

Evaluation of Intel Architecture
An Experimental Manual for
Computer Architecture, Advanced Microprocessor,
System On Chip (SoC) and Compiler Design
In association with Intel Collaboration Program



Designed by: Zeenat Shareef, MTech (Mobile and Pervasive Computing)
Under the guidance of: Dr. S.R.N Reddy, HOD and Associate Professor, CSE
Mr. Naveen , Mr. Sumit Verma, Intel

Department of Computer Science
Indira Gandhi Delhi Technical University for Women
Kashmere Gate, Delhi-110006

LIST OF EXPERIMENTS

EXP. No	Description of Experiment
1.	To familiarize with Intel Edison.
2.	Write the steps to install the drivers and IDE for Intel Edison
3.	Write the steps to configure Intel Edison and enable the WIFI module
4.	To enable the Bluetooth module in Intel Edison and connect with a device.
5.	Write the steps to blink the LED on the Intel Edison using Eclipse CDT remote explorer(WiFi).

EXPERIMENT 1

AIM: To familiarize with Intel Edison.

INTEL EDISON- A SOC based on Intel Atom

The Intel Edison compute module is designed to lower the barriers to entry for anyone prototyping and producing IoT and wearable computing products. Intel Edison contains the core system processing and connectivity elements: processor, PMIC, RAM, eMMC, and Wi-Fi/BT. Intel Edison is a module that interfaces with end-user systems via a 70-pin connector. The Intel Edison compute module does not include any video input or output interfaces (MIPI CSI, MIPI DSI, HDMI, etc.). Internal image processing and graphics processing cores are disabled (ISP, PowerVR, VED, VEC, VSP, etc.). Intel Edison relies on the end-user support of input power.

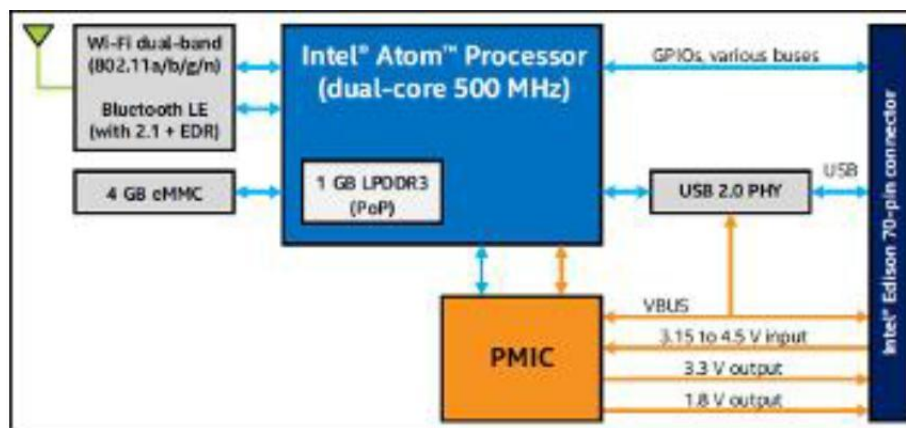


Fig1: Block Diagram of Intel Edison

i) SoC: Main SoC of the board is new Intel Atom "Tangier" (Z34XX), produced with 22 nm which combines a dual-core Atom running Yocto Linux at 500MHz with Intel's MCU-like Quark processor clocked at 100MHz. The Quark is currently inactive, but will eventually run a "ViperOS" RTOS derived from the VxWorks platform offered by Intel subsidiary Wind River. "Tangier," a stripped down version of Intel's Atom Z34xx ("Merrifield").

Intel Atom Processor Z34xx Series is the next generation 22 nm SoC product targeted for the smartphone market segment. The SoC contains dual IA-32 cores operating at 500 MHz.

ii) Managed NAND (eMMC) flash

Edison uses 4 GB of managed NAND to store the file system and user data. Managed NAND flash contains a full MMC controller, wear-leveling firmware, and all the other features that are typically found in MMC cards, except it is available in a small BGA form-factor.

- Bus mode – Data bus width: 1 bit (default), 4 bits, 8 bits
 - Data transfer rate: up to 200 MBps (HS200)

iii) DDR SDRAM

Edison supports 1 GB LPDDR3 memory at speeds up to 1033 MT/s.

- 8 banks , Row addresses R0-R13 , Column addresses C0-C9
- Dual-channel 32 bits
- 400 MHz clock max (800 MT/s)

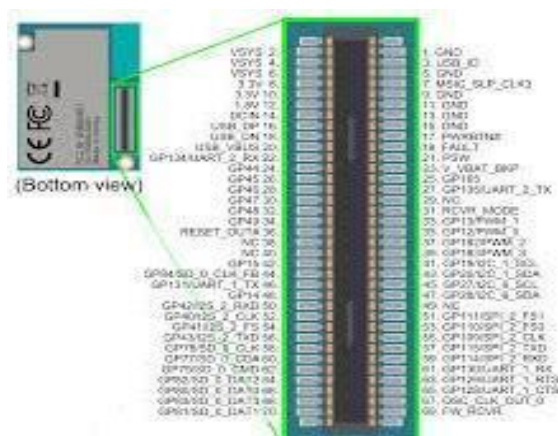
iv) Wi-Fi / BT module

The Murata integrated Wi-Fi BT module is built around a Broadcom BCM43340 Wi-Fi /BT device. The Broadcom BCM43340 single chip quad device provides the highest level of integration for a mobile or handheld wireless system, with integrated dual band (2.4 / 5 GHz) IEEE 802.11a/b/g/n MAC/baseband/radio with Bluetooth 4.0.

- Dual-band 2.4 GHz and 5 GHz IEEE 802.11 a/b/g/n.
- Single-stream IEEE 802.11n support for 20 MHz and 40 MHz channels provides PHY layer rates up to 150 Mbps for typical upper layer throughput in excess of 90 Mbps.
- Complies with Bluetooth Core Specification Version 4.0 with provisions for supporting future specifications. Bluetooth Class 1 or Class 2 transmitter operation.
- Security: – WPA and WPA2 (personal) support for powerful encryption and authentication.

v) 70-pin interface connector

The Edison module connects to the end user device via a 70-pin connector. The connector on Edison is a Hirose 70-pin DF40 Series “header” connector sometimes referred to as a “plug” connector.



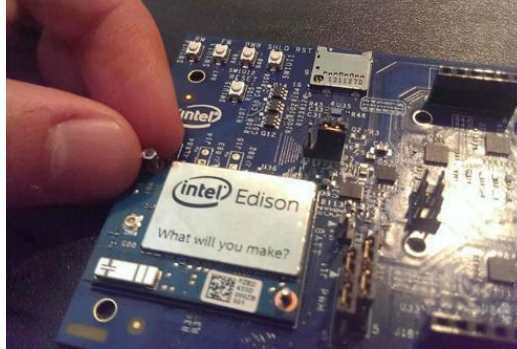
EXPERIMENT 2

AIM: Write the steps to install the drivers and IDE for Intel Edison.

REQUIREMENTS: Intel Edison, 12 V power supply, micro USB cable

PROCEDURE:

1. Unpack the Intel Edison board and screw the nuts in the expansion board. Here, we have the arduino expansion board.



2. Download the Windows Drivers setup 1.2.1 from <http://www.intel.com/support/edison/sb/CS-035180.htm> so that the Intel Edison will be connected as a COM port to the laptop.
3. Download the latest Yocto image from the above mentioned site and extract it on the Edison flash.
4. Download putty to connect through serial and SSH connections to access the linux side of Edison.(<http://www.putty.org/>)
5. Now install the IDE through which you would like to program the Edison
 1. Arduino Software 1.5.3 - Intel 1.0.4 software - <http://www.intel.com/support/edison/sb/CS-035180.htm>
 2. Eclipse(CDT) - <https://software.intel.com/en-us/iot/downloads>
 3. Intel XDK

EXPERIMENT 3

AIM: Write the steps to configure Intel Edison and enable the WIFI module.

REQUIREMENTS: Intel Edison, 12 V adapter, USB cable, Putty.

PROCEDURE:

1. Connect the 2nd micro USB cable(serial port) of Intel Edison with your system and open putty.
2. In putty, write the COM Port number(Open Device manager) for that and change the baud rate from 9600 to 115200 and click on open.
3. Press enter a couple of times and the login screen of Intel Edison opens up with name of the Linux Distribution prompting the user to enter the login name and password.

The default edison login is **root**

The default password is **root**.

4. To configure Intel Edison with a new name and password, write the following command
configure_edison --setup
5. A screen opens up which asks the user to enter the password. Write the password and press enter. This password will be the entry point to this device and login password.
6. Write the name of the device (atleast 5 characters long) and confirm it by pressing „Y“ for yes.
7. It then asks if the user wants to configure the wifi and connect to a network. Select „Yes“, and the system will start scanning for the available wifi connections.

The Network Name along with their SSID appears on the screen. Type the SSID of the network you would like to connect with. Then it will ask for the network password for the connection. Type in the password and press enter.

Here, the SSID and connection name is **5: EMBEDDED_LAB**

The password is **igdtuw#@#246**

8. If the connection is successful then it will give the IP address of the device through which one can connect and work with the Intel Edison.

So, here we get a message on the screen that the connection has been successful and get the IP address as 172.16.3.179.

To check if the connection is actually successful then write the IP address that you got here in the browser.

9. Now to wirelessly connect to Intel Edison through WiFi, open SSH in putty and type the IP address (Here, we have 172.16.3.179).

On successful connection, the Intel Edison command line opens up. Open this will username (default- root) and password- the one you typed in step 5.

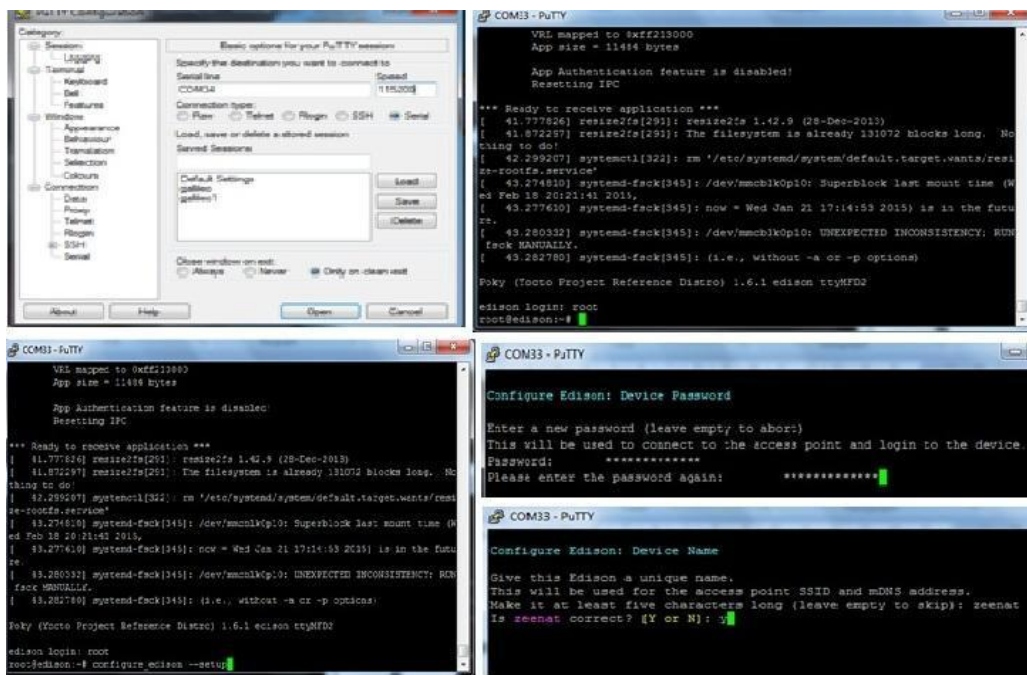
Now you have enabled the Wifi Module and can write commands as you used to do in serial terminal.

To check if the device is connected to the internet or not, type **ifconfig** in the command line.

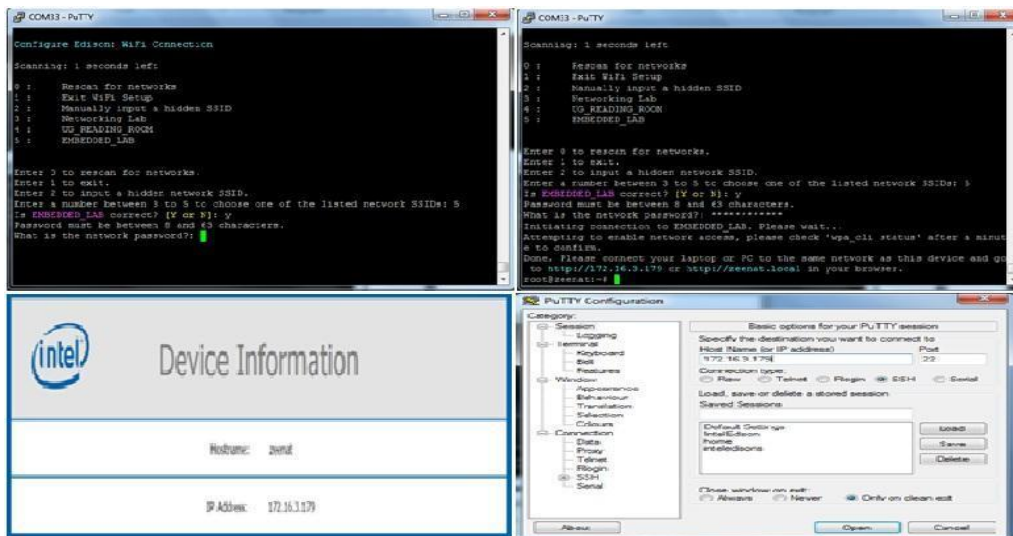
There in the wlan0 section in the screen, you see the inet address as 172.16.3.179. Thus this proves that you are connected to the network.

Thus you can work with your Edison wirelessly through the network.

SNAPSHOTS:



Snapshots for Steps [2],[3],[4],[5],[7]



Snapshots for Steps [8],[9],[10]

RESULT:

Thus we have configured the Intel Edison with new name and password. We have successfully connected to a wifi network and accessed our Intel Edison via that network.

EXPERIMENT 3

AIM: To enable the Bluetooth module in Intel Edison and connect with a device.

REQUIREMENTS: Intel Edison, 12 V adapter, USB cable, Putty.

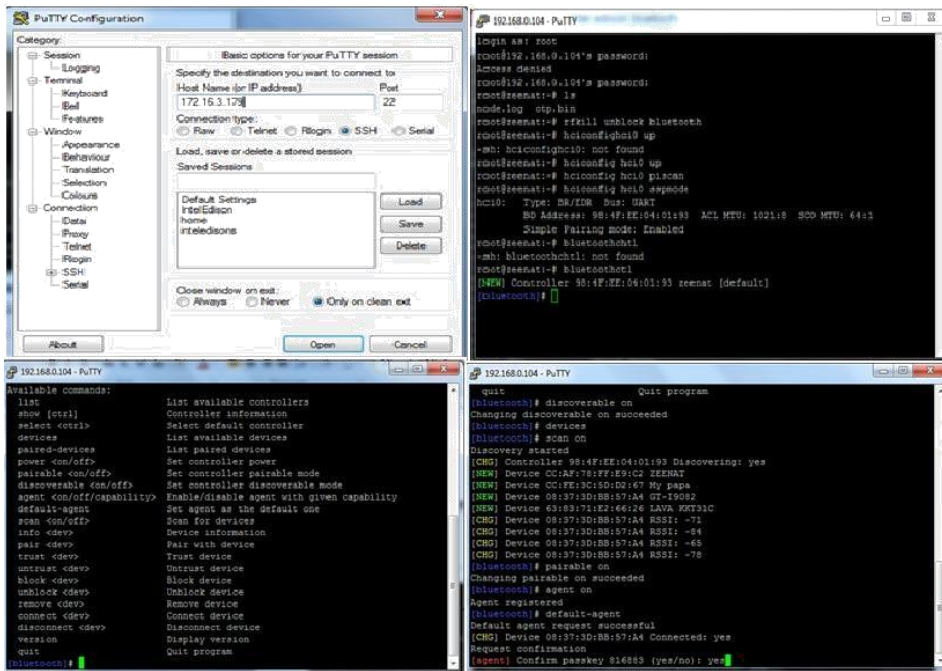
PROCEDURE:

1. As we have already connected our Intel Edison with the network, we can have access to Linux Distribution via SSH after providing the IP address of the Edison device.
2. You will get a login screen. Enter by providing the default login as **root** and your password.
3. Write the following command

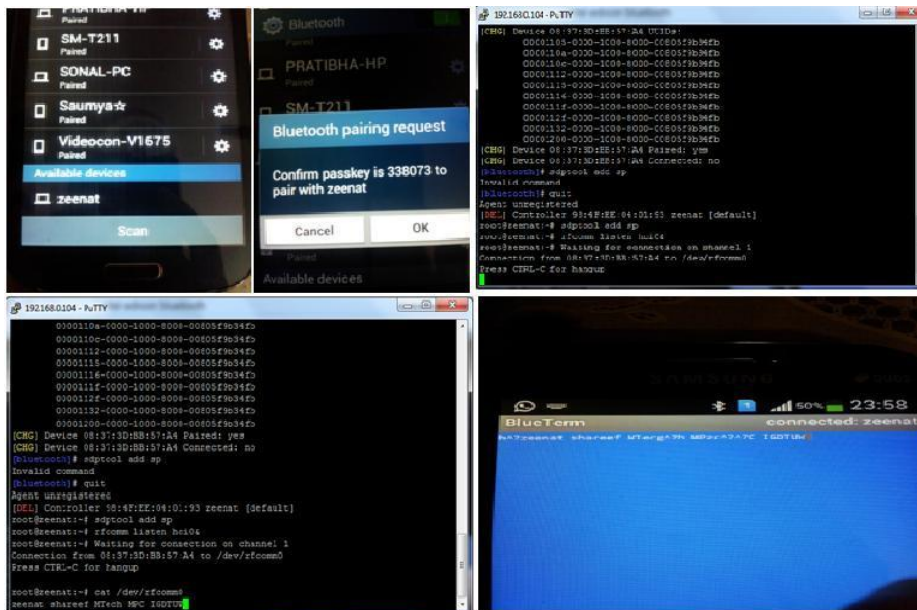
rfkill unblock bluetooth

This will unblock the Bluetooth.

4. Next to configure the UART and fire up the radio type **hciconfig hci0 up**
5. Now type **bluetoothctl** which will take us to the controller we are working with. You will enter the Bluetooth command line.
6. To have a view of all the available commands type **help** and you will get the screen as shown.
7. To let the device become discoverable and pairable type **discoverable on and pairable on**.
8. Now we will fire the agent. So type **agent on** and then **default-agent**. This will look after pairing and all.
9. Now take your device- here we have taken the device as an android phone and scan for nearby Bluetooth radios. Select the Intel Edison device name and the phone with it.
10. Now quit from the terminal
11. Now configure the Bluetooth to listen to other Bluetooth devices by typing **rfcomm listen hci0&**.
12. Now open **blueterm** in your android phone and connect with Intel Edison. Thus connection is established between Intel Edison and the android phone.
13. To check if the data sent by the android phone is received or not, type
cat /dev/rfcomm0.
14. Now, whatever you type from your blueterm should be received in the Intel Edison screen



Snapshots for Steps [1],[2],[6],[8]



Snapshots for Steps [9],[10],[13],[14]

RESULT: Thus we have established connection with another bluetooth device and sent characters.

EXPERIMENT 4

AIM: Write a program to blink the LED on the Intel Edison using Eclipse CDT remote explorer(WiFi).

REQUIREMENTS: Intel Edison, 12 V adapter, USB cable, Putty, Eclipse IOT kit for windows

PROCEDURE:

1. Download Intel-iot-devkit (<https://software.intel.com/en-us/iot/downloads>)
2. Open devkit-launcher.
3. Go to Remote System Explorer
4. On the left corner of the screen you will have to create a new connection.
5. To connect to the remote device, make a new SSH connection.
6. Define the new connection with the IP address of the device to connect to and give a name to this connection.
7. After establishing the connection you will get setup as shown in the figure and you will have to right click on the Edison and select connect.
8. It will then prompt you to get the password for the network.
9. Now you will get a message that the system is connected with the terminal of linux.
10. Now go to C++ and select the remote led blinking example on the project explorer
11. Debug the application. Now go to Run->Run Configurations
Select the debug file that you want to run. The connection as the remote connection of Edison. Click on Apply. Then Run.
12. You will get the output on the terminal and the Led will blink on the board.

CODE(Sample Example Code):

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>

#include "mraa/gpio.h"

int
main(int argc, char **argv)
{
    mraa_platform_t platform = mraa_get_platform_type();
    mraa_gpio_context gpio;

    char board_name[] = "Intel Edison LED Blinking";
    int ledstate = 0;

    switch (platform) {
        case MRAA_INTEL_GALILEO_GEN1:
```

```

strcpy(board_name, "Intel Galileo
Gen1"); gpio = mraa_gpio_init_raw(3);
break;
case MRAA_INTEL_GALILEO_GEN2:
strcpy(board_name, "Intel Galileo
Gen2"); gpio = mraa_gpio_init(13);
break ;
default:
gpio = mraa_gpio_init(13);
}

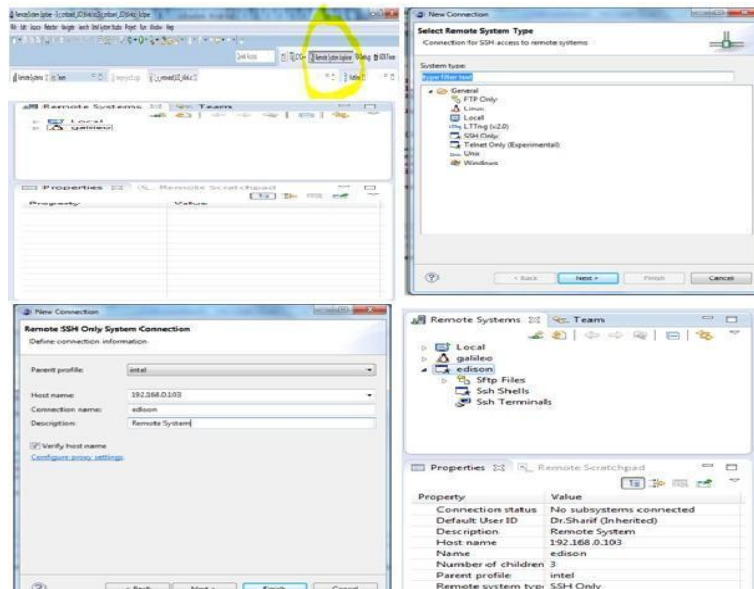
fprintf(stdout, "Welcome to libmraa\n Version: %s\n Running on %s\n",
mraa_get_version(), board_name);

mraa_gpio_dir(gpio, MRAA_GPIO_OUT);
for (;;) {
ledstate = !ledstate;
mraa_gpio_write(gpio,
!ledstate); sleep(1);
}

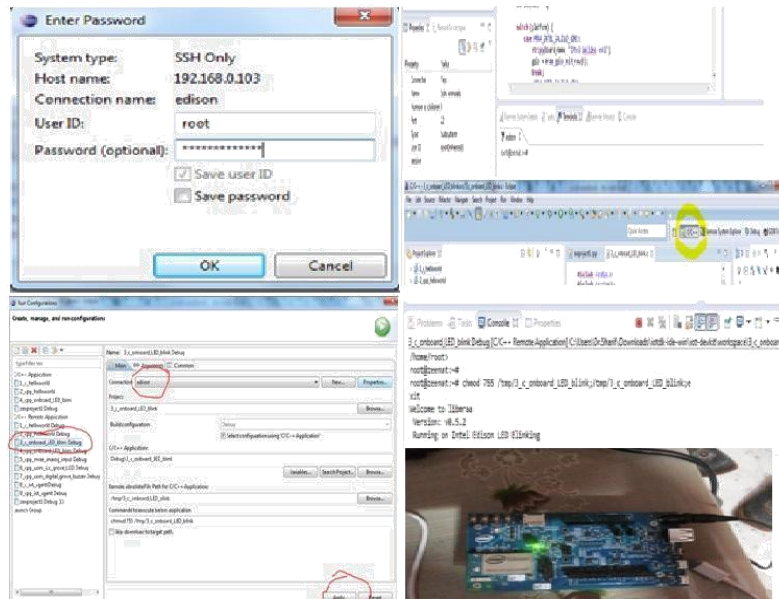
return 0;
}

```

SNAPSHOTS



Snapshots for Steps [3],[4],[5],[6],[7]



Snapshots for Steps [8],[9],[10],[11],[12]

RESULT: Thus we have successfully program the Intel Edison using Eclipse C++ Remote Explorer.