

SPRINT-1

Team ID	PNT2022TMID42646
Project Name	Smart farmer - IoT Enabled smart farming application.

Python Code

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

# Provide your IBM Watson Device Credentials
organization = "ma3ge3"
deviceType = "iot_device_2"
deviceId = "iot_device"
authMethod = "token"
authToken = "9688309024"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status = cmd.data['command']
    if status == "motoron":
        print("motor is on")
    elif status == "motoroff":
        print("motor is off")
    else:
        print("please send proper command")

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod,
                    "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
# .....

except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()
```

```
# Connect and send a datapoint "hello" with value "world" into the cloud as
an event of type "greeting" 10 times
deviceCli.connect()
```

```
while True:
```

```
    # Get Sensor Data from DHT11
```

```
    temperature = random.randint(70, 80)
```

```
    humidity = random.randint(50, 60)
```

```
    soil_moisture = random.randint(21, 40)
```

```
    data = {'temperature': temperature, 'humidity': humidity,
'soil_moisture': soil_moisture}
```

```
    # print data
```

```
    def myOnPublishCallback():
```

```
        print("Published Temperature = %s C" % temperature, "Humidity = %s
%%" % humidity, "Soil_moisture = %s %%" % soil_moisture,"to IBM Watson")
```

```
        success = deviceCli.publishEvent("venkatesh_smartfarmer", "json", data,
qos=0, on_publish=myOnPublishCallback)
```

```
        if not success:
```

```
            print("Not connected to IoTf")
```

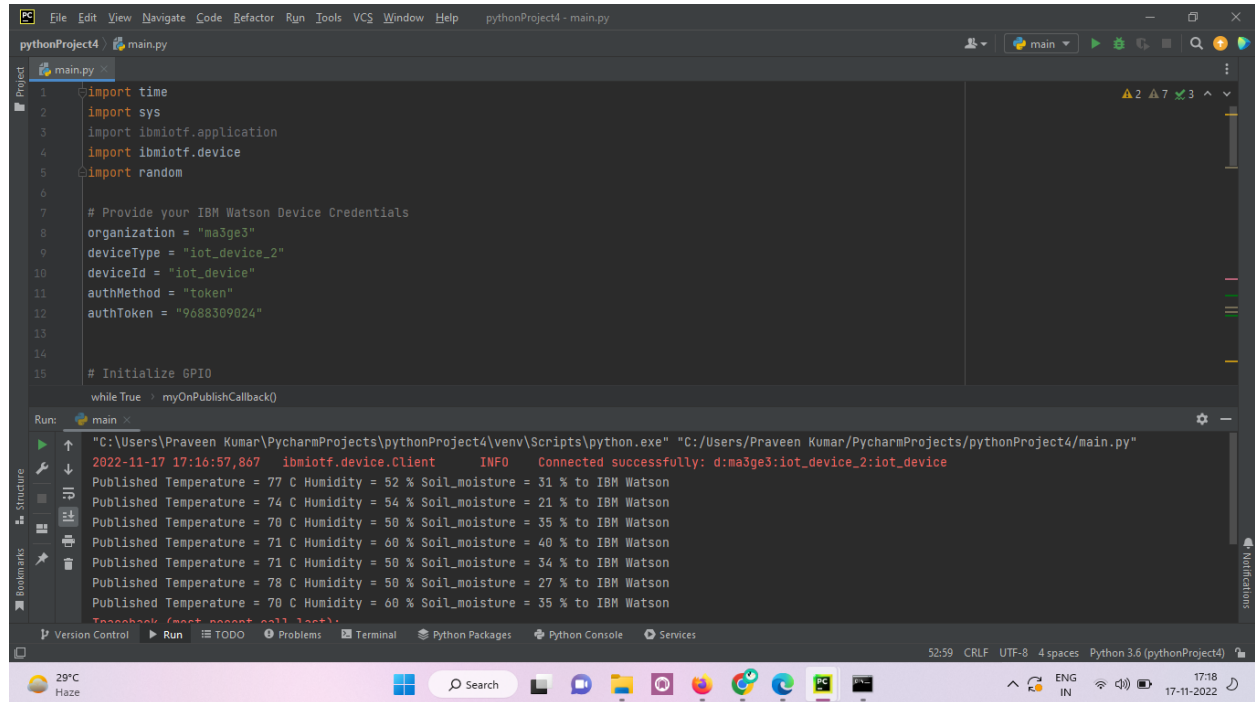
```
            time.sleep(10)
```

```
        deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device and application from the cloud
```

```
deviceCli.disconnect()
```

Output:



The screenshot displays the PyCharm IDE interface. The main editor window shows a Python script named `main.py` with the following code:

```
1 import time
2 import sys
3 import ibmiotf.application
4 import ibmiotf.device
5 import random
6
7 # Provide your IBM Watson Device Credentials
8 organization = "ma3ge3"
9 deviceType = "iot_device_2"
10 deviceId = "iot_device"
11 authMethod = "token"
12 authToken = "9688309024"
13
14 # Initialize GPIO
15 while True:
16     myOnPublishCallback()
```

The Run console at the bottom shows the execution output:

```
Run: main
"C:\Users\Praveen Kumar\PycharmProjects\pythonProject4\venv\Scripts\python.exe" "C:\Users\Praveen Kumar\PycharmProjects\pythonProject4/main.py"
2022-11-17 17:16:57,867 ibmiotf.device.Client INFO Connected successfully: d:ma3ge3:iot_device_2:iot_device
Published Temperature = 77 C Humidity = 52 % Soil_moisture = 31 % to IBM Watson
Published Temperature = 74 C Humidity = 54 % Soil_moisture = 21 % to IBM Watson
Published Temperature = 70 C Humidity = 50 % Soil_moisture = 35 % to IBM Watson
Published Temperature = 71 C Humidity = 60 % Soil_moisture = 40 % to IBM Watson
Published Temperature = 71 C Humidity = 50 % Soil_moisture = 34 % to IBM Watson
Published Temperature = 78 C Humidity = 50 % Soil_moisture = 27 % to IBM Watson
Published Temperature = 70 C Humidity = 60 % Soil_moisture = 35 % to IBM Watson
Traceback (most recent call last):
```

The bottom status bar indicates the system is at 29°C with Haze weather, and the time is 17:18 on 17-11-2022.

Connecting Sensors with ESP32 RASP using C++ code

```
#include <WiFi.h>
#include
<PubSubClient.h>
#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE
DHT22#define LED 5
DHT dht (DHTPIN, DHTTYPE);

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
#define ORG "ma3ge3"
#define DEVICE_TYPE "iot_device"
#define DEVICE_ID "iot_device_1"
#define TOKEN "M)N_yRZp8uxW43vqa-
"String data3;
float h, t;

//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";//
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
void setup()
{
    Serial.begin(115200
    );dht.begin();
    pinMode(LED,OUTPUT)
    ; delay(10);
    Serial.println();
    wificonnect();
    mqttconnect();
}
void loop()
{
    h = dht.readHumidity();
    t = dht.readTemperature();
    Serial.print("temperature:");
    Serial.println(t);
    Serial.print("humidity:");
```

```

    Serial.println(h);

    PublishData(t,h
    );delay(1000);
    if (!client.loop())
        {mqttconnect();
        }
    }
}
void PublishData(float temp, float humid)
    {mqttconnect();
    String payload =
    "{\"temperature\":\"";payload +=
    temp;
    payload += ","
    "\"humidity\":\"";payload +=
    humid;
    payload += "}";

    Serial.print("Sending payload: ");
    Serial.println(payload);

    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish ok");
    } else {
        Serial.println("Publish failed");
    }
}
}
void mqttconnect() {
    if (!client.connected()) {
        Serial.print("Reconnecting client to
        ");Serial.println(server);
        while (!client.connect(clientId, authMethod, token)) {
            Serial.print(".")
            );delay(500);
        }
        initManagedDevice();
        Serial.println();
    }
}
}
void wificonnect()
{
    Serial.println();

```

```

Serial.print("Connecting to
");

WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi
connected");Serial.println("IP
address: ");
Serial.println(WiFi.localIP());
}

void initManagedDevice() {
    if (client.subscribe(subscribetopic)) {
        Serial.println((subscribetopic));
        Serial.println("subscribe to cmd OK");
    } else {
        Serial.println("subscribe to cmd FAILED");
    }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
    Serial.print("callback invoked for topic: ");
    Serial.println(subscribetopic);
    for (int i = 0; i < payloadLength; i++) {
        //Serial.print((char)payload[i]);
        data3 += (char)payload[i];
    }

    Serial.println("data: "+
data3);if(data3=="lighton")
    {
Serial.println(data3);
digitalWrite(LED,HIGH);
    }

    else
    {
Serial.println(data3

```

```

);
digitalWrite(LED,LOW
);

}
data3="";

}

```

Circuit Diagram

