PROBLEM STATEMENT:

IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

DOMAIN:

INTERNET OF THINGS

ASSIGNMENT 4:

DISTANCE DETECTION USING ULTRASONIC SENSOR

BY

MULLAIYARASI . K - 623519106019

PREMALATHA .V - 623519106025

SARASWATHI . K -623519106036

SRINITHI . B -623519106041

QUESTION-1:

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100 cmssend "alert" to IBM cloud and display in device recent events.

WOKWI LINK:

https://wokwi.com/projects/347936554967106131

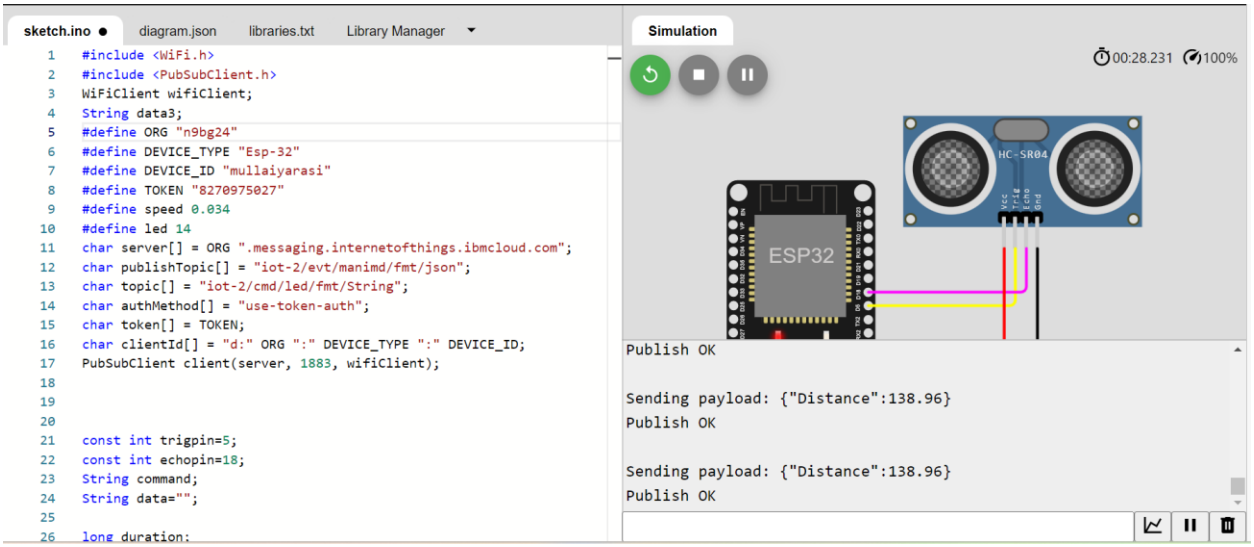
CODE ;

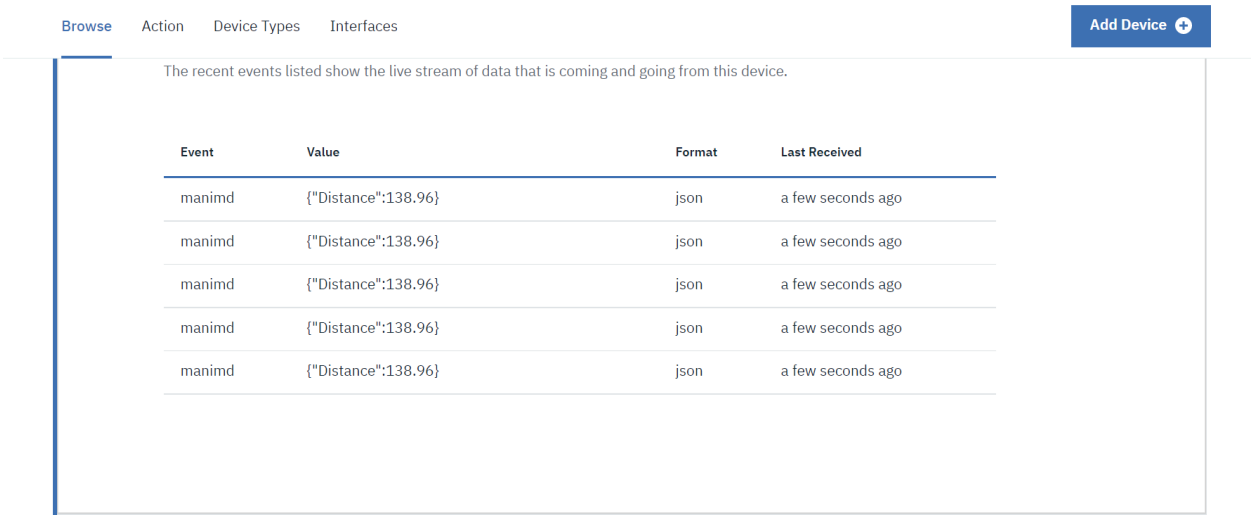
```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "n9bg24"
#define DEVICE_TYPE "Esp-32"
#define DEVICE_ID "mullaiyarasi"
#define TOKEN "8270975027"

#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/manimd/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);



const int trigpin=5;
const int echopin=18;
String command;
String data="";

long duration;
float dist;



void setup()
{
  Serial.begin(115200);
  pinMode(led, OUTPUT);
```

```
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
}

void loop() {
  bool isNearby = dist < 100;
  digitalWrite(led, isNearby);

  publishData();
  delay(500);

  if (!client.loop()) {
    mqttConnect();
  }
}

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

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

void initManagedDevice() {
  if (client.subscribe(topic)) {
    // Serial.println(client.subscribe(topic));
    Serial.println("IBM subscribe to cmd OK");
  } else {
```

```
    Serial.println("subscribe to cmd FAILED");
  }
}
void publishData()
{
  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  dist=duration*speed/2;
  if(dist<100){
    String payload = "{\"Alert Distance\":";
    payload += dist;
    payload += "}";

    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);
    if (client.publish(publishTopic, (char*) payload.c_str())) {
      Serial.println("Publish OK");
    }

  }
    if(dist>100){
    String payload = "{\"Distance\":";
    payload += dist;
    payload += "}";

    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);
     if(client.publish(publishTopic, (char*) payload.c_str())) {
      Serial.println("Publish OK");
    }else {
      Serial.println("Publish FAILED");
    }

  }


  }
```

OUTPUT:


sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "n9bg24"
#define DEVICE_TYPE "Esp-32"
#define DEVICE_ID "mullaiyarasi"
#define TOKEN "8270975027"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/manimd/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
const int trigpin=5;
const int echopin=18;
String command;
String data="";
long duration;
Simulation
00:28.231
100%
HC-SR04
ESP32
Publish OK
Sending payload: {"Distance":138.96}
Publish OK
Sending payload: {"Distance":138.96}
Publish OK


Browse
Action
Device Types
Interfaces
Add Device
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
manimd
{"Distance":138.96}
json
a few seconds ago
manimd
{"Distance":138.96}
json
a few seconds ago
manimd
{"Distance":138.96}
json
a few seconds ago
manimd
{"Distance":138.96}
json
a few seconds ago
manimd
{"Distance":138.96}
json
a few seconds ago