

Embedded Software Kata

[Find the up to date Kata on Google Drive here](#)

Goal

The goal of this Kata is for you to build a device to detect whether or not the phone booth is occupied. You'll be using a sonar sensor and beaglebone to create a device that will display the distance between a possible phone booth occupant and the sonar sensor.

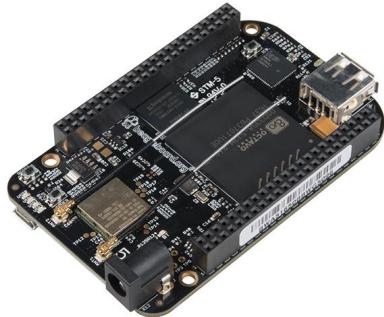
Tools needed	2
Setting up a Beaglebone	3
Get a Linux image on the Beaglebone	3
Copy the downloaded image to your MicroSD	3
Flash the downloaded image on your BBB	4
Getting Python on your BBB	5
Setting up the Sonar Sensor HC-SR04	6
Understanding the HCSR04 Sensor	6
Why do I need a resistor on the echo path into the GPIO?	8
Using a Multimeter to find the right resistor	8
Running the Python Script on the BBB	8
Troubleshooting	9
Appendix I Internet Access	10
To get internet access on a non-wifi enabled beaglebone	10
Setup Wifi on BBB w/ Wifi	11
Confirm your network connectivity	12
Appendix II SSH to your BBB over USB	12
Installing Mac Drivers	12

Tools needed

BeagleBone Black, aka BBB, with or without wifi

For this Kata, the BBB will need to be connected to a 5V AC Adaptor. Power through the microUSB is not enough.

You'll also need a way to connect to the BBB, usb c to micro usb is recommended.



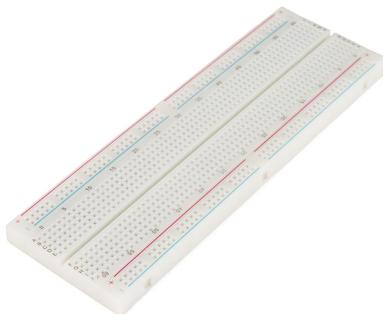
Non wifi - https://smile.amazon.com/BeagleBone-Black-Single-Computer-Development/dp/B00LC1924G/ref=sr_1_3?keywords=beaglebone+black&qid=1558014930&s=gateway&sr=8-3

Wifi -

https://smile.amazon.com/Electronics123-BeagleBone-Black-Wireless/dp/B073HL8C2X/ref=sr_1_6?keywords=beaglebone+black+wifi&qid=1558015123&s=gateway&sr=8-6

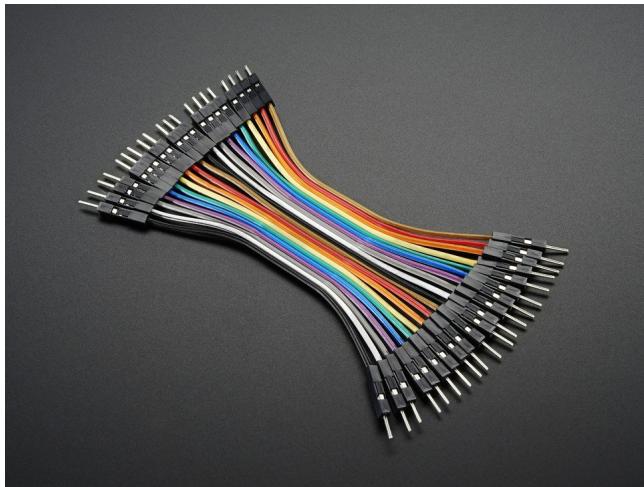
Breadboard

<http://wiring.org.co/learning/tutorials/breadboard/>



Breadboard Wires

https://smile.amazon.com/dp/B005TZJ0AM/ref=cm_sw_r_tw_dp_U_x_c0w3Cb5A487T6



Resistors

Particularly 1kOhm - Brown, Black, Red, Gold

This calculator can help will tell you what colors you need.

<https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code-4-band>



Setting up a Beaglebone

Get a Linux image on the Beaglebone

Download a Linux Image [Debian 9.5 2018-10-07 4GB SD IoT](#)

<https://beagleboard.org/latest-images>

Place a microSD card into your computer's SD slot using an SD Card adaptor

Copy the downloaded image to your MicroSD

- **The etcher way:**

Download and install [Etcher](#).

Put the SD card into your mac

Open Etcher and select the image to be flashed onto the SD card

Eject the SD card from your computer

- **Using sudo dd**

From the command terminal on your mac run

```
diskutil list
```

and make a note of which disk it is in, ie disk2,disk3,disk4, example below uses disk3 (at the bottom of the image)

AMAC021401ESGJM:~ bryan.karczynski\$ diskutil list				
<pre>/dev/disk0 (internal):</pre>				
#:	TYPE	NAME	SIZE	IDENTIFIER
0:	GUID_partition_scheme		500.3 GB	disk0
1:	EFI	EFI	314.6 MB	disk0s1
2:	Apple_APFS Container	disk1	500.0 GB	disk0s2
<pre>/dev/disk1 (synthesized):</pre>				
#:	TYPE	NAME	SIZE	IDENTIFIER
0:	APFS Container Scheme -		+500.0 GB	disk1
		Physical Store disk0s2		
1:	APFS Volume	Macintosh HD	210.5 GB	disk1s1
2:	APFS Volume	Preboot	46.5 MB	disk1s2
3:	APFS Volume	Recovery	509.8 MB	disk1s3
4:	APFS Volume	VM	1.1 GB	disk1s4
<pre>/dev/disk2 (external, physical):</pre>				
#:	TYPE	NAME	SIZE	IDENTIFIER
0:	FDisk_partition_scheme		*18.9 MB	disk2
1:	Windows_FAT_16	BEAGLEBONE	17.8 MB	disk2s1
<pre>/dev/disk3 (external, physical):</pre>				
#:	TYPE	NAME	SIZE	IDENTIFIER
0:	FDisk_partition_scheme		*31.9 GB	disk3
1:	Windows_FAT_32	BEAGLE_BONE	74.0 MB	disk3s1
2:	Linux		3.6 GB	disk3s2

Run this command from the folder containing your downloaded debian image to copy the image to your sd card:

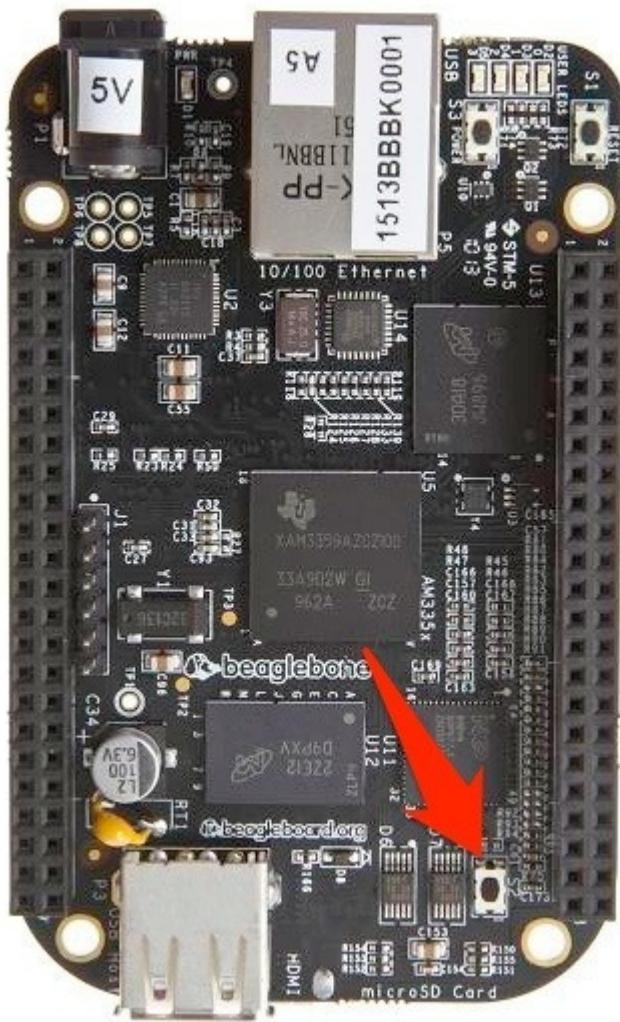
```
sudo dd bs=1m if=bone-debian-9.5-iot-armhf-2018-10-07-4gb.img  
of=/dev/rdisk3 conv=sync
```

There are other ways to get the downloaded image to the microSD card, Finder, cp,etc.

Flash the downloaded image on your BBB

With no power going to the BBB, place the microSD card into the BBB. Hold down the User Boot button (Seen in the image below) and supply power to the BBB. After 4 blue lights have all lit and gone out, you can let go of the User Boot button. After 30-45 minutes the blue lights

should stop blinking and the BBB will be imaged. Once it's done, the bank of 4 LED's to the right of the Ethernet will all turn off. You can then power down your BeagleBone Black. For more details of imaging a BBB, see <https://learn.adafruit.com/beaglebone-black-installing-operating-systems/flashing-the-beaglebone-black>



Getting Python on your BBB

You will need internet access for your BBB to get the necessary packages your code will be using. On a non-wifi BBB, to set up port forwarding and have internet through your USB, follow [Appendix I](#).

apt-get is used by linux to add software packages

[For more information on apt-get](#)

Use these two commands on the commandline in your BBB to get python

```
sudo apt-get update  
sudo apt-get install build-essential python-dev python-setuptools  
python-pip python-smbus -y
```

Next we will need Adafruit for pin communication on the BBB. Using python pip enter this command on the BBB:

```
sudo pip install Adafruit_BBIO
```

Next we will set up the physical components of the BBB and the Sensor

Setting up the Sonar Sensor HC-SR04

This is a diagram of the physical setup we are attempting to achieve:

[This setup is from this github project.](#)

Understanding the HCSR04 Sensor

The Sonar Sensor has four pins:

VCC - Power

Trig - Sends signal out

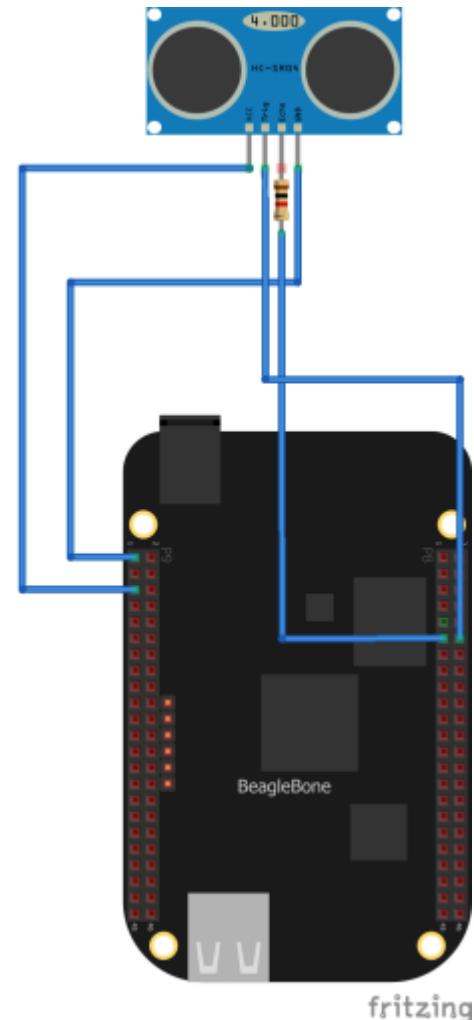
Echo - listens for bounceback

Grnd - Ground

The last thing we should connect is the power to the sensor. The Grnd and VCC manage the power supply to the Sensor. The Trig will be connected to a pin defined as outward from the BBB to the sensor to cause a pulse that will be listened for by the echo pin. The echo pin will hear the bounce back and send a signal back into the BBB. The time between the emitting of the pulse from the trigger to the time the pin receives back a signal in the echo is the distance-time. This distance-time will be used to calculate the distance an object is from the sensor.

The sensor needs 5V power so we will connect that last.

[Documentation on the Sonar Sensor](#)



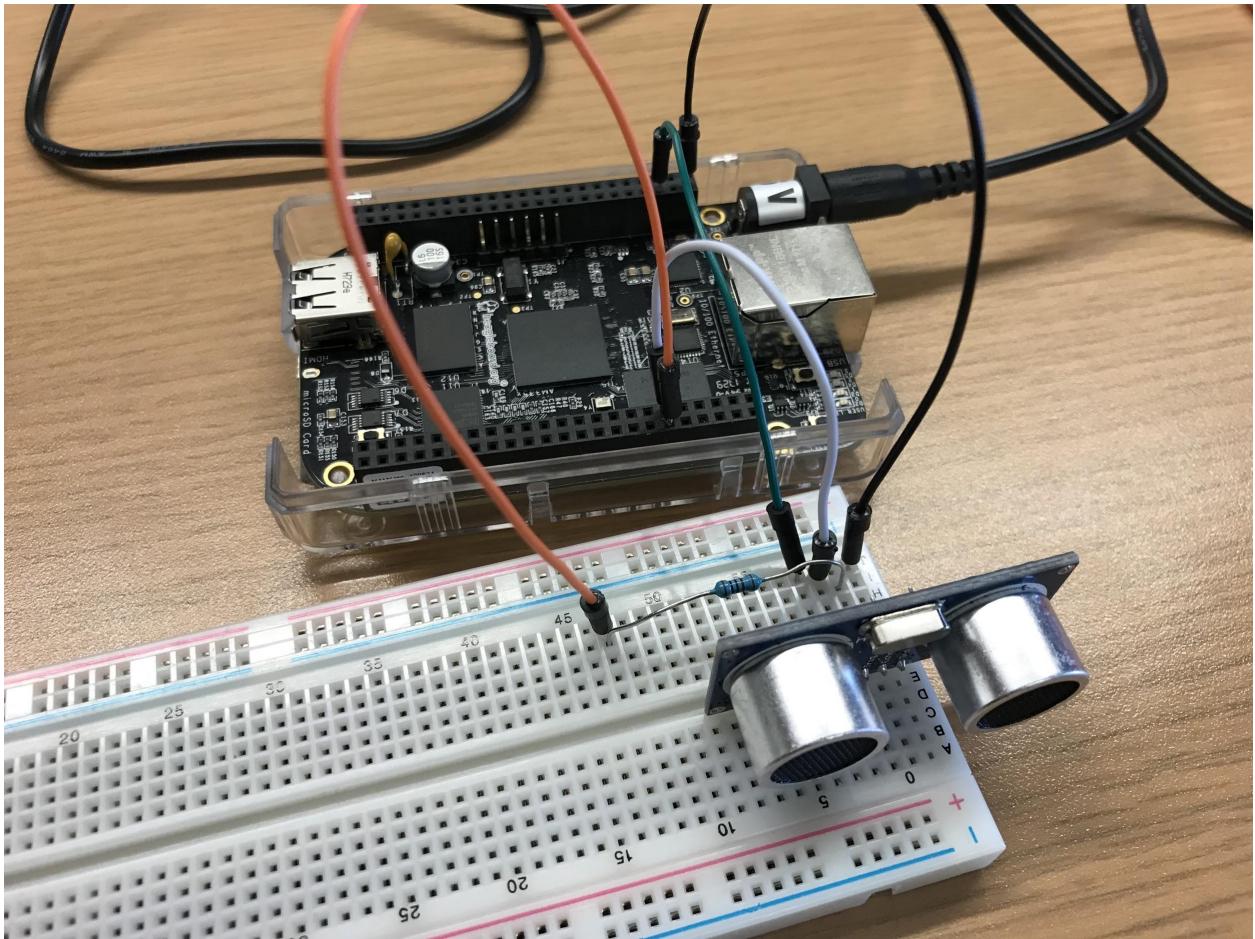
The black bars on the left and right of the BBB in the image below are pinholes that can take in wires for sending and receiving signals on the BBB. We will be using P8_11 for Echo and P8_12 for Trigger, and P9_1 for ground, and P9_5 for 5V power. GPIO means general purpose input and output.

Beaglebone Black Pinout Diagram

P9			P8		
Function	Physical Pins	Function	Function	Physical Pins	Function
DGND	1	DGND	DGND	1	DGND
VDD 3.3 V	3	VDD 3.3 V	MMC1_DAT6	3	MMC1_DAT7
VDD 5V	5	VDD 5V	MMC1_DAT2	5	MMC1_DAT3
SYS 5V	7	SYS 5V	GPIO_66	7	GPIO_67
PWR_BUT	9	SYS_RESET	GPIO_69	9	GPIO_68
UART4_RXD	11	GPIO_60	GPIO_45	11	GPIO_44
UART4_TXD	13	EHRPWM1A	EHRPWM2B	13	GPIO_26
GPIO_48	15	EHRPWM1B	GPIO_47	15	GPIO_46
SPI0_CS0	17	SPI0_D1	GPIO_27	17	GPIO_65
I2C2_SCL	19	I2C_SDA	EHRPWM2A	19	MMC1_CMD
SPI0_DO	21	SPI0_SLCK	MMC1_CLK	21	MMC1_DAT5
GPIO_49	23	UART1_TXD	MMC1_DATA4	23	MMC1_DAT1
GPIO_117	25	UART1_RXD	MMC1_DATO	25	GPIO_61
GPIO_115	27	SP11_CS0	LCD_VSYNC	27	LCD_PCLK
SP11_DO	29	GPIO_112	LCD_HSYNC	29	LCD_AC_BIAS
SP11_SCLK	31	VDD_ADC	LCD_DATA14	31	LCD_DATA15
AIN4	33	GND_ADC	LCD_DATA13	33	LCD_DATA11
AIN6	35	AIN5	LCD_DATA12	35	LCD_DATA10
AIN2	37	AIN3	LCD_DATA8	37	LCD_DATA9
AIN0	39	AIN1	LCD_DATA6	39	LCD_DATA7
GPIO_20	41	ECAPWMO	LCD_DATA4	41	LCD_DATA5
DGND	43	DGND	LCD_DATA2	43	LCD_DATA3
DGND	45	DGND	LCD_DATA0	45	LCD_DATA1

If you haven't already, [check out this tutorial on how to use a breadboard](#). Particularly on how rows and columns work.

First place the sensor in row F of the breadboard, as seen below. Connect a wire for ground from P9_1(DGND, physical pin 1 on the left, above) anywhere in the first column of the breadboard. This should be in the same column as the Grd on the Sensor. Next, place a 1k Ohm Resistor from second column, ECHO to another column with no Sensor pins (column 5 or greater). Connect a wire with the other end of the resistor row to P8_11 in the BBB. This is the path the signal will return from into the BBB. Next, plug in a wire from Column 3 on the breadboard, Trig pin, to P8_12. This connection will send a short signal out from the BBB to the sensor to start the pulse. Lastly connect the 5V power from P9_5 to VCC, fourth column on the board. It should look something like the image below.

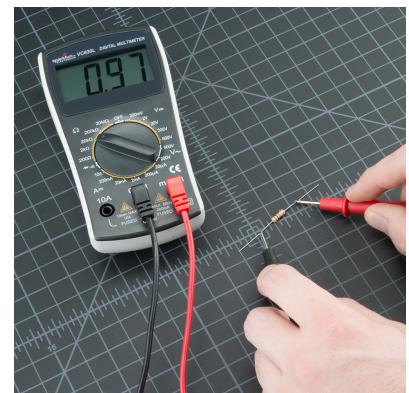


Why do I need a resistor on the echo path into the GPIO?

The BBB can receive a max 3.3V. Any more voltage could damage the BBB. The resistor drops the incoming voltage to an appropriate amount to be able to be picked up by the BBB.

Using a Multimeter to find the right resistor

Sometimes the colors of a resistor can be difficult to see. Try using a voltmeter by pressing the positive(red) and negative(black) leads to each side of a resistor. The voltmeter should be set to the Ohms setting to see the resistance, as seen in the image below.



Running the Python Script on the BBB

There is a short Python Script available on this GitHub page that can be run to use the Sonar Sensor. To get the code onto the BBB you can either:

- Git Clone from command -- With port forwarding internet access or wifi enabled on the BBB, you can SSH into the BBB and do a git clone.
- SCP -- Run this command on your mac with your IP address. See SSH into BBB in the Appendix for more info on getting the IP Address.

```
scp sonar.py debian@192.168.7.1
```

Once the file `sonar.py` is on the BBB you can run the file by SSH via the BBB:

```
sudo python sonar.py
```

You should see distances from the sonar sensor measure in cm. Flat objects will display better results. Try to put a hand or breadboard in front of the sensor and move it around. Can you change the code to make the distance in inches? Can you change how frequently the sonar calculation is done? Try cutting the time between sonar bursts in half?

If this does not work try looking through the Troubleshooting section. You can also contact me as the author of the kata, Bryan.Kulczycki@accenture.com. Also connect with Magnus Stahre and Mark Protas who helped me with this project.

Troubleshooting

No IP Address when connected BBB to computer

This may happen when the dongle used to connect the USB is in cooperation with the BBB. Try a new dongle.

Power troubles

For this Kata, the BBB will need to be connected to a 5V AC Adaptor. Power through the microUSB is not enough. You will not get a signal back from the sensor if the power is too low.

Be careful with print statements

I had put in a print statement to the console to troubleshoot why I wasn't getting a signal back. Apparently, the print statement was taking time to get to the screen. The process is single threaded, so by the time the print statement had made it to the console, the code had moved ahead and processed. When I took the print statement out the code worked fine.

Internet Connectivity on BBB

If you are getting errors like "Failed to fetch
`http://httpredir.debian.org/debian/dists/jessie/InRelease`" then run these commands on your BBB to add a line to your `resolv.conf` on the BBB

```
sudo touch /etc/resolv.conf
sudo vi /etc/resolv.conf
```

> add the following to the `resolv.conf` file:

```
debian@beaglebone:~$ echo "nameserver 8.8.8.8" | sudo tee -a  
/etc/resolv.conf  
nameserver 8.8.8.8
```

Connection Issues:

Your system date could also cause you get connections refused in apt-get.

`sudo date -s "2019-05-13"` can be used to set the system date. Make sure to **use the current date**.

Appendix I Internet Access

To get internet access on a non-wifi enabled beaglebone

Make sure your BBB has network connectivity.

For troubleshooting connectivity, this section provides suggestions.

If you don't need help with connectivity, skip this section.

Reference on stack overflow: [How do I share my mac's internet connection with my beaglebone black?](#)

Backup your `pf.conf` file (mac) to make sure you can roll back your network settings, if needed

```
sudo cp /etc/pf.conf /etc/pf.conf.bck
```

On your machine, find the name of the network interface for your BBB by searching with `ifconfig`, for the IP address used by BBB. Example:

```
ifconfig | grep -C 5 192.168.6.1
```

If your IP address was 192.168.7.1, then you would use this IP address instead of 192.168.6.1.

Turn on port forwarding on your machine:

```
sudo sysctl net.inet.ip.forwarding=1
```

Forward traffic from BBB to the internet on your machine

```
echo "nat on en0 from en7:network to any -> (en0)" | sudo pfctl -f --e
```

Where en7 is the network interface name you obtained from ifconfig grep.

On the BBB:

```
sudo route add default gw 192.168.6.1 usb1
```

Where usb1 is the network interface that matches up with your ssh IP address, when you run ifconfig on the BBB.

This should give you an internet connection. Verify connectivity with the instructions below.

Setup Wifi on BBB w/ Wifi

To connect to your Wireless network type the following command in the terminal window:

```
sudo connmanctl
connmanctl> enable wifi
Enabled wifi
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AO TNCPA97AB9
wifi_506583d4fc5e_544e434150413937414239_managed_psk
wifi_506583d4fc5e_hidden_managed_psk
DIRECT-roku-876
wifi_506583d4fc5e_4449524543542d726f6b752d383736_managed_psk
BTHub6-H5H7
wifi_506583d4fc5e_4254487562362d48354837_managed_psk
virginmedia2029431
wifi_506583d4fc5e_76697267696e6d6564696132303239343331_managed_psk
VM046693-2G
wifi_506583d4fc5e_564d3034363639332d3247_managed_psk
BTWifi-with-FON
wifi_506583d4fc5e_425457696692d776974682d464f4e_managed_none
connmanctl> agent on
Agent registered
connmanctl> connect
wifi_506583d4fc5e_544e434150413937414239_managed_psk
Passphrase? xxxxxxxxxxxx
```

```
connected wifi_506583d4fc5e_544e434150413937414239_managed_psk  
connmanctl> quit
```

You should now be connected to your local wifi. You can check that you have an IP address by typing the following in the terminal window:

```
ifconfig -a
```

You should now see an IP address under `Wlan0` and you can now connect to this IP address in the future to program your BeagleBone Black Wireless.

Confirm your network connectivity

Run the following command while ssh into your beaglebone:

```
ping 8.8.8.8
```

You should see packets returned with a ms time that is taken to return.

If you are getting errors like "Failed to fetch
`http://httpredir.debian.org/debian/dists/jessie/InRelease`"
then do this before trying to update again:

On the BBB run these commands:

```
sudo touch /etc/resolv.conf
```

```
sudo vi /etc/resolv.conf
```

> add the following to the `resolv.conf` file:

```
nameserver 8.8.8.8
```

Your system date could also cause you get connections refused in apt-get
`sudo date -s "2019-05-17"`

Can be used to set the system date. Use the current date.

Appendix II SSH to your BBB over USB

Installing Mac Drivers

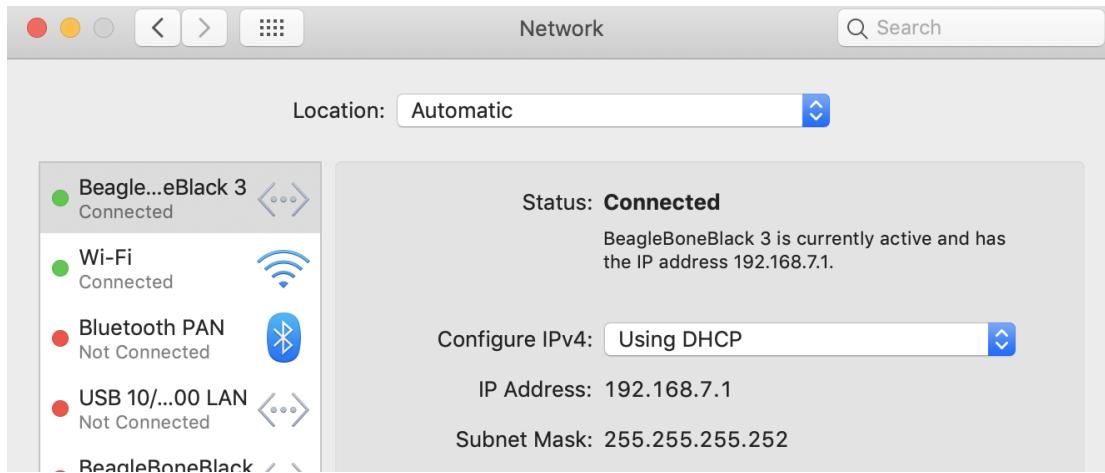
For the BBB to be able to do ad hoc networking with your computer, there is nothing to install on the BBB itself. It's all set up and ready to go. However, to be able to run this on a Mac, you will need to install the network driver which is here:

<http://beagleboard.org/static/Drivers/MacOSX/RNDIS/HoRNDIS.pkg>

Once the driver is installed open Network from System Preferences and you should see your BBB showing up as a network connection.

To SSH into your BBB, make sure your BBB is connected to your computer: connect the micro usb port on the BBB to the usb port using the correct dongle on your mac. After connecting,

you should be able to see the IP address, 192.168.7.1, for your BBB under System Preferences
-> Network



Opening a terminal you should be able to `ssh debian@192.168.7.1`

When asked for the password use `temppwd`.

This should give you a command line to run commands on the debian.

```
I ~ => ssh debian@192.168.7.2
The authenticity of host '192.168.7.2 (192.168.7.2)' can't be established.
ECDSA key fingerprint is SHA256:+D+YtzxWhwSy/afBfhoX4UI0Kx1hNKv3zsobYK5aWJ4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.7.2' (ECDSA) to the list of known hosts.
debian@192.168.7.2's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Oct  8 16:17:42 2018
debian@beaglebone:~$
```