



# VL53L0X Distance Sensor User Manual

## OVERVIEW

VL53L0X Distance Sensor is a Time-of-Flight (ToF) ranging module based on the VL53L0X from ST, with accurate ranging up to 2m, it is controlled through I2C interface, and pretty low power consumption.

The VL53L0X is a ToF sensor which embeds the ST' s second generation FlightSense patented technology.

Unlike conventional ranging sensors, the VL53L0X is able to provide accurate distance measurement whatever the target color and reflectance, achieves better anti-interference capability.

## FEATURES

- I2C communication interface, control the module on/off via IO pins
- Onboard voltage translator, compatible with 3.3V/5V operating voltage
- Comes with development resources and manual (examples for Raspberry Pi/Arduino/STM32)

## SPECIFICATIONS

- Operating voltage: 3.3V/5V
- Dimension: 20mm × 24mm

- Mounting holes size: 2.0mm
- Ranging distance: 30 ~ 2000mm
- Ranging accuracy:  $\pm 5\%$  (high speed mode),  $\pm 3\%$  (high accuracy mode)
- Ranging time (min): 20ms (high speed mode), 200ms (high accuracy mode)
- Field of view: 25°
- Laser wavelength: 940nm
- Operating temperature: -20 ~ 70°C

## PINOUTS

**VCC:** 3.3V/5V power input

**GND:** ground

**SDA:** I2C data pin

**SCL:** I2C clock pin

**SHUT:** shutdown control, connects to IO pin

**INT:** interrupt output, connects to IO pin

## HARDWARE

### VL53L0X

VL53L0X Time-of-Flight (ToF) ranging sensor is ST' s second generation laser-ranging module housed in the smallest package on the market today. The VL53L0X is a fully integrated miniature module which integrates embedded infrared ranging sensor, VCSEL (Vertical Cavity Surface-Emitting Laser), couples with internal physical infrared filters and a leading-edge SPAD array (Single Photon Avalanche Diodes). The VL53L0x embeds ST' s second generation FlightSense™ patented technology, has longer ranging distance, higher rate and accuracy, opening the door to various new application

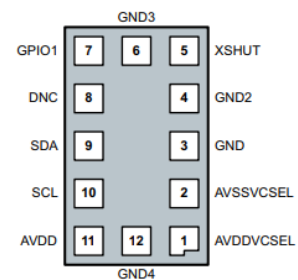
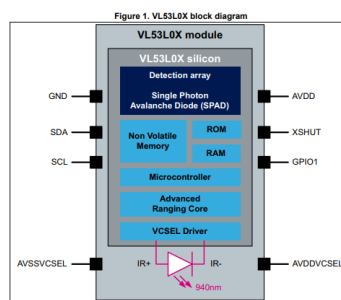
VL53L0X can measure absolute distances up to 2m t whatever the target reflectances unlike conventional technologies. It' s ultra-Low designs supports for wireless and IOT application.

#### **Features:**

- Fully integrated miniature module
  - 940 nm laser VCSEL
  - VCSEL driver
  - Ranging sensor with advanced embedded micro controller
  - 4.4 x 2.4 x 1.0 mm
- Fast, accurate distance ranging
  - Measures absolute range up to 2 m

- Reported range is independent of the target reflectance
- Advanced embedded optical cross-talk compensation to simplify cover glass selection
- Eye safe
  - Class 1 laser device compliant with latest standard IEC 60825-1:2014 - 3rd edition
- Easy integration
  - Single reflowable component
  - No additional optics
  - Single power supply
  - I2C interface for device control and data transfer
  - Xshutdown (reset) and interrupt GPIO
  - Programmable I2C address

### Pictures:



For more details, please read the datasheet

## USING WITH DEMO CODES

### WORKING WITH RASPBERRY PI

The demo code is based on open-source project on github:

[https://github.com/cassou/VL53L0X\\_rasp](https://github.com/cassou/VL53L0X_rasp)

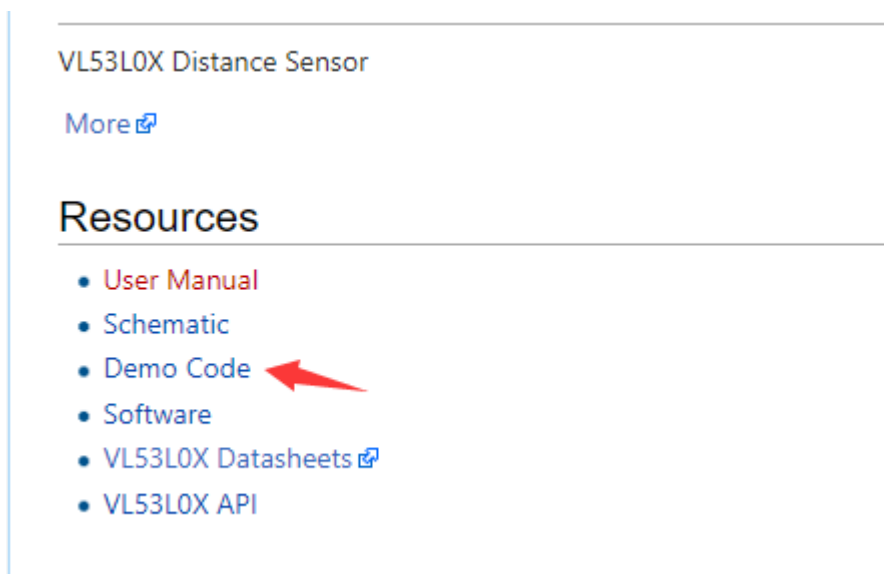
### HARDWARE CONNECTION

The connection is based on Raspberry Pi 3 Mode B+

VL53L0X Distance Sensor	Raspberry Pi
VCC	3.3V
GND	GND
SDA	SDA.1
SCL	SCL.1

### COMPILING AND RUNGING

1. Download demo code from Wiki



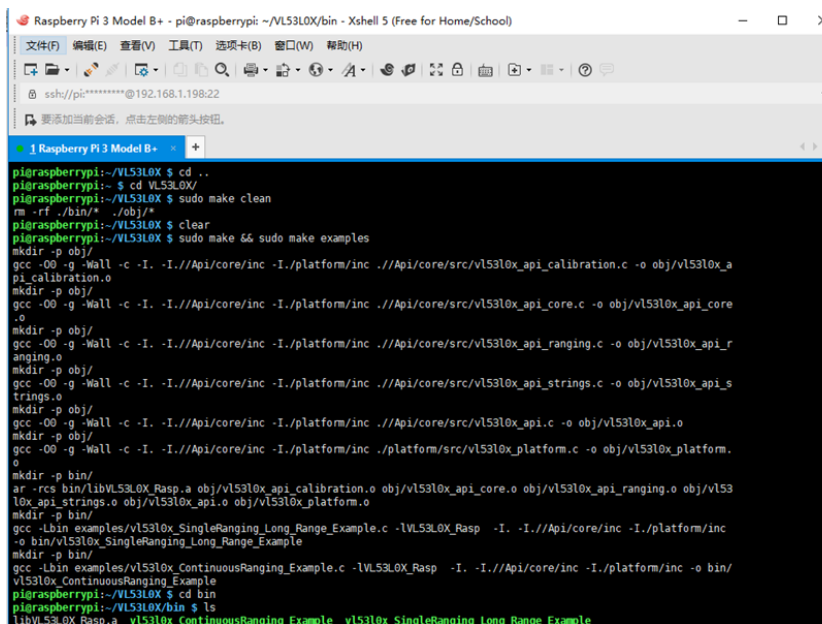
VL53L0X Distance Sensor

[More](#)

### Resources

- [User Manual](#)
- [Schematic](#)
- [Demo Code](#)
- [Software](#)
- [VL53L0X Datasheets](#)
- [VL53L0X API](#)

2. Unzip and copy the code of Raspberry Pi to your Pi (Recommend put on /home/pi/)
3. Enter the path of the demo code on Terminal and run this compiling command  
`sudo make clean && sudo make && sudo make examples`
4. After compiling, enter bin folder and list the files as below:

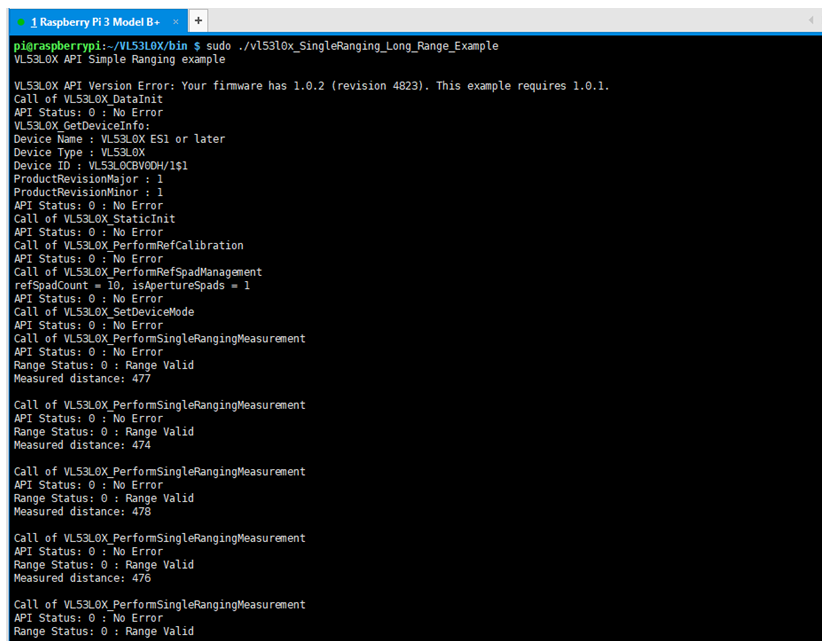


```

pi@raspberrypi:~/VL53L0X/bin - Xshell 5 (Free for Home/School)
ssh://pi*****@192.168.1.198:22
1 Raspberry Pi 3 Model B+
pi@raspberrypi:~/VL53L0X $ cd ..
pi@raspberrypi:~ $ cd VL53L0X/
pi@raspberrypi:~/VL53L0X $ sudo make clean
rm -rf ./bin/* ./obj/*
pi@raspberrypi:~/VL53L0X $ clear
pi@raspberrypi:~/VL53L0X $ sudo make && sudo make examples
mkdir -p obj/
gcc -O0 -g -Wall -c -I. -I./Api/core/inc -I./platform/inc ./Api/core/src/vl53l0x_api_calibration.c -o obj/vl53l0x_a
pi_calibration.o
mkdir -p obj/
gcc -O0 -g -Wall -c -I. -I./Api/core/inc -I./platform/inc ./Api/core/src/vl53l0x_api_core.c -o obj/vl53l0x_api_core
.o
mkdir -p obj/
gcc -O0 -g -Wall -c -I. -I./Api/core/inc -I./platform/inc ./Api/core/src/vl53l0x_api_ranging.c -o obj/vl53l0x_api_r
anging.o
mkdir -p obj/
gcc -O0 -g -Wall -c -I. -I./Api/core/inc -I./platform/inc ./Api/core/src/vl53l0x_api_strings.c -o obj/vl53l0x_api_s
trings.o
mkdir -p obj/
gcc -O0 -g -Wall -c -I. -I./Api/core/inc -I./platform/inc ./Api/core/src/vl53l0x_api.c -o obj/vl53l0x_api.o
mkdir -p obj/
gcc -O0 -g -Wall -c -I. -I./Api/core/inc -I./platform/inc ./platform/src/vl53l0x_platform.c -o obj/vl53l0x_platform.
o
mkdir -p bin/
ar -rcs bin/libVL53L0X_Rasp.a obj/vl53l0x_api_calibration.o obj/vl53l0x_api_core.o obj/vl53l0x_api_ranging.o obj/vl53
l0x_api_strings.o obj/vl53l0x_api.o obj/vl53l0x_platform.o
mkdir -p bin/
gcc -Lbin examples/vl53l0x_SingleRanging_Long_Range_Example.c -lVL53L0X_Rasp -I. -I./Api/core/inc -I./platform/inc
-o bin/vl53l0x_SingleRanging_Long_Range_Example
mkdir -p bin/
gcc -Lbin examples/vl53l0x_ContinuousRanging_Example.c -lVL53L0X_Rasp -I. -I./Api/core/inc -I./platform/inc -o bin/
vl53l0x_ContinuousRanging_Example
pi@raspberrypi:~/VL53L0X $ cd bin
pi@raspberrypi:~/VL53L0X/bin $ ls
libVL53L0X_Rasp.a vl53l0x_ContinuousRanging_Example vl53l0x_SingleRanging_Long_Range_Example

```

5. Execute command to run the code `vl53l0x_SingleRanging_Long_Range_Example`.



```

pi@raspberrypi:~/VL53L0X/bin $ sudo ./vl53l0x_SingleRanging_Long_Range_Example
VL53L0X API Simple Ranging example
VL53L0X API Version Error: Your firmware has 1.0.2 (revision 4823). This example requires 1.0.1.
Call of VL53L0X_DataInit
API Status: 0 : No Error
VL53L0X_GetDeviceInfo:
Device Name : VL53L0X ES1 or later
Device Type : VL53L0X
Device ID : VL53L0CBV00H/1$1
ProductRevisionMajor : 1
ProductRevisionMinor : 1
API Status: 0 : No Error
Call of VL53L0X_StaticInit
API Status: 0 : No Error
Call of VL53L0X_PerformRefCalibration
API Status: 0 : No Error
Call of VL53L0X_PerformRefSpadManagement
refSpadCount = 10, isApertureSpads = 1
API Status: 0 : No Error
Call of VL53L0X_SetDeviceMode
API Status: 0 : No Error
Call of VL53L0X_PerformSingleRangingMeasurement
API Status: 0 : No Error
Range Status: 0 : Range Valid
Measured distance: 477

Call of VL53L0X_PerformSingleRangingMeasurement
API Status: 0 : No Error
Range Status: 0 : Range Valid
Measured distance: 474

Call of VL53L0X_PerformSingleRangingMeasurement
API Status: 0 : No Error
Range Status: 0 : Range Valid
Measured distance: 478

Call of VL53L0X_PerformSingleRangingMeasurement
API Status: 0 : No Error
Range Status: 0 : Range Valid
Measured distance: 476

```

6. Execute the command to run code *vl53l0x\_ContinuousRanging\_Example*.

```

1 Raspberry Pi 3 Model B+
pi@raspberrypi:~/VL53L0X/bin $ sudo ./vl53l0x_ContinuousRanging_Example
VL53L0X PAL Continuous Ranging example

VL53L0X API Version Error: Your firmware has 1.0.2 (revision 4823). This example requires 1.0.1.
Call of VL53L0X_DataInit
API Status: 0 : No Error
VL53L0X_GetDeviceInfo:
Device Name : VL53L0X ES1 or later
Device Type : VL53L0X
Device ID : VL53L0CDBV0DH/1$1
ProductRevisionMajor : 1
ProductRevisionMinor : 1
Call of VL53L0X_StaticInit
API Status: 0 : No Error
Call of VL53L0X_PerformRefCalibration
API Status: 0 : No Error
Call of VL53L0X_PerformRefSpadManagement
API Status: 0 : No Error
Call of VL53L0X_SetDeviceMode
API Status: 0 : No Error
Call of VL53L0X_StartMeasurement
API Status: 0 : No Error
In loop measurement 0: 512
In loop measurement 1: 515
In loop measurement 2: 517
In loop measurement 3: 511
In loop measurement 4: 506
In loop measurement 5: 500
In loop measurement 6: 503
In loop measurement 7: 508
In loop measurement 8: 509
In loop measurement 9: 501
In loop measurement 10: 504
In loop measurement 11: 509
In loop measurement 12: 501
In loop measurement 13: 506
In loop measurement 14: 505
In loop measurement 15: 507
In loop measurement 16: 504
In loop measurement 17: 504
In loop measurement 18: 504
In loop measurement 19: 506
In loop measurement 20: 499
In loop measurement 21: 505
In loop measurement 22: 496

```

## WORKING WITH ARDUINO

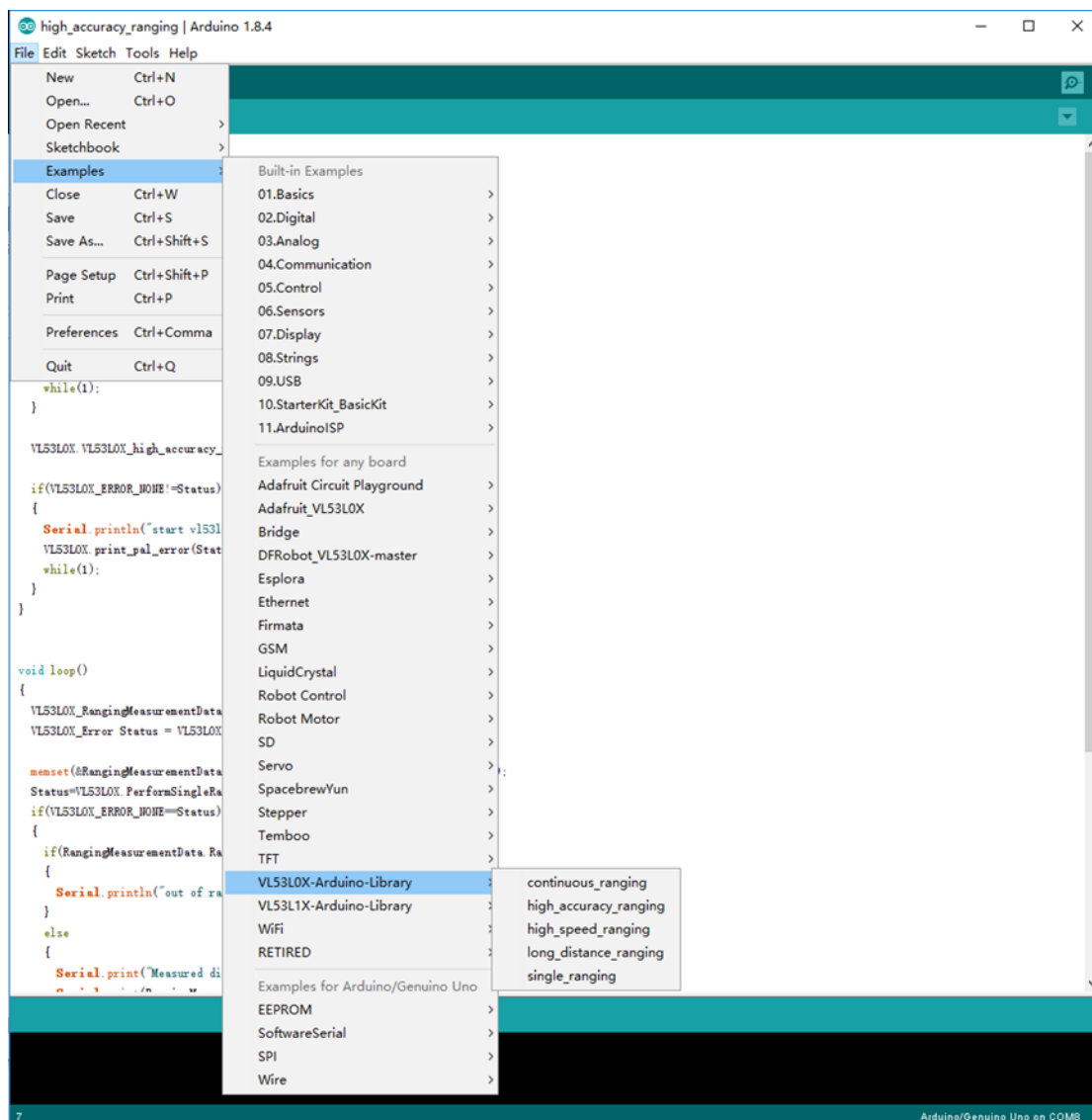
### HARDWARE CONNECTION

The connection is based on Waveshare UNO PLUS

VL53L0X Distance Sensor	Arduino
<b>VCC</b>	3.3V
<b>GND</b>	GND
<b>SDA</b>	SDA
<b>SCL</b>	SCL

## RUNING

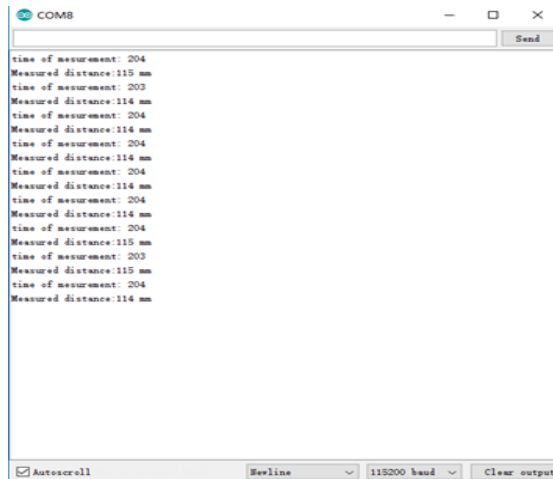
Copy the VL53L0X-Arduino-Library folder which is under Arduino demo code directory to Arduino IDE' s Libraries directory which is under the installation directory of your IDE. Then Open Arduino IDE, and choose File->Examples-> VL53L0X-Arduino-Library





Compiling and run 5 examples for test as below:

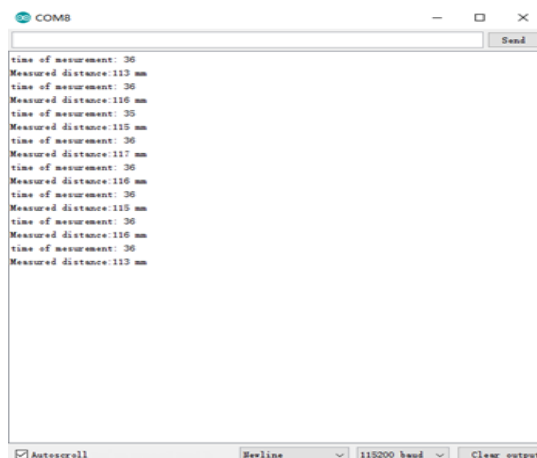
1. `high_accuracy_ranging`: High accuracy ranging which will cost more times



```
COM8
time of measurement: 204
Measured distance: 115 mm
time of measurement: 203
Measured distance: 114 mm
time of measurement: 204
Measured distance: 114 mm
time of measurement: 204
Measured distance: 114 mm
time of measurement: 204
Measured distance: 114 mm
time of measurement: 204
Measured distance: 114 mm
time of measurement: 204
Measured distance: 114 mm
time of measurement: 204
Measured distance: 115 mm
time of measurement: 203
Measured distance: 115 mm
time of measurement: 204
Measured distance: 114 mm
```

Autoscroll | Newline | 115200 baud | Clear output

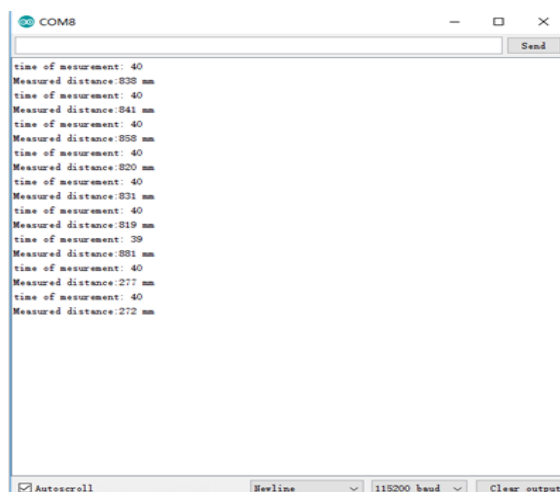
2. `high_speed_ranging`: Fast ranging, cost less time and low accuracy



```
COM8
time of measurement: 36
Measured distance: 113 mm
time of measurement: 36
Measured distance: 116 mm
time of measurement: 35
Measured distance: 115 mm
time of measurement: 36
Measured distance: 117 mm
time of measurement: 36
Measured distance: 116 mm
time of measurement: 36
Measured distance: 115 mm
time of measurement: 36
Measured distance: 116 mm
time of measurement: 36
Measured distance: 113 mm
```

Autoscroll | Newline | 115200 baud | Clear output

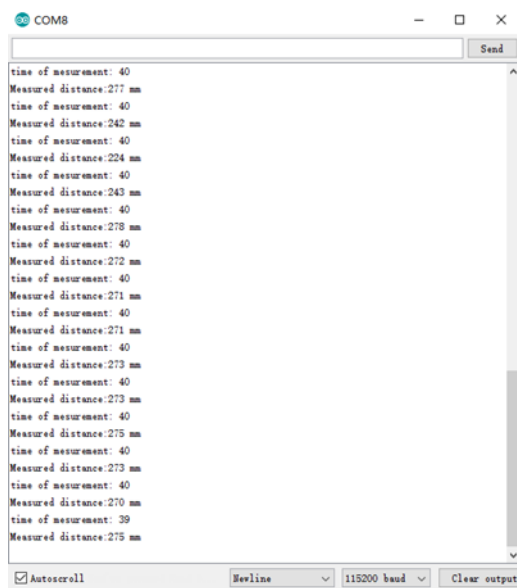
3. `long_distance_ranging`: Long distance ranging, with bigger measure rang



```
COM8
time of measurement: 40
Measured distance: 838 mm
time of measurement: 40
Measured distance: 841 mm
time of measurement: 40
Measured distance: 858 mm
time of measurement: 40
Measured distance: 820 mm
time of measurement: 40
Measured distance: 831 mm
time of measurement: 40
Measured distance: 819 mm
time of measurement: 39
Measured distance: 881 mm
time of measurement: 40
Measured distance: 277 mm
time of measurement: 40
Measured distance: 272 mm
```

Autoscroll | Newline | 115200 baud | Clear output

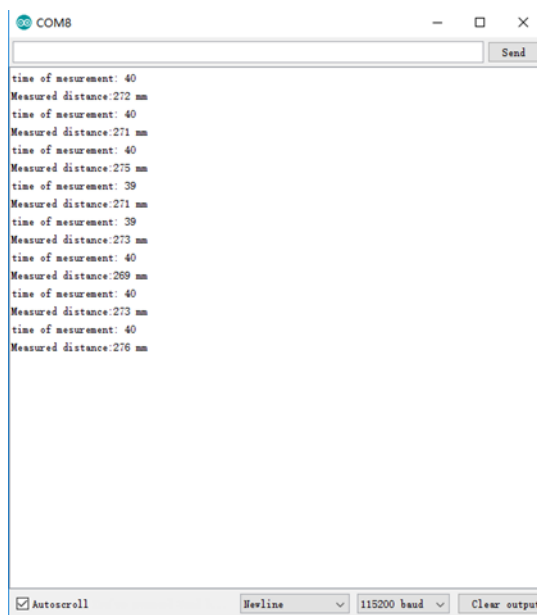
#### 4. single\_ranging: Single ranging mode



```
COM8
time of measurement: 40
Measured distance:277 mm
time of measurement: 40
Measured distance:242 mm
time of measurement: 40
Measured distance:224 mm
time of measurement: 40
Measured distance:243 mm
time of measurement: 40
Measured distance:278 mm
time of measurement: 40
Measured distance:272 mm
time of measurement: 40
Measured distance:271 mm
time of measurement: 40
Measured distance:271 mm
time of measurement: 40
Measured distance:273 mm
time of measurement: 40
Measured distance:273 mm
time of measurement: 40
Measured distance:275 mm
time of measurement: 40
Measured distance:273 mm
time of measurement: 40
Measured distance:270 mm
time of measurement: 39
Measured distance:275 mm
```

Autoscroll Newline 115200 baud Clear output

#### 5. continuous\_ranging: Continuous ranging mode



```
COM8
time of measurement: 40
Measured distance:272 mm
time of measurement: 40
Measured distance:271 mm
time of measurement: 40
Measured distance:275 mm
time of measurement: 39
Measured distance:271 mm
time of measurement: 39
Measured distance:273 mm
time of measurement: 40
Measured distance:269 mm
time of measurement: 40
Measured distance:273 mm
time of measurement: 40
Measured distance:276 mm
```

Autoscroll Newline 115200 baud Clear output

For more information about different ranging modes, please read VL53L0X API

## WORKING WITH STM32

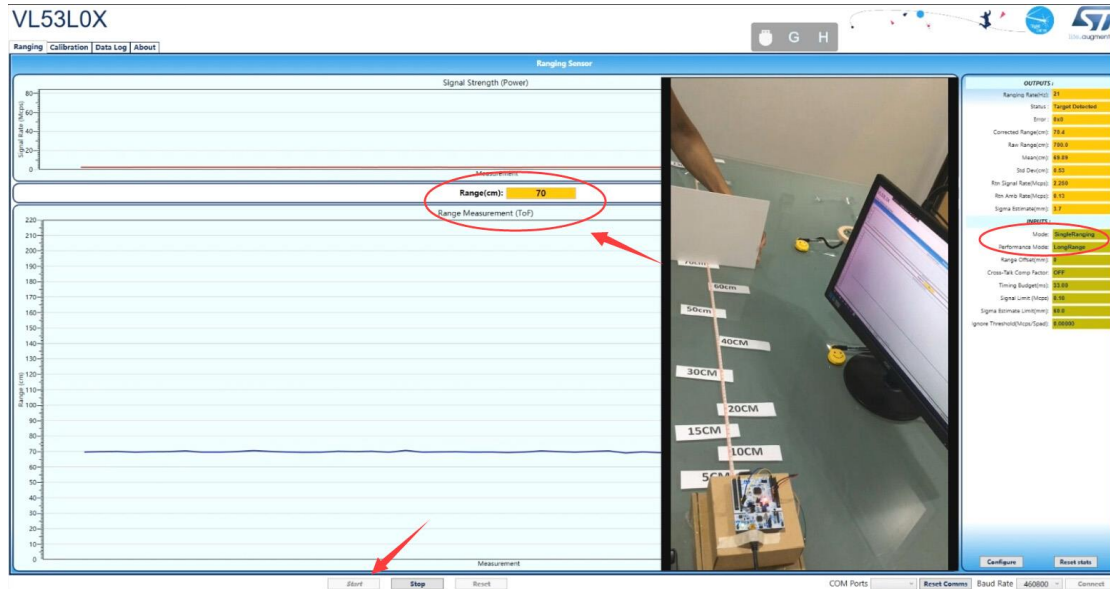
### HARDWARE CONNECTION

This connection is based on XNUCLEO-F411RE/ NUCLEO-F401RE:

VL53L0X Distance Sensor	XNUCLEO-F411RE/NUCLEO-F401RE
VCC	3.3V
GND	GND
SDA	SDA
SCL	SCL

### RUNNING VL53L0X\_GUI

1. Connecting sensor to NUCLEO-F401RE board as [Hardware connection](#) above
2. Install ST' s VL53L0X\_GUI software and run it. The software will write the test firmware to NUCLEO board automatically. (**Note: for this example, you need to use the NUCLEO-F401RE development board, and it is not an open-source code**)
3. Switch to Low-Power Automous, Distance Mode choose Short, then click Start.  
Distance curve will be showed on the software



For more information about STM32 examples VL53L0X, please visit ST website.