

# APPLE-1 microSD STORAGE CARD

SD CARD OS 1.1 – PCB v1.13/1.16

*Apple-1 microSD Storage Card* è una scheda elettronica che permette al vostro computer Apple-1, Originale o Replica, di accedere ai files contenuti sulla memoria rimovibile di tipo SD.

Essa garantisce nuove esperienze d'uso del vostro computer Apple-1 consentendovi di salvare i programmi scritti da voi, di aggiungerne altri con la massima facilità, e molte altre caratteristiche interessanti.

Questo documento contiene:

1. Contenuto della confezione
2. Rischio cariche elettrostatiche
3. Caratteristiche e descrizione del prodotto
4. Installazione della scheda sul computer Apple-1
5. Funzionamento
6. Configurazione jumpers
7. Note finali



L'immagine sopra riportata è puramente indicativa. Il prodotto finale potrebbe essere diverso a causa di migliorie e/o di disponibilità dei componenti.

## 1. CONTENUTO DELLA CONFEZIONE

- Scheda *Apple-1 microSD Storage* completa di SD card.

## 2. RISCHIO CARICHE ELETTROSTATICHE

*Apple-1 microSD Storage* è sensibile all'elettricità statica, come il vostro computer Apple, e potrebbe venirne danneggiata. Prima di qualsiasi operazione sul dispositivo è necessario scaricare l'elettricità statica accumulata dal vostro corpo e prevenirne un nuovo accumulo.

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

### 3. CARATTERISTICHE E DESCRIZIONE DEL PRODOTTO

*Apple-1 microSD Storage* è basata sul noto chip *Versatile Interface Adapter* (VIA) 65C22, che in abbinamento al microcontrollore ATMEGA 329P e ad un firmware appositamente scritto, gestisce le operazioni di lettura/scrittura sul filesystem della scheda di memoria.

L'interfaccia utente è di tipo testuale e richiama l'esperienza d'uso del **prompt comandi di MS-DOS o la CLI di Linux**. Anche i comandi a disposizione dell'utente sono molto simili, quando non uguali, a quelli dei sistemi operativi citati.

Sarà dunque possibile caricare / salvare / copiare ogni tipo di file. Sarà inoltre possibile creare e gestire directory e subdirectory a piacimento.

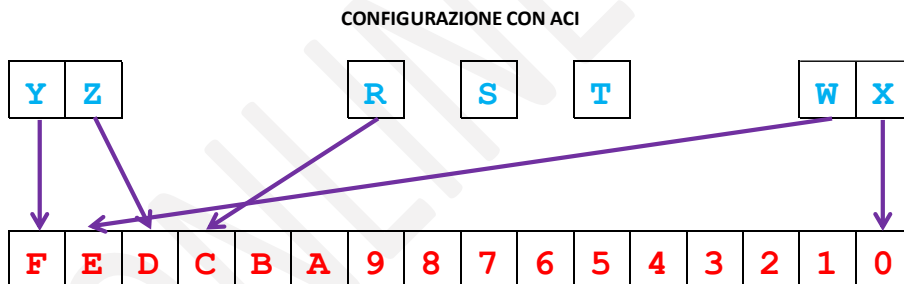
La scheda mette inoltre a disposizione una espansione di memoria che porta il computer ad avere **32 kB contigui di RAM, più i 4 kB dedicati al BASIC, per un totale di 36 kB**. Questa configurazione permette di far girare sostanzialmente ogni programma per Apple-1.

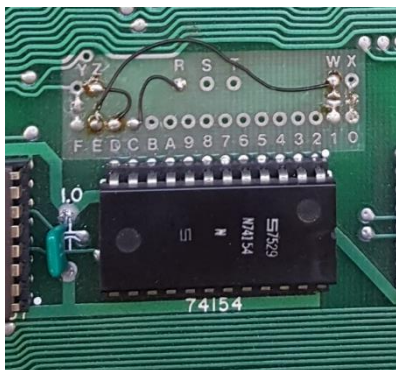
In caso di guasti, o per svolgere dei test, la scheda è anche in grado di rimpiazzare la memoria RAM presente sulla scheda madre, come verrà descritto nell'**Appendice 2 - Memory Replacement**.

### 4. INSTALLAZIONE DELLA SCHEDA SUL COMPUTER APPLE-1

#### 4.1 VERIFICHE PRELIMINARI

Il computer Apple-1 deve essere predisposto nella configurazione "CON ACI", ossia con i seguenti collegamenti in area CHIP SELECT:





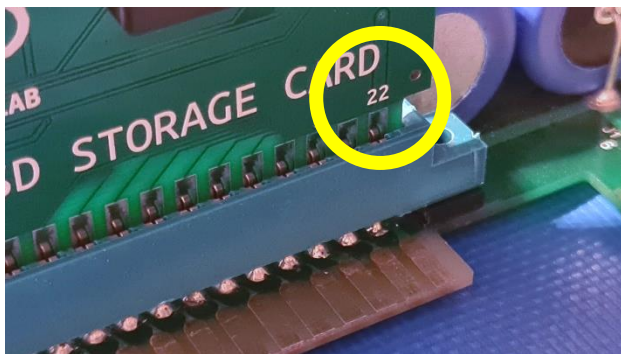
Lo schema e la foto riassumono i collegamenti necessari.

Questi collegamenti sono standard sulla maggior parte degli Apple-1 in circolazione, sia Originali che Repliche.

Non sono dunque necessari collegamenti dedicati.

## 4.2 INSTALLAZIONE DELLA SCHEDA

La scheda va inserita come riportato in figura:



Il numero "22" sulla scheda e il numero "22" sulla motherboard devono trovarsi sulla stessa estremità.

Il numero "22" sulla scheda deve dunque trovarsi rivolto verso il lato esterno della motherboard dell'Apple-1.

L'utilizzo della scheda in abbinamento ad altre schede e/o dispositivi tipo BUS EXTENDER potrebbe portare a malfunzionamenti: non utilizzate questo genere di dispositivi.

**ATTENZIONE:** L'accensione con la scheda orientata non correttamente **danneggia istantaneamente** il computer e la scheda stessa.

Il microcontrollore è dotato di porta USB che può essere usata come debug.

Per prevenire qualsiasi danno il cavo USB deve essere collegato a dispositivi esterni **solo quando il computer è acceso**. Per nessuna ragione un dispositivo esterno deve fornire alimentazione al microcontrollore quando il computer Apple-1 è spento.

## 5. FUNZIONAMENTO

---

All'accensione del computer Apple-1 premete come di consueto i tasti CLEAR SCREEN e RESET.

L'*entry point* del programma di interfaccia utente è \$8000, pertanto esso andrà lanciato con il comando:

**8000R** e premere ENTER (da qui in avanti rappresentato con l'indicazione {ENTER})

Comparirà immediatamente il prompt comandi dell'ambiente **SD CARD OS**:

```
8000: A9
*** SD CARD OS 1.1
/>
```

A questo punto sarà possibile impartire i comandi desiderati per sfogliare le cartelle, muoversi tra di esse, caricare files, etc. Prima del prompt > verrà visualizzata la directory corrente (> è la directory principale). Se necessario, sono disponibili due livelli di aiuto. Il primo è:

**?** {ENTER}

a seguito del quale comparirà immediatamente la lista di tutti i comandi disponibili:

```
COMMANDS LIST:
READ, WRITE, DIR, TIME, LOAD, RUN, SAVE, TYPE, DUMP,
ASAVE, BAS, DEL, LS, CD, MKDIR, RMDIR, RM, MD, RD, PWD,
TEST, HELP, ?, MOUNT, EXIT.

USE HELP COMMAND FOR MORE DETAILS
```

Il secondo livello consiste di richiamare la sintassi di ogni comando disponibile, digitando la parola **HELP** seguita dal comando per il quale si necessita di aiuto. Ad esempio:

**HELP SAVE** {ENTER}

Comparirà subito l'aiuto desiderato:

```
SYNTAX:
SAVE FILENAME [START] [END]

SAVES A FILE TO THE SD CARD. IF START AND END ARE
SPECIFIED, A BINARY FILE WITH TAG #06 WILL BE CREATED
WITH THE MEMORY CONTENT FROM THE ADDRESS RANGE START-END
(INCLUDED). IF START AND END ARE NOT SPECIFIED, THE
BASIC PROGRAM CURRENTLY LOADED IN MEMORY WILL BE CREATED
WITH THE CORRESPONDING #F1 TAG.
```

Il comando DIR visualizza tutti i files presenti nella directory corrente, corredati da altre informazioni qui riassunte:

```

/BASIC>DIR {ENTER}
FACTORIALS      32767  BAS  $0200
WORDSEARCH     16128  BAS  $0300
BASIC           4096  BIN  $E000
CONCENTRATION  16128  BAS  $0300
PRIMEFINDER     2560  BAS  $0800
/BASIC>

```

↓                      ↓                      ↓  
Dimensione in bytes    File type    Start Address

La visualizzazione della directory può essere **temporaneamente messa in pausa** tramite la pressione di qualunque tasto (tranne ESC).

Per riprendere la visualizzazione dei file rimanenti basterà premere il tasto **ENTER**.

Per interrompere la visualizzazione prima che tutti i files siano stati visualizzati premere il tasto **ESC**.

È stata implementata anche la **funzionalità di BACKSPACE**: premendo il tasto underscore “\_” durante una qualunque digitazione si otterrà una nuova riga priva dell’ultimo carattere digitato. Questo consente di correggere eventuali errori di digitazione **senza dover riscrivere tutta la riga**.

Esempio:

```

BASIC>LOAD CONCENTARTI    premere il tasto underscore _
>LOAD CONCENTART           premere il tasto underscore _
>LOAD CONCENTAR           premere il tasto underscore _
>LOAD CONCENTA            premere il tasto underscore _
>LOAD CONCENT             continuare a scrivere normalmente
>LOAD CONCENTRATION

```

*Nota Bene: queste funzioni potrebbero non essere disponibili su alcune tastiere.*

Altri comandi sono stati pensati per **velocizzare e facilitare il caricamento e l’esecuzione** dei programmi. Uno di questi è il comando RUN che permette in caricamento e l’esecuzione di programmi BASIC o binari senza dovere rientrare esplicitamente nell’ambiente BASIC o passare dal WOZ monitor.

Un’altra caratteristica importante di **SD CARD OS** è quella di ricercare un file anche se il suo nome non risulta scritto per intero.

Ad esempio, per eseguire direttamente il programma BASIC “Star Trek” si dovranno semplicemente digitare i comandi evidenziati in rosso:

```

/> CD BASIC {ENTER}                      ← cambia directory in BASIC
/BASIC> LOAD BASIC                      ← carica il programma BASIC
FOUND BASIC#06E000                      ← SD CARD OS ha trovato il file con il nome più simile

```

LOADING

← il file viene caricato

```
BASIC#06E000
E000.EFFF (4096 BYTES)
OK
/BASIC> RUN STARTR {ENTER}
FOUND STARTREK#F10300
LOADING
```

← il caricamento è terminato  
← viene mostrata l'area di RAM dove il programma è stato caricato  
← fine operazioni  
← carica ed esegue il programma col nome più simile a STARTR  
← SD CARD OS trova il primo file il cui nome inizia con STARTR  
← il file viene caricato

```
STARTREK#F10300
(LOMEM=$0300 HIMEM=$1000)
OK
TYPE A NUMBER ?
```

← il caricamento è terminato  
← vengono dichiarati i valori HIMEM e LOMEM usati dal programma  
← fine operazioni  
← il programma STAR TREK è ora in esecuzione

La funzionalità di caricamento con il nome scritto parzialmente è disponibile per i soli comandi LOAD e RUN.

Se desiderate salvare i vostri programmi scritti in INTEGER BASIC sarà necessario uscire al WOZ monitor mediante il tasto RESET. Dopodiché sarà necessario lanciare il programma di gestione con 8000R e usare il comando SAVE:

```
8000R {ENTER}
8000: A9
*** SD CARD OS 1.1
/>SAVE PROVA {ENTER}
SAVING
PROVA#F10800
(LOMEM=$0800 HIMEM=$1000)
OK
/>
```

Il vostro programma verrà **salvato nella directory corrente** (la directory principale ✓ nel caso in esempio). Verranno salvati anche i puntatori HIMEM e LOMEM. Nell'esempio sono stati riportati i valori di default.

Se desiderate verificare che il vostro programma sia stato salvato potete impartire ora i comandi LS oppure DIR, in modo da apprezzare la diversa nomenclatura visualizzata:

```
/>LS {ENTER}
2560 PROVA#F10800

/>DIR {ENTER}
PROVA 2560 BAS $0800
```

Il vero nome del file è visualizzato dal comando LS, il comando DIR lo interpreta in modo da essere leggibile più facilmente. Nell'**Appendice 1** trovate la descrizione di questo tipo di nomenclatura.

*Nota Bene:* i comandi di cancellazione RM e DEL necessitano del vero nome del file.

Per continuare a lavorare sul vostro programma: uscite al WOZ monitor mediante la pressione del tasto RESET o con il comando EXIT e rientrate nell'ambiente BASIC con il consueto comando E2B3R.

**SD CARD OS** mette a disposizione molti altri comandi che vi permetteranno di salvare su SD CARD non solo i vostri programmi scritti in INTEGER BASIC, ma anche quelli binari.

Oltre a ciò, è stata mantenuta la compatibilità con **AppleSoft Basic Lite**, che permette di scrivere ed eseguire programmi più complessi di quelli gestibili da INTEGER BASIC. Tutti i dettagli sono descritti nell'**Appendice 3** di questo documento.

I firmware - sia dell'EEPROM a bordo della scheda sia del microcontrollore - sono stati messi a disposizione dal loro creatore, **Antonino Porcino**, sul seguente repository GitHub:

<https://github.com/nippur72/apple1-sdcard>

L'elenco dei comandi e della loro sintassi vengono per comodità riportati anche nell'**Appendice 1** di questo documento.

La scheda di memoria SD può essere rimossa in sicurezza in qualunque momento, a patto che non ci siano operazioni di lettura/scrittura in corso.

A valle della sostituzione della scheda è necessario **premere il tasto RESET del microcontrollore** (e non del computer) affinché essa venga letta correttamente da **SD CARD OS**.

In alternativa al RESET potete usare il comando MOUNT.

Tali operazioni non hanno effetto sul contenuto della RAM del computer né sul suo funzionamento.

In caso di messaggi di errore (ad. Es. I/O ERROR) durante le operazioni: uscire dal programma di gestione mediante il tasto RESET e rientrare con 8000R.



## 6. CONFIGURAZIONE JUMPERS / CONNETTORI DI ESPANSIONE / IRQ

La configurazione della scheda è affidata ad alcuni jumper/pad il cui funzionamento è qui di seguito descritto.

Sono presenti anche due connettori ausiliari che rendono disponibili le linee di I/O inutilizzate del chip 65C22 e del microcontrollore: potrete dunque usarle per i vostri progetti, pur continuando a usufruire di tutti i servizi offerti dalla scheda stessa.

### 6.1 ALLOCAZIONE DELLA MEMORIA RAM ESPANSA



Normalmente questi due jumper sono inseriti nella loro posizione perché consentono di realizzare la contiguità della memoria RAM dall'indirizzo \$0000 all'indirizzo \$7FFF.

Rimuovendo questi jumper, indipendenti tra loro, è possibile creare delle discontinuità nella mappatura come qui specificato:

#### 1000-1FFF ENABLE JUMPER

Rimuovendo questo jumper la scheda non gestirà più la RAM tra gli indirizzi \$1000 e \$1FFF (4 kB totali).

#### 16/32K JUMPER

Rimuovendo questo jumper la scheda limiterà l'espansione di RAM ai primi 16 kB, ossia fino all'indirizzo \$3FFF. Gli indirizzi da \$4000 a \$7FFF risulteranno quindi non allocati e a disposizione di altri dispositivi/periferiche.

#### 2000-3FFF ENABLE JUMPER

**(solo PCB versione 1.16)** Rimuovendo questo jumper la scheda non gestirà più la RAM tra gli indirizzi \$2000 e \$3FFF (8 kB totali)

### 6.2 CONNETTORI DI ESPANSIONE

Il connettore **J1** riporta le porte di I/O inutilizzate afferenti al 65C22:

PIN	1	2	3	4	5	6	7	8	9	10	11	12
PORT	GND	N/C	PB2	PB3	PB4	PB5	PB6	CA1	CA2	CB1	CB2	+5V



(il PIN 1 è di forma quadrata, e si trova a sinistra)



Il connettore **J2** riporta le linee di I/O inutilizzate afferenti al microcontroller:

PIN	PORT
1	GND
2	A7
3	A6
4	A5
5	A4
6	A3
7	A2*
8	+5V



(Il PIN 1 è quello di forma quadrata, posizionato in alto)

\* La linea A2 è collegata anche alla linea PB1 del 65C22. Tale collegamento non è attualmente utilizzato dal firmware ma è stata cablata per usi futuri. Usare con cautela per evitare danni al 65C22 o al microcontrollore.

*Nota Bene* Le attestazioni delle linee 65C22<->microcontrollore in uso sono dichiarate nei sorgenti del software. Se per i vostri progetti personali basati sul 65C22 necessitate anche di tali linee: rimuovete microcontrollore e lettore di schede SD e usate i rispettivi connettori per i vostri collegamenti.

### 6.3 IRQ



È stata prevista una linea dedicata per la linea IRQ del 65C22 verso il 6502 dell'Apple-1. Collegare tra di loro le piazzole indicate nell'immagine a fianco se per i vostri progetti desiderate sfruttare la caratteristica del 65C22 di generare interrupt (ad esempio: timer, etc.)

Se desiderate scrivere dei programmi che facciano uso del 65C22, il base address del dispositivo è \$A000.

Se necessitate delle linee attualmente afferenti al microcontrollore o al lettore di SD card, questi ultimi sono rimovibili.

Unitamente ai connettori J1 e J2 sopra descritti disporrete dunque della totalità delle linee di I/O del 65C22.

Se invece necessitate di spazio EEPROM, il range di indirizzi ad essa dedicati è da \$8000 a \$9FFF, per un totale di 8 kB.

## 7. NOTE FINALI

---

La scheda di memoria SD va obbligatoriamente formattata secondo lo standard **FAT32**.

Se il vostro Sistema Operativo non prevede tale opzione è possibile usare un qualunque programma esterno in grado di farlo: lo standard è ben documentato e ci sono moltissime utility di terze parti in grado di fare questa operazione.

Per movimentare/copiare/rinominare i files non sono necessari tool particolari, un qualunque file browser funzionerà senza problemi.

Se aggiornate i files della SD Card con un computer esterno abbiate cura di rimuovere manualmente eventuali directory "di servizio" create automaticamente dal vostro Sistema Operativo, tipo: *System information / Deleted Items*, etc. Esse potrebbero inficiare la visualizzazione dei contenuti sullo schermo.

La nomenclatura *tagged* dei files, meglio descritta nell'**Appendice 1** di questo documento, è conforme allo standard PRODOS/CFFA1. È pertanto possibile trasferire files tra le due architetture senza dover manipolare o convertire i files.

Le schede di memoria SD, in generale, **non sono dispositivi di memorizzazione permanente**.

È indispensabile, pertanto, fare sempre una copia di riserva dei contenuti della scheda di memoria per evitare perdite accidentali di dati.

Il prodotto è in continua evoluzione: visitate spesso la pagina di riferimento:

<https://p-l4b.github.io/sdcard>

per essere sempre informati sulle novità e sugli aggiornamenti!

---

*Ci auguriamo che tu possa divertirti ad utilizzare Apple-1 microSD Storage Card !!*

**APPLE-1 microSD  
STORAGE CARD**

INFO | ORDINI | SUPPORTO: [p-l4b @ protonmail.com](mailto:p-l4b@protonmail.com)

**P-L4B @ PROTONMAIL.COM**

## SD CARD

- Se non specificato diversamente, tutti i numeri sono espressi in esadecimale
- Gli argomenti tra parentesi [ ] sono opzionali
- Sono permessi percorsi annidati, mediante il carattere / ad esempio: /, /folder1/foo
- Se non viene specificato nessun path si intende quello corrente

## TAGGED FILE NAMES

I nomi dei files contengono il carattere di tag # che ha un significato speciale: la parte dopo # indica il tipo di file (due caratteri) e l'indirizzo RAM di caricamento in esadecimale (4 caratteri).

**#06** per files binari

**#F1** per programmi INTEGER BASIC

**#F8** per programmi AppleSoft BASIC

Esempio:

**BASIC#06E000** è un file binario chiamato **BASIC** che va caricato a partire dall'indirizzo **\$E000**.

**STARTREK#F10300** è un programma **BASIC** chiamato **STARTREK** che va caricato dall'indirizzo **\$0300**.

I comandi LOAD, RUN, SAVE e DIR usano queste informazioni supplementari per facilitare le operazioni.

Per esempio, per caricare in memoria ed eseguire i due files sopra riportati sarà sufficiente dare:

LOAD BASIC

RUN STARTREK

## COMANDI

### **READ filename startaddress**

Legge un file binario dalla SD card e lo carica in memoria a partire dall'indirizzo specificato.

### **WRITE filename startaddress endaddress**

Scrive il segmento di memoria tra startaddress ed endaddress (incluso) su un file della SD card.

### **TYPE filename**

Legge il file ASCII indicato dalla SD card e lo visualizza sullo schermo. Premere un tasto qualsiasi per interrompere la visualizzazione e premere ENTER per continuarla. Premere ESC per tornare al prompt comandi.

**DUMP filename [start] [end]**

Legge in file indicato dalla SD card e lo visualizza sullo schermo in formato esadecimale.

start ed end sono opzionali e possono essere usati per visualizzare solo una porzione del file. Premere un tasto qualsiasi per interrompere la visualizzazione e premere ENTER per continuarla. Premere ESC per tornare al prompt comandi.

**LOAD filename**

Carica un file dalla SD card. filename si riferisce a un "tagged file name" sopra menzionato. Per comodità, filename può essere solo una parte del nome completo del file: verrà caricato il primo file corrispondente.

**SAVE filename [start] [end]**

Salva un file sulla SD card.

Se start ed end sono presenti verrà creato un file binario con il tag #06 contenente quanto presente in RAM nell'intervallo start-end (inclusi).

Se start ed end non sono indicati, verrà creato un file con il tag #F1 contenente il programma INTEGER BASIC attualmente presente in memoria.

**ASAVE filename**

Salva il programma AppleSoft BASIC attualmente in memoria sulla SD card, con tag #F8.

**RUN filename**

Come LOAD ma esegue il file al termine del caricamento. I files binari sono eseguiti dall'indirizzo esadecimale espresso dal tag; i programmi BASIC sono automaticamente lanciati dall'interprete BASIC.

**DEL filename RM filename**

Cancella un file dalla SD card.

**DIR [path] / LS [path]**

Elenca i files presenti nella directory indicata, oppure nella directory corrente se tale indicazione non viene data. LS ha un formato conciso e veloce. Premere un tasto qualsiasi per interrompere la visualizzazione e premere ENTER per continuarla. Premere ESC per tornare al prompt comandi.

**CD path**

Cambia la directory di lavoro. La directory corrente è inoltre visualizzata prima del prompt.

**MD path / MKDIR path**

Crea la directory indicata.

**PWD**

Visualizza sullo schermo la directory corrente.

**MOUNT**

Forza la rilettura della scheda SD dopo che è stata rimossa e reinserita.

**RD path / RMDIR path**

Cancella la directory indicata. La directory da cancellare deve essere vuota (nessun file o sottodirectory all'interno).

**BAS**

Visualizza i puntatori LOMEM e HIMEM relativi al programma BASIC attualmente in memoria.

**address R**

Esegue il programma in memoria dall'indirizzo specificato. Indirizzi utili: 6000R AppleSoft BASIC *cold start* (necessario almeno una volta dopo il caricamento), 6003R AppleSoft BASIC *warm start* (non cancella il programma già presente in RAM), E000R Integer BASIC *cold start*, EFECR Integer BASIC "RUN" (può essere usato come *warm entry point*), 8000R SD CARD OS command prompt.

**TIME value**

Modifica il valore di Timeout per le operazioni di lettura/scrittura sulla SD card. Solo per test, non modificare.

**TEST**

Test interno.

**?**

Mostra a schermo l'elenco dei comandi disponibili.

**HELP command**

Mostra l'help per il comando indicato.

**EXIT**

Uscita al WOZ Monitor.

ONLY

## APPENDICE 2 – MEMORY REPLACEMENT

---

Questa funzione può risultare utile in caso di test, guasti o indisponibilità di componenti.

Le operazioni qui di seguito descritte sono da considerarsi temporanee ed eseguite a proprio rischio e pericolo.

*È indispensabile mettere in atto tutte le procedure di sicurezza contro l'accumulo di cariche elettrostatiche già descritte in questo documento.*

Apple-1 può avere fino a 8 kBytes di RAM sulla mainboard principale. Essi sono organizzati in due "banchi" da 4 kBytes aventi otto chip ciascuno, per un totale di sedici chip.

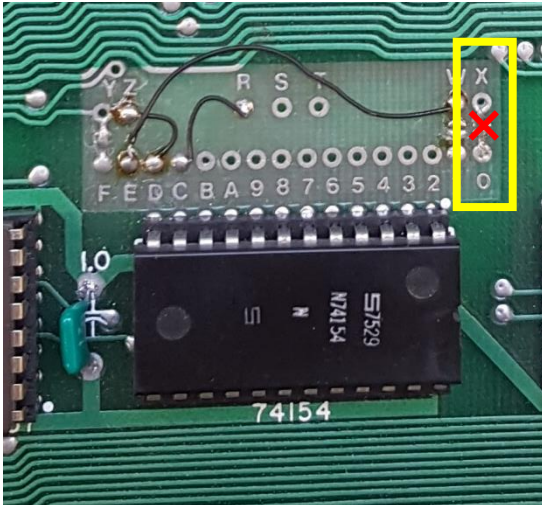
In accordo alla configurazione "con ACI" descritta nel Manuale Operativo del computer, e indicata in questo manuale, essi sono così suddivisi:

- Banco "O", indirizzi da \$0000 a \$0FFF, chip dalla posizione B11 alla B18.
- Banco "E", indirizzi da \$E000 a \$EFFF, chip dalla posizione A11 alla A18.

Seguire le istruzioni dai punti 7.1 o 7.2 (o entrambi) in funzione del banco che si desidera rimpiazzare.

## 7.1. SOSTITUZIONE DEL BANCO DI MEMORIA "0" (INDIRIZZI DA \$0000 A \$0FFF)

**7.1.1** Su Apple-1: aprire/dissaldare il collegamento tra la piazzola "X" e la piazzola "0" qui raffigurato nel quadrato giallo in figura:



Non deve rimanere alcun collegamento elettrico tra le due piazzole.

**7.1.2** Rimuovere tutti gli otto i chip di memoria dalla posizione B11 alla posizione B18.

**7.1.3** Cortocircuitare con un jumper o con in collegamento temporaneo le piazzole evidenziate in giallo:



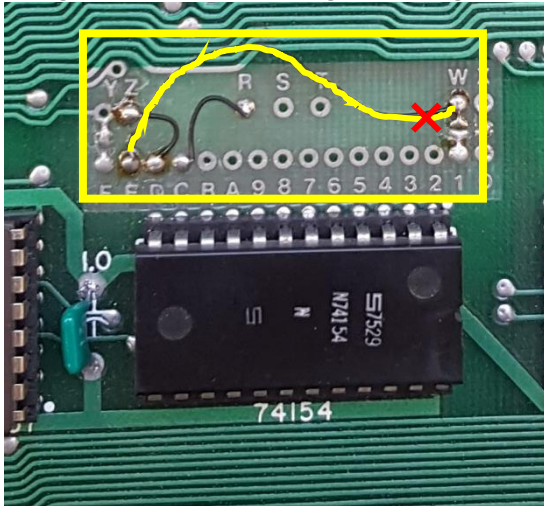
**7.1.4** Accendere il computer e operare normalmente.

A fine lavori ripristinare i collegamenti originali, rimettere i chip di memoria al loro posto e rimuovere il collegamento temporaneo sulla *Apple-1 microSD Storage Card*.



## 7.2. SOSTITUZIONE DEL BANCO DI MEMORIA "E" (INDIRIZZI DA \$E000 A \$EFFF)

**7.2.1** Su Apple-1: aprire/dissaldare il collegamento tra la piazzola "W" e la piazzola "E" qui raffigurato nel quadrato giallo in figura:



Solitamente il collegamento è effettuato con un sottile cavetto.

Dissaldarlo in prossimità della piazzola "W" e isolare l'estremità appena dissaldata del cavetto, per assicurarsi che non faccia contatti accidentali con altri componenti.

**7.2.2** Rimuovere tutti gli otto i chip di memoria dalla posizione A11 alla posizione A18.

**7.2.3** Cortocircuitare con un jumper o con in collegamento temporaneo le piazzole evidenziate in giallo:



**7.2.4** Accendere il computer e operare normalmente.

*Nota Bene:*

In questa particolare configurazione il segmento di memoria da \$6000 a \$6FFF non risulterà utilizzabile. In configurazione "32K", dunque, la massima dimensione di memoria contigua sarà limitata a 24 kBytes (indirizzi da \$0000 a \$5FFF) anziché 32 kBytes. Questo impedirà la corretta esecuzione dei programmi di lunghezza superiore a 24576 bytes. La configurazione "16K" invece, opererà normalmente.

A fine lavori ripristinare i collegamenti originali, rimettere i chip di memoria al loro posto e rimuovere il collegamento temporaneo sulla *Apple-1 microSD Storage Card*.

## APPENDICE 3 – APPLESOFT BASIC LITE

---

AppleSoft BASIC è una variante del Microsoft BASIC (da cui il nome) che venne sviluppata nel 1977 come sostituto dell'INTEGER BASIC per i computer Apple II.

Essa dispone di un maggior numero di comandi e istruzioni ma soprattutto è in grado di gestire i numeri in virgola mobile. Le funzioni grafiche non sono ovviamente disponibili.

Per il progetto "*Replica-1*" ne è stata realizzata una versione modificata appositamente, reperibile qui:

<https://cowgod.org/replica1/applesoft/>

Per garantirne il funzionamento anche con *Apple-1 microSD Storage Card* è stata dunque realizzata una versione apposita, in grado di mettere a disposizione i comandi LOAD/SAVE/MENU come sulla versione originale, pur con qualche differenza.

Sulla SD card è stata predisposta una cartella apposita, chiamata ASOFT, al suo interno troverete l'interprete AppleSoft BASIC e un programma di esempio.

### AVVIO DI APPLESOFT BASIC LITE

Analogamente al tradizionale INTEGER BASIC, AppleSoft BASIC va **caricato ed eseguito per primo almeno una volta**. In questo esempio si assume di essere già all'interno del programma di gestione, i comandi da digitare sono in rosso:

```
*** SD CARD OS 1.1
/>>CD ASOFT {ENTER}
/ASOFT>RUN APPLESOFT {ENTER}
FOUND APPLESOFT-SD#066000
LOADING

APPLESOFT-SD#066000
$6000-$7FFF (8192 BYTES)
OK

*** APPLESOFT BASIC LITE SD V1.1 ***
22525 BYTES FREE

└
```

Adesso è possibile scrivere i propri programmi in AppleSoft BASIC.

## EDITOR DI LINEA

E' stato implementato un **editor di linea** che mediante, la pressione del tasto BACKSPACE (Control H), vi consente di **cancellare l'ultimo carattere digitato**:

```
] 10 PRINT "THIS IS A TEST" → premere BACKSPACE o Control H)
] 10 PRINT "THIS IS A TES" → premere BACKSPACE o Control H)
] 10 PRINT "THIS IS A TE" → premere BACKSPACE o Control H)
] 10 PRINT "THIS IS A T" etc.
```

*Nota Bene: questa funzione potrebbe non essere disponibile su alcune tastiere.*

## CARICAMENTO DEI PROGRAMMI

Per caricare un programma da SD card, sovrascrivendo quello corrente in memoria, basterà impartire dal prompt di AppleSoft BASIC il comando LOAD seguito dal nome del file desiderato tra doppi apici, ad es.:

```
]LOAD "LEMO" {ENTER}
FOUND LEMO#F8801
LOADING
LEMO#F8801
$0801-$23F8 (7160 BYTES)
OK
]RUN {ENTER}
```

## SALVATAGGIO DEI PROGRAMMI

Per salvare i vostri programmi potete usare il comando SAVE. La sintassi è analoga al comando LOAD:

```
]SAVE "MYPROG" {ENTER}
SAVING
MYPROG#F80801:
$0801-$080C (12 BYTES) → l'indirizzo finale e dimensione saranno quelli relativi al vostro programma
]
```

## INTEGRAZIONE CON SD CARD OS

È possibile entrare e uscire dal menu principale di SD CARD OS direttamente da AppleSoft BASIC, usando il comando MENU:

```
┌ MENU {ENTER}
```

```
*** SD CARD OS 1.1
```

```
└ASOFT┐
```

Anche dall'ambiente SD CARD OS potete caricare e salvare i vostri programmi AppleSoft BASIC, **avendo l'accortezza di usare il comando **ASAVE – AppleSoft SAVE**** - e non il comando SAVE.

Il comando SAVE di SD CARD OS è dedicato esclusivamente al salvataggio dei programmi INTEGER BASIC.

È altresì necessario rientrare nell'ambiente AppleSoft BASIC mediante il *warm entry point* **6003**, pena la cancellazione del programma attualmente in memoria:

```
└ASOFT┐ 6003R {ENTER}
```

```
┌
```

Come si vede in questo esempio, il *warm-start* di AppleSoft BASIC tramite l'indirizzo 6003 non causa la visualizzazione della scritta di benvenuto.

I programmi salvati con AppleSoft BASIC avranno tag #F8 ed estensione "ASB" (AppleSoft BASIC).

Suggeriamo di fare pratica con i vari menu e comandi di caricamento e salvataggio per evitare di incorrere in perdite di dati.