Assignment -4

Assignment Date	31 oct 2022
Student Name	P.Abinaya
Student Roll Number	811019106004
Maximum Marks	2 Marks

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.

Coding:

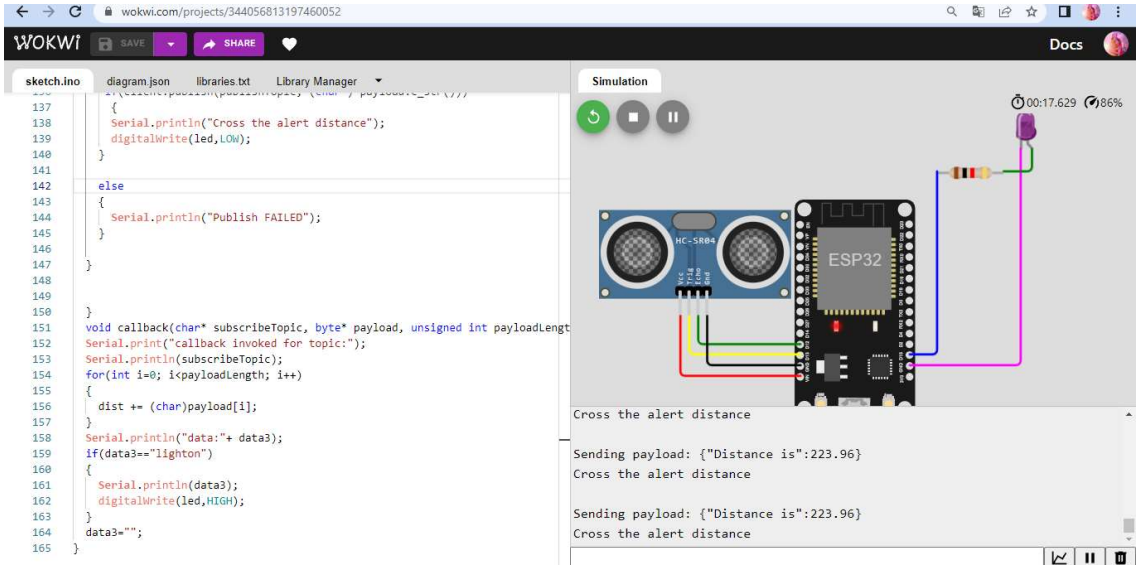
```
#include <WiFi.h>
#include <PubSubClient.h>

WiFiClient wifiClient;
String data3;
#define speed 0.034
#define led 15
const int trigpin=13;
const int echopin=12;
String command;
String data="";

long duration;
float dist;
//-------credentials of IBM Accounts------
#define ORG "szro21"
#define DEVICE_TYPE "ammudevicetype"
#define DEVICE_ID "123deviceid"
#define TOKEN "0987654321"

//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/command/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
```

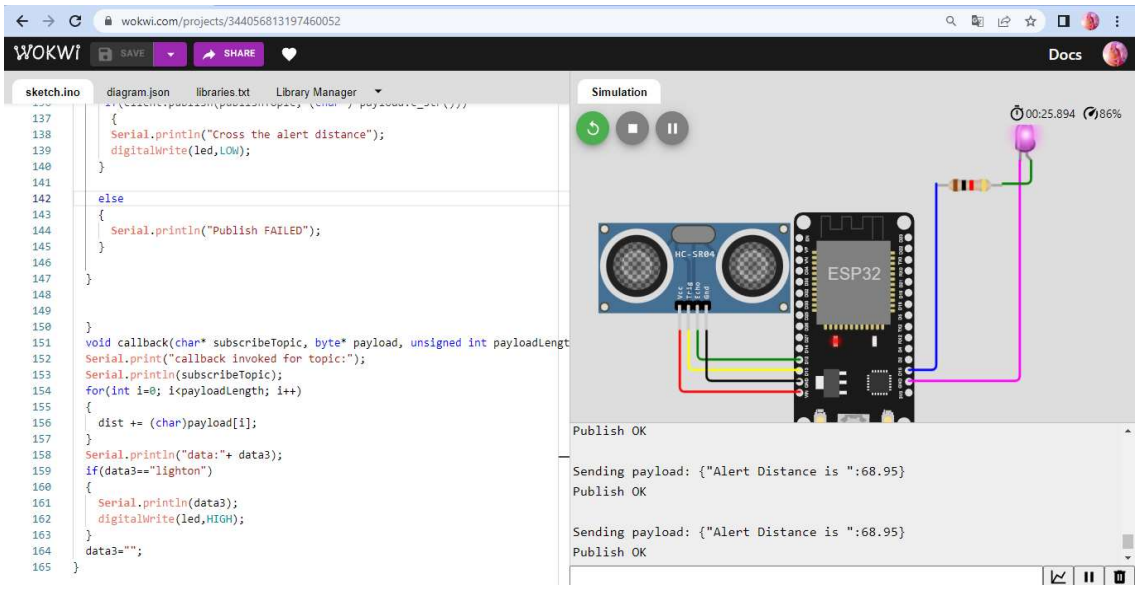
```
void setup()
{
  Serial.begin(115200);
  pinMode(led, OUTPUT);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
}

void loop()
{
  bool Nearby = dist < 100;
  digitalWrite(led, Nearby);

  publishData();
  delay(500);

  if (!client.loop())
   {
    mqttConnect();
  }
}

void wifiConnect()
 {
  Serial.print("Connecting to ");
   Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());
}

void mqttConnect()
{
  if (!client.connected())
   {
    Serial.print("Reconnecting MQTT client to ");
    Serial.println(server);
    while (!client.connect(clientId, authMethod, token))
     {
```

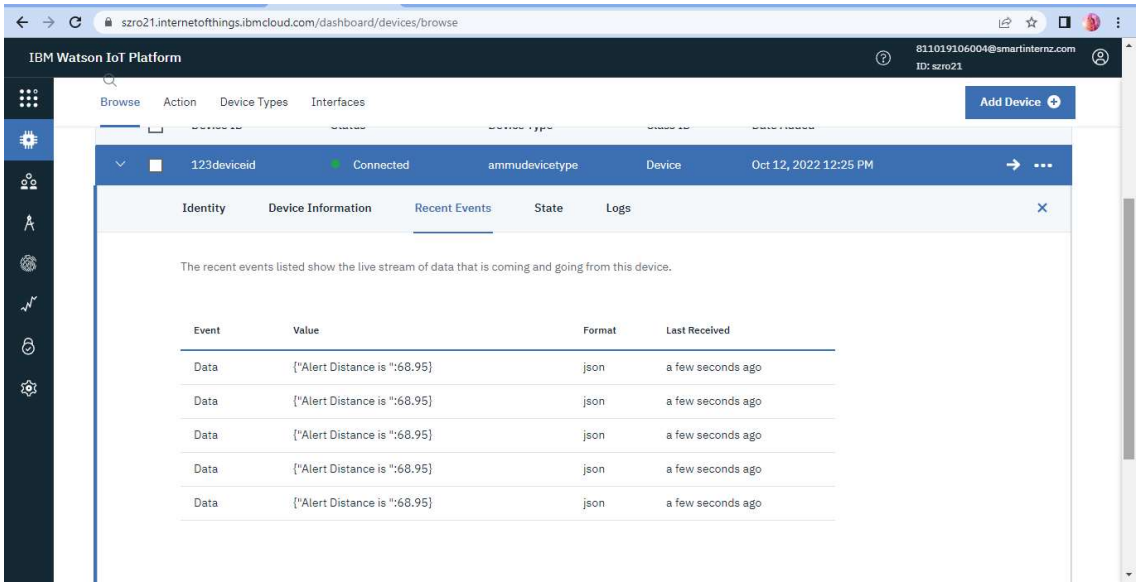
```
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void initManagedDevice()
 {
  if (client.subscribe(topic))
  {
    // Serial.println(client.subscribe(topic));
    Serial.println("IBM subscribe to cmd OK");
  }
   else
    {
    Serial.println("subscribe to cmd FAILED");
  }
}
/*.....................................retrieving to
Cloud.............................*/
void publishData()
{
  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  dist=duration*speed/2;
  if(dist<100)
  {
    String payload = "{\"Alert Distance is \":";
    payload += dist;
    payload += "}";

    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);
    if (client.publish(publishTopic, (char*) payload.c_str()))
    {
      Serial.println("Publish OK");// if it sucessfully upload data on the
cloud then it will print publish ok in Serial monitor or else it will print
publish failed
      digitalWrite(led,HIGH);
    }
  }
     if(dist>100)
```

```
    {
    String payload = "{\"Distance is\":";
    payload += dist;
    payload += "}";

    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);

     if(client.publish(publishTopic, (char*) payload.c_str()))
      {
      Serial.println("Cross the alert distance");
      digitalWrite(led,LOW);
    }

    else
    {
      Serial.println("Publish 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++)
  {
    dist += (char)payload[i];
  }
  Serial.println("data:"+ data3);
  if(data3=="lighton")
  {
    Serial.println(data3);
    digitalWrite(led,HIGH);
  }
  data3="";
}
```

Connection:


wokwi.com/projects/344056813197460052
WOKWi
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:17.629
86%
HC-SR04
ESP32
{
Serial.println("Cross the alert distance");
digitalWrite(led,LOW);
}
else
{
Serial.println("Publish FAILED");
}
}
}
void callback(char* subscribeTopic, byte* payload, unsigned int payloadLengt
Serial.print("callback invoked for topic:");
Serial.println(subscribeTopic);
for(int i=0; i<payloadLength; i++)
{
dist += (char)payload[i];
}
Serial.println("data:"+ data3);
if(data3=="lighton")
{
Serial.println(data3);
digitalWrite(led,HIGH);
}
data3="";
}
Cross the alert distance
Sending payload: {"Distance is":223.96}
Cross the alert distance
Sending payload: {"Distance is":223.96}
Cross the alert distance


wokwi.com/projects/344056813197460052
WOKWi
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:25.894
86%
HC-SR04
ESP32
{
Serial.println("Cross the alert distance");
digitalWrite(led,LOW);
}
else
{
Serial.println("Publish FAILED");
}
}
}
void callback(char* subscribeTopic, byte* payload, unsigned int payloadLengt
Serial.print("callback invoked for topic:");
Serial.println(subscribeTopic);
for(int i=0; i<payloadLength; i++)
{
dist += (char)payload[i];
}
Serial.println("data:"+ data3);
if(data3=="lighton")
{
Serial.println(data3);
digitalWrite(led,HIGH);
}
data3="";
}
Publish OK
Sending payload: {"Alert Distance is ":68.95}
Publish OK
Sending payload: {"Alert Distance is ":68.95}
Publish OK



Wokwi link:

https://wokwi.com/projects/344056813197460052