Experiment 1

Name :- Bhushan Nikumbhe
Roll. No.23
Div B



Aim :- Study of Rasberry-Pi, Beagle board, Arduino and other micro controller

Theory :-

Stydy of Raspberry Pi3 :-

The Raspberry Pi is a series of small single-board computers developed in the united kingdom by the Raspberry Pi foundation to promote the technique of basic computer science in schools and in developing countries.

A Raspberry Pi Zero with smaller size and reduced input/output (I/O) and general-purpose input/output (GPIO) capabilities was released in November 2015 for US$5. Raspberry Pi 3 Model B was released in Feb 2016 & has on-board wifi, Bluetooth and USB boot capabilities, By 2017, it became the newest mainline Raspberry Pi Zero with wifi and bluetooth for US$10. Processor speed ranges from 700 MHz to 1.2 GHz for the pi 3; on board memory ranges from 256mb to 1GB RAM. Secure digital cards (SD) are used to store operating system and program memory in either SDHC or microcosmic sizes.

The board hase one to four USB ports. For video output, HDMI and composite video are supported, with a standard 3.5 mm phono jack for audio output. Lower level output is provided by a number of GPIO pins which suppart common protocols like I2C. The B-models have an 8P8C Ethernet port and the Pi3 and Pi zero w have on-board wifi 802.11n and bluetooth. prices range US$5 to $35

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 and Newark element14. The Beagle Board was also designed with open source software development in mind, and as a way of demonstrating the texas Instrument's OMAP3530 system-on-a-chip. The board was developed by a small ~~time~~ team of engineers as an educational board that could be used in colleges around world to teach open source hardware & software capabilities.

Study of Arduino:-

The Arduino project started at the Interaction Design Institute Ivrea (IDII), Italy. At that time, the student used as BASIC stamp microcontroller at a cost of $100, a considerable expenses for many students. In 2003 Hernando Barrage created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas, who are known for work on the Processing language. The project goal was to create simple, low cost tools for creating digital projects by non-engineers. The wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller. In 2003, Massimo Banzi, with David Mellis, another IDII student, and David Cuartiells, added support for the cheaper ATmega8 micro controller to wiring. But instead of continuing the work on wiring, they forked the project and renamed it Arduino.

Conclusion:-

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

Experiment 2

Page No. 1

Date

Aim :- Study of diffirent operating stystems for Raspberry Pi. Understanding the process of OS installation on Raspberry - Pi

Theory :-

Introduction :-

The Raspberry Pi is a wonderful but powerful little computer that fits the palm of your hand. Despite of its size it has enough power to run your operating systems smoothly, home media centre, a VPN and a lot more. The Raspberry Pi has a SD card slot for mass storage and will attempt to boot off that device from SD card when the board is powered on by 5v micro USB supply.

Brief Discussion of operating Systems.

No matter how good and powerful the hardware of the Raspberry Pi is without an operating System, it is just a piece of silicon, fibreglass, and a few other semiconducte materials. There are several different operating Systems for the Raspberry Pi including RISC OS, pidora, Arch Linux and Raspbian.

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 specifically for the Raspberry Pi. Raspbian is the default free and open Source operating System that often comes with Raspberry Pi kit, Raspbian is a official operating system of Raspberry Pi foundation. Raspian is a version of debian which is specifically designed and optimized for the raspberry Pi hardware and the build consist more than 35,000 Raspbian packages. Raspbian is still under active development phase with a emphasis on improving the capability, stability and performance. Raspbian comes with python programming and are used for to a windows based System as it bears some resemblance to windows. Raspbian is designed to be easy to use and is recommended operating System for beginners to start off with their Raspberry Pi.

2. Pidora.

After waiting for a long, Raspberry Pi users are finally getting an optimized version of Fedora, the pidora, to replace the current Rasbian OS. The news cause excitement among the Raspberry Pi community, who are finally getting the opportunity to enjoy fedora on their devices after the previous attempt to introduce Fedora Remix for Pi ended up as a failure. However, the seneca centre for Development of open technology (CDOT), the authority group behind Pidora, is confident that the Raspberry Pi community would love the newly optimized OS, coupled with greater speed and most of the features of Fedora 18. The current ~~to~~ Rasbian OS, which was a remix of the open source ~~Deb~~ Debian OS chip based on ARMv6 would make way for Pidora, currently available for download on the CDOT website.

3. Arch Linux

One of the greatest advantages of the Arch Linux distribution is its simplicity in approach and attitude. Arch gives you ability to build your system from the ground up, including only the software you actually need. This minimizes the amount of SD card memory it takes to hold the operating system for your Raspberry Pi, leaving more space for everything else you'll be doing. Arch has now finished it's transition to System D from the ~~old inter~~ old initscripts.

4. OSMC

OSMC (Open Source Media Centre) is a free and open source media player based on Linux. Founded in 2014 OSMC lets you play back media from your local network, attached storage and the internet. OSMC is the leading media centre in terms of feature set and community and is based on the kodi project. Everything is easily managed through the OSMC interface. This OS comes with over 30,000 packages from Debian respository.

5. RetroPIE

Retro Pie allows you to turn your ~~respo~~ Raspberry Pi into a retro gaming machine. Its platform developed on the based of Raspbian, Emulation Station, RetroPie enable you to play your favourite Arcade, home-console, and classic PC games with the minimum set-up. For technocrat users, it also provides a large variety of configuration tools to customize the system as per user need and purpose.

6. RISC OS

RISC OS is a British operating system originally designed by Acorn computers Ltd in Cambridge, England. It was specifically designed to run on the ARM chipset. It is fast, compact and efficient. RISC OS is not a version of Linux, nor it is in any way related to windows and interestingly was developed by the original ARM team. RISC OS PI comes with a small set of utilities and applications

7. Firefox OS

Firefox OS is an OS which is mac associated with being a Linux kernel-based open-source operating system primarily designed for smart phones and tablet computers. It was primarily designed as a community based alternative system utilize open standard, and HTML 5 application, Javascript and open web API's.
This OS is based on Mozilla technology. The device is affordable and flexible as it can ~~be~~ run a number of operating systems and might therefore be a very suitable device to provide an entry level upgrade in network protection.

8. Kali Linux.

Kali linux is a Debian-based security adding Linux distibution. It is specially designed for digital forensics and penetration testing. Kali linux provides many pre-installed packages with numerous penetration-texting programms, like nmap, Wire shak, John the Ripper, Air crack-ng, Burp Suite and OWASP ZAP. Raspberry Pi has changed the way of programming and usability. But without operating System it is just a piece of semi-conductor material. Operating System have made the Raspberry Pi more popular and user friendly. We have gone through 8 different operating System. Each operating has its own features.

Conclusion:-

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

# Experiment - 3



Aim :- Study of connectivity and configuration of Raspberry-Pi / Beagle board circuit with basic peripherals, LEDS, Understanding GPIO and its use in program.

Theory :-
connectivity and configuration of Raspberry-Pi Guides to configure Raspberry Pi.

1. raspi-config
The Raspberry Pi configuration tool in Raspbian, allowing you to easily enable features such as the camera, and to change your specific settings, such as keyboard layout.

2. config.txt
The Raspberry Pi configuration file

3. Wireless
configuring your Pi to connect to a wireless network using the Raspberry Pi 3 and Pi zero w's in built wireless connectivity, or a USB wireless dongle

4. Wireless Access Point
configuring your Pi as a wireless access point using the raspberry

Pi 3 and Pi zero w's in built wireless connectivity, or a USB wireless dongle.

5. Audio config
Switch your audio output between HDMI and the 3.5 mm Jack.

6. Camera config.
Installing and setting up the raspberry Pi camera board.

7. External ~~External~~ Storage config.
Mounting and Setting up external Storage on Raspberry Pi

8. Localisation
Setting up your Pi to work in your local language/time zone.

9. Default pin configuration.
changing the default pin states.

10. Device Trees config.
Device tree, overlays and parameters

11. Kernel Command Line
The linux Kernel accepts a command line of parameter during boot. On the raspberry Pi, this

command line is defined in a file boot partition called cmdline.txt This is a simple text file that can be edited using any text editor e.g. Nano.

sudo nano /boot/cmdline.txt

12. UART configuration

The SoCs used on the raspberry Pis have two built-in UARTs, a PL011 and a mini UART. They are implemented using different hardware blocks, so they have slightly different characteristics. However, both are 3.3v devices, which means extra care must be taken when connecting up to an RS232 or other system that utilizes different voltages levels. An adapter must be used to convert the voltage levels between the two protocols. Alternatively, 3.3v USB UART adapters can be purchased for very low prices.

Connectivity of Raspberry Pi:-

connectivity is truly superb for such a tiny device, especially on the B version of the Raspberry Pi There are two USB 2.0 ports that can be used to hook up peripherals or adaptas, and this can be further expanded with a powered hub. It's worth noting that both ports already share the bandwidth of a single channel to the system bus.

GPO Mode

The GPIO. BOARD option specifies that you are referring to the pins by the number of the pin the plug i.e. the number's printed on the board and in the middle of the diagram below.

Unfortunately the BCM numbers changed between variables of the pi1 model B.

- The Model B+ uses the same numbering as the Model B 2.0, and adds new pin (board numbers 27 - 40).
- The raspberry Pi zero, Pi 2B and Pi 3B use the same numbering as the B+.

Building a Circuit:-

Two momentary switches are wired to GPIO pins 23 and 24 (pins 16 and 18 on the board). The switch on pin 23 is tied to 3.3v, while the switch on pin 24 is tied to ground. The reason for this is that the Raspberry pi as internal pull up and pull down resister that can be specified when the pin declarations are made.

Resister :-

You must always use resistors to connect LEDs up to the GPIO pins of the Raspberry Pi the Raspberry Pi can only supply a small current (about 60mA). The LEDs will want to draw more, and if allowed to they will burn out the raspberry Pi. Therefore putting the resistors in the circuit will ensures that only this small current will flow and the Pi will not be damaged.

Resistors are a way of limiting the amount of electricity going through a circuit specifically, they limit the amount of current that is allowed to flow. The measure of resistance is called the ohm (Ω). and the larger the resistance,

the more it limits the current.
The value of resistor is marked with colored bands along the length of the resistor body.

Jumper Wires :-
Jumper wires are used on bread-boards to 'jump' from one connection to other.
• The one you will using in this circuit have different connectors on each end.
• The end with the 'pin' will go into the bread board.
• The end with the piece of plastic with a hole in it will go onto the Raspberry Pi's GPIO pins

Conclusion :-
Thus, we have studied connectivity and configuration of Raspberry Pi and also use of GPIO.

# Experiment-4

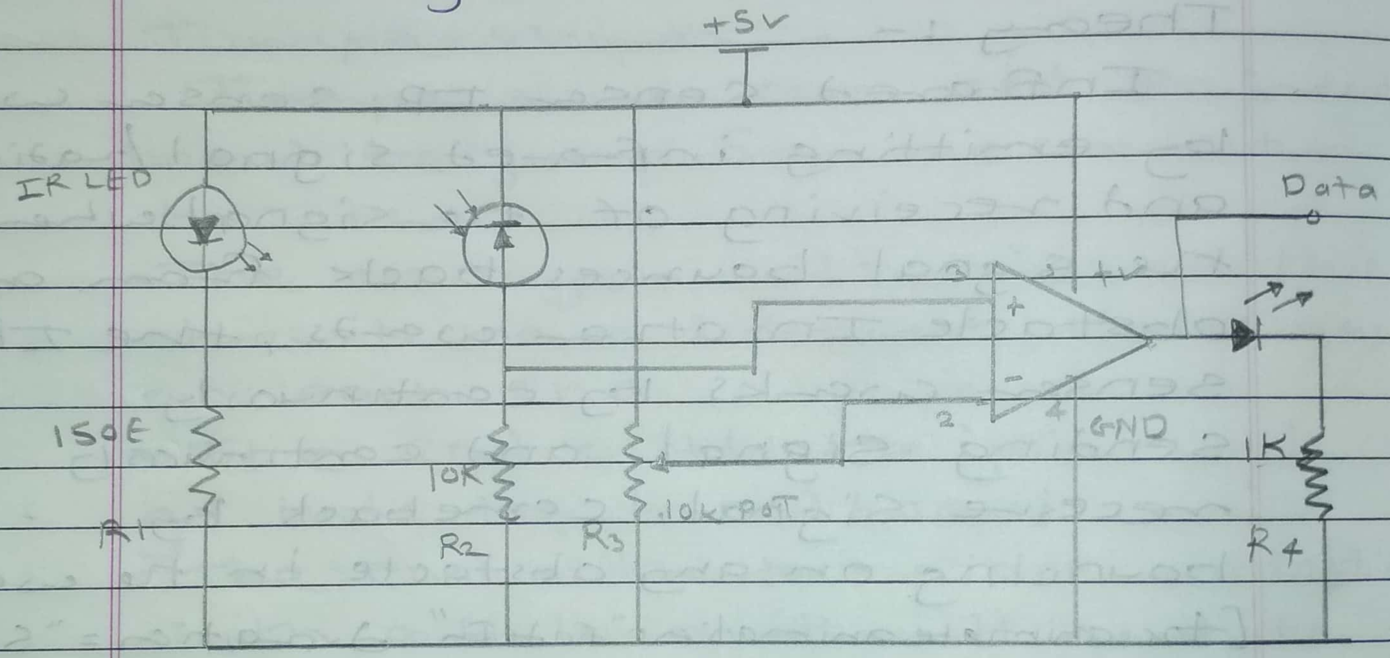


Aim:- Understanding the connectivity of Raspberry-Pi/Beagle board circuit with IR sensor. write an application to detect obstacle and notify ~~the~~ user using LED's

Theory :-
Infrared sensor IR sensor work by emitting infrared signal/radiation and receiving of the signal when the signal bounces back from any obstacle In other words, the IR sensor works by continuosly sending signal and continuosly receive signal, comeback by bouncing on any obstacle in the way.
[tx animate animation="fadeIn" duration="5" delay="4." inline="no"]
[/tx animate]

Components : IR Sensor
1. Emitter: This component continuously emits the infrared signal
2. Receiver : It awaits for signal which is bounced back by obstacle
3. Indicator : on board LED to Signal if obstacle is deducted by the sensor.

4. Output : Could be used as Input for furthe processing of the signal.
5. Ground : Ground / Neagtive point of the circuit.
6. voltage : Input 3.3v




Circuit Diagram of IR sensor

Objective:-
We will be creating a circuit using following components to detect obstacle.
1. Raspberry Pi 3
2. IR (Infrared) sensor
3. 1 LED
4. 1 Resistor (330Ω)
5. Few ~~jm~~ jumpe cables
6. 1 Breadboard.

Circuit: To detect obstacles
We will be creating a circuit which will turn on the LED when an obstacle is detected. And as soon as the obstacle is removed from the way the LED will turn off. In order to achive that follow the steps to create required circuit.

Part 1: Connecting IR sensor
IR sensor has 3 pins, viz VCC, GND and OUT. we will use GPIO 17 for receiving input from the sensor.
1. connect GPIO 17 from the Raspberry Pi to Breadboard (5a)
2. connect OUT pin of the sensor with the Breadboard (5c)
This will send input received from sensor to GPIO 17, which will be processed further.
3. Connect GND with negative line on the left side of the breadboard
4. connect GND of the IR sensor to Breadboard (10c).
5. Connect GND from step 3 to breadboard (10a)
6. connect VCC of the IR sensor to breadboard (15c).
7. Connect 3v3 (pin#1) to positive line on left side of the breadboard.

8. connect 3v3 (connected in step7) to the Breadboard (15 a).

Now the circuit is complete and sensor will detect the obstacle. It can be tested by putting anything in front of the IR sensor on board LED will be on if obstacle is put in front of the sensor, else it will be off.

Part 2 : Connecting LED
Objective is to turn on the LED when obstacle is detected.

1. connect GPIO 4 from the board to the breadboard (20 a)
2. connect positive point of the LED to the breadboard (20 c)
3. connect negative point of the LED to the breadboard (22 e)
4. Use resister (330Ω) to connect negative to the negative point of the LED (22 a)

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

```
from GPO zero import LED
from signal import pause
import R pi. GPO as GPO
import time
GPO. set mode (GPIO.BCM)
LED_PIN = 27
IR_PIN = 17

indicator = LED (LED_PIN)
GPIO. setup (IR_PIN, GPIO.IN)

count = 1

while True:
  got_something = GPIO.input (IR_PIN)
   if got_something:
     indicator. on ()
     print ("{:73} Got something".format
                                              (count))

  else:
    indicator. off ()
    print ("{:3} Nothing detected". format (count))
    count += 1

time. sleep (0,2)
```

Part 4: Executing the code

1) Open terminal
2) Navigate to the directory where the above code is saved
3) Type $ python3 ir_obstacle.py and press<enter>

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

conclusion:

Thus, we done connectivity of Raspberry-Pi / Beagle board circuit with IR sensor.

# Experiment 5



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

Theory :-

Raspberry Pi camera Module the 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 ~~tools~~ video, as well as photographs. We can also use the libraries we bundle with the camera to create effects.

It Support 1080p30, 720p60 and 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, and 3. It can be accessed through the MMAL and V4L APIs and 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 and in wildlife camera traps.

PI camera

Camera Preview

```
from 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 the image:

```
from picamera import Picamera
from time import sleep
camera = Picamaa()
camera.start_preview()
sleep(10)
camera.capture("/home/pi/Desktop/image1.jpg")
Camera.stop_preview()
```

Recording the video

```
from pi camera import picamera
from time import sleep
camera = Picamera()
camera.start-preview()
camera.start-recording("/home/pi/video.h264")
sleep(10)
camera.stop-recording()
camera.stop_preview()
```

converting and playing video

The video format need to get converted to MP4, so install gpac

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

Now convert the video to MP4

```
MP4Box -fps30 -add video.h264 video.mp4
```

Conclusion :-

Thus, we have studied Pi camera and also stored the images and videos using Pi camera.

# Experiment 6

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

Theory :-
ZigBee is a communication device used for data transfer between the controller, 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 and secure networking. In main applications are in the field of wireless sensor network based on industries as it requires short range low-rate wireless data transfer. The technology defined by the ZigBee Specification intended to be simpler and less expensive than other wireless networks.

Here we make use of an interface of ZigBee with RaspBerry Pi2 for a proper wireless communication. Raspberry Pi2 has got four USB ports. So it is better to use a ZigBee Dongle for this interface. Now we want to check the communication between the two paired ZigBee modules.

Python script to perform zigbee communication

```
Import serial
# Enable USB communication
Ser = Serial.serial("/dev/ttyUSB0", 9600, Timeout=.5)
while True:
    Ser.write("Hello User2\n") #write a data
    incoming = Ser.readline().strip()
    print "Received Data:" + incoming
```

Conclusion:

Thus, we have done Zigbee communication between two Raspberry Pi Devices.

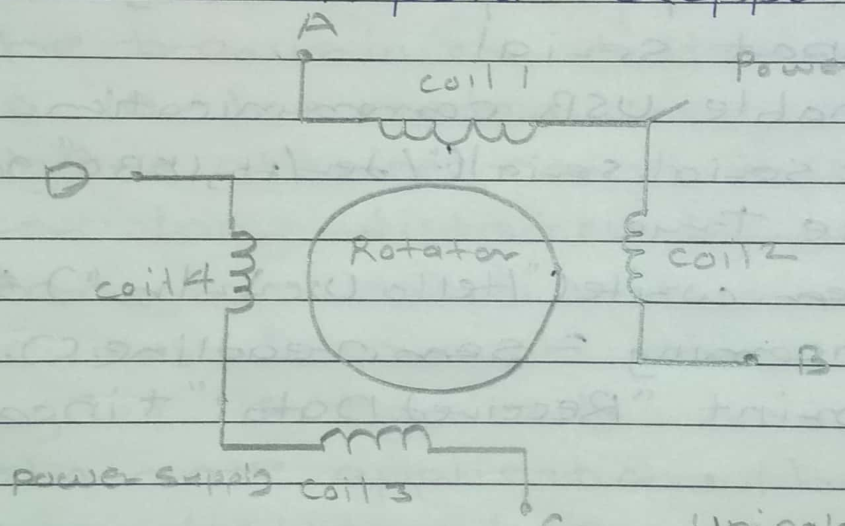
# Experiment 7

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

## Theory :-

### Stepper Motor :-

In Stepper motor, as the name itself says, the rotation of shaft is in ~~left~~ step form. There are different types of stepper motor; here we will be using the most popular one that is Unipolar Stepper Motor.




Unipolar stepper motor

To rotate this Four Stage Stepper Motor, we will deliver power pulses by using Stepper Motor Driver circuit. The driver circuit takes logic triggers from PI. If we could the control the logic triggers, we control the power pulses and hence the speed of stepper motor.

There are 40 GPIO output pins in Raspberry Pi 2 But out of 40 only 26 GPIO pins can be programmed. Some of these pins perform some special functions, with special GPIO put aside, we have only 17 GPIO remaining. Each of these 17 GPIO pin can deliver a maximum of 15mA current And the sum of currents from all GPIO Pins cannot exceed 50mA.

There are +5V and +3.3v power output pins on the board for connecting other modules and sensors. These power rails cannot be used to drive the stepper Motor, because we need more power to rotate it. So we have to deliver the power to rotate stepper motor from another power source. My stepper motor has a voltage rating of 9v so I am using a 9v battery as my second power source. Search your stepper motor model number to know voltage rating. Depending on the ratings, choose the secondary source approximately.

Sample program

Python Program

Steppe motor interfacing with Ralpbari pi

```
import RPi.GPIO as GPIO
from the import sleep
import sys

# assign GPIO pins for motor
motor_channel = (29,31,33,35)
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
# for defining more than 1 GPIO channel as I/P
GPIO. ~~ste~~setup(motor_channel, GPIO.OUT)

motor_direction = input('select moter direction a = anticlockwise, c= clockwise: ')
while True:
 try:
 if (motor_direction == 'c'):
 print('moter running clockwise\n')
GPIO.output(motor_channel, (GPIO.HIGH, GPIO.LOW, GPIO.LOW, GPIO.HIGH))
 sleep (0.02)
GPIO.output(motor_channel, (GPIO.HIGH, GPIO.HIGH, GPIO.LOW, GPIO.LOW))
 sleep (0.02)
GPIO.output(motor_channel (GPIO.LOW, GPIO.HIGH, GPIO.HIGH, GPIO.LOW)),
sleep (0.02)
```

```
GPIO.output(motor_channel, (GPIO.LOW, GPIO.LOW, GPIO.HIGH, GPIO.HIGH))
sleep(0.02)

elif (moter_direction == 'a'):
  print('motor running anti-clockwise\n')

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

conclusion :-

Thus, we have implemented application of stepper Motors using Python with Raspberry Pi.

# Experiment 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, the other three being actual GPIO pins used to control each of the individual LED's.

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 package needed to allow you to download my sample code, and to give python access to the GPIO pins on the pi. Enter the following at the 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, 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 and sys so we can send an appropriate exit signal back to the operating system before terminating.

```
import RPi.GPIO as GPIO
import time
import signal
import sys
```

Next we put the GPIO library into "BCM" or "Brodcom" mode and sets pin 9 (red LED), 10 (amber LED) and 11 (green LED) to be used as output:

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

The main pat of the program will run in an infinite loop until the user exits it by stopping python with ctrl c. It's good idea to add a handle function that will run whenever this happens, so that we can turn off all the lights prior to exiting.

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

The main body of the code then consists of an infinite loop that turns on the red light (pin 9), waits, turns on the amber light (pin 10), waits, than cycles through the rest of the traffic light pattens by turning the appropriate LED's on and 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 and exits cleanly back to the operating system.

Conclusion :-

Thus, we have implemented the application for traffic ~~high~~ signal using Raspberry Pi.

Experiment 9



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

Theory :-

IOT platform :-

The IOT platforms are suites of component those help to setup and manage the internet connected device. A person can remotely collect data, monitor, and manage all internet connected devices from a single system. There are a bunch of IOT platforms available online but building an IOT solution for a company is all depend on IOT platform host and suppot quality.

IOT cloud Platforms :-

1. Kaa IOT platform.
2. SiteWhere open platform for the Internet of Things.
3. Thingspeak : An open IOT platform with MATLAB analytics
4. DeviceHive. IOT Made Easy
5. Zetta : API - First Internet of Things platform.

Kaa - Features
1. Manage an unlimited number of connected devices.
2. Set up cross device interoperability
3. Perform A/B Service testing.
4. Perform real-time device monitoring
5. collect and analyze sensor data.
6. Analyze user behavior deliver targeted notifications.
7. Create cloud services for smart ~~po~~ products.

Sitewhere features:-
1. Run any number of IOT applications on a single site where instance
2. Spring delivers the core configuration framework
3. Connect devices with MQTT, AMQP, stomp and other protocols.
4. Add devices through self-registration, REST Services, or in batches
5. Default database storage in MongoDB
6. Eclipse californium for CoAP messaging
7. InfluxDB for event data storage.
8. Grafana to visualize sitewhere data
9. HBase for non-relational data store.

ThingSpeak - Features
1. collect data in private channels
2. Share data with public channels
3. RESTful and MQTT APIs.
4. MATLAB analytics & visualization.
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 behavior by running your Customjavascript code.
4. It supports the Big data solutions such as ElasticSearch, Apacespark, Cassandra and Kafka for real-time & batch processing
5. connect and device via REST API, websockets or MQTT.
6. It comes with Apachespark and Spark streaming support.
7. Supports libraries written in various programming language, including Android and iOS libraries.
8. It allows running batch analytics and machine learning on top of your device data.

Zetta- Features
1. Built around Node.js, REST, websockets and a flow-based "reactive programming"
2. Support wide range of hacker boards
3. Zetta allows you to assemble smartphone apps, device apps and cloud apps.

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

# Experiment 10



Aim:- Create a simple web interface for Raspberry-Pi/Beagle board to control the connected LED remotely through the 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 LGPLv3 License and is usable from C, C++ and RTB as well as many other languages with suitable wrappers.

Install WiringPi

WiringPi is not included with Raspbian, so, to begin, you'll need to download and install it. That means your Pi will need connection to the internet - either via Ethernet or wifi. We can do using Git to download the latest version.

pi@raspberrypi-$git clone git://git.drogon.net/wiringPi

pi@raspberrypi-$cdwiringPi pi@raspberrypi ~/wiringpi/wiringpi$/build.

GPIO command line utility:-
Task: connect the LED GND to short pin GPIO18 to long pin
Remember: GPIO18 is PIN 1 in wiring PI
GPIO Command Line Utility
1. Glow the LED by value
gpio write 1 1
2. off the LED by
gpio write 1 0

Web Interface to LED
1. create the front page using HTML which contains two buttons to put the LED in ON or OFF state.
2. Control the data input from buttons using PHP page

Conclusion:-
Thus, we have created simple web interface for RaspberryPi/Beagle board to control the connected LEDs remotely through the interface.

# Experiment 11

Page No. 1
Date

Aim:- Develope 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 suitable web interface.

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 and lots of information maybe even pictures and attachments.
3. This is where ~~email~~ Python's email package comes in. keeps in mind that it's not possible to send an email message using the email package alone. You need a combination of both email & smtplib.

How to send emails?

1. Set up the SMTP server and log into your account.

2. Create the MIME Multipart message object and load it with appropriate headers for From, To, and subject fields.
3. Add your message body.
4. Send the message using the SMTP server object.

The smtplib

1. The Smtplib modules defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener ~~dow~~ daemon
2. SMTP stands for Simple Mail Transfer Protocol. The smtplib modules is useful for communicating with mail servers to send mail
3. Sending mail is done with python's smtplib using an SMTP server
4. Actual usage varies depending on complexity of the email and settin of the email server, the instruction here are based on sending email through Gmail.

Servo Motor

1. A Servo motor is a combination of DC motor, position control System

and gears. Servos have many applications in the modern world and with that, they are available in different shapes and sizes.
2. A servo motor mainly has three wires, one is for positive voltage, another is for ground and last one is for position setting.
3. A Servo motor is a combination of DC motor, position control system and gears.

Steps :-
• Create the lock / unlock application to control the servo motor lock. change its owner and group as www-data. Location: /var/www/html
• Write the application to read the image and send it as email attachment to the user. Location /home/pi
• Write application using HTML-PHP to control the servo motor lock. Location: /var/www/html

Conclusion :-
Thus, we have developed smart application for smart Home System