

## Assignment-4

**Team ID:** PNT2022TMID52019

**Name:** Yogeswar. K

**1. Write Code and connections in wokwi for ultrasonic sensor. whatever distance is less than 100 cm send "Alert" to IBM cloud and display in devicerecent events.**

**Solutin:**

```
//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
mint1= 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("*****");

}

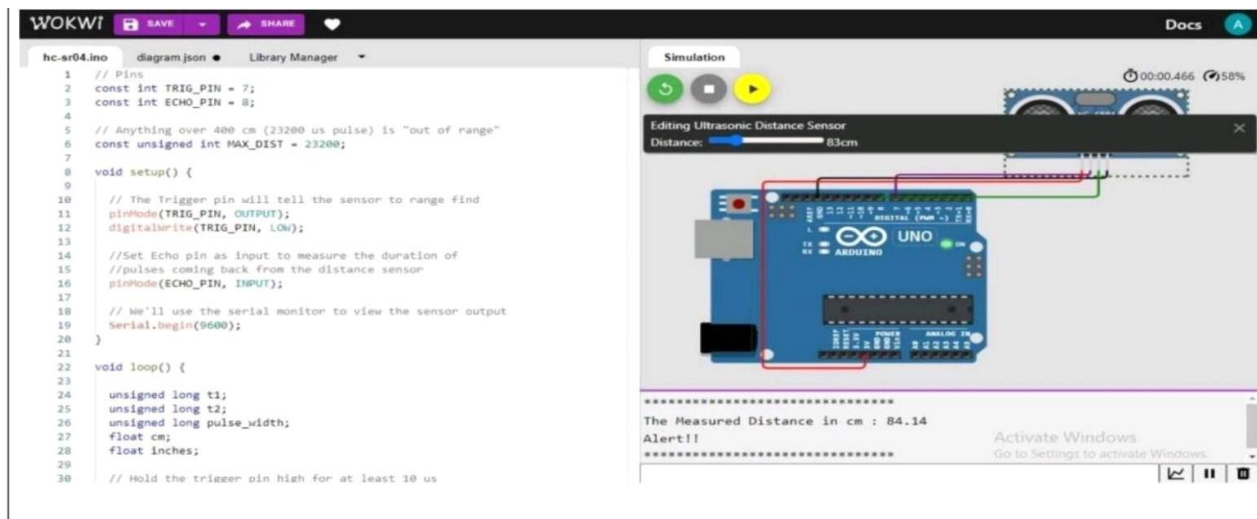
//wait at least 1000ms before next
measurementDelay(1000);

}

```

## Output:

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



- If the distance is more than 100 cm, it won't alert

WOKWI

SAVE

SHARE

Docs

hc-sr04.ino

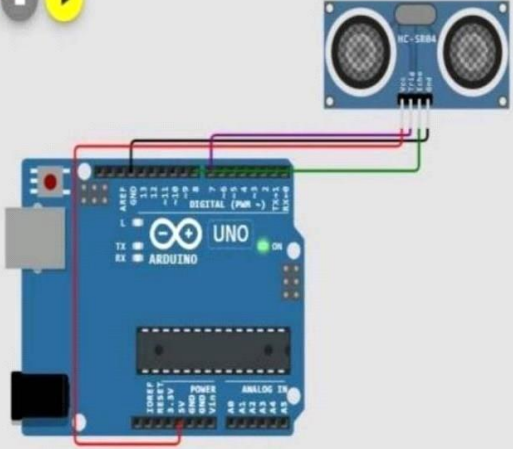
diagram.json

Library Manager

```
1 // Pins
2 const int TRIG_PIN = 7;
3 const int ECHO_PIN = 8;
4
5 // Anything over 400 cm (23200 us pulse) is "out of range"
6 const unsigned int MAX_DIST = 23200;
7
8 void setup() {
9
10 // The Trigger pin will tell the sensor to range find
11 pinMode(TRIG_PIN, OUTPUT);
12 digitalWrite(TRIG_PIN, LOW);
13
14 //Set Echo pin as input to measure the duration of
15 //pulses coming back from the distance sensor
16 pinMode(ECHO_PIN, INPUT);
17
18 // We'll use the serial monitor to view the sensor output
19 Serial.begin(9600);
20 }
21
22 void loop() {
23
24 unsigned long t1;
25 unsigned long t2;
26 unsigned long pulse_width;
27 float cm;
28 float inches;
```

Simulation


00:00.599 62%



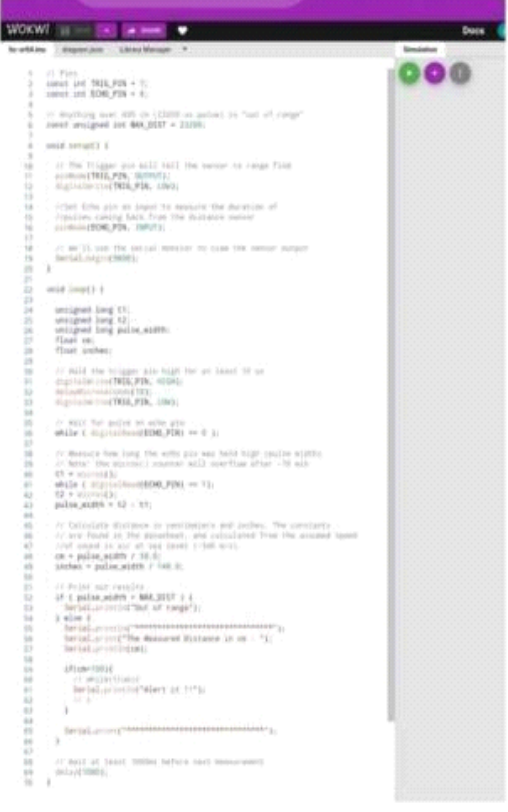
\*\*\*\*\*  
The Measured Distance in cm : 227.10  
\*\*\*\*\*

Activate Windows  
Go to Settings to activate Windows.

- Simulation and code execution



The left panel shows the Wokwi simulation interface. At the top, there's a header with the Wokwi logo, a file icon, a dropdown menu, a 'SHARE' button, a heart icon, and a user profile icon. Below the header, there are two tabs: 'Simulation' (active) and 'Code'. The 'Simulation' tab displays a virtual Arduino Uno board connected to a breadboard with an ultrasonic sensor (HC-SR04) and a potentiometer. The sensor is connected to the Arduino's digital pins (VCC to 5V, GND to GND, and Trig to digital pin 2, Echo to digital pin 3). The potentiometer is connected to the Arduino's analog pins (VCC to 5V, GND to GND, and Wiper to analog pin A0).



The right panel shows the Wokwi code editor. It has a header with the Wokwi logo, a file icon, a dropdown menu, a 'SHARE' button, a heart icon, and a user profile icon. Below the header, there are two tabs: 'Code' (active) and 'Data'. The 'Code' tab displays the C++ code for the simulation. The code is as follows:

```

1 // Pin
2 const int TRIG_PIN = 2;
3 const int ECHO_PIN = 3;
4
5 // Anything over 200 cm (2000 us pulse) is "out of range"
6 const unsigned int MAX_DIST = 2000;
7
8 void setup() {
9
10   // The trigger pin will tell the sensor to range find
11   pinMode(TRIG_PIN, OUTPUT);
12   digitalWrite(TRIG_PIN, LOW);
13
14   // Echo pin is input to measure the duration of
15   // pulses coming back from the distance sensor
16   pinMode(ECHO_PIN, INPUT);
17
18   // We'll use the serial monitor to view the sensor output
19   Serial.begin(9600);
20 }
21
22 void loop() {
23
24   unsigned long t1;
25   unsigned long t2;
26   unsigned long pulse_width;
27   float cm;
28   float inches;
29
30   // Hold the trigger pin high for at least 10 us
31   digitalWrite(TRIG_PIN, HIGH);
32   delayMicroseconds(10);
33   digitalWrite(TRIG_PIN, LOW);
34
35   // Wait for pulse to echo pin
36   while (digitalRead(ECHO_PIN) == LOW);
37
38   // Measure how long the echo pin was held high (pulse width)
39   // Note: the distance sensor will overflow after ~10 m/s
40   t1 = micros();
41   while (digitalRead(ECHO_PIN) == HIGH);
42   t2 = micros();
43   pulse_width = t2 - t1;
44
45   // Calculate distance in centimeters and inches. The constants
46   // are found in the spreadsheet, and calculated from the sound speed
47   // of sound is not at sea level (~340 m/s)
48   cm = pulse_width / 58.8;
49   inches = pulse_width / 148.0;
50
51   // Print our results
52   if (pulse_width < MAX_DIST) {
53     Serial.println("Range of range");
54   } else {
55     Serial.println("Out of range");
56     Serial.println("The measured distance is cm : ");
57     Serial.println(cm);
58   }
59
60   delay(1000);
61   // We'll use the serial monitor to view the sensor output
62   Serial.println("Start of 1st");
63 }
64
65 // Wait at least 1000ms before next measurement
66 delay(1000);
67
68 }

```

