

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);

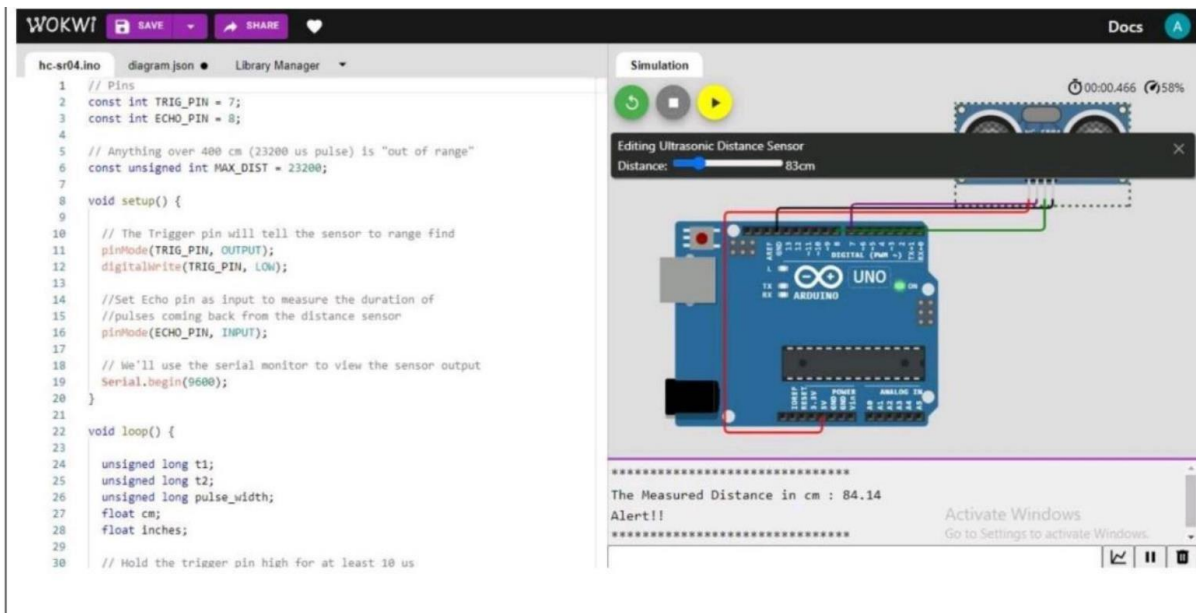
if( cm < 100 ){
    //while(true){
        Serial.println("Alert!!");
    //}
}
Serial.print("*****");
}

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

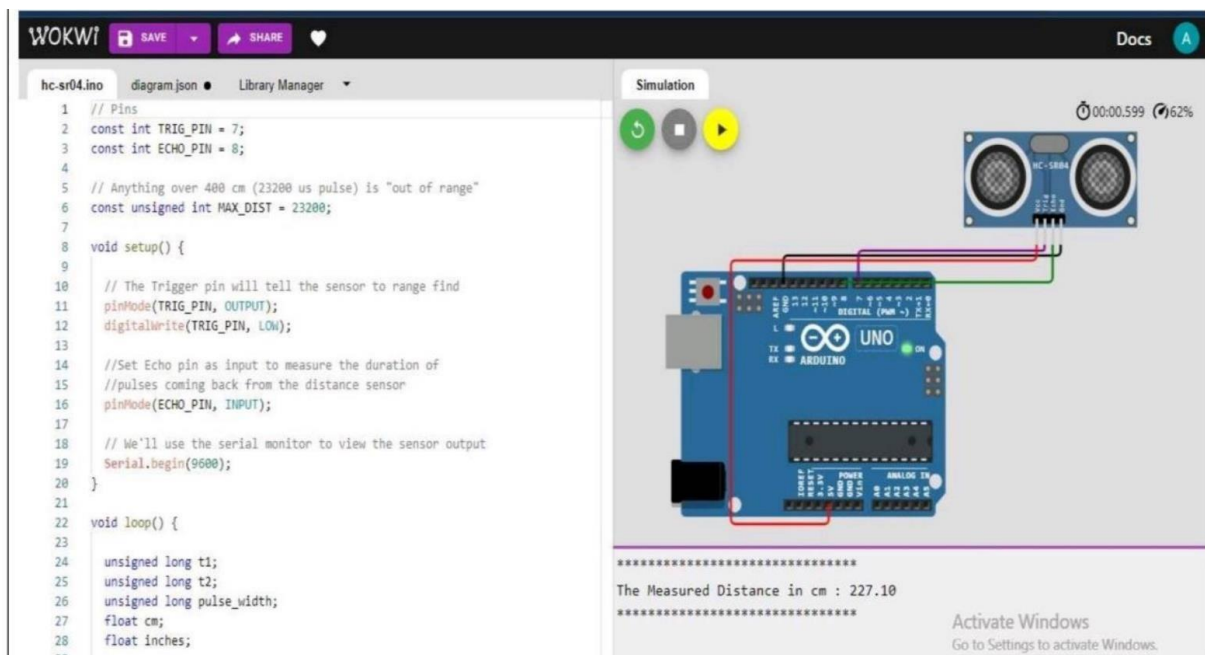
```

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 // Trig
2 const int TRIG_PIN = 12;
3 const int ECHO_PIN = 11;
4
5 // Anything over 400 cm (1200 us pulse) is "out of range"
6 const unsigned int MAX_DIST = 2200;
7
8 void setup() {
9
10 // The Trigger pin will tell the sensor to "send Ping"
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 see 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 on echo pin
36 while ( !digitalRead(ECHO_PIN) ) {}
37
38 // Measure how long the echo pin was held high (pulse width)
39 // Note: the default counter will overflow after ~18 min
40 t1 = micros();
41 while ( !digitalRead(ECHO_PIN) ) {}
42 t2 = micros();
43 pulse_width = t2 - t1;
44
45 // Calculate distance in centimeters and inches. The constants
46 // are found in the datasheet, and calculated from the assumed speed
47 // of sound in air at sea level (340 m/s).
48 cm = pulse_width / 58.0;
49 inches = pulse_width / 148.0;
50
51 // Print our results
52 if ( pulse_width > MAX_DIST ) {
53   Serial.println("Out of range!");
54 } else {
55   Serial.println("The Measured Distance is cm : ");
56   Serial.print(cm);
57 }
58
59 if(cm>100){
60   Serial.println("Alert of 100");
61 }
62
63 Serial.println("~~~~~");
64 }
65
66 // Wait at least 100ms before next measurement
67 delay(100);
68 }

```

