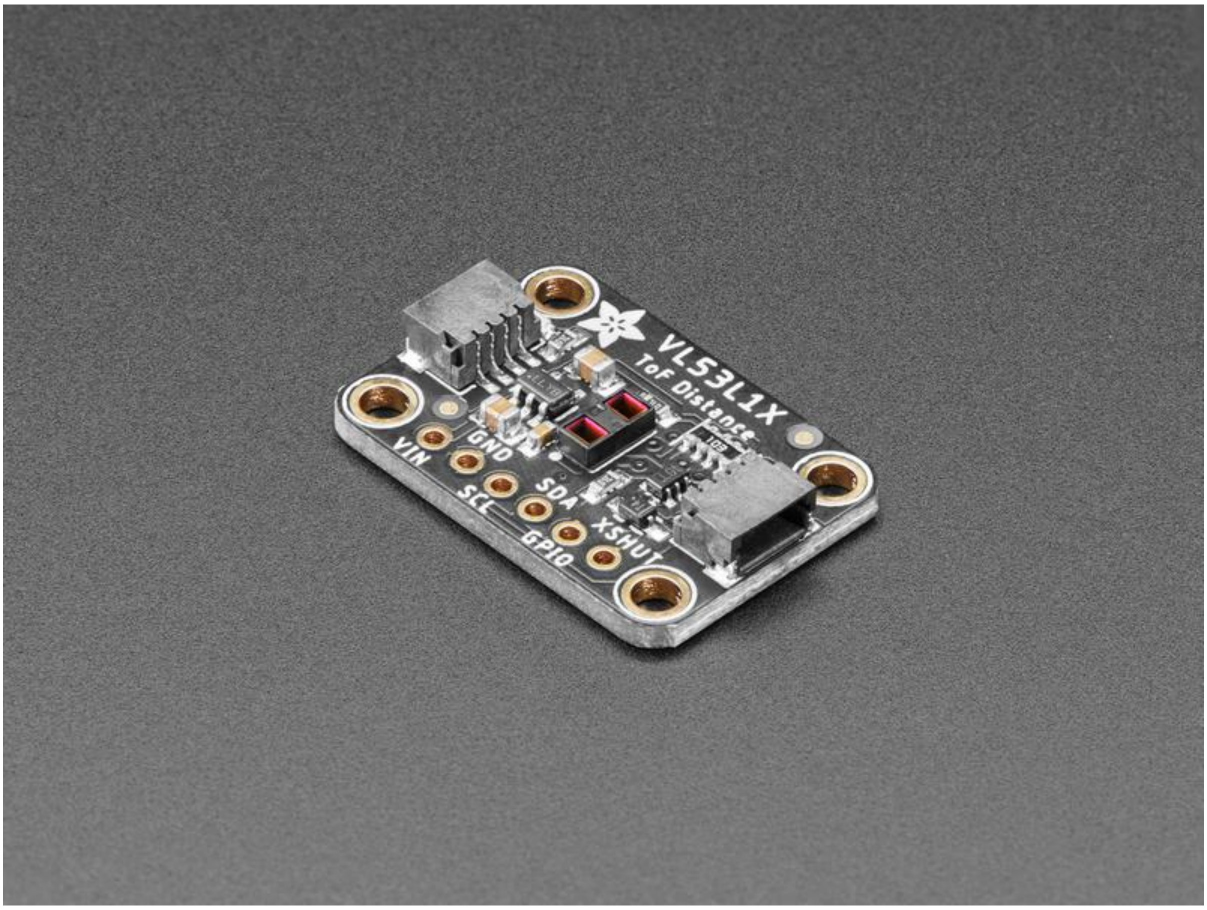
Adafruit VL53L1X Time of Flight Distance Sensor

Created by Kattni Rembor

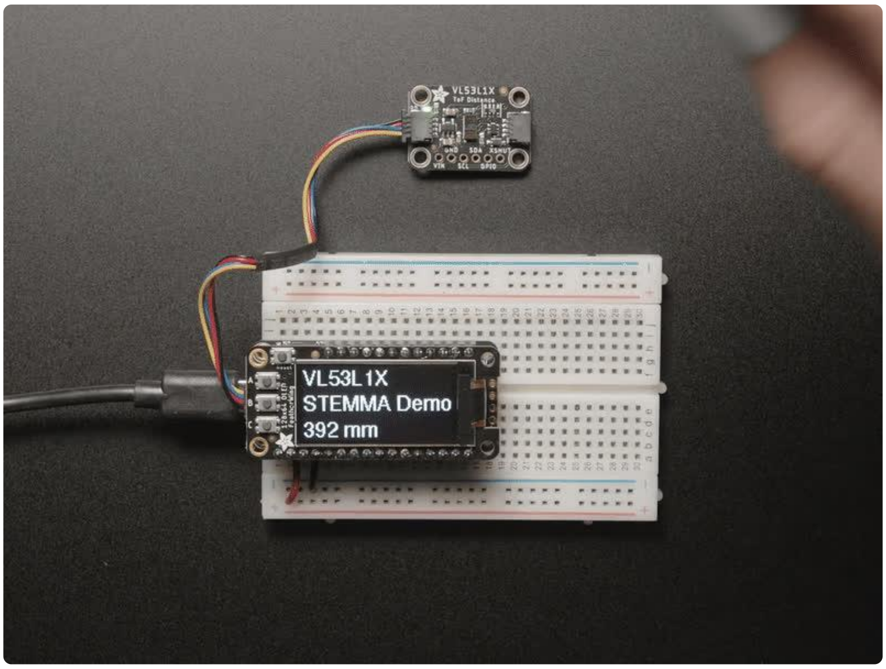


https://learn.adafruit.com/adafruit-vl53l1x

Last updated on 2021-11-15 08:29:56 PM EST

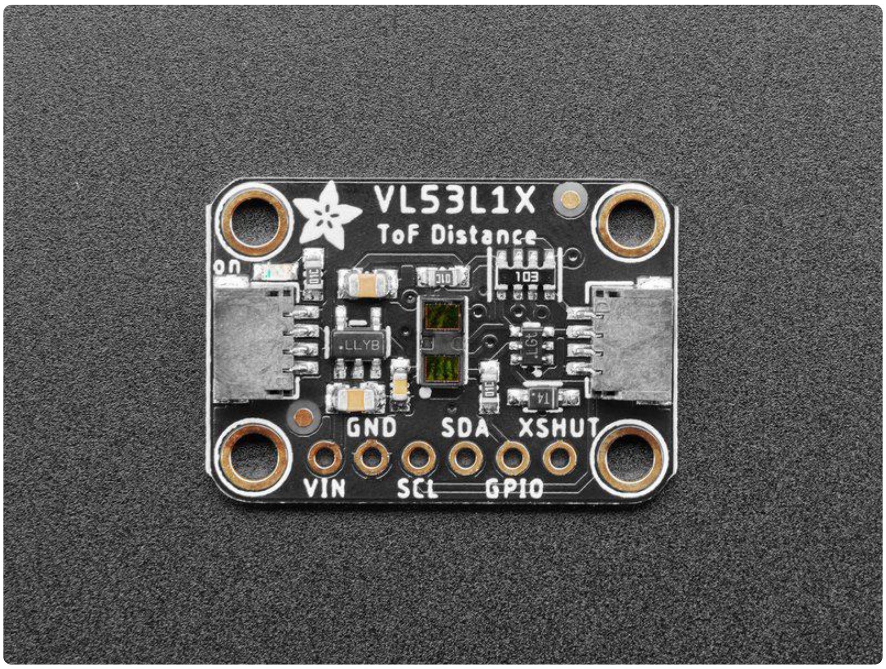
Table of Contents

Overview

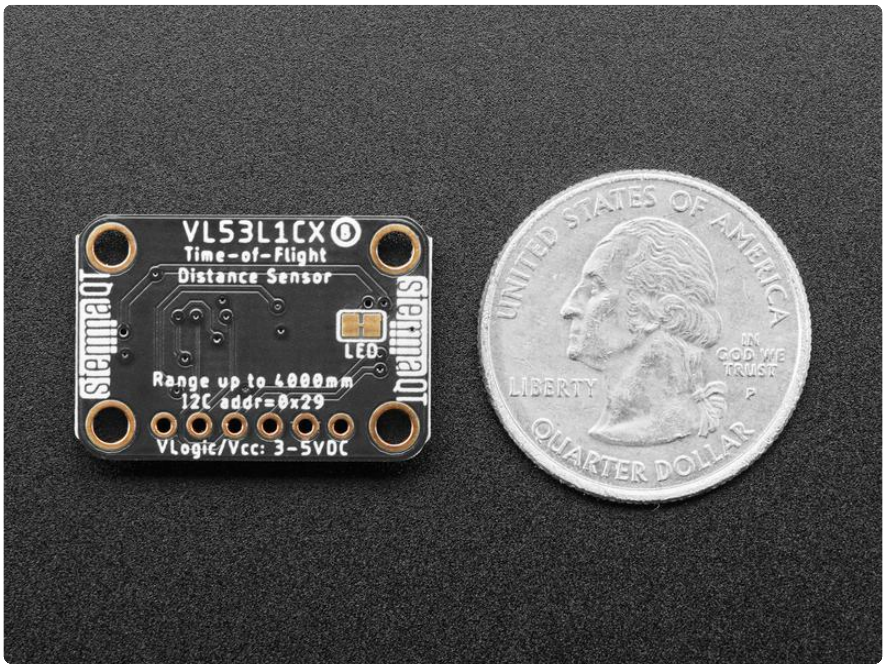


The Adafruit VL53L1X Time of Flight Distance Sensor (also known as VL53L1CX) is a Time of Flight distance sensor that has a massive 4 meter range and LIDAR-like precision. The sensor contains a very tiny invisible laser source and a matching sensor. The VL53L1X can detect the "time of flight", or how long the light has taken to bounce back to the sensor.

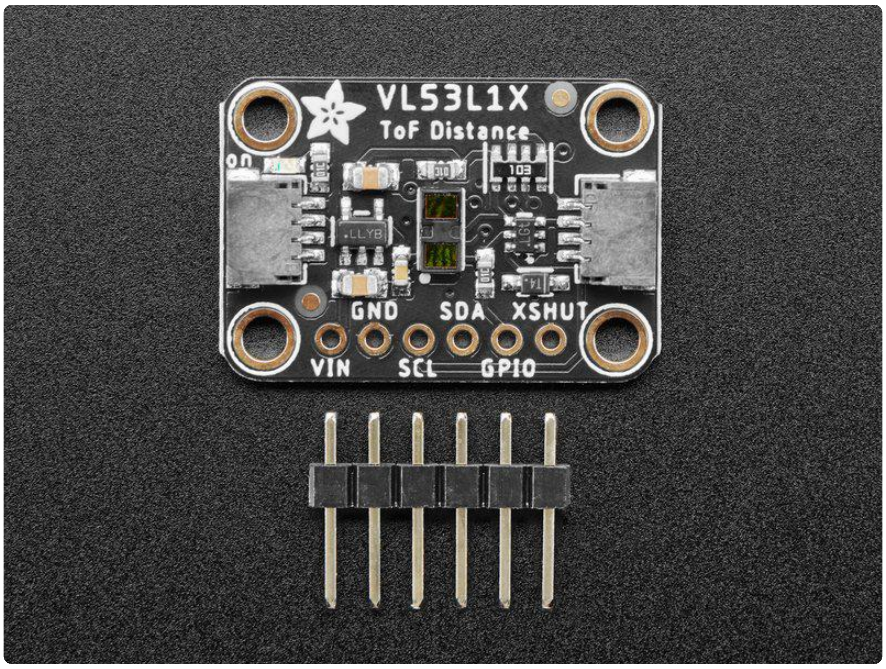
This breakout ships with a protector over the sensor. It must be removed before use! See details below.



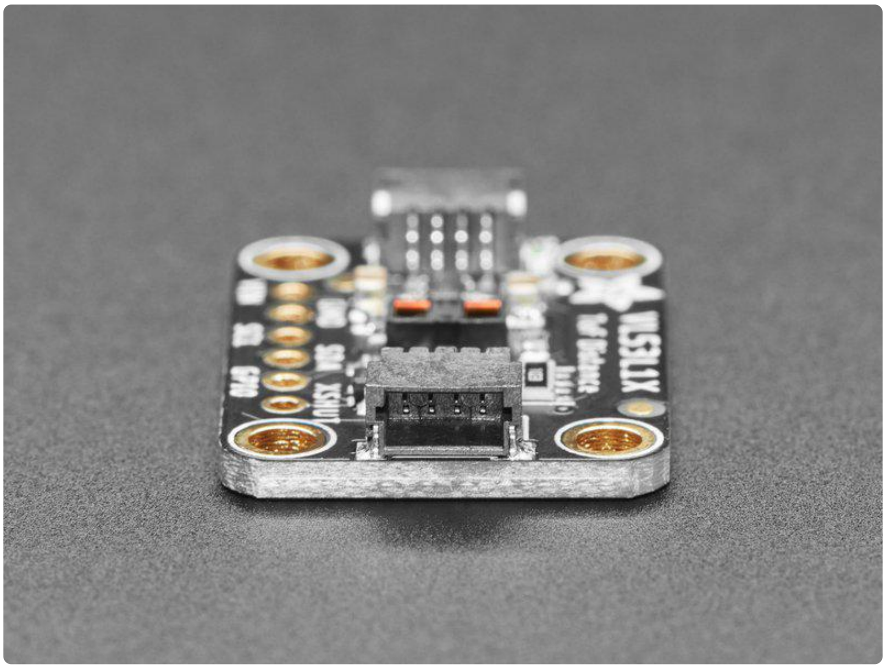
Since the sensor uses a very narrow light source, it is good for determining the distance of only the surface directly in front of it. Unlike sonar that bounces ultrasonic waves, the 'cone' of sensing is very narrow. Unlike IR distance sensors that try to measure the amount of light bounced, the VL53L1X 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 'next generation' of the VL53L0X ToF sensor (http://adafru.it/3317) and can handle about ~30 to 4000mm of range distance, with up to 50Hz update rate. If you need an even smaller/closer range, check out the VL6180X (https://adafru.it/y6B) which can measure 5mm to 200mm and also contains a light sensor.



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 Raspberry Pi, or the 5V level of a Metro 328 or Arduino Uno, 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.



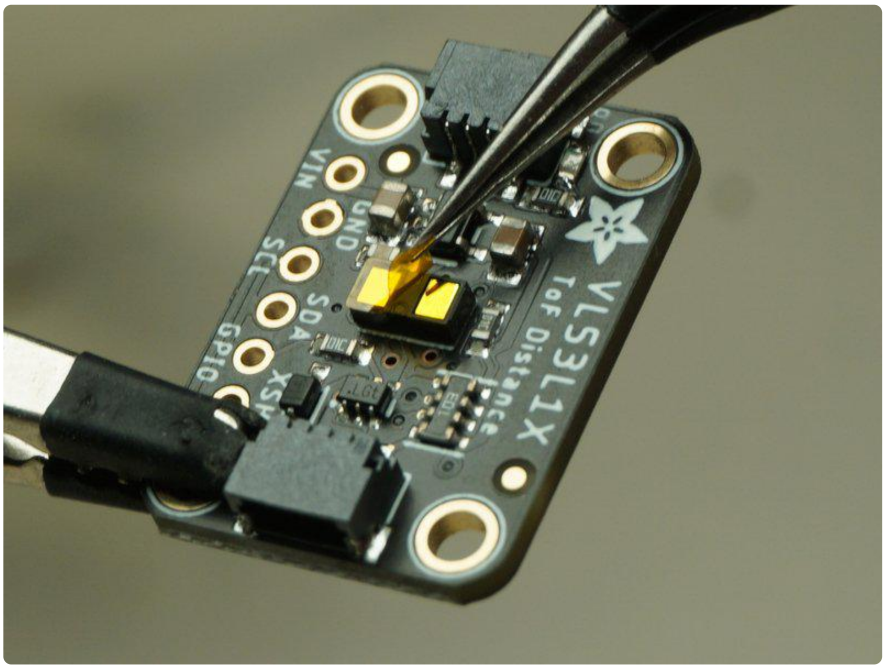
As if that weren't enough, we've also added SparkFun qwiic (https://adafru.it/Fpw) compatible STEMMA QT (https://adafru.it/Ft4) connectors for the I2C bus so you don't even need to solder. Just wire up to your favorite micro with a plug-and-play cable to get ToF data ASAP. For a no-solder experience, just wire up to your favorite micro, like the STM32F405 Feather (https://adafru.it/Iqc) using a STEMMA QT adapter cable. (https://adafru.it/JnB) The Stemma QT connectors also mean the VL53L1X can be used with our various associated accessories. (https://adafru.it/Ft6) QT Cable is not included, but we have a variety in the shop (https://adafru.it/JnB)

Communicating to the sensor is done over I2C with an API written by ST, so it's 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 (https://adafru.it/VA4).

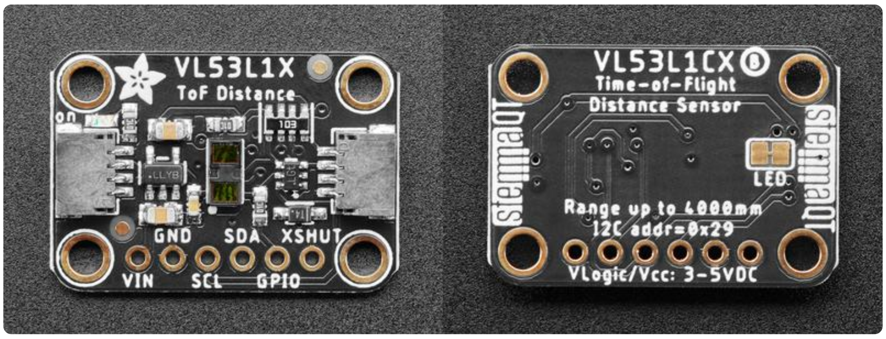
Removing the Protective Tape

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 SCD-4x 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/Ft6).

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.

Jumper

- 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.

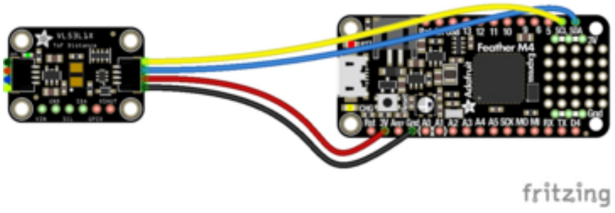
Python & CircuitPython

It's easy to use the VL53L1X with Python or CircuitPython, and the Adafruit CircuitPython VL53L1X (https://adafru.it/VA6) module. This module allows you to easily write Python code that reads the distance from the VL53L1X sensor.

You can use this sensor with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library (https://adafru.it/BSN).

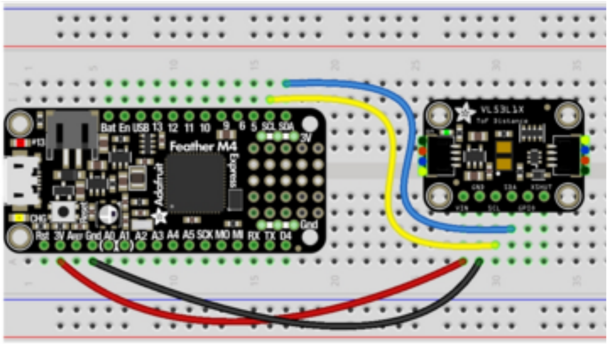
CircuitPython Microcontroller Wiring

First wire up a VL53L1X to your board exactly as shown below. Here's an example of wiring a Feather M4 to the sensor with I2C using one of the handy STEMMA QT (https://adafru.it/Ft4) connectors:



- Board 3V 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)

You can also use the standard 0.100" pitch headers to wire it up on a breadboard:

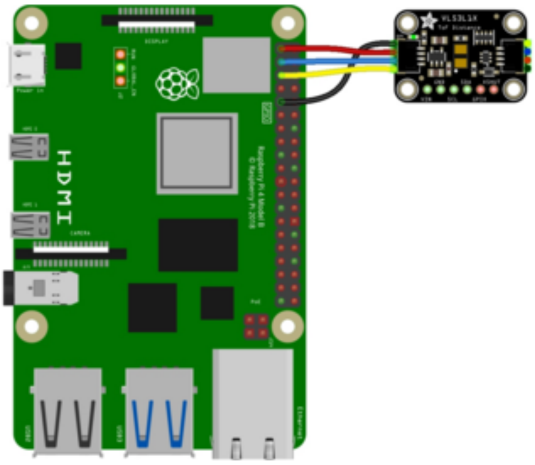


- Board 3V to sensor VIN (red wire)
- Board GND to sensor GND (black wire)
- Board SCL to sensor SCL (yellow wire)
- Board SDA to sensor SDA (blue wire)

Python Computer Wiring

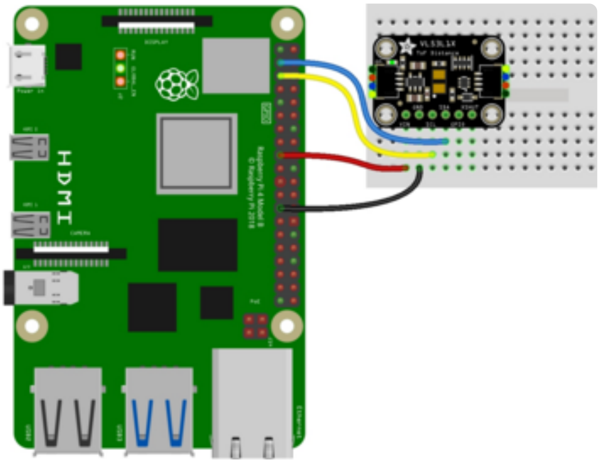
Since there's dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported (https://adafru.it/BSN).

Here's the Raspberry Pi wired to the sensor using I2C and a STEMMA QT (https://adafru.it/Ft4) connector:



- Pi 3V to sensor VIN (red wire)
- Pi GND to sensor GND (black wire)
- Pi SCL to sensor SCL (yellow wire)
- Pi SDA to sensor SDA (blue wire)

Finally here is an example of how to wire up a Raspberry Pi to the sensor using a solderless breadboard:



- Pi 3V to sensor VIN (red wire)
- Pi GND to sensor GND (black wire)
- Pi SCL to sensor SCL (yellow wire)
- Pi SDA to sensor SDA (blue wire)

Python Installation of VL53L1X Library

You'll need to install the Adafruit_Blinka library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready (https://adafru.it/BSN)!

Once that's done, from your command line run the following command:

- `pip3 install adafruit-circuitpython-vl53l1x`

If your default Python is version 3, you may need to run `pip` instead. Make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

CircuitPython Usage

To use with CircuitPython, you need to first install the VL53L1X library, and its dependencies, into the lib folder on your CIRCUITPY drive. Then you need to update code.py with the example script.

Thankfully, we can do this in one go. In the example below, click the Download Project Bundle button below to download the necessary libraries and the code.py file in a zip file. Extract the contents of the zip file, and copy the entire lib folder and the code.py file to your CIRCUITPY drive.

Your CIRCUITPY/lib folder should contain the following folder and file:

- adafruit_bus_device/
- adafruit_vl53l1x.mpy

Python Usage

Once you have the library `pip3` installed on your computer, copy or download the following example to your computer, and run the following, replacing code.py with whatever you named the file:

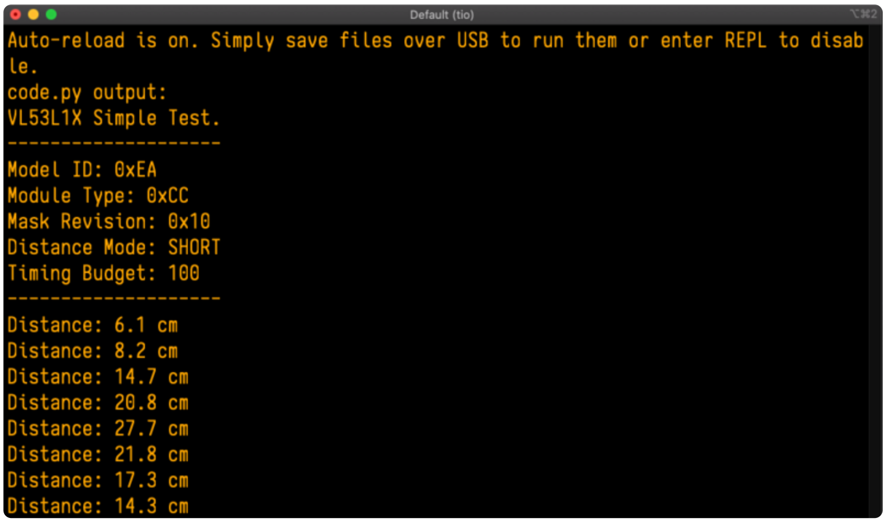
`python3 code.py`

Example Code

```
# SPDX-FileCopyrightText: 2017 Scott Shawcroft, written for Adafruit Industries
# SPDX-FileCopyrightText: Copyright (c) 2021 Carter Nelson for Adafruit Industries
#
# SPDX-License-Identifier: Unlicense

# Simple demo of the VL53L1X distance sensor.
# Will print the sensed range/distance every second.

import time
import board
import adafruit_vl53l1x

i2c = board.I2C()

vl53 = adafruit_vl53l1x.VL53L1X(i2c)

# OPTIONAL: can set non-default values
vl53.distance_mode = 1
vl53.timing_budget = 100

print("VL53L1X Simple Test.")
print("--------------------")
model_id, module_type, mask_rev = vl53.model_info
print("Model ID: 0x{:0X}".format(model_id))
print("Module Type: 0x{:0X}".format(module_type))
print("Mask Revision: 0x{:0X}".format(mask_rev))
print("Distance Mode: ", end="")
if vl53.distance_mode == 1:
    print("SHORT")
elif vl53.distance_mode == 2:
    print("LONG")
else:
    print("UNKNOWN")
print("Timing Budget: {}".format(vl53.timing_budget))
print("--------------------")

vl53.start_ranging()

while True:
    if vl53.data_ready:
        print("Distance: {} cm".format(vl53.distance))
        vl53.clear_interrupt()
        time.sleep(1.0)
```

If running CircuitPython: Once everything is saved to the CIRCUITPY drive, connect to the serial console (https://adafru.it/Bec) to see the data printed out!

If running Python: The console output will appear wherever you are running Python.



```
Auto-reload is on. Simply save files over USB to run them or enter REPL to disab
le.
code.py output:
VL53L1X Simple Test.
--------------------
Model ID: 0xEA
Module Type: 0xCC
Mask Revision: 0x10
Distance Mode: SHORT
Timing Budget: 100
--------------------
Distance: 6.1 cm
Distance: 8.2 cm
Distance: 14.7 cm
Distance: 20.8 cm
Distance: 27.7 cm
Distance: 21.8 cm
Distance: 17.3 cm
Distance: 14.3 cm
```

Now try holding your hand in front of the sensor, and moving it closer and further away to see the values change!

First you import the necessary modules and libraries. Then you instantiate the sensor on I2C.

Then you're ready to read data from the sensor, including the initial information printed to the serial console.

Finally, inside the loop, you check the distance every second.

That's all there is to using the VL53L!X with CircuitPython!

Python Docs

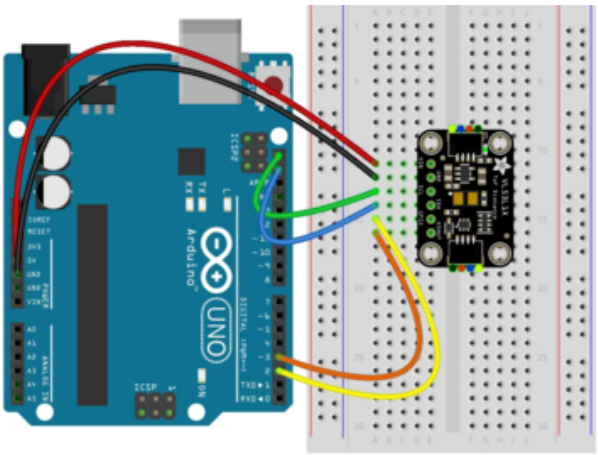
Python Docs (https://adafru.it/VA7)

Arduino

Using the VL53L1X with Arduino involves wiring up the sensor to your Arduino-compatible microcontroller, installing the Adafruit VL53L1X (https://adafru.it/VA4) library and running the provided example code.

Wiring

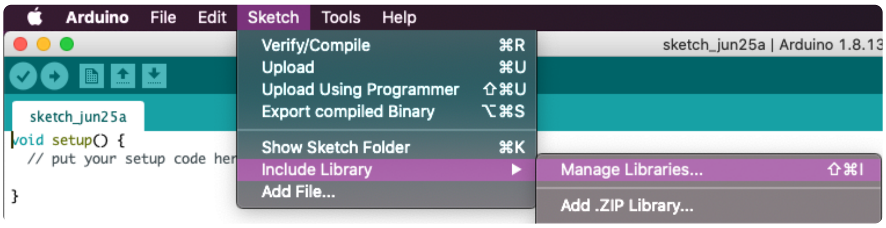
Wire as shown for a 5V board like an UNO. If you are using a 3V board, like an Adafruit Metro, wire the board's 3V pin to the VL53L1X Vin.



- Board 5V to sensor Vin
- Board GND to sensor GND
- Board SCL to sensor SCL
- Board SDA to sensor SDA
- Board 2 to sensor GPIO
- Board 3 to sensor XSHUT

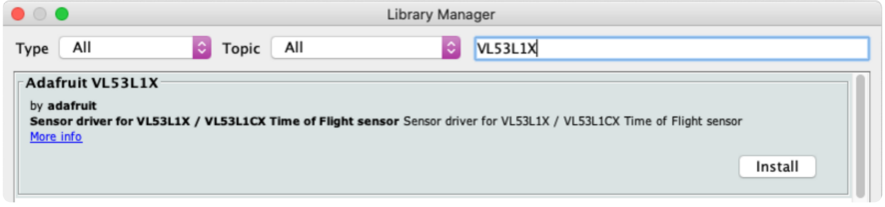
Library Installation

You can install the VL53L1X library for Arduino using the Library Manager in the Arduino IDE.



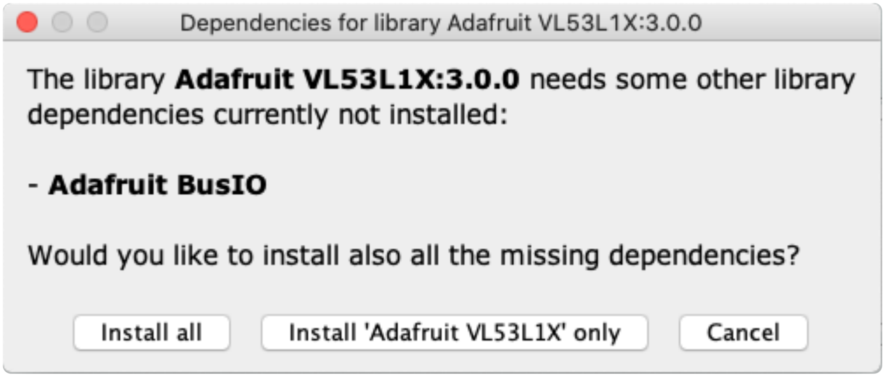
```
void setup() {
  // put your setup code her
}
```

Click the Manage Libraries ... menu item, search for VL53L1X , and select the Adafruit VL53L1X library:



Adafruit VL53L1X
by **adafruit**
Sensor driver for VL53L1X / VL53L1CX Time of Flight sensor Sensor driver for VL53L1X / VL53L1CX Time of Flight sensor
More info

If asked about dependencies, click "Install all".


Dependencies for library Adafruit VL53L1X:3.0.0
Install all
Install 'Adafruit VL53L1X' only
Cancel

The library **Adafruit VL53L1X:3.0.0** needs some other library dependencies currently not installed:

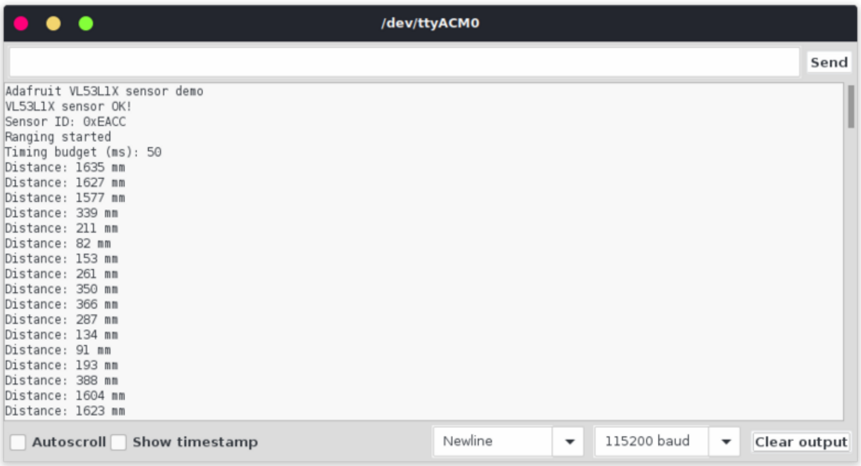
- **Adafruit BusIO**

Would you like to install also all the missing dependencies?

Load Example

Open up File -> Examples -> Adafruit VL53L1X -> VL53L1X_simpletest and upload to your Arduino wired to the sensor.

Upload the sketch to your board and open up the Serial Monitor (Tools -> Serial Monitor) at 115200 baud. You should see the the values from the senor being printed out.



```
Adafruit VL53L1X sensor demo
VL53L1X sensor OK!
Sensor ID: 0xEACC
Ranging started
Timing budget (ms): 50
Distance: 1635 mm
Distance: 1627 mm
Distance: 1577 mm
Distance: 339 mm
Distance: 211 mm
Distance: 82 mm
Distance: 153 mm
Distance: 261 mm
Distance: 350 mm
Distance: 366 mm
Distance: 287 mm
Distance: 134 mm
Distance: 91 mm
Distance: 193 mm
Distance: 388 mm
Distance: 1604 mm
Distance: 1623 mm
```

Arduino Docs

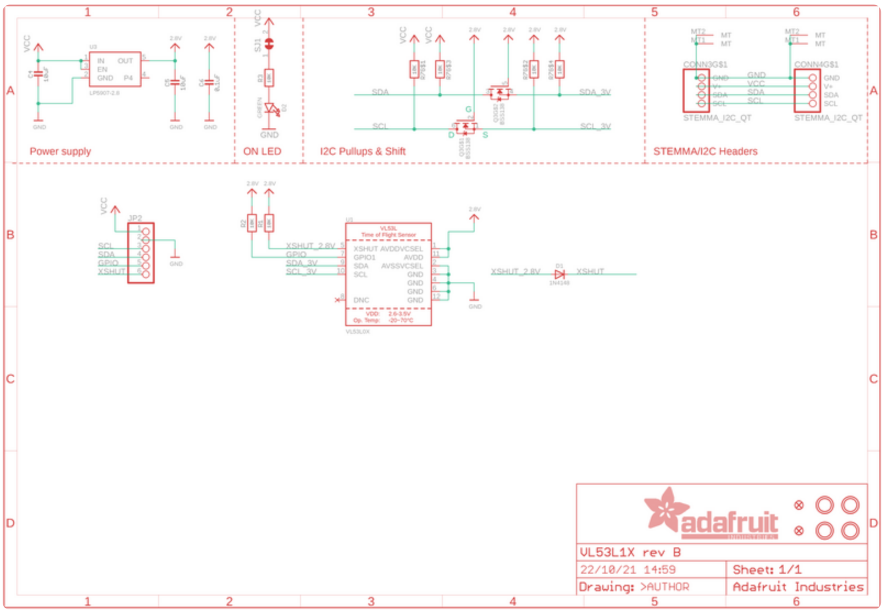
[Arduino Docs](https://adafru.it/VBt) (https://adafru.it/VBt)

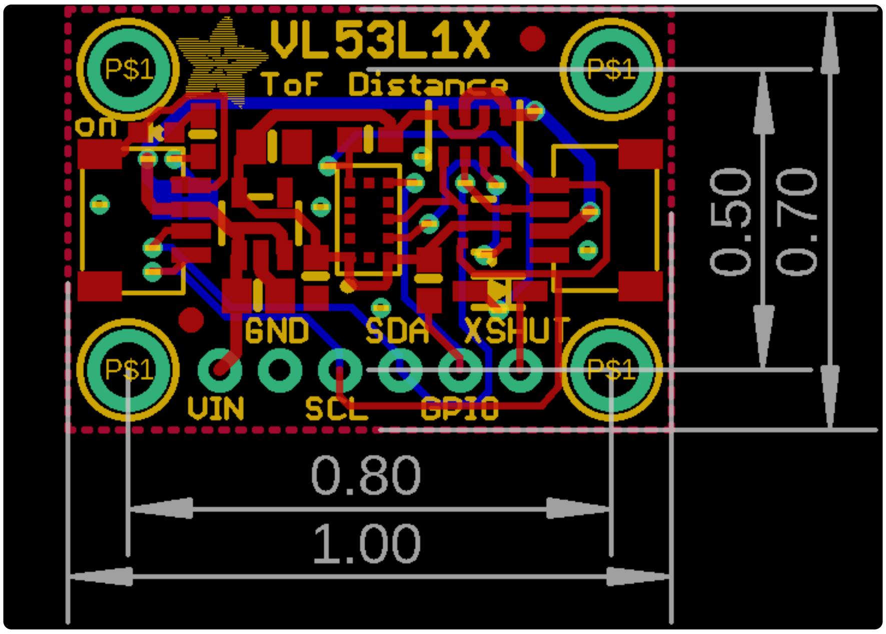
Downloads

Files

- VL53L1X datasheet (https://adafru.it/VB8)
- EaglеCAD PCB files on GitHub (https://adafru.it/VB9)
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/VBa)

Schematic and Fab Print




VL53L1X
ToF Distance
on
P$1
P$1
P$1
P$1
GND
SDA
XSHUT
VIN
SCL
GPIO
0.50
0.70
0.80
1.00