Booting Windows CE using RedBoot from a Windows Based Host PC

by Gerry Vahe

1. PURPOSE

This application note will discuss a non-native way to load Windows CE® onto the Cirrus Logic EDB9312 development board. Traditionally, to load a Windows CE image onto a hardware platform such as the EDB9312 requires that the host PC have Platform Builder® installed. (Platform Builder is the Microsoft tool for Windows CE development and is used in conjunction with EBOOT to load an image onto a development board.) This application note will discuss the use of the RedBoot® boot loader instead. This application note assumes the user has a RAM based Windows CE image, binary code for RedBoot, and the EP9312 download utility. This method provides a way to quickly download a Windows CE image to the development board that does not require Platform Builder to be installed on the host PC.

2. BACKGROUND of RedBoot

RedBoot is a complete bootstrap environment for embedded systems. RedBoot is a versatile code module that can be readily adapted to boot any operating system. Based on the eCos Hardware Abstraction Layer, RedBoot inherits the eCos qualities of reliability, compactness, configurability, and portability.

All the source code of RedBoot is available royalty-free under the eCos License. The RedBoot source files needed for the EDB9312 are available without charge from Cirrus Logic.

RedBoot allows download and execution of embedded applications via serial or ethernet connections. This allows RedBoot to retrieve its IP parameters via BOOTP or DHCP, and program images to be downloaded using TFTP. Images can also be downloaded over a serial connection, using X- or Y-modem. It can be used for both product development (debug support) and in deployed products in the field (FLASH update and network booting).

An interactive command-line interface is provided to allow management of the FLASH images, image download, RedBoot configuration, etc., accessible via serial or ethernet. For unattended or automated startup, boot scripts can be stored in FLASH allowing for example loading of images from FLASH or a TFTP server.

3. WHAT DOES THE BOOT LOADER NEED TO DO?

Integrated 32-bit processors are quite complex. Processor products such as the Cirrus Logic 9312 are “Systems On a Chip” (SOC), and have many peripherals that need to be put into the “Quiescent” state before the chip can begin executing more complex code. Two examples of these peripherals are the PLL and the SDRAM controller.

As it turns out, the EP9312 has a few unique features that most other SOC processors don't have. It has an internal BootROM (masked in the silicon, unchangeable by the user) that performs the functions that a boot loader performs. While this is an outstanding feature, the BootROM needs to make certain assumptions in order to be functional in *most* situations. For example, the memory interface is set to less than optimal speed. More importantly, the PLL is set for EXTERNAL 1:1 mode. In most systems, this means that the EP9312 would run at about 14 MHz. This is where the RedBoot boot loader comes in.

RedBoot configures the optimum settings for the task at hand. Generally, 14 MHz would be inadequate for transferring a file over an Ethernet link. The boot loader sets up memory (including SDRAM and FLASH access times) and sets up the PLL to run the processor at a more effective speed. After the main hardware is put into a “quiescent” state, the boot loader starts initializing its features. It sets up a TFTP stack, sets up a FLASH file system, gets the serial port going for user interaction and performs other tasks needed for effective program loading.

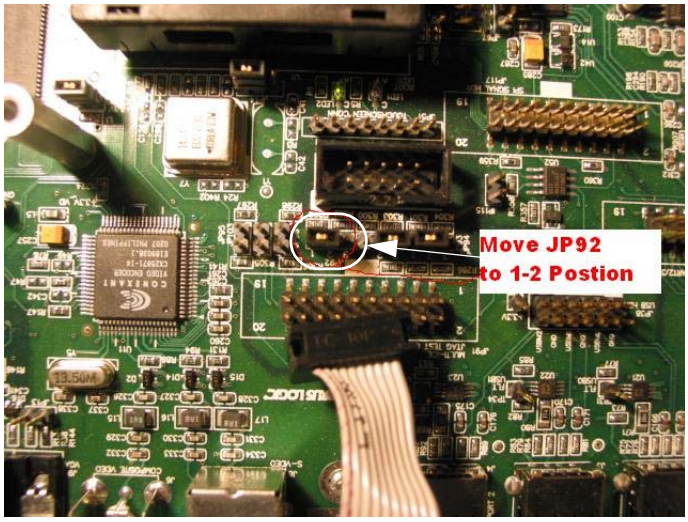
4. GETTING STARTED: Downloading RedBoot

There are two files needed to download RedBoot:

- `download.exe`
- `redboot.bin`

1) Power down the EDB9312 board, and change JP92 on your board to connect position 1-2. (This task may be easier if you remove the touchscreen cable from P91.)

Figure 1. Jumper Positions



2) For simplicity, place the download.exe and the RedBoot binary file in the same directory on the host PC.

3) Connect a null modem cable to the bottom serial port on the EDB9312 board, and to your host PC. Determine which communications port on the host PC you are connecting to.

4) MAKE SURE YOU DISABLE ANY PALM SYNC SOFTWARE RUNNING ON THE HOST PC.

5) Open a DOS window and execute the following command:

`Download -pX redboot.bin`
(Where `X` is the communications port number you determined in step 3.)
- You should see a message “Waiting for board to wake up”.

6) Power up the EDB9312 board. You should see a series of status messages, such as “Erasing FLASH”, “Programming FLASH” and so on, sequentially, each labeled with percent complete.

7) When the process is complete, power down the EDB9312 board, put jumper JP92 back to the 2-3 position, and reconnect the touchscreen cable.

8) Set up HyperTerminal (or your favorite Windows communication program) to 57600, N, 8, 1. Keep the null modem cable connected to the serial port in the lower connector on EDB9312 board and the communications port on the host PC.

9) Power up the board. After a slight delay, you should see RedBoot booting up.

5. CONFIGURING RedBoot

When RedBoot loads, execute the following commands:

- `fis init -f` This initializes the FLASH file system, which stores images and configurations. This will take a little while. You only have to do this once.
- `fconfig -i` This will configure how RedBoot behaves, and configure the network parameters. This will start an interactive script. Respond as shown below:

  -For “Init Nonvolatile Config”, enter **Y** to continue.

  -For “Boot Script”, accept **False** and press **Enter**.

  -For “BOOTP”, backspace to change to **False** and press **Enter**.

  -Leave “Gateway IP” address **Blank**

  -Set “Local IP” address to: **192.168.1.2**

  -Set “Local Address Mask” to: **255.255.255.0**

  -Set “Default Server IP” address to: **192.168.1.1**

  -Leave “DNS Server IP” address **Blank**

  -“Hardware Mac Address”: press **Enter**

  -“GDB Connection Port”: press **Enter**

  -Leave “Force Console” **False** and press **Enter**.

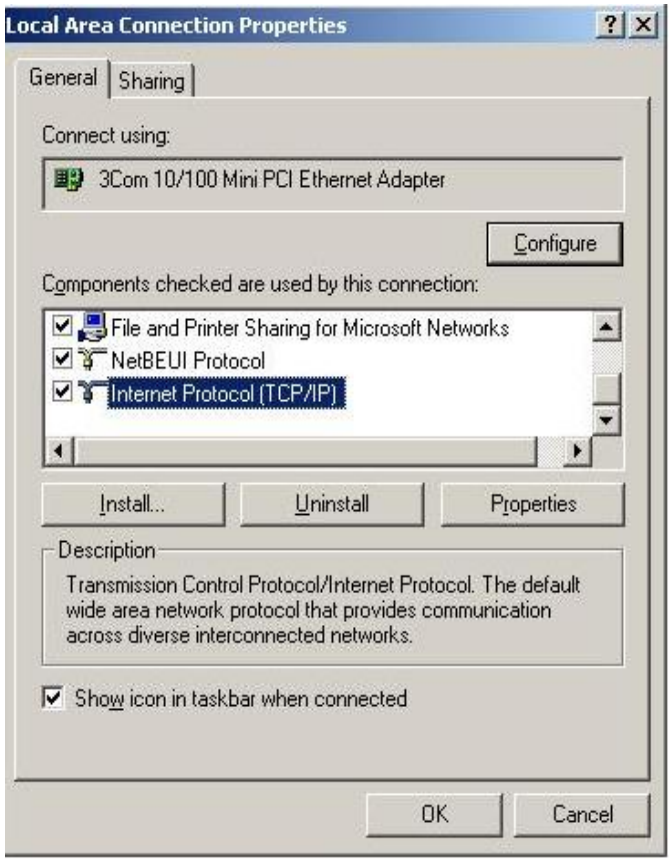
  -Leave “Network Debug At Boot” **False** and press **Enter**.

  -For “Update Nonvolatile Config”, enter **Y** to finish.

  Please, keep in mind that these parameters are subject to change. This demonstration is set up to use an ethernet cross-over cable connected to a host PC. If the host PC is on a company wide network, it is recommended that the IT department assign these IP addresses (static addresses).
- All OS's are different, but to configure the LAN settings in Windows 2000, you right click on “My Network Places”, then right click on “Local Area Connection”. You should get a

dialog box like the one shown in Figure 2. (This dialog box looks the same in Windows XP.)

Figure 2. Internet Protocol Dialog Box



- Find “Internet Protocol (TCP/IP)” at the bottom and select “Properties”. Configure your host PC LAN so that it does *not* use DHCP to get an IP address. Instead, click the selection to “Use this address”, enter “192.168.1.1” as the IP address, and press TAB. The netmask should be filled in automatically.
- Reset the EDB9312 board. This can be done by either cycling power to the board, or pressing the “Reset” button near the ATX power supply connector. After the board has been reset, check to be sure communication is re-established. From the HyperTerminal prompt, type the following command:

  ```
  ping -v -n 5 -h 192.168.1.1
  ```

  If the EDB9312 is configured correctly and communications have been re-established correctly, the response will be:

  ```
  "Received 5 of 5 Expected"
  ```

  If not, check Link Lights on the host PC and the EDB9312 board and make sure you have a null modem cable.

6. LOADING WINDOWS CE OVER ETHERNET

The Cirrus Logic Windows CE BSP has many configurable options. You can select whether you want to build the image for ROM or RAM. There are build differences, depending on the targeted memory option.

If you build the image file for ROM, you want to be able to put that image into the FLASH memory and have it stand alone. In this case, RedBoot would *not* be in the FLASH. This file will contain all of the required boot code to get the application specific Windows CE image up and running. A drawback of this approach is that if you don't have a JTAG ICE handy to program the FLASH memory, downloading a large image to FLASH memory using the serial port via the Mask ROM can be very slow.

The other option is to build an image file for RAM. Basically, this image has all of the boot information stripped off. Any hardware initialization specific to the Windows CE image would be executed out of SDRAM. This depends on the boot loader initializing SDRAM, since it is usually impossible to initialize SDRAM to a different parameter when running out of SDRAM. In this case, RedBoot handles this for us.

To load the Windows CE image over Ethernet using RedBoot, we are going to need an TFTP server running on the host PC. It is highly recommended that you use “Solar Winds TFTP” because this program is freely available and quite reliable. It can be found under the “Free Downloads” at www.solarwinds.net.

After installing this program, you will need to configure a few things. After running the program, select “File”, then “Configure” You will need to point the TFTP root directory to where the image we will be loading is stored. (This can be any directory.) You will also need to go to the security tab, and select “Transmit And Receive Files”.

Going back to the RedBoot prompt, make sure your TFTP server is running on your host PC and type the following:

```
load -r -v -h 192.168.1.1 -b 0x200000 nk.nb0
```

You can view the loading progress in the TFTP window on your host PC. You should also see a spinning bar in your RedBoot window.

The above “`load`” command took the image from your host PC, and copied it over the Ethernet link to the SDRAM on the EDB9312 board. If you cycle power to the board (power down and back up again) after that step, the image is *gone*. RedBoot remains in FLASH, so it will boot when the power is cycled. If you desire to make the Windows CE image permanent, a few more steps are required. Since the FLASH on the board is limited to a bit under 32 Mbytes when RedBoot is loaded, the Windows CE image will take up the remaining space of the FLASH due to its large size.

At this point, type “go” at the RedBoot prompt, and Windows CE will boot.

7. MAKING THE IMAGE PERMANENT

To make the image permanent in the FLASH file system, a few RedBoot commands are required after loading the Windows CE image to the RAM of the EDB9312 board.

After executing the load command described above, enter the following commands:

```
fis create -b 0x200000 -l 0x1d00000 wince
```

This command will take some time to run. The system is now copying the image that was loaded in RAM into the FLASH file system on the EDB9312 board. The elements of the command are as follows:

The ”`-b 0x200000`“ portion of the command means that when the image is loaded, the default starting address is set to 0x0020_0000.

The "`-l 0x1D00000`" portion of the command may change based on the size of your RAM Windows CE image. It tells the FLASH file system how large your image file is. In the case of the image used in this application note, it was 30,408,704 bytes long. In hexadecimal, that's 0x01D0_0000.

The "`wince`" at the end of the command directs the system to name the file in the FLASH file system "`wince`". This name can be anything you choose.

From this point on, when you can power cycle the EDB9312 board, you can now run Windows CE with the boot time considerably faster than if the image was booted directly out of FLASH. However the file is loaded in FLASH memory. If you power cycle the board, the image will be gone from SDRAM. The image file will need to be reloaded there with a RedBoot command. However, copying from FLASH to SDRAM is considerably faster than loading the image over the Ethernet link.

On the host PC in HyperTerminal, type:

```
Fis load wince
```

When this process is complete, you can type "go", and Windows CE will boot. These two steps can be used between power cycles to boot Windows CE.

There are other options in RedBoot, with bootscripts that would allow you to automatically perform the "`fis load wince`" command and then the following "go" command. This would automate the process, and not require interaction from the host PC. However, those topics will be discussed in a separate application note.

Revision	Date	Changes
1	14 June 2004	Initial Release

Contacting Cirrus Logic Support

For all product questions and inquiries contact a Cirrus Logic Sales Representative.
To find one nearest you go to **http://www.cirrus.com**