

VSB ENGINEERING COLLEGE, KARUR-639111  
ASSIGNMENT 4

Name: Dhayananth E.P

**Project Title:** Smart Farmer- IoT Enabled Smart Farming Application

**Project Domain:** Internet of Things

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

**Solution:**

```
//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 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 (- 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);

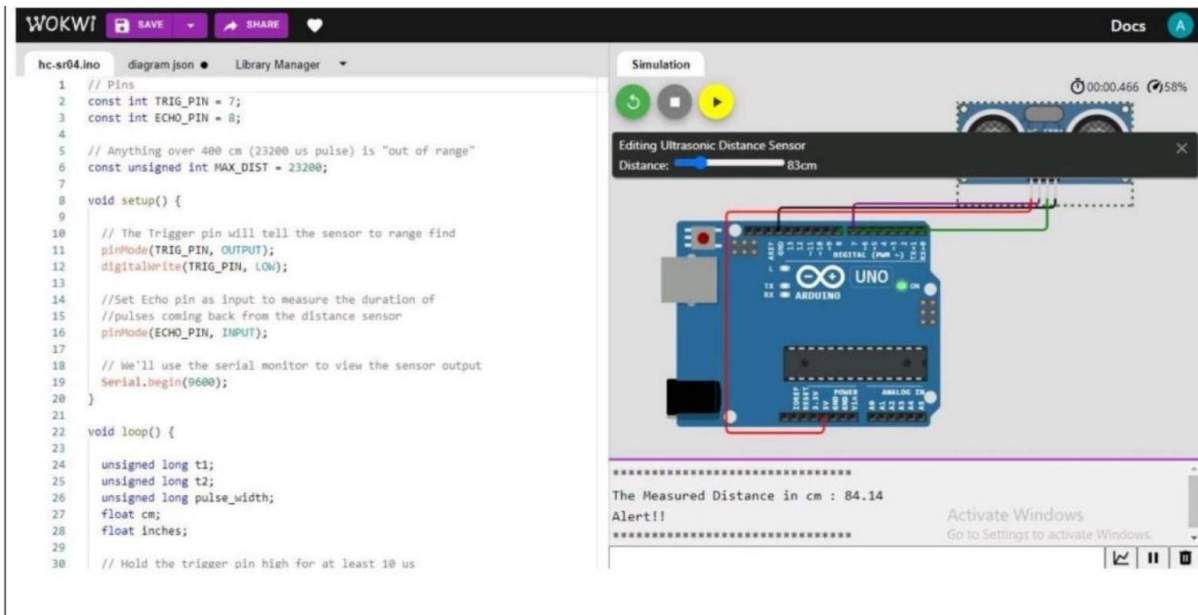
if( cm < 100 ){
    //while(true){
        Serial.println("Alert!!");
        //}
    }
Serial.print("*****");
}

//wait at least 1000ms before next measurement
Delay(1000);
}

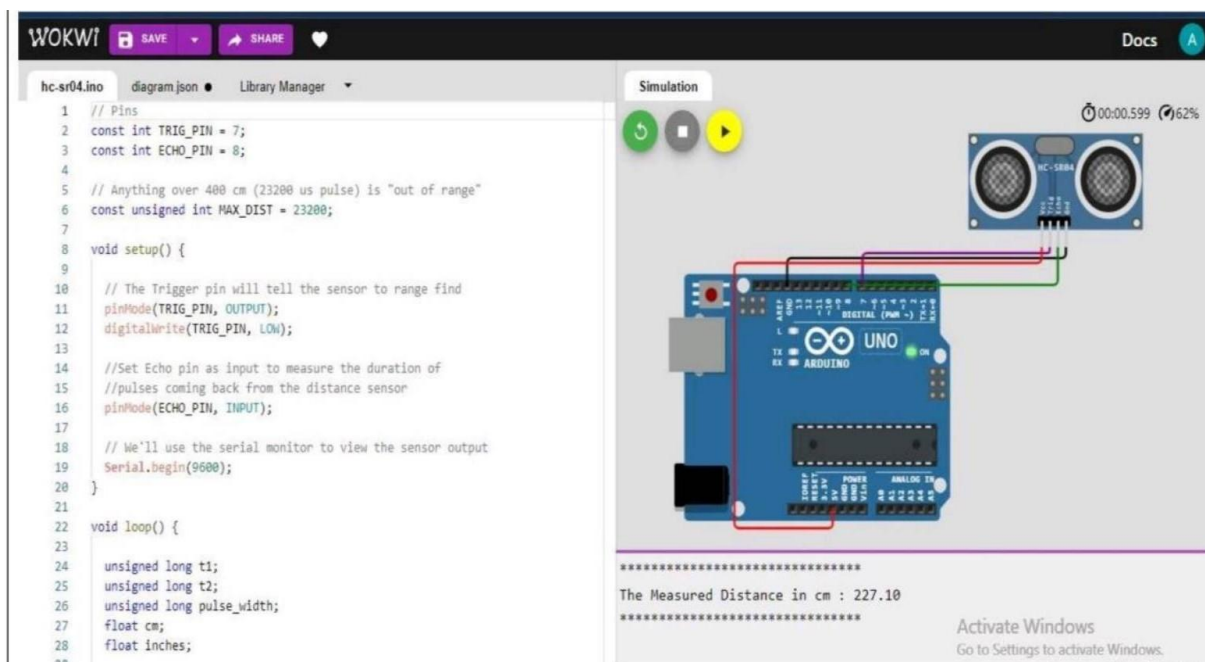
```

## Output:

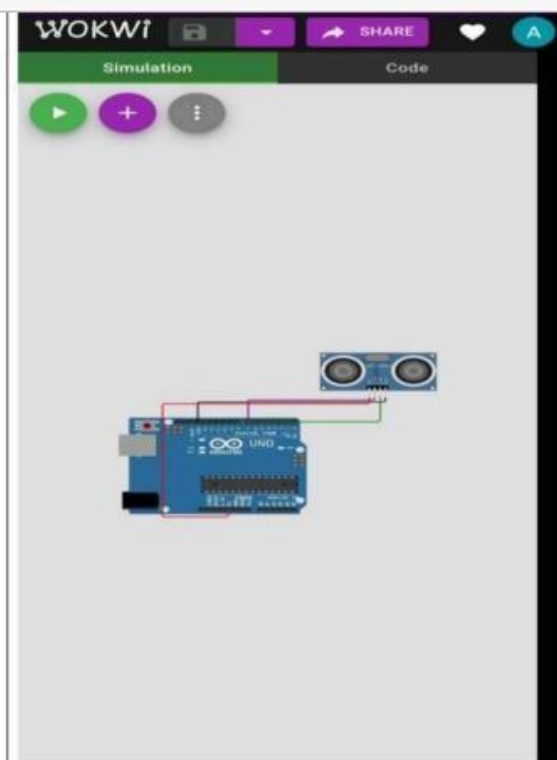
**1.If the distance is less than 100 cms ,it alerts.**



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



3.Simulation and code execution



```

1 // Pin
2 const int TRIG_PIN = 2;
3 const int ECHO_PIN = 3;
4
5 // Working 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 // Set the 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
23 void loop() {
24
25   unsigned long t1;
26   unsigned long t2;
27   unsigned long pulse_width;
28   float cm;
29   float inches;
30
31 // Hold the trigger pin high for at least 10 us
32 digitalWrite(TRIG_PIN, HIGH);
33 delayMicroseconds(10);
34 digitalWrite(TRIG_PIN, LOW);
35
36 // Wait for pulse so echo pin
37 while (digitalRead(ECHO_PIN) == 0);
38
39 // Measure how long the echo pin was held high (pulse width)
40 // Note: the Arduino's counter will overflow after ~18 ms
41 t1 = micros();
42 while (digitalRead(ECHO_PIN) == 1);
43 t2 = micros();
44 pulse_width = t2 - t1;
45
46 // Calculate distance in centimeters and inches. The constants
47 // are found in the datasheet, and calculated from the assumed speed
48 // of sound in air at sea level (~340 m/s).
49 cm = pulse_width / 58.8;
50 inches = pulse_width / 148.3;
51
52 // Print our results
53 if ( pulse_width > MAX_DIST ) {
54   Serial.println("out of range");
55 } else {
56   Serial.println("The Measured Distance in cm : ");
57   Serial.println(cm);
58   Serial.println(inches);
59
60 // Clear/Reset
61 // digitalWrite(
62   Serial.println("Alert it !!");
63 }
64
65 // Serial.println("The Measured Distance in cm : ");
66 // Serial.println(cm);
67
68 // Wait at least 100ms before next measurement
69 delay(100);
70 }

```

