

ASSIGNMENT-4

Ultrasonic sensor simulation in Wokwi

Team ID		PNT2022TM11782	
Title	Gas Leakage Monitoring and Alerting System		
Name	Rohith kumar S		
Minimum Marks	2 Marks		

Question:

Write a code and connections in Wokwi for the Ultrasonic sensor.
Whenever the distance is less than 100cms send an “Alert” to 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 "dl9wub"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "3C-91-80-49-01-C9"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "cE&QcASnabqYe18-1f" //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(100);
PublishData(distance);
delay(100);
if (!client.loop())
{mqttconnect();
}
}
delay(100);
}
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(100);
}
initManagedDevice();
Serial.println();
}
}
void wificonnect()
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED)
{delay(100);
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="";
}
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "dl9wub"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "3C-91-80-49-01-C9"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "cE&QcASnabqYe18-1f" //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(100);
  PublishData(distance);
  delay(100);
  if (!client.loop())
  {mqttconnect();
  }
}
delay(100);
}
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(100);
}
initManagedDevice();
Serial.println();
}
}

void wificonnect()
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED)
  {delay(100);
  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="";
}

```

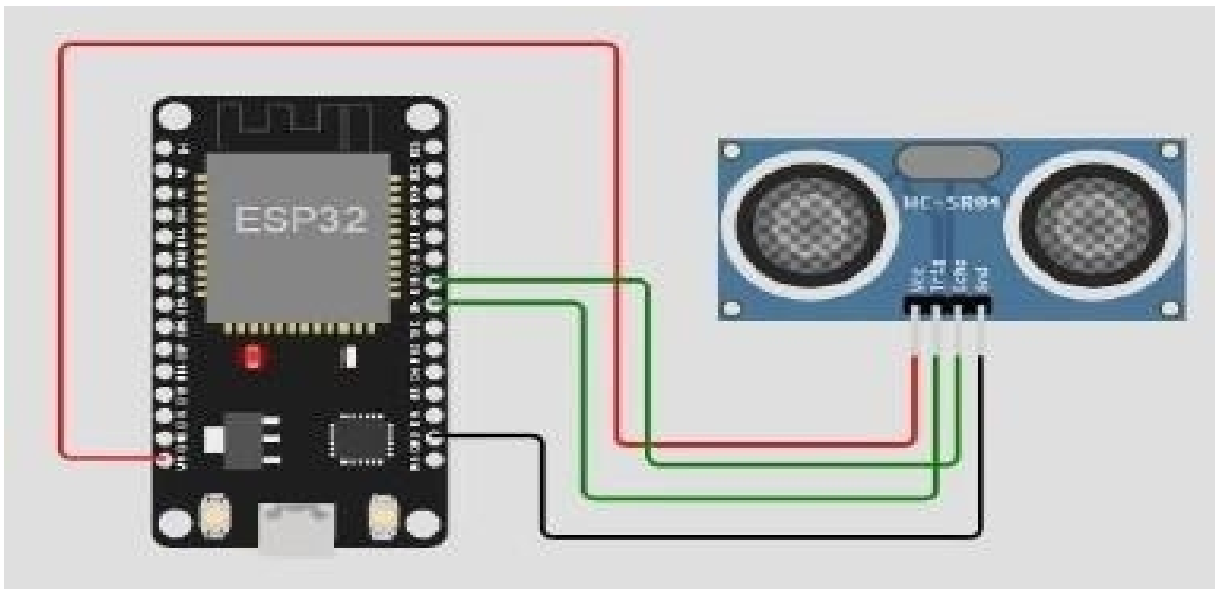
Diagram.json:

```

{
  "version": 1,
  "author": "sweetysharon",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -4.67, "left": -112.87, "attrs":
  {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": 15.96, "left": 89.17, "attrs":
  {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [ ] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [ ] ],
    [
      "esp:VIN",
      "ultrasonic1:VCC",
      "red",
      [ "h-37.16", "v-178.79", "h200", "v173.33", "h100.67" ]
    ],
    [ "esp:GND.1", "ultrasonic1:GND", "black", [ "h39.87", "v44.04", "h170" ] ],
    [ "esp:D5", "ultrasonic1:TRIG", "green", [ "h54.54", "v85.07", "h130.67" ] ],
    [ "esp:D18", "ultrasonic1:ECHO", "green", [ "h77.87", "v80.01", "h110" ] ]
  ]
}

```

Circuit Diagram:



Output:

```
Connecting to .....  
WiFi connected  
IP address:  
10.10.0.2  
Reconnecting client to d19wub.messaging.internetofthings.ibmcloud.com  
iot-2/cmd/test/fmt/String
```

IBM Cloud Output:

The screenshot displays the IBM Watson IoT Platform dashboard in a web browser. The browser's address bar shows the URL `d19wub.internetofthings.ibmcloud.com/dashboard/devices/browse`. The dashboard header includes the user email `913019104028@smartinternz.com` and the ID `ID: d19wub`. A sidebar on the left contains navigation icons for various platform features. The main content area has tabs for `Browse`, `Action`, `Device Types`, and `Interfaces`. The `Browse` tab is active, showing a table of device data. The table has columns for `Property`, `Value`, `Type`, `Event`, and `Last Received`. It displays two rows of data: one for `Distance` (399.96) and one for `ALERT!!` (Distance less than 100cms). Below the table, it indicates `Items per page: 50` and `1-1 of 1 item`. A status box at the bottom right of the dashboard area shows `0 Simulations running`. The Windows taskbar at the bottom of the screen shows the system time as `02:08` on `31-10-2022`.

IBM Watson IoT Platform

913019104028@smartinternz.com
ID: d19wub

Add Device +

Browse Action Device Types Interfaces

Identity Device Information Recent Events State Logs

Showing Raw Data | No Interfaces Available

Property	Value	Type	Event	Last Received
Distance	399.96	Number	Data	a few seconds ago
ALERT!!	Distance less than 100cms	String	Data	a few seconds ago

Items per page: 50 | 1-1 of 1 item

1 of 1 page

0 Simulations running

Type here to search

02:08
31-10-2022