Intel Galileo Setup

Download Arduino IDE for Intel Galileo

- https://downloadcenter.intel.com/download/24782

IDE is available for Windows, Linux and Mac OS. Download Arduino IDE according to the operating system.

Driver Installation (only for windows)

- https://software.intel.com/en-us/articles/getting-started-with-the-intel-galileo-board-on-windows#terminal

Install Arduino IDE

- https://software.intel.com/en-us/articles/install-arduino-ide-on-intel-iot-platforms

Run a sample arduino sketch on Galileo

- https://software.intel.com/en-us/articles/intel-iot-platforms-blink-led-arduino-ide

Run a full linux distribution on Galileo

By default, galileo runs little linux os (poky distribution). The small linux image only allows execution of Arduino sketches. But with an SD card, we can boot the Galileo off a bigger Linux image, which provides access to **WiFi, Python, Node JS, SSH, openCV, ALSA, java and other things.**

Download debian image from the following link

- http://sourceforge.net/projects/galileodebian/

Install the image onto a SD card

For Windows:

Download a tool called Rawrite32 and use it to write the image to your SD card.

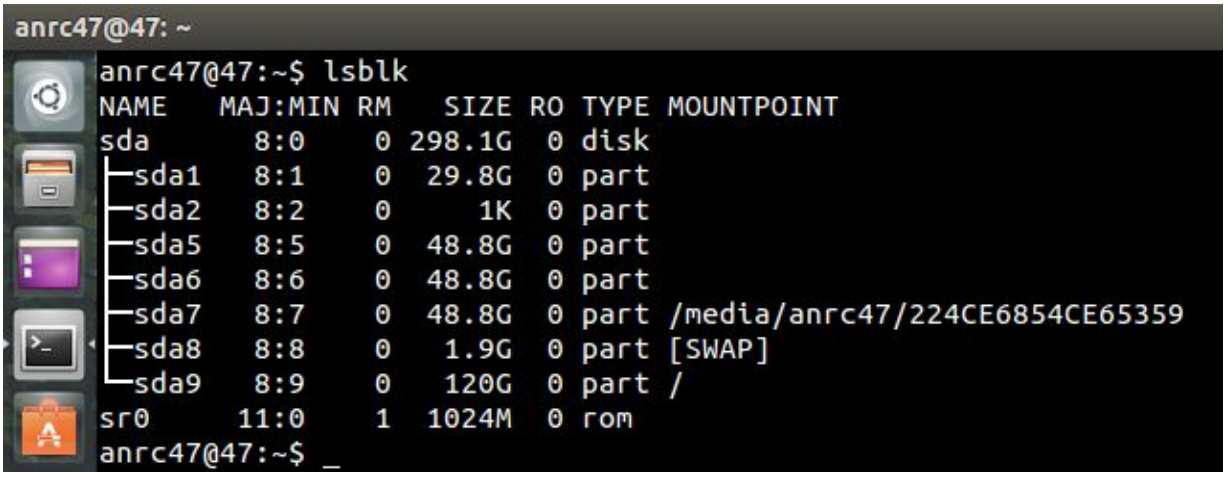
- ➢ http://www.netbsd.org/~martin/rawrite32/download.html

For Ubuntu Linux:

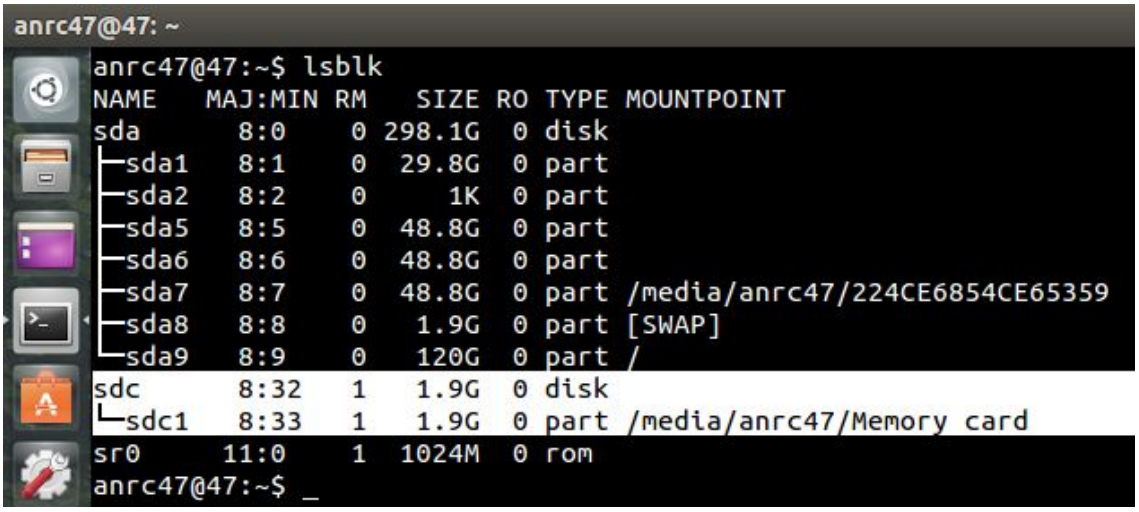
On a host machine, use **dd** command to write the downloaded image to whichever device is your SD card.

- ➢ Open a Terminal program (Ctrl + Alt + t in Ubuntu 14.04 LTS) and run the following command
  $ lsblk

Following image shows the sample output.



- ➢ Now attach SD card to your host machine using SD card reader and type **lsblk** command.



According to above sample picture, SD card is connected as **sdc** and it has only one partition, **sdc1**.

- Before burning the image to an SD card, unmount all the partitions. A/c to above picture, we have only one partition. To unmount, run the following command
  $ sudo umount /dev/sdc1

To perform this action, you need to have a root user permission so add **sudo** at the beginning.

```
anrc47@47:~$ sudo umount /dev/sdc1
[sudo] password for anrc47:
anrc47@47:~$ _
```

- Otherwise, you can use Disks program in Ubuntu (14.04 LTS) to note down the device name of an SD card and to unmount all the partitions as well.
- Extract the downloaded image (Right click on downloaded image and select **Extract Here** option or use tar command).
- After the unmount, run the following command
  $ sudo dd bs=4M if=<path to the downloaded image> of=/dev/<device name>

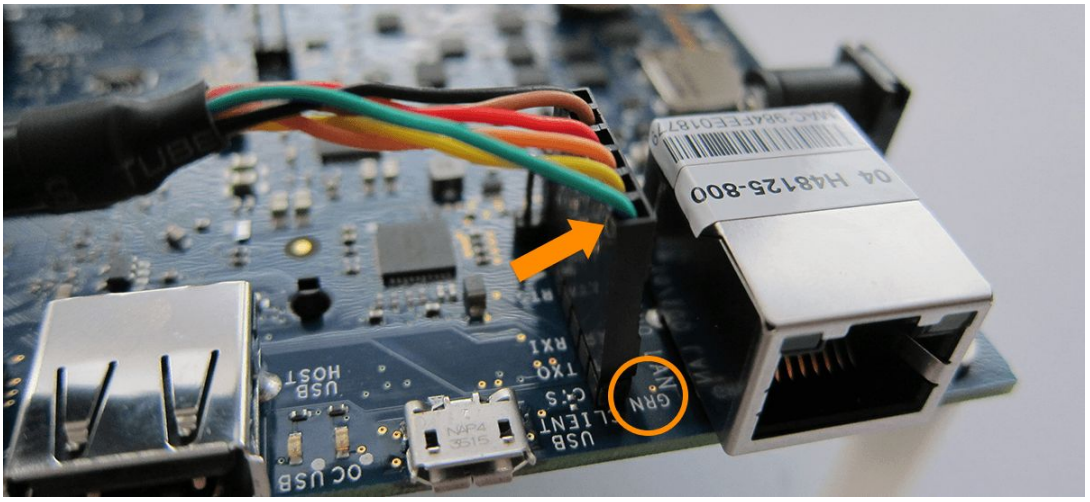
```
anrc47@47: ~
anrc47@47:~$ sudo dd bs=4M if=/home/anrc47/galileo-1.2.img of=/dev/sdc
[sudo] password for anrc47: _
```

It might take some time. After completion, eject the SD card (Don't just unplug it).

- Now insert the micro SD card to the SD card slot on galileo and power up (For this, you don't need micro usb cable. Micro usb cable is only required while uploading arduino sketch)
- Now you can boot your galileo with bigger linux os (with debian distro)

Setup Serial Terminal on the Intel® Galileo

First, boot the galileo with SD card. Now, connect your USB to FTDI cable to galileo board as shown below.



For Windows*:

- https://software.intel.com/en-us/articles/getting-started-with-the-intel-galileo-board-on-windows#terminal

For Ubuntu Linux:

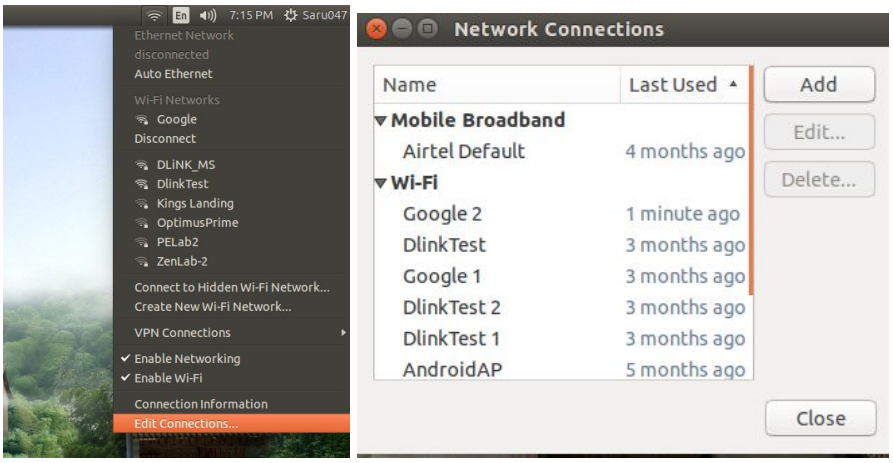
- In terminal, run the following command
  $ sudo screen /dev/ttyUSB0 115200

A window will open. Enter the username as **root** and the password is **root**.

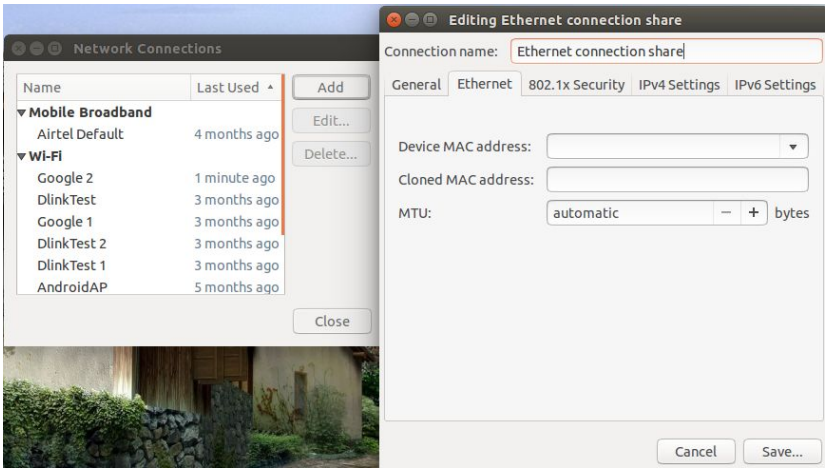
Now, you will be inside a Galileo. Do whatever you want.....!!!

Login to the Galileo using SSH (only for Ubuntu linux users):

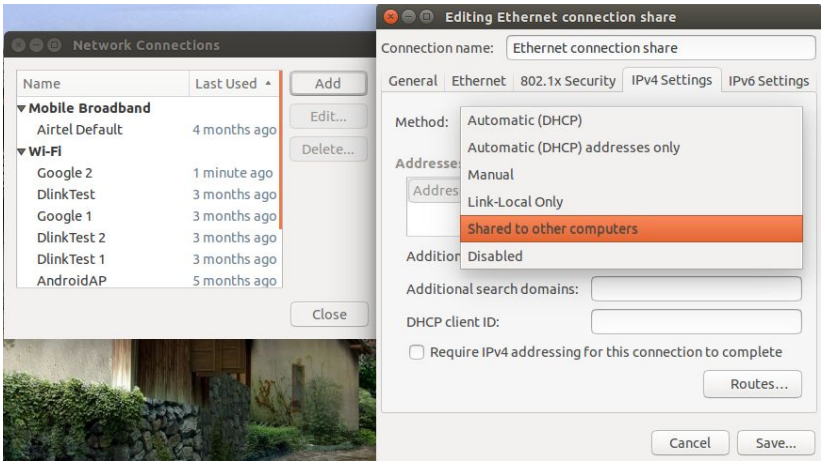
- Connect your host machine to an available wifi network
- Connect an Ethernet cable between your host machine and the galileo
- Now, share the internet from your host machine to the galileo (bridge connection). Your laptop/system acts like a server and assigns an IP address to the galileo. Follow the below pictures
- Open your Network Manager in ubuntu and select **Edit Connections** options and it opens a new window and click on **Add** button



- Create an ethernet interface



- In the interface window, click on **IPv4 Settings** tab and select **Shared to other computers** option in the dropdown menu. After click on **save** and exit



- Restart the networking services.
- Next, run the following command

  $ ifconfig

```
anrc47@47: ~
anrc47@47:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 40:61:86:ea:9e:22
          inet addr:10.42.0.1  Bcast:10.42.0.255  Mask:255.255.255.0
          inet6 addr: fe80::4261:86ff:feea:9e22/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:104933 errors:0 dropped:0 overruns:0 frame:0
          TX packets:110883 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:57570723 (57.5 MB)  TX bytes:16095004 (16.0 MB)

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:18267 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18267 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:18283498 (18.2 MB)  TX bytes:18283498 (18.2 MB)

wlan4     Link encap:Ethernet  HWaddr e8:4e:06:0d:7a:45
          inet addr:192.168.0.114  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::ea4e:6ff:fe0d:7a45/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:370 errors:0 dropped:0 overruns:0 frame:0
          TX packets:408 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:76651 (76.6 KB)  TX bytes:89871 (89.8 KB)
```

- In ubuntu machine, ethernet address should be **10.42.0.1** like the above sample picture
- Now install **nmap** or **arp-scan** package in your host machine. To do that run the following command

  $ sudo apt-get install nmap arp-scan
- Run the below command to get the ip address of the galileo in ubuntu terminal after the installation of nmap and arp-scan package

  $ nmap 10.42.0/24

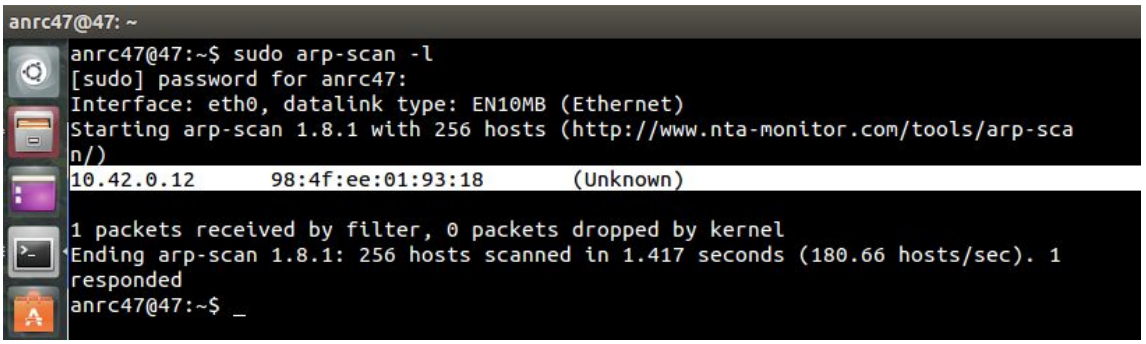
```
anrc47@47: ~
anrc47@47:~$ nmap 10.42.0/24

Starting Nmap 6.40 ( http://nmap.org ) at 2015-04-25 20:10 IST
Nmap scan report for 10.42.0.1
Host is up (0.00013s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds

Nmap scan report for 10.42.0.12
Host is up (0.026s latency).
All 1000 scanned ports on 10.42.0.12 are closed

Nmap done: 256 IP addresses (2 hosts up) scanned in 2.92 seconds
anrc47@47:~$ _
```

It should show **2 hosts up** like the above picture. If it fails, disconnect ethernet cable from galileo and reconnect it or you might have made a mistake while preparing the SD card.

- A/c to the example, **10.42.0.12** is the ip address of the galileo

- Now login to the galileo using the below command from your machine
  ssh <user name>@<ip address>
  For example:, **$ ssh root@10.42.0.12**

Or you can use **arp-scan** command to get the ip address of the galileo. So in ubuntu terminal run the following command
$ sudo arp-scan -l



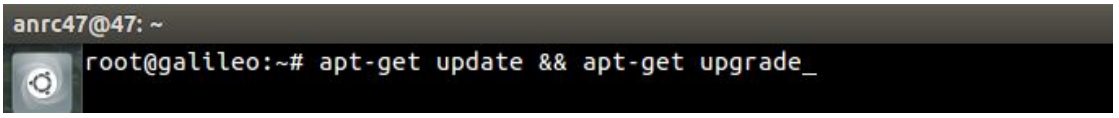
- A/c to the above picture, **10.42.0.12** is the ip address of the galileo
- Now login to the galileo using the below command from your machine
  ssh <user name>@<ip address>
  For example:, **$ ssh root@10.42.0.12**

Initial setup after Galileo is booted with debian os

In Galileo:

After galileo is booted with full linux distribution, provide internet for galileo.

- In terminal, run the following command (**this is mandatory**)
  # apt-get update && apt-get upgrade
  (don't need to add sudo here.)



This might take some time, depends on your internet connectivity.

- **apt-get update** updates the list of available packages and their versions, but it does not install or upgrade any packages
- **apt-get upgrade** actually installs newer versions of the packages you have

Install few necessary packages so run the following command.
 # apt-get install wpasupplicant wireless-tools locales python-smbus

```
anrc47@47: ~
root@galileo:~# apt-get install wpasupplicant wireless-tools locales python-smbus_
```

After installation of above packages, you need to configure few things.

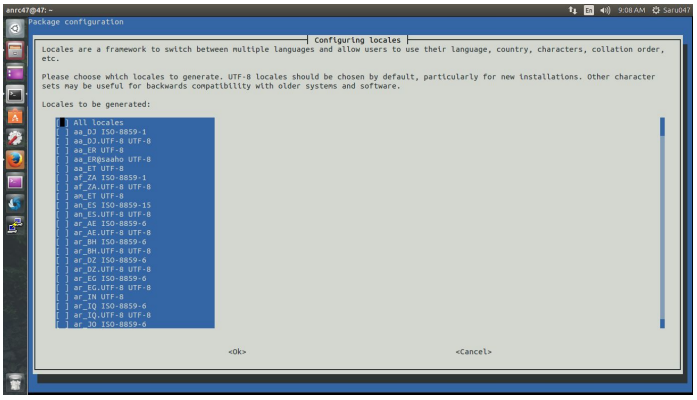
Locale settings for debian (in galileo) - optional

- Re-configure locale settings. In terminal, run the following command

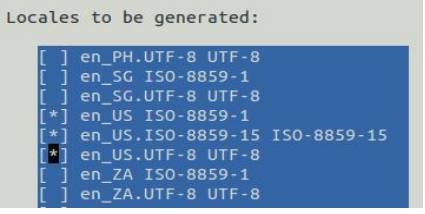
  # dpkg-reconfigure locales

```
anrc47@47: ~
root@galileo:~# dpkg-reconfigure locales_
```

A new window will open as shown below.



- Select the following three in the menu using space bar and press enter.
  - en_US.ISO-8859-1
  - en_US.ISO-8859-15
  - en_US.UTF-8



- In the next window select en_US.UTF-8 as default locale and press enter and locale generation will be completed and reboot the Galileo.

Wireless support for Intel Galileo

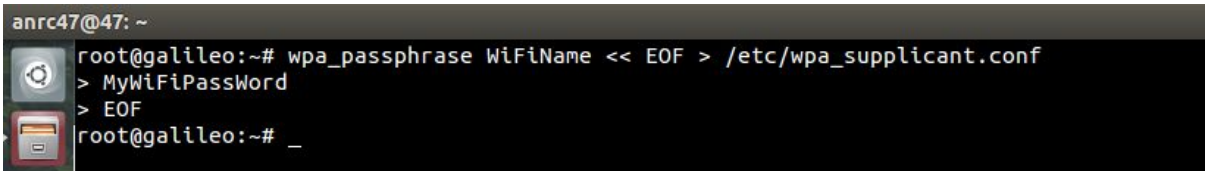
Wifi configuration in Galileo:

One of significant advantages of galileo over similar SBCs is availability of mini PCI Express connector. It can be used to add a high speed WiFi adapter without taking the USB port.

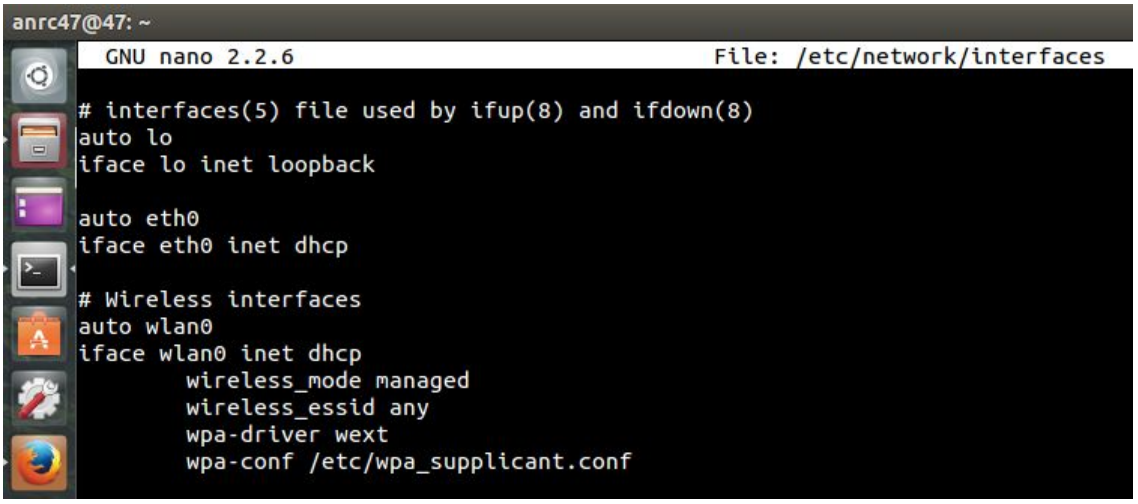
- Follow the below link to install the firmware for Intel® Centrino® Advanced-N 6235:
  - **http://www.malinov.com/Home/sergey-s-blog/intelgalileo-addingwifi**

After proper firmware is installed use the following steps to configure wireless card (this assumes using WPA security).

- Run the following command to generate wpa_supplicant configuration file for your network. Replace **WiFiName** with the SSID of your wireless network and **MyWiFiPassWord** with the real password. Run the following command
  # wpa_passphrase IISC1 << EOF > /etc/wpa_supplicant.conf



- If you wish to configure your galileo to connect to the wireless network automatically, edit /etc/network/interfaces file. In terminal, run the following command
  # nano /etc/network/interfaces



After adding wireless interfaces block to the interfaces file. To save it, press ctrl+o and enter. To exit, press ctrl+x.

- Restart the networking services, run the following command
  # /etc/init.d/networking restart

```
File  Edit  View  Search  Terminal  Help
root@galileo:~# /etc/init.d/networking restart_
```

  or run
  # **ifdown wlan0 && ifup wlan0**

```
anrc47@47: ~
root@galileo:~# ifdown wlan0 && ifup wlan0_
```

Wifi configuration in Galileo when multiple networks are available:

- Edit /etc/wpa_supplicant file. Run the following command
  # **nano /etc/wpa_supplicant.conf**

Use **priority** keyword in the configuration file as shown below.

```
anrc47@47: ~
  GNU nano 2.2.6                    File: /etc/wpa_supplicant.conf

# reading passphrase from stdin_
network={
        ssid="Google"
        psk="googleServer"
        priority=3
}

#Add a similar network block

network={
        ssid="DlinkTest"
        psk="desedlink"
        priority=5
}

#The priority of a network when selecting among multiple networks;
#a higher value means a network is more desirable.
```

After the configuration, restart networking services.

Load the i2c driver

In Galileo, by default debian os won't load the i2c drivers.

Load i2c drivers manually:

```
anrc47@47: ~
root@galileo:~# modprobe i2c-dev
root@galileo:~# ls /dev/ | grep i2c
i2c-0
root@galileo:~# _
```

Load i2c drivers automatically at boot time:

- Edit /etc/modules file. In terminal, run the following command
  # nano /etc/modules.
- Add **i2c-dev** line to the file and save it

```
anrc47@47: ~
  GNU nano 2.2.6                                File: /etc/modules

#Always load these modules
pch_udc
g_serial

#Load i2c driver at boot
i2c-dev_
```

Configuration for Intel Galileo Light Map (*):

* These steps are mandatory.

- Copy **cxcfiles** folder to the bin directory of the Galileo.

  - **Method 1: Use scp command in ubuntu terminal to transfer cxcfiles folder**
    - In ubuntu terminal, run the following command

    $ scp -r <path to the cxcfiles directory> root@<ip address of Galileo>:/bin/
    Example: **$ scp -r /home/anrc/IoTMaterials/cxcfiles root@10.42.0.12:/bin/**

```
anrc47@47: ~                                                    En  11:16 AM  Saru047
anrc47@47:~$ scp -r /home/anrc47/cxcfiles/ root@192.168.0.106:/bin/
root@192.168.0.106's password:
rebootgalstatcheck                                    100%  102     0.1KB/s   00:00
crongalstatcheck                                      100%  100     0.1KB/s   00:00
rebootstatuscheck.py                                  100% 3223     3.2KB/s   00:00
cronstatuscheck.py                                    100% 3458     3.4KB/s   00:00
anrc47@47:~$ _
```

  - **Method 2: Download cxcfiles tar file from cxc.co.in site**
    - **In Galileo terminal, run the below commands**
      # wget http://cxc.co.in/cxcfiles.tar.gz

```
Terminal
anrc47@47: ~
root@galileo:~# wget http://cxc.co.in/cxcfiles.tar.gz
--2015-04-27 12:46:07--  http://cxc.co.in/cxcfiles.tar.gz
Resolving cxc.co.in (cxc.co.in)... 107.180.3.99
Connecting to cxc.co.in (cxc.co.in)|107.180.3.99|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1771 (1.7K) [application/x-gzip]
Saving to: `cxcfiles.tar.gz'

100%[=====================================>] 1,771       --.-K/s   in 0s

2015-04-27 12:46:07 (6.21 MB/s) - `cxcfiles.tar.gz' saved [1771/1771]

root@galileo:~# _
```

- Extract the downloaded tar file, so run the below commands
  # tar -xvf cxcfiles.tar.gz

```
Terminal
anrc47@47: ~
root@galileo:~# tar -xvf cxcfiles.tar.gz
cxcfiles/cronstatuscheck.py
cxcfiles/rebootstatuscheck.py
cxcfiles/crongalstatcheck
cxcfiles/
cxcfiles/rebootgalstatcheck
root@galileo:~# _
```

- **copy the unzipped cxcfiles folder to the bin diectory**
  # cp -r cxcfiles /bin

```
Terminal
anrc47@47: ~
root@galileo:~# cp -r cxcfiles /bin
root@galileo:~# _
```

Follow either **method 1** or **method 2**

➢ In galileo, give executable permissions for cxcfiles. To do that run the following command
 # chmod 777 -R /bin/cxcfiles

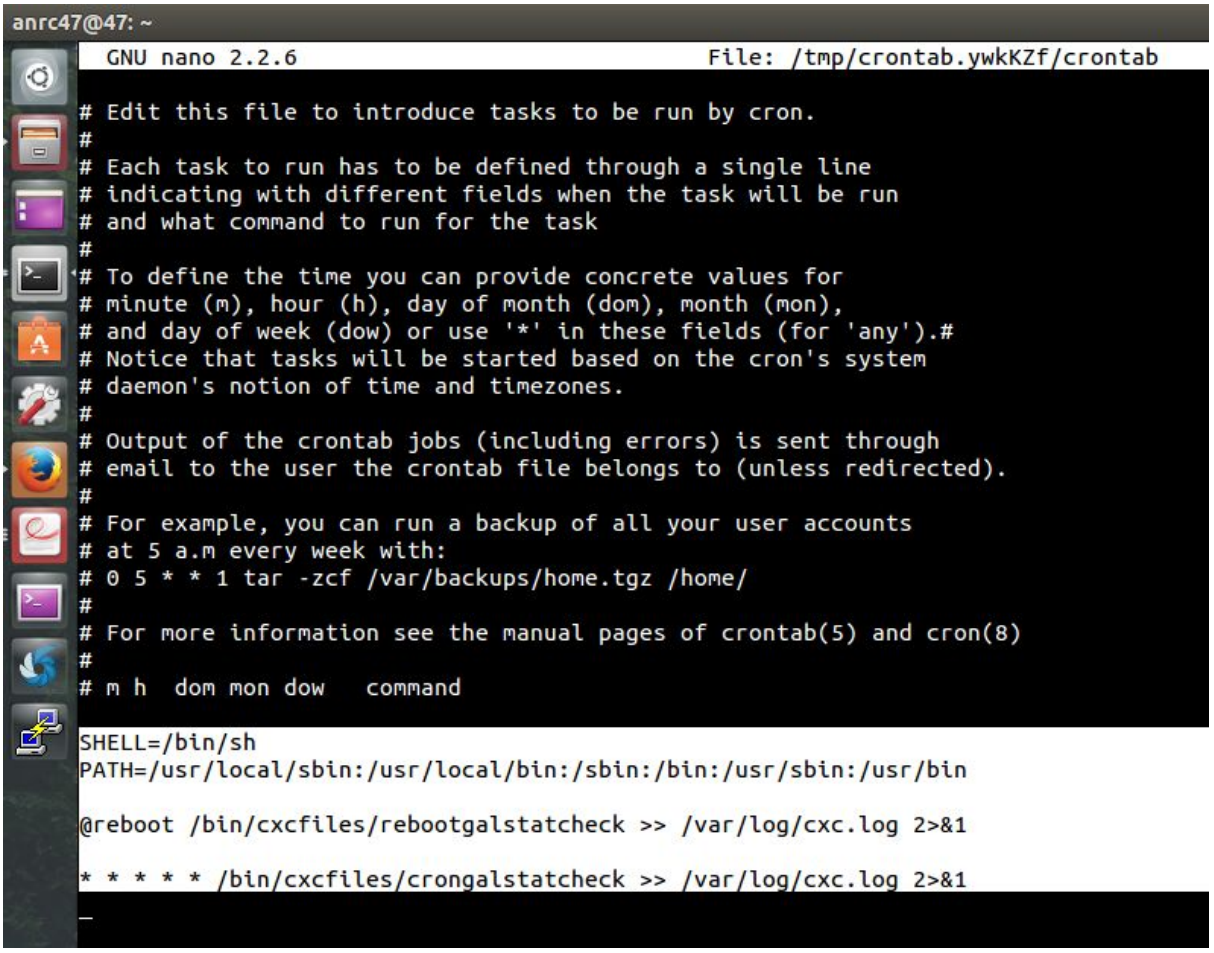
```
anrc47@47: ~
root@galileo:~# chmod 777 -R /bin/cxcfiles/
```

➢ In galileo, edit crontab file. In terminal, run the following command
 # crontab -e

➢ Add the following lines to the crontab file and save it

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
@reboot /bin/cxcfiles/rebootgalstatcheck >> /var/log/cxc.log 2>&1
***/15 * * * * /bin/cxcfiles/crongalstatcheck >> /var/log/cxc.log 2>&1**

To save it, press ctrl+o and enter. To exit, press ctrl+x.



Note:

- Technical support will only be offered to students who are using linux operating system in their host machine (Ubuntu 14.04 LTS distribution). To download Ubuntu 14.04, click here
- All our experimentation is tested with linux operating system
- Technical support will not be offered for windows and mac OS