Assignment – 4

Write code and connections in wokwi for ultrasonic sensor

Assignment Date	**30.10 2022**
Student Name	**CHANDRU S**
Student Roll Number	**811019106007**
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.

Solution :

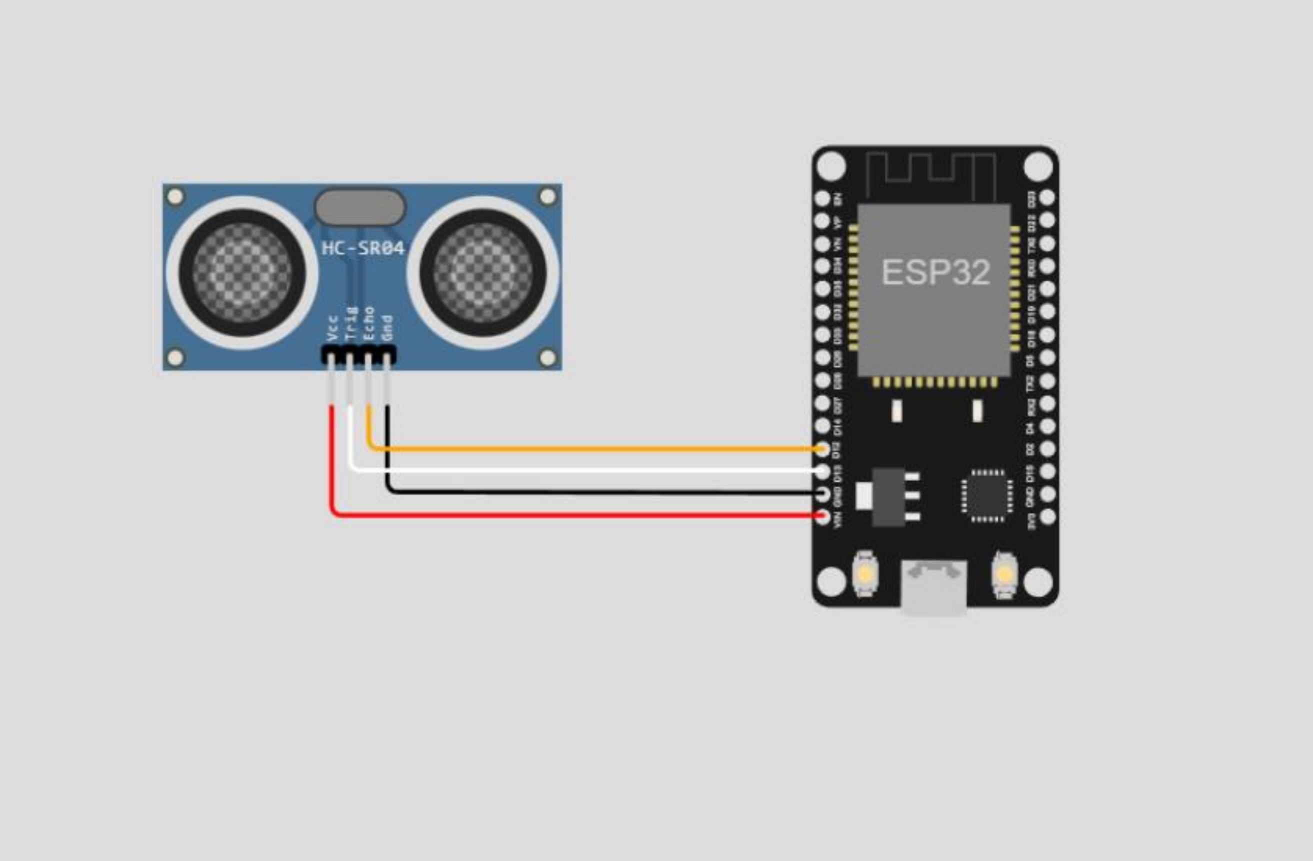
Wokwi Project Link :

https://wokwi.com/projects/346927211495817810

Code:

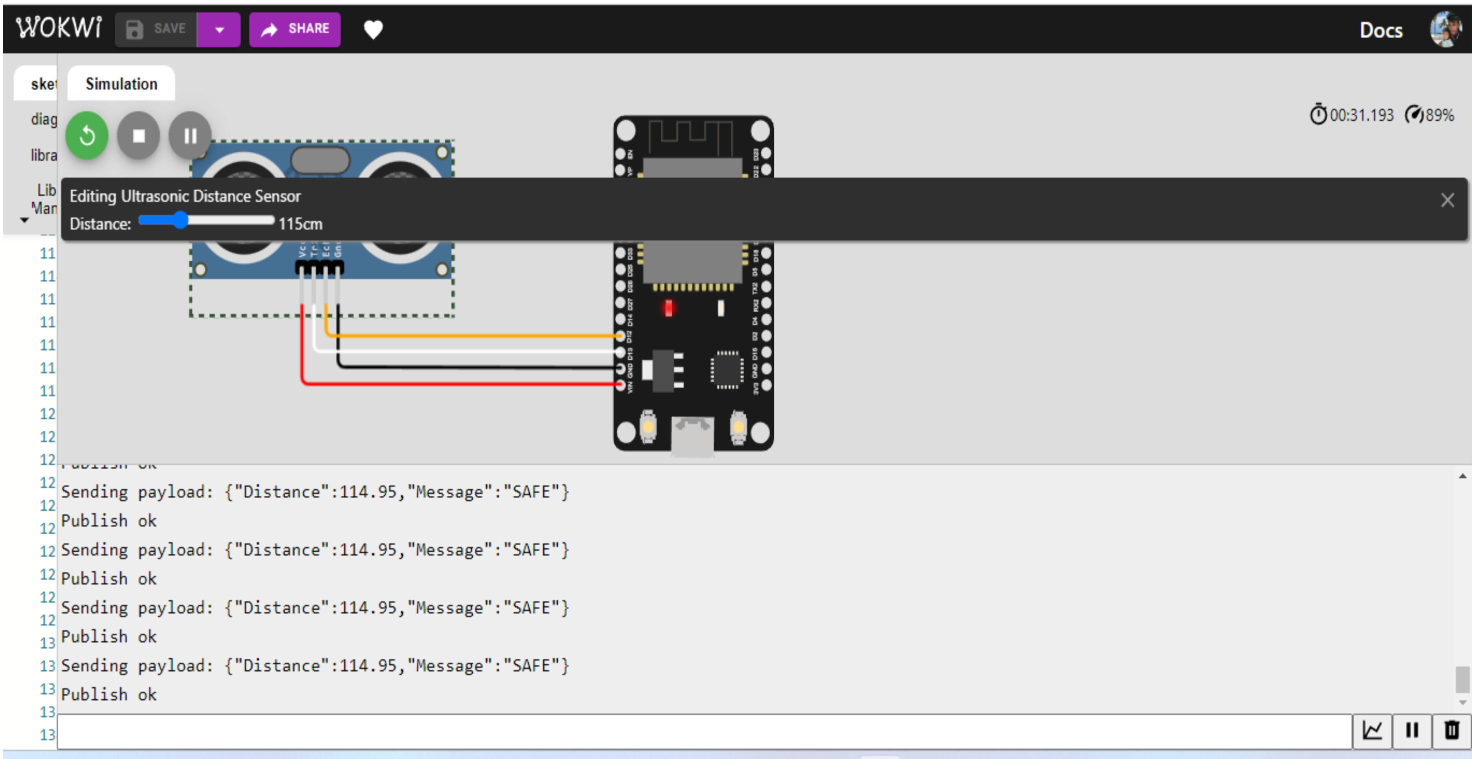
```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#define TRIG_PIN 13
#define ECHO_PIN 12



void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

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

#define ORG "rlbybh"//IBM ORGANITION ID
#define DEVICE_TYPE "b11m3edevicetype"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "b11m3edeviceid"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "Eqwj&XXgCyiaFbqr7K"     //Token



//-------- 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/command/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


void setup()// configureing the ESP32
{
  Serial.begin(115200);
  pinMode(TRIG_PIN, OUTPUT);
  digitalWrite(TRIG_PIN, LOW);
  pinMode(ECHO_PIN, INPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  float duration_us = pulseIn(ECHO_PIN, HIGH);
  float distance = 0.017 * duration_us;

  if(distance<100)
  {
    PublishData(distance,"ALERT");
  }else{
    PublishData(distance,"SAFE");
  }

  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}



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

void PublishData(float d,char s[]) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"Distance\":";
  payload+=d;
  payload+=",";
  payload+="\"Message\":";
  payload+="\"";
  payload+=s;
  payload+="\"";
  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)
{
  }
```

Circuit Diagram :



Output :

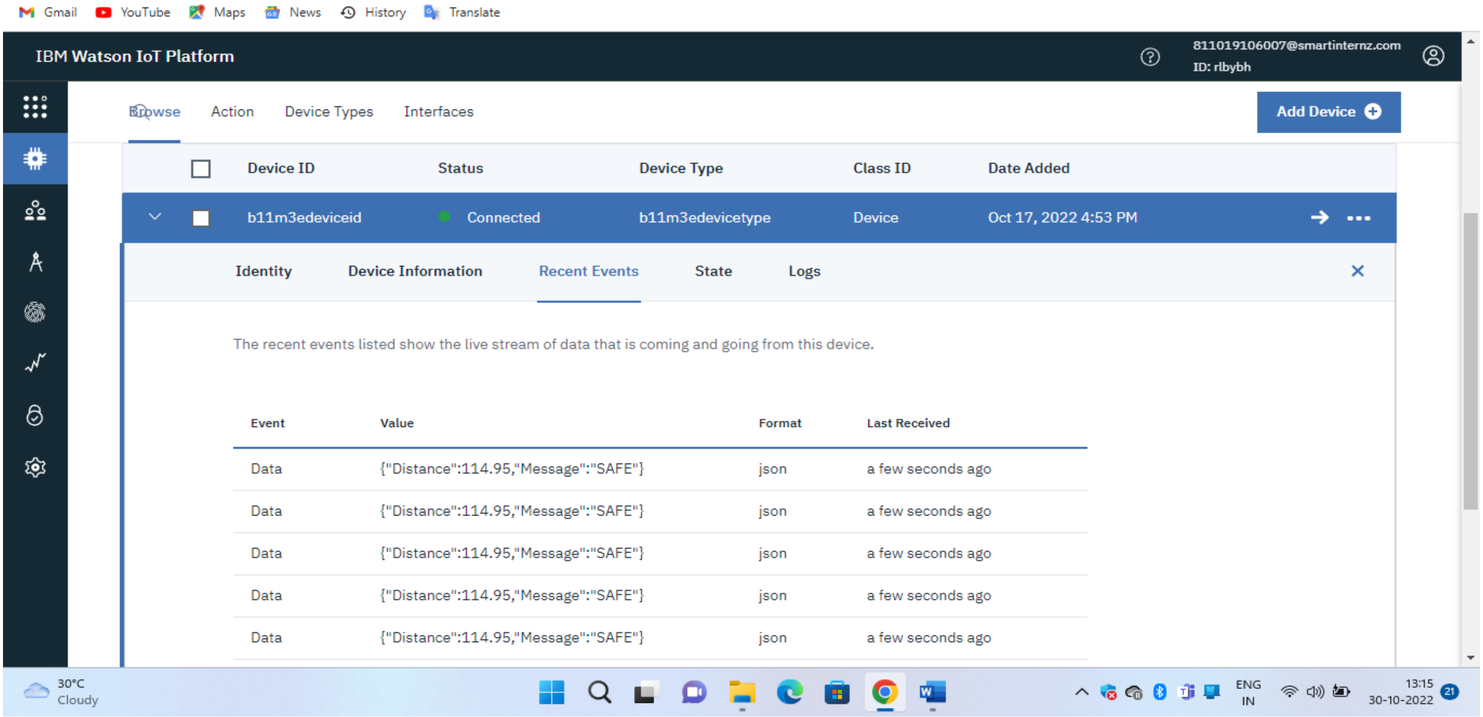
When Distance is Greater than 100 cm it shows “ SAFE”



Editing Ultrasonic Distance Sensor
Distance: 115cm

```
Sending payload: {"Distance":114.95,"Message":"SAFE"}
Publish ok
Sending payload: {"Distance":114.95,"Message":"SAFE"}
Publish ok
Sending payload: {"Distance":114.95,"Message":"SAFE"}
Publish ok
Sending payload: {"Distance":114.95,"Message":"SAFE"}
Publish ok
```

SAFE Message In IBM Watson Device Recent Events :

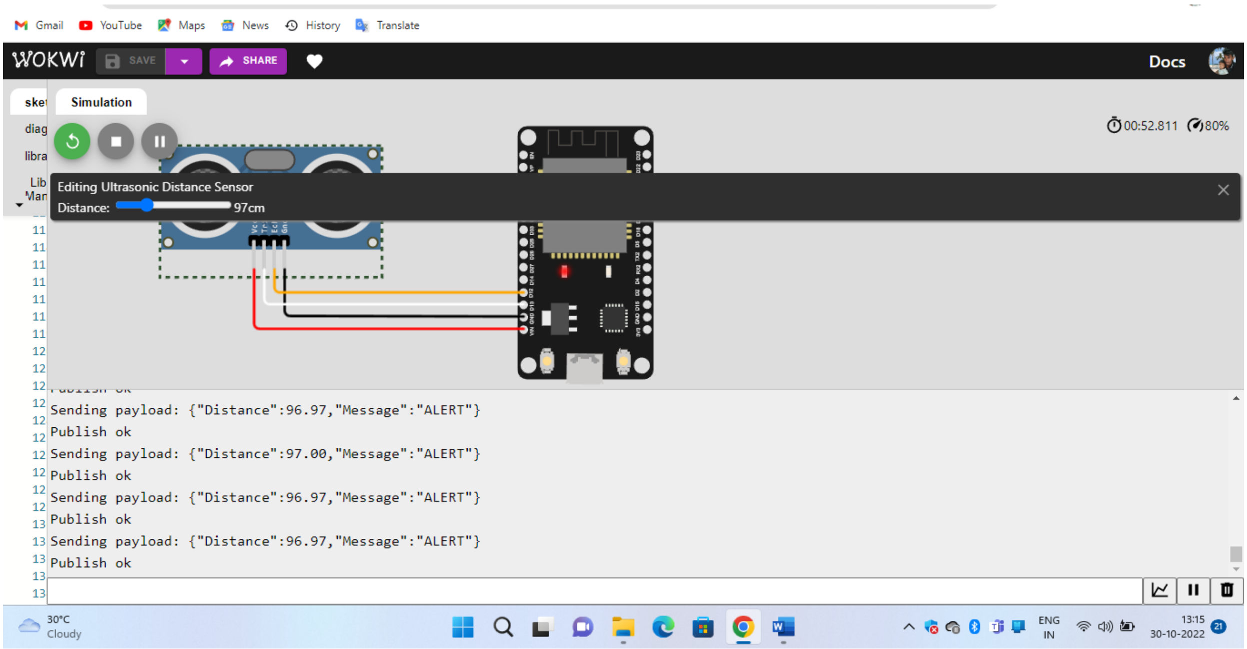


Device ID	Status	Device Type	Class ID	Date Added
b11m3edeviceid	Connected	b11m3edevicetype	Device	Oct 17, 2022 4:53 PM

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

Event	Value	Format	Last Received
Data	{"Distance":114.95,"Message":"SAFE"}	json	a few seconds ago
Data	{"Distance":114.95,"Message":"SAFE"}	json	a few seconds ago
Data	{"Distance":114.95,"Message":"SAFE"}	json	a few seconds ago
Data	{"Distance":114.95,"Message":"SAFE"}	json	a few seconds ago
Data	{"Distance":114.95,"Message":"SAFE"}	json	a few seconds ago

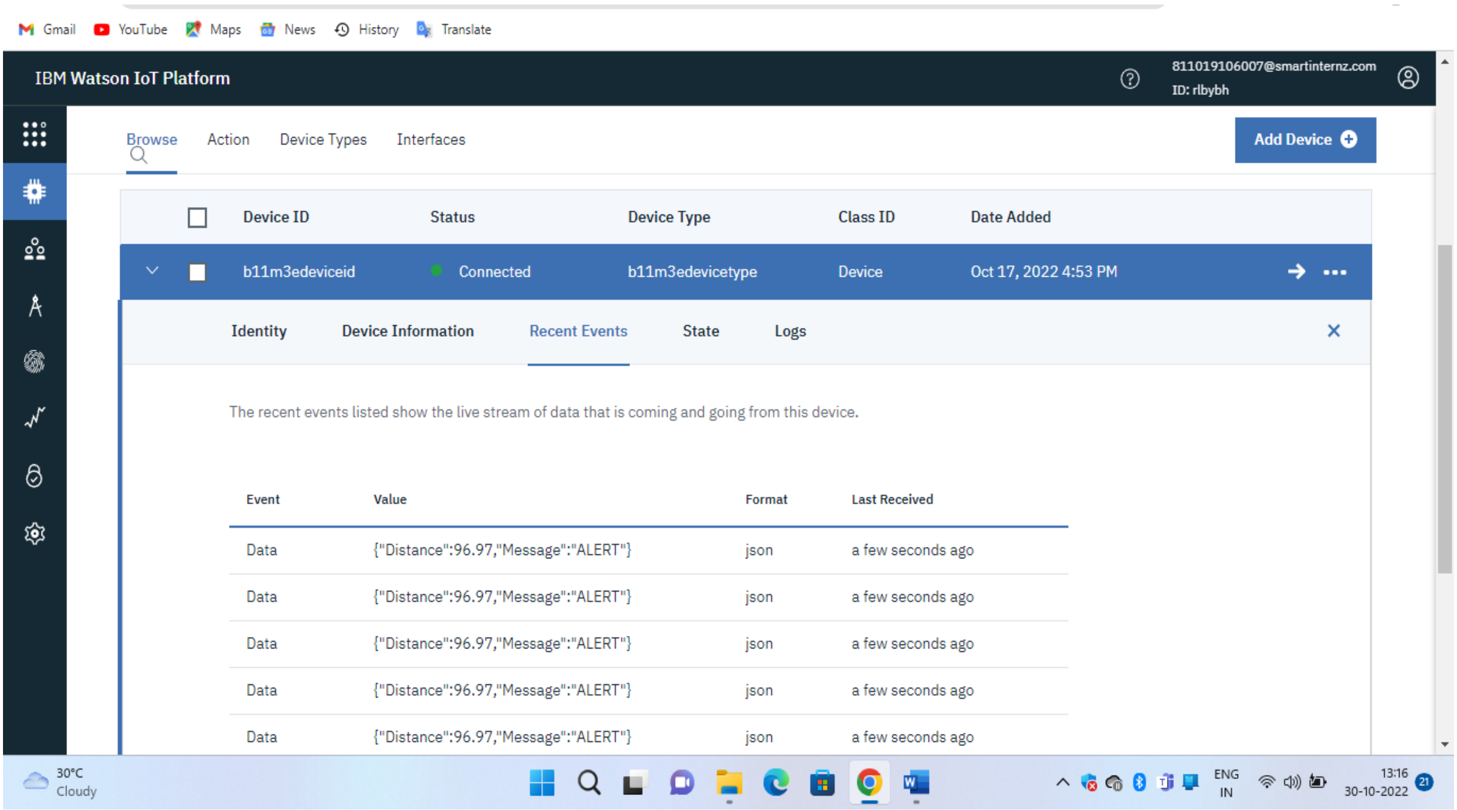
When Distance is Greater than 100 cm it shows “ ALERT”



Editing Ultrasonic Distance Sensor
Distance: 97cm

```
Sending payload: {"Distance":96.97,"Message":"ALERT"}
Publish ok
Sending payload: {"Distance":97.00,"Message":"ALERT"}
Publish ok
Sending payload: {"Distance":96.97,"Message":"ALERT"}
Publish ok
Sending payload: {"Distance":96.97,"Message":"ALERT"}
Publish ok
```

ALERT Message In IBM Watson Device Recent Events :



Device ID	Status	Device Type	Class ID	Date Added
b11m3edeviceid	● Connected	b11m3edevicetype	Device	Oct 17, 2022 4:53 PM

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

Event	Value	Format	Last Received
Data	{"Distance":96.97,"Message":"ALERT"}	json	a few seconds ago
Data	{"Distance":96.97,"Message":"ALERT"}	json	a few seconds ago
Data	{"Distance":96.97,"Message":"ALERT"}	json	a few seconds ago
Data	{"Distance":96.97,"Message":"ALERT"}	json	a few seconds ago
Data	{"Distance":96.97,"Message":"ALERT"}	json	a few seconds ago