

### **PIR (motion) sensor**

```
import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(24,GPIO.OUT)

GPIO.setup(18,GPIO.IN)


while(True):

    myin=GPIO.input(18)

    if myin==True:

        print("Motion Detected")

        time.sleep(1)

        GPIO.output(24,True)

        time.sleep(0.5)

        GPIO.output(24,False)

        time.sleep(0.5)
```

### **IR sensor**

```
import RPi.GPIO as IO

IO.setwarnings(False)

IO.setmode(IO.BOARD)

IO.setup(8,IO.IN)

IO.setup(3,IO.OUT)


while 1:

    if(IO.input(8)==True):
```

```
        print("Obstacle Detected!!")
        IO.output(3,True)
    else
        print("Obstacle Not Detected")
        IO.output(3,False)
```

### **Ultrasonic Sensor**

```
import RPi.GPIO as GPIO

import time

TRG=21

ECHO=20


GPIO.setmode(GPIO.BCM)

while True:

    print("distance measurement in progress");


    GPIO.setup(TRIG,GPIO.OUT)

    GPIO.setup( ECHO,GPIO.IN)

    GPIO.output(TRIG,False)

    print("waiting for sensor to settle");


    time.sleep(0.2)

    GPIO.output(TRIG,True)

    time.sleep(0.00001)

    GPIO.output(TRIG,False)


    while GPIO.input(ECHO)==0:
```

```
        pulse_start=time.time()

while GPIO.input(ECHO)==1:

    pulse_end=time.time()

    pulse_duration=pulse_end-pulse_start

    distance=pulse_duration*17150

    distance=round(distance,2)

    print("distance:",distance,"cm")

    time.sleep(2)
```

## **LDR**

```
int ldr;

void setup() {

    pinMode(13,OUTPUT);

    Serial.begin(9600);

}

void loop() {

    ldr = analogRead(A0);

    if(ldr<50)

    {

        digitalWrite(ledPin,HIGH);

    }

    else

    {

        digitalWrite(ledPin,LOW);

    }
```

```

        Serial.println(ldr);

        delay(100);

    }

```

### UART with tx-rx

Arduino	RPi
<b>Tx</b>	<b>8</b>
<b>Rx</b>	<b>10</b>
<b>Gnd</b>	<b>6</b>

```

import serial

from time import sleep

ser=serial.Serial('/dev/ttyS0',9600)

```

```

while True:

    received_data=ser.read()

    sleep(0.03)

    data_left=ser.inWaiting()

    received_data+=ser.read(data_left)

    print(received_data)

```

Note : run **sudo chmod 666 /dev/ttyS0** to give permission to use ttyS0 before executing code

### USB UART with RPi direct communication

Arduino	RPi
<b>Tx</b>	<b>Tx</b>
<b>Rx</b>	<b>Rx</b>
<b>Gnd</b>	<b>Gnd</b>

```

import serial

if name=='main':

```

```

ser=serial.Serial('/dev/ttyACM0',9600,timeout=1)

ser.flush()

while True:

    if ser.in_waiting>0:

        line=ser.readline().decode('utf-8').rstrip()

        print(line)

```

Note : run **sudo chmod 666 /dev/ttyACM0** to give permission to use ttyS0 before executing code

### Arduino Code For USB UART

```

void setup()

{

    Serial.begin(9600);

}

void loop()

{

    Serial.println("Hello From Arduino!");

    delay(1000);

}

```

- 
1. Connect IR sensor with RPi .sense object using IR sensor and update timing in SQLite db also publish msg “object detected” on MQTT topic name :: “IR\_Sensor” and display time on screen

IR	RPi
Data	3 (without using LED) 8 (while using LED)
Vcc	2
Gnd	6

If using LED

LED	RPi
-	ground
+	3

```
import RPi.GPIO as IO
```

```
import time
```

```
import sqlite3
```

```
import paho.mqtt.client as mqtt
```

```
connection = sqlite3.connect('test.db')
```

```
cursor = connection.cursor()
```

```
cursor.execute("""CREATE TABLE irTB ( sensorData VARCHAR(100), currentTime  
TIMESTAMP);""")
```

```
insertQuery = """INSERT INTO irTB  VALUES (?, ?, ?);"""
```

```
IO.setwarnings(False)
```

```
IO.setmode(IO.BOARD)
```

```
IO.setup(8,IO.IN)
```

```
IO.setup(3,IO.OUT)
```

```
while 1:
```

```
    if(IO.input(8)==True):
```

```
        print("Obstacle Detected!!")
```

```
        cursor.execute(insertQuery, ("Obstacle Detected", currentDateTime))
```

```
        connection.commit()
```

```
        broker_url = "broker.emqx.io"
```

```
        broker_port = 1883
```

```

client = mqtt.Client()

client.connect(broker_url,broker_port)


client.publish(topic="IR_Sensor", payload="object detected", qos=0, retain=False)

IO.output(3,True)

else

print("Obstacle Not Detected")

IO.output(3,False)

cur.execute("select * from irTB ");

ans=cur.fetchall();

for row in ans:

    print(ans);

connection.close()

```

2. Sense light intensity using LDR on arduino board and send sensed light intensity value to RPi using i2c bus on RPi received content through i2c bus is update in SQLite DB and display on screen

Requirement : Arduino, LDR, RPi, cable

I2c :

Arduino	RPi
A4-SDA	3-SDA
A5-SCL	5-SCL
GND	20

LDR : connects with arduino because RPi has no analog convertor pin

LDR	Arduino
2 joint leg	A0
Register	GND
LDR	5v

*LDR arduino code : (Slave)*

```
#include<Wire.h>
```

```
int ldr; //set A0 (Analog Input) for LDR
```

```
void setup(){
```

```
    //join i2c bus as slave with address 8
```

```
    Wire.begin(0x8);
```

```
    Serial.begin(9600);
```

```
    //call receiveEvent when data received
```

```
    Wire.onReceive(receiveEvent);
```

```
    //call get request from master
```

```
    Wire.onRequest(sendData);
```

```
    pinMode(13,OUTPUT);
```

```
    digitalWrite(13,LOW);
```

```
//fn that executes whenever data is received from master
```

```
void receiveEvent (int howMany){
```

```
    while(Wire.available()){
```

```
        c=Wire.read(); //receive byte as a character
```

```
        digitalWrite(13,c);
```

```
    }
```

```
}
```

```
void loop(){
```

```
    ldr=analogRead(A0);    //reads the value of LDR(light)
```

```
    if(ldr<50)
```

```
    {        digitalWrite(13,HIGH);        }
```

```
    else
```

```
    {        digitalWrite(13,LOW);        }
```



```

        Serial.println(ldr);      //prints the value of ldr to Serial monitor

        delay(100);

    }      //Serial.println(value);

    void sendData(){

        Wire.write(ldr);

    }

```

*I2CPY* : (Master)

```
from smbus import SMBus
```

```
import time
```

```
import sqlite3
```

```
conn=sqlite3.connect("testDB.db");
```

```
cur=conn.cursor();
```

```
conn.execute("""CREATE TABLE ldrTB(LDRDATA TEXT NOT NULL);""")
```

```
conn.commit();
```

```
addr = 0x8
```

```
bus = SMBus(1)
```

```
print("Enter 1 for ON or 0 for OFF")
```

```
while 1:
```

```
    y = bus.read_byte_data(addr,0x1)
```

```
    print (y)
```

```
    cur.execute("""INSERT INTO ldrTB(LDRDATA) VALUES(?)",(y));
```

```
    conn.commit()
```

```
cur.execute("""select * from ldrTB""");
```

```
ans=cur.fetchall();
```

```
for row in ans:
```

```
    print(ans);
```

```
conn.commit()
```

```
conn.close()
```

3. Connect ultrasonic sensor with RPi. Sense object distance and update distance and timing in SQLite DB also publish measured distance value on MQTT topic name :: "ultra\_Sensor" and display on screen

Ultrasonic	RPi
ECHO	38(GPIO 20)
TRIG	40(GPIO 21)
VCC	2
GND	39

```
import RPi.GPIO as GPIO
```

```
import time
```

```
import sqlite3
```

```
import paho.mqtt.client as mqtt
```

```
TRIG=16
```

```
ECHO=20
```

```
connection = sqlite3.connect('test.db')
```

```
cursor = connection.cursor()
```

```
cursor.execute("""CREATE TABLE ultraTB (sensorData VARCHAR(100), currentTime  
TIMESTAMP);""")
```

```
insertQuery = """INSERT INTO ultraTB VALUES (?, ?, ?);"""
```

```
GPIO.setmode(GPIO.BCM)
```

```
while True:
```

```
    print("distance measurement in progress");
```

```
GPIO.setup(TRIG,GPIO.OUT)

GPIO.setup(ECHO,GPIO.IN)


GPIO.output(TRIG,False)

print("waiting for sensor to settle");


time.sleep(0.2)

GPIO.output(TRIG,True)


time.sleep(0.00001)

GPIO.output(TRIG,False)


while GPIO.input(ECHO)==0:

    pulse_start=time.time()

while GPIO.input(ECHO)==1:

    pulse_end=time.time()


pulse_duration=pulse_end-pulse_start

dist=pulse_duration*17150


dist=round(dist,2)

print("distance:",dist,"cm")

time.sleep(0.1)


cursor.execute(insertQuery, (dist, currentDateTime))

connection.commit()
```

```

broker_url = "broker.emqx.io"

broker_port = 1883


client = mqtt.Client()

client.connect(broker_url,broker_port)


client.publish(topic="UltraMP", payload=dist, qos=0, retain= False)

cur.execute("select * from ultraTB ");

ans=cur.fetchall();

for row in ans:

    print(ans);

connection.close()

```

4. Sense light intensity using LDR on arduino board and send sensed light intensity value to RPi using SPI bus on RPi received content through SPI bus is update in SQLite DB and display on screen

Requirement : Arduino, LDR, RPi, cable

SPI :

Arduino	RPi
11	19-MOSI
12	21-MISO
13	23-SCLK
GND	25-ground

LDR : connects with arduino because RPi has no analog convertor pin

LDR	Arduino
2 joint leg	A0
Register	GND
LDR	5v

5. Use virtual box to create SQLite DB with name manualrollno DB. Insert following fields in DB table "Sequence no, name, age, gender" manually. Insert atleast 5 to 6 records in DB. Once

DB record entry completed then access it one by one and publish on topic name :: Virtual\_A and display on screen

```
import paho.mqtt.client as mqtt
```

```
import sqlite3
```

```
connection=sqlite3.connect("manualrollno.db");
```

```
cur=connection.cursor();
```

```
create_tbl="""CREATE TABLE IF NOT EXISTS TEST(sequenceno INT, name TEXT, age INT, gender TEXT)"""
```

```
cur.execute(create_tbl);
```

```
connection.commit();
```

```
//insert 5-6 data manually
```

```
broker_url = "broker.emqx.io"
```

```
broker_port = 1883
```

```
client = mqtt.Client()
```

```
client.connect(broker_url, broker_port)
```

```
cur.execute("select * from TEST");
```

```
ans=cur.fetchall();
```

```
for row in ans:
```

```
    print(ans);
```

```
    client.publish(topic=" Virtual_A ",payload=ans,qos=0,retain=False)
```

```
connection.commit();
```

```
client.loop_forever()
```

6. Connect PIR sensor with RPi . sense object using PIR sensor and update timing in SQLite db also publish msg “PIR object detected” on MQTT topic name :: “PIR\_Sensor” and display time on screen

PIR	RPi
GND	6
VCC	2
OUT(DATA)	12

If using LED

LED	RPi
-	Ground
+	Output-18

PIRpy :

```
import RPi.GPIO as GPIO
```

```
import time
```

```
import sqlite3
```

```
import paho.mqtt.client as mqtt
```

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(24,GPIO.OUT)
```

```
GPIO.setup(18,GPIO.IN)
```

```
connection = sqlite3.connect('test.db')
```

```
cursor = connection.cursor()
```

```
cursor.execute("""CREATE TABLE pirTB (
```

```
    sensorData VARCHAR(100),
```

```
    currentTime TIMESTAMP);""")
```

```
insertQuery = """INSERT INTO pirTB
```

```
VALUES (?, ?, ?);"""
```

```

while(True):

    myin=GPIO.input(18)

    if myin==True:

        print("Motion Detected")

        cursor.execute(insertQuery, ("Motion Detected", currentDateTime))

        connection.commit()


        broker_url = "broker.emqx.io"

        broker_port = 1883


        client = mqtt.Client()

        client.connect(broker_url,broker_port)


        client.publish(topic="PIR_Sensor", payload="motion detected", qos=0,
        retain= False)


        time.sleep(1)

        GPIO.output(24,True)

        time.sleep(0.5)


        GPIO.output(24,False)

        time.sleep(0.5)


cur.execute("select * from pirTB ");

ans=cur.fetchall();

for row in ans:

    print(ans);

connection.close()

```

