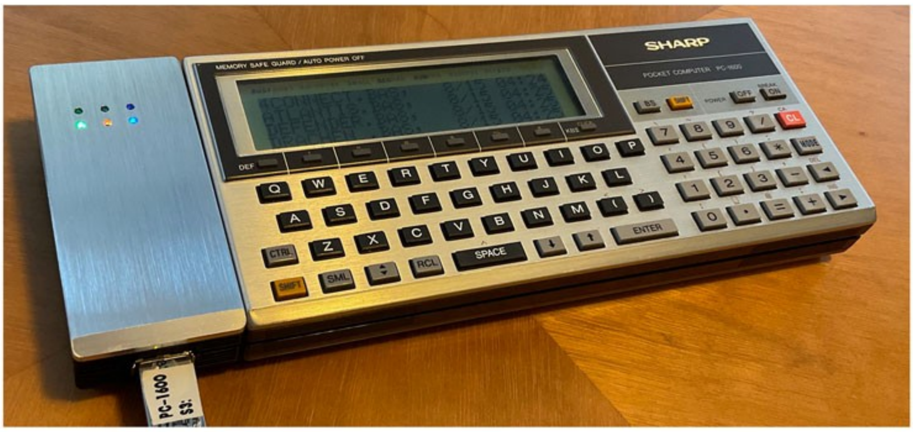
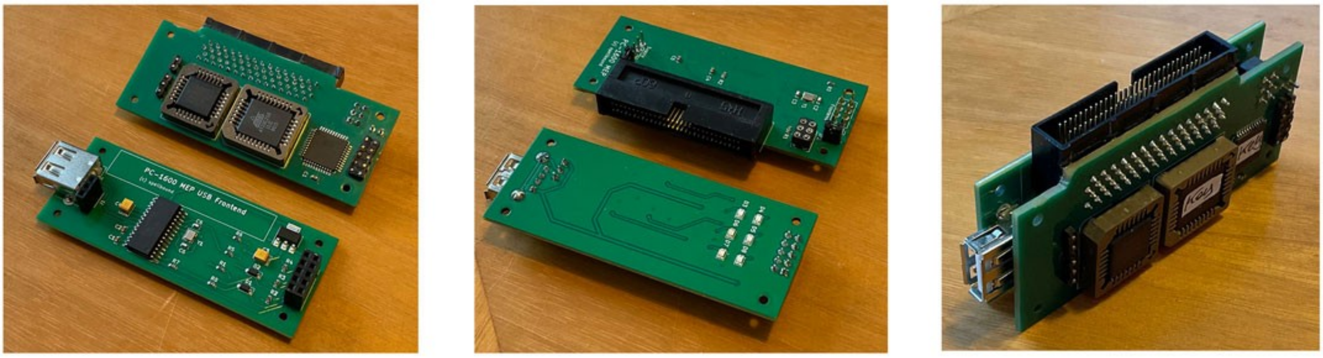
SHARP PC-1600
Modular Extension Platform (MEP) rev3
USB Memory Stick Reader/Writer



4-layer PCB backend, 2-layer PCB frontend, double-sided assembly



Industry grade 3D-printed shell, CNC-milled brushed aluminium cover

The module fully integrates with the PC-1600 OS and presents the memory stick as a device S3. So read and write access is as easy and natural as with a RAM-disk S2. Supported standard commands:
FILES,[B]SAVE,[B]LOAD,COPY,KILL,NAME,OPEN,PRINT#,INPUT#,CLOSE
Plus two extra commands from the modules ROM extension:
CDIR: Change directory on the memory stick (UNIX-style)
LDIR: List subdirectories of current directory

Table of Contents

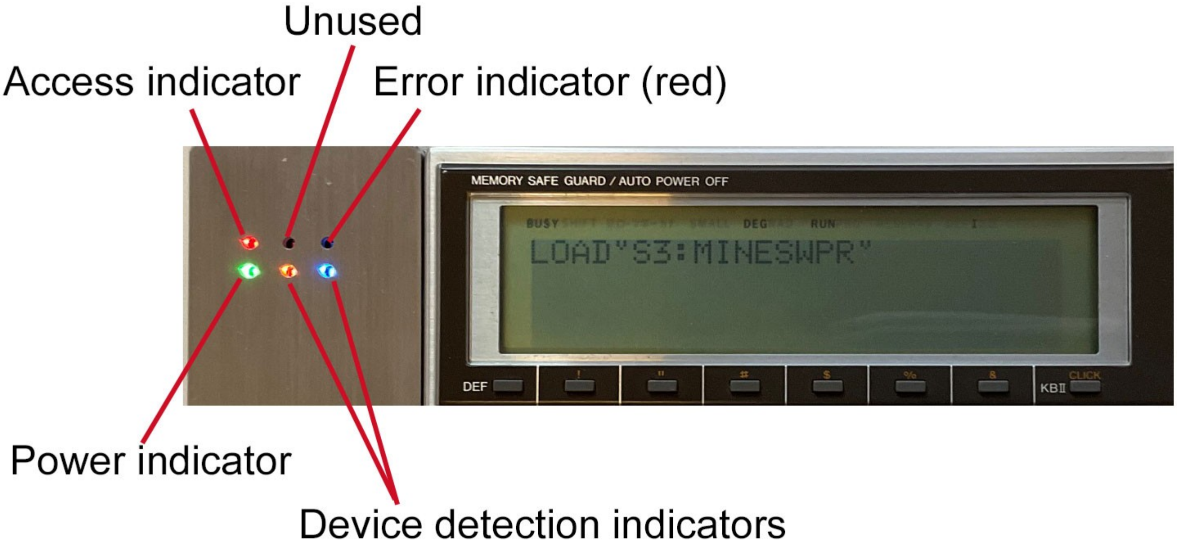
Generic MEP rev3 Hardware Features....3
LED Indicators - Memory Stick Application....3
Preparation....4
Standard BASIC File Commands....5
Directories – Additional BASIC Commands....6
Usage with DiskWorks....7
Assembler Programming - IOCS File API....8
Appendix - Modification for Extra On/Off-Switch....9
Copyright....10
Contact....10

Generic MEP rev3 Hardware Features

- Modular design with exchangeable frontend hardware. UART Rx/Tx communication between frontend and backend with up to 57600baud.
- Backend:
  - ATmega32A microcontroller with I/O-port communication (INP/OUT) to the PC-1600 with application code
  - 28C256 EEPROM with socket. Contains PC-1600 ROM extension
  - 22V10C PLD with socket. Provides addressing:
    ROM extension at #7,&4000, microcontroller port: &90
  - Onboard SPI socket for programming of the microcontroller
  - Compatible 60pin bus connector.
- USB-Frontend:
  - Hobbytronics USB-Host Controller with preloaded application
- Compatibility: **Only SHARP PC-1600**.

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

LED Indicators - Memory Stick Application



Preparation

In the following the terms USB-stick, memory stick and flash drive are used synonymously.

When the MEP module comes with the flash drive app installed, 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.

Assure your PC-1600 is equipped with fresh Batteries, or connected to an EA-160 or equivalent power adapter.

Connect/disconnect the module to/from the 60pin bus interface of the PC-1600 or CE-1600P only when the computer is switched off.

Supported devices: USB 2.0 memory sticks. Some USB 3.0 devices may also work (untested), but watch their power demand/battery drain.

Before using the module you need to format the flash drive with FAT32 (recommended) or exFAT. NTFS is not supported. Depending on the capacity of the flash drive you need to partition it first, since these file systems do not support large memory capacities. When formatting, choose "S3" as the new device name.

Hot-plugging (i.e. plug/unplug the memory stick while the module is powered) may work depending on your memory stick, but is not recommended. If the error LED is on or blinking after hot-plugging, you definitely need to turn the computer off and shouldn't use hot-plugging any further.
In the worst case the computer could crash or read or write errors may occur.
In that case you must perform a reset of the PC-1600.

The module has been tested with different USB 2.0 products. However the USB-frontend relies on certain timings, so if you use very old or very slow USB-sticks, you may experience read or write errors.

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, as long as no CE-1600F is connected:

- `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 this MEP app. 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 this MEP app 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.

This MEP app is restricted to the 8.3 (FAT) file format like the PC-1600. Avoid longer file and directory names and usage of special characters or spaces.

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 MEP flash drive app can navigate through that structure with the aid of the CDIR command.

LDIR

This additional command 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

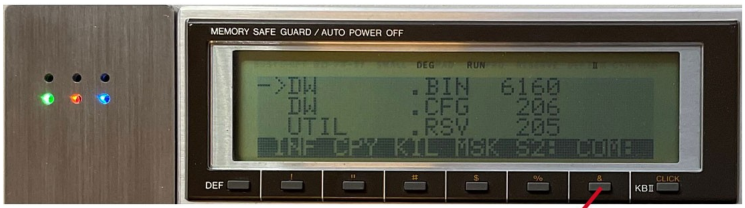
DiskWorks (DW.BIN) is the file browser software for the SHARP PC-1600.

You can download it e.g. from here: https://www.sharp-pc-1600.de/Down_Maschine.html or contact the MEP rev3 creator.

The MEP 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 and you don't have a CE-1600F connected (originally the device name Y is associated with the CE-1600F).

Here is a short usage introduction:

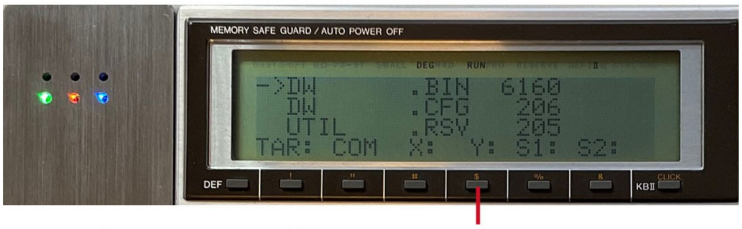
DiskWorks default screen showing content of S2



```
->DW        .BIN  6160
  DW        .CFG   206
  UTIL      .RSV   205
INF CPY KIL MSK S2: COM:
```

Press function key ‘&’ to select 2nd device

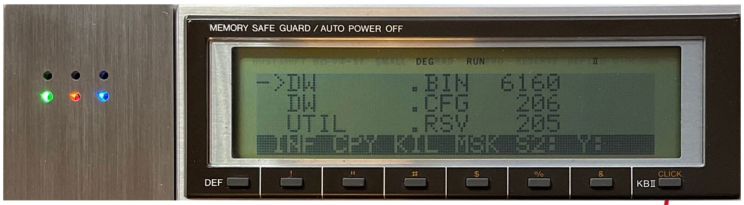
Device selection menu



```
->DW        .BIN  6160
  DW        .CFG   206
  UTIL      .RSV   205
TAR: COM  X:  Y: S1: S2:
```

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

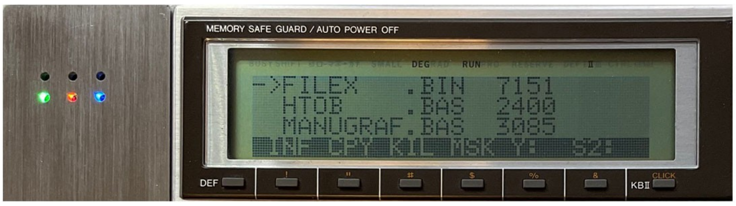
Swap primary and secondary device



```
->DW        .BIN  6160
  DW        .CFG   206
  UTIL      .RSV   205
INF CPY KIL MSK S2: Y:
```

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

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



```
->FILEX     .BIN  7151
  HTOB      .BAS  2400
  MANUGRAF.BAS  3085
INF CPY KIL MSK Y:  S2:
```

Assembler Programming - IOCS File API

The MEP flash drive app provides a ROM extension for the PC-1600. This 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 MEP app 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.

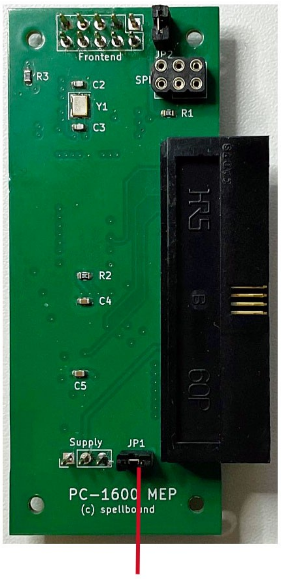
Appendix - Modification for Extra On/Off-Switch

To my knowledge some of the very first PC-1600 models have a permanent 4.5V power supply at the 60pin bus interface, even if the computer is switched off.

If you have such a very old PC-1600 model, then you probably want an extra ON/OFF-switch for the MEP rev3 module.

You can add that quite easily:

- Loosen the four M2-screws on the shells bottom side and remove the aluminium cover.
- Remove the four M2 O-rings and the screws.
- Carefully pull out the PCB-assembly from the shell.
- Separate the frontend PCB (containing the USB-jack) and the backend PCB (containing the 60pin plug).
- Remove the power supply jumper JP1 from the backend PCB (see image).
- Attach a micro ON/OFF-switch to an appropriate position on the shell, so that it does not interfer with the PCB-assembly. Check before you drill any holes into the shell.
- Connect the switch to the JP1 pins instead of the jumper.
- Re-assemble the module.



Power supply jumper

ATTENTION

Be aware that the modules ROM registers at boot time of the PC-1600 OS, so you must switch the modified module on before you turn on the computer and switch it off after you turn the computer off.

Never operate the module at the PC-1600 without the jumpers or when the modified module is switched off! LEDs will light up[1] or flicker in that case, but the module is not functional then.

[1] The IC's propagate high signal levels from the data or address bus to VCC, which in turn drives the power LED, although there is no stable power supply.

Copyright

Contact

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