# Evelta VL53L1X ToF Distance Sensor Breakout - 4 Meter

## User Manual

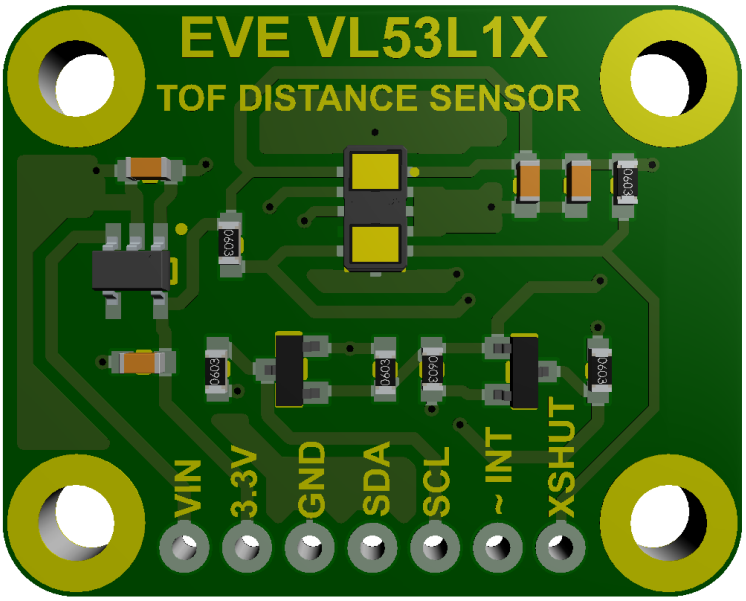
**Overview**

This Evelta Distance Sensor Breakout utilizes the VL53L1X next generation ToF (Time of Flight) sensor module. The VL53L1X is a state-of-the-art, Time-of-Flight (ToF), laser-ranging sensor, enhancing the ST FlightSense product family. It is the fastest miniature ToF sensor on the market with accurate ranging up to 4 m and fast ranging frequency up to 50 Hz VL53L1X_web.jpgHoused in a miniature and reflowable package, it integrates a SPAD receiving array, a 940 nm invisible Class1 laser emitter, physical infrared filters, and optics to achieve the best ranging performance in various ambient lighting conditions with a range of cover window options. Unlike conventional IR sensors, the VL53L1X uses ST's latest generation ToF technology which allows absolute distance measurement whatever the target color and reflectance.

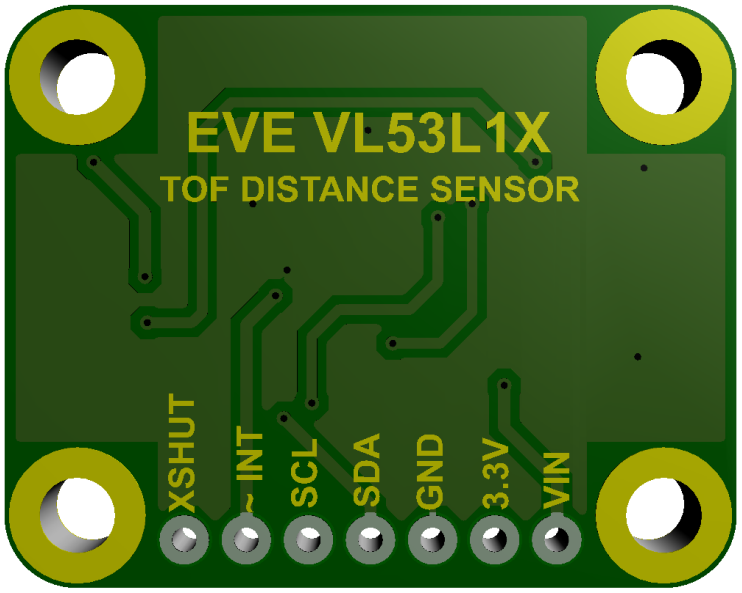
The I2C address of the VL53L1X is 0x29 and is hardware defined. A multiplexer/Mux is required to communicate to multiple VL53L1X sensors on a single bus. If you need to use more than one VL53L1X sensor.

**Board Features**

- Operating Voltage - 3/5V
- Power Consumption - 20 mW @10Hz
- Measurement Range - ~40mm to 4,000mm
- Resolution - +/-1mm
- Light Source - Class 1 940nm VCSEL
- I2C Address - 0x29
- Field of View - 15~27 degree
- Max Read Rate - 50Hz

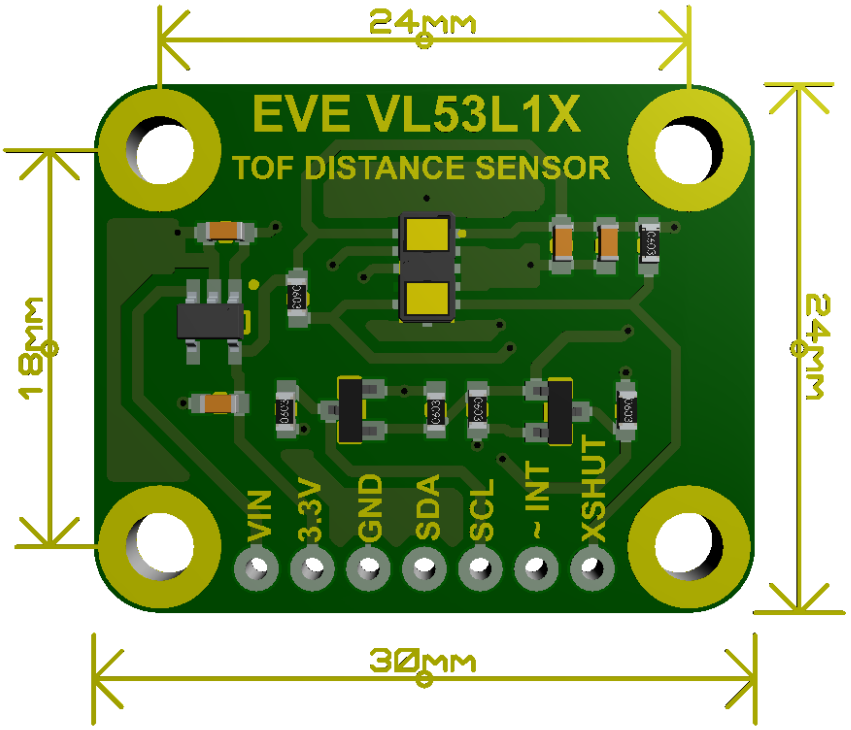



Board Front




Board Back

**Board Dimensions**



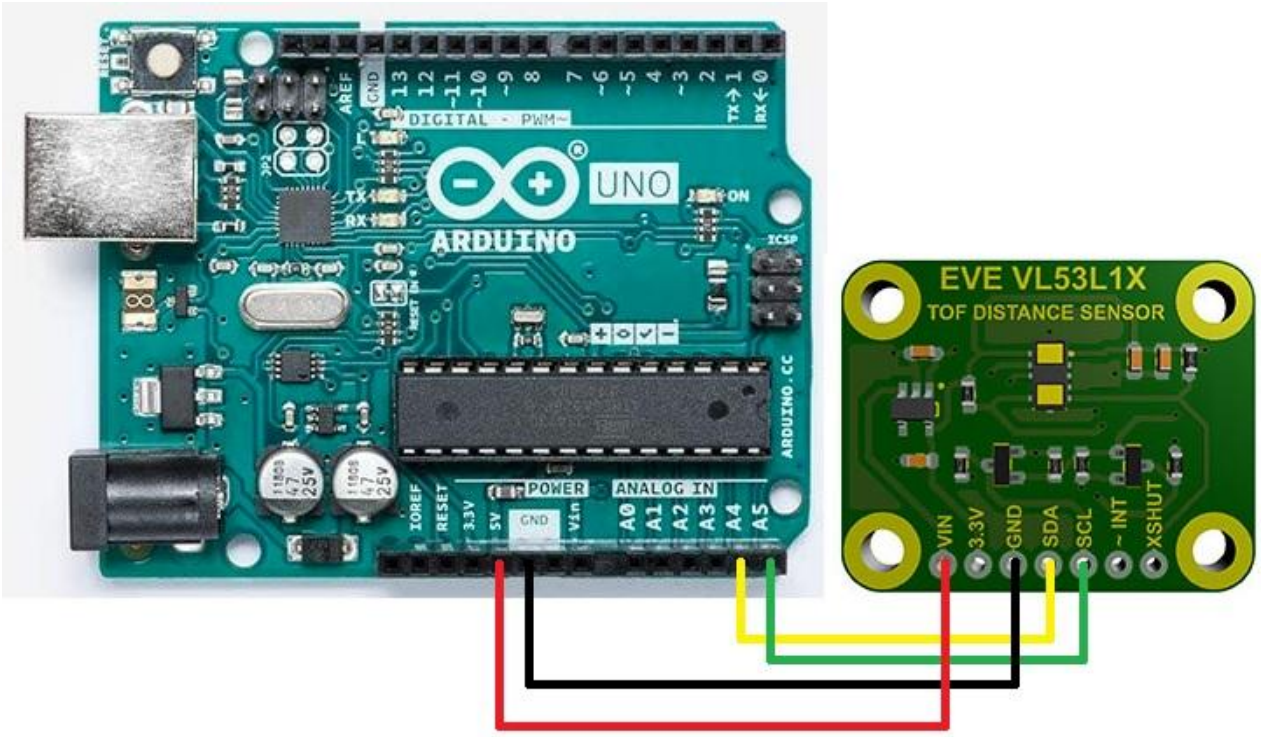

**Board Pinouts**

The following table lists all of the VL53L1X's pins and their functionality.

| Pin | Description | Direction |
|---|---|---|
| GND | Ground | In |
| 3.3V | Power | In |
| SDA | Data | In |
| SCL | Clock | In |

| | | |
|---|---|---|
| ~INT | Interrupt, goes low when data is ready. | Out |
| XSHUT | Shutdown, can be pulled low to put the IC in shutdown mode. | In |

**Evelta VL53L1X ToF Distance Sensor Arduino Connection**




**5V Arduino boards**
(including Arduino Uno, Leonardo, Mega; Pololu A-Star 32U4)

Arduino VL53L1X board
------- -------------
5V - VIN
GND - GND
SDA - SDA
SCL - SCL

**3.3V Arduino boards**
(including Arduino Due)

Arduino VL53L1X board
------- -------------
3V3 - VIN
GND - GND
SDA - SDA
SCL - SCL

To evaluate the board you'll need the SparkFun VL53L1X Arduino library, which is an easy to use wrapper of ST's driver. You can obtain these libraries through the Arduino Library Manager. Search for Sparkfun VL53L1X Arduino Library to install the latest version. If you prefer downloading the libraries from the GitHub repository and manually installing it, you can grab them here.

## Arduino Example Code

### Read Distance Code

To get started with the first example, open up File > Examples > SparkFun VL53L1x 4M Laser Distance Sensor > Example1_ReadDistance. In this example, we begin by creating a SFEVL53L1X object called distanceSensor with our wire port, Wire, and then our shutdown and interrupt pins. Then we initialize our sensor object in the *setup()* loop. The code to do this is shown below and is repeated in some form in all of the examples.

```
#include <Wire.h>
#include "SparkFun_VL53L1X.h"

//Optional interrupt and shutdown pins.
#define SHUTDOWN_PIN 2
#define INTERRUPT_PIN 3

SFEVL53L1X distanceSensor(Wire, SHUTDOWN_PIN, INTERRUPT_PIN);

void setup(void)
{
  Wire.begin();

  Serial.begin(9600);
  Serial.println("VL53L1X Qwiic Test");

  if (distanceSensor.init() == false)
    Serial.println("Sensor online!");

}
```

Once we've initialized our sensor, we can start grabbing measurements from it. To do this, we send some configuration bytes to our sensor using *distanceSensor.startRanging()* to initiate the measurement. We then wait for data to become available and when it does, we read it in, convert it from millimeters to feet, and print it out over serial. The *void loop()* function that does this is shown below.

```
void loop(void)
{
  distanceSensor.startRanging(); //Write configuration bytes to initiate
measurement
  int distance = distanceSensor.getDistance(); //Get the result of the
measurement from the sensor
  distanceSensor.stopRanging();

  Serial.print("Distance(mm): ");
  Serial.print(distance);

  float distanceInches = distance * 0.0393701;
  float distanceFeet = distanceInches / 12.0;

  Serial.print("\tDistance(ft): ");
  Serial.print(distanceFeet, 2);

  Serial.println();
}
```

**Output**

Opening your serial monitor to a baud rate of 9600 should show the distance between the sensor and the object it's pointed at in both millimeters and feet. The output should look something like the below image.