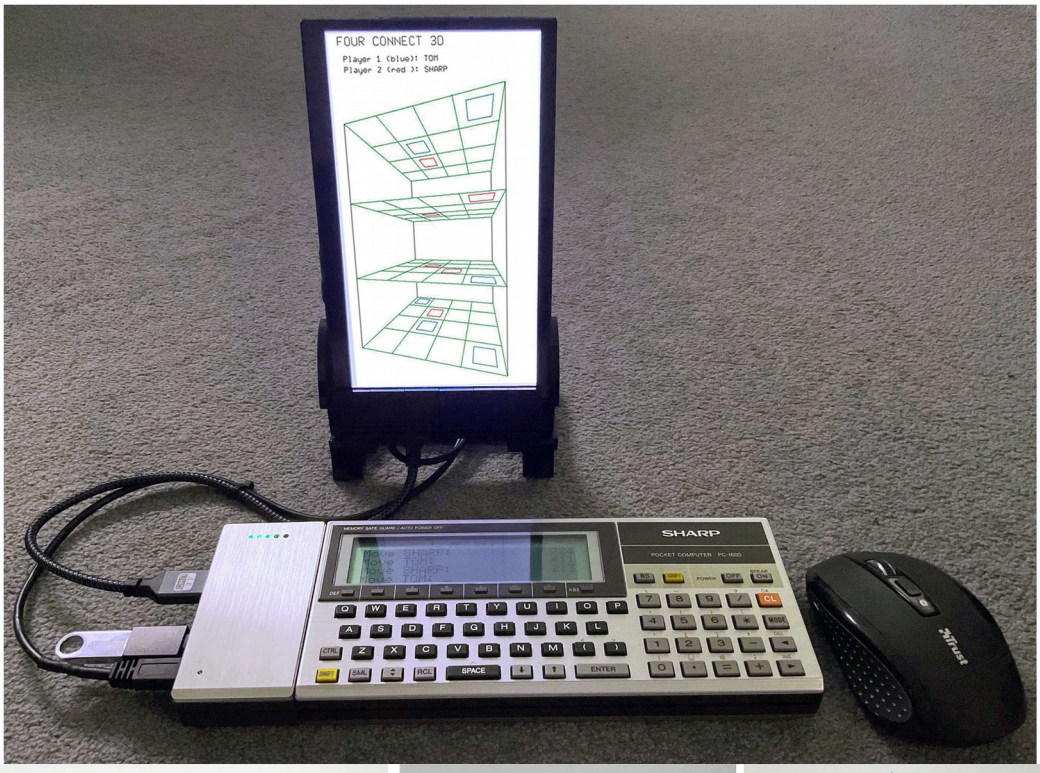
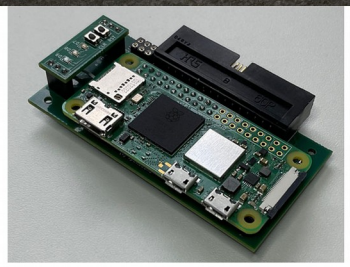
SHARP PC-1600
CE-1600P Compatible HDMI Plotter
USB Memory Stick Reader/Writer





This module is the modern equivalent of the CE-1600P DIN A4 plotter and the CE-1600F disk drive. It fully integrates with the PC-1600 OS.
Connect a monitor via HDMI 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. You can even plot without a monitor and save the image.

An external keyboard (USB or Bluetooth) can be used to type on the PC-1600.

The module supports an additional, 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 S3:
`CDIR`(change directory) and `LDIR` (list subdirectories).
S3 can either be mapped to a mounted USB stick or to a local directory on the modules SD card. From there it can even be exposed as a **WiFi network drive** so you have direct remote access from a Windows, macOS, iOS or Linux system.

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

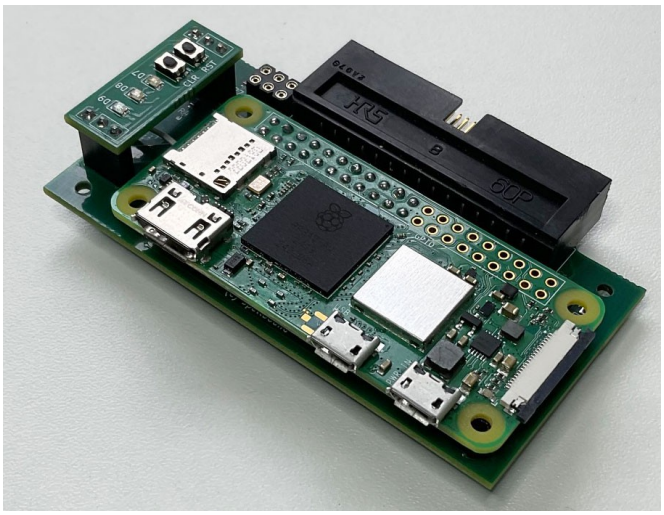
Table of Contents

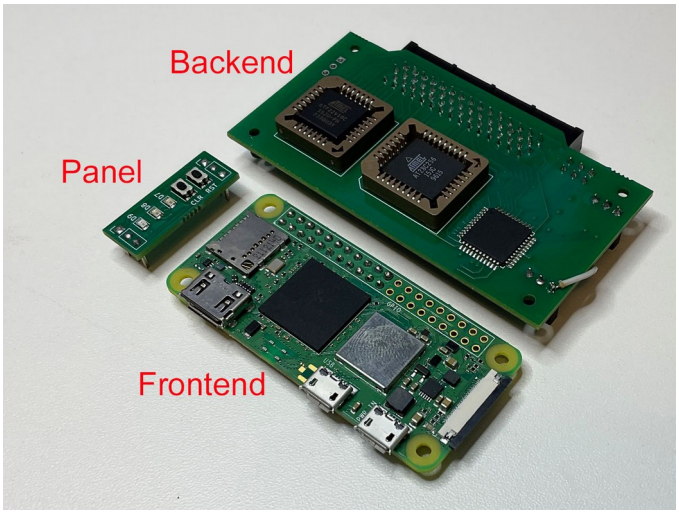
Generic MEPrev4 Hardware Features....3
LED Indicators and Push Buttons....4
Required and Recommended Accessories....6
Module Boot and Shutdown....9
Plotter Operation....10
  Standard BASIC Plotter Commands....10
  Canvas Clear Button....10
  Mouse Functions....10
  Additional BASIC Plotter Commands....11
  Assembler Programming - IOCS Printer API....11
File Device S3 Operation....12
  Standard BASIC File Commands....13
  Directories – Additional BASIC File Commands....14
  Usage with DiskWorks....15
  Assembler Programming - IOCS File API....16
External Keyboard Support....17
  Keyboard Layout Change....17
  Restrictions....18
Headless Mode....18
Raspberry Pi Configuration....19
Application Configuration....21
File Device S3 on internal SD Card....22
Appendix A – S3 NAS Configuration....23
Appendix B – List of Additional BASIC Commands....26
Appendix C – Sample Plot....26
Appendix D – Tablet as Monitor....27
Appendix E – Symptoms, Causes and Solutions....28
Copyright....29
Contact....29

Generic MEPrev4 Hardware Features

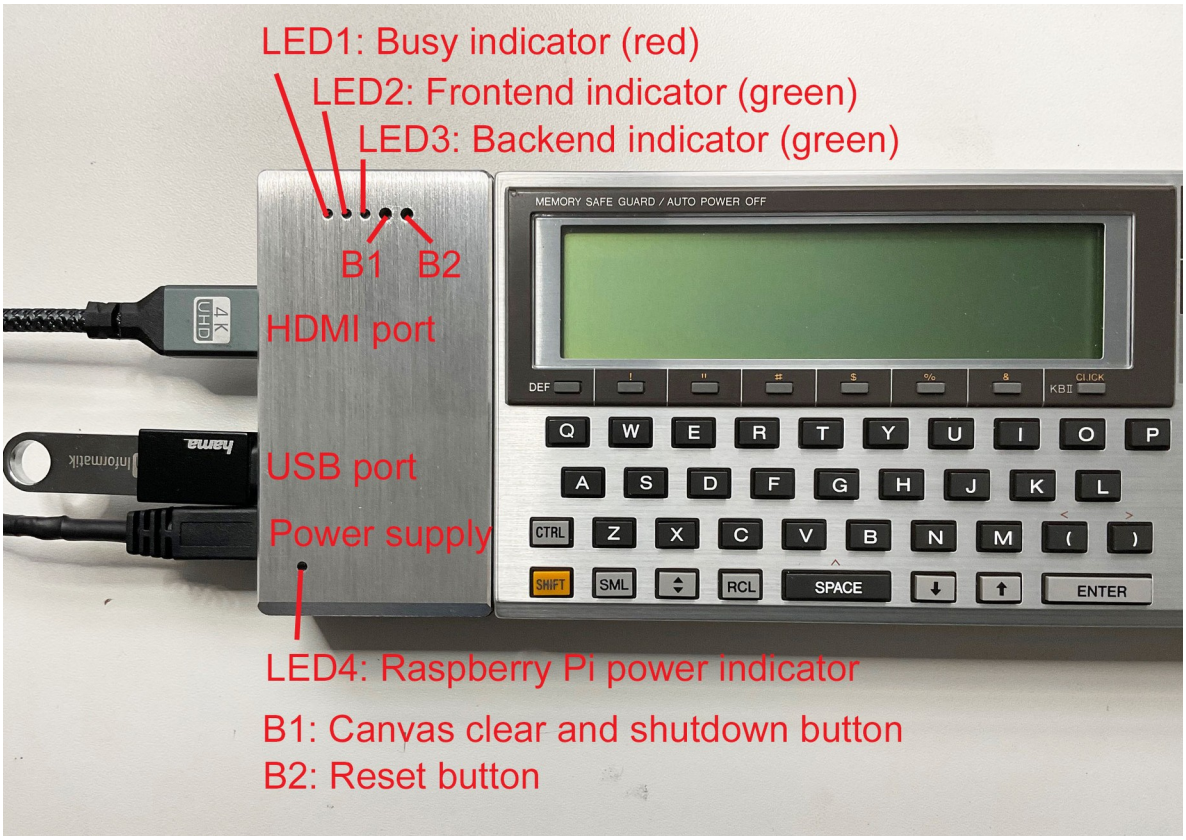
- Modular design with a Raspberry Pi as frontend hardware.
  UART Rx/Tx communication between frontend and backend up to 500k baud.
- Backend:
  - ATmega32A microcontroller for realtime I/O-port communication (INP/OUT) with the PC-1600.
  - 28C256 EEPROM with socket. Contains PC-1600 ROM extensions.
  - 22V10C PLD with socket. Provides addressing for ROM extensions and I/O-ports.
  - Onboard SPI socket for programming of the microcontroller.
  - SHARP compatible 60-pin 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.
  - Bluetooth connectivity.
  - WiFi connectivity.
- 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 D – Tablet as Monitor).
- **USB port**
  Connect a USB device via micro USB (or an adaptor) here. A minimal setup could be a USB stick or a USB 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). See Raspberry Pi Configuration for details.

- **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 S3 file device 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.

Required and Recommended Accessories

Depending on the usage scenario there are different requirements and options for connecting the module to respective accessories.

Do not connect the module to a CE-1600P. The ROMs would be conflicting, since the module is a modern replacement of that vintage hardware.

It is not possible to connect USB devices or hubs to the power supply port. So if you want to use more than one cabled USB device simultaneously, you need to connect a USB hub to the USB port.

1. **Required base accessory**
   In **every scenario** you need:
   - USB power adaptor or power bank.
   - USB power cable with micro USB plug for connection with the modules power supply port.
2. **Wireless accessory**
   The module supports Bluetooth devices – i.e. a keyboard and a mouse. But at least for the initial configuration of a wireless setup, you need a cable bound USB setup (see **Required setup for Raspberry Pi configuration** below).
3. **HDMI-plotter related accessory**
   You can plot without a connected HDMI monitor (called headless mode) and even save screenshots, but if you want to see the plots like on a CE-1600P, you need:
   - Full HD monitor with own power supply.
   - HDMI cable with mini HDMI plug (or adaptor) for the module side.
   - Optional for plot zoom & pan: USB or Bluetooth mouse.
4. **USB stick related accessory**
   A fully integrated file device S3 for the PC-1600 can either be provided by a mounted USB stick or by a local directory on the SD card of the modules frontend (Raspberry Pi). A monitor / HDMI connection is not reqired for S3 operation.
   For USB stick use, you need:
   - USB-A to micro USB adaptor or adaptor cable or USB hub with micro USB plug.
   - USB stick formatted with FAT32 and volume name S3 connected to the adaptor.
5. **Keyboard related accessory**
   The module features external keyboard input (USB or Bluetooth) for the PC-1600, so you can type on a modern keyboard just like on the micro keyboard of the PC-1600 (see External Keyboard Support for binding of PC-1600 specific keys). An external keyboard is also required for any type of configuration. For a USB based setup you need:
   - USB-A to micro USB adaptor or adaptor cable or USB hub with micro USB plug.
   - USB keyboard.

6. **Required setup for Raspberry Pi configuration**
   Linux experts can configure WiFi or Bluetooth device connections via the command line, but the normal user would do this via the Raspberry Pi desktop user interface (see Raspberry Pi Configuration).
   In order to enter and use the Raspberry Pi desktop you need a keyboard and a mouse. At least for the initial configuration of a wireless setup, you need a cable bound USB setup.
   You can leave the PC-1600 unconnected to the module.
   - Full HD monitor with own power supply.
   - HDMI cable with mini HDMI plug (or adaptor) for the module side.
   - USB hub (with at least 2 ports) with a micro USB plug or adaptor for the module side.
   - USB keyboard connected to the hub.
   - USB mouse connected to the hub.

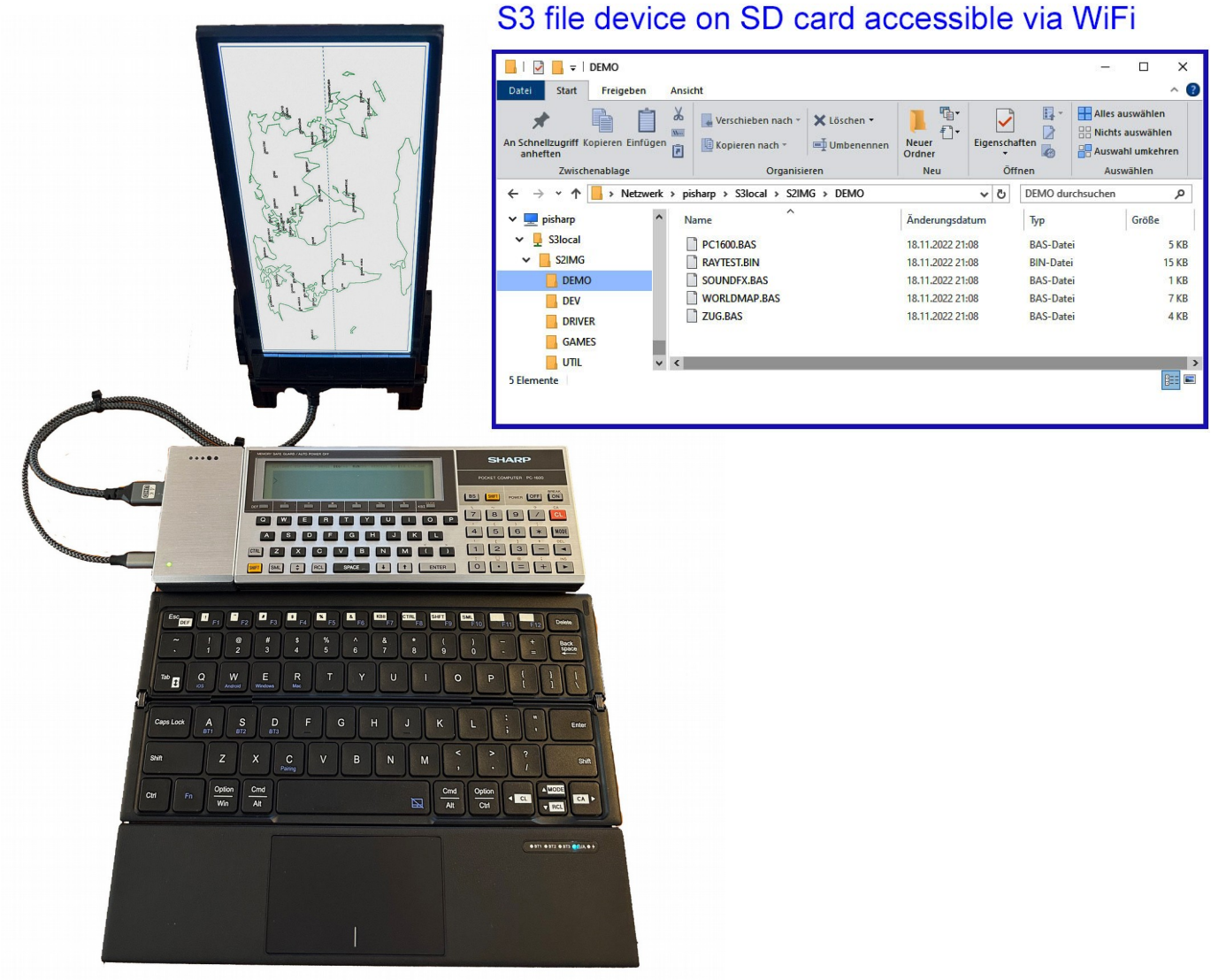
The following picture shows a possible fully cable bound setup.

General note for connector hardware selection:

The physical port placement of the Raspberry Pi Zero 2 restricts the usable plugs and adaptors, since the power supply port and the USB port are placed quite close together. So you need very narrow micro USB plugs in order to fit side by side.

To be on the safe side you can look after Raspberry Pi Zero compatible accessory.

In contrast the following setup shows a fully featured wireless and portable configuration:



This sample setup consists of:

- Mini HDMI monitor.
- Small switchable power bank attached to the back side of the monitor for supply of the monitor and the module.
- Foldable Bluetooth keyboard with integrated track pad for mouse functionality (plot zoom & pan, Raspberry Pi configuration via the Pi-OS desktop).
- HDMI cable.
- Micro USB power cable.
- S3 file device for the PC-1600 mapped to a local directory on the Raspberry Pi's SD card.
- S3 directory configured as WiFi network drive, so it is directly accessible (read and write) by any SMB client – here the standard Windows Explorer.

Module Boot and Shutdown

Please read this section carefully before you use the module for the first time.

The Raspberry Pi Zero 2 frontend of MEPrev4 module is powered by USB power supply. However the PC-1600 has its own power supply (batteries or EA-160). Both are independent from each other, so both can be booted separately.

If you switch the PC-1600 on before the modules frontend has booted, the PC-1600 is blocked and waits, indicating this via LED1. As soon as the module has booted, the PC-1600 continues its operation. This is intended and normal behavior.

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 standard power off routine of the PC-1600 communicates with the connected peripherals. So if the module is shutdown first, the PC-1600 will be blocked and waits (LED1 on) when switched off until the module has been started again. This is normal behavior. Therefore the PC-1600 should always be powered off first.

The modules frontend is a Linux based computer by itself. In order to avoid data loss or even corruption of its OS, you should always perform a shutdown instead of pulling the USB power supply, just as with any modern computer.

In most cases pulling the power supply without a previous shutdown will be harmless, but this is basically a matter of chance.

The modules ROM extension provides a special BASIC command for a convenient power off / shutdown procedure of the PC-1600 and the module:

1. At the PC-1600 type `SHUTDOWN`. The PC-1600 powers off immediately.
2. The Raspberry Pi frontend performs a regular shutdown afterwards.
3. Wait until LED4 has permanentely turned off, then disconnect the power supply.

Alternatively you can shutdown the module separately via push button B1:

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.
3. Wait until LED4 has permanentely turned off, then disconnect the 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.

When no monitor is connected, the module boots in headless mode (see Headless Mode) – i.e. the plotter emulation runs in background.

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. It works in headless mode too.

Mouse Functions

When a mouse is connected to the module (USB or Bluetooth) and the module is not running in headless mode, 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 file device is mounted (see File Device S3 Operation), the whole virtual DIN A4 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.

Additional BASIC Plotter Commands

The ROM-extension of the module provides additional, non-standard BASIC commands that can be used in RUN and PRO mode as well as from within a BASIC program.

LCLR

The additional command `LCLR` has no parameters and clears the plotting canvas (i.e. virtual DIN A4 paper). This command is equivalent to pushing the canvas clear button (see above). The cursor is set to the top line. However the cursors x-position and the current `TEXT/GRAPH` mode are not affected.
The command works in headless mode too.

LSAVE"<base-name>"

The additional command `LSAVE` has one string parameter that specifies the base name for the file to where the current plotting canvas content is to be exported. That base name is extended automatically by `.PNG`, so do not specify any file extension within the base name.

This command requires a mounted file device S3 (see File Device S3 Operation). LSAVE works in headless mode too. The file <base-name>.PNG is written to the currently selected S3 directory (see Directories – Additional BASIC File Commands). Existing files are overwritten.

Example:
The sequence

```
LCLR
TEST
CDIR"/PLOTS"
LSAVE"PLOT1"
```

clears the canvas, plots the CE-1600P test pattern, navigates to the directory `S3:/PLOTS` and saves the canvas to the image file `PLOT1.PNG` into that directory.

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.

Also the core of the additional BASIC commands, which are of course not part of the standard plotter IOCS routine, can be invoked via assembler:

`LCLR #7,&4029`

Parameters: none, returns: none.

`LSAVE #7,&402C`

Parameters: DE-reg: base name, B-reg: size of base name
Returns: C-flag as success indicator, Error code in A-reg, BASIC error no in &F89B.

File Device S3 Operation

The module supports an additional file device named "S3" or "Y" respectively that fully integrates with the PC-1600 operating system. From the PC-1600's perspective it behaves just like another RAM disk.

There are two alternative ways of mounting S3:

1. Connect a prepared USB stick to the USB port of the module.
2. Create a directory named `S3local` in the home directory of the default user of the Raspberry Pi frontend (see File Device S3 on internal SD Card).

So, the module provides both data exchange with a modern computer (without requiring the RS-232 interface) and mass storage capability for the PC-1600.

In order to use S3 file access, a connected monitor is not mandatory (see Headless Mode).

Before using a USB stick as an S3 file device 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).

If you have both, a connected S3 USB stick plus the S3local directory, then the USB stick overshadows S3local so that the PC-1600 only 'sees' the USB stick. When you unplug the USB stick, S3 'falls back' to S3local in this case. LED2 indicates the current state of the S3 mapping: If its on, S3 is mapped to the USB stick. If its off and you have S3 access, then its mapped to S3local. Switching between these S3 mappings (by plugging or unplugging an S3 USB stick) causes the module to reset the currently selected S3 subdirectory (see Directories – Additional BASIC File Commands) to the respective root directory.

The S3local directory can be configured as a NAS (Network Attached Storage) – i.e. a remote directory in your WLAN that can directly be accessed (read and write) from any Windows, macOS, Linux, iOS or Android system in the same WLAN. So with this option you can have seemless integration of a PC-1600 file device with your preferred modern platform. You can e.g. edit a BASIC file directly on S3 from your laptop with no file transfer at all.

For S3 configuration as a NAS see Appendix A – S3 NAS Configuration.

Standard BASIC File Commands

You can use the following standard BASIC file commands to access the S3 file device. 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 a S3 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 File 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:
This example is a sequence of commands, starting in the root directory of S3.

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 subdirectory structure itself has to be created on a modern computer (USB stick) or on the modules Raspberry Pi (S3local), 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 S3 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 Contact.

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
&1A  SET DMA
```

In order to access S3 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.

External Keyboard Support

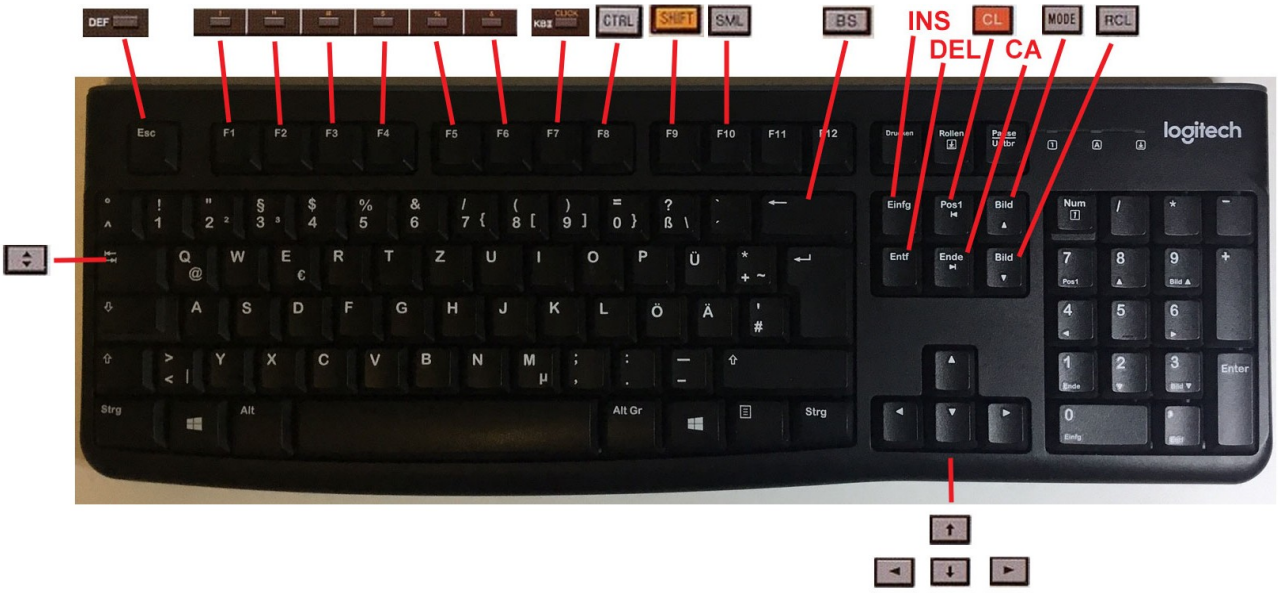
If a keyboard (USB or Bluetooth) is connected to the MEPrev4 module while it is in normal operation, keystrokes are transmitted via the 60-pin bus interface and injected into the key buffer of the PC-1600 by the modules ROM. This allows you to type on the external keyboard just as you would on the integrated keyboard.

The control keys Shift, Ctrl, Alt, Alt Gr, Caps Lock, Num etc. are not sent directly to the PC-1600, but change the subsequent input as usual. The same applies to language-specific modifier keys such as accent aigu (´), accent grave (`) or accent circonflexe (^).

Language-specific keys like ä, ö, ü, ß, é, è, ê, ç and many others are supported as long as they are part of the PC-1600 code page (i.e. IBM CP437).

Keys that have no equivalent in the PC-1600 code page are ignored.

Finally there is a specific mapping for the PC-1600 special keys, which is shown in the following figure. Since not all keyboards have an INS key, Ctrl-V can be used alternatively.



Keyboard Layout Change

The default language/layout setting of the module for external keyboards is the US standard layout. To change this (or to pair a Bluetooth keyboard) you can enter Raspberry Pi desktop mode as described in Raspberry Pi Configuration.

Then go to

Main menu -> Raspberry symbol -> Preferences-> Raspberry Pi Configuration -> Localisation -> Set Keyboard

and make the desired changes.

To make the module usable again, you need to go back to command line mode as described in Raspberry Pi Configuration as well.

Restrictions

The module supports external keyboards for every functionality of the PC-1600 that relies on its internal keybuffer. However there are certain routines that bypass the keybuffer and perform a direct scan of the *internal* keyboard matrix of the PC-1600. These routines cannot, of course, be supported by the module for external keyboard use. Namely:

BASIC

- `INKEY$`

Assembler

- `KEYDIRECT: #0,&0175`
- `KEYSTRB: #0,&0178`

Headless Mode

The module can run in headless mode – i.e. without a graphical user interface (GUI) that displays the plotter canvas (virtual DIN A4 paper).

The module starts in headless mode, if

- no HDMI monitor is connected to the module at boot time,
- a connected monitor is technically incompatible (e.g. non-native HDMI connection),
- a connected monitor does not support the screen resolution defined via the `lprintusb_drm` command line parameters (either default or explicitely specified – see Application Configuration),
- `lprintusb_drm` is started with the option `-headless` (see Application Configuration). In this case the headless mode is entered even if a HDMI monitor is connected. Instead of the graphical user interface you can watch the log output of the program then.

All BASIC plotter and file commands are supported in headless mode too. So you can e.g. run a program written for the CE-1600P and save the plotting result as a .PNG file to the S3 file device even without a connected monitor (see Additional BASIC Plotter Commands).

Raspberry Pi Configuration

The frontend of the MEPrev4 module is a standard Raspberry Pi Zero 2 installed with the standard Raspberry OS (a special Linux distribution). On top there are installed two programs belonging to the MEPrev4. These are started automatically.

So you can configure the MEPrev4 frontend just like any other Raspberry Pi. You may want to connect it to Bluetooth devices like a wireless mouse and keyboard in order to get rid of the cables and the USB hub. Or you can connect it to WiFi in order to configure S3 as a network drive (see Appendix A – S3 NAS Configuration).

For the MEPrev4 the Raspberry Pi frontend must be configured for command line mode without the desktop/graphical user interface. But for a user friendly connectivity configuration you can enter desktop mode and switch back afterwards for normal module operation.

This section describes the necessary steps to enter desktop mode and switch back to command line mode.

Configure desktop mode:

1. Setup the required accessory for Raspberry Pi configuration – see item 6 in Required and Recommended Accessories. But do not connect a PC-1600 to the module.
2. Boot the module as described in Module Boot and Shutdown.
3. When the main app has started, press Ctrl-X on the keyboard. You are now at the command line interface (Linux shell).
4. Type `sudo raspi-config`
5. Go to `System Options -> Boot / Auto Login` and activate `Desktop Autologin`. Depending on the installed OS version `Boot` and `Auto Login` may be separated in two different submenus. Leave `Auto Login` unchanged in that case and just set `Boot` to `Desktop`. Then `<Finish>` and reboot by `<Yes>`.
6. The Raspberry Pi reboots and starts the graphical desktop.

Device configuration:

In the top menu you can e.g. activate WiFi or Bluetooth and connect the desired devices.

These are standard Rasperry OS procedures and you'll find respective tutorials online if needed.

Configure command line mode:

The module cannot operate in desktop mode, so you need to go back to command line mode again:

1. Open a terminal window.
   ATTENTION:
   The start of a terminal window automatically launches the modules main app (`lprintusb_drm`) in that window, but that fails with an error, because the main app cannot run in desktop mode. Just ignore this and continue using the terminal window.
2. Type `sudo raspi-config`
3. Go to `System Options -> Boot / Auto Login`.
4. Activate `Console Autologin`. Depending on the installed OS version `Boot` and `Auto Login` may be separated in two different submenus. Leave `Auto Login` unchanged in that case and just set `Boot` to `Console`. Then `<Finish>` and reject reboot by `<No>`.
5. Type `shutdown -P now`
6. Wait until LED4 (Raspberry power supply LED) is off.
7. Unplug the USB power supply.

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

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 S3. 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, fitting of the virtual plotter paper, as well as screen rotation options (see title page foto).

Accessing and changing command line options for the main application:

1. Setup the required accessory for Raspberry Pi configuration – see item 6 in Required and Recommended Accessories.
2. Boot the module as described in Module Boot and Shutdown.
3. When the main app has started, press Ctrl-X 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 DIN A4 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 (see Headless Mode). You can even force headless mode with the `-headless` option.
6. You can switch on the PC-1600 for testing.
7. Switch off the PC-1600. Type Ctrl-X 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.

File Device S3 on internal SD Card

The additional file device S3 (respectively Y) can be provided not only by a USB stick mounted to the module, but alternatively by a local directory on the SD card of the Raspberry Pi. The behavior is described in File Device S3 Operation.

This directory must be named `S3local`, and it must be created in the home directory of the default user (sharp). It is not automatically created by the software.

Here is a description of the necessary steps for this setup.

Creating the S3local directory:

1. Setup the required accessory for Raspberry Pi configuration –
   see item 6 in Required and Recommended Accessories.
   But do not connect a PC-1600 to the module.
2. Boot the module as described in Module Boot and Shutdown.
3. When the main app has started, press Ctrl-X on the keyboard.
   You are now at the command line interface (Linux shell).
4. To navigate to the respective home directory type `cd /home/sharp`
5. To create the required directory type `mkdir S3local`
6. To set permissions type `chmod 777 S3local`
7. Optional: you can create a subdirectory structure for S3local by using the `cd` and `mkdir` commands. E.g.:
   ```
   cd S3local
   mkdir GAMES
   mkdir DEMO
   mkdir DEMO/PLOTTER
   ```
   ATTENTION:
   Subdirectory names must be in 8.3 format, upper case and free of special characters or white spaces in order to be 'seen' by the modules LDIR and CDIR commands.
8. Type `shutdown -P now`
9. Wait until LED4 (Raspberry power supply LED) is off.
10. Unplug the USB power supply.

You cannot access S3local from the PC-1600 when an S3 USB stick is mounted to the module.

To make advanced use of S3local, you can configure it as a WiFi network drive for remote access as described below.

Another advantage of this solution is that you can edit the subdirectory structure of S3local directly from your preferred device (Windows, macOS, iOS, Linux, etc.).

Appendix A – S3 NAS Configuration

This section describes how to install and configure the Linux Samba server for remote access of the S3local directory (see above) as a NAS (Network Attached Storage). This is a bit advanced.

Install and configure Samba for remote access of S3local:

1. Setup the required accessory for Raspberry Pi configuration –
   see item 6 in Required and Recommended Accessories.
   But do not connect a PC-1600 to the module.
2. Enter the Raspberry Pi's desktop mode as described in Raspberry Pi Configuration, connect the Raspberry Pi to your WLAN, re-configure command line mode and reboot.
3. When the main app has started, press Ctrl-X on the keyboard.
   You are now at the command line interface (Linux shell).
4. To update your system and install Samba type
   ```
   sudo apt-get update
   sudo apt-get install samba samba-common-bin
   ```
   This may take a while.
5. To check the status of the service type
   ```
   sudo systemctl status smbd
   sudo systemctl status nmbd
   ```
6. Now lets rename/backup the original Samba config file. Type
   ```
   sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.org
   ```
7. Next create and edit a new config file. Type
   ```
   sudo nano /etc/samba/smb.conf
   ```
8. Edit the file so that it matches this content exactly:
   ```
   [global]
   workgroup = WORKGROUP
   security = user
   client min protocol = SMB2
   client max protocol = SMB3
   [S3local]
   comment = MEPr4-S3local
   path = /home/sharp/S3local
   writable = yes
   browsable = yes
   # for macOS/iOS
   vfs objects = catia fruit streams_xattr
   fruit:metadata = stream
   # mask for files: rw-rw-r--
   create mask = 0664
   # mask for dirs: rwxrwxr-x
   directory mask = 0775
   ```

9. Save changes by pressing Ctrl-O then Return and exit the editor by pressing Ctrl-X.
10. To check the file for syntax errors type
    `testparm -s`
    and re-edit if there are errors.
11. Now you need to set a password for network access for user sharp. Type
    `sudo smbpasswd -a sharp`
    and type in a password of your choice.
12. Shutdown the module by `shutdown -P now`
13. Connect a PC-1600 and boot again.

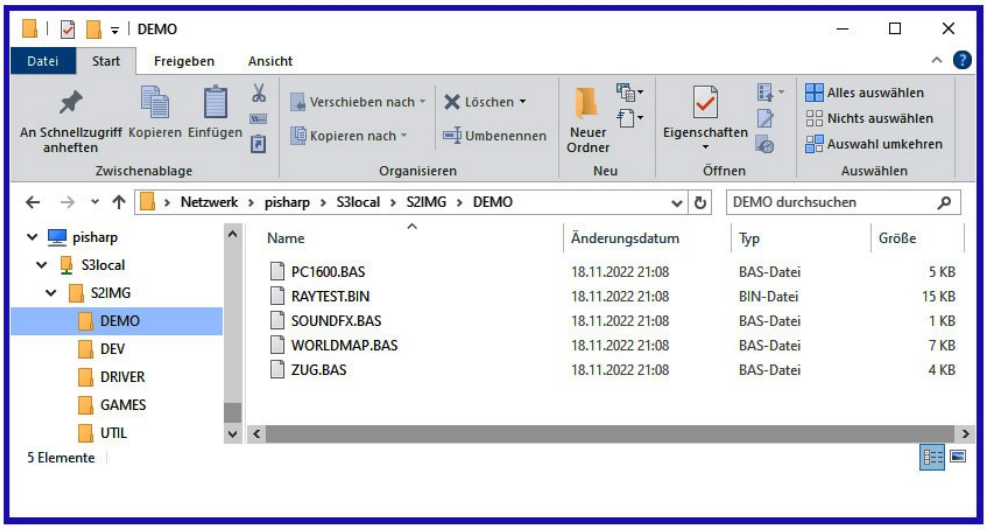
The S3local directory should be accessible now via any SMB client (as long as the module is powered on). The hostname of the module is pisharp. Always connect as user sharp.

Windows and macOS/iOS have integrated SMB support. You need to enter the network password for user sharp you have assigned during the setup above to get access to the network drive.

Windows:
At the Windows Explorer's navigation bar just type \\pisharp\S3local.
The Explorer will ask you for the user (sharp) and password.

macOS/iOS:
In the File app select 'connect with server' and enter smb://pisharp-local/S3local as address, check 'connect as registered user' and enter user (sharp) and password.

S3 file device on SD card accessible via WiFi



For test you could enter a 'hello world' program on the PC-1600 and save it by
`SAVE"S3:HELLO",A`

After a short time you should see the file in the Explorer or File app. Open the file with an editor, make some changes and save it. Now load back the file to the PC-1600 by
`LOAD"S3:HELLO"`

You should see the changes at the PC-1600.

ATTENTION:

There is a known issue with the WiFi Power Management of the Raspberry Pi Zero:

After a period of inactivity, the driver puts the WLAN chip into sleep mode. While it should wake up immediately upon a network request, this fails to work with many routers, resulting in a total loss of connectivity for the session. A reboot is usually required to restore the connection.

If you experience this behavior, it is highly recommended to disable WiFi Power Management on your Raspberry Pi. Here is how to do it:

Disable the WiFi Power Management:

1. Setup the required accessory for Raspberry Pi configuration –
   see item 6 in Required and Recommended Accessories.
   But do not connect a PC-1600 to the module.
2. Boot the module as described in Module Boot and Shutdown.
3. When the main app has started, press Ctrl-X on the keyboard.
   You are now at the command line interface (Linux shell).
4. Create and edit a new power management config file. Type
   `sudo nano /etc/NetworkManager/conf.d/wifi-powersave.conf`
5. Edit the file so that it matches this content exactly:
   ```
   [connection]
   wifi.powersave = 2
   ```
6. Save changes by pressing Ctrl-O then Return and exit the editor by pressing Ctrl-X.
7. Shutdown the module by `shutdown -P now`

Appendix B – List of Additional BASIC Commands

The following list is a comprehension of all additional BASIC commands that the modules ROM extension provides in alphabetical order:

`CDIR` – Change S3 directory. See Directories – Additional BASIC File Commands.

`LCLR` – Clear plotter canvas. See Additional BASIC Plotter Commands.

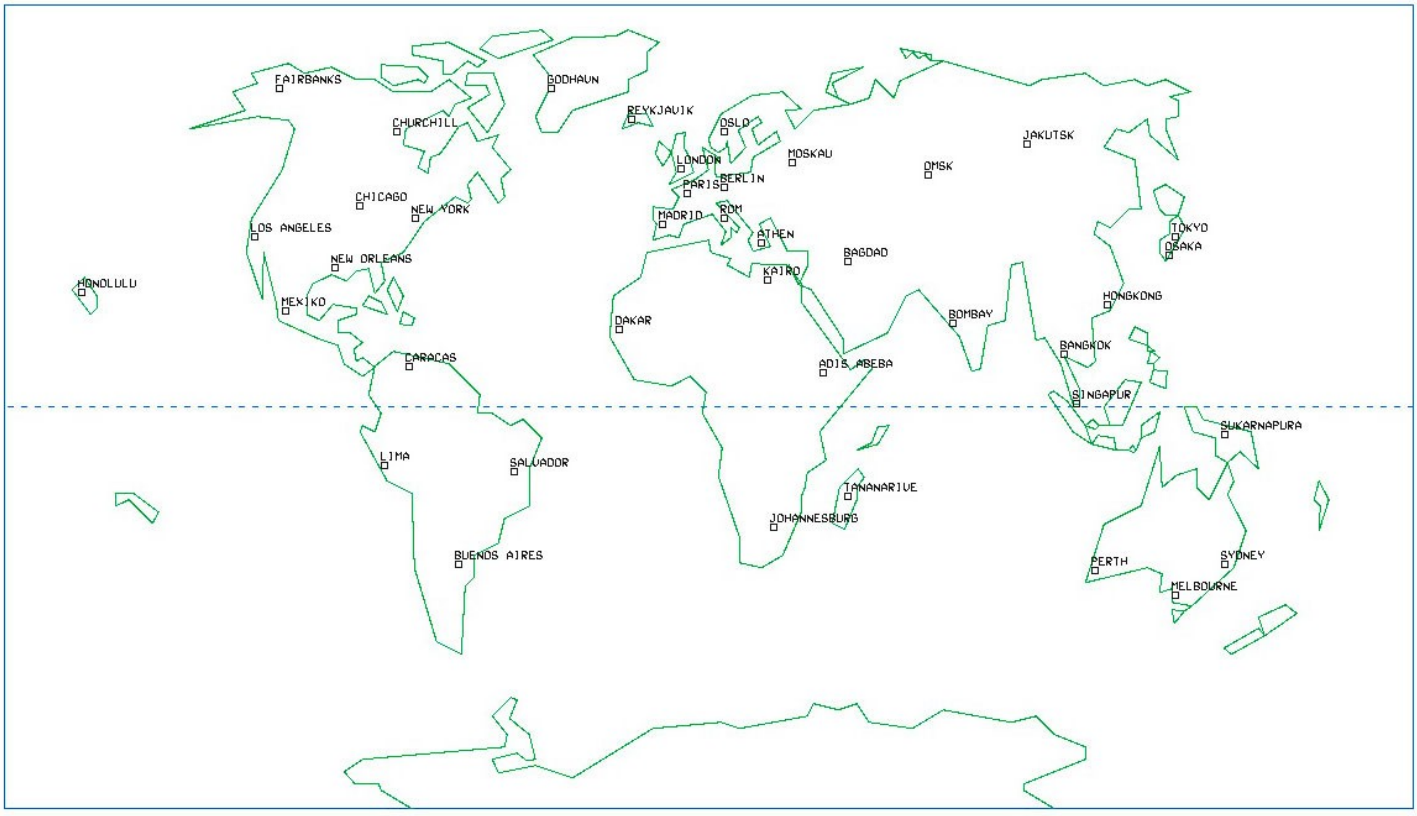
`LDIR` – List S3 subdirectories. See Directories – Additional BASIC File Commands.

`LSAVE` – Save plotter image to S3. See Additional BASIC Plotter Commands.

`SHUTDOWN` – Shutdown the PC-1600 plus the module. See Module Boot and Shutdown.

Appendix C – Sample Plot

WORLDMAP.BAS (rotated by 90°)



Appendix D – 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*.

Appendix E – Symptoms, Causes and Solutions

Symptom	Possible Causes	Solutions
The device detection indicator (LED2) does not light up after inserting a USB stick.	The USB stick is not formatted with FAT32.	Format the USB stick with the FAT32 file system.
	The volume name of the USB stick is different from S3.	(Re)name the volume to S3.
Certain files or directories on S3 are not 'seen' by the PC-1600.	The names are non 8.3 or they contain lower case or special characters.	This is intended behavior. Eventually rename those files or directories.
LED1 lights up permanentely and the PC-1600 is inoperable.	The modules main software application is not running while the PC-1600 is on and tries to perform an operation on the module (e.g. boot, reset, plotting, file-i/o or power off). In this case the PC-1600 is blocked and waits until the module is ready (again).	This is intended and normal behavior for synchronization. (Re)start the module to resolve. Always power off the PC-1600 first and then shutdown the module.
After booting the module with a connected monitor, the plotter emulation window does not start. Instead there are logs on the screen and a line near the bottom says `INFO: entering headless mode.`	If the preceeding log line says `ERROR: no compatible HDMI-monitor found`, either the monitor or a connected adaptor (e.g. DVI or VGA) is not compatible with the module.	Shutdown the module (see Module Boot and Shutdown). Connect a (different) monitor via a native HDMI port. Ideally the monitor is capable of running full HD (i.e. 1920x1080).
	If the preceeding log line says `ERROR: screen resolution ...x... not supported by monitor/hardware`, the desired screen resolution, either default (i.e. 1920x1080) or configured via command line parameter (see Application Configuration), is not supported by the monitor, the Raspberry Pi or the GUI library.	Test and configure another screen resolution setting via the respective command line parameter of the program and the autostart config in the .bashrc file (see Application Configuration).

Copyright

Contact

www.silicium.org/forum -> PM to user spellbound