

ASSIGNMENT4

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud.

Wokwi link: <https://wokwi.com/projects/347203268647060051>

CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "j8ux59"
#define DEVICE_TYPE "Ranjani"
#define DEVICE_ID "Ranjani76"
#define TOKEN "Nisha80*"
#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/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;
int 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(1000);
}
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)
{
DynamicJsonDocument doc(1024); String payload;
doc["Distance Alert:"]=dist;
serializeJson(doc, payload);
delay(3000);
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:

The screenshot displays the Wokwi online Arduino IDE interface. On the left, the 'sketch.ino' file contains the following code:

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4 WiFiClient wifiClient;
5 #define ORG "j8ux59"
6 #define DEVICE_TYPE "Ranjani"
7 #define DEVICE_ID "Ranjani176"
8 #define TOKEN "Nisha80*"
9 #define speed 0.034
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/Data/fmt/json";
12 char topic[] = "iot-2/cmd/home/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
16 PubSubClient client(server, 1883, wifiClient);
17 void publishData();
18 const int trigpin=5;
19 const int echopin=18;
20 String command;
21 String data="";
22 long duration;
23 int dist;
24 void setup()
25 {
26   Serial.begin(115200);

```

On the right, the 'Simulation' tab shows a visual representation of the hardware: an ESP32 microcontroller board connected to an HC-SR04 ultrasonic sensor. Below the simulation, the output console shows the following messages:

```

Publish OK
Sending payload: {"Distance Alert":87}
Publish OK
Sending payload: {"Distance Alert":87}
Publish OK

```

The bottom of the image shows the Windows taskbar with the time 21:41 and date 01-11-2022.

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Service Details - IBM Cloud', 'IBM Watson IoT Platform', and 'Wokwi - Online Arduino and ESP32 Simulator'. The main header shows the 'IBM Watson IoT Platform' logo and a user profile with ID: j8ux59. The left sidebar contains icons for various functions. The main content area is titled 'Browse' and shows a search bar with the text 'Search by Device ID'. Below this, a table lists devices. The selected device is 'Ranjani76', which is 'Disconnected'. The 'Recent Events' tab is active, showing a list of events. The events table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The events are as follows:

Event	Value	Format	Last Received
Data	{"Distance Alert":"87"}	json	a few seconds ago
Data	{"Distance Alert":"87"}	json	a few seconds ago
Data	{"Distance Alert":"80"}	json	a few seconds ago
Data	{"Distance Alert":"29"}	json	a few seconds ago
Data	{"Distance Alert":"29"}	json	a few seconds ago

A tooltip at the bottom right indicates '0 Simulations running'.