**Assignment - 4**

|  |  |
| --- | --- |
| **Assignment Date** | **16 October 2022** |
| **Student Name** | **Preethi B** |
| **Student Roll Number** | **2019504048** |
| **Maximum Marks** | **2 Marks** |

**Question:**

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

**Source Code:**

#include <WiFi.h> #include <PubSubClient.h> void callback(char\* subscribetopic,byte\* payload, unsigned int payloadLength);

#define ORG "2evnkc"//IBM ORGANITION ID

#define DEVICE\_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform

#define DEVICE\_ID "PREETHI22"//Device ID mentioned in ibm watson IOT Platform #define TOKEN "098890098" //Token

String data3;

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

char publishTopic[]="iot-2/evt/distance/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);

#define ECHO\_PIN 12

#define TRIG\_PIN 13 #define led 2 void

setup() {

// put your setup code here, to run once:

Serial.begin(115200); pinMode(led, OUTPUT); pinMode(TRIG\_PIN, OUTPUT);

pinMode(ECHO\_PIN, INPUT);

wificonnect();

mqttconnect();

}

float readDistanceCM() {

digitalWrite(TRIG\_PIN, LOW);// Clear the trigger delayMicroseconds(2); digitalWrite(TRIG\_PIN, HIGH);// Sets the trigger pin to HIGH state for 10 microseconds

delayMicroseconds(10); digitalWrite(TRIG\_PIN, LOW);

int duration=pulseIn(ECHO\_PIN, HIGH);

//Serial.println(duration);

//duration = pulseIn(ECHO\_PIN, HIGH); return duration\*0.017;

//Serial.println(duration); } void loop() {

float distance = readDistanceCM(); //Serial.println(distance); bool isNearby = distance < 100; digitalWrite(led, isNearby); Serial.print("Measured distance: "); Serial.println(distance); if(distance<100){ PublishData2(distance); }else{

PublishData1(distance);

}

//PublishData(distance); delay(1000);

if(!client.loop()){

mqttconnect();

}

//delay(2000);

}

void PublishData1(float dist){ mqttconnect();

String payload= "{\"distance\":"; payload += dist; 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 PublishData2(float dist){ mqttconnect();

String payload= "{\"alert - distance\":\"";

payload += dist; 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 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); if(data3=="lighton"){ Serial.println(data3); digitalWrite(led,HIGH);

}else{

Serial.println(data3);

digitalWrite(led,LOW);

}

data3="";

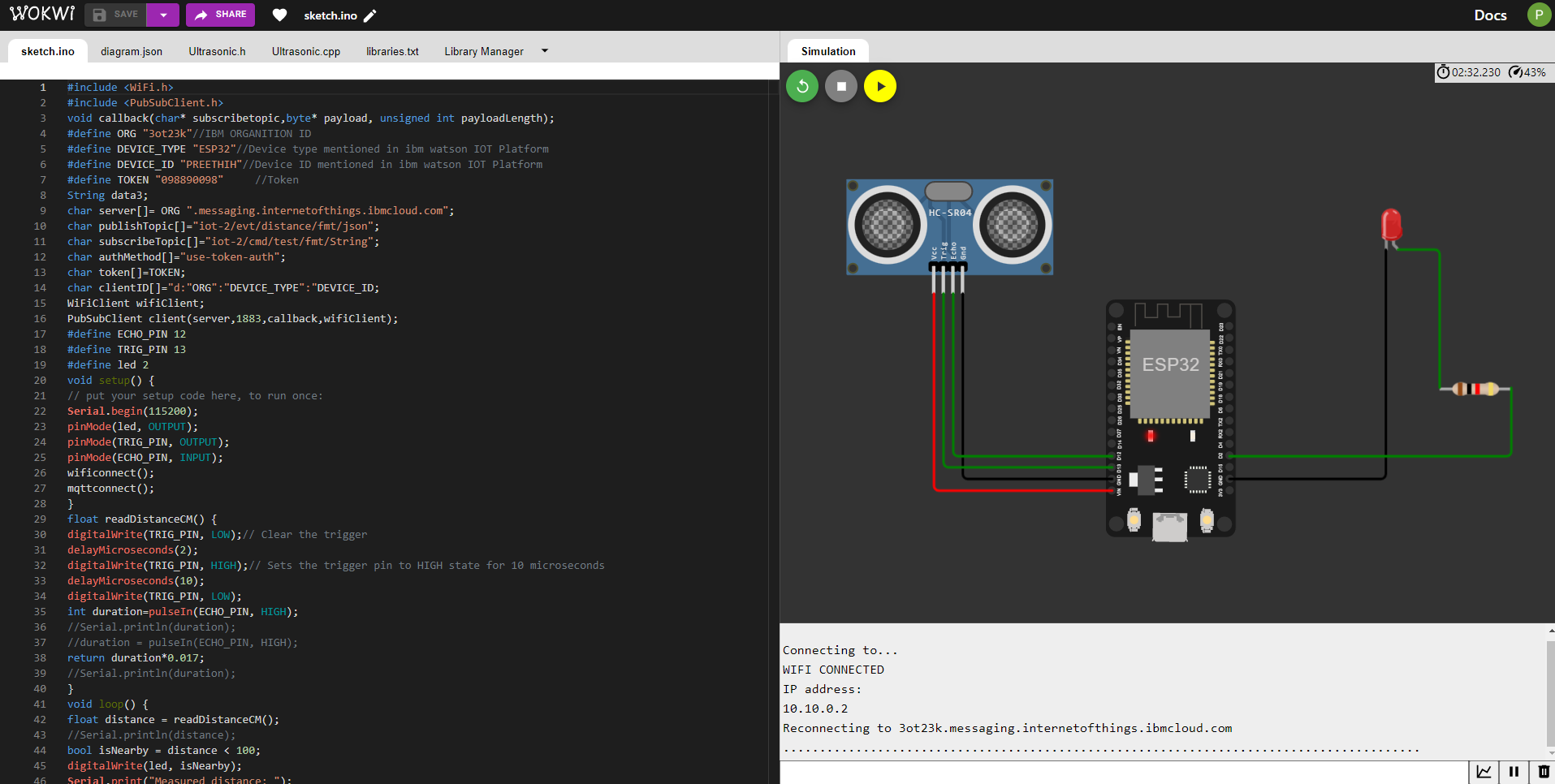
}

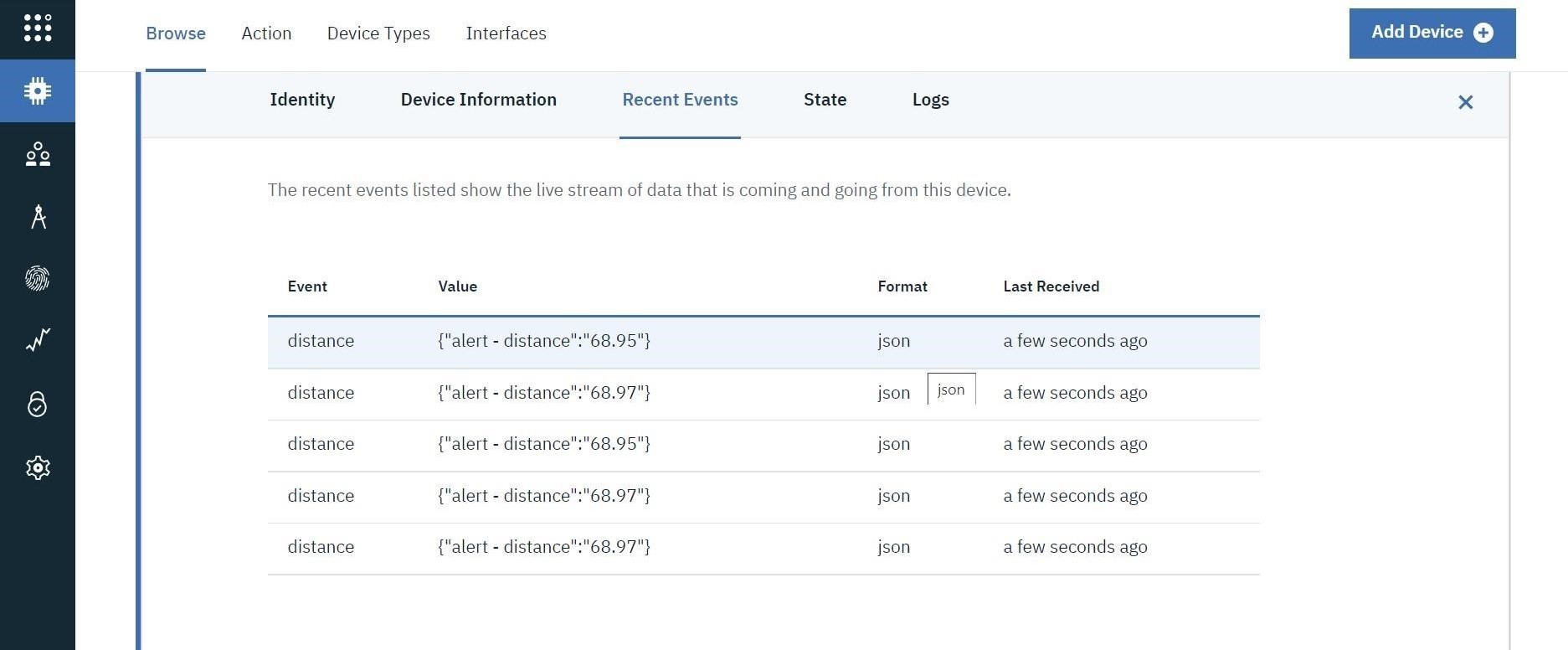
**Reference:**

**https://wokwi.com/projects/348747738584711762**

**Output:**

## Case -1: Less than 100cm – (Bulb glows and Message - “Alert”)





## Case -2: More than 100cm – (bulb off and Message “distance”

