

ASSIGNMENT -4

WRITE A CODE AND CONNECTION IN WOWKI FOR ULTASONIC SENSOR.WHENEVER DISTANCE IS LESS THAN 100 CMS SEND “ALERT” TO IBM CLOUD AND DISPLAY IN DEVICE RECENT EVENTS

CODE

```
// 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

pinMode(TRIG_PIN, OUTPUT);

digitalWrite(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 min

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 (~340 m/s).

cm = pulse_width/58.0;
inches = pulse_width/148.0;

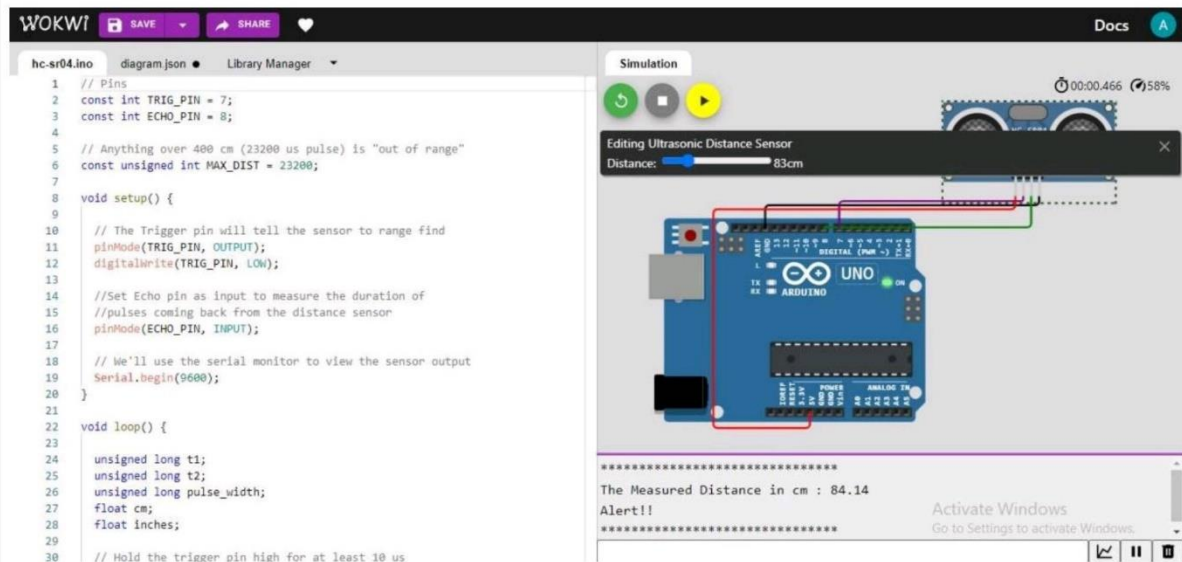
// Print out results
if (pulse_width > MAX_DIST) { Serial.println("Out of range");
} else {
Serial.println("*****");
Serial.print("The Measured Distance in cm: ");
Serial.println(cm);
if(cm<100){
// while(true)
{
Serial.println("Alert!!");
// }
}
Serial.print("*****");
}

// Wait at least 1000ms before next measurement
delay(1000);
}

```

Output:

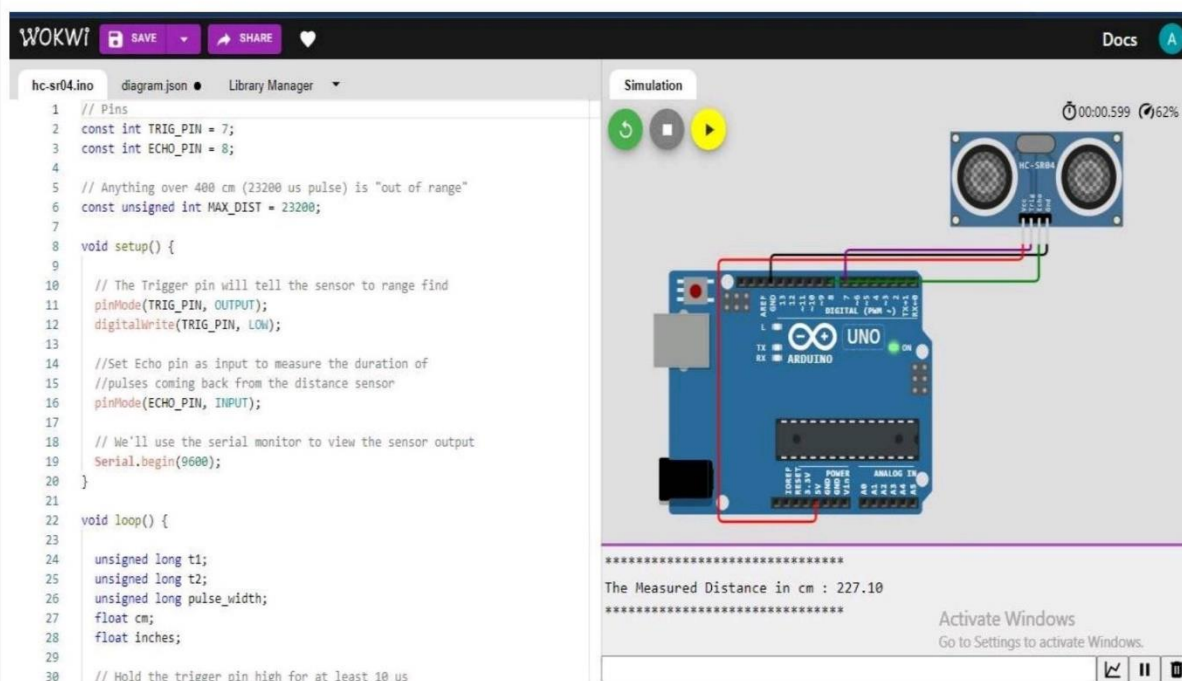
1) If the distance is less than 100 cms , it alerts.



The screenshot shows the Wokwi simulation interface. On the left, the code for `hc-sr04.ino` is displayed. The code sets the trigger pin to 7 and the echo pin to 8. It defines a maximum distance of 23200 cm. In the `setup()` function, the trigger pin is configured as an output and the echo pin as an input. The `loop()` function measures the distance and prints it to the serial monitor. The simulation on the right shows an Arduino Uno board connected to an HC-SR04 sensor. A dialog box titled "Editing Ultrasonic Distance Sensor" shows the distance set to 83cm. The serial monitor output shows "The Measured Distance in cm : 84.14" and "Alert!!".

```
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;
29
30 // Hold the trigger pin high for at least 10 us
```

2) If the distance is more than 100 cms, it won't alert.



The screenshot shows the Wokwi simulation interface. On the left, the code for `hc-sr04.ino` is displayed. The code sets the trigger pin to 7 and the echo pin to 8. It defines a maximum distance of 23200 cm. In the `setup()` function, the trigger pin is configured as an output and the echo pin as an input. The `loop()` function measures the distance and prints it to the serial monitor. The simulation on the right shows an Arduino Uno board connected to an HC-SR04 sensor. The serial monitor output shows "The Measured Distance in cm : 227.10".

```
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;
29
30 // Hold the trigger pin high for at least 10 us
```

3) Simulation and code execution

