

Assignment -4

Question-1:

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.

CODE:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT

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

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

#define ORG "f59trs" //IBM ORGANITION ID
#define DEVICE_TYPE "ultrasonicsensor" //Device type mentioned in ibm watson
IOT Platform
#define DEVICE_ID "distancedetection" //Device ID mentioned in ibm watson
IOT Platform
#define TOKEN "AIGMGaaF01nawa1QA3" //Token
String data3;
float dist;

//----- 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 performed
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; //clientid

// -
WiFiClient wifiClient; // creating the instance for wifi client
PubSubClient client(server, 1883, callback, wifiClient);
//calling the predefined client id by passing parameter like server id, port and
wificredential

int LED = 4;
int trig = 5;
int echo = 18;
void setup()
{
  Serial.begin(115200);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(LED, OUTPUT);
  delay(10);
  wifiConnect();
  mqttConnect();
}
```

```

void loop()// Recursive Function
{

digitalWrite(trig,LOW);
digitalWrite(trig,HIGH);
delayMicroseconds(10);
digitalWrite(trig,LOW);
float dur = pulseIn(echo,HIGH);float dist
= (dur * 0.0343)/2; Serial print
("Distancein cm");Serial println(dist);


PublishData(dist);
delay(1000);
if (!client.loop()) {
    mqttconnect();
}
}

/*.....retrieving to
Cloud. ....*/

void PublishData(float dist) { mqttconnect();//function call for
connecting to ibm
/*
    creating the String in in form JSon to update the data toibm
cloud */
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";
}

String payload = "{\"distance\":";payload +=
dist;
payload += ", \"object\":\":";payload +=
object;
payload += "\"}";

Serial print("Sending payload: ");
Serial println(payload);

```

```

        if (client.publish(publishTopic, (char*) payload.c_str())) {
            Serial.println("Publish ok");// if it sucessfully upload dataon the cloud then it will
            print publish ok in Serial monitor or else it will print publish failed } 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() //function defination for wificonnect
    {
        Serial.println(); Serial.print("Connecting
            to ");

        WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentialsto 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
    intpayloadLength) {

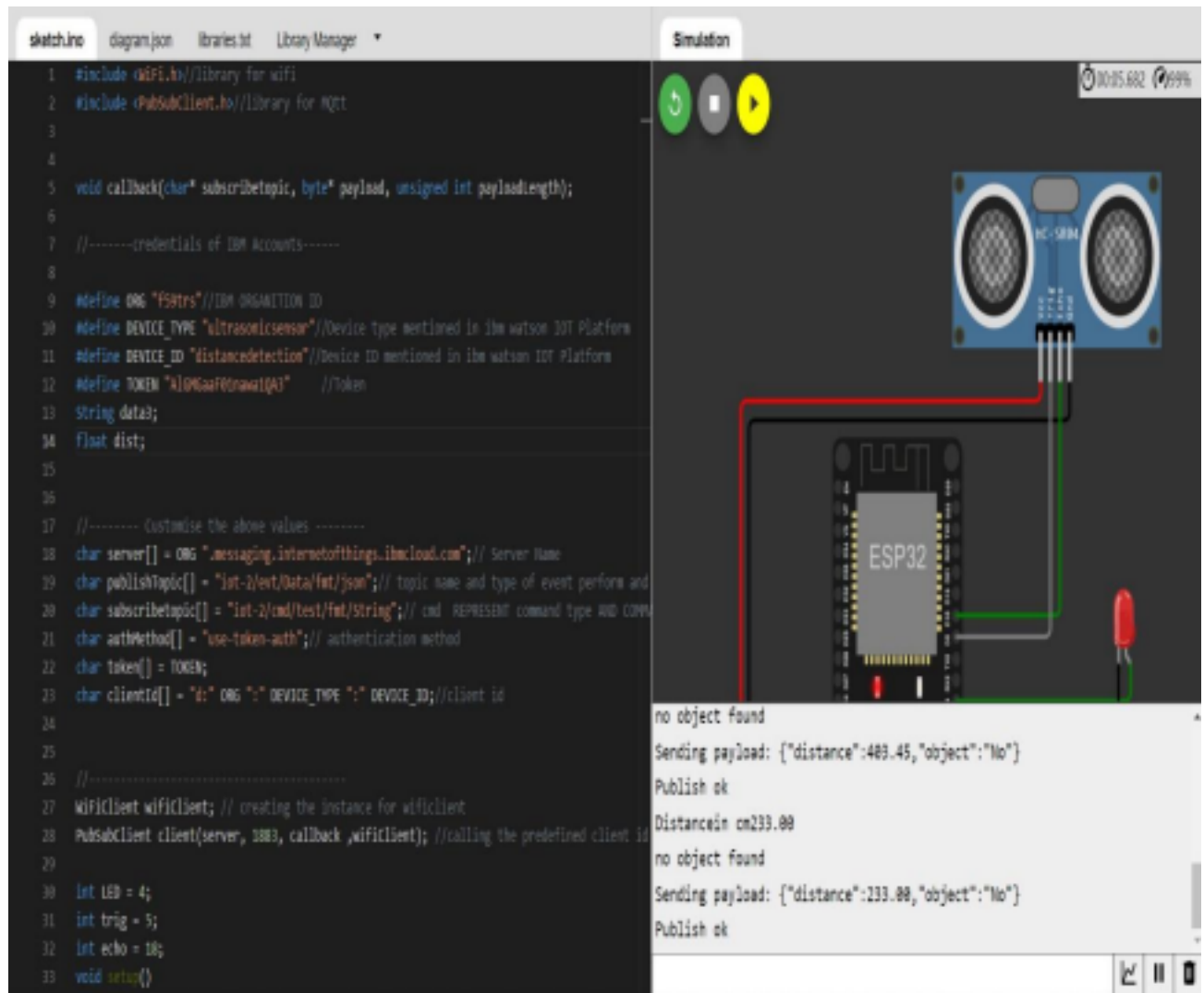
        Serial.print("callback invoked for topic: ");
        Serial.println(subscribetopic);
        for (int i = 0; i < payloadLength; i++) {

```

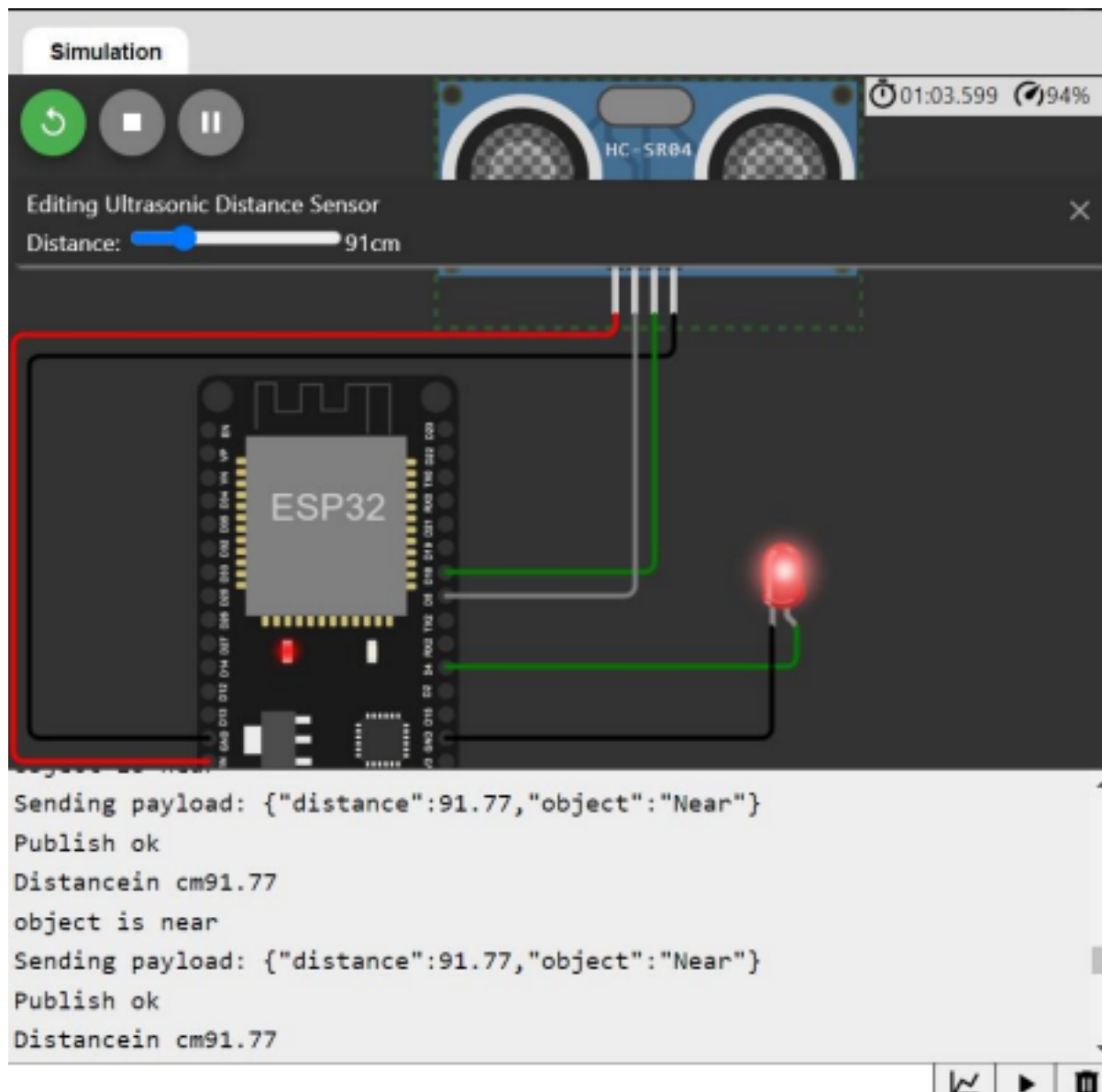
```
        //Serial.print((char)payload[i]);data3 +=  
        (char)payload[i];  
    }  
  
    // Serial.println("data: " + data3);  
    // if(data3=="Near")  
    // {  
    // Serial.println(data3);  
    // digitalWrite(LED,HIGH);  
  
    // }  
  
    // else  
    // {  
    // Serial.println(data3);  
    // digitalWrite(LED,LOW);  
  
    // }  
    data3="";  
}
```

OUTPUT:

When object is not near to the ultrasonic sensor



Data sent to the IBM cloud device when the object is far



Data sent to the IBM cloud device when the object is near

The screenshot displays a web application interface for managing devices. At the top, there are navigation tabs: 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is located below these tabs. On the right, there is a button labeled 'Add Device' with a plus icon.

The main content area shows a table of devices. The table has columns: 'Device ID', 'Status', 'Device Type', 'Class ID', 'Date Added', and 'Descriptive Location'. One device is listed: 'distancetodection' with status 'Connected', device type 'ultrasonicsensor', class ID 'Device', and date added 'Oct 19, 2022 11:56 AM'. To the right of this row is a plus icon and three dots.

Below the device list, there is a section titled 'Recent Events' with a close button 'X'. This section contains a message: 'The recent events listed show the live stream of data that is coming and going from this device.'

Below the message is a table of recent events with columns: 'Event', 'Value', 'Format', and 'Last Received'. The table contains five rows of data:

Event	Value	Format	Last Received
Data	{"distance":91.77,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.75,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.77,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.79,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.8,"object":"Near"}	json	a few seconds ago

At the bottom right of the interface, there is a status indicator: '0 Simulations running'.

Done by:

S.Krishnananth- 720819106053