Sprint 4

IoT based Smart crop protection system for Agriculture

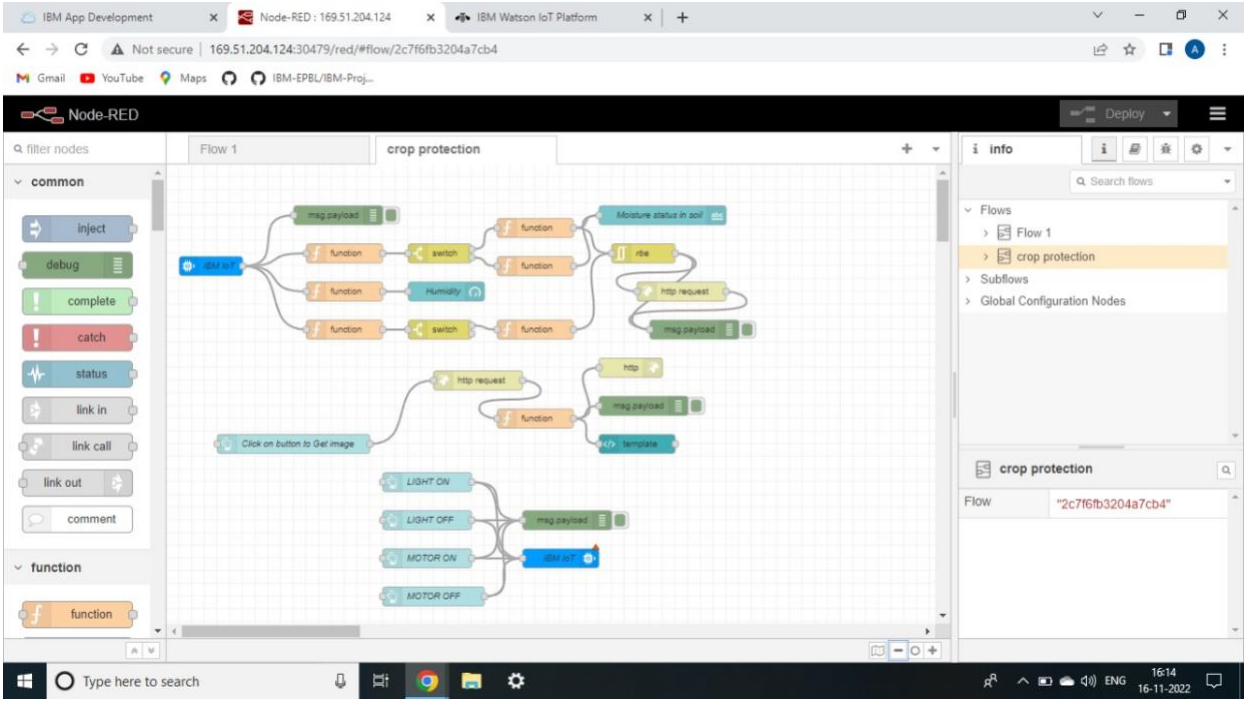
Team ID:PNT2022TMID45173

Sprint-4

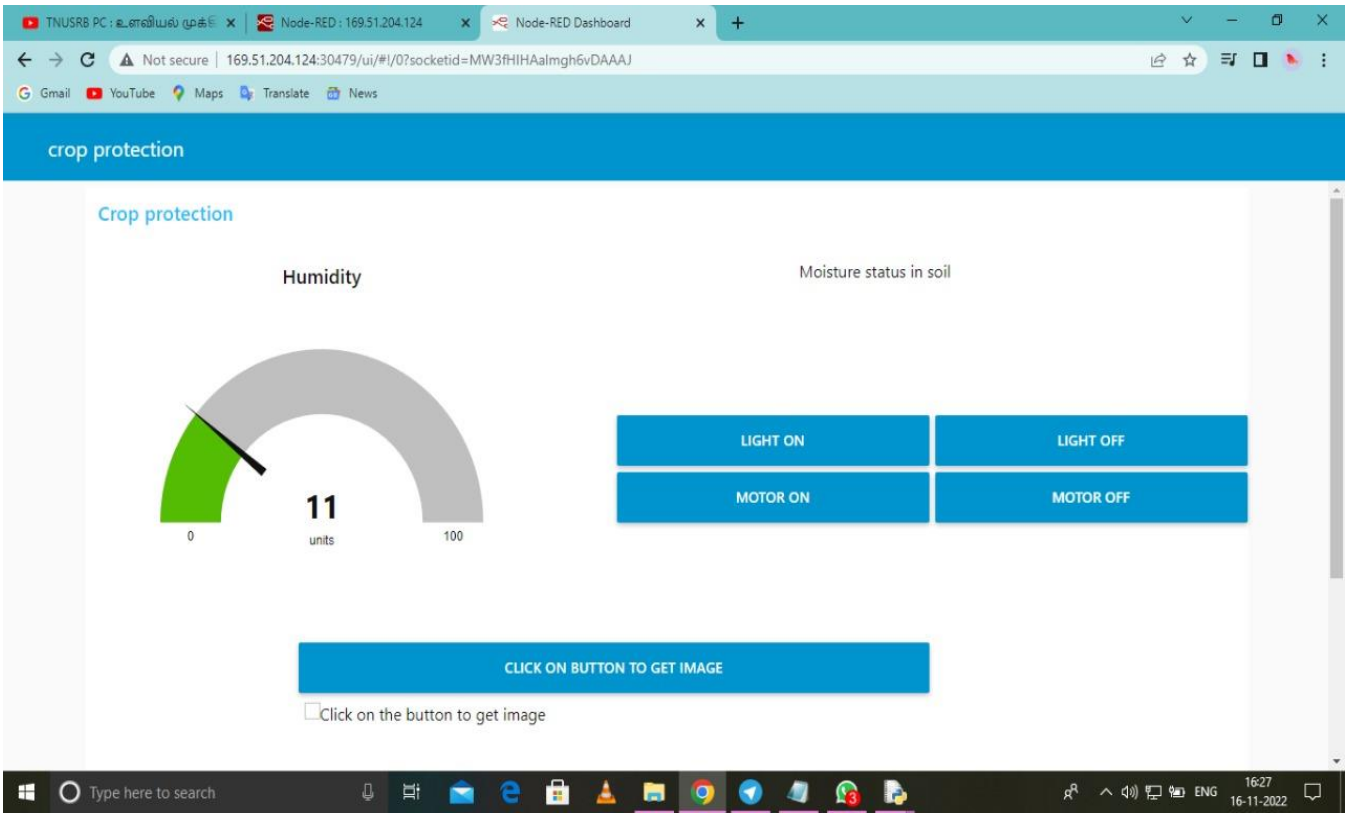
Web UI (to make the user interact with the software) / Run a simulation using the wokwi online platform

PROCESS

Using Node-Red for the Web UI process



Output in Node-RED Dashboard:



Program :

python ibm final.py - C:\Users\ELCOT\Documents\python ibm final.py (3.7.0)

File Edit Format Run Options Window Help

```python
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig = {
   "identity": {
       "orgId": "rvp7mx",
       "typeId": "NodeMCU",
       "deviceId": "12345"
   },
   "auth": {
       "token": "12345678"
   }
}
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect ()

def myCommandCallback (cmd) :
     print ("Message received from IBM IoT Platform: %s" %cmd.data['command'])
     m=cmd.data['command']
     if (m=="motoron"):
          print ("Motor is switched on")
     elif (m=="motoroff"):
          print ("Motor is switched OFF")
     print (" ")

while True:
     soil=random.randint (0,100)
     temp=random.randint (-20, 125)
     hum=random.randint (0, 100)
     myData={'soil moisture': soil, 'temperature':temp, 'humidity':hum}
     client.publishEvent (eventId="status", msgFormat="json",
     data=myData, qos=0 , onPublish=None)
     print ("Published data Successfully: %s", myData)
     time.sleep (2)
     client.commandCallback = myCommandCallback
client.disconnect ()
```

Ln: 5 Col: 13

Type here to search

ENG 19:19 17-11-2022

ibm publish.py - C:\Users\VS\OneDrive\Desktop\ibm publish.py (3.7.0)

File Edit Format Run Options Window Help

```python
authMethod = "token"
authToken = "12345678"

# 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
        moist=random.randint (0,100)
        temp=random.randint(-20,125)
        hum=random.randint(0,100)

        data = { 'moist':moist, 'temp' : temp, 'hum': hum}
        #print data
        def myOnPublishCallback():
            print ("Published temp = %s C" % temp, "hum = %s %%" % hum,"moist = %s %%" % moist, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "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()
```

Snipping Tool

Screenshot copied to clipboard and saved

Select here to mark up and share the image

Ln: 39 Col: 0

30°C Cloudy

ENG IN 13:54 16-11-2022

Output:

IDLE Shell 3.8.8

File Edit Shell Debug Options Window Help

```
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:/Users/HP/Desktop/crop/crop_protect.py ================
2021-04-06 12:52:19,640   wiotp.sdk.device.client.DeviceClient   INFO    Connecte
d successfully: d:hj5fmy:NodeMCU:12345
'sample' successfully created.
File opened
{'Animal': False, 'moisture': 17, 'humidity': 41}
Publish Ok..
{'Animal': [redacted], 'moisture': 84, 'humidity': 16}
Publish Ok..
{'Animal': False, 'moisture': 46, 'humidity': 43}
Publish Ok..
{'Animal': False, 'moisture': 0, 'humidity': 3}
Publish Ok..
{'Animal': False, 'moisture': 73, 'humidity': 68}
Publish Ok..
{'Animal': False, 'moisture': 26, 'humidity': 26}
Publish Ok..
{'Animal': False, 'moisture': 96, 'humidity': 59}
Publish Ok..
```

Ln: 10 Col: 11

WOKWI

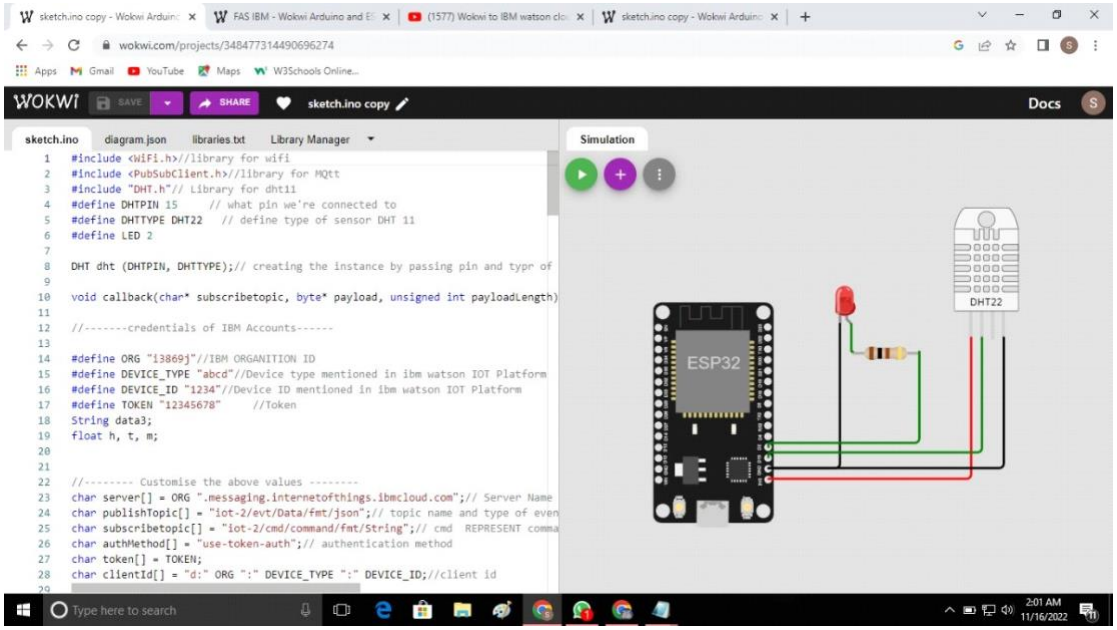
Create your project in the online platform of the WOKWI and execute it using the IBM Credential.

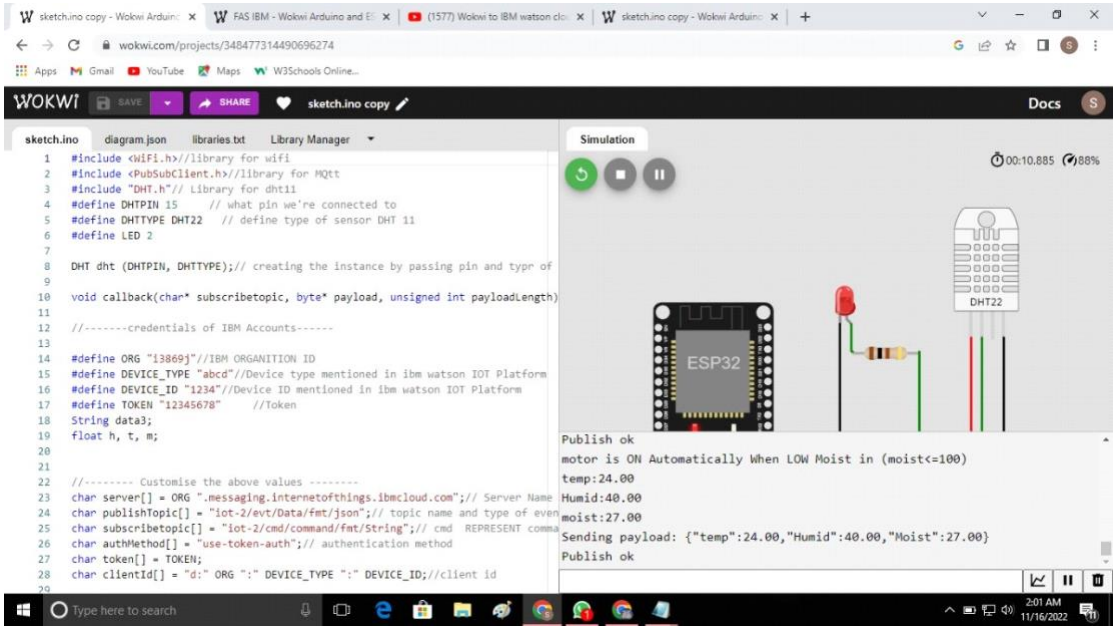
CODE

```
#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 connected


void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);


//-------credentials of IBM Accounts------


#define ORG "rvp7mx"//IBM ORGANITION ID

#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform

#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform

#define TOKEN "12345678"     //Token

String data3; float h, t, m;



//-------- 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 and
format in which data to be send

char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type
AND COMMAND IS TEST OF FORMAT STRING

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 id by passing
parameter like server id,portand wificredential



void setup()// configureing the ESP32
{
  Serial.begin(115200);
dht.begin();
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  randomSeed(analogRead(0));
wificonnect();   mqttconnect();
}

void loop()// Recursive Function
{

  h = dht.readHumidity();   t
= dht.readTemperature();
m = random(100);
  Serial.print("temp:");
  Serial.println(t);
  Serial.print("Humid:");
  Serial.println(h);
```

```
  Serial.print("moist:");
  Serial.println(m);

  PublishData(t, h,m);
delay(1000);   if
(!client.loop()) {
mqttconnect();
  }
{
   if(m<=100){
     Serial.print("motor is ON Automatically When LOW Moist in (moist<=100) ");
Serial.print("\n");
   }
else{
     Serial.println("Moist level is normal");
     }
  }
}
```




sketch.ino copy - Wokwi Arduino
FAS IBM - Wokwi Arduino and E
(1577) Wokwi to IBM watson clo
sketch.ino copy - Wokwi Arduino
wokwi.com/projects/348477314490696274
Apps Gmail YouTube Maps W3Schools Online...
WOKWI SAVE SHARE sketch.ino copy 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
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
//-------credentials of IBM Accounts------
#define ORG "13869j"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //Token
String data3;
float h, t, m;
//-------- 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 even
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT comma
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
Simulation
00:10.885 88%
DHT22
ESP32
Publish ok
motor is ON Automatically When LOW Moist in (moist<=100)
temp:24.00
Humid:40.00
moist:27.00
Sending payload: {"temp":24.00,"Humid":40.00,"Moist":27.00}
Publish ok
Type here to search
2:01 AM
11/16/2022