

Project Development Phase

Sprint-1

Python Script

Date	10 November 2022
Team ID	PNT2022TMID30551
Project Name	IOT-BASED CROP PROTECTION SYSTEM FOR AGRICULTURE

Description:

Instead of generating sensor values from the hardware circuits, we are using random module to generate sensor data and to automate IOT based crop protection system through the python code. The data generated from the python code are being stored in the IBM cloud.

Python Code:

```
#include <ESP8266WiFi.h>

>#include <WiFiClient.h>

#include
<PubSubClient.h>

#include "DHT.h"

const char* ssid =
"SMART-G";

const char* password = "10112019";

#define DHTPIN D6

#define G D0

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

#define ID "ryup3j"

#define DEVICE_TYPE "ESP8266"

#define DEVICE_ID "TEST"
```

```
#define TOKEN "TEST-12345"
```

```
char server[] = ID ".messaging.internetofthings.ibmcloud.com";
```

```
char publish_Topic1[] = "iot-2/evt/Data1/fmt/json";
```

```
char publish_Topic2[] = "iot-2/evt/Data2/fmt/json";
```

```
char publish_Topic3[] = "iot-2/evt/Data2/fmt/json";
```

```
char publish_Topic4[] = "iot-2/evt/Data2/fmt/json";
```

```
char authMethod[] = "use-token-auth";
```

```
char token[] = TOKEN;
```

```
char clientId[] = "d:" ID ":" DEVICE_TYPE ":" DEVICE_ID;
```

```
WiFiClient wifiClient;
```

```
PubSubClient client(server, 1883, NULL, wifiClient);
```

```
void setup() {
```

```
  pinMode(D0,OUTPUT);
```

```
  digitalWrite(D0,HIGH);
```

```
  Serial.begin(115200);
```

```
  dht.begin(); Serial.println();
```

```
    WiFi.begin(ssid, password);
```

```
    while (WiFi.status() != WL_CONNECTED) {
```

```
      delay(500);
```

```
      Serial.print(".");
```

```
    }
```

```
    Serial.println("");
```

```

Serial.println(WiFi.localIP());

if (!client.connected()) {
  Serial.print("Reconnecting client to ");
  Serial.println(server);
  while (!client.connect(clientId, authMethod, token)) {
    Serial.print(".");
    delay(500);

  }
  Serial.println("Connected TO IBM IoT cloud!");
}
}

```

```

long previous_message = 0;

void loop() {
  client.loop();
  long current = millis();
  if (current - previous_message > 3000) {
    previous_message = current;
    float hum = dht.readHumidity();
    float temp =
    dht.readTemperature();
    float MOI = map(analogRead(A0), 0, 1023, 100, 0);
    float bi = map(digitalRead(D1), 0, 1, 100, 0 );

    if (isnan(hum) || isnan(temp) ){
      Serial.println(F("Failed to read from DHT sensor!"));
    }
    return;
  }
}

```

```
}
```

```
Serial.print("Temperature: ");
```

```
Serial.print(temp);
```

```
Serial.print("°C");
```

```
Serial.print(" Humidity: ");
```

```
Serial.print(hum);
```

```
Serial.print("%");
```

```
Serial.print("SOIL MOITURE: ");
```

```
Serial.print(MOI);
```

```
Serial.print("ANIMAL AND BIRD: ");
```

```
Serial.print(bi);
```

```
if(MOI<=10)
```

```
{
```

```
    digitalWrite(D0,LOW);
```

```
    delay(100);
```

```
    digitalWrite(D0,HIGH);
```

```
}
```

```
else
```

```
{
```

```
    digitalWrite(D0,HIGH);
```

```
}
```

```
String payload = "{\"d\":{\"Name\":\"\" DEVICE_ID \"\"\"";
```

```
payload += "\",\"Temperature\":";
```

```

    payload += temp;
    payload += "}}";

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

    if (client.publish(publish_Topic1, (char*) payload.c_str())) {
        Serial.println("Published successfully");
    } else {
        Serial.println("Failed");
    }

    String payload1 = "{\"d\":{\"Name\":\"\" DEVICE_ID \"\"";
    payload1 += "\",\"Humidity\":";
    payload1 += hum;
    payload1 += "}}";
    Serial.print("Sending payload: ");
    Serial.println(payload1);
    Serial.println('\n');

    if (client.publish(publish_Topic2, (char*) payload1.c_str())) {
        Serial.println("Published successfully");
    } else {
        Serial.println("Failed");
    }
}

```

```
String payload3 = "{\"d\":{\"Name\":\"\" DEVICE_ID \"\"\"";  
payload3 += "\",\"Moiture\":\"\"";  
payload3 += MOI;  
payload3 += "\"}"}";
```

```
Serial.print("Sending payload: ");  
Serial.println(payload3);
```

```
if (client.publish(publish_Topic3, (char*) payload3.c_str())) {  
    Serial.println("Published successfully");  
} else {  
    Serial.println("Failed");  
}
```

```
String payload4 = "{\"d\":{\"Name\":\"\" DEVICE_ID \"\"\"";  
payload4 += "\",\"Animal&Bird\":\"\"";  
payload4 += bi;  
payload4 += "\"}"}";
```

```
Serial.print("Sending payload: ");  
Serial.println(payload4);
```

```
if (client.publish(publish_Topic4, (char*) payload4.c_str())) {  
  
    Serial.println("Published successfully");  
}
```

```

    } else {
        Serial.println("Failed");
    }

}

}
}

```

OUTPUT:

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. The 'Browse' tab is active, displaying a list of devices. One device, 'TEST', is selected and expanded, showing its details and recent events.

Device Details:

- ID: 12345
- Status: Disconnected
- Device Type: arduino
- Device Name: Device
- Last Seen: Nov 3, 2022 11:46 AM

Selected Device Details:

- ID: TEST
- Status: Connected
- Device Type: ESP8266
- Device Name: Device
- Last Seen: Nov 13, 2022 1:07 PM

Recent Events:

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data2	{"d":{"Name":"TEST","Animal&Bird":100}}	json	a few seconds ago
Data2	{"d":{"Name":"TEST","Moiture":55}}	json	a few seconds ago
Data2	{"d":{"Name":"TEST","Humidity":62}}	json	a few seconds ago
Data1	{"d":{"Name":"TEST","Temperature":29.8}}	json	a few seconds ago
Data2	{"d":{"Name":"TEST","Animal&Bird":0}}	json	a few seconds ago



