Assignment
4

Assignment Date	28 October 2022
Student Name	Jasith S
Student Roll Number	722819106033
Maximum Marks	2 Marks

QUESTION:

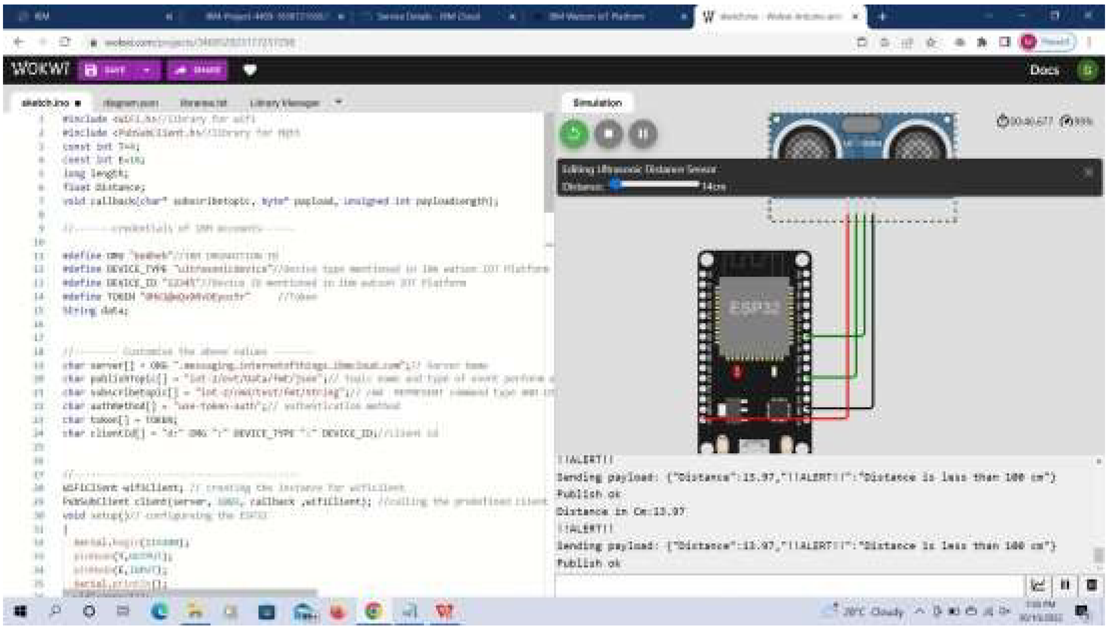
Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

SOLUTION:

```
#include <WiFi.h>//library for wifi #include
<PubSubClient.h>//library for MQtt const
int T=4; const int E=18; long length; float
Distance;
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

#define ORG "6e8heb"
#define DEVICE_TYPE
"ultrasonicdevice" #define DEVICE_ID
"12345"
 #define TOKEN
"0Md1@wQs0RvOEyoz5r" String data;


 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_


WiFiClient wifiClient;
PubSubClient client(server, 1883, callback
 ,wifiClient); void setup()
{
 Serial.begin(115200);
 pinMode(T,OUTPUT);
 pinMode(E,INPUT);
 Serial.println();
 wificonnect();
 mqttconnect();
}

void loop()
{ digitalWrite(T,LOW);
 delay(1000);
 digitalWrite(T,HIGH);
 delay(1000);
 digitalWrite(T,LOW);
 length=pulseIn(E,HI
 G H);
 Distance=length*(0.
 03 4/2);
  Serial.print("Distance in
 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 is less than 100
 cm\""; 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++)
    { data += (char)payload[i];
  }

  Serial.println("data: "+
data); data=""; }
```

OUTPUT



```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for mqtt
const int T=4;
const int E=15;
long length;
float distance;
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-----credentials of IBM accounts------

#define ORG "bsdheb"//IBM ORGANITION ID
#define DEVICE_TYPE "ultrasonicdevice"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "[illegible]"     //Token
String data;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform a
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND CO
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client
void setup()// configureing the ESP32
{
  Serial.begin(115200);
  pinMode(T,OUTPUT);
  pinMode(E,INPUT);
  Serial.println();
```

```
!!ALERT!!
Sending payload: {"Distance":13.97,"!!ALERT!!":"Distance is less than 100 cm"}
Publish ok
Distance in Cm:13.97
!!ALERT!!
Sending payload: {"Distance":13.97,"!!ALERT!!":"Distance is less than 100 cm"}
Publish ok
```

Wokwi simulation link:

https://wokwi.com/projects/346952923117257298