

# SOEN 422

## Tutorial 3

# Outline

1. Installing the Beaglebone Black USB Drivers
2. Installing Ubuntu on the Beaglebone Black
3. Digital Output on GPIO

# Installing Beaglebone Black USB Drivers

- Depending on your operating system, you may need to install USB drivers for the Beaglebone Black on your workstation.
  - See the [Beaglebone Black website](#) under Step 2 for more details.

# Installing Ubuntu - Downloading Image

- The default OS on the Beaglebone Black is Angstrom Linux, however, we will be using Ubuntu Linux because it has many more drivers and packages available.
- The installation method we will be using is to flash an image onto the Beaglebone Black's memory which is already installed and configured.
- The latest stable version of Ubuntu Linux that runs on the Beaglebone Black is Ubuntu 14.04 (armhf, hard-float). The image can be found [here](#). Installing other distributions has the same procedure as long as you can find the appropriate disk image.

# Installing Ubuntu - Writing Image to Micro SD Card

- The procedure for writing the image to the micro SD card depends on your workstation's operating system.

# Installing Ubuntu - Write Image (Windows)

- First decompress the image using an archive utility such as [7Zip](#).
- Next, install the [Image Writer for Windows](#) application.
- Connect the micro SD card to your workstation and open the image writer application.
- Navigate to your OS image and select the drive that the image is to be written to, be sure that this drive is correct (very important or you may lose data accidentally). When ready, press the "Write" button.
- When the writing is complete, you may eject the micro SD card.

# Installing Ubuntu - Write Image (OS X)

- Detailed instructions are available from [Adafruit](#).

# Installing Ubuntu - Write Image (Linux)

- Before plugging in the SD card to your workstation, run the following command.

```
ls /dev/sd*
```

- Take note of which devices are present (sda, sdb, and so on..).
- Plug in the SD card to your workstation and re-run the above command. A new device should be listed, this is the device we need to write to.
- Now we need to simply decompress the image and write it to the device. It is very important to ensure you have selected the correct drive, otherwise you may lose data accidentally.

```
xz -cd {your_disk_image}.img.xz > /dev/sd{X}
```

- In the above command, replace the path of the disk image with the one you downloaded and the device letter with the one which corresponds with your SD card (eg. /dev/sdb). This may take some time and the procedure is complete when the command prompt re-appears.



# Flashing the Image onto the Beaglebone Black

- To flash the image onto your Beaglebone Black, insert the micro SD card into the micro SD card slot.
- While holding down the button above this micro SD slot, apply power to the Beaglebone Black. Once the Beaglebone Black begins to boot, you may stop holding the button. The flashing process may take up to 45 minutes and will be complete when all four LEDs are steadily lit. At this point power off, remove the micro SD card and power back on to complete the process.
- **Note:** To be sure that the image is flashing, you should see all four LEDs light up at the beginning of the process.

# SSH into Beaglebone Black

- Username: ubuntu
- Password: temppwd

```
ssh ubuntu@192.168.7.2
```

# Digital Output

- The GPIO pins on the Beaglebone Black are managed by the Linux OS through the device tree, an abstraction of the device's hardware represented in the filesystem.
- These pins can be manipulated by simply writing to files on the file system

# Digital Output

- First, we need to tell the operating system which pin we are using. See this [diagram](#) for pin number and locations.

```
#include <stdio.h>

// Open the export file
FILE *exportFile;
exportFile = fopen("/sys/class/gpio/export", "w");

// Write the pin that we want to use to the file:
fprintf(exportFile, "67");

// Close the file
fclose(exportFile);
```

# Digital Output

- Now, we want to set the data direction on this pin.

```
// Open the Pin 67 Data Direction File
FILE *dataDirection67;
dataDirection67 = fopen("/sys/class/gpio/gpio67/direction", "w");

// Write the direction as output to the file:
fprintf(dataDirection67, "out");

// Close the file.
fclose(dataDirection67);
```

# Digital Output

- Finally, we can write the value to the pin.

```
// Open the Pin 67 value file
FILE *value67;
value67 = fopen("/sys/class/gpio/gpio67/value", "w");

// Write the direction as HIGH (1), for LOW, use 0.
fprintf(value67, "1");

// Close the file.
fclose(value67);
```

# Copyrights



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).