

ASSIGNMENT-4

NAME: BHUVANESHWARI S

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 displays the Wokwi simulation environment. On the left, the code editor shows a C++ sketch for an Arduino Uno. The code initializes serial communication at 9600 baud, sets pin modes for digitalWrite and digitalWrite, and implements a loop that generates a pulse width modulation (PWM) signal on pin 7. The pulse width is calculated based on a random duration and a distance threshold. If the distance is less than or equal to 100, an alert is printed to the serial monitor.

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

The central panel shows a circuit diagram of an Arduino Uno connected to a motor and a sensor. The right panel displays the serial monitor output, showing the alert message "ALERT!!" being printed.

ALERT!!
ALERT!!
ALERT!
ALERT!
ALERT!
ALERT!
ALERT!

The screenshot displays the IBM Watson IoT Platform interface. The left panel shows the code editor for a Python script, which generates random data for pH, turbidity, and temperature, and publishes it to the IoT platform.

```
42 pH = random.r  
43 turbidity = random.randint(1,  
44 temperature = random.randint(0,  
45  
46 data = {'pH': pH, 'turbid': tur  
47  
48  
49 # print(data)  
50 def myOnPublishCallback():  
51 while True
```

The right panel shows the "Recent events" section, which displays a stream of data events. 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)

The bottom panel shows the "Items per page" dropdown set to 50, and the "1-3 of 3 items" indicator.

