Assignment -4

Assignment Date	19 October 2022
Student Name	M. Deepak
Student Roll Number	511319104013
Maximum Marks	2 Marks

Question-1:

Write code and connections in wokwi for ultrasonic sensor.

Whenever distance is less than 100cms send “alert” to IBM cloud and display in device recent events.

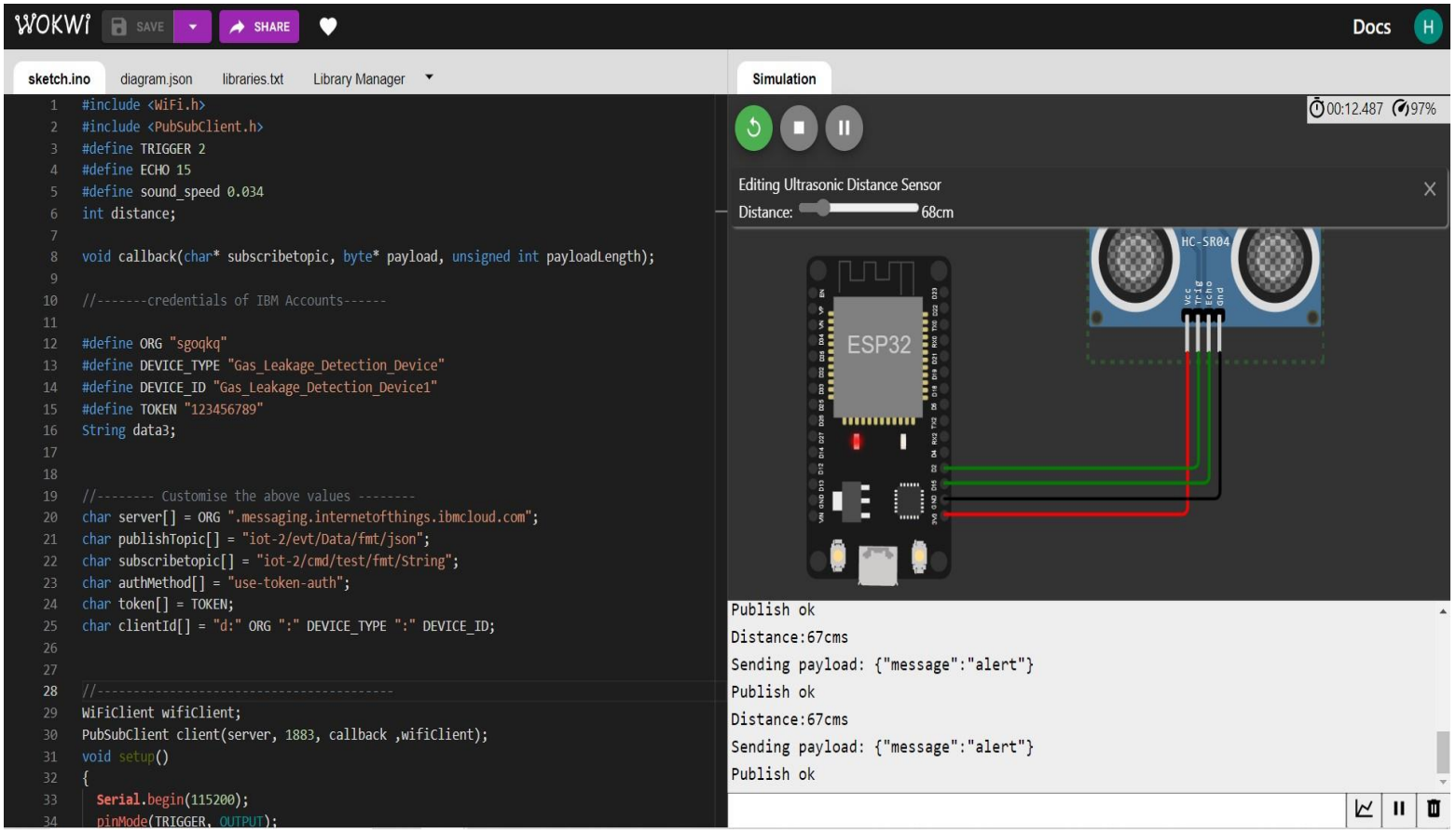
Wokwi Project Link: https://wokwi.com/projects/346235279031403092

```
#include <WiFi.h>
#include <PubSubClient.h>
#define TRIGGER 2
#define ECHO 15
#define sound_speed 0.034
int distance;
 void callback(char* subscribetopic, byte* payload, unsigned
int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "sgoqkq"
#define DEVICE_TYPE "Gas_Leakage_Detection_Device"
#define DEVICE_ID "Gas_Leakage_Detection_Device1"
#define TOKEN "123456789"
String data3;


//-------- Customise the above values --------
char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; char
publishTopic[] = "iot-2/evt/Data/fmt/json"; char
subscribetopic[] = "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); void
setup()
{
  Serial.begin(115200);
pinMode(TRIGGER, OUTPUT);
pinMode(ECHO, INPUT);
delay(10);
Serial.println();
wificonnect();
mqttconnect();
```

```
}  void
loop()
{    digitalWrite(TRIGGER,
HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER, LOW);
   int duration=pulseIn(ECHO,HIGH);
distance=(duration*sound_speed)/2;
Serial.print("Distance:");
  Serial.print(distance);
Serial.println("cms");   if(distance<100){
    PublishData(distance);
  }   delay(1000);   if
(!client.loop()) {
mqttconnect();
  }
}



/*.....................................retrieving to
Cloud.............................*/
 void PublishData(int d)
{   mqttconnect();

  String payload = "{\"message\":\"alert\"}";

  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++) {
data3 += (char)payload[i];
  }

  Serial.println("data: "+ data3);

data3="";


}
```

Wokwi Platform Coding and Circuit Design




```
#include <WiFi.h>
#include <PubSubClient.h>
#define TRIGGER 2
#define ECHO 15
#define sound_speed 0.034
int distance;

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "sgoqkq"
#define DEVICE_TYPE "Gas_Leakage_Detection_Device"
#define DEVICE_ID "Gas_Leakage_Detection_Device1"
#define TOKEN "123456789"
String data3;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "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);
void setup()
{
  Serial.begin(115200);
  pinMode(TRIGGER, OUTPUT);
```

IBM IoT Platform Device Recent Events