

M4 Board instructions (updated 2024-11-27)

Change log

v1.0.0

- **Initial version**

v1.0.1

- M4 rom defaults to Rom 6. Changes in romcore emulation to support Gate Array 40007.

v1.0.2

- **|CD** command now accepts no parameters and will prompt with cursor, so Basic 1.0 users can use SHIFT+CURSORKEYS+COPY.

v1.0.3

- Allow to remap entire romboard for compatibility with XMEM. Now supporting 32 romslots.

v1.0.4

- Ability to replace lowerrom.

v1.0.5

- FTELL, FSIZE commands implemented and new RSX commands **|ROMUP**, **|ROMSET**, **|ROMUPD**

v1.0.6

- **Fix** WEB UI instability issue. Fix romupload for roms with or without AMSDOS header, and related bugs.

v1.0.7

- Fixes and buffering for cas_in_char, fix **|ROMUP** to work from .dsk files, fix cat/dir loop, fix httpget to work with IP addresses aswell as hostnames.

v1.0.8

- **FIX** IP addresses for httpget (again). Fix |dir bug. Fix hardreset. Added C_GETPATH command (requested).
- **Added** ability to load your own build of M4ROM from root of microSD, should be named M4ROM.BIN

v1.0.9

- Nonblocking NETAPI (refer to new tcpecho sample for details). Added ESC check to cat/|dir.
- M4 board now looks for autoexec.bas at bootup, similar to old dos machines, autoexec.bat. Make your own startup script in basic.

- Direct SD card sector access.
- Various minor fixes.

v2.0.0

- Stability fixes for netapi.
- **Added** RSX commands |GETPATH, |UDIR, |LONGNAME by SOS
- **Added** C_GETNETWORK command.
- **Added** status byte after response string to C_NETSTAT
- **Fix** |cd, "/" to go to root if inside a DSK image.
- **Fix** SD hotswapping.
- Remove KM_RESET from cat/dir when pressing ESC.
- Display current path when using cat/dir.
- Increased AMSDOS readspeed (using bigger buffer).
- Faster network speed when using netapi.
- Keep CPC in reset until init. done. (Fixes issue with C4CPC and M4 at the same time).
- Many amsdos compatibility fixes.
- DSK images, handle ASCII art with cat.
- **Fix** multiple network connections.
- **Added** |LS command for long filenames (15 mode 0, 35 mode 1, 75 mode 2).
- |cd can use long names too.
- autoexec.bas method changed, should no longer give problems with certain basic games.
- Remote run - run remote files from WEB interface (control)
- |TAPE support, use |SD to switch back to M4
- |DISC supported if AMSDOS present, use |SD to switch back to M4
- **Fix** |era long filenames
- |ren fix for long filenames
- **Added** support for ROMDOS dsk images (D1/D2/D10/D20/D40) and PARADOS 80.

v2.0.1

More fixes for |era & |ren (now case sensitive)

- **Added** |FCP command to copy between discs and microSD.

v2.0.2

Fix carry flag of CAS OUT CHAR

- **Added** tiny |M4HELP
- Fixes |FCP when using wildcard on Drive B, system protected files, excessive 1 byte on <2KB files, display of filename while copying & basic protected files.
- **Fix** C_WRITESECTOR for DSK images.
- **Fix** DSK images not having entries ordered.
- **Added** support for cpr dsk images.
- **Added** |SNA command to load snapshots from microSD card.

v2.0.3

- **Added** javascript/html5 based navigation to the webinterface, with multfile upload, download, remote run, delete & make directory.
- **Added** ability to CD into DSK & CPR DSK images from the Web UI and use remote run too.
- Fixes with running binary files with remote run and improved by changing directory to actual remote folder aswell.
- |FCP Fix dodgy filenames, removing system attributes.
- |M4HELP now lists the first 32 roms. Use |M4HELP,romnumber to display commands for a specific rom.
- Full amsdos header read into amsdos buffer area (will let you retain unused part of header when copying/opening).
- **Fix** autoexec.bas launching when soft-resetting inside a .DSK
- **Fix** |cd into .dsk's with write protection attribute set
- **Fix** web remote run for filenames using html encoded characters
- **Fix** |M4help, using all screen modes

v2.0.4

- **Added** |ctrup and |ctr RSX for CPC plus owner, giving the ability to launch cartridge images.
- **Fix** parsing bug when file not found
- Many DSK fixes, now games like Chibi Akumas show loading screen (again), games with fragmented blocks work (ie. Ninja Grannies)
- WEB UI, now you can pick "CD ON CPC" when using file browser or from Control, to change the directory on CPC itself
- Besides using the webui a http req (ie. wget) can also change directory like this: <http://ipnum/config.cgi?cd2=/DEMOS> or even into a DSK file.
- [...] in the file browser was changed to [BACK] and [...] to [REFRESH] to make it more obvious.
- |ls now supports DSK images too (will probably garbage up if using ascii art though).
- |httpget will use attachment filename, rather than url if present.
- New command |dskx. With this you can extract dsk images to files. Usage |dskx,"somedisk.dsk","/mypath" and all files are extracted to given path.
- CPC+ init. no longer clears screen.
- **Fix** downloading files from the WEB UI
- Support 512KB cartridge images (mainly used by CPC+)
- **Added** option in web interface to downgrade romboard to 16 roms, to be able to fit 512 KB cartridges.
- **Added** option to upload & start cartridge images via web interface.
- Set DHCP hostname to netbios name.
- **Fix** crash when using run" without filename multiple times.

- Improve timings of ROMDIS.
- Only test IO bit 13 for rom selection (as CPC seems to do), was 0xDF.
- **Fix** Reset (clear rom state).
- Change remote Reset, M4 reset & pause to acknowledge (IPC) commands or they could sometimes be ignored.
- Updated ESP side SDK base to latest (### v2.2.0), with whatever fixes that comes with it (ie. WPA/WPA2 vulnerability issue).
- **Fix** Remote .sna for long filenames.
- **Fix** for loaders using BE7D to restore drive letter (not checking if it is 00 00).
- Change Amsdos workram 0-2 (ie. A700-A702) to 0 when using |SD and restore when using |DISC (only if AMSDOS or PARADOS present).
- DSK: Do not show files with system attribute set. (makes better catart too).
- DSK: Show '*' after file with "Read Only" attribute set.
- DSK: Do not reject strange sector numbers (like AMSOFT disk protection).

v2.0.5

- **Fix** CPC+ cartridge issue with demos Eerie Forest & CRT3 (and maybe others)
- **Fix** issue with certain Amsoft games
- **Fix** rom issue.
- **Fix** SNA pen 0 bug.
- **Fix** SNA screen mode bug.
- **Added** SNA v3 (compressed sna's) support.
- **Fix** for upgrading ESP firmware (do not erase if CRC does not match).
- **Added** |ROMSOFF command
- **Added** upgrade report.txt (useful for diagnosing, upgrade problems).
- **Added** Hack Menu, now you can save snapshots while playing games etc.
- **Fix** DSK file size 0 bug
- **Fix** |longname / C_FSTAT bug, now (again) displaying long filenames properly
- **Fix** Hack menu - Slowdown keyboard reading.
- **Added** Hack menu - Support full keyboard character set (QWERTY only for now).
- **Added** Hack menu - Pokes
- **Added** Hack menu - Display memory
- **Fix** Hack menu - Flaw in IM 2 detection.
- **Fix** Hack menu - Resume "some games" (Pinball Fantasies, atleast on my one CPC).
- **Added** Hack menu - AZERTY keyboard support (Huge thanks to @dxs for the painful keymap table! And thanks to @ZbyniuR for the keyboard detection method).
- **Added** ability to launch Hack Menu with an external button.
- It requires to solder two wires to the M4 board PCB (the wifi module). If you install this, it is at your OWN risk. You can also just launch hack menu from the Web UI. Instructions are here:

<http://www.spinpoint.org/cpc/m4hack1.jpg> <http://www.spinpoint.org/cpc/m4hack2.jpg>

- **Fix** file output of Maxam "write" command
- **Fix** |HTTPGET strip "http://" from URL if used
- **Fix** |HTTPGET freeze when URL does not resolve (2 sec DNS lookup timeout)
- **Added** |HTTPGET use @ symbol at start of request to silent output text of console, ie. |httpget,"@spinpoint.org/cpc/m4info.txt"
- **Added** |HTTPGET use > symbol followed by filename at end of request string to output into desired file, ie. |httpget,"spinpoint.org/cpc/m4info.txt>info.txt"
- ADD C_NMI command to trigger NMI/HACKMENU from z80 code for debug
- **Fix** WEB server missing HTML escape tags in files upload page.

v2.0.6

- **Fix** bug in |FCP corrupting (amsdos) files when transferring from microSD to DISC.
- **Fix** |ROMUP issue (multiple rom upload)
- **Fix** (internal) for NMI/HACKMENU

v2.0.7

- Changed behaviour of RMR "ghost" register to support FutureOS and IMPdraw running from ROM on Amstrad PLUS.
- **Added** C_ROMLOW command to map in and out lowerrom(s). See under developer info.
- NMIROM.BIN is now loaded from microSD card root by default, if present, otherwise internal HACK MENU is used.
- New HACK Menu improvements by Cebe74 (Thanks!):
 - Reorganized layout
 - Menu keyboard shortcuts
 - Memory dump feature

v2.0.8

- **Fix** DNS issue, preventing name resolving under certain conditions. (A big thanks to Poulette73 for testing).
- **Fix** bug in net accept.

Download latest version

http://www.spinpoint.org/cpc/M4FIRM_v208.zip

Please notice, on M4 boards shipped after 2016-10-01: Due to a new flashchip on the wifi-board, do not use firmware versions lower than v1.0.9. If done by accident, use v1.0.9 or higher to restore it.

Updates can manually be downloaded from here: http://www.cpcwiki.eu/index.php/M4_Board

When manually downloading an upgrade, you must unpack the .zip archive to the root of the microSD card. Then insert the microSD card to the M4 and powercycle the CPC/M4 board, after about 20 seconds it should be updated.

Hardware

The M4 Board can be powered directly from the CPC or via USB power supply (5V). There is one single jumper on the board, to set this. If set at EXT it will use the CPC's power supply and if set at USB it will use the power from the USB socket.

There is two status LED's:

- **ON**, indicates if the board has power (blue led).
- **DD**, Disc Drive, this will flash when there is reading/writing from/to the SD card.

The switch on the board, can be used to reboot the M4 board aswell as the CPC.

Newer revisions of the M4 board has an additional switch, to reset the CPC itself, having the advantage you will not loose whatever path you are in.

M4 has been tested on all CPC models, ranging from CPC464 to CPC6128+. No DDI interface required or other external hardware.

Setup

Insert a microSD card to the M4 Board. This should preferably be formatted as FAT32. Boot up the CPC. At this point you can use the SD filesystem natively like SAVE/LOAD/RUN/CAT etc. LOAD/SAVE should behave as normal, if you don't specify an extension, it will be auto added (., .bas, .bin). .bak files will be created if overwriting a file with SAVE.

If you wish to use the web interface, romboard, realtime clock or download stuff from the internet. You need to setup the network connection.

It's done via the rsx command |netset,"<insert parameters>".

Parameters

Parameter	Description
name	netbios name, only use UPPERCASE letters and numbers.
ssid	your wireless accesspoint/router name (remember name is case sensitive).
pw	password for your wireless ap/router.
dhcp	0 =disable DHCP and use static ip settings, 1 =use DHCP (static IP settings are ignored)
ip	static ip number for your CPC
gw	gateway for your network
nm	netmask for your network (usually 255.255.255.0)
dns1	dns server, you can use ie. 8.8.8.8 (google dns)
dns2	dns server backup, you can use ie. 8.8.4.4 (google dns)
ntp	ntp time server, this will be used to retrieve time
tz	time zone, can be +/- 12, set it to your time zone

Parameter	Description
start	when start is set to 1, settings are not applied yet. This is if your parameters are longer than 255 chars (cpc basic line limit), then you can issue the command twice.

Example usage, using DHCP

```
|netset,"name=CPC6128, ssid=NETGEAR, pw=12345678, dhcp=1, dns1=8.8.8.8, dns2=8.8.4.4"
```

This is enough to set you up to the internet. You can alter the rest of the settings from the web interface of the M4 Board.

First type: |netstat to see if you are connected, or there's some error. If connected, it will display your IP number.

You can type this into firefox browser (other browsers may be fine, that's just what I used) or the netbios name, and access the rest of the configuration from there under settings. When using netbios name, make sure you type it like this: <http://CPC6128> (remember the http) It might take a bit the first time as your computer will have to associate the netbios name to the IP. Similar when using the IP number <http://192.168.1.20> (what you used or got from |netstat)

After the initial setup this is all saved to the microSD card under the folder m4 as config.txt. Settings can be altered there manually aswell with a text editor. Next power cycle these settings will be read and cpc auto-connected to the internet.

Using a static IP is recommended, as this will make the board connect almost instantly.

File system

Filenames will be displayed in 8.3 format for compatibility. Directories will display in full name if less than 17 characters. If longer short directory names will be used. When displaying a directory/catalog, directories will have a > character in front of the directory name.

Using |LS command to display a directory you can display long filenames, only limited by the screen mode.

If a file called "AUTOEXEC.BAS" is put in root of microSD, this will be autostarted when the CPC is turned on. This is useful for ie. making custom settings, like setting screen mode & colours.

Example:

```
10 mode 2
20 ink 1,26
```

save"autoexec.bas

Notice: This should be done on the CPC itself, not in notepad on PC, unless you manually add the correct fileheader.

Commands

It has the same commands as the original system, ie. load, save, run, openin, openout, cat etc.

However to deal with directories, wifi and more some rsx commands were added and some have extended usability.

Notice on CPC's with Basic 1.0 (CPC464), RSX commands work in a different way, you can't pass the parameters directly to the rsx. You will have to do it via a variable. Ie. change into a sub directory called "DISCS". Type: A\$="DISCS":|cd,@A\$ To make it easier to navigate for basic 1.0 users a prompt was build into |cd command. You can press |cd (enter) then type or use cursor copy for the name.

On basic 1.1 it would be: |cd,"DISCS".

With the M4 board you can replace the lowerrom, so you can use Basic 1.1 on CPC464 too. I have made a small program to do this for you, you can download it from the CPC if you have setup the WiFi

```
a$="spinpoint.org/cpc/ROMUP464.BIN"  
|httpget,@a$
```

or download via your webbrowser <http://www.spinpoint.org/cpc/ROMUP464.BIN> and copy it to the microSD card.

You can run it by **RUN"ROMUP464**

It will also set M4 rom to 7, which is best for compatibility with games. You can change it from the webinterface again.

|CD

Change dir, as known from DOS, you can use it like:

- |cd, "games and it will go to the subdirectory games, but you can also do
- |cd, "games/batman" and go directly to another sub-directory.
- |cd, ". ." takes you one step back and
- |cd, "/" takes you to the root of the sd.

DSK files widely used by emulators, can also be used as a folder, where you do ie.

- |cd, "robocop.dsk" and it takes you inside the DSK file as it was a folder.
- |cd, ". ." takes you back out.

Notice: DSK files are read only for now.

It's also possible to |CD into converted CPR dsk images

(http://www.cpcwiki.eu/index.php/Converted_GX4000_Software) similar to use .dsk images.

|CTR

Added v2.0.4.

Start cartridge image flashed into M4 flash (and crash if not present). **CPC PLUS only!**

| CTRUP

Added v2.0.4.

| CTRUP, "cartimg.cpr". Upload cartridge image to M4 flash (max. size used is 256KB). Format can be binary or .cpr

Format can be either plain binary dump or .CPR format.

| COPYF

Copy file, src file, dest file. Paths can be used too. (for microSD only)

| FCP

Added v2.0.1.

Copy files between microSD and floppy drives.

- | FCP, "MYFILE.BIN", "A:" will copy MYFILE.BIN to drive A. You can also address microSD with drive letter C:
- | FCP, "A:MYFILE.BIN", "C:" will copy MYFILE.BIN to current directory in microSD.

There is one wild card option, "*" to copy all files from a disc or directory.

- | FCP, "A:*", "C:" will copy all files on disc A to current directory.

| REN

Rename "*new file*", "*old file*".

Just like with CPC discs, but you can also use it to move one file or folder from one directory to another.

Example: | ren, "games/robocop.dsk", "robocop.dsk" moves the file from current directory, into the games directory.

| ERA

Erase a file or folder. Wildcards can be used to delete multiple files/folders.

Examples:

| era, "*.bak" erase all .bak files in current directory.

| era, "games/r*.d?k" removes all matches under games directory.

| MKDIR

Create directory in current dir, or in remote dir.

| mkdir, "games/discs" creates a subfolder under games directory.

| DISC

Added v2.0.0. Switch to AMSDOS (floppy discs) if present

|SD

Added v2.0.0. Switch to M4 (microsd)

|SNA

Added v2.0.2. Start SNAPSHOT image made by an emulator. I.e. |SNA, "FRANKIE.SNA"

|DIR

Display directory, similar to cpc discs, you can add wildcards to it, unlike the cat command.

Example: |dir, "games/discs/b?t*.dsk" etc.

|DSKX

Extract files from DSK image.

Usage |dskx, "somedisk.dsk", "/mypath" and all files are extracted to given path.

|LS

Added v2.0.0. Display directories as long names. Max. length depends on the screen mode. Same parameters as |dir can be used.

|TAPE

Added v2.0.0. Switch to tape. Use |SD to switch back to M4 (microSD).

|TIME

Displays current time, if appropriate ntp server was setup.

|NETSET

See above (setup)

|NETSTAT

Will display current status of network connection

|HTTPGET

Download a file from the internet to the current directory.

Parameters are url:port/name. If no port is specified port 80 is used.

Example:

|httpget, "spinpoint.org:80/battro.dsk" or |httpget, "spinpoint.org/battro.dsk" will work.

| HTTPMEM

Downloads a file from the internet to memory, max size at once is 0x4000 bytes, however a offset can be specified, so more can be read in chunks.

Parameters are *url:port/name*, *memory address*, *size to read*

Examples

- |HTTPMEM, "spinpoint.org/cpc/FIST.BIN", &C000, &4000
- with offset: |HTTPMEM, "spinpoint.org/battro.dsk, offset=0x10000", &8000, &1000 ..
This will read from offset 0x10000 into the file.

| M4ROMOFF

Disable M4 rom until next power cycle.

Use this if you want to access your real disc drive or start a CPC+ cartridge.

| M4HELP

Displays first 32 roms. |M4help, romnumber - lists commands commands for given rom.

| ROMSOFF

Added v2.0.5.

Without paramters, all roms of M4 board is disabled until CPC is rebooted or M4 reset is pressed.

Optional paramteres: "except rom number", "reset" (**0** = do not reset, **1** = do reset). With this it's possible to disable all roms except one.

Example: |ROMSOFF, 6, 1 (will disable all other roms than rom 6 and reset the CPC).

| VERSION

Displays current firmware version of ESP8266 and Cortex M4.

| UPGRADE

Downloads latest upgrade if available.

Afterwards you need to powercycle the M4 board, either by the reset switch or turning power off/on.

Upgrade process will start and the board will display again once finished.

| ROMUP

Added v1.0.5.

Upload rom to a given slot. Ie. |ROMUP, "UTOPIA.ROM", 15 . will upload Utopia rom to slot 15

Notice ROMUP only works with files present on the microSD card, not from floppy disc.

| ROMSET

Added v1.0.5.

Used to enable or disable rom (must be uploaded already). `|ROMSET, 15, 0` will disable rom 15. `|ROMSET, 15, 1` will re-enable it.

| ROMUPD

Added v1.0.5.

Will apply rom modifications live (new roms will be flashed into M4 flash, without reboot need)

| GETPATH

Added v1.1.0.

Shows the actual selected path.

E.g. `|GETPATH` shows "m4" when the "A:\m4"-directory is selected.

| LONGNAME

Added v1.1.0.

Shows the complete long file or directory-name.

E.g. `cat` displays "*ROMCON~1.BIN*" then `|LONGNAME, "ROMCON~1.BIN"` shows "*romconfig.bin*".

| UDIR

Added v1.1.0.

No use in plain BASIC.

| WIFI

Added v2.0.2.

Use `|wifi, 0` to turn off WiFi and `|wifi, 1` to turn it back on.

* see more under Developer information.

Note about wildcards: Files use extensions, so when filtering files via wildcards, you typically use `.` format, Directories do not have extensions, so you only need `*` for directories. Valid wildcards are `*` and `?`.

Roms

V1.0.0 (OLD early version!) By default M4 Board rom is mapped as ROM 7, this is for highest compatibility with games/demos/applications. It's possible to map it as another ROM or disable it using the webinterface. Not all CPC6128's will allow to "override" rom7, and the board will keep resetting the CPC if that is the case. To fix this if the boards WiFi isn't setup.

Create a m4 folder in the root of the microSD card and create a plain text file named "config.txt" Put in your WiFi routers detail like this:

```
ssid=dd-wrt
pw=hackaway
name=CPC664
dhcp=0
ip=10.0.0.37
nm=255.255.255.0
gw=10.0.0.1
dns1=8.8.8.8
dns2=8.8.4.4
tz=1
ntp=ntp.inet.tele.dk
```

You need to change SSID and password to match your Router/AP. And use IP numbers that fit your network

Powercycle the M4 board in the CPC. Type the IP number you gave the board into your browser, as <http://ipnumber>. Then webinterface will show up, now go to "Roms" under "M4 Rom Config" set Rom number to ie. 6. Then Go to "Control" and press M4 reset. M4 Board will now work on your 6128.

Better is to download latest upgrade from: http://cpcwiki.eu/index.php/M4_Board

And follow instructions below.

V1.0.1 up This version will default M4 Board ROM to 6, if your CPC supports overriding ROM 7, you should change it in the web interface, as it provides better compatibility. If your CPC6128 does not support overriding ROM 7, a modified lowerrom was made to give virtually same compatibility as using physical rom 7. For download and info to do this: http://cpcwiki.eu/index.php/M4_Board#Various files

The basics of how to use ROMS on CPC and config with M4

CPC464: Rom slot 0: -- This is occupied by BASIC ROM internally in the CPC, it CAN be replaced with the romboard, but it must be with another BASIC rom! Rom slots 1-7: -- Free to use Rom slots 7-31: -- are NOT initialized by the standard system. Use a modified lowerrom (FW316) or booster rom (uploading in a slot below 8) to initialize these roms at startup.

It is highly recommended that you put M4 rom in slot 7, this will give you best possible compatibility with games

CPC664/CPC464+/CPC6128+: Rom slot 0: -- This is occupied by BASIC ROM internally in the CPC, it CAN be replaced with the romboard, but it must be with another BASIC rom! Rom slots 1-6: -- Free to use Rom slot 7: -- Occupied by AMSDOS disc rom internally in the CPC, it CAN be overwritten with ie. parados or other disc rom, if access to disc drive is still wanted. Rom slots 8-15: -- Free to use Rom slots 16-31: -- are NOT initialized by the standard system. Use a modified lowerrom (FW316) or booster rom (uploading in a slot below 8) to initialize these roms at startup.

It is possible to set M4 rom to 7 for best compatibility, however you loose simulataneos disc drive & sd card access, if you wish to keep both M4 and disc drive, set M4 rom to 6.

CPC6128: Rom slot 0: -- This is occupied by BASIC ROM internally in the CPC, it CAN be replaced with romboard, but it must be with another BASIC rom! Rom slots 1-6: -- Free to use Rom slot 7: -- Occupied by AMSDOS disc rom internally in the CPC, it CANNOT be overwritten on most CPC6128, with the exception of MC20C (iirc) motherboard. Do not use! Rom slots 8-15: -- Free to use Rom slots 16-31: -- are NOT initialized by the standard system. Use a modified lowerrom (FW316) or booster rom (uploading in a slot below 15) to initialize these roms at startup.

It's recommended to set M4 rom to 6.

M4 Rom slot: As for uploading a rom in same slot as M4 rom is set to, it's possible, but it will only be active, if you type |M4ROMOFF.

Lowerrom: If "Lower-rom enabled", you must upload a lower rom to the corresponding slot or machine will reset/crash whatever. It is recommended to use ie. slot 31 for lowerrom replacement, because if you decide to disable it, the slot will not be initialized in a normal system config (and lowerrom should not be initialized like a normal rom == crash).

If using a CPC6128 and having problems with game compatibility it is recommend to use the modified lowerrom for M4 and put M4 rom in slot 6. Get modified lowerrom here:

http://www.cpcwiki.eu/index.php/M4_Board For other machines, simply set M4 rom to slot 7, if you want to use disc drive type |M4ROMOFF

Upgrading Basic v1.0 on CPC464 to Basic V1.1: Both basic and lowerrom must be replaced or it will not work, download ie. the roms for CPC6128 from here: http://www.cpcwiki.eu/index.php/ROM_List Upload OS rom to ie. slot 31 and tick "Enable lower-rom" and set it to slot 31. Upload BASIC 1.1 rom to slot 0.

Additional documentation

Extended User Manual by Csaba Toth:

https://www.spinpoint.org/cpc/Amstrad_CPC_M4_Board__Extended_User_Manual.pdf

3D printed cases

- M4 board revision 2.0 EDGE connector, by povvercrazy: <https://www.thingiverse.com/thing:2645945>
- M4 board revision 2.0 EDGE connector, by gschizas: <https://www.thingiverse.com/thing:3280950>
- M4 board revision 2.5B IDC connector, by Rick_B-Lair: <https://www.thingiverse.com/thing:2858771>
- M4 board revision 2.5C EDGE connector, by carlesoriol: <www.thingiverse.com/thing:4829910>
- M4 board revision 2.4, 2.5B & 2.5C, both IDC and EDGE connector, by yawor: <https://github.com/yawor/m4board-enclosure>
- M4 board revision 2.5 EDGE connector, by greeno76: <https://www.thingiverse.com/thing:6846777>

Tools/Programs

cpcxfer a tool to transfer files between PC and CPC wireless from command line. Source can be found here (feel free to improve it, it's very basic): <https://github.com/M4Duke/cpcxfer>

CPCRulez Front end by hERMOL (<https://cpcrulez.fr/forum/viewtopic.php?f=2&t=6426&start=15#p56177>): <https://cpcrulez.fr/f/106x93> (RulezCharge v0.8b8m4)

M4 Front End by Abalore: https://www.cpcwiki.eu/index.php/M4FE_-_A_front-end_for_M4

Disk menu by Joe Halstead, modified by Takis Kalatzis for M4: <http://www.cpcwiki.eu/forum/amstrad-cpc-hardware/amstrad-cpc-wifi/msg134659/#msg134659>

YANCC - Norton Commander Clone by SOS, compatible with M4:
<http://www.cpcwiki.eu/forum/applications/yet-another-norton-commander-clone>

File Launcher by SOS, compatible with M4: [http://www.cpcwiki.eu/forum/applications/filelauncher-\(acmedosm4dos\)](http://www.cpcwiki.eu/forum/applications/filelauncher-(acmedosm4dos))

Menu program for M4 (with program & games download direct from the internet) by rafa32gr:
<http://amstrad.esy.es/>

Front end for M4 (launch programs & games quickly with UI): <http://www.cpcwiki.eu/forum/amstrad-cpc-hardware/front-end-for-m4-board-in-rom/msg168434/#msg168434>

SymbOS with M4 support, allowing you to transfer files to/from real floppy discs, download from internet, chat on IRC and more: <http://www.cpcwiki.eu/forum/applications/symbos-cpc-updates-and-infos/msg137192/#msg137192>

Simple telnet client for M4: <http://www.cpcwiki.eu/forum/applications/telnet-client-for-cpc/>

ANSI telnet client M4EWEN by Fergus Leen: <https://github.com/fergusleen/m4ewenterm>

Navigate and run stuff from your Android phone by Frange: <https://github.com/Frange/Amstrad-M4-Board>

Games

A chess game using M4 to connect over the internet or LAN by Joseman:
[http://www.cpcwiki.eu/forum/programming/cpchessnet-\(developing\)/](http://www.cpcwiki.eu/forum/programming/cpchessnet-(developing)/)

Defence game, with online highscore table for registered version, by Shining: <http://cpc.scifinet.org/>

Other and info on how to improve compatibility: www.cpcwiki.eu/forum/technical-support/the-compatibility-of-the-m4-card/

Resources

Make your own M4 ROM or add additions to the existing, source code can be found here:
<https://github.com/M4Duke/m4rom> It's compiled (assembled) with sdsz80 (part of SDCC package).

Create your own HACK MENU for M4: <https://github.com/M4Duke/m4hackmenu>

Various source code examples for M4 board <https://github.com/M4Duke/M4examples>

Problems?

If by some odd chance, everything goes wrong after making configuration changes. Take the microSD card into a PC and delete the files in m4 directory, take the power of the board, re-insert microSD and set it up again.

Dirty edge connector! If any odd issues, like resets, freeze, graphic glitches etc. Make sure your edge connector is cleaned, it can be root of many problems, if it's not properly cleaned.

Webinterface

To come.

Developer information

The way the hardware works is by the CPC sending commands via a port and a acknowledge via another port (command finished, please process). The Cortex M4 mcu will now process the command (do cross talk with the ESP wifi chip if needed) and respond to the CPC via a buffer inside the M4 rom address space.

Ports:

DATAPORT is 0xFE00 ACKPORT is 0xFC00

Offset 0x3F02 (0xFF02) contains a pointer to the internal rom buffer used for responses. Please read the value from here as the buffer position may change in a later firmware.

Commands are send like this:

```
offset 0 - command size (not including this byte)
offset 1 - command lo
offset 2 - command hi
offset 3.... - data for command.
```

Responses are the same (except size is more like response size, then it can have a datasize too)

Commands:

```
C_OPEN          0x4301      data[0] = mode, data[1] = filename. Return
data[0] = fd (*), data[1] = res.
C_READ          0x4302      data[0] = fd, data[1..2] = read size. Return
data[0] = res, data[1..] = data.
C_WRITE         0x4303      data[0] = fd, data[1..] = data. Return data[0]
= res.
C_CLOSE         0x4304      data[0] = fd. Return data[0] = res.
C_SEEK          0x4305      data[0] = fd, data[1..5] = offset. Return
data[0] = res.
C_READDIR       0x4306      No args. Return data[0] = dir entry
(formatted). Loop until response size == 2.
From v110 b9, one argument can be given data[0]
= max name len (to support lfn).
C_EOF           0x4307      data[0] = fd. Return data[0] == 0 (not eof).
Only valid for read fd.
C_CD            0x4308      data[0...] = "directory name". Return
data[0] = res.
C_FREE          0x4309      Freespace sd card / DSK image. Return
data[0] = "\r\n%iK free\r\n\r\n".
```

```

C_FTELL          0x430A      Implemented v1.0.5. data[0] = fd. Return
data[0-3] = current file pos
C_READSECTOR     0x430B      For diskimages only. data[0] = track,
data[1] = sector, data[2] = drive (not used). Return data[0] = res.
data[1..513] sector data
C_WRITESECTOR    0x430C      For diskimages only. data[0] = track,
data[1] = sector, data[2] = drive (not used) data[3..515] sector data
C_FORMATTRACK    0x430D      Not implemented yet
C_ERASEFILE      0x430E      data[0] = "filename". Return data[0] = res;
C_RENAME         0x430F      data[0] = "newname\oldname": Return
data[0] = res, data[1..] "ascii error".
C_MAKEDIR        0x4310      data[0] = "dirname". Return data[0] = res;
C_FSIZE          0x4311      Implemented v1.0.5. data[0] = fd. Return
data[0-3] = file size
C_READ2          0x4312      Implemented v1.0.7. Same as C_READ, except
without checking (and seeking past) for AMSDOS header.
C_GETPATH        0x4313      Implemented v1.0.8. Return data = current
path in ascii.
C_SDREAD         0x4314      Implemented v1.0.9. Read RAW sdcard sector.
data[0..3] = Physical LBA, data[4] = numsectors (max. 4).
return data[0] = error. data[1..] = sector data
C_SDWRITE        0x4315      Implemented v1.0.9. Write RAW sdcard
sector. data[0..3] = Physical LBA, data[4] = numsectors (max. 4),
data[5..] sector data. Return data[0] = error.
Error codes for C_SDREAD & C_SDWRITE: 0:
Successful, 1: R/W Error, 2: Write Protected, 3: Not Ready, 4: Invalid
Parameter
C_FSTAT          0x4316      Implemented v1.1.0. File stat.
data[0]=file/dir. Return:
data[0] = error (0 = ok).
data[1..4] = File size in bytes.
data[5..6] = Date. bit15:9 Year origin from
1980 (0..127) bit8:5 Month (1..12) bit4:0 Day (1..31)
data[7..8] = Time. bit15:11 Hour (0..23)
bit10:5 Minute (0..59) bit4:0 Second / 2 (0..29)
data[9] = Attrib. 0x01 Read only, 0x02 Hidden,
0x04 System, 0x10 Directory, 0x20 Archive.
data[10..22] = Short name.
data[23..280] = Long name.
C_NMI           0x431D      Implemented v2.0.5. No args. Trigger
NMI/Hackmenu
C_HTTPGET        0x4320      data[0] = "url:port/file". Return data[0] =
"Downloaded...".
C_SETNETWORK     0x4321      data[0] = setupstring (see |netset above).
C_M4OFF          0x4322      No args. Disable M4 rom and reset.
C_NETSTAT        0x4323      No args. Return data[0] = "network status
string".
From v1.1.0 there is a status byte after the
return string, which can be:
0 = Idle.
1 = Connecting.
2 = Wrong password
3 = No accesspoint found
4 = Connection failed

```

5 = Connected (and got IP)
 Rest are unknown status (ERROR!)
 C_TIME 0x4324 No args. Return data[0] = "hh:mm:ss yyyy-mm-dd".
 C_DIRSETARGS 0x4325 data[0] = "folder/match string" (wildcards etc.) Should be used before C_READDIR, if no args, use null string.
 C_VERSION 0x4326 No args. Return current M4 and ESP version strings.
 C_UPGRADE 0x4327 No args. Downloads upgrades from internet, if newer version available.
 C_HTTPGETMEM 0x4328 data[0..1] = size, data[2] = "url:port/file, offset=". Return data[0..1] = downloaded size. (into internal buf, see C_COPYBUF).
 C_COPYBUF 0x4329 data[0..1] = offset into internal buf (0..3FFF), data[2..3] = size. Returns data[0] = data (dest).
 C_COPYFILE 0x432A data[0..] = "srcfile\0destfile\0". Return data[0] = res, data[1..] = ascii error..
 C_ROMSUPDATE 0x432B No Args and return. Updates romchanges made to romconfig.bin and romslots.bin without rebooting.
 C_NETSOCKET 0x4331 Implemented v1.0.9. data[0] = domain, data[1] = type, data[2] = protocol. Return data[0] = socket number or 0xFF (error). Only TCP protocol for now.
 C_NETCONNECT 0x4332 Implemented v1.0.9. data[0] = socket, data[1..4] = ip, data[5-6] = port. Return data[0] = 0 (OK) or ERROR 0xFF.
 C_NETCLOSE 0x4333 Implemented v1.0.9. data[0] = socket.
 C_NETSEND 0x4334 Implemented v1.0.9. data[0] = socket, data[1..2] = size, data[3...] = outgoing data. Return data[0] = 0 OK, 0xFF error.
 C_NETRECV 0x4335 Implemented v1.0.9. data[0] = socket, data[1..2] = receive size (don't flood buffer, max. 0x800). Return data[0] = 0. data[1..2] = actual received size. data[3...] = received data. Look at sockinfo for status.
 C_NETHOSTIP 0x4336 Implemented v1.0.9. data[0..] = hostname string\0. Return data[0] = 1 Lookup in progress. Any other = error. Look in M4rom sockinfo for IP and status
 C_NETRSSI 0x4337 Implemented v1.0.9. No args. Return RSSI value for connection (31 = fail, rest? = signal strength).
 C_NETBIND 0x4338 Implemented v1.0.9. data[0] = socket, data[1..4] = IP number, data[5..6] = port number. return 0 = ok otherwise error.
 C_NETLISTEN 0x4339 Implemented v1.0.9. data[0] = socket. Return return 0 = ok otherwise error.
 C_NETACCEPT 0x433A Implemented v1.0.9. data[0] = socket. Return return 0 = ok otherwise error. Refer to sockinfo for status.
 C_GETNETWORK 0x433B Implemented v1.1.0. No args. Returns the following structure:

```

char name[15+1];           // 0
unsigned char ssid[32];   // 16
unsigned char password[64]; // 48
unsigned int ip;          // 112
unsigned int nm;          // 116
unsigned int gw;          // 120
unsigned int dns1;        // 124

```

```

unsigned int dns2;           // 128
unsigned int dhcp;          // 132
int timezone;              // 136
unsigned char ntpserver[48]; // 140
unsigned char reserved1;    // 188
unsigned char reserved2;    // 189
unsigned char mac[6];       // 190

```

C_WIFIPOW 0x433C Implemented ?. WiFi power data[0] = 0 turn OFF power, 1 = turn ON power.

C_ROMCP 0x43FC Implemented ?. ROM copy within M4 ROMS (RAM) data[0-1] = dest offset, data[2] = src bank (0 for M4 rom, not used yet), data[3-4] = src offset, data[5] = dest bank (0 for M4 rom, not used yet), data[6-7] = size

C_ROMWRITE 0x43FD Implemented ?. Write to ROM (M4 or HACK/NMI rom). data[0-1] = dest offset, data[2-3] = size, data[4] = rom (255 = NMI/HACK ROM, 0 = M4 ROM).

C_CONFIG 0x43FE Write to M4 ROM config area. data[0] = offset in config area (0xFF04+offset), using command size(-3).

C_ROMLOW 0x433D Implemented v2.0.7. Map lower rom : data[0] = 0 system lower rom, 1 = lowerrom from romboard if enabled, 2 = lowerrom from HACK menu.

C_OPEN flags:

For system compatiblity, there is two file descriptors like AMSDOS 1 = read, 2 = write.

However if you OR the open mode with 0x80, you get "real mode" fatfs and can have 10 open file descriptors, this is recommended for any developers.

```

#define FA_OPEN_EXISTING    0x00
#define FA_READ            0x01
#define FA_WRITE           0x02
#define FA_CREATE_NEW       0x04
#define FA_CREATE_ALWAYS   0x08
#define FA_OPEN_ALWAYS     0x10
#define FA_REALMODE        0x80

```

Responses are read from rom by mapping M4 rom.

At offset 0xFF00, there is a link table for things that may move as firmware gets updates:

```

0xFF00       .dw #0x109           ; rom version
0xFF02       .dw rom_response     ; 0x800+some byte buffer for responses
from commands
0xFF04       .dw rom_config       ; Internal config, that is only reset by
power cycle. This buffer can be written with <1 byte size> .dw C_CONFIG
<offset> <data>
0xFF06       .dw sock_info       ; socket structure for NETAPI, works
like read only hardware registers
0xFF08       .dw helper_functions ; useful functions, to have executed in
rom.

```

Socket info structure (ptr read from 0xFF06), there is a total of 4 sockets for use and "socket 0" reserved C_NETHOSTIP. User sockets are 1-4.

offset:

```

(socket*16)+0   status   : current status 0=idle, 1=connect in progress,

```

```

2=send in progress, 3=remote closed connectoion,
4=wait incoming (accept), 5=dnslookup in progress, 240-
255 = error code
(socket*16)+1  lastcmd : last command updating sock status 0=none, 1=send,
2=dnslookup, 3=connect, 4=accept, 5=recv, 6=error handler
(socket*16)+2  received: (2 bytes) - data received in internal buffer
(ready to get with C_NETRECV)
(socket*16)+4  ip_addr : (4 bytes) - ip addr of connected client in
passive mode
(socket*16)+8  port : (2 bytes) - port of the same..
(socket*16)+10 reserved: (6 bytes) - not used yet (alignment!).

```

See source code examples here: <https://github.com/M4Duke/M4examples>

Special RSX commands

|GETPATH From ASM, you can use the getpath-function with the register: A=255 and DE=Bufferaddress for Path Then the RSX has no screen-output, the actual path is written to the memory-address, which DE points to.

|LONGNAME From ASM, you can call the longname-function with the register: A=255 and DE=Bufferaddress for file/dirname and IX=normal RSX-Pointer Then the RSX has no screen-output, the longname is written to the memory-address, which DE points to.

|UDIR This functions makes a "CAT" and, after one directory-entry is read from the SD-Card, call an ASM-Program from the user, until the last entry is read. With your own function you can build the CAS_CATALOG_BUFFER. It will be called with:

- HL = Location of null terminated file/folder name string
- B = File flag (1 = directory, 0 = file)

The length of the file is witten to: Low-Word: &AD86 High-Word: &AD88

You can patch the addresses &bea3, &bea4,&bea5 with a jump to your own function (e.g. "JP &A000") When the memory address &bea3 is "0" the function does nothing. The M4-ROM initialized this address with "0" during hard-reset. This function is similar to the same function of ACME-DOS, with the main difference, that directory-names could have more than 11 Bytes length (max. 16 Bytes).