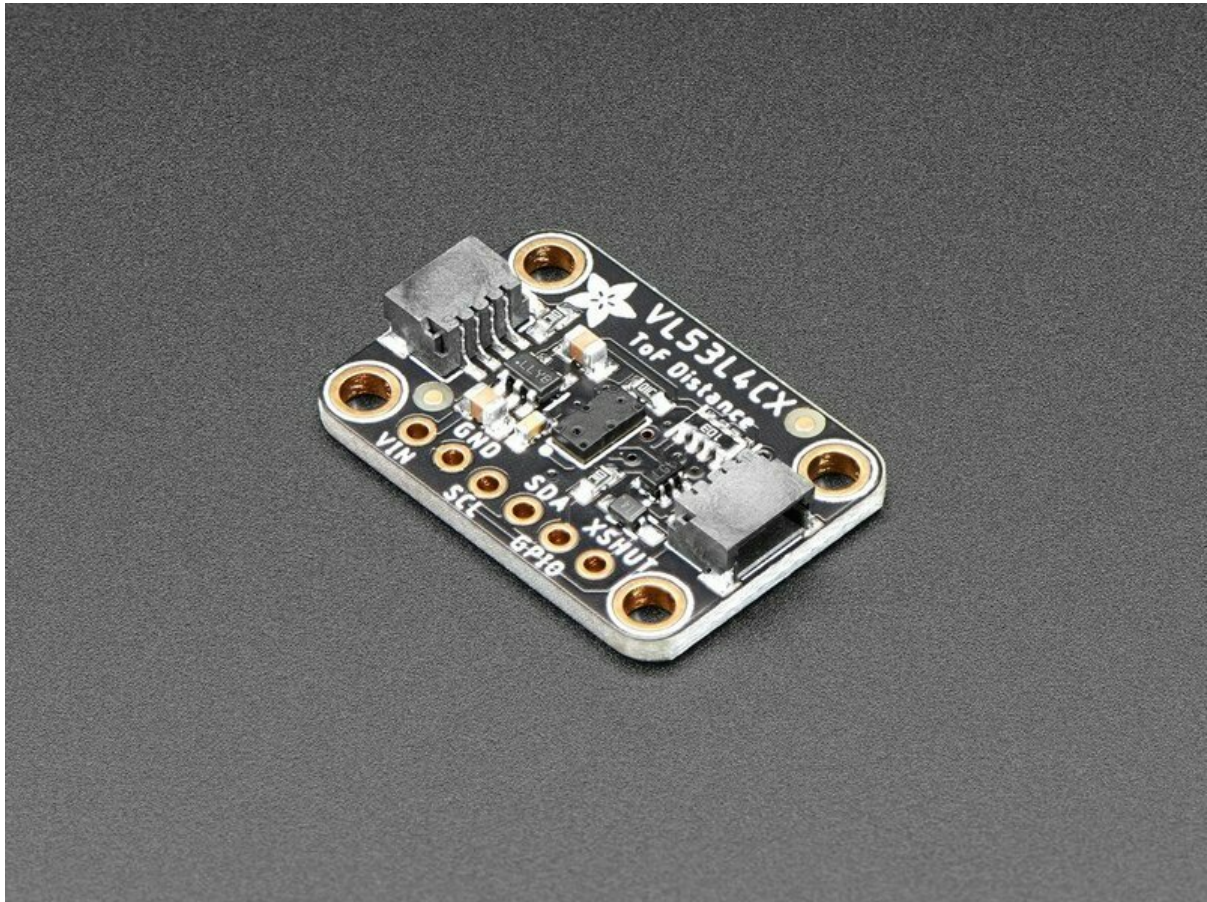




Adafruit VL53L4CX Time of Flight Distance Sensor

Created by Liz Clark



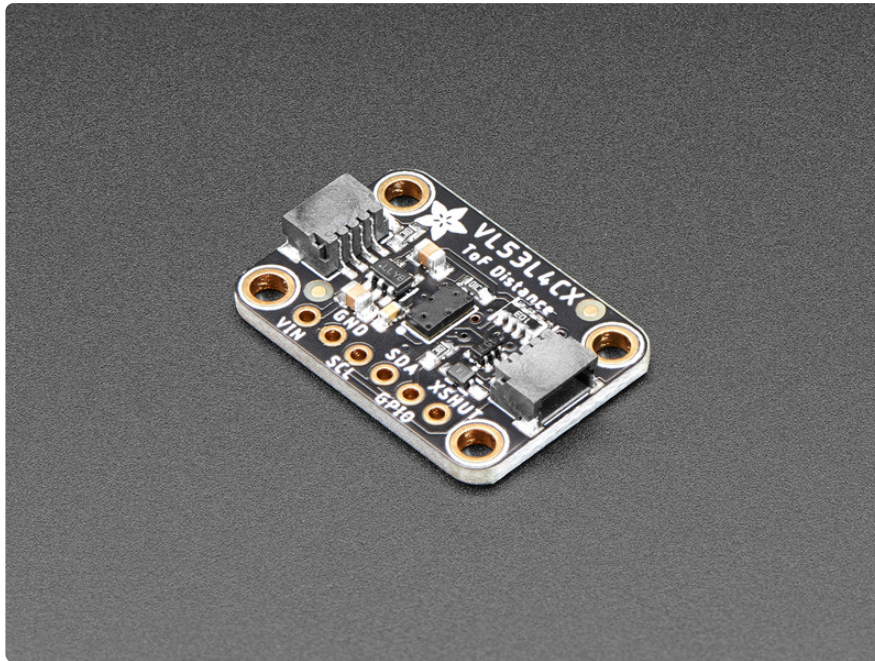
<https://learn.adafruit.com/adafruit-vl53l4cx-time-of-flight-distance-sensor>

Last updated on 2024-03-08 04:05:03 PM EST

Table of Contents

Overview	3
<hr/>	
• Removing the Protective Tape	
Pinouts	6
<hr/>	
• Power Pins	
• I2C Logic Pins	
• Other Pins	
• Power LED and Jumper	
Arduino	7
<hr/>	
• Wiring	
• Library Installation	
• Load Example	
Arduino Docs	10
<hr/>	
Downloads	10
<hr/>	
• Files	
• Schematic and Fab Print	

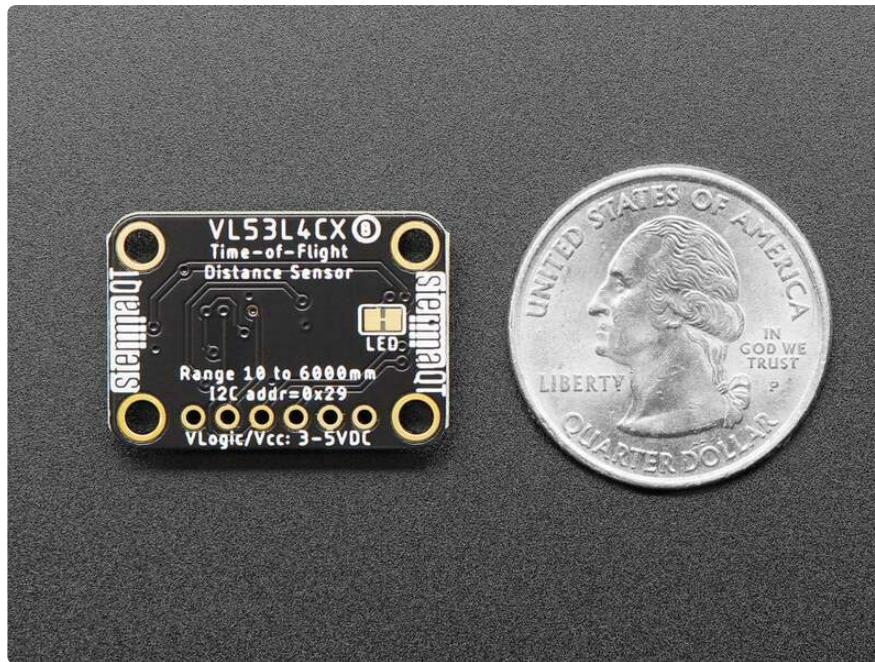
Overview



The **Adafruit VL53L4CX Time of Flight Sensor** is another great [Time of Flight \(https://adafruit.it/ZZD\)](https://adafruit.it/ZZD) distance sensor from ST in the VL5 series of chips. This one is great for **long distances** - it goes up to 6 meters compared with 4 meter max of the VL53L1X.

The sensor contains a very tiny invisible laser source and a matching sensor. The VL53L4CX 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 VL53 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.



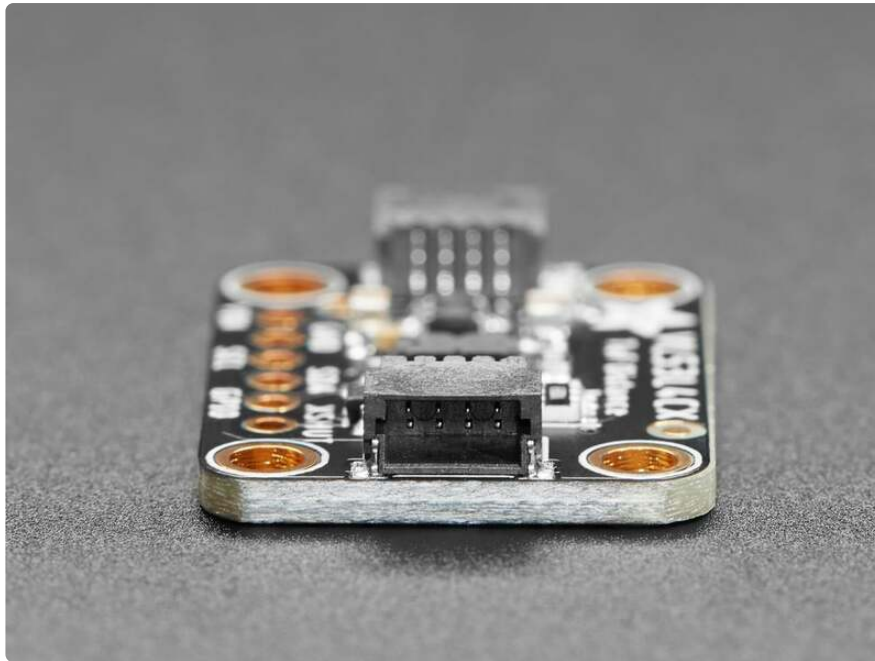
This breakout ships with a protector over the sensor. It must be removed before use! See details below.

This is a 'big sister' of the [VL53L4CD ToF sensor](http://adafru.it/5396) (<http://adafru.it/5396>) and can handle about **~1 to 6000mm** of range distance, it also has some ability to do 'multi object detection'. Basically it can identify when more than one object is in view and tell you the two distances.

Please note, the Arduino driver for this chip does not support 'small memory' boards like the ATmega328 - you'll need a SAMD21, SAMD51, ESP, etc chip with 50K of flash memory available!



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. Works great with the **3.3V logic level of a Feather**, or **the 5v level of a Arduino Mega**, this breakout is ready to work with most common microcontrollers or SBCs. and since it speaks I2C, you can easily connect it up with two data wires plus power and ground.



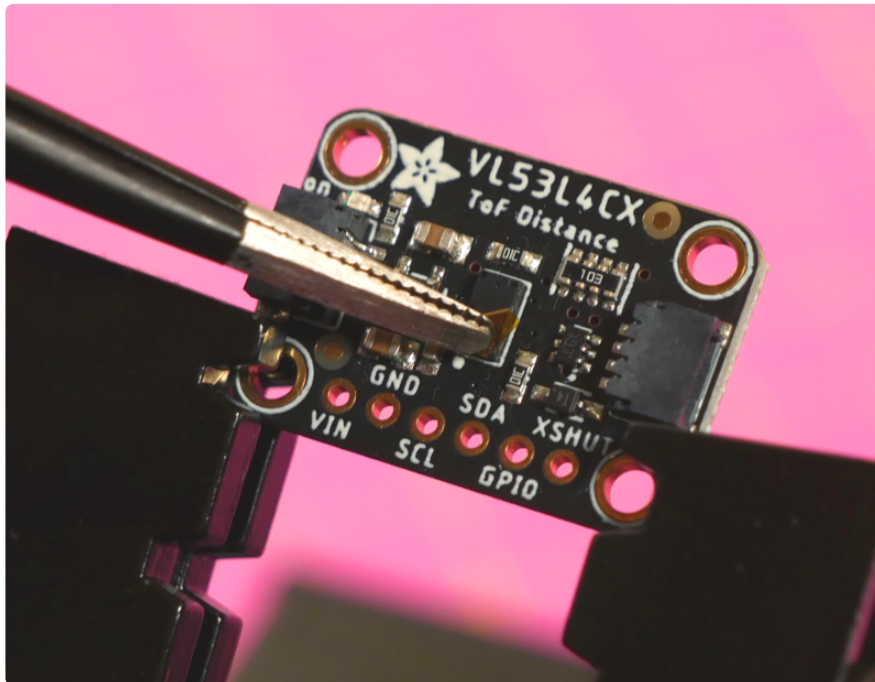
As if that weren't enough, we've also added [SparkFun qwiic \(https://adafru.it/Fpw\)](https://adafru.it/Fpw) compatible [STEMMA QT \(https://adafru.it/Ft4\)](https://adafru.it/Ft4) connectors for the I2C bus so you **don't even need to solder**. Just wire up to your favorite micro with a plug-and-play cable to get ToF data ASAP. For a no-solder experience, [just wire up to your favorite micro, like the STM32F405 Feather \(http://adafru.it/4382\)](http://adafru.it/4382) using a [STEMMA QT adapter cable. \(https://adafru.it/JnB\)](https://adafru.it/JnB) The Stemma QT connectors also mean the VL53L4CX can be used with our [various associated accessories. \(https://adafru.it/\)](https://adafru.it/) [Ft6\) QT Cable is not included, but we have a variety in the shop \(https://adafru.it/17VE\)](https://adafru.it/17VE)

Communicating to the sensor is done over I2C with an API written by ST, [they have an Arduino library with an example for communication here \(https://adafru.it/ZPd\)](https://adafru.it/ZPd).

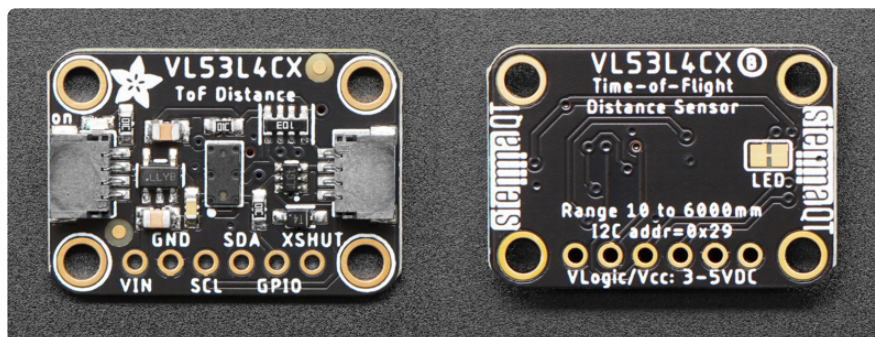
Removing the Protective Tape

Be careful when removing the tape! You don't want to damage the sensor.

Using tweezers (or some other appropriate tool), CAREFULLY remove the protective tape over the sensor, as seen in the image below. There is a small tab on the side of the tape that you can use to remove it.



Pinouts



This breakout ships with a protector over the sensor. It must be removed before use! See details at the bottom of the Overview page.

Power Pins

- **VIN** - This is the power pin. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 3V microcontroller like a Feather M4, use 3V, or for a 5V microcontroller like Arduino, use 5V.
- **GND** - This is common ground for power and logic.

I2C Logic Pins

The default I2C address for the VL53L4CX is **0x29**.

- **SCL** - I2C clock pin, connect to your microcontroller I2C clock line. There's a **10K pullup** on this pin.
- **SDA** - I2C data pin, connect to your microcontroller I2C data line. There's a **10K pullup** on this pin.
- **STEMMA QT** (<https://adafru.it/Ft4>) - These connectors allow you to connect to development boards with **STEMMA QT** connectors or to other things with [various associated accessories](https://adafru.it/JRA) (<https://adafru.it/JRA>).

Other Pins

- **GPIO** - This is the interrupt output pin, it is 2.8V logic level output - it can be read by 3.3V and most 5V logic microcontrollers.
- **XSHUT** - This is the shutdown pin. It is active low, and is logic-level shifted so you can use 3V or 5V logic.

Power LED and Jumper

- **Power LED** - This LED is a green LED located on the front of the board. It is labeled **on**.
- **LED jumper** - This jumper is located on the back of the board. Cut the trace on this jumper to cut power to the "on" LED.

Arduino

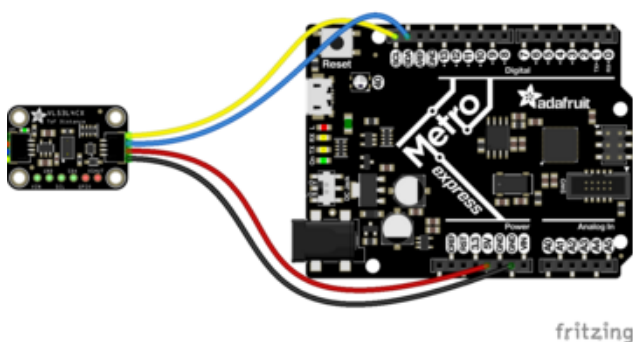
Using the VL53L4CX with Arduino involves wiring up the sensor to your Arduino-compatible microcontroller, installing the [STM32duino VL53L4CX](https://adafru.it/ZPd) (<https://adafru.it/ZPd>) library and running the provided example code.

Please note, the Arduino driver for this chip does not support 'small memory' boards like the ATmega328 - you'll need a SAMD21, SAMD51, ESP, etc chip with 50K of flash memory available!

Wiring

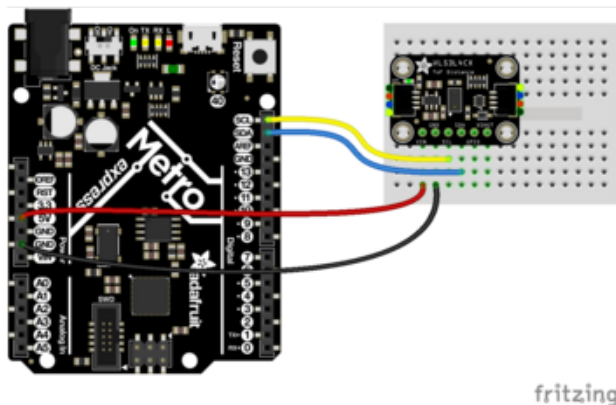
Wire as shown for a **5V** board. If you are using a **3V** board, like an Adafruit Feather, wire the board's **3V** pin to the VL53L4CX **VIN**.

Here is an Adafruit Metro Express wired up to the VL53L4CX using the STEMMA QT connector:



Board 5V to sensor VIN (red wire)
Board GND to sensor GND (black wire)
Board SCL to sensor SCL (yellow wire)
Board SDA to sensor SDA (blue wire)

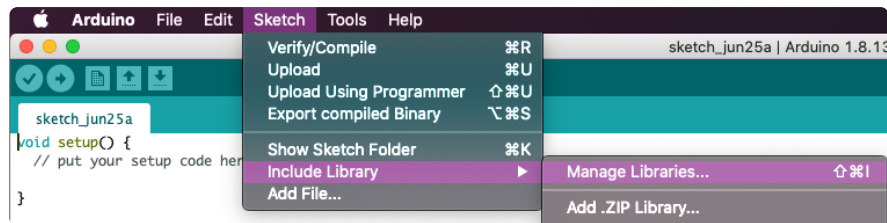
Here is an Adafruit Metro Express wired up using a solderless breadboard:



Board 5V to sensor VIN (red wire)
Board GND to sensor GND (black wire)
Board SCL to sensor SCL (yellow wire)
Board SDA to sensor SDA (blue wire)

Library Installation

You can install the **VL53L4CX** library for Arduino using the Library Manager in the Arduino IDE.



Click the **Manage Libraries ...** menu item, search for **VL53L4CX**, and select the **STM32duino VL53L4CX** library:



Load Example

Open up **File -> Examples -> STM32duino VL53L4CX -> VL53L4CX_Sat_HelloWorld** and upload to your Arduino wired to the sensor.

Note: this is only the loop in the screenshot.

```

98 void loop()
99 {
100     VL53L4CX_MultiRangingData_t MultiRangingData;
101     VL53L4CX_MultiRangingData_t *pMultiRangingData = &MultiRangingData;
102     uint8_t NewDataReady = 0;
103     int no_of_object_found = 0, j;
104     char report[64];
105     int status;
106
107     do {
108         status = sensor_vl53l4cx_sat.VL53L4CX_GetMeasurementDataReady(&NewDataReady);
109     } while (!NewDataReady);
110
111     //Led on
112     digitalWrite(LedPin, HIGH);
113
114     if ((!status) && (NewDataReady != 0)) {
115         status = sensor_vl53l4cx_sat.VL53L4CX_GetMultiRangingData(pMultiRangingData);
116         no_of_object_found = pMultiRangingData->NumberOfObjectsFound;
117         sprintf(report, sizeof(report), "VL53L4CX Satellite: Count-%d, #Objs-%ld ", pMultiRangingData->StreamCount, no_of_object_found);
118         SerialPort.print(report);
119         for (j = 0; j < no_of_object_found; j++) {
120             if (j != 0) {
121                 SerialPort.print("\r\n          ");
122             }
123             SerialPort.print("status=");
124             SerialPort.print(pMultiRangingData->RangeData[j].RangeStatus);
125             SerialPort.print(", D=");
126             SerialPort.print(pMultiRangingData->RangeData[j].RangeMilliMeter);
127             SerialPort.print("mm");
128             SerialPort.print(", Signal=");
129             SerialPort.print(((float)pMultiRangingData->RangeData[j].SignalRateRtnMegaCps / 65536.0));
130             SerialPort.print(" Mcps, Ambient=");
131             SerialPort.print(((float)pMultiRangingData->RangeData[j].AmbientRateRtnMegaCps / 65536.0));
132             SerialPort.print(" Mcps");
133         }
134     }

```

Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. You should see the values from the sensor being printed out. The values read in the example are:

- **Count** - number of readings sent to the Serial Monitor

- **Objs** - Number of objects in front of the sensor
- **status** - Range status of the sensor
- **D** - Measurement of distance of an object from the sensor in millimeters
- **Signal** - Rate of the signal from the sensor
- **Ambient** - Ambient noise around the sensor

```
COM32
Starting...
VL53L4CX Satellite: Count=0, #Objs=0
VL53L4CX Satellite: Count=0, #Objs=1 status=6, D=11mm, Signal=25.20 Mcps, Ambient=0.00 Mcps
VL53L4CX Satellite: Count=1, #Objs=1 status=0, D=14mm, Signal=23.98 Mcps, Ambient=0.00 Mcps
VL53L4CX Satellite: Count=2, #Objs=1 status=0, D=12mm, Signal=24.69 Mcps, Ambient=0.00 Mcps
VL53L4CX Satellite: Count=3, #Objs=1 status=0, D=19mm, Signal=31.92 Mcps, Ambient=0.00 Mcps
VL53L4CX Satellite: Count=4, #Objs=1 status=0, D=143mm, Signal=0.30 Mcps, Ambient=0.01 Mcps
VL53L4CX Satellite: Count=5, #Objs=0
VL53L4CX Satellite: Count=6, #Objs=1 status=12, D=1mm, Signal=0.14 Mcps, Ambient=0.07 Mcps
VL53L4CX Satellite: Count=7, #Objs=2 status=7, D=251mm, Signal=2.30 Mcps, Ambient=0.16 Mcps
status=4, D=2374mm, Signal=0.29 Mcps, Ambient=0.16 Mcps
VL53L4CX Satellite: Count=8, #Objs=1 status=0, D=86mm, Signal=91.85 Mcps, Ambient=0.12 Mcps
VL53L4CX Satellite: Count=9, #Objs=1 status=0, D=46mm, Signal=61.07 Mcps, Ambient=0.02 Mcps
VL53L4CX Satellite: Count=10, #Objs=1 status=0, D=0mm, Signal=183.34 Mcps, Ambient=0.02 Mcps
```

Arduino Docs

[Arduino Docs \(https://adafru.it/ZPd\)](https://adafru.it/ZPd)

Downloads

Files

- [VL53L4CX Datasheet \(https://adafru.it/ZZa\)](https://adafru.it/ZZa)
- [EagleCAD PCB files on GitHub \(https://adafru.it/ZZb\)](https://adafru.it/ZZb)
- [Fritzing object in the Adafruit Fritzing Library \(https://adafru.it/-1f\)](https://adafru.it/-1f)

Schematic and Fab Print

