Getting started with the STMicroelectronics X-CUBE-NFC7 software package for STM32CubeMX

Introduction

This document provides the guidelines to configure and use the X-CUBE-NFC7 software package V1.0.0 for STM32CubeMX (minimum required version V6.3.0). The document contains a description of the provided sample applications, a description of the steps required to configure a generic project using the X-NUCLEO-NFC07A1 expansion board with a Nucleo board or ST25DVXXKC component with custom boards, as well as a description of the steps to configure and use the sample applications provided in the package.

Information and documentation related to the NFC components, the X-NUCLEO-NFC07A1 expansion board and the ST expansion software for NFC tag are available on www.st.com.

Contents

List of figures

1 Acronyms and abbreviations

Table 1: list of acronyms

Acronym	Description
NFC	Near Field Communication
NDEF	NFC data exchange format
RFID	Radio frequency identification
HAL	Hardware Abstraction Layer
I2C	Inter-Integrated Circuit
NVIC	Nested Vectored Interrupt Controller
RTC	Real Time Operating System
RTOS	Serial Peripheral Interface
U(S)ART	Universal (Synchronous) Asynchronous Receiver Transmitter
URI	Uniform resource identifier

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 SLA0095.

4 Sample Applications and Examples Description

In this section, a short overview of the sample applications and examples included in the X-CUBE-NFC7 pack is provided.
The sample applications/examples:

- are ready-to-use projects that can be generated through the STM32CubeMX for any Nucleo board and using the X-NUCLEO-NFC07A1 expansion board.
- show the users how to use the APIs to correctly initialize and use the dynamic NFC/RFID tag IC (ST25DVXXKC device).

4.1 NFC07A1_NDEF_URI

This application shows how to use the X-NUCLEO-NFC07A1 write an NDEF message to the ST25DVXXKC EEPROM using the NDEF lib middleware. The yellow LED is switched ON when the message has been successfully written.

4.2 NFC07A1_EnergyHarvesting

This sample shows how to enable the energy harvesting. You can either enable the EH dynamic (step 1) or static (step 2) register:

1. Press the user button to enable the EH dynamic register and allow energy harvesting until the chip is reset The EH dynamic is enabled and allows energy harvesting until the chip is reset. The blue LED is switched ON when the dynamic register is enabled.
2. Press the user button for more than 2 seconds. The EH static register is enabled and allows energy harvesting by default each time the chip is powered. The green LED is switched ON when the static register is enabled.

4.3 NFC07A1_GeneralPurposeOutput

This example shows how to enable and use the GPO.
After initialization, an interrupt is programmed to detect field changes in proximity of the

ST25DVXXKC. The green LED is switched ON when the field is detected and switched OFF when the field disappears.

4.4 NFC07A1_I2CPROTECTION

This example shows how to create areas in the ST25DVXXKC and how to protect them. Text is displayed on a UART console (via ST-LINK) if a PC is connected. The example serial settings can be configured by user changing the settings of USART2 in the STM32CubeMX GUI.

4.5 NFC07A1_Mailbox

This example shows how to write a message into the mailbox and how to read mailbox status register of ST25DVXXKC device. Text is displayed on a UART console (via ST-LINK) if a PC is connected. The example serial settings can be configured by user changing the settings of USART2 in the STM32CubeMX GUI.

5 Installing the X-CUBE-NFC7 pack in STM32CubeMX

After downloading (from www.st.com), installing and launching the STM32CubeMX (V≥5.4.0), the X-CUBE-NFC7 pack can be installed in few steps.

1. From the menu, select Help > Manage embedded software packages

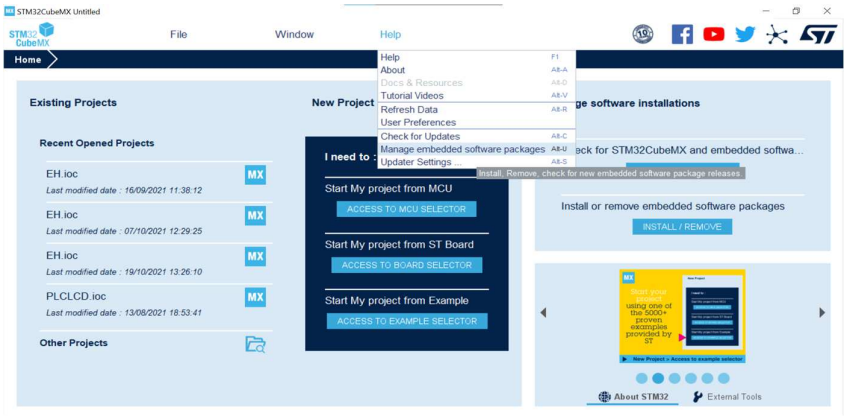

***Figure 1** Managing embedded software packs in STM32CubeMX*

2. From the Embedded Software Packages Manager window, press the ‘Refresh’ button to get an updated list of the add-on packs. Go to the ‘STMicroelectronics’ tab to find the X-CUBE-NFC7 pack.

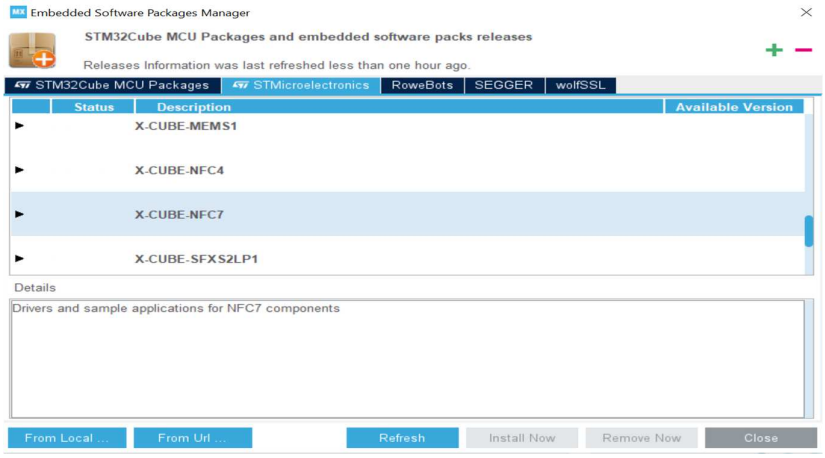

Figure 2 *Installing the X-CUBE-NFC7 pack in STM32CubeMX*

3. Select it checking the corresponding box and install it pressing the 'Install Now' button. Once the installation is completed, the corresponding box will become green, the 'Close' button can be pressed and the configuration of a new project can start.

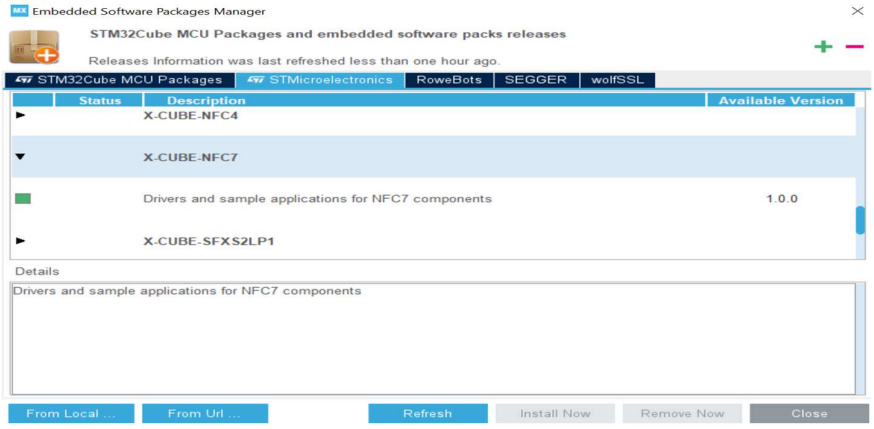

Figure 3 *The X-CUBE-NFC7 pack in STM32CubeMX*

6 Starting a new project

After launching the STM32CubeMX, you can choose if starting a New Project from the MCU Selector or from the Board Selector.

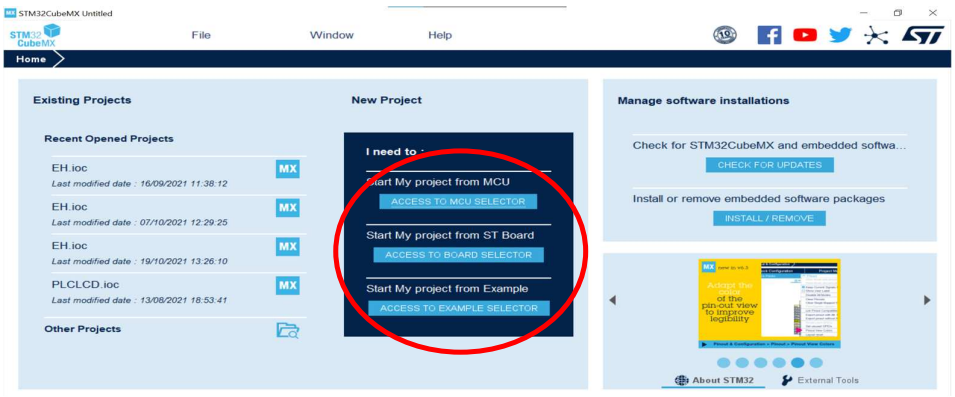

Figure 4 *STM32CubeMX main page*

The **MCU/Board selector** window will pop up. From this window, the STM32 MCU or platform can be selected.

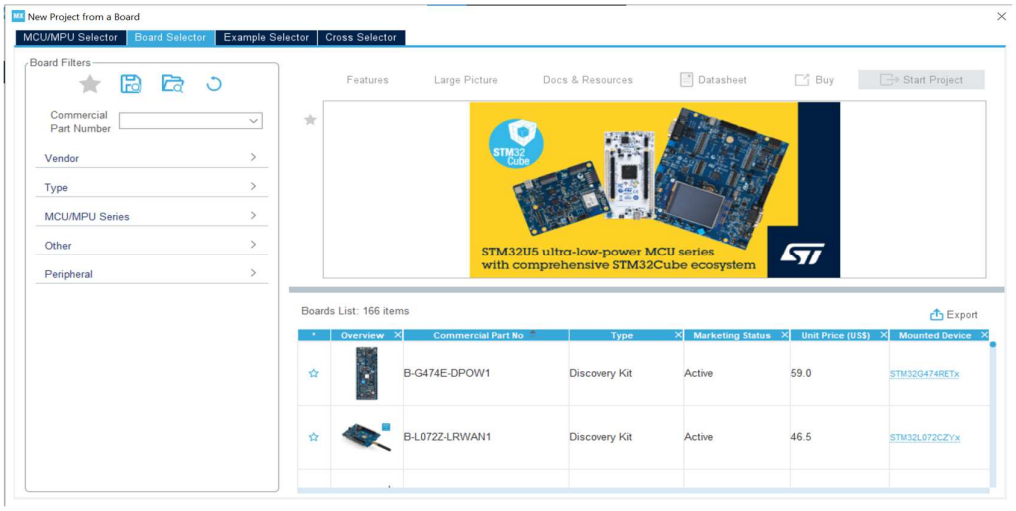

Figure 5 *STM32CubeMX* ***MCU/Board Selector*** *windows*

After selecting the MCU or the Board, the selected STM32 pinout will appear. From this window the user can set up the project, by adding one or more Additional Software and peripherals and configuring the clock.

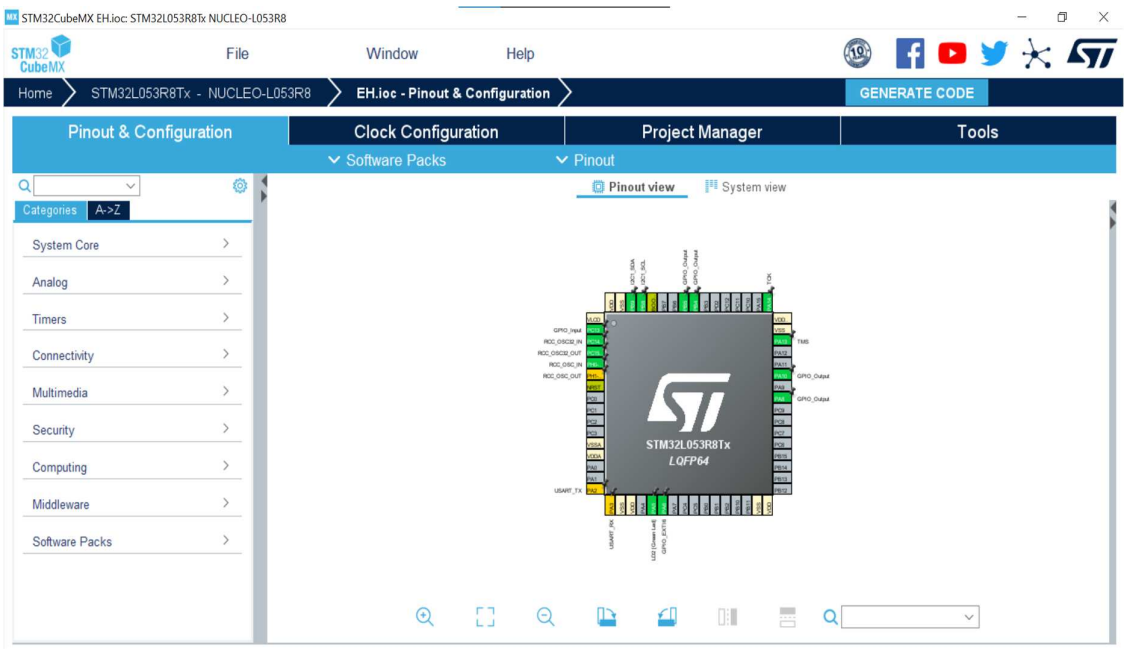

***Figure 6** STM32CubeMX **Pinout & Configuration** window*

To add the X-CUBE-NFC7 additional software to the project, the “Additional Softwares” button must be clicked.
From the Additional Software Component Selection window, the user can either choose to generate, for the selected MCU/Board, one of the enclosed sample applications or a new project. In this latter case, the user must just implement the main application logic without bothering with the pinout and peripherals configuration code that will be automatically generated by STM32CubeMX.

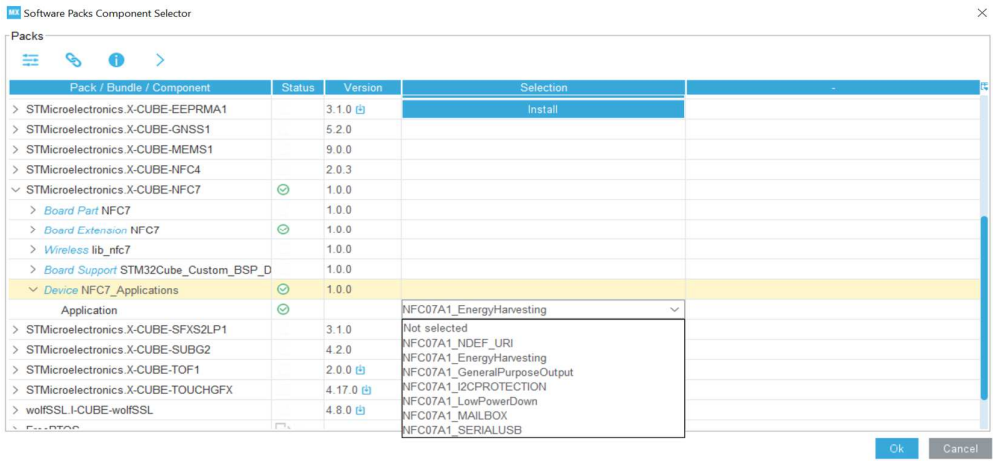

***Figure 7** STM32CubeMX **Additional Software Components selection** window*

7 STM32 Configuration Steps

The X-NUCLEO-NFC07A1 interfaces with the STM32 microcontroller via the I2C bus. Hence, assuming a user wants to interface the ST X-NUCLEO-NFC07A1 expansion board with a STM32 Nucleo 64 pins board (e.g. a Nucleo-F401RETx) no particular hardware modification must be done.

Figure 8 *STM32 Nucleo 64 and Nucleo 144 pins with X-NUCLEO-NFC07A1*

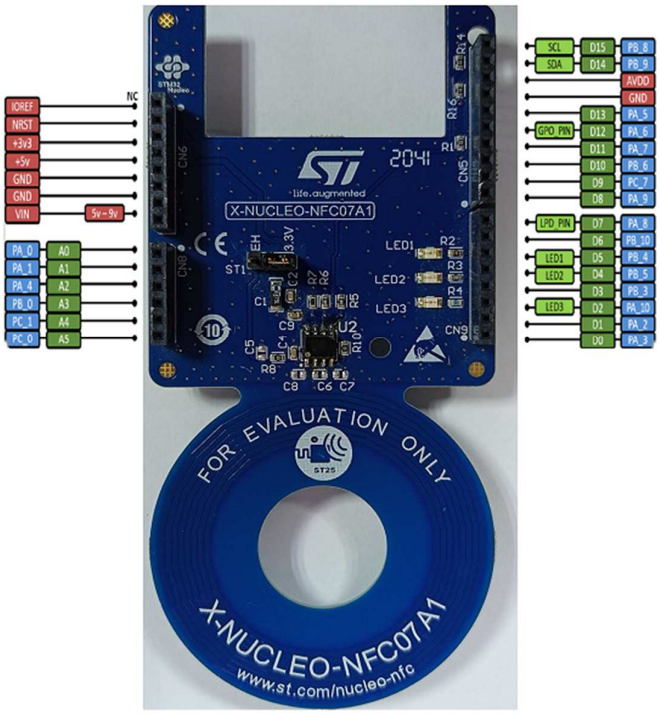

***Figure** 9 X-**NUCLEO-NFC07A1** pinout*

7.1 Use of NFC component without NDEF Library for X-NUCLEO-NFC07A1

This section outlines how to configure STM32CubeMX with X-NUCLEO-NFC07A1 when the use of the sample example is required without the dependencies over NDEF library. With such setup, only driver layers will be configured.

To add the X-CUBE-NFC7 additional software to the project, the “Additional Softwares” button must be clicked. From the “Additional Software Components selection” window, the user has to select the example from the “Device” class and “Board Extension” class as shown in the figure below.

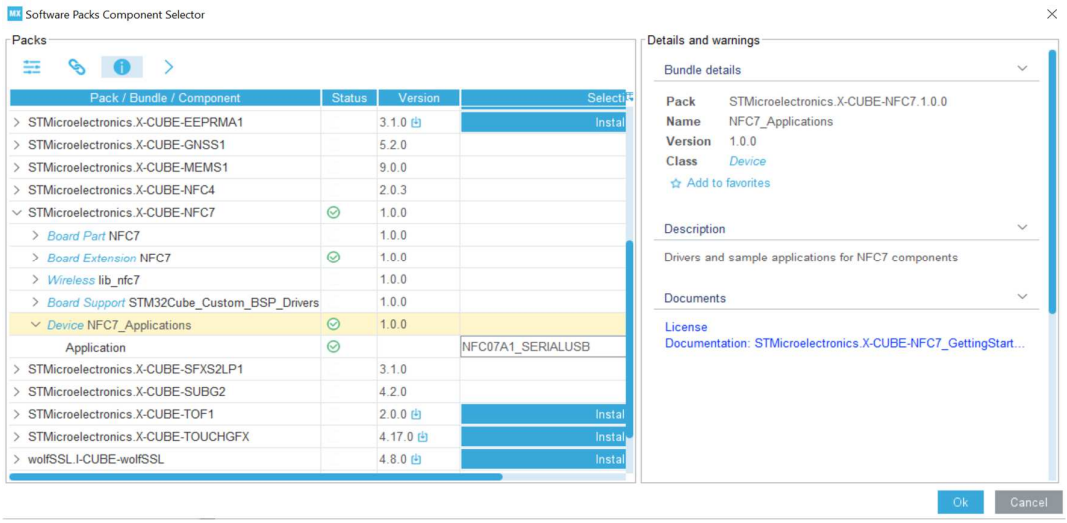

Figure 10 *STM32CubeMX Additional Software Components selection window*

From the **Pinout & Configuration** tab:

- from the **Pinout** scheme, click on PB8 and set it as I2C1_SCL;
- from the **Pinout** scheme, click on PB9 and set it as I2C1_SDA;
- enable the I2C1 as I2C from the “Connectivity” category;
- Configure the I2C1 settings with I2C speed at 400KHz (Fast Mode) from the “Configuration” view;

From the **Pinout** scheme set:

Nucleo 64		**Nucleo 144**	
PB8	I2C1_SCL	PB8	I2C1_SCL
PB9	I2C1_SDA	PB9	I2C1_SDA

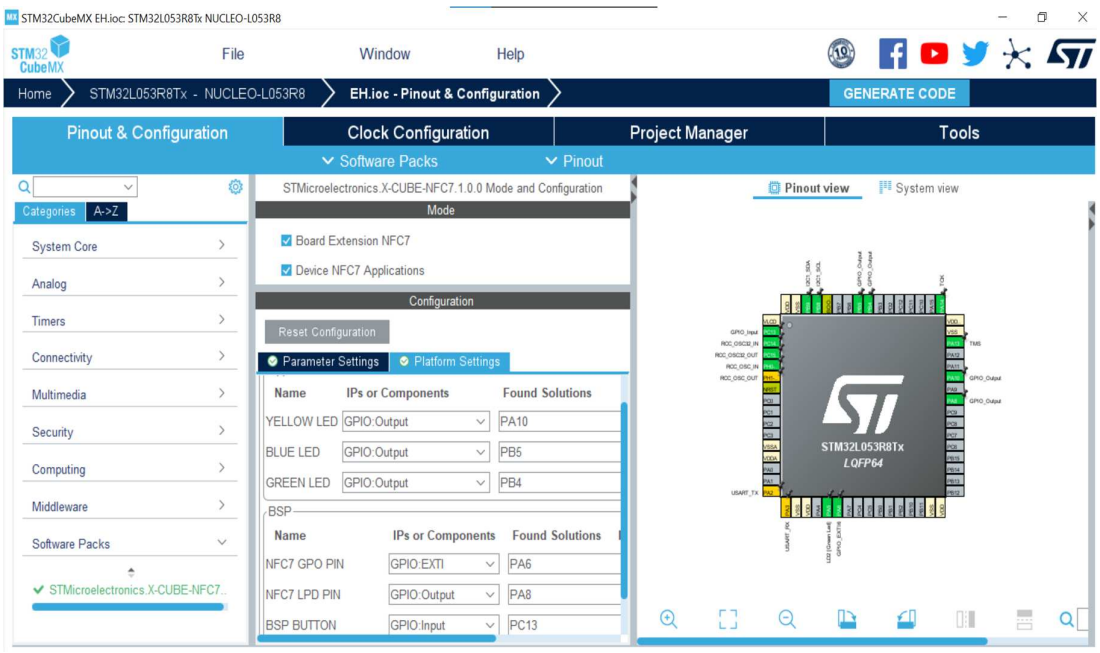

***Figure** 11 STM32CubeMX Pinout & Configuration tab and I2C settings*

From the **Additional Software** category, press the 'Stmicroelectronics.X-CUBE-NFC7.1.0.0' item, enable the "Board Extension NFC" checkbox from the "Mode" view and set the following Platform Settings from the "Configuration" view (take into account that according the example chosen some settings can appear or not):

-

Name	BSP_Api	Supported IPs	Nucleo 64	Nucleo 144
NFC07A1 BUS IO driver	BSP_BUS_DRIVER	I2C:I2C	I2C1	I2C1
NFC7 GPO PIN	HAL_EXTI_DRIVER	GPIO:EXTI	PA6	PA6
YELLOW LED		GPIO:Output	PA10	PF15
BLUE LED		GPIO:Output	PB5	PF14
GREEN LED		GPIO:Output	PB4	PE11
BSP USART	BSP_COMMON_DRIVER	USART:Asynchronous	USART2	USART3
BSP BUTTON	BSP_COMMON_DRIVER	GPIO:EXTI	PC13	PC13

Once all the above described steps have been performed, the source code of the project using the **STMicroelectronics X-CUBE-NFC7** software can be generated clicking the “GENERATE CODE” button.

7.2 Use of NDEF Library with sample applications for X-NUCLEO-NFC07A1

This section outlines how to configure STM32CubeMX with X-NUCLEO-NFC07A1 when the use of the sample applications is required. With such setup, all the components of the expansion software package, including applications, will be properly configured.
To add the X-CUBE-NFC7 additional software to the project, the “Additional Softwares” button must be clicked. From the “Additional Software Components selection” window, the user has to select the “Board Extension” class, NFC07A1_NDEF_URI application from the “Device” class and select the “Wireless” class with “Basic” Interface as shown in the figure below.

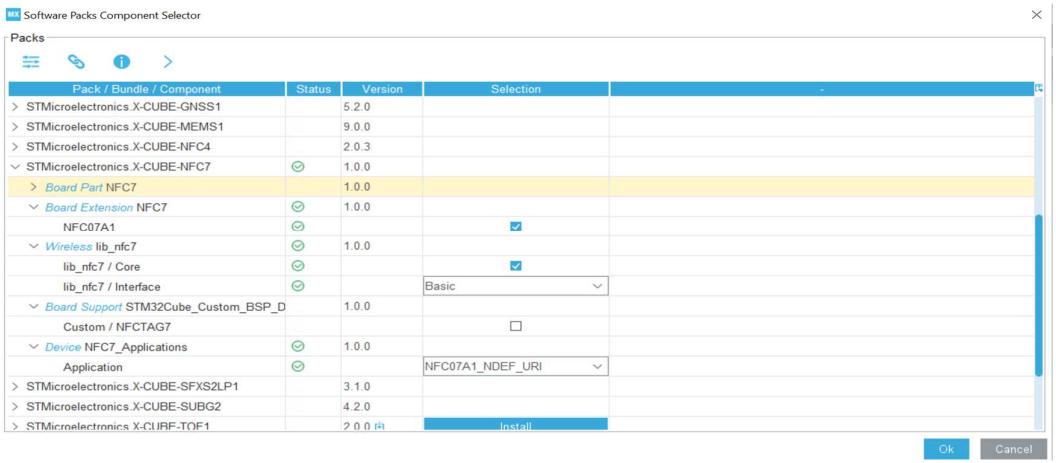

***Figure 12** STM32CubeMX **Additional Software Components selection** window*

From the **Pinout & Configuration** tab:

- from the **Pinout** scheme, click on PB8 and set it as I2C1_SCL;
- from the **Pinout** scheme, click on PB9 and set it as I2C1_SDA;
- enable the I2C1 as I2C from the “Connectivity” category;

From the **Pinout** scheme, if not already set, set:

Nucleo 64			**Nucleo 144**		
PIN	*Mode*	*Label*	*PIN*	*Mode*	*Label*
PB5	GPIO_Output		PF14	GPIO_Output	
PB4	GPIO_Output		PE11	GPIO_Output	
PA5	GPIO_Output	LD2 [Green Led]	PB7	GPIO_Output	LD2[Blue]
PC13	GPIO_EXTI13	B1 [Blue PushButton]	PC13	GPIO_EXTI13	USER_Btn[B1]

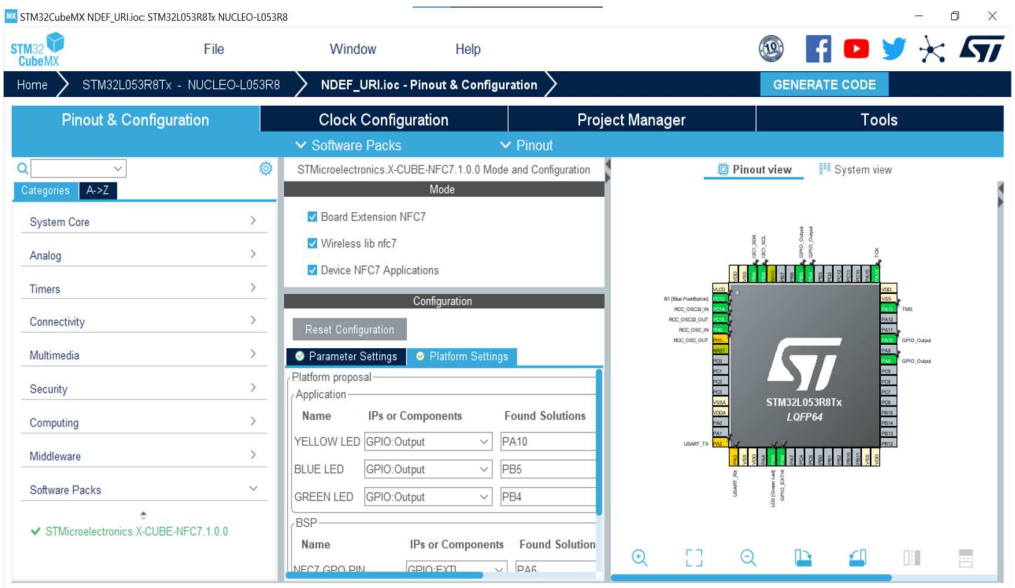

***Figure 13** STM32CubeMX **Pinout & Configuration** tab*

From the **Additional Software** category, press the 'Stmicroelectronics.X-CUBE-NFC7.1.5.0' item, enable the "Board Extension NFC" and the "Device Application" checkboxes from the "Mode" view and set the following Platform Settings from the "Configuration" view (take into account that according the example chosen some settings can appear or not):

Name	BSP_Api	Supported IPs	Nucleo 64	Nucleo 144
NFC07A1 BUS IO driver	BSP_BUS_DRIVER	I2C:I2C	I2C1	I2C1
NFC7 GPO PIN	HAL_EXTI_DRIVER	GPIO:EXTI	PA6	PA6
YELLOW LED		GPIO:Output	PA10	PF15
BLUE LED		GPIO:Output	PB5	PF14
GREEN LED		GPIO:Output	PB4	PE11

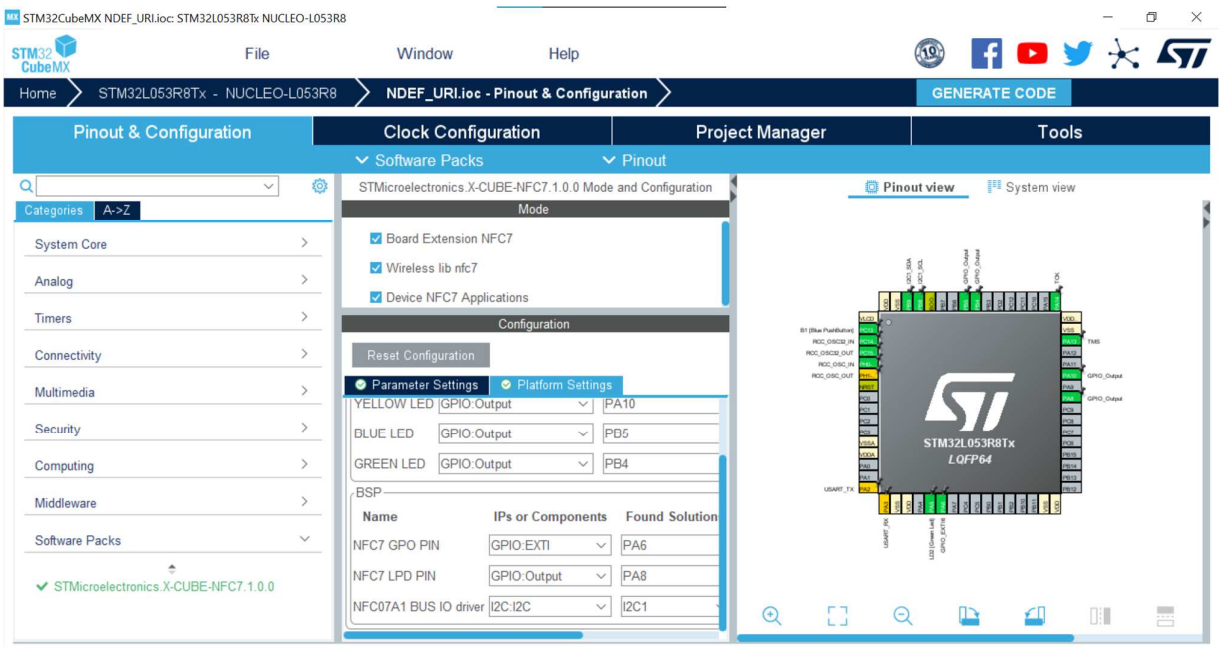

***Figure 14** STM32CubeMX **Pinout & Configuration** tab and Additional Software settings for NFC07A1_NDEF_URI applications*

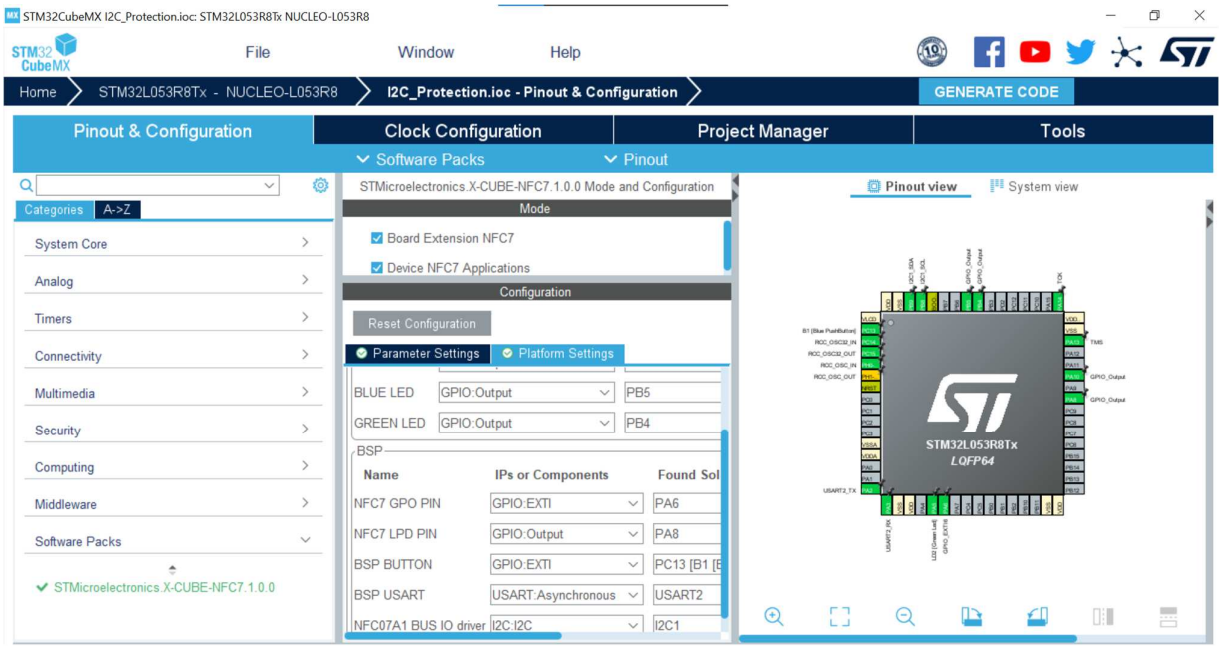

***Figure 15** STM32CubeMX **Pinout & Configuration** tab and Additional Software settings for NFC07A1_I2CProtection example*

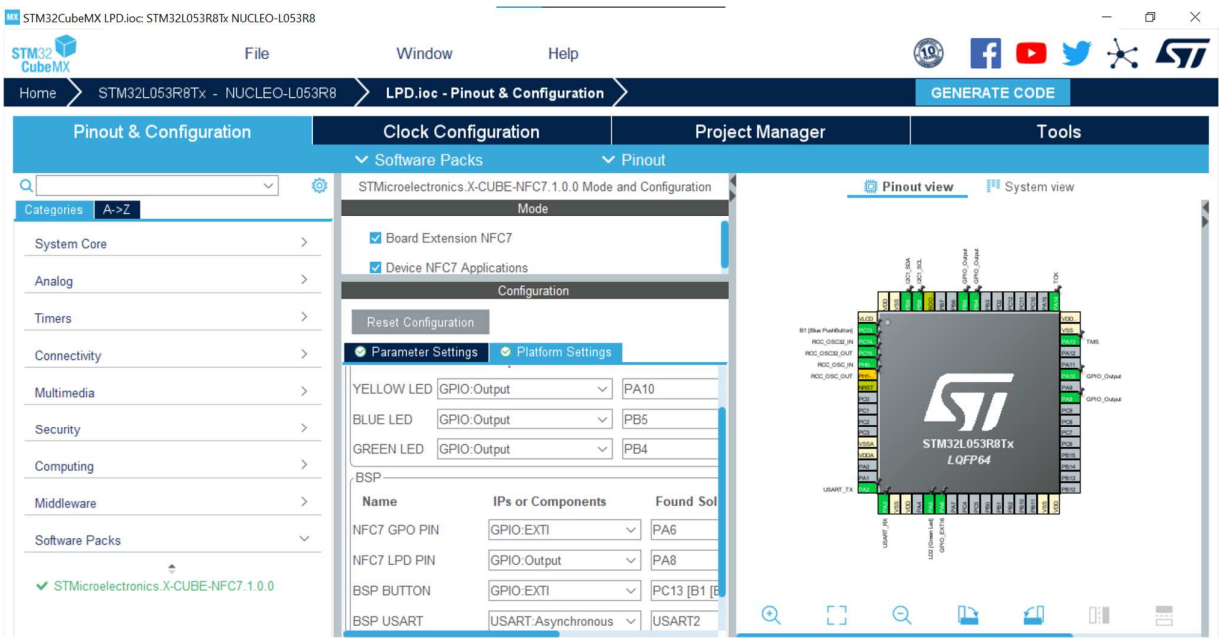

***Figure 16** STM32CubeMX **Pinout & Configuration** tab and Additional Software settings for NFC07A1_LowPowerDown example*

***Figure 17** STM32CubeMX **Pinout & Configuration** tab and Additional Software settings for NFC07A1_Mailbox example*

For all the sample applications, the default parameters can be used.

From the **Configuration & Pinout** tab, click on “System Core” category and then on NVIC item

to enable the EXTI line interrupts:

Nucleo 64	Nucleo 144
EXTI line 6 interrupt	EXTI line 6 interrupt
EXTI line 13 interrupt	EXTI line 13 interrupt

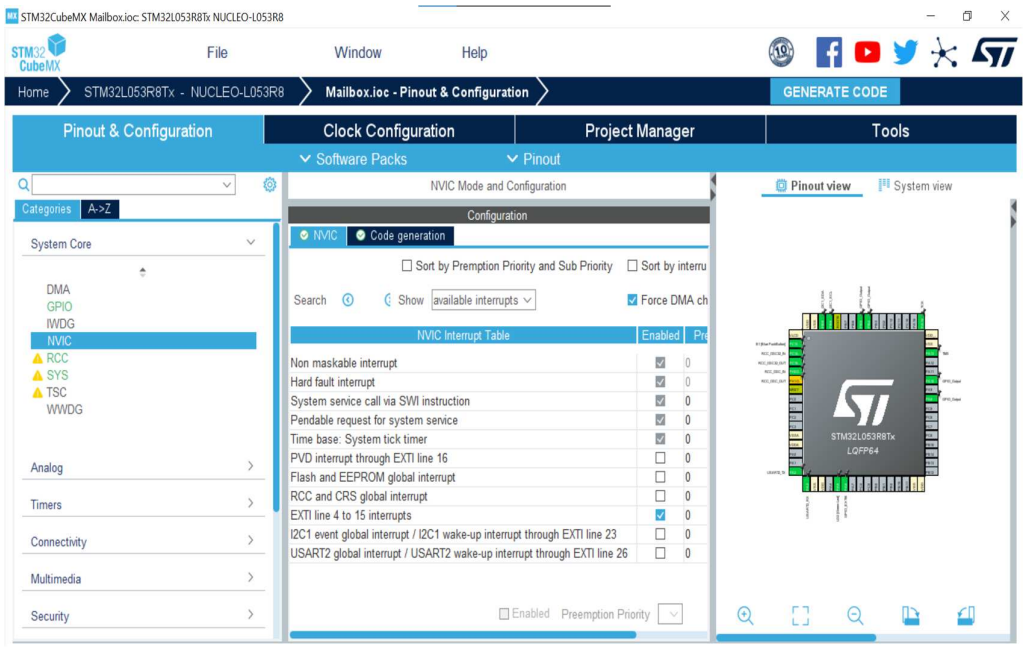

***Figure 18** STM32CubeMX NVIC Configuration*

From the **Configuration & Pinout** tab, click on “Connectivity” category and then on I2C1 item to set the I2C speed at 400KHz or 1MHz (for STM32L0/STM32L4 families). If STM32L0/STM32L4 MCU families are used, kindly set the Coefficient of Digital Filter to 2 in Parameter settings:

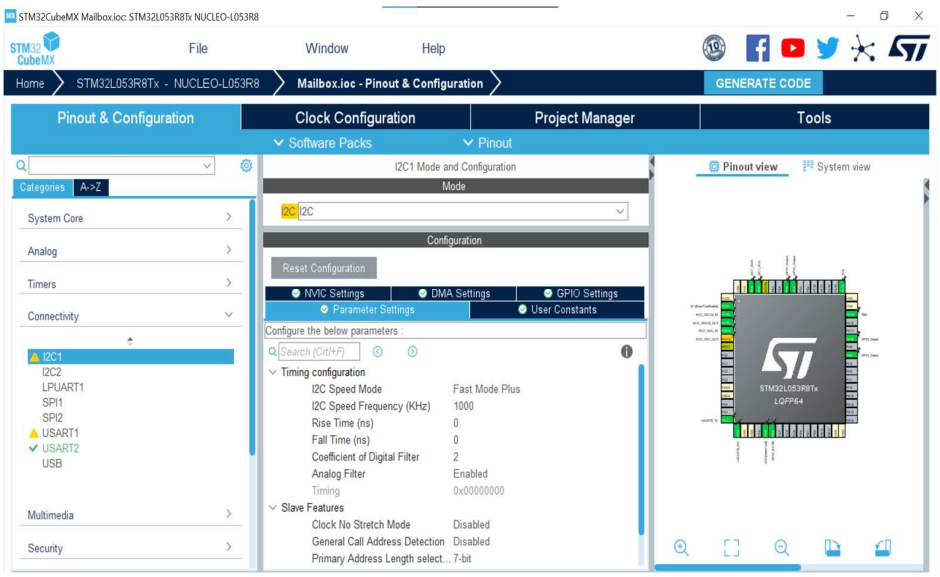

Figure 19 *STM32CubeMX I2C Configuration*

From the **Configuration & Pinout** tab, click on “Connectivity” category and then on USART2 item and check that the following configuration is set:

Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1

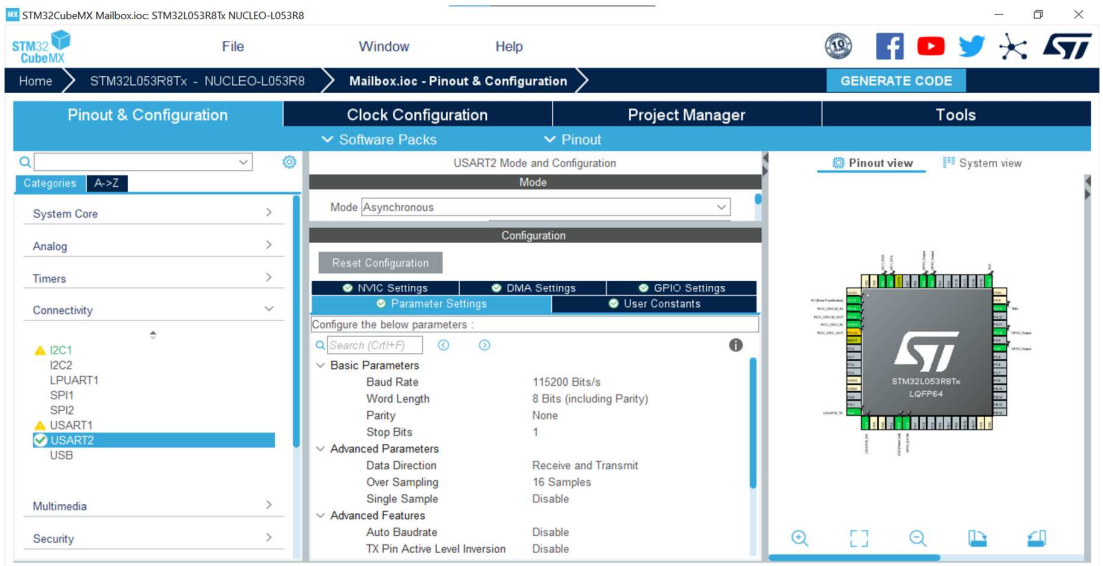

Figure 20 *STM32CubeMX USART Configuration*

Once all the above described steps have been performed, the sample applications for NFC07A1 using the **STMicroelectronics X-CUBE-NFC7** software can be generated clicking the “GENERATE CODE” button.

7.3 Use of ST25DVXXKC component without sample applications for custom boards

This section outlines how to configure STM32CubeMX with a custom board that mounts ST25DVXXKC device. With such setup, only drivers layer will be configured. This setup is useful when user does not intend to leverage the sample application provided in the package, to develop his own application code. In this case you can configure the ST25DVXXKC device in order to be used. To add the X-CUBE-NFC7 additional software to the project, the “Additional Softwares” button must be clicked. From the “Additional Software Components selection” window, the user has to select the “Board Components” class and the “Board Support” class in “STM32Cube_Custom_BSP_Drivers” bundle as shown in figure below.

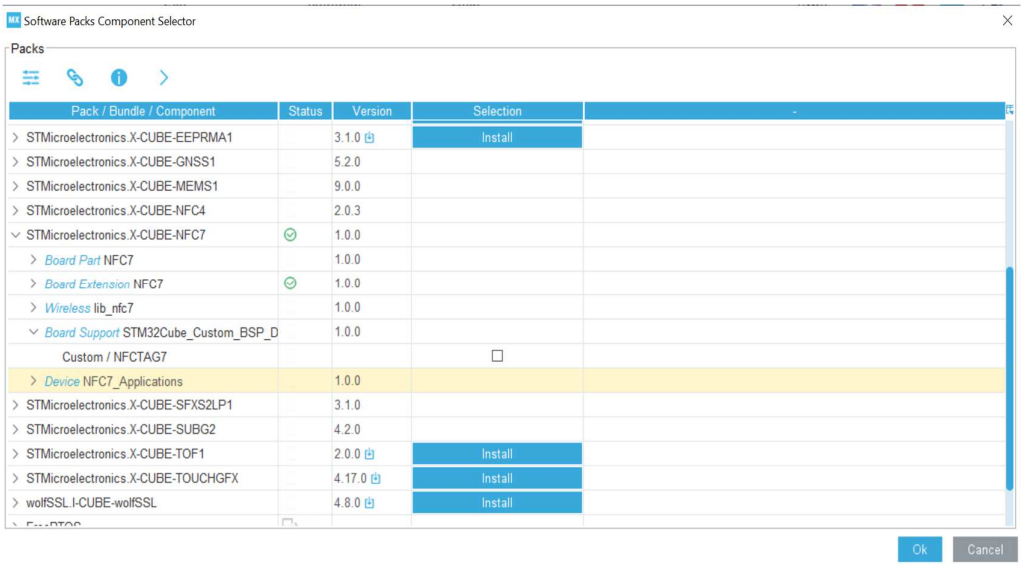

***Figure 21** STM32CubeMX **Additional Software Components selection** window*

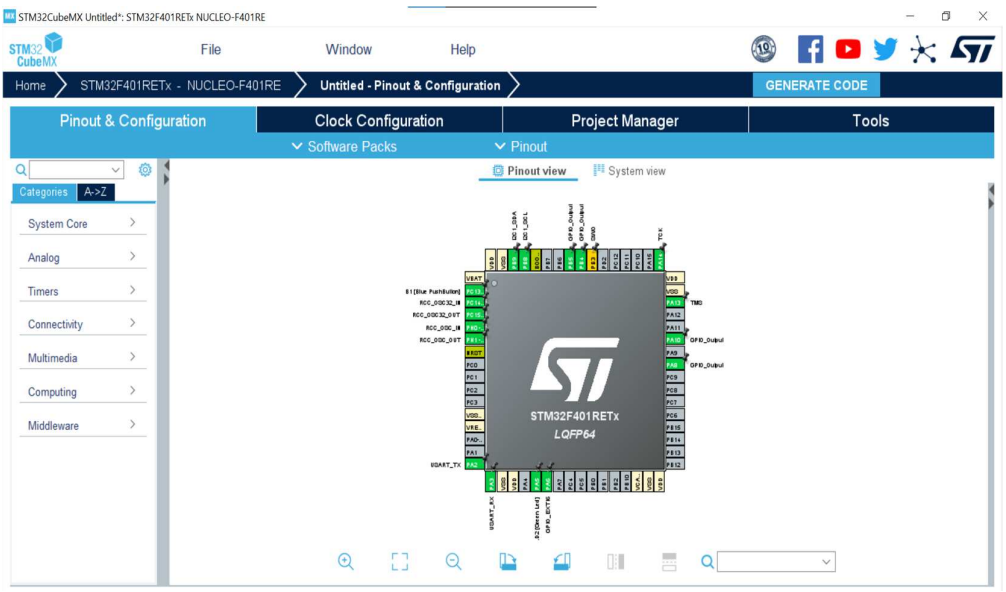

***Figure 22** STM32CubeMX **Pinout & Configuration** tab*

From the **Pinout & Configuration** tab:

- from the **Pinout** scheme, click on PB8 and set it as I2C1_SCL;
- from the **Pinout** scheme, click on PB9 and set it as I2C1_SDA;
- enable the I2C1 as I2C from the “Connectivity” category;

From the **Pinout** scheme, if not already set, set:

Nucleo 64			**Nucleo 144**		
PIN	*Mode*	*Label*	*PIN*	*Mode*	*Label*
PB5	GPIO_Output		PF14	GPIO_Output	
PB4	GPIO_Output		PE11	GPIO_Output	
PA5	GPIO_Output	LD2 [Green Led]	PB7	GPIO_Output	LD2[Blue]
PC13	GPIO_EXTI13	B1 [Blue PushButton]	PC13	GPIO_EXTI13	USER_Btn[B1]

. You can see an example of configuration below.

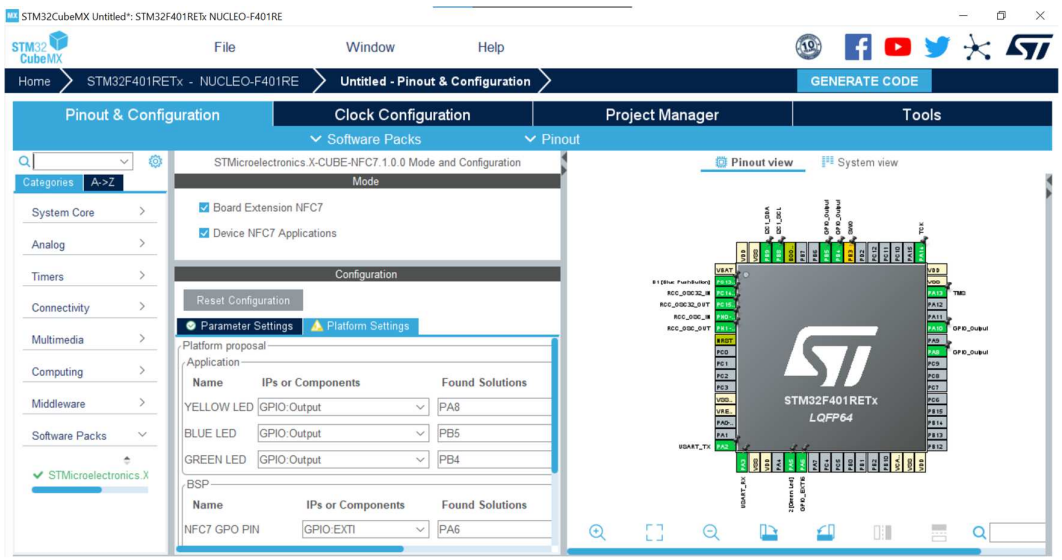

***Figure 23** STM32CubeMX **Parameter Settings** for a custom board*

Set the following Platform Settings from the “Configuration” view:

Name	BSP_Api	Supported IPs	Nucleo 64	Nucleo 144
ST25DVXXKC BUS IO driver	BSP_BUS_DRIVER	I2C:I2C	I2C1	I2C1
ST25DVXXKC GPO PIN	HAL_EXTI_DRIVER	GPIO:EXTI	PA6	PA6
ST25DVXXKC LPD PIN		GPIO:Output	PA8	PF13

.

From the **Configuration & Pinout** tab, click on “System Core” category and then on NVIC item to enable the EXTI line interrupts.

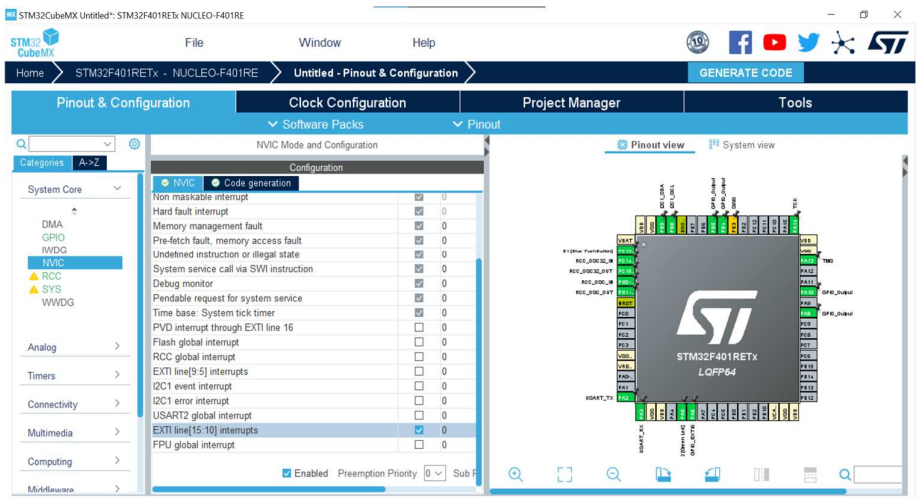

Figure 24 *STM32CubeMX NVIC Configuration*

From the **Configuration & Pinout** tab, click on “Connectivity” category and then on I2C1 item to set the I2C speed at 400KHz (Fast Mode):

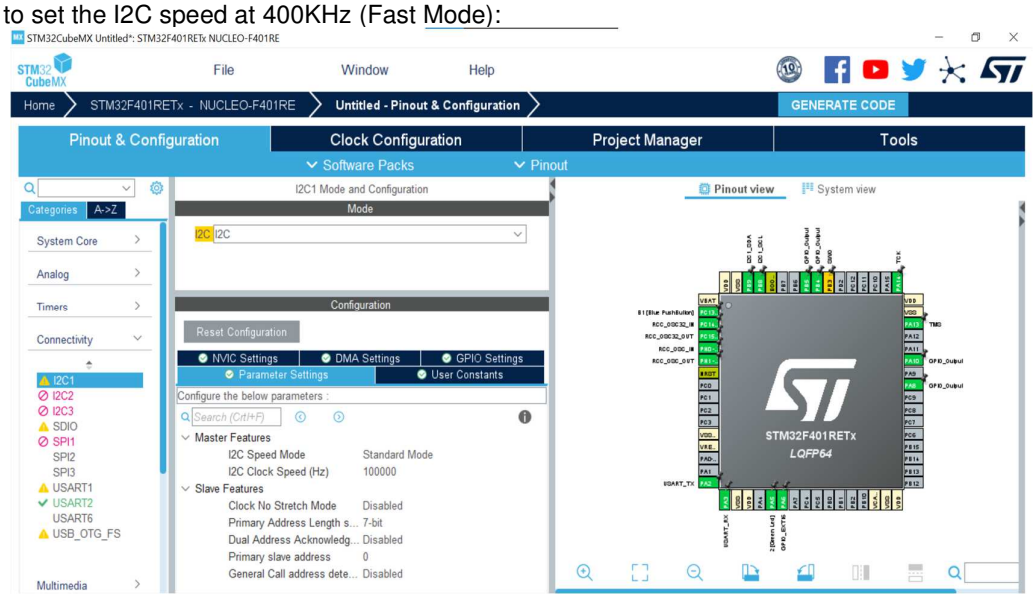

Figure 25 *STM32CubeMX I2C Configuration*

From the **Configuration & Pinout** tab, click on “Connectivity” category and then on USART2 button and check the following configuration is set:

Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1

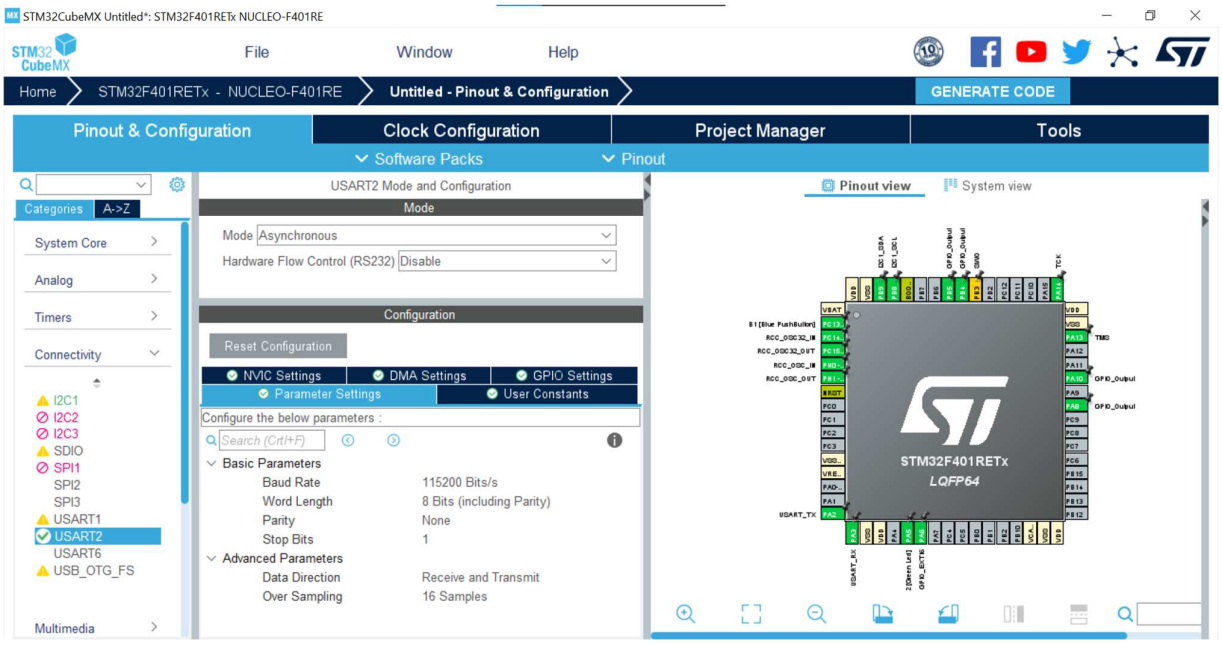

Figure 26 *STM32CubeMX USART Configuration*

Once all the above described steps have been performed, the sample applications for a custom board using the **STMicroelectronics X-CUBE-NFC7** software can be generated clicking the "GENERATE CODE" button.

8 Generated Folders Structure

When generating a project, two models of folders structure can be adopted when using a high level firmware component (i.e. a middleware in the STM32Cube MCU package):

- **Basic Structure**: the basic structure is often used with HAL examples and single package projects. This structure consists of having the IDE configuration folder in the same level as the sources (organized in *Inc* and *Src* subfolders).
- **Advanced Structure**: the advanced structure provides a more efficient and organized folders model that allows ease middleware applications integration when several packages are used.
  In the Advanced mode *Src* and *Inc* are generated under folder *Core*.
  For each package, the list of the generated files is under <*Package_Name*> (*X-CUBE-NFC7* for the X-CUBE-NFC7 pack), at the same level as *Core* and containing inside the *App* and the *Target* subfolder.

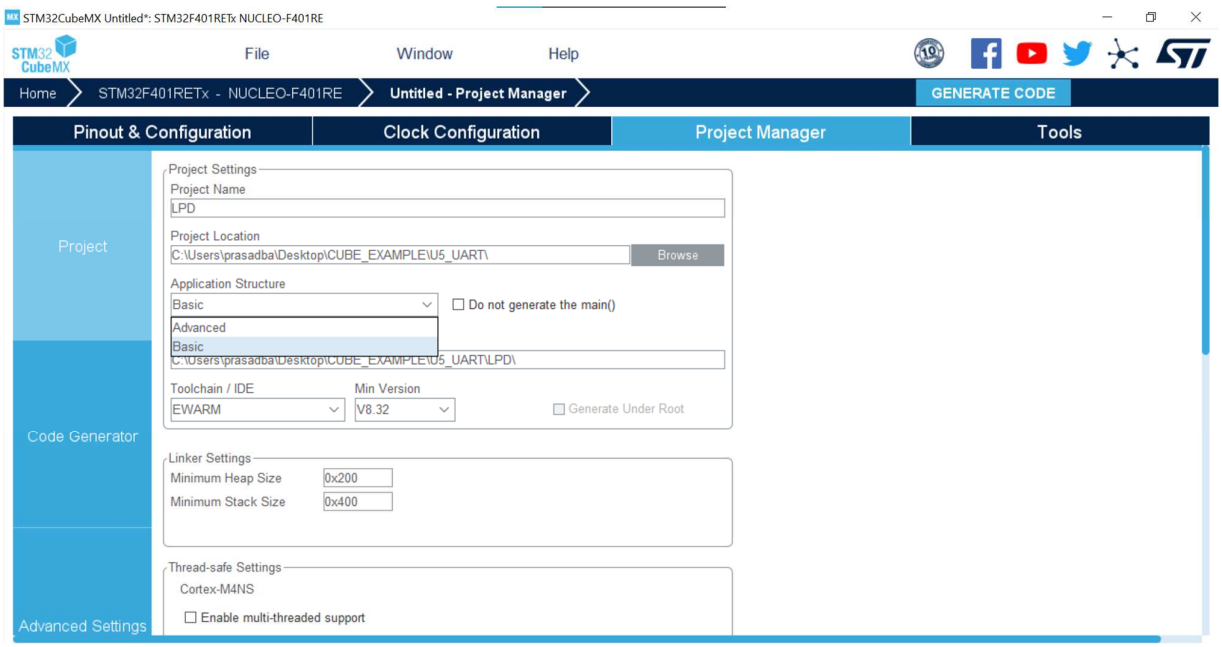

***Figure 27** STM32CubeMX **Application Structure** Configuration*

9 Known Limitations and workarounds

STM32CubeMX NFC7 pack v1.0.0 is fully compatible with STM32CubeMX v6.4.0 and upwards. It is not fully compatible with previous version of STM32CubeMX (<=v6.4.0).

9 References

[1]UMxxxx– User Manual - *Getting started with the X-CUBE-NFC7 dynamic NFC/RFID tag IC software expansion for STM32Cube*

10 Revision history

Table 2: Document revision history

Date	Version	Changes
25-Oct-2021	1	Initial release

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.