

SD-DOS

for

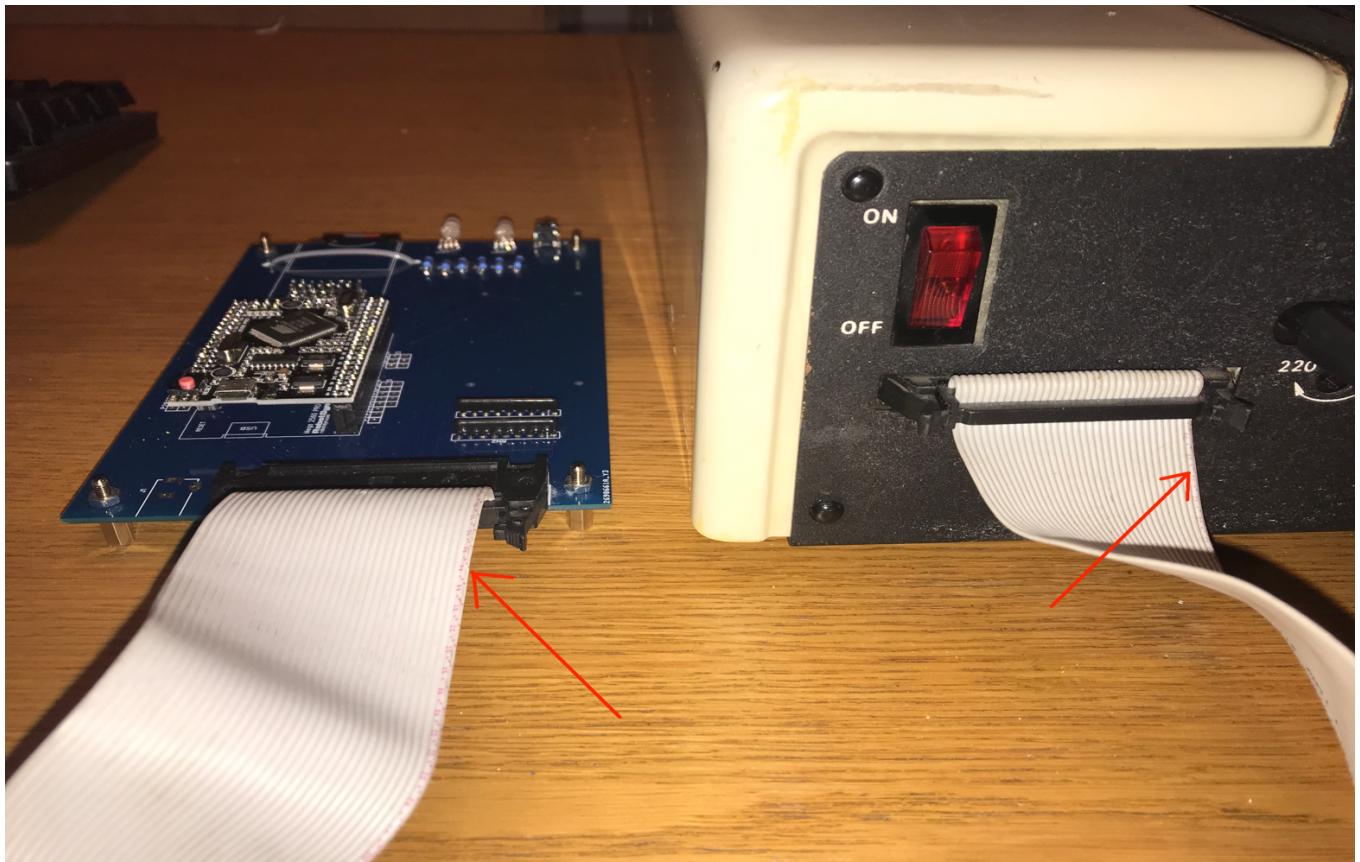
DAI

PERSONAL
COMPUTER

v 1.0

Part I: Installation of SD-DOS

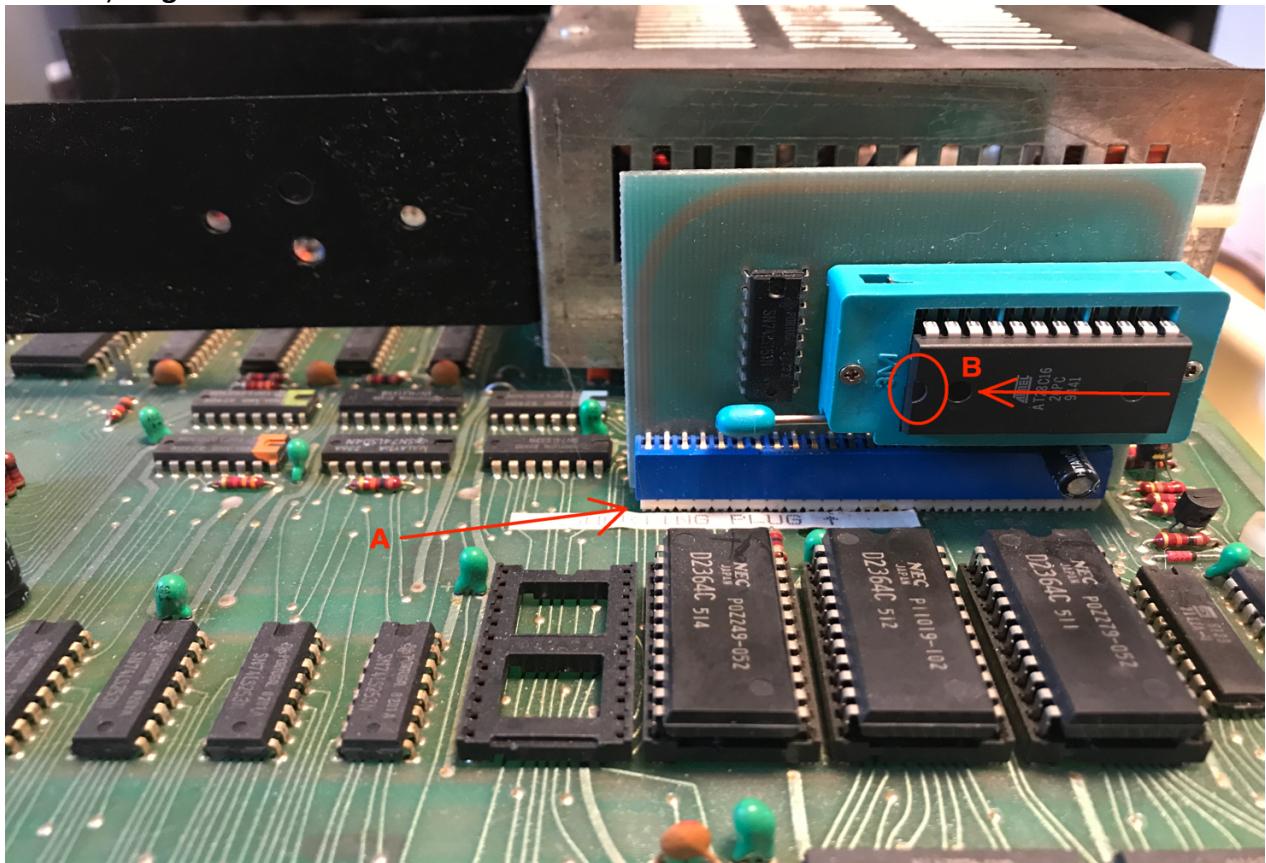
1) Connect the SD-DOS to the DAI using DCE PORT:



For this you can use a 34 wires IDE cable like those connecting IDE controller to Hard Disk in the old PC. These cables are easy to find.

Be careful: Pin1 of DCE DAI must be connected to Pin 1 of the SD-DOS. Which means the rightest position. In the cables, often, the wire of the pin 1 is colored (red or blue) like on the photo. Be sure to respect that.

2) Plug the XBUS card in the DAI

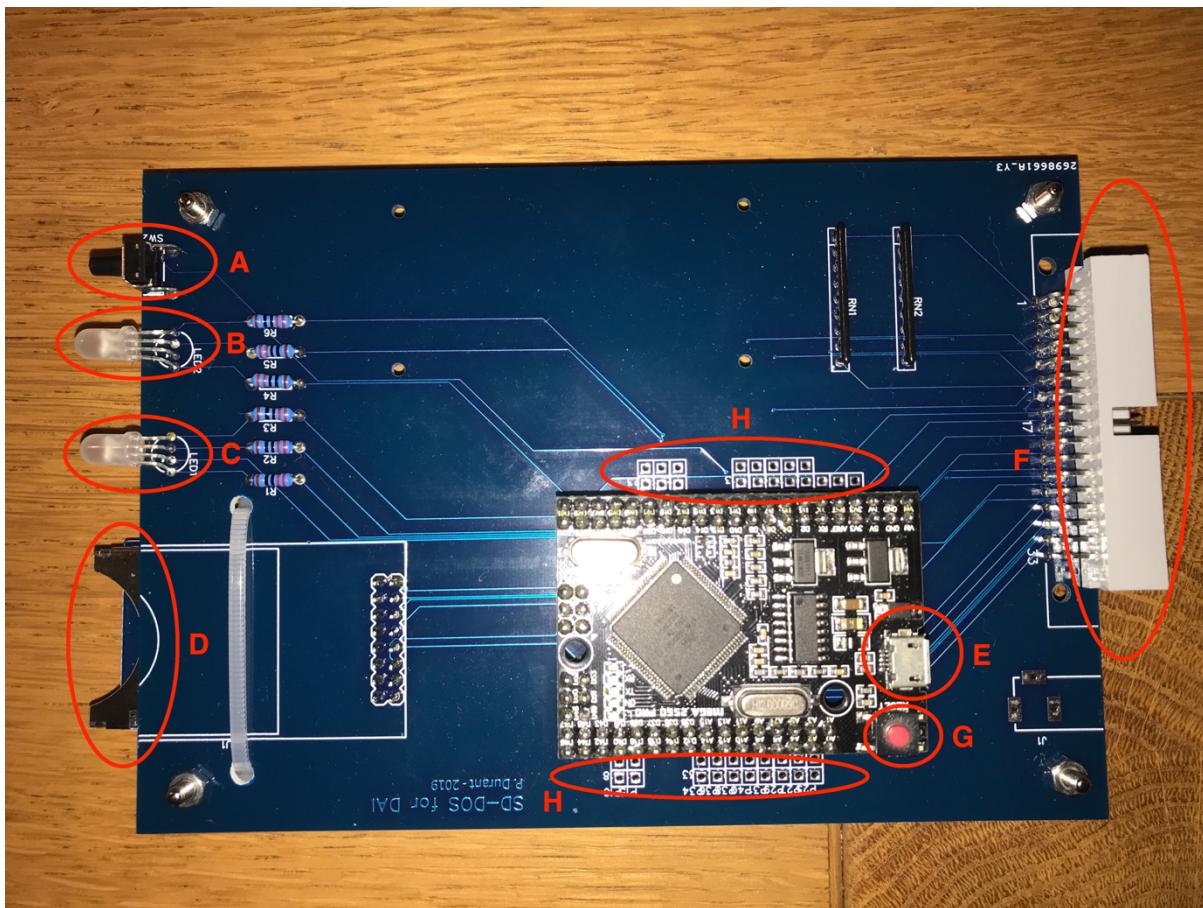


Plug the card in the XBUS connector. Don't hesitate to verify that the connect is OK.
If you already have a memocom card, unplug the old memocom eprom and plus the new eprom received with the SD-DOS. Respect the correct direction when you plus the eprom. Like on the photo, the "notch" must be at left.

- 3) Connect the micro-USB port of the Arduino card (see next photo) on a USB power unit. A 1A seems enough. I succeed using a simple charger of iPhone.
- 4) You can then switch-on the DAI. If the Led become BLUE, push on the reset button of the SD-DOS

Once all is connected and turned on, type “CALLM 61440**” in order to start the SD-DOS in the ROM !!!**

Part II: Hardware description



A: It's the RESET button of the card. It has the same function than G. Sometimes, when something goes wrong, when the LED remain white, or blue even if nothing is happening anymore on the DAI, a RESET can be necessary. Or when you enter a new SD-card in the reader

B: Same LED than the Arduino little LED. It blinks when you switch-on the SD-DOS or once you reset it.

C: It's the LED of the SD-DOS. In the next pages, when we write "the Led", we reference this Led !

WHITE (not switched) means: NO CARD or CARD NOT RECOGNIZED.

GREEN means: a CARD is PRESENT AND RECOGNIZED.

RED means: AN ERROR OCCURRED

BLUE means: SD-DOS IS WRITING OR READING

D: It's the card reader. Insert the here the Sd-card with the connectors turned down

E: It's the micro-USB connector of the Arduino. Connect here a USB power unit (like an iPhone charger). Use a good "full" usb-micro cable. I experiment it with some "simple" micro-usb cable provided with charger for mouse and it didn't work !

If you want to change the programs in the Arduino, you have to connect your computer to this usb port (see instructions provided on the Arduino web site).

F: connector used to connect the SD-DOS to the DAI (see part I). Again, be careful: PIN 1 (most right) of this connector must be linked to the PIN1 of the DAI DCE port.

G: reset button of the Arduino, see A

H: connection for the unused Arduino I/O pins. Not used by SD-DOS, but it allows you use them for possible extension/new module. At your own risks of course ☺

Part III: the SD-Card

Introduce the Sd-card in the reader and press on the reset button. The led must then become GREEN. If not, this indicate a problem: the card is not recognized. Check the card with your usual modern computer.

The Sd-card must be FORMATTED on your usual computer(pc, mac , Linux). For old standard Sd-card from 16Mb to 2Gb, use the FAT16 format. For the new SDHC from 4Gb to 32Gb, use FAT32 (not sure Arduino can manage card with more than 64Gb).

Part IV: the existing commands

In order to remain totally compatible with the DAI, the existing commands LOAD/SAVE/LOADA/SAVEA/R/W are used by SD-DOS. Only some routines of these commands have been adapted in order to manage the new data flow from/to the DCE port. This means also that we continue to use the standard DAI error message even if sometimes they are not adapted anymore at a support like the Sd-card....

For example, as the primary peripheral of the DAI at this time was the tape recorder, and as DAI want to provide the possibility to give a name to a file, they decided to include the name of the file in the header of the file itself. But, today, with modern storage peripherals like Sd-card, the name of the file is stored in a File Allocation Table, so no need to store it in the file. That's why, with SD-DOS for DAI, you can have a difference between the name inside the file and the name on the SD-card. When you do a RENAME or a COPY, the new name is the name used in the FAT, not the name inside the file, which is not important anymore. This explain that you can LOAD a file with a name but see another name displayed immediately after the command: the name displayed is the old name inside the file. This is not a problem at all, your file will be correctly loaded with the name found on the Sd-card.

Other problem: the files WITHOUT names !

With a standard old tape, a file can be loaded or saved Without specifying a name in the command. In this case, the DAI simply load or save the next file on the tape without checking the name.

Of course, on a modern support like the Sd-card, this is IMPOSSIBLE ! A file MUST have a name.

So, how SD-DOS solve this issue ?

When you save or load without specifying a name with SD-DOS, a name is automatically used: "AUT0xx" where "xx" is an index number.

Two indexes are maintained: one for the reading of the files, one for the writing of the files. The new command "INDXAUTO" can show you the value of each index but also can modify this value.

So, for example, if the index for reading file is at 5, typing the command "LOAD" will load the file "AUT05.BAS" and modify the value of this index to 6 for the next read.

1) LOAD

Same syntax than the DAI standard.

With a name: LOAD "name". In this case, the file with the name "name.BAS" will be loaded from the Sd-card. The extension ".BAS" is automatically added, don't add it in the LOAD command !

Without a name: LOAD. In this case the file with the name "AUT0xx.BAS" will be loaded from the Sd-card. "xx" is the value of the index for reading.

2) SAVE

Same syntax than the DAI standard.

With a name: SAVE "name". In this case, the file with the name "name.BAS" will be saved to the Sd-card. The extension ".BAS" is automatically added, don't add it in the SAVE command !

Without a name: SAVE. In this case the file with the name "AUTOxx.BAS" will be saved to the Sd-card. "xx" is the value of the index for writing.

3) LOADA / SAVEA

Same than LOAD/SAVE, read your DAI manual. The only difference is the extension which is ".TAB" in this case.

4) R/W (READ & WRITE under UT)

Same than LOAD/SAVE, read your DAI manual. The only difference is the extension which is ".BIN" in this case.

Part V: the new commands

As you can see, in the new commands, contrary at the standard DAI command, the name of the file is not between " " AND the extension of the file must be specified !

Also, error messages are new and clear.

1) INDEXAUTO

INDEXAUTO S: display the value of the index used when Loading file without filename.

INDEXAUTO L: display the value of the index used when Saving file without filename.

INDEXAUTO S > x: Change the value of the index used when Saving file without filename to the new value x.

INDEXAUTO L > x: Change the value of the index used when Loading file without filename to the new value x.

2) DIR

Display the content of the main directory (for the moment, SD-DOS don't manage the sub-directory, possible in a future development). Press SPACE to interrupt the list display, press again SAPCE to continue the display.

3) RENAME

RENAME oldfilename.XXX > newfilename.XXX (where XXX is BAS, BIN or TAB)

Rename can change the name of a file. No new file is created.

4) COPY

COPY filenameorigin.XXX > newfilenamedestination.XXX (where XXX is BAS, BIN or TAB)

Copy can make a copy of a file with another name the name of a file. A new file is created, the original one is preserved.

5) DELETE

DELETE filename.XXX (where XXX is BAS, BIN or TAB)

Delete the file.

6) TAP

TAP

Redirect all the standard DAI commands (LOAD, SAVE, etc...) to the TAPE interface !

This authorize you to switch from Sd-card to tape. So, for example, you can read a file on a tape, switch to Sd-card and save it on the Sd-card.

7) SDC

SDC

Redirect all the standard DAI commands (LOAD, SAVE, etc...) to the SD-DOS interface ! This authorize you to switch from tape to Sd-card. So, for example, you can read a file on a tape, switch to Sd-card and save it on the Sd-card.

8) CLS

CLS

Clear the text screen.

9) TESTMEM

TESTMEM

Test your memory. This is a way to test the RAM of your DAI. This test has been written by Bruno Vivien. You can find more information on it on his website:

<http://bruno.vivien.pagesperso-orange.fr/DAI/reparation/testmem.htm>.

Don't be surprised, this test can take 15 minutes.

10) ABOUTDOS

ABOUTDOS

Give the version number of the SD-DOS

KNOW BUGS: ABOUTDOS & DIR with a limited number of files to display, when used at the beginning, display what they have to display BUT don't give back the control to the command line. The solution is simple: just push on the "BREAK" key for 2-3 seconds. Once some displays have been done, the problem disappear. Probably a problem of buffer fore screen output. Will try to solve it later as it's not blocking at all.

PART VI: how SD-DOS works

There was no real DISK standard for DAI. Various disk solution existed (the DAI one, Ken-DOS, adapter for CMD 1541, etc...) but no-one became a real standard. In fact, on the DAI, the real standard was the normal tape and , but more limited, the memocom tape. A good illustration of this is there are people searching for the DAI DOS for their DAI disk drives since years without success....

So, the ambition of the SD-DOS was not to emulate one of the disk solutions but to offer a new solution in order to replace the use of the tape, extremely slow and uncomfortable.

I decide to build a solution based on an “exchange of messages” between the DAI and the Arduino. The DAI send ORDER and the Arduino send ANSWER.

For example, when you LAOD a file, the DAI sent the message to the Arduino “send me the file with the name xxx” and the Arduino, read the Sd-card , find the file and, as answer, send it to the DAI byte by byte.

This means that other new commands are possible, the limit is in fact the limit of the ROM where we can store the new command code (between F000 and F7FF)

