

ASSIGNMENT 4

NAME: AKSHAYA J

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);

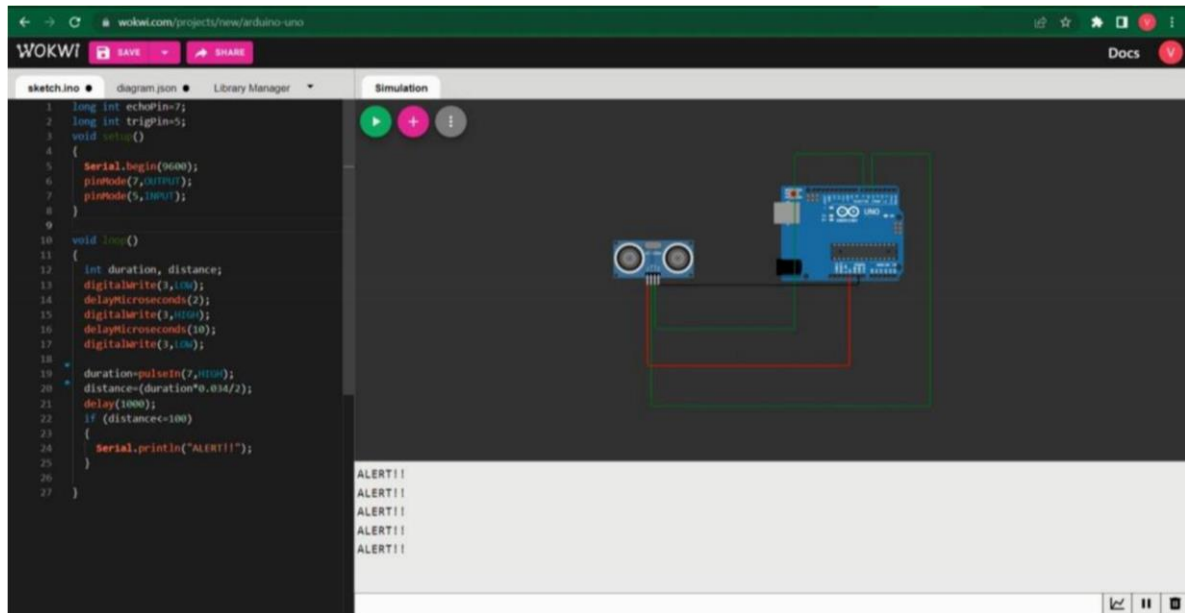
void loop()

    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:



IBM Watson IoT Platform interface showing a list of recent events. The interface includes tabs for Browse, Action, Device Types, and Interfaces, and an Add Device button. The main content area displays a table of events with columns for Event and Value. Below the table, there is a status bar showing the device ID 00003, its status as Disconnected, and the device name Micro_controller_2.

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}

Items per page 50 | 1-3 of 3 items 1 of 1 page

Python code snippet for generating random data and publishing it to the IBM Watson IoT Platform. The code uses the random module to generate random values for pH, turbidity, and temperature, and the paho-mqtt module to publish the data to a specific topic.

```
import random
import time
import paho.mqtt.client as mqtt

# Generate random data
pH = random.randint(2, 14)
turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)

# Create a dictionary for the data
data = {"pH": pH, "turbid": turbidity, "temp": temperature}

# Print the data
print(data)

# Define the publish callback function
def on_publish(client, userdata, result):
    pass

# Create an MQTT client
client = mqtt.Client()

# Connect to the MQTT broker
client.connect("192.168.1.100", 1883, 60)

# Subscribe to the topic
client.subscribe("demo")

# Publish the data
client.publish("demo", json.dumps(data))

# Keep the client alive
while True:
    time.sleep(10)
    client.publish("demo", json.dumps(data))
```

Run: Test_python_3.7.4

Published pH= 12 Turbidity:564 Temperature:54
Published pH= 2 Turbidity:571 Temperature:98
Published pH= 7 Turbidity:577 Temperature:65
Published pH= 8 Turbidity:352 Temperature:13
Published pH= 5 Turbidity:862 Temperature:88
Published pH= 3 Turbidity:834 Temperature:7
Published pH= 9 Turbidity:213 Temperature:89
Published pH= 14 Turbidity:677 Temperature:22
Published pH= 11 Turbidity:292 Temperature:108
Published pH= 2 Turbidity:53 Temperature:21
Published pH= 6 Turbidity:499 Temperature:69
Published pH= 11 Turbidity:238 Temperature:26
Published pH= 2 Turbidity:443 Temperature:43
Published pH= 6 Turbidity:986 Temperature:91
Published pH= 5 Turbidity:593 Temperature:85
Published pH= 14 Turbidity:308 Temperature:86
Published pH= 4 Turbidity:532 Temperature:8

IBM Watson IoT Platform interface showing a list of recent events. The interface includes tabs for Browse, Action, Device Types, and Interfaces, and an Add Device button. The main content area displays a table of events with columns for Event and Value. Below the table, there is a status bar showing the device ID 00003, its status as Disconnected, and the device name Micro_controller_2.

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}

Items per page 50 | 1-3 of 3 items 1 of 1 page