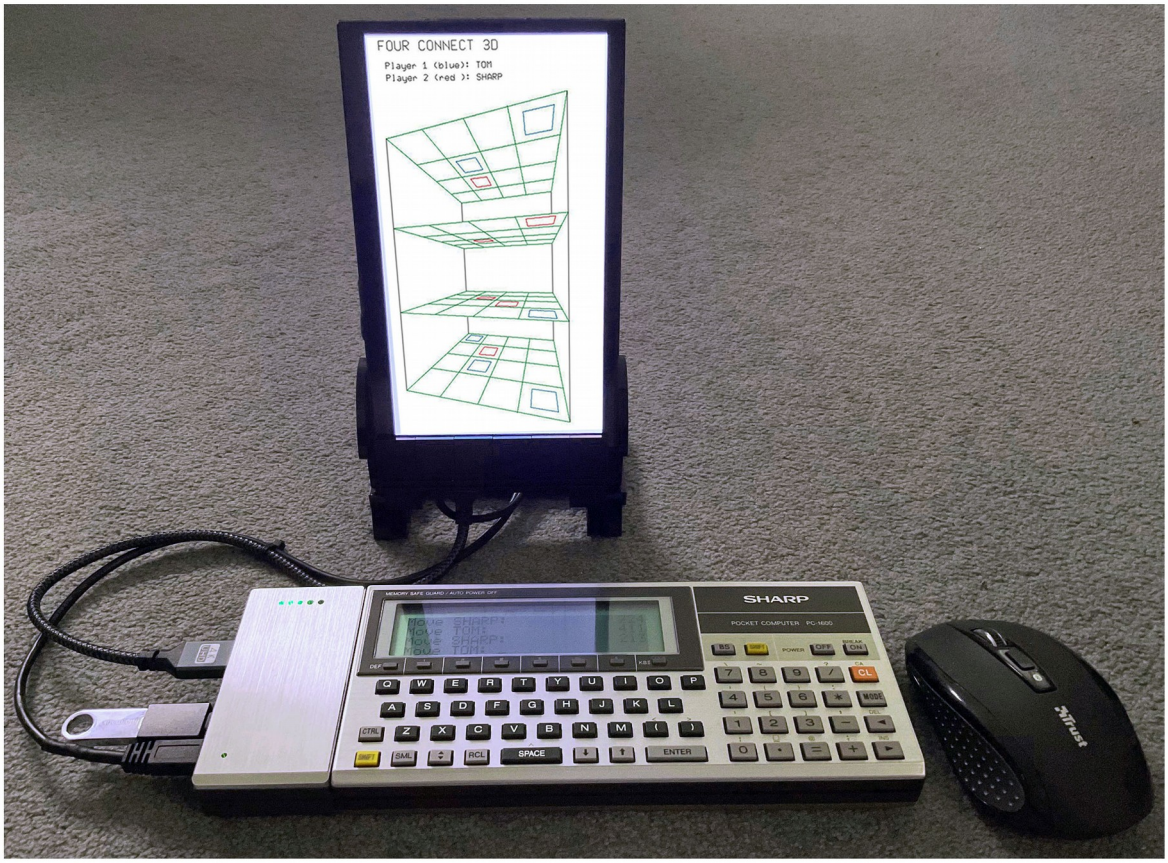
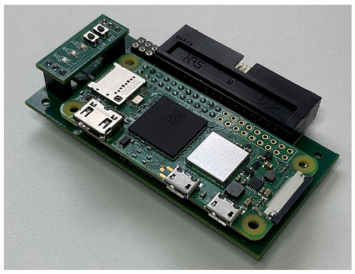
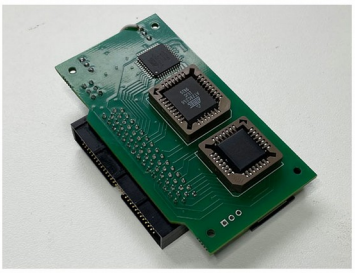
SHARP PC-1600
CE-1600P Compatible HDMI Plotter
USB Memory Stick Reader/Writer







This module is the modern equivalent of the CE-1600P DinA4 plotter and the CE-1600F disk drive. It fully integrates with the PC-1600 OS.
Connect any HDMI monitor and watch the plotter drawing text and graphics on the screen. **All BASIC commands of the CE-1600P are supported 1:1!** Connect a mouse to zoom in and out, move the plot or take a screenshot, even while the plotter is drawing.

Additionally the module presents a connected memory stick as a fully integrated file device S3, just like another RAM disk. So read and write access is completely natural using the commands `FILES`, `[B]SAVE`, `[B]LOAD`, `COPY`, `KILL` etc.
Two extra commands are supported to list and navigate directories on the stick: `CDIR` (change directory) and `LDIR` (list subdirectories).

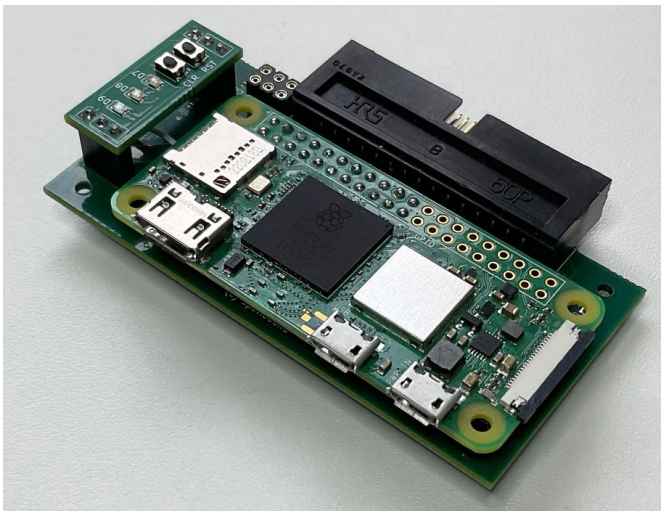
The module comes with a 3D printed shell and a CNC-milled and brushed aluminium cover.

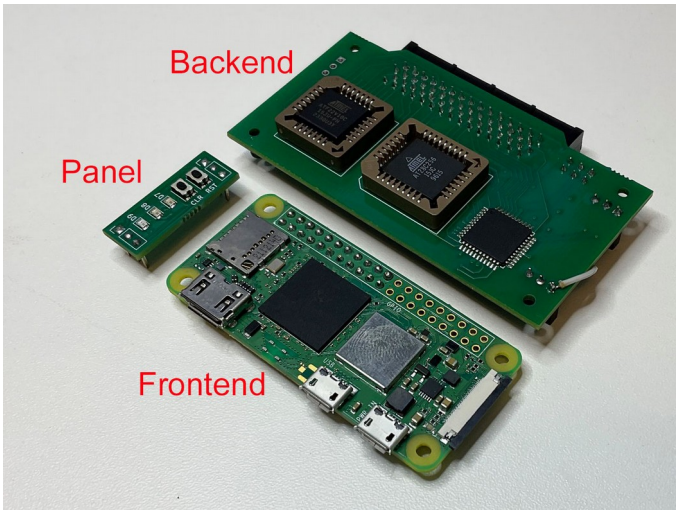
Table of Contents - Extract

Generic MEP rev4 Hardware Features

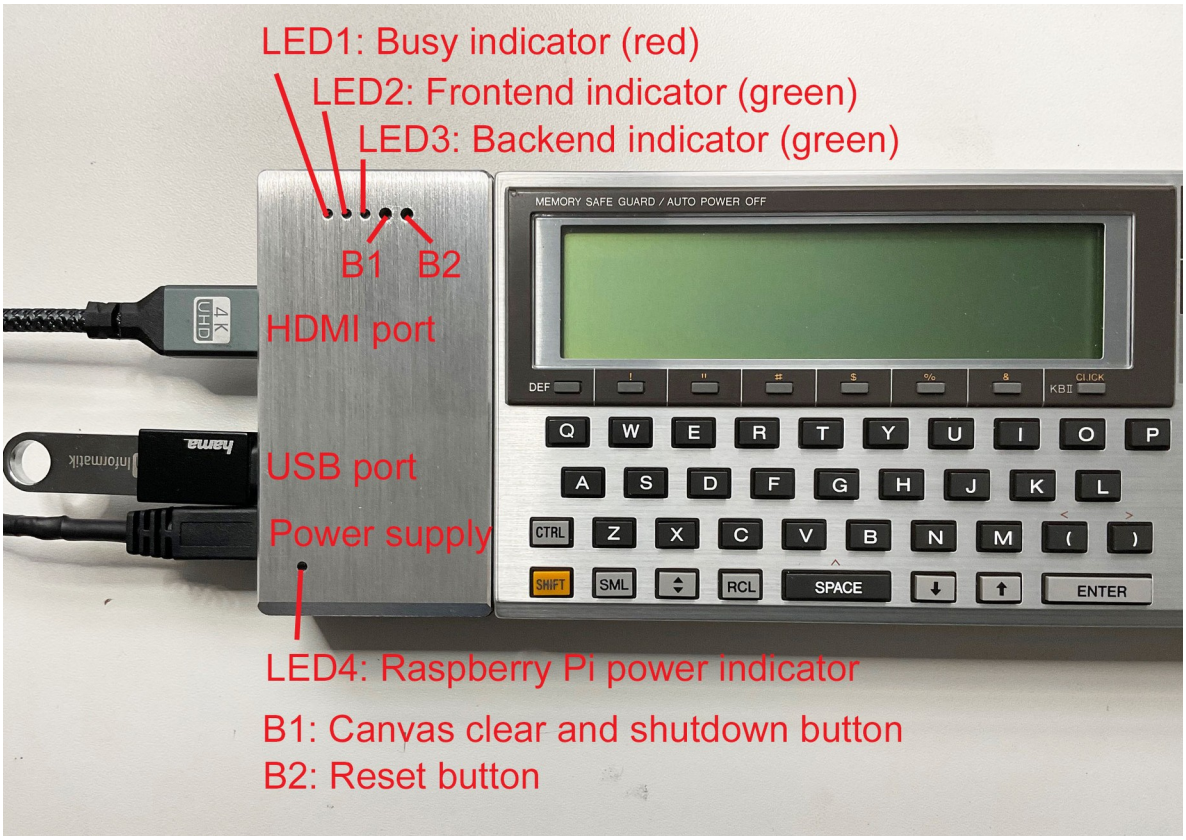
- Modular design with a Raspberry Pi as frontend hardware.
  UART Rx/Tx communication between frontend and backend up to 500k baud.
- Backend:
  - ATmega32A microcontroller for realtime I/O-port communication (INP/OUT) with the PC-1600.
  - 28C256 EEPROM with socket. Contains PC-1600 ROM extensions.
  - 22V10C PLD with socket. Provides addressing for ROM extensions and I/O-ports.
  - Onboard SPI socket for programming of the microcontroller.
  - SHARP compatible 60pin bus connector.
- Frontend:
  - Raspberry Pi Zero 2 W with GPIO-pins soldered to fit the backend jack.
  - Ports: 1x mini HDMI, 1x micro USB power supply, 1x micro USB jack.
- Panel:
  - 3 indicator LEDs and 2 push buttons.
- Compatibility: **Only SHARP PC-1600**.

PC-1500/A not supported – do not connect the module to a PC-1500/A !





LED Indicators and Push Buttons



- **Power supply:**
  The modules Raspberry Pi frontend cannot be powered by the PC-1600 but needs its own 5V USB power supply. Connect a respective USB cable with a micro USB plug.
  As soon as a power supply is connected, the module starts booting, regardless of whether a PC-1600 is connected or switched on.
- **HDMI port:**
  Connect a full HD (i.e. 1920x1080) capable monitor via mini HDMI plug here. The monitor needs its own power supply. You can configure the module for other resolutions (see Application Configuration).
  Do not use a non-HDMI port with an adaptor (e.g. DVI or VGA) on the monitor side. However USB-C typically works together with a video capture card (see Appendix B – Tablet as Monitor).
- **USB port**
  Connect a USB device via micro USB (or an adaptor) here. The minimal setup could be a USB stick or a mouse, but you can connect a USB hub with different devices as well, e.g. a mouse, a keyboard and a USB stick.
  To use an USB-stick as file device for the PC-1600 it needs to be formatted with FAT32 and the volume name must be S3.
- **Wireless connectivity:**
  Since the modules frontend is a Raspberry Pi, it can be connected to WiFi and Bluetooth devices (mouse, keyboard).

- **LED1 – Busy indicator (red):**
  This LED lights up when the PC-1600 'talks' to the module. There are two different appearances:
  1. Flickering: Plotter or USB-stick access commands are performed.
  2. Permanentely on: The PC-1600 tries to communicate with the module, but its main software application is not running (yet). The PC-1600 is blocked then and waits until the module is ready (see Module Boot and Shutdown).
- **LED2 – Frontend indicator (green):**
  This indicator serves two purposes:
  1. Flashing 3 times: This indicates a successful module boot (see Module Boot and Shutdown).
  2. Permanentely on during a session: This indicates that a USB stick named S3 has been detected and is ready for file operations from the PC-1600 (see File Device S3 Operation).

  During boot or shutdown (and even after shutdown) of the module LED2 may be on as well, but that has no specific meaning.
- **LED3 – Backend indicator (green):**
  Flashing 3 times: The modules backend has successfully (re)started. This happens when the PC-1600 is switched on or resetted (see B2 – Reset button).
- **LED4 – Raspberry Pi power inidicator:**
  This is the built in power indicator of the modules frontend. For the modules operation there are two specific meanings:
  1. On/flickering at boot time or during a session: Valid USB power supply detected.
  2. Off after module shutdown: Shutdown finished. You can disconnect the USB power supply (see Module Boot and Shutdown).
- **B1 – Canvas clear and shutdown button:**
  This push button must be operated with a respective tool, since it is accessible through a hole in the aluminium cover only.
  B1 serves two purposes:
  1. Press: Clear the virtual plotter paper aka canvas (see Canvas Clear Button).
  2. Hold for more than 4 seconds: Perform a module shutdown (see Module Boot and Shutdown).
- **B2 – Reset button:**
  This is a peripheral reset button that must be operated with a respective tool, since it is accessible through a hole in the aluminium cover only. It can operate in two modes:
  1. Press: Normal reset of the PC-1600 and the module.
  2. Hold ON-key of the PC-1600 while pressing: Hard/total reset of the PC-1600 and reset of the module.

Module Boot and Shutdown

...

The recommended startup sequence is as follows:

1. Connect the module to the PC-1600 and all desired accessory, except the USB power supply. Switch on the monitor if connected via HDMI.
2. Connect the USB power supply to the module. The Raspberry Pi starts booting.
   - LED4 flashes and indicates operation.
   - Wait until LED2 has flashed 3 times (frontend started).
3. Switch on the PC-1600.
   - LED1 shows activity during boot.
   - Wait until PC-1600 prompt (>) appears on the LCD display.

...

The recommended shutdown sequence is as follows:

1. Switch off the PC-1600.
2. Press and hold the push button B1 for more than 4 seconds. The Raspberry Pi frontend is performing a shutdown.
   - Wait until LED4 has permanentely turned off.
3. Disconnect the modules power supply.

Plotter Operation

When a full HD monitor is connected to the module when it boots (see above), a CE-1600P plotter emulation is started and a screen representing a blank Din A4 paper is shown.

There are two symbols shown that do not belong to the plot itself:

- A gray crosshair pointer, that marks the current plotter head position.
- A gray triangle as the mouse pointer.

As a first test you could type the command `TEST` at the PC-1600. Watch the crosshair pointer move and draw the typical test pattern consisting of four squares in black, blue, green and red.

Standard BASIC Plotter Commands

All plotter BASIC commands of the original CE-1600P are supported by the module, e.g.:

`LPRINT, LLIST, CSIZE, LF, COLOR, TEXT, GRAPH, LLINE,` etc.

See the CE-1600P manual for a full description
(e.g.from here: https://www.sharp-pc-1600.de/PDF/Plotter_Eng_Manual2.pdf).

The cassete interface commands of the CE-1600P are not supported.

Canvas Clear Button

Pressing (not holding) the button B1 (see LED Indicators and Push Buttons) serves as a "paper" clear function.

Mouse Functions

When a mouse is connected to the module (either via USB or via Bluetooth) the following functions are available:

- **Left mouse button hold plus mouse move:**
  Drag the virtual paper over the screen, e.g. to move to areas that are currently clipped because of a zoom.
- **Mouse wheel:**
  Zoom in and out at cursor position.
- **Right mouse button click:**
  Reset virtual paper position and zoom.
- **Middle mouse button click:**
  If an S3 USB stick is mounted the whole virtual DinA4 paper is exported to `S3:/PLOTXXXX.PNG`, where `XXXX` is a 4 digit number starting from `0000` and incrementing after each screenshot. The numbering restarts with each session from `0000` and existing files are overwritten.

Assembler Programming - IOCS Printer API

The module is also fully compatible with the printer assembler API of the CE-1600P.

See https://www.sharp-pc-1600.de/PDF/PC1600TechnicalReference.pdf for details.

File Device S3 Operation

The module works as a USB flash drive interface that fully integrates with the PC-1600 operating system by providing a native file device named "S3:" or "Y:" respectively.
So the module provides a seamless data and program exchange between a PC/MAC and the PC-1600 as well as a mass storage capability for the PC-1600.

In order to use the USB stick file access you do not need a monitor connected to the modules HDMI port. When no monitor is connected, the module starts in headless mode. But you should not unplug or plug a monitor during a session. In headless mode plotter commands are still executed in background.

Before using the module for USB stick access you need to format it with FAT32. When formatting, choose "S3" as the new device name. Otherwise the USB stick will not be detected by the module.

As long as a S3 USB stick is detected during a session, LED2 lights up permanentely (see LED Indicators and Push Buttons).

During the session you may connect or remove the S3 USB stick (hot plug) but not during file operations (i.e. LED1 shows activity).

Standard BASIC File Commands

You can use the following standard BASIC file commands to access the connected USB memory stick. Instead of the device name S3 you can use Y as well:

- `FILES"S3:"`
  `FILES"S3:<search-pattern>"`
  Search patterns may include wildcards `*,?`
  Examples: `FILES"S3:"  FILES"S3:*.BAS"  FILES"S3:A*.???"`
- `[B]SAVE"S3:<filename>"[,A]`
  Examples: `SAVE"S3:TEST.BAS",A  BSAVE"S3:TEST.BIN",...`
- `[B]LOAD"S3:<filename>"`
  Examples: `LOAD"S3:TEST.BAS"  BLOAD"S3:TEST.BIN"`
- `COPY"<device>:<filename>"TO"<device>:<filename>"`
  S3 can be used as source- or target-device name or both
  Examples:
  ```
  COPY"S3:TEST.BAS"TO"S2:TEST.BAS"
  COPY"S2:TEST.BAS"TO"S3:TEST.BAS"
  COPY"S3:TEST.BAS"TO"S3:TEST1.BAS"
  ```
- `KILL"S3:<filename>"`
  Example: `KILL"S3:TEST.BAS"`
- `NAME"S3:<old-filename>"AS"S3:<new-filename>"`
  Example: `NAME"S3:TEST.TXT"AS"S3:TEST.BAS"`
- `OPEN"S3:<filename>" FOR [OUTPUT|INPUT] AS #<fileno>`
  `PRINT#<fileno>,<data>`
  `INPUT#<fileno>,<variables>`
  `CLOSE#<fileno>`
  Example:
  ```
  10 OPEN "S3:MYFILE1.TXT"FOR INPUT AS #1
  20 OPEN "S3:MYFILE2.TXT"FOR OUTPUT AS #2
  30 INPUT #1,I$:PRINT #2,I$
  40 CLOSE #1:CLOSE #2
  ```

The following BASIC file commands are not supported by the module. If you use them with the device names "S3:" or "Y:" they yield an `ERROR 158`.

- `DSKF"S3:"`
  `SET"S3:<filename>",["P" | " "]`
  `OPEN"S3:<filename>" FOR APPEND AS #<fileno>`

Of course you can access the files on the USB stick via a PC or MAC too (read & write).

A minor restriction of the module is the fact, that only one file for read and one for write can be open simultanously. Setting the PC-1600 system variable `MAXFILES` to higher values than 2 has no effect on S3.

The module is restricted to the 8.3 (FAT) file format like the PC-1600. Files or directories with non 8.3 names or lower case or special characters are filtered and not "seen" by the PC-1600.

Directories – Additional BASIC Commands

The ROM-extension provides two additional, non-standard BASIC commands that give access to (sub-)directories on the connected flash drive:
`CDIR` (i.e. "change directory") and `LDIR` (i.e. "list directories").

CDIR"<path>"

There is no specification of a device since this command only operates on S3.

Like the `FILE` command the `CDIR` command outputs information to the LCD-display. In this case it's the prompt which tells the current selected (sub-)directory in UNIX-like notation.

Examples:
Here S3 is assumed to be the device name of the flash drive, assigned when formatted. Furthermore this example is a sequence of commands, starting in the root directory.

Command	Semantics	Prompt
`CDIR"."`	Show current dir (here: root)	`S3:/>`
`CDIR"UTIL"`	Relative path, one dir down	`S3:/UTIL>`
`CDIR"../GAMES"`	One dir up, one down	`S3:/GAMES>`
`CDIR"/DEV/ASM"`	Absolute path, two dirs down	`S3:/DEV/ASM>`
`CDIR".."`	One dir up	`S3:/DEV>`
`CDIR"/"`	Absolute path to root	`S3:/>`

The selected directory however acts like a context for the standard BASIC file commands (see above). So if you navigate to different directories on the flash drive, the `FILE` command will report the content of that directory only. This context concept holds for all standard BASIC commands and the IOCS file routine (see Assembler Programming - IOCS File API), since the PC-1600 operating system has no notion of directories. In consequence you cannot e.g. `LOAD` from a different directory than the currently selected one (e.g. `LOAD"S3:/UTIL/TEST.BAS"` is not possible).

This isolation of the directory concept from the PC-1600 OS is very important to maintain compatibility with existing PC-1600 programs and the OS itself. Consequently (sub-)directories are not 'seen' by the `FILE` command.

The directory structure itself has to be created on a modern computer, but the module can navigate through that structure with the aid of the `CDIR` command.

LDIR

The additional command `LDIR` has no parameters and lists the subdirectories (not files) of the directory currently selected by `CDIR`. Output format and usage is equivalent to the standard `FILES` command.

So you can use a USB stick as a structured mass storage for the PC-1600 !

Usage with DiskWorks

DiskWorks (DW.BIN) is the file browser software for the SHARP PC-1600.

You can download it e.g. from here: https://www.sharp-pc-1600.de/Down_Maschine.html
or seek Fehler: Referenz nicht gefunden.

The module registers S3 as the main device name and Y as the secondary device name. So you can use existing PC-1600 software like DiskWorks as far as it supports access to Y (originally the device name Y is associated with the CE-1600F).

Here is a short usage introduction:

DiskWorks default screen showing content of S2

Press function key ‘&’ to select 2nd device

Device selection menu

Press function key ‘$’ to select Y
(i.e. the alternative name for S3)

Swap primary and secondary device

Press function key ‘KBII/CLICK” to swap S2 and Y

Browse selected directory on the USB stick and copy files from/to S2

Assembler Programming - IOCS File API

This module provides ROM extensions for the PC-1600. The file api extension registers to the standard IOCS file routine of the PC-1600. In fact this is the only mandatory integration with the PC-1600 OS that has to be implemented by a PC-1600 peripheral file device. All standard BASIC file commands rely on that very same IOCS file routine, which is a very elegant and open design by the SHARP engineers of the 1980's.

By the way, this hooking to the standard IOCS file routine is the foundation of compatibility with existing PC-1600 file browser applications like DiskWorks.

The API of the standard IOCS file routine is a `CALL` to a specific ROM address:

```
FILE &01DE
```

Parameters: C-reg: function code, DE-reg: FileControl Block (FCB) pointer
Function codes:

```
&0F  OPEN FILE
&10  CLOSE FILE
&11  SEARCH FIRST
&12  SEARCH NEXT
&13  DELETE FILE
&14  SEQUENTIAL RD
&15  SEQUENTIAL WR
&16  CREATE FILE
&17  RENAME FILE
```

In order to access the USB stick through this API you need to set the 4-byte device name (FDVNO0..3) of the FCB to "`S3  `" or "`Y   `" respectively.

For further information about the IOCS file API and the structure of the FCB please refer to the PC-1600 Technical Reference chapters 3.3.1 and 3.3.2:
https://www.sharp-pc-1600.de/PDF/PC1600TechnicalReference.pdf

It is also possible to `CALL` the core of the `CDIR` command (see above), which is of course not part of the standard IOCS file routine:

```
CDIR #7,&4020
```

Parameters: DE-reg: path string, B-reg: size of path string
Returns: prompt string at `&FB10`, C-flag: success/error, BASIC error no in `&F89B`

The prompt string is limited to 26 characters (i.e. one LCD display line) and is terminated by `CR` (i.e. `&0D`).

Furthermore you can put `SEARCH FIRST` and `SEARCH NEXT` into a directory list mode instead of the standard file list mode by calling:

`DIRMODE #7,&4023` (no parameters)

To switch back to file list mode call:

`FILEMODE #7,&4026` (no parameters)

The modules additional BASIC command `LDIR` is based on this mechanism too.

Application Configuration

The main software component of the MEPrev4 is installed on the Raspberry Pi frontend. This program contains the CE-1600P emulation and the file access logic for the USB stick. It is named `lprintusb_drm` and located in the directory `/home/sharp/MEP`.

`lprintusb_drm` has several command line options to change the screen resolution, scaling and fitting of the virtual plotter paper, as well as screen rotation options. The title page of this document shows a configuration with a screen rotation by 270°.

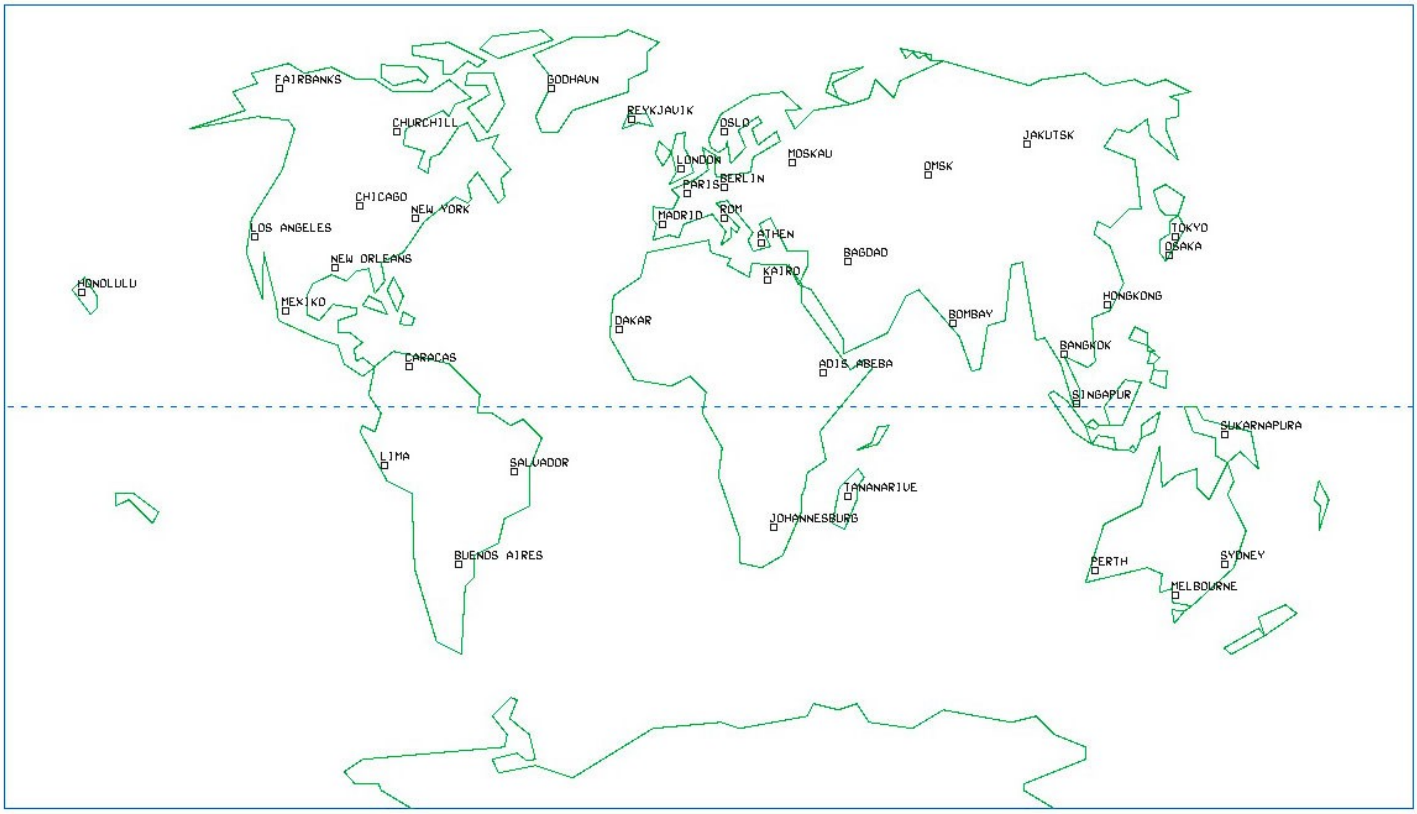
Accessing and changing command line options for the main application:

1. Setup the required accessory for Raspberry Pi configuration.
2. Boot the module as described in Module Boot and Shutdown.
3. When the plotter emulation has started, press ESC on the USB keyboard.
   You are now at the command line interface (Linux shell).
4. Type `./lprintusb_drm -?` on the USB keyboard.
5. All available command line parameters are shown and described. You can try them just by invoking the program including some options.
   E.g. `./lprintusb_drm -fitf -1600x900 -rotate:1` uses a screen resolution of 1600x900, fits the whole virtual DinA4 paper on the screen and rotates the screen by 90° clockwise.
   Wheras `./lprintusb_drm` without any option uses a resolution of 1920x1080 and displays the virtual paper so that one plotter unit is represented by one pixel with no screen rotation (i.e. landscape). If you specify a screen resolution that is not supported, the program starts in headless mode.
6. You can switch on the PC-1600 for testing.
7. Switch off the PC-1600. Use the ESC-key on the USB keyboard to terminate the emulation and return to the command line. Repeat from 5. until you have found a desired configuration.
8. Type `cd ..`
9. Type `nano .bashrc`
10. Scroll down to the end of the file. There you find the autostart configuration of `lprintusb_drm` including the current options (default: no options).
11. Change the options to your preference.
12. Save changes by pressing `Ctrl-O` then `Return` and exit the editor by pressing `Ctrl-X`.
13. Type `shutdown -P now`
14. Wait until LED4 (Raspberry power supply LED) is off.
15. Unplug the USB power supply.

The application configuration is finished and the MEPrev4 module is operational again.

Appendix A – Sample Screenshot

WORLDMAP.BAS (rotated by 90°)



Appendix B – Tablet as Monitor

It is possible to utilize a tablet with an USB-C input as a monitor for the module.
The following picture shows this scenario with an iPad Pro:

To connect a tablet with USB-C input to the module you need specific accessories:

1. A video capture card with HDMI input and USB-C output.
2. An HDMI cable with a standard HDMI and a mini HDMI plug.
3. An app for the tablet, that can turn it into a monitor.
   A simple example for the iPad Pro would be the app *MoniCon*.