NADSBox

User's Guide v1.0.2
For FW Version 1.0

December 2008

© Kenneth D. Pettit
All Rights Reserved

www.kenpettit.com
www.club100.org

Preface

Thank you for purchasing the NADSBox (New Age Digital Storage Box) Serial SD/MMC Card reader. This product is a result of many years of planning, developing, re-developing, stopping and restarting, testing and otherwise spending untold hours staring at my laptop's display scratching my head. The project started in 2002 with high expectations of supporting both Secure Digital and Compact Flash memory formats, including a battery charging circuit, supporting flash download with 5V charge pump circuits, etc. etc.

Ultimately it turned out the project was too large and complex and other tasks would always take a higher priority. A couple of times between 2002 and 2008 I would kick the project off again and make good progress, but was not able to drive it to completion before some other roadblock would stop it again. During all those years it was always in the back of my mind and my dream to complete the product.

Finally after many years of partial development and dealing with that nagging voice saying "finish me, finish me!", I have found new motivation and the end result is before you. The whole story is far more involved than this, and I have spent more hours than I could ever justify had this not been a hobby project. I hope the NADSBox is useful and provides many hours of enjoyment. I will endeavor to support new features and bug fixes in a reasonable timeframe.

Ken Pettit

Limited Warranty

Kenneth D. Pettit's warranty obligations are limited to the terms set forth below: Kenneth D. Pettit ("Manufacturer") warrants the NADSBox Serial SD/MMC Card reader identified herein ("Product") against defects in materials and workmanship for a period of NINTEY (90) DAYS from the date of original retail purchase. If a defect exists, at his option Manufacturer will (1) repair the Product at no charge, using new or refurbished replacement parts, (2) exchange the Product with a Product that is new or which has been manufactured from new or serviceable used parts and is at least functionally equivalent to the original Product, or (3) refund the purchase price of the Product. A replacement Product/part assumes the remaining warranty of the original Product. When a Product or part is exchanged, any replacement item becomes your property and the replaced item becomes the Manufacturer's property. When a refund is given, your Product becomes the Manufacturer's property. This Limited Warranty is offered by the Manufacturer through his official distributor, Club 100 ("Vendor").

EXCLUSIONS AND LIMITATIONS

This Limited Warranty applies only to the hardware Product manufactured by the Manufacturer. The Limited Warranty does not apply to any non-Manufacturer or non-Vendor hardware, even if packaged or sold with the Manufacturerer's Product. The Manufacturer and his Vendor are not liable for any damage to or loss of any programs, data, or other information stored on any media, or any non-Manufacturer hardware or part not covered by this warranty. Recovery of any user data is not covered under this Limited Warranty. This warranty does not apply: (a) to damage caused by accident, abuse, misuse, or misapplication; (b) to damage caused by service (including hardware upgrades and expansions) performed by anyone who is not a Service Provider authorized by the Manufacturer; or (c) to a Product or a part that has been modified in any way, other than firmware upgrades released by the Manufacturer and programmed using the procedures described herein, without the written permission of the Manufacturer.

Disclaimer

IMPORTANT—READ CAREFULLY: This Agreement is a legal agreement between you (either an individual or a single entity) and the Manufacturer for this use of the Product. By purchasing, borrowing or otherwise acquiring and using the Product you agree to be bound by the terms of this Disclaimer. If you do not agree to the terms of this disclaimer, do not use this Product. This document is owned by the Manufacturer and is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties.

DISCLAIMER OF IMPLIED WARRANTIES. To the maximum extent permitted by applicable law, the Manufacturer and his Vendor provide this Product (and all intellectual property therein) and any (if any) support services related to the Product ("Support Services") *AS IS AND WITH ALL FAULTS*, and hereby disclaim all warranties and conditions not otherwise specified herein, either express, implied or statutory, including, but not limited to, any (if any) implied warranties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, and of lack of negligence or lack of workmanlike effort, all with regard to the Product, any intellectual property therein and the provision of or failure to provide Support Services. The entire risk as to the quality of or arising out of the use or performance of the Product, any intellectual property therein, and Support Services, if any, remains with you.

EXCLUSION OF INCIDENTAL, consequential and certain other damages, to the maximum extent permitted by applicable law, in no event shall the Manufacturer or his Vendor be liable for any special, incidental, indirect, or consequential damages whatsoever (including, but not limited to, damages for loss of profits or confidential or other information, for business interruption, for personal injury, for loss of privacy, for failure to meet any duty including of good faith or of reasonable care, for negligence, and for any other pecuniary or other loss whatsoever) arising out of or in any way related to the use or inability to use the Product, any intellectual property therein, the provision of or failure to provide Support Services, or otherwise under or in connection with any provision of this document, even in the event of the fault, tort (including negligence), strict liability, breach of contract or breach of warranty of the Manufacturer or any Vendor, and even if the Manufacturer or any Vendor has been advised of the possibility of such damages.

APPLICABLE LAW. This Agreement is governed by the laws of the State of California.

ENTIRE AGREEMENT. This Agreement is the entire agreement between you and the Manufacturer relating to the Product and the Support Services (if any) and they supersede all prior or contemporaneous oral or written communications, proposals and representations with respect to the Product or any other subject matter covered by this Agreement.

Introduction

The New Age Digital Storage Box (NADSBox) is an SD / MMC/ SDHC memory card reader with an RS-232 interface targeted primarily for Model 100/102/200 laptops. It communicates through the RS-232 interface using the same protocol implemented by the Tandy ® Portable Disk Drives, and it reads and writes files using standard FAT format. This means the NADSBox will work with any existing software written for the Model 100/102/200 and files can be copied easily to and from any desktop or laptop PC that supports an SD/SDHC card reader.

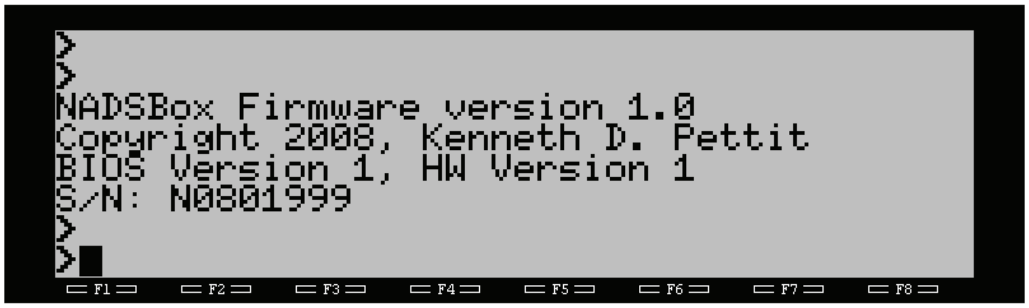
Future versions of the NADSBox firmware will enable use with other computers and for use in other applications, such as data logging through the addition of xmodem file transfer protocol and clear text file writing and appending.

Below is a summary of the features of the NADSBox hardware and firmware:

NADSBox Features	
SD, SDHC and MMC cards	96K Internal File Storage
FAT12, FAT16 and FAT32	Configurable as DCE or DTE
Supports TPDD Protocol	Battery Backed Real Time Clock
Command Line Control	Can be powered through DB-9
Program Flash Upgradeable	AA-Cell Voltage Monitoring
Flash Recover Feature	Expandable HW and SW

Table 1. NADSBox Features

One of the features available on the NADSBox is the command line interface. This interface provides additional functionality not available using existing DOS applications for the Model 100 series laptops. Usage of this interface is described in the Command Line Interface section. Below is a screenshot from the command line interface.



```
>
>
NADSBox Firmware version 1.0
Copyright 2008, Kenneth D. Pettit
BIOS Version 1, HW Version 1
S/N: N0801999
>
>
```

Figure 1. Command Line Interface Screenshot

Quick Start

Okay, okay, so you don't want to read a bunch before trying our your new NADSBox. Chances are you've already totally ignored the documentation and just started using you new device, but here are the Quick Start steps anyway, just for completeness.

Install Batteries

Install 2 AA cells in the battery compartment on back of the NADSBox. The orientation of the AA cells is indicated on the plastic in the battery compartment. Accidentally inserting the batteries in reverse will not damage the unit. Any type of batteries may be used, although testing has only been performed using Alkaline cells.

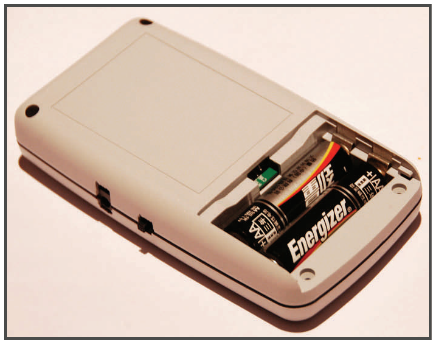


Figure 2. Battery Compartment

Connect Data Cable

Connect the NADSBox to the laptop using the supplied DB25 to DB9 adapter or other cable. The NADSBox is shipped for use with a standard modem cable (DCE) configuration, but a null modem cable can also be used by changing the DCE / DTE configuration jumpers on the RS-232 configuration header, J3. For details on the RS-232 wiring and configuration options, see the Serial Cable Configuration section.

Accept the License Agreement

To activate the NADSBox, run TELCOM with the settings 98N1DNN and go into TERM mode. Turn the unit on and enter the word "activate". This acknowledges agreement to the conditions identified in the Limited Warranty and Disclaimer sections on page 3. Activation can also be performed at 9600 baud by pressing "Enter" to change baud.

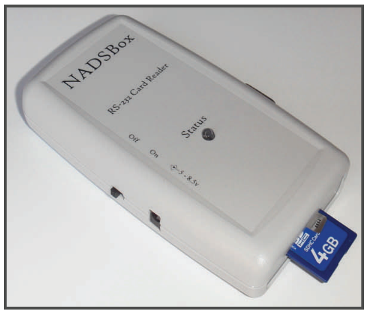


Figure 3. Insert Card

Insert Memory Card

Insert a Secure Digital (SD), Secure Digital High Capacity (SDHC) or MultiMediaCard (MMC) into the memory card socket as shown. The firmware has been tested extensively with various memory card types and sizes ranging from 16MB MMC cards to 4GB SDHC. Additionally the firmware has been tested using FAT12, FAT16 and FAT32 file systems. If the NADSBox is powered on, the LED will blink several times indicating the firmware is reading the card contents.

Launch TS-DOS, TEENY, etc.

Now you can launch TS-DOS, TEENY or any other application that communicates using the TPDD protocol. Ensure the NADSBox is powered on and the status LED has stopped flashing (the flash duration is dependent on the number of used sectors on the card). Press F4 to establish communications with the NADSBox. TS-DOS should respond by presenting a list of files on the card. Use TS-DOS as usual.

Bootstrapping a DOS

Interfacing with the NADSBox for saving and loading files requires a DOS to be loaded on the Model 100/102/200 for the most effective operation. Since the TPDD protocol is fully supported, any existing DOS can be used, however it must first be loaded on the laptop. If the laptop is equipped with an Option ROM based DOS, simply invoke the DOS as usual. If no DOS loaded, the first step for using the NADSBox will be to load or "bootstrap" a DOS into the laptop. This section provides details for bootstrapping either TS-DOS or TEENY. The NADSBox includes a bootstrap command '*boot'* for loading a DOS from the SD card. This command currently only supports bootstrapping of TS-DOS, but may be expanded in future releases to allow automatic bootstrapping of additional DOSes, such as TEENY

Bootstrapping TS-DOS

Bootstrapping TS-DOS is accomplished by typing a small bootstrap program on the laptop or transferring it from the SD card using TELCOM and then running it. The bootstrap program sends the 'boot' command to the NADSBox and then loads and runs the program loader which, in turn, loads TS-DOS. The boot loader program is shown in Figure 4. The load is designed to work with Model 100, 102, 200 and the NEC PC-8201a laptops. The "IF" statement in line 10 tests for the NEC laptop and can be omitted if variable P$ is assigned properly based on the laptop model.

```
10 P$="COM:98N1DNN":IFPEEK(1)=148THENP$="COM:9N81NN"
20 OPENP$FOROUTPUTAS1
30 PRINT#1,"boot ts-dos";CHR$(13);
40 LOADP$,R
```

Figure 4. Bootstrap program listing.

Type the program into BASIC and save it as "BOOTST.BA". It can also be loaded from the SD card by configuring TELCOM as described in the Command Line Interface section and using the 'type' command to echo the file "BOOTST.DO" to the model 100. To load the file using this method, perform the following:

1. Ensure the file "BOOTST.DO" is on the SD card in the current working directory.
2. Type command "type bootst.do" (don't hit Enter yet).
3. Enable the "Capture to file" feature by pressing F2.
4. Specify the filename "BOOTST.DO".
5. Hit enter to echo the file to the laptop.
6. When the file has been received, press F2 again to terminate the capture to file.
7. Exit TELCOM and edit the BOOTST.DO file to remove any extra characters, such as the command prompt.
8. Go into BASIC and load the BOOTST.DO file and save it as BOOTST.BA.

Now that the bootstrap program has been loaded or entered on the laptop, ensure the files shown below are loaded on the SD card in the current working directory and run the BOOTST.BA program.

LOADER.DO	DOS100.CO
DOS200.CO	DOSNEC.CO

When the bootstrap program is executed, it will send a 'boot ts-dos' command to the NADSBox which, in response, will send the LOADER.DO file to the laptop. The laptop will convert the LOADER.DO file to a BASIC program and automatically run it. This BASIC program will determine which model is being used and request NADSBox to upload the appropriate .CO file to complete the upload. The upload may take 20 roughly 20 seconds, so be patient. During the upload, the NADSBox Status LED will flash to indicate activity.

Upon completion of the BOOTST.BA execution, a new file named "TS-DOS.CO" will appear on the MENU and HIMEM will be updated to allow execution of TS-DOS.

Bootstrapping TEENY

TEENY is another DOS available for the Model 100/102, 200 and NEC laptops that is distributed independently from the NADSBox. Automatic bootstrapping of TEENY is currently not supported by the 'boot' command, however it can still be loaded from the SD card using the .DO versions of TEENY and the 'type' command. To bootstrap TEENY using the NADSBox, perform the following procedure:

1. Obtain the appropriate ".DO" version of TEENY for the laptop model.
2. Copy the TEENYX.DO file onto the SD card and insert into the NADSBox.
3. Configure TELCOM as described in the Command Line Interface section.
4. Use the 'cd' command to change to the directory with the TEENYX.DO file.
5. Press F2 to capture to a file on the laptop.
6. Supply the name "TEENYX.DO" and press Enter.
7. Press CTRL-E. This will turn the NADSBox echo off for the current command.
8. Type "type teenyx.do" (without the quotes) and hit Enter. The command will not be echoed so it is not added to the captured file.
9. After the file download has completed, press F2 to terminate the file capture.
10. Exit TELCOM and run BASIC.
11. From BASIC, load the "TEENYX.DO" file and save it as "TEENYX.BA".
12. Run the TEENYX.BA program to install TEENY.

Interfacing with TS-DOS

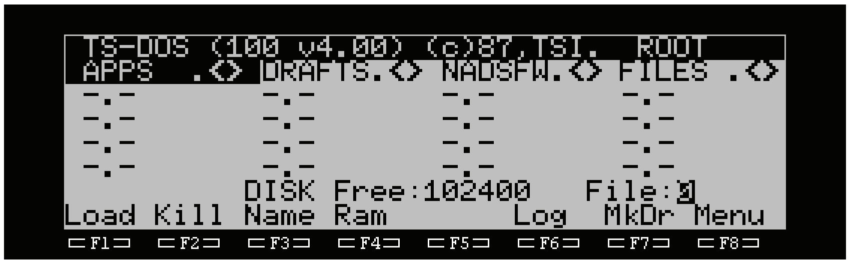
The NADSBox supports the TPDD Protocol as well as TS-DOS extensions for directory management. All TPDD commands, data and filenames are translated into appropriate FAT format accesses. Interfacing with TS-DOS is fairly straight forward since the NADSBox "looks" like a TPDD. Any differences or special considerations when using TS-DOS with the NADSBox are listed in Table 2. The instructions in this section assume that you are familiar with the basic operation of TS-DOS. This manual is not meant to be a comprehensive user's guide for TS-DOS. For additional information on installing and using TS-DOS, please refer to the TS-DOS documentation at Club100.org.

TS-DOS Considerations
TPDD/TPDD-2 Sector access is not supported in v1.0
TS-DOS has a problem displaying more than 2 screens of files from the NADSBox.
TS-DOS can only display files with 6.2 format
TS-DOS can only display directories with 6.0 format
Directories can only be deleted if they are empty
The Format command is currently not supported
The maximum DISK Free that TS-DOS can report is 102400 bytes

Table 2. TS-DOS special considerations

Displaying a File List

Launch TS-DOS in the normal manner either from Option ROM or by using the RAM version. Ensure a properly formatted memory card is inserted into the NADSBox, that the unit is powered on, and that it is connected to the laptop. Press F4 and TS-DOS should establish communications with the NADSBox and display a list of files and directories from the current working directory (see Figure 5). The bottom-most name of the current working directory is displayed in reverse video in the upper right corner of the display. The word "ROOT" in this example indicates the root directory, not a subdirectory. Use the arrow keys and enter key to select files and or directories and the Enter key to navigate to various sub-directories.

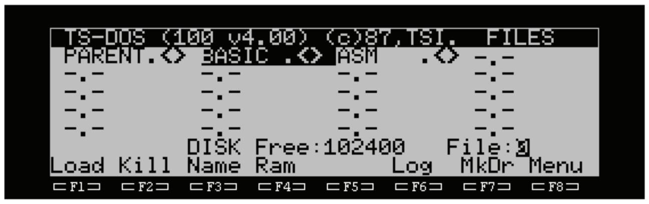


```
 TS-DOS (100 v4.00) (c)87,TSI.  ROOT
 APPS  .<> DRAFTS.<> NADSFW.<> FILES .<>
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
          DISK Free:102400   File:0
Load Kill Name Ram       Log  MkDr Menu
 F1   F2   F3   F4   F5   F6   F7   F8
```

Figure 5. TS-DOS list of files and directories from NADSBox.

Navigating Directories

TS-DOS allows navigation of directories by highlighting a directory in the file list and hitting Enter. This will send a series of commands to the NADSBox to set a reference directory and then "Open" that directory. The NADSBox responds to the "Open" command by performing a Change Directory (see the command *'cd'*) to the referenced directory. Subsequent file listings will be read from the new directory. Additionally, the NADSBox will report the new directory location to TS-DOS for display in the upper right corner of the display (see Figure 6).



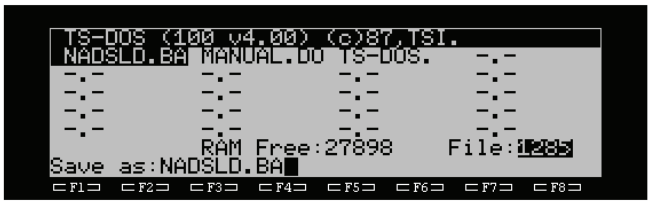
```
 TS-DOS (100 v4.00) (c)87,TSI.  FILES
 PARENT.<> BASIC .<> ASM   .<> -.-
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
          DISK Free:102400   File:3
Load Kill Name Ram       Log  MkDr Menu
 F1   F2   F3   F4   F5   F6   F7   F8
```

Figure 6. Directory Navigation.

For the first entry in any subdirectory, the NADSBox will report the name as "PARENT.<>" to allow navigation to the parent directory of the current working directory. When the NADSBox receives this entry as the Reference Directory followed by an Open command, it triggers a Change Directory operation to the parent directory (the ".." directory).

Copying Files to NADSBox

To copy (save) files to the NADSBox, ensure the file list window is showing the local RAM contents (Press F4 if needed) and move the cursor to the desired file to be copied. Press the F1 key to "Save" the file and type the target filename. Note that the filename must follow the Model 100/102/200 standard 6.2 file format even though the NADSBox can support 8.3 format filenames. Press Enter and TS-DOS will copy the file to the SD card into the current working directory.

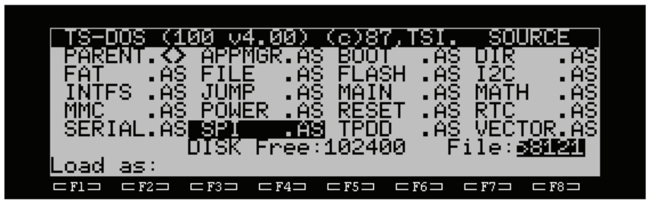


```
 TS-DOS (100 v4.00) (c)87,TSI.
 NADSLD.BA MANUAL.DO TS-DOS.   -.-
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
          RAM Free:27898    File:<28>
Save as:NADSLD.BA
 F1   F2   F3   F4   F5   F6   F7   F8
```

Figure 7. Copying files to the NADSBox.

Loading Files from NADSBox

To load files from the NADSBox, ensure TS-DOS is displaying files from the SD card, and navigate to the appropriate directory and file (see Figure 8). Note that filenames reported from the SD card will be truncated to 6.2 format if the name is larger, and the NADSBox will not be able to "Find" the truncated filename supplied by TS-DOS. Press the F1 "Load" key and type the target filename within RAM. Press Enter and TS-DOS will copy the file from the NADSBox into the laptop's RAM. Note that in the example shown, the file load operation will fail because of the file size and that loading files with non-standard Model 100/102/200 extensions may cause erratic behavior or even a cold-boot.

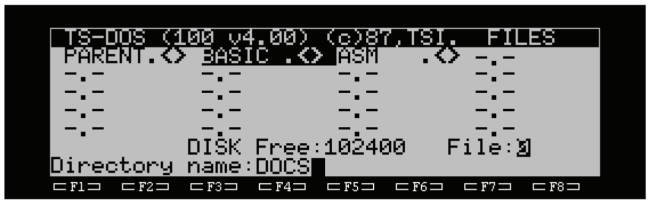


```
 TS-DOS (100 v4.00) (c)87,TSI.  SOURCE
 PARENT.<> APPMGR.AS BOOT  .AS DIR   .AS
 FAT   .AS FILE  .AS FLASH .AS I2C   .AS
 INTFS .AS JUMP  .AS MAIN  .AS MATH  .AS
 MMC   .AS POWER .AS RESET .AS RTC   .AS
 SERIAL.AS SPI   .AS TPDD  .AS VECTOR.AS
          DISK Free:102400   File:<8121>
Load as:
 F1   F2   F3   F4   F5   F6   F7   F8
```

Figure 8. Loading files from NADSBox.

Creating and Deleting Directories

To create a new subdirectory using TS-DOS, use the F4 key to ensure the file list is displaying the NADSBox contents. Navigate to the parent directory where the subdirectory should be created and press F7, "Mkdr". This will cause TS-DOS to prompt you for the name (see Figure 9) of the new directory. Once the name is entered, the NADSBox will create the new directory within the current working directory, "FILES" in the example below. To Delete a directory, highlight the directory name and press F2, "Kill". When prompted if you are "Sure?", enter "Y". The directory will be deleted provided it is empty. If the directory is not empty, you must navigate to the directory and delete each file or subdirectory individually first.

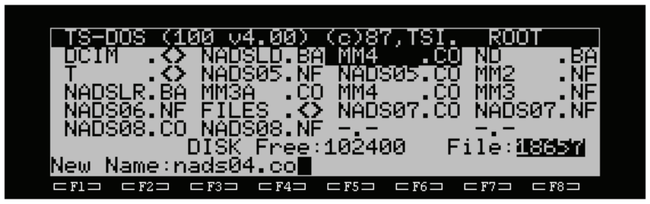


```
TS-DOS (100 v4.00) (c)87,TSI.  FILES
 PARENT.<> BASIC .<> ASM    .<> -.-
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
 -.-       -.-       -.-       -.-
          DISK Free:102400   File:3
Directory name:DOCS
```

Figure 9. Creating a new directory

Renaming Files / Directories

To rename files using TS-DOS, use the F4 key to ensure the file list is displaying the NADSBox contents. Navigate to the file or directory to be renamed and press F3, "Name". This will cause TS-DOS to prompt you for the new name. Type the new name and press Enter. TS-DOS will send commands to the NADSBox to rename the file or directory and then refresh the file list with newly renamed file or directory.



```
TS-DOS (100 v4.00) (c)87,TSI.  ROOT
 DCIM   .<> NADSLD.BA MM4    .CO ND     .BA
 T      .<> NADS05.NF NADS05.CO MM2    .NF
 NADSLR.BA MM3A   .CO MM4    .CO MM3    .NF
 NADS06.NF FILES  .<> NADS07.CO NADS07.NF
 NADS08.CO NADS08.NF -.-       -.-
          DISK Free:102400   File:18657
New Name:nads04.co
```

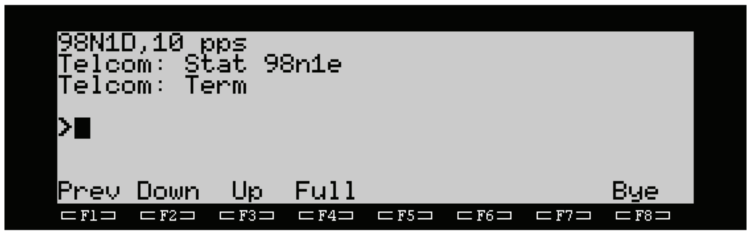
Figure 10. Renaming a file or directory.

Using TS-DOS Extensions for BASIC and TEXT

When using the NADSBox with the resident portion of TS-DOS, the standard "0:" and "1:" notations can be used in both BASIC and TEXT as described in the TS-DOS documentation. However since the NADSBox uses a FAT format when accessing the memory card, it does not have a notion of "sides" of a disk, so both "0:" and "1:" will refer to the same location, namely the region of the FAT volume associated with the current working directory. Otherwise load and save operations using "0:" or "1:" will cause TS-DOS to perform those operations on the card inserted in the NADSBox.

Copying files using TELCOM

In certain instances, it is desirable to copy files to and from the NADSBox without first loading a DOS, such as when bootstrapping TEENY or typing large text documents that require maximum memory on the Model 100/102/200. The NADSBox supports such file copy operations using the Command Line Interface commands 'copy' and 'type'. To use these operations, the TELCOM "STAT" settings should be configured to enable flow control (XON/XOFF). The setting should be configured to "98N1ENN" or equivalent based on the model being used (see Figure 11). This allows both the laptop and the NADSBox to pause the transfer when card I/O operations must be carried out.



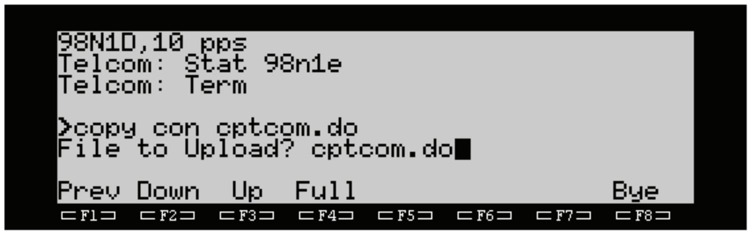
```
98N1D,10 pps
Telcom: Stat 98n1e
Telcom: Term

>
```

Figure 11. TELCOM settings for 'copy con' and 'type'.

Copying to the NADSBox—copy con

The 'copy' command is one of the command line interface commands (see the Command Line Interface section) for copying files on the media card. Normal copy operations will copy from a media card file to a new file as specified in the command invocation. However, there are two special "device files" that can be specified as the source file which cause the copy command to accept source data from the serial port instead of reading the data from the media card. These two device files are "con" and "conb" which stand for "console" and "console binary", where "console" refers to the device connected to the serial port of the NADSBox. Device files cannot be a destination file.

Specifying either of the device files in the 'copy' command allows data to be uploaded from the laptop (or any other PC for that matter) using the "UP" function within TELCOM. To copy files using this method, set the STAT settings as described above, go into TERM mode, type "Enter" to get a command prompt and perform the following steps (refer to Figure 12):



```
98N1D,10 pps
Telcom: Stat 98n1e
Telcom: Term

>copy con cptcom.do
File to Upload? cptcom.do
```

Figure 12. Using 'copy con' to upload files to NADSBox.

1. Type the command "copy con filename" where filename is a valid FAT or Model 100 formatted filename. For compatibility with Model 100/102/200 DOSes, the filename should be restricted to 6.2 format (8.2 for WP-2) .
2. Press F3 ("UP") to instruct the laptop to perform a file upload operation.
3. Type the name of the file to be uploaded and press Enter. This is the name of the file on the laptop that appears on the Menu.
4. Wait for the file upload to complete. The NADSBox will echo the file contents to the laptop during the upload and the status LED will flash as data is written to the card.
5. After the upload has completed, type CTRL-Z to indicate "End of File".

During the copy process, the 'copy' command will send XON / XOFF characters to the laptop as needed to control the flow of incoming data. If the laptop (or any other computer connected to the serial port) is not properly configured to use flow control, then the NADSBox will likely lose data received from the serial port. Writing data to the FAT file system takes an indeterminate amount of time based on the size of the media card, the available free space and the number of file entries in the current directory. While the NADSBox is performing FAT access operations, it will continue to receive data until it's 160 byte RX buffer is full, at which point it will drop characters. The firmware has been tested extensively with flow control turned on and turned off. When flow control is on, the transfer is very reliable, although there is no error checking that a the TPDD protocol provides. A file on the media card named "con" or "conb" must be renamed using the 'ren' command before they can be used with the 'copy' command.

Device File Details

The reason two different device files are provided is to allow transfer of binary data to the NADSBox. When binary data is transferred, it must be processed by the 'copy' command exactly as received. When a CTRL-Z byte (1Ah) is received, the 'copy' command must be able to determine if it is valid data to be written to the media card or if it is an "End of File" indication. Additionally, when receiving TEXT data from the Model 100, using the 'con' device file, the NADSBox will automatically expand all Carriage Return (0Dh) bytes to a Carriage Return / Line Feed pair (0Dh 0Ah). This expansion is not acceptable when receiving binary files. The specifics for each device file are described in Table 3.

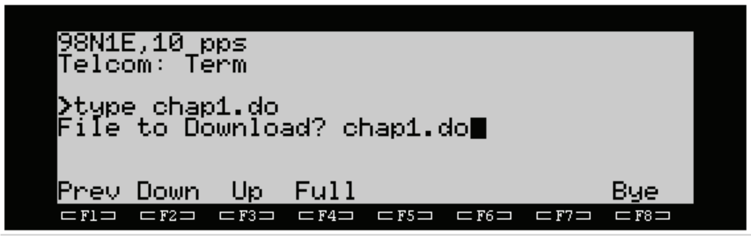
File	Operating feature
con	Expands CR (0Dh) to CR/LF (0Dh 0Ah)
	Detects CR/LF in received data and skips the expansion if found
	Does not echo XON / XOFF bytes received nor write them to the file. Honors the XON / XOFF flow control.
	Closes the file and terminates immediately when CTRL-Z (1Ah) received.
conb	No expansion or modification of any data. Written as received.
	Writes all CTRL-Z (1Ah) bytes to the file that are received within two to three seconds of a previous byte.
	Ignores XON / XOFF flow control characters and does not echo them to the serial port but writes the to the file.
	Closes the file and terminates when a CTRL-Z character is received three seconds or more after reception of any other byte. This is called a "Guard Time". The first byte of the file cannot be CTRL-Z and the final CTRL-Z (End of File marker) is not written to the file.

Table 3. Copy command device file behavior.

Copying from the NADSBox—type

To transfer files from the NADSBox to the laptop within TELCOM (or any computer connected to the serial port), the type command is used to echo the file's contents while performing a download ("DOWN") on the laptop. When using TELCOM's "DOWN" feature to download a file to the laptop, care must be taken to avoid writing unwanted characters to the file, or it will have to be edited. During default operating mode, the Command Line Interface echo's all bytes it receives, including Carriage Returns, and then provides a ">" prompt which is preceded by a CR/LF pair. When TELCOM is downloading to a file, these bytes will be written to the file in addition to the desired content.

To get around this, the firmware's serial receive state machine and the Command Line Interface provide a mechanism to temporarily disable the echo and command prompt reporting. This feature is activated by sending a CTRL-E (05h) character ("E" for Echo) as the first byte of the command. If the CTRL-E is sent as the first byte, then echo will be disabled for the entire command and no feedback will be provided. Since no characters are echoed to the display, it is difficult to detect errors. In firmware version 1.1, a feature will be added to allow typing CTRL-E as the last character before the Enter (CR) key. Typing CTRL-E in the middle of the command line will cause an error condition to be reported. The temporary echo will be restored immediately after the ">" prompt *would have been* sent. To use the 'type' command to send files to the laptop from the NADSBox, perform the following steps (refer to Figure 13):



```
98N1E,10 pps
Telcom: Term

>type chap1.do
File to Download? chap1.do
```

Figure 13. Using 'copy con' to upload files to NADSBox.

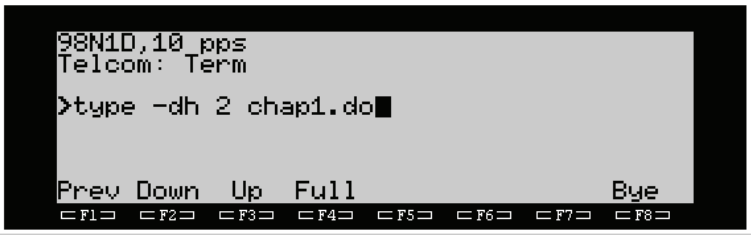
1. Go into TELCOM and set the STAT settings with flow control enabled as described above.
2. Type CTRL-E. This will turn off echo to the RS-232 port.
3. Type the command "type filename" but do not hit Enter. The command will not be displayed. Filename must be a valid FAT filename format, although for compatibility with a Model 100/102/200 DOS, it must follow the 6.2 format (8.2 for WP-2),
4. Press F2 "DOWN" to initiate a file download operation on the laptop.
5. Enter the filename of the destination file on the laptop and press Enter.
6. Now press enter to complete the 'type' command and begin echoing of the file contents. Note that the Enter key will not be echoed.
7. When the file contents have been completely transferred to the laptop (i.e. when no more data is received), press F2 again to terminate the download operation.
8. Note that the ">" prompt is not displayed on the screen, but the NADSBox is ready to accept the next command.
9. Optionally press Enter to display the ">" prompt.

Setting the 'type' Echo Speed

The 'type' command sends data at a conservative rate to allow the Model 100/102/200 to receive the data and have time to write it to memory and perform other operations, such as BASIC tokenization. This is useful when performing a LOAD "COM:98N1E" command after having send the NADSBox a "type somefile.do" command that contains a BASIC program in ASCII format. But this also means that while in TELCOM, the laptop may be able to accept data at a faster rate without losing any information.

To control the flow of data sent to the laptop during both Command Line Interface and TPDD protocol operations, the NADSBox has two controls that it manipulates based on the current mode. These two controls are timer speed and Inter Character Delay (ICD). The firmware uses one of the processor's four internal timers to manage several operations, including sleep period, 5ms system ticks, LED flash timing and serial port timing. This timer is selectively configured for either low speed mode (1.25ms) or high speed mode (1.25ms / 8 = 156.25µs). The ICD specifies the number of timer ticks to wait between successive characters transmitted to the client computer and is also selectively set based on the mode.

When processing the 'type' command, the firmware configures the timer for low speed and sets the ICD for 28 ticks between characters. This is means the default delay between characters is 35ms during when the file contents are being sent to the laptop. This delay can be manipulated by specifying command line options when the 'type' command is issued. Specifying the '-h' option will switch the timer to high speed mode and specifying the '-d num' option will set the ICD value to the decimal value supplied (see Figure 14 and the 'type' command description for full syntax details).



```
98N1D,10 pps
Telcom: Term

>type -dh 2 chap1.do
```

Figure 14. Setting the 'type' echo speed.

Care should be taken when specifying faster echo speeds as the NADSBox is capable of overrunning the Model 100 receive buffer. When using TELCOM with flow control enabled, XOFF characters sent from the laptop will pause the transmission, but extensive testing using short delay settings has not been performed. Notice that the settings in Figure 14 will configure high speed mode with 2 ticks between transmitted characters, or 313µs. This transmission speed will certainly overrun the TELCOM's abilities, however it would be a valid and desirable operational mode with a desktop PC connected to the NADSBox.

Note that the timer controls affect only the operation of the 'type' command. Upon completion of the command, the timer is retuned to the default speed and ICD settings.

Command Line Interface

The Command Line Interface is an alternate method (from the TPDD Protocol) of accessing the NADSBox. It provides functionality for accessing the file structure on the memory card, displaying card and status information, performing flash upgrades, etc. To access the Command Line Interface from the Model 100/102/200, run the TELCOM application and change the port settings as follows:

Parameter	Setting
Baud Rate	19200 or 9600
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	Enabled or Disabled

Table 4. RS-232 Settings.

For Model 100 and 102 laptops, the preferred STAT setting is 98N1D or 98N1E and for the Model 200, the preferred STAT setting is 98N1DNN or 98N1ENN. After setting the STAT settings, go into TERM mode and press Enter. Pressing the Enter key notifies the NADSBox to use the command line interface rather than the TPDD protocol. The NADSBox will reply with a ">" prompt to indicate it is ready to accept commands. A list of commands can be seen by entering the '*help*' command, are also shown in Table 5 and are described in detail in the Command Details section.

Supported Commands			
appmgr	boot	cd	config
copy	date	del	dir
flash	help	info	mkdir
pwd	ren	rmdir	stats
time	trace	type	ver

Table 5. Supported Commands.

The NADSBox uses a 160 byte RX buffer so command line entries are limited to this length. Any data received that is beyond the 160 byte limit will be ignored. No provision are made for editing the command line except using the backspace key. Also due to limited RAM space in the PIC® processor, there is no command history buffer.

For commands that operate on files, this version of the firmware does not support multiple file handles or multiple directory search capability. This means operations other than the *'cd'* command must be performed within the current directory.

Command Line and TPDD Interoperability

While the Command Line Interface provides functionality not currently available with any other TPDD replacement solution, it must interoperate and play well with the pre-existing TPDD protocol. It is important to understand how these two have been integrated and the consequences of certain command line actions as detailed below that can affect the TPDD protocol.

The Serial Receive State Machine

The first thing to understand when using the Command Line Interface is that the NADSBox operations revolve around a state machine (Figure 15) that monitors every received character and directs traffic to various modules (TPDD handler, command handlers, trace buffers, transmit logic for XON/XOFF flow control and "expected response" handlers). The state machine starts off in the IDLE state and spends most of the time in this state. From the IDLE state, it can transition to either the TPDD_INIT state, transition to the CMD_LINE state or stay in IDLE. Receipt of an "M", "R" or "Z" character will initiate the TPDD_INIT.state and then other states from there.. Receipt of XON, XOFF, CTRL-E or CTRL-C will not change the state from IDLE. Any other character will cause a transition to the CMD_LINE state.

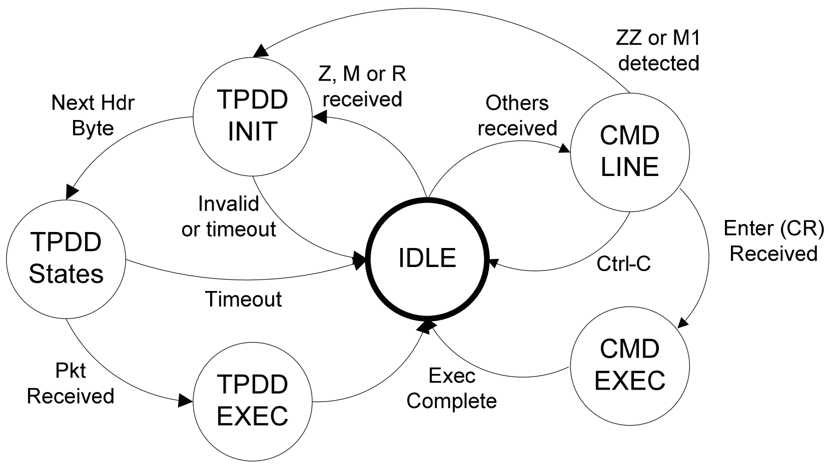


Figure 15. Serial Receive State Machine.

It is important to understand that typing "M", "R" or "Z" as the first letter at the command prompt will temporarily confuse the NADSBox because it will interpret this as a request to go into TPDD protocol initialization state. These characters will not be echoed to the display. In future releases, additional upper case letters will likely cause similar problems if / when FDC sector access emulation is added. Typing "ZZ" as the first two characters will cause even more confusion. Typing these characters as an argument to a command, such as a filename, will pose no problem since the state machine will not be in the IDLE state. Luckily this "confusion" will be temporary as there are protocol reset timeouts as described below. Commands are only recognized when typed using lower case letters, so there should never be a problem.

Protocol Timeouts

To ensure the NADSBox is always responsive, it enforces a couple timeouts related to both command line operations and potentially failed TPDD operations (laptop crashed, unplugged, etc.) The primary timeout used is a four second inactivity timer. This timer is reset upon receipt of every byte and expires if not additional data is received within four seconds. When this timeout expires, the firmware follows the actions described in Table 6 based on the state machine state and current baud rate. As can be seen in the table, if a command is currently processing or waiting for a response (such as the 'flash' command), then the NADSBox will not respond to TPDD requests and will appear to be malfunctioning. When switching between Command Line Interface and TPDD modes, ensure a modal or active command is not executing, such as "copy con" or "dir" on a large directory.

Mode	Actions Taken on 4s Timeout
TPDD_INIT or subsequent states	Reset to IDLE state immediately. Allows reset after failed TPDD access.
CMD_LINE state	Set high-speed "ZZ" or "M1" reception detection mode. Stay in CMD_LINE state. Allows transition from CMD_LINE mode to TPDD_INIT mode if "ZZ" or "M1" sequence received within 160ms period.
9600 Baud Mode	Go into high-speed timer, no-sleep mode to monitor change to 19200 baud.
`Command Processing	No action.
Awaiting Response (Modal command)	No action.

Table 6. Four second protocol timeout actions.

If the NADSBox receiver is in the CMD_LINE mode when the four second inactivity timer expires, it will begin monitoring valid TPDD sequences of "ZZ" and "M1". If either of these two sequences are received with less than 160ms delay between the characters, the state will change to TPDD_INIT and the process the bytes as a TPDD request. This enables the firmware to operate properly with a TPDD client even if a partial command line was typed without terminating with a Carriage Return.

Baud Rate Detection

To support the WP-2, auto baud rate detection between 9600 and 19200 baud has been added to the NADSBox receiver, and the detection for 9600 and 19200 work very differently. When power is first applied, the serial port is configured to operate at 19200 baud. In this mode, the firmware uses the low speed timer mode (1.25ms) and puts the processor to sleep frequently to save power, therefore true auto baud rate calculations that monitor bit pulse durations are not performed.

In 19200 baud mode, the serial receive interrupt service routine monitors the receive data while in IDLE mode to detect receipt of a Carriage Return byte (Enter key) that was transmitted at 9600 baud. When the CR byte is transmitted at 9600 baud and received at 19200 baud, the three high-order bits will be set, and 99% of the time, the byte value will be E6h. So the serial receive ISR checks for this bit pattern with some leniency on the lower bits to determine when to switch to 9600 baud. As a result, while the laptop is transmitting at 19200 baud, the NADSBox can be confused into switching to 9600 baud by typing some of the Shift-Graph characters that match the detection patterns. But these aren’t valid command sequences and the baud rate detection only occurs while in IDLE mode.

While in 9600 baud, when the serial receive state machine is IDLE and the four second inactivity timeout has expired, the auto baud rate detection will go into baud acquire mode. In this mode, the firmware monitors and calculates data bit pulse widths to detect a 19200 baud bit pattern. When 19200 baud is detected, it changes the receiver baud rate to 19200 and reconstructs the received byte using the calculated pulse timing for each data bit transition. Bit patterns that don’t produce a single bit pulse (such as ASCII “0”, 00110000b and ASCII “8”, 00111000) will not cause a transition to 19200 baud. The TPDD header sequence is detected and causes a change to 19200.

Modal Commands

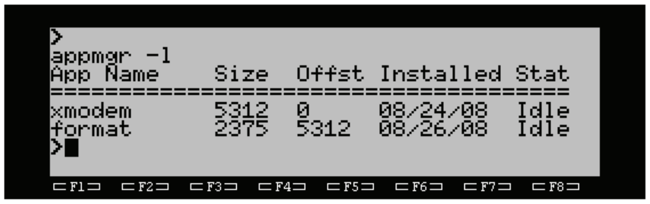
Two of the commands place the NADSBox into a mode that requires a specific response character received through the serial port before the state will transition back t IDLE. These command are the ‘copy con’ and ‘flash’ commands. The ‘copy con’ comand requires a CTRL-Z to indicate “End of File” and the ‘flash’ command requires typing either ‘y’ or ‘n’ to confirm the flash update. As App Manager applications are developed and released, those may additionally have modal options. When using modal commands, ensure the command is properly terminated before trying to use any DOS on the laptop.

Command Details

The follow sections provide a detailed description of each of the commands supported by the NADSBox. The firmware will report the user of invalid commands, invalid command syntax or other errors that might occur. The error messages reported are command specific. For a list of possible error responses, see Table 8 at the end of this section.

---

appmgr
(not fully supported in v1.0)



```
>
appmgr -l
App Name     Size  Offst Installed Stat
=======================================
xmodem       5312  0     08/24/08  Idle
format       2375  5312  08/26/08  Idle
>
```

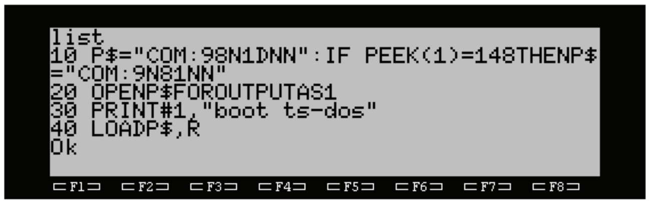
Format: appmgr [-adfl] name

Options:	a	Add new program
	d	Delete program
	f	Show free space
	l	List installed programs

Description
The Application manager (appmgr) command manages installable applications in the processor's upper 30K sector of program flash. Installable applications include things like xmodem, format and clear text transfer. The appmgr command will be completed in a future release and installable applications will be released separately.

---

boot



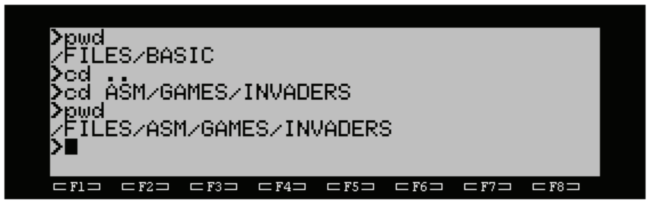
```
list
10 P$="COM:98N1DNN":IF PEEK(1)=148THENP$
="COM:9N81NN"
20 OPENP$FOROUTPUTAS1
30 PRINT#1,"boot ts-dos"
40 LOADP$,R
Ok
```

Format: boot dosname

Options: No options

Descripton
Performs a bootstrap operation of the specified DOS. Currently only "ts-dos" is supported, however additional DOSes may be added in future releases. While this is a command line interface command, it is really only useful when executed from a BASIC bootstrap program. This command reads the file "LOADER.DO" from the current working directory and echoes it to the laptop. Then it communicates with the LOADER.DO program to determine the model type and sends the appropriate DOS file based on the model type. The DOS file must exist in the current working directory on the SD card. When the 'boot' command characters are sent quickly, the command will not be echoed so it doesn't interfere with the BASIC load operation.

cd



```
>pwd
/FILES/BASIC
>cd ..
>cd ASM/GAMES/INVADERS
>pwd
/FILES/ASM/GAMES/INVADERS
>█
```

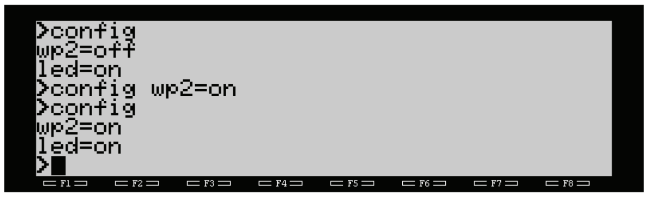
Format: cd dirname

Options: No options

Descripton
Changes the current directory to the dirname specified. This command will parse the dirname argument and change directories recursively if multiple directory names are provided and separated with either the '/' or '\' character. The standard "." and ".." notations are accepted to identify the current and parent directories respectively. All directory searches will be performed relative to the current working directory unless the provided argument begins with the '/' or '\' character.

---

config



```
>config
wp2=off
led=on
>config wp2=on
>config
wp2=on
led=on
>█
```

Format: config [option=value]

Options: No options

Descripton
The config command sets device operational configuration options. Configuration options are saved to the processor's internal data EEPROM and restored upon power-up. No spaces are allowed before or after the equal sign. Multiple options can be configured with a single command line by separating them with a space. Issuing the conig command with no arguments will print the current value of each option. Current options are:

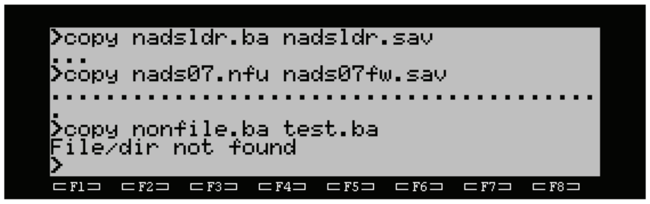
led:	on / off	Turns the Status LED "Normal Power" flash on or off
wp2:	on / off	Configures TPDD protocol for WP-2's 8.2 filename format

Example:

```
config wp2=on
config led=off wp2=off
config
```

When using the WP-2 with the NADSBox, the wp2 options should be turned on prior to performing any TPDD mode accesses. The NADSBox will automatically detect when an 8.2 filename format request is made and switch to 8.2 format, however the WP-2 typically performs directory request operations before sending a packet containing an 8.2 filename. Failure to turn the wp2 option on could cause the WP-2 to crash because it does not handle the 6.2 filename format elegantly. When the wp2 option is turned on, the NADSBox will automatically detect accesses from TS-DOS or file open/save requests with a 6.2 filename format and revert back to 6.2 format automatically.

copy



```
>copy nadsldr.ba nadsldr.sav
...
>copy nads07.nfu nads07fw.sav
.....................................
.
>copy nonfile.ba test.ba
File/dir not found
>
```

Format: copy source destination

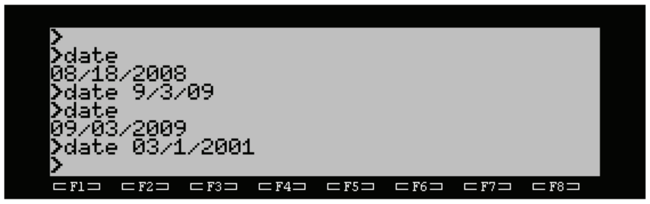
Options: No options

Descripton
The copy command copies the source file to the destination file within the current directory. Due to RAM limitations and the lack of multiple file handles, the copy operation is slow and large (40—50K) file copy operations could take considerable time. During the copy process, the NADSBox will print a '.' character to the serial port for every sector (512 bytes) of data copied. Note that the command will not prompt the user if the destination file already exists in the current working directory and will overwrite existing data.

The source argument can be one of the special device files, "con" or "conb" to copy data from the console (RS-232 port) to the specified destination file. For full details and usage of the "con" and "conb" device files, see the Copying Files Using TELCOM section.

---

date



```
>
>date
08/18/2008
>date 9/3/09
>date
09/03/2009
>date 03/1/2001
>
```

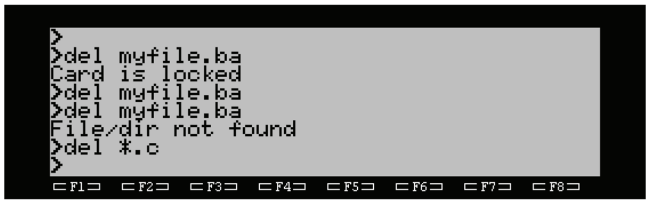
Format: date [mm/dd/yy]
date [mm/dd/yyyy]

Options: No options

Descripton
Sets or displays the Real-Time-Clock date. If no argument is provided, the current date will be displayed. If an argument is specified, it must follow one of the two formats shown. The supplied date will be validated for format (not for accuracy based on leap year, days per month, etc.) and saved to the RTC chip. If the 1st date format is specified, the year will default to the 21st century. If the 2nd format is specified, the century will be saved in the processors internal EEPROM and all dates will be displayed relative to that century.

Note that while the NADSBox firmware will support dates though the year 9999 (v0.9), the structure of the FAT file system reports all dates relative to 1980 and only supports dates through the year 2107. Anyone hoping to live longer than that will have to determine how to deal with this Y2.108K issue. Then NADSBox firmware deals with it by inspecting the century field stored in the processor's internal EEPROM and making all FAT entry dates relative to that year (2000 for 20, 2100 for 21, etc.) minus 20. So for a century value of 20, all dates are relative to the year 1980, for a century value of 21, all dates are relative to the year 2080, etc.

del



```
>
>del myfile.ba
Card is locked
>del myfile.ba
>del myfile.ba
File/dir not found
>del *.c
>
```

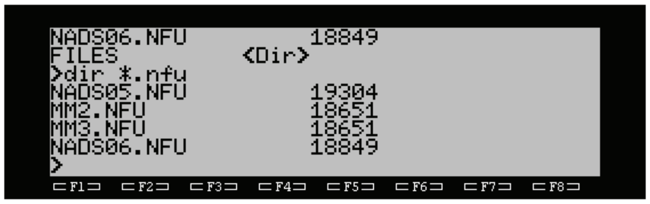
Format: del filename

Options: No options

Descripton
Deletes the specified file within the current directory. The wildcard character '*' can be used in either portion of the filename, however only the first matching file will be deleted. Multiple file deletion capability will be added in future releases. Note that the wildcard character can be used only to specify the entire filename field or completion of a filename field only. A wildcard format such as **"*_O.BA"** is not permitted because of the need to support both 6.2 and 8.3 filename formats.

---

dir



```
NADS06.NFU        18849
FILES          <Dir>
>dir *.nfu
NADS05.NFU        19304
MM2.NFU           18651
MM3.NFU           18651
NADS06.NFU        18849
>
```

Format: dir [-dfw] [file or directory]

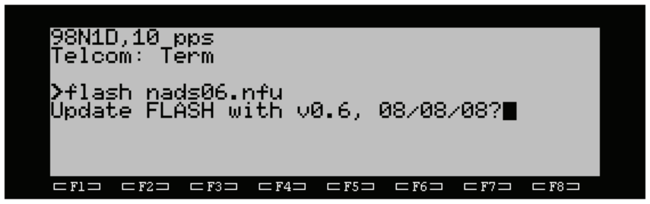
Options:

Options:	d	Only subdirectories
	f	Only files
	w	Use wide format

Descripton
Prints a listing of the files and/or directories in the current directory of within a subdirectory of the current directory. The wildcard character '*' can be used to specify matching of any characters. If no file or directory argument is provided, the command will display a listing of all files and/or directories in the current working directory based on any option flags specified. If a file argument is provided, the command will display that file or all files matching any wildcard pattern specified. If a directory argument is supplied, the command will list all files within that subdirectory according to an options provided. Pressing CTRL-C during a listing will abort the list operation.

---

flash



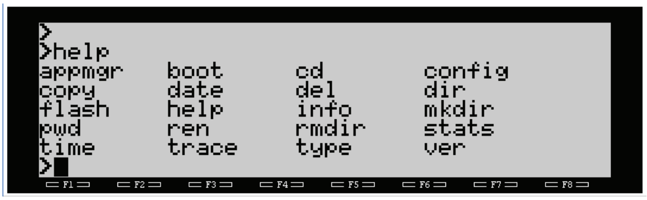
```
98N1D,10 pps
Telcom: Term

>flash nads06.nfu
Update FLASH with v0.6, 08/08/08?
```

Format: flash filename

Options: No options

Descripton
Performs a firmware flash update using the firmware in the specified filename within the current working directory. For further information regarding this command, see the section Upgrading The Firmware.

help



```
>
>help
appmgr   boot     cd       config
copy     date     del      dir
flash    help     info     mkdir
pwd      ren      rmdir    stats
time     trace    type     ver
>
```

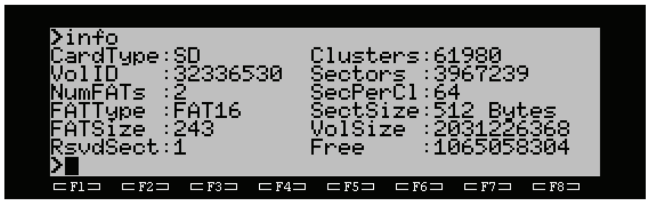
Format: help [command]

Options: No options

Descripton
Prints a list of supported commands to the serial port. If an argument is supplied, the help command will display limited help for that command consisting of the options and invocation syntax.

---

info



```
>info
CardType:SD         Clusters:61980
VolID   :32336530   Sectors :3967239
NumFATs :2          SecPerCl:64
FATType :FAT16      SectSize:512 Bytes
FATSize :243        VolSize :2031226368
RsvdSect:1          Free    :1065058304
>
```

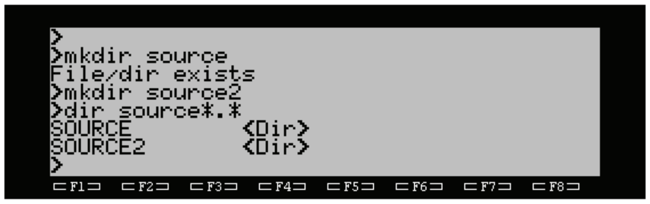
Format: info

Options: No options

Descripton
Prints information regarding the memory card inserted in the SD socket including the card type, FAT format, capacity and free space (along with other information). This command was implemented primarily for debug use, but is provided also for informational purposes. Note that the VolSize and Free fields currently only support volumes up to 4GB because they use 32-bit Byte math. The SDHC specification extended the maximum card capacity beyond 4G by performing address operations in block mode (512 bytes) vs. byte mode, however the NADSBox firmware for reporting these two field has not been updated yet. It will properly report SDHC values for cards that are 4GB or less. Future releases will fix this issue.

---

mkdir



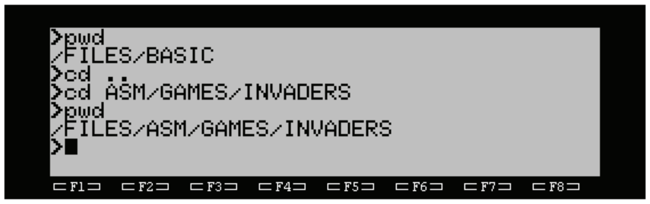
```
>
>mkdir source
File/dir exists
>mkdir source2
>dir source*.*
SOURCE         <Dir>
SOURCE2        <Dir>
>
```

Format: mkdir dirname

Options: No options

Descripton
Creates a new sub-directory within the current working directory. Directory names can be any standard FAT format (8.3 format), however to be compatible with TS-DOS, they should be restricted to 6 characters with no extension. To denote directory access, TS -DOS uses "<>" as the extension which would interfere with a user supplied extension. The directory must not already exist in the current working directory and must follow standard FAT filename convention.

pwd



```
>pwd
/FILES/BASIC
>cd ..
>cd ASM/GAMES/INVADERS
>pwd
/FILES/ASM/GAMES/INVADERS
>█
```

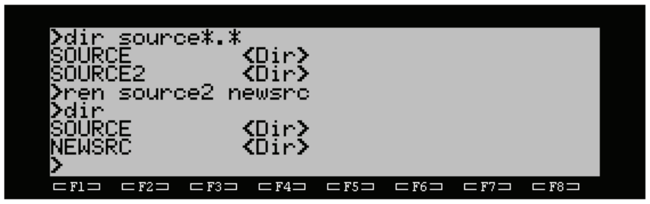
Format: pwd

Options: No options

Descripton
Displays the current working directory. The display uses the '/' character to separate subdirectory names, however both '/' and '\' characters are valid input when entering subdirectories. The current working directory will remain active even if switching between TELCOM and TS-DOS

---

ren



```
>dir source*.*
SOURCE         <Dir>
SOURCE2        <Dir>
>ren source2 newsrc
>dir
SOURCE         <Dir>
NEWSRC         <Dir>
>
```

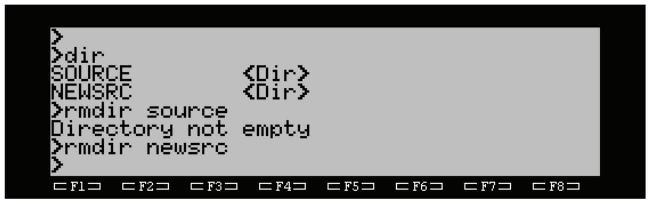
Format: ren source destination

Options: No options

Descripton
Renames the source file or directory as the destination file or directory. Note that due to limitations of the current firmware, the source and destination must be in the current working directory. Future version of firmware will add multiple director management and enable moving files and directories between different subdirectories. The destination name must be a valid FAT format and must not exist in the current working directory.

---

rmdir



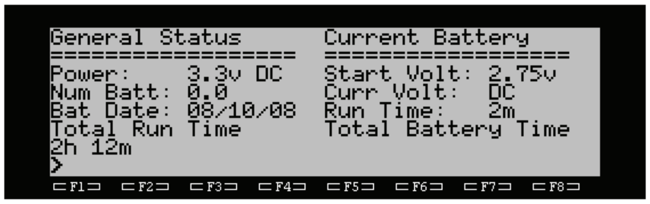
```
>
>dir
SOURCE         <Dir>
NEWSRC         <Dir>
>rmdir source
Directory not empty
>rmdir newsrc
>
```

Format: rmdir dirname

Options: No options

Descripton
Deletes the sub-directory specified by the argument provided. The directory must exist within the current working directory and must be empty. Future versions of the firmware will provide a '-r' option for recursively deleting the directory contents as part of the rmdir operation.

stats



```
General Status      Current Battery
==================  =================
Power:     3.3v DC  Start Volt: 2.75v
Num Batt:  0.0      Curr Volt:  DC
Bat Date:  08/10/08 Run Time:    2m
Total Run Time      Total Battery Time
2h 12m
>
```

Format: stats

Options: No options

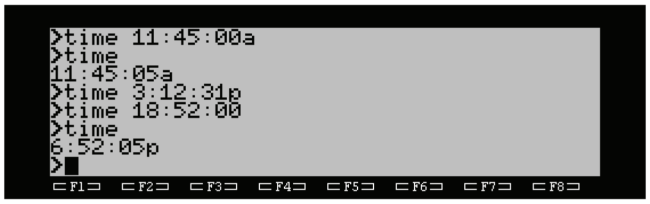
Descripton

Displays current and statistical power information. While the NADSBox is powered on, the firmware keeps track of total operating time, operating time on batteries, number of batteries that have been discharged during operation, etc. The command will display that historical data and information about the current power state including current power source, starting and current battery voltage and V_{CC} voltage level. The battery voltage and V_{CC} voltage are approximate. See Table 7 for statistics descriptions.

Statistic	Description
Power	Shows power source and V_{CC} voltage
Num Batt	Number of batteries discharged during lifetime
Bat Date	Date the current battery was installed
Start Volt	Starting voltage of the current battery
Curr Volt	Current battery voltage. Show "DC" if DC plug inserted
Run Time	Power-on time for the current batter
Total Run Time	Total time the NADSBox has been powered on
Total Battery Time	Total time the NADSBox has run on batteries

Table 7. Statistics descriptions.

time



```
>time 11:45:00a
>time
11:45:05a
>time 3:12:31p
>time 18:52:00
>time
6:52:05p
>
```

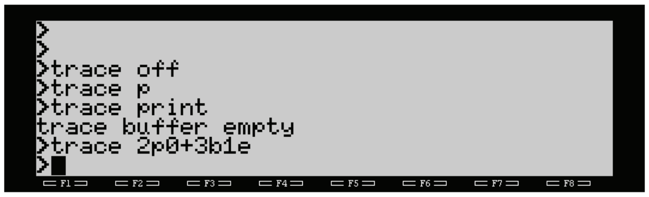
Format: time [hh:mm:ss] [a/p]

Options: No options

Descripton

Displays or sets the current time in the RTC chip. If no argument is supplied the command will read the current time from the RTC chip and print it to the serial port. If a time argument is specified, the command will parse the time and write it to the RTC chip. If an AM/PM argument is not supplied, the firmware will default to AM. A space between the time and AM/PM argument is optional. Time can also be specified in 24 hour format and the firmware will determine AM/PM. Specifying time in 24 hour format does not change the format NADSBox uses to report time - it is always reported in 12 hour format.

trace



```
>
>
>trace off
>trace p
>trace print
trace buffer empty
>trace 2p0+3b1e
>
```

Format:	trace [on off reset] trace print [filename] trace nPo+mBc
Options:	No options

Descripton

Manages a 1K serial data trace buffer for logging both received and transmitted data. The trace buffer is divided into a 114 byte section to record the byte RX/TX direction and a 910 byte storage section.. The trace buffer works by recording every RX and TX byte received after a specified trigger condition has been detected. Data will be logged until the trace buffer is full, and then logging will stop. After data has been logged to the trace buffer, it an be printed or saved using the 'trace print' or 'trace print filename' commands respectively.

Trigger Conditions

To enable trace buffer logging, a trigger condition must be specified. Triggers can be as simple as "start with the next byte" or more elaborate, such as "start after 3 packets have been received that contain opcode 07h, followed by receipt of 5 bytes of value 2Ah". Specify a trigger using the 'trace #Po+#Bc" format of the command. This will also enable the trace operation so it starts looking for the trigger condition. Data logging will begin with the packet or byte that triggered the trace log. The trigger condition format is as follows:

nPo	Search for receipt of 'n' TPDD packets containing opcode 'o'. The 'n' parameter is represented in decimal format and is optional. If 'n' is not specified, it will default to a value of one The 'o' parameter is specified in hexadecimal format and is also optional. If 'o' is not specified, then all packets will be counted to match the 'n' parameter. 'P' can be upper or lower case.
mBc	Search for receipt or transmission of 'm' bytes of value 'c'. The 'm' parameter is represented in decimal format and is optional. If 'm' is not specified, it will default to four so the 'CRLF>" prompt following the 'trace' command is not logged. The 'c' parameter is specified in hexadecimal format and is optional. If 'c' is not specified, all bytes will be counted to match the 'm' parameter. 'B' can be upper or lower case.
.	
nPo+mBc	Trigger when the packet condition has been met followed by the byte condition being met.
mBc+nPo	Trigger when the byte condition has been met followed by the packet condition being met.

Managing the Trace State

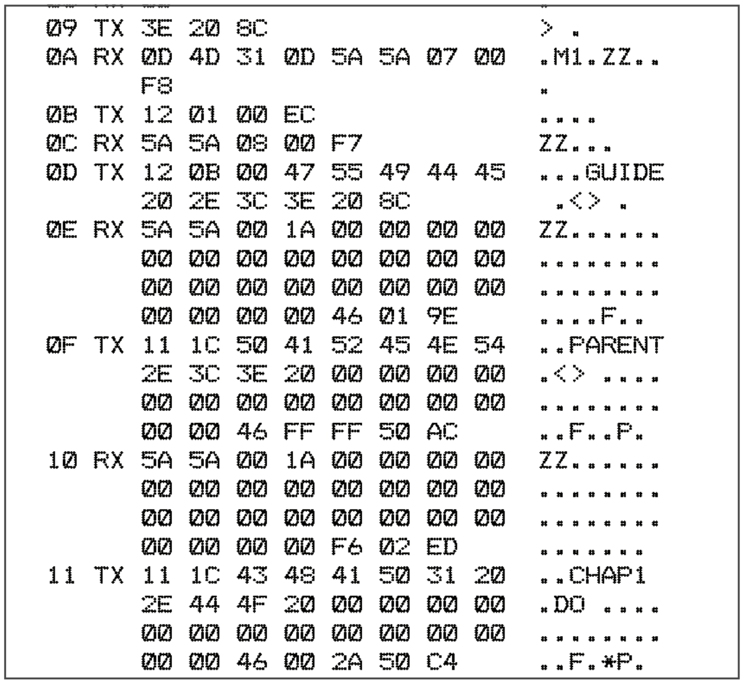
Once a trigger condition has be specified, the trace operation can be turned on or off, or can be "retriggered" using the same trigger condition previously specified. To manage the trace state, use any of the following commands:

trace off	Turns data tracing off. Any data in the trace buffer will retained.
trace on	Turns tracing on using the previously configured trigger.
trace reset	Retriggers the trace using the previously configured trigger. Erases existing trace buffer data.

When using 'trace off', If the trigger condition has been met and the trace buffer is not full, the 'trace off' command will be logged to the trace buffer. The firmware will automatically detect and remove this command from the buffer, therefore it should be the first command issued after capturing a trace.

Printing and Saving Traces

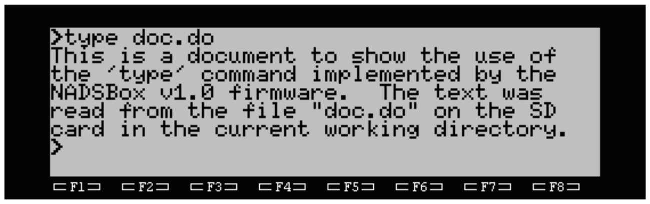
After the trace buffer has been filled or partially filled with data, the 'trace print' command will display the buffer contents formatted for a 40-column display. During the trace print operation, the CTRL-S and CTRL-Q sequences can be used to pause and resume the output, and CTRL-C will terminate the print. The same formatted buffer report can be saved to a file in the current working directory of the SD card using the 'trace print filename' command. A sample output of the 'trace print' command is shown in Figure 16. As shown in the figure, RX and TX bytes are grouped and each group is assigned a sequence number for reference. Bytes are reported in the order they were received or transmitted.



```
09 TX 3E 20 8C                  > .
0A RX 0D 4D 31 0D 5A 5A 07 00   .M1.ZZ..
      F8                        .
0B TX 12 01 00 EC               ....
0C RX 5A 5A 08 00 F7            ZZ...
0D TX 12 0B 00 47 55 49 44 45   ...GUIDE
      20 2E 3C 3E 20 8C          .<> .
0E RX 5A 5A 00 1A 00 00 00 00   ZZ......
      00 00 00 00 00 00 00 00   ........
      00 00 00 00 00 00 00 00   ........
      00 00 00 00 46 01 9E      ....F..
0F TX 11 1C 50 41 52 45 4E 54   ..PARENT
      2E 3C 3E 20 00 00 00 00   .<> ....
      00 00 00 00 00 00 00 00   ........
      00 00 46 FF FF 50 AC      ..F..P.
10 RX 5A 5A 00 1A 00 00 00 00   ZZ......
      00 00 00 00 00 00 00 00   ........
      00 00 00 00 00 00 00 00   ........
      00 00 00 00 F6 02 ED      .......
11 TX 11 1C 43 48 41 50 31 20   ..CHAP1
      2E 44 4F 20 00 00 00 00   .DO ....
      00 00 00 00 00 00 00 00   ........
      00 00 46 00 2A 50 C4      ..F.*P.
```

Figure 16. Sample trace print output.

type



```
>type doc.do
This is a document to show the use of
the 'type' command implemented by the
NADSBox v1.0 firmware.  The text was
read from the file "doc.do" on the SD
card in the current working directory.
>
```

Format: type filename

Options: h High speed timer
d # Set Inter-Char Delay

Descripton
Echoes the contents of the specified file to the RS-232 port. The operation can be paused using the CTRL-S key and resumed using CTRL-Q. Also, the echo can be cancelled by keying CTRL-C. The contents of the file are sent in RAW format with no formatting for word-wrap or "pretty" display of binary data.

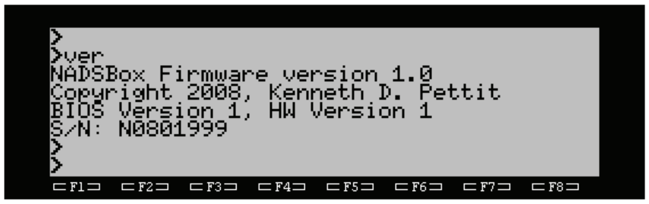
The default 'type' sped can be modified using the 'h' and 'd #' options. For complete details of these options, see the Setting the 'type' Echo Speed subsection of the Copying Files using TELCOM section. Note that if the 'd' option is supplied, a decimal formatted timer-tick count must also be supplied separated by a space. The default Inter-Character Delay (ICD) is 28 timer ticks of the low speed timer (1.25ms) for an ICD of 35ms. This value allows adequate time for processing of BASIC tokens when performing a LOAD "COM:98N1D" operation from BASIC. Care should be taken when reducing the ICD to ensure received data is not lost.

Sample 'type' speed settings:

type -h file	ICD = 28 * 156.25µs = 4.374ms
type -d 15 file	ICD = 10 * 1.25ms = 18.75ms
type –dh 5 file	ICD = 5 * 156.25µs = 781.25µs

---

ver



```
>
>ver
NADSBox Firmware version 1.0
Copyright 2008, Kenneth D. Pettit
BIOS Version 1, HW Version 1
S/N: N0801999
>
>
```

Format: ver

Options: No options

Descripton
Displays the current firmware version, the BIOS & hardware versions and unit's serial number. The BIOS and hardware versions are programmed at the factory and are specific to the particular NADSBox. This information is used when upgrading the firmware in the event the hardware or BIOS code must be revised in future versions of the NADSBox. The serial number can also be viewed within the unit's battery compartment and on the PCBA.

Error Responses

The following table lists all potential error responses from the command line interface commands. This list does not include responses specific to any applications that were installed using the Application Manager.

Error text	Command(s)
unknown command	General error message
syntax error	Any
argument invalid or missing	cd, copy, del, mkdir, ren, rmdir
dir not found	cd
dir not empty	rmdir
Card not formatted	cd, copy, del, dir, flash, info, mkdir, pwd, ren, rmdir
Invalid format on card	
No Card Inserted	
Card communication error	
Unexpected end of cluster	
Sector read error	
Low voltage detected	
Sector write error	copy, del, mkdir, ren, rmdir
Card is full	copy, mkdir
Card is Locked	copy, del, mkdir, ren, rmdir
Directory full	copy, mkdir
File is Read Only	copy, del, ren
Invalid filename	copy, mkdir, ren
File/dir exists	copy, mkdir, ren
File/dir not found	copy, del, ren, rmdir
trace buffer empty	trace
invalid trigger format	
count too large	
invalid trigger value	
unknown config option	config
invalid option value	config

Table 8. Command Line Error Responses.

About the Hardware

Architecture

The NADSBox hardware is based around a Microchip PIC® 18LF2620 microcontroller with interfaces for RS-232 and SD/MMC memory cards. Also included in the architecture are external Real-Time-Clock and Serial EEPROM storage and a power selector / monitor circuit. Figure 17 shows the details of the hardware architecture.

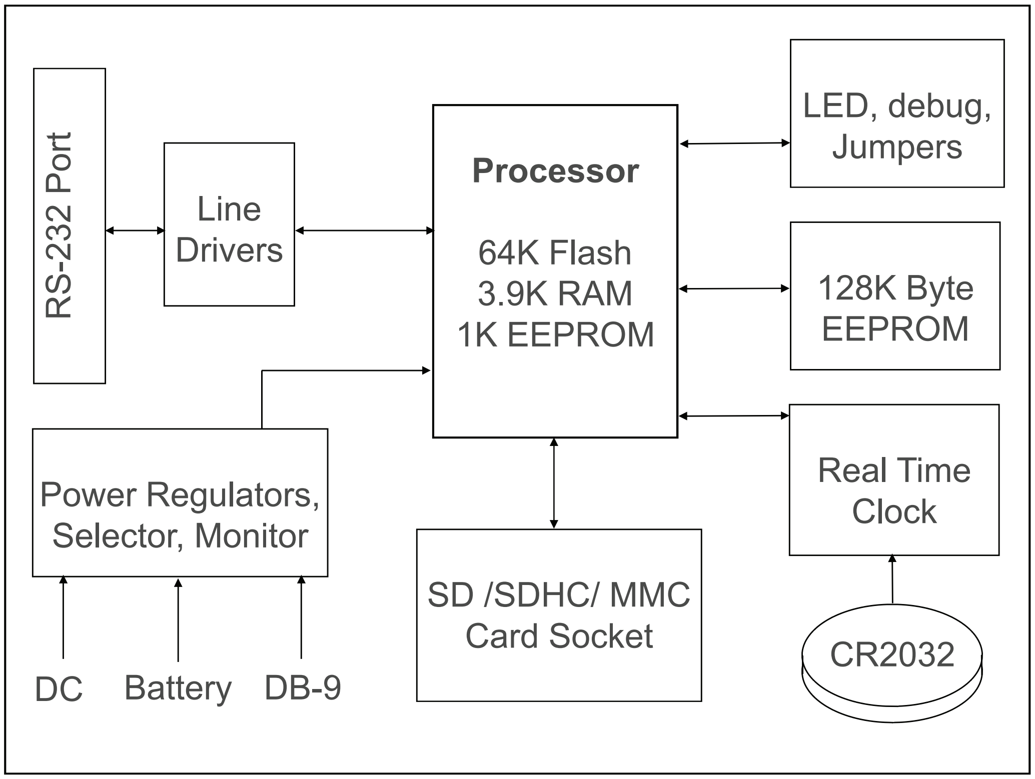


Figure 17. NADSBox Hardware Block Diagram.

The components used in the NADSBox are designed for low power operation and the firmware puts the components in "sleep" mode when not being accessed to reduce operating power even further, including the processor. The firmware uses an interrupt service routine to read and store commands from the RS-232 port and then processes those commands after it has been completely received.

During operation, the firmware also monitors card insertion and removal operations, operational time and power status / battery voltage. As the power state changes or the battery voltage drops too low, the firmware provides memory card write protection as needed to avoid damage to the card or corruption to the FAT structure or other contents.

Power Options

The NADSBox can be powered from 2 AA cells, from an external DC source, or from voltage provided on the DB-9 connector. When multiple power sources are available, the internal power selector circuit will choose power in the following order:

1. DC-Jack Power
2. Battery Power
3. DB-9 Power

The reasoning for selecting battery power before DB-9 power is because the assumption is that DB-9 power will be provided through a hardware modification on the Model 100/102/200. If user has enabled this option and has additionally provided AA cells in the NADSBox, this is an indication to conserve power draw from the laptop batteries by using local battery power.

AA Cell Power

The NADSBox incorporates a low-voltage DC to DC converter specifically designed for use in battery powered applications. The converter will provide the required V_{CC} voltage with input voltages as low as 1.8v. The circuit also includes a "reverse battery" protection diode to protect against accidental reversed installation of AA cells. As a result, the voltage delivered to the DC-DC converter is about 0.2v lower than that of the AA cells. This increases the required minimum voltage for guaranteed working range closer to 2.0v. As the voltage drops below this point, the regulator will provide a V_{CC} voltage that is less than the target 3.3V.

Since most memory cards can operate as low as 2.7v, the NADSBox firmware allows normal operation with V_{CC} voltages lower than 3.3v. The firmware continually monitors the voltage on the AA cells as well as the generated V_{CC} voltage, and allows operation with AA cell voltages as low as 1.82v, which will generate approximately 2.9v V_{CC} voltage. When operating in this region, the firmware will present "warning" flashes (two quick flashes) on the status LED. If the AA cell voltage drops to 1.82v or below, the firmware will disallow SD card accesses to protect the memory card. The protection will be indicated by three quick flashes which are more frequent than the warning flashes.

The memory card protection will continue until fresh AA cells or DC voltage is supplied. The warning flashes will be visible as long as the generated voltage is high enough to drive the LED, or approximately 2.5v. If the warning protection LED indication is not visible, the firmware will continue protection of card access. The PIC® processor will stop functioning if the V_{CC} voltage falls below 2.0v, although the RS-232 line drivers will fail before this occurs..

Since the onboard regulator will convert lower voltages to the required V_{CC} level, any type of AA cell can be used with the NADSBox. For a report of the current voltage on the AA cells and the generated V_{CC} voltage, use the '*stats'* command.

DC Power

Another option for powering the NADSBox is via an external DC power source. The onboard 1.3mm DC-Jack accepts voltages from 4.5V to 8.5V, center pin positive. The external adapter should be able to supply a minimum of 150ma of current. Power may also be supplied via a standard PC USB port using a USB to DC Plug converter cable.

When operating from DC Power, the AA cells will be disconnected from the circuit via a mechanical switch in the DC Jack,. This provides maximum battery power savings when using DC power. However it also means the battery voltage cannot be measured while the DC plug is connected. When reporting power statistics via the *'stats'* command, the firmware will automatically detect the presence of DC power and skip the battery voltage calculation process.

DB-9 Power

Power can additionally be provided across the RS-232 connector on Pin 9. Since this is a non-standard usage of this pin, this feature is disabled when the NADSBox is shipped. To enable the feature, a configuration jumper must be installed on header J6 to allow the input voltage to drive the power circuit (refer to Figures 18 and 19). The voltage on the DB-9 connector should be in the range of 4.5V to 9V. The NADSBox is shipped with 2 spare jumpers on header J8 in the GND and VCC positions. Either of these jumpers can be removed and installed on J6. To power the NADSBox through Pin 9 from the Model 100/102/200, an internal modification must be made to the laptop to provide 5V on pin 25 of the RS-232 port, and a cable must be provided which wires DB-25 Pin 25 to DB-9 pin 9.

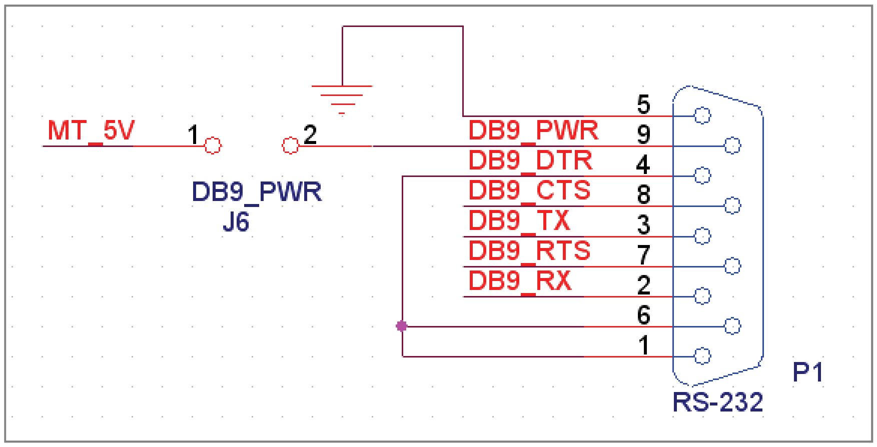


Figure 18. DB-9 Wiring Schematic.

The NADSBox hardware and firmware can detect the presence of DB-9 power, however it cannot calculate the supplied voltage level. This means the firmware cannot detect when low voltage conditions exist and provide card protection in the event of power loss. If voltage is being supplied from a Model 100/102/200 laptop across the DB-9 interface, it is the responsibility of the user to ensure card read/write operations are not performed when the laptop battery voltage is critically low.

Serial Cable Configuration

The RS-232 port provides fixed loopback of DTR and DSR signals and configurable wiring of RX/TX and RTS/CTS signals. This provides an easy method of using either a standard modem cable or a NULL modem cable. The RX/TX and RTS/CTS options are configured by specific placement of jumpers on header J3 (see Figures 18 and 19).

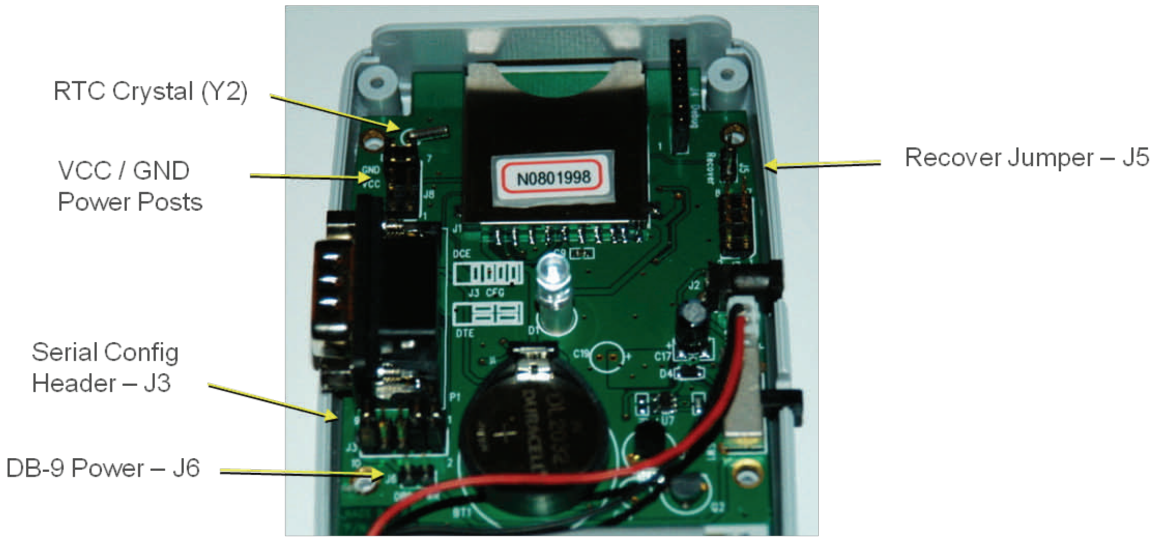


Figure 19. Configuration Jumper Locations.

Pins 1-4 set the wiring for RX and TX signals, and pins 5-10 set the wiring for RTS and CTS. Table 9 shows the options for jumper settings on J3. The factory configuration is DCE configuration with RTS and CTS in loopback mode.

Mode	Pin Connections	Connections
DCE	1 <—> 2, 3 <—> 4	Tx to DB-9(2), Rx to DB-9(3)
	5 <—> 7, 6 <—> 8	CTS to DB-9(7), RTS to DB-9(8)
DTE	1 <—> 3, 2 <—> 4	Tx to DB-9(3), Rx to DB-9(2)
	5 <—> 6, 7 <—> 8	CTS to DB-9(8), RTS to DB-9(7)
Loopback	9 <—> 10, 5-8 NC	CTS/RTS Loopback: DB-9(7) to DB-9(8)

Table 9. RS-232 Configuration Header (J3) Jumper Settings

A diagram of the J3 Configuration jumper is shown in Figure 20. Also shown are the jumper orientations for both the DCE and DTE configurations. The diagrams are also displayed on the silkscreen of the PCBA. Note that the CTS/RTS Loopback jumper should not be installed while jumpers are installed on pins 5-8.

NOTE: The jumper diagram detail on the PCB silkscreen for J3 is incorrect. The correct orientation for DCE and DTE are as shown in Figure 20.

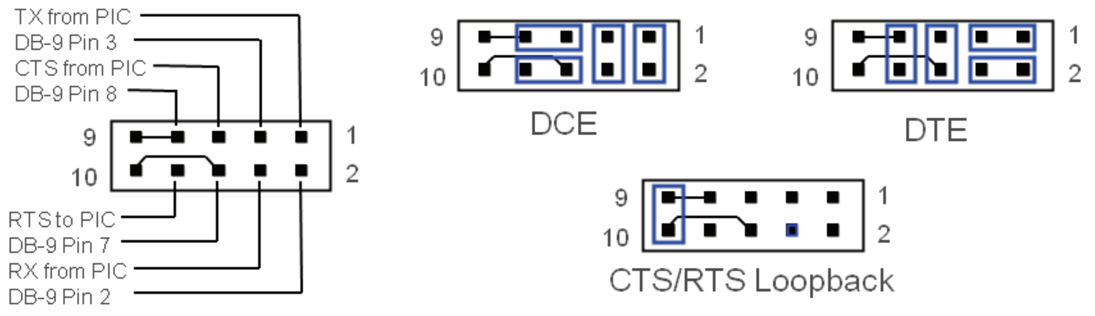


Figure 20. RS-232 Jumper (J3) Configuration.

The RS-232 connector is wired on the PCB to provide automatic DSR loopback. Figure 18 details the loopback configuration of the RS-232 port. As can be seen in the diagram, the NADSBox provides DSR / DTR loopback directly on the PCB with no configuration option.

Memory Card Support

As discussed in previous sections, the NADSBox firmware supports Mulitmedia Card (MMC), Secure Digital (SD) and SD High Capacity (SDHC) card formats. Version 1.0 has been tested with cards and configurations shown in Table 10. Note that with firmware version 1.0, SDHC cards larger than 4G are supported and should work properly, however the free space will be reported incorrectly. This is because the 32-bit math calculations are still byte-based instead of block-based and will overflow for anything larger than 4GB.

Type	Size	File System
MMC	16 M	FAT12, Floppy Style & Partition Table
	32M	FAT16, Floppy Style
	128M	FAT16, Floppy Style & Partition Table
	128M	FAT32, Partition Table
SD	1G	FAT32, Partition Table
	2G	FAT32, Partition Table
SDHC	4G	FAT32, Partition Table

Table 10. Memory Cards and Formats Tested

When a memory card is inserted, the NADSBox will delay approximately 0.25 seconds prior to communicating with the card to allow the voltage to stabilize. Then the firmware will begin communication and determine the type and size of the card inserted, validate operating voltages and determine the file system format. Anytime the card is being accessed, the status LED will flash to indicate the activity.

After the card initialization has completed and the file system has been determined, the firmware will search for the first free cluster on the active volume. This process may take a few seconds depending on the number of files and used clusters on the card. On a large card with limited free space, the operation may take longer than just a few seconds. This is a limitation of the FAT format—the format does not provide record-keeping for free clusters, and the firmware must search all FAT entries until a free cluster is found. While the firmware is initializing the card and searching for the first free cluster, no card activity from the TPDD protocol or command line interface will be processed.

After initialization is complete, the memory card and file system information can be viewed using the *'info'* command from the command line interface. For SD and SDHC cards with a "Card Lock" switch, the hardware and firmware will detect if the card is locked and disallow write operations to the card. As with any SD card reader, memory cards can be inserted and removed with the power active. However removal of a card during active read / write operations can cause data corruption and should be avoided.

Status LED

The status LED provides system operational status with various flash patterns. The LED is blue and requires 2.9V for full illumination. Under normal operation with full V_{CC} voltage, the LED will blink with full brightness. But if operating on low battery power, the LED brightness will begin to fade. Battery low voltage warning indicators will be generated well before the voltage drops below the operating range of the LED.

Table 11 lists the various status indications the firmware reports on the status LED along with the associated flash pattern. The "Idle, normal power" state flashing can be turned off using the 'config led=off' command and back on using 'config led=on'

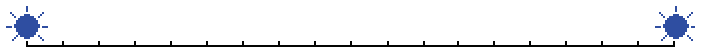
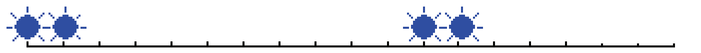
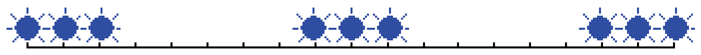
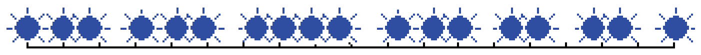
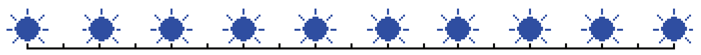
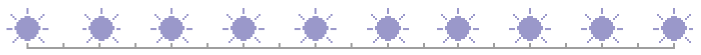
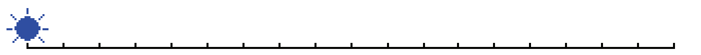
Mode	Pattern
Idle, normal power	
Low Power Warn	
Low Power Protect	
Card Access	
Flash Upgrade	
Flash EEPROM write	
Power On (no card)	

Table 11. Status LED flash modes and patterns.

External EEPROM

The NADSBox contains a 128K byte serial EEPROM that is used for storing program flash images during Flash Upgrade operations and for storing a flash recovery image. The EEPROM is divided into 4 sectors of 32K each (see Figure 15). The topmost sector, Sector 3, contains the flash recovery image and this sector is locked to prevent accidental erasure or modification.

The next sector down, sector 2, is a dual purpose sector. During normal operation, this sector is part of the 96K Internal File System (IntFS) used for storing frequently access files, or any file the user wants to save on the NADSBox without regard to a memory card installed. During flash update operations, if this sector is not empty then its contents are copied to the currently loaded SD/SDHC/MMC card and then used to temporarily hold the flash upgrade image. After the flash upgrade is completed, the original contents of he sector are reloaded from the memory card. The lower 64K (sectors 0 and 1) are used solely for IntFS operations

Real-Time-Clock and Battery

The Real-Time-Clock (RTC) in the NADSBox is implemented with a Dallas Semiconductor DS1302 chip. This device is a small, low-power device that uses a 32.768KHz crystal and a secondary .battery / super-Cap input for data retention when the primary power is turned off. The DS1302 contains a 2-digit year field and will keep track of leap-year days through the 21st century.

To accommodate the lack of a century field, the NADSBox firmware saves the century information entered using the *'date'* command in the processor's internal data EEPROM and uses that information when displaying and calculating dates. This will make the NADSBox operational into the 22nd century, and beyond (with the exception of leap year calculation).

The DS1302 RTC chip also contains 31 bytes of battery backed RAM that the NADSBox firmware uses for temporary storage of power and operational statistics prior to permanent storage. The battery backed ram collects usage information for a period of 1 week and then that data is added to the processor's internal data EEPROM. This method is used to reduce the number of write operations made to the internal EEPROM, thereby extending it's life and reliability.

For secondary RTC power, the NADSBox contains a CR-2032 coin cell holder and is shipped with a CR-2032 coin cell installed. Given that a typical 3V lithium coin cell has a capacity between 180 and 230mAH, and that the DS1302 draws roughly 500uA of power from the battery, the coin cell should last on the order of around 40 years according to the calculations. Assuming inefficiencies in the circuitry, the actual usable lifetime will probably be less than that, perhaps 20 years.

Caution areas if you open your NADSBox

At some point you may decide to open the NADSBox case to see what's "under the hood". Note that most of the interesting components are on the bottom side of the PCBA. If you decide to open the case, there are a couple of items which require special care as described below.

Real-Time-Clock (RTC) Crystal

The NADSBox real-time-clock uses a 32.768Khz crystal (Y2) to keep it's internal clock chain running. This crystal is a through-hole cylindrical component on the top of the PCBA near the SD Memory card connector and one of the mounting holes (refer to Figure 19). Be careful not to disturb this component—the oscillator circuit is very sensitive for proper timekeeping and the leads on the crystal are small and will break easily if bent too much, thus rending the RTC inoperable.

V_{CC} / GND Posts

Jumper J8 near the RTC crystal and the SD Memory card connector is a debug and expansion header (see Figure 19). This connector carries both V_{CC} (+3.3V) and GND signals, and these signals are adjacent on the header. The NADSBox is shipped with jumpers installed on these posts to prevent accidental shorting of the header pins.

However it may be necessary at some point to remove one of the jumpers to perform a Flash Recovery operation. If this operation is needed as some point or there is another reason to remove both the GND and V_{CC} post jumpers, be careful not to short the bare posts together as this may damage the onboard voltage regulators.

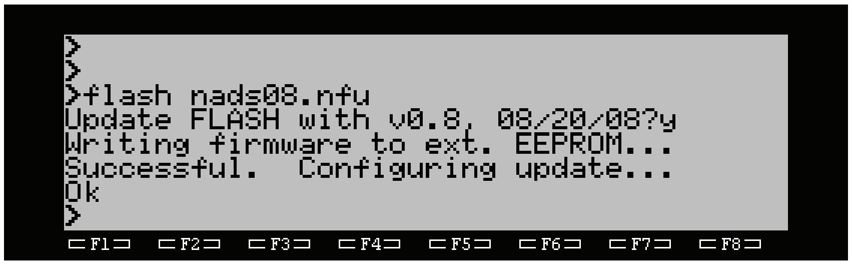
Upgrading The Firmware

The NADSBox firmware is upgradeable "in the field" through a simple command line utility. The *'flash'* command reads the new firmware from the inserted SD / SDHC / MMC card and performs the firmware upgrade automatically. The procedure is described here.

Upgrade Procedure

To upgrade the NADSBox firmware, follow the steps described below: NOTE: For the firmware upgrade process, either ensure DC power is applied to the NADSBox or that the batteries are not extremely weak. The PIC® processor can reprogram it's own flash program memory at Vcc voltages as low as 2V, however it takes much longer and may produce less than desirable effects if the voltage is too low.

1. Download the new firmware file from either the Manufacturer's or Vendor's website. Note that all released firmware files will have the format "nadsXY.nfu" where X is the major version number and Y is the minor version number, and the extension ".nfu" for "NADSBox Firmware Upgrade".
2. Use a modern SD card reader to copy the firmware file to an SD/SDHC/MMC card.
3. Insert the SD card into the NADSBox and ensure the unit is powered on.
4. Ensure the NADSBox is not in Low Power Warn or Low Power Protect mode (see the Status LED section).
5. On the Model 100/102/200, run TELCOM and set the STAT settings per the Command Line Interface section (98N1DNN, etc.). Alternately, use any PC with a serial port and data cable connection with a terminal program.
6. From TELCOM, go into TERM mode and press Enter.
7. Type the command "flash nadsXY.nfu" where XY is the particular version of firmware you downloaded (see Figure 21).
8. The *'flash'* command will read and validate the new firmware from the SD card, report the version number and release date, and ask for confirmation to continue.
9. Type "y" to perform the firmware upgrade. DO NOT TURN OFF THE NADSBox.
10. The Status LED will blink rapidly indicating the firmware is being upgraded. This will happen a couple of different times. Also the firmware will display two messages indicating the status of the upgrade process.
11. The upgrade should take 4-10 seconds based on the size of the new firmware.
12. Upon completion of the upgrade, the firmware will display "Ok". The upgrade is complete.



```
>
>
>flash nads08.nfu
Update FLASH with v0.8, 08/20/08?y
Writing firmware to ext. EEPROM...
Successful.  Configuring update...
Ok
>
```

Figure 21. NADSBox Firmware Upgrade.

Validating the Upgrade

After the NADSBox firmware has been upgraded, the Status LED should automatically flash indicating the new firmware is reading the contents of the SD card. If the Status LED remains on constantly (this should never happen under normal conditions), then there was a problem during the upgrade process and the firmware must be recovered using the steps described in the Program Flash Recovery section.

For positive validation of successful firmware upgrade, use the *'ver'* command described in the Command Line Interface section to display the new firmware version, and use the *'dir'* command as a sanity check that the NADSBox is still able to access the SD card.

Errors that can occur

During the firmware upgrade there are several errors that can occur and that may be reported by the NADSBox. These error responses and conditions are:

"Invalid file format"
Indicates that the file specified in the *'flash'* command is not a valid NADSBox Flash Upgrade file. Valid flash upgrade files contain a header with specific validation information and programming instructions that the *'flash'* command must detect before proceeding with the upgrade.

"Improper BIOS version" / "Improper HW Revision"
In the event future NADSBoxes must be shipped with a different hardware configuration or BIOS modification that is incompatible with previous NADSBoxes and separate NFU files must be provided, this message could be reported if the selected NFU file does not match the hardware and / or BIOS level of the NADSBox being upgraded.

"File checksum error"
The downloaded file has the proper format but has a data error somewhere in the file that is causing the checksum comparison to fail. Download the original file again or recopy to the SD card and try again.

"Invalid flash format"
The NFU file has the proper format and passed the checksum validation test, but one or more format errors exist that were detected during the 2nd phase validation process. Download the NFU file again and/or recopy to the SD card and try again.

"Error writing EEPROM"
After the firmware was validated, there was an error copying the data into the external EEPROM. This could be caused by weak batteries causing write operations to the EEPROM to fail. Use the 'stats' command to check that the Vcc voltage is 3.3v. If the problem persists, it is likely a hardware failure with the EEPROM.

Theory of Operation

The NADSBox firmware upgrade is made possible by the PIC®18LF series microcontroller's ability to erase and reprogram it's own flash program memory. To perform reliable upgrades, the firmware splits the operation into a couple of steps and by dividing the program memory space into sectors. The sectors (shown in Figure 22) consist of a 2K Boot Sector, a 32K Firmware Sector, and a 30K Application Sector.

During firmware upgrade operations, the data must be read from a FAT formatted volume using the code in the Firmware Sector. If the code in the Firmware Sector were to be written directly and the new firmware moved the location of active functions (as it most likely will), then upgrade would fail when an executing routine was overwritten with new instructions. Additionally, the program flash must first be erased, and then programmed, meaning the required instructions would not even exist any longer.

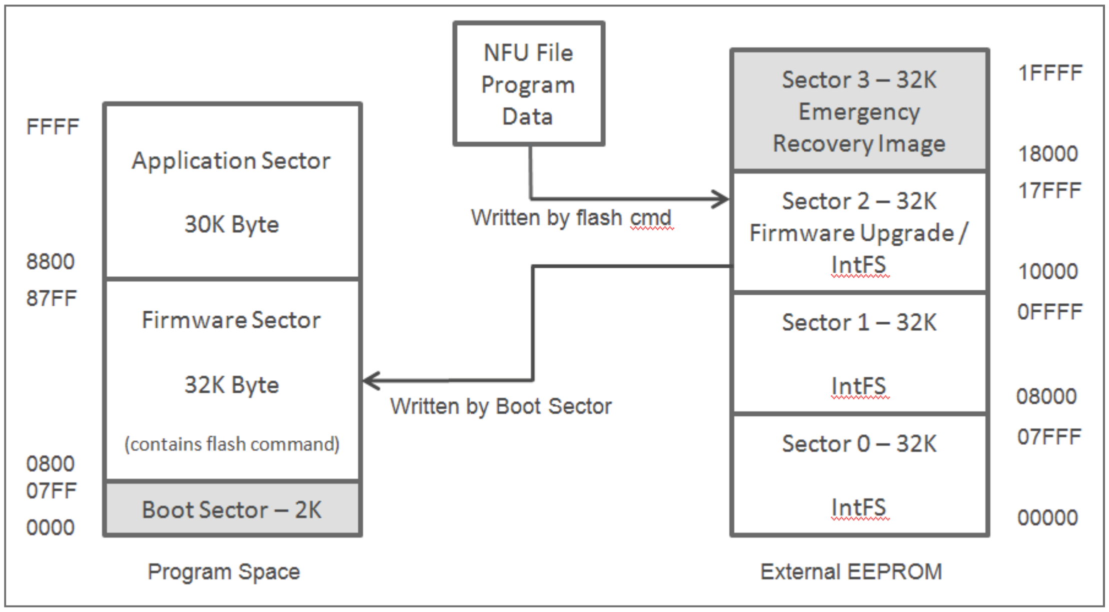


Figure 22. Firmware Upgrade Diagram. (Gray shading indicates write protected.)

To get around this issue, the upgrade firmware is split into two steps with each step located in different sectors, and an external 128K Byte EEPROM is used to temporarily store program data read from the SD card. During the first step, the *'flash'* command, which is located in the Firmware Sector, reads the new firmware from the SD card and writes it to Sector 2 of the External EEPROM. This allows all the data to be read from the card while known firmware is executing in the Firmware Sector.

After the Firmware Sector writes the new firmware to Sector 2 of the EEPROM, it writes a command to the processor's internal EEPROM to instruct the Boot Sector to perform the upgrade. Then it performs a software reset to force a reboot to the Boot Sector.

After the *'flash'* command in the Firmware sectors performs a software reset, the code in the Boot Sector will detect the flash upgrade request command in the processor's internal Data EEPROM and jump to it's firmware update routine. This routine is used for both firmware upgrades and emergency flash recovery operations. For the firmware upgrade operation, the routine will read the new firmware data from Sector 2 of the external EEPROM and write it to the Firmware Sector. Since the code for performing the erase and program operations in guaranteed to be located at a different address that the address being updated, there is no possibility of the processor executing unknown instructions.

After the Boot Sector has completed the programming of the Firmware Sector, it clears the command written to the internal data EEPROM written by the *'flash'* command to complete the upgrade process. Then the boot code jumps to the main entry point of the Firmware Sector to start executing the new code.

The Boot Sector is write protected at the factory and cannot be un-write protected without erasing the part completely using an external device programmer. This means the Boot Sector is always protected from corruption even if the code in the Firmware or Application sectors "crash" and start executing random instructions. This also means the code in the Boot Sector is not field upgradeable. For this reason, the code in the Boot Sector has been very well tested and contains only the minimal functionality needed for upgrade and recovery operations.

Figure 15 indicates that Sector 2 of the external EEPROM is used for Firmware Upgrade / IntFS. The Internal Filesystem (IntFS) will be added in a future version of the firmware and will be a 96K storage area for files to be saved on the NADSBox directly without the need for an SD card present. Since firmware upgrade operations do not (or at least shouldn't) occur that frequently, and since Sector 2 is only needed for 4-10 seconds, the firmware will also use Sector 2 as part of the IntFS. During firmware upgrades, the *'flash'* command will determine if files are stored in Sector 2 and automatically copy them to the SD card containing the new firmware. Then after the firmware upgrade operation is complete, the original Sector 2 contents will be restored with the IntFS data.

Program Flash Recovery

The NADSBox firmware upgrade routine has been tested extensively and should never fail. However sometimes things just go wrong and the unexpected happens. In the event firmware upgrade operation fails and leaves the NADSBox in a state where it is unable to read FAT formatted SD cards, the unit is equipped with a flash recover mechanism to restore the firmware to the factory default. This recovery operation is performed using a routine in the Boot Sector (see Upgrading The Firmware, Theory of Operation) by either copying a factory installed and write protected image from the external EEPROM or by reading and programming an image from the RS-232 port. Both methods are described here.

Recovery Jumper Operation

To recover the NADSBox firmware to a factory loaded default, a jumper must be installed on header J5, the "Recover" header (refer to Figure 19). The code in the Boot Sector will check the status of this jumper on power-up and automatically perform the recovery operation by copying the contents of Sector 3 in the external EEPROM to the Firmware Sector (see Figure 23).

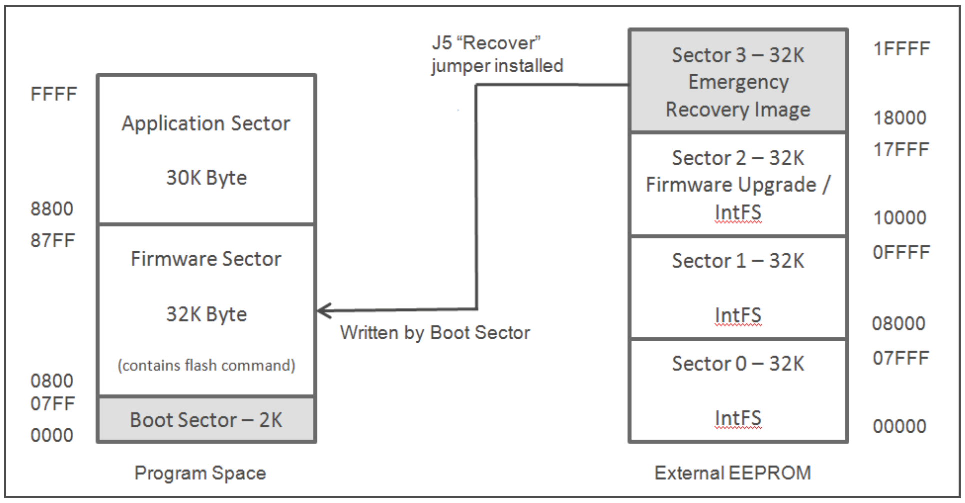


Figure 23. Flash Recovery Operation. (Gray shading indicates write protected.)

To perform the recovery operation, follow these steps:

1. Power the NADSBox off.
2. Open the NADSBox case by removing the 4 screws on the back cover. Note that two of the screws are located under the battery compartment cover.
3. Locate and remove the jumper from the Vcc posts on J8 (see Figure 20).
4. Install the jumper on header J5, the "Recover" header.
5. Turn the NADSBox power on and wait 2-3 seconds. The Status LED will flash.
6. When the Status LED stops flashing, the recovery operation is complete.
7. Remove the jumper from header J5 and return it to the Vcc posts on J8.
8. Carefully place the NADSBox cover back on and screw the two halves together.

Serial Recovery Operation

The Serial Recovery Operation is performed in a similar manner as the Recovery Jumper Operation except a host computer is connected to the RS-232 port with a serial boot program running. During the Recovery Jumper Operation described in the previous section, the firmware in the Boot Sector sends a *Boot Request* command across the RS-232 port and waits approximately 2 seconds for a *Boot Acknowledge* response from a host computer. If the *Boot Acknowledge* response is not detected, the Boot Sector continues with the Recovery Jumper Operation as described previously.

If the *Boot Acknowledge* response is detected, the firmware configures the recovery operation to receive all flash program data from the RS-232 port using the serial recovery protocol described below and shown in Figure 24. Since the NADSBox doesn't know the speed of the host PC, it sets no time limit for the reception of program data packets, only a retry count for each packet. Additionally, during the Serial Recovery Operation, the NADSBox reprograms the Firmware Sector in 64 byte increments. If the host computer connection is broken during this process, it will likely leave the data in the Firmware Sector in an unknown state and the operation will have to be started from the beginning.

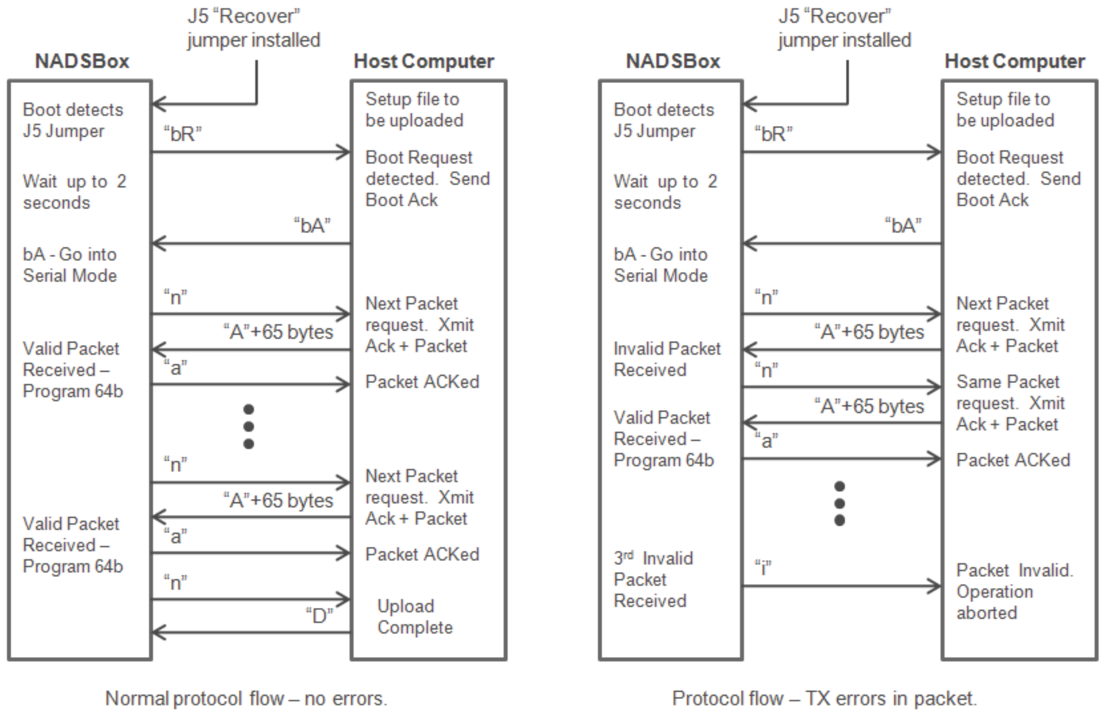

J5 "Recover" jumper installed
NADSBox
Host Computer
Boot detects J5 Jumper
Setup file to be uploaded
"bR"
Boot Request detected. Send Boot Ack
Wait up to 2 seconds
"bA"
bA - Go into Serial Mode
"n"
Next Packet request. Xmit Ack + Packet
"A"+65 bytes
Valid Packet Received – Program 64b
"a"
Packet ACKed
"n"
Next Packet request. Xmit Ack + Packet
"A"+65 bytes
Valid Packet Received – Program 64b
"a"
Packet ACKed
"n"
"D"
Upload Complete
Normal protocol flow – no errors.
J5 "Recover" jumper installed
NADSBox
Host Computer
Boot detects J5 Jumper
Setup file to be uploaded
"bR"
Boot Request detected. Send Boot Ack
Wait up to 2 seconds
"bA"
bA - Go into Serial Mode
"n"
Next Packet request. Xmit Ack + Packet
"A"+65 bytes
Invalid Packet Received
"n"
Same Packet request. Xmit Ack + Packet
"A"+65 bytes
Valid Packet Received – Program 64b
"a"
Packet ACKed
3rd Invalid Packet Received
"i"
Packet Invalid. Operation aborted
Protocol flow – TX errors in packet.

Figure 24. Serial Recovery Protocol.

The protocol starts with the NADSBox sending the characters "bR" to the host to initiate a *Boot Request.* When the host detects "bR", it acknowledges the request by sending "bA" characters. When the NADSBox receives the Boot Acknowledge, it will begin requesting program data packets by sending the character 'n". The host acknowledges the request by sending "A" followed by 65 bytes of data.

After receiving the requested program data packet, the NADSBox boot loader will validate the integrity of the received packet to ensure there were no transmission errors and to ensure the data is properly formatted NADSBox Flash Upgrade information. If the packet if validated successfully, the firmware will acknowledge receipt of the packet by sending the host an "a" character. Then the firmware will erase and reprogram the next 64 bytes of program flash in the Firmware Sector.

If the packet received is invalid, the firmware will send an "n" instead of an "a" character to indicate the packet received was invalid and should be retransmitted. The firmware will not program packets that have been identified as invalid. The host must respond by resending the previously transmitted packet. The NADSBox serial loader will retry each packet a maximum of 4 times. If a valid packet cannot be received on the 4th attempt, the firmware will send an "i" character, abort the flash upgrade operation and jump to an infinite loop.

The serial boot loader will accept a maximum of 512 packets or a total of 32K bytes of program information to be programmed in to the Firmware Sector. If the host computer reaches the end of the NFU file, however, and fewer than 512 packets have been transmitted, the host can indicate the firmware upload has completed by sending a "D" character in response to the next packet ("n") request. The 1st 64 bytes of the NFU file contain header information that is not written to the NADSBox across the serial interface.

The Serial Recovery Option has no provisions for validating the uploaded firmware against the NADSBox's BIOS level and hardware revision or executing other specific upgrade instructions contained in the NFU file like is possible using the *'flash'* command. For this reason, this option should be used as a last resort method of bringing a "dead" NADSBox (should never happen) back to life.

The listing in Figure 25 shows an sample NADSBox Serial Loader application. This program uses direct memory accesses of the laptop's file directory to locate the NADSBox binary code, so it is specific to the Model 100 and Model 102 laptops. To make this program compatible with other models, the absolute addresses in lines 40 and 100 need to be modified. Since the program is written in BASIC, it executes rather slowly and requires a ".CO" formatted version of the NFU file to already exist on the Model 100 / 102. The program was written primarily for testing the NADSBox boot loader firmware.

A Windows® serial boot loader program will also be developed and documented herein at some point.

```
5 MAXFILES=2
10 PRINT "NADSBox Flash Loader"
20 A$="":INPUT"File to upload";A$
30 IF A$="" THENPRINT"Aborted": GOTO900
32 FORX=1TOLEN(A$):A=ASC(MID$(A$,X,1))
34 IF A>=97AND A<=122 THEN MID$(A$,X,1)=CHR$(AAND223)
40 NEXT:D=63931
50 A=PEEK(D)+PEEK(D+1)*256:D=D+2
60 N$="":FORX=1TO6:IF PEEK(D)<>32 THEN N$=N$+CHR$(PEEK(D))
70 D=D+1:NEXT:N$=N$+".":FOR X=1 TO 2
80 IF PEEK(D)<>32 THEN N$=N$+CHR$(PEEK(D))
90 D=D+1:NEXT:IF A$=N$ THEN 200
100 IF D>64137 THEN 130
110 IF PEEK(D)=0 THEN 130
120 D=D+1:GOTO 50
130 PRINT"File not found":GOTO 20
200 PRINT"Checking file format...":GOTO 700
210 P$="COM:98N1D":B=A:E=A+L:C=INT((L-64)/65)
220 OPENP$FORINPUTAS1:OPENP$FOROUTPUTAS2
230 R$=INPUT$(1,#1):PRINTR$:IFR$<>"R"THEN230
240 PRINT#2,"bA";:PRINT"bA"
250 R$=INPUT$(1,#1):IFR$<>"n"THEN500
255 IF B>ETHENPRINT#2,"D";:GOTO800
260 PRINT#2,"A";:PRINTC;:FORX=0TO64:IFB+X>ETHEND=255ELSED=PEEK(B+X)
270 PRINT#2,CHR$(D);:NEXT:R$=INPUT$(1,#1)
280 IFR$="n"THEN260
285 IFR$="i"THEN850
290 IFR$<>"a"THEN500
300 B=B+65:C=C-1:IFC=0THEN800
310 GOTO 250
500 PRINT"Protocol Error!":GOTO 900
700 A=A+2:L=PEEK(A)+PEEK(A+1)*256:A=A+4
710 S$=CHR$(PEEK(A))+CHR$(PEEK(A+1))
720 IF S$<>"NF"THEN850
730 V$=CHR$(PEEK(A+6)+48)+"."+CHR$(PEEK(A+7)+48)
740 D$="":FORX=0TO7:D$=D$+CHR$(PEEK(A+8+X)):NEXT
745 PRINT"File length = ";L
750 PRINT"Upload FW v";V$;" dated ";D$;
760 INPUTQ$:IF Q$="n"ORQ$="N"THEN 20
770 IFQ$<>"y"ANDQ$<>"Y"THEN750
775 A=A+64
780 PRINT"Waiting for request from NADSBox..."
790 GOTO 210
800 PRINT#2,"D";:PRINT:PRINT"Complete!"
810 GOTO 900
850 PRINT:PRINT"Invalid FLASH file!"
900 A=0
```

Figure 25. NADSBox Serial Loader Program.