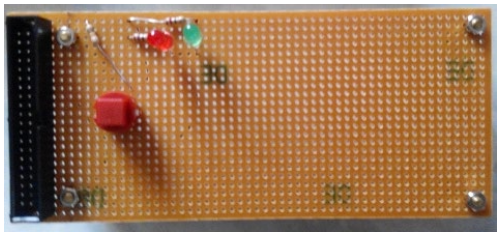
# Python Programming using RPi & I/O board

**Lab 3**

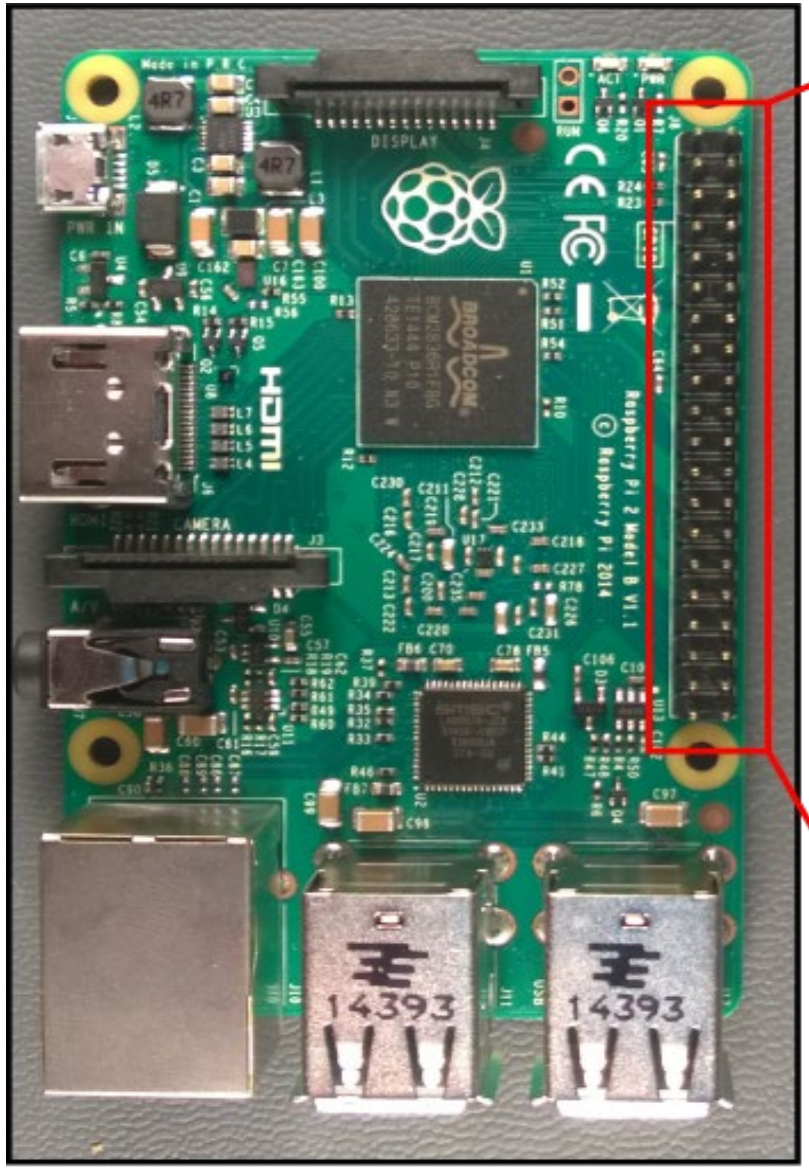


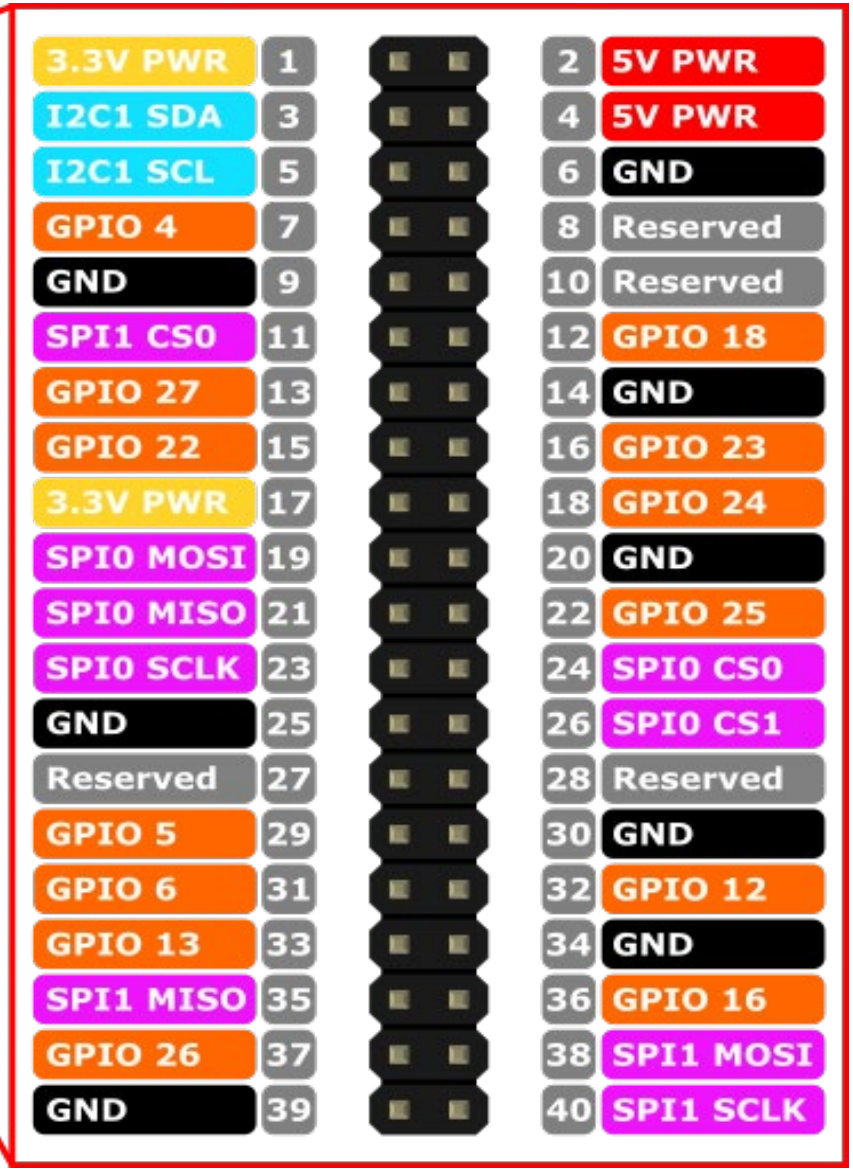
# Learning outcomes:

## At the end of this Lab 3, you should be able to:

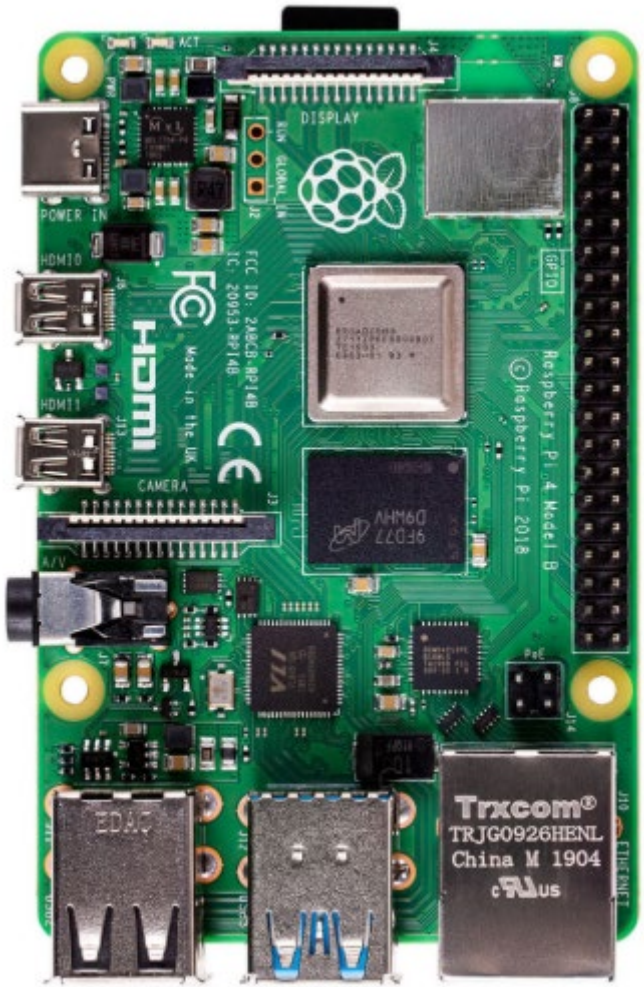
- identify RPi GPIO pins
- interface I/O board
- control I/O peripherals
- complete examples

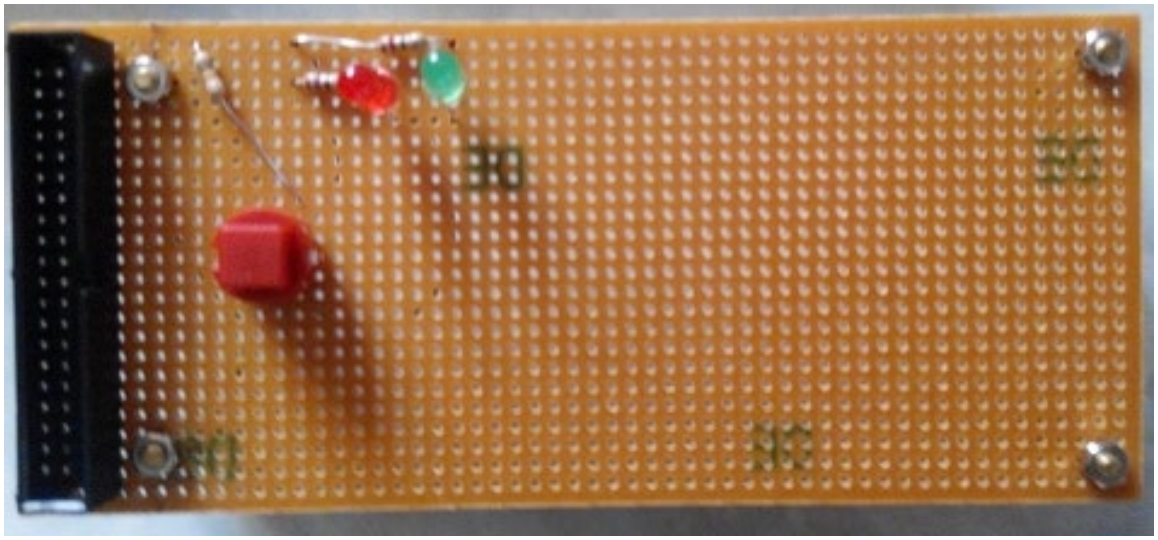
# RPi GPIO pin diagram (40 pin)





# RPi and I/O board:



# General Purpose Input Output (GPIO)

Using Python on the RPi opens up the opportunity to connect to the real world through the Pi's GPIO pins. This can be done with the RPi GPIO library. It is preinstalled on recent Raspbian images:

To control the GPIO pins you'll need root access, so run sudo python

In your Python script, import the GPIO module, set the board mode to that of your preference, set up the pins you want to use:

```
import RPi.GPIO as G

G.setmode(G.BCM)  # set board mode to Broadcom

G.setup(27, G.OUT)  # set up pin 27
G.setup(22, G.OUT)  # set up pin 22

G.output(27, 1)  # turn on pin 27
G.output(22, 1)  # turn on pin 22
```

# Example G1: Turn ON Red LED

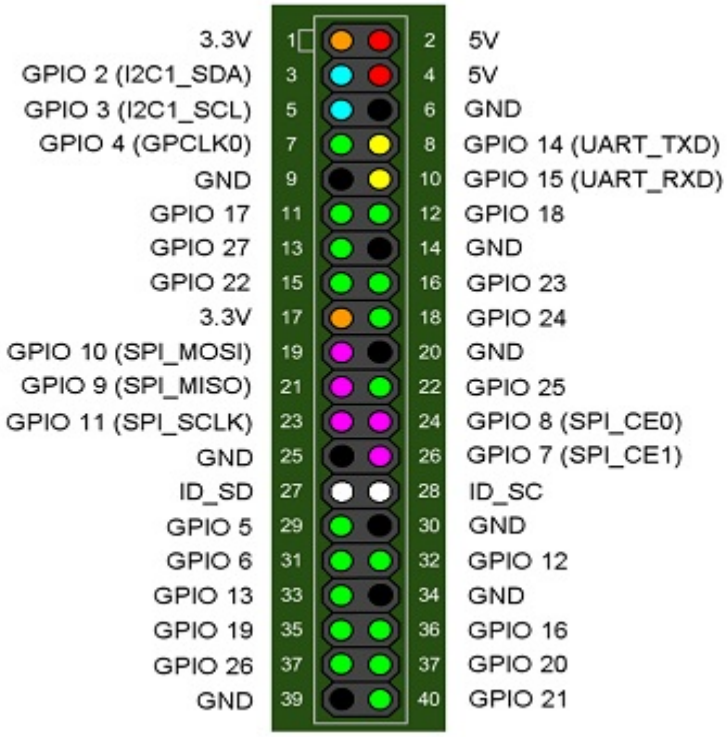
**To control GPIO, you need an Adminstrator rights.**

**Launch LXTerminal**

**pi@raspberry ~ $ sudo idle**

**A new shell will be launched, then go to File >New Window and type the below code:**

```
import RPi.GPIO as G    # Import GPIO library
G.setmode(G.BCM)        # Use board pin numbering
G.setup(27, G.OUT)      # Setup GPIO Pin 27 to OUT
G.output(27, True)      # Turn on GPIO pin 27
raw_input('Press to exit')
G.cleanup()
```




Save the file with any file name (for example **G1.py)**

**Run > Run Module**

Then observe the Red LED status

# Example G2: Turn OFF Red LED

```
import RPi.GPIO as G
G.setmode(G.BCM)
G.setup(27, G.OUT)
G.output(27, False)  # you can also use 0 or OFF instead of False
raw_input('Press to exit')
G.cleanup()
```

Save the file with **G2.py** name
**Run > Run Module**
Then observe the Red LED status

Modify the above program to turn ON Green LED along with the Red LED.

The Green LED is connected to GPIO **22**

# Example G3: Blink RED & GREEN LEDs

```
import RPi.GPIO as G
import time     # Import 'time' library. Allows us to use 'sleep'
G.setwarnings(False)  # to suppress warning messages

G.setmode(G.BCM)
G.setup(27, G.OUT)
G.setup(22, G.OUT)

G.output(27, True)
G.output(22, True)

time.sleep(3)   # time in seconds

G.output(27, False)
G.output(22, False)

time.sleep(3)

Raw_input('Press to exit')
GPIO.setwarnings(False)

G.Cleanup()
```

# Example G4: Using function (Blink RED LED)

```
import RPi.GPIO as G
import time
G.setmode(G.BCM)
G.setup(27, G.OUT)

#Define a function named Blink()
def Blink(numTimes,  speed):
        for i in range(0,numTimes):            # Run loop numTimes
            print (“Iteration “ + str(i+1) )  # Print current loop
            G.output(27,True)                  # Turn ON red LED on GPIO 27

             time.sleep(speed)   # Wait
             G.output(27, False)  # Turn OFF
             time.sleep(speed)  # Wait

        print (“Done”)     # When loop is complete, print "Done"
        G.cleanup()

iterations = raw_input(‘Enter total number of times to blink: ‘)
speed = raw_input(‘Enter length of each blink(seconds): ‘)

# Start Blink() function. Convert input from strings to numeric data & pass to Blink()
Blink(int(iterations),float(speed))
```

# Example G5: Using while (Blink RED LED)

```
import RPi.GPIO as G
import time
try:
        # Set up the G channels
         G.setmode(G.BCM)
         G.setup(27, G.OUT)



        t = 1.5
        while True:
              G.output(27, True)
              time.sleep(t)
              G.output(27, False)
              time.sleep(t)
except KeyboardInterrupt:
        G.cleanup()
```

**Note: Press Ctrl+C to stop the program.**

# Example G6: Input (GPIO 4)

```
import RPi.GPIO as G
import time
try:
        # Set up the GPIO channels
         G.setmode(G.BCM)
         G.setup(27, G.OUT)
         G.setup(22, G.OUT)
         G.setup(4, G.IN)

         t = 1.5
         while True:
              while(G.input(4) == 0):
                  pass
                  print("You pressed the button")
                  G.output(27, True)
                  G.output(22, True)
                  time.sleep(t)
```

# Example G6: Input (GPIO 4) continued…

```
# Note the indentation, type exactly below the time.sleep(t)
                G.output(27, False)
                G.output(22, False)
                time.sleep(t)

except KeyboardInterrupt:
        G.cleanup()
```

# Example G6a: Design a counter: Try out!

- Using the 2 LEDs and a switch, write a program which counts the number of times you have pressed the switch and display the count on the screen.
- Turn ON both the LEDs when the switch is pressed
- Turn OFF  both the LEDs when the switch is released.