INTRODUCTION TO RASPBERRY PI BOARD

- The Raspberry Pi is a tiny and affordable computer that you can use to learn programming through fun, practical project.
- Raspberry Pi is the name of a series of a single-board computers made by Raspberry Pi Fourdation.
- Raspberry Pi is used to learn programming skills, build hardware projects, do home automation, Edge computing
- Several generations of Raspberry Pi have been released All model of Raspberry Pi is feature a Broadcom system o a chip (SoC) with an integrated ARM compatible central processing unit (CPU) and on-chip GPU.
- Processor speeds ranges from 700MHz to 1.2GHz for RPi 3 and on-board memory range from 256MB to 1GB RAM.
- SD cards are used to store OS and program memory in ~~and~~ either SDHC or MicroSDHC sizes.
- Depending on model, the boards have either a single USB port or upto four USB ports.
- For video output, HDMI and composite video are supported.

- Lower level output is provided by a number of GPIO pins which support common protocols like I²C.

- The Raspberry Pi Pin configuration is as follows:

Alternate function					Alternate function
	3.3V PWR	1	2	5V PWR	
I2C1 SDA	GPIO 2	3	4	5V PWR	
I2C1 SCL	GPIO 3	5	6	GND	
	GPIO 4	7	8	UART0 TX	
	GND	9	10	UART0 RX	
	GPIO 17	11	12	GPIO 18	
	GPIO 27	13	14	GND	
	GPIO 22	15	16	GPIO 23	
	3.3V PWR	17	18	GPIO 24	
SPI0 MOSI	GPIO 10	19	20	GND	
SPI0 MISO	GPIO 9	21	22	GPIO 25	
SPI0 SCLK	GPIO 11	23	24	GPIO 8	SPI0 CS0
	GND	25	26	GPIO 7	SPI0 CS1
	Reserved	27	28	Reserved	
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
SPI1 MISO	GPIO 19	35	36	GPIO 16	SPI1 CS0
	GPIO 26	37	38	GPIO 20	SPI1 MOSI
	GND	39	40	GPIO 21	SPI1 SCLK

fig: Raspberry Pi Pin Configuration

Experiment - 1

PROCEDURE TO LOAD OS ON RASPBERRY PI

Aim: To install / load operating system on Raspberry Pi.

Apparatus: 1) Raspberry Pi board
2) SD card
3) SD card reader
4) PC

Procedure:

1. Download the Raspberry Pi operating system
. The recommended OS is called Raspbian.

2. Unzip the file that you just downloaded.
a) Right click on the file and choose "Extract all".
b) Follow the instructions – you will end up with a file ending in .img.

This .img file can only be written to your SD card by special disk imaging software, so...

3) Download the Win32DiskImager software
a. Unzip it in the same way you did the Raspbian .zip file.
b. You now have a new folder called win32diskimager-binary

You are now ready to write the Raspbian image to your SD card.

4. Writing Raspbian to the SD card
a) Plug your SD card into your PC.
b) Run the file named Win32DiskImager.exe
c) If the SD card you are using isn't found automatically then click on the drop down box and select it.

d) In the Image File box, choose the Raspbian .img file that you downloaded.

e) Click Write

f) After a few minutes you will have an SD card that you can use in your Raspberry Pi.

5) Booting your Raspberry Pi for the first time

a) Follow the Quick start guide on page 1

b) On first boot you will come to the Raspi-config window

c) Change settings such as timezone and locale if you want

d) Finally, select the second choice:

expand_rootfs

and say 'yes' to a reboot

e) The Raspberry Pi will reboot and you will see

raspberrypi login:

f) Type:

Pi

g) You will be asked for your Password

h) Type:

raspberry

i) You will then see the prompt:

Pi@raspberry ~ $

j) Start the desktop by typing:

startx

k) You will find yourself in a familiar-but-different desktop environment.

l) Experiment, explore and verify.

Result:

Installed Raspberry Pi OS for Raspberry Pi board using SD card.

Experiment-2

BASIC ARITHMETIC OPERATIONS

Aim:- To perform basic arithmetic operations using Raspberry Pi

Apparatus: (1) PC
(2) RaspberryPi board
(3) Power supply cable

Procedure:

- Using the python programming language for writing the code for arithmetic operations is simple and easy.
- Open the RaspberryPi software and write code in thony.

Boot Pi → Go to main menu → Programming → Select Thonny.

Program:

```
num1 = int(input('Enter First number:'))
num2 = int(input('Enter second number:'))

add = num1 + num2

dif = num1 - num2

mul = num1 * num2

div = num1 / num2

floor_div = num1 // num2

Power = num1 ** num2

modulus = num1 % num2.

Print ('Sum of', num1, 'and', num2, 'is:', add)

Print ('Diff of', num1, 'and', num2, 'is:', dif)
```

```
print ('Product of', num1, 'and', num2, 'is:', mul)
print ('Division of', num1, 'and', num2, 'is:', Div)
print ('Floor Division of', num1, 'and', num2, 'is:', floor_div)
print ('Exponent of', num1, 'and', num2, 'is:', power)
print ('Modulus of', num1, 'and', num2, 'is:', modulus)
```

Output:

```
Enter First number: 4
Enter Second number: 2
Sum of 4 and 2 is: 6
Diff of 4 and 2 is: 2
Product of 4 and 2 is: 8
Division of 4 and 2 is: 2
Floor Division of 4 and 2 is: 2
Exponent of 4 and 2 is: 16
Modulus of 4 and 2 is: 0
(Remainder)
```

Result:

Verified the authmetic operators output with theoretical values.

Experiment-3

BLINKING OF LED

Aim: To perform blinking of LED operation using Raspberry Pi board and to verify practically

Apparatus: 1) PC or desktop
2) Raspberry Pi board
3) LED (pin 7)
4) Jumping wires.

Procedure:

- Open the Raspberry Pi software and follow the below steps.

  Boot Pi → Go to main menue → Programming → select Thonny.

- Write the program for blinking of Led in thonny.

Program:

```
import RPi.GPIO as G
import time

G.setmode(G.BOARD)
G.setup(7, G.OUT)
while True:
    G.output(7, True)
    print("Led is On")
    time.sleep(1)
    G.output(7, False)
    print("Led is Off")
    time.sleep(1)
```

Result:

Blinking of LED operation is performed and observed practically.

Experiment-4

CONTROLLING LED WITH PUSH BUTTON

Aim: To perform controlling LED with push button operation using Raspberry Pi board and to verify the output.

Apparatus: (1) PC or desktop
(2) Push button
(3) LED
(4) Jumping wires
(5) Raspberry Pi board

Procedure:

- Open the Raspberry Pi software and follow the below ste

Boot Pi → Go to main menu → Programming → Select Thon

- Connect the LED and Push button to the Raspberry Pi board according the circuit requirements.

Program:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(24, GPIO.OUT)
try:
    while True:
        button_state = GPIO.input(23)
```

```
if button_state == False:
    GPIO.output(24) True)
    print('Button Pressed..')
    time.sleep(0.2)
else:
    GPIO.output(24, False)
except:
    GPIO.cleanup()
```

Output:

Button Pressed....

Result:

Verified the output for controlling LED with Push Button.

Experiment-5

DHT11 SENSOR TEMPERATURE AND HUMIDITY

Aim: To measure the temperature and humidity using DHT11 sensor.

Apparatus: 1) DHT11 Sensor

2) PC / desktop

3) Jumping wires

4) Raspberry Pi board.

Procedure:

To install DHT11 sensor library

1. To download module from Git, Install Git

"sudo apt-get install git-core"

2. Download Adafruit_DHT library

"git clone https://github.com/adafruit/Adafruit_Python_DH

3. Navigate to Adafruit_Python_DHT directory

"cd Adafruit_Python_DHT"

4. For Python3

"sudo apt-get install build-essential python3-de

5. Install DHT Module

For Python3

"sudo python3 setup.py install"

Program:

```
# Import Sensor Module
import Adafruit_DHT
try:
    while True:
        humidity, temperature = Adafruit_DHT.read_retry(11,3)
        print("Humidity={}%; Temperature={}C".
              format(humidity, temperature))
except KeyboardInterrupt:
    print("Aborted by user")
```

Result:

Observed different results ~~for~~ of temperature and humidity based on the changing room conditions using DHT11 sensor.

Experiment - 6

DHT11 SENSOR WITH THINGSPEAK

Aim: To measure the temperature and humidity values using DHT11 sensor with thingspeak cloud platform.

Apparatus: (1) PC or desktop

(2) thingspeak IOT cloud platform

(3) DHT11 sensor

(4) Jumping wires

(5) Raspberry Pi board

Procedure:

Thingspeak setup for Pi weather station.

Step 1: Thingspeak Account setup

- For creating your channel on Thingspeak you first need to sign up on Thingspeak.

Step 2: Create a channel for your data

- Once you sign in, create a new channel by clicking "New Cha button.

Step 3: API key

- Click on "API keys" button to get your unique API key for uploading your sensor data.
- Copy your "API key".

• Install following in the terminal

sudo pip install thingspeak

sudo pip3 install thingspeak.

Program:

```
import thingspeak
import time
import Adafruit_DHT

channel_id = xxxxxx
write_key = 'xxxxx'
Pin = 3
sensor = 11
def measure (channel):
 try:
    humidity, temperature = Adafruit_DHT.read_retry
                                (sensor, pin)
    response = channel.update({'field1': temperature,
                               'field2': humidity})
    print ("Humidity = {} %; Temperature = {} C". format
                               (humidity, temperature)
 except:
    print ("connection failed")
if __name__ == "__main__":
    channel = thingspeak.Channel (id = channel_id, api_key
                                  = write_key)
    while True:
        measure(channel)
        time.sleep(15)
```

Result:

The room temperature and humidity values are observed using the DHT11 sensor with thingspeak Platform.

Experiment-7

CAMERA INTERFACING

Aim: To perform camera interfacing using Raspberry Pi board.

Apparatus: (1) PC or desktop
(2) Raspberry Pi board
(3) Camera Module
(4) Power cable.

Procedure:

- Connect the camera module to the Raspberry Pi board.
- Enabling the camera
  - Open Raspberry Pi configuration tool from the main menu
  - Open preferences
  - Raspberry Pi configuration
  - Interfaces ⇒ Camera Enabled
    SSH Enabled.
- Taking a picture in terminal.
  1. raspistill -o pic.jpg (image captured after 5 seconds)
- Recording a video with Raspberry pi camera module
  1. raspivid -o video.h264

Python code to capture the image

To install library for the camera module

sudo apt-get install python-picamera

Program:

```
from time import sleep
from picamera import PiCamera

camera = PiCamera()
camera.resolution = (1024, 768)
camera.start_preview()
sleep(2)
camera.capture('picture1.jpg')

#Take a picture including the annotation
camera.annotate_text = 'Hello world!'
camera.start_preview()
sleep()
camera.capture('picture2.jpg')

# Capture and crop the picture
camera.start_preview()
sleep(2)
camera.capture('picture3.jpg', resize=(320, 240))
camera.stop_preview()

# Recording Video using Pi camera
camera.resolution = (640, 480)
camera.start_recording('my_video.h264')
camera.wait_recording(30)
camera.stop_recording()
```

Result:

Observed the camera interfacing and captured images and recoded video using the Pi camera.

Experiment-8

MOTION DETECTION PIR USING LED AND CAMERA

Aim: To perform motion detection PIR using LED and camera

Apparatus: (1) PC or desktop
(2) PIR sensor
(3) LED
(4) Jumping wires
(5) Raspberry Pi board
(6) Camera Module.

Procedure:

- Connect the camera module and Led to the raspberryPi board.
- Write the program code in thonny.

Program:

```
import RPi.GPIO as GPIO
import time
from picamera import PiCamera
GPIO.setmode(GPIO.BOARD)
pir = 8
led = 10
GPIO.setup (pir, GPIO.IN)
GPIO.setup (led, GPIO.OUT)
print ("Sensor initializing.....")
time.sleep(2)
print ("Active")
```

```
print ("Press Ctrl+C to end program")
camera = PiCamera()
camera.resolution = (1024, 768)

try:
    while True:
        if GPIO.input(pir) == True:
            print ("Motion Detected!")
            GPIO.output(led, True)
            camera.start_preview()
            camera.annotate_text = 'Intruder!'
            time.sleep(2)
            camera.capture('foo1.jpg', resize=(320,240))
            camera.start_preview()
            time.sleep(4)
            GPIO.output(led, False)
            time.sleep(2)
    except KeyboardInterrupt:
        pass
    finally:
        GPIO.output(led, False)
        GPIO.cleanup()
        print ("Program ended")
```

Output:

```
Sensor initializing.....
Active
Press Ctrl+C to end program
Motion Detected
Program Ended
```

Result:

The functionality of PIR sensor along with camera module is observed practically for intruder detection and verified.

Experiment-9

NODERED

Aim: To write a program that sends sensor data to Cloud using NodeRed service to perform Data Analytics using Rpi3.

Apparatus: (1) PC or Monitor

(2) Raspberry Pi board.

Procedure:

- Start up your Raspberry Pi.
- Click on the Raspberry Pi icon, then the programming menu to open Node-RED.
- You should see a window displaying information about Node-RED starting up.
- Now go to the internet menu and open chromium web browser.
- In chromium, locate the address bar at the top and type in localhost:1880, then press enter. This will display the Node-RED interface.
- Programs in Node-RED are called flows
- The coloured blocks on the left side are nodes.
- Scroll down to bottom to see nodes labelled Raspberry Pi.
- You will see two nodes with label "rpi gpio":
  - → Raspberry icon on left is used for inputs
  - → Raspberry icon on right is used for outputs (Eg: LED)

- Drag an output node onto the blank page in middle
- Double-click on the node, box appears to let you configure the node.
- After configuration is done. Click Done.

Result :

Performed data analytics using NodeRed service with Rpi 3 board is verified.