

# ASSIGNMENT-4

## QUESTION:

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

**CASE 1:** Distance less than 100cm → It Alerts

The screenshot displays the Wokwi IDE interface. On the left, the code for `hc-sr04.ino` is shown. The code initializes the trig and echo pins, reads the distance in centimeters, and checks if it is less than 100cm. If so, it prints "Measured distance: " followed by the distance value to the serial monitor. The right side shows a simulation of an Arduino Uno board connected to an HC-SR04 ultrasonic sensor. The sensor's VCC is connected to the 5V pin, GND to GND, TRIG to digital pin 5, and ECHO to digital pin 4. A red LED is connected to the 5V pin. The serial monitor at the bottom shows the output: "Measured distance: 79.87", "Measured distance: 79.78", "Measured distance: 79.78", "Measured distance: 79.78", "Measured distance: 79.87", "Measured distance: 79.78", and "Measured distance: 79.78". The simulation status bar indicates a runtime of 00:16.097 and 85% battery.

```
1 // hc-sr04.ino
2 #include <Arduino.h>
3
4 // Define pins
5 const int TRIG_PIN = 5;
6 const int ECHO_PIN = 4;
7
8 // Initialize pins
9 pinMode(TRIG_PIN, OUTPUT);
10 pinMode(ECHO_PIN, INPUT);
11
12 // Function to read distance in cm
13 float readDistanceCM() {
14   digitalWrite(TRIG_PIN, LOW);
15   delayMicroseconds(2);
16   digitalWrite(TRIG_PIN, HIGH);
17   delayMicroseconds(10);
18   digitalWrite(TRIG_PIN, LOW);
19   int duration = pulseIn(ECHO_PIN, HIGH);
20   return duration * 0.034 / 2;
21 }
22
23 // Main loop
24 void loop() {
25   float distance = readDistanceCM();
26
27   bool isNearby = distance < 100;
28   digitalWrite(LED_BUILTIN, isNearby);
29
30   Serial.print("Measured distance: ");
31   Serial.println(readDistanceCM());
32
33   delay(100);
34 }
```

## CASE 2: Distance more than 100cm → It won't Alert

WOKWI

SAVE SHARE

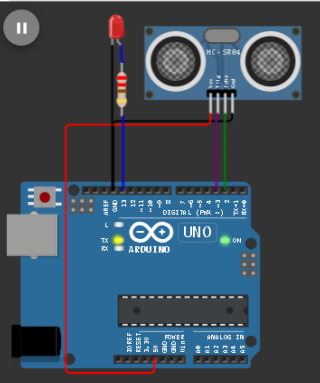
Docs

hc-sr04.ino diagram.json Library Manager

```
1 #define ECHO_PIN 2
2 #define TRIG_PIN 3
3
4 void setup() {
5   Serial.begin(115200);
6   pinMode(LED_BUILTIN, OUTPUT);
7   pinMode(TRIG_PIN, OUTPUT);
8   pinMode(ECHO_PIN, INPUT);
9 }
10
11 float readDistanceCM() {
12   digitalWrite(TRIG_PIN, LOW);
13   delayMicroseconds(2);
14   digitalWrite(TRIG_PIN, HIGH);
15   delayMicroseconds(10);
16   digitalWrite(TRIG_PIN, LOW);
17   int duration = pulseIn(ECHO_PIN, HIGH);
18   return duration * 0.034 / 2;
19 }
20
21 void loop() {
22   float distance = readDistanceCM();
23
24   bool isNearby = distance < 100;
25   digitalWrite(LED_BUILTIN, isNearby);
26
27   Serial.print("Measured distance: ");
28   Serial.println(readDistanceCM());
29
30   delay(100);
31 }
32
```

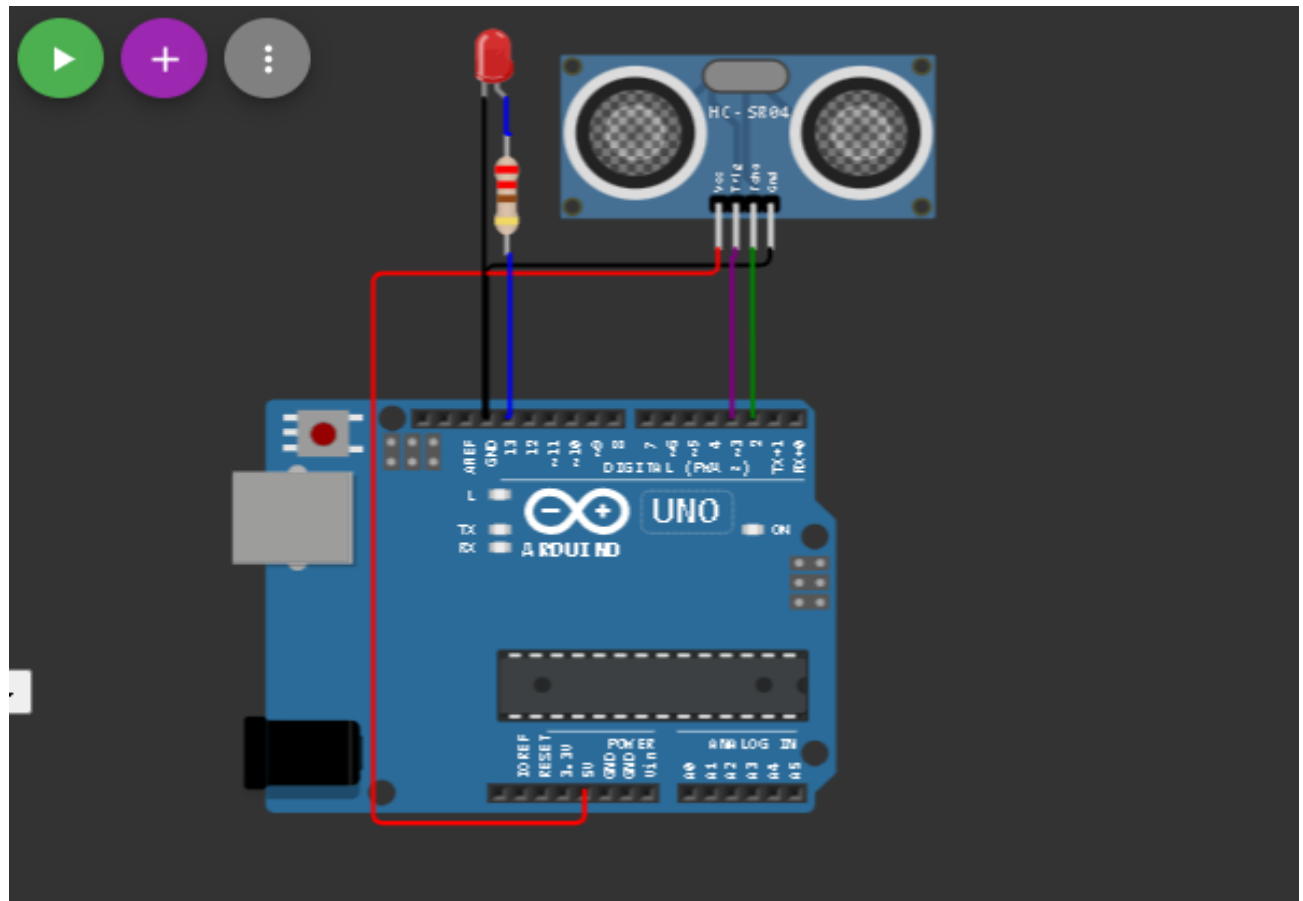
Simulation

00:14.080 92%



Measured distance: 118.58  
Measured distance: 118.56  
Measured distance: 118.58  
Measured distance: 118.58  
Measured distance: 118.58  
Measured distance: 118.56  
Measured distance: 118.58

## Circuit Design:



## **CODING:**

```
#define ECHO_PIN 2
#define TRIG_PIN 3

void setup() {
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  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() {
  float distance = readDistanceCM();

  bool isNearby = distance < 100;
  digitalWrite(LED_BUILTIN, isNearby);

  Serial.print("Measured distance: ");
  Serial.println(readDistanceCM());

  delay(100);
}
```

```

t1 = 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.println("The Measured Distance in cm:");
    Serial.println(cm);

    if (cm < 100)
    {
        //while (true)
        {
            Serial.println("Alert!");
        }
    }
    Serial.println("*****");
}

//wait at least 1000ms before next measurement

delay(1000);
}

```

#### WOKWI LINK:

<https://wokwi.com/projects/3466596108174628>

