-

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)

INTERNET OF THINGS LAB

MASTER MANUAL

SEMESTER: IV YEAR-I SEM

Regulation: R20

A.Y: 2023-24

CMR ENGINEERING COLLEGE

(Approved By AICTE-New Delhi,Affiliated to JNTUH)

KANDLAKOYA (V), MEDCHAL (M), R.R.DIST.

OBJECTIVES:

1.To assess the vision and introduction of IoT.

2. To Understand IoT Market perspective.

3. To Implement Data and Knowledge Management and use of Devices in IoT Technology.

4. To Understand State of the Art - IoT Architecture.

5. To classify Real World IoT Design Constraints, Industrial Automation in IoT.

OUTCOMES:

At the end of the course the student will be able to

1. Identify and differentiate various components used in IoT Architecture.

2. Write & execute programs in python programming language.

3. Use python programming language to interface with raspberry.

4. Demonstrate the various real time applications using Raspberry Pi.

SI.NO	TOPIC	PAGE NO
1	Introduction About IOT	
2	Introduction About Python	
3	Introduction About Raspberry Pi 3	
4	**Week 1:** Start Raspberry Pi and try various Linux commands in command terminal window: *ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less, ps, sudo, cron, chown, chgrp, ping etc.*	

5	**Week 2:** a)Run some python programs on Pi like: b)Read your name and print Hello message with name c)Read two numbers and print their sum, difference, product and division. d)Word and character count of a given string e)Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input f) Print a name 'n' times, where name and n are read from standard input, using for and while loops. g)Handle Divided by Zero Exception. h)Print current time for 10 times with an interval of 10 seconds. i) Read a file line by line and print the word count of each line.	
6	**Week 3:** Light an LED through Python program	
7	**Week 4:** Get input from two switches and switch on corresponding LEDs	
8	**Week 5:** Flash an LED at a given on time and off time cycle, where the two times are taken from a file.	
9	**Week 6:** Flash an LED based on cron output (acts as an alarm)	
10	**Week 7:** Switch on a relay at a given time using cron, where the relay's contact terminals are connected to a load.	
11	**Week 8:** Get the status of a bulb at a remote place (on the LAN) through web.	

1.INTRODUCTION TO IOT

DEFINITION OF IOT:

The **Internet of Things** (**IoT**) is a network comprised of physical objects capable of gathering and sharing electronic information

(OR)

The **internet of things** (**IoT**) is a computing concept that describes the idea of everyday physical objects being connected to the internet and being able to identify themselves to other devices

HISTORY OF IOT:

The term Internet of Things is 16 years old. But the actual idea of connected devices had been around longer, at least since the 70s. Back then, the idea was often called "embedded internet" or "pervasive computing".

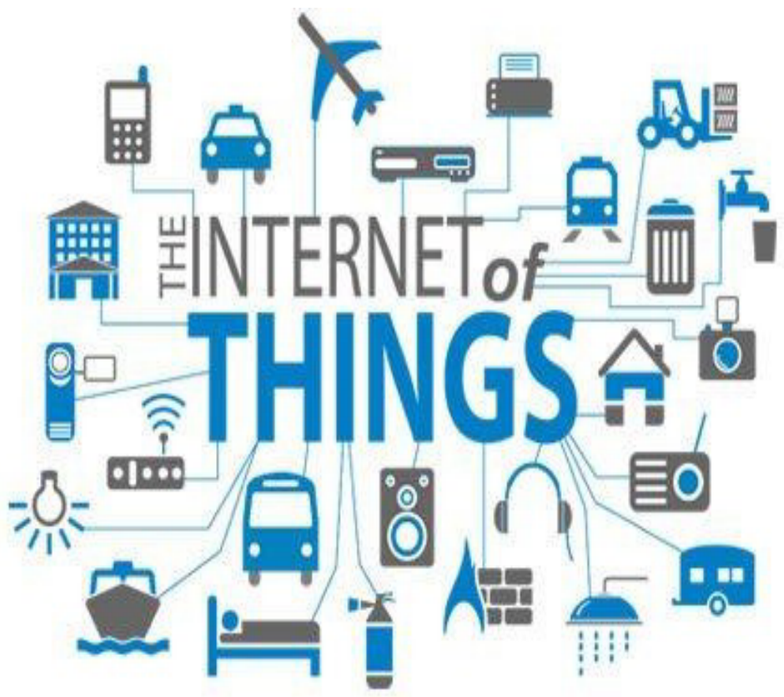


Figure1. Internet of things

Figure2. Kevin Ashton, Inventor of the Internet of Things

The term " Internet of Things" was coined by Kevin Ashton in 1999 during his work at Procter & Gamble. Ashton who was working in supply chain optimization, wanted to attract senior management's attention to a new exciting technology called RFID. Because the internet was the hottest new trend in 1999 and because it somehow made sense, he called his presentation " Internet of Things". Even though Kevin grabbed the interest of some P&G executives, the term Internet of Things did not get widespread attention for the next 10 years.

IOT ARCHITECTURE:

IOT architecture consists of different layers of technologies supporting IOT. It serves to illustrate how various technologies relate to each other and to communicate the scalability, modularity and configuration of IOT deployments in different scenarios

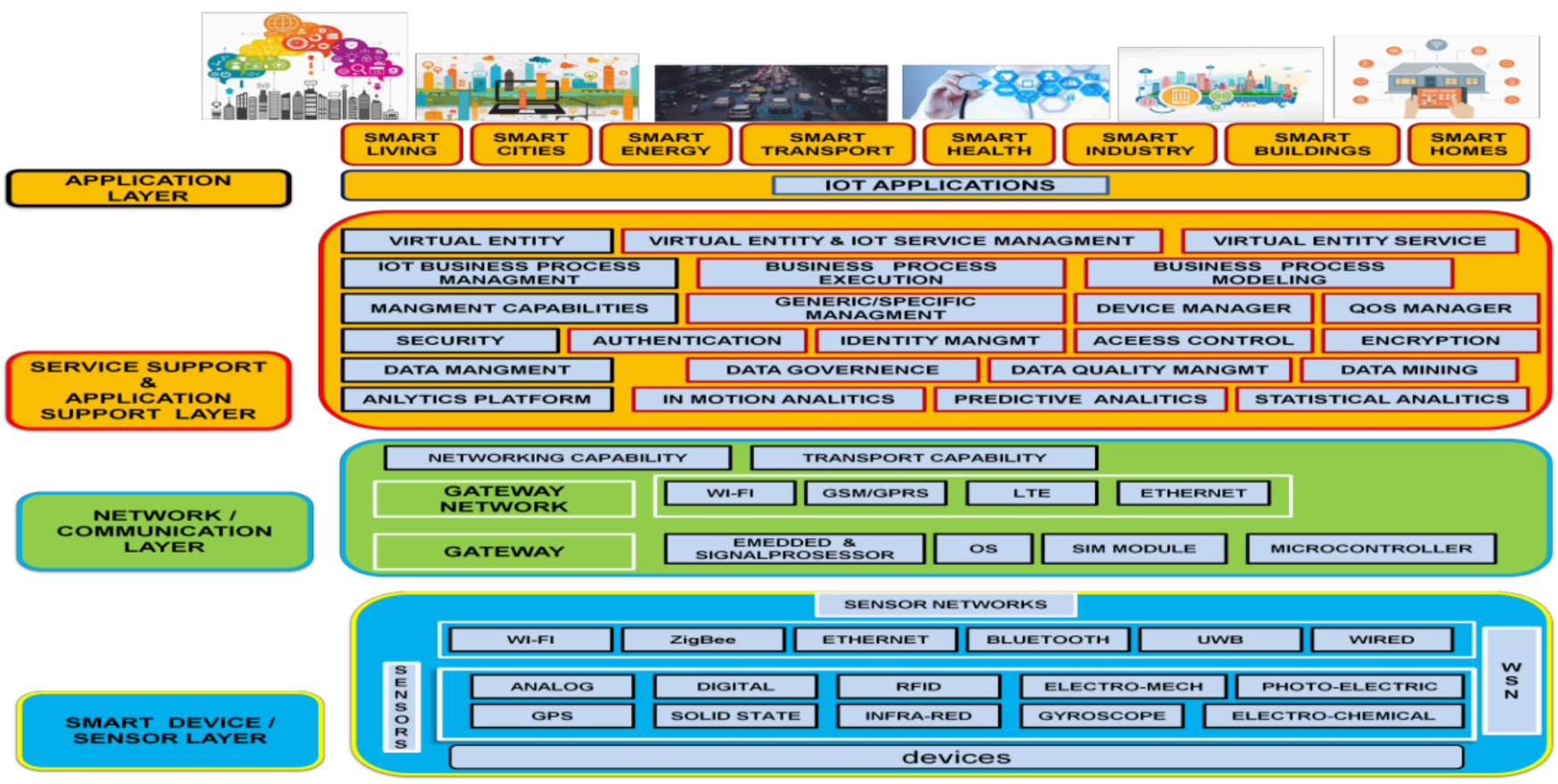


Figure3.IoTArchitecture

ENABLING TECHNOLOGIES FOR IOT:

Enabling technologies for the Internet of Things are considered in and can be grouped into three categories:

1. Technologies that enable " things" to acquire contextual information,
2. Technologies that enable " things" to process contextual information, and
3. Technologies to improve security and privacy.

FUNDAMENTAL CHARACTERISTICS OF IOT:

There are 7 crucial Internet of Things characteristics:

Intelligence

Connectivity

Dynamic Nature

Enormous scale

Sensing

Heterogeneity

Security

BENEFITS OF IOT:

1. **Data Management:** Using devices we collect lots of raw data. These data are filtered and managed by the IOT systems to produce an informative data. IOT reduces the overall complexity of data management.

2. **Good Decision Making:** More informative data will help you to take right decision. Calculated charts and reports generated by the IOT systems will help you to take good decisions for your businesses.

3. **Tracking and Safety:** Using IOT system we can track and monitor all the objects functionalities and their outputs. For safety of life we can track the quality and expiration date of the products before its consumption.

4. **Save Time & Money**: IOT system perform task rapidly and doesn't requires human which makes these devices more cost effective and fast

INTRODUCTION TO BASIC COMPONENTS:

1. **Jumper wires** – Jumper wires/cables or simply jumpers are available in various colours with male-male, female-female or male-female connectors at the two ends, and are used for interconnecting components on the breadboard or Arduino board.

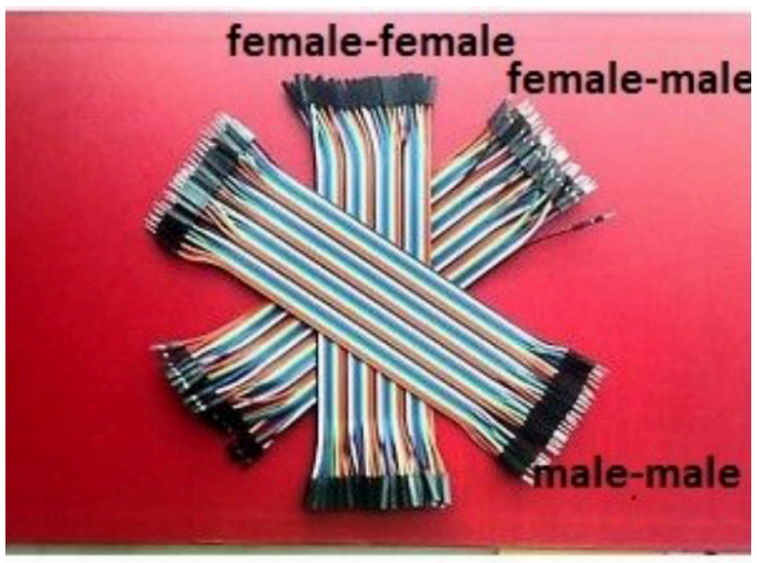


Figure 4. Types of jumper wires

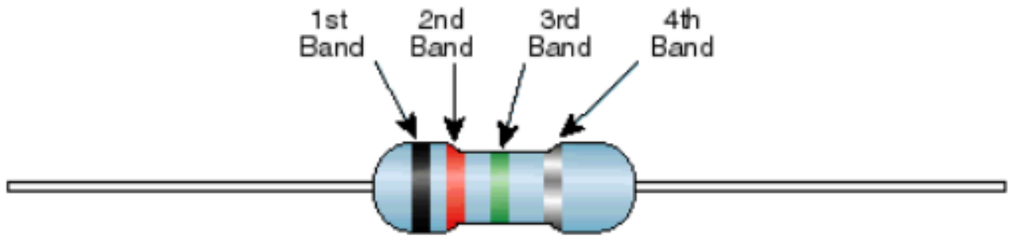
2. LEDs (Light Emitting Diodes),

LEDs consume very low power and so are long-lasting, with little energy wasted in heat. They are used in flashlights, on front of appliances like TV (to indicate condition, like green for on, red for *standby*, etc.).

Figure.5 Types of LEDs

3. **Resistors:** These are color-coded (with bands or stripes), and used to control the voltages and currents in the circuit, e.g., to limit current in LEDs to set light intensity and avoid damaging the LEDs. Note that unlike an LED, a resistor is reverse (i.e., polarity does not matter).

Standard EIA Color Code Table **4 Band**: ±2%, ±5%, and ±10%



Color	1st Band (1st figure)	2nd Band (2nd figure)	3rd Band (multiplier)	4th Band (tolerance)
Black	0	0	10^0	
Brown	1	1	10^1	
Red	2	2	10^2	±2%
Orange	3	3	10^3	
Yellow	4	4	10^4	
Green	5	5	10^5	
Blue	6	6	10^6	
Violet	7	7	10^7	
Gray	8	8	10^8	
White	9	9	10^9	
Gold			10^{-1}	±5%
Silver			10^{-2}	±10%

Chart Provided By XICON

Figure 6 Color Coding

4. **Potentiometer:** A potentiometer – also called a ***pot*** – is a *variable* resistor whose value can be varied from zero to a fixed (final) value. By varying the resistor, we can vary audio volume (in radio), speed (in fan regulator), light (in dimmer switch), etc.

Figure7. Potentiometer

5. Diode: A diode is an electronic device with two *terminals* (wires). It lets current flow in one direction only (from positive or anode to negative or cathode), and acts like a gate or switch. An LED is a special type of diode, where reversing the connection will damage the device.

Signal diode handles low currents, whereas *power diodes* are used with large currents/voltages.

Figure8. Diode

6. Photoresistor: Also called light-dependent resistor (LDR) or photocell, it is a light-controlled variable resistor, i.e., its resistance changes with the intensity of light to which it is exposed.

Figure9 Photoresistor (LDR)

7. Buzzer: It is a *piezo* element, and produces a sound of a specified tone or frequency. It can also be used to *sense* or detect vibrations (e.g., knock or tap).

Figure10 Buzzer

8. Temperature Sensor: When the temperature sensor is connected to a power supply, it produces a voltage corresponding to the ambient temperature. By reading this voltage with an Arduino, we can determine the temperature and record it in the computer, etc.

Figure11. LM35

9. DC Motor: DC motors are used in toys, RC cars, robots, and many other situations. By controlling the voltage given to a motor using the Arduino, we can control the motion (angle or speed) of the platform (e.g., robot manipulator or mobile robot) to which the motor is attached.

Figure12. DC Motor

10. Push Button: Push buttons are switches which are used to turn on and off devices like motors, LEDs, and so on.

Figure13. Push Button

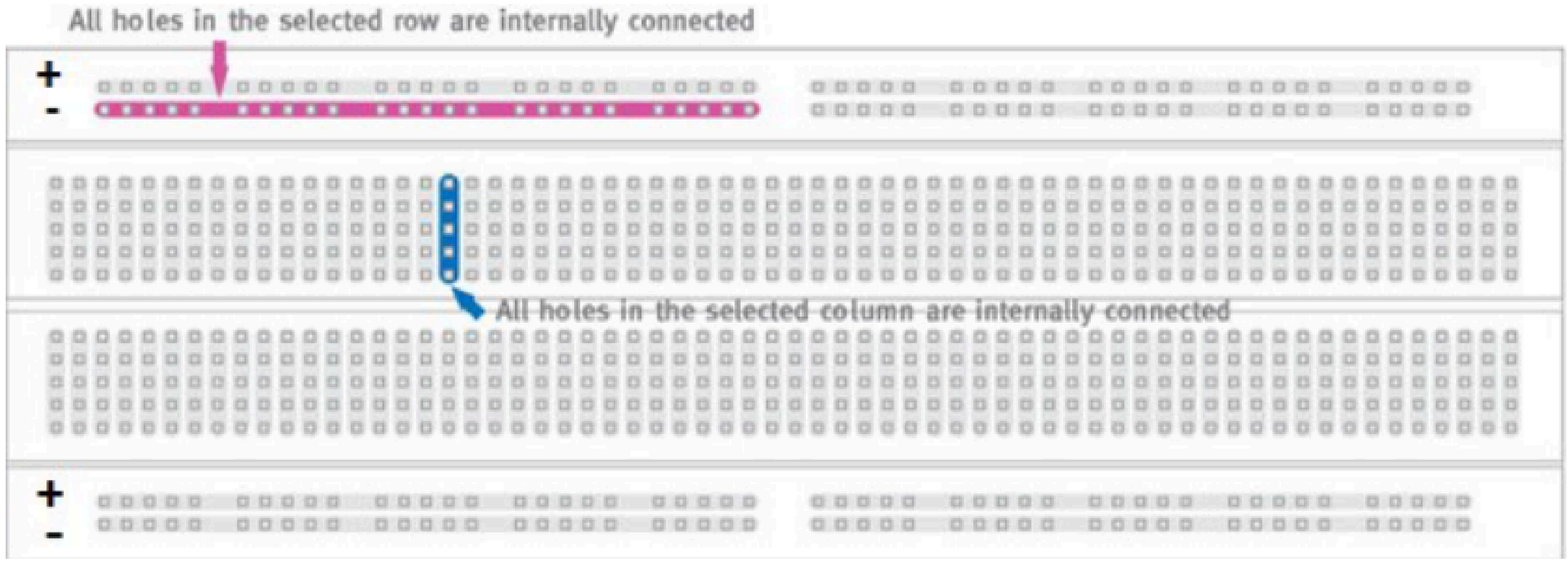
11. RC Servo: RC (for radio control) servos are motors used in toy cars, RC planes and boats (to control fins), etc, generally with less than one full rotation (whereas DC motors rotate continuously).

Figure14. RC Servo

12. Relays: A relay is a switch that is turned ON and OFF by an electrical signal (e.g., sent from Arduino). Typically, it used to turn on or off large-current devices (e.g., motors).

Figure15. Relay

13. Breadboard: A breadboard is used to build and test electronic circuits. No soldering is required so we can change connections and replace components easily. Parts will be available for re-use afterwards aft er successful testing.



2.INTRODUCTION TO PYTHON:

What is Python?

- Python is an interpreted, interactive, object-oriented programming language.
- Python is a high-level general-purpose programming language
- Python is an easy to learn, powerful programming language.

- It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes.
- Python combines remarkable power with very clear syntax.
- It has interfaces to many system calls and libraries, as well as to various window systems, and is extensible in C or C++.
- It is also usable as an extension language for applications that need a programmable interface.
- Finally, Python is portable: it runs on many Unix variants, on the Mac, and on Windows 2000 and later.
- Python is ideal language for scripting and rapid application development in many areas on most platforms.

The Python Software Foundation is an independent non-profit organization that holds the copyright on Python versions 2.1 and newer. The PSF’ s mission is to advance open source technology related to the Python programming language and to publicize the use of Python. The PSF’ s home page is athttps://www.python.org/psf/. You can do anything you want with the source.

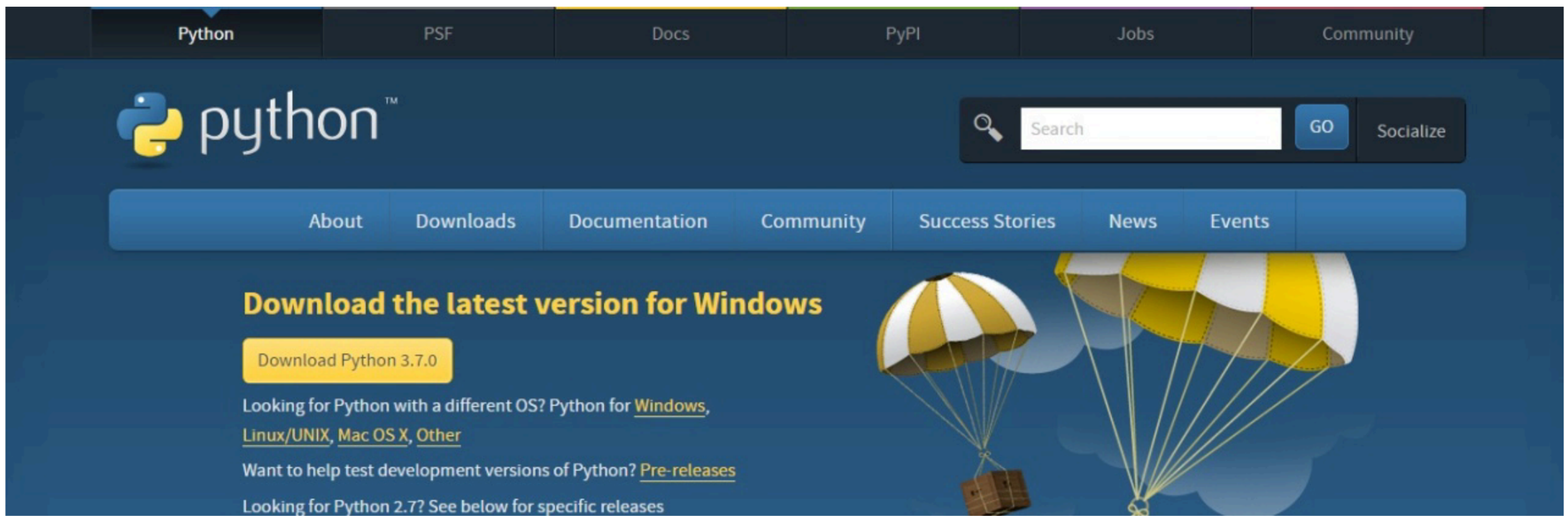
The name "Python" was adopted from the same series "Monty Python's Flying Circus".

PYTHON INSTALLATION:

Downloading steps:

1. **Go to: *https://www.python.org/downloads/***

The following page will appear in your browser.



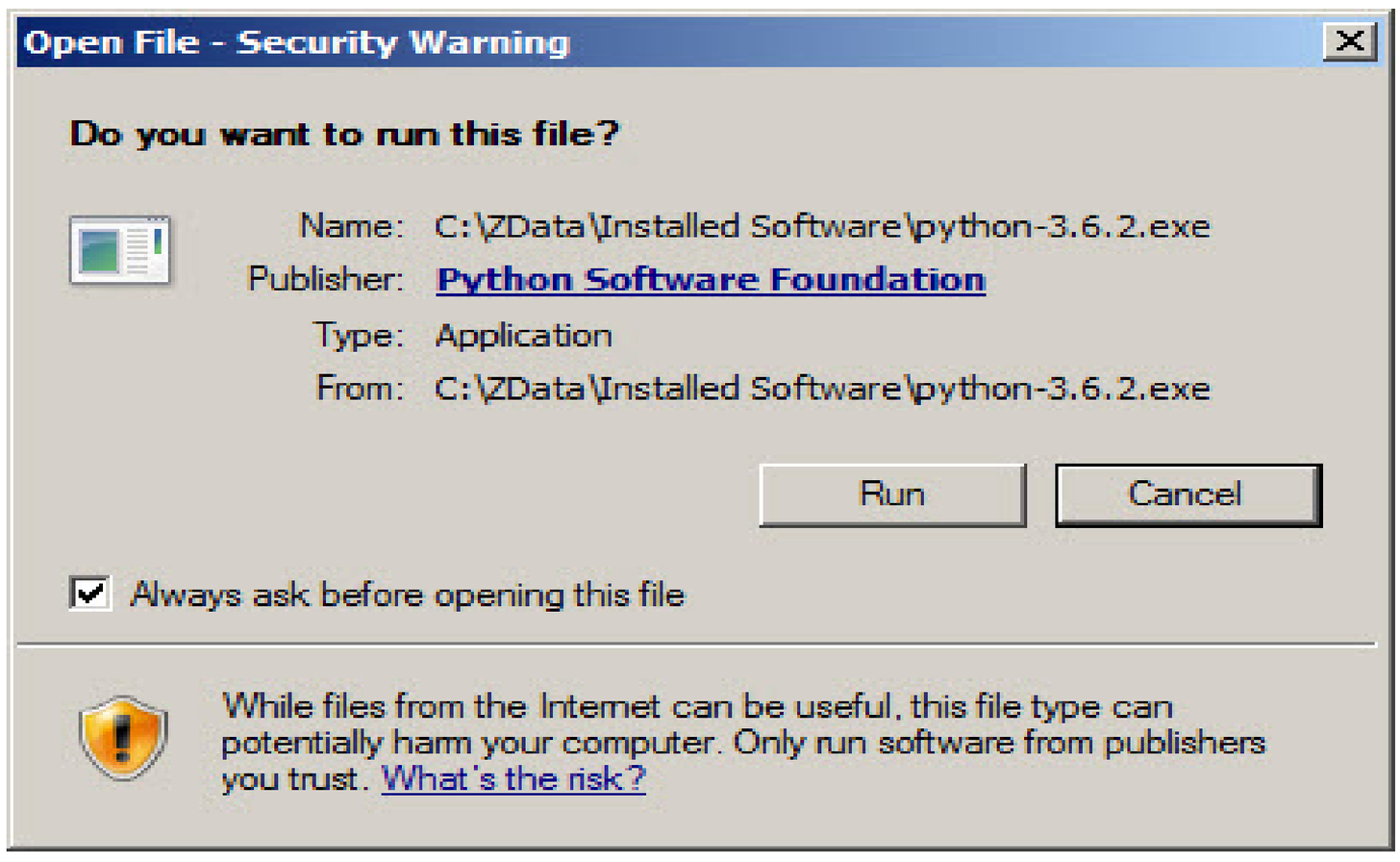
2. Click the **Download Python 3.6.2** button.

The file named **python-3.6.2.exe** should start downloading into your standard download folder. This file is about 30 Mb so it might take a while to download fully if you are on a slow internet connection (it took me about 10 seconds over a cable modem). The file appears as an exe file.

3. Move this file to a more permanent location, so that you can install Python (and reinstall it easily later, if necessary).

4 Double-click the icon labeling the file **python-3.6.2.exe**

. 5 An **Open File - Security Warning** pop-up window will appear.

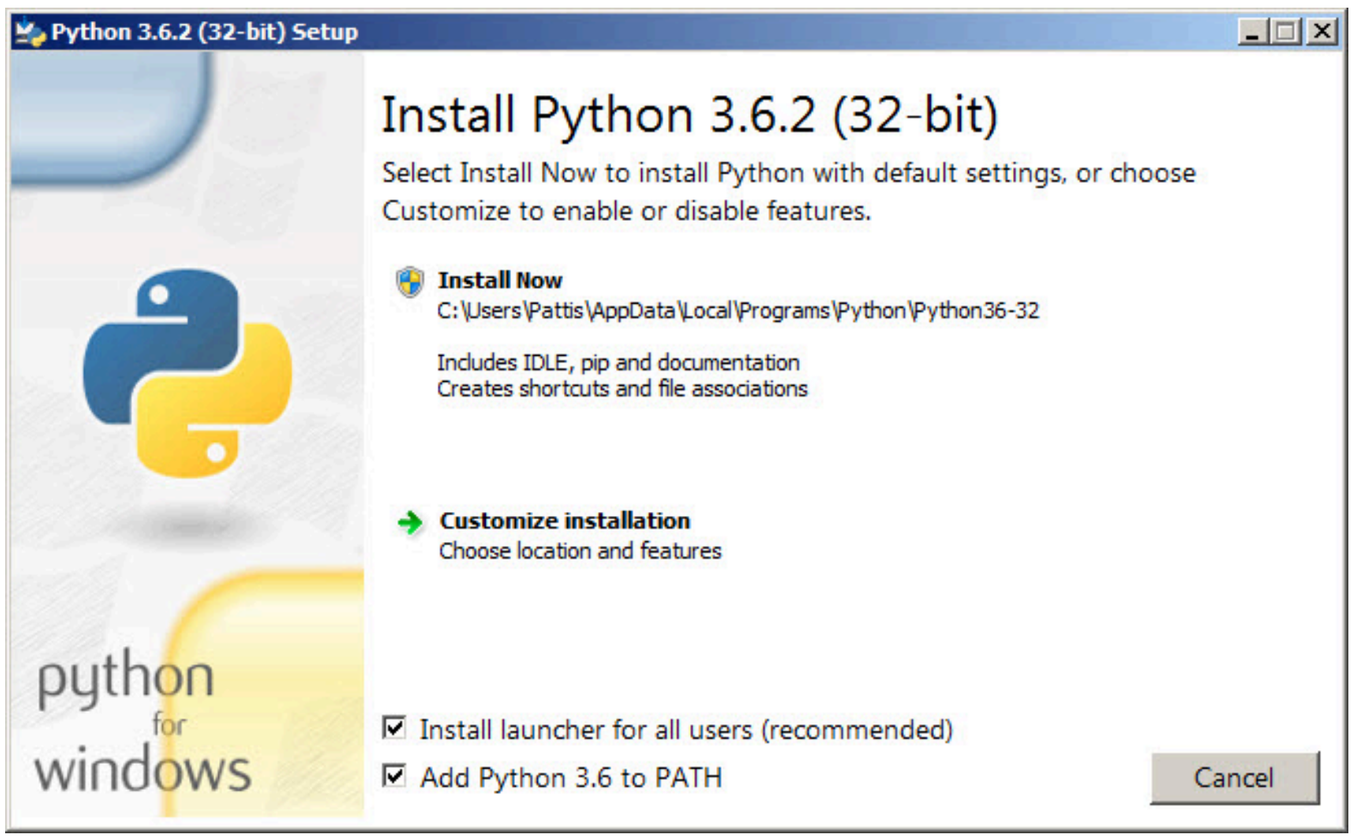


4. Click **Run**.

A **Python 3.6.2 (32-bit) Setup** pop-up window will appear.

Ensure that the **Install launcher for all users (recommended)** and the **Add Python 3.6 to PATH** checkboxes at the bottom are checked.

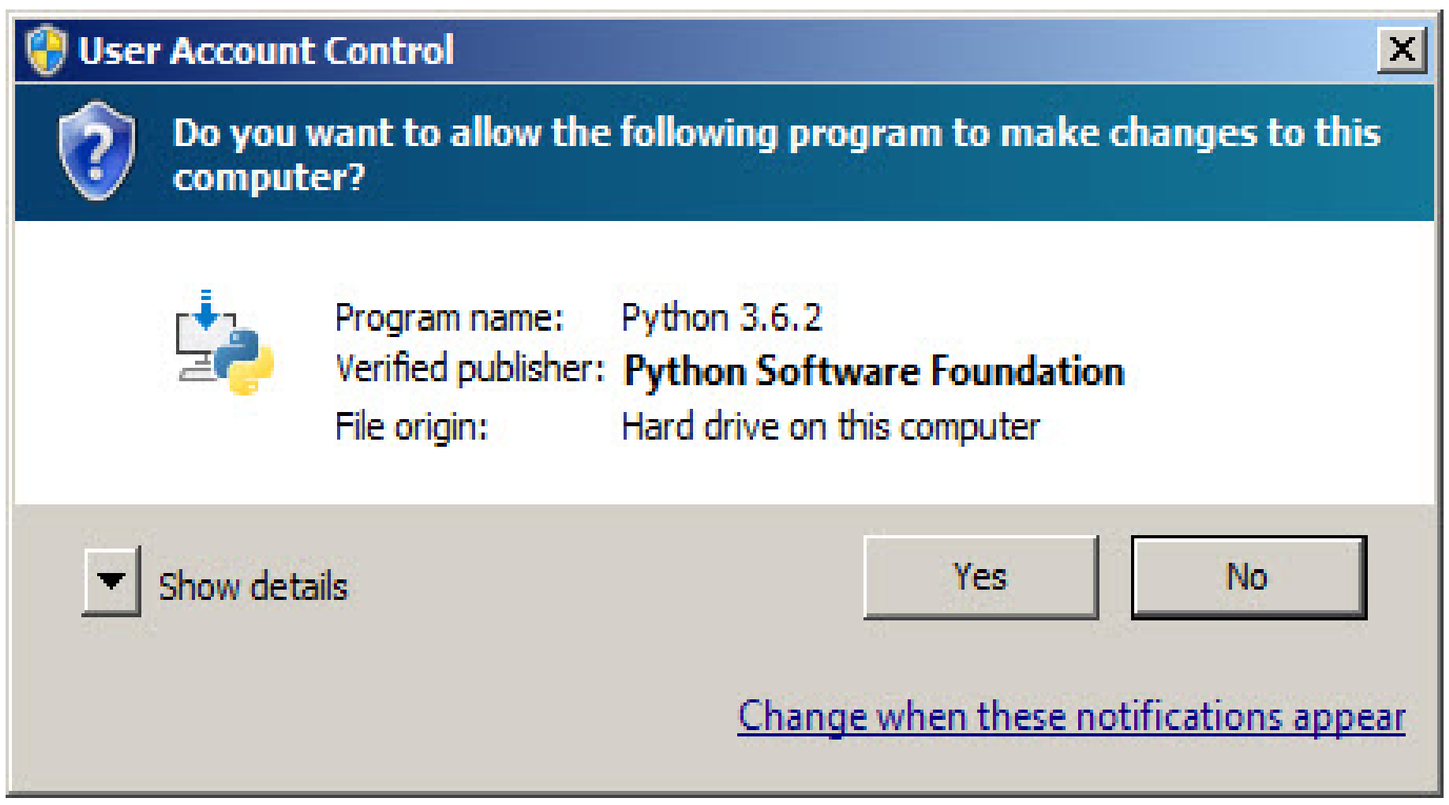
If the Python Installer finds an earlier version of Python installed on your computer, the **Install Now** message will instead appear as **Upgrade Now** (and the checkboxes will not appear).



5. Highlight the **Install Now** (or **Upgrade Now**) message, and then click it.

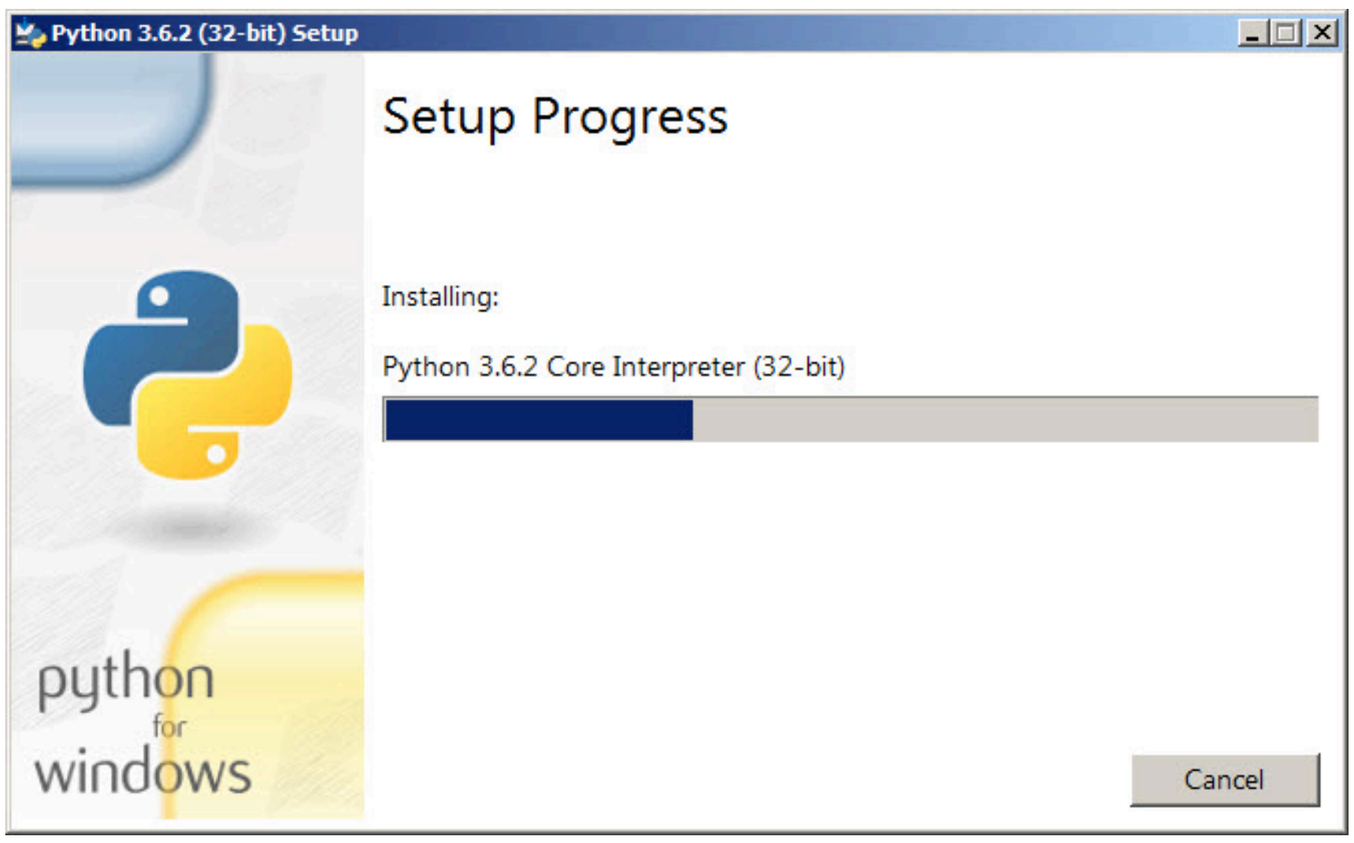
   A **User Account Conrol** pop-up window will appear, posing the question

Do you want to allow the following program to make changes to this computer?

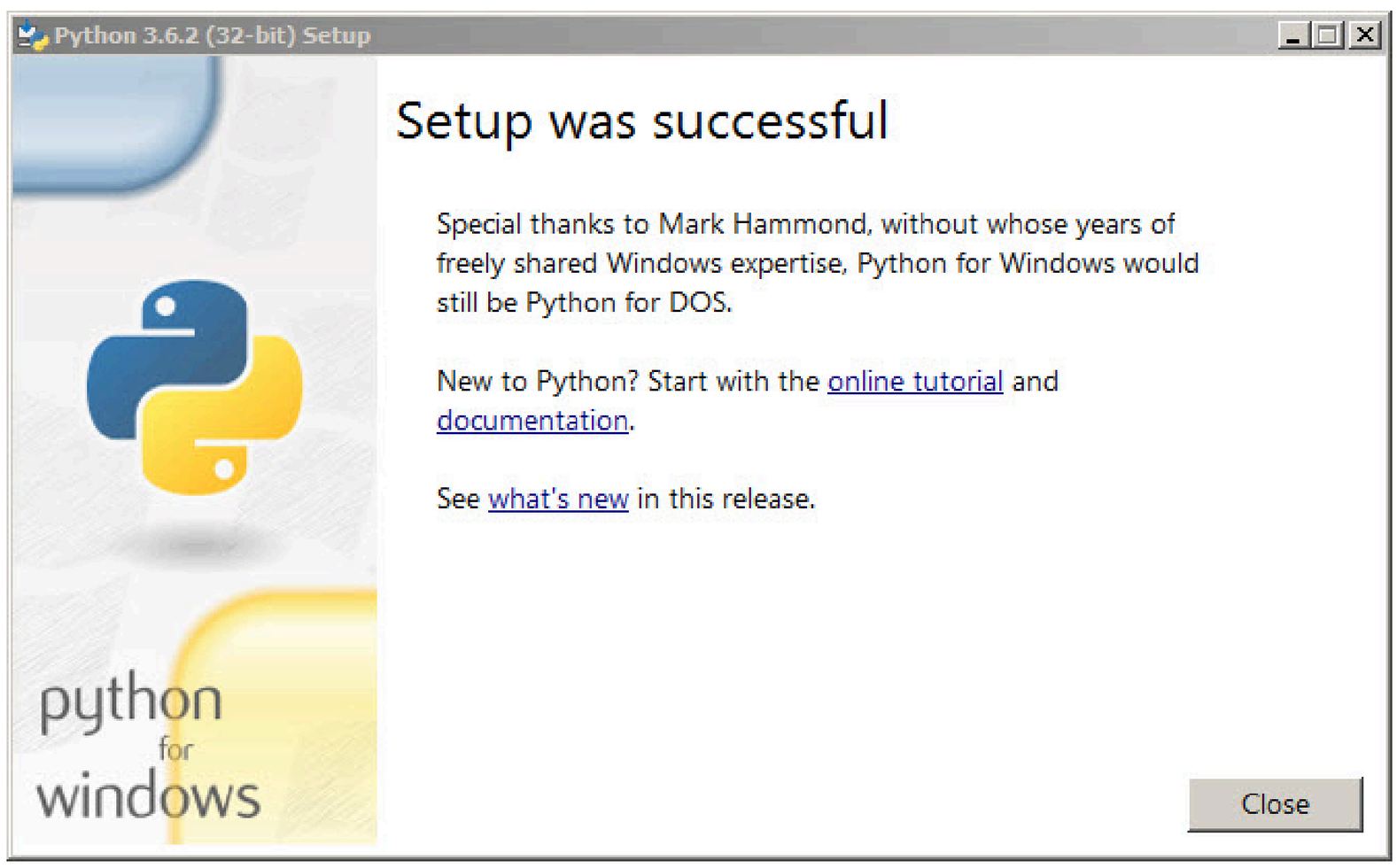


6. Click the **Yes** button.

A new **Python 3.6.2 (32-bit) Setup** pop-up window will appear with a **Setup Progress** message and a progress bar.



During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new **Python 3.6.2 (32-bit) Setup** pop-up window will appear with a **Setup was successfuly** message.

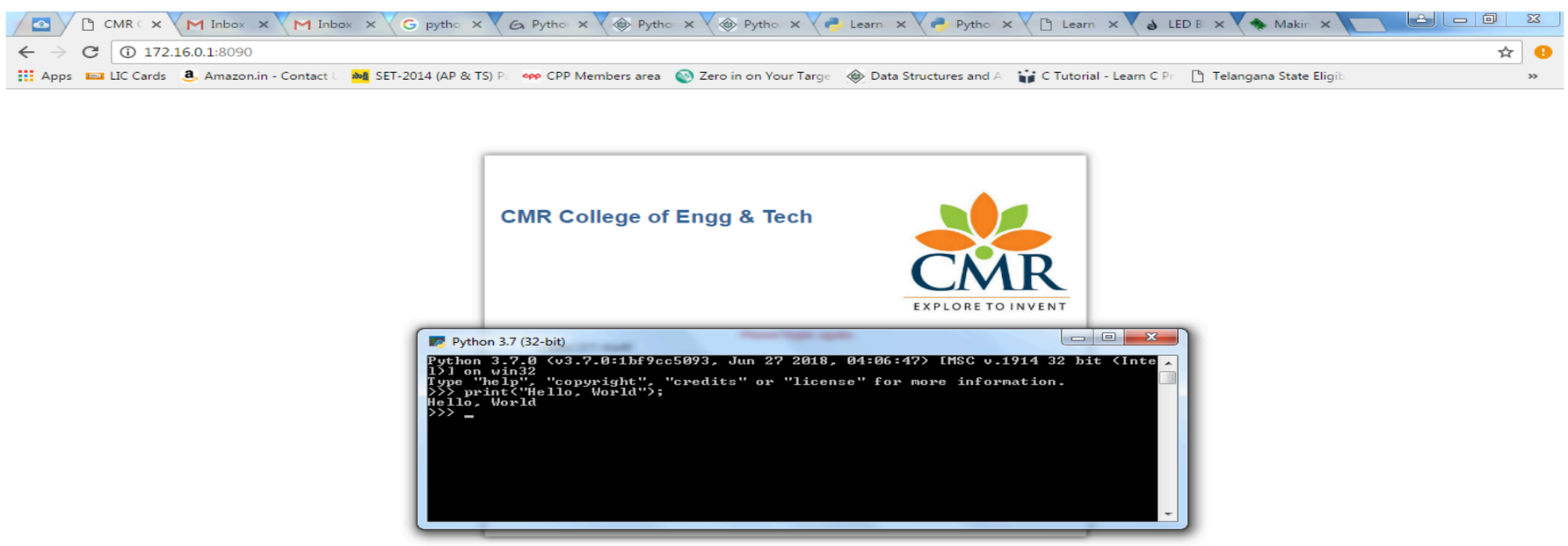


7. Click the **Close** button.

Python should now be installed.

Understanding Python Basics:

Let's write our first Python program, "Hello, World!". It's a simple program that prints Hello World!

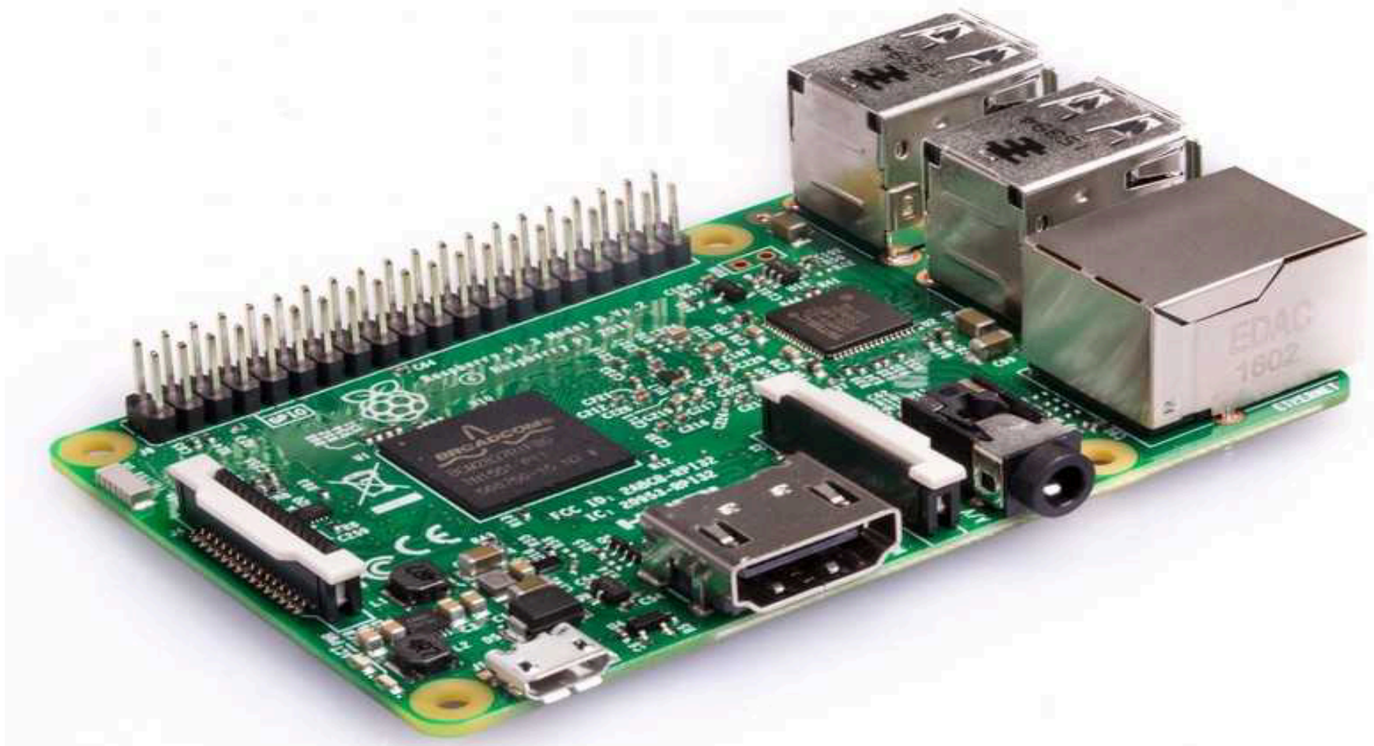


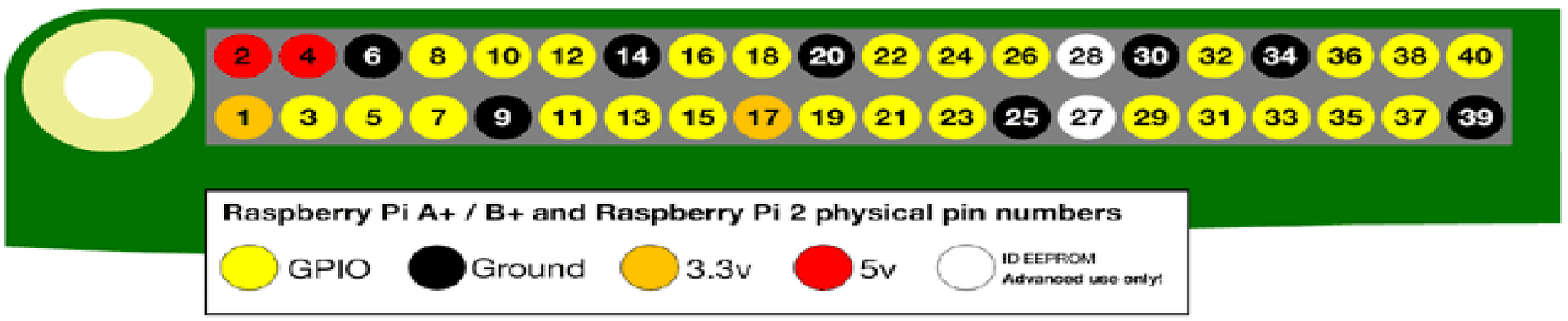
3. ABOUT RSBERRY PI 3

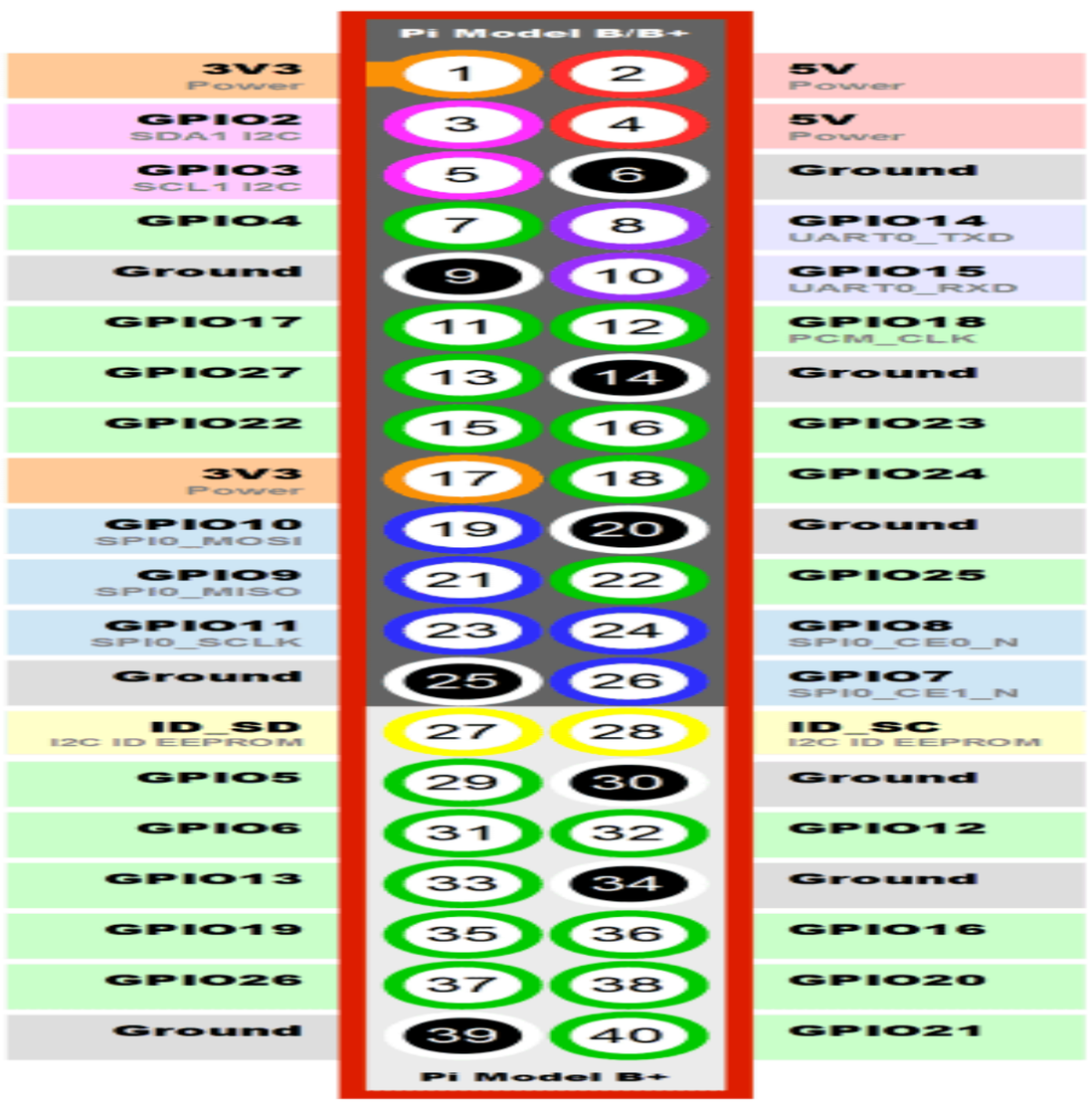
Raspberry pi

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It’ s capable of doing everything you’ d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

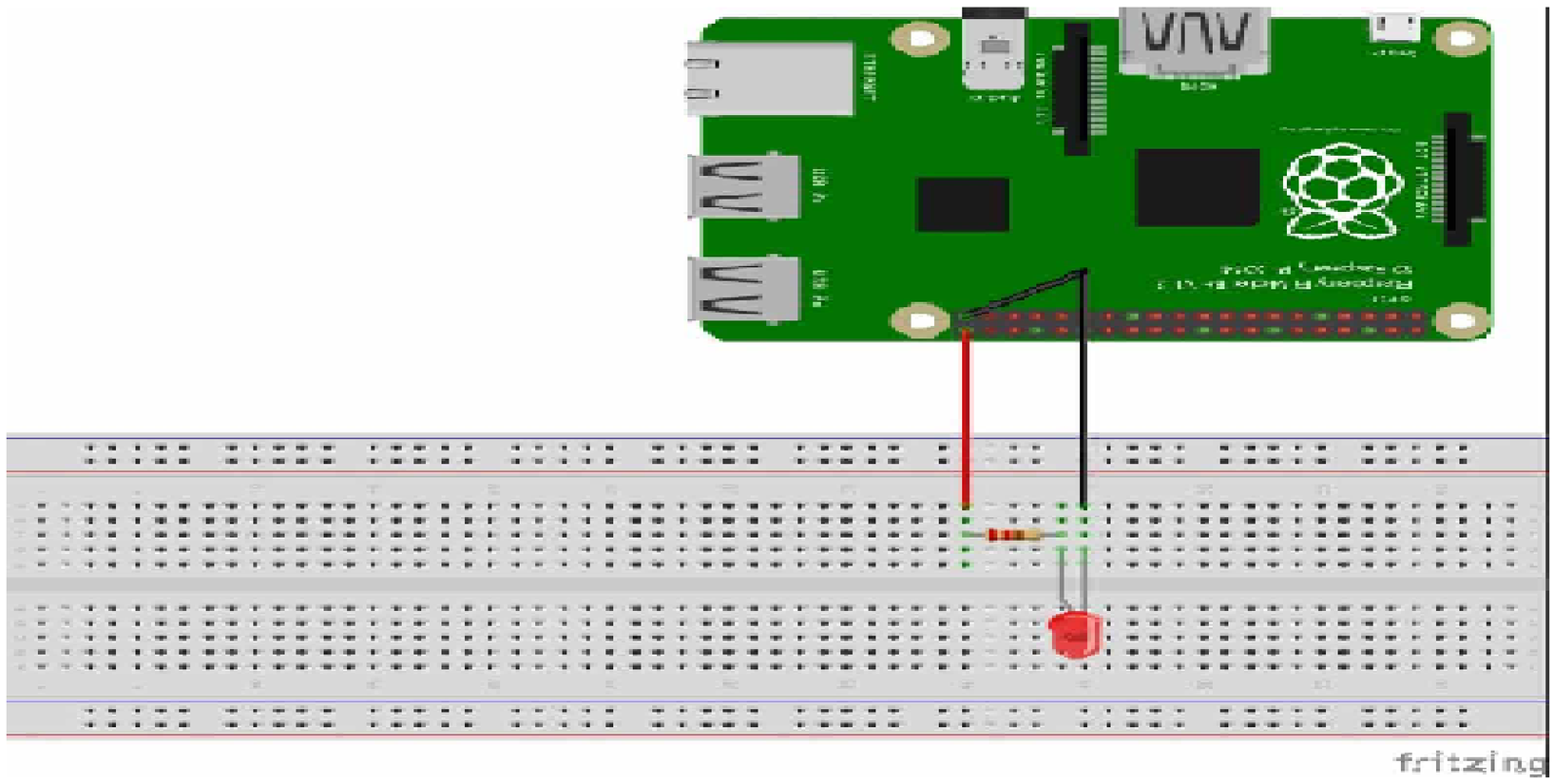
Picture:




2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
Raspberry Pi A+ / B+ and Raspberry Pi 2 physical pin numbers
GPIO
Ground
3.3v
5v
ID EEPROM
Advanced use only!


Pi Model B/B+
3V3 Power
GPIO2 SDA1 I2C
GPIO3 SCL1 I2C
GPIO4
Ground
GPIO17
GPIO27
GPIO22
3V3 Power
GPIO10 SPI0_MOSI
GPIO9 SPI0_MISO
GPIO11 SPI0_SCLK
Ground
ID_SD I2C ID EEPROM
GPIO5
GPIO6
GPIO13
GPIO19
GPIO26
Ground
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
21 22
23 24
25 26
27 28
29 30
31 32
33 34
35 36
37 38
39 40
5V Power
5V Power
Ground
GPIO14 UART0_TXD
GPIO15 UART0_RXD
GPIO18 PCM_CLK
Ground
GPIO23
GPIO24
Ground
GPIO25
GPIO8 SPI0_CE0_N
GPIO7 SPI0_CE1_N
ID_SC I2C ID EEPROM
Ground
GPIO12
Ground
GPIO16
GPIO20
GPIO21
Pi Model B+

Circuit Explanation:

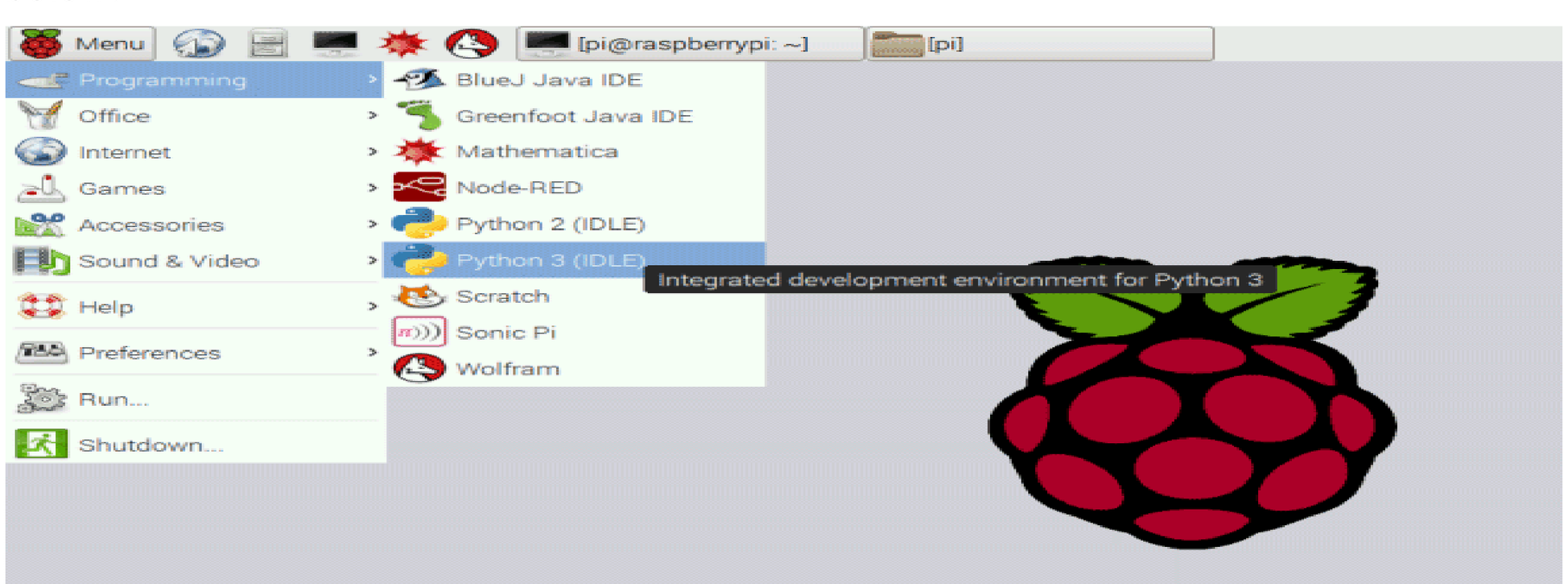


As shown in the circuit diagram we are going to connect an LED between PIN40 (GPIO21) and PIN39 (GROUND). As said earlier, we cannot draw more than 15mA from any one of these pins, so to limit the current we are connecting a 220Ω or 1KΩ resistor in series with the LED.

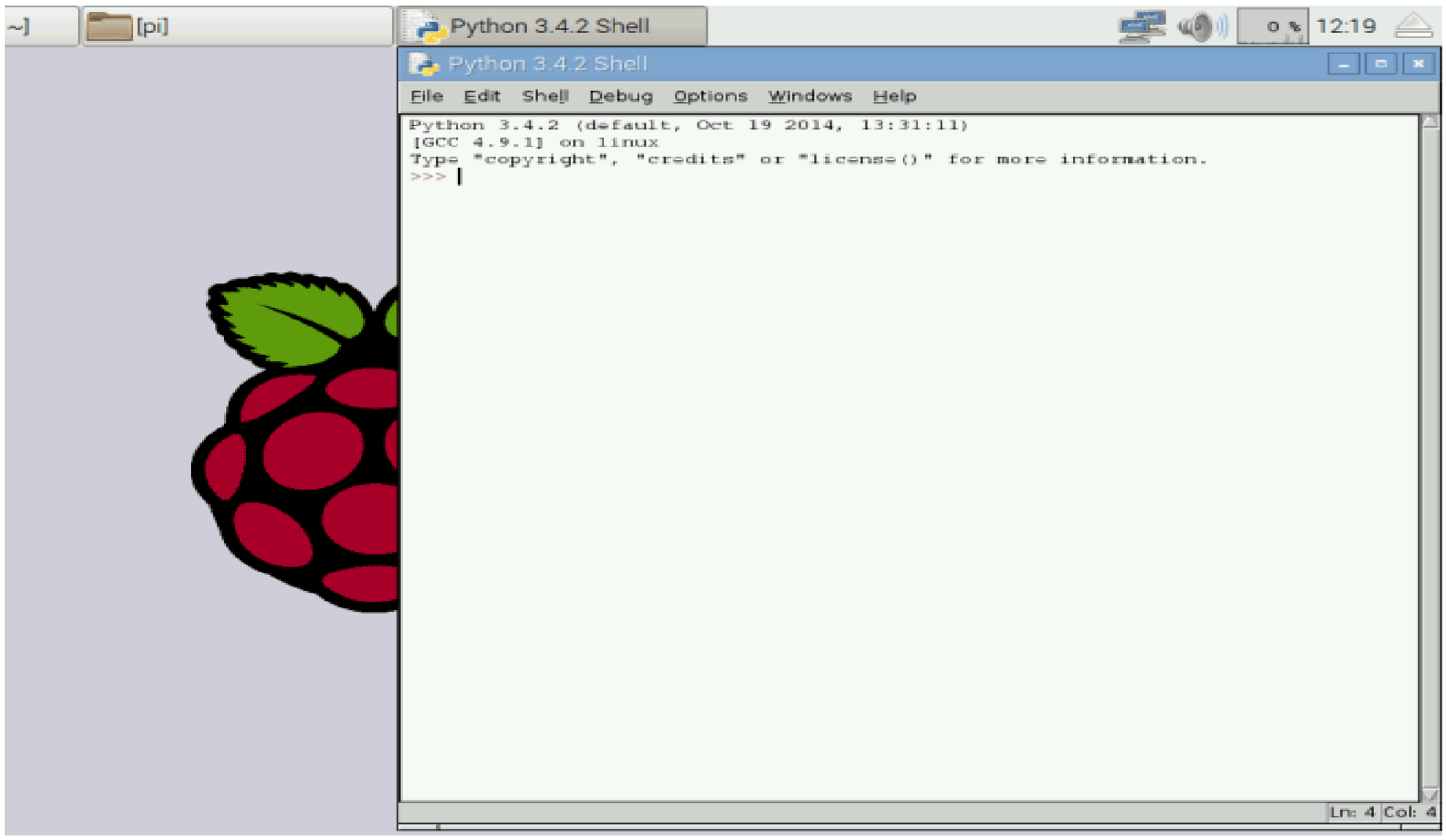
Working Explanation:

Since we have everything ready, turn ON your PI and go to the desktop.

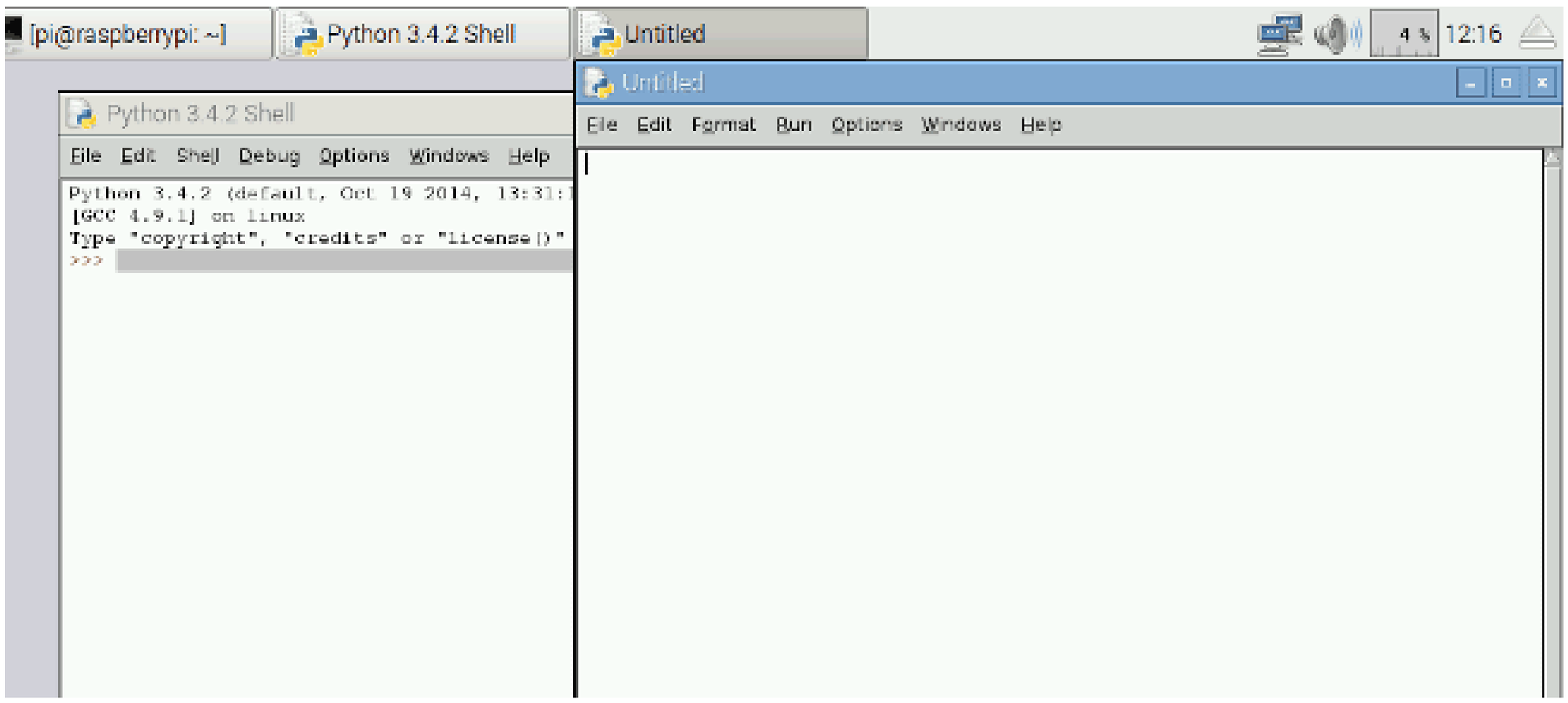
1. On the desktop, go the Start Menu and choose for the **PYTHON 3**, as shown in figure below.



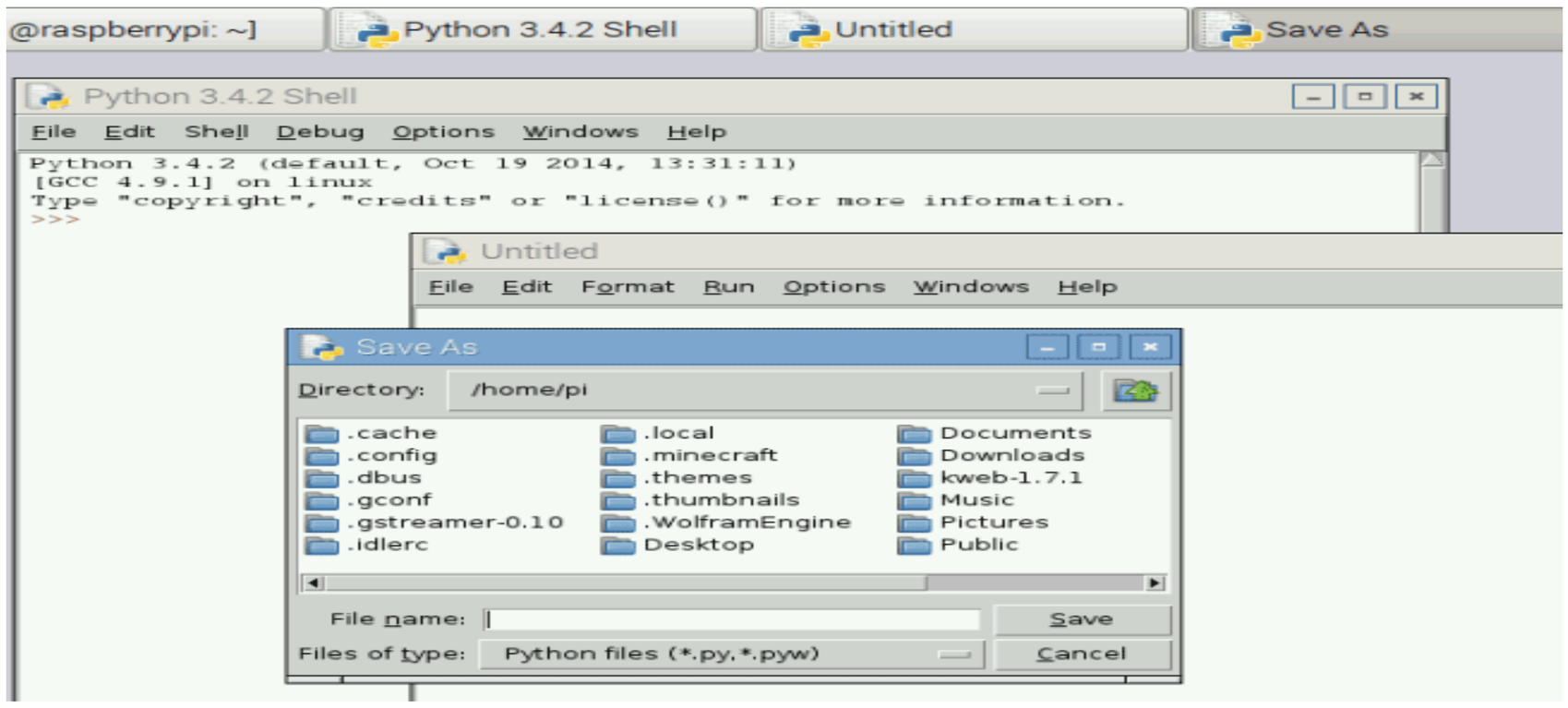
2. After that, PYHON will run and you will see a window as shown in below figure.



3. After that, click on *New File* in *File* Menu, You will see a new Window open,



4. Save this file as *blinky* on the desktop,



5. After that write the program for *blinky* as given below and execute the program by clicking on " RUN" on ' DEBUG' option.

About Raspbian OS

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.

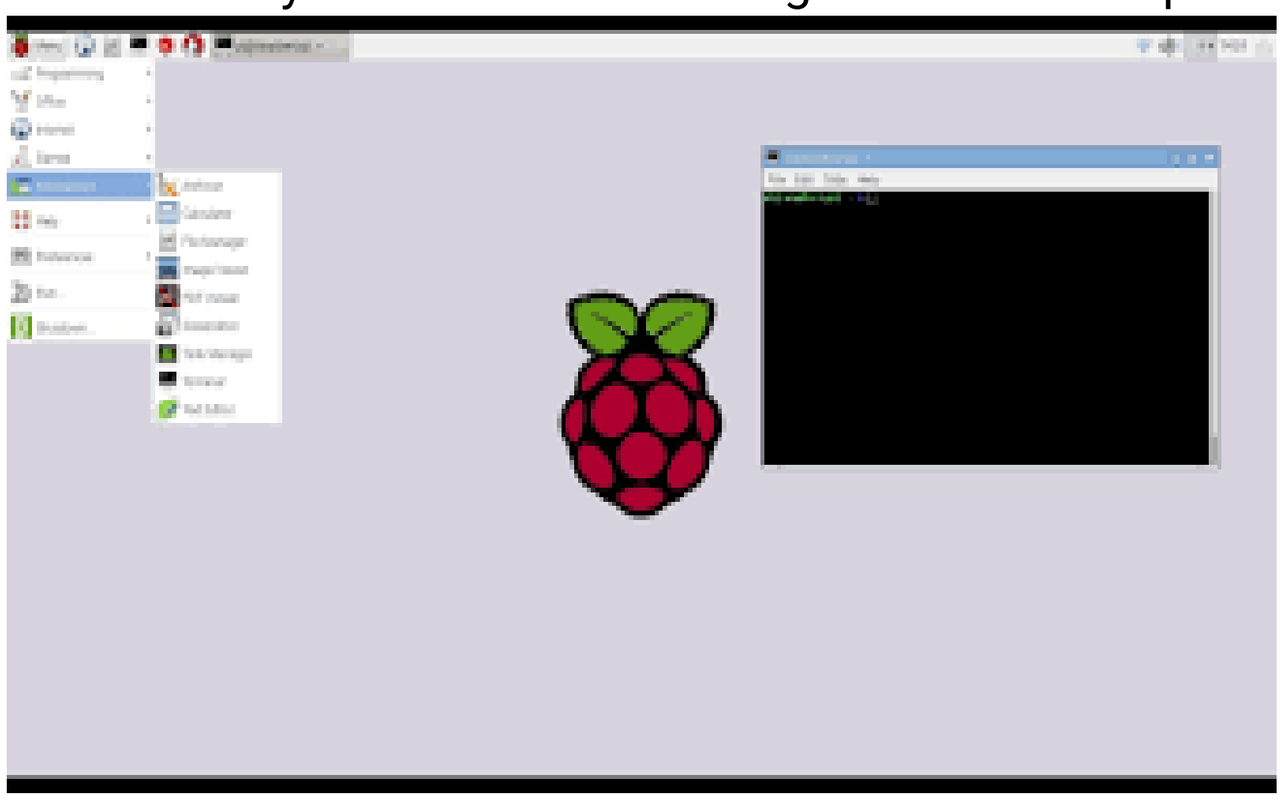
The initial build of over 35,000 Raspbian packages, optimized for best performance on the Raspberry Pi, was completed in June of 2012. However, Raspbian is still under active development with an emphasis on improving the stability and performance of as many Debian packages as possible.

Note: Raspbian is not affiliated with the Raspberry Pi Foundation. Raspbian was created by a small, dedicated team of developers that are fans of the Raspberry Pi hardware, the educational goals of the Raspberry Pi Foundation and, of course, the Debian Project.

A Raspbian image is a file that you can download onto an SD card which in turn can be used to boot your Raspberry Pi and Via APC into the Raspbian operating system. Using a Raspbian image is the easiest way for a new user to get started with Raspbian.

Raspbian is an unofficial port of Debian Wheezy armhf with compilation settings adjusted to produce optimized "hard float" code that will run on the Raspberry Pi. This provides significantly faster performance for applications that make heavy use of floating point arithmetic operations. All other applications will also gain some performance through the use of advanced instructions of the ARMv6 CPU in Raspberry Pi.

Although Raspbian is primarily the efforts of Mike Thompson (mpthompson) and Peter Green (plugwash), it has also benefited greatly from the enthusiastic support of Raspberry Pi community members who wish to get the maximum performance from their device.



WEEK1:

1. Start Raspberry Pi and try various Linix commands in command terminal window:

ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less, ps, sudo, cron, chown, chgrp, ping etc.

ls

The ls command lists the content of the current directory (or one that is specified). It can be used with the -l flag to display additional information (permissions, owner, group, size, date and timestamp of last edit) about each file and directory in a list format. The -a flag allows you to view files beginning with . (i.e. dotfiles).

Syntax of ls command :

ls [options] [file | directory]

EX: [root@localhost /]# ls

bin boot cg roup dev etc home lib lib

cd

Using cd changes the current directory to the one specified. You can use relative (i.e. cd directoryA) or absolute (i.e. cd /home/pi/directoryA) paths.

Syntax:

$ cd [directory name]

EX:vi@tecmint:~$ cd /usr/local

avi@tecmint:/usr/local$

mkdir

You can use mkdir to create a new directory, e.g. mkdir newDir would create the directory newDir in the present working directory.

Syntax: mkdir [dir-name]

Ex: mkdir test-dir

Rmdir

To remove empty directories, use rmdir. So, for example, rmdir oldDir will remove the directory oldDir only if it is empty.

Syntax:rmdir [file1][file2]

Ex: rmdir hai hello

rm

The command rmremoves the specified file (or recursively from a directory when used with -r). Be careful with this command: files deleted in this way are mostly gone for good!

Syntax:

rm [OPTION]... FILE...

Example:

$ rm a.txt

cp

Using cp makes a copy of a file and places it at the specified location (this is similar to copying and pasting). For example, cp ~/fileA /home/otherUser/would copy the file fileA from your home directory to that of the user otherUser (assuming you have permission to copy it there). This command can either take FILE FILE (cp fileA fileB), FILE DIR(cp fileA /directoryB/) or -r DIR DIR (which recursively copies the contents of directories) as arguments.

Syntax:

cp [OPTION] Source Destination

ex: $ cp a.txt b.txt

mv

The mv command moves a file and places it at the specified location (so where cp performs a 'copy-paste', mv performs a 'cut-paste'). The usage is similar to cp. So mv ~/fileA /home/otherUser/ would move the file fileA from your home directory to that of the user otherUser. This command can either take FILE FILE (mv fileA fileB), FILE DIR (mv fileA /directoryB/) or DIR DIR (mv /directoryB /directoryC) as arguments. This command is also useful as a method to rename files and directories after they've been created.

Syntax:mv [Option] source destination

Ex:

```
$mva.txtgeek.txt
```

touch

The command touch sets the last modified time-stamp of the specified file(s) or creates it if it does not already exist.

Syntax: touch file_name

Ex:

```
ubuntu@ip-172-31-36-210:~$ touch File1
ubuntu@ip-172-31-36-210:~$
ubuntu@ip-172-31-36-210:~$ ls
File1
ubuntu@ip-172-31-36-210:~$ ll
total 40
drwxr-xr-x 4 ubuntu ubuntu 4096 Dec 15 09:34 ./
drwxr-xr-x 3 root   root   4096 Dec 14 06:38 ../
-rw------- 1 ubuntu ubuntu  736 Dec 14 18:10 .bash_history
-rw-r--r-- 1 ubuntu ubuntu  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Aug 31  2015 .bashrc
drwx------ 2 ubuntu ubuntu 4096 Dec 14 06:46 .cache/
-rw-rw-r-- 1 ubuntu ubuntu    0 Dec 15 09:34 File1
-rw------- 1 ubuntu ubuntu 1428 Dec 14 09:26 .mysql_history
-rw-r--r-- 1 ubuntu ubuntu  655 May 16  2017 .profile
drwx------ 2 ubuntu ubuntu 4096 Dec 14 06:38 .ssh/
-rw-r--r-- 1 ubuntu ubuntu    0 Dec 14 07:00 .sudo_as_admin_successful
-rw------- 1 root   root    578 Dec 14 07:08 .viminfo
ubuntu@ip-172-31-36-210:~$
```

cat

You can use cat to list the contents of file(s), e.g. cat thisFile will display the contents of thisFile. Can be used to list the contents of multiple files, i.e. cat *.txt will list the contents of all .txt files in the current directory.

Syntax: $cat filename

Ex: $cat -n filename

Output

It will show content with line number

example:-cat-n geeks.txt

1)This is geeks

2)A unique array

chmod

You would normally use chmod to change the permissions for a file. The chmodcommand can use symbols u (user that owns the file), g (the files group) , and o (other users) and the permissions r (read), w (write), and x (execute). Using chmod u+x *filename* will add execute permission for the owner of the file.

Syntax :

chmod [reference][operator][mode] file...

ex:

SReference	Class	Description
u	owner	file's owner
g	group	users who are members of the file's group
o	others	users who are neither the file's owner nor members of the file's group
a	all	All three of the above, same as ugo

chown

The chown command changes the user and/or group that owns a file. It normally needs to be run as root using sudo e.g. sudo chown pi:root *filename* will change

Syntax:

chown [OPTION]… [OWNER][:[GROUP]] FILE…

chown [OPTION]… – reference=RFILE FILE…

Example: To change owner of the file:

Ex:chown owner_name file_name

```
root@kali:~# ls -l
total 80
drwxr-xr-x  2 root   root    4096 Feb  2 05:29 Desktop
drwxr-xr-x  2 root   root    4096 Feb  2 05:05 Documents
drwxr-xr-x  2 root   root    4096 Feb  3 06:41 Downloads
-rw-r--r--  1 root   root     202 Feb  4 12:08 example.a
-rw-r--r--  1 master group1    12 Feb  4 12:04 file1.txt
-rw-r--r--  1 master group1    61 Feb  4 12:06 file2.txt
-rw-r--r--  1 root   group1    7 Feb  4 12:09 greek1
-rw-r--r--  1 master group1    6 Feb  4 12:10 greek2
-rw-r--r--  1 master group1    6 Feb  4 12:21 greek3
-rw-r--r--  1 root   root     208 Feb  4 12:23 greeks.a
drwxr-xr-x  2 root   root    4096 Feb  2 05:05 Music
drwxr-xr-x  2 root   root    4096 Feb  2 05:05 Pictures
drwxr-xr-x  2 root   root    4096 Feb  2 05:05 Public
-rw-r--r--  1 root   root       6 Feb  5 14:04 star1.txt
-rw-r--r--  1 root   root       7 Feb  5 14:01 star2.txt
-rw-r--r--  1 root   root       6 Feb  5 14:01 star3.txt
-rw-r--r--  1 root   root       8 Feb  5 12:33 star.a
-rw-r--r--  1 root   root     208 Feb  5 13:12 super.a
drwxr-xr-x  2 root   root    4096 Feb  2 05:05 Templates
drwxr-xr-x  2 root   root    4096 Feb  2 05:05 Videos
root@kali:~#
```

sudo

The sudo command enables you to run a command as a superuser, or another user. Use sudo -s for a superuser shell. For more details see Root user / sudo

Syntax:

sudo -V

Ex:

-V: The -V (version) option causes sudo to print the version number and exit. If the invoking user is already root, the -V option will print out a list of the defaults sudo was compiled with.

```
himanshu@himanshu-VirtualBox: ~
himanshu@himanshu-VirtualBox:~$ sudo -V
Sudo version 1.8.21p2
Sudoers policy plugin version 1.8.21p2
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.21p2
himanshu@himanshu-VirtualBox:~$
```

gzip

The unzip command extracts the files from a compressed zip file.

Syntax :

gzip [Options] [filenames]

Ex:

Gzip my doc.txt

Options :

-f option : Sometimes a file cannot be compressed. Perhaps you are trying to compress a file called “ myfile1” but there is already a file called “ myfile1.gz” . In this instance, the “ gzip” command won’ t ordinarily work.

ping

The ping utility is usually used to check if communication can be made with another host. It can be used with default settings by just specifying a hostname (e.g. ping raspberrypi.org) or an IP address (e.g. ping 8.8.8.8). It can specify the number of packets to send with the -c flag.

Syntax:

sudo ping – v

ex:

WEEK 2:

2. Run some python programs on Pi like:

A. Read your name and print Hello message with name

```
>>> name = "Sarah"
>>> "Hello " + name
'Hello Sarah'
```

B. Read two numbers and print their sum, difference, product and division.

```
print("Enter 'x' for exit.");
print("Enter any two number: ");
num1 = input();
num2 = input();
if num1 == 'x':
   exit();
else:
   ch = input("Enter operator (+,-,*,/): ");
   if ch == '+':
            res = int(num1) + int(num2);
            print(num1, "+", num2, "=", res);
   elif ch == '-':
            res = int(num1) - int(num2);
            print(num1, "-", num2, "=", res);
   elif ch == '*':
            res = int(num1) * int(num2);
            print(num1, "*", num2, "=", res);
   elif ch == '/':
            res = int(num1) / int(num2);
            print(num1, "/", num2, "=", res);
   else:
            print("Strange input..exiting..");
```

Output:

```
Enter 'x' for exit.
Enter any two number:
2
5
Enter operator (+,-,*,/): +
2 + 5 = 7
Enter 'x' for exit.
Enter any two number:
6
10
Enter operator (+,-,*,/): *
6 * 10 = 60
Enter 'x' for exit.
Enter any two number:
10
2
Enter operator (+,-,*,/): /
10 / 2 = 5.0
Enter 'x' for exit.
Enter any two number:
12
20
Enter operator (+,-,*,/): -
12 - 20 = -8
Enter 'x' for exit.
Enter any two number:
x
```

C. Word and character count of a given string

```
print("Enter 'x' for exit.");

string = input("Enter any string to count word: ");

if string == 'x': exit();

 else: word_length = len(string.split());

 print("\nNumberof words =",word_length);
```

Output:

Enter any **string** to **count word: hello**

Number of words:5

D. Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input

Python Program to find the area of triangle

```
a = 5

b = 6

c = 7

# Uncomment below to take inputs from the user

# a = float(input('Enter first side: '))

# b = float(input('Enter second side: '))

# c = float(input('Enter third side: '))

# calculate the semi-perimeter

s = (a + b + c) / 2

# calculate the area

area = (s*(s-a)*(s-b)*(s-c)) ** 0.5

print('The area of the triangle is %0.2f' %area)
```

OutputThe area of the triangle is 14.70

Python Program to find Area Of Circle using Radius

```
PI = 3.14
radius = float(input(' Please Enter the radius of a circle: '))
area = PI * radius * radius
circumference = 2 * PI * radius

print(" Area Of a Circle = %.2f" %area)
```

output:

please enter the radius of the circle:6

Area of the circle:113.04

Python Program to find Area of a Rectangle

```
width = float(input('Please Enter the Width of a Rectangle: '))
height = float(input('Please Enter the Height of a Rectangle: '))

# calculate the area
Area = width * height

# calculate the Perimeter
Perimeter = 2 * (width + height)

print("\n Area of a Rectangle is: %.2f" %Area)
```

output:

```
Please Enter the Width of a Rectangle: 8
Please Enter the Height of a Rectangle: 6

Area of the rectangle:40.00
```

E. Print a name 'n' times, where name and n are read from standard input, using for and while loops.

```
import sys

text = ""

while 1:

  c = sys.stdin.read(1)

  text = text + c

  if c == '\n':

    break

print("Input: %s" % text)
```

Output:

Though it's possible to read input like this, usually the function input() is used.

```
>>> name = input("What's your name?\n")

What's your name?

Tux

>>> print(name)

Tux

>>>
```

F.Handle Divided by Zero Exception.

```
try:
    print 1/0
except ZeroDivisionError:
    print "You can't divide by zero!"
```

Then Python will print this:

You can't divide by zero!

G. Print current time for 10 times with an interval of 10 seconds.

```
import time
from threading import Event, Thread

class RepeatedTimer:

  """Repeat `function` every `interval` seconds."""

  def __init__(self, interval, function, *args, **kwargs):
    self.interval = interval
    self.function = function
    self.args = args
    self.kwargs = kwargs
    self.start = time.time()
    self.event = Event()
    self.thread = Thread(target=self._target)
    self.thread.start()
                    def _target(self):
    while not self.event.wait(self._time):
      self.function(*self.args, **self.kwargs)
  @property
  def _time(self):
    return self.interval - ((time.time() - self.start) % self.interval)
  def stop(self):
    self.event.set()
    self.thread.join()
# start timer
timer = RepeatedTimer(10, print, 'Hello world')
# stop timer
timer.stop()
```

H. Read a file line by line and print the word count of each line.

$pythonreadline.py

```
filepath = 'Iliad.txt'

with open(filepath) as fp:

  line = fp.readline()

  cnt = 1

  while line:

    print("Line {}: {}".format(cnt, line.strip()))

    line = fp.readline()

    cnt += 1
```

output:

```
$ python forlinein.py

 Line 0: BOOK I

Line 1:

Line 2: The quarrel between Agamemnon and Achilles--Achilles withdraws

Line 3: from the war, and sends his mother Thetis to ask Jove to help

Line 4: the Trojans--Scene between Jove and Juno on Olympus.

Line 5:

 Line 6: Sing, O goddess, the anger of Achilles son of Peleus, that brought

 Line 7: countless ills upon the Achaeans. Many a brave soul did it send

Line 8: hurrying down to Hades, and many a hero did it yield a prey to dogs and

Line 9: vultures, for so were the counsels of Jove fulfilled from the day on ...
```

3.Light an LED through Python program

Aim: write the python program to light LED

Program:

AIM: To Blink an LED (Light Emitting Diode) Using Raspberry pi 3

Components Required: Connecting pins

1KΩresistor

Raspberry pi 3 B+ Model

LED Bread Board

HDMI Cable Power Cable

Circuit Explanation: As shown in the circuit diagram we are going to connect an LED between PIN18 and PIN39 (GROUND) so to limit the current we are connecting a 1KΩ resistor in series with the LED.

Working Explanation: Turn ON your PI and go to the desktop

1. On the desktop, go the Start Menu and choose for the PYTHON 3, as shown in figure below. 2. After that, PYHON will run and you will see a window as shown in below figure.

3. After that, click on New File in File Menu, You will see a new Window open

4. Save this file as blinkpy on the desktop

5. After that write the program for blinkpy as given below and execute the program by clicking on " RUN" on ' DEBUG' option.

Code Explanation:

In the Python program, first we have imported two packages RPi.GPIO and Time. The package RPi.GPIO will help us in controlling the GPIO Pins of the Raspberry Pi. The first important function of the RPi.GPIO Module is the setmode(). Using GPIO.setmode(), we can select either GPIO Numbering of the Pins or Physical Numbering. By using GPIO.setmode(GPIO.BOARD), we are selecting the Physical Numbering Scheme. NOTE: For GPIO Numbering, you can use GPIO.setmode(GPIO.BCM). The next function is the setup(pin,mode). This function will allow us to set the pin as either input (GPIO.IN) or as output (GPIO.OUT). In the program, I have set the ledPin as output by using GPIO.setup(ledPin, GPIO.OUT).After setting the LED Pin as OUTPUT, now we need to set the state of this OUTPUT i.e. HIGH (GPIO.HIGH) or LOW (GPIO.LOW). For this, we need to use the function output(pin, state). So, in our program, we need to use GPIO.output(ledPin, GPIO.HIGH) for turning ON the LED and GPIO.output(ledPin, GPIO.LOW) for turning it OFF.

Code: import RPi.GPIO as GPIO

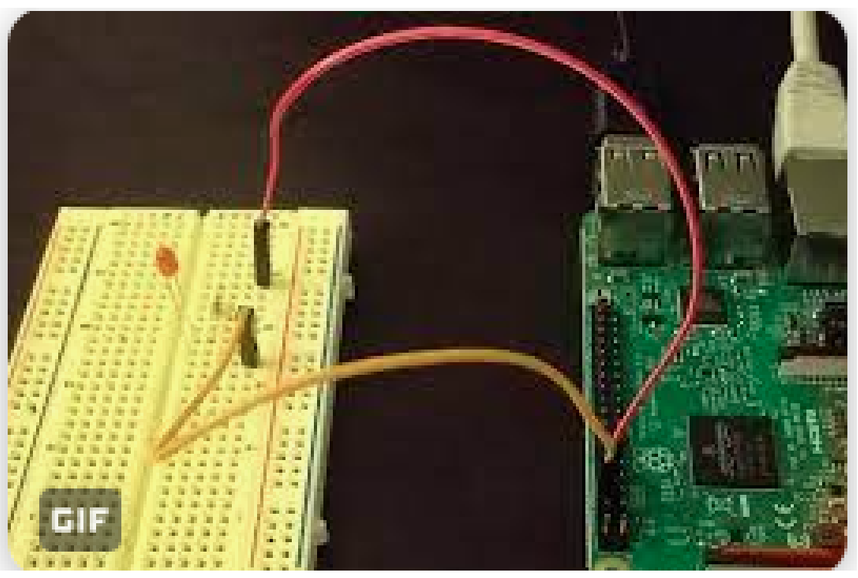
import time LED_PIN=18

GPIO.set mode(GPIO.BCM)

GPIO.setup(LED_PIN,GPIO.OUT)

```
while True: print(" LED IS ON")
GPIO.output(18,GPIO.HIGH)
time.sleep(1)
print("LED IS OFF")
 GPIO.output(18,GPIO.LOW)
 time.sleep(1)
Output: LED IS ON LED IS OF

import RPi.GPIO as GPIO
import time LED_PIN=18
 GPIO.set mode(GPIO.BCM)
GPIO.setup(LED_PIN,GPIO.OUT)
while True: print(" LED IS ON")
GPIO.output(18,GPIO.HIGH)
time.sleep(1)
print("LED IS OFF")
 GPIO.output(18,GPIO.LOW)
 time.sleep(1)
Output: LED IS ON LED IS OF
```

OUTPUT:



WEEK 4

Aim: Write the pythonprogram Get input from two switches and switch on corresponding LEDs

Program:

Aim: Get input from two switches and switch on corresponding LEDs

Hardware Required

- *Raspberry Pi*
- Micro USB Power Adapter
- *Bread Board*
- *LED*
- Push Button
- *Resistor -330 ohm and 10K ohm*
- *Jumper wires - Male to Female and Male to Male*

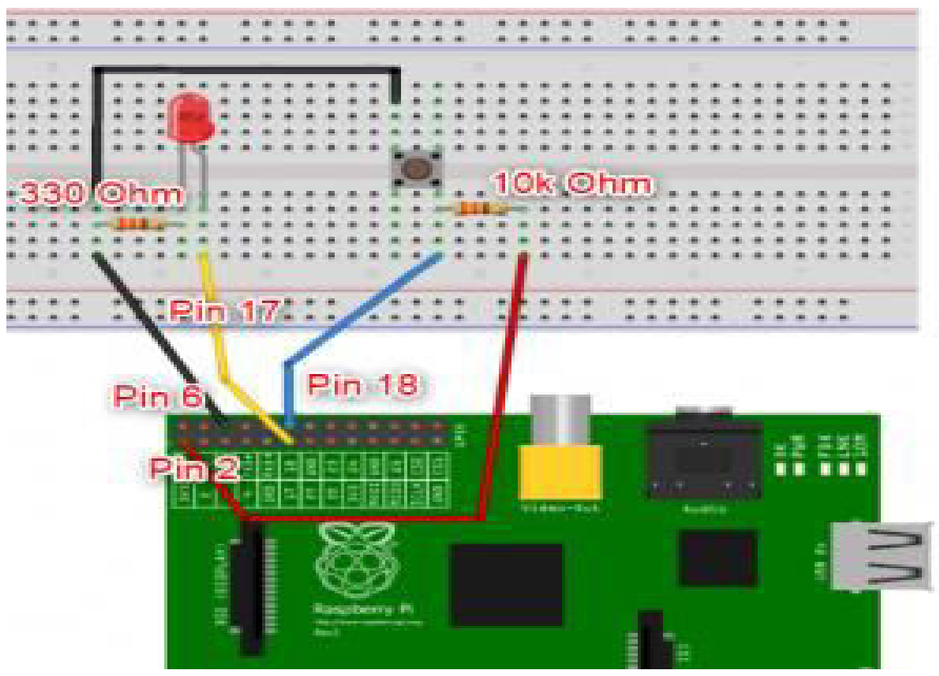
Using a breadboard is the best way to do prototyping the electronic circuits rather soldering the

components together on a PCB.We use breadboard to test circuit design & it is easy to make the changes , as we just need to re-plug the wires.

The details for GPIO pins I have described in my previous post, refer the post my IoT devices .

Lets see how we need to do the connection or circuit for this raspberry pi push button led control.

I have used the GPIO pin 2 for 5 v , GPIO pin 6 for GND, GPIO pin 17 as Output and GPIO pin18 as Input for the circuit. Connect the GPIO PIN 17 to the anode of LED & the GPIO PIN 18 tothe Push Button. Connect the resistors , GND and 5V as shown in the below diagram. Once we have the circuit setup & the raspberry pi is powered on, it is now time to write the python code.

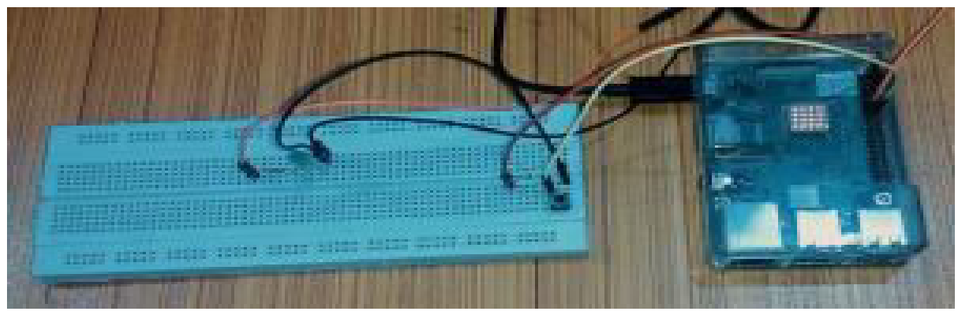


PROGRAM

```
import RPi.GPIO as GPIO import
time
GPIO.setmode(GPIO.BOARD)
GPIO.setup(17, GPIO.OUT)

#LED is connected to Pin 17

GPIO.setup(18, GPIO.INPUT)

#PushButton is connected to Pin
18try:
while True:

if GPIO.input (18) == 0:
print ("led is On")
GPIO.output (17, True)
if GPIO.input (18) == 1:
print ("led is OFF")
GPIO.output (17, False)
finally:
GPIO.cleanup ()
```

The real diagram of using a push button with Raspberry Pi GPIO

OUTPUT:



WEEK 5:

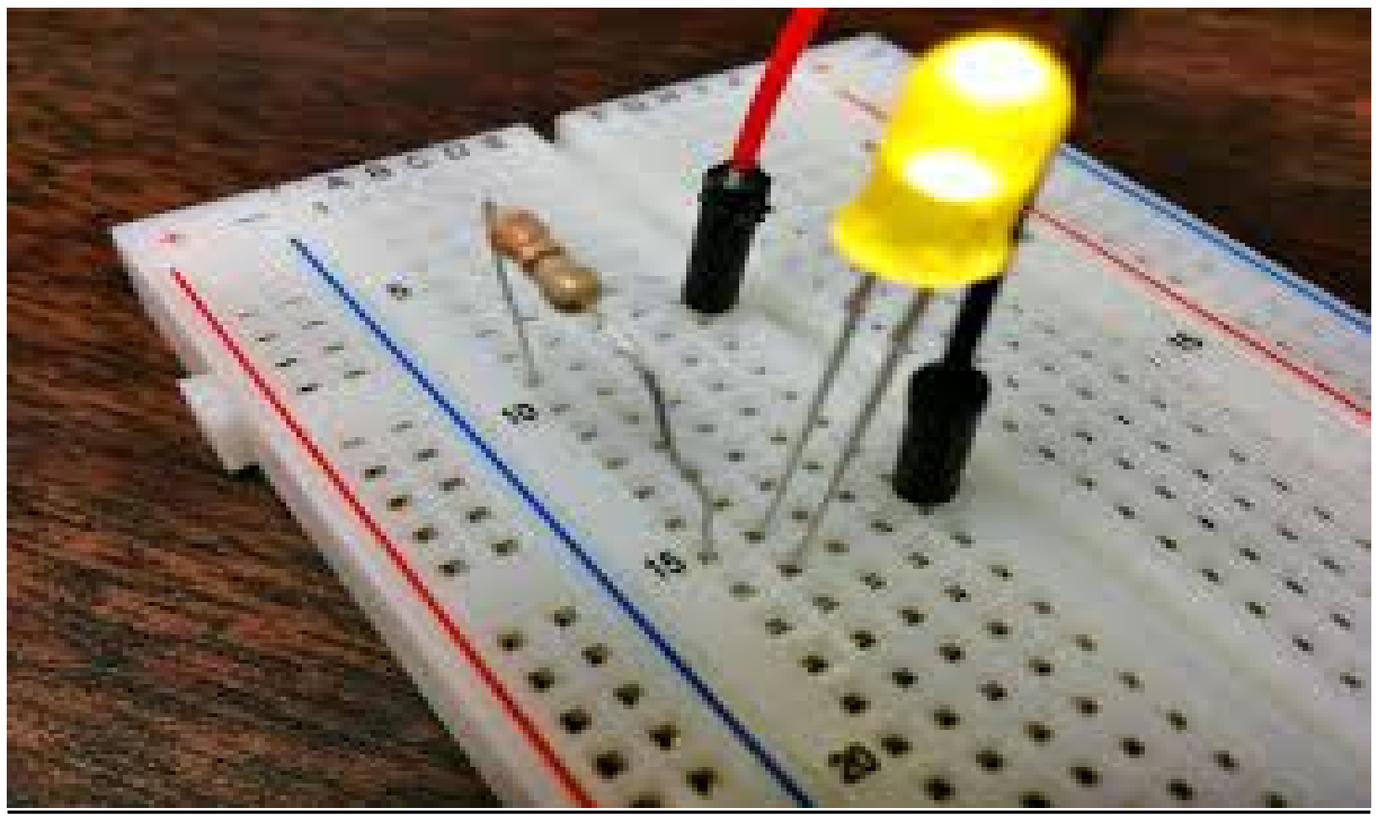
Aim:Flash an LED at a given on time and off time cycle, where the two times are taken from a file.

Program:

```
import RPI.GPIOasGPIO
```

```
import time
    LED_PIN=11
   GPIO.setmode(GPIO.BOARD)     ## Use board pin numbering
   GPIO.setup(11, GPIO.OUT)     ## Setup GPIO Pin 11 to OUT
   while True:
            GPIO.output(11,True)  ## Turn on Led
            time.sleep(1)        ## Wait for one second
            GPIO.output(11,False) ## Turn off Led
          time.sleep(1)        ## Wait for one second
```

OUTPUT:



WEEK:6

Aim: Write the python Flash an LED based on cron output (acts as an alarm)

Components Required:

Connecting pins

1KΩ resistor

Raspberry pi 3 B+ Model

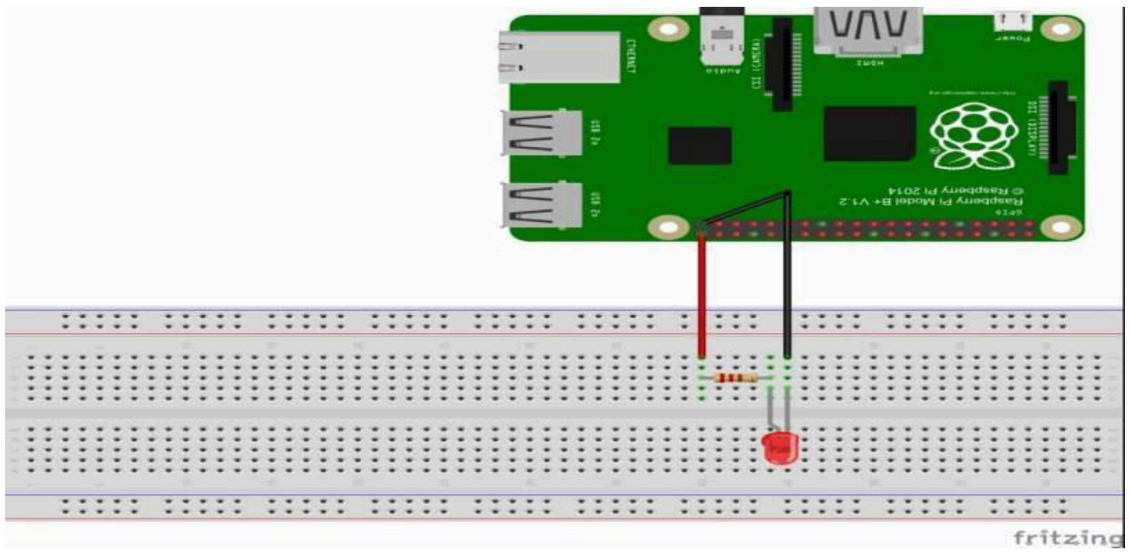
LED

Bread Board

HDMI Cable

Power Cable

Circuit Explanation:



As shown in the circuit diagram we are going to connect an LED between PIN11 and PIN6(GROUND) so to limit the current we are connecting a 1KΩ resistor in series with the LED.

Working Explanation:

Turn ON your PI and go to the desktop.

1. On the desktop, go the Start Menu and choose for the **PYTHON 3**, as shown in figure below.
2. After that, PYHON will run and you will see a window as shown in below figure.
3. After that, click on *New File* in *File* Menu, You will see a new Window open
4. Save this file as ***alarm.py*** on the desktop
5. After that write the program for ***alarm.py*** as given below and execute the program by clickingon " RUN" on ' DEBUG' option.

Code Explanation:

In the Python program, first we have imported two packages RPi.GPIO and Time. The package RPi.GPIO will helpus in controlling the GPIO Pins of the Raspberry Pi.

The first important function of the RPi.GPIO Module is the setmode(). Using GPIO.setmode(), we can select either GPIO Numbering of the Pins or Physical Numbering. By using GPIO.setmode(GPIO. BOARD**),** we are selecting thePhysical Numbering Scheme.

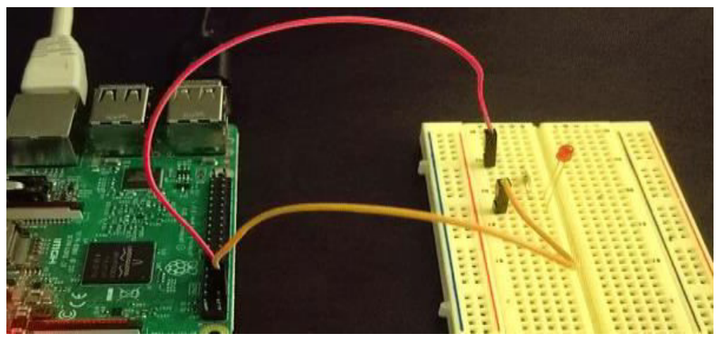
Create two functions setup( ) and blink The next function is the *setup(Ledpin,GPIO.OUT)*. This function will allow us to set the pin as either input (GPIO.IN) or as output (GPIO.OUT). In the program, I have set the ledPin as output by using **GPIO.setup(ledPin, GPIO.HIGH)**. setup( ) After setting the LED Pin as OUTPUT, now we need to set the state of this OUTPUT i.e. HIGH (GPIO.HIGH) or LOW (GPIO.LOW). For this, we need to use the function output(pin, state). So, in our program, we need to use **GPIO.output(ledPin, GPIO.HIGH)** for turning ON the LED and **GPIO.output(ledPin, GPIO.LOW)** for turning it OFF. Clean up GPIO.

Code:

```
import RPi.GPIO as GPIO
 import time
 LedPin = 11  # pin11
def setup():
 GPIO.setmode(GPIO.BOARD)  # Numbers GPIOs by physical location
GPIO.setup(LedPin, GPIO.OUT) # Set LedPin's mode is output
GPIO.output(LedPin, GPIO.HIGH) # Turn ON led
def blink():
 while True:  GPIO.output(LedPin, GPIO.HIGH) # led on
 time.sleep(1)
GPIO.output(LedPin, GPIO.LOW) # led off
 time.sleep(1)
 def destroy():
 GPIO.output(LedPin, GPIO.LOW) # led off
 GPIO.cleanup()
```

Output:

LED IS ON
LED IS OFF

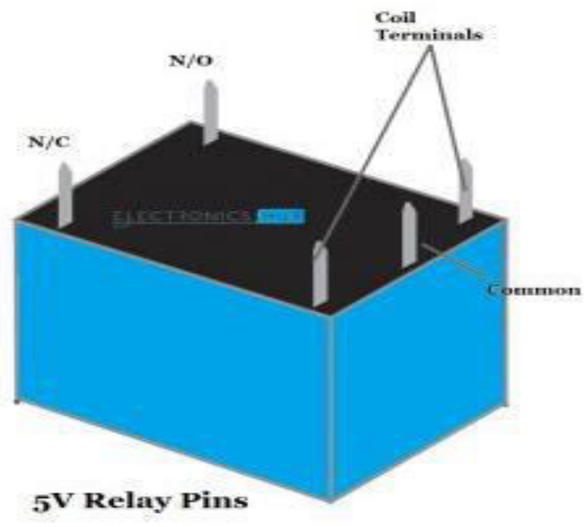


Week-7

Aim: switch on relay given using cron where relay contact terminals areconnected to a load

Relay

In layman terms, a relay is a switch. Technically speaking, a relay is an electromagnetic switch where a small control signal (usually from a microcontroller) at the input of the Relay will control a high voltage supply (usually AC mains).Since this is a Raspberry Pi based project, let us talk with respect to Raspberry



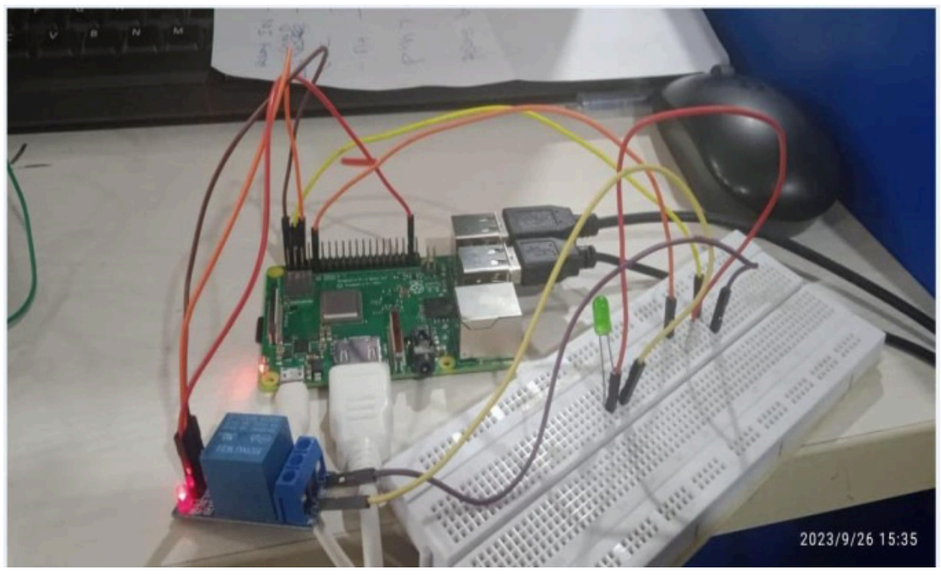
Components Required

- Raspberry Pi 3 Model B
- 5V Relay Module
- Two Small Incandescent Bulbs (for demonstration in the output)
- Connecting wires
- Power Supply
- Computer
- Working

The main concept behind this project is to understand the working and use of a relay and also control arelay using Raspberry Pi.

There is nothing special going in the project. All you need to do is to control the GPIO pins connected tothe Relay Module. If the GPIO Pin is made HIGH, the corresponding load will be switched ON.To turn OFF the load, make the GPIO pin LOW.

Connections:



Code:

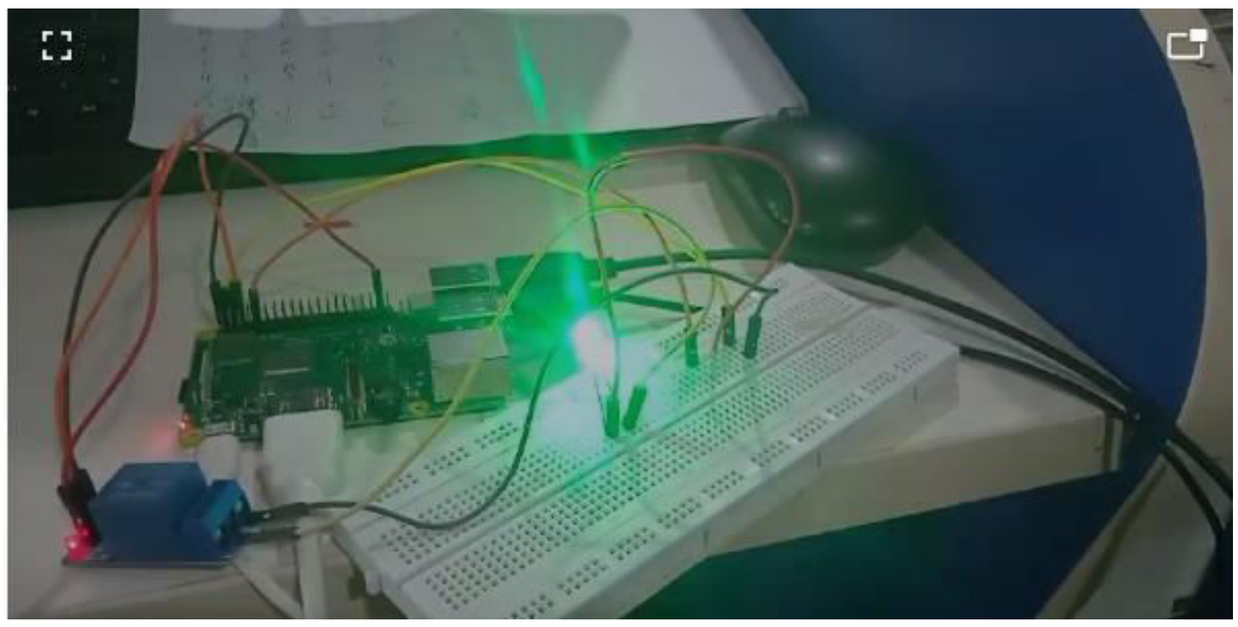
```
import time
import RPi.GPIO as GPIO
relay_ch = 26
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(relay_ch, GPIO.OUT)
GPIO.output(relay_ch, GPIO.LOW)
time.sleep(1)
GPIO.output(relay_ch, GPIO.HIGH)
GPIO.cleanup()
```

Output:

Relay

onLed

on



Week-8

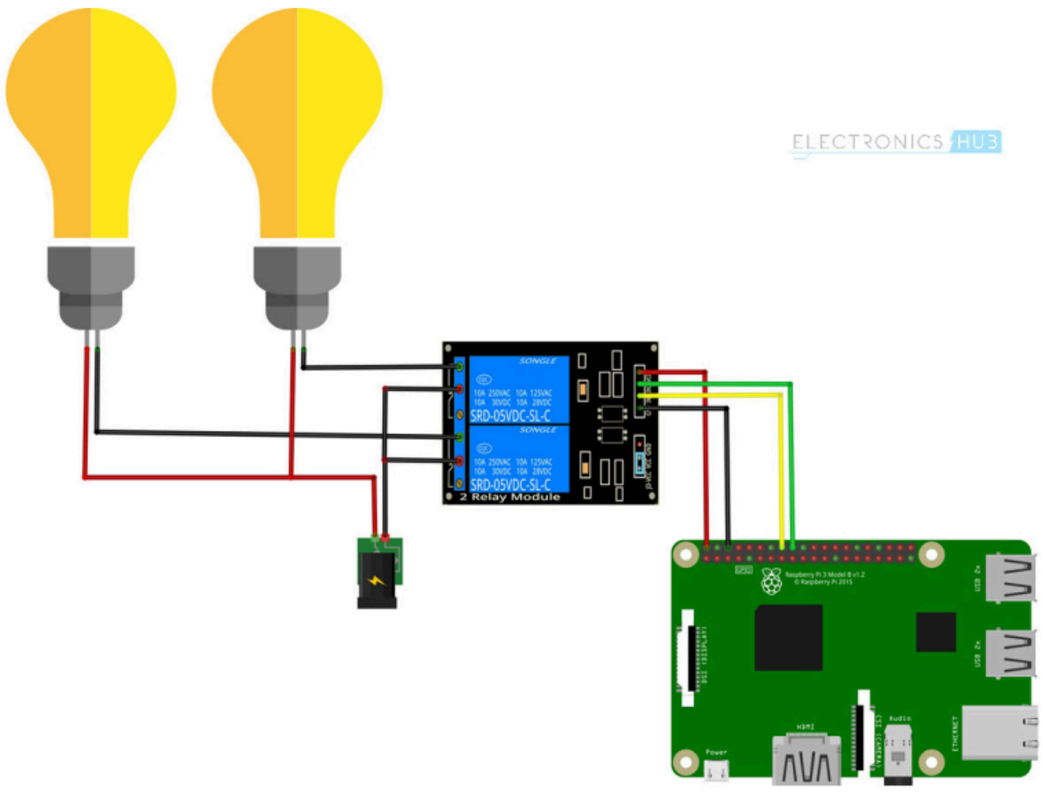
Aim: Get the status of a bulb at a remote place (on the LAN) through web.

Components Required

- Raspberry Pi 3 Model B
- 2-Channel Relay Module
- Two Small Incandescent Bulbs (for demonstration in the output)
- Connecting wires
- Power Supply
- Computer

Circuit Diagram

The following image shows the connections with respect to the project of How to Control a Relay using Raspberry Pi.



Working

The main concept behind this project is to understand the working and use of a relay and also control a relay using Raspberry Pi.

There is nothing special going in the project. All you need to do is to control the GPIO pins connected to the Relay Module. If the GPIO Pin is made HIGH, the corresponding load will be switched ON.

To turn OFF the load, make the GPIO pin LOW.

Program:

```
import RPi.GPIO as GPIO
import time
in1 = 16
in2 = 18
GPIO.setmode(GPIO.BOARD)
GPIO.setup(in1, GPIO.OUT)
GPIO.setup(in2, GPIO.OUT)
GPIO.output(in1, False)
GPIO.output(in2, False)
    try:
  while True:
   for x in range(5):
       GPIO.output(in1, True)
       time.sleep(0.1)
       GPIO.output(in1, False)
       GPIO.output(in2, True)
       time.sleep(0.1)
       GPIO.output(in2, False)
  GPIO.output(in1,True)
   GPIO.output(in2,True)
    for x in range(4):
   GPIO.output(in1, True)
   time.sleep(0.05)
```

```
        GPIO.output(in1, False)
        time.sleep(0.05)
    GPIO.output(in1,True)
        for x in range(4):
        GPIO.output(in2, True)
        time.sleep(0.05)
        GPIO.output(in2, False)
        time.sleep(0.05)
            GPIO.output(in2,True)
             except KeyboardInterrupt:
           GPIO.cleanup()
```

Output:

Bulb ON

Bulb OFF

LEAD1

Aim:To interface the ultrasonic sensor with the raspberry pi3 to determine the distance of the object from the sensor.

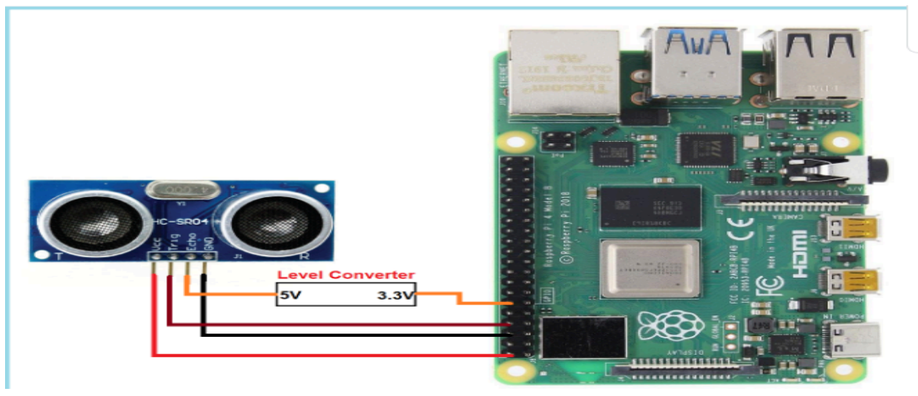
Hardware Required

- Raspberry Pi
- Ultrasonic Sensor
- Jumper wires

Software Required

- Raspberry Pi OS

Board connection:



- VCC pin: For Power supply.
- TRIG pin: This pin is the input pin and transmits the waves.
- ECHO pin: This pin is the output pin and it detects the reflected wave.

- GND pin: This pin is the ground pin.

coding

```
import RPi.GPIO as GPIO

import time GPIO.setmode(GPIO.BCM)

GPIO_TRIG = 11

GPIO_ECHO = 18

GPIO.setup(GPIO_TRIG, GPIO.OUT)

GPIO.setup(GPIO_ECHO, GPIO.IN)

GPIO.output(GPIO_TRIG, GPIO.LOW)

Time.sleep(2)

 GPIO.output(GPIO_TRIG, GPIO.HIGH)

Time.sleep(0.00001)

GPIO.output(GPIO_TRIG, GPIO.LOW)

while GPIO.input(GPIO_ECHO)==0:

start_time = time.time()

while GPIO.input(GPIO_ECHO)==1:

 Bounce_back_time = time.time()

 pulse_duration = Bounce_back_time - start_time  distance = round(pulse_duration * 17150, 2)

 print (f"Distance: {distance} cm")

GPIO.cleanup()
```

output

```
Distance Measurement In Progress
Waiting For Sensor
Distance: 19.38 cm
```

Lead 2

AIM: Blinking of led using Arduino Uno

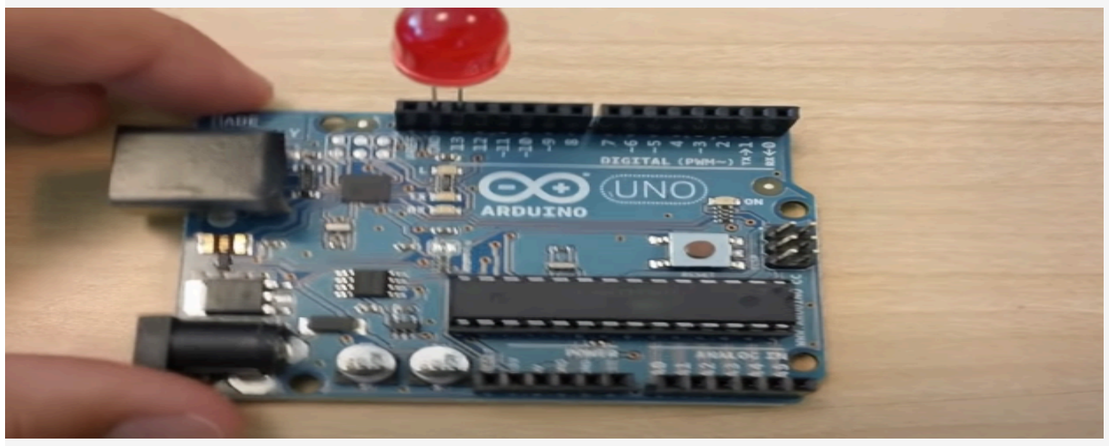
Components required:

ARDUINO UNO

LED

BREAD BOARD

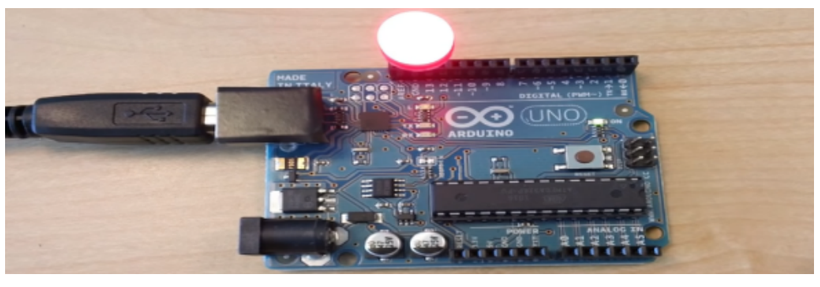
Circuit connection



Coding:

```
Int LED PIN=13;

void setup()

{

  pinMode(LED PIN, OUTPUT);

}

void loop()

{

  digitalWrite(LED PIN, HIGH);

  delay(1000); // Wait for 1000 millisecond(s)

  digitalWrite(LED PIN, LOW);

  delay(1000); // Wait for 1000 millisecond(s)

}
```

Output:



Output:

VIVA QUESTIONS:

Q1. What are the main parts of IoT systems?

Answer:

IoT system consists of three main parts:

1. Sensors
2. Network connectivity
3. Data storage applications.

Q2. What are security concerns related to IoT?

Answer:

Data security and privacy are major concerns related to IoT. These devices are vulnerable to hacking and cloud endpoints could be used by hackers to attack servers. Software developers and device designers have to ensure adequate security and privacy measures.

Q3. Explain the IoT protocol stack.

Answer:

IoT has 4 protocol layers:

1. Sensing and information: Includes various smart sensor devices based on GPS, RFID, Wi-Fi etc.
2. Network connectivity: Layer is based on a wired and wireless network such as WLAN, WMAN, Ethernet, optical fiber and more.
3. Information processing layer
4. Application layer

Q4. Explain the basic architecture of the IoT network.

Answer:

IoT has three main parts namely sensors, network connectivity and data storage applications. Sensors either communicate directly with the central server for data storage or communicate via gateway devices. A gateway can handle various wireless interfaces that' s why one gateway can handle multiple technologies and multiple sensors.

Q5. What is the top 5 Machine to Machine (M2M) applications in the world?

Answer:

They are as follows:

1. Asset tracking and monitoring in some form or some other (stolen automobiles, fleet, construction system, and many others seems to be the biggest.
2. Insurance telematics is huge as it gives insurance groups the possibility to cut the threat and force higher/extra appealing pricing.
3. Utilities/automatic meter reading/clever grids – plenty of regulation and funding into this in the intervening time. There a plenty of countrywide solutions because the requirements and business case are driven in very numerous ways.
4. Automotive is also very big and is driven by consumer's demand.
5. mHealth is also present in small scale.

Q6. What is the difference between a wireless sensor network (WSN) and the Internet of Things(IoT) network?

Answer:

WSN: Wi-Fi sensor community is the foundation of IoT packages. WSN is a network of motes, fashioned to look at, to take a look at or to monitor bodily parameters of desired utility.

e.g. motes deployed in agricultural land, screen temp-humidity or maybe soil moisture, who gathers statistics and ideal statistics analysis procedure consequences approximately crop yields-high quality or amount.

IoT: IoT is a community of bodily objects managed and monitored over the internet. Now just as win, in its application, you will stumble upon the monitoring of physical parameters. But preferred results are little different. IoT is about M2M, it's far greater than bringing smartness into daily gadgets.

e.g. device hooked in your thermostat monitors surrounding temperature and adjust

Q7. How can you measure power consumption used by Raspberry Pi?

The power consumption for Raspberry Pi varies from model to model. However, it is relatively low than a complete computer system. It can be measured using a multimeter.

Q8. Are we likely to confront any overheating problems with Raspberry Pi?

No, Raspberry Pi is free of overheating issues and does not require a fan for cooling too. It is designed in such a way that it can withstand about 100 degree Celsius. This will help us finish projects in the safest way possible.

Q9. Why do you think we must use the Raspberry Pi?

Raspberry Pi is ideal for creating projects. We can use it when we need computers but not advanced processing and hence save space. Raspberry Pi can be used as an alternative to a desktop or even use it as a photography tool at a low cost. There are options to create Google Home, light sensing switch and a retro gaming console. The Raspberry Pi has a wide range of uses that are essential in companies that idealize innovation.

Q10. How is Raspberry Pi used in IoT?

Raspberry Pi can be used as a platform to develop many Internet of Things project. It is simple to use Raspberry Pi because it uses Linux OS in a small card like a computer. It is also available in low cost. IoT, on the other hand, refers to a network of devices and software that help us to connect and exchange data. Raspberry Pi is easy to set up, so it is recommended for IoT.

Q11. What are the generations of Raspberry Pi available?

Starting from Raspberry Pi Zero, there have been many improvements in the versions. Raspberry Pi 1, Pi 2, Pi 3 are available and A, A+, B indicate the power consumption.

Q12. Tell about any interesting project with Raspberry Pi.

Recently, the Astro Pi project is accelerating the use of Raspberry Pi in space stations. This is aiming to write codes in space using Raspberry Pi.

Q13. How is Raspberry Pi different from Arduino?

Raspberry Pi uses Linux OS and is a general purpose microcomputer. It is capable of running multiple programs at a time, while the Arduino is a simple microcomputer that is capable of running one program only.

Q14. What kind of projects have you done with Raspberry Pi?

A lot of innovative projects can be done using Raspberry Pi. Like using cameras, using it like a media center and more.
You can explain about your project in detail and how your idea will benefit the world.

Q15. What is the language used by Raspberry Pi?

Raspberry Pi uses Python as its official programming language. Python is one of the most user-friendly programming language used. It is also preferred by many companies for system development. Raspberry Pi helps us to quickly release our projects with Python. Raspberry Pi is preloaded with Python which has comprehensive syntax.

Q16. Can Raspberry Pi be used as a server?

Yes, like we use Raspberry Pi for the desktop we can use it for the server as well. Raspberry Pi allows us to utilise it like a Web server that can help us create a simple site or even store data in the cloud. This can be accessed at any time. We can develop new templates without paying any monthly hosting fee. Raspberry Pi is thus an affordable web server too. This is very suitable for small business and for people who want to learn easy web languages.

Q17. What are GPIO Pins used in Raspberry Pi boards?

GPIO is the acronym for General Purpose Input Output Pin. They are more like switches on the board that gives an output voltage when high and give no voltage when turned to low. This is used in the Raspberry Pi boards to make an interface between the Raspberry Pi and all the components of the board. This enables multiple interactions and makes internal properties of devices easily available on board.

Q18. How does the Raspberry Pi work?

The memory card slot is used for inserting an SD card that acts as the hardware of Raspberry Pi. The USB port, HDMI port, and the audio/video port help us connect with a monitor, TV or any other device. This is how Raspberry Pi is capable of working anytime and anywhere. The processor gives the correct speed for running our computer programs all the time.

Q19. What are the different components of a Raspberry Pi board?

The Raspberry Pi board contains a 700 or 900 MHz processor with a minimum memory provision of 128 MB. It has an additional slot for memory card too. There is a graphics set up, USB port for connecting keyboard or mouse. Raspberry Pi comes along with an audio or video output option to connect your monitor. There is an HDMI port too for connecting with TV.

Q20. What is Raspberry Pi?

Raspberry Pi is a microcomputer that we can connect with any keyboard, mouse or a monitor. Raspberry Pi is able to perform all tasks like creating word documents, spreadsheet, browsing, coding, games and much more. It is a tiny Linux computer that is used by many companies for performing all computer-based activities, learning and electronic projects. The Raspberry Pi uses a 32 bit ARM processor and was first developed by Raspberry Pi Foundation. The low cost makes it more popular among young people.