**Assignment -4**

ESP-32 Programming

|  |  |
| --- | --- |
| Assignment Date | 30 October 2022 |
| Student Name | T.Rahul |
| Student Roll Number | 912419104023 |
| Maximum Marks | 2 Marks |

**Question-1:**

**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.**

**CODE:**

#include <WiFi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

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

#define ORG "sgbugf"//IBM ORGANITION ID

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

#define DEVICE\_ID "assignment4"//Device ID mentioned in ibm watson IOT Platform

#define TOKEN "4pa8p+(xlY@Pbra!vx"//Token

#define speed 0.034

#define led 14

//-------- 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 and format in which data to be send

char topic[] = "iot-2/cmd/home/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

//-----------------------------------------

PubSubClient client(server, 1883, wifiClient);

void publishData();

const int trigpin=5;

const int echopin=18;

String command;

String data="";

long duration;

float dist;

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

void setup()

{

**Serial**.begin(115200);

  pinMode(led, OUTPUT);

  pinMode(trigpin, OUTPUT);

  pinMode(echopin, INPUT);

  wifiConnect();

  mqttConnect();

}

void loop() {

  bool isNearby = dist < 100;

  digitalWrite(led, isNearby);

  publishData();

  delay(1000);

  if (!client.loop()) {

    mqttConnect();

  }

}

void wifiConnect() {

**Serial**.print("Connecting to "); **Serial**.print("Wifi");

  WiFi.begin("Wokwi-GUEST", "", 6);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(1000);

**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(1000);

    }

    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");

  }

}

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 = "{\"Normal Distance\":";

    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(dist>101 && dist<111){

    String payload = "{\"Alert distance\":";

    payload += dist;

    payload += "}";

**Serial**.print("\n");

**Serial**.print("Sending payload: ");

**Serial**.println(payload);

     if(client.publish(publishTopic, (char\*) payload.c\_str())) {

**Serial**.println("Warning crosses 110cm -- it automaticaly of the loop");

      digitalWrite(led,HIGH);

    }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);

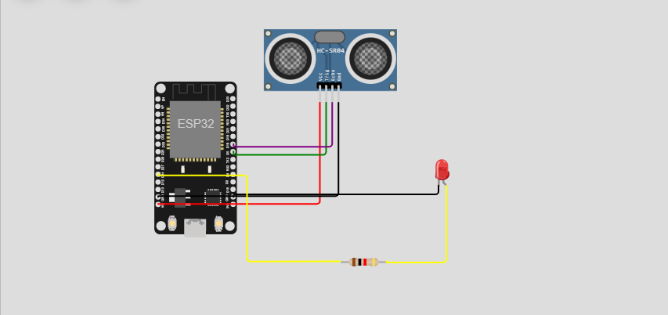
    digitalWrite(led,HIGH);

  }

  data3="";

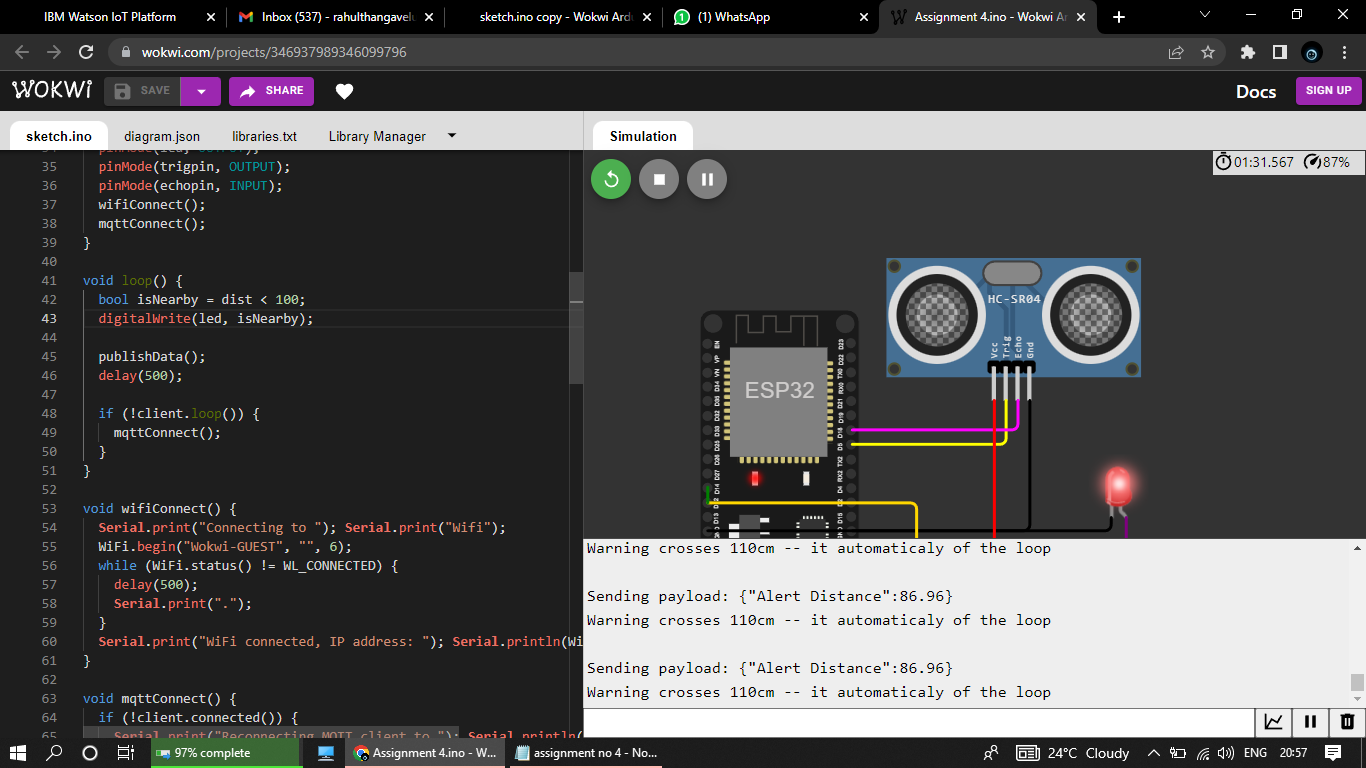
}

**CIRCUIT DIAGRAM:**

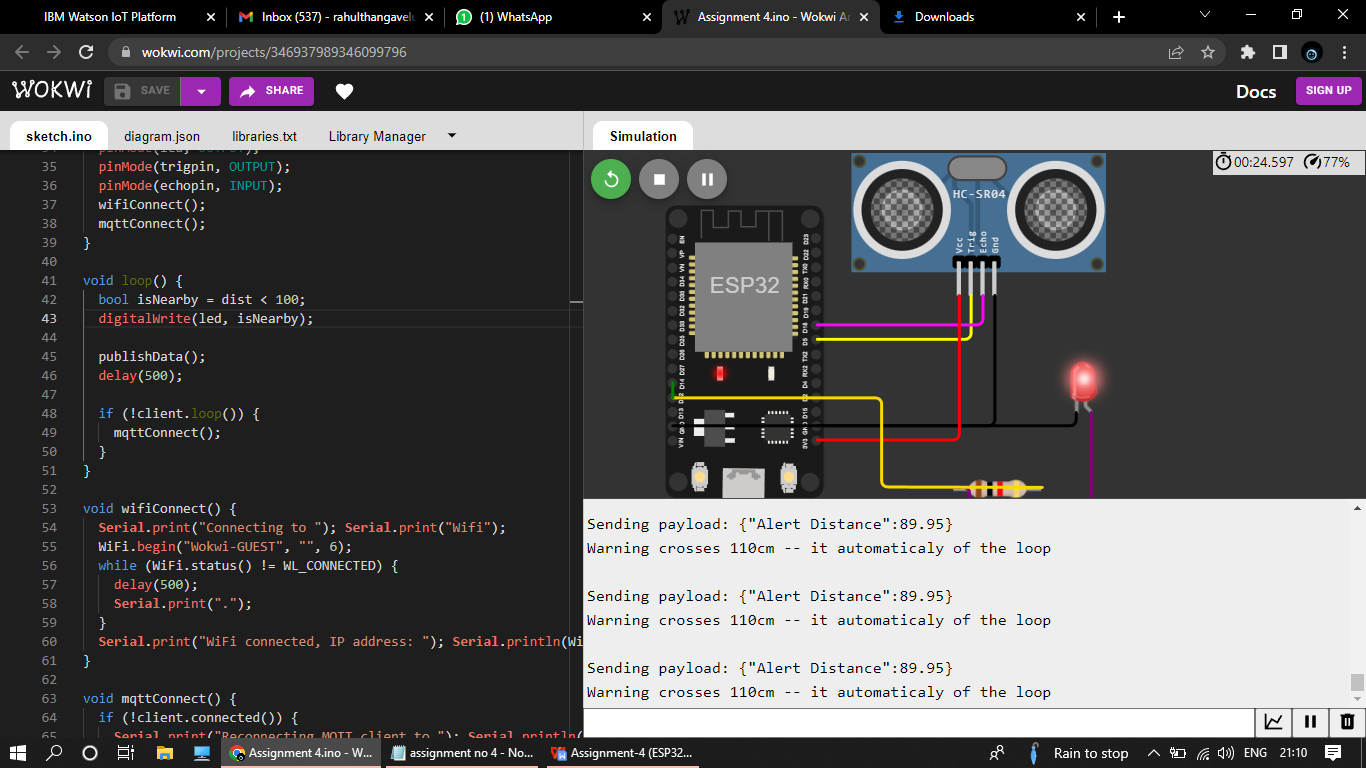
****

**OUTPUT:**

1. **When Distance < 100 cm, alert with warning message occurs.**



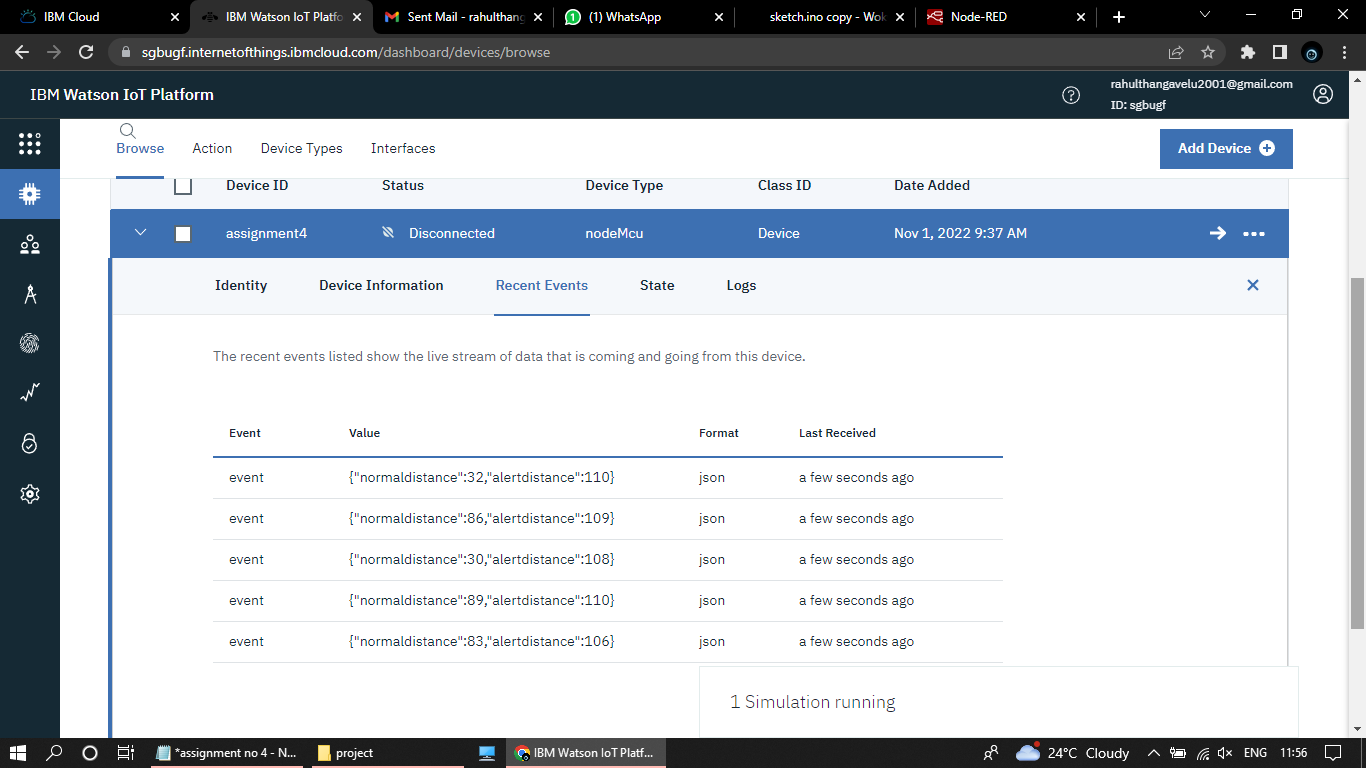
1. **When distance > 100cm <110cm, it will show normal distance.**



**Wokwi simulation link:**

<https://wokwi.com/projects/346841205890351700>

**IBM CLOUD OUTPUT:**

****