UM2931

User manual

A guide to using the VL53L4CD ultra lite driver (ULD)

Introduction

The VL53L4CD is a state of the art, Time-of-Flight (ToF), laser-ranging sensor enhancing the ST FlightSense product family. Housed in a miniature reflowable package, it integrates a SPAD array, physical infrared filters, and an integrated firmware to achieve the best ranging performance in various ambient lighting conditions with a range of cover glass materials.

The purpose of this user manual is to handle the VL53L4CD Time-of-Flight sensor, using the ultra lite driver (ULD) API. It describes the main functions to program the device, the calibrations, and the output results.

Figure 1. **VL53L4CD module**

References:

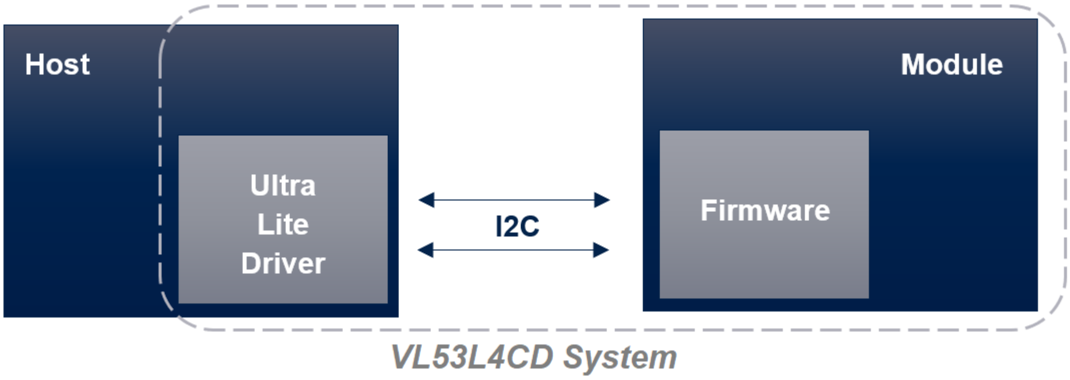
1. VL53L4CD datasheet (DS13812)

1 Functional description

1.1 System overview

The VL53L4CD system is composed of a hardware module and the ultra lite driver software (VL53L4CD ULD) running on a host (see figure below). The hardware module contains the ToF sensor. ST delivers the software driver, which is referred to in this document as "the driver". This document describes the functions of the driver that are accessible to the host. These functions control the sensor and get the ranging data.

Figure 2. VL53L4CD system overview



1.2 Schematics and I²C configuration

The communication between driver and firmware is handled by I²C, with a capability of operating up to 1 MHz (fast mode plus). The implementation requires pull-ups on the SCL and SDA lines. Refer to the VL53L4CD datasheet (DS13812) for more information.

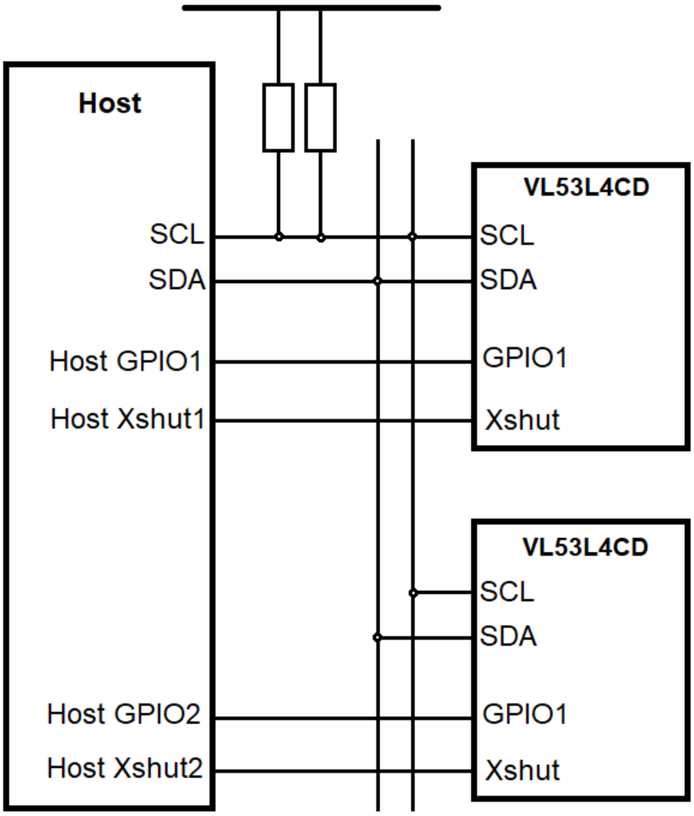
By default, the sensor is programmed to run on I²C fast mode (up to 400 kHz). The fast mode plus (up to 1 MHz) can be enabled using a platform compilation key in the file platform.h:

```
#define VL53L4CD_I2C_FAST_MODE_PLUS
```

The VL53L4CD device has a default I²C address of 0x52. However, it is possible to change the default address to avoid conflict with other devices, or facilitate adding multiple VL53L4CD modules to the system for a greater system field of view (FoV). The I²C address can be reprogrammed using the function *VL53L4CD_SetI2CAddress()*.

The following figure represents the required connection to have multiple sensors on a single I²C bus. For the schematics, refer to the VL53L4CD datasheet.

Figure 3. **Multiple sensors on I²C bus**



The host hardware design must ensure that the sensor XSHUT pins can be controlled individually. Each XSHUT pin must be connected to a host MCU GPIO pin.

To change the I²C address, the host must:

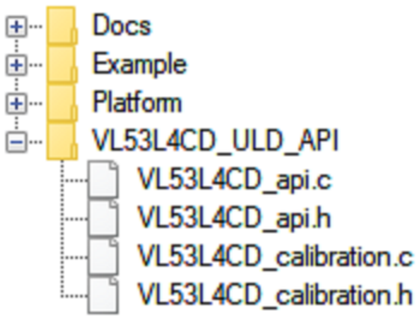
1. Put the device in hardware standby by setting the XSHUT pin low on all the VL53L4CD devices.
2. Raise the XSHUT pin of one of the VL53L4CD (for example, current_tof).
3. Call VL53LX_SetDeviceAddress(current_tof, newAddress) to program the new address.
4. Repeat the steps from [2 to 3] to change the address of all ToFs.

2 Package content and data flow

2.1 Driver architecture and content

The VL53L4CD ultra lite driver package is composed of four folders. The driver is located in the folder /VL53L4CD_ULD_API. The following figure represents the driver architecture.

Figure 4. **Driver architecture**



User also needs to implement two files located into the /Platform folder. The proposed platform is an empty dummy, and must be filled with dedicated functions.

Note: *Platform.h file contains mandatory macros to use ULD. All the file content is mandatory to use the ULD correctly.*

2.2 Calibrations

To benefit from the full performance of the sensor, the VL53L4CD driver includes two calibration functions (offset and crosstalk), which need to be run once at the production line. Run these calibration procedures to compensate the following:

- device-to-device variation (for example, when the absolute measured distance is not the same from one device to another)
- the presence of the cover glass that may affect the device ranging performances

2.2.1 Calibration order

Calibration data, stored in the host system, have to be loaded in the VL53L4CD sensor at each startup using a dedicated driver function. The calibration sequence order is important. Offset calibration should be run first followed by crosstalk calibration. The two calibrations may be run sequentially one after the other, or may be run individually. When run individually, make sure the offset data have been programmed into the sensor before running crosstalk calibration.

2.2.2 Offset calibration procedure

Offset corresponds to the difference between real distance and measured distance. A default offset is programmed into the sensor, but a calibration may be required to improve accuracy.

To perform the calibration, ST recommends using a 17% reflective target at 100 mm from the device in dark conditions, and call function *VL53L4CD_CalibrateOffset ()*. The function finds the offset, applies the offset, and returns the offset correction value. It takes a few seconds to be performed. Offset calibration must be performed before crosstalk calibration. The following table shows the available settings for running offset calibration.

Table 1. **Available settings for running offset calibration**

Setting	Min	Proposed by ST	Max
Distance [mm]	10	100	255
Number of samples	5	20	255

Note: Increasing the number of samples increases the accuracy, but also increases the time for calibration. The time relative to number of samples is linear, and values follow the approximate timeout:

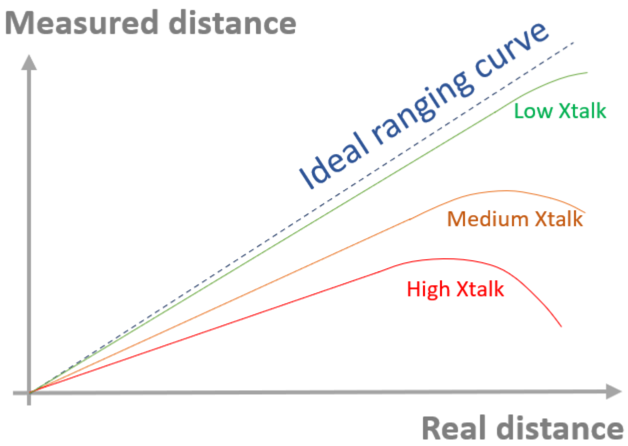
- *5 samples ≈ 750 milliseconds*
- *10 samples ≈ 1000 milliseconds*
- *20 samples ≈ 1500 milliseconds*

Users can also call functions *VL53L4CD_GetOffset()* and *VL53L4CD_SetOffset()* to get the programmed value, or set a new one.

2.2.3 Crosstalk calibration procedure

For aesthetic and protective purposes, a protective window can be added on top of the module. Due to this cover glass, a part of the light is reflected and returned to the receiver, causing a crosstalk (Xtalk) phenomenon. Depending on the cover glass quality, the amount of return signal may be significant and may affect sensor performances, as shown in the following figure.

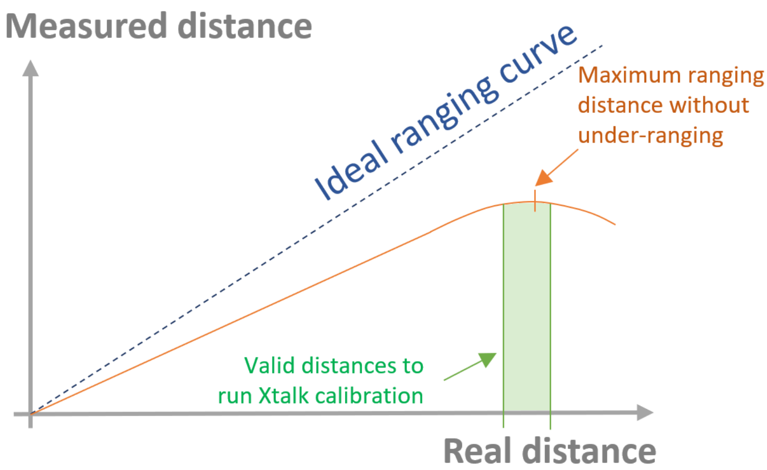
Figure 5. Impact on ranging with several crosstalk



The VL53L4CD has a built-in correction that allows the crosstalk compensation. The calibration needs to be performed on a 17% reflective target. Offset calibration needs to be performed before crosstalk calibration.

The calibration distance depends on the cover glass quality. The distance should correspond to the maximum ranging distance without under-ranging. This point can be found doing a full sweep, with a target from near to far, getting the ranging distance vs target distance.

Figure 6. Valid area for running crosstalk calibration



The calibration is performed calling function *VL53L4CD_CalibrateXtalk()*. The function finds the crosstalk, applies the crosstalk, and returns the crosstalk correction value. It takes a few seconds to be performed. The following table shows the available settings for running crosstalk calibration.

Table 2. Available settings for running crosstalk calibration

Setting	Min	Proposed by ST	Max
Distance [mm]	Depends on customer cover glass	Depends on customer cover glass	Depends on customer cover glass
Number of samples	5	20	255

Note: *Increasing the number of samples increases the accuracy, but also increases the time for calibration. The time relative to the number of samples is linear, and the values follow the approximate timeout:*

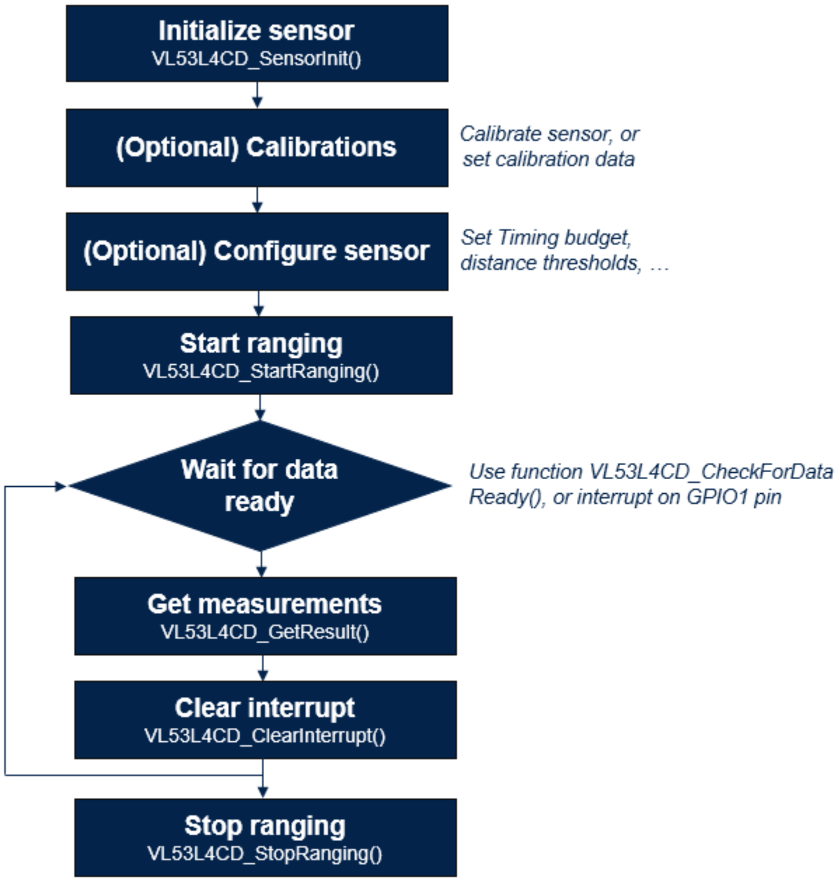
- *5 samples ≈ 750 milliseconds*
- *10 samples ≈ 1000 milliseconds*
- *20 samples ≈ 1500 milliseconds*

Users can also call functions *VL53L4CD_GetXtalk()* and *VL53L4CD_SetXtalk()* to get the programmed value, or set a new one.

2.3 Ranging flow

The following figure shows the typical ranging flow to use the VL53L4CD sensor.

Figure 7. Ranging flow



2.4 Example of configurations

The VL53L4CD sensor can be configured to achieve many usecases. There are several examples of configurations. Best performances can be reached using a high timing budget. The average power consumption during ranging can be reduced by changing the InterMeasurement.

Table 3. Example of VL53L4CD configurations

Usecase	Timing budget (ms)	InterMeasurement (ms)
Fast ranging	10	0
High accuracy	200	0
Low power	50	1000

3 Available features

The VL53L4CD ULD API contains several functions in order to tune the sensor, depending of the usecase. All functions available in the driver are described in the sections below.

3.1 Initialization

The first step to using the VL53L4CD sensor is initialization. To do so, power on the sensor, then call the function *VL53L4CD_SensorInit()*. It performs a boot and prepares the sensor.

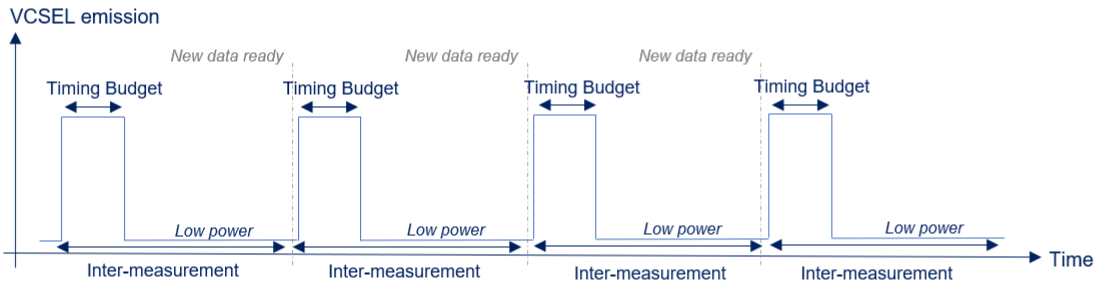
3.2 Range timing

The range timing is a single function which allows the user to define the VCSEL timeout and the ranging frequency of the sensor. It is composed of two elements:

- The timing budget: It corresponds to the VCSEL enabled time. The user can choose a value between 10 ms and 200 ms. If the InterMeasurement is set to 0, the VCSEL is always enabled, so the TimingBudget is equal to the ranging period between consecutive measurements.
- The InterMeasurement: It allows the user to define the time between two consecutive measurements. To use the InterMeasurement, the user needs to set a value greater than the TimingBudget. When the TimingBudget is consumed, the device goes into low power mode until the InterMeasurement is reached. A value set to 0 disables the InterMeasurement.

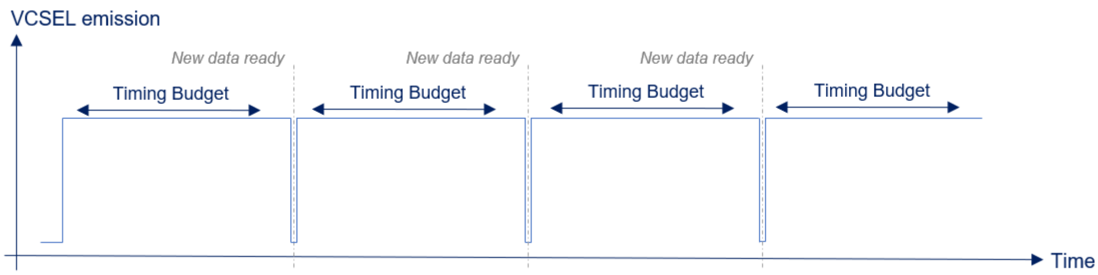
The following figure shows the difference between TimingBudget and InterMeasurement, if the InterMeasurement is greater than TimingBudget.

Figure 8. TimingBudget and InterMeasurement (not equal to 0)



If the InterMeasurement is set to 0, the TimingBudget defines the ranging measurement period, as shown in the following figure.

Figure 9. TimingBudget and InterMeasurement (equal to 0)



Increasing the timing budget increases the number of photons emitted for a single measurement. This improves the measurement accuracy and maximum ranging distance.

The InterMeasurement can be used to reduce the power consumption. The VL53L4CD power consumption depends of the duty cycle Timing Budget/InterMeasurement. A high InterMeasurement ensures a low power consumption.

The minimum and maximum value of each timing is defined in the following table.

Table 4. Minimum and maximum range timing

Timing	Min (ms)	Default (ms)	Max (ms)
TimingBudget	10	50	200
InterMeasurement	TimingBudget (or 0 to disable the feature)	0	5000

3.3 Sigma and signal thresholds

Two thresholds can be used to improve accuracy or maximum ranging distance.

The sigma corresponds to the standard deviation of the returned pulse. If the computed sigma is above the programmed value, the result status in structure *VL53L4CD_Result_t* is equal to 1 instead of 0. The value can be changed using the function *VL53L4CD_SetSigmaThreshold()*. The default value is 15 mm, minimum is 0 mm (no threshold), and maximum is 16383 mm.

The signal corresponds to the minimal number of photons to consider a valid distance. If the computed signal is below the programmed value, the result status in structure *VL53L4CD_Result_t* is equal to 2 instead of 0. The value can be changed using the function *VL53L4CD_SetSignalThreshold()*. The default value is 1024 kcps, minimum is 0 kcps (no threshold), and maximum is 65535 kcps.

The best accuracy can be reached using a low sigma and a high signal. But it also reduces the maximum ranging distance. Benefits of the two thresholds are shown in the following table.

Table 5. Sigma and signal thresholds change effect

Limit	Action	Effect on standard deviation	Effect on maximum ranging distance
Sigma threshold	Increase limit	-	+
	Decrease limit	+	-
Signal threshold	Increase limit	+	-
	Decrease limit	-	+

3.4 Detection thresholds and detection windows

In addition to the regular ranging capability, the sensor can be programmed to detect an object under certain predefined criteria. If programmed conditions are met, an interrupt is raised on PIN GPIO1. This mechanism triggers an interrupt only in particular conditions. The available detection condition usecases include:

- Below a distance threshold
- Above a distance threshold
- Within two distance thresholds
- Outside of two distance thresholds

The thresholds can be changed using the function *VL53L4CD_SetDetectionThresholds()*. By default there are no thresholds programmed.

Note: The detection threshold needs to be used with InterMeasurement enabled (not equal to 0).

3.5 Temperature update

Ambient temperature affects ranging accuracy. In order to ensure the best performances, a temperature update needs to be applied to the sensor. This update needs to be performed when the temperature might have changed by more than 8 degrees Celsius.

A dedicated function named *VL53L4CD_Start_Temperature_Update()* has to be used. The function can only be used if the sensor is not ranging. Otherwise, the ranging needs to be stopped using the function *VL53L4CD_StopRanging()*. After calling this function, the ranging can restart normally.

4 Ranging results

The VL53L4CD ULD API contains several functions in order to tune the sensor, depending on the usecase. All functions available in the driver are described in the following sections.

4.1 Getting ranging results

During the ranging session, there are two ways to know if new ranging data are available:

- Polling mode: Continuous use of function *VL53L4CD_CheckForDataReady()*. The argument indicates if new data is ready.
- Interrupt mode: Wait for a physical interrupt on PIN GPIO1.

4.2 Available data

An extensive list of data is output from the VL53L4CD. The following table explains the parameters available to the user.

Table 6. Description of available output using VL53L4CD sensor

Element	Format	Unit	Description
Status	unsigned 8 bits	None	Measurements validity. See Table 7. List of available status for more information.
Distance	unsigned 16 bits	Millimeter	Measured distance of the target in millimeters.
Ambient rate	unsigned 16 bits	Kcps	Ambient rate measurement performed on the return array, with no active photon emission, to measure the ambient signal rate due to noise.
Signal	unsigned 16 bits	Kcps	Quantity of photons measured during the VCSEL pulse.
Number of SPADs enabled	unsigned 16 bits	None	Number of SPADs enabled for the current measurement. A far or low reflective target will activate more SPADs.
Sigma	unsigned 16 bits	Millimeter	Sigma estimator for the noise in the reported target distance.

4.3 Results interpretation

The data returned by the VL53L4CD can be filtered in order to take into account the measurement status. A new status is computed for each new measurement. If the sensor is not able to measure a correct distance due to environment conditions, an invalid status is reported. The following table gives the list of available statuses.

Table 7. List of available status

Status	Gravity	Description
0	None	Returned distance is valid
1	Warning	Sigma is above the defined threshold (see Section 3.3 Sigma and signal thresholds)
2	Warning	Signal is below the defined threshold (see Section 3.3 Sigma and signal thresholds)
3	Error	Measured distance is below detection threshold
4	Error	Phase out of valid limit
5	Error	Hardware fail
6	Warning	Phase valid but no wrap around check performed
7	Error	Wrapped target, phase does not match
8	Error	Processing fail
9	Error	Crosstalk signal fail
10	Error	Interrupt error
11	Error	Merged target
12	Error	Signal is too low
255	Error	Other error (for example, boot error)

Revision history

Table 8. **Document revision history**

Date	Version	Changes
16-Sep-2021	1	Initial release
16-Aug-2022	2	Corrected the values for result status in structure *VL53L4CD_Result_t* in Section 3.3 Sigma and signal thresholds
23-Nov-2022	3	Table 1. Available settings for running offset calibration: updated minimum distance from 50 mm to 10 mm.

Contents