

Assignment-4

| | |
|--------------|--|
| Date | 24 October 2022 |
| Name | Vaishnavi KS |
| Roll Number | 620119106100 |
| Team ID | PNT2022TMID30932 |
| Project Name | IoT Based Smart Crop Protection System for Agriculture |

Question :

Write code and connections in wokwi for ultrasonic sensors. That whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events.

Upload document with wokwi share link and images.

Wokwi:

<https://wokwi.com/projects/346691618263269971>

Code:

```
#include <WiFi.h>
#include <PubSubClient.h>

WiFiClient wifiClient;

WiFiClient wifiClient;

#define ORG "7pe9sv"
#define DEVICE_TYPE "ESP32_Controller"
#define DEVICE_ID "BME280_Sensor"
#define TOKEN "SoqTMunRtwqr9LFDp7"
#define speed 0.034

char server[] = ORG".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/event_1/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=5;
const int echopin=18;
String command;
String data="";
long duration;
float dist;
void setup()
{
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
```

```

pinMode(echopin, INPUT);
wifiConnect();
mqttConnect();
}
void loop() {
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("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");

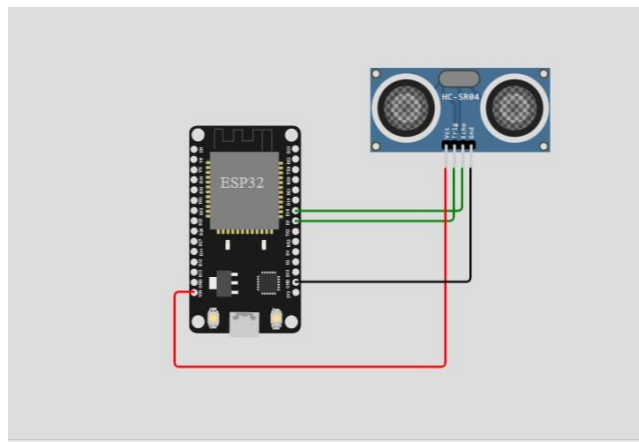
```

```

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

```

Diagram:



Wokwi Output:

WOKWI

sketch.ino

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4 WiFiClient wificlient;
5
6 #define ORG "7pe9sv"
7 #define DEVICE_TYPE "ESP32_Controller"
8 #define DEVICE_ID "BME280_Sensor"
9 #define TOKEN "SoqTMunRtuqr9LFDp7"
10 #define speed 0.034
11
12
13 char server[] = ORG".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/event_1/fmt/json";
15 char topic[] = "iot-2/cmd/home/fmt/String";
16 char authMethod[] = "use-token-auth";
17 char token[] = TOKEN;
18 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
19 PubSubClient client(server, 1883, wificlient);
20 void publishData();
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25 long duration;
26 float dist;
27 void setup()
28 {

```

Simulation

07:31.658 92%

Publish OK

Sending payload: {"Alert distance":99.98}

Publish OK

Sending payload: {"Alert distance":99.98}

Publish OK

Show all

Assignment_4.pdf Sprint 1 Report.pdf

11:35 AM 12.11.2022

IBM cloud output:

The screenshot displays the IBM Watson IoT Platform interface. At the top, the navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar labeled 'Search by Device ID' is present. The main content area shows a table of devices. The selected device is 'BME280_Sensor', which is 'Connected' and has a device type of 'ESP32_Controller'. Below the device header, there are tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a list of events with columns for 'Event', 'Value', 'Format', and 'Last Received'. The events listed are 'event_1' with a value of '{"Alert distance":99.99}' or '{"Alert distance":99.98}', in 'json' format, received 'a few seconds ago'. A message at the bottom of the events list states '0 Simulations running'.

| Device ID | Status | Device Type | Class ID | Date Added |
|---------------|-----------|------------------|----------|------------------|
| BME280_Sensor | Connected | ESP32_Controller | Device | 8 Nov 2022 18:58 |

| Event | Value | Format | Last Received |
|---------|--------------------------|--------|-------------------|
| event_1 | {"Alert distance":99.99} | json | a few seconds ago |
| event_1 | {"Alert distance":99.99} | json | a few seconds ago |
| event_1 | {"Alert distance":99.98} | | |
| event_1 | {"Alert distance":99.98} | | |