Assignment – 4
Distance Detection using Ultrasonic Sensor

Name	Rashmi
Team ID	PNT2022TMID14815
Roll Number	7179KCTKCTKCTKCTKCTKCTKCT19BEC025

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.

Wokwi Link:

https://wokwi.com/projects/346942115517825620

Code:

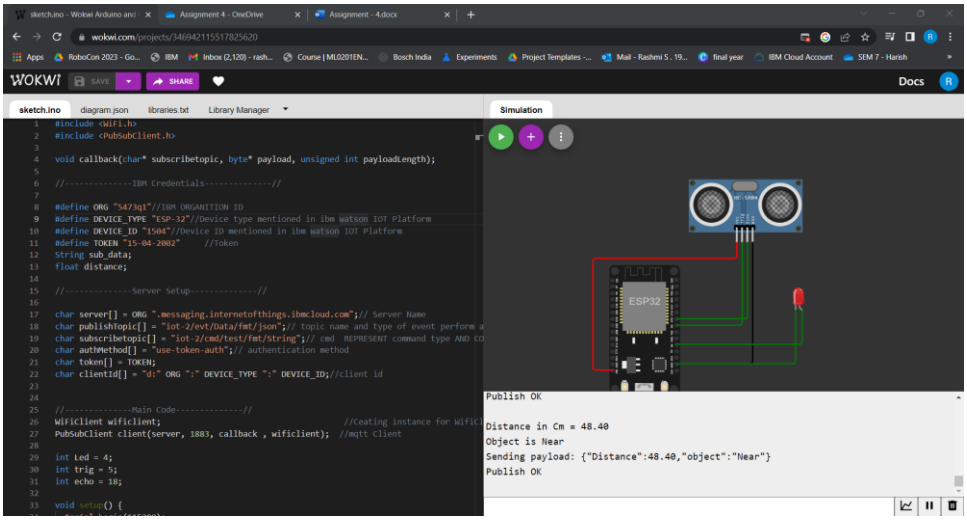
```
#include <WiFi.h>

#include <PubSubClient.h>

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

//-------------IBM Credentials-------------//

#define ORG "5473q1"//IBM ORGANITION ID

#define DEVICE_TYPE "ESP-32"//Device type mentioned in ibm watson IOT Platform

#define DEVICE_ID "1504"//Device ID mentioned in ibm watson IOT Platform

#define TOKEN "15-04-2002"     //Token

String sub_data;

float distance;

//-------------Server Setup-------------//

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name

char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send

char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING

char authMethod[] = "use-token-auth";// authentication method

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-------------Main Code-------------//

WiFiClient wificlient;
//Ceating instance for WifiClient

PubSubClient client(server, 1883, callback , wificlient);  //mqtt Client

int Led = 4;

int trig = 5;

int echo = 18;

void setup() {

  Serial.begin(115200);

  pinMode(trig, OUTPUT);

  pinMode(Led, OUTPUT);

  pinMode(echo, OUTPUT);

  delay(10);

  wificonnect();

  mqttconnect();

}

void loop() {

  digitalWrite(trig, LOW);

  digitalWrite(trig, HIGH);

  delayMicroseconds(10);

  digitalWrite(trig, LOW);

  float duration = pulseIn(echo, HIGH);
```

```
  float distance = (duration * 0.0343) / 2;
  Serial.print("Distance in Cm = ");
  Serial.println(distance);
  Publish_Data(distance);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}
void Publish_Data(float dist) {
  mqttconnect();  //Connect to Server
  /* Creating the String in JSON format to send to the Cloud
       according to the diatance from the Ultrasonic Sensor*/
  String object;
  if (dist < 100) {
    digitalWrite(Led, HIGH);
    Serial.println("Object is Near");
    object = "Near";
  }
  else {
    digitalWrite(Led, LOW);
    Serial.println("No Object Found");
    object = "No Object";
  }
  String payload = "{\"Distance\":";
  payload += dist;
  payload += "," "\"object\":\"";
  payload += object;
  payload += "\"}";
  Serial.print("Sending payload: ");
  Serial.println(payload);
  //Publish payload Message
  if (client.publish(publishTopic , (char*) payload.c_str())) {
    Serial.println("Publish OK");
  } else {
    Serial.println("Publish Failed");
  }
  Serial.println("");
}
//--------------User Fuctions--------------//
//Connect to Mqtt
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() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
```

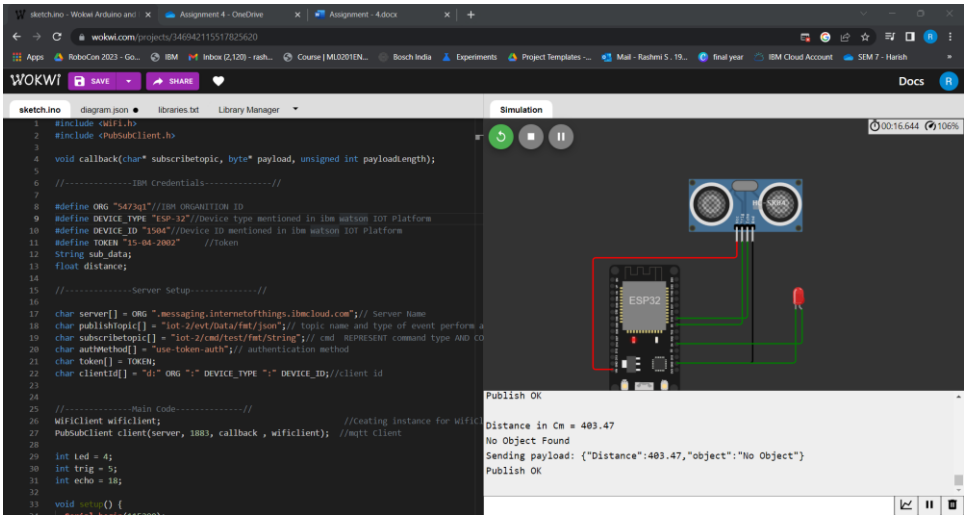
```
}
void initManagedDevice() {
 if (client.subscribe(subscribetopic)) {
   Serial.println((subscribetopic));
   Serial.println("subscribe to cmd OK");
 } else {
   Serial.println("subscribe to cmd FAILED");
 }
}
void callback(char* subscribetopic, byte* payload,
unsigned int payloadLength)
{
 Serial.print("Callback invoked for topic: ");
 Serial.println(subscribetopic);
 for (int i = 0; i < payloadLength; i++) {
  //Serial.print((char)payload[i]);
  sub_data += (char)payload[i];
 }
 Serial.println("data: "+ sub_data);
sub_data="";
}
```

Output:

Object is Near:



Object is Far:



IBM Watson IoT Platform: