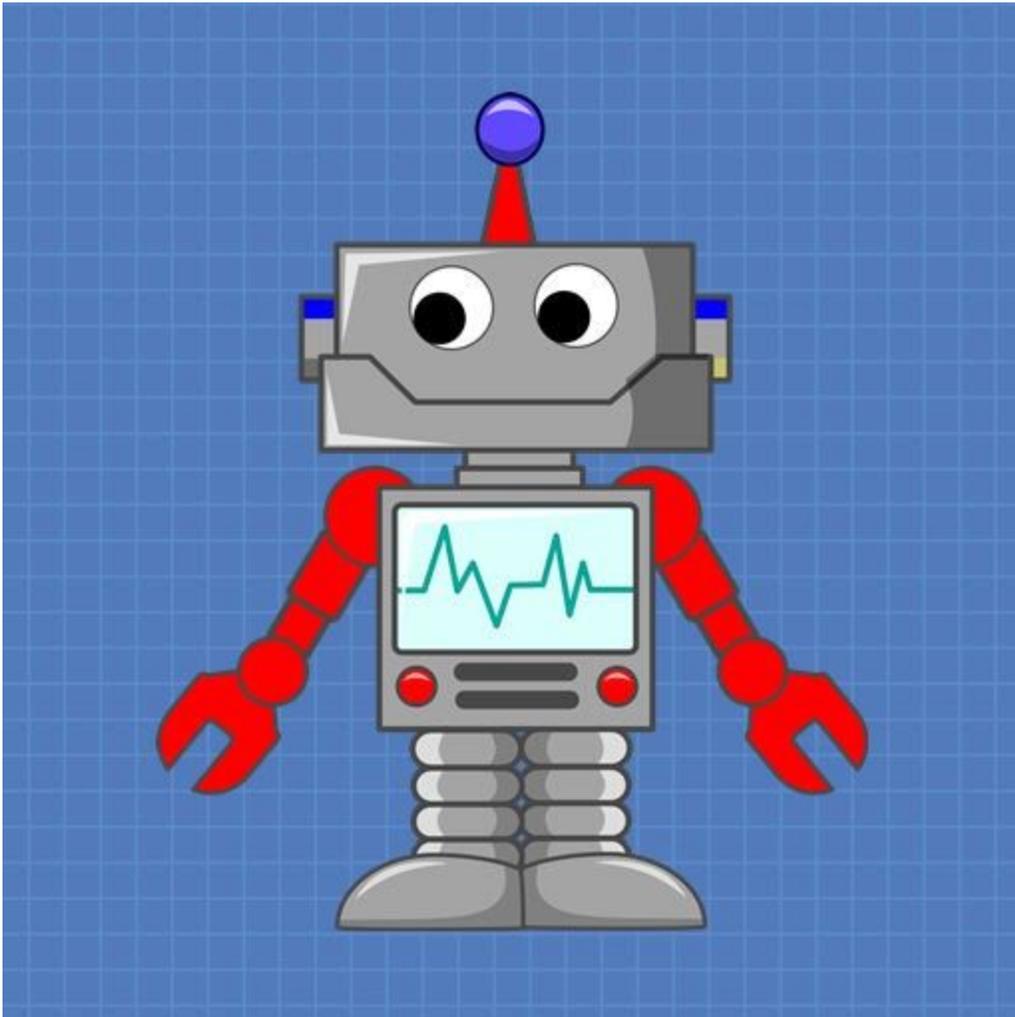


LASER vs Ultrasonic – Distance Sensor Tests



DroneBot Workshop Tutorial

<https://dronebotworkshop.com>

Table of Contents

Distance Sensing	3
Our Contestants – HC-SR04 vs. TOF10120	4
Testing Parameters	5
HC-SR04	6
TOF10120	8
Comparing the two Devices	9
Sensor Test Sketches	11
HC-SR04 Hookup	11
HC-SR04 Code	12
HC-SR04 Quick Check	17
TOF10120 Hookup	18
TOF10120 Code	19
TOF10120 Quick Check	23
Performance Tests	24
DroneBot Workshop Distance Testbed	24
Testing Sequence	25
HC-SR04	25
TOF10120	27
Conclusion	29

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

Today we're having a contest of sorts, a match between two popular types of distance sensors. Get ready to place your bets on the winner!

I've got a bit of a different type of experiment going on in the workshop today, it's a battle between two types of distance sensors. One of them uses lasers and is extremely small, and the other is [our old friend the HC-SR04](#) which chirps out ultrasonic sound waves.

We'll compare the performance of these sensors in the special DroneBot Workshop Distance Tester, constructed especially for this article and video!



We'll compare both devices for accuracy in the 10 – 100 cm range, against both reflective and absorbent surfaces. We'll also compare the spec sheets to see where each sensor shines.

Come and get a good seat for the match and we'll get started!

Distance Sensing

Distance Sensing is a common and even critical function in many modern designs. It's used extensively in robotics but also finds use in a variety of different applications such as security sensors, process control designs, quadcopters, and vehicle safety systems. It even finds a home in hands-free bathroom fixtures.

We have discussed these types of sensors before, including an extensive article about using the popular HC-SR04 ultrasonic distance sensor.

All of these types of sensors use the same basic principle, they just execute it differently:

- A signal is transmitted at the target from the sensor.
- The target reflects the signal back to the sensor.
- The sensor receives the reflected signal.
- The time delay between the sending of the signal and its reception is used to calculate the distance to the target.

The HC-SR04 used ultrasonic sound at a frequency of about 40 kHz, too high for us humans to hear but potentially audible to some animals. Fifi and Fido beware!

Sound is only one of many different mediums that you can use to do this trick. The most well-known application of this principle is RADAR, which uses microwaves.

Light is also a popular method of building a distance sensor. We have already seen this used in a very advanced way when we discussed using LiDAR a while back.

Light sensors have several advantages over their ultrasonic counterparts. They can be made very small and can be waterproof. And because the speed of light is so much

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

quicker than sound, light sensors can respond very quickly and can be queried more often.

But light sensors also require advanced electronics, capable of measuring the incredibly small time differences encountered when working with light. Consider this:

- In one second sound will travel 343 meters (0.343 Km), a little longer than three American football fields placed end-to-end.
- In one second light will travel 299,792,458 meters (299,792.458 Km), about three quarters the distance from the Earth to the Moon.

Advanced light sensors make use of lasers, combine that with the high-speed electronics and the cost of light sensors can be substantially higher than ultrasonic sensors.

Ultrasonic sensors also have some advantages over light sensors. Not only are they lower in cost, they generally have longer measurement ranges, at least in low-cost sensors.

Each type has advantages and disadvantages in different environments, and both can be subject to external interference.

In robotics, it is common to use both types to get the advantages of each type of sensor.

Our Contestants – HC-SR04 vs. TOF10120

It's time now to meet today's contestants – yes that did sound a bit like a game show host speaking, didn't it?

We have our old friend the HC-SR04 Ultrasonic Sensor pitted up against the newcomer, a TOF10120 Laser range finder. This new guy is being offered on all the usual channels (Amazon, eBay, etc) and is very, very tiny.

Testing Parameters

I'm not expecting any real winners or losers today, as each sensor has its strengths and weaknesses. Here is what we'll be comparing:

- **Cost** – How expensive is the device?
- **Interface** – How does the sensor connect up to your microcontroller?
- **Size & Weight** – How large is the sensor, how much does it weigh?
- **Current** – How much current does the sensor consume?
- **Maximum Rated Range** – What is the maximum range the sensor can detect?
- **Minimum Rated Range** – What is the minimum range the sensor can detect?
- **Range Test** – A test of the sensor at 5 preset distances, each with two different target surfaces. How accurate are they?

Let's take a closer look at the two sensors.

HC-SR04

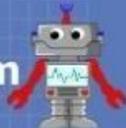
The HC-SR04 is a very popular ultrasonic distance sensor. It is widely available and is a staple of low-cost collision-avoidance robots.



HC-SR04 vs TOF10120

- Range rated 2 cm - 400 cm
- Working Voltage 5 VDC
- Working Current 15 mA
- Dimensions 45*20*15 mm
- 40 KHz sound pulses
- Ultrasonic transducers

4.000
Y1
HC-SR04
T R
VCC Trig Echo GND

DroneBotWorkshop.com 

The image is a blue-themed slide with a white grid background. It features a list of specifications for the HC-SR04 sensor on the left. On the right, there is a photograph of the sensor module with a digital display showing '4.000' and 'Y1'. Below the sensor, the labels 'T' and 'R' are visible, along with the pin headers labeled 'VCC', 'Trig', 'Echo', and 'GND'. The DroneBotWorkshop.com logo, which includes a small robot character, is positioned in the bottom right corner of the slide.

This is a very versatile sensor, I have done a complete article and video devoted solely to the HC-SR04 as it is such an important component. [If you want to bring yourself up to speed on the HC-SR04, then that article is a great place to start.](#)



The HC-SR04 is pretty simple to use. It has four pins:

- VCC – Power, +5 VDC
- Trig – the Trigger input, the input that you use to initiate a measurement. Each time you input a pulse of at least 10ms the sensor sends eight sets of 40 kHz pulses.
- Echo – The received pulse. It stays HIGH while the sensor waits for returned reflection data.
- GND – Ground

You can tie the Trigger and Echo pins together and use the sensor in “3-wire mode” to save on an I/O pin.

The HC-SR04 has a rated range of 2-400 cm.

TOF10120

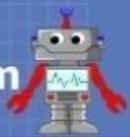
The TOF10120 is a Laser “time of flight” sensor that uses VCSEL (Vertical-Cavity Surface-Emitting Laser) technology to measure distance. It is a very tiny device, measuring only 20 * 13 * 5 mm.



HC-SR04 vs TOF10120

- Range rated 10 cm - 180 cm
- Working Voltage 3 - 5 VDC
- Working Current 35 mA
- Dimensions 20*13*5 mm
- 940 nm wavelength light pulses
- Vertical-Cavity Surface-Emitting Laser (VCSEL)

DroneBotWorkshop.com



This device has a multitude of uses, including in those appliances like hand dryers and automatic sinks which use the sensor to detect movement.

This device communicates over I2C, it also has a serial connection.

The TOF10120 comes complete with an interface cable for use during experiments. It brings out the 6 pins on the sensor. The cable is color-coded as follows:



The device operates on voltages between 3 and 5-volts, so it is suitable for use with 3.3-volt logic. It has a stated range of 10 – 180 cm.

Comparing the two Devices

As these devices use different technologies (sound vs. light) to achieve their goals they each have specific advantages and disadvantages. Here are a few things to consider when choosing which one to use:

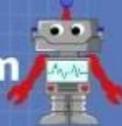
HC-SR04 vs TOF10120



- Low cost (\$1.25 USD)
- Exposed to environment
- 1-pin or 2-pin digital I/O
- Larger size
- Slower response
- Wider field of view

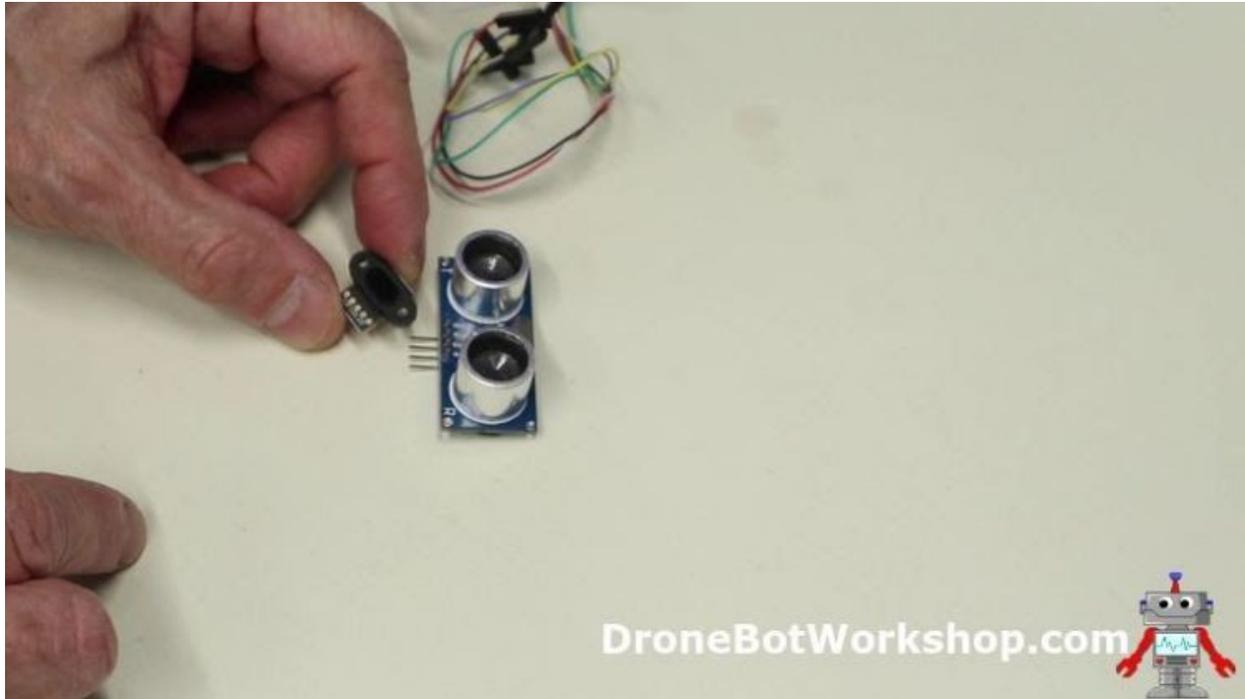
- Higher cost (\$10.00 USD)
- Sealed to environment
- Serial connection & I2C Bus
- Very small
- Quick response
- Narrow field of view

[DroneBotWorkshop.com](https://dronebotworkshop.com)



- The HC-SR04 is less expensive you can buy several of them for the price of one TOF10120.
- The TOF10120 is a sealed unit and is ideal for applications where moisture or dirt may be present.
- The TOF10120 is much smaller than the HC-SR04.
- As light travels much faster than sound the TOF10120 responds much quicker than the HC-SR04.
- The HC-SR054 has a wider field of view, which may or may not be a disadvantage, depending upon your application.
- The TOF10120 can be used with 3.3-volt or 5-volt logic. The HC-SR04 can only be used with 5-volt logic.
- Currently, there is very little English documentation for the TOF10120.

Having said all that, either unit would be a good choice for many common applications.



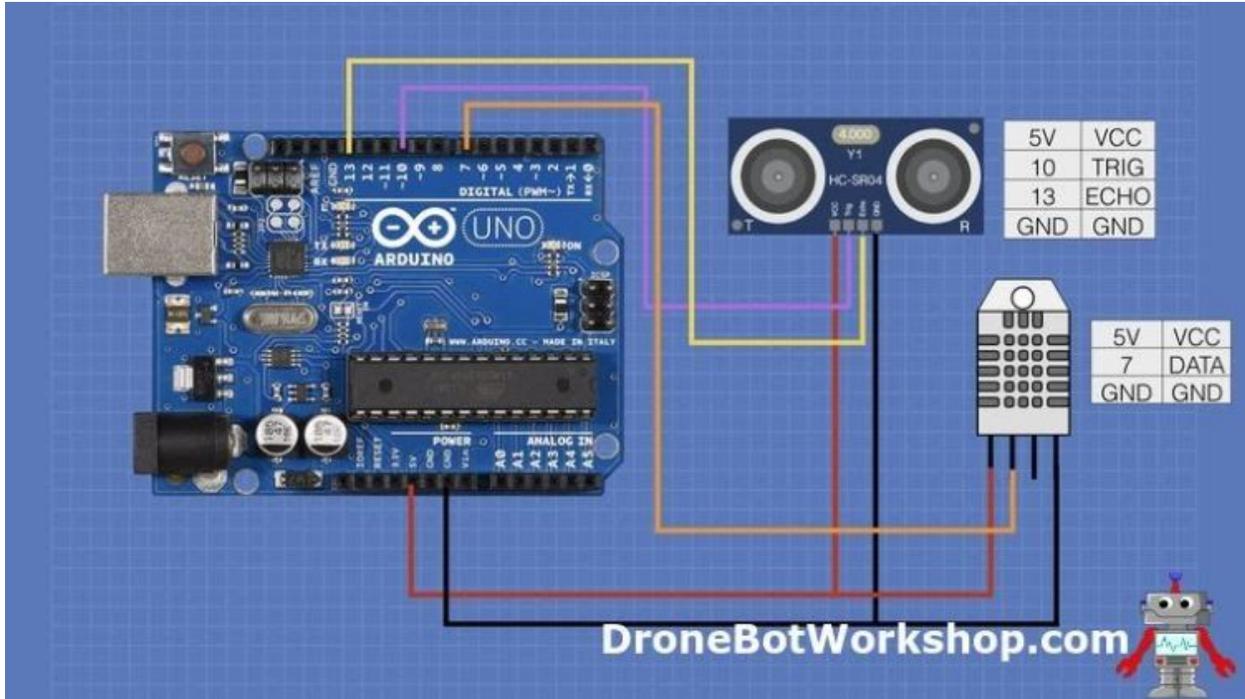
Sensor Test Sketches

To run our tests, we will need a sketch for each of the sensors. Nothing fancy, just something that will read the sensor and report the reading on the serial monitor will be sufficient.

So before we move to the testbed we will first hook up our two sensors and check them out with a simple sketch.

HC-SR04 Hookup

I'm going to use one of the experiments I performed in the article and video I made about the HC-SR04 as my test sketch for the ultrasonic sensor. And I'm going to add an additional component as well.



The extra part is a DHT-22 temperature and humidity sensor. I'm adding this because the speed of sound is influenced by environmental factors like temperature and humidity, so I wanted to factor that in when calculating distance.

I'm running the HC-SR04 in a 4-wire mode in this hookup.

HC-SR04 Code

This sketch is an old one from early 2017, however, it serves its purpose.

```
1  /*
2   HC-SR04 with Temp and Humidity Demonstration
3   HC-SR04-Temp-Humid-Demo.ino
4   Demonstrates enhancements of HC-SR04 Ultrasonic Range Finder
5   With DHT22 Temperature and Humidity Sensor
6   Displays results on Serial Monitor
7
8   DroneBot Workshop 2017
9   http://dronebotworkshop.com
10 */
11
12 // Include DHT Libraries from Adafruit
13 // Dependant upon Adafruit_Sensors Library
14 #include "DHT.h";
15
16 // Include NewPing Library
17 #include "NewPing.h"
18
19 // Define Constants
20
21 #define DHTPIN 7    // DHT-22 Output Pin connection
22 #define DHTTYPE DHT22 // DHT Type is DHT 22 (AM2302)
23 #define TRIGGER_PIN 10
24 #define ECHO_PIN 13
25 #define MAX_DISTANCE 400
26
27 NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
28
29 // Define Variables
```

```
30
31 float hum;    // Stores humidity value in percent
32 float temp;  // Stores temperature value in Celcius
33 float duration; // Stores HC-SR04 pulse duration value
34 float distance; // Stores calculated distance in cm
35 float soundsp; // Stores calculated speed of sound in M/S
36 float soundcm; // Stores calculated speed of sound in cm/ms
37 int iterations = 5;
38
39 // Initialize DHT sensor for normal 16mhz Arduino
40
41 DHT dht(DHTPIN, DHTTYPE);
42
43 void setup() {
44     Serial.begin (9600);
45     dht.begin();
46 }
47
48 void loop()
49 {
50
51     delay(2000); // Delay so DHT-22 sensor can stabilize
52
53     hum = dht.readHumidity(); // Get Humidity value
54     temp= dht.readTemperature(); // Get Temperature value
55
56     // Calculate the Speed of Sound in M/S
57     soundsp = 331.4 + (0.606 * temp) + (0.0124 * hum);
58
```

```
59     // Convert to cm/ms
60
61     soundcm = soundsp / 10000;
62
63     duration = sonar.ping_median(iterations);
64
65     // Calculate the distance
66     distance = (duration / 2) * soundcm;
67
68     // Send results to Serial Monitor
69
70     Serial.print("Sound: ");
71     Serial.print(soundsp);
72     Serial.print(" m/s, ");
73     Serial.print("Humid: ");
74     Serial.print(hum);
75     Serial.print(" %, Temp: ");
76     Serial.print(temp);
77     Serial.print(" C, ");
78     Serial.print("Distance: ");
79
80     if (distance >= 400 || distance <= 2) {
81         Serial.print("Out of range");
82     }
83     else {
84         Serial.print(distance);
85         Serial.print(" cm");
86         delay(500);
87     }
```

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

```
88  
89   Serial.println(" ");  
90 }
```

The sketch requires three libraries, as the *Adafruit DHT Library* is dependent upon the *Adafruit Universal Sensor Library*. Both libraries can be installed using your Arduino IDE Library Manager.

You'll also need to install the *NewPing Library*, again you can install this by searching for it in the Library Manager.

The sketch is pretty simple, if you'd like a detailed explanation of it please see the original article about using the HC-SR04.

Most of the action in the sketch goes on in the Loop. The first thing that occurs here is the temperature and humidity are read from the DHT-22. These are then used to calculate the speed of sound in centimeters per millisecond, factoring in temperature and humidity.

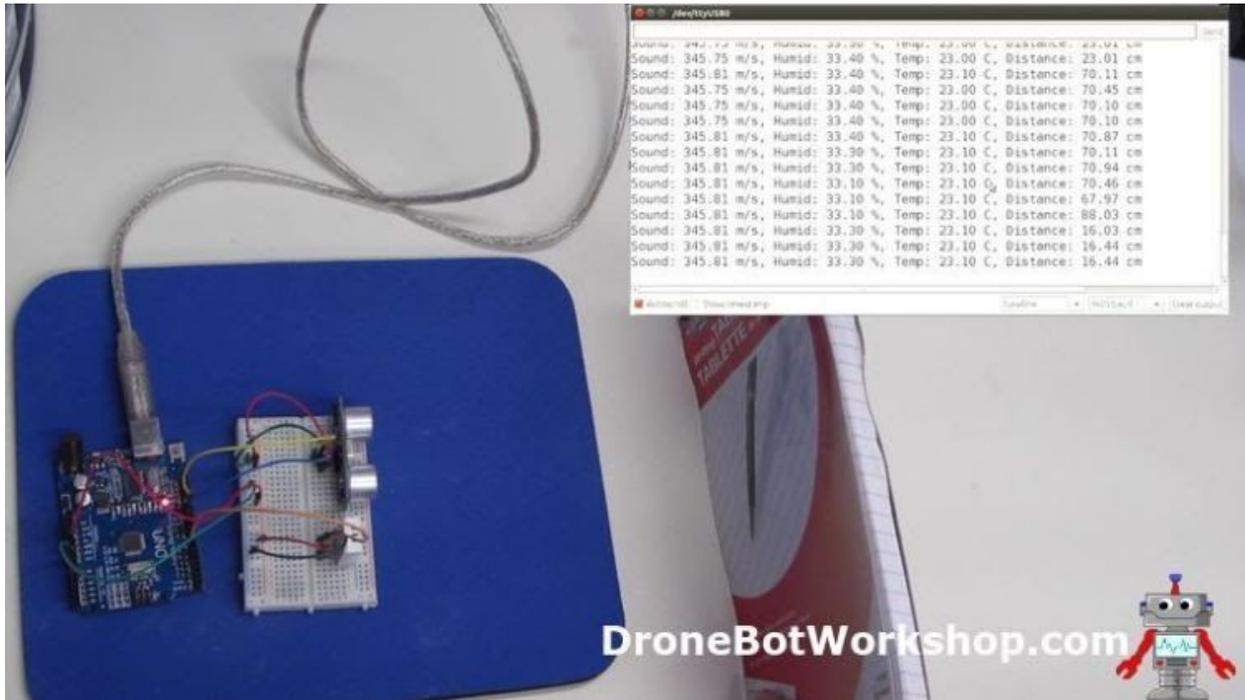
This value is applied to the HC-SR04 time delay reading, which is then divided in half to determine the distance.

You'll note that there are a couple of time delays applied to the readings, to allow both the DHT-22 and HC-SR04 to stabilize between readings. This adds up to a total delay of about 2.5 seconds, so that's how often we can expect a reading.

The results of the distance equation, as well as the temperature and humidity readings, are displayed on the serial monitor.

HC-SR04 Quick Check

So let's give the ultrasonic sensor a quick test.

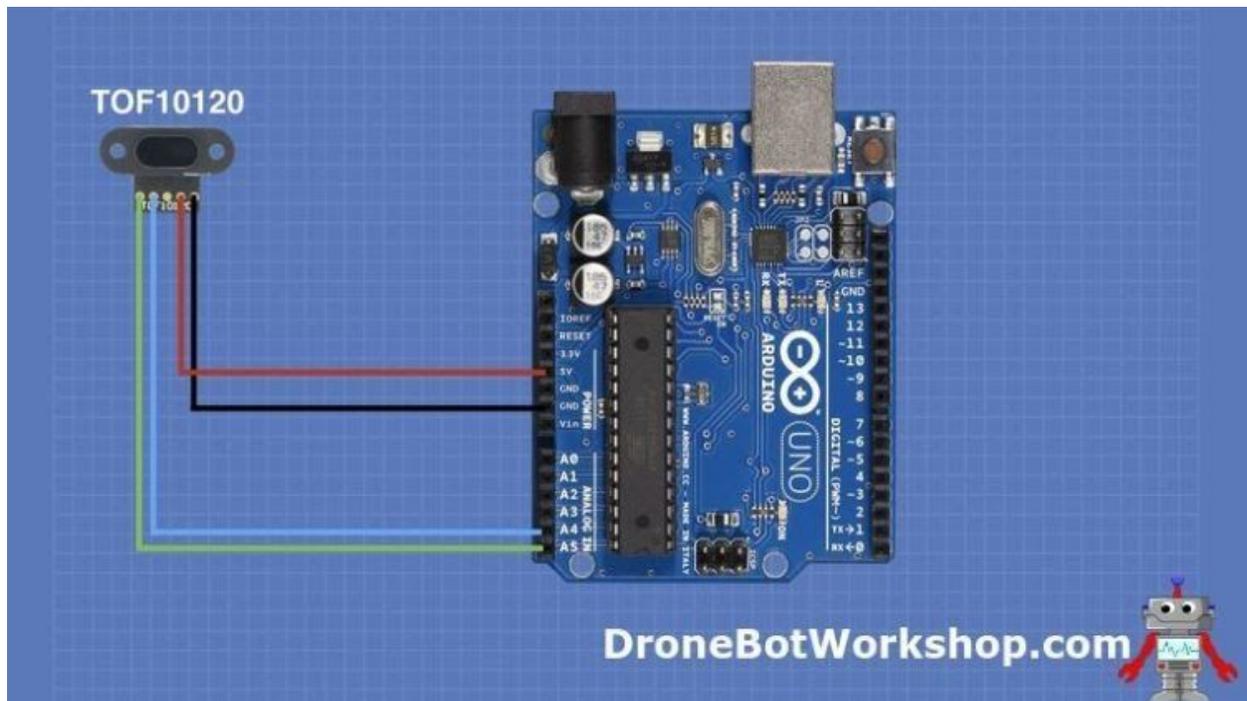


Hook up everything as shown in the diagram and load the sketch, being sure to remember to install the three libraries first. Then use a test object as a target, I used a notepad when I did this.

The resulting readings seemed reasonable, so I'll consider our HC-SR04 to be working properly.

TOF10120 Hookup

Hooking up the TOF10120 is very simple, thanks to the cable provided with the unit. As the cable uses female Dupont pins I just used small jumper wires to connect it directly to my Arduino.



As we are using the I2C connection there are only four connections to the Arduino – power, ground, SDA, and SCL. I did not use pull-up resistors and everything seemed to work correctly.

There are two other connections to the sensor serial port that are left unconnected in this experiment.

TOF10120 Code

One difficulty in working with the TOF10120 is a lack of documentation. While there is a wealth of documentation, app notes and sample code for the HC-SR04 there is very little available for the TOF10120, at least in English.

The following code came from a YouTube user named “SurtrTech” who claims he/she obtained it from a Chinese website. Unfortunately, no link to the actual source was provided, so I can’t credit the original author.

```
1  /*
2   TOF10120 Distance Sensor Demonstration
3   TOF10120-Demo.ino
4   Demonstrates use of TOF10120 Distance Sensor
5   Adapted from code from SurtrTech
6   Displays results on Serial Monitor
7
8   DroneBot Workshop 2019
9   https://dronebotworkshop.com
10 */
11
12 #include <Wire.h>
13
14 unsigned char ok_flag;
15 unsigned char fail_flag;
16
17 unsigned short lenth_val = 0;
18 unsigned char i2c_rx_buf[16];
19 unsigned char dirsend_flag=0;
20
21
22 void setup() {
23   Wire.begin();
24   Serial.begin(9600,SERIAL_8N1);
25   printf_begin();
26
27 }
28
29 void loop() {
```

```
30
31
32     int x=ReadDistance();
33
34     Serial.print(x);
35     Serial.println(" mm");
36
37 }
38
39 int serial_putc( char c, struct __file * )
40 {
41     Serial.write( c );
42     return c;
43 }
44
45 void printf_begin(void)
46 {
47     fdevopen( &serial_putc, 0 );
48 }
49
50
51
52 void SensorRead(unsigned char addr,unsigned char* datbuf,unsigned char cnt)
53 {
54     unsigned short result=0;
55     // step 1: instruct sensor to read echoes
56     Wire.beginTransmission(82); // transmit to device #82 (0x52)
57     // the address specified in the datasheet is 164 (0xa4)
58     // but i2c adressng uses the high 7 bits so it's 82
```

```
59   Wire.write(byte(addr));      // sets distance data address (addr)
60   Wire.endTransmission();      // stop transmitting
61   // step 2: wait for readings to happen
62   delay(1);                    // datasheet suggests at least 30uS
63   // step 3: request reading from sensor
64   Wire.requestFrom(82, cnt);   // request cnt bytes from slave device #82
65   (0x52)
66   // step 5: receive reading from sensor
67   if (cnt <= Wire.available()) { // if two bytes were received
68       *datbuf++ = Wire.read(); // receive high byte (overwrites previous reading)
69       *datbuf++ = Wire.read(); // receive low byte as lower 8 bits
70   }
71 }
72
73 int ReadDistance() {
74     SensorRead(0x00, i2c_rx_buf, 2);
75     lenth_val=i2c_rx_buf[0];
76     lenth_val=lenth_val<<8;
77     lenth_val|=i2c_rx_buf[1];
78     delay(300);
79     return lenth_val;
}
```

The sketch doesn't require any additional libraries, the only library it uses is the Arduino Wire Library which simplifies using I2C.

The real action in this sketch occurs in the *SensorRead* function. If you step through this function you'll see how the I2C bus is used to contact the sensor, request a reading and add the result to a data buffer.

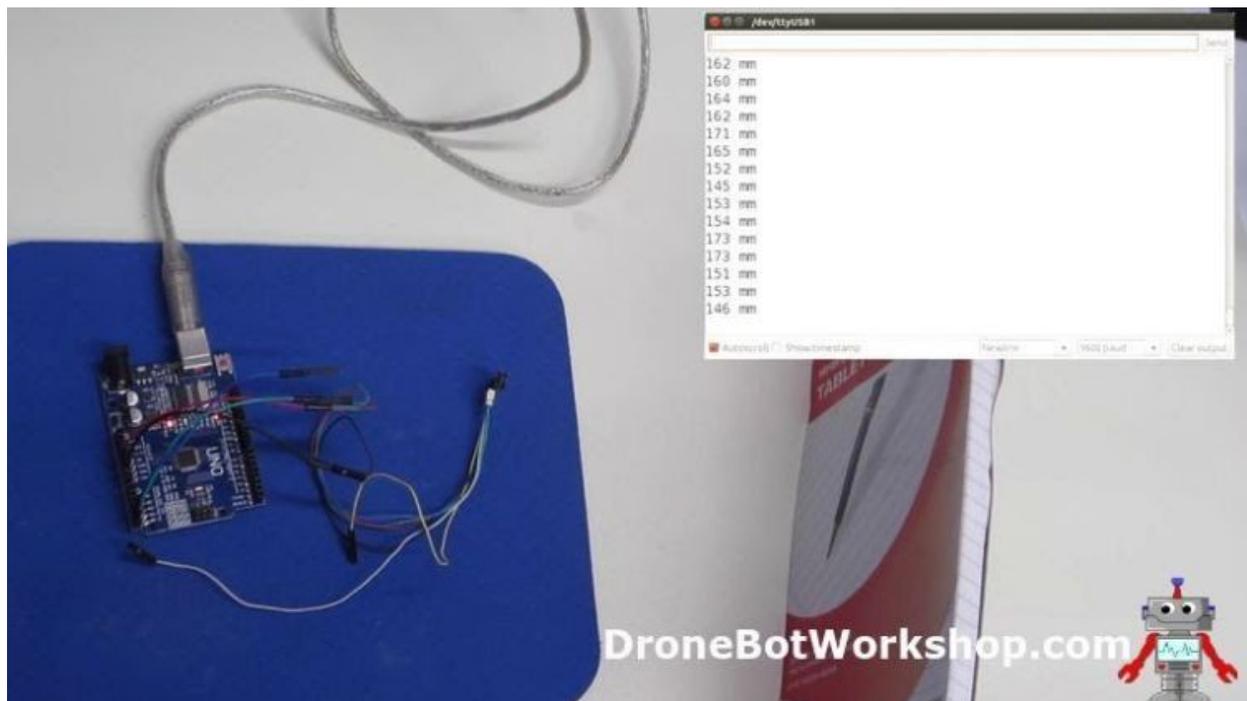
For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

Another function, *ReadDistance*, calls the *SensorRead* function every time it wants a reading. It then takes the value off the buffer, delays 300 ms and then outputs it as an integer. The integer will be the distance in millimeters.

The Loop just calls *ReadDistance* and prints the value to the serial monitor.

TOF10120 Quick Check

After wiring up the TOF10120 and loading the sketch I tested everything out, again using a notepad as a target.



The device responded quickly every time I moved my target, so it looks like it's working.

Note the two unconnected wires from the sensor, these are for the serial I/O port.

Performance Tests

Now that we have our two distance sensors working it's time to get to the actual competition!

Ideally, to do this test properly, I would have used a large room with controlled temperature, humidity and light levels. Unfortunately, that isn't feasible, so I did the tests on my workbench. It is certainly not as scientific, but it does represent more of a "real world" situation.

DroneBot Workshop Distance Testbed

To conduct these tests I created the "DroneBot Workshop Distance Testbed", which is an eloquent name for some scrap wood from the workshop!

The testbed consists of a 4-foot long piece of particle board that has several wood spacers glued to it. These spacers are made to hold a reflector block in several pre-measured positions.

The reflector block itself is a piece of wood with two different surfaces. On one side there is a small piece of plywood, which provides a smooth surface. On the other side, I affixed a piece of foam, the same thickness as the plywood. This provides an absorbent irregular surface.

The sensors and the Arduino Uno driving them are placed on an elevated surface at the "zero-point" of the testbed. The surface is elevated so that the bottom board itself doesn't interfere with the sensors.

Testing Sequence

To keep things at an even playing field, I'm going to calibrate each sensor at the 25 cm mark with the reflective (plywood) surface. Then we will test the sensors in all the remaining positions, with both the plywood and foam sides.

The preset positions are as follows:

- 10 cm
- 25 cm
- 50 cm
- 75 cm
- 100 cm (1 meter)

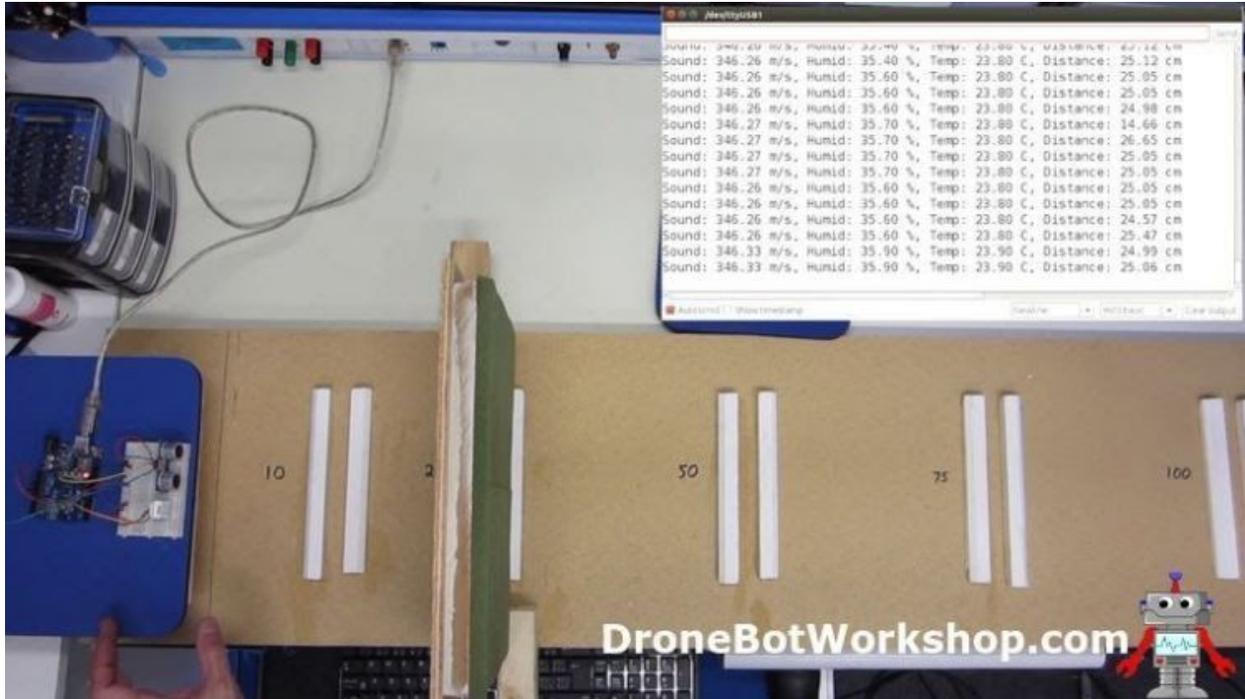
I'm limited to a one-meter test due to the size of my workbench!

So let's see how the two sensors performed.

HC-SR04

The ultrasonic sensor performed surprisingly well, although I did add the DHT-22 temperature and humidity sensor to help improve its performance.

After calibrating the sensor at 25 cm I tested it at 10 cm and 50 cm. Both readings were well within a centimeter accuracy, which is pretty good for a cheap sensor.



After trying to move past 50 cm, however, I experienced problems. I know from previous experience that the sensor is capable of much greater distances than that, but on my testbed, I couldn't get much past the 50 cm mark. So all of the reading above that were pretty well meaningless.

What I expect is happening is that the sensor is triggering on something besides the testbed on my workbench. This sensor has a pretty wide field of view, and so it is likely hitting on either the parts cabinets or myself. After moving out of the way without any effect on the readings I'm blaming the parts cabinets!

Things got more interesting when I tried the foam side of the reflector.

As I had suspected, the sound waves don't reflect off of the foam. Instead, they are bouncing off the surface of the wooden block, passing through the foam as if it wasn't there.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

As a result, the tests at 10, 25 and 50 cm were almost a centimeter off. Both my plywood and foam are a quarter-inch thick, which is about a sixth of a centimeter, and that seemed to account for the difference.

As I knew that the tests after 50 cm were pointless I didn't bother to record them.

What impresses me about this sensor is that it performed really well until I ran into the interference issue. When you consider it's price it's no wonder it's such a popular device.

TOF10120

The Laser time-of-flight sensor has a range of 10 cm to 180 cm, so I was curious how it would perform at the 10 cm mark. It has millimeter accuracy so 10 cm is 100 mm, and it came within 5 mm of the mark.

The sensor also did very well at the 50 cm mark.



But on the higher points, 75 and 100 centimeters, the readings were off by a great deal, over 10%.

The documentation that I was able to obtain for the sensor claims that the range and accuracy are affected by many factors, including ambient light and the reflectivity of the surface being detected.

So I tried the experiments with the lights out and with a white (and black) sheet of paper over the plywood. Neither seems to change the results.

One of the more interesting results was what happened when I used the foam side.

I had expected that the sensor might get fooled by the foam and give erratic readings. But the readings were pretty steady, but they were all out by the same factor – the thickness of the foam.



It seems the laser beam passed right through the foam and hit the underlying board. Just as the ultrasonic sound waves did. I hadn't expected that.

Perhaps my foam is "invisible" in infrared light?

One thing that can't help but impress you when using this sensor is how fast the data gets updated. It really would be great for moving objects, like robots, quadcopters and model boats.

Conclusion

Both sensors performed well in the 10 – 50 cm range, and both of them were affected by the foam side of the test target.

The ultrasonic sensor readings above 50 cm were meaningless, probably due to external interference from my workbench.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

The laser sensor accuracy after 50 cm really dropped, although since my next test point was at 75 cm I 'm not certain exactly where the drop was.

When factoring in the cost difference the HC-SR04 is a clear winner, providing that you don't require very fast readings or an environmentally-sealed sensor. But you also need to remember that I use dan DHT-22 temperature and humidity sensor in my circuit. If you add in the DHT-22 cost the two sensors come out to be within a couple of dollars of each other.

The high data rate of the TOF10120 is impressive, and I know I'll be making use of one in a robotics project soon.

In fact, I think an even better idea is to build a detector that uses both of these sensors, gaining the advantages of each one. I'll get to work on that right now!