**` `a.**

**Module 2**

**Intel Edison Tutorial**

**About Intel Edison**

* [Homepage](https://software.intel.com/en-us/iot/hardware/edison)
  + Explore other hardware in addition to Intel Edison
* [Specifications](http://www.intel.com/content/dam/support/us/en/documents/edison/sb/edison_pb_331179002.pdf)
* [Pinouts](http://www.intel.com/content/www/us/en/support/boards-and-kits/000006090.html)
* [Intel Edison Compute Module - Hardware Guide](http://download.intel.com/support/edison/sb/edisonmodule_hg_331189004.pdf)

**Setting Up Intel Edison** (11/6/2017)

# [https://software.intel.com/en-](https://software.intel.com/en-us/get-started-edison-windows)

# [us/get-started-edison-windows](https://software.intel.com/en-us/get-started-edison-windows)1

# You can choose between windows, linux, or MacOS

# Step 3: Choose IDE

* + - You can login directly with serial terminal and program Intel Edison through command line: [Setting up a serial terminal on a system with Windows](https://software.intel.com/en-us/setting-up-serial-terminal-on-system-with-windows)
* Download and install useful programs <https://software.intel.com/en-us/setting-up-serial-terminal-on-system-with-windows>
  + Login to your Edison using the serial terminal interface
  + Add the following repositories in the conf file by entering the following command on your Edison.

vi /etc/opkg/base-feeds.conf

or

nano /etc/opkg/base-feeds.conf

Inside the file,

lines. (Quick tutorial of the [vi](http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Editors/ViIntro.html) file editor. A more exhaustive list [here](http://www.tutorialspoint.com/unix/unix-vi-editor.htm).) Press "i" to begin editing.

src/gz all <http://repo.opkg.net/edison/repo/all>  
src x86 <http://iotdk.intel.com/repos/1.1/iotdk/x86>  
src i586 <http://iotdk.intel.com/repos/1.1/iotdk/i586>  
src/gz edison <http://repo.opkg.net/edison/repo/edison>  
src/gz core2-32 <http://repo.opkg.net/edison/repo/core2-32>  
  
Save and quit the file by pressing the "Esc" button followed by ":wq".  
Back on the terminal, execute the following command:  
  
opkg update

* + cdGit

opkg install git

* + Alternate text editors

opkg install

nano

* + PIP

opkg install python-pip

Pip installs when you update the opkg repo above. However for pip to install the python packages, setuptools need to be installed.

Install this as follows:

wget https://bitbucket.org/pypa/setuptools/downloads/ez\_setup.py --no-check-certificate

python ez\_setup.py

The above process allows you to install pip's python packages.

* + Mraa

Mraa is the Low Level Skeleton Library for IO Communication on GNU/Linux platforms. Refer this [link](https://github.com/intel-iot-devkit/mraa/tree/master/examples/python) for a number of examples.

opkg install mraa

* + **Gattool (as part of BlueZ)**

This is a tool used to access the “services” running on your bluetooth device. We shall use it for accessing the Bluetooth Low Energy services.

The installation instructions are available [here](https://software.intel.com/en-us/articles/using-the-generic-attribute-profile-gatt-in-bluetooth-low-energy-with-your-intel-edison).

root@edison: cd ~  
root@edison: wget https://www.kernel.org/pub/linux/bluetooth/bluez-5.24.tar.xz –-no-check-certificate  
root@edison: tar -xf bluez-5.24.tar.xz  
root@edison: cd bluez-5.24  
root@edison: ./configure --disable-systemd –disable-udev  
root@edison:make

root@edison: make install

* + Boto3

Boto3 is the Amazon Web Services SDK for python developers. For a detailed look at Boto3 & its usage, please refer to its [documentation](https://boto3.readthedocs.io/en/latest/).

* + Bloop (Mac Only)

If you are using a Mac (OS X) to access the board and do your work, you may also find the following tool called Bloop useful. This is entirely optional, but might be convenient if you are interested in streamlining your work on the Intel Edison.

Bloop is a CLI tool for helping developers work on Intel Edison. ([Bleep](https://github.com/rwaldron/johnny-five/issues/495) is the partner library which you can use on the Edison board itself). Bloop is specifically meant to help automate basic tasks revolving around connecting to Edison, getting a terminal, copying files to it and SSH-ing into it. This is part of the [Johnny-five](https://github.com/rwaldron/johnny-five/) IoT & robotics programming framework.

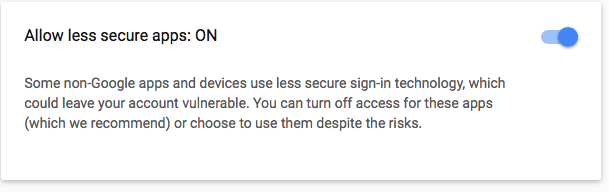
**WiFi and Intel Edison**

* opkg install mraa
  + Call it when in intel edison command line interface
  + configure\_edison --wifi
  + Pingls
  + See more detailed instructions [hiere](https://software.intel.com/en-us/connecting-your-intel-edison-board-using-wifi).
* ping [www.google.com](http://www.google.com)
  + Check that you are getting a response back.
* **Headless Intel Edison Setup**
  + Challenge for class: These [instructions](http://elinux.org/RPi_Email_IP_On_Boot_Debian) are for creating a headless Raspberry Pi. Can each group port this code to create a headless Intel Edison?
  + root@michael\_edison: ip route list
    - default via 192.168.5.1 dev wlan0
    - 192.168.2.0/24 dev usb0 src 192.168.2.15
    - 192.168.5.0/24 dev wlan0 src 192.168.5.44
  + nano mail\_ip\_address.py
    - Copy the python script from [instructions](http://elinux.org/RPi_Email_IP_On_Boot_Debian) to mail\_ip\_address.py
    - Modify the email address and password fields
    - Modify the “send to” email address
    - Modify the subject line to change Raspberry Pi to Intel Edison
  + python mail\_ip\_address.py
    - Do you get an error? What error do you get?

msg['Subject'] = 'IPs For IE on %s' % today.strftime('%b %d %Y)

I am getting this errror.

* + - Go to https://myaccount.google.com/security
      * Go to “Connected Apps & Sites”



* + - Rerun python mail\_ip\_address.py
  + Configure mail\_ip\_address.py to be an executable  
    - chmod +x mail\_ip\_address.py
  + cd /etc
  + cd init.d/
  + ls -l
    - What file is in this folder?

root@group3\_papel:~# chmod +x mail.py

root@group3\_papel:~# cd /etc

root@group3\_papel:/etc# cd init.d/

root@group3\_papel:/etc/init.d# ls -l

total 4

-rwxr-xr-x 1 root root 1893 Jun 6 21:39 fuse

* + nano startup.sh

#!/bin/sh

sleep 30

python /home/root/mail\_ip\_address.py # use the right file path for you

exit 0

* + Create an executable of shell script
    - chmod u+x startup.sh
  + ./startup.sh
    - Do you get an email?
    - Note the 30 sec delay.
  + **update-rc.d startup.sh defaults**
  + ls -l /etc/rc?.d/\*startup\*
    - What is the output?

lrwxrwxrwx 1 root root 20 Jun 6 22:18 /etc/rc0.d/K20startup.sh -> ../init.d/sta rtup.sh

lrwxrwxrwx 1 root root 20 Jun 6 22:18 /etc/rc1.d/K20startup.sh -> ../init.d/sta rtup.sh

lrwxrwxrwx 1 root root 20 Jun 6 22:18 /etc/rc2.d/S20startup.sh -> ../init.d/sta rtup.sh

lrwxrwxrwx 1 root root 20 Jun 6 22:18 /etc/rc3.d/S20startup.sh -> ../init.d/sta rtup.sh

lrwxrwxrwx 1 root root 20 Jun 6 22:18 /etc/rc4.d/S20startup.sh -> ../init.d/sta rtup.sh

lrwxrwxrwx 1 root root 20 Jun 6 22:18 /etc/rc5.d/S20startup.sh -> ../init.d/sta rtup.sh

lrwxrwxrwx 1 root root 20 Jun 6 22:18 /etc/rc6.d/K20startup.sh -> ../init.d/sta rtup.sh

* Finally, unplug your Intel Edison, and plug it back in. Wait 30 sec, and you should get an email of your IP address.

**Yes, got an email !!**