

Aryan Gupta  
Partner: Anthony Samagaio

Video Link: <https://youtu.be/BoFbs03vxjQ>

Objective: The video will demonstrate a program to measure the distance to an object by measuring the distance from an ultrasonic sensor five times with a delay of one seconds between each measurement. The final distance was calculated to be the median of the five data points. This process was repeated after 5 seconds.

Commentary:

0:01: Behold... My box... (Insert Thor: Ragnarok meme here)

0:03: Needed to use a level converter since my sensor wouldn't run off of 3.3V

0:07: The output from the MSP432

0:13: Placed hand in front of sensor to simulate errors

0:16: Median of dataset was successfully able to remove errors

0:27: One more run without hand in front of sensor

Code:

```
/*  
HC-SR04 Ultrasonic Distance Sensor Example  
Demonstrates sensing distance with the HC-SR04 using Texas Instruments  
LaunchPads.
```

```
Created by Frank Milburn 5 Jun 2015  
Released into the public domain.
```

```
Modified by James Conrad 8 Jun 2020  
Modified by Aryan Gupta 12 Jun 2020  
*/
```

```
//another reason why I hate arduino  
//https://arduinojson.org/v6/error/macro-min-passed-3-arguments-but-takes-just-2/  
#undef min  
#undef max  
#include <algorithm>
```

```
const int trigPin = 32; //This is Port Pin 3.5 on the MSP432 Launchpad  
const int echoPin = 33; //This is Port Pin 5.1 on the MSP432 Launchpad
```

```
constexpr size_t NUM_SAMPLES = 5;
```

```

void setup() {
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    Serial.begin(9600);
    Serial.println("Starting HC-SR04 Test...");
}

long get_sonar_val() {
    long samples[NUM_SAMPLES];
    long inches;
    long centimeters;

    for (int i = 0; i < NUM_SAMPLES; ++i) {
        digitalWrite(trigPin, LOW);    // send low to get a clean pulse
        delayMicroseconds(5);          // let it settle
        digitalWrite(trigPin, HIGH);   // send high to trigger device
        delayMicroseconds(10);         // let it settle

        samples[i] = pulseIn(echoPin, HIGH);

        inches = samples[i] / 148;
        centimeters = samples[i] / 58;

        Serial.print("Distance = ");
        Serial.print(inches);
        Serial.print(" inches");
        Serial.print(" ");
        Serial.print(centimeters);
        Serial.println(" centimeters");

        delay(1000);
    }

    std::sort(samples, samples + NUM_SAMPLES);

    return samples[(NUM_SAMPLES / 2) + 1];
}

void loop() {
    long pulseLength;
    long inches;

```

```
long centimeters;

pulseLength = get_sonar_val();
inches = pulseLength / 148;
centimeters = pulseLength / 58;

Serial.print("Median distance = ");
Serial.print(inches);
Serial.print(" inches");
Serial.print(" ");
Serial.print(centimeters);
Serial.println(" centimeters");
delay(5000);
}
```