EE192 Getting Started with the Beaglebone Blue

Version: January 22, 2019

GSI: Justin Yim

1. Flashing the Operating System

Required materials:

- Beaglebone Blue and power (micro USB or 9-18V barrel connector)
- Windows, Mac, or Linux Computer with an SD card slot and microSD adapter
- microSD card at least 4 GB in size

The board is already flashed. Follow this procedure if you need to re-flash your board. For more information, see http://strawsondesign.com/docs/librobotcontrol/flashing.html The Beaglebone Blue can boot from either its onboard eMMC flash memory or from a microSD card. In Debian, the Programmable Real-time Units (PRUs) will work only if the OS is flashed to the onboard memory (we never got them to work in Ubuntu).

Process:

If you already have a microSD card with the desired image, skip to step 3. Otherwise, follow steps 1 and 2 to prepare your microSD card.

1. Download the OS image to your computer (we pre-installed Ubuntu 18.04):

   Ubuntu 18.04: https://rcn-ee.net/rootfs/2018-12-10/flasher/BBB-eMMC-flasher-ubuntu-18.04.1-console-armhf-2018-12-10-2gb.img.xz

   Debian 9.5: http://strawsondesign.com/docs/images/BBBL-blank-debian-9.5-iot-armhf-2018-10-07-4gb.img.xz

2. Write the image to the microSD card:
   a. Download and install Etcher on your computer: https://www.balena.io/etcher/
   b. Insert the microSD card and then start Etcher.
   c. In Etcher, select the image. The microSD card should be automatically selected. Flash it. This can take about 15 minutes.
3. Flash the Beaglebone Blue eMMC:
   a. Insert the microSD card into the Beaglebone Blue while it is powered off.
   b. Hold down the Beaglebone Blue “SD” button (in the corner) while applying power. You can release the button when the LEDs flash.
   c. After a minute, the blue LEDs will flash in a pattern bouncing back and forth. After about five minutes the LEDs will turn off and you can remove power from the Beaglebone Blue. Remove the microSD card so that it does not flash the Beaglebone Blue again.
4. Connect to the Beaglebone Blue and test that it is working.

2. Connecting to the Beaglebone Blue

Required materials:

- Beaglebone Blue
- MicroUSB cable
- Windows, Mac, or Linux Computer with a USB connection

The Cory 204 lab computers' Windows operating systems can connect to the Beaglebone Blue through either an Ubuntu virtual machine or the SSH client PuTTY. If you'd like to use your personal computer, follow the setup steps below for your operating system. For more information, see: http://beagleboard.org/getting-started/
and http://strawsondesign.com/docs/librobotcontrol/networking_usb.html

Mac

- Download the two drivers:
  Network:
  https://beagleboard.org/static/Drivers/MacOSX/RNDIS/HoRNDIS.pkg
  Serial:
  https://beagleboard.org/static/Drivers/MacOSX/FTDI/EnergiaFTDIDrivers2.2.18.pkg
  and install them. The Serial driver may ask you to reboot your computer.

Windows

- Download the driver:
  For 64 bit: https://beagleboard.org/static/Drivers/Windows/BONE_D64.exe
  For 32 bit: https://beagleboard.org/static/Drivers/Windows/BONE_DRV.exe
  and install it. You may need to reboot your computer after installation.
- Download and install an SSH client like putty: https://www.putty.org/

Linux

- No additional steps; you should be ready to go. :)

Connecting with USB:

1. Plug the micro USB B into the Beaglebone Blue and attach the USB A to your computer. Wait for the Beaglebone Blue to power on and boot.
2. Lab computers (using Ubuntu):
   a. Start the VirtualBox Ubuntu image, wait for it to boot, and open a terminal
   b. Pass the Beaglebone Blue USB device into the Ubuntu virtual machine

   Alternatively, open PuTTY and follow step 3:
3. The Beaglebone Blue creates a network connection and emulates an Ethernet adapter. Your computer will receive IP addresses 192.168.7.1 and 192.168.6.1. The Beaglebone Blue has IP addresses 192.168.7.2 and 192.168.6.2. SSH into

the Beaglebone Blue at 192.168.7.2 or 192.168.6.2 with username `ubuntu` (or `debian`) and passphrase `temppwd`. In Linux or Mac:

```
ssh ubuntu@192.168.7.2
```

If you receive a "REMOTE HOST IDENTIFICATION HAS CHANGED" error, you have probably previously connected to a different computer on 192.168.7.2 (like another BBBL). Remove the old ECDSA key with something like:

```
ssh-keygen -R 192.168.7.2
```

Connecting with the Beaglebone Blue's Wi-Fi:

The Beaglebone Blue has a wireless access point called `Beaglebone-XXXX` where `XXXX` depends on your individual board. Its passphrase is `BeagleBone` (case sensitive). The Beaglebone Blue can remain connected to another network while this access point is active. For more information see: http://beagleboard.org/getting-started

1. Connect to the Beaglebone Blue Wi-Fi network from your laptop (the lab computers may not have Wi-Fi cards). Your computer should receive an IP address in the 192.168.8.x range. The Beaglebone Blue will have IP address 192.168.8.1
2. SSH into the Beaglebone Blue at 192.168.8.1 in the same manner as for the USB connection:

   ```
   ssh ubuntu@192.168.8.1
   ```

3. Connecting to Wi-Fi:

Once the Beaglebone Blue connects to a Wi-Fi network like CalVisitor it will remember it and automatically connect the next time it boots (it should already do this). For more information, see http://strawsondesign.com/docs/librobotcontrol/networking_wifi.html

Connecting to CalVisitor:

1. On the Beaglebone Blue run:

   ```
   connmanctl
   ```

   and then in connmanctl enter `services`. This should print the names of available Wi-Fi networks. You should see CalVisitor among them.

2. In connmanctl connect to the CalVisitor network with:

   ```
   connect wifi_8030dc034755_43616c56697369746f72_managed_none
   ```

   don't worry, you can tab complete. The green light in the center of the board should turn on. You can test the connection by pinging something like google.com.

Connecting to a Password Secured Network:

1. To connect to a network NAME with passphrase PASSPHRASE, create a file `NAME.config` in `/var/lib/connman/` containing the following text:

   ```
   [service_NAME]
   Type = wifi
   Name = NAME
   Passphrase = PASSPHRASE
   ```

   This can be accomplished with a text editor like vim, emacs, or nano run as root.

2. The Beaglebone Blue may automatically connect to the network once this file is created. If not, run `connmanctl` and enter:

   ```
   tether wifi off
   enable wifi
   scan wifi
   services
   agent on
   connect wifi_f45eab…_managed_psk
   PASSPHRASE
   quit
   ```

4. Setting up the Robot Control Library

We have already installed the Robotics Control Library librobotcontrol. If you need to reinstall it, follow this procedure. For more information see: http://strawsondesign.com/docs/librobotcontrol/installation.html (specifically method 2) Later in the course we will provide skeleton code for ROS in Ubuntu 18.04 with updated instructions.

Robot Control Library in Ubuntu 18.04 or Debian 9.5:

1. Connect to CalVisitor to make sure you have an internet connection.
2. Download the librobotcontrol Debian package (the following is one line):

   ```
   wget
   https://github.com/StrawsonDesign/librobotcontrol/rele
   ases/download/V1.0.4/librobotcontrol_1.0.4_armhf.deb
   ```

3. Install the package with:

   ```
   sudo dpkg -i librobotcontrol_1.0.4_armhf.deb
   ```

   You may be prompted to select a program to run on boot. You can pick rc_blink if you'd like the green and red LEDs to blink on boot, or leave it on none.

4. Clone the librobotcontrol repository for the source code (one line):

   ```
   git clone
   https://github.com/StrawsonDesign/librobotcontrol.git
   ```

5. To make sure the installation worked, run:

   ```
   rc_test_drivers
   ```

   and you should see the following output if everything is working (Ubuntu 18.04):

   ```
   Kernel: 4.14.79-ti-r84
   rcn-ee.net console Ubuntu Image 2018-12-10
   Debian: buster/sid

   PASSED: gpio 0
   PASSED: gpio 1
   PASSED: gpio 2
   PASSED: gpio 3
   PASSED: pwm0
   PASSED: pwm1
   PASSED: pwm2
   PASSED: eqep0
   ```

```
PASSED: eqep1
PASSED: eqep2
ERROR:  pru-rproc driver not loaded
PASSED: uart1
PASSED: uart2
PASSED: uart4
PASSED: uart5
PASSED: i2c1
PASSED: i2c2
PASSED: spi
PASSED: LED
PASSED: ADC iio

Currently running on a:
MODEL_BB_BLUE
Robot Control library Version:
1.0.4
```

The PRU should return PASSED if you are using Debian 9.5.

5. Building and Running a Project

Later in the course we will provide skeleton code for ROS in Ubuntu 18.04 with updated instructions.

Robot Control Library:

For more information, see http://strawsondesign.com/docs/librobotcontrol/manual.html or take a look at the examples in the library source code ~/librobotcontrol/examples/

1. Create a new project:
    a. Decide on a project name (not new_project_name) and copy the project template (the following is one line):

    ```
    cp -r /usr/share/robotcontrol/rc_project_template
    ~/new_project_name
    ```

    b. Rename the C source code file:

    ```
    mv rc_project_template.c new_project_name.c
    ```

    c. Edit the Makefile line starting with TARGET to:

    ```
    TARGET = new_project_name
    ```

2. Build the project by calling

    ```
    make
    ```

    from the project folder

3. Run the project with

    ```
    ./new_project_name
    ```

    from the project folder. By default it will toggle the green and red LEDs when you push the "PAU" (pause) button.