ASSIGNMENT-4

Assignment Date	27 october 2022
Student Name	K.BRINDHA DEVI
Student Roll Number	711619106701
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.Upload document with wokwi share link and images of ibm cloud.

SOLUTION:

PROGRAM:

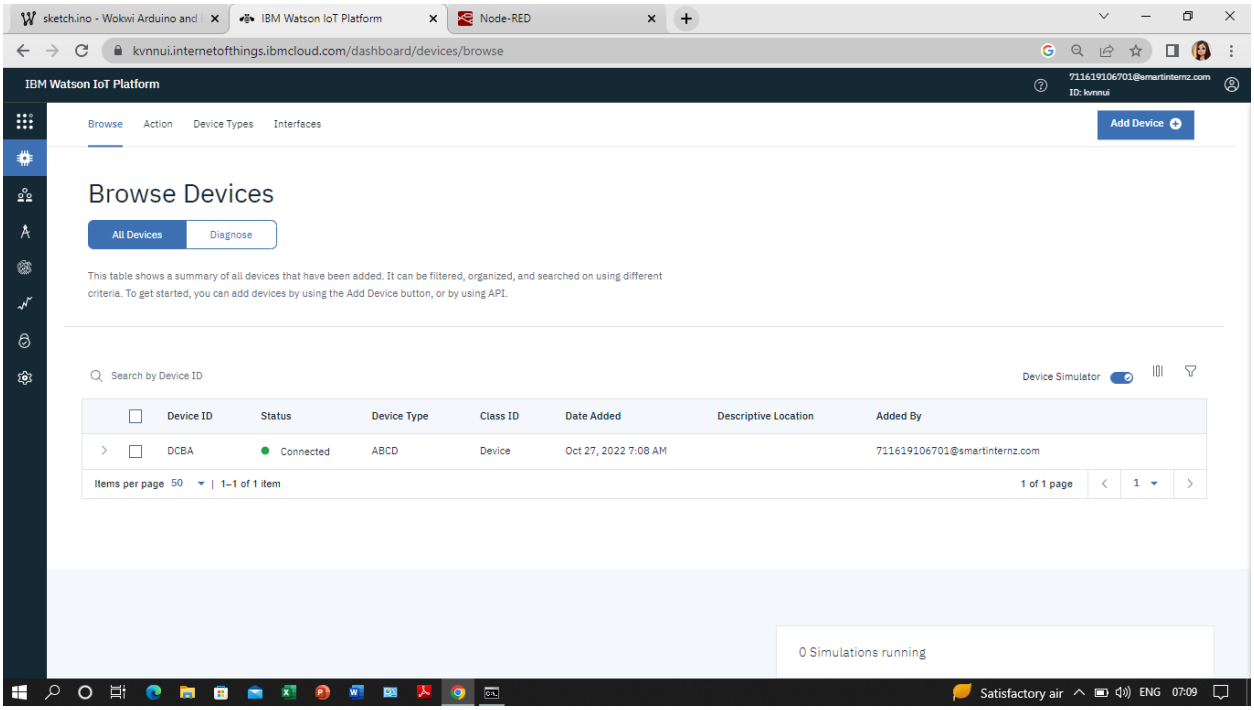
```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;

#define DEVICE_TYPE "ABCD"
#define DEVICE_ID "DCBA"
#define TOKEN "987654321"
#define speed 0.034
#define led 14
#define ORG "kvnnui"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status/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;
```

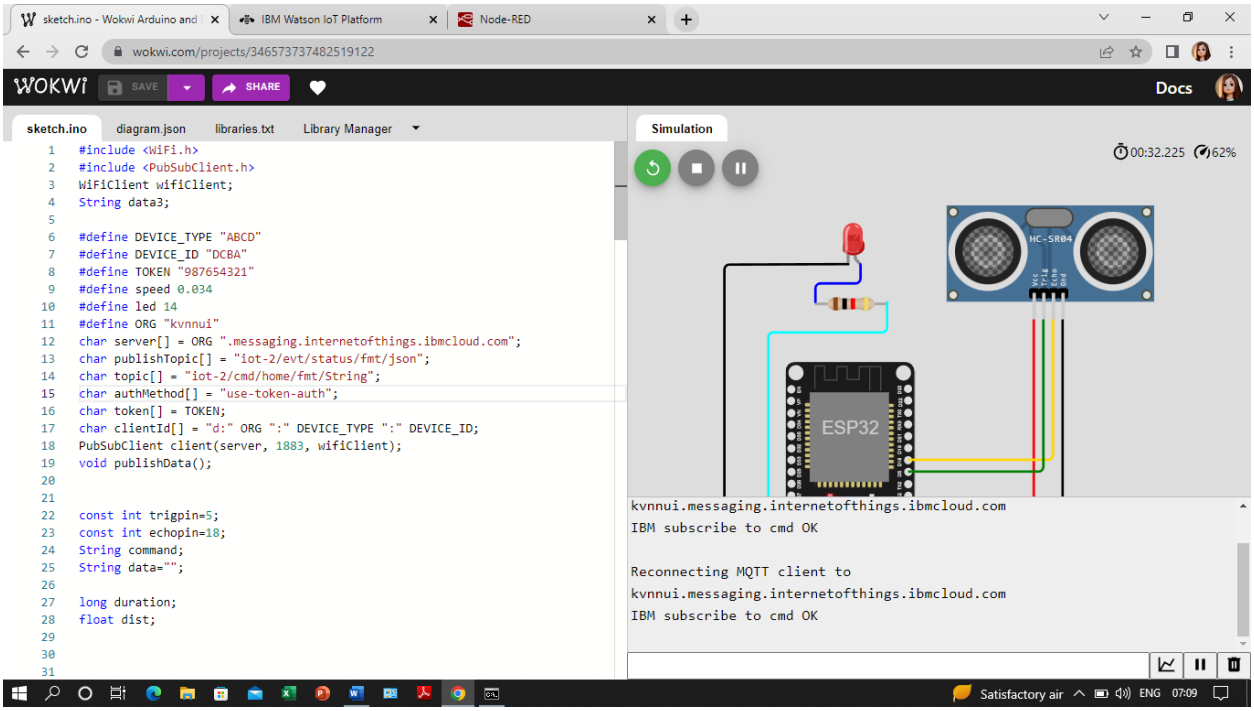
```
void setup()
{
  Serial.begin(115200);
  pinMode(led, OUTPUT);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
}

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() {
```

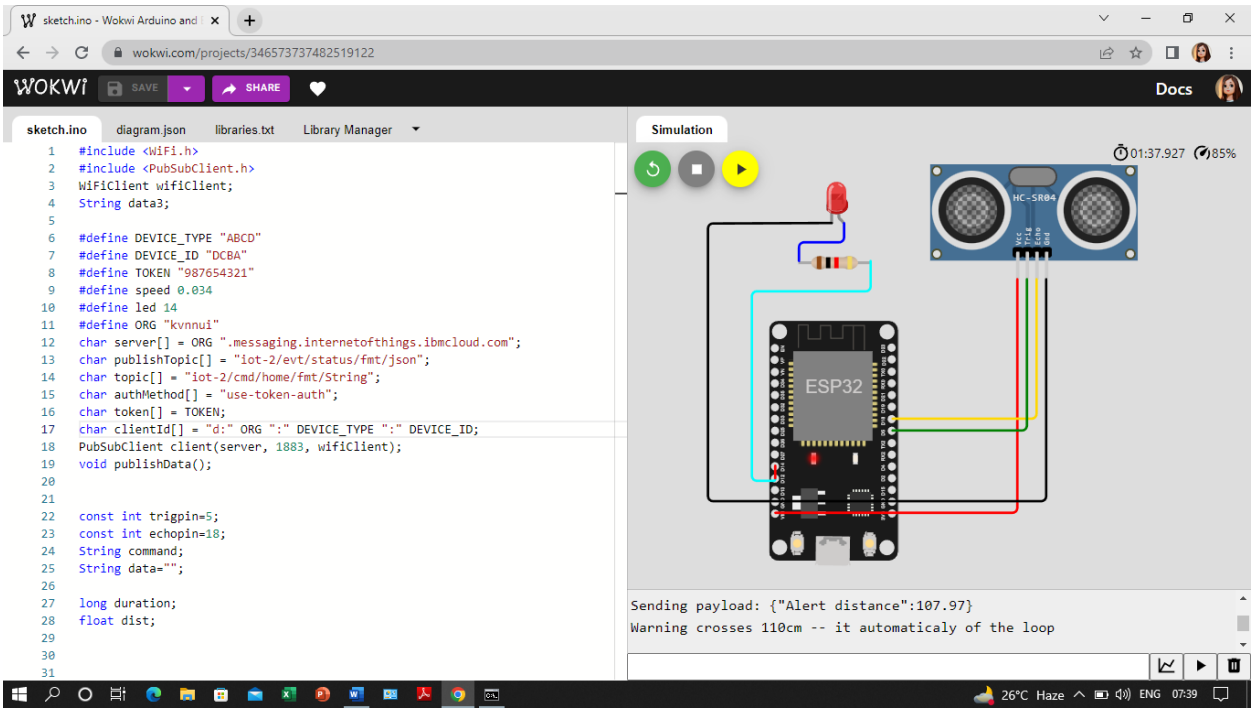
```
  if (client.subscribe(topic)) {
    // Serial.println(client.subscribe(topic));
    Serial.println("IBM subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}
void publishData()
{
  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  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");
    }

  }
```

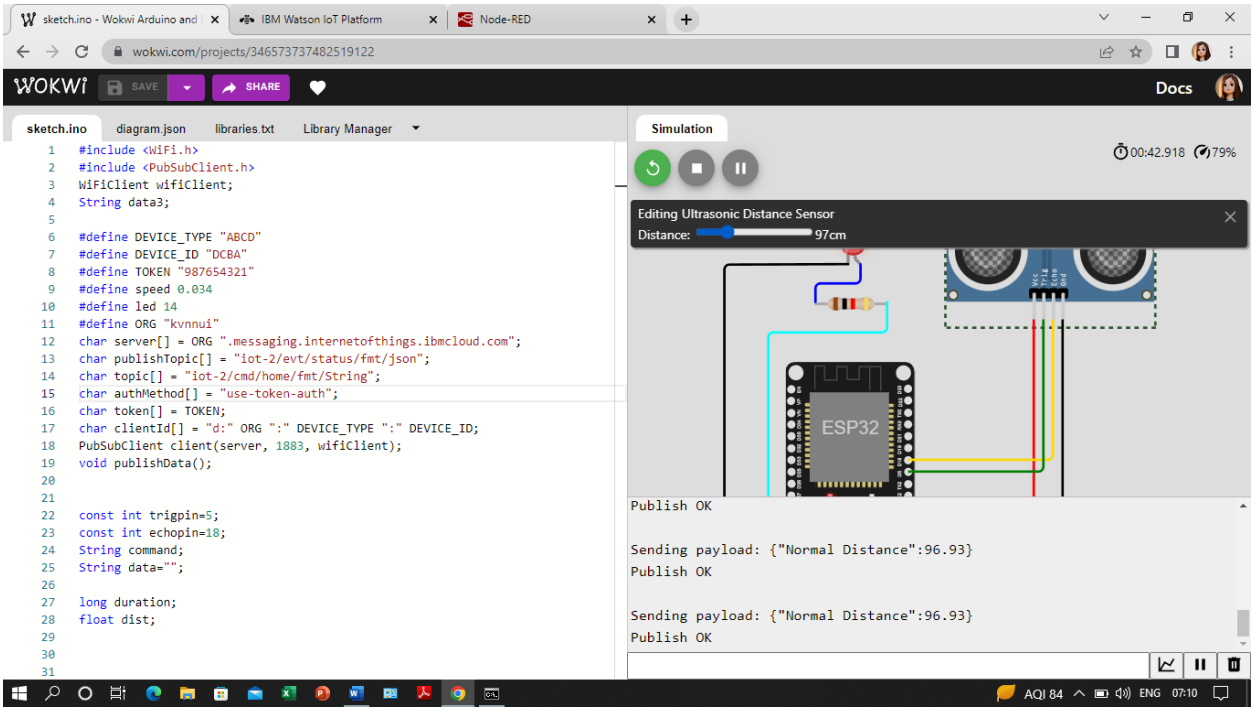
```
  }
  void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength){
  Serial.print("callback invoked for topic:");
  Serial.println(subscribeTopic);
  for(int i=0; i<payloadLength; i++){
    dist += (char)payload[i];
  }
  Serial.println("data:"+ data3);
  if(data3=="lighton"){
    Serial.println(data3);
    digitalWrite(led,HIGH);
  }
  data3="";
}
```

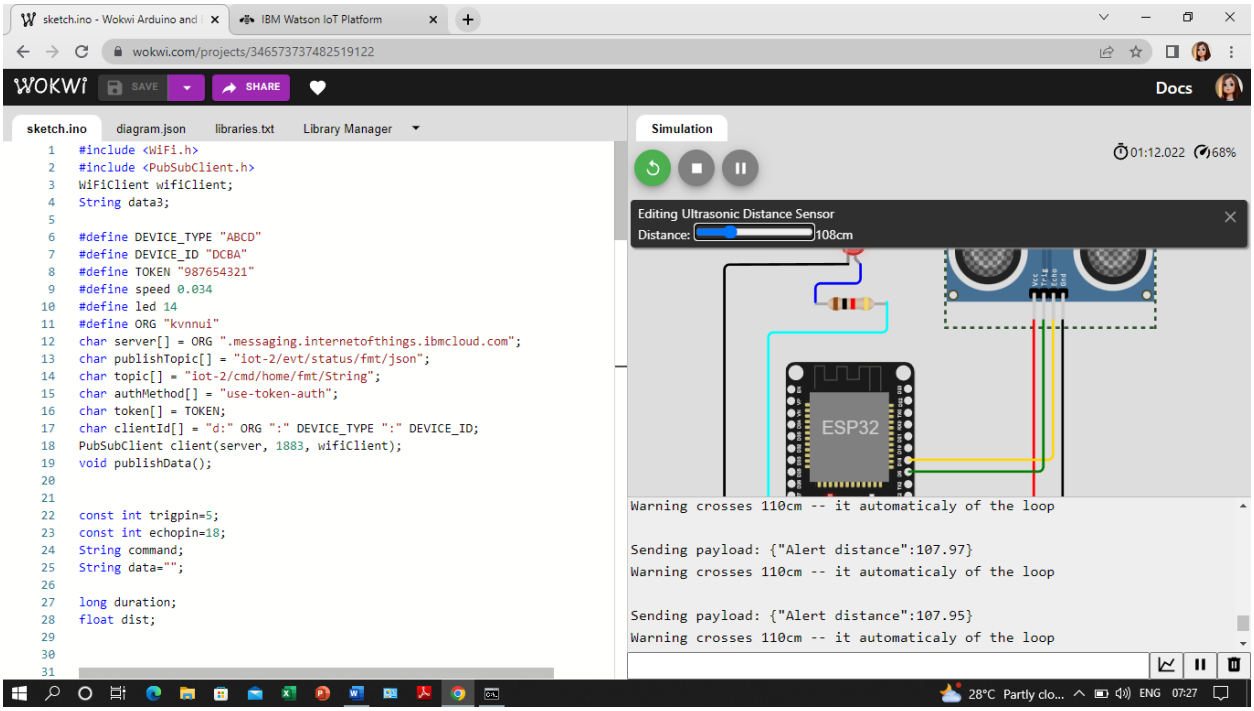
SIMULATION SCREENSHOTS:

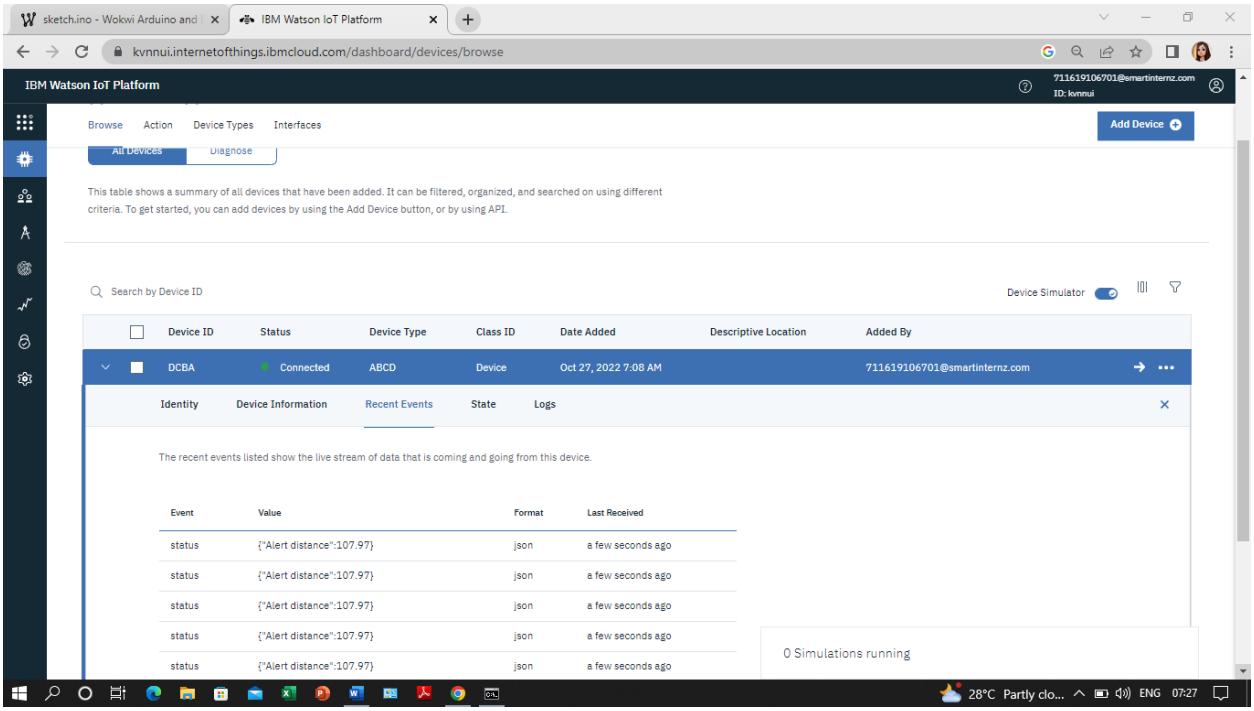



sketch.ino - Wokwi Arduino and
IBM Watson IoT Platform
Node-RED
wokwi.com/projects/346573737482519122
WOKWI
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define DEVICE_TYPE "ABCD"
#define DEVICE_ID "DCBA"
#define TOKEN "987654321"
#define speed 0.034
#define led 14
#define ORG "kvnnui"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status/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;
Simulation
00:32.225
62%
HC-SR04
ESP32
kvnnui.messaging.internetofthings.ibmcloud.com
IBM subscribe to cmd OK
Reconnecting MQTT client to
kvnnui.messaging.internetofthings.ibmcloud.com
IBM subscribe to cmd OK
Satisfactory air
ENG
07:09


sketch.ino - Wokwi Arduino and
wokwi.com/projects/346573737482519122
WOKWI
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define DEVICE_TYPE "ABCD"
#define DEVICE_ID "DCBA"
#define TOKEN "987654321"
#define speed 0.034
#define led 14
#define ORG "kvnnui"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status/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;
Simulation
01:37.927
85%
HC-SR04
ESP32
Sending payload: {"Alert distance":107.97}
Warning crosses 110cm -- it automaticaly of the loop
26°C Haze
ENG
07:39


sketch.ino - Wokwi Arduino and
IBM Watson IoT Platform
Node-RED
wokwi.com/projects/346573737482519122
WOKWI
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define DEVICE_TYPE "ABCD"
#define DEVICE_ID "DCBA"
#define TOKEN "987654321"
#define speed 0.034
#define led 14
#define ORG "kvnnui"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status/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;
Simulation
00:42.918
79%
Editing Ultrasonic Distance Sensor
Distance:
97cm
ESP32
Publish OK
Sending payload: {"Normal Distance":96.93}
Publish OK
Sending payload: {"Normal Distance":96.93}
Publish OK
AQI 84
ENG
07:10


sketch.ino - Wokwi Arduino and
IBM Watson IoT Platform
wokwi.com/projects/346573737482519122
WOKWI
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define DEVICE_TYPE "ABCD"
#define DEVICE_ID "DCBA"
#define TOKEN "987654321"
#define speed 0.034
#define led 14
#define ORG "kvnnui"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status/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;
Simulation
01:12.022
68%
Editing Ultrasonic Distance Sensor
Distance:
108cm
ESP32
Warning crosses 110cm -- it automaticaly of the loop
Sending payload: {"Alert distance":107.97}
Warning crosses 110cm -- it automaticaly of the loop
Sending payload: {"Alert distance":107.95}
Warning crosses 110cm -- it automaticaly of the loop
28°C Partly clo...
ENG
07:27



WOKWI LINK:

https://wokwi.com/projects/346573737482519122