Repurposing KwikByte FRC Driver Station

Carlos Gross Jones

February 15, 2017

Abstract

1 Introduction

2 Connection & Communication

Communication with the DS9260 takes place primarily over the DE9 "competition port". Despite appearances, this is not a plain serial port. While many of the pin functions are not yet know, basic RS232 serial capability is present on the normal pins: 2 (RX), 3 (TX), and 5 (ground). Therefore, a custom null-modem adapter was made which connects only those three pins. (The "USB Adapter Clip" from KwikByte was likely a similar connector, plus a FT232 or similar chip for USB operation.)

Opening the serial port (115200, 8, N, 1) presents a banner and login prompt:

```
 --------------------
-                    -
- KwikByte DS9260    -
-                    -
- www.kwikbyte.com   -
-                    -
--------------------
DS9260 login:
```

While the root account is password protected, trial and error revealed a user account called "default" that did not require a password. This allowed exploration of much of the filesystem. It appears to be a fairly standard embedded Linux system with BusyBox. The application software binaries are in /ds60x/bin, consisting of nine files: lcd, mgr_joysx, recv_pc, send_fms, send_robot, manager, recv_fms, recv_robot, and send_pc.

For firmware updates and other pre-boot functionality, the normal boot process can be interrupted and a "utility loader" sent by serial as described in App. A. This has basic facilities for memory read and write, as well as interaction with the SPI Flash memory.

3 Dissection of Factory Image

The first software item investigated was the image "raw_otb.bin" provided by KwikByte (also mirrored on carlosgj.org for posterity).

This file is not a first-level bootloader, but is instead called from the kernel loader. As such, it is expected to contain at least the Linux kernel and an initrd. In fact, the most recognizable thing in the file is the Linux boot parameters at offset 0x58:

```
console=ttyS0,115200 root=/dev/ram rw initrd=0x22400000,851719 mem=64M
```

The purpose of the following data is unknown. However, an educated guess would be that a compressed Linux kernel exists in the file. Thus, the gzip "magic number", 0x1F8B, plus the expected compression method, 0x08, should be found (see the gzip standard for details). A search for 0x1F8B08 showed the first occurrence at offset 0x44F8. Metadata present in the gzip header reveals that the data was zipped on Fri, 03 Oct 2008 at 23:56:34 GMT on a Unix system. A partial copy of the raw_otb.bin file from 0x44F8 to the end can indeed be unzipped, resulting in what appears to be a Linux kernel based on literal strings in the binary. In fact, one such string offers potentially valuable information:

```
Linux version 2.6.23DS60v1.0 (root@kbdev-laptop13) (gcc version 3.4.2) #7
Fri Oct 3 16:56:31 MST 2008
```

During the unzipping process, gzip noted that "trailing garbage" was ignored, indicating that the compressed archive beginning at 0x44F8 does not in fact extend to the end of the file. As the gzip standard specifies, the last four bytes of a member contain the size of the uncompressed data. Noting that the size of the unzipped kernel is 0x2BE6B0, this value was found in raw_otb.bin at offset 0x160421, indicating the end of the first gzip archive. Shortly afterwards, at offset 0x160494, another occurrance of 0x1F8B08 is found, indicating the start of another archive. The metadata shows that it was zipped on Mon, 03 Nov 2008 at 20:23:39 GMT, with maximal compression, on a Unix system, and additionally that the original filename was "initrd_img". Copying 0x160494 onwards results in another valid gzip, which successfully uncompresses into a filesystem.

As a first step towards repurposing the DS9260, root access was needed to more thoroughly explore the system. First, on the host computer (running Debian Jessie), the gunzipped initrd was mounted. Then, the root entry in /etc/shadow was edited to remove the password, resulting in:

```
root::10933:0:99999:7:::
bin:*:10933:0:99999:7:::
daemon:*:10933:0:99999:7:::
adm:*:10933:0:99999:7:::
lp:*:10933:0:99999:7:::
sync:*:10933:0:99999:7:::
shutdown:*:10933:0:99999:7:::
halt:*:10933:0:99999:7:::
uucp:*:10933:0:99999:7:::
operator:*:10933:0:99999:7:::
nobody:*:10933:0:99999:7:::
default::10933:0:99999:7:::
```

This "rooted" filesystem image was then gzipped, making sure that the gzip header and parameters matched the original. Knowing that the compressed initrd begins at 0x160494 of raw_otb.bin, and that raw_otb.bin is written to Flash at 0x42000 (per App. A), the rooted gzip was written to SPI at 0x1A2494 (0x42000+0x160494) using the Util Loader. This resulted in a working system with a root account which does not require a password. Surprisingly, although the rooted initrd was slightly larger than the original, the "initrd=0x22400000,851719" boot argument did not have to be changed.

A Utility Loader & Firmware Update Instructions

FIRST DRIVER STATION UTILITY LOADER RE-IMAGE INSTRUCTIONS

1 Introduction

This document describes steps to load the Driver Station (DS) v1.0 Utility Loader (UL). The UL can be used to re-image the DS, perform low-level operations on the board, or load alternate operating systems or user programs.

*1.1 *** Warnings ****

The boot loader(s) are write-protected. Do not unlock these sections. Doing so may render the unit unusable.

Do not overwrite the released image(s) or otherwise modify these sections with your own code. Doing so may render you ineligible to compete.

Be careful when using some of the UL commands. For example, you should not set a system-defined input pin as an output as this could damage the Driver Station.

The steps listed here do not violate these warnings. To be safe, just follow the instructions.

1.2 Equipment

You need:

1) A means of communicating with the DS at a low-level. This document uses the DS USB Adapter Clip with supplied USB extension cable.
2) A PC. The host OS can be Windows® or Linux. Other OS may work, but have not been tested.

1.3 Software

You need:

1) A host PC terminal emulator program like HyperTerm, minicom, Kermit, etc.
2) The DS UL binary image.
   ftp://kwikbyte.com/pub/DS/binary/utilLoader.bin
   or
   http://www.kwikbyte.com/driverstation/binary/utilLoader.bin

3) The original factory firmware.
   ftp://kwikbyte.com/pub/DS/binary/raw_otb.bin
   or
   http://www.kwikbyte.com/driverstation/binary/raw_otb.bin

1.4 Support

Please read the instructions carefully. If you have questions or helpful comments, please send them to driverstation@kwikbyte.com.

2 Re-image Instructions

2.1 Load the UL

The DS firmware provides a means of updating the system before the OS is loaded. During boot, the DS waits for two characters (must be caps) from the host: K B.

1) Insert the USB extension cable into a free USB port on the PC.

2) Insert the DS USB Adapter Clip on the end of the USB extension cable. Notice the little green LEDs flash on the Adapter Clip.

3) When using Windows®, a driver may be required to use the clip. In that case, see http://www.ftdichip.com/Drivers/VCP.htm. The device used is FT232R. Select your OS, understand the comments listed on the web page, download and install the driver.

4) The OS detects the Adapter Clip and reports that it is ready for use.

5) Start the terminal emulator: e.g., HyperTerm (Start -> All Programs -> Accessories -> Communications -> Hyperterm).

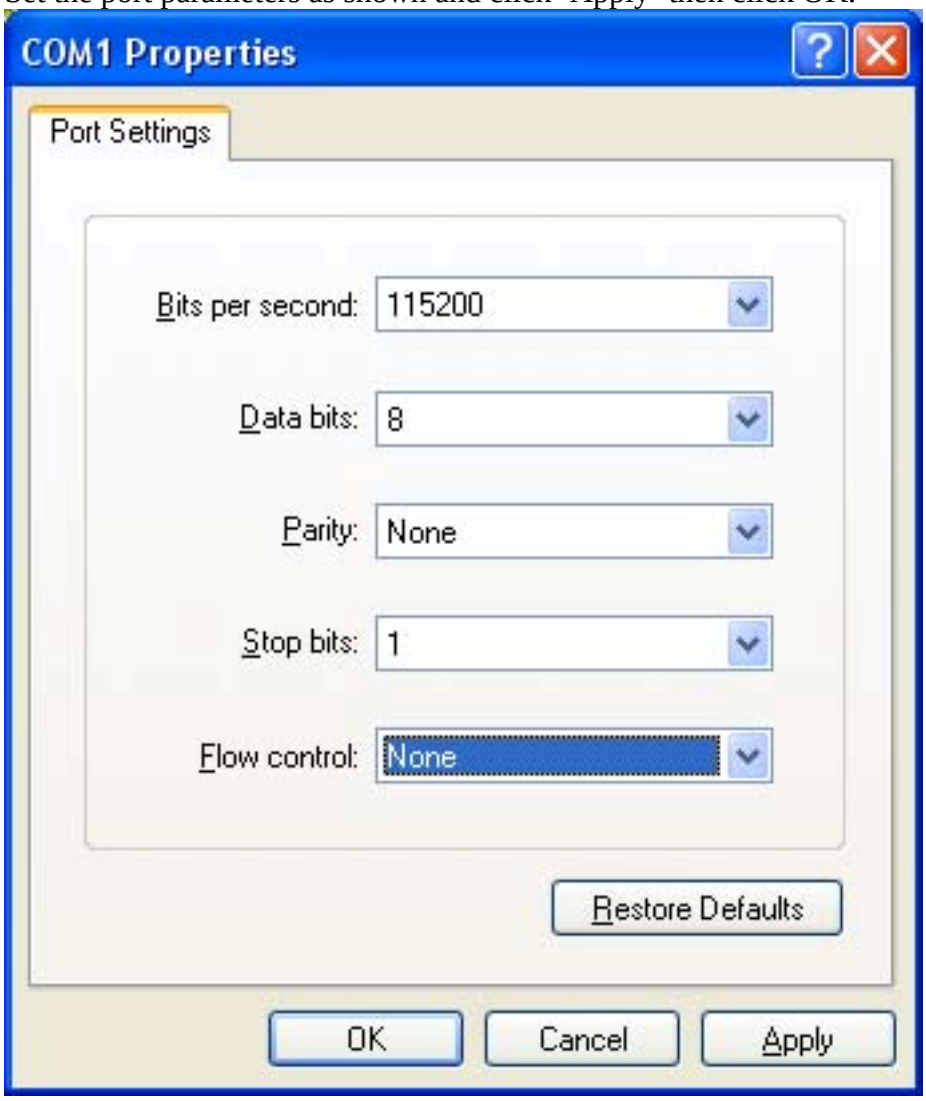
6) If asked, you do not have to make HyperTerm the default telnet program.

7) Provide a meaningful name in the Connection Description and click OK.

8) Depending on your computer configuration, the Adapter Clip may be recognized as a different serial port than the one shown. Trial and error will tell which port is correct. On many PCs, the last serial channel listed (e.g., COM4) is correct.

Select the port and click OK.

9) Set the port parameters as shown and click ‘Apply’ then click OK.

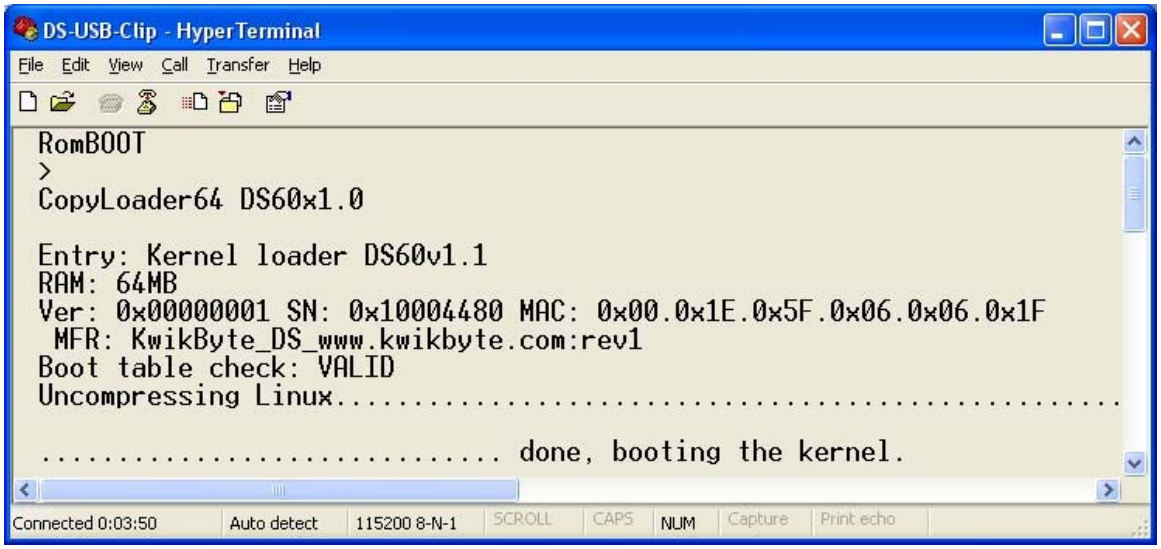


10) Now, you see an empty window.

11) Plug the Adapter Clip into the Driver Station Competition Port.

12) Apply power to the Driver Station by plugging-in the Driver Station power cord.

13) Now, you will see the output from the Driver Station. If not, you chose the wrong serial port. Close HyperTerm and go back to Step #5.

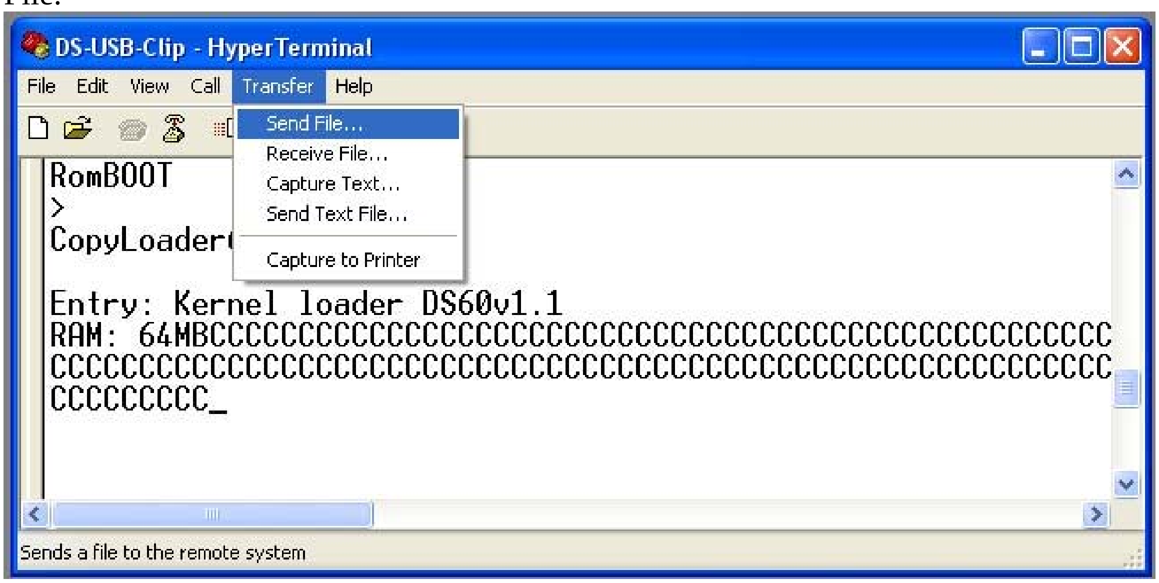


14) Remove power from the Driver Station by unplugging the power cord. Look at the first few lines of output. This is showing the Driver Station loading the Linux kernel and beginning to boot the system! In the next step we will trigger a secret hook in the "Kernel loader" by pressing some keys at the right moment. Timing is important because the kernel loader only looks for these characters for a short period before booting the normal system.

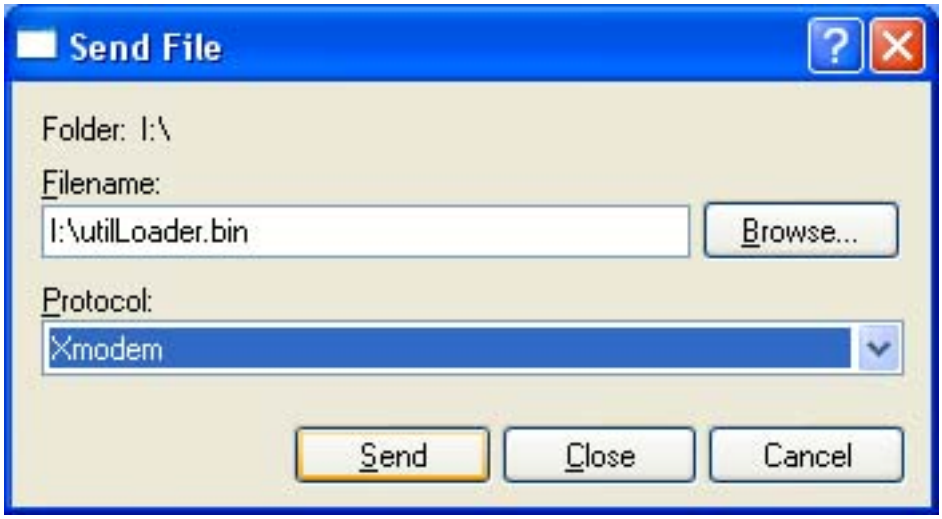
15) Get ready to type the two, capital case characters K B right after you see the "RAM: 64MB" output on the screen. On the Driver Station, hold down all three white pushbuttons (up, down, and select) – and keep them held down during this step. Apply power to the Driver Station by plugging-in the power cord. As soon as the "RAM:" line is output, type the two keys.

16) If you executed the previous step correctly, you now see the Driver Station sending 'C' characters at about one per second. These characters continue until you send the UL image to the Driver Station.

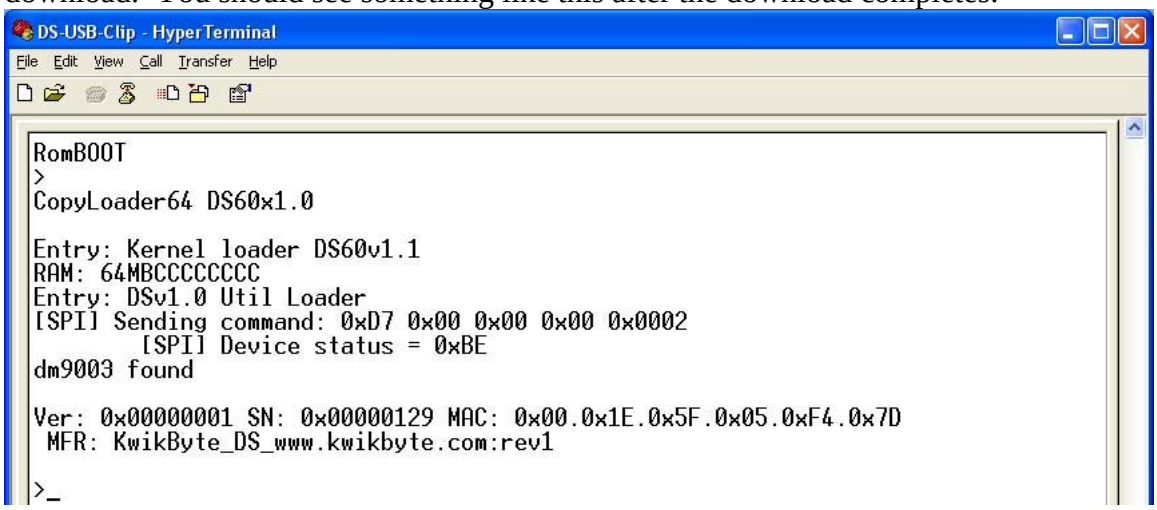
17) Send the utilLoader.bin file to the Driver Station by selecting Transfer -> Send File.



18) Select the file utilLoader.bin and set the transfer as Xmodem – this is **not** the same as 1K Xmodem. Then, click Send.



19) Watch the transfer progress. The first version of the UL takes about 12 seconds to download. You should see something like this after the download completes:



NOTE: The UL was not written specifically for network capability. Some Driver Stations correctly identify the dm9003 net chip. The network function will be used in a different document – when we boot a different Linux kernel and mount a larger file system on a USB stick. For now, we will use serial and just understand that Ethernet is a lot faster than serial!

The LCD display may also change when running the UL. This is normal.

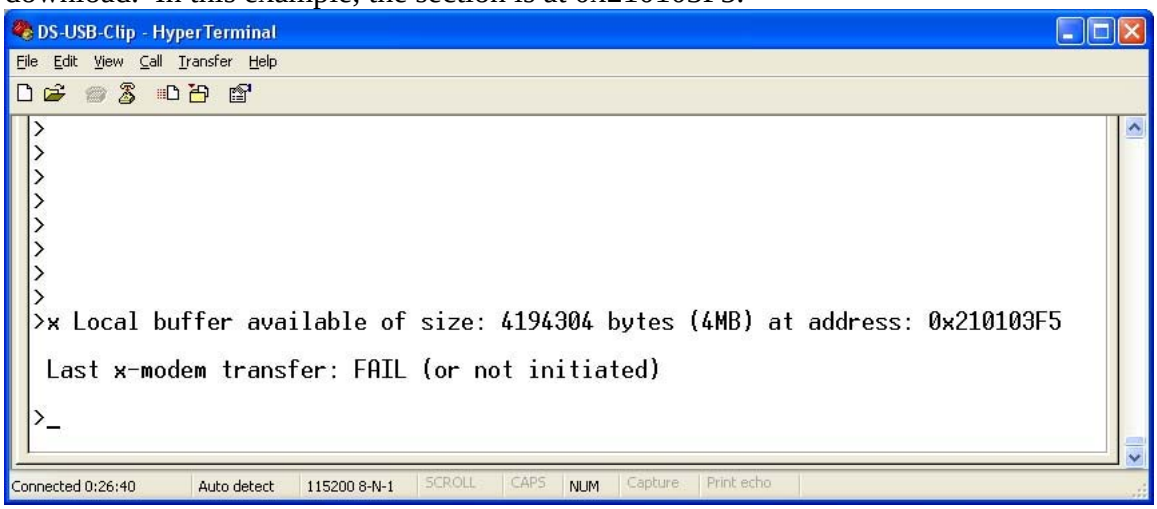
20) Some *very* brief help is available with ? Enter.

NOTE: The UL will repeat the last command if you type only Enter without entering a new command.

Be careful! Some of these commands can damage the Driver Station if used

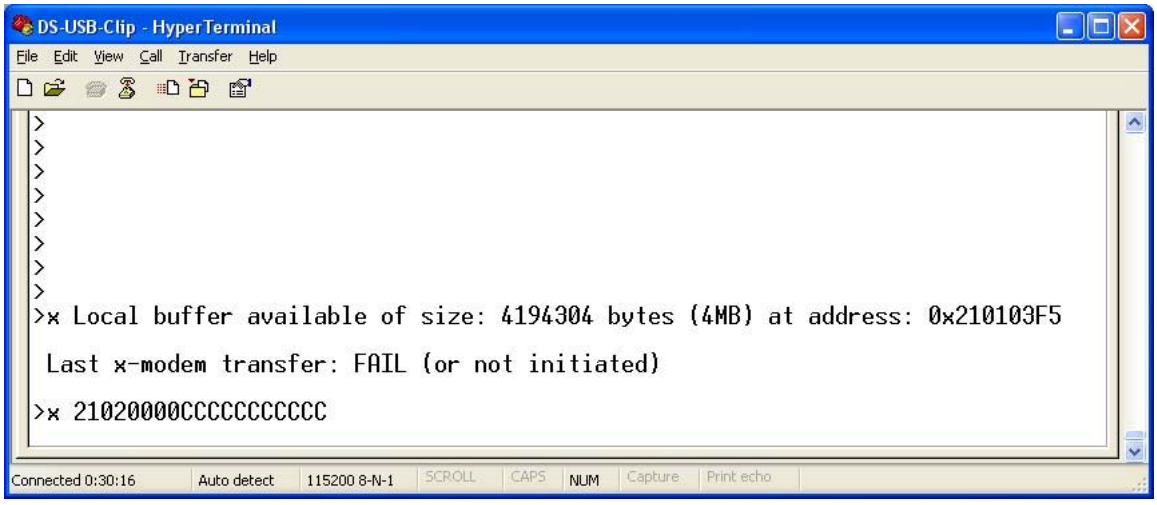
incorrectly.

21) Type x Enter. This reports the location of a free memory section we can use for download. In this example, the section is at 0x210103F5.



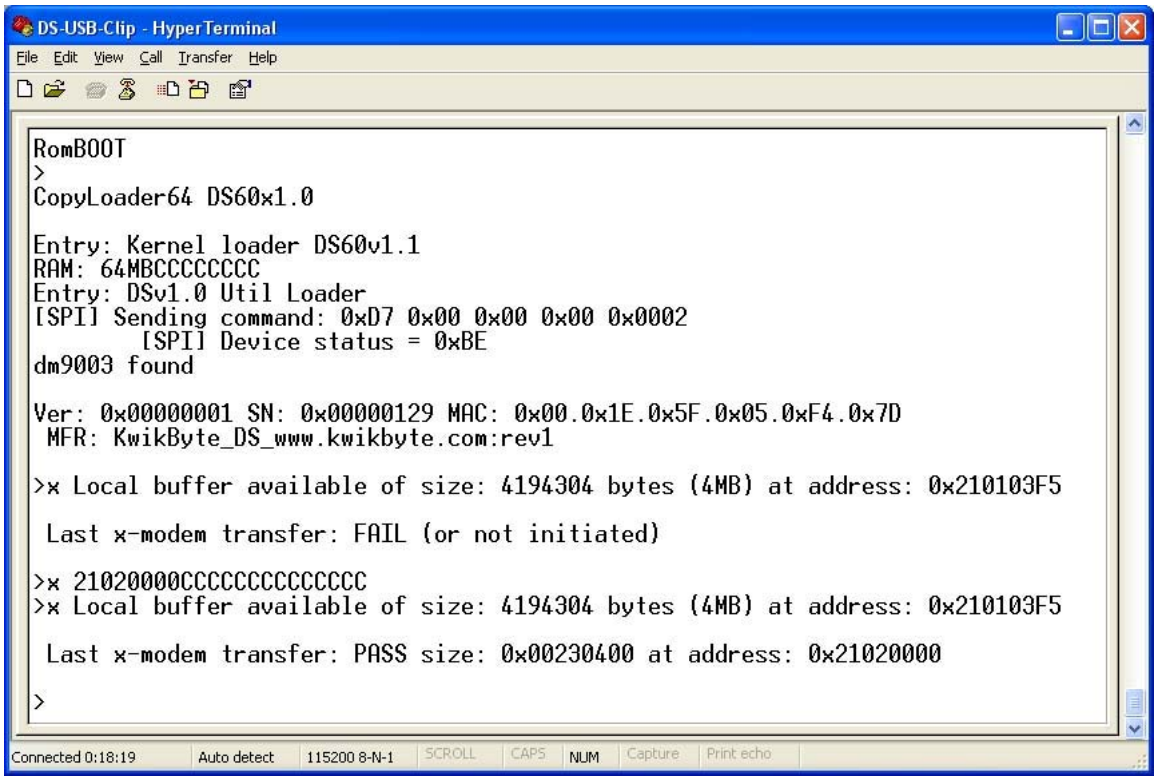
22) Let's initiate a download to receive the factory Driver Station Firmware. We will round-up the address provided by the buffer to a nice even number – 0x21020000. This works fine as long as the image to be received is less than the space allocated (4MB). Type x 21020000 Enter.

23) Now you see more 'C' characters indicating that the Driver Station is waiting for another Xmodem transfer.



24) As before, execute a transfer from the PC using Transfer -> Send File. Select the original raw_otb.bin Driver Station firmware for Xmodem download – just like we did earlier. Click Send and watch the progress. This will take a long time depending on the image size.

25) The Driver Station now has the firmware image in RAM. We need to program it to non-volatile memory (NVM). The NVM used on the Driver Station is serial flash. Type x Enter to get the size of the last transfer from the Driver Station's view.



26) From this example, the transfer size is 0x230400. Let's program flash with a command of the format "spi_write <flash destination> <source address> <size>". In this case,

spi_write 42000 21020000 230400 Enter

You should see lots of "[SPI]" messages scrolling during the programming operation. This command also takes a *long* time – depending on image size. Of course, there are faster ways to do this but let's start with this method. When the program operation is complete, the Driver Station reports "Buffer Written".

The re-image is complete. Reset the Driver Station by entering the command reset Enter.

3 Revisions

22DEC2008 Creation

B Kernel Loader Update Instructions

FIRST DRIVER STATION KERNEL LOADER UPDATE INSTRUCTIONS

1 Introduction

This document describes steps to update the Driver Station (DS) v1.0 Kernel Loader (KL). The KL is used to boot the operating system (Linux) on the DS or load alternate operating systems or user programs.

The purpose is to open the Driver Station for use as an embedded software development/learning tool.

*1.1 *** Warnings ****

You must follow the instructions exactly!
Potential risk: you could lock-up the Driver Station – making it unusable.
If you are not comfortable with the risk, do not perform this procedure.

The loader does not impact the applications running on the Driver Station during normal operation.

1.2 Documentation

In addition to this document, you need a copy of the Utility Loader document:
http://www.kwikbyte.com/driverstation/doc/DS_utility_loader.pdf

1.3 Equipment

You need:

1) A means of communicating with the DS at a low-level. This document uses the DS USB Adapter Clip with supplied USB extension cable.
2) A PC. The host OS can be Windows® or Linux. Other OS may work, but have not been tested.

1.4 Software

You need:

1) A host PC terminal emulator program like HyperTerm, minicom, Kermit, etc.
2) The DS UL binary image.
   ftp://kwikbyte.com/pub/DS/binary/utilLoader.bin
   or
   http://www.kwikbyte.com/driverstation/binary/utilLoader.bin

3) The new kernel loader (current version is v1.2).
   http://www.kwikbyte.com/driverstation/binary/kernelLoaderv12.bin

4) The altLoader program.
   http://www.kwikbyte.com/driverstation/binary/altLoader.bin

1.5 Support

Please read the instructions carefully. If you have questions or helpful comments, please send them to driverstation@kwikbyte.com.

2 Kernel Loader Update Instructions

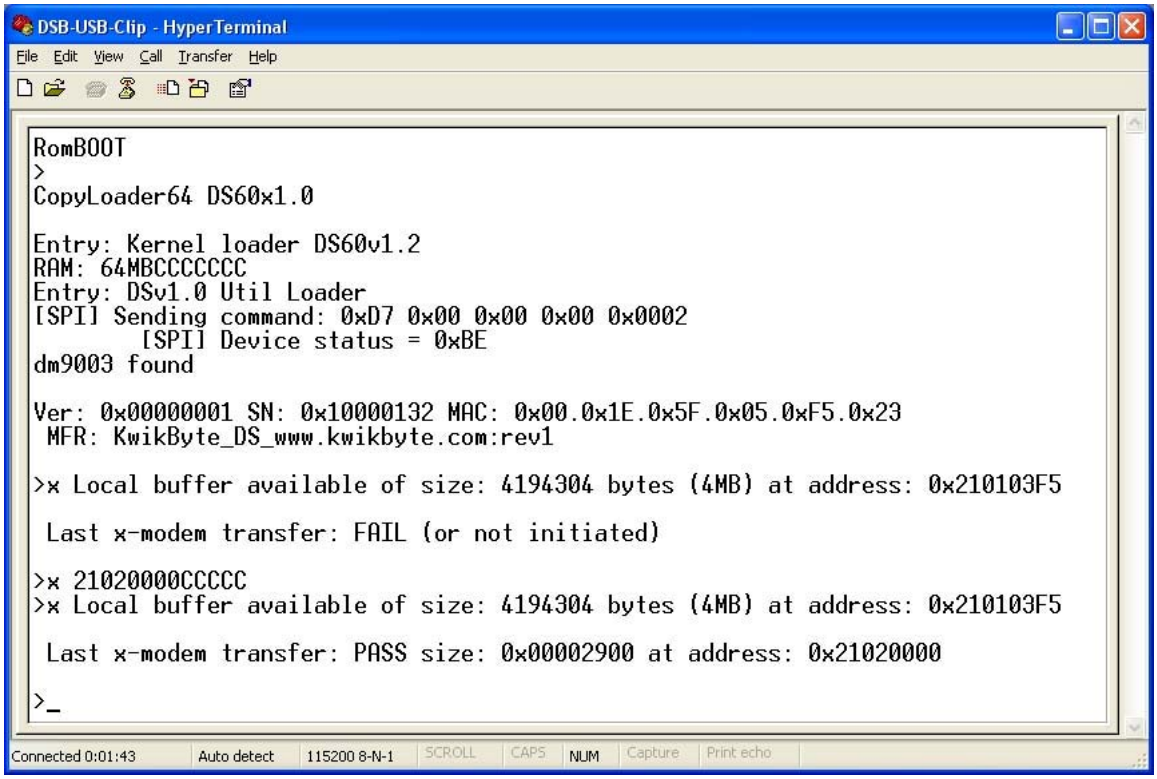
2.1 Load the UL

Follow the instructions from the UL document exactly (http://www.kwikbyte.com/driverstation/binary/utilLoader.bin) through step #23. Finish step #23. Stop. Do not perform step #24. You can now close that document because we won't use it any more in this process.

2.2 Update the KL

1) As performed previously, execute a transfer from the PC using Transfer -> Send File. Select the kernelLoaderv12.bin file for Xmodem download – just like we did earlier. Click Send and watch the progress. This only takes a few seconds because the file size is small.

2) The Driver Station now has the kernelLoaderv12.bin image in RAM. We need to program it to non-volatile memory (NVM). The NVM used on the Driver Station is serial flash. Type x Enter to get the size of the last transfer from the Driver Station's view.



3) This step is critical!!! Pay very close attention to the commands and make sure you type them exactly as listed.

   From this example, the transfer size is 0x2900. Unlock the boot loader with the following two commands:

   spi_erpx Enter
   spi_wrpx 0 Enter

   Let's program flash with a command of the format "spi_write <flash destination> <source address> <size>". In this case,

   spi_write 1080 21020000 2900 Enter

   Restore the lock on the boot loader with the following commands:

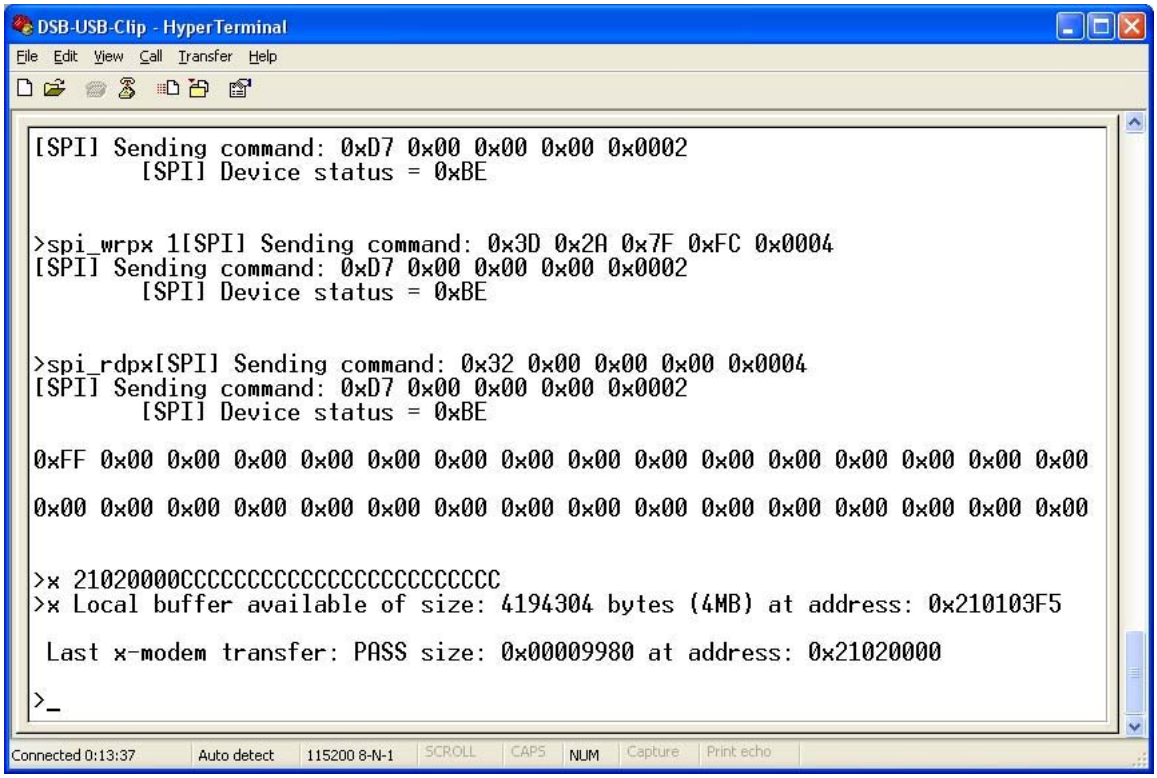
   spi_erpx Enter
   spi_wrpx 1 Enter

   You should see lots of "[SPI]" messages scrolling during these operations.

4) The kernel loader is now updated. Let's download the alternate OS loader while we are here. The rest of the steps are not as important as step #3 because they are considered recoverable. Initiate a transfer, again, to the same buffer address:

x 21020000 Enter

5) Notice the 'C' characters again and send the altLoader.bin file for Xmodem transfer. This takes about 6 seconds to transfer. Verify the transfer size by typing x Enter.



The transfer size is shown as 0x9980.

6) Now, let's program this image at the beginning of the last 256-page block of flash:

spi_write 7FE000 21020000 9980 Enter

7) Again, many [SPI] type message scroll by during the programming operation. After about 20 seconds, it completes with "Buffer written" message.

We're done with this process! Verify everything runs as normal by entering the command reset Enter.

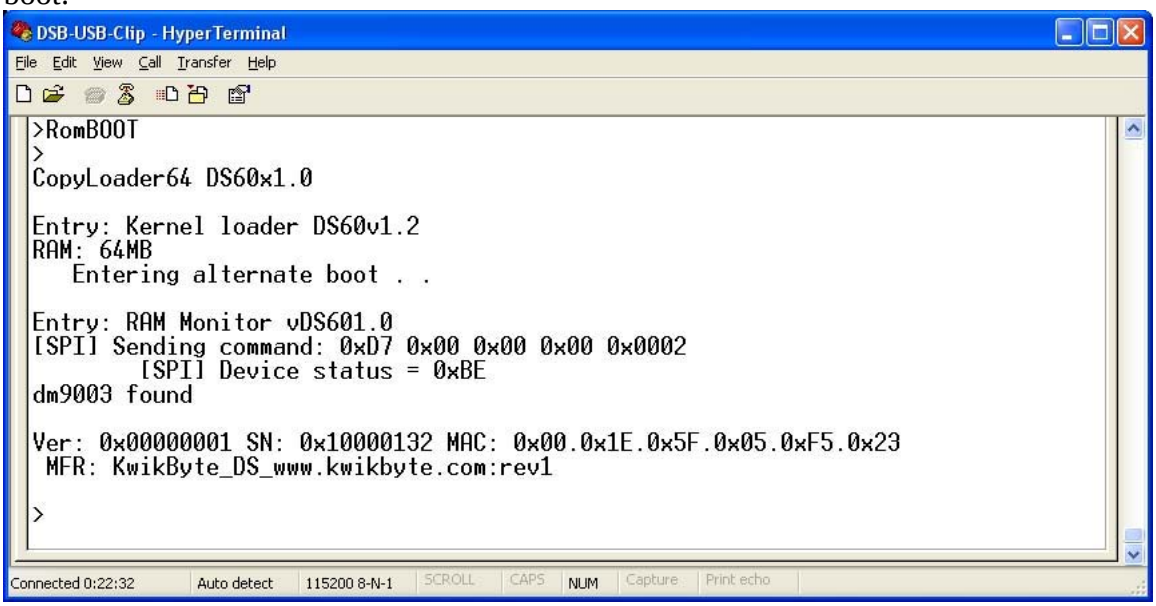
The only observable differences are

1) The boot-up logo is really strange! We will fix that (easily) in the next installment by loading your custom image.
2) The version reported by the kernel loader is now 1.2.

2.3 Results

The new kernel loader obtains the boot-up logo information from a separate flash location. This makes it very easy to modify the logo without changing the extra-important boot sections.

The new kernel loader also accepts a new "special key" sequence. If you apply power to the DS while holding the up-arrow and SELECT buttons, the DS will perform an alternate boot:



This feature will be used to boot another version of Linux.

3 Revisions

16JAN2009 Creation

C Logo Update Instructions

FIRST DRIVER STATION KERNEL LOADER UPDATE INSTRUCTIONS

1 Introduction

This document describes steps to change the boot-up logo on the Driver Station (DS) v1.0. The updated kernel loader (KL) is required in order to perform this procedure. The KL must be version 1.2 or later.

By this time, you should be familiar with how-to connect to the DS using the low-level serial clip. This document assumes you already have the terminal window up and running with a live connection to the DS.

1.1 Documentation

It is assumed the steps in the KL document have been completed already:
http://www.kwikbyte.com/driverstation/doc/DS_kernelLoaderv12.pdf

1.2 Equipment

You need:

1) A means of communicating with the DS at a low-level. This document uses the DS USB Adapter Clip with supplied USB extension cable.

2) A PC. The host OS can be Windows® or Linux. Other OS may work, but have not been tested.

1.3 Software

You need:

1) A host PC terminal emulator program like HyperTerm, minicom, Kermit, etc.

2) The bmpConversion utility program:
http://www.kwikbyte.com/driverstation/binary/bmpToLogo.bin

3) A Windows® formatted bitmap image for the new logo. This must be monochrome image of size 128x64. We will call this image newlogo.bmp.

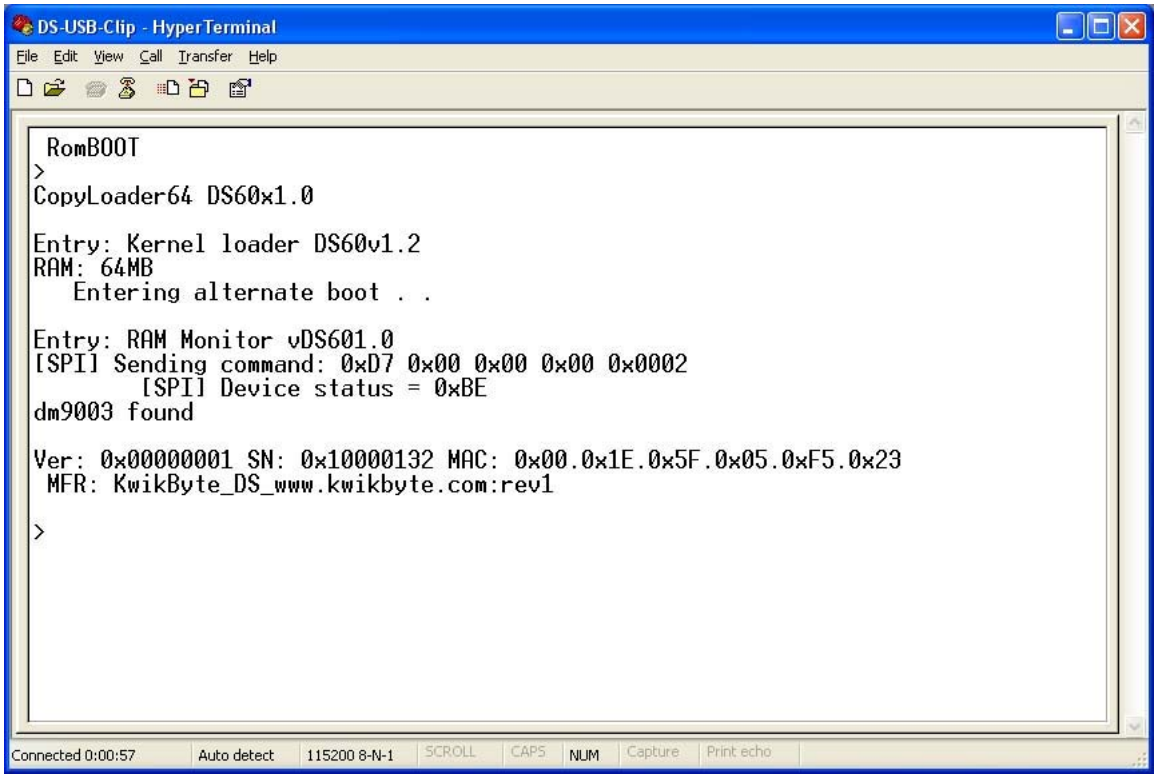
1.4 Support

Please read the instructions carefully. If you have questions or helpful comments, please send them to driverstation@kwikbyte.com.

2 Logo Update Instructions

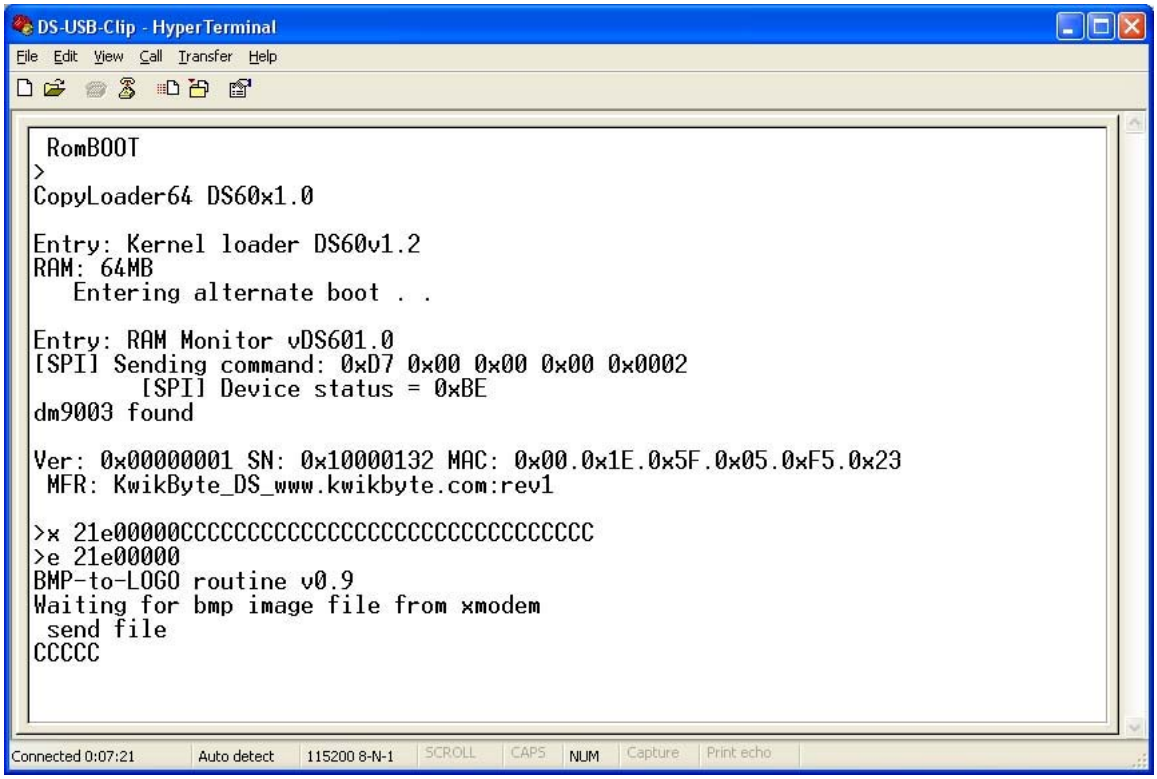
2.1 Boot the Alternate Loader

1) Hold down the 'UP Arrow' and 'SELECT' buttons while applying power (cold-boot) to the DS. You should see this screen:

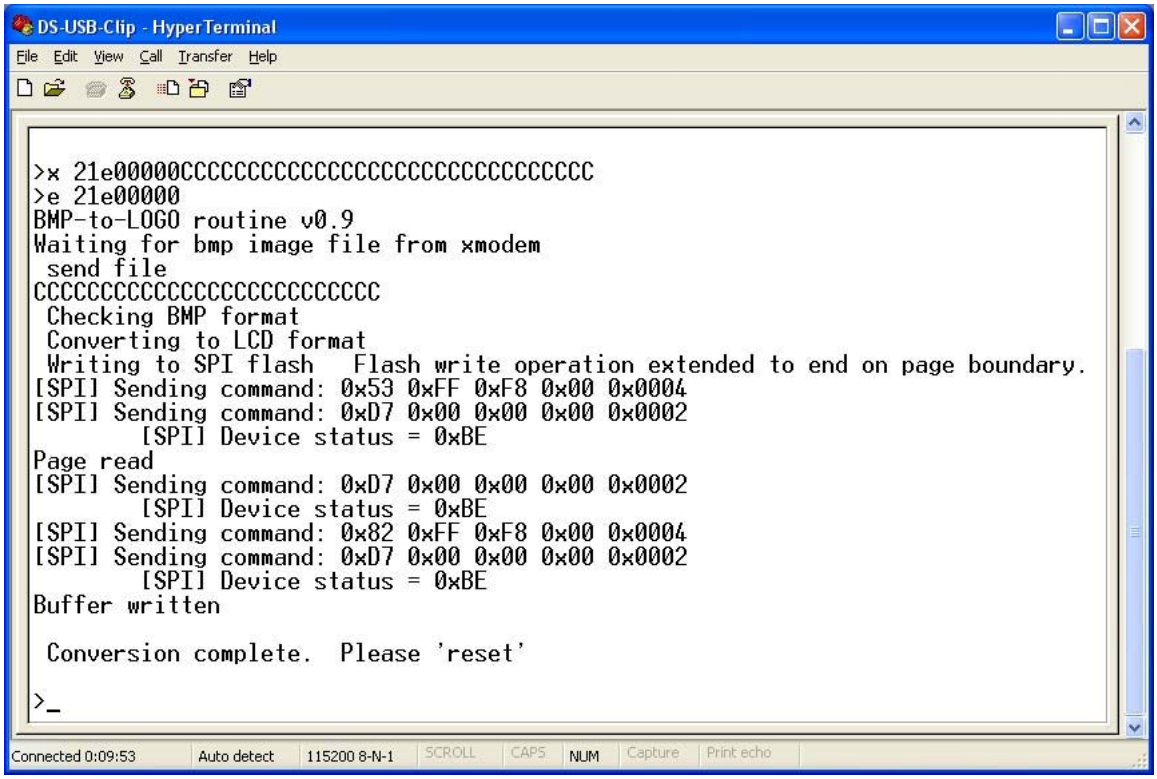


2) Start a transfer to address 0x21e00000 by typing x 21e00000 Enter. Notice the 'C' characters indicating the DS is waiting for a Xmodem transfer.

3) Send the bmpToLogo.bin file to the DS using Transfer->Send File with Xmodem protocol. This takes about 4 seconds once the transfer is started.
4) Now, execute the utility by typing e 21e00000 Enter.



5) Send the newlogo.bmp file to the DS using Transfer->Send File with Xmodem protocol (just like before). If the file is in the correct format and size, you will see this screen reporting 'Conversion complete':



6) Now type reset Enter and watch the DS boot with your new logo. You can repeat this process to update the logo again. You can change the polarity of the image with your favorite image editing software program.

3 Revisions

19JAN2009 Creation