life.augmented

UM2853

User manual

How to use the VL53L3CX with STMicroelectronics' X-CUBE-TOF1 Time-of-Flight sensor software packages for STM32CubeMX

Introduction

The X-CUBE-TOF1 expansion software package for STM32Cube runs on the STM32 and includes drivers that recognize the sensors and perform simple ranging on single or multiple devices.

The expansion is built on STM32Cube software technology to ease portability across different STM32 microcontrollers.

The software comes with a sample implementation of the drivers running on different Time-of-Flight (ToF) sensor evaluation boards connected to a featured STM32 Nucleo development board.

This user manual focuses on the VL53L3CX ToF ranging sensor with multitarget detection. For further information on the ToF sensors supported by X-CUBE-TOF1, refer to the software page of www.st.com.

The VL53L3CX evaluation boards supported by the X-CUBE-TOF1 expansion software package include:

- X-NUCLEO-53L3A2 expansion board
- VL53L3CX-SATEL breakout boards

The X-CUBE-TOF1 software provides the following sample applications for the VL53L3CX:

- 53L3A2_SimpleRanging for X-NUCLEO-53L3A2 and optional cover glass for a calibration application
- 53L3A2_MultiSensorRanging for X-NUCLEO-53L3A2 and VL53L3CX-SATEL
- VL53L3CX_SimpleRanging for VL53L3CX-SATEL

Visit the STM32Cube ecosystem web page on www.st.com for further information.

1 Acronyms and abbreviations

Acronym	Definition
API	application programming interface
BSP	board support package
HAL	hardware abstraction layer
I2C	inter-integrated circuit
IDE	integrated development environment
MCU	microcontroller unit
NVIC	nested vector interrupt control
PCB	printed circuit board
SDK	software development kit
ToF	Time-of-Flight sensor
USB	universal serial BUS

2 X-CUBE-TOF1 software expansion for STM32Cube

2.1 Overview

The X-CUBE-TOF1 software package expands the STM32Cube functionality. The key features are:

- Complete software to build applications using the VL53L3CX evaluation boards listed in Section Introduction.
- Several application examples to show the innovative technology for the accurate distance ranging capability.
- Sample application to transmit real-time sensor data to a PC.
- Precompiled binaries available on all evaluation boards listed in Section Introduction connected to a NUCLEO-F401RE or NUCLEO-L476RG development board.
- Package compatible with STM32CubeMX, can be downloaded from, and installed directly into, STM32CubeMX.
- Easy portability across different MCU families, thanks to STM32Cube.
- Free, user-friendly license terms.

2.2 Architecture

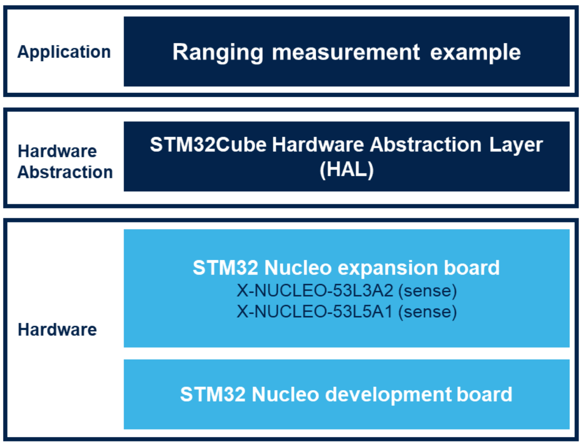
This software is a fully compliant expansion of STM32Cube enabling development of applications using Time-of-Flight sensors.

The software is based on the hardware abstraction layer for the STM32 microcontroller, STM32CubeHAL. The package extends STM32Cube by providing a board support package (BSP) for the sensor expansion board, and a sample application for serial communication with a PC.

The software layers used by the application software to access the sensor expansion board are:

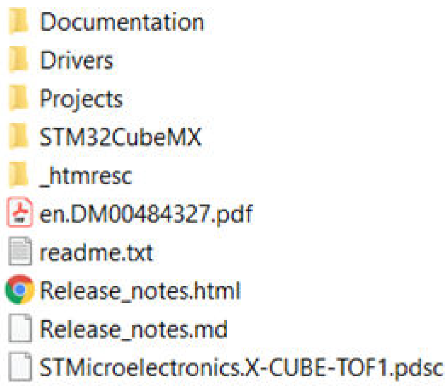
- The STM32Cube HAL driver layer. It provides a simple, generic, and multi-instance set of APIs (application programming interfaces) to interact with the upper layers (application, libraries, and stacks). It includes generic and extension APIs and is based on a generic architecture, which allows the layers built on it (such as the middleware layer) to implement their functionalities without dependence on the specific hardware configuration of a given microcontroller unit (MCU). This structure improves library code reusability and guarantees high portability across other devices.
- The BSP layer. It provides supporting software for the peripherals on the STM32 Nucleo board, except for the MCU. It has a set of APIs to provide a programming interface for certain board-specific peripherals (for example, the LED, the user button etc.), and allows identification of the specific board version. For the sensor expansion board, it provides the programming interface for various ToF sensors and provides support for initializing and reading sensor data.

Figure 1. X-CUBE-TOF1 software architecture



2.3 Folder structure

Figure 2. **X-CUBE-TOF1 package folder structure**



The following folders are included in the software package:

- The [**Documentation**] folder contains a compiled HTML file generated from the source code and detailed documentation regarding the software components and APIs.
- The [**Drivers**] folder contains the HAL drivers, the board-specific drivers for each supported board or hardware platform, including those for the on-board components and the CMSIS layer, which is a vendor-independent hardware abstraction layer for the Cortex-M processor series.
- The [**Projects**] folder contains several examples and applications for NUCLEO-L476RG and NUCLEO-F401RE platforms to show the use of sensor APIs provided with three development environments (IAR Embedded Workbench® for Arm™, MDK-ARM microcontroller development kit, STM32CubeIDE).
- The [**STM32CubeMX**] folder contains all the templates used by the CubeMX ToF pack.

2.4 APIs

Detailed technical information about the APIs available to the user can be found in the compiled HTML file X-CUBE-TOF1.chm in the [**Documentation**] folder of the software package, where all the functions and parameters are fully described.

3 VL53L3CX sample application descriptions

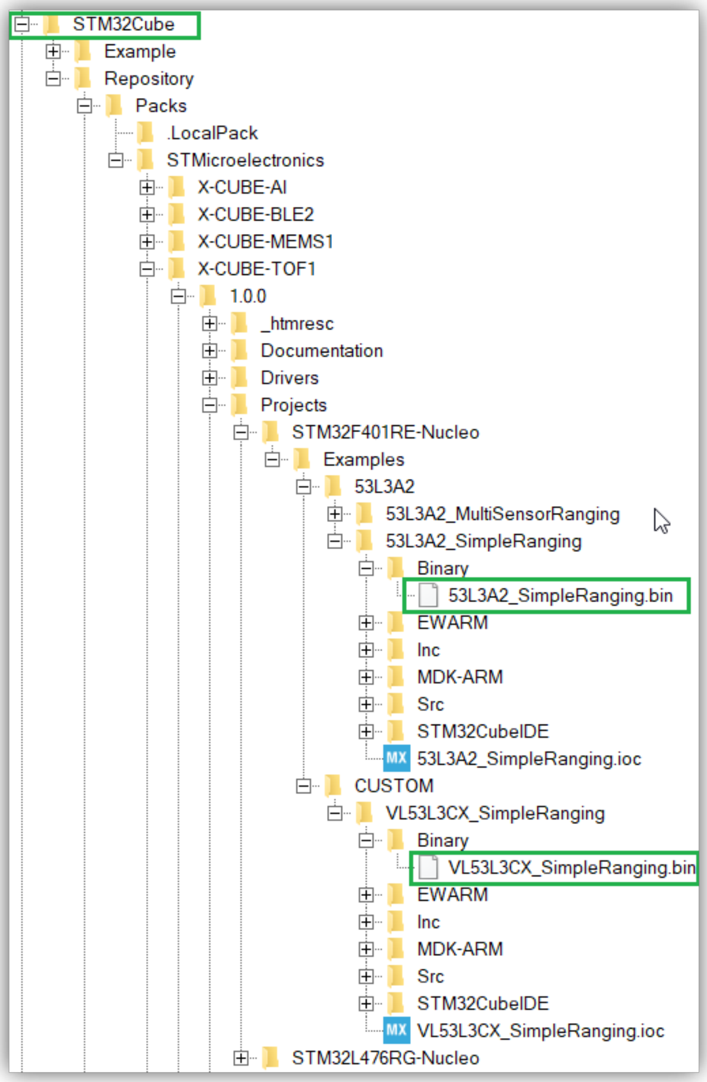
In this section, a short overview of the sample applications included in the X-CUBE-TOF1 pack is provided.

The sample applications:

- are ready-to-use projects that can be generated through the STM32CubeMX for any Nucleo board and using the X-NUCLEO-53L3A2 expansion board
- are ready-to-use projects that can be generated through the STM32CubeMX for any board equipped with an STM32 MCU and using the several supported ToF components.
- show the users how to use the APIs of the several ToF components to correctly initialize and use the ST ToF devices.

The precompiled binaries of the sample applications can be found under C:\Users\username\STM32Cube\Repository\Packs\STMicroelectronics\X-CUBE-TOF1\1.0.0\Projects\STM32F401RE-Nucleo\Examples\53L3A2\53L3A2_SimpleRanging\Binary as shown in the figure below. The user can directly use these binaries (which are built for the NUCLEO-F401RE and L476RG), or generate a new application for other STM32 Nucleo or STM32 MCU using the STM32CubeMX.

Figure 3. Precompiled projects location



3.1 53L3A2_SimpleRanging

This sample application shows how to use the X-NUCLEO-53L3A2 expansion board and a NUCLEO-F401RE or Nucleo L476RG to send the ranging data to a serial terminal, such as the Tera Term. In this example, the ranging data are displayed on the serial terminal.

The ranging data can be read by polling a register or triggering an interrupt. To select the data reading mode, refer to Section 4.2.1 How to generate the 53L3A2_SimpleRanging example with CubeMX.

This application can be run by loading the prebuilt binary 53L3A2_SimpleRanging.bin located as shown in Figure 3. Precompiled projects location or from a new project created with the STM32CubeMX.

1. After flashing the STM32 Nucleo board, either with the prebuilt binary file or from an IDE, open Tera Term and configure it with the settings below.

Figure 4. Tera Term, serial port setup

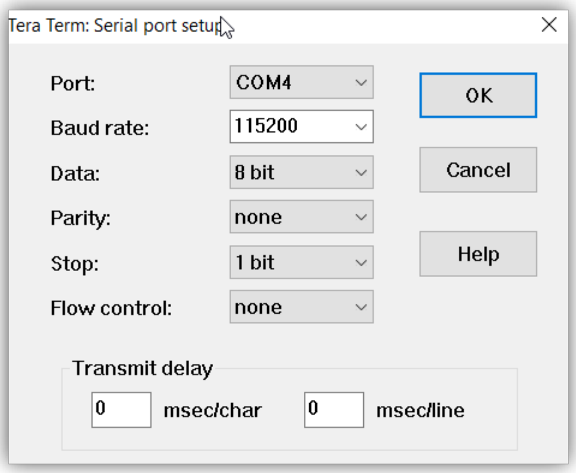
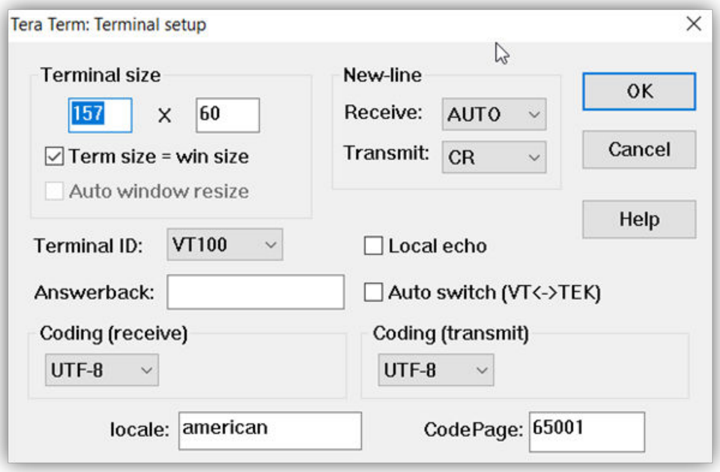
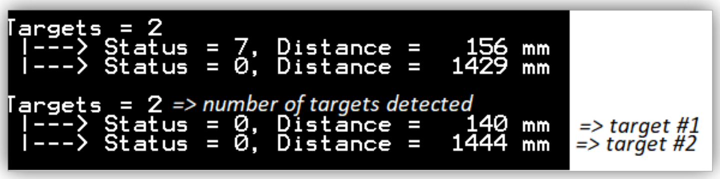


Figure 5. Tera Term, terminal setup



2. Place your hand in front of the sensor. The ranging data should be displayed on the serial terminal as shown below.

Figure 6. Ranging data



Note: Remove the protective film from the top of the ToF before first use.

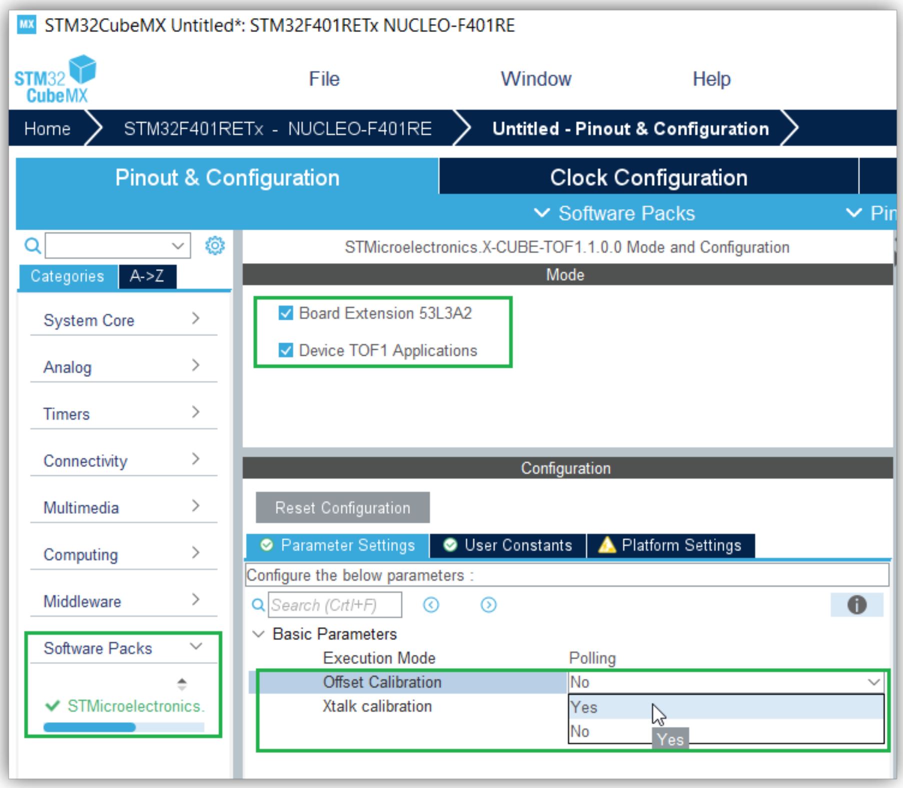
3.2 Offset and xtalk calibration applications

These sample applications show how to perform the calibrations (offset and crosstalk).

The sample applications are included in the 53L3A2_SimpleRanging application but they cannot be run directly from the prebuilt binary file. They can only be included only when generating a project with STM32CubeMX.

1. Select and configure the 53L3A2_SimpleRanging application in the software pack as described in Section 4.2 Use of expansion software with sample applications.
2. Complete the application configuration by selecting the calibration options as shown below.

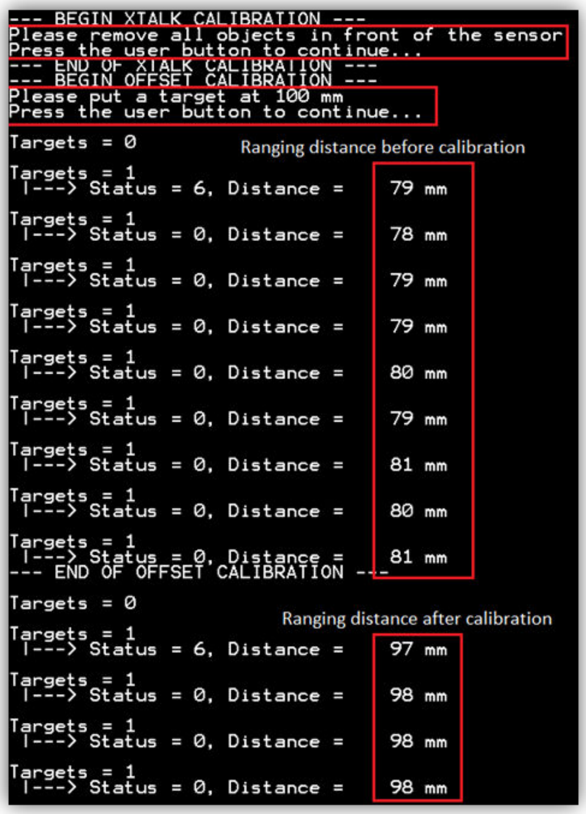
Figure 7. STM32CubeMX, offset and xtalk calibration



3. To test these applications, the cover glass kit (rectangle cover glass and spacers) and a fix target at 100 mm for the offset calibration are required. The calibration distance can be changed in the source code.

4. Run the application from the project generated through the STM32CubeMX and follow the instructions displayed on the serial terminal as shown below to perform the calibrations.

Figure 8. Calibration



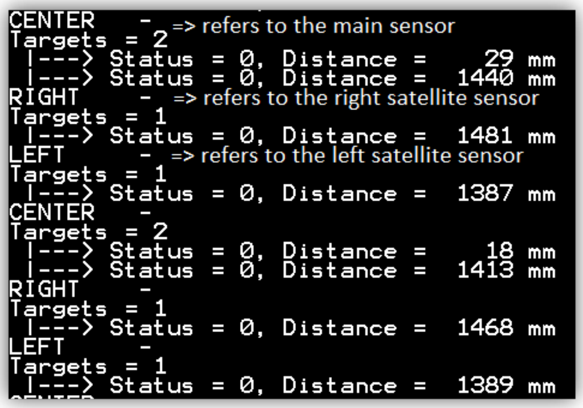
3.3 53L3A2_MultiSensorRanging

This sample application shows how to make three ToFs run simultaneously.

To test this application, two breakout boards VL53L3CX-SATEL, an X-NUCLEO-53L3A2, and a NUCLEO-F401RE or Nucleo L476RG are required. In this example, the ranging data is displayed on the serial terminal as shown below. This application can be run by loading the prebuilt binary 53L3A2_MultiSensorRanging.bin or from a new project created with STM32CubeMX.

Note: In this application, the ranging data is read by polling a register. No interrupt option is implemented.

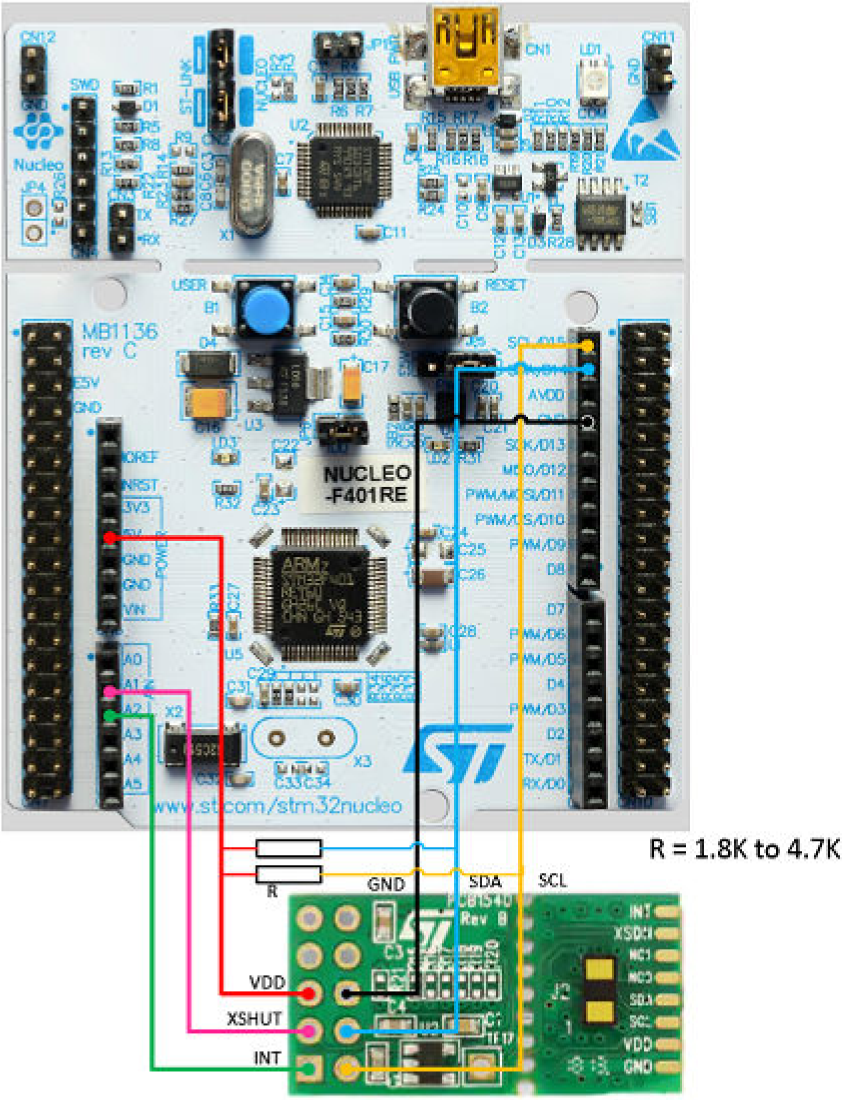
Figure 9. Multiple sensors ranging



3.4 VL53L3CX_SimpleRanging

This sample application shows how to range with the VL53L3CX_SATEL connected directly to the Nucleo F401RE or Nucleo L476RG without the expansion board.

Figure 10. VL53L3CX_SATEL connection



Note: Two resistors R [1.8K to 4.7K] must be added on SDA and SCL lines.

To test this application, one VL53L3CX-SATEL breakout board and one F401RE Nucleo are required. In this example, the ranging data is displayed on the serial terminal as shown in the figure below. This application can be run by flashing the Nucleo with the prebuilt binary VL53L3CX_SimpleRanging.bin from: C:\Users\user_name\STM32Cube\Repository\Packs\STMicroelectronics\X-CUBE-TOF1\2.0.0\Projects\NUCLEO-F401RE\Examples\CUSTOM\VL53L3CX_SimpleRanging\Binary.

To begin testing, open the Tera Term and set the baud rate to 460800 as shown below.

Figure 11. **Tera Term: serial port setup**

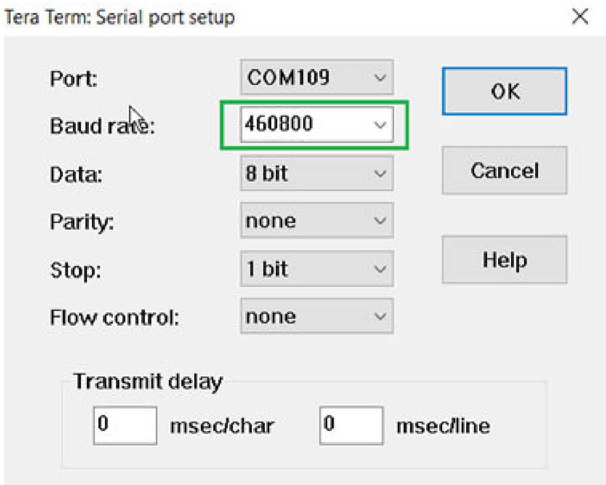
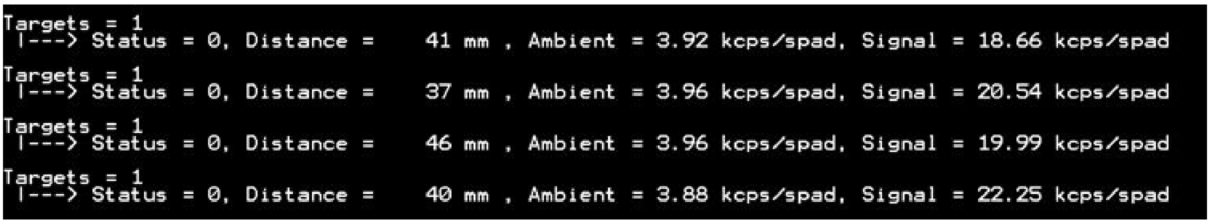


Figure 12. **Ranging result displayed on a terminal**



4 VL53L3CX configuration steps

The X-NUCLEO-53L3A2 interfaces with the STM32 microcontroller via the I2C pin. If a user wants to interface the X-NUCLEO-53L3A2 expansion board with an STM32 Nucleo 64 pins board (for example, a NUCLEO-F401RE), no particular hardware modification must be done. The X-NUCLEO-53L3A2 pin out is shown in Figure 14. X-NUCLEO-53L3A2 pinout.

Figure 13. STM32 Nucleo 64 pins and X-NUCLEO-53L3A2

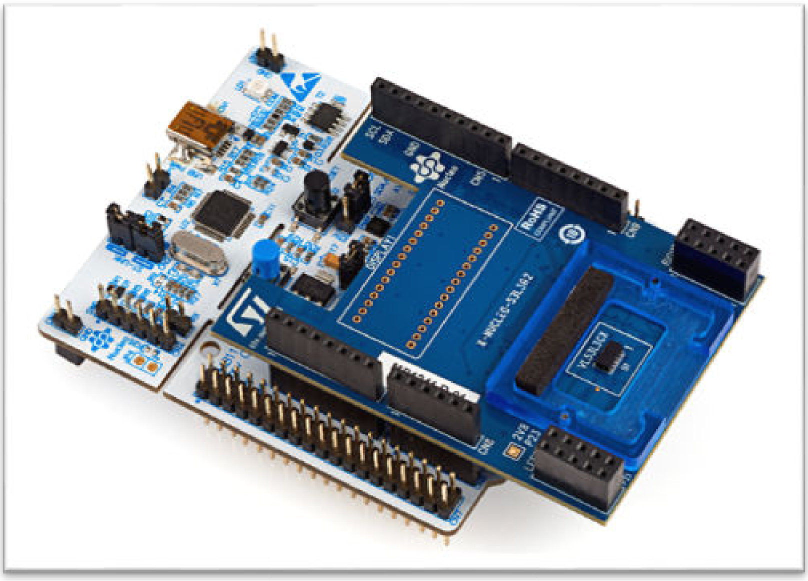
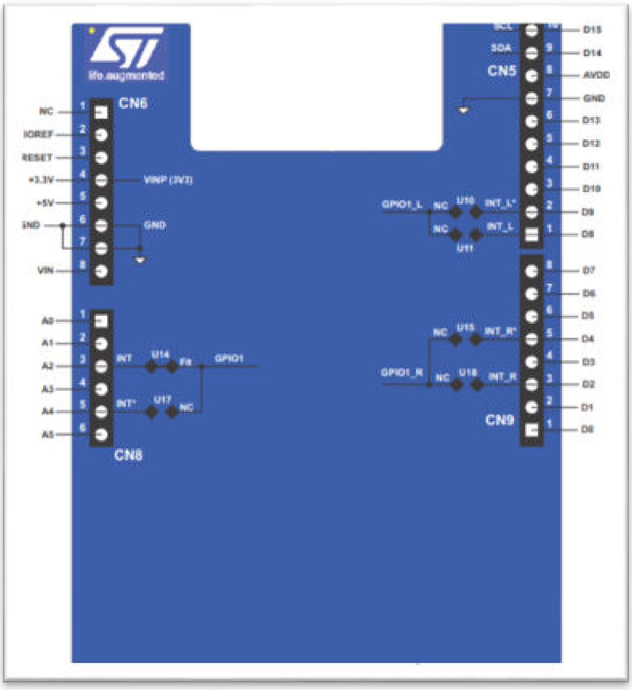


Figure 14. X-NUCLEO-53L3A2 pinout

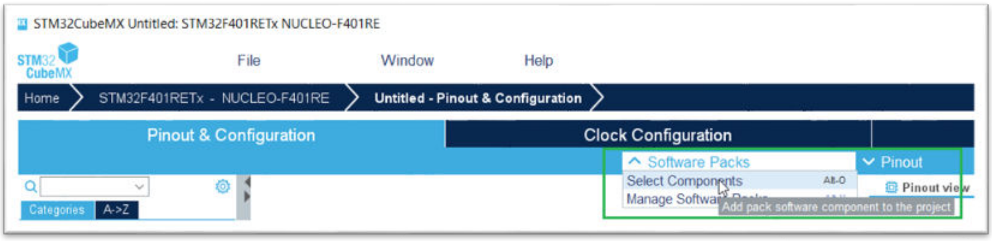


4.1 Use of expansion software without sample applications

This section describes how to configure STM32CubeMX with the X-NUCLEO-53L3A2 when the use of the sample applications is not required. With such a setup, only the driver layers are configured.

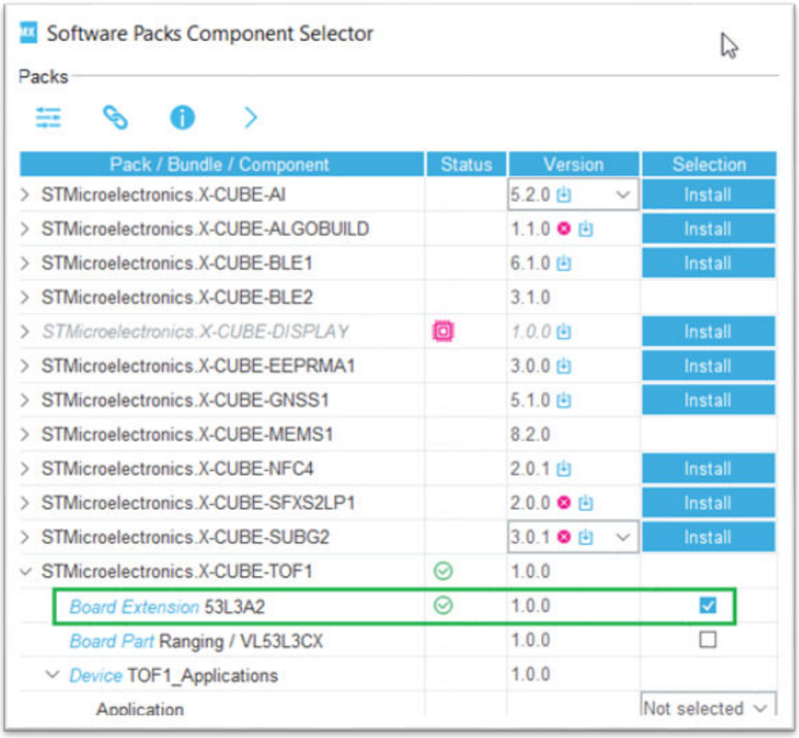
1. To add the X-CUBE-TOF1 SW pack to the project, click on the [**Software Packs**] button then [**Select Components**].

Figure 15. Select components



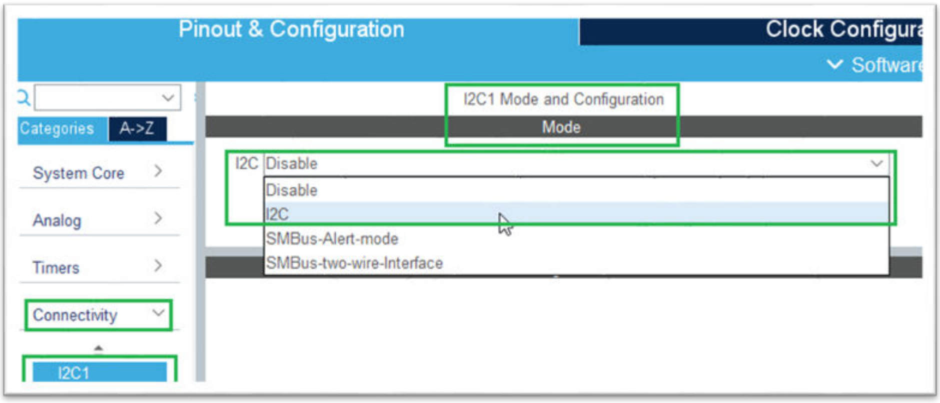
2. From the [**Software Packs Component Selector**] window, select only the [**Board Extension**] class.

Figure 16. Select board extension only



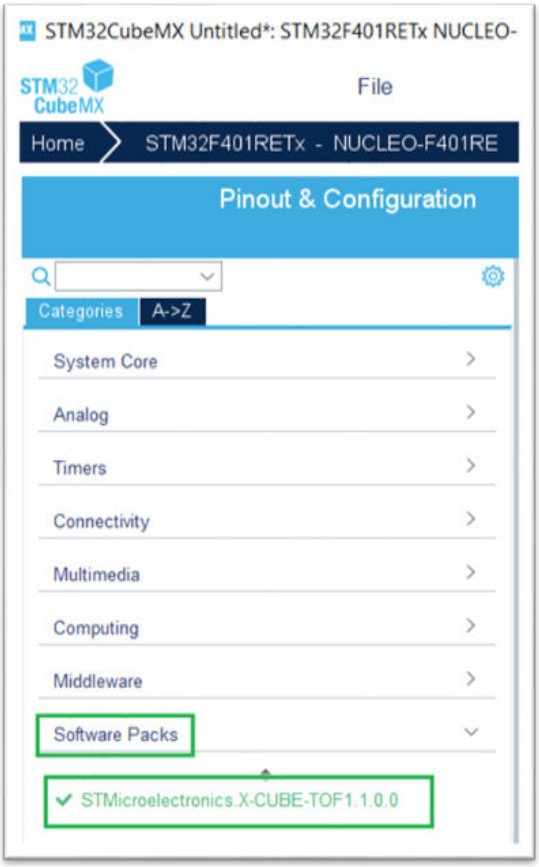
3. Enable I2C1 as shown below.

Figure 17. I2C configuration



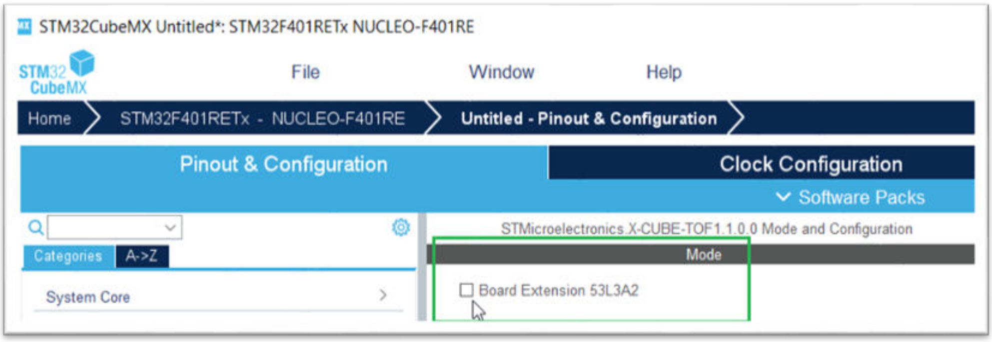
4. From the [**Software Packs**] drop-down menu, select [**STMicroelectronics.X-CUBE-TOF1**].

Figure 18. Software packs



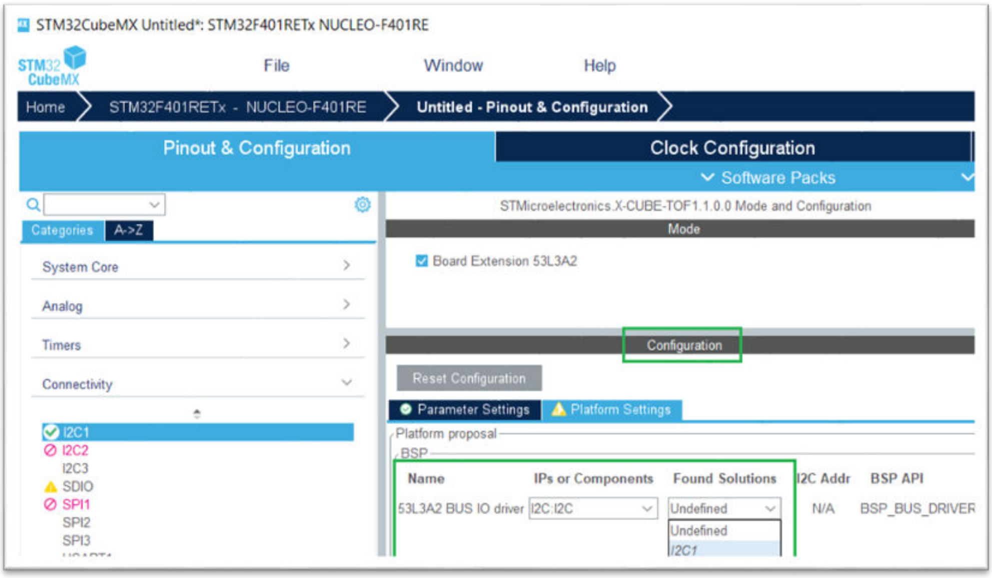
5. From the [**Mode**] view, select the [**Board Extension 53L3A2**].

Figure 19. **Mode view**



6. From the [**Configuration window**], enable the I2C1.

Figure 20. **Configuration window**



7. Once all the steps above have been performed, click on [**Project Manager**] to name the project and select the Toolchain/IDE for which codes to generate.

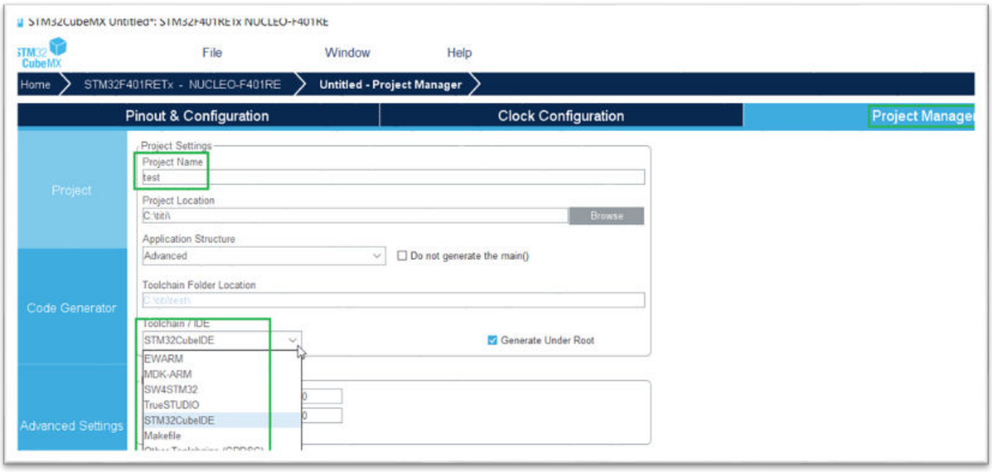


Figure 21. Project manager

8. Generate the source code of the project using the X-CUBE-TOF1 software by clicking on the [**GENERATE CODE**] button.

Figure 22. Generate code

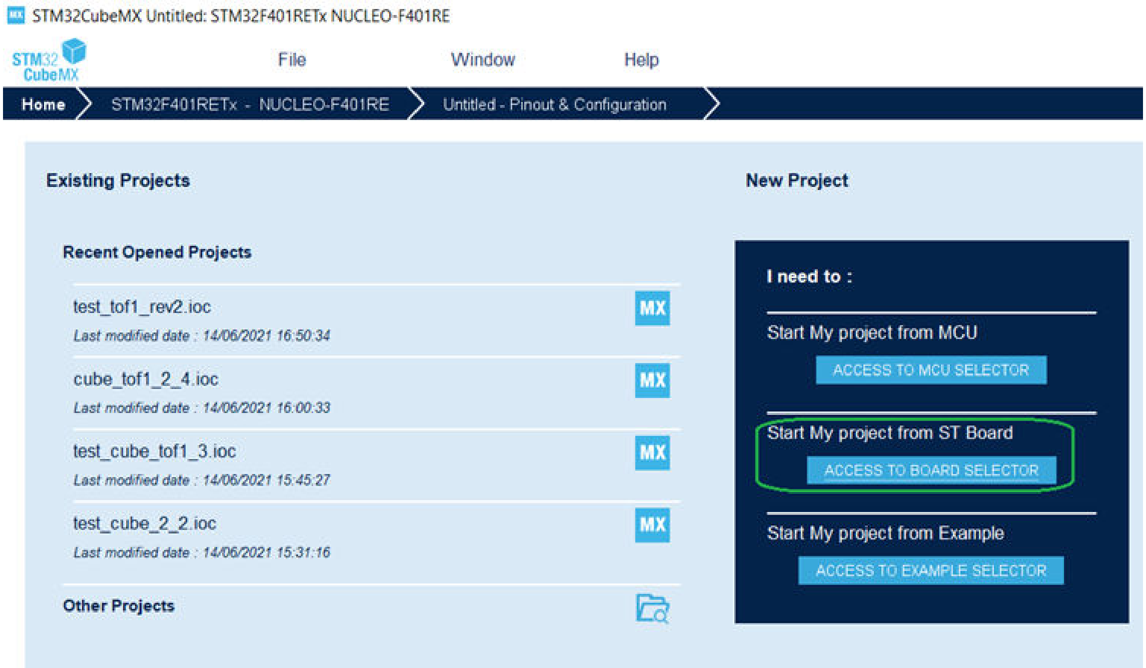
4.2 Use of expansion software with sample applications

This section describes how to configure STM32CubeMX with X-NUCLEO-53L3A2 when the use of the sample applications is desired. With such a setup, all the components of the expansion software package, including applications, are properly configured.

4.2.1 How to generate the 53L3A2_SimpleRanging example with CubeMX

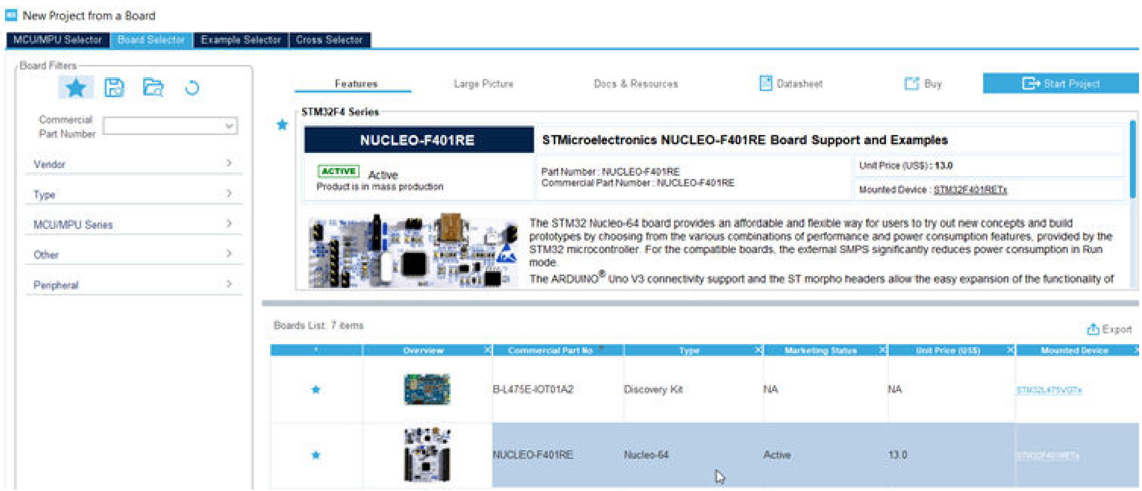
1. Open STM32-CubeMX and click on [**ACCESS TO BOARD SELECTOR**].

Figure 23. Access to board selector



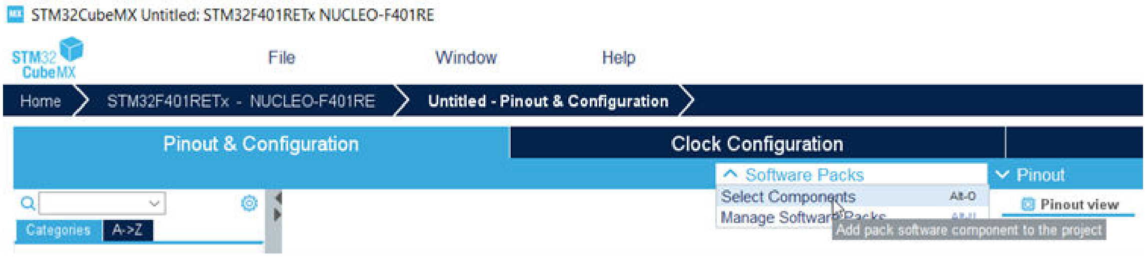
2. Search and select the F401RE board.

Figure 24. F401RE board



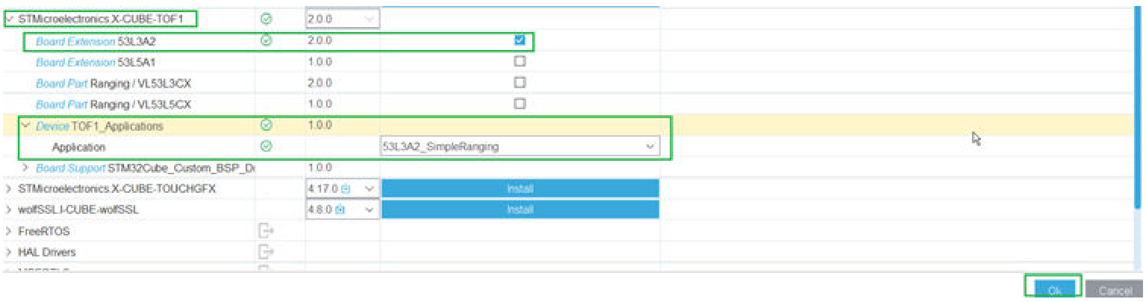
3. Click on [**Select Components**].

Figure 25. Select components



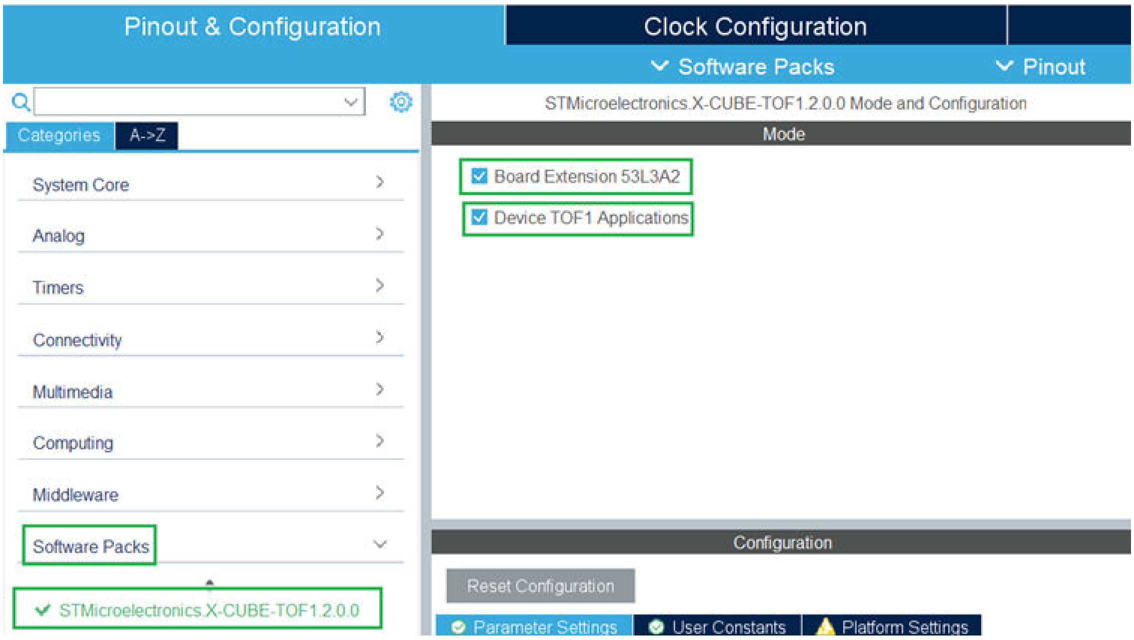
4. Click on [**X-CUBE-TOF1**]. Select [**53L3A2 Board Extension**], then select [**53L3A2_SimpleRanging**]. Click OK.

Figure 26. 53L3A2_SimpleRanging



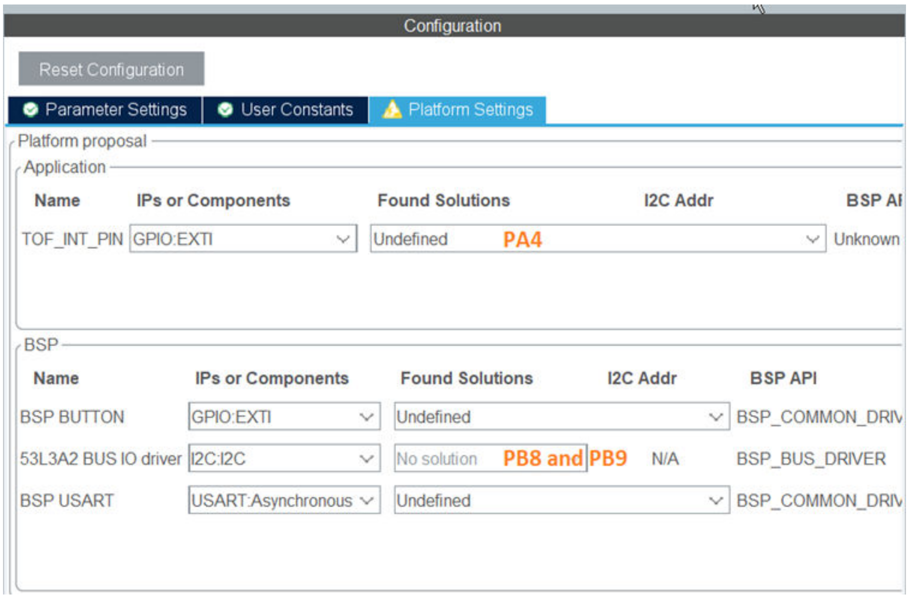
5. Click on [**Software Packs**], select [**STMicroelectronics X-CUBE-TOF1**], select the [**Board Extension 53L3A2**] box, then select the [**Device TOF1 Applications**] box.

Figure 27. **Device TOF1 applications**



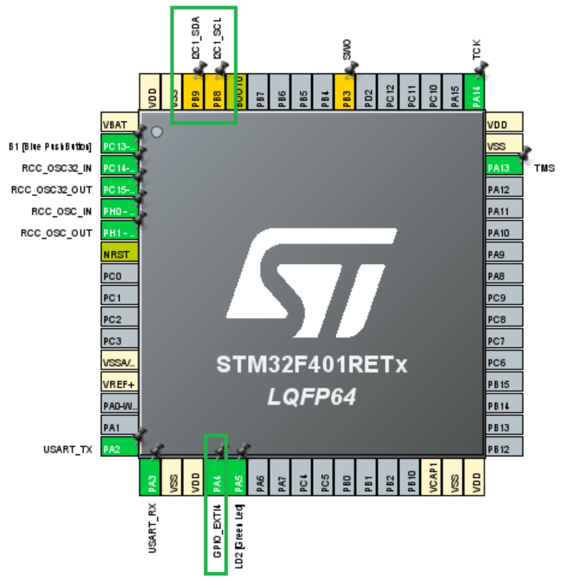
6. Configure the GPIOs for the application.

Figure 28. **GPIO configuration**



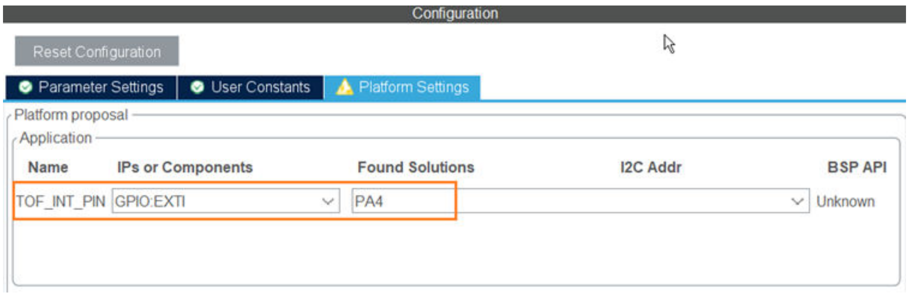
7. Select the GPIO pins.

Figure 29. **GPIO pin selection**



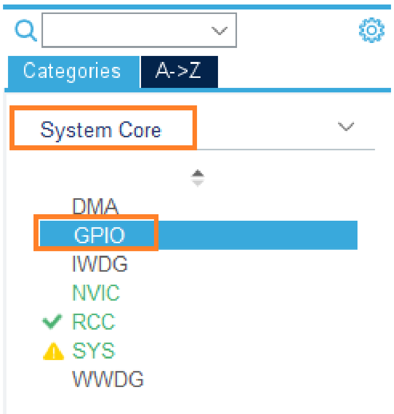
8. Link the GPIOs to the corresponding pin names.

Figure 30. **GPIO and pin name correspondence**



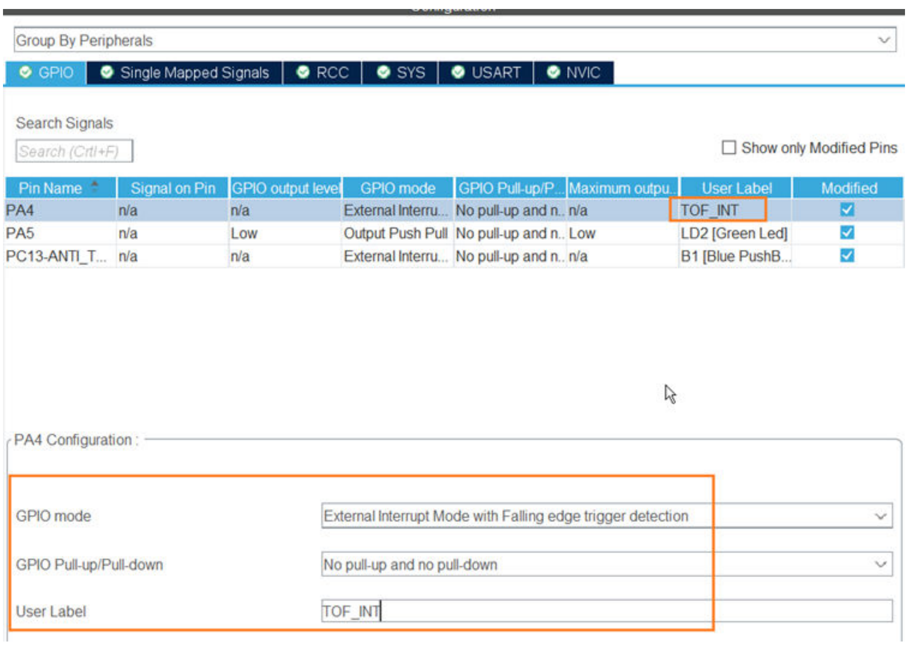
9. Click on GPIO to open the GPIO configuration window.

Figure 31. GPIO configuration window



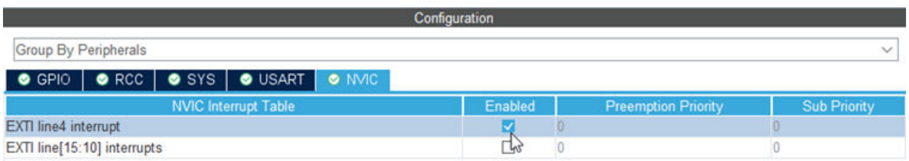
10. Name and configure the pins as shown below.

Figure 32. Pin names and configuration



11. Activate the NVIC interrupt vector as shown below.

Figure 33. Activation of NVIC interrupt vector



12. Configure the I2C and BSP

Figure 34. Configuration of I2C and BSP

13. Select PB9 and PB8 for SDA and SCL.

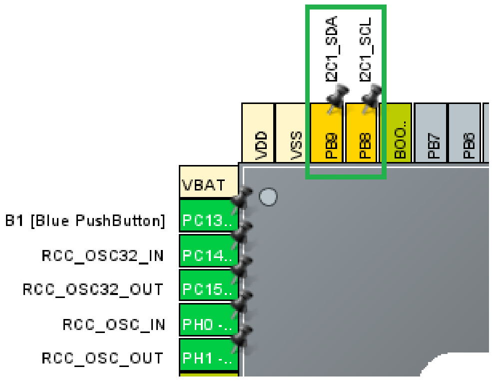


Figure 35. PB9 and PB8 selection (for SDA and SCL)

14. Click on [**Connectivity**]. Select [**I2C1**], enable the I2C, and select [**Fast Mode**].

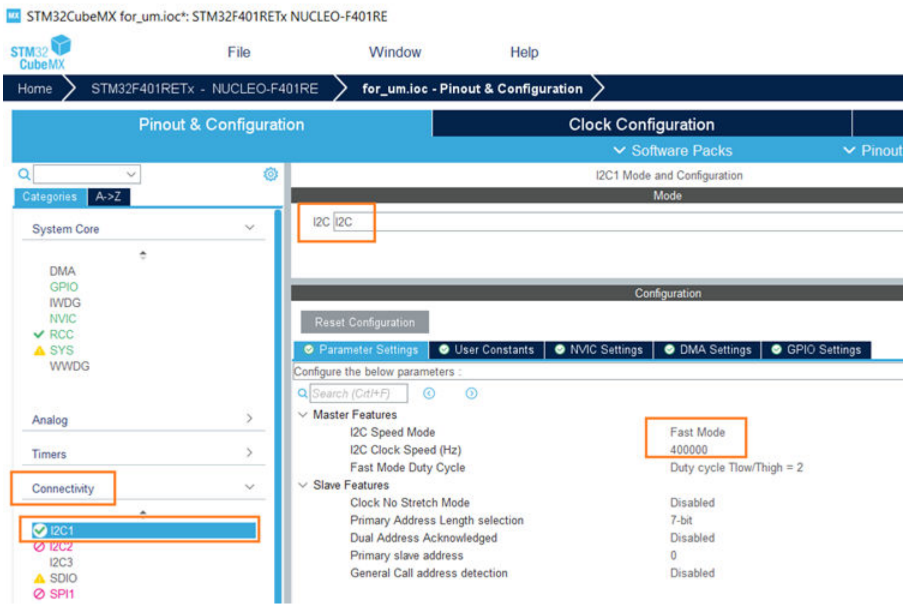
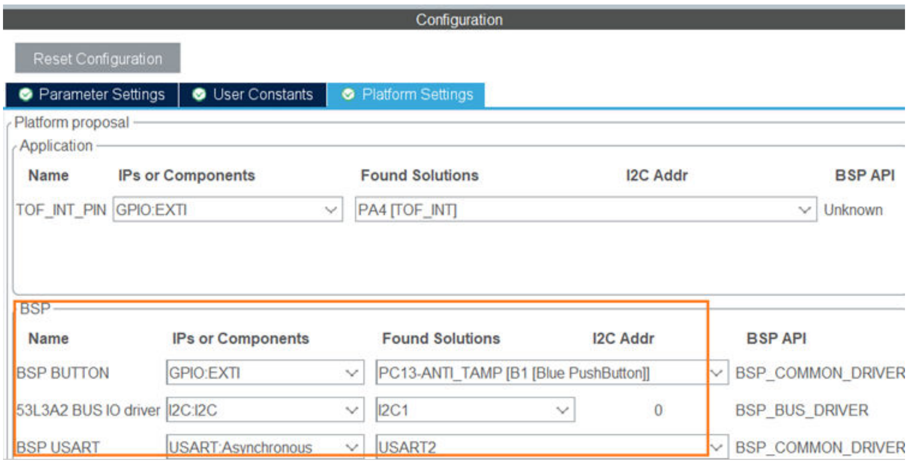


Figure 36. Fast mode selection

15. Return to the [**Software Pack**] view and configure the I2C and BSP as shown below.

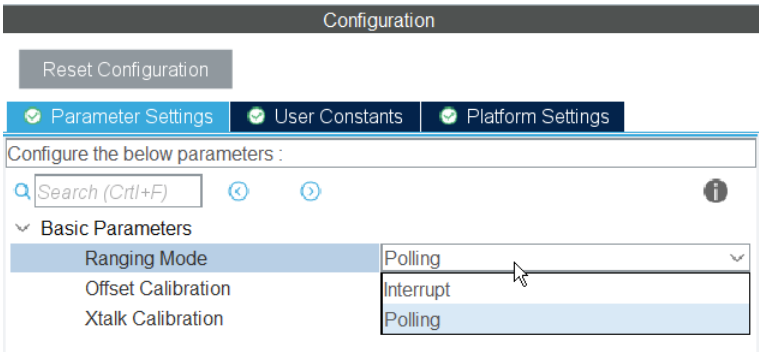
Figure 37. Configuration of I2C and BSP



Note: The ranging distance data can be read by polling a register or triggering an interrupt on pin PA4.

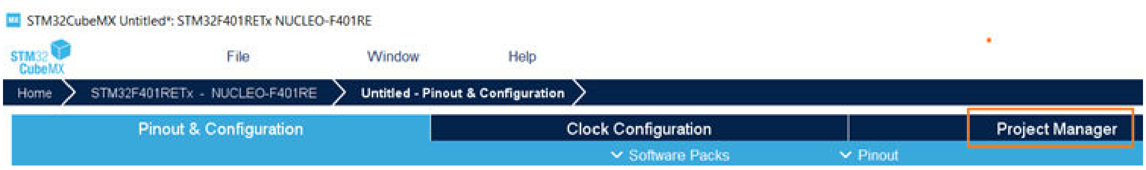
16. Select either polling or interrupt. By default, polling is selected.

Figure 38. Selection of polling or interrupt



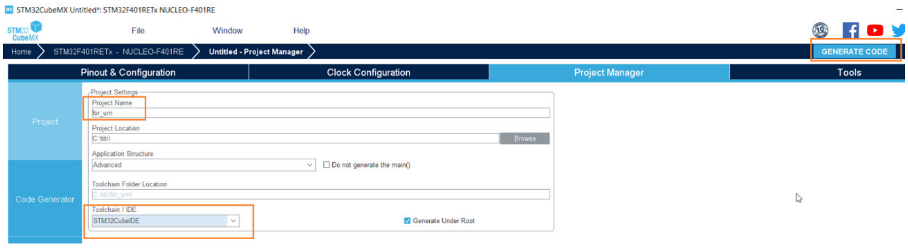
17. Click on [**Project Manager**].

Figure 39. Project manager



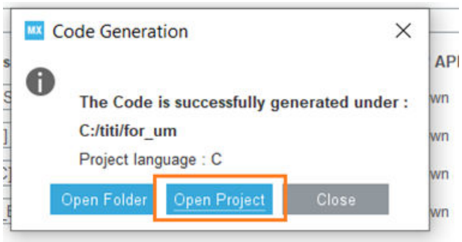
18. Name the project by selecting [**Toolchain**] and then selecting [**Generate Code**].

Figure 40. Project name



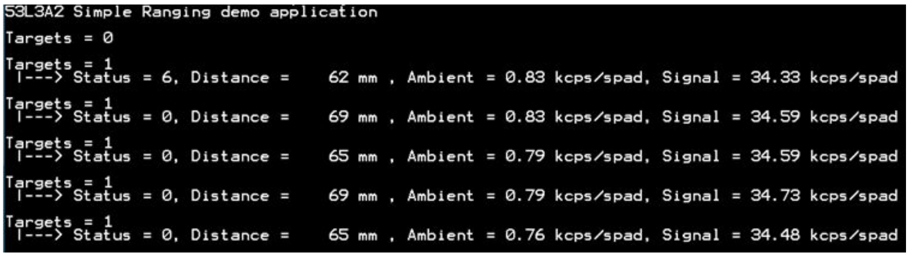
19. Click [**Open Project**] on the pop-up window when code generation is complete.

Figure 41. Open the project



20. Build and run the project. The results should look as shown below.

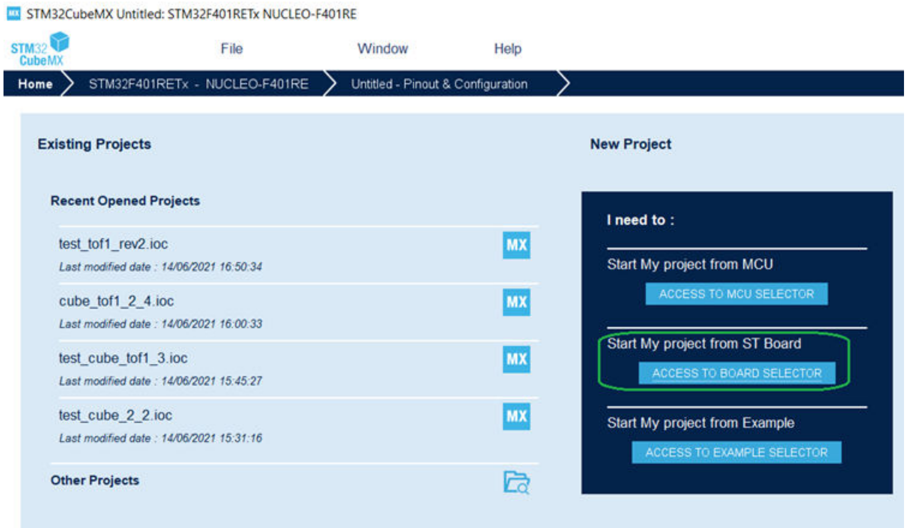
Figure 42. Build and run the project



4.2.2 How to generate the 53L3A2_MultipleSensorRanging example with CubeMX

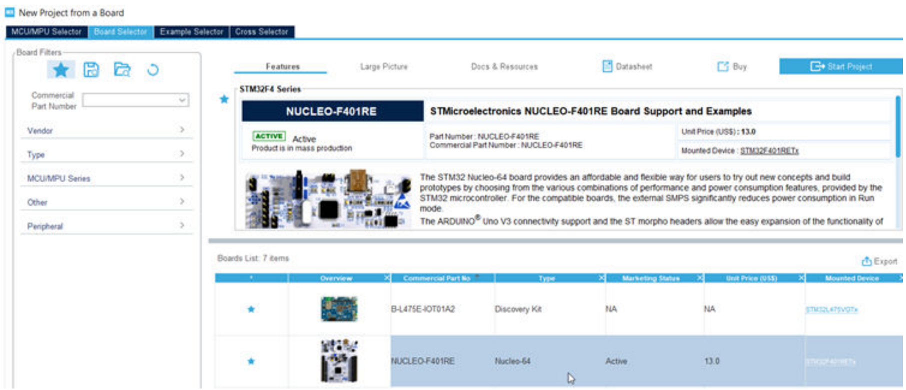
1. Open STM32-CubeMX and click on [**ACCESS TO BOARD SELECTOR**].

Figure 43. Access to board selector



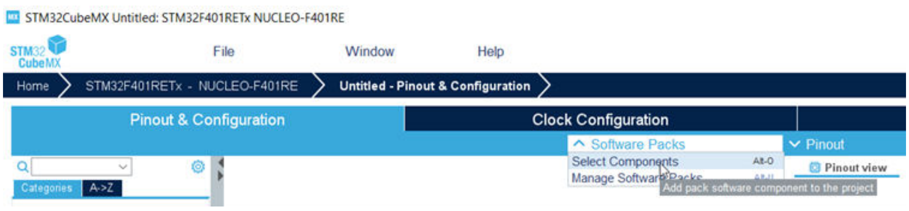
2. Search and select the F401RE board.

Figure 44. F401RE board



3. Click on [**Select Components**].

Figure 45. Select components

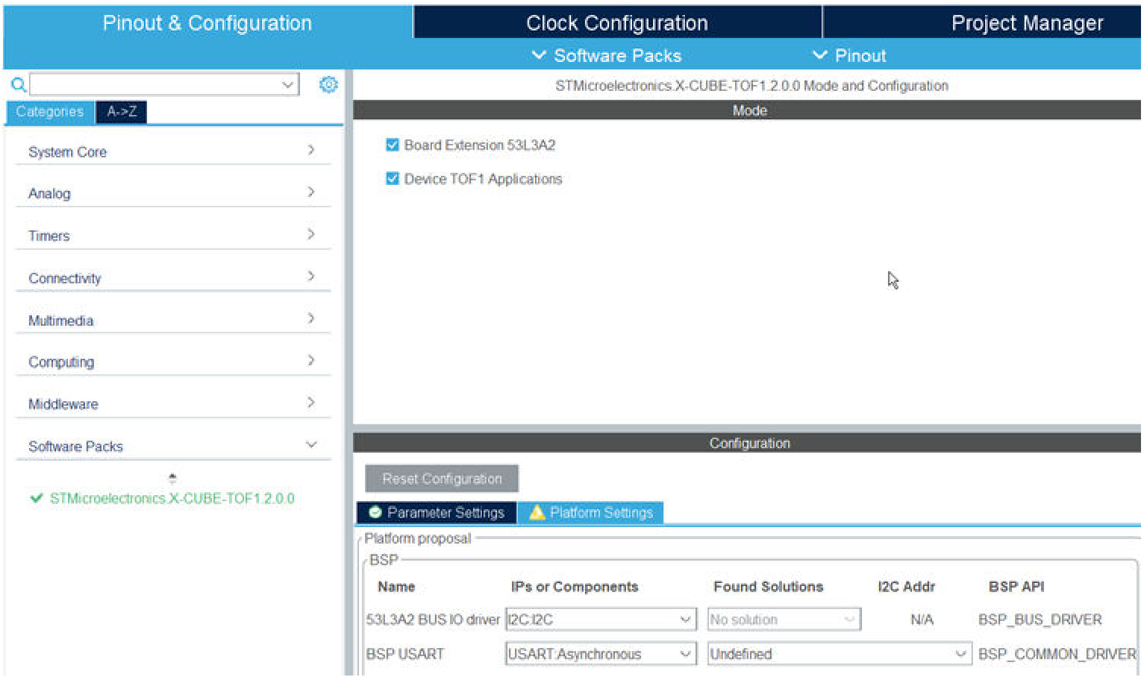


4. Click on [**X-CUBE-TOF1**]. Select [**53L3A2 Board Extension**] then select [**53L3A2_MultiSensorRanging**]. Click [**OK**] (in the bottom right-hand corner).

Figure 46. **53L3A2_MultiSensorRanging**

5. Click on [**Software Packs**]. Select [**STMicroelectronics X-CUBE-TOF1**], select the [**Board Extension 53L3A2**] box, and then select the [**Device TOF1 Applications**] box.

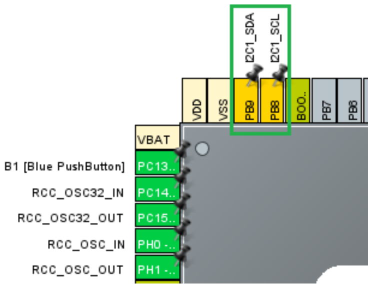
Figure 47. **Device TOF1 applications**



Note: Only the I2C is needed to setup.

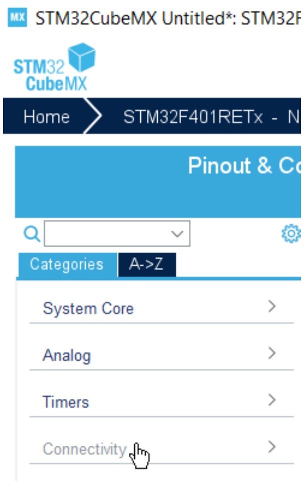
6. Select PB9 and PB8 for SDA and SCL.

Figure 48. PB9 and PB8 selection (for SDA and SCL)



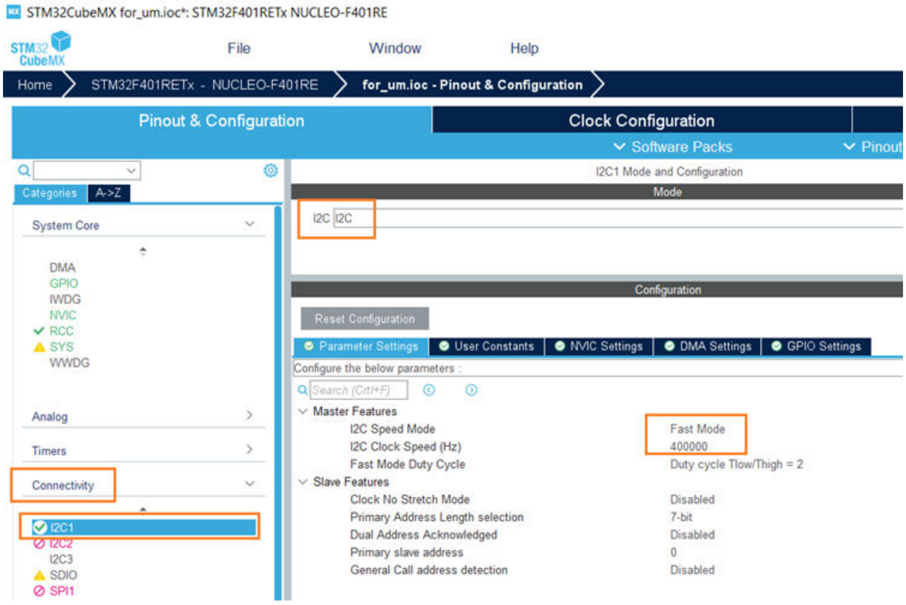
7. Click on [**Connectivity**].

Figure 49. Connectivity



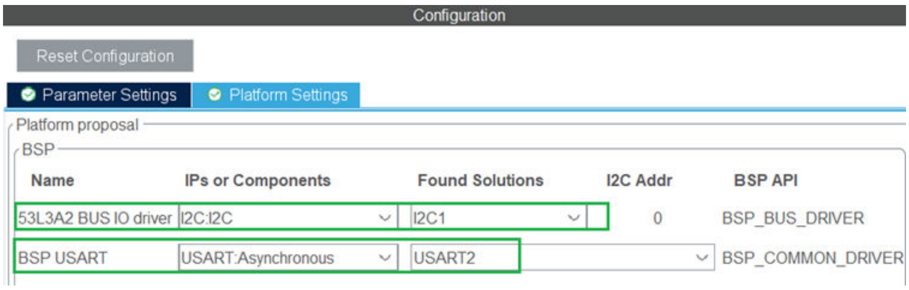
8. Select[**I2C1**]. Enable the I2C and select [**Fast mode**].

Figure 50. Fast mode selection



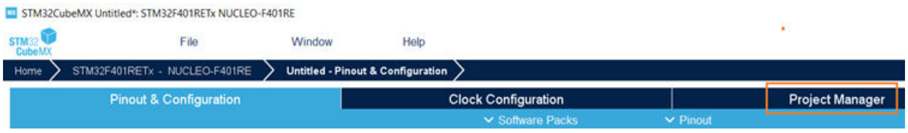
9. Return to the [**Software Pack**] view and configure the I2C and BSP as shown below.

Figure 51. Configuration of I2C and BSP



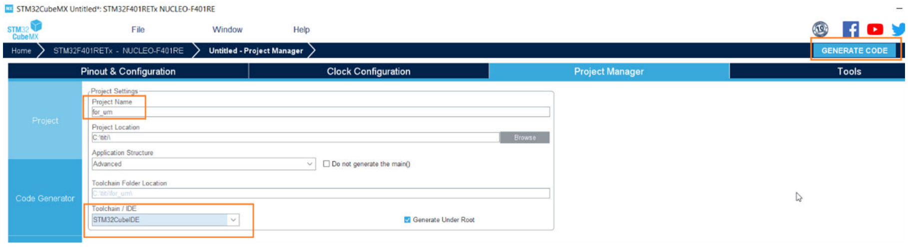
10. Click on [**Project Manager**].

Figure 52. Project manager



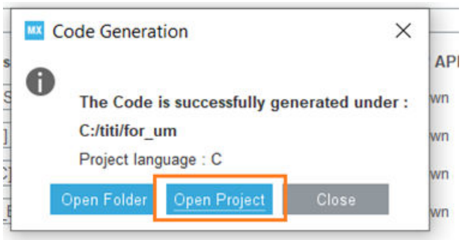
11. Name the project by selecting [**Toolchain**] and then selecting [**Generate Code**].

Figure 53. Project name



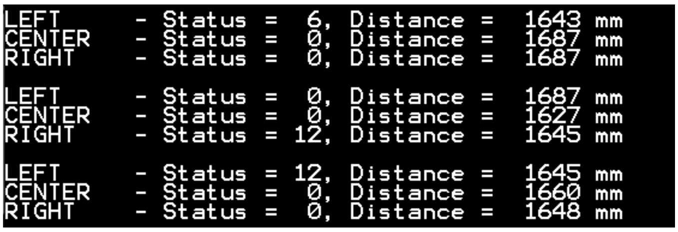
12. Click [**Open Project**] on the pop-up window when the code generation is complete.

Figure 54. Open the project



13. Build and run the project. The results should look as shown below.

Figure 55. Build and run the project



4.2.3 How to generate the VL53L3_SimpleRanging example with CubeMX

In this example, the following material is required:

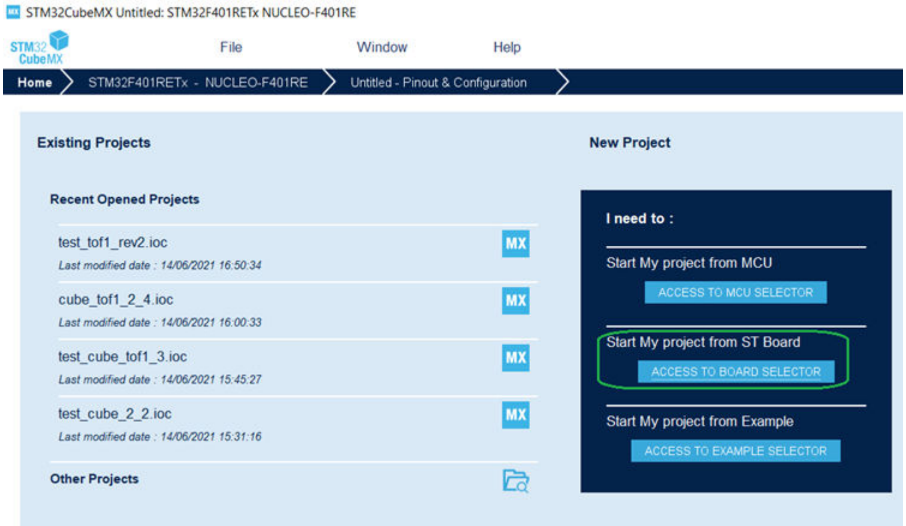
- A NUCLEO-F401RE
- VL53L3CX-SATEL
- Dupont wires

Note: Only the green VL53L3CX-SATEL PCB version works. The blue PCB cannot be used in this example.

To operate this example, the breakout board is connected directly to the NUCLEO-F401RE board without the X-NUCLEO-53L3A2 expansion board.

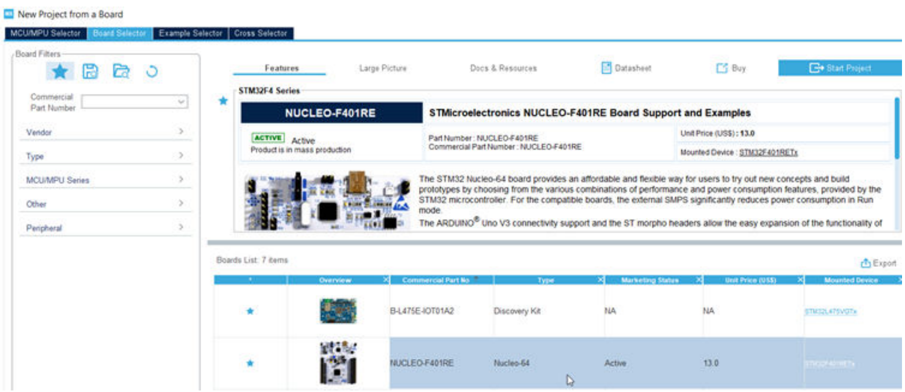
1. Open STM32-CubeMX and click on [**ACCESS TO BOARD SELECTOR**].

Figure 56. Access to board selector



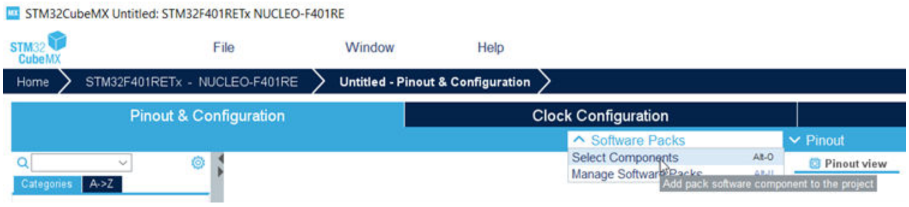
2. Search and select the F401RE board.

Figure 57. F401RE board



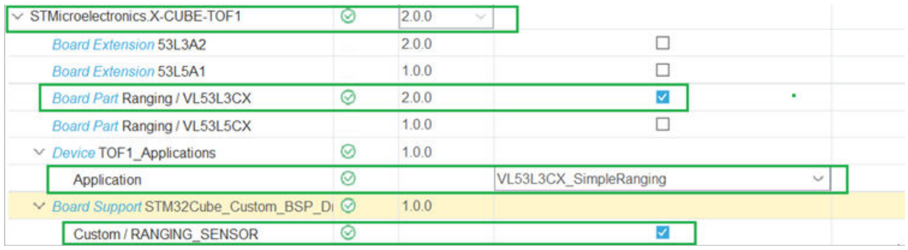
3. Click on [**Select Components**].

Figure 58. Select components



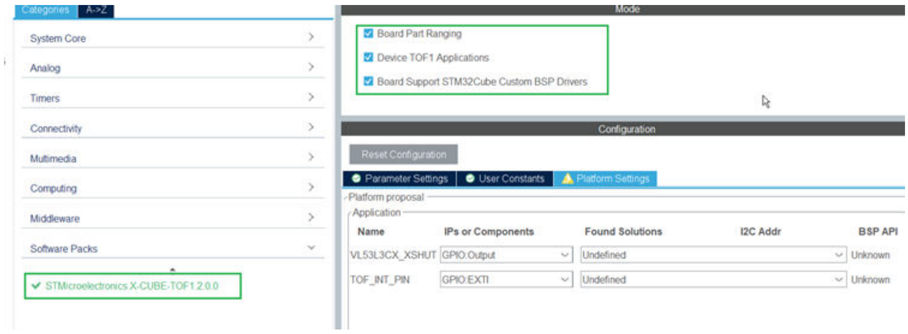
4. Click on [**X-CUBE-TOF1**]. Select [**53L3A2 Board Extension**] and then select [**VL53L3CX_SimpleRanging**]. Click [**OK**].

Figure 59. VL53L3CX_SimpleRanging



5. Click on [**Software Packs**]. Select [**STMicroelectronics X-CUBE-TOF1**], [**Board Part Ranging**] box, then [**Device TOF1 Applications**] and [**Board Support STM32Cube Custom BSP Drivers**].

Figure 60. Board support STM32Cube custom BSP drivers



6. Implement the connections shown below.

Figure 61. Connections 1

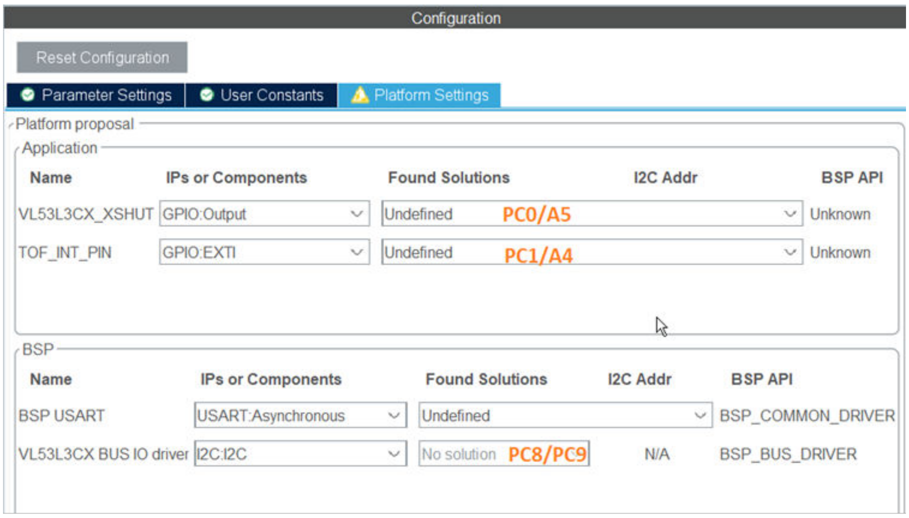
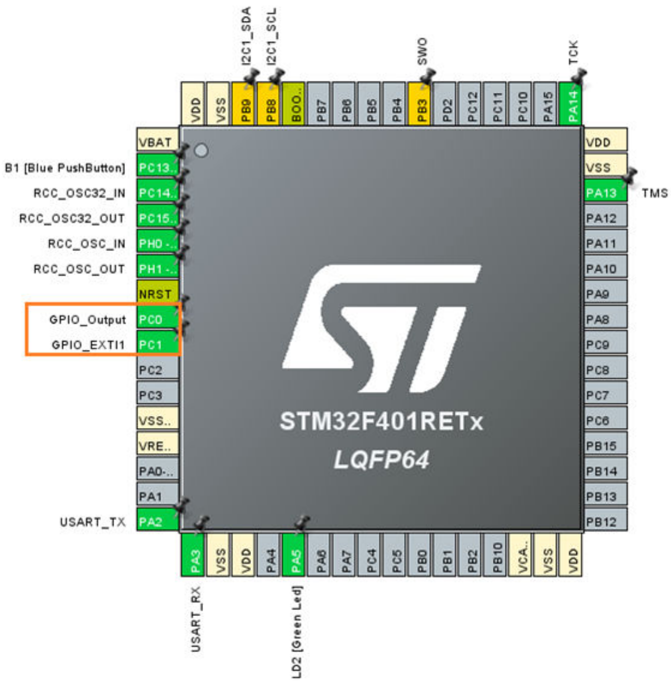
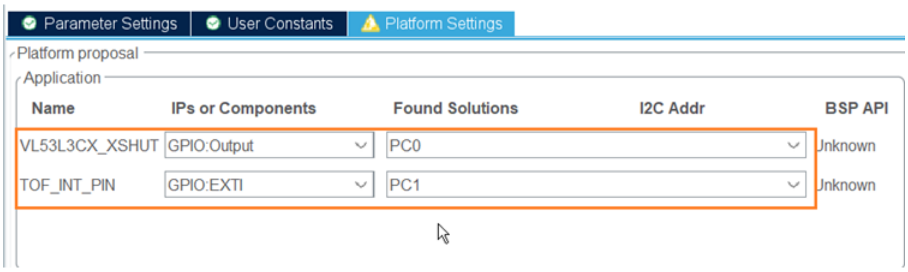


Figure 62. Connections 2



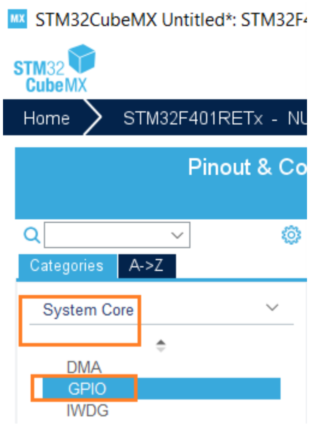
7. Link the GPIOs to the corresponding pin names.

Figure 63. GPIO and pin name correspondence



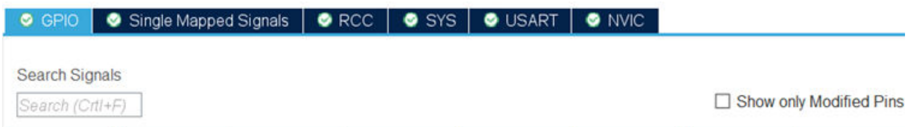
8. Click on [**System Core**], then on [**GPIO**] to open the GPIO configuration window.

Figure 64. GPIO configuration window



9. Name and configure the GPIO pins as shown below.

Figure 65. GPIO pin name and configuration



Pin Name	Signal on Pin	GPIO output le...	GPIO mode	GPIO Pull-up/...	Maximum outp...	User Label	Modified
PA5	n/a	Low	Output Push P...	No pull-up and ...	Low	LD2 [Green L...	✓
PC0	n/a	High	Output Push P...	Pull-up	Low	TOF_RST	✓
PC1	n/a	n/a	External Interru...	No pull-up and ...	n/a	TOF_INT	✓
PC13-ANTI_T...	n/a	n/a	External Interru...	No pull-up and ...	n/a	B1 [Blue Push...	✓

10. Activate the NVIC interrupt vector as shown below.

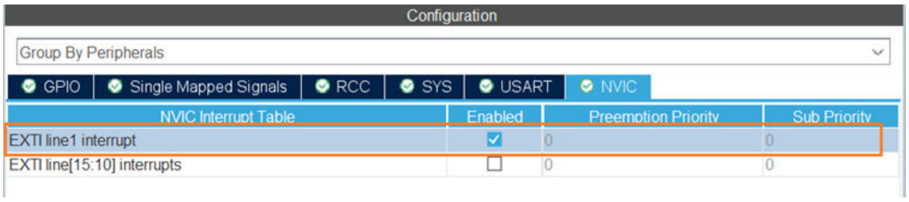


Figure 66. Activation of NVIC interrupt vector

11. Configure the I2C and BSP

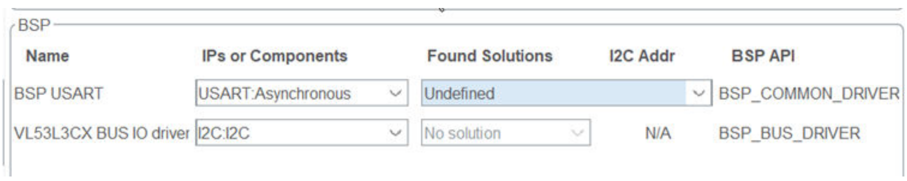


Figure 67. Configuration of I2C and BSP

12. Select PB9 and PB8 for SDA and SCL.

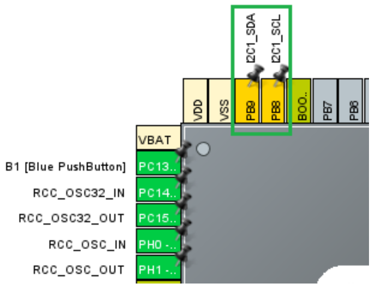
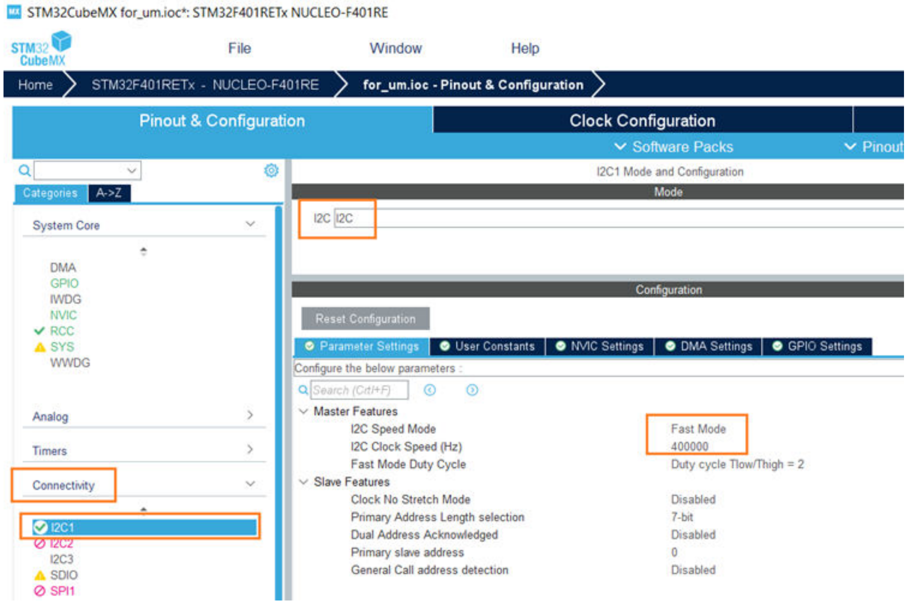


Figure 68. PB9 and PB8 selection (for SDA and SCL)

13. Click on [**Connectivity**]. Select [**I2C1**] and enable the I2C, then select [**Fast mode**].

Figure 69. **Fast mode selection**

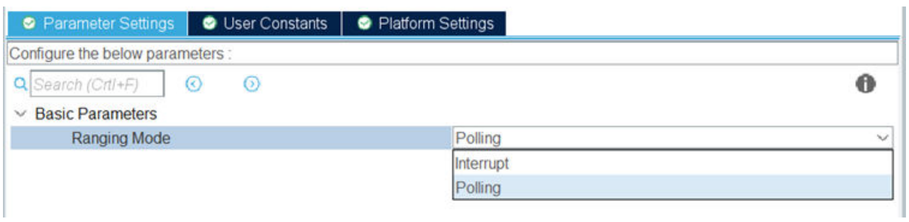


14. Return to the [**Software Pack**] view and configure the I2C and BSP as shown below.

Figure 70. **Configuration of I2C and BSP**

15. Select either polling or interrupt. By default, [**Polling**] is selected.

Figure 71. **Selection of polling or interrupt**



16. Click on [**Project Manager**].

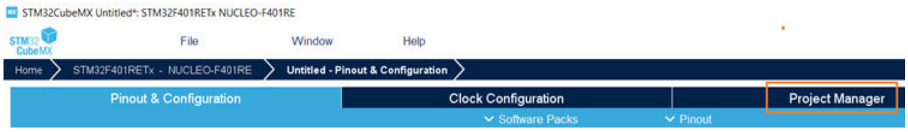


Figure 72. Project manager

17. Name the project by selecting [**Toolchain**] and then selecting [**Generate Code**].

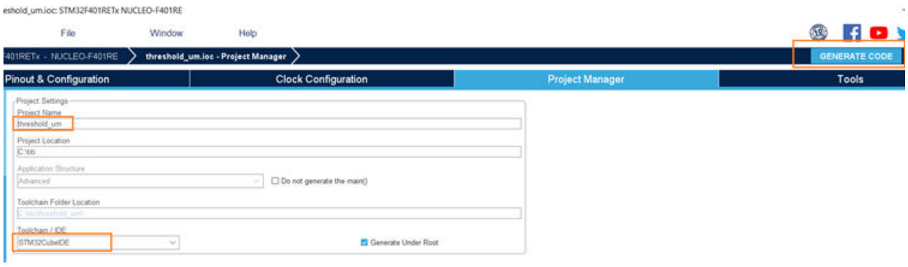


Figure 73. Project name

18. Click [**Open Project**] on the pop-up window when the code generation is complete.

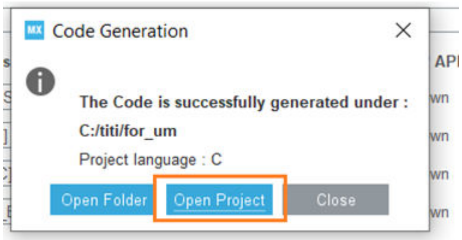


Figure 74. Open the project

19. Build and run the project. The results should look as shown below.

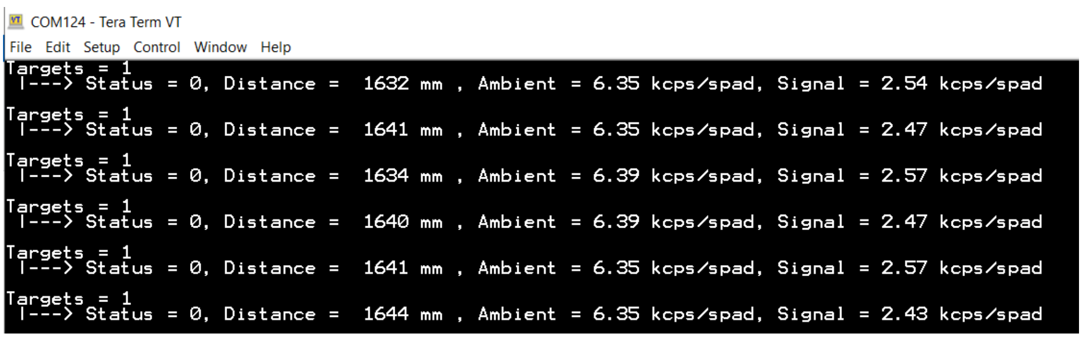


Figure 75. Build and run the project

5 System setup guide

5.1 Hardware description

5.1.1 STM32 Nucleo

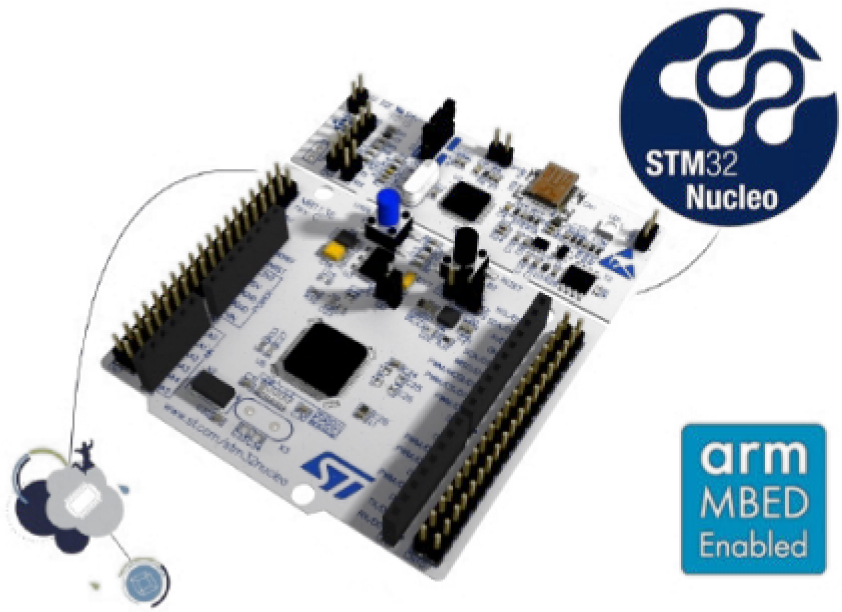
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

The Arduino® connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from. The STM32 Nucleo board does not require separate probes as it integrates the ST-LINK/V2-1 debugger/ programmer.

The STM32 Nucleo board comes with the comprehensive STM32 software HAL library together with various packaged software examples for different IDEs (IAR EWARM, Keil® MDK-ARM, STM32CubeIDE, Mbed™, and GCC/LLVM).

All STM32 Nucleo users have free access to the Mbed™ online resources (compiler, C/C++ SDK and developer community) at www.mbed.org to build complete applications easily.

Figure 76. STM32 Nucleo board



Information regarding the STM32 Nucleo board is available at www.st.com/stm32nucleo.

5.1.2 VL53L3CX boards

5.1.2.1 X-NUCLEO-53L3A2 expansion board

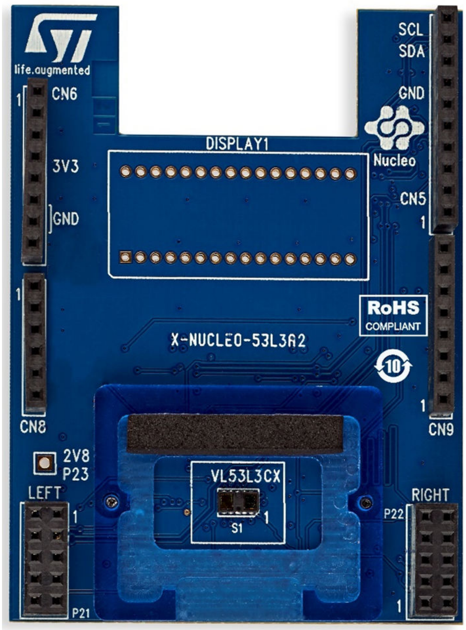
The X-NUCLEO-53L3A2 is an expansion board for any Nucleo 64 development board. It provides a complete evaluation kit allowing anyone to learn, evaluate, and develop their applications using the VL53L3CX, ranging sensor with multitarget detection.

The X-NUCLEO-53L3A2 expansion board is delivered with a cover glass holder in which three different spacers of 0.25, 0.5, and 1 mm height can be fitted below the cover glass to simulate various air gaps.

Two VL53L3CX breakout boards can be connected using two 10-pin connectors.

The X-NUCLEO-53L3A2 expansion board is compatible with the STM32 Nucleo board family, and with the Arduino® UNO R3 connector layout.

Figure 77. X-NUCLEO-53L3A2 expansion board



5.1.2.2 *VL53L3CX-SATEL breakout boards*

The VL53L3CX-SATEL breakout boards can be used for easy integration into customer devices.

Thanks to the voltage regulator and level shifters, the VL53L3CX breakout boards can be used in any application with a 2.8 V to 5 V supply.

The PCB section supporting the VL53L3CX module is perforated so that developers can break off the mini PCB for use in a 2.8 V supply application using flying leads. This makes it easier to integrate the VL53L3CX-SATEL breakout boards into development and evaluation devices due to their small form factor.

Figure 78. **VL53L3CX-SATEL breakout board**

5.2 Software description

The following software components are required in order to establish a suitable development environment for creating applications for the STM32 Nucleo equipped with the sensor expansion board:

- X-CUBE-TOF1: an STM32Cube expansion for sensor application development. The X-CUBE-TOF1 firmware and associated documentation is available on www.st.com.
- Development tool-chain and compiler: The STM32Cube expansion software supports the three following environments:
  - IAR Embedded Workbench for Arm® (EWARM) toolchain + STLINK
  - RealView microcontroller development kit (MDK-ARM-STR) toolchain + STLINK
  - STM32CubeIDE for STM32 + STLINK

5.3 Hardware setup

The following hardware components are required:

- One STM32 Nucleo development platform (suggested order code: NUCLEO-F401RE or NUCLEO-L476RG)
- An X-NUCLEO-53L3A2 expansion board or a VL53L3CX-SATEL breakout board
- One USB type A to mini-B USB cable to connect the STM32 Nucleo to a PC

5.4 Software setup

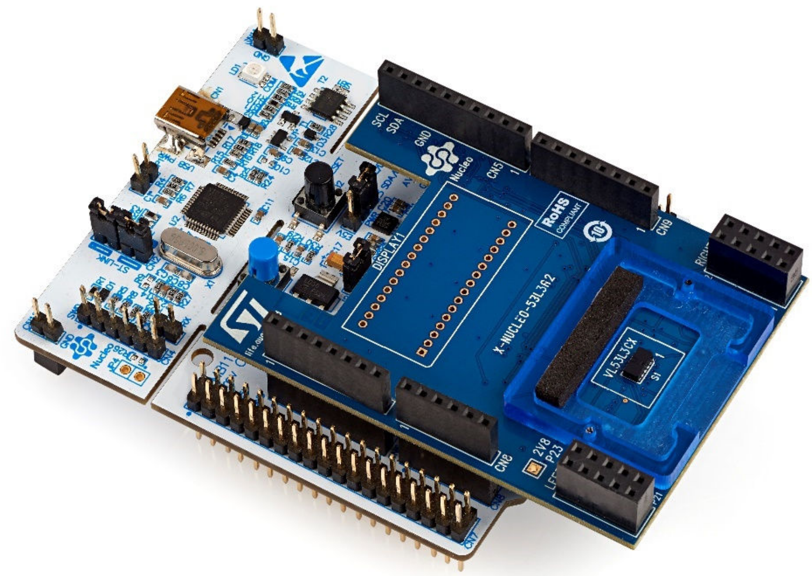
To set up the SDK, run the sample testing scenario based on the GUI utility and customize applications, select one of the integrated development environments supported by the STM32Cube expansion software and follow the system requirements and setup information provided by the IDE provider.

5.5 STM32 Nucleo and sensor expansion board setup

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. Developers can download the relevant version of the ST-LINK/V2-1 USB driver by searching STSW-LINK008 or STSW-LINK009 (depending on your version of Windows®) on www.st.com.

The X-NUCLEO expansion boards can be easily connected to the STM32 Nucleo board through the Arduino® UNO R3 extension connector. It can interface with the external STM32 microcontroller on the STM32 Nucleo via the inter-integrated circuit (I²C) transport layer.

Figure 79. **Sensor expansion board plugged to STM32 Nucleo board**



Revision history

Table 1. Document revision history

Date	Version	Changes
27-Oct-2021	1	Initial release
30-Jan-2023	2	Updated Figure 10. VL53L3CX_SATEL connection and added note after figure.

Contents