
EPSON Backpack+ Drive+ Manual

EPSON PX-8 & HX-20 Version

(inc Universal Backpack Drive)



Credit: (c) Jeff Birt 2021

Version 1.3

12 July 2024

Document Changes	5
Release Notes.....	5
Introduction.....	6
Supported PX-8 Disk BASIC Features.....	7
Supported HX-20 Disk BASIC Features	7
Using This Manual.....	8
PART I: Quick Start Guide.....	9
Use with the Epson PX-8 Laptops	9
Use with the Epson HX-20 Laptops.....	11
Setting the Time and Date on the EBPD+.....	12
Updating EBPD+ drive Firmware using the SD card	13
Troubleshooting the EBPD+	14
PART II: Command-Line Interface (CLI).....	15
Using CoolTerm™ with the EPSON Backpack+ Drive from a Mac/PC	16
Command Summary.....	17
Alphabetical List of Commands with description.....	19
backup - Backup Command.....	19
disk - Setting update file	19
cat – List contents of file.....	20
cd - Change Directory.....	20
cdir – Directory listing for Virtual Floppy image (PX8).....	21
cp – Copy file.....	21
date – Set or print date	21
day – Set or print day of week	22
dir – Directory Listing	22
ext – Extract a file from a Virtual Floppy image (PX8)	22
factory – Factory reset Command.....	23
fmt – Format a Virtual Floppy image (PX8)	23
help – Help Command	23
impc - Import Conversion command routine (Z88).....	24
info – EPSON Backpack+ Drive Information	24
ins – Insert a file into a Virtual Floppy image (PX8)	24
ls - List Directory contents.....	25
mkdir – Make directory	25

mnt – Mount a Virtual Floppy image (PX8).....	25
mv - Move file	26
pwd – List present directory name.....	26
ren - Rename file.....	26
reboot – Reboot EPSON module	26
reset – reset Epson Backpack+ Drive	26
rm – Delete file	27
rmdir – Remove/Delete Directory	27
set – Set Command options	27
time – Set or print time	28
type – List the contents of the file.....	28
unmt – UN-MounT a Virtual Floppy image (PX8)	28
xmodr – receive a file using the XMODEM protocol ignoring CTRL-Z.....	29
xmodt – transmit a file using the XMODEM protocol.....	29
xmodz – receive a file using the XMODEM protocol.....	29
Part III: Understanding the Epson Backpack+ Drive	30
References and Background Reading	31
Theory of Operation	32
Hardware Operation.....	32
Firmware Operation	36
Annex A: Firmware Updates.....	37
Installing the Epson Backpack+ Drive Bootloader	37
Updating EPSON Backpack+ Drive Firmware using the SD card	37
Recovering a ‘bricked’ Epson Backpack+ Drive.	38
Annex B: SD card Format	39
Annex C: PX-8 Cable Configuration	40
Annex D: HX-20 Cable Configuration	41
Annex E: AVR port usage.....	42
EPSON Backpack+ Drive port usage.....	42
Annex F: EPSON Backpack+ Drive Part List	44
Annex G: PCBs	46
Adaptor PCB	46
Epson Backpack+ Drive PCB	46

DB25 PCB	47
Annex H: Open-source license reproductions.....	48
Bootloader for EPSON Backpack+ Drive.....	48
SD Card management firmware for the Backpack+ Drive.....	48

Document Changes

Date	Version	Changes
11 Aug 202	1.0	Original draft for new BPD+ motherboard
18 Aug 202	1.2	Updated xmod CLI commands
19 Dec 2023	1.3	Corrected several command formats
12 July 2024	1.4	Added in support for Universal BPD

Release Notes

- 11 Aug 2022 – First Version for new BPD+ motherboard

IMPORTANT: Parts of the Epson Backpack+ Drive (EBPD+) and Universal Backpack firmware use Open Source software that were/are available on Github or other web sites. The license agreements for the Open Source software are reproduced with links to those relevant parts in the Annex. As always, the web pages should be consulted for more up-to-date and definitive information.

1. Although not used in the EBPD+ or UBPD the optiboot_dx software did help with the AVR boot loader. https://github.com/SpenceKonde/DxCore/tree/master/megaavr/bootloaders/optiboot_dx
2. Avr_boot: https://github.com/zevero/avr_boot bootloader for the EBPD+.
3. The XMODEM software deployed on the EBPD+ uses some of the ideas in <https://github.com/Thuffir/xmodem>
4. FatFS: http://elm-chan.org/fsw_e.html SD Card file management software.

Welcome to the EPSON Backpack+ Drive User Manual

Introduction

The ability to store computer files on contemporary media such as Secure Digital (SD) cards has become an almost standard feature on modern computers. In general, this benefit is somewhat lacking for older computers such as the EPSON PX-8 and HX-20 series of laptops. There are, of course, a few modern solutions that provide support for the EPSON machines.

The manual also covers the use of the Universal Backpack Drive (UBPD). Some commands listed may not be available of the EBPD+.

The EBPD+ is a simple, easy-to-use storage device that supports a micro SD card and serial port it was designed to address this storage oversight for vintage computers. The EBPD+ can mimic, as closely as possible, the capabilities of both the original PF10 and TF20 Epson Drive units thereby allowing the laptop to read and write files from/to the SD card on board. The form factor of the EBPD+ also complements the HX-20 and PX-8 laptops. The EBPD+ will work with both the HX-20 and PX-8 using the Epson Serial protocol (EPSP) (NOTE: It may also work with the PX-4, being a close cousin of the PX-8, however this has not been tested).

The EBPD+ can support reading/writing of files to/from a virtual floppy disk image when used with the PX-8 or individual files on virtual drives when used with the HX-20. It also supports the Format command and copy disk command to allow the formatting and copying of virtual disk images in the drive when used with the PX-8. The HX-20 does not support the format or copy commands. In addition the Command Line Interface (CLI) provides commands that allow new virtual floppy disk images to be mounted, un-mounted and created for use with the PX-8. The virtual floppy disk images can also be mounted with write protection. To allow for the management of the virtual floppy disk images CLI commands are provided to extract and insert files from/to the virtual floppy disk images. This allows files to be more easily transferred to a PC/MAC. There is also a command to list the directory of a virtual floppy disk image to aid with the management of these files. As the HX-20 Disk BASIC deals with files rather than floppy disk images the regular CLI copy, rename, ls etc. commands can be used to manage the virtual drives of the HX-20.

To simplify using the EBPD+ with the PX-8 there is also an adapter board that accepts the EBPD+ and plugs into both the Serial and RS232 ports on the laptop using readily available 8pin mini DIN cables. Throwing a switch on the adapter board then connects the EBPD+ to either the Serial or RS232 port. The C:TERM program can then be used to access the EBPD+ when in RS232 mode for configuration operations.

The adapter board can also be used with the HX-20 however only the serial port is currently supported there is no support for a terminal program primarily because of the lack of availability of a suitable program on the HX-20.

Supported PX-8 Disk BASIC Features

Most of the features on the PX-8 are supported as they operate on a virtual disk image that is controlled by the operating system. However features that directly manipulate the media are not supported. Formatting of a disk is supported. Virtual floppy disk images can be mounted with a write protect option which will be reported when write actions are requested to that disk.

The PX-8 ‘copydisk’ program will load, however only the format option is supported. All other operations will cause a *BDOS DISK ERROR*.

Supported HX-20 Disk BASIC Features

The following table lists the supported HX-20 Disk BASIC features. Some features are not supported because they do not apply to the SD card features and others because it was not possible to test them with the files available for the HX-20.

Command	Comments
CLOSE	Supported.
CVD	Supported.
CVI	Supported.
CVS	Supported.
DSKF	Supported. Always reports max space on HX-20.
DSKI\$	Not Supported
DSKO\$	Not Supported
EOF	Supported.
FIELD	Supported.
FILES	Supported.
FILNUM	Supported.
FRMAT	Supported: performs no action on the drive.
GET	Supported.
INPUT\$	Supported.
KILL	Supported.
LINE INPUT	Supported.
LIST	Supported.
LOAD	Supported.
LOADM	Unknown: could not find test files.

LOC	Supported.
LOF	Supported.
LSET	Supported
MERGE	Supported.
MKD\$	Supported.
MKI\$	Supported.
MKS\$	Supported.
NAME	Supported.
OPEN	Supported.
PRINT\$	Supported.
PRINT # USING	Supported.
PUT	Supported.
RESET	Supported: performs no action on the drive.
RSET	Supported.
RUN	Supported.
SAVE	Supported.
SAVEM	Unknown: could not find test files.
SYSGEN	Supported: performs no action on the drive.
WHILE/WEND	Supported.

The HX-20 Epson ‘copy.utl’ program will load, however only the file copy option is supported. The disk copy operation will return the ‘write protected drive’ error.

The use of random files is limited to the size of a disk 278Kbytes (approx). This will limit a random file to about 2500 entries.

Using This Manual

The “Quick Start” section contains the basic instructions to set up and use your EBPD+ module.

The “Command Line Interface (CLI)” section describes the use of the "CLI" mode and explains how to issue commands to the EBPD+ Module. Users should, at minimum, mount a virtual floppy disk image and set the date and time so that file timestamps are correct when the SD card is used on a modern computer.

The “Theory of Operation” section describes the hardware and firmware operation in detail, including procedures for programming new firmware.

PART I: Quick Start Guide

This section provides a quick guide setting up and using the EBPD+ with the PX-8 or HX-20 laptop. Depending on the source of the EBPD+ and the configuration of your laptop, some of these steps might not be required. Loading of the firmware is only required for modules that you have constructed, complete units will come already loaded with firmware -- see Annex A.

Although a number of machines have been tested there may be other vintage machines that could use the XMODEM or upload features built into the BPD+ feel free the experiment. The firmware on the BPD+ can be changed to support the Tandy Model 100/120/200, NEC 8201, Cambridge Z88 and NTS Dreamwriter.

Use with the Epson PX-8 Laptops

The following steps outline how to access the CLI of EBPD+.

Step	Operation
1	Insert a CR1025 battery into the holder near the power switch.
2	Insert AA battery into holder. Alkaline batteries are preferred. Make sure the orientation is correct.
3	Set the DTR/DSR and CTS-RTS switch on the EBPD+ to open mode. (Switch set away from the dot on the case.)
4	Prepare a micro SD card with the boot files. The required files can be downloaded from the latest Release at https://github.com/Jeff-Birt/Backpack ; extract the “sector1” directory (and all its files) to the root of the SD card. The “sector1” directory should now contain the “help” directory. Two other directories should have been created: PF10 and TF20. The PF10 directory should be used for PX-8 virtual disk images. The TF20 directory is used by the HX-20.
5	Insert the micro SD card into the EBPD+. The card should be inserted with label-side toward the DB25 connector.
6	Ensure the EBPD+ is switched off, and plug it into the laptop RS232 port. If the adaptor board is being used switch to R. This will connect the EBPD+ to the RS232 port and set the baud rate to 19200. If no adaptor board is being used then plug the 8pin mini-DIN into the RS232 port of the PX-8 and the RS232 configured DB25 into the EBPD+. (Wiring diagram for this DB25 is shown in the annex.)
8	Open the TERM program on the PX-8 using C:TERM. NOTE: The C:CONFIG program should be used first to set up the serial interface for 19200bps, 8bits and 1 stop bit. The standard TERM options should work with the EBPD+. Now power on the EBPD+. The Green and Yellow LEDs should blink briefly. The screen should show ‘CLI’ followed by a prompt ‘#’. The EBPD+ is now in CLI mode.

-
- 9 The CLI can now be used to set the clock or mount/unmounts virtual floppy disk images. See the CLI command section for the mount and unmout commands.
-

The TERM program can now be exited. If using the adapter board the switch should be set to S or Serial. If using a cable the Serial port cable should now be connected to the EBPD+. The drive is now ready for use with the PX-8 drives or BASIC.

- 10 IMPORTANT NOTE: If individual cables are used for the Serial and RS-232 port they are wired differently. Any attempt to use the same or swapping cables will prevent the EBPD+ from working.
-

Use with the Epson HX-20 Laptops

The following steps outline how to access the CLI of the EBPD+. Setting the time/date for the HX-20 requires the use of a separate computer with an RS-232 interface. A suitable Terminal program for the HX-20 is not available.

Step	Operation
1	Insert a CR1025 battery into the holder near the power switch.
2	Insert AA battery into holder. Alkaline batteries are preferred. Make sure the orientation is correct.
3	Set the DTR/DSR and CTS-RTS switch on the EBPD+ to open mode. (Switch set away from the dot on the case.)
4	Prepare a micro SD card with the boot files. The required files can be downloaded from the latest Release at https://github.com/Jeff-Birt/Backpack ; extract the “sector1” directory (and all its files) to the root of the SD card. The “sector1” directory should now contain the “help” directory. Two other directories should have also been created PF10 and TF20. The TF20 directory is used by the HX-20 and should contain four subdirectories A, B, C and D to represent the 4 drives of the TF-20. The A subdirectory should contain BOOT80.SYS and DBASIC.SYS that are used to boot Disk Basic on the HX-20. The PF10 directory should be used for PX-8 virtual disk images.
5	Insert the micro SD card into the EBPD+. The card should be inserted with label-side toward the DB25 connector.
6	Ensure the EBPD+ is switched off, and plug it into the laptop Serial port using the serial cable (Wiring diagram for this DB25 is shown in the annex.). If the adapter board is being used switch to S. This will connect the EBPD+ to the serial port and set the baud rate to 38400. The HX-20 uses a 5pin DIN connector for the serial port.
7	The DIP switches inside the bottom cover of the HX-20 should have DIP 4 set to ON so the HX-20 searches for the disk drive when entering BASIC in order to boot Disk BASIC. Further details can be found in the HX-20 user guide.
	On the main menu of the HX-20 select BASIC (option 2) this should load Disk BASIC from the SD card. When the loading process has completed the BASIC prompt will return. The BASIC can now be used to access the EBPD+. For example typing FILES should list the files on the A: drive – the drive should contain at least BOOT80.SYS and DBASIC.SYS.

Setting the Time and Date on the EBPD+

The Time, Date and Day functions are controlled through the CLI mode of the EBPD+. To access the CLI mode use the steps outlined in the previous section. The time, date and day of week will need to be reset whenever the CR1025 RTC battery is removed or replaced.

Step	Operation
1	Access the CLI via the RS232 connection as described above.
2	Set the current date with the ‘date’ command. August 11, 2022 is given as an example date 11/08/2022<Enter>
3	The exact format will depend on how the date format has been configured. The factory default is dd/mm/yyyy.
4	Set the current time using the ‘time’ command. 4:20 PM is given as an example time 04:21:00 p<Enter>
5	Set the day of the week using the ‘day’ command. Friday is given as an example day fri<Enter>

Updating EBPD+ drive Firmware using the SD card

1. Copy the new firmware file to the root of the SD card.
2. Start the C:TERM program and access the CLI (See “Command Line Interface”). See steps 1-5 above.
3. Enter command: DISK U <filename><Enter> where <filename> is the name of the new firmware file.
4. Press Y to enable update
5. The boot status for sector “U” should show an asterisk next to the selected boot filename
6. Reboot the EBPD+: REBOOT<Enter>
The boot process should display the message “Normal Update” followed by “Processing File”.
7. Verify the new firmware version: INFO<Enter>

The green and yellow LEDs will also flash to provide a visual indication of the update progress.

Troubleshooting the EBPD+

Yellow LED stays on continuously

Check that a properly-formatted micro SD card has been inserted fully into the slot. The label side should face the serial connector.

PART II: Command-Line Interface (CLI)

The EBPD+ has two primary states: the CPM state and CLI state (see above for description). In the CPM state, the drive will service EPSON Serial Protocol Commands. To access the EBPD+ CLI from the laptop the C:TERM application can be used for the PX-8 (currently no suitable Terminal program has been found for the HX-20). Set the serial port to 19200bps, 8bits, no parity and 1 stop. See your laptop's manual for more information on the use of the C:TERM application. Once connected, the CLI state should be enabled immediately if connected to the RS232 port. This should show the text 'CLI' followed by the '#' prompt. You can now issue CLI commands to the EBPD+.

All the virtual floppy disk images should be placed in a directory called PF10 at the root level. This directory should also be used for files to be inserted into images and will be the location of files extracted from images. The floppy disk image files should use the file tail '.BPD'.

The HX-20 uses the TF-20 directory that contains four subdirectories: A, B, C and D. These four subdirectories are used to represent the maximum 4 drives that an HX-20 drive can support. Any files to be used or files created by the HX-20 will be stored in these directories. The A directory should also contain two files that should never be removed: BOOT80.SYS and DBASIC.SYS. The BOOT80.SYS file is the bootloader for the Disk BASIC contained in DBASIC.SYS.

If the SD card is not installed then the CLI commands associated with the file system on the SD card will not work. Check that the SD card is installed correctly if a CLI command returns an error.

If a command is successfully completed then generally the "OK" response will be generated. Other positive and negative responses are generated depending on the command and the particular CLI state.

If an unrecognized command is entered, the "Unknown Command" error will be generated. Check the spelling of the command to make sure the command is valid and that it is available in the version of firmware on the Drive. It is anticipated that new commands might be added in future and existing ones might be enhanced.

If an unexpected or invalid parameter is entered for a command, the "Bad Parameter" error will be generated. Check that the parameters are valid for the command and that they are formatted correctly.

All the filenames used in the commands should follow the 8.3 format for the EBPD+ system. The Drive does not support long filenames on the SD card. Although the HX-20 allow spaces in the filenames this feature is not supported by the EPSON BPD if a space is desired the '_' should be used. This is a restriction of the SD Card file management system.

Relevant commands can accept control keys to change their actions. The following control keys are recognized:

Ctrl-C : terminates any command in progress

Ctrl-S : resume listing

Ctrl-Q : pause listing -- make sure pause is ended as the module will not return to CPM state

Ctrl-Z : close open file

<space> : continue

<enter> : continue

The example commands shown below use the following nomenclature:

<...> : Required field(s)

[...] : Optional field(s)

| : alternate options

Using CoolTerm™ with the EPSON Backpack+ Drive from a Mac/PC

For convenience, the EBPD+ module CLI can also be accessed from a Mac or PC through an RS232 connection using any serial terminal program. This is required when configuring the unit for use with the HX-20 as it does not have a built in Terminal program. The program CoolTerm is tested and recommended. CoolTerm should be configured to access the correct *serial port* (e.g. USBSERIAL) associated with the serial cable plugged into the EBPD+. Configure the Serial Port as: baud rate: 19200, Data Bits: 8, Parity: None and Stop Bit 1. The DTR and RTS should be set ON. Most of the Terminal default modes will work however “*Enter Key Emulation*” should be set to CR only, and the “*Convert Non-Printable characters*” and “*Handle BS and DEL characters*” should be checked.

Other serial terminal programs can be used, but configuration details will vary and are not covered here.

Command Summary

Command	Parameters	Description
backup	<filename>	Backup current flash firmware
cat	<filename>	List out filename to screen
cat	<filename>	List out filename to screen
cd	<directory name> ..	Change directory
cp	<src filename> <dest filename>	Copy src to dest file
date	[d/m/y m/d/y y/m/d]	Print or set date
day	[mon tue wed thu fri sat sun]	Set or print day.
dir	[*.<filetail>] [-w -d -m]	Directory listing
disk	[U <filename>]	Show Virtual Floppy image in use for the PX8 and sets the update file
cdir	<virtual Floppy>	Directory listing Virtual Floppy image
ext	<filename> <virtual Floppy>	Extract filename from Virtual Floppy image
factory		Reset to factory settings
fmt	<virtual Floppy>	Format a Virtual Floppy image
help		Print list of commands
info		List information
ins	<filename> <virtual Floppy>	Insert filename into Virtual Floppy image
ls	[*.<filetail>] [-w -d -m]	Directory listing
mkdir	<directory name>	Make directory
mnt	<d e f g>:<virtual Floppy> [+]	Mount a Virtual Floppy image
mv	<src filename> <dest filename>	Copy src to dest file
pwd		Present directory name
ren	<src filename> <dest filename>	Rename src file to dest file
reboot		Reboot the module
reset		Reboot the module
rm	<filename> <*.*>	Delete filename
rmdir	<directory name>	Delete directory

Command	Parameters	Description
set	[time <24 ampm> date <mdy dmy ymd> lines <num 1-255> update <OFF ON> sleep <0 - 60>]	Set various configuration options on module.
time	[hh:mm:ss hh:mm:ss [a p]]	Print or set time
type	<filename>	List out filename
umnt	<drive letter>	Un-mount a drive letter
xmodr ¹	[-c] <dest filename>	XMODEM receive to dest filename
xmodt ¹	<src filename>	XMODEM transmit src filename

¹Note: xmodr and xmodt have only undergone limited testing. Before using these two commands you must verify they work in your application.

Alphabetical List of Commands with description

backup - Backup Command

```
backup <filename>
```

This command allows the user to backup the current firmware image stored in the flash memory of the Microchip AVR64DD32. The backup file generated can be used to restore the EBPD+ should something fail during an upgrade or to revert to an earlier version (refer to Annex A for further details). In order to keep the SD card system files in line with the file structures it is recommended that the <filename> should use the 8.3 format with the file type being set to .bin. However there are no restrictions on the filename that can be used. While the file save is in operation progress dots are printed on the screen. Once the image has been saved to the SD card it will be verified and once again progress dots are used to indicate the verification is taking place. If the verification does not find any problems then *Success!* is printed on the screen. Note: the number of dots shown during the verify is less than the save as the first area of the Flash memory is used by the bootloader and is not save or verified.

If the command is entered without a filename this will generate a Bad Parameter error.

Example:

```
# backup epsonav.bin
epsonav.bin
Saving Binary Image
=====
Saved
Verifying
=====
Success!
```

disk - Setting update file

```
disk [U]
```

This command allows the EBPD+ to display the mounted disks and set the file to be used to update the drive.

The U drive, stores the filename to be used when updating the drive firmware from SD card.

If the command is entered without any parameters then the current virtual floppy disk image settings and update file are listed to the screen.

When changing the setting for the “U” (update) virtual drive, the filename will be set to the selected file, it will also prompt with “Enable Update [y/N] ?” (The default is no). If “Y” is entered, an update will be scheduled the next time the board is power cycled or the reboot/reset command is used. The file list is again printed out and if the update has been enabled a * will be printed next to the update filename. See

Annex A for updating the firmware on the EBPD+. The update file needs to be located at the top/root directory of the SD card -- it should not be in a sub-directory. (NOTE: No check is made on the existence of the file in the root directory.)

WARNING: The file that is selected with the new firmware image for the update Sector should be a valid firmware file for the module otherwise the module will be corrupted and a forced update will have to be initiated to recover the EBPD+, if the bootloader has not been corrupted. If the bootloader has been corrupted then it will need to be reloaded using an Atmel-ICE and Microchip virtual studio.

cat – List contents of file

```
cat <filename>
```

This command lists the contents of a file to the screen. It is useful for reviewing file contents and making sure the data is correct. Ctrl-S and Ctrl-Q can be used to pause and resume the print out. Using Ctrl-C will terminate the listing. **NOTE:** The command must be allowed to complete, otherwise the EBPD+ will never return to the CPM state.

Example:

```
# cat boot
boot
10 CLS;"---INITIAL PROGRAM LOADER II---
20 ?"      WAIT A MINUTE!":CLOSE
30 IF PEEK(1)=171 THEN M=4 ELSE M=3
40 OPEN"COM:98N1DNN" FOR OUTPUT AS #1
50 ?#1,"FF";CHR$(M);
60 FOR I=1 TO 10:NEXT:CLOSE
70 RUN"COM:98N1ENN
#
```

cd - Change Directory

```
cd <directory name> | ..
```

This command is used change to another directory. The command must be followed by a valid directory name or list. The ‘..’ notation can be used to go up one level of directory. Directory names can be concatenated to move down the tree more quickly e.g. cd dir1/dir2 will move to dir2.

In the example below the first command will enter directory xxd. A successful command will show the directory entered. The second *cd* command moves up one directory to the top directory shown by the /. The third command moves down two levels to the trial directory. The subsequent command then moves back to the top directory.

The cd command is very similar in operation to that found on other operating systems however it does have some limitations when running on a small-embedded MCU.

Example:

```
# cd xxsd  
/XXSD  
OK  
# cd ..  
/  
OK  
# cd xxsd/trial  
/XXSD/TRIAL  
OK  
#  
# cd .../..  
/  
OK  
#
```

cdir – Directory listing for Virtual Floppy image (PX8)

```
cdir <virtual floppy image>
```

This command is used to list the directory of a virtual floppy disk image. The directory format is similar to that produced on the PX-8.

cp – Copy file

```
cp <src filename> <dest filename>
```

This command is used to copy the contents of the src filename to a dest filename. Progress dots are used to show the state of the copy. If the src filename does not exist then a “Bad Parameter” error will be generated. If the dest filename already exists, the option will be given to overwrite the file.

Example:

```
# cp testr.co test1.co  
testr.co -> test1.co  
.....  
OK  
#
```

date – Set or print date

```
date [d/m/y | m/d/y | y/m/d]
```

This command is used to print out the current date on the EBPD+ or set the date. If no parameters are provided the date will be printed out. If a date in the correct format follows then the date will be reset to the new value.

In the example below the date is set to March 15, 2021 (set to dmy format).

See also `day`, `time` and `set` commands.

Example Mega:

```
# date
15/03/2021
#
```

day – Set or print day of week

`day [mon | tue | wed | thu | fri | sat | sun]`

This command is used to either print out the current day of week or set the day of week. If no parameter is provided the day of week is printed. If a valid day is provided this will set the day to the new value.

The example below shows the commands used to display the current day, then change the day to ‘sat’ and then display the new day.

See also `date`, `time` and `set` commands.

Example:

```
# day
Fri
# day sat
OK
# day
Sat
#
```

dir – Directory Listing

`dir [*.<filetail>] [-w | -d | -m]`

See `ls` command.

ext – Extract a file from a Virtual Floppy image (PX8)

`ext <filename> <virtual floppy image>`

This command is used to extract a file from a virtual floppy image. If the file exists it will be extracted with the same name as the requested file and stored in the same directory as the virtual floppy disk image. If the <filename> or <virtual floppy image> is not found a ‘file not found’ error will be generated. The <virtual floppy image> should not contain a file tail.

factory – Factory reset Command

```
factory
```

This command is used to restore the EBPD+ to a factory condition; it will set all the parameters back to their default condition. Any parameters previously set will need to be re-entered.

fmt – Format a Virtual Floppy image (PX8)

```
fmt <virtual floppy image>
```

This command is to format a virtual floppy image. It will set the file to 0xE5 and delete any contents. Before formatting begins it will request ‘Format floppy [y/N]?’ confirmation.

The <virtual floppy image> should not contain a file tail.

USE WITH CAUTION ALL DATA WILL BE DELETED.

help – Help Command

```
help [cmd]
```

This command is used to provide brief help information on the CLI commands available on the EBPD+. If the help command is entered without a command name a complete list of the commands will be displayed. If the help command includes a command name then information on that command will be displayed. If the help files are not present on the drive in /sector0/help then a very brief list of commands will be displayed as shown below. Reference should be made to this manual for a more detailed explanation of each command.

The help files can be edited using a text editor app as they are text files so they can be tailored to your requirements if desired.

Example with no help files present:

```
# help
Commands (see manual for details)
backup boot bye cal cat cd cp
date day dir factory help info
ls mkdir mv pwd ren reboot rm rmdir set
time type mnt umnt fmt ins ext xmodr xmodt
#
#
```

impc - Import Conversion command routine (Z88)

impc <src filename> [dest filename]

This command will convert a regular file to a format suitable for sending to the Z88 imp-exp utility. The file is converted to an ESCaped file as per the Z88 User Guide.

info – EPSON Backpack+ Drive Information

info

This command is used to display information about the EBPD+. The version and build date will vary depending on the firmware loaded onto the drive. The card type should reflect the type of SD card inserted into the drive. The U field shows the use time of the Drive in D:H:M, this timer only runs when the device is running and not asleep. The Hardware (HW) version is used to indicate the version of hardware being used.

The information for the EBPD+ includes the board voltage; it should be close to 3.3V for the battery version. The Battery voltage can be used to gauge the status of the battery. A voltage below 1.4V will bring up a LOW warning.

NOTE: The firmware continually checks access to the SD Card to determine the capabilities of the battery; it may therefore appear ‘OK’ but actually ‘low’ in terms of available energy to run the Drive.

Example:

```
# info
AVR EPSON V1.10
Main = 3.29V Battery = 1.68V OK
Thu 11/08/2022 4:07:39P U 000:00:55
Card Type = SDHC & Size = 15.5GB
Built Aug 10 2022 15:38:34 HW V:01
Ser:42225361510046160154012200000000
```

ins – Insert a file into a Virtual Floppy image (PX8)

ins <filename> <virtual floppy image>

This command is used to insert a file into a virtual floppy image. If the file exists and the virtual floppy disk image has sufficient space the file will be inserted with the same name. The filename should conform to the CP/M 2.2 file-naming convention otherwise it may fail to be recognized when being used in the drive. If the <filename> or <virtual floppy image> is not found a ‘file not found’ error will be generated.

The <virtual floppy image> should not contain a file tail.

WARNING: To avoid problems with the SD card filename convention do not use spaces within filenames.

ls - List Directory contents

```
ls [*.<filetail>] [-w | -d]
```

This command provides a listing of the current directory. This allows the maintenance of the SD card without the need to resort to another machine. It can use three modifiers to change the format of the output. In addition it has very limited wild card capabilities using the * option.

The following modifiers are used by the ls command:

- w lists the valid file types across the screen in four columns. This option provides a very easy to view list of files similar to the PX4/8 screens.
- d lists the files with their creation dates. The files are time stamped using the time from the on board RTC.

The wildcard option * can be used with various file tails to list those file types e.g. ls *.bas will list all the files with the .BAS file tail in a particular directory. This option can be combined with the other modifiers.

mkdir – Make directory

```
mkdir <directory name>
```

This command creates a new directory. A duplicate directory name will cause an error.

Neither the PX-8 nor HX-20 can support directory structures however this command might be useful for SD card housekeeping purposes.

mnt – Mount a Virtual Floppy image (PX8)

```
mnt [<d|e|f|g>:<virtual floppy image> [+]]
```

This command is used to mount a virtual floppy image at the desired drive name (i.e. D, E, F or G). If the virtual floppy image exists it will be mounted. If the same image exists on another drive it will be un-

mounted from that drive before mounting on the new drive. A virtual image cannot exist on two drives as would be the case with a real PF10.

If the command includes a ‘+’ the image will be write protected any attempt to write to it by the PX-8 will cause a write protection fault. The `ins` command however will still work.

If the `<virtual floppy image>` is not found it will be generated as a blank disk image. This can be used to create new virtual floppy disk images. The image name should be unique. All images are stored in the PF10 directory on the SD card.

If the command is used without any parameters the currently mounted disks are shown.

The `<virtual floppy image>` should not contain a file tail.

mv - Move file

```
mv <src filename> <dest filename>
```

This command is used to rename a file from src filename to dest filename. If the dest filename already exists this will result in a Check filenames error and the command will not be completed. Similarly if the src filename does not exist then the Check filenames error will be generated.

The mv command can also take directory names to move files from one directory to another.

pwd – List present directory name

```
pwd
```

This command is used to indicate the present working directory.

ren - Rename file

```
ren <src filename> <dest filename>
```

This command is used to rename the src filename to dest filename. If the dest filename already exists this will generate a Check filenames error. If the src filename does not exist it will generate a Check filenames error.

reboot – Reboot EPSON module

```
reboot
```

This command is used to reboot the EBPD+. Generally this command is used to initiate an update procedure.

reset – reset Epson Backpack+ Drive

```
reset
```

This command is used to reboot the EBPD+. Generally this command is used to initiate an update procedure on the EBPD+.

rm – Delete file

```
rm <filename> | <*.*>
```

This command is used to delete a file from the current directory. The filename must exist in the directory otherwise the *No Files found* error would be generated.

The `rm` command can also be used with the wildcard `*.*` to remove all the files in a directory. This operation requires a yes response to the `Sure [y/N]?` query. Use with caution it will delete everything in a directory!

rmdir – Remove/Delete Directory

```
rmdir <directory name>
```

This command is used to remove a directory. The directory must exist and be empty for the command to complete. If the directory does not exist the error '*Invalid directory*' will be generated. If the directory is not empty then the error '*Directory not empty*' will be generated, in this case all the files need to be removed from the directory before the operation can be completed. If the directory name supplied is not a directory then the error '*Not a Directory*' will be generated.

set – Set Command options

```
set [time <24 | ampm> | Date <mdy | dmy | ymd> | Lines <num 1-255> |  
update <OFF | ON> | sleep <0 - 60>]  
  
mode <MT | WP2 | GEN | Z88 | EPSON| SHARP | SHARPx | HEX>] - ONLY UBPD
```

This command is used to set various configuration parameters on the EBPD+. If the command is entered without any parameters then the current settings are displayed. The various settings can be concatenated into one line if desired.

- `time`: sets the time format to either 24 hour or `ampm`.
- `date`: sets the date format to either mon/day/year `mdy` or day/month/year `dmy` or year/month/day `ymd`.
- `lines`: sets the number of lines to be displayed on the screen before pausing, the value can range from 1 to 255. The default value 7 allows the use of the full display on a Model 100/102. For use on a Model 200 it might be more useful to set the number of display lines higher.
- `update`: sets the update feature `on` or `off`. The update feature allows the updating of the firmware on the EBPD+ see Annex A for further details.

- sleep: sets the inactivity time before sleeping, the value can range from 0 to 60 mins. The 0 setting will turn off the sleep mode. The default value is 2 mins. If the EBPD+ sleeps then it might require a power cycle when used with some software. Generally it will wake when it receives a character on the serial interface.

See also date, time, boot and day commands.

Example:

```
# set date dmy
OK
# date
05/03/2021
# set date myd
Bad Parameter
# set date mdy
OK
# date
03/05/2021
#
```

time – Set or print time

time [hh:mm:ss | hh:mm:ss [a|p]]

This command is used to set the real time clock on the EBPD+. If the hours are >12 and <=23 then it will assume the 24-hour clock has been used and set the time accordingly. If the hours <= 12 it will then look for trailing a or p to determine am or pm, if neither are present it will assume am time.

The format of the printed time is determined by the set command.

See also set, date and day commands.

type – List the contents of the file

type <filename>

See cat command.

unmt – UN-MounT a Virtual Floppy image (PX8)

unmt <d|e|f|g>

This command is used to un-mount a virtual floppy image at the desired drive name (i.e. D, E, F or G). An empty drive is shown as <EMPTY>.

xmodr – receive a file using the XMODEM protocol ignoring CTRL-Z

```
xmodr [-c] <dest filename>
```

This command allows files to be sent to the BPD+ using the XMODEM protocol. The xmodr command can take the -c qualifier to use the CRC option. The default is to use the checksum method.

This method ignores any CTRL-Z (0x1A) characters that might be embedded in the file allowing the transfer of files that use the full ASCII range. It does however mean that the standard XMODEM padding at the end of a file will remain and not be removed. This might be considered a none standard XMODEM implementation but it does allow the transfer of Z88 applications.

xmodt – transmit a file using the XMODEM protocol

```
xmodt <src filename>
```

This command allows files to be sent from the Drive using the XMODEM protocol. xmodt can only use the checksum method.

xmodz – receive a file using the XMODEM protocol

```
xmodz [-c] <dest filename>
```

This command allows files to be sent to the BPD+ using the XMODEM protocol. The xmodz command can take the -c qualifier to use the CRC option. The default is to use the checksum method.

This method will recognize the CTRL-Z (0x1A) characters that might be embedded at the end of a file and remove them. It does however mean it cannot be used with files that contain embedded CTRL-Z (0x1A) characters. This is the more standard XMODEM receive protocol method.

Part III: Understanding the Epson Backpack+ Drive

This section of the document discusses the theory of operation EBPD+ to provide further information if you want to build your own EBPD+ or repurpose the module to support other devices. As the EBPD+ features are provided entirely in firmware they should be easily updatable to correct problems or add new options. Alternative firmware could be developed to support a number of different serial to SD card interfaces as required.

NOTE: The Universal BPD uses the same hardware structure as the EBPD+ and the following descriptions can be used to understand the new hardware platform.

The EBPD+ uses a micro SD card (4GB, 8GB or 16GB recommended) to support the file system and provide storage for the laptop data. The SD card is removable to allow easier transfer of data to/from other PC/MAC computers. When copying files to or from modern computers, care should be taken not to introduce extra carriage return or line feed characters that might impact the ability of the laptop to read the transferred files successfully.

The EBPD+ includes a battery-backed Real Time Clock (RTC) to timestamp the files created on the SD card by the laptop. Although the original PF10/TF-20 units did not include a timestamp feature, this addition allows easier maintenance of the SD card file system when accessed by a modern computer. The RTC date and time are set using the built-in CLI (described above) accessible via the serial interface. The ability to timestamp files could also be applied to file data if a logging application was to be written for the Drive.

The EBPD+ can be constructed with two power supply options; both options are exclusive and cannot be combined. The first option uses a 1.5V AA alkaline battery to supply the power. This power source should provide about around 20 hours of active use or about 500 hours in sleep mode with a good quality AA battery. The delay before sleep can be set via the CLI.

References and Background Reading

This section provides some useful background reading to understand the EPSON module as well as the genesis of the module. There are probably many other solutions and references that can be found to aid in understanding if required, as well as act as inspiration to build something of your own.

1. EPSON PX8 Users Manual.
2. EPSON PF10 Manual.
3. EPSON TECHNICAL MANUAL PX-B. Downloaded from www.Manualslib.com manuals search engine
4. EPSP Protocol: <https://fjkraan.home.xs4all.nl/comp/hx20/epsp.html>
5. TF-20 Dual disk manual. https://electrickery.nl/comp/tf20/doc/tf-20_manual.pdf
6. PXVFS documentation: <https://fjkraan.home.xs4all.nl/comp/px8/px8vfs/index.html>
7. Documentation on PX8: <https://fjkraan.home.xs4all.nl/comp/px8/>
8. PF10 Directory Structure:
<https://fjkraan.home.xs4all.nl/comp/px4/vfloppy/epsonCPMdirectory.html>
9. Overview of PX8 and HX-20: <https://mh-aerotools.de/hp/hx-20/>
10. Microchip: Microchip: AVR64DD32 Datasheet.
11. Maxim Integrated: DS1338 I2C RTC with 56-Byte NV RAM Datasheet. RTC clock.
12. Texas Instruments: TRS3243E 3- to 5.5-V multichannel 600kbps RS-232 line driver/receiver with +/-15-kV IEC-ESD protection datasheet. Serial port interface.
13. <https://www.duracell.com/wp-content/uploads/2020/02/MN15US11191.pdf> performance data for Duracell AA battery.

Theory of Operation

This section describes the hardware and firmware of the EBPD+ to assist with understanding the operational aspects of the device, especially if the unit is to be used in other applications beyond those outlined in this manual.

Hardware Operation

A block diagram of the EBPD+ is shown in Fig 1. The module is designed to be as simple and low cost as possible while still providing useful and versatile functionality. As mentioned above the unit uses the Microchip AVR64DD32 set to run at 8MHz.

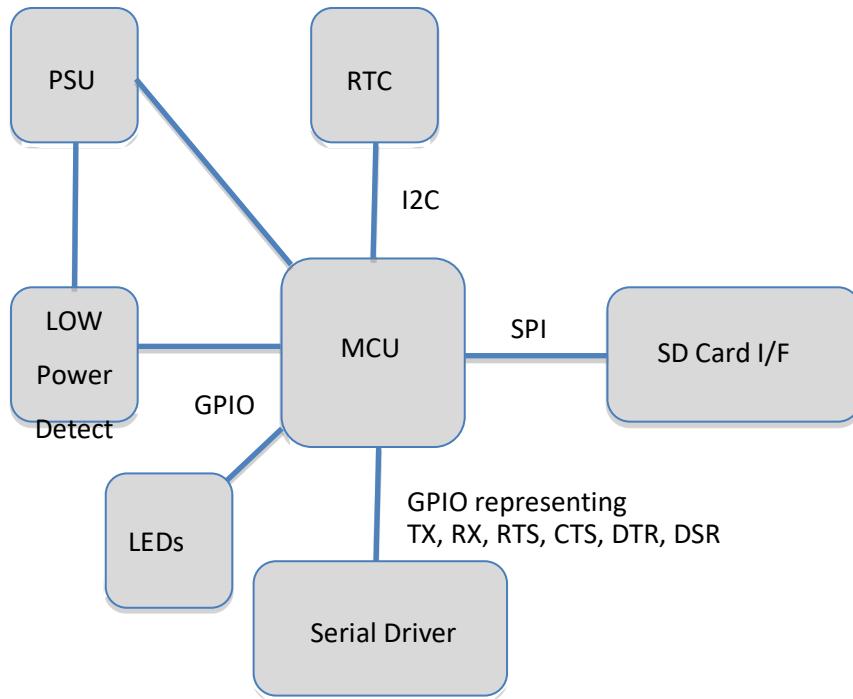


Figure 1: Epson Backpack+ Drive Block Diagram

The MCU interfaces to 4 separate sections: SD Card interface, Serial Driver interface, RTC and LEDs.

The EBPD+ can be powered either by a 1.5V AA battery or an external supply depending on the power ICs populated during construction.

The hardware interface to the micro SD card is via the SPI bus, which is also shared with the MCU programming pins. The SPI bus takes care of hardware aspects of reading and writing data to/from the SD card. The yellow LED is used to indicate the status of the SD card, although not exclusively. A short blink of the yellow LED indicates that the SD Card is being accessed. A solid yellow LED indicates a problem with the SD card, either it is missing, improperly inserted, badly formatted or corrupt. The SD card should

not be removed while the unit is powered on as this will potentially corrupt the SD Card or damage the card/processor. If the SD card is accidentally removed the MCU will reset and wait for it to be re-inserted. The unit has been tested with 4GB, 8GB and 16GB micro SD cards other larger cards could work but their behavior might be unreliable. In particular with larger SD cards the directory structure can become unwieldy and slow down the operation of the MCU.

The MCU interfaces to the RTC (DS1338) via the I2C bus to reduce the number of GPIO lines used. The RTC supports both time, date and day functionality although it only stores the lower two digits of the century. Currently the upper two digits are hardcoded into the MCU, unfortunately usage is limited to the 21st Century. The RTC is battery backed up using a 3V 30mAh CR1025, with an 800nA standby current this should give anywhere from 3-5 years of time retention depending on the quality of the battery and storage conditions of the unit. If the unit is to be stored for any length of time it is recommended that all batteries be removed to prevent leaks or corrosion of the board traces. If the RTC fails then an “RTC Failure” message will be printed on power up, the message could mean either a low voltage on the backup battery or an improperly fitted backup battery.

The serial interface uses the TX/RX pins of UART0 on the MCU and 4 GPIO pins to provide the control lines RTS, CTS, DTS and DSR lines. The control pins are there to allow compatibility between the laptop software and EBPD+ as well as future upgrades to support other machines. A Texas Instruments TRS3243E IC provides the serial voltages and interface. In sleep mode the serial interface is shutdown to save power. A rapidly blinking green LED indicates activity on the serial port, however the LED is also used to indicate other states (see below). The sleeping MCU is taken out of sleep mode when a character is received on the serial line.

There are two options to build the power supply. The first option uses a Torex XC9142D32DMR boost converter to provide 3.3V from a 1.5V AA battery. The second option uses a regular LDO to provide 3.3V from a wide input range of 3.6-6.0V. This allows flexibility when using the EBPD+ in different roles. There is an onboard low voltage detection (LVD) circuit that interrupts the MCU should the voltage fall below ~2.7V. This LVD allows the MCU to suspend operations when the voltage is too low to safely write to the micro SD Card. In a low voltage condition a ‘Low Batt?’ warning will be printed to the serial port. To increase battery life, the MCU can go to sleep after a set number of minutes of no activity. This feature can be controlled on the EBPD+. A sleep time from 3 – 10mins is recommended. When the MCU is in the sleep state the current drain decreases to <1mA. During normal operation the average current drain is ~30mA depending on the precise activity, this can increase to 140mA when the SD Card is being accessed. While the EBPD+ is working it also checks access to the sector2 directory every 15s to determine the quality of the battery. If the sector2 directory is missing or and the SD card is present it assumes the battery is low and will initiate a low power shutdown.

Two LEDs (Green and Yellow) are driven directly from GPIO pins on the MCU. The LEDs are used to visually signal various states of the EBPD+. Their primary use is to indicate activity on the serial port with

a blinking Green LED and activity on the micro SD Card with a blinking Yellow LED, as described above. On power up the Green and Yellow LEDs will blink alternatively to indicate successful power on. If the Yellow LED remains illuminated then a problem exists with the micro SD Card. The table below lists some of the states indicated by the LEDs.

Green LED	Yellow LED	Comment
blinking	X	Serial port activity
X	blinking	SD card Activity
off	steady	SD card problem
on on off	on off off	Entering sleep mode
on ~2s off ~2s	on ~2s off ~2s	Cycle repeats for 15s. Fatal error Possible low battery or SD card issue
During Update procedures		
on blink	on off	Update failed, file not found
on off	on blink	Update succeeded
rapid blink	off	Error occurred during update. Repeat.

The serial port daughter board provides the interface to the laptop. This board includes a small switch that allow the DSR & DTR and CTS & RTS lines to be looped together or connected straight through. This loopback is required to support some laptop software. They should be left in open mode when used with any PX-8 or HX-20.

The programming pin header can be used to reprogram the MCU on the EBPD+ using either the Microchip PIC 4 or Atmel ICE. Pin 2 is used by the bootloader to determine whether or not an update should be forced. When pin 1 and 2 are shorted using a shorting header no action will be taken on reboot or power cycle. If pin 1 and 2 are not shorted then an attempt will be made to update the firmware from the SD card (see Annex A). A shorting pin is required at all times to make sure it is available when needed. If the shorting header were not required for normal operation then finding one would be impossible when required.

Pin Number	Description	Comment
1	GND	DB-25 pin 7
2	Update/Recovery mode	N/C
3	UPDI	DB-25 pin 10
4	Vcc	DB-25 pin 9
5	NC	DB-25 pin 11

6	NC	DB-25 pin 12
7	RESET	DB-25 pin 13
8	GND	DB-25 pin 7

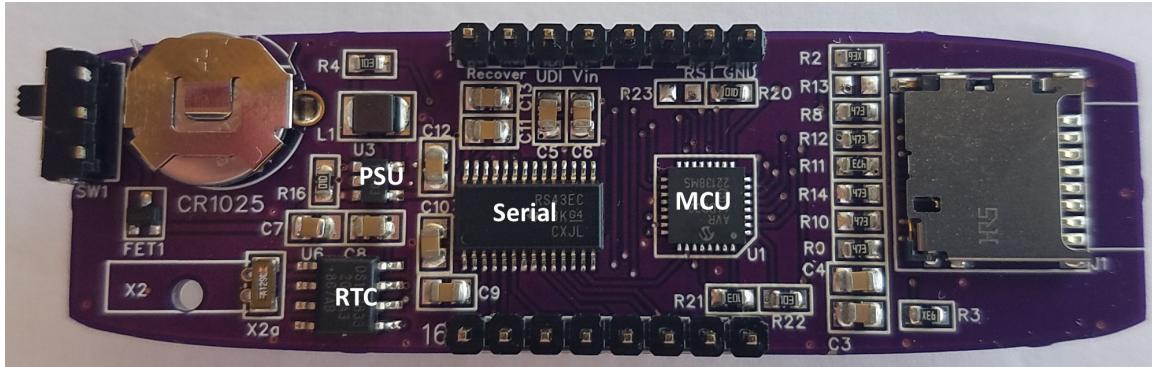
Table 1: Programming header pin descriptions

The following table lists the serial port pins available on the header from the laptop perspective. These pins use +/-5V and are not TTL level compatible. This information could be used to redesign the serial port daughter board to support other formats of serial interface.

Pin Number	Description	Comment
9	GND	DB-25 pin 7
10	DSR	DB-25 pin 6
11	CTS	DB-25 pin 5
12	Tx	DB-25 pin 3
13	DTR	DB-25 pin 20
14	RTS	DB-25 pin 4
15	Rx	DB-25 pin 2
16	Reserved	DB-25 pin 14

Table 1: Serial port header pin descriptions from laptop

Figures 2 and 3 show the completed boards for reference. Note the board color and parts might vary depending on supplier.

**Figure 2: Annotated EPSON Backpack+ Drive without Serial Port**

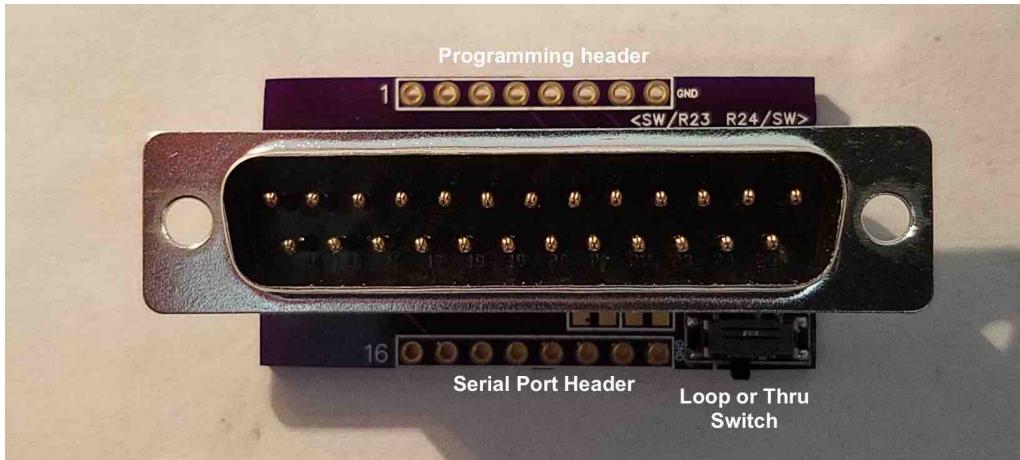


Figure 3: Annotated Serial Port Connector (V1.1)

Firmware Operation

The firmware can operate in three separate states: CPM state, CLI state and Sleep state. The CPM state is the default state of operation that is used to provide the disk services to the laptop. While in the CPM state, the serial port is continuously monitored for CPM Commands (e.g. from PIP).

If the EBPD+ is powered on with the RTS and DTR line set ‘high’ then the module will enter the CLI mode. To use CLI mode the EBPD+ must be moved to the RS232 port and the DTS/DSR and CTS/RTS lines looped back together. The firmware will then enter CLI mode.

The Sleep state is entered when no activity has been detected on the serial port for a set length of time. On the EBPD+ the inactivity time before sleep can be set via the CLI. There is also an option to turn off the sleep mode if required. As described above in the hardware section in the Sleep state the processor, serial interface and SD Card are all put into a low power mode to conserve the battery. The Sleep state is exited either when a character is received on the serial interface or the unit is power cycled/reset.

Annex A: Firmware Updates

The EBPD+ employs a bootloader to permit field upgrades without the need to use a device programmer - once the bootloader has been installed. The EBPD+ is updated using the SD card. Suitable .bin files for the bootloader can be found on the support page for the EBPD+.

Installing the Epson Backpack+ Drive Bootloader

The bootloader is based on the zevero/avrboot ([github: https://github.com/zevero/avr_boot](https://github.com/zevero/avr_boot)) software it uses a stand-alone MMC boot loader by ChaN and can be used on any device that has about 4Kbytes of boot flash memory. Installing the bootloader will require an AVR programmer such as the Microchip PIC 4 or Atmel-ICE and the Microchip visual studio to load the .bin file. The bootsize should be confirmed to be 0x0B. Once programmed the bootloader will search for the default file BPDP3000.BIN on the SD card. If the file is found this will be programmed into the flash memory and the EBPD+ should be operational. If the file/SD card is not found then the bootloader will continue to loop until an SD card with the file is found. The update file BPDP3000.BIN should be located in the root directory of the SD card.

The shorting plug should be removed when programming the bootloader, see the hardware description above for the location of the pin.

Updating EPSON Backpack+ Drive Firmware using the SD card

1. Copy the new firmware file to the root of the SD card.
2. Start the TELCOM program and access the CLI (See “Command Line Interface”)
3. Enter command: DISK U <filename><Enter>
4. Press Y to enable update
5. The boot status for sector “U” should show an asterisk next to the selected update filename
6. Reboot the EBPD+: REBOOT<Enter>
The boot process should display the message “Normal Update” followed by “Processing File”.
7. Verify the new firmware version: INFO<Enter>

The programming pins can remain shorted during this operation. Once the BIN file has been loaded into memory the EBPD+ should reboot with the new firmware. The green and yellow LEDs will flash to provide a visual indication of the update progress. There might be garbage printed to the screen before the CLI and # prompt appear.

Example using reboot command: (The version shown might be different)

```
reboot
AVR BPD+ BOOT V1.1
Normal Update
Processing File
Update ran
```

Recovering a ‘bricked’ Epson Backpack+ Drive.

If after an update or due to other reasons the Mega EBPD+ stops responding to CPM or CLI commands then it is possible to perform a hard recovery provided the bootloader has not been corrupted. If the bootloader has been corrupted then it will be necessary to start the bootloader installation operation assuming a blank MCU as described earlier.

To perform a hard recovery first power down the EBPD+ and remove the shorting header from the programming pins. The SD card shall be inserted with a copy of a known good firmware image and titled BPDP3000.BIN – this is the default image, at the root level of the directories. The unit can then be powered up to reprogram the firmware. If the serial port is connected the text below should be visible, if it is not displayed on power up then it is very likely the bootloader is corrupt. A successful read of the SD card should result in a recovered Mega EBPD+.

```
AVR BPD+ BOOT V1.1
Recovery mode using: BPDP3000.BIN
Processing File
Update ran
```

Annex B: SD card Format

Outline structure of SD card for use with the EBPD+s. The SD card should be no larger than 16GB as larger sizes may cause file-handling issues.

/ <update files>

CPMDRIVE.BIN

etc etc

/Sector2

/help

<help files>

MANCLI.DO

etc etc.

/pf10

virtual floppy images <- Images should end with .BPD

etc etc.

/tf20

/A

BOOT80.SYS <- File must be in A drive Directory

DBASIC.SYS <- File must be in A drive directory

/B

/C

/D

Annex C: PX-8 Cable Configuration

The EPSON PX-8 uses an 8pin Mini DIN connector for both the serial and RS232 ports. If the adapter board is not used then it will be necessary to make two cables one for use with the RS232 port and one for use with the Serial port.

RS232 Port Pin out

8pin Mini DIN Male	BPD DB-25 Female	Comment
1	7	GND
2	2	TXD
3	3	RXD
4	LOOP1	RTS
5	LOOP1	CTS
6	LOOP2	DSR
7	LOOP2	DTR
8		CD

Pins 4,5, 6 and 7 of the 8pin mini DIN are not connected. The RTS/CTS and DSR/DTR need to be looped back when connected to the BPD.

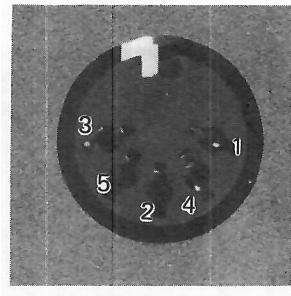
Serial Port Pin out

8pin Mini DIN Male	BPD DB-25 Female	Comment
1	7	GND
2	2	PTX
3	3	PRX
4		
5		
6		PIN
7		POUT
8		

The RTS/CTS and DSR/DTR need to be open when connected to the BPD

Annex D: HX-20 Cable Configuration

The EPSON HX-20 uses a 5pin DIN connector for the serial port. If the adapter board is to be used then it will be necessary to make a 5pin DIN to 8pin mini DIN to cable. Otherwise a 5pin DIN to 25pin Female DB-25 will be required. The cable pin outs are given below. Only 3 pins are required for the serial port to operate: RX, TX and GND.



1 = GND, 2 = TXD, 3 = RXD

5pin DIN Serial port on HX-20.

This would be the view from the rear of a 5pin DIN male connector – solder cup location

Serial Port Pin out 8pin Mini DIN to 5pin DIN

5pin DIN Male	8pin Mini DIN Male	Comment
1	1	GND
2	2	PTX
3	3	PRX
4		
5		

Serial Port Pin out DB-25 to 5pin DIN

5pin DIN Male	BPD DB-25 Female	Comment
1	7	GND
2	2	PTX
3	3	PRX
4		
5		

Annex E: AVR port usage

The following tables list the ports used by the EBPD+ the information is provided to allow the reuse of the Backpack Drive in other applications.

EPSON Backpack+ Drive port usage

- * PA0 = Unused
- * PA1 = Unused
- * PA2 = SDA
- * PA3 = SCL
- * PA4 = MOSI - out
- * PA5 = MISO - in
- * PA6 = SCK - out
- * PA7 = SS - out
- * DDRA = 11011100 = 0xDC
- * PORTA = 11011100 = 0xDC

- * PC0 = INT_UART_RX Interrupt from UART
- * PC1 = RS232_SHDN RS232 Shutdown pin - OUT
- * PC2 = RS232_EN RS232 Enable pin - OUT
- * PC3 = SDDET - IN
- * PC4 = unused
- * PC5 = unused
- * PC6 = unused
- * PC7 = Unused
- * DDRC = 00000110 = 0x06
- * PORTC = 00000010 = 0x02

- * PD0 = unused
- * PD1 = LED2 - OUT
- * PD2 = LED1 - OUT
- * PD3 = RTC_INT RTC Interrupt
- * PD4 = UART_TX - OUT
- * PD5 = UART_RX - IN
- * PD6 = SEL2 - IN
- * PD7 = VBATT - IN
- * DDRD = 00010110 = 0x16
- * PORTD = 00010110 = 0x16

```
* PF0 = unused
* PF1 = unused
* PF2 = DTR - IN
* PF3 = RTS - IN
* PF4 = DSR - OUT
* PF5 = CTS - OUT
* PF6 = RESET- - IN
* PF7 = UDI      - IN
* DDRF = 00110000 = 0x30
* PORTF = 00110000 = 0x30
```

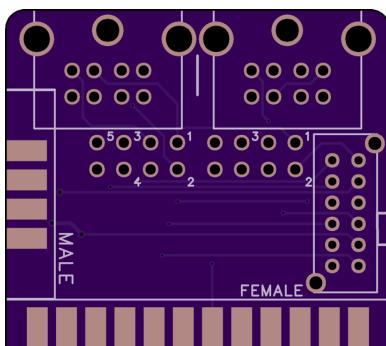
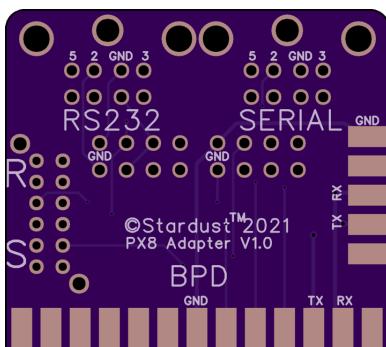
Annex F: EPSON Backpack+ Drive Part List

Manufacturer Part Number	Digi-Key Part Number	Ref	Qty	Description
SPC02SYAN	S9001-ND		1	CONN JUMPER SHORTING GOLD FLASH
PH1-08-UA	2057-PH1-08-UA-ND		2	CONN HEADER VERT 8POS 2.54MM
1024TR	36-1024CT-ND	B1	1	BATTERY HOLDER AA SMD TAB
3031	36-3031-ND	B2	1	BATTERY RETAINER COIN PC PIN
CL21B104KBCNFNC	1276-2444-1-ND	C4, C5, C11, C12	4	CAP CER 0.1UF 50V X7R 0805
CL21B105KOFNNNG	1276-6471-1-ND	C3, C6	2	CAP CER 1UF 16V X7R 0805
GRM21BR71A106KA73K	490-14381-1-ND	C7,C8	2	CAP CER 10UF 10V X7R 0805
CL21B474KOFNNNG	1276-6483-1-ND	C9, C10, C13	3	CAP CER 0.47UF 16V X7R 0805
APA3010YC-GX	754-1576-1-ND	D1	1	LED YELLOW CLEAR SMD R/A
APA3010SGC-GX	754-1586-1-ND	D2	1	LED GREEN CLEAR SMD R/A
DMP2240UW-7	DMP2240UWDICT-ND	FET1	1	MOSFET P-CH 20V 1.5A SOT323
DM3D-SF	HR1941CT-ND	J1	1	CONN MICRO SD CARD PUSH-PULL
DB25-PT-1	2057-DB25-PT-1-ND	K1	1	CONN D-SUB PLUG 25POS VERT SLDR

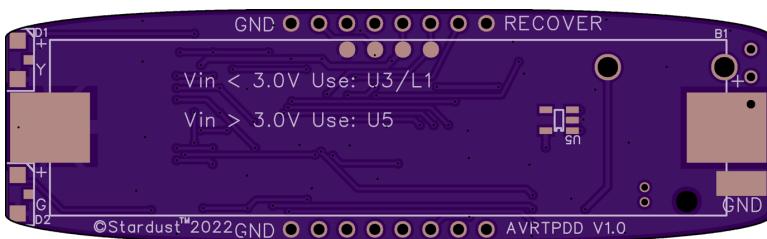
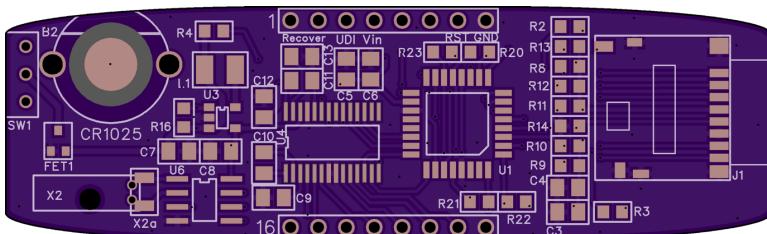
Manufacturer Part Number	Digi-Key Part Number	Ref	Qty	Description
LQH32CN4R7M53L	490-4057-1-ND	L1	1	FIXED IND 4.7UH 650MA 195 MOHM
RC0603FR-0790R9L	311-90.9HRCT-ND	R2, R3	2	RES SMD 90.9 OHM 1% 1/10W 0603
RC0603JR-0710KL	311-10KGRCT-ND	R4, R21, R22	3	RES SMD 10K OHM 5% 1/10W 0603
RC0603FR-07100KL	311-100KHRCT-ND	R16, R20	2	RES SMD 100K OHM 1% 1/10W 0603
RC0603JR-0747KL	311-47KGRCT-ND	R8, R9, R10, R11, R12, R14	6	RES SMD 47K OHM 5% 1/10W 0603
MHSS1105	679-1849-ND	SW1	1	SWITCH SLIDE SPDT 300MA 6V
EG1257	EG2586CT-ND	SW2	2	SWITCH SLIDE SPDT 300MA 4V
AVR64DD32-I/PT	150-AVR64DD32-I/PT-ND	U1	1	64KB, 8KB RAM, 32P, 24MHZ, MVIO
XC9142B33DMR-G	865-XC9142B33DMR-G	U3	1	IC REG BOOST ADJ 350MA SOT23-6
TRS3243ECPW	TRS3243ECPW-ND	U4	1	IC TRANSCEIVER FULL 3/5 28TSSOP
DS1338Z-33T&R	DS1338Z-33+CT-ND	U6	1	IC RTC CLK/CALENDAR I2C 8-SOIC
AB38T-32.768KHZ	535-9034-ND	X2	1	CRYSTAL 32.7680KHZ 12.5PF TH
VMK3-9001-32K7680000	150-VMK3-9001- 32K7680000TR-ND	X2a Alternate	1	CRYSTAL 32.7680KHZ 12.5PF TH

Annex G: PCBs

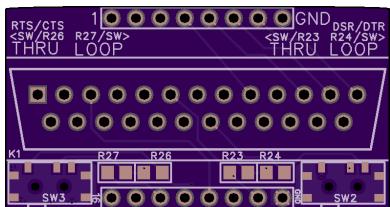
Adaptor PCB



Epson Backpack+ Drive PCB



DB25 PCB



Annex H: Open-source license reproductions

Licenses for the various open source software used in the project are reproduced below. Reference should be made to the relevant Github pages for up to date copies.

Bootloader for EPSON Backpack+ Drive

Github: https://github.com/zevero/avr_boot

License from Github:

Copyright (c) 2015, zevero All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SD Card management firmware for the Backpack+ Drive

url: <http://elm-chan.org/fsw/ff/doc/appnote.html#license>

General license statement

```
/*
 / FatFs - Generic FAT Filesystem Module Rx.xx
 /-----
 // Copyright (C) 20xx, ChaN, all right reserved.
 // FatFs module is an open source software. Redistribution and use of FatFs in
 // source and binary forms, with or without modification, are permitted
 // provided that the following condition is met:
 // 1. Redistributions of source code must retain the above copyright notice,
 //    this condition and the following disclaimer.
 // This software is provided by the copyright holder and contributors "AS IS"
 // and any warranties related to this software are DISCLAIMED.
 // The copyright owner or contributors be NOT LIABLE for any damages caused
 // by use of this software.
 /-----*/
```