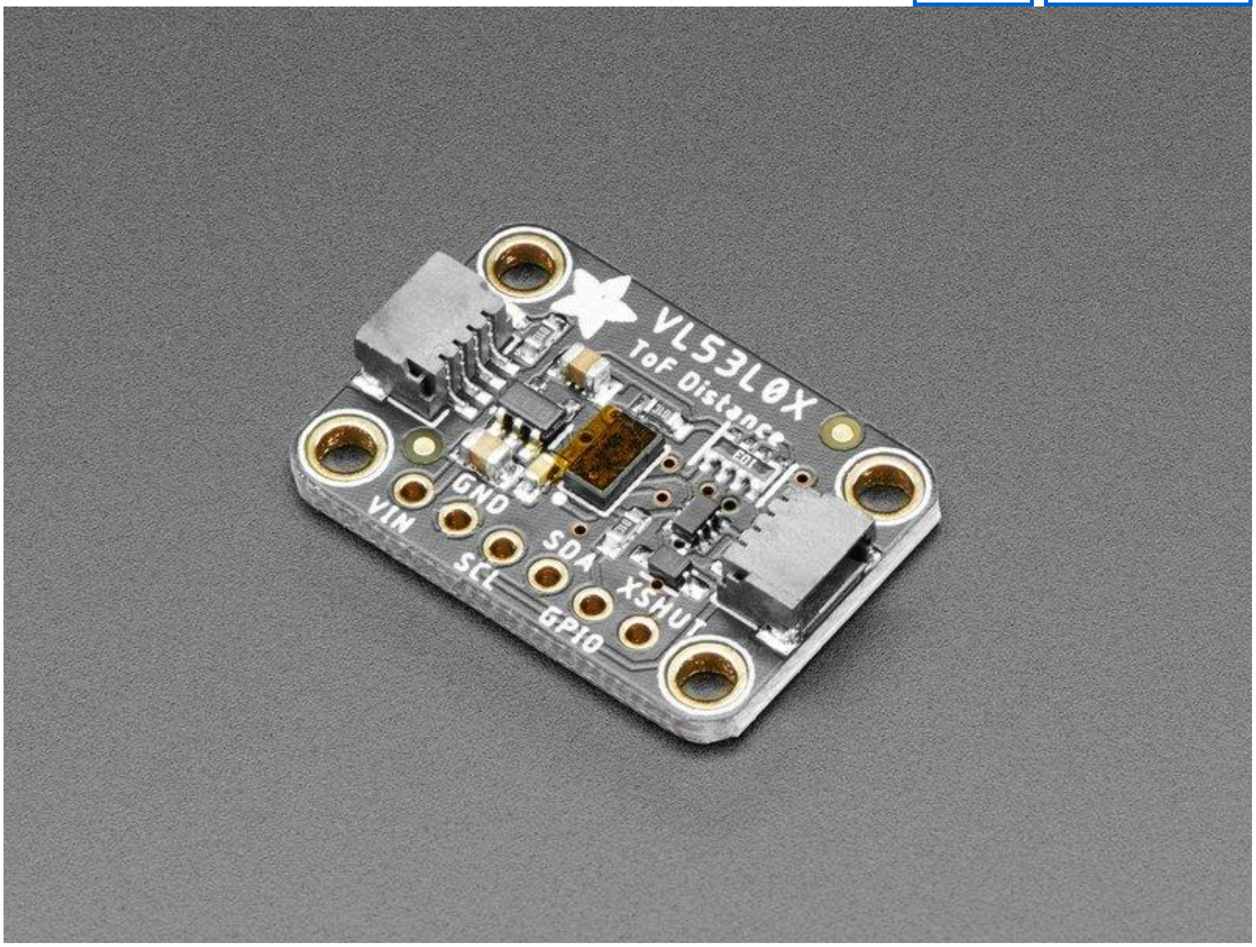
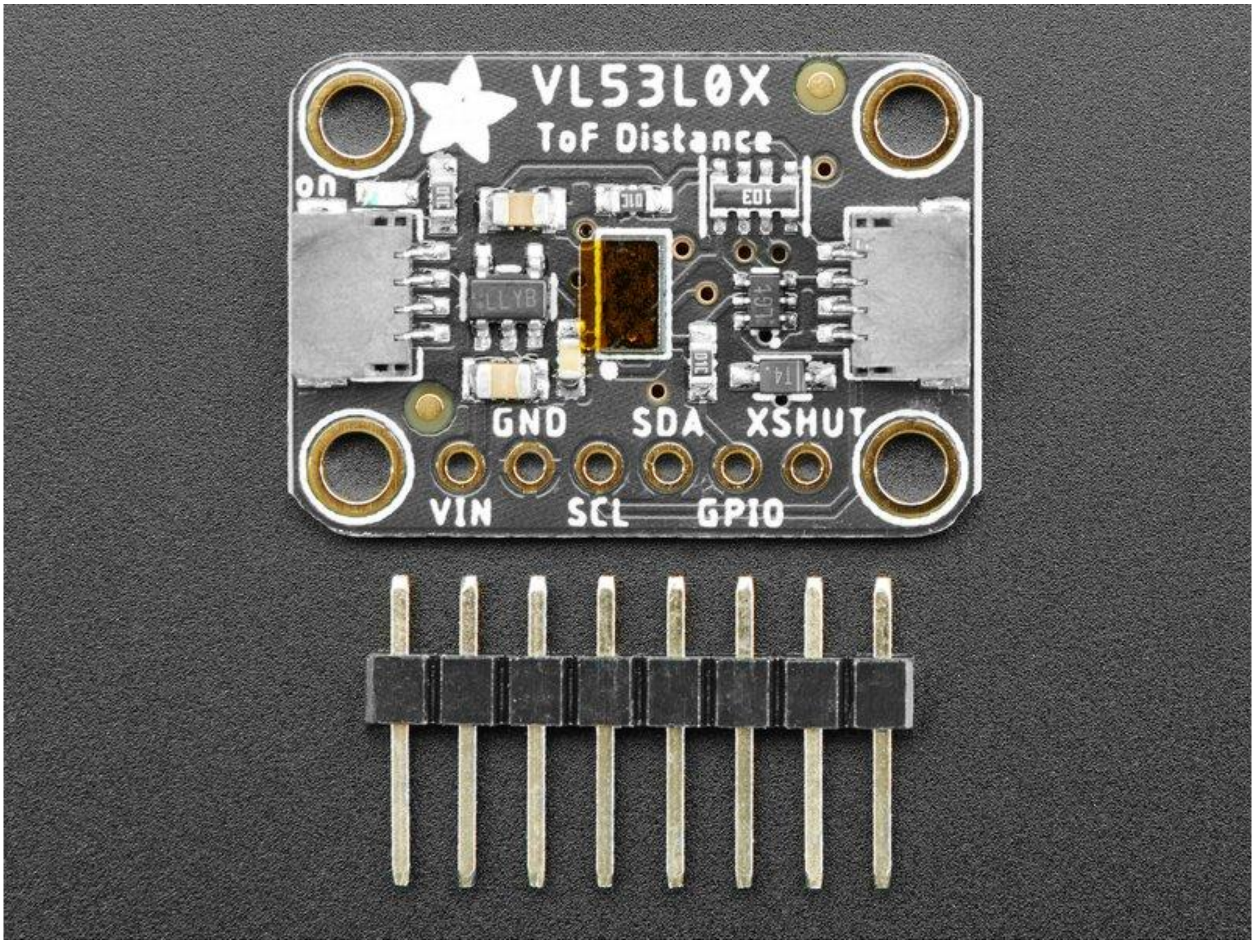
Overview

Save Subscribe

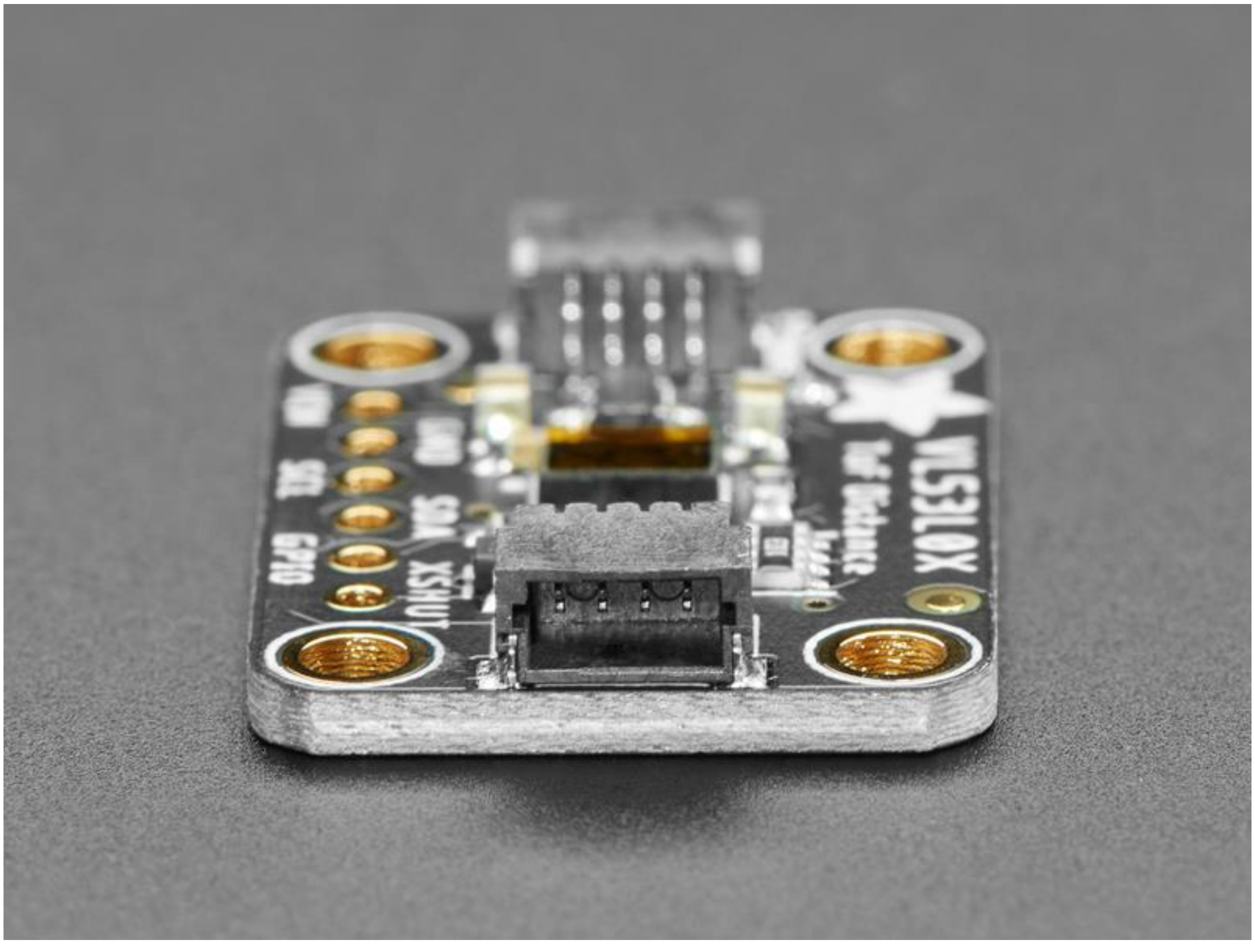


The VL53L0X is a *Time of Flight* distance sensor like no other you've used! The sensor contains a very tiny invisible laser source, and a matching sensor. The VL53L0X can detect the "time of flight", or how long the light has taken to bounce back to the sensor. Since it uses a very narrow light source, it is good for determining distance of only the surface directly in front of it. Unlike sonars that bounce ultrasonic waves, the 'cone' of sensing is very narrow. Unlike IR distance sensors that try to measure the amount of light bounced, the VL53L0x is much more precise and doesn't have linearity problems or 'double imaging' where you can't tell if an object is very far or very close.

This is the 'big sister' of the VL6180X ToF sensor, and can handle about 50 - 1200 mm of range distance. If you need a smaller/closer range, check out the VL6180X which can measure 5mm to 200mm.

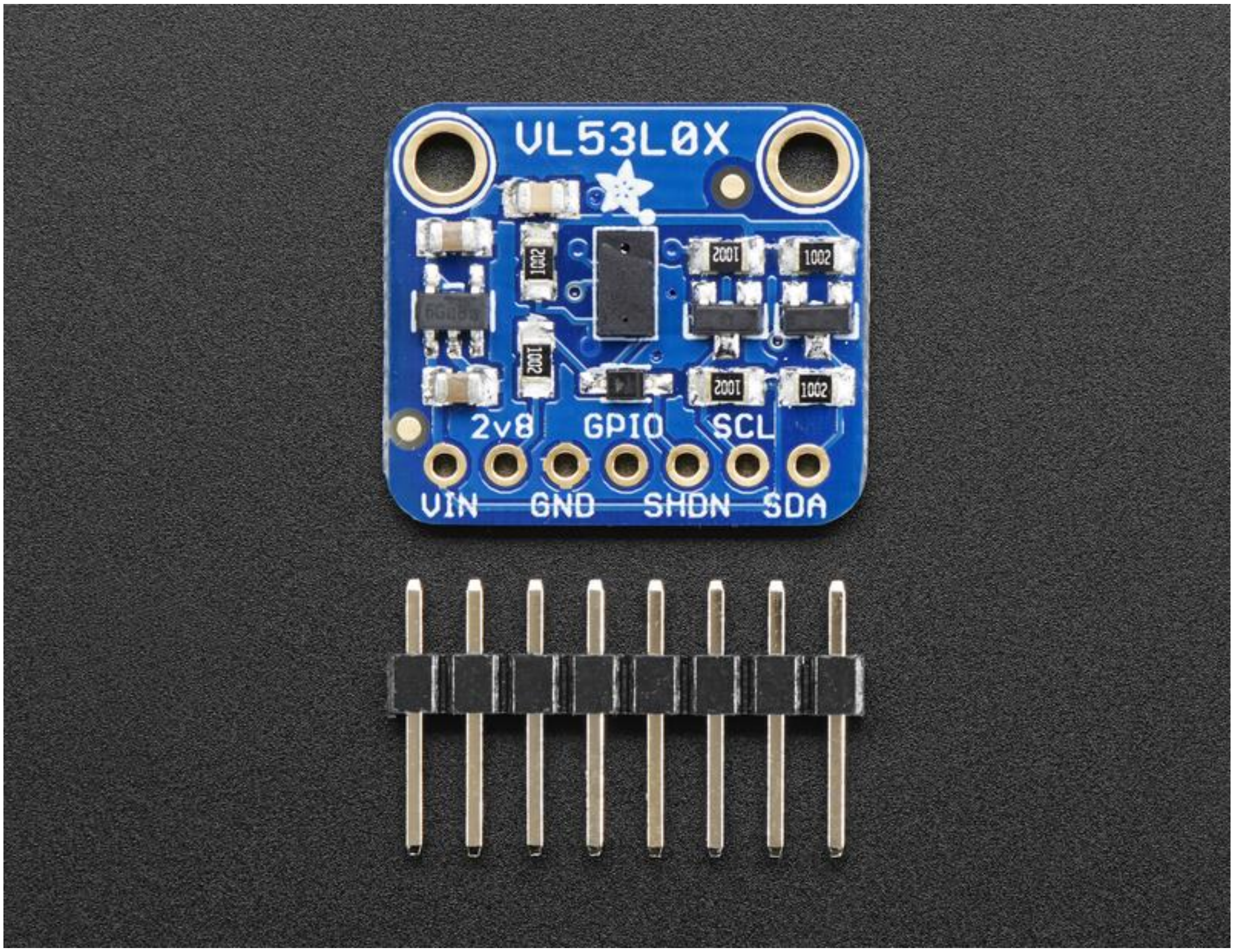


The sensor is small and easy to use in any robotics or interactive project. Since it needs 2.8V power and logic we put the little fellow on a breakout board with a regulator and level shifting. You can use it with any 3-5V power or logic microcontroller with no worries. Each order comes with a small piece of header. Solder the header onto your breakout board with your iron and some solder and wire it up for instant distance-sensing-success!



As if that weren't enough, we've also added [SparkFun qwiic] compatible [STEMMA QT] connectors for the I2C bus **so you don't even need to solder.** Just wire up to your favorite micro with a plug-and-play cable to get 6-DoF data ASAP. For a no-solder experience, [just wire up to your favorite micro, like the STM32F405 Feather] using a [STEMMA QT adapter cable.] The Stemma QT connectors also mean the VL53L0X can be used with our [various associated accessories.]

There are two versions of this board - the STEMMA QT version shown above, and the original header-only version shown below. Code works the same on both!

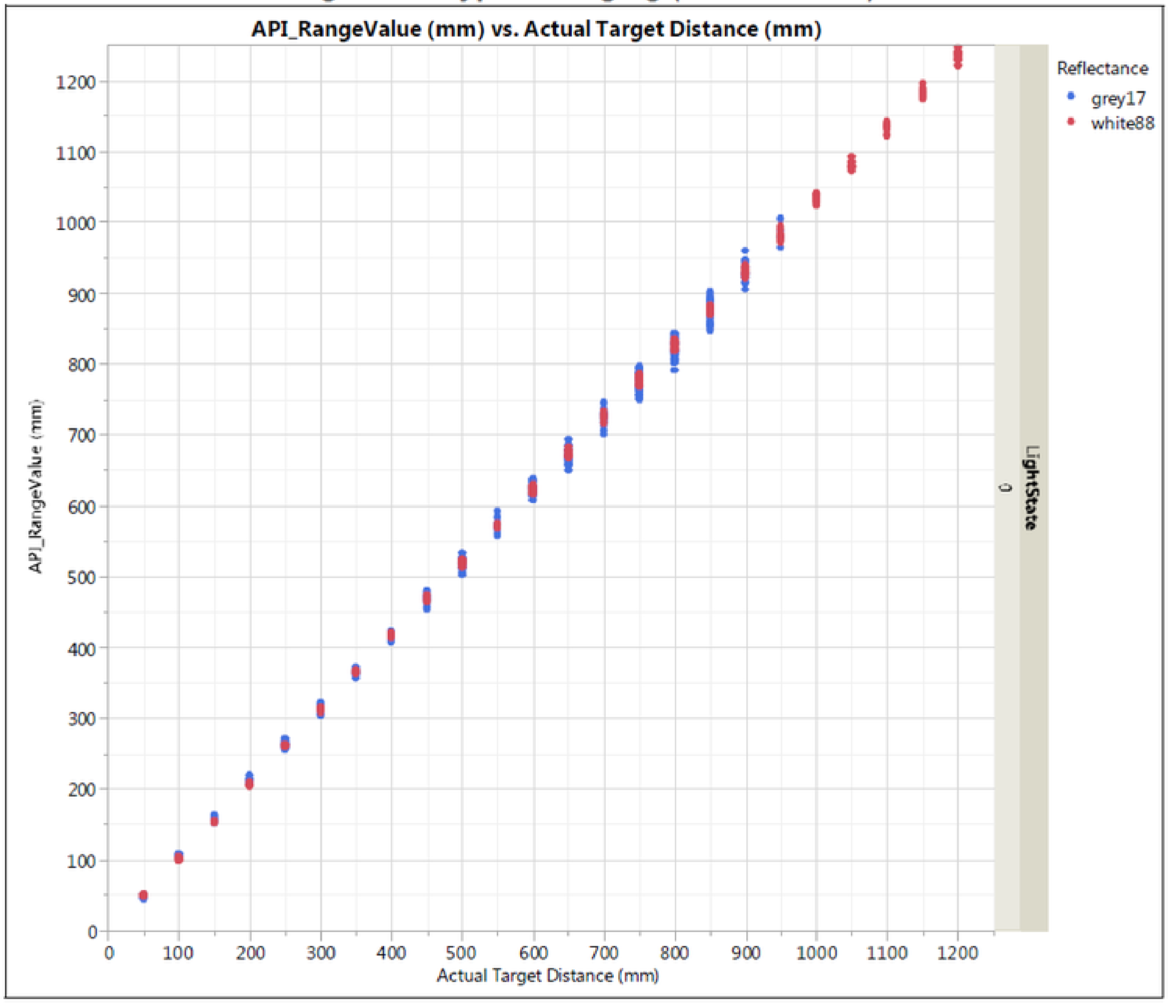


Communicating to the sensor is done over I2C with an API written by ST, so its not too hard to port it to your favorite microcontroller. We've written a wrapper library for Arduino so you can use it with any of your Arduino-compatible boards.

Sensing Capablities

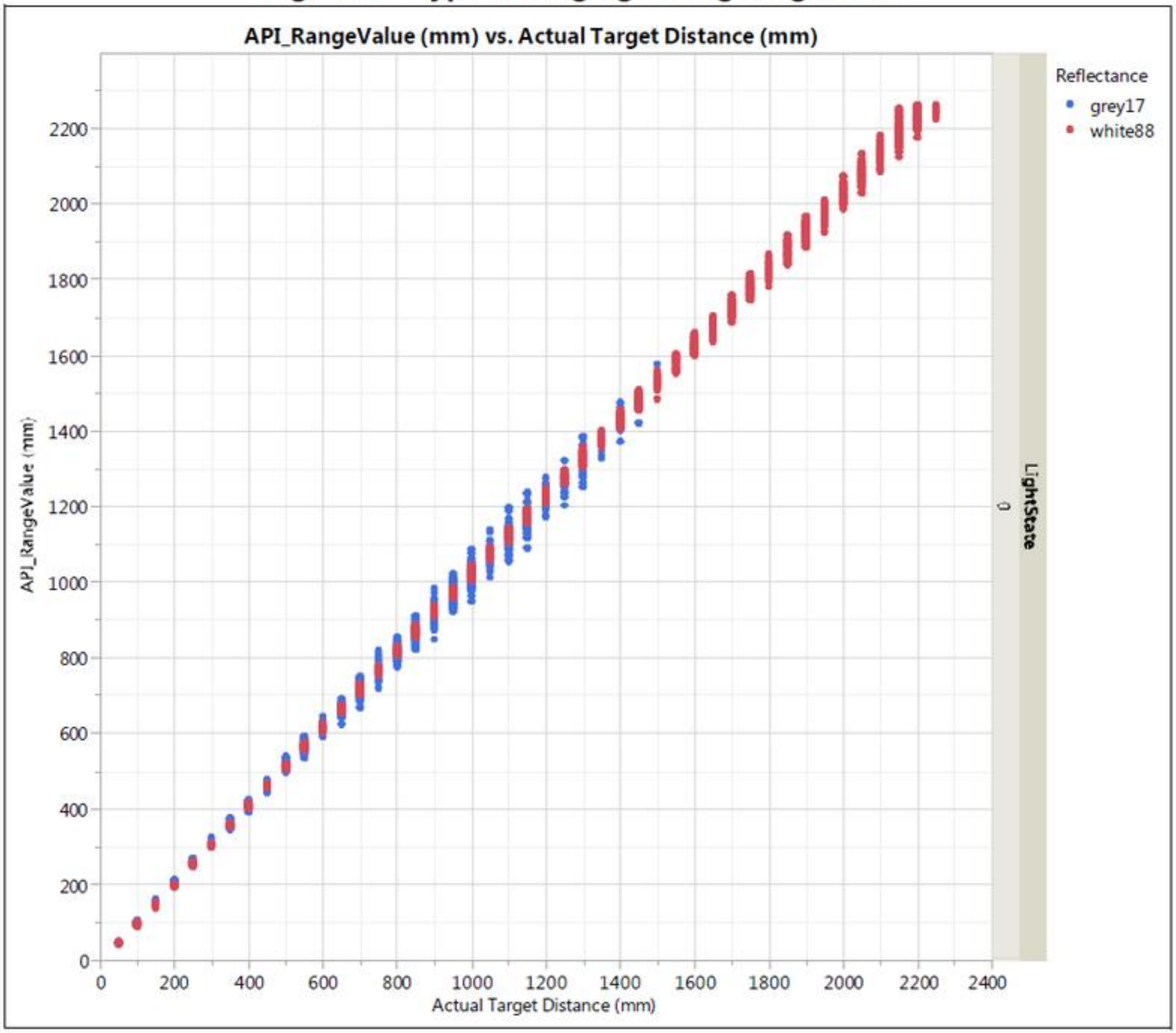
The sensor can measure approximately 50mm to 1.2 meter in default mode.

Figure 20. Typical ranging (default mode)



In 'long range' mode you can detect as far as 1.5 to 2 meters on a nice white reflective surface.

Figure 21. Typical ranging - Long range mode



Depending on ambient lighting and distance, you'll get 3 to 12% ranging accuracy - better lighting and shiny surfaces will give you best results. Some experimentation will be necessary since if the object absorbs the laser light you won't get good readings.

Table 11. Max ranging capabilities with 33ms timing budget

Target reflectance level (Full FOV)	Conditions	Indoor (2)	Outdoor overcast (2)
White Target (88%)	Typical	200cm+ (1)	80cm
	Minimum	120cm	60cm
Grey Target (17%)	Typical	80cm	50cm
	Minimum	70cm	40cm

Note (1): using long range API profile

Table 12. Ranging accuracy

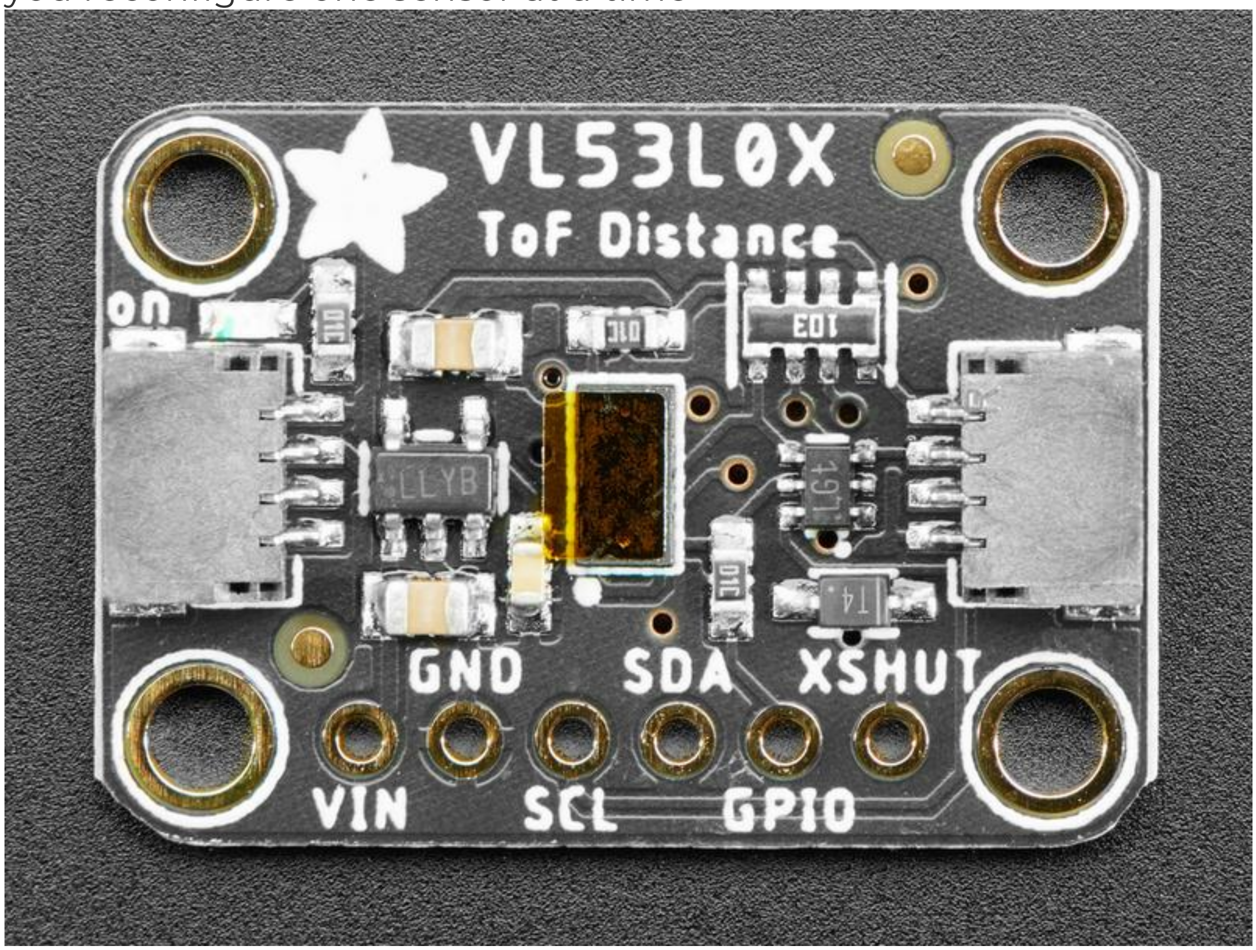
	Indoor (no infrared)			Outdoor		
Target reflectance level (Full FOV)	Distance	33ms	66ms	Distance	33ms	66ms
White Target (88%)	at 120cm	4%	3%	at 60cm	7%	6%
Grey Target (17%)	at 70cm	7%	6%	at 40cm	12%	9%

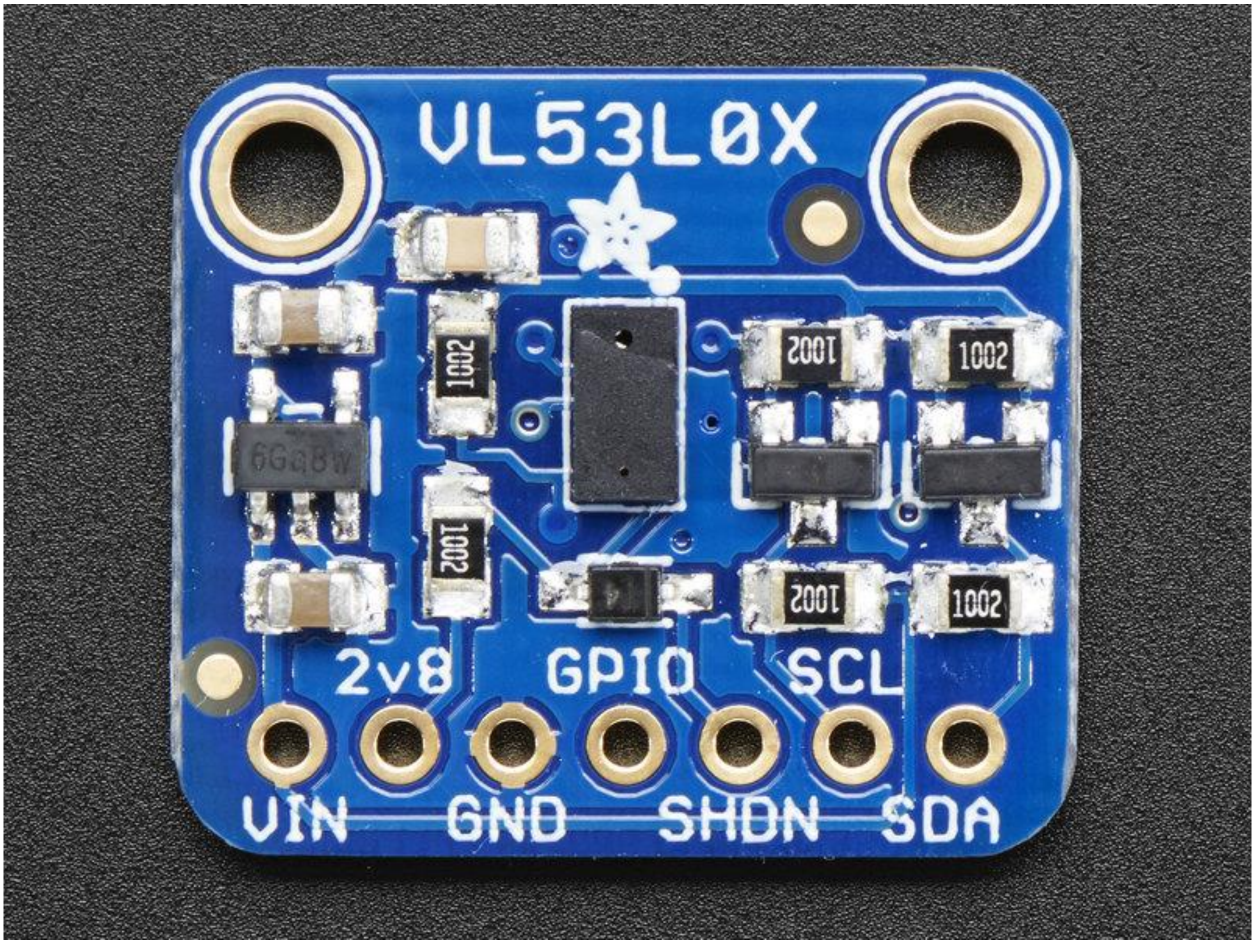
Pinouts

Save Subscribe

The VL53L0X is a I2C sensor. That means it uses the two I2C data/clock wires available on most microcontrollers, and can share those pins with other sensors as long as they don't have an address collision.

For future reference, the default I2C address is **0x29**.
You *can* change it, but only in software. That means you have to wire the SHUTDOWN pin and hold all but one sensor in reset while you reconfigure one sensor at a time





Power Pins:

- **Vin** - this is the power pin. Since the chip uses 2.8 VDC, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V micro like Arduino, use 5V
- **2v8** - this is the 2.8V output from the voltage regulator, you can grab up to 100mA from this if you like
- **GND** - common ground for power and logic

I2C Logic pins:

- **SCL** - I2C clock pin, connect to your microcontrollers I2C clock line.

- **SDA** - I2C data pin, connect to your microcontrollers I2C data line.
- STEMMA QT **-** These connectors allow you to connectors to dev boards with **STEMMA QT** connectors or to other things with various associated accessories.

Control Pins:

- **GPIO** - this is a pin that is used by the sensor to indicate that data is ready. It's useful for when doing continuous sensing. Note there is no level shifting on this pin, you may not be able to read the 2.8V-logic-level voltage on a 5V microcontroller (we could on an arduino UNO but no promises). Our library doesn't make use of this pin but for advanced users, it's there!
- **SHDN** - the shutdown pin for the sensor. By default it's pulled high. There's a level-shifting diode so you can use 3-5V logic on this pin. When the pin is pulled low, the sensor goes into shutdown mode.