

APPLE-1 JUKE-BOX EEPROM

v1.09

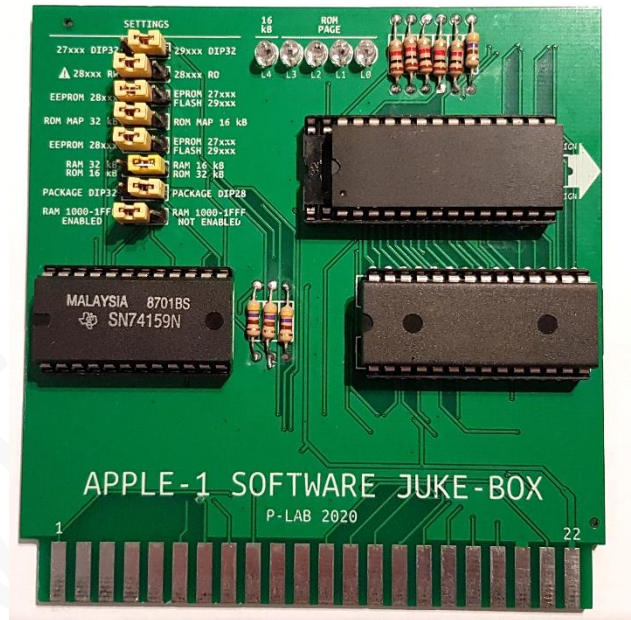
Avete deciso di scrivere i vostri programmi in BASIC e le vostre routine in Linguaggio Macchina? Bene! Ci congratuliamo con voi!

La *Apple-1 Software Juke-Box* semplificherà la vostra vita, mediante un **supporto di memoria dedicato** su cui potrete salvare i frutti del vostro lavoro.

Questo supporto è una **EEPROM (Electrically Erasable Programmable Read-Only Memory)**.

Essa contiene al suo interno tutto quello che vi serve, ossia:

- L'Integer BASIC,
- Il Program Manager come già descritto nel manuale d'uso principale,
- Un programma speciale che vi consente di **salvare i vostri programmi direttamente sulla EEPROM**.



1. SOSTITUZIONE DEL DISPOSITIVO DI MEMORIZZAZIONE

Per prima cosa è necessario sostituire il dispositivo di memorizzazione esistente (ad esempio la FLASH Memory 29c020 in dotazione) con la **EEPROM 28c256** fornita a corredo.

Tutte queste operazioni vanno eseguite a computer spento.

Mettete in atto le precauzioni contro l'accumulo di cariche elettrostatiche descritte nel manuale principale.

Estraete con cautela la FLASH ed inserite la EEPROM, facendo estrema attenzione a non piegare i piedini in fase di inserimento nello zoccolo.

Fate combaciare il lato destro della EEPROM con il bordo della scheda e tenete la tacca verso sinistra, nello stesso verso degli altri Circuiti Integrati presenti, in accordo alla foto sopra riportata e alla serigrafia sul Circuito Stampato.

Controllate almeno tre volte che non ci siano pin piegati, da tutte le angolazioni.

Ora, **riconfigurate i vari jumpers** come indicato nella foto a destra.

Ponete attenzione al secondo jumper dall'alto: 28xxx RW/RO.

Se esso sarà posizionato dal **lato RW (Read/Write)** la scrittura sul dispositivo sarà abilitata e quindi potrete salvare i vostri programmi. Viceversa, il dispositivo sarà in *sola lettura (Read Only)*.



2. MAPPATURA DELLA EEPROM

La EEPROM 28c256 può memorizzare 32 kBytes. Essa è stata divisa in BLOCCHI (Blocks) da 4 kBytes, e risulterà dunque mappata sul computer secondo il seguente schema di indirizzamento:

START ADDRESS		END ADDRESS
\$4000	4 kB reserved for BASIC	\$4FFF
\$5000	Block 6 4 kB	\$5FFF
\$6000	Block 5 4 kB	\$6FFF
\$7000	Block 4 4 kB	\$7FFF
\$8000	Block 3 4 kB	\$8FFF
\$9000	Block 2 4 kB	\$9FFF
\$A000	Block 1 4 kB	\$AFFF
\$B000	Block 0 / 2 kB	\$B7FF
\$B800	RESERVED	\$BFFF

Alcune aree sono state utilizzate per memorizzare l'Integer Basic e i vari programmi di gestione ossia il Program Manager tradizionale e l'utility di salvataggio che verrà tra poco descritta.

È importante notare che sono stati previsti sei Blocchi da 4 kBytes e uno da 2 kBytes.

3. UTILIZZO

L'uso della EEPROM 28c256 e dei programmi in essa contenuti è assolutamente identico a quanto già visto nel manuale principale. Il Program Manager si avvia come di consueto con il comando:

BD00R seguito dal tasto ENTER (da qui in avanti rappresentato con l'indicazione {ENTER})

Comparirà quanto segue:

```
BD00: A5  
&
```

Richiediamo ora la lista dei file disponibili mediante il consueto comando **D** (Directory):

```
&D {ENTER}
PAGE 2

A FREEBLK1 $0000-$0000
B FREEBLK2 $0000-$0000
C FREEBLK3 $0000-$0000
D FREEBLK4 $0000-$0000
E FREEBLK5 $0000-$0000
F FREEBLK6 $0000-$0000
G FREEBLK0 $0000-$0000
H BASIC $E000-$F000
OK

&
```

N.B. Ignorare l'indicazione della pagina (PAGE), e dei LED dato che la capacità della EEPROM è di 32 kB.

L'elenco dei programmi ci mostra tutti i Blocchi liberi (FREEBLK), numerati come nell'esempio precedente. In questo elenco appariranno i programmi che salverete in BASIC e in Linguaggio Macchina, oltre al BASIC stesso.

3.1 SALVATAGGIO DI PROGRAMMI BASIC

Supponiamo che abbiate caricato l'interprete BASIC e che abbiate scritto il vostro programma. Per salvare il programma è necessario dapprima tornare al WOZ Monitor mediante pressione del tasto RESET. Al prompt “\” lanciare il **Programma di Salvataggio** (il cui *entry-point* è 0xB800) con il comando:

```
B800R {ENTER}
```

Comparirà quanto segue:

```
B800: 20
#
```

Il simbolo “#” è il prompt generato dal Programma di Salvataggio. Esso indica che è pronto ad accettare comandi. Diversamente dal Program Manager, i comandi da impartire sono composti da una unica lettera. La pressione della lettera corrispondente ad un comando valido causa l'esecuzione immediata del comando scelto.

La pressione di una lettera non ammessa causerà la stampa a schermo di un Help minimale:

```
#W/S/L/X?
```

Il significato dei comandi è il seguente:

- W** (Write) scrive su EEPROM una porzione di 4 kBytes partendo da un indirizzo RAM arbitrario,
- S** (Save) salva su EEPROM un programma scritto in BASIC,
- L** (Loader) lancia il Program Manager,
- X** (eXit) esce al WOZ Monitor.

Visto che desideriamo salvare un programma in BASIC sceglieremo il comando **S**.

Immediatamente dopo la pressione del tasto "S" comparirà il messaggio:

SAVE BASIC TO BLOCK :

Ora è necessario specificare su quale Blocco salvare il programma BASIC, tenendo presente che il Blocco 0 non può essere utilizzato. Scegliamo per esempio il **Blocco 1**. Dopo la pressione del tasto "1" comparirà il messaggio:

WITH NAME :

Scrivere ad esempio **TEST 1** e premere ENTER per confermare.

Tenere presente che il nome del programma può avere al massimo otto caratteri.

Alla pressione dell'ottavo carattere (o del tasto ENTER) il nome verrà ritenuto completo e **il processo di salvataggio inizierà immediatamente**.

L'avanzamento del processo di salvataggio è rappresentato con una sequenza di punti che si accodano.

.....

Ogni punto rappresenta la scrittura in EEPROM di 256 bytes. A fine processo compariranno dunque 16 punti.

La scrittura non è istantanea: **per salvare 4 kBytes occorreranno circa 25 secondi**.

Tale tempo è assolutamente normale ed è conforme alle specifiche di funzionamento della EEPROM in uso.

Al termine della scrittura in EEPROM comparirà il prompt del Programma di Salvataggio:

#

Ora, se vogliamo verificare che il nostro programma sia stato salvato correttamente abbiamo la possibilità di lanciare direttamente il Program Manager mediante la pressione del tasto L.

Comparirà immediatamente il prompt "&" e sarà possibile richiedere la lista dei programmi, come di consueto:

premere L → N.B. la lettera L non comparirà a schermo.

&D {ENTER}

PAGE 2

A	TEST 1	\$0280-\$1000	BAS
B	FREEBLK2	\$0000-\$0000	
C	FREEBLK3	\$0000-\$0000	
D	FREEBLK4	\$0000-\$0000	
E	FREEBLK5	\$0000-\$0000	
F	FREEBLK6	\$0000-\$0000	
G	FREEBLK0	\$0000-\$0000	
H	BASIC	\$E000-\$F000	

OK

&

Ricordiamo che il primo indirizzo è il punto di partenza del programma e il secondo è il primo indirizzo libero.

Dal momento del salvataggio il vostro programma BASIC entra a far parte in maniera permanente dei programmi presenti sulla EEPROM. Per caricarlo/ricaricarlo sarà sufficiente dare il comando **LA** dal programma di gestione.

Lo stesso metodo di salvataggio si applica per qualunque altro Blocco.

Notate che non è previsto alcun avviso in caso di sovrascrittura di un Blocco già scritto.

Di default il salvataggio dei programmi BASIC ha come indirizzo massimo di memoria 0x0FFF.

Vedremo nei capitoli successivi come sia possibile salvare programmi che superano tale limite.

3.2 SALVATAGGIO DI PROGRAMMI IN LINGUAGGIO MACCHINA

Le routine in linguaggio macchina, che possono risiedere in qualunque porzione della memoria RAM, possono essere salvate mediante il comando **W** del Programma di Salvataggio.

Oltre al nome (supponiamo sia TESTPROG) verrà richiesto anche l'indirizzo esadecimale di partenza del salvataggio. **A partire da tale locazione verranno copiati in EEPROM, sul Blocco scelto, tutti i 4 kBytes successivi.**

Supponiamo che la routine che desideriamo salvare parta dall'indirizzo 0x0300.

Una volta lanciato il Programma di Salvataggio con **B800R** premeremo il tasto **W**.

Comparirà infatti il seguente messaggio:

```
SAVE MEMORY FROM: $
```

A questo punto digiteremo l'indirizzo di partenza, in formato esadecimale su quattro cifre premendo i tasti **0300**. Alla pressione del quarto tasto l'indirizzo di partenza verrà consolidato. Verrà dunque richiesto il nome del file:

```
WITH NAME:
```

Analogamente al caso precedente, occorre dare un nome al file che si desidera salvare. Alla pressione del tasto ENTER o alla digitazione dell'ottavo carattere del nome (**TESTPROG** in questo esempio) il nome verrà consolidato.

Comparirà infine la richiesta del Blocco sul quale salvare il programma:

```
TO BLOCK:
```

Scegliere un Blocco e digitare il numero corrispondente (es. **3**). **Il salvataggio inizierà immediatamente.**

È disponibile, all'occorrenza, un mini Blocco da 2 kBytes, definito come Blocco 0.

Come nel caso precedente l'avanzamento del processo di salvataggio sarà rappresentato con una sequenza di punti che si accodano, seguiti dal carattere “#” alla fine delle operazioni.

```
.....  
#
```

A questo punto sarà possibile verificare l'avvenuto salvataggio, che comparirà come voce “C” dell'elenco:

```
# premere L → N.B. la lettera L non comparirà a schermo.
```

```
&D {ENTER}
```

```
PAGE 2
```

```
A TEST 1      $0280-$1000  BAS  
B FREEBLK2   $0000-$0000  
C TESTPROG   $0300-$1300  
D FREEBLK4   $0000-$0000  
E FREEBLK5   $0000-$0000  
F FREEBLK6   $0000-$0000  
G FREEBLK0   $0000-$0000  
H BASIC      $E000-$F000
```

```
OK
```

```
&
```

Il caricamento della porzione di memoria contenente la vostra routine avviene in maniera tradizionale. In questo esempio mediante il comando **LC**. Notate l'assenza del suffisso “BAS” in corrispondenza di TESTPROG.

3.3 HIMEM E LOMEM

I comandi BASIC LOMEM e HIMEM servono per modificare la quantità di RAM disponibile per i programmi BASIC. LOMEM definisce l'estremo inferiore e HIMEM l'estremo superiore dell'area di memoria ad essi dedicata.

I valori di default di queste impostazioni sono: LOMEM = **0x0800**, HIMEM = **0x1000**.

Questo limita la RAM a disposizione dell'utente a:

$0x1000-0x0800=0x0800$. In decimale questo valore equivale a 2048 bytes, ossia 2 kBytes.

Il Programma di Salvataggio, tuttavia, **assume come valore di default per LOMEM l'indirizzo 0x0280** e pertanto salva anche il contenuto di tutta la memoria "inutilizzata" dal BASIC con le impostazioni di default.

Le ragioni sono qui di seguito elencate:

- Nell'architettura Apple-1 l'indirizzo 0x0280 (640 in decimale) è il primo indirizzo di RAM "libero" e non asservito al funzionamento stesso della macchina. Pertanto esso rappresenta il valore minimo assoluto ammissibile per il comando LOMEM.
- Dall'indirizzo 0x0280 all'indirizzo 0x0800 ci sono ben 1408 bytes a disposizione, che possono essere utilizzati sia dal BASIC che da altri programmi.
- Non è raro trovare programmi scritti in BASIC in cui il primo comando è LOMEM. Questo assicura la compatibilità di tali programmi con il sistema di salvataggio adottato.
- È altrettanto comune allocare nel range 0x0280..0x07FF le proprie routine in linguaggio macchina affinché siano poi richiamate dal BASIC con delle opportune CALL.
- Il sistema adottato garantisce, in questo caso, anche il salvataggio automatico e trasparente di tali routines, unitamente al programma BASIC vero e proprio.
- Non da ultimo: il salvataggio di tutto il range di memoria da 0x0280 a 0x0FFF (3456 bytes in totale, a cui si sommano altri puntatori afferenti al BASIC stesso necessari per eseguire i programmi) riduce la quantità di spazio non utilizzato sulla EEPROM.

Quanto esposto sopra è anche la ragione per cui il BLOCCO 0 non è utilizzabile per salvare programmi in BASIC: molto semplicemente esso non è grande abbastanza.

Il comando HIMEM lavora in maniera analoga, spostando verso l'alto il limite della memoria utilizzabile dal programma BASIC.

Se si desiderasse scrivere un programma BASIC molto lungo (ipotizziamo 6 kBytes totali) bisognerebbe dare per primo il comando HIMEM=8192 (0x2000 in esadecimale).

Dato che LOMEM non è stato toccato la RAM a disposizione per il programma sarebbe $0x2000-0x0800=0x1800$, equivalenti a 6kBytes, e questo soddisferebbe le nostre esigenze.

Ma come possiamo salvare il nostro programma BASIC di 6 kBytes se le dimensioni dei Blocchi dedicati al BASIC sono di soli 4 kBytes? La soluzione è usare **un Blocco da 4 kBytes per salvare una parte del programma e un secondo Blocco da 4 kBytes per il resto**.

La prima parte del programma (da 0x0280 a 0x0FFF) andrà dunque salvata come programma BASIC tradizionale (anche se ovviamente incompleto) con il comando **S**.

La seconda parte del programma (da 0x1000 a 0x1FFF) andrà salvata con il comando **W** (specificando naturalmente come indirizzo di partenza la locazione \$1000). Questo salverà i rimanenti 4 kBytes.

Per ricaricare il programma in maniera corretta, sarà ovviamente necessario caricare entrambe le parti.

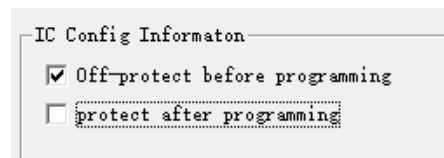
4. LIMITAZIONI E AVVERTENZE

Il salvataggio su EEPROM è da considerarsi sperimentale, tuttavia consente di salvare i propri lavori con relativa sicurezza e, soprattutto, di poterli ricaricare in maniera istantanea.

Non ci assumiamo alcuna responsabilità, ovviamente, per danni, anche gravi o letali, causati a persone / cose / proprietà intellettuali durante l'installazione o l'utilizzo di questo dispositivo.

A questo proposito:

- **Tutti i Blocchi sono riscrivibili** decine di migliaia di volte, ma non esiste la funzione di UNDELETE. Una volta che un Blocco (o un singolo byte) è stato riscritto, il contenuto precedente è *perso per sempre*.
- All'interno del Programma di Salvataggio non c'è un tasto o una funzione "Annulla Operazione". Se fate degli errori durante l'inserimento di nomi/blocchi/indirizzi, e la scrittura su EEPROM non è ancora iniziata, l'unico modo di interrompere il Programma di Salvataggio è usare il tasto RESET e iniziare da capo.
- Il posizionamento del ponticello **nella posizione RW rende riscrivibile TUTTA la EEPROM**, non soltanto i Blocchi sopraelencati. Questo significa che impartendo da WOZ Monitor un comando di scrittura afferente al range di indirizzi 0x4000 .. 0xBFFF il contenuto della EEPROM verrà modificato.
- Se i dati modificati erroneamente afferiscono al BASIC, o ai Programmi di Gestione, o peggio ancora ai vostri programmi, il computer potrebbe funzionare male e voi potreste perdere dei dati preziosi.
- Per tale ragione usate sempre molta cautela con i comandi del WOZ Monitor quando la modalità di scrittura è abilitata.
- Il produttore della EEPROM dichiara che i contenuti che scrivete sulla EEPROM non degraderanno per *almeno dieci anni*. Per un computer che ha più di quarant'anni non sono poi molti... quindi: **FATE DEI BACKUP!** I backup saranno anche la vostra assicurazione per le scritture accidentali.
- Se doveste decidere di programmare voi stessi una 28c256 assicuratevi di lasciare la protezione in scrittura DISABILITATA al termine dell'operazione. Nel programmatore TL-866, ad esempio, la scelta si opera deselezionando la casella *protect after programming*. Riferitevi sempre al manuale del vostro programmatore di EEPROM.
- Se la EEPROM risulta protetta in scrittura tramite la *feature* sopra descritta (oppure perché non avete spostato il ponticello in posizione RW) il Programma di Scrittura si bloccherà. La sequenza di punti che normalmente indica che la scrittura è in corso non comparirà e l'unica operazione possibile sarà un RESET.



Ci rendiamo conto che le funzioni descritte in questo manuale sono spartane, ma abbiamo preferito privilegiare la funzionalità rispetto alla forma, con particolare attenzione a massimizzare la memoria disponibile per voi.

5. API

La routine di copia dalla RAM alla EEPROM (e viceversa) viene allocata dal Programma di Scrittura tra gli indirizzi 0x023A e 0x027F (al termine del keyboard buffer, per ottimizzare l'uso della RAM).

È possibile usarla al di fuori dei programmi di gestione per copiare segmenti arbitrari di memoria.

La routine è disponibile dopo avere lanciato il Programma di Scrittura almeno una volta (ad es. **B800R** {ENTER} seguito da **X** oppure RESET).

Sei bytes di Pagina Zero definiscono gli indirizzi di partenza, di destinazione e la quantità di bytes da copiare:

\$40-\$41: indirizzo sorgente (es. 0x0280)
\$42-\$43: indirizzo di destinazione (es. 0xA000)
\$44-\$45: numero di bytes da copiare (es. 512 bytes, ossia 0x0200 in esadecimale)

Mediante WOZ Monitor inserite gli indirizzi desiderati (in formato Little Endian, secondo gli esempi di cui sopra):

```
40 : 80 02 00 A0 00 02 {ENTER}
```

Eeguire ora la routine di copia con:

```
23AR {ENTER}
```

Comparirà la solita sequenza di punti fino al termine della copia:

```
023A : A0 . .
```

La routine di copia non è stata pensata per essere lanciata dal WOZ monitor, ma da un programma. È pertanto possibile che al termine della copia la routine rimanga “appesa” e non rilasci il terminale. In questo caso premete il tasto RESET.

Le cause di questo comportamento sono molteplici e non sono oggetto di trattazione in questo documento, tuttavia esse non sono indice di malfunzionamento e solitamente non inficiano l’esito della copia.

Inoltre, dato che la routine è allocata al termine del buffer di tastiera, è altresì possibile che vengano stampati degli extra caratteri: ignorateli.

La routine di copia funziona in entrambi le direzioni: RAM→EEPROM e EEPROM→RAM, basta impostare opportunamente i puntatori di Pagina Zero.

Tutti i programmi salvati con questo procedimento risulteranno “invisibili” al Programma di Gestione.

Se desiderate avvalervi di tale metodologia di salvataggio vi consigliamo dunque di dedicare un intero Blocco a questa modalità in accordo alla mappa descritta nel paragrafo 2.

Appuntatevi poi tutte le informazioni necessarie al recupero dei vostri programmi.

Buona programmazione con *Apple-1 Juke-Box!*

APPLE-1 JUKE-BOX

INFO | ORDINI | SUPPORTO: p-l4b@protonmail.com