Backpack Drive Manual

Version 1.4

17 May 2021

Document Changes

Date	Version	Changes
7 Mar 2021	1.0	Original draft
15 Mar 2021	1.1	- Mega-Backpack Drive port use - Annex on Boot loading options - Updated CLI to new software versions - Incorporated some review feedback
27 Mar 2021	1.2	- Corrected command descriptions - Tandy WP-2 Support options included
27 Apr 2021	1.3	- Corrected command descriptions - Reformatted CLI section - Included review feedback
17 May 2021	1.4	- Added step by step guides for Tandy WP2 - Added Tandy WP-2 Cable Requirements

Release Notes

- 14 Apr 2021 – First Version

IMPORTANT: Parts of the Backpack Drive 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. Optiboot_flash: https://github.com/MCUdude/optiboot_flash derived from https://github.com/Optiboot/optiboot bootloader for Mini-BACKPACK DRIVE.
2. Avr_boot: https://github.com/zevero/avr_boot bootloader for Mega-BACKPACK DRIVE.
3. FatFS: http://elm-chan.org/fsw_e.html SD-Card file management software.

Welcome to the Backpack Drive User Manual

Introduction

The ability to store computer files on contemporary media such as SD cards has become an almost standard feature on modern computers. In general this benefit is somewhat lacking for older machines such as the Tandy™ Model 100, 102 and 200 series of laptops. There are, of course, a multitude of excellent modern solutions readily available for these vintage laptops, including smartphone storage and DOS/Linux-based software, but they are not always convenient. The Backpack Drive, a simple embedded processor unit supporting a micro SD card and serial port, was designed to address this storage oversight. The Backpack Drive mimics, as closely as possible, the capabilities of the original Tandy Portable Disk Drive 2 (TPDD2) thereby allowing the laptop to read and write files from/to the on-board SD card. The form factor of the Backpack Drive also complements the Tandy and NEC 8201a/8300 series of laptops. The Backpack Drive will work with the Tandy WP-2 Word Processor (WP-2) as it uses the same TPDD2 interface, although there are some limitations on the features supported. All the file management applications listed in the Annex have been tested on these machines and work for those parts that do not rely on the behavior of magnetic media. For example, features such as bank (disk side) switching, formatting and sector access are not supported.

Although the Drive uses on-board firmware to implement the TPDD 2 ‘operation mode’ (a.k.a. M1 mode) which allows laptop to take advantage of SD card. The TPDD2 module does not currently support ‘FDC emulation mode’ (a.k.a. M0 mode) which would allow track or sector-based access. This mode is used by a number of other machines including most notably the Brother™ line of knitting machines, Pattern Programming devices e.g. PPD-100 and the FB-100 floppy disk drive used to store patterns. This mode will be added in a future release when suitable test hardware can be acquired.

Using This Manual

The “Quick Start” section contains the basic instructions to set up and use your TPDD 2 module. It is suitable for use with Tandy 100/102/200 and NEC machines.

The “Command Line Interface” section describes the use of the "CLI" mode and explains how to issue commands to the TPDD Module. Users should, at minimum, 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 Backpack Drive with the Tandy and NEC laptops. Depending on the source of the Backpack Drive 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.

Use with the Tandy and NEC laptops

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	For use with Tandy 100/102/200 or NEC laptops, set the DTR/DSR and CTS-RTS switch on the Backpack Drive to loop-back mode. (Switch set toward 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 "sector0" directory (and all its files) to the root of the SD card. The "sector0" directory should now contain the following files: BOOT, TS-DOS.100, TS-DOS.200, TS-DOS.NEC as well as the "help" directory.
5	Insert the micro SD card into the Backpack Drive. The card should be inserted with label-side toward the DB25 connector.
6	Ensure the Backpack Drive is switched off, and plug it into the laptop serial port.
7	If your laptop has TS-DOS in ROM (ROM or REX based), simply install TS-DOS and start using the Backpack Drive. No further steps are needed. (See the TS-DOS manual for instructions on installing TS-DOS from ROM.) If your laptop does not have TS-DOS in ROM continue following these instructions to load a RAM resident version of TS-DOS. Enter BASIC and type: `run "com:98n1enn"<Enter>` (for Tandy) `run "com:9n81xn"<Enter>` (for NEC)
8	Power on the Backpack Drive. The Green and Yellow LEDs should blink briefly, indicating the start of the boot load process.
9	After a few seconds the text "INITIAL PROGRAM LOADER II" should appear on your laptop screen. The yellow LED will blink occasionally during the load process until "Installing TS-DOS" is displayed and dots begin to appear on the screen. *NOTE: Depending on the boot file used the text may differ slightly.*

10	Wait approximately 3 minutes for TS-DOS to load and follow the instructions on your laptop screen to complete the installation of TS-DOS. *Note: TS-DOS is resident in your laptop's RAM and will be erased by a cold-start or power loss. In such cases, repeat this process from step 6 to restore TS-DOS to your laptop.*
11	Optional: Set the real-time clock (See "Setting the Time and Date")

Setting the Time and Date

The Time, Date and Day functions are controlled through the CLI mode of the Backpack Drive. To access the CLI mode, a terminal program such as the TELECOM application can be used. The time, date and day of week will need to be reset whenever the CR1025 RTC battery is removed or replaced.

Step	Operation
1	Return to the system menu Tandy: Press F8 NEC: Press Shift+F5
2	Start the TELCOM application
3	Set the serial port for 19200bps, 8bit, no parity, 1 stop bit Tandy: `STAT 98n1dnn <Enter>` NEC: `STAT 9n81xn <Enter>`
4	Enter terminal mode `TERM <Enter>`
5	Press `<Enter>` four more times until you see the '`C`' and a '`#`' prompt on your screen. You are now in CLI mode.
6	Set the current date with the '`date`' command. April 23, 2021 is given as an example `date 23/04/2021<Enter>` The exact format will depend on how the date format has been configured. The factory default is dd/mm/yyyy.
7	Set the current time using the '`time`' command. 4:20 PM is given as an example `time 04:21:00 p<Enter>`
8	Set the day of the week using the '`day`' command. Friday is given as an example `day fri<Enter>`

Updating Mega-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"). See steps 1-5 above.

3. Enter command: `BOOT 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 Backpack Drive: `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.

Use with the Tandy Word Processor 2 (WP-2)

The Tandy WP-2 machine is restricted to `.DO` text files and `.DA` dictionary files. The machine also requries 8.2 format filenames. In order to support these WP-2 specific features the Backpack drive needs to be configured for WP-2 support. The configuration can be undertaken on a Tandy or NEC laptop using the TELECOM app or using a regular serial port on a computer with a suitable terminal program such as CoolTerm as described in Command Line Interface section.

Using the FILES app on the WP-2

Step	Operation
1	Return to the system menu Tandy: Press F8 NEC: Press Shift+F5
2	Start the TELCOM application
3	Set the serial port for 19200bps, 8bit, no parity, 1 stop bit Tandy: `STAT 98n1dnn <Enter>` NEC: `STAT 9n81xn <Enter>`
4	Enter terminal mode `TERM <Enter>`
5	Press `<Enter>` four more times until you see the '`C`' and a '`#`' prompt on your screen. You are now in CLI mode.

6	Use the set command to configure for use with the WP-2.
	`set mode wp2`
	This will configure the drive to use 8.2 file formats and disable the boot options.
	NOTE: This step must be completed before attempting to use the Backpack Drive on the WP-2.
	To set the drive back for use with the Tandy or NEC laptops use:
	`set mode mt`

7	Plug the Backpack Drive into the WP-2 using a null modem adapter than has a 9-pin female to 25-pin female convertor. The pin connections are shown in the Annex. The BELKIN Pro Series DB9 Female/DB25 Female Adapter F2L089 seems to work well in this application.
8	On the WP-2 you can now use F2-Files option to access the file menu. Select the `diskette` option to bring up a list of available files from the SD Card. Files can then be copied to/from the WP-2 RAM Disk or Memory.

The following operations have been tested with the WP-2.

- File copy to/from the Backpack Drive.
- File rename and merge file on the Backpack Drive.
- File delete on the Backpack Drive.

Other options have not been tested but might work.

Using the TELECOM app on the WP-2

To use the TELECOM app on the WP-2 with the Backpack Drive the serial port must be configured as follows:

```
Device: RS232
Baud: 9600
Word: 8
Parity: None
Stop: 1
XON/OXOFF: Enable
Printer Echo: Off
Duplex: Full
```

The TELCOM app can be entered using F2-9 [TELECOM] this should bring up "`RS232C is Ready`" on the LCD screen. The ‘u’ key should be pressed this will then show “`Set to 9600bps`”. Now press `<Enter>` four times until you see the ‘`C`’ and a ‘`#`’ prompt on your screen. You are now in CLI mode. The mode will time out after 10s of inactivity.

CP/M-100 and REXCPM

The backpack drive can be used with REXCPM and CP/M-100 by changing two settings.

To bootstrap the REXCPM:

1. Copy the "RXCINI.DO" file into the "sector0" directory on the SD-Card. Copy the remaining REXCPM files to the root of the SD card.
2. Set the bootstrap file for sector 3 to "RXCINI.DO" with `boot 3 RXCINI.DO`. See "boot" command in the CLI reference.
3. Set the "mode" to "MT" with `set mode MT`. See the "set" command in the CLI reference.
4. Power off the Backpack Drive
5. Start BASIC.
6. Type `RUN"COM:98N1ENN"<Enter>`
7. Power on the Backpack Drive and follow the on-screen instructions.

To install CP/M-100 on the REXCPM:

1. Set the "mode" to "GEN" with `set mode GEN.` See "set" command in the CLI reference. This will allow the backpack Drive to access large CP/M files.
2. Start BASIC and type `CLEAR 0,60000<Enter>`
3. Run the "CPMUPD.CO" program, following the instructions on the CP/M-100 wiki page.

Troubleshooting

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.

TS-DOS hangs when accessing the disk

Is your Micro-SD card installed correctly? You can verify this by entering CLI mode and issuing the `info` command. If you see the message "No card?" check that you have inserted a working micro-SD card with the required files (See "Quick Start Guide"). You can also check the SD card is present by using the `ls` command. This command should list the files on the SD card. It should show sector0 as one of the directories if you are at the root directory or the drive has been powered cycled.

TS-DOS does not load when bootstrapping (Quick-start procedure)

Make sure that the power switch on the Backpack Drive is set to off. Start the TELCOM program to access the CLI mode (See “Setting the Time and Date for the TELCOM STAT procedure”). Power on the Backpack Drive and look for startup output messages.

“No SD Card” - Check that a properly-formatted micro SD card has been inserted fully into the slot. The label side should face the serial connector. If the SD card is new then it should already be formatted correctly. The drive can only support SD cards up to 16GB, however larger SD cards might work they could be unreliable.

“No Boot File” – Check that the SD card contains the “sector0” files. See “Quick Start Guide” step 5.

PART II: Command-Line Interface (CLI)

The Backpack Drive has two primary states: the TPDD state and CLI state (see above for description). In the TPDD state, the drive will service TPDD Protocol Commands from e.g. TS-DOS on the laptop. To access the Backpack Drive CLI from the laptop the TELECOM application can be used. Set the serial port to 19200bps, 8bits, no parity and 1 stop bit (see "Quick Start" for STAT commands). See your laptop's manual for more information on the use of the TELECOM application. Once connected, the CLI state is accessed by pressing `<Enter>` (CR) four (4) times. This action should bring up the 'C' followed by the '`#`' prompt. You can now issue CLI commands to the Backpack Drive.

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 if the files are Backpack Drive system files or the 6.2 format if the file is to be used by the laptop (the WP-2 uses 8.2 format files). This will prevent issues with the various file types being confused by the laptop file management applications. For convenience the file types .BX and .BY used by REX to store ROM and RAM images are accessible by TS-DOS. These files should only be loaded by REXMGR, but they can be deleted and renamed in TS-DOS. The Backpack Drive does not support long filenames on the SD-Card.

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 TPDD 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

All commands follow the same format: command, followed by a parameter list if required. Only a few commands are available on the Mini-Backpack Drive.

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

For convenience, the TPPD 2 module CLI can also be accessed from a Mac or PC through an RS232 connection using any serial 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 Backpack Drive. Configure the Serial Port as: baud rate: 19200, Data Bits: 8, Parity: None and Stop Bit 1. The DTR and RTS can 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 configurations details will vary and are not covered here.

Command Summary

(*supported by Mini-Backpack Drive)

Command	Parameters	Description
backup	<filename>	Backup current flash firmware
boot	[0|3|4|5|U <filename>]	Set boot sector filenames
bye		Exit CLI State
cal	<voltage>	Calibrate ADC
cat	<filename>	List out filename to screen
cd	<directory name> | ..	Change directory
chk	<src filename> [dest filename]	Check the file format is valid
codo	<src filename> [dest filename]	Change .CO file to loadable .DO file
cp	<src filename> <dest filename>	Copy src to dest file
date	[d/m/y | m/d/y | y/m/d]	Print or sent date
day	[mon | tue | wed | thu | fri | sat | sun]	Set or print day.
dir	[*.<filetail>][-w | -d | -m]	Directory listing
download	<src filename.DO>	Download src file
factory		Reset to factory settings
help		Print list of commands
info		List information
ls	[*.<filetail>][-w | -d | -m]	Directory listing
mkdir	<directory name>	Make directory
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
set	[time <24 | ampm> | date <mdy | dmy | ymd> | flow <ON | OFF> |	Set various configuration options on module.

Command	Parameters	Description
	lines <num 1-255> | update <OFF | ON> | sleep <0 – 60> | boot <ON | OFF> | echo <ON | OFF> | mode <WP2 | MT | GEN>]	
time	[hh:mm:ss | hh:mm:ss [a|p]]	Print of set time
type	<filename>	List out filename
upload	<src filename>	Upload src filename
uploadz	<src filename>	Upload src filename

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 Atmega 644PA. The backup file generated can be used to restore the Mega-Backpack Drive 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.

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

Example:

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

boot - Setting IPL Boot Sector Parameters (Mega/Mini)

```
boot [0|3|4|5|U <filename>]
```

This command allows the Backpack Drive to be configured for different boot and update options.

When the Backpack Drive is powered on it will check the status of the RTS and DTR lines. If these two lines are ‘high’ then it will enter bootstrap mode and output the file located in the ‘sector0’ directory indicated in the boot parameter list. In the example below the boot command is entered without any parameters to display the contents of all the virtual boot sectors. In this case Sector 0 (labeled IPL) is linked with the file *boot*, Sector 3 (M100) is linked with *TS-DOS.100*, Sector 4 (T200) with *TS-DOS.200* and Sector 5 (NEC) with *TS-DOS.NEC*. Each file associated with a sector must exist on the SD card in the

"sector0" directory otherwise an error will be generated when an access is attempted. The *sector0* directory mimics the method provided by the real TPDD 2 drive to upload a file to a laptop that may have no other DOS access capability. The Sector 3, 4 and 5 files are accessed using the boot program or similar that sends another command to the Backpack Drive to select the file to download. For example if the serial port receives 'FF3' it will send the Sector 3 file. This approach uses the same file that is used on the system floppy of the TPDD 2 drive to perform the IPL operation. (see Annex for a more in depth description.)

The U sector, unique to the Mega-Backpack 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 boot sector settings and update file are listed to the screen (as shown in the example below). To change a boot setting, the command is entered with the desired virtual sector to change and the new filename. If the virtual sector and/or filename are not valid this will result in a "`Bad Parameter`" error. (NOTE: No check is made on the existence of the file in the sector0 directory).

When changing the setting for the "U" (update) virtual sector, the filename will again be set to the selected file and 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 boot sector list is again printed out and if the update has been enabled a `*` will printed next to the update sector filename. See Annex A for updating the firmware on the Mega-Backpack Drive. 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.)

The Mini-Backpack Drive only supports sectors 0, 3, 4 and 5 -- it does not support the update virtual sector.

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 Backpack Drive, if the bootloader has not been corrupted. If the bootloader has been corrupted this will require the use of an AVR programmer such as the Sparkfun AVR Pocket programmer and the Arduino IDE to download a new bootloader. The Arduino IDE will also need to be modified to take the bootloader. See Annex A for instructions on recovering a bricked Backpack Drive.

Example (Mega):

```
# boot
Boot Sector Settings
0: IPL  boot
3: M100 TS-DOS.100
4: T200 TS-DOS.200
5: NEC  TS-DOS.NEC
TPDD Update File
U: Upd  TPDD2000.BIN
#
```

Example (Mini):

```
# boot
Boot
0:boot
3:TS-DOS.100
4:TS-DOS.200
5:TS-DOS.NEC
#
```

bye – Terminate CLI session

`bye`

This command is used to return to the TPDD state. Normally after 10s of inactivity the Backpack Drive will return to the TPDD state automatically this prevents the Backpack Drive being locked in the CLI state. The automatic return will not happen if the output in the CLI has been paused.

cal – Calibrate Command

`cal <voltage>`

This command is used to calibrate the onboard ADC voltage reference. It should not be required during normal operation of the module. To use the command correctly, it is necessary to be able to measure the voltage on the board to 2 decimal places. For example if the voltage is measured at 3.30V then the `<voltage>` should be set as 330. The command would then be entered as `cal 330`.

When the command is run it will print out the current calibration value and the ADC value followed by the new calibration value.

The voltage is only read by the `info` command to display to the user. A voltage detection IC is used to read the condition of power supply.

Example

```
# cal 330
Current Cal Value: 11264
Calibrating
383
New Cal Value: 11473
OK
```

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 Backpack Drive will never return to the TPDD 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 xxd
/XXD
OK
# cd ..
/
OK
# cd xxd/trial
/XXD/TRIAL
OK
#
# cd ../..
/
OK
#
```

chk – Check file contents

`chk <src filename> [dest filename]`

This command is used to check the validity of various files on the SD-Card. In moving files from one media to another it is possible to acquire extra carriage returns, line feeds or in the case of the laptop extra ctrl-z's at the end of the file. If only the src filename is provided this command will review the file type and report any errors found. If errors are found then adding a destination filename will generate a file that has been cleaned up or at least attempted. The src filename and dest filename cannot be the same. If is not always successful and the file may remain damaged for use with the laptop.

The examples below show the operation of the command. In the first case the *loadup.ba* file is not a real .BA file it is a .do file so is reported as a Bad Parameter. The next example *ipl.ba* is a tokenized basic program that can be checked. In this case it reports the file size and bytes read, these values should be the same. It also looks for the two types of errors and reports none. The third example is a larger .DO file and in this case progress dots are displayed to show the operation is taking place.

Example:

```
# chk loadup.ba
loadup.ba
Bad Parameter
# chk ipl.ba
ipl.ba
.
BASIC Tokenized file.
File size: 18
Bytes read: 18
CR/LF Errors: 0
EOF Errors: 0
OK
# chk loadup.do
loadup.do
.....................................
File size: 2168
Bytes read: 2168
CR/LF Errors: 0
EOF Errors: 0
OK
#
```

codo – Convert .CO file to .DO file

`codo <src filename> [dest filename]`

This command is used to convert a `.CO` file (machine object) into a downloadable `.DO` file. This command can be used to aid in transferring `.CO` files to machines that do not have a DOS loaded. It is based on the method used by TSLOAD.DO. The src filename must have the `.CO` file type. The file type is checked during the read process and the routine will generate an error if it is the wrong type. If no dest filename is provided, it will be created from the src filename. If the dest filename exists it will abort and generate a "`File Exists`" error response. The command displays progress dots and reports the number of bytes read.

Example:

(The generated file has been listed using the "`cat`" command after conversion)

```
# codo testr.co
testr.co -> testr.do
.....
Created .do of 830 equivalent bytes
OK
# cat testr.do
testr.do
1 '.do Loader Version 0.3
2 'Method adapted from TSLOAD.DO
10 CLEAR824,61788
20 CLS:PRINT" Loading .co file..."
30 READP1:READP2:READP3
40 A=P1:FORI=0TOP2-1:P=P+2:IFP>LEN(D$)THENREADD$:P=1
50 B=(ASC(MID$(D$,P,1))-65)*16+ASC(MID$(D$,P+1,1))-65:POKEA+I,B:NEXT
60 SAVEM"testr",P1,P1+P2,P3:PRINT"Done":END
200 DATA 61788,824,61824
210 DATA
DOLPNDLBNLOAOGIAMCHAHCCBAAPDCCDBPDOBMJCBNLAICCCAPLCBCAPLBBCCPLAG
220 DATA AGMDECCFCBJJPBCCCAPLCBKNPBCCCCPLCBLIPBCCCEPLCBLBPBCCCGPLMJPBDOCE
230 DATA
DCDEPDKPDCDFPDDOCEDCDGPDDOABMDNNBEPBMDFJENPBCBDFPDMDMNBHPBCBDFPD
240 DATA
MNMHBIDKDEPDEPDKDGPDLJNKNBPBMNOAPBNKJEBEKPEPAGAADMDCDGPDCBAAPDAJ
250 DATA
HOMDIKEOMNDIPDNICKDBPDDGANCDDGAKCDDGAAHNDCDEPDKPMNGCHGKPMJKIDJAB

#
```

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 (Mega/Mini)

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

This command is used to print out the current date on the Backpack Drive 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).

On the Mini Backpack Drive the only input format available is `y/m/d`.

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

Example Mega:

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

day – Set or print day of week (Mega/Mini)

`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.

download – Download file

```
download <src filename.DO>
```

This command allows files to be downloaded to laptop serial port from the SD-card. The src filename provided will be downloaded. The routine will wait until it receives an ENTER or for 10s before beginning to send characters. The download can be stopped using Ctrl-C. At the end of the download, the Backpack Drive will wait for an ENTER or for 2mins before closing the file. This should allow enough time to complete any actions on the host.

On the laptop the TELCOM application should be used to perform the download. Within TELECOM use F2 to start and stop downloads of the file. **Note:** TELCOM can only perperly download .DO files.

factory – Factory reset Command

```
factory
```

This command is used to restore the Mega-Backpack Drive to a factory condition; it will set all the parameters back to their default condition.

help – Help Command

```
help [cmd]
```

This command is used to provide brief help information on the CLI commands available on the Backpack Drive. 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 the TEXT app as they are .DO 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 chk codo cp
date day dir download factory help info
ls mkdir mv pwd ren reboot rm rmdir set
time type upload uploadz
#
```

info – TPDD Information (Mega/Mini)

`info`

This command is used to display information about the Backpack Drive. 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 information for the Mega-Backpack Drive includes the board voltage which should be close to 3.3V for the battery version.

The information for the Mini-Backpack Drive includes the limited commands available on the Mini. The power information on the Mini drive is limited to OK or Replace battery.

Example Mega Drive:

```
# info
Minimum-TPDD V0.19  Voltage=3.00V
Fri 03/05/2021 17:52:44
Card Type = SDHC & Size = 15.5GB
Built Feb  4 2021 16:21:43
#
```

Example Mini Drive:

```
# info
Mini-TPDD 0.16  Power OK
Fri 2021/03/05 20:30:41
Card Type = SDHC & Size = 15.5GB
Built Feb  4 2021 16:19:44
CMDS: date [Y/M/D] time [24H] set
CMDS: day [sun,..] reset boot [loc file]
#
```

ls - List Directory contents

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

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 file tails `.CO`, `.DO` and `.BA` are considered valid file types on the laptop. Other file types are visible when used with the basic command.

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 TS-DOS screens.
- `-d` lists the files with their creation dates. The files are time stamped using the time from the on board RTC.
- `-m` lists the valid file types is left justified column format.

The `-m` modifier is exclusive and will override the other modifiers. Examples of using these modifiers are shown below.

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

The file tail `.<>` is used to indicate a directory in the `-w` and `-d` lists. The / is used to indicate a directory in the other lists. This is the same format expected by TS-DOS.

Example:

```
# ls -w
DOS100.DO DOS100.CO XXD   .<> DOS200.DO
TEENYX.DO LOADO .BA TSLD2 .DO OPLML3.DO
REMTST.DC FLO100.DO RAMFIX.BA RF249 .CO
FLO200.DO TEST  .DO TSLD2 .CO TSD200.BX
CRLF1 .DO IPLTST.BA CRLF1F.DO JUNS  .DO
DSKMGR.BA FRED  .<> FMT200.BA TEST  .BX
DOS200.CO DOSXMA.DO RF249 .DO SAR200.BX
OK
#
```

```
# ls -m
DOS100.DO 13940
DOS100.CO 5791
XXD   .<> /
DOS200.DO 13525
TEENYX.DO 2378
LOADO .BA 2521
TSLD2 .DO 1576
OPLML3.DO 945
REMTST.DC 1136
FLO100.DO 1052
RAMFIX.BA 1281
RF249 .CO 4934
FLO200.DO 1076
TEST  .DO 1
OK
#
```

```
# ls -d
SPOTLI~1 /
DOS100.DO 13940 22:48:28  10/05/2017
DOS100.CO 5791 22:05:06  10/06/2017
BOOT 223 21:47:15  09/19/2017
BASIC200.BIN 5376  9:11:07  03/09/2018
XXD /
DOS200.DO 13525 16:09:06  10/09/2017
break
OK
#
```

mkdir – Make directory

```
mkdir <directory name>
```

This command creates a new directory. A duplicate directory name will cause an error. **WARNING:** Do not name a directory "UP" as this is used for internal directory control.

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 exits 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 form 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 TPDD module

```
reboot
```

This command is used to reboot the Backpack Drive. Generally this command is used to initiate an update procedure.

reset – reset Backpack Drive (Mega/Mini)

```
reset
```

This command is used to reboot the Backpack Drive. Generally this command is used to initiate an update procedure on the Mega Backpack Drive.

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. **WARNING**: Do not try to delete the `UP. <>` directory this is used internally for navigating the directory structure.

set – Set Command options

```
set [time <24 | ampm> | Date <mdy | dmy | ymd> | Flow <ON | OFF> |
Lines <num 1-255> | update <OFF | ON> | sleep <0 – 60> |
boot <ON | OFF> | mode <MT | WP2>]
```

This command is used to set various configuration parameters on the Backpack Drive. 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/mon/year `dmy` or year/mon/day `ymd`.

- `flow`: sets up flow control on the serial port. Flow control on `on` or flow control `off`. This should allow XON/XOFF to work on the serial port however the laptop does not implement this option robustly and it may not function as expected.

- `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 Mega-Backpack Drive 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 Backpack Drive 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.

- `boot`: sets the boot option `on` or `off`. When set `off` the power on boot option is disabled.

- `echo`: sets the echo option `on` or `off`. When set `off` the console characters are not echoed.

- `mode`: is used to set the filtering mode of the Backpack Drive. There are currently 3 modes:

    - `MT`: Tandy 100/102/200 and NEC laptop support - Only common Tandy/NEC files (.BA, .DO, .CO) are shown. File sizes are limited to 64KB.

    - `WP2`: WP-2 word processor support - Disables the bootstrapping, and sets filename length to 8 characters. File types are limited to .DO and .DA. The WP-2 *diskette* option can be used to copy and save files from memory; a few other options also appear to work.

    - `GEN`: Generic mode - No file filtering is implemented. This mode is suggested for use with CP/M-100 and is required for installing CP/M-100.

      In future other options might be added such as FB-100 support.

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 (Mega/Mini)

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

This command is used to set the real time clock on the Backpack Drive. 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.

On the Mini-Backpack Drive the command is limited to 24-hour clock input.

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

type – List the contents of the file

`type <filename>`

See `cat` command.

upload – Upload file to Backpack Drive

`upload <src filename>`

This command allows files to be uploaded on the serial port from the host to the SD-card. The command waits for the first char before receiving the file. The upload terminates 3s after the last character is received. Once completed it will display the number of characters uploaded this should correspond to the size of the file. If the disk is full during the upload it will abort and save the partial file. If the src filename exists it will generate a '*File exists*' error.

On the laptop the TELCOM application should be used to perform the upload. Within TELECOM use F3 to configure upload. To exit use ctrl-c during the upload process.

uploadz – Upload file to Backpack Drive terminate with Ctrl-Z

```
uploadz <src filename>
```

This command allows files to be uploaded on the serial port from the host to the SD-card. The command will be terminated and the file written to disk at the first ctrl-z received. If the disk is full during the upload it will abort and save the partial file. If the upload file exists it will abort.

On the laptop the TELCOM application should be used to perform the upload. Within TELECOM use F3 to configure upload. To exit use ctrl-z during the upload process.

Part III: Understanding the Backpack Drive

This section of the document discusses the theory of operation Backpack Drive to provide further information if you want to build you own Drive or repurpose the module to support other devices. As the Backpack Drive features are provided entirely in firmware they should be easily updateable 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.

The Backpack Drive 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 Backpack Drive includes a battery-backed Real Time Clock (RTC) to timestamp the files created on the SD-Card by the laptop. Although the original TPDD2 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 Backpack Drive 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. The second power supply option uses an external DC supply of 3.6 to 6.0 volts, which is fed to a 3.3V low drop out (LDO) regulator. Using the +5V sourced from the BCR port to drive the TPDD 2 module makes for a compact system that draws power directly from the laptop. Of course an external battery pack could also be used on the LDO version of the TPDD 2 module.

The main processor board can be constructed without mounting the RS-232 connector, making it small enough to mount internally to the Model 100/102. In this scenario, the TPDD2 module would be wired directly into the laptop motherboard. It would be necessary to modify the laptop case to provide access to the SD card, if desired. Be aware that the TPDD2 module would still occupy the serial port electrically, unless steps were taken to power down the TPDD2 module (via e.g. an external switch or jumper) when not in use. **NOTE:** ***This option has not been tested.***

Two different versions of the Backpack Drive module can be built, utilizing different processors: a Mega-Backpack Drive that uses the Atmega 644PA processor or a Mini-Backpack Drive that uses the Atmega 328PB processor. Both the Mega and Mini offer the same TPDD2 drive performance, but the Mega-

Backpack Drive offers a much wider array of CLI commands to easily maintain the SD-Card. The Mini-Backpack Drive uses the same processor as used in the Arduino™ family and could possibly be programmed from the Arduino IDE if desired; the bootloader is the same as used by the Arduino family.

Backpack Drive Feature Summary

The following features can be found on the Mega- and Mini-Backpack Drives:

Feature	Mega	Mini
Atmega 644PA		
Atmega 328PB		
BACKPACK DRIVE (M1) Support for:		
Tandy (100/102/200)		
NEC (8201A, 8300)		
WP-2		
RTC (Date, Time, Day)		
RS232 (CTS, RTS, DSR, DTR)		
AA 1.5V battery supply	Opt	□□□
External Supply (3.7 – 6.0V)	□□□	□□□
Micro SD-Card (4, 8, 16GB)		
CLI Interface		
FDC Emulation (M0)		

References and Background Reading

This section provides some useful background reading to understand the TPPD 2 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. TRS-80 Model 100 Owners Manual
2. TRS-80 Model 102 Owners Manual, Catalog No. 26-3803
3. TRS-80 Model 200 Owners Manual
4. TRS-80 Model 100 Reference Manual, Catalog No. 26-3810
5. TRS-80 Model 102 Reference Manual, Catalog No. 26-3803
6. Tandy Portable Disk Drive 1 Operation Manual, Catalog No. 26-3814
7. Tandy Portable Disk Drive Service Manual, Catalog No. 26-3808 (appears to be for TPDD 1)
8. Tandy Portable Disk Drive 2 Operation Manual, Catalog No. 26-3814
9. TPDD Base Protocol: http://bitchin100.com/wiki/index.php?title=TPDD_Base_Protocol definition of the TPDD Base Protocol used on the TPDD 2 drive and partly on the TPDD 1 drive. Incomplete, references out to "dlplus" as an example. Expands on the information in the TPDD1 Service manual. Covers use of the various protocol messages. This is independently provided resource.
10. TPDD Directory Protocol: http://bitchin100.com/wiki/index.php?title=Desklink/TS-DOS_Directory_Access definition of the directory DME response for TS-DOS. When combined with the base protocol covers the complete TPDD TS-DOS software. The extension covers the use of directories primarily used by TS-DOS. Fits into the base protocol and uses one message to signal directory support. This is an independently provided resource.
11. dl.c V1.3 (URL: http://m100.bbsdev.net/) helps to understand the TPDD Base protocol outlined in the Tandy Ref manual and TPDD base protocol web page. Also recommended reading on the protocol web page. Software later extended and enhanced to dlplus.c (URL: https://github.com/bkw777/dlplus). Both written for *nix's but are nevertheless useful references to complement the protocol descriptions.
12. dlplus Github repository https://github.com/bkw777/dlplus download source for the various bootloader options and file management applications/manuals. Includes 'how to' guides for loading and using the various options. Some of the more pertinent text is provided in the Annex to keep the information together, however the repository is the best source for the information.
13. TS-DOS Manual: A Disk-Operating System for the Tandy 100, 102, 200 and the NEC PC-8201 and PC-8300.

14. Microchip: Atmega 164A/164PA/324A/324PA/644A/644PA/1284/1284P Datasheet. MCU used on the Mega TPDD2 module.
15. Microchip: ATmega 328PB Datasheet. MCU used on the Mini TPDD2 module.
16. Maxim Integrated: DS1338 I2C RTC with 56-Byte NV RAM Datasheet. RTC clock.
17. 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.
18. Mircochip: MCP1640/B/C/D Boost Regulator Datasheet. Boost power supply.
19. REX Wiki Page: http://bitchin100.com/wiki/index.php?title=Rex used to store OPTROMS on the laptop and NEC machines, includes TS-DOS.
20. Arduino IDE used to burn the bootloaders.
21. Sparkfun AVR pocket programmer https://www.sparkfun.com/products/9825 used to install the bootloaders on the Backpack Drive. This programmer should be used with care as it does not respect the power in-line when used with modules that use <5V. Once programmed remove immediately to avoid any damage to the Atmega parts. Other programming options are available.
22. XLOADER.EXE: https://github.com/binaryupdates/xLoader used to download hex files to the Backpack Drive
23. 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 Backpack Drive 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 Backpack Drive 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 can be built with either the Atmega 644PA Micro-controller Unit (MCU) or the Atmega 328PB MCU; both MCUs run at 8MHz. A different PCB is used for the two MCU types although the rest of the circuit is identical.

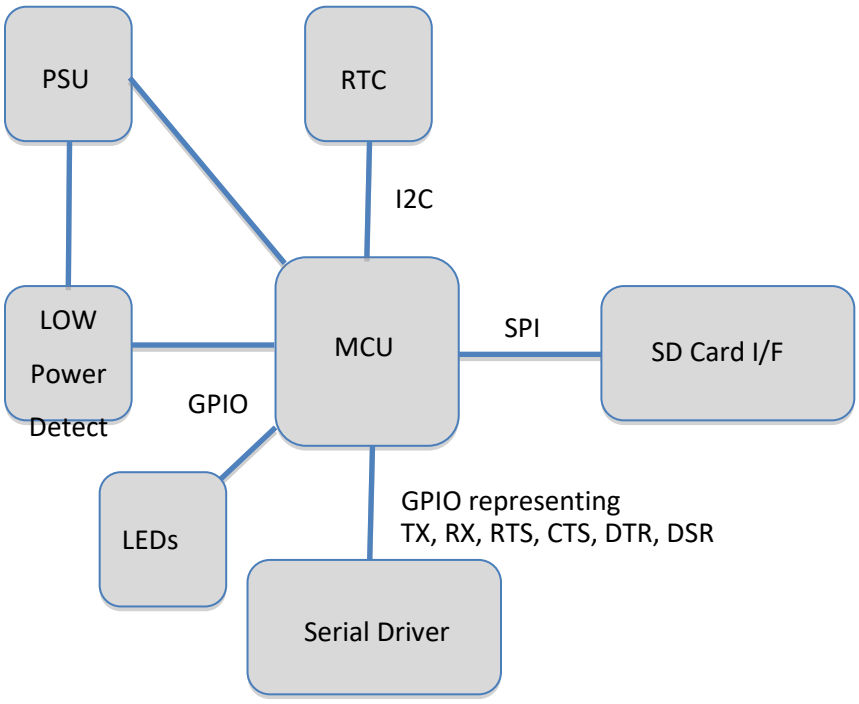


Figure 1: Backpack Drive Block Diagram

The MCU interfaces to 4 separate sections: SD Card interface, Serial Driver interface, RTC and LEDs. The Backpack Drive can be powered either by a 1.5V AA battery or an external supply depending on the power ICs populated during construction. In addition to help prevent SD card corruption issues the power supply is monitored by a voltage reference chip that alerts the MCU to low voltage conditions (<3.1V).

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 accidently 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 currently provide no hardware flow control although software flow control can be enabled via the CLI for the Mega unit. The control pins are there to allow compatibility between the laptop software and Backpack Drive 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. On the Tandy 100/102 at least the DSR/DTR (and CTS/RTS line on the 200) line need to be active for TS-DOS to work correctly. To provide this continued access, the LOOP switch on the line needs to set to LOOP such that TS-DOS will send characters to wake up the Backpack drive. If any file management application reports a failure to communicate with the drive then power cycling the drive should solve the problem, after which you can retry the operation. This problem is present when using the REX to save files to the drive. It can also cause problems with the Model 200, which has inconsistent serial port behaviors.

There are two options to build the power supply. The first option uses a Microchip 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 Backpack Drive 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. The battery or input voltage should be checked. When using TS-DOS a low volt condition is flagged in the directory area to indicate a problem. 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 Mega unit using the CLI. It is also possible

to turn off sleep mode if desired. Blinking yellow and green LEDs (see table below) are used to indicate entry into the sleep state. The MCU will remain sleeping until a new character is received on the serial port at which point it will begin normal activity. As mentioned above while in sleep mode all the pins on the serial port are disconnected and will float to almost 0V. If the serial port requires control lines to be active for operation then the loop back switches should be set to LOOP mode so the serial port appears active. This is the case with the Model 100/102 and 200. Alternatively, if the loop back is insufficient and the unit has entered sleep then it can be power cycled when it is once again needed. 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 ~55-65mA depending on the precise activity, this can increase to 140mA when the SD Card is being accessed.

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 Backpack Drive. 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 list 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 allows 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 LOOP mode when used with any laptop. The through connection is provided to allow the use of the board with other serial ports. The daughter card could be modified to support other serial interface connectors if desired, for example a female DB-9 for use with the WP-2. The upper connector on the serial board is connected to the

programming pins of the MCU. The programming pins also terminate on unused pins on the RS232 connector. This feature could be used to reprogram the MCU even it was located internally to the laptop.

The programming pin header can be used to reprogram either the Mega- or Mini-Backpack Drive using an AVR ISP programmer. On the Mega, 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. On the Mini-Backpack Drive pin 2 is used to set WP-2 mode.

Pin Number	Description	Comment
1	GND	DB-25 pin 7
2	Update (Mega)/ Reserved (Mini)	N/C
3	SPI MISO	DB-25 pin 10
4	Vcc	DB-25 pin 9
5	SPI SCLK	DB-25 pin 11
6	SPI MOSI	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 dependent on supplier.

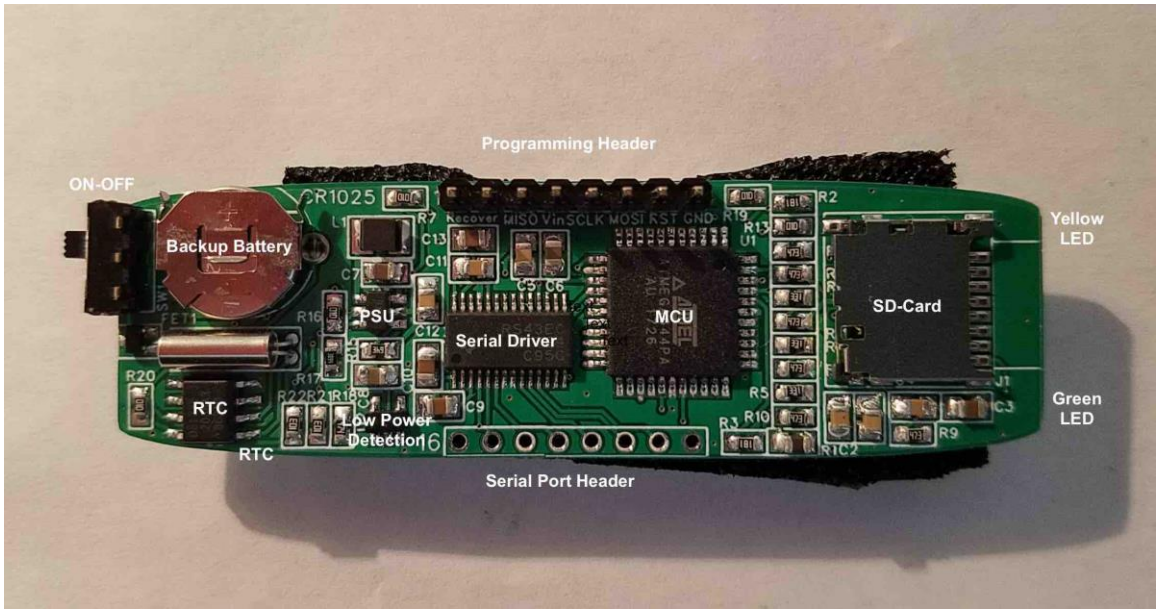


Figure 2: Annotated Backpack Drive without Serial Port (Mega Version 1.1)

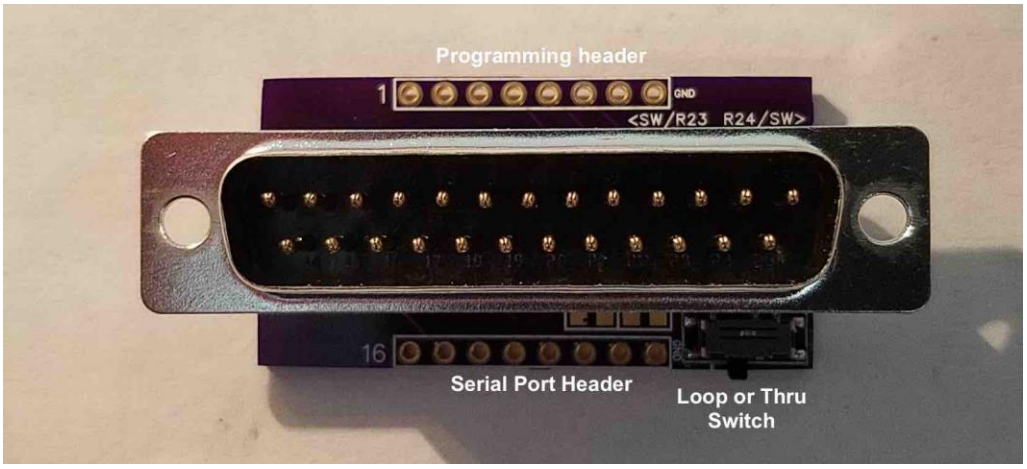


Figure 3: Annotated Serial Port Connector (V1.1)

Firmware Operation

The firmware can operate in three separate states: TPDD state, CLI state and Sleep state. The TPDD state is the default state of operation that is used to provide the TPDD services to the laptop. While in the TPDD state, the serial port is continuously monitored for TPDD Commands (e.g. from TS-DOS), original TPDD2 IPL commands ('FF?') for particular sectors or ENTER/RETURN keys (CR) to potentially start entry to the CLI state. The TPDD Protocol Commands generally start with "ZZ" making them easy to detect. The Boot Loader Commands (See Annex) start with "FF" and will cause the TPDD service to dump the file indicated in the desired Boot Sector (see CLI Boot command). If 4 ENTERs in sequence are received then

the CLI state will be entered. The TPDD state is used to service TS-DOS or similarly based file management applications on the laptop.

If the Backpack Drive is powered on with the RTS and DTR line set 'high' then the module will dump the file indicated in Boot Sector 0 to the serial port. This allows the Backpack Drive to automatically upload a file to the laptop without any action required by the user (apart from configuring the boot files – see "Quick Start Guide"). This mimics the behavior of the original TPDD 2 drive in sending boot sector software to the laptop to allow the TPDD software to be bootstrapped onto the machine.

Using TS-DOS (version 4 or above) in Option ROM is highly advised, as this will make accessing the drive much easier and more reliable. The firmware has been tested with REX Classic and REX# models.

Entering four CRs in sequence will move the Backpack Drive to the CLI state from the TPDD state. This can be done from any terminal program or from TELECOM on the laptop. Once in the CLI state a 'C' and '#' prompt will be displayed, at this point a number of commands can be used to configure the Backpack Drive and perform SD-Card management. See the Command Line Interface (CLI) section for a description of the commands that can be used. If no activity is detected on the serial port for 10s while in the CLI state then the TPDD State will be re-entered. Typing '`bye`' on the terminal will enter the TPDD State immediately.

The Sleep state is entered when no activity has been detected on the serial port for a set length of time. On the Mega-Backpack Drive the inactivity time before sleep can be set via the CLI. There is also an option to turn off the sleep mode if required. The Mini-Backpack Drive will sleep after 5 mins and this time cannot be changed. 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.

NOTE: The REX uses built-in TPDD access routines to save RAM or ROM images to the Backpack Drive. However the REX TPDD client software is not as robust as TS-DOS. When saving such images, please save to a directory that does not contain many files in order to avoid REXMGR timing out while waiting for a response from the Backpack Drive. The Backpack Drive checks for conflicting file names prior to writing any file to ensure that files are not overwritten. It is generally seen as good housekeeping to save images in a separate directory to simplify SD Card maintenance.

Annex A: Firmware Updates

Both the Mega and Mini -Backpack Drive support bootloaders, this allows field upgrades without the need to use an AVR ISP programmer once the bootloader has been installed. The bootloaders used on the Mega and Mini platforms are different as they have different sizes of boot area flash available. The Mini-Backpack Drive only supports updates via the serial port. The Mega-Backpack Drive is updated using the SD-Card. The hex files for the bootloaders can be found on the support page for the Backpack Drive.

Installing the Mini-Backpack Drive Bootloader

The Mini boot loader uses the open source optiboot_flash bootloader[1] (github: https://github.com/Optiboot/optiboot) the same bootloader used by the Arduino 328P family of boards. The bootloader is very small and only occupies ~512bytes. To install the bootloader requires an AVR ISP Programmer (e.g. Sparkfun AVR pocket programmer https://www.sparkfun.com/products/9825) and suitable programming software to load the bootloader hex file. This operation can be completed with the Arduino IDE burn bootloader option but will require modification of the *boards.txt* file (there are a number of web sites that discuss the modification of the boards.txt file for use with non-Arduino boards). The additional text to add to the *boards.txt* file is provided at the end of this Annex. Once installed the bootloader will be entered to wait for the firmware hex file to be loaded through the serial port using, for example, Xloader.exe (a Windows program that provides a very simple GUI interface to AVR Dude. The Xloader support file needs to modified as shown below to support the Mini Backpack Drive). Once the firmware hex file has been uploaded the Mini-Backpack Drive should be fully operational.

Updating the Mini-Backpack Drive Firmware using the Serial Port

The firmware on the Mini-Backpack Drive can be updated very easily with the bootloader. With the new firmware hex file in hand prepare to upload it by briefly shorting the onboard reset pin to ground (see hardware description for pin location), the green and yellow LEDs should flash in sequence to indicate the bootloader is now waiting for a download. Using reset puts the Mini-Backpack Drive into bootloader mode. The serial port should be connected to the computer with the new firmware hex file then use a suitable upload program, such as Xloader.exe, to send the file to the Mini Backpack Drive. Once programming has finished the module should return to normal operation. The serial port should be set to 57600bps. Power cycling the Mini-Backpack Drive before loading the new firmware when in bootloader mode should bring it back to a normal running condition.

If the Mini Backpack Drive does not enter bootloader mode after reset then it will be necessary to re-install the bootloader as described above.

[1] Use 'make atmega328pb AVR_FREQ=8000000L BAUD_RATE=57600 LED=E0 LED_START_FLASHES=2 UART=0 ASM_OUTPUT=1' to compile the bootloader for the Mini-TPDD. This sets the baud rate to 57600, LED to pin E0 and CPU freq to 8MHz.

Installing the Mega-Backpack Drive Bootloader

The Mega bootloader uses the zevero/avrboot (github: https://github.com/zevero/avr_boot) this software uses the 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 ISP Programmer (e.g. Sparkfun AVR pocket programmer https://www.sparkfun.com/products/9825) and suitable programming software to load the bootloader hex file. This operation can be completed with the Arduino IDE burn bootloader option but will require modification of the *boards.txt* file. The additional text to add to the *boards.txt* file is provided at the end of the Annex. Once programmed the bootloader will search for the default file TPDD2000.BIN on the SD-Card. If the file is found this will be programmed into the flash memory and the Mega-Backpack Drive 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 TPDD2000.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 Mega-Backpack Drive Firmware using the SD card

8. Copy the new firmware file to the root of the SD card.
9. Start the TELCOM program and access the CLI (See “Command Line Interface”)
10. Enter command: `BOOT U <filename><Enter>`
11. Press `Y` to enable update
12. The boot status for sector “U” should show an asterisk next to the selected boot filename
13. Reboot the Backpack Drive: `REBOOT<Enter>`
    The boot process should display the message “Normal Update” followed by “Processing FIle”.
14. 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 Mega-Backpack Drive should reboot with the new firmware. The green and yellow LEDs will flash to provide a visual indication of the update progress.

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

```
reboot
TPDD2 BOOT V2.1
Normal Update
Processing File
Update ran
```

Recovering a 'bricked' Mega-Backpack Drive.

If after an update or due to other reasons the Mega Backpack Drive stops responding to TPDD 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 Backpack Drive 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 TPDD2000.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 Backpack Drive.

```
TPDD2 BOOT V2.1
Recovery mode using: TPDD2000.BIN
Processing File
Update ran
```

Arduino board.txt for Mini-Backpack Drive

NOTE: Create a backup of Boards.txt before editing. The Arduino IDE should be closed before editing the file. A subdirectory used to store the hex file will have to be created in this example case the subdirectory is called tpddboot and resides in the bootloader directory of the Arduino IDE. There are a number of websites that describe how to modify the boards.txt file. The new device will appear under the pro mini directory as a sub directory in the board type menu.

```
## Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328PB
## ----------------------------------------------------
## TPDD Drive config

pro.menu.cpu.8MHzatmega328pb=TPDD (3.3V, 8 MHz)
```

```
pro.menu.cpu.8MHzatmega328pb.upload.maximum_size=30720
pro.menu.cpu.8MHzatmega328pb.upload.maximum_data_size=2048
pro.menu.cpu.8MHzatmega328pb.upload.speed=57600

pro.menu.cpu.8MHzatmega328pb.bootloader.low_fuses=0xE2
pro.menu.cpu.8MHzatmega328pb.bootloader.high_fuses=0xD4
pro.menu.cpu.8MHzatmega328pb.bootloader.extended_fuses=0xF7
#Boot file 512B
pro.menu.cpu.8MHzatmega328pb.bootloader.file=tpddboot/optiboot_flash_at
mega328pb_UART0_57600_8000000L_E0.hex

pro.menu.cpu.8MHzatmega328pb.build.mcu=atmega328pb
pro.menu.cpu.8MHzatmega328pb.build.f_cpu=8000000L
```

Arduino board.txt for Mega-Backpack Drive

NOTE: Create a backup of Boards.txt before editing. The Arduino IDE should be closed before editing the file. A subdirectory used to store the hex file will have to be created in this case the subdirectory is called tpddboot and resides in the bootloader directory of the Arduino IDE. There are a number of websites that describe how to modify the boards.txt file. The new device will appear as a new device in the board type menu.

```
##############################################################
## Arduino 644PA TPDD Drive
## ---------------------------------------------------
##############################################################
atmega644.name=TPDD W/ ATmega644P

atmega644.upload.protocol=arduino
atmega644.upload.maximum_size=62488
atmega644.upload.speed=57600

atmega644.bootloader.tool=avrdude
atmega644.bootloader.low_fuses=0xE2
atmega644.bootloader.high_fuses=0xD0
atmega644.bootloader.extended_fuses=0xFF
atmega644.bootloader.path=atmega
atmega644.bootloader.file=tpddboot/AVRboot.hex
atmega644.bootloader.unlock_bits=0x3F
```

```
atmega644.bootloader.lock_bits=0x3F

atmega644.build.mcu=atmega644p
atmega644.build.f_cpu=8000000L
atmega644.build.core=arduino
atmega644.build.variant=standard
```

XLoader.exe devices.txt file for Mini Backpack Drive

NOTE: Create a backup of devices.txt before editing. Xloader.exe should be closed before editing the file.. The new device will appear as a new device in devices menu. This is a copy of the full file contents it is the `TPDD(ATmega328pb);m328pb;stk500;57600;` line that is of interest. This sets the processor to the 328PB and the serial port speed to 57600bps the rate required by the bootloader.

```
Mega(ATMEGA1280);m1280;stk500;57600;
Duemilanove/Nano(ATmega328);m328p;stk500;57600;
Duemilanove/Nano(ATmega168);m168;stk500;19200;
Uno(ATmega328);m328p;stk500;115200;
TPDD(ATmega328pb);m328pb;stk500;57600;
Mega(ATMEGA2560);atmega2560;stk500v2;115200;
```

Annex B: SD-Card Format

Outline structure of SD-Card for use with the Backpack Drives. The SD-Card should be no larger than 16GB as larger sizes may cause file-handling issues.

/ <update files>
TPDD2000.BIN

etc etc

/Sector0
<boot files>
BOOT
BOOTN
TS-DOS.100
TS-DOS.200
TS-DOS.NEC

/help
<help files>
MANCLI.DO
etc etc.

Annex C: Using the Laptop IPL Boot loading option

The Tandy 100/102/200 and NEC PC-8201a/PC-8300 laptops have no 'built-in' disk management software. Therefore, in order to deal with a floppy disk-based file system the laptop has to load a file management application (a.k.a. DOS). The real Tandy Portable Disk Drive 2 (TPDD2) uses a very simple method to load in this application software. When the TPDD2 is powered on if it detects the serial port is active (i.e. DTR is active) then it writes an Initial Program Loader (IPL) program (example shown below) located in sector 0 of the floppy disk onto the serial port. The laptop is set up to read and run the program received on the serial port by first entering `run "com:98N1ENN"` in BASIC prior to powering on the drive. Using this BASIC command also sets the serial port to the active state. This IPL program when completely loaded is run under BASIC. The IPL program then makes a request for the full File Management Application to the drive. In the case of the real TPDD 2 drive the application would normally be called *floppy*. As the drive can support the Model 100, 102 and 200 the IPL program first checks to see which version of *floppy* should be requested. Once loaded the file management program then installs itself as an application that can be used to access the files on the floppy disk. Over time many different versions of file management software have been written for the laptop these range from the sophisticated TS-DOS to the very simple and small Tiny. All the alternate file management applications load in a very similar manner, the IPL when run makes the same request but the application provided is the alternate software.

The Mini and Mega-Backpack Drives mimic the method used by the real TPDD 2 drive by providing an option to boot a file management application. First the SD-Card needs to contain a directory called *sector0* located in the root directory where all the alternate file management applications and the IPL program can be stored. The Backpack Drive then maintains a 'sector 0' boot record in flash memory to point to the required application program. The default case is to use the standard IPL program shown below but to then use the, for example, TS-DOS file management applications for the 100/102, 200 and NEC machines. The IPL program and file management application can be changed using the *boot* CLI command (see the CLI command section). Although it is possible to boot directly into the required file management application by setting it as the IPL file this might cause performance issues as every time the unit is powered on with an active serial port this large file will be dumped to the serial port.

IPL file contents for Model 100/102/200 (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
```

IPL file contents for Model 100/102/200/NEC (BOOTN):

```
10 CLS:PRINT"---INITIAL PROGRAM LOADER III---"
20 PRINT"     WAIT A MINUTE!":CLOSE
25 A$="COM:98N1ENN"
30 IF PEEK(1)=171 THEN M=4 ELSE M=3
35 IF PEEK(1)=148 THEN M=5:A$="COM:9N81XN"
40 OPENA$ FOR OUTPUT AS #1
50 PRINT#1,"FF";CHR$(M);
60 FOR I=1 TO 10:NEXT:CLOSE
70 RUNA$
```

The various different file management applications can be found on Github https://github.com/bkw777/dlplus having been collated by enthusiastic laptop users. Some of the details are reproduced below to highlight areas applicable to the Backpack Drive **however** reference should be made to the Github repository for more up to date information and any manuals etc.

The Easter egg - a reward for getting this far: The RTC on the TPPD 2 module has a battery backup that should last a few years under normal operation. An exclusive extension to the boot feature allows a query for the time, date and day to be extracted that can be used to set the time, date and day on the laptop should it have to be cold reset. The BASIC file below performs the action of requesting the data by sending 'FF7' to the module. It should respond with the current date, time and day. The date and time have to be set to `mdy` and `24` respectively for the laptop. International machines might require setting to the local format for this data.

Example laptop and Mega Backpack Drive RTC management for 100/102 (SYNC.DO):

```
1MAXFILES=2:DEFSTRAF:F="COM:98N1E":OPENFFOROUTPUTAS1:OPENFFORINPUTAS2:C=CHR$
(13):PRINT#1,"FF";CHR$(7)
5INPUT#2,A,B,C:PRINTA,B,C
10DATE$=LEFT$(A,9)
15TIME$=LEFT$(B,8)
20DAY$=LEFT$(C,3)
```

IPL run options

There are a few different versions of the run command that can be used to start the IPL download on the family member machines:

- Tandy TRS-80 Model 100/102: `run "com:98n1e"`
- Tandy TRS-80 Model 200: `run "com:98n1enn"`
- NEC PC-8201, 8201a & 8300: `run "com:9n81xn"`

- Olivetti M-10: `run "com:98n1e"`
- Kytronic KC-85: `run "com:98n1e"`

All except the NEC machines can use: `run "com:98n1enn"`

File Management Application options

It should be noted that some commands available in the File Management Applications listed below such as format, bank etc. are not available on the Backpack Drive as they either do not exist or are not appropriate. In addition, the free space available will generally always be at the maximum as the SD-Card is much larger than the floppy disks the application was designed to use. All the Tandy and NEC versions of file management software available on the GitHub repository have been tested and work with the Backpack Drive. The Backpack Drive also supports directory access but that is only available with TS-DOS.

All the file management files should be in the *sector0* directory on the SD-Card

File management options are listed below for convenience however reference should be made to the Github repository to retrieve manuals and resolve any issues.

Although the file management software can be loaded onto a laptop a better option is to use the TS-DOS OPTROM so as to avoid using RAM and to speed up operations should the RAM be corrupted. The OPTROM is available online as a binary or hex download for programming into a user created ROM or to load into a REX Classic. The use of OPTROMs and REX is beyond the scope of this manual.

The Backpack Drive does not support FDC Emulation mode (M0) i.e. track/sector access. File management programs that require FDC Emulation mode will not work.

TS-DOS

TS-DOS is the most sophisticated and complete file management application however it is also quite large when used as the RAM version. The application does support directories as used on the Backpack Drive. There are 3 version of TS-DOS for the Model 100/102, Model 200 and NEC machines. They all use the run commands described above to load the IPL and then the application. The boot CLI command should be used to set up the files as required. The options below set up TS-DOS for all three machines.

3: TS-DOS.100
4: TS-DOS.200
5: TS-DOS.NEC

For all 3 versions when the installation has completed, which will take several minutes, while still in BASIC type *SAVE "TS-DOS"* this will produce TS-DOS.BA in the main menu. To run TS-DOS simply use the .BA program. The TMP.DO file created in the process can be deleted it is no longer required.

The manual for TS-DOS can be found on the Github repository.

DSKMGR

The DSKMGR file management application has a smaller footprint than TS-DOS and provides similar functionality without some of the ease of use, it does not support directories. There are 2 version of DSKMGR for the Model 100/102 and Model 200. They all use the run commands described above to load the IPL and then the application. The boot command should be used to set up the files as:

3: DSKMGR.100
4: DSKMGR.200

The installers can take several minutes to run.

On the M100/102 the progress is indicated by the PC = xxxx text. When complete this will ask a few questions expecting *Y* or *N* responses. The Self-Clearing and High-Speed should be Y. The sound option is a personal preference. Once returned to the BASIC prompt type *NEW*. The DSKMGR then needs to be loaded to the directory by first using *LOAD "DRIVER.DO"*, then *SAVE "DSKMGR"*. The DRIVER.DO file can be removed with KILL “DRIVER.DO”. Then use NEW to clean up the BASIC area. Use DSKMG.BA in the menu to run the application, not DSKMGR.CO.

On the Model 200 simply type NEW and exit BASIC.

The GitHub repository contains versions for other family members however these have not been tested but should be easy to configure on the Backpack Drive. The repository also contains the manual for using the DSKMGR application.

TEENY

Teeny is the simplest file management application with a very rudimentary CLI to save and load files. It does not support directories. It is also perhaps the hardest to load onto the machine although it has a very small footprint.

There are 3 version of Teeny for the Model 100/102, Model 200 and NEC machines. They all use the run commands described above to load the IPL and then the application. The boot command should be used to set up the files as

3: TEENY.100
4: TEENY.200
5: TEENY.NEC

Loading the application can take a few minutes although it is relatively quick due to its small size. Once loaded the actions taken are different for all 3 machines:

On the Model 100 use *NEW* followed by *CALL 9643* this should generate a list starting with TOP:xxxx. The value xxxx from TOP is then used with *CLEAR 0,xxxx* to set up space for *TEENY.CO*. Teeny can then be used from the main menu.

On the Model 200 use *NEW* followed by *CALL 13072* this should generate a list starting with TOP:xxxx. The value xxxx from TOP is then used with *CLEAR 0,xxxx* to set up space for *TEENY.CO*. Teeny can then be used from the main menu.

On the NEC machine in BASIC type:

```
NEW
L=747
P=PEEK(-3196)+PEEK(-3195)*256
CLEARFREE(""),P-L
```

Teeny can then be used from the main menu.

TINY

Tiny is similar to Teeny in operation however it only occupies 760 bytes of memory making it the smallest footprint. There is only a Model 100/102 version of this file management system and it does not support directories. The boot command should be used to set up the files as:

```
3: TINY.100
```

Once loaded enter NEW and then CLEAR 0,62200. Tiny can then be run from the main menu.

Annex D: WP-2 Cable Configuration

The Backpack Drive can be used with the Tandy WP-2 however an adapter cable is required. The Tandy WP-2 has to use a null cable with a DB-25 Female to DB-9 Female connector. The BELKIN Pro Series DB9 Female/DB25 Female Adapters F2L089 is one possible option. The pin connections are list in the table below.

DB-9 Female	DB-25 Female	Comment
2	3	RX
3	2	TX
4	20	DTR
5	7	GND
6	6	DSR
7	4	RTS
8	5	CTS

Annex E: Atmega port usage

The following tables list the ports used by the Mega and Mini Backpack Drives the information is provided to allow the reuse of the Backpack Drives in other applications.

Mega-Backpack Drive port usage

- PA0 = Unused
- PA1 = Unused
- PA2 = Unused
- PA3 = Unused
- PA4 = Unused
- PA5 = Unused
- PA6 = Unused
- PA7 = Unused

- PB0 = Green LED Control - OUT
- PB1 = Yellow LED Control - OUT
- PB2 = SDDET SD Card Detect - IN
- PB3 = SDCS SD Chip Select - OUT
- PB4 = Unused - used as SS pin do not change to an input!
- PB5 = MOSI Output Part of SPI - OUT
- PB6 = MISO Input part of SPI - IN
- PB7 = CLK Output part of SPI - OUT

- PC0 = Unused
- PC1 = Unused
- PC2 = RS232_SHDN RS232 Shutdown pin - OUT
- PC3 = RS232 EN RS232 Enable pin - OUT
- PC4 = RS232 Invalid indication - IN
- PC5 = SEL2 Jumper setting switch - IN
- PC6 = LBI Low battery indicator - IN
- PC7 = Unused

- PD0, PD1 = UART

- PD2 = RTC_INT RTC Interrupt - INT0 - PCINT18
- PD3 = INT_UART_RX Interrupt from UART - INT1 - PCINT19 - PCIE2
- PD4 = RTS from Serial Port - IN
- PD5 = DTR from Serial Port - IN
- PD6 = DSR to Serial Port - OUT
- PD7 = CTS to Serial Port - OUT

Mini-Backpack Drive port usage

- PB0 = Unused
- PB1 = Unused
- PB2 = Reserved (Tied to SS-)
- PB3 = MOSI Output Part of SPI - OUT
- PB4 = MISO Input part of SPI - IN
- PB5 = CLK Output part of SPI - OUT
- PB6 = Unused
- PB7 = Unused

- PC0 = DSR to Serial Port - OUT
- PC1 = RTS from Serial Port - IN
- PC2 = DTR from Serial Port - IN
- PC3 = Unused
- PC4 = unused
- PC5 = Unused
- PC6 = RESET-
- PC7 = Unused

- PD0 = UART
- PD1 = UART
- PD2 = Unused
- PD3 = INT_UART_RX Interrupt from UART - INT1 - PCINT19 - PCIE2
- PD4 = LBI Low battery indicator - IN
- PD5 = RS232 Shutdown - OUT
- PD6 = SDCS SD Chip Select - OUT
- PD7 = SDDET SD Detect - IN

- PE0 = LED1 - OUT
- PE1 = LED2 - OUT
- PE2 = RS232_EN - OUT
- PE3 = CTS to Serial Port - OUT

Annex F: Combo Part List

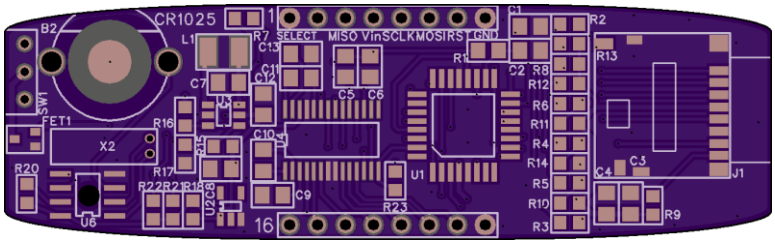
Manufacturer Part Number	Digi-Key Part Number	Ref	Qty	Description
SPC02SYAN	S9001-ND		1	CONN JUMPER SHORTING GOLD FLASH
AB38T-32.768KHZ	535-9034-ND	X1	1	CRYSTAL 32.7680KHZ 12.5PF TH
TRS3243ECPW	TRS3243ECPW-ND	U4	1	IC TRANSCEIVER FULL 3/5 28TSSOP
EG1257	EG2586CT-ND	SW2	1	SWITCH SLIDE SPDT 300MA 4V
3031	36-3031-ND	B2	1	BATTERY RETAINER COIN PC PIN
ATMEGA644PA-AUR	ATMEGA644PA-AURCT-ND	U1 (Mega)	1	IC MCU 8BIT 64KB FLASH 44TQFP
ATMEGA328PB-AUR	ATMEGA328PB-AURTR-ND	U1 (Mini)	1	IC MCU 8BIT 32KB FLASH 44TQFP
MHSS1105	679-1849-ND	SW1	1	SWITCH SLIDE SPDT 300MA 6V
1024TR	36-1024CT-ND	B1	1	BATTERY HOLDER AA SMD TAB
DS1338Z-3+T&R	DS1338Z-3+CT-ND	U6	1	IC RTC CLK/CALENDAR I2C 8-SOIC
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
DB25-PT-1	2057-DB25-PT-1-ND	K1	1	CONN D-SUB PLUG

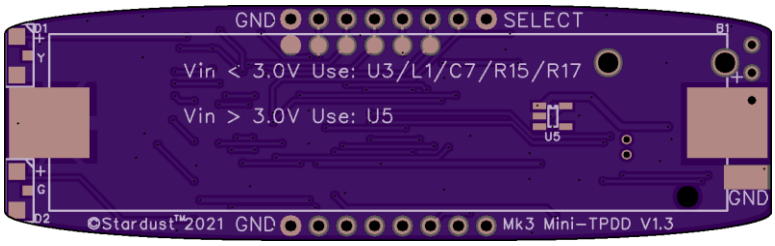
Manufacturer Part Number	Digi-Key Part Number	Ref	Qty	Description
				25POS VERT SLDR
C0805C106M8RACTU	399-7413-1-ND	C8	1	CAP CER 10UF 10V X7R 0805
PH1-08-UA	2057-PH1-08-UA-ND		2	CONN HEADER VERT 8POS 2.54MM
LBR2012T100K	587-2045-1-ND	R1	1	FIXED IND 10UH 150MA 360 MOHM
RC0603JR-0710KL	311-10KGRCT-ND	R21, R22	2	RES SMD 10K OHM 5% 1/10W 0603
RC0603JR-07470KL	311-470KGRCT-ND	R18	1	RES SMD 470K OHM 5% 1/10W 0603
RC0603FR-07100KL	311-100KHRCT-ND	R7, R13, R16, R19, R20, R25	6	RES SMD 100K OHM 1% 1/10W 0603
RC0603FR-07330RL	311-330HRCT-ND	R4, R5, R6	3	RES SMD 330 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
MCP1640CT-I/CHY	MCP1640CT-I/CHYCT-ND	U3	1	IC REG BOOST ADJ 350MA SOT23-6
DM3D-SF	HR1941CT-ND	J1	1	CONN MICRO SD CARD PUSH-PULL
CL21B105KOFNNNG	1276-6471-1-ND	C3, C6	2	CAP CER 1UF 16V X7R 0805
CL21B104KBCNFNC	1276-2444-1-ND	C1, C2, C3, C5, C11, C12	6	CAP CER 0.1UF 50V X7R 0805
CL21B474KOFNNNG	1276-6483-1-ND	C9, C10, C13	3	CAP CER 0.47UF 16V X7R 0805
RC0603FR-0790R9L	311-90.9HRCT-ND	R2, R3	2	RES SMD 90.9 OHM 1% 1/10W 0603

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-074M53L	13-RC0603FR-074M53LCT-ND	R15	1	RES SMD 4.53M OHM 1% 1/10W 0603
CRCW06033M09FKEA	541-3.09MHCT-ND	R16	1	RES SMD 3.09M OHM 1% 1/10W 0603
DMP2240UW-7	DMP2240UWDICT-ND	FET1	1	MOSFET P-CH 20V 1.5A SOT323
CL21B475KOFNFNE	1276-2970-1-ND	C7	1	CAP CER 4.7UF 16V X7R 0805
GRM21BR71A106KA73K	490-14381-1-ND	C8	1	CAP CER 10UF 10V X7R 0805
CRGCQ0603F180R	A129679CT-ND	R2, R3	2	CRGCQ 0603 180R 1%
RMCF0603JT130R	RMCF0603JT130RCT-ND	R2, R3	2	RES 130 OHM 5% 1/10W 0603
NCP302LSN27T1G	NCP302LSN27T1GOSCT-ND	U2	1	IC SUPERVISOR 1 CHANNEL 5TSOP
MCP1640BT-I/CHY	MCP1640BT-I/CHYCT-ND	U3	1	IC REG BOOST ADJ 350MA SOT23-6

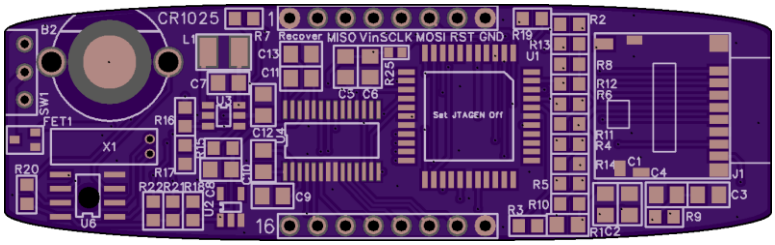
Annex G: PCBs

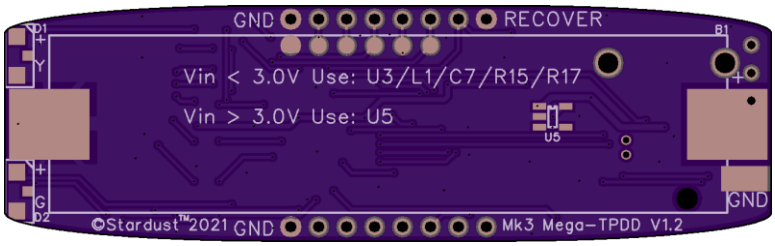
Mini-Backpack Drive PCB



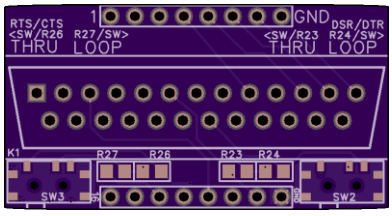


Mega-Backpack Drive PCB





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 Mini-TPDD-2

Github: https://github.com/Optiboot/optiboot

License from Github:

Notes on GPLv2 WRT the Optiboot bootloader:
In general, it can be hard to interpret exactly how the GPL license applies to microcontroller embedded software. Assumptions about logically distinct applications, oeprating systems, dynamically loaded libraries, and the end-user ability to re-compile, re-link, or re-install, don't quite "fit."

There is a "Bootloader Exception" which seems applicable and I've included below, but it's current use seems to be in the PyInstaller utility, which is still very far from a Flashed microcontroller bootloader.

The INTENT of the Optiboot maintainers is that Optiboot may be treated much like the operating system on a larger computer, and that it's GPLv2 status has no effect on the application itself. That means that there are no restrictions to using Optiboot with commercial, closed source, or proprietary applications, and no "viral OSSW" infection of any applications by the bootloader. The only requirement is that fixes and enhancements to Optiboot itself be fed back to the Optiboot project.

Bootloader Exception
In addition to the permissions in the GNU General Public License, the authors give you unlimited permission to link or embed compiled bootloader and related files into combinations with other programs, and to distribute those combinations without any restriction coming from the use of those files. (The General Public License restrictions do apply in other respects; for example, they cover modification of the files, and distribution when not linked into a combined executable.)

Bootloader for Mega-TPDD-2

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.
/----------------------------------------------------------------------------*/
```