Installing a working version of Debian on the Beaglebone

7.28.2018 -- SDB

The Beaglebone Black (BBB) and Beaglebone Green (BBG) devices are great platforms for learning about embedded computing. Unfortunately, there is a great deal of version skew between different Debian images running on the Beaglebone. That means the instructions for creating a project will work on one Debian version, but fail miserably with a different one. Since Beaglebones ship with many different Debian versions, version skew presents a particular problem for students who want to work with the ADC-001 experimenter's platform. The same problem affects most other Beaglebone capes.

To mitigate this problem, I ***strongly*** recommend you reflash your BBB immediately after delivery to run the Debian 9.2 IoT release. This is the version I use with my ADC-001 cape. This document explains how to reflash your device. The process is composed of the following steps:

1. Download the Debian 9.2 IoT image onto your host PC.
2. Burn the Debian image onto a microSD card using your PC.
3. Install the microSD card into your BBB and use it to reflash the device.
4. Configure your BBB to work with the ADC-001 via the PRUSS uio interface.
5. Check that the ADC-001 works.

The remainder of this document provides detailed instructions for installing this version onto your BBB. I assume your host PC run some version of Windows. Instructions for updating the BBB using a Linux host are available at https://github.com/brorson/MusicPlaypen.

Required software and hardware

You will need the following software and hardware to complete the reflash.

Item	HW/SW	Available from	Explanation
7-zip	SW		Compression utility.
microSD to USB reader	HW	https://www.adafruit.com/product/939	Allows microSD card to be read via USB port.
microSD card	HW	https://www.adafruit.com/product/2693	Removable memory.
Win32diskimager	SW	https://sourceforge.net/projects/win32diskimager/	Utility to burn images to microSD cards.
FTDI 5V USB cable	HW	https://www.adafruit.com/product/70	Serial debug cable.
PuTTY	SW	https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html	Terminal program for Windows.
ADC-001 basic install	SW	https://github.com/brorson/ADC-001_basic_code	Basic test program for ADC-001.

Procedure

Download the Debian image onto your PC

1. Find the Debian image at the website https://beagleboard.org/latest-images. The exact image to get is “Debian 9.2 2017-10-10 4GB SD IoT”.
2. The file to download is called “bone-debian-9.2-iot-armhf-2017-10-10-4gb.img.xz”. This is a compressed file. The compression format is called “XZ”. You need to uncompress the file before using it. To uncompress the file, download and install the utility 7-zip from https://www.7-zip.org/.
3. Once you have installed 7-zip, use it to uncompress your .xz file. The result will be a simple .img file. This is the Debian image you want to burn to microSD. Note where you dumped this file (i.e. what directory it lives in).

Burn the Debian image onto a microSD card using your PC

A good supplement to my burn instructions can be found at https://learn.sparkfun.com/tutorials/sd-cards-and-writing-images.

1. If you don't have a microSD card reader on your PC, you need to get a USB adapter which accepts a microSD card. One such adapter is available from Adafruit: https://www.adafruit.com/product/939
2. You will also need to have a microSD card. If you don't have one, you can get one from Adafruit: https://www.adafruit.com/product/2693.
3. You need a utility to burn the image onto your microSD card. A common utility used to burn microSD cards is Win32diskimager, available from https://sourceforge.net/projects/win32diskimager/. Download and install this utility onto your PC.
4. Install the microSD card into your PC (likely in the USB carrier). Note which drive letter is assigned to the card. Ignore any messages from Windows about formatting the disk – don't format the disk.
5. Run Win32DiskImager.exe (you may need to run as Administrator).
6. In the program, select the drive of your microSD card.
7. Click the folder icon and select the image you downloaded.
8. Click “Write” and wait until it is done. The progress bar will sweep across the window, and a pop-up window will appear when you are done.
9. After writing the image, it's a good idea to click on “verify only” to check that the image was written correctly to the microSD card.

Reflash your BBB with the new image

1. To monitor the reflash you need a serial debug cable. The debug cable required is a 5V FTDI USB to serial cable. Such a cable is available from Adafruit: https://www.adafruit.com/product/70. More information about the debug port on the BBB is presented at https://elinux.org/Beagleboard:BeagleBone_Black_Serial.

2. You also need a serial terminal. The best option on Windows is to install PuTTY onto your PC. This program is available for download from https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html. Install using the .msi because you want all the programs which come in the PuTTY suite.
3. Hook up the serial debug cable to the debug header on BBB. (Remove any cape from the BBB before hooking up the debug cable since the cape will block access to the debug header.) Make sure pin 1 on the cable (indicated by an arrow on the connector) is connected to pin 1 on the BBB (indicated by a white dot next to pin 1).
4. Figure out what COM port was assigned to the serial debug cable by looking in the Windows Device Manager.
5. Start PuTTY (or other terminal program) to watch boot sequence and interact with BBB. Go into the serial config menu and configure PuTTY to use the following settings:
    - Baud: 115,200
    - Bits: 8
    - Parity: N
    - Stop Bits: 1
    - Handshake: None

    Then select the COM port assigned to the serial debug cable and open a serial connection to that port.
6. Start with the BBB unpowered. Install the microSD into the slot on the BBB, then hold down the S2 (power) button while applying power. Hold the button down until the LEDs flash, then release it. Let the unit boot up. In PuTTY watch the BBB boot messages until the login prompt appears.
7. When the prompt appears, log in as root.
8. Edit /boot/uEnv.txt to reflash the BBB's onboard eMMC memory upon boot: edit the last line in uEnv.txt to comment in the flasher script. When you're done, the last lines in uEnv.txt should read as follows (note the last line is commented in):

```
##enable Generic eMMC Flasher:
##make sure, these tools are installed: dosfstools rsync
cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh
```

9. Power cycle the BBB. Hold down S2 (power) again until the LEDs flash. Then release the button and wait for the eMMC to be reflashed. Flashing takes a long time. Go have a cup of coffee.
10. Once flashed, the BBB will power itself down.
11. Remove the microSD card, then apply power again and watch the reboot messages using PuTTY.
12. Once you get the login prompt, log in and verify the new Debian version is correct. Issue the following commands and verify you get the following responses:

```
root@beaglebone:~# uname -a
Linux beaglebone 4.4.91-ti-r133 #1 SMP Tue Oct 10 05:18:08 UTC
2017 armv7l GNU/Linux
root@beaglebone:~# cat /etc/dogtag
```

```
BeagleBoard.org Debian Image 2017-10-10
root@beaglebone:~# cat /etc/debian_version
9.2
```

Configure the BBB to work with the ADC-001

This step involves editing config files on the BBB so that your environment is sane, and the BBB will work with the ADC-001.

1. Log into the BBB via the serial debug connection.
2. Allow user root to log in via other ssh. Edit the file /etc/ssh/sshd_config, and change the line

   From: `#PermitRootLogin without-password`

   To: `PermitRootLogin yes`

   After reboot, you will be able to ssh into the BBB as user root, password root.
3. Next edit the file /boot/uEnv.txt to disable communication with the PRU via rproc and enable communication via uio. Find the PRUSS OPTIONS section and edit it to read as follows:

   ```
   ###PRUSS OPTIONS
   ###pru_rproc (4.4.x-ti kernel)
   #uboot_overlay_pru=/lib/firmware/AM335X-PRU-RPROC-4-4-TI-00A0.dtbo
   ###pru_uio (4.4.x-ti & mainline/bone kernel)
   uboot_overlay_pru=/lib/firmware/AM335X-PRU-UIO-00A0.dtbo
   ```

   The important change is to comment out the RPROC overlay and comment in the UIO overlay.
4. Next edit the device tree to blacklist rproc and enable uio for communication with the PRU. The file to edit is `/opt/source/dtb-4.4-ti/src/arm/am335x-bonegreen.dts`. Edit the file so that it reads as follows:

   ```
   /* pruss: pick one: */

   /*
    * /etc/modprobe.d/pruss-blacklist.conf
    *
    * blacklist uio_pruss
    */

   /* #include "am33xx-pruss-rproc.dtsi" */

   /*
    * /etc/modprobe.d/pruss-blacklist.conf

    * blacklist pruss
    * blacklist pruss_intc
    * blacklist pru-rproc
   */

   #include "am33xx-pruss-uio.dtsi"
   ```

   The important changes are the comment out the “#include rproc” line and comment in the

"#include uio" line.

5. Then rebuild the device tree as follows (note that you need to be root in order to complete these commands successfully):

```
cd /opt/source/dtb-4.4-ti
make
make install
```

   The last step ("make install") copies the device tree files into /boot/dtbs, where they may be found during boot-up.

6. Edit /boot/uEnv.txt and change the empty dtb= line to read:

```
dtb=am335x-bonegreen.dtb
```

7. While in /boot/uEnv.txt, tell the BBB where to find the ADC-001 device tree by changing this line:

   From: `#dtb_overlay=/lib/firmware/<file8>.dtbo`
   To: `dtb_overlay=/lib/firmware/ADC_001-00A0.dtbo`

   Then save uEnv.txt and exit the editor.

8. Now blacklist the rproc drivers by creating a new file /etc/modprobe.d/pruss-blacklist.conf. In the file place the following lines:

```
blacklist pruss
blacklist pruss_intc
blacklist pru-rproc
```

9. When done, power cycle the BBB and log into it again. (It may take a minute or more to boot up – be patient.) Issue the command `lsmod | grep uio` to check for success. You should get all ***three*** lines if you have successfully enabled the PRUSS uio system like this:

```
root@beaglebone:~# lsmod | grep uio
uio_pruss               4629  0
uio_pdrv_genirq         4243  0
uio                    11100  2 uio_pruss,uio_pdrv_genirq
```

Check the ADC-001 works

1. Remove the debug cable and install the ADC-001 cape onto your BBB.
2. Download the basic ADC-001 code from GitHub to your host PC. The code is in the repo https://github.com/brorson/ADC-001_basic_code. I suggest you download the package as a .zip file. Note the directory where you have placed the .zip file (probably under Downloads).
3. From your windows PC, use sftp (provided by PuTTY) to copy the .zip file to your BBB. When you start sftp you will be placed into an FTP command window. Sftp will start in its own directory, so you must first cd into the directory where the .zip file lives before copying it down to your BBB. Use the following commands to copy the .zip file down:

```
lcd <directory where your .zip file lives>
open root@192.168.7.2
<password>
put ADC-001_basic_code-master.zip
```

```
bye
```

4. Now log into your BBB again as root. You should find the .zip file in your home directory. Unzip is using the command “unzip ADC-001_basic_code-master.zip”.
5. Now you can cd into the project directory and build the code as follows:

   ```
   cd ADC-001_basic_code-master
   make
   ```

   You will see build message scoll by as the code builds. You may ignore any warnings you see about unreachable statements, or about clock skew and incomplete builds.
6. Reboot. This allows the BBB to find the new device tree file you placed into /lib/firmware in step 5.
7. Now run the program by typing “./main” at the command line. The program will execute a series of tests. It will first tell you the test it is about to run, then it will run the test and give you the result. If you get all the way through to the end, the program will go into a loop and read and print values from the ADC-001 forever. You can close the test program by hitting <ctrl>-c at your keyboard. If you get a “segmentation fault” error before this point, then there is a problem.
8. You are done.