

S.Sandhiya Assignment -4

Question-1: Write code and connections in wokwi for ultrasonic sensor.
Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events.

Solution:

```
#define ECHO_PIN 2
#define TRIG_PIN 3
#define organization ="9nptzb"
#define deviceType=" Arduino"
#define deviceId ="20019"
#define authMethod ="use-token-auth"
#define authToken ="987654321"

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(TRIG_PIN,OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

float readDistanceCM() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
```

```
int duration = pulseIn(ECHO_PIN, HIGH);  
return duration * 0.034 / 2;
```

```
}
```

```
void loop() {  
    // put your main code here, to run repeatedly:  
    float distance = readDistanceCM();  
    if(distance <= 100)  
    {  
        Serial.println("person detected ");  
    }  
    else{  
        Serial.print("Measured distance: ");  
        Serial.println(readDistanceCM());  
    }  
    delay(1000);  
}
```

}

Output:

The screenshot displays the Wokwi online IDE interface. On the left, the `sketch.ino` file is open, showing the following code:

```
1 #define ECHO_PIN 2
2 #define TRIG_PIN 3
3 #define organization "9nptzb"
4 #define deviceType "Arduino"
5 #define deviceId "20019"
6 #define authMethod "use-token-auth"
7 #define authToken "987654321"
8
9 void setup() {
10   // put your setup code here, to run once:
11   Serial.begin(9600);
12   pinMode(TRIG_PIN, OUTPUT);
13   pinMode(ECHO_PIN, INPUT);
14 }
15 float readDistanceCM() {
16   digitalWrite(TRIG_PIN, LOW);
17   delayMicroseconds(2);
18   digitalWrite(TRIG_PIN, HIGH);
19   delayMicroseconds(10);
20   digitalWrite(TRIG_PIN, LOW);
21   int duration = pulseIn(ECHO_PIN, HIGH);
22   return duration * 0.034 / 2;
23 }
24
25 }
26
27 void loop() {
28   // put your main code here, to run repeatedly:
29   float distance = readDistanceCM();
```

On the right, the simulation shows an HC-SR04 ultrasonic sensor connected to an Arduino Uno. The sensor's VCC pin is connected to the Arduino's 5V pin, GND to GND, TRIG to digital pin 3, and ECHO to digital pin 2. The simulation output window displays the following data:

```
Measured distance: 395.25
Measured distance: 395.25
Measured distance: 395.25
Measured distance: 395.25
Measured distance: 395.27
Measured distance: 395.25
Measured distance: 395.27
```

The Wokwi interface also shows a timer at 00:20.487 and 100% battery level. The bottom status bar indicates the system time as 8:59 AM on 11/4/2022.

Python 3.10: C... X IBM Cloud X [GitHub] Please X Sign in to GitH... X Service Details X IBM Watson IoT X New Arduino U X Speed Dial X + Q _ X

wokwi.com/projects/new/arduino-uno

Learn Python Tutori... 25+ Aesthetic Color...

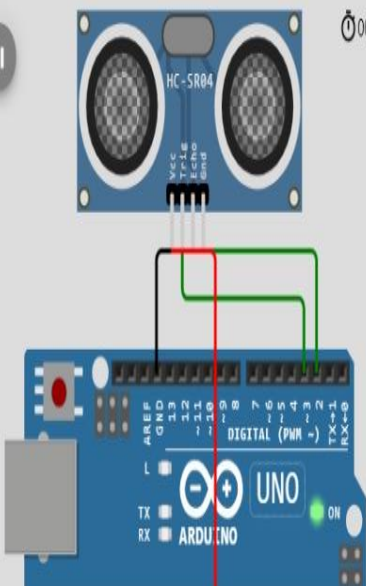
WOKWI SAVE SHARE Docs

sketch.ino diagram.json Library Manager

```
1 #define ECHO_PIN 2
2 #define TRIG_PIN 3
3 #define organization "gnptzb"
4 #define deviceType "Arduino"
5 #define deviceId "20019"
6 #define authMethod "use-token-auth"
7 #define authToken "987654321"
8
9 void setup() {
10   // put your setup code here, to run once:
11   Serial.begin(9600);
12   pinMode(TRIG_PIN, OUTPUT);
13   pinMode(ECHO_PIN, INPUT);
14 }
15 float readDistanceCM() {
16   digitalWrite(TRIG_PIN, LOW);
17   delayMicroseconds(2);
18   digitalWrite(TRIG_PIN, HIGH);
19   delayMicroseconds(10);
20   digitalWrite(TRIG_PIN, LOW);
21   int duration = pulseIn(ECHO_PIN, HIGH);
22   return duration * 0.034 / 2;
23 }
24
25 void loop() {
26   // put your main code here, to run repeatedly:
27   float distance = readDistanceCM();
```

Simulation

00:20.487 100%



Measured distance: 395.25
Measured distance: 395.25
Measured distance: 395.25
Measured distance: 395.25
Measured distance: 395.27
Measured distance: 395.25
Measured distance: 395.27

24°C Cloudy 8:59 AM 11/4/2022

Wokwi link: <https://wokwi.com/projects/347412887308862034>

