Prahladrai Dalmia Lions College Malad (West)

PRESENTS,

Internet Of Things Practical

T.Y.B.Sc.I.T – SEM V

9029120671 / 7021047199

Mumbai University — Course Code: USIT5P2

B. Sc. (Information Technology)		Semester – V	
Course Name: Internet of Things: Theory and Practice Practical		Course Code: USIT5P2	
Periods per week (1 Period is 50 minutes)		3	
Credits		2	
		Hours	Marks
Evaluation System	Practical Examination	2	50
	Internal	--	--

Practical No	Details
0	Starting Raspbian OS, Familiarizing with Raspberry Pi Components and interface, Connecting to ethernet, Monitor, USB.
1	Displaying different LED patterns with Raspberry Pi.
2	Displaying Time over 4-Digit 7-Segment Display using Raspberry Pi
3	Interfacing 16X2 LCD with Raspberry Pi to display different messages.
4	Raspberry Pi Based Oscilloscope
5	Controlling Raspberry Pi with WhatsApp.
6	Fingerprint Sensor interfacing with Raspberry Pi
7	Raspberry Pi GPS Module Interfacing
8	IoT based Web Controlled Home Automation using Raspberry Pi
9	Interfacing Pi Camera with Raspberry Pi
10	Interfacing Raspberry Pi with RFID.
11	Installing Windows 10 IoT Core on Raspberry Pi (Demo Practical)

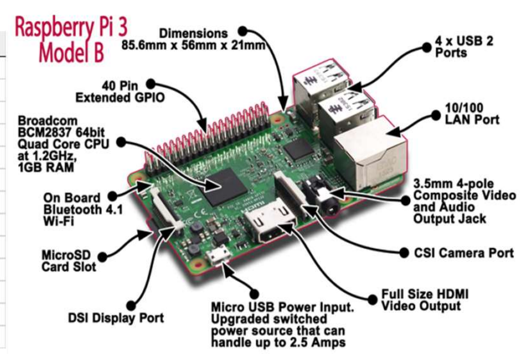
https://www.youtube.com/playlist?list=PLcFZfz96iqW8D6BqZDEZh57jkBYTAcDVg

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 electronics projects, and for may 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 provide 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 Broadcons System on Chip with an integrated ARM-compatible central processing unit (CPU) & on-chip graphics processing unit (GPU).

	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



GPIO Header

Pin Number 1 & 17: It provides 3.3v current to the Raspberry Pi.
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 are all 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 11,12,13,15,16,18,22,29,31,32,33,35,36,37,38 & 40: True GPIO (General Purpose Input Output) pins that we can use to connect LED's.

Raspberry Pi 3 GPIO Header

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

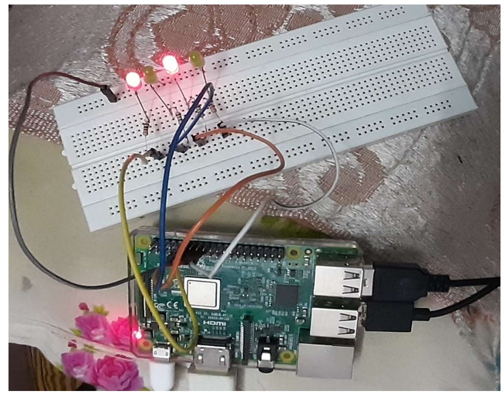
Practical 0: Starting Raspbian OS, Familiarizing with Raspberry Pi Components & interface, Connecting to ethernet, Monitor, USB

- 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 recommended 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 systems 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 https://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 1: Displaying different LED patterns with Raspberry https://www.youtube.com/watch?v=tRvY_3ENsJ4

- **Components Required:** Raspberry Pi Model B+, 5V Micro USB Power Supply, VGA to HDMI Converters, Monitors, USB Keyboard & Mouse, 4 LED lights, 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 convertor. 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 220 Ω 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 USB 5v Power supply to Raspberry Pi.
4. Connect Raspberry Pi to breadboard using jumper wires.
5. Connect the cathode of LED to the Ground pins.
6. Connect anode of the LED with resistors.




```
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 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 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.HIGH)
    GPIO.output(27,GPIO.LOW)
    GPIO.output(22,GPIO.HIGH)
```

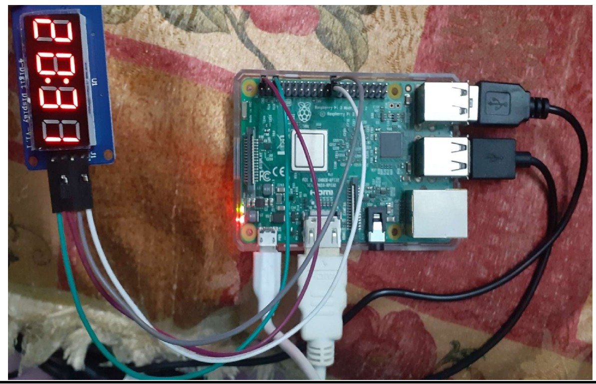
Practical 2: Displaying Time over 4-Digit 7-Segment Display using Raspberry Pi

https://www.youtube.com/watch?v=xt2KMEuuGTc

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

- CLK to 23rd Pin (GPIO 11)
- DIO to 24th Pin (GPIO 08)
- VCC to 2nd Pin
- GND to 6th Pin




```
import sys
import time
import datetime
import RPi.GPIO as GPIO
import tm1637
Display=tm1637.TM1637(11,8,tm1637.BRIGHT_TYPICAL)
Display.Clear()
Display.SetBrightnes(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)
```

Practical 4: Raspberry Pi based Oscilloscope

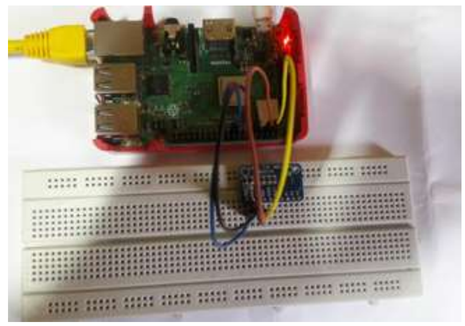
https://www.youtube.com/watch?v=TfzFjh3htKI&t=22s

https://circuitdigest.com/microcontroller-projects/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 its 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-board analog to digital converter (ADC).
- **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. Connect ADC to Raspberry Pi using jumper wires.

- VDD to 17th Pin (3.3 V)
- GND to 9th Pin (GND)
- SCL to 5th Pin (GPIO 03)
- SDA to 3trd Pin (GPIO 02)

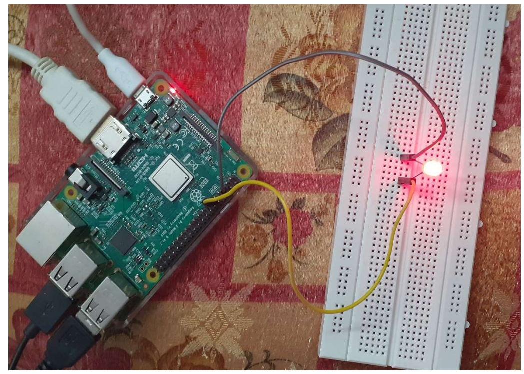



```
import time
import numpy.core.multiarray
from drawnow import *
import Adafruit_ADS1x15
adc = Adafruit_ADS1x15.ADS1115()
GAIN = 1
Val = [ ]
cnt = 0
plt.ion()
adc.start_adc(0, gain=GAIN)
print('Reading ADS1x15 channel 0')
def makeFig():
    plt.ylim(-5000,5000)
    plt.title('Osciloscope')
    plt.grid(True)
    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)
```

Practical 5: Controlling Raspberry Pi with Whatsapp (Telegram)

- **Components Required:** Raspberry 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 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. 1 led
5. 1 resistor of 220 Ω (Optional).
6. 2 jumper wires.
7. 1 mobile phone (smart phone) or system with Telegram.




```
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
# to use Raspberry Pi board pin numbers
GPIO.setmode(GPIO.BOARD)
# set up GPIO output channel
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)
```

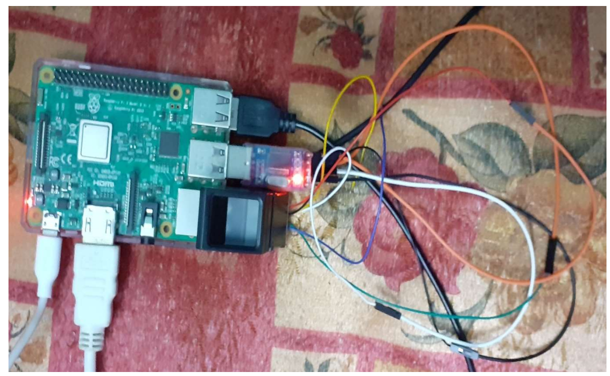
Practical 6: Fingerprint Sensor interfacing with Raspberry Pi

https://tutorials-raspberrypi.com/how-to-use-raspberry-pi-fingerprint-sensor-authentication/

- **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.
- **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.
- **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




```
import time
from 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, 0XFFFFFFFF, 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 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 i==3:
            deleteFinger()
        elif i==4:
            break
        else:
            print("Invalid option !!!Enter Correct")
```

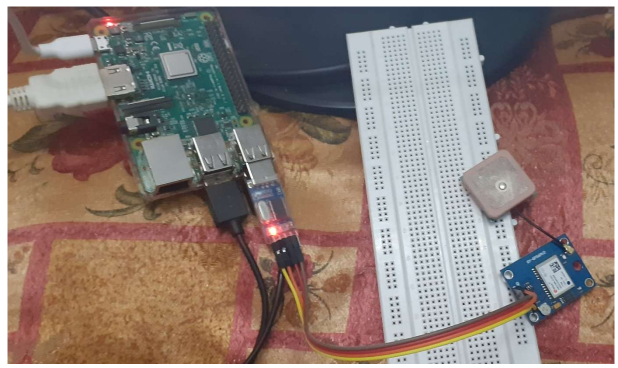
Practical 7: Raspberry Pi GPS Module Interfacing

https://www.instructables.com/Interfacing-GPS-Module-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), 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.
- **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

- Vcc to Pin 2
- Rx to Pin 8 (Tx)
- Tx to Pin 10 (Rx)
- Gnd to Pin 6



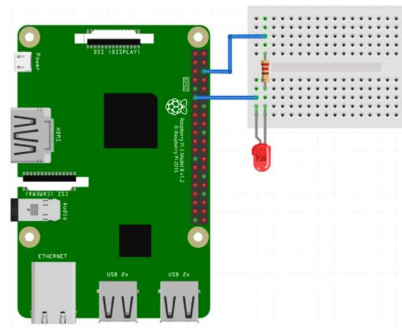
Save → gpss.py
Run → terminal → sudo python gpss.py


```
import serial
Import time
import string
import pynmea2

while True:
  port="/dev/ttyAMAO"
  ser=serial.Serial(port,baudrate=9600,timeout=0.5)
  dataout =pynmea2.NMEAStreamReader()
  newdata=ser.readline()
  if newdata[0:6]=="$GPRMC":
     newmsg=pynmea2.parse(newdata)
     lat=newmsg.latitude
     lng=newmsg.longitude
     gps="Latitude=" +str(lat) + "and Longitude=" +str(lng)
     print(gps)
```

Practical 8: IoT based Web Controlled Home Automation using Raspberry Pi

- **Components Required:** Raspberry Pi Model B+, 5V Micro USB Power Supply, VGA to HDMI Converters, Monitors, USB Keyboard & Mouse, 1 LED lights, 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 convertor. 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 220 Ω 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:**

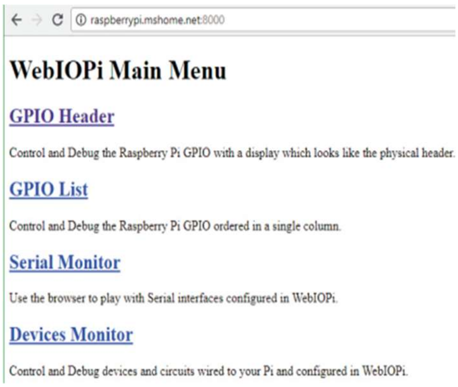
1. Connect VGA to HDMI convertor, with Raspberry Pi & monitor.
2. Connect the USB keyboard and mouse with Raspberry Pi.
3. Connect the micro USB 5v Power supply to Raspberry Pi.
4. Connect Raspberry Pi to breadboard using jumper wires.
5. Connect the cathode of LED to the Ground pins.
6. Connect anode of the LED with resistors.

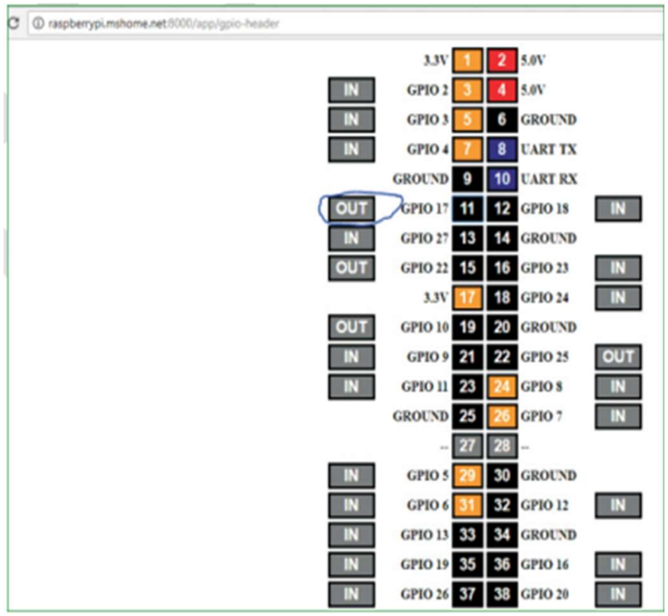



```
>sudo apt-get update
>sudo apt-get upgrade
>sudo reboot
>cd ~
>wget http://sourceforge.net/projects/webiopi/files/WebIOPi-0.7.1.tar.gz
>tar xvzf WebIOPi-0.7.1.tar.gz
>cd WebIOPi-0.7.1/
>wget https://raw.githubusercontent.com/doublebind/raspi/master/webiopi-pi2bplus.patch
>patch -p1 -i webiopi-pi2bplus.patch
>sudo chmod 775 setup.sh
>sudo ./setup.sh
>sudo reboot
>sudo webiopi -d -c /etc/webiopi/config
```

After issuing the command above on the pi, point the web browser of your computer connected to the raspberry pi to **http://thepi'sIPaddress:8000**

```
>sudo /etc/init.d/webiopi start
>sudo /etc/init.d/webiopi stop
```





Practical 9: Visitor Monitoring with Raspberry Pi & Pi Camera

Copyright © PROF. GUFRAN QURESHI

- **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.
- **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. Pi Camera Module.

- Open the Raspberry Pi camera notch
- Insert camera module in the notch
- Close the notch.

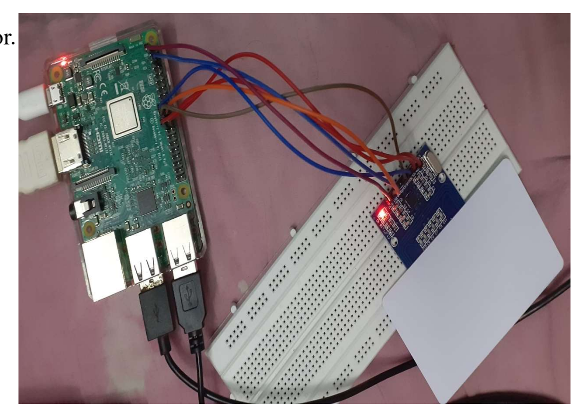
```
from picamera import PiCamera
from time import sleep

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()
```

Practical 10: 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 RF
- ID 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 13.56MHz electromagnetic field that it uses to communicate with the RFID tags.
- **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. Connect RFID RC522 to Raspberry Pi using jumper wires.

- **SDA** connects to **Pin 24**.
- **SCK** connects to **Pin 23**.
- **MOSI** connects to **Pin 19**.
- **MISO** connects to **Pin 21**.
- **GND** connects to **Pin 6**.
- **RST** connects to **Pin 22**.
- **3.3v** connects to **Pin 1**.



Write Code:

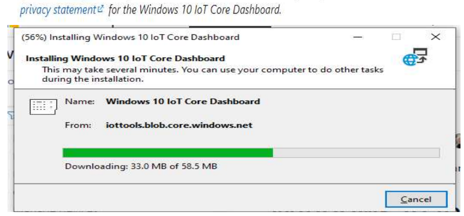
```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522

rfid = SimpleMFRC522()

try:
        print("Hold tag near the module...")
        rfid.write("Prof. Gufran")
        print("Written")
finally:
        GPIO.cleanup()
```

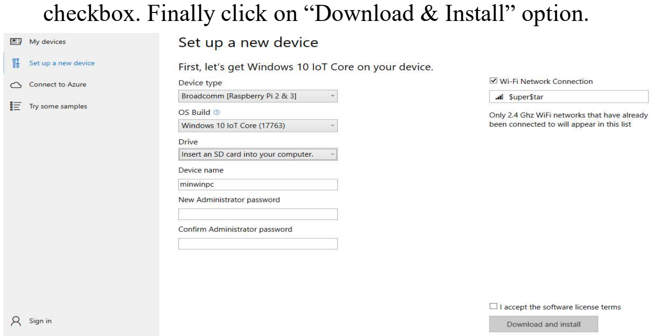
Read Code:

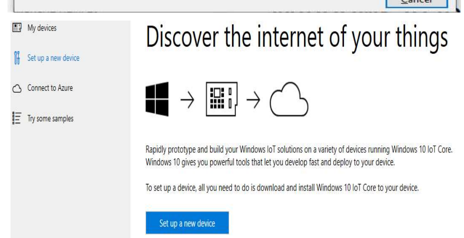
```
import RPi.GPIO as GPIO
from mfrc522 import
SimpleMFRC522

rfid = SimpleMFRC522()

while True:
        id, text = rfid.read()
        print(id)
        print(text)
```


Practical 11: 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”.