# Assignment - 4

Assignment Date	20 October 2022
Student Name	Mr.N.Nithyananthan
Student Roll Number	621319106312
Maximum Marks	2 Marks

## **Question:**

Write a code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100cms send "alert" to ibm cloud and display in device recent events.

### **Solution:**

```
(Link >>> <a href="https://wokwi.com/projects/347240656089907795">https://wokwi.com/projects/347240656089907795</a>)
```

#### **Code in Wokwi:**

```
(sketch.ino)
```

```
#include<WiFi.h>//library for wifi
#include<PubSubClient.h>//library for MQtt
voidcallback(char* subscribetopic, byte* payload, unsignedintpayloadLength);
// credentials of IBM Accounts
#defineORG "n6rl9n"//IBM Organition ID
#defineDEVICE_TYPE "NaaghuIoT"//Device type mentioned in ibm watson IOT Platform
#defineDEVICE_ID "06112002"//Device ID mentioned in ibm watson IOT Platform
#defineTOKEN "98765432"//Token
Stringdata3;
// Customise the above values
charserver[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
charpublishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event
perform and format in which data to be send
charsubscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command type
AND COMMAND IS TEST OF FORMAT STRING
charauthMethod[] = "use-token-auth";// authentication method
chartoken[] = TOKEN;
charclientId[] = "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
constinttrigpin = 5;
constintechopin = 18;
```

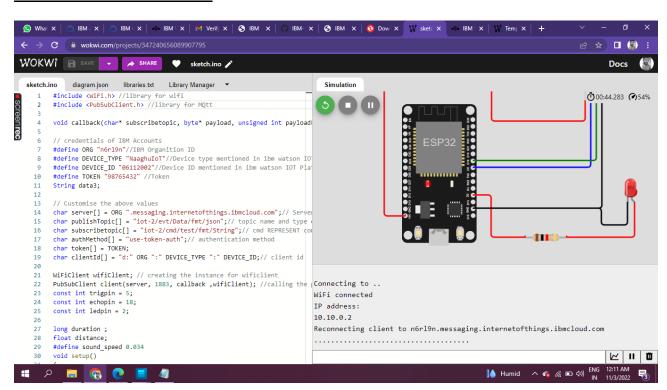
```
constintledpin = 2;
longduration ;
floatdistance;
#definesound speed 0.034
voidsetup()
{
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, OUTPUT);
  pinMode(ledpin, OUTPUT);
  wificonnect();
  mqttconnect();
}
voidloop()
  digitalWrite(trigpin, LOW);
  digitalWrite(trigpin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin, LOW);
  duration= pulseIn(echopin,HIGH);
  distance = duration * sound_speed /2;
  if(distance<=100)</pre>
  {
    PublishData(distance);
    delay(1000);
    if(!client.loop())
      mqttconnect();
    digitalWrite(ledpin, HIGH);
    Serial.println("Alert !!");
    Serial.println(distance);
  }
  else
  {
    digitalWrite(ledpin, LOW);
  }
  delay(10); // this speeds up the simulation
}
// Retrieving to Cloud
voidPublishData(floatdistance)
  mqttconnect();// Function call for connecting to ibm
  // creating the String in in form JSon to update the data to ibm cloud
  Stringpayload = "{\"Alert !! \": ";
  payload += distance;
  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");
  }
}
voidmqttconnect()
  if(!client.connected())
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while(!!!client.connect(clientId, authMethod, token))
      Serial.print(".");
      delay(500);
    initManagedDevice();
    Serial.println();
}
voidwificonnect() // Function defination for wificonnect
  Serial.println();
 Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);// Passing the wifi credentials to establish the
  while(WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
voidinitManagedDevice()
  if(client.subscribe(subscribetopic))
    Serial.println((subscribetopic));
   Serial.println("subscribe to cmd OK");
  }
  else
  {
```

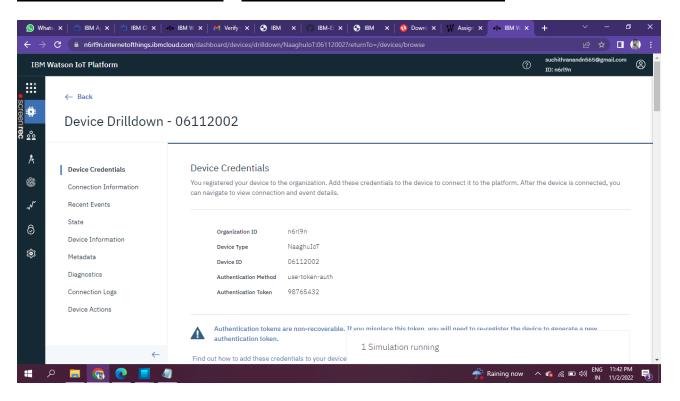
```
Serial.println("subscribe to cmd FAILED");
  }
}
voidcallback(char* subscribetopic, byte* payload, unsignedintpayloadLength)
  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for(inti = 0; i < payloadLength; i++)</pre>
    //Serial.print((char)payload[i]);
   data3 += (char)payload[i];
  Serial.println("data: "+ data3);
  if(data3=="lighton")
  {
      Serial.println(data3);
  }
  else
  {
      Serial.println(data3);
data3="";
(diagram.json)
  "version": 1,
  "author": "312. Nithyananthan N",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 35.33, "left": -109.33,
"attrs": {} },
      "type": "wokwi-hc-sr04",
      "id": "ultrasonic1",
      "top": -61.7,
      "left": 78.5,
      "attrs": { "distance": "164"}
    },
      "type": "wokwi-led",
      "id": "led1",
      "top": 134.3,
      "left": 195.16,
      "attrs": { "color": "red"}
    },
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 214.96,
      "left": 65.17,
      "attrs": { "value": "1000"}
    }
```

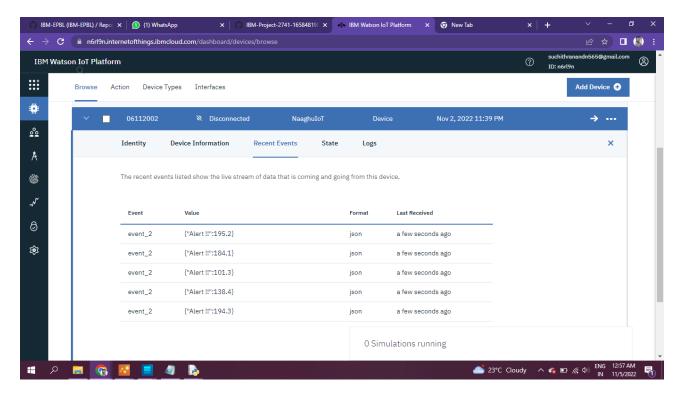
#### Wokwi Sketch and Simulation:

PubSubClient



## Add a new device in IBM Cloud & Output in IBM Cloud (Watson Platform):





#### **Conclusion:**

Whenever the distance is less than 100 cms send an "Alert" to the IBM cloud and display in the device **Recent Events**.