THE NEW COLLEGE

(AN AUTONOUMOUS INSTITUTION AFFILIATED TO THE UNIVERSITY OF MADRAS
& ACCREDITED BY NAAC WITH 'A++' GRADE IN THE 4TH CYCLE)
SPONSORED BY : THE MUSLIM EDUCATIONAL ASSOCIATION OF SOUTHERN INDIA
(MEASI)

YA ALLAH
INCREASE MY KNOWLEDGE

DEPARTMENT OF COMPUTER APPLICATIONS

(B.C.A)

E-Content

Subject	: INTERNET OF THINGS
Class	: III B.C.A.,
Unit	: IV

UNIT-4: Basic Building Blocks of an IoT Device, Raspberry Pi, Programming Raspberry Pi using Python, Basics of IoT Protocols: HTTP, UPnP, MQTT, CoAP and XMPP

4. IoT Device

A "Thing" in Internet of Things (IoT) can be any object that has a unique identifier and which can send/receive data (including user data) over a network (e.g., smart phone, smart TV, computer, refrigerator, car, etc.).

IoT devices are connected to the Internet and send information about them or about their surroundings (e.g. information sensed by the connected sensors) over a network (to other devices or servers/storage) or allow actuation upon the physical entities/environment around them remotely.

IoT Device Examples

- A home automation device that allows remotely monitoring the status of appliances and controlling the appliances.
- An industrial machine which sends information a bouts its operation and health monitoring data to a server.
- A car which sends information about its location to a cloud-based service.
- A wireless-enabled wearable device that measures data about a person such as the number of steps walked and sends the data to a cloud-based service.

4.1 Basic building blocks of an IoT Device

1. Sensing: Sensors can be either on-board the IoT device or attached to the device.
2. Actuation: IoT devices can have various types of actuators attached that allow taking actions upon the physical entities in the vicinity of the device.
3. Communication: Communication modules are responsible for sending collected data to other devices or cloud-based servers/storage and receiving data from other devices and commands from remote applications.
4. Analysis & Processing: Analysis and processing modules are responsible for making sense of the collected data.

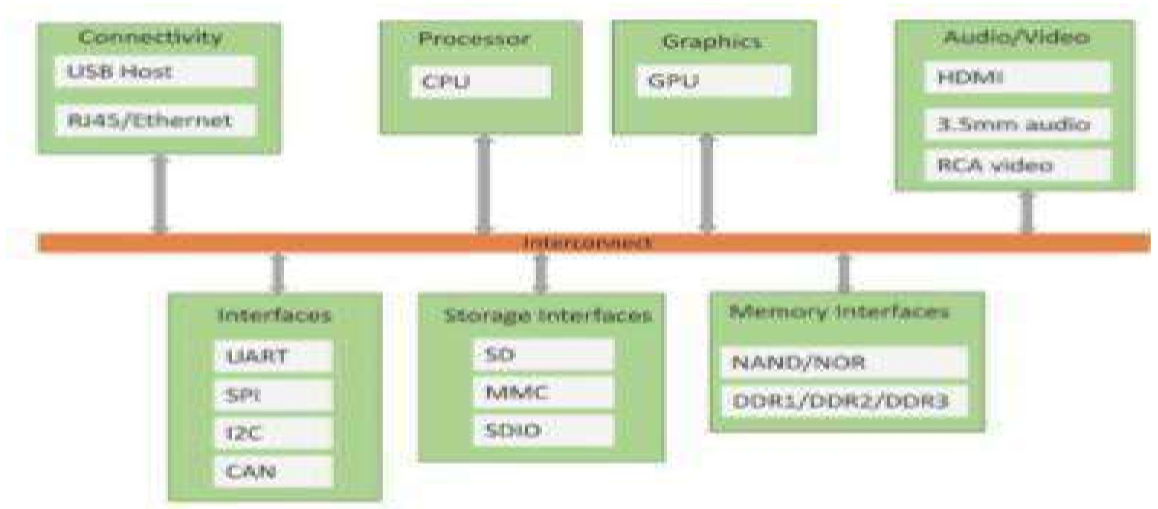


Fig 4.1 Block diagram of an IOT device

4.2 Exemplary Device: Raspberry Pi

Raspberry Pi is a low-cost mini-computer with the physical size of a credit card. Raspberry Pi runs various flavors of Linux and can perform almost all tasks that a normal desktop computer can do. Raspberry Pi also allows interfacing sensors and actuators through the general purpose I/O pins. Since Raspberry Pi runs Linux operating system, it supports Python "out of the box". Raspberry Pi is a low-cost mini-computer with the physical size of a credit card. Raspberry Pi runs various flavors of Linux and can perform almost all tasks that a normal desktop computer can do. Raspberry Pi also allows interfacing sensors and actuators through the general purpose I/O pins. Since Raspberry Pi runs Linux operating system, it supports Python "out of the box".

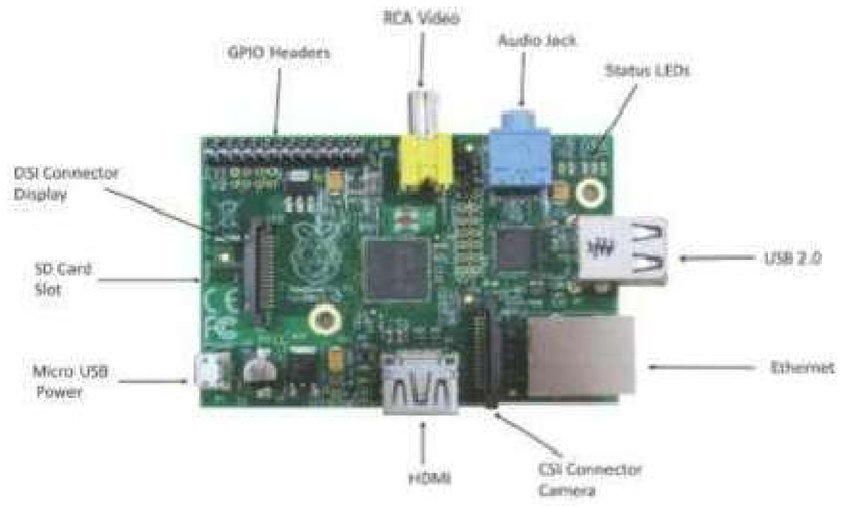


Fig 4.2 Raspberry Pi Board

The following are the various components in a Raspberry Pi Board

Processor & RAM: Raspberry Pi is based on an ARM processor. The latest version of Raspberry Pi (Model B, Revision 2) comes with 700 MHz Low Power ARM1176JZ-F processor and 512 MB SDRAM.

USB Ports: Raspberry Pi comes with two USB 2.0 ports. The USB ports on Raspberry Pi can provide a current upto 100mA. For connecting devices that draw current more than 100mA, an external USB powered hub is required.

Ethernet Ports: Raspberry Pi comes with a standard RJ45 Ethernet port. One can connect an Ethernet cable or a USB Wi-Fi adapter to provide Internet connectivity.

HDMI Output: The HDMI port on Raspberry Pi provides both video and audio output. The Raspberry Pi can be connected to a monitor using an HDMI cable. For monitors that have a DVI port but no HDMI port, and can use an HDMI to DVI adapter/cable.

Composite Video Output: Raspberry Pi comes with a composite video output with an RCA jack that supports both PAL and NTSC video output. The RCA jack can be used to connect old televisions that have an RCA input only.

Audio Output: Raspberry Pi has a 3.5mm audio output jack. This audio jack is used for providing audio output to old televisions along with the RCA jack for video. The audio quality from this jack is inferior to the HDMI output.

GPIO Pins: Raspberry Pi comes with a number of general purpose input/output pins. There are four types of pins on Raspberry Pi - true GPIO pins, 12C interface pins, SPI interface pins and serial Rx and Tx pins.

Display Serial Interface (DSI): The DSI interface can be used to connect an LCD panel to Raspberry Pi.

Camera Serial Interface (CSI): The CSI interface can be used to connect a camera module to Raspberry Pi.
Status LEDs: Raspberry Pi has five status LEDs.

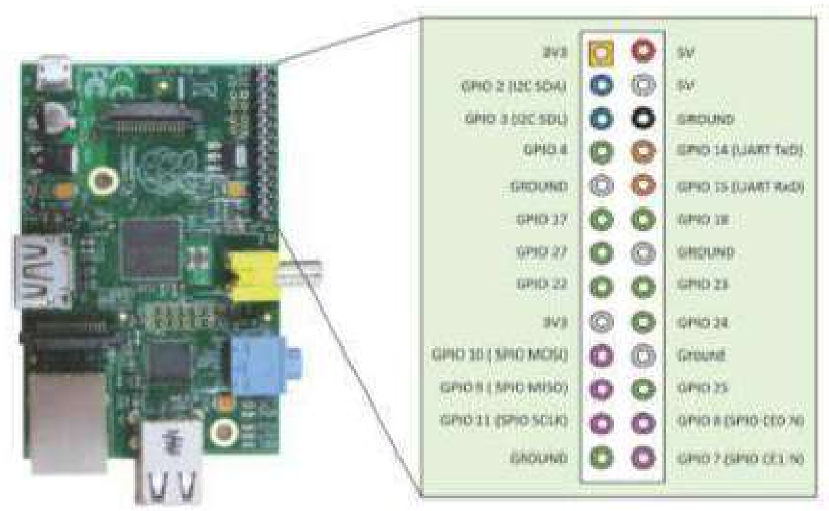


Fig 4.3 Raspberry Pi GPIO Pins

SD Card Slot: Raspberry Pi does not have a built in operating system and storage. An SD card can be plugged-in loaded with a Linux image to the SD card slot. Appendix-A provides instructions on setting up New Out-of-the-Box Software (NOOBS) on Raspberry Pi. It requires at least an 8GB card for setting up NOOBS.

Power Input: Raspberry Pi has a micro-USB connector for power input.

Linux on Raspberry Pi

1. Raspbian: Raspbian Linux is a Debian Wheezy port optimized for Raspberry Pi.
2. Arch: Arch is an Arch Linux port for AMD devices.
3. Pidora: Pidora Linux is a Fedora Linux optimized for Raspberry Pi.
4. RaspBMC: RaspBMC is an XBMC media-center distribution for Raspberry Pi.
5. OpenELEC: OpenELEC is a fast and user-friendly XBMC media-center distribution.
6. RISC OS: RISC OS is a very fast and compact operating system.

Fig 4.4 shows the Raspberry Linux Desktop. Fig 4.5 shows the default File Explorer on Raspberry Pi. Fig 4.6 shows the default Console on Raspberry Pi. To configure Raspberry Pi, the raspi-config tool is used which can be launched from the command-line as ($raspi-config) as shown in Fig 4.8. Using the Configuration tool, one can expand partition to fill SD Card, set keyboard layout, change password, set locale and time zone, change memory split, enable or disable SSH server and change boot behaviour, It is recommended to expand the root file-system to use entire space on SD card.

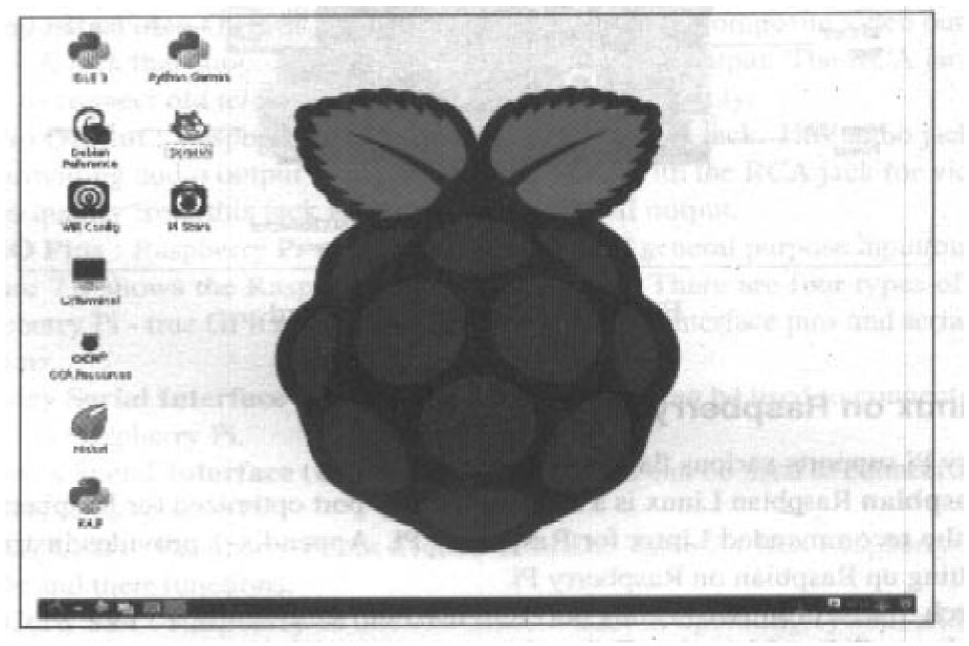


Fig 4.4 Raspberry Linux Desktop

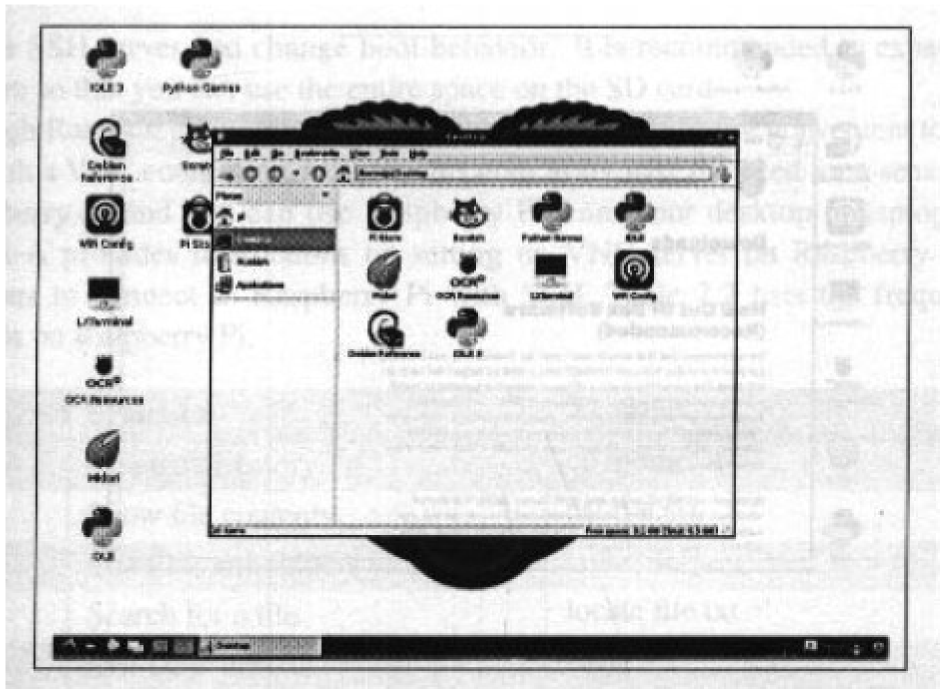
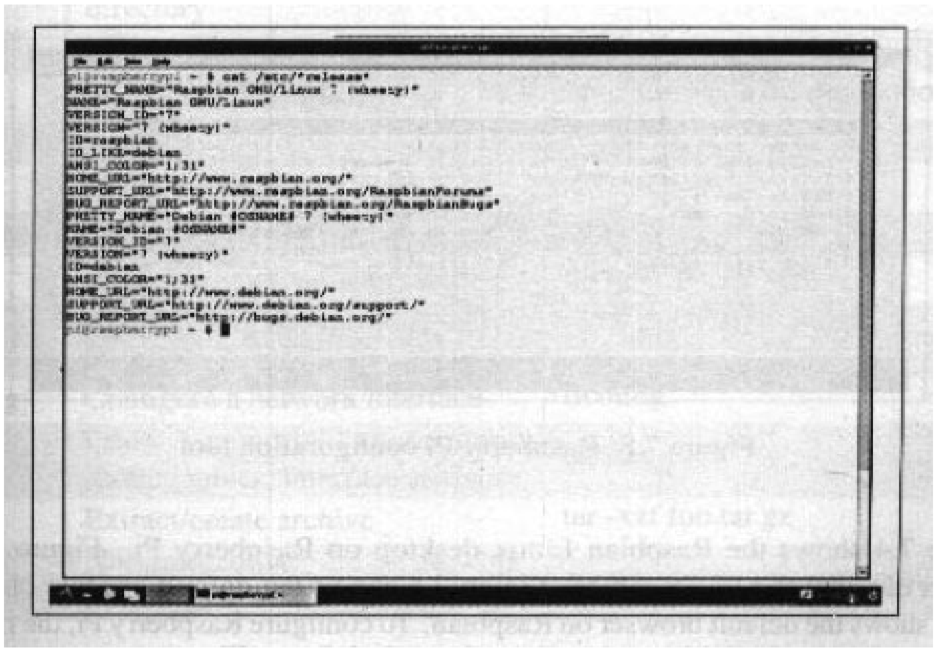


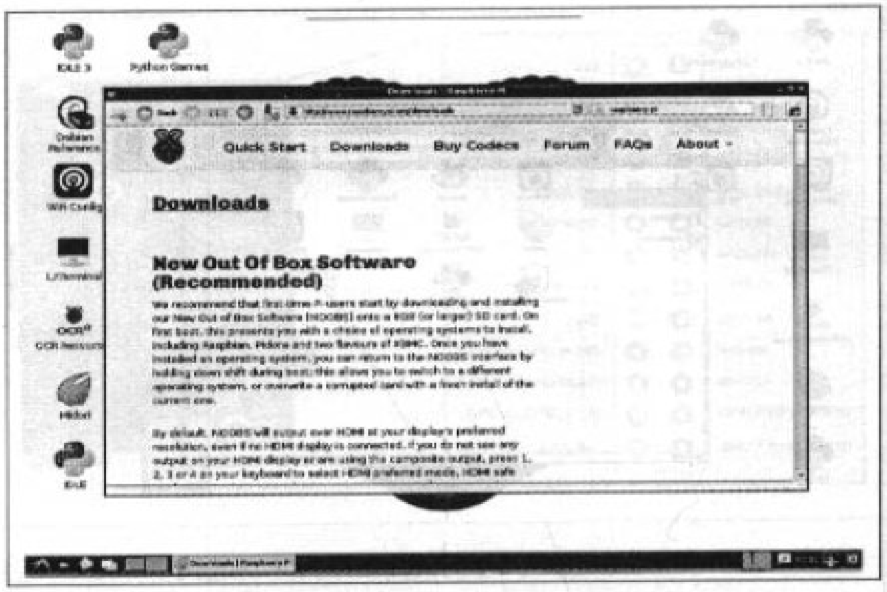
Fig 4.5 File Explorer on Raspberry Pi

Raspberry pi comes with an HDMI output, it s more convenient to access the device with a VNC connection or SSH. This does with the need for a separate display for Raspberry pi and can use Raspberry pi from desktop or laptop computer. The following are frequently used commands on Raspberry Pi.



```
pi@raspberrypi ~ $ cat /etc/*release*
PRETTY_NAME="Raspbian GNU/Linux 7 (wheezy)"
NAME="Raspbian GNU/Linux"
VERSION_ID="7"
VERSION="7 (wheezy)"
ID=raspbian
ID_LIKE=debian
ANSI_COLOR="1;31"
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
PRETTY_NAME="Debian #OSNAME# 7 (wheezy)"
NAME="Debian #OSNAME#"
VERSION_ID="7"
VERSION="7 (wheezy)"
ID=debian
ANSI_COLOR="1;31"
HOME_URL="http://www.debian.org/"
SUPPORT_URL="http://www.debian.org/support/"
BUG_REPORT_URL="http://bugs.debian.org/"
pi@raspberrypi ~ $
```

Fig 4.6 Console on Raspberry Pi



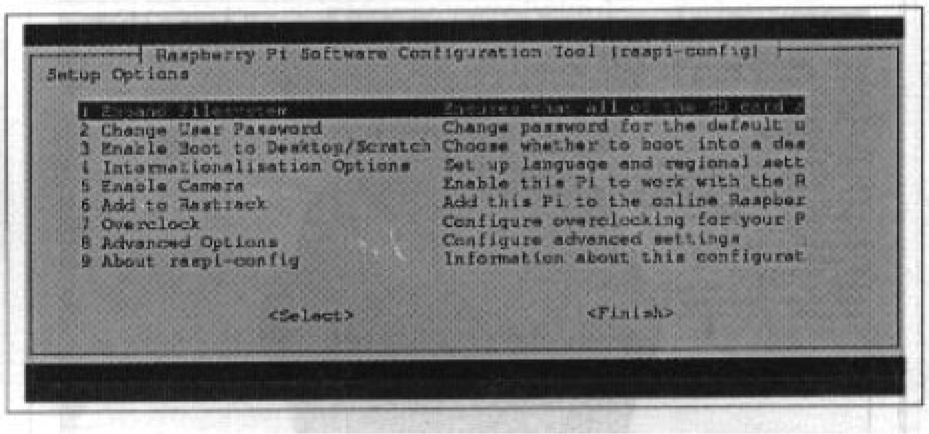
Downloads

New Out Of Box Software (Recommended)

We recommend that first-time Pi users start by downloading and installing our New Out of Box Software (NOOBS) onto a 8GB (or larger) SD card. On first boot, this presents you with a choice of operating systems to install, including Raspbian, Pidora and two flavours of XBMC. Once you have installed an operating system, you can return to the NOOBS interface by holding down shift during boot; this allows you to switch to a different operating system, or overwrite a corrupted card with a fresh install of the current one.

By default, NOOBS will output over HDMI at your display's preferred resolution, even if no HDMI display is connected. If you do not see any output on your HDMI display or are using the composite output, press 1, 2, 3 or 4 on your keyboard to select HDMI preferred mode, HDMI safe

Fig 4.7 Browser on Raspberry Pi

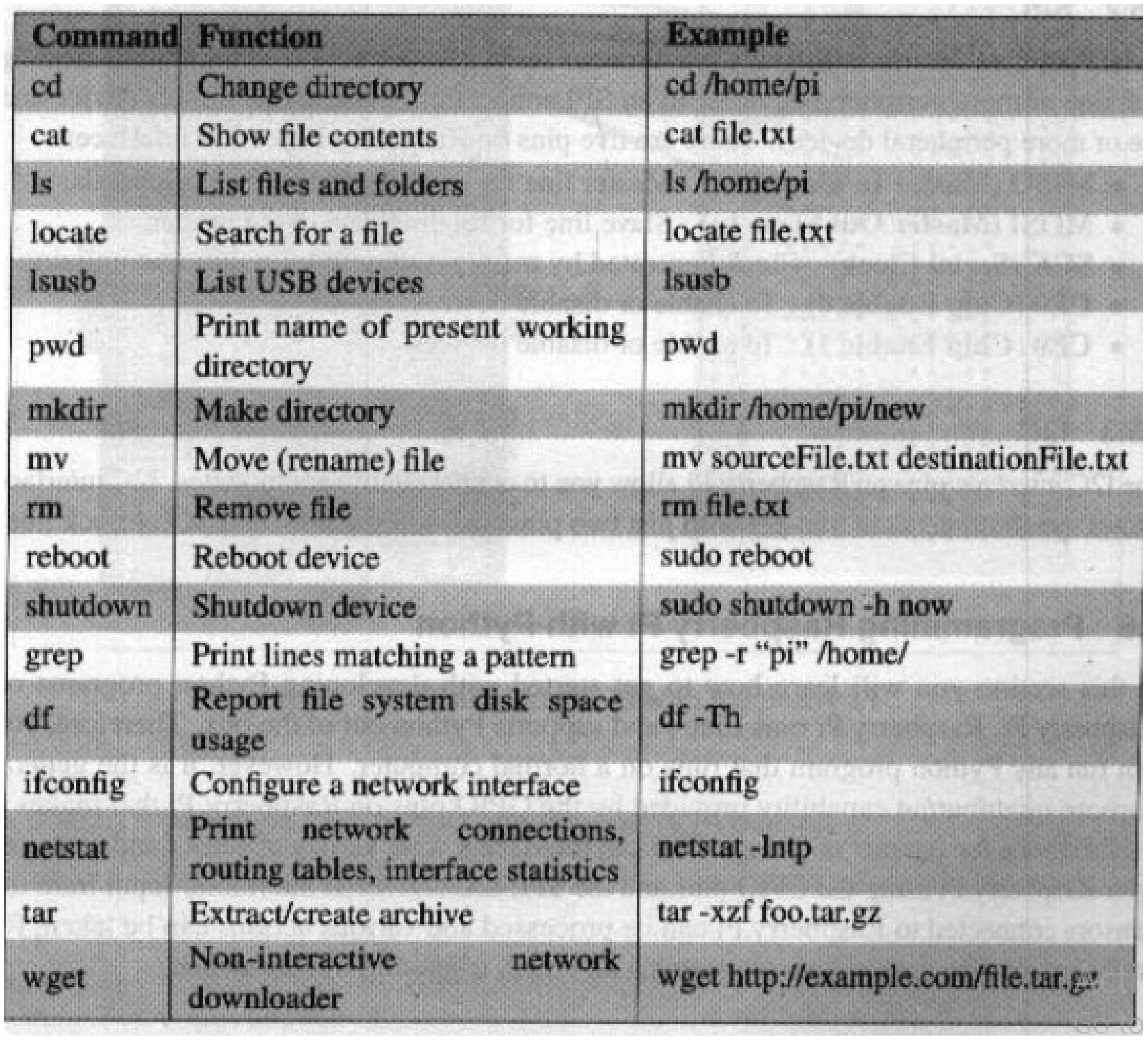


```
Raspberry Pi Software Configuration Tool (raspi-config)
Setup Options

   1 Expand Filesystem                  Ensures that all of the SD card s
   2 Change User Password               Change password for the default u
   3 Enable Boot to Desktop/Scratch     Choose whether to boot into a des
   4 Internationalisation Options       Set up language and regional sett
   5 Enable Camera                      Enable this Pi to work with the R
   6 Add to Rastrack                    Add this Pi to the online Raspber
   7 Overclock                          Configure overclocking for your P
   8 Advanced Options                   Configure advanced settings
   9 About raspi-config                 Information about this configurat

              <Select>                          <Finish>
```

Fig 4.8 Raspberry Pi Configuration tool



Command	Function	Example
cd	Change directory	cd /home/pi
cat	Show file contents	cat file.txt
ls	List files and folders	ls /home/pi
locate	Search for a file	locate file.txt
lsusb	List USB devices	lsusb
pwd	Print name of present working directory	pwd
mkdir	Make directory	mkdir /home/pi/new
mv	Move (rename) file	mv sourceFile.txt destinationFile.txt
rm	Remove file	rm file.txt
reboot	Reboot device	sudo reboot
shutdown	Shutdown device	sudo shutdown -h now
grep	Print lines matching a pattern	grep -r "pi" /home/
df	Report file system disk space usage	df -Th
ifconfig	Configure a network interface	ifconfig
netstat	Print network connections, routing tables, interface statistics	netstat -lntp
tar	Extract/create archive	tar -xzf foo.tar.gz
wget	Non-interactive network downloader	wget http://example.com/file.tar.gz

4.3 Raspberry Pi Interfaces

Raspberry Pi has serial, SPI and I2C interfaces for data transfer as shown in Fig: 7.3.

4.3.1 Serial

The serial interface on Raspberry Pi has receive (Rx) and transmit (Tx) pins for communication with serial peripherals.

4.3.2 SPI

Serial Peripheral Interface (SPI) is a synchronous serial data protocol used for communicating with one or more peripheral devices. In an SPI connection, there is one master device and one or more peripheral devices. There are five pins on Raspberry Pi for SPI interface

MISO (Master In Slave Out): Master line for sending data to the peripherals.
MOSI (Master Out Slave In): Slave line for sending data to the master.
SCK (Serial Clock): Clock generated by master to synchronize data transmission .
CEO (Chip Enable 0): To enable or disable devices.
CEO (Chip Enable 1): To enable or disable devices.

4.3.3 I2C

The I2C interface pins on Raspberry Pi allow connecting hardware modules. I2C interface allows synchronous data transfer with just two pins - SDA (data line) and SCL (clock line).

4.4 Programming Raspberry Pi with Python

In this section you will learn how to get started with developing Python programs on Raspberry Pi. Raspberry Pi runs Linux and supports Python out of the box. Therefore, you can run any Python program that runs on a normal computer. However, it is the general purpose input/output capability provided by the GPIO pins on Raspberry Pi that makes it useful device for Internet of Things. You can interface a wide variety of sensor and actuators with Raspberry Pi using the GPIO pins and the SPI, I2C and serial interfaces. Input from the sensors connected to Raspberry Pi can be processed and various actions can be taken, for instance, sending data to a server, sending an email, triggering a relay switch.

4.4.1 Controlling LED with Raspberry Pi

Let us start with a basic example of controlling an LED from Raspberry Pi. Figure 4.9 shows the schematic diagram of connecting an LED to Raspberry Pi.

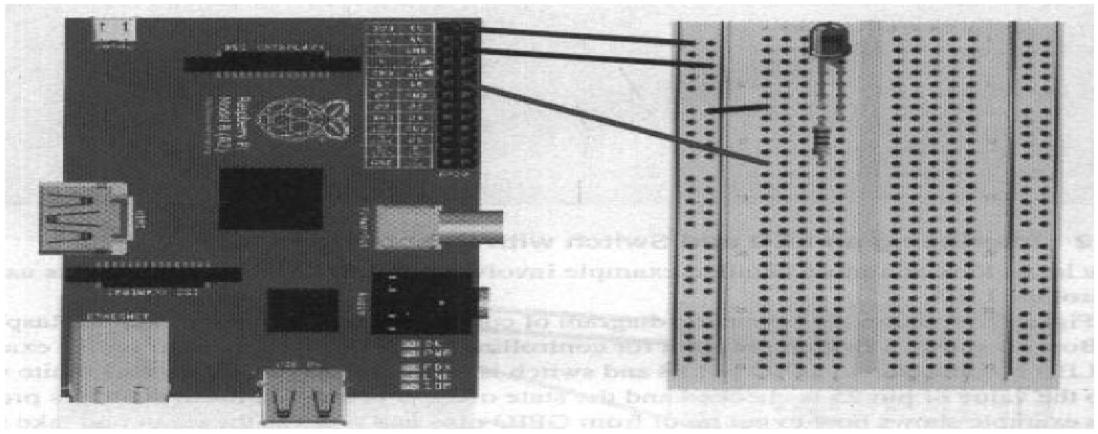


Fig 4.9: Controlling LED with Raspberry Pi

Box 7.1 shows how to turn the LED to any other GPIO pin as well.

Box 7.1: Switching LED on/off from Raspberry Pi console

```
$echo 18> /sys/class/gpio/export
$cd /sys/class/gpio/gpio18
#Set pin 18 direction to out
Secho out > direction.
#Turn LED on
Secho 1 > value.
#Turn LED off
Secho 0 > value.
```

Box 7.2 shows a Python Program for blinking an LED connected to Raspberry Pi every second. The program shows the RPi. GPIO module to control the GPIO on Raspberry Pi. In this program, set 18 pin direction to output and then write True/False alternatively after a delivery of one second.

Box 7.2: Python program for blinking LED

```
import RPi.GPIO as GPIO import time.
GPIO.setmode (GPIO.BCM) GPIO.setup(18, GPIO.OUT)
while True:
GPIO.output (18, True)
time.sleep(1)
GPIO.output (18, False):
time.sleep (1)
```

4.4.2 Interfacing an LED and Switch with Raspberry Pi

Now let us look at a more detailed example involving an LED and a switch that is used to control the LED.

Figure 4.10 shows the schematic diagram of connecting an LED and switch to Raspberry Pi. Box 7.3 shows a Python program for controlling an LED with a switch. In this example the LED is connected to GPIO pin 18 and switch is connected to pin

25. In the infinite while loop the value of pin 25 is checked and the state of LED is toggled if the switch is pressed. This example shows how to get input from GPIO pins

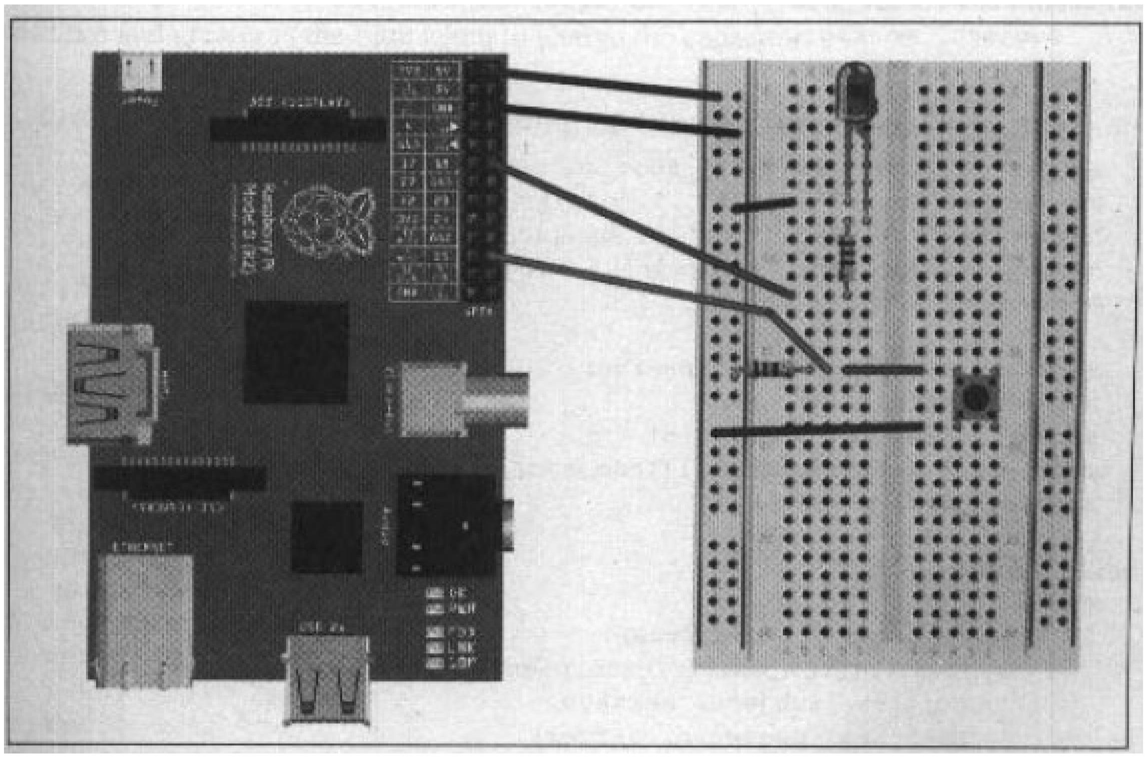


Fig 4.10:Interfacing an LED and Switch with Raspberry Pi

and process the input and take some action. The action in this example is toggling the state of an LED. Let us look at another example, in which the action is an email alert.

Box 7.3: Python program for controlling an LED with a switch

```
from time import sleeP import RPi.GPIO asGPIO GPIO.setmode(GPIO.BCM)
#Switch Pin GPIO.setup(25,GPIO.IN) #LEDPin
GPIO.setup(18,GPIO.OUT)
state=false
deftoggleLED(pin):
state = not state GPIO.output(pin,state)
whileTrue:
try:
if (GPIO.input(25) ==True):
toggleLED(pin)
```

```
sleep(.01)
exceptKeyboardInterrupt:
exit()
```

Box 7.4 shows a Python program for sending an email on switch press. Note that the structure of this program is similar to the program in Box 7.3. This program uses the Python SMTP library for sending an email when the switch connected to Raspberry Pi is pressed.

Basics of IoT Protocols

HTTP

Hypertext transfer protocol is the application layer protocol that forms the foundation of World Wide Web. HTTP includes commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS etc. The protocol follows a request-response model where are client sends request to server using the HTTP commands. HTTP is a stateless protocol and each HTTP request is independent of the other requests. HTTP client can be a browser or an application running on the client (e.g. an application running on an IoT device, a mobile application or other software).

UPnP

UPnP (Universal Plug and Play) is an extension of the Plug and Play permissions and protocols. It uses standard protocols for data transfer, such as TCP/IP, HTTP, and DHCP.

The universal approach allows UPnP to establish both wired (e.g., Ethernet, Firewire) and wireless (e.g., WiFi, Bluetooth) connections without requiring any additional drivers. This enables any device compatible with UPnP to take part in data transfer, regardless of its OS, programming language, product type, or manufacturer.

Nevertheless, despite all the perks of this protocol, it comes with certain security risks. The problem is, Universal Plug and Play doesn’t require any authentication as it was built with the assumption that every device in the network is reliable and friendly by default. So, if a device gets infected with malware, the whole system will be compromised, and data packages will get intercepted.

MQTT

Message Queue Telemetry Transport it is a lightweight message protocol based on public -subscribe model. MQTT uses client server architecture. The clients such as an IoT device connect to the server also called the MQTT broker and publishers message to topic on the server. The broker forward the message to the clients subscribed to topic. MQTT is well suited for constrained environments.

MQTT (Message Queue Telemetry Transport) is an open protocol for asynchronous data exchange between physically scattered devices that works at the application layer. The protocol uses port 1883 by default (or port 8883 if an SSL connection is established). It's not suitable for transferring voice or video data.

MQTT is used when there's a need to minimize the data packet size and when there are restrictions on channel bandwidth.

XMPP

Extensible Message and Presence Protocol for real time communication and streaming XML data between network entities. Support client-server and server-server communication.