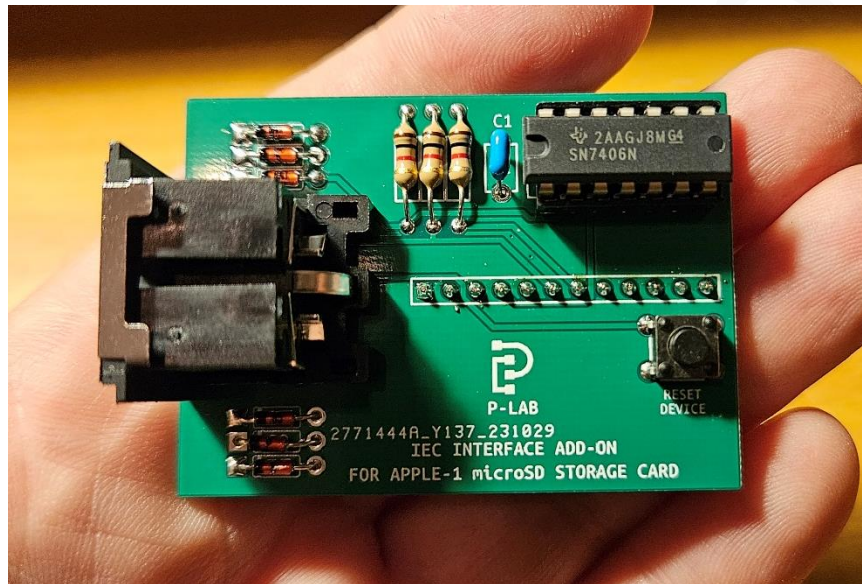


IEC add-on for APPLE-1 microSD STORAGE CARD

SD CARD OS 1.3 – PCB all versions



This small and inexpensive expansion card for your *Apple-1 microSD Storage Card* (<https://p-lab.github.io/sdcard/>) allows you to connect **one or more Commodore Disk Drives or emulators** and integrate them seamlessly into your Apple-1 system.

You will then be able to read and save your programs on the 5"1/4 disks, as well as copy them to the SD-CARD, and vice versa.

To work properly, the expansion requires an upgrade of the Operating System to version 1.3 or later, the details will be described in more detail in this document.

This project is released under a Creative Commons CC BY 4.0 license.

<https://creativecommons.org/licenses/by/4.0/deed>

1. DAMAGES FROM ELETROSTATIC DISCHARGES

Before handling the boards put in place all precautions against static electricity build-up, as already described in the *Apple-1 microSD Storage Card Reference Document*.

2. FLOPPY DISK COMPATIBILITY

The *IEC Interface* and the SD OS 1.3 Operating System are **generally compatible with any Commodore or compatible Disk Drive** based on the IEC Serial Bus (the classic six-pin circular connector) such as, for example, models 1541 and 1541/II.

Even "cycle exact" devices based on Raspberry PI have been found to be perfectly compatible. For the **complete and updated list** on brands/models of devices found compatible please refer to the project site:

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

Every contribution is welcome!

Please feel free to report any compatible (or not) devices to us at the e-mail address at the bottom of this document.

This project was developed with Disk Drives that weren't somehow modified to increase their speed. Compatibility any kind of modification should be guaranteed if a "basic" mode of operation is also offered in addition to the "turbo" mode.

Your contribution is precious: help us compile our grid of compatible devices, the Apple-1 community will be grateful!

As in the Commodore architecture, up to four Disk Drives can be concatenated, numbered as usual from 8 to 11.

3. BEFORE WE START: DISK DRIVE HEALTH CHECK

Disk Drives are very delicate devices with mechanical and electronic parts that may no longer function properly after several decades of operation (or inactivity).

Before connecting the Drive to your Apple-1 system, it is therefore essential to check on another system that it is working properly in read and write mode.

Make sure that the diskettes you will be using are in excellent condition and possibly already tested on other systems. Often, even if NOS (New Old Stock) the magnetic surface may be degraded.

4. SUBSTITUTION / EEPROM FIRMWARE UPGRADE

Although the EEPROM is a device that can be rewritten relatively easily by the end user, it is possible that the model on board your Apple-1 microSD Storage Card may not be compatible with version 1.3 of SD OS that is the subject of this document.

In fact, we have found that under certain circumstances and with certain Apple-1 chipsets, problems can arise when writing to disk, especially long files.

It is a well-known fact that each Apple-1 unit, Original or Replica, is different from all others and can be tolerant to deviations from the design values of its internal timings.

From the case studies analyzed and their distribution, it has been determined that **to achieve dependable long write operation with the IEC interface, the access time to the EEPROM data of the microSD Storage Card must not be less than 200 ns.**

Device speed is usually written down after the model of the EEPROM (28C64x) with a hyphen and a number (or something like 15PU-) that should usually be interpreted as follows:

28C64A-20 → 200 ns → the EEPROM is suitable for operation with the *IEC interface*

28C64B-15 → 150 ns → the EEPROM may cause some write malfunction with the *IEC interface*.

As a rule of thumb: if a "15" appears somewhere after the model, it is almost certainly a 150ns device.

However, combinations of PCB versions and components on your Apple-1 may also work with the faster EEPROMs.

We would also like to point out that replacement is recommended **only if you plan to use the IEC interface**. For conventional operations with SD-CARD memory, the EEPROM in your possession is perfectly suitable.

If you have any questions, before starting any upgrade or assembly activities, please contact us by attaching a clear, close-up photo of the EEPROM, it is the 28 pins Integrated Circuit located immediately below the SD Card reader.

Instead of the EEPROM, it is also possible to use a **27C64 EPROM**: if you have the possibility of programming one, you can safely replace the 28C64 with a 27C64. Access time must be less than 200 ns: if the access time is not clearly stated refer to the EPROM datasheet.

For **Briel Computer Replica-1+ / Replica-1 TEN owners**, replacing the stock EEPROM with a slower one will not be necessary: reprogramming alone will be sufficient.

The SD OS 1.3 firmware to be uploaded to the E/EPROM is available on the reference site and can be uploaded using standard E/EPROM programmers.

If you have any doubts, please do not hesitate to contact us.

5. ASSEMBLY OF THE EXPANSION BOARD

This product is not sold as a kit to assemble, however, assembly of it requires basic level soldering skills and there are no critical components. Refer to schematic and Bill of Materials.

You must also work on your *Apple-1 microSD Storage Card* and solder a 1x12 pin strip-line connector to the J1 expansion slot, so that you can remove the *IEC interface* when necessary.

It is therefore essential, before handling/soldering the cards, to put in place all precautions against static electricity build-up, as already described in the Apple-1 microSD Storage Card Reference Document.

We do not take any responsibility for any damage, even serious or lethal, caused to people / things / intellectual property while installing or using this device.

When the work is completed, do not forget to thoroughly clean the cards of any solder residue.

6. FINAL ASSEMBLY AND SWITCH ON

Connect the 6-pin DIN serial cable to the Disk Drive as usual. The other end will of course have to be connected to the *IEC Interface*, whose serial cable is known to be thick and rather stiff.

It is therefore recommended to:

- connect the serial cable to the *IEC Interface* only.
- mount the *IEC Interface* on the microSD Storage Card paying attention to the pin matching on the J1 connector, the DIN connector must be on the left side.
- mount *microSD Storage Card + IEC Interface* on the Apple-1 computer making sure that everything stays firmly in place.

Schottky diodes have been provided on the *IEC Interface* to protect the VIA 65C22.

However, extreme caution and anti-ESD protections are always recommended during each step.

Double-sided tape can be used between the microSD Storage Card and the *IEC Interface* to reduce mechanical stress on the J1 connector and the possibility of accidental disconnection of the board.

If Disk Drive 1541 or 1541/II are used, there isn't any particular switch on sequence.

If a Pi1541-type emulator is used, it should be switched on only after starting SD OS 1.3 (using the usual 8000R command).

Otherwise, the Pi1541 selection keys might not work. This is not a symptom of bugs or malfunction but originates from the Raspberry Pi checking the transmission lines during its boot. If SD OS is not running those lines will not be in their nominal state and will prevent the Pi1541 from booting properly.

If needed, a RESET button has been provided for the Drive(s). It has no effect on Apple-1.

If you wish to connect multiple Drives in cascade, make sure that each one has its own unique identifier (8 to 11) and therefore there are no conflicts.

7. OPERATIONS

The commands of the *microSD Storage Card* haven't changed for the part related to SD CARD operations.

We have added some commands dedicated to program handling (both binary and INTEGER BASIC). We also have kept the possibility to issue to the Disk Drives most of the CBM DOS commands, such as commands for Formatting, disk Validation, Drive initialization, etc.

So... Congratulations! You have given your Apple-1 the ability to load and save your games and programs also on floppy disk!

See Appendix 1 of this document for the complete list of commands.

8. FINAL NOTES

The *IEC expansion* was made almost for fun... and to use some of the Versatile Interface Adapter (VIA) 65C22 I/O lines left unused by the *microSD Storage Card* project.

It still needed a considerable number of hours of design, programming, testing, and debugging. The reward for such efforts, besides our satisfaction in using it, was the understanding and practical application of data transmission/storage techniques far outdated to today's.

This has once again enriched our knowledge of the history of computing, and we hope it will do the same for you!

SD CARD OS 1.3 was created and made available by its creator, **Antonino Porcino**, on the following GitHub repository:

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

In the Bibliography/Credits section, we have tried to list all the projects and documents that have been inspirational for both firmware and hardware.

The product is constantly evolving visit the reference pages often:

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

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

to stay up to date about new features and updates!

We you will enjoy our microSD Storage Card!

APPLE-1 MICROSD
STORAGE CARD

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

[P-L4B @ PROTONMAIL.COM](mailto:p-l4b@protonmail.com)

BIBLIOGRAPHY / CREDITS

Implementing Commodore's IEC bus protocol on a KIM-1 single board computer [KIM-1 Videos]

<https://www.youtube.com/watch?v=cK7mzGhBfHs>

IEC dissected

<https://commodore.software/downloads/download/215-articles-and-guides/15601-iec-dissected>

1540/1541 technical manual. Schematics and other information for drives up to 1541 B.

<http://www.zimmers.net/anonftp/pub/cbm/schematics/drives/new/1541/index.html>

Commodore C64 Manual: Commodore 1541 Disk Drive User's Guide (1982-09)(Commodore)

https://archive.org/details/Commodore_1541_Disk_Drive_Users_Guide_1982-09_Commodore/mode/2up

APPENDIX 1 – IEC COMMANDS

Here follows the list of commands dedicated to the Apple-1 environment. All commands intended for IEC devices begin with the @ character. Parameters in square brackets are optional. Wildcard characters such as % and * are generally allowed in file name convention.

@DEV [device]

Show or change the number of the IEC device you want to use. The default is 8. The information is kept even if the computer is RESET.

Examples: the first command displays the current device, the second sets it to 10.

```
@DEV
DEVICE: 8
/>
```

```
@DEV 10
/>
```

@\$ or \$DIR

Shows the directory of the disk inserted in the current device. Wildcards such as % and * are allowed to be used to filter the output of the command. Press any key to temporarily stop the display, press ENTER to resume it. Pressing ESC during the display will end the directory display and go back to the command prompt.

Examples:

```
/>@$
"DISKLABEL      " A1 2A
17  "BASIC"      PRG
64  "ELIZA"      PRG
16  "GOMOKO"     PRG
64  "STARTREK2003" PRG
16  "STARTREK"  PRG
46  "ADVENTURE"  PRG
441 BLOCKS FREE
/>
```

```
/>@DIR STAR*
"DISKLABEL      " A1 2A
64  "STARTREK2003" PRG
16  "STARTREK"  PRG
441 BLOCKS FREE
/>
```

@L FILENAME [startaddress]

Reads the FILENAME binary file from disk and loads it into memory from startaddress. If the address is omitted the load will be from the default location contained in the file itself. The use of wildcards such as % and * for the filename is allowed.

Examples:


```

/>@L BASIC
SEARCHING
LOADING

BASIC
$E000-$EFFF (4096 BYTES)
OK
/>

/>@L ADVENT* 280
SEARCHING
LOADING

ADVENT*
$0280-$3007 (11656 BYTES)
OK
/>

```

@S FILENAME startaddress endaddress

Writes to disk a binary file called FILENAME with the memory contents read from startaddress to endaddress included. Omission of any of the addresses will produce an error message.

Example:

```

/>@S BASIC E000 EFFF
SAVING

BASIC
$E000-$EFFF (4096 BYTES)
/>

```

@R FILENAME [startaddress]

Reads the FILENAME binary file from disk, loads it into memory from startaddress then run it. If the address is omitted the load will be from the default location contained in the file itself. The use of wildcards such as % and * for the filename is allowed.

Example:

```

/>@R BASIC
SEARCHING
LOADING

BASIC
$E000-$EFFF (4096 BYTES)
>

```

Note how the > prompt in this example belongs to the INTEGER BASIC environment, a symptom that it has been loaded and is correctly running.

@BL FILENAME

Reads the INTEGER BASIC file called FILENAME from disk and loads it into memory. The use of wildcards such as % and * for the filename is allowed.

Example:

```

/>@BL ELIZA

```

```
SEARCHING
LOADING

ELIZA
(LOMEM=$0300 HIMEM=$4000)
OK
/>
```

Trying to load a binary file as an INTEGER BASIC type file will cause an error message:

```
/>@BL APPLE30TH
SEARCHING
LOADING
?NOT A BASIC FILE
/>
```

@BR FILENAME

Reads the INTEGER BASIC file called FILENAME from disk, loads it into memory, and runs it. The BASIC interpreter must already be present in memory. The use of wildcards such as % and * for the filename is allowed.

Example:

```
/>@R ELIZA
SEARCHING
LOADING

ELIZA
(LOMEM=$0300 HIMEM=$4000)
OK
*****
      ELIZA
CREATIVE COMPUTING
MORRISTOWN, NEW JERSEY
*****

HI, I'M ELIZA, WHAT'S YOUR PROBLEM?
```

Trying to load a binary file as an INTEGER BASIC type file will cause an error message:

```
/>@BR APPLE30TH
SEARCHING
LOADING
?NOT A BASIC FILE
/>
```

@BS FILENAME

Writes to disk an INTEGER BASIC file called FILENAME. Information and contents related to HIMEM and LOMEM, if any, are kept and saved as well.

Example:

```
/>@BS MYPRG
SAVING

MYPRG
```

```
(LOMEM=$0800 HIMEM=$1000)
OK
/ >
```

@ERR

Reads the Drive Error Channel and reports the Error Number, Error Name, Track, and Block where the error occurred. The command should be issued in case of Drive error (LED flashes quickly) to investigate the cause.

Examples (the first message usually appears at power-up and it's not a real error message):

```
/ > @ERR
73, CBM DOS V2.6 1541,00,00

/ > @ERR
00, OK,00,00

/ > @ERR
63, FILE EXISTS,00,00
```

For a comprehensive description of error messages refer to the Disk Drive 1541 manual listed in Bibliography.

@CMD COMMAND

Sends the specified **COMMAND** to the Disk Drive via channel #15, the opening and closing of which is managed directly by SD OS. Some examples of commands:

Format a diskette, name it MYNEWDISK and assign ID A1 to it:

```
/ > @CMD N0:MYNEWDISK,A1
```

Delete the file called WRONGFILE:

```
/ > @CMD S0:WRONGFILE
```

Initialize the Drive and clear the error channel (this is equivalent to a soft reset):

```
/ > @CMD I
```

Reorder the files on the diskette and the BAM (Block Availability Map), also removing any files that were not saved properly (often marked by the "*" character next to the file type, such as *PRG):

```
/ > @CMD U
```

For the complete list of commands refer to the **Disk Drive 1541 Manual** listed in Bibliography.