Getting started with the software package for Bluetooth low energy and dynamic NFC tag in FP-SEC-BLENFC1

Introduction

The FP-SEC-BLENFC1 software runs on STM32 microcontrollers with the BlueNRG-MS Bluetooth low energy communication protocol. It writes NDEF protocol information for secure Bluetooth pairing, storing the BLE MAC address and the connection pin on the NFC tag.

The FP-SEC-BLENFC1 also exports a Bluetooth characteristic for switching the STM32 Nucleo board LED on and off, and one for transmitting simulated temperature sensor data.

The software, based on STM32Cube technology, provides a sample implementation for the featured STM32 Nucleo development boards and expansion boards.

1 Acronyms and abbreviations

Table 1. List of acronyms

Acronym	Description
BLE	Bluetooth low energy
NFC	near field communication
NDEF	NFC data exchange format

2 FP-SEC-BLENFC1 software description

2.1 Overview

The key features of the FP-SEC-BLENFC1 package are:

- Complete middleware to build applications with the ST25DV04K dynamic NFC/RFID tag using NDEF standard
- A very low power Bluetooth low energy (BlueNRG-MS) single-mode network processor, compliant with Bluetooth specifications core 4.2 (X-NUCLEO-IDB05A1) for transmitting information to one client
- Easy portability across different MCU families, thanks to STM32Cube
- Compatible with BlueMS application for Android/iOS (Version 2.1.0 and above) available at the respective Play/iTunes stores
- Free, user-friendly license terms
- Sample implementation available for X-NUCLEO-NFC04A1 and X-NUCLEO-IDB05A1 expansion boards on a NUCLEO-F401RE or NUCLEO-L053R8 development board.

This software creates three Bluetooth services:

- one console service with two characteristics
  - An stdin/stdout feature that implements bi-directional client-server communication
  - An stderr feature that implements a one-way channel from the STM32 Nucleo board to an Android/iOS device
- One configuration service used for sending configuration signal to FP-SEC-BLENFC1
- The last service is used for exposing these characteristics:
  - The status of the STM32 Nucleo LED (on/off)
  - One simulated temperature sensor

This package is compatible with the BlueMS Android/iOS application (Version 2.1.0 and above) available at the respective Play/iTunes stores. This application can be used to display information sent by the Bluetooth low energy protocol.

2.2 Architecture

This 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 BlueNRG-MS and the dynamic NFC tag expansion boards, and some middleware components for communication with other Bluetooth low energy devices, and to enable data exchange with an NFC-capable device using the NDEF standard.

The implementation makes use of low power consumption strategies suitable for this field of application, compliant with the Bluetooth specifications core 4.2 (X-NUCLEO-IDB05A1).

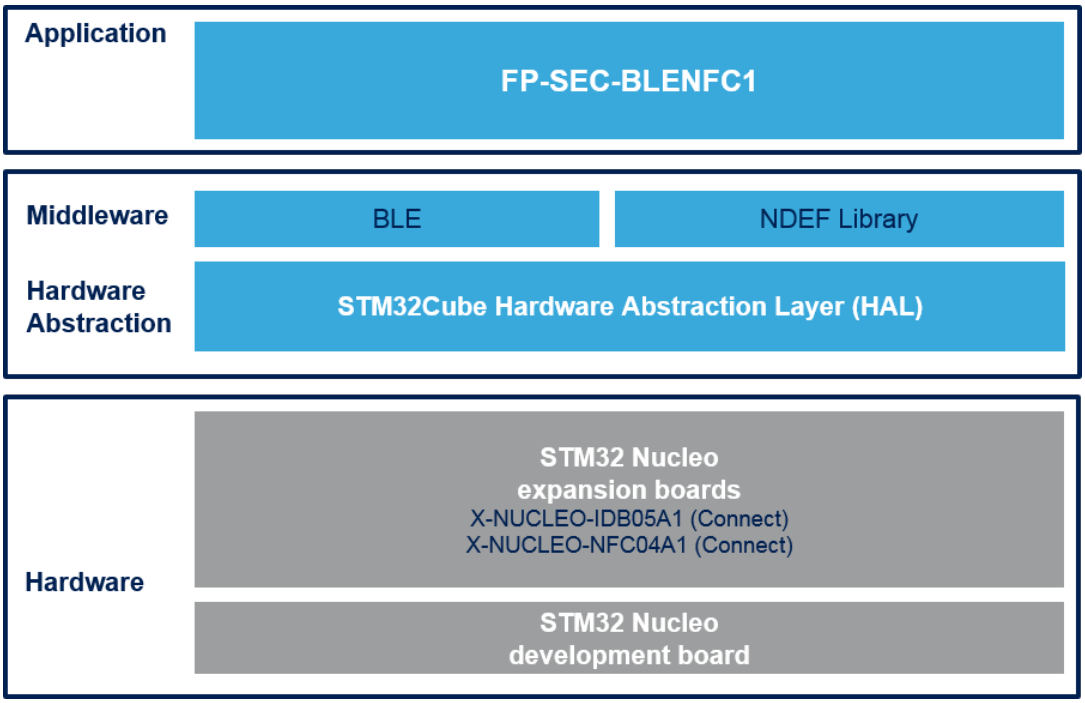
The drivers abstract low-level details of the hardware and allow the middleware components and applications to access to dynamic NFC tag in a hardware-independent manner.

The package also includes a sample application that the developer can use to start experimenting with the code. The sample application was developed to enable NFC pairing and bi-directional communication to a Bluetooth low energy-enabled device, such as a smartphone (Android or iOS-based).

The software layers used by the application software to access and use the Sensors expansion board are:

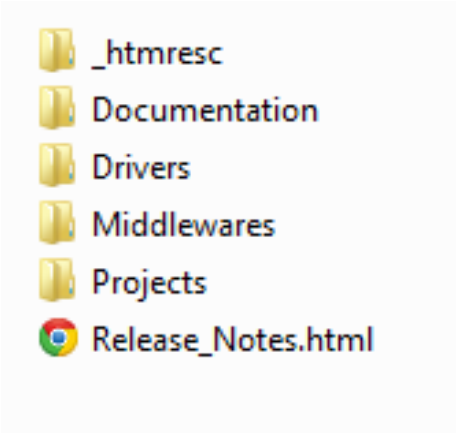
- **STM32Cube HAL layer**: consists of a set of simple, generic, multi-instance APIs (application programming interfaces) which interact with the upper layer applications, libraries and stacks. These generic and extension APIs are based on a common framework which allows any layers they built on, such as the middleware layer, to implement their functions without requiring specific hardware information for a given microcontroller unit (MCU). This structure improves library code reusability and guarantees easy portability across other devices.
- **Board Support Package (BSP) layer**: provides software support for the STM32 Nucleo board peripherals, excluding the MCU. These specific APIs provide a programming interface for certain board specific peripherals (like LEDs, user buttons, etc.) and can also be used to fetch individual board version information. It also provides support for initializing, configuring and reading data.

Figure 1. **FP-SEC-BLENFC1 software architecture**



2.3 Folder structure

Figure 2. **FP-SEC-BLENFC1 package folder structure**



The following folders are included in the software package:

- **Documentation**: contains a compiled HTML file detailing the software components and APIs.
- **Drivers**: 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**: contains libraries and protocols for BlueNRG-MS Bluetooth low energy and NDEF library.
- **Projects**: contains a sample application used for transmitting Console service by using the Bluetooth low energy protocol, provided for the NUCLEO-F401RE/NUCLEO-L053R8 platforms with IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM) and System Workbench for STM32 integrated development environments.

2.4 APIs

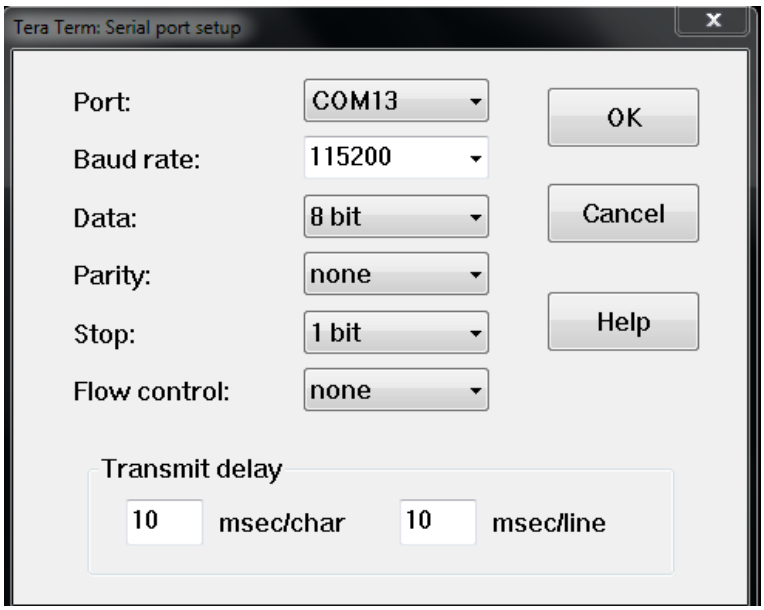
Full function and parameter descriptions for the user APIs can be found in a compiled HTML file in the Documentation folder.

2.5 Sample application description

A sample application using the X-NUCLEO-NFC04A1 and X-NUCLEO-IDB05A1 expansion boards connected to a NUCLEO-F401RE or NUCLEO-L053R8 board is provided in the Projects directory. Ready-to-build projects are available for multiple IDEs.

The user can control application behavior via UART by launching a terminal application and applying the settings below.

Figure 3. Terminal setting



When you first press the reset button, the application starts initializing the interfaces and implements the NDEF protocol to write the www.st.com/stm32ode URI to the ST25DV04K dynamic NFC tag on the X-NUCLEO-NFC04A1 expansion board (see). When an Android device reads the content of the NFC tag, the browser automatically loads and tries to connect to the same URI.

When you press the blue user button (see Figure 5. UART console output when the BLE services are started), the program:

1. initializes the SPI interface used for communicating with the BlueNRG expansion board
2. determines which BlueNRG expansion board X-NUCLEO-IDB05A1 is connected to the STM32 Nucleo board and the hardware and firmware versions
3. creates the unique BLE MAC address and PIN necessary for the connection
4. initializes the BLE console service, configuration service and the service for exposing the hardware features
5. changes the content of the ST25DV04K dynamic NFC tag (using NDEF) in order to write all the information necessary to automatically launch the BlueMS Android application (name of application, BLE advertising data, BLE MAC address and BLE connection PIN).

*Note: When reading the above NFC content **on an Android device only** with the BlueMS application installed, it can automatically launch the application and connect the device with the STM32 Nucleo board, without having to scan for the board or manually insert the PIN.*

The `NFC_SECURE_CONNECTION` define in the Projects\Multi\Applications\BLENFC1\Inc\nfc_config.h file controls whether the STM32 Nucleo board only accepts secure connections (default) or any connection (define is commented), so you don't have to enter the BLE connection PIN for a device to connect to the STM32 Nucleo board.

Figure 4. **Initialization phase**

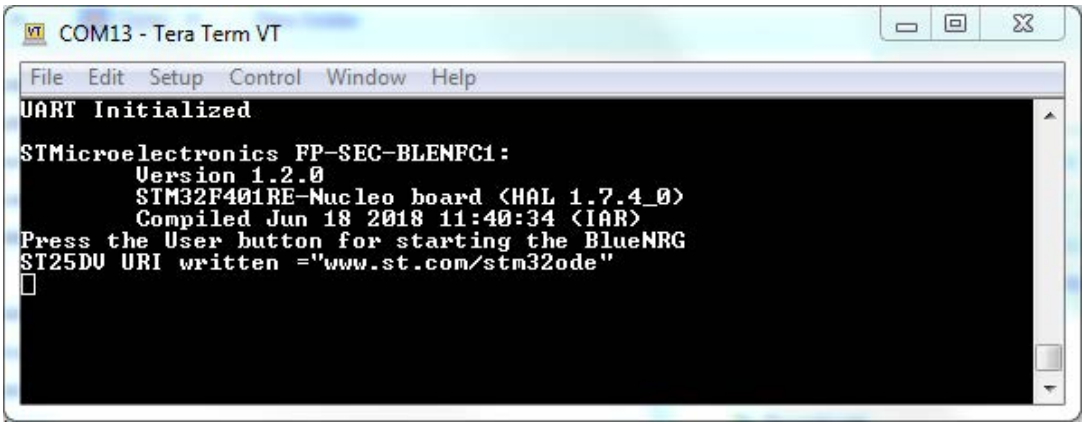
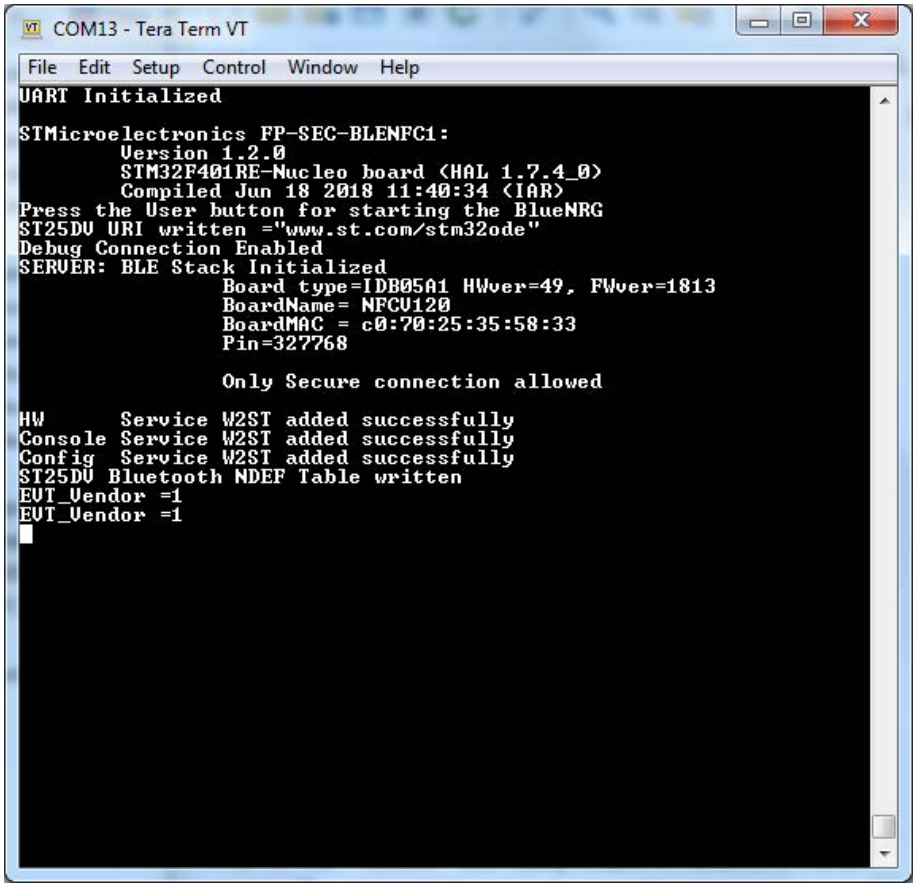
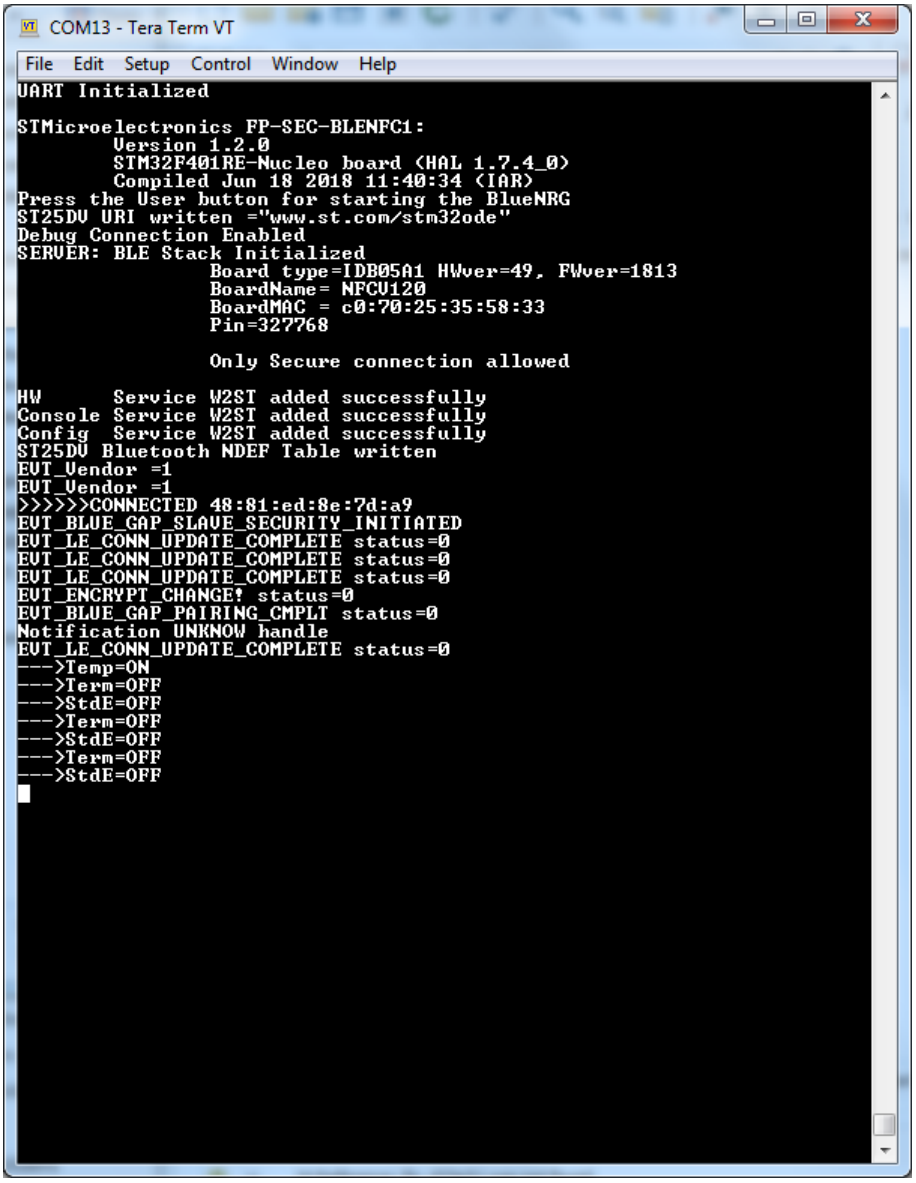


Figure 5. **UART console output when the BLE services are started**



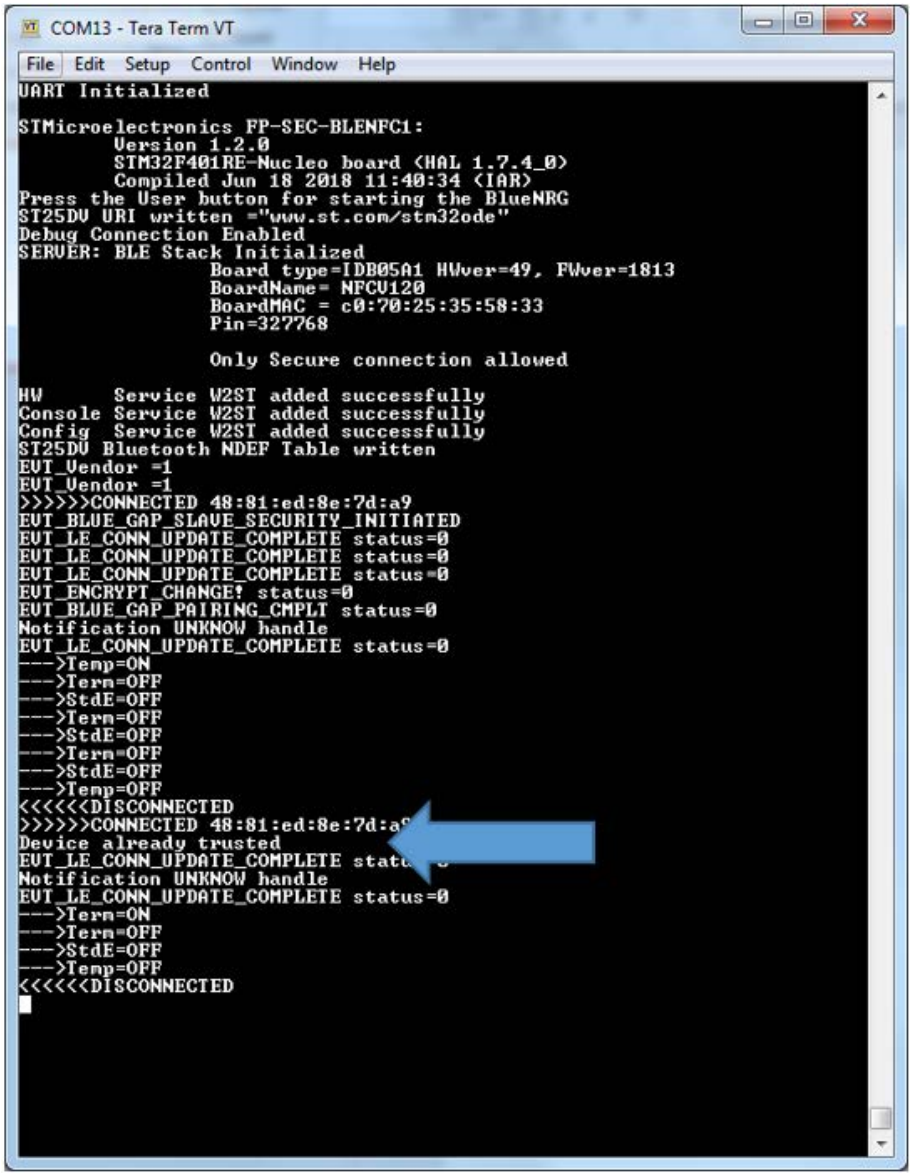
Connection of an Android/iOS device to the STM32 Nucleo board starts with the secure pairing procedure where ping information is sent to the stdout console BLE characteristic.

Figure 6. **UART console output when a device first connects with the board**



The application has a white list of one element, so subsequent connections with the last trusted device are automatically authenticated.

Figure 7. **UART console output when a device is already trusted**



If you are using the NUCLEO-F401RE board and it is currently disconnected, you can press the blue user button to prepare the content of the ST25DV04K dynamic tag for an email reporting information on the latest connection. It contains the BLE MAC address of the device that attempted the last connection and the BLE MAC address of the trusted device. Pressing the blue user button again triggers the information necessary for BLE connection to be written to the ST25DV04K device.

Subsequent pressing of the same button enables switching between the email preparation and BLE connection. In Figure 8. UART console output during multiple connections (allowed and not allowed):

- Initially (green section), the device supplies the right PIN and is connected to the board
- then another device (red section) tries to connect with an incorrect PIN and is rejected
- pressing the user button (blue section) triggers the writing either the BLE connection information or the log email (see Figure 9. Log email prepared on the ST25DV04K dynamic NFC tag) to the ST25DV04K dynamic NFC tag. In the latter case, an Android device will automatically prepare the email to be sent.

Figure 8. **UART console output during multiple connections (allowed and not allowed)**

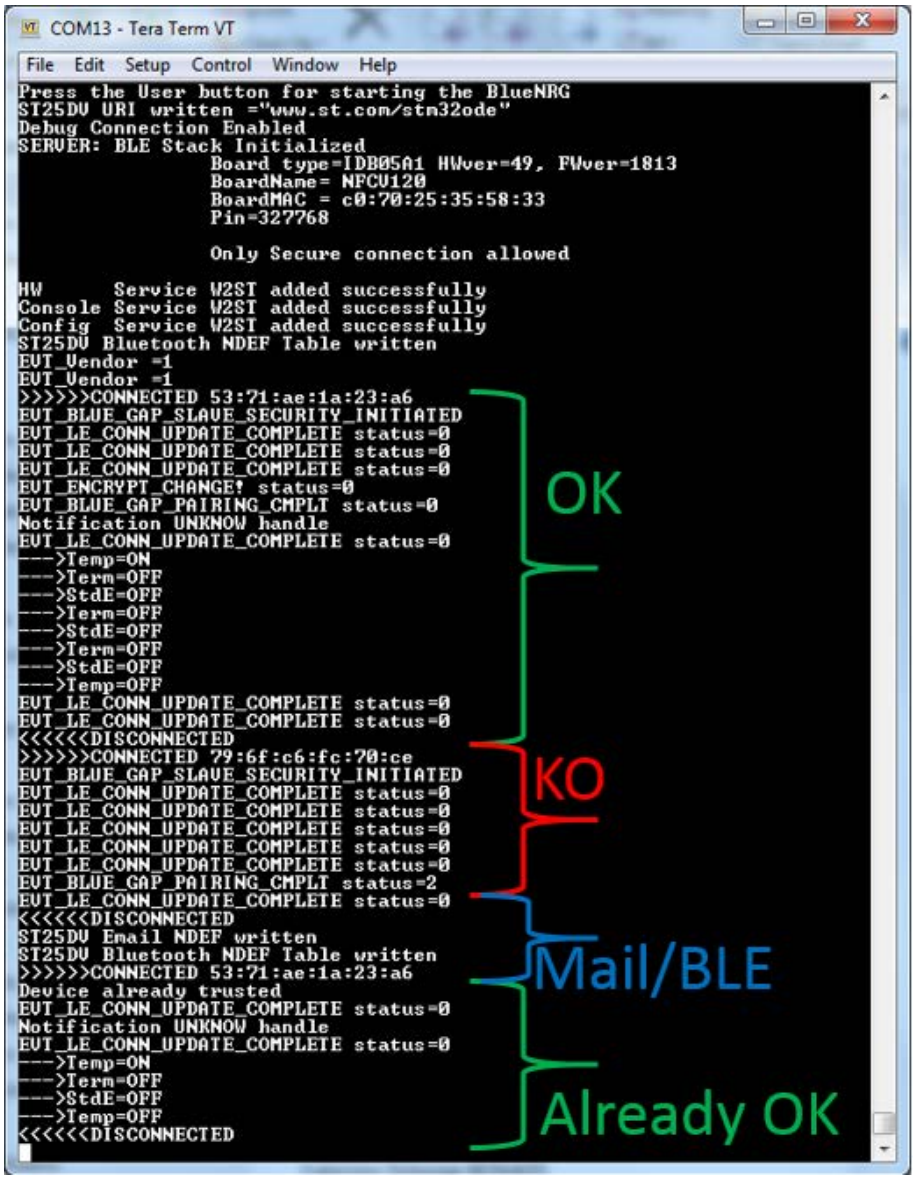
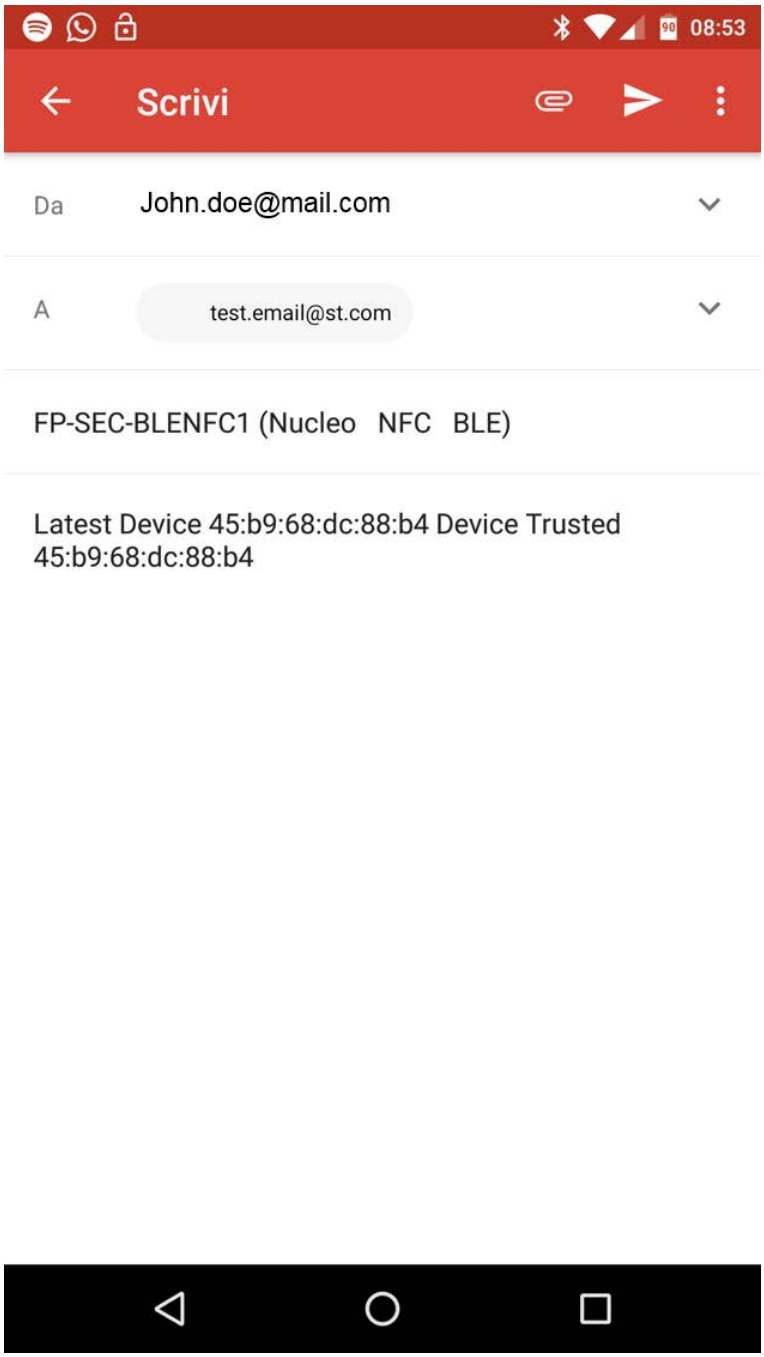


Figure 9. **Log email prepared on the ST25DV04K dynamic NFC tag**

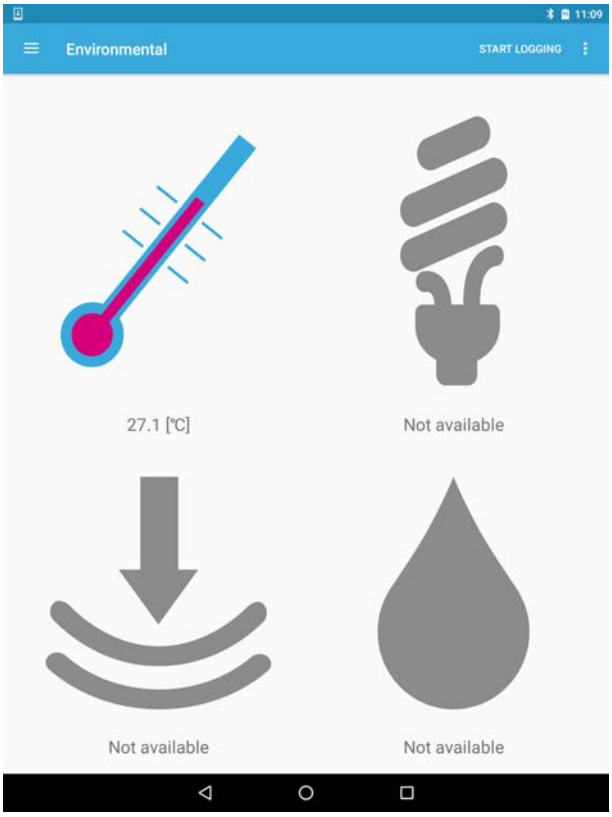


2.6 Android and iOS sample client application

The FP-SEC-BLENFC1 software for STM32Cube is compatible with the BlueMS Android/iOS applications (Version 2.1.0 and above), available at the respective Play/iOS stores. We are using the Android version to demonstrate how the application works.

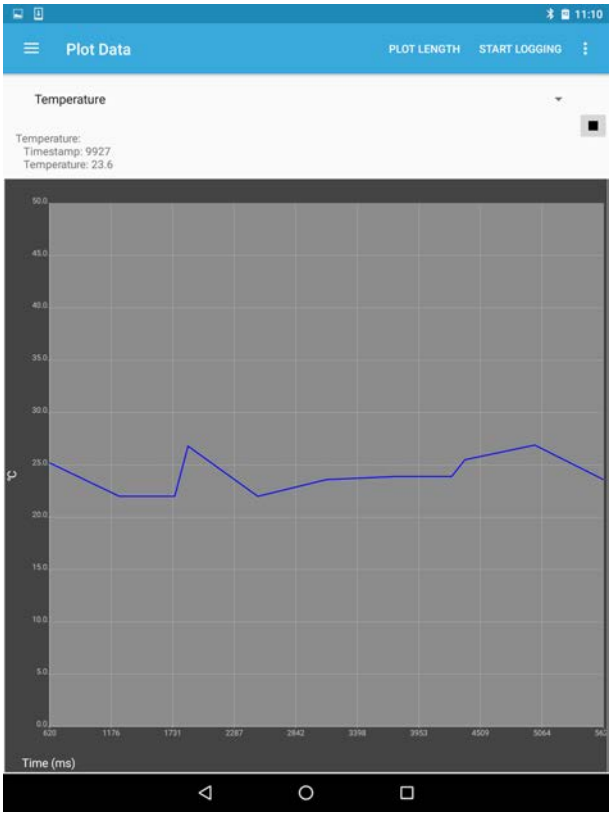
Step 1. Following connection, BlueMS opens the page below.
You will see the value of the simulated temperature sensor transmitted via BLE.

Figure 10. **BlueMS (Android version) initial page after BLE connection**



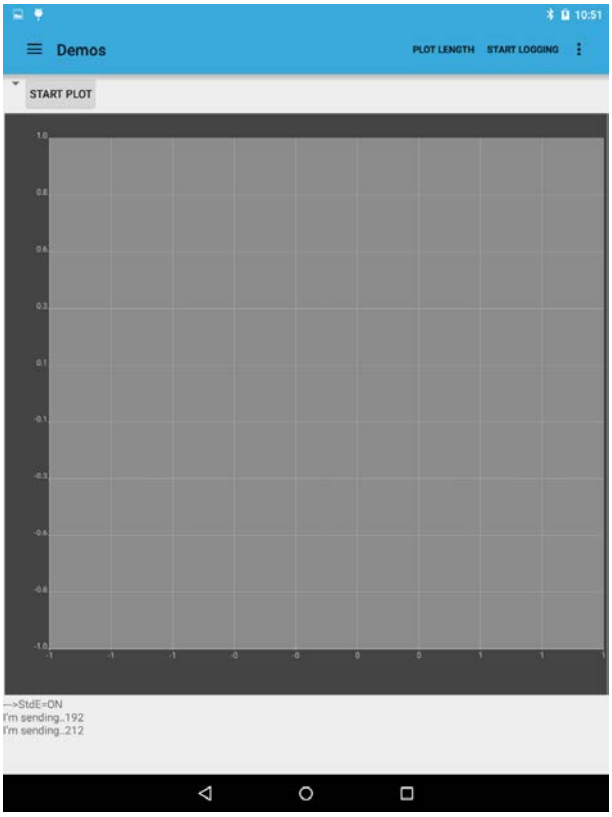
Step 2. On the next page is possible to plot the simulated temperature sensor.

Figure 11. **BlueMS (Android version) plot page**



Step 3. Launch the Serial Console from the overflow menu.
It reads the stdout/stderr characteristics.

Figure 12. **BlueMS (Android version) Serial console (stdout/stderr)**

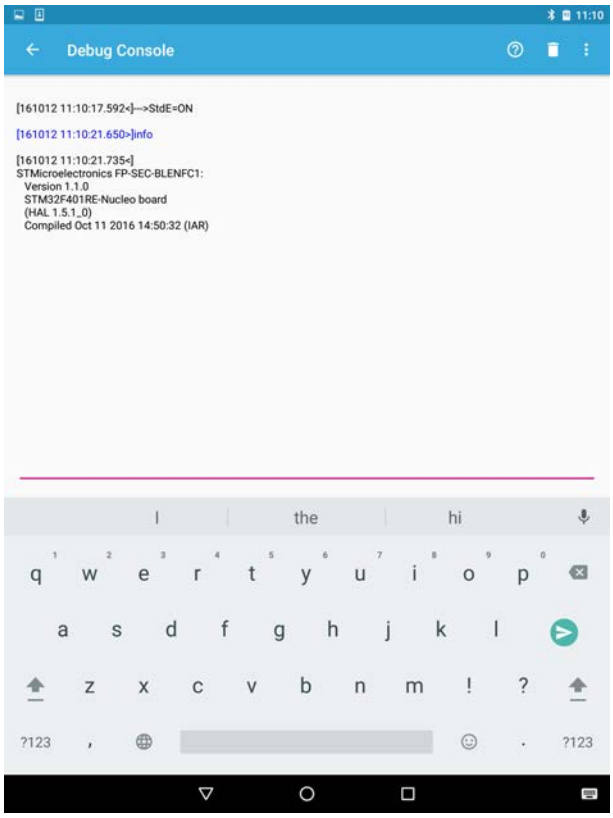


Step 4. Launch the Debug Console from the overflow menu and the "info" command.
The board will return:

- Functional pack name and version
- STM32 Nucleo board model
- Hardware abstraction layer version
- When it was compiled, with which IDE

If you enter something different, the board replies with the same message.

Figure 13. **BlueMS (Android version) Debug console (stdin/stdout/stderr)**



Step 5. Switch the LED on the STM32 Nucleo board on and off in the Switch status page.

Figure 14. **BlueMS (Android version) Switch status page**

3 System setup guide

3.1 Hardware description

3.1.1 STM32 Nucleo platform

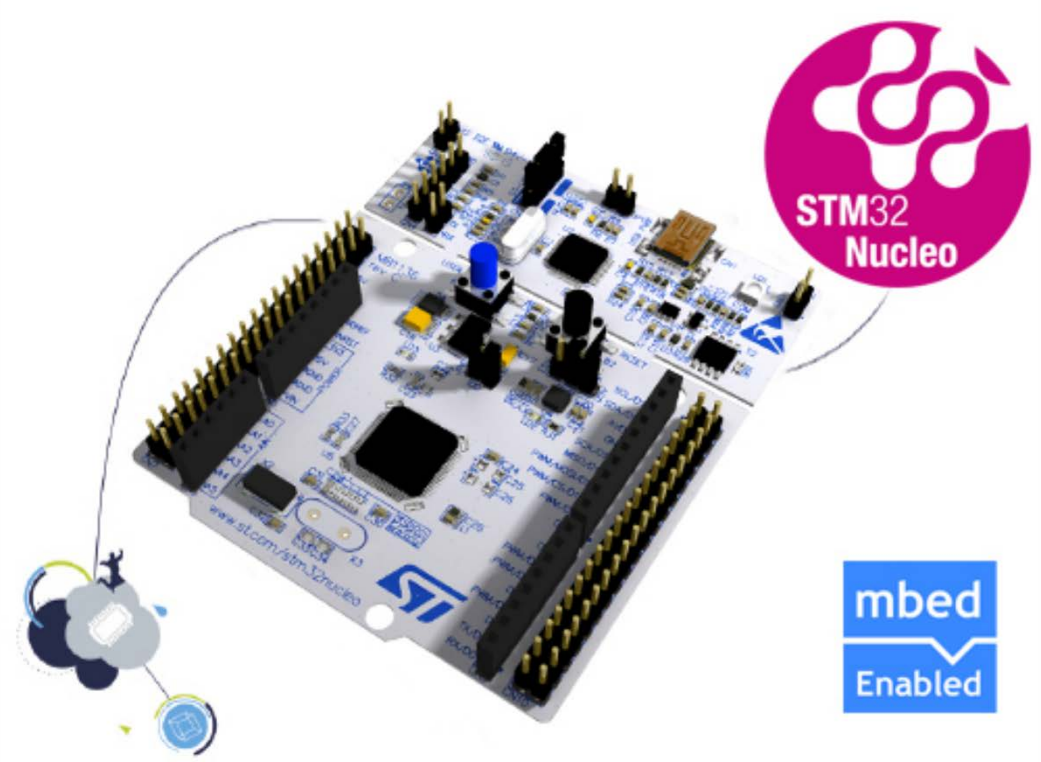
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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 15. STM32 Nucleo board

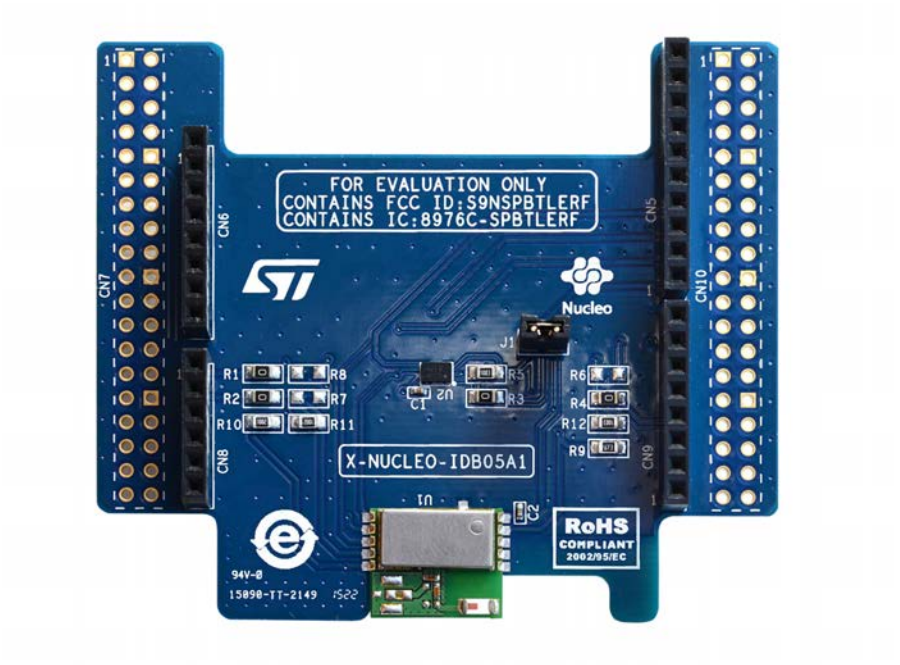


Information regarding the STM32 Nucleo board is available at www.st.com/stm32nucleo

3.1.2 X-NUCLEO-IDB05A1 expansion board

The X-NUCLEO-IDB05A1 is a Bluetooth low energy expansion board based on the SPBTLE-RF BlueNRG-MS RF module to allow expansion of the STM32 Nucleo boards. The SPBTLE-RF module is FCC (FCC ID: S9NSPBTLERF) and IC certified (IC: 8976C-SPBTLERF). The BlueNRG-MS is a very low power Bluetooth low energy (BLE) single-mode network processor, compliant with Bluetooth specification v4.2. X-NUCLEO-IDB05A1 is compatible with the ST morpho and Arduino™ UNO R3 connector layout. This expansion board can be plugged into the Arduino UNO R3 connectors of any STM32 Nucleo board.

Figure 16. **X-NUCLEO-IDB05A1 expansion board**



Information about the X-NUCLEO-IDB05A1 expansion board is available on www.st.com at http://www.st.com/x-nucleo

3.1.3 X-NUCLEO-NFC04A1 expansion board

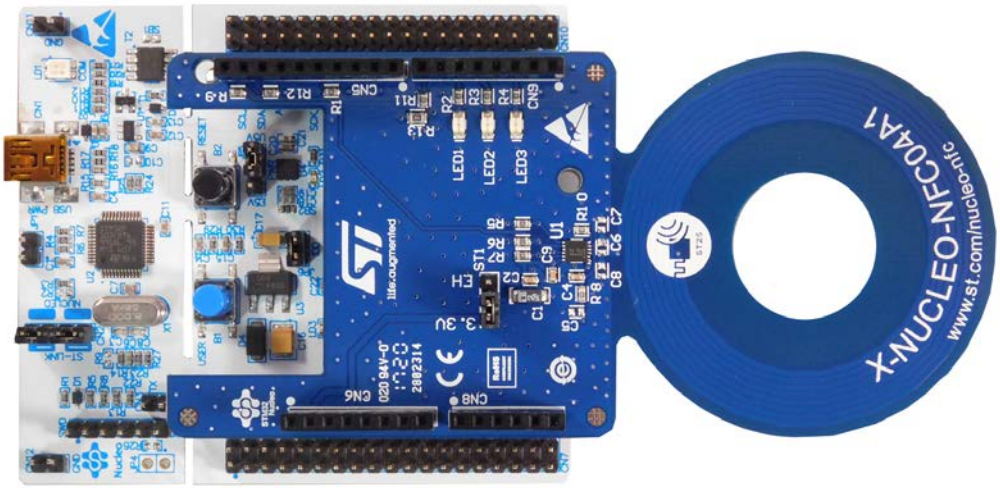
The X-NUCLEO-NFC04A1 dynamic NFC/RFID tag IC expansion board is based on the ST25DV04K NFC Type V/ RFID tag IC with a dual interface 4 Kbits EEPROM that also features an I²C interface. It can be powered by the pin of Arduino connector or directly by the received carrier electromagnetic field.

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 17. **X-NUCLEO-NFC04A1 expansion board**

Figure 18. **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 at http://www.st.com/x-nucleo

3.2 Software description

The following software components are needed in order to set up a suitable development environment for creating applications for the STM32 Nucleo equipped with the NFC and BlueNRG expansion boards:

- FP-SEC-BLENFC1: a Bluetooth Low Energy and Dynamic NFC tag software for STM32Cube
- Development tool-chain and Compiler: the STM32Cube expansion software supports the following environments:
  - IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
  - System Workbench for STM32 + ST-LINK

3.3 Hardware and software setup

3.3.1 Hardware setup

The following hardware components are needed:

1. One STM32 Nucleo development platform (order code: NUCLEO-F401RE or NUCLEO-L053R8)
2. One NFC expansion board (order code: X-NUCLEO-NFC04A1)
3. One BlueNRG Bluetooth low energy expansion board (order code: X-NUCLEO-IDB05A1)
4. One USB type A to Mini-B USB cable to connect the STM32 Nucleo board to the PC

3.3.2 Software setup

3.3.2.1 *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.

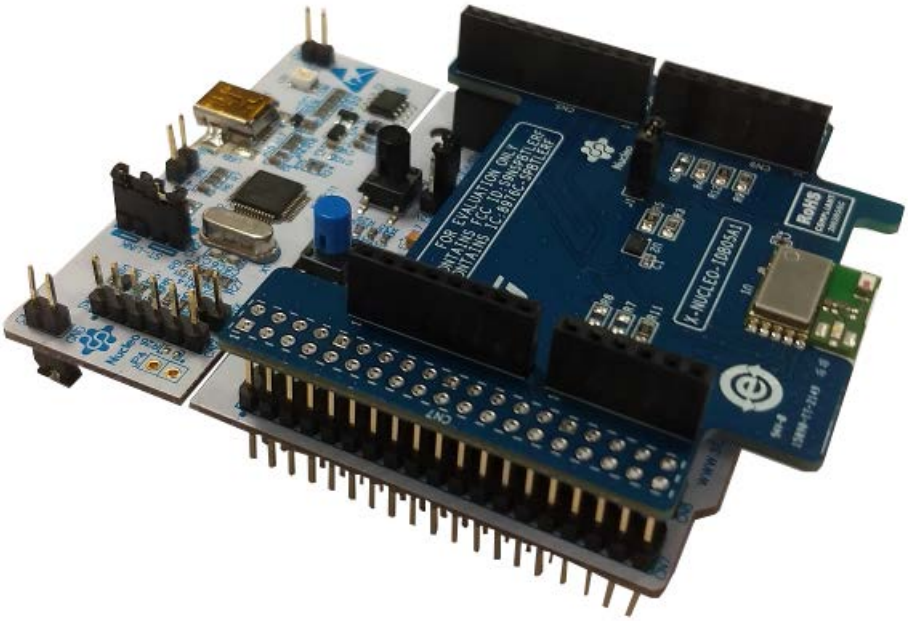
3.3.3 System setup guide

3.3.3.1 *STM32 Nucleo and sensor 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-LINK008 or STSW-LINK009 on www.st.com (based on the Microsoft Windows operating system).

The BlueNRG X-NUCLEO-IDB05A1 expansion board is easily connected to the STM32 Nucleo board through the Arduino UNO R3 extension connector.

Figure 19. STM32 Nucleo development board plus X-NUCLEO-IDB05A1 expansion board

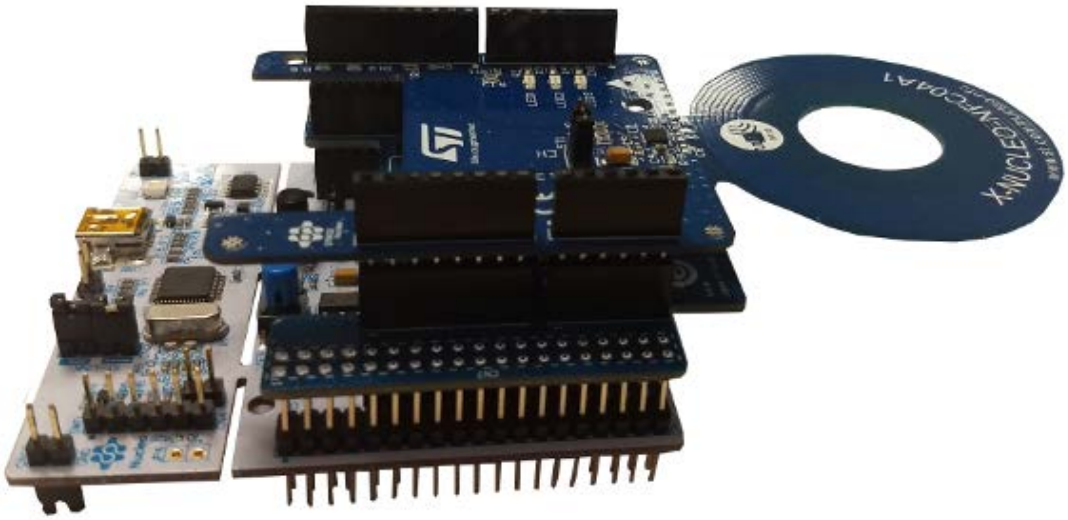


The Dynamic NFC tag board X-NUCLEO-NFC04A1 can be easily connected to X-NUCLEO-IDB05A1 expansion board through the Arduino UNO R3 extension connector.

Note: Due to the hardware conflict with the X-NUCLEO-IDB05A1, it is necessary to remove the 0 ohm resistors, R1 and R11, on the X-NUCLEO-NFC04A1.

Figure 20. Modifications required for the X-NUCLEO-NFC04A1 expansion board

Figure 21. **STM32 Nucleo development board plus X-NUCLEO-IDB05A1 and X-NUCLEO-NFC04A1 ST25DV4K dynamic NFC tag expansion boards**



Revision history

Table 2. **Document revision history**

Date	Version	Changes
02-Feb-2016	1	Initial release.
28-Jun-2018	2	Throughout document: - updated content to reflect release V1.2.0 of the FP-SEC-BLENFC1 software package; - replaced X-NUCLEO-NFC01A1 with X-NUCLEO-NFC04A1 compatibility information; - removed references to the X-NUCLEO-IDB04A1. Updated Introduction, Section 2.1 Overview , Section 2.2 Architecture and Section 3.3.3.1 STM32 Nucleo and sensor expansion board setup.

Contents

List of figures

List of tables