Assignment - 4

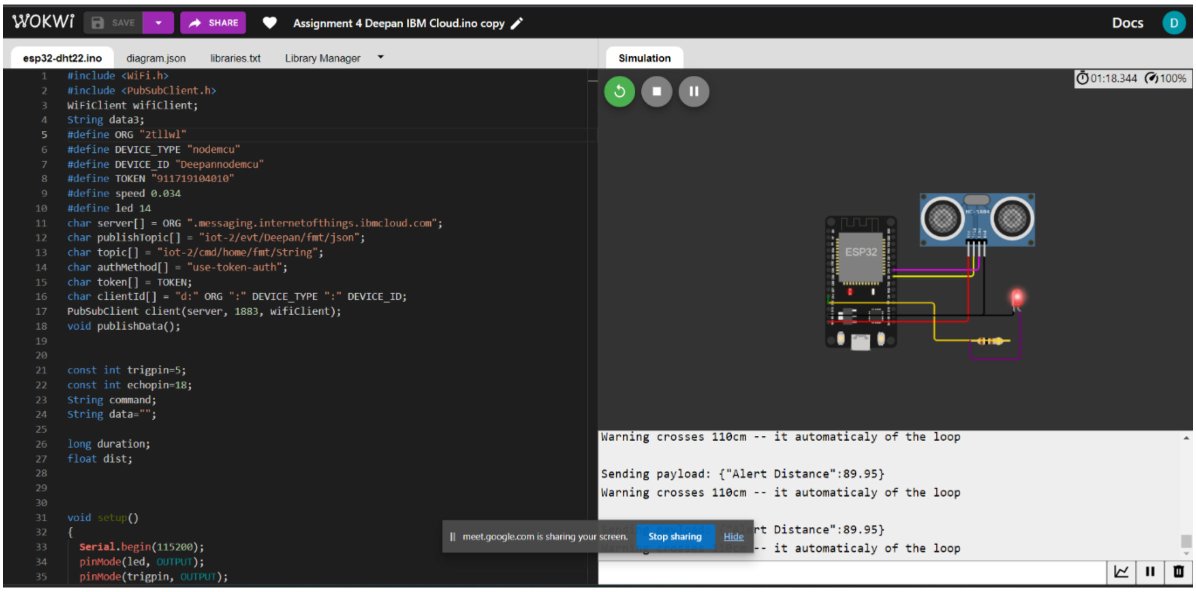
wokwi for the ultrasonic sensor

Date	01 November 2022
Student Name	Deepan R
Student Roll No	911719104010
Maximum Marks	2 Marks

Question-1:

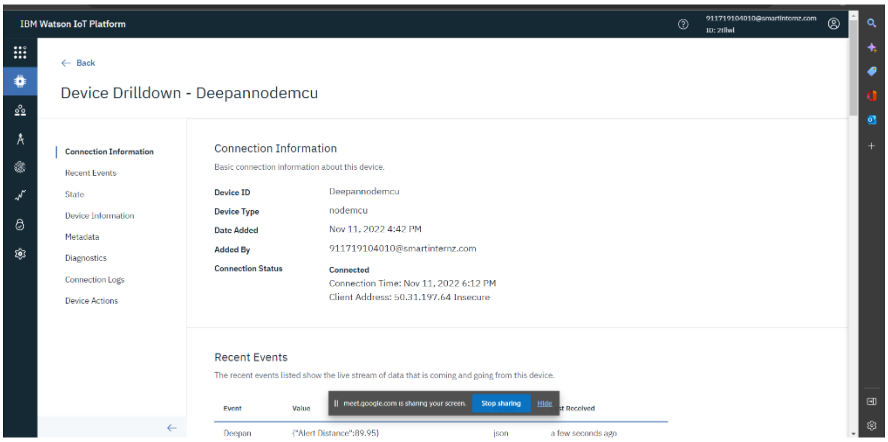
Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud.

Output :



```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "2tllwl"
#define DEVICE_TYPE "nodemcu"
#define DEVICE_ID "Deepannodemcu"
#define TOKEN "911719104010"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Deepan/fmt/json";
char topic[] = "iot-2/cmd/home/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();


const int trigpin=5;
const int echopin=18;
String command;
String data="";

long duration;
float dist;



void setup()
{
  Serial.begin(115200);
  pinMode(led, OUTPUT);
  pinMode(trigpin, OUTPUT);
```

```
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert Distance":89.95}
Warning crosses 110cm -- it automaticaly of the loop

rt Distance":89.95}
-- it automaticaly of the loop
```


IBM Watson IoT Platform
911719104010@smartinternz.com
ID: 2tllwl
Back
meet.google.com is sharing your screen.
Stop sharing
Hide

Device Drilldown - Deepannodemcu

Connection Information
Recent Events
State
Device Information
Metadata
Diagnostics
Connection Logs
Device Actions

Connection Information

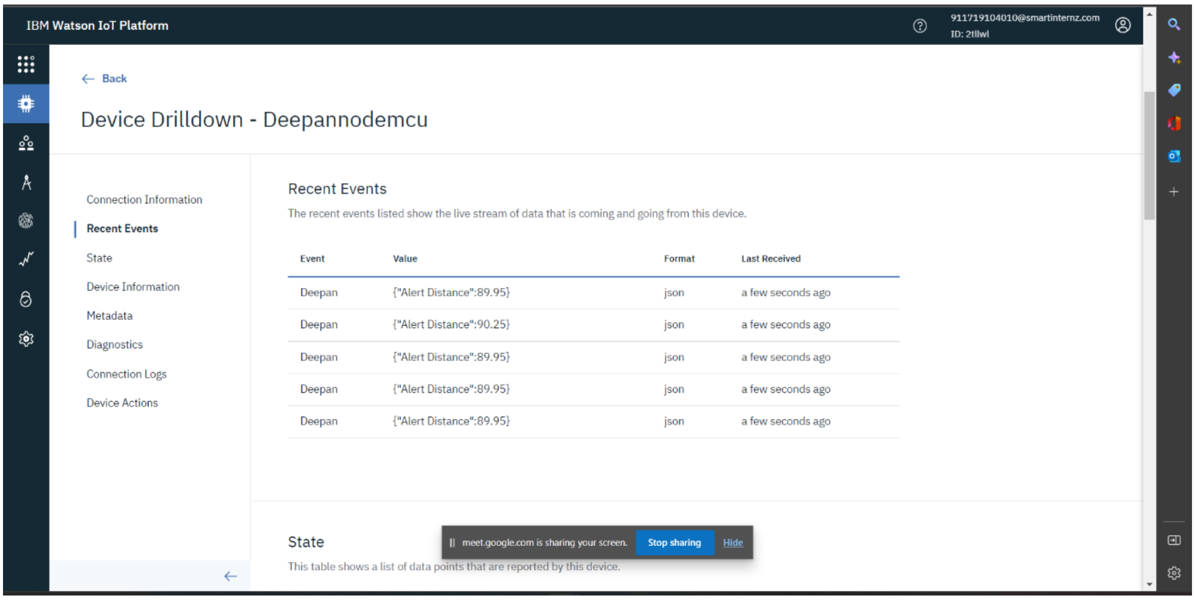
Basic connection information about this device.

Device ID	Deepannodemcu
Device Type	nodemcu
Date Added	Nov 11, 2022 4:42 PM
Added By	911719104010@smartinternz.com
Connection Status	Connected Connection Time: Nov 11, 2022 6:12 PM Client Address: 50.31.197.64 Insecure

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value		t Received
Deepan	{"Alert Distance":89.95}	json	a few seconds ago



Device Drilldown - Deepannodemcu

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Deepan	{"Alert Distance":89.95}	json	a few seconds ago
Deepan	{"Alert Distance":90.25}	json	a few seconds ago
Deepan	{"Alert Distance":89.95}	json	a few seconds ago
Deepan	{"Alert Distance":89.95}	json	a few seconds ago
Deepan	{"Alert Distance":89.95}	json	a few seconds ago

State

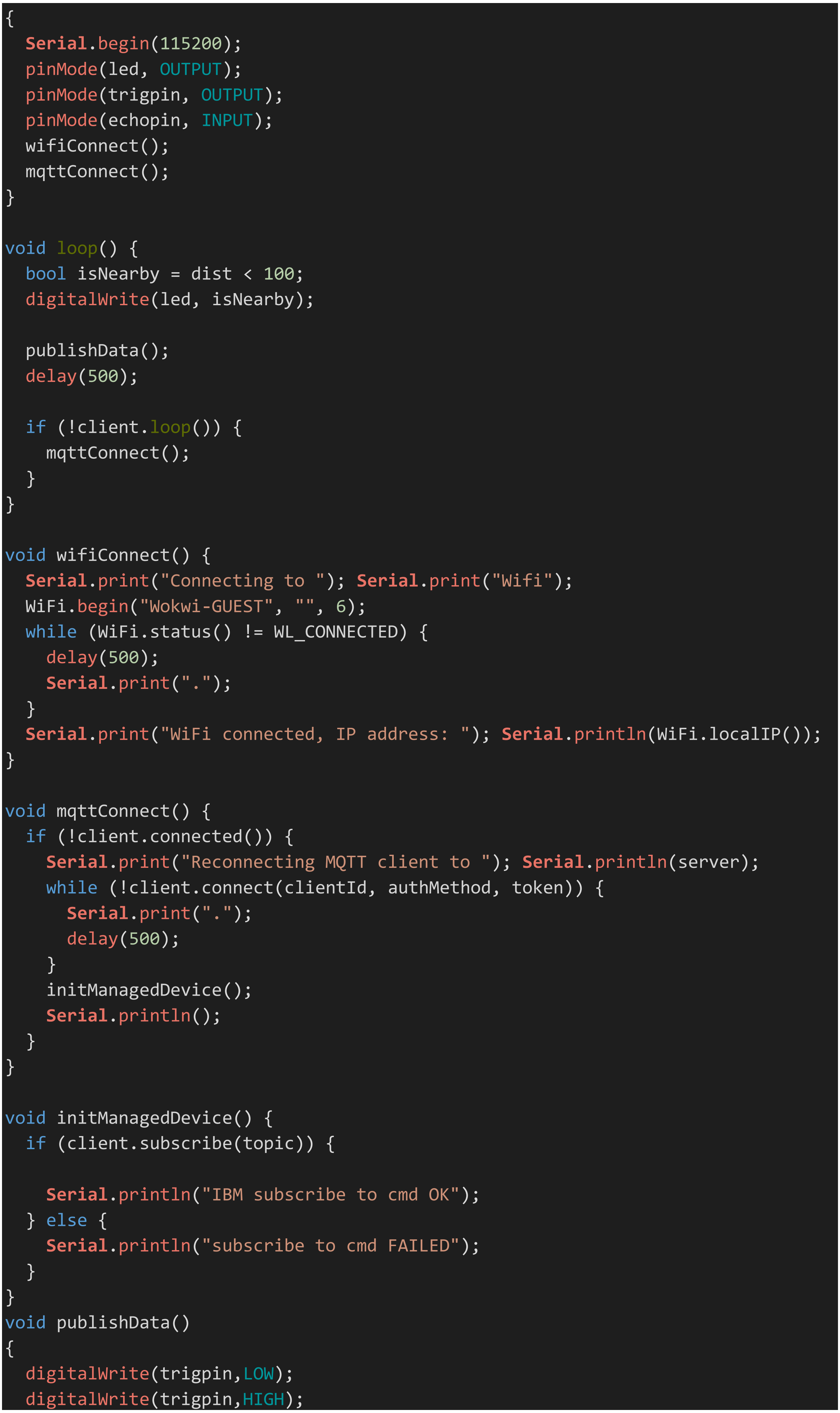
meet.google.com is sharing your screen. Stop sharing Hide

This table shows a list of data points that are reported by this device.

Link : Assignment 4 Deepan IBM Cloud.ino copy - Wokwi Arduino and ESP32 Simulator

Code :

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "2tllwl"
#define DEVICE_TYPE "nodemcu"
#define DEVICE_ID "Deepannodemcu"
#define TOKEN "911719104010"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Deepan/fmt/json";
char topic[] = "iot-2/cmd/home/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();


const int trigpin=5;
const int echopin=18;
String command;
String data="";

long duration;
float dist;



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(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("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 = "{\"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);
    }

  }
    if(dist>101 && dist<111){
    String payload = "{\"Normal Distance\":";
    payload += dist;
    payload += "}";

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

    }

  }



  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="";
}
```