Rev 1

VL53L5CX Linux driver

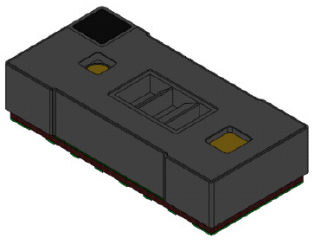
Getting started with Linux Driver and Raspberry PI 3

Introduction

The VL53L5CX is a state of the art, Time-of-Flight (ToF), laser-ranging sensor enhancing the ST FlightSense product family. Housed in a miniature reflowable package, it integrates a SPAD array, physical infrared filters, and diffractive optics (DOE) to achieve the best ranging performance in various ambient lighting conditions with a range of cover glass materials.

The purpose of this user manual is to describes how to start using the Linux Driver based on VL53L5CX Ultra Lite Driver (ULD).

Figure 1. VL53L5CX sensor module



References:

1) VL53L5CX User Manual (UM2284)

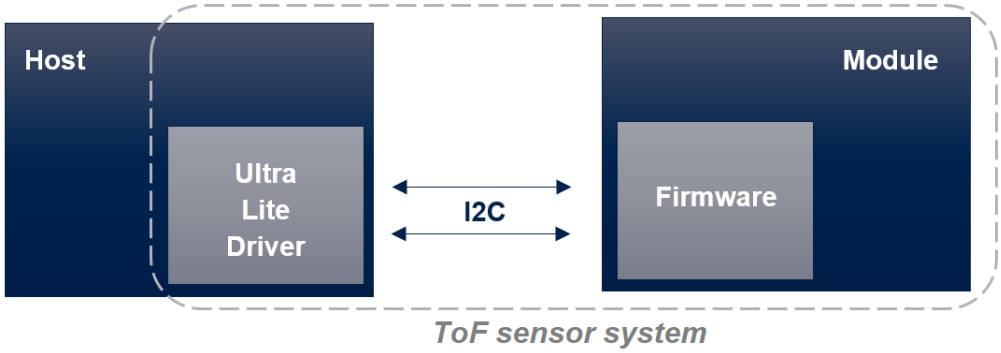
Contents

1 Functional description

1.1 System overview

The VL53L5CX system is composed of a hardware module and the Ultra Lite Driver software (VL53L5CX ULD) running on a host (see figure below). The hardware module contains the ToF sensor. ST delivers the software driver which is referred to in this document as "the driver". This document describes the functions of the driver which are accessible to the host. These functions control the sensor and get the ranging data.

Figure 2. VL53L5CX system overview

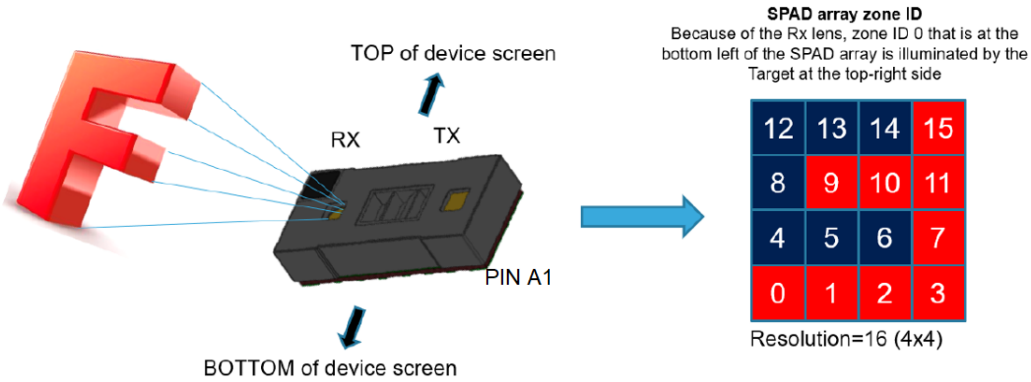


1.2 Effective orientation

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

For an application, when the user is looking at the screen of the device, it is recommended to orientate the VL53L5CX with the RX aperture to the left and the TX aperture to the right as shown in *Figure 3*.

Figure 3. VL53L5CX effective orientation

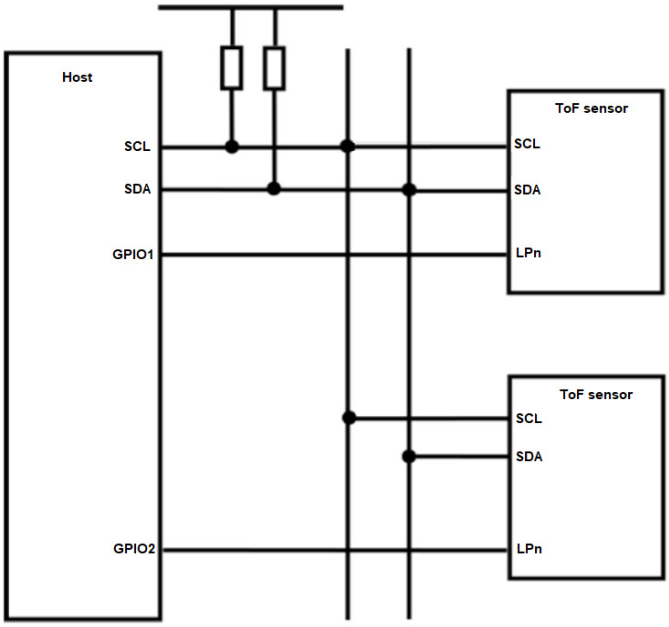


1.3 Schematics and I2C configuration

The communication between driver and firmware is handled by I2C, with a capability of operating up to 1MHz. The implementation requires pull-ups on the SCL and SDA lines. Please see VL53L5CX datasheet for more information.

The VL53L5CX device has a default I2C address of 0x52. However, it is possible to change the default address to avoid conflicts with other devices, or facilitate adding multiple VL53L5CX modules to the system for a greater system FoV. The I2C address can be changed using vl53l5cx_set_i2c_address() function.

Figure 4. Multiple sensors on I2C bus



To allow a device to have its I2C address changed without affecting others on the I2C bus, it is important to disable the I2C communication of the devices not being changed. The procedure is the following one :

1). Power up the system as normal.

2). Pull down the LPn pin of the device that will not be having its address changed.

3). Pull up the LPn pin of the device that has the I2C address changed.

4). Program the I2C address to the device using function *vl53l5cx_set_i2c_address()*.

5). Pull up the LPn pin of the device not being reprogrammed.

All devices should now be available on the I2C bus. Repeat the above steps for as many VL53L5CX devices in the system that require a new I2C address

2 Package content and data flow

2.1 Driver architecture and content

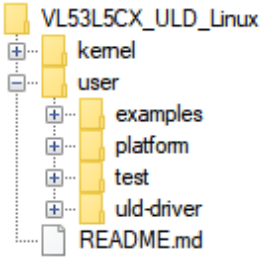
The VL53L5CX Linux driver is based on the VL53L5CX Ultra Lite Driver (ULD). All driver ULD features are described into the User Manual UM2884.

The Linux driver is a package composed of several folders:

- kernel / user folder: contain the two supported modes of the driver
- examples: contain several examples to use the sensor
- platform: contain mandatory macros and platform functions. It needs to be filled by the customer.
- test: contain the main input file
- uld-driver: contain the VL53L5CX Ultra Lite Driver (ULD)

The described driver architecture is represented on *Figure 5*.

Figure 5. Linux driver architecture



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

2.2 Supported system modes

The VL53L5CX Linux driver support the User space mode and Full Kernel mode. The mode can be selected using compilation flag STMVL53L5CX_KERNEL into the test Makefile. By default the driver is set in User space mode.

3 Running example on Raspberry PI 3

The proposed Linux implementation is customized to run on a Raspberry PI 3, but can be adapted to run on any Linux embedded application, as far as the VL53L5CX sensor is connected through I2C.

3.1 Hardware connection

The most simple example to run an example with Linux driver is to use a VL53L5CX SATEL. Pins need to be connected following the schematics *Figure 6*.

Figure 6. VL53L5CX SATEL to Raspberry PI 3 - schematics

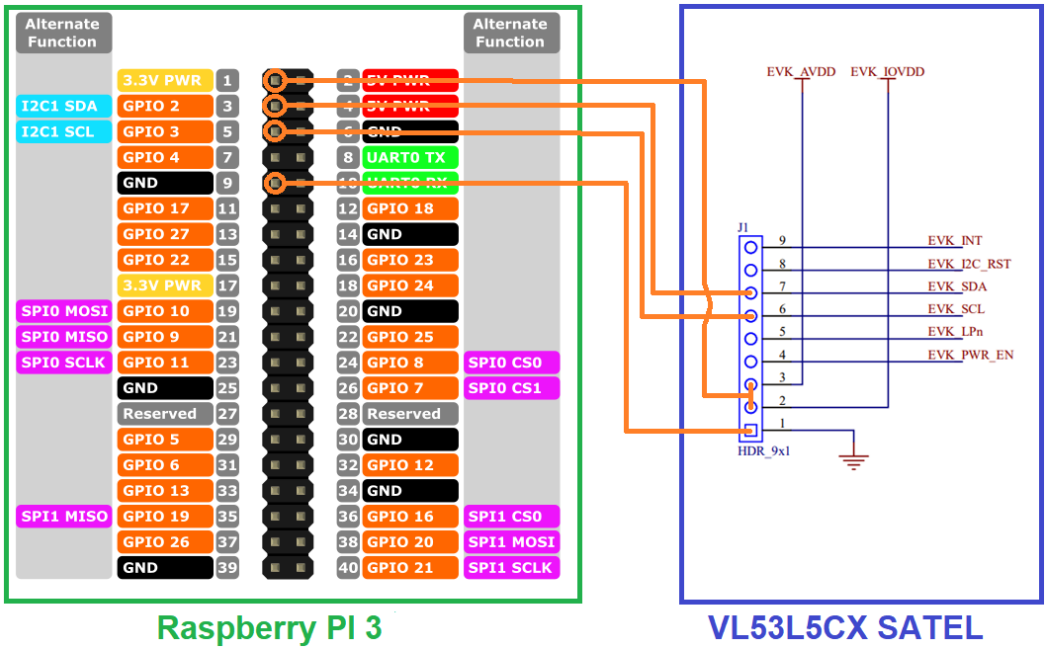
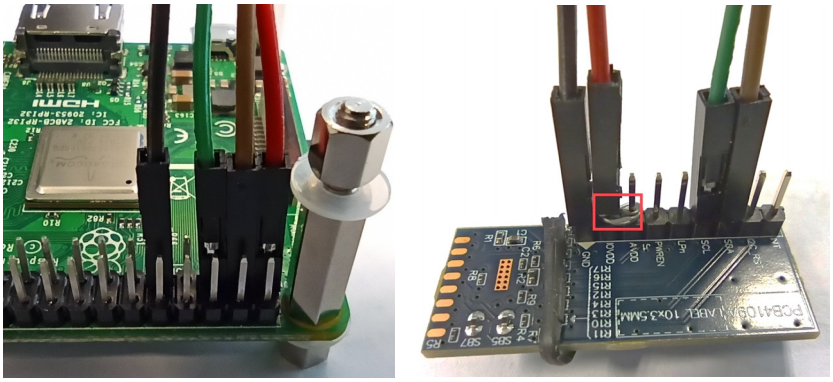


Figure 7. VL53L5CX SATEL to Raspberry PI 3 - photos



Note: Sensor AVDD and IOVDD are both connected to 3V3 of Rpi board.

3.2 User space compilation

User space mode only require to compile the makefile located into the /test folder. The following sequence is required:

- Go to test folder
  - cd VL53L5CX_ULD_Linux/user/test
- Build program
  - make

Then user can run application as described in section 3.4.

3.3 Kernel compilation

Kernel compilation requires installation if kernel source headers and device tree blog to be run. The following sequence is required:

- Install the Raspberry PI Kernel source headers (refer to official documentation to download a version matching with kernel version)
- Update the boot configuration, adding or un-commenting the following lines of file /boot/config.txt
  - dtparam=i2c_arm=on
  - dtparam=i2c1=on
  - dtparam=i2c1_baudrate=1000000
  - dtoverlay=stmvl53l5cx
- Compile the device tree blob
  - cd VL53L5CX_ULD_Linux
  - make dtb
  - sudo reboot
- Go to test folder and enable kernel cflag into the test Makefile
  - cd VL53L5CX_ULD_Linux/user/test
  - CFLAGS_RELEASE += STMVL53L5CX_KERNEL
- Build program
  - make
- Compile the kernel module
  - cd VL53L5CX_ULD_Linux/kernel
  - make clean
  - make
  - sudo make insert

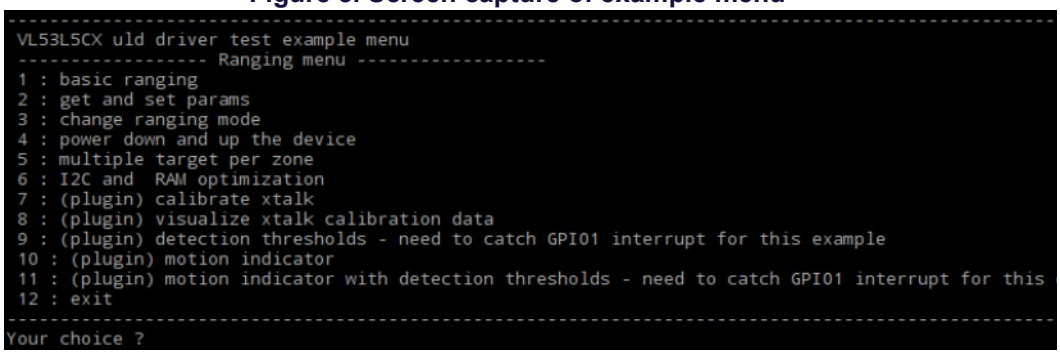
Then user can run application as described in section 3.4.

3.4 Run a test application

After compilation done, user can go to the test folder and run the example menu. The menu allows using examples located in the /examples folder.

- cd VL53L5CX_ULD_Linux/user/test
- ./menu

Figure 8. Screen capture of example menu



Note: Examples 9 and 11 requires to catch interrupt raised on INT pin. User need to add wires and to implement GPIO callback for using such examples.

4 Document revision history

Table 1. Acronyms and abbreviations

Date	Version	Changes
28-May-2021	1.0	Initial release