# UM2045
# User manual

# Getting started with the X-CUBE-NFC3 13.56-MHz multi-protocol contactless transceiver IC software expansion for STM32Cube

## Introduction

This document describes how to get started with the X-CUBE-NFC3 software expansion for STM32Cube.

X-CUBE-NFC3 provides a complete middleware for STM32 to build applications using 13.56-MHz multi-protocol contactless transceiver IC (CR95HF-VMD5T). It is easily portable across different MCU families, thanks to STM32Cube. This package contains a sample application example to detect NFC tags of different class, to write URI text into the detected passive tag, and then to read back the URI text from the tag.

The software provides implementation examples for STM32 Nucleo platforms equipped with the X-NUCLEO-NFC03A1 expansion board, featuring 13.56-MHz multi-protocol contactless transceiver IC (CR95HF-VMD5T).

The software is based on STM32Cube technology and expands STM32Cube based packages.

# Contents

# List of tables

# List of figures

# 1 List of acronyms

**Table 1. List of acronyms**

| Acronym | Description |
|---|---|
| AAR | Android Application Record |
| BSP | Boot Support Package |
| CMSIS | ARM® Cortex® Microcontroller Software Interface Standard |
| HAL | Hardware Abstraction Layer |
| IDE | Integrated Development Environment |
| LED | Light Emitting Diode |
| MCU | Micro Controller Unit |
| NDEF | NFC Data Exchange Format |
| NFC | Near Field Communication |
| SMS | Short Message Service |
| SPI | Serial Peripheral Interface |
| UART | Universal Asynchronous Receiver Transmitter |
| URI | Uniform Resource Identifier |

# 2 References

- CR95HF-VMD5T datasheet (available on *www.st.com*)
- STM32 microcontroller datasheets (available on *www.st.com*)
- Nucleo board user manuals (available on *www.st.com*)

# 3 What is STM32Cube?

## 3.1 STM32Cube overview

STMCube™ initiative was originated by STMicroelectronics to ease developers' life by reducing development efforts, time and cost. STM32Cube covers STM32 portfolio.

STM32Cube Version 1.x includes:

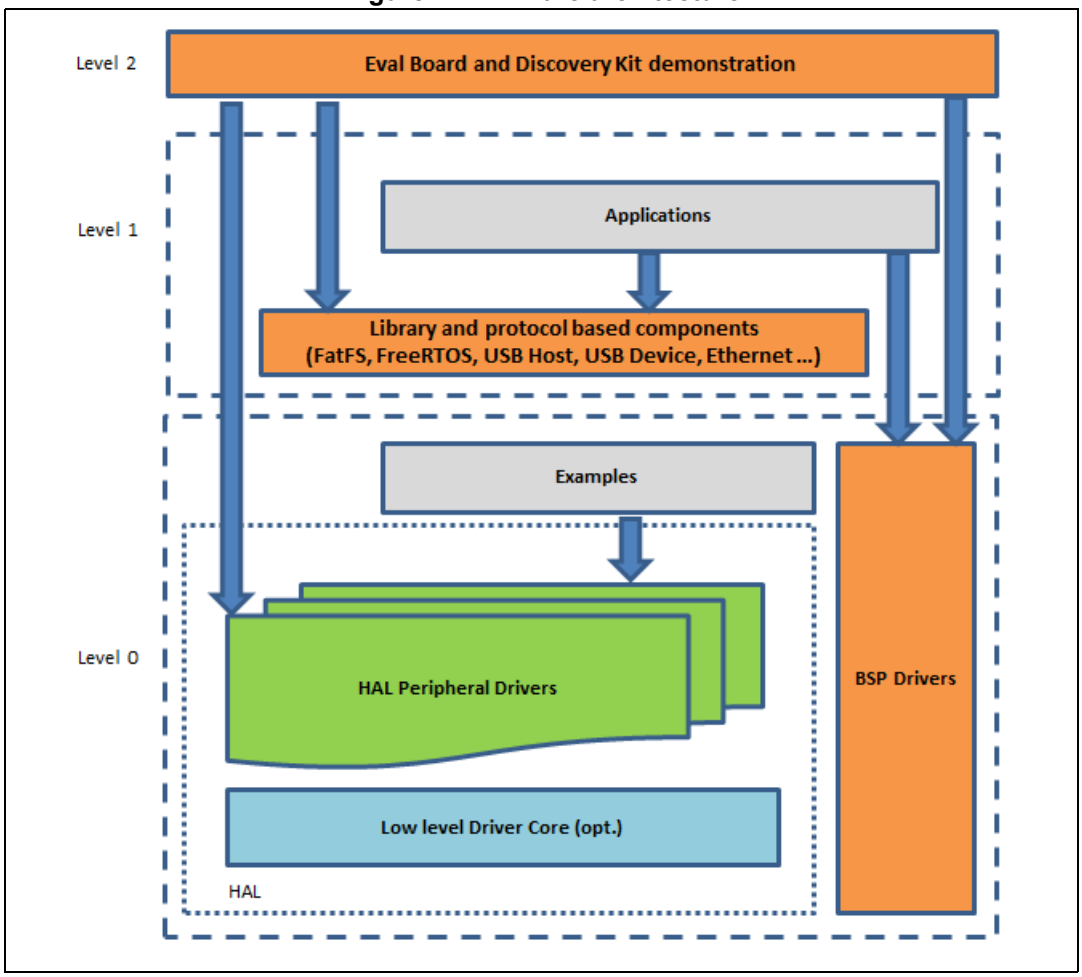
- The STM32CubeMX, a graphical software configuration tool that allows to generate C initialization code using graphical wizards.
- A comprehensive embedded software platform, delivered per series (such as STM32CubeF4 for STM32F4 series)
  - The STM32Cube HAL, an STM32 abstraction layer embedded software,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

Information about STM32Cube are available on *www.st.com*.

## 3.2 STM32Cube Architecture

The STM32Cube firmware solution is built around three independent levels that can easily interact with each other's as described in the figure 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…) and 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 provide specific APIs to the BSP driver external components and could be portable on any other board.
  - **BSP driver**: it permits to link the component driver to a specific board and provides a set of friendly used APIs. The APIs naming rule is BSP_FUNCT_Action(): ex. BSP_LED_Init(),BSP_LED_On().

Level 0 is based on modular architecture allowing to port it easily on any hardware by just implementing the low level routines.

- **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). HAL provides a generic, multi instance and functionalities oriented APIs which permit to offload the user application implementation by providing ready to use process. As an example, for the communication peripherals (I2S, UART…), it provides APIs allowing to initialize and configure the peripheral, manage data transfer based on polling, interrupt or DMA process, and manage communication errors that may raise during communication.
  The HAL Drivers APIs are split in two categories:
  – **generic APIs**: it provides common and generic functions to all the STM32 series.
  – **extension APIs**: it provides customized functions for a specific family or a specific part number.
- **Basic peripheral usage examples**: this layer encloses the examples build over the STM32 peripheral using only the HAL and BSP resources.

**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 interactions between the components of this layer is done directly by calling the feature APIs while the vertical interaction with the low level drivers is done through specific callbacks and static macros implemented in the library system call interface. As example, the FatFs implements the disk I/O driver to access microSD drive or the USB Mass Storage Class.
- **Examples based on the Middleware components**: each Middleware component comes with one or more examples (called also Applications) showing how to use it. Integration examples that use several Middleware components are provided as well.

**Level 2:** This level is composed of a single layer:

This layer is global real-time and graphical demonstration. It is based on the Middleware service layer, the low level abstraction layer and the basic peripheral usage applications for board based functionalities.

# 4 X-CUBE-NFC3 software expansion for STM32Cube

## 4.1 Overview

X-CUBE-NFC3 is a software package that expands the functionality provided by STM32Cube.

The key features of the package are:

- Complete middleware to build applications using 13.56-MHz multi-protocol contactless transceiver IC (CR95HF-VMD5T).
- Easy portability across different MCU families, thanks to STM32Cube.
- Sample application example to detect NFC tags of different class.
- Free user-friendly license terms.
- Examples implementation available on board X-NUCLEO-NFC03A1 plugged on top of one NUCLEO-F401RE or NUCLEO-F103RB.

This software is gathering 13.56-MHz multi-protocol contactless transceiver IC drivers for the CR95HF-VMD5T device, running on STM32.

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

The software comes with examples of implementation of such drivers, running on X-NUCLEO-NFC03A1 plugged on top of NUCLEO-F401RE or NUCLEO-F103RB.

The package extends STM32Cube by providing a Board Support Package (BSP) for the X-NUCLEO-NFC03A1 expansion board and some middleware components for NDEF application drivers.

The drivers abstract low-level details of the hardware and allow the middleware components and applications to access NDEF data in a hardware independent fashion.

The package also includes a sample application example that the developer can use to start experimenting with the code. The sample example is developed to detect NFC tags of different class. Other NDEF application drivers that have been included in the middleware are:

- NDEF AAR (to add AAR in the tag)
- NDEF Email (to manage NDEF file that represents Email)
- NDEF Geo (to manage NDEF file that represents geo-location)
- NDEF MyApp (to manage the NDEF file of a private application)
- NDEF SMS (to manage NDEF file that represents SMS)
- NDEF Text (to manage Text NDEF file)
- NDEF URI (to manage NDEF file that represent URI)
- NDEF Vcard (to manage NDEF file that represents Vcard)

## 4.2 Architecture

This software is an expansion for STM32Cube, it fully complies with the architecture of STM32Cube and it expands it in order to enable development of applications using 13.56 MHz multi-protocol contactless transceiver IC. Refer to *Section 3.2* for an introduction about the STM32Cube architecture.

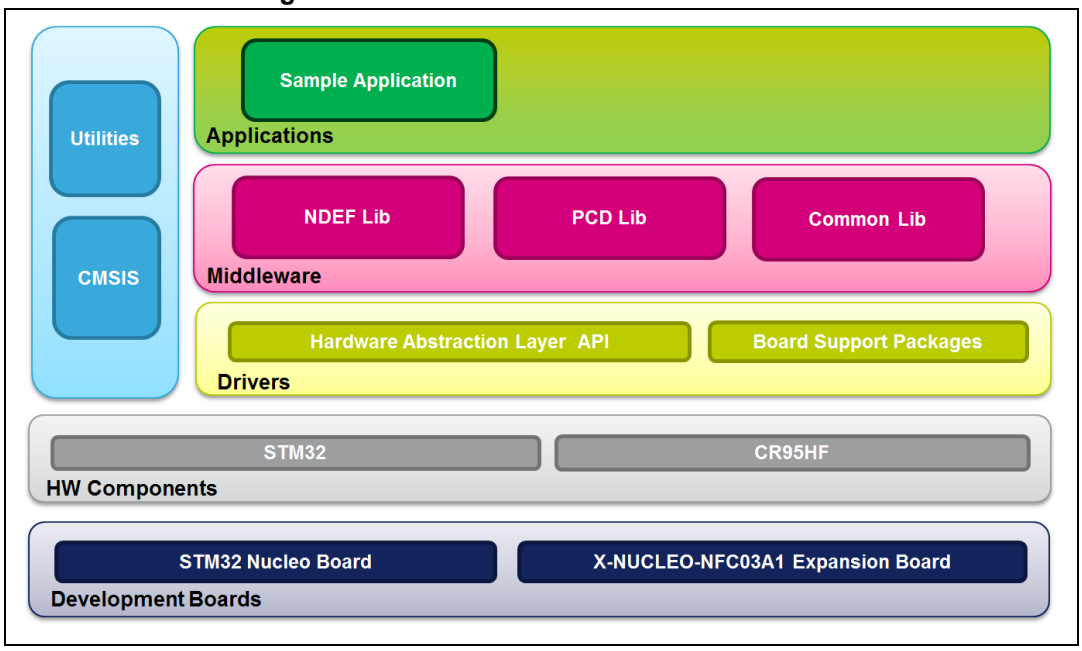
The software is based on the STM32CubeHAL, the hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a Board Support Package (BSP) for the X-NUCLEO-NFC03A1 expansion board.

The software layers used by the application software to access and use the X-NUCLEO-NFC03A1 expansion board are the following:

- STM32Cube HAL Layer: the HAL driver 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 it, such as the middleware layer, to implement their functionalities without dependencies on the specific hardware configuration for a given Microcontroller Unit (MCU). This structure improves the library code reusability and guarantees an easy 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. This software is included in the board support package (BSP). This is a limited set of APIs which provides a programming interface for certain board specific peripherals, e.g. the LED, the user button etc. This interface also helps in identifying the specific board version.

The next figure outlines the software architecture of the package:

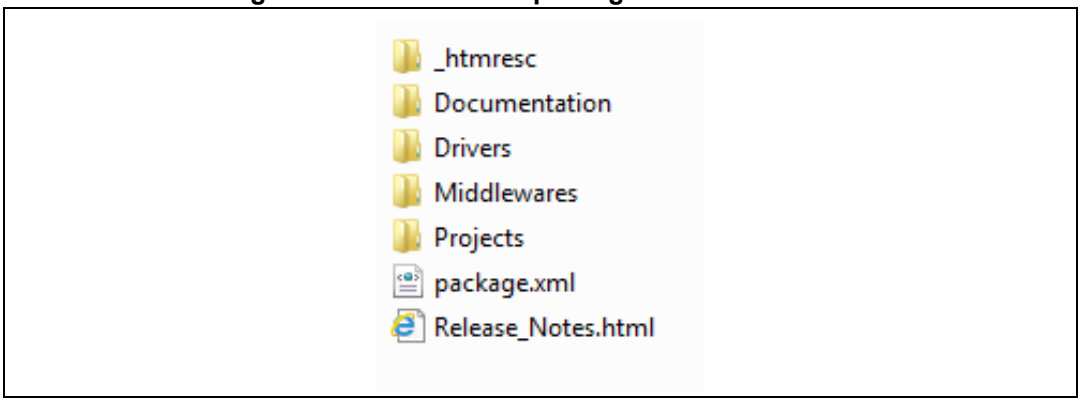
**Figure 2. X-CUBE-NFC3 software architecture**



## 4.3 Folders Structure

This section provides an overview of the package folders structure, shown in *Figure 3*

**Figure 3. X-CUBE-NFC3 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, and documentation with details on the software components and on the APIs.
- **Drivers**: this folder contains the HAL drivers, the board specific drivers for each supported board or hardware platform, including the on-board components ones and the CMSIS layer which is a vendor-independent hardware abstraction layer for the Cortex®-M processor series.
- **Middlewares**: this folder contains NDEF application drivers and protocols related to the communication of NFC data.
  - NDEF AAR (To add AAR (Android Application Record) in the tag)
  - NDEF Email (To manage NDEF file that represents Email)
  - NDEF Geo (To manage NDEF file that represents geo-location)
  - NDEF MyApp (To manage the NDEF file of a private application)
  - NDEF SMS (To manage NDEF file that represents SMS)
  - NDEF Text (To manage Text NDEF file)
  - NDEF URI (To manage URI NDEF file)
  - NDEF Vcard (To manage Vcard NDEF file)
- **Projects**: this folder contains a sample application example Tag Detect, provided for the NUCLEO-F103RB and NUCLEO-F401RE platforms with three development environments (IAR™ Embedded Workbench® for ARM®, RealView® Microcontroller Development Kit (MDK-ARM™), and System Workbench for STM32 (SW4STM32)).

## 4.4 APIs

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

## 4.5 Sample application description

An example application using the X-NUCLEO-NFC03A1 expansion board with either NUCLEO-F401RE or NUCLEO-F103RB board is provided in the “Projects” directory. Ready to be built projects are available for multiple IDEs. By default it will run using SPI interface. To run the application using UART interface, it needs to select the UART target in the project options.

In this application, NFC tags of different classes are detected by the 13.56-MHz multi-protocol contactless transceiver IC (CR95HF-VMD5T).

The firmware package contains an HTML documentation file generated from Doxygen, that is used as sample application reference.

After system initialization and clock configuration, if no tag is detected LED1 keeps blinking, otherwise the LED pattern detailed in *Table 2* is visible on the board.

**Table 2. LED pattern vs. tag detection**

| NFC Tag Type | LEDs ON |
|---|---|
| NFCTYPE1 | LED2, LED3, LED4 |
| NFCTYPE2 | LED2 |
| NFCTYPE3 | LED3 |
| NFCTYPE4A | LED4 |
| NFCTYPE4B | LED3, LED4 |
| NFCTYPE5 | LED2, LED3 |

The board is configured and enumerated as an ST Virtual communication port under windows, as shown in *Figure 4.*

**Figure 4. ST Virtual communication port enumeration**

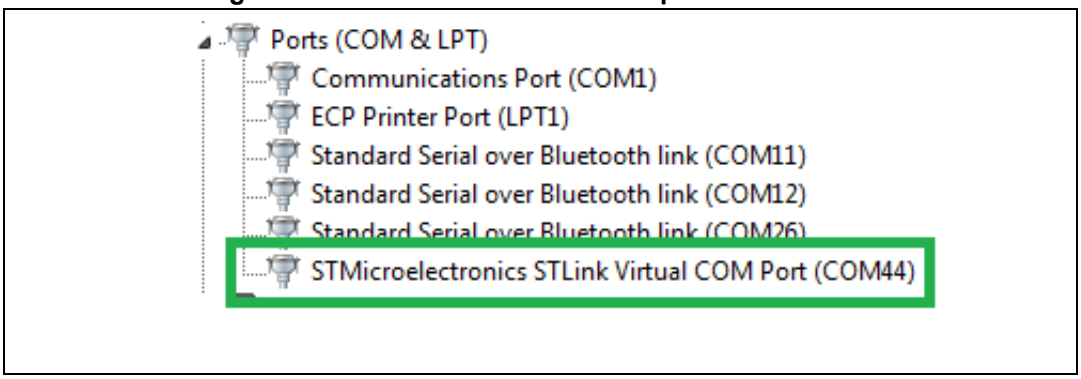
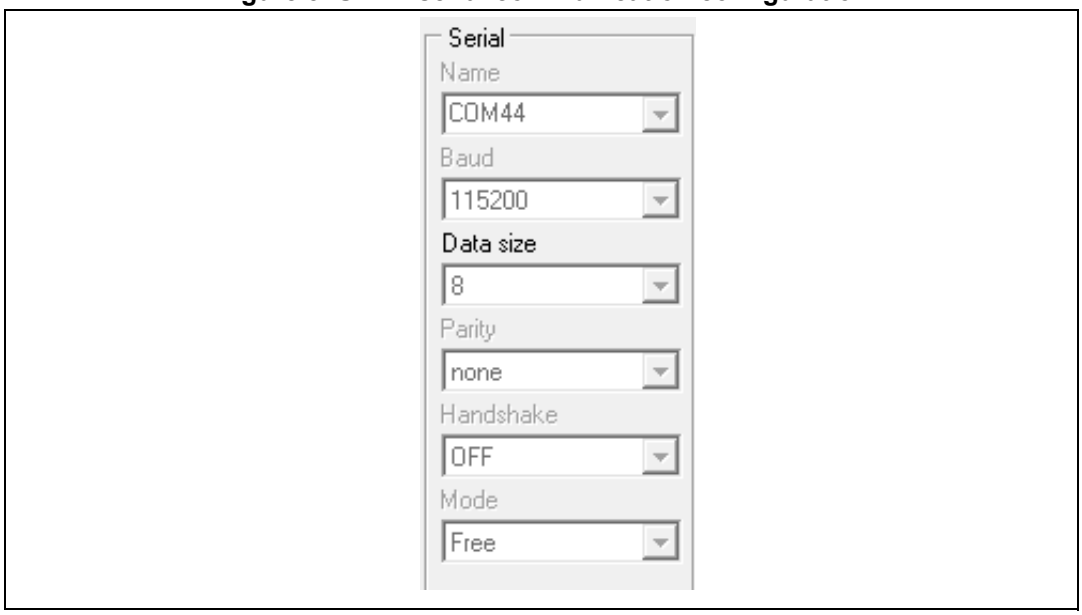


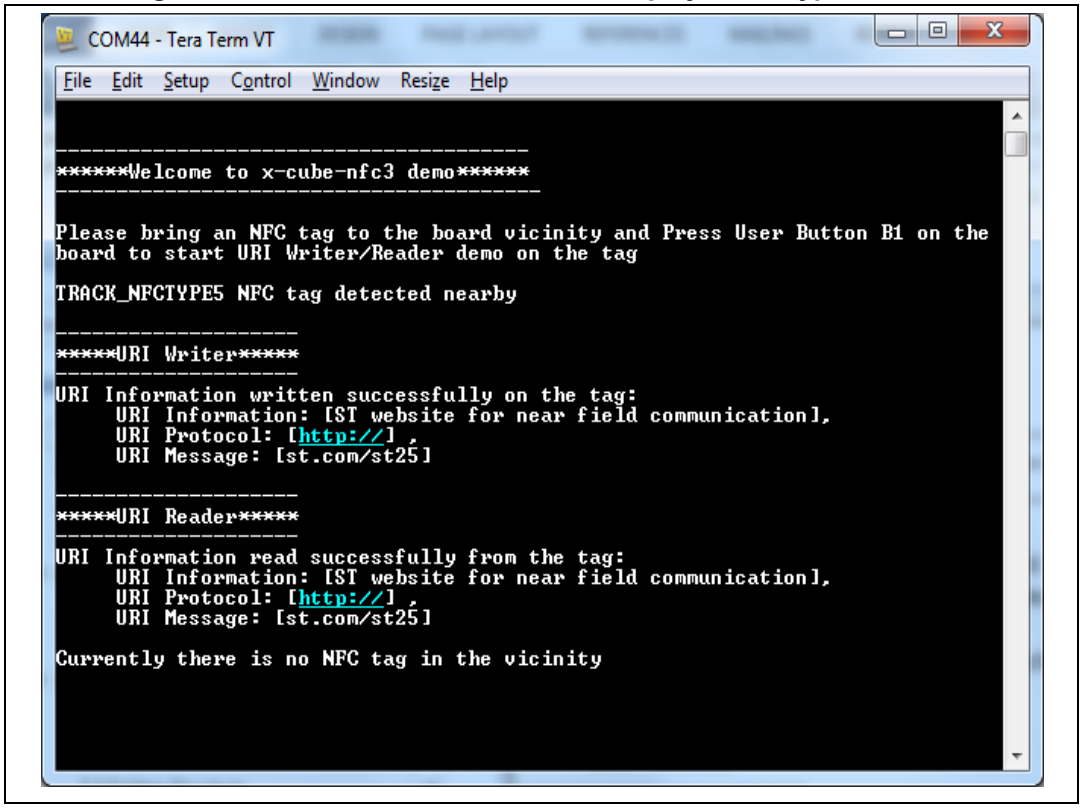
*Figure 5* shows an example of UART serial communication configuration.

**Figure 5. UART serial communication configuration**



After a successful connection, the user can view the messages on the Hyperterminal, as shown in *Figure 6*, and then can bring a passive NFC tag near to the NFC3 shield board Antenna area. Once the tag is detected, a message will be displayed showing the class of detected tag. Keeping the tag nearby, the user can press “User Button B1” to start write and read URI example message on the tag.

**Figure 6. UART serial communication displayed on Hyperterminal**



# 5 System Setup Guide

## 5.1 Hardware Description

This section describes the hardware components required to develop a sensor-based application.

The following sub-sections describe the individual components.

### 5.1.1 STM32 Nucleo platform

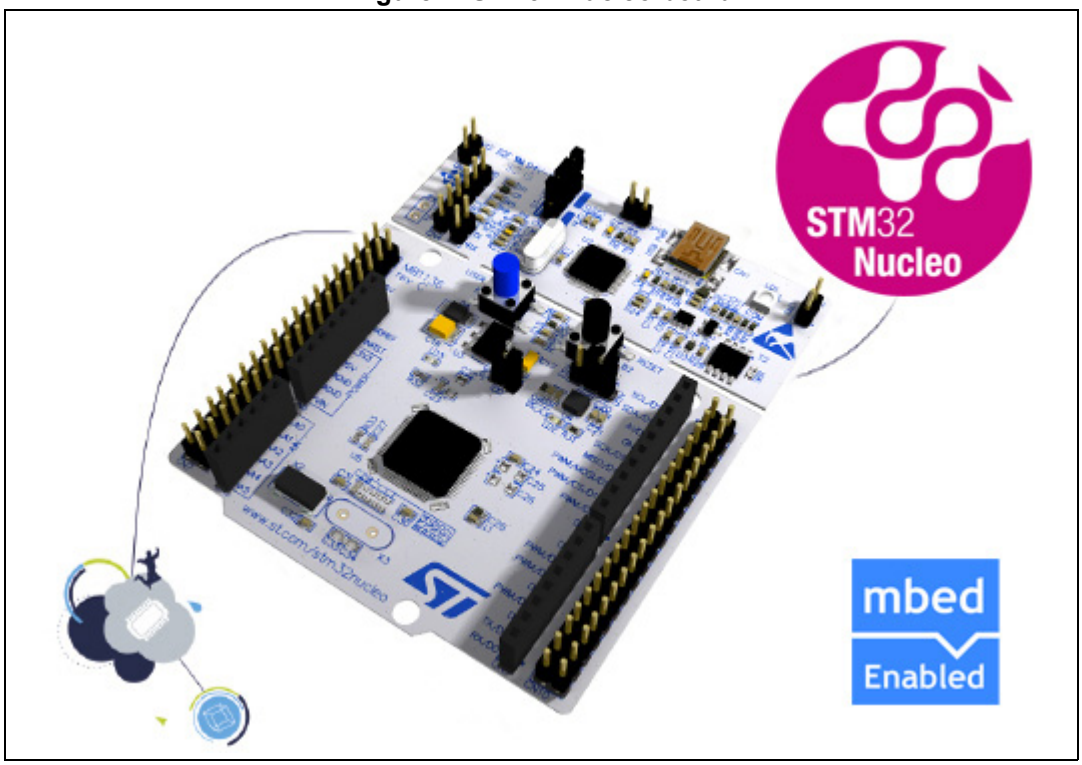
The STM32 Nucleo boards provide an affordable and flexible way for users to test new ideas and build prototypes with any STM32 microcontroller lines.

The Arduino™ connectivity support and ST morpho headers make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide choice of specialized expansion boards.

The STM32 Nucleo board does not require any separate probe, as it integrates the ST-LINK/V2-1 debugger/programmer. The STM32 Nucleo board comes with the STM32 comprehensive software HAL library together with various packaged software examples.

Information about the STM32 Nucleo boards is available on STMicroelectronics website *www.st.com.*

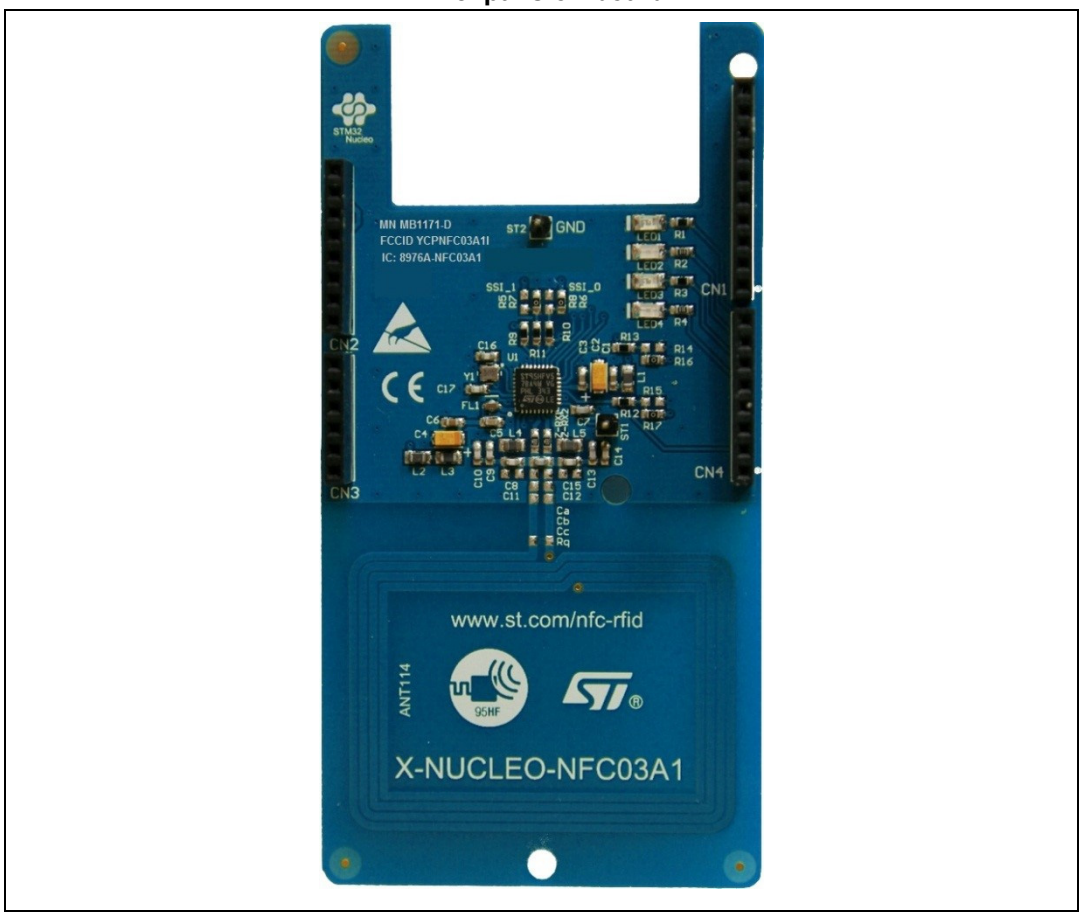
**Figure 7. STM32 Nucleo board**



### 5.1.2 X-NUCLEO-NFC03A1 expansion board

The X-NUCLEO-NFC03A1 is a contactless transceiver IC expansion board that can be used with the STM32 Nucleo platform. It is also compatible with Arduino™ UNO R3 connector layout, and is designed around the STMicroelectronics 13.56-MHz multi-protocol contactless transceiver IC (CR95HF-VMD5T). The X-NUCLEO-NFC03A1 interfaces with the STM32 MCU via SPI/UART pin.

**Figure 8. X-NUCLEO-NFC03A1 13.56-MHz multi-protocol contactless transceiver IC expansion board**




Information about the STM32 Nucleo boards is available on STMicroelectronics website *www.st.com.*

## 5.2 Software Description

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

- X-CUBE-NFC3: an expansion for STM32Cube dedicated to NFC applications development. The X-CUBE-NFC3 firmware and its related documentation are available on *www.st.com*.
- Development tool-chain and Compiler: the STM32Cube expansion software supports the three following environments:
  - IAR™ Embedded Workbench for ARM® (EWARM) toolchain + ST-Link
  - RealView®Microcontroller Development Kit (MDK-ARM™) toolchain + ST-LINK
  - System Workbench for STM32 (SW4STM32) + ST-LINK

## 5.3 Hardware and Software Setup

This section describes the hardware and software setup procedures. It also describes the required system setup.

### 5.3.1 Hardware Setup

The following hardware components are needed:

- One STM32 Nucleo Development platform (order code: either NUCLEO-F401RE or NUCLEO-F103RB)
- One 13.56-MHz multi-protocol contactless transceiver IC (CR95HF-VMD5T) expansion board (order code: X-NUCLEO-NFC03A1)
- One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC

### 5.3.2 Software Setup

This section lists the minimum requirements for the developer to setup the SDK.

#### Development Tool-chains and Compilers

Select one of the Integrated Development Environments supported by the STM32Cube expansion software.

Read the system requirements and setup information provided by the selected IDE provider.

### 5.3.3 System Setup Guide

This section describes how to setup different hardware parts before writing and executing an application on the STM32 Nucleo board with the Sensors expansion board.

#### STM32 Nucleo and 13.56-MHz multi-protocol contactless transceiver IC expansion boards setup

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer.

The developer can download the ST-LINK/V2-1 USB driver by looking at the STSW-LINK009 software on *www.st.com*.

The X-NUCLEO-NFC03A1 expansion board can be easily connected to the STM32 Nucleo motherboard through the Arduino™ UNO R3 extension connector. It is capable of interfacing with the external STM32 microcontroller on STM32 Nucleo board using SPI/UART transport layer. By default it will run using SPI interface. The user can choose to compile with one of the two interfaces by selecting the right target under Projects properties.

# 6 Revision history

Table 3. Document revision history

| Date | Revision | Changes |
|---|---|---|
| 26-May-2016 | 1 | Initial release |