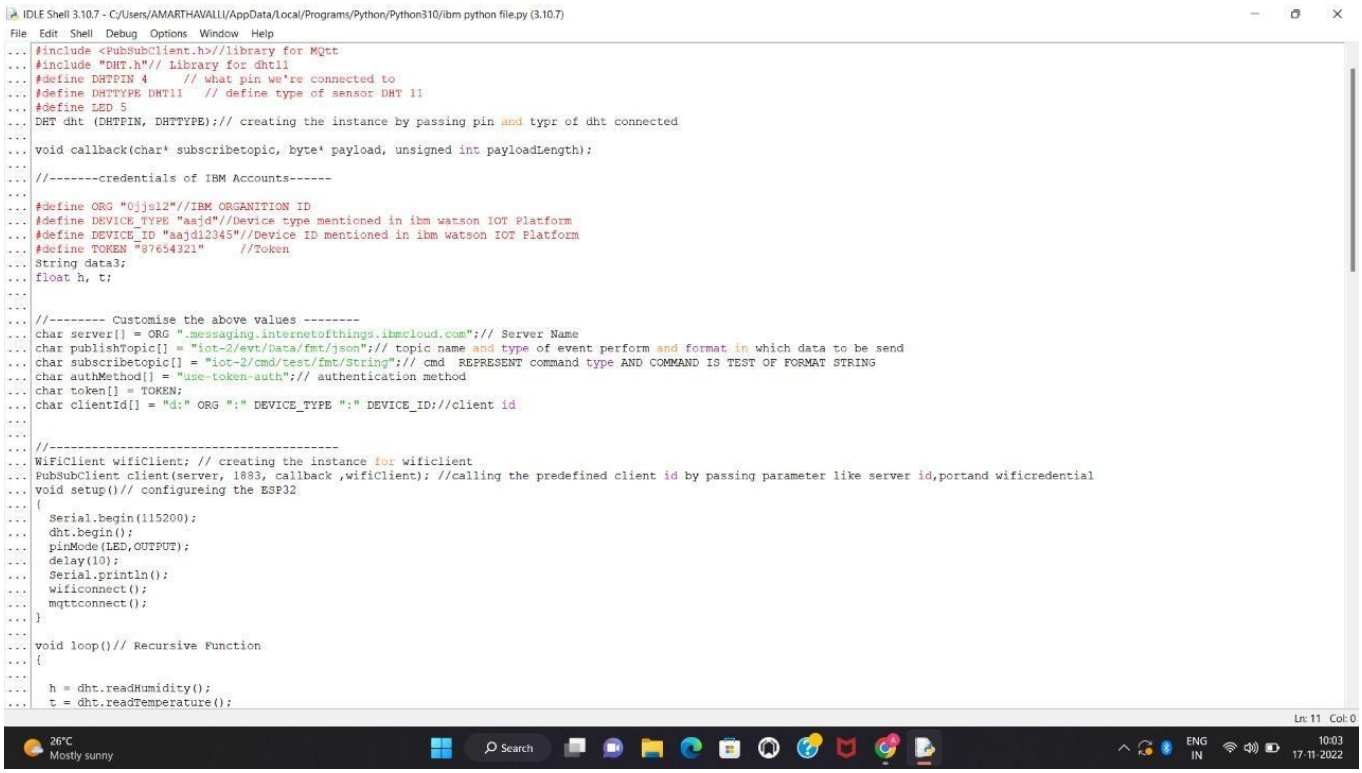
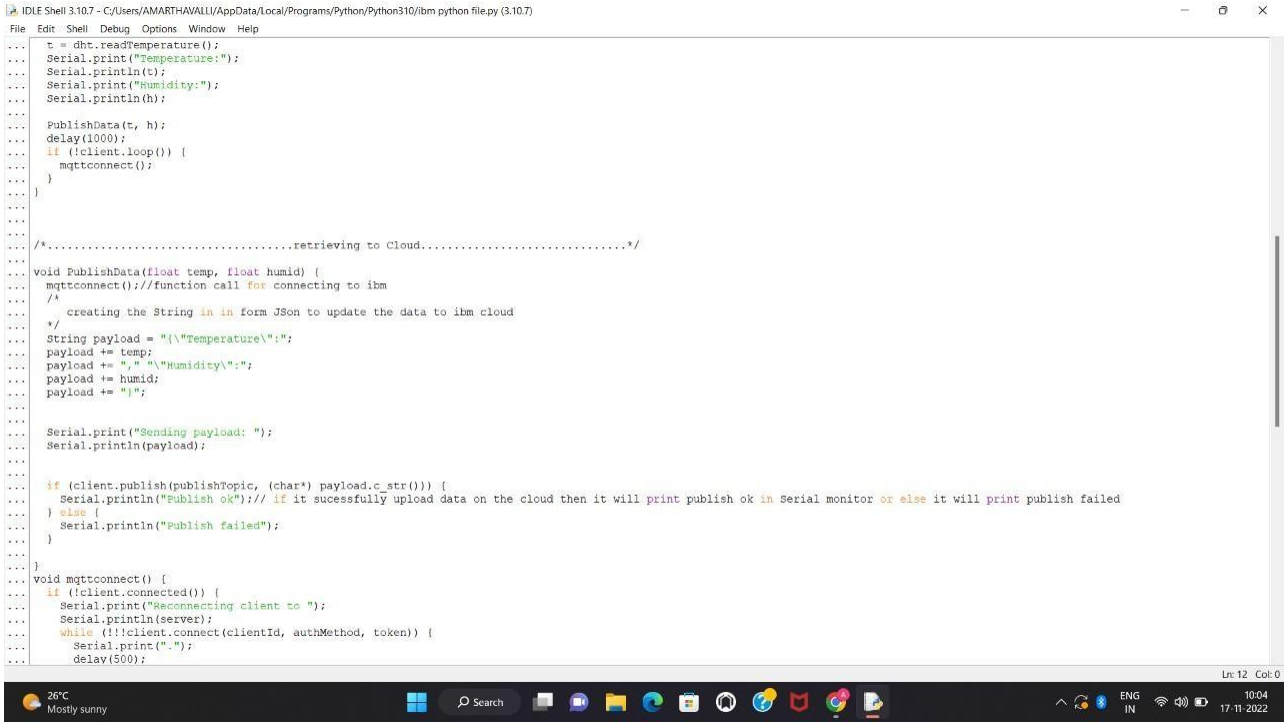
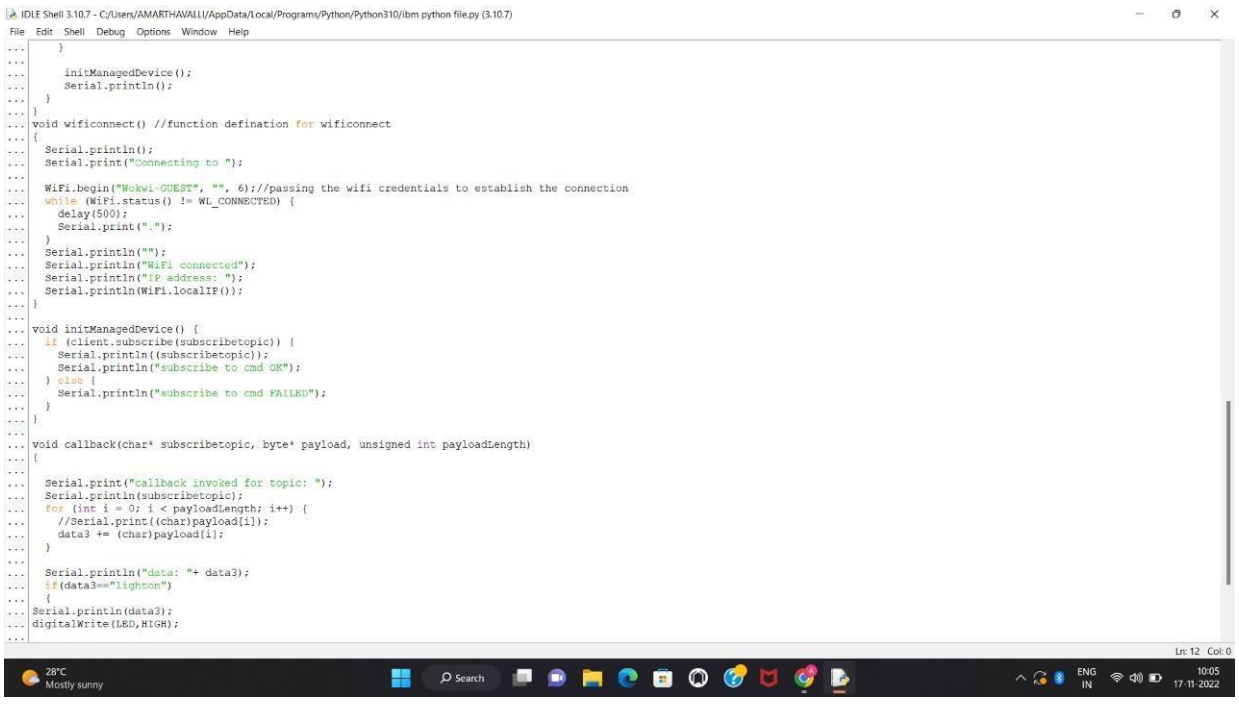
Develop the python Script

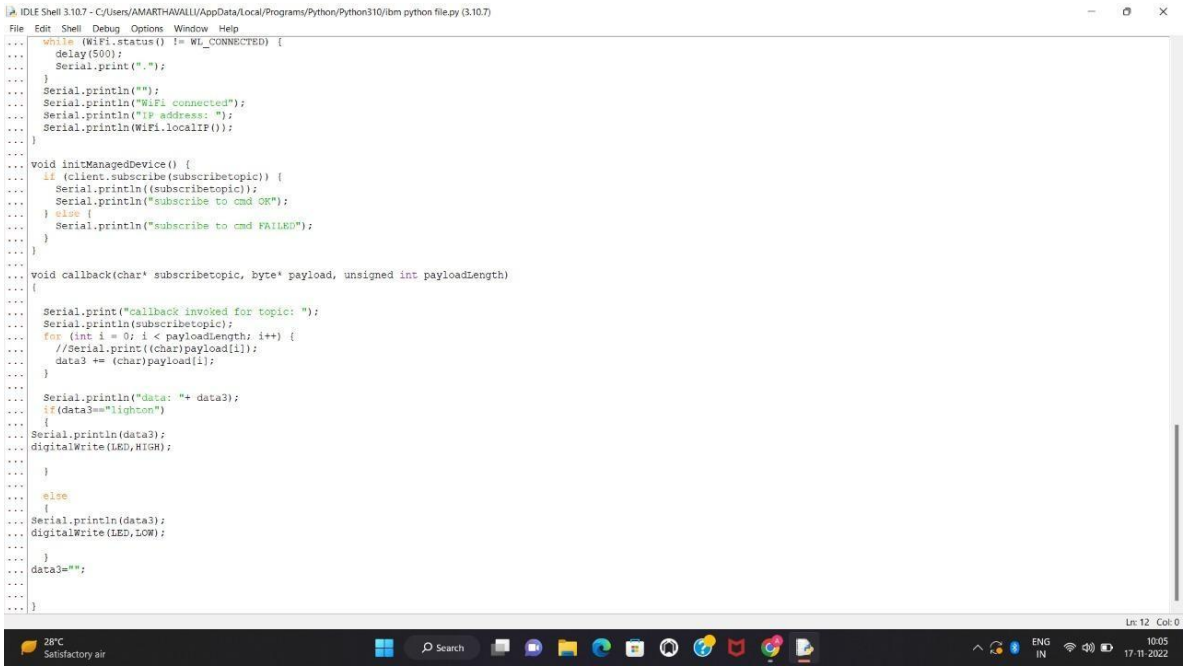
Publish Data to the IBM Cloud

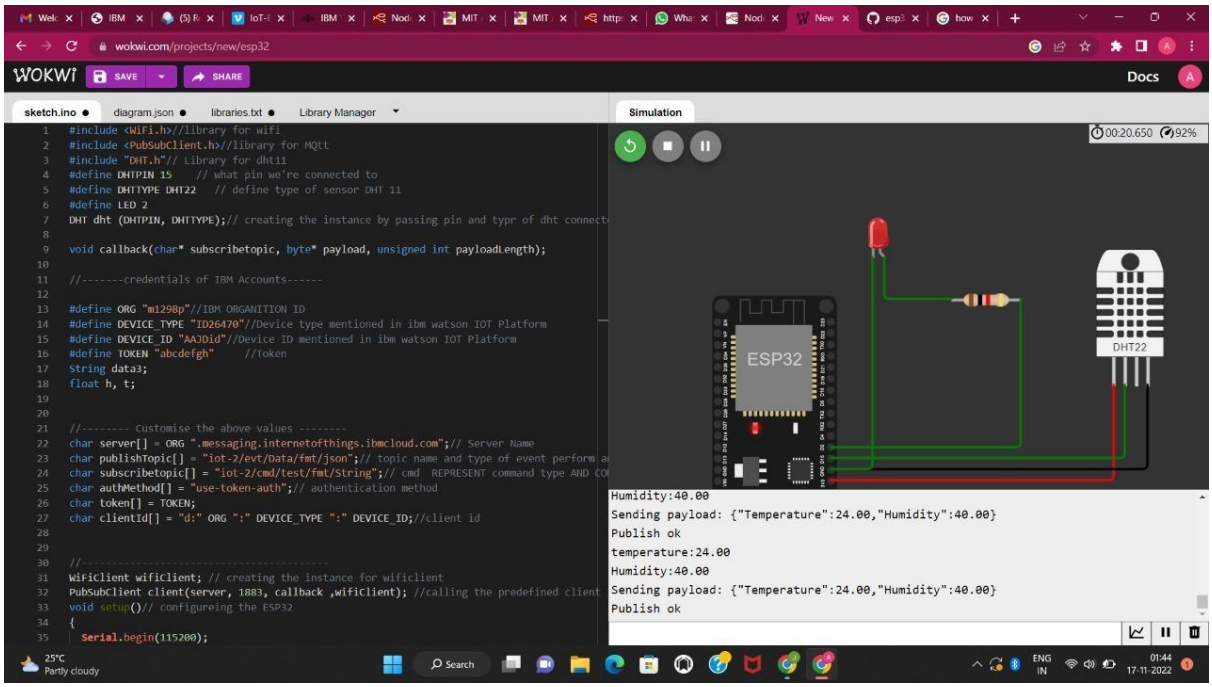
Date	18 November 2022
Team Id	PNT2022TMID19670




IDLE Shell 3.10.7 - C:/Users/AMARTHAVALLI/AppData/Local/Programs/Python/Python310/ibm python file.py (3.10.7)
File Edit Shell Debug Options Window Help
t = dht.readTemperature();
Serial.print("Temperature:");
Serial.println(t);
Serial.print("Humidity:");
Serial.println(h);
PublishData(t, h);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
/*.....retrieving to Cloud.....*/
void PublishData(float temp, float humid) {
mqttconnect();//function call for connecting to ibm
/*
creating the String in in form JSon to update the data to ibm cloud
*/
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");// if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish failed
} 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);
Ln: 12 Col: 0
26°C
Mostly sunny
Search
ENG
IN
10:04
17-11-2022


IDLE Shell 3.10.7 - C:/Users/AMARTHAVALLI/AppData/Local/Programs/Python/Python310/ibm python file.py (3.10.7)
File Edit Shell Debug Options Window Help
}
initManagedDevice();
Serial.println();
}
}
void wificonnect() //function defination for wificonnect
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
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);
Ln: 12 Col: 0
28°C
Mostly sunny
Search
ENG
IN
10:05
17-11-2022


IDLE Shell 3.10.7 - C:/Users/AMARTHAVALLI/AppData/Local/Programs/Python/Python310/ibm python file.py (3.10.7)
File Edit Shell Debug Options Window Help
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="";
}
Ln: 12 Col: 0
28°C
Satisfactory air
Search
ENG
IN
10:05
17-11-2022


wokwi.com/projects/new/esp32
WOKWI
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 15 // what pin we're connected to
#define DHTTYPE DHT22 // define type of sensor DHT 11
#define LED 2
DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht connect
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
//-------credentials of IBM Accounts------
#define ORG "m1298p"//IBM ORGANITION ID
#define DEVICE_TYPE "ID26470"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "AAJDid"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "abcdefgh" //Token
String data3;
float h, t;
//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform a
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command type AND CO
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
//---
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client
void setup()// configureing the ESP32
{
Serial.begin(115200);
Simulation
00:20.650
92%
ESP32
DHT22
Humidity:40.00
Sending payload: {"Temperature":24.00,"Humidity":40.00}
Publish ok
temperature:24.00
Humidity:40.00
Sending payload: {"Temperature":24.00,"Humidity":40.00}
Publish ok
25°C
Partly cloudy
Search
ENG
IN
01:44
17-11-2022