

Handout On
Arm Tool Chain and Build Kernel Image
HL
S12

I. Introduction

The cross compiler tool to build Linux kernel image for ARM11 target platform is arm-linux-gcc-4.5.1. We will describe how to use this tool in this handout.

First, locate the ARM cross compiler tool, also refers to as ARM tool chain, from the Linux distribution CD, locate arm-linux-gcc-4.5.1-v6-vfp-20101103.tgz , copy it to /tmp directory, then

```
#cd /tmp
#tar xvzf arm-linux-gcc-4.5.1-v6-vfp-20101103.tgz -C /
```

The untar operation will install the Arm tool chain in the following directory, /opt/FriendlyARM/toolschain/4.5.1 Note the above command option -C changes the directory during the untar operation.

Second, change the PATH by editing the “.bashrc” file,

```
#gedit /root/.bashrc
```

by adding to the last line of the PATH definition, as

```
export PATH=$PATH: /opt/FriendlyARM/toolschain/4.5.1/bin,
```

Then, after saving it, reboot the system to allow it to take effect, then check

```
#arm-linux-gcc -v
```

Table 1. Two steps to bring up ARM development board

Steps	Description	Note
Step 1. Prepare BIOS on the SD card	Use the manufacture’s program to format SD card and to load the manufacturer's boot loader, superboot.bin, to the host platform.	
Step 2. Copy images to the SD card	Copy the already built Linux kernel and file systems to the SD card.	

II. Prepare SD Card

2.1 Background on ARM11 BIOS

There are 2 different BIOS for ARM11 development board as described in Table 3.

Table 3. BIOS for ARM11 board.

Name of the BIOS	Note
U-boot.bin	Open source, but can not be written to SDHC card exceeding 2GB.
Superboot.bin	The manufacturer's bootloader for the board.

2.2 Prepare BIOS for ARM11

We will discuss Windows 7 or XP for BIOS prep on SD card. Note you will have to have at least 4G card. From distribution CD, from \tools folder find “SD-Flasher.exe”, then right click on

the mouse to open it as an “administrator”, the pop-up window will appear as in Figure 1, then click on “...” button to find the superbios from the \tools folder, and select it. Then insert the SD card, click “scan” on the right panel, you will see the card. Note this card under available is “No”. Then click on the “re-layout” button on the bottom of the pop-up window. Click on “yes” when prompt.

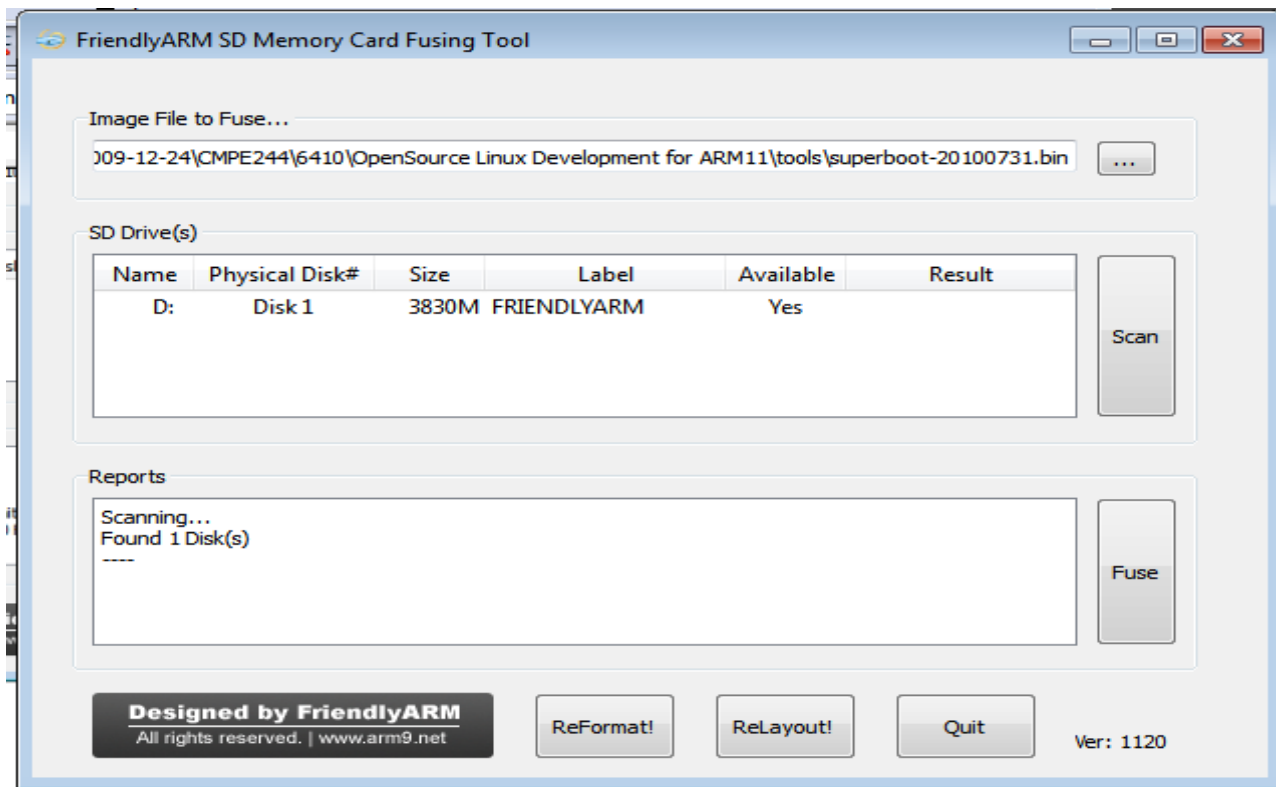


Figure 1. Using “SD-Flasher.exe” to prepare BIOS SD card.

Now, it is ready to write the BIOS to the SD card by clicking on Fuse on the right panel, once it is done, it will show in the “Report” window as in Figure 3.

Now start hyperterminal by setting hyper-terminal connection between the host computer and the ARM 11 board by using the RS232 cable provided from the development kit, (the RS232 serial cable should be straight through cable with direct connection) and connect the host PC and the ARM11 board. Note: Use COM0 port (the one next to the wall mount power connector), the settings for the hyper-terminal is given as follows:

1. bit rate 115200;
2. 8N1;
3. No flow control.

Once this setting is done, place the SW2 switch (the switch on the edge of the board next to the bank of push button switches) towards the outer edge of the board, then power on the ARM11 development kit, you can see the hyper-terminal console now displays the boot message as in Figure 4.

To see if the BIOS works, insert it to the ARM11 board, and set the S2 switch to SDBOOT, then turn the power on, when you see the LED1 flashing, that means superbios is working correctly. Now use the hyper-terminal connection, you can see the superbios message as in Figure 4. Note you can also restore SD card for your computer use by using SD-Flasher re-format function.

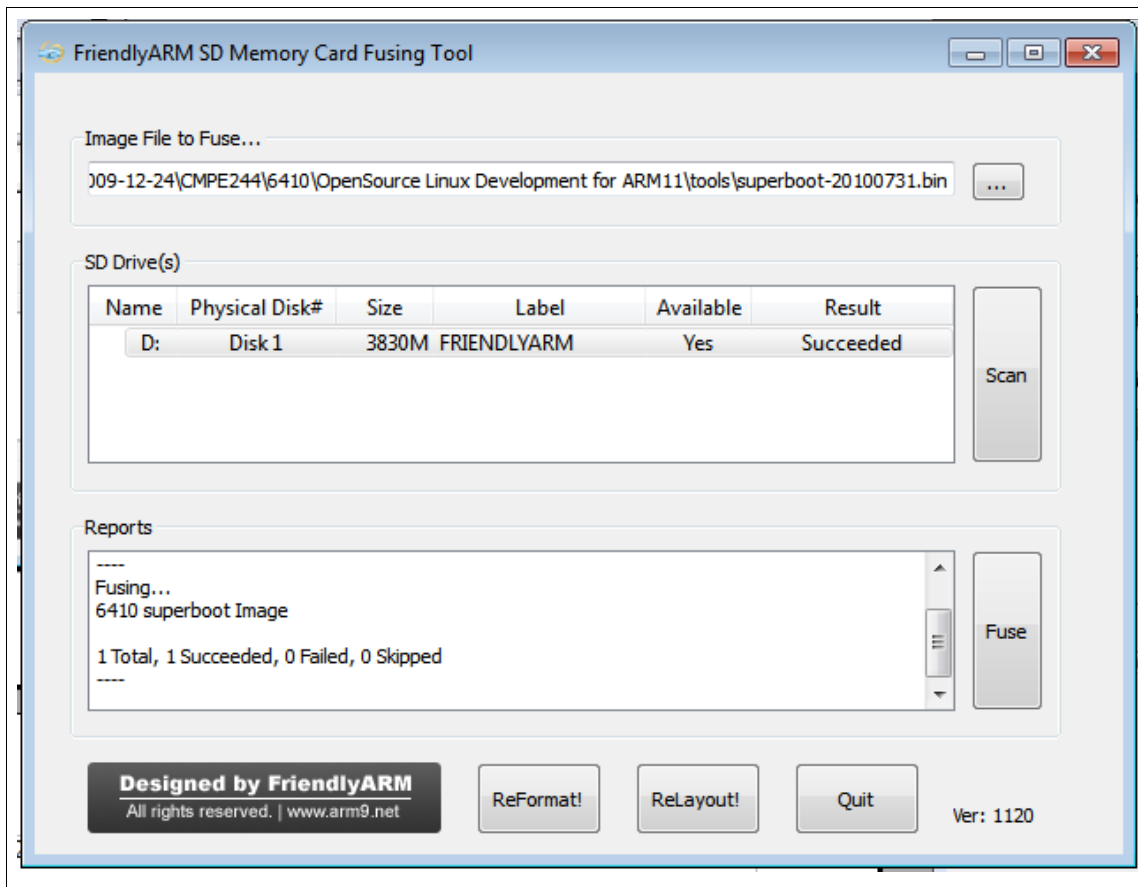


Figure 2. The BIOS “superboot” was written to SD card successfully.

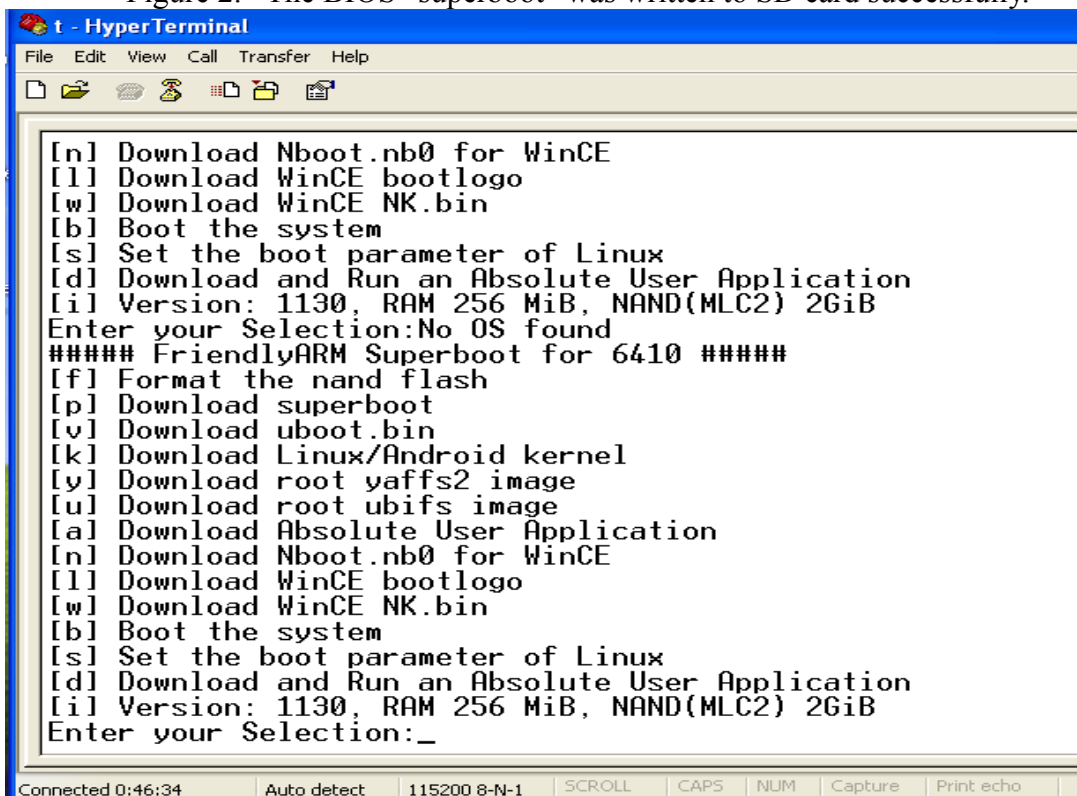


Figure 3. The hyper-terminal console displays the superboot message.
(END)