

**Design & Development of mobile device using  
Raspberry Pi: A Practical Approach  
(Experimental Manual for M.Tech Students) for  
SoC and mobile design (Version 1, 2013-2014) with  
support of ITRA, MHRD and NOKIA projects**



Designed & Developed By: Ms. Nidhi Agarwal

Under the Guidance of: Dr. SRN Reddy, Associate Professor, CSE

**Computer Science & Engineering Department  
Indira Gandhi Delhi Technical University for Women  
Kashmere Gate, Delhi-110006**

## LIST OF EXPERIMENTS

<b>Exp No.</b>	Description of experiment	Page No.
<b>Exp. 1</b>	To study the architecture of SOC Broadcom-2835 application board of Raspberry Pi.	3-10
<b>Exp. 2</b>	To demonstration the OS (Debian) for RPi in a SD card preparation, configuration of Raspberry Pi during first booting and use of remote SSH like putty	11-17
<b>Exp.3</b>	To demonstrate the basic linux commands on Raspberry pi.	18-20
<b>Exp.4</b>	To create a database & Store the value in Raspberry Pi.	21-23
<b>Exp.5</b>	To install Android on Raspberry Pi.	24
<b>Exp.6</b>	To Setup RPi first time without using screen, mouse, keyboard.	25-26
<b>Exp.7</b>	To interface ADC at GPIOs of Raspberry Pi for measuring analog voltage.	27-30

## Experiment No.-1

**1.1 Objectives:** To Study the architecture of SOC Broadcom-2835 application board of Raspberry Pi.

**1.2 Hardware requirement:** Raspberry Pi board (Broadcom BCM 2835) Model B.

### 1.3 Introduction:

Ultra-low-cost credit-card-sized single-board Linux computer Raspberry Pi developed by Raspberry foundation (UK), is a small, powerful and lightweight ARM11 (Broadcom 2835) based development board or a computer which is able to perform operations similar to a PC. The powerful graphics capabilities and HDMI video output make it ideal for multimedia applications such as media centres and narrowcasting solutions. With access to the internet, through Ethernet or Wi-Fi (with a USB dongle), and a high definition output, the Raspberry Pi is an incredibly versatile piece of computing kit. The Raspberry Pi is based on a Broadcom BCM2835 chip that is based on ARMv6. It does not feature a built-in hard disk or solid-state drive, instead relying on an SD card for booting and long-term storage.

This single board computer is inexpensive yet comes packed with ethernet, USB, a high powered graphics engine, digital I/O ports and enough CPU power to accomplish your projects.[1.2]

The Foundation provides Debian and Arch Linux ARM distributions for download. Tools are available for Python as the main programming language.[1.1]

The Raspberry Pi Model B (figure 1.1) features [4]:

- The Broadcom BCM2835 ARM11 (used in most smart-phones) 700Mhz ‘System on Chip’ Processor (Similar performance to a 300MHz Pentium 2 Processor).
- Integrated Videocore 4 Graphics Processing Unit (GPU) capable of playing Full 1080p High Definition Blu-Ray Quality Video (Roughly equivalent graphical processing power of an Xbox 1).
- 512Mb SDRAM
- The free, versatile, and highly developer friendly Debian GNU/Linux Operating System
- 2 x USB Ports
- HDMI Video Output
- RCA Video Output
- 3.5mm Audio Output Jack
- 10/100Mb Ethernet Port for Internet Access
- 5V Micro USB Power Input Jack
- SD, MMC, SDIO Flash Memory Card Slot

- 26-pin 2.54mm Header Expansion Slot (Which allow for peripherals and expansion boards)

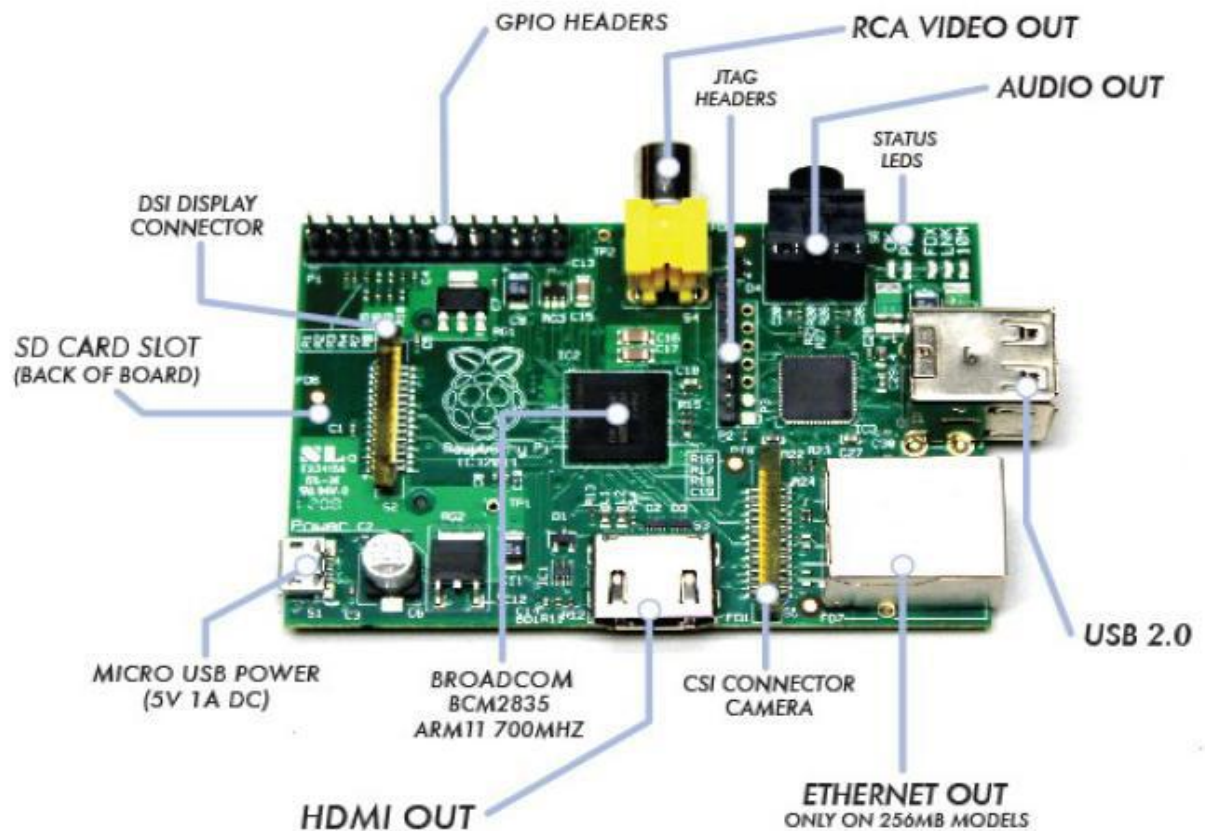


Figure 1.1: Raspberry pi (top view)

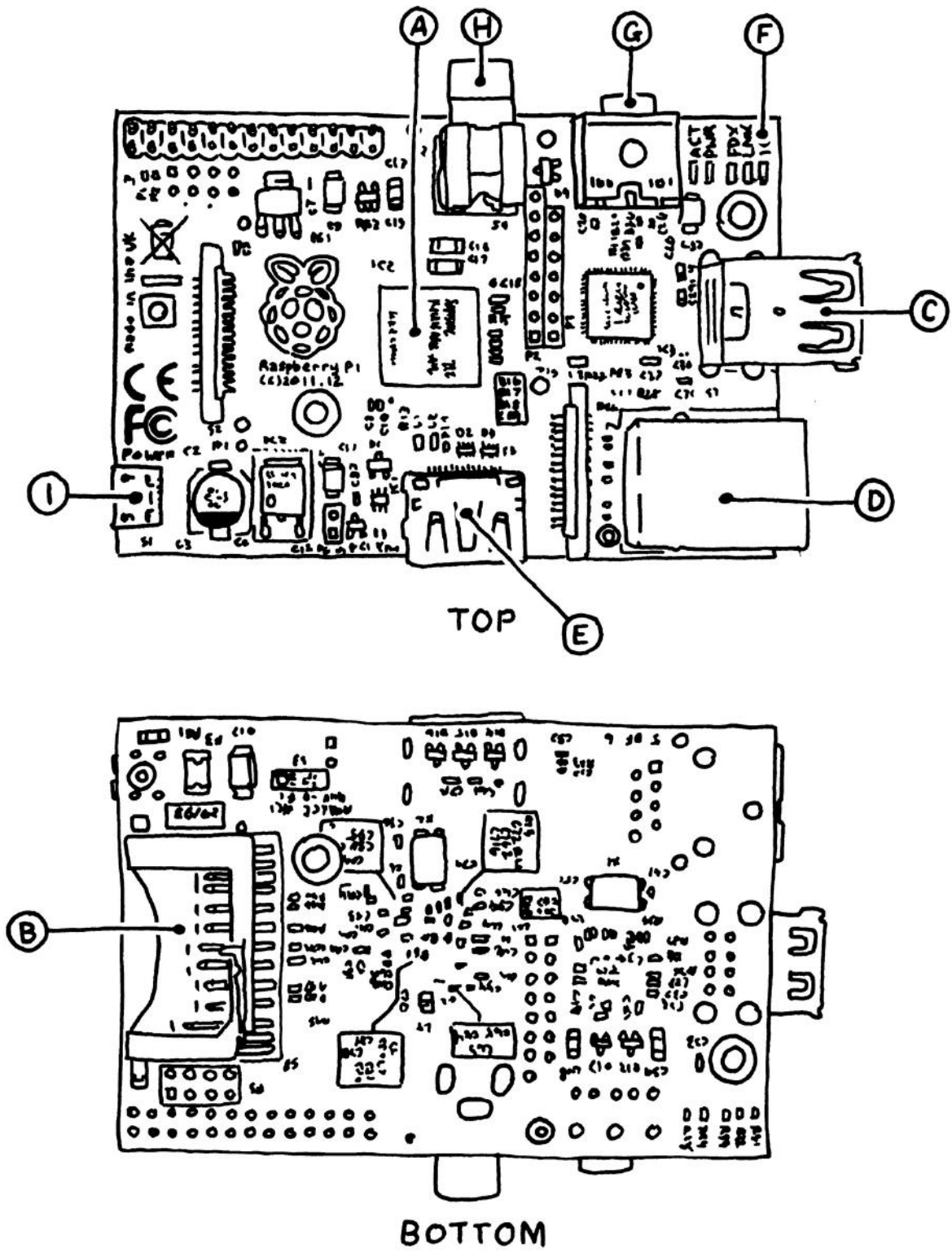


Figure 1.2: Scratch view of raspberry pi

**Table 1.1: Hardware Specification & Discription:**

Mark	Name	Discription
<b>A</b>	<b><i>The Processor</i></b>	This chip is a 32 bit, 700 MHz System on Chip, which is built on the ARM11 architecture(ARMv6 instruction set). SoC-Broadcom BCM2835
<b>B</b>	<b><i>The Secure Digital Card slot</i></b>	SD / MMC / SDIO card slot (3.3V card power support only)
<b>C</b>	<b><i>The USB port</i></b>	Model B has 2 Nos. USB 2.0 ports. Model A have only one current limited USB port.
<b>D</b>	<b><i>Ethernet port</i></b>	Model B has standard RJ45 connector and Model A does not.
<b>E</b>	<b><i>HDMI connector</i></b>	The HDMI port provides digital video and audio output hence TV can be connected. 14 different video resolutions are supported.
<b>F</b>	<b><i>Status LEDs</i></b>	<p><b>ACT</b> Green Light when the SD card is accessed (marked OK on earlier boards).</p> <p><b>PWR</b> Red Hooked upto 3.3V power.</p> <p><b>FDX</b> Green ON if network adapter is full duplex.</p> <p><b>LNX</b> Green Network activity light.</p> <p><b>100</b> Yellow On if the network connection is 100Mbps</p>
<b>G</b>	<b><i>Analog Audio out</i></b>	This is a standard 3.5mm mini analog audio jack,intented to drive high impedance load(like amplified speakers).
<b>H</b>	<b><i>Composite video out</i></b>	This is a standard RCA-type jack that provides NTSC or PAL video signals.
<b>I</b>	<b><i>Memory (SDRAM)</i></b>	512 MB in Model B and 256 MB in Model A
<b>J</b>	<b><i>Supportable Operating Systems</i></b>	Arch Linux ARM, Debian GNU/Linux, Gentoo, Fedora, FreeBSD, NetBSD, Plan 9, Raspbian OS, RISC OS, Slackware Linux

## 1.4 BLOCK DIAGRAM:

### BCM2835 Block Diagram

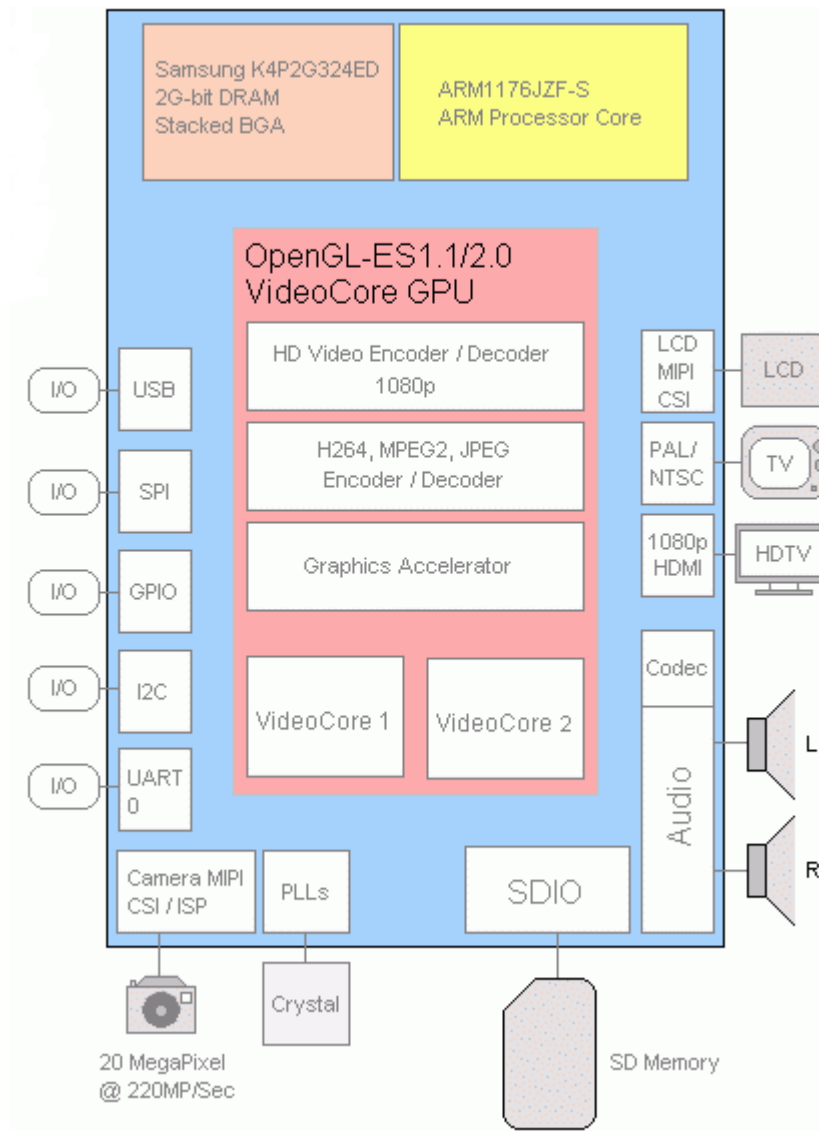


Figure 1.3. Block diagram of Broadcom SOC<sup>[2]</sup>

### 1.4.1 DETAILED DESCRIPTION OF COMPONENTS:

- 1) **System Timer:** The System Timer peripheral provides four 32-bit timer channels and a single 64-bit free running counter. Each channel has an output compare register, which is compared against the 32 least significant bits of the free running counter values.
- 2) **The Processor:** At the heart of the Raspberry Pi is the same processor you would have found in the iPhone 3G and the Kindle 2, so you can think of the capabilities of the Raspberry Pi as comparable to those powerful little devices. This chip is a 32 bit,

700 MHz System on a Chip, which is built on the ARM11 architecture. ARM chips come in a variety of architectures with different *cores* configured to provide different capabilities at different price points. The Model B has 512MB of RAM and the Model A has 256 MB. (The first batch of Model Bs had only 256MB of RAM.).

- 3) **Interrupt controller:** The interrupt controller can be programmed to interrupt the processor when any of the status bits are set. The GPIO peripheral has three dedicated interrupt lines. Each GPIO bank can generate an independent interrupt. The third line generates a single interrupt whenever any bit is set.
- 4) **General Purpose Input/Output (GPIO):** 3.3 volt logic via 26 pin header (**NOT** 5 volt or short tolerant) Pins can be configured to be input/output. General Purpose Input/Output (GPIO) is a generic pin on a chip whose behavior can be controlled by the user at run time. True GPIO (General Purpose Input Output) pins that you can use to turn LEDs on and off etc. I2C interface pins that allow you to connect hardware modules with just two control pins. SPI interface with SPI devices, a similar concept to I2C but uses a different standard.
- 5) **PCM / I2S Audio:** The PCM audio interface is an APB peripheral providing input and output of telephony or high quality serial audio streams. It supports many classic PCM formats including I2S. The PCM audio interface has 4 interface signals; PCM\_CLK - bit clock. PCM\_FS - frame sync signal. PCM\_DIN - serial data input. PCM\_DOUT - serial data output. PCM is a serial format with a single bit data\_in and out.
- 6) **DMA Controller:** The BCM2835 DMA Controller provides a total of 16 DMA channels. Each channel operates independently from the others and is internally arbitrated onto one of the 3 system busses
- 7) **UART:** The BCM2835 device has two UARTS. On mini UART and PL011 UART. The PL011 UART is a Universal Asynchronous Receiver/Transmitter. This is the ARM UART (PL011) implementation. The UART performs serial-to-parallel conversion on data characters received from an external peripheral device or modem, and parallel-to-serial conversion on data characters received from the Advanced Peripheral Bus (APB).
- 8) **Pulse Width Modulator:** PWM controller incorporates the following features:
  - Two independent output bit-streams, clocked at a fixed frequency.
  - Bit-streams configured individually to output either PWM or a serialized version of a 32-bit word.
  - PWM outputs have variable input and output resolutions.
  - Serialize mode configured to load data to and/or read data from a FIFO storage block, that can store up to eight 32-bit words.
  - Both modes clocked by clk\_pwm which is nominally 100MHz, but can be varied by clock manager.
- 9) **CPU**
  - ARM 1176JZF-S (armv6k) 700MHz
  - RISC Architecture and low power draw



- Not compatible with traditional PC software

#### 10) MEMORY

- **RAM:-** 512MB (Model B rev.2), 256 MB (Model A, Model B rev.1)
- **SD Card:-** At least 4GB SD card is needed, and it should be a Class 4 card. Class 4 cards are capable of transferring at least 4MB/sec. Some of the earlier Raspberry Pi boards had problems with Class 6 or higher cards, which are capable of faster speeds but are less stable. One can also use micro SD card using adapter. As there is no hard drive on the Pi; everything is stored on an SD Card. A protective case is needed as the solder joints on the SD socket may fail if the SD card is accidentally bent.

**11) Two USB 2.0 ports in RPi:** Dual USB sockets on RPi model B, single on model A. It can be expandable via regular or powered hubs. On the Model B there are two USB 2.0 ports, but only one on the Model A. Some of the early Raspberry Pi boards were limited in the amount of current that they could provide. Some USB devices can draw up 500mA. The original Pi board supported 100mA or so, but the newer revisions are up to the full USB 2.0 spec.

**12) Ethernet port:** The model B has a standard RJ45 Ethernet port. The Model A does not, but can be connected to a wired network by a USB Ethernet adapter (the port on the Model B is actually an onboard USB to Ethernet adapter). WiFi connectivity via a USB dongle is another option.

**13) HDMI connector:** The HDMI port provides digital video and audio output. 14 different video resolutions are supported, and the HDMI signal can be converted to DVI (used by many monitors), composite (analog video signal usually carried over a yellow RCA connector), or SCART (a European standard for connecting audio-visual equipment) with external adapters.

#### 14) Video:

- HDMI or (digital) DVI via cheap adaptor/cable
- Composite NTSC/PAL via RCA
- Wide range of resolutions
- NO VGA without an add-on, nontrivial converter (Adafruit)

#### 15) Audio:

- Via HDMI or from stereo jack
- Output only
- Support Maturity appears to be lagging

#### 16) Networking

- 10/100Mbps via RJ45 on model B
- Wireless via USB add-on supported

**17) Power:** There is no power switch on the Pi. Micro-USB connector is used to supply power (this isn't an additional USB port; it's only for power). Micro-USB was selected because cheap USB power supplies are easy to find.

- Primary power via microUSB plug: a 1Amp cell charger works well, but to use a USB hard drive, 2 Amp power is needed.

- Model A about a quarter amp less
- PC USB port does **not** work

**1.5 Result:** The Broadcom SOC board (2835) i.e. Raspberry pi block diagram and its features has been studied and board found working.

**1.6 Conclusion:-** Target board of such types can be used in low cost system designs using very less amount of components and can be used for many user defined applications or customizations

**1.7 Remarks:-** The stiff cables microcontroller on all sides make it hard to keep flat, and some of the components like the SD card slot can be mechanically damaged even through normal use. The Pi contains six layers of conductive traces connecting various components, unlike a lot of simple microcontroller PCBs that just have traces on the top and the bottom. There are four layers of thin traces sandwiched in between the top and bottom; if the board gets flexed too much you can break some of those traces in a manner that is impossible to debug.

#### **1.8 References:-**

- 1.1. [http://en.wikipedia.org/wiki/Raspberry\\_Pi](http://en.wikipedia.org/wiki/Raspberry_Pi)
- 1.2 <https://www.eventbrite.com/e/introduction-to-raspberry-pi-november-tickets-8830824233>
- 1.3. Getting Started with Raspberry Pi by O'REILLY.
- 1.4 <http://www.raspberrypi.org/>
- 1.6 <http://elinux.org/RaspberryPiBoard/>
- 1.7 <http://thepihut.com/pages/the-raspberry-pi>
- 1.8 Datasheet BCM2835

## Experiment No -2

**2.1 Objectives:** To demonstrate the OS (Debian) for RPi in a SD card preparation, configuration of Raspberry Pi during first booting and use of remote SSH like putty.

**2.2.1 Software Requirement:** Debian/Linux image file. Win32 Disk imager.

**2.2.2 Hardware requirement:** Raspberry Pi board (Broadcom BCM 2835) Model B. SD card at least 4GB, Monitor, USB Keyboard and USB mouse.

### 2.3 Prerequisites:

#### 1. Operating System:

Raspberry pi can support many operating systems as listed in experiment 1. Basically Debian/Linux is used image file of which (with name wheezy-raspbian.zip) can be download from the link given below:

<http://www.raspberrypi.org/downloads>

#### 2. Win32 disk:

This software will be used to burn/write the image file on the sd card.

Download it from the link given below.

<http://sourceforge.net/projects/win32diskimager/>

### 2.4 Preparing an RPi OS in a SD card:

#### Procedure:

**Step 1:** Extract the Image file.

**Step 2 :** Insert SD Card into pc/laptop (system on which OS and win32disk is downloaded).

Make sure that SD card is empty and have atleast 4GB memory.[2.1]

**Step 3 :** Open win32 disk. The UI view of it is shown in figure-2.1.

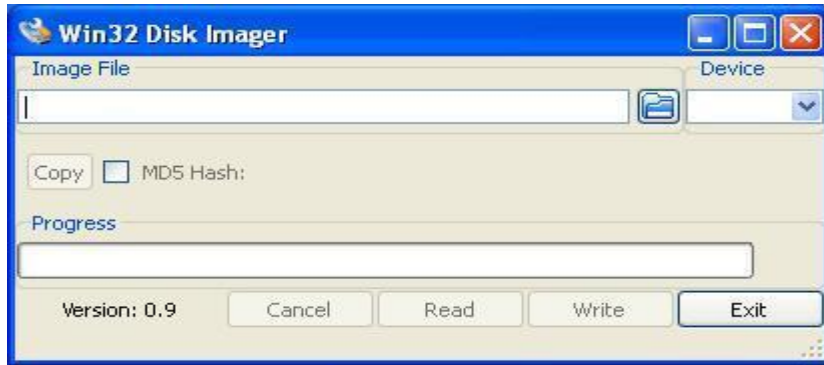
**Step 4 :** Browse the image file which has been downloaded from Raspberry pi website .

**Step 5:** Click on “write”.

**Step 6:** When writing is complete, remove the SD card and insert it into the raspberry pi.

**Step7:** Switch on the power supply of raspberry pi. Blinking Status of green LED shows that booting has started.

**Step 8:** Blinking Status of red LED shows that Raspberrypi is working fine.



**Figure-2.1 : Win32 disk imager user interface.**

## 2.5 Configuration of Pi:

### Procedure:

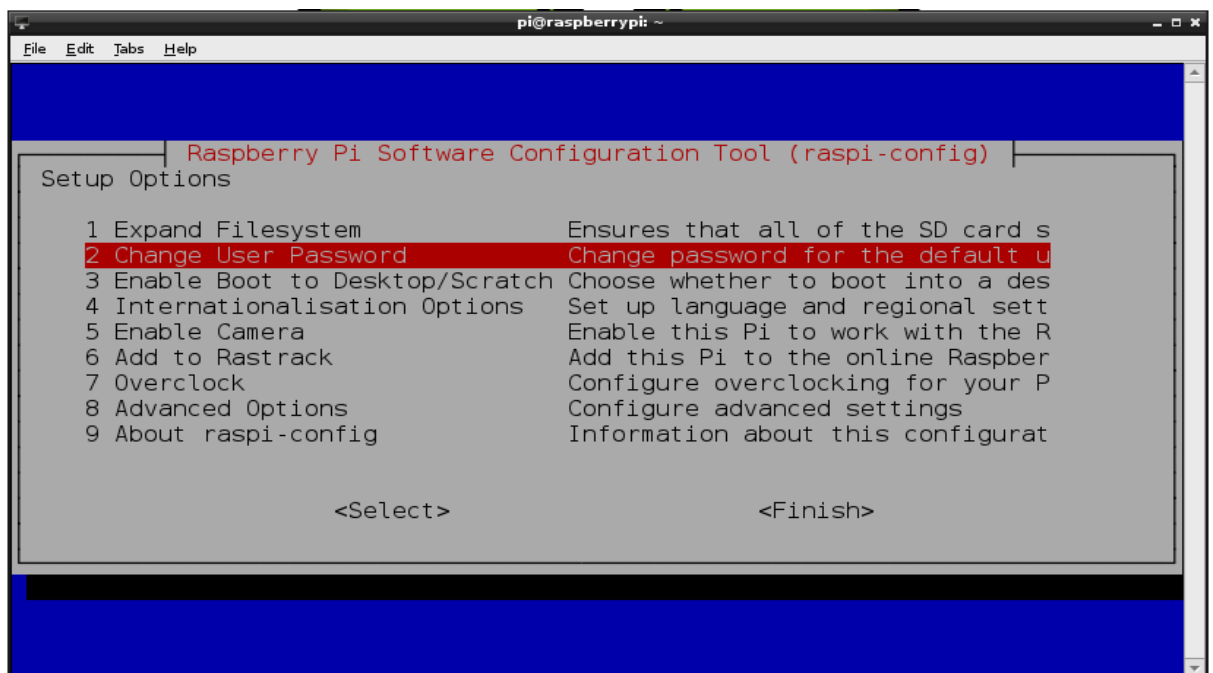
**Step1:** Default login is 'pi'. Password of RPi can be changed. By default it is 'raspberrypi'.

**Step2:** Enter into configuration mode by typing 'raspi-config' and press enter. Figure-2.2.

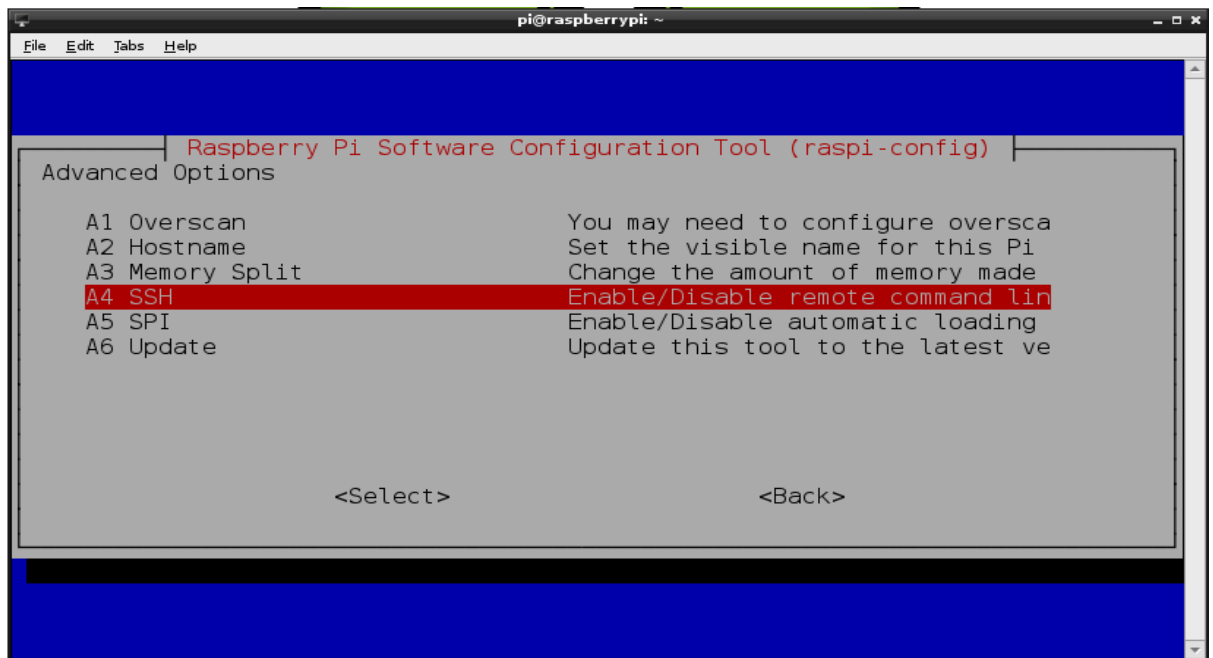
**Step3:** Enable SSH Server. In Model B, the SSH option is in 'Advanced Options'. Figure 2.3.

**Step4:** Press enter on Finish.

**Step5:** For desktop view, enter the command 'startx'. [2.2]



**Figure 2.2 Raspberry pi configuration, Change password**



**Figure 2.3 Raspberry pi configuration, SSH setting**

## 2.6 Static IP settings:

### Procedure:

**Step1 :** Open LX terminal.

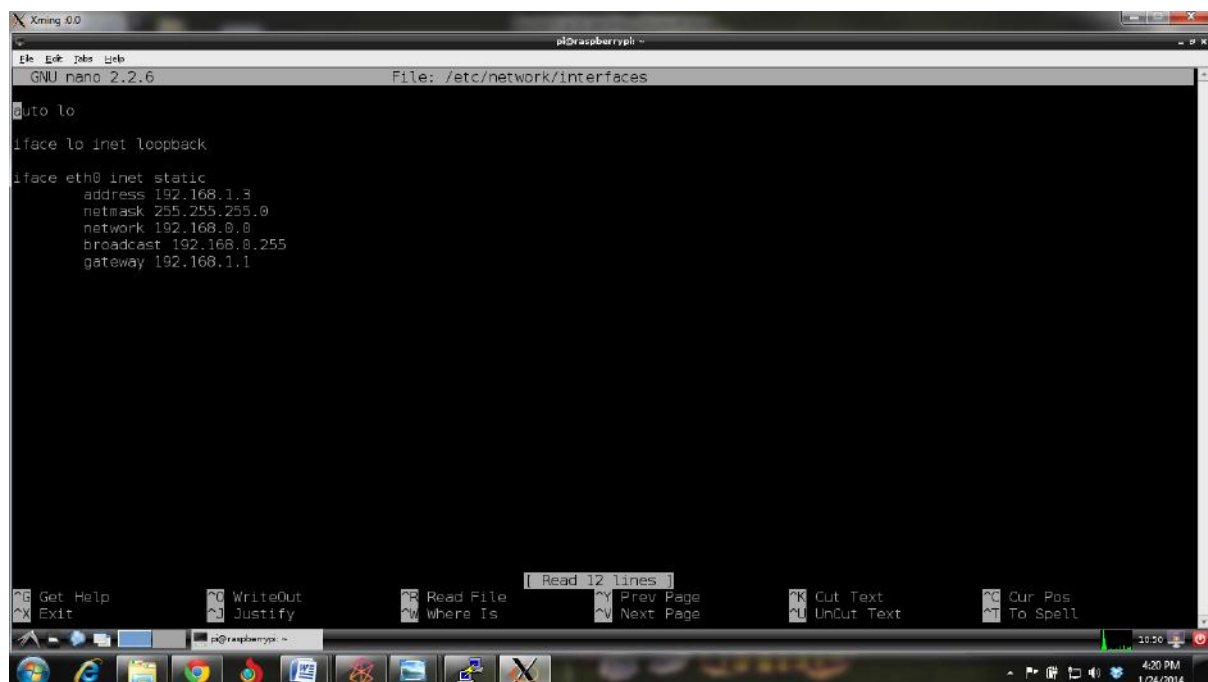
**Step2 :** Enter the command 'sudo nano /etc/network/interfaces'.

**Step3 :** In the end of 3<sup>rd</sup> line, replace the word 'dhcp' with "static".

**Step4 :** Enter the IP address ,netmask and gateway in next line. Figure-2.4.

**Step5 :** Press ctrl+x to save the file, then press 'y' and then press 'Enter'.

**Step6 :** For testing the static IP, enter the command 'ifconfig'. The second line of first paragraph will show the IP address of connected network. Make sure that Ethernet cable is connected.



**Figure 2.4 Static ip setting**

## 2.7 Remote access of Raspberry Pi:

Putty is used to connect two systems remotely using local Ethernet ports. By this way raspberry pi can be used in headless mode. The IP address of remote system is required in this case (If the remote system is in password mode, password will be required). Putty is specific to a particular operating system.

### Prerequisites:

1. **Putty:** Download from link given below:  
<http://putty.en.softonic.com/>
2. **Xming:** This software is used for the desktop view of the remote system. This software can be download from the given link below:  
<http://sourceforge.net/projects/xming/>

### Procedure:

**Step1 :** Enable SSH server

**Step2 :** One must know the IP address of the network by which pi can be accessed remotely using Ethernet connection. To know this type command ' ifconfig' into terminal command prompt, press enter and note down IP details

**Step3 :** Putty settings:

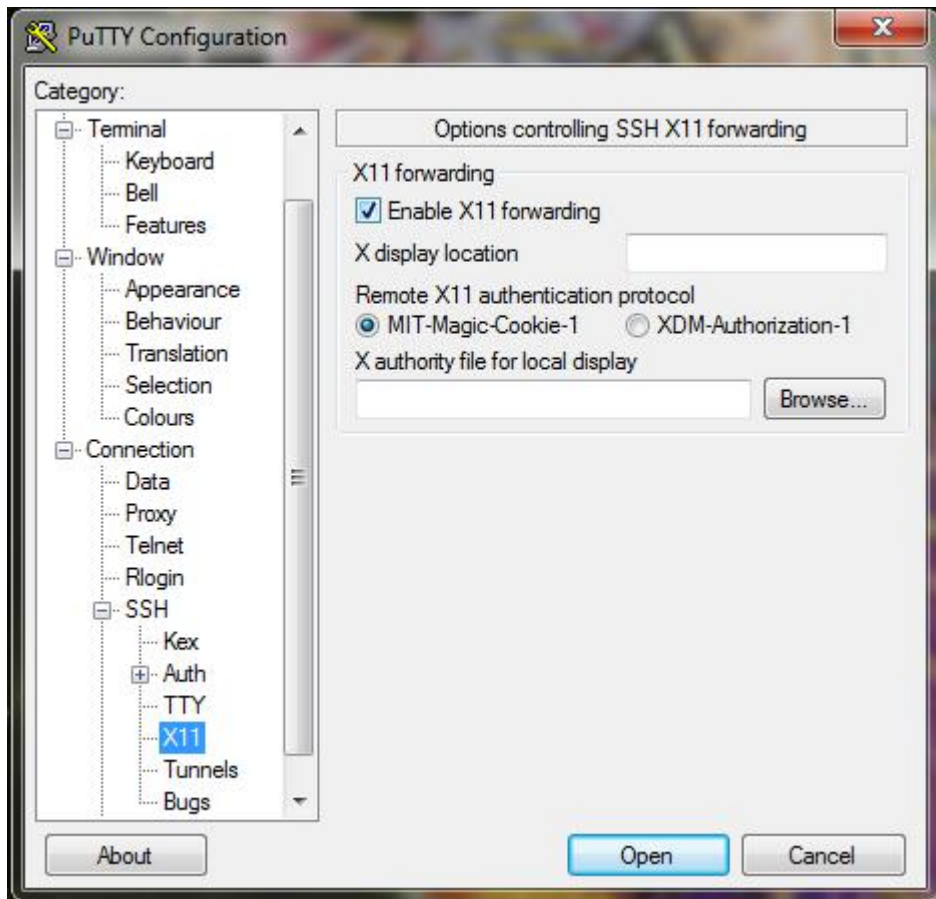
- Start Putty.
- Open SSH>X11>click on "Enable X11 forwarding" as shown in figure-2.5.
- Open Session>Enter a new session "raspberrypi".
- Enter IP address of the network connected with pi as shown in figure-2.6.
- Open.

**Step4 :** Enter the username and password as pi and raspberry respectively.

**Step5:** Open Xming. The Xming view is shown in figure-2.7.

**Step6:** Enter the command in command line of putty as 'startlxde'.

Desktop view of raspberry pi will be seen then.



**Figure 2.5 : Putty settings X11 enabling**



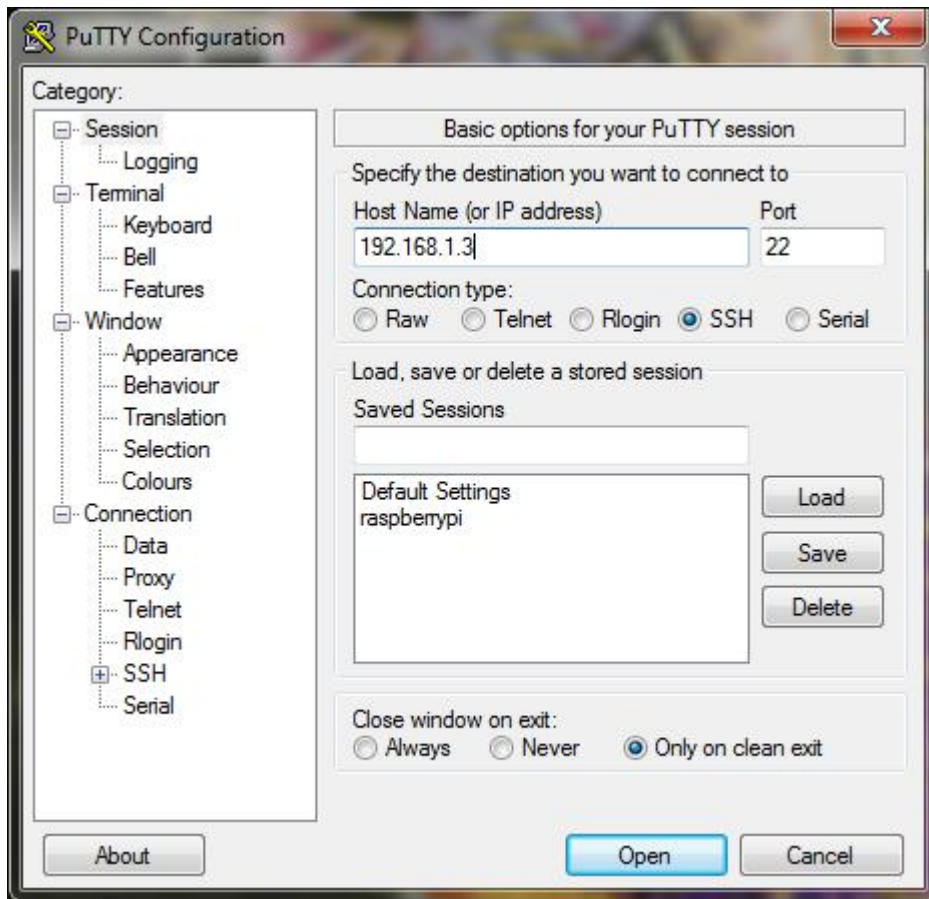


Figure 2.6:Putty setting, IP address

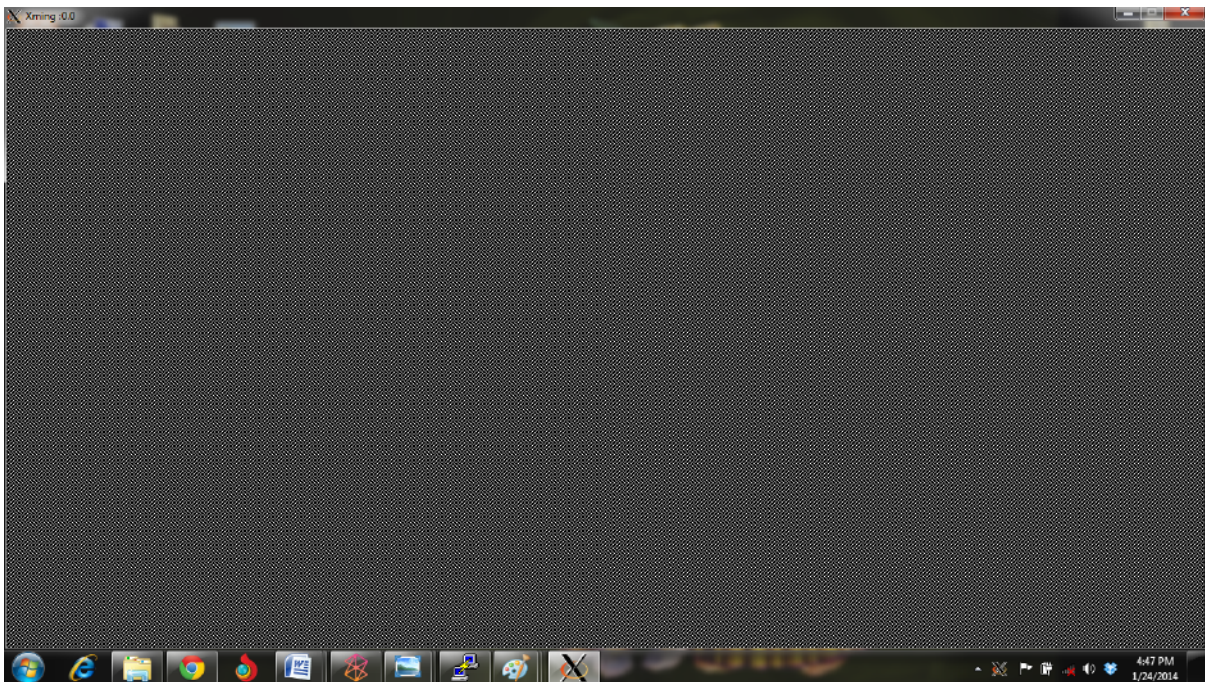


Figure 2.7: Xming UI view



**2.8 Result:** SD card image for raspberry pi is prepared and raspberry pi's initial settings have been done and checked.

**2.9 Conclusion:-** Small SD card is working as main memory of this system.

**2.10 Remarks:-** To avoid additional Monitor, keyboard and mouse one can use laptop/desktop for working on RPi, using SSH facility

### **2.11 References:-**

- 1.1. [http://en.wikipedia.org/wiki/Raspberry\\_Pi](http://en.wikipedia.org/wiki/Raspberry_Pi)
- 1.2. <https://www.eventbrite.com/e/introduction-to-raspberry-pi-november-tickets-8830824233>
- 1.3. Getting Started with Raspberry Pi by O'REILLY.
- 1.4 <http://www.raspberrypi.org/>
- 1.6 <http://elinux.org/RaspberryPiBoard/>
- 1.7 <http://thepihut.com/pages/the-raspberry-pi>
- 1.8 [http://en.wikipedia.org/wiki/Raspberry\\_Pi](http://en.wikipedia.org/wiki/Raspberry_Pi)

## Experiment No -3

**3.1 Objectives:** To demonstrate the basic linux commands on raspberry pi.

**3.2.1 Software Requirement:** Debian/Linux image file.

**3.2.2 Hardware requirement:** Raspberry Pi board (Broadcom BCM 2835) Model B. SD card at least 4GB, Monitor, USB Keyboard and USB mouse.

### 3.3 Procedure:

**Step1:** Switch on the raspberry pi.

**Step2:** Enter login and password and then write 'startx'.

**Step3:** Desktop view of raspberry pi will be open then.

**Step4:** Open LX Terminal.

**Step5:** Type 'ls' for directory view. Figure 3.1(1)

**Step6:** Type 'mkdir dir name' for creating directory. Figure 3.1(2)

**Step7:** Type 'rmdir dir name' for removing directory. Figure 3.1(3)

```
pi@raspberrypi ~ $ ls
2014-06-25-042003_1280x1024_scrot.png  Documents  indiecity  pitooth
bluetoothkeyboard.py                  enquire.py  led.py      python_games
bluetooth.py                           index.html  minirc.dfl  Scratch
button2.py                              index.html.1  nidhi       transmit.py
button.py                                index.html.2  ocr_pi.png
Desktop                                  index.html.3  pardeep

pi@raspberrypi ~ $ mkdir igdtuw
pi@raspberrypi ~ $ ls
2014-06-25-042003_1280x1024_scrot.png  igdtuw      nidhi
bluetoothkeyboard.py                  index.html  ocr_pi.png
bluetooth.py                           index.html.1  pardeep
button2.py                              index.html.2  pitooth
button.py                                index.html.3  python_games
Desktop                                  indiecity    Scratch
Documents                                led.py       transmit.py
enquire.py                               minirc.dfl

pi@raspberrypi ~ $ rmdir igdtuw
pi@raspberrypi ~ $ ls
2014-06-25-042003_1280x1024_scrot.png  Documents  indiecity  pitooth
bluetoothkeyboard.py                  enquire.py  index.html  python_games
bluetooth.py                           index.html.1  index.html.2  Scratch
button2.py                              index.html.2  index.html.3  transmit.py
button.py                                index.html.3  ocr_pi.png
Desktop                                  indiecity    pardeep
Documents                                led.py       minirc.dfl
```

Figure 3.1 Linux commands

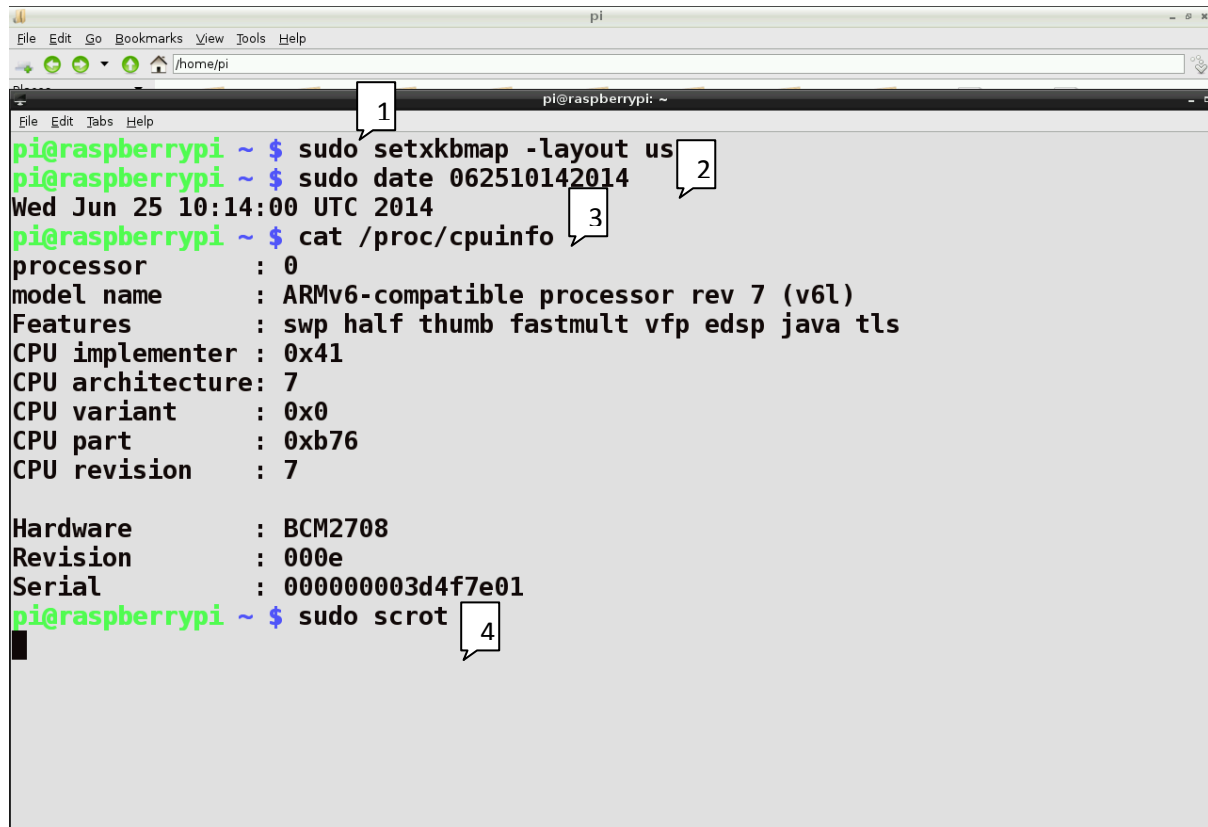
**Step8:** Type ‘`sudo setxkbmap –layout us`’ for setting keyboard layout as per US mode.

Figure 3.2(1).

**Step9:** Type ‘`sudo date mmddhhmmyyyy`’ for setting date and time. Figure 3.2(2).

**Step10:** Type ‘`cat /proc/cpuinfo`’ for getting cpu information. Figure 3.2(3).

**Step11:** Type ‘`sudo scrot`’ for getting screen shots. Figure 3.2(4).

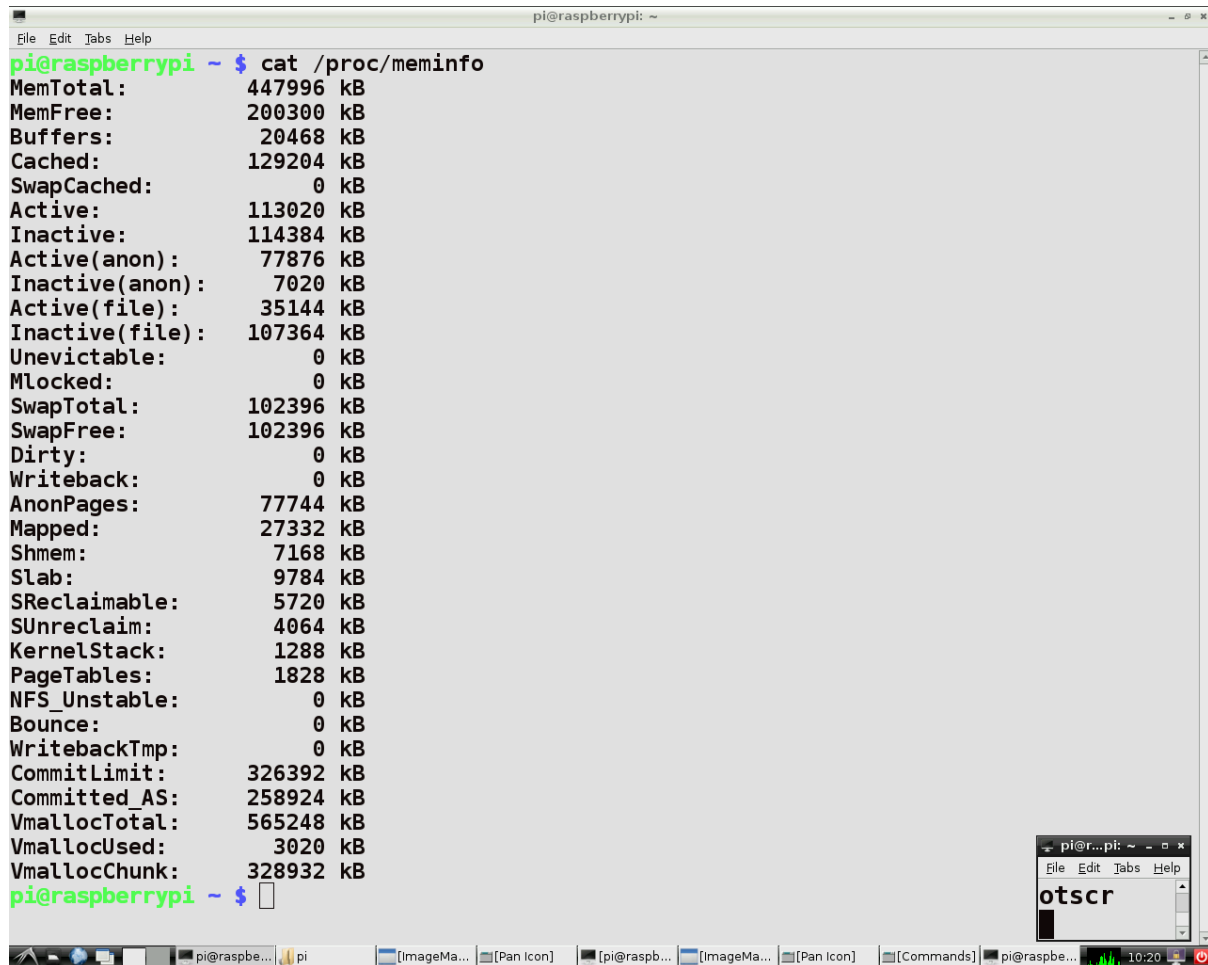


```
pi@raspberrypi ~ $ sudo setxkbmap -layout us
pi@raspberrypi ~ $ sudo date 062510142014
Wed Jun 25 10:14:00 UTC 2014
pi@raspberrypi ~ $ cat /proc/cpuinfo
processor       : 0
model name     : ARMv6-compatible processor rev 7 (v6l)
Features       : swp half thumb fastmult vfp edsp java tls
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xb76
CPU revision   : 7

Hardware       : BCM2708
Revision      : 000e
Serial        : 000000003d4f7e01
pi@raspberrypi ~ $ sudo scrot
```

**Figure 3.2 Linux commands**

**Step12:** Type 'cat /proc/meminfo' for getting cpu information. Figure 3.3.



```
pi@raspberrypi ~ $ cat /proc/meminfo
MemTotal:      447996 kB
MemFree:       200300 kB
Buffers:       20468 kB
Cached:        129204 kB
SwapCached:    0 kB
Active:        113020 kB
Inactive:      114384 kB
Active(anon):  77876 kB
Inactive(anon): 7020 kB
Active(file):  35144 kB
Inactive(file): 107364 kB
Unevictable:   0 kB
Mlocked:       0 kB
SwapTotal:     102396 kB
SwapFree:      102396 kB
Dirty:         0 kB
Writeback:     0 kB
AnonPages:     77744 kB
Mapped:        27332 kB
Shmem:         7168 kB
Slab:          9784 kB
SReclaimable: 5720 kB
SUnreclaim:   4064 kB
KernelStack:  1288 kB
PageTables:    1828 kB
NFS_Unstable:  0 kB
Bounce:        0 kB
WritebackTmp:  0 kB
CommitLimit:  326392 kB
Committed_AS: 258924 kB
VmallocTotal: 565248 kB
VmallocUsed:   3020 kB
VmallocChunk: 328932 kB
pi@raspberrypi ~ $
```

**Figure 3.3 Linux commands**

**3.4 Result:** Linux commands on raspberry pi have been executed and checked.

**3.5 Conclusion:-** Linux commands can be viewed in detail.

**3.6 References:-**

- 1.1. [http://en.wikipedia.org/wiki/Raspberry\\_Pi](http://en.wikipedia.org/wiki/Raspberry_Pi)
- 1.2. <https://www.eventbrite.com/e/introduction-to-raspberry-pi-november-tickets-8830824233>
- 1.3. Getting Started with Raspberry Pi by O'REILLY.
- 1.4 <http://www.raspberrypi.org/>
- 1.6 <http://elinux.org/RaspberryPiBoard/>
- 1.7 <http://thepihut.com/pages/the-raspberry-pi>
- 1.8 [http://en.wikipedia.org/wiki/Raspberry\\_Pi](http://en.wikipedia.org/wiki/Raspberry_Pi)

## EXPERIMENT-4

**Objectives:** To create a database & store the value in Raspberry Pi.

**Introduction:** When we connect any sensor to the ADC, it will give the readings which we can see by interfacing the ADC with raspberry pi. If the local storage of these readings required then it is necessary to create a local database in raspberry pi. So this experiment explains how to create a local database in raspberry pi and even how to store the readings in database.

### Software Requirements:

Python is main programming language on raspberry pi.

LEAFPAD: This is a tab inbuilt in raspberry pi. Programs are written in this this tab. Opening procedure of a leafpad is showing in fig-1.

Python libraries should be updated and it will also required internet connection.

### Hardware Requirements:

ADC: There are many ADC's available in market from Adafruit itself. I am using here 'mcp3008'. Because it is cheap and compatible.

POTENTIOMETER: You can take potentiometer of any value as you want.

CONNECTING WIRES: Here we need to connect ADC with raspberry pi through wires.

### Procedure:

**Step1:** First of all updates the packages using command:

```
$ sudo get-apt update
```

**Step2:** Download the database (for example MySQL). Open the LX-terminal and type the following command:

```
$ sudo apt-get install mysql-server python-mysqldb
```

This command will download the MySQL database in raspberry pi. During downloading , a password will have to enter, this is the root password of the database[4.1].

**Step3:** Open the database by entering following command in LX-terminal:

```
$ mysql -u root -p
```

Enter password:

```
Mysql>CREATE DATABASE abcd;
```

```
USE abcd;
```

Now for create an account, enter the following commands:

```
Mysql >CREATE USER 'pardeep'@'localhost' IDENTIFIED BY 'password';
```

```
Mysql>GRANT ALL PRIVILEGES ON abcd.*TO 'pardeep'@'localhost';
```

```
Mysql>FLUSH PRIVILEGES;
```

```
Mysql>quit
```

**Step4:** Now database is ready and we need to create a table as well.

Open the mysql database using command:

```
$ mysql -u pardeep -p
```

Password:

Use abcd;

Now create a table.

For example I am here creating a table which stores the potentiometer data with current date and time.

```
Mysql>CREATE TABLE igdtu(potdata FLOAT,time TIME,date DATE);
```

If you don't get any error, table has created.

You can check it using mysql command:

```
Mysql>show tables;
```

**Step5:** Open the leafpad and write the python program as given in 'Program' portion. Save it as name followed by '.py'. [4.2]

For example abc.py

**Step6:**Open the LX-terminal and run the file abc.py using command:

```
$ sudo python abc.py
```

Running of this file read the data from potentiometer and save it into a local database.

Now you can check your stored data into database.

### Program:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import spidev
import time
import datetime
import MySQLdb
import RPi.GPIO as GPIO

spi = spidev.SpiDev()
spi.open(0,0)

# read SPI data from MCP3008 chip, 8 possible adc's (0 thru 7)
def readadc(adcnum):
    if not 0 <= adcnum <=7:
        return -1
    r = spi.xfer2([1,(8+adcnum)<<4,0])
    adcout = ((r[1]&3) << 8) + r[2]
    return adcout
```

```
db = MySQLdb.connect("localhost","username","password","databasename")

curs = db.cursor()

while True:
    val=readadc(0)
    print val*0.0032160313
        curs.execute("INSERT INTO abc (date,time,potdata)
VALUES(now(),now(),'%s')" ,(readadc(0)))
    time.sleep(2)
```

### **References:**

1. <http://raspberrypiwebserver.com/sql-databases/using-mysql-on-a-raspberry-pi.html>
2. Getting Started with Raspberry Pi by O'REILLY.

## Experiment No-5

**Objectives:** To install Android on Raspberry Pi.

### Prerequisites:

#### Operating System:

Android has vast applications. It can be run on raspberry pi. Download the image file of android OS from the link as given below(with name CynogenMOD 7.2)[5.1]:

[http://androidpi.wikia.com/wiki/Android\\_Pi\\_Wiki](http://androidpi.wikia.com/wiki/Android_Pi_Wiki)

#### Win32 disk:

The downloading Procedure of Win32 Disk is same as experiment-2.

### Procedure:

**Step 1:** Extract the Image file.

**Step 2 :** Insert SD Card into pc/laptop (system on which OS and win32disk is downloaded). Make sure that SD card is empty and have atleast 4GB memory.

**Step 3 :** Open win32 disk. The UI view of it is shown in fig-2.1.

**Step 4 :** Browse the image file which has downloaded .

**Step 5:** Click on “write”.

**Step 6:** When writing is complete, remove the SD card and insert it into the raspberry pi.

**Step7:** ON the power of raspberry pi. Blinking Status of green LED shows that booting has started.[2]

### References:

5.1 [http://androidpi.wikia.com/wiki/Android\\_Pi\\_Wiki](http://androidpi.wikia.com/wiki/Android_Pi_Wiki)

5.2. Getting Started with Raspberry Pi by O'REILLY



## Experiment No.-6

**Objectives:** To set up RPi first time without using screen, mouse, keyboard.

### Prerequisites:

1. Raspberry pi board.
2. SDHC card with a fresh Debian Linux.
3. PC/Laptop.
4. Two Ethernet connections(both should be on local network).
5. Putty & Xming (as explains in previous experiments).
6. IP Scanner: This software is used to scan the IP addresses on a local network).Fig-1 shows the UI of IP scanner. Download this software from the link as given below:  
<http://download1.famatech.com/download/files/ipscan23.exe>

### Procedure:

**Step1:** Insert the SD card and Ethernet cable in raspberry pi and ON its power.[6.1]

**Step2:** Connect second Ethernet cable to PC/laptop, Open the IP scanner on PC/laptop and scan the IPs. There should be a Raspberry pi's IP if everything is fine. Note this IP.

**Step3:** Open putty, and write this IP in it. Forward X11 and click on 'open'.

Step4: Then Username and password is required.

By default the username is 'pi'.

Password is 'raspberry'.

**Step5:** Open Xming and write command 'startlxde' in LX-terminal of pi (i.e in putty window).

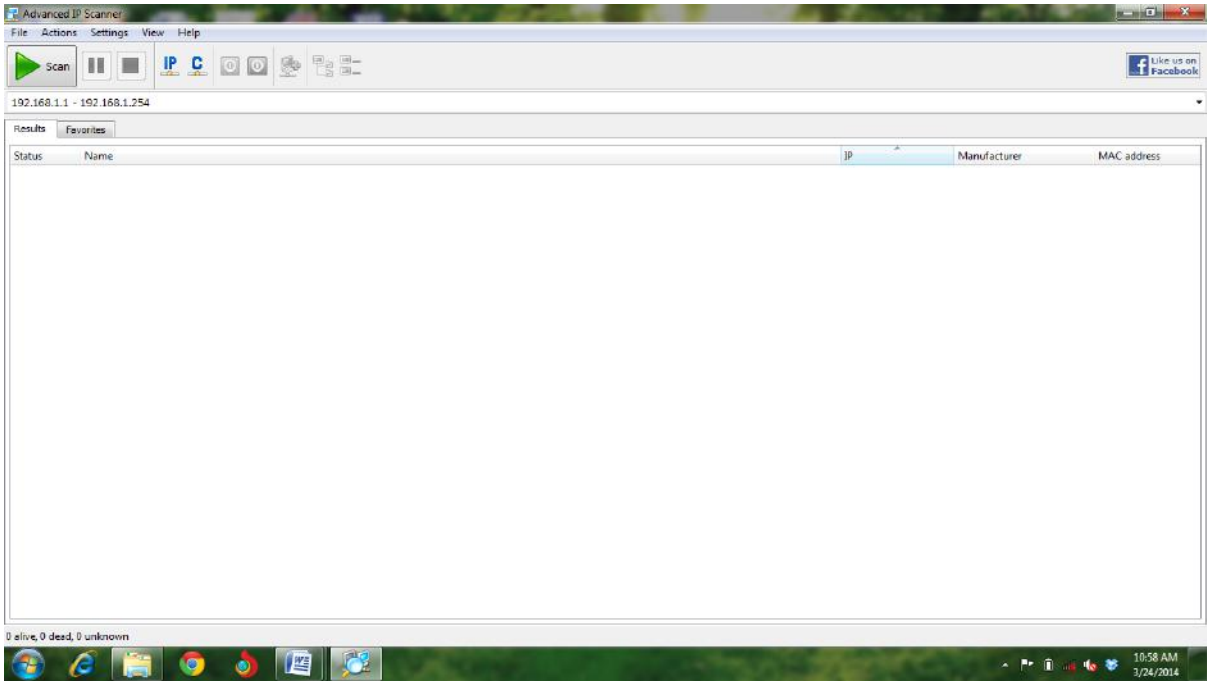
**Step6:** Then setup the RPi as discussed in experiment no.-2.

### Screenshots:

Fig-6.1: UI of IP scanner.

### References:

6.1. Getting Started with Raspberry Pi by O'REILLY



## Experiment No. 7

### Analog Input at GPIO of Raspberry Pi

**Objective:** To install OS (De-bian) into Adafruit's RaspberryPi and interface ADC at GPIOs of RaspberryPi, for measuring analog voltage.

**Software Requirement:** Win32 disk imager, Remote SSH client like Putty,

**Hardware Requirement:** Target board Raspberry Pi, 4GB SD Card, Ethernet cable, ADC MCP8008, Connecting wires .

#### Procedure:

1. Download image from link below for Raspbian i.e. Debian OS for Raspberry Pi.  
<http://www.raspberrypi.org/downloads>
2. Write this image into SD Card using Win32 disk imager. Download disk imager from link below.  
<http://sourceforge.net/projects/win32diskimager/>
3. Download remote SSH client Putty from link below.  
<http://putty.en.softonic.com/>
4. For desktop view on remote client download Xming from link below.  
<http://sourceforge.net/projects/xming/>
5. For detecting ip addresses of clients present in a network download Netscanner from link below.  
[http://download.cnet.com/Net-Scan/3000-2112\\_4-10580446.html](http://download.cnet.com/Net-Scan/3000-2112_4-10580446.html)
6. Put SD Card into Raspberry Pi's SD card slot, connect Ethernet cable and power on raspberrypi board.
7. Connect your PC with Ethernet cable (from same modem), and start net scanner.
8. Netscanner will scan your clients present into that particular modem and give ip address of raspberrypi board.
9. Run Xming
10. Run Putty with host address of raspberrypi as given by netscanner, with SSH->X11 enable.
11. Default login id for raspberrypi is pi and password raspberry. To view desktop on remote PC write on command line  
\$ startlxde
12. Raspberry default desktop will be open at your PC then.
13. Open LX terminal on raspberrypi's desktop.
14. Load the SPI device into raspberrypi using command prompt, type command  
\$ modprobe spidev
15. Check 'spidev' installation using command  
\$ lsmod

16. Install the 'git' package for enabling installation of other packages from Github using command

```
$ sudo apt-get install git-core
```

17. To compile 'py-spidev' package, install python-dev package using command

```
$ sudo apt-get install python-dev
```

18. Download and compile py-spidev' package, using commands

```
mkdir py-spidev
```

```
cd py-spidev
```

```
wget https://raw.githubusercontent.com/doceme/py-spidev/master/setup.py
```

```
wget https://raw.githubusercontent.com/doceme/py-spidev/master/spidev\_module.c
```

```
sudo python setup.py install
```

19. Interface ADC MCP3008 as per circuit diagram. Figure 6.1

20. Write python source code on leafpad editor. Save using .py extension.

21. Run desired .py extension file on LX terminal using command

```
$ sudo python filename.py
```

22. Values of voltages as input to ADC can be seen at LX terminal.

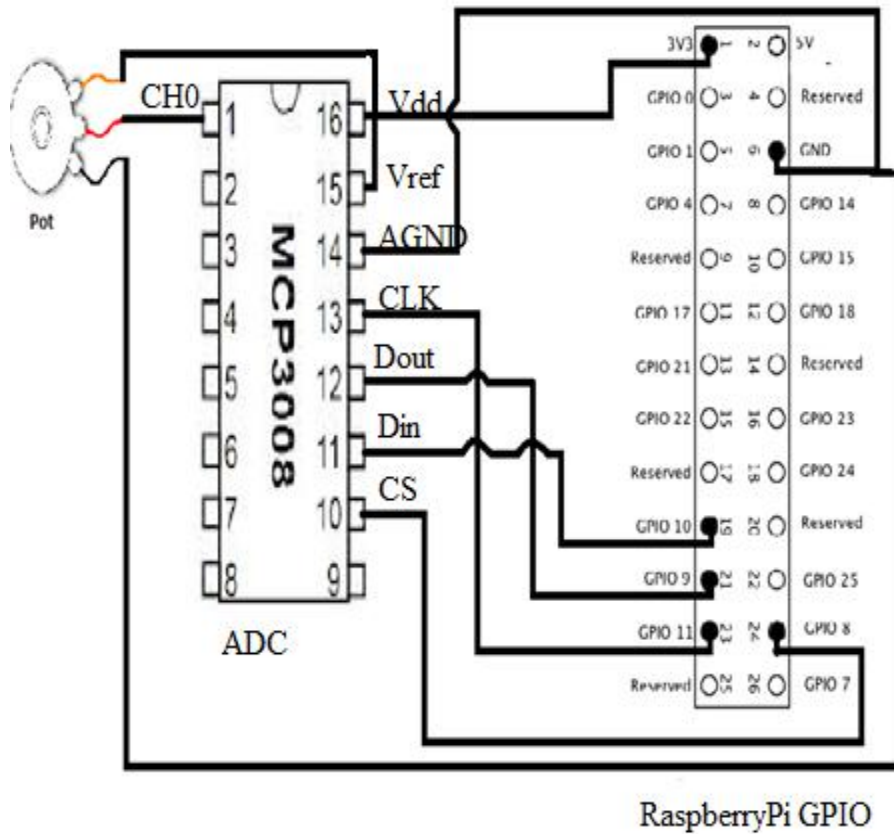


Figure 6.1 Interfacing connections for ADC with Raspberry Pi GPIO

**Source Code: Table 6**

<pre> ;Program written using python for raspberrypi board ;through SPI at GPIO port of raspberrypi ; Analog input given at CH0 of ADC ;Output seen at LX terminal  Import spidev Import time  Spi=spidev.SpiDev() Spi.open(0,0) Adc's(0 thru 7) </pre>	<pre> Def readadc(adcnum):     If not 0&lt;=adcnum &lt;=7:         Return -1     R=spi.xfer2([1,(8+adcnum)&lt;&lt;4,0])     Adcout=((R[1]&amp;3)&lt;&lt;8)+R[2]     Return adcout  While True:     Val=readadc(0)     Print Val*0.0032160313     Time.sleep(2) </pre>
--	---

**Result:** Output is observed at LX terminal of raspberrypi's desktop, as per variations in analog input voltage.

**Conclusion:-** Analog voltage inputs are taken by ADC MCP3008, converted into 10 bit digital value and communicated serially using SPI to raspberrypi board. MCP3008 having 8 input channels and hence 8 analog inputs can be used simultaneously and can be given serially to raspberrypi.

**Remark:-** All eight analog to digital channels can be used. Range for analog inputs data should be in the range of raspberry pi's digital input range i.e. 0.0 to 3.3 Volts only.

**Reference:**

1. Getting Started with Raspberry Pi by O'REILLY
2. Datasheet MCP 3008

## **Appendix A: List of Experiments** **2<sup>nd</sup> Semester M.TECH (MPC)**

<b>L</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>2</b>	<b>1</b>

**Paper Title:** MMC-524

**Paper Code:** SoC Architecture for Mobile and Pervasive Computing Lab Based on Raspberry Pi/Intel Atom Board

1. Study the Raspberry Pi (Broadcom 2835 SoC) (Broadcom 2835 SoC) architecture and its peripherals.
2. Customize the Raspberry Pi (Broadcom 2835 SoC) for Programming with Linux and Android OS and boot the same for application development.
3. Write a program to print Hello World/ Your Name using Raspberry Pi (Broadcom 2835 SoC) editor leaf pad.
4. Write a program to interface Buzzer and LED at different GPIO pins, write delay routine and generate 10 different patterns using Raspberry Pi (Broadcom 2835 SoC).
5. Write a program to interface Motion Sensor, IR sensor using Raspberry Pi (Broadcom 2835 SoC).
6. Write a program to interface ADC using Raspberry Pi (Broadcom 2835 SoC). Give complete schematic diagram with code.
7. Customize the environment for remotely accessing the Raspberry Pi (Broadcom 2835 SoC) through SSH server.
8. Introduction to Panda Board RevEA2 including TI SoC and OMAP4 SoC architecture and its peripherals.
9. Write a program to interface switches and LEDs. Use switches to select different pattern of LEDs for Panda Board.
10. Introduction to Intel Atom Board with E6XX series SoC architecture and its peripherals.
11. Customize the Intel Atom SoC for Programming with Windows, Linux and Android.
12. Write a program to interface LED at different GPIO pins, write a delay routine and generate 10 different patterns using Intel Atom SoC.
13. Write a program to interface Motion sensor, IR sensor using Intel Atom SoC.
14. Communication between two different Intel Atom SoC using LAN.
15. Customize and Control Intel Atom SoC with Android Phone.
16. Display the IP address of Intel Atom SoC using external LCD display.
17. Introduction to Snapdragon SoC architecture and its peripherals.