

hr-sr04-Ultrasonic-Simulation.ino

```
#include "Ultrasonic.h"
```

```
/*  
    Pass as a parameter the trigger and echo pin, respectively,  
    or only the signal pin (for sensors 3 pins), like:  
    Ultrasonic ultrasonic(13);  
*/
```

```
Ultrasonic ultrasonic(12, 13);  
int distance;
```

```
void setup() {  
    Serial.begin(9600);  
}
```

```
void loop() {  
    // Pass INC as a parameter to get the distance in inches  
  
    distance = ultrasonic.read(CM);  
  
    Serial.print("Distance in CM: ");  
    Serial.println(distance);  
  
    distance = ultrasonic.read(INC);  
  
    Serial.print("Distance in Inches: ");  
    Serial.println(distance);  
  
    delay(1000);  
}
```

diagram.json

```
{  
    "version": 1,  
    "author": "Sasi Balaji",  
    "editor": "wokwi",  
    "parts": [  
        { "type": "wokwi-arduino-uno", "id": "uno", "top": 259.31, "left": 31.06,  
"attrs": {} },  
        {  
            "type": "wokwi-hc-sr04",  
            "id": "ultrasonic",  
            "top": 86.99,  
            "left": 109.89,  
            "attrs": { "distance": "20" }  
        }  
    ]  
}
```

```

],
"connections": [
  [ "uno:GND.1", "ultrasonic:GND", "black", [ "v-8", "*", "v8" ] ],
  [ "uno:13", "ultrasonic:ECHO", "green", [ ] ],
  [ "uno:12", "ultrasonic:TRIG", "purple", [ "*", "v4" ] ],
  [ "uno:5V", "ultrasonic:VCC", "red", [ "v16", "h-96", "*", "v12" ] ]
],
"dependencies": {}
}

```

Ultrasonic.h

```

#ifndef Ultrasonic_h
#define Ultrasonic_h

/*
 * Values of divisors
 */
#define CM 28
#define INC 71

class Ultrasonic {
public:
  Ultrasonic(uint8_t sigPin) : Ultrasonic(sigPin, sigPin) {};
  Ultrasonic(uint8_t trigPin, uint8_t echoPin, unsigned long timeOut =
20000UL);
  unsigned int read(uint8_t und = CM);
  unsigned int distanceRead(uint8_t und = CM) __attribute__((deprecated
("This method is deprecated, use read() instead.")));
  void setTimeout(unsigned long timeOut) {timeout = timeOut;}
  void setMaxDistance(unsigned long dist) {timeout = dist*CM*2;}

private:
  uint8_t trig;
  uint8_t echo;
  boolean threePins = false;
  unsigned long previousMicros;
  unsigned long timeout;
  unsigned int timing();
};

#endif // Ultrasonic_h

```

Ultrasonic.cpp

```

#if ARDUINO >= 100
#include <Arduino.h>

```

```

#else
    #include <WProgram.h>
#endif

#include "Ultrasonic.h"

Ultrasonic::Ultrasonic(uint8_t trigPin, uint8_t echoPin, unsigned long
timeOut) {
    trig = trigPin;
    echo = echoPin;
    threePins = trig == echo ? true : false;
    pinMode(trig, OUTPUT);
    pinMode(echo, INPUT);
    timeout = timeOut;
}

unsigned int Ultrasonic::timing() {
    if (threePins)
        pinMode(trig, OUTPUT);

    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);

    if (threePins)
        pinMode(trig, INPUT);

    previousMicros = micros();
    while(!digitalRead(echo) && (micros() - previousMicros) <= timeout); // wait
for the echo pin HIGH or timeout
    previousMicros = micros();
    while(digitalRead(echo) && (micros() - previousMicros) <= timeout); // wait
for the echo pin LOW or timeout

    return micros() - previousMicros; // duration
}

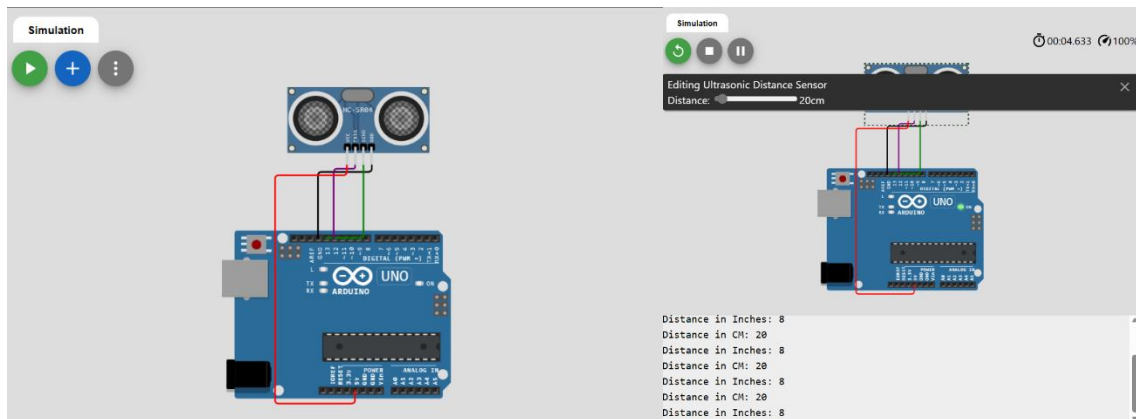
/*
 * If the unit of measure is not passed as a parameter,
 * sby default, it will return the distance in centimeters.
 * To change the default, replace CM by INC.
 */
unsigned int Ultrasonic::read(uint8_t und) {
    return timing() / und / 2; //distance by divisor
}

```

```

/*
 * This method is too verbal, so, it's deprecated.
 * Use read() instead.
 */
unsigned int Ultrasonic::distanceRead(uint8_t und) {
    return read(und);
}

```



IBM Cloud Output:

Simulation

00:04.633 100%

Editing Ultrasonic Distance Sensor
Distance: 20cm

Distance in Inches: 8
Distance in CM: 20
Distance in Inches: 8
Distance in CM: 20
Distance in Inches: 8
Distance in CM: 20
Distance in Inches: 8

IBM Cloud Output:

Browse Action Device Types Interfaces

Add Device

Identity	Device Information	Recent Events	State	Logs
The recent events listed show the live stream of data that is coming and going from this device.				
Event	Value	Format	Last Received	
event_1	{"distance":7,"Alert":"Distance less than 10"}	json	a few seconds ago	
event_1	{"distance":9,"Alert":"Distance less than 10"}	json	a few seconds ago	
event_1	{"distance":8,"Alert":"Distance less than 10"}	json	a few seconds ago	
event_1	{"distance":9,"Alert":"Distance less than 10"}	json	a few seconds ago	