How to use the VL53L5CX 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 sensor evaluation boards connected to a featured STM32 Nucleo development board.

In this user manual, we focus on the VL53L5CX Time-of-Flight 8x8 multizone ranging sensor with wide field of view. For further information on the Time-of-Flight sensors supported by X-CUBE-TOF1, please refer to the software page of www.st.com.

The VL53L5CX evaluation boards supported by the X-CUBE-TOF1 expansion software package include:

- X-NUCLEO-53L5A1 expansion board
- VL53L5CX-SATEL breakout boards

The X-CUBE-TOF1 software provides the following sample applications for the VL53L5CX:

- 53L5A1_SimpleRanging for X-NUCLEO-53L5A1
- 53L5A1_MultiSensorRanging for X-NUCLEO-53L5A1 and VL53L5CX-SATEL
- 53L5A1_ThresholdDetection for X-NUCLEO-53L5A1
- VL53L5CX_SimpleRanging for VL53L5CX-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 VL53L5CX 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.
- Pre-compiled 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

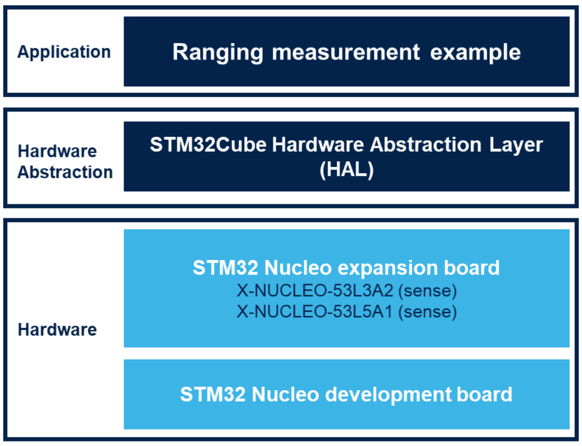
This software is a fully compliant expansion of STM32Cube enabling development of applications using Time-of-Flight sensors.

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 (e.g. 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 Time-of-Flight sensors and provides support for initializing and reading sensor data.

Figure 1. X-CUBE-TOF1 software architecture



2.3 Folder structure

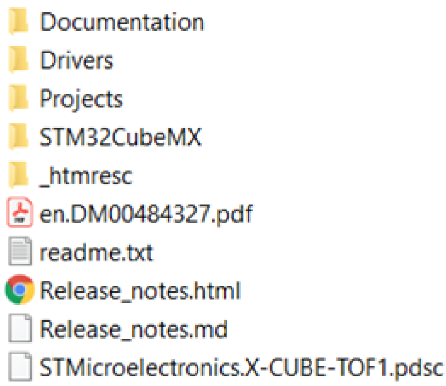


Figure 2. **X-CUBE-TOF1 package folder structure**

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.

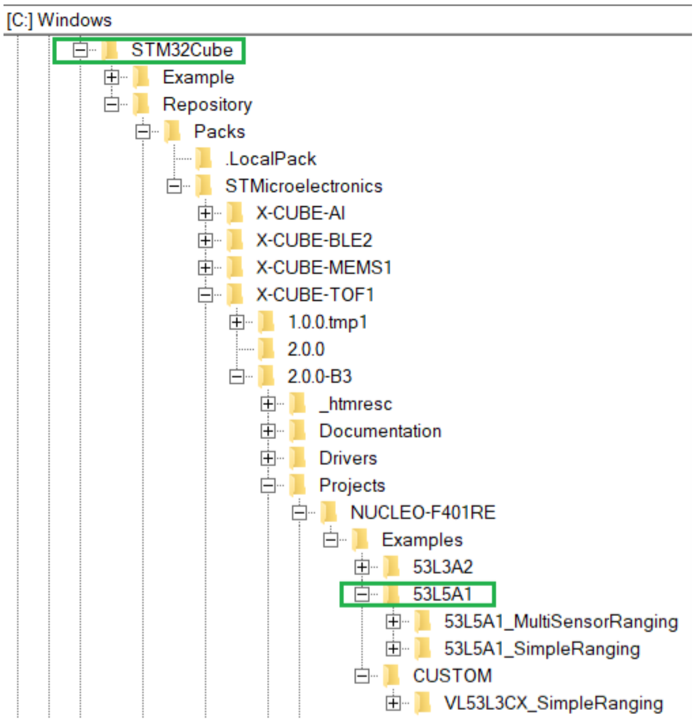
3 VL53L5CX sample application description

This sample application shows how to use the X-NUCLEO-53L5A1 expansion board and a STM32 Nucleo board 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.

Ranging data are read by polling the device for completion or by triggering an interrupt.

This application can be run by loading the prebuilt binary 53L5A1_SimpleRanging.bin at C:\Users\username\STM32Cube\Repository\Packs\STMicroelectronics\X-CUBE-TOF1\2.0.0-B3\Projects\NUCLEO-F401RE\Examples\53L5A1\53L5A1_SimpleRanging\Binary as shown in Figure 3. VL53L5CX precompiled projects. The STM32Cube directory is located at C:\Users\YourUserName\STM32Cube as shown below.

Figure 3. VL53L5CX precompiled projects



1. Flash the Nucleo F401RE board with the prebuilt binary.

2. Open the Tera Term and configure it with the settings below.

Figure 4. **Tera Term serial port setup**

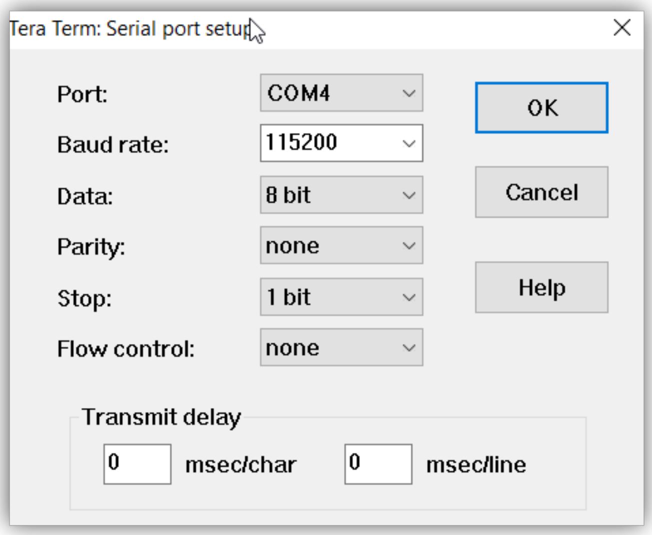
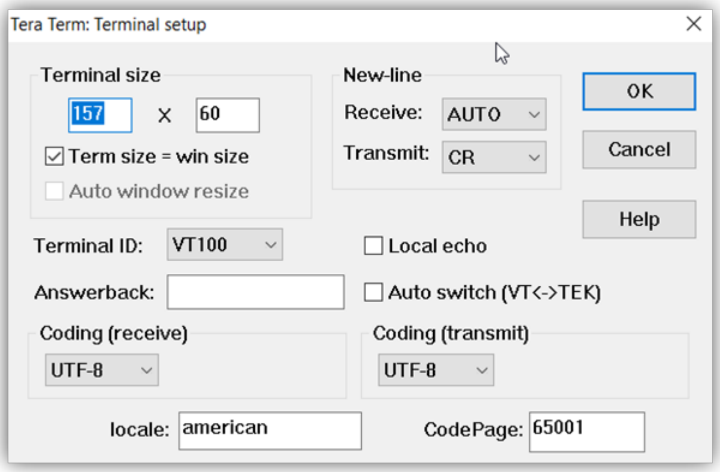
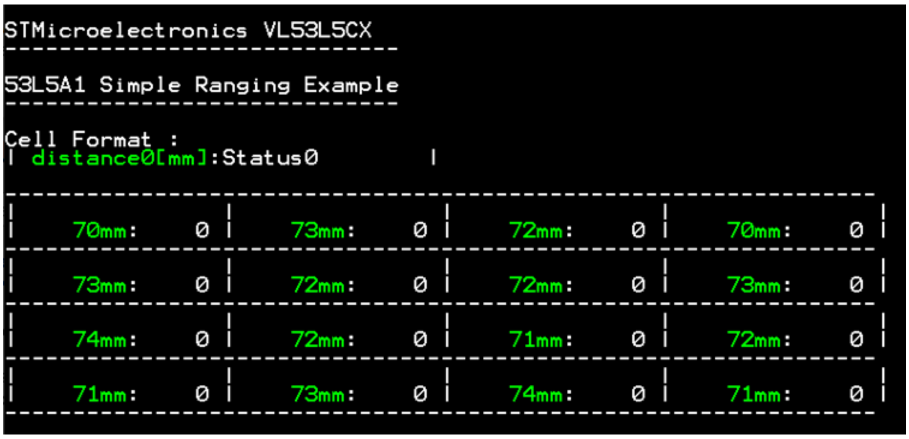


Figure 5. **Tera Term terminal setup**



3. Wave your hand in front of the sensor to display the ranging data on the serial terminal as shown below.

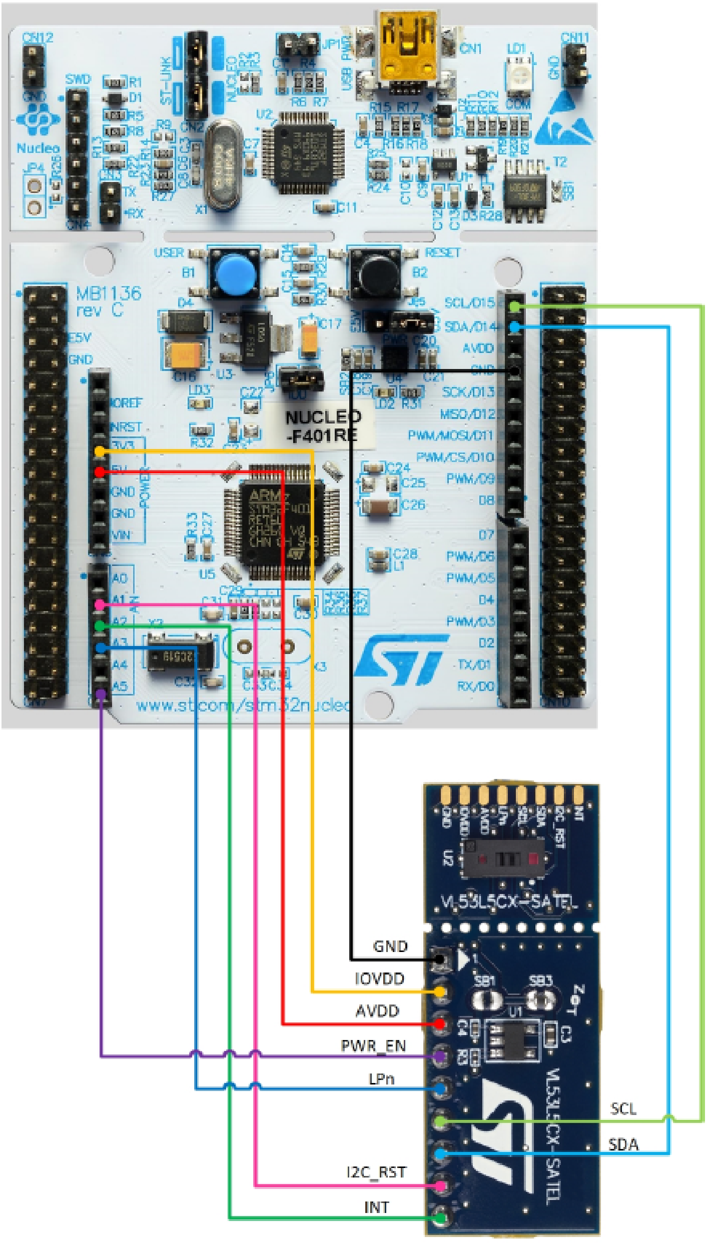
Figure 6. VL53L5CX ranging data



3.1 VL53L5CX_SimpleRanging

This sample application shows how to run the ranging distance with the VL53L5CX_SATEL (see figure below) connected directly to the Nucleo without the expansion board X-NUCLEO-53L5A1.

Figure 7. **VL53L5CX_SATEL wiring**



To test this application, a breakout board VL53L5CX-SATEL, and a F401RE Nucleo are required. In this example the ranging data are displayed on the serial terminal as shown below. This application can be run by flashing the Nucleo with the prebuilt binary VL53L5CX_SimpleRanging.bin from C:\Users\user_name\STM32Cube\Repository\Packs\STMicroelectronics\X-CUBE-TOF1\2.0.0\Projects\NUCLEO-F401RE\Examples\CUSTOM\VL53L5CX_SimpleRanging\Binary.

1. Open the Tera Term and set the baud rate to 460800

Figure 8. **Setting the baud rate**

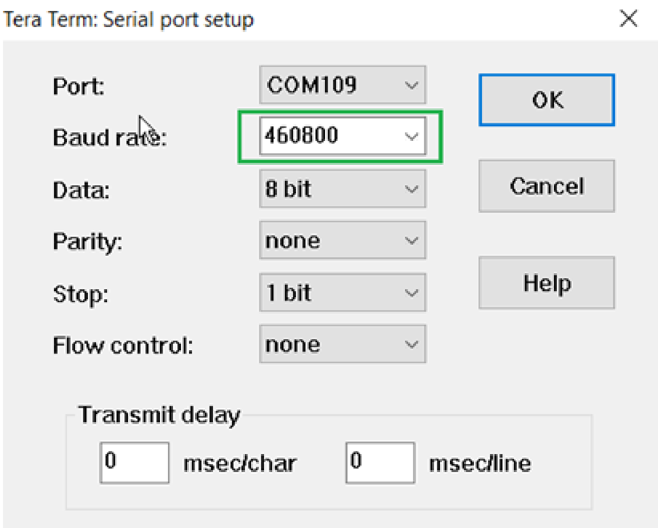
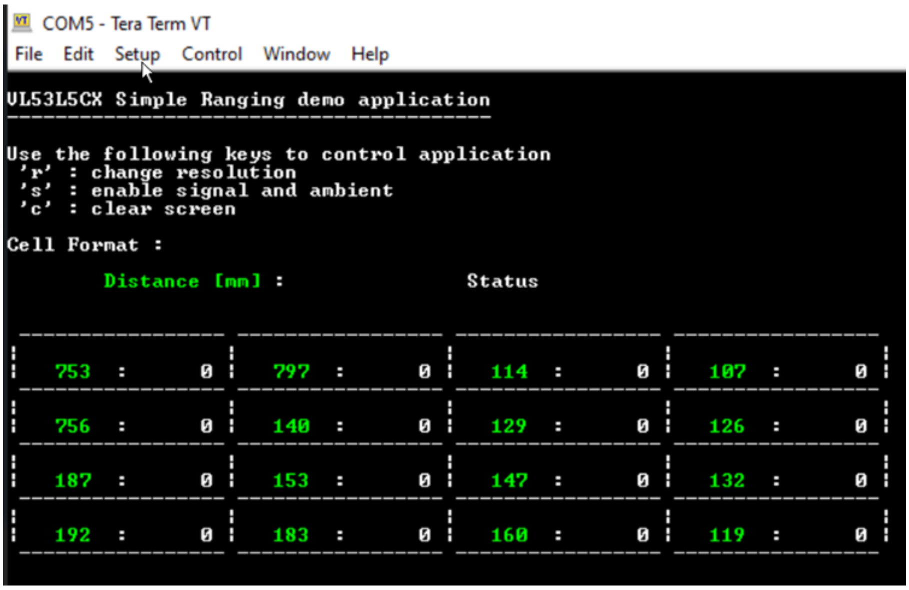


Figure 9. **Ranging results displayed on a terminal**



3.2 53L5A1_MultiSensorRanging

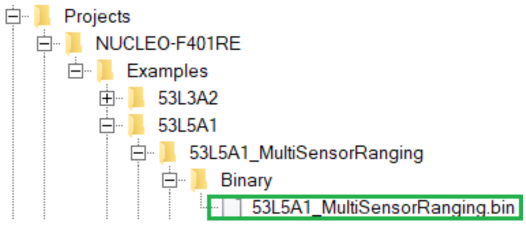
This sample application shows how to make three ToF sensors run simultaneously.

To test this application, two VL53L5CX-SATEL breakout boards, a X-NUCLEO-53L5A1, and a Nucleo F401RE are required. Ranging data are displayed on the serial terminal as shown below. This application is run by loading the prebuilt binary 53L5A1_MultiSensorRanging.bin.

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

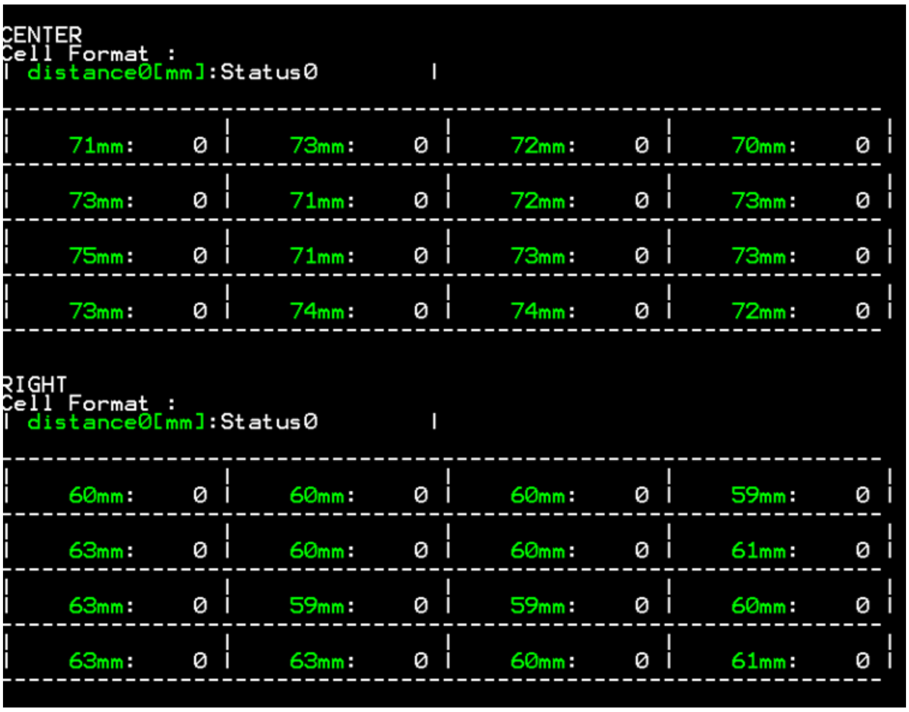
1. Flash the Nucleo F401RE board with the prebuilt binary.

Figure 10. Prebuilt binary



2. Open the Tera Term and configure it with the settings in Section 3 VL53L5CX sample application description.
3. Wave your hand in front of the sensor to display the ranging data on the serial terminal as shown below.

Figure 11. VL53L5CX ranging data



4 VL53L5CX configuration steps

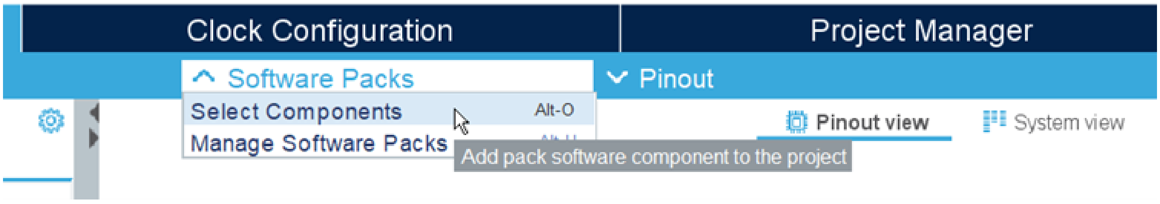
The X-NUCLEO-53L5A1 interfaces with the STM32 microcontroller via the I2C pin. Assuming the user wants to interface the X-NUCLEO-53L5A1 expansion board with a STM32 Nucleo 64-pin board (e.g. a Nucleo-F401RE), no particular hardware modification is necessary.

4.1 Use of expansion software without sample applications

This section outlines how to configure the STM32CubeMX with the X-NUCLEO-53L5A1 when the use of the sample applications is not required. With such a setup, only the driver layers are configured.

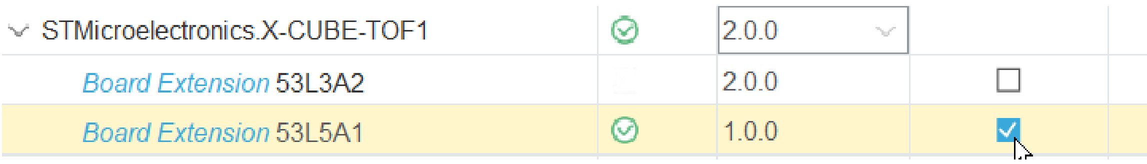
1. Add the X-CUBE-TOF1 SW pack to the project by clicking on the Software Packs button. Then, Select Components as shown below.

Figure 12. **Add X-CUBE-TOF1 SW pack to the project**



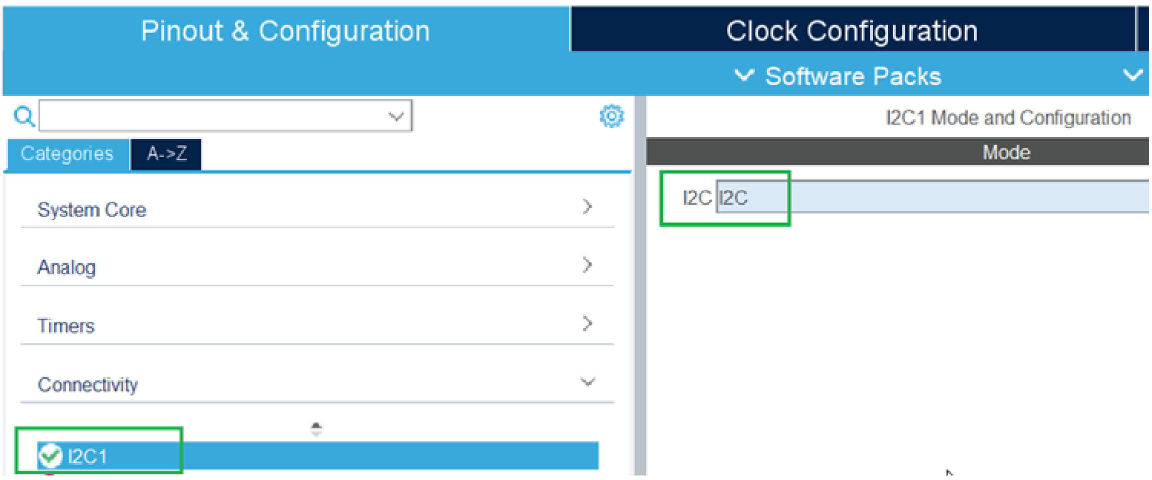
2. From the Software Packs Component Selector window, select the appropriate Board Extension class e.g. in this case 53L5A1.

Figure 13. **Select Board Extension class**



3. Enable the I2C1 as shown below.

Figure 14. **Enable I2C1**

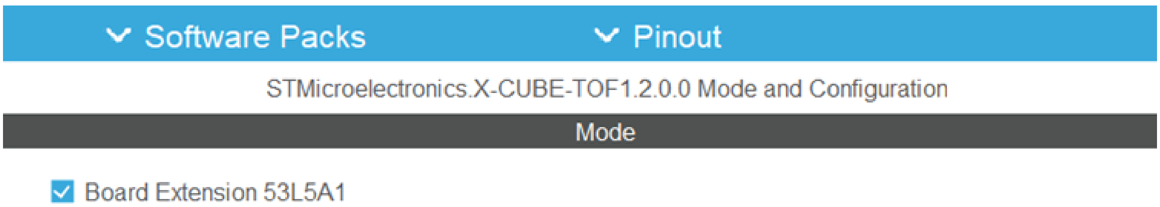


4. From the Software Packs dropdown menu, select STMicroelectronics.X-CUBE-TOF1.2.0.0.

Figure 15. Select STMicroelectronics.X-CUBE-TOF1

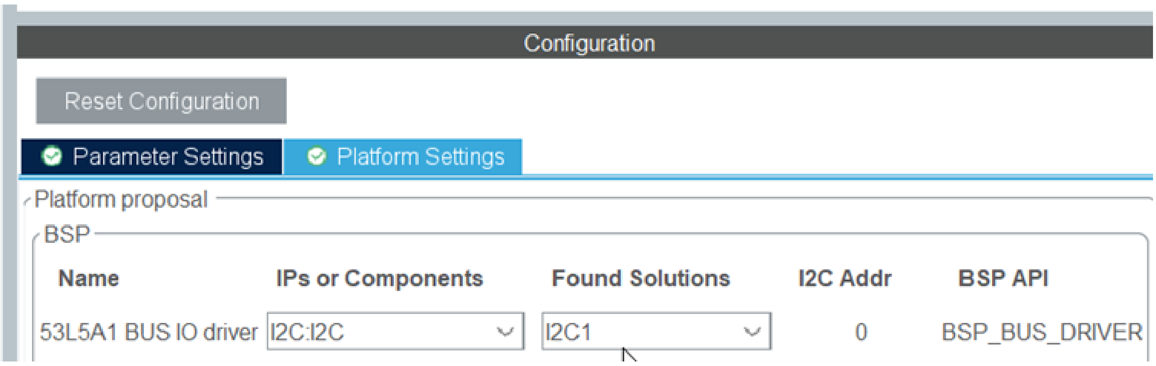
5. From the Mode view, select the Board Extension 53L5A1.

Figure 16. Select Board Extension 53L5A1



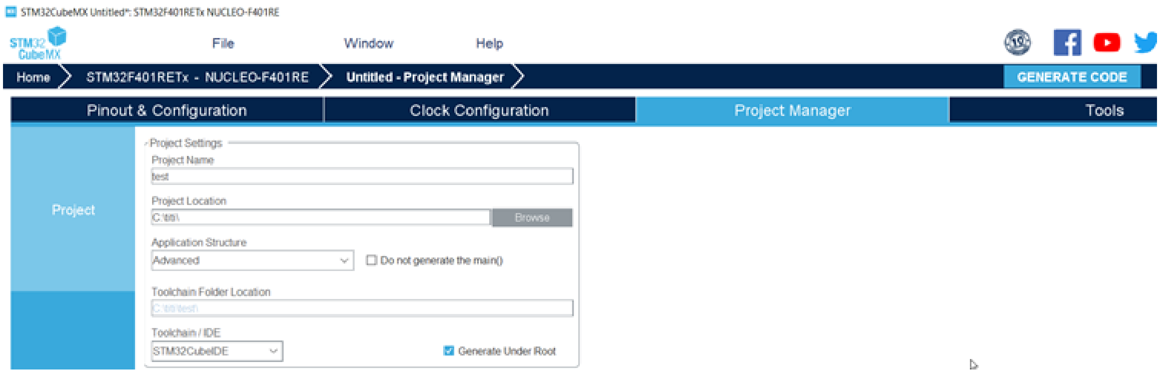
6. From the Configuration window, enable the I2C1.

Figure 17. Enable I2C1



7. Next, click on the Project Manager to name the project and select the appropriate Toolchain/IDE to generate the codes.

Figure 18. Name project and select appropriate Toolchain/IDE



8. Click on the GENERATE CODE button to generate the source code of the project using the X-CUBE-TOF1 software.

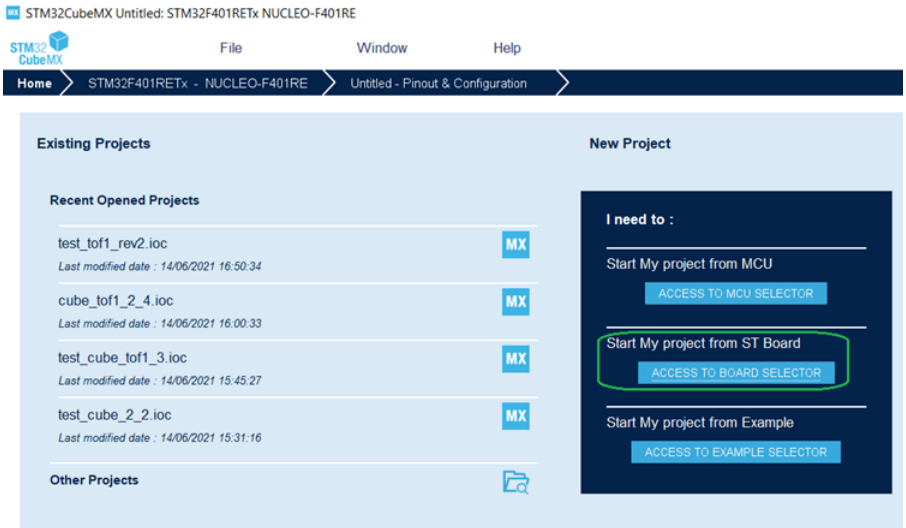
Figure 19. Generate source code of project using X-CUBE-TOF1 software

4.2 Use of expansion software with sample applications

4.2.1 How to generate 53L5A1_SimpleRanging example with CubeMX

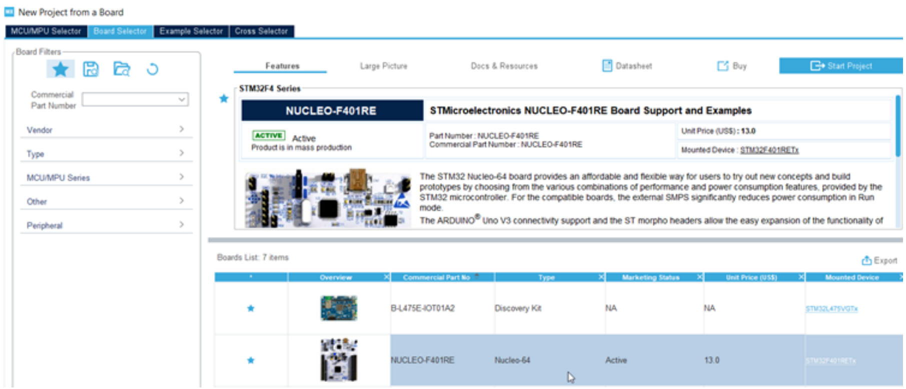
1. Open STM32CubeMX and select “Access to board selector".

Figure 20. Access to board selector



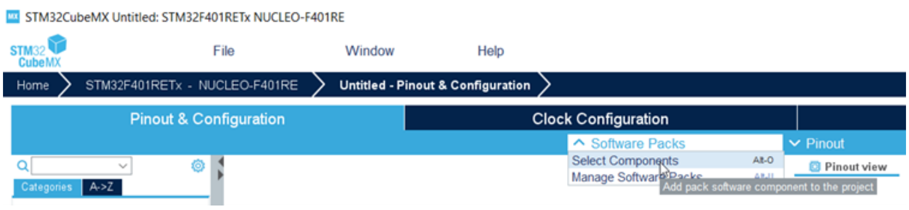
2. Select the F401RE board

Figure 21. F401RE board



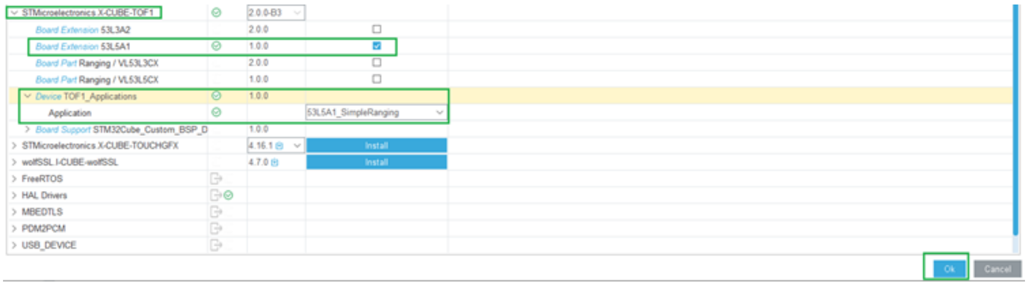
3. Right click “Select Components”

Figure 22. Components



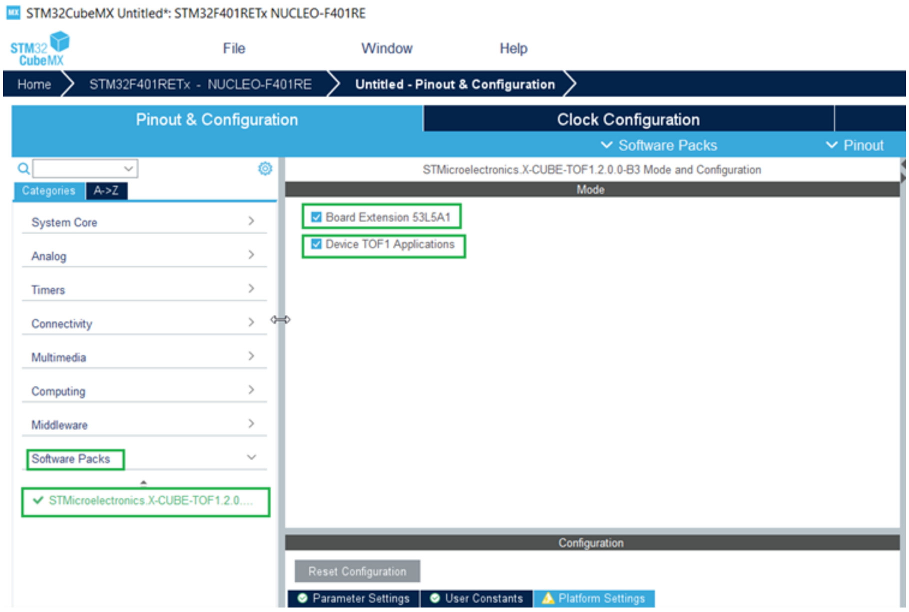
4. Select "X-CUBE-TOF1", then select "53L5A1 Board Extension", next select "53L5A1_SimpleRanging", and finally "OK".

Figure 23. **53L5A1_SimpleRanging**



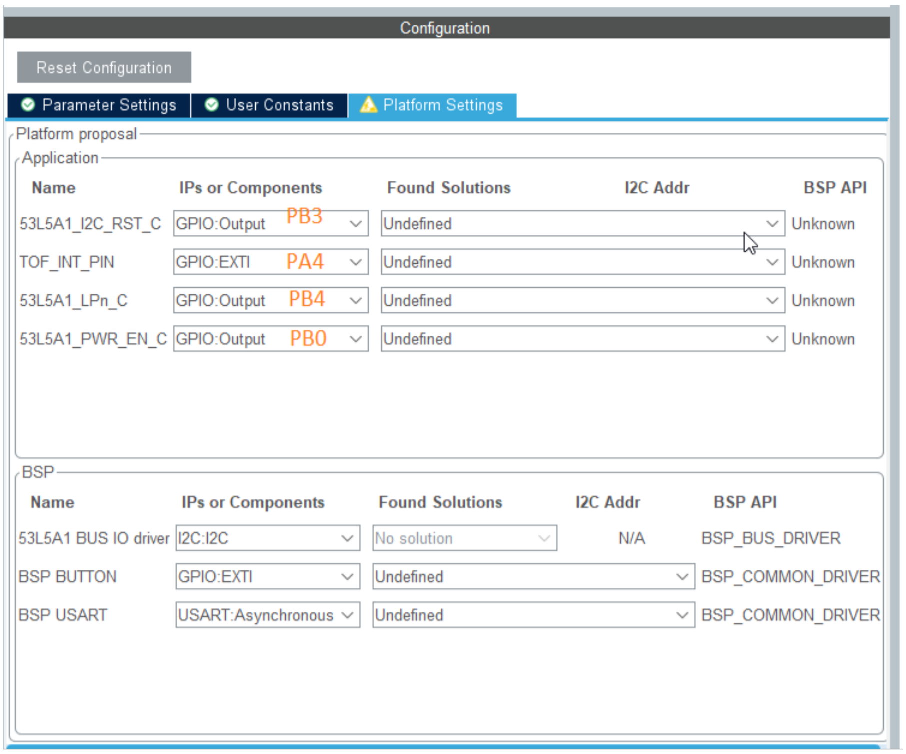
5. Select "Software Packs", then select "STMicroelectronics X-CUBE-TOF1", next select "Board Extension 53L5A1 box", and finally select the "Device TOF1 Applications".

Figure 24. **Device TOF1 application box selection**



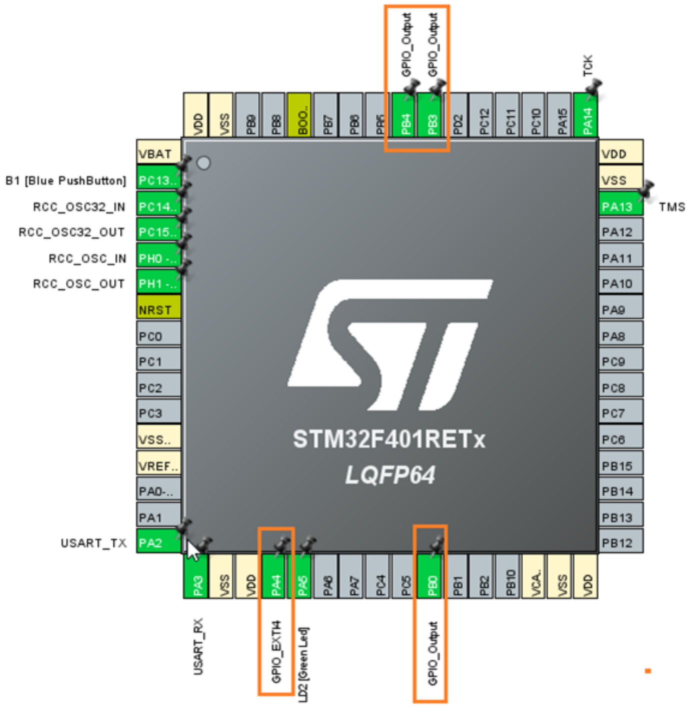
6. Configure the GPIOs for the application.

Figure 25. Configuration of GPIOs for application



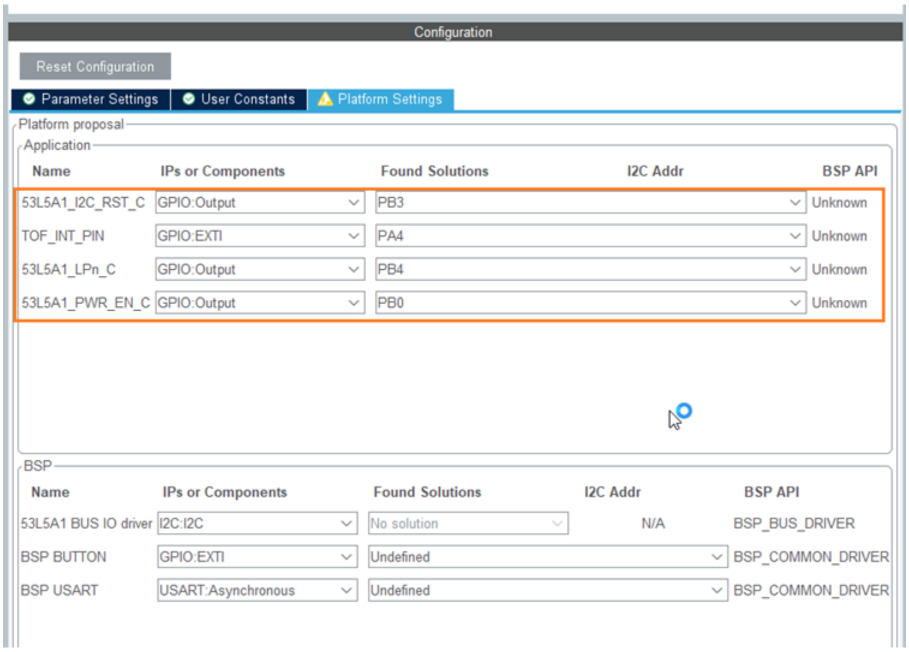
7. Select the GPIO pins as shown below.

Figure 26. GPIO pin selection



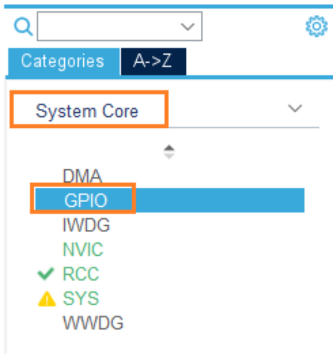
8. Link the GPIOs to the corresponding pin names as shown below.

Figure 27. GPIO and pin name correspondance



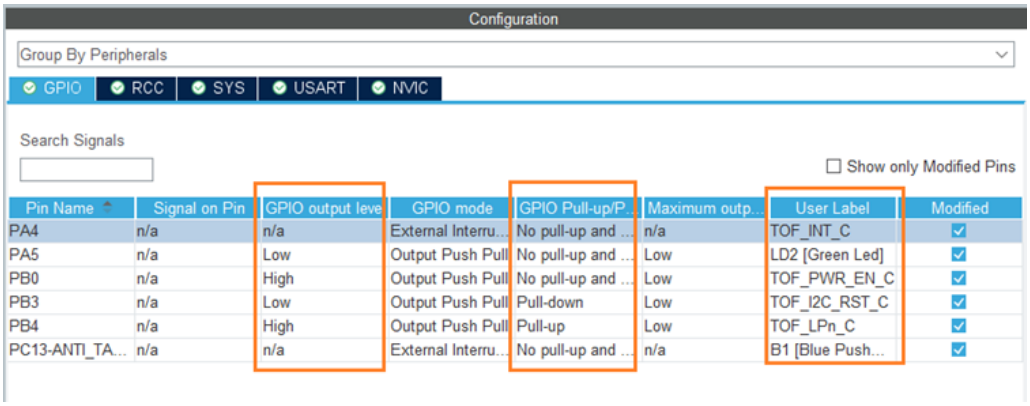
9. Select "GPIO" to open the GPIO configuration window as shown below.

Figure 28. GPIO configuration window



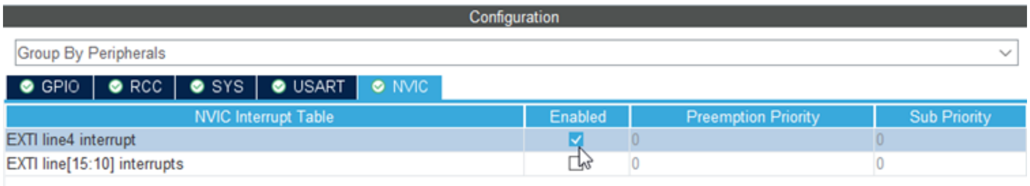
10. Name and configure the pins as shown below.

Figure 29. Pin name configuration



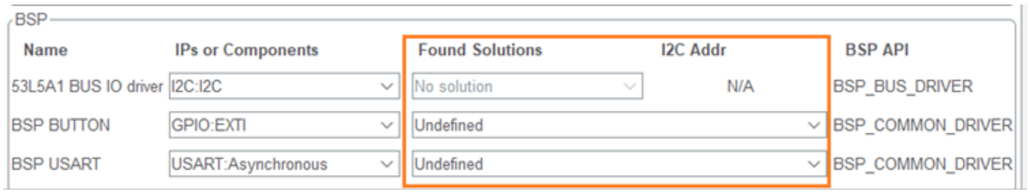
11. Activate the NVIC interrupt vector as shown below.

Figure 30. NVIC interrupt vector activation



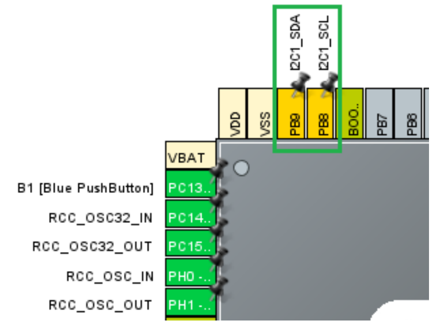
12. Configure the I2C and BSP as shown below.

Figure 31. I2C and BSP configuration



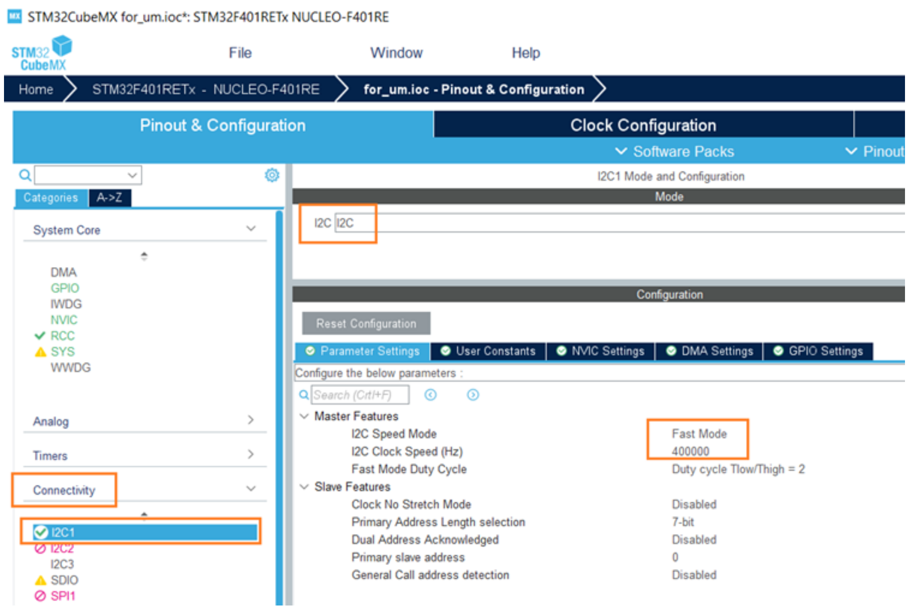
13. Select the pins PB9 and PB8 for SDA and SCL as shown below.

Figure 32. Pin selection for SDA and SCL



14. Select "Connectivity", then select "I2C1", next select "Enable I2C", and finally select "Fast Mode".

Figure 33. Fast mode selection



15. Select "Project Manager".

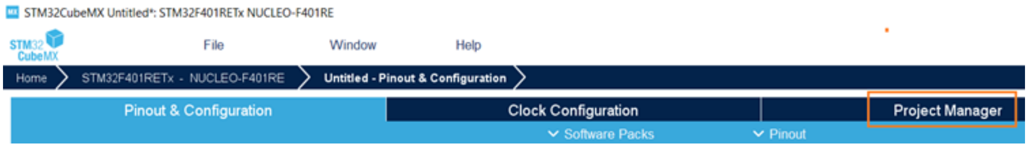


Figure 34. Project manager

16. Name the project, select "Toolchain", and then select “Generate Code” as shown below.

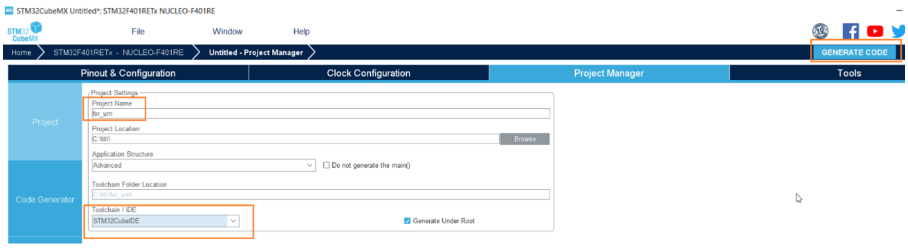


Figure 35. Code generation

17. Select “Open Project” on the pop-up window when code generation is complete, as shown below.

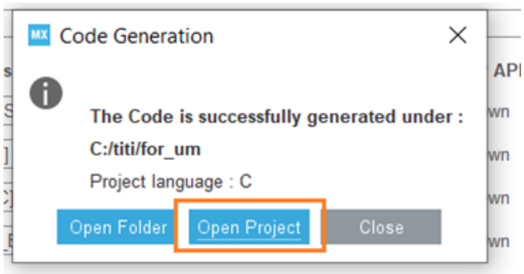
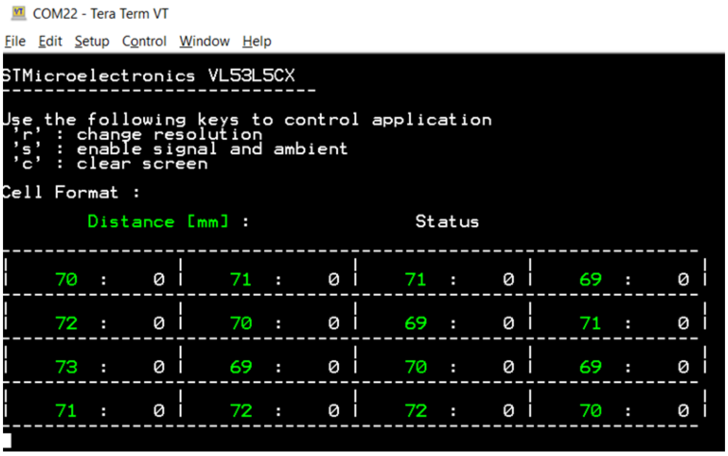


Figure 36. Open project

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

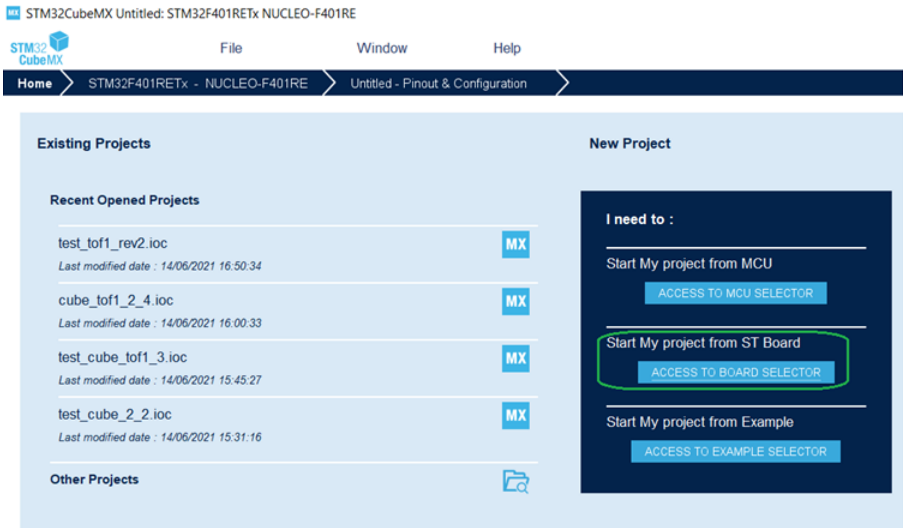
Figure 37. Results



4.2.2 How to generate 53L5A1_MultipleSensorRanging example with CubeMX

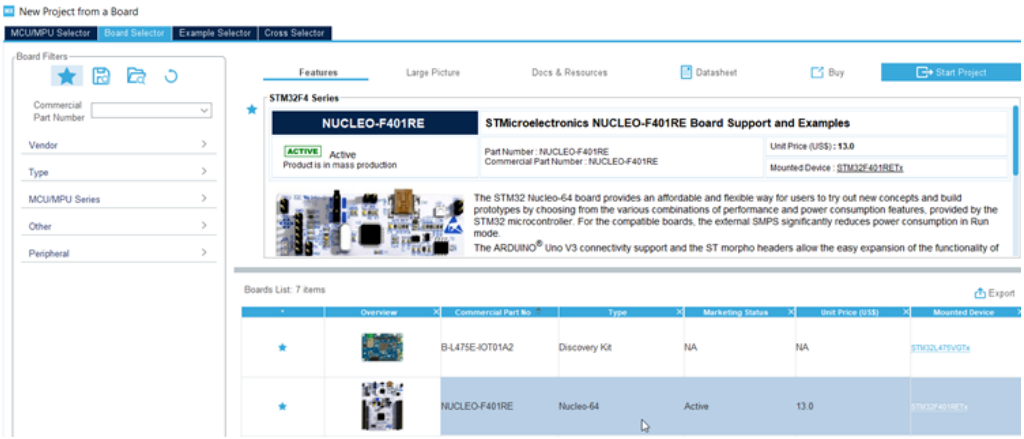
1. Open STM32CubeMX and select “Access to board selector".

Figure 38. Access to board selector



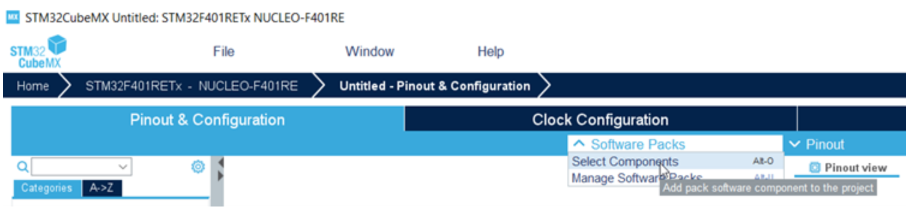
2. Select the F401RE board

Figure 39. F401RE board



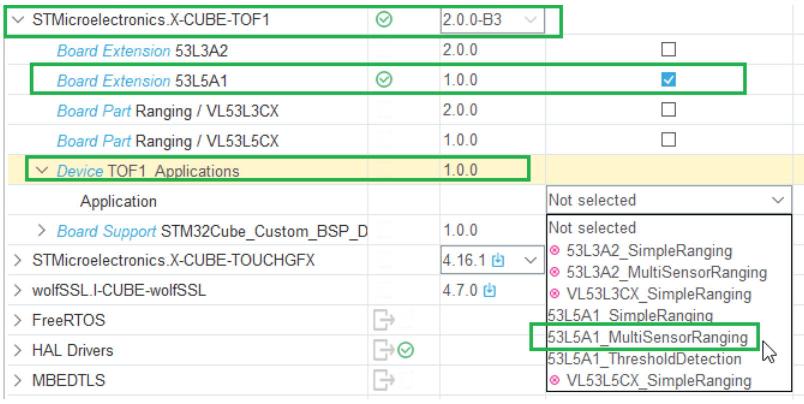
3. Right click “Select Components”

Figure 40. Components



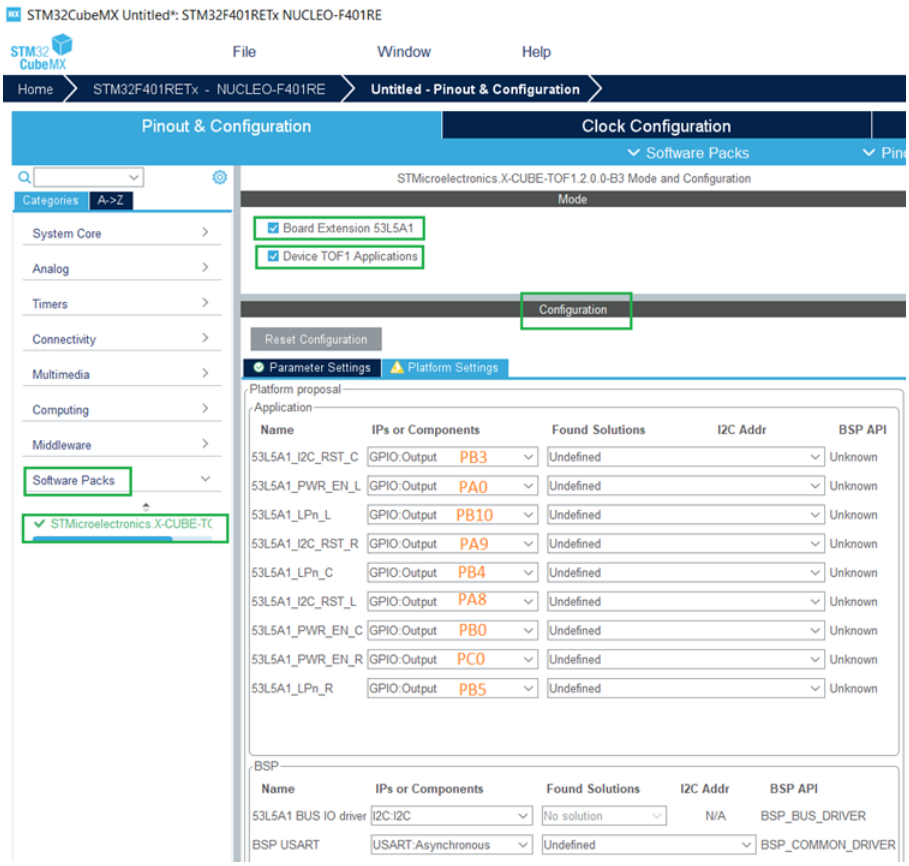
4. Select "X-CUBE-TOF1", then select "53L5A1 board", next select "53L5A1_MultiSensorRanging", and finally "OK" (bottom righthand corner).

Figure 41. 53L5A1_MultiSensorRanging



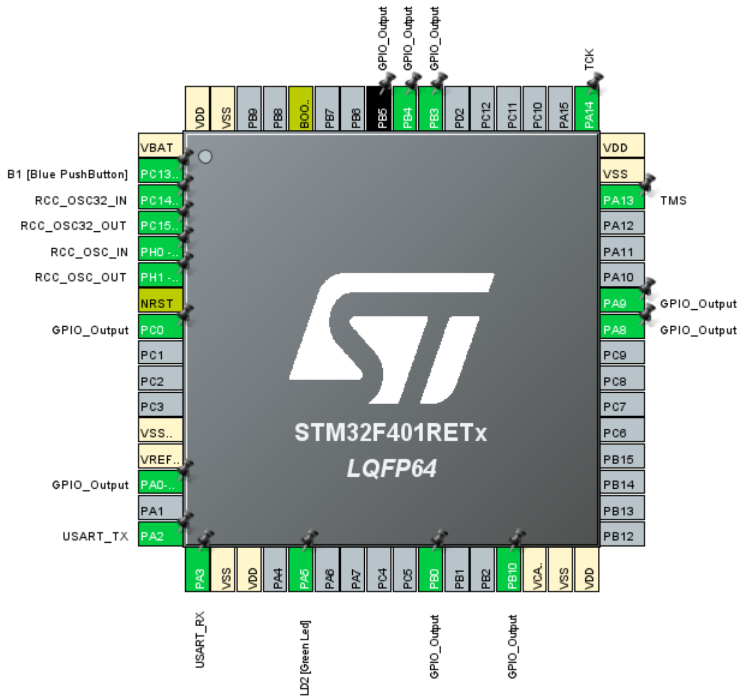
5. Select "Software Packs", then select "STMicroelectronics X-CUBE-TOF1", next select "Board Extension 53L5A1 box", and finally select the "Device TOF1 Applications box".

Figure 42. Device TOF1 application box selection



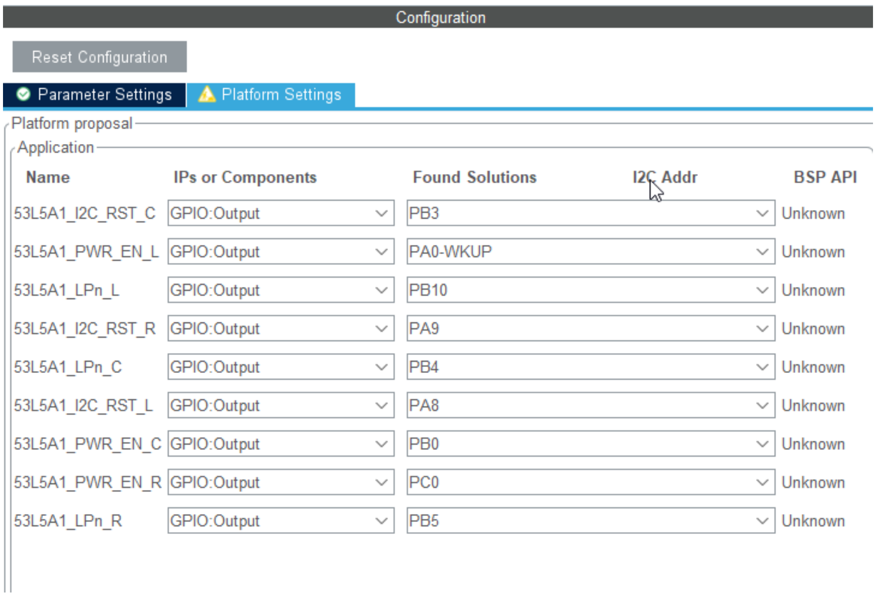
6. Select the GPIOs pins as shown below.

Figure 43. GPIO pin selection



7. Link the GPIOs to the corresponding pin names as shown below.

Figure 44. GPIO and pin name correspondance



Configuration

Reset Configuration

Parameter Settings | Platform Settings

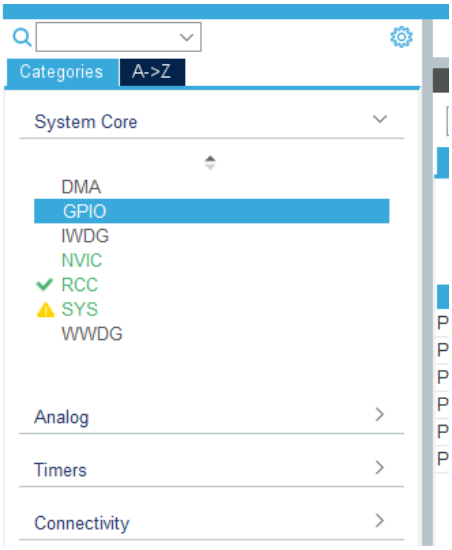
Platform proposal

Application

Name	IPs or Components	Found Solutions	I2C Addr	BSP API
53L5A1_I2C_RST_C	GPIO:Output	PB3		Unknown
53L5A1_PWR_EN_L	GPIO:Output	PA0-WKUP		Unknown
53L5A1_LPn_L	GPIO:Output	PB10		Unknown
53L5A1_I2C_RST_R	GPIO:Output	PA9		Unknown
53L5A1_LPn_C	GPIO:Output	PB4		Unknown
53L5A1_I2C_RST_L	GPIO:Output	PA8		Unknown
53L5A1_PWR_EN_C	GPIO:Output	PB0		Unknown
53L5A1_PWR_EN_R	GPIO:Output	PC0		Unknown
53L5A1_LPn_R	GPIO:Output	PB5		Unknown

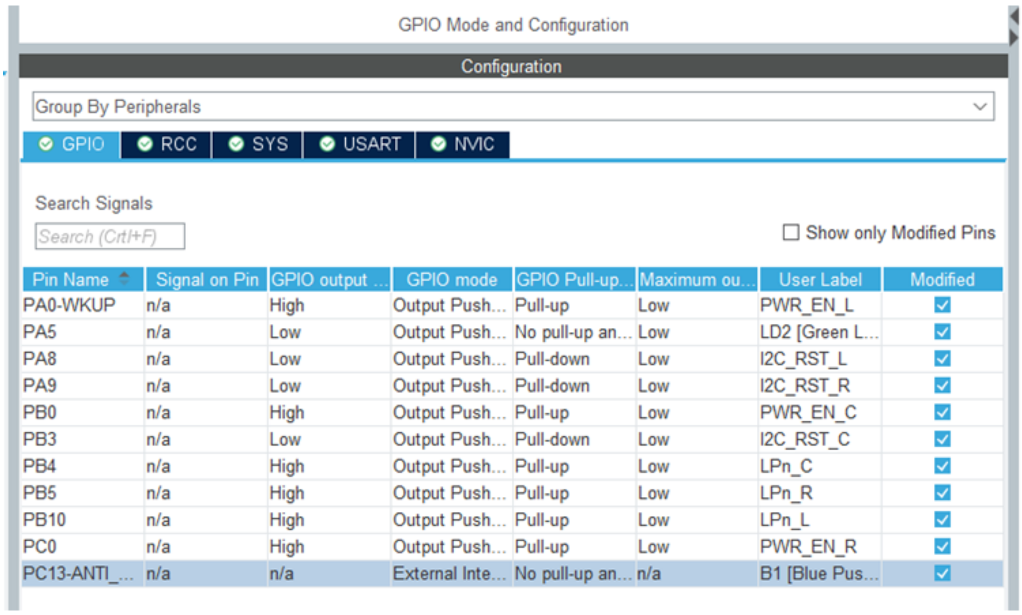
8. Select "GPIO" to open the GPIO configuration window as shown below.

Figure 45. GPIO configuration window



9. Name and configure the pins as shown below.

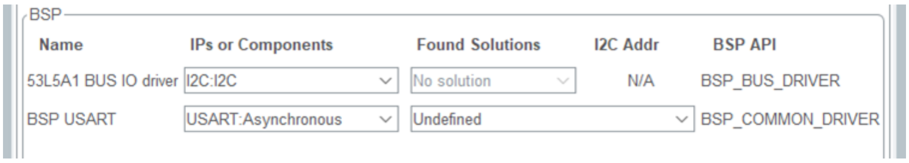
Figure 46. Pin name configuration



Pin Name	Signal on Pin	GPIO output ...	GPIO mode	GPIO Pull-up...	Maximum ou...	User Label	Modified
PA0-WKUP	n/a	High	Output Push...	Pull-up	Low	PWR_EN_L	✓
PA5	n/a	Low	Output Push...	No pull-up an...	Low	LD2 [Green L...	✓
PA8	n/a	Low	Output Push...	Pull-down	Low	I2C_RST_L	✓
PA9	n/a	Low	Output Push...	Pull-down	Low	I2C_RST_R	✓
PB0	n/a	High	Output Push...	Pull-up	Low	PWR_EN_C	✓
PB3	n/a	Low	Output Push...	Pull-down	Low	I2C_RST_C	✓
PB4	n/a	High	Output Push...	Pull-up	Low	LPn_C	✓
PB5	n/a	High	Output Push...	Pull-up	Low	LPn_R	✓
PB10	n/a	High	Output Push...	Pull-up	Low	LPn_L	✓
PC0	n/a	High	Output Push...	Pull-up	Low	PWR_EN_R	✓
PC13-ANTI_...	n/a	n/a	External Inte...	No pull-up an...	n/a	B1 [Blue Pus...	✓

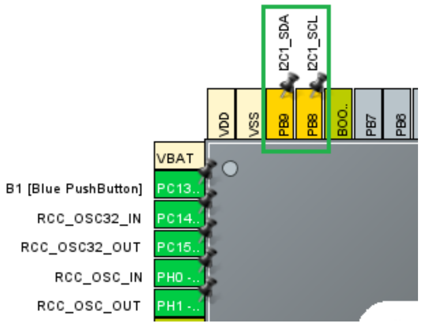
10. Configure the I2C and UART as shown below.

Figure 47. I2C and UART configuration



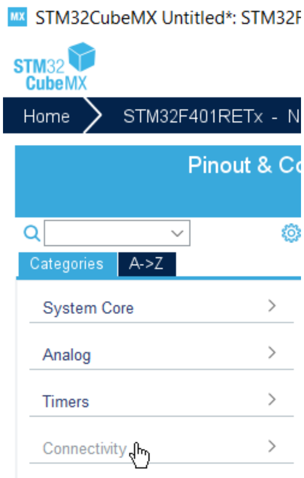
11. Select the pins PB9 and PB8 for SDA and SCL as shown below.

Figure 48. Pin selection for SDA and SCL



12. Select "Connectivity"

Figure 49. Connectivity selection



13. Select "I2C1", then enable the I2C, and select "Fast Mode" as shown below.

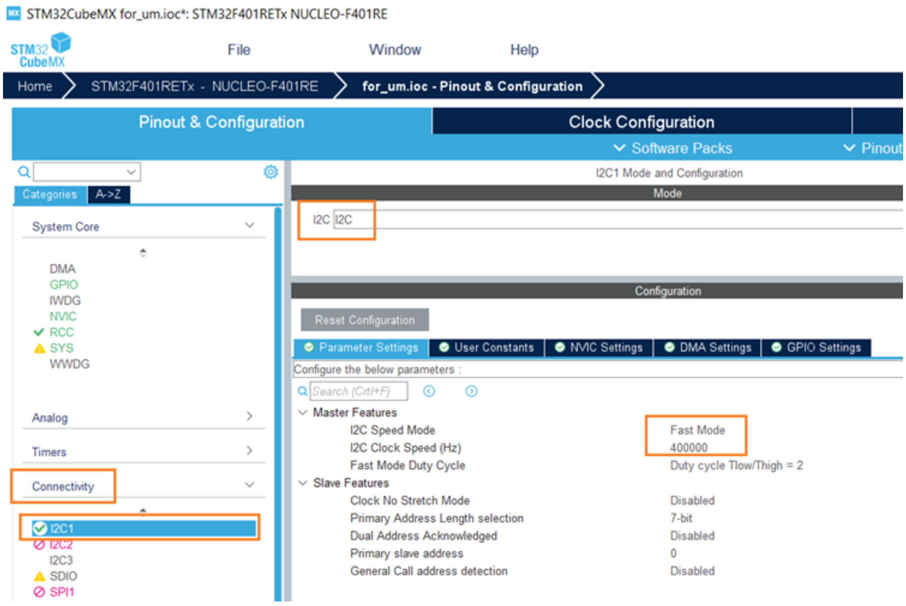


Figure 50. **Fast mode selection**

14. Configure the I2C and BSP as shown below.

Figure 51. **I2C and BSP configuration**

15. Select "Project Manager".

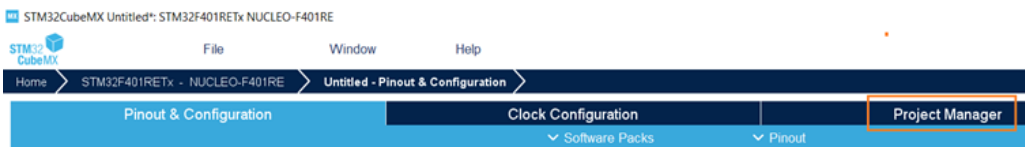


Figure 52. **Project manager**

16. Name the project, select "Toolchain", and then select “Generate Code” as shown below.

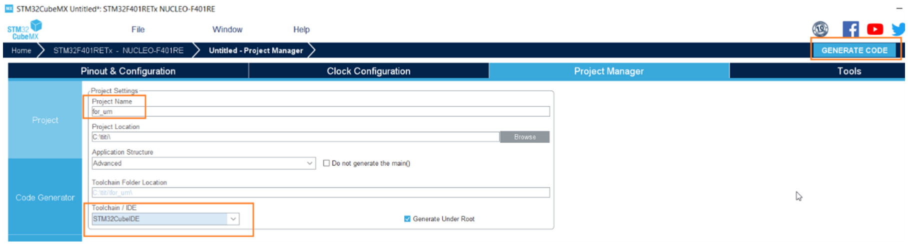


Figure 53. Code generation

17. Select “Open Project” on the pop-up window when code generation is complete, as shown below.

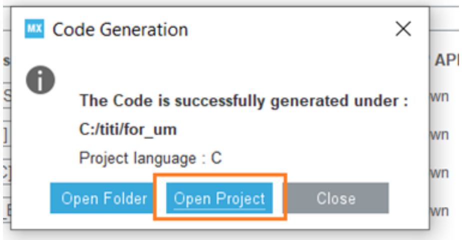


Figure 54. Open project

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

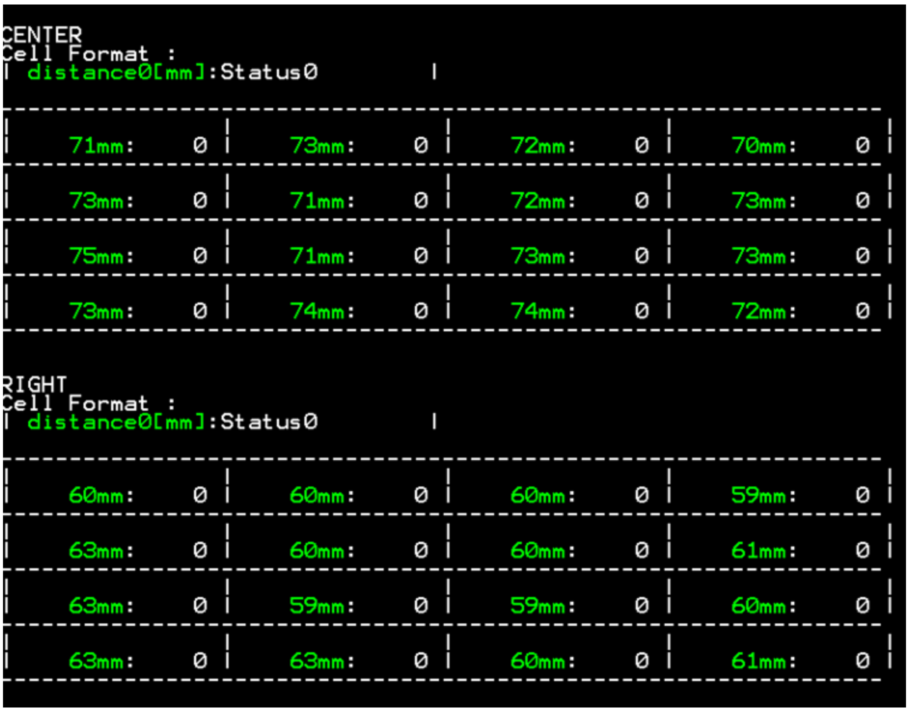
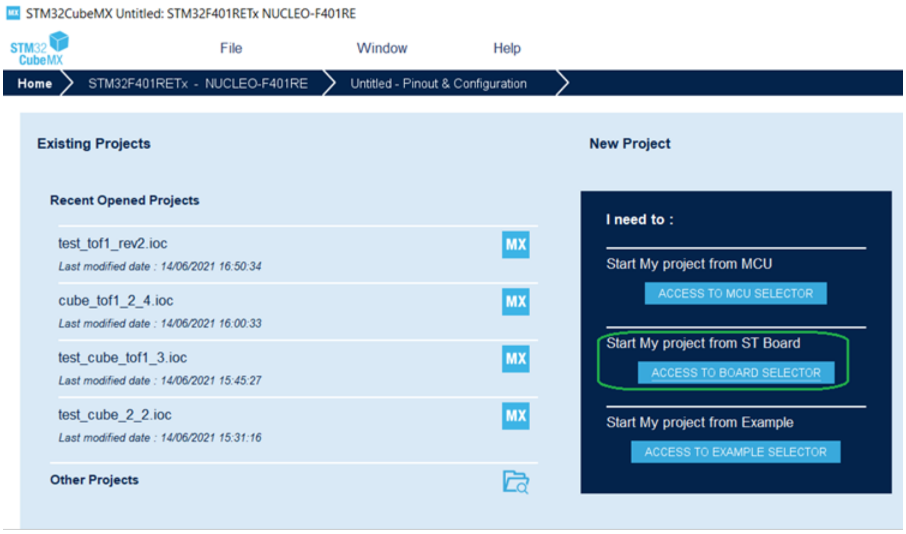


Figure 55. Results

4.2.3 How to generate the 53L5A1_ThresholdDetection example with CubeMX

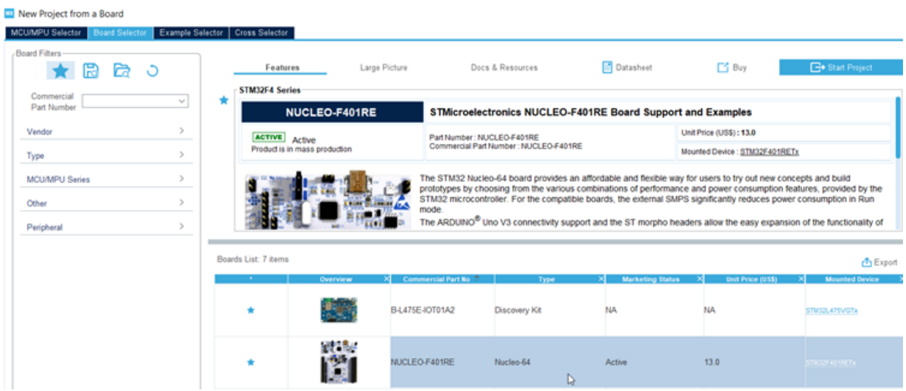
1. Open STM32CubeMX and select “Access to board selector".

Figure 56. Access to board selector



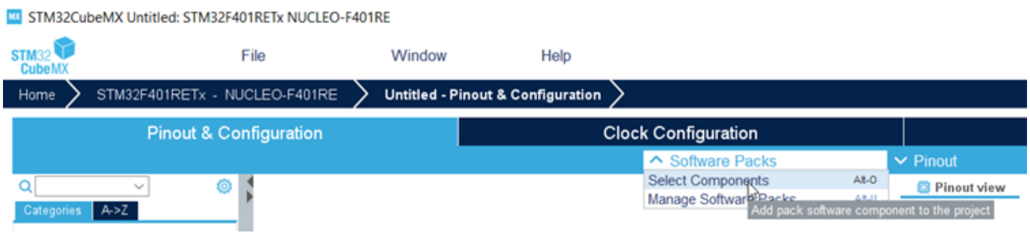
2. Select the F401RE board

Figure 57. F401RE board



3. Right click “Select Components”

Figure 58. Components



4. Select the board and application as shown below, then select "OK" (bottom righthand corner).

Figure 59. Board selection

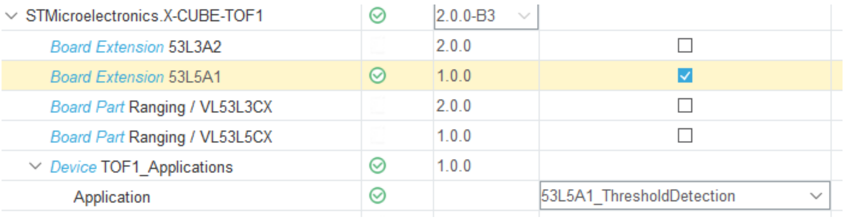
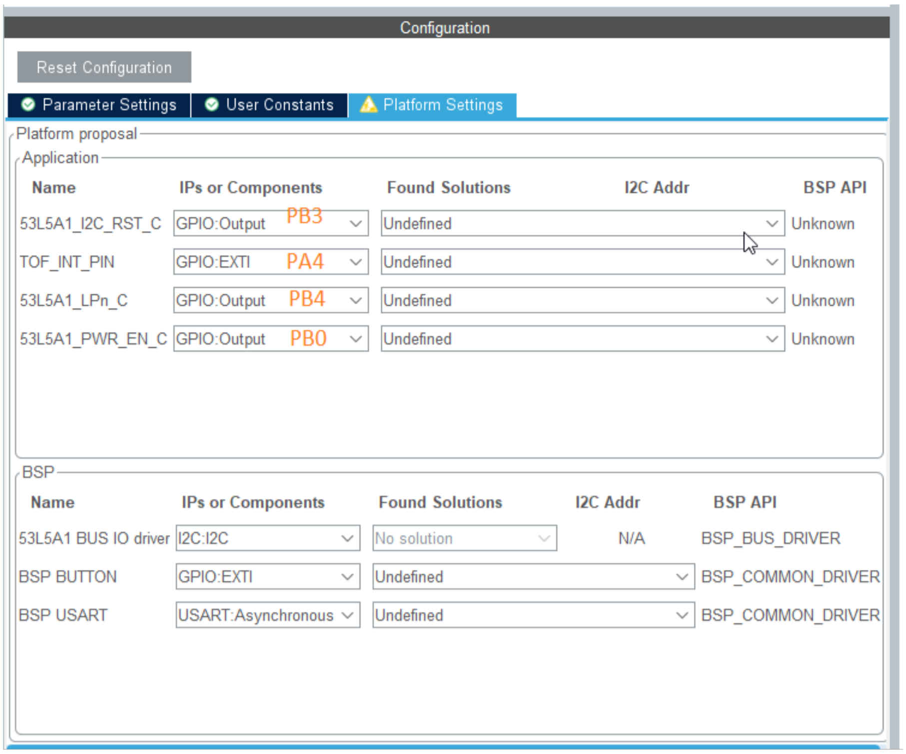
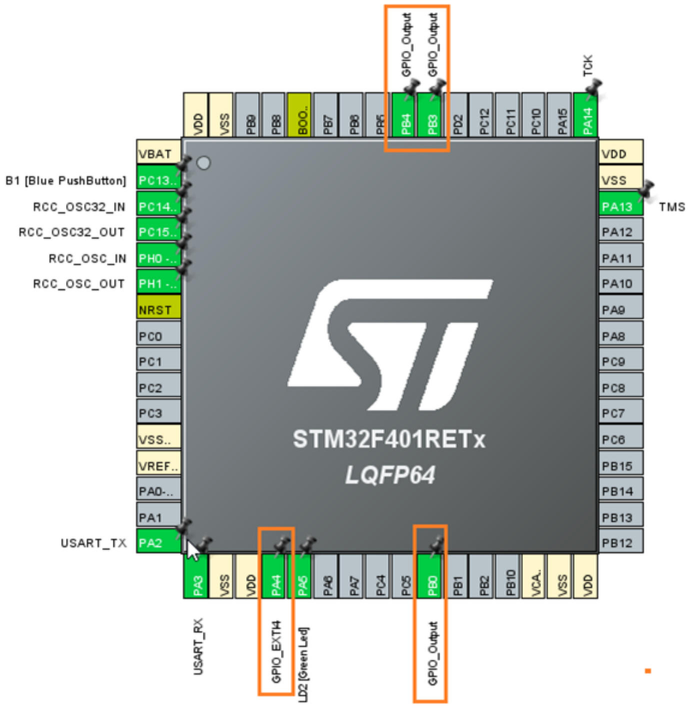


Figure 60. Application selection



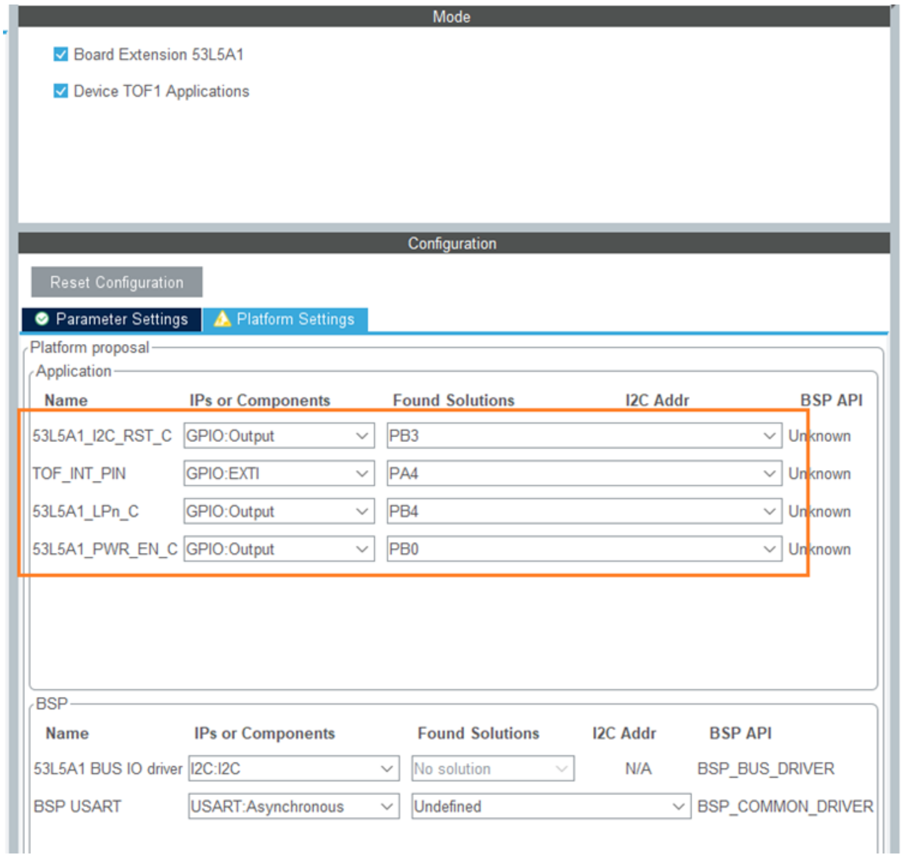
5. Select the GPIOs pins by referring to the X-NUCLEO-53L5A1 schematic as shown below.

Figure 61. GPIO pin selection



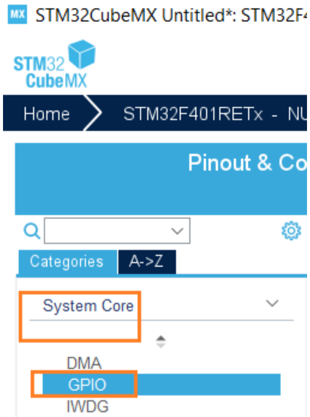
6. Link the GPIOs to the corresponding pin names as shown below.

Figure 62. **GPIO and pin name correspondance**



7. Select "System Core" then "GPIO" to open the GPIO configuration window as shown below.

Figure 63. **GPIO configuration window**



8. Name and configure the pins as shown below.

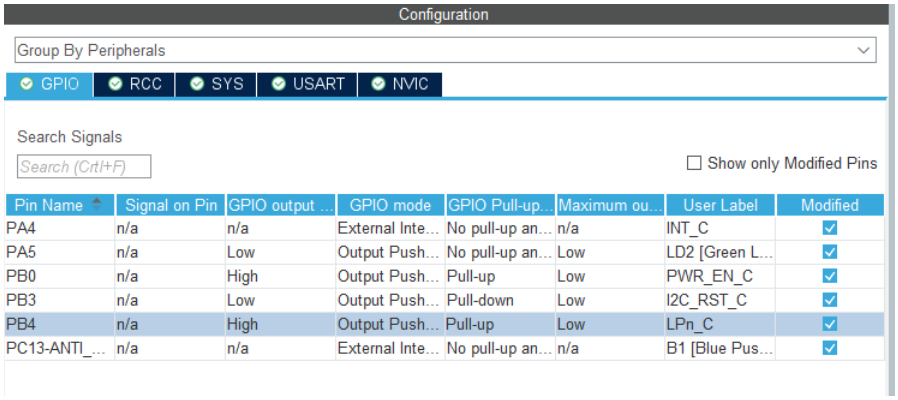


Figure 64. Pin name configuration

9. Activate the NVIC interrupt vector as shown below.

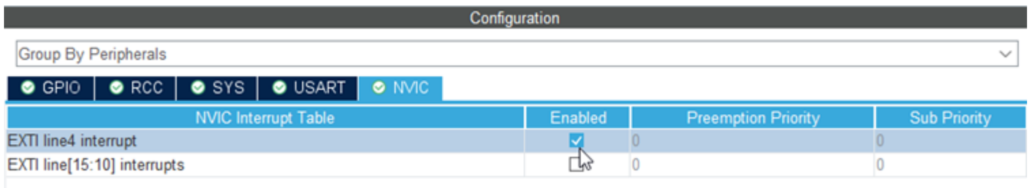


Figure 65. NVIC interrupt vector activation

10. Configure the I2C and BSP as shown below.

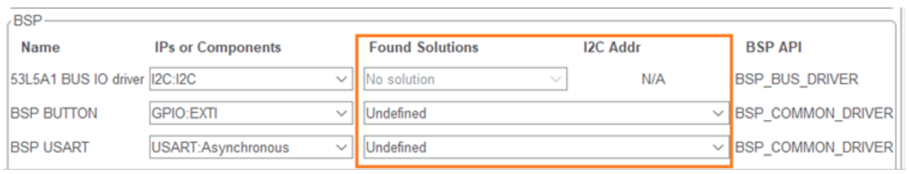
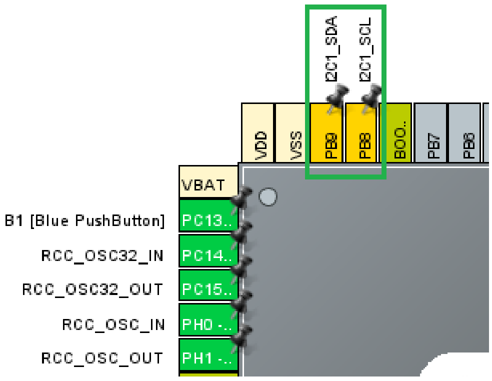


Figure 66. I2C and BSP configuration

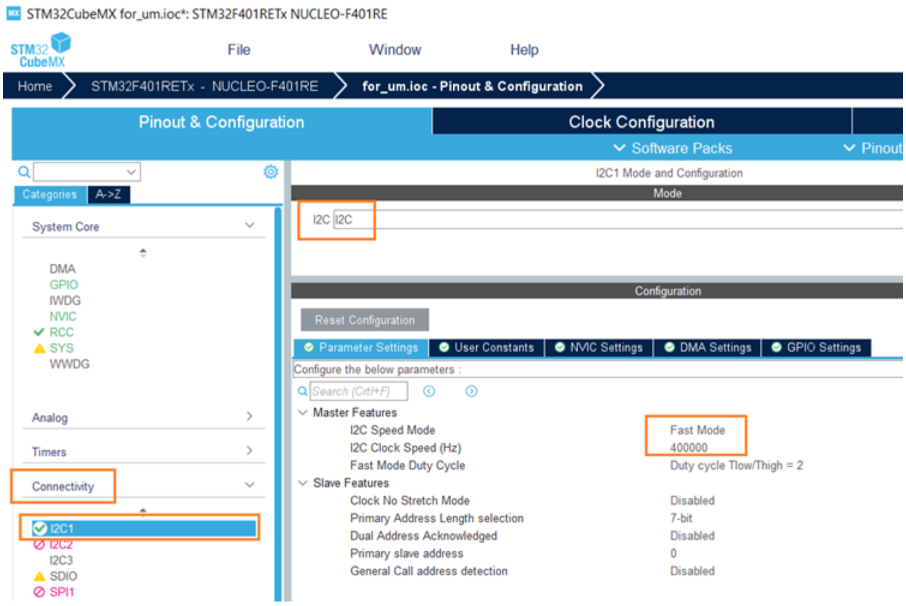
11. Select the pins PB9 and PB8 for SDA and SCL as shown below.

Figure 67. Pin selection for SDA and SCL



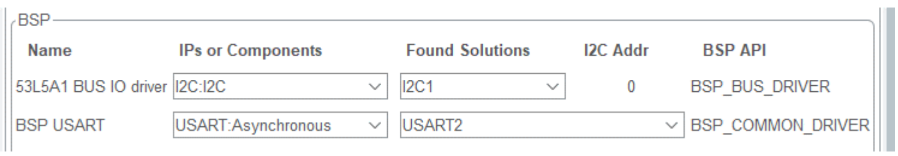
12. Select "Connectivity" and "I2C1", then enable the I2C and select "Fast Mode" as shown below

Figure 68. Fast mode selection



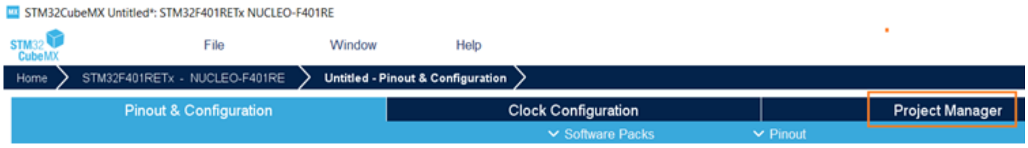
Configure the I2C and BSP as shown below.

Figure 69. I2C and BSP configuration



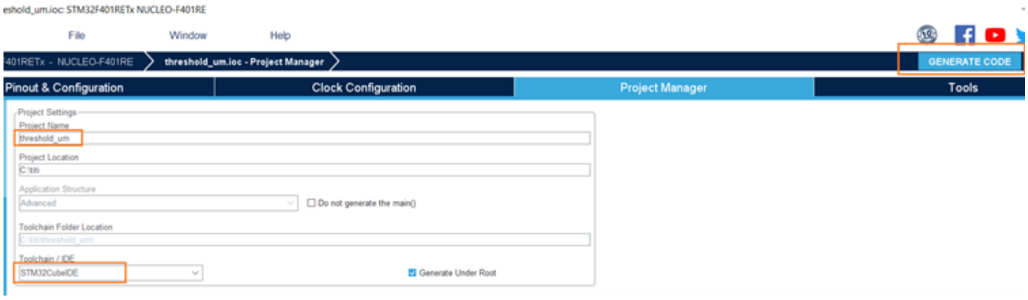
13. Select "Project Manager".

Figure 70. Project manager



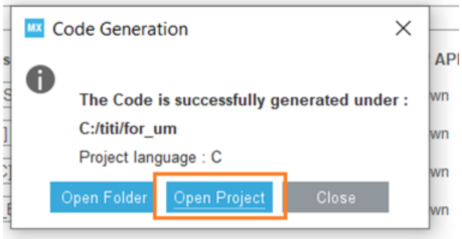
14. Name the project, select "Toolchain", and then select “Generate Code” as shown below.

Figure 71. Code generation



15. Select “Open Project” on the pop-up window when code generation is complete, as shown below.

Figure 72. Open project



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

Figure 73. Results

```
STMicroelectronics VL53L5CX
---------------------------

53L5A1 Simple Ranging Example
-----------------------------

Cell Format :
| distance0[mm]:Status0        |

----------------------------------------------------------------
|   576mm:     1 |   514mm:     0 |     2mm:     1 |   448mm:     0 |
----------------------------------------------------------------
|   552mm:     0 |   512mm:     0 |   506mm:     1 |   473mm:     0 |
----------------------------------------------------------------
|   330mm:     0 |   514mm:     1 |   502mm:     0 |   453mm:     0 |
----------------------------------------------------------------
|   292mm:     0 |   404mm:     1 |   484mm:     0 |   456mm:     1 |
----------------------------------------------------------------
```

17. An interrupt occurs if the target distance (d) = [200 mm et 600 mm] as shown below.

Figure 74. Interrupt

app_x-cube-tof1.c

```
ITConfig.Criteria = RS_IT_IN_WINDOW;
ITConfig.LowThreshold = 200; /* mm */
ITConfig.HighThreshold = 600; /* mm */
```

4.2.4 How to generate VL53L5CX_SimpleRanging example with CubeMX

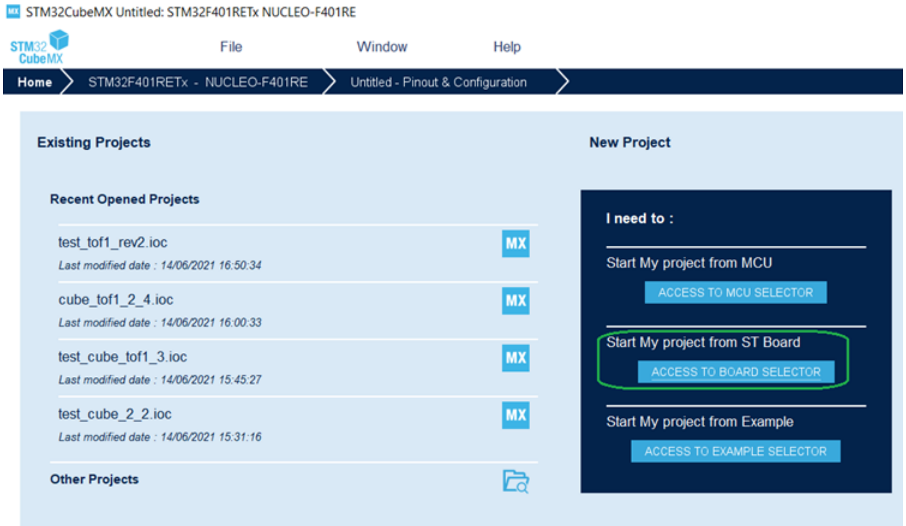
Required material for this example is as follows.

- Nucleo F401RE
- VL53L5CX-SATEL
- Dupont wires

The breakout board is connected directly onto the Nucleo F401RE without the X-NUCLEO-53L5A1 expansion board .

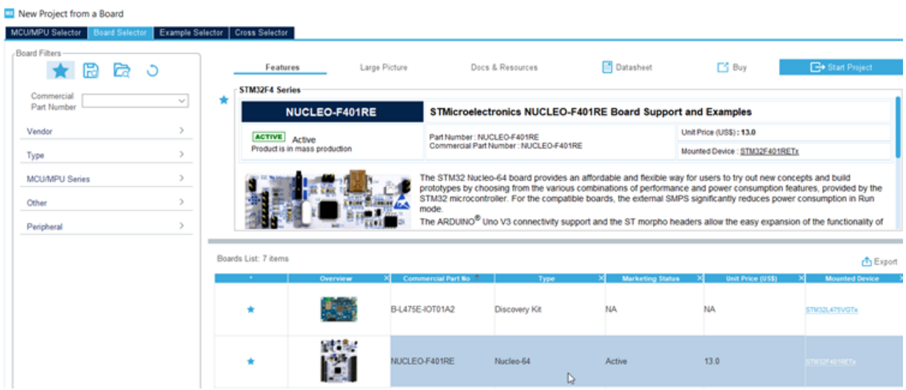
1. Open STM32CubeMX and select “Access to board selector".

Figure 75. Access to board selector



2. Select the F401RE board

Figure 76. F401RE board



3. Right click “Select Components”

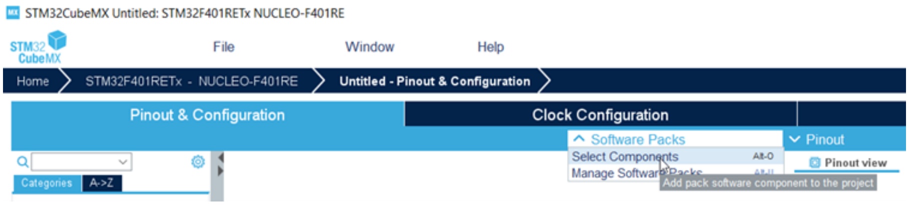


Figure 77. Components

4. Select the board and application as shown below, then select "OK" (bottom righthand corner).

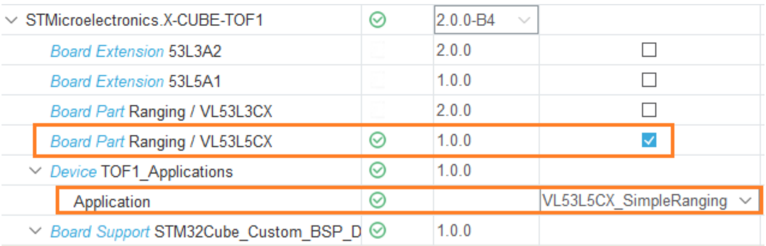


Figure 78. Board selection

5. Select "Software Packs", then select "STMicroelectronics X-CUBE-TOF1", next select "Board Part Ranging box", select the "Device TOF1 Applications box", and finally select "Board Support STM32Cube Custom BSP Drivers".

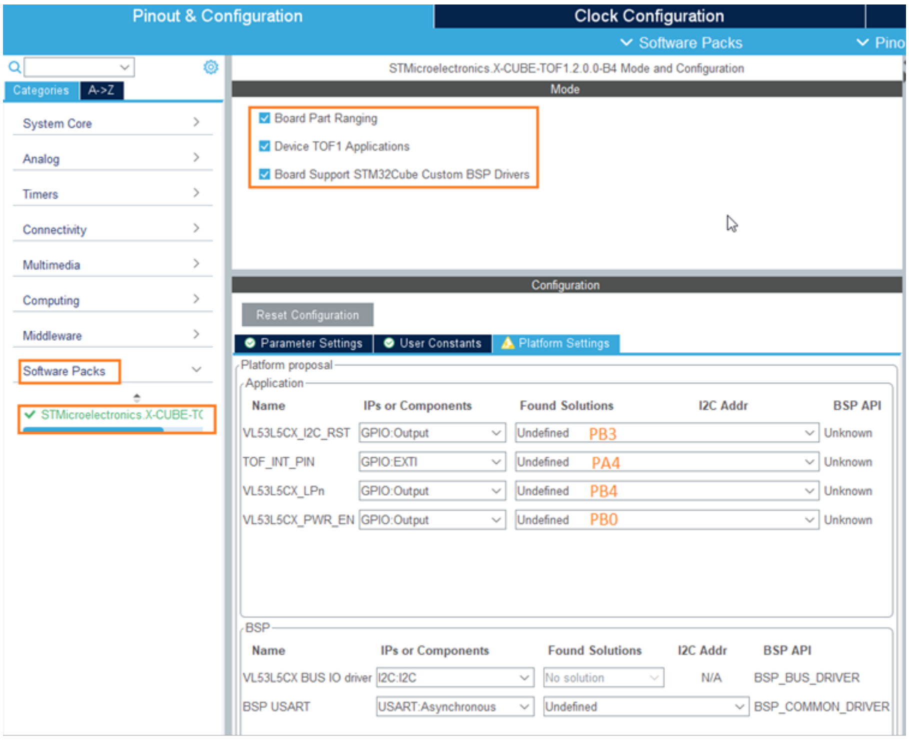


Figure 79. Board support STM32Cube custom BSP drivers

6. Select the GPIOs pins by referring to the X-NUCLEO-53L5A1 schematic as shown below. Use the same GPIO pin on the expansion board as the central sensor then, the user can choose the other GPIO pins e.g. PB3, PA4, PB4, PB0.

Figure 80. GPIO pin selection (1)

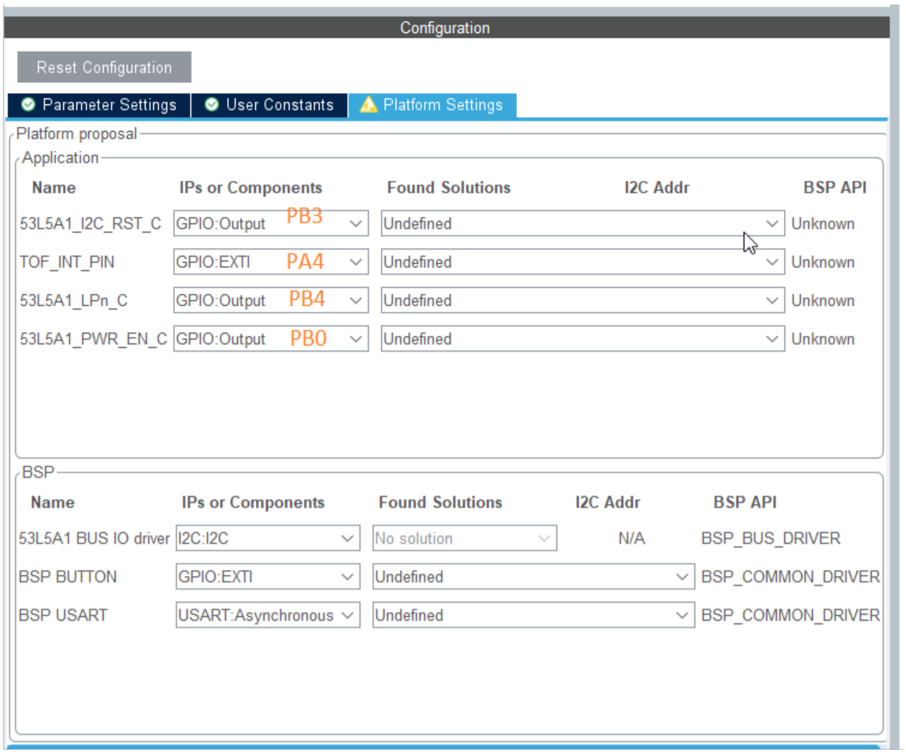
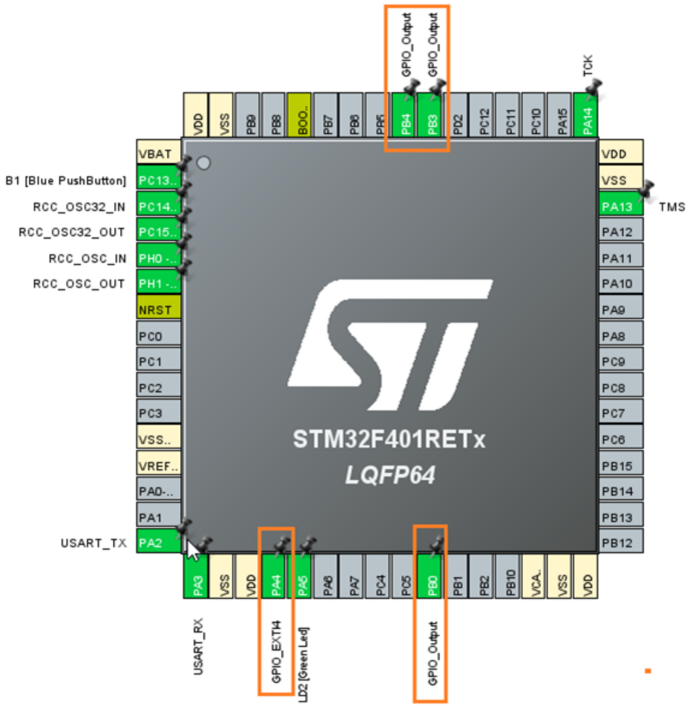
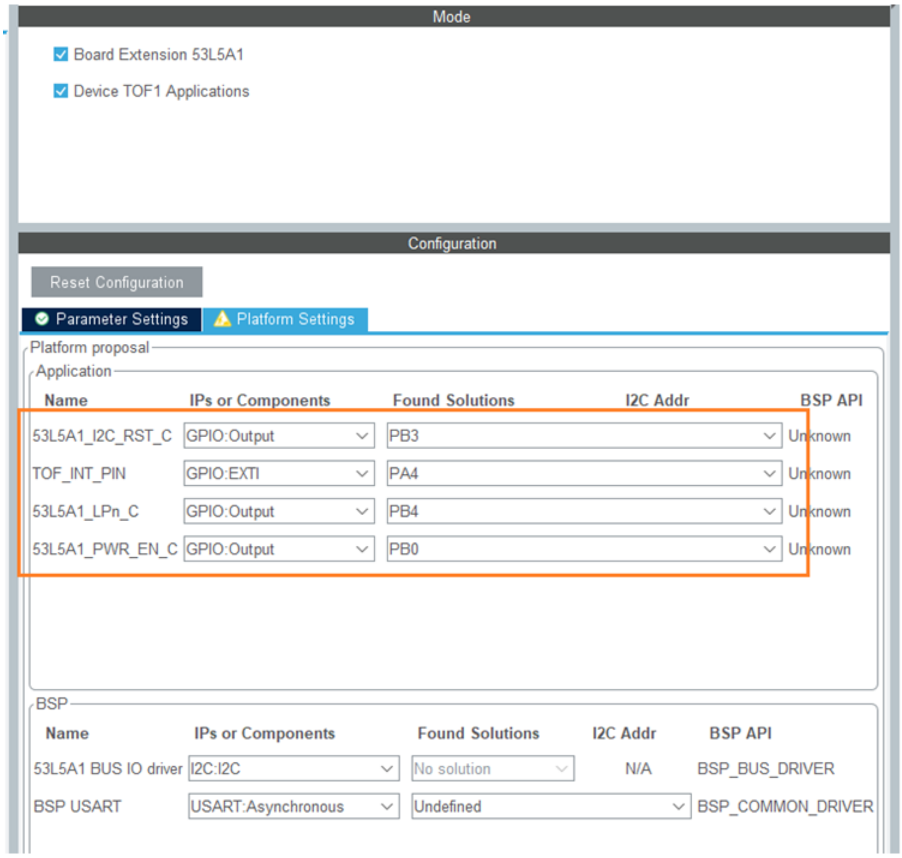


Figure 81. GPIO pin selection (2)



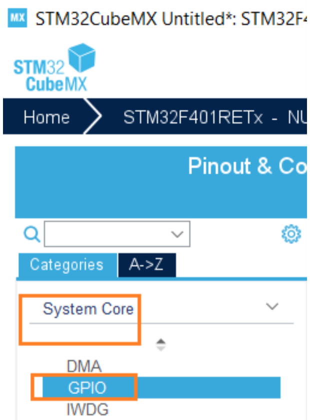
7. Link the GPIOs to the corresponding pin names as shown below.

Figure 82. GPIO and pin name correspondance



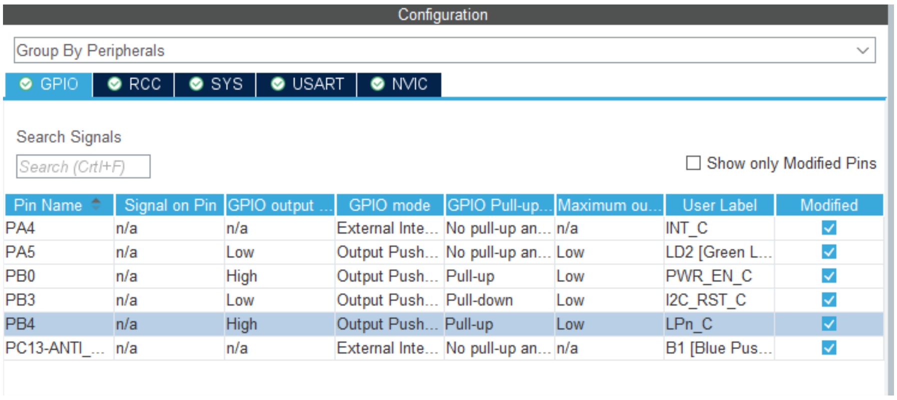
8. Select "System Core" then "GPIO" to open the GPIO configuration window as shown below.

Figure 83. GPIO configuration window



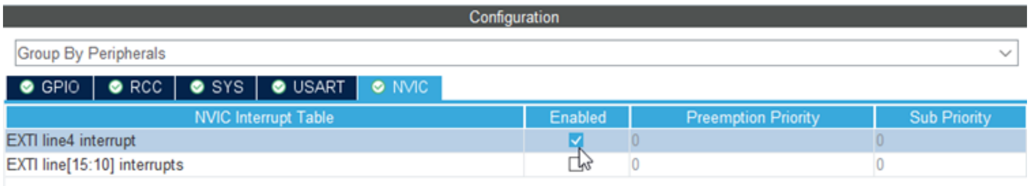
9. Name and configure the pins as shown below.

Figure 84. Pin name configuration



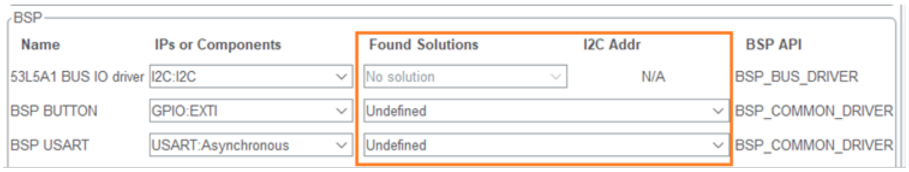
10. Activate the NVIC interrupt vector as shown below.

Figure 85. NVIC interrupt vector activation



11. Configure the I2C and BSP as shown below.

Figure 86. I2C and BSP configuration



12. Select the pins PB9 and PB8 for SDA and SCL as shown below.

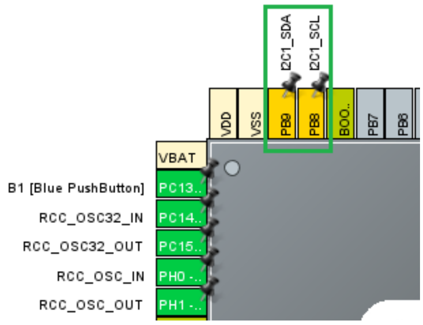


Figure 87. Pin selection for SDA and SCL

13. Select "Connectivity" and "I2C1", then enable the I2C and select "Fast Mode" as shown below

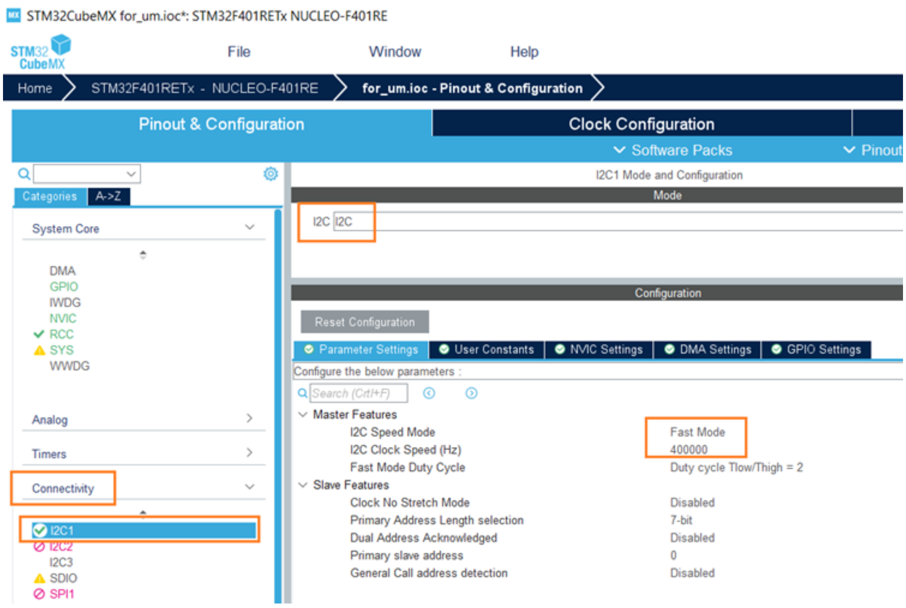
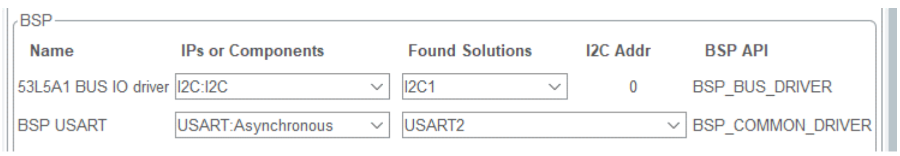


Figure 88. Fast mode selection

Configure the I2C and BSP as shown below.



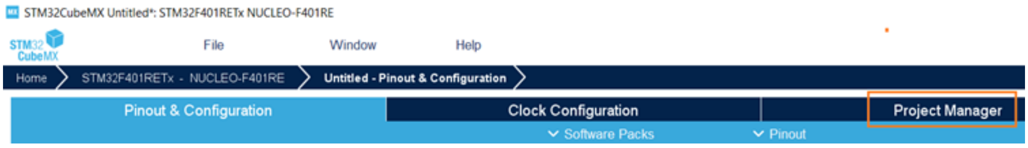
BSP

Name	IPs or Components	Found Solutions	I2C Addr	BSP API
53L5A1 BUS IO driver	I2C:I2C	I2C1	0	BSP_BUS_DRIVER
BSP USART	USART:Asynchronous	USART2		BSP_COMMON_DRIVER

Figure 89. I2C and BSP configuration

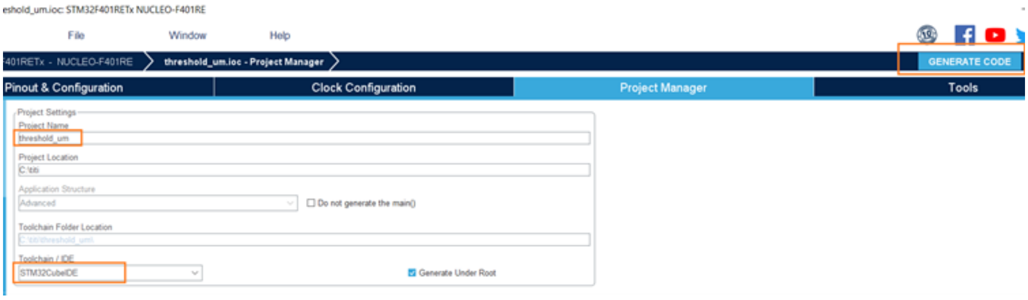
14. Select "Project Manager".

Figure 90. Project manager



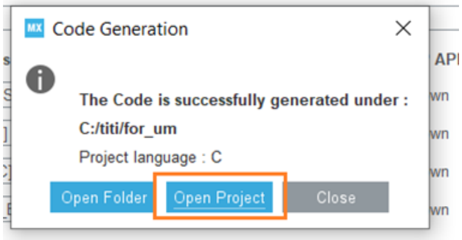
15. Name the project, select "Toolchain", and then select “Generate Code” as shown below.

Figure 91. Code generation



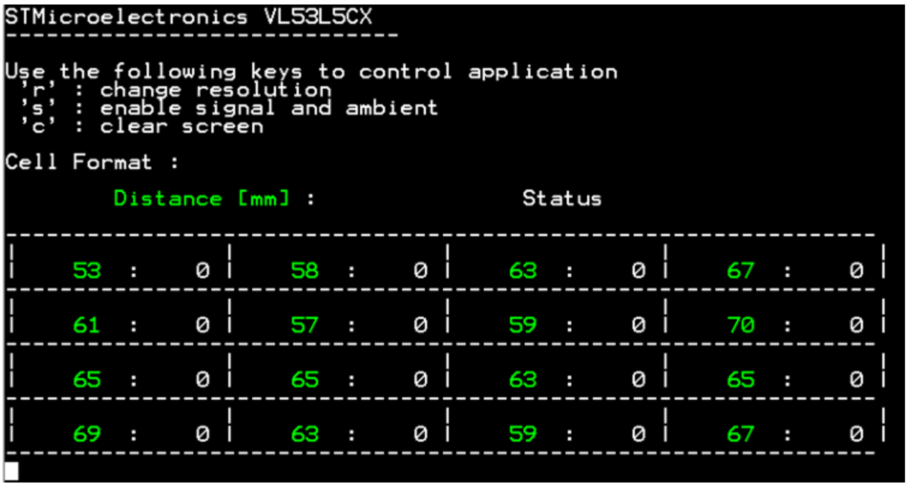
16. Select “Open Project” on the pop-up window when code generation is complete, as shown below.

Figure 92. Open project



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

Figure 93. Results



5 System setup guide

5.1 Hardware description

5.1.1 STM32 Nucleo

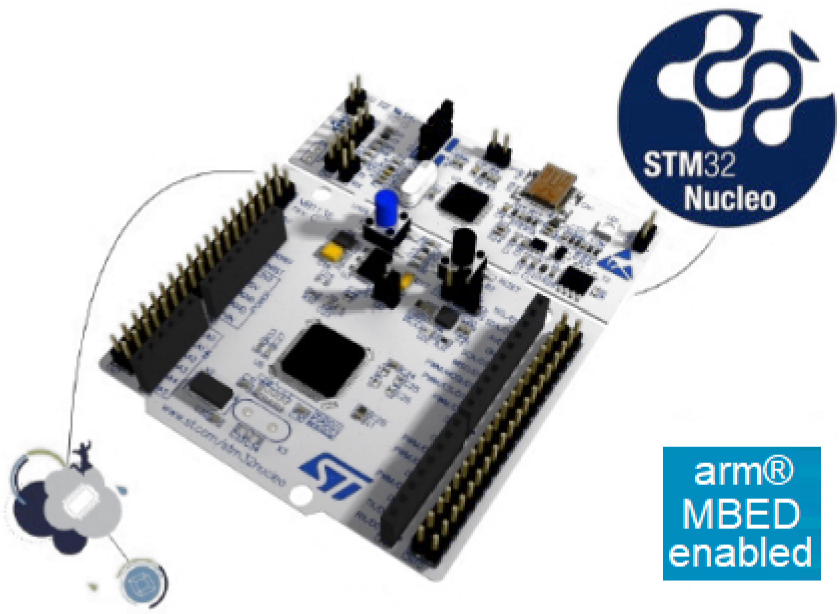
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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 ARM®).

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

Figure 94. STM Nucleo board



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

5.1.2 X-NUCLEO-53L5A1 expansion board

The X-NUCLEO-53L5A1 is an expansion board for any NUCLEO development board. It provides a complete evaluation kit allowing anyone to learn, evaluate, and develop their applications using the VL53L5CX, ranging sensor with multitarget detection..

The X-NUCLEO-53L5A1 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.

The X-NUCLEO-53L5A1 expansion board is compatible with the STM32 nucleo board family, and with the Arduino UNO R3 connector layout.

Several ST expansion boards can be superposed through the Arduino connectors which allows, for example, the development of VL53L5CX applications with Bluetooth or Wi-Fi interfaces.

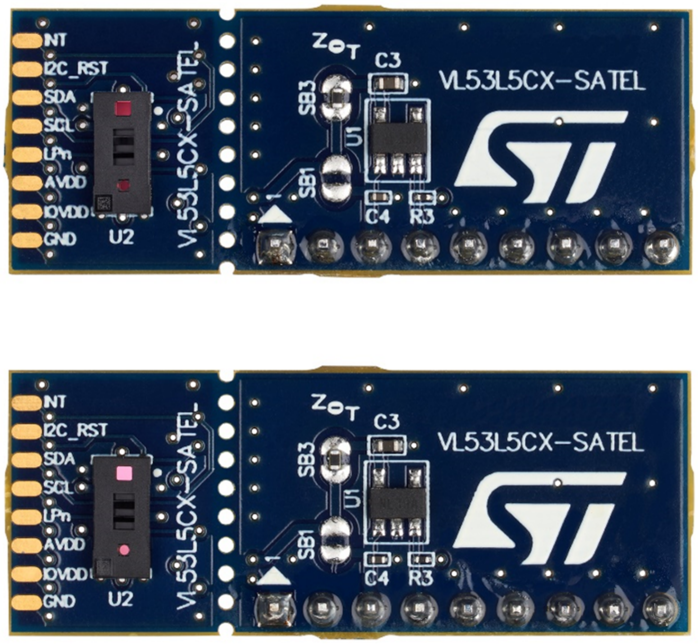
Figure 95. X-NUCLEO-53L5A1 expansion board

5.1.3 VL53L5CX-SATEL breakout boards

The VL53L5CX-SATEL breakout boards can be used for easy integration into customer devices. Thanks to the voltage regulator, the VL53L5CX breakout boards can be used in any application with a 3.3 V to 5 V supply.

The PCB section supporting the VL53L5CX module is perforated so that developers can break off the mini PCB for use in a 3.3 V supply application using flying leads. This makes it easy to integrate the VL53L5CX-SATEL breakout boards into development and evaluation devices due to their small form factor.

Figure 96. VL53L5CX-SATEL breakout boards



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 + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM®-STR) toolchain + ST-LINK
  - STM32CubeIDE for STM32 + ST-LINK

5.3 Hardware setup

The following hardware components are required:

1. One STM32 Nucleo development platform (suggested order code: NUCLEO-F401RE or NUCLEO-L476RG)
2. An X-NUCLEO-53L5A1 expansion board or a VL53L5CX-SATEL breakout board
3. One USB type A to mini-B USB cable to connect the STM32 Nucleo to a PC

5.4 Software setup

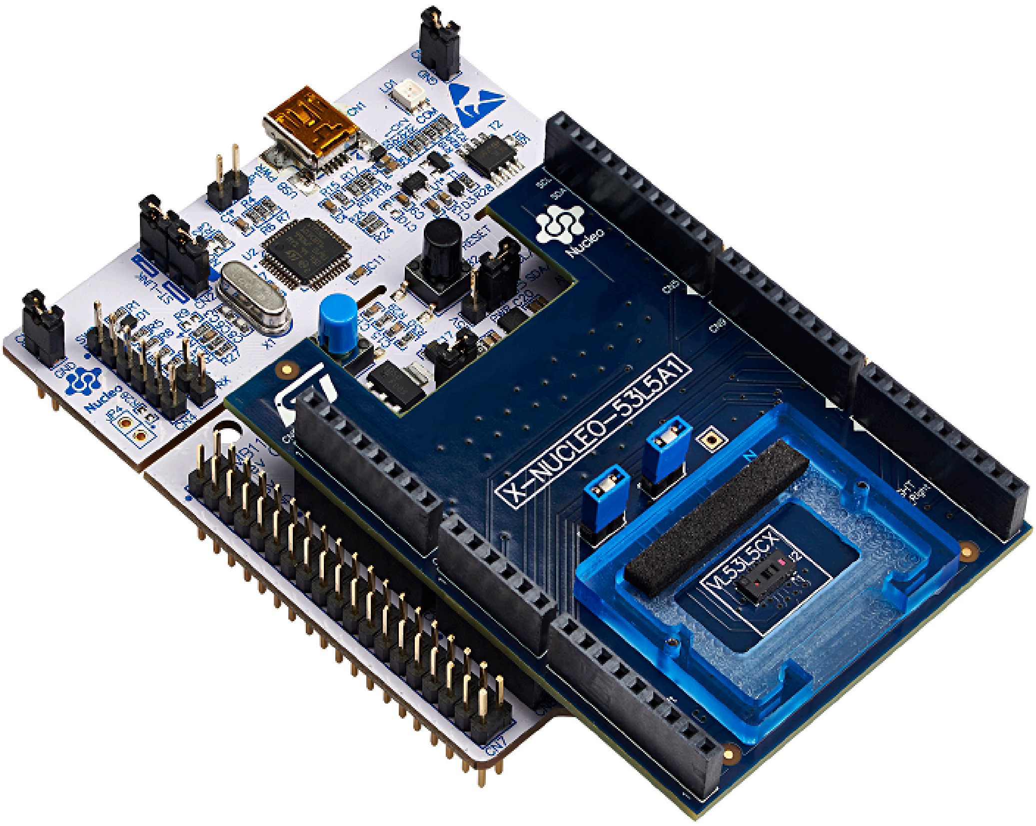
To set up the SDK, run the sample testing scenario based on the GUI utility and customize applications, select one of the integrated development environments supported by the STM32Cube expansion software and follow the system requirements and setup information provided by the IDE provider.

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 and can interface with the external STM32 microcontroller on STM32 Nucleo via the Inter-Integrated Circuit (I²C) transport layer.

Figure 97. Sensor expansion board plugged to STM32 Nucleo board



Revision history

Table 1. Document revision history

Date	Version	Changes
17-Jan-2022	1	Initial release

Contents