

## Assingment 4

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. Upload document with wokwi share link and images of IBM cloud

Assignment Date	November 11 2022
Student Name	BOOPATHI S
Student Roll Number	510419104018
Maximum Marks	2 MARKS

### CODE :

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "ytluse"//IBM ORGANITION ID
#define DEVICE_TYPE "2702"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "0+n)Eh+1NX0y3?rG!8" //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("Reconnecting client to ");

```

```

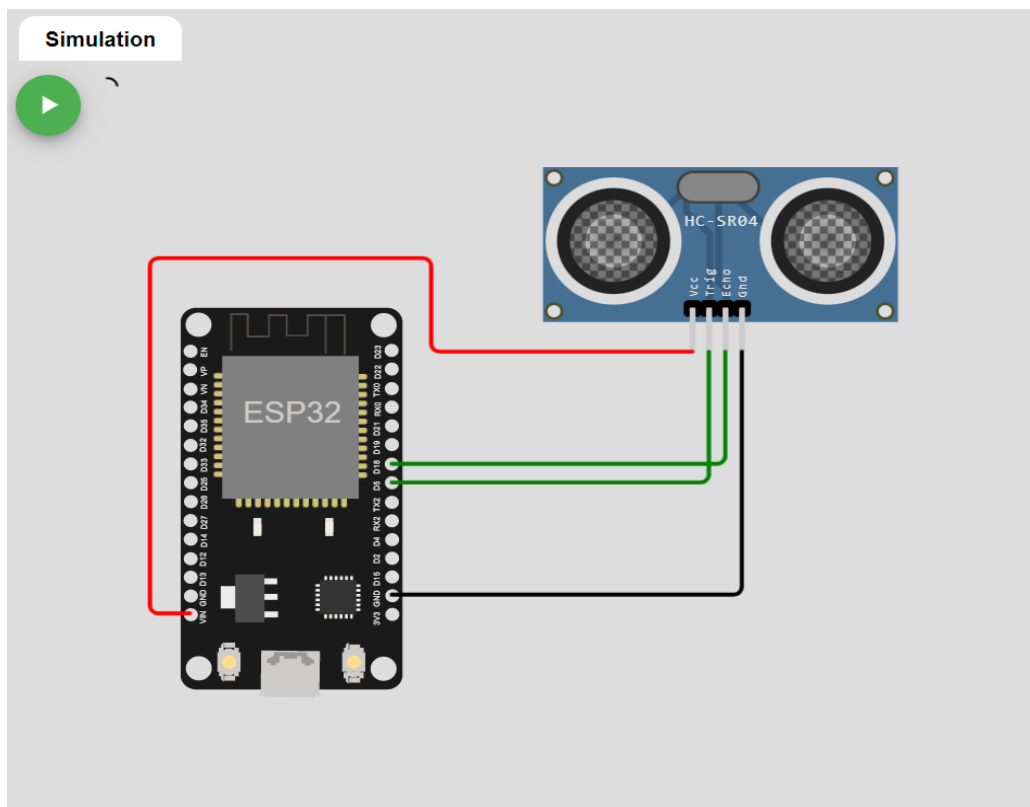
Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void wificonnect()
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
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++) {
//Serial.print((char)payload[i]);
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
data3="";
}

```

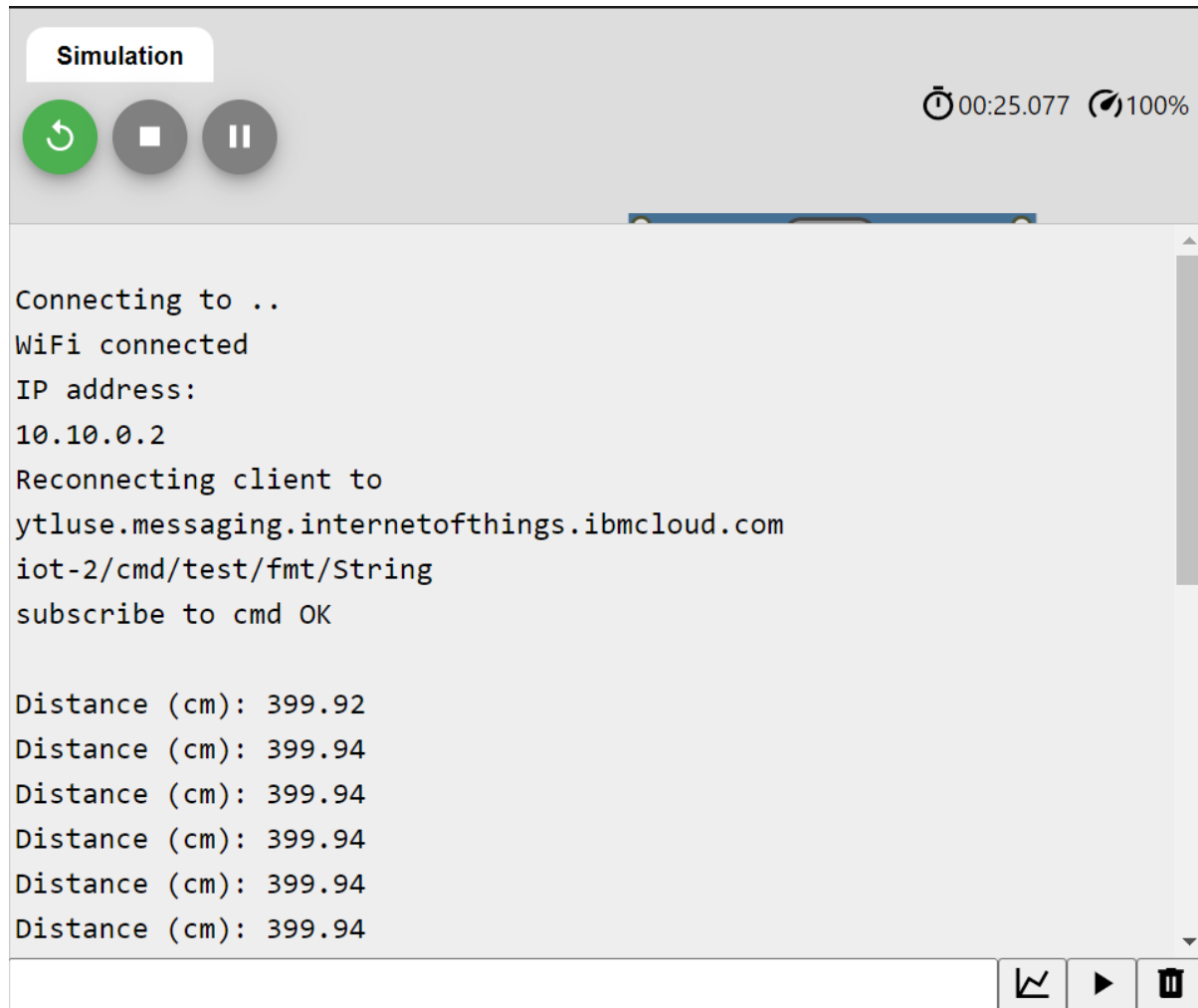
## Diagram.json:

```
{
  "version": 1,
  "author": "SANDEEP I",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -81.7, "left": -168.2,
    "attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -152.1, "left": 5.02,
    "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [ ] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [ ] ],
    [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v124.8", "h-167.67" ] ],
    [ "ultrasonic1:ECHO", "esp:D18", "green", [ "v0" ] ],
    [ "ultrasonic1:TRIG", "esp:D5", "green", [ "v0" ] ],
    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v0", "h-124.8", "v-48", "h-134.4",
    "v182.4" ] ]
  ]
}
```

## Circuit Diagram:



## Output:



The screenshot shows a Wokwi simulation window. At the top, there is a 'Simulation' tab. Below it, on the left, are three circular buttons: a green one with a circular arrow (refresh), a grey one with a square (stop), and a grey one with two vertical bars (pause). On the right, there is a clock icon showing '00:25.077' and a refresh icon with '100%'. The main area is a terminal window with a light grey background and a vertical scrollbar on the right. It contains the following text:

```
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): 399.94  
Distance (cm): 399.94  
Distance (cm): 399.94  
Distance (cm): 399.94  
Distance (cm): 399.94
```

At the bottom right of the terminal window, there are three icons: a line graph (plot), a play button (run), and a trash can (clear).

## Wokwi simulation link:

<https://wokwi.com/projects/347656637677306451>

## IBM Cloud Output:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar labeled 'Search by Device ID' is present. The main content area shows a table of devices. The first device, ID 12, is in a 'Disconnected' state. Below the table, the 'Recent Events' tab is selected, showing a live stream of data. The events are 'eventflow' messages containing JSON data with random numbers, temperature, and humidity. A status box at the bottom right indicates '1 Simulation running'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12	Disconnected	123	Device	Nov 10, 2022 7:26 PM	

Event	Value	Format	Last Received
eventflow	{"randomNumber":49,"temp":95,"Humid":81}	json	a few seconds ago
eventflow	{"randomNumber":64,"temp":96,"Humid":94}	json	a few seconds ago
eventflow	{"randomNumber":60,"temp":99,"Humid":61}	json	a few seconds ago

1 Simulation running