Name - Nirant Deore
TE (computer)
Div - A
Roll NO-31
Batch - TA2

Saathi

ESIOT

Date __/__/__ Assignment No-1

- **Aim :-** To study Raspberry Pi , Beagle board, Arduino & other micro-controller.

- **Theory :-** • **Study of Rasberry Pi :-** It is a series of small single board computers developed in United Kingdom by Raspberry Pi foundation to promote the teaching of basic computer science in schools & in developing countries. A Raspberry Pi zero with smaller size & reduced i/p & o/p & general input output capabilities was released in Nov 2015 for US $ 5 on 28 feb 2017 the Raspberry Pi zero was launched which is like Raspberry Pi zero with Wifi & Bluetooth fos US $10 processor speed ranges from 700 MHz to 1.2 GHz for Pi 3 onboard memory ranges from 256 MB to 1 GB RAM. The boards have 1 to 4 USB ports for video o/p HDMI & composite video are supported with std 3.5 mm phone jack for audio o/p.

• **Study of Beagle Board :-** The beagle board is low power single board computer produced by texas instruments in association with Digi key & network element 14. The Beagle board is also designed from open source software developement in mind. The board was developed by small team of Engineers as an educational board that could be used in colleges. The board was designed using cadence or CAD for schematics & cadence Allegro for PCB manufacturing no. stimulation software is used

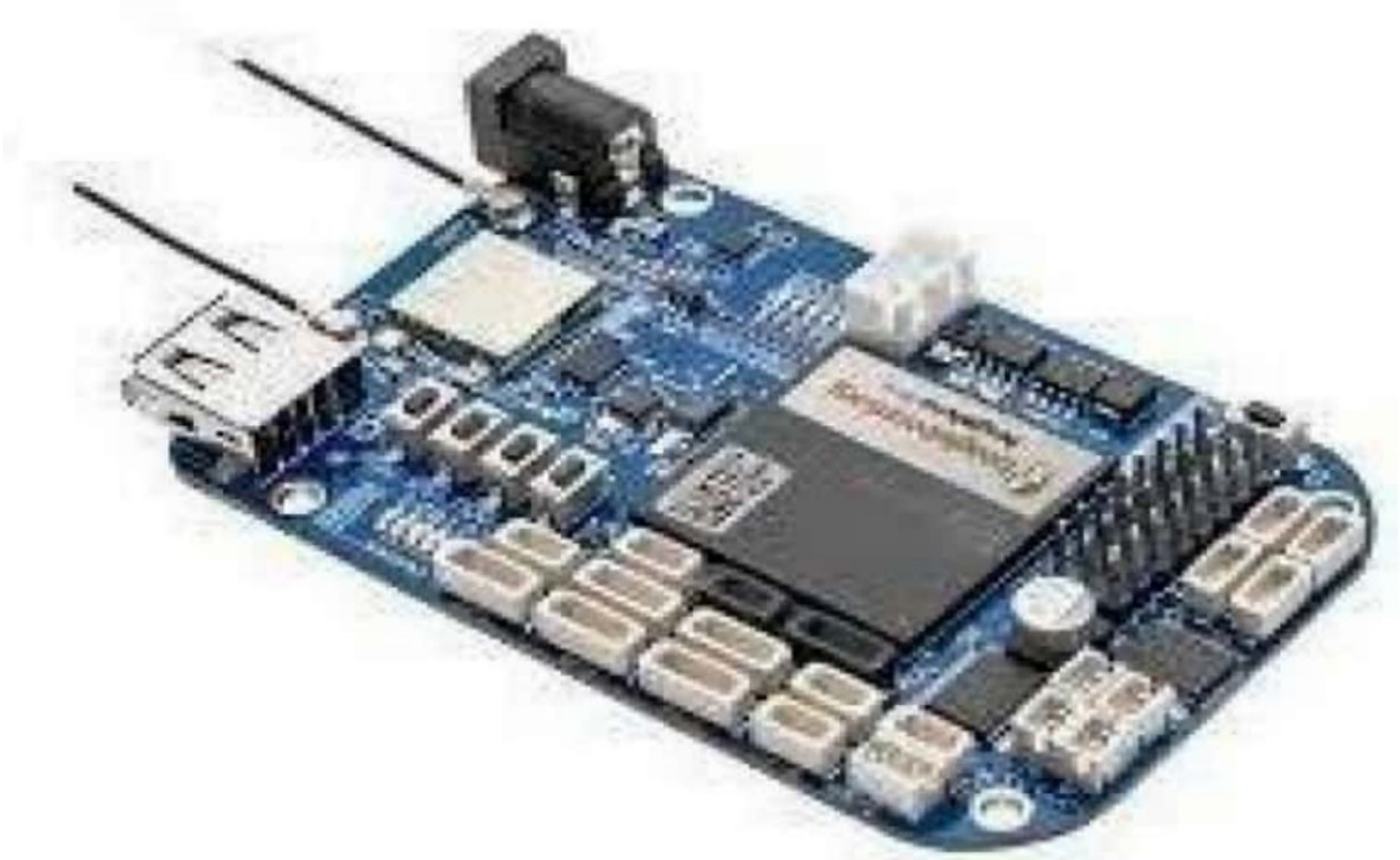
Page No.

Scanned with CamScanner

Raspberry Pi 3 Model B

Fig.1. Raspberry Pi3 Kit

Scanned with CamScanner



Beagle Board Fig.2.

Scanned with CamScanner

Arduino Kit Fig.3.

Scanned with CamScanner

- Study of Arduino:- The Arduino project started at the Interaction design Institute IVREA (IDII) at that time students used Basic stamp micro-and at a cost of $100 a considerable expense for many students.

The initial arduino core team consisted of Massimo Banzi, Dried Curatilles. In october 2016 febrico musto, Arduino's former CEO, secured a 50% ownership of company. In April 2017 wired reported that musto has fabricated this academic record on his company's website personal linked in accounts & even on Italian business documents. Musto was until recently listed a holding a PHD from MIT. In some cases his bios has claimed an MBA from New York University. wired reported that neither university has any record of Musto's attendance & musto later admitted interview that he never earned those degree's.

ARM recognized independence as a core value of Arduino without any lock with ARM architecture. Arduino intends to continue to work with all technology vendors & architecture.

- Conclusion:- Thus, we have studied the history of Raspberry Pi, beagle board & Arduino

Scanned with CamScanner

Name - Nirant Deore
TE (Computer)
Div - A
Roll No - 31
Batch - TA2

Saathi

ESIOT

Date __/__/__ Assignment No - 2

- **Aim:-** Study of different O.S. for Raspberry Pi understanding the process of os installation on Raspberry Pi.

No matter how good & powerful the hardware of Raspberry Pi is without an O-S it is just piece of silicon, fiberglass & few other semi-conductor materials. There are several different os. for Raspberry Pi.

i) **Rasphian:-** It is most popular linux based O.S. for Raspberry Pi. It is an open source o.s. based on debian which has been motivated modified specifically for Raspberry Pies, free & open source operating system & often comes with Raspberry Pi Kit. It is an official O.S. of Raspberry Pi foundation.

ii) **Pidora:-** After waiting for long Raspberry Pi users are finally getting an optimized version of fedora, the Pidora, to replace the current Rasbian OS. The News channel excitement among Raspberry Pi community who are finally getting the opportunity to explore fedora Remix for Pi ended up as failure. However the center for developement of open technology the authority group behind pidero is confident that the Raspberry Pi community would newly optimized os.

iii) **Arch Linux:-** It is an excellent choice for many reasons one of the greatest advantag

Scanned with CamScanner

is its simplicity in approach and attitude Arch gives you the ability to build your system from ground up, including only the software you actually need. Arch has now finished its transition to system D from initscripts.

iv) OSMC:- OSMC (Open source media centre) is a free & open source media player based on linux founded in 2014, OSMC let you playback media from your local network attached storage and internet. OSMC is leading media centres in terms of feature set & community & is based on Kodi project

v) Retro PIE:- It allows you to turn your Raspberry PI into a retro gaming machine. It platform developed on the base of Raspbian Emulation station, Retropie enable you to pay PC games with minimum setup.

vi) RISC OS:- It is deg designed by Aron computers ltd cambridge England. It is first release in 1987. It specifically designed to run on ARM chipset. It is fast compact & efficient.

vii) Firefox OS:- Firefox OS which is more associated with being a linux kernel based open source O.S. primarily designed for smartphones & tablet computers. It was

Scanned with CamScanner

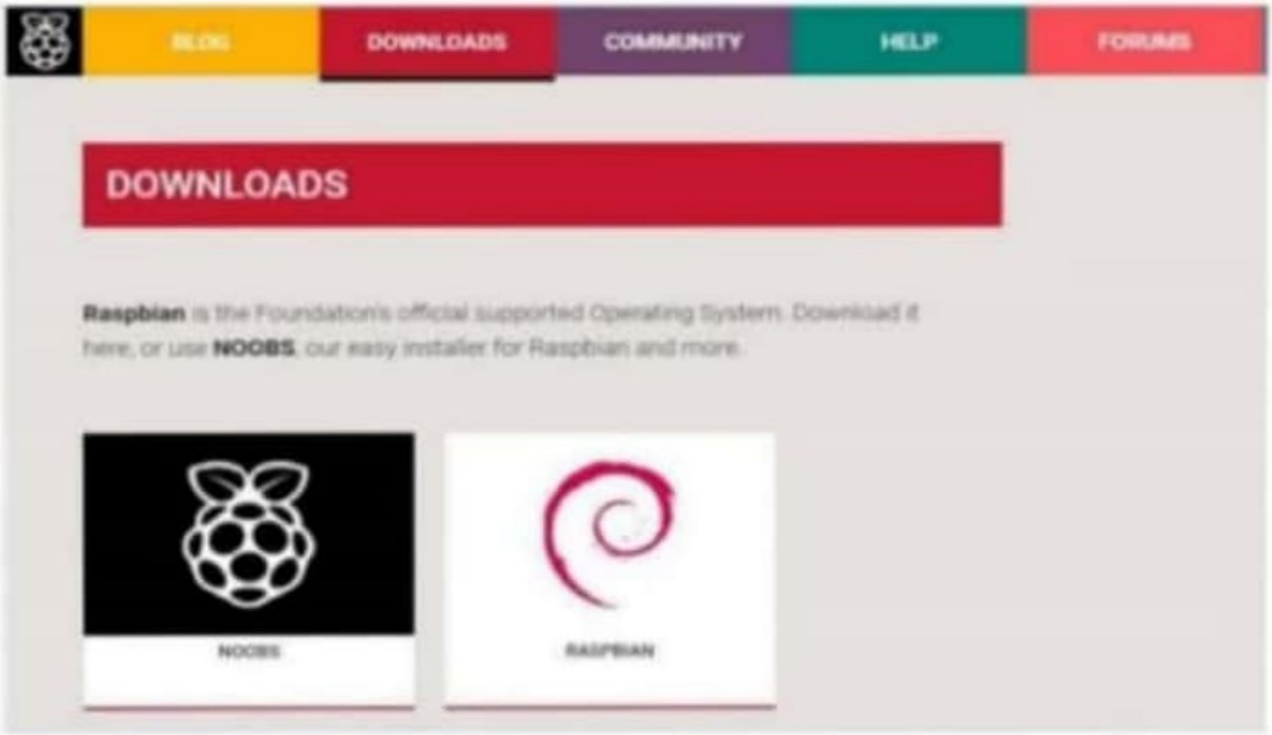
primarily designed as a community based alternative system utilizing open standards & HTML5 applications.

viii) KALI Linux:- It is debian based seculity auditing linux distribution. It is specially designed for digital forensics & penetratn testing. It is maintained & funded by offensive security limited.

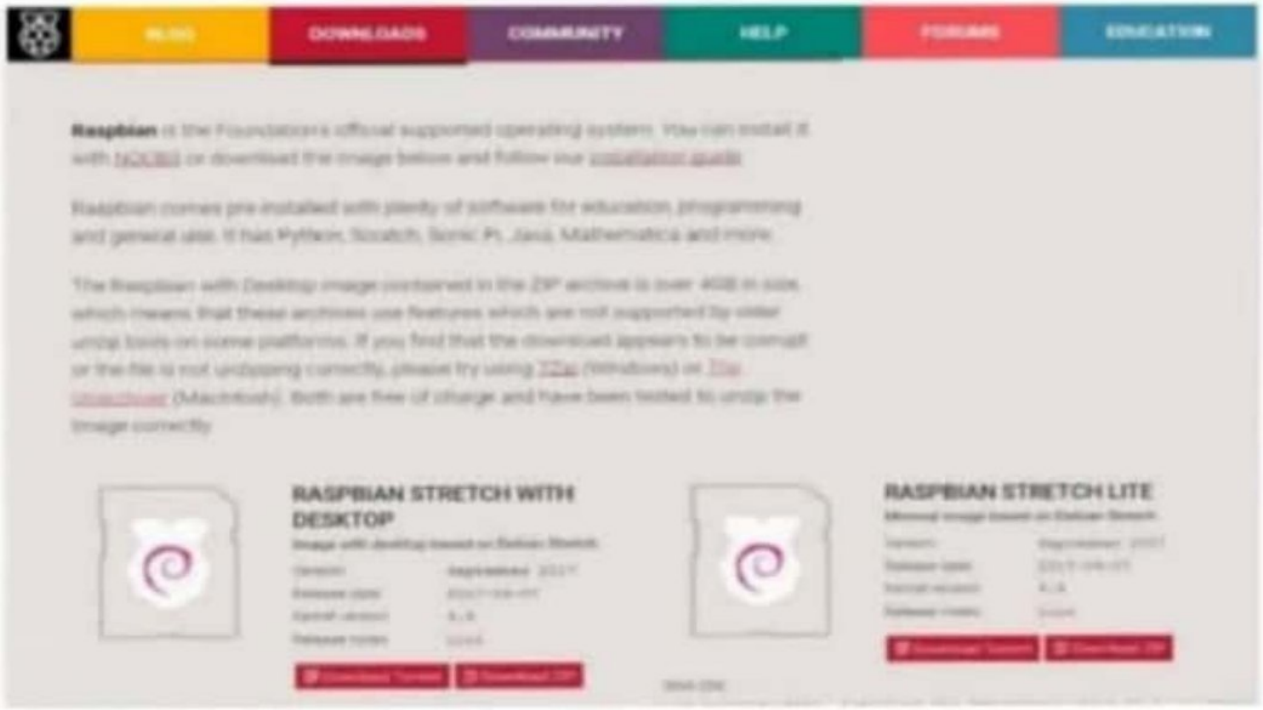
- Conclusion:- Thus, we have studied installatn for various Os in Raspberry Pi.

Scanned with CamScanner

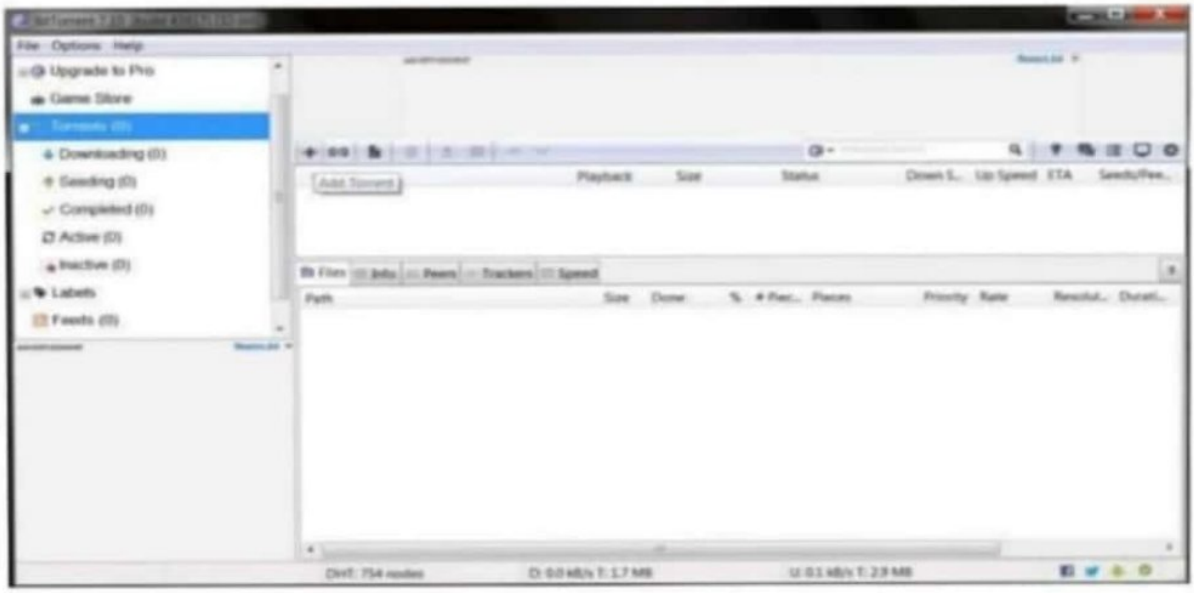
3.



4. Click on the "RASPBIAN" option.

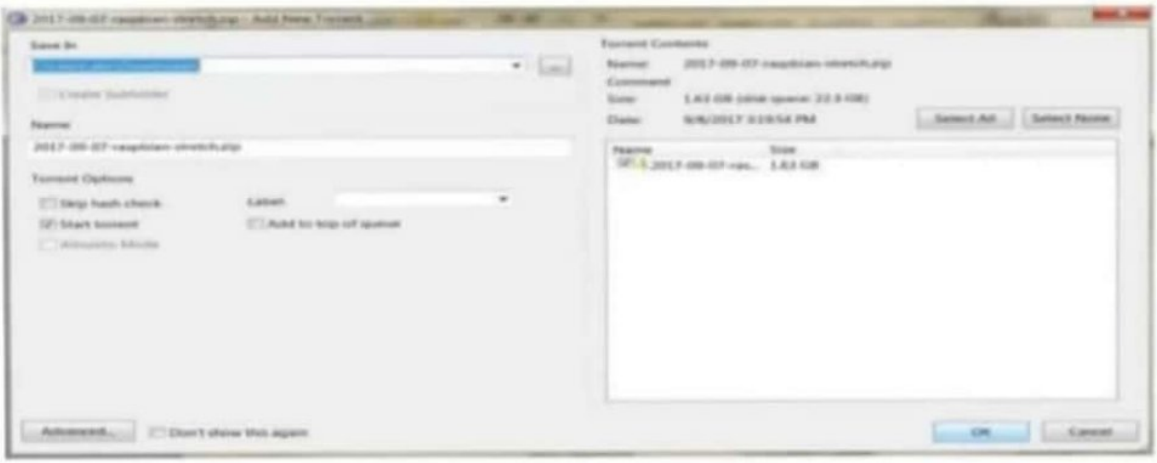


Scanned with CamScanner



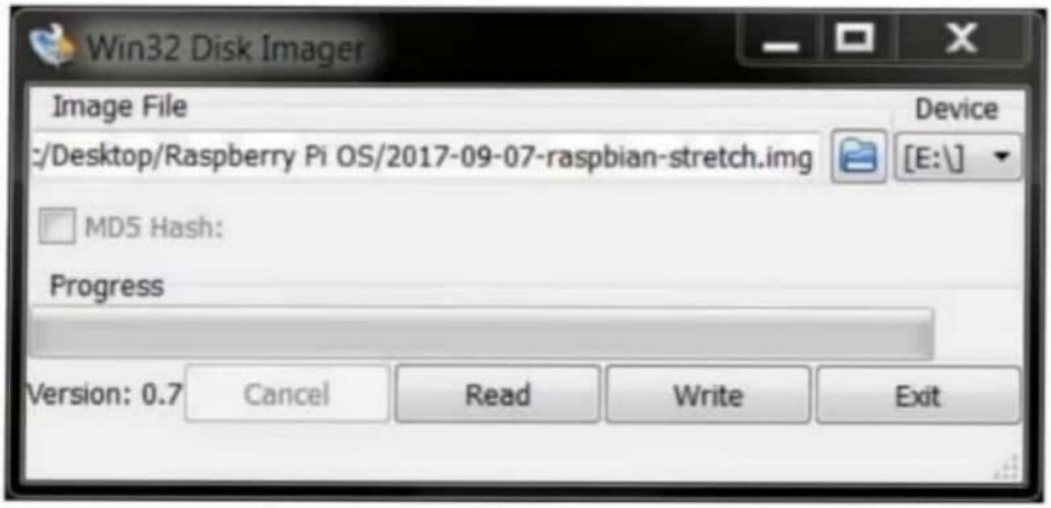
12. Here select the path of downloaded "Torrent file".

13. After selecting the torrent file, following window appears. Here click on OK



Scanned with CamScanner

18. Open the unzipped file in the "Image file" option byselecting the path from the Blue icon. The selected path is shown in the below image.
19. Now plug-in the SD card reader having SD card inside it, in the USB port of your PC.
20. Ensure that your SD card reader is having the same drive which is shown in the Device option (near the blue icon)



21. After ensuring that the "Image file path" and the "Device" are selected correctly, now click'Write' button to write the image on the SD card.
22. After this the following window appears.

Scanned with CamScanner

Scanned with CamScanner

Name- Nirant Deore
TE(Computer)
Div-A
Roll No-31
Batch-TA2.

Saathi

Date __/__/__ ESIOT Assignment No-3

- **Aim:-** Study of connectivity & configuration of Raspberry Pi / Beagle board circuit with basic peripherals, LED understanding GPIO & its use in program.

- **Theory:-** Connectivity & configuration of Raspberry Pi Guides to configure Raspberry Pi.

i) **Raspi-config:-** The raspberry Pi configuration in Raspbian allowing you to easily enable features such as camera & change your specific setting such as keyboard layout.

ii) **Config.txt:-** The raspberry Pi configuration file.

iii) **Wireless:-** Configuration to a Pi as a wireless access point using Raspberry Pi 3 & PI zero was In built wireless connectivity or USB wireless Dongle.

iv) **Audio config:-** Switch your audio output betn HDMI & the 3.5mm jack.

v) **Camera config:-** Installing & shifting setting up Raspberry Pi camera board.

vi) **External Storage Config:-** Mounting & setting up external storage on Raspberry Pi.

Scanned with CamScanner

vii) Localisation:- Setting up your pi to work in local language.

viii) Default Pin Config:- Changing default pin states.

ix) Device tree config:- Device trees, overlays & parameters.

x) Kernel Command Line:- Linear Linux kernel accepts a command line of parameters during boot.

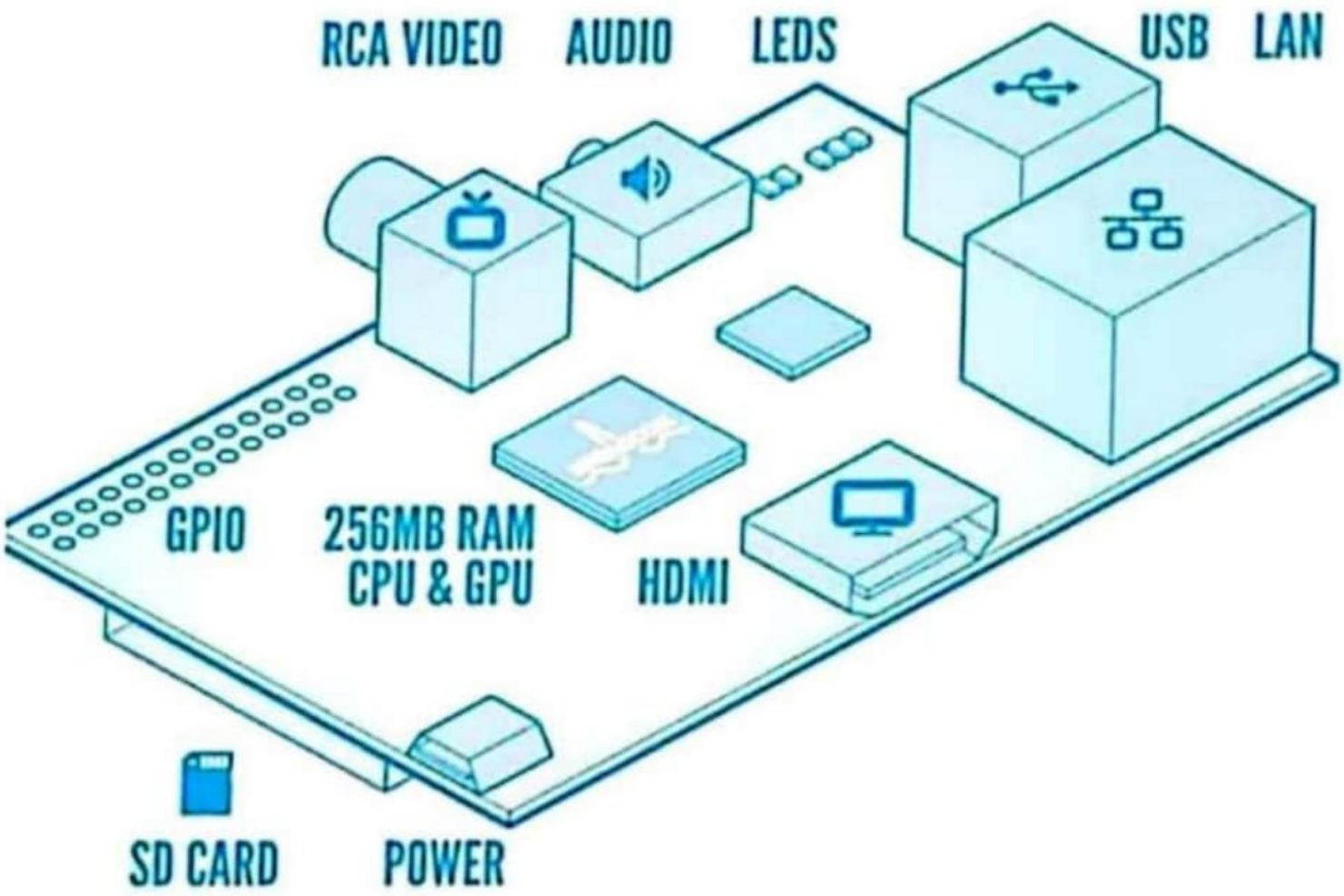
xi) VART configuration

xii) Screensaver

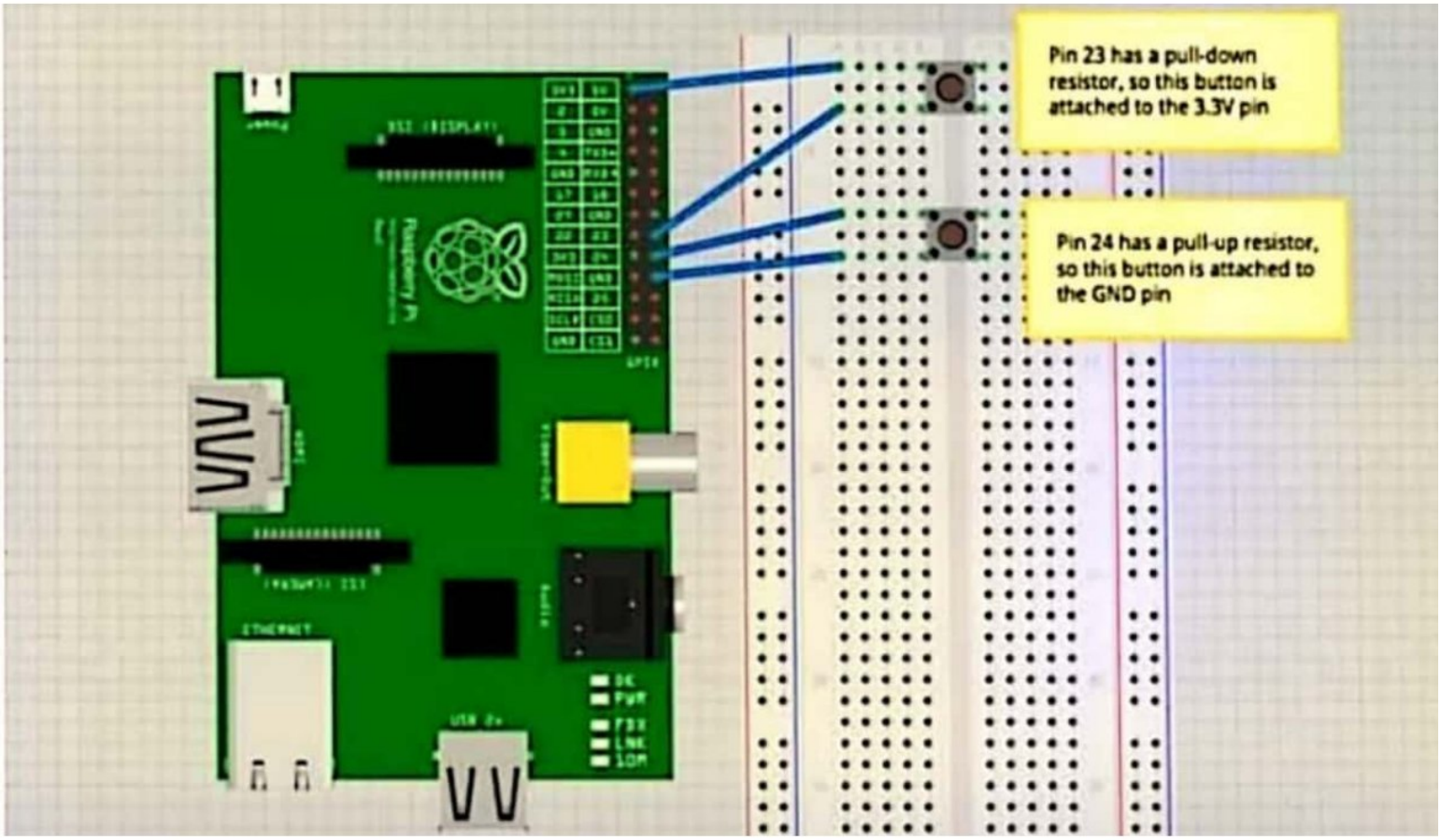
xiii) Wireless Access Point

- Connectivity of Raspberry Pi:- Connectivity is truly superb for such tiny device especially on be version of Raspberry Pi There are 2 USB 2.0 ports that can be used to hookup peripherals or adapters & this can be further expanded with power hub. Its worth nothing that both parts already share the bandwidth of single channel to system bus.

Scanned with CamScanner



Scanned with CamScanner



Scanned with CamScanner

- GPIO mode :- This option specifies that you are referring to pins by number of pin the plug and in the middle of diagrams below.

The GPIO .BCM option means that you are referring to the pins by the board can SOC channel number, this are the numbers after "GPIO" in green rectangles around outside of below diagrams.

i) Unfortunately the BCM numbers changed between versions of Pi1 model B.

ii) The Raspberry Pi zero, Pi2B & Pi3B uses the same numbering as B+.

- Resistor:- You must always use resistors to connect LEDS upto the GPIO pins of the Raspberry Pi can only supply a small current. The LED's will want to draw more & if allowed they will burn out the Raspberry Pi. Therefore putting resistors in the circuit will ensure that only this small current will flow & Pi will not be damaged.

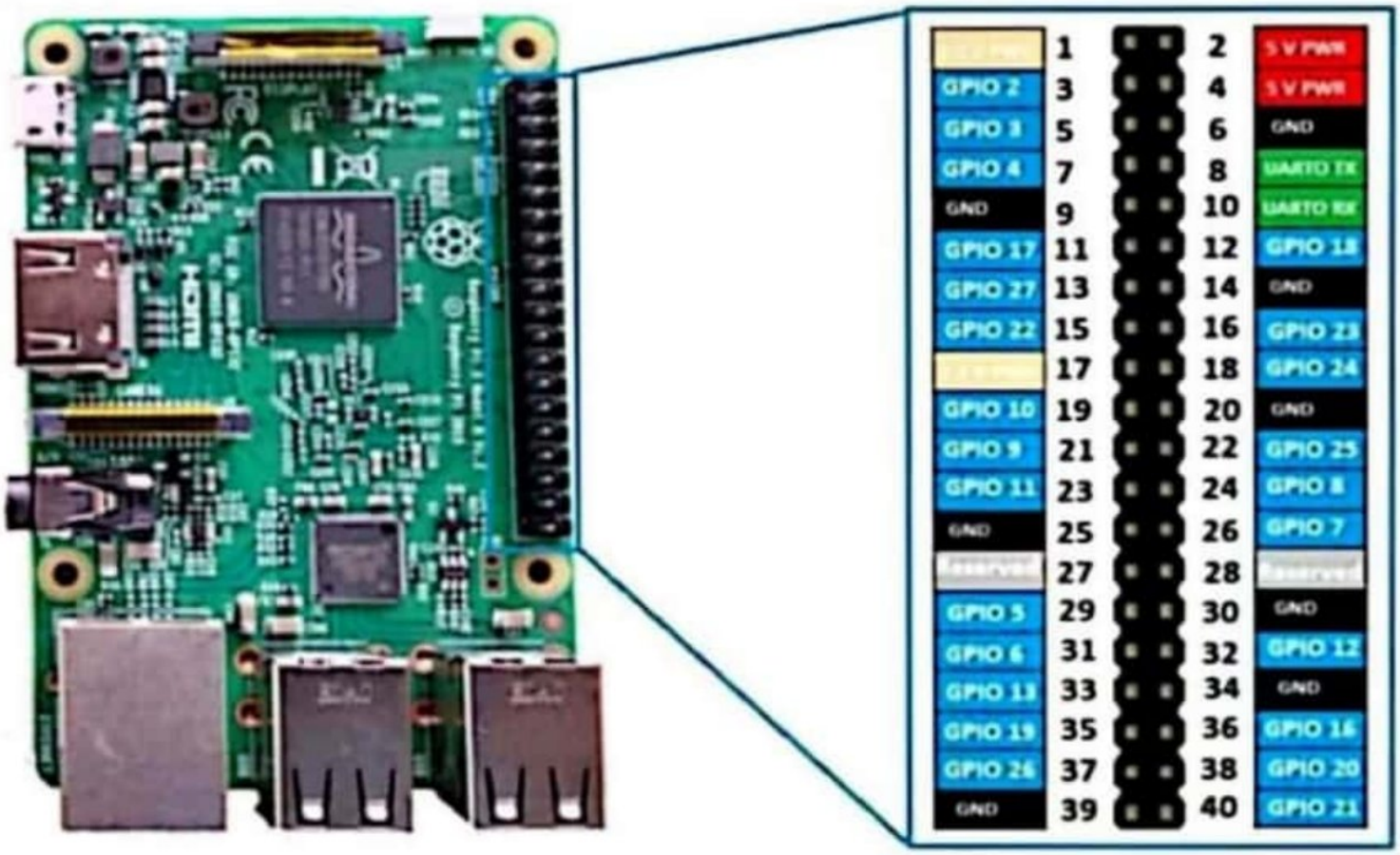
- Jumper wires :- This are used in bread boards to jump from one connection to another.
  - The ones you will be using in this circuit have different connectors on each end.
  - The end with the pin will go into the bread board.

Scanned with CamScanner

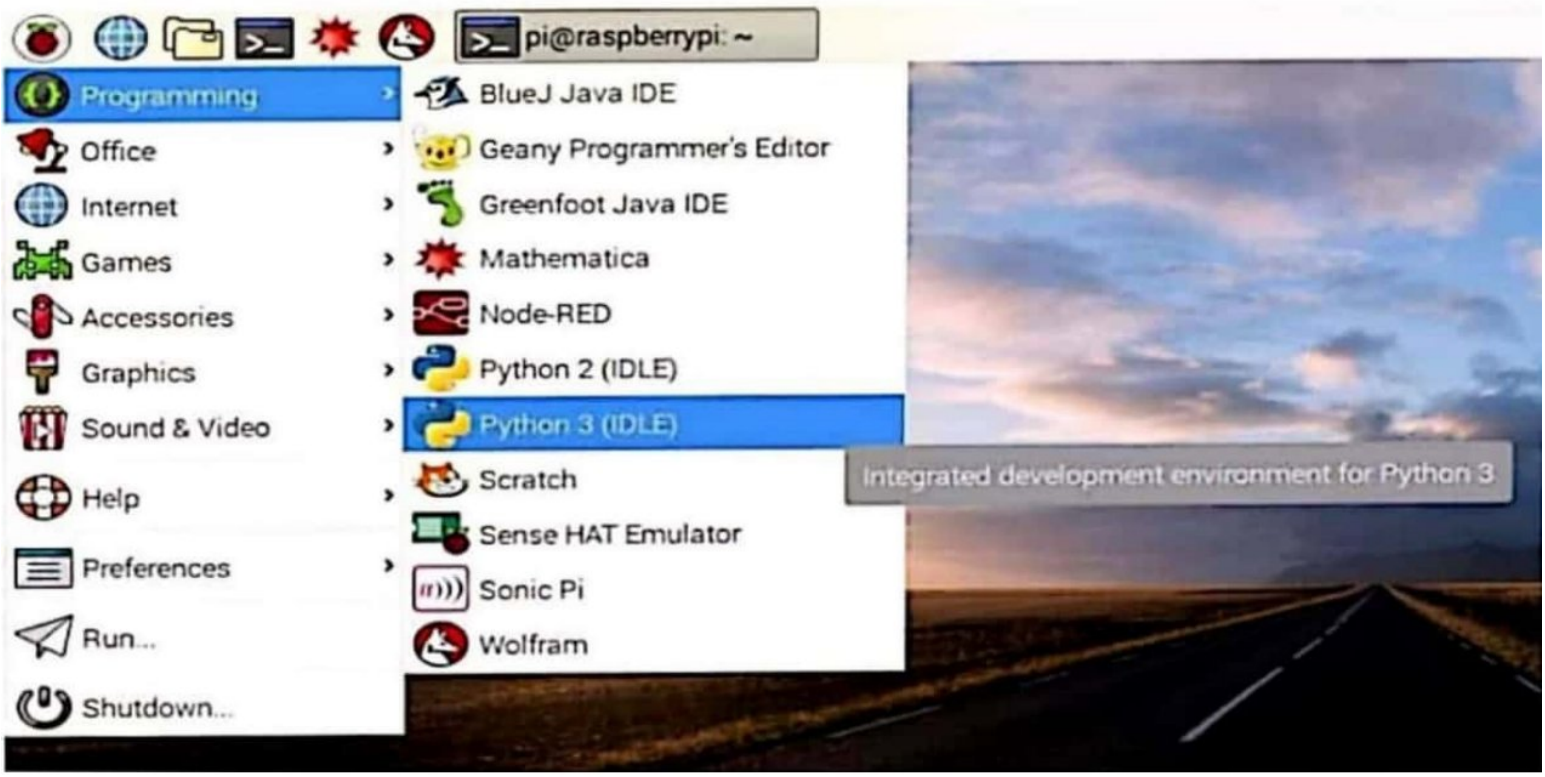
- The end with the piece of plastic with a hole in it, it will go onto the Raspberry Pi's GPIO pins.

- Conclusion:- Thus, we have studied about connectivity & config. of Raspberry Pi & also use GPIO.

Scanned with CamScanner



Scanned with CamScanner



Scanned with CamScanner

Name - Nirant Deore
TE (Computer)
Div - A
Roll No-31
Batch - TA2.

Saathi

Date __/__/__

ESIOT Assignment No-4

- **Aim:-** Understanding the connectivity of Raspberry Pi / Beagle board circuit with IR sensor. Write an application to detect obstacle & notify user using LED's.

- **Theory:-** Infrared sensor works by emitting infrared signal / radiation & receiving of the signal. When signal bounces book from any obstacle. In other words the IR sensor works by contionously sending signal & continously receiving signal comback by bouncing on any obstacle in the way.

- **Component IR Sensor:-**

i) **Emitter:-** This component continously with infrared signal.

ii) **Receiver:-** It waits for the signal which is bounced back by a obstacle.

iii) **Indicator:-** On board LED to signal if obstacle is deducted by sensor.

iv) **Output:-** Could be used as input for fulthe processing of signal.

v) **Ground:-** around / Negative point of circuit

vi) **Voltage:-** Input 3.3V.

Scanned with CamScanner

- Objective:- We will be creating a circuit using following components to detect obstacle.

i) Raspberry Pi 3
ii) IR sensor
iii) 1 LED
iv) 1 Resistor (330Ω)
v) few jumper cables
vi) 1 Breadboard

- Circuit:- To detect obstacles:-

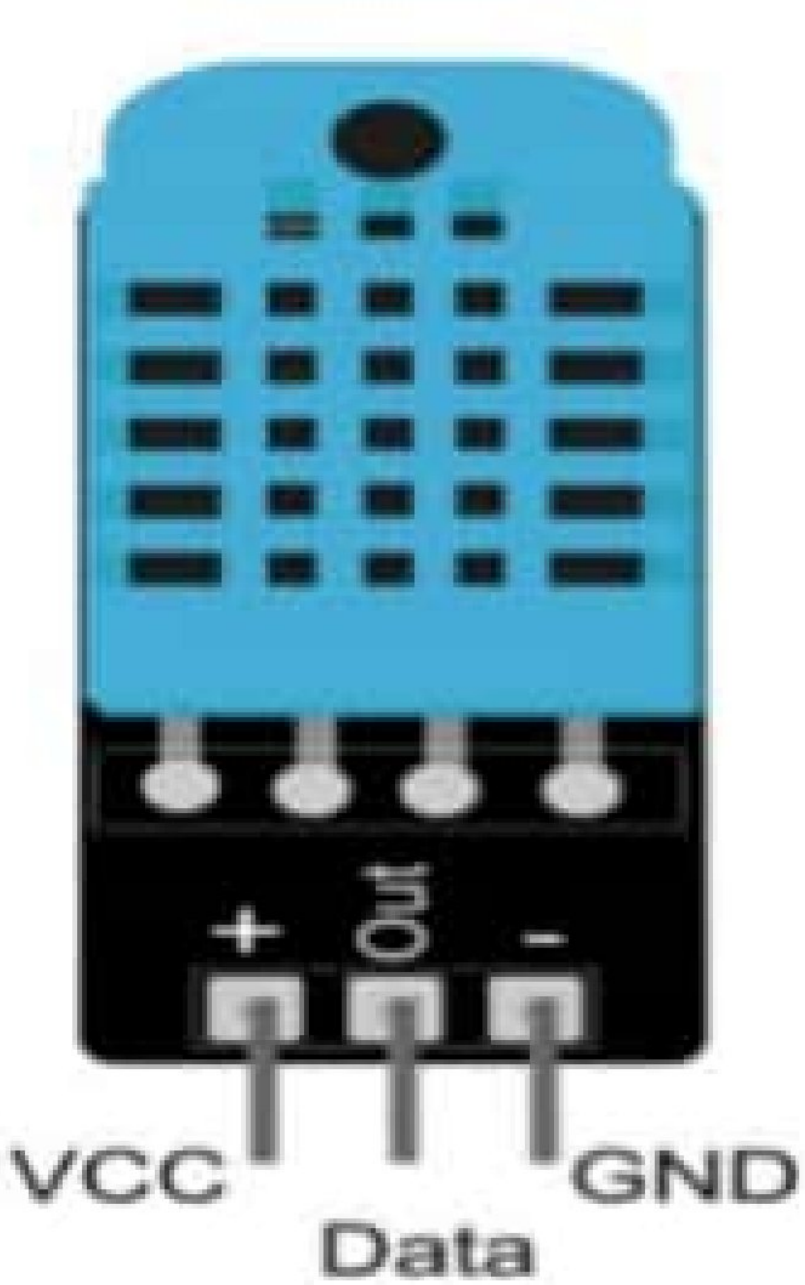
We will be creating a circuit which will turn on LED when obstacle occurs. and as soon as obstacle removed from the way the LED will turn off.

- Part 1:- Connecting IR sensors:-

IR sensor has 3pins viz Vcc, GND & OUT we will use GPIO 17 from receiving input from sensor.

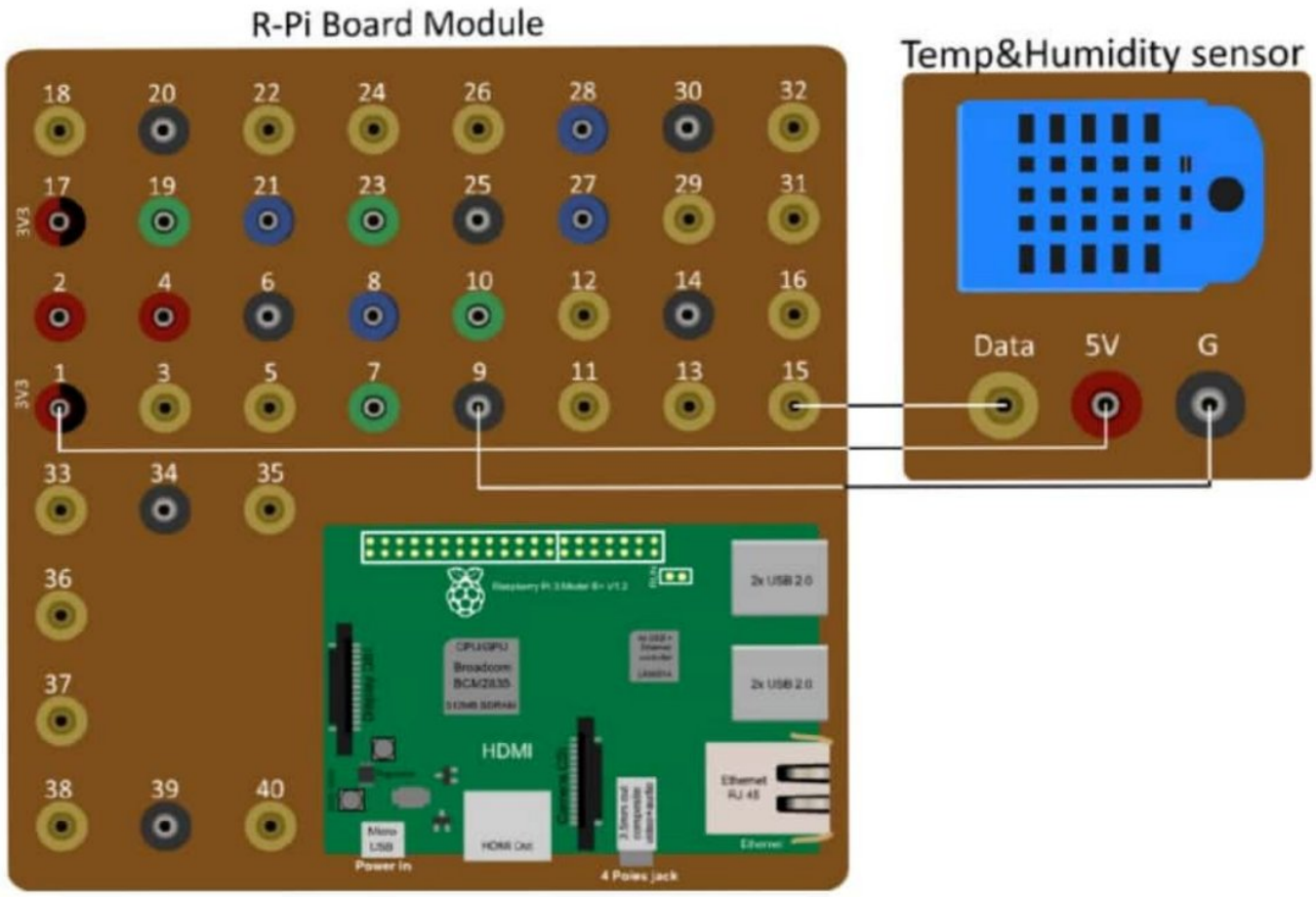
i) Connect GPIO 17 from Raspberry Pi to bread board.
ii) Connect out pin of sensor with bread board.
iii) Connect GND with Negative line on left side of bread board.
iv) Connect GND of IR sensor to bread board.
v) Connect GND from step 3 to breadboard.
vi) Connect VCC of IR sensor to bread board.
vii) Connect 3V3 to +ve line of leftside of breadboard.
viii) Connect 3r3 to breadboard.

Scanned with CamScanner



Scanned with CamScanner

Interface diagram:



Scanned with CamScanner

- **Part 2:-** Connecting LED:-

Objective is to turn on LED when obstacle is detected.

i) Connect GPIO 4 from board to breadboard.
ii) Connect positive point of LED to breadboard
iii) Connect negative point to the breadboard.
iv) Connect resistor (330Ω) to connect negative to point of LED.

Now we are ready to send signal based on input received from IR sensor to turn on left LED.

- **Part 3:-** Code to connect IR sensor IIP with LED status.

```
from GPO zero import LED
from signal import pause
import RPi GPO as GPO
import time
GPO.set mode (GPIO.BCM)
LED_PIN = 27
indicator = (LED-PIN)
GPIO. setup ( IR_PIN , GPIO.IN)
count = 1
while True:
    got something = GPIO.input (IR_PIN)
    if got_something
        indicator.ones
    print "{: >3 } got.something". format (count) }
else
```


Scanned with CamScanner

```
Indicator.Off()
Printf("{:>3} Nothing detected".format(count))
count+=1
time.sleep=(0.2)
```

- Part 4 :- Executing the table :-

i) Open terminal (On Pi itself or login through SSH login)

ii) Navigate to the directory where the above code is saved.

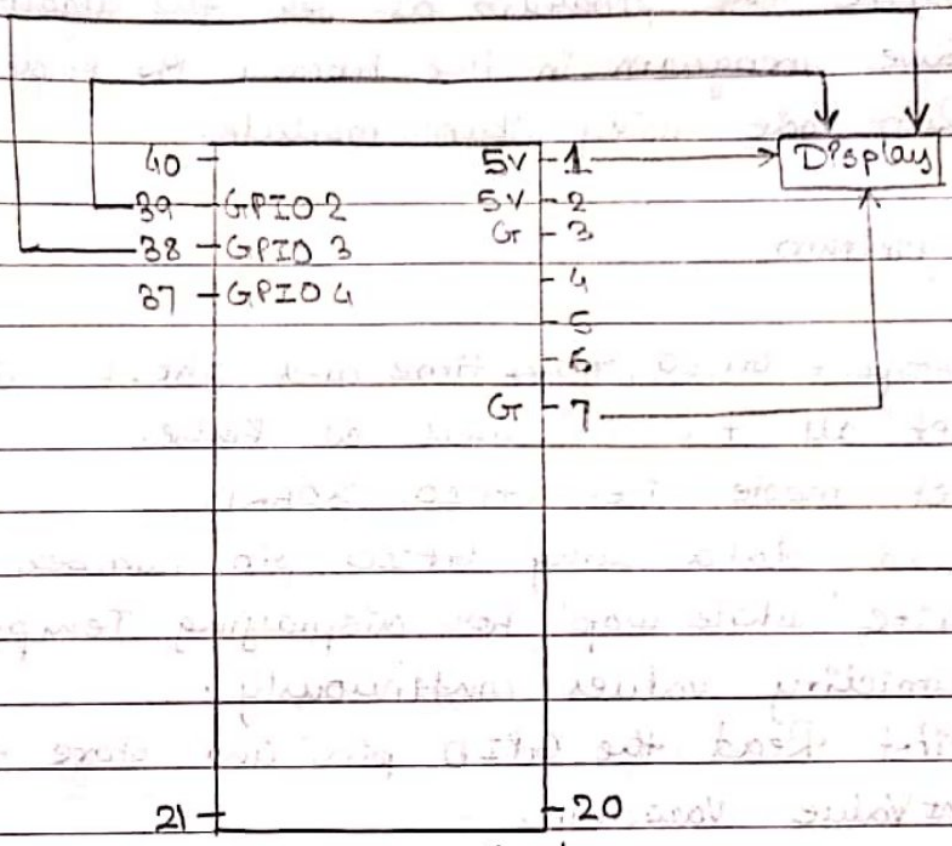
iii) Type $ python 3 ir_obstance.py & press <enter>

On terminal it will start printing the status based on conditions.

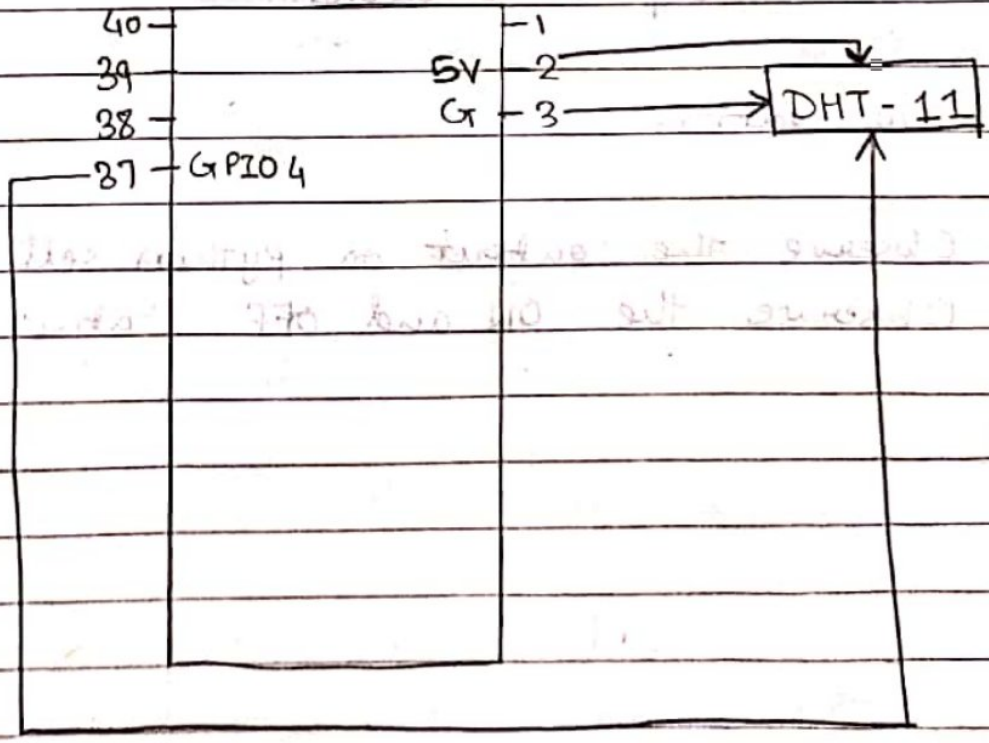
- Conclusion :- Thus, we done connectivity of Raspberry Pi / Beagle board circuit with IR sensor, write an application to detect obstacle & notify user using LED's

Scanned with CamScanner

* Temp Sensor (DHT-11)



GPIO Display



DHT - 11

Scanned with CamScanner

```
import lcddriver import
time import Adafruit_DHT
display = lcddriver.lcd()

try:     print("Press CTRL + C for stop this
script!")

    def long_string(display, text = '', num_line = 1, num_cols = 16):
if(len(text) > num_cols):
            display.lcd_display_string(text[:num_cols],num_line)
time.sleep(1)              for i in range(len(text) - num_cols +
1):                 text_to_print = text[i:i+num_cols]
                display.lcd_display_string(text_to_print,num_line)
time.sleep(0.5)              time.sleep(1)         else:
display.lcd_display_string(text,num_line)



    long_string(display, "DHT LCD R Pi!",1)
time.sleep(1)


    long_string(display, "Rohit World ",2)
time.sleep(1)     display.lcd_clear()

    while True:
            humidity, temperature = Adafruit_DHT.read_retry(11,4)
if(temperature != None and humidity != None):
display.lcd_clear()
                display.lcd_display_string('Temp:{0:0.1f} C
'.format(temperature),1)
                display.lcd_display_string('Humidity:{0:0.1f} %
'.format(humidity),2)
time.sleep(1)
                  except
KeyboardInterrupt:
print("Cleaning up!")
display.lcd_clear()
    display.lcd_display_string ('AsknCapture',1)
display.lcd_display_string('8237877250',2)
time.sleep(10)     display.lcd_clear()
```


Scanned with CamScanner

Name- Nirant Deore
TE (Computer)
Div-A
Roll No-31

Date __/__/__

ESIOT
Assignment No-5

- **Aim:-** Understanding & connecting of Raspberry Pi /Beagle board with camera. Write an application to capture & store the image.

- **Theory:-** Raspberry Pi camera module V2 replaced the original camera module in April 2016. The V2 camera module has a Sony IMX 219.8 megapixel sensor. The camera module can be used to take H.D. video as well as stills photograph its easy to use for beginners, but has plenty to offer advanced users if you are looking to expand your knowledge

- **Pi camera:-** i) Camera Preview:-

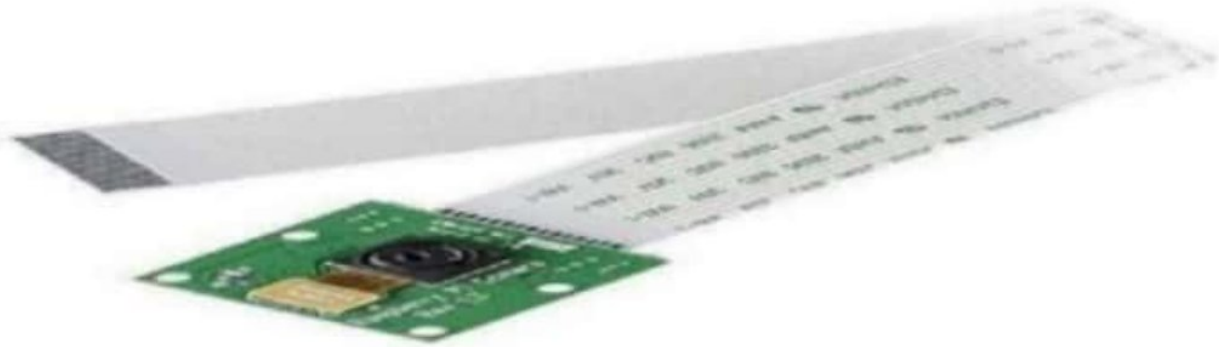
```
from Picamera import PiCamera
from time import sleep
camera = Pi Camera()
Camera. start_Preview()
sleep (10)
camera. stop_Preview()
```

ii) Rotating the Camera:-

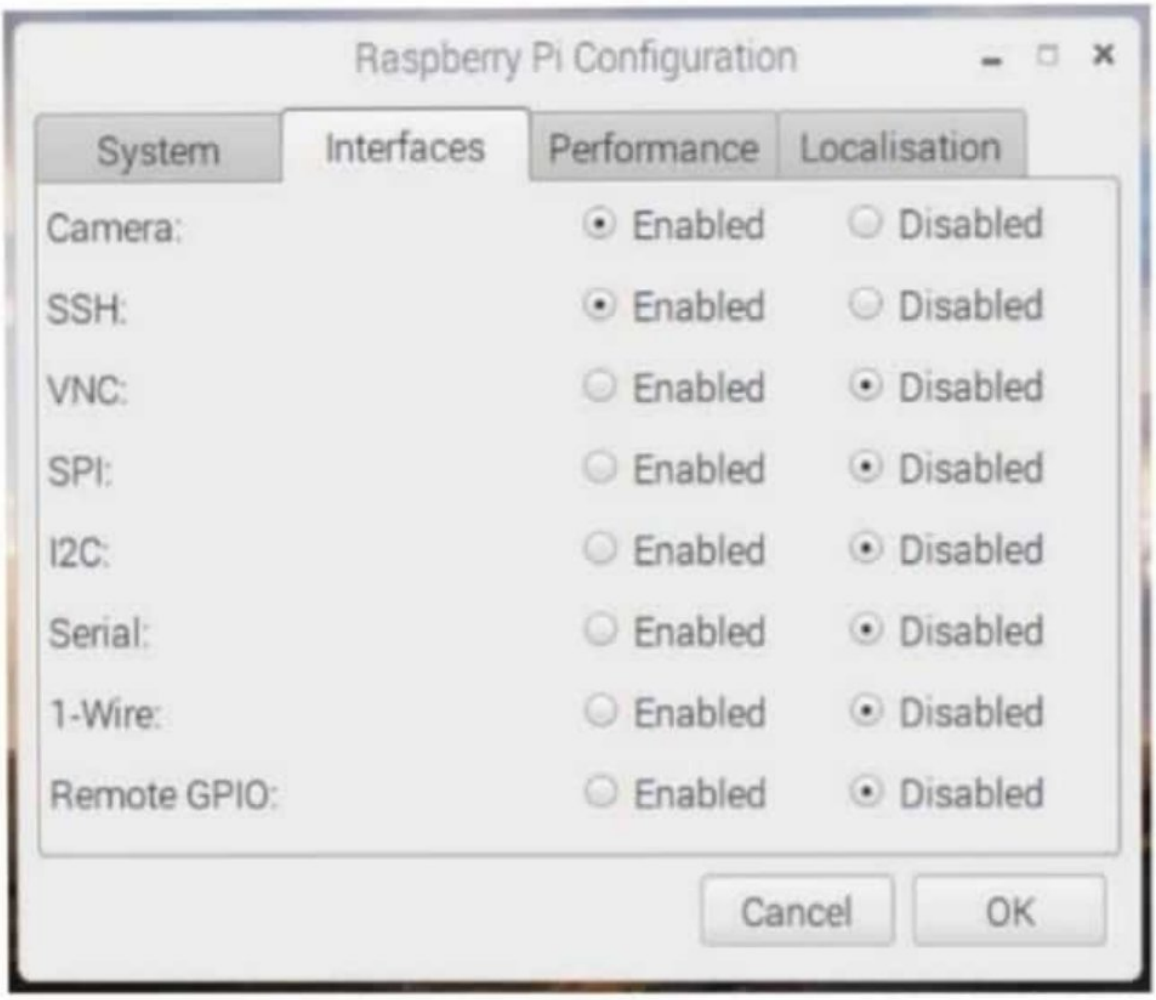
```
Camera.rotation = 180
Camera: start .preview()
sleep (10).
camera. stop-Preview()
```


Scanned with CamScanner

Pi Camera



Open Raspberry Pi Configuration and Enable the Camera



Scanned with CamScanner

iii) Storing the image:-

```
from Picamera import Picamera
from time import sleep
camera = Pi Camera()
camera.start_Preview()
sleep(10)
camera.capture ~~frevie~~ (/home/pi/Desktop/image1.jpg
Camera.stop_preview().
```

iv) Recording the video:-

```
from Pi camera input Pi camera
from time import sleep
camera = Pi camera()
Camera.start.preview()
Camera.start.recording (/home/pi/video.h26
sleep(10)
camera.stop.recording()
camera.stop_preview()
```

v) Converting & Playing Video:- The video format need to get converted to MP4.
So install gpae.
sudo apt-getinstall gpac
Now convert the video to MP4.
MP4 Box -fps 30 -add video.h264 video.mp4

- Conclusion:- Thus we have studied Pi camera & also stored the images & videos using Pi-camera.

Scanned with CamScanner

Name-Nirant Deore
TE (Computer)
Div-A
Roll No-31

Date __/__/__ ESIOT Assignment No-6

Saathi

- **Aim:-** Understanding & Connecting Raspberry Pi with a zigbee module. Write a network applicatn for communication betn two devices using Zigbee

- **Theory:-** Zigbee is a communicatn device used for data transfer betn controllers, computers systems real anything with serial port. As it works with low power consumptn the transmis-sion distance is limited to 10-100mts line of sight zigbee devices can be transmit data over long distances by passing data to a mesh networks of intermediate devices to reach more distant ones. The technology defined by the Zigbee specification is an less expensive than networks.

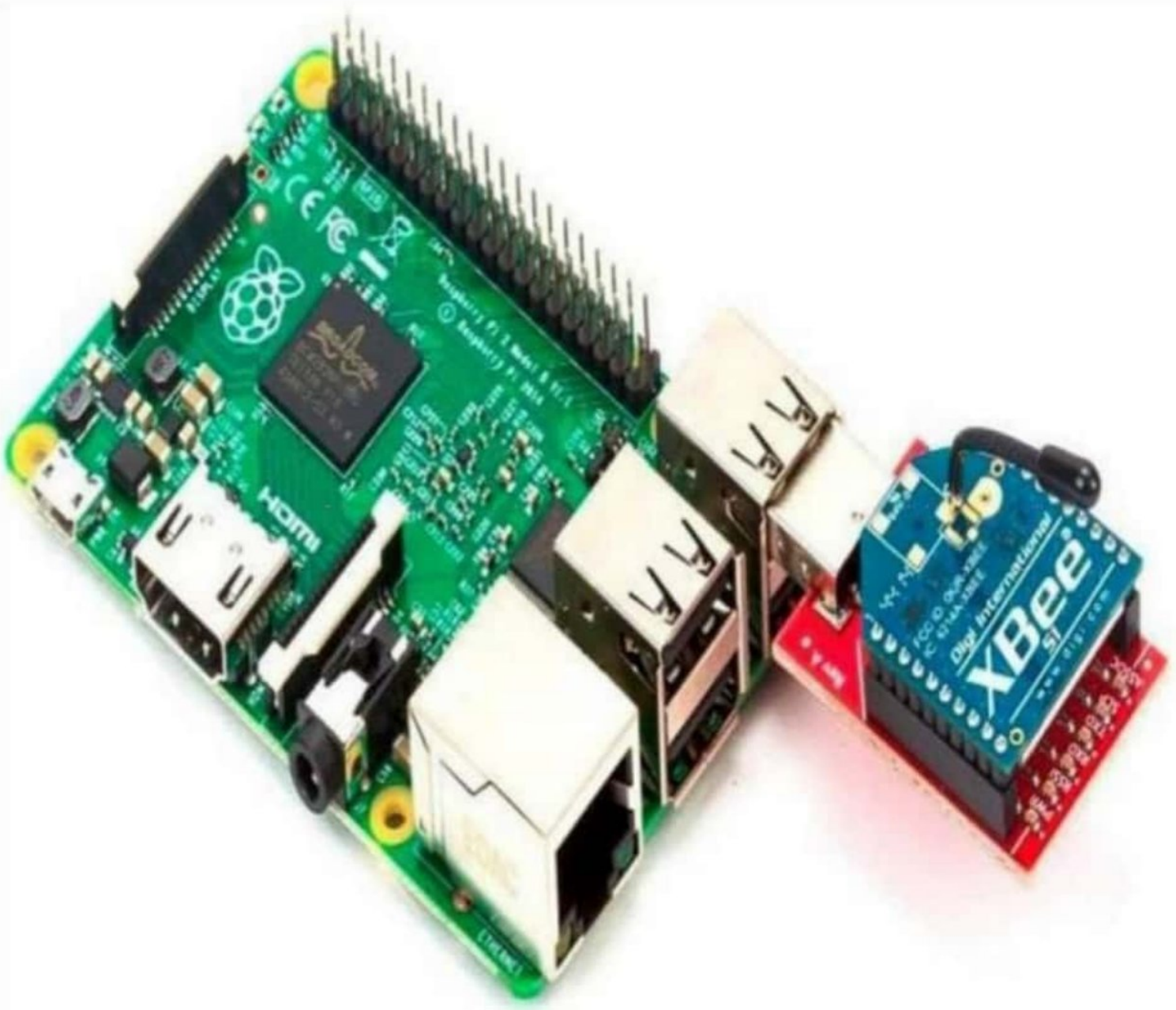
Python script to perform ZigBee communicat-ion.

```
import USB communication.
import serial
#Enable USB communication.
Ser = Serial.serial('/dev/ttyUSB0', 9600,)
while: True:
    ser.write("Hello User /r/n') #write a data
    incoming = ser.readline().strip()
    print 'Received data:' +incoming.
```

- **Conclusion:-** Thus, we have done ZigBee communication betn two Raspberry Pi devices.

Page No.

Scanned with CamScanner



Interfacing of Zigbee

Scanned with CamScanner

Name-Nirant Deore
TE computer
Div-A
Roll No-31

ESIOT

Date __/__/__ Assignment No-7

Saathi (5)

- Aim:- Write an application using Raspberry PI/ Beagle board to control the operation of stepper motor.

- Theory:-

- Steppes Motor:- In steper motor, as the name itself says the rotation of shalf is in step form. There are different types of stepper motor and here we will be using most popular one that is unipolar stepper motor. Unlike DC motor, we can rotate stepper motor to any particular angle by giving it proper instructions.

To rotate this 4 stage stepper motor we will deliver power pulses by using stepper motor driving circuit. The driver circuit takes logic triggers from PI. If we control the logic triggers we control the power pulses & hence control the speed of stepper motor.

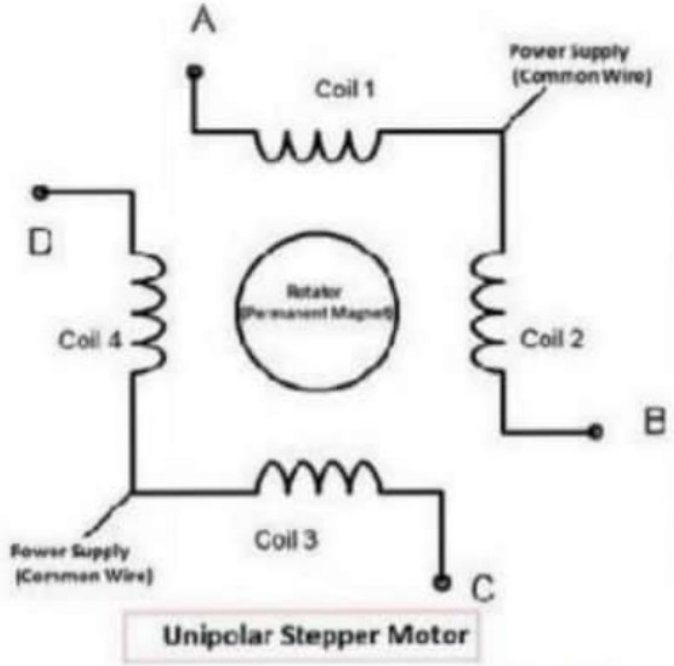
There are 40 GPIO output pins in Raspberry Pi. But out of 40 only 26 GPIO pins can be programmed. Some of this pins perform special functions with special GPIO put aside, we have only 17 GPIO remaining. And the sum of currents from all GPIO pins cannot exceed 50mA.

- Sample Program:- Stepper motor interfacing with Raspberry Pi.

```
import RPi GPIO as GPIO.
from time import sleep
```

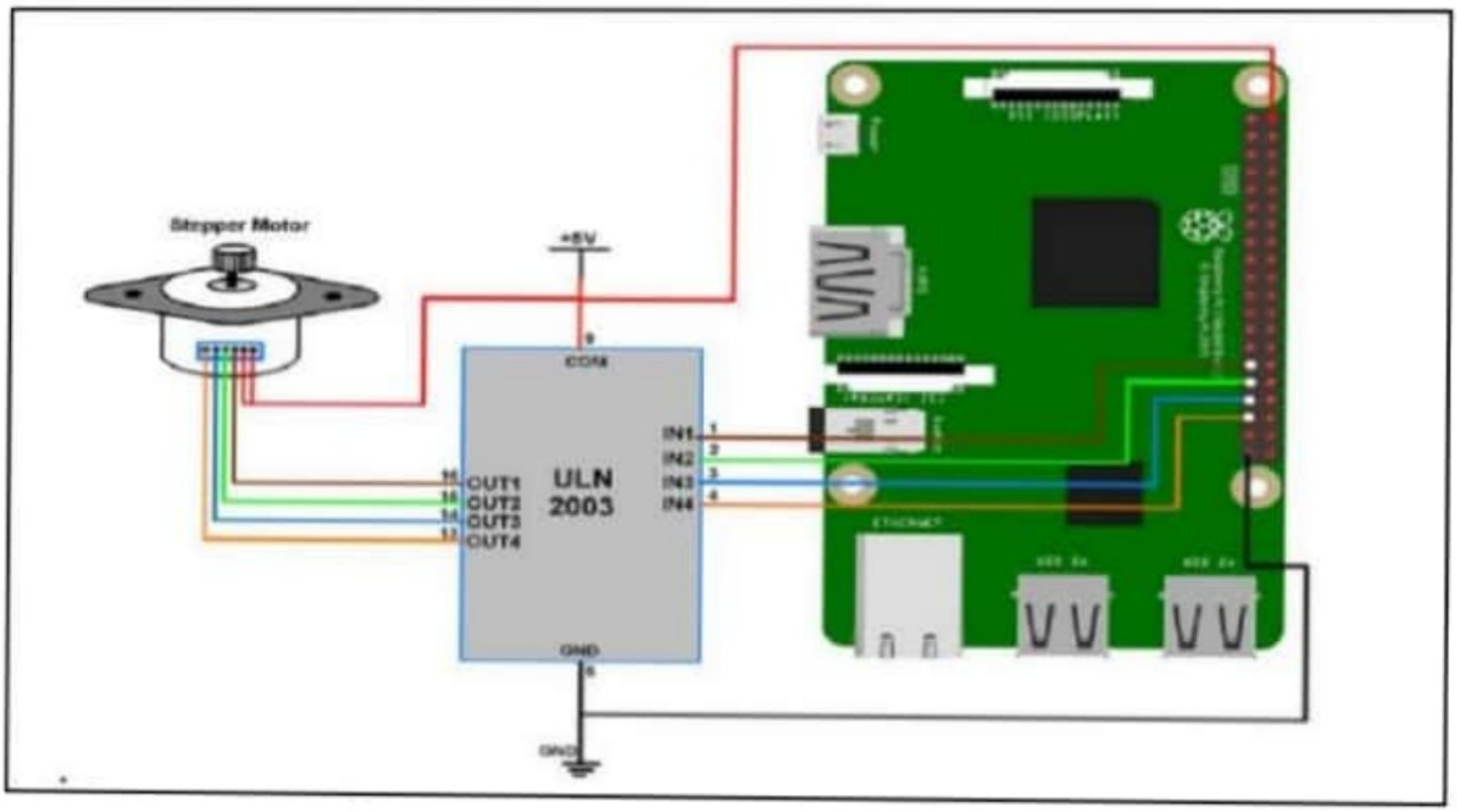
Page No.

Scanned with CamScanner



Unipolar Stepper motor Fig.1

Scanned with CamScanner



Scanned with CamScanner

```
import sys.
#asign GPIO pins from motor.
motor_channel =(29,31,33,35)
GPIO.Setwarnings(FALSE)
GPIO.setmode(GPIO.Board)
#for defining more than 1GPIO channelas
i/p o/p use GPIO.setup(motor_channel,GPIO.ou
motor_direction = input('select motor directiona=
                                anticlockwise)
while True:
try:
    if (motordirection =='(' ):
    print('motor running clockwise)
GPIO.output(motor_channel,[GPIO.High,GPIO.low
sleep(0.02)                GPIO.Low,GPIO.High
    else if (motor direction == 'd'):
    print('motor running anticlockwise)
    GPIO.output(motor_channel(GPIO.High,GPIO.low,
    sleep(0.02)                GPIO.low,GPIO.High)

#press ctrl+c for keyboard intrupt
except keyboard interrupt:
#query for setting motor direction or exit
motor_direction = input('select motor direction a=
                        anticlockwise, c= clockwise)
#check for exit.
    if (motor_direction == 'q')
    print('motor stopped')
    systexit(0).
```

- **Conclusion:-** Thus, we have implemented applicat^n of stepper motor wing python with Raypberry Pi

Scanned with CamScanner

Name-Nirant Deore
TE Computer
Div-A
Roll No-31

Date __/__/____ ESIOT Assignment No-8

Saathi

- **Aim:-** Write an application using Raspberry Pi/ Beagle board to control the operation of hardware stimulated traffic signal.

- **Theory:-** • Attaching the traffic lights:-

The low voltage labs traffic light connected to the Pi using four pins. One of these needs to be ground, the other being actual GPIO pins used to control each of individual LEDs.

Before powering up the Pi, attach the traffic lights so that the pins connect to the GPIO pins highlighted in red.

- Programming the traffic lights:-

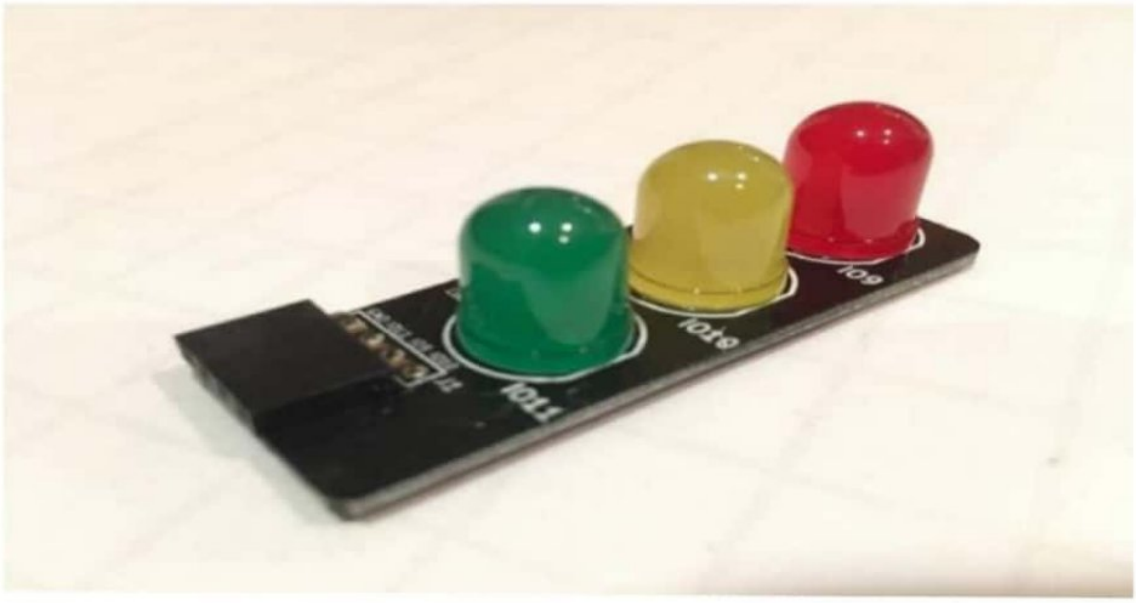
first you need to install a couple of extra software packages needed to allow you to download my sample code & to give python access to GPIO pins on Pi. Each of following at command line

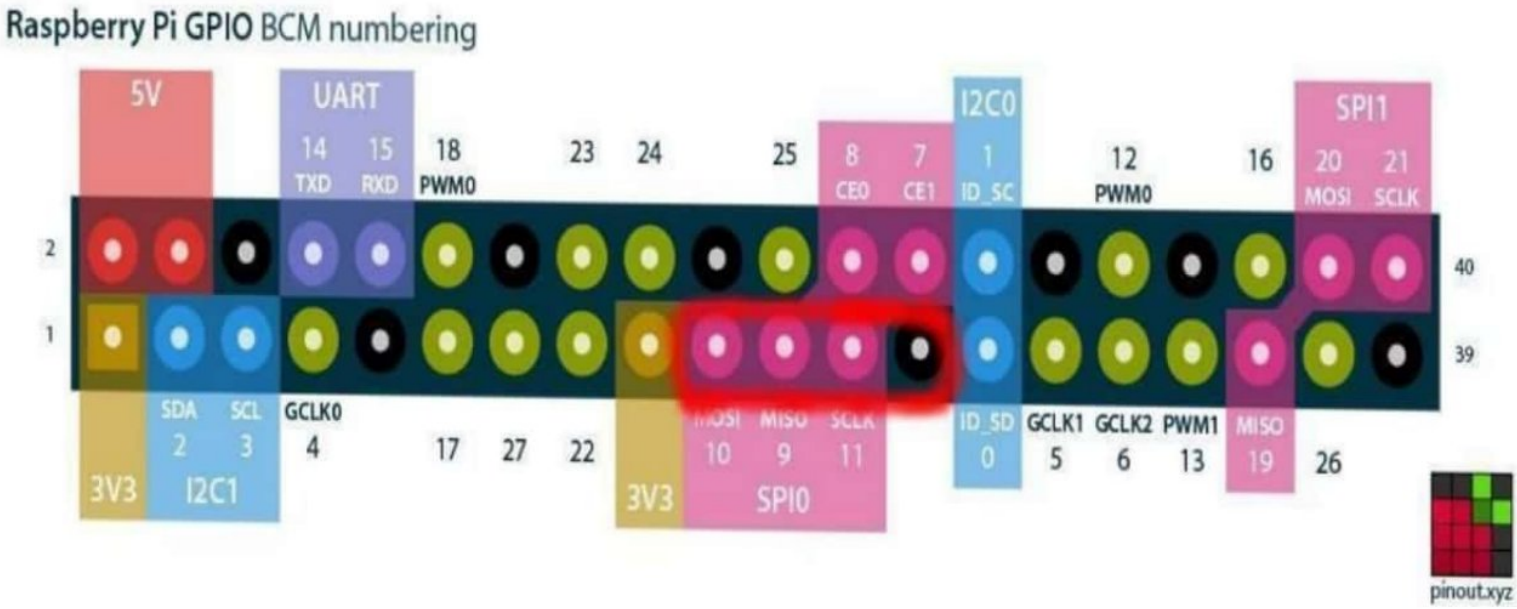
```
sudo apt-get install python-dev python-rpi
                     gpio.git
```

- How it works:- The code for this is very simple. It starts by importing the RPI GPIO library plustime which gives us a timed wait fun^n, signal that allows us to trap the signal sent when the user tries to quit the program & sys so we can send an appropriate exit signal back to

Page No.

Scanned with CamScanner





Scanned with CamScanner

```
import RPI , GPIO as GPIO
import time
import signal
import sys
```

Next we put the GPIO library into "BCM" or "Broadcam' & sets pins 9, 10 & 11 to be used as outputs.

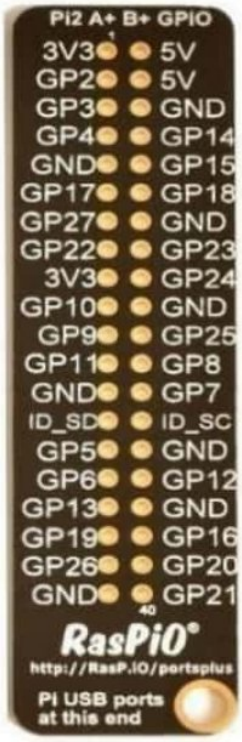
```
# setup
GPIO. ~~sk~~setmode(GPIO.BCM)
GPIO. setup ( 9, GPIO .OUT)
GPIO .setup ( 10, GPIO.OUT)
GPIO.setup ( 11, GPIO,OUT)
```

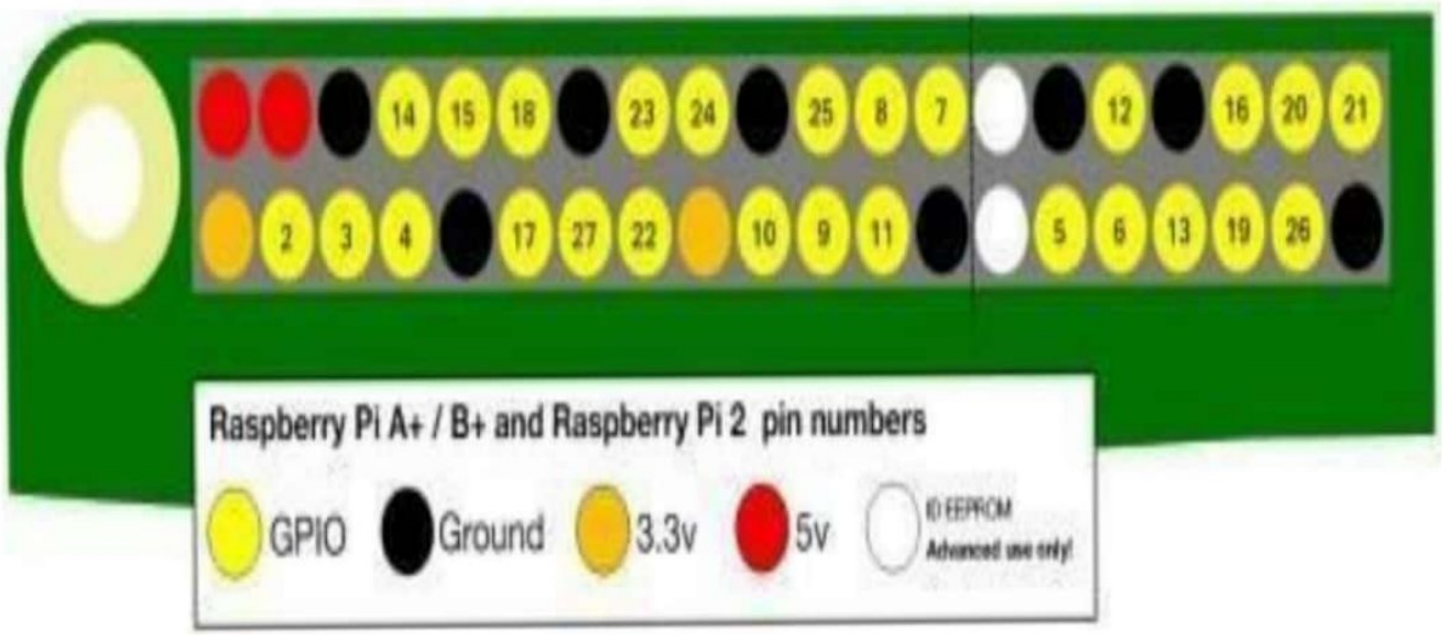
The main part of program will run in an infinite loop until the user exists by stopping python with ctrl+c

```
# Turn off all lights when user ends demo
def all lights off (signal, frame):
    GPIO . output (9, false)
    GPIO. output (10, false)
    GPIO. output (11, False)
    GPIO . cleanup ( )
    sys.exit (0)
signal.signal (signal. SIGINT .all lights off)
```

The main body of code then consists of an infinite loop while loop that turns on red light (Pin9) wait, turns on another light (Pin10) waits, then cycles through the rest of traffic light pattern by turning the appropriate LED's on & off.

Scanned with CamScanner



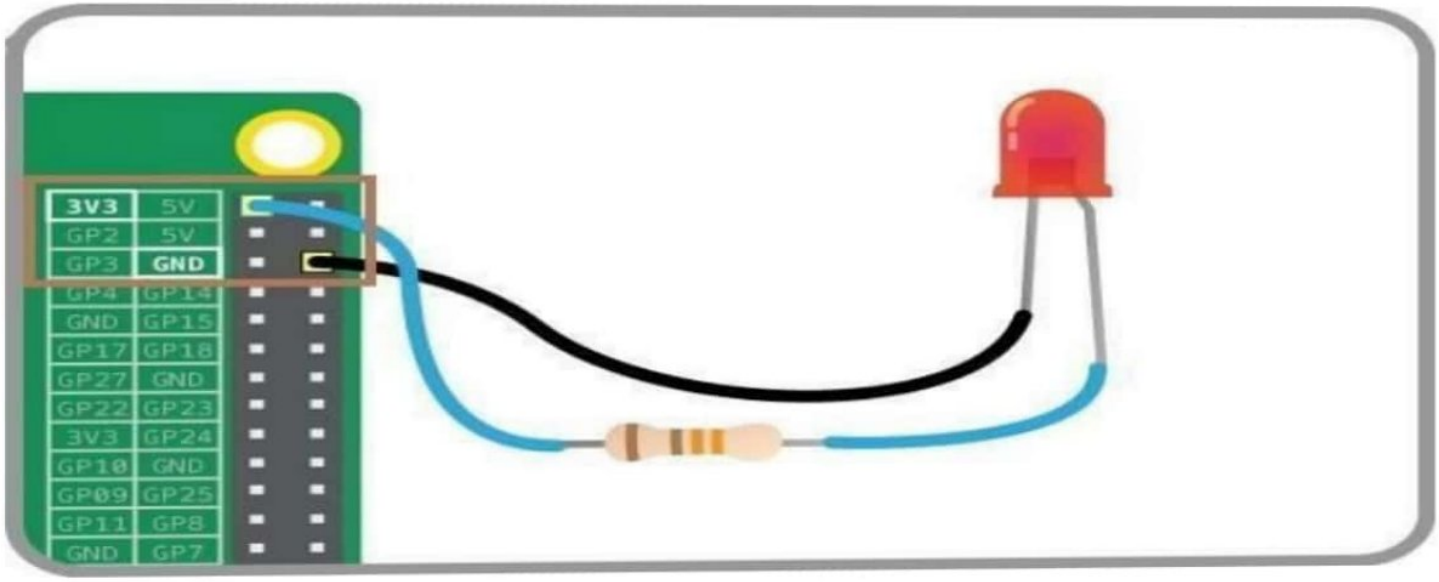


Scanned with CamScanner

When control-C is passed an interrupt SIGINT is sent. This is handled by the all lights off function that switches all lights off tidies up the GPIO library stock f exists cleanly back to O.S.

- Conclusion:- Thaus, we have implemented the application for triaffic signal lights using Raspberry Pi.

Scanned with CamScanner



Scanned with CamScanner

Scanned with CamScanner

Name-Nirant Deore
TE Computer
Div-A
Roll No-31.

Date __/__/__ ESIOT Assignment No-9

- **Aim:-** Write an application using Raspberry Pi to control the operation of hardware stimulated lift elevator.

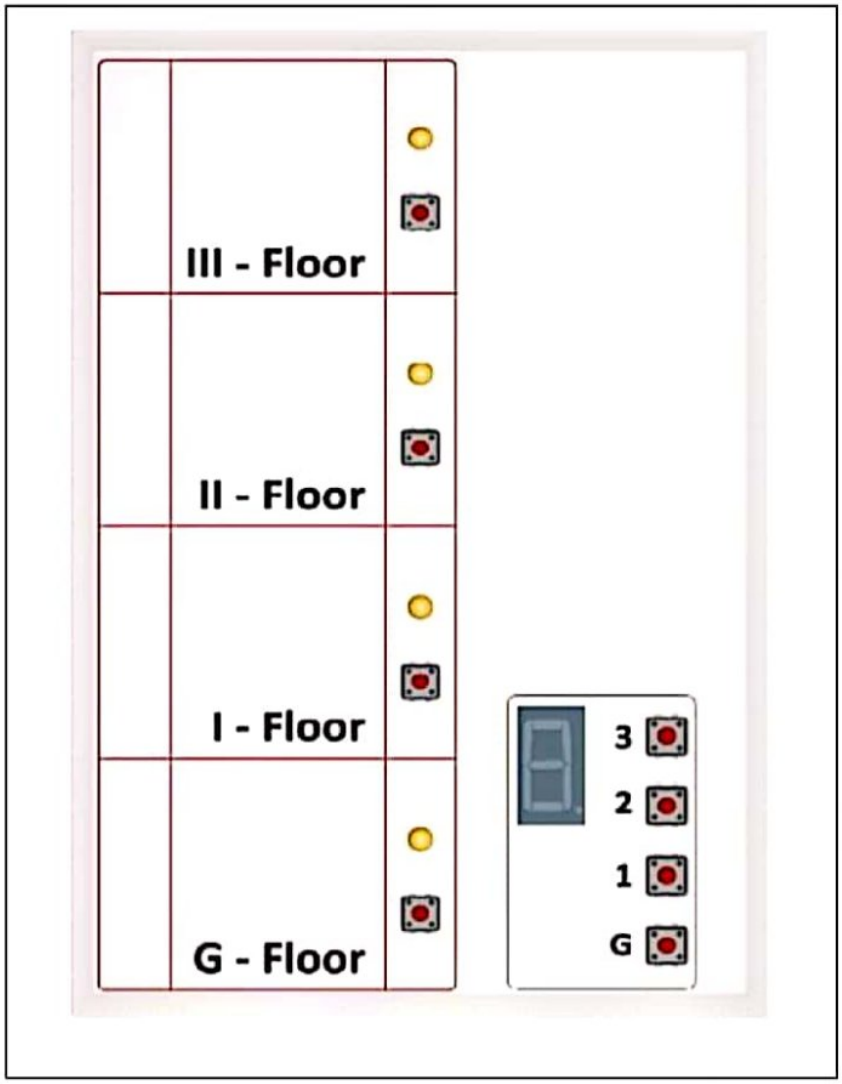
- **Objectives:-** i) To understand the working principl of lift elevator.
ii) To interfaci the lift elevator module with Raspbery Pi model.
iii) To program the Raspberry Pi model to control operation of lift elevator module.

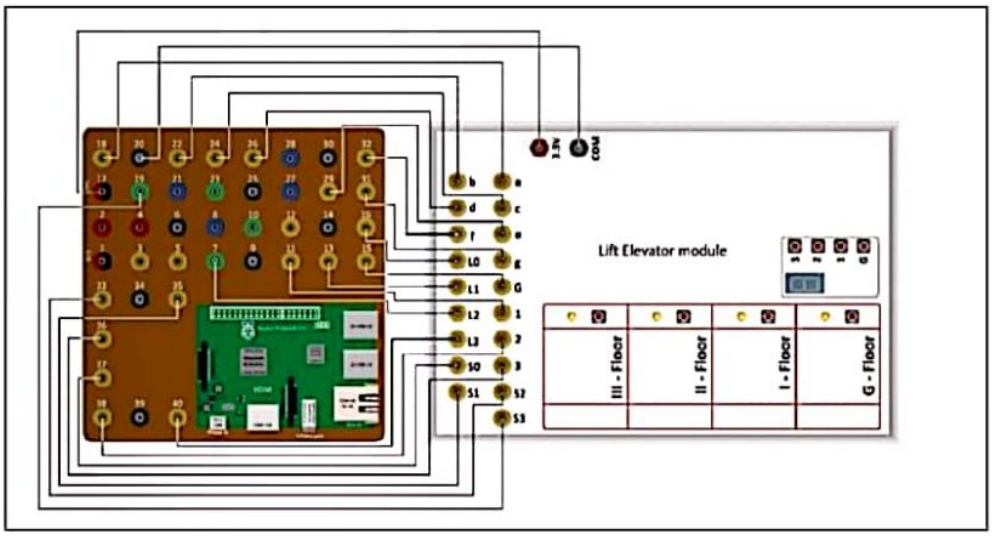
- **Software:-** Raspbian OS (IDLE)

- **Hardware:-** i) Raspberry Pi Board module.
ii) Push buttons (qty-8)
iii) Raspbian OS (IDLE)
iv) LEDs (qty-4)
v) Monitor.

- **Theory:-** • Lift Elevator module has two parts :-
i) Moving part inside the lift &
ii) Stationery part outside the lift at each floor to call lift.
iii) In this stimulation module, we have considered four floors of building.
iv) The moving part also contains a seven segment display to indicate the current floor number when lift is moving.
v) By pressing one of these buttons, the user indicates destination floor.

Scanned with CamScanner





Scanned with CamScanner

vi) At each floor, the stationary part contains a button for calling the lift.
vii) In real life as soon as the entering user get finished the lift is closed & the lift starts moving towards destination.
viii) In our module, this situation is indicated by "LED OFF" status. So the LED OFF status indicates that the lift is moving towards the destination floor.

- Safety Precautions:- i) First, make all the connections as per steps given below.
ii) Power supply.

- Steps for assembling circuit:-

i) Connect all pins of lift elevator module to pins of Raspberry Pi module as showing fig.
ii) Write program as per our algorithm.
iii) Save Program.
iv) Run code using Run module.

- Algorithm:-

i) Import GPIO & time libraries.
ii) Set GPIO mode as perboard
iii) Declare 4 push button pins of stationary part.
iv) Declare 4 LED pins at each floor for detection of door close & open.

Scanned with CamScanner

v) Set the push button pins as input.
vi) Set the seven segment display pins display in variables.
vii) In loop if Button one is pressed then lift at floor 1 & LED at floor 1 get ON for 5 second gets off.
viii) The seven segment displays the floor number of destination.

Scanned with CamScanner

```
#Interfacing Lift Elevator module with Raspberry-Pi-3
import RPi.GPIO as GPIO
import time

FloorButton0 = 37
FloorButton1 = 35
FloorButton2 = 33
FloorButton3 = 19

LiftButton0 = 15
LiftButton1 = 11
LiftButton2 = 38
LiftButton3 = 36

#GPIO setup for the LEDs
FloorLed0 = 16
FloorLed1 = 13
FloorLed2 = 7
FloorLed3= 40

#GPIO setup for the Seven Segment Display
segAPin=18
segBPin=22
segCPin=24
segDPin=26
segEPin=29
segFPin=32
segGPin=31

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

GPIO.setup(FloorButton0, GPIO.IN)
GPIO.setup(FloorButton1, GPIO.IN)
GPIO.setup(FloorButton2, GPIO.IN)
GPIO.setup(FloorButton3, GPIO.IN)

GPIO.setup(LiftButton0, GPIO.IN)
GPIO.setup(LiftButton1, GPIO.IN)
GPIO.setup(LiftButton2, GPIO.IN)
GPIO.setup(LiftButton3, GPIO.IN)

GPIO.setup(FloorLed0, GPIO.OUT) #Floor 1
```


Scanned with CamScanner

```
GPIO.setup(FloorLed1, GPIO.OUT) #Floor 2

GPIO.setup(FloorLed2, GPIO.OUT) #Floor 3

GPIO.setup(FloorLed3, GPIO.OUT) #Floor 4

GPIO.setup(segAPin, GPIO.OUT)


GPIO.setup(segBPin, GPIO.OUT)
GPIO.setup(segCPin, GPIO.OUT)
GPIO.setup(segDPin, GPIO.OUT)
GPIO.setup(segEPin, GPIO.OUT)
GPIO.setup(segFPin, GPIO.OUT)
GPIO.setup(segGPin, GPIO.OUT)

digitclr=[0,0,0,0,0,0,0]
digit0=[1,1,1,1,1,1,0]
digit1=[0,1,1,0,0,0,0]
digit2=[1,1,0,1,1,0,1]
digit3=[1,1,1,1,0,0,1]

gpin=[18,22,24,26,29,32,31]
#routine to clear and then write to display
def digdisp(digit):
for x in range (0,7):
GPIO.output(gpin[x], digitclr[x])

for x in range (0,7):
GPIO.output(gpin[x], digit[x])

while True:

if (GPIO.input(FloorButton0)== True) :
GPIO.output(FloorLed0,1)
print"0"

digdisp(digit0)
time.sleep(1)
GPIO.output(FloorLed0,0)
time.sleep(3)

while True:

if(GPIO.input(LiftButton1)== True):
print'floor ONE'
```


Scanned with CamScanner

```
digdisp(digit0)
time.sleep(1)
digdisp(digit1)
time.sleep(2)
break

elif (GPIO.input(LiftButton2)== True):
print'floor TWO'
digdisp(digit0)
time.sleep(1)
digdisp(digit1)
time.sleep(1)
digdisp(digit2)
time.sleep(2)
break

elif (GPIO.input(LiftButton3)== True):
print'floor THREE'
digdisp(digit0)
time.sleep(1)
digdisp(digit1)
time.sleep(1)
digdisp(digit2)
time.sleep(1)
digdisp(digit3)
time.sleep(2)
break

elif (GPIO.input(FloorButton1) == True):

GPIO.output(FloorLed1, 1)
print"1"
digdisp(digit0)
time.sleep(1)
digdisp(digit1)
time.sleep(1)
time.sleep(4)

GPIO.output(FloorLed1, 0)

while True:

if(GPIO.input(LiftButton0)== True):
print 'floor ZERO'
digdisp(digit0)
time.sleep(2)
```


Scanned with CamScanner

```
break

elif (GPIO.input(LiftButton2)== True):
print'floor TWO'
digdisp(digit2)
time.sleep(2)
break

elif (GPIO.input(LiftButton3)== True):
print'floor THREE'
digdisp(digit2)
time.sleep(1)
digdisp(digit3)
time.sleep(2)
break

elif (GPIO.input(FloorButton2) == True):

GPIO.output(FloorLed2, 1)

print"2"

digdisp(digit0)
time.sleep(1)
digdisp(digit1)
time.sleep(1)
digdisp(digit2)
time.sleep(1)
time.sleep(5)
GPIO.output(FloorLed2, 0)

while True:

if(GPIO.input(LiftButton0)== True):
print 'floor ZERO'
digdisp(digit1)
time.sleep(1)
digdisp(digit0)
time.sleep(2)
break

elif (GPIO.input(LiftButton1)== True):
print 'floor ONE'
digdisp(digit1)
time.sleep(2)
break
```


Scanned with CamScanner

```
elif (GPIO.input(LiftButton3)== True):
print'floor THREE'
digdisp(digit3)
time.sleep(2)
break

elif (GPIO.input(FloorButton3) == True):

GPIO.output(FloorLed3, 1)

print"3"
digdisp(digit0)
time.sleep(1)
digdisp(digit1)
time.sleep(1)
digdisp(digit2)
time.sleep(1)
digdisp(digit3)
time.sleep(6)

GPIO.output(FloorLed3, 0)

while True:

if (GPIO.input(LiftButton0)== True):
print 'floor ZERO'
digdisp(digit2)
time.sleep(1)
digdisp(digit1)
time.sleep(1)
digdisp(digit0)
time.sleep(2)
break

elif (GPIO.input(LiftButton1)== True):
print 'Floor ONE'
digdisp(digit2)
time.sleep(1)
digdisp(digit1)
time.sleep(2)
break

elif (GPIO.input(LiftButton2)== True):
print'floor TWO'
digdisp(digit2)
time.sleep(2)
```


Scanned with CamScanner

```
break

else:
####    time.sleep(3)
digdisp(digit0)
GPIO.output(FloorLed0, 0)
GPIO.output(FloorLed1, 0)
GPIO.output(FloorLed2, 0)
GPIO.output(FloorLed3, 0)

else:
####    time.sleep(3)
digdisp(digit0)
GPIO.output(FloorLed1, 0)
GPIO.output(FloorLed2, 0)
GPIO.output(FloorLed3, 0)
GPIO.output(FloorLed0, 0)
```


Scanned with CamScanner

Name - Nircunt Deore
TE Computer
Div - A
Roll No - 31

Date __/__/__ ESIOT Assignment NO-10

Saathi

- **Aim:-** Create a small dashboard application to be deployed on cloud. Different publisher devices can publish their information & interested application can subscribe.

- **Theory:-** •**IoT Platforms:-** The IOT platforms are suites of components those help to setup & manage the internet connected device. A person can remotely collect data, monitor & manage all internet connected devices from single system.

• **IOT Cloud Platforms:-**

i) Kaa IoT platform.
ii) Sitewhere: open platform for IOT
iii) Thingspeak: An open IoT platform with MATLAB analytics.
iv) Device hire: IOT made easy.
v) Zetta: API - first IOT platform.

• **Kaa features:-**

i) Manage an unlimited numbers of connected devices
ii) Setup cross device interoportability.
iii) Perform AIB service testing.
iv) Perform real time device monitoring.
v) Collect & analyze sensor data.
vi) Analyze user behaviour deliver target notifications.
vii) Create cloud services for smart products

Page No.

Scanned with CamScanner

• Siteware features:-

i) Run any number of IoT applications on a single site where instance.
ii) spring delivers the core configuration network.
iii) Connect devices with MCETT, AMBP, STOMP & other protocols.
iv) Add devices through self registration, REST services, or in batches.
v) Default database storage is mongoDB.
vi) Eclipse californium for coAP messaging
vii) Influx DB for event data storage
viii) HBase for non-relational datastore

• Device-hive features:-

i) Directly integrate with Alexa.
ii) Visualization dashboard of your device.
iii) Customize Devicehive behaviour or by running your custom javascript code.
iv) Connect any device via REST API, websockets or MQTT.
v) It comes with Apache spark & spark streaming support.

• Thingspeak-features:-

i) Collect data in private channels.
ii) Share data with public channels.
iii) Restful & MQTT APIs
iv) MATLAB analytics & visualization.
v) Alerts.

Scanned with CamScanner

Analytics

MATLAB Analysis

Explore and transform data.

MATLAB Visualizations

Visualize data in MATLAB plots.

Plugins

Display data in gauges, charts, or custom plugins.

Actions

ThingTweet

Connect a device to Twitter® and send alerts.

TweetControl

Listen to the Twitterverse and react in real time.

TimeControl

Automatically perform actions at predetermined times with ThingSpeak apps.

React

React when channel data meets certain conditions.

TalkBack

Queue up commands for your device.

ThingHTTP

Simplify device communication with web services and APIs.

Scanned with CamScanner

vi) Event scheduling.
vii) App integrations.
viii) World wide community.

- Zetta features:-

i) Built around NodeJS, REST, websockets & flow based "reactive programming".
ii) Supports wide range of hacker boards.
iii) Zetta allows you to assemble smartphone apps device apps & cloud apps.

- Conclusion:- Thus, we have designed small application using Thingspeak.

Scanned with CamScanner

Name- Nirant Deore
TE Computer
Div-A
Roll No-31

Date __/__/__ ESIOT Assignment No-11

- **Aim:-** Create a simple web interface for Raspberry Pi / Beagle board to control the connected LED's remotly through interfac

- **Theory:-** • **Coding Pi:-** Wiring Pi is a pin base GPIO access library written in C for the BCM2835 used in the Raspberry Pi. It is released under GHU LGPLV3 license and is usuable from C, C++ & RTB as well as many other languages with suitable wrappe

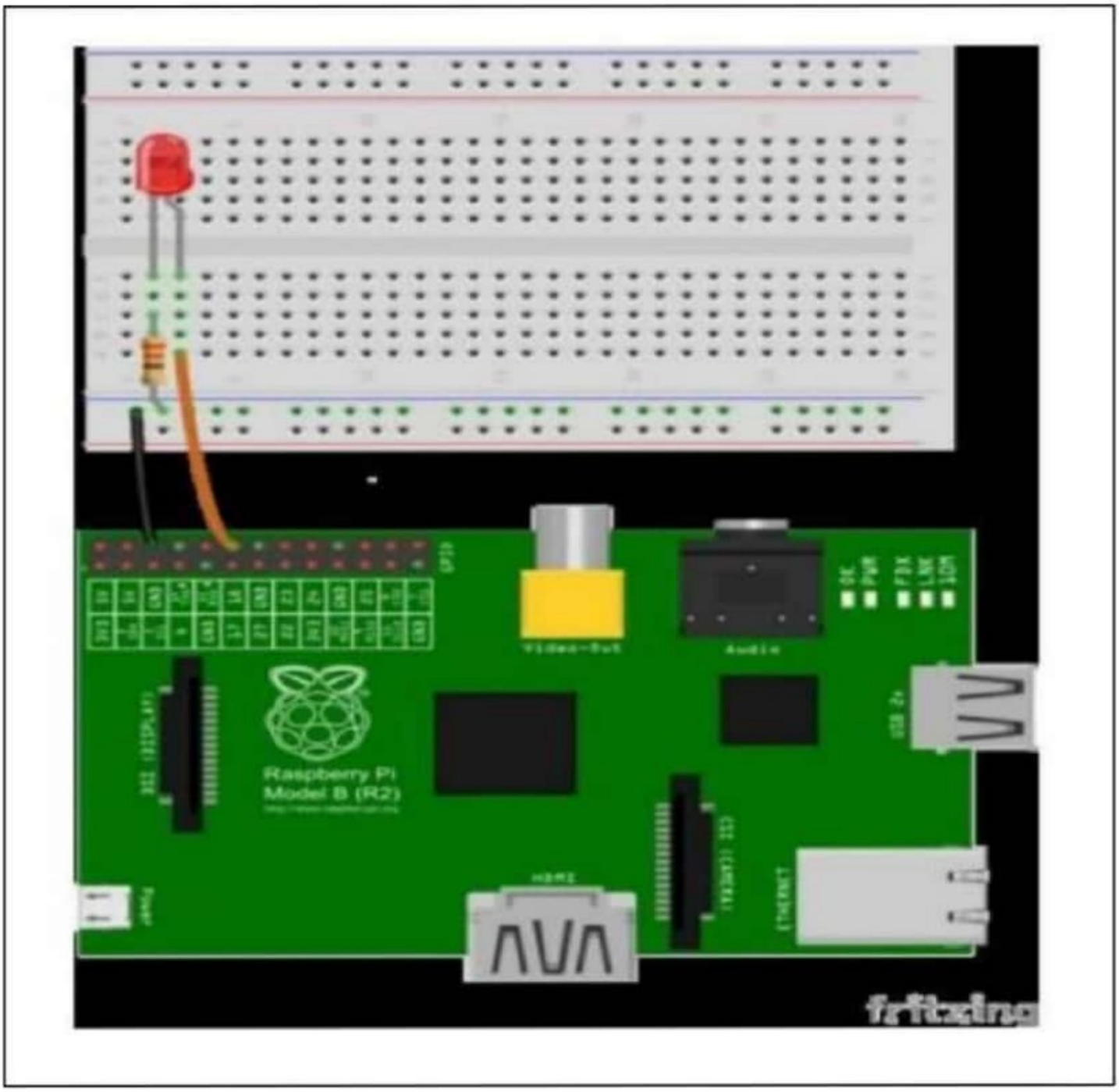
• **Install Wiring Pi:-** Wiring Pi is not included with Raspbian. So to begin, you'll need to download & install it.

• **GPIO command line utility:-**
Task: connect the LED GND to short pin GPIO18 to long pin.
GPIO command line utility.
i) Glow the LED by value.
gpio write 11.
ii) Off the LED by
gpio write 10.

• **Webinterface to LED:-**

i) Create the front page using HTML with contai two buttons to put the LED in ON or off state.

Scanned with CamScanner



Scanned with CamScanner

ii) Control the data input from buttons using PHP page.

- Conclusion:- Thus, we have created simple web interface for Raspberry Pi / Beagle board to control the connected LED's remotely through Interface.

Scanned with CamScanner

Name-Nirant Deore
Div-A
TE Computer
Roll No-31

Date __/__/__ ESIOT Assignment No-12

- Aim :- To Develope a Real time application like smart home with following requirement When user enters into house the required appliances like fan, light should be switched ~~off~~ ON. Appliances should also also get controlled remotely by a suitab web interface. The objective of this applicat is student should construct. Complete smart application in group.

- Theory :- • Basics :- i) Send emails using Python

i) The smtplib module of python is basically all you need to ~~know~~ send simple emails, without any subject line.
ii) But for real emails, you do need a subject line & lots of information.

ii) How to send emails ?

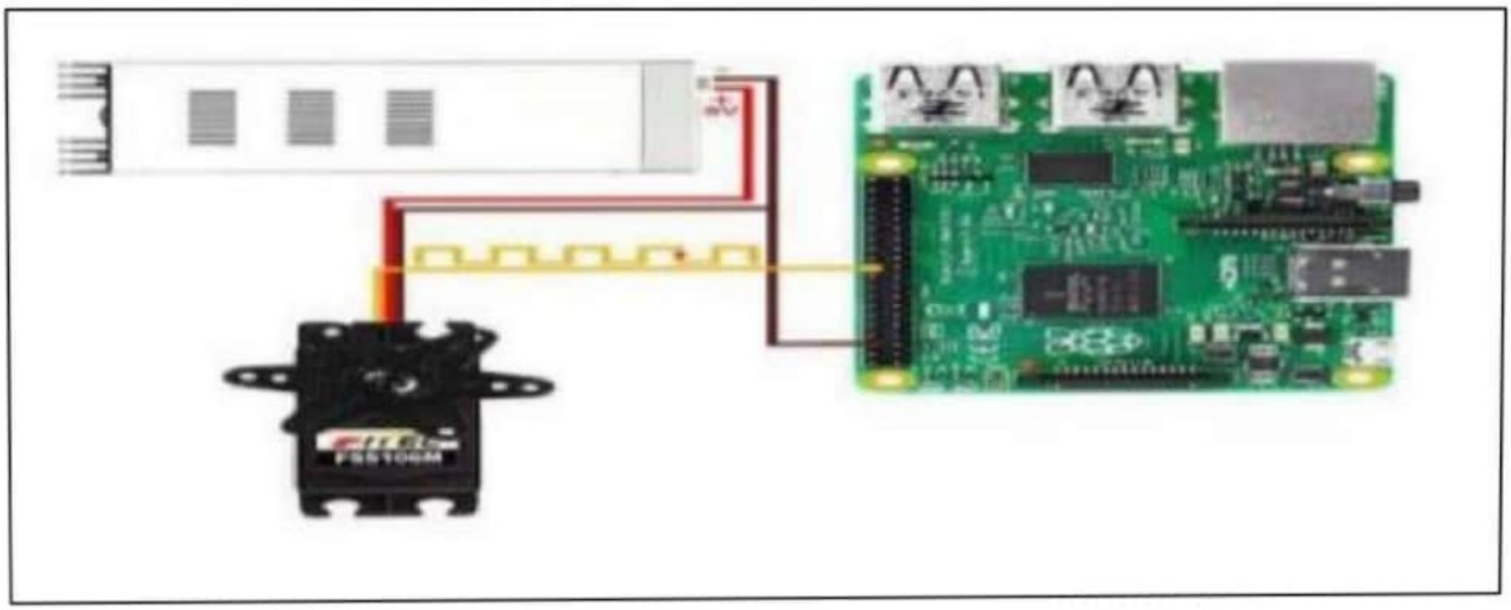
i) Setup the SMTP server and log into your account.
ii) Add your message body
iii) send the message using the SMTP server objeet.

iii) The Smtplib :-

i) The smtplib module defines on SMTP client session object that can be used to send mail to any internet machine with SMTP.

Page No.

Scanned with CamScanner



Scanned with CamScanner

ii) SMTP stands for simple mail Transfer Protocol. The SMTPlib modules is useful for communicating with mail servers to send mail.

iii) Sending mail is done with Python's smtplib using an SMTP server.

iv) Actual usage varies depending on complexity of email & settings of email server, the instructions here are based on sending email through gmail.

- Steps:-

i) Create the lock/unlock application to control the serve motor lock, change its owner & group.

ii) Write application to read the image & send it as email attachment to user.

iii) Write application using HTML-PHP to control the servo motor lock.

- Conclusion:- Thus, we have developed short application for smart home system.

Scanned with CamScanner