Getting started with the STMicroelectronics X-CUBE-TOF1, Time-of-Flight sensors, software package for STM32CubeMX

Introduction

This document provides the guidelines to configure and use the Time-of-Flight sensor software package for CubeMX. The document contains a description of the provided sample applications, as well as a description of the steps to configure and use the sample application provided in the package.

Information and documentation related to the ST Time-of-Flight sensors, the X-NUCLEO-53L3A2 expansion board and the ST expansion software for the Time-of-Flight sensor are available on www.st.com.

Contents

List of figures

1 Acronyms and abbreviations

Table 1: list of acronyms

Acronym	Description
HAL	Hardware Abstraction Layer
IP	Internet Protocol
NVIC	Nested Vectored Interrupt Controller
PCB	Printed Circuit Board
RTOS	Real Time Operating System
U(S)ART	Universal (Synchronous) Asynchronous Receiver Transmitter
USB	Universal Serial BUS
TCP	Transmission Control Protocol
ToF	Time-of-Flight

2 What is STM32Cube?

STM32Cube™ represents an original initiative by STMicroelectronics to ease developers' life by reducing development effort, time and cost. STM32Cube covers the STM32 portfolio. Version 1.x of STM32Cube includes:

- STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards.
- A comprehensive embedded software platform delivered per series (such as the STM32CubeF4 for STM32F4 series).
    - STM32Cube HAL, an STM32 abstraction layer embedded software, ensuring maximized portability across the STM32 portfolio.
    - a consistent set of middleware components, such as RTOS, USB, TCP/IP, graphics.
    - all embedded software utilities, including a full set of examples.

3 License

The software provided in this package is licensed under Software License Agreement SLA0081.

4 Sample Applications Description

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 pre-compiled binaries of the sample applications can be found as shown in the figure below. The user may 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.

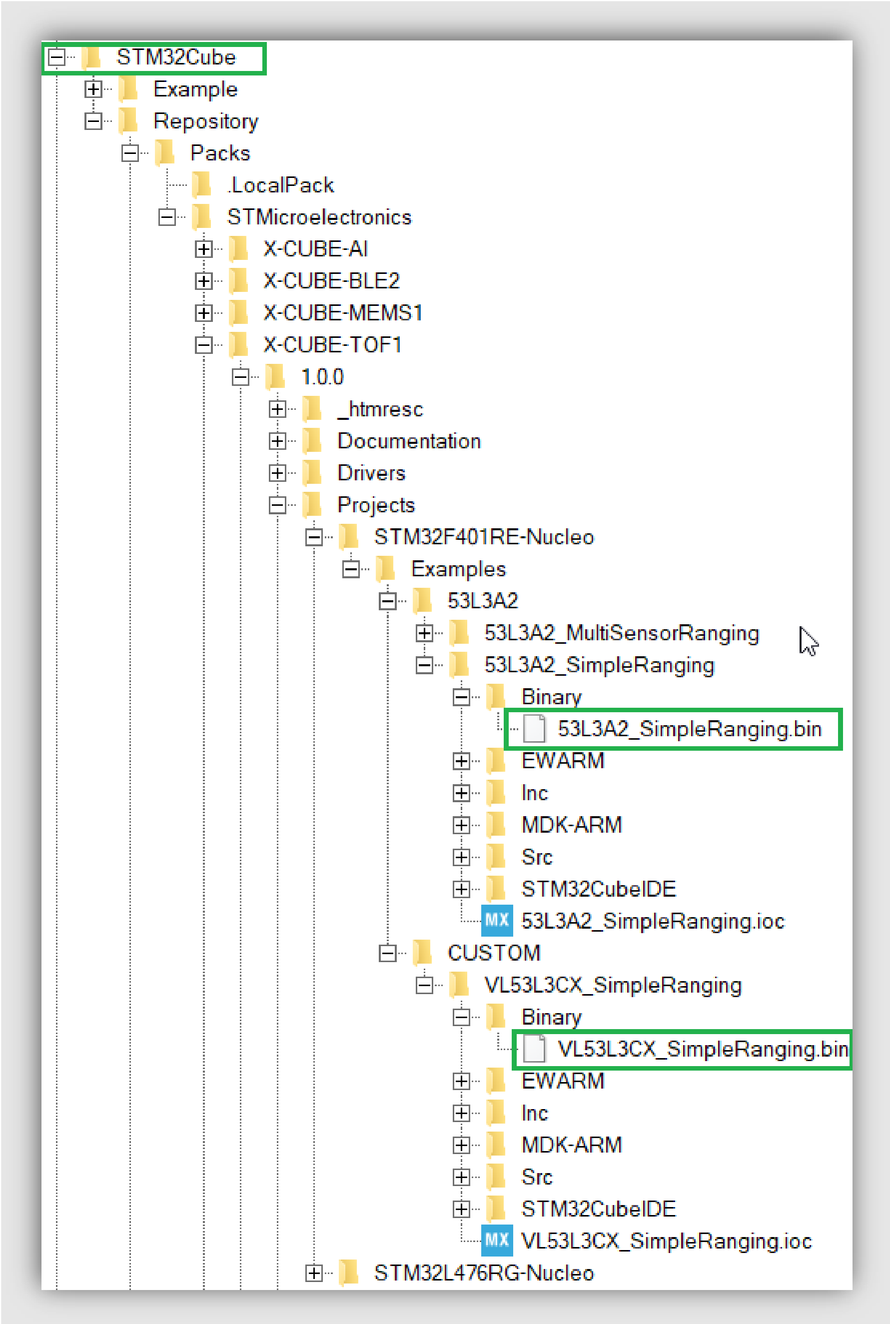


Figure 1: Precompiled projects location

4.1 53L3A2_SimpleRanging

This sample application shows how to use the X-NUCLEO-53L3A2 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 will be displayed on the serial terminal.
The ranging data can be read by polling the device for completion or triggering an interrupt. To select the data reading mode, refer to the *Figure 26: STM32CubeMX-Polling or Interrupt Configuration*.

This application can be run by loading the prebuilt binary 53L3A2_SimpleRanging.bin located as shown in the *Figure 1: Precompiled projects location* or from a new project created with the STM32CubeMX.

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

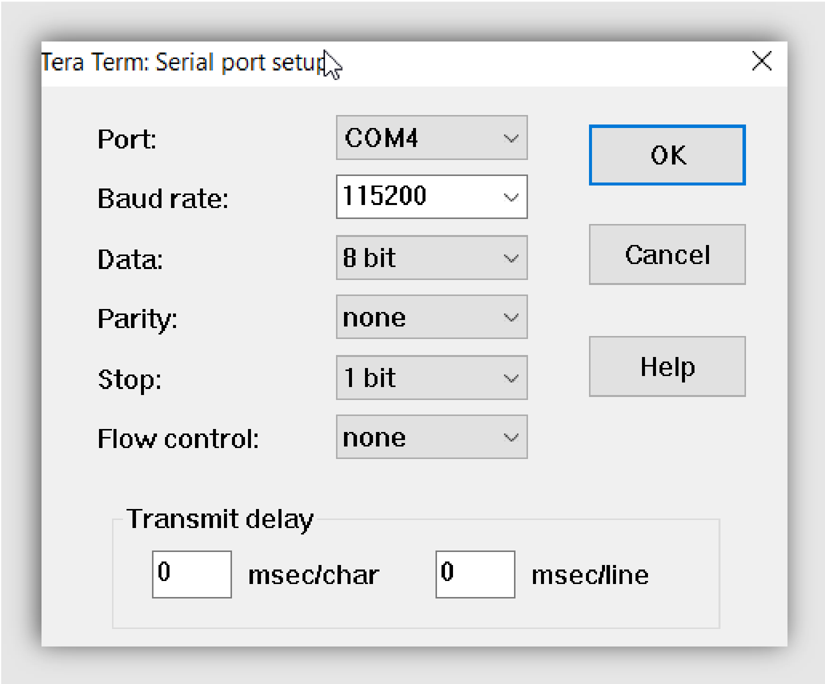


Figure 2: Tera Term, Serial port setup

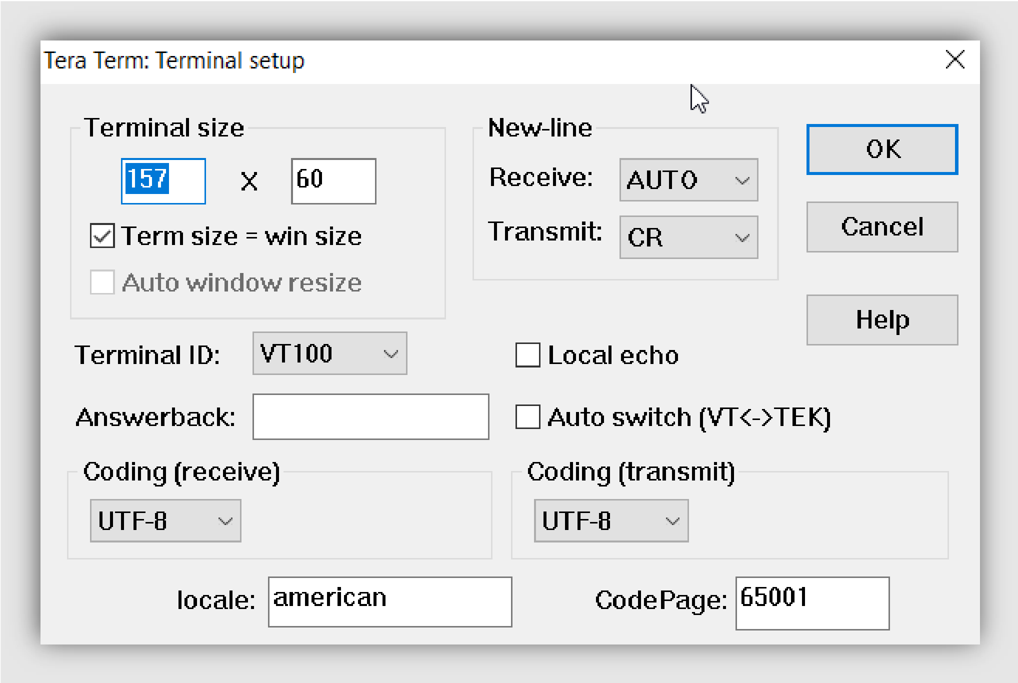


Figure 3: Tera Term, Terminal setup

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

---

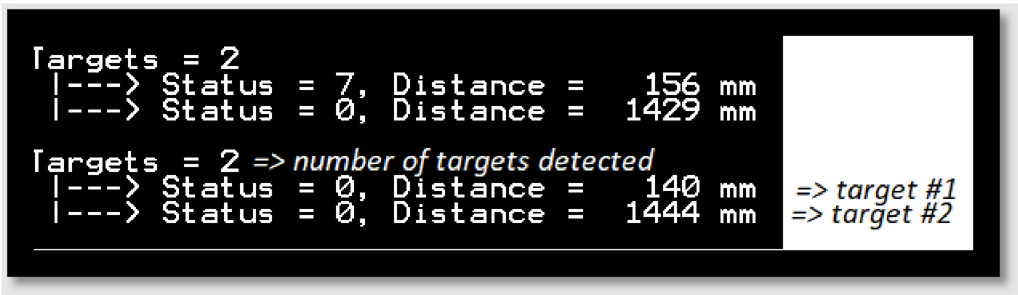


Figure 4: Ranging data

Note: remove the protection film on the top of the ToF at first use.

4.2 Offset and Xtalk Calibration applications

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

Select and configure the 53L3A2_SimpleRanging application in the software pack as shown in 5.2

Complete the application configuration by selecting the calibration options as shown below

---

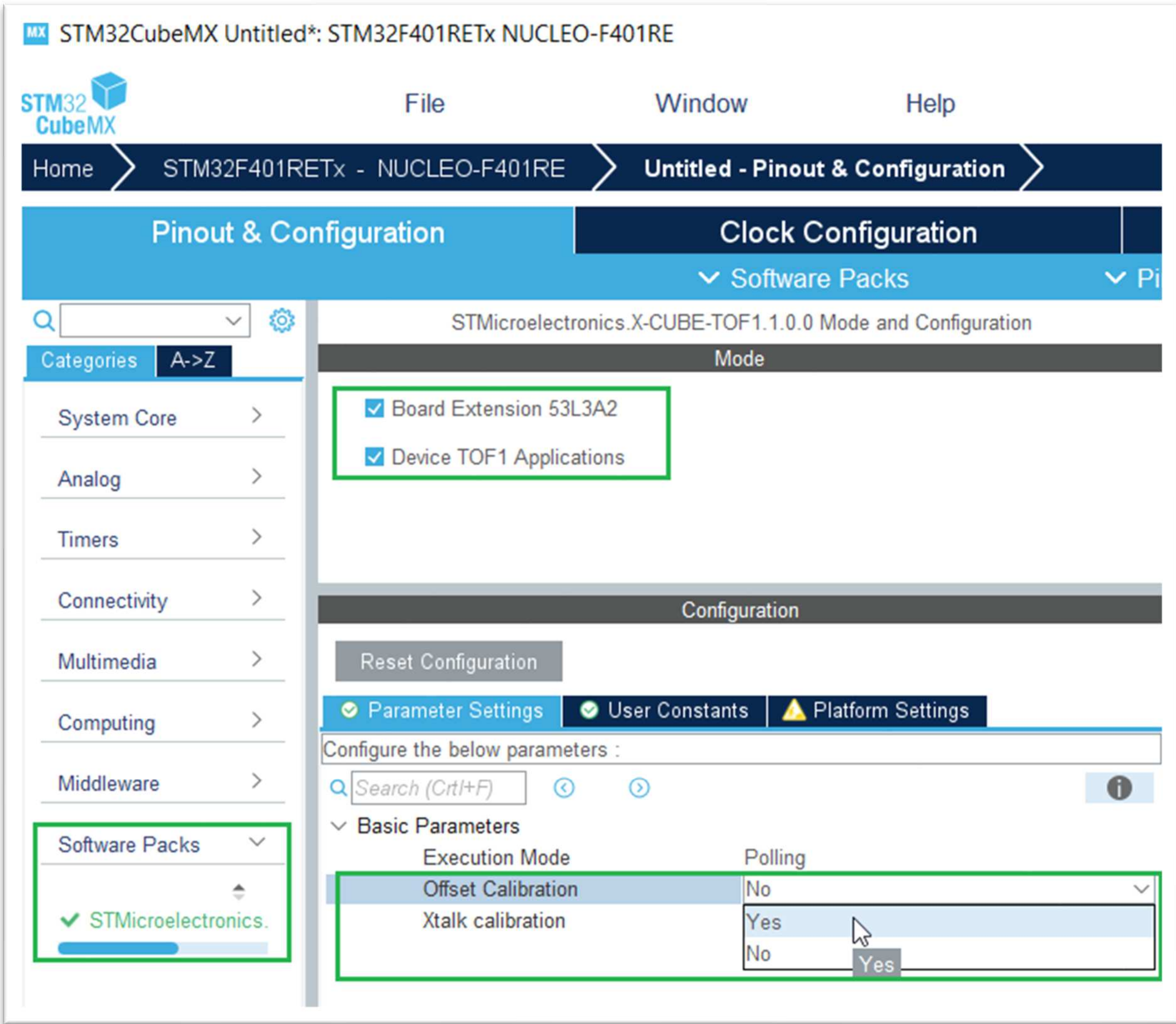


Figure 5: STM32CubeMX-Offset and Xtalk calibration

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. Run the application from the project generated through the STM32CubeMX (*it is the only way*) and follow the instructions displayed on the serial terminal as shown below to perform the calibrations.

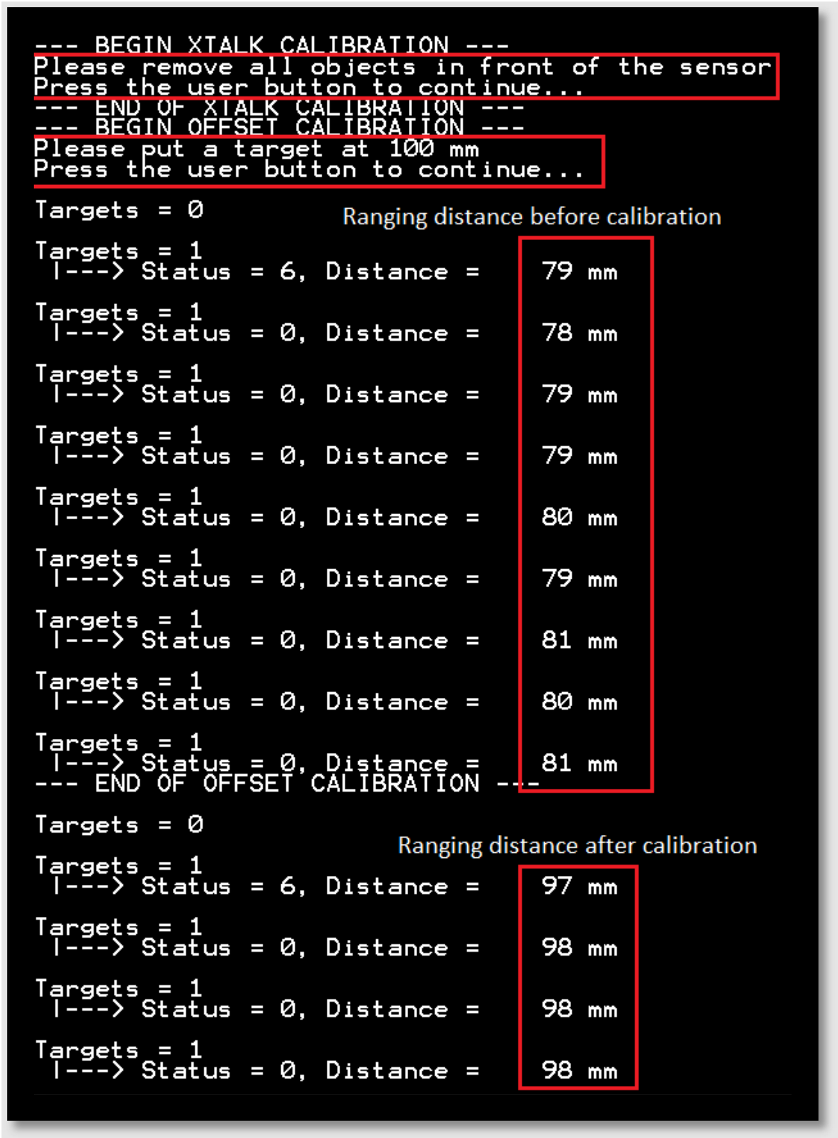


Figure 6: Calibration

4.3 53L3A2_MultiSensorRanging

This sample application shows how to make 3 ToF sensors running simultaneously.
To test this application, two satellite boards VL53L3CX-SATEL, a X-NUCLEO-53L3A2 and a STM32 Nucleo are required. In this example the ranging data will be 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 here.

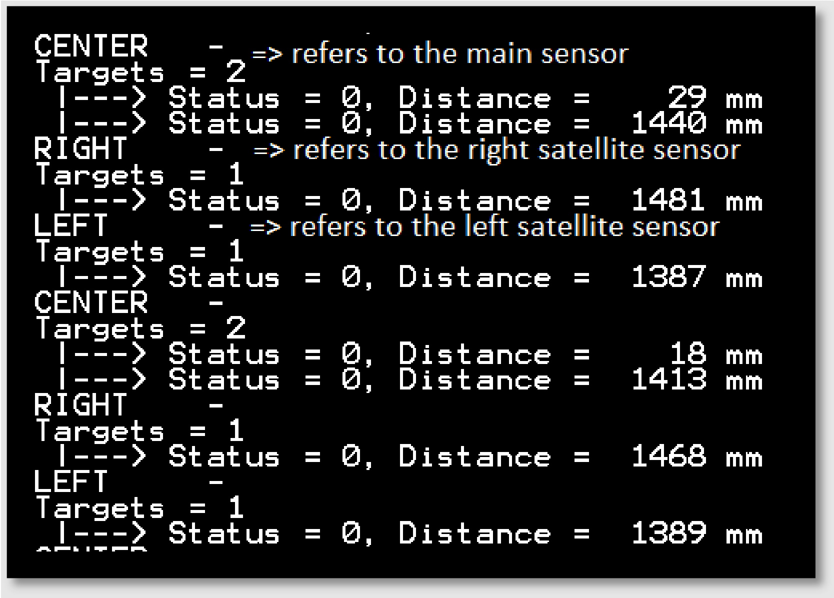


Figure 7: Multiple sensors ranging

4.4 VL53L3CX_SimpleRanging

This example application shows how to make ranging with a satellite board (VL53L3CX-SATEL) connected directly to a STM32 Nucleo without the X-NUCLEO-53LA3 expansion board. To test this application, a VL53L3CX-SATEL (see note below) and a STM32 Nucleo are required. In this example the ranging data will be displayed on the serial terminal as shown below.

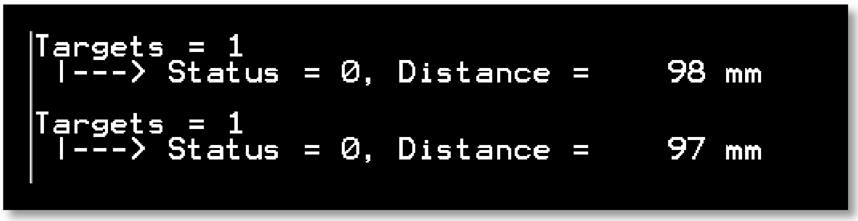


Figure 8: Ranging with a satellite

This application can be launched by loading the prebuilt binary *VL53L3CX_SimpleRanging.bin* or from a new project created with STM32CubeMX.
When create a new project with STM32CubeMX, make sure that the GPIOs and I2C are correctly configured. To create this application with STM32CubeMX follow the steps shown in the figures from *Figure 28 to Figure 30*.

Note : Unless adding two 4.7k pull up resistances on the STM32 Nucleo SCL and SDA pins, the satellite board (blue color PCB), provided with the X-NUCLEO-53L3A2 pack, cannot be used directly on the STM32 Nucleo. For this application, the VL53L3CX-SATEL breakout board (green color PCB) is required and may be purchased separately from any electronics distributor.

Connect the satellite board to the Nucleo board as shown below

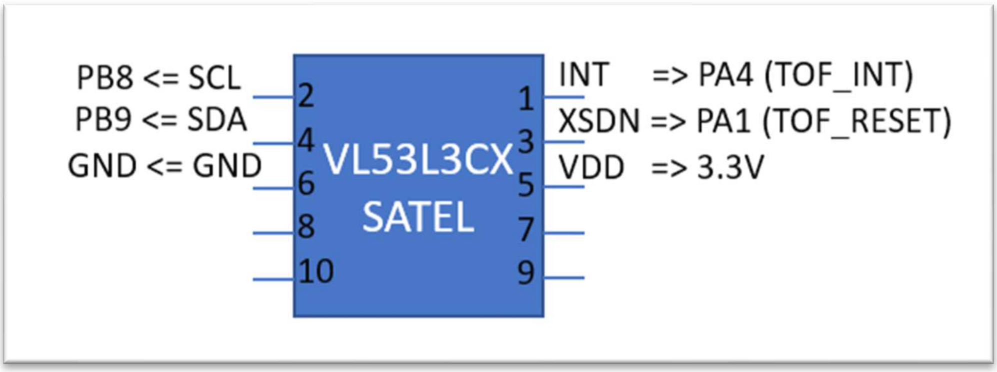


Figure 9: VL53L3CX-SATEL to F401RE connection

5 Configuration Steps

The X-NUCLEO-53L3A2 interfaces with the STM32 microcontroller via the I2C pin.
Hence, assuming the user wants to interface the X-NUCLEO-53L3A2 expansion board with a STM32 Nucleo 64 pins board (e.g. a Nucleo-F401RE), no particular hardware modification must be done. The X-NUCLEO-53L3A2 pin out is shown in the *Figure 11: X-NUCLEO-53L3A2 pinout*

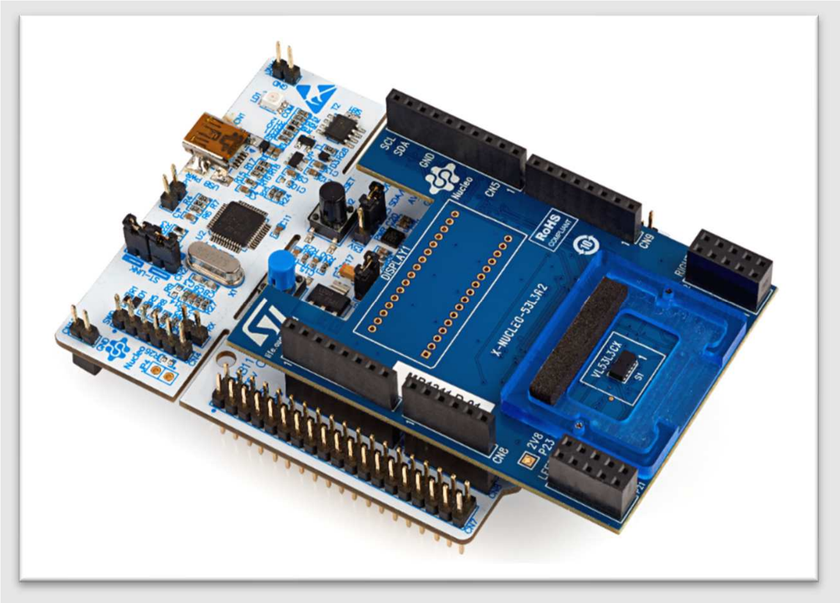


Figure 10: STM32 Nucleo 64 pins and X-NUCLEO-53L3A2

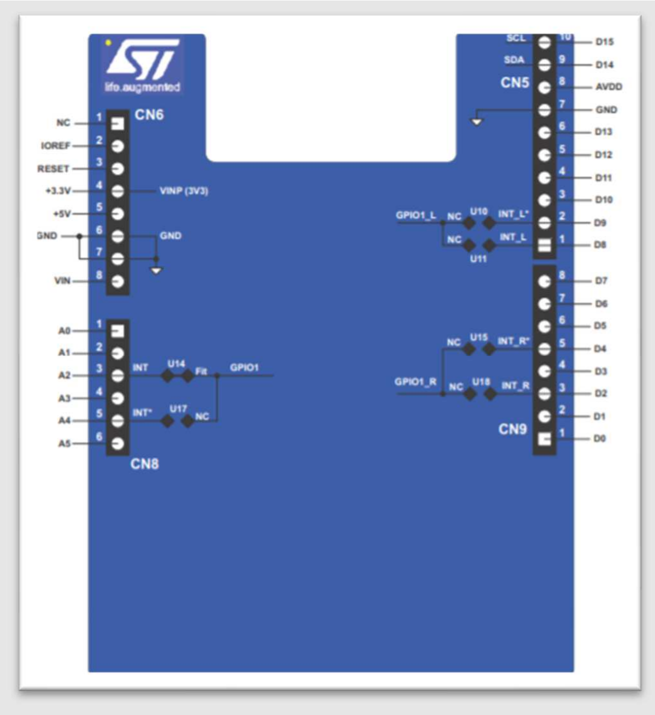


Figure 11: X-NUCLEO-53L3A2 pinout

5.1 Use of Expansion Software without sample applications

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

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

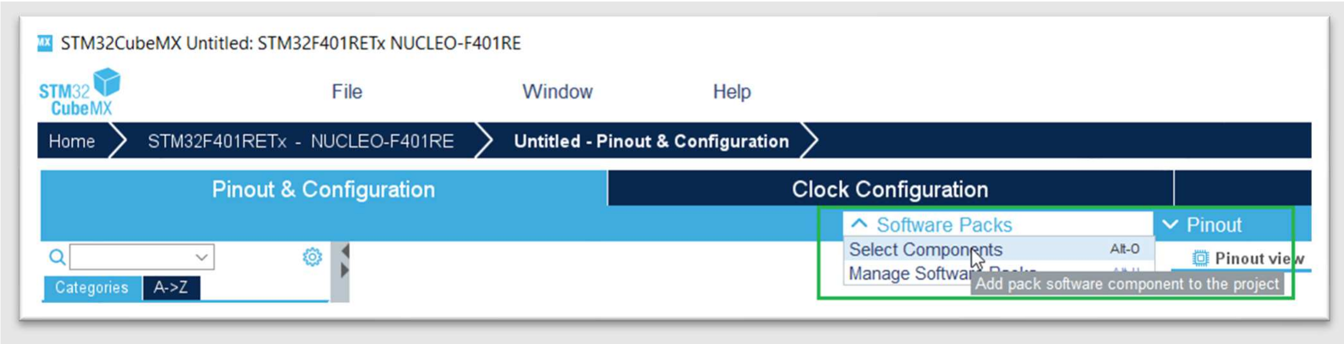


Figure 12: STM32CubeMX-Select Components

From the “Software Packs Component Selector” window, select only the “Board Extension” class

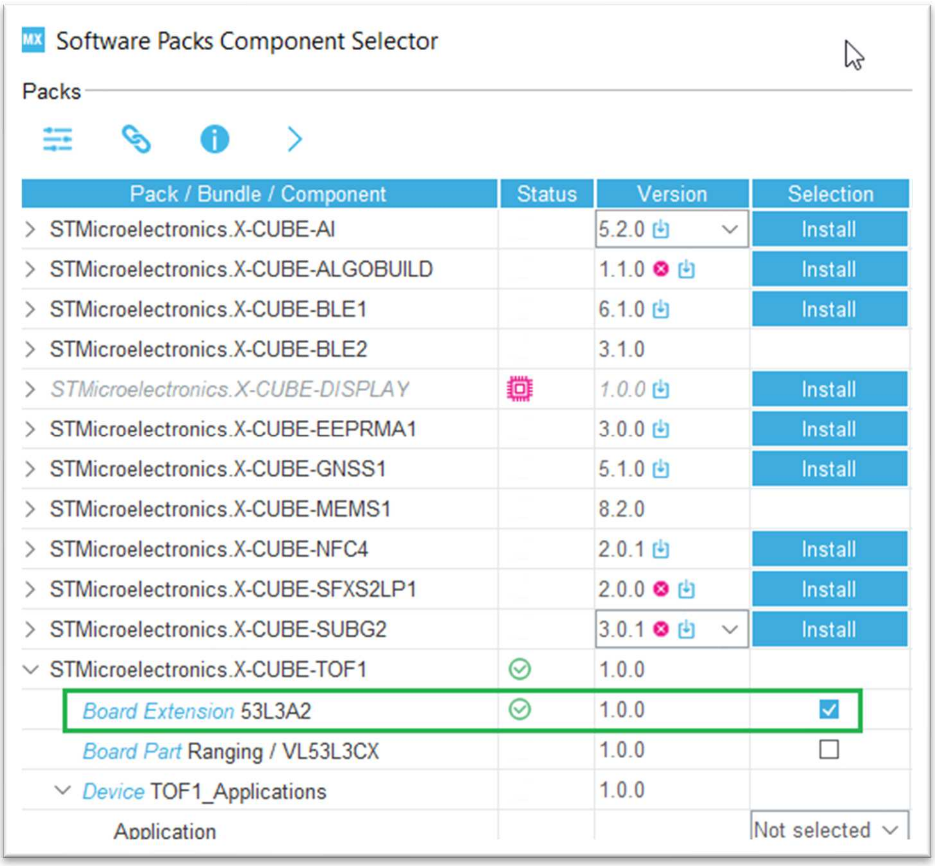


Figure 13: STM32CubeMX-Select Board Extension Only

Enable I2C1 as shown below

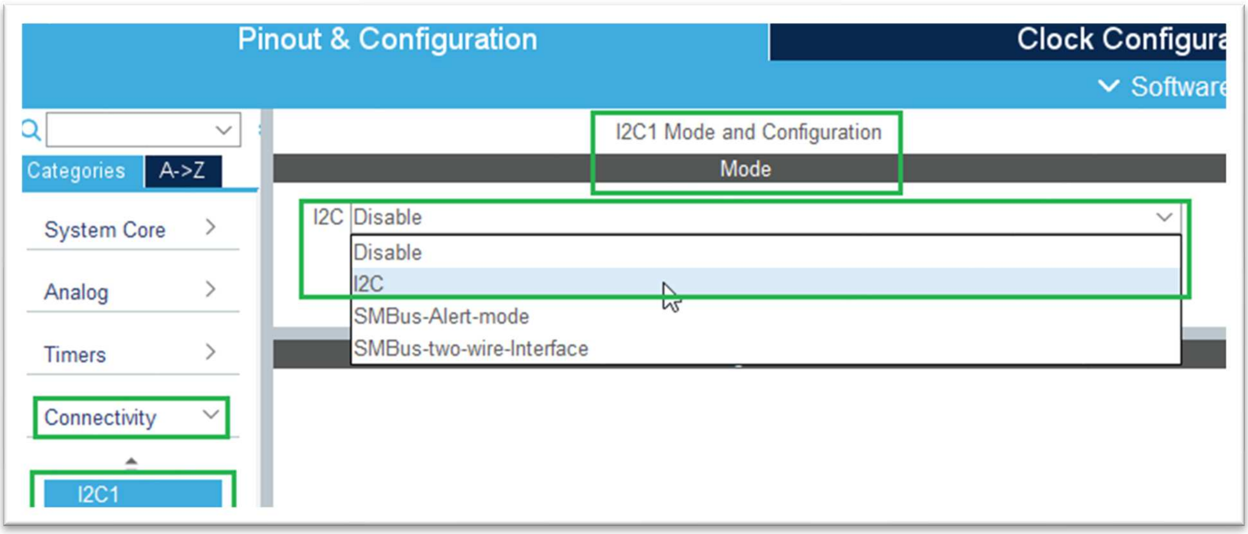


Figure 14: STM32CubeMX-I2C configuration

From the Software Packs category, press on the 'STMicroelectronics.X-CUBE-TOF1' item.

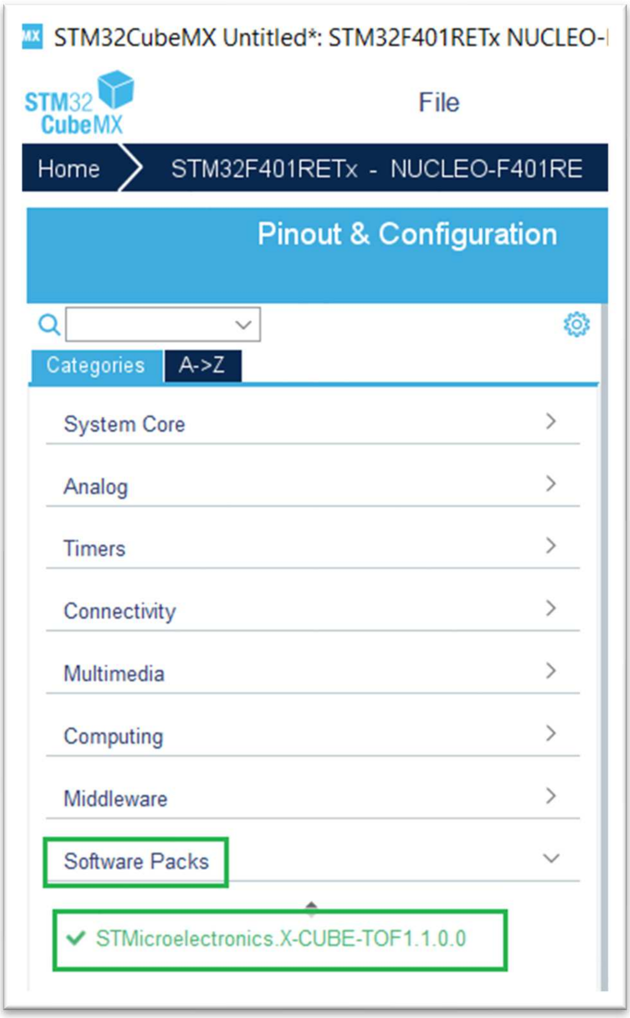


Figure 15: STM32CubeMX-Software packs

From the "Mode" view, enable the "Board Extension 53L3A2".

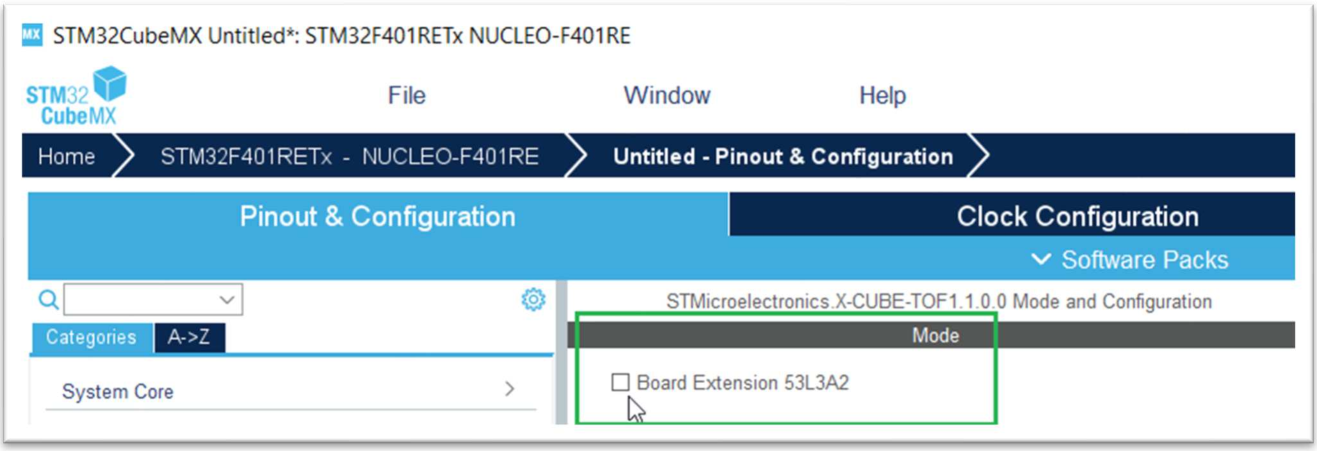


Figure 16: STM32CubeMX-Mode view

From the "Configuration" window, enable the I2C1

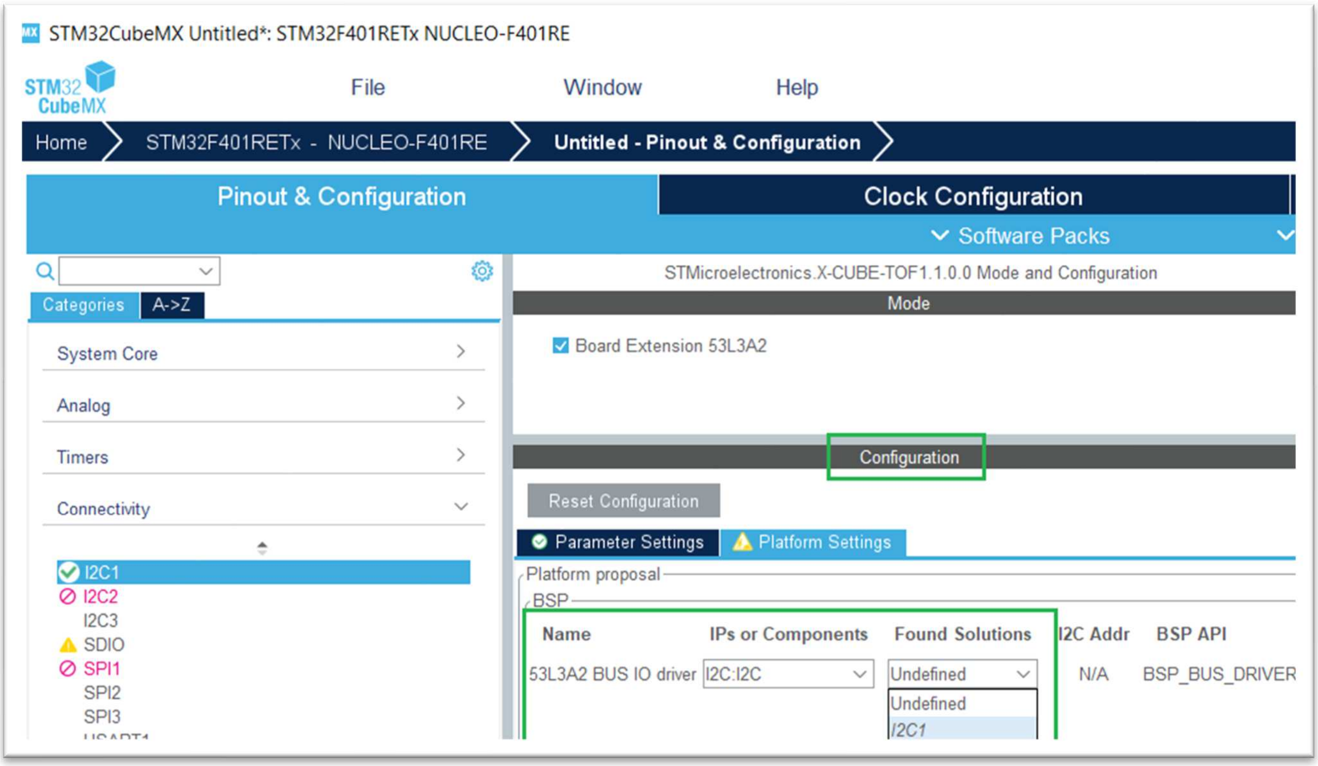


Figure 17: STM32CubeMX-Configuration window

Once all the described steps above have been performed click on "Project Manager" to name the project and select the Toolchain/IDE for which one the code will be generated.

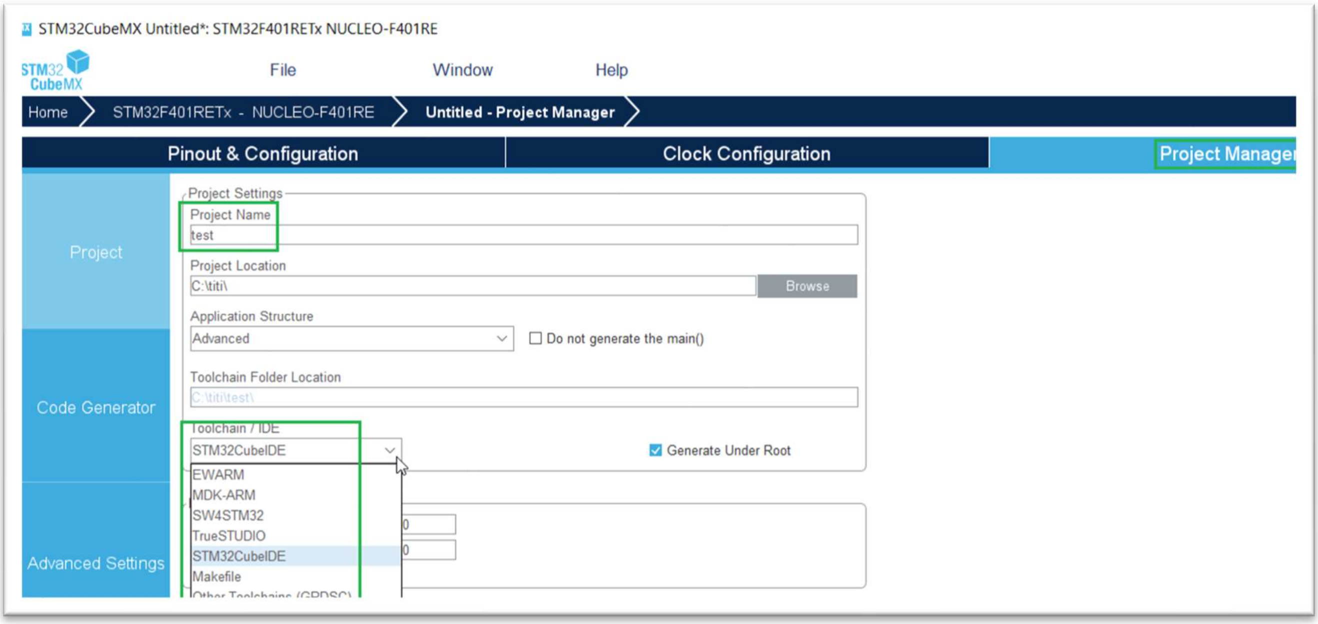
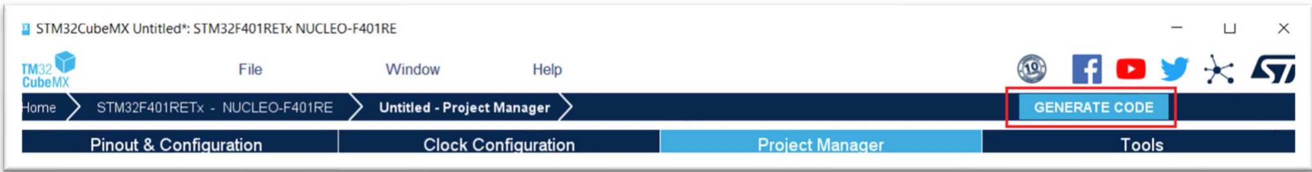


Figure 18: STM32CubeMX-Project manager

Generate the source code of the project using the X-CUBE-TOF1 software by clicking on the "GENERATE CODE" button.



5.2 Use of Expansion Software with sample applications for X-NUCLEO-53L3A2

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

From the Table 1: Pinout Scheme, if not already set, set as shown in the figures below

Nucleo 64			**Nucleo 144**		
PIN	*Mode*	*Label*	*PIN*	*Mode*	*Label*
PB8	I2C1_SCL	SCL	PB8	I2C1_SCL	SCL
PB9	I2C1_SDA	SDA	PB9	I2C1_SDA	SDA
PA4	GPIO_EXTI4	TOF_INT	PC3	GPIO_EXTI3	TOF_INT
PC13	GPIO_EXTI13	B1 [Blue Pushbutton]	PC13	GPIO_EXTI13	USER_Btn[B1]
PA5	GPIO_Output	LD2 [Green Led]	PB7	GPIO_Output	LD2[Blue]
PA2	USART2_TX	USART_TX	PD8	USART3_TX	USART_TX
PA3	USART2_RX	USART_RX	PD9	USART3_RX	USART_RX

Table 1: Pinout Scheme

The UART2 and LD2 [Green LED] are already set by default.
The B1 [Blue Pushbutton] is required only for the calibration applications, refer to 4.2

Enable and configure the I2C1

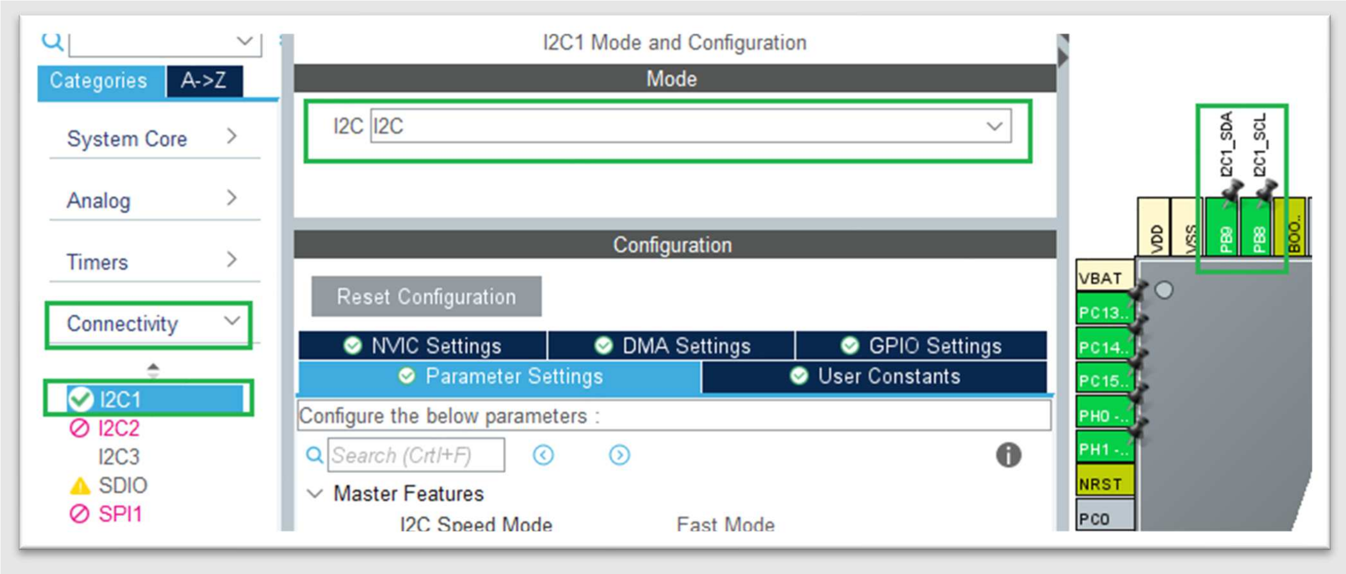


Figure 19: STM32CubeMX-Pinout & Configuration, I2C

Configure the TOF_INT (GPIO_EXTI4)

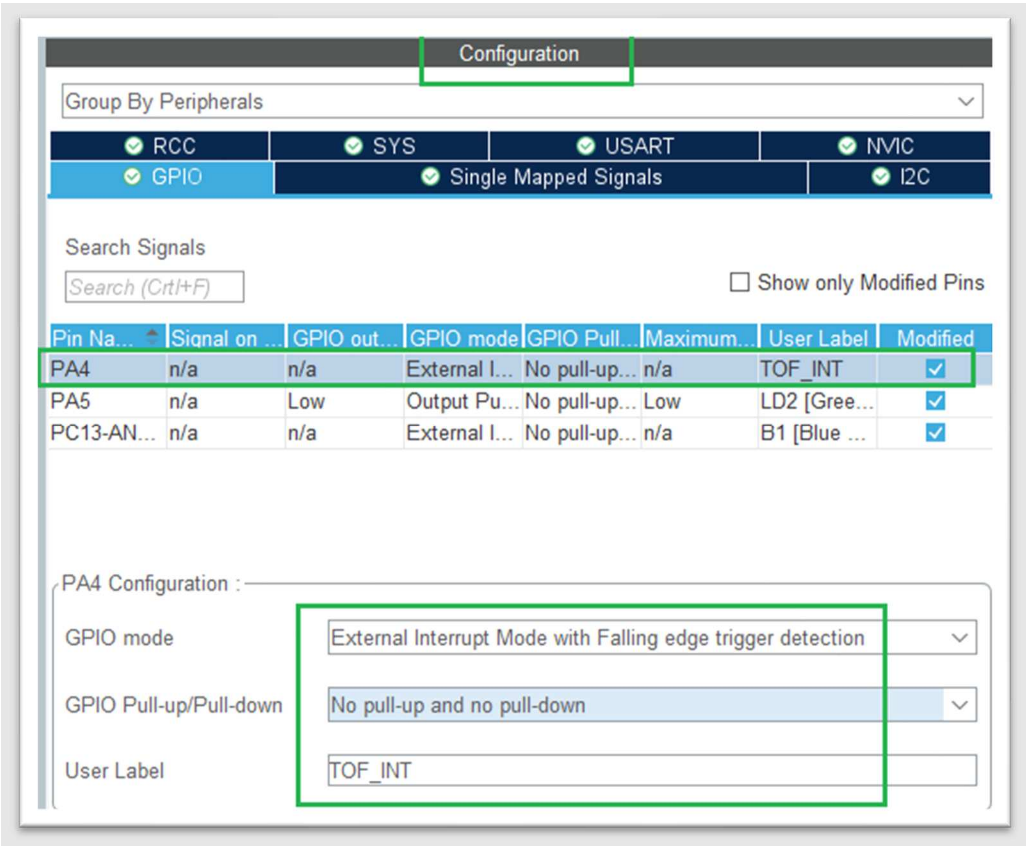


Figure 20: STM32CubeMX-Pinout & Configuration-GPIO_EXTI4, TOF_INTERRUPT

Enable the NVIC (TOF_INT and Blue PushButton)

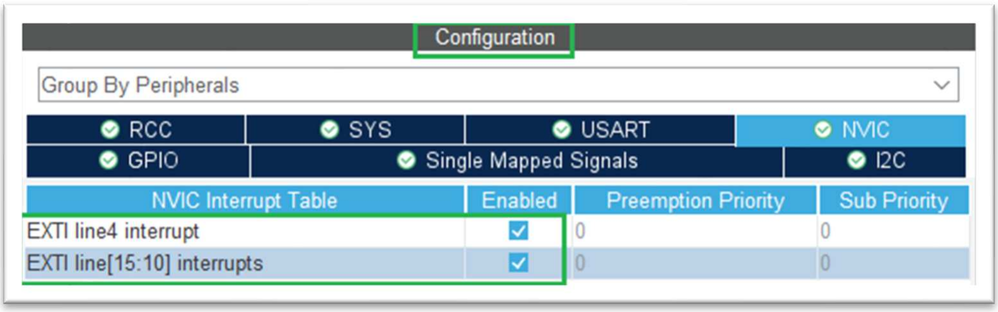


Figure 21: STM32CubeMX-Pinout & Configuration, TOF_INTERRUPT NVIC enable

Now, add the X-CUBE-TOF1 software pack to the project and create a new application based on the 53L3A2_SimpleRanging.

Click on "Select Components" from "Software Packs"

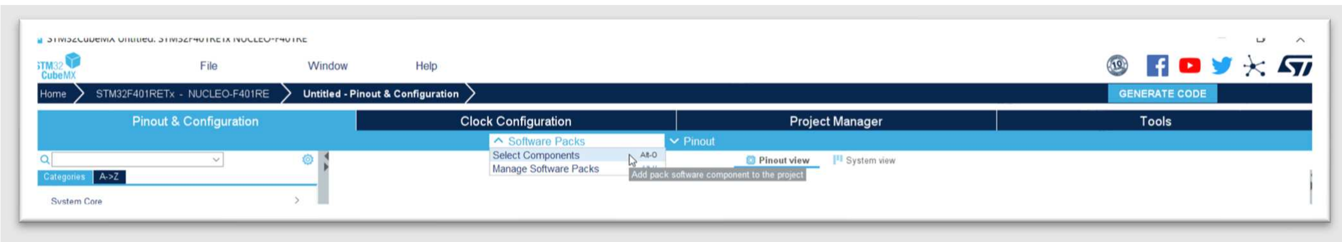


Figure 22: STM32CubeMX-Software Packs

Select the "Board Extension" class and an application from the "Device" class as shown in the figure below, then click on "Ok".

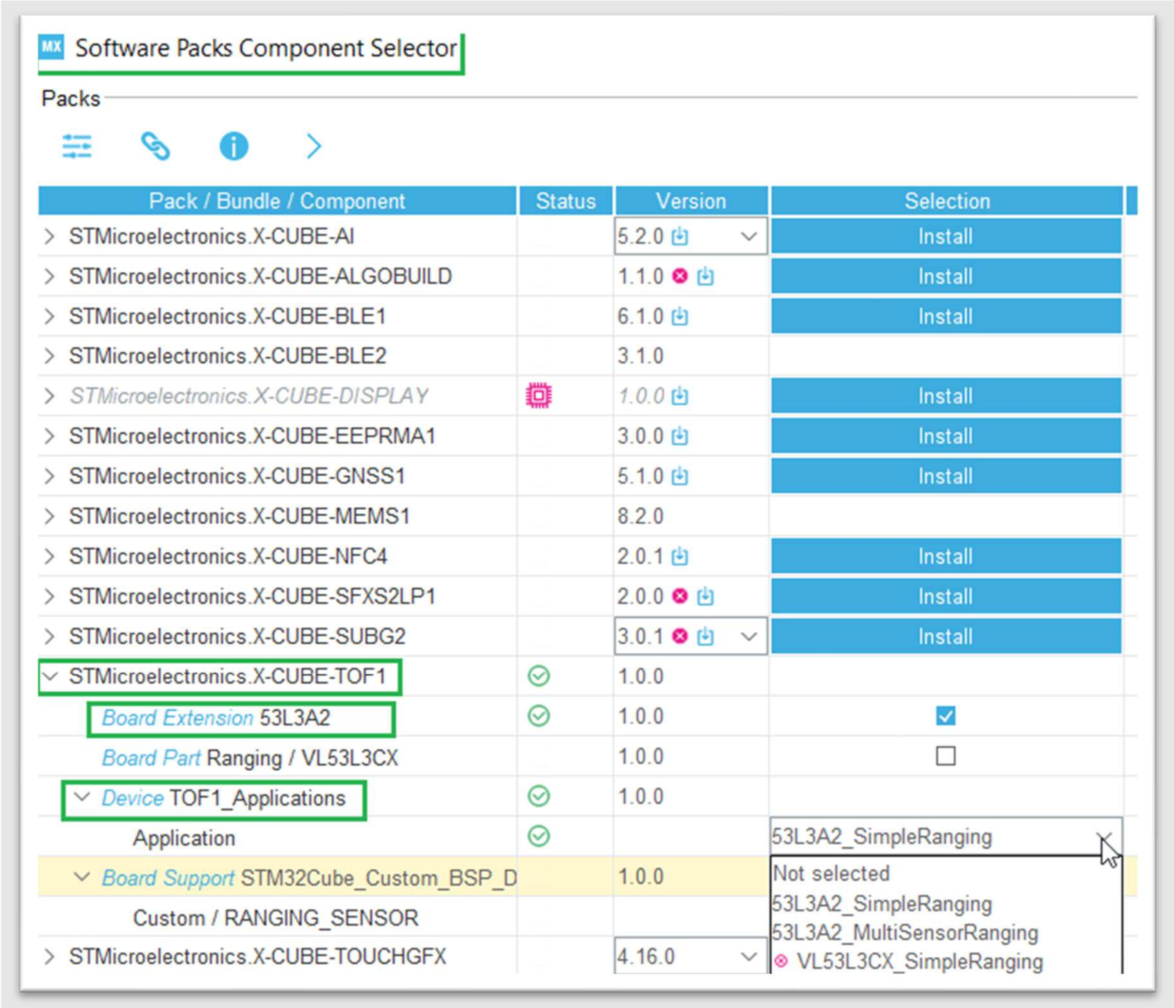


Figure 23: STM32CubeMX-Software Packs Component Selector for the 53L3A2

Once the board and the application have been selected, configure the Board and the Application as shown in the figure below.

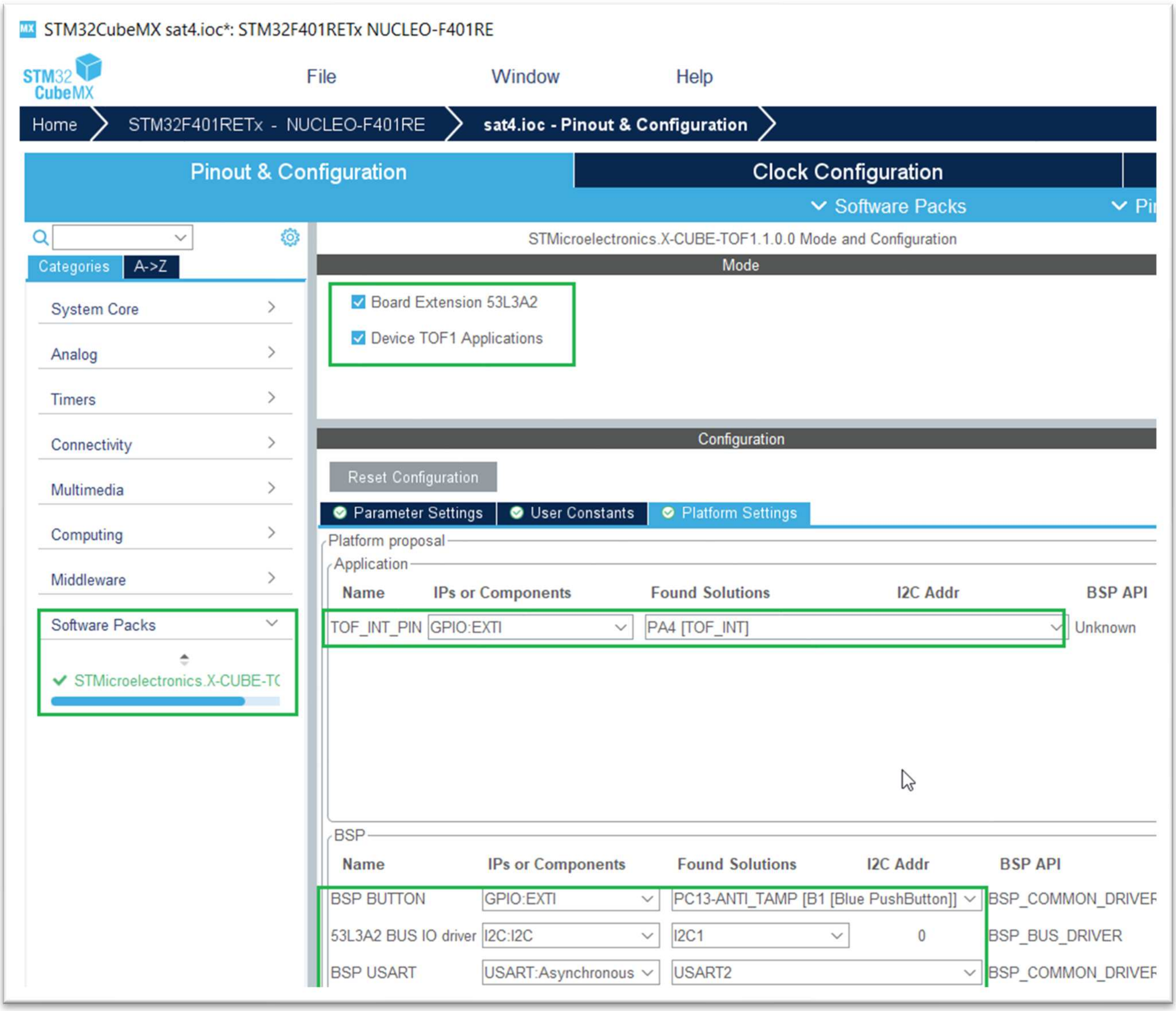


Figure 24: STM32CubeMX-Board and Application Configuration 1/2

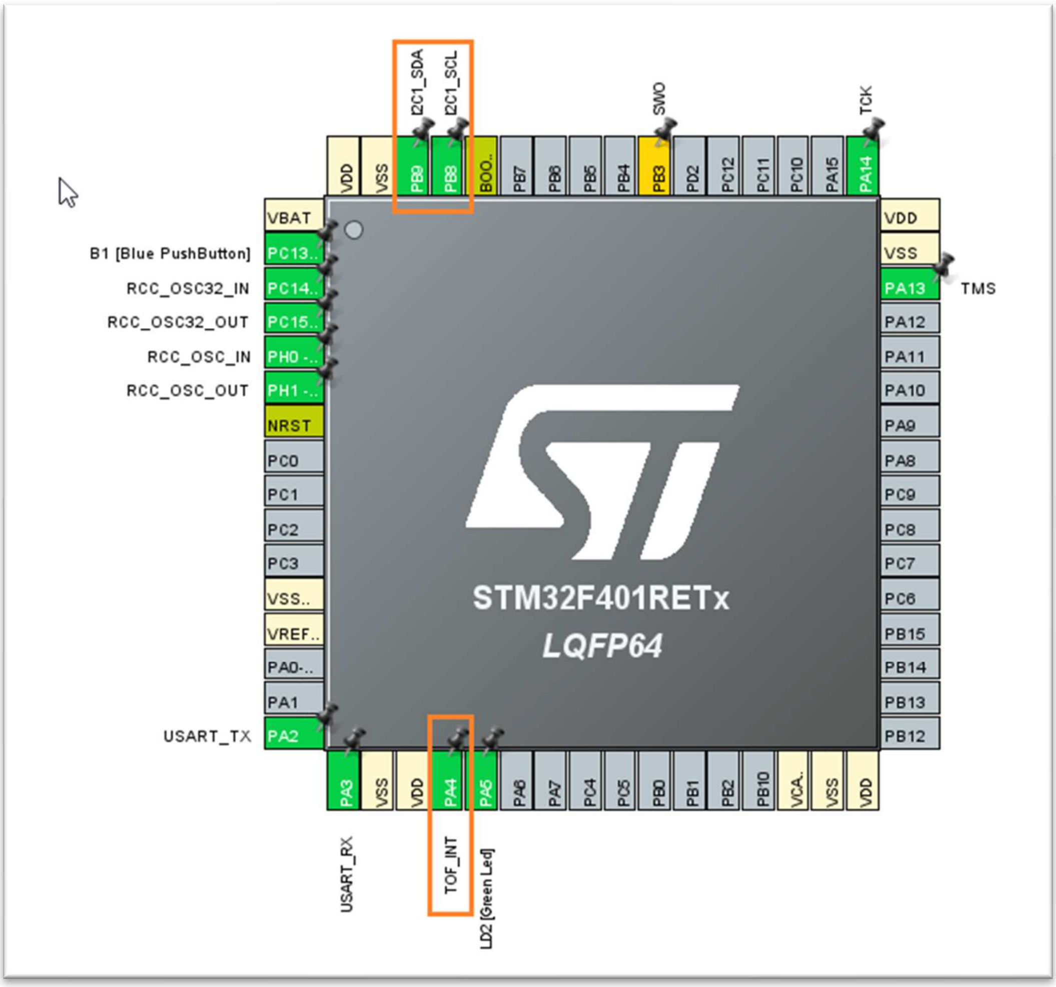


Figure 25: STM32CubeMX-Board and Application Configuration 2/2

THE TOF SENSOR RANGING DATA CAN BE READ EITHER IN POLLING MODE OR IN INTERRUPT MODE BY SELECTING THE OPTION SHOWN IN THE FIGURE BELOW. IF RANGING BY INTERRUPT THEN "INTERRUPT" BE SELECTED, OTHERWISE "POLLING" MUST BE SELECTED.

Select the data reading mode as shown below

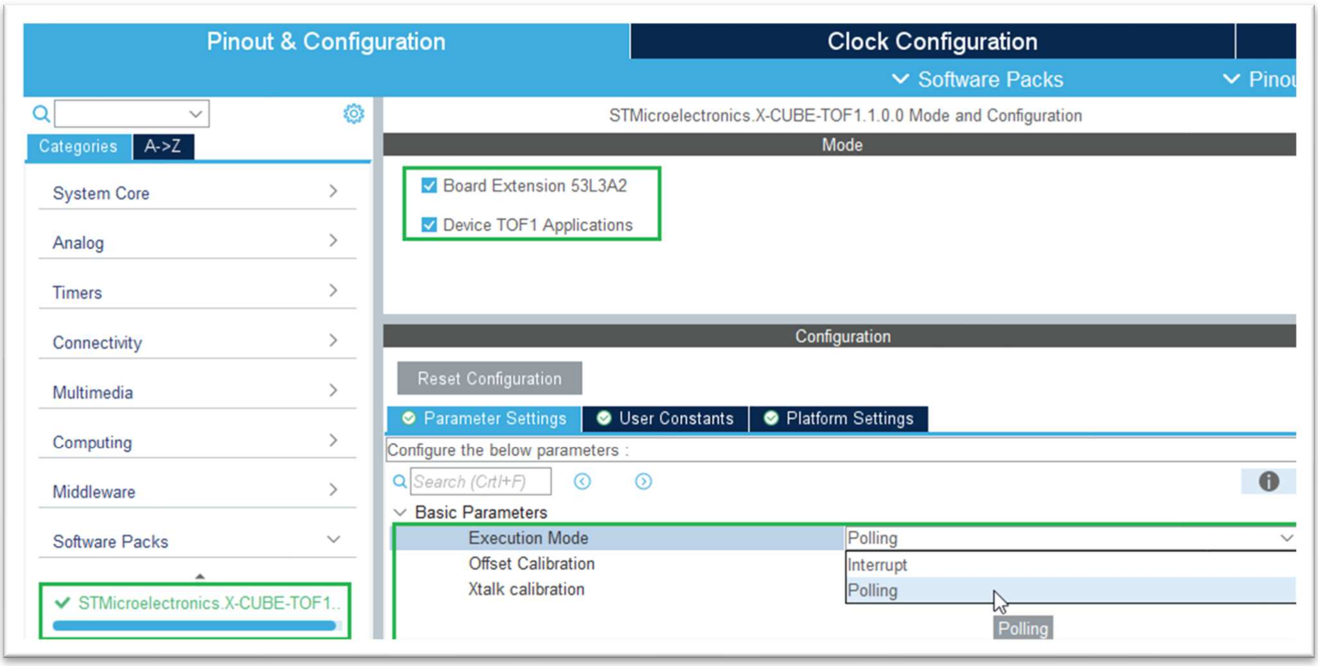


Figure 26: STM32CubeMX-Polling or Interrupt Configuration

Once all the described steps above have been performed, generate the source code of the project by clicking on the "GENERATE CODE" button after naming the project and selecting the toolchain/IDE as shown in the figure below.

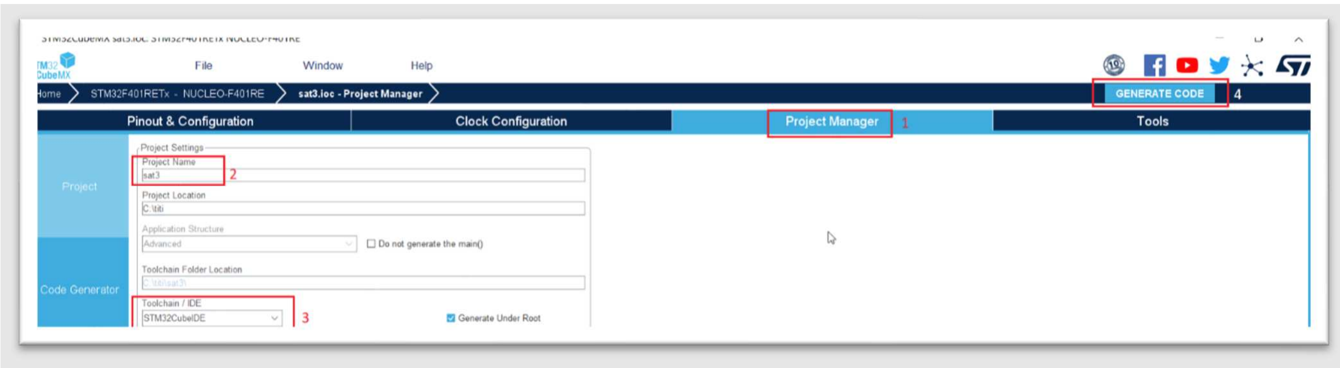


Figure 27: STM32CubeMX-Generate Code

IN THIS SECTION, THE FOLLOWING FIGURES ARE USED TO ILLUSTRATE THE DIFFERENT STEPS TO FOLLOW TO GENERATE THE RANGING APPLICATION *with the satellite connected directly to the STM32 Nucleo*.

Select the "Board Extension" class and an application from the "Device" class as shown in *the figure above, then click on "Ok".*

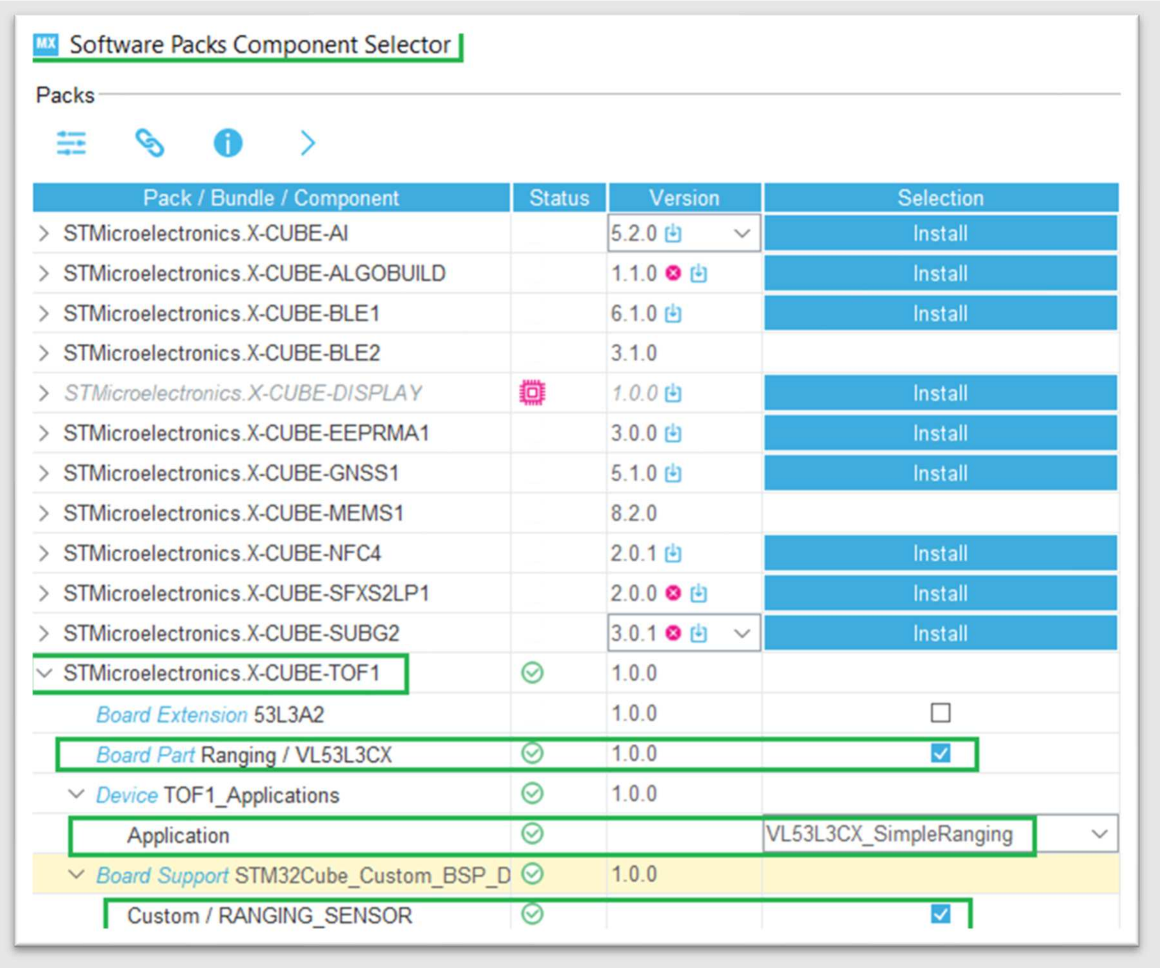


Figure 28: STM32CubeMX-Ranging with satellite (VL53L3CX_SimpleRanging)

Once the board and the application have been selected, configure the Board and the Application as shown in the figure above.

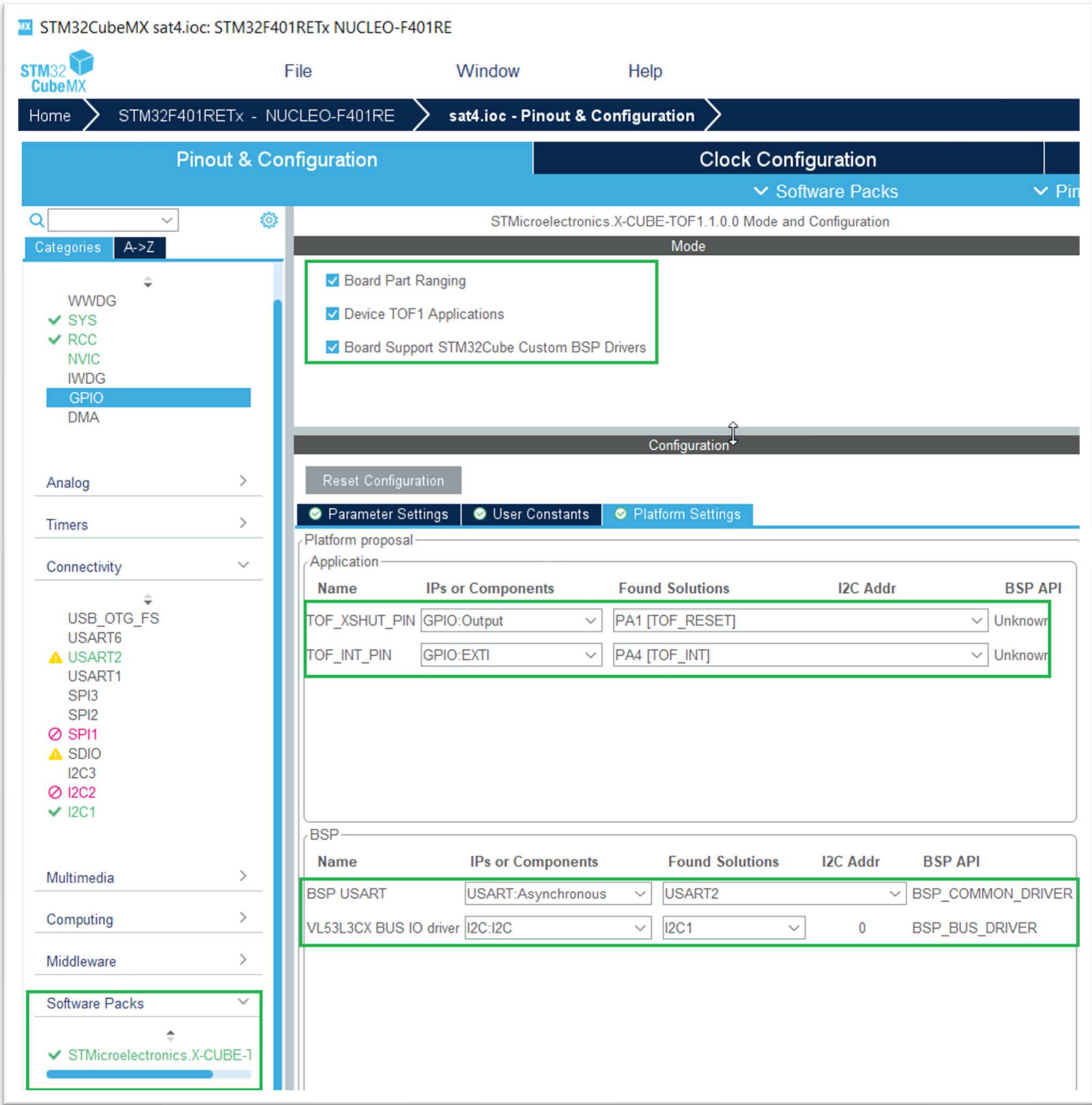


Figure 29: STM32CubeMX-VL53L3CX_SimpleRanging-GPIOs and I2C configuration

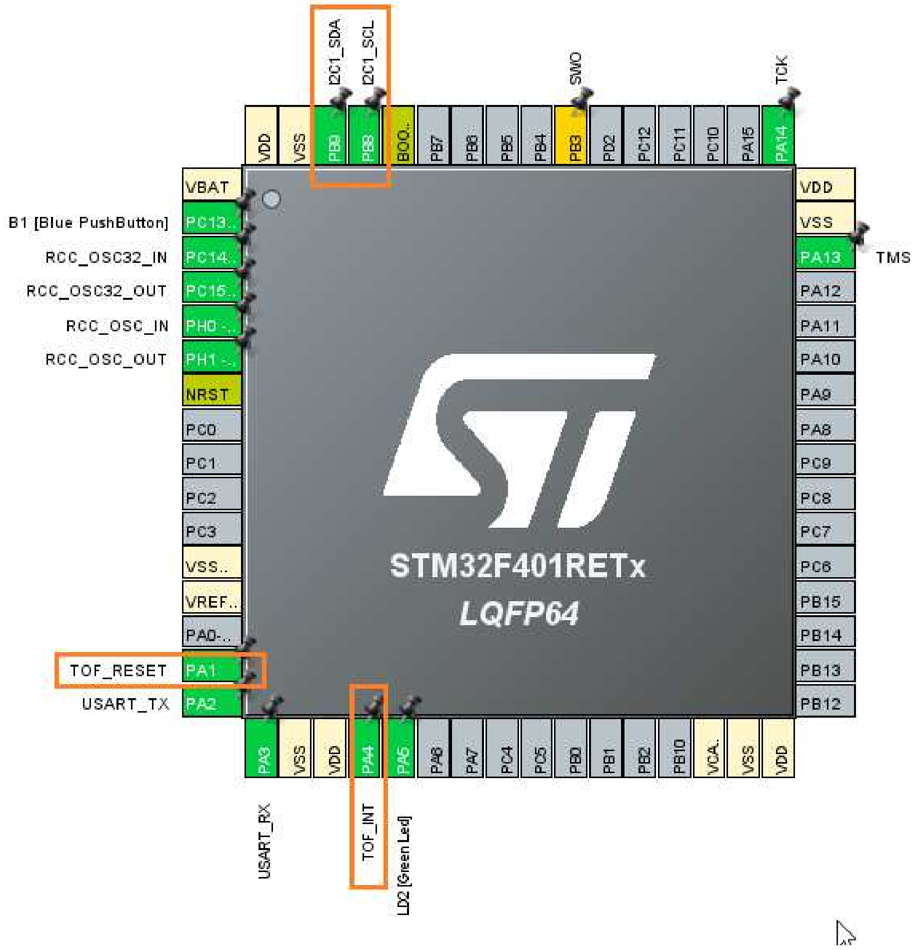


Figure 30: STM32CubeMX-VL53L3CX_SimpleRanging-Pinout Top view

---

Once all the described steps above have been performed, generate the source code of the project by clicking on the "GENERATE CODE" button after naming the project and selecting the toolchain/IDE as shown in the Figure 27: STM32CubeMX-Generate Code

---

6 References

[1] UM2778 – User Manual - VL53L3CX Time-of-Flight ranging module with multi object detection driver.

7 Revision history

Table 2: Document revision history

Date	Version	Changes
12-Mars-2021	1	Initial release.