

# ASSIGNMENT-4

**NAME : ABEL FRANCIS**

**REG.NO : 110719106001**

## OBJECTIVE

Write Code and connections in wokwi for ultrasonic sensor. whatever distance is less than 100 cms send “Alert” to ibm cloud aand display indevicerecent events.

## SOLUTION

```
//Pins
```

```
const int TRIG_PIN = 7 ;
```

```
const int ECHO_PIN = 8;
```

```
//Anything over 400 cm (23200 us pulse) is "out of range"
```

```
const unsigned int MAX_DIST = 23200;
```

```
void setup() {
```

```
// The Trigger pin will tell the sensor to range find
```

```
Pin Mode(TRIG_PIN, OUTPUT);
```

```
digital Write(TRIG_PIN, LOW);
```

```
//Set Echo pin as input to measure the duration of
//pulses coming back from the distance sensor
pinMode(ECHO_PIN, INPUT ) ;

// We'll use the serial monitor to view the sensor output
Serial.begin(9600);

}

void loop()
{ unsigned long
  t1;
  unsigned long t2;
  unsigned long pulse_width;
  float cm;
  float inches;

  // Hold the trigger pin high for at least 10 us
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);

  digitalWrite(TRIG_PIN, LOW);

  // Wait for pulse on echo pin
  while (digitalRead( ECHO_PIN )==0 );

  // Measure how long the echo pin was held high (pulse
  width)

  // Note: the micros() counter will overflow after-70 min
  t1= micros ();
```

```

while (digitalRead(ECHO_PIN) == 1);
t2= micros ();
pulse_width = t2-t1;
// Calculate distance in centimeters and inches. The
constants
//are found in the datasheet, and calculated from the
assumed speed
// of sound in air at sea level (- 340m/s)
cm=pulse_Width / 58 ;
inches = pulse_width/148.0;
// Print out results
if (pulse_width >MAX _ DIST ){Serial.println("Out of
range"); }
else {
Serial.println("*****");
Serial.print("The Measured Distance in cm: ");
Serial.println(cm);
if( cm < 100 )
{ //while(true)
{ Serial.println("Alert!!"); //
}
}

```

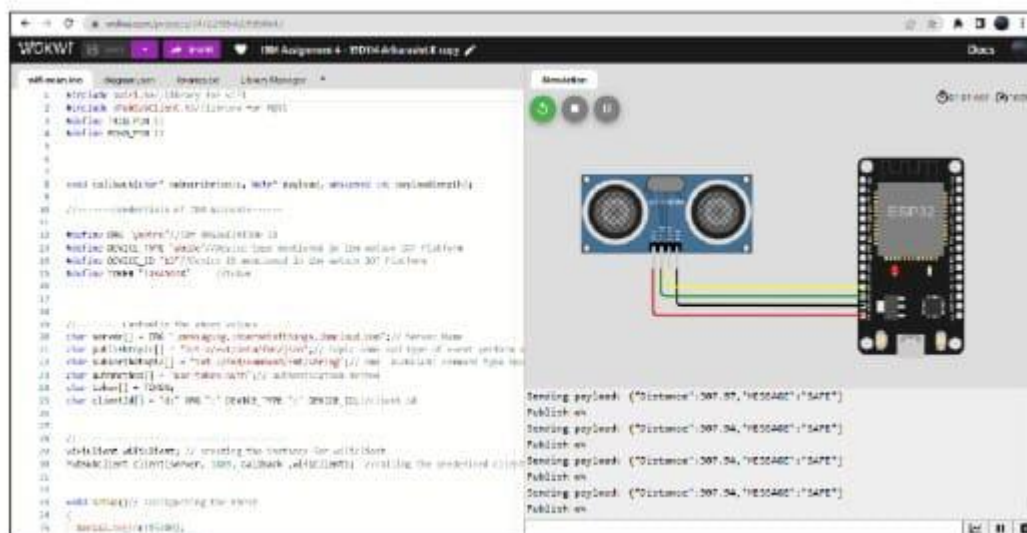
```
Serial.print("*****"); }
```

```
//wait at least 1000ms before next measurement
Delay(1000);
```

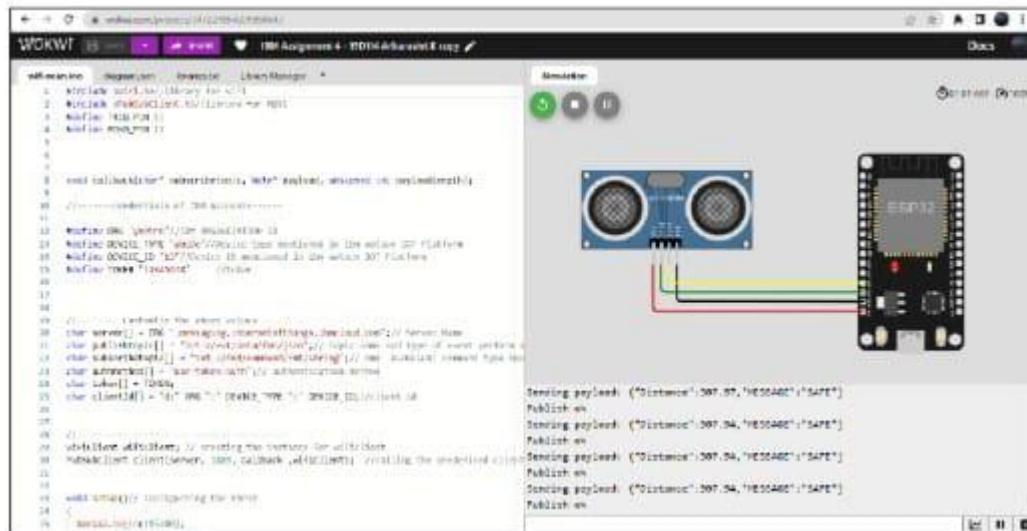
$$\}$$

## OUTPUT

**1.If the distance is less than 100cm, it alerts.**



**2. If the distance is more than 100 cm, it won't alert.**



### 3. Simulation and code execution

IBM cloud output:

Identity	Device Information	Recent Events	State	Logs
The recent events listed show the live stream of data that is coming and going from this device.				
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":5,"Alert":"Distance less than 10"]	json	a few seconds ago	
event_1	["distance":5,"Alert":"Distance less than 10"]	json	a few seconds ago	

WOKWI

SAVE

SHARE

Docs

SIGN UP

sketch.ino

diagram.json

libraries.txt

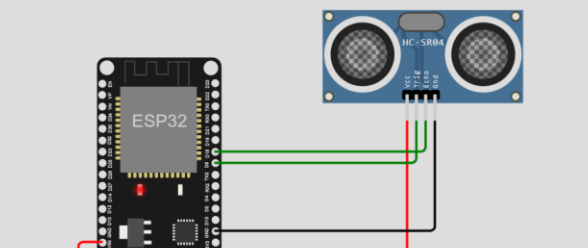
Library Manager

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribtopic, byte* payload, unsigned int
4 payloadlength);
5 //-----credentials of IBM Accounts-----
6 #define ORG "olxobn"//IBM ORGANITION ID
7 #define DEVICE_TYPE "ESP32PROJECT"//Device type mentioned in ibm watson IOT Platform
8 #define DEVICE_ID "ESP32"//Device ID mentioned in ibm watson IOT Platform
9 #define TOKEN "ESP32PROJECT" //Token
10 String data3;
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/data/fmt/json";
13 char subscribtopic[] = "iot-2/cmd/test/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 WiFiClient wificlient;
18 PubSubClient client(server, 1883, callback ,wificlient);
19 const int trigPin = 5;
20 const int echoPin = 18;
21 #define SOUND_SPEED 0.034
22 long duration;
23 float distance;
24 void setup() {
25   Serial.begin(115200);
26   pinMode(trigPin, OUTPUT);
27   pinMode(echoPin, INPUT);
28   wificlient.connect();
29   mqttconnect();
30 }
31 void loop()
32 {
33   digitalWrite(trigPin, LOW);
34   delayMicroseconds(2);
35   digitalWrite(trigPin, HIGH);
36   delayMicroseconds(10);
37   digitalWrite(trigPin, LOW);
38   duration = pulseIn(echoPin, HIGH);
39   distance = (duration/2) /SOUND_SPEED;
40   Serial.print("Distance (cm): ");
41   Serial.println(distance);
42   client.publish(publishTopic, String(distance));
43   delay(1000);
44 }
```

Simulation

00:15.255 98%

Restart the simulation



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