EECS 4221:
Operating System Design
Winter 2019

Alternative Programming Assignment 1
BeagleBone Black and Angstrom Linux

Niruyan Rakulan
214343438
Electrical Engineering Undergraduate

Table of Contents

A. Features of Beaglebone Black and Angstrom

Most Important Features of BeagleBone Black

The BeagleBone Black Board is a relatively low-cost development board. It features a Texas Instruments Sitara AM335x Cortex A8 ARM microprocessor (running at 1GHz), 512MB of DDR3 ram, 4KB of EEPROM, and 4GB of eMMC storage(Rev C).The board features 92 Pins(46 on header P8 and 46 on header P9), many of which are GPIOs allowing to be interfaced with sensors (ie. photoresistor) and actuators (ie. motor); the modes of the pins can be changed to serve other purposes such as ADC, and PWM.

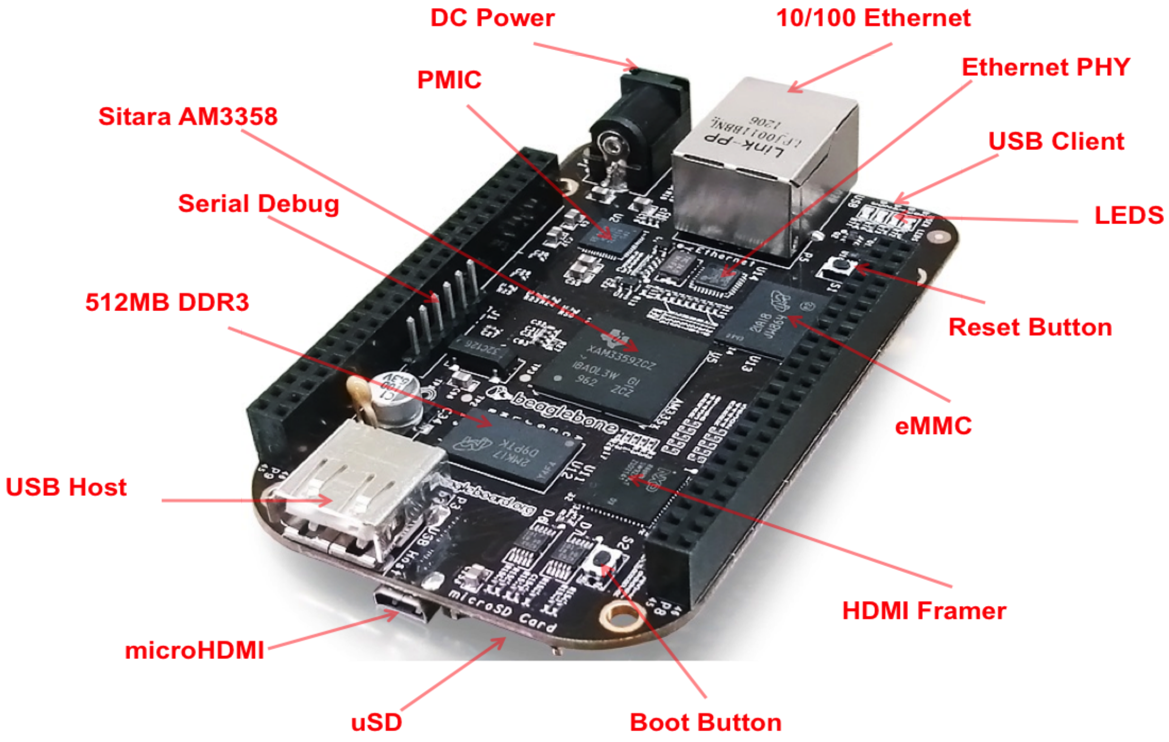
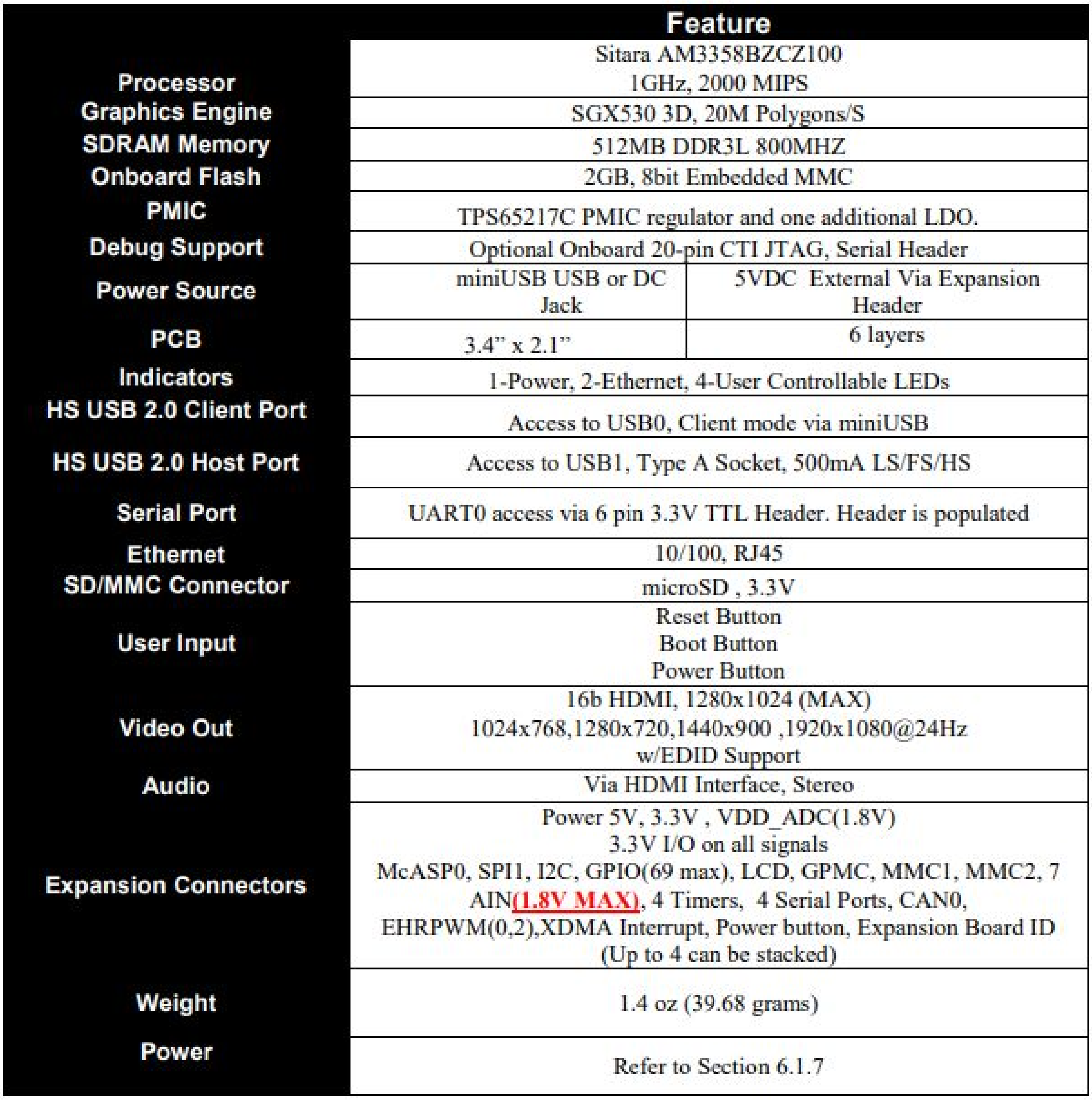


Figure 1: BeagleBone Black board layout. From http://beagleboard.org/Support/bone101

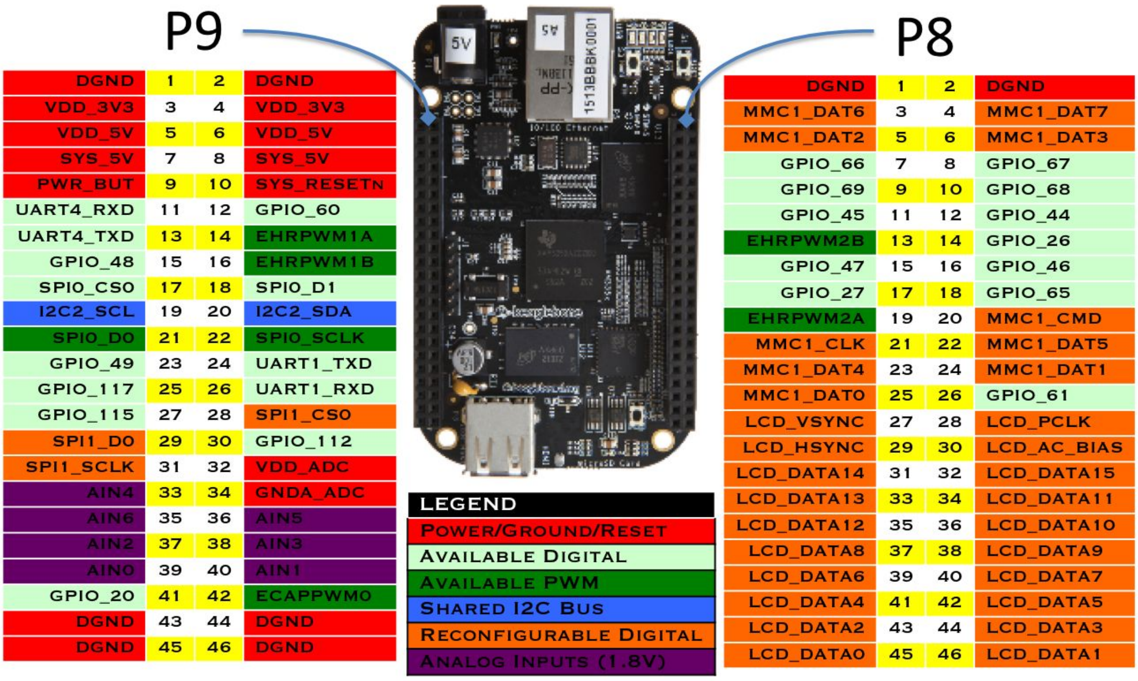
The BeagleBone can either be interfaced thru a tethered computer, or as a standalone desktop (with the board attached to a monitor thru a HDMI cable); this report will use the tethered approach to a Windows 10 computer. The tethered approach powers the board, as well as allows the user to use their PC to interface with the board.



	Feature	
Processor	Sitara AM3358BZCZ100 1GHz, 2000 MIPS	
Graphics Engine	SGX530 3D, 20M Polygons/S	
SDRAM Memory	512MB DDR3L 800MHZ	
Onboard Flash	2GB, 8bit Embedded MMC	
PMIC	TPS65217C PMIC regulator and one additional LDO.	
Debug Support	Optional Onboard 20-pin CTI JTAG, Serial Header	
Power Source	miniUSB USB or DC Jack	5VDC External Via Expansion Header
PCB	3.4" x 2.1"	6 layers
Indicators	1-Power, 2-Ethernet, 4-User Controllable LEDs	
HS USB 2.0 Client Port	Access to USB0, Client mode via miniUSB	
HS USB 2.0 Host Port	Access to USB1, Type A Socket, 500mA LS/FS/HS	
Serial Port	UART0 access via 6 pin 3.3V TTL Header. Header is populated	
Ethernet	10/100, RJ45	
SD/MMC Connector	microSD , 3.3V	
User Input	Reset Button Boot Button Power Button	
Video Out	16b HDMI, 1280x1024 (MAX) 1024x768,1280x720,1440x900 ,1920x1080@24Hz w/EDID Support	
Audio	Via HDMI Interface, Stereo	
Expansion Connectors	Power 5V, 3.3V , VDD_ADC(1.8V) 3.3V I/O on all signals McASP0, SPI1, I2C, GPIO(69 max), LCD, GPMC, MMC1, MMC2, 7 AIN(1.8V MAX), 4 Timers, 4 Serial Ports, CAN0, EHRPWM(0,2),XDMA Interrupt, Power button, Expansion Board ID (Up to 4 can be stacked)	
Weight	1.4 oz (39.68 grams)	
Power	Refer to Section 6.1.7	

Figure 2: BeagleBone Black specs. From https://media.digikey.com/pdf/Data%20Sheets/Circuitco%20Elect/BB-BBLK-000%20Manual.pdf

Cape Expansion Headers



P9

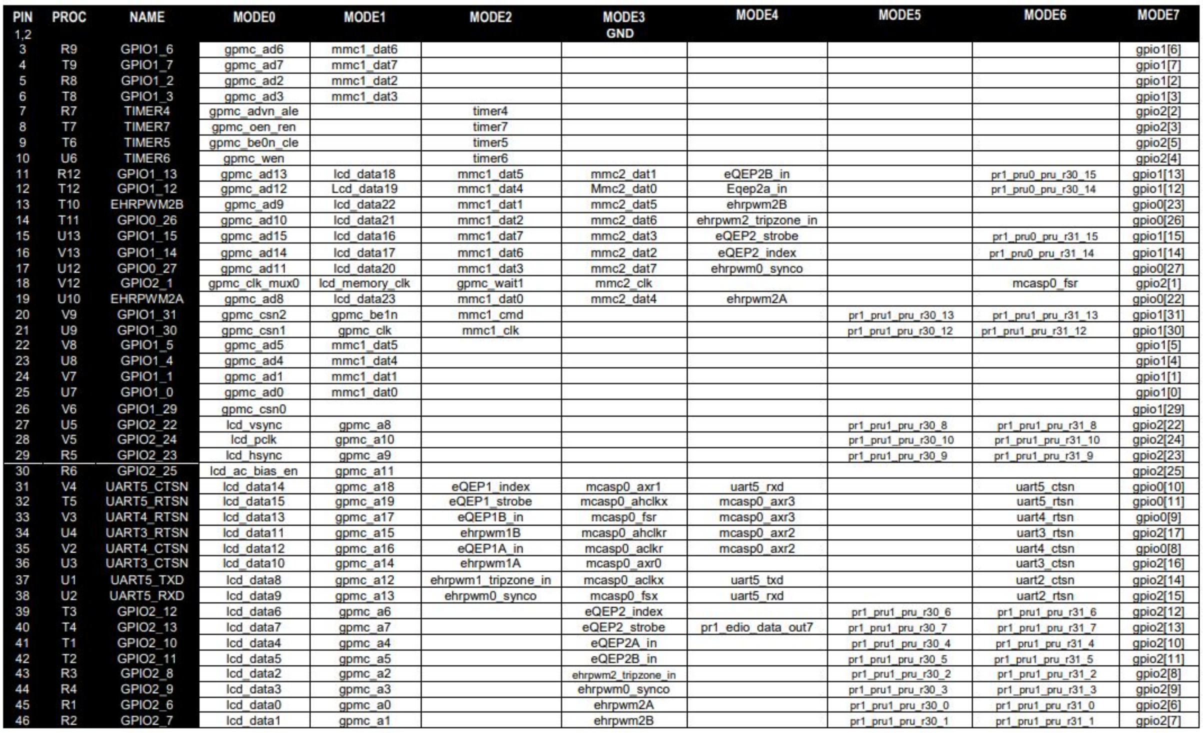
DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETn
UART4_RXD	11	12	GPIO_60
UART4_TXD	13	14	EHRPWM1A
GPIO_48	15	16	EHRPWM1B
SPI0_CS0	17	18	SPI0_D1
I2C2_SCL	19	20	I2C2_SDA
SPI0_D0	21	22	SPI0_SCLK
GPIO_49	23	24	UART1_TXD
GPIO_117	25	26	UART1_RXD
GPIO_115	27	28	SPI1_CS0
SPI1_D0	29	30	GPIO_112
SPI1_SCLK	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	ECAPPWM0
DGND	43	44	DGND
DGND	45	46	DGND

P8

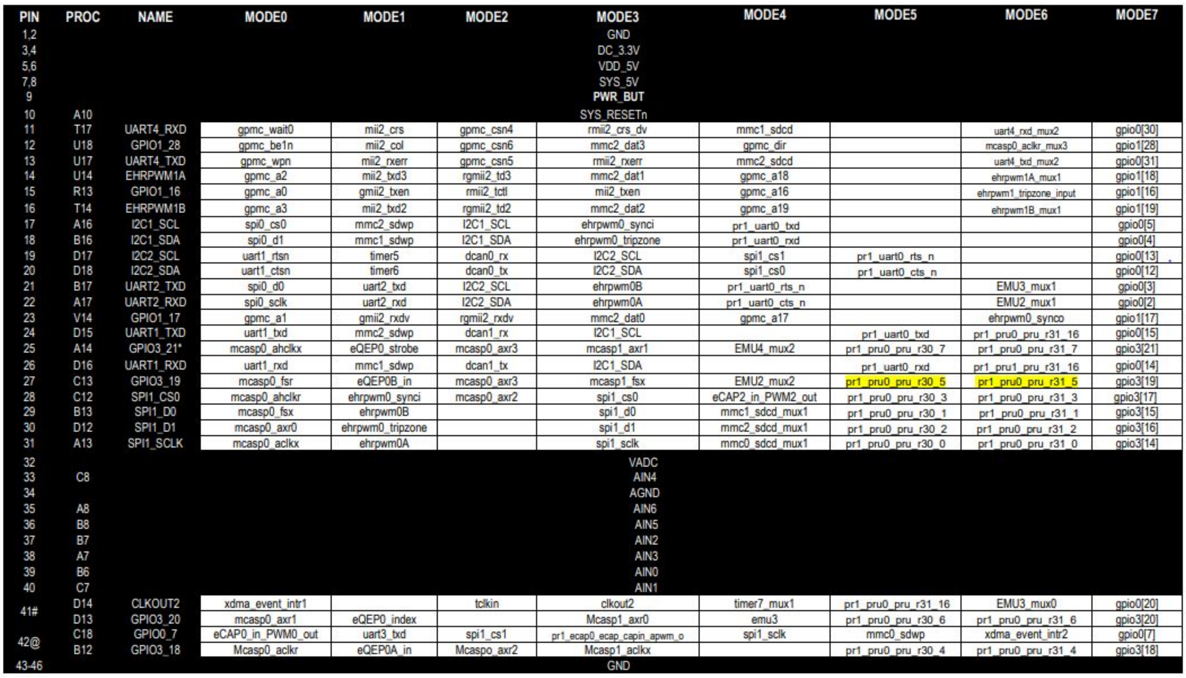
DGND	1	2	DGND
MMC1_DAT6	3	4	MMC1_DAT7
MMC1_DAT2	5	6	MMC1_DAT3
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
EHRPWM2B	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
EHRPWM2A	19	20	MMC1_CMD
MMC1_CLK	21	22	MMC1_DAT5
MMC1_DAT4	23	24	MMC1_DAT1
MMC1_DAT0	25	26	GPIO_61
LCD_VSYNC	27	28	LCD_PCLK
LCD_HSYNC	29	30	LCD_AC_BIAS
LCD_DATA14	31	32	LCD_DATA15
LCD_DATA13	33	34	LCD_DATA11
LCD_DATA12	35	36	LCD_DATA10
LCD_DATA8	37	38	LCD_DATA9
LCD_DATA6	39	40	LCD_DATA7
LCD_DATA4	41	42	LCD_DATA5
LCD_DATA2	43	44	LCD_DATA3
LCD_DATA0	45	46	LCD_DATA1

LEGEND
- Power/Ground/Reset
- Available Digital
- Available PWM
- Shared I2C Bus
- Reconfigurable Digital
- Analog Inputs (1.8V)

Figure 3: Pin layout of board. From http://beagleboard.org/Support/bone101



PIN	PROC	NAME	MODE0	MODE1	MODE2	MODE3	MODE4	MODE5	MODE6	MODE7
1,2						GND				
3	R9	GPIO1_6	gpmc_ad6	mmc1_dat6						gpio1[6]
4	T9	GPIO1_7	gpmc_ad7	mmc1_dat7						gpio1[7]
5	R8	GPIO1_2	gpmc_ad2	mmc1_dat2						gpio1[2]
6	T8	GPIO1_3	gpmc_ad3	mmc1_dat3						gpio1[3]
7	R7	TIMER4	gpmc_advn_ale		timer4					gpio2[2]
8	T7	TIMER7	gpmc_oen_ren		timer7					gpio2[3]
9	T6	TIMER5	gpmc_be0n_cle		timer5					gpio2[5]
10	U6	TIMER6	gpmc_wen		timer6					gpio2[4]
11	R12	GPIO1_13	gpmc_ad13	lcd_data18	mmc1_dat5	mmc2_dat1	eQEP2B_in		pr1_pru0_pru_r30_15	gpio1[13]
12	T12	GPIO1_12	gpmc_ad12	Lcd_data19	mmc1_dat4	Mmc2_dat0	Eqep2a_in		pr1_pru0_pru_r30_14	gpio1[12]
13	T10	EHRPWM2B	gpmc_ad9	lcd_data22	mmc1_dat1	mmc2_dat5	ehrpwm2B			gpio0[23]
14	T11	GPIO0_26	gpmc_ad10	lcd_data21	mmc1_dat2	mmc2_dat6	ehrpwm2_tripzone_in			gpio0[26]
15	U13	GPIO1_15	gpmc_ad15	lcd_data16	mmc1_dat7	mmc2_dat3	eQEP2_strobe		pr1_pru0_pru_r31_15	gpio1[15]
16	V13	GPIO1_14	gpmc_ad14	lcd_data17	mmc1_dat6	mmc2_dat2	eQEP2_index		pr1_pru0_pru_r31_14	gpio1[14]
17	U12	GPIO0_27	gpmc_ad11	lcd_data20	mmc1_dat3	mmc2_dat7	ehrpwm0_synco			gpio0[27]
18	V12	GPIO2_1	gpmc_clk_mux0	lcd_memory_clk	gpmc_wait1	mmc2_clk			mcasp0_fsr	gpio2[1]
19	U10	EHRPWM2A	gpmc_ad8	lcd_data23	mmc1_dat0	mmc2_dat4	ehrpwm2A			gpio0[22]
20	V9	GPIO1_31	gpmc_csn2	gpmc_be1n	mmc1_cmd			pr1_pru1_pru_r30_13	pr1_pru1_pru_r31_13	gpio1[31]
21	U9	GPIO1_30	gpmc_csn1	gpmc_clk	mmc1_clk			pr1_pru1_pru_r30_12	pr1_pru1_pru_r31_12	gpio1[30]
22	V8	GPIO1_5	gpmc_ad5	mmc1_dat5						gpio1[5]
23	U8	GPIO1_4	gpmc_ad4	mmc1_dat4						gpio1[4]
24	V7	GPIO1_1	gpmc_ad1	mmc1_dat1						gpio1[1]
25	U7	GPIO1_0	gpmc_ad0	mmc1_dat0						gpio1[0]
26	V6	GPIO1_29	gpmc_csn0							gpio1[29]
27	U5	GPIO2_22	lcd_vsync	gpmc_a8				pr1_pru1_pru_r30_8	pr1_pru1_pru_r31_8	gpio2[22]
28	V5	GPIO2_24	lcd_pclk	gpmc_a10				pr1_pru1_pru_r30_10	pr1_pru1_pru_r31_10	gpio2[24]
29	R5	GPIO2_23	lcd_hsync	gpmc_a9				pr1_pru1_pru_r30_9	pr1_pru1_pru_r31_9	gpio2[23]
30	R6	GPIO2_25	lcd_ac_bias_en	gpmc_a11						gpio2[25]
31	V4	UART5_CTSN	lcd_data14	gpmc_a18	eQEP1_index	mcasp0_axr1	uart5_rxd		uart5_ctsn	gpio0[10]
32	T5	UART5_RTSN	lcd_data15	gpmc_a19	eQEP1_strobe	mcasp0_ahclkx	mcasp0_axr3		uart5_rtsn	gpio0[11]
33	V3	UART4_RTSN	lcd_data13	gpmc_a17	eQEP1B_in	mcasp0_fsr	mcasp0_axr3		uart4_rtsn	gpio0[9]
34	U4	UART3_RTSN	lcd_data11	gpmc_a15	ehrpwm1B	mcasp0_ahclkr	mcasp0_axr2		uart3_rtsn	gpio2[17]
35	V2	UART4_CTSN	lcd_data12	gpmc_a16	eQEP1A_in	mcasp0_aclkr	mcasp0_axr2		uart4_ctsn	gpio0[8]
36	U3	UART3_CTSN	lcd_data10	gpmc_a14	ehrpwm1A	mcasp0_axr0			uart3_ctsn	gpio2[16]
37	U1	UART5_TXD	lcd_data8	gpmc_a12	ehrpwm1_tripzone_in	mcasp0_aclkx	uart5_txd		uart2_ctsn	gpio2[14]
38	U2	UART5_RXD	lcd_data9	gpmc_a13	ehrpwm0_synco	mcasp0_fsx	uart5_rxd		uart2_rtsn	gpio2[15]
39	T3	GPIO2_12	lcd_data6	gpmc_a6		eQEP2_index		pr1_pru1_pru_r30_6	pr1_pru1_pru_r31_6	gpio2[12]
40	T4	GPIO2_13	lcd_data7	gpmc_a7		eQEP2_strobe	pr1_edio_data_out7	pr1_pru1_pru_r30_7	pr1_pru1_pru_r31_7	gpio2[13]
41	T1	GPIO2_10	lcd_data4	gpmc_a4		eQEP2A_in		pr1_pru1_pru_r30_4	pr1_pru1_pru_r31_4	gpio2[10]
42	T2	GPIO2_11	lcd_data5	gpmc_a5		eQEP2B_in		pr1_pru1_pru_r30_5	pr1_pru1_pru_r31_5	gpio2[11]
43	R3	GPIO2_8	lcd_data2	gpmc_a2		ehrpwm2_tripzone_in		pr1_pru1_pru_r30_2	pr1_pru1_pru_r31_2	gpio2[8]
44	R4	GPIO2_9	lcd_data3	gpmc_a3		ehrpwm0_synco		pr1_pru1_pru_r30_3	pr1_pru1_pru_r31_3	gpio2[9]
45	R1	GPIO2_6	lcd_data0	gpmc_a0		ehrpwm2A		pr1_pru1_pru_r30_0	pr1_pru1_pru_r31_0	gpio2[6]
46	R2	GPIO2_7	lcd_data1	gpmc_a1		ehrpwm2B		pr1_pru1_pru_r30_1	pr1_pru1_pru_r31_1	gpio2[7]



PIN	PROC	NAME	MODE0	MODE1	MODE2	MODE3	MODE4	MODE5	MODE6	MODE7
1,2						GND				
3,4						DC_3.3V				
5,6						VDD_5V				
7,8						SYS_5V				
9						**PWR_BUT**				
10	A10					SYS_RESETn				
11	T17	UART4_RXD	gpmc_wait0	mii2_crs	gpmc_csn4	rmii2_crs_dv	mmc1_sdcd		uart4_rxd_mux2	gpio0[30]
12	U18	GPIO1_28	gpmc_be1n	mii2_col	gpmc_csn6	mmc2_dat3	gpmc_dir		mcasp0_aclkr_mux3	gpio1[28]
13	U17	UART4_TXD	gpmc_wpn	mii2_rxerr	gpmc_csn5	rmii2_rxerr	mmc2_sdcd		uart4_txd_mux2	gpio0[31]
14	U14	EHRPWM1A	gpmc_a2	mii2_txd3	rgmii2_td3	mmc2_dat1	gpmc_a18		ehrpwm1A_mux1	gpio1[18]
15	R13	GPIO1_16	gpmc_a0	gmii2_txen	rmii2_tctl	mii2_txen	gpmc_a16		ehrpwm1_tripzone_input	gpio1[16]
16	T14	EHRPWM1B	gpmc_a3	mii2_txd2	rgmii2_td2	mmc2_dat2	gpmc_a19		ehrpwm1B_mux1	gpio1[19]
17	A16	I2C1_SCL	spi0_cs0	mmc2_sdwp	I2C1_SCL	ehrpwm0_synci	pr1_uart0_txd			gpio0[5]
18	B16	I2C1_SDA	spi0_d1	mmc1_sdwp	I2C1_SDA	ehrpwm0_tripzone	pr1_uart0_rxd			gpio0[4]
19	D17	I2C2_SCL	uart1_rtsn	timer5	dcan0_rx	I2C2_SCL	spi1_cs1	pr1_uart0_rts_n		gpio0[13]
20	D18	I2C2_SDA	uart1_ctsn	timer6	dcan0_tx	I2C2_SDA	spi1_cs0	pr1_uart0_cts_n		gpio0[12]
21	B17	UART2_TXD	spi0_d0	uart2_txd	I2C2_SCL	ehrpwm0B	pr1_uart0_rts_n		EMU3_mux1	gpio0[3]
22	A17	UART2_RXD	spi0_sclk	uart2_rxd	I2C2_SDA	ehrpwm0A	pr1_uart0_cts_n		EMU2_mux1	gpio0[2]
23	V14	GPIO1_17	gpmc_a1	gmii2_rxdv	rgmii2_rxdv	mmc2_dat0	gpmc_a17		ehrpwm0_synco	gpio1[17]
24	D15	UART1_TXD	uart1_txd	mmc2_sdwp	dcan1_rx	I2C1_SCL		pr1_uart0_txd	pr1_pru0_pru_r31_16	gpio0[15]
25	A14	GPIO3_21*	mcasp0_ahclkx	eQEP0_strobe	mcasp0_axr3	mcasp1_axr1	EMU4_mux2	pr1_pru0_pru_r30_7	pr1_pru0_pru_r31_7	gpio3[21]
26	D16	UART1_RXD	uart1_rxd	mmc1_sdwp	dcan1_tx	I2C1_SDA		pr1_uart0_rxd	pr1_pru1_pru_r31_16	gpio0[14]
27	C13	GPIO3_19	mcasp0_fsr	eQEP0B_in	mcasp0_axr3	mcasp1_fsx	EMU2_mux2	pr1_pru0_pru_r30_5	pr1_pru0_pru_r31_5	gpio3[19]
28	C12	SPI1_CS0	mcasp0_ahclkr	ehrpwm0_synci	mcasp0_axr2	spi1_cs0	eCAP2_in_PWM2_out	pr1_pru0_pru_r30_3	pr1_pru0_pru_r31_3	gpio3[17]
29	B13	SPI1_D0	mcasp0_fsx	ehrpwm0B		spi1_d0	mmc1_sdcd_mux1	pr1_pru0_pru_r30_1	pr1_pru0_pru_r31_1	gpio3[15]
30	D12	SPI1_D1	mcasp0_axr0	ehrpwm0_tripzone		spi1_d1	mmc2_sdcd_mux1	pr1_pru0_pru_r30_2	pr1_pru0_pru_r31_2	gpio3[16]
31	A13	SPI1_SCLK	mcasp0_aclkx	ehrpwm0A		spi1_sclk	mmc0_sdcd_mux1	pr1_pru0_pru_r30_0	pr1_pru0_pru_r31_0	gpio3[14]
32						VADC				
33	C8					AIN4				
34						AGND				
35	A8					AIN6				
36	B8					AIN5				
37	B7					AIN2				
38	A7					AIN3				
39	B6					AIN0				
40	C7					AIN1				
41#	D14	CLKOUT2	xdma_event_intr1		tclkin	clkout2	timer7_mux1	pr1_pru0_pru_r31_16	EMU3_mux0	gpio0[20]
	D13	GPIO3_20	mcasp0_axr1	eQEP0_index		Mcasp1_axr0	emu3	pr1_pru0_pru_r30_6	pr1_pru0_pru_r31_6	gpio3[20]
42@	C18	GPIO0_7	eCAP0_in_PWM0_out	uart3_txd	spi1_cs1	pr1_ecap0_ecap_capin_apwm_o	spi1_sclk	mmc0_sdwp	xdma_event_intr2	gpio0[7]
	B12	GPIO3_18	Mcasp0_aclkr	eQEP0A_in	Mcaspo_axr2	Mcasp1_aclkx		pr1_pru0_pru_r30_4	pr1_pru0_pru_r31_4	gpio3[18]
43-46						GND				

Figure 4: Expansion Header P8 and P9 Pinout. From https://media.digikey.com/pdf/Data%20Sheets/Circuitco%20Elect/BB-BBLK-000%20Manual.pdf

The above diagrams and charts are important to know when working with the BeagleBone Black Board. For more info, refer to the BeagleBone Black System Reference Manual from https://media.digikey.com/pdf/Data%20Sheets/Circuitco%20Elect/BB-BBLK-000%20Manual.pdf

Most Important Features of Angstrom

Angstrom is a Linux distribution primarily focused for embedded devices. Angstrom is a monolithic kernel suited for embedded devices given that resources are scarce on an embedded device. The distribution is open source, allowing for many to support the project; their github can be found at https://github.com/Angstrom-distribution. Angstrom uses opkg as its package manager; opkg is a lightweight package manager mainly found on embedded devices. Since Angstrom Os is a Linux Distribution, the features and advantages found in Linux are transferred over such as it being open source, lightweight, ease of use(gui or command line), and compatibility.

The OS will be flashed to the onboard flash memory. The flash file will be provided from https://beagleboard.org/latest-images. The latest version of Angstrom on the site is from 2013-09-04. BeagleBone Black has since transitioned from Angstrom to Debian; this is probably due to Debian being more popular and more supported. To learn more about Angstrom Distribution, you can visit: http://www.angstrom-distribution.org/

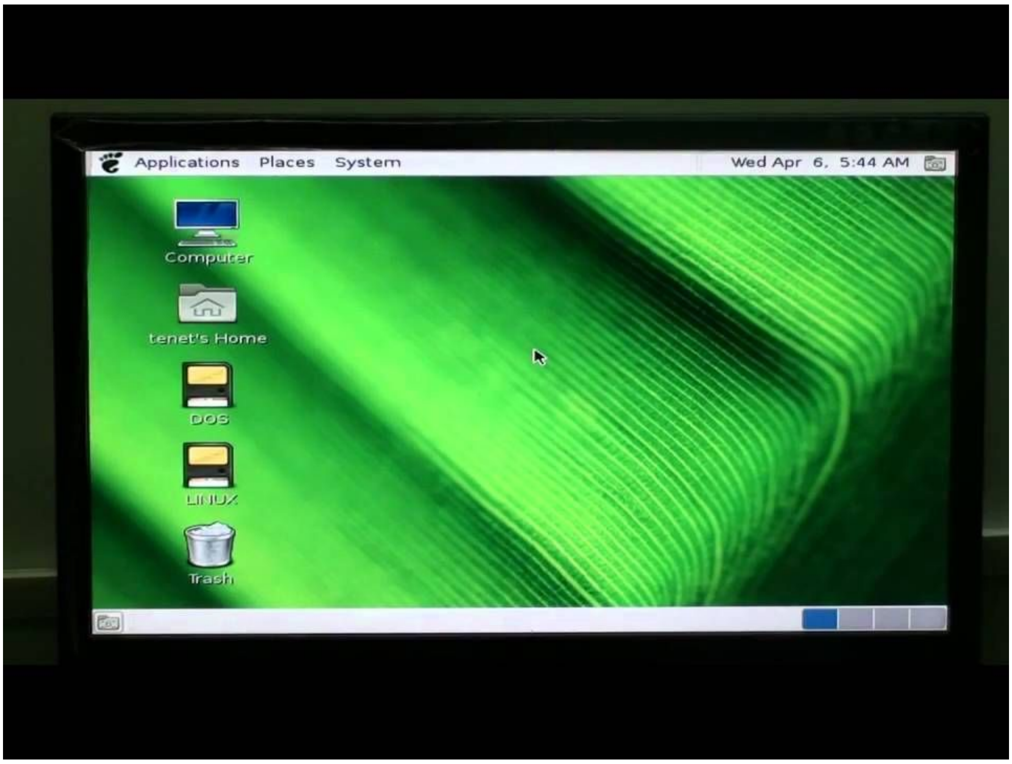
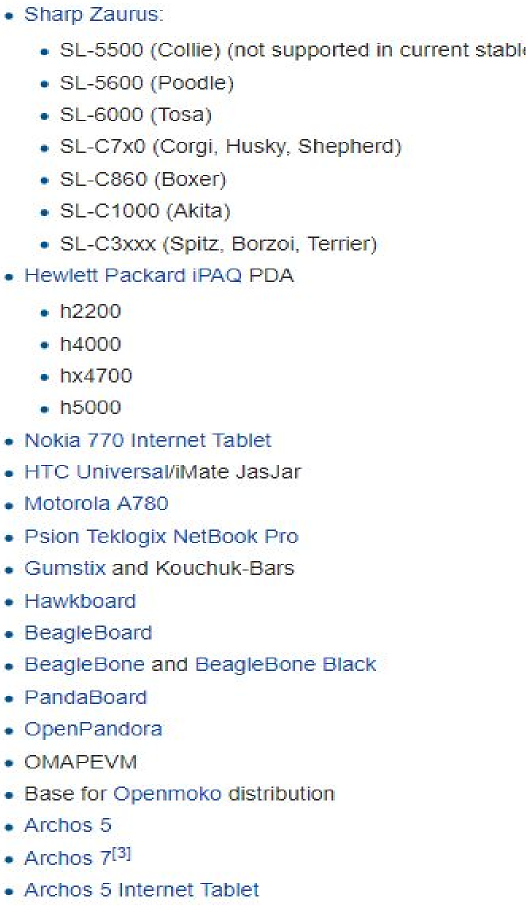


Figure 5: Image of Angstrom OS from https://www.youtube.com/watch?v=kyGSaHr00xo



- Sharp Zaurus:
  - SL-5500 (Collie) (not supported in current stabl
  - SL-5600 (Poodle)
  - SL-6000 (Tosa)
  - SL-C7x0 (Corgi, Husky, Shepherd)
  - SL-C860 (Boxer)
  - SL-C1000 (Akita)
  - SL-C3xxx (Spitz, Borzoi, Terrier)
- Hewlett Packard iPAQ PDA
  - h2200
  - h4000
  - hx4700
  - h5000
- Nokia 770 Internet Tablet
- HTC Universal/iMate JasJar
- Motorola A780
- Psion Teklogix NetBook Pro
- Gumstix and Kouchuk-Bars
- Hawkboard
- BeagleBoard
- BeagleBone and BeagleBone Black
- PandaBoard
- OpenPandora
- OMAPEVM
- Base for Openmoko distribution
- Archos 5
- Archos 7[3]
- Archos 5 Internet Tablet

Figure 5: List of devices that support Angstrom OS

B. Setting Up Beaglebone Black and Angstrom

Steps to Setup BeagleBone Black

Step 0: Board Safety

In an effort to minimize risk of damage to the board, the use of an anti-static wristband and an anti-static mat is highly encouraged to minimize the risk of Electrostatic Discharge.

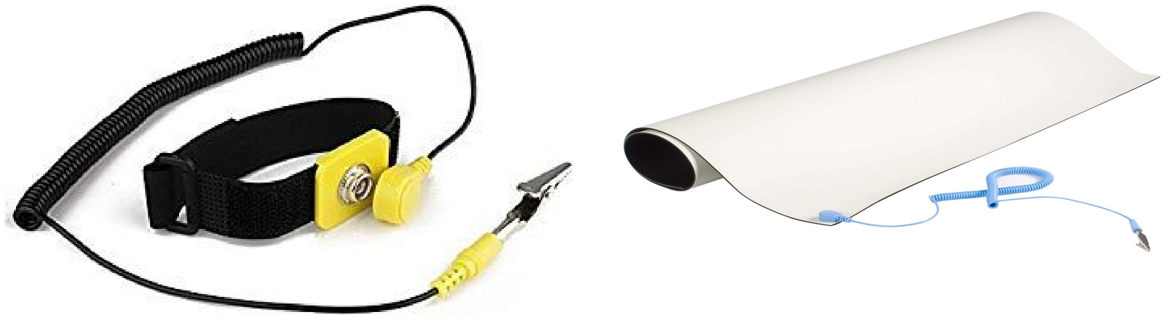


Figure 6: Use of anti-static wrist band, and anti-static mat encouraged to minimize board damage. From amazon.ca.

Before attaching any peripherals to the board, it is best to consult the BeagleBone Black System Reference Manual in case the current drawn, or voltage provided to the pins damages the board.

Step 1: Getting Started WIth The BeagleBone Board

This step is to ensure the board is in working order.

1. Using the mini-USB to USB cable provided, plug the BeagleBone Black board to your PC. Note: Plug the mini-USB part of the cable to the mini-USB port on the board, and not the USB Host port. You should note that the Power LED will glow a constant blue, and the USER LED 1 should blink in a “heartbeat” manner.

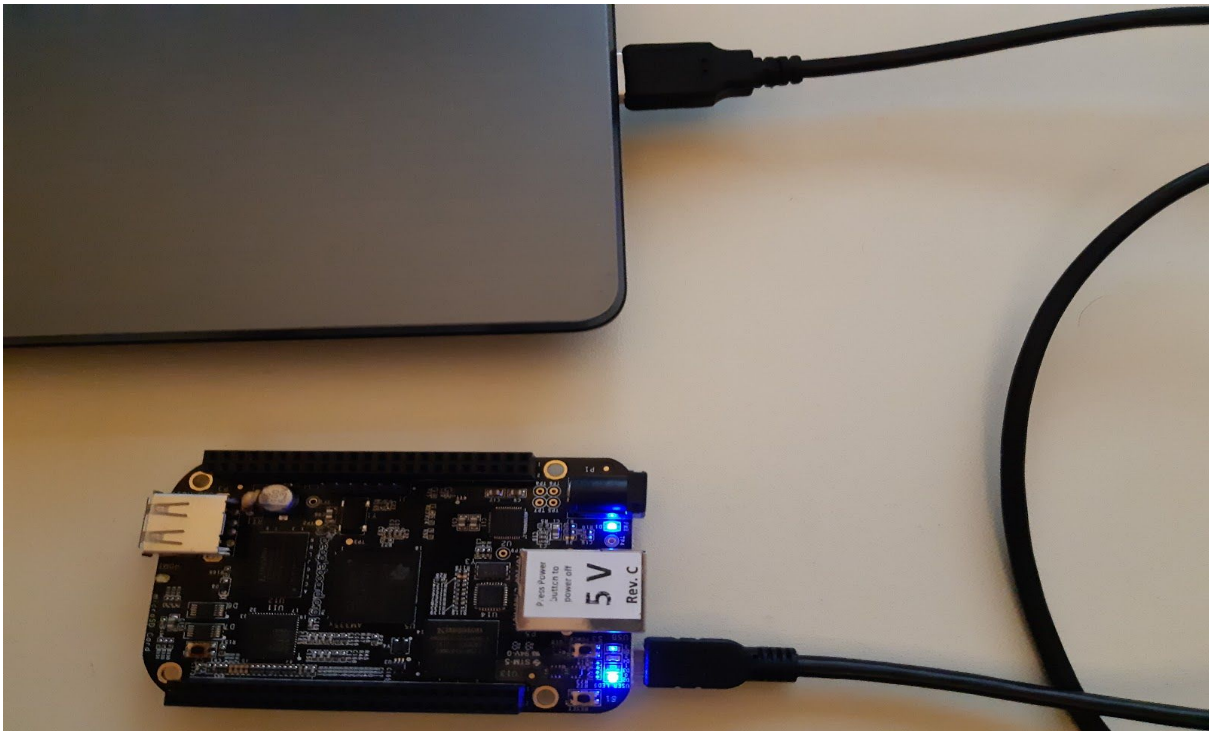
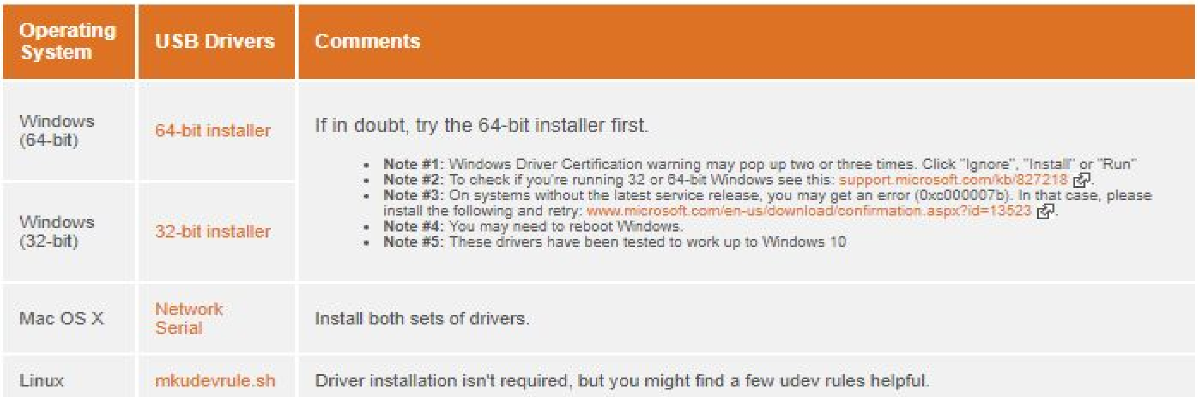


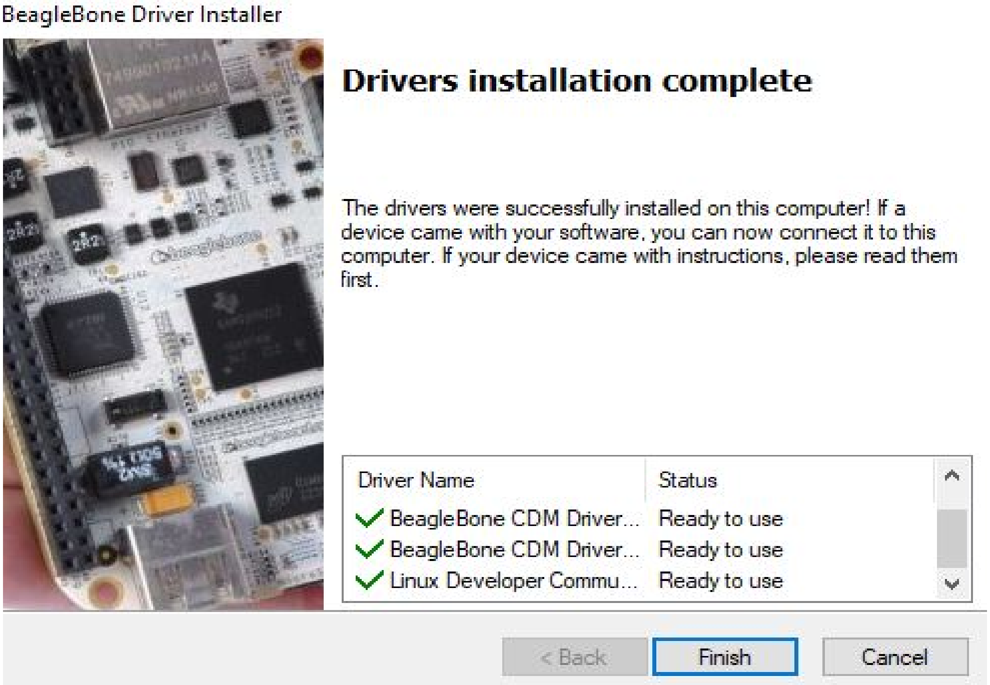
Figure 7: Tethering the Board to PC thru USB will LEDs glowing; power LED constant blue, and the USER LED will blink.

2.Then go to http://beagleboard.org/getting-started, and download the drivers needed for your system to enable network-over-USB access to your Beagle Board.



Operating System	USB Drivers	Comments
Windows (64-bit)	64-bit installer	If in doubt, try the 64-bit installer first. • **Note #1**: Windows Driver Certification warning may pop up two or three times. Click "Ignore", "Install" or "Run" • **Note #2**: To check if you're running 32 or 64-bit Windows see this: support.microsoft.com/kb/827218. • **Note #3**: On systems without the latest service release, you may get an error (0xc000007b). In that case, please install the following and retry: www.microsoft.com/en-us/download/confirmation.aspx?id=13523. • **Note #4**: You may need to reboot Windows. • **Note #5**: These drivers have been tested to work up to Windows 10
Windows (32-bit)	32-bit installer	
Mac OS X	Network Serial	Install both sets of drivers.
Linux	mkudevrule.sh	Driver installation isn't required, but you might find a few udev rules helpful.

Figure 8: Download drivers for your system to allow your computer to be able to communicate with the board.


BeagleBone Driver Installer

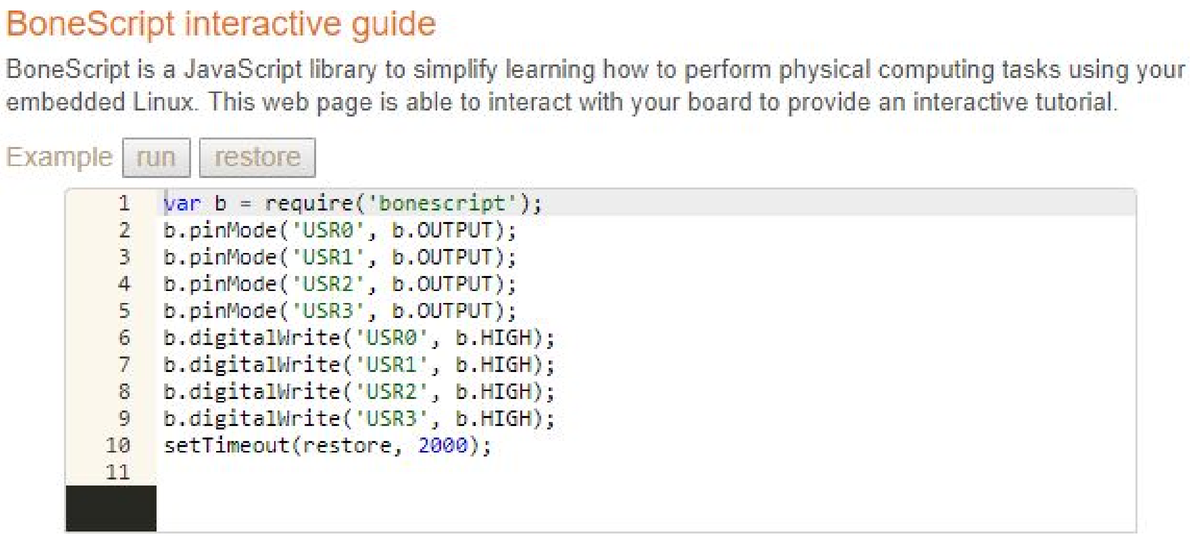
Drivers installation complete

The drivers were successfully installed on this computer! If a device came with your software, you can now connect it to this computer. If your device came with instructions, please read them first.

Driver Name	Status
✓ BeagleBone CDM Driver...	Ready to use
✓ BeagleBone CDM Driver...	Ready to use
✓ Linux Developer Commu...	Ready to use

Figure 9: Once installed, status should be “Ready To Use”.

3. Once the drivers are installed, using Chrome or Firefox, go to http://192.168.7.2. Once there, a sample BoneScript tutorial is available to be used (Figure 10); running the code will cause all the USER LEDs to light up momentarily(as shown in Figure 11).


BoneScript interactive guide

BoneScript is a JavaScript library to simplify learning how to perform physical computing tasks using your embedded Linux. This web page is able to interact with your board to provide an interactive tutorial.

```
var b = require('bonescript');
b.pinMode('USR0', b.OUTPUT);
b.pinMode('USR1', b.OUTPUT);
b.pinMode('USR2', b.OUTPUT);
b.pinMode('USR3', b.OUTPUT);
b.digitalWrite('USR0', b.HIGH);
b.digitalWrite('USR1', b.HIGH);
b.digitalWrite('USR2', b.HIGH);
b.digitalWrite('USR3', b.HIGH);
setTimeout(restore, 2000);

```

Figure 10: Sample BoneScript code provided by http://beagleboard.org/getting-started

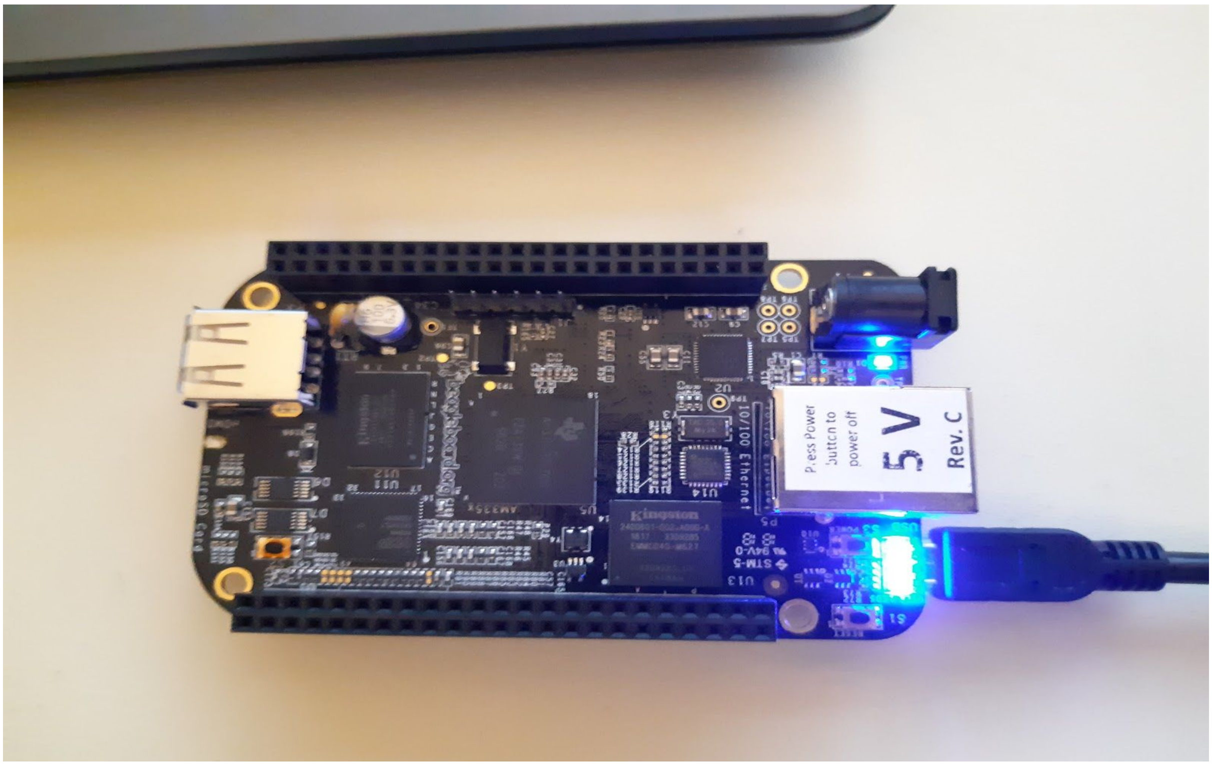
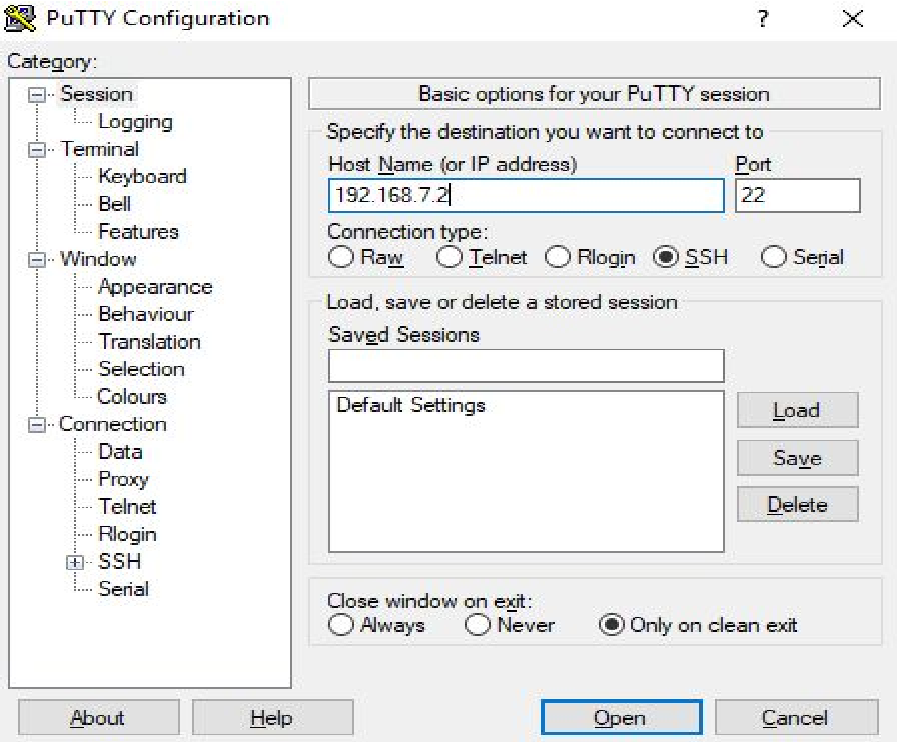


Figure 11: Once the code in the sample is run, the 4 USR LEDs will flash.

4. Communicating with board can be done through a terminal emulator such as Putty through SSH. The Host Name of the board is 192.168.7.2 (for Windows), with username “root”. The terminal displays the current OS of the board (Debian Image 2016-05-13).


PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.7.2

Port: 22

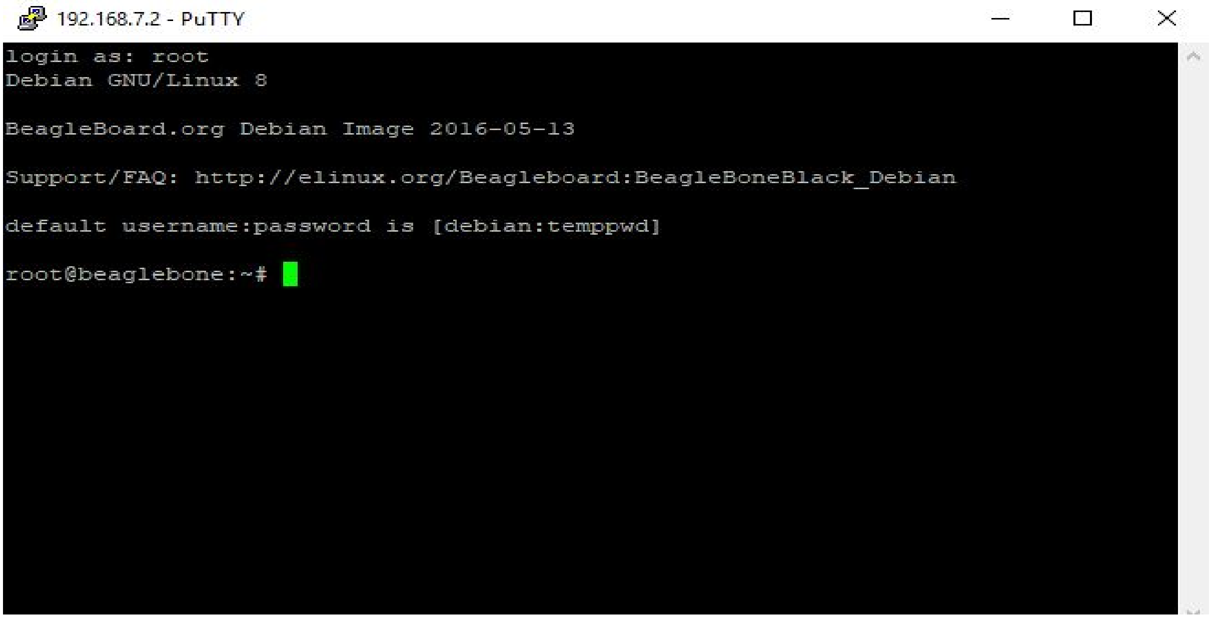
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

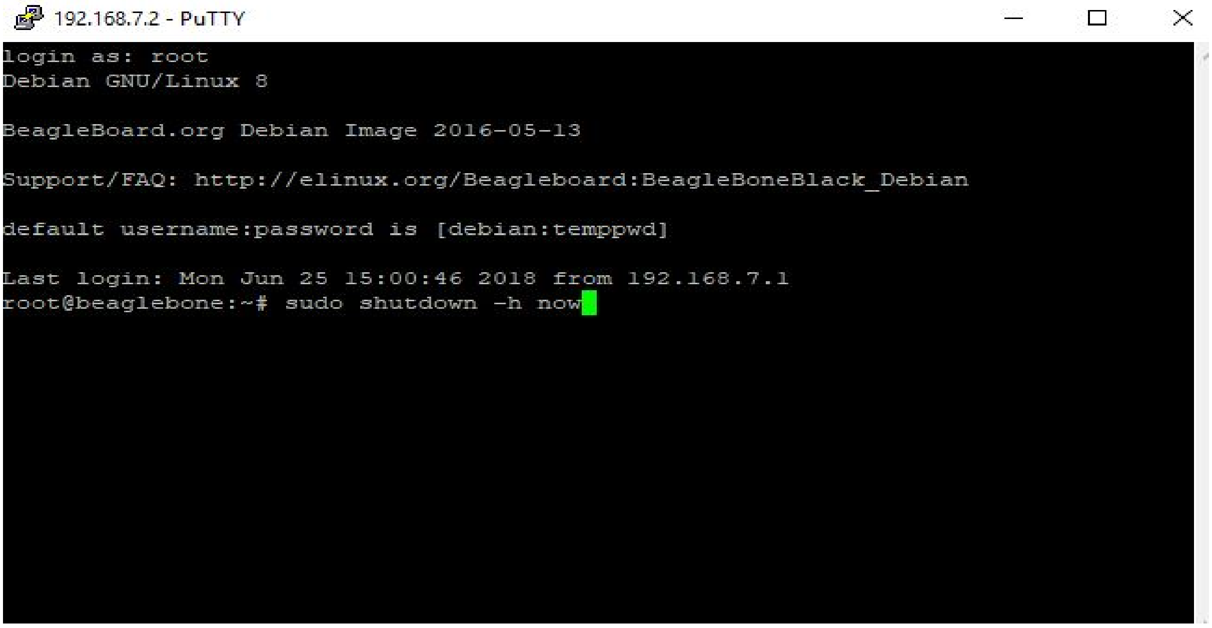


```
login as: root
Debian GNU/Linux 8

BeagleBoard.org Debian Image 2016-05-13

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

root@beaglebone:~#
```

Figure 12: The Putty interface

5. To turn off the board, it is not good practice to unplug the board in case it damages the board; to turn it off properly, you can either hold down the power button, or issue the command:

```
      shutdown -h now
```



```
login as: root
Debian GNU/Linux 8

BeagleBoard.org Debian Image 2016-05-13

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

Last login: Mon Jun 25 15:00:46 2018 from 192.168.7.1
root@beaglebone:~# sudo shutdown -h now
```

Figure 13: The shutdown command for the BeagleBoard.

Steps to Install Angstrom

Step 2: Installing Angstrom

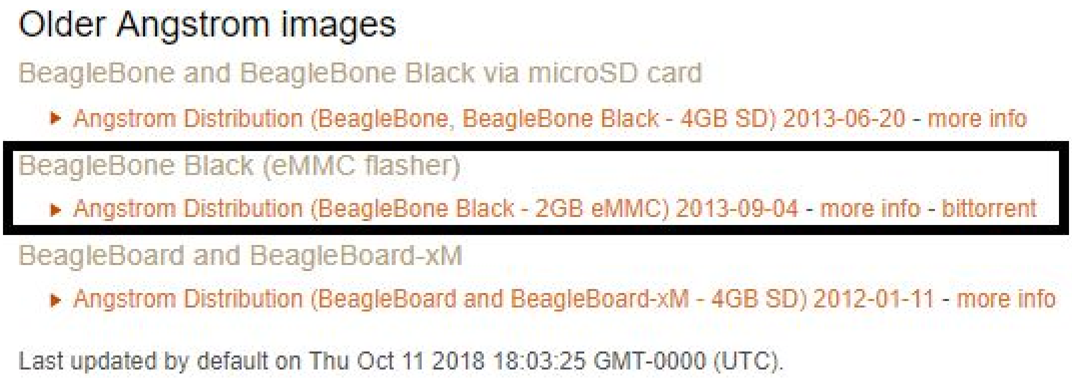
Housekeeping:
Things Needed

1. BeagleBone Black Board, and Mini-USB Cable
2. PC (with Micro-SD Card Reader) with internet access
3. Micro-SD Card
4. USB Wall Charger

1. Traverse to https://beagleboard.org/latest-images, and download:

BeagleBone Black (eMMC flasher) Angstrom Distribution (BeagleBone Black - 2GB eMMC) 2013-09-04
(as shown in Figure 14)

under Older Angstrom images to your PC. This version of Angstrom was chosen over the newer ones since the support for the Beaglebone Black board by the newer versions of Angstrom OS is not known.



Older Angstrom images

BeagleBone and BeagleBone Black via microSD card

- Angstrom Distribution (BeagleBone, BeagleBone Black - 4GB SD) 2013-06-20 - more info

BeagleBone Black (eMMC flasher)

- Angstrom Distribution (BeagleBone Black - 2GB eMMC) 2013-09-04 - more info - bittorrent

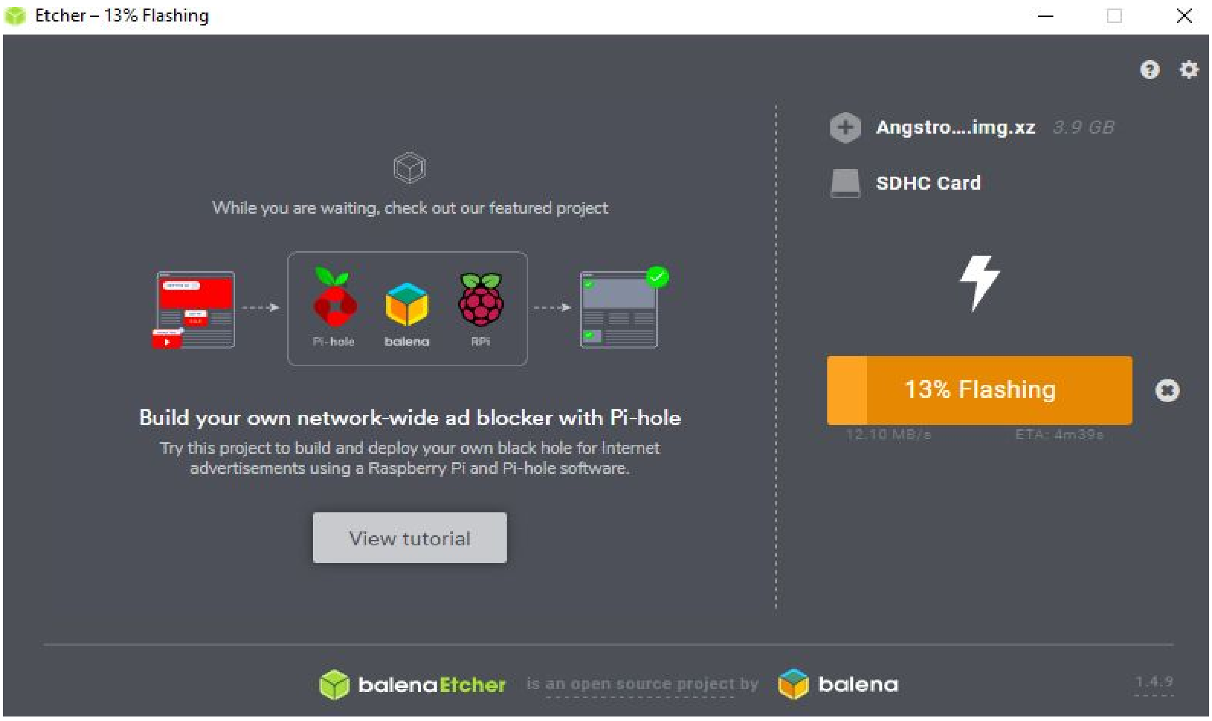
BeagleBoard and BeagleBoard-xM

- Angstrom Distribution (BeagleBoard and BeagleBoard-xM - 4GB SD) 2012-01-11 - more info

Last updated by default on Thu Oct 11 2018 18:03:25 GMT-0000 (UTC).

Figure 14: Download this version of Angstrom.

2. Download Etcher as suggested by http://beagleboard.org/getting-started; this application will flash your micro-SD card with the image file. Insert the micro-SD card into your PC (either natively or through an adapter). If prompted with a prompt to format the drive by Windows, ignore it. Then choose the correct image file, and correct drive to flash, and click on the “Flash!” Button. This process takes about 5 minutes.


Etcher – 13% Flashing

While you are waiting, check out our featured project

Build your own network-wide ad blocker with Pi-hole

Try this project to build and deploy your own black hole for Internet advertisements using a Raspberry Pi and Pi-hole software.

Angstro....img.xz 3.9 GB

SDHC Card

13% Flashing

12.10 MB/s ETA: 4m39s

Figure 15: The Etcher interface once Flash begins.

3. Disconnect any power (either from USB or 5V port) and any other peripherals attached to Beagleboard. Once the flashing has completed, remove the micro-SD Card from your PC and insert it into the board.

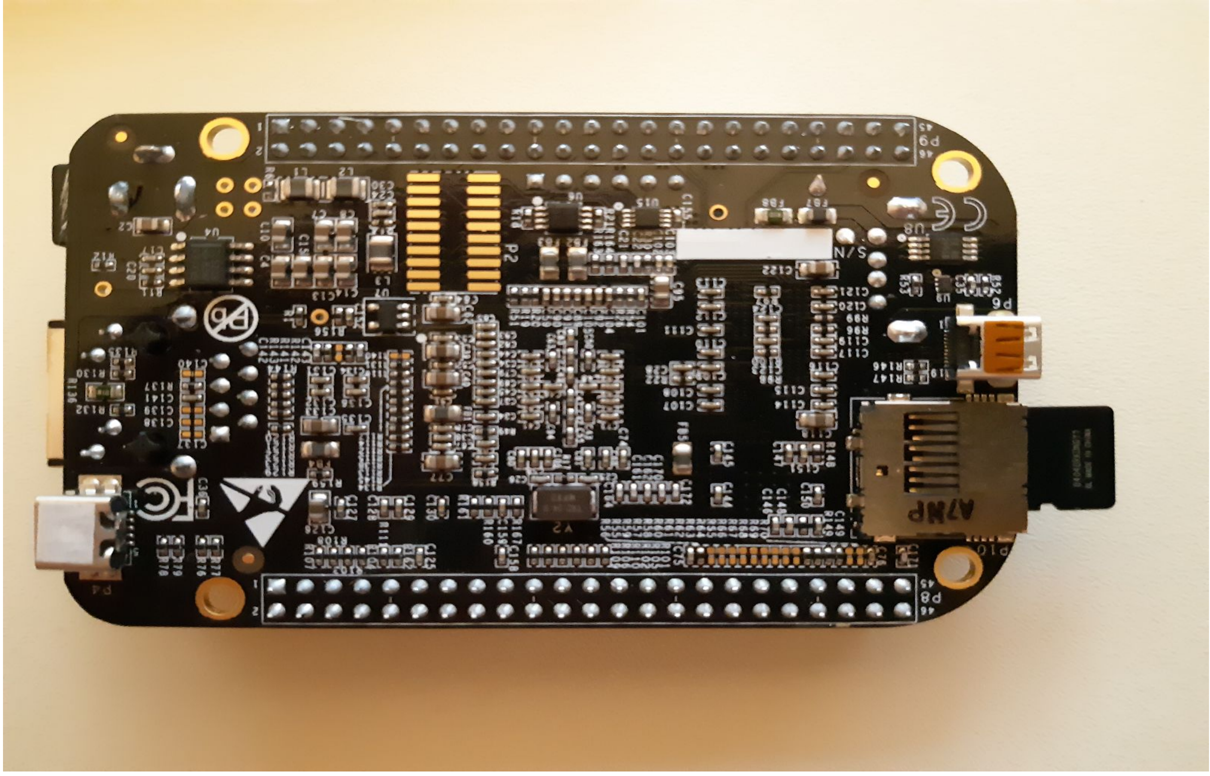


Figure 16: Insert Micro-SD Card into the board.

4. To power the board during the eMMC Flash stage, we are going to use a USB wall charger (5V, 1A) as supposed to using the PC to power it; this is because the current output on PCs are limited to 500mA, and the board current usage during this stage might exceed this value.

Figure 17: USB Wall Charger

5. While holding down the Boot Button on the Beagle Board (refer to Figure 18 below), insert the Micro-USB Wire from the USB Wall Charger to the board; the boot button is held down as power is being applied so that the board flashes from the Micro-SD Card. Let go of boot button once the USER LEDs start flashing. The USER LEDs will start to blink in a random order(for me LEDs 0, 2, and 3 were blinking). Once the board has been flashed, all four USER LEDs will light up simultaneously (Figure 18). This process will take around 30 minutes; if it takes longer than that for you, remove the USB wire and try the above steps again.

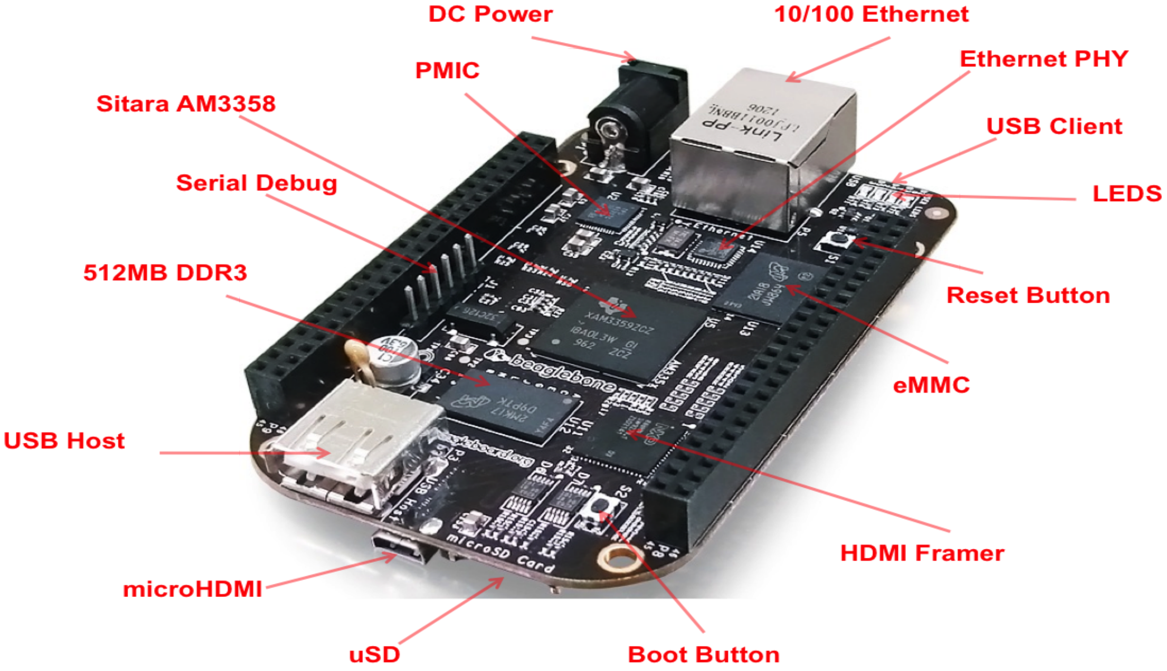


Figure 18: Boot Button located near the end of the board.

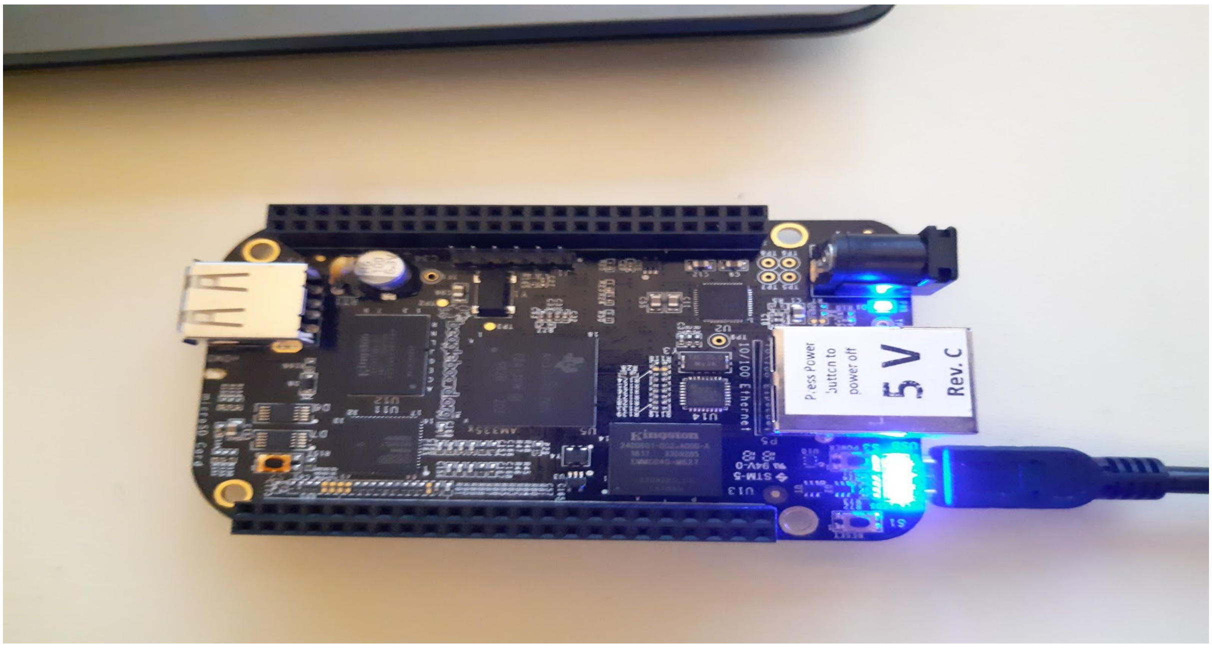
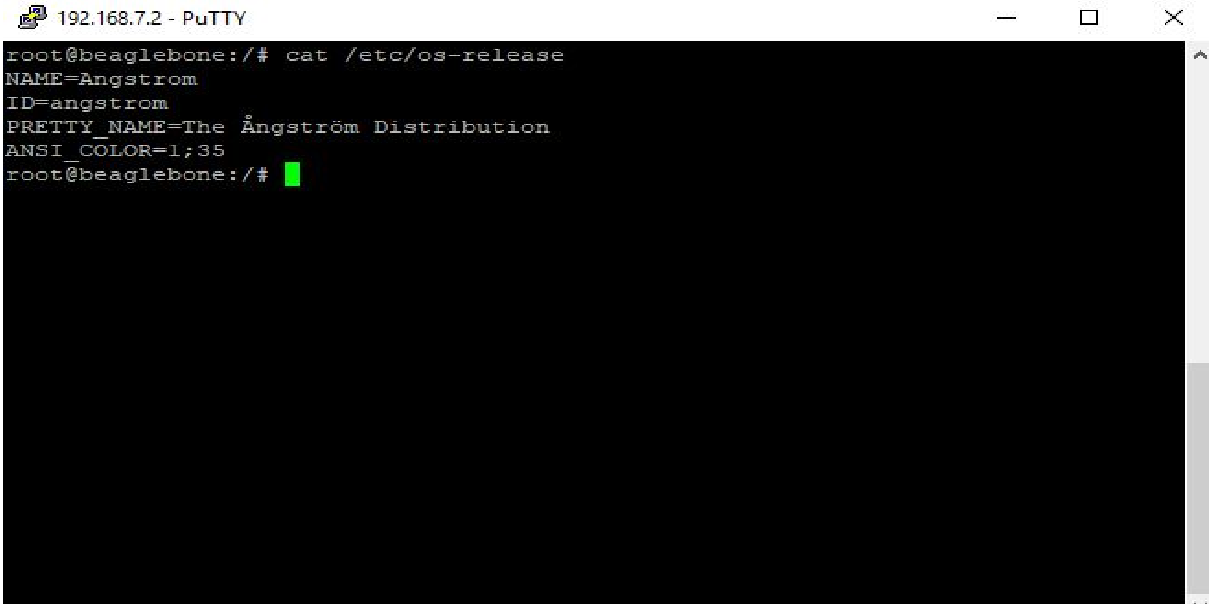


Figure 19: All 4 USER LEDs lit up simultaneously.

6. Unplug the USB cable, and remove the Micro-SD Card from the board. Plug the BeagleBone Black Board to your PC, and SSH to 192.168.7.2 (for Windows) using a terminal emulator (such as Putty).

7. The default username is “root”, and there is no password. To verify that you have Angstrom installed, type

```
cat /etc/os-release
```



```
root@beaglebone:/# cat /etc/os-release
NAME=Angstrom
ID=angstrom
PRETTY_NAME=The Ångström Distribution
ANSI_COLOR=1;35
root@beaglebone:/#
```

Figure 20: Angstrom has been installed onto the Beagle Board.

C. Example Programs For The BeagleBone Black

Sample C-File: Hello World

Step 3: Hello World

Now that Angstrom has been flashed onto the board, it can used like another other Linux Distribution. It can be used to create Bash Scripts, as well as C Files. We will now create our first Hello World program in C.

1. Change to a directory (cd) you are comfortable working in, or make a new directory (mkdir).

2. Open a text editor such as nano, and create a new C File; name it “hello.c”:

```
nano hello.c
```

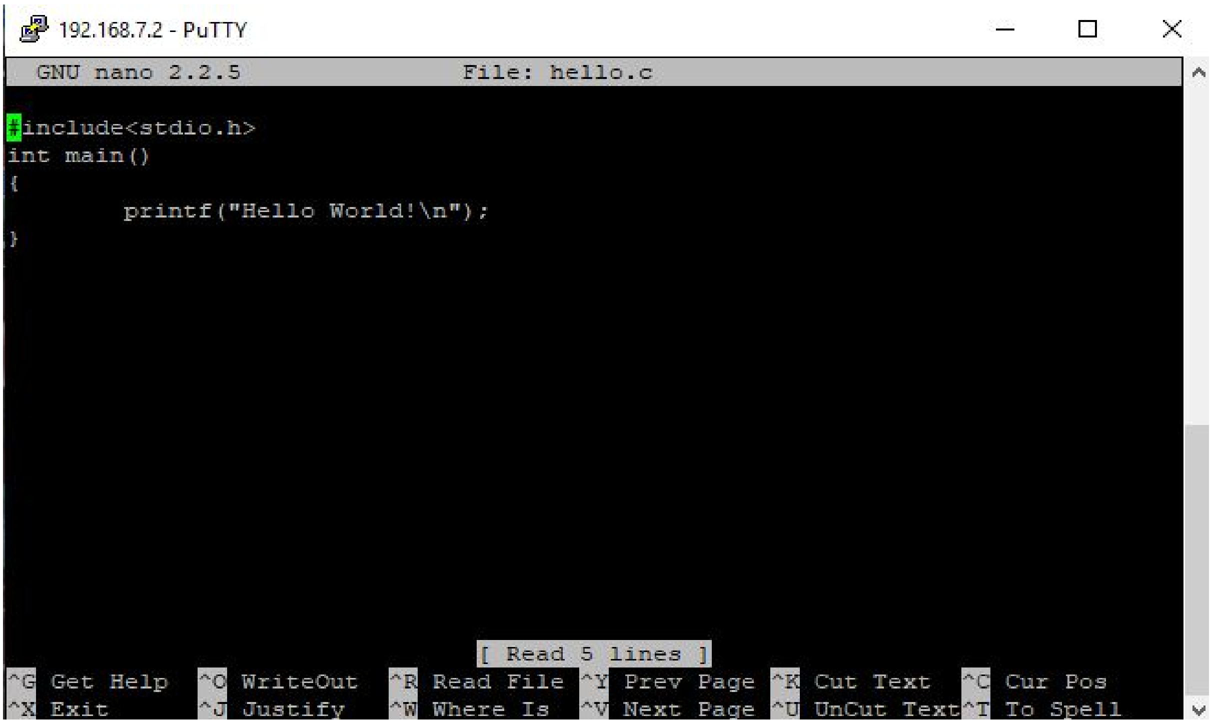
3. Enter the following lines to create your hello.c file:

```
#include<stdio.h>
int main()
{
     printf(“Hello World”);
}
```

Save the file and exit.

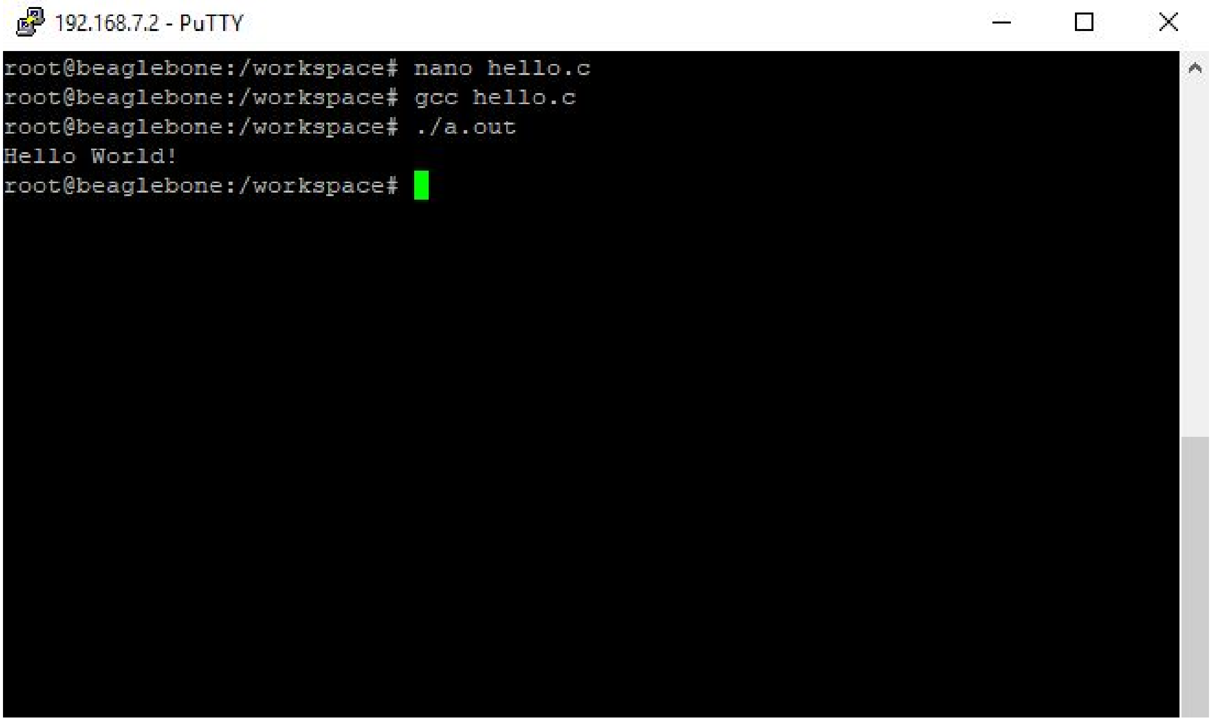
4. Compile the file, and run the executable.

```
gcc hello.c

./a.out
```



```
  GNU nano 2.2.5                File: hello.c

#include<stdio.h>
int main()
{
        printf("Hello World!\n");
}

                              [ Read 5 lines ]
^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

Figure 21: hello.c in nano



```
root@beaglebone:/workspace# nano hello.c
root@beaglebone:/workspace# gcc hello.c
root@beaglebone:/workspace# ./a.out
Hello World!
root@beaglebone:/workspace#
```

Figure 22: “Hello World!” has been printed onto the terminal.

Connecting To The Internet and Correcting Time

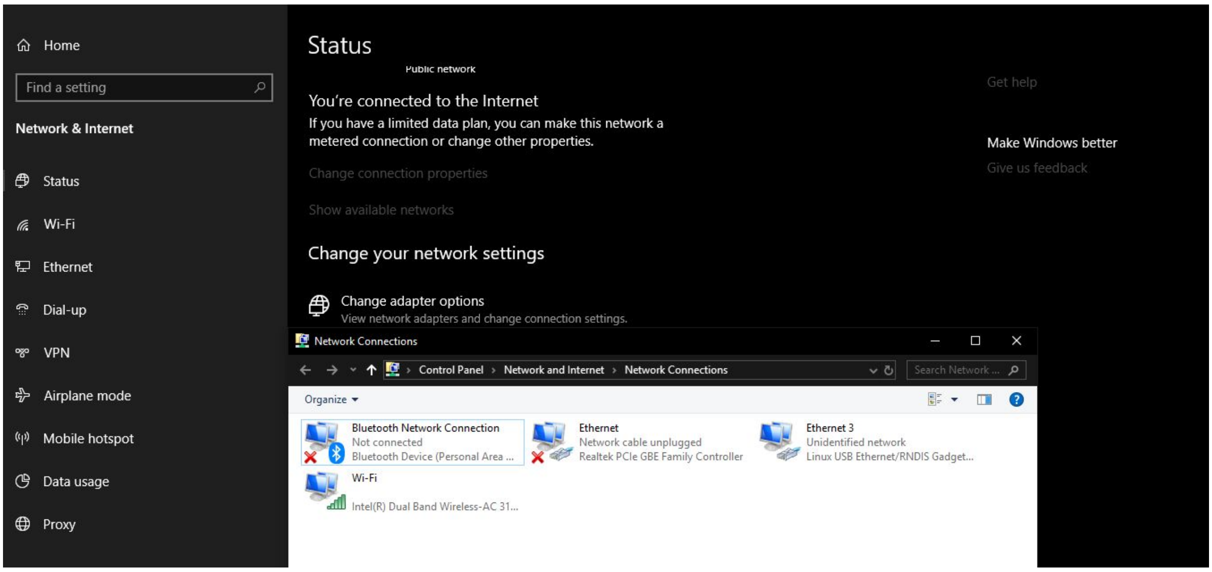
Step 4: Internet Access on BeagleBone Black Board Thru USB

These next steps will allow your BeagleBone Black board to access the internet through USB.

1. On Windows, search for “Network and Sharing Center”, and then go to “Change Adapter Options”. A window with Ethernet and Wifi connections should open up.

OR

Right-click Lan, or Wifi symbol on bottom right of taskbar, and “Open Network and Internet Settings”, and then click “Change Adapter Options”.



Status

Public network

You’re connected to the Internet

If you have a limited data plan, you can make this network a metered connection or change other properties.

Change your network settings

Change adapter options

View network adapters and change connection settings.

Network Connections

Bluetooth Network Connection
Not connected
Bluetooth Device (Personal Area ...

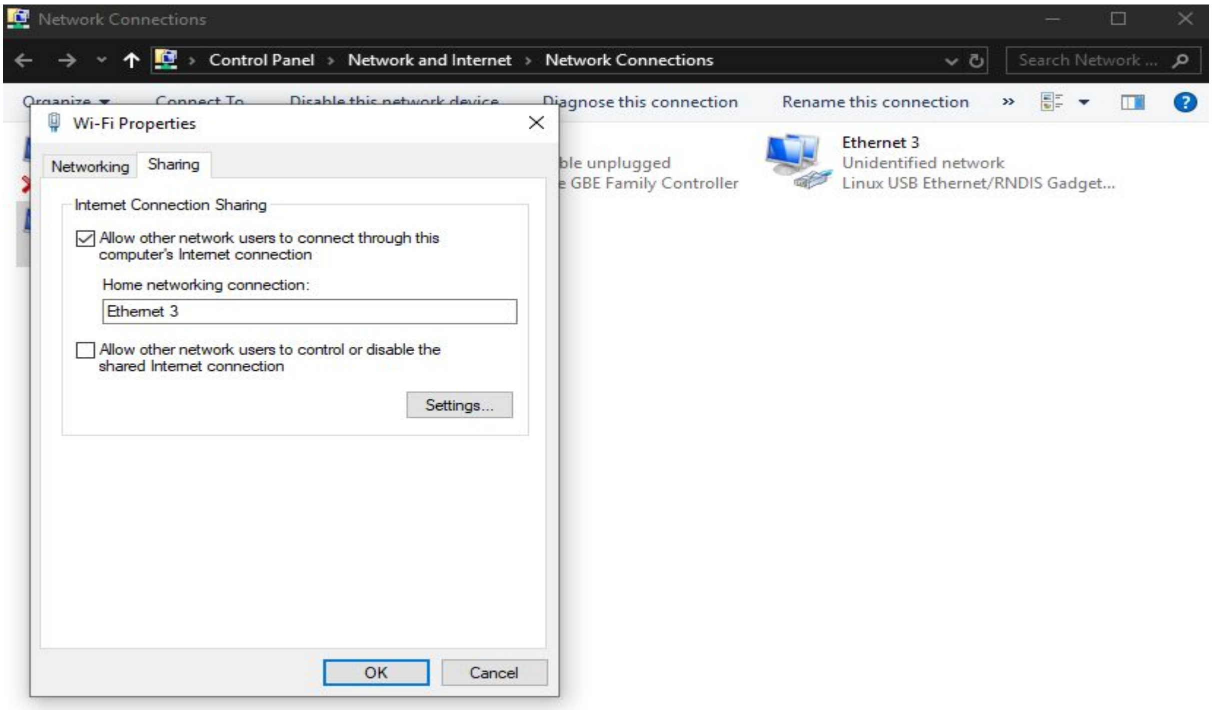
Ethernet
Network cable unplugged
Realtek PCIe GBE Family Controller

Ethernet 3
Unidentified network
Linux USB Ethernet/RNDIS Gadget...

Wi-Fi
Intel(R) Dual Band Wireless-AC 31...

Figure 23: Window that opens up with Ethernet and Wifi Connections

2. Right click on the adapter that your PC is connected to the internet with (either Ethernet or Wifi), and go to “Properties”. Go to the “Sharing Tab” near the top of the window and check the “Allow other network users to connect through this computer’s Internet Connection”, and in the box below “Home Networking Connection”, enter the adapter that is referring to the BeagleBone Network (Ethernet 3 in my case), and close this window. Note: This step has to be done every time your PC reboots.



Wi-Fi Properties

Internet Connection Sharing

☑ Allow other network users to connect through this computer's Internet connection

Home networking connection: Ethernet 3

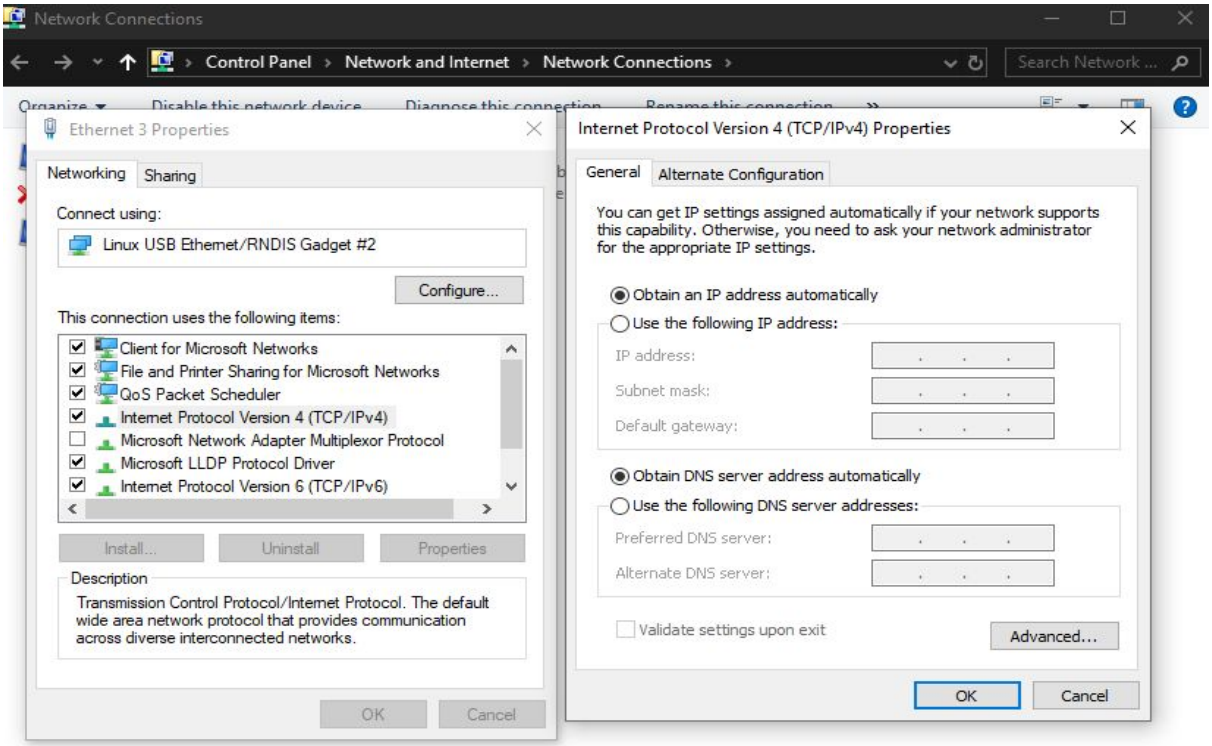
□ Allow other network users to control or disable the shared Internet connection

ble unplugged
e GBE Family Controller

Ethernet 3
Unidentified network
Linux USB Ethernet/RNDIS Gadget...

Figure 24: Allow the Beralge Board to connect to the internet through USB.

3. Right click the adapter referring to the board (Ethernet 3 in my case), and highlight Internet Protocol Version 4 with the mouse, and hit the Properties button. Check the “Obtain the IP address automatically” and the “Obtain DNS server address automatically” bubble. Close all the windows. Note: This step has to be done every time your PC reboots.



Ethernet 3 Properties

Connect using:

Linux USB Ethernet/RNDIS Gadget #2

This connection uses the following items:

☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☑ QoS Packet Scheduler
☑ Internet Protocol Version 4 (TCP/IPv4)
□ Microsoft Network Adapter Multiplexor Protocol
☑ Microsoft LLDP Protocol Driver
☑ Internet Protocol Version 6 (TCP/IPv6)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically
○ Use the following IP address:

IP address: . . .
Subnet mask: . . .
Default gateway: . . .

● Obtain DNS server address automatically
○ Use the following DNS server addresses:

Preferred DNS server: . . .
Alternate DNS server: . . .

□ Validate settings upon exit

Figure 25: Obtain IP Address Automatically as supposed to using the IP Address assigned by Windows.

4. If there had been a Putty session to the board, it would have been aborted; reconnect to the board through SSH, via Putty as done in Step 1.

5. Once you are able to SSH with the board, make a new script, named mystartup in nano. The script will allow for access to the internet, and change the time, and date as stated by the ntp servers.

```
nano mystartup
```

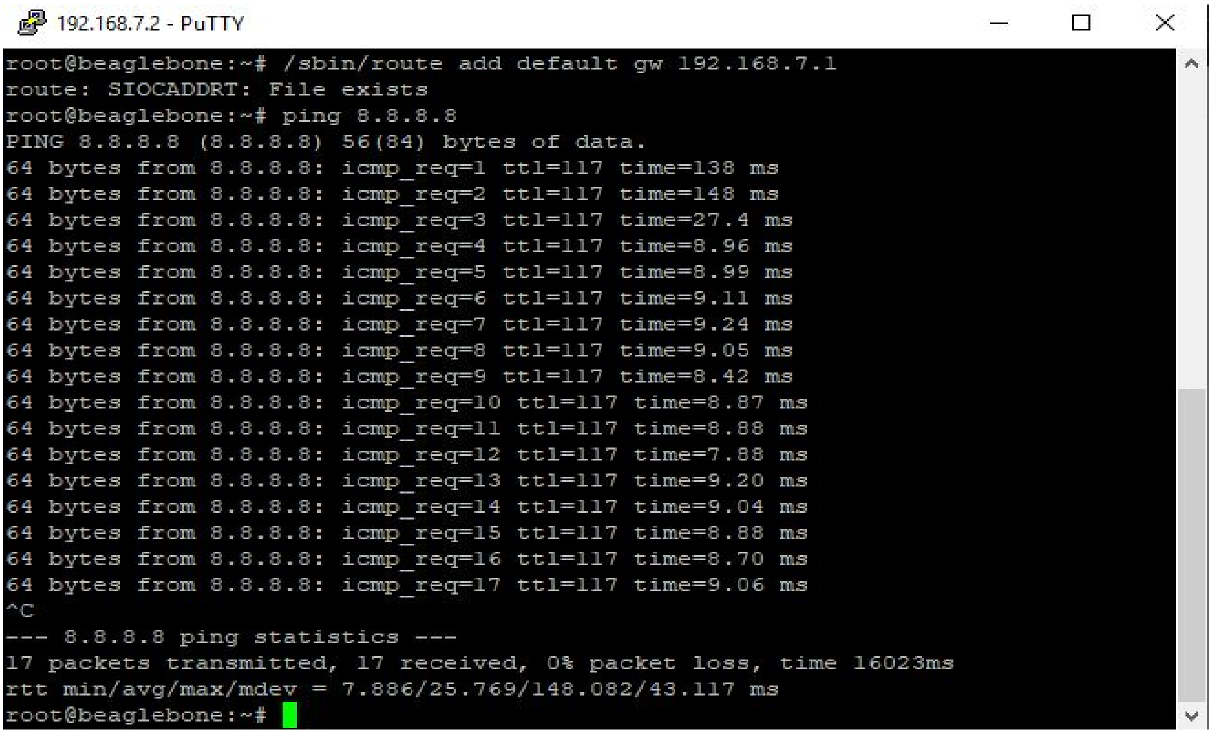
In the file write the following commands:

```
/sbin/route add default gw 192.168.7.1

echo "USB Internet Sharing Enabled!"
```

These commands will change the default gateway on the board to `192.168.7.1,` and will display a prompt once it is done.

```
echo “nameserver 8.8.8.8” >> /etc/resolv.conf

echo "Nameserver added!"
```

These commands will add the Google Nameserver, allowing you to ping websites like “google.com” and “youtube.com”



```
root@beaglebone:~# /sbin/route add default gw 192.168.7.1
route: SIOCADDRT: File exists
root@beaglebone:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_req=1 ttl=117 time=138 ms
64 bytes from 8.8.8.8: icmp_req=2 ttl=117 time=148 ms
64 bytes from 8.8.8.8: icmp_req=3 ttl=117 time=27.4 ms
64 bytes from 8.8.8.8: icmp_req=4 ttl=117 time=8.96 ms
64 bytes from 8.8.8.8: icmp_req=5 ttl=117 time=8.99 ms
64 bytes from 8.8.8.8: icmp_req=6 ttl=117 time=9.11 ms
64 bytes from 8.8.8.8: icmp_req=7 ttl=117 time=9.24 ms
64 bytes from 8.8.8.8: icmp_req=8 ttl=117 time=9.05 ms
64 bytes from 8.8.8.8: icmp_req=9 ttl=117 time=8.42 ms
64 bytes from 8.8.8.8: icmp_req=10 ttl=117 time=8.87 ms
64 bytes from 8.8.8.8: icmp_req=11 ttl=117 time=8.88 ms
64 bytes from 8.8.8.8: icmp_req=12 ttl=117 time=7.88 ms
64 bytes from 8.8.8.8: icmp_req=13 ttl=117 time=9.20 ms
64 bytes from 8.8.8.8: icmp_req=14 ttl=117 time=9.04 ms
64 bytes from 8.8.8.8: icmp_req=15 ttl=117 time=8.88 ms
64 bytes from 8.8.8.8: icmp_req=16 ttl=117 time=8.70 ms
64 bytes from 8.8.8.8: icmp_req=17 ttl=117 time=9.06 ms
^C
--- 8.8.8.8 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16023ms
rtt min/avg/max/mdev = 7.886/25.769/148.082/43.117 ms
root@beaglebone:~#
```

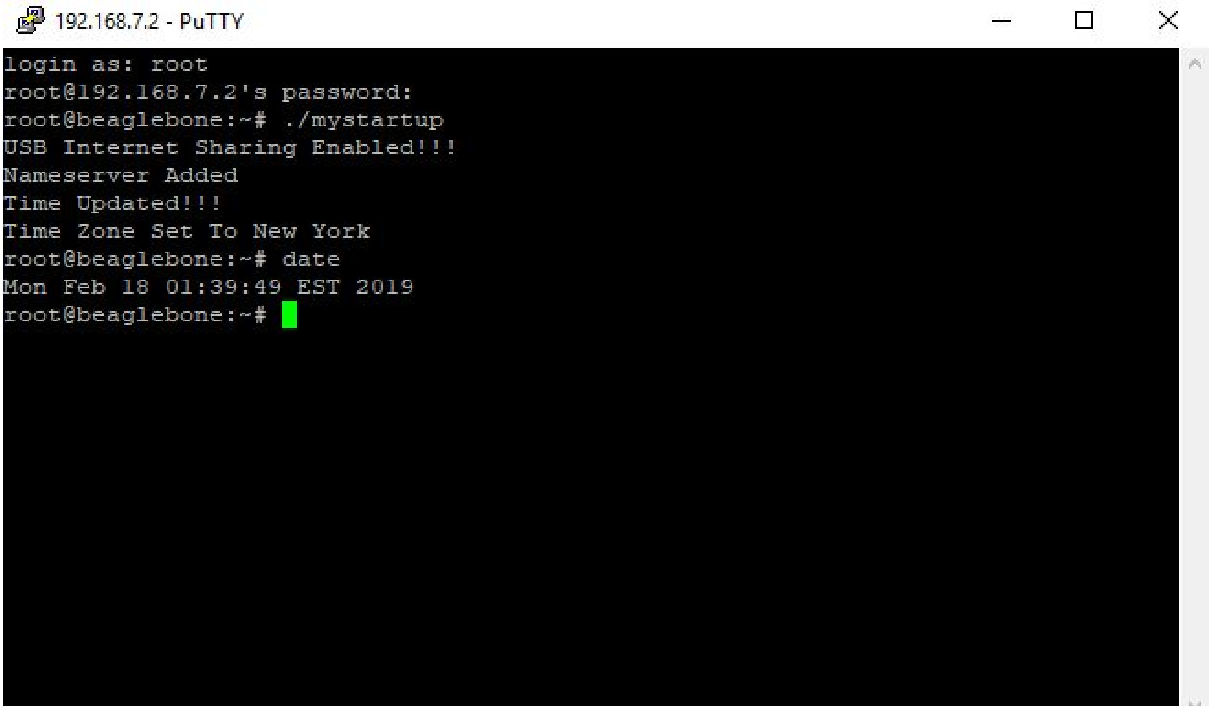
Figure 26: Pinging 8.8.8.8

The board when first booted will state that the date is Jan 1, 2000. To change the time and timezone add the following lines to the script:

```
/usr/bin/ntpdate -b -s -u ca.pool.ntp.org

echo "Time Updated"

cp /usr/share/zoneinfo/America/New_York /etc/localtime

echo “Time Zone Changed”
```

Save and close the file. Make the file executable:

```
chmod +x mystartup
```



```
login as: root
root@192.168.7.2's password:
root@beaglebone:~# ./mystartup
USB Internet Sharing Enabled!!!
Nameserver Added
Time Updated!!!
Time Zone Set To New York
root@beaglebone:~# date
Mon Feb 18 01:39:49 EST 2019
root@beaglebone:~#
```

Figure 27: The mystartup script running

Make sure to run the script before going through with the next step to make sure it works.

```
./mystartup
```

6. In order to have the mystartup run at boot when it is connected to the PC, we have to edit /etc/profile file.

```
nano /etc/profile
```

At the very end of the file, have it run the executable mystartup, and save and close the file.

```
./mystartup
```

Sample 7-Seg Program

Step 5: Display Time on 4 Digit 7 Segment Display

In order to control the GPIO pins we can use C/C++, Python, Bash, or Assembly. For this report we will use Python. We will first have to install the IO Python Library provided by Adafruit; the control of GPIO ports using this library is very similar to using the Arduino. This can be done by following the instructions on (also found below): https://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black/installation. We will then use this library to control the GPIO pins in the board to light the current time onto a 7-Seg LED Display.

1. Once SSH'd to Beaglebone using Putty, type the following lines to install the IO Python Library provided by Adafruit

```
/usr/bin/ntpdate -b -s -u pool.ntp.org

opkg update && opkg install python-pip python-setuptools python-smbus

pip install Adafruit_BBIO
```

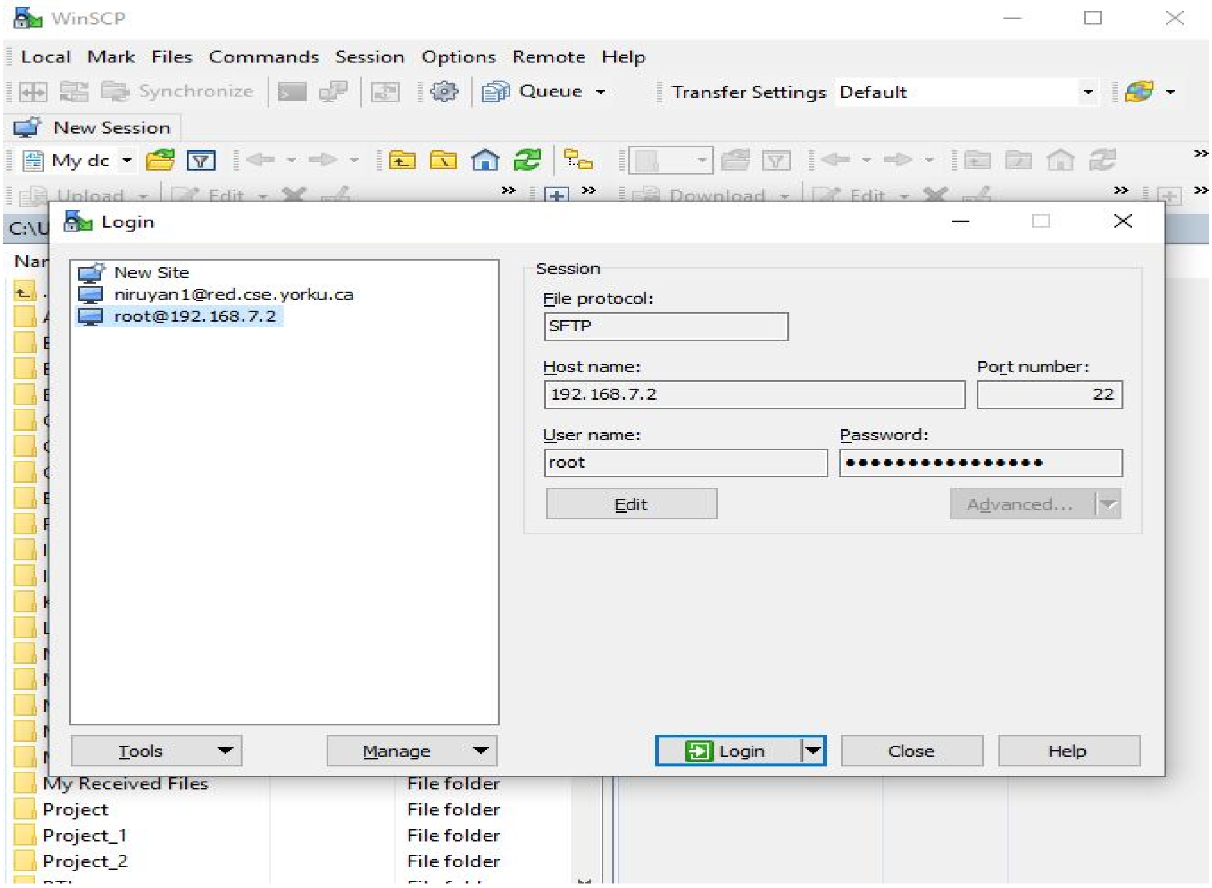
To make sure that your installation was successful type:

```
python -c "import Adafruit_BBIO.GPIO as GPIO; print GPIO"
```

You should see:

```
<module 'Adafruit_BBIO.GPIO' from
'/usr/local/lib/python2.7/dist-packages/Adafruit_BBIO/GPIO.so'>
```

2. To create Python files, we can use a text editor available such as nano, or create them on Windows using a more advanced Python IDE on Windows (i.e. Notepad) and then transfer them onto the BeagleBoard. To transfer files between the BeagleBoard and your Windows machine, you can use WinSCP; SSH to the board in WinSCP (Figure 28).



New Site
niruyan1@red.cse.yorku.ca
root@192.168.7.2

Session
File protocol: SFTP
Host name: 192.168.7.2
Port number: 22
User name: root
Password: ●●●●●●●●●●●●●●●●

Figure 28: WinSCP interface

3. Wire the 7-Segment Board properly such that the GPIO ports can control the display (as shown in Figure 30), and the current through the LEDs are not excessive; 1K ohm resistors were used to limit current in the LEDs.

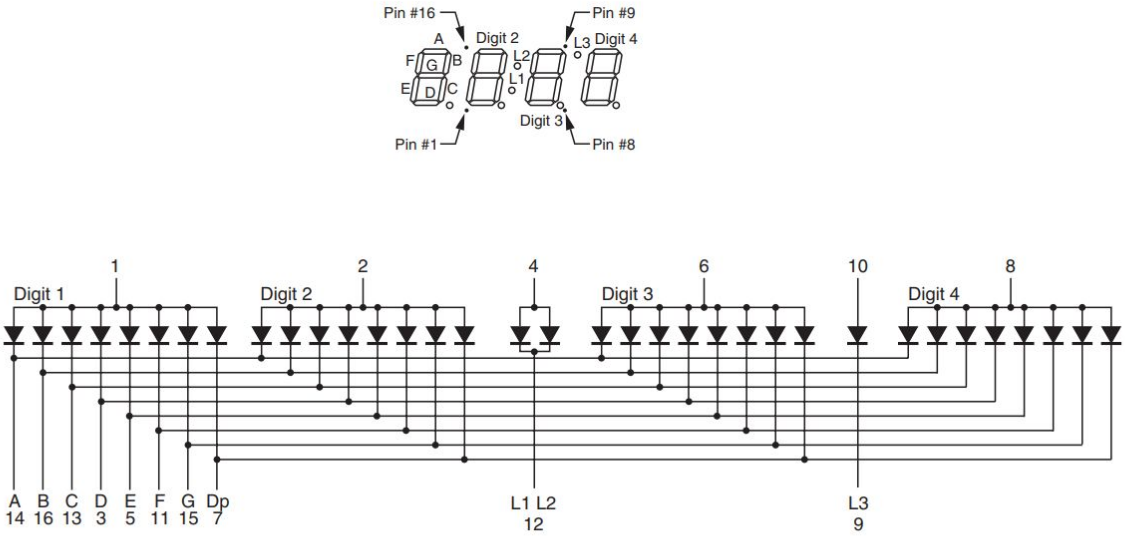


Figure 29: 7 Segment Display Pin Layout from https://www.mouser.com/datasheet/2/143/ds300180-48190.pdf

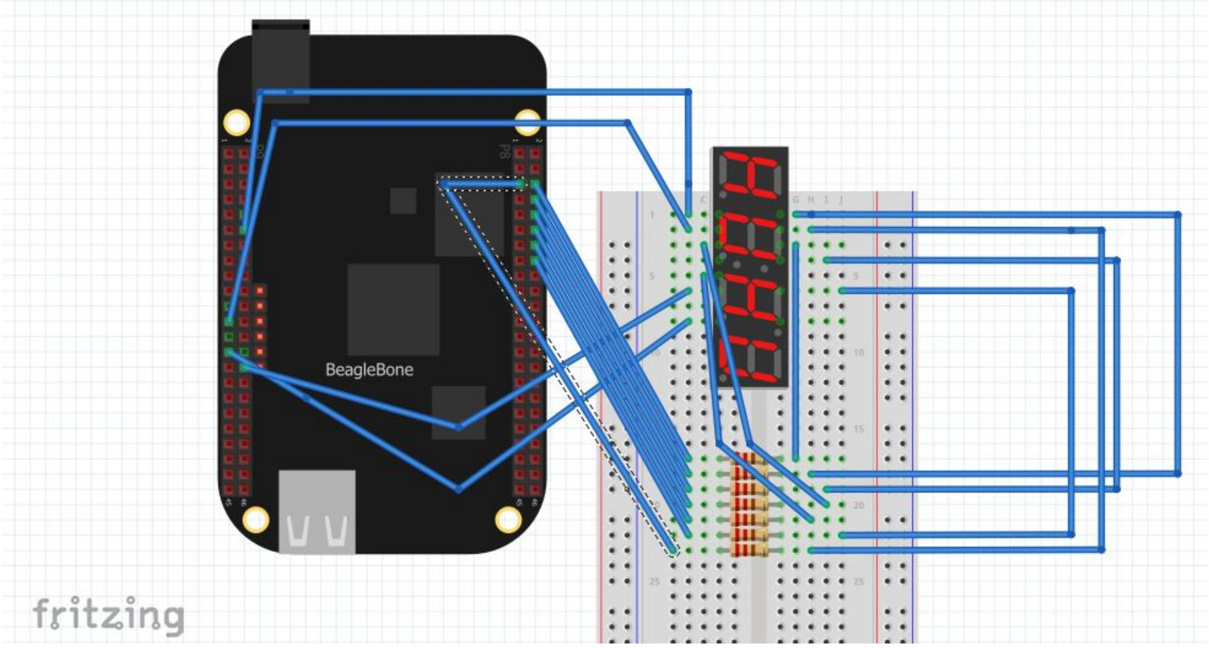


Figure 30: Wiring for the 7-Seg Display.

4. Create your python file (ledseg.py) that is able to control the 7 Segment Display; an example file is shown below:

%%
ledseg.py

```
import Adafruit_BBIO.GPIO as GPIO
import datetime
import sys
from time import sleep

#Resets the GPIO Pins from previous program
GPIO.cleanup()

#Function that will control 1 Digit given digit and number
def lightswitch(digit, num):
#initially make them low to clear the LEDs from previous call
     GPIO.output(digit1, GPIO.LOW)
     GPIO.output(digit2, GPIO.LOW)
     GPIO.output(digit3, GPIO.LOW)
     GPIO.output(digit4, GPIO.LOW)
#initially make them low to clear the LEDs from previous call
     GPIO.output(sega, GPIO.HIGH)
     GPIO.output(segb, GPIO.HIGH)
     GPIO.output(segc, GPIO.HIGH)
     GPIO.output(segd, GPIO.HIGH)
     GPIO.output(sege, GPIO.HIGH)
     GPIO.output(segf, GPIO.HIGH)
```

```
    GPIO.output(segg, GPIO.HIGH)
#if digit is 1,2,3 … change the GPIOs as intended
    if num == 1:
        GPIO.output(segb, GPIO.LOW)
        GPIO.output(segc, GPIO.LOW)
    elif num == 2:
        GPIO.output(sega, GPIO.LOW)
        GPIO.output(segb, GPIO.LOW)
        GPIO.output(segd, GPIO.LOW)
        GPIO.output(sege, GPIO.LOW)
        GPIO.output(segg, GPIO.LOW)
    elif num == 3:
        GPIO.output(sega, GPIO.LOW)
        GPIO.output(segb, GPIO.LOW)
        GPIO.output(segc, GPIO.LOW)
        GPIO.output(segd, GPIO.LOW)
        GPIO.output(segg, GPIO.LOW)
    elif num == 4:
        GPIO.output(segb, GPIO.LOW)
        GPIO.output(segc, GPIO.LOW)
        GPIO.output(segf, GPIO.LOW)
        GPIO.output(segg, GPIO.LOW)
    elif num == 5:
        GPIO.output(sega, GPIO.LOW)
        GPIO.output(segc, GPIO.LOW)
        GPIO.output(segd, GPIO.LOW)
        GPIO.output(segf, GPIO.LOW)
        GPIO.output(segg, GPIO.LOW)
    elif num == 6:
        GPIO.output(sega, GPIO.LOW)
        GPIO.output(segc, GPIO.LOW)
        GPIO.output(segd, GPIO.LOW)
        GPIO.output(sege, GPIO.LOW)
        GPIO.output(segf, GPIO.LOW)
        GPIO.output(segg, GPIO.LOW)
    elif num == 7:
        GPIO.output(sega, GPIO.LOW)
        GPIO.output(segb, GPIO.LOW)
        GPIO.output(segc, GPIO.LOW)
    elif num == 8:
        GPIO.output(sega, GPIO.LOW)
        GPIO.output(segb, GPIO.LOW)
        GPIO.output(segc, GPIO.LOW)
        GPIO.output(segd, GPIO.LOW)
        GPIO.output(sege, GPIO.LOW)
        GPIO.output(segf, GPIO.LOW)
```

```
            GPIO.output(segg, GPIO.LOW)
        elif num == 9:
            GPIO.output(sega, GPIO.LOW)
            GPIO.output(segb, GPIO.LOW)
            GPIO.output(segc, GPIO.LOW)
            GPIO.output(segf, GPIO.LOW)
            GPIO.output(segg, GPIO.LOW)
        elif num == 0:
            GPIO.output(sega, GPIO.LOW)
            GPIO.output(segb, GPIO.LOW)
            GPIO.output(segc, GPIO.LOW)
            GPIO.output(segd, GPIO.LOW)
            GPIO.output(sege, GPIO.LOW)
            GPIO.output(segf, GPIO.LOW)
        else:
            GPIO.output(sega, GPIO.HIGH)
            GPIO.output(segb, GPIO.HIGH)
            GPIO.output(segc, GPIO.HIGH)
            GPIO.output(segd, GPIO.HIGH)
            GPIO.output(sege, GPIO.HIGH)
            GPIO.output(segf, GPIO.HIGH)
            GPIO.output(segg, GPIO.HIGH)
#light the correct digit(hour,minute)
        if digit == 1:
            GPIO.output(digit1, GPIO.HIGH)
        elif digit == 2:
            GPIO.output(digit2, GPIO.HIGH)
        elif digit == 3:
            GPIO.output(digit3, GPIO.HIGH)
        elif digit == 4:
            GPIO.output(digit4, GPIO.HIGH)
        else:
            GPIO.output(digit1, GPIO.LOW)
            GPIO.output(digit2, GPIO.LOW)
            GPIO.output(digit3, GPIO.LOW)
            GPIO.output(digit4, GPIO.LOW)


#Name your pins and set it output;setup such that no LEDs light up
digit1="P9_12"
GPIO.setup(digit1,GPIO.OUT)
digit2="P9_23"
GPIO.setup(digit2, GPIO.OUT)
digit3="P9_30"
GPIO.setup(digit3, GPIO.OUT)
digit4="P9_27"
```

```
GPIO.setup(digit4, GPIO.OUT)

sega="P8_8"
GPIO.setup(sega,GPIO.OUT)
segb="P8_10"
GPIO.setup(segb,GPIO.OUT)
segc="P8_12"
GPIO.setup(segc,GPIO.OUT)
segd="P8_14"
GPIO.setup(segd,GPIO.OUT)
sege="P8_16"
GPIO.setup(sege,GPIO.OUT)
segf="P8_18"
GPIO.setup(segf,GPIO.OUT)
segg="P8_7"
GPIO.setup(segg,GPIO.OUT)

#While loop that runs forever that constantly updates the time
while 1:
      current_time=datetime.datetime.now()
      lightswitch(1,current_time.hour/10)
      #Refresh rate of around 1000Hz; very little artifacting
      sleep(0.001)
      lightswitch(2,current_time.hour%10)
      sleep(0.001)
      lightswitch(3,current_time.minute/10)
      sleep(0.001)
      lightswitch(4,current_time.minute%10)
      sleep(0.001)
```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Press Ctrl-Z or Ctrl-C to exit the program.

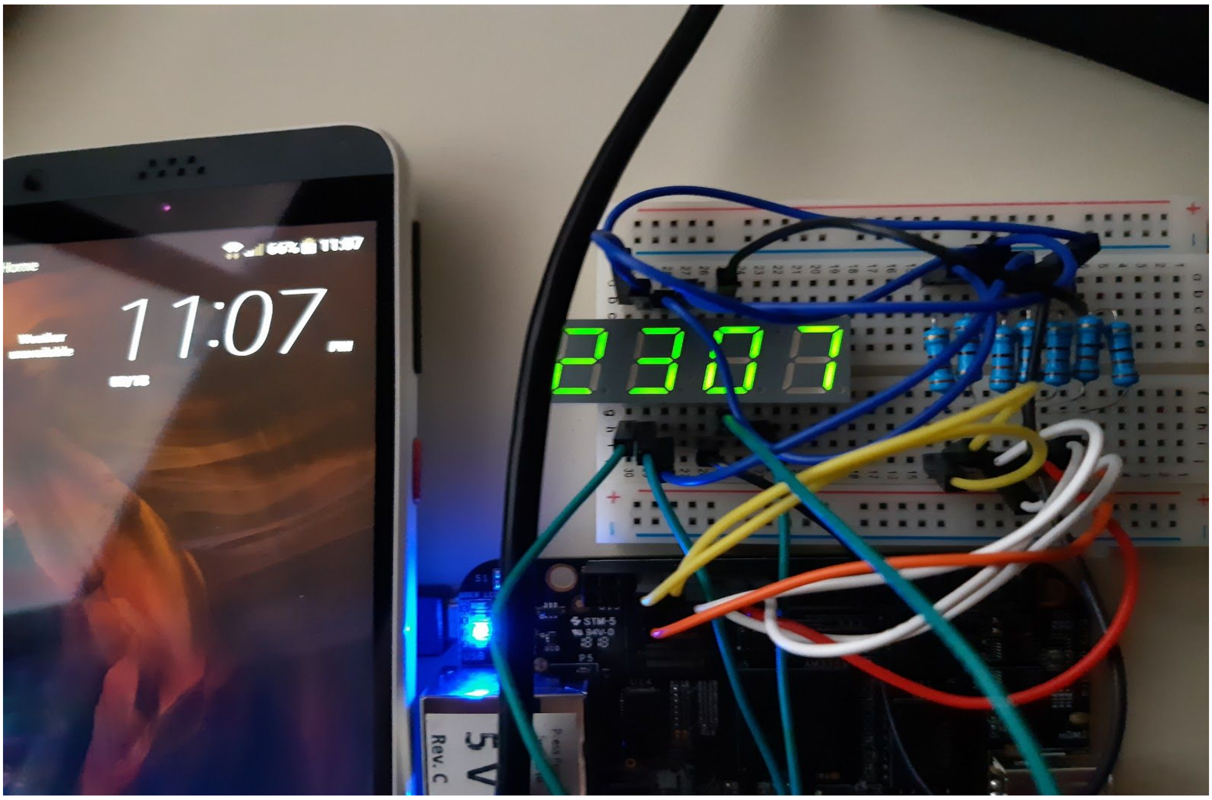


Figure 30: The program running at 11:07 pm (23:07)

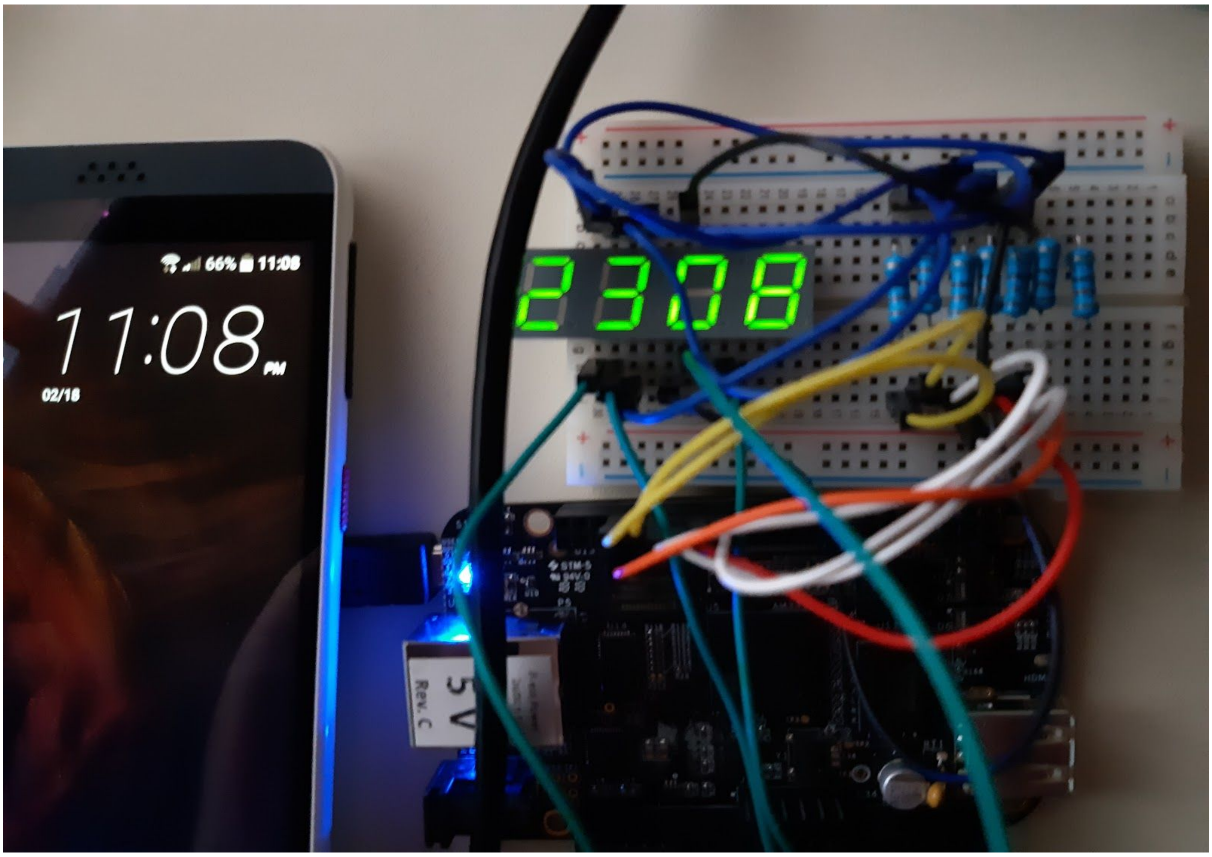


Figure 31: The program running 1 minute later at 11:08 pm (23:08)

This is a simple program that grabs the internet time, and displays the time on the 7-Seg LED. This program can be made better to include the use of interrupts (like a button connected to a GPIO_IN pin) to stop the time from displaying time, as supposed to using the terminal to terminate the program. The amount of times the board fetches the internet time can be made less frequent to reduce the CPU load; this can be done thru the use of timers and interrupts.

Troubles and Solutions

Downloading Drivers for Windows

When downloading the drivers for Windows in Step 1 (in order to enable communication between the PC and the board), the driver installation might fail due to the driver signatures being outdated. Make sure you download the latest drivers from; http://beagleboard.org/getting-started

Cannot Network Through Board

Consult Step 4.3 and Step 4.4; have to redo these steps every time your PC reboots. If still cannot connect to Network, re-flash board.

Cannot SSH To Board

If you were able to SSH to the board previously, and are not able to afterwards, try reflashing the board. The boot file `/etc/profile` might have been incorrectly configured..

References

1.beagleboard.org: Getting Started. http://beagleboard.org/getting-started
2. The Angstrom Distribution. http://www.angstrom-distribution.org/
3. BeagleBone Black System Reference Manual
https://media.digikey.com/pdf/Data%20Sheets/Circuitco%20Elect/BB-BBLK-000%20Manual.pdf
4. Exploring BeagleBone By Derek Molloy **ISBN-10:** 1118935128
5. Automatically Setting the Beaglebone Black Time Using NTP
http://derekmolloy.ie/automatically-setting-the-beaglebone-black-time-using-ntp/
6. How to Connect a BeagleBone Black to the Internet Using USB
https://www.digikey.ca/en/maker/blogs/how-to-connect-a-beaglebone-black-to-the-internet-using-usb
7. Beaglebone Black LESSON 4: Digital Write to the GPIO Pins from Python
https://www.youtube.com/watch?v=Yg54VE9CeRE
8. Setting Up IO Python Library on BeagleBone Black
https://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black/installation