

## Getting started with the X-CUBE-NFC4 dynamic NFC/RFID tag IC software expansion for STM32Cube

---

### Introduction

The X-CUBE-NFC4 software expansion for STM32Cube provides a complete middleware for STM32 to build applications using dynamic NFC/RFID tag IC (ST25DV device).

The software is based on STM32Cube technology and expands STM32Cube based packages. It 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-NFC04A1 expansion board plugged on top of NUCLEO-F401RE or NUCLEO-L053R8.

## Contents

<b>1</b>	<b>Acronyms and abbreviations .....</b>	<b>5</b>
<b>2</b>	<b>What is STM32Cube? .....</b>	<b>6</b>
2.1	STM32Cube architecture .....	6
<b>3</b>	<b>X-CUBE-NFC4 software expansion for STM32Cube .....</b>	<b>8</b>
3.1	Overview .....	8
3.2	Architecture .....	8
3.3	Folder structure .....	9
3.4	APIs .....	10
3.5	Sample application description.....	10
3.5.1	Drive ST25DV by PC software via USB sample description .....	10
3.5.2	Energy harvesting sample .....	16
3.5.3	Activate GPO .....	16
3.5.4	Activate LPD .....	16
3.5.5	Set I <sup>2</sup> C protection.....	16
3.5.6	Use ST25DV mailbox .....	16
3.5.7	Write URI NDEF .....	17
<b>4</b>	<b>System setup guide.....</b>	<b>18</b>
4.1	Hardware description .....	18
4.1.1	STM32 Nucleo platform.....	18
4.1.2	X-NUCLEO-NFC04A1 expansion board .....	18
4.2	Hardware and software setup .....	20
4.2.1	Hardware setup .....	20
4.2.2	Software setup.....	20
4.3	System setup guide.....	21
4.3.1	STM32 Nucleo and X-NUCLEO-NFC04A1 dynamic NFC/RFID tag IC expansion board setup .....	21
<b>5</b>	<b>Revision history .....</b>	<b>22</b>

List of tables

Table 1: List of acronyms.....5

Table 2: Document revision history .....22

## List of figures

Figure 1: Firmware architecture .....	6
Figure 2: X-CUBE-NFC4 software architecture .....	9
Figure 3: X-CUBE-NFC4 package folder structure .....	9
Figure 4: STM32 Nucleo board detected as COM4 in the Device Manager .....	11
Figure 5: STM32 Nucleo board - PC: communication protocol user interface .....	12
Figure 6: Automatic connect STM32 Nucleo board button .....	12
Figure 7: Connect STM32 Nucleo board button .....	12
Figure 8: Disconnect STM32 Nucleo board button .....	12
Figure 9: X-NUCLEO-NFC04A1 expansion board detection process .....	13
Figure 10: X-NUCLEO-NFC04A1 expansion board user interface button .....	13
Figure 11: X-NUCLEO-NFC04A1 expansion board user interface: EEPROM tab .....	13
Figure 12: X-NUCLEO-NFC04A1 expansion board user interface: EEPROM write button .....	14
Figure 13: X-NUCLEO-NFC04A1 expansion board user interface: prepare NDEF file tab .....	14
Figure 14: X-NUCLEO-NFC04A1 expansion board user interface: write NDEF file tab .....	15
Figure 15: X-NUCLEO-NFC04A1 expansion board user interface: read NDEF file tab .....	15
Figure 16: X-NUCLEO-NFC04A1 expansion board user interface: edit CCfile tab .....	15
Figure 17: STM32 Nucleo board .....	18
Figure 18: X-NUCLEO-NFC04A1 expansion board .....	19
Figure 19: X-NUCLEO-NFC04A1 expansion board plugged to an STM32 Nucleo board .....	20

# 1 Acronyms and abbreviations

**Table 1: List of acronyms**

Acronym	Description
NFC	Near field communication
NDEF	NFC data exchange format
RFID	Radio frequency identification
URI	Uniform resource identifier
AAR	Android application record
SMS	Short message service
MCU	Micro controller unit
BSP	Boot support package
HAL	Hardware abstraction layer
LED	Light emitting diode
CMSIS	The ARM® Cortex® microcontroller software interface standard

## 2 What is STM32Cube?

STM32Cube™ represents the STMicroelectronics initiative to make developers' lives easier by reducing development effort, time and cost. STM32Cube covers the STM32 portfolio.

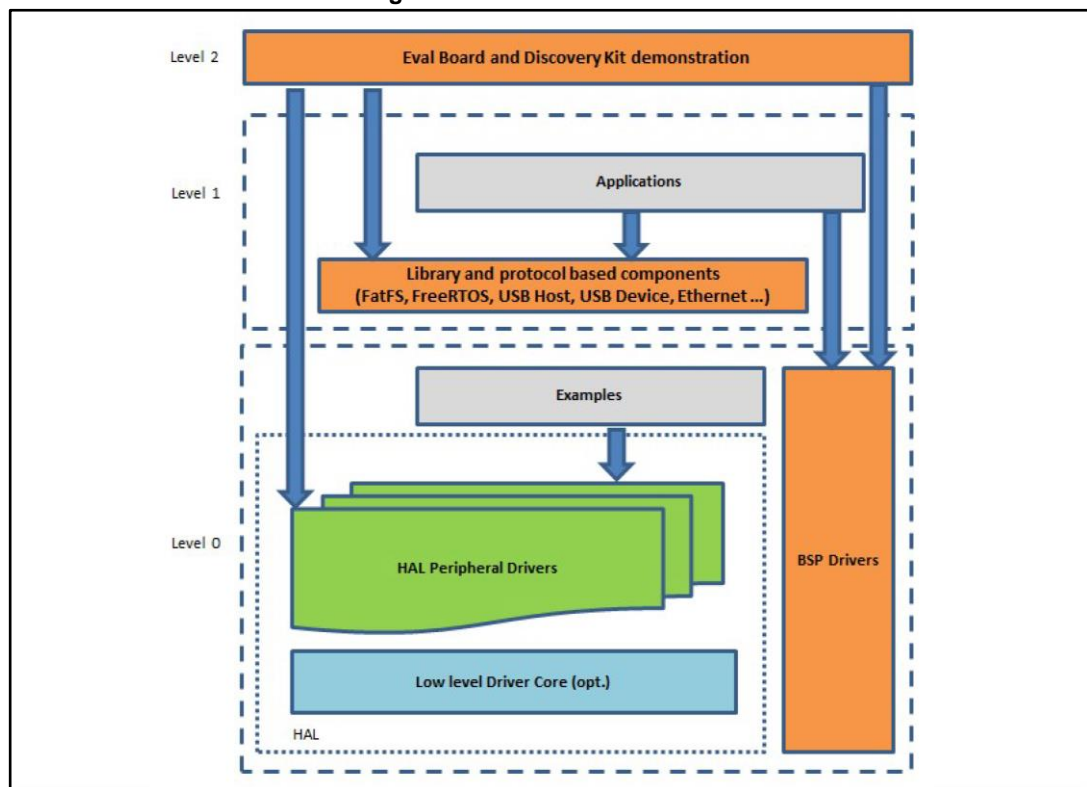
STM32Cube version 1.x includes:

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

### 2.1 STM32Cube architecture

The STM32Cube firmware solution is built around three independent levels that can easily interact with one another, as described in the diagram below.

**Figure 1: Firmware architecture**



**Level 0:** This level is divided into three sub-layers:

- Board Support Package (BSP): this layer offers a set of APIs relative to the hardware components in the hardware boards (Audio codec, IO expander, Touchscreen, SRAM driver, LCD drivers. etc...); it is based on modular architecture allowing it to be easily

ported on any hardware by just implementing the low level routines. It is composed of two parts:

- Component: is the driver relative to the external device on the board and not related to the STM32, the component driver provides specific APIs to the external components of the BSP driver, and can be ported on any other board.
- BSP driver: links the component driver to a specific board and provides a set of easy to use APIs. The API naming convention is BSP\_FUNCT\_Action(): e.g., BSP\_LED\_Init(), BSP\_LED\_On().
- Hardware Abstraction Layer (HAL): this layer provides the low level drivers and the hardware interfacing methods to interact with the upper layers (application, libraries and stacks). It provides generic, multi-instance and function-oriented APIs to help offload user application development time by providing ready to use processes. For example, for the communication peripherals (I<sup>2</sup>C, UART, etc.) it provides APIs for peripheral initialization and configuration, data transfer management based on polling, interrupt or DMA processes, and communication error management. The HAL Drivers APIs are split in two categories: generic APIs providing common, generic functions to all the STM32 series and extension APIs which provide special, customized functions for a specific family or a specific part number.
- Basic peripheral usage examples: this layer houses the examples built around the STM32 peripherals using the HAL and BSP resources only.

**Level 1:** This level is divided into two sub-layers:

- Middleware components: set of libraries covering USB Host and Device Libraries, STemWin, FreeRTOS, FatFS, LwIP, and PolarSSL. Horizontal interaction among the components in this layer is performed directly by calling the feature APIs, while vertical interaction with low-level drivers is managed by specific callbacks and static macros implemented in the library system call interface. For example, FatFs implements the disk I/O driver to access a microSD drive or USB Mass Storage Class.
- Examples based on the middleware components: each middleware component comes with one or more examples (or applications) showing how to use it. Integration examples that use several middleware components are provided as well.

**Level 2:** This level is a single layer with a global, real-time and graphical demonstration based on the middleware service layer, the low level abstraction layer and basic peripheral usage applications for board-based functions.

## 3 X-CUBE-NFC4 software expansion for STM32Cube

### 3.1 Overview

The X-CUBE-NFC4 software package extends the STM32Cube functionality.

The key features are:

- Complete middleware to build applications using dynamic NFC/RFID tag IC (ST25DV)
- Easy portability across different MCU families, thanks to STM32Cube
- Sample application to communicate with PC software
- Samples to use the basic ST25DV features
- Free user-friendly license terms
- Sample implementation available on the X-NUCLEO-NFC04A1 expansion board plugged on top of a NUCLEO-F401RE or a NUCLEO-L053R8 board

The package extends STM32Cube by providing a board support package (BSP) for the X-NUCLEO-NFC04A1 expansion board and some middleware components for communication with PC software and NDEF application library.

The drivers abstract hardware low-level details and allow the component and applications to access NDEF data in a hardware independent fashion or to communicate with PC software through USB.

The package includes a sample application and six samples to activate ST25DV features:

- Drive ST25DV by PC software via USB
- Energy harvesting enabling
- GPO interrupt activation
- Low power down activation
- I<sup>2</sup>C protection setting
- ST25DV Mailbox usage
- URI NDEF writing

### 3.2 Architecture

This software is a fully compliant expansion for STM32Cube enabling development of applications using dynamic NFC/RFID tag IC<sup>a</sup>.

The software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a board support package (BSP) for the X-NUCLEO-NFC04A1 expansion board.

The software layers used by the application software to access and use the X-NUCLEO-NFC04A1 expansion board are:

- **STM32Cube HAL driver layer:** this layer provides a generic multi instance simple set of APIs (application programming interfaces) to interact with the upper layers (application, libraries and stacks). It is composed of generic and extension APIs. It is directly built around a generic architecture and allows the layers that are built upon, such as the middleware layer, to implement their functionalities without dependencies on the specific hardware configuration for a given Microcontroller Unit (MCU). This

---

<sup>a</sup> Refer to the previous chapter for an overview of the STM32Cube architecture.

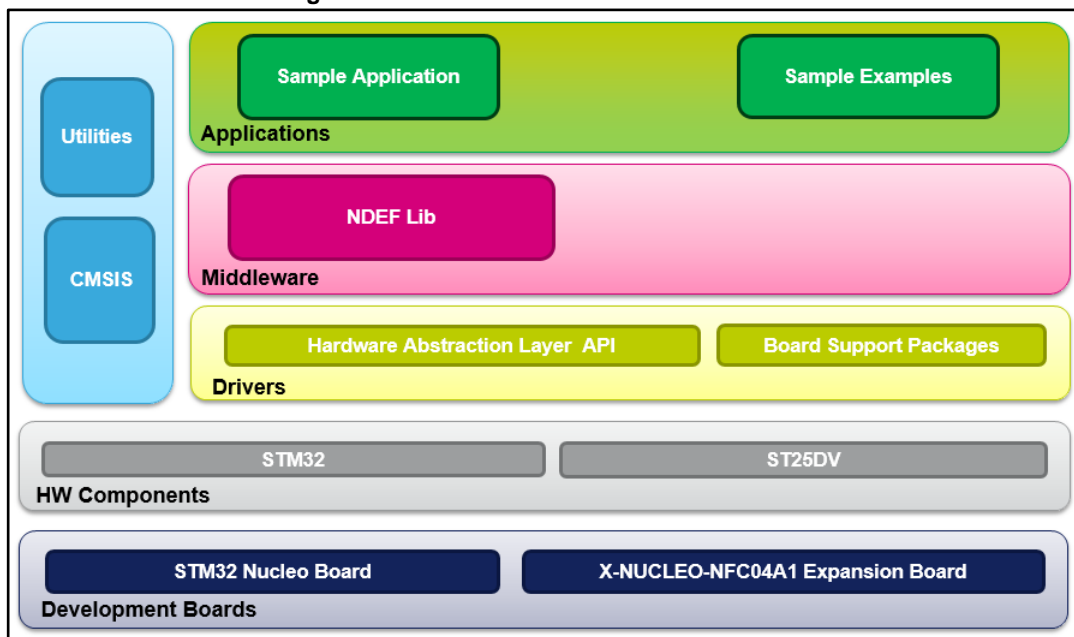


structure improves the library code reusability and guarantees high portability on other devices.

- **Board Support Package (BSP) layer:** the software package needs to support the peripherals on the STM32 Nucleo board apart from the MCU. It contains a limited set of APIs which provides a programming interface for certain board specific peripherals (e.g. the LED, the user button etc.) and allows identifying the specific board version.

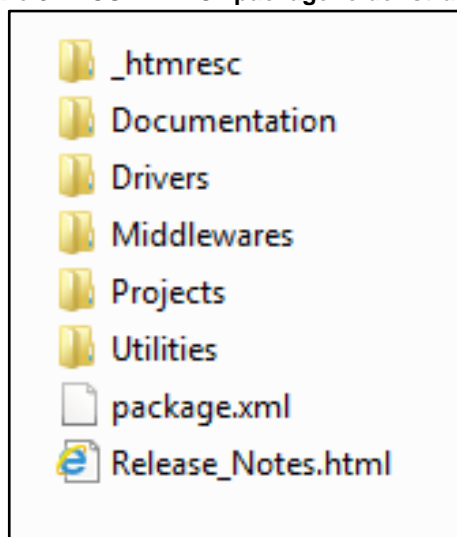
The following figure outlines the software architecture of the package:

Figure 2: X-CUBE-NFC4 software architecture



### 3.3 Folder structure

Figure 3: X-CUBE-NFC4 package folder structure



The following folders are included in the software package:

- Documentation: contains a compiled HTML file generated from the source code and documentation detailing the software components and APIs.
- Drivers: contains the HAL drivers and the board specific drivers for each supported board or hardware platform, including the on-board components and the CMSIS layer - a vendor-independent hardware abstraction layer for the ARM Cortex-M processor series.
- Middlewares: contains NDEF application drivers and protocols related to NFC data communication.
  - NDEF AAR (to add AAR (Android application record) in the tag)
  - NDEF Email (to manage the NDEF file that represents e-mail)
  - NDEF Geo (to manage the NDEF file that represents geo-location)
  - NDEF MyApp (to manage the NDEF file of a private application)
  - NDEF SMS (to manage the NDEF file that represents SMS)
  - NDEF Text (to manage the text NDEF file)
  - NDEF URI (to manage the URI NDEF file)
  - NDEF Vcard (to manage the Vcard NDEF file)
- Projects: contains the sample applications described in [Section 3.1: "Overview"](#) for the NUCLEO-L053R8 and NUCLEO-F401RE platforms with three development environments (IAR Embedded Workbench for ARM, RealView MDK-ARM Microcontroller Development Kit, System Workbench for STM32).
- Utilities: contains a PC software tool<sup>a</sup> that can drive the ST25DV; communication is performed through the ST-LINK USB way.

## 3.4 APIs

Detailed technical information fully describing the APIs available to the user can be found in a compiled HTML file located inside the software package Documentation folder.

## 3.5 Sample application description

### 3.5.1 Drive ST25DV by PC software via USB sample description

A sample application using the X-NUCLEO-NFC04A1 expansion board with the NUCLEO-F401RE or NUCLEO-L053R8 board is provided in the "Projects" directory. Ready-to-use projects are available for multiple IDEs.

In this application, a PC software tool<sup>b</sup> can drive the ST25DV through the ST-LINK USB.

After system initialization and clock configuration, three LEDs (green, blue and orange) are sequentially turned ON. After NFC tag and UART initialization, the three LEDs are sequentially turned OFF. When the firmware receives a valid frame, the blue LED flashes once.

The device is recognized as a serial COM port (see [Section 3.5.1.1: "Device recognition as a serial COM port"](#)) and you can open the X-NUCLEO-NFC04A1 expansion board user interface (see [Section 3.5.1.2: "Opening the user interface for the X-NUCLEO-NFC04 expansion board"](#)).

---

<sup>a</sup> The executable is available in the Utilities/PC\_Software/X-NUCLEO-NFC\_software folder of the X-CUBE-NFC4 package

<sup>b</sup> The executable is available in the Utilities/PC\_Software/X-NUCLEO-NFC\_software folder of the X-CUBE-NFC4 package.

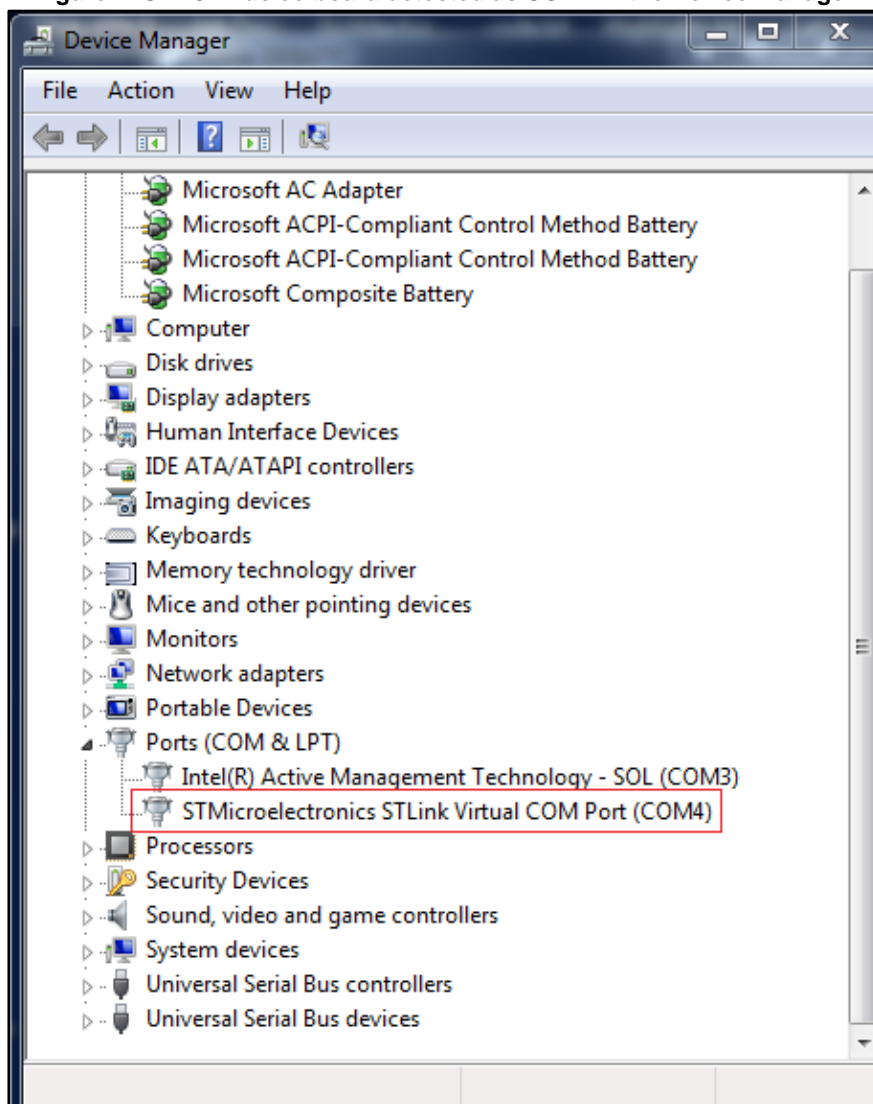
### 3.5.1.1 Device recognition as a serial COM port

- 1 Connect the STM32 Nucleo board to your computer via the USB cable.
- 2 Install the serial COM port driver.
- 3 Check the board is well connected and recognized.
- 4 Open the Windows Device Manager tool and check that the STMicroelectronics STLink Virtual COM Port is detected successfully.



The COM port number assigned to the STM32 Nucleo board is shown in the Device Manager.

Figure 4: STM32 Nucleo board detected as COM4 in the Device Manager

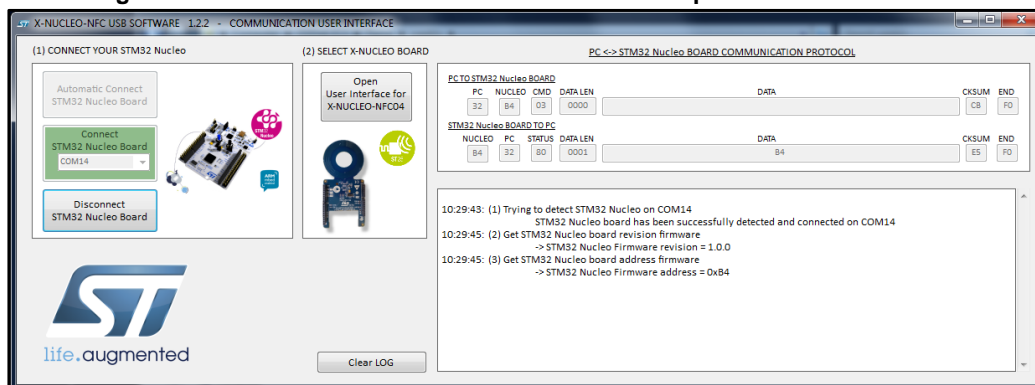




In case of error or warning in the Device Manager window, check that the STM32 Virtual COM port driver has been successfully installed or the STM32 Nucleo board is connected to your computer.

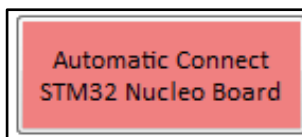
- 5 Launch X-NUCLEO-NFC\_software.exe.  
The first window will help to manage USB communication.

**Figure 5: STM32 Nucleo board - PC: communication protocol user interface**



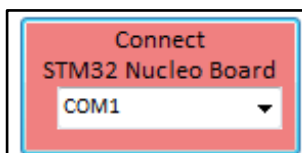
- 6 Select **Automatic connect STM32 Nucleo board** button.  
All the COM port (from COM1 to COM256) are scanned to detect the STM32 Nucleo board.

**Figure 6: Automatic connect STM32 Nucleo board button**



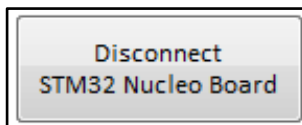
- 7 Connect the STM32 Nucleo board by selecting the COM port number and clicking on the button to detect the board used.

**Figure 7: Connect STM32 Nucleo board button**



- 8 Disconnect the STM32 Nucleo board by clicking on the button shown below:

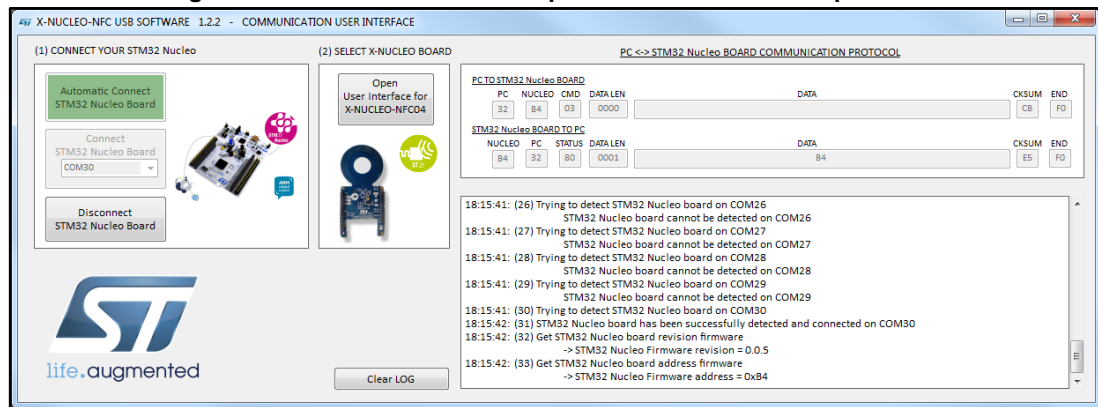
**Figure 8: Disconnect STM32 Nucleo board button**



### 3.5.1.2 Opening the user interface for the X-NUCLEO-NFC04 expansion board

When the STM32 Nucleo board is connected, the communication user interface detects the X-NUCLEO-NFC04A1 expansion board.

Figure 9: X-NUCLEO-NFC04A1 expansion board detection process



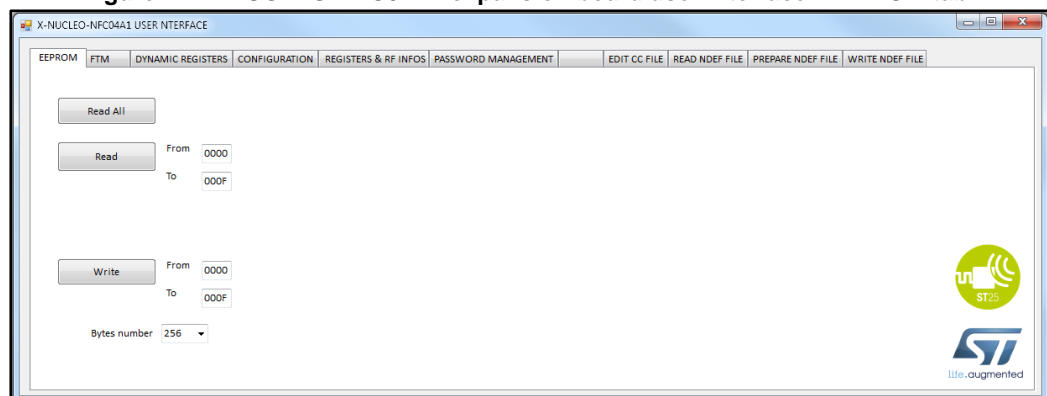
- 1 Open the user interface for the X-NUCLEO-NFC04A1 by clicking on the related button.

Figure 10: X-NUCLEO-NFC04A1 expansion board user interface button



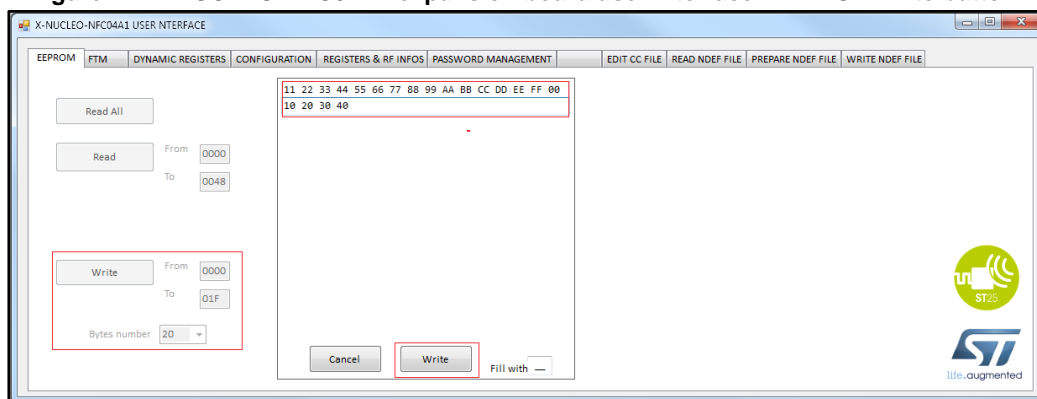
You are now connected to the STM32 Nucleo firmware and can interact with the ST25DV device by using the new window interface.

Figure 11: X-NUCLEO-NFC04A1 expansion board user interface: EEPROM tab



- 2 Access ST25DV registers and memory area by selecting the various tabs. For example, in the first tab (shown in the above image), you can access the ST25DV EEPROM content: **Read all** allows you to read all the memory, whereas **Read** from "start address" to "end address" allow you to read only a part of the memory and **Write** allows you to write into the memory (see the following picture).

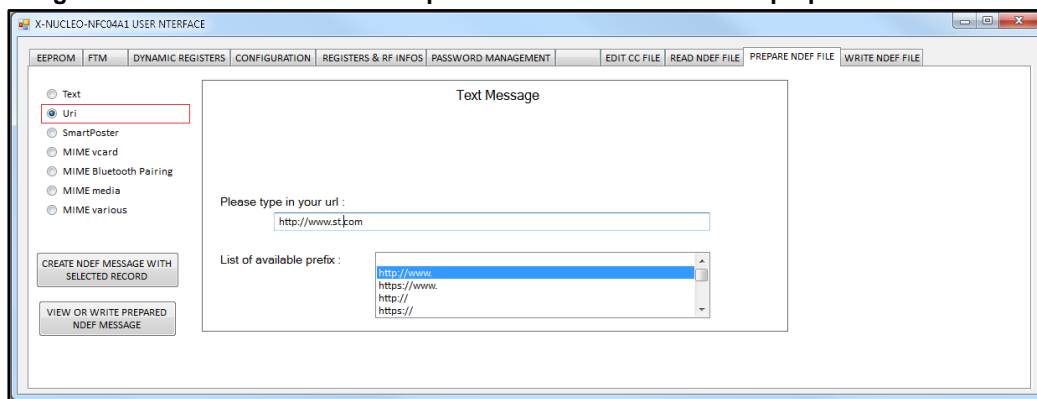
Figure 12: X-NUCLEO-NFC04A1 expansion board user interface: EEPROM write button



The other tabs are designed to access all ST25DV registers<sup>a</sup>:

- **FTM** tab refers to the fast transfer mode (using the Mailbox feature)
- **Dynamic registers** tab refers to volatile registers
- **Configuration** tab refers to static registers
- **Registers & RF infos** tab refers to read-only registers
- **Password management** tab refers to the I<sup>2</sup>C session feature
- The empty tab is a separator
- The remaining tabs are dedicated to NDEF type 5, to format (see Figure 13: "X-NUCLEO-NFC04A1 expansion board user interface: prepare NDEF file tab"), write (Figure 14: "X-NUCLEO-NFC04A1 expansion board user interface: write NDEF file tab") or read (see Figure 15: "X-NUCLEO-NFC04A1 expansion board user interface: read NDEF file tab") an NDEF file

Figure 13: X-NUCLEO-NFC04A1 expansion board user interface: prepare NDEF file tab



<sup>a</sup> For details on each register, refer to the ST25DV datasheet on [www.st.com](http://www.st.com).

Figure 14: X-NUCLEO-NFC04A1 expansion board user interface: write NDEF file tab

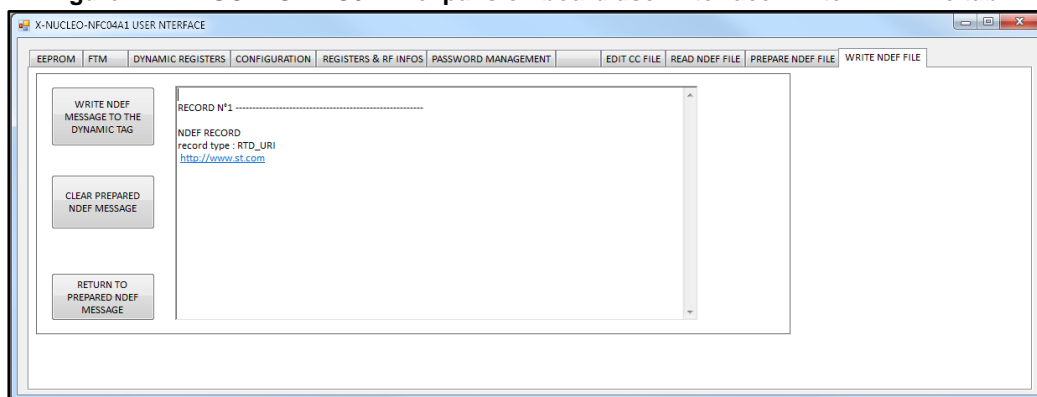
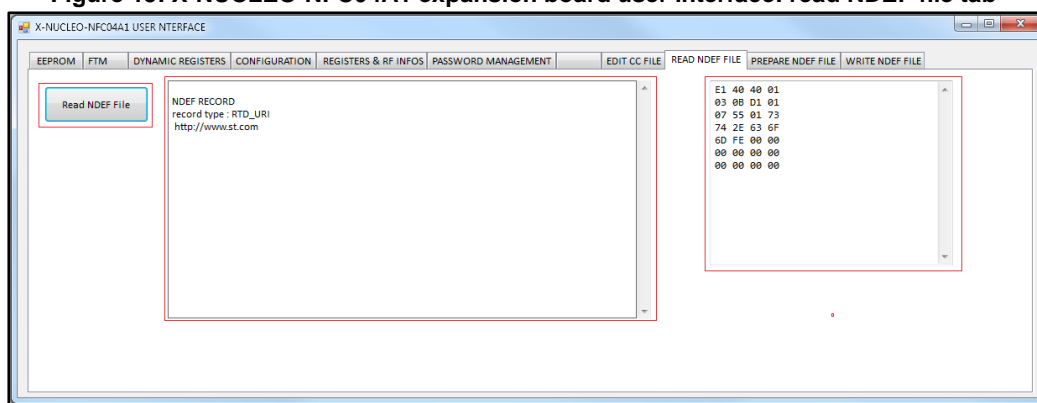


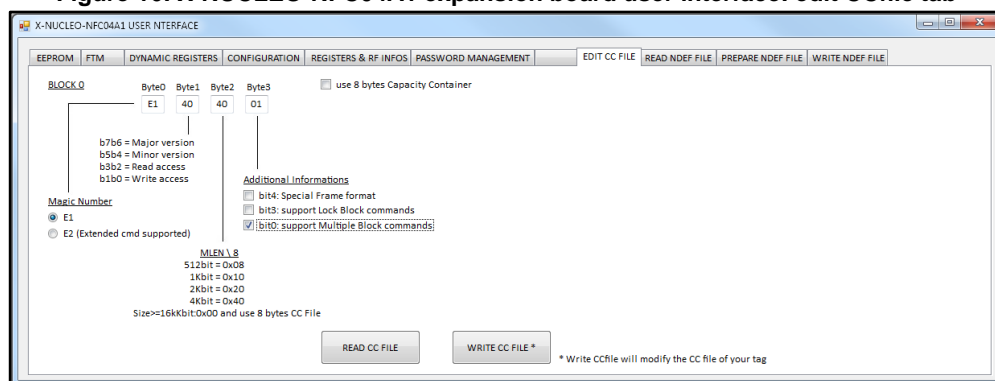
Figure 15: X-NUCLEO-NFC04A1 expansion board user interface: read NDEF file tab



- 3 Format the memory if there is no NDEF nor CCFile
- 4 Use the **EDIT CC File** tab to format your tag to comply with the Type 5 Tag Operation v1.0 NFC Forum specification, as the Type 5 tag must be filled with a CCFile at the beginning of the memory.

In the ST25DV available on the X-NUCLEO-NFC04 expansion board, the CCFile is composed of four bytes to be configured manually. The description of each byte is given in the dialog box<sup>a</sup>.

Figure 16: X-NUCLEO-NFC04A1 expansion board user interface: edit CCfile tab



<sup>a</sup> For further details on CCFile format, please refer to the NFC Forum specification Type 5 Tag Operation v1.0.

### 3.5.2 Energy harvesting sample

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.

### 3.5.3 Activate GPO

This sample shows how to enable and use the GPO.

After initialization, an interrupt is programmed to detect field changes in proximity of the ST25DV. The green LED is switched ON when the field is detected and switched OFF when the field disappears.

### 3.5.4 Activate LPD

This sample shows how to activate low power down (LPD) pin.

By pressing the user button, the LPD pin is activated (green LED is switched ON) or deactivated.

When the LPD pin is activated, the ST25DV  $V_{CC}$  is cut off, the power consumption is minimum and communication via I<sup>2</sup>C is not available.

### 3.5.5 Set I<sup>2</sup>C protection

This sample shows how to create areas in the ST25DV and how to protect them.

Text is displayed on a UART console (via ST-LINK) if a PC is connected.

### 3.5.6 Use ST25DV mailbox

This sample shows how to write a message into the mailbox and how to read mailbox status register.

Text is displayed on a UART console (via ST-LINK) if a PC is connected.



### 3.5.7 Write URI NDEF

This sample shows how to write an NDEF message to the ST25DV EEPROM using the NDEF lib middleware. The yellow LED is switched ON when the message has been successfully written.



The tag URI message can be read using an Android smartphone and the ST25 NFC app<sup>a</sup>, available on Google Play.

---

<sup>a</sup> For further information, refer to [http://www.st.com/content/st\\_com/en/products/embedded-software/st25-nfc-rfid-software/stsw-st25001.html](http://www.st.com/content/st_com/en/products/embedded-software/st25-nfc-rfid-software/stsw-st25001.html)

## 4 System setup guide

### 4.1 Hardware description

#### 4.1.1 STM32 Nucleo platform

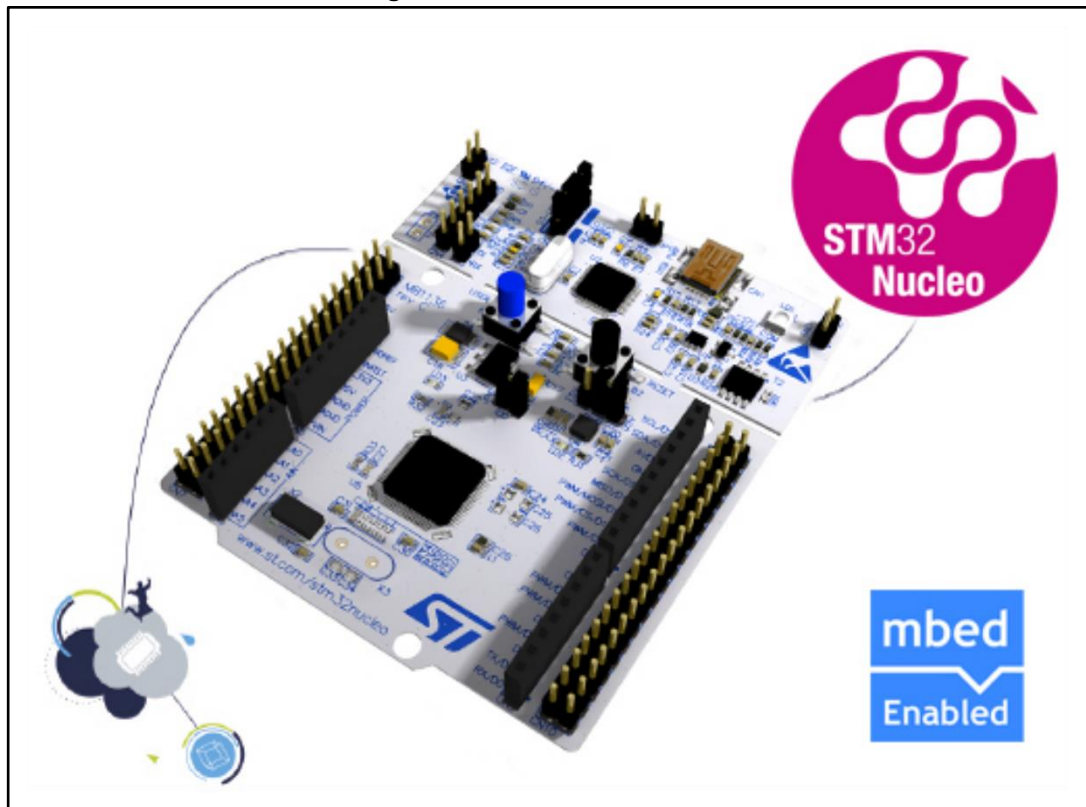
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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 17: STM32 Nucleo board



Information regarding the STM32 Nucleo board is available at [www.st.com/stm32nucleo](http://www.st.com/stm32nucleo)

#### 4.1.2 X-NUCLEO-NFC04A1 expansion board

The X-NUCLEO-NFC04A1 dynamic NFC/RFID tag IC expansion board is based on the ST25DV04K NFC Type V/RFID tag IC with a dual interface 4 Kbits EEPROM that also features an I<sup>2</sup>C interface. It can be powered by the pin of Arduino connector or directly by the received carrier electromagnetic field.

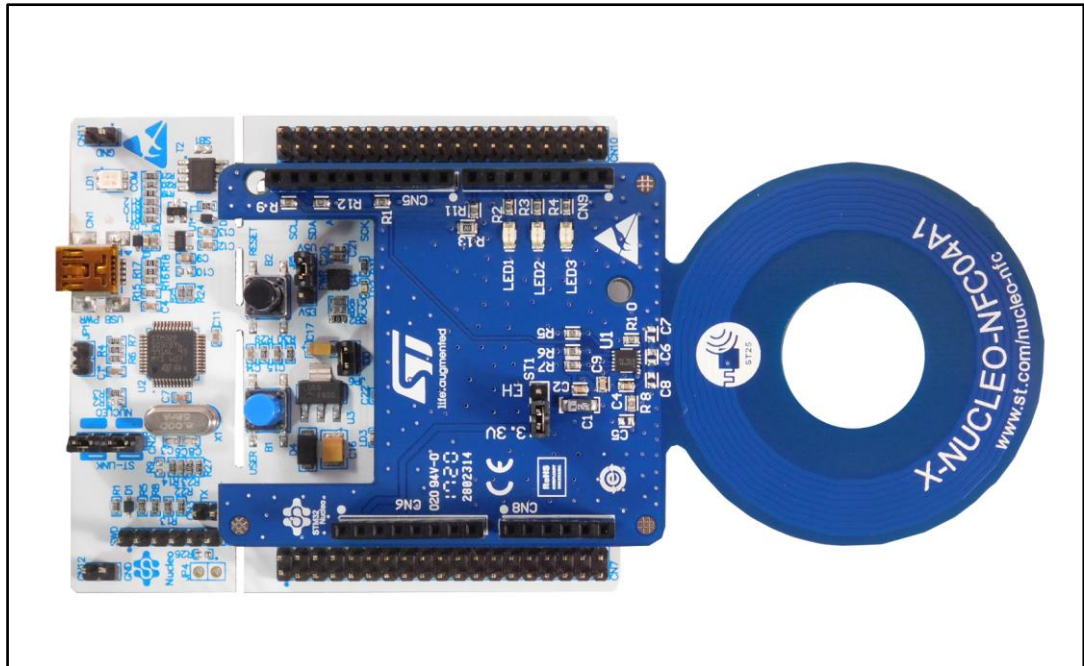
The X-NUCLEO-NFC04A1 expansion board is compatible with the Arduino™ UNO R3 connector pin assignment and can easily be plugged onto any STM32 Nucleo board. Various expansion boards can also be stacked to evaluate different devices operating together with the dynamic NFC tag.

The board also features an antenna with a 54 mm iso 24.2 diameter, single layer, copper etched on PCB.

Figure 18: X-NUCLEO-NFC04A1 expansion board



Figure 19: X-NUCLEO-NFC04A1 expansion board plugged to an STM32 Nucleo board



Information about the X-NUCLEO-NFC04A1 expansion board is available on [www.st.com](http://www.st.com) at <http://www.st.com/x-nucleo>

## 4.2 Hardware and software setup

### 4.2.1 Hardware setup

The following hardware components are needed:

- One STM32 Nucleo development platforms (suggested order code: NUCLEO-F401RE or NUCLEO-L053R8)
- One NFC/RFID tag IC expansion board (order code: X-NUCLEO-NFC04A1). The power setting three pin jumper on the shield should be closed at pins (3.3 V and ST1)
- One USB type A to Mini-B USB cable to connect the two STM32 Nucleo board to the PC

### 4.2.2 Software setup

This section lists the minimum requirements for the developer to set up the SDK.

#### 4.2.2.1 Development Tool-chains and Compilers

Select one of the Integrated Development Environments supported by the STM32Cube expansion software<sup>a</sup> and follow the system requirements and setup information provided by the selected IDE provider.

<sup>a</sup> IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-LINK, RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK and System Workbench for STM32 (SW4STM32) + ST-LINK

## 4.3 System setup guide

This section describes how to set up the different components before writing and executing applications on the STM32 Nucleo board with the dynamic NFC/RFID tag IC expansion board.

### 4.3.1 STM32 Nucleo and X-NUCLEO-NFC04A1 dynamic NFC/RFID tag IC expansion board setup

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. The developer can download the relevant version of the ST-LINK/V2-1 USB driver by searching STSW-LINK009 on [www.st.com](http://www.st.com).

The X-NUCLEO-NFC04A1 expansion board can be easily connected to the STM32 Nucleo board through the Arduino UNO R3 extension connector (see [Figure 19: "X-NUCLEO-NFC04A1 expansion board plugged to an STM32 Nucleo board"](#)) and can interface with the external STM32 microcontroller on the STM32 Nucleo via the I<sup>2</sup>C transport layer.

## 5 Revision history

Table 2: Document revision history

Date	Version	Changes
20-Jun-2017	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.

© 2017 STMicroelectronics – All rights reserved