

ASSIGNMENT 4

Team ID	PNT2022TMID44989
Name	M.Srikanth
Project Name	Real-Time River Water Quality Monitoring and Control System
Marks	2marks

Assignment 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.

Program Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
#define ORG "s2qhvm"
#define DEVICE_TYPE "Laptop"
#define DEVICE_ID "0410"
#define TOKEN "20011004"
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/event8/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034 long duration;
float distance;
```

```

void setup()
{
    Serial.begin(115200);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    wificonnect();
    mqttconnect();
}

void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH);
    distance = duration * SOUND_SPEED/2;
    Serial.print("Distance (cm): ");
    Serial.println(distance);
    if(distance<100)
    {
        Serial.println("ALERT!!");
        delay(1000);
        PublishData(distance);
        delay(1000);
        if (!client.loop())
        { mqttconnect();
        }
    }
    delay(1000);
}

void PublishData(float dist)
{
    mqttconnect();
    String payload = "{\"Distance\": "; payload += dist; payload +=
    ", \"ALERT!!\": \"\"Distance less than 100cms\"\""; payload += "}";
    Serial.print("Sending payload: ");
    Serial.println(payload);
    if (client.publish(publishTopic, (char*) payload.c_str()))
    { Serial.println("Publish ok");
    }
    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()
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);
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++)
  {
    data3 += (char)payload[i];
  }
  Serial.println("data: " + data3);
  data3="";
}

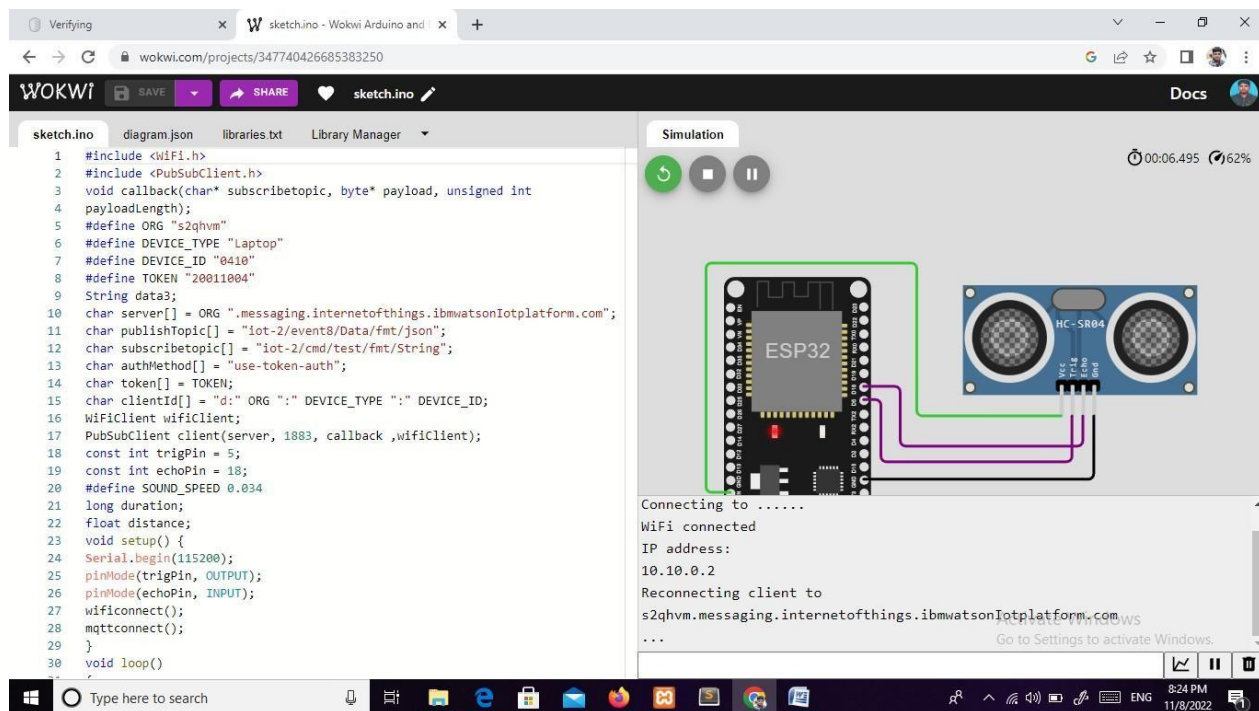
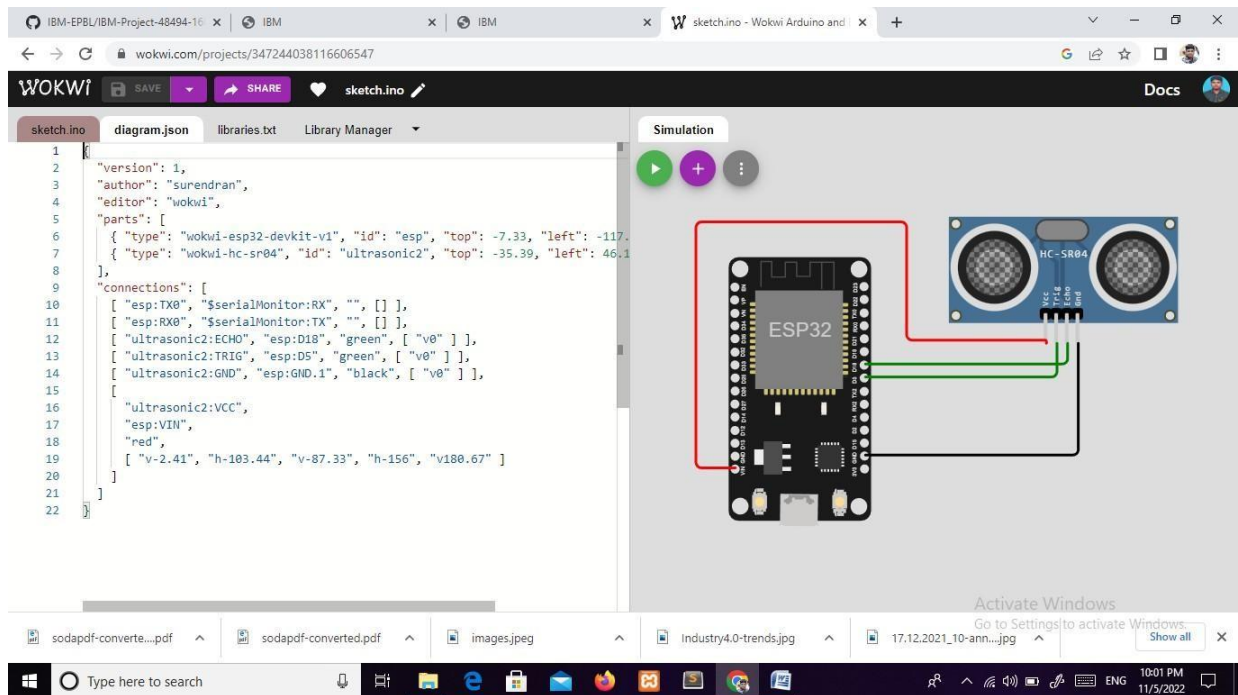
```

Diagram.json:

```

{
  "version": 1,
  "author": "surendran",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -7.33, "left": -117.34, "attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic2", "top": -35.39, "left": 46.16, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "ultrasonic2:ECHO", "esp:D18", "green", [ "v0" ] ],
    [ "ultrasonic2:TRIG", "esp:D5", "green", [ "v0" ] ],
    [ "ultrasonic2:GND", "esp:GND.1", "black", [ "v0" ] ],
    [
      "ultrasonic2:VCC",
      "esp:VIN",
      "red",
      [ "v-2.41", "h-103.44", "v-87.33", "h-156", "v180.67" ]
    ]
  ]
}

```



The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area shows details for a device named '0410', which is 'Disconnected' and identified as a 'Laptop'. The 'Recent Events' tab is active, displaying a table of events. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. Below the table, a status indicator shows '1 Simulation running'. An 'Activate Windows' watermark is visible in the bottom right corner of the browser window.

Event	Value	Format	Last Received
event8	{"distance":132}	json	a few seconds ago
event8	{"distance":193}	json	a few seconds ago
event8	{"distance":27}	json	a few seconds ago
event8	{"distance":80}	json	2 minutes ago

Reference link: <https://wokwi.com/projects/347740426685383250>