Getting started with the 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. Step-By-Step Section describes the steps required to configure a generic project using the X-NUCLEO-NFC07A1 expansion board with any 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.

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_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.

4.3 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

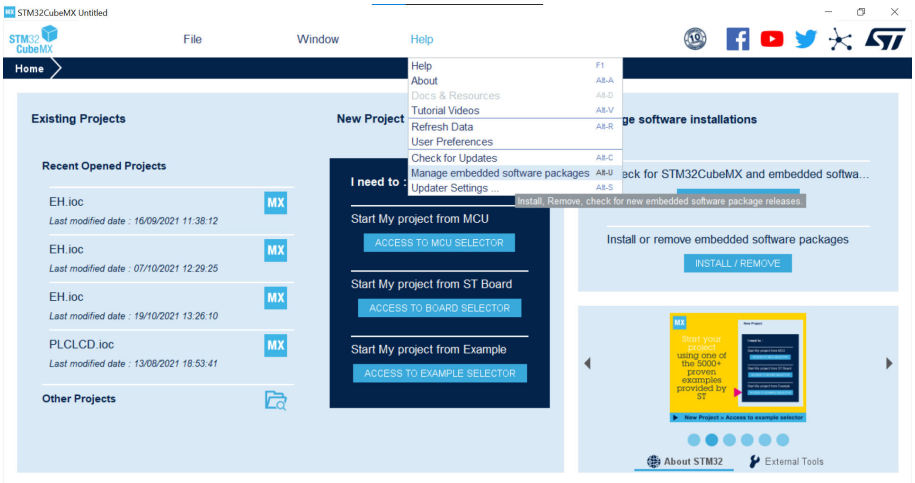
status register of ST25DVXXKC device. Text is displayed on a UART console (via ST-LINK) if a PC is connected. The example serial settings can be configured by user changing the settings of USART2 in the STM32CubeMX GUI.

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.

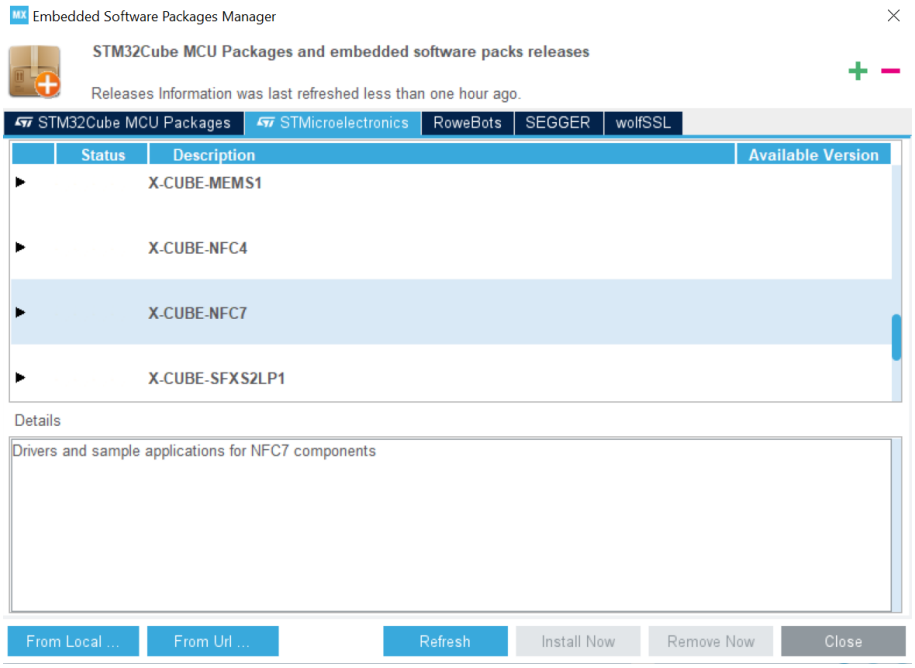
Step 1. From the menu, select Help > Manage embedded software packages

Figure 1. Managing embedded software packs in STM32CubeMX



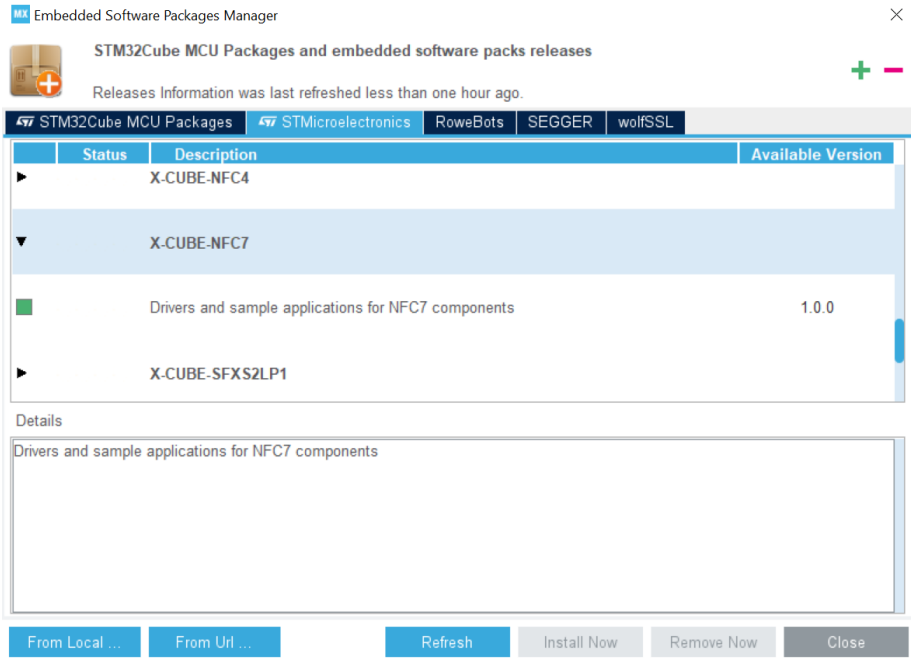
Step 2. From the Embedded Software Packages Manager window, press the ‘Refresh’ button to get an updated list of the add-on packs. Go to the ‘STMicroelectronics’ tab to find the X-CUBE-NFC7 pack.

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



Step 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.

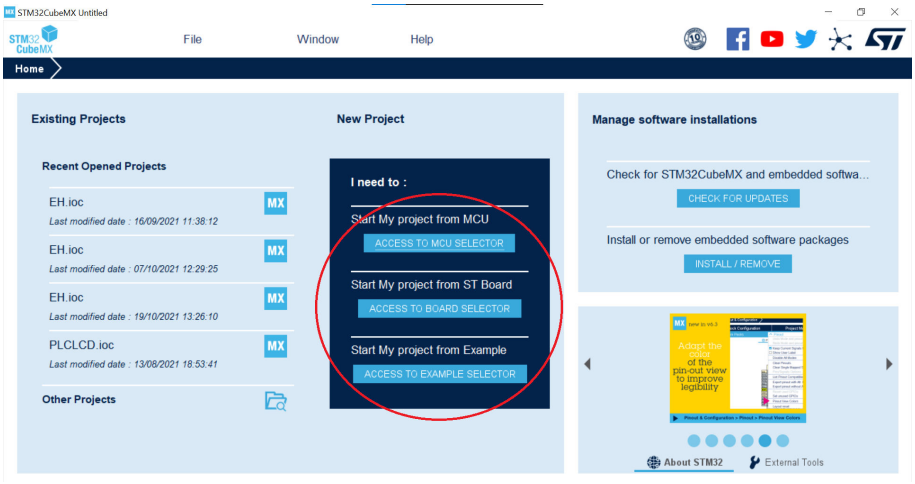
Figure 3. The X-CUBE-NFC7 pack in STM32CubeMX



6 Starting a new project

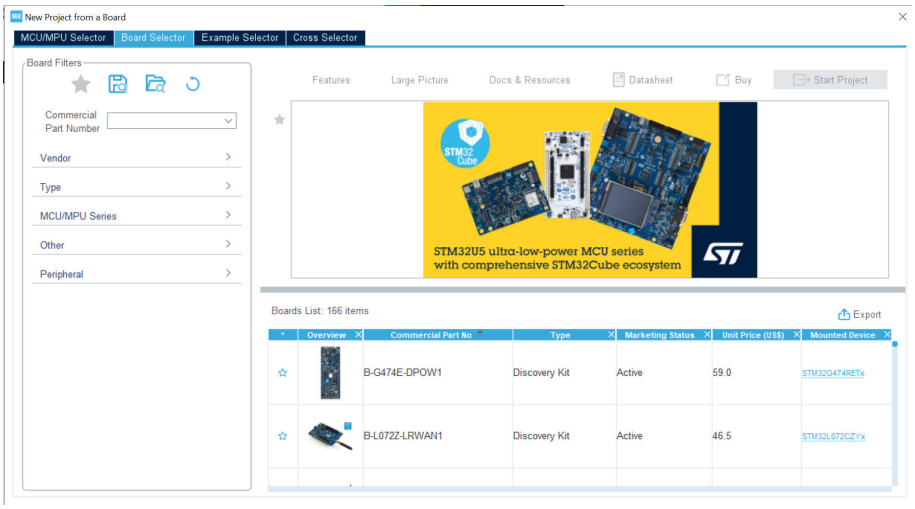
After launching the STM32CubeMX, you can choose if starting a New Project from the MCU Selector or from the Board Selector.

Figure 4. STM32CubeMX main page



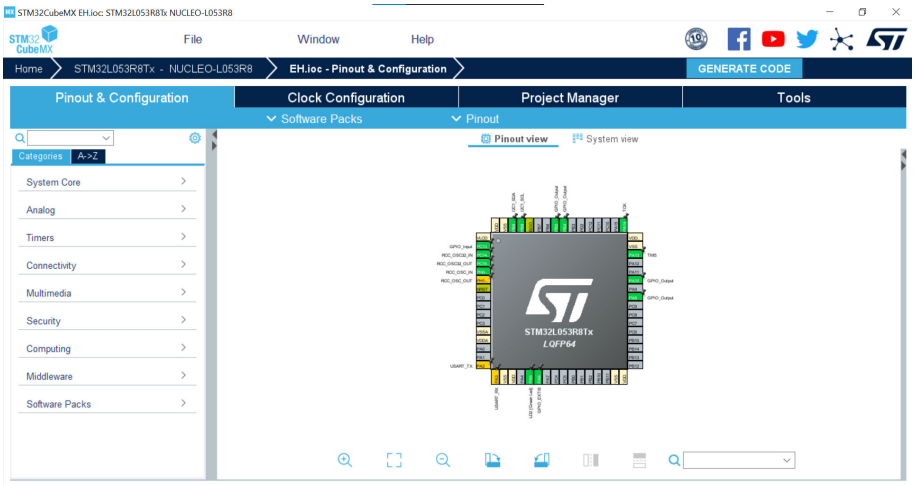
The **MCU/Board selector** window will pop up. From this window, the STM32 MCU or platform can be selected.

Figure 5. STM32CubeMX MCU/Board Selector window



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 pack and peripherals and configuring the clock.

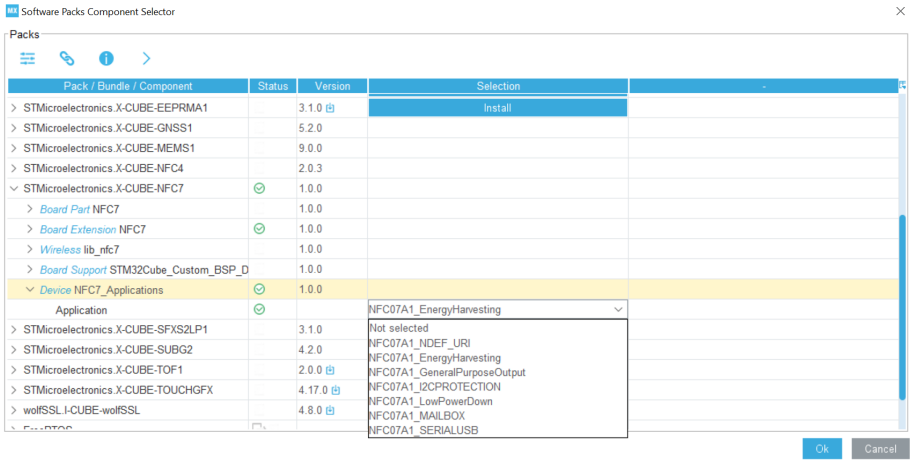
Figure 6. STM32CubeMX Pinout & Configuration window



To add the X-CUBE-NFC7 additional pack to the project, the “Additional packs” button must be clicked.

From the Additional pack 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 Corresponding pack 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

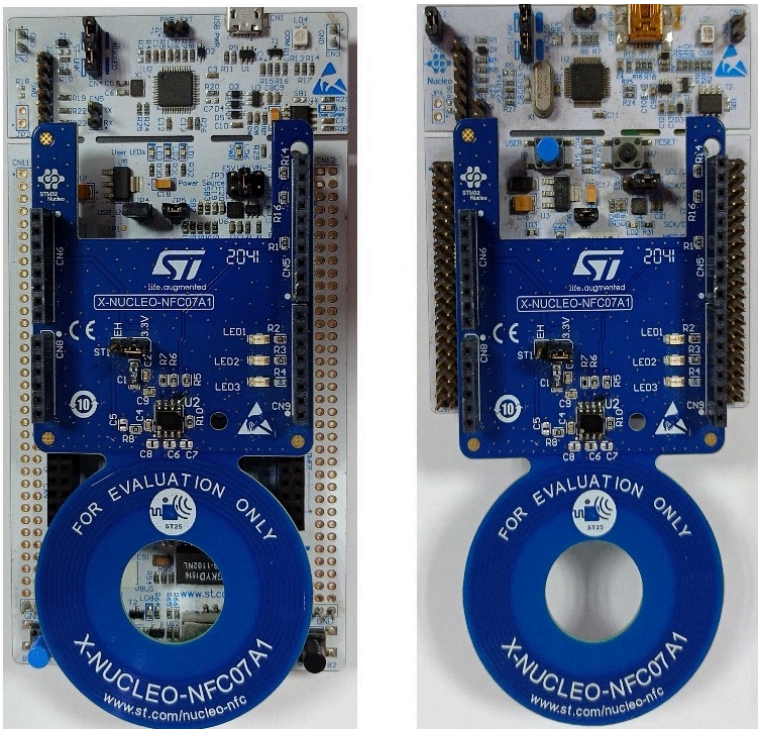
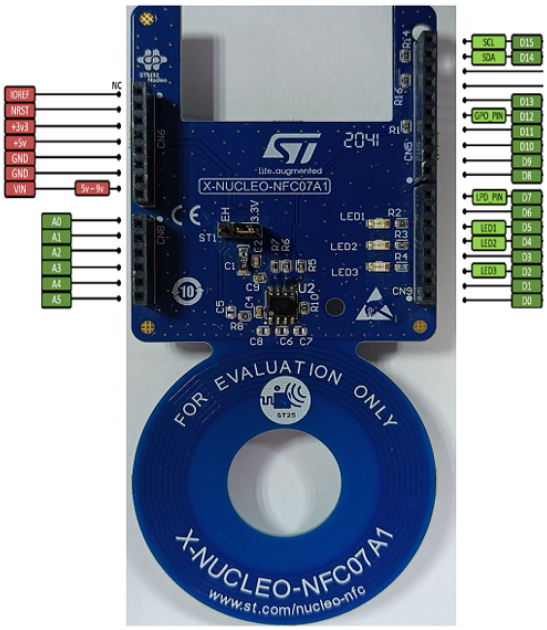


Figure 9. X-NUCLEO-NFC07A1 pinout

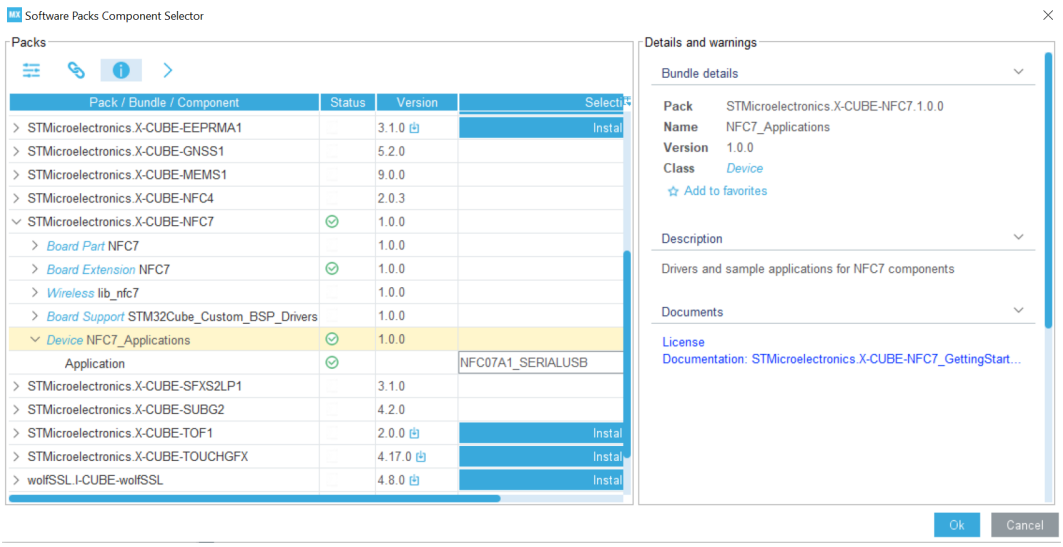


7.1 Use of NFC component without NDEF Library for X-NUCLEO-NFC07A1

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 pack to the project, the “Corresponding pack button” button must be clicked. From the “Corresponding pack 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 Corresponding pack Components selection window



From the **Pinout & Configuration** tab:

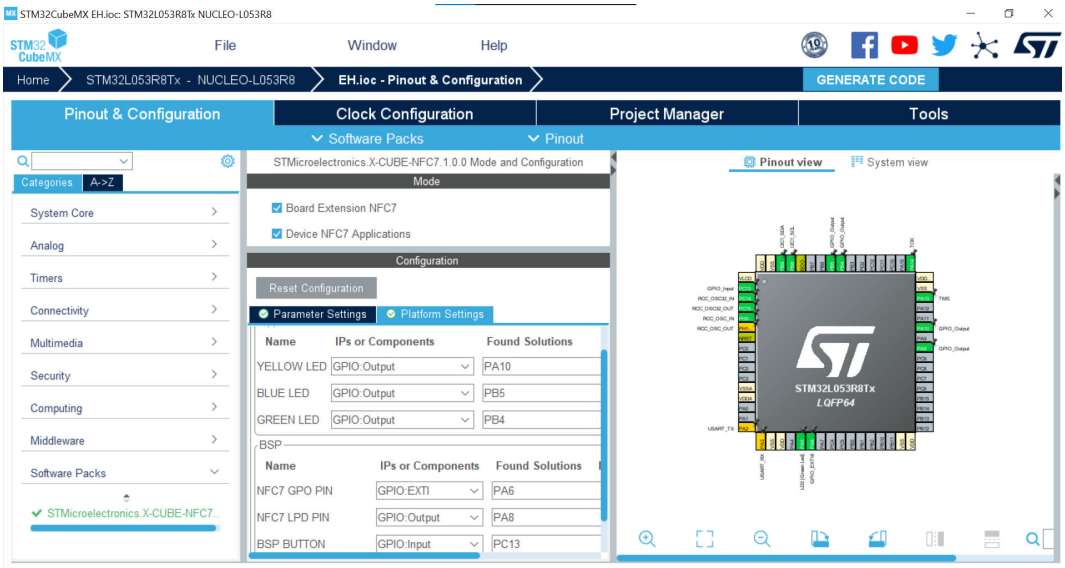
- from the **Pinout** scheme, click on PB8 and set it as I2C1_SCL;
- from the **Pinout** scheme, click on PB9 and set it as I2C1_SDA;
- enable the I2C1 as I2C from the “Connectivity” category;
- Configure the I2C1 settings with I2C speed at 400KHz (Fast Mode) from the “Configuration” view;

From the **Pinout** scheme set:

Table 2. Pinout scheme

Nucleo 64		Nucleo 144		Nucleo-WL55JC		Nucleo-WB55RG	
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
PB8	I2C1_SCL	PB8	I2C1_SCL	PA12	I2C2_SCL	PB8	I2C1_SCL
PB9	I2C1_SDA	PB9	I2C1_SDA	PA11	I2C2_SDA	PB9	I2C1_SDA

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



From the **Additional pack** 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):

Table 3. **IPs/Component selection**

Name	BSP_Api	Supported IPs	Nucleo 64	Nucleo 144
NFC07A1 BUS IO driver	BSP_BUS_DRIVER	I2C:I2C	I2C1(1)	I2C1
NFC7 GPO PIN	HAL_EXTI_DRIVER	GPIO:EXTI	PA6	PA6
YELLOW LED	HAL_GPIO_DRIVER	GPIO:Output	PA10(2)	PF15
BLUE LED	HAL_GPIO_DRIVER	GPIO:Output	PB5(3)	PF14
GREEN LED	HAL_GPIO_DRIVER	GPIO:Output	PB4(4)	PE11
BSP USART	BSP_COMMON_DRIVER	USART:Asynchronous	USART2	USART3
BSP BUTTON	BSP_COMMON_DRIVER	GPIO:EXTI	PC13(5)	PC13

1. *I2C2 in case of NUCLEO-WL55JC1.*
2. *PB12 for NUCLEO-WL55JC1 and PC6 for NUCLEO-WB55RG.*
3. *PC10 for NUCLEO-WB55RG.*
4. *PB8 for NUCLEO-WL55JC1 and PA15 for NUCLEO-WB55RG.*
5. *PA6 in case of NUCLEO-WB55RG*

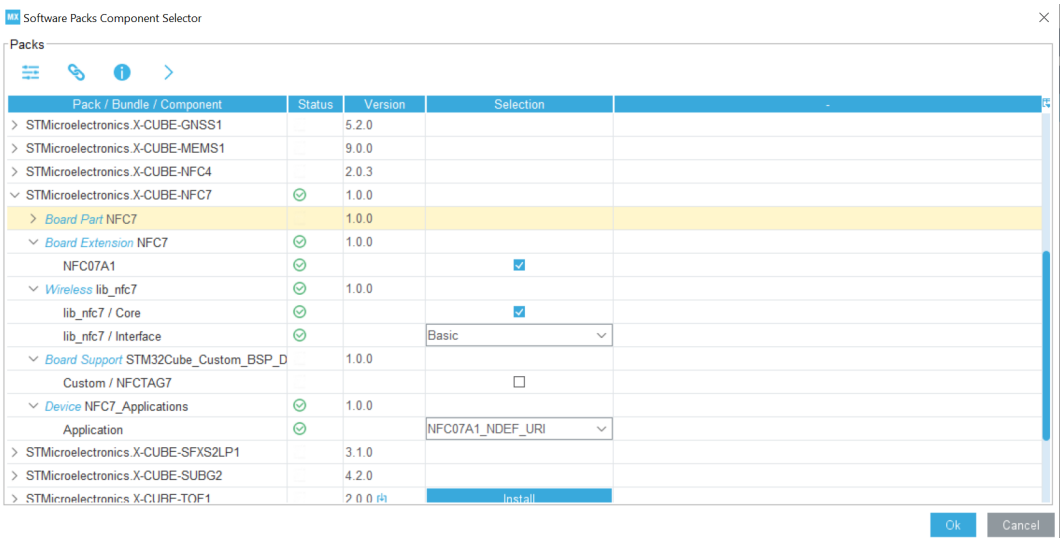
Once all the above-described steps have been performed, the source code of the project using the **STMicroelectronics X-CUBE-NFC7** software can be generated clicking the “GENERATE CODE” button.

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 pack to the project, the “Corresponding pack button” button must be clicked. From the “Corresponding pack 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 Corresponding pack 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.

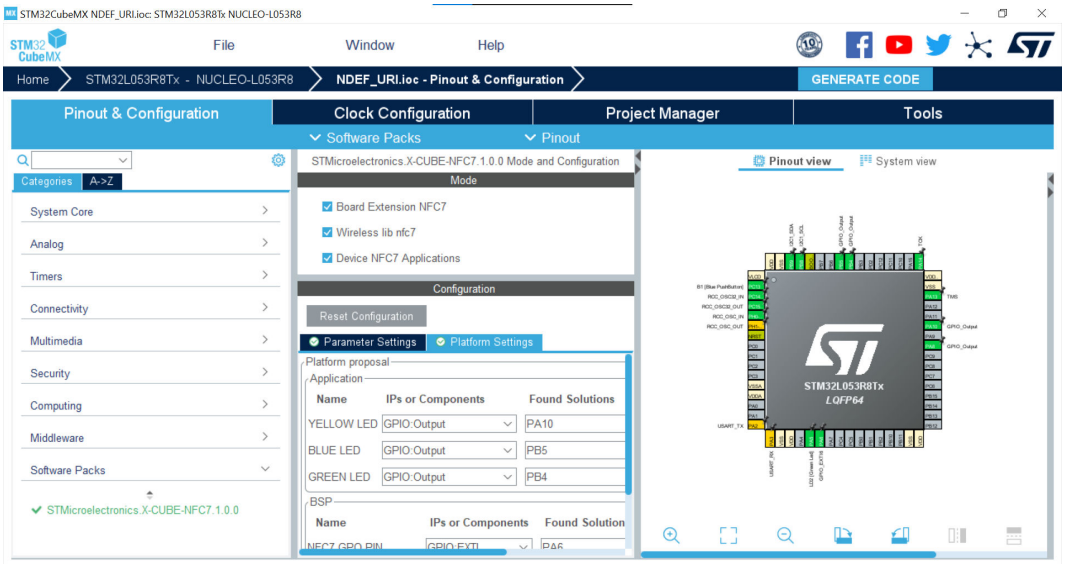
From the **Pinout** scheme, if not already set, set:

Table 4. Pin selection for NUCLEO Boards

Nucleo 64			Nucleo 144		
PIN	*Mode*	*Label*	*PIN*	*Mode*	*Label*
PB5(1)	GPIO_Output		PF14	GPIO_Output	
PB4(2)	GPIO_Output		PE11	GPIO_Output	
PA5(3)	GPIO_Output	LD2 [Green Led]	PB7	GPIO_Output	LD2[Blue]
PC13(4)	GPIO_EXTI13	B1 [Blue PushButton]	PC13	GPIO_EXTI13	USER_Btn[B1]

1. *PC10 for NUCLEO-WB55RG.*
2. *PB8 for NUCLEO-WL55JC1 and PA15 for NUCLEO-WB55RG.*
3. *PB9 for NUCLEO-WL55JC1 and PB0 for NUCLEO-WB55RG.*
4. *PA6 for NUCLEO-WB55RG.*

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



From the **Additional pack** 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):

Table 5. **I2C and Pin selection for NUCLEO Boards**

Name	BSP_Api	Supported IPs	Nucleo 64	Nucleo 144
NFC07A1 BUS IO driver	BSP_BUS_DRIVER	I2C:I2C	I2C1(1)	I2C1
NFC7 GPO PIN	HAL_EXTI_DRIVER	GPIO:EXTI	PA6	PA6
YELLOW LED		GPIO:Output	PA10(2)	PF15
BLUE LED		GPIO:Output	PB5(3)	PF14
GREEN LED		GPIO:Output	PB4(4)	PE11

1. *I2C2 in case of NUCLEO-WL55JC1.*
2. *PB12 for NUCLEO-WL55JC1 and PC6 for NUCLEO-WB55RG.*
3. *PC10 for NUCLEO-WB55RG.*
4. *PB8 for NUCLEO-WL55JC1 and PA15 for NUCLEO-WB55RG.*

Figure 14. STM32CubeMX Pinout & Configuration tab and Additional pack settings for NFC07A1_NDEF_URI applications

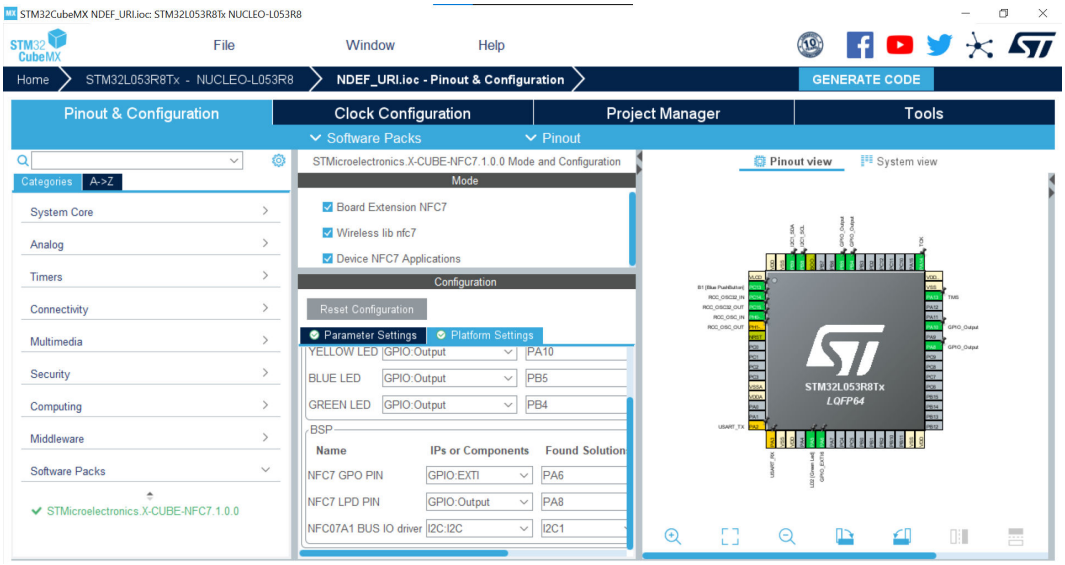


Figure 15. STM32CubeMX Pinout & Configuration tab and Additional pack settings for NFC07A1_I2CProtection example

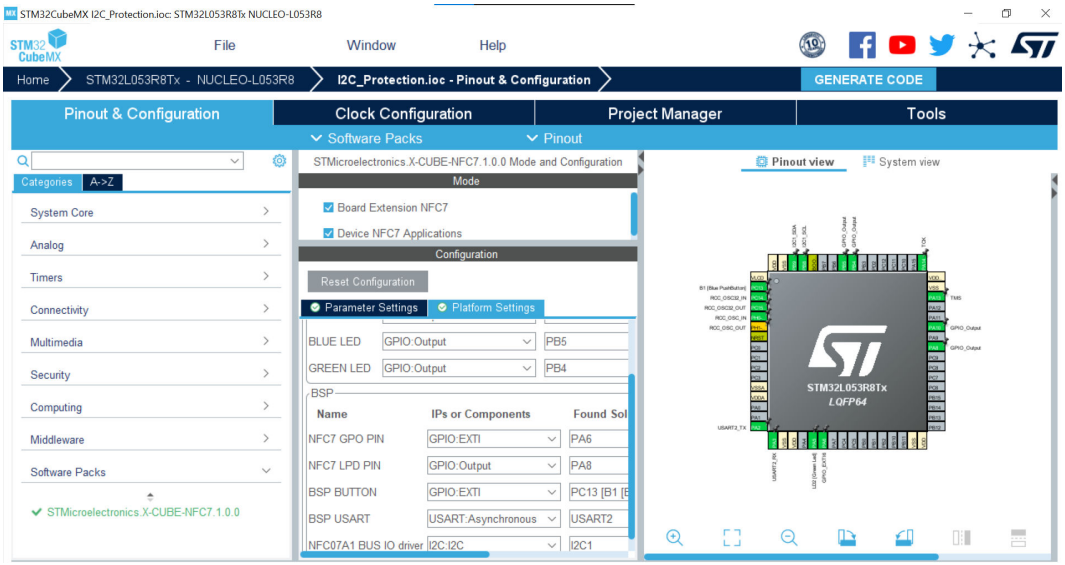


Figure 16. STM32CubeMX Pinout & Configuration tab and Additional pack settings for NFC07A1_LowPowerDown example

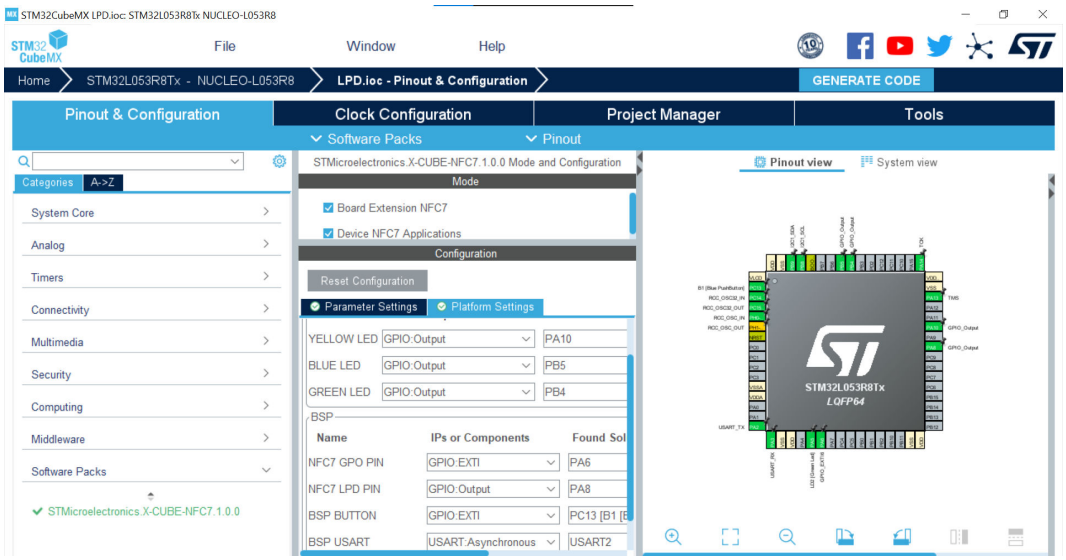
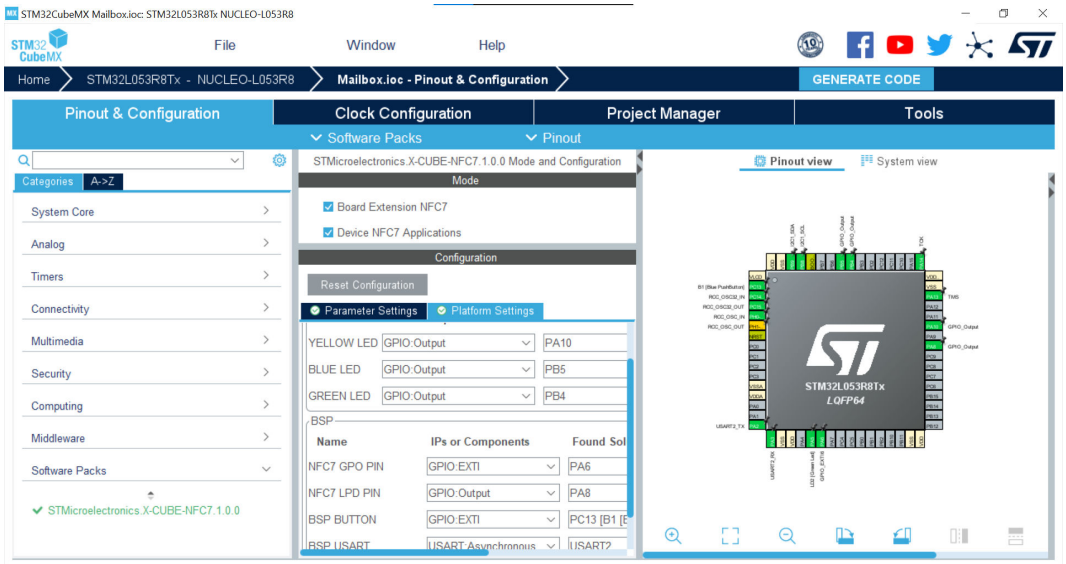


Figure 17. STM32CubeMX Pinout & Configuration tab and Additional pack settings for NFC07A1_Mailbox example



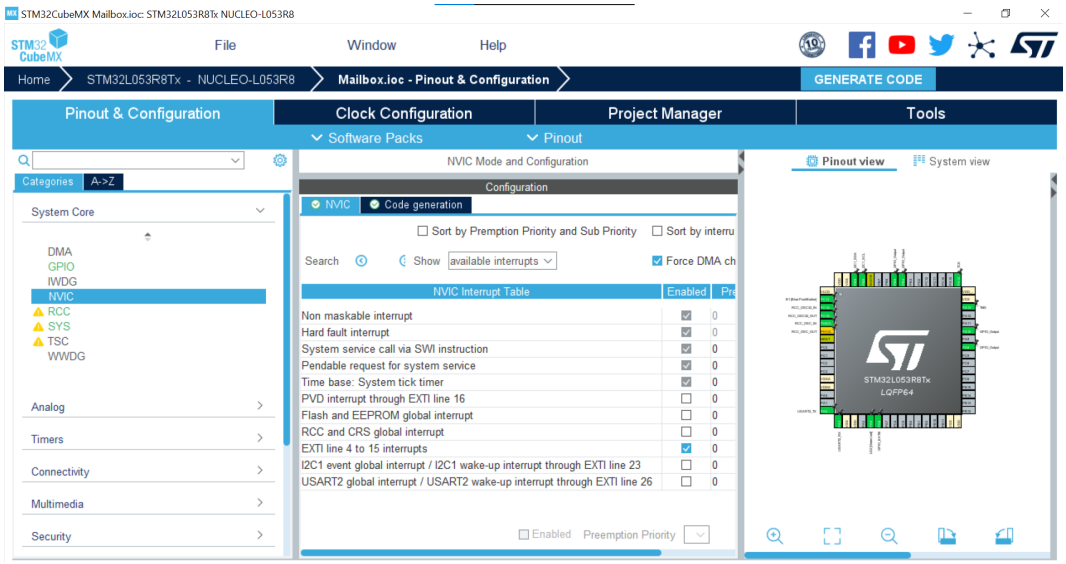
For all the sample applications, the default parameters can be used.

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

Table 6. EXTI Line Selection

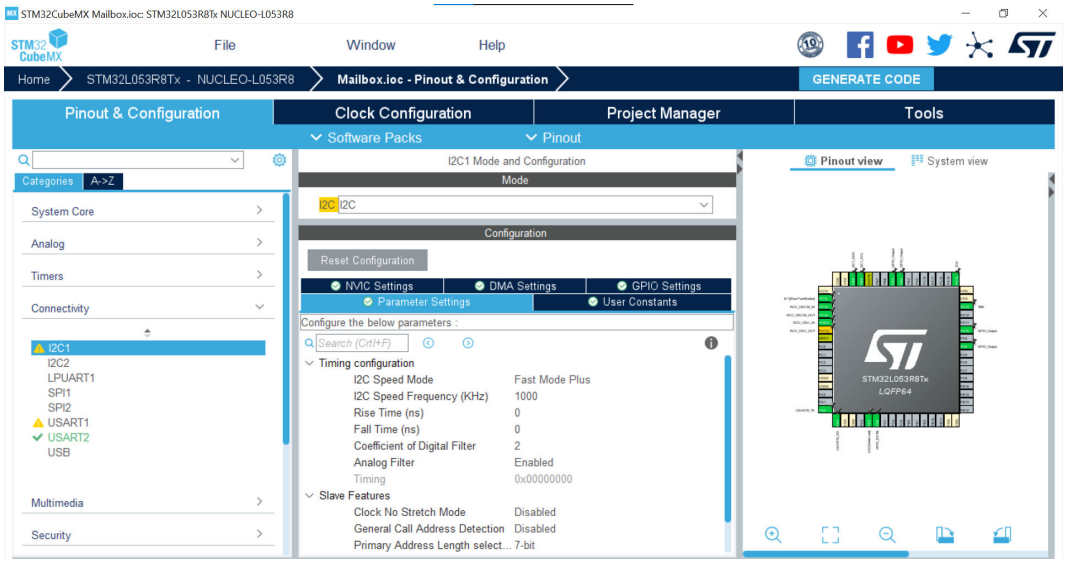
Nucleo 64	Nucleo 144
EXTI line 6 interrupt	EXTI line 6 interrupt
EXTI line 13 interrupt	EXTI line 13 interrupt

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**

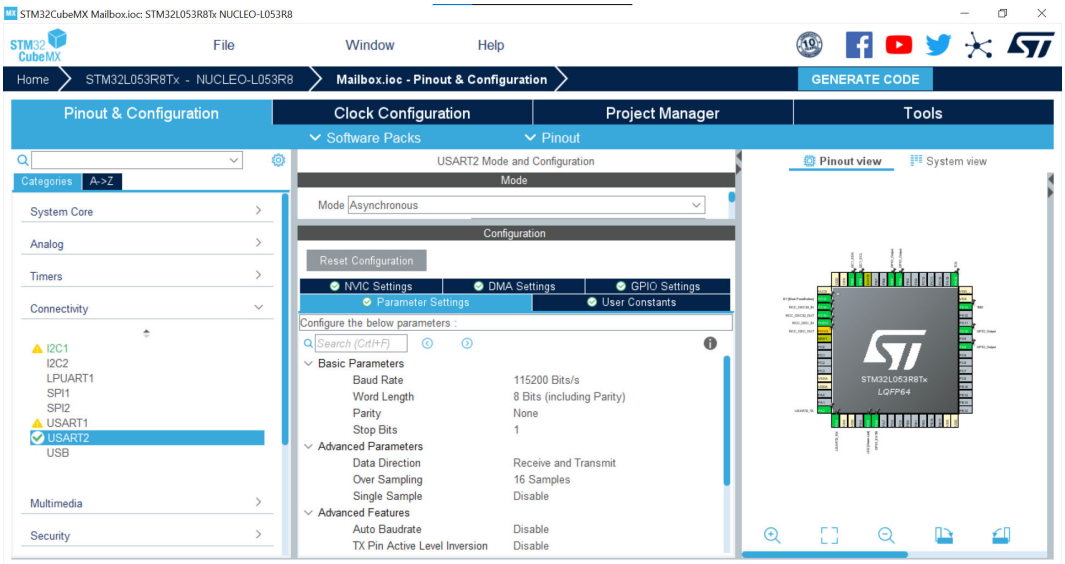


From the **Configuration & Pinout** tab, click on “Connectivity” category and then on USART2 item and check that the following configuration is set:

Table 7. **VCOM configuration**

Baud Rate	Word Length	Parity	Stop Bits
115200 Bits/s	8 Bits (including Parity)	None	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 pack to the project, the “Corresponding pack button” button must be clicked. From the “Corresponding pack 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 Corresponding pack Components selection window**

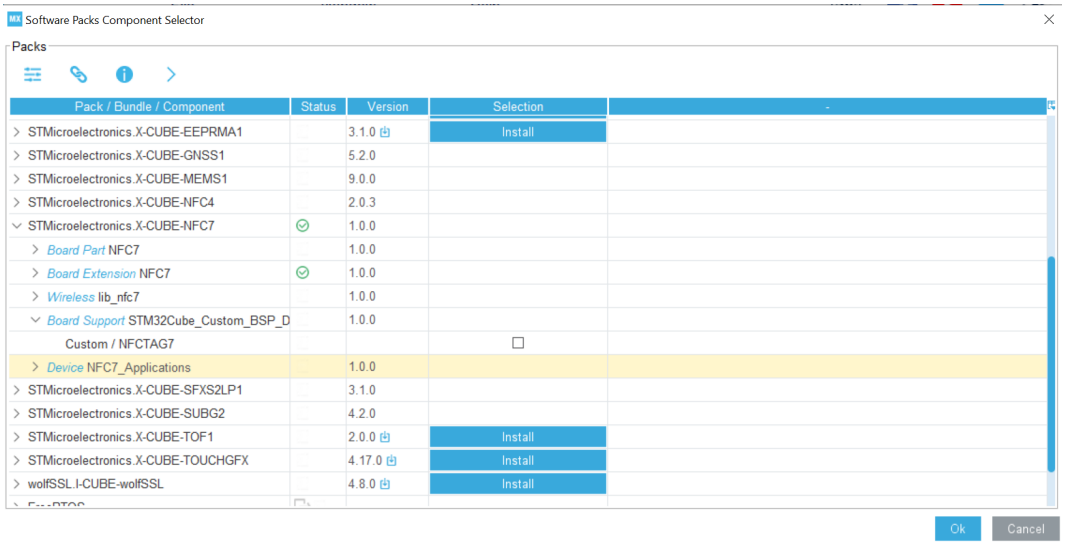
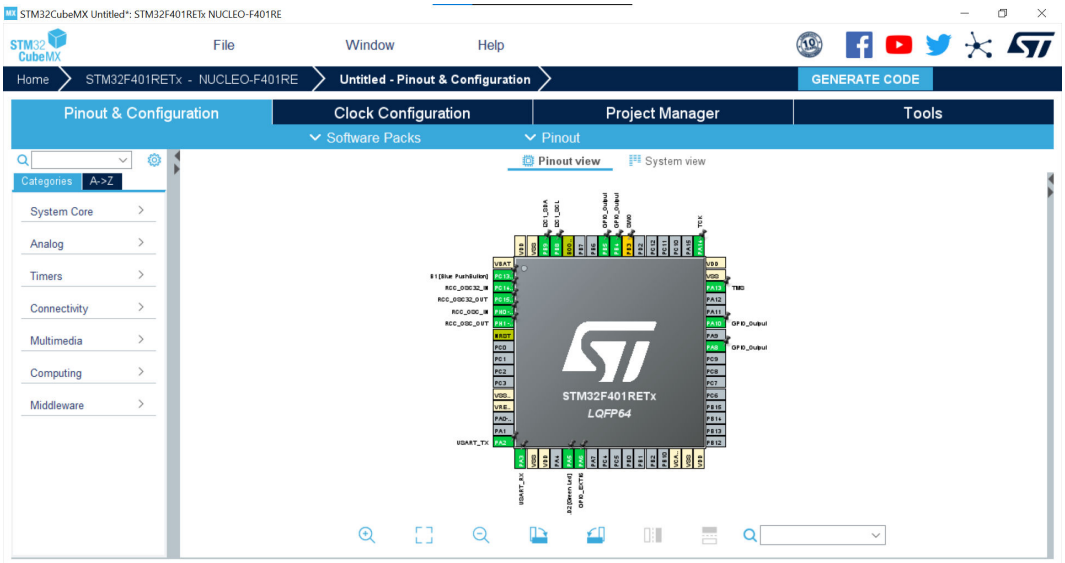


Figure 22. **STM32CubeMX Pinout & Configuration tab**



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:

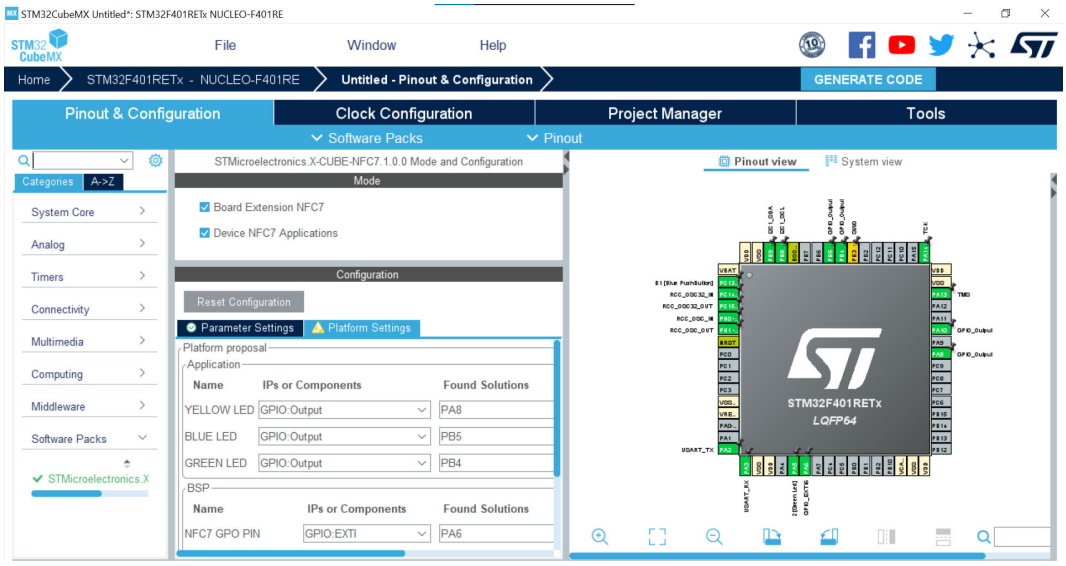
Table 8. **Pinout and configuration**

Nucleo 64			Nucleo 144		
PIN	**Mode**	**Label**	**PIN**	**Mode**	**Label**
PB5(1)	GPIO_Output		PF14	GPIO_Output	
PB4(2)	GPIO_Output		PE11	GPIO_Output	
PA5(3)	GPIO_Output	LD2 [Green Led]	PB7	GPIO_Output	LD2[Blue]
PC13w(4)	GPIO_EXTI13	B1 [Blue PushButton]	PC13	GPIO_EXTI13	USER_Btn[B1]

1. *PC10 for NUCLEO-WB55RG.*
2. *PB8 for NUCLEO-WL55JC1 and PA15 for NUCLEO-WB55RG.*
3. *PB9 for NUCLEO-WL55JC1 and PB0 for NUCLEO-WB55RG.*
4. *PA6 for NUCLEO-WB55RG.*

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:

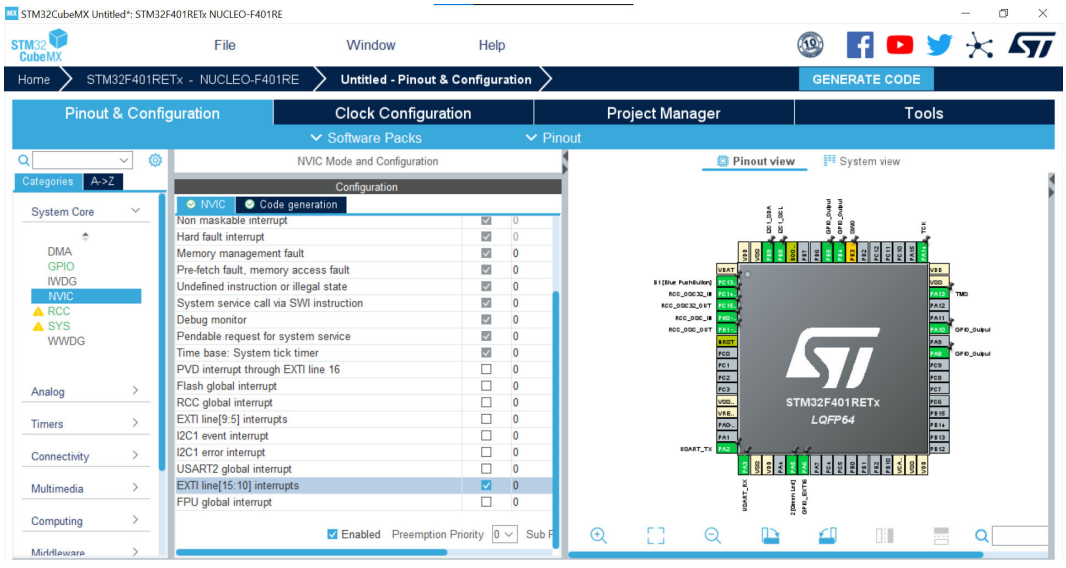
Table 9. **Platform settings**

Name	BSP_Api	Supported IPs	Nucleo 64	Nucleo 144
ST25DVXXKC BUS IO driver	BSP_BUS_DRIVER	I2C:I2C	I2C1(1)	I2C1
ST25DVXXKC GPO PIN	HAL_EXTI_DRIVER	GPIO:EXTI	PA6(2)	PA6
ST25DVXXKC LPD PIN		GPIO:Output	PA8(3)	PF13

1. *I2C2 for NUCLEO-WL55JC1.*
2. *PB4 in case of NUCLEO-WB55RG.*
3. *PC1 for NUCLEO-WL55JC1, PE4 for NUCLEO-WB55RG.*

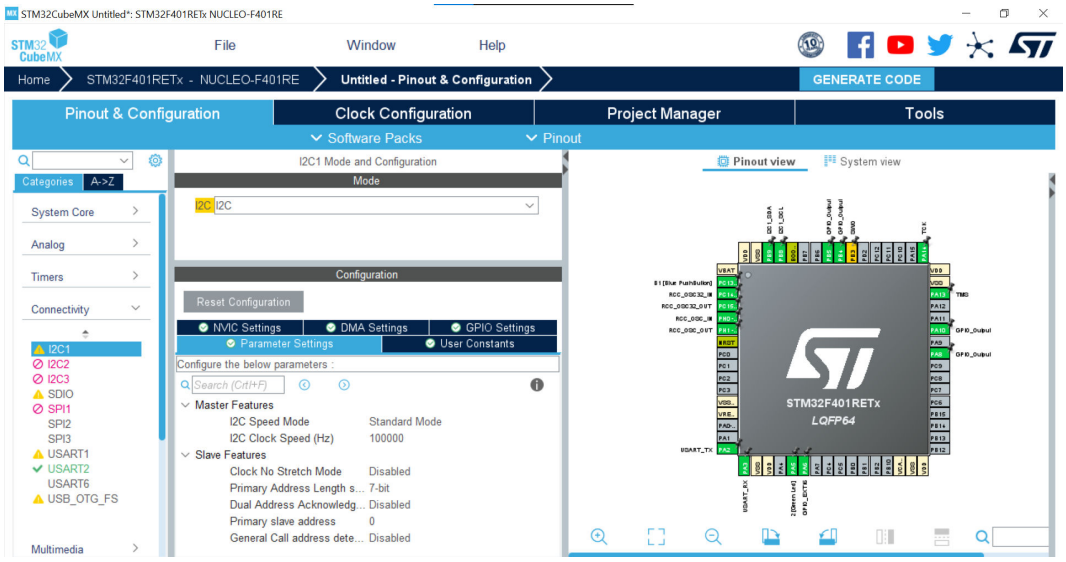
From the **Configuration & Pinout** tab, click on “System Core” category and then on NVIC item to enable the EXTI line interrupts.

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

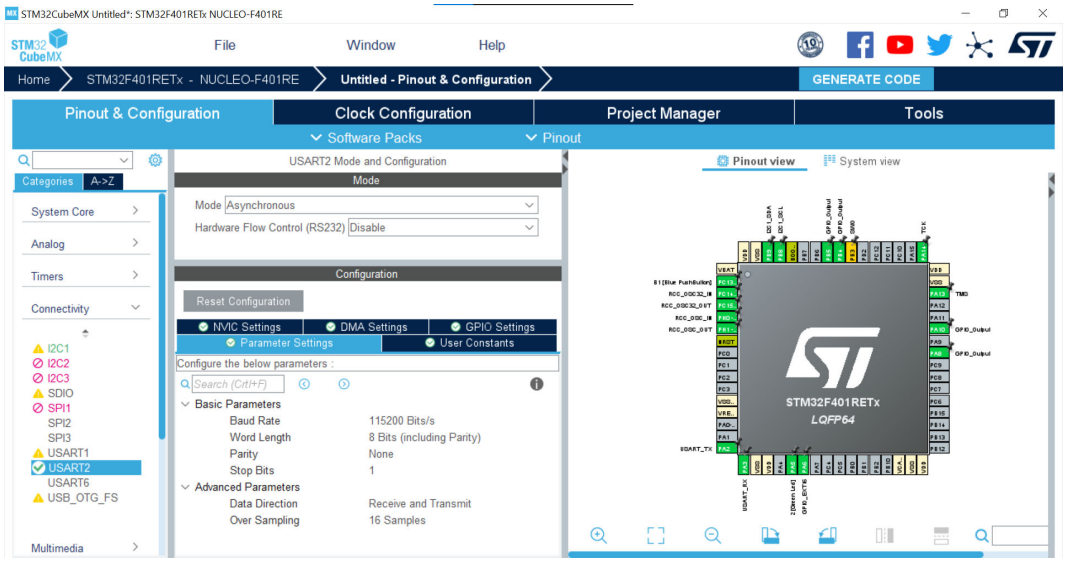


From the **Configuration & Pinout** tab, click on “Connectivity” category and then on USART2 button and check the following configuration is set:

Table 10. VCOM /UART configuration

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

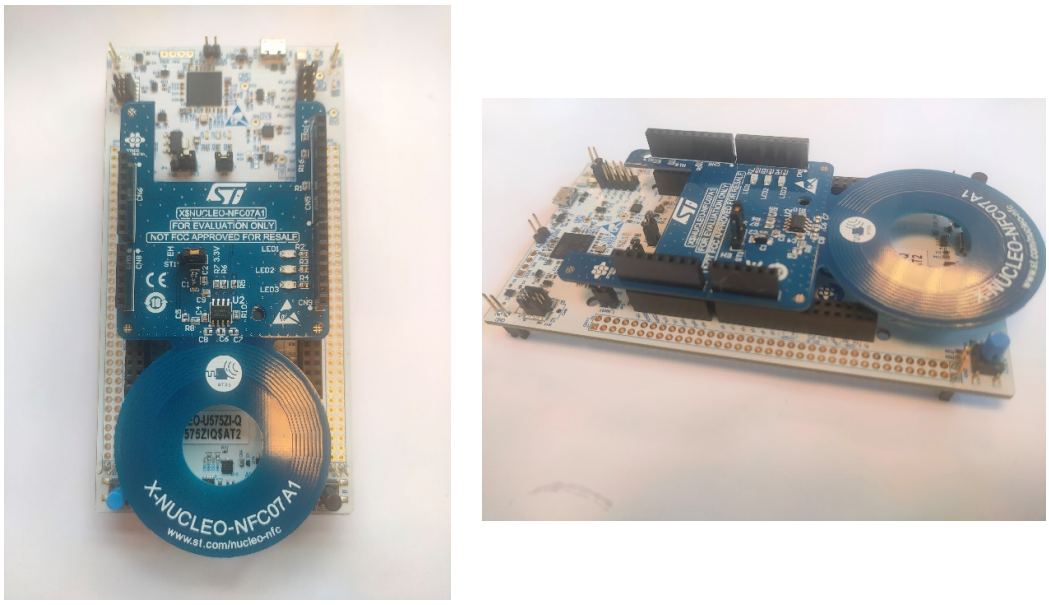
7.4 Porting of X-NUCLEO-NFC07A1 step by step guide

In this Section, we will see how to use the X-NUCLEO-NFC07A1 expansion board on any Nucleo board.
In this example, we will do a porting on a Nucleo 144 board NUCLEO-U575ZI-Q:

Figure 27. NUCLEO-144

The following pictures show how the X-NUCLEO-NFC07A1 expansion board can be stacked on a Nucleo 144 board:

Figure 28. **X-NUCLEO-NFC07A1 stacked over NUCLEO-144**



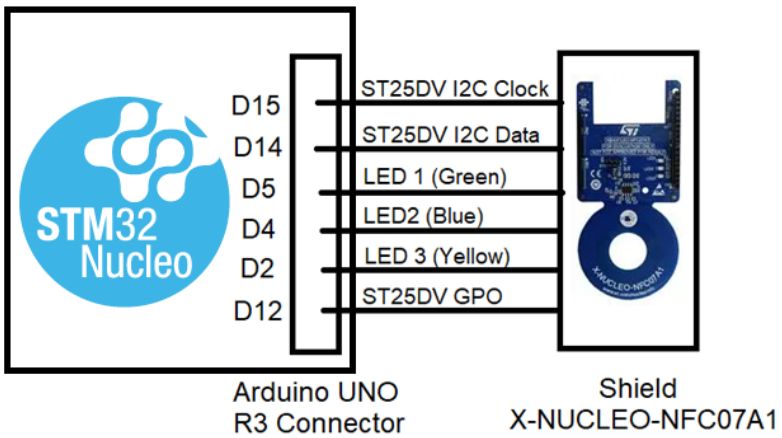
7.4.1 Overview

The X-NUCLEO-NFC07A1 expansion board is using the following pins:

- 2 pins for the I2C bus (Clock and Data)
- 3 pins for the LEDs
- 1 pin for the GPO (= General Purpose Output)

They are connected to the following pins of the Arduino UNO connector:

Figure 29. **Connection of Arduino connector with X-NUCLEO-NFC07A1**



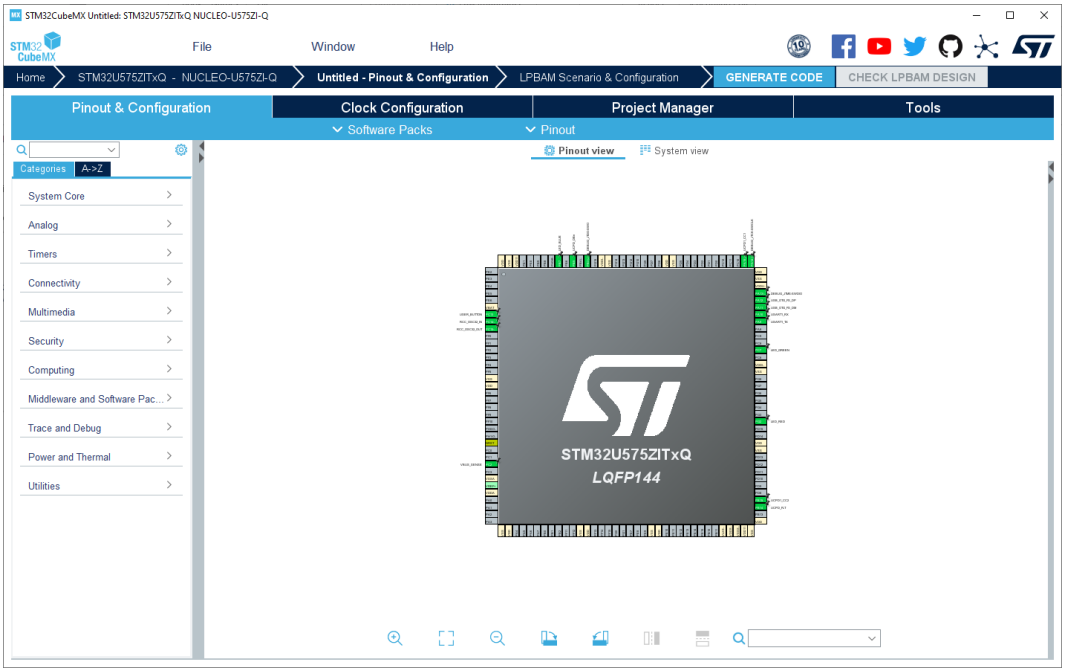
The GPIOs of the Nucleo board should be configured to drive the I2C Bus, the GPO signal and the LEDs of the X-NUCLEO-NFC07A1 expansion board.

7.4.2 Project Creation

In STM32CubeMX, click on “Access to board selector”, select the board NUCLEO-U575ZI-Q and start a project :

Figure 30. **Access to board selector**

Figure 31. **Pinout and configuration tab**



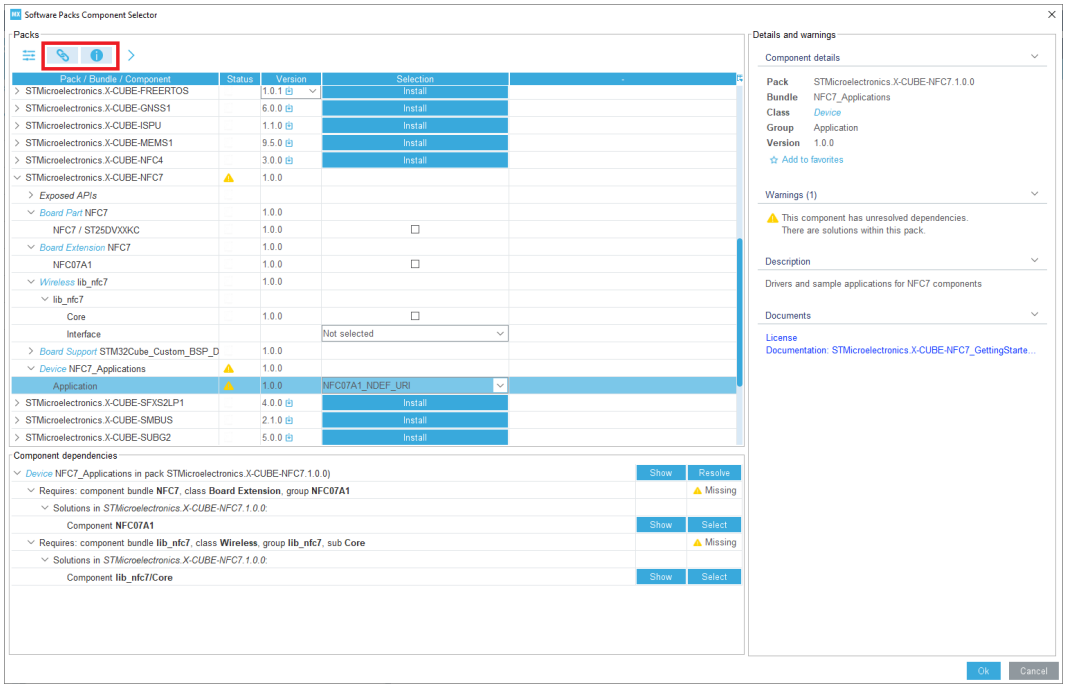
7.4.3 Import of X-CUBE-NFC7 software pack:

In the Pinout & Configuration tab, click on Software packs -> Select component (NB: It is also possible to click on Middleware and Software packs -> X-CUBE-NFC7).

Expand X-CUBE-NFC7 and select Device NFC7_Applications -> Application.

Note: Click on the 2 icons circled in red to enable the information pane and the dependency pane:

Figure 32. **Component Selector**

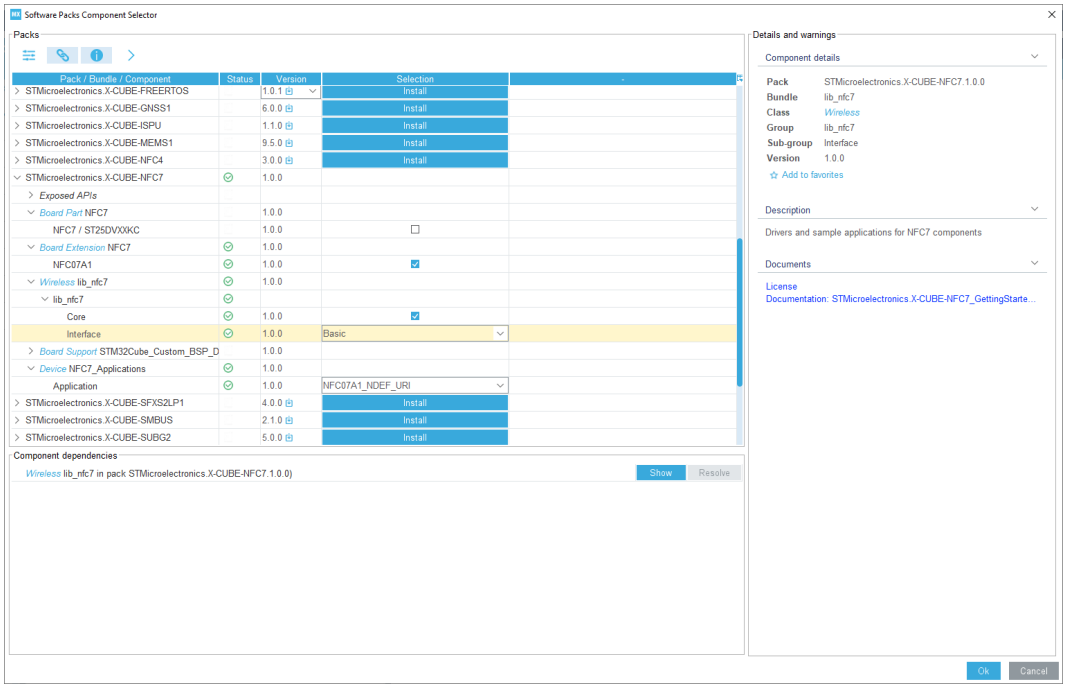


In this tutorial, we will use the NFC07A1_NDEF_URI example. Select this application in the drop down menu.

At that step, some exclamation marks are shown to indicate some dependency problems. Click on “Resolve” to fix them.

This will select the needed dependencies:

Figure 33. **Component Selection**

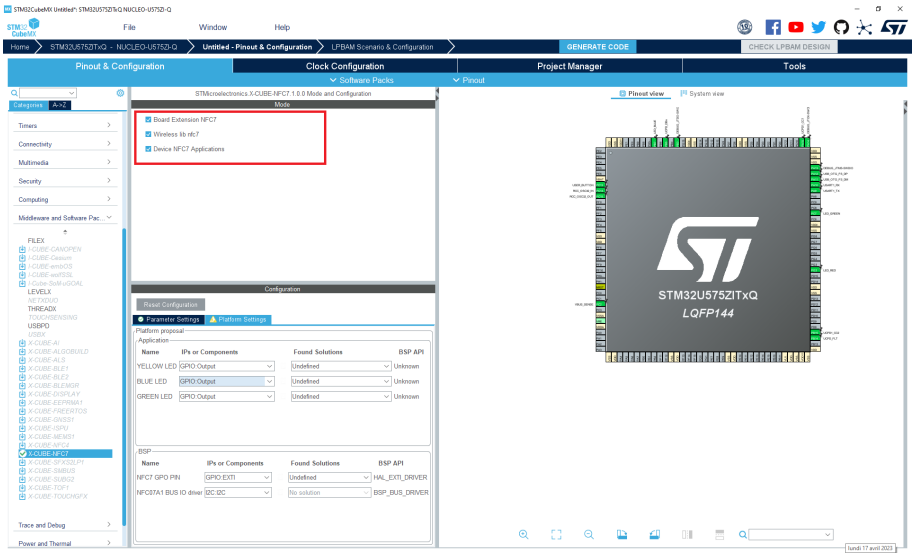


Click on OK button to validate.

7.4.4 Configuration of X-CUBE-NFC7

In Middleware and Software packs, tick the following boxes:

Figure 34. Modes and Configuration selection



7.4.5 Configuration of the I2C bus

- Open the User Manual of the X-NUCLEO-NFC07A1 expansion board and search which signals of the Arduino connector are used for I2C Clock and Data (SCL and SDA). This is D15 (for I2C clock) and D14 (for I2C data):

Figure 35. UM2860 (Table 2)

Signal	Connector	Pin number	X-NUCLEO-NFC07A1 expansion board
D15		10	ST25DV_SCL
D14	CN5 Digital	9	ST25DV_SDA
AREF		8	-

- Open the User Manual of the Nucleo board and seach for D14 and D15:

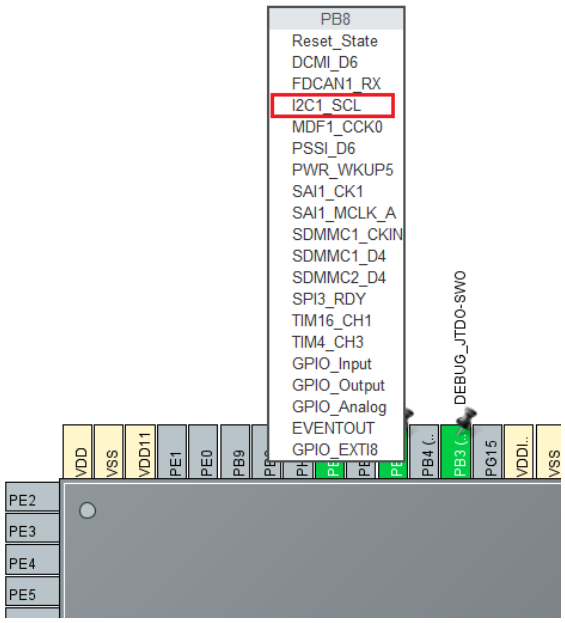
Figure 36. UM2861 (Table 15)

MCU function	STM32 pin	Signal name	Pin name	Pin	Pin	Pin name	Signal name	STM32 pin	MCU function
SAI2_A	PC6	SAI_C_MCLK	D16	1	2	D15	I2C_A_SCL	PB8	I2C1
SAI2_A	PD11	SAI_C_SD	D17	3	4	D14	I2C_A_SDA	PB9	I2C1
SAI2_A	PB13	SAI_C_SCK	D18	5	6	VREFP	-	-	-
SAI2_A	PD12	SAI_C_FS	D19	7	8	GND	-	-	-
SAI1_B/ SPI3	PA4	SAI_D_FS	D20(7)	9	10	D13	SPI_A_SCK	PA5	SPI1
SAI1_B/ SPI3	PB4	SAI_D_MCLK	D21 (7)(8)	11	12	D12	SPI_A_MISO	PA6	SPI1
SAI1_B/ SPI3	PB5	SAI_D_SD/ SPI_B_MOSI	D22 (5)(7)(8)	13	14	D11	SPI_A_MOSI/ TIM_E_PWM1	PA7	SPI1
SAI1_B/ SPI3	PB3	SAI_D_SCK/ SPI_B_SCK	D23(6) (7)(8)	15	16	D10	SPI_A_CS / TIM_B_PWM3(1)	PD14	SPI1/ TIM4_CH3
SAI1/SPI3	PA4	SPI_B_NSS	D24 (8)	17	18	D9	TIM_B_PWM2	PD15	TIM4_CH4
SAI1/SPI3	PB4	SPI_B_MISO	D25 (8)	19	20	D8	IO	PF12	-

For this Nucleo board, they are connected to GPIOs PB8 and PB9.

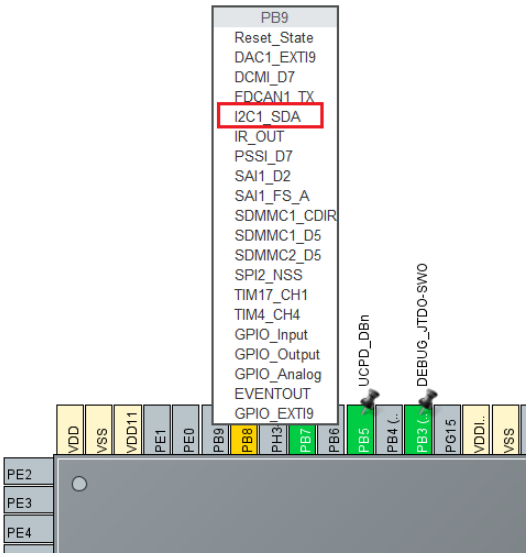
- In the Pinout view, select PB8 and connect it to an I2C Clock. In this example, we connect it to I2C1_SCL:

Figure 37. I2C clock line selection



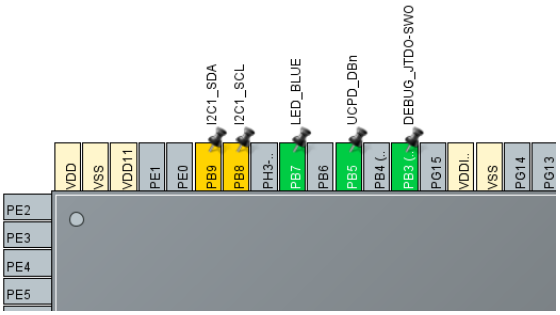
- In the Pinout view, select PB9 and connect it to an I2C Data. In this example, we connect it to I2C1_SDA:

Figure 38. I2C Data line selection



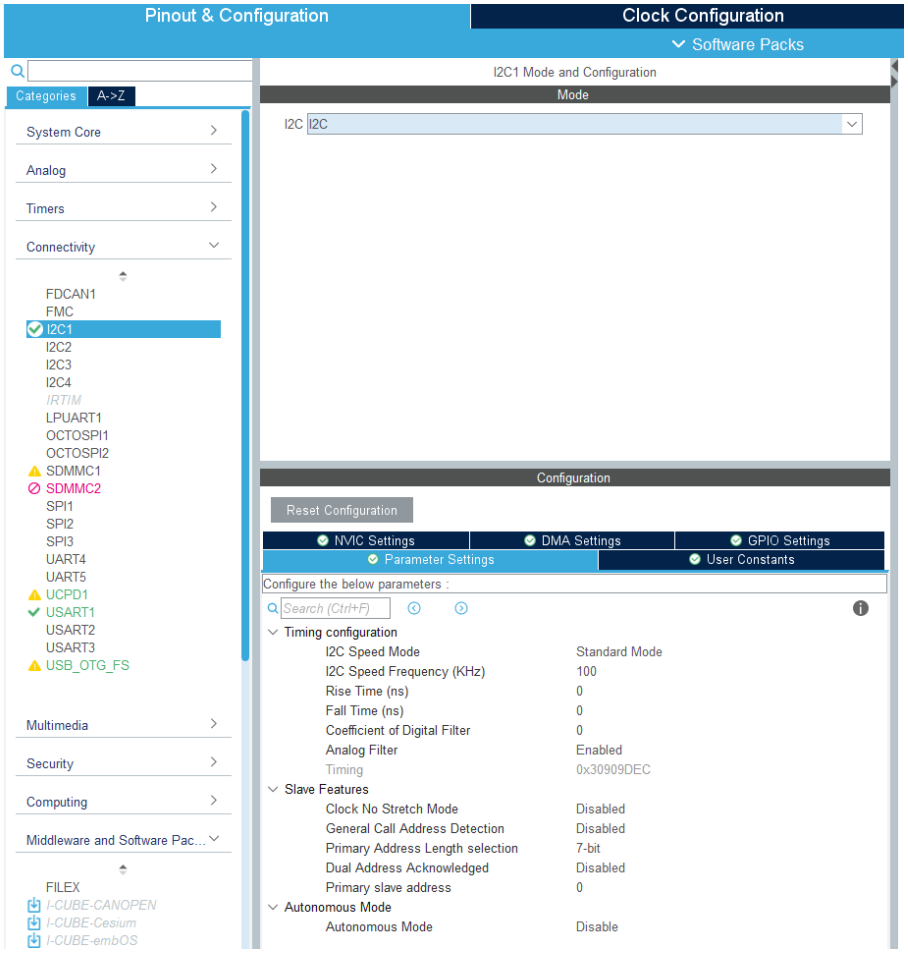
We get the following result:

Figure 39. All the pins selected for use



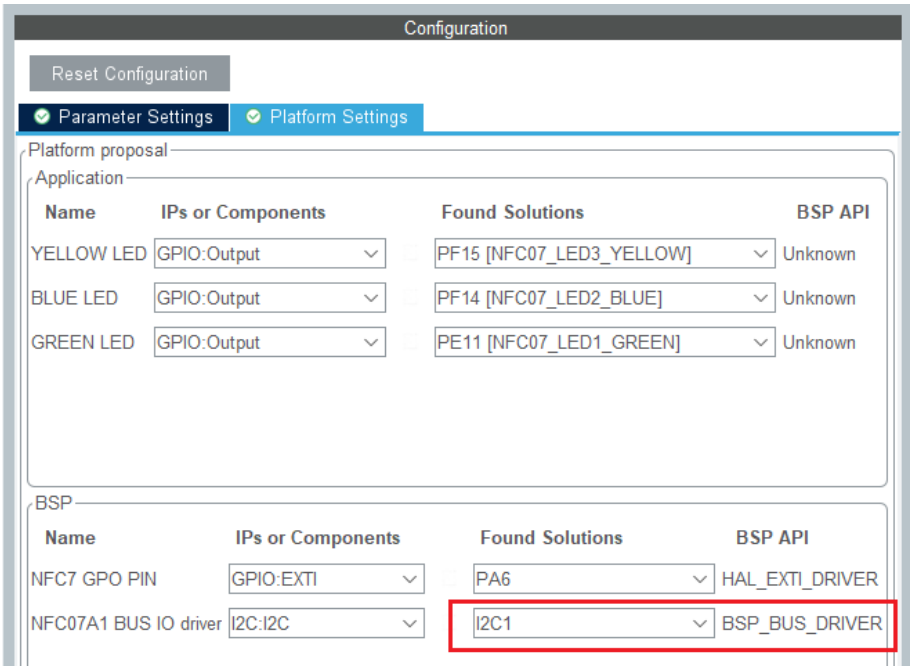
- We should now configure the I2C bus (I2C1 in this example).
  Go to Connectivity -> I2C1 and set the configuration of this I2C bus:

Figure 40. I2C Mode setting



- Go back to Middleware and Software packs -> X-CUBE-NFC7. In the Platform Settings, indicate the I2C bus to use to drive the NFC07A1 board:

Figure 41. Platform Settings



7.4.6 Configuration of the GPO

- Open the User Manual of the X-NUCLEO-NFC07A1 expansion board and search which signal of the Arduino connector is connected to the GPO. This is D12:

Figure 42. Pin Mapping from Arduino and MCU - UM2860 (Table 2)

Signal	Connector	Pin number	X-NUCLEO-NFC07A1 expansion board
GND		7	GND
D13		6	-
D12		5	ST25DV_GPO
D11	CN5 Digital	4	-
D10		3	-
D9		2	-
D8		1	-
D7	CN9 Digital	8	-
D6		7	-
D5		6	MCU_LED1
D4		5	MCU_LED2
D3		4	-
D2		3	MCU_LED3
D1		2	-
D0		1	-

- Open the User Manual of the Nucleo board and search for D12:

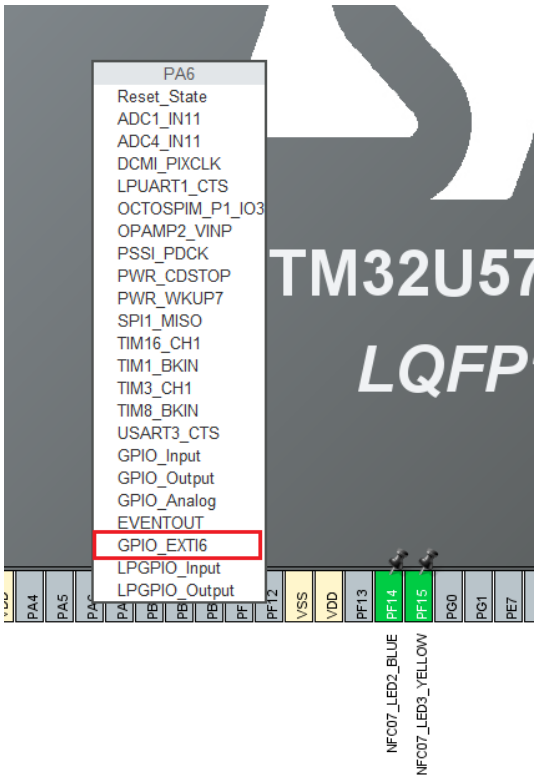
Figure 43. Pinout of the connectoe - UM2861 (Table 15)

MCU function	STM32 pin	Signal name	Pin name	Pin	Pin	Pin name	Signal name	STM32 pin	MCU function
SAI2_A	PC6	SAI_C_MCLK	D16	1	2	D15	I2C_A_SCL	PB8	I2C1
SAI2_A	PD11	SAI_C_SD	D17	3	4	D14	I2C_A_SDA	PB9	I2C1
SAI2_A	PB13	SAI_C_SCK	D18	5	6	VREFP	-	-	-
SAI2_A	PD12	SAI_C_FS	D19	7	8	GND	-	-	-
SAI1_B/ SPI3	PA4	SAI_D_FS	D20(7)	9	10	D13	SPI_A_SCK	PA5	SPI1
SAI1_B/ SPI3	PB4	SAI_D_MCLK	D21 (7)(8)	11	12	D12	SPI_A_MISO	PA6	SPI1
SAI1_B/ SPI3	PB5	SAI_D_SD/ SPI_B_MOSI	D22 (5)(7)(8)	13	14	D11	SPI_A_MOSI/ TIM_E_PWM1	PA7	SPI1
SAI1_B/ SPI3	PB3	SAI_D_SCK/ SPI_B_SCK	D23(6) (7)(8)	15	16	D10	SPI_A_CS / TIM_B_PWM3(1)	PD14	SPI1/ TIM4_CH3
SAI1/SPI3	PA4	SPI_B_NSS	D24 (8)	17	18	D9	TIM_B_PWM2	PD15	TIM4_CH4
SAI1/SPI3	PB4	SPI_B_MISO	D25 (8)	19	20	D8	IO	PF12	-

For this Nucleo board, D12 is connected to GPIO PA6.

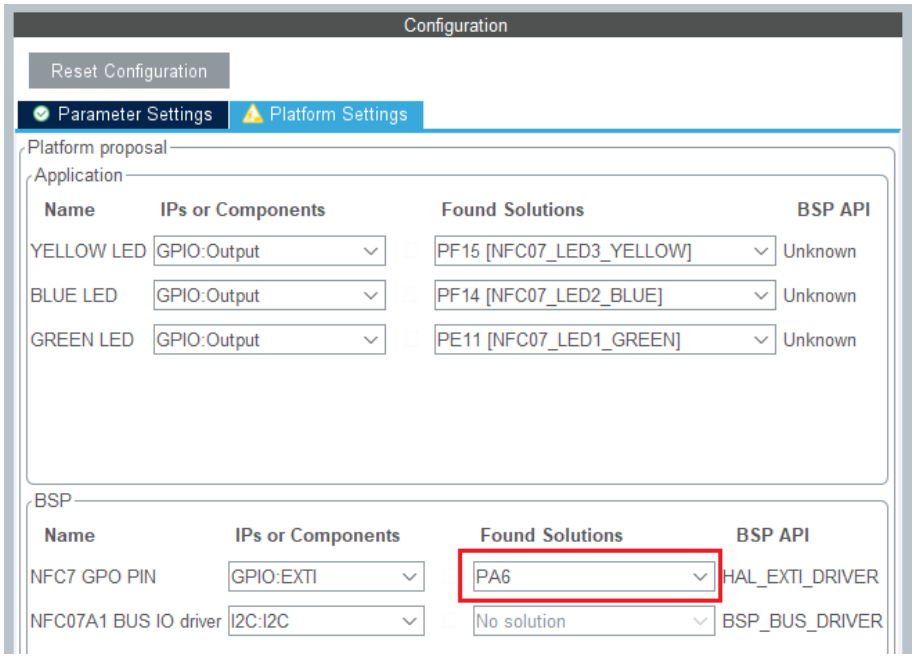
- In the Pinout view, select PA6 and set it as GPIO External Interrupt (GPIO_EXTI6):

Figure 44. External Interrupt Pin selection



- In X-CUBE-NFC7 platform settings, configure NFC7 GPO PIN:

Figure 45. Platform Setting



7.4.7 Configuration of the LEDS

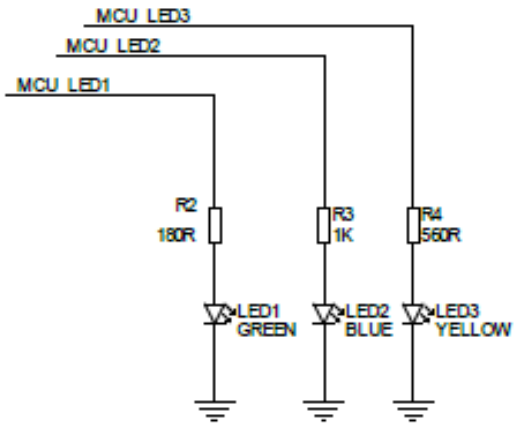
- Open the User Manual of the X-NUCLEO-NFC07A1 expansion board and search which signals of the Arduino connector are connected to the LEDs. This is D2, D4, D5:

Figure 46. LED Pins Selection - UM2860 (Table 2)

Signal	Connector	Pin number	X-NUCLEO-NFC07A1 expansion board
GND	CN5 Digital	7	GND
D13		6	-
D12		5	ST25DV_GPO
D11		4	-
D10		3	-
D9		2	-
D8		1	-
D7	CN9 Digital	8	-
D6		7	-
D5		6	MCU_LED1
D4		5	MCU_LED2
D3		4	-
D2		3	MCU_LED3
D1		2	-
D0		1	-

LED1 is the Green LED, LED2 is the Blue LED and LED3 is the Yellow LED:

Figure 47. LED Pins schematics



- Open the User Manual of the Nucleo board and seach for D2, D4, D5:

Figure 48. **Connector pinout - UM2861 (Table 18)**

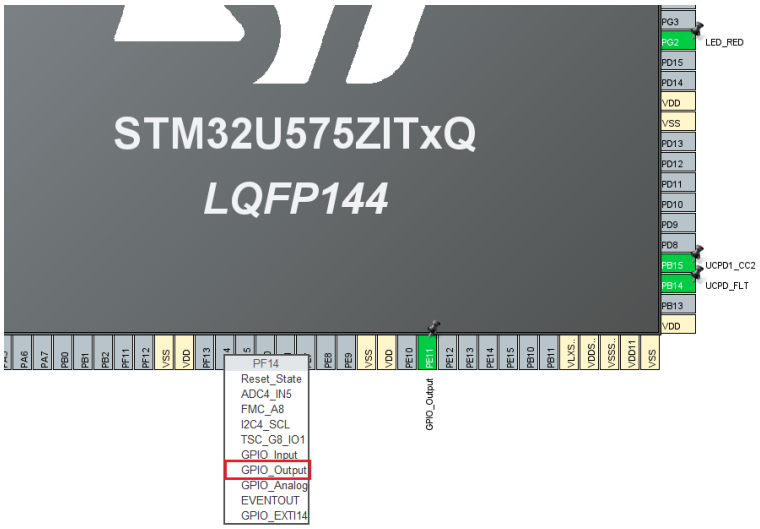
MCU function	STM32 pin	Signal name	Pin name	Pin	Pin	Pin name	Signal name	STM32 pin	MCU function
AVDD	-	-	AVDD	1	2	D7	IO	PF13	IO
AGND	-	-	AGND	3	4	D6	TIM_A_PWM1	PE9	TIM1_CH1
GND	-	-	GND	5	6	D5	TIM_A_PWM2	PE11	TIM1_CH2
ADC1_IN16	PB1	ADC_A_IN	A6	7	8	D4	IO	PF14	IO
ADC1_IN3	PC2	ADC_B_IN	A7	9	10	D3	TIM_A_PWM3	PE13	TIM1_CH3
ADC1_IN6	PA1	ADC_C_IN	A8	11	12	D2	IO	PF15	IO
OCTOSPI1	PA2	OCTOSPI_CS	D26(3)	13	14	D1(1)	USART_A_TX	PG7(2)	LPUART1
OCTOSPI1	PB10	OCTOSPI_CLK	D27(4)	15	16	D0(1)	USART_A_RX	PG8(2)	LPUART1
-	-	-	GND	17	18	D42	TIM_A_PWM1N	PE8	TIM1_CH1N
OCTOSPI1	PE15	OCTOSPI_IO3	D28(4)	19	20	D41	TIM_A_ETR	PE7	TIM1_ETR
OCTOSPI1	PB0	OCTOSPI_IO1	D29(3)	21	22	GND	-	-	-
OCTOSPI1	PE12	OCTOSPI_IO0	D30(4)	23	24	D40	TIM_A_PWM2N	PE10	TIM1_CH2N
OCTOSPI1	PE14	OCTOSPI_IO2	D31(4)	25	26	D39(4)	TIM_A_PWM3N	PE12	TIM1_CH3N
-	-	-	GND	27	28	D38(4)	TIM_A_BKIN2	PE14	TIM1_BKIN2
TIM2_CH1	PA0	TIM_C_PWM1	D32	29	30	D37(4)	TIM_A_BKIN1	PE15	TIM1_BKIN
TIM1_CH1	PA8	TIM_D_PWM1	D33	31	32	D36(4)	TIM_C_PWM2	PB10	TIM2_CH3
TIM4_ETR	PE0	TIM_B_ETR	D34	33	34	D35	TIM_C_PWM3	PB11	TIM2_CH4

1. *The default configuration for the D0/D1 signal is LPUART1 on PG7 and PG8, USART1 on PA9 and PA10 is connected by default on STLINK-V3E.*

For this Nucleo board, D2, D4, D5 are connected to GPIOs PF15, PF14 and PE11.

- In the Pinout view, select PF14 and set it as GPIO_Output:

Figure 49. **GPIO_Output selection**



- Add a label to this output (for example “NFC07_LED2_BLUE”):

Figure 50. **Entering user Label**

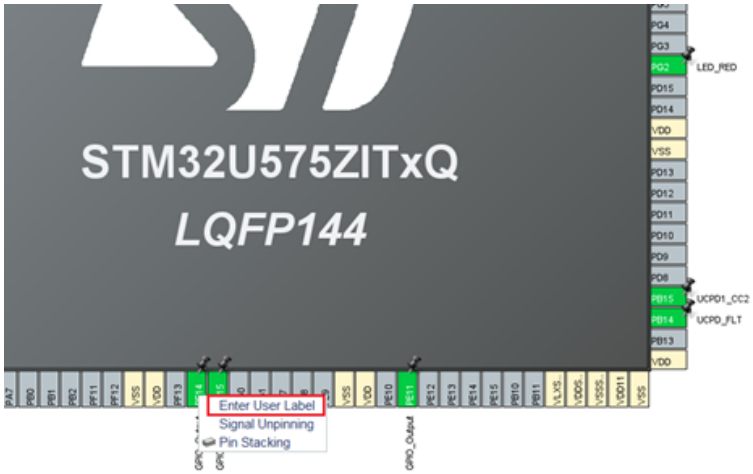
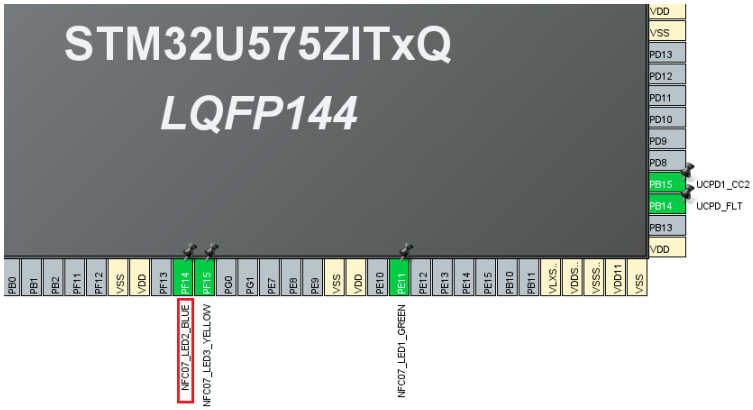
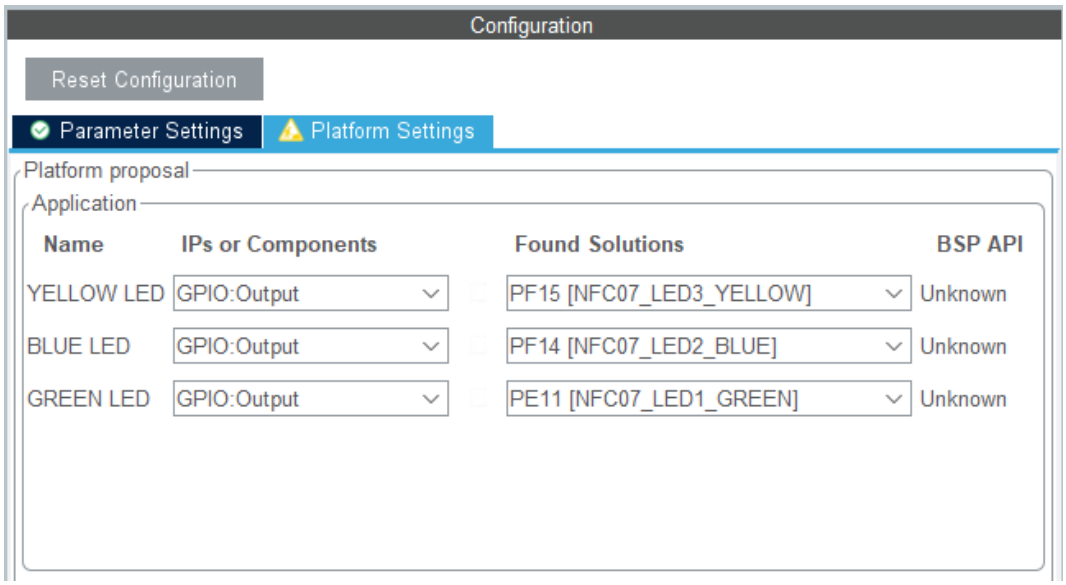


Figure 51. **LED selection user Label**



- In X-CUBE-NFC7 platform settings, configure the blue led to map it to PF4 NFC07_LED2_BLUE:

Figure 52. **Blue LED selection**



- Then do the same for the other LEDs.
  The X-NUCLEO-NFC07A1 expansion board is then ready to use. Generate the source code and build the project.

8 Generated Folders Structure

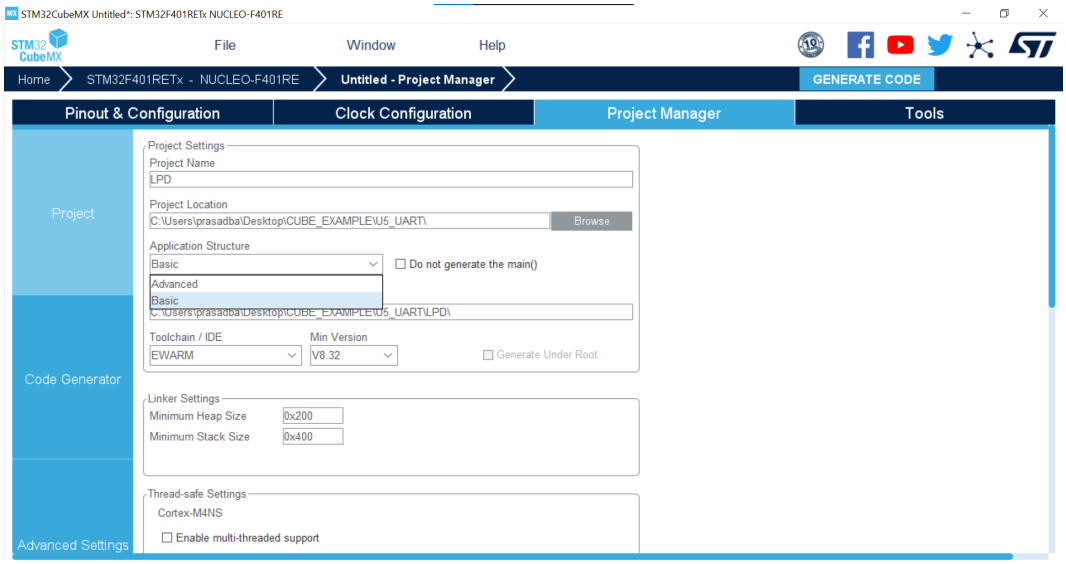
When generating a project, two models of folders structure can be adopted when using a high level firmware component (i.e. a middleware in the STM32Cube MCU package):

- **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 53. STM32CubeMX Application Structure Configuration



9 Known limitations and workarounds

STM32CubeMX NFC7 pack v1.0.0 is fully compatible with STM32CubeMX v 6.4.0 and upwards. It is not fully compatible with the previous version of STM32CubeMX (<= v 6.4.0).

10 References

UM2961 User Manual - *Getting started with the X-CUBE-NFC7 dynamic NFC/RFID tag IC software expansion for STM32Cube*

Revision history

Table 11. **Document revision history**

Date	Revision	Changes
04-Jun-2024	1	Initial release.

Contents

List of tables

List of figures