

# ASSIGNMENT-4

## UltrasonicsensorsimulationinWokwi

TeamID		PNT2022TM11782
Title	GasLeakageMonitoringandAlertingSystem	
Name	Tamilarasan V	
MinimumMarks	2Marks	

### Question:

Write a code and connections in Wokwi for the Ultrasonic sensor. Whenever the distance is less than 100cm send an "Alert" to IBM cloud and display in the device recent events

### Code:

```
#include<WiFi.h>#include<PubSubClient.h>
void callback(char* subscribtopic, byte* payload, unsigned intpayloadLength);
//-----credentialsofIBMAccounts-----
#defineORG"d19wub"/IBMORGANITIONID
#defineDEVICE_TYPE"ESP32"//DevicetypementionedinibmwatsonIOTPlatform
#define DEVICE_ID "3C-91-80-49-01-C9"//Device ID mentioned in ibm watson IOT Platform
#defineTOKEN"cE&QcASnabqYe18-1f"//Token
Stringdata3;
char server[] = ORG
".messaging.internetofthings.ibmcloud.com";charpublishTopic[]="iot-2/evt/Data/fmt/json";
char subscribtopic[] = "iot-2/cmd/test/fmt/String";charauthMethod[]="use-token-auth";
chartoken[]=TOKEN;
charclientId[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID;
WiFiClientwifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);constinttrigPin=5;
const int echoPin = 18;#define
SOUND_SPEED 0.034longduration;
float
distance;voidsetup()
{
  Serial.begin(115200);pinMode(trigPin,OUTPUT);pinMode(echoPin,INPUT);wificonnect();mqttconnect();
}
voidloop()
```

```

{
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("Sendingpayload:");
Serial.println(payload);

if(client.publish(publishTopic,(char*)payload.c_str())){
Serial.println("Publishok");
} else{
Serial.println("Publishfailed");
}
}
voidmqttconnect(){
if (!client.connected())
{Serial.print("Reconnectingclientto");Serial.println(server);
while(!client.connect(clientId,authMethod,token)){
Serial.print(".");delay(100);
}
}
initManagedDevice();
Serial.println();
}
}
voidwificonnect()
{
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());
}
voidinitManagedDevice(){
if(client.subscribe(subscribetopic)){

```

```

Serial.println(subscribetopic);Serial.println("subscribetocmdOK");
} else{
Serial.println("subscribetocmdFAILED");
}
}
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
Serial.print("callbackinvokedfortopic:");
Serial.println(subscribetopic);
for(int i=0; i<payloadLength; i++){
//Serial.print((char)payload[i]);data3+=(char)p
ayload[i];
}
Serial.println("data: " + data3);data3="";
}
#include<WiFi.h>#include<PubSub
bClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
//-----credentialsofIBMAccounts-----
#define ORG"d19wub"//IBMORGANITIONID
#define DEVICE_TYPE"ESP32"//DevicetypementionedinibmwatsonIOTPlatform
#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 0.034long duration;
float
distance;void setup()
{
Serial.begin(115200);pinMode(tri
gPin,OUTPUT);pinMode(echoPin,
INPUT);wifiConnect();mqttConne
ct();
}
void loop()
{
digitalWrite(trigPin,
LOW);delayMicroseconds(2);digitalWrite(trigPin,
HIGH);delayMicroseconds(10);digitalWrite(trigPi
n,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("Sendingpayload:");
Serial.println(payload);

if(client.publish(publishTopic,(char*)payload.c_str())){
  Serial.println("Publishok");
} else{
  Serial.println("Publishfailed");
}
}
voidmqttconnect(){
if (!client.connected())
{Serial.print("Reconnectingclientto");Serial.println(server);
while(!client.connect(clientId,authMethod,token)){
  Serial.print(".");delay(100);
}
initManagedDevice();
Serial.println();
}
}
voidwificonnect()
{
  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());
}
voidinitManagedDevice(){
if (client.subscribe(subscribetopic))
{Serial.println((subscribetopic));Serial.println("subscr
ibetocmdOK");

```

```

} else{
Serial.println("subscribetocmdFAILED");
}
}
void callback(char*subscribetopic,byte*payload,unsignedintpayloadLength)
{
Serial.print("callbackinvokedfortopic:");
Serial.println(subscribetopic);
for(inti=0;i<payloadLength;i++){
//Serial.print((char)payload[i]);data3+=(char)p
ayload[i];
}
Serial.println("data: "+ data3);data3="";
}

```

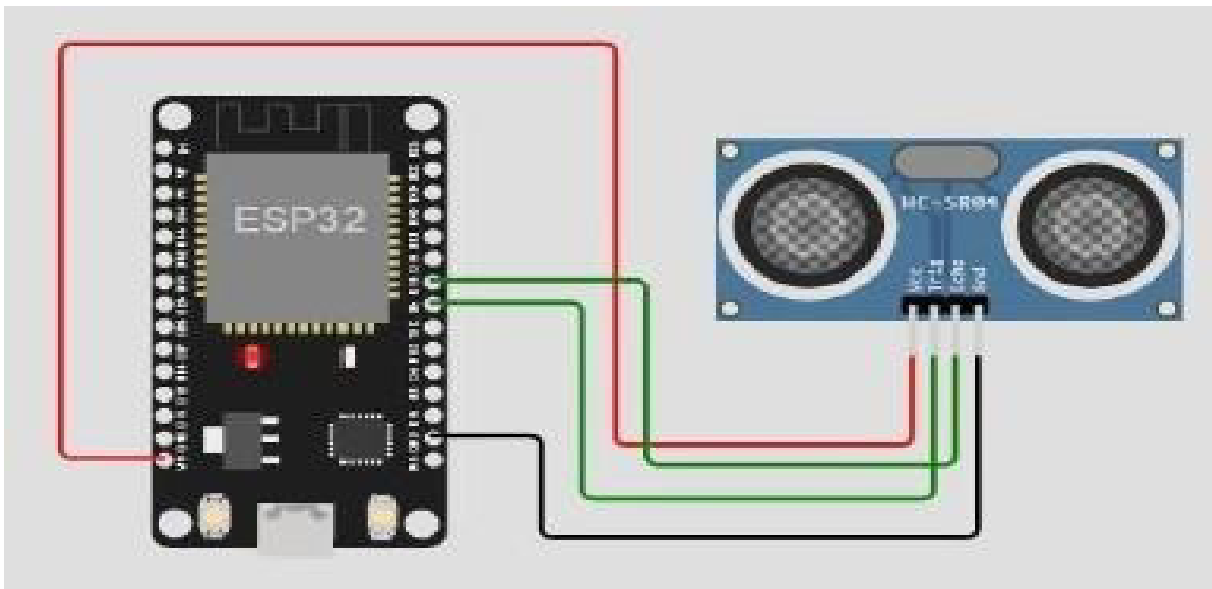
## Diagram.json:

```

{
  "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:VC
    C","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"]
  ]
}

```

## CircuitDiagram:



## 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
```

# IBMCloudOutput:

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 IBM Watson IoT Platform logo and a user profile for `913019104028@smartinternz.com` with ID `d19wub`. The main navigation bar has tabs for `Browse`, `Action`, `Device Types`, and `Interfaces`, along with an `Add Device` button. The `Browse` tab is active, showing a table of device data. The table has columns for `Property`, `Value`, `Type`, `Event`, and `Last Received`. The data shows a `Distance` of `399.96` (Number) and an `ALERT!!` (String) with the value `Distance less than 100cms`. The `Last Received` for both is `a few seconds ago`. The table is titled `Showing Raw Data | No Interfaces Available`. Below the table, it indicates `Items per page: 50` and `1 of 1 page`. At the bottom of the dashboard, a box states `0 Simulations running`. The Windows taskbar at the bottom shows the search bar and various application icons, with the system clock displaying `02:08 31-10-2022`.

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