

#### ASSIGNMENT-4

NAME: J.JAYA SANTHIA

Qn: 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

#### LINK:

<https://wokwi.com/projects/new/arduino-uno>

#### CODE:

```
long          int
echoPin=7;    long
int trigPin=5; void
setup()

    Serial.begin(9600);
    pinMode(7,OUTPUT);
    pinMode(5,INPUT);
}

void loop()

    int duration, distance;
    digitalWrite(3,LOW);
    delayMicroseconds(2);
    digitalWrite(3,HIGH);
    delayMicroseconds(10);
    digitalWrite(3,LOW);

    duration=pulseIn(7,HIGH);
    distance=(duration*0.034/
    2); delay(1000); if
    (distance<=100)
```

```
Serial.println("ALERT!!");
```

## SIMULATIONS:

WOKWI

Docs

The screenshot shows the Wokwi IDE interface. On the left, a C++ sketch is displayed in the editor. The sketch initializes an Arduino Uno with two ultrasonic sensors (HC-SR04) connected to pins 7, 8, and 5. The `loop()` function calculates the distance to the nearest object using the sensors. If the distance is less than or equal to 100 cm, it prints "ALERT!!" to the serial monitor.

```
4 {  
5   Serial.begin(9600);  
6   pinMode(7, OUTPUT);  
7   pinMode(5, INPUT);  
8 }  
9  
10  
11  
12 void loop()  
13 {  
14   int duration, distance;  
15   digitalWrite(3, LOW);  
16   delayMicroseconds(2);  
17   digitalWrite(3, HIGH);  
18   delayMicroseconds(10);  
19   digitalWrite(3, LOW);  
20  
21   duration = pulseIn(7, HIGH);  
22   distance = (duration * 0.034 / 2);  
23   delay(1000);  
24   if (distance <= 100)  
25   {  
26     Serial.println("ALERT!!");  
27   }  
28 }
```

On the right, a simulation of the Arduino Uno is shown with two ultrasonic sensors connected to it. Below the simulation, the serial monitor displays the output of the sketch:

```
ALERT!!  
ALERT!!  
ALERT! ALERT!  
ALERT! t  
ALERT! t ALERT!
```

The screenshot shows the IBM Watson IoT Platform interface. On the left, a Python script is displayed in the editor. The script generates random values for pH, turbidity, and temperature, and publishes them as events to the IoT platform.

```
42 pH = random.random() * 14  
43 turbidity = random.randint(1, 1000)  
44 temperature = random.randint(0, 100)  
45  
46 data = {'pH': pH, 'turbid': turbidity, 'temp': temperature}  
47  
48 # print(data)  
49 def myOnPublishCallback():  
50     while True:
```

On the right, the IBM Watson IoT Platform console is shown. It displays a list of events published by the device. The events are listed in a table with columns for Event and Value.

Event	Value
demo	("pH":12,"turbid":93,"temp":87)
demo	("pH":7,"turbid":873,"temp":94)
demo	("pH":3,"turbid":204,"temp":19)
demo	("pH":11,"turbid":304,"temp":77)
demo	("pH":13,"turbid":16,"temp":50)

At the bottom, the console shows the device status: 00003, Disconnected, Micro\_controller\_2.

