

Assignment - 4

Assignment Date	17 October 2022
Student Name	VINITHA SELVI A
Student Roll Number	49621911113
Maximum Marks	2 Marks

Question-1:

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

CODE 1 :

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribtopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "prcaq4"//IBM ORGANITION ID
#define DEVICE_TYPE "IOT"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "15072002"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "1911113abcdefgh" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribtopic[] = "iot-2/cmd/test/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); const
int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  wificonnect(); mqttconnect();
}
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW); duration
= pulseIn(echoPin, HIGH);
distance = duration * SOUND_SPEED/2;
```

```

Serial.print("Distance (cm): ");
Serial.println(distance);
if(distance<100)
{
Serial.println("ALERT!!");
delay(1000);
PublishData(distance);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
delay(1000);
}
void PublishData(float dist) {
mqttconnect();
String payload = "{\"Distance\".";
payload += dist;
payload += ",\"ALERT!!\":\"\\"Distance less than 100cms\\\""; 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
client 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++) {
//Serial.print((char)payload[i]);
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
data3="";
}
```

Wokwi Link :

<https://wokwi.com/projects/347021585567187540>

Output and Simulation :

The screenshot displays the Wokwi web interface for an Arduino simulation. On the left, the `sketch.ino` file is open, showing an Arduino sketch that uses the `PubSubClient` library to connect to the IBM Watson IoT Platform. The sketch defines an ESP32 device with specific credentials and a callback function to handle incoming data. On the right, the simulation window shows an ESP32 board connected to an HC-SR04 ultrasonic sensor. The sensor's distance is set to 90cm. Below the sensor, the simulation output shows the JSON payload being sent: `{"Distance":89.98,"ALERT!!":"Distance less than 100cms"}`. The output also shows the distance reading: `Distance (cm): 89.98` and the alert message: `ALERT!!`.

Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

The screenshot shows the IBM Watson IoT Platform dashboard. The device `15072002` is listed as `Connected`. The `Recent Events` tab is selected, showing a list of events. The events are as follows:

Event	Value	Format	Last Received
Data	<code>{"Distance":89.98,"ALERT!!":"Distance less than ...</code>	json	a few seconds ago
Data	<code>{"Distance":89.98,"ALERT!!":"Distance less than ...</code>	json	a few seconds ago