SPRINT-4

Date	19th NOVEMBER 2022
Team ID	PNT2022TMID35809
Project Name	Project - IOT based Smart Crop Protection for Agriculture

CODE:

```
#include <WiFi.h>

#include <PubSubClient.h>

#include "DHTesp.h"

#include "DHT.h"

#include <ESP32Servo.h>


const int DHT_PIN = 15;

const int servoPin = 18;


Servo servo;


DHTesp dhtSensor;

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

```
#define ORG "ubxjry"
 #define DEVICE_TYPE "nodemcu"
 #define DEVICE_ID "1234"
 #define TOKEN "87654321"
 String data3;
char server[]= ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[]="iot-2/evt/data/fmt/json";
char subscribeTopic[]="iot-2/cmd/command/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);
#define ECHO_PIN 14
#define TRIG_PIN 12
#define BUZZER_PIN 5
#define led 27


int threshold=40;
int trigger;
int pos = 90;
```

```
void setup() {
// put your setup code here, to run once:

servo.attach(servoPin, 500, 2400);

Serial.begin(115200);
dhtSensor.setup(DHT_PIN, DHTesp::DHT22);

Serial.begin(115200);
pinMode(led, OUTPUT);
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT_PULLUP);
pinMode(BUZZER_PIN, OUTPUT);
wificonnect();
mqttconnect();

}
float readDistanceCM() {
digitalWrite(TRIG_PIN, LOW);// Clear the trigger
delayMicroseconds(2);
```

```
digitalWrite(TRIG_PIN, HIGH);// Sets the trigger pin to HIGH
state for 10 microseconds
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
int duration=pulseIn(ECHO_PIN, HIGH);
//Serial.println(duration);
//duration = pulseIn(ECHO_PIN, HIGH);
return duration*0.017;
//Serial.println(duration);
}

void loop() {
TempAndHumidity  data = dhtSensor.getTempAndHumidity();
  Serial.println("Temp: " + String(data.temperature, 2) + "°C");
  Serial.println("Humidity: " + String(data.humidity, 1) + "%");
  Serial.println("---");
  delay(1000);


float distance = readDistanceCM();
float temperature = data.temperature;
float humidity = data.humidity;
float soilmoisture=random(0,100);
```

```
Serial.print("Soil Moisture: ");
Serial.println(soilmoisture);
//Serial.println(distance);
bool isNearby = distance < 100;
digitalWrite(led, isNearby);
Serial.print("Measured distance: ");
Serial.println(distance);

if (distance < 100){
   tone(BUZZER_PIN, 31);
   delay(100); // turn on Piezo Buzzer
}else{
   noTone(BUZZER_PIN);
}  // turn off Piezo Buzzer

PublishData(distance,temperature, humidity, soilmoisture);
if (temperature>50 & humidity<50 & soilmoisture<threshold){


  for (pos = 90; pos >= 0; pos -= 1) {
    servo.write(pos);
    delay(50);
```

```
}
  for (pos = 0; pos <= 90; pos += 1) {
    servo.write(pos);
    delay(50);
  }

}

//PublishData(distance);
delay(1000);
if(!client.loop()){
mqttconnect();
}
//delay(2000);
}


void PublishData(float dist,float temp,float humid,float soilmois){
mqttconnect();
String payload="{\"dist\":";
payload += dist;
payload += "," "\"temp\":";
```

```
payload += temp;
payload += "," "\"humid\":";
payload += humid;
payload += "," "\"soilmois\":";
payload += soilmois;
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 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));
```

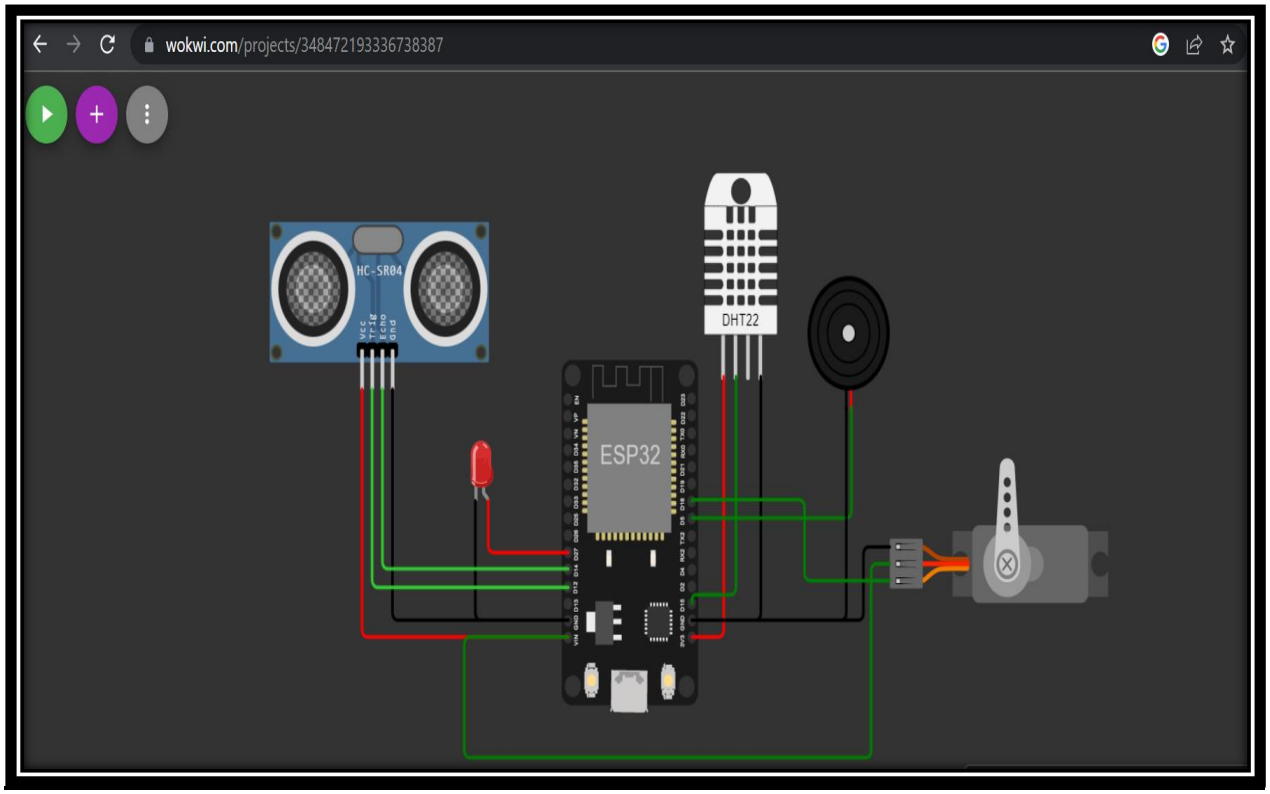
```
Serial.println("subscribe to cmd ok");
}else{
Serial.println("subscribe to cmd failed");}}
void callback(char* subscribeTopic, byte* payload, unsigned int
payloadLength){
Serial.print("callback invoked for topic:");
Serial.println(subscribeTopic);
for(int i=0; i<payloadLength; i++){
data3 += (char)payload[i];}
Serial.println("data:"+ data3);
if(data3=="motor on"){
Serial.println(data3);
for (pos = 90; pos >= 0; pos -= 1) {
   servo.write(pos);
   delay(50);
}
}else{

Serial.println(data3);
for (pos = 0; pos <= 90; pos += 1) {
   servo.write(pos);
   delay(50);
```

}}

data3="";

}

CIRCUIT DESIGN:

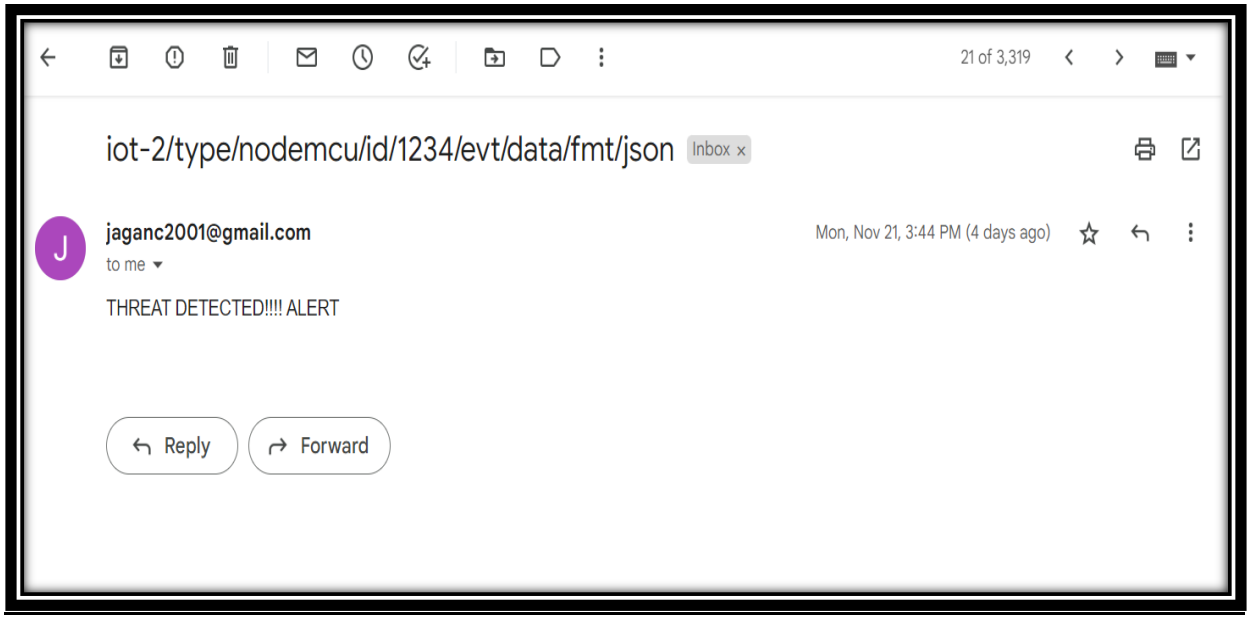


DISPLAYING FIELD DATA:

WEB UI DISPLAY:

APP DISPLAY:

THREAT ALERT INDICATION THROUGH EMAIL:



IBM WATSON DATA COLLECTION: