PYTHON CODE (GAS, TEMPERATURE, HUMIDITY, PRESSURE)

Date	17 Nov 2022
Team ID	PNT2022TMID30436
Project Name	Gas Leakage Monitoring and Alerting System
Maximum Mark	4 marks

PYTHON CODE

```
  #include <LiquidCrystal.h>
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "onj4zr",
        "typeId": "test",
        "deviceId":"61"
    },
    "auth": {
```

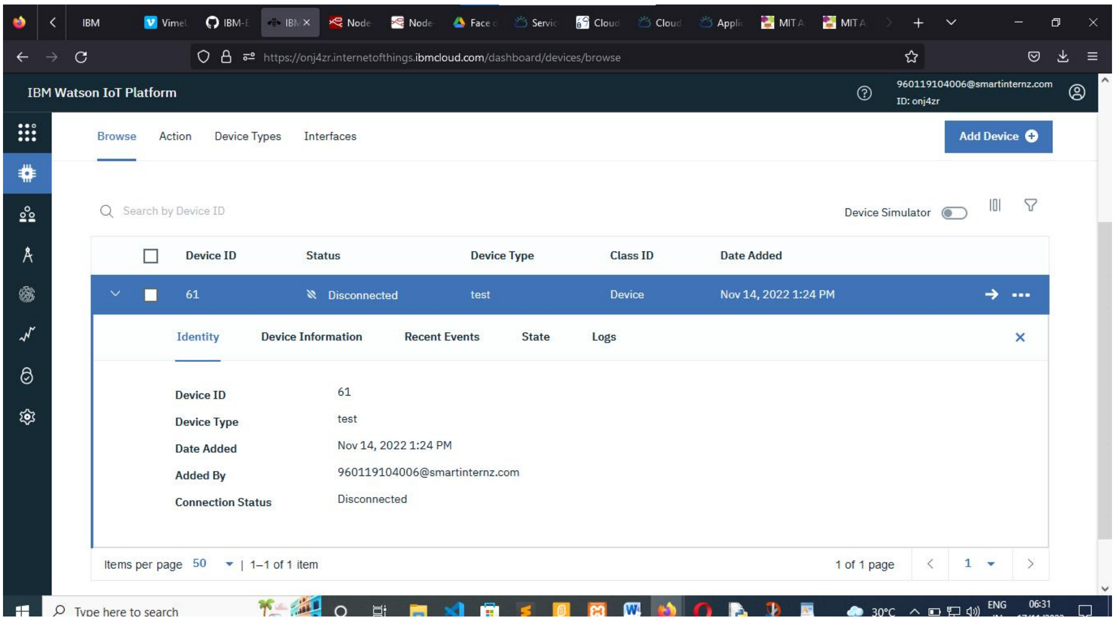
```
                "token": "Wlu5ClM7!7-r@Ot+9w"
        }
}

def myCommandCallback(cmd):
        print("Message received from IBM IoT Platform: %s" %
cmd.data['command'])
        m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig,
logHandlers=None)
client.connect()



#LiquidCrystal lcd(6, 7, 8, 9, 10, 11);
float gasPin = A0;
float gasLevel;
int ledPin = 2;
int buttonPin = 3;
int buzzPin = 4;
int buttonState;
int fan = 5;
```

```
void setup(){
   pinMode(ledPin, OUTPUT);
   pinMode(buttonPin, INPUT);
   pinMode(gasPin,INPUT);
   pinMode(fan,OUTPUT);
   Serial.begin(9600);
   lcd.begin(16, 2);
   lcd.setCursor(0,0);
   lcd.print(" Welcome");
   lcd.setCursor(0,2);
   lcd.print("PNT2022TMID51246");
   delay(500);
   lcd.clear();
}

void loop(){
   // Read the value from gas sensor and button
   gasLevel = analogRead(gasPin);
   buttonState = digitalRead(buttonPin);

   // call the function for gas detection and button work
```

```
    gasDetected(gasLevel);
    buzzer(gasLevel);
    exhaustFanOn(buttonState);
}

// Gas Leakage Detection & Automatic Alarm and Fan ON
void gasDetected(float gasLevel){
    if(gasLevel >= 200){
      digitalWrite(buzzPin,HIGH);
        digitalWrite(ledPin,HIGH);
        digitalWrite(fan,HIGH);
        lcd.setCursor(0,0);
      lcd.print("GAS:");
        lcd.print(gasLevel);
      lcd.setCursor(0,2);
      lcd.print("FAN ON");
      delay(1000);
      lcd.clear();
    }else{
      digitalWrite(ledPin,LOW);
        digitalWrite(buzzPin,LOW);
        digitalWrite(fan,LOW);
        lcd.setCursor(0,0);
      lcd.print("GAS:");
        lcd.print(gasLevel);
      lcd.setCursor(0,2);
      lcd.print("FAN OFF");
      delay(100);
      lcd.clear();
    }
}
//BUZZER
void buzzer(float gasLevel){
if(gasLevel>=200)
    {
    for(int i=0; i<=30; i=i+10)
    {
    tone(4,i);
    delay(300);
    noTone(4);
    delay(4300);
    }
    }
}
```

```
// Manually Exhaust FAN ON
void exhaustFanOn(int buttonState){
  if(buttonState == HIGH){
    digitalWrite(fan,HIGH);
    lcd.setCursor(0,0);
   lcd.print("Button State:");
    lcd.print(buttonState);
    lcd.setCursor(0,2);
    lcd.print("FAN ON");
    delay(10000);
    lcd.clear();
  }



}
```

OUTPUT:



https://onj4zr.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform

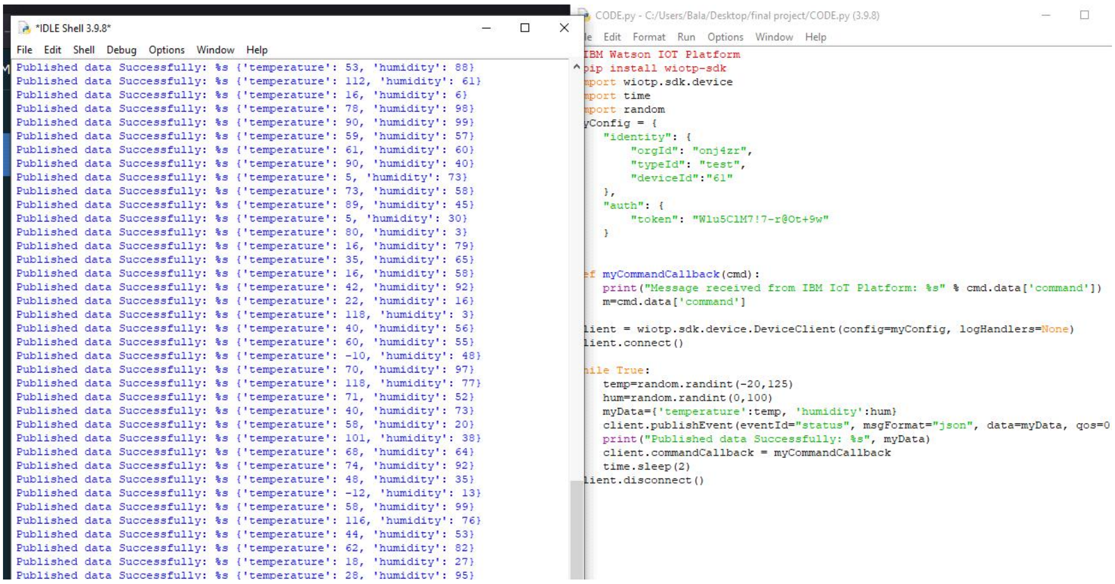
960119104006@smartinternz.com
ID: onj4zr

☐	Device ID	Status	Device Type	Class ID	Date Added
☐	61	Disconnected	test	Device	Nov 14, 2022 1:24 PM

Device ID	61
Device Type	test
Date Added	Nov 14, 2022 1:24 PM
Added By	960119104006@smartinternz.com
Connection Status	Disconnected

Items per page 50 | 1–1 of 1 item

1 of 1 page



IDLE Shell 3.9.8

```
Published data Successfully: %s {'temperature': 53, 'humidity': 88}
Published data Successfully: %s {'temperature': 112, 'humidity': 61}
Published data Successfully: %s {'temperature': 16, 'humidity': 6}
Published data Successfully: %s {'temperature': 78, 'humidity': 98}
Published data Successfully: %s {'temperature': 90, 'humidity': 99}
Published data Successfully: %s {'temperature': 59, 'humidity': 57}
Published data Successfully: %s {'temperature': 61, 'humidity': 60}
Published data Successfully: %s {'temperature': 90, 'humidity': 40}
Published data Successfully: %s {'temperature': 5, 'humidity': 73}
Published data Successfully: %s {'temperature': 73, 'humidity': 58}
Published data Successfully: %s {'temperature': 89, 'humidity': 45}
Published data Successfully: %s {'temperature': 5, 'humidity': 30}
Published data Successfully: %s {'temperature': 80, 'humidity': 3}
Published data Successfully: %s {'temperature': 16, 'humidity': 79}
Published data Successfully: %s {'temperature': 35, 'humidity': 65}
Published data Successfully: %s {'temperature': 16, 'humidity': 58}
Published data Successfully: %s {'temperature': 42, 'humidity': 92}
Published data Successfully: %s {'temperature': 22, 'humidity': 16}
Published data Successfully: %s {'temperature': 118, 'humidity': 3}
Published data Successfully: %s {'temperature': 40, 'humidity': 56}
Published data Successfully: %s {'temperature': 60, 'humidity': 55}
Published data Successfully: %s {'temperature': -10, 'humidity': 48}
Published data Successfully: %s {'temperature': 70, 'humidity': 97}
Published data Successfully: %s {'temperature': 118, 'humidity': 77}
Published data Successfully: %s {'temperature': 71, 'humidity': 52}
Published data Successfully: %s {'temperature': 40, 'humidity': 73}
Published data Successfully: %s {'temperature': 58, 'humidity': 20}
Published data Successfully: %s {'temperature': 101, 'humidity': 38}
Published data Successfully: %s {'temperature': 68, 'humidity': 64}
Published data Successfully: %s {'temperature': 74, 'humidity': 92}
Published data Successfully: %s {'temperature': 48, 'humidity': 35}
Published data Successfully: %s {'temperature': -12, 'humidity': 13}
Published data Successfully: %s {'temperature': 58, 'humidity': 99}
Published data Successfully: %s {'temperature': 116, 'humidity': 76}
Published data Successfully: %s {'temperature': 44, 'humidity': 53}
Published data Successfully: %s {'temperature': 62, 'humidity': 82}
Published data Successfully: %s {'temperature': 18, 'humidity': 27}
Published data Successfully: %s {'temperature': 28, 'humidity': 95}
```

CODE.py - C:/Users/Bala/Desktop/final project/CODE.py (3.9.8)

```
IBM Watson IOT Platform
pip install wiotp-sdk
mport wiotp.sdk.device
mport time
mport random
yConfig = {
    "identity": {
        "orgId": "onj4zr",
        "typeId": "test",
        "deviceId":"61"
    },
    "auth": {
        "token": "W1u5C1M7!7-r@Ot+9w"
    }


ef myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

lient = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
lient.connect()

hile True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
lient.disconnect()
```