

## A guide for using the artificial intelligence enabler VL53L7CH and VL53L8CH multizone Time-of-Flight (ToF) sensors

### Introduction

The purpose of this user manual is to explain how to handle the VL53L7CH and VL53L8CH Time-of-Flight (ToF) sensors, using the ultra light driver (ULD) API. It describes the main functions to program the device, the calibrations, and the output results.

The CNH data transform STMicroelectronics Time-of-Flight ranging sensor into a versatile optical sensor, which can enable endless AI-based applications. This CNH raw data sent to the host, on top of the standard ranging data, opens the door to many new applications beyond simple distance measurements. From solid material (carpet, wood, glass, mirror...) to gas or liquid (water, oil, chemical...), it becomes possible to detect the location and the size of a cup in a coffee machine or beverage dispenser, to sense the floor material for robotics, and develop advanced shape, motion, or hand posture recognition.

**Figure 1. VL53L7CH and VL53L8CH sensor modules**



### References

VL53L7CH datasheet (DS14309)

VL53L8CH datasheet (DS14310)

# 1 Acronyms and abbreviations

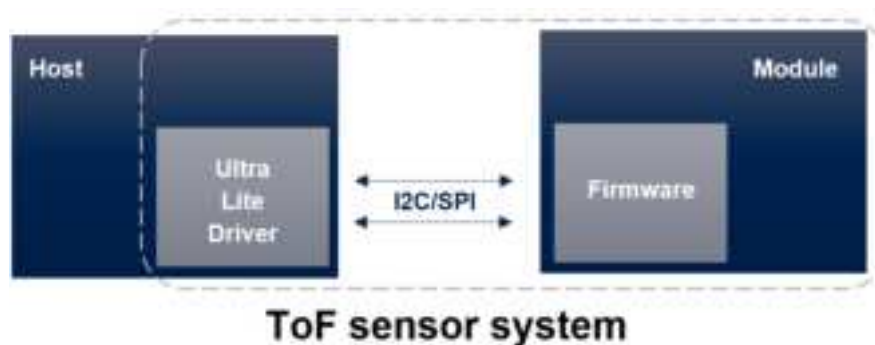
Acronym/abbreviation	Definition
CNH	compact normalized histogram
DOE	diffractive optical element
FoV	field of view
I2C	inter-integrated circuit (serial bus)
Kcps/SPAD	Kilo-count per second per SPAD (unit used to quantify the number of photons into the SPAD array)
RAM	random access memory
SCL	serial clock line
SDA	serial data
SPAD	single photon avalanche diode
ToF	Time-of-Flight
ULD	ultra lite driver
VCSEL	vertical-cavity surface-emitting diode
Xtalk	crosstalk

## 2 Functional description

### 2.1 System overview

The system is composed of a hardware module and the ultra light driver software (VL53LMZ ULD) running on a host (see figure below). The hardware module contains the ToF sensor. STMicroelectronics delivers the software driver, which is referred to in this document as "the driver". This document describes the functions of the driver, which are accessible to the host. These functions control the sensor and get the ranging data.

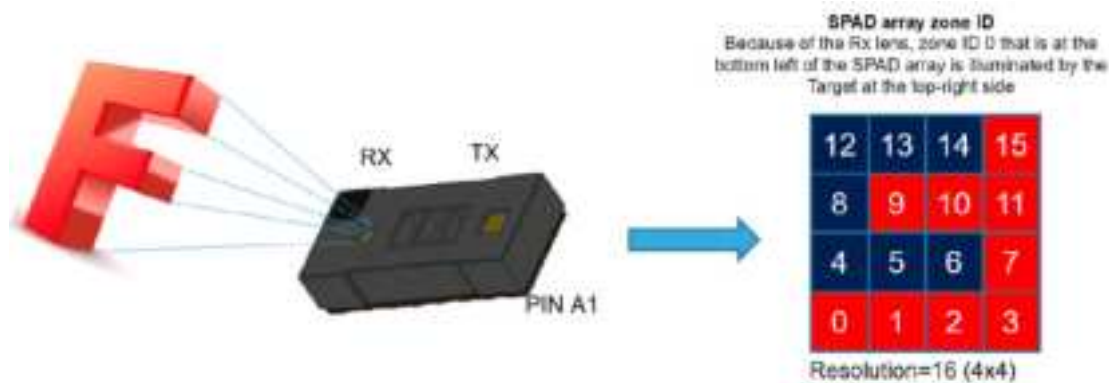
Figure 2. System overview



### 2.2 Effective orientation

The module includes a lens over the Rx aperture, which flips (horizontally and vertically) the captured image of the target. Consequently, the zone identified as zone 0, in the bottom left of the SPAD array, is illuminated by a target located at the top right-hand side of the scene.

Figure 3. Module effective orientation



## 2.3 Schematics and I2C/SPI configuration

The communication between driver and firmware is handled by the I2C for both sensors, with a maximum speed of 1 MHz. In addition, the VL53L8CH also has the possibility to communicate with SPI. In this case, the maximum SPI speed is 3 MHz. The implementation of each communication protocol requires pull-ups as described in the product datasheets.

The devices have a default I2C address of 0x52. However, it is possible to change the default address to avoid conflicts with other devices, or to facilitate adding multiple modules to the system for a greater system FoV. The I2C address can be changed using the `vl53lmz_set_i2c_address()` function. To use the SPI, the multisensor is wired using an independent slave configuration (the NCS pin).

In order to change the device I2C address without affecting others on the I2C bus, it is important to disable the I2C communication of the devices not being changed. The procedure is as follows:

1. Power up the system as normal.
2. Pull down the LPn pin of the device that will not have its address changed.
3. Pull up the LPn pin of the device that has the I2C address changed.
4. Program the I2C address to the device using function `vl53lmz_set_i2c_address()` function.
5. Pull up the LPn pin of the device not being reprogrammed.

All devices should now be available on the I2C bus. Repeat the above steps for all the devices in the system that require a new I2C address.

Figure 4. Multiple sensors on I2C bus

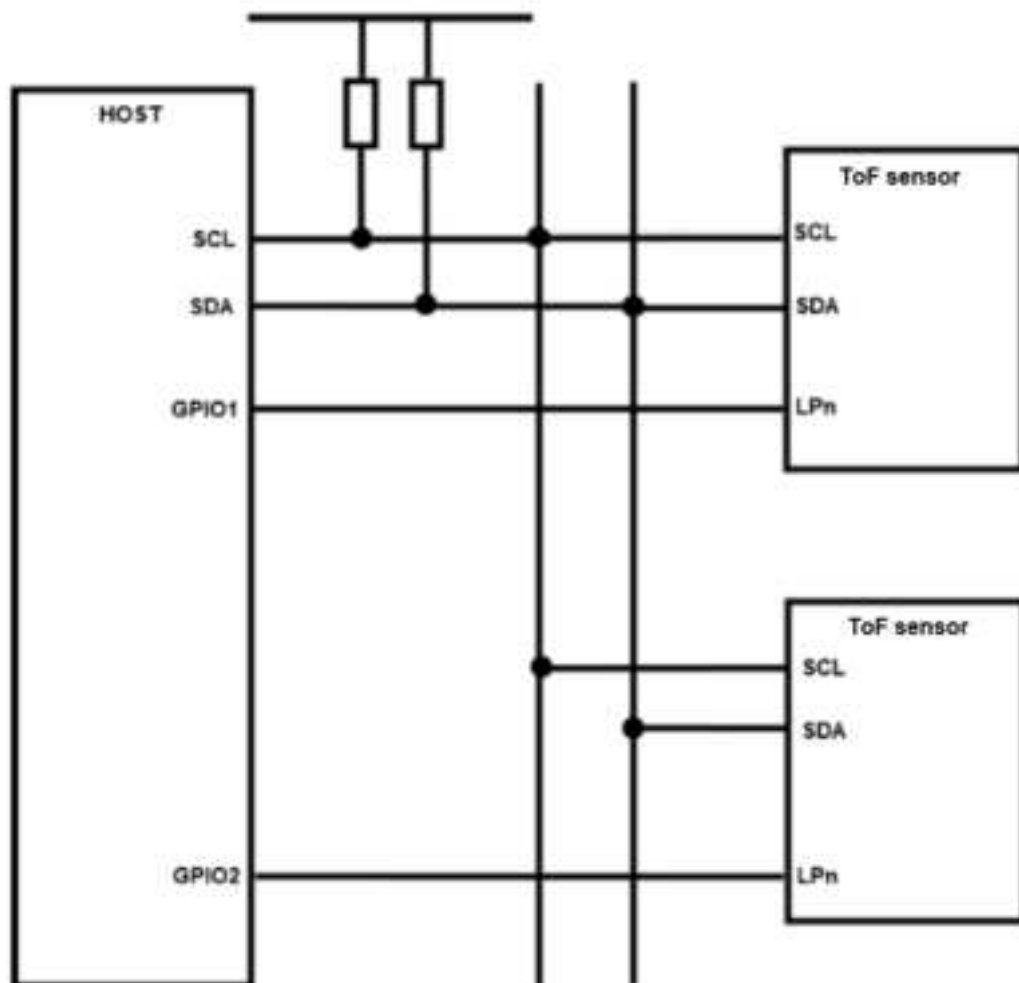
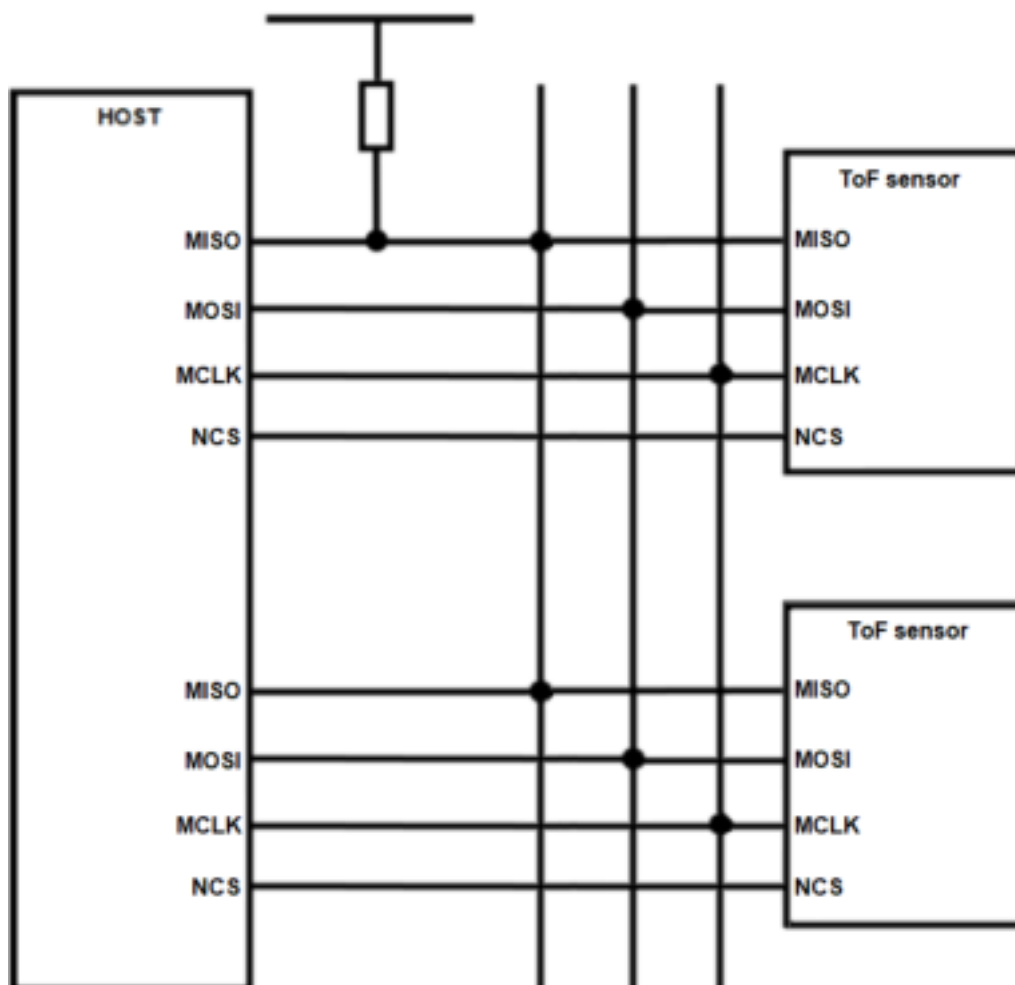


Figure 5. Multiple sensors on SPI



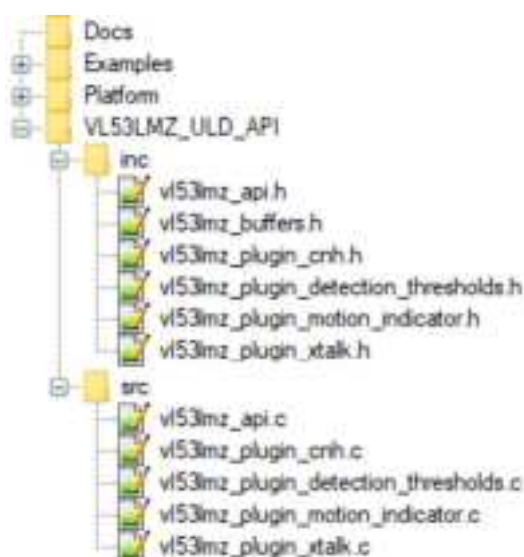
## 3 Package content and data flow

### 3.1 Driver architecture and content

The driver package is composed of four folders. The driver is located in the folder /VL53LMZ\_ULD\_API.

The driver is composed of mandatory and optional files. Optional files are plugins used to extend the ULD features. Each plugin starts with the word "vl53lmz\_plugin" (e.g vl53lmz\_plugin\_xtalk.h). If the user does not want the proposed plugins, they can be removed without impacting the other driver features. The following figure represents the mandatory files and the optional plugins.

Figure 6. Driver architecture



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

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

## 3.2 Calibration flow

Crosstalk (Xtalk) is defined as the amount of signal received on the SPAD array, which is due to VCSEL light reflection inside the protective window (cover glass) added on top of the module. The Time-of-Flight modules are self-calibrated, and can be used without any additional calibration.

Xtalk calibration may be required if the module is protected by a cover glass. The sensors are immune to crosstalk beyond 60 cm thanks to a histogram algorithm. However, at short distances below 60 cm, crosstalk can be larger than the actual returned signal. This gives a false target reading or makes targets appear closer than they really are. All crosstalk calibration functions are included in a crosstalk plugin (optional). The user needs to use the file 'vl53lmz\_plugin\_xtalk'.

The crosstalk can be calibrated once, and data can be saved so it can be reused later. A target at fixed distance, with a known reflectance is required. The minimum distance required is 600 mm, and the target must cover the whole FoV. Depending on the setup, the user can modify settings in order to adapt the crosstalk calibration, as proposed in the following table.

**Table 1. Available settings for calibration**

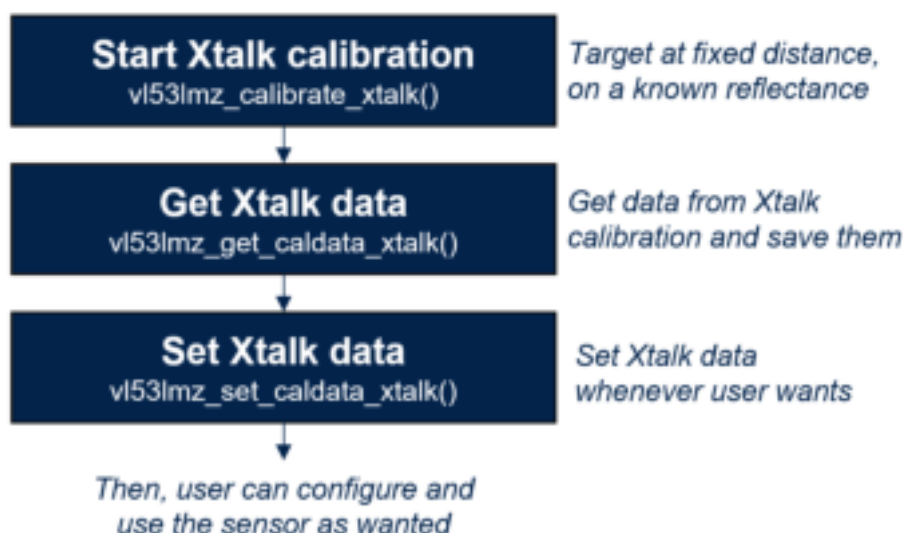
Setting	Min	Proposed by STMicroelectronics	Max
Distance [mm]	600	600	3000
Number of samples	1	4	16
Reflectance [%]	1	3	99

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

- 1 sample  $\approx$  1 second
- 4 samples  $\approx$  2.5 seconds
- 16 samples  $\approx$  8.5 seconds

The calibration is performed using the function `vl53lmz_calibrate_xtalk()`. This function can be used at any time. However, the sensor must be initialized first. The following figure represents the crosstalk calibration flow.

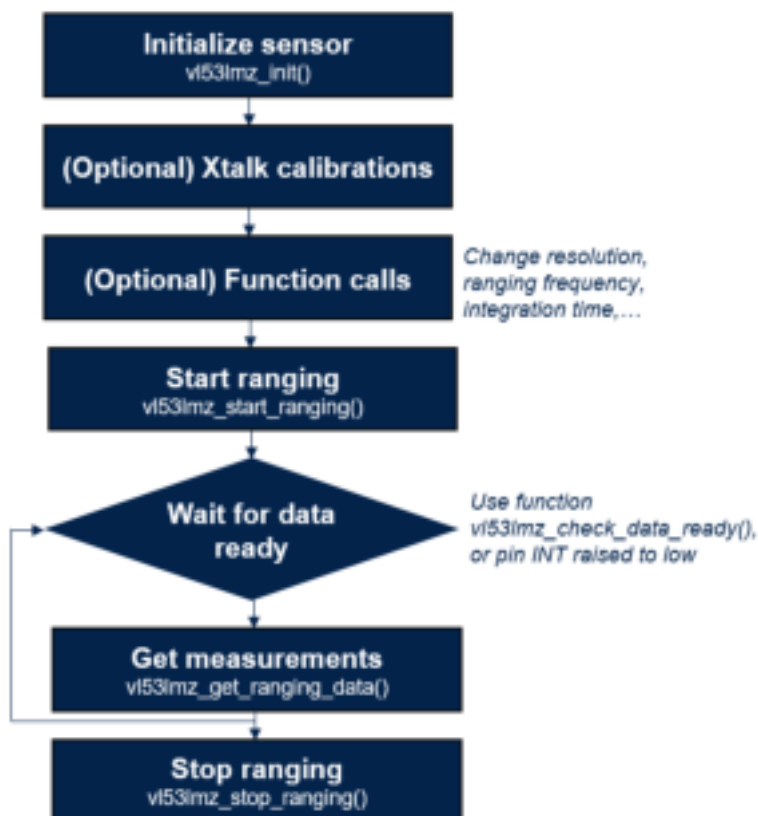
**Figure 7. Xtalk calibration flow**



### 3.3 Ranging flow

The following figure represents the ranging flow used to get measurements. Crosstalk calibration and optional function calls must be used before starting the ranging session. The get/set functions cannot be used during a ranging session, and 'on-the-fly' programming is not supported.

**Figure 8. Typical ranging flow**





## 4 Available features

The sensor driver includes several functions, which allow the user to tune the sensor, depending on the use case. All the functions available for the driver are described in the following sections.

### 4.1 Initialization

Initialization must be done before using the Time-of-Flight sensor. This operation requires the user to:

1. Power on the sensor
  - For VL53L7CH: VDDIO, AVDD, LPn pins set to high
  - For VL53L8CH: VDDIO, AVDD, CORE\_1V8 pins set to high
2. Call the function `vl53lmz_init()`. The function copies the firmware (~84 Kbytes) to the module. This is done by loading the code over the I2C/SPI interface, and performing a boot routine to complete the initialization.

### 4.2 Sensor reset management

To reset the device, the following pins need to be toggled:

1. Set pins VDDIO, and AVDD (and CORE\_1V8 for VL53L8CH) to low.
2. Wait 10 ms.
3. Set pins VDDIO, and AVDD (and CORE\_1V8 for VL53L8CH) pins to high.

*Note:* Toggling only I2C\_RST pin resets the I2C communication.

### 4.3 Resolution

The resolution corresponds to the number of available zones. The VL53L7CH and VL53L8CH sensors have two possible resolutions: 4x4 (16 zones) and 8x8 (64 zones). By default the sensor is programmed in 4x4.

The function `vl53lmz_set_resolution()` allows the user to change the resolution. As the ranging frequency depends on the resolution, this function must be used before updating the ranging frequency. Moreover, changing the resolution also increases the traffic size on the I2C/SPI bus when the results are read.

### 4.4 Ranging frequency

Ranging frequency can be used to change the measurement frequency. As the maximum frequency is different between 4x4 and 8x8 resolutions, this function needs to be used after choosing a resolution. The minimum and maximum allowed values are listed in the following table.

**Table 2. Minimum and maximum ranging frequencies**

Resolution	Min ranging frequency [Hz]	Max ranging frequency [Hz]
4x4	1	60
8x8	1	15

Ranging frequency can be updated using the function `vl53lmz_set_ranging_frequency_hz()`. By default, the ranging frequency is set to 1 Hz.

## 4.5 Ranging mode

Ranging mode allows the user to choose between ranging in high performance or low power consumption. There are two modes proposed:

- **Continuous:** The device continuously grabs frames with a ranging frequency defined by the user. The VCSEL is enabled during all ranging, so maximum ranging distance and ambient immunity are better. This mode is advised for fast ranging measurements or high performances.
- **Autonomous:** This is the default mode. The device continuously grabs frames with a ranging frequency defined by the user. The VCSEL is enabled during a period defined by the user, using the function `vl53l0x_set_integration_time_ms()`. As the VCSEL is not always enabled, the power consumption is reduced. The benefits are more obvious with a reduced ranging frequency. This mode is advised for low power applications.

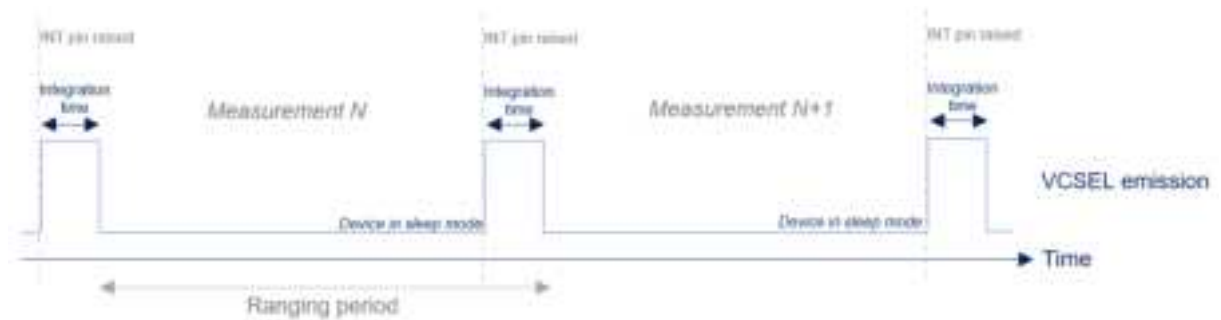
The ranging mode can be changed using the function `vl53l0x_set_ranging_mode()`.

## 4.6 Integration time

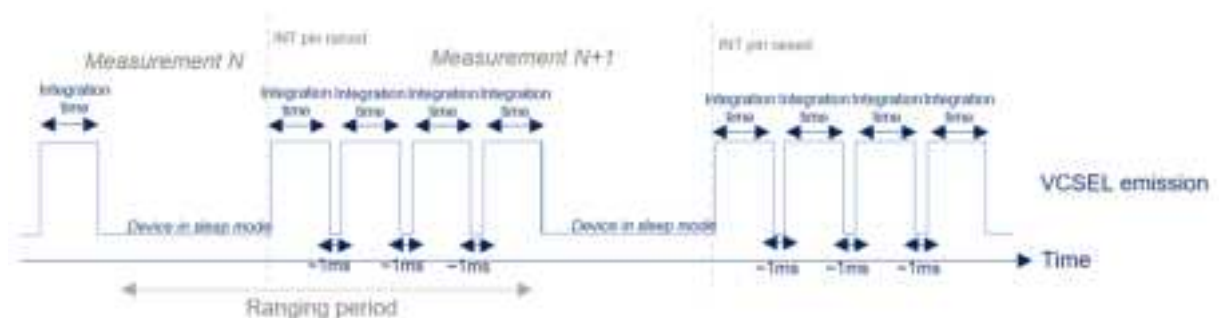
Integration time is a feature only available using autonomous ranging mode (refer to [Section 4.5 Ranging mode](#)). It allows the user to change the time while VCSEL is enabled. Changing integration time if ranging mode is set to continuous has no effect. The default integration time is set to 5 ms.

The effect of integration time is different for 4x4 and 8x8 resolutions. Resolution 4x4 is composed of one integration time, and 8x8 resolution is composed of four integration times. The following figures represent the VCSEL emission for both resolutions.

**Figure 9. Integration time for 4x4 autonomous**



**Figure 10. Integration time for 8x8 autonomous**



The sum of all integration times + 1 ms overhead must be lower than the measurement period. Otherwise, the ranging period is automatically increased to fit the integration time value.

## 4.7 Power modes

Power modes can be used to reduce the power consumption when the device is not used. The Time-of-Flight sensors can operate in one of the following power modes:

- Wake-up: The device is set in HP idle (high power), waiting for instructions.
- Sleep: The device is set in LP idle (low power), the low power state. The device cannot be used until set in wake-up mode. This mode retains the firmware and the configuration.

The power mode can be changed using the function `vl53lmz_set_power_mode()`. The default mode is wake-up.

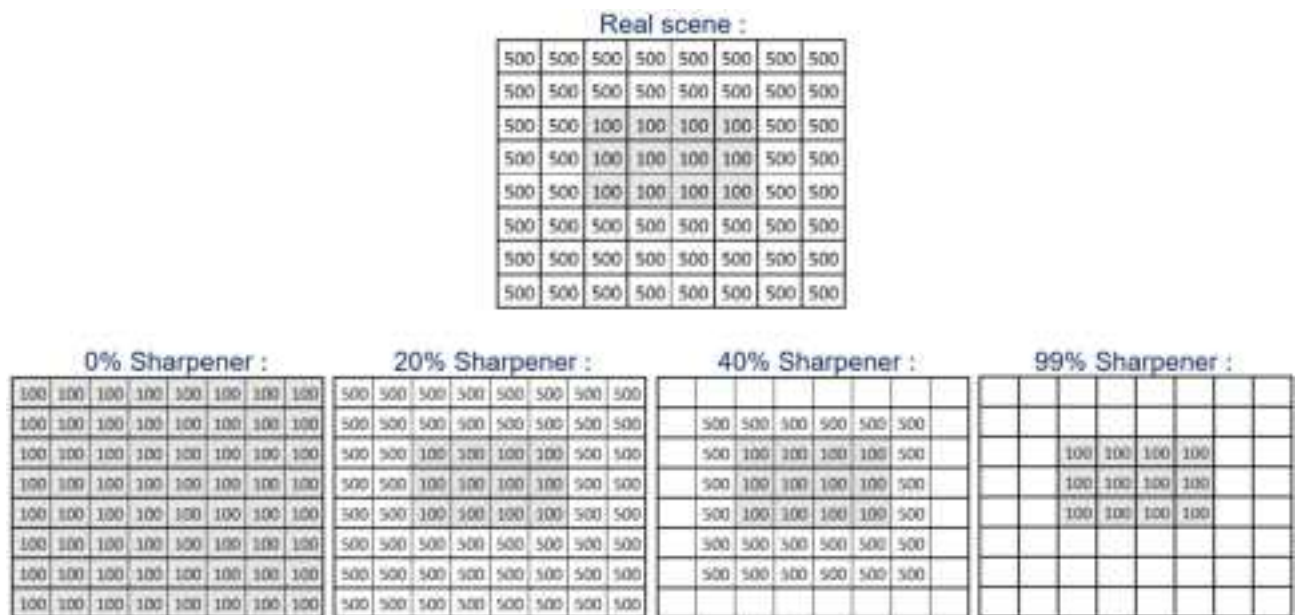
*Note: If the user wants to change the power mode, the device must not be in a ranging state.*

## 4.8 Sharpener

The signal returned from a target is not a clean pulse with sharp edges. The edges slope away and may affect the distances reported in adjacent zones. The sharpener is used to remove some or all of the signal caused by veiling glare.

The example shown in the following figure represents a close target at 100 mm centered in the FoV, and another target, further behind at 500 mm. Depending on the sharpener value, the close target may appear in more zones than the real one.

**Figure 11. Example of scene using several sharpener values**



Sharpener can be changed using the function `vl53lmz_set_sharpener_percent()`. The allowed values are between 0% and 99%. The default value is 5%.

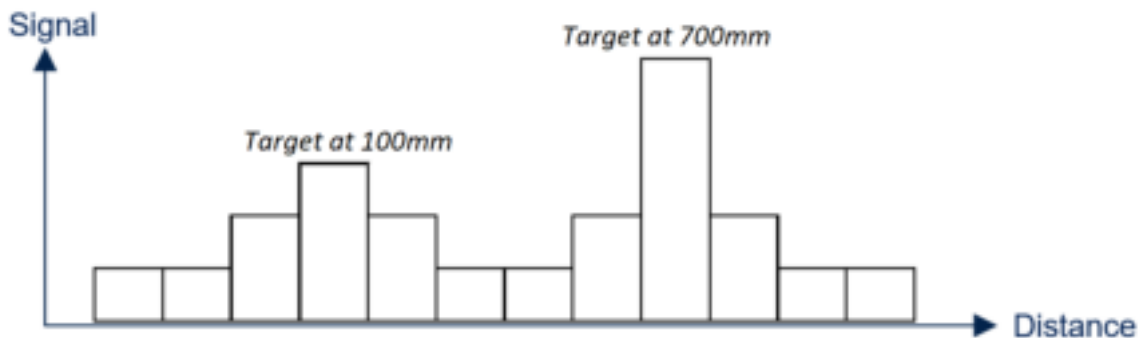
## 4.9 Target order

The sensors can measure several targets per zone. Thanks to the histogram processing, the host is able to choose the order of reported targets. There are two options:

- Closest: The closest target is the first reported
- Strongest: The strongest target is the first reported

The target order can be changed using the function `vl53lmz_set_target_order()`. The default order is "Strongest". The example in the following figure represents the detection of two targets. One at 100 mm with a low reflectance, and one at 700 mm with a high reflectance.

**Figure 12. Example of histogram with two targets**



## 4.10 Multiple targets per zone

The sensors can measure up to four targets per zone. The user can configure the number of targets returned by the sensor.

*Note:* The minimum distance between two targets to be detected in the same zone is 600 mm.

The selection is not possible from the driver; it has to be done in the 'platform.h' file. The macro `VL53LMZ_NB_TARGET_PER_ZONE` needs to be set to a value between 1 and 4. The target order described in [Section 4.9 Target order](#) directly impacts the order of detected target. By default, the sensor only outputs a maximum of one target per zone.

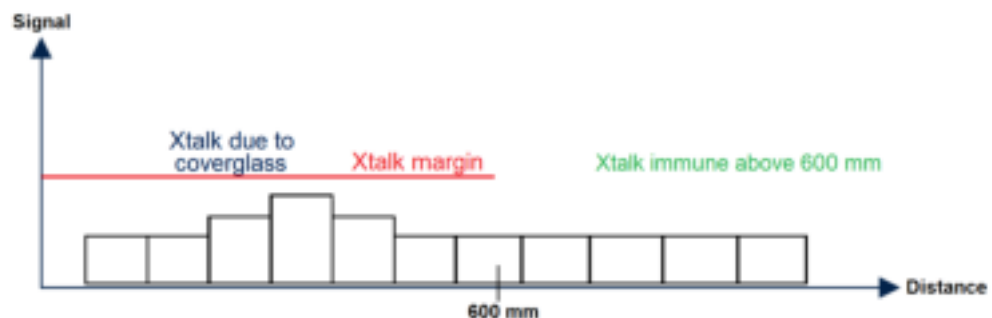
*Note:* An increased number of targets per zone increases the RAM size required by the driver.

## 4.11 Xtalk margin

The Xtalk margin is an additional feature only available using the plugin Xtalk. The .c and .f files `vl53lmz_plugin_xtalk` need to be used.

The margin is used to change the detection threshold when a cover glass is present on the top of the sensor. The threshold can be increased to ensure that the cover glass is never detected, after setting Xtalk calibration data. For example, the user can run a crosstalk calibration on one single device, and reuse the same calibration data for all other devices. The Xtalk margin can be used to tune the Xtalk correction. The figure below represents the Xtalk margin.

**Figure 13. Xtalk margin**



## 4.12 Detection thresholds

In addition to the regular ranging capabilities, the sensor can be programmed to detect an object under certain predefined criteria. This feature is available using the plugin "detection thresholds", which is an option not included by default in the API. The files called 'vl53lmz\_plugin\_detection\_thresholds' need to be used.

The feature can be used to trigger an interrupt to INT pin when conditions defined by the user are met. There are three possible configurations:

- Resolution 4x4: using one threshold per zone (total of 16 thresholds)
- Resolution 4x4: using two thresholds per zone (total of 32 thresholds)
- Resolution 8x8: using one threshold per zone (total of 64 thresholds)

Whatever the configuration used, the procedure for creating thresholds and the RAM size are the same. For each threshold combination, several fields need to be filled:

- Zone id: id of the selected zone (refer to [Section 2.2 Effective orientation](#))
- Measurement: measurement to catch (distance, signal, number of SPADs, ...)
- Type: windows of measurements (in windows, out of windows, below low threshold, ...)
- Low threshold: low threshold user for trigger. User does not need to set the format, it is automatically handled by the API.
- High threshold: high threshold user for trigger. User does not need to set the format, it is automatically handled by the API.
- Mathematic operation: only used for 4x4 – two threshold combinations per zone. The user can set a combination using several thresholds in one zone.

## 4.13 Interrupt autostop

The interrupt autostop feature is used to abort the ranging session during a measurement. By default, the sensor cannot be stopped during a measurement, because the frame measurements need to be completed. However, by using autostop, the frame measurements are aborted when an interrupt is triggered.

The autostop feature is useful when it is combined with a detection threshold. When a target is detected, the current measurement is automatically aborted. Autostop can be used in a customer state machine to switch quickly to another sensor configuration.

An interrupt autostop feature can be enabled by using the function `vl53lmz_set_detection_threshold_auto_stop()`. After a measurement is aborted, it is recommended to stop the sensor by using the function `vl53lmz_stop_ranging()`.

## 4.14 Motion indicator

The VL53L7CH and VL53L8CH sensors have an embedded firmware feature allowing motion detection in a scene. The motion indicator is computed between sequential frames. This option is available using the plugin 'vl53lmz\_plugin\_motion\_indicator'.

The motion indicator is initialized using the `vl53lmz_motion_indicator_init()` function. To change the sensor resolution, update the motion indicator resolution using the dedicated function: `vl53lmz_motion_indicator_set_resolution()`.

The user may also change the minimum and maximum distances for detecting motion. The difference between the minimum and maximum distances cannot be greater than 1500 mm. By default, distances are initialized with values between 400 mm and 1500 mm.

Results are stored in the field 'motion\_indicator'. In this field, the array 'motion' gives a value containing the motion intensity per zone. A high value indicates high motion variation between frames. A typical movement gives a value between 100 and 500. This sensitivity depends on the integration time, target distance, and target reflectance.

An ideal combination for low power applications is the use of the motion indicator with autonomous ranging mode, and detection thresholds programmed on the motion. This allows detection of movement variations in the FoV with minimum power consumption.

## 4.15 External synchronization pin (VL53L8CH only)

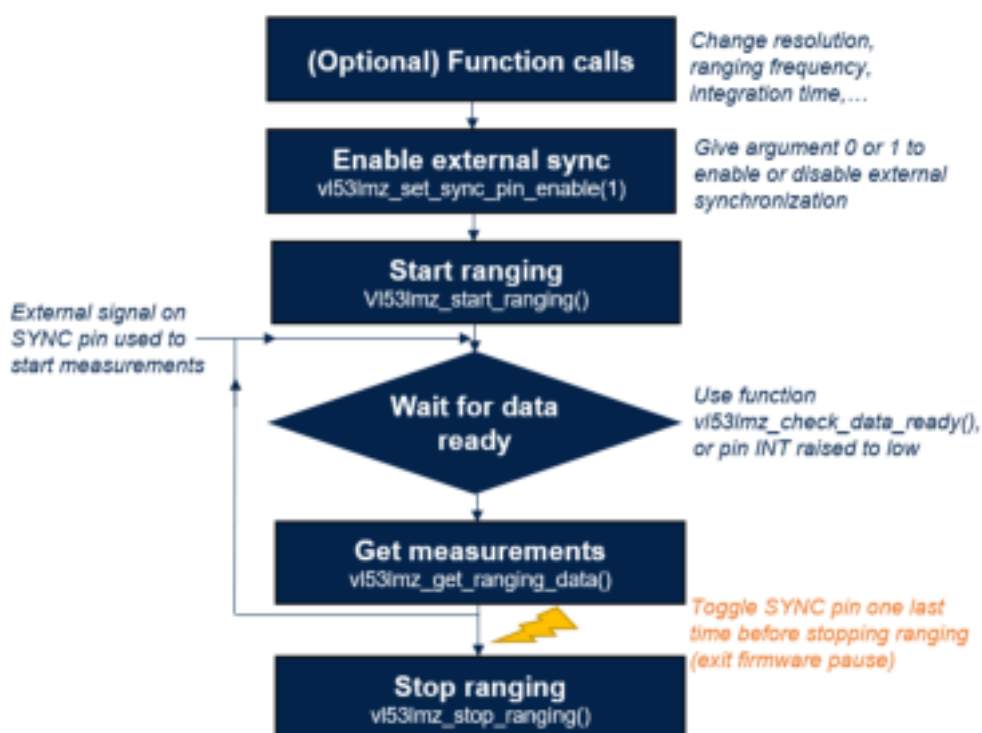
An external trigger source can be used with VL53L8CH to synchronize acquisitions. When the external synchronization is enabled, the VL53L8CH waits for an interrupt on the SYNC pin to start the next acquisition. To use this feature, the SYNC pin (B1) needs to be connected as described in the product datasheet.

There are no specific requirements for using the external synchronization. However, the VL53L8CH ranging frequency should be higher than the external signal frequency.

The external synchronization can be enabled or disabled by using the function `vl53lmz_set_external_sync_pin_enable()`. Ranging can be started as usual by using the function `vl53lmz_start_ranging()`. When a user wants to stop the sensor, it is recommended to toggle the SYNC pin to unpause the VL53L8CH firmware.

A topical flow for using the external synchronization pin is shown below.

Figure 14. External synchronization flow



## 5 Using the compact normalized histograms

### 5.1 Introduction

This section describes the compact normalized histogram (CNH) data that the VL53L7CH and VL53L8CH sensors can output.

The name derives from the following features of the data:

- Compact: Options are available to reduce the amount of data compared to the native “raw” histogram data.
- Normalized: Raw data are adjusted to compensate variations caused by frame-to-frame adjustments.
- Histogram: Primary data are in the form of histograms recording return-signal-strength vs range.

The configuration options available for the CNH data allow minimization of the total data transferred from the device to host. This enables optimization of framerate achievable for specific applications.

### 5.2 CNH configuration

CNH configuration to optimize data size is possible in two ways:

1. Zone aggregation. This is done by, optionally, summing data from multiple raw sensor zones into a lesser number of, aggregated, data structures.
2. Histogram configuration. Reducing the size of the raw histogram data by:
  - a. Defining a window of bins of interest.
  - b. Binning the bins to generate a lower resolution (in terms of target range) histogram.

### 5.3 Zone aggregation

Zone aggregation provides a means of combining and merging data from multiple device zones into a single CNH aggregate. This may be used to reduce the spatial resolution of data generated by the sensor. This is useful for applications that demand higher framerate but can accept lower resolution data.

The required zone aggregation is defined by an array, which is downloaded to the sensor before starting to range. This data structure is termed the AggregateMap. It contains the same number of elements as the zones which the device will run in, either 16 or 64. The AggregateMap defines the mapping from zones to CNH aggregates, with each entry in the map defining to which CNH aggregate the zone the data should be combined. Each entry in the map contains either the aggregate ID(number) to map the zone data to, or the number ‘-1’ to indicate the zone should not be mapped to any aggregate. Each zone can be mapped to only a single aggregate, but aggregates can receive data from multiple zones.

Aggregate IDs used in the aggregate map must be in a continuous range starting from 0. The total number of CNH aggregates generated is defined by the maximum entry in the aggregate map.

Data from multiple zones is combined by summing the data into the specified CNH aggregate.

To ease the creation of the aggregate maps the “vl53lmz\_plugin\_cnh module” contains the function vl53lmz\_cnh\_create\_agg\_map().

```
uint8_t vl53lmz_cnh_create_agg_map(VL53LMZ_Motion_Configuration *p_mi_config,
    int16_t resolution,
    int16_t start_x,
    int16_t start_y,
    int16_t merge_x,
    int16_t merge_y,
    int16_t cols,
    int16_t rows );
```

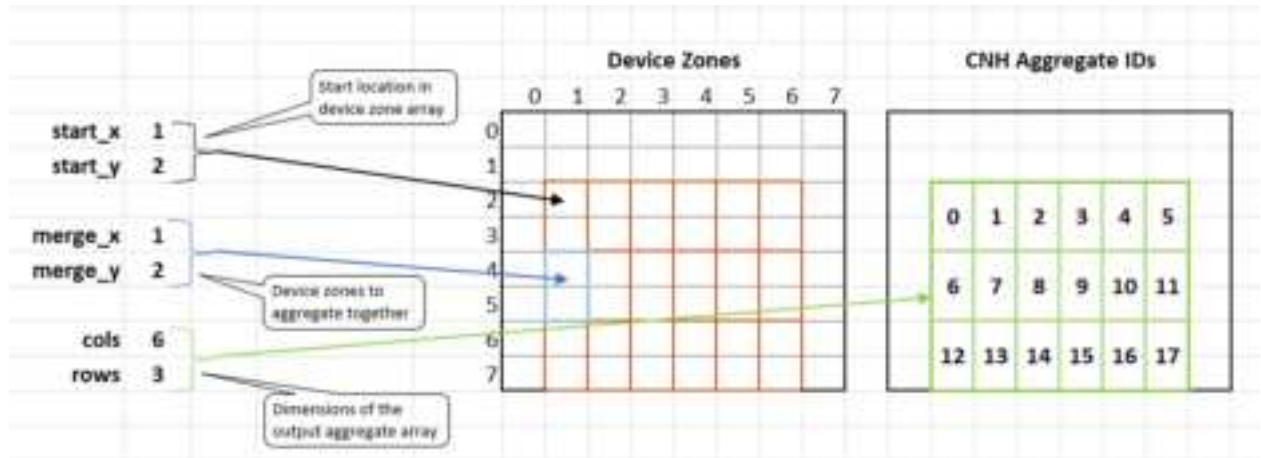
Parameters are:

- start\_x and start\_y: Defines the X and Y coordinates of the first zone to include in the CNH output.
- merge\_x and merge\_y: Defines how many raw zones to merge into a single aggregate. For instance merge\_x =2 and merge\_y =1 will merge a 2x1 area of raw zones into a single aggregate.
- cols and rows: Defines the number of columns and rows of aggregate data.

The aggregate map parameters define a mapping from the raw device zones, either 4x4, or 8x8, to an array of aggregate zones of size cnhCols x cnhRows. The figure below is an example of how the parameters may be used.



Figure 15. Example of CNH configuration



## 5.4 Histogram configuration

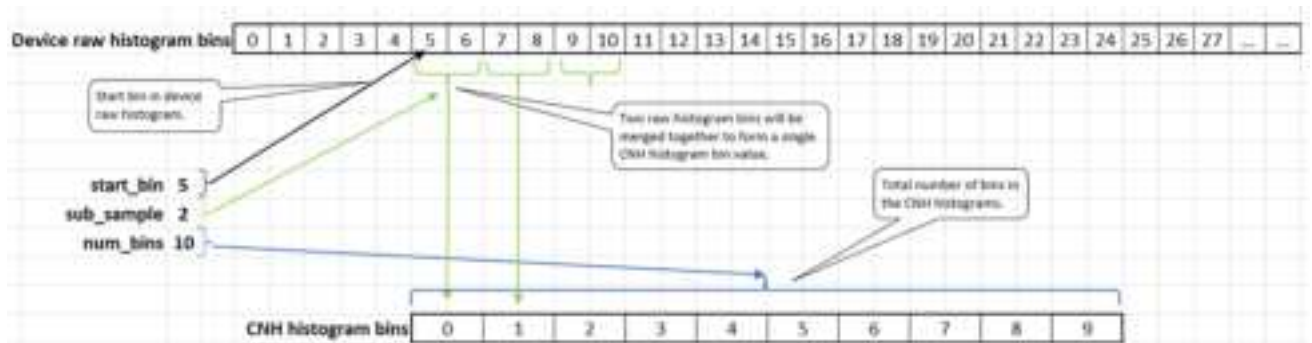
Configuration settings allow the histogram data to be transferred to the host to be tailored. These settings apply to all the aggregated histograms. There is no option to set them on a per-aggregate basis.

Three parameters are used to configure what histogram data to place into the CNH data.

- **start\_bin**: Defines the first bin within the raw device histogram that should be included in the CNH data.
- **num\_bins**: Defines how many bins of data the CNH histograms should contain.
- **sub\_sample**: Defines the number of raw histogram bins that should be combined (summed together) for each CNH histogram bin.

An example of the use of these parameters is shown below.

Figure 16. Example of histogram configuration



These histogram settings are parameters of the `vl53lhz_cnh_init_config()` function.

```
uint8_t vl53lhz_cnh_init_config( VL53LMZ_Motion_Configuration *p_mi_config,
    int16_t start_bin,
    int16_t num_bins,
    int16_t sub_sample );
```

## 5.5 Ambient light level data

In addition to the histogram, the CNH data also records the ambient light level the sensor zones were exposed to. For each CNH histogram there is an accompanying ambient light level, this value is accumulated over the same time period as the histogram and across the same set of zones that make up the CNH aggregate.

The ambient level is thus not in absolute units, such as W/m<sup>2</sup>, rather it is in arbitrary counts with the same scaling as used for histogram bins.



## 5.6 CNH data size constraints

The maximum size of the CNH data on the VL53L7CH and VL53L8CH devices is limited to 6160 bytes. Care must be taken with the CNH configuration to ensure this limit is not exceeded.

- Each histogram bin value requires 5 bytes.
- Each ambient level value requires 5 bytes.
- There is a fixed overhead of 28 bytes that must also be accommodated.

The total memory required therefore is:

$$\text{Total\_CNH\_Bytes} = \text{<histogram bins>} * 5 + \text{<number of aggregates>} * 5 + 28$$

$$\text{Total\_CNH\_Bytes} = (\text{rows*cols*num\_bins}) * 5 + (\text{rows*cols}) * 5 + 28$$

## 5.7 CNH value format

Data values for histogram bins and ambient level data use a 5 bytes-per-value format to support a very wide dynamic range of sensor data.

For each value, 4 bytes are used to store a signed integer value and one byte is used to store a scaling value.

The final value should be calculated as a floating-point value as follows:

$$\text{Floating-point value} = 32\text{b\_signed\_integer} / (2^{8\text{b\_integer}})$$

## 5.8 CNH data

Because the size of the CNH data may vary widely based on the CNH configuration in operation, it is not efficient to use a fixed size of data transfer as it used for the zone based target range data. Instead, the VL53L7CH/ VL53L8CH ULD driver is enhanced to allow the size the data transferred to be set during initialization.

Refer to the example code contained within the ULD, file Example\_12\_cnh\_data.c, for an example of how this should be done.

Data generated by the sensor within the CNH buffer is split into a header block and four data sub-blocks. The size and location of the sub-blocks vary depending on the CNH configuration in operation. The order of blocks is as follows:

1. Header block: 28 bytes of data
2. Histogram data: (rows\*cols\*num\_bins) 32b integers
3. Histogram data: (rows\*cols\*num\_bins) 8b scalars
4. Ambient level: (rows\*cols) 32b integers
5. Ambient level: (rows\*cols) 8b scalars

Within the histogram, block data is ordered by aggregate ID then bin number.

Within the ambient block, data is ordered by aggregate ID.

Start locations of the blocks are aligned to 4-byte boundaries, with padding bytes inserted as necessary.

Within the vl53lmz\_plugin\_cnh module, the function vl53lmz\_cnh\_get\_block\_addresses() calculates the start location of each aggregate data area.

## 6 Ranging results

### 6.1 Available data

An extensive list of target and environment data may be output during ranging activities. The following table describes the parameters available to the user.

**Table 3. Available ranging output using the sensors**

Element	Nb bytes (RAM)	Unit	Description
Ambient per SPAD	256	Kcps/SPAD	Ambient rate measurement performed on the SPAD array, with no active photon emission, to measure the ambient signal rate due to noise.
Number of targets detected	64	None	Number of detected targets in the current zone. This value should be the first one to check to know a measurement validity.
Number of SPADs enabled	256	None	Number of SPADs enabled for the current measurement. A far or low reflective target activates more SPADs.
Signal per SPAD	256 x nb targets programmed	Kcps/SPAD	Quantity of photons measured during the VCSEL pulse.
Range sigma	128 x nb targets programmed	Millimeter	Sigma estimator for the noise in the reported target distance.
Distance	128 x nb targets programmed	Millimeter	Target distance
Target status	64 x nb targets programmed	None	Measurements validity. See <a href="#">Section 6.5 Results interpretation</a> for more information.
Reflectance	64 x number targets programmed	Percent	Estimated target reflectance in percent
Motion indicator	140	None	Structure containing the motion indicator results. The field 'motion' contains the motion intensity.

**Note:** For several elements (signal per spad, sigma, ...) access to data is different if the user has programmed more than one target per zone (see [Section 4.10 Multiple targets per zone](#)). See example codes for more information.

### 6.2 Customize output selection

By default, all sensor outputs are enabled. If needed, the user can disable some sensor output.

Disabling measurements is not available on the driver; it must be performed in the 'platform.h' file. The user can declare the following macros to disable outputs:

```
#define VL53LMZ_DISABLE_AMBIENT_PER_SPAD
#define VL53LMZ_DISABLE_NB_SPADS_ENABLED
#define VL53LMZ_DISABLE_NB_TARGET_DETECTED
#define VL53LMZ_DISABLE_SIGNAL_PER_SPAD
#define VL53LMZ_DISABLE_RANGE_SIGMA_MM
#define VL53LMZ_DISABLE_DISTANCE_MM
#define VL53LMZ_DISABLE_TARGET_STATUS
#define VL53LMZ_DISABLE_REFLECTANCE_PERCENT
#define VL53LMZ_DISABLE_MOTION_INDICATOR
```

Consequently, the fields are not be declared in the results structure, and the data is not transferred to the host. The RAM size and I2C/SPI size are reduced.

To ensure data consistency, STMicroelectronics recommends keeping 'number of target detected' and 'target status' enabled. It allows filtering the measurements depending on the target status (refer to [Section 6.5 Results interpretation](#)).

### 6.3 Getting ranging results

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

- Polling mode: Continuously uses the function `vl53lmz_check_data_ready()`. It detects a new stream count returned by the sensor.
- Interrupt mode: Waits for an interrupt raised on INT pin. The interrupt is automatically cleared after ~100  $\mu$ s.

When new data is ready, the results can be read using the function `vl53lmz_get_ranging_data()`. It returns an updated structure containing all selected output. As the device is asynchronous, there is no interrupt to clear to continue the ranging session.

This feature is available for both continuous and autonomous ranging modes.

### 6.4 Using raw firmware format

After transferring ranging data through I2C/SPI, there is a conversion between the firmware format and the host format. This operation is typically performed to have a ranging distance in millimeters as a default output of the sensor. If the user wants to use the firmware format, the following macro must be defined in the platform file:

```
#define VL53LMZ_USE_RAW_FORMAT
```

### 6.5 Results interpretation

The data returned by the sensors can be filtered in order to take into account the target status. The status indicates the measurement validity. The full status list is described in the following table.

**Table 4. List of available target status**

Target status	Description
0	Ranging data are not updated
1	Signal rate too low on SPAD array
2	Target phase
3	Sigma estimator too high
4	Target consistency failed
5	Range valid
6	Wrap around not performed (typically the first range)
7	Rate consistency failed
8	Signal rate too low for the current target
9	Range valid with large pulse (may be due to a merged target)
10	Range valid, but no target detected at previous range
11	Measurement consistency failed
12	Target blurred by another one, due to sharpener
13	Target detected but inconsistent data. Frequently happens for secondary targets.
255	No target detected (only if number of targets detected is enabled)

To have consistent data, the user needs to filter invalid target status. To give a confidence rating, a target with status 5 is considered as 100% valid. A status of 6 or 9 can be considered with a confidence value of 50%. All other statuses are below 50% confidence level.

## 6.6 Driver errors

When an error occurs using the sensor, the driver returns a specific error. The following table lists the possible errors.

**Table 5. List of errors available using the driver**

Target status	Description
0	No error
1	Timeout reached, the sensor took too much time to reply
2	Corrupted frame (only occur during ranging)
3	The module has been damaged and laser safety is compromised. The sensor cannot start.
4	There is no known module detected
66	The internal firmware is not able to give a correct answer
127	User programmed an incorrect setting (unknown resolution, ranging frequency too high, ...)
255	Major error. Usually a timeout error, due to an I2C/SPI error.
other	Combination of multiple errors described above

*Note: More error codes can be implemented by the host using the platform files.*

## Revision history

**Table 6. Document revision history**

Date	Version	Changes
16-Jun-2023	1	Initial release

## Contents

<b>1</b>	<b>Acronyms and abbreviations</b>	<b>2</b>
<b>2</b>	<b>Functional description</b>	<b>3</b>
2.1	System overview	3
2.2	Effective orientation	3
2.3	Schematics and I2C/SPI configuration	4
<b>3</b>	<b>Package content and data flow</b>	<b>6</b>
3.1	Driver architecture and content	6
3.2	Calibration flow	7
3.3	Ranging flow	8
<b>4</b>	<b>Available features</b>	<b>9</b>
4.1	Initialization	9
4.2	Sensor reset management	9
4.3	Resolution	9
4.4	Ranging frequency	9
4.5	Ranging mode	10
4.6	Integration time	10
4.7	Power modes	11
4.8	Sharpener	11
4.9	Target order	12
4.10	Multiple targets per zone	12
4.11	Xtalk margin	12
4.12	Detection thresholds	13
4.13	Interrupt autostop	13
4.14	Motion indicator	13
4.15	External synchronization pin (VL53L8CH only)	14
<b>5</b>	<b>Using the compact normalized histograms</b>	<b>15</b>
5.1	Introduction	15
5.2	CNH configuration	15
5.3	Zone aggregation	15
5.4	Histogram configuration	16
5.5	<b>Ambient light level data</b>	16
5.6	CNH data size constraints	17
5.7	CNH value format	17
5.8	CNH data	17

---

<b>6</b>	<b>Ranging results.....</b>	<b>18</b>
6.1	Available data .....	18
6.2	Customize output selection .....	18
6.3	Getting ranging results .....	19
6.4	Using raw firmware format .....	19
6.5	Results interpretation .....	19
6.6	Driver errors.....	20
	<b>Revision history .....</b>	<b>21</b>

**IMPORTANT NOTICE – READ CAREFULLY**

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2023 STMicroelectronics – All rights reserved