

Name: Mainak Sen

Roll No: 211115

1.Ultrasonic Sensor

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

GPIO_TRIGGER = 18
GPIO_ECHO = 24

GPIO.setup(GPIO_TRIGGER , GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)

def distance():
    GPIO.output(GPIO_TRIGGER, True )
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False )

    starttime = time.time()
    stoptime = time.time()

    while GPIO.input(GPIO_ECHO)==0:
        starttime = time.time()

    while GPIO.input(GPIO_ECHO)==1:
        stoptime = time.time()

    timeescaped = stoptime - starttime

    distance = (timeescaped * 34300) /2
    return distance

if __name__ == '__main__':
    try:
        while True:
            dist = distance()
            print(f'Measured distance = {round(dist)}')
            # add the requests code here + imported
            time.sleep(0.2)
    except KeyboardInterrupt:
```

```
print('why you stopped ??????????????????')
GPIO.cleanup()
```

2.Light Sensor

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
LIGHT_PIN = 24
GPIO.setup(LIGHT_PIN, GPIO.IN)
lold = not GPIO.input(LIGHT_PIN)
print('Starting up the LIGHT Module (click on STOP to exit)')
time.sleep(0.2)
while True:
    if GPIO.input(LIGHT_PIN) <= 10:
        if GPIO.input(LIGHT_PIN):
            print ('\u263e')
        else:
            print ('\u263c')
    lold = GPIO.input(LIGHT_PIN)
    time.sleep(0.2)
```

3.IR Sensor

```
import RPi.GPIO as GPIO
import time

sensor = 16
#buzzer = 18

GPIO.setmode(GPIO.BCM)
GPIO.setup(sensor,GPIO.IN)
#GPIO.setup(buzzer,GPIO.OUT)

#GPIO.output(buzzer,False)
print("IR Sensor Ready.....")
print(" ")

while True:
    if not GPIO.input(sensor):
        #print(GPIO.input(sensor))
        #GPIO.output(buzzer,True)

        print("Object Detected")
        time.sleep(0.5)
    else:
        print("Object not Deteceted")
        time.sleep(0.5)
```

4. Rain sensor

```
from time import sleep
from gpiozero import Buzzer, InputDevice

#buzz = Buzzer(13)
no_rain = InputDevice(18)

def buzz_now(iterations):
    for x in range(iterations):
        #buzz.on()
        sleep(0.1)
        buzz.off()
        sleep(0.1)

while True:
    if not no_rain.is_active:
        print("It's raining - get the washing in!")
        #buzz_now(5)
        # insert your other code or functions here
        # e.g. tweet, SMS, email, take a photo etc.
        sleep(1)
```

i.e. The buzz codes are commented out as we were not given any buzzer. But pin 13 can be used for that as per the code

5.16 x 2 LCD Display

```
#!/usr/bin/python
# Example using a character LCD connected to a Raspberry Pi
import time
import Adafruit_CharLCD as LCD

# Raspberry Pi pin setup
lcd_rs = 25
lcd_en = 24
lcd_d4 = 23
lcd_d5 = 17
lcd_d6 = 18
lcd_d7 = 22
lcd_backlight = 2
```

```
# Define LCD column and row size for 16x2 LCD.
lcd_columns = 16
lcd_rows = 2

lcd = LCD.Adafruit_CharLCD(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6,
lcd_d7, lcd_columns, lcd_rows, lcd_backlight)

lcd.message('Hello\nworld!')
# Wait 5 seconds

time.sleep(5.0)
lcd.clear()
text = raw_input("Type Something to be displayed: ")
lcd.message(text)

# Wait 5 seconds
time.sleep(5.0)
lcd.clear()
lcd.message('Goodbye\nWorld!')

time.sleep(5.0)
lcd.clear()
```

6. Bluetooth to receive data from Phone Bluetooth console

```
#include <SoftwareSerial.h>
SoftwareSerial EEBlue(10, 11); // RX | TX
void setup()
{
    Serial.begin(9600);
    EEBlue.begin(9600); //Default Baud for comm, it may be
    different for your Module.
    Serial.println("The bluetooth gates are open.\n Connect to HC-05
    from any other bluetooth device with 1234 as pairing key!");
}

void loop()
{
    // Feed any data from bluetooth to Terminal.
    if (EEBlue.available())
        Serial.write(EEBlue.read());

    // Feed all data from termial to bluetooth
    if (Serial.available())
        EEBlue.write(Serial.read());
}
```