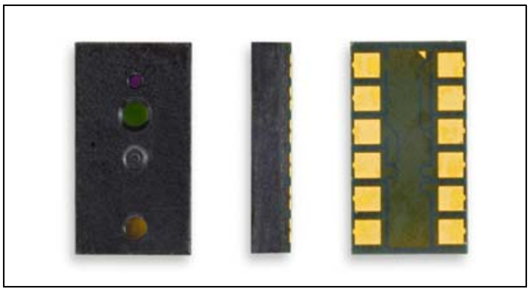
# VL6180X

## Proximity and ambient light sensing (ALS) module

**Datasheet - production data**



## Features

- Three-in-one smart optical module
  - Proximity sensor
  - Ambient Light Sensor
  - VCSEL light source
- Fast, accurate distance ranging
  - Measures absolute range from 0 to above 10 cm
  - Independent of object reflectance
  - Ambient rejection
  - Crosstalk compensation for cover glass
  - Ranging beyond 100mm is possible with certain target reflectances and ambient conditions but not guaranteed
- Gesture recognition
  - Distance and signal level can be used by host system to implement gesture recognition
  - Demo systems available.
- Ambient light sensor
  - High dynamic range
  - Accurate/sensitive in ultra-low light
  - Calibrated output value in lux
- Easy integration
  - Single reflowable component
  - No additional optics or gasket
  - Single power supply
  - $I^2C$ interface for device control and data
- Two programmable GPIO
  - Window and thresholding functions for both ranging and ALS

## Description

The VL6180X is the latest product based on ST's patented **Flight**Sense™ technology. This is a ground-breaking technology allowing absolute distance to be measured independent of target reflectance. Instead of estimating the distance by measuring the amount of light reflected back from the object (which is significantly influenced by color and surface), the VL6180X precisely measures the time the light takes to travel to the nearest object and reflect back to the sensor (Time-of-Flight).

Combining an IR emitter, a range sensor and an ambient light sensor in a three-in-one ready-to-use reflowable package, the VL6180X is easy to integrate and saves the end-product maker long and costly optical and mechanical design optimizations.

The module is designed for ultra low power operation. Ranging and ALS measurements can be automatically performed at user defined intervals. Multiple threshold and interrupt schemes are supported to minimize host operations.

Host control and result reading is performed using an $I^2C$ interface. Optional additional functions, such as measurement ready and threshold interrupts, are provided by two programmable GPIO pins.

## Applications

- Smartphones/portable touchscreen devices
- Tablet/laptop/gaming devices
- Domestic appliances/industrial devices

# Contents

# List of tables

# List of figures

# 1 Overview

This datasheet is applicable to the final VL6180X ROM code revision.

## 1.1 Technical specification

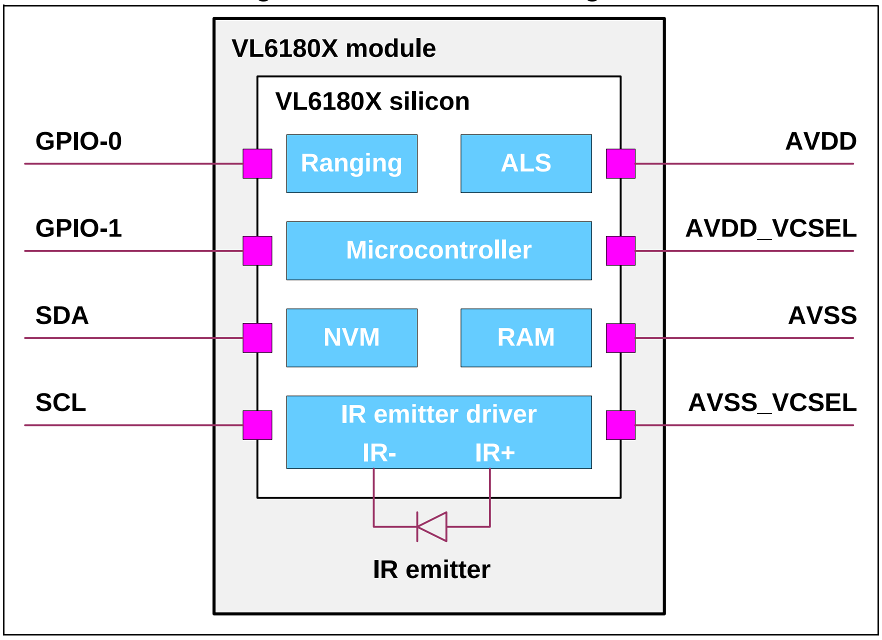
**Table 1. Technical specification**

| Feature | Detail |
|---|---|
| Package | Optical LGA12 |
| Size | 4.8 x 2.8 x 1.0 mm |
| Ranging | 0 to 100 mm(1) |
| Ambient light sensor | < 1 Lux up to 100 kLux(2)<br>16-bit output(3)<br>8 manual gain settings |
| Operating voltage:<br>• Functional range<br>• Optimum range(4) | <br>2.6 to 3.0 V<br>2.7 to 2.9 V |
| Operating temperature:<br>• Functional range<br>• Optimum range(4) | <br>-20 to 70°C<br>-10 to 60°C |
| Typical power consumption | Hardware standby (GPIO0 = 0): < 1 µA<br>Software standby: < 1 µA<br>ALS: 300 µA<br>Ranging: 1.7 mA (typical average)(5) |
| IR emitter | 850 nm |
| $I^2C$ | 400 kHz serial bus<br>Address: 0x29 (7-bit) |

1. Ranging beyond 100 mm is possible with certain target reflectances and ambient conditions but not guaranteed
2. When used under a cover glass with 10% transmission in the visible spectrum
3. Digital output easily converted to Lux
4. Please refer to *Table 19.: Ranging specification*
5. Assumes 10 Hz sampling rate, 17% reflective target at 50 mm

## 1.2 System block diagram

**Figure 1. VL6180X block diagram**




## 1.3 Device pinout

*Figure 2* shows the pinout of the VL6180X.

**Figure 2. VL6180X pinout**

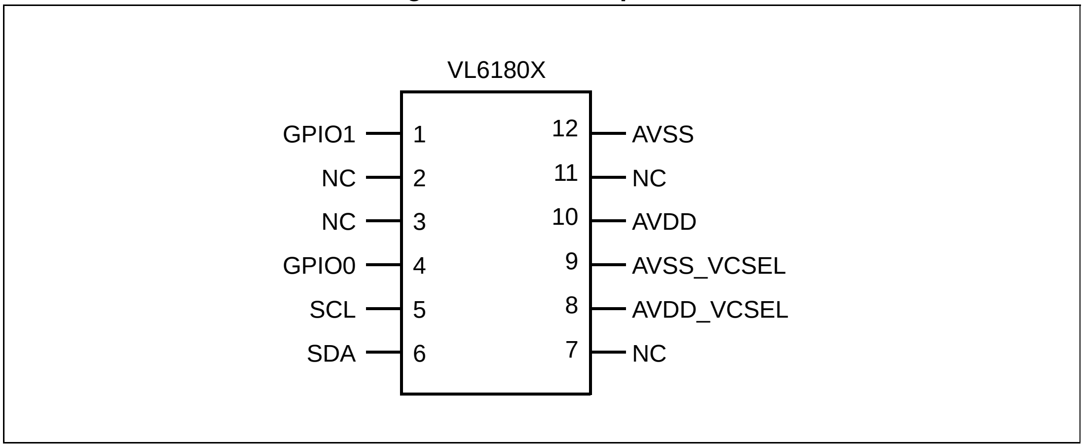


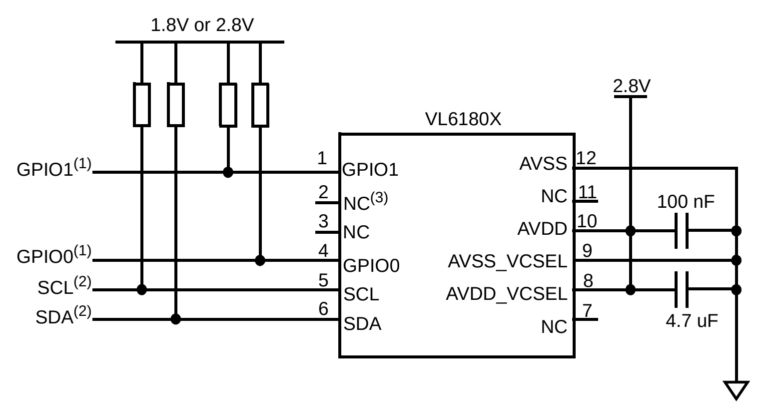
**Table 2. VL6180X pin numbers and signal descriptions**

| Pin number | Signal name | Signal type | Signal description |
|---|---|---|---|
| 1 | GPIO1 | Digital I/O | Interrupt output. Open-drain. |
| 2 | NC | | No connect or ground |
| 3 | NC | | No connect or ground |
| 4 | GPIO0/CE | Digital I/O | Power-up default is chip enable (CE). It should be pulled high with a 47 kΩ resistor. |
| 5 | SCL | Digital input | $I^2C$ serial clock |
| 6 | SDA | Digital I/O | $I^2C$ serial data |
| 7 | NC | | No connect or ground |
| 8 | AVDD_VCSEL | Supply | VCSEL power supply. 2.6 to 3.0 V |
| 9 | AVSS_VCSEL | Ground | VCSEL ground |
| 10 | AVDD | Supply | Digital/analog power supply. 2.6 to 3.0 V |
| 11 | NC | | No connect or ground |
| 12 | AVSS | Ground | Digital/analog ground |

## 1.4 Application schematic

*Figure 3* shows the schematic of the VL6180X.

**Figure 3. Root part number 1 schematic**

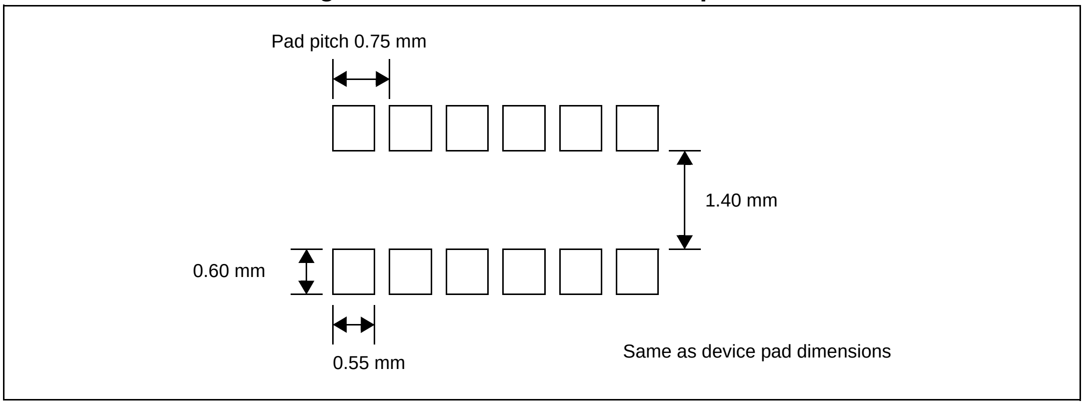



1. Open drain. Recommend 47 kΩ
2. Open drain. Pull up resistors typically fitted once per $I^2C$ bus at host
3. No connects can also be grounded if required

*Note: Capacitors on AVDD and AVDD_VCSEL should be placed as close as possible to the supply pads.*

## 1.5 Recommended solder pad dimensions

**Figure 4. Recommended solder pattern**




## 1.6 Recommended reflow profile

The recommend reflow profile is shown in *Figure 5* and *Table 3*.

**Figure 5. Recommended reflow profile**

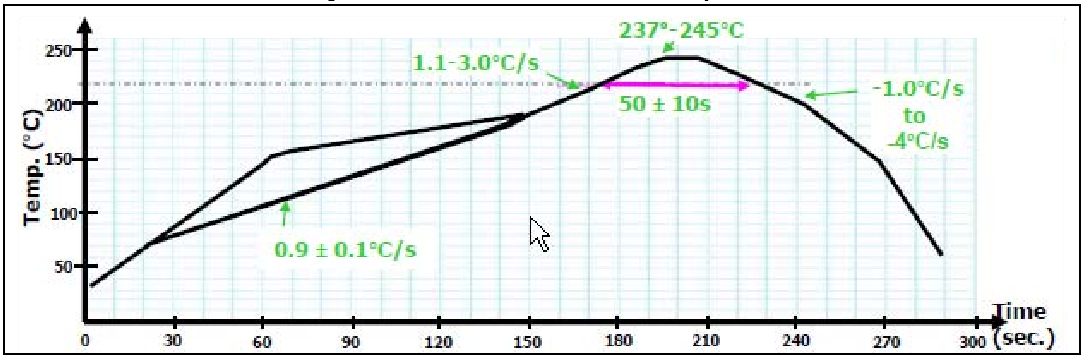



**Table 3. Recommended reflow profile**

| Profile | Ramp to strike | |
|---|---|---|
| Temperature gradient in preheat | (T= 70 - 180°C): | 0.9 +/- 0.1°C/s |
| Temperature gradient | (T= 200 - 225°C): | 1.1 - 3.0°C/s |
| Peak temperature in reflow | 237°C - 245°C | |
| Time above 220°C | 50 +/- 10 seconds | |
| Temperature gradient in cooling | -1 to -4 °C/s (-6°C/s maximum) | |
| Time from 50 to 220°C | 160 to 220 seconds | |

*Note: As the VL6180X package is not sealed, only a dry re-flow process should be used (such as convection re-flow). Vapor phase re-flow is not suitable for this type of optical component.*

The VL6180X is an optical component and as such, it should be treated carefully. This would typically include using a ‘no-wash’ assembly process.

# 2 Functional description

This section gives an overview of the key features of the VL6180X and describes the different modes of operation of the ALS and proximity sensors.

Typical ranging performance of the VL6180X is shown in *Figure 6*. This demonstrates the reflectance independence and range accuracy of the VL6180X from 0 to 100 mm for 3%, 5%, 17% and 88% reflective targets. The example shown here is with ST cover glass and a 1.0 mm air gap.

*Figure 7* shows typical ALS linearity vs gain over a wide dynamic range. More details about the ambient light sensor can be found in *Section 2.13*.

**Figure 6. Typical ranging performance**

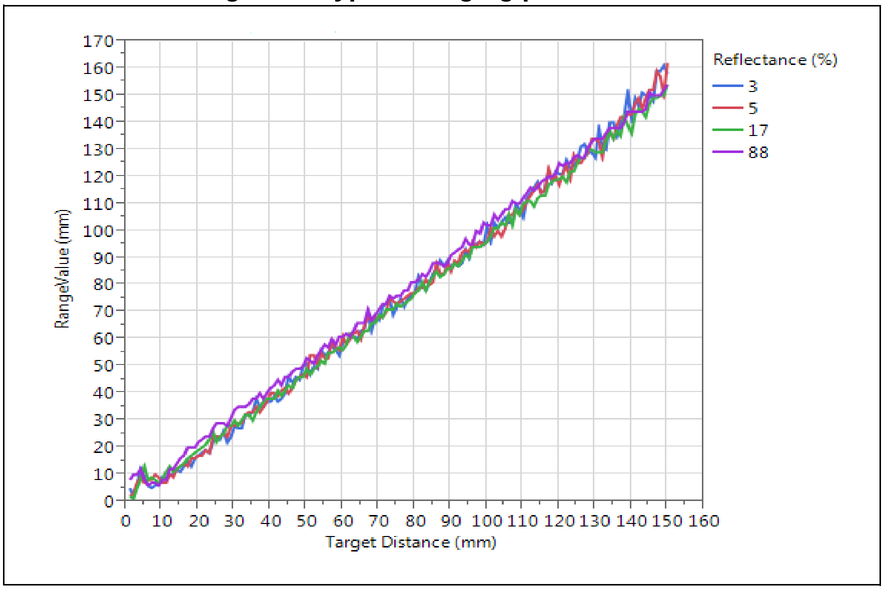
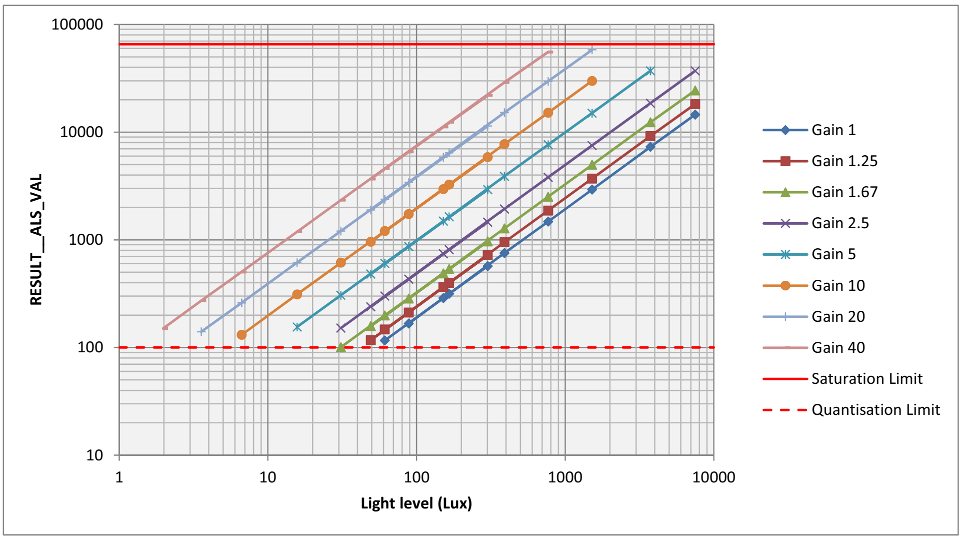


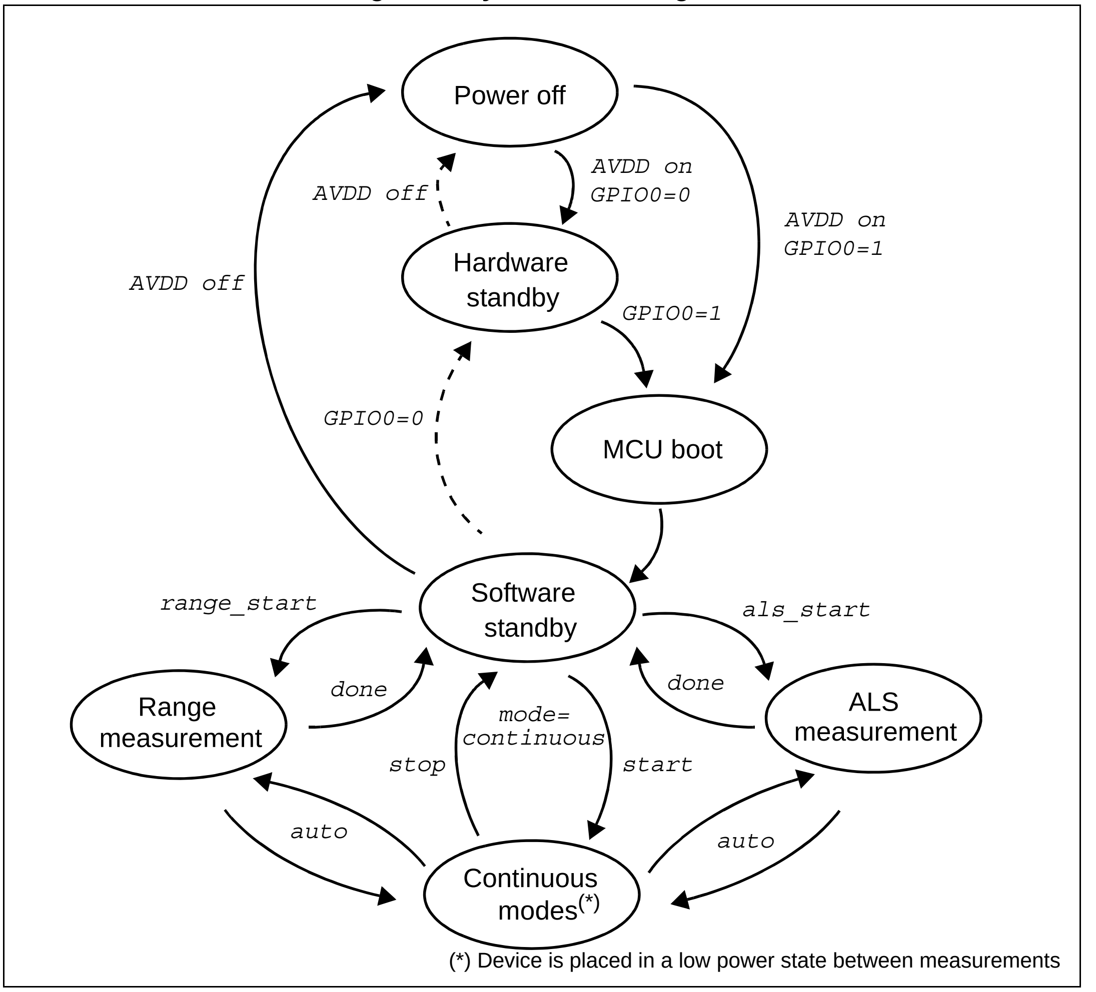

**Figure 7. ALS linearity**



## 2.1 System state diagram

*Figure 8* describes the main operating states of the VL6180X. Hardware standby is the reset state (GPIO0=0)[a]. The device is held in reset until GPIO0 is de-asserted. Note that the device will not respond to $I^2C$ communication in this mode. When GPIO0=1, the device enters software standby after the internal MCU boot sequence has completed. Once in software standby, ST recommended register initialization settings[b] can be applied along with any required application specific register settings. Thereafter, the host can command single-shot range or ALS measurements or alternatively program one of the continuous operating modes where the device uses an internal timer to schedule measurements at specified intervals. See *Section 2.4.3: Interleaved mode*.

**Figure 8. System state diagram**




(*) Device is placed in a low power state between measurements

a. Use of GPIO0 is optional

b. Please contact STMicroelectronics for the latest settings

## 2.2 Timing diagram

*Figure 9* and *Table 4.*show the Root part number 1 power-up timing constraints.

- AVDD_VCSEL must be applied before or at the same time as AVDD.
- GPIO0 defaults to an active low shutdown input. When GPIO0 = 0, the device is in hardware standby. If GPIO0 is not used it should be connected to AVDD.
- The internal microprocessor (MCU) boot sequence commences when AVDD is up and GPIO0 is high whichever is the later.
- GPIO1 power-up default is output low. It is tri-stated during the MCU boot sequence.

*Note: In hardware standby, GPIO1 is output low and will sink current through any pull-up resistor. This leakage can be minimized by increasing the value of the pull-up resistor.*

- After the MCU boot sequence the device enters software standby. The software standby state can be determined by polling `SYSTEM__FRESH_OUT_OF_RESET{0x16}`. Host initialization can commence immediately after entering software standby.

**Figure 9. Power-up timing**

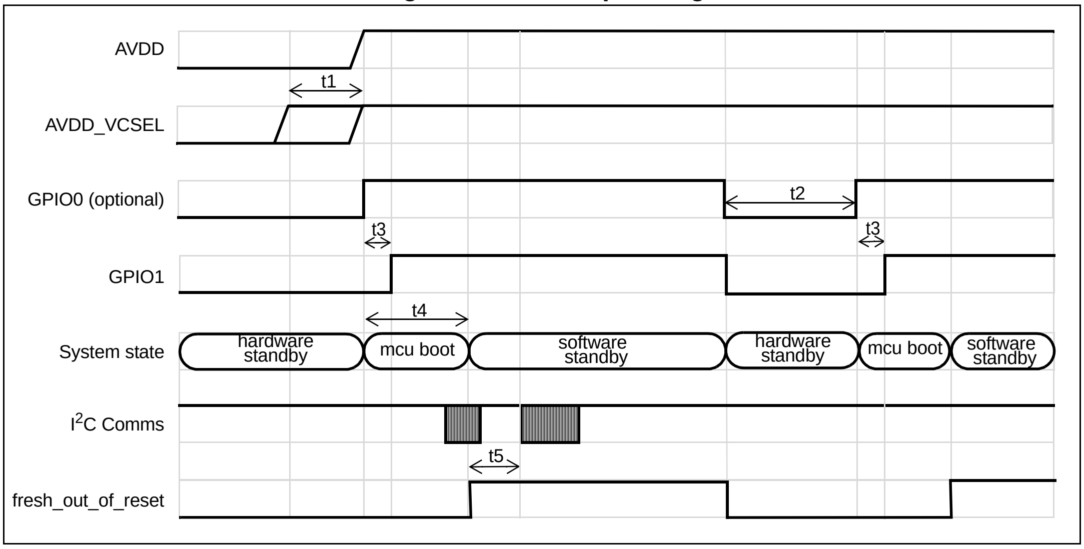


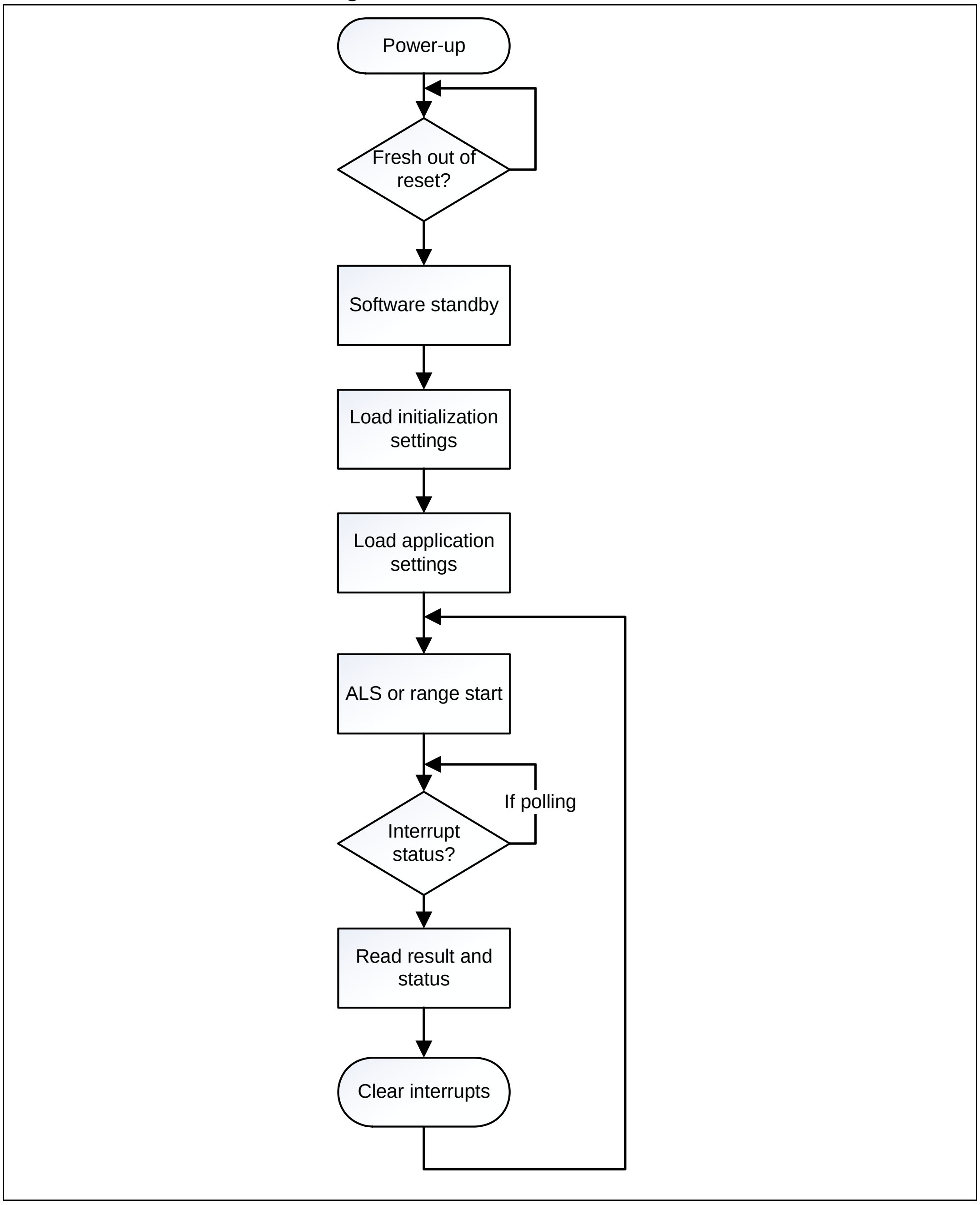

**Table 4. Power-up timing constraints**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| t1 | AVDD_VCSEL power applied after AVDD | - | 0 | ms |
| t2 | Minimum reset on GPIO0 | 100 | - | ns |
| t3 | GPIO1 output low after hardware standby | - | 400 | µs |
| t4 | MCU boot | - | 1 | ms |
| t5 | Software standby to host initialization | - | 0 | ms |

## 2.3 Software overview

*Figure 10* shows a simple start-up routine from initialization to completing a range or ALS measurement.

**Figure 10. Software overview**



## 2.4 Operating modes

*Table 5.* describes the operating modes of this device.

- Modes 1 and 2 are single-shot range and ALS measurements.
- Modes 3 and 4 are stand-alone, continuous operation for either range or ALS.
- Modes 5 and 6 are for mixed continuous and single-shot mode operations where regular measurements are required from one of the sensors and only occasional measurements are required from the other.

*Note: In modes 5 and 6, single-shot operation takes the priority i.e. if a scheduled measurement is in progress when the host requests a single-shot measurement, the scheduled measurement will be aborted and will resume on the next available time slot.*

- Mode 7 allows both ALS and range measurements to be scheduled at regular intervals. The ALS measurement is completed first immediately followed by a range measurement. Interleaved mode is described in more detail in *Section 2.4.3*.

**Table 5. Operating modes**

| Mode | Function | Range | | ALS | | Priority |
|---|---|---|---|---|---|---|
| | | Single | Continuous | Single | Continuous | |
| 1 | Range single-shot | • | | | | Range |
| 2 | ALS single-shot | | | • | | ALS |
| 3 | Range continuous | | • | | | Range |
| 4 | ALS continuous | | | | • | ALS |
| 5 | Range continuous and ALS single-shot | | • | • | | ALS |
| 6 | Range single-shot and ALS continuous | • | | | • | Range |
| 7 | Interleaved mode:<br>Range Continuous and ALS Continuous | | • | | • | - |

### 2.4.1 Single-shot range/ALS operation

A single-shot range or ALS measurement is performed as follows:

- Write 0x01 to the `SYSRANGE__START register{0x18}`.
- When the measurement is completed, bit 2 of `RESULT__INTERRUPT_STATUS_GPIO{0x4F}` will be set.
- Similarly, a single-shot ALS measurement is initiated by writing 0x01 to the `SYSALS__START register{0x38}`.
- When the measurement is completed, bit 5 of `RESULT__INTERRUPT_STATUS_GPIO{0x4F}` will be set.
  Note that in both cases the start bit, (bit 0) auto-clears.
- The range result is read from `RESULT__RANGE_VAL{0x62}`.
- The ALS result is read from `RESULT__ALS_VAL{0x50}`.
- Interrupt status flags are cleared by writing a ‘1’ to the appropriate bit of `SYSTEM__INTERRUPT_CLEAR{0x15}`.
- Bit 0 of `RESULT__RANGE_STATUS{0x4D}` and `RESULT__ALS_STATUS{0x4E}` indicate when either sensor is ready for the next operation.
- Error codes are indicated in bits [7:4] of the status registers

A detailed description of all the user accessible registers is given in *Section 6: Device registers*.

*Note: Single-shot ALS and range operations cannot be performed simultaneously. Only one of these operations should be performed at any one time and once started must be allowed to complete before another measurement is started. This is because any current operation will be aborted if another is started.*

### 2.4.2 Continuous range/ALS operation

A continuous range or ALS measurement is performed as follows:

- Write 0x03 to the `SYSRANGE__START` or `SYSALS__START` registers.
  In both cases, bit 1 of the register sets the mode to continuous
- When a measurement is completed either bit 2 or bit 5 of `RESULT__INTERRUPT_STATUS_GPIO{0x4F}` will be set.
- Results are read from `RESULT__RANGE_VAL{0x62}` or `RESULT__ALS_VAL{0x50}`.
- Interrupt status flags are cleared by writing a ‘1’ to the appropriate bit of `SYSTEM__INTERRUPT_CLEAR{0x15}`.
- Thereafter, measurements will be scheduled according to the relevant inter-measurement period (see `SYSRANGE__INTERMEASUREMENT_PERIOD{0x1B}` or `SYSALS__INTERMEASUREMENT_PERIOD{0X3E}`).
- Continuous mode operation can be stopped by writing 0 to either `START` register. Continuous operation will be halted immediately and any pending measurement will be aborted.

*Note: It is not recommended to run range and ALS continuous modes simultaneously (i.e. asynchronously). Instead, mode 7 ‘interleaved mode’ in Table 5. should be used. In ‘interleaved mode’, scheduled range and ALS measurements operate off a single timer with a range measurement proceeding immediately after every ALS measurement.*

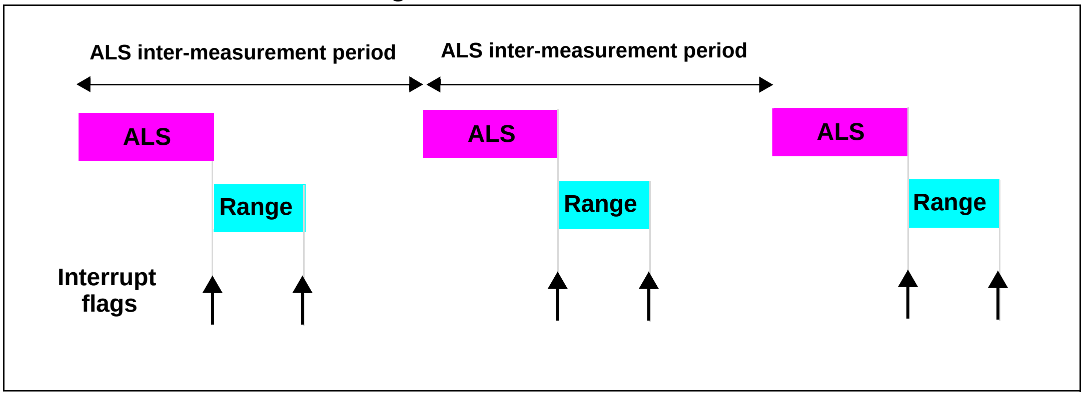
### 2.4.3 Interleaved mode

*Figure 11.* describes the continuous interleaved mode of operation where an ALS measurement is immediately followed by a range measurement and repeated after an interval specified by the ALS inter-measurement period.

To enable interleaved mode, set `INTERLEAVED_MODE_ENABLE{0x2A3} = 1. Use SYSALS__START` and `SYSALS__INTERMEASUREMENT_PERIOD` to control interleaved operation.

*Note: Continuous range settings have no effect in this mode.*

**Figure 11. Interleaved mode**




*Note: To ensure correct operation in any of the continuous modes, the user must ensure that the inter-measurement period is sufficient for the operation to be completed within the inter-measurement period. Failure to do so could result in unpredictable behavior.*

### 2.4.4 Continuous mode limits

To take account of oscillator tolerances and internal processing overheads it is necessary to place the following constraints on continuous mode operations. The following equations define the minimum inter-measurement period to ensure correct operation:

`Continuous range:`

SYSRANGE__MAX_CONVERGENCE_TIME + 5 ≤ SYSRANGE__INTERMEASUREMENT_PERIOD * 0.9

`Continuous ALS:`

SYSALS__INTEGRATION_TIME * 1.1 ≤ SYSALS__INTERMEASUREMENT_PERIOD * 0.9

`Interleaved mode:`

(SYSRANGE__MAX_CONVERGENCE _TIME + 5) + (SYSALS__INTEGRATION_TIME * 1.1) ≤ SYSALS__INTERMEASUREMENT_PERIOD * 0.9

*Table 6.* gives an example how to apply these limits in continuous interleaved mode operating at a sampling rate of 10 Hz.

**Table 6. Interleaved mode limits (10 Hz operation)**

| Parameter | Period (ms) |
|---|---|
| SYSALS__INTERMEASUREMENT_PERIOD | 100 |
| **Effective ALS INTERMEASUREMENT PERIOD** | **90** |
| SYSRANGE__MAX_CONVERGENCE_TIME | 30 |
| Total RANGE EXECUTION TIME | 35 |
| SYSALS__INTEGRATION_TIME | 50 |
| **Total ALS INTEGRATION TIME** | **55** |
| TOTAL EXECUTION TIME | 90 |

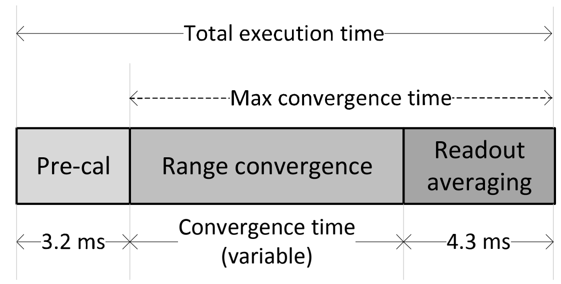
## 2.5 Range timing

*Figure 12* gives a breakdown of total execution time for a single range measurement.

- The pre-calibration phase is fixed (3.2 ms).
- The range convergence time is variable and depends on target distance/reflectance (see *Table 7*).
- The recommended readout averaging period is 4.3 ms. Readout averaging helps to reduce measurement noise. The recommended setting for `READOUT__AVERAGING_SAMPLE_PERIOD{0x10A}` is 48[(c)] but is programmable in the range 0-255. Note however that lower settings will result in increased noise.

*Note: When a target is detected, register `RESULT__RANGE_RETURN_CONV_TIME{0x80}` returns the actual convergence time before readout averaging. Range convergence and readout averaging must be completed within the specified max convergence time.*

**Figure 12. Total range execution time**




c. Default readout averaging period is calculated as follows: 1300 µs + (48 x 64.5 µs) = 4.3 ms

Table 7. Typical range convergence time (ms)

| Range (mm) | Target reflectance | | | |
|---|---|---|---|---|
| | 3% | 5% | 17% | 88% |
| 10 | 0.43 | 0.33 | 0.18 | 0.18 |
| 20 | 0.94 | 0.73 | 0.28 | 0.18 |
| 30 | 1.89 | 1.40 | 0.51 | 0.18 |
| 40 | 3.07 | 2.25 | 0.81 | 0.18 |
| 50 | 4.35 | 3.24 | 1.18 | 0.24 |
| 60 | 5.70 | 4.22 | 1.60 | 0.32 |
| 70 | 7.07 | 5.35 | 2.07 | 0.49 |
| 80 | 8.41 | 6.45 | 2.58 | 0.50 |
| 90 | 9.58 | 7.56 | 3.14 | 0.61 |
| 100 | 10.73 | 8.65 | 3.69 | 0.73 |

## 2.6 Interrupt modes

The VL6180X can be configured to generate an ALS or range interrupt flag under any of the following conditions:

- New sample ready
- Level low (RESULT__RANGE_VAL < SYSRANGE__THRESH_LOW)
- Level high (RESULT__RANGE_VAL > SYSRANGE__THRESH_HIGH)
- Out of window (RESULT__RANGE_VAL < SYSRANGE__THRESH_LOW) OR (RESULT__RANGE_VAL > SYSRANGE__THRESH_HIGH)

In new sample ready mode, an interrupt flag will be raised at the end of every measurement irrespective of whether the measurement is valid or if an error has occurred. This mode is particularly useful during development and debug. In level interrupt mode the system will raise an interrupt flag if either a low or high programmable threshold has been crossed. Out of window interrupt mode activates both high and low level thresholds allowing a window of operation to be specified. Interrupt modes for Range and ALS are configured via register SYSTEM__INTERRUPT_CONFIG_GPIO{0x14}.

*Note: In level or window interrupt modes range errors will only trigger an interrupt if the logical conditions described above are met.*

## 2.7 Range error codes

The system carries out a number of range checks during every range measurement to ensure the validity of each range result. Register RESULT__RANGE_STATUS{0x4D} returns an error code if one of the checks fails. *Table 8* gives a summary of the possible error codes.

**Table 8. Range error codes**

| Bits [7:4] | Error code | Description | Range (mm) |
|---|---|---|---|
| 0 | No error | Valid measurement | 0 - 200(1) |
| 1-5 | System error | System error detected. No measurement possible. | 255 |
| 6 | Early convergence estimate | ECE check failed | 255 |
| 7 | Max convergence | System did not converge before the specified max. convergence time limit | 255 |
| 8 | Range ignore | Ignore threshold check failed | 255 |
| 9-10 | Not used | - | - |
| 11 | SNR | Ambient conditions too high. Measurement invalidated | 255 |
| 12 | Raw range underflow | `RESULT__RANGE_RAW` < 0 (because offset is programmable a negative range result is possible) | 0 |
| 13 | Raw range overflow | `RESULT__RANGE_RAW` is out of range. This occurs typically around 200 mm | 255 |
| 14 | Range underflow | `RESULT__RANGE_VAL` < 0 (because offset is programmable a negative range result is possible) | 0 |
| 15 | Range overflow | `RESULT__RANGE_VAL` is out of range. This occurs typically around 200 mm | 255 |

1. Range overflow occurs typically around 200 mm.

## 2.8 Range checks

Error codes 6, 8 and 11 in *Table 8* are configurable by the user. They can be enabled/disabled via register `SYSRANGE__RANGE_CHECK_ENABLES{0x2D}` by setting or clearing the appropriate bit. The register default is 0x11 i,e, ECE and SNR enabled.

### 2.8.1 Early convergence estimate (ECE)

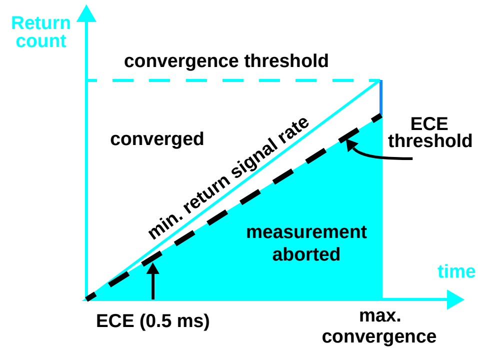
Early convergence estimate (ECE) is a programmable feature designed to minimize power consumption when there is no target in the field-of-view (FOV).

The system is said to have ‘converged’ (i.e. range acquired), when the convergence threshold[d] is reached before the max. convergence time limit (see *Figure 13*). This ratio specifies the minimum return signal rate required for convergence. If there is no target in the FOV, the system will continue to operate until the max. convergence time limit is reached before switching off thereby consuming power. With ECE enabled, the system estimates the

d. For standard ranging, the convergence threshold is set to 15360. The convergence threshold register is not accessible by the user.

return signal rate 0.5 ms after the start of every measurement. If it is below the ECE threshold, the measurement is aborted and an ECE error is flagged.

**Figure 13. Early convergence estimate (ECE)**




ECE is enabled by setting bit 0 of `SYSRANGE__RANGE_CHECK_ENABLES{0x02D}`. If enabled, the ECE threshold must be specified. To set the ECE threshold 20% below the minimum convergence rate, the ECE threshold is calculated as follows:

$$\text{ECE threshold} = \frac{80\% \times 0,5 \times 15360}{\texttt{SYSRANGE\_\_MAX\_CONVERGENCE\_TIME (in ms)}}$$

The 16-bit ECE threshold should be written to `SYSRANGE__EARLY_CONVERGENCE_ESTIMATE{0x22}`. For example, if `SYSRANGE__MAX_CONVERGENCE_TIME{0x1c}` is set to 30 ms, the ECE threshold is 204. If the return count is less than 204 after 0.5 ms, the measurement will be aborted.

*Note: The optimum value for the ECE threshold should be determined in the final application.*

### 2.8.2 Range ignore

In a system with cover glass, the return signal from the glass (cross-talk) may be sufficient to cause the system to converge and return a valid range measurement even when there is no target present. The range ignore feature is designed to ensure that the system does not range on the glass. (Cross-talk is described in more detail in *Section 2.12.2*).

The ignore threshold is enabled by setting bit 1 of `SYSRANGE__RANGE_CHECK_ENABLES{0x02D}`. If enabled, the ignore threshold must be specified. In the follow example, the ignore threshold is set 20% above the system cross-talk:

$$\texttt{SYSRANGE\_\_RANGE\_IGNORE\_THRESHOLD\{0x26\}} = \text{cross-talk (Mcps)} \times 120\%$$

A range ignore error will be flagged if the return signal rate is less than the ignore threshold.

*`SYSRANGE__RANGE_IGNORE_VALID_HEIGHT`* should be set to 255.

*Note: The optimum value for the ignore threshold should be determined in the final application.*

### 2.8.3 Signal-to-noise ratio (SNR)

In high ambient conditions range accuracy can be impaired so the SNR threshold is used as a safety limit to invalidate range measurements where the ambient/signal ratio is considered too high.The default ambient/signal ratio limit is 10 (i.e. an SNR of 0.1) which is then encoded in 4.4 format as follows:

`SYSRANGE__MAX_AMBIENT_LEVEL_MULT{0x2C}`= 10 x 16 = 160

To enable the SNR check, set bit 4 in `SYSRANGE__RANGE_CHECK_ENABLES (0x02D)`. A lower setting results in a more aggressive filter which will result in a lower effective range but greater accuracy. A higher setting results in a less aggressive filter which will result in a greater effective range but lower accuracy.

The SNR value can be calculated as follows:

$$\mathrm{SNR} = \frac{\mathrm{RESULT\_\_RANGE\_RETURN\_SIGNAL\_COUNT}\{0x6C\}}{\mathrm{RESULT\_\_RANGE\_RETURN\_AMB\_COUNT}\{0x74\} * 6}$$

*Note: The SNR value is the inverse of the ambient/signal ratio limit {0x2C}.*

*Note: The optimum value for SNR threshold should be determined in the final application.*

## 2.9 Manual/autoVHV calibration

SPAD[e] sensitivity is temperature dependent so VHV[f] calibration is used to regulate SPAD sensitivity over temperature in order to minimize signal rate variation. VHV calibration is performed either manually by the host processor or automatically by internal firmware. Execution time is typically 200 µs so has no impact on normal operation.

A VHV calibration is run once at power-up and then automatically after every N range measurements defined by the `SYSRANGE__VHV_REPEAT_RATE{0x31} register`. AutoVHV calibration is disabled by setting this register to 0. Default is 255. If autoVHV is disabled it is recommended to run a manual VHV calibration periodically to recalibrate for any significant temperature variation. A manual VHV calibration is performed by setting `SYSRANGE__VHV_RECALBRATE{0x2E}` to 1. This register auto-clears. This operation should only be performed in software standby.

## 2.10 History buffer

The history buffer is a 8 x 16-bit memory which can be used to store the last 16 range measurements (8-bit) or 8 ALS samples (16-bit). Use of the history buffer is controlled via register `SYSTEM__HISTORY_CTRL{0x12}`. There are 3 basic functions:

- enable
- range or ALS selection
- clear buffer

e. Photon detectors - Single Photon Avalanche Diodes

f. VHV is an adjustable SPAD bias voltage and stands for Very High Voltage (typically around 14 V). Also sometimes referred to as CP (Charge Pump).

The buffer is read via eight 16-bit registers (`RESULT__HISTORY_BUFFER_0{0x52}` to `RESULT__HISTORY_BUFFER_7{0x60})`. The buffer holds the last 16 x 8-bit range or 8 x 16-bit ALS results as shown in *Table 9*.

**Table 9. History buffer**

| History buffer | Range | | ALS |
|---|---|---|---|
| | (High byte) | (Low byte) | (Word) |
| 0 | Range [15] (newest) | Range [14] | ALS [7] (newest) |
| 1 | Range [13] | Range [12] | ALS [6] |
| 2 | Range [11] | Range [10] | ALS [5] |
| 3 | Range [9] | Range [8] | ALS [4] |
| 4 | Range [7] | Range [6] | ALS [3] |
| 5 | Range [5] | Range [4] | ALS [2] |
| 6 | Range [3] | Range [2 | ALS [1] |
| 7 | Range [1] | Range [0] (oldest) | ALS [0] (oldest) |

*Note: Only one data stream (ALS or range) can be buffered at one time. There is no associated time stamp information.*

*The clear buffer command is not immediate; it takes effect on the next range or ALS start command.*

The history buffer works independently of interrupt control i.e. the history buffer records all new samples; its operation is unchanged in threshold and window modes.

## 2.11 Current consumption

*Table 10.* gives an overview of current consumption in different operating states.

**Table 10. Typical current consumption in different operating states**

| Mode | Current | Conditions |
|---|---|---|
| Hardware standby | < 1 µA | Shutdown (GPIO0 = 0). No I²C comms |
| Software standby | < 1 µA | After MCU boot. Device ready |
| ALS | 300 µA | During integration |
| Ranging | 1.7 mA | Average consumption during ranging(1) |

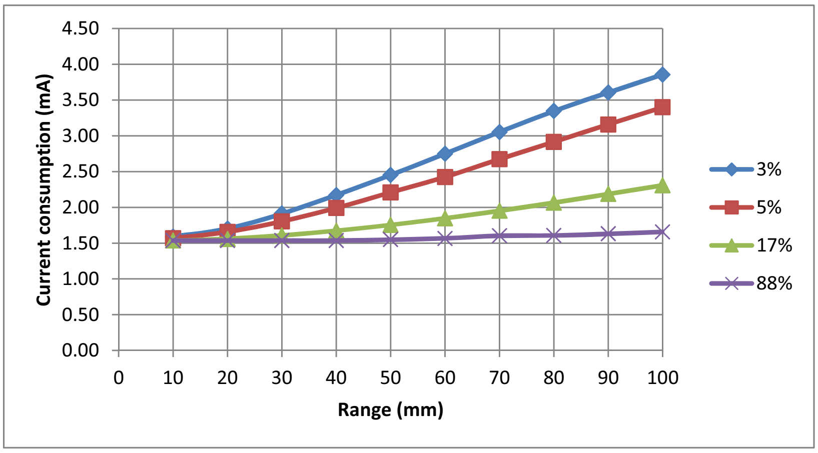
1. 10 Hz sampling rate, 17% reflective target at 50 mm.

### 2.11.1 Ranging current consumption

*Figure 14.* shows typical ranging current consumption of the VL6180X. Current consumption depends on target distance, target reflectance and sampling rate. The example shown here is based on default settings and a sampling rate of 10 Hz. The average current consumption for a 17% reflective target at 50 mm operating at 10 Hz is 1.7 mA. At different sampling rates

the current consumption scales accordingly i.e. the average current consumption at 1 Hz under the same conditions would be 0.17 mA.

**Figure 14. Typical ranging current consumption (10 Hz sampling rate)**




The minimum average current consumption in *Figure 14.* is 1.5 mA, 0.5 mA of which comes from pre-calibration before each measurement and 1.0 mA from post-processing (readout averaging). Pre-calibration is a fixed overhead but readout averaging can be reduced or effectively disabled by setting the `READOUT__AVERAGING_SAMPLE_PERIOD{0x10A}` to zero (default setting is 48).

*Note: Decreasing the READOUT__AVERAGING_SAMPLE_PERIOD will increase sampling noise. It is recommended that any change in setting be properly evaluated in the end application.*

Minimum current consumption scales with sampling rate i.e. at a sampling rate of 1 Hz the current consumption associated with pre- and post-processing will be 0.15 µA.

## 2.11.2 Current consumption calculator

*Table 11.* gives a breakdown of typical current consumption for pre-calibration, ranging and readout averaging.

**Table 11. Breakdown of current consumption**

| Label | Phase | I (mA) | t (ms) | Q (µC) = I x t |
|---|---|---|---|---|
| $Q_1$ | Pre-calibration | 13.0 | 3.2 | 41.6 |
| $Q_2$ | Ranging | 22.0 | per ms | 22.0 per ms |
| $Q_3$ | Readout averaging | 25.0 | per ms | 25.0 per ms |

Current consumption can then be calculated as follows:

I (µA) = sampling_rate * [$Q_1$ + ($Q_2$ * `RESULT__RANGE_RETURN_CONV_TIME in ms`) + $Q_3$ * (1.3 + (`READOUT__AVERAGING_SAMPLE_PERIOD` * 0.0645 ms))]

*Table 7.* gives typical convergence times for different target reflectance.

So, for example, `RESULT__RANGE_RETURN_CONV_TIME for a 3% target at 50 mm is 4.35 ms. At 10 Hz sampling rate this gives:`

I (µA) = 10 * [41.6 + (22 * 4.35) + 25 * (1.3 + (48 * 0.0645))] = 2472 µA

### 2.11.3 Current distribution

*Table 12.* shows how current consumption is distributed between the two supplies in ranging mode. AVDD_VCSEL supplies the VCSEL current and AVDD supplies all other functions.

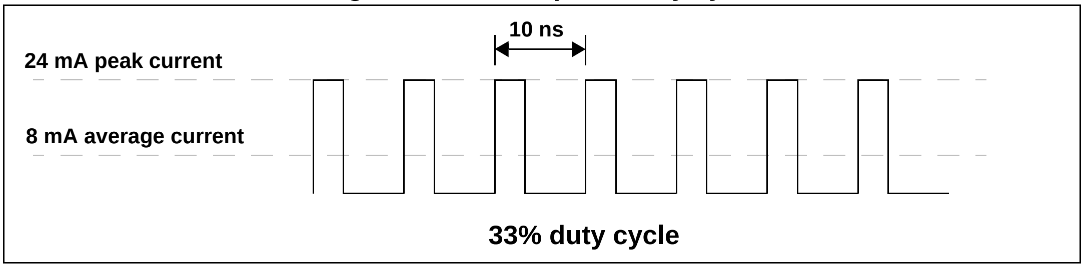
*Note: The VCSEL driver is pulsed at 100 MHz with a 33% duty cycle (see Figure 15.) so average current consumption on AVDD_VCSEL is one third of the peak.*

**Table 12. Current consumption on AVDD and AVDD_VCSEL**

| Power supply(1) | Current | Note |
|---|---|---|
| AVDD | 14 mA | Average during active ranging |
| AVDD_VCSEL | 8 mA(2) | Average during active ranging (33% duty cycle). |

1. Normally, both supplies will be driven from a common source giving a peak instantaneous current demand of 38 mA.
2. Peak emitter current during ranging is 24 mA.

**Figure 15. VCSEL pulse duty cycle**



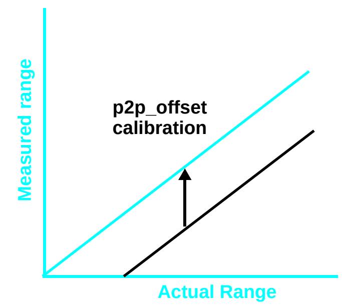
## 2.12 Other system considerations

This section describes part-to-part range offset and system cross-talk. In addition, a procedure for cross-talk calibration is given.

### 2.12.1 Part-to-part range offset

The VL6180X is factory calibrated to produce an absolute linear range output as shown in *Figure 16*. The part-to-part range offset is calibrated during manufacture and stored in `SYSRANGE__PART_TO_PART_RANGE_OFFSET{0x24}` (two's complement). `RESULT__RANGE_RAW{0x64}` reports the range with the part-to-part offset already applied.

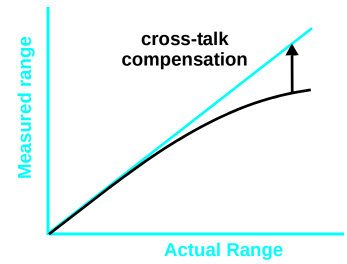
**Figure 16. Part-to-part range offset**




### 2.12.2 Cross-talk

Cross-talk is defined as the signal return from the cover glass. The magnitude of the cross-talk depends on the type of glass, air gap and filter material. Cross-talk results in a range error (see *Figure 17*) which is proportional to the ratio of the cross-talk to the signal return from the target. The true range is recovered by applying automatic cross-talk compensation.

**Figure 17. Cross-talk compensation**




To enable cross-talk compensation it is necessary to write the calibrated cross-talk value to `SYSRANGE__CROSSTALK_COMPENSATION_RATE{0x1E}` in 9.7 format. A cross-talk calibration procedure is described in *Section 2.12.4*.

### 2.12.3 Offset calibration procedure

Complete steps 1-3 to see if part-to-part offset calibration is required.

1. Position a white target (88% reflectance[g]) at a distance of 50 mm from the top of the cover glass.
2. Perform a minimum of 10 range measurements and compute the average range (from `RESULT__RANGE_VAL{0x62}`).
3. If the average range is within the 50 ± 3 mm, offset calibration is not required. Otherwise, complete this calibration procedure.
4. Set `SYSRANGE__PART_TO_PART_RANGE_OFFSET{0x24}` = 0.
5. Perform a minimum of 10 range measurements and compute the average range (from `RESULT__RANGE_VAL{0x62}`).
6. Calculate the part-to-part offset as follows:

$$\text{part-to-part offset} = 50\text{ mm} - \text{average range}$$

7. Write the part-to-part offset result (in two's complement notation) to `SYSRANGE__PART_TO_PART_RANGE_OFFSET`.

### 2.12.4 Cross-talk calibration procedure

This section describes a procedure for calibrating system cross-talk.

1. Perform offset calibration if required (see Section 2.12.3).

*Note: If the offset is incorrectly calibrated, cross-talk calibration will be inaccurate.*

2. Position a black target (3% reflectance[h]) at a distance of 100 mm from the top of the cover glass.
3. Ensure `SYSRANGE__CROSSTALK_COMPENSATION_RATE{0x1E} = 0`.
4. Perform a minimum of 10 range measurements and compute the average return rate (from `RESULT__RANGE_RETURN_RATE{0x66}`) and the average range (from `RESULT__RANGE_VAL{0x62}`).
5. Calculate the cross-talk factor as follows:

$$\text{cross-talk (in Mcps)} = \text{average return rate} \times \left(1 - \frac{\text{average range}}{100\text{ mm}}\right)$$

6. Write the cross-talk result in 9.7 format to `SYSRANGE__CROSSTALK_COMPENSATION_RATE`.

   For example, cross-talk = 0.4 Mcps => 0.4 x 128 = 51.2. Write 51 to `SYSRANGE__CROSSTALK_COMPENSATION_RATE`.

*Note: Cross-talk compensation is only applied to targets above 20 mm. This is to ensure that cross-talk correction is not applied to near targets where the signal rate is decreasing. The cross-talk height qualifier is defined in register `SYSRANGE__CROSSTALK_VALID_HEIGHT{0x21}`. The default is 20 mm.*

g. Target reflectance should be high but absolute value is not critical.

h. Target reflectance should be low but absolute value is not critical.

### 2.12.5 Cross-talk limit

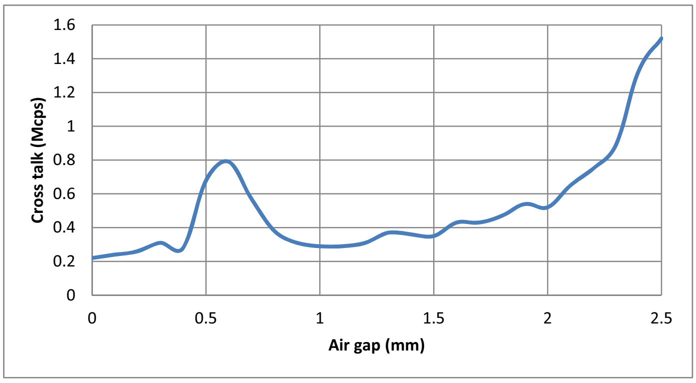
A practical limit for cross-talk is < 3.0 Mcps. This is based on two factors:

1. The return rate for a 3% reflective target at 100 mm without glass is typically around 1.5 Mcps. If glass is added with a cross-talk of 3.0 Mcps, the resultant return rate will be 4.5 Mcps. This results in a cross-talk correction factor of x3 so for a 100 mm target the raw range will be in the region of 30 mm. To ensure the `SYSRANGE__CROSSTALK_VALID_HEIGHT` restriction is not breached, the minimum raw range allowing for noise margin is around 30 mm.
2. A cross-talk correction factor of x3 also means that any range noise will be multiplied by 3 so noise also becomes a limiting factor.

### 2.12.6 Cross-talk vs air gap

*Figure 18* shows the typical cross-talk vs air gap using ST cover glass with oval aperture. Above 2.5 mm, the cross-talk rises rapidly.

**Figure 18. Cross-talk vs air gap**



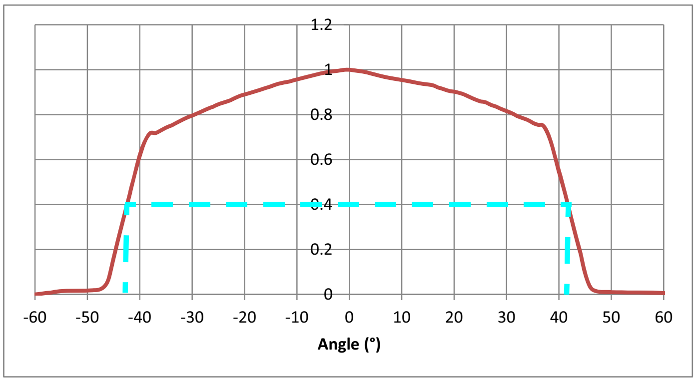
## 2.13 Ambient light sensor (ALS)

The VL6180X contains an ambient light sensor capable of measuring the ambient light level over a wide dynamic range. This section describes the main features of the ALS. The ALS performance specification can be found in *Section 4.2*.

### 2.13.1 Field of view

*Figure 19* shows the ALS field of view which is typically 42 degrees (half angle, 40% of peak) in both X and Y.

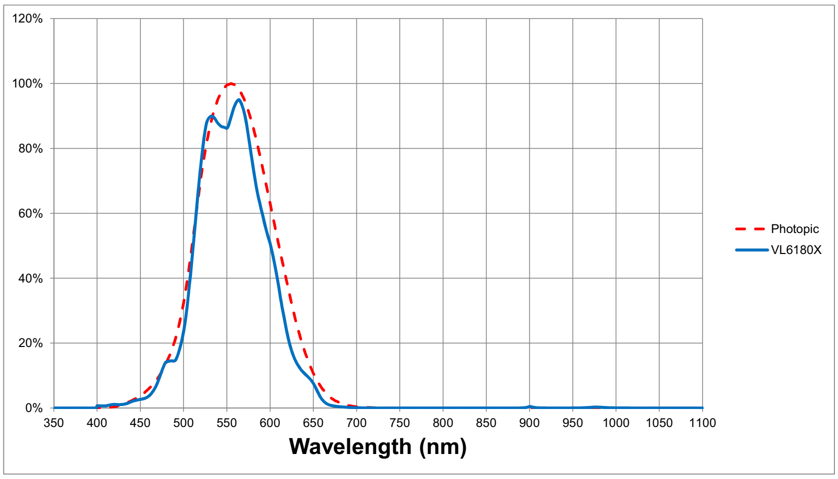
**Figure 19. ALS angular response**




### 2.13.2 Spectral response

The spectral response of the ALS compared to photopic response is shown in *Figure 20*.

**Figure 20. ALS spectral response**



### 2.13.3 ALS dynamic range

*Table 13* shows the range of measurable light at all gains both with and without glass. In most applications operating at a single gain setting should be possible.

**Table 13. ALS dynamic range(1)**

| Analogue gain setting | Dynamic range (no glass) | | Dynamic range (10% transmissive glass) | |
|---|---|---|---|---|
| | Min. (Lux)(2) | Max. (Lux) | Minimum (Lux) | Maximum (Lux) |
| 1 | 3.20 | 20800 | 32.0 | >100,000 |
| 1.25 | 2.56 | 16640 | 25.6 | >100,000 |
| 1.67 | 1.93 | 12530 | 19.3 | >100,000 |
| 2.5 | 1.28 | 8320 | 12.8 | 83,200 |
| 5 | 0.64 | 4160 | 6.4 | 41,600 |
| 10 | 0.32 | 2080 | 3.2 | 20,800 |
| 20 | 0.16 | 1040 | 1.6 | 10,400 |
| 40 | 0.08 | 520 | 0.8 | 5,200 |

1. ALS lux resolution = 0.32 lux/count
2. Minimum of 10 counts

### 2.13.4 ALS count to lux conversion

The output from the ambient light sensor is a 16-bit register, `RESULT__ALS_VAL{0x50}`. The count output is proportional to the light level and can be converted into lux using the following equation:

$$\text{Light level (in lux)} = \text{ALS lux resolution} \times \frac{\text{RESULT\_\_ALS\_VAL}}{\text{Analog gain}} \times \frac{100\text{ ms}}{\text{ALS integration time}}$$

The factory calibrated ALS lux resolution is 0.32 lux/count for an analog gain of 1 (calibrated without glass). The ALS lux resolution will require re-calibration in the final system where cover glass is used. This can be done by recording the count output with and without glass under the same conditions and multiplying the ALS lux resolution by the ratio of the two counts as follows:

$$\text{ALS lux resolution (with glass)} = \frac{\text{RESULT\_\_ALS\_VAL (without glass)}}{\text{RESULT\_\_ALS\_VAL (with glass)}} \times \text{ALS lux resolution (without glass)}$$

### 2.13.5 Integration period

The integration period is the time over which a single ALS measurement is made. The default integration period is 100ms. Integration times in the range 50-100 ms are recommended to reduce impact of light flicker from artificial lighting.

### 2.13.6 ALS gain selection

Eight analog gain settings are available which can be selected manually depending on the range and resolution required. *Table 14.*shows the actual characterized gains versus the design targets. If a gain setting other than gain 20 is used, marginally greater accuracy can be achieved by using the actual gain values in the light level equation in *Section 2.13.4* when calculating the lux light level.

**Table 14. Actual gain values**

| Register setting {0x3F} | Analog gain setting | Actual gain values |
|---|---|---|
| 0x46 | 1 | 1.01 |
| 0x45 | 1.25 | 1.28 |
| 0x44 | 1.67 | 1.72 |
| 0x43 | 2.5 | 2.60 |
| 0x42 | 5 | 5.21 |
| 0x41 | 10 | 10.32 |
| 0x40 | 20 | 20 |
| 0x47 | 40 | 40 |

*Note: The upper nibble of SYSALS__ANALOGUE_GAIN should always be set to 0x4.*

# 3 Electrical characteristics

## 3.1 Absolute maximum ratings

**Table 15. Absolute maximum ratings**

| Parameter | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|
| AVDD | -0.5 | - | 3.6 | V |
| AVDD_VCSEL | -0.5 | - | 3.6 | V |
| SCL, SDA, GPIO0 and GPIO1 | -0.5 | - | 3.6 | V |

*Note: Stresses above those listed in Table 15. may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of the specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

## 3.2 Normal operating conditions

**Table 16. Normal operating conditions**

| Parameter | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|
| Voltage (AVDD and AVDD_VCSEL) | | | | |
| Voltage (optimum operating) | 2.7 | 2.8 | 2.9 | V |
| Voltage (functional operating) | 2.6 | 2.8 | 3.0 | V |
| Temperature | | | | |
| Temperature (optimum operating) | -10 | | +60 | °C |
| Temperature (functional operating) | -20 | - | +70 | °C |
| Temperature (test) | +21 | - | +25 | °C |
| Temperature (storage) | -40 | - | +85 | °C |

## 3.3 Current consumption

**Table 17. Current consumption(1)**

| Parameter | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|
| Hardware Standby | - | - | 1 | µA |
| Software Standby | - | - | 1 | µA |
| ALS operation | - | 300 | 350 | µA |

1. Measured at room temperature (23°C)

## 3.4 Electrical characteristics

**Table 18. Digital I/O electrical characteristics**

| Symbol | Parameter | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| **CMOS digital I/O (SDA, SCL, GPIO0 and GPIO1)** | | | | | |
| $V_{IL}$ | Low level input voltage | -0.5 | - | 0.6 | V |
| $V_{IH}$ | High level input voltage | 1.12 | - | AVDD+0.5 | V |
| $V_{OL}$ | Low level output voltage (8mA load) | - | - | 0.4 | V |
| $V_{OH}$ | High level output voltage (8mA load) | AVDD-0.4 | - | - | V |
| $I_{IL}$ | Low level input current | - | - | -10 | µA |
| $I_{IH}$ | High level input current | - | - | 10 | µA |

# 4 Performance specification

## 4.1 Proximity ranging (0 to 100mm)

The following table specifies ranging performance up to 100mm. Ranging beyond 100mm is possible with certain target reflectances and ambient conditions but not guaranteed. These results are derived from characterization of both typical and corner samples (representative of worst case process conditions).

Unless specified otherwise, all results were performed at room temperature (23°C), nominal voltage (2.8V) and in the dark. Results are based on the average of 100 measurements for a 17% reflective target @ 50mm.

**Table 19. Ranging specification**

| Parameter | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|
| Noise(1) | - | - | 2.0 | mm |
| Range offset error(2) | - | - | 13 | mm |
| Temperature dependent drift (3) | - | 9 | 15 | mm |
| Voltage dependent drift (4) | - | 3 | 5 | mm |
| Convergence time (5) | - | - | 15 | ms |

1. Maximum standard deviation of 100 measurements
2. Maximum offset drift after 3 reflow cycles. This error can be removed by re-calibration in the final system
3. Tested over optimum operating temperature range (see *Table 16.: Normal operating conditions*)
4. Tested over optimum operating voltage range (see *Table 16.: Normal operating conditions*)
5. Based on a 3% reflective target @ 100 mm

### 4.1.1 Max range vs. ambient light level

The data shown in this section is worst case data **for reference only.**

*Table 20* shows the worst case maximum range achievable under different ambient light conditions

.

**Table 20. Worst case max range vs. ambient 0 to 100mm(1)(2)**

| Target reflectance | In the dark(3) | Worst case indoor light (1 kLux diffuse halogen) | High ambient light (5 kLux diffuse halogen) | Unit |
|---|---|---|---|---|
| 3% | > 100 | > 80 | > 40 | mm |
| 5% | > 100 | > 90 | > 45 | mm |
| 17% | > 100 | > 100 | > 60 | mm |
| 88% | > 100 | > 100 | > 70 | mm |

1. Tested in an integrating sphere (repeatable lab test, not representative of real world ambient light) at 1 kLux and 5 kLux (halogen light source) using 80 x 80 mm targets. Due to high IR content, 5 kLux halogen light approximates to 10 kLux to 15 kLux natural sunlight.
2. SNR limit of 0.1 applied. Note: maximum range could be increased by reducing the SNR limit to 0.06
3. Also applicable to lighting conditions with low IR content e.g typical office fluorescent lighting

## 4.2 ALS performance

The following table specifies ALS performance. These results are derived from characterization of typical samples (without cover glass). Unless specified otherwise, all tests were performed at room temperature (23°C), nominal voltage (2.8V) and using a halogen light source.

**Table 21. ALS performance**

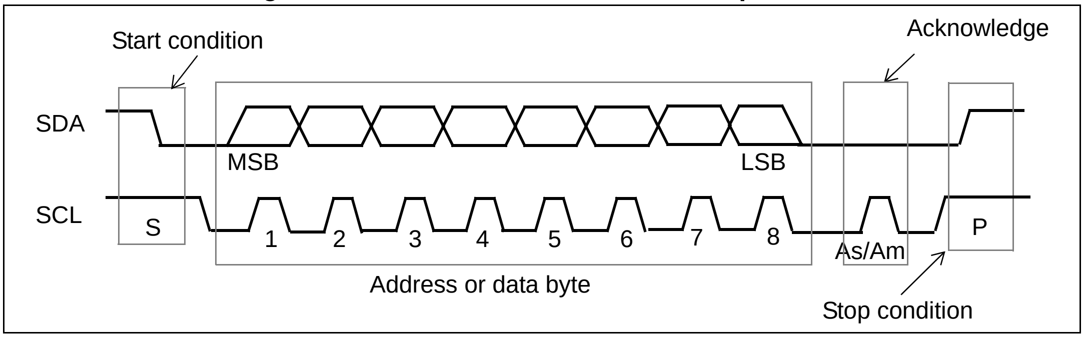
| Parameter | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|
| ALS sensitivity(1) | 0.28 | 0.32 | 0.36 | Lux/count |
| Angular response(2) | - | 42 | - | degrees |
| Spectral response | - | photopic | - | - |
| Dynamic Range(3) | 0.002 | - | 20971 | Lux |
| Linearity error (1 to 300 lux)(4) | - | - | 5 | % |
| Linearity error (300 to 7500 lux)(4) | - | - | 10 | % |
| Gain error (@ gain 20) | - | - | 1 | % |
| Gain error (gains 1 to 10) | - | - | 7 | % |

1. 535nm LED @ 1 kLux. Measured @ gain 20.
2. Half angle. 40% transmission.
3. Minimum of one count at gain 40 and 400 ms ALS integration time.
4. Test conditions: -10°C to +60°C; analog gains 1 to 20

# 5 I²C control interface

The VL6180X is controlled over an I²C interface. The default I²C address is 0x29 (7-bit). This section describes the I²C protocol.

**Figure 21. Serial interface data transfer protocol**




Information is packed in 8-bit packets (bytes) always followed by an acknowledge bit, As for sensor acknowledge and Am for master acknowledge. The internal data is produced by sampling SDA at a rising edge of SCL. The external data must be stable during the high period of SCL. The exceptions to this are start (S) or stop (P) conditions when SDA falls or rises respectively, while SCL is high.

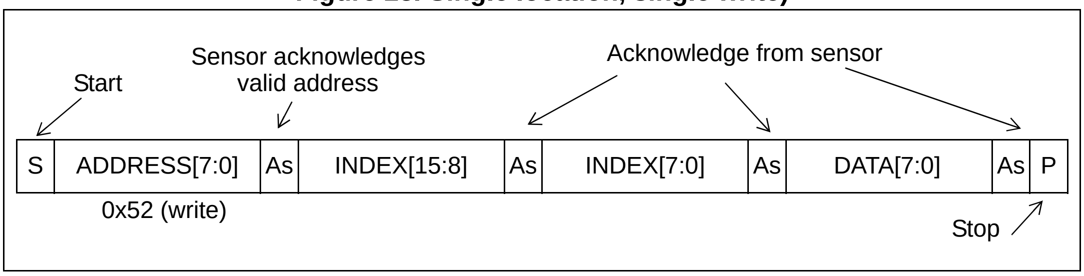
A message contains a series of bytes preceded by a start condition and followed by either a stop or repeated start (another start condition but without a preceding stop condition) followed by another message. The first byte contains the device address (0x52) and also specifies the data direction. If the least significant bit is low (0x52) the message is a master write to the slave. If the lsb is set (0x53) then the message is a master read from the slave.

**Figure 22. I²C device address**

| MSBit | | | | | | | LSBit |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | R/$\overline{W}$ |

All serial interface communications with the sensor must begin with a start condition. The sensor acknowledges the receipt of a valid address by driving the SDA wire low. The state of the read/write bit (lsb of the address byte) is stored and the next byte of data, sampled from SDA, can be interpreted. During a write sequence the second and third bytes received provide a 16-bit index which points to one of the internal 8-bit registers.

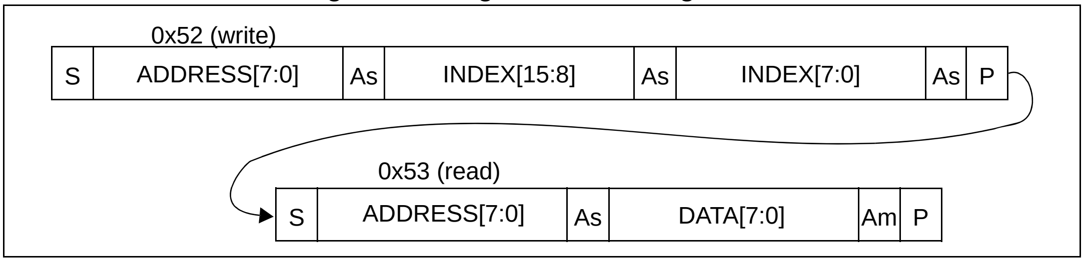
**Figure 23. Single location, single write)**



As data is received by the slave it is written bit by bit to a serial/parallel register. After each data byte has been received by the slave, an acknowledge is generated, the data is then stored in the internal register addressed by the current index.

During a read message, the contents of the register addressed by the current index is read out in the byte following the device address byte. The contents of this register are parallel loaded into the serial/parallel register and clocked out of the device by the falling edge of SCL.

**Figure 24. Single location, single read**




At the end of each byte, in both read and write message sequences, an acknowledge is issued by the receiving device (that is, the sensor for a write and the master for a read).

A message can only be terminated by the bus master, either by issuing a stop condition or by a negative acknowledge (that is, **not** pulling the SDA line low) after reading a complete byte during a read operation.

The interface also supports auto-increment indexing. After the first data byte has been transferred, the index is automatically incremented by 1. The master can therefore send data bytes continuously to the slave until the slave fails to provide an acknowledge or the master terminates the write communication with a stop condition. If the auto-increment feature is used the master does **not** have to send address indexes to accompany the data bytes.

**Figure 25. Multiple location write**

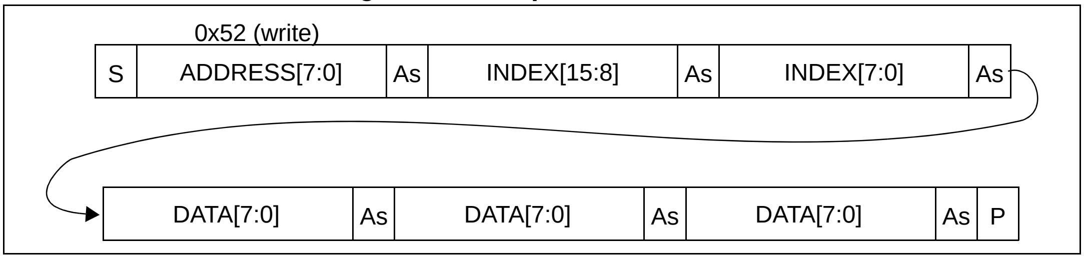
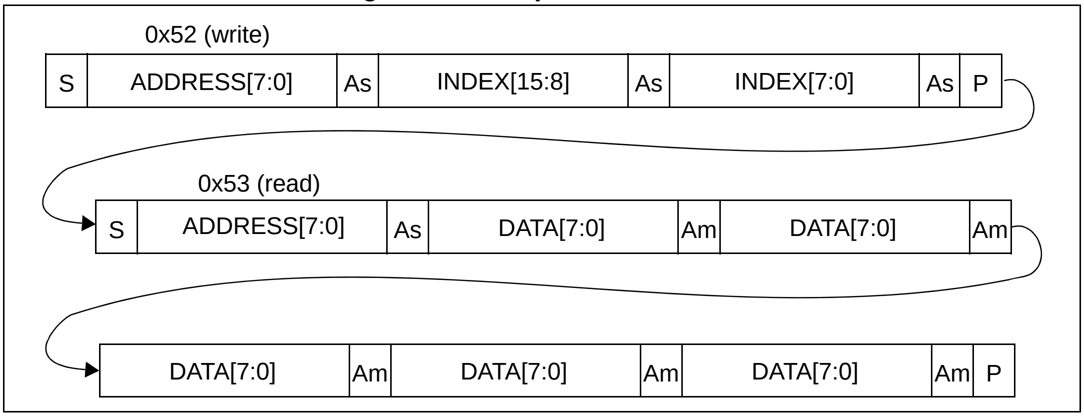


**Figure 26. Multiple location read**



# 6 Device registers

This section describes in detail all user accessible device registers. Registers are grouped by function as shown in *Table 22.* to make them easier to read but also to simplify multi-byte read/write $I^2C$ accesses (burst mode). More details in *Section 5*. Reset values are given for each register which denotes the register value in software standby.

**Table 22. Register groups**

| Register group | Address range |
|---|---|
| IDENTIFICATION | 0x00 - 0x0F |
| SYSTEM SETUP | 0x10 - 0x17 |
| RANGE SETUP | 0x18 - 0x37 |
| ALS SETUP | 0x38 - 0x40 |
| RESULTS | 0x4D - 0x80 |

Note that registers can be 8-,16- or 32-bit. Multi-byte registers are always addressed in ascending order with MSB first as shown in *Table 23.*

**Table 23. 32-bit register example**

| Register address | Byte |
|---|---|
| Address | MSB |
| Address + 1 | .. |
| Address + 2 | .. |
| Address + 3 | LSB |

## 6.1 Register encoding formats

Some registers are encoded to allow rational numbers to be expressed efficiently. *Table 24* gives an explanation of 9.7 and 4.4 encoding formats.

**Table 24. 9.7 and 4.4 register formats**

| Format | Description |
|---|---|
| 4.4 | 8 bits = 4 integer bits + 4 fractional bits (stored as 1 byte)<br>Encoding example: the value 4.2 is **multiplied** by 16 ($2^4$) rounded and stored as 67 decimal.<br>Decoding example: 67 is **divided** by 16 = 4.19. |
| 9.7 | 16 bits = 9 integer bits + 7 fractional bits (stored over 2 bytes)<br>Encoding example: the value 4.2 is **multiplied** by 128 ($2^7$) rounded and stored as 537 decimal.<br>Decoding example: 537 is **divided** by 128 = 4.19. |

**Table 25. Register summary**

| Offset | Register name | Reference |
|---|---|---|
| 0x000 | IDENTIFICATION__MODEL_ID | *Section 6.2.1 on page 43* |
| 0x001 | IDENTIFICATION__MODEL_REV_MAJOR | *Section 6.2.2 on page 43* |
| 0x002 | IDENTIFICATION__MODEL_REV_MINOR | *Section 6.2.3 on page 43* |
| 0x003 | IDENTIFICATION__MODULE_REV_MAJOR | *Section 6.2.4 on page 44* |
| 0x004 | IDENTIFICATION__MODULE_REV_MINOR | *Section 6.2.5 on page 44* |
| 0x006 | IDENTIFICATION__DATE_HI | *Section 6.2.6 on page 44* |
| 0x007 | IDENTIFICATION__DATE_LO | *Section 6.2.7 on page 45* |
| 0x008:0x009 | IDENTIFICATION__TIME | *Section 6.2.8 on page 45* |
| 0x010 | SYSTEM__MODE_GPIO0 | *Section 6.2.9 on page 46* |
| 0x011 | SYSTEM__MODE_GPIO1 | *Section 6.2.10 on page 47* |
| 0x012 | SYSTEM__HISTORY_CTRL | *Section 6.2.11 on page 48* |
| 0x014 | SYSTEM__INTERRUPT_CONFIG_GPIO | *Section 6.2.12 on page 49* |
| 0x015 | SYSTEM__INTERRUPT_CLEAR | *Section 6.2.13 on page 49* |
| 0x016 | SYSTEM__FRESH_OUT_OF_RESET | *Section 6.2.14 on page 50* |
| 0x017 | SYSTEM__GROUPED_PARAMETER_HOLD | *Section 6.2.15 on page 50* |
| 0x018 | SYSRANGE__START | *Section 6.2.16 on page 51* |
| 0x019 | SYSRANGE__THRESH_HIGH | *Section 6.2.17 on page 51* |
| 0x01A | SYSRANGE__THRESH_LOW | *Section 6.2.18 on page 52* |
| 0x01B | SYSRANGE__INTERMEASUREMENT_PERIOD | *Section 6.2.19 on page 52* |
| 0x01C | SYSRANGE__MAX_CONVERGENCE_TIME | *Section 6.2.20 on page 52* |
| 0x01E | SYSRANGE__CROSSTALK_COMPENSATION_RATE | *Section 6.2.21 on page 53* |
| 0x021 | SYSRANGE__CROSSTALK_VALID_HEIGHT | *Section 6.2.22 on page 53* |
| 0x022 | SYSRANGE__EARLY_CONVERGENCE_ESTIMATE | *Section 6.2.23 on page 53* |
| 0x024 | SYSRANGE__PART_TO_PART_RANGE_OFFSET | *Section 6.2.24 on page 54* |
| 0x025 | SYSRANGE__RANGE_IGNORE_VALID_HEIGHT | *Section 6.2.25 on page 54* |
| 0x026 | SYSRANGE__RANGE_IGNORE_THRESHOLD | *Section 6.2.26 on page 54* |
| 0x02C | SYSRANGE__MAX_AMBIENT_LEVEL_MULT | *Section 6.2.27 on page 55* |
| 0x02D | SYSRANGE__RANGE_CHECK_ENABLES | *Section 6.2.27 on page 55* |
| 0x02E | SYSRANGE__VHV_RECALIBRATE | *Section 6.2.29 on page 56* |
| 0x031 | SYSRANGE__VHV_REPEAT_RATE | *Section 6.2.30 on page 56* |
| 0x038 | SYSALS__START | *Section 6.2.31 on page 57* |
| 0x03A | SYSALS__THRESH_HIGH | *Section 6.2.32 on page 57* |
| 0x03C | SYSALS__THRESH_LOW | *Section 6.2.33 on page 58* |

**Table 25. Register summary (continued)**

| Offset | Register name | Reference |
|---|---|---|
| 0x03E | SYSALS__INTERMEASUREMENT_PERIOD | Section 6.2.34 on page 58 |
| 0x03F | SYSALS__ANALOGUE_GAIN | Section 6.2.35 on page 59 |
| 0x040 | SYSALS__INTEGRATION_PERIOD | Section 6.2.36 on page 59 |
| 0x04D | RESULT__RANGE_STATUS | Section 6.2.37 on page 60 |
| 0x04E | RESULT__ALS_STATUS | Section 6.2.38 on page 61 |
| 0x04F | RESULT__INTERRUPT_STATUS_GPIO | Section 6.2.39 on page 62 |
| 0x050 | RESULT__ALS_VAL | Section 6.2.40 on page 62 |
| 0x052:0x060 (0x2) | RESULT__HISTORY_BUFFER_x | Section 6.2.41 on page 63 |
| 0x062 | RESULT__RANGE_VAL | Section 6.2.42 on page 64 |
| 0x064 | RESULT__RANGE_RAW | Section 6.2.43 on page 64 |
| 0x066 | RESULT__RANGE_RETURN_RATE | Section 6.2.44 on page 64 |
| 0x068 | RESULT__RANGE_REFERENCE_RATE | Section 6.2.45 on page 65 |
| 0x06C | RESULT__RANGE_RETURN_SIGNAL_COUNT | Section 6.2.46 on page 65 |
| 0x070 | RESULT__RANGE_REFERENCE_SIGNAL_COUNT | Section 6.2.47 on page 66 |
| 0x074 | RESULT__RANGE_RETURN_AMB_COUNT | Section 6.2.48 on page 66 |
| 0x078 | RESULT__RANGE_REFERENCE_AMB_COUNT | Section 6.2.49 on page 66 |
| 0x07C | RESULT__RANGE_RETURN_CONV_TIME | Section 6.2.50 on page 67 |
| 0x080 | RESULT__RANGE_REFERENCE_CONV_TIME | Section 6.2.51 on page 67 |
| 0x10A | READOUT__AVERAGING_SAMPLE_PERIOD | Section 6.2.52 on page 67 |
| 0x119 | FIRMWARE__BOOTUP | Section 6.2.52 on page 67 |
| 0x120 | FIRMWARE__RESULT_SCALER | Section 6.2.53 on page 68 |
| 0x212 | I2C_SLAVE__DEVICE_ADDRESS | Section 6.2.55 on page 68 |
| 0x2A3 | INTERLEAVED_MODE__ENABLE | Section 6.2.56 on page 69 |

## 6.2 Register descriptions

### 6.2.1 IDENTIFICATION__MODEL_ID

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| identification__model_id | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x000

**Type:** R/W

**Reset:** 0xB4

**Description:**

| | |
|---|---|
| [7:0] | identification__model_id: Device model identification number. 0xB4 = VL6180X |

### 6.2.2 IDENTIFICATION__MODEL_REV_MAJOR

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | identification__model_rev_major | | |
| R | | | | | R/W | | |

**Address:** 0x001

**Type:** R/W

**Reset:** 0x1, register default overwritten at boot-up by NVM contents.

**Description:**

| | |
|---|---|
| [2:0] | identification__model_rev_major: Revision identifier of the Device for major change. |

### 6.2.3 IDENTIFICATION__MODEL_REV_MINOR

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | identification__model_rev_minor | | |
| R | | | | | R/W | | |

**Address:** 0x002

**Type:** R/W

**Reset:** 0x3, register default overwritten at boot-up by NVM contents.

**Description:**

| | |
|---|---|
| [2:0] | identification__model_rev_minor: Revision identifier of the Device for minor change.<br>IDENTIFICATION__MODEL_REV_MINOR = 3 for latest ROM revision |

### 6.2.4 IDENTIFICATION__MODULE_REV_MAJOR

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | identification__module_rev_major | | |
| R | | | | | R/W | | |

**Address:** 0x003

**Type:** R/W

**Reset:** 0x1, register default overwritten at boot-up by NVM contents.

**Description:**

| | |
|---|---|
| [2:0] | identification__module_rev_major: Revision identifier of the Module Package for major change. Used to store NVM content version. Contact ST for current information. |

### 6.2.5 IDENTIFICATION__MODULE_REV_MINOR

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | identification__module_rev_minor | | |
| R | | | | | R/W | | |

**Address:** 0x004

**Type:** R/W

**Reset:** 0x2, register default overwritten at boot-up by NVM contents.

**Description:**

| | |
|---|---|
| [2:0] | identification__module_rev_minor: Revision identifier of the Module Package for minor change. Used to store NVM content version. Contact ST for current information. |

### 6.2.6 IDENTIFICATION__DATE_HI

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| identification__year | | | | identification__month | | | |
| R/W | | | | R/W | | | |

**Address:** 0x006

**Type:** R/W

**Reset:** 0xYY, register default overwritten at boot-up by NVM contents.

**Description:** Part of the register set that can be used to uniquely identify a module.

| | |
|---|---|
| [7:4] | identification__year: Last digit of manufacturing year (bits[3:0]). |
| [3:0] | identification__month: Manufacturing month (bits[3:0]). |

### 6.2.7 IDENTIFICATION__DATE_LO

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| identification__day | | | | | identification__phase | | |
| R/W | | | | | R/W | | |

**Address:** 0x007

**Type:** R/W

**Reset:** 0xYY, register default overwritten at boot-up by NVM contents.

**Description:** Part of the register set that can be used to uniquely identify a module.

| | |
|---|---|
| [7:3] | identification__day: Manufacturing day (bits[4:0]). |
| [2:0] | identification__phase: Manufacturing phase identification (bits[2:0]). |

### 6.2.8 IDENTIFICATION__TIME

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| identification__time | | | | | | | | | | | | | | | |
| R/W | | | | | | | | | | | | | | | |

**Address:** 0x008:0x009

**Type:** R/W

**Reset:** 0xYYYY, register default overwritten at boot-up by NVM contents.

**Description:** Part of the register set that can be used to uniquely identify a module.

| | |
|---|---|
| [15:0] | identification__time: Time since midnight (in seconds) = register_value * 2 |

### 6.2.9 SYSTEM__MODE_GPIO0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | system__gpio0_is_xshutdown | system__gpio0_polarity | system__gpio0_select | | | | RESERVED |
| R | R/W | R/W | R/W | | | | R/W |

**Address:** 0x010

**Type:** R/W

**Reset:** 0x60

**Description:**

| | |
|---|---|
| [6] | system__gpio0_is_xshutdown: Priority mode - when enabled, other bits of the register are ignored. GPIO0 is main XSHUTDOWN input.<br>0: Disabled<br>1: Enabled - GPIO0 is main XSHUTDOWN input. |
| [5] | system__gpio0_polarity: Signal Polarity Selection.<br>0: Active-low<br>1: Active-high |
| [4:1] | system__gpio0_select: Functional configuration options.<br>0000: OFF (Hi-Z)<br>1000: GPIO Interrupt output |
| [0] | Reserved. Write 0. |

### 6.2.10 SYSTEM__MODE_GPIO1

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | system__gpio1_polarity | system__gpio1_select | | | | RESERVED |
| R | | R/W | R/W | | | | R/W |

**Address:** 0x011

**Type:** R/W

**Reset:** 0x20

**Description:**

| | |
|---|---|
| [5] | system__gpio1_polarity: Signal Polarity Selection.<br>0: Active-low<br>1: Active-high |
| [4:1] | system__gpio1_select: Functional configuration options.<br>0000: OFF (Hi-Z)<br>1000: GPIO Interrupt output |
| [0] | Reserved. Write 0. |

### 6.2.11 SYSTEM__HISTORY_CTRL

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | system__history_buffer_clear | system__history_buffer_mode | system__history_buffer_enable |
| R | | | | | R/W | R/W | R/W |

**Address:** 0x012

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [2] | system__history_buffer_clear: User-command to clear history (FW will auto-clear this bit when clear has completed).<br>0: Disabled<br>1: Clear all history buffers |
| [1] | system__history_buffer_mode: Select mode buffer results for:<br>0: Ranging (stores the last 8 ranging values (8-bit)<br>1: ALS (stores the last 8 ALS values (16-bit) |
| [0] | system__history_buffer_enable: Enable History buffering.<br>0: Disabled<br>1: Enabled |

### 6.2.12 SYSTEM__INTERRUPT_CONFIG_GPIO

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | als_int_mode | | | range_int_mode | | |
| R | | R/W | | | R/W | | |

**Address:** 0x014

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [5:3] | als_int_mode: Interrupt mode source for ALS readings:<br>0: Disabled<br>1: Level Low (value < thresh_low)<br>2: Level High (value > thresh_high)<br>3: Out Of Window (value < thresh_low OR value > thresh_high)<br>4: New sample ready |
| [2:0] | range_int_mode: Interrupt mode source for Range readings:<br>0: Disabled<br>1: Level Low (value < thresh_low)<br>2: Level High (value > thresh_high)<br>3: Out Of Window (value < thresh_low OR value > thresh_high)<br>4: New sample ready |

### 6.2.13 SYSTEM__INTERRUPT_CLEAR

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | int_clear_sig | | |
| R | | | | | R/W | | |

**Address:** 0x015

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [2:0] | int_clear_sig: Interrupt clear bits. Writing a 1 to each bit will clear the intended interrupt.<br><br>Bit [0] - Clear Range Int<br>Bit [1] - Clear ALS Int<br>Bit [2] - Clear Error Int. |

### 6.2.14 SYSTEM__FRESH_OUT_OF_RESET

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | fresh_out_of_reset |
| R | | | | | | | R/W |

**Address:** 0x016

**Type:** R/W

**Reset:** 0x1

**Description:**

| | |
|---|---|
| [0] | fresh_out_of_reset: Fresh out of reset bit, default of 1, user can set this to 0 after initial boot and can therefore use this to check for a reset condition |

### 6.2.15 SYSTEM__GROUPED_PARAMETER_HOLD

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | grouped_parameter_hold |
| R | | | | | | | R/W |

**Address:** 0x017

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [0] | grouped_parameter_hold: Flag set over $I^2C$ to indicate that data is being updated<br>0: Data is stable - FW is safe to copy<br>1: Data being updated - FW not safe to copy<br>Usage: set to 0x01 first, write any of the registers listed below, then set to 0x00 so that the settings are used by the firmware at the start of the next measurement.<br>SYSTEM__INTERRUPT_CONFIG_GPIO<br>SYSRANGE__THRESH_HIGH<br>SYSRANGE__THRESH_LOW<br>SYSALS__INTEGRATION_PERIOD<br>SYSALS__ANALOGUE_GAIN<br>SYSALS__THRESH_HIGH<br>SYSALS__THRESH_LOW |

### 6.2.16 SYSRANGE__START

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | sysrange__mode_select | sysrange__startstop |
| R | | | | | | R/W | R/W |

**Address:** 0x018

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [1] | sysrange__mode_select: Device Mode select<br>0: Ranging Mode Single-Shot<br>1: Ranging Mode Continuous |
| [0] | sysrange__startstop: StartStop trigger based on current mode and system configuration of device_ready. FW clears register automatically.<br>Setting this bit to 1 in single-shot mode starts a single measurement.<br>Setting this bit to 1 in continuous mode will either start continuous operation (if stopped) or halt continuous operation (if started).<br>This bit is auto-cleared in both modes of operation. |

### 6.2.17 SYSRANGE__THRESH_HIGH

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| sysrange__thresh_high | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x019

**Type:** R/W

**Reset:** 0xFF

**Description:**

| | |
|---|---|
| [7:0] | sysrange__thresh_high: High Threshold value for ranging comparison. Range 0-255mm. |

### 6.2.18 SYSRANGE__THRESH_LOW

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| sysrange__thresh_low | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x01A

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [7:0] | sysrange__thresh_low: Low Threshold value for ranging comparison. Range 0-255mm. |

### 6.2.19 SYSRANGE__INTERMEASUREMENT_PERIOD

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| sysrange__intermeasurement_period | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x01B

**Type:** R/W

**Reset:** 0xFF

**Description:**

| | |
|---|---|
| [7:0] | sysrange__intermeasurement_period: Time delay between measurements in Ranging continuous mode. Range 0-254 (0 = 10ms). Step size = 10ms. |

### 6.2.20 SYSRANGE__MAX_CONVERGENCE_TIME

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | sysrange__max_convergence_time | | | | | |
| R | | R/W | | | | | |

**Address:** 0x01C

**Type:** R/W

**Reset:** 0x31

**Description:**

| | |
|---|---|
| [5:0] | sysrange__max_convergence_time: Maximum time to run measurement in Ranging modes. Range 1 - 63 ms (1 code = 1 ms); Measurement aborted when limit reached to aid power reduction. For example, 0x01 = 1ms, 0x0a = 10ms.<br>Note: Effective max_convergence_time depends on readout_averaging_sample_period setting. |

### 6.2.21 SYSRANGE__CROSSTALK_COMPENSATION_RATE

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| sysrange__crosstalk_compensation_rate | | | | | | | | | | | | | | | |
| R/W | | | | | | | | | | | | | | | |

**Address:** 0x01E

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [15:0] | sysrange__crosstalk_compensation_rate: User-controlled crosstalk compensation in Mcps (9.7 format). |

### 6.2.22 SYSRANGE__CROSSTALK_VALID_HEIGHT

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| sysrange__crosstalk_valid_height | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x021

**Type:** R/W

**Reset:** 0x14

**Description:**

| | |
|---|---|
| [7:0] | sysrange__crosstalk_valid_height: Minimum range value in mm to qualify for crosstalk compensation. |

### 6.2.23 SYSRANGE__EARLY_CONVERGENCE_ESTIMATE

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| sysrange__early_convergence_estimate | | | | | | | | | | | | | | | |
| R/W | | | | | | | | | | | | | | | |

**Address:** 0x022

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [15:0] | FW carries out an estimate of convergence rate 0.5ms into each new range measurement. If convergence rate is below user input value, the operation aborts to save power.<br>Note: This register must be configured otherwise ECE should be disabled via `SYSRANGE__RANGE_CHECK_ENABLES`. |

### 6.2.24 SYSRANGE__PART_TO_PART_RANGE_OFFSET

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| sysrange__part_to_part_range_offset | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x024

**Type:** R/W

**Reset:** 0xYY, register default overwritten at boot-up by NVM contents.

**Description:**

| | |
|---|---|
| [7:0] | sysrange__part_to_part_range_offset: 2s complement format. |

### 6.2.25 SYSRANGE__RANGE_IGNORE_VALID_HEIGHT

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| sysrange__range_ignore_valid_height | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x025

**Type:** R/W

**Reset:** 0x0, register default overwritten at boot-up by NVM contents.

**Description:**

| | |
|---|---|
| [7:0] | sysrange__range_ignore_valid_height: Range below which ignore threshold is applied. Aim is to ignore the cover glass i.e. low signal rate at near distance. Should not be applied to low reflectance target at far distance. Range in mm.<br>Note: It is recommended to set this register to 255 if the range ignore feature is used. |

### 6.2.26 SYSRANGE__RANGE_IGNORE_THRESHOLD

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| sysrange__range_ignore_threshold | | | | | | | | | | | | | | | |
| R/W | | | | | | | | | | | | | | | |

**Address:** 0x026

**Type:** R/W

**Reset:** 0xYY

**Description:**

| | |
|---|---|
| [15:0] | sysrange__range_ignore_threshold: User configurable min threshold signal return rate. Used to filter out ranging due to cover glass when there is no target above the device. Mcps 9.7 format.<br>Note: Register must be initialized if this feature is used. |

### 6.2.27 SYSRANGE__MAX_AMBIENT_LEVEL_MULT

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| sysrange__max_ambient_level_mult | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x02C

**Type:** R/W

**Reset:** 0xA0, register default overwritten at boot-up by NVM contents.

**Description:**

| | |
|---|---|
| [7:0] | sysrange__max_ambient_level_mult: User input value to multiply return_signal_count for AMB:signal ratio check. If (amb counts * 6) > return_signal_count * mult then abandon measurement due to high ambient (4.4 format). |

### 6.2.28 SYSRANGE__RANGE_CHECK_ENABLES

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | sysrange__signal_to_noise_enable | 0 | 0 | sysrange__range_ignore_enable | sysrange__early_convergence_enable |
| R | | | R/W | R/W | R | R/W | R/W |

**Address:** 0x02D

**Type:** R/W

**Reset:** 0x11, register default overwritten at boot-up by NVM contents.

**Description:**

| | |
|---|---|
| [4] | sysrange__signal_to_noise_enable: Measurement enable/disable |
| [1] | sysrange__range_ignore_enable: Measurement enable/disable |
| [0] | sysrange__early_convergence_enable: Measurement enable/disable |

### 6.2.29 SYSRANGE__VHV_RECALIBRATE

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | sysrange__vhv_status | sysrange__vhv_recalibrate |
| R | | | | | | R/W | R/W |

**Address:** 0x02E

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [1] | sysrange__vhv_status: FW controlled status bit showing when FW has completed auto-vhv process.<br>0: FW has finished autoVHV operation<br>1: During autoVHV operation |
| [0] | sysrange__vhv_recalibrate: User-Controlled enable bit to force FW to carry out recalibration of the VHV setting for sensor array. FW clears bit after operation carried out.<br>0: Disabled<br>1: Manual trigger for VHV recalibration. Can only be called when ALS and ranging are in STOP mode |

### 6.2.30 SYSRANGE__VHV_REPEAT_RATE

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| sysrange__vhv_repeate_rate | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x031

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [7:0] | sysrange__vhv_repeat_rate: User entered repeat rate of auto VHV task (0 = off, 255 = after every 255 measurements) |

### 6.2.31 SYSALS__START

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | sysals__mode_select | sysals__startstop |
| R | | | | | | R/W | R/W |

**Address:** 0x038

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [1] | sysals__mode_select: Device Mode select<br>0: ALS Mode Single-Shot<br>1: ALS Mode Continuous |
| [0] | sysals__startstop: Start/Stop trigger based on current mode and system configuration of device_ready. FW clears register automatically.<br>Setting this bit to 1 in single-shot mode starts a single measurement.<br>Setting this bit to 1 in continuous mode will either start continuous operation (if stopped) or halt continuous operation (if started).<br>This bit is auto-cleared in both modes of operation.<br>See *6.2.56: INTERLEAVED_MODE__ENABLE* for combined ALS and Range operation. |

### 6.2.32 SYSALS__THRESH_HIGH

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| sysals__thresh_high | | | | | | | | | | | | | | | |
| R/W | | | | | | | | | | | | | | | |

**Address:** 0x03A

**Type:** R/W

**Reset:** 0xFFFF

**Description:**

| | |
|---|---|
| [15:0] | sysals__thresh_high: High Threshold value for ALS comparison. Range 0-65535 codes. |

### 6.2.33 SYSALS__THRESH_LOW

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| sysals__thresh_low | | | | | | | | | | | | | | | |
| R/W | | | | | | | | | | | | | | | |

**Address:** 0x03C

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [15:0] | sysals__thresh_low: Low Threshold value for ALS comparison. Range 0-65535 codes. |

### 6.2.34 SYSALS__INTERMEASUREMENT_PERIOD

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| sysals__intermeasurement_period | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x03E

**Type:** R/W

**Reset:** 0xFF

**Description:**

| | |
|---|---|
| [7:0] | sysals__intermeasurement_period: Time delay between measurements in ALS continuous mode. Range 0-254 (0 = 10ms). Step size = 10ms. |

### 6.2.35 SYSALS__ANALOGUE_GAIN

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | sysals__analogue_gain_light | | |
| R | | | | | R/W | | |

**Address:** 0x03F

**Type:** R/W

**Reset:** 0x06

**Description:**

| | |
|---|---|
| [2:0] | sysals__analogue_gain_light: ALS analogue gain (light channel)<br>0: ALS Gain = 20<br>1: ALS Gain = 10<br>2: ALS Gain = 5.0<br>3: ALS Gain = 2.5<br>4: ALS Gain = 1.67<br>5: ALS Gain = 1.25<br>6: ALS Gain = 1.0<br>7: ALS Gain = 40<br>Controls the “light” channel gain.<br>Note: Upper nibble should be set to 0x4 i.e. For ALS gain of 1.0 write 0x46. |

### 6.2.36 SYSALS__INTEGRATION_PERIOD

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | sysals__integration_period | | | | | | | | |
| R | | | | | | | R/W | | | | | | | | |

**Address:** 0x040

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [8:0] | sysals__integration_period: Integration period for ALS mode. 1 code = 1 ms (0 = 1 ms).<br>Recommended setting is 100 ms (0x63). |

## 6.2.37 RESULT__RANGE_STATUS

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| result__range_error_code | | | | result__range_min_threshold_hit | result__range_max_threshold_hit | result__range_measurement_ready | result__range_device_ready |
| R | | | | R | R | R | R |

**Address:** 0x04D

**Type:** R

**Reset:** 0x1

**Description:**

| | |
|---|---|
| [7:4] | result__range_error_code: Specific error codes<br>0000: No error<br>0001: VCSEL Continuity Test<br>0010: VCSEL Watchdog Test<br>0011: VCSEL Watchdog<br>0100: PLL1 Lock<br>0101: PLL2 Lock<br>0110: Early Convergence Estimate<br>0111: Max Convergence<br>1000: No Target Ignore<br>1001: Not used<br>1010: Not used<br>1011: Max Signal To Noise Ratio<br>1100: Raw Ranging Algo Underflow<br>1101: Raw Ranging Algo Overflow<br>1110: Ranging Algo Underflow<br>1111: Ranging Algo Overflow |
| [3] | result__range_min_threshold_hit: **Legacy register - DO NOT USE**<br>Use instead *6.2.39: RESULT__INTERRUPT_STATUS_GPIO* |
| [2] | result__range_max_threshold_hit: **Legacy register - DO NOT USE**<br>Use instead *6.2.39: RESULT__INTERRUPT_STATUS_GPIO* |
| [1] | result__range_measurement_ready: **Legacy register - DO NOT USE**<br>Use instead *6.2.39: RESULT__INTERRUPT_STATUS_GPIO* |
| [0] | result__range_device_ready: Device Ready. When set to 1, indicates the device mode and configuration can be changed and a new start command will be accepted. When 0, indicates the device is busy. |

### 6.2.38 RESULT__ALS_STATUS

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| result__als_error_code | | | | result__als_min_threshold_hit | result__als_max_threshold_hit | result__als_measurement_ready | result__als_device_ready |
| R | | | | R | R | R | R |

**Address:** 0x04E

**Type:** R

**Reset:** 0x1

**Description:**

| | |
|---|---|
| [7:4] | result__als_error_code: Specific error and debug codes<br>0000: No error<br>0001: Overflow error<br>0002: Underflow error |
| [3] | result__als_min_threshold_hit: **Legacy register - DO NOT USE**<br>Use instead *6.2.39: RESULT__INTERRUPT_STATUS_GPIO* |
| [2] | result__als_max_threshold_hit: **Legacy register - DO NOT USE**<br>Use instead *6.2.39: RESULT__INTERRUPT_STATUS_GPIO* |
| [1] | result__als_measurement_ready: **Legacy register - DO NOT USE**<br>Use instead *6.2.39: RESULT__INTERRUPT_STATUS_GPIO* |
| [0] | result__als_device_ready: Device Ready. When set to 1, indicates the device mode and configuration can be changed and a new start command will be accepted. When 0 indicates the device is busy. |

### 6.2.39 RESULT__INTERRUPT_STATUS_GPIO

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| result_int_error_gpio | | result_int_als_gpio | | | result_int_range_gpio | | |
| R | | R | | | R | | |

**Address:** 0x04F

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [7:6] | result_int_error_gpio: Interrupt bits for Error:<br>0: No error reported<br>1: Laser Safety Error<br>2: PLL error (either PLL1 or PLL2) |
| [5:3] | result_int_als_gpio: Interrupt bits for ALS:<br>0: No threshold events reported<br>1: Level Low threshold event<br>2: Level High threshold event<br>3: Out Of Window threshold event<br>4: New Sample Ready threshold event |
| [2:0] | result_int_range_gpio: Interrupt bits for Range:<br>0: No threshold events reported<br>1: Level Low threshold event<br>2: Level High threshold event<br>3: Out Of Window threshold event<br>4: New Sample Ready threshold event |

### 6.2.40 RESULT__ALS_VAL

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| result__als_ambient_light | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | |

**Address:** 0x050

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [15:0] | result__als_ambient_light: 16 Bit ALS count output value. Lux value depends on Gain and integration settings and calibrated lux/count setting. |

### 6.2.41 RESULT__HISTORY_BUFFER_x

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESULT__HISTORY_BUFFER_0 | result__history_buffer_0 | | | | | | | | | | | | | | | |
| RESULT__HISTORY_BUFFER_1 | result__history_buffer_1 | | | | | | | | | | | | | | | |
| RESULT__HISTORY_BUFFER_2 | result__history_buffer_2 | | | | | | | | | | | | | | | |
| RESULT__HISTORY_BUFFER_3 | result__history_buffer_3 | | | | | | | | | | | | | | | |
| RESULT__HISTORY_BUFFER_4 | result__history_buffer_4 | | | | | | | | | | | | | | | |
| RESULT__HISTORY_BUFFER_5 | result__history_buffer_5 | | | | | | | | | | | | | | | |
| RESULT__HISTORY_BUFFER_6 | result__history_buffer_6 | | | | | | | | | | | | | | | |
| RESULT__HISTORY_BUFFER_7 | result__history_buffer_7 | | | | | | | | | | | | | | | |
| | R | | | | | | | | | | | | | | | |

**Address:** 0x052 + x * 0x2 (x=0 to 7)

**Type:** R

**Reset:** 0x0

**Description:** See also *6.2.11: SYSTEM__HISTORY_CTRL*

| | |
|---|---|
| RESULT__HISTORY_BUFFER_0: [15:0] | result__history_buffer_0: Range/ALS result value.<br>Range mode; Bits[15:8] range_val_latest; Bits[7:0] range_val_d1;<br>ALS mode; Bits[15:0] als_val_latest |
| RESULT__HISTORY_BUFFER_1: [15:0] | result__history_buffer_1: Range/ALS result value.<br>Range mode; Bits[15:8] range_val_d2; Bits[7:0] range_val_d3;<br>ALS mode; Bits[15:0] als_val_d1 |
| RESULT__HISTORY_BUFFER_2: [15:0] | result__history_buffer_2: Range/ALS result value.<br>Range mode; Bits[15:8] range_val_d4; Bits[7:0] range_val_d5;<br>ALS mode; Bits[15:0] als_val_d2 |
| RESULT__HISTORY_BUFFER_3: [15:0] | result__history_buffer_3: Range/ALS result value.<br>Range mode; Bits[15:8] range_val_d6; Bits[7:0] range_val_d7;<br>ALS mode; Bits[15:0] als_val_d3 |
| RESULT__HISTORY_BUFFER_4: [15:0] | result__history_buffer_4: Range/ALS result value.<br>Range mode; Bits[15:8] range_val_d8; Bits[7:0] range_val_d9;<br>ALS mode; Bits[15:0] als_val_d4 |
| RESULT__HISTORY_BUFFER_5: [15:0] | result__history_buffer_5: Range/ALS result value.<br>Range mode; Bits[15:8] range_val_d10; Bits[7:0] range_val_d11;<br>ALS mode; Bits[15:0] als_val_d5 |
| RESULT__HISTORY_BUFFER_6: [15:0] | result__history_buffer_6: Range/ALS result value.<br>Range mode; Bits[15:8] range_val_d12; Bits[7:0] range_val_d13;<br>ALS mode; Bits[15:0] als_val_d6 |
| RESULT__HISTORY_BUFFER_7: [15:0] | result__history_buffer_7: Range/ALS result value.<br>Range mode; Bits[15:8] range_val_d14; Bits[7:0] range_val_d15;<br>ALS mode; Bits[15:0] als_val_d7 |

### 6.2.42 RESULT__RANGE_VAL

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| result__range_val | | | | | | | |
| R | | | | | | | |

**Address:** 0x062

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [7:0] | result__range_val: Final range result value presented to the user for use. Unit is in mm. |

### 6.2.43 RESULT__RANGE_RAW

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| result__range_raw | | | | | | | |
| R | | | | | | | |

**Address:** 0x064

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [7:0] | result__range_raw: Raw Range result value with offset applied (no cross talk compensation applied). Unit is in mm. |

### 6.2.44 RESULT__RANGE_RETURN_RATE

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| result__range_return_rate | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | |

**Address:** 0x066

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [15:0] | result__range_return_rate: sensor count rate of signal returns correlated to IR emitter. Computed from RETURN_SIGNAL_COUNT / RETURN_CONV_TIME. Mcps 9.7 format |

### 6.2.45 RESULT__RANGE_REFERENCE_RATE

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| result__range_reference_rate | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | |

**Address:** 0x068

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [15:0] | result__range_reference_rate: sensor count rate of reference signal returns. Computed from REFERENCE_SIGNAL_COUNT / RETURN_CONV_TIME. Mcps 9.7 format<br>Note: Both arrays converge at the same time, so using the return array convergence time is correct. |

### 6.2.46 RESULT__RANGE_RETURN_SIGNAL_COUNT

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| result__range_return_signal_count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Address:** 0x06C

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [31:0] | result__range_return_signal_count: sensor count output value attributed to signal correlated to IR emitter on the Return array. |

### 6.2.47 RESULT__RANGE_REFERENCE_SIGNAL_COUNT

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| result__range_reference_signal_count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Address:** 0x070

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [31:0] | result__range_reference_signal_count: sensor count output value attributed to signal correlated to IR emitter on the Reference array. |

### 6.2.48 RESULT__RANGE_RETURN_AMB_COUNT

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| result__range_return_amb_count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Address:** 0x074

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [31:0] | result__range_return_amb_count: sensor count output value attributed to uncorrelated ambient signal on the Return array. Must be multiplied by 6 if used to calculate the ambient to signal threshold. |

### 6.2.49 RESULT__RANGE_REFERENCE_AMB_COUNT

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| result__range_reference_amb_count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Address:** 0x078

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [31:0] | result__range_reference_amb_count: sensor count output value attributed to uncorrelated ambient signal on the Reference array. |

### 6.2.50 RESULT__RANGE_RETURN_CONV_TIME

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| result__range_return_conv_time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Address:** 0x07C

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [31:0] | result__range_return_conv_time: sensor count output value attributed to signal on the Return array. |

### 6.2.51 RESULT__RANGE_REFERENCE_CONV_TIME

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| result__range_reference_conv_time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Address:** 0x080

**Type:** R

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [31:0] | result__range_reference_conv_time: sensor count output value attributed to signal on the Reference array. |

### 6.2.52 READOUT__AVERAGING_SAMPLE_PERIOD

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| readout__averaging_sample_period | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x10A

**Type:** R/W

**Reset:** 0x30

**Description:**

| | |
|---|---|
| [7:0] | readout__averaging_sample_period: The internal readout averaging sample period can be adjusted from 0 to 255. Increasing the sampling period decreases noise but also reduces the effective max convergence time and increases power consumption:<br>Effective max convergence time = max convergence time - readout averaging period (see *Section 2.5: Range timing*). Each unit sample period corresponds to around 64.5 µs additional processing time. The recommended setting is 48 which equates to around 4.3 ms. |

### 6.2.53 FIRMWARE__BOOTUP

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | firmware__bootup |
| R | | | | | | | R/W |

**Address:** 0x119

**Type:** R/W

**Reset:** 0x1

**Description:**

| | |
|---|---|
| [0] | firmware__bootup: FW must set bit once initial boot has been completed. |

### 6.2.54 FIRMWARE__RESULT_SCALER

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | firmware__als_result_scaler | | | |
| R | | | | R/W | | | |

**Address:** 0x120

**Type:** R/W

**Reset:** 0x1

**Description:**

| | |
|---|---|
| [3:0] | firmware__als_result_scaler: Bits [3:0] analogue gain 1 to 16x |

### 6.2.55 I2C_SLAVE__DEVICE_ADDRESS

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | super_i2c_slave__device_address | | | | | | |
| R | R/W | | | | | | |

**Address:** 0x212

**Type:** R/W

**Reset:** 0x29

**Description:**

| | |
|---|---|
| [6:0] | super_i2c_slave__device_address: User programmable $I^2C$ address (7-bit). Device address can be re-designated after power-up. |

### 6.2.56 INTERLEAVED_MODE__ENABLE

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| interleaved_mode__enable | | | | | | | |
| R/W | | | | | | | |

**Address:** 0x2A3

**Type:** R/W

**Reset:** 0x0

**Description:**

| | |
|---|---|
| [7:0] | Interleaved mode enable: Write 0x1 to this register to select ALS+Range interleaved mode.<br>Use SYSALS__START and SYSALS__INTERMEASUREMENT_PERIOD to control this mode.<br>A range measurement is automatically performed immediately after each ALS measurement. |

# 7 Outline drawing

**Figure 27. Outline drawing (page 1/2)**

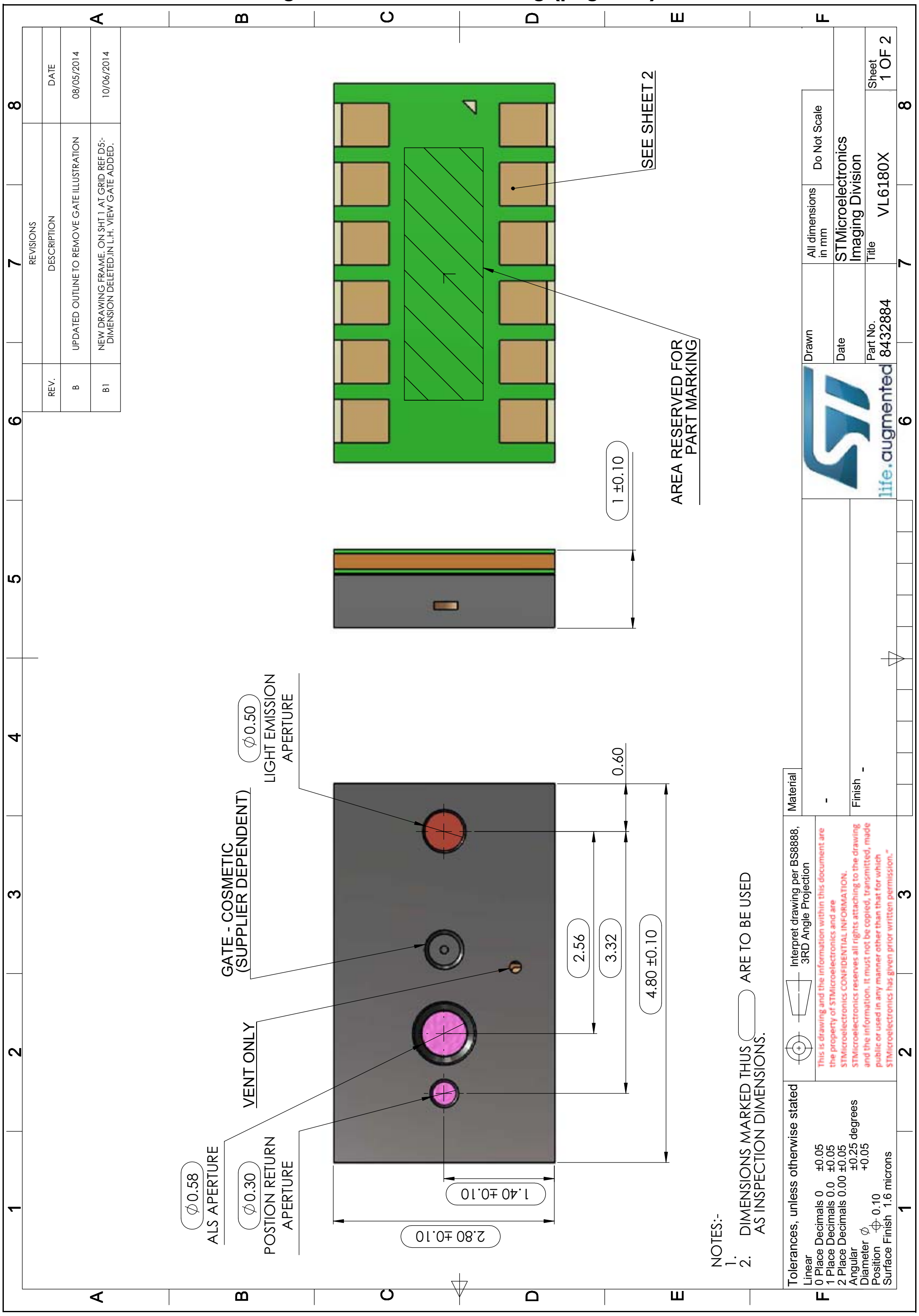
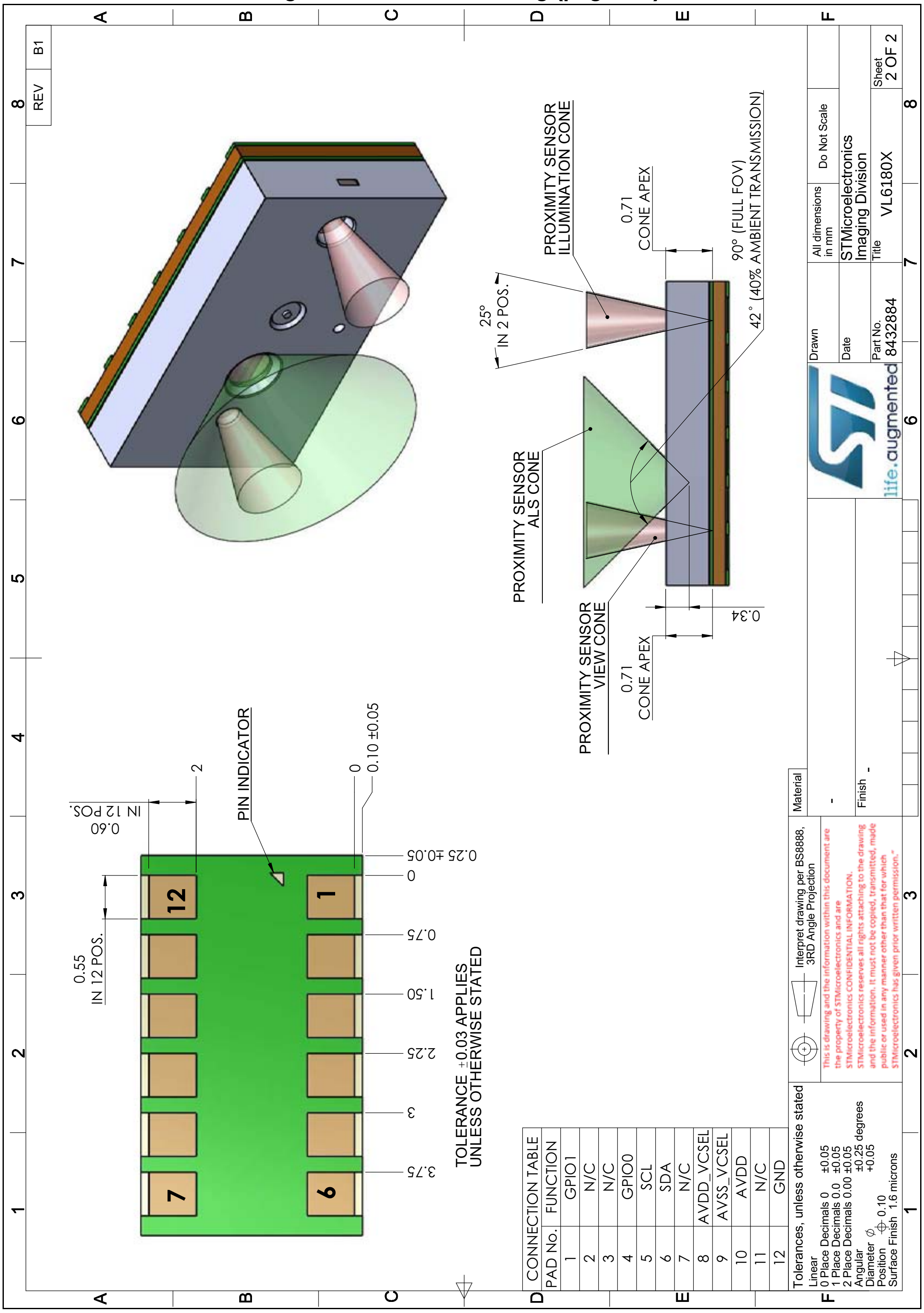


**Figure 28. Outline drawing (page 2/2)**



# 8 Laser safety considerations

The VL6180X contains a laser emitter and corresponding drive circuitry. The laser output is designed to remain within Class 1 laser safety limits under all reasonably foreseeable conditions including single faults in compliance with IEC 60825-1:2007. The laser output will remain within Class 1 limits as long as the STMicroelectronics recommended device settings are used and the operating conditions specified in this datasheet are respected. The laser output power must not be increased by any means and no optics should be used with the intention of focusing the laser beam.

**Figure 29. Class 1 laser product label**


## 8.1 Compliance

Complies with 21 CFR 1040.10 and 1040.11 except for deviations pursuant to Laser Notice No.50, dated June 24, 2007.

# 9 Ordering information

VL6180X is currently available in the following format. More detailed information is available on request.

**Table 26. Delivery format**

| Order code | Description |
|---|---|
| VL6180XV0NR/1 | Tape and reel (5000 units in a reel) |

## 9.1 Traceability and identification

Latest ROM revision can be identified as follows:

0x002 IDENTIFICATION__MODEL_REV_MINOR = 3

The minimum information required for traceability is the content of the following registers:

0x006 - IDENTIFICATION__DATE_HI

0x007 - IDENTIFICATION__DATE_LO

0x008 - IDENTIFICATION__TIME (16-bit)

0x00A - IDENTIFICATION__CODE

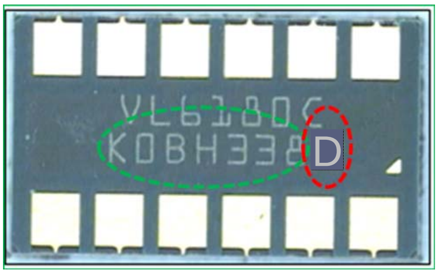
With this information, the module can be uniquely identified.

Preferably, all the IDENTIFICATION register contents should be provided for traceability.

## 9.2 Part marking

Devices are marked on the underside as shown below. 1st line is the product ID. 2nd line is the manufacturing info. (circled in green), where the 1st four letters are the lot ID and the last 3 digits are the year + week number. Here: 338 is 201**3** wk**38**. The final letter, circled in red, is the ROM revision ('D').

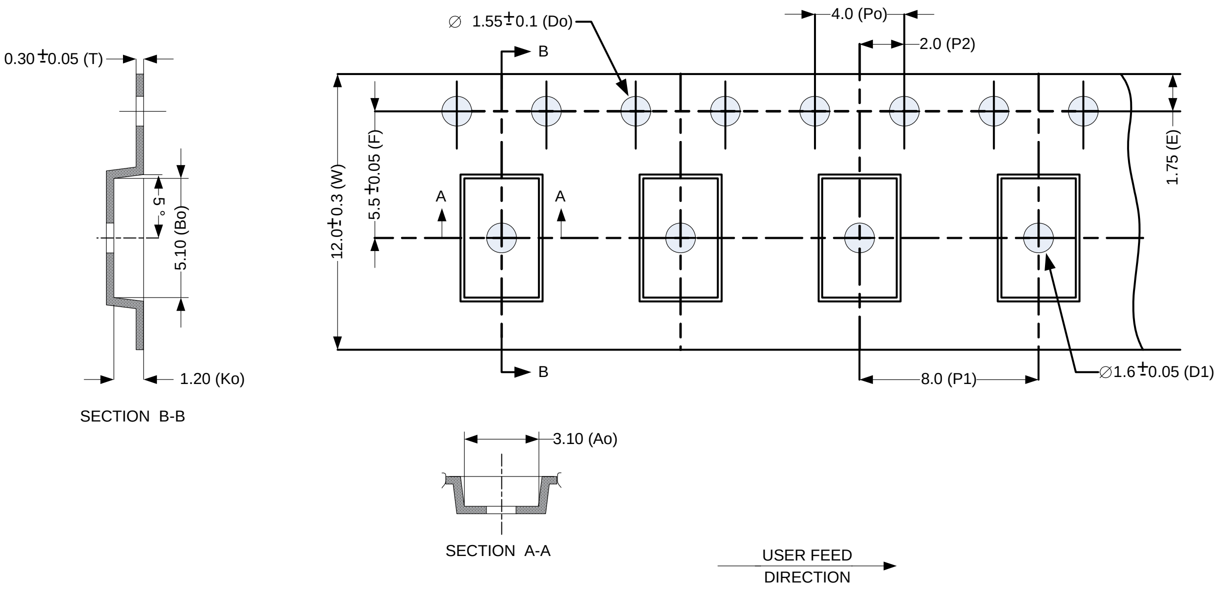
**Figure 30. Part marking**



## 9.3 Packaging

The Root part number 1 is available in tape and reel packaging as shown in *Figure 31.*

**Figure 31. Tape and reel packaging**

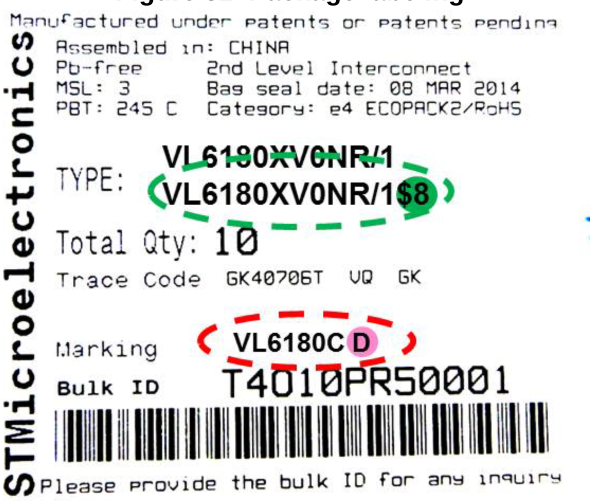



| Ao ±0.1 | Bo ±0.1 | Ko ±0.1 | E ±0.1 | F±0.05 | Po ±0.1 | P1 ±0.1 | P2 ±0.1 | Do ±0.1 | T±0.05 | W±0.3 |
|---|---|---|---|---|---|---|---|---|---|---|
| 3.10 | 5.10 | 1.20 | 1.75 | 5.5 | 4.0 | 8.0 | 2.0 | 1.55 | 0.30 | 12.0 |

### 9.3.1 Package labeling

The labeling on the packing carton is shown in *Figure 32.* The latest ROM revision is indicated alongside the order code (shaded green) and also after the product marking (shaded pink).

**Figure 32. Package labeling**



## 9.4 Storage

The Root part number 1 is a MSL 3 package.

**Table 27. Storage conditions**

| Level | Floor Life (out of bag) at Factory Ambient <30$^{o}$C/60% RH |
|---|---|
| 3 | 1 Week |

After this limit, dry bake to be done; 3 hours at 125$^{o}$C.

## 9.5 ROHS compliance

The Root part number 1 is Ecopack2 compliant as per ST definition.

Devices which are ROHS compliant even with use of ROHS exemption(s) and free of Halogenated flame retardant are named ECOPACK2 devices with the following definition:

- ROHS compliant even with use of ROHS exemption(s)
- 500 ppm maximum of Antimony as oxide or organic compound in each organic assy Materials (glue, substrate, mod compounds, housing...). Antimony in ceramic parts, in glass and in solder alloy is not restricted.
- 900 ppm maximum Bromine + Chlorine in each organic ass materials (glue, substrate, mold compounds, housing...)

These values are referring to maximum total content not to extractable ions content. Purchasing specification of assembly materials can impose lower values for technical reasons.

ECOPACK2 devices are of course fully compliant to ST banned and declarable substances specification and for example cannot contain red Phosphorus flame retardant.

# 10 ECOPACK®

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: *www.st.com*. ECOPACK® is an ST trademark.

# 11 Revision history

**Table 28. Document revision history**

| Date | Revision | Changes |
|---|---|---|
| 23-Sep-2013 | 1 | Initial release. |
| 30-Jan-2014 | 1.1 | General update for latest ROM revision:<br>*Section 1.1: Technical specification* updated<br>*Section 1.4: Application schematic* updated<br>*Section 1.5: Recommended solder pad dimensions* updated<br>Notes added to *Figure 5.: Recommended reflow profile*<br>*Section 2.13: Ambient light sensor (ALS)* updated.<br>*Section 3.1: Absolute maximum ratings* added<br>*Section 3.2: Normal operating conditions* extended<br>*Section 4: Performance specification* added<br>Revised outline drawing added to *Section 7: Outline drawing*<br>Class 1 laser product label added to *Section 7: Outline drawing*<br>*Section 9: Ordering information* added information relating to device marking and package labeling |
| 02-Apr-2014 | 1.2 | Updates to the following sections:<br>*Section 1.5: Recommended solder pad dimensions*<br>*Section 3.2: Normal operating conditions*<br>*Section 3.4: Electrical characteristics*<br>*Section 4.1: Proximity ranging (0 to 100mm)*<br>Added *Section 4.2: ALS performance*<br>Corrected error codes in.*Section 6.2.38: RESULT__ALS_STATUS*<br>Updated *Section 6.2.20: SYSRANGE__MAX_CONVERGENCE_TIME*<br>Product code changed to VL6180X |
| 09-Apr-14 | 2 | Add documentation reference number (026171)<br>Update Disclaimer |
| 15-May-14 | 3 | ALS linearity spec updated in *Section 4.2: ALS performance*<br>Updated some detail in *Table 1.: Technical specification*<br>Added comment to *Section 1.3: Device pinout* stating that pins labeled ‘no connect’ can optionally be connected to ground<br>Added test condition to *Section 3.3: Current consumption*<br>Errata corrections in *6.2.8*, *6.2.35* and *6.2.54*<br>*Section 7: Outline drawing* updated (no dimensional changes)<br>Dry bake conditions updated in *Section 9.4: Storage* |
| 28-May-14 | 4 | Added *Section 8.1: Compliance* |

Table 28. Document revision history (continued)

| Date | Revision | Changes |
|---|---|---|
| 16-Jun-14 | 5 | Re-write of *Section 2: Functional description*.<br>*Section 6: Device registers*: Added introduction and minor corrections<br>*Section 7: Outline drawing* updated to Rev B1. Supplier dependent gate mark added. |
| 20-Aug-2014 | 6 | Updates:<br>*Section 2.8.3: Signal-to-noise ratio (SNR)*: Clarified SNR calculation.<br>*Section 6: Device registers*: Corrected a clarified some register descriptions.<br>Typical ranging performance graph updated.<br>Delivery & manufacturing info updated. |