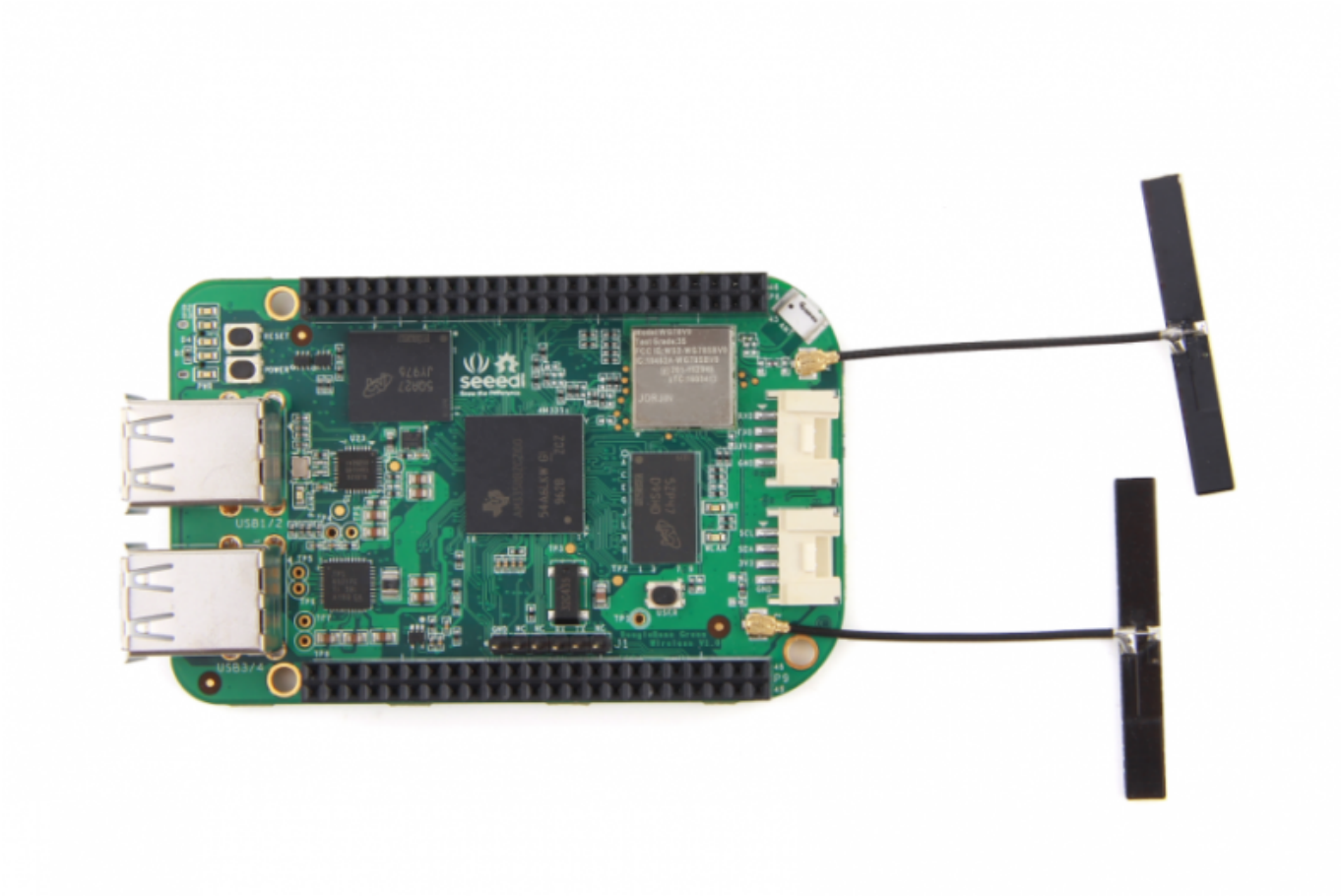
BeagleBone Green Wireless SKU: 102010048



SeeedStudio BeagleBone Green Wireless (BBGW) is a joint effort by BeagleBoard.org and Seeed Studio. It is based on the open-source hardware design of BeagleBone Black and developed into this differentiated version. SeeedStudio BeagleBone Green Wireless has included a high-performance flexible WiFi/Bluetooth interface and two Grove connectors, making it easier to connect to the large family of Grove sensors. The on-board HDMI and Ethernet are removed to make room for these wireless features and Grove connectors.

https://www.youtube.com/embed/tlA97rhKG10

Features

- **Fully Compatiable with BeagleBone Black**
- **Processor: AM335x 1GHz ARM® Cortex-A8**
  - 512MB DDR3 RAM
  - 4GB 8-bit eMMC on-board flash storage
  - 3D graphics accelerator
  - NEON floating-point accelerator
  - 2x PRU 32-bit microcontrollers
- **Connectivity**
  - USB client for power & communications

    - USB host with 4-port hub
    - WiFi 802.11 b/g/n 2.4GHz
    - Bluetooth 4.1 with BLE
    - 2x 46 pin headers
    - 2x Grove connectors (I2C and UART)

- **Software Compatibility**

    - Debian
    - Android
    - Ubuntu
    - Cloud9 IDE on Node.js w/ BoneScript library
    - plus much more

Specification

Item	Value
Processor	AM335x 1GHz ARMR Cortex-A8
RAM	512MB DDR3
on-board Flash Storage	4GB eMMC
CPU Supports	NEON floating-point & 3D graphics accelerator
Micro USB Supports	powering & communications
USB	USB2.0 Host *4
Grove Connectors	2 (One I2C and One UART)
GPIO	2 x 46 pin headers
Ethernet	Wi-Fi 802.11b/g/n 2.4GHz and Bluetooth 4.1 LE
Operating Temperature	0 ~ 75
Size	L:89mm W:54.6mm H:16mm
Weight	43.3g
Package size	L: 140mm W: 75mm H: 25mm
Gross Weight	88g

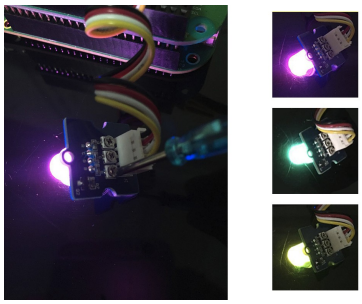
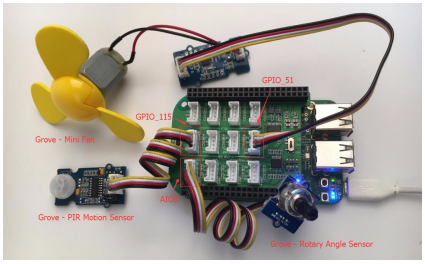
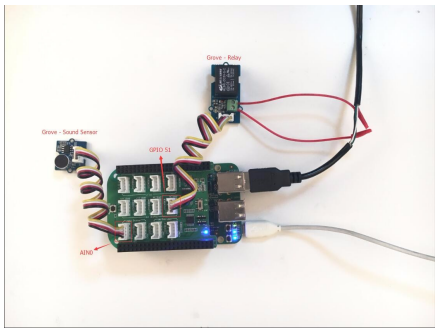
Application Ideas

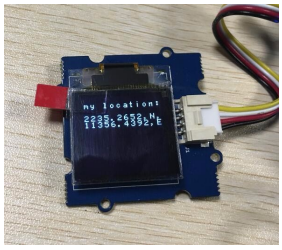
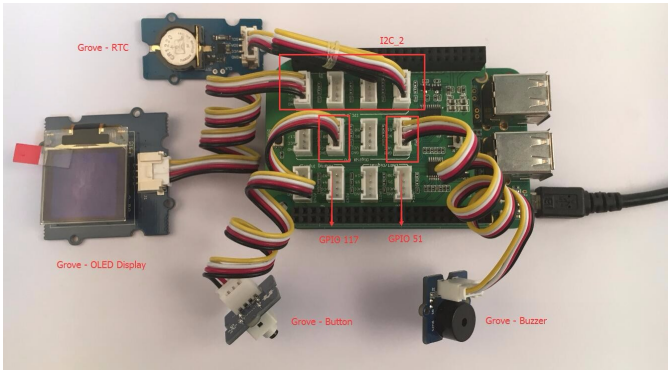
- Internet of Things
- Smart House
- Industrial
- Automation & Process Control
- Human Machine Interface
- Sensor Hub

- Robot

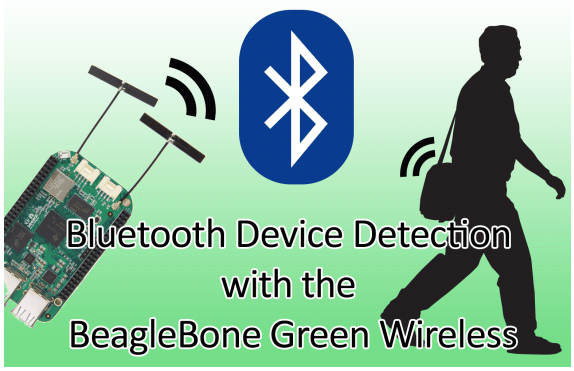
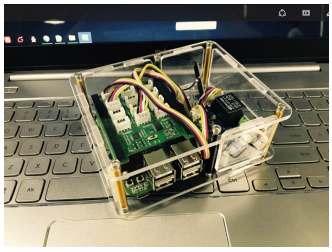
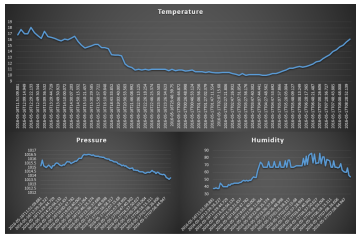
BBGW Starter Tutorial #1-#6

Here are some tutorials for the starters to use some Grove modules to realize their interesting ideas with BeagleBone Green Wireless(BBGW). The tutorials are based on Python and mraa/upm library.

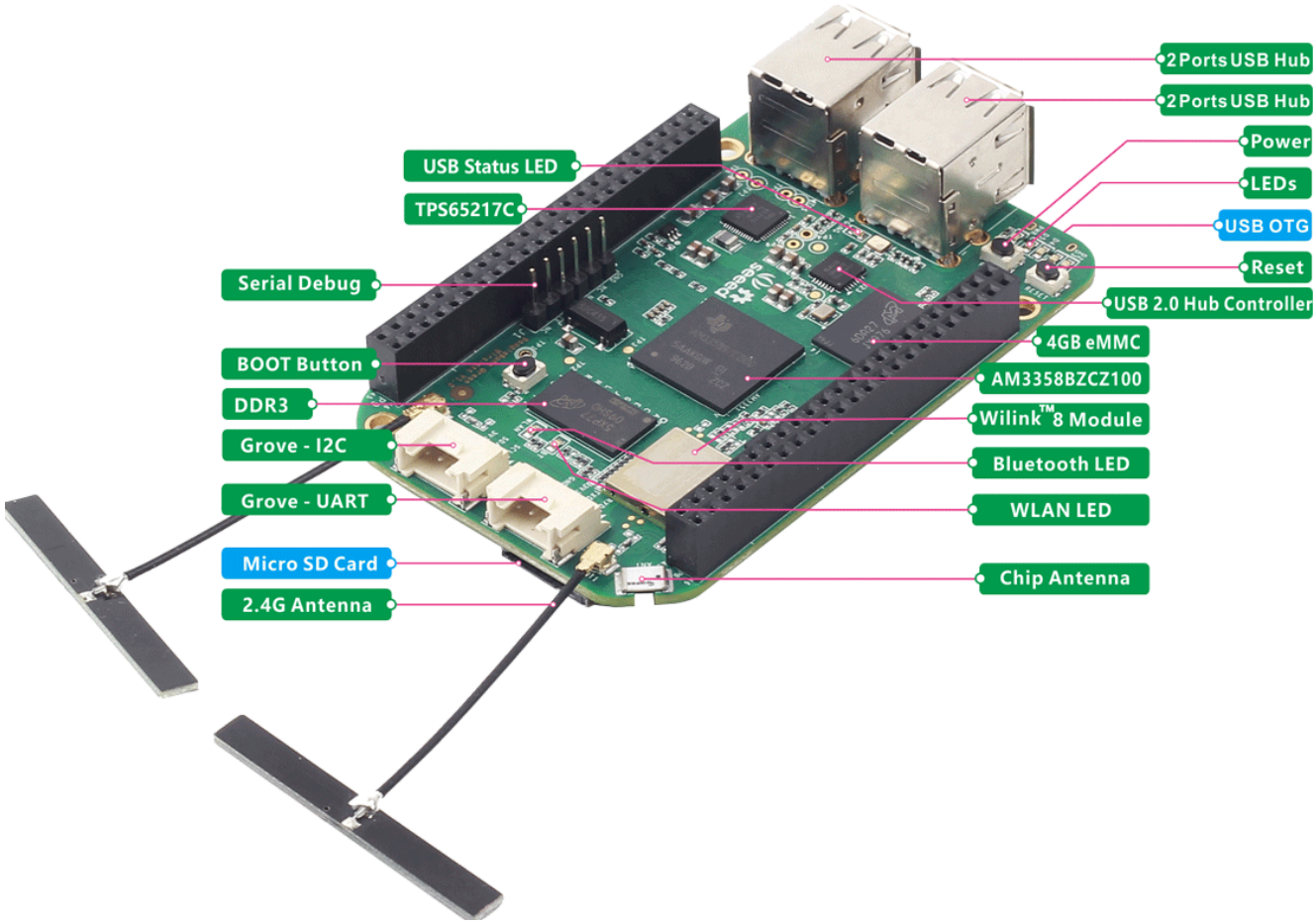
#1 The Breath LED	#2 Storm on your table	#3 Speak Louuuuuudly
		
MAKE IT NOW!	MAKE IT NOW!	MAKE IT NOW!

#4 How hot is it today?	#5 Where are you?	#6 My Little alarm clock
		
MAKE IT NOW!	MAKE IT NOW!	MAKE IT NOW!

Funny Projects

Bluetooth Device Detection	Home Control Center	SAP HCP IoT Service
		
MAKE IT NOW!	MAKE IT NOW!	MAKE IT NOW!

Hardware Overview

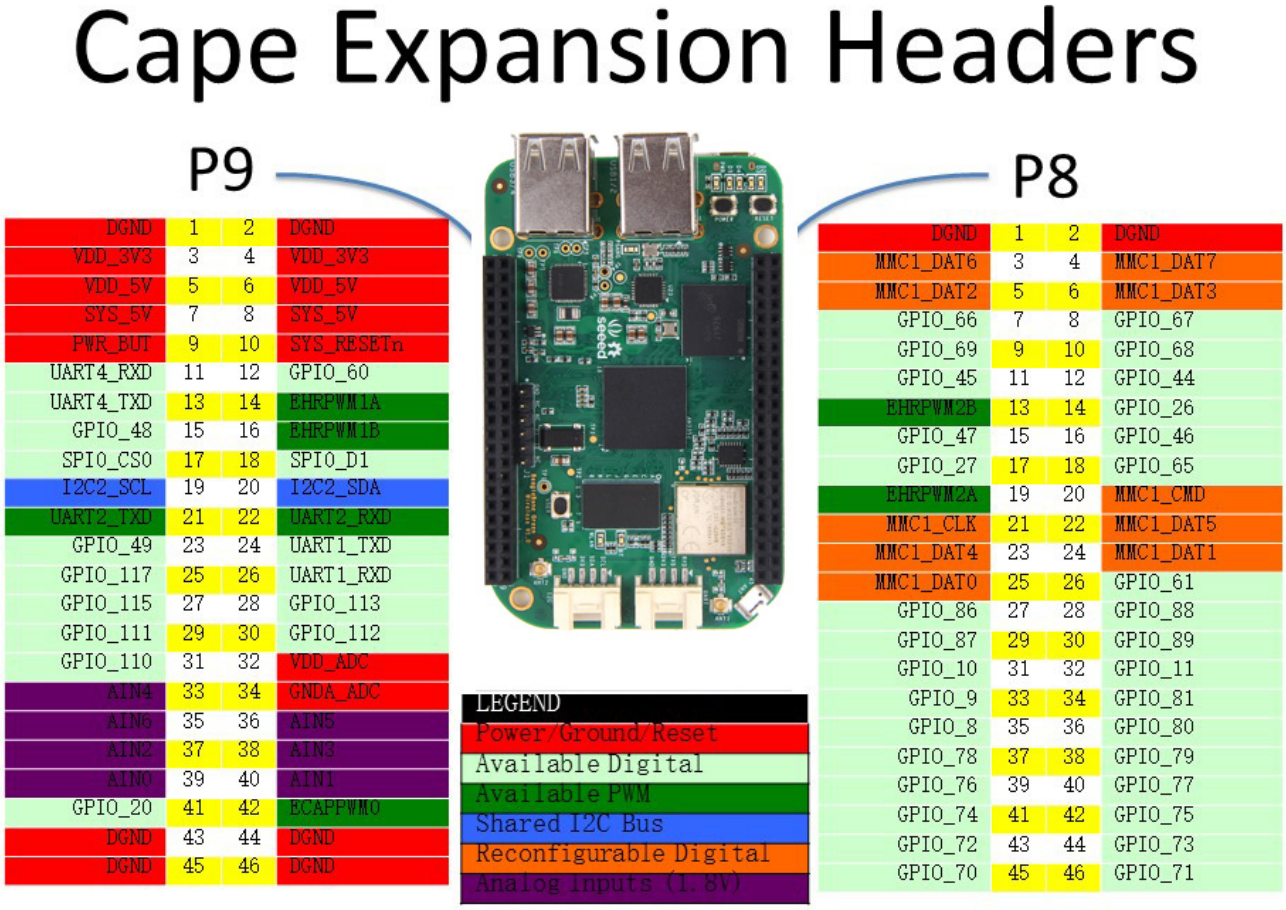


Pin map

Each digital I/O pin has 8 different modes that can be selected, including GPIO.

65 Possible Digital I/Os

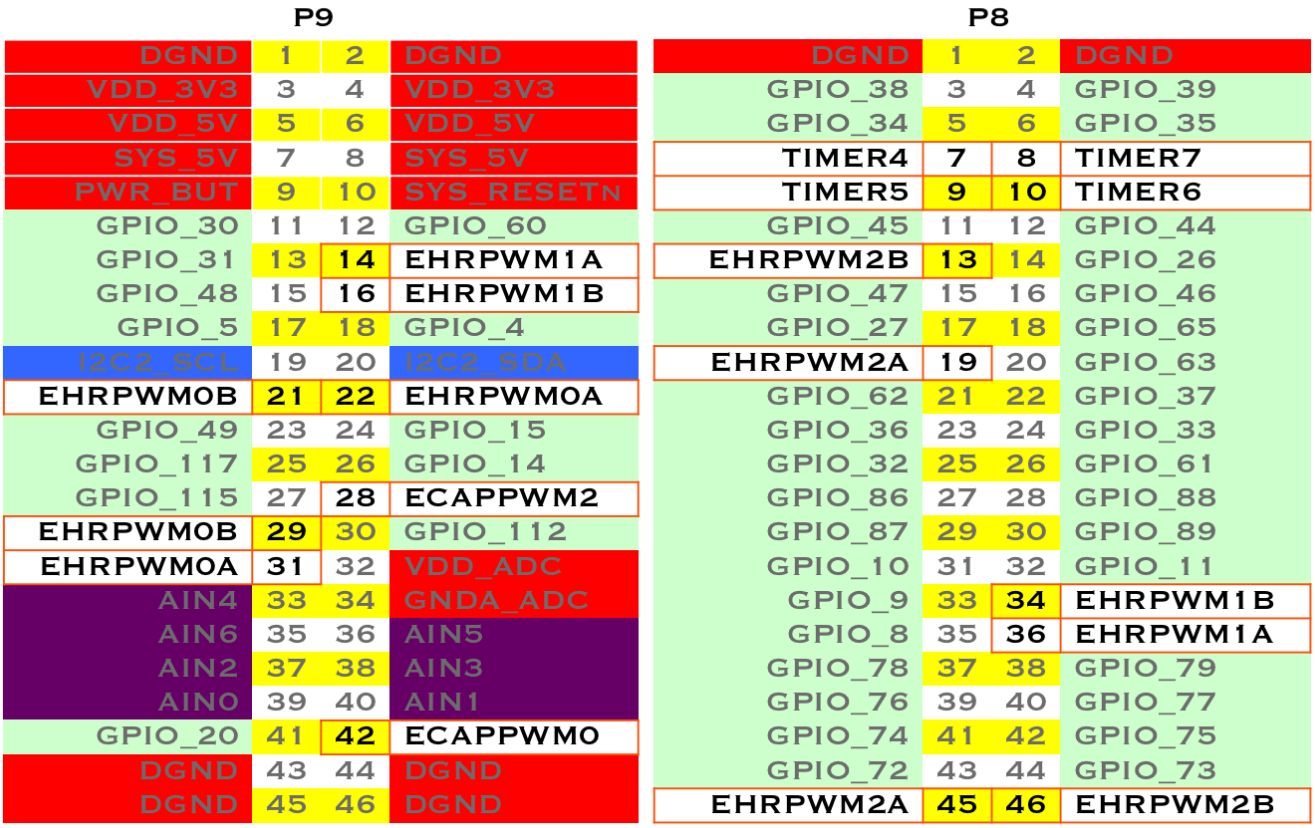
!!!Note In GPIO mode, each digital I/O can produce interrupts.



PWMs and Timers

!!!Note Up to 8 digital I/O pins can be configured with pulse-width modulators (PWM) to produce signals to control motors or create pseudo analog voltage levels, without taking up any extra CPU cycles.

8 PWMs and 4 timers



Analog Inputs

!!!Note Make sure you don't input more than 1.8V to the analog input pins. This is a single 12-bit analog-to-digital converter with 8 channels, 7 of which are made available on the headers.

7 analog inputs (1.8V)

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
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
GPIO_5	17	18	GPIO_4
I2C2_SCL	19	20	I2C2_SDA
GPIO_3	21	22	GPIO_2
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	GPIO_113
GPIO_111	29	30	GPIO_112
GPIO_110	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	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

UART

!!!Note There is a dedicated header for getting to the UART0 pins and connecting a debug cable. Five additional serial ports are brought to the expansion headers, but one of them only has a single direction brought to the headers.

4 UARTs and 1 TX only

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	GPIO_50
GPIO_48	15	16	GPIO_51
GPIO_5	17	18	GPIO_4
UART1_RTSn	19	20	UART1_CTSn
UART2_TXD	21	22	UART2_RXD
GPIO_49	23	24	UART1_TXD
GPIO_117	25	26	UART1_RXD
GPIO_115	27	28	GPIO_113
GPIO_111	29	30	GPIO_112
GPIO_110	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	UART3_TXD
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
UART5_CTSn+	31	32	UART5_RTSn
UART4_RTSn	33	34	UART3_RTSn
UART4_CTSn	35	36	UART3_CTSn
UARR5_TXD+	37	38	UART5_RXD+
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

I2C

!!!Note The first I2C bus is utilized for reading EEPROMS on cape add-on boards and can't be used for other digital I/O operations without interfering with that function, but you can still use it to add other I2C devices at available addresses. The second I2C bus is available for you to configure and use.

2 I2C ports

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
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
I2C1_SCL	17	18	I2C1_SDA
I2C2_SCL	19	20	I2C2_SDA
I2C2_SCL	21	22	I2C2_SDA
GPIO_49	23	24	I2C1_SCL
GPIO_117	25	26	I2C1_SDA
GPIO_115	27	28	GPIO_113
GPIO_111	29	30	GPIO_112
GPIO_110	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	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

####SPI

!!!Note For shifting out data fast, you might consider using one of the SPI ports.

2 SPI ports

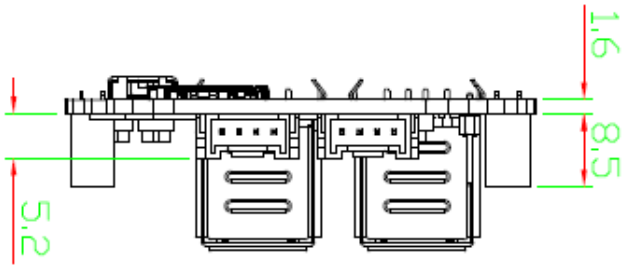
P9

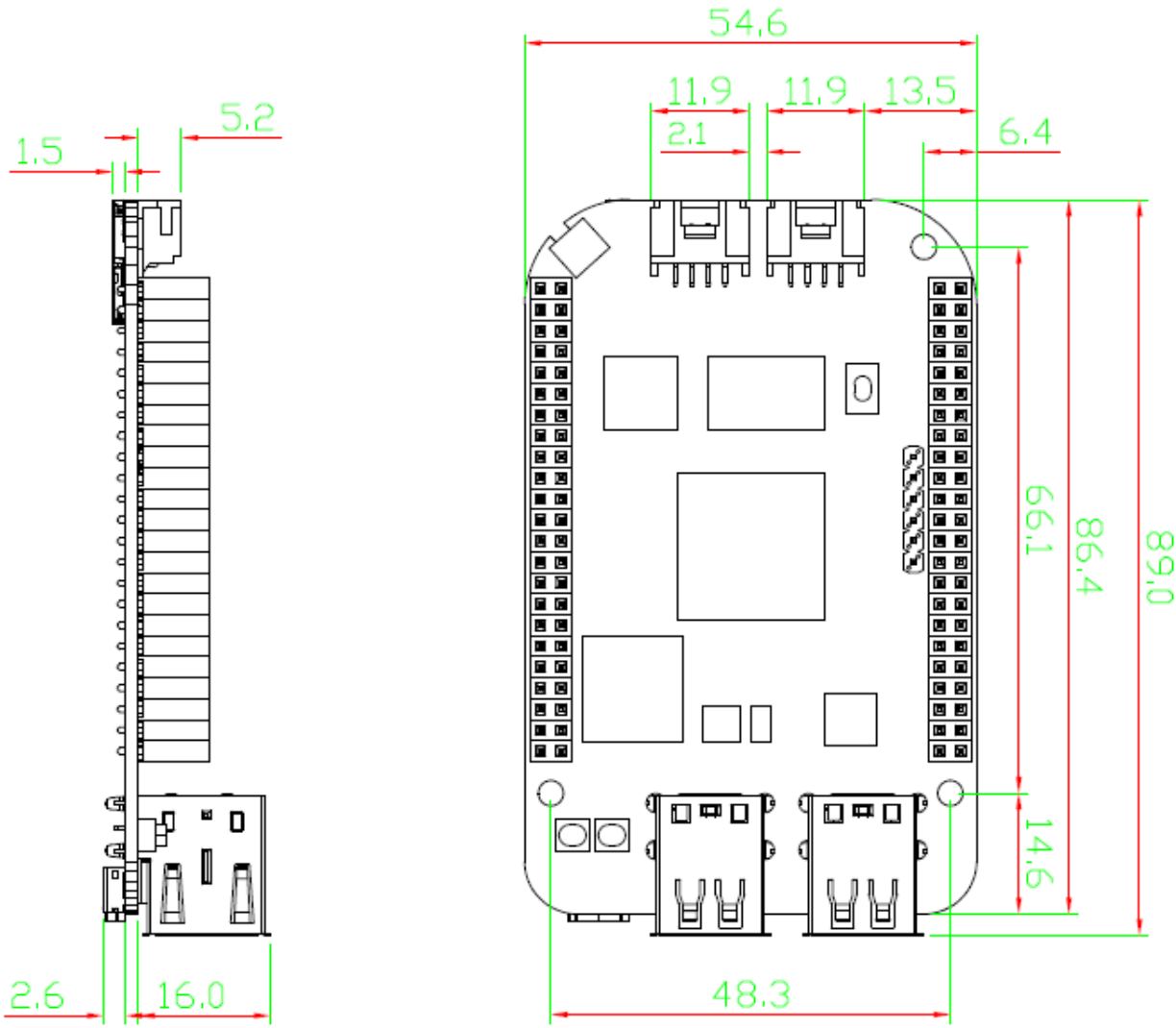
DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETn
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
SPI0_CS0	17	18	SPI0_D1
SPI1_CS1	19	20	SPI1_CS0
SPI0_D0	21	22	SPI0_SCLK
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	SPI1_CS0
SPI1_D0	29	30	SPI1_D1
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	SPI1_CS1
DGND	43	44	DGND
DGND	45	46	DGND

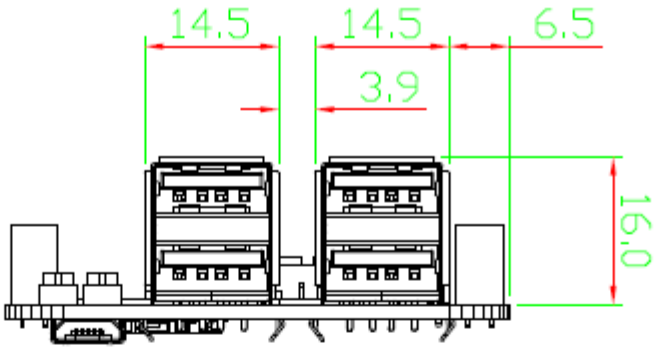
P8

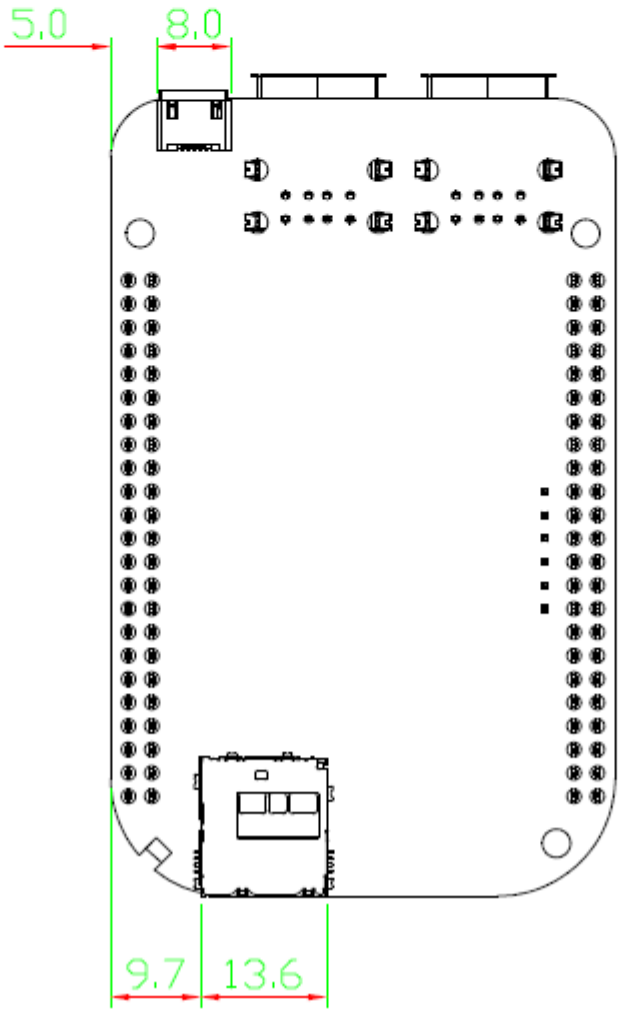
DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

Mechincal Drawing


1.6
8.5
5.2


54.6
11.9
11.9
13.5
2.1
6.4
1.5
5.2
66.1
86.4
89.0
14.6
2.6
16.0
48.3





Getting Started

!!!Note This chapter is writing under Win10. The steps are familiar for the other operate systems.

STEP1. Plug in your BBGW via USB

Use the provided micro USB cable to plug your BBGW into your computer. This will both power the board and provide a development interface. BBGW will boot Linux from the **on-board 2GB** or 4GB eMMC.

BBGW will operate as a flash drive providing you with a local copy of the documentation and drivers. Note that this interface may not be used to re-configure the microSD card with a new image, but may be used to update the boot parameters using the uEnv.txt file.

You'll see the PWR LED lit steadily. Within 10 seconds, you should see the other LEDs blinking in their default configurations.

- D2 is configured at boot to blink in a heartbeat pattern
- D3 is configured at boot to light during microSD card accesses
- D4 is configured at boot to light during CPU activity
- D5 is configured at boot to light during eMMC accesses

STEP2. Install Drivers

Install the drivers for your operating system to give you network-over-USB access to your Beagle. Additional drivers give you serial access to your board.

Operating System	USB Drivers	Comments
Windows (64-bit)	64-bit installer	
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.

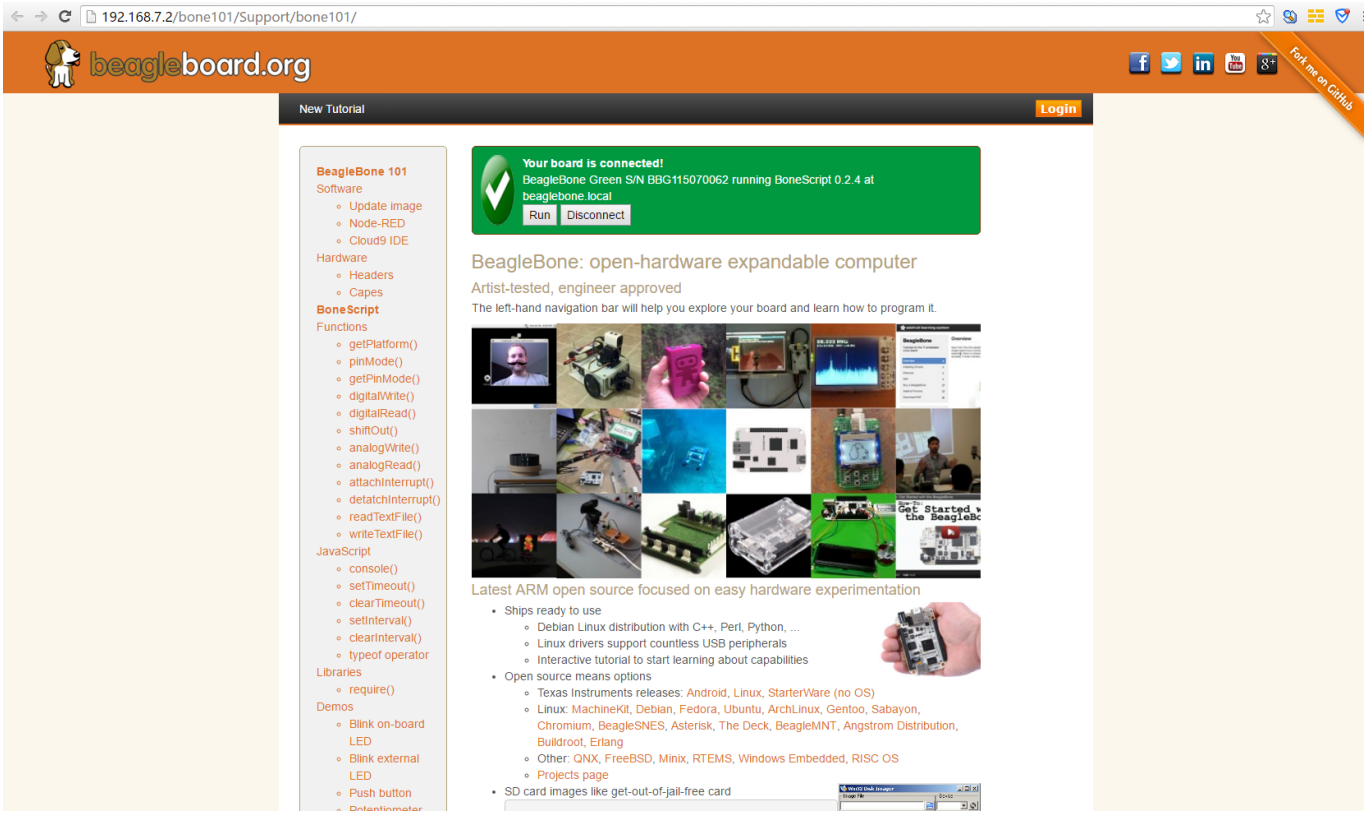
!!!Note For window system, please note that:

```
* Windows Driver Certification warning may pop up two or three times. Click
"Ignore", "Install" or "Run"
* To check if you're running 32 or 64-bit Windows see [this]
(https://support.microsoft.com/kb/827218).
*  On systems without the latest service release, you may get an error
(0xc000007b). In that case, please [install](https://www.microsoft.com/en-
us/download/confirmation.aspx?id=13523) and retry:
* You may need to reboot Windows.
* These drivers have been tested to work up to Windows 10
```

STEP3. Browse to your Beagle

Using either Chrome or Firefox (Internet Explorer will NOT work), browse to the web server running on your board. It will load a presentation showing you the capabilities of the board. Use the arrow keys on your keyboard to navigate the presentation.

Click http://192.168.7.2 to launch to your BBGW. Older software images require you to EJECT the BEAGLE_BONE drive to start the network. With the latest software image, that step is no longer required.

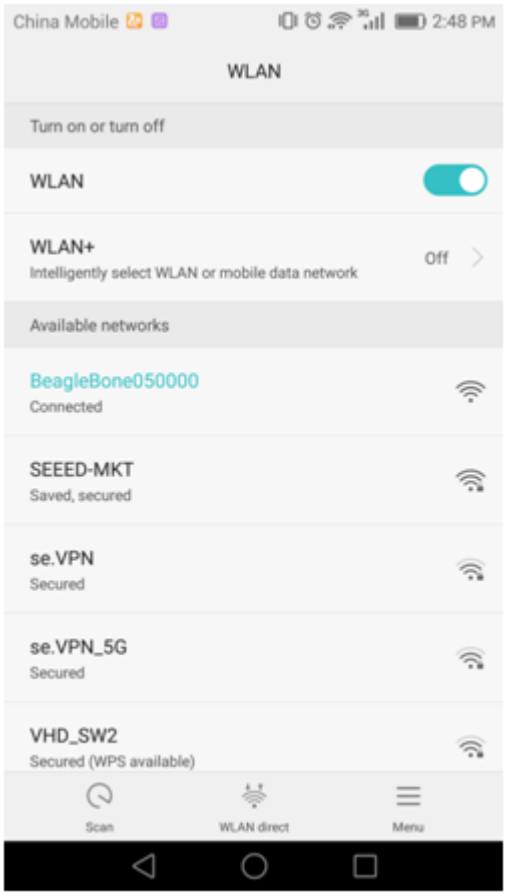


STEP4. Cloud9 IDE

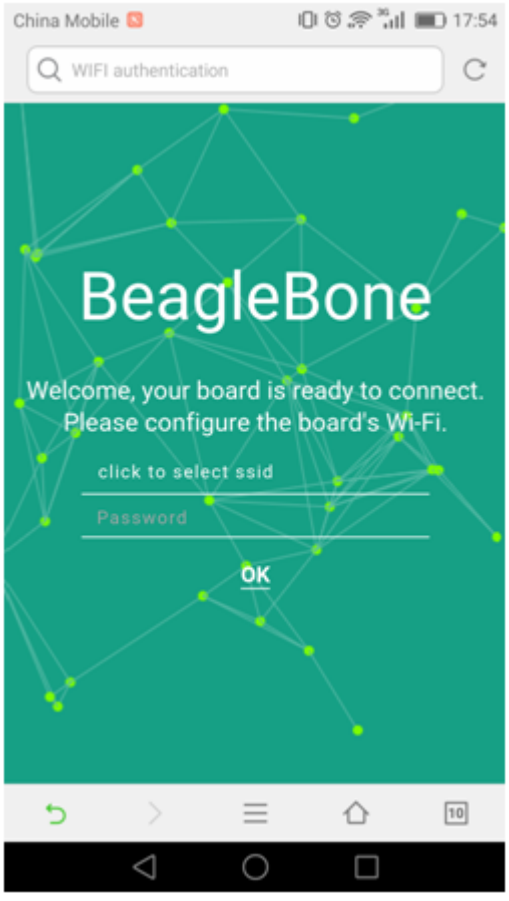
To begin editing programs that live on your board, you can use the Cloud9 IDE by click

STEP5. Connect your BBGW to Wi-Fi

Using your smart phone or computer to scan local Wi-Fi network and connect to the AP named "BeagleBone XXX"



After connection succeeded, it will head to the login page automatically. Select the SSID of your Wi-Fi and enter the passwd, click OK.

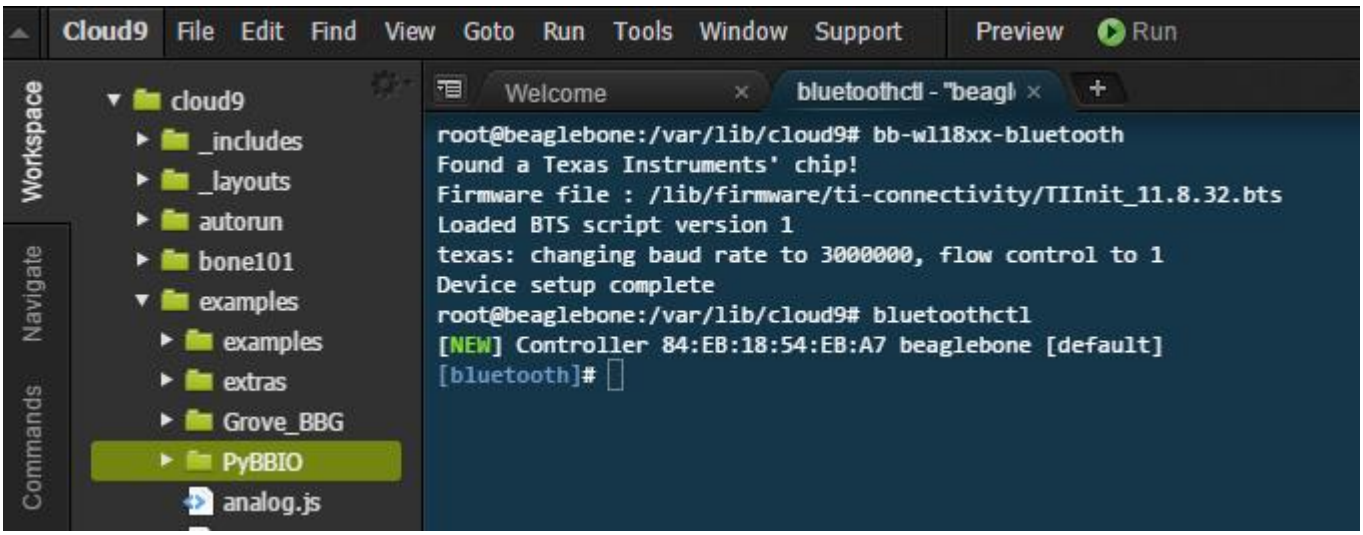


Now your BBGW is connected to Wi-Fi.

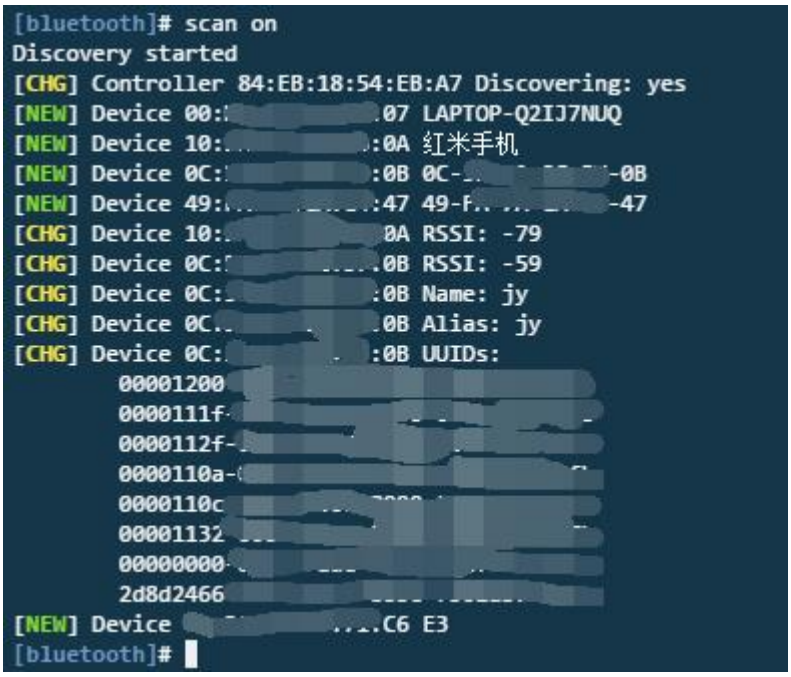
STEP6. Connect your BBGW to your Bluetooth Device

Connect to Cloud9 IDE and start a new terminal. Start the bluetooth config with the command:

```
bb-wl18xx-bluetooth
bluetoothctl
```

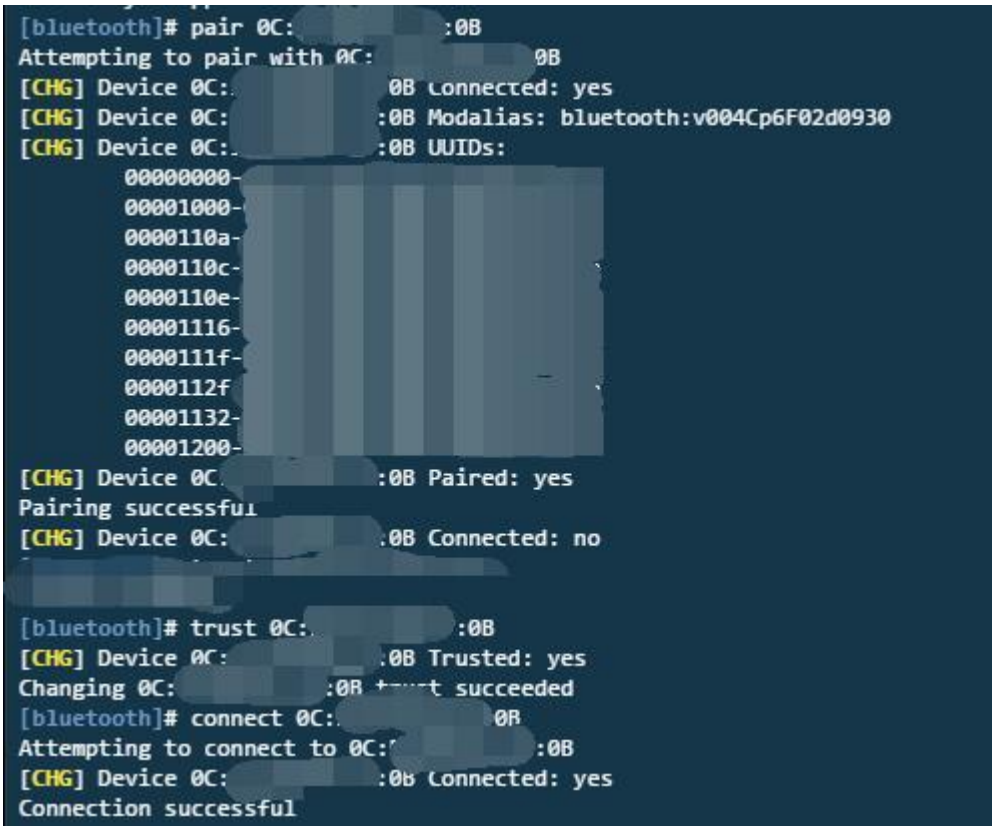


Type `scan on` to scan local bluetooth devices. My device named "jy" is found.



Copy the device mac address, then contect to the device with the command:

```
pair 0C:xx:xx:xx:xx:0B
trust 0C:xx:xx:xx:xx:0B
connect 0C:xx:xx:xx:xx:0B
```



Now your BBGW is connected to your bluetooth device. Type `quit` back to the terminal. Play music on BBGW, then you will hear music on your bluetooth speaker device.

Update to latest software

You need to update the board to latest software to keep a better performance, here we will show you how to make it step by step.

STEP1. Download the latest software image

First of all, you have to download the suitable image here.

Download the lastest image of BBGW

!!!Note Due to sizing necessities, this download may take about 30 minutes or more.

The file you download will have an **.img.xz** extension. This is a compressed sector-by-sector image of the SD card.

STEP2. Install compression utility and decompress the image

Download and install 7-zip.

!!!Note Choose a version that suitable for your system.

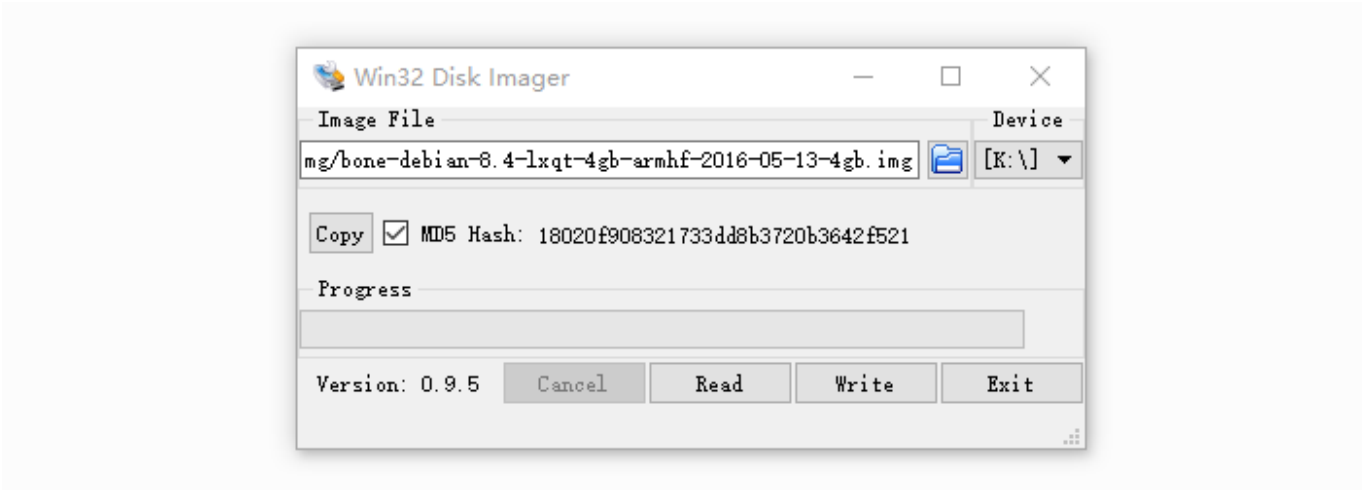
Use 7-zip to decompress the SD card **.img file**

STEP3. Install SD card programming utility

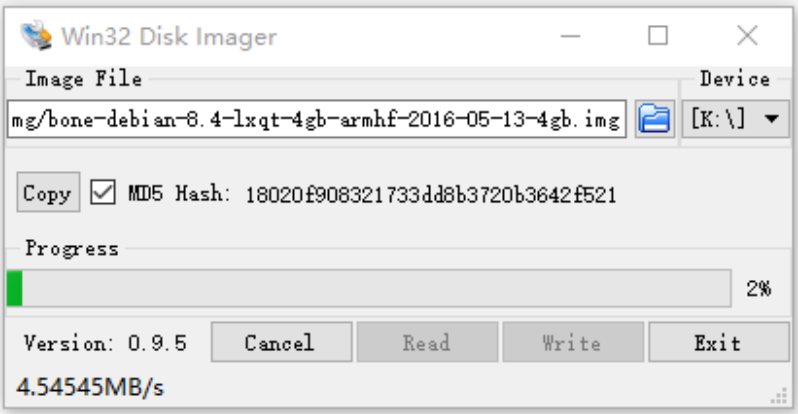
Download and install Image Writer for Windows. Be sure to download the binary distribution.

STEP4. Write the image to your SD card

You need a SD adapter to connect your microSD card to your computer at the first. Then use the software Image Write for Windows to write the decompressed image to your SD card.



Click on **Write** button, then the process is started.



!!!Note * You may see a warning about damaging your device. This is fine to accept as long as you are pointing to your SD card for writing. * You should not have your BeagleBone connected to your computer at this time. * This process may need up to 10 minutes.

STEP5. Boot your board off of the SD card

Insert SD card into your (powered-down first) board. Then the board will boot from the SD card.

!!!Note If you don't need to write the image to your on-board eMMC, you don't need to read the last of this chapter. Otherwise pleas go ahead.

If you desire to write the image to your on-board eMMC, you need to launch to the board, and modify a file.

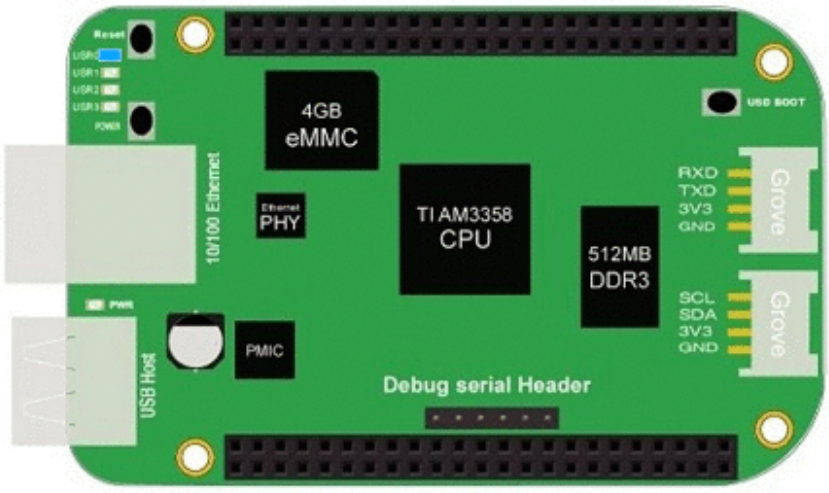
In **/boot/uEnv.txt**:

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

Change to:

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

Then you will find the 4 user led light as below:



!!!Note If you don't find the upper tracing light, please power down and power up the board.

When the flashing is complete, all 4 USRx LEDs will be **off**. The latest Debian flasher images automatically power down the board upon completion. This can take up to **10 minutes**. Power-down your board, remove the SD card and apply power again to be complete.

Program Grove Module with Mraa and UPM

We have provided Mraa library and UPM library to make it easy for developers and sensor manufacturers to map their sensors & actuators on top of supported hardware and to allow control of low level communication protocol by high level languages & constructs.

What are Mraa and UPM?

Mraa is a C/C++ library with bindings to Python, Javascript and Java to interface with the I/O on BBG, BBGW and other platforms, with a structured and sane API where port names/numbering matches the board that you are on. Use of Mraa does not tie you to specific hardware with board detection done at runtime you can create portable code that will work across the supported platforms. UPM is a high level repository for sensors that use MRAA. Each sensor links to MRAA and are not meant to be interlinked although some groups of sensors may be. Each sensor contains a header which allows to interface with it. Typically a sensor is represented as a class and instantiated.The constructor is expected to initialise the sensor and parameters may be used to provide identification/pin location on the board.

Install and update

Mraa and UPM are already installed in the system image of BBGW, so **you don't need to install it.** However if you want to update the library, or want to upgrade the library, use `apt-get update` and `apt-get upgrade` please. Refer to https://github.com/intel-iot-devkit/mraa and https://github.com/intel-iot-devkit/upm for more information.

Mraa Example

- light a led

```
import mraa
import time
#mraa.gpio60 = P9_14 = GPIO_50
led = mraa.Gpio(60)
led.dir(mraa.DIR_OUT)

while True:
    led.write(1)
    time.sleep(1)
    led.write(0)
    time.sleep(1)
```

- Grove - PIR Sensor

```
import mraa
import time
#mraa.gpio73 = P9_27 = GPIO_115
pir = mraa.Gpio(73)
pir.dir(mraa.DIR_IN)

while True:
    print (pir.read())
    time.sleep(1)
```

- Grove - Rotary Angle Sensor

```
import mraa
import time
#mraa.aio1 = AIN0
rotary = mraa.Aio(1)

while True:
    print(rotary.read())
    time.sleep(1)
```

- More Tutorials

Grove - 3-Axis Digital Accelerometer(±16g) Grove - Variable Color LED

Grove - Mini Fan Grove - PIR Motion Sensor Grove - Rotary Angle Sensor

Grove - Relay Grove - Sound Sensor

Grove - OLED Display 0.96” Grove - Light Sensor Grove - Temperature Sensor

Grove - GPS Grove - Button(P) Grove - Buzzer Grove - RTC v2.0

Mraa Map for BBGW

BBGW Mraa Gpio					
Mraa	phy	GPIO	Mraa	phy	GPIO
7	P8_07	GPIO_66	42	P8_42	GPIO_75
8	P8_08	GPIO_67	43	P8_43	GPIO_72
9	P8_09	GPIO_69	44	P8_44	GPIO_73
10	P8_10	GPIO_68	45	P8_45	GPIO_70
13	P8_13	GPIO_23	46	P8_46	GPIO_71
19	P8_19	GPIO_22	57	P9_11	GPIO_30
27	P8_27	GPIO_86	59	P9_13	GPIO_31
28	P8_28	GPIO_88	60	P9_14	GPIO_50
29	P8_29	GPIO_87	61	P9_15	GPIO_48
30	P8_30	GPIO_89	62	P9_16	GPIO_51
31	P8_31	GPIO_10	63	P9_17	GPIO_5
32	P8_32	GPIO_11	64	P9_18	GPIO_4
33	P8_33	GPIO_9	67	P9_21	GPIO_3
34	P8_34	GPIO_81	68	P9_22	GPIO_2
35	P8_35	GPIO_8	69	P9_23	GPIO_49
36	P8_36	GPIO_80	70	P9_24	GPIO_15
37	P8_37	GPIO_78	71	P9_25	GPIO_117
38	P8_38	GPIO_79	72	P9_26	GPIO_14
39	P8_39	GPIO_76	73	P9_27	GPIO_115
40	P8_40	GPIO_77	87	P9_41	GPIO_20
41	P8_41	GPIO_74	88	P9_42	GPIO_7

BBGW Mraa I2C			
Mraa	I2C	PIN	FUN
0	I2C1	P9_17	I2C1_SCL
		P9_18	I2C1_SDA
1	I2C2	P9_19	I2C2_SCL
		P9_20	I2C2_SDA

BBGW Mraa PWM		
Mraa	PWM	PIN
68	EHRPWM0A	P9_22
67	EHRPWM0B	P9_21
36	EHRPWM1A	P8_36
60		P9_14
62	EHRPWM1B	P9_16
34		P8_34
19	EHRPWM2A	P8_19
45		P8_45
13	EHRPWM2B	P8_13
46		P8_46
88	ECAPPWM0	P9_42

BBGW Mraa ADC	
Mraa	ADC
1	AIN0
2	AIN1
3	AIN2
4	AIN3
5	AIN4
6	AIN5
7	AIN6

BBGW Mraa UART				
MRAA	PIN	FUN	UART	DEV
0	P9_24 P9_26	UART1_TXD UART1_RXD	UART1	/dev/ttyO1
1	P9_21 P9_22	UART2_TXD UART2_RXD	UART2	/dev/ttyO2
2			UART3	/dev/ttyO3
3	P9_13 P9_11	UART4_TXD UART4_RXD	UART4	/dev/ttyO4
4	P8_37 P8_38	UART5_TXD+ UART5_RXD+	UART5	/dev/ttyO5

Grove for BBG

Grove is a modular, standardized connecter prototyping system. Grove takes a building block approach to assembling electronics. Compared to the jumper or solder based system, it is easier to connect, experiment and build and simplifies the learning system, but not to the point where it becomes dumbed down. Some of the other prototype systems out there takes the level down to building blocks. Good stuff to be learned that way, but the Grove system allows you to build real systems. It requires some learning and expertise to hook things up.

Below listed the Grove modules that work well with BBG.

SKU	Name	Interface	link
101020054	Grove - 3-Axis Digital Accelerometer(+16g)	I2C	link
101020071	Grove - 3-Axis Digital Accelerometer(+400g)	I2C	link
101020034	Grove - 3-Axis Digital Compass	I2C	link
101020050	Grove - 3-Axis Digital Gyro	Analog	link
101020081	Grove - 6-Axis Accelerometer&Compass v2.0	I2C	link
101020072	Grove - Barometer Sensor(BMP180)	I2C	link
104030010	Grove - Blue LED	I/O	link
101020003	Grove - Button	I/O	link

SKU	Name	Interface	link
111020000	Grove - Button(P)	I/O	link
107020000	Grove - Buzzer	I/O	link
104030006	Grove - Chainable RGB LED	I2C	link
101020030	Grove - Digital Light Sensor	I2C	link
103020024	Grove - Finger-clip Heart Rate Sensor	I2C	link
101020082	Grove - Finger-clip Heart Rate Sensor with shell	I2C	link
113020003	Grove - GPS	UART	link
104030007	Grove - Green LED	I/O	link
103020013	Grove - I2C ADC	I2C	link
103020006	Grove - I2C Hub	I2C	link
101020079	Grove - IMU 10DOF	I2C	link
101020080	Grove - IMU 9DOF v2.0	I2C	link
101020040	Grove - IR Distance Interrupter	I/O	link
104030011	Grove - OLED Display 0.96''	I2C	link
104030008	Grove - OLED Display 1.12''	I2C	link
104030005	Grove - Red LED	I/O	link
103020005	Grove - Relay	I/O	link
316010005	Grove - Servo	I/O	link
101020023	Grove - Sound Sensor	Analog	link
101020004	Grove - Switch(P)	I/O	link
101020015	Grove - Temperature Sensor	Analog	link
101020019	Grove - Temperature&Humidity Sensor Pro	Analog	link

Cape for BBG

You will need some expansion board when you start a project. There're many cape for BBG already, they include LCD display, motor driver as well as HDMI expansion etc. Below is some of them recommend.

Grove Cape	Motor Bridge Cape	HDMI Cape

Grove Cape	Motor Bridge Cape	HDMI Cape
		
GET ONE NOW!	GET ONE NOW!	GET ONE NOW!
Grove Cape	**5 Inch LCD**	**7 Inch LCD**
		
GET ONE NOW!	GET ONE NOW!	GET ONE NOW!

References

There're many references to help you to get more information about the board.

- BeagleBoard Main Page
- BeagleBone Green Wireless info at BeagleBoard page
- BeagleBoard Getting Started
- Troubleshooting
- Hardware documentation
- Projects of BeagleBoard

FAQs

Q1: How to connect your BBGW to your Bluetooth Device with Debian 9.5 2018-10-07 4GB image?

A1: Please follow below instructions. thanks.

```
udebian@beaglebone:~$ uname -a
Linux beaglebone 4.14.71-ti-r80 #1 SMP PREEMPT Fri Oct 5 23:50:11 UTC 2018 armv7l
GNU/Linux
debian@beaglebone:~$ bluetoothctl --agent
[NEW] Controller A4:D5:78:6D:6F:E0 beaglebone [default]
Agent registered
[bluetooth]# scan on
Discovery started
[CHG] Controller A4:D5:78:6D:6F:E0 Discovering: yes
```

```
[NEW] Device C0:72:BC:0A:E6:1E HB7910703602
[NEW] Device E8:19:C4:6B:03:5C YONGNUO LED
[NEW] Device C0:F6:1E:02:0A:1E HB7690138998
```

Resources

- **[PDF]** BeagleBone_Green_Wireless Schematic
- **[Zip]** AM335X Datasheet
- **[3D]** BBGW 3D Model
- **[PDF]** BBGW Mechincal Drawing

Project

BeagleBone Green Wireless Irrigation Control: BeagleBone Green Wireless Irrigation Control using HTML5, WebSockets, and Ecmascript 6.

https://www.hackster.io/Greg-R/beaglebone-green-wireless-irrigation-control-ce7c4b/

Tech Support

Please submit any technical issue into our forum or drop mail to techsupport@seeed.cc.