

Assignment -4

Assignment Date	05 November 2022
Team ID	PNT2022TMID30928
Project Name	IoT Based Smart Crop Protection System for Agriculture

QUESTION:

Write code and connections in wokwi for ultrasonic sensor. 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 of ibm cloud.

CODE:

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

WiFiClient wifiClient;

#define ORG "x3pm5f"//IBM ORGANITION ID
#define DEVICE_TYPE "nandhu"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "2010"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "sB+LxFDqRyHa!t_-YV" //Token
#define speed 0.034

char server[] = ORG".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Distance/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();
}
}

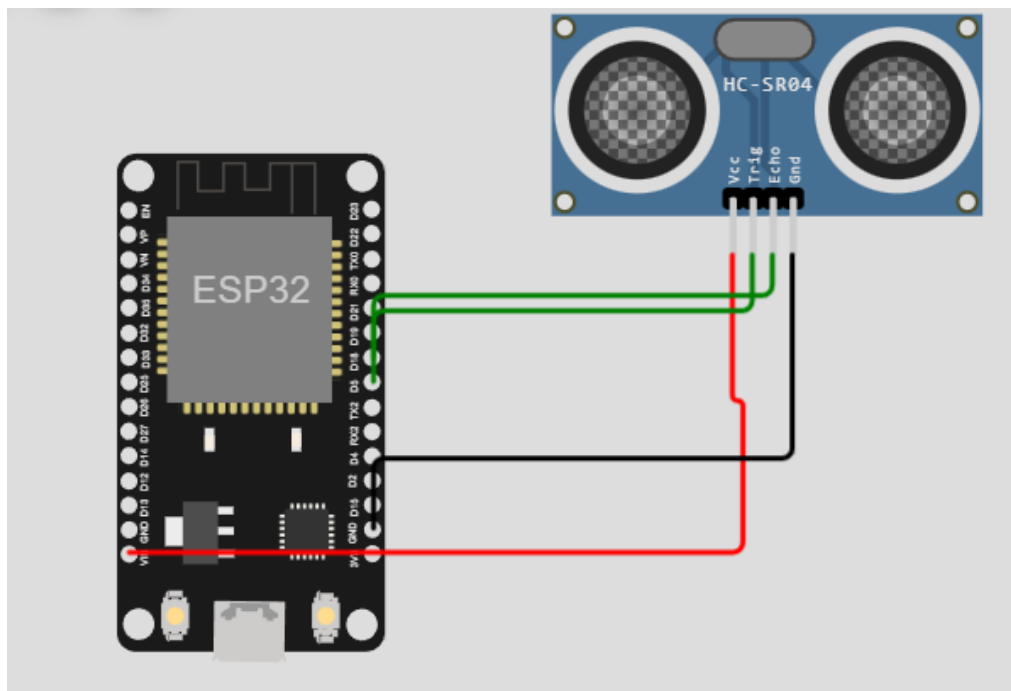
void initManagedDevice() {
if (client.subscribe(topic)) {
// Serial.println(client.subscribe(topic));
Serial.println("subscribe to cmd OK");
}
else {
Serial.println("subscribe to cmd FAILED");
}
}

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");  
}  
}  
}  
}
```

CONNECTIONS:



WOKWI LINK:

<https://wokwi.com/projects/348308707121037908>

OUTPUT:

The screenshot shows the Wokwi IDE interface. On the left, the sketch code is displayed:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4 WiFiClient wifiClient;
5
6 #define ORG "x3pm5f" // IBM ORGANIZATION ID
7 #define DEVICE_TYPE "nandhu" // Device type mentioned in IBM Watson IoT Platform
8 #define DEVICE_ID "2010" // Device ID mentioned in IBM Watson IoT Platform
9 #define TOKEN "sB+LxFDqRyHalt_-YV" // Token
10 #define speed 0.034
11
12
13 char server[] = ORG".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/Distance/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 {
29   Serial.begin(115200);
30   pinMode(trigpin, OUTPUT);
31 }
```

The simulation window on the right shows a virtual circuit with an ESP32 microcontroller and an HC-SR04 ultrasonic sensor. The output log displays the following messages:

```
Publish OK
Sending payload: {"Alert distance":99.98}
Publish OK
Sending payload: {"Alert distance":99.98}
Publish OK
```

The screenshot shows the IBM Watson IoT Platform dashboard. The device 'nandhu' (ID: 2010) is shown as 'Connected'. The 'Recent Events' tab is selected, displaying a table of live data events:

Event	Value	Format	Last Received
Distance	{"Alert distance":99.98}	json	a few seconds ago
Distance	{"Alert distance":99.98}	json	a few seconds ago
Distance	{"Alert distance":99.98}	json	a few seconds ago
Distance	{"Alert distance":99.98}	json	a few seconds ago
Distance	{"Alert distance":99.98}	json	a few seconds ago

At the bottom of the dashboard, it indicates '0 Simulations running'.