Getting started with the X-CUBE-NFC6 high performance HF reader/NFC initiator. IC software expansion for STM32Cube

Introduction

The X-CUBE-NFC6 software expansion for STM32Cube provides complete middleware for STM32 to control applications using the ST25R3916/ST25R3916B high performance NFC front-end IC supporting NFC initiator, target, reader, and card emulation modes.

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

The software comes with sample implementations of the drivers running on the X-NUCLEO-NFC06A1/X-NUCLEO-NFC08A1 expansion board plugged on top of a NUCLEO-G0B1RE or NUCLEO-L476RG board.

Contents

List of tables

List of figures

1 Acronyms and abbreviations

Table 1. **List of acronyms**

Acronym	Description
NFC	Near field communication
RFAL	RF abstract layer
P2P	Peer to peer
MCU	Microcontroller unit
BSP	Board support package
HAL	Hardware abstraction layer
LED	Light emitting diode
SPI	Serial peripheral interface
CMSIS	The ARM® Cortex® microcontroller software interface standard

2 X-CUBE-NFC6 software expansion for STM32Cube

2.1 Overview

The X-CUBE-NFC6 software package expands the STM32Cube functionality.

The package key features are:

- Complete middleware to build applications using the ST25R3916/ST25R3916B high performance HF reader/NFC front end IC
- Sample application to detect NFC tags of distinct types and mobile phones supporting P2P, card emulation mode and read/write.
- Sample application to read and write NDEF messages.
- Sample implementations available for the X-NUCLEO-NFC06A1/X-NUCLEO-NFC08A1 expansion board plugged onto a NUCLEO-G0B1RE or NUCLEO-L476RG development board
- Easy portability across different MCU families, thanks to STM32Cube
- Complete RF/NFC abstraction (RFAL) for all major technologies including complete ISO-DEP and NFC-DEP layers.
- Free, user-friendly license terms

This software contains high performance HF reader/NFC front end IC drivers for the ST25R3916/ST25R3916B device, running on STM32. It is built on top of STM32Cube software technology to ease portability across different STM32 microcontrollers.

This firmware package includes component device drivers, a board support package and a sample application demonstrating usage of X-NUCLEO-NFC06A1/X-NUCLEO-NFC08A1 expansion board with STM32 Nucleo boards.

A sample application configures the ST25R3916/ST25R3916B in a polling loop for active and passive device detection. When a passive tag or active device is detected, the reader field signals the detected technology by switching a corresponding LED on. It is also possible to set the ST25R3916/ST25R3916B in an inductive wake-up mode by pressing the user button. During this polling loop the sample application also sets the ST25R3916/ST25R3916B in card emulation mode to detect the presence of a reader.

The demo logs all activities with ST-LINK Virtual Com Port to the host system.

The supported RFID technologies in this demo are:

- ISO14443A/NFCA
- ISO14443B/NFCB
- Felica/NFCF
- ISO15693/NFCV
- Active P2P
- Card Emulation Type A and F

The second sample application uses the lib NDEF middleware to read or write NDEF messages from/to tags. It is available in a second project target for the three development environment tools.

2.2 Architecture

This fully compliant software expansion for STM32Cube lets you develop applications using the ST25R3916/ST25R3916B high performance HF reader/NFC initiator IC. It is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller and extends STM32Cube with a board support package (BSP) for the X-NUCLEO-NFC06A1/X-NUCLEO-NFC08A1 expansion board.

Application software can access and use the X-NUCLEO-NFC06A1/X-NUCLEO-NFC08A1 expansion board through the following layers:

- **STM32Cube HAL layer**: the HAL driver layer provides a simple set of generic, multi-instance APIs (application programming interfaces) to interact with the upper layers (application, libraries and stacks). These generic and extension APIs are directly built on a common architecture and allow overlying layers like middleware to implement their functions without depending on specific microcontroller unit (MCU) hardware information. This structure improves the library code reusability and guarantees easy portability across other devices.

- **Board support package (BSP) layer** provides support for the peripherals on the STM32 Nucleo board (apart from the MCU). This set of APIs provides a programming interface for certain board-specific peripherals like the LED, the user button etc. This interface also helps you identify the specific board version.
- **Middleware NRF abstraction layer (RFAL)**: RFAL provides several functions for RF/NFC communication. It groups the different RF ICs (existing ST25R3911B product family and future ST25R391x devices) under a common and easy to use interface.

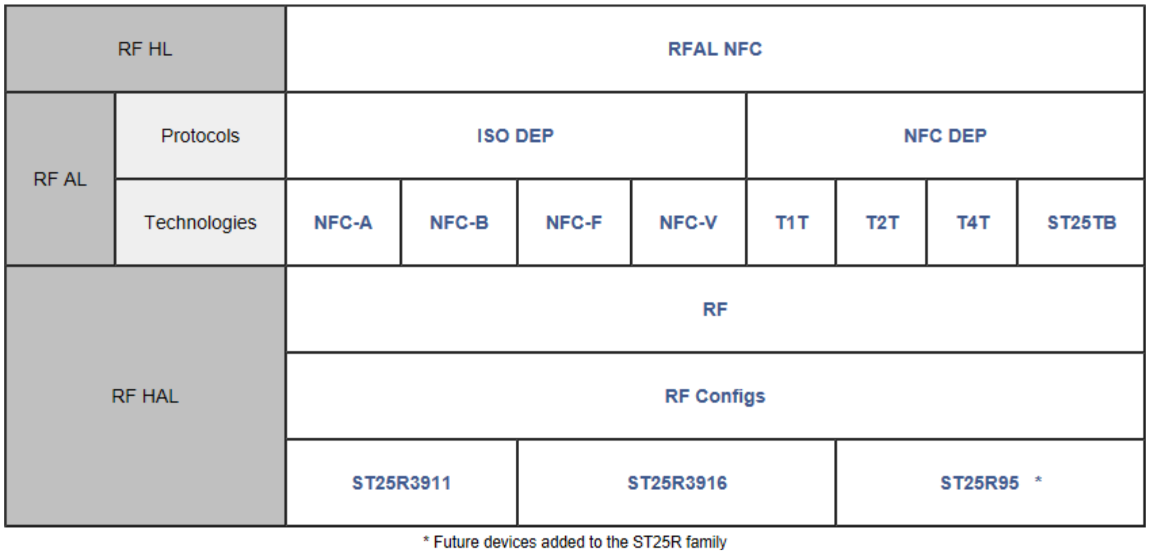
The protocols provided by RFAL are:

- ISO-DEP (ISO14443-4 Data Link Layer, T=CL)
- NFC-DEP (ISO18092 Data Exchange Protocol)
- NFC-A \ ISO14443A (T1T, T2T, T4TA)
- NFC-B \ ISO14443B (T4TB)
- NFC-F \ FeliCa (T3T)
- NFC-V \ ISO15693 (T5T)
- P2P \ ISO18092 (NFCIP1, Passive-Active P2P)
- ST25TB (ISO14443-2 Type B with Proprietary Protocol)

Internally, the RFAL is divided into three sub layers:

- RF HL - RF higher layer
- RF HAL- RF hardware abstraction layer
- RF AL - RF abstraction layer

Figure 1: RFAL block diagram



The modules in the RF HAL are chip-dependent, they implement the RF IC driver, configuration tables and specific instructions for the HW to perform the physical RF functions.

The interface for the caller is a shared RF header file which provides the same interface for upper layers (for all chips).

The RFAL can be broken down into two further sub layers:

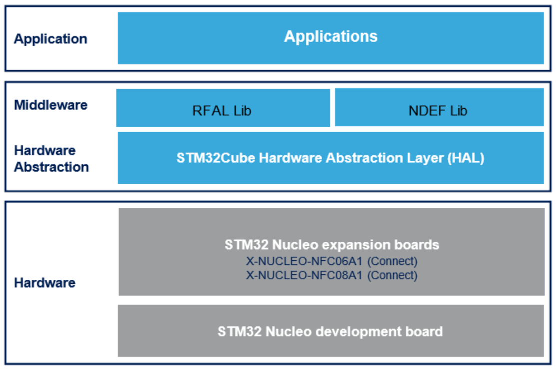
- Technologies: technology modules which implement all the specifics, framing, timings, etc.
- Protocols: protocol implementation including all the framing, timings, error handling, etc.

On top of these, the application layer uses RFAL functions like NFC Forum Activities (NFCC), EMVCo, DISCO/ NUCLEO demo, etc.

The RFAL NFC module provides an interface to perform the common activities as poller/listener device.

Access to the lowest functions of the ICs is granted by the RF module. The caller can make direct use of any of the RF technology or protocol layers without requiring any specific hardware configuration data.

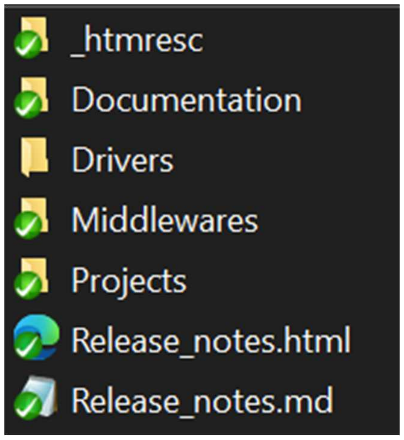
Figure 2: X-CUBE-NFC6 software architecture



Application
Applications
Middleware
RFAL Lib
NDEF Lib
Hardware Abstraction
STM32Cube Hardware Abstraction Layer (HAL)
Hardware
STM32 Nucleo expansion boards
X-NUCLEO-NFC06A1 (Connect)
X-NUCLEO-NFC08A1 (Connect)
STM32 Nucleo development board

2.3 Folder structure

Figure 3: X-CUBE-NFC6 package folders structure



The following folders are included in the software package:

- **Documentation**: this folder contains a compiled HTML file generated from the source code which details the software components and APIs.
- **Drivers**: this folder contains the HAL drivers, the board-specific drivers for each supported board or hardware platform, including the on-board components, and the CMSIS vendor-independent hardware abstraction layer for the Cortex-M processor series.
- **Middlewares**: this folder contains RFAL (RF abstraction layer). RFAL provides several functions required to perform RF/NFC communication.

  The RFAL groups the different RF ICs (ST25R3911/ST25R3916 and future ST25R391x devices) under a common and easy to use interface.

- **Projects:** this folder contains two sample application examples:
  - Tag Detect-Card emulation.
  - Read and Write of NDEF messages.

  They are provided for the NUCLEO-L476RG platform for three development environments (IAR Embedded Workbench for ARM, Keil Microcontroller Development Kit (MDK-ARM), and STM32CubeIDE.

2.4 APIs

Detailed technical information about the APIs available to the user can be found in a compiled CHM file located inside the “RFAL” folder of the software package where all the functions and parameters are fully described.

Detailed technical information about the NDEF APIs is available in the .chm file stored in the "doc" folder.

2.5 Sample application

A sample application using the X-NUCLEO-NFC06A1/X-NUCLEO-NFC08A1 expansion board with the NUCLEO-G0B1RE/NUCLEO-L476RG development board is provided in the “Projects” directory. Ready-to-build projects are available for multiple IDEs.

In this application, NFC tags of different types of mobile phones supporting P2P are detected by the ST25R3916/ST25R3916B high performance HF reader/NFC front end IC (for further details, refer to the CHM documentation file generated from the source code).

After system initialization and clock configuration, LED101, LED102, LED103, LED104, LED105 and LED106 blink for 3 times. Then LED106 glows to indicate the reader field has been activated.

When a tag is detected in the proximity, a LED is switched on as listed below.

Table 2. **LED Lit on tag detection.**

NFC tag type	LED lit on tag detection
NFC TYPE F	LED101/Type F
NFC TYPE B	LED102/Type B
NFC TYPE A	LED103/Type A
NFC TYPE V	LED104/Type V
NFC TYPE AP2P	LED105/Type AP2P

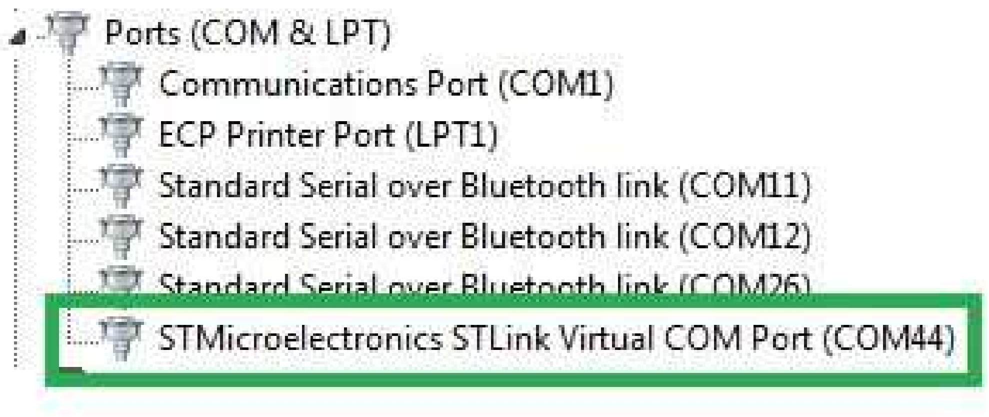
If a reader approaches the X-NUCLEO-NFC06A1/X-NUCLEO-NFC08A1 expansion board, the software enters card emulation mode and, depending on the command type sent, it switches NFC TYPE A and/or NFC TYPE F LED on.

By default, the X-NUCLEO-NFC06A1/X-NUCLEO-NFC08A1 does not write any data to the tag, but this possibility can be enabled by a pre-processor define in the file demo.h.

Card emulation and poller mode can also be enabled/disabled with the same procedure.

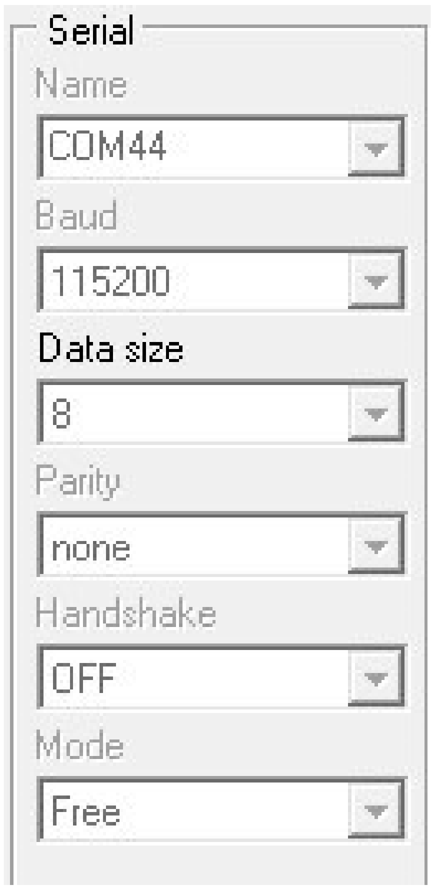
The ST virtual communication port interface is also included in the package. Once the board is powered on, the board is initialized and enumerated as STLink Virtual COM Port.

Figure 4: Virtual COM Port enumeration


Ports (COM & LPT)
Communications Port (COM1)
ECP Printer Port (LPT1)
Standard Serial over Bluetooth link (COM11)
Standard Serial over Bluetooth link (COM12)
Standard Serial over Bluetooth link (COM26)
STMicroelectronics STLink Virtual COM Port (COM44)

After checking the virtual COM port number, open a Windows terminal (HyperTerminal or similar) with the configuration shown below (enable option: Implicit CR on LF, if available).

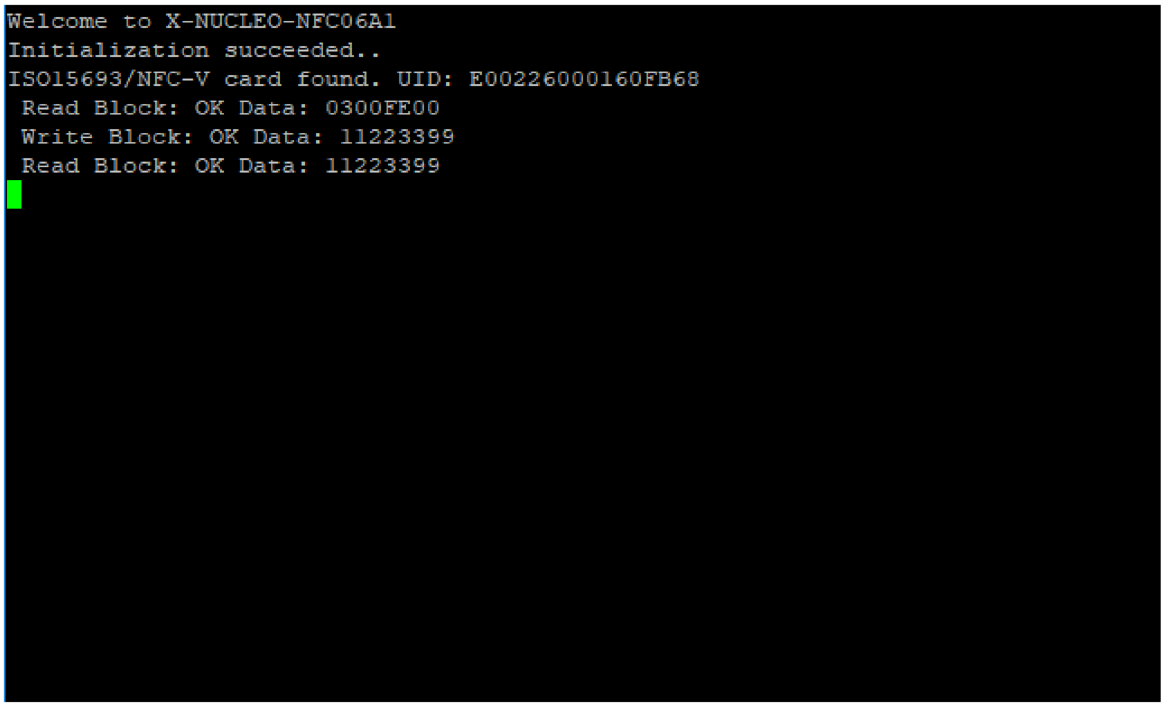
Figure 5: UART serial communication configuration


Serial

Name	COM44
Baud	115200
Data size	8
Parity	none
Handshake	OFF
Mode	Free

The terminal window returns several messages similar to those shown below to confirm successful connection.

Figure 6: X-NUCLEO-NFC06A1 expansion board successful initialization



```
Welcome to X-NUCLEO-NFC06A1
Initialization succeeded..
ISO15693/NFC-V card found. UID: E00226000160FB68
 Read Block: OK Data: 0300FE00
 Write Block: OK Data: 11223399
 Read Block: OK Data: 11223399
```

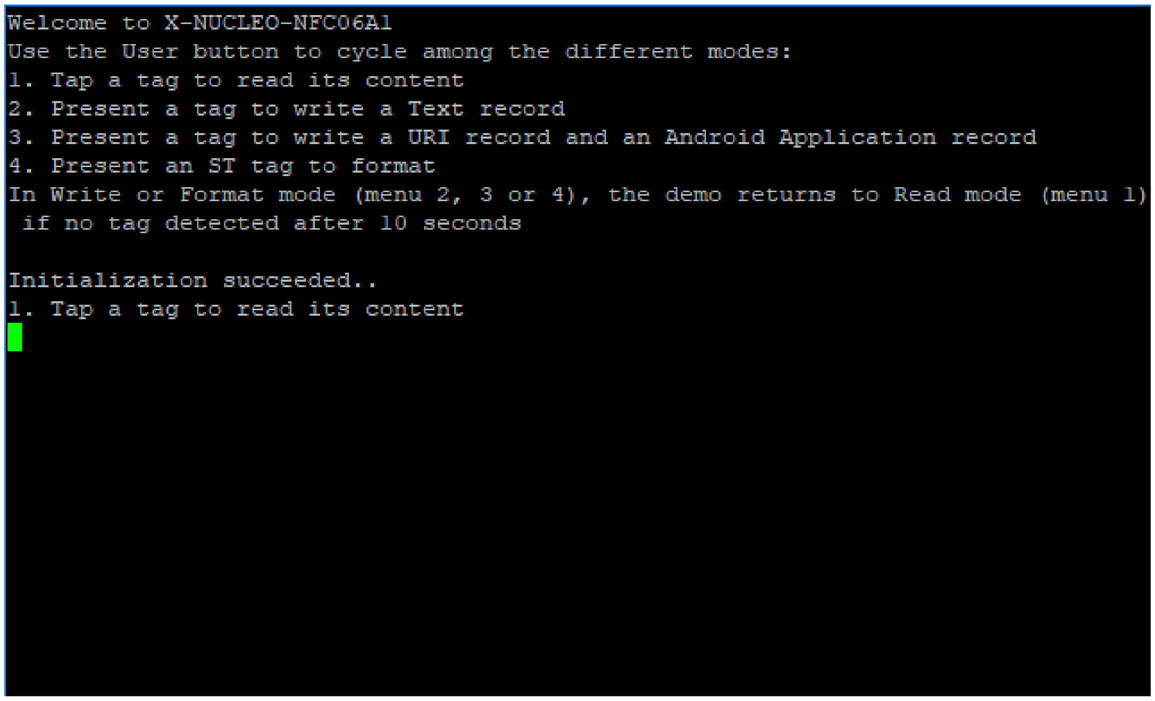
The second sample application is available by selecting the second project target called "STM32L476RG-Nucleo_PollingTagDetectNDEF". This application manages NDEF message on tags.

When the firmware starts, a menu is displayed on the console log.

The user button allows you to cycle through several options, including read NDEF content, write a text record, write an URI record and format tag for NDEF content.

After selecting the demo, tap a tag to see the demo running.

Figure 7: X-NUCLEO-NFC06A1 expansion board user button options



```
Welcome to X-NUCLEO-NFC06A1
Use the User button to cycle among the different modes:
1. Tap a tag to read its content
2. Present a tag to write a Text record
3. Present a tag to write a URI record and an Android Application record
4. Present an ST tag to format
In Write or Format mode (menu 2, 3 or 4), the demo returns to Read mode (menu 1)
 if no tag detected after 10 seconds

Initialization succeeded..
1. Tap a tag to read its content
```

3 System setup guide

3.1 Hardware description

3.1.1 STM32 Nucleo platform

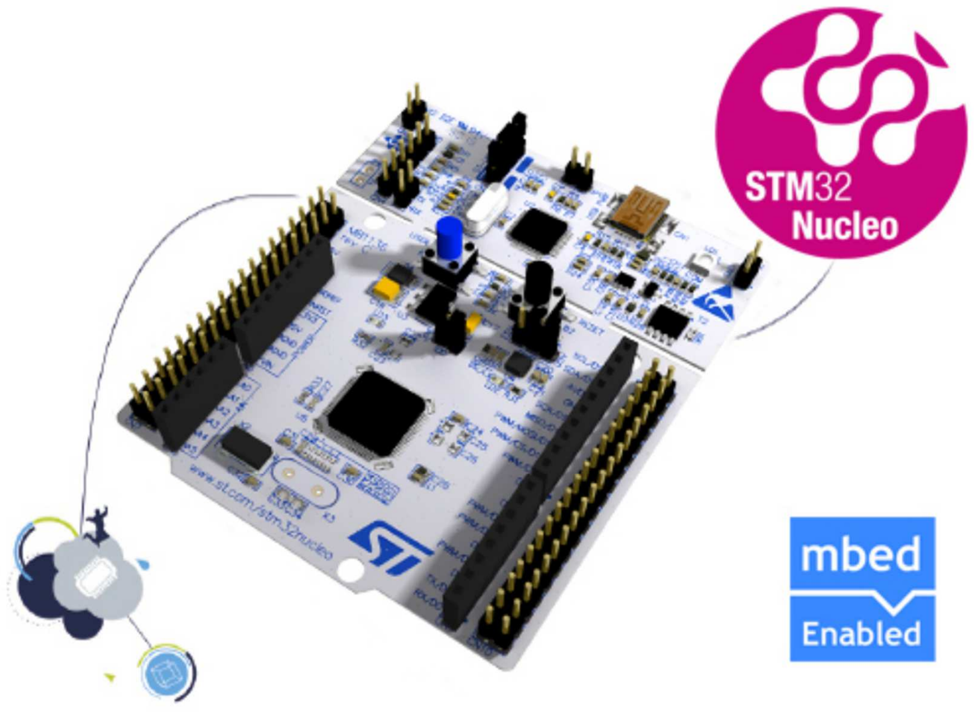
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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.

Figure 8: STM32 Nucleo board



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

3.1.2 X-NUCLEO-NFC06A1 expansion board

The X-NUCLEO-NFC06A1 NFC card reader expansion board is based on the ST25R3916 device.

The expansion board is configured to support ISO14443A/B, ISO15693, FeliCa™ and AP2P communication.

The ST25R3916 manages frame coding and decoding in reader mode for standard applications, such as NFC, proximity and vicinity HF RFID standards. It supports ISO/IEC 14443 Type A and B, ISO/IEC 15693 (single subcarrier only) and ISO/IEC 18092 communication protocols as well as the detection, reading and writing of NFC Forum Type 1, 2, 3, 4 and 5 tags.

The on-board low power capacitive sensor performs ultra-low power wake-up without switching the reader field on and traditional inductive wake-up to select amplitude or phase measurement.

The automatic antenna tuning (AAT) technology enables operation close to metallic parts and/or in changing environments.

Figure 9: X-NUCLEO-NFC06A1 expansion board

3.1.3 X-NUCLEO-NFC08A1 expansion board

The X-NUCLEO-NFC08A1 NFC card reader expansion board is based on the ST25R3916B device.

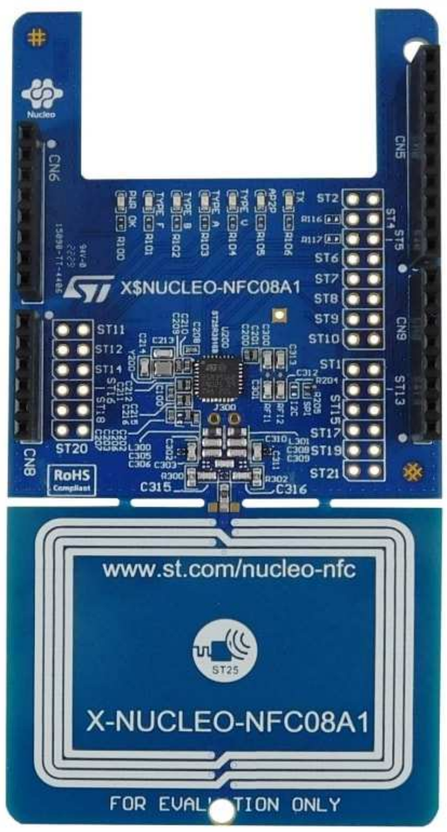
The expansion board is configured to support ISO14443A/B, ISO15693, FeliCa™ and AP2P communication.

The ST25R3916B manages frame coding and decoding in reader mode for standard applications, such as NFC, proximity and vicinity HF RFID standards. It supports ISO/IEC 14443 Type A and B, ISO/IEC 15693 (single subcarrier only) and ISO/IEC 18092 communication protocols as well as the detection, reading and writing of NFC Forum Type 1, 2, 3, 4 and 5 tags.

The on-board low power capacitive sensor performs ultra-low power wake-up without switching the reader field on and traditional inductive wake-up to select amplitude or phase measurement.

The automatic antenna tuning (AAT) technology enables operation close to metallic parts and/or in changing environments.

Figure 10: X-NUCLEO-NFC08A1 expansion board



3.2 Software description

The following software components are needed in order to setup the suitable development environment for creating applications for the STM32 Nucleo equipped with the NFC expansion board:

- X-CUBE-NFC6: an expansion for STM32Cube dedicated to NFC applications development. The X-CUBE-NFC6 firmware and related documentation is available on www.st.com.
- Development toolchain and Compiler. The STM32Cube expansion software supports the three following environments:
  - IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-LINK
  - Keil Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
  - STM32CubeIDE + ST-LINK

3.3 Hardware setup

The following hardware components are required:

- One STM32 Nucleo development platform (suggested order code: NUCLEO-L476RG)
- One ST25R3916 high performance HF reader/NFC front end IC expansion board (order code: X-NUCLEO-NFC06A1)
- One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC.

3.4 Software setup

3.4.1 Development toolchains and compilers

Select one of the integrated development environments (IDE) supported by the STM32Cube expansion software and read the system requirements and setup information provided by the IDE provider.

3.5 System setup guide

3.5.1 STM32 Nucleo and X-NUCLEO-NFC06A1 expansion board setup

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. You can download the ST-LINK/V2-1 USB driver at STSW-LINK009.

The X-NUCLEO-NFC06A1 expansion board is easily plugged onto the STM32 Nucleo development board through the Arduino™ UNO R3 extension connector.

It interfaces with the STM32 microcontroller on STM32 Nucleo board through the SPI transport layer.

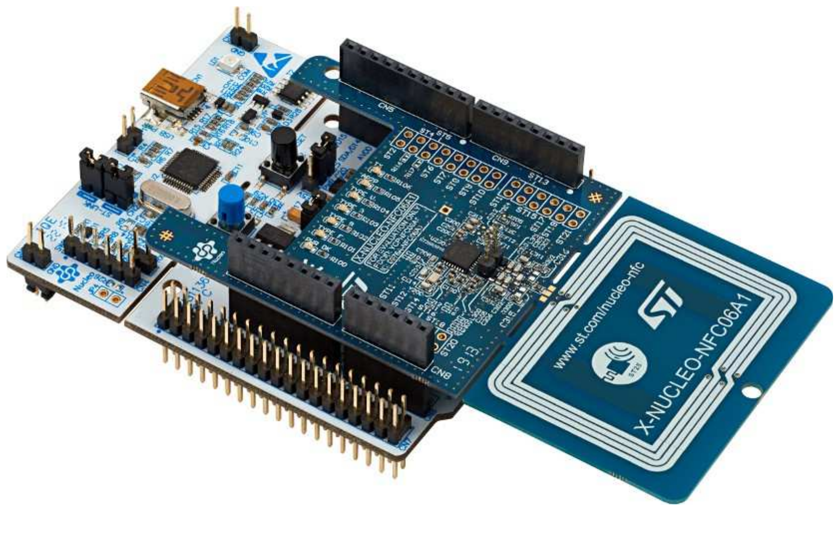
Default hardware configuration is set up for SPI communication.

I²C communication is also possible, but it requires the following hardware modifications:

- solder ST2 and ST4 jumpers.
- solder R116 and R117 pull-up resistors.
- remove the SPI solder bridge.
- put the I²C solder bridge.

You have to use the pre-processor compilation flag `RFAL_USE_I2C` and rename `USE_HAL_SPI_REGISTER_CALLBACKS` by `USE_HAL_I2C_REGISTER_CALLBACKS`, if needed, to activate the I²C driver compilation.

Figure 11: X-NUCLEO-NFC06A1 expansion board plus NUCLEO-L476RG development board

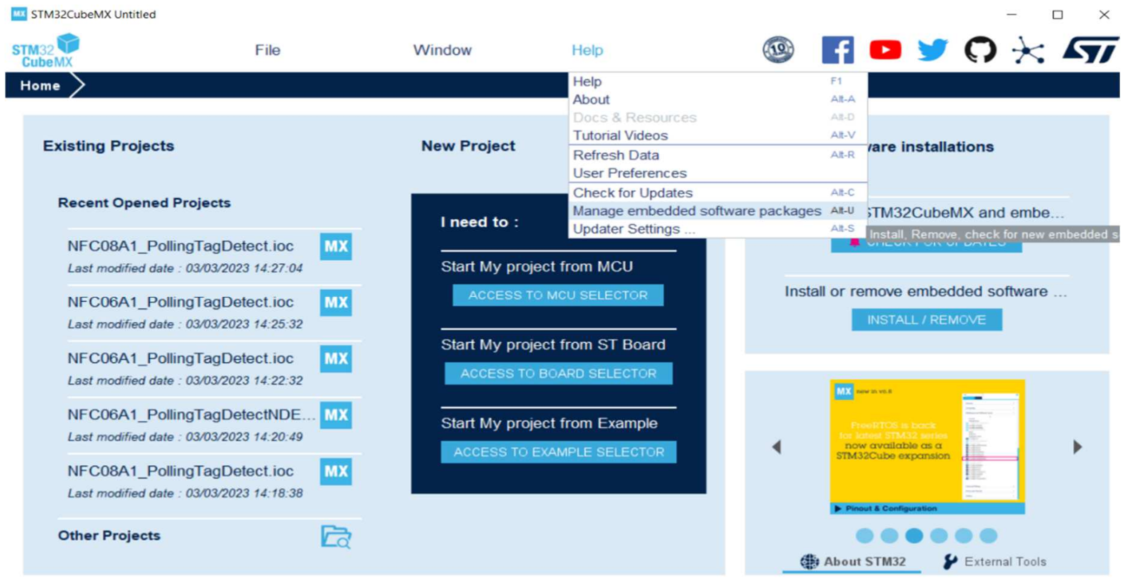


4 Installing the X-CUBE-NFC6 Pack in STM32CubeMX

After downloading (from www.st.com), installing and launching the STM32CubeMX, the X-CUBE-NFC6 pack can be installed in a few steps.

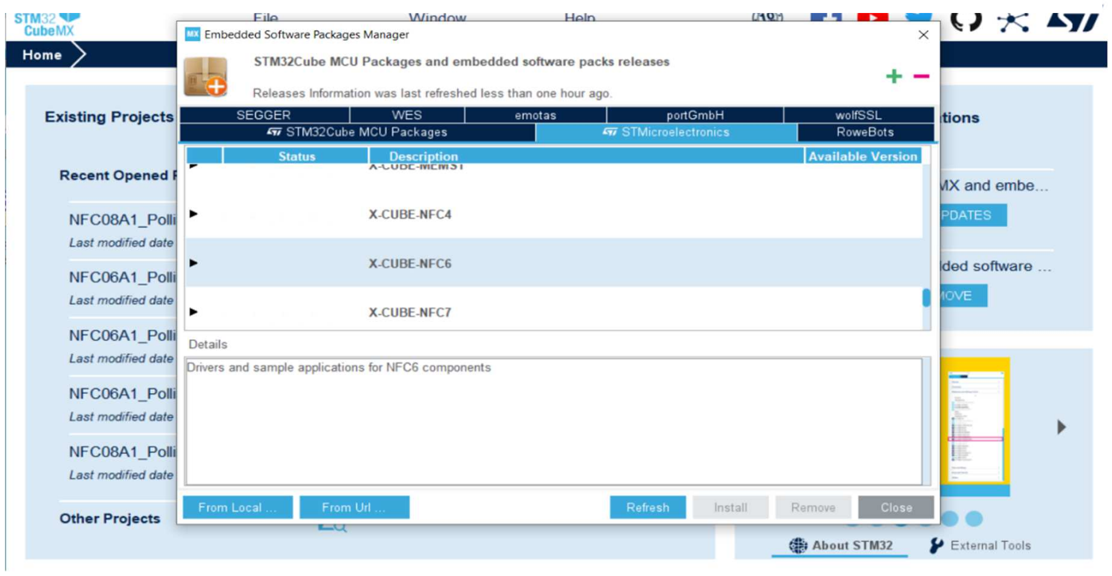
1. From the menu, select Help > Manage embedded software packages.

Figure 12: Managing embedded software packs in STM32CubeMX


STM32CubeMX Untitled
File
Window
Help
Home
Help F1
About Alt-A
Docs & Resources Alt-D
Tutorial Videos Alt-V
Refresh Data Alt-R
User Preferences
Check for Updates Alt-C
Manage embedded software packages Alt-U
Updater Settings ... Alt-S
Install, Remove, check for new embedded s
Existing Projects
New Project
Recent Opened Projects
NFC08A1_PollingTagDetect.ioc
Last modified date : 03/03/2023 14:27:04
NFC06A1_PollingTagDetect.ioc
Last modified date : 03/03/2023 14:25:32
NFC06A1_PollingTagDetect.ioc
Last modified date : 03/03/2023 14:22:32
NFC06A1_PollingTagDetectNDE...
Last modified date : 03/03/2023 14:20:49
NFC08A1_PollingTagDetect.ioc
Last modified date : 03/03/2023 14:18:38
Other Projects
I need to :
Start My project from MCU
ACCESS TO MCU SELECTOR
Start My project from ST Board
ACCESS TO BOARD SELECTOR
Start My project from Example
ACCESS TO EXAMPLE SELECTOR
STM32CubeMX and embe...
Install or remove embedded software ...
INSTALL / REMOVE
About STM32
External Tools

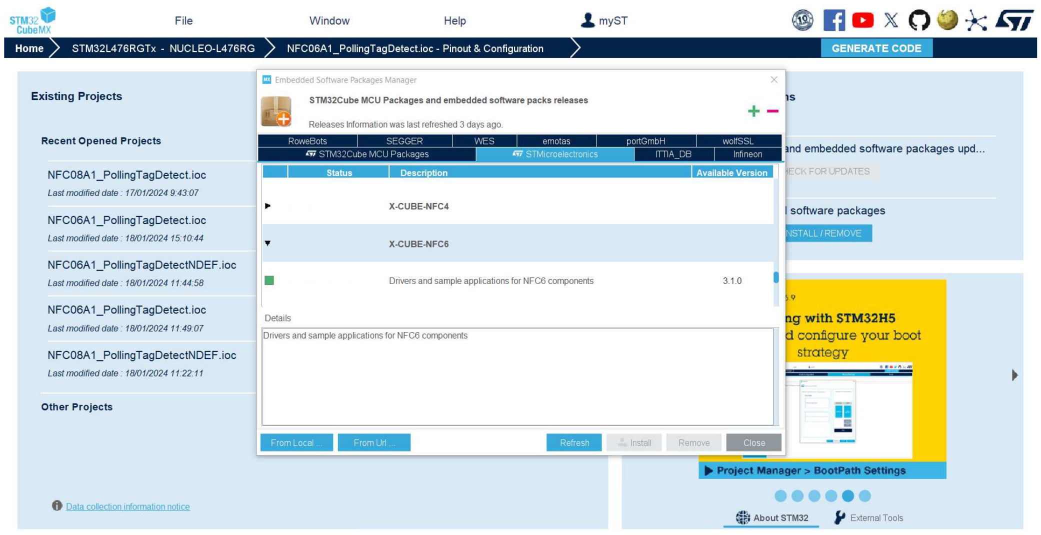
2. From the Embedded Software Packages Manager window, press the ‘Refresh’ button to get an updated list of the add-on packs. Go to the ‘STMicroelectronics’ tab to find the X-CUBE-NFC6 pack.

Figure 13: Installing the X-CUBE-NFC6 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 14: The X-CUBE-NFC6 pack in STM32CubeMX


STM32 CubeMX
File
Window
Help
myST
Home
STM32L476RGTx - NUCLEO-L476RG
NFC06A1_PollingTagDetect.ioc - Pinout & Configuration
GENERATE CODE

Existing Projects

Recent Opened Projects

NFC08A1_PollingTagDetect.ioc
Last modified date : 17/01/2024 9:43:07

NFC06A1_PollingTagDetect.ioc
Last modified date : 18/01/2024 15:10:44

NFC06A1_PollingTagDetectNDEF.ioc
Last modified date : 18/01/2024 11:44:58

NFC06A1_PollingTagDetect.ioc
Last modified date : 18/01/2024 11:49:07

NFC08A1_PollingTagDetectNDEF.ioc
Last modified date : 18/01/2024 11:22:11

Other Projects

Data collection information notice

Embedded Software Packages Manager

STM32Cube MCU Packages and embedded software packs releases

Releases Information was last refreshed 3 days ago.

RoweBots | SEGGER | WES | emotas | portGmbH | wolfSSL
STM32Cube MCU Packages | STMicroelectronics | ITTIA_DB | Infineon

Status	Description	Available Version
▸	X-CUBE-NFC4	
▾	X-CUBE-NFC6	
🟩	Drivers and sample applications for NFC6 components	3.1.0

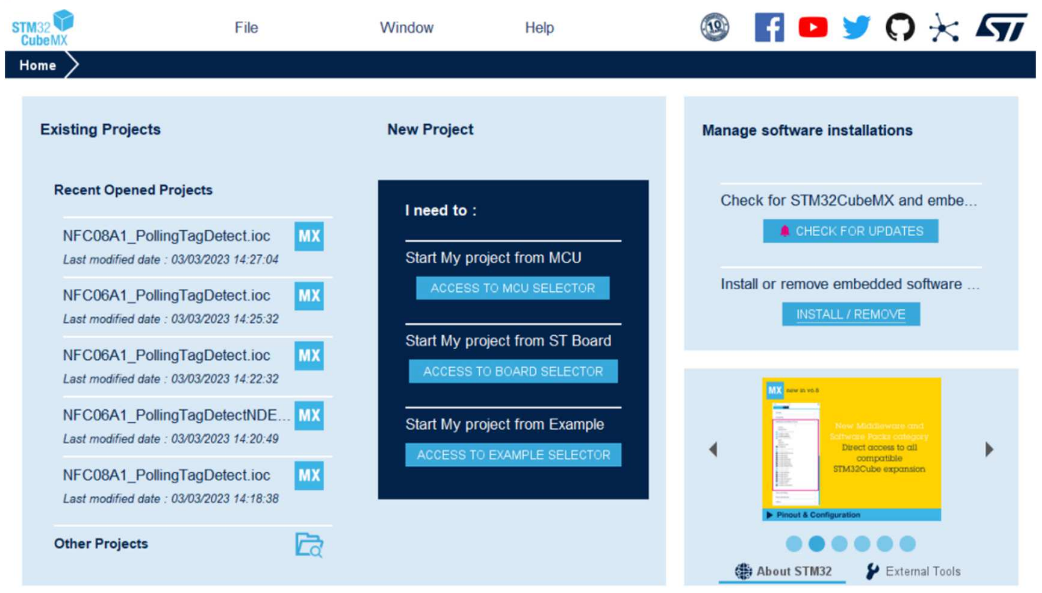
Details

Drivers and sample applications for NFC6 components

5 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 15: The X-CUBE-NFC6 pack in STM32CubeMX



STM32 CubeMX
File
Window
Help
Home
Existing Projects
New Project
Manage software installations
I need to :
Start My project from MCU
ACCESS TO MCU SELECTOR
Start My project from ST Board
ACCESS TO BOARD SELECTOR
Start My project from Example
ACCESS TO EXAMPLE SELECTOR
Check for STM32CubeMX and embe...
CHECK FOR UPDATES
Install or remove embedded software ...
INSTALL / REMOVE
Other Projects
About STM32
External Tools

Recent Opened Projects

NFC08A1_PollingTagDetect.ioc
Last modified date : 03/03/2023 14:27:04

NFC06A1_PollingTagDetect.ioc
Last modified date : 03/03/2023 14:25:32

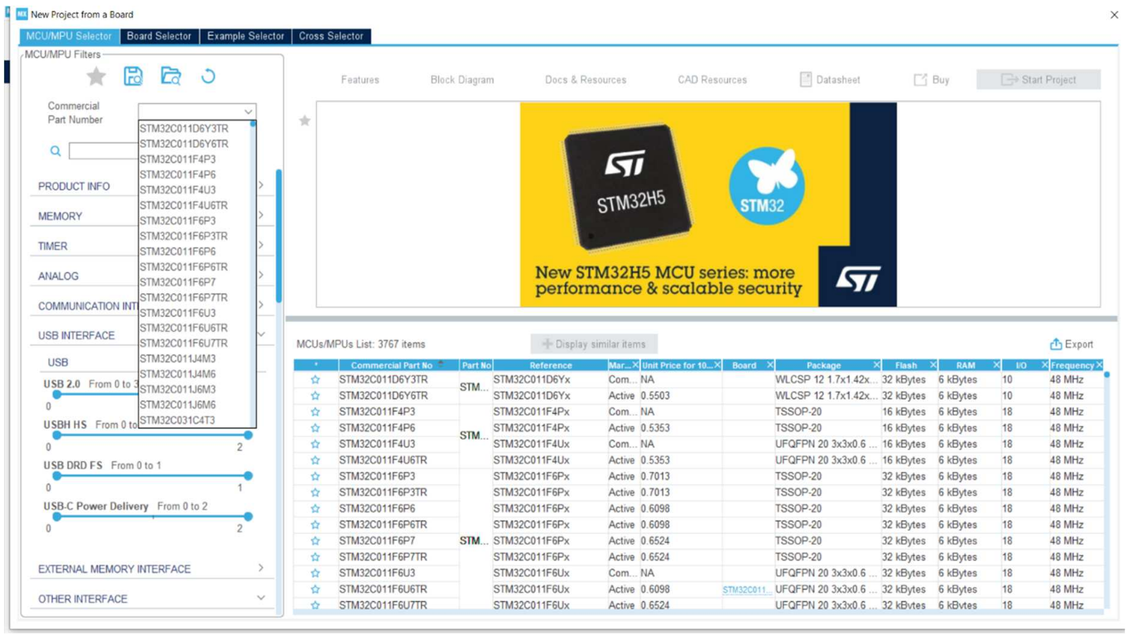
NFC06A1_PollingTagDetect.ioc
Last modified date : 03/03/2023 14:22:32

NFC06A1_PollingTagDetectNDE...
Last modified date : 03/03/2023 14:20:49

NFC08A1_PollingTagDetect.ioc
Last modified date : 03/03/2023 14:18:38

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

Figure 16: STM32CubeMX MCU/Board Selector windows



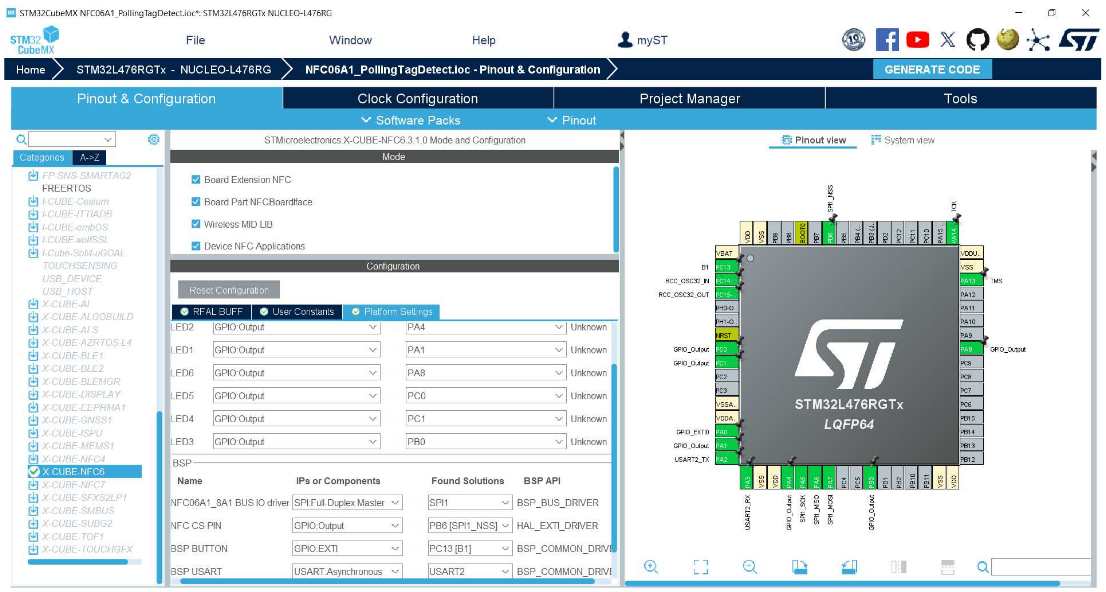
STM32C011D6Y3TR
STM32C011D6Y6TR
STM32C011F4P3
STM32C011F4P6
STM32C011F4U3
STM32C011F4U6TR
STM32C011F6P3
STM32C011F6P3TR
STM32C011F6P6
STM32C011F6P6TR
STM32C011F6P7
STM32C011F6P7TR
STM32C011F6U3
STM32C011F6U6TR
STM32C011F6U7TR
STM32C011J4M3
STM32C011J4M6
STM32C011J6M3
STM32C011J6M6
STM32C031C4T3

MCUs/MPUs List: 3767 items

*	Commercial Part No	Part No	Reference	Mar...	Unit Price for 10...	Board	Package	Flash	RAM	I/O	Frequency
☆	STM32C011D6Y3TR	STM...	STM32C011D6Yx	Com...	NA		WLCSP 12 1.7x1.42x...	32 kBytes	6 kBytes	10	48 MHz
☆	STM32C011D6Y6TR		STM32C011D6Yx	Active	0.5503		WLCSP 12 1.7x1.42x...	32 kBytes	6 kBytes	10	48 MHz
☆	STM32C011F4P3		STM32C011F4Px	Com...	NA		TSSOP-20	16 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F4P6	STM...	STM32C011F4Px	Active	0.5353		TSSOP-20	16 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F4U3		STM32C011F4Ux	Com...	NA		UFQFPN 20 3x3x0.6 ...	16 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F4U6TR		STM32C011F4Ux	Active	0.5353		UFQFPN 20 3x3x0.6 ...	16 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F6P3		STM32C011F6Px	Active	0.7013		TSSOP-20	32 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F6P3TR		STM32C011F6Px	Active	0.7013		TSSOP-20	32 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F6P6		STM32C011F6Px	Active	0.6098		TSSOP-20	32 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F6P6TR		STM32C011F6Px	Active	0.6098		TSSOP-20	32 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F6P7	STM...	STM32C011F6Px	Active	0.6524		TSSOP-20	32 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F6P7TR		STM32C011F6Px	Active	0.6524		TSSOP-20	32 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F6U3		STM32C011F6Ux	Com...	NA		UFQFPN 20 3x3x0.6 ...	32 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F6U6TR		STM32C011F6Ux	Active	0.6098	STM32C011...	UFQFPN 20 3x3x0.6 ...	32 kBytes	6 kBytes	18	48 MHz
☆	STM32C011F6U7TR		STM32C011F6Ux	Active	0.6524		UFQFPN 20 3x3x0.6 ...	32 kBytes	6 kBytes	18	48 MHz

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 17: STM32CubeMX Pinout & Configuration window



FP-SNS-SMARTAG2
FREERTOS
I-CUBE-Cesium
I-CUBE-ITTIADB
I-CUBE-embOS
I-CUBE-wolfSSL
I-Cube-SoM-uGOAL
TOUCHSENSING
USB_DEVICE
USB_HOST
X-CUBE-AI
X-CUBE-ALGOBUILD
X-CUBE-ALS
X-CUBE-AZRTOS-L4
X-CUBE-BLE1
X-CUBE-BLE2
X-CUBE-BLEMGR
X-CUBE-DISPLAY
X-CUBE-EEPRMA1
X-CUBE-GNSS1
X-CUBE-ISPU
X-CUBE-MEMS1
X-CUBE-NFC4
X-CUBE-NFC6
X-CUBE-NFC7
X-CUBE-SFXS2LP1
X-CUBE-SMBUS
X-CUBE-SUBG2
X-CUBE-TOF1
X-CUBE-TOUCHGFX

STMicroelectronics.X-CUBE-NFC6.3.1.0 Mode and Configuration

Mode

- ☑ Board Extension NFC
- ☑ Board Part NFCBoardIface
- ☑ Wireless MID LIB
- ☑ Device NFC Applications

Configuration

Reset Configuration

RFAL BUFF | User Constants | Platform Settings

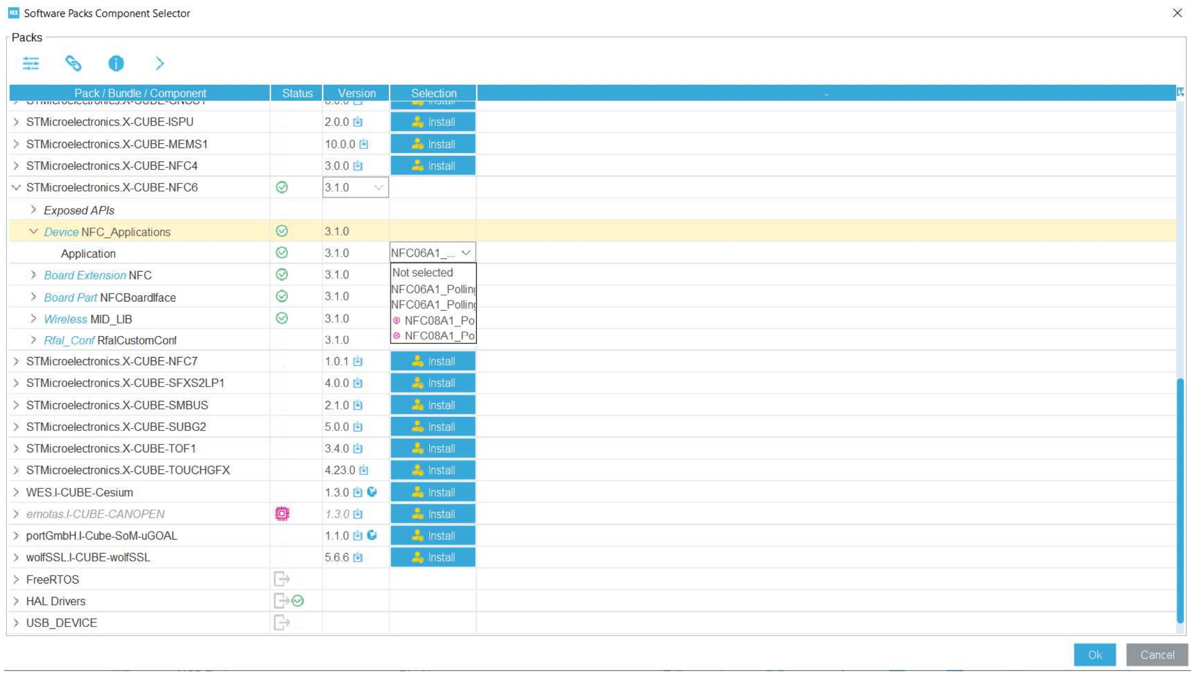
LED2	GPIO:Output	PA4	Unknown
LED1	GPIO:Output	PA1	Unknown
LED6	GPIO:Output	PA8	Unknown
LED5	GPIO:Output	PC0	Unknown
LED4	GPIO:Output	PC1	Unknown
LED3	GPIO:Output	PB0	Unknown

BSP

Name	IPs or Components	Found Solutions	BSP API
NFC06A1_8A1 BUS IO driver	SPI:Full-Duplex Master	SPI1	BSP_BUS_DRIVER
NFC CS PIN	GPIO:Output	PB6 [SPI1_NSS]	HAL_EXTI_DRIVER
BSP BUTTON	GPIO:EXTI	PC13 [B1]	BSP_COMMON_DRIVE
BSP USART	USART:Asynchronous	USART2	BSP_COMMON_DRIVE

To add the X-CUBE-NFC6 additional software to the project, the "Additional Software's" 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 18: STM32CubeMX Additional Software Components selection window



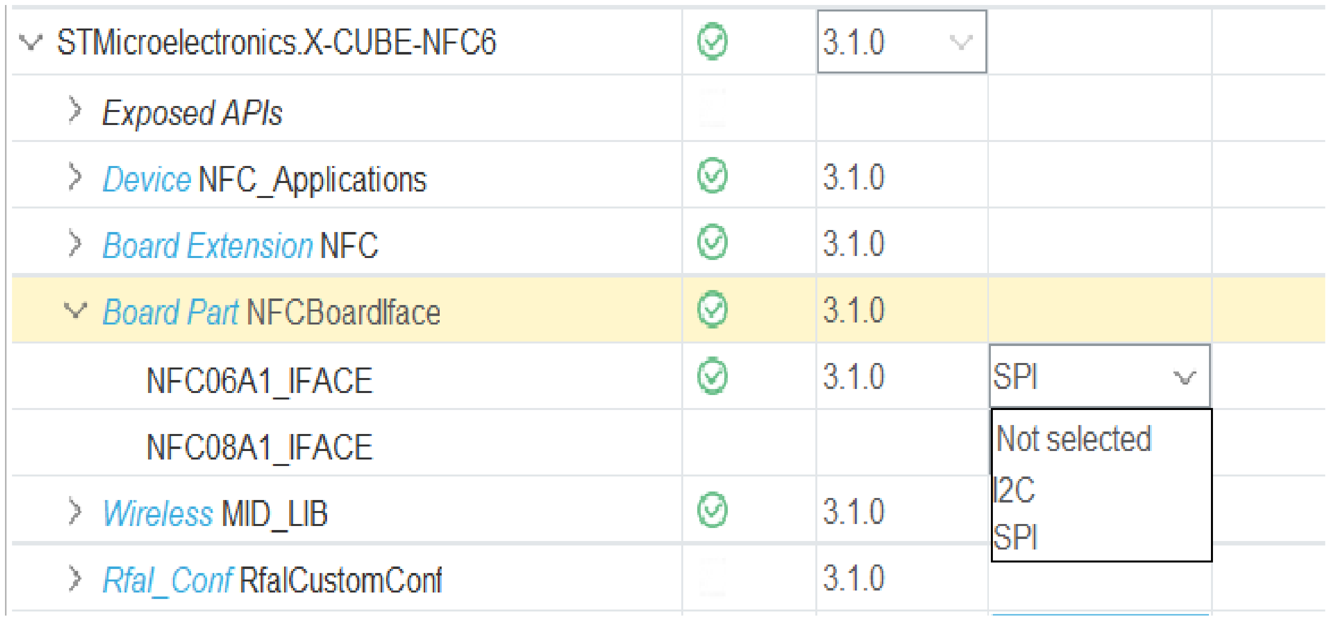
Pack / Bundle / Component	Status	Version	Selection
STMicroelectronics.X-CUBE-ISPU		2.0.0	Install
STMicroelectronics.X-CUBE-MEMS1		10.0.0	Install
STMicroelectronics.X-CUBE-NFC4		3.0.0	Install
STMicroelectronics.X-CUBE-NFC6	✓	3.1.0	
Exposed APIs			
Device NFC_Applications	✓	3.1.0	
Application	✓	3.1.0	NFC06A1_...
Board Extension NFC	✓	3.1.0	Not selected
Board Part NFCBoardIface	✓	3.1.0	NFC06A1_Pollin / NFC06A1_Pollin
Wireless MID_LIB	✓	3.1.0	NFC08A1_Po
Rfal_Conf RfalCustomConf		3.1.0	NFC08A1_Po
STMicroelectronics.X-CUBE-NFC7		1.0.1	Install
STMicroelectronics.X-CUBE-SFXS2LP1		4.0.0	Install
STMicroelectronics.X-CUBE-SMBUS		2.1.0	Install
STMicroelectronics.X-CUBE-SUBG2		5.0.0	Install
STMicroelectronics.X-CUBE-TOF1		3.4.0	Install
STMicroelectronics.X-CUBE-TOUCHGFX		4.23.0	Install
WES.I-CUBE-Cesium		1.3.0	Install
emotas.I-CUBE-CANOPEN		*1.3.0*	Install
portGmbH.I-Cube-SoM-uGOAL		1.1.0	Install
wolfSSL.I-CUBE-wolfSSL		5.6.6	Install
FreeRTOS			
HAL Drivers	✓		
USB_DEVICE			

6 Use of NFC component without NDEF Library for X-NUCLEO-NFC06A1

This section outlines how to configure STM32CubeMX with X-NUCLEO-NFC06A1 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-NFC6 additional software to the project,
the “Additional Software’s” button must be clicked. From the “Additional Software Components
selection” window, the user must select the example from the “Device” class and “Board Extension.”
class as shown in the figure below.
By default, X-NUCLEO-NFC06A1/ X-NUCLEO-NFC08A1 is configured in SPI.

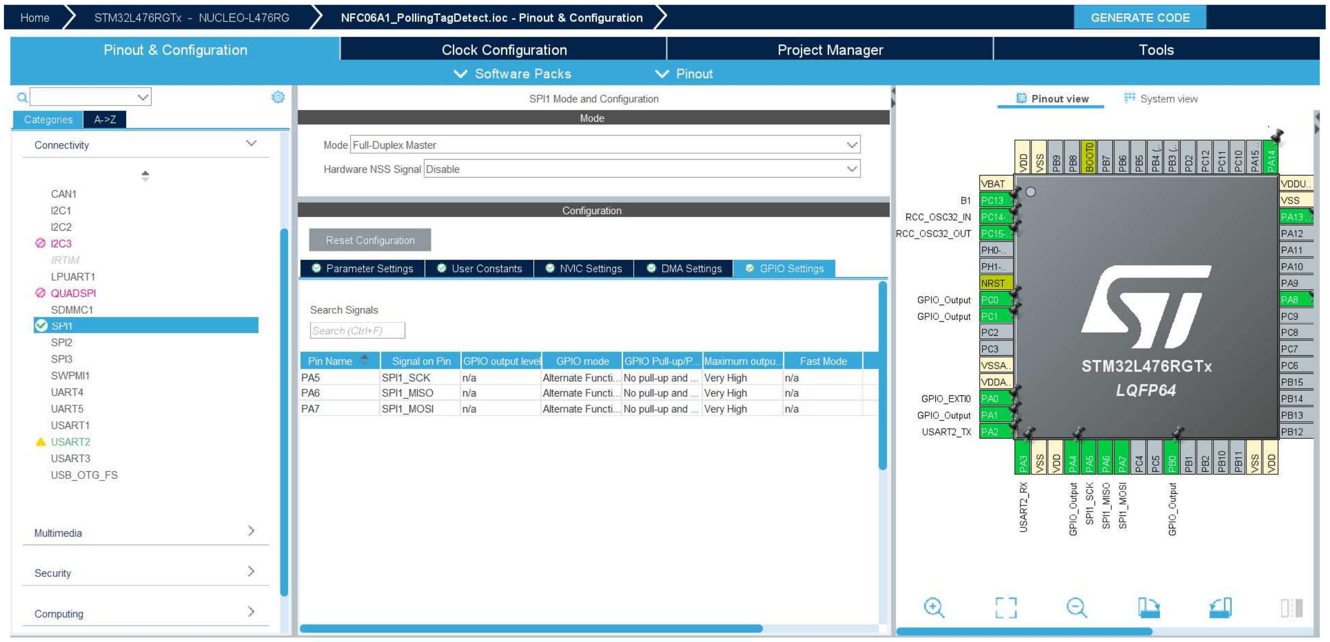
6.1 For SPI Communication the following pin configuration is required.

Figure 19: STM32CubeMX Additional Software Components selection window



STMicroelectronics.X-CUBE-NFC6	✓	3.1.0	
Exposed APIs			
Device NFC_Applications	✓	3.1.0	
Board Extension NFC	✓	3.1.0	
Board Part NFCBoardIface	✓	3.1.0	
NFC06A1_IFACE	✓	3.1.0	SPI
NFC08A1_IFACE			Not selected / I2C / SPI
Wireless MID_LIB	✓	3.1.0	
Rfal_Conf RfalCustomConf		3.1.0	

Figure 20: GPIO Pin Settings for SPI.



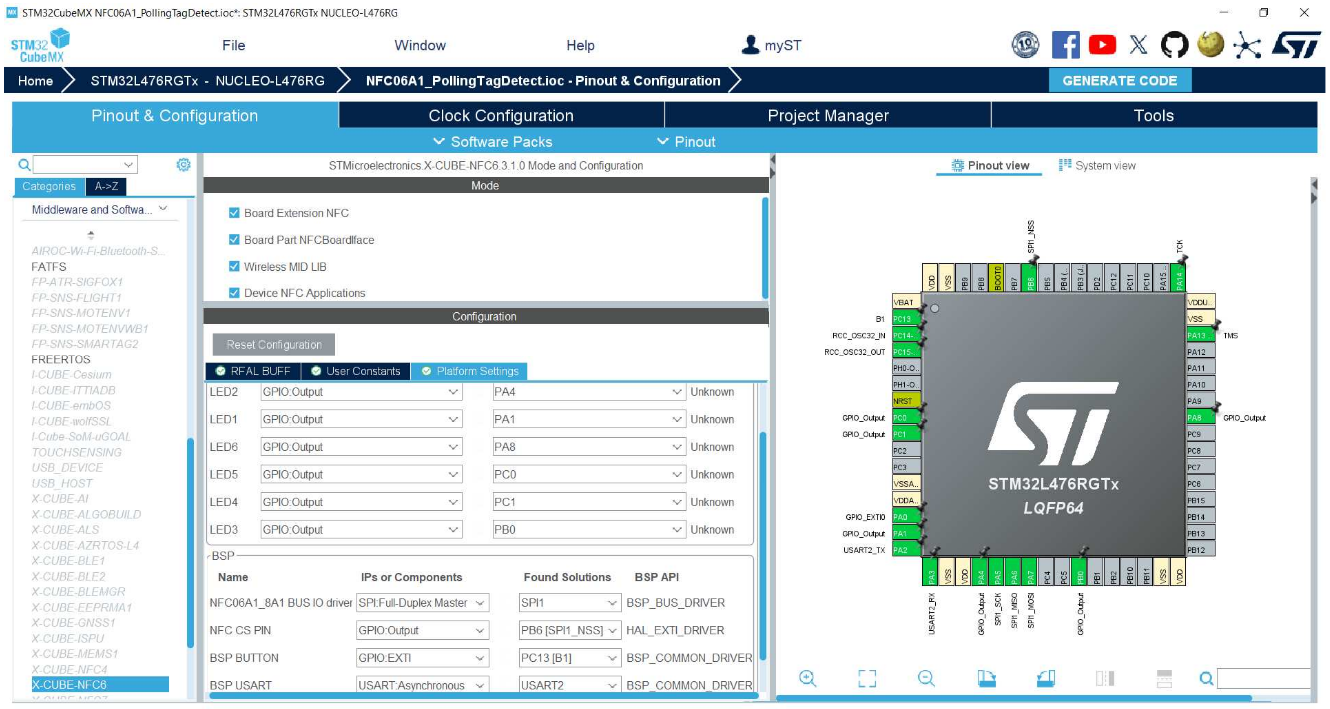
SPI1 Mode and Configuration

Mode: Full-Duplex Master

Hardware NSS Signal: Disable

Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/P...	Maximum outpu...	Fast Mode
PA5	SPI1_SCK	n/a	Alternate Functi...	No pull-up and ...	Very High	n/a
PA6	SPI1_MISO	n/a	Alternate Functi...	No pull-up and ...	Very High	n/a
PA7	SPI1_MOSI	n/a	Alternate Functi...	No pull-up and ...	Very High	n/a

Figure 21: Pin Connection of SPI enabling and selecting in software packs.



STMicroelectronics.X-CUBE-NFC6.3.1.0 Mode and Configuration

- [✓] Board Extension NFC
- [✓] Board Part NFCBoardIface
- [✓] Wireless MID LIB
- [✓] Device NFC Applications

LED2	GPIO:Output	PA4	Unknown
LED1	GPIO:Output	PA1	Unknown
LED6	GPIO:Output	PA8	Unknown
LED5	GPIO:Output	PC0	Unknown
LED4	GPIO:Output	PC1	Unknown
LED3	GPIO:Output	PB0	Unknown

BSP

Name	IPs or Components	Found Solutions	BSP API
NFC06A1_8A1 BUS IO driver	SPI:Full-Duplex Master	SPI1	BSP_BUS_DRIVER
NFC CS PIN	GPIO:Output	PB6 [SPI1_NSS]	HAL_EXTI_DRIVER
BSP BUTTON	GPIO:EXTI	PC13 [B1]	BSP_COMMON_DRIVER
BSP USART	USART:Asynchronous	USART2	BSP_COMMON_DRIVER

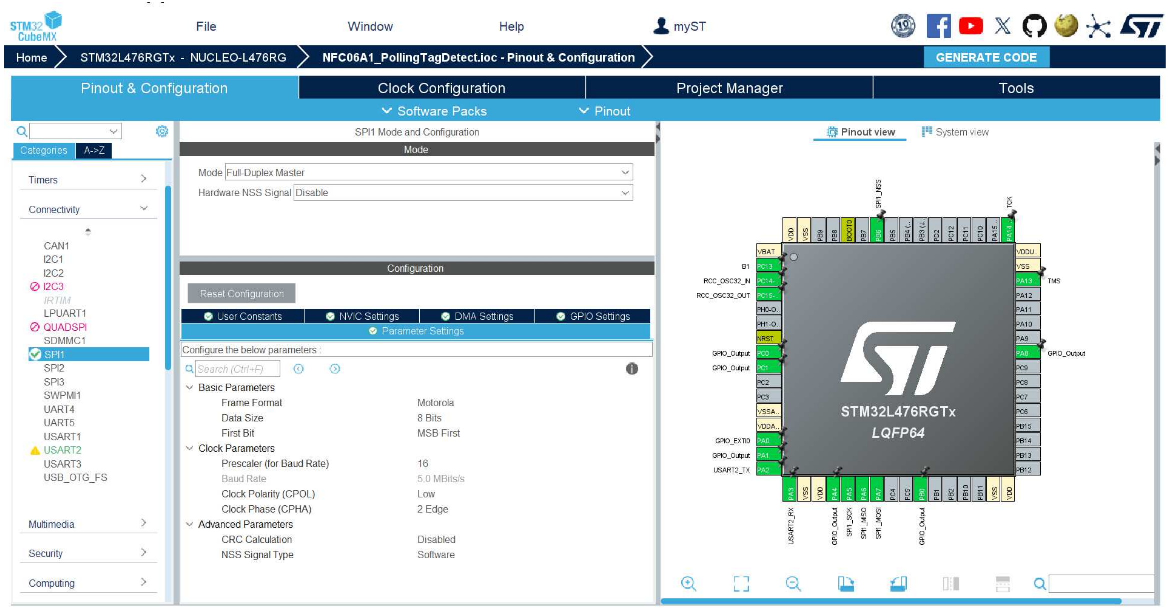
Table 3. **Detailed Pin Description for SPI of NUCLEO-L476RG.**

Name	BSP_API	Supported IPs	Nucleo-64
NFC06A1 BUS IO Driver	BSP_BUS_DRIVER	SPI	SPI1
NFC INT PIN	HAL_EXTI_DRIVER	GPIO EXTI	PA0
NFC06A1 LED Pins LED2	GPIO Output Pin	Output	PA4
NFC06A1 LED Pins LED1	GPIO Output Pin	Output	PA1
NFC06A1 LED Pins LED6	GPIO Output Pin	Output	PA8
NFC06A1 LED Pins LED5	GPIO Output Pin	Output	PC0
NFC06A1 LED Pins LED4	GPIO Output Pin	Output	PC1
NFC06A1 LED Pins LED3	GPIO Output Pin	Output	PB0
BSP BUTTON	BSP_COMMON_DRIVER	EXTI push Button	PC13[B1]
BSP USART	BSP_COMMON_DRIVER	USART	USART2
NFC CS Pin	HAL_EXTI_DRIVER	GPIO Output Pin	PB6 SPI1_NSS

Table 4. **Detailed Pin Description for SPI of NUCLEO-G0B1RE.**

Name	BSP_API	Supported IPs	Nucleo-64
NFC06A1 BUS IO Driver	BSP_BUS_DRIVER	SPI	SPI1
NFC INT PIN	HAL_EXTI_DRIVER	GPIO EXTI	PA0
NFC06A1 LED Pins LED2	GPIO Output Pin	Output	PA4
NFC06A1 LED Pins LED1	GPIO Output Pin	Output	PA1
NFC06A1 LED Pins LED6	GPIO Output Pin	Output	PA8
NFC06A1 LED Pins LED5	GPIO Output Pin	Output	PB11
NFC06A1 LED Pins LED4	GPIO Output Pin	Output	PB12
NFC06A1 LED Pins LED3	GPIO Output Pin	Output	PB1
BSP BUTTON	BSP_COMMON_DRIVER	EXTI push Button	PC13[B1]
BSP USART	BSP_COMMON_DRIVER	USART	USART2
NFC CS Pin	HAL_EXTI_DRIVER	GPIO Output Pin	PB0

Figure 22: Parameter Configuration of SPI



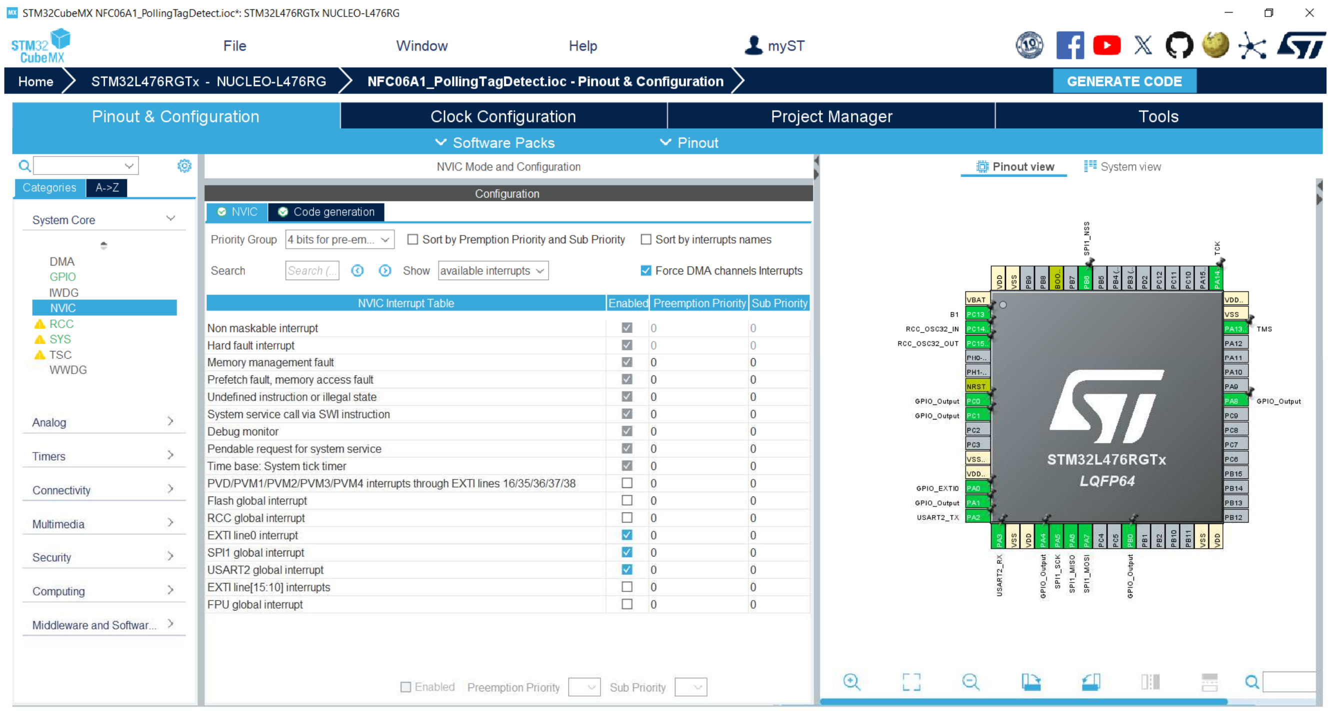
Mode: Full-Duplex Master
Hardware NSS Signal: Disable

Configure the below parameters :

Basic Parameters	
Frame Format	Motorola
Data Size	8 Bits
First Bit	MSB First
Clock Parameters	
Prescaler (for Baud Rate)	16
Baud Rate	5.0 MBits/s
Clock Polarity (CPOL)	Low
Clock Phase (CPHA)	2 Edge
Advanced Parameters	
CRC Calculation	Disabled
NSS Signal Type	Software

From the Configuration & Pinout tab, click on “System Core” category and then on NVIC item to enable the EXTI0 line interrupt and set it as “External Interrupt Mode with Rising edge trigger detection”.

Figure 23: STM32CubeMX NVIC Configuration of SPI.



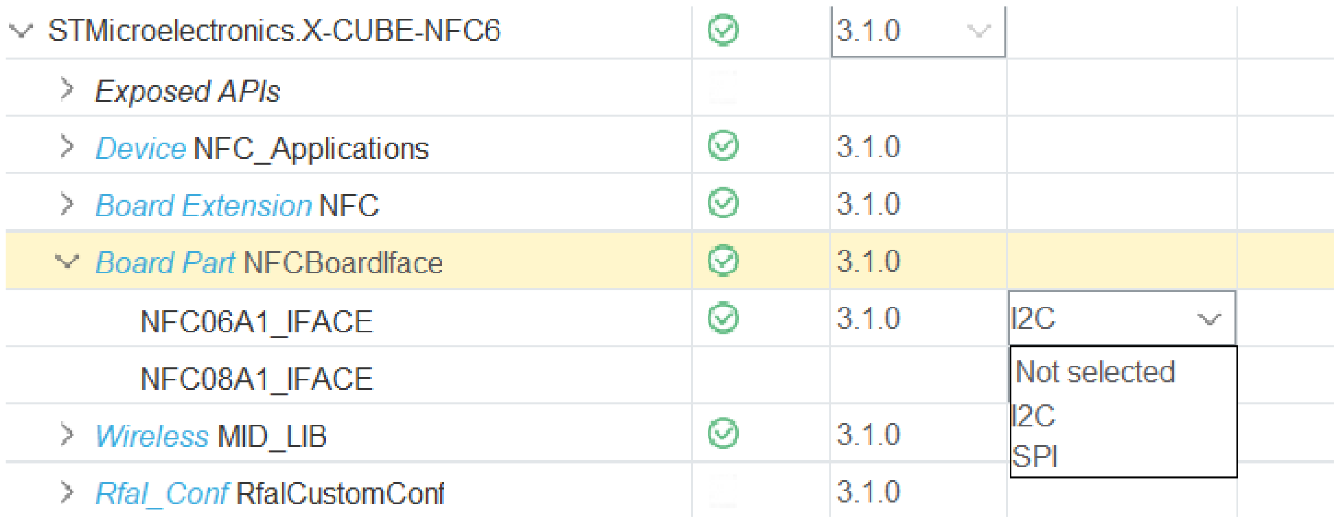
Priority Group: 4 bits for pre-em... ☐ Sort by Premption Priority and Sub Priority ☐ Sort by interrupts names
Search: Search (... Show: available interrupts ☑ Force DMA channels Interrupts

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	✓	0	0
Hard fault interrupt	✓	0	0
Memory management fault	✓	0	0
Prefetch fault, memory access fault	✓	0	0
Undefined instruction or illegal state	✓	0	0
System service call via SWI instruction	✓	0	0
Debug monitor	✓	0	0
Pendable request for system service	✓	0	0
Time base: System tick timer	✓	0	0
PVD/PVM1/PVM2/PVM3/PVM4 interrupts through EXTI lines 16/35/36/37/38	☐	0	0
Flash global interrupt	☐	0	0
RCC global interrupt	☐	0	0
EXTI line0 interrupt	✓	0	0
SPI1 global interrupt	✓	0	0
USART2 global interrupt	✓	0	0
EXTI line[15:10] interrupts	☐	0	0
FPU global interrupt	☐	0	0

☐ Enabled Preemption Priority Sub Priority

6.2 For I2C Communication the following pin configuration is required.

Figure 24: STM32CubeMX Additional Software Components selection window

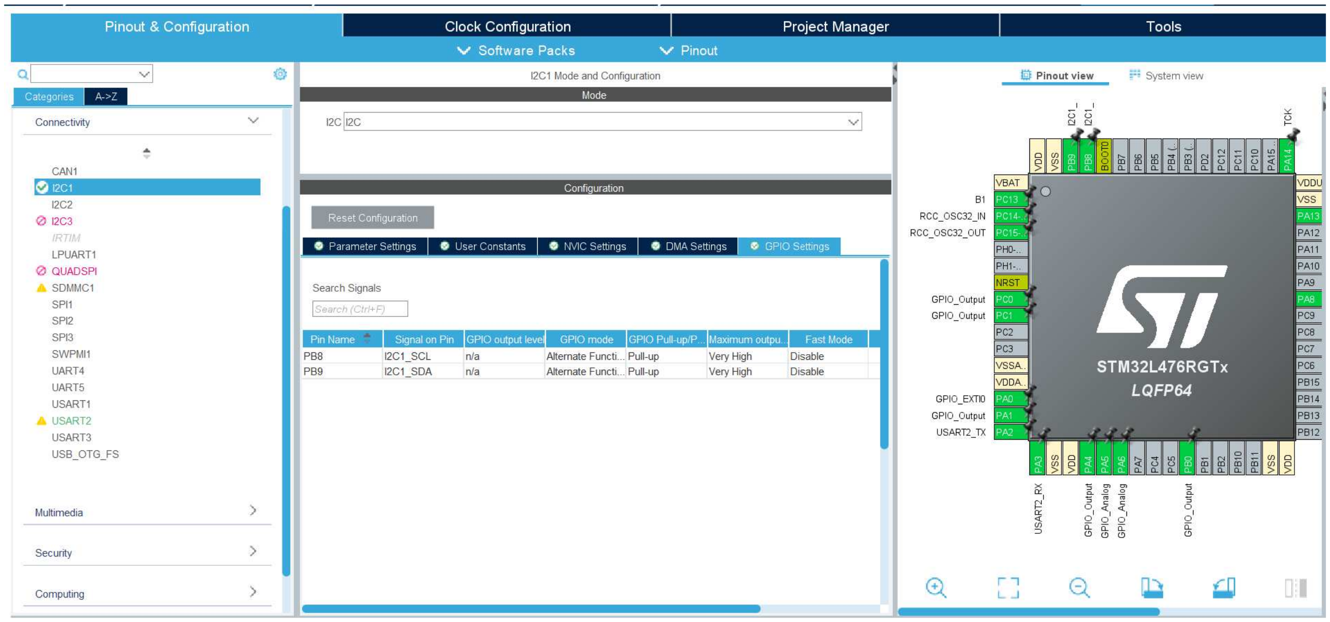


∨ STMicroelectronics.X-CUBE-NFC6	✓	3.1.0	
> *Exposed APIs*			
> *Device* NFC_Applications	✓	3.1.0	
> *Board Extension* NFC	✓	3.1.0	
∨ *Board Part* NFCBoardIface	✓	3.1.0	
NFC06A1_IFACE	✓	3.1.0	I2C
NFC08A1_IFACE			Not selected / I2C / SPI
> *Wireless* MID_LIB	✓	3.1.0	
> *Rfal_Conf* RfalCustomConf		3.1.0	

Hardware Side:
Solder two 0-ohm resistors to ST2 & ST4.
Solder the R116 & R117 pull up resistors.
Remove the SPI solder bridge (R205) and connect the I2C solder bridge (R204) on the board.
Connect jumper wires from ST2 to ST5 and ST3 to ST6.

Software Side:
Enable the I2C1 inside the Cube-MX and disable the SPI1 to select the Communication mode as I2C, and the pins of I2C1 are PB8 as I2C1_SCL & PB9 as I2C1_SDA and set both in Pull Up mode.
Enable the I2C event interrupt and set its priority as 1 and priority 2 to EXTI line0 interrupt in System Core -> NVIC.
Configure PA5 and PA6 in GPIO_Analog mode.

Figure 25: GPIO Pin Configuration for I2C.



I2C1 Mode and Configuration

Mode

I2C: I2C

Configuration

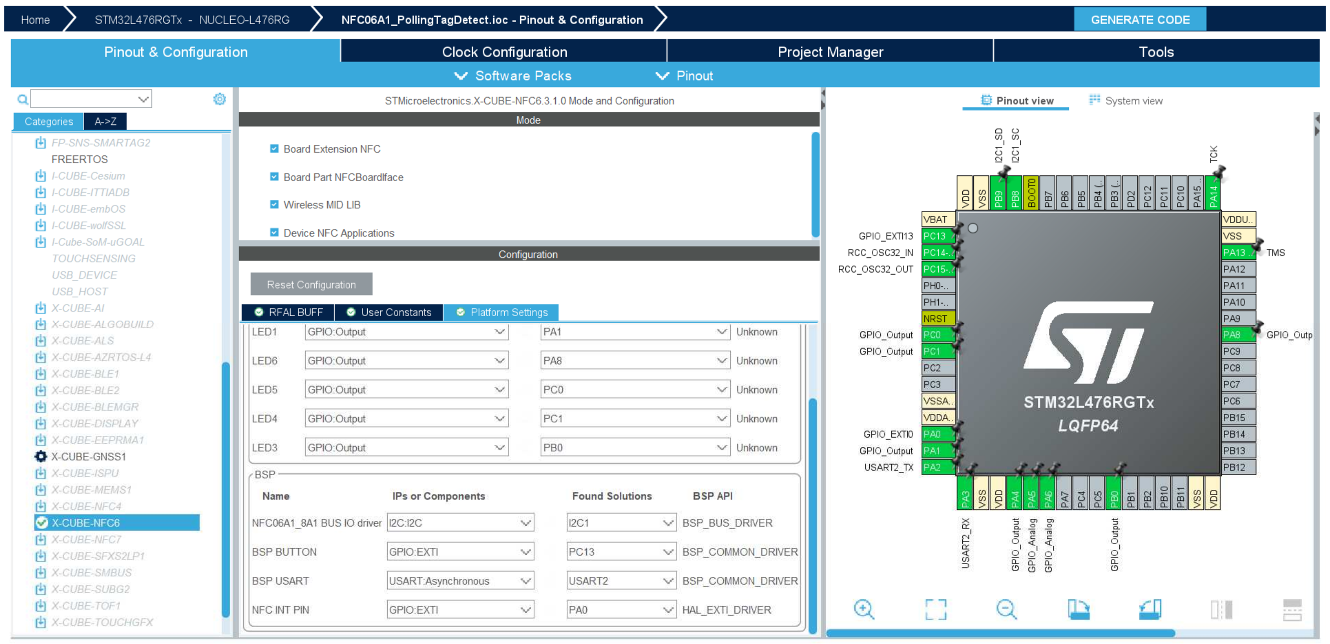
Reset Configuration

Parameter Settings | User Constants | NVIC Settings | DMA Settings | GPIO Settings

Search Signals

Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/P...	Maximum outpu...	Fast Mode
PB8	I2C1_SCL	n/a	Alternate Functi...	Pull-up	Very High	Disable
PB9	I2C1_SDA	n/a	Alternate Functi...	Pull-up	Very High	Disable

Figure 26: Pin Connection of I2C enabling and selecting in software packs.



STMicroelectronics.X-CUBE-NFC6.3.1.0 Mode and Configuration

Mode

- ☑ Board Extension NFC
- ☑ Board Part NFCBoardIface
- ☑ Wireless MID LIB
- ☑ Device NFC Applications

Configuration

Reset Configuration

RFAL BUFF | User Constants | Platform Settings

LED1	GPIO:Output	PA1	Unknown
LED6	GPIO:Output	PA8	Unknown
LED5	GPIO:Output	PC0	Unknown
LED4	GPIO:Output	PC1	Unknown
LED3	GPIO:Output	PB0	Unknown

BSP

Name	IPs or Components	Found Solutions	BSP API
NFC06A1_8A1 BUS IO driver	I2C:I2C	I2C1	BSP_BUS_DRIVER
BSP BUTTON	GPIO:EXTI	PC13	BSP_COMMON_DRIVER
BSP USART	USART:Asynchronous	USART2	BSP_COMMON_DRIVER
NFC INT PIN	GPIO:EXTI	PA0	HAL_EXTI_DRIVER

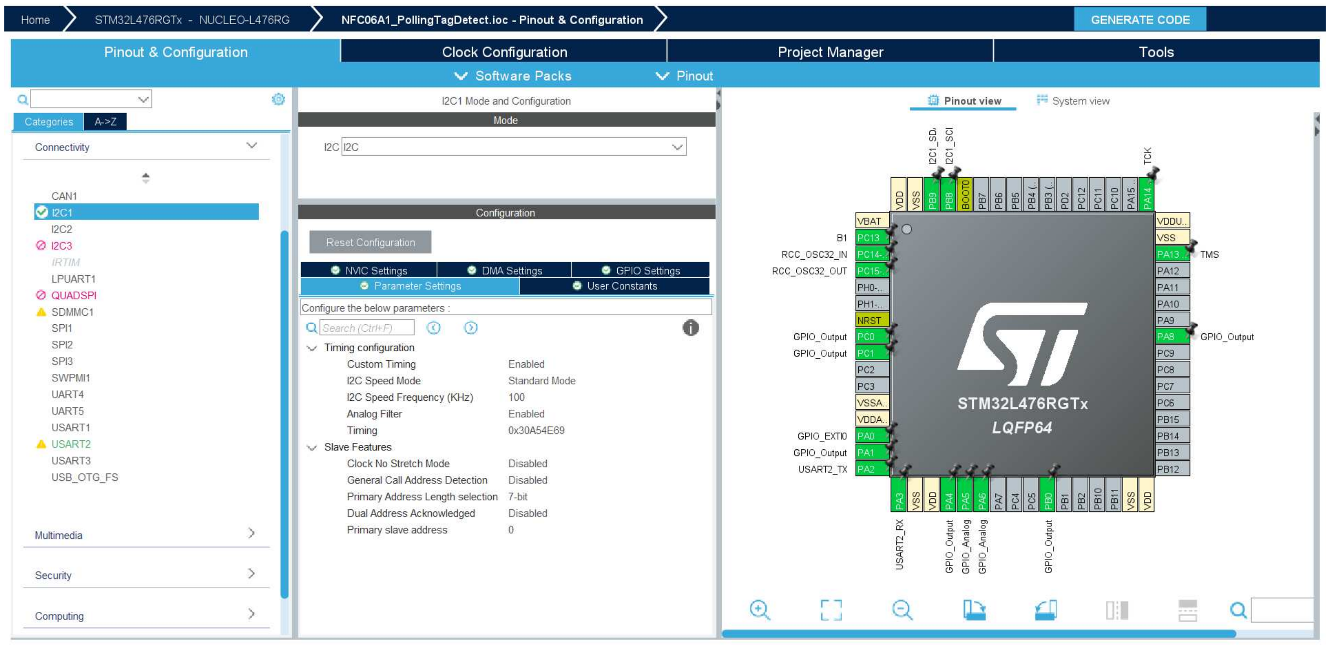
Table 5. **Detailed Pin Description for I2C of NUCLEO-L476RG.**

Name	BSP_API	Supported IPs	Nucleo-64
NFC06A1 BUS IO Driver	BSP_BUS_DRIVER	I2C	I2C1
NFC INT PIN	HAL_EXTI_DRIVER	GPIO EXTI	PA0
NFC06A1 LED Pins LED2	GPIO Output Pin	Output	PA4
NFC06A1 LED Pins LED1	GPIO Output Pin	Output	PA1
NFC06A1 LED Pins LED6	GPIO Output Pin	Output	PA8
NFC06A1 LED Pins LED5	GPIO Output Pin	Output	PC0
NFC06A1 LED Pins LED4	GPIO Output Pin	Output	PC1
NFC06A1 LED Pins LED3	GPIO Output Pin	Output	PB0
BSP BUTTON	BSP_COMMON_DRIVER	EXTI push Button	PC13[B1]
BSP USART	BSP_COMMON_DRIVER	USART	USART2

Table 6. **Detailed Pin Description for I2C of NUCLEOG-G0B1RE.**

Name	BSP_API	Supported IPs	Nucleo-64
NFC06A1 BUS IO Driver	BSP_BUS_DRIVER	I2C	I2C1
NFC INT PIN	HAL_EXTI_DRIVER	GPIO EXTI	PA0
NFC06A1 LED Pins LED2	GPIO Output Pin	Output	PA4
NFC06A1 LED Pins LED1	GPIO Output Pin	Output	PA1
NFC06A1 LED Pins LED6	GPIO Output Pin	Output	PA8
NFC06A1 LED Pins LED5	GPIO Output Pin	Output	PB11
NFC06A1 LED Pins LED4	GPIO Output Pin	Output	PB12
NFC06A1 LED Pins LED3	GPIO Output Pin	Output	PB1
BSP BUTTON	BSP_COMMON_DRIVER	EXTI push Button	PC13[B1]
BSP USART	BSP_COMMON_DRIVER	USART	USART2

Figure 27: Parameter Configuration for I2C



I2C: I2C

Configure the below parameters :

Timing configuration

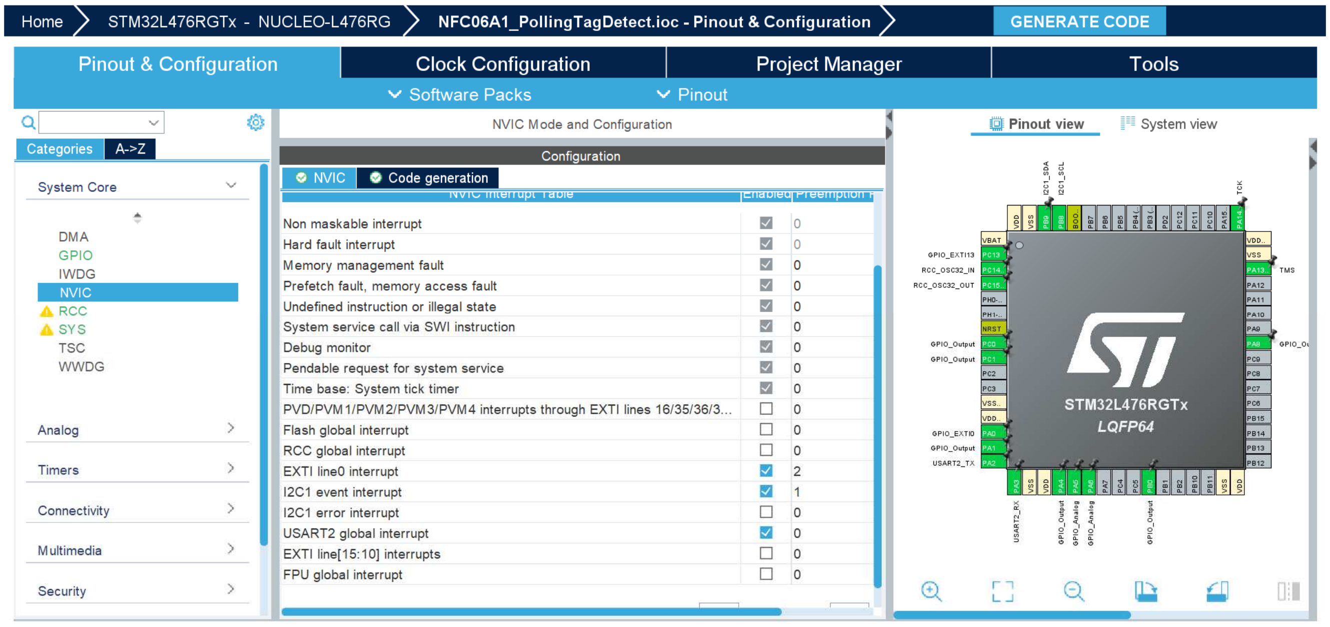
Custom Timing	Enabled
I2C Speed Mode	Standard Mode
I2C Speed Frequency (KHz)	100
Analog Filter	Enabled
Timing	0x30A54E69

Slave Features

Clock No Stretch Mode	Disabled
General Call Address Detection	Disabled
Primary Address Length selection	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0

From the Configuration & Pinout tab, click on “System Core” category and then on NVIC item to enable the EXTI line0 interrupt and set it as “External Interrupt Mode with Rising edge trigger detection”.

Figure 28: STM32CubeMX NVIC Configuration for I2C.

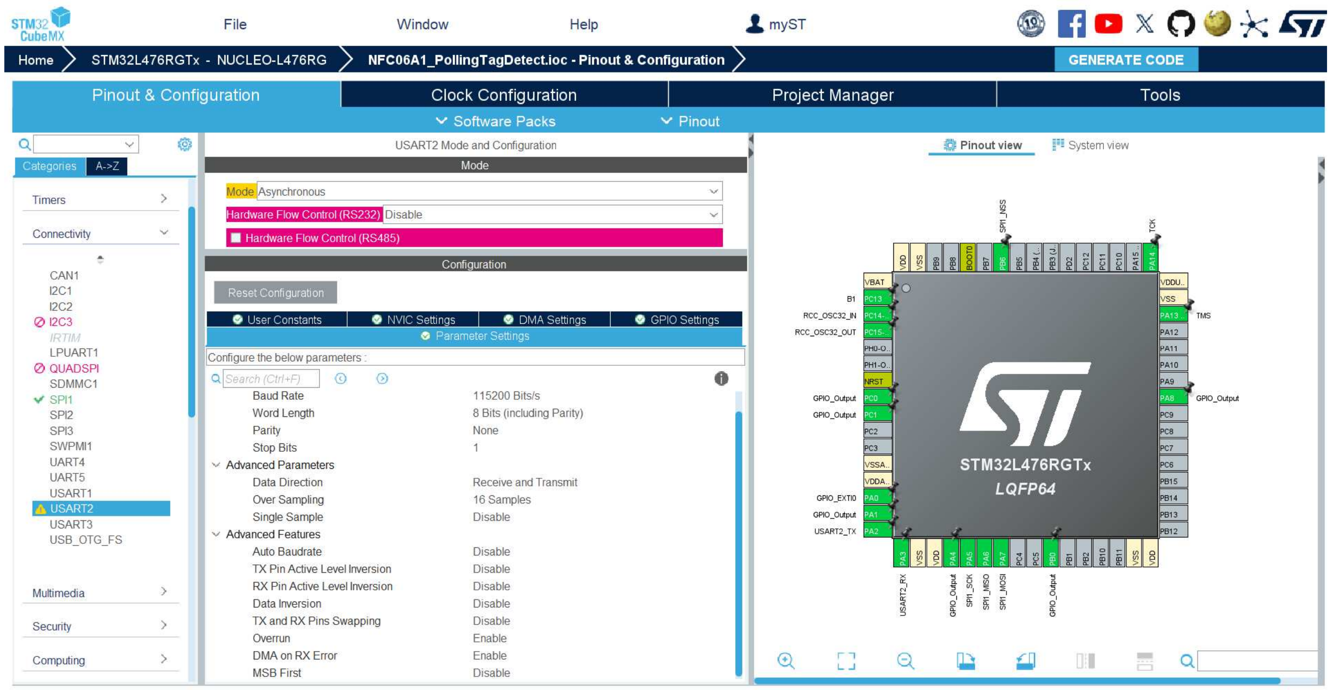


Home > STM32L476RGTx - NUCLEO-L476RG > NFC06A1_PollingTagDetect.ioc - Pinout & Configuration — GENERATE CODE

NVIC Mode and Configuration — Configuration — NVIC | Code generation

NVIC Interrupt Table	Enabled	Preemption P
Non maskable interrupt	✓	0
Hard fault interrupt	✓	0
Memory management fault	✓	0
Prefetch fault, memory access fault	✓	0
Undefined instruction or illegal state	✓	0
System service call via SWI instruction	✓	0
Debug monitor	✓	0
Pendable request for system service	✓	0
Time base: System tick timer	✓	0
PVD/PVM1/PVM2/PVM3/PVM4 interrupts through EXTI lines 16/35/36/3…	☐	0
Flash global interrupt	☐	0
RCC global interrupt	☐	0
EXTI line0 interrupt	✓	2
I2C1 event interrupt	✓	1
I2C1 error interrupt	☐	0
USART2 global interrupt	✓	0
EXTI line[15:10] interrupts	☐	0
FPU global interrupt	☐	0

Figure 29: Parameter Configuration of USART for I2C/SPI



Home > STM32L476RGTx - NUCLEO-L476RG > NFC06A1_PollingTagDetect.ioc - Pinout & Configuration — GENERATE CODE

USART2 Mode and Configuration

Mode

Mode	Asynchronous
Hardware Flow Control (RS232)	Disable
☐ Hardware Flow Control (RS485)	

Configuration — Reset Configuration — User Constants | NVIC Settings | DMA Settings | GPIO Settings | Parameter Settings

Configure the below parameters :

Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
Advanced Parameters	
Data Direction	Receive and Transmit
Over Sampling	16 Samples
Single Sample	Disable
Advanced Features	
Auto Baudrate	Disable
TX Pin Active Level Inversion	Disable
RX Pin Active Level Inversion	Disable
Data Inversion	Disable
TX and RX Pins Swapping	Disable
Overrun	Enable
DMA on RX Error	Enable
MSB First	Disable

6.3 RFAL Feature selection in Custom configuration.

Below is the list of RFAL Features, that can be selected as per user:

RFAL_FEATURE_LISTEN_MODE
RFAL_FEATURE_WAKEUP_MODE
RFAL_FEATURE_LOWPOWER_MODE
RFAL_FEATURE_NFCA
RFAL_FEATURE_NFCB
RFAL_FEATURE_NFCF
RFAL_FEATURE_NFCV
RFAL_FEATURE_T1T
RFAL_FEATURE_T2T
RFAL_FEATURE_T4T
RFAL_FEATURE_ST25TB
RFAL_FEATURE_ST25xV
RFAL_FEATURE_DYNAMIC_ANALOG_CONFIG
RFAL_FEATURE_DPO
RFAL_FEATURE_ISO_DEP
RFAL_FEATURE_ISO_DEP_POLL
RFAL_FEATURE_ISO_DEP_LISTEN
RFAL_FEATURE_NFC_DEP

Above Modules depends on other and that further depends on others.

- **NDEF**
  - **RFAL_FEATURE_T1T**
  - **RFAL_FEATURE_T2T**
  - **RFAL_FEATURE_NFCF**
  - **RFAL_FEATURE_T4T**
  - **RFAL_FEATURE_NFCV**
  - **RFAL_FEATURE_ST25xV**
- **RFAL_FEATURE_LISTEN_MODE** (Dependency none, base RF driver)
- **RFAL_FEATURE_WAKEUP_MODE** (Dependency none, base RF driver)
- **RFAL_FEATURE_LOWPOWER_MODE** (Dependency none, base RF driver)
- **RFAL_FEATURE_NFCA** (Dependency none, base RF driver)
- **RFAL_FEATURE_NFCB** (Dependency none, base RF driver)
- **RFAL_FEATURE_NFCF** (Dependency none, base RF driver)
- **RFAL_FEATURE_NFCV** (Dependency none, base RF driver)
- **RFAL_FEATURE_T1T**
  - **RFAL_FEATURE_NFCA** (will compile without it, but not a valid config)
- **RFAL_FEATURE_T2T**
  - **RFAL_FEATURE_NFCA** (will compile without it, but not a valid config)
- **RFAL_FEATURE_T4T**
  - **RFAL_FEATURE_ISO_DEP**
- **RFAL_FEATURE_ST25TB**
  - **RFAL_FEATURE_NFCB**
- **RFAL_FEATURE_ST25xV**
  - **RFAL_FEATURE_NFCV**

- **RFAL_FEATURE_DYNAMIC_ANALOG_CONFIG** (Dependency none, base RF driver)

- **RFAL_FEATURE_DPO** (Dependency none, base RF driver)

- **RFAL_FEATURE_ISO_DEP**
    - **RFAL_FEATURE_ISO_DEP_POLL || RFAL_FEATURE_ISO_DEP_LISTEN**
    - **RFAL_FEATURE_NFCA** (will compile without it, but not a valid config)
    - **RFAL_FEATURE_NFCB**
- **RFAL_FEATURE_NFC_DEP**
    - **RFAL_FEATURE_NFCA** (will compile without it, but not a valid config).
    - **RFAL_FEATURE_NFCF**

Revision history

Table 7. **Document revision history.**

Date	Version	Changes
18-Jul-2019	1	Initial release.
4-May-2022	2	Updated for X-NUCLEO-NFC08A1 support
5-Sep-2023	3	Updated for CubeMX Support
19-Jan-2024	4	Updated for support of NUCLEO-G0B1RE board