N.S.N. COLLEGE OF ENGINEERING AND TECHNOLOGY

N.S.N. Kalvi Nagar, NH-7, Manalmedu, Karur - 639 003, TN, India.
(Approved by AICTE New Delhi, Affiliated to Anna University, Chennai)
(An ISO 9001:2008 Certified Institution)

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

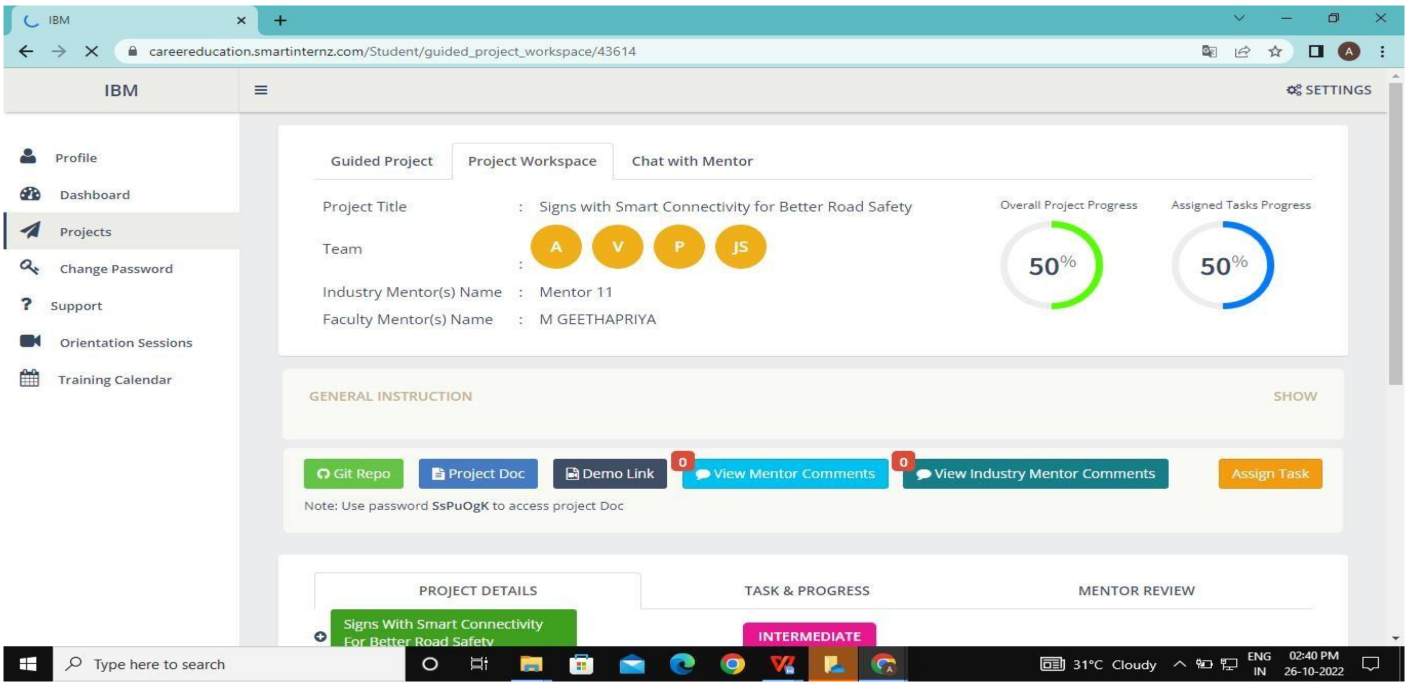
## IV YEAR / VII SEMESTER (ODD)

## BATCH: 2019-2023

## ACADEMIC YEAR 2022-2023

## ASSIGNMENT - IV

| | |
|---|---|
| TEAM ID | : PNT2022TMID48721 |
| TITLE OF THE PROJECT | : SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY |
| DOMIN | : INTERNET OF THINGS (IOT) |
| TEAM LEAD | : AYYAPPAN S |
| TEAM MEMBER | : VIGNESH K |
| TEAM MEMBER | : PARTHIBAN M |
| TEAM MEMBER | : JABAR SATHIK S |
| INDUSTRY MENTOR | : MENTOR 11 |
| FACULTY MENTOR(S) NAME | : M GEETHAPRIYA |



Project Title : Signs with Smart Connectivity for Better Road Safety

Team : A V P JS

Industry Mentor(s) Name : Mentor 11

Faculty Mentor(s) Name : M GEETHAPRIYA

Overall Project Progress 50%

Assigned Tasks Progress 50%

Note: Use password **SsPuOgK** to access project Doc

Signs With Smart Connectivity For Better Road Safety

# ASSIGNMENT - IV

## DISTANCE DETECTION USING ULTRASONIC SENSOR

**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.**

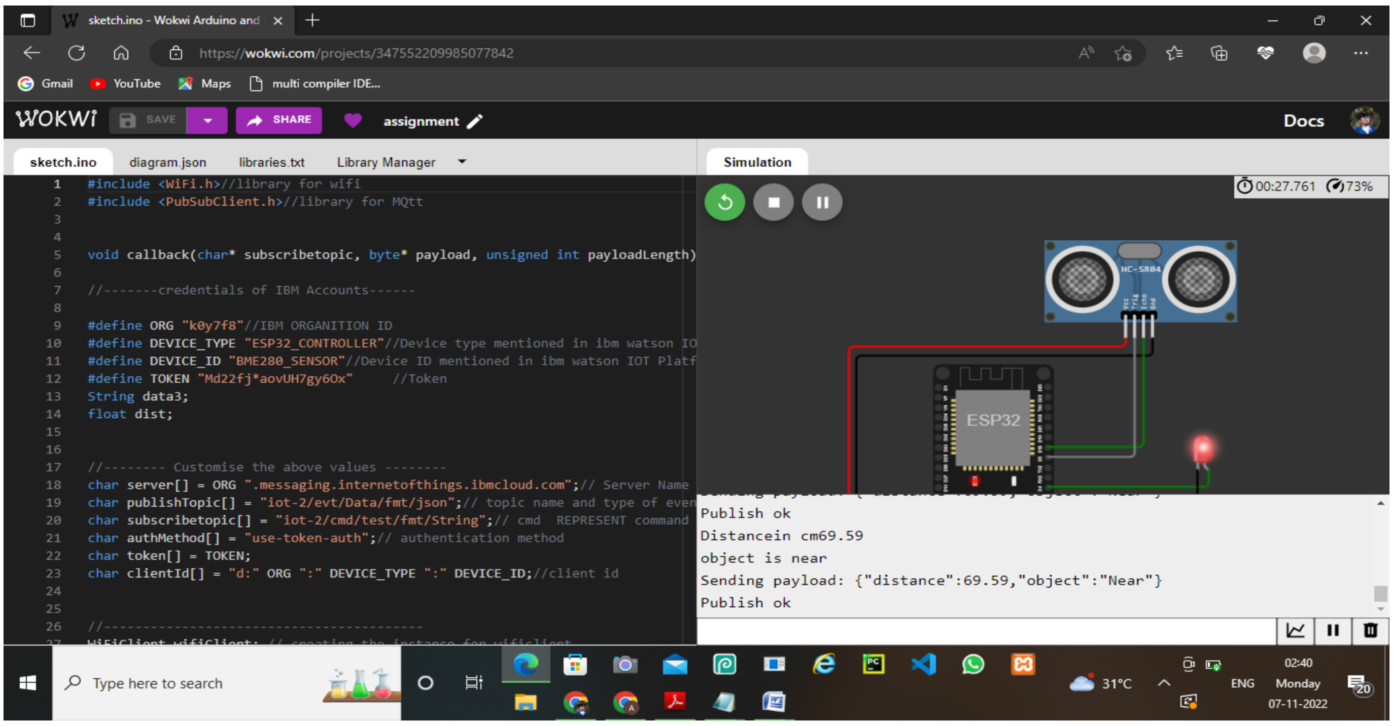
WOKWI LINK : https://wokwi.com/projects/347552209985077842

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 "k0y7f8"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32_CONTROLLER"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "BME280_SENSOR"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "Md22fj*aovUH7gy6Ox"     //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 perform 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;//client id


//-------------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand 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 to ibm 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 data on 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 credentials to 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 int payloadLength)
{

  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 :

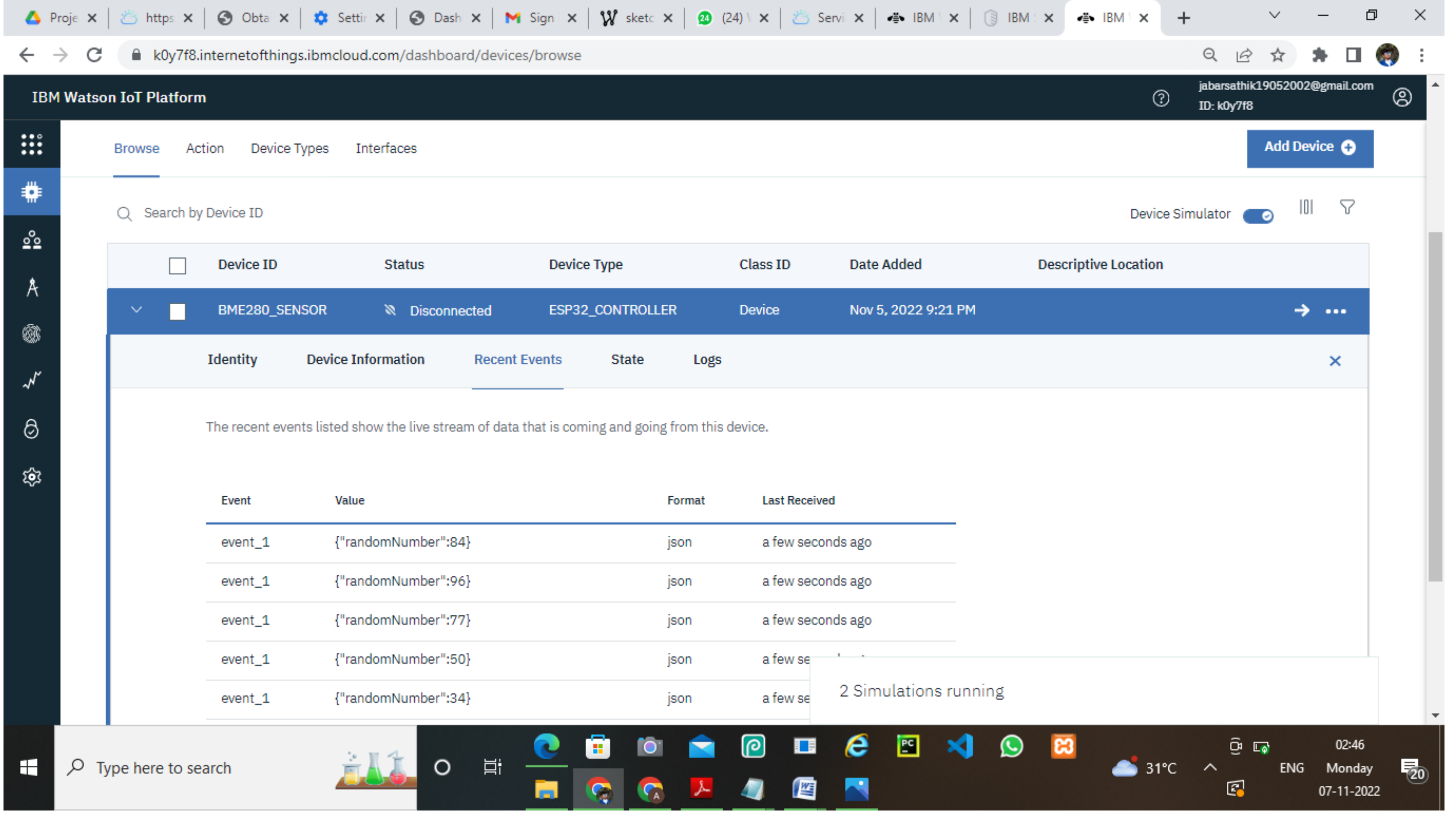
### When object is not near to the ultrasonic sensor



```
#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 "k0y7f8"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32_CONTROLLER"//Device type mentioned in ibm watson IO
#define DEVICE_ID "BME280_SENSOR"//Device ID mentioned in ibm watson IOT Platf
#define TOKEN "Md22fj*aovUH7gy6Ox"     //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 even
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//----------------------------------------
```

```
Publish ok
Distancein cm69.59
object is near
Sending payload: {"distance":69.59,"object":"Near"}
Publish ok
```

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

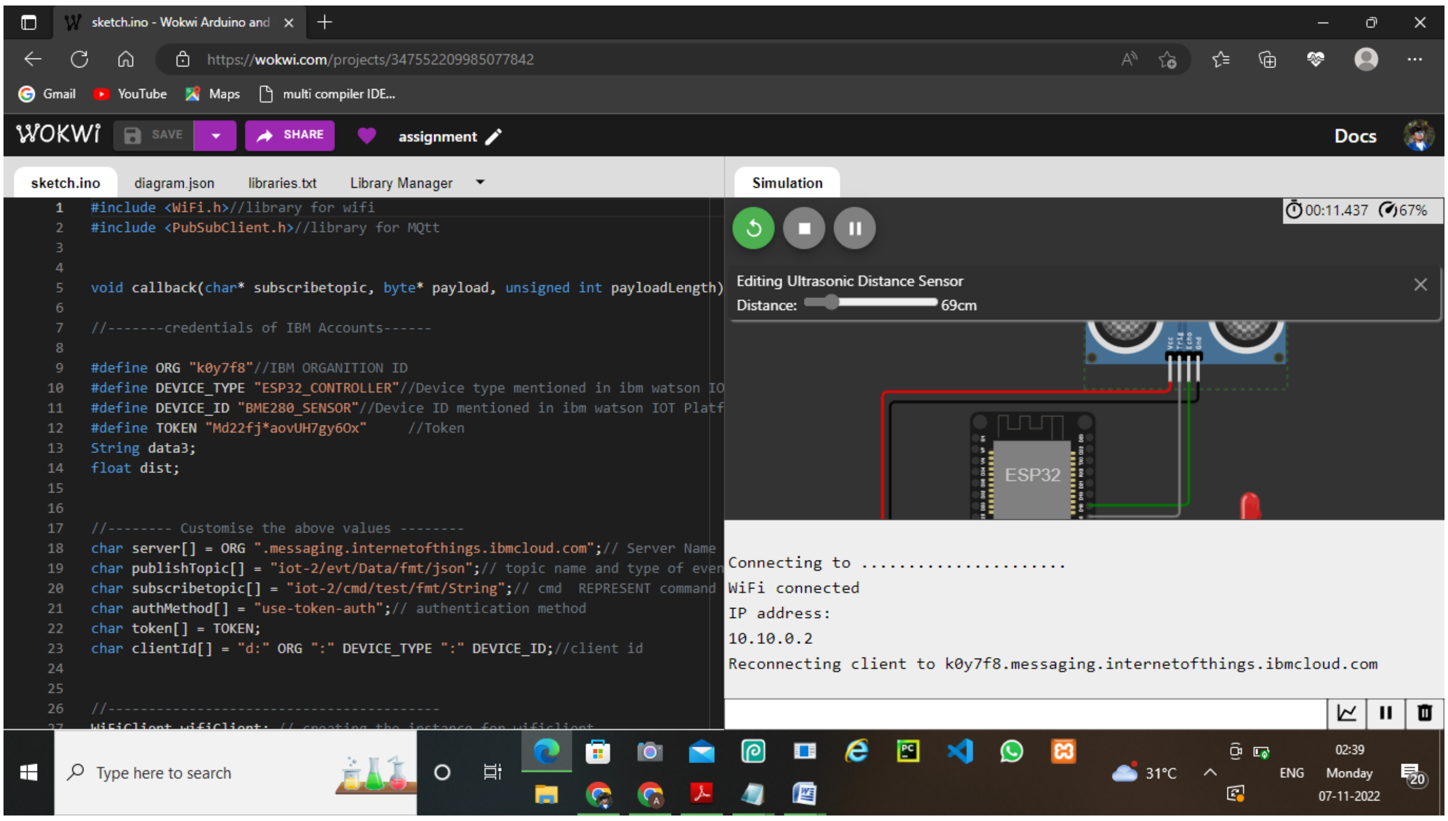


| Device ID | Status | Device Type | Class ID | Date Added | Descriptive Location |
|---|---|---|---|---|---|
| BME280_SENSOR | Disconnected | ESP32_CONTROLLER | Device | Nov 5, 2022 9:21 PM | |

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

| Event | Value | Format | Last Received |
|---|---|---|---|
| event_1 | {"randomNumber":84} | json | a few seconds ago |
| event_1 | {"randomNumber":96} | json | a few seconds ago |
| event_1 | {"randomNumber":77} | json | a few seconds ago |
| event_1 | {"randomNumber":50} | json | a few se |
| event_1 | {"randomNumber":34} | json | a few se |

## When object is  nearer to the ultrasonic sensor



```
#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 "k0y7f8"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32_CONTROLLER"//Device type mentioned in ibm watson IO
#define DEVICE_ID "BME280_SENSOR"//Device ID mentioned in ibm watson IOT Platf
#define TOKEN "Md22fj*aovUH7gy6Ox"     //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 even
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
```

```
Connecting to ......................
WiFi connected
IP address:
10.10.0.2
Reconnecting client to k0y7f8.messaging.internetofthings.ibmcloud.com
```