Name: Date Vaishnavi B.
Div: A Roll no: 30
(TA-1) ES & IOT Lab.

Experiment No: 1

Aim: Study of Raspberry-Pi, Beagled board, Arduino & other micro controller (History & Elevation)

Theory: Study of Raspberry Pi3 -

The Raspberry Pi is a series of small single board computers developed in the UK by the Raspberry Pi foundation to promote the teaching of basic computer science in the schools & in developing conties.

The first generation (Raspberry Pi1 Model B) was released in feb 2012, followed by the simpler & cheaper model A In 2014, the foundation released a board with an improved design Raspberry Pi1 Model B+.

A Raspberry Pi zero with smaller size & reduced input/output (I/O) & general purpose Input/output (GPIO) capabilites was released in november 2015 & US $5 Raspberry Pi 3 model B was released in feb 2016 & has on-board wifi, Bluetooth & USB boot capabilities

Fig.1. Raspberry Pi3 Kit

History & Elevation:

In 2006, early concept of the Raspberry Pi were based on the Atmel ATmega644 microcontroller. Its schematics & PCB layout are publically available. Foundation trustee Eben upton assembled a group of teachers, academics & computer ethusiasts to device a computer to inspire children.

The first ARM prototype version of the computer was mounted in a package the same size as a USB memory stick.

Study of Beagle Board:

The Beagle board is a low-power open source single-board computer produced by Texas instruments in association with Digi-key & Network element 14. The Beagle board was also designed by open source software development in mind & as a way of demonstrating the Texas instruments OMAP3530 system-on-a-chip.

Study of Arduino:

The Arduino project started at the Interaction Design institute Ivrea (IDII) in Inveatly.

At that time, the students used a Basic stamp microcontroller at a cost of $100, a considerable expense of many students. In 2003 Hrmando Barrage created the development platform wiring as a master's

Fig.2.Beagle Bone Kit

Arduino Kit Fig.3.

theste project at IDII, under the expression supervision of massimo Banzi & Casey Reas who are known for work on the processing language

Following the completion of the Wiring Platform, lighter & less expensive version & were distributed in the open-source community.

In October 2016, Federico musta Arduino's former CEO, secured as 50% ownership of the company. In april 2017, wired reported that musto has "fabricated his academic record on his company's website.

Around the same time, Massimo Banzi announced that the Arduino foundation would be "a new beginning for the Arduino". But a year later, the foundation still hasn't been established & the state of the project remains sun clear. In october 2017, Arduino announced its partnership with ARM Holdings. The announcement said, in part, "ARM recognized independence as a core value of Arduino without any lock-in with the ARM architecture". Arduino intends to continue to work with all technology vendors & architectures.

Conclusion:

Thus, we have studied history of Raspberry Pi, Beagle bone & Arduino.

Name: Date Vaishnavi B
Div: A Roll no: 30
(TA-1) ES&IOT Lab

Experiment NO: 2

Aim: Study of different operating system for Raspberry - Pi.
Understanding the process os installation on Raspberry - Pi.

Theory: **Introduction** -

The Raspberry Pi is a wonderful but power full little computer that fits the palm of your hand. Despite of its size it has enough power to run your ~~oppo~~ operating system smoothly, home media center, a VPN & a lot more.

The Raspberry Pi has a SD card slot for mass storage & will attempt to boot of & that device from SD card when the board is powered on by 5V micro USB supply.

No matter how good & powerfull the hardware of Raspberry Pi is without an operating system it is just a piece of silicon, fiberglass & a few other slimy condition materials.

① **Raspbian** - Currently, Raspbian is the most popular linux based operating system for the Raspberry - Pi. Raspbian is an open source

operating system based on Debian, which has been modified specially for the Raspberry-Pi:
For a beginers it's a good place to star & especially if you are starting with the programming & are used to windows based system as it bears same resembalance to the windows.

(2) Pidoso - After waiting for a long, Raspberry Pi. Users are finally getting an optimised version of Fedora, to replace the current Rasbian as The news caused excitement among the Raspberry Pi community. Who are finally getting the opposturnity to enjoy fedora on their devices after the previous attempt to introduce fedura Remix for Pi ended up as a failure coupled with greater speed & most of features of Fedura 18.

(3) Arch linux - Arch linux is an excellent choice for reasons one of the greatest advantages of the Arch linux distribution is its simplicity in approch & attitude.

④ OSMC - OSMC (open source media center) is a free & open source media player based on the linux. Founded in 2014, OSMC let's you play back media from your local network, attched storage & the internet OSMC is the leading media center in terms of features set & community & is based on the kodi project.

⑤ Retrol PIE - Retrol Pie allows you to turn your Raspberry Pi into a retro-gaming machine. It's platform developed on the base of Raspbian, emulation station, Retro Pie enable you to play your favourite Arode, home-~~for~~ console, classic pc games with minimum setup.

⑥ RISC OS - RISC OS is British operating System originally designed by Acorn Computers lid in cambridge england & was fisst released in 1987. It was specifically designed to run at the ARM chiped. It is fast, compact & efficient. RIS os is not a version of Linux, no is it any way related to windows & interstingly was developed by the original ARM team.

⑧ Kali linux: Kali linux is a Debion based Security auditing linux distibution. It is specially designed for digital forensics & the penetration testing. It is

funded by offering Security Ltd.
Kali Linux provides many pre-installed package with nemerous penetration-testing programs.

Conclusion: Thus, we have studied installation for various OS in Raspberry Pi.

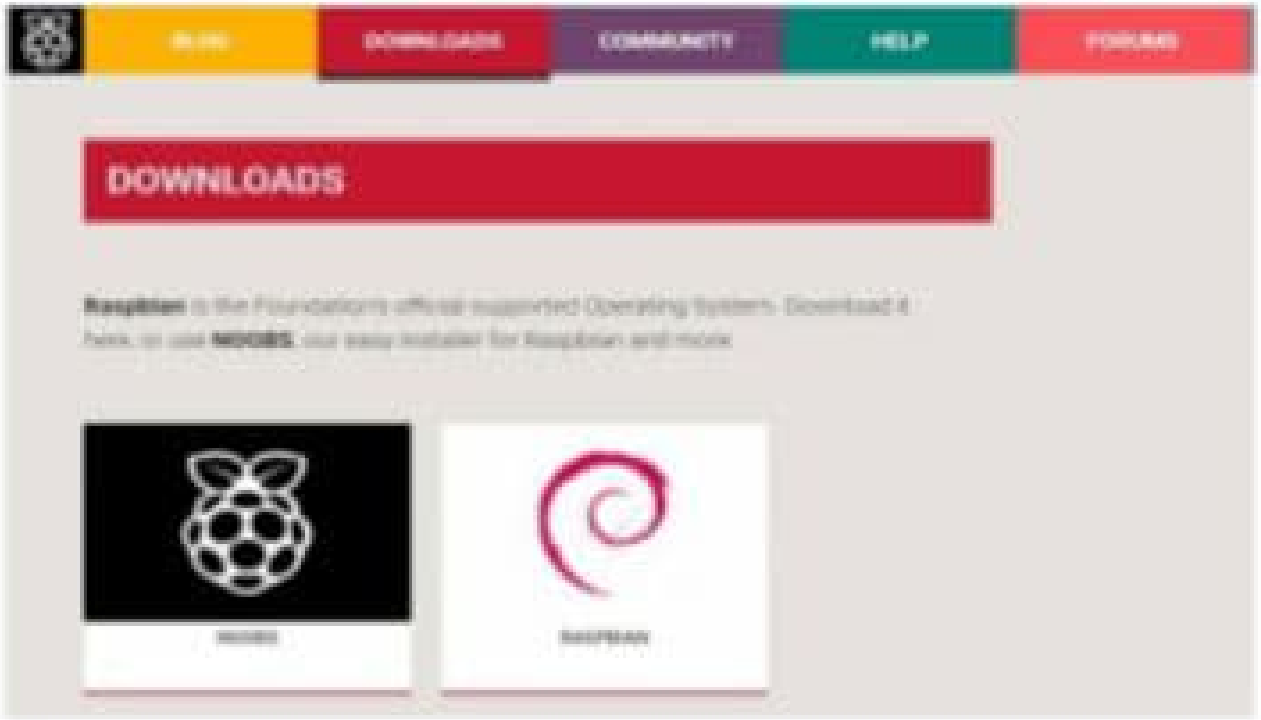
Installing OS for Raspberry - Pi3

Aim: To understand the OS installation for Raspberry Pi3.

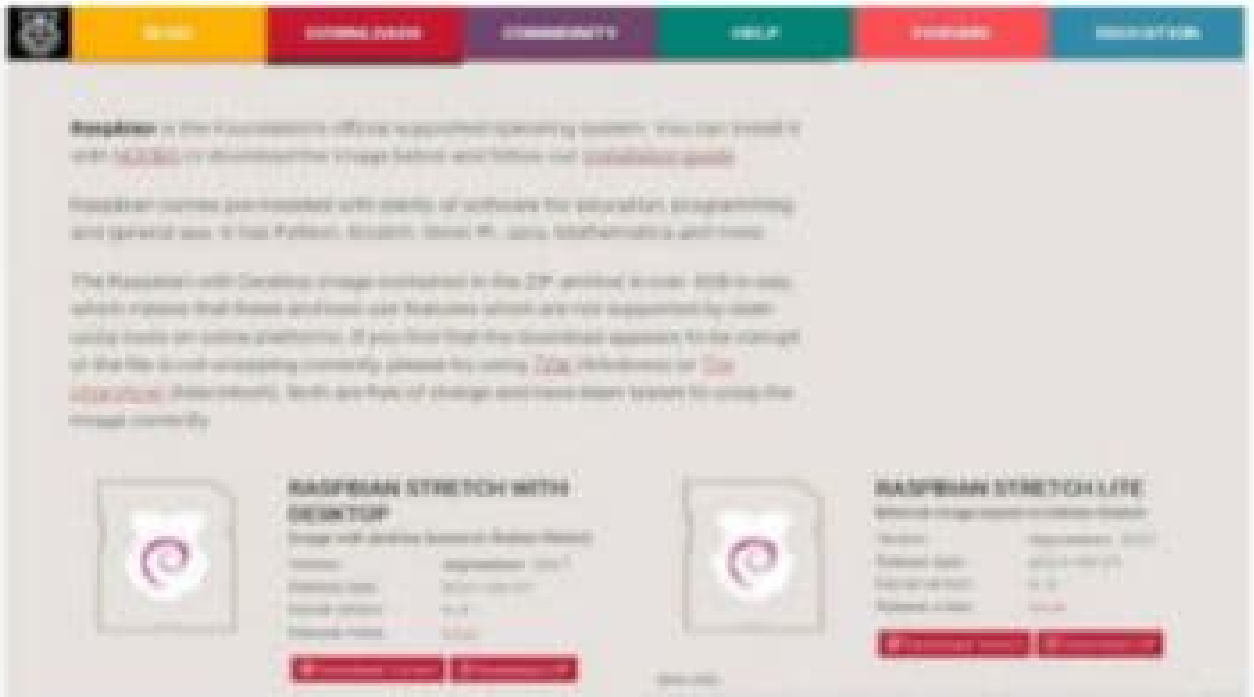
Process of OS installation on Raspberry Pi Board -

1. Open the website www.raspberry.org
2. Click on the "Downloads" tab.
3. Click on the "RASBIAN" option.

3.



4. Click on the "RASPBIAN" option.



5. we requires - "Raspbian stretch with desktop so under this heading" click on Download option
6. A "Torrent file is download
7. But the actual os is present in the zip file of this torrent
8. Now open the "Bit Torrent" software

10. Now open the "Bit Torrent" software.
11. Click on the option "+" and under this click on "Add Torrent".

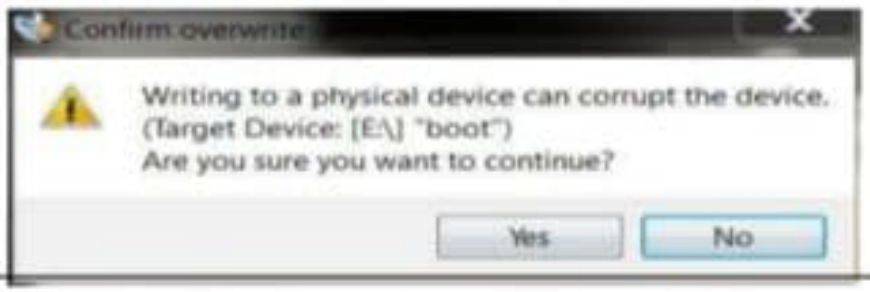
12. Here select the path of downloaded "Torrent file".

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

14. Using this 7Zip software, unzip of the file. After this we get the required disk image of theRaspbian OS (approx. 4GB)
15. Now we have to write this disk image on SD card.
16. To write the OS on SD card, we require the software "win32 disk imager". So download thissoftware and install it.
17. After completion of the installation, the following window appears.

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.



23. Here click 'Yes' and Confirm the overwrite
24. Image file will be written on SD card.
25. After the procedure is completed, it gives "Write Successful" message.
26. Congratulations! Your SD card is ready with your OS to work in the Raspberry-Pi-3 board.
27. Insert this SD card in Raspberry pi3.

Name: Date Vaishnavi B.
Div: A Roll no: 30
(TA-1) FS & IOT

Experiment No: 3

Aim: study of connectivity & configuration of Raspberry-Pi / Beagle board Circuit with basic Peripherals, LED's Understanding GPIO & it's use in the Program.

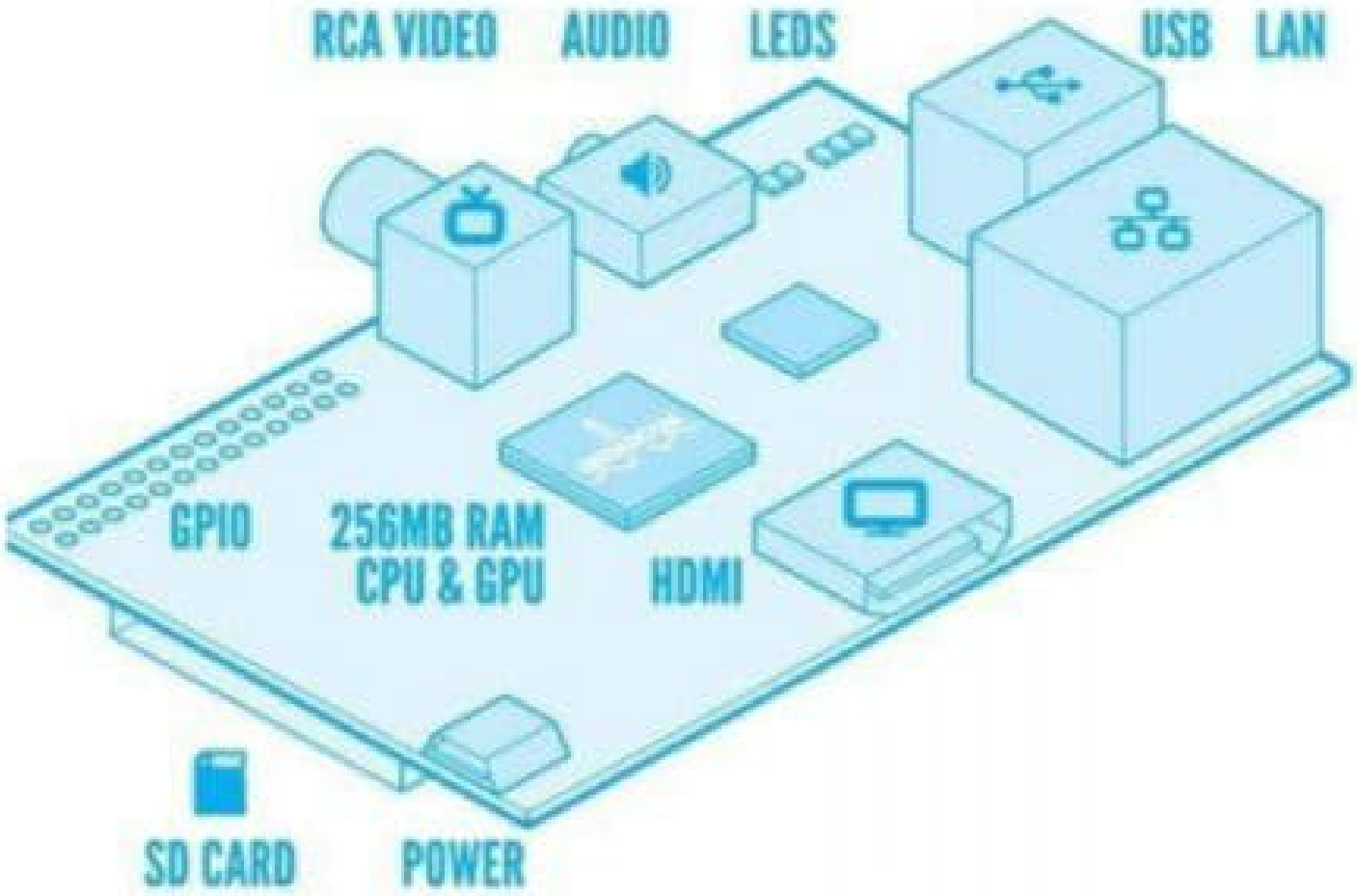
Theory: Connectivity & configuration of Raspberry-Pi

Guides to configure Raspberry-Pi

1. raspi-confi
2. config.txt
3. Wireless
4. Wireless Access Point
5. Audio config.
6. Camera config
7. External storage config
8. Localaisation
9. Default pin config.
10. Device Tress config
11. Kernel Command line
12. VART Configuration
13. Screensaver

Connectivity of Raspberry-pi:

Connectivity is truely suberb for such a tiny device especially on the B version of the Raspberry P There are two USB 2.0 ports that can be used to hook up peripherals or adapters & this can be forther emparded with a powered hub. It's worth nothing that both ports already share the bandwith of signal channel of the system bus.



For video, there's a full-size HDMI port, making the Raspberry Pi compatible with

GPO Mode:-

The GPIO.BOARD option specifies that you are referring to the pins by the number of the pin the plug.

The GPIO.BCM option means that you are referring to the pins by the "Broadcom Soc Channel number, these are the ~~#~~ numbers after "GPIO in the green rectangle around the outsides of below diagrams:

- The model B+ uses the same numbering as the Model Br2.0 & adds new pins (27-40)
- The Raspberry Pi Zero, Pi 2B & Pi 3B use the same numbering as the B+

Building a circuit:

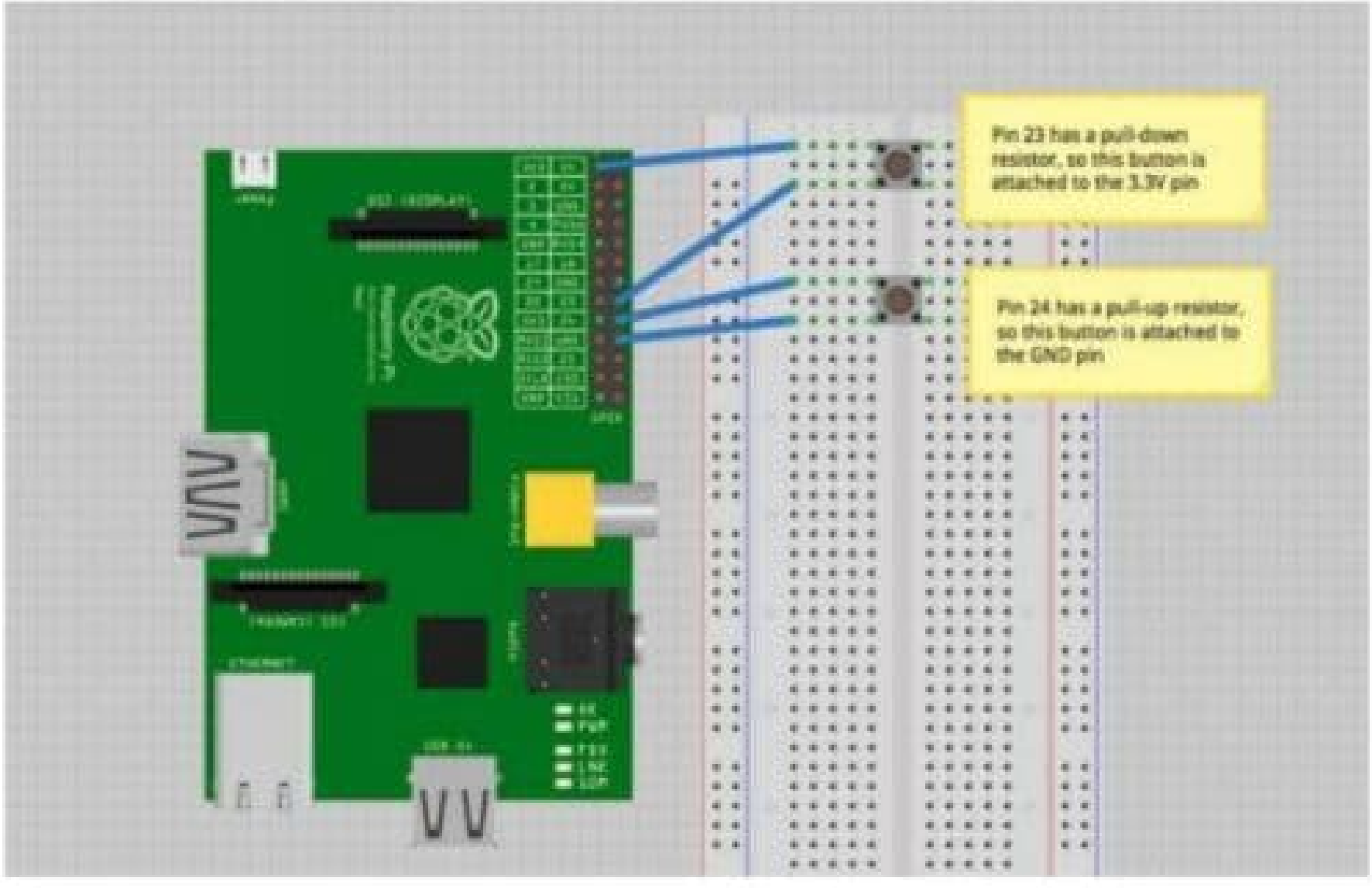
In the circuit shown below. two momentary switches are wised to GPIO pins 25 & 24 (16 & 18 on board). The switch on pin 25 is tied to 3.3V, while the switch on pin 24 is tied to ground.

To set up pins write:

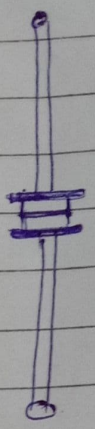
GPIO.setup (23.GPIO.IN, pull-up-down = GPIO PUD-DOWN)

GPIO.setup (24.GPIO.IN, pull-up-down = GPIO PUD-UP)

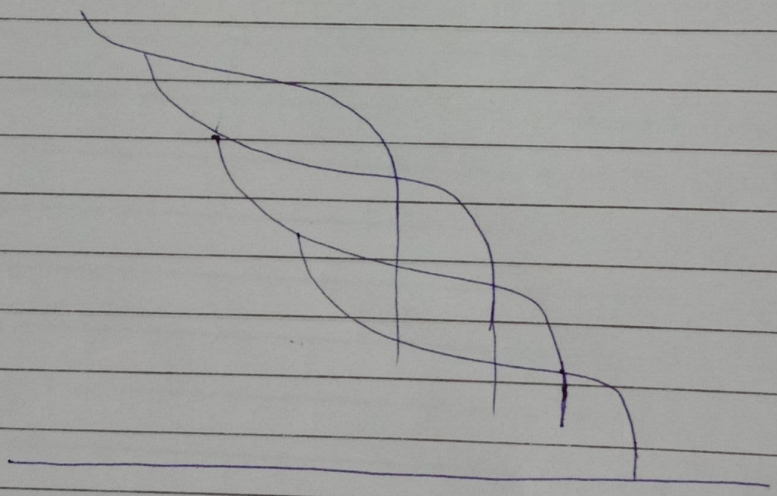
2021.03.19 16:17


Pin 23 has a pull-down resistor, so this button is attached to the 3.3V pin
Pin 24 has a pull-up resistor, so this button is attached to the GND pin

Register: You must Always use registors to connect LEDs up to GPIO pins of the Raspberry Pi. The Raspberry Pi can only supply a small current (about 60mA). The LEDs will want to draw more & if allowed to they will burn out the Raspberry Pi. Therefore putting the registors in the circuit will resume ensure that only this small current will flow & Pi will not be damaged.



Jumper Wires:



Jumpers wires are used on breadboards to 'jump' from one connection to other.

- The ones you will be using in this circuit have different connectors on each end.
- The end with the Pin will go into breadboard.

Conclusion: Thus, we have studied connectivity & configuration of Raspberry Pi & also us...

Name: Date Vaishnavi B.
Div: A Roll no: 30
(TA-1) ES&IOT lab.

Experiment NO: 4

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

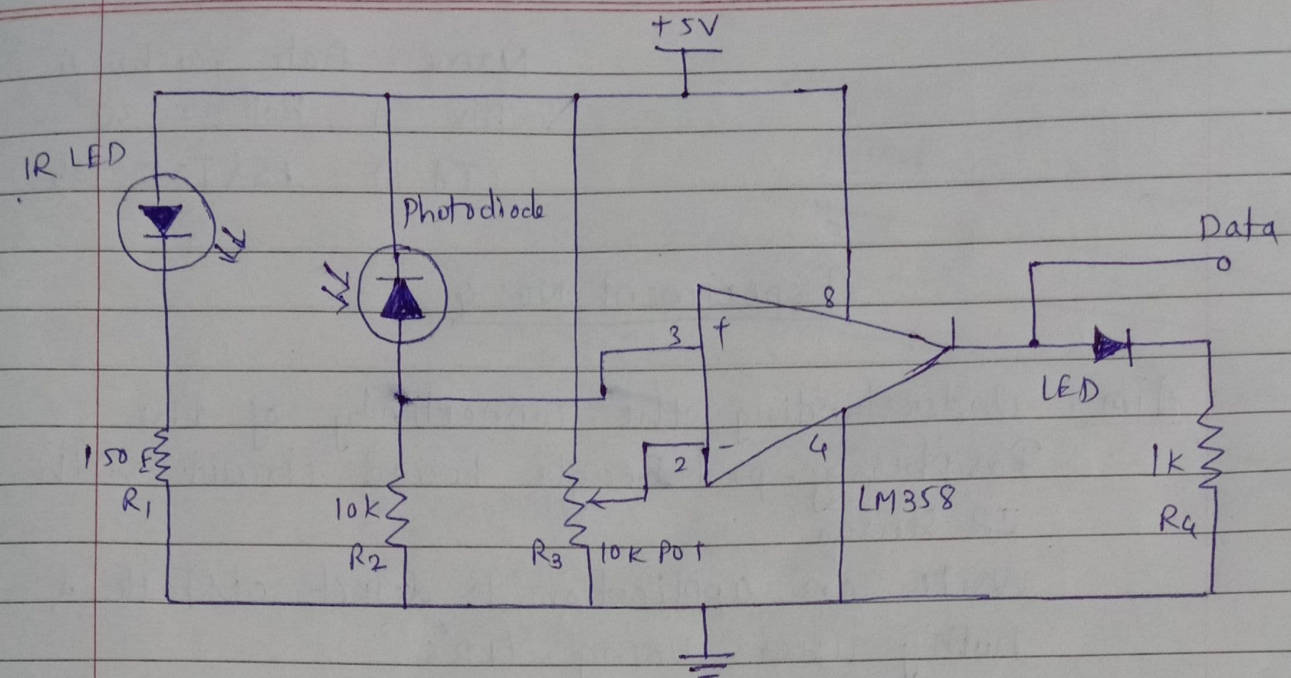
Theory: Components : IR Sensor

1. Emitter - This component continously emits the signal.
2. Receiver - It waits for the signal which is bounced back by obstacle.
3. Indicator - on board LED to signal if obstacle is deduced by the sensor
4. Output - could be used as Input for further processing of the signal.
5. Ground - Ground / Negative Point of the circuit
6. Voltage - Input 3.3 V

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

1. Raspberry Pi 3
2. IR Sensor
3. 1 LED
4. 1 register (33 Ω)
5. Few jumper cables
6. 1 Breadboard.

IR Sensor Fig.1



Circuit Diagram of IR sensor

Circuit : To detect obstacles -

Part 1 : Connecting IR Sensor

IR Sensor has 3 pins, Vin, Vcc, GND & OUT we will use GPIO 17 for receiving input from the sensor

1. Connect GPIO 17 form the Raspberry Pi to Breadboard (5a)
2. Connect out Pin of the Sensor with the Breadboard (5c)
3. Connect GND with negative line on left side of the breadboard
4. Connect GND of IR Sensor to Breadboard (10c)
5. Connect GND from step 3 to Breadboard (10a)
6. connect Vcc of the IR Sensor to Breadboard.
7. Connect 3V3 (Pin #1) to positive line the left side of the breadboard
8. Connect 3V3 to the breadboard (15a)

Now the circuit is complete & sensor will detect the obstacle. It can be tested by putting anything in front of IR sensor

Part 2: Connecting LED

Objective is to turn on the LED when obstacle is detected.

1. Connected GPIO4 follower the board to the breadboared (20a)
2. Connect +ve point of LED to the breadboard (20c)
3. Connect negative point of LED to the breadboared (22c)

Now, we are ready to send signal based on the input received from IR sensor to turn on/off the LED.

Part 3: Code to connect IR Sensor I/P with LED Status.

Part 4: Executing the code

1. open terminal
2. Navigate to directory where the above code is saved
3. Type $ python3 is-obstacle.py & press <enter>

Conclusion:

Thus we done connectivity of Raspberry-Pi/ Beagle board circuit with IR Sensor

Name: Date Vaishnavi B
Div: A Roll no: 50
(TA-1) FS&IOT Lab

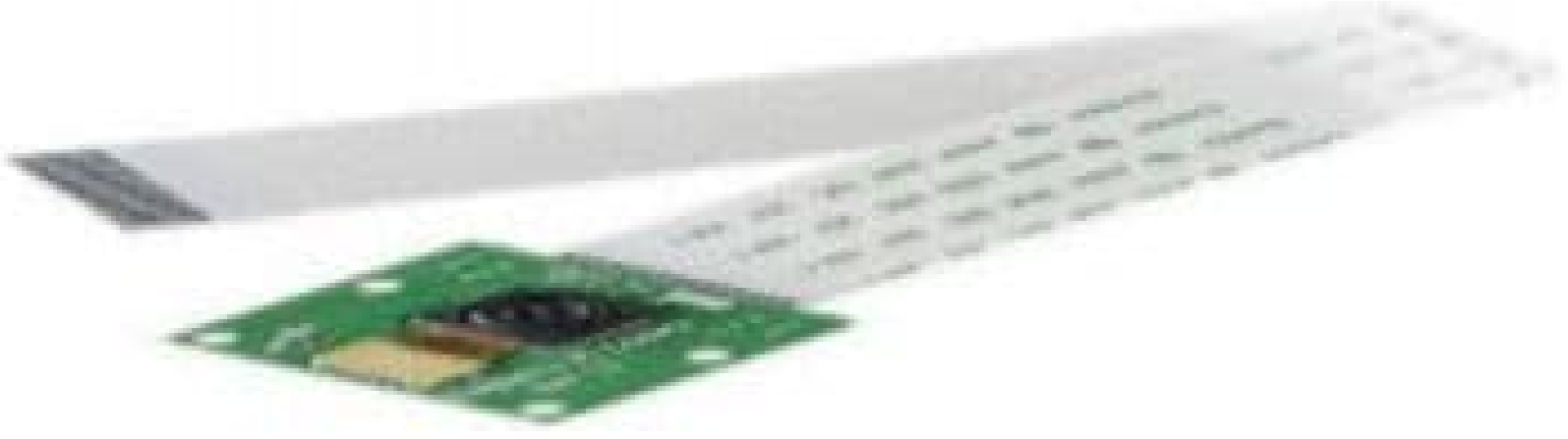
Expeciment No: 5

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

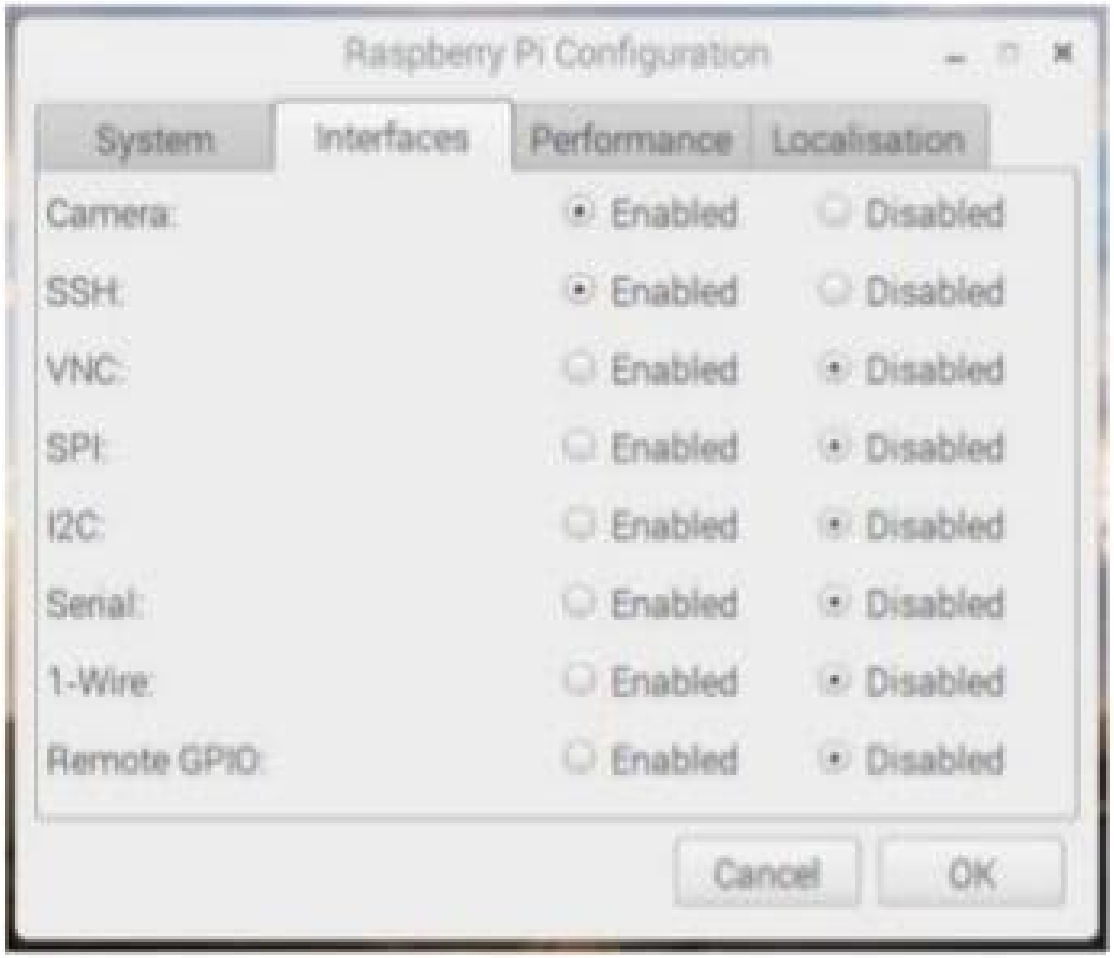
Theory: Raspberry Pi Camera Module V2 replaced the original camera module in april 2016. The V2 camera module has a Sony IMX219 8-megapixel sensor. The camera module can be used to take high-definition video, as well as stills photographs. It's easy to use for begineers, but has plenty to offer advanced users if you're looking to expand your knowledge. We can also use the libraries we bundle with camera to create effects.

It supports 1080p30, 720p60 & VGA90 video modes, as well as still capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi. The Camera works with all models of Raspberry Pi 1, 2, & 3. It can be accessed through the MMAL & V4L APIs & there are numerous third-party libraries built for it, including the Pi-Camera python library. The camera module is very popular in home security applications, & in wildlife camera traps.

Pi Camera



Open Raspberry Pi Configuration and Enable the Camera



Camera preview:

```
form picamera import PiCamera
from time import sleep
camera = PiCamera()
camera.start_preview()
sleep(10)
camera.stop_preview()
```

Rotating the Camera

```
camera.rotation = 180
camera.start_preview()
Sleep(10)
camera.stop.preview()
```

Storing image

```
from picamera import PiCamera
from time import sleep
camera = PiCamera()
camera.start_preview()
sleep(10)
camera.capture('/home/pi/Desktop/img.jpg')
camera.stop - preview()
```

Converting & Playing video

The video format need to get converted to MP4.

```
Sudo apt-get install gpac
```

Now convert video to MP4:

```
MP4BOX -fps 30 -add video.h24 video.mp4
```

<u>Conclusion</u>: Thus, we have studied Pi camera & also stored images & video using Pi camera.

Name: Date Vaishnavi B.
Div: A Roll no: 30
(TA-1) ES & IoT lab.

Experiment NO.6

Aim: understanding & connectivity of Raspberry-Pi / Beagle board with a Zigbee module. Write a network application for communication between two devices using Zigbee.

Theory: ZigBee is communication device used for the data transfer between the controllers, computers, systems, really anything with a serial port. As it works with low power consumption, the transmission distances is limited to 10-100 meters line-of-sight. ZigBee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. ZigBee is typically used in low data rate applications that require long battery life & secure networking. Its main applications are in the field of wirless sensor network based on industries as it requires short-range low-rate wirless data transfer. The technology defined by ZigBee specification is intended to be simpler & less expensive than other wirless networks.


XBee

python Script to perform Zigbee Communication -

```
Import serial
#Enable USB communication
ser = Serial.serial('/dev/ttyUSB0', 9600, TimeOut = .5)
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 between two Raspberry Pi Devices.

Name: Date Vaishnavi B
Div: A Roll no: 30
(TA-1) ES & IOT lab

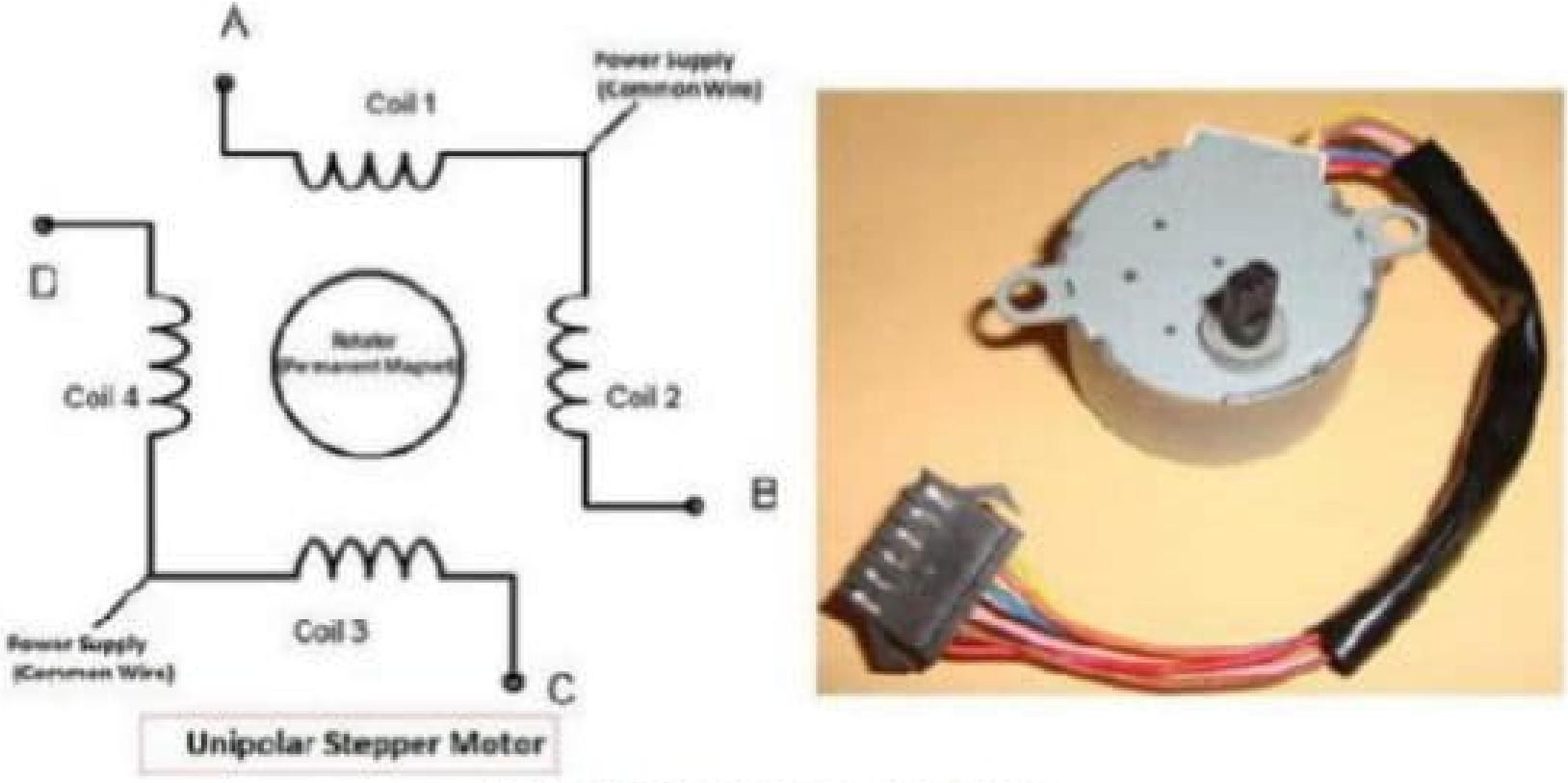
Experiment No: 7

Title: Write an application using Raspberry-Pi / Beagle board to control to operation of Stepper motor.

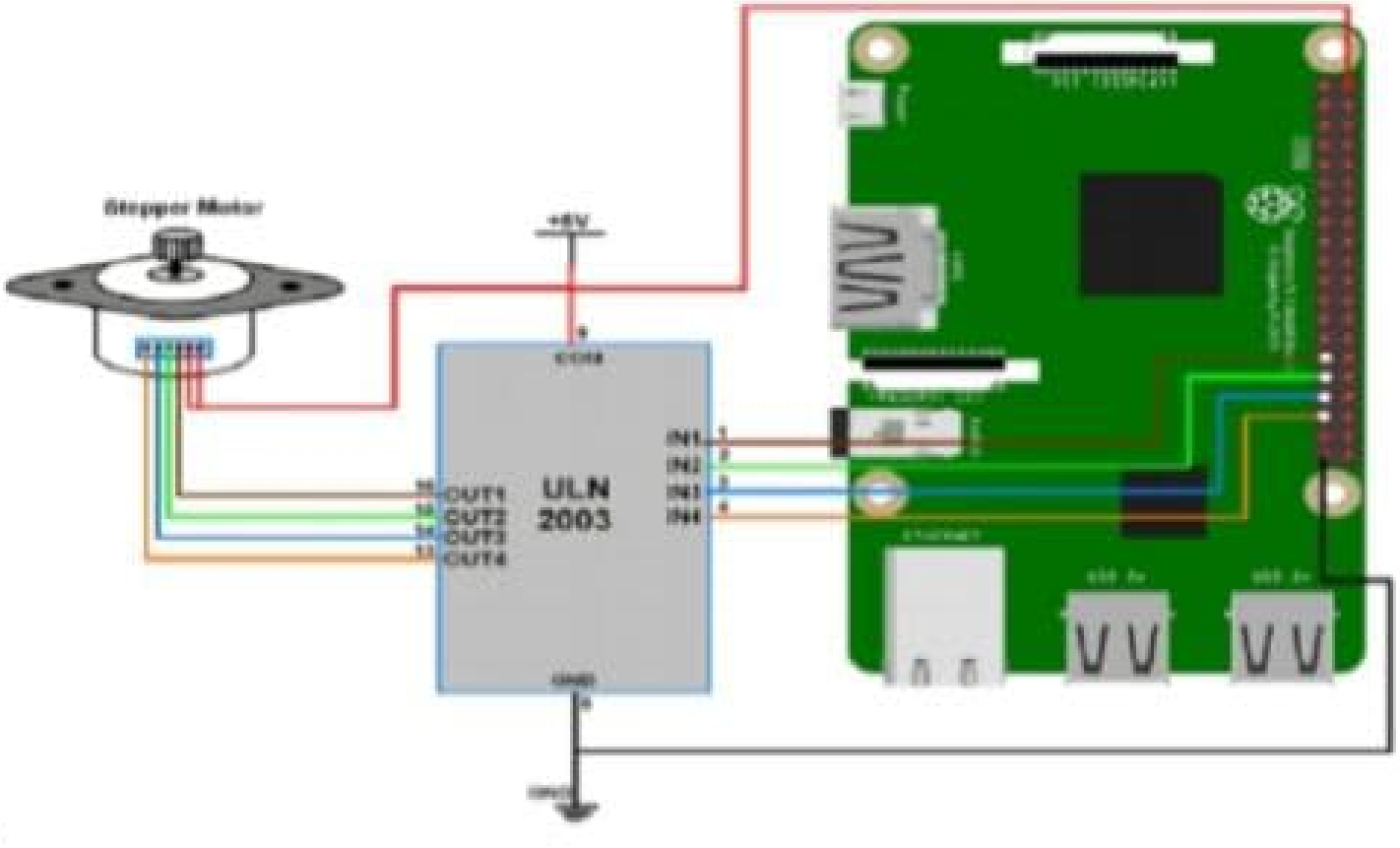
Theory: Stepper Motor

In stepper motor, as the name itself says, the rotation of shaft is in step form. There are different types of stepper motor; in here we will be using the 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 four stages stepper motor, we will deliver power pulses by using stepper motor driver circuit. The driver circuit tasks logic triggers from PC. If we control the logic triggers, we control the power pulses & hence the speed of stepper motor. Search your Stepper Motor model no. to know voltage rating. Depending on the ~~Depen~~ rating choose the secondary source appropriately.



Unipolar Stepper motor Fig.1


Stepper Motor
+5V
ULN
2003
OUT1
OUT2
OUT3
OUT4
IN1
IN2
IN3
IN4

Conclusion: Thus, we have implemented application of stepper motors using Python with Raspberry Pi.

Name: Date Vaishnavi S
Div: A Roll no: 30
(TA-1) ES & IOT Lab

Experiment NO: 8

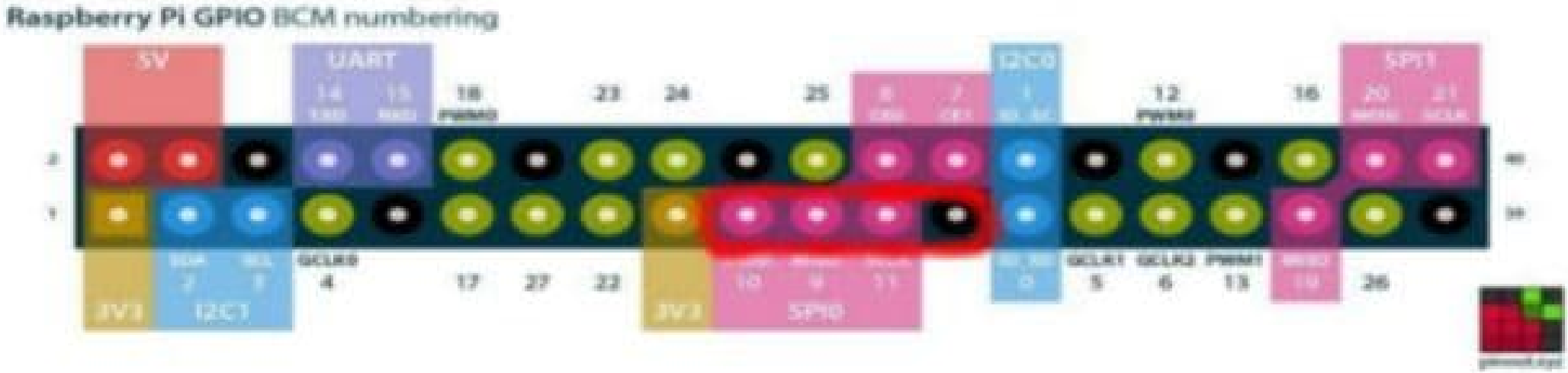
Aim: Write an application using Raspberry-Pi/ Beagle board to control the operation of a hardware simulated traffic signal.

Theory: Attaching the Traffic lights.

The low voltage labs traffic lights connect to the Pi using four pins. one of these needs to be ground, other three 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:

How It Works: The code for this is very simple. It starts by importing the Rpi.GPIO library, plus time which gives us a timed wait function, signal that allows us to trap the signal sent when the user tries to quit the program & sys so we can send appropriate exit signal back to the operating system before terminating.


Raspberry Pi GPIO BCM numbering
5V
UART
14
15
18
PWM0
23
24
25
8
7
I2C0
1
12
PWM0
16
SPI1
20
21
2
1
40
39
3V3
I2C1
SDA
2
SCL
3
GCLK0
4
17
27
22
3V3
10
9
11
SPI0
ID_SD
0
GCLK1
5
GCLK2
6
PWM1
13
MISO
19
26

```
#Setup
GPIO.Setmode(GPIO.BCM)
GPIO.setup(9, GPIO.OUT)
GPIO.setup(10, GPIO.OUT)
GPIO.setup(11, GPIO.OUT)
```

The main body of the code then consists of an infinite while loop that turns on the red light (Pin 9), waits, turns on the amber light (Pin 10), waits, then cycles through the rest of the traffic light pattern by turning the appropriate LEDs on & off.

When control-C is pressed an interrupt signal SIGINT is sent. This is handled by the all lights off function that switches all the lights off, tidies up the GPIO library state & exits cleanly back to operating system.

Conclusion: Thus, we have implemented the application for traffic signals using Raspberry Pi.

Name: Date Vaishnavi B
Div: A Roll no: 30
(TA-1) ES & IOT Lab

Experiment No: 9

Title: Write an application using Raspberry-Pi/ Begalboard to control the operation of a hardware simulated lift elevator lift : elevator simulation using Raspberry pi board.

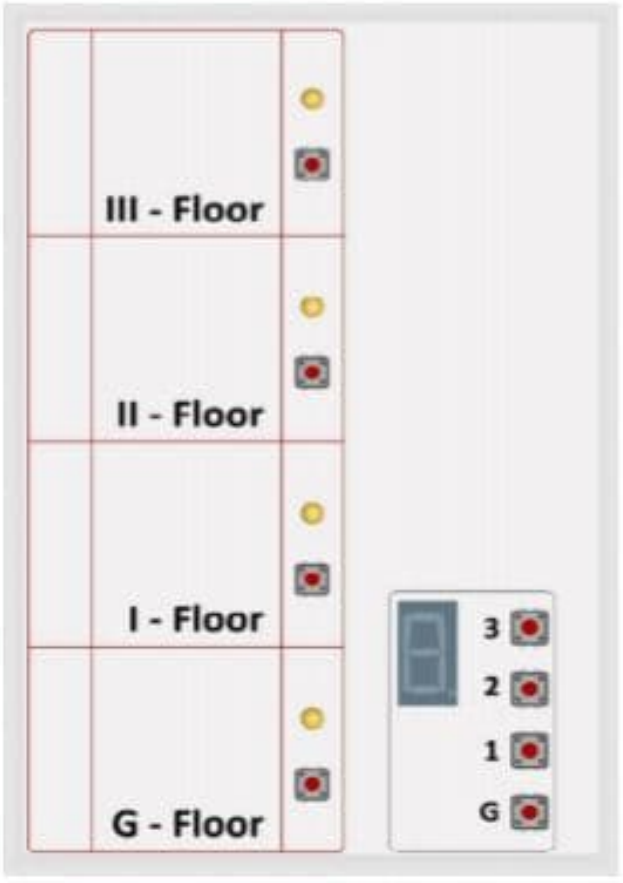
Aim: 1. To understand the working principle of lift elevator.
2. To interface the lift elevator module with Raspberry Pi model.
3. To program the Raspberry Pi model to control operation of lift elevator module.

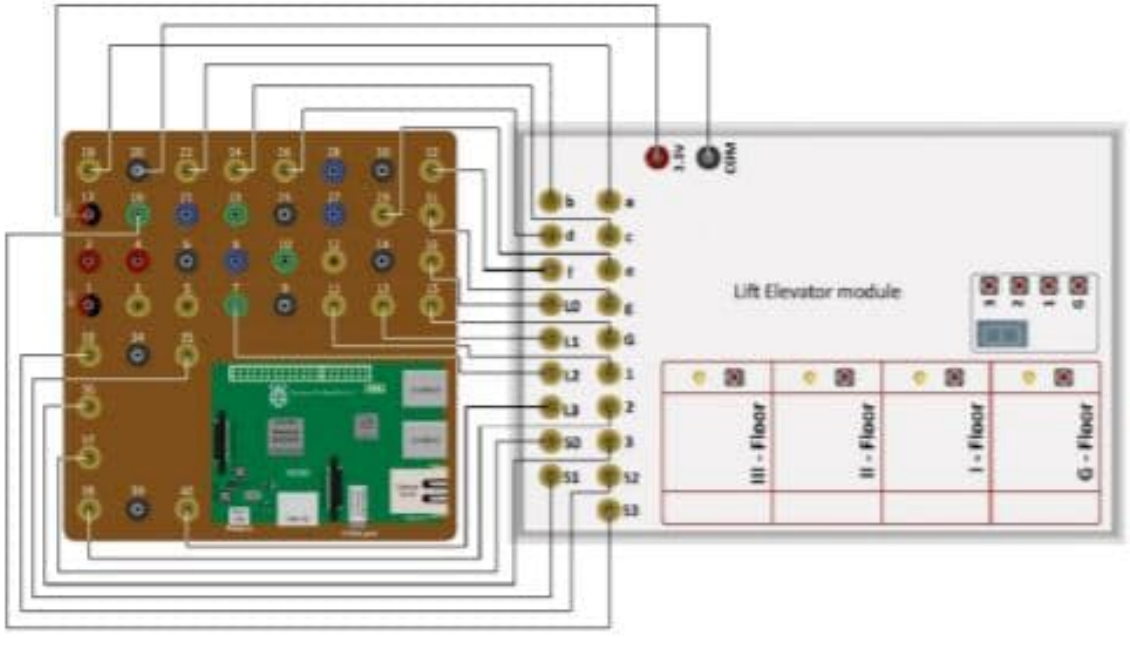
Software: Raspbian OS (IDLE)

Hardware modules: 1 Raspberry Pi Board module
2. Push buttons (qty. 8)
3. Seven Segment Display (qty. 1)
4. leds (qty. 4)
5. Monitor

Theory: Lift Elevator Module has 2 Parts:
1. Moving part inside the lift &
2. Stationary part outside lift at each floor to call the lift.
3. In this simulation module, we have considered four floors of building.

Interface diagram:





4. So moving part contains four push buttons.
5. The moving part also contains seven segment display to indicate current floor number when lift is moving.
6. By pressing one of these buttons, User indicates the destination floor.
7. At each floor, Stationary part contains a buttons for calling the lift
8. In real life, when the lift is called by any floor, Lift starts moving towards particular floor
9. In real life, as soon as entering users get finished, lift door is closed & lift starts moving towards destination.

Procedure: 1. connect all pins of lift elevator module as shown in fig.
2. write program as per algorithm
3. Save program
4. Run code using run module.

Observation: observe the output on LEDs & Seven Segment Display.

Name: Date Vaishnavi B.
Div: A Roll no: 30
(TA-1) ES & IOT Lab

Experiment NO: 10

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

Theory:

IOT cloud Platforms-
1. Kaa IOT platform
2. Site Where: open Platform for IOT
3. Things Speak: An Open IOT Platform with MATLAB analytics
4. DeviceHive: IOT Made Easy
5. Zetta: API - first IOT Platform

Kaa features
1. Manage unlimited no. of connected devices
2. Set up cross-device interoperability
3. Perform A|B Service testing
4. Perform real-time device monitoring
5. Perform remote device provisioning & configuration
6. Collect & analyze sensor data

Sitewhere - features

1. Default database storage is MongoDB
2. Eclipse californium for CoAP messaging
3. InfluxDB for event data storage
4. Grafana to visualize Sitewhere data
5. HBase for non-relational data store.
6. Spring delivers the core configuration framework

ThingSpeak - features

1. Collect data in private channels
2. Share data with public channels
3. RESTful & MQTT APIs
4. MATLAB analytics & visualizations
5. Alerts
6. Event scheduling
7. App integrations
8. Worldwide community

DeviceHive - features

1. Directly integrate with Alexa
2. Visualization dashboard of your choice.
3. Customize DeviceHive behaviour by running your custom js code
4. Connect any device via REST API, websockets or MQTT
5. It comes with Apache Spark & Spark streaming support

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 Twittersphere 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.

Zetta - features

1. Built around Node.js, REST, Websockets & flow-based "reactive-programming".
2. Supports wide range of hacker boards.
3. Zetta allows you to assemble Smartphone apps, device apps & cloud apps.

Conclusion:

Thus, we have designed small application using ThingSpeak.

Name: Date vaishnavi B
Div: A Roll no: 30
(TA-1) ES& IOT Lab

Experiment NO: 11

Aim: Create a simple web interface for Raspberry-Pi Beagle board to control the connected LEDs remotely through interface

Theory:

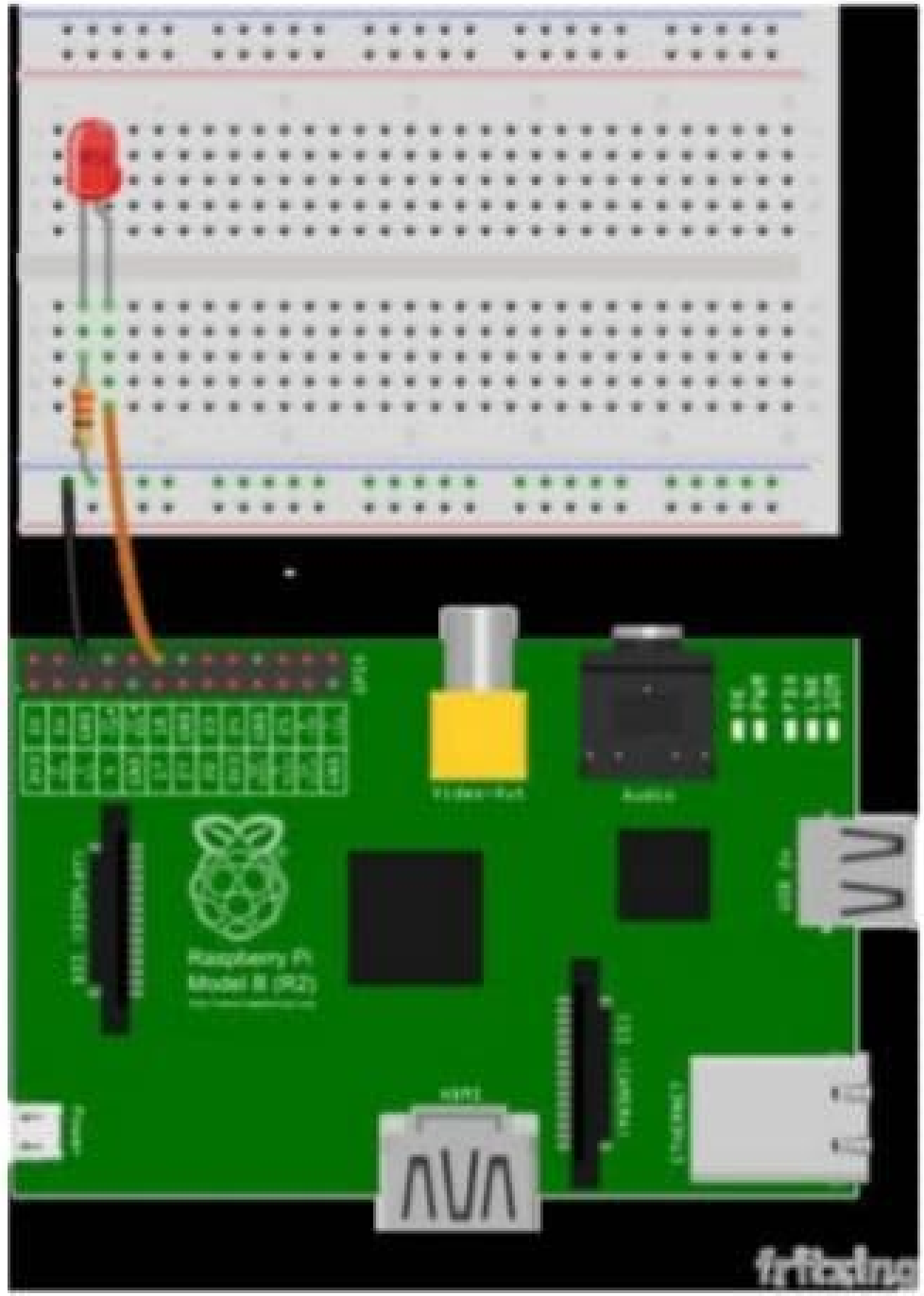
WiringPi - WiringPi is a PIN based GPIO access library written in C for the BCM2835 used in the Raspberry Pi. It's released under the GNU LGPL v3 license & is usable from C, C++ & RTB as well as many other languages with suitable wrappers

Install WiringPi -

```
pi@raspberrypi ~ $git clone git://git.drogon.net/wiringPi
pi@raspberrypi ~ $cd wiringPi
   Pi@raspberrypi~/wiringPi/wiringPi$./build
```

GPIO Command Line utility

1. Glow the LED by value
   gpio write 1 1
2. off the LED by
   gpio write 1 0


Raspberry Pi
Model B (R2)
fritzing

Web Interface to LED

1. Create the front page using HTML which contains two buttons to put the LED is ON or Off State

2. 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 LEDs remotely through the interface.

Name: Date Vaishnavi S
Div: A Roll no: 30
(TA-1) ES & IOT lab

Experiment NO: 12

Aim: Develop a Real time application like Smart home with following requirements: When user enters into house the required appliances like fan, light should be switched ON. Appliances should also get controlled remotely by a suitable web interface. The objective of this application is Student should construct complete smart application in group.

Theory: Basics - Send emails using python :

1. The Smtplib module of python is basically all you need to send simple emails without any subject line or such additional information.

2. But for real emails, you do need a subject line & lots of information - may be even pictures & attachements

3. This is where python's email package comes in Keep in mind that it's not possible to send an email message using email address alone. You need a combination of both email & Smtplib.

How to send emails?

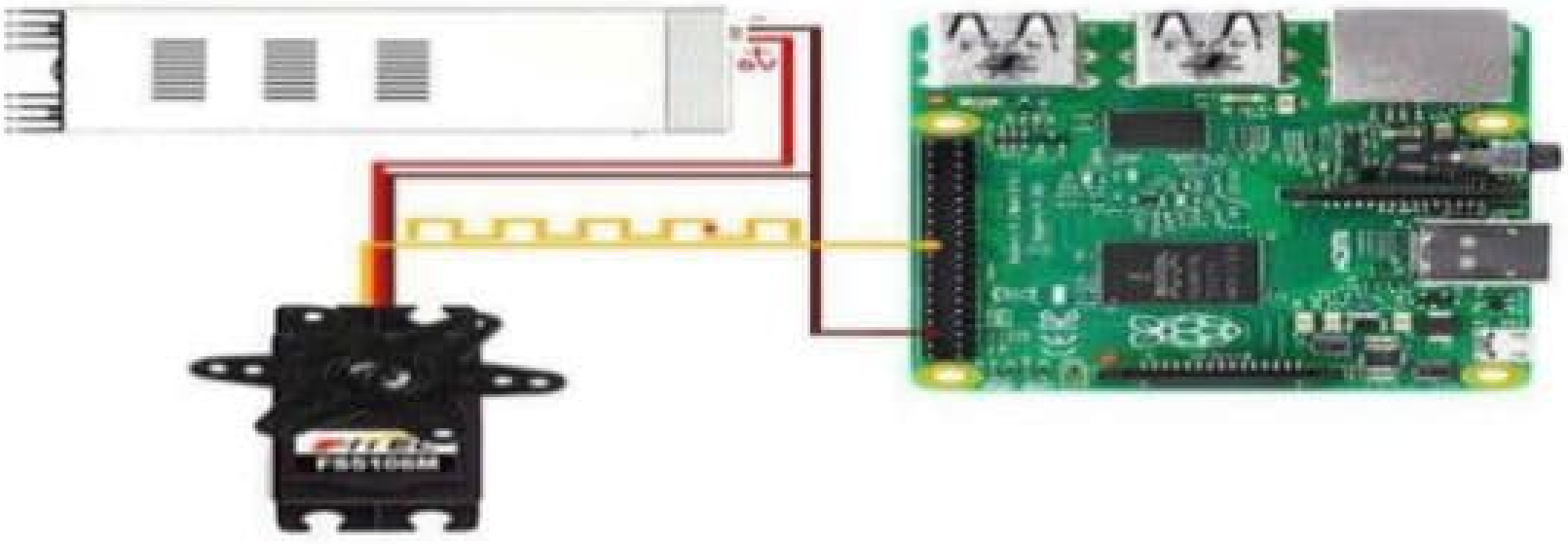
1. Set up SMTP Server & log into your account
2. Create the MIME Multipart meassage object & load it with appropriate headers for from, To, & Subjects fields.
3. Add your message body
4. Send the message using SMTP Server object

Servo Motor: It is a combination of DC motor, Position control system & gears. It mainly has three wires, one is for positive voltage, other is for ground & last one is for position setting. Red wire is connected to power, Brown wire is connected to ground & Orange is connected to signal

Steps: 1. Create the lock/unlock application to control the servo motor lock. Change its owner & group as www-data. Location: /var/www/html
2. Write application to read image & send it as email attachment to user. location: /home/pi
3. write application using HTML-PHP to control the servo motor lock location: var/www/html

Conclusion:

Thus, we have developed short application for Smart Home System.



Fig