Practical No :- 1

Introduction to Raspberry Pi

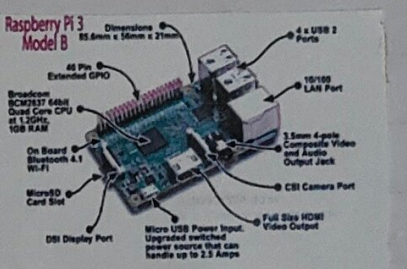
Definition :- The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little computer which can be used in electronic projects, and for many of the things that our desktop PC does, like spreadsheets, word-processing, browsing the internet, and playing games. It also plays high-definition video.

- Raspbian is the official operating system for all models of the Raspberry Pi.
- Raspberry pi is slower than a modern laptop or desktop, but is still a complete linux computer and can provides all expected functionalities at a low-power consumption level.
- The default username is 'pi' and password 'raspberry'
- the processor used in raspberry pi is Cortex-A53 (ARM v8) 64 bit SoC architecture. Several generations of Raspberry pi's have been released. All models feature a Broadcom system on chip with an integrated ARM-compatible central processing unit (CPU) & on-chip graphics processing unit (GPU).

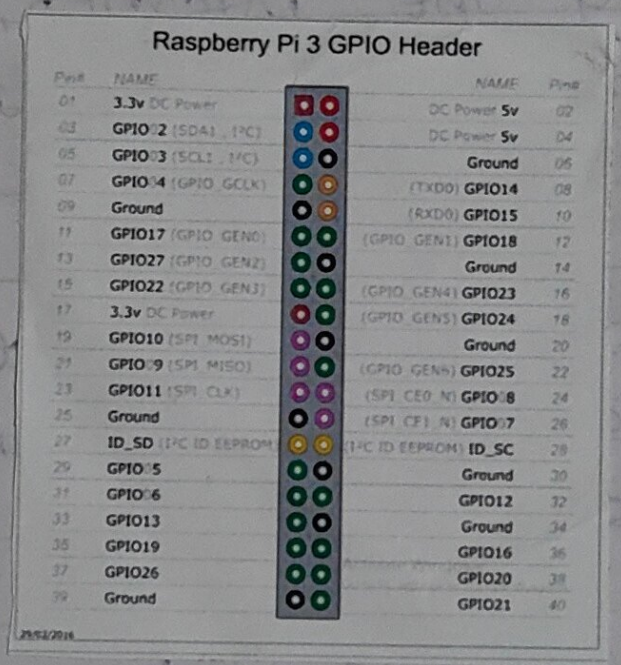
GPIO Header

- Pin number 1 & 17 : It provide 3.3v current to the Raspberry Pi.

	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	$35	$5	$35	$35



- Pin Number 2 & 4 : The 5v power pins are connected directly to the pi's power input.
- Pin Number 6, 9, 14, 20, 25, 30, 34, & 39 : The Ground pins on the Raspberry pi and all ~~ecto~~ electrically Connected so it doesn't matter which one we use if we're wiring up a voltage supply.
- Pin Number 3, 5, 27, & 28 : I2C interface pins that allow us to connect hardware modules with just two control pin. GPIO pins 0 and 1 are present on the board (physical pins 27 and 28) but are reserved for advanced use.
- Pin Number 19, 21, 23, 24 & 26 : SPI interface with SPI devices, a similar concept to I2C but uses a different standard.
- Pin Number 8 & 10 : It is serial Rx & Tx pins for communication with serial peripherals.
- Pin Number 7 : General purpose Clock pins can be set up to output a fixed frequency without any ongoing software control.
- Pin Number 11, 12, 13, 15, 16, 18, 22, 29, 31, 32, 35, 36, 37, 38 & 40 :- True GPIO (General purpose Input/output) pins that we can use to connected LED'S.


Raspberry Pi 3 GPIO Header
Pin# NAME
01 3.3v DC Power
03 GPIO02 (SDA1 , I²C)
05 GPIO03 (SCL1 , I²C)
07 GPIO04 (GPIO_GCLK)
09 Ground
11 GPIO17 (GPIO_GEN0)
13 GPIO27 (GPIO_GEN2)
15 GPIO22 (GPIO_GEN3)
17 3.3v DC Power
19 GPIO10 (SPI_MOSI)
21 GPIO09 (SPI_MISO)
23 GPIO11 (SPI_CLK)
25 Ground
27 ID_SD (I²C ID EEPROM)
29 GPIO05
31 GPIO06
33 GPIO13
35 GPIO19
37 GPIO26
39 Ground
NAME Pin#
DC Power 5v 02
DC Power 5v 04
Ground 06
(TXD0) GPIO14 08
(RXD0) GPIO15 10
(GPIO_GEN1) GPIO18 12
Ground 14
(GPIO_GEN4) GPIO23 16
(GPIO_GEN5) GPIO24 18
Ground 20
(GPIO_GEN6) GPIO25 22
(SPI_CE0_N) GPIO08 24
(SPI_CE1_N) GPIO07 26
(I²C ID EEPROM) ID_SC 28
Ground 30
GPIO12 32
Ground 34
GPIO16 36
GPIO20 38
GPIO21 40
29/02/2016

Practical No:0

Aim:- Starting Raspbian Os, familiarizing with Raspberry Pi Components & interface, Connecting to ethernet, Monitor, USB.

- Go to the official website of Raspberry pi which is https://www.raspberrypi.org/
- Click on the option named as "Software" -> "See all download options".
- The Raspbian with Desktop image contained in the zip archive is over 4GB in size.

Note: There are 3 .zip file.

1. Raspberry Pi Os with desktop and recommeded Software.
2. Raspberry Pi Os with desktop.
3. Raspberry Pi Os Lite.

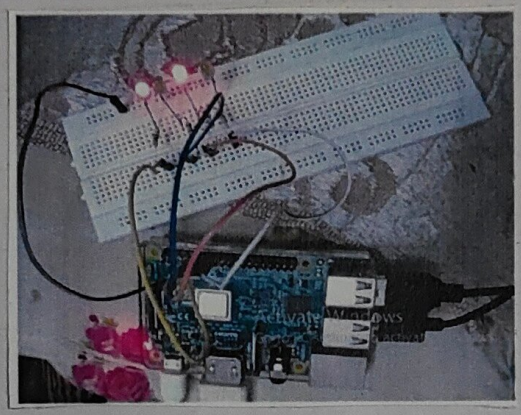
- We will use the 2nd option i.e. Raspberry Pi os with desktop.
- After downloading the above image, click the "installation guide".
- Official images for recommended operating system are available to download from the Raspberry Pi website. Alternate distributions are available from third-party vendors. We will need to use an image writing tool to install the image we have downloaded on our SD card.
- Etcher is a graphical SD card writing tool that work on MAC Os, Linux & windows. It is the easiest option for most users.
- Download Etcher & install it. http://www.balena.io/etcher/.

- Connect an SD card reader with the SD Card inside.
- Open Etcher & select from our hard drive the Raspberry pi. img or .zip file we want to write to the SD card
- Select the SD card we wish to write our image to
- Review our selections & click 'Flash' to begin writing data on the SD card.

Practical No 1

Aim:- Displaying different LED patterns with Raspberry

- Components Required:- Raspberry pi Model B+, 5V Micro USB power supply, VGA to HDMI converters, Monitors, USB keyboard & Mouse, Resistors of 220 Ω, Breadboard, jumper wires (male to female) to connect breadboard & Raspberry Pi.
- Theory :- Connect our Raspberry pi with the monitor using the VGA to HDMI converter. Connect a 5V micro USB power supply to give power to the Raspberry pi. Connect the keyboard & mouse to the pi. Open the python shell after booting the system. Now, write a python code for blinking of LED's Save the code. Connect the anode of the LED with the 220Ω resistor & Connect the resistor to 3.3 power i.e. pin number 1 of GPIO header. Once the connections are done, run the code to See the blinking of the LED's.
- Connectors:
  1. Connect VGA to HDMI Convertor, with Raspberry Pi & Monitor.
  2. Connect the USB keyboard and mouse with Raspberry pi.
  3. Connect the ~~micro~~ micro USB 5V power supply to Raspberry pi.
  4. Connect Raspberry pi to breadboard using jumper wires.
  5. Connect the cathode of LED to the GPIO pins.
  6. Connect anode of the LED with resistors.



- Code

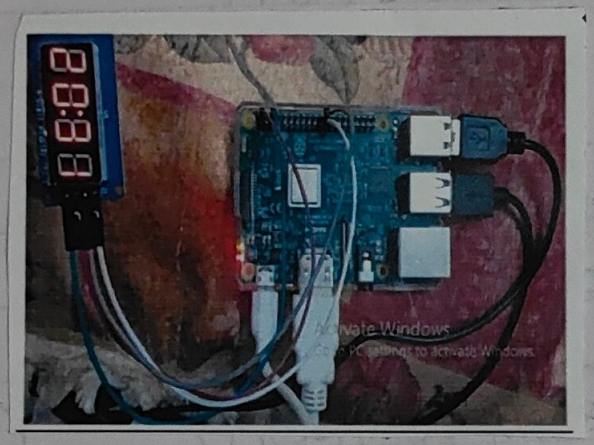
```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(17, GPIO.OUT)
GPIO.setup(18, GPIO.OUT)
GPIO.setup(27, GPIO.OUT)
GPIO.setup(22, GPIO.OUT)

while 1:
    time.sleep(.4)
    print("LED ON")
    GPIO.output(17, GPIO.HIGH)
    GPIO.output(18, GPIO.HIGH)
    GPIO.output(27, GPIO.HIGH)
    GPIO.output(22, GPIO.HIGH)

    time.Sleep(.4)
    Print("LED ON")
    GPIO.output(17, GPIO.HIGH)
    GPIO.output(18, GPIO.LOW)
    GPIO.output(27, GPIO.HIGH)
    GPIO.output(22, GPIO.LOW)

    time.sleep(.4)
    print("LED OFF")
    GPIO.output(17, GPIO.low)
    GPIO.output(18, GPIO.low)
    GPIO.output(27, GPIO.low)
    GPIO.output(22, GPIO.low)

    time.sleep(.4)
    print("LED OFF")
    GPIO:output(17, GPIO.low)
    GPIO.output(18, GPIO.HIGH)
    GPIO.output(27, GPIO.low)
    GPIO.output(22, GPIO.HIGH)
```

Practical No2

Aim:- Displaying Time over 4-Digit 7-Segment Display using Raspberry Pi

- Components Required:- Raspberry Pi Model B+, 5V micro USB power supply, VGA to HDMI Converters, Monitors, USB Keyboard & Mouse, tm1637 4digit 7 segment display board (Common anode) which contains tm1637 driver chip, jumper wires (female to female) to Connect tm 1637 & Raspberry Pi.
- Theory :- To display small amount of data with Raspberry Pi, we can use 4digit 7-segment Display. 7 segment Display has seven segment in it and each segment has one LED inside it to display the numbers by lighting up the corresponding segments.
- Connections:-

1. Connect VGA to HDMI converter, with Raspberry Pi & Monitor.
2. Connect the USB keyboard and mouse with Raspberry pi.
3. Connect the micro USB 5v power supply to Raspberry Pi.
4. Connect tm 1637 to Raspberry Pi using jumper wires.
   - CLK to 23rd pin (GPIO 11)
   - DIO to 24rd pin (GPIO 08)
   - VCC to 2nd pin
   - GND to 6th pin.


Activate Windows

Code

```
import sys
import time
import datetime
import RPi.GPIO as GPIO
import tm1637
Display = tm1637.TM1637(11,8,tm1637.BRIGHT_TYPICAL)
Display.clear()
Display.setBrightness(1)

while(True):
    now = datetime.datetime.now()
    hour = now.hour
    minute = now.minute
    second = now.second
    currenttime = [int(hour/10), hour%10, int(minute
                  /10), minute%10]
    Display.show(currenttime)
    Display.showDoublepoint(second%2)
    time.sleep(1)
```

```
> cd Desktop
> md 4Digits
> cd 4Digits
> wget https://raspberry-tips.net/files/
        tm1637.py
CLK - Clock - (23rd pin (GPIO 11))
DIO - Data I/O - (24 pin (GPIO 08))
```

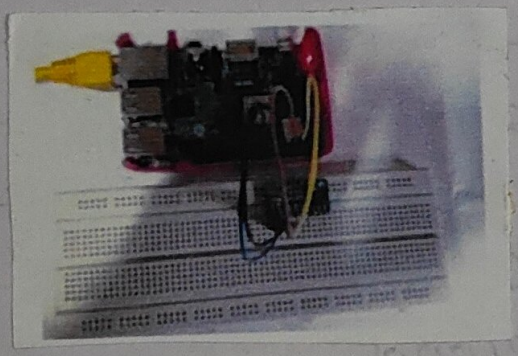
VCC - Voltage Common Collector - 2 pin
GND - Ground - 6 pin

practical No 3

Aim:- Raspberry pi based oscilloscope.

- Components Required:- Raspberry pi Model B+, 5V micro USB power supply, VGA to HDMI Converters, Monitors, USB keyboard & mouse, ADS1115 ADC, LDR (optional as it's meant for test), 10K to 1K resistor, jumper wires Breadboard.
- Theory :- ADS1115 ADC Chip is used to Convert the analog input signals to digital signals which can be visualized with the Raspberry pi. This chip is important because the Raspberry pi does not have an on-based analog to digital Converter (ADC).
- Connections:-

1. Connect VGA to HDMI Converter, with Raspberry Pi & monitor.
2. Connect the USB keyboard and mouse with Raspberry Pi.
3. Connect the micro USB 5v power Supply to Raspberry pi.
4. Connect ADC to Raspberry pi using jumper wires.
   - VDD to 17th pin (3.3V)
   - GND to 9th pin (GND)
   - SCL to 5th pin (GPIO 03)
   - SDA to 3rd pin (GPIO 02)



```
import time
import numpy.core.multiarray
from drawnow import *
import Adafruit_ADS1x15
adc = Adafruit_ADS1x15.ADS1115()
GAIN = 1
val = []
cnt = 0
plt.ion()
adc.start_adc(0, gain=GAIN)
print('Reading ADS1x15 channel 0')
def makeFig():
    plt.ylim(-5000,5000)
    plt.title('Oscilloscope')
    plt.grid(True)
    plt.ylabel('ADC outputs')
    plt.plot(val, 'ro-', label='lux')
    plt.legend(loc='lower right')
while(True):
    value = adc.get_last_result()
    print('Channel 0: {0}'.format(value))
    time.sleep(0.5)
    val.append(int(value))
    drawnow(makeFig)
    plt.pause(.00001)
    cnt = cnt+1
    if(cnt>50):
        val.pop(0)
```

```
> Sudo apt-get update
> sudo apt-get upgrade
> mkdir practical2
> Cd practical2
> Sudo apt-get install build-essential python-
    dev python-smbus git
> git clone https://github.com/adafruit/Adafruit-
    python-ADS1x15.git
> Cd Adafruit-Python-ADS1x15
> Sudo Python setup.py install.
> Sudo. apt-get install python-matplotlib
> pip install numpy == 1.19.3
> Sudo pip install draconolo.
```

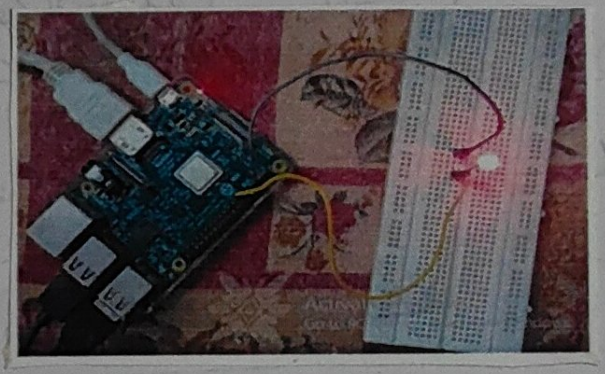
Practical No.04

Aim:- Controlling Rasberry Pi with Whatsapp (Telegram)

- Components Requised: Rasberry pi Model B+, 5V micro USB power supply, VGA to HDMI converters, monitors, USB keyboard & Mouse, LED light, Resistors of 220 Ω, Breadboard, jumper wires (male to female) to connect breadboard & Raspberry Pi.
- Theory : The program will primarily check for two words, they are on and off. once detecting either one of these two words, it will look for other keywords like white, yellow, green and red. The respective colour LED will be toggled only if the word is detected. We will also update a string for the detected words to send a message back to telegram bot.
- Connectors:

1. Connect VGA to HDMI converter, with Raspberry Pi & monitor.
2. Connect the USB keyboard and mouse with Raspberry Pi.
3. Connect the micro USB 5v power supply to Raspberry Pi.
4. 1 led
5. 1 resistor of 220 Ω (optional).
6. 2 jumper wires
7. 1 mobile Phone (smart phone) or System.

o Code



```
import sys
import time
import random
import datetime
import telepot
import RPi.GPIO as GPIO
GPIO.setwarnings(False)

#LED
def On(pin):
  GPIO.output(pin,GPIO.HIGH)
  return
def off(pin):
   GPIO.output(pin, GPIO.LOW)
   return
   GPIO:setmode(GPIO.BOARD)
   GPIO.setup(11, GPIO.OUT)
def handle(msg):
   chat_id = msg['chat']['id']
   command = msg['text']
   print('Got command: %s' % command)
   if command == 'On':
     bot.sendMessage(chat_id, on(11))
   elif command == 'off':
     bot.sendMessage(chat_id, off(11))
   bot = telepot.Bot('Bot Token')
   bot.message_loop(handle)
   print('I am listening...')
while 1:
     time.sleep(10)
```

Controlling Pi using Telegram.

> Sudo pip install telepot

OR

> Sudo pip3 install telepot

Telegram.

BotFather.

Practical No 6.

practical 6. Fingerprint Sensor interfacing with Raspberry pi

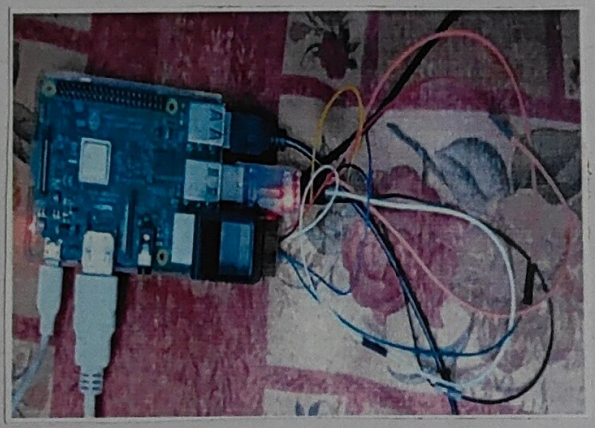
- Components Required: Raspberry pi Model B+, 5V Micro USB power Supply, VGA to HDMI Converters, Monitors, USB keyboard & Mouse, Jumper wires (female to female) USB to TTL Converter, Raspberry pi Fingerprint sensor.

→ Connectors:

1. Connect VGA to HDMI convertor, with Raspberry pi & monitor.
2. Connect the USB keyboard and mouse with Raspberry pi.
3. Connect the micro USB 5v Power Supply to Raspberry pi.
4. Jumper wires to Connect GPS module with Raspberry pi
5. USB to TTL Connection:
   - Black → GND to GND of TTL
   - Red → +3.3V to VCC of TTL
   - yellow → Tx to Rx of TTL
   - White → Rx to Tx of TTL

o Code

```
import time
form pyfingerprint.pyfingerprint import PyFingerprint
import RPi.GPIO as gpio.
print("Successfully imported fingerprint module")
gpio.setwarnings(False)
```



```
gpio.setmode(gpio.BCM)
f = PyFingerprint('/dev/ttyUSB0', 57600, 0xFFFFFFFE, 0x00000000)
def enrollFinger():
    print('Enrolling Finger')
    time.sleep(3)
    print('Place Finger')
    while(f.readImage()==False):
        pass
    f.convertImage(0x01)
    result = f.searchTemplate()
    positionNumber = result[0]
    if(positionNumber >=0):
        print('Template already exists at position#'
            +str(positionNumber))
        time.sleep(2)
        return
        print('Remove finger')
        print('waiting for same finger')
        time.sleep(3)
    while(f.readImage()==False):
        pass
    f.convertImage(0x02)
if(f.compareCharacteristics()==0):
    print('Fingers do not Match')
    time.sleep(2)
    return
else:
    f.createTemplate()
    positionNumber = f.storeTemplate()
```

```
        print('Finger enrolled successfully')
        print('Stored at pos:' + str(positionNumber))
            time.sleep(2)
def searchFinger():
    try:
        print('Waiting for finger...')
        time.sleep(2)
        while(f.readImage() == False):
            time.sleep(4)
            return
        f.convertImage(0x01)
        result = f.searchTemplate()
        positionNumber = result[0]
        if positionNumber == -1:
            print('No Match found')
            time.sleep(4)
            return False
        else:
            print('Found template at position' + str
                (positionNumber))
            time.sleep(4)
            return True
        except Exception as e:
            print('Operation failed')
            print('Exception message:' + str(e))
            exit(1)
    return
def deleteFinger():
    try:
```

```
print('waiting for finger...')
time.sleep(2)
while(f.readImage()==False):
    time.sleep(4)
    return
f.convertImage(0x01)
result = f.searchTemplate()
positionNumber = result[0]
if positionNumber==-1:
    print('No Match found')
    time.sleep(4)
    return ~~True~~ False
else:
  if(f.deleteTemplate(positionNumber)=True):
      print('Finger Deleted')
      time.sleep(4)
      return True.
  except Exception as e:
      print('Operation failed')
      print('Exception message:' + str(e)) exit(1)
  return
time.sleep(1)
print('start')
while True:
  i = int(input("\nEnter:\n1. Enroll\n2. Search\n3.
      Delete\n4. Exit\n"));
if i==1:
    enrollFinger()
elif i==2:
```

```
            searchFinger()
    elif ch==3:
            deleteFinger()
    elif ch==4:
            break
    else:
        print("Invalid option !!! Enter correct")
```

Practical No:7

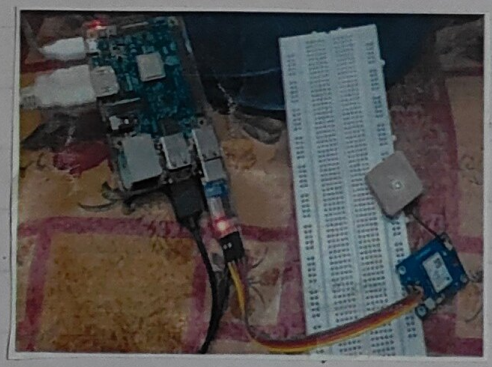
Aim:- Raspberry Pi GPS Module Interfacing.

- Components Required:- Raspberry Pi Model B+, 5V Micro USB power supply, VGA to HDMI Converters, Monitors, USB keyboard & Mouse, jumper wires (female to female), GPS module with antenna.
- Theory: Raspberry Pi, interfaced with a GPS module, can be used for developing an advanced real-time navigation system. Incorporating the pi's image processing, audio processing and web interface capabilities along with the GPS data we can develop advanced navigation schemes for real-time implementation.
- Connections:

1. Connect VGA to HDMI converter, with Raspberry pi & monitor.
2. Connect the USB keyboard and mouse with Raspberry pi.
3. Connect the micro USB 5V power supply to Raspberry pi.
4. Jumper wires to connect GPS module with Raspberry pi
   - VCC to pin 2
   - Rx to pin 8 (Tx)
   - Tx to pin 10 (Rx)
   - Gnd to pin 6

o Code

```
import time
```



```
import serial
import string
import pynmea2
import RPi.GPIO as gpio
gpio.setmode(gpio.BCM)
Port = "/dev/ttyAMA0"
ser = serial.Serial(Port, baudrate = 9600, timeout = 0.5)
while 1:
    try:
        data = ser.readline()
    except:
        print("loading")
if data[0:6] == '$GPGGA':
    msg = pynmea2.parse(data)
    print(msg)
    time.sleep(2)
```

practical No. 9

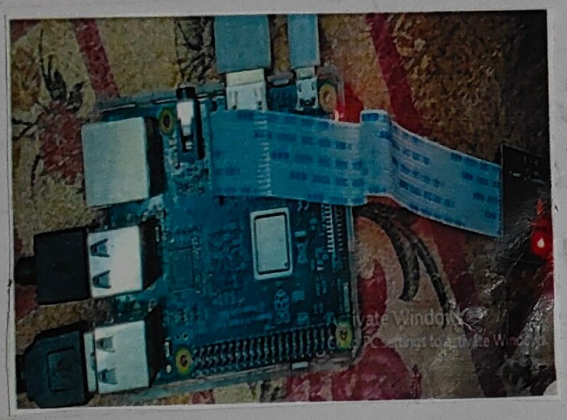
Aim: Visitor Monitoring with Raspberry Pi & Pi Camera

- Components Required: Raspberry Pi Model B+, 5V Micro USB Power Supply, VGA to HDMI Converters, Monitors, USB Keyboard & Mouse, Pi Camera Module.
- Theory: Here we are interfacing Pi Camera with Raspberry Pi to capture the image of every visitor which has entered through the Gate or door. In this Project, whenever any person is arrived at the Gate, he has to press a button to open the Gate, and as soon as he/she press the button, his/her picture will be captured and saved in the system with the Date and time of the entry.
- Connections:

1. Connects VGA to HDMI converts with Raspberry Pi & Monitor.
2. Connects the USB keyboard and mouse with Raspberry Pi.
3. Connect the micro USB 5v power supply to Raspberry Pi.
4. Pi Camera Module.
   - Open the Raspberry Pi Camera notch
   - Insert Camera module in the notch.
   - Close the notch

- Code

```
from picamera import PiCamera
from time import sleep
```



```
Camera = PiCamera()
Camera.resolution = (1280, 780)

Camera.start_preview()
Camera.start_recording('/home/pi/Desktop/camera/video
                       h264')

for i in range(5):
    sleep(5)
        Camera.capture('/home/pi/Desktop/camera/images
                       %s.jpg'%i)

Camera.stop_recording()
Camera.stop_preview()
```

> Sudo apt-get update.
> sudo apt-get upgrade.
> cd Desktop
> mkdir camera.
> cd camera.
> Sudo pip install picamera.
> sudo apt-get install python-picamera. raspistill
  -o image.jpg.

login = pi
password = Raspberry.

Practical No 10

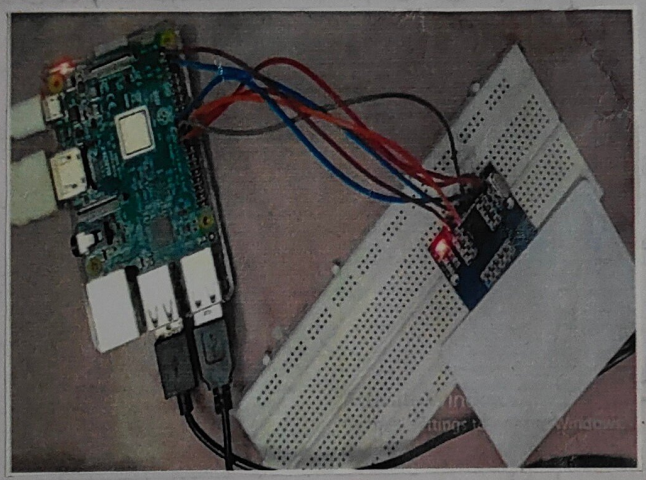
Aim:- Interfacing Raspberry Pi with RFID

- Components Required:- Raspberry Pi Model B+, 5V Micro USB power Supply, VGA to HDMI Converters, Monitors, USB Keyboard & Mouse, RFID Reader (RC 522), RFID Tags or Cards, jumper wires (female to female) to connect RFID RC522 & Raspberry Pi.
- Theory:- The RFID RC522 is a very low-cost RFID (Radio-frequency identification) reader and writer that is based on the MFRC522 microcontroller. This microcontroller provides its data through the SPI protocol and works by creating a 13.56 MHz electromagnetic field that it uses to communicate with the RFID tags.

:- Connectors:-

1. Connect VGA to HDMI converter, with Raspberry Pi & Monitor.
2. Connect the USB keyboard and mouse with Raspberry Pi.
3. Connect the micro USB 5v power supply to Raspberry Pi.
4. Connect RFID RC522 to Raspberry Pi using jumper wires.

- SDA connects to pin 24.
- SCK connects to pin 23.
- MOSI Connects to pin 19.
- ~~MOSI~~ MISO Connects to: Pin 21.



- GND Connected to pin 6
- RST Connects to pin 22.
- 3.3V Connects to pin 1

- Write Code

```
import RPi.GPIO as GPIO
from mfrc522 import
SimpleMFRC522.

GPIO.setwarnings(False)
reader = SimpleMFRC522()

try:
    text = input('New Data:')
    print('Now place your tag to write')
    reader.write(text)
    print('Written Successfully')
    finally:
        GPIO.cleanup()
```

- Read Code.

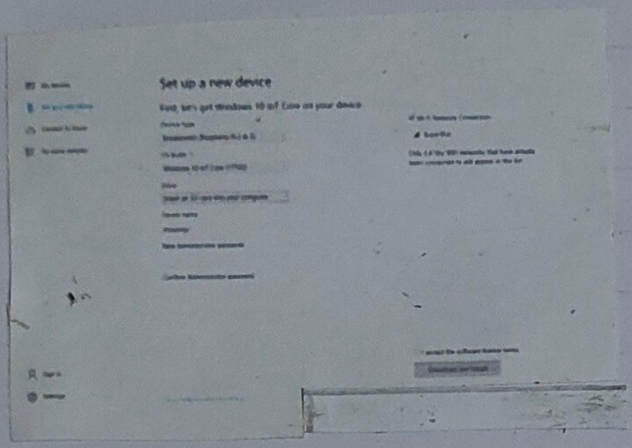
```
import RPi.GPIO as GPIO
from mfrc522 import
SimpleMFRC522

GPIO.setwarnings(False)
reader = SimpleMFRC522()
```

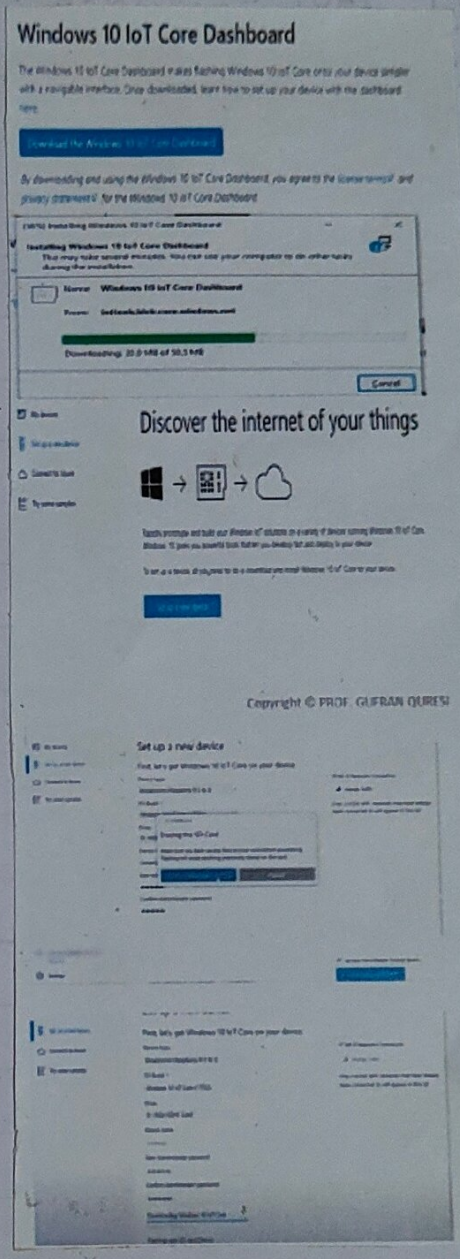
```
try:
    id, text = reader.read()
    print(id)
    print(text)
finally:
    GPIO.cleanup()
```

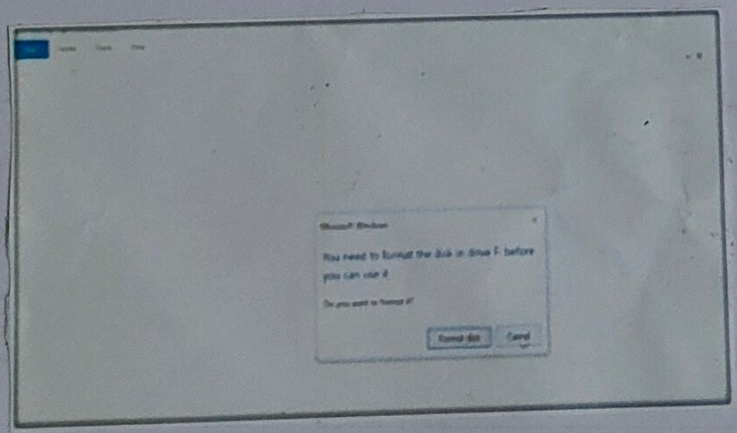
Practical No:- 12

Installing windows 10 IoT Core on Raspberry pi

- Go to the official website of microsoft which is https://docs.microsoft.com/en-us/windows/iot-core/downloads.
Click on the option "Download the windows 10 IoT Core Dashboard".
- After downloading the above image, click the "Install" option.
- After installation. Click on "Run" option.
- IoT Dashboard window will appear. Click on "set up a new device" option.
- A window will appear as shown below. Insert the SD card. & select the drive. Set the root password and check the software license terms checkbox. Finally click on "Download & Install" option.
- A pop-up window will appear to erase the SD Card. Click on "Continue" option to erase existing data if any.
- This will start downloading windows 10 IoT Core. Once downloaded it will start with flashing the SD card Device.
- Lastly a popup menu will appear to format or cancel the SD card. Simply click on "Cancel option.


Set up a new device


Windows 10 IoT Core Dashboard
Discover the internet of your things
Copyright © PROF. GUFRAN QURESI



Set up a new device