 D4 75 RE1 CMP TAST1,Y ;Vede se è un tasto programmato
D440:F0 07 D449 76 BEQ TROWO ;con la mela-piena
D442:CB 77 INY
D443:C0 18 78 CPY $27
D445:90 F6 D430 79 BEC RE1
D447:7A 80 DFB $7A ;$C0C2 -> PLY
D448:60 81 RTS
D449:8F A9 D4 82 TROWO LDA LUNGH2,Y ;Sostituisco con la parola abbinata
D44C:85 04 83 STA LBN
D44E:8F D0 D4 84 LDA BYLOW2,Y
D451:8D 3F 03 85 STA LEGGI+4
D454:8F F7 D4 86 LDA $BYHI,Y
D457:8D 40 03 87 STA LEGGI+5
D45A:A0 00 88 LDY $0
D45C:20 3B 03 89 LETTURA2 JSR LEGGI ;Guarda sopra
D45F:20 66 D4 90 JSR ROUTINE
D462:90 F6 D45C 91 BCC LETTURA2
D464:7A 92 DFB $7A ;$C0C2 -> PLY
D465:60 93 RTS
D466:90 0B 02 94 ROUTINE STA BUFF,X ;ROUTINE che inserisce nel buffer di INPUT
D469:5A 95 DFB $5A ;$C0C2 -> PHY
D46A:0F 80 96 ORA $80 ;i comandi (un carattere per volta)
D46C:20 2E 03 97 JSR VIDOUT ;e li stampa
D46F:7A 98 DFB $7A ;$C0C2 -> PLY
D470:EB 99 INX
D471:CB 100 INY
D472:C4 04 101 CPY LBN
D474:60 102 RTS
D475: 103 *
D475: 104 * I tasti programmati sono progressivi dalla A alla Z
D475: 105 *
D475:C1 C2 C3 C4 106 TAST1 DFB $C1,$C2,$C3,$C4,$C5,$C6,$C7,$C8,$C9,$DA,$DB,$DC,$DD
D482:CC CF D0 D1 107 DFB $CE,$CF,$D0,$D1,$D2,$D3,$D4,$D5,$D6,$D7,$D8,$D9,$DA
D48F: 108 *
D48F: 109 * Lunghezza dei comandi BASIC.SYSTEM
D48F: 110 *
D48F:04 07 03 06 111 LUNGH DFB 6,7,3,6,4,5,3,3,4,3,4,4,5,4,6,6,7,4,5,8,5,4,4,5
D4A9: 112 *
D4A9: 113 * Lunghezza dei comandi APPLESOFT
D4A9: 114 *
D4A9:04 06 04 04 115 LUNGH2 DFB 4,6,4,4,3,4,4,4,5,7,5,4,7,5,6,6,6,4,4,4,5,4,3,4
D4C3: 116 *
D4C3: 117 * Byte basso della tavola dei comandi BASIC.SYSTEM
D4C3: 118 *
D4C3:A0 BE BE 85 119 BYLOW DFB $A0,$BE,$8E,$85,$90,$C0,$8B,$A6,$82,$83,$C5,$82,$81
D4D0:DE 95 DB AC 120 DFB $DE,$95,$8B,$AC,$A7,$79,$7B,$A9,$C9,$99,$86,$84,$8A
D4D0: 121 *
D4D0: 122 * Byte basso della tavola dei comandi APPLESOFT
D4D0: 123 *
D4D0:00 49 F9 DA 124 BYLOW2 DFB $00,$49,$F9,$0A,$00,$03,$93,$29,$80,$4F,$A4,$04,$50
D4EA:04 9C CB A9 125 DFB $06,$9C,$CB,$A9,$40,$8A,$EF,$5B,$EF,$56,$42,$44,$4C
D4F1: 126 *
D4F1: 127 * Byte alto della tavola dei comandi APPLESOFT
D4F1: 128 *
D4F1:01 D1 D0 D0 129 BYHI DFB $01,$01,$00,$00,$00,$00,$01,$01,$00,$01,$01,$01,$01,$02
D504:00 D1 D1 D1 130 DFB $00,$01,$01,$01,$02,$01,$01,$02,$00,$01,$02,$02,$02
D511: 131 *
D511: 132 * Tasti da ridefinire senza i tasti mela
D511: 133 *
D511:DF AD FF DB 134 TAST DFB $DF,$AD,$FF,$DB,$A1,$A4,$FC,$89,$8A,$8B
D51B: 135 *
D51B: 136 * Valore da ridefinire ai suddetti tasti
D51B: 137 *
D51B:A0 DF BD A4 138 RIDEFI DFB $A0,$DF,$BD,$A4,$8A,$0B,$A1,$AC,$A8,$A9

```

Listato del programma INTEGRATO. Chi non dispone di un assembler deve caricare solo i codici delle prime tre colonne.

Per l'utilizzo, si carica la routine dal monitor e la si attiva, dopo aver eventualmente settato l'ora, con CALL 768.

Un'altra semplice ed interessante modifica della routine permette di visualizzare costantemente l'ora dello schermo, mentre Apple continua indisturbato il suo lavoro.

L'idea in questo caso è stata quella di utilizzare direttamente le prime 8 locazioni della pagina di testo (\$400-\$407) ed in esse inserire le cifre da 0 a 9 come codici di schermo e su di essi agire con i relativi incrementi e controlli.

Le locazioni dalla \$400 in poi contengono rispettivamente decine di ore, ore, due punti, decine di minuti, minuti, due punti, decine di secondi, secondi.

Da notare che, dopo aver caricato in memoria la routine dal monitor, l'ora può essere settata senza ricorrere a complesse POKE, ma con una banale PRINT; ad esempio:

```
TEXT: HOME: PRINT "21:34:48": POKE 34,1: CALL 768
```

Il POKE 34,1 serve a proteggere la prima linea dello schermo da indesiderati scroll. Se si lavora in 80 colonne, ricordando che le locazioni in oggetto riguardano le colonne pari (seconda, quarta, ecc.) mentre le colonne dispari risiedono nel secondo banco di memoria, è sufficiente modificare il comando PRINT aggiungendo uno spazio prima di ciascun carattere:

```
TEXT: HOME: PRINT "2 1: 3 4: 4 8": POKE 34,1: CALL 768
```

Integrato ProDOS

di Antonio Guglielmino
Catania

Colgo al volo la richiesta fatta sul n. 53 di MC riguardante le modifiche al programma INTEGRATO pubblicato a pag. 114 del n. 49 di febbraio.

Il programma che presento non è una modifica al suddetto programma, ma una nuova versione che utilizza, invece dei caratteri di controllo, i tasti mela-piena e mela-vuota. Ho dedicato il tasto mela-vuota per i comandi del ProDOS ed il tasto mela-piena per i comandi BASIC Applesoft. Ho anche ridefinito alcuni tasti, molto importanti per chi programma sia in Basic che in Assembler con l'Apple IIc. Per mettere in funzione il programma basta battere -MELA.2 e si rilocherà una parte, quella che si occupa delle operazioni di I-O, nella pagina 3 ed il re-

sto nel Bank-Switching n. 2 della Language Card «NON OCCUPATA DAL PRODOS». Ho voluto fare questa prova per far vedere che non necessariamente bisogna rubare spazio al basic od al ProDOS per programmi in linguaggio macchina ed anche per utilizzare la memoria che solitamente non vediamo mai.

Ma torniamo al programma. Dobbiamo, come al solito, batterlo o con l'assemblatore EDASM. Se si battono i codici esadecimale, l'unica cosa che dobbiamo fare è quella di salvare il programma con BSAVE MELA.2, A\$2000, E\$2224. Viceversa, se si utilizza l'EDASM bisogna seguire attentamente queste istruzioni:

- 1) battere il primo listato in assembler LOADER.MELA.
- 2) Battere il secondo listato in assembler MELA.
- 3) Assemblarli entrambi.
- 4) Entrare in ambiente BASIC.SYSTEM.
- 5) Caricare il programma LOADER.MELA.0 con:

```
BLOAD LOADER.MELA.0,A$2000
```

- 6) Caricare il programma MELA.0 con:

```
BLOAD MELA.0,A$2050
```

- 7) Caricare il programma MELA.1 con:

```
BLOAD MELA.1,A$2100
```

- 8) Salvare il tutto con:

```
BSAVE MELA.2,A$2000,E$2224
```

9) Se tutto è stato fatto correttamente si potrà lanciare il programma. NB. I nomi dei file, ad esempio MELA.0, sono prodotti dall'assemblatore EDASM.

NB2. Il programma gira sul IIc o sul Iie enhanced.

Una volta eseguito il programma, esso non si può più scollegare neanche con il CONTROL-RESET e per poterlo eventualmente fare bisogna rilanciare il BASIC.SYSTEM essendo il programma «LINKATO» ad esso.

Integrato per DOS

Questa invece è la versione in DOS 3.3 del programma.

Per battere il programma bisogna caricare i codici esadecimale in memoria con una certa prassi:

- 1) Entrare in monitor con: CALL -151.
- 2) Battere il primo listato (MELA.DOS.1) a partire dalla locazione \$0300.
- 3) Salvarlo con BSAVE MELA.DOS.1,A\$300,LS\$71.
- 4) Battere: C081 N C081 <RETURN> si attiverà la L.C.
- 5) Battere l'altro listato a partire dalla locazione \$D400.
- 6) Salvarlo con BSAVE MELA.DOS.2,A\$D400,LS\$160.
- 7) Riattivare la ROM con: C08A <RETURN>.

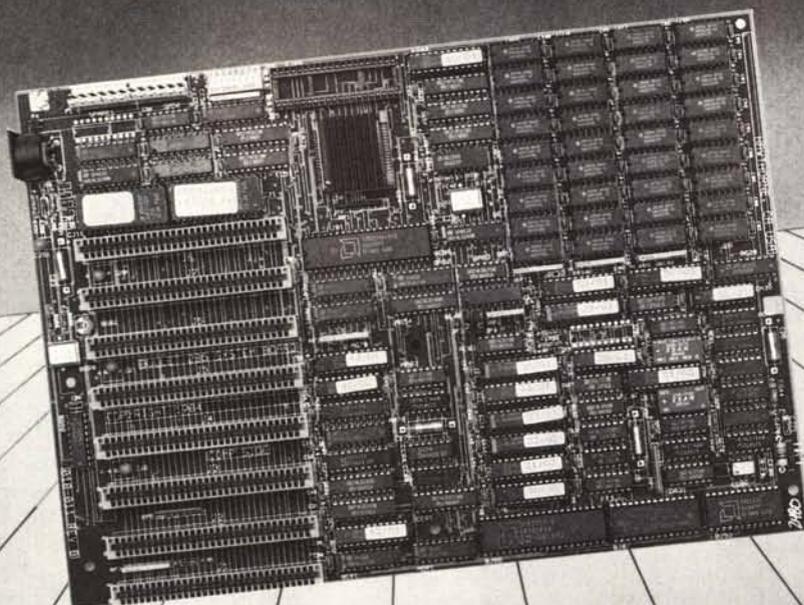
Per lanciare il programma bisogna attivare la L.C. in scrittura entrando in monitor (CALL-151) e battere C081 N C081, caricare i due programmi (BLOAD MELA.DOS.1 e BLOAD MELA.DOS.2) e lanciaarli con 300G se si è ancora in monitor, con il tasto RESET se già stato attivato, oppure con CALL 768 da basic.

NB2. Il programma gira sul IIc o sul Iie enhanced.

```
MELA.DOS 1
0300-- 608FA79D4A1C4
0301-- 2000000000000000
0302-- 0000000000000000
0303-- 0000000000000000
0304-- 0000000000000000
0305-- 0000000000000000
0306-- 0000000000000000
0307-- 0000000000000000
0308-- 0000000000000000
0309-- 0000000000000000
0310-- 0000000000000000
0311-- 0000000000000000
0312-- 0000000000000000
0313-- 0000000000000000
0314-- 0000000000000000
0315-- 0000000000000000
0316-- 0000000000000000
0317-- 0000000000000000
0318-- 0000000000000000
0319-- 0000000000000000
0320-- 0000000000000000
0321-- 0000000000000000
0322-- 0000000000000000
0323-- 0000000000000000
0324-- 0000000000000000
0325-- 0000000000000000
0326-- 0000000000000000
0327-- 0000000000000000
0328-- 0000000000000000
0329-- 0000000000000000
0330-- 0000000000000000
0331-- 0000000000000000
0332-- 0000000000000000
0333-- 0000000000000000
0334-- 0000000000000000
0335-- 0000000000000000
0336-- 0000000000000000
0337-- 0000000000000000
0338-- 0000000000000000
0339-- 0000000000000000
0340-- 0000000000000000
0341-- 0000000000000000
0342-- 0000000000000000
0343-- 0000000000000000
0344-- 0000000000000000
0345-- 0000000000000000
0346-- 0000000000000000
0347-- 0000000000000000
0348-- 0000000000000000
0349-- 0000000000000000
0350-- 0000000000000000
0351-- 0000000000000000
0352-- 0000000000000000
0353-- 0000000000000000
0354-- 0000000000000000
0355-- 0000000000000000
0356-- 0000000000000000
0357-- 0000000000000000
0358-- 0000000000000000
0359-- 0000000000000000
0360-- 0000000000000000

MELA.DOS 2
0400-- 0000000000000000
0401-- 0000000000000000
0402-- 0000000000000000
0403-- 0000000000000000
0404-- 0000000000000000
0405-- 0000000000000000
0406-- 0000000000000000
0407-- 0000000000000000
0408-- 0000000000000000
0409-- 0000000000000000
0410-- 0000000000000000
0411-- 0000000000000000
0412-- 0000000000000000
0413-- 0000000000000000
0414-- 0000000000000000
0415-- 0000000000000000
0416-- 0000000000000000
0417-- 0000000000000000
0418-- 0000000000000000
0419-- 0000000000000000
0420-- 0000000000000000
0421-- 0000000000000000
0422-- 0000000000000000
0423-- 0000000000000000
0424-- 0000000000000000
0425-- 0000000000000000
0426-- 0000000000000000
0427-- 0000000000000000
0428-- 0000000000000000
0429-- 0000000000000000
0430-- 0000000000000000
0431-- 0000000000000000
0432-- 0000000000000000
0433-- 0000000000000000
0434-- 0000000000000000
0435-- 0000000000000000
0436-- 0000000000000000
0437-- 0000000000000000
0438-- 0000000000000000
0439-- 0000000000000000
0440-- 0000000000000000
0441-- 0000000000000000
0442-- 0000000000000000
0443-- 0000000000000000
0444-- 0000000000000000
0445-- 0000000000000000
0446-- 0000000000000000
0447-- 0000000000000000
0448-- 0000000000000000
0449-- 0000000000000000
0450-- 0000000000000000
0451-- 0000000000000000
0452-- 0000000000000000
0453-- 0000000000000000
0454-- 0000000000000000
0455-- 0000000000000000
0456-- 0000000000000000
0457-- 0000000000000000
0458-- 0000000000000000
0459-- 0000000000000000
0460-- 0000000000000000
0461-- 0000000000000000
0462-- 0000000000000000
0463-- 0000000000000000
0464-- 0000000000000000
0465-- 0000000000000000
0466-- 0000000000000000
0467-- 0000000000000000
0468-- 0000000000000000
0469-- 0000000000000000
0470-- 0000000000000000
0471-- 0000000000000000
0472-- 0000000000000000
0473-- 0000000000000000
0474-- 0000000000000000
0475-- 0000000000000000
0476-- 0000000000000000
0477-- 0000000000000000
0478-- 0000000000000000
0479-- 0000000000000000
0480-- 0000000000000000
0481-- 0000000000000000
0482-- 0000000000000000
0483-- 0000000000000000
0484-- 0000000000000000
0485-- 0000000000000000
0486-- 0000000000000000
0487-- 0000000000000000
0488-- 0000000000000000
0489-- 0000000000000000
0490-- 0000000000000000
0491-- 0000000000000000
0492-- 0000000000000000
0493-- 0000000000000000
0494-- 0000000000000000
0495-- 0000000000000000
0496-- 0000000000000000
0497-- 0000000000000000
0498-- 0000000000000000
0499-- 0000000000000000
0500-- 0000000000000000
0501-- 0000000000000000
0502-- 0000000000000000
0503-- 0000000000000000
0504-- 0000000000000000
0505-- 0000000000000000
0506-- 0000000000000000
0507-- 0000000000000000
0508-- 0000000000000000
0509-- 0000000000000000
0510-- 0000000000000000
0511-- 0000000000000000
0512-- 0000000000000000
0513-- 0000000000000000
0514-- 0000000000000000
0515-- 0000000000000000
0516-- 0000000000000000
0517-- 0000000000000000
0518-- 0000000000000000
0519-- 0000000000000000
0520-- 0000000000000000
0521-- 0000000000000000
0522-- 0000000000000000
0523-- 0000000000000000
0524-- 0000000000000000
0525-- 0000000000000000
0526-- 0000000000000000
0527-- 0000000000000000
0528-- 0000000000000000
0529-- 0000000000000000
0530-- 0000000000000000
0531-- 0000000000000000
0532-- 0000000000000000
0533-- 0000000000000000
0534-- 0000000000000000
0535-- 0000000000000000
0536-- 0000000000000000
0537-- 0000000000000000
0538-- 0000000000000000
0539-- 0000000000000000
0540-- 0000000000000000
0541-- 0000000000000000
0542-- 0000000000000000
0543-- 0000000000000000
0544-- 0000000000000000
0545-- 0000000000000000
0546-- 0000000000000000
0547-- 0000000000000000
0548-- 0000000000000000
0549-- 0000000000000000
0550-- 0000000000000000
0551-- 0000000000000000
0552-- 0000000000000000
0553-- 0000000000000000
0554-- 0000000000000000
0555-- 0000000000000000
0556-- 0000000000000000
0557-- 0000000000000000
0558-- 0000000000000000
0559-- 0000000000000000
0560-- 0000000000000000
```

Dump esadecimale della versione DOS 3.3 del programma INTEGRATO.



AFFIDABILITÀ E VELOCITÀ CON BULLET 286

La **BULLET 286** trasforma il tuo XT in un COMPUTER con maggior po-

tenza e velocità dell'AT. Questo prodotto a tecnologia avanzata utilizza il MICROPROCESSORE INTEL 80286, in una situazione di completa compatibilità HARDWARE e SOFTWARE PC-XT. Rimpiazzando semplicemente la scheda madre del Vostro XT con la nuova BULLET 286, potrete disporre di velocità e potenza maggiore del PC/AT.

La **BULLET 286** contiene una CPU che lavora senza WAIT STATE. Tutti i programmi applicativi e sistemi operativi che sono disponibili per il PC-XT possono lavorare senza incompatibilità sulla BULLET 286 (PC-DOS, CP/M, UCSD Pascal, Pick, Oasis, Unix-derivatives). La **BULLET 286** è disponibile anche con 1 MByte di RAM integrata sulla scheda.

Formaster · Fourmaster · Fourmaster · Fo

DAI PIÙ MUSCOLI AL TUO PC



Fourmaster continua ad offrirti il miglior prodotto per il tuo Personal Computer.

VANTAGGI FOURMASTER®

- Organizzazione efficiente e capillare
- Know - How di prodotto a disposizione del cliente
- Miglior rapporto prezzo/prestazioni
- Completa assistenza tecnica
- Garanzia sul prodotto di 12 mesi



Fourmaster® s.r.l.

Via Pellizzari, 28 20059 VIMERCATE (MI)

Tel. 663180-664623-664581/2

● ROMA: 06/5420305-5423716

● TORINO: 011/6199817 - 617362

● VERONA: 045/48347

● PADOVA: 049/725359

● BOLOGNA: 051/271018

Desidero ricevere ulteriori informazioni sui prodotti da voi distribuiti.

Nome Cognome

Azienda

Via

CAP. Città

Inviare il coupon a: Fourmaster s.r.l. Via Pellizzari, 28 - 20059 Vimercate (MI)

Fourmaster · Fourmaster ·