

This manual is a guide to help build applications on Intel Edison

Things You Need

- Host Machine with Ubuntu 14.04 LTS connected to Internet
 - Intel Edison and Arduino Breakout Kit
 - 12V DC Power Adapter (Working Range : 7V to 15 V). **Also can be powered by 5V from laptop. Check current rating and USB standard.**
 - MicroUSB Cable (2)
 - Grove Kit
-
- Intel® Edison Mini Breakout Kit is also available from Sparkun. But, we focus on Arduino Break out Kit

Connecting Your Host Machine to Internet

- Check internet availability using browser

Know your EDISON



Key features of Intel Edison

- Intel® Atom™ system-on-a-chip (SoC) based on leading-edge 22 nm [Silvermont](#) microarchitecture including a dual-core CPU and [single core microcontroller](#) ([MCU](#))
- Integrated Wi-Fi, Bluetooth LE, memory, and storage
- Support for more than 30 industry-standard I/O interfaces via a 70-pin connector
- Support for Arduino, Yocto Linux, Ubilinux, Python, Node.js, and Wolfram

Key features of Arduino Expansion Kit



- Compatible with Arduino Uno (except only 4 PWM instead of 6 PWM).
- 20 digital input/output pins including 4 pins as PWM outputs
- 6 analog inputs
- 1 UART (RX/TX)
- 1 I2C
- 1 ICSP 6-pin header (SPI)
- [Pin Headers](#) Micro USB device connector OR (via mechanical switch) dedicated standard size USB host Type-A connector. [Try Type-A USB host with say Pendrive](#)
- [Pin Headers](#) Micro USB device (connected to UART)
- SD Card connector
- DC power jack (7V – 15V DC input @ 500mA)

Assembling the Intel Edison Chip with the Arduino expansion board

Requirements

- Intel Edison module
- Arduino expansion board
- **Note:** An external power supply is the preferred way of powering the Intel Edison board. However, you can power your board over USB if you do not have an external power supply. For details, see the Powering your board over USB section.

Assemble your board

1. Place the Intel Edison module within the white outline on your expansion board, lining up the holes on the module with the screws on the expansion board.



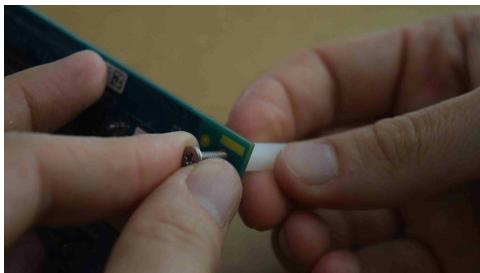
2. Press down on the module just below the words **What will you make?** until you feel a snap. When you turn the attached module and expansion board on their side, both pieces should fit evenly and sit in parallel with each other.



3. Use the two hex nuts (included in the package) to secure the module to the expansion board.



4. Insert a screw in the corner hole and attach the plastic spacer.



5. Repeat for the other three corner spacers.

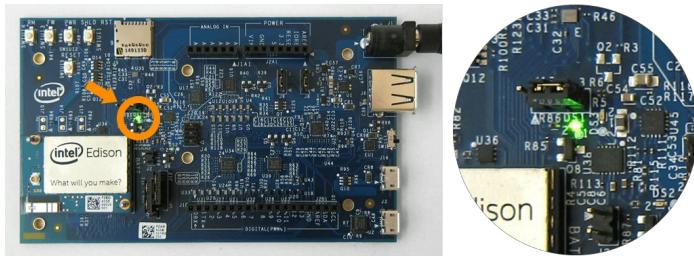


Connect the board to your system

1. Plug in the 12 V DC power supply.



2. A green LED should light up on the expansion board. If it doesn't, check your connection.



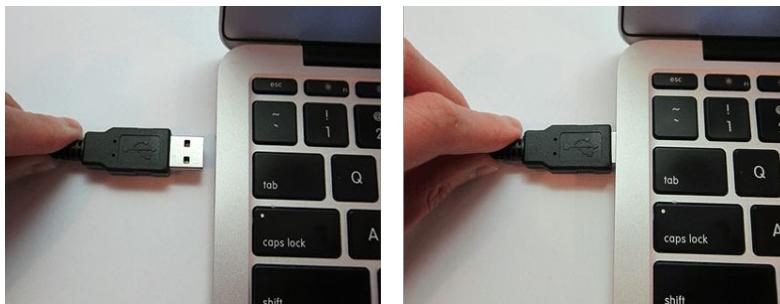
3. Find the microswitch in between the USB ports on the expansion board. Switch the microswitch down towards the micro-USB ports, if it isn't already.



4. Plug in one of the micro-USB cables to the **middle** USB connector on the expansion board.



5. Plug in the other end of the USB cable to your computer.



How do you know when the board is ready?

You will know that your board is fully initialized when your computer mounts a new drive (much like inserting a SD card into your computer). If you do not see a new drive, or the LED light (DS1 on the Arduino expansion board) is occasionally turning on and off, check the connection of your power supply.

Flashing Edison

Create a folder 'workshop_edison' in the 'home' directory.

1. Click on "Release 2.1 Yocto* complete image" from Intel Edison® Board "Firmware Software Release 2.1" section in the [link](#) here. It Downloads a zip file called 'edison-image-ww25.5-15' and place it in the 'workshop_edison' folder. Right Click on 'edison-image-ww25.5-15' and click on 'Extract here'

2. **Remove the USB cable first and then remove Power Supply from the Intel Edison Board.**

Install the Dependencies: **sudo apt-get install dfu-util**

```
# cd workshop_edison/edison-image-ww25.5-15  
# sudo ./flashall.sh
```

Wait till you get the message " Please Plug and reboot the board "

Now plug your USB cable and power cable. Wait for a while till it completes.

Firmware update takes relatively longer time in VM. Please be patient till you get the above messages

You are done with the Firmware Update! :)

Install Arduino (Linux)

1. To check if you have Java installed, open a terminal and enter the command:

```
# java
```

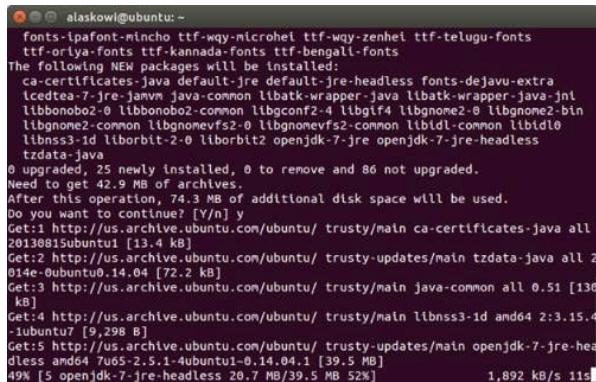


```
alaskowi@ubuntu:~$ java
The program 'java' can be found in the following packages:
* default-jre
* gcj-4.8-jre-headless
* openjdk-7-jre-headless
* gcj-4.6-jre-headless
* openjdk-6-jre-headless
Try: sudo apt-get install <selected package>
alaskowi@ubuntu:~$ sudo apt-get install default-jre
```

If you see the above message, you do not have Java installed and you will need to install it. To install the Java package, enter the command:

```
sudo apt-get install default.jre
```

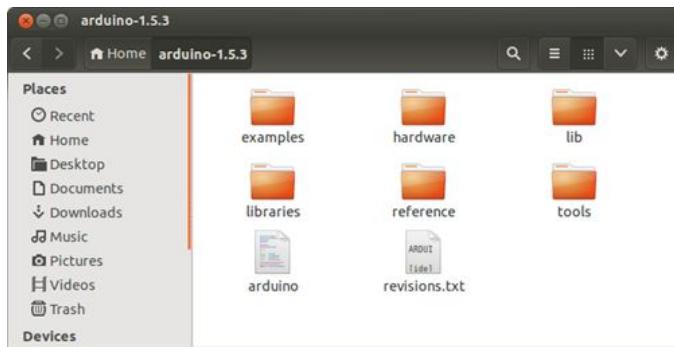
You may be prompted to enter your user password



```
alaskowi@ubuntu:~$ fonts-ipafont-mincho ttf-wqy-microhei ttf-wqy-zhenhei ttf-telugu-fonts
ttf-orlja-fonts ttf-kannada-fonts ttf-bengali-fonts
The following NEW packages will be installed:
ca-certificates-java default-jre default-jre-headless fonts-dejavu-extra
icedtea-7-jre-javaw java-common libatk-wrapper-java libatk-wrapper-java-jni
libbonobo2-0 libbonobo2-common libgconf2-4 libgif4 libgnome2-0 libgnome2-bin
libgnome2-common libgnomevfs2-0 libgnomevfs2-common libidl-common libidl0
libnss3-1d liborbit2-2.0 liborbit2 openjdk-7-jre openjdk-7-jre-headless
tzdata-java
0 upgraded, 25 newly installed, 0 to remove and 86 not upgraded.
Need to get 42.9 MB of archives.
After this operation, 74.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty/main ca-certificates-java all
2013081Subntul [13.4 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main tzdata-java all 2
014e-0Ubuntu0.14.04 [72.2 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu/ trusty/main java-common all 0.51 [130
kB]
Get:4 http://us.archive.ubuntu.com/ubuntu/ trusty/main libnss3-1d amd64 2:3.15.4
-lubuntu7 [9.298 B]
Get:5 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main openjdk-7-jre-hea
dless amd64 7u05-2.5.1-4ubuntu1-0.14.04.1 [39.5 MB]
49% [5 openjdk-7-jre-headless 28.7 MB/39.5 MB 52%]
```

2. Download the Arduino IDE from the [Arduino Software page](#). Under the **Arduino Software 1.6.0 - Intel 1.0.4 section**, download the Arduino version for your operating system.
3. Move it to the folder 'workshop_edison'

4. Right Click and on the .txz file and click ‘Extract here’
5. The extracted folder should contain a file named arduino, as well as several folders.



6. Open up a new Terminal window.

Navigate to the Arduino IDE folder. In this example, the command will be

```
# cd home/<system name>/workshop_edison/arduino-x.x.x/
```

where *x.x.x* is the Arduino IDE version number you downloaded

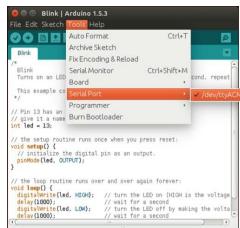
7. To run Arduino , enter the command:

```
. /arduino.
```

Blinking an LED with the Arduino IDE on the Intel® Edison board

1. Run Arduino.
2. Open the LED blink example sketch by choosing **File > Examples > 1.Basics > Blink**.
3. Choose **Tools > Board> Intel Edison**

Find the port for your board, which is likely to be Tools-> Serial Port-> **/dev/ttyACM***.

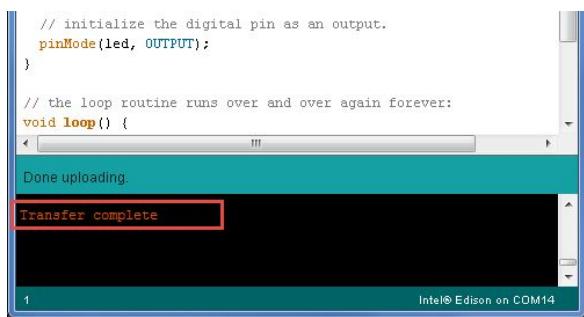


In case you do not see the Com port :

Type **sudo chmod a+rwx /dev/ttyACM***

4. Once you have selected your port, click on '**upload**' the sketch to your board.

You should see **Done Uploading** and **Transfer complete** messages when your sketch has uploaded.



The DS2 LED on your board should now blink on or off every second.



Setting up a serial terminal on a system with Linux*

1. Plug in your second USB cable to the **edge** USB connector on the board.



2. Plug the other end of the USB cable in to your computer.



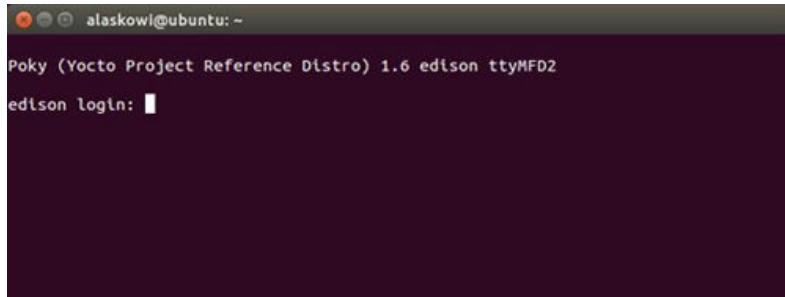
1. If you do not have the screen shell session manager installed, open a new Terminal window. Enter the command:

```
sudo apt-get install screen
```

2. If prompted, enter your password.
3. To connect to the board, enter the command (where ttyUSB0 is your connected device):

```
sudo screen /dev/ttyUSB0 115200
```

If prompted, enter your password to continue.
4. Press **Enter** twice. A login screen is displayed.



A terminal window titled "alaskowi@ubuntu: ~". The title bar shows three icons: a red circle, a grey circle, and a black circle. The window content displays the text "Poky (Yocto Project Reference Distro) 1.6 edison ttyMFD2" followed by the prompt "edison login: [REDACTED]" where the password has been obscured.

5. At the login prompt, type **root** and press **Enter**.
6. Press **Enter** when prompted for a password. The following screen is displayed:



A terminal window showing a root prompt. The title bar is not visible. The window content displays the text "root@Edison1:~# [REDACTED]" where the password has been obscured.

Now that you have set up a serial terminal for your board, continue by connecting your board to a network.

Connecting your Intel Edison board using Wi-Fi

This guide contains steps to set up network access to your Intel® Edison board and obtain an IP address.

Set up Wi-Fi

1. Establish a serial communication session with your board. (Done above)
2. To configure your Wi-Fi, enter the command:

```
configure_edison --wifi
```

(If you get an error saying **configure_edison: not found**, you need to update your firmware.)

3. When asked if you want to set up Wi-Fi, type **Y** and press **Enter**.
4. Your board will scan for Wi-Fi networks for 10 seconds. When it is finished, a list of available networks will be displayed. If you don't see any networks, enter **0** to rescan
5. Choose the network you would like to connect to, type the corresponding number from the list, and press **Enter**. To confirm your entry, type **Y** and press **Enter**. In this example, to connect to the **kafka** network, enter **16**.

```
13 :     STSPDX
14 :     belkin.19a
15 :     WiFiRSU_953c2
16 :     kafka

Enter 0 to rescan for networks.
Enter 1 to input a hidden network SSID.
Enter a number between 2 to 16 to choose one of the listed network SSIDs: 16
```

6. If your network requires a password or other information, enter the appropriate network credentials.
7. The board will attempt to make a connection to the network. When you see a **Done** message, your board is connected to a Wi-Fi network.

```
Enter a number between 2 to 16 to choose one of the listed network SSIDs: 16
Is kafka correct? [Y or N]: y
Password must be between 8 and 63 characters.
What is the network password?: *****
Initiating connection to kafka. Please wait...
Attempting to enable network access... Please check 'wpa_cli status' after a minute
e to confirm.
Done. Please connect y
to http://10.0.0.26
root@myedison:~#
```



8. Note the IP Address, as shown in the image above. This is your board's IP Address. Alternately, enter the command:

9. ifconfig

```
root@raspberrypi:~# ifconfig  
eth0      Link encap:Ethernet HWaddr b8:27:eb:00:00:00  
          brd ff:ff:ff:ff:ff:ff  MTU:1500 Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
wlan0     Link encap:Ethernet HWaddr fc:c2:de:3e:91:b4  
          inet addr:192.168.0.105  Bcast:0.0.0.0  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:9 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:45 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:2664 (2.6 KiB)  TX bytes:11045 (10.7 KiB)
```

10. Make note of your **wlan0** IP address, as shown above.

11. To verify connectivity, you may want to ping your board from another computer on the same network using the IP Address obtained above. Alternately, you can try accessing your board by typing in your IP Address into a browser of another computer on the same network.

If you are having problems connecting, try running the following commands in a serial communication session with your board:

```
ifconfig usb0 down  
ifconfig wlan0 down  
ifconfig usb0 up  
ifconfig wlan0 up
```

Interfacing basic grove sensors (Hardware Components for IoT)



Download the library for arduino to work with the sensor kit from the [link](#). Click on **Download Zip**

Unzip the downloaded file and copy it to the arduino libraries folder, under the

home/workshop_edison/arduino-1.6.0+Intel/Libraries directory.

Cut the folder from:

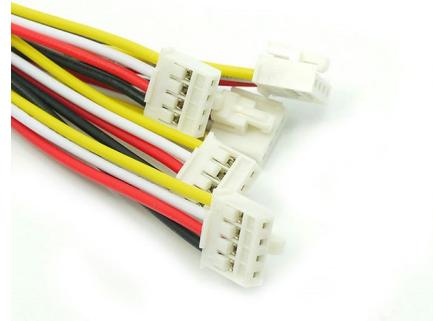
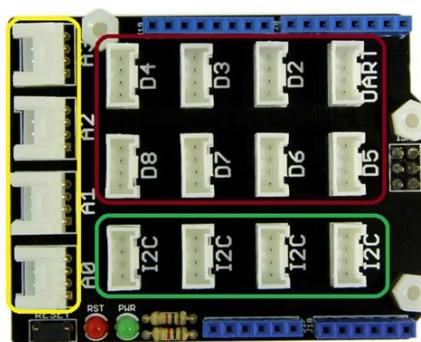
workshop_edison>arduino-1.6.0>libraries>Sketchbook_Starter_kit_V2.0-master>libraries> Grove_LCD_RGB_Backlight

And paste it in :

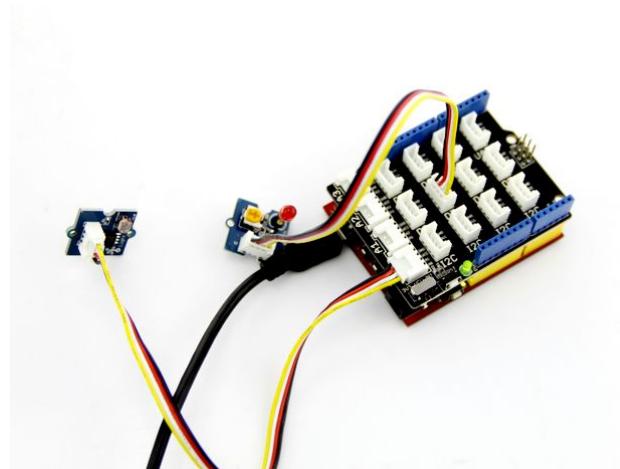
workshop_edison>arduino-1.6.0>libraries>

Restart the Arduino IDE.

1. Grove Base Shield



2. Grove – Light Sensor

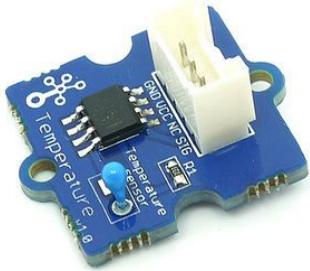


Example program 1 :

This example will turn on an LED when the light intensity based on a preset threshold.

- Connect jumper cable (Vcc, GND, NC, data) of light sensor to A0 on the Grove Shield
- Connect the jumper cable of LED module (Vcc, GND, NC, data) to D7 (Vcc, GND, d7, d8) on the Grove shield.
- Check that the voltage switch on the Grove Shield is towards 5v
- Connect the Arduino compatible Grove shield to the Edison board.
- Power up the Edison and connect the USB cable and select the com port and board.
- Go to **File>Examples>Sketchbook_Starter_Kit_V2.0-master**
Choose **Grove_Light_Sensor**.
- Check Serial port and board
- Click on '**Verify**' then '**Upload**'. Open the Serial monitor.
- Click on Serial Monitor to see the Sensor Reading

3. Grove – Temperature Sensor



Example Program 2 :

Remove the Light Sensor from the A0 port and Connect the Temperature sensor into A0 port.

This example shows how to convert the raw output of the sensor into the specific temperature value. You can see the output data in Celsius in the Serial monitor

- Connect the jumper cable of Temperature Sensor (Vcc, GND, NC, Data) to A0 on the Grove shield
- Go to **File>Examples>Sketchbook_Starter_Kit_V2.0-master**
Choose **Grove_Temperature_Sensor**
- Check Serial com port and board and **Upload**
- Click on **Serial Monitor** to see the Sensor Reading

Tips :

The intrinsic function to find the temperature is a variation of the Steinhart-Hart equation, a widely used approximation that gives T (temperature) as a function of R (resistance of the thermistor). The original equation is:

$$\frac{1}{T} = A + B \ln(R) + C(\ln(R))^3$$

where:

- T is the temperature (in kelvins)
- R is the resistance at T (in ohms)
- A , B , and C are the Steinhart–Hart coefficients which vary depending on the type and model of thermistor and the temperature range of interest.

All three parameters in the equation above can be described by one parameter: B . Within the industry, people characterize thermistors by B value, as defined in our example. So the built-in function is actually a variation based on the B parameter equation.

4. Grove - LCD RGB Backlight



- 16X2 Programmable Multicolor display
- Interface : I2C

NOW LETS BRING THE 3 COMPONENTS TOGETHER.....

Connect the room temperature and the presence of light can be coded to be displayed on the LCD screen to the I2C port.

Connect the Light Sensor to the A1 port, LCD display to the

- Connect the LCD Display to the I2C port on the Grove shield.
- Copy the code below to a new Arduino sketch and click File--Save As-- 'onLCD'--Ok
- Check Serial com port and board and Upload
- Sensor readings are displayed on the LCD display

Example program 3

```

#include <math.h>
#include <Wire.h>
#include "rgb_lcd.h"
int a;
float temperature;
int B=3975;           //B value of the thermistor
float resistance;
rgb_lcd lcd;
const int thresholdvalue=25; //The threshold for which the LED should turn on.
float Rsensor;
const int colorR = 0;
const int colorG = 255;
const int colorB = 0;
void setup()
{
    Serial.begin(9600); // set up the LCD's number of columns and rows:
    lcd.begin(16, 2);
    lcd.setRGB(colorR, colorG, colorB);
}
void loop()
{
    lcd.setCursor(0,0);
    a=analogRead(A0);
    resistance=(float)(1023-a)*10000/a; //get the resistance of the sensor;
    temperature=1/(log(resistance/10000)/B+1/298.15)-273.15;
    //convert to temperature via datasheet ;
    lcd.print("Temp :");
    lcd.print(temperature);

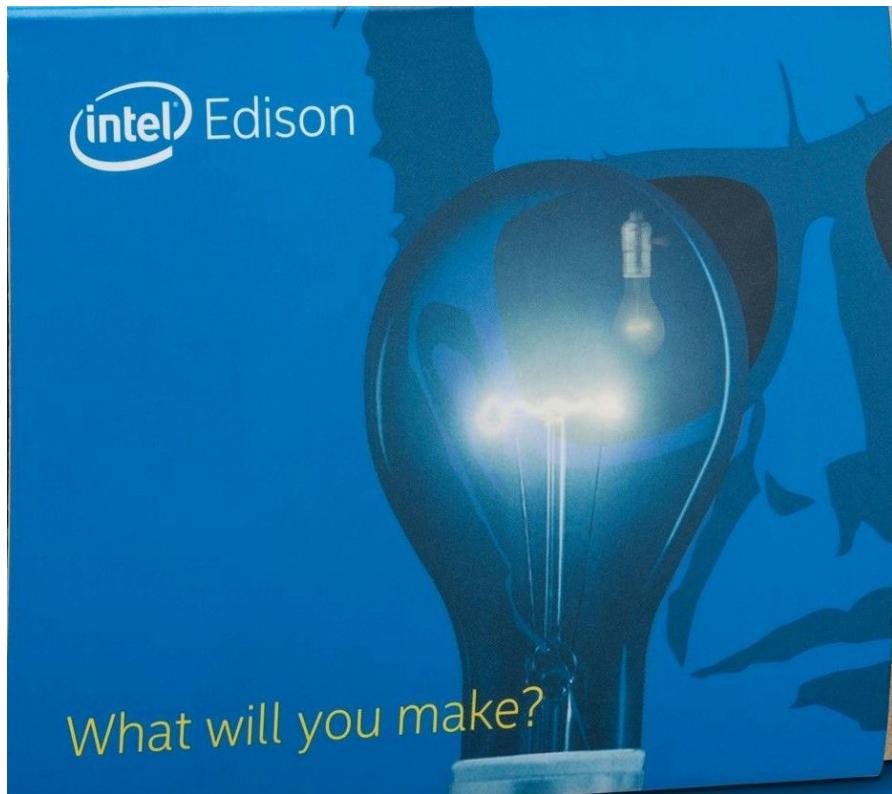
    //display light sensor output
    lcd.setCursor(0,2);
    int sensorValue = analogRead(A1);
    Rsensor=(float)(1023-sensorValue)*10/sensorValue;
    if(Rsensor<thresholdvalue)
    {
        lcd.print("Light on");
    }
    else
        lcd.print("Light off");

    delay(500);
    lcd.clear();
}

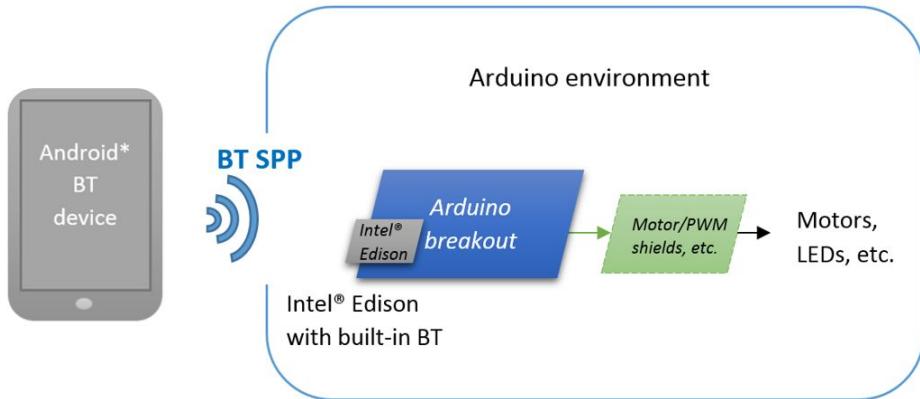
```

Disconnect all the sensors and grove modules, remove USB cable and Power off the board

**Congratulations !! You have
completed Level 1**



Bluetooth on Intel Edison using arduino IDE



In the shell of Intel Edison, ie
root@edison:~#

1. Type the following command

```
mkdir /home/root/bluetooth
```

```
cd /home/root/bluetooth
```

```
wget https://software.intel.com/sites/default/files/managed/6c/16/bluetooth-service.tar.gz
```

2.

```
tar -xvf bluetooth-service.tar.gz
```

3.

```
cp bluetooth-spp-pin.service /lib/systemd/system
```

4. Enable the systemd service

```
systemctl enable bluetooth-spp-pin
```

5. Reboot your device

```
reboot
```

6. Double check the service

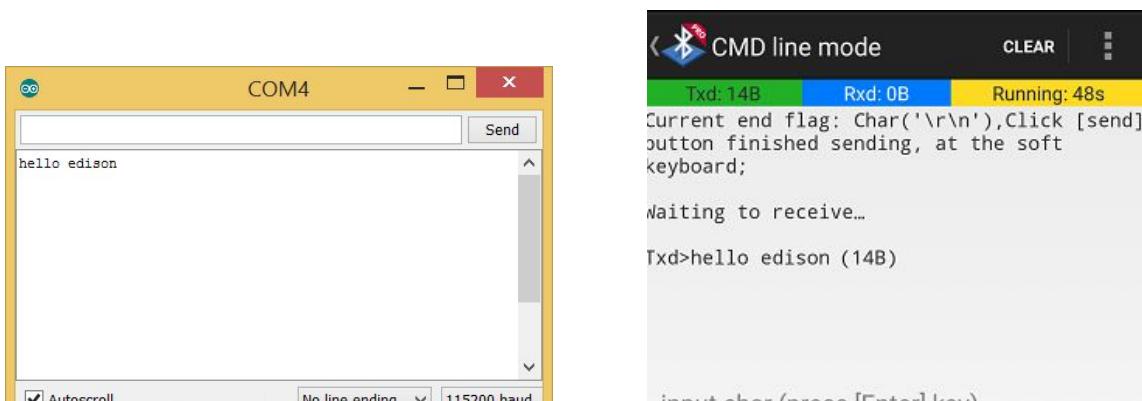
```
sudo systemctl status bluetooth-spp-pin
```

Setup for Edison Arduino sketch:

1. Download the library [Intel-Edison-BT-SPP-Library.zip](#).
2. Extract to your Arduino library path, `workshop_edison>arduino-1.6.0>libraries>`
3. Restart your Intel® Arduino IDE v1.6.0 or later version
4. Open the Example `bt_test` under File->Examples->Intel Edison BT SPP Driver Library
5. Verify and Upload the sketch to your Edison
6. Open Serial Monitor so we can check the output after we send something to it.

Setup for your Phone:

1. Download and install any BlueTooth SPP APP from PlayStore. [Bluetooth spp tools pro](#).
2. Turn on BT on your phone and connect to your Edison module
3. Enter the PIN code, the default PIN is 8888, feel free the change it later at line 70 of `bluetooth-pin-service.py`
4. Connect to your Edison in the APP, for ex: myedison
5. Select Byte Stream Mode. Send something to your Edison, you should be able to see the result on Serial Monitor of Arduino IDE



Uploading Data to dweet.io from intel Edison using Arduino IDE

```
// post to dweet.io through proxy network

#include <WiFi.h>
#include <SPI.h>
unsigned long prev_millis=0; // to hold delay in data reading

unsigned long lastConnectionTime = 0;           // last time you connected to the server, in
milliseconds
const unsigned long postingInterval = 10L * 1000L; // delay between updates, in milliseconds
// the "L" is needed to use long type numbers

WiFiClient client; // for post_data();

int analogIn=A0;

char ssid[] = "CXC";           // Router DDID
char password[] = "cxcatiisc"; // router password

WiFiServer server(80);

void setup()
{
  Serial.begin(9600);           // initialize serial communication
  wifi_connect();
}

void loop()
{
  if (millis() - lastConnectionTime > postingInterval)
    post_data();
}

void wifi_connect()
{
```

```

//*****
***// BELOW IS WIFI CONNECTION SET UP
//*****
**
Serial.print("Attempting to connect to Network named: "); // attempt to connect to Wifi network:
Serial.println(ssid); // print the network name (SSID);
WiFi.begin(ssid, password); // Connect to WPA/WPA2 network.
while ( WiFi.status() != WL_CONNECTED)
{ Serial.print("."); delay(300); } // print dots while we wait to connect

Serial.println("\nYou're connected to the network");
Serial.println("Waiting for an ip address");
Serial.println("\nIP Address obtained");
printWifiStatus(); // you're connected now, so print out the status
Serial.println("Starting webserver on port 80");
server.begin(); // start the web server on port 80
Serial.println("Webserver started!");
}

```

```

void printWifiStatus() {
    // print the SSID of the network you're attached to:
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    // print your WiFi IP address:
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);

    // print the received signal strength:
    long rssi = WiFi.RSSI();
    Serial.print("signal strength (RSSI):");
    Serial.print(rssi);
    Serial.println(" dBm");
    // print where to go in a browser:
    Serial.print("To see this page in action, open a browser to http://");
    Serial.println(ip);
}

```

//A way to check if one array ends with another array

```

boolean endsWith(char* inString, char* compString)
{
int compLength = strlen(compString);
int strLength = strlen(inString);

//compare the last "compLength" values of the inString
    int i;
    for (i = 0; i < compLength; i++)
    {
        char a = inString[(strLength - 1) - i];
        char b = compString[(compLength - 1) - i];
        if (a != b) {
            return false;
        }
    }
    return true;
}

void startup_led(){ // LED BLINKS ON start UP
for(int l_b=0; l_b<3; l_b++)
{
    digitalWrite(13, HIGH);
    delay(500);
    digitalWrite(13, LOW);
    delay(500);
}
}

void post_data() {

// close any connection before send a new request.
// This will free the socket on the WiFi shield
client.stop();

int analogVal = analogRead(analogIn);
String s="POST http://www.dweet.io/dweet/for/a_trial?hello=";
s.concat(analogVal);
char dweetServer[]="www.dweet.io";
int dweetport=80;

```

```
if (client.connect(dweetServer,dweetport)){
    Serial.println("connected to dweet"); Serial.println(s);
    client.println(s);
    client.println("Host: www.dweet.io");
    client.println("Connection: close");
    client.println();
}
else
    Serial.println("connection failed");

lastConnectionTime=millis();

}
```

Loading Debian (Ubilinux) on the Edison

If you want more features and a full-fledged [package manager](#), it is worth installing a different version of Linux on the Edison.

IMPORTANT: You can **potentially brick your Edison** by following this tutorial. Do so at your own risk! Be especially careful at the part where it says to not unplug your Edison for at least 2 minutes. If you do brick your Edison, read the [Troubleshooting section](#) for possible solutions.

Install Ubilinux

We need to download the Ubilinux image, which we will install on the Edison, and some software depending on your current operating system.

IMPORTANT: Make sure the Edison is not powered or connected to your host computer at this time.

Download the Ubilinux image by navigating to <http://www.emutexlabs.com/ubilinux> and click on “ubilinux for Edison.”

Move it to the ‘**workshop_edison**’ directory.

Open a terminal and navigate to your ‘**workshop_edison**’ directory.

```
# cd workshop_edison
```

Extract the contents of the downloaded file, right click on the file and click ‘**Extract here**’

Note that the version number (XXXXXX) will be whichever version you downloaded (e.g. ubilinux-edison-141030.tar.gz).

Move into the “toFlash” directory:

```
# cd toFlash
```

Run the install script:

```
# sudo ./flashall.sh
```

You will see a message like “Now waiting for dfu device.” At that, plug in your Edison (Use J16 for the Intel Arduino and Mini-Breakout boards. Use “OTG” for the SparkFun Edison Base Block). You should see the script start to flash the Edison in the terminal. Wait while that finishes (it could take a few minutes).

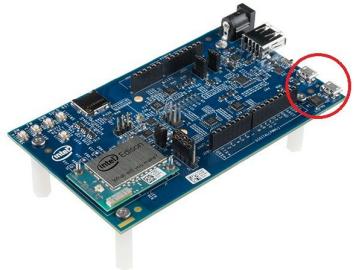
```
sgmustadio@shiva: ~/Downloads/toFlash
[25.0 kB]
Fetched 25.0 kB in 0s (122 kB/s)
Selecting previously unselected package dfu-util.
(Reading database ... 197141 files and directories currently installed.)
Unpacking dfu-util (from .../dfu-util_0.5-1_amd64.deb) ...
Processing triggers for man-db ...
Setting up dfu-util (0.5-1) ...
sgmustadio@shiva:~/Downloads/toFlash$ sudo ./flashall.sh
Using U-Boot target: edison-blank
Now waiting for dfu device 8087:0a99
Flashing IFWI
#####
[#####] finished!
#####
[#####] finished!
Flashing U-Boot
#####
[#####] finished!
Flashing U-Boot Environment
#####
[#####] finished!
Flashing U-Boot Environment Backup and rebooting to apply partiton changes
#####
[#####] finished!
Now waiting for dfu device 8087:0a99
Flashing boot partition (kernel)
#####
[#####] finished!
Flashing rootfs, (it can take up to 5 minutes... Please be patient)
[
```

Once the flashing process is complete, you will get a message like “Your board needs to reboot twice to complete the flashing procedure, please do not unplug it for 2 minutes.” You probably want to wait that 2 minutes (**I know you would not want to risk bricking your Edison**).

Congratulations! You have successfully installed the UBILINUX!

Log Into Ubilinux

Once you have flashed your Edison, you can log in over a serial connection.



IMPORTANT: You are using the Intel Arduino board, plug in the other USB cable at this time (you should have 2 USB cables running from your Edison to your computer).

You will need to use both USB ports on the Arduino Breakout board

Open a terminal and connect using the screen command:

sudo screen /dev/ttyUSB0 115200

Note: that your USB serial device may not be exactly ttyUSB0. If you have other USB serial devices plugged in, it could be ttyUSB1, ttyUSB2, and so on.

Press 'enter' to see the login screen.

Enter the default login credentials:

ubilinux login: root

Password: edison

You should now be logged into
Ubilinux.

A screenshot of a terminal window. The title bar says "sgmustudio@shiva: ~". The window content shows:

```
sgmustudio@shiva: ~
Debian GNU/Linux 7 ubilinux ttyMFD2
ubilinux login: [REDACTED]
```

The terminal window has a dark background and light-colored text.

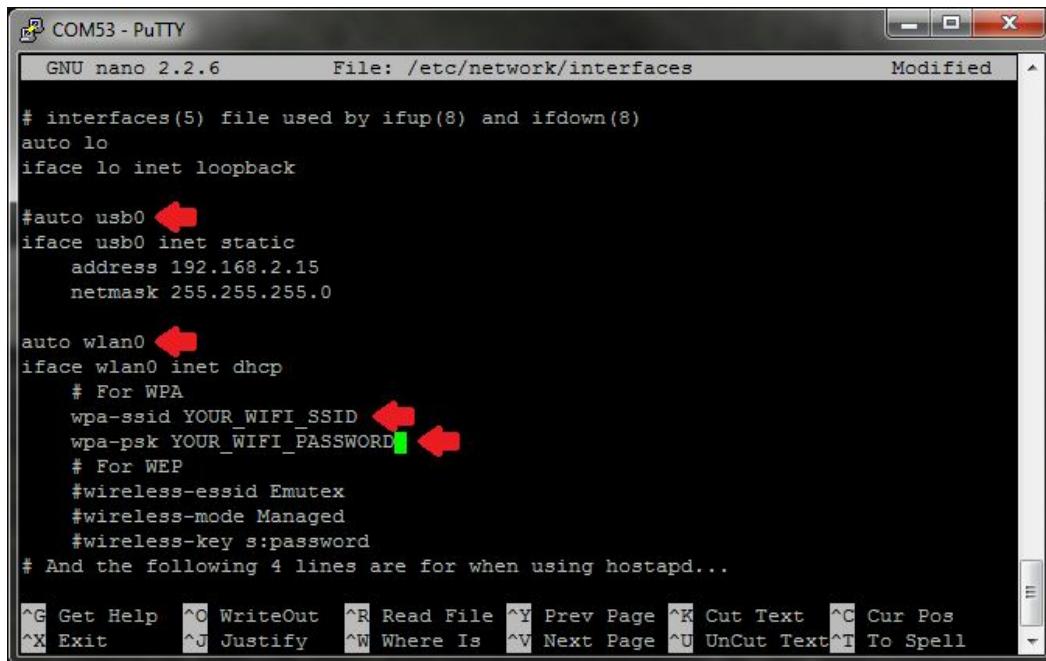
Enable WiFi:

The easiest way to get on a local WiFi Access Point is to store the SSID and password in a file. Use your favorite Linux text editor to open the interfaces file:

```
# nano /etc/network/interfaces
```

Navigate down and remove the “#” in front of “auto wlan0” so that our Edison will try to automatically connect to WiFi on boot.

Go down to the “wpa-ssid” and “wpa-psk” lines. Change the default SSID and password (PSK) to your desired WiFi SSID and password. Press ‘Ctrl-X’ to exit and press ‘y’ to save the file when prompted.



```
COM3 - PuTTY
GNU nano 2.2.6      File: /etc/network/interfaces      Modified

# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

#auto usb0 ←
iface usb0 inet static
    address 192.168.2.15
    netmask 255.255.255.0

auto wlan0 ←
iface wlan0 inet dhcp
    # For WPA
    wpa-ssid YOUR_WIFI_SSID ←
    wpa-psk YOUR_WIFI_PASSWORD ←
    # For WEP
    #wireless-essid Emutex
    #wireless-mode Managed
    #wireless-key s:password
    # And the following 4 lines are for when using hostapd...
    ^G Get Help  ^C 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
```

Test WiFi Connection

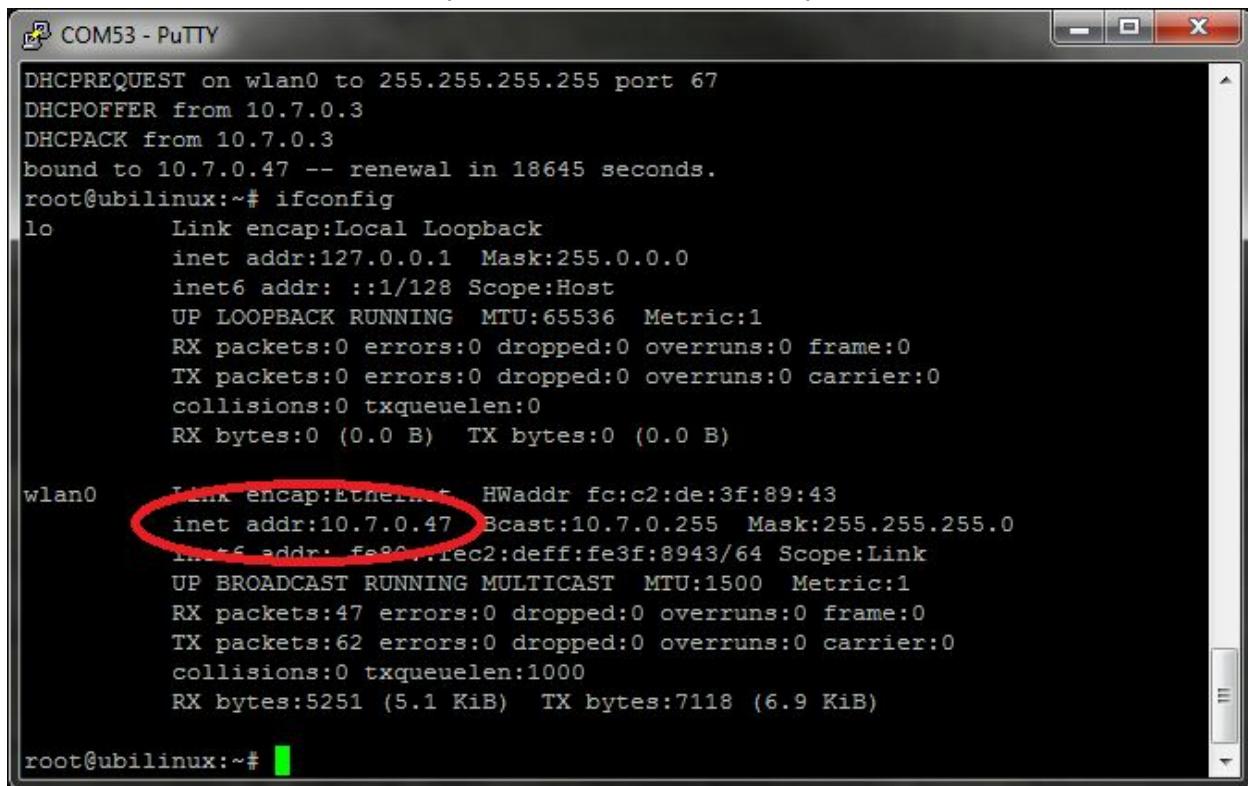
Restart the wireless driver by entering:

```
# ifdown wlan0 && ifup wlan0
```

Once that configures, your Edison should be connected to your local WiFi. You can check by entering:

ifconfig

You should see an “inet addr” entry under “wlan0” that shows your IP Address.



```
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPoffer from 10.7.0.3
DHCPACK from 10.7.0.3
bound to 10.7.0.47 -- renewal in 18645 seconds.
root@ubilinux:~# ifconfig
lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
              UP LOOPBACK RUNNING MTU:65536 Metric:1
              RX packets:0 errors:0 dropped:0 overruns:0 frame:0
              TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:0
              RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0    Link encap:Ethernet HWaddr fc:c2:de:3f:89:43
       inet addr:10.7.0.47  Bcast:10.7.0.255  Mask:255.255.255.0
        inet6 addr: fe80::fc:c2ff:fe3f:8943/64 Scope:Link
              UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
              RX packets:47 errors:0 dropped:0 overruns:0 frame:0
              TX packets:62 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:5251 (5.1 KiB)  TX bytes:7118 (6.9 KiB)
root@ubilinux:~#
```

To ensure you have access to the Internet, you can ping a known site, such as www.google.com by the command : ***ping www.google.com***

Set the time manually

date --set '1998-11-02 21:08:00'

Run the below command,

apt-get update

```
# apt-get upgrade
```

Now let's do something a little complex (:P), **write a python program** that will continuously blink the LED connected to pin #13,

Requirements: Installed Ubilinux on Edison. (As already shown in this Document)

1. Login as root. Type >> **nano blinkyLED.py** and hit enter.
2. You will find a empty file. Paste the python code below into file

```
#!/usr/bin/env python

import mraa
import time

led = mraa.Gpio(13)
led.dir(mraa.DIR_OUT)

while True:
    led.write(1)
    time.sleep(0.2)
    led.write(0)
    time.sleep(0.2)
```

3. Press **CTRL+X** and hit **Y** and **ENTER**
4. Type>> **python blinkyLED.py** and hit **ENTER**

**Similarly, Reading an values of the POT connected at A0:
Copy the below code to a new read_val.py file and do as above.**

```
#!/usr/bin/env python

import mraa
import sys
import time

pot = mraa.Aio(0)
while 1:
    potVal = float(pot.read())
    print potVal
    time.sleep(0.5)
```

Let us post temperature data to dweet.io

To install the *requests* package Enter >>

```
apt-get install python-pip
pip install requests
```

Connect the Grove's Temperature Sensor to A0 pin.

Copy the below code to a new <somename>.py file and

```
# This example send the temperature from INTEL EDISON
# to http://dweet.io/, a IoT on Cloud.
# with name = <any name> for every 2 seconds
#
# To check this dweet, visit here on browser
# http://dweet.io/follow/<any\_name>

import requests
import time
import mraa
import math

dweetIO = "https://dweet.io/dweet/for/"
myName = "<any name>"
myKey = "data"
pinA0=mraa.Aio(0)
B = 3975

while True:

    val = float(pinA0.read())

    # scale the temp value as needed
    resistance =(1023-val)*10000/val
    temperature = 1/(math.log(resistance/10000) /B+1/298.15)-273.15;
    print(val)
    # send to Cloud, dweet.io

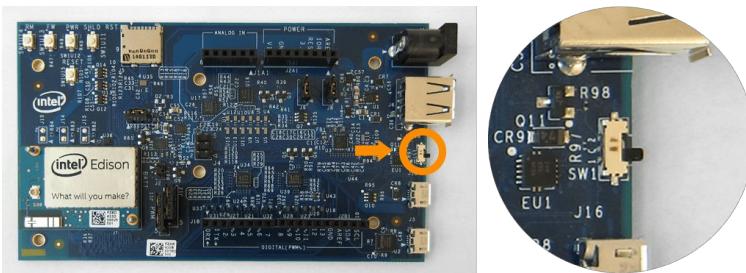
    urlString = dweetIO+myName+'?' +myKey+'=' +str(temperature)
    print(urlString)
    response = urllib2.urlopen(urlString)
    headers = response.info()
    print response.read()
    print response.code
    print "Temperature value has been updated: " + str(temperature)
    time.sleep(3)
```

Misc:

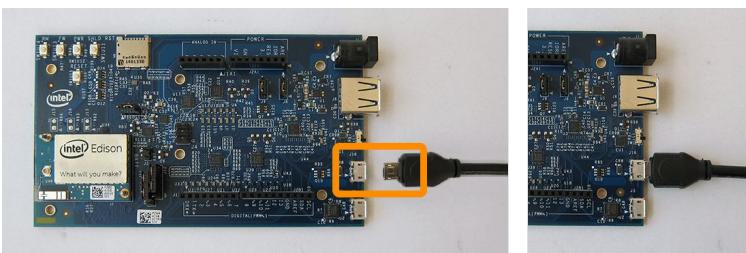
Powering your board over USB

While an external power supply is the preferred way of powering the Intel Edison board, you can also power your board over USB, as described in this section. Keep in mind the following:

- Certain USB slots on your system might not be able to provide enough power to the board. This will ultimately result in some very unpredictable behavior from the board, especially when using Wi-Fi* or driving motors.
 - If you are connecting your board to a laptop, be sure to plug your laptop in to help ensure that your board has enough power.
1. Find the microswitch in between the USB ports on the expansion board. Switch the microswitch down towards the micro-USB ports, if it isn't already.



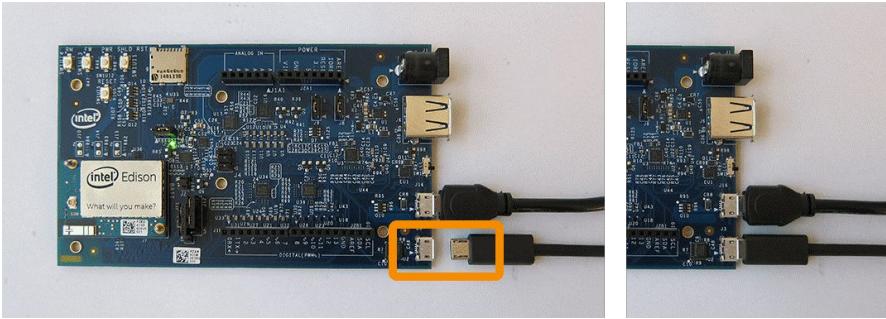
2. Plug in one of the micro-USB cables to the **middle** USB connector on the expansion board.



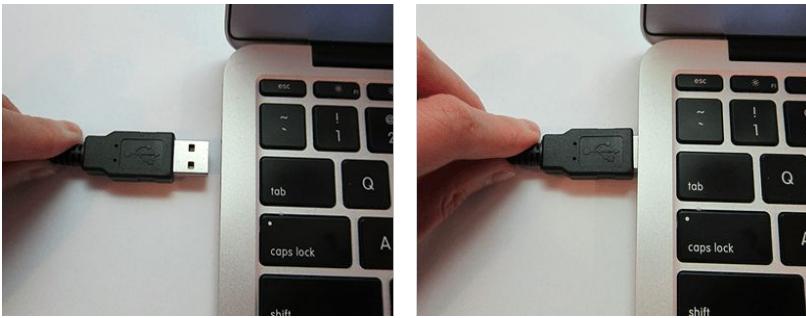
3. Plug in the other end of the USB cable to your computer.



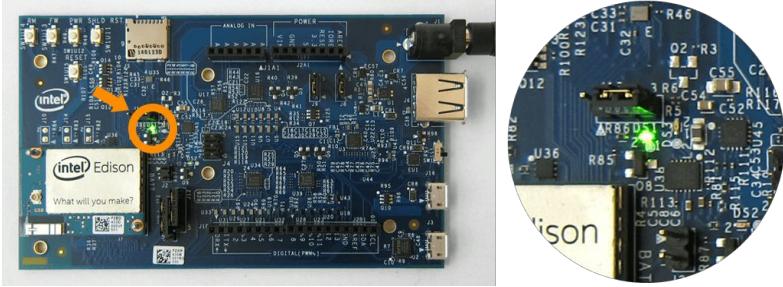
4. Plug in your second USB cable to the **edge** USB connector on the board.



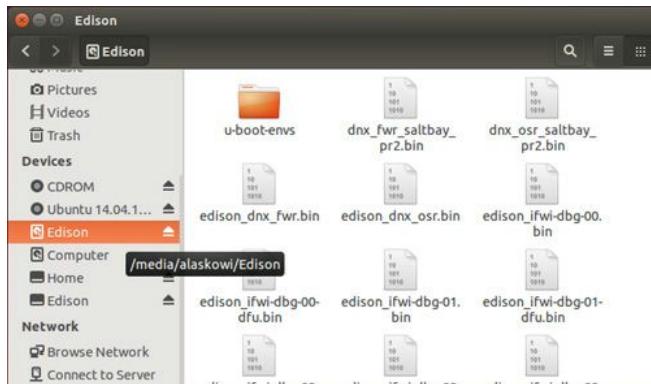
5. Plug the other end of the USB cable in to your computer.



6. A green light should light up on the expansion board. If it doesn't, check your connection, plug in your laptop power adapter, try a different USB port on your computer, or try using a USB hub that has a power supply.



- Once you have connected your board, you should see a drive called **Edison** show up in Files..



Remove old images

- Open up a new Terminal window.
- Enter the command: `cd /media/username`
- Type `ls` and press **Enter**. Verify that the Edison directory is in the list.*
- To remove all visible files and folder, enter the command: `rm -rf Edison/*`
- To remove all hidden files and folders, enter the command: `rm -rf Edison/\.*`
- To view files on the mounted Edison drive, enter the command: `ls -lag Edison`.
- Verify that the files have been removed.

*You will have to use `/media/username/Edison`, not `/media/psf/Edison`.

Installing Flash Tool Lite on Linux*

Installation

The Linux installation instructions are only for 64-bit operating systems; the versions supported in Ubuntu are therefore Ubuntu 12.04 LTS 64-bit and greater.

3. Install the dependent packages for the tool.:

- Ubuntu 12.04LTS:

```
sudo apt-get install gdebi ia32-libs --yes
```

- Ubuntu 13.04 64-bit and later:

```
sudo apt-get install gdebi libncurses5:i386  
libstdc++6:i386 --yes
```

Create a folder 'workshop_edison' in the 'home' directory.

4. Download the Flash Tool Lite installation package for Linux from the [Edison Downloads page](#), Place the phoneflashtoollite_5.2.4.0_linux_x86_64.deb file in the 'workshop_edison' folder.

Double click on the .deb file.

The **Ubuntu Software Center** will prompt you to install. Click install.

Wait for it to Install. Click on '**FORWARD**' when prompted.

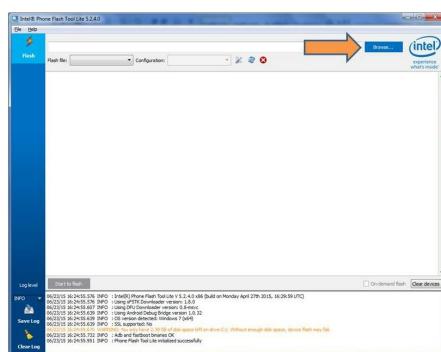
Flash Tool is installed.

5. Click on "**Release 2.1 Yocto* complete image**" from **Intel Edison® Board "Firmware Software Release 2.1"** section in the [link](#) here. It Downloads a zip file called '**edison-image-ww25.5-15**' and place it in the 'workshop_edison' folder.
Right Click on '**edison-image-ww25.5-15**' and click on '**Extract here**'

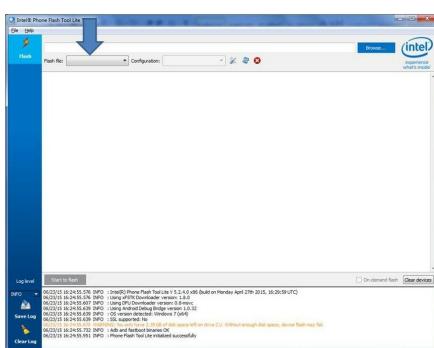
Flashing the Edison:

Remove the Power supply and the USB cables from the Intel Edison Board.

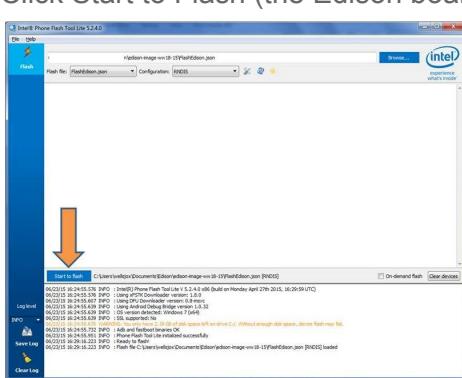
1. Launch “Flash Tool Lite” icon created on the Desktop.
2. Click on ‘Browse’. Browse to ‘/home/<system name>/workshop_edison/edison-image-ww25.5-15/. Select FlashEdison.json , Click on OPEN.



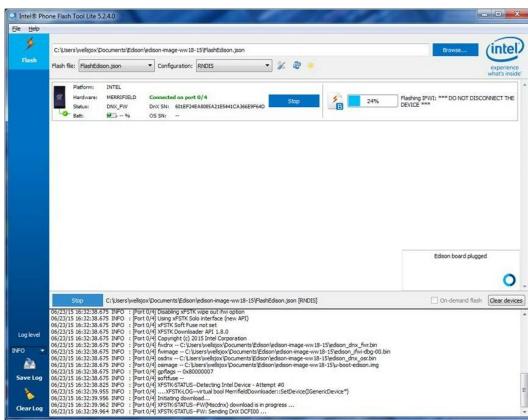
3. In the bottom display, after the tool has prepared the .json file it should report “ Ready to flash.”



4. On the Configuration drop down, choose CDC if your host machine is OS X or Linux, choose RNDIS for Windows.
5. Click Start to Flash (the Edison board is not yet plugged in)



6. Plug the USB cable into the Multigadget port of the Edison board. Additionally plug the external power supply in after the USB cable. You should see the Flash Tool detect the board and begin the flash process.

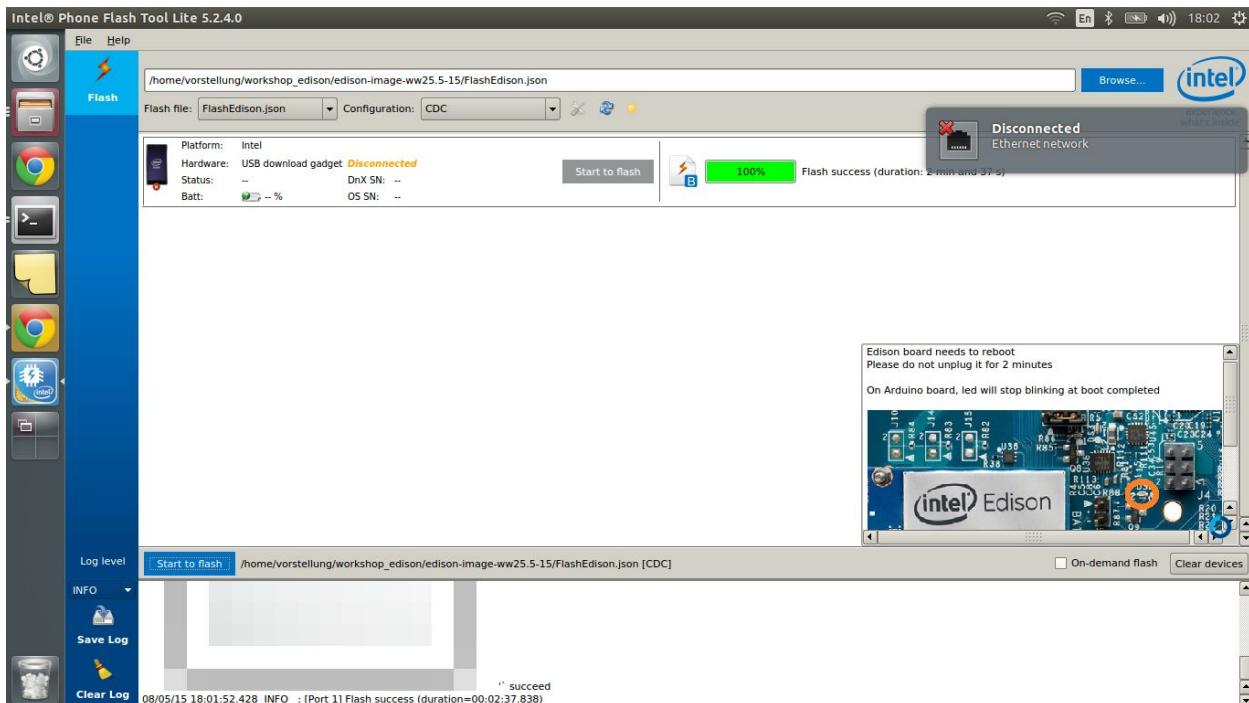


Example of firmware flash progress:



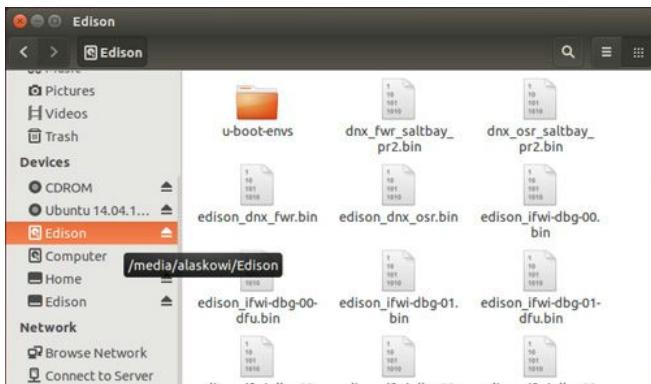
ERROR : [Port 1] Command `"/usr/lib/phoneflashtoollite/xfstkFlashTool"
"/home/vorstellung/workshop_edison/edison-image-ww25.5-15/edison_dnx_fwr.bin"
"/home/vorstellung/workshop_edison/edison-image-ww25.5-15/edison_ifwi-dbg-00.bin"
"/home/vorstellung/workshop_edison/edison-image-ww25.5-15/edison_dnx_osr.bin"
"/home/vorstellung/workshop_edison/edison-image-ww25.5-15/u-boot-edison.img"
"0x80000007" "none" "none" failed
08/05/15 17:50:24.546 ERROR : [Port 1] Flash failed (Command type: Ifwi)

ERROR : [Port 1] Command `"/usr/lib/phoneflashtoollite/dfu-util" "--alt" "u-boot-env1" "-D"
"/home/vorstellung/workshop_edison/edison-image-ww25.5-15/u-boot-envs/edison-blankcdc.bin
" "-t" "4096" "-R" "-d" "8087:0a99" failed
08/05/15 17:52:23.062 ERROR : [Port 1] Flash failed (Command type: DFU)



Wait till the LED goes OFF. Once process is complete, you will find the screen as above.

- Once you have connected your board, you should see a drive called **Edison** show up in Files..



Remove old images

8. Open up a new Terminal window.
9. Enter the command: `cd /media/username`
10. Type `ls` and press **Enter**. Verify that the Edison directory is in the list.*
11. To remove all visible files and folder, enter the command: `rm -rf Edison/*`
12. To remove all hidden files and folders, enter the command: `rm -rf Edison/\.*`
13. To view files on the mounted Edison drive, enter the command: `ls -lag Edison`.
14. Verify that the files have been removed.

*You will have to use `/media/username/Edison`, not `/media/psf/Edison`.

Installation

The Linux installation instructions are only for 64-bit operating systems; the versions supported in Ubuntu are therefore Ubuntu 12.04 LTS 64-bit and greater.

6. Install the dependent packages for the tool.:

- Ubuntu 12.04LTS:

```
sudo apt-get install gdebi ia32-libs --yes
```

- Ubuntu 13.04 64-bit and later:

```
sudo apt-get install gdebi libncurses5:i386  
libstdc++6:i386 --yes
```

Create a folder 'workshop_edison' in the 'home' directory.

7. Download the Flash Tool Lite installation package for Linux from the [Edison Downloads page](#), Place the phoneflashtoollite_5.2.4.0_linux_x86_64.deb file in the 'workshop_edison' folder.

Double click on the .deb file.

The **Ubuntu Software Center** will prompt you to install. Click install.

Wait for it to Install. Click on '**FORWARD**' when prompted.

Flash Tool is installed.

8. Click on "**Release 2.1 Yocto* complete image**" from **Intel Edison® Board "Firmware Software Release 2.1"** section in the [link](#) here. It Downloads a zip file called '**edison-image-ww25.5-15**' and place it in the 'workshop_edison' folder.
Right Click on '**edison-image-ww25.5-15**' and click on '**Extract here**'

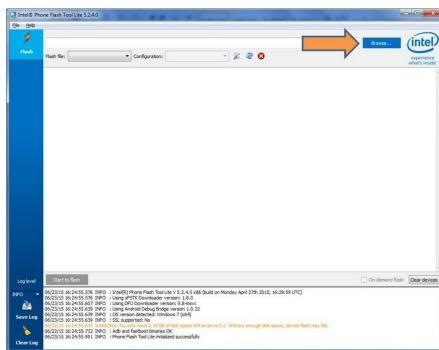
Flashing the Edison:

Remove the Power supply and the USB cables from the Intel Edison Board.

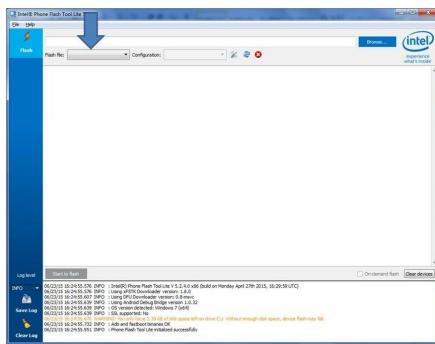
7. Launch "Flash Tool Lite" icon created on the Desktop.

8. Click on '**Browse**'. Browse to '/home/<system name>/workshop_edison/edison-image-ww25.5-15/'. Select **FlashEdison.json**,

Click on **OPEN**.

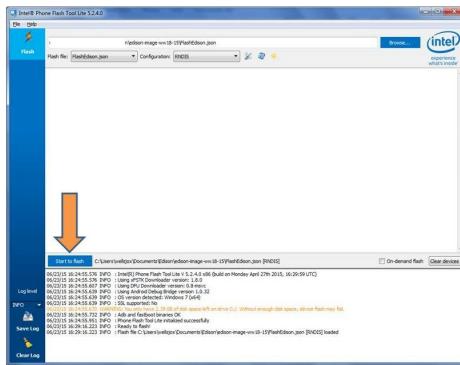


9. In the bottom display, after the tool has prepared the .json file it should report “ **Ready to flash.**”

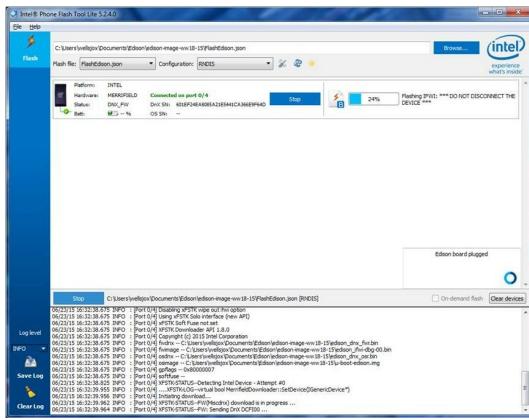


10. On the Configuration drop down, choose CDC if your host machine is OS X or Linux, choose RNDIS for Windows.

11. Click Start to Flash (the Edison board is not yet plugged in)



12. Plug the USB cable into the Multigadget port of the Edison board. Additionally plug the external power supply in after the USB cable. You should see the Flash Tool detect the board and begin the flash process.



Example of firmware flash progress:



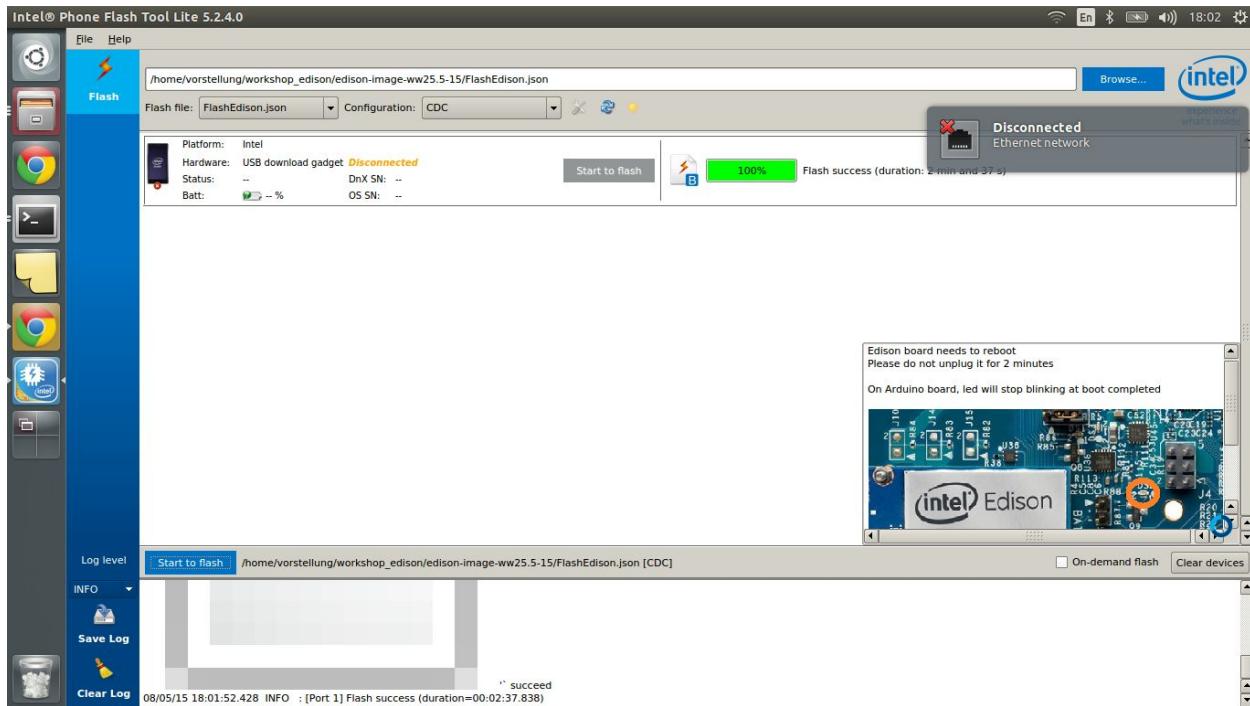
```

ERROR : [Port 1] Command `"/usr/lib/phoneflashtoollite/xfstkFlashTool"
"/home/vorstellung/workshop_edison/edison-image-ww25.5-15/edison_dnx_fwr.bin"
"/home/vorstellung/workshop_edison/edison-image-ww25.5-15/edison_ifwi-dbg-00.bin"
"/home/vorstellung/workshop_edison/edison-image-ww25.5-15/edison_dnx_osr.bin"
"/home/vorstellung/workshop_edison/edison-image-ww25.5-15/u-boot-edison.img"
"0x80000007" "none" "none" failed
08/05/15 17:50:24.546 ERROR : [Port 1] Flash failed (Command type: Ifwi)
  
```

```

ERROR : [Port 1] Command `"/usr/lib/phoneflashtoollite/dfu-util" "--alt" "u-boot-env1" "-D"
"/home/vorstellung/workshop_edison/edison-image-ww25.5-15/u-boot-envs/edison-blankcdc.bin
" "-t" "4096" "-R" "-d" "8087:0a99" failed
  
```

08/05/15 17:52:23.062 ERROR : [Port 1] Flash failed (Command type: DFU)



Wait till the LED goes OFF. Once process is complete, you will find the screen as above.

Setting up SSH

Setup a password for the Edison Board:

```
Configure Edison: Device Password

Enter a new password (leave empty to abort)
This will be used to connect to the access point and login to the device.
Password: *****
Please enter the password again: *****
The device password has been changed.

First-time root password setup complete. Enabling SSH on WiFi interface.
root@edisonhub:~#
```

If you plan on having it easy ! and have your Edison plugged in one corner of your house and not have to connect the USB cable to the Serial Port every time, using SSH to get into your Intel Edison is a great idea.

To use SSH with the latest version of the Yocto image(edison-image-rel1-maint-rel1-ww42-14.zip) you need to setup a password.

To create a password, at the command line type in **`configure_edison --password`**
At the prompt type in the new password and type it again to confirm as shown in the image above.

Type the command >> **`reboot`**

(Remember the IP address in the previous step the edison had from the router)

Launch a new Terminal window on your System.

And run the following command >> **`ssh root@192.168.1.9`**

Enter your password that you had set up a couple of steps ago and this is what you see "root@edisonhub:~# " prompt ,your SSH setup is complete . And yeah! you can use your Edison from anywhere in the house.

Light map:

In ubuntu:

```
scp -r cxcfiles root@192.168.0.104:/root  
scp lightinstall.sh root@192.168.0.104:/root
```

In edison:

```
./lightinstall.sh
```

.

Control Light bulb through dweet

- Login as root. Type >> **nano controllight.py** and hit enter.
- You will find a empty file. Paste the python code below into file

```
#!usr/bin/env python

import time
import urllib2
import os
import mraa
import json

controlio = 6

def _WriteFile(fullFilename, value):
    """
    Created this to reduce typing
    """
    myFile = open(fullFilename, "w")
    myFile.write(value)
    myFile.close()

def _ReadFile(fullFilename):
    """
    Created this to reduce typing
    """
    myFile = open(fullFilename, "r")
    readValue = myFile.read().strip()
    myFile.close()
    return readValue

def loop():
    while(1):

        response = urllib2.urlopen(geturl)
        data=response.read()
        #print data
```

```

j=json.loads(data)
#print j
value=j['with'][0]['content']['lightstatus']
#print str(value)

#if(pinValue != prevVal):
if(value == 0):
    if(int(pin.read()) == 1):
        pin.write(0)
        print "Light OFF\n"

else:
    if(int(pin.read()) == 0):
        pin.write(1)
        print "Light ON\n"

time.sleep(3)

if __name__ == "__main__":
    print "====="
    print "      Intel Edison IOT      "
    print "====="

#url = 'https://dweet.io:443/dweet/for/mytryiiscedison?pot='
#geturl='https://dweet.io/get/latest/dweet/for/testck'

try:

    pin = mraa.Gpio(controlio)
    pin.dir(mraa.DIR_OUT)
    pin.write(1)
    loop()

except KeyboardInterrupt:
    print "Keyboard exception"
    print "End ..."

except:
    print "End ..."

pin.write(1)

```

- Press ***CTRL+X*** and hit ***Y*** and ***ENTER***
- Type>> ***python controllight.py*** and hit ***ENTER***

