

Assignment - 4

Assignment Date	17 October 2022
Student Name	ABISWETHA S
Student Roll Number	49621911002
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 "viqkt9"//IBM ORGANITION ID
#define DEVICE_TYPE "IOT_FIRE"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "28062002"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "1911002abi" //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 IDE interface. On the left, the 'sketch.ino' file is open, showing a C++ sketch for an ESP32. The sketch includes libraries for WiFi, PubSubClient, and MQTT. It defines an IOT_Fire device type and connects to the IBM Watson IoT Platform. The main loop checks the distance from an ultrasonic sensor (HC-SR04) and sends an alert message to the IoT cloud if the distance is less than 100 cm.

On the right, the 'Simulation' window shows a visual representation of the ESP32 and the HC-SR04 sensor. Below the simulation, a console window displays the following output:

```
Connecting to ...
WiFi connected
IP address:
10.10.0.2
Reconnecting client to viokt9.messaging.internetofthings.ibmcloud.com
.....
```

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 'Browse' tab is selected, displaying a list of devices. The device '261021' is highlighted, and its 'Recent Events' are shown in a table.

Event	Value	Format	Last Received
Data	{"Distance":89.98,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":89.96,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":89.98,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":89.98,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":89.98,"ALERT!":"Distance less than ...	json	a few seconds ago