

## ASSIGNMENT – 4

### Ultrasonic sensor simulation in Wokwi

|               |  |
|---------------|--|
| Date          | 04 NOVEMBER 2022   |
| Team ID       | PNT2022TMID36354   |
| Project Name  | Real-Time River Water Quality Monitoring and Control System. |
| Maximum Marks | 2 Marks  |

#### QUESTIONS:

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

#### CODE:

```
#include <WiFi.h> #include
<PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "kotoq5"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/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("Reconnect
ing client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print("."); delay(500);
}
initManagedDevice();

```

```

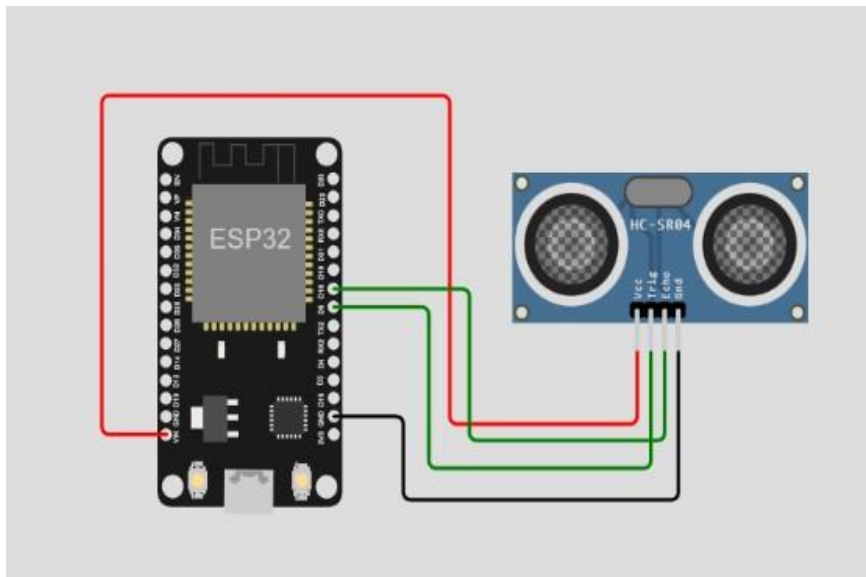
Serial.println();
} }
void wificonnect()
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6); while
(WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
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++) {
//Serial.print((char)payload[i]); data3 +=
(char)payload[i];
}
Serial.println("data: "+ data3); data3="";
}

```

**.json CODE:**

```
1 {
2   "version": 1,
3   "author": "sweetysharon",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -4.67, "left": -114.67, "attrs": {} },
7     { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": 15.96, "left": 89.17, "attrs": {} }
8   ],
9   "connections": [
10    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
11    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
12    [
13      "esp:VIN",
14      "ultrasonic1:VCC",
15      "red",
16      [ "h-37.16", "v-178.79", "h200", "v173.33", "h100.67" ]
17    ],
18    [ "esp:GND.1", "ultrasonic1:GND", "black", [ "h39.87", "v44.04", "h170" ] ],
19    [ "esp:D5", "ultrasonic1:TRIG", "green", [ "h54.54", "v85.07", "h130.67" ] ],
20    [ "esp:D18", "ultrasonic1:ECHO", "green", [ "h77.87", "v80.01", "h110" ] ]
21  ]
22 }
```

## CIRCUIT DIAGRAM:



Wokwi simulation link:

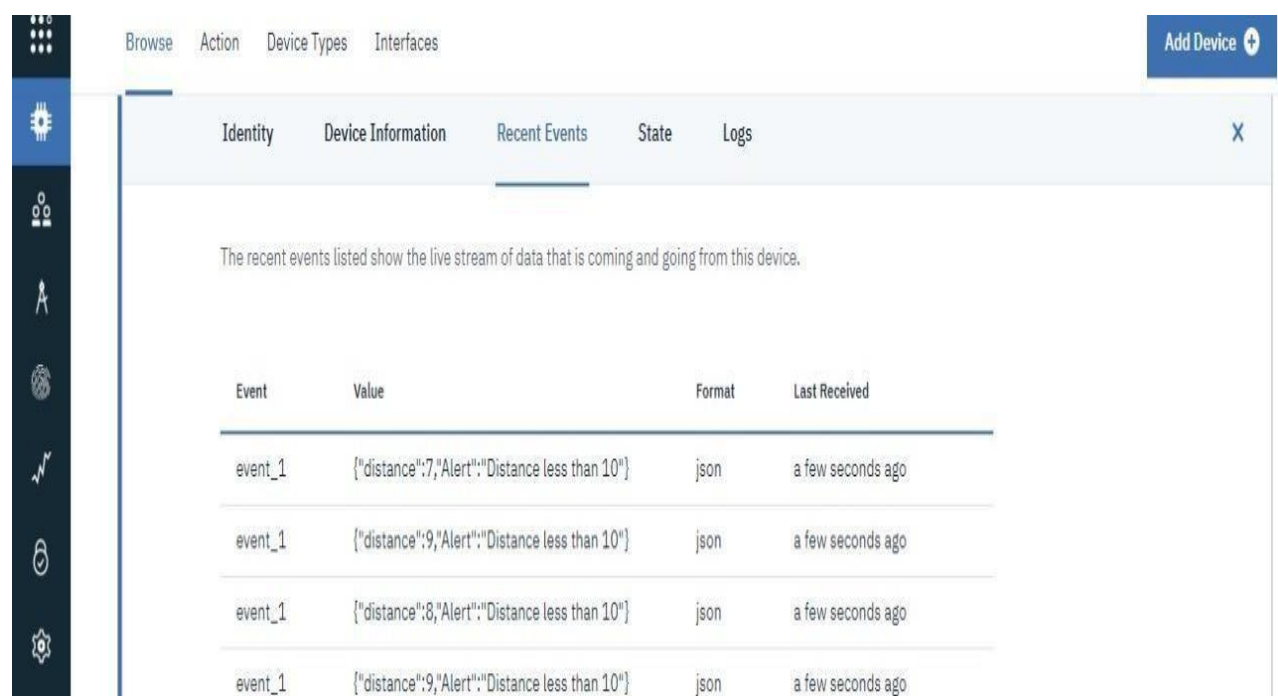
<https://wokwi.com/projects/346404308518961748>

WOKWI OUTPUT:

```
Connecting to ....
WiFi connected
IP address:
10.10.0.2
Reconnecting client to ytluse.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe to cmd OK

Distance (cm): 399.92
Distance (cm): 399.96
Distance (cm): 399.94
Distance (cm): 399.98
Distance (cm): 399.94
Distance (cm): 399.92
Distance (cm): 399.94
```

## IBM CLOUD OUTPUT:



The screenshot displays the IBM Cloud IoT Platform interface. On the left is a dark sidebar with various icons. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A blue 'Add Device' button with a plus icon is on the right. Below the navigation bar, a tabbed interface shows 'Identity', 'Device Information', 'Recent Events' (selected), 'State', and 'Logs'. A close button 'X' is on the right of the tabs. The main content area has a heading 'The recent events listed show the live stream of data that is coming and going from this device.' Below this is a table with four columns: 'Event', 'Value', 'Format', and 'Last Received'. The table contains four rows of data, all with 'event\_1' in the 'Event' column and 'a few seconds ago' in the 'Last Received' column.

| Event   | Value  | Format | Last Received     |
|---------|--|--------|-------------------|
| event_1 | {"distance":7,"Alert":"Distance less than 10"} | json   | a few seconds ago |
| event_1 | {"distance":9,"Alert":"Distance less than 10"} | json   | a few seconds ago |
| event_1 | {"distance":8,"Alert":"Distance less than 10"} | json   | a few seconds ago |
| event_1 | {"distance":9,"Alert":"Distance less than 10"} | json   | a few seconds ago |