UM2028
User manual

Getting started with FP-SEC-WIFINFC1 WI-FI and Dynamic NFC tag software expansion for STM32Cube

Introduction

Developers can use the FP-SEC-WIFINFC1 software expansion package for STM32Cube to negotiate secure Wi-Fi connections on any NFC-capable phone; Wi-Fi credentials are stored on a dynamic NFC tag device, in compliance with the NDEF standard.

The expansion is built on STM32Cube software technology to ease portability across different STM32 microcontrollers. The software runs on the STM32 microcontroller and includes drivers for the ST Wi-Fi module (SPWF01SA) and dynamic NFC tag device (M24SR64-Y).

A sample implementation of the drivers running on the X-NUCLEO-IDW01M1 and the X- NUCLEO-NFC01A1 expansion boards plugged on top of a NUCLEO-F401RE STM32 Nucleo board is bundled with the package.

Information regarding STM32Cube is available on www.st.com at http://www.st.com/stm32cube

Contents

List of tables

List of figures

1 Acronyms and abbreviations

Table 1: Acronyms and abbreviations

Acronym	Description
NFC	Near field communication
NDEF	NFC data exchange format
SSID	Service set identifier

2 FP-SEC-WIFINFC1 software description

2.1 Overview

The key features of the FP-SEC-WIFINFC1 package are:

- Complete middleware to build applications using Wi-Fi connectivity (SPWF01SA Serial-to-Wi-Fi Module) and M24SR64-Y dynamic NFC/RFID tag using NDEF standard
- The package is compatible with the motion sensor LSM6DS3 DIL24 expansion component
- Easy portability across different MCU families, thanks to STM32Cube
- Wi-Fi network pairing parameters such as SSID and password can be written by any NFC-capable phone and can be read by the application software to connect with any wireless network.
- Sample implementation available on X-NUCLEO-NFC01A1 and X-NUCLEO-IDW01M1 boards when plugged onto NUCLEO-F401RE board

This software prepares the NFC hardware to be able to write the NDEF Wi-Fi pairing message from an NFC-capable phone and then connect to a wireless network via the Wi-Fi expansion board.

This package is compatible with the Android M24SR application.

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 Wi-Fi and the dynamic NFC tag expansion boards
- Middleware components to enable communication with other Wireless networks and store secure Wi-Fi pairing information on an NFC/RFID tag. This information is compliant with the NDEF standard and can be read/written by any NFC-capable device.

The drivers abstract low-level hardware details so that middleware components and applications can access the dynamic NFC tag device in a hardware-independent manner.

The package includes a sample application that the developer can use to start experimenting with the code. For this purpose, the sample application was developed to enable Wi-Fi - NFC pairing and secure wireless network connection.

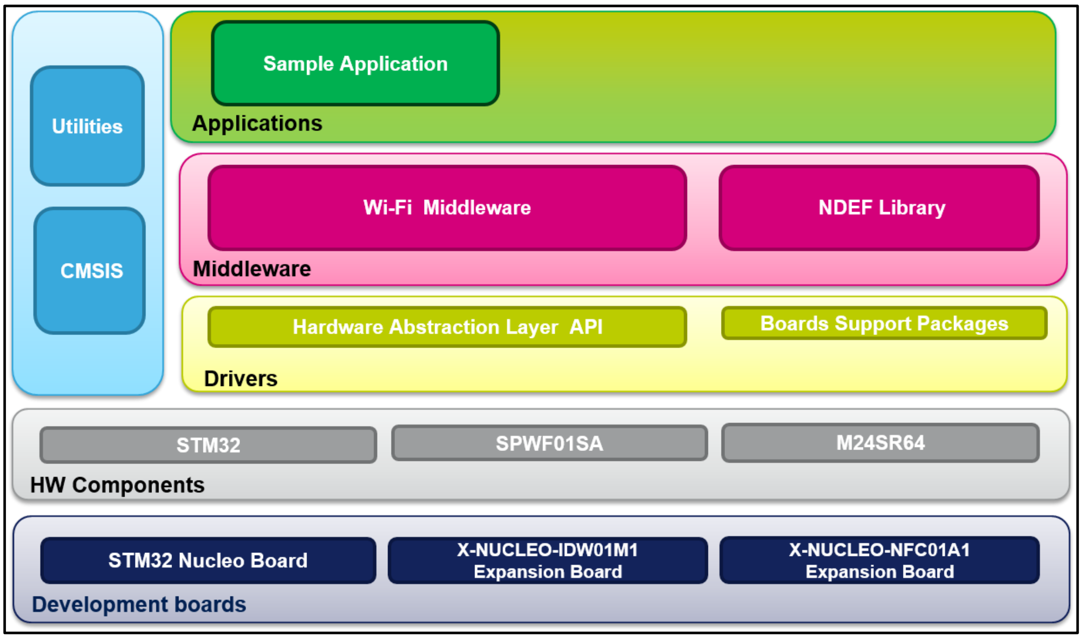
The software layers used by the application software to access and use the Wi-Fi and NFC expansion boards are:

- **STM32Cube HAL layer**: consists of a simple, generic, multi-instance set of APIs (application programming interfaces) which interact with upper layer applications, libraries and stacks. These generic and extension APIs are based on a common framework so that any layers they are built on, such as the middleware layer, can 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**: the software package supports the peripherals on the STM32 Nucleo board (apart from the MCU) through the board support package (BSP). This is a limited set of APIs which provides a programming interface for certain

board-specific peripherals like the LED and the user button. This interface also helps in identifying the specific board version.

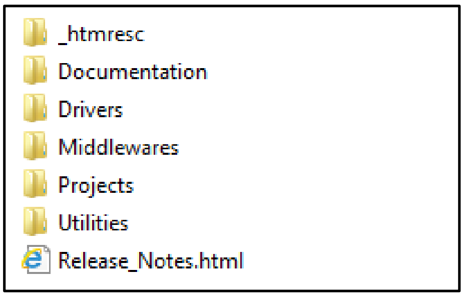
- **Middleware**: the middleware provides the software for Wi-Fi and NFC hardware for applications to interact with them. It also contains the NDEF library for reading and writing the the dynamic NFC tag data.

Figure 1: FP-SEC-WIFINFC1 software architecture



2.3 Folders structure

Figure 2: FP-SEC-WIFINFC1 package folders structure



_htmresc
Documentation
Drivers
Middlewares
Projects
Utilities
Release_Notes.html

The following folders are included in the software package:

- Documentation: this folder contains a compiled HTML file generated from the source code, detailing 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 libraries and protocols related to the NFC NDEF Library for Wi-Fi pairing.
- Projects: this folder contains a sample application used for connecting to wireless networks using information stored in the NFC tag; it is provided for the NUCLEO-F401RE platform in the IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDKARM), and System Workbench for STM32 development environments

- Utilities: this folder contains the Android source code installer for the NDEF record writing/reading sample application.

2.4 APIs

Detailed technical information about the available APIs can be found in a compiled HTML file located inside the package "Documentation" folder, where all the functions and parameters are fully described.

`TT4_WriteWifiToken()` writes the network SSID, network Key, authentication type and encryption type Wi-Fi credentials to the NFC tag.

`TT4_ReadWifiToken()` reads the same Wi-Fi credentials from the NFC tag.

The NDEF message format is shown in the following table.

Table 2: Wi-Fi NDEF format

Offset (Octets)	Content	Length (Octets)	Explanation
0	0xD2	1	NDEF record header (MB=1b, ME=1b, CF=0b, SR=1b, IL=0b, TNF=010b)
1	0x17	1	Payload type name length: 23 octets
2	0x56	1	Payload length: 86 octets
3	0x61 0x70 0x70 0x6C 0x69 0x63 0x61 0x74 0x69 0x6F 0x6E 0x2F 0x76 0x6E 0x64 0x2E 0x77 0x66 0x61 0x2E 0x77 0x73 0x63	23	Payload type name: "application/vnd.wfa.wsc"
26	0x10 0x4A	2	Attribute ID: Version
28	0x00 0x01	2	Attribute Length: 1 octet
30	0x10	1	Version: 1.0
31	0x10 0x0E	2	Attribute ID: Credential
33	0x00 0x43	2	Attribute Length: 67 octets
35	0x10 0x26	2	Attribute ID: Network Index
37	0x00 0x01	2	Attribute Length: 1 octet
39	0x01	1	Network Index: 1
40	0x10 0x45	2	Attribute ID: SSID
42	0x00 0x08	2	Attribute Length: 8 octets
44	0x48 0x6F 0x6D 0x65 0x57 0x4C 0x41 0x4E	8	SSID: "HomeWLAN"
52	0x10 0x03	2	Attribute ID: Authentication Type
54	0x00 0x02	2	Attribute Length: 2 octets
56	0x00 0x20	2	Authentication Type: WPA2-Personal
58	0x10 0x0F	2	Attribute ID: Encryption Type
60	0x00 0x02	2	Attribute Length: 2 octets
62	0x00 0x08	2	Encryption Type: AES

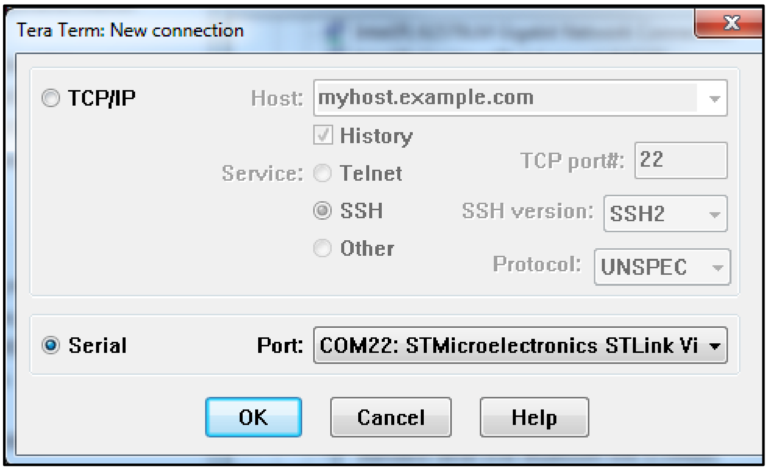
Offset (Octets)	Content	Length (Octets)	Explanation
64	0x10 0x27	2	Attribute ID: Network Key
66	0x00 0x0E	2	Attribute Length: 14 octets
68	0x4D 0x79 0x50 0x72 0x65 0x53 0x68 0x61 0x72 0x65 0x64 0x4B 0x65 0x79	14	Network Key: “MyPreSharedKey”
82	0x10 0x20	2	Attribute ID: MAC Address
84	0x00 0x06	2	Attribute Length: 6 octets
86	0xFF 0xFF 0xFF 0xFF 0xFF 0xFF	6	MAC Address: “FF:FF:FF:FF:FF:FF”
92	0x10 0x49	2	Attribute ID: Vendor Extension
94	0x00 0x06	2	Attribute Length: 6 octets
96	0x00 0x37 0x2A	3	Vendor ID: WFA (0x00372A)
99	0x02	1	Subelement ID: Network Key Shareable
100	0x01	1	Subelement Length: 1 octet
101	0x01	1	Network Key Shareable: TRUE
102	0x10 0x49	2	Attribute ID: Vendor Extension
104	0x00 0x06	2	Attribute Length: 6 octets
106	0x00 0x37 0x2A	3	Vendor ID: WFA (0x00372A)
109	0x00	1	Subelement ID: Version2
110	0x01	1	Subelement Length: 1 octet
111	0x20	1	Version2: 2.0

2.5 Sample application description

A sample application using the X-NUCLEO-NFC01A1 and X-NUCLEO-IDW01M1 expansion boards connected to a NUCLEO-F401RE board is provided in the “Projects/ Multi/Applications/Nfc_WiFi_Connect” directory. Ready-to-build projects are available for multiple IDEs.

The user can monitor all application behavior via UART by launching a terminal application and setting the UART port to 115200 bps, 8 bit, No Parity, 1 stop bit.

Figure 3: Tera Term setup



Tera Term: New connection

○ TCP/IP

Host: myhost.example.com

☑ History

Service: ○ Telnet

● SSH

○ Other

TCP port#: 22

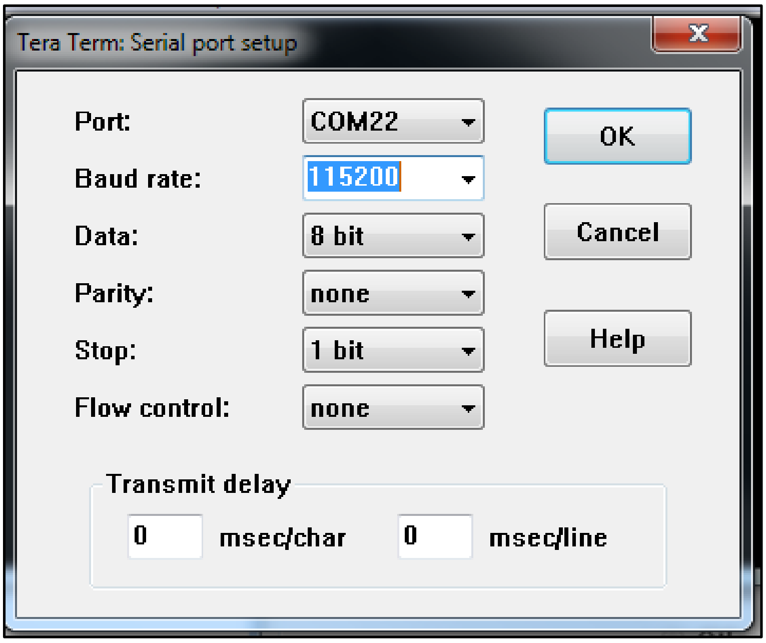
SSH version: SSH2

Protocol: UNSPEC

● Serial

Port: COM22: STMicroelectronics STLink Vi

Figure 4: Serial port setup



Tera Term: Serial port setup

Port: COM22

Baud rate: 115200

Data: 8 bit

Parity: none

Stop: 1 bit

Flow control: none

Transmit delay

0 msec/char 0 msec/line

Once you have set up the system and connected to a PC as per *Section 3: "System setup guide"*, the application starts initializing the M24SR64-Y dynamic NFC tag on the X-NUCLEO-NFC01A1 expansion board and Wi-Fi Module on X-NUCLEO-IDW01M1. The yellow (LED3) and Blue (LED2) LEDs light up to signal correct initialization of the board.

2.6 Android sample client application

The FP-SEC-WIFINFC1 software package for STM32Cube is compatible with the ST “M24SR Demo” application available from the Google Play store. Install it on an NFC-capable Android phone and launch it. The source code for this application can be downloaded from *www.st.com*.

The application is used for writing the SSID and SSID Key to the NFC tag as shown below.

Figure 5: M24SR demo application

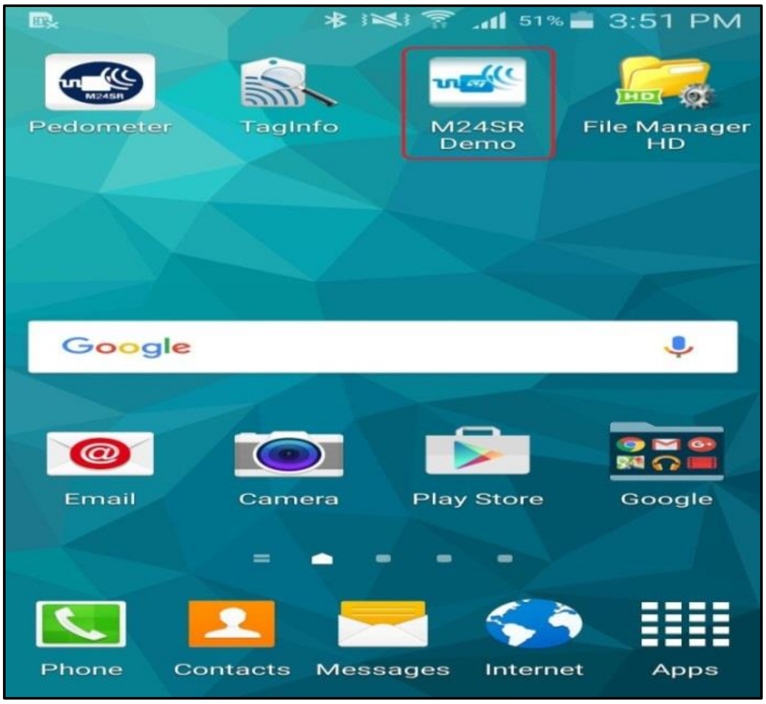
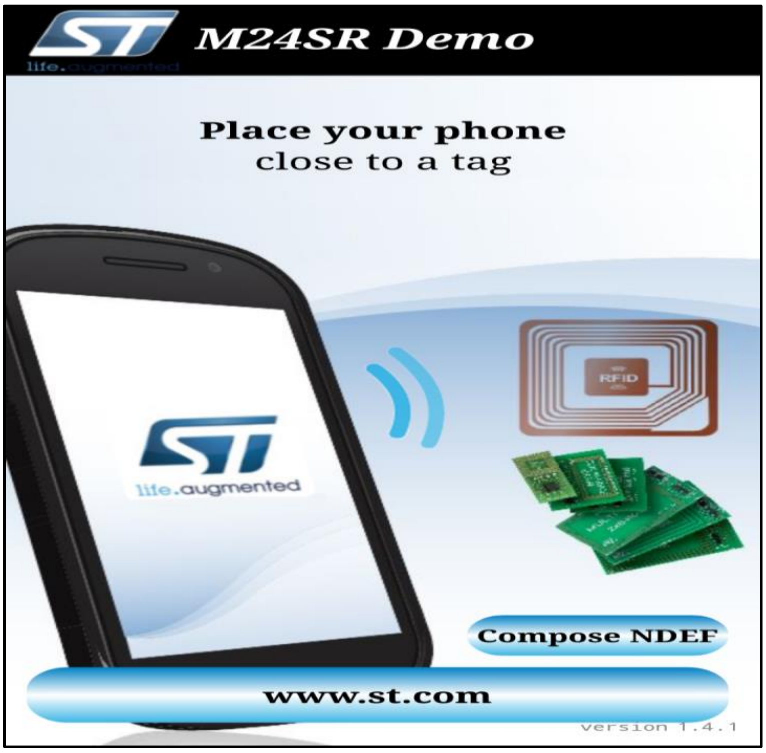
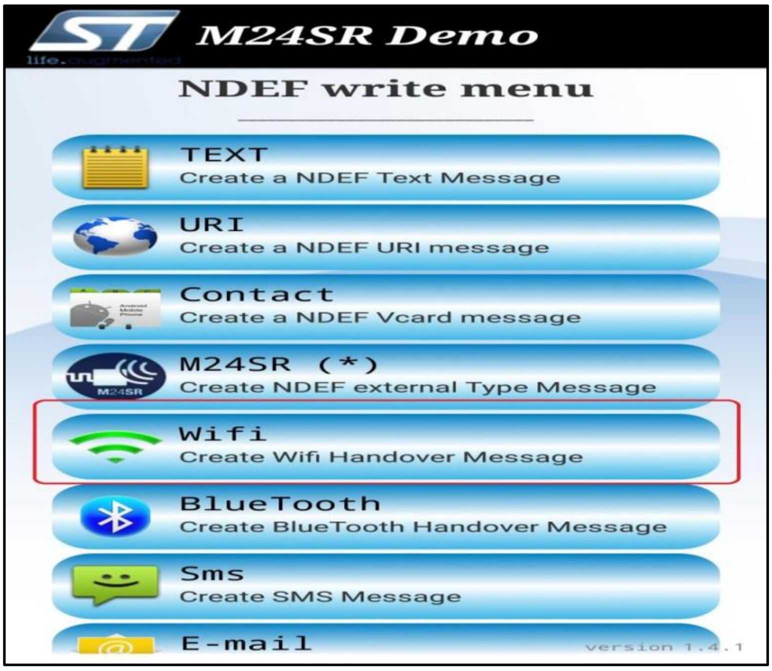


Figure 6: M24SR DEmo application launch screen



You will obtain the default tag information from the NFC board when your phone, with above application running, nears the NFC tag. Click “NDEF Editor” or “Compose NDEF” and click Wi-Fi (Create Wi-Fi Handover Message)

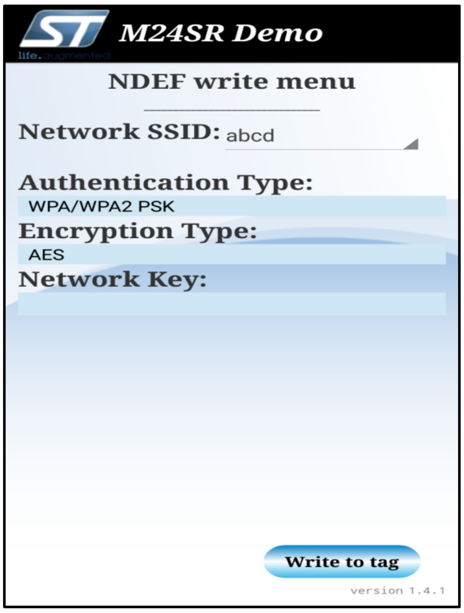
Figure 7: Create Wi-Fi NDEF pairing message



M24SR Demo
life.augmented
NDEF write menu
TEXT
Create a NDEF Text Message
URI
Create a NDEF URI message
Contact
Create a NDEF Vcard message
M24SR (*)
Create NDEF external Type Message
Wifi
Create Wifi Handover Message
BlueTooth
Create BlueTooth Handover Message
Sms
Create SMS Message
E-mail
version 1.4.1

Select the Network SSID from the dropdown menu to find the network or hotspot you want to connect to, write the corresponding network key and then press Write to Tag. Leave Authentication Type and Encryption Type as shown below.

Figure 8: Fill the required SSID and Network Key fields



M24SR Demo

NDEF write menu

Network SSID: abcd

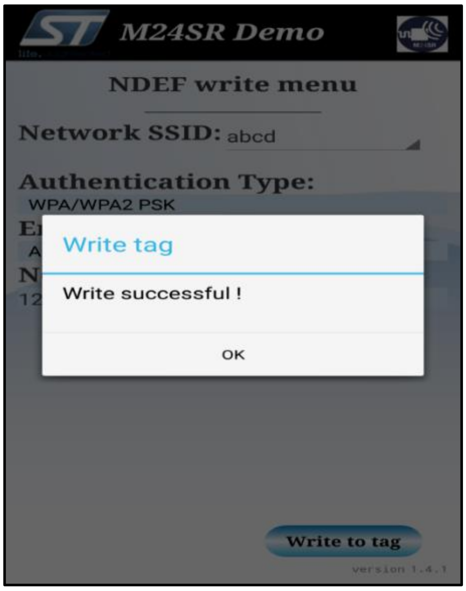
Authentication Type: WPA/WPA2 PSK

Encryption Type: AES

Network Key:

Once the tag information is written successfully, the following message will appear.

Figure 9: NDEF tag write confirmation



NDEF write menu

Network SSID: abcd

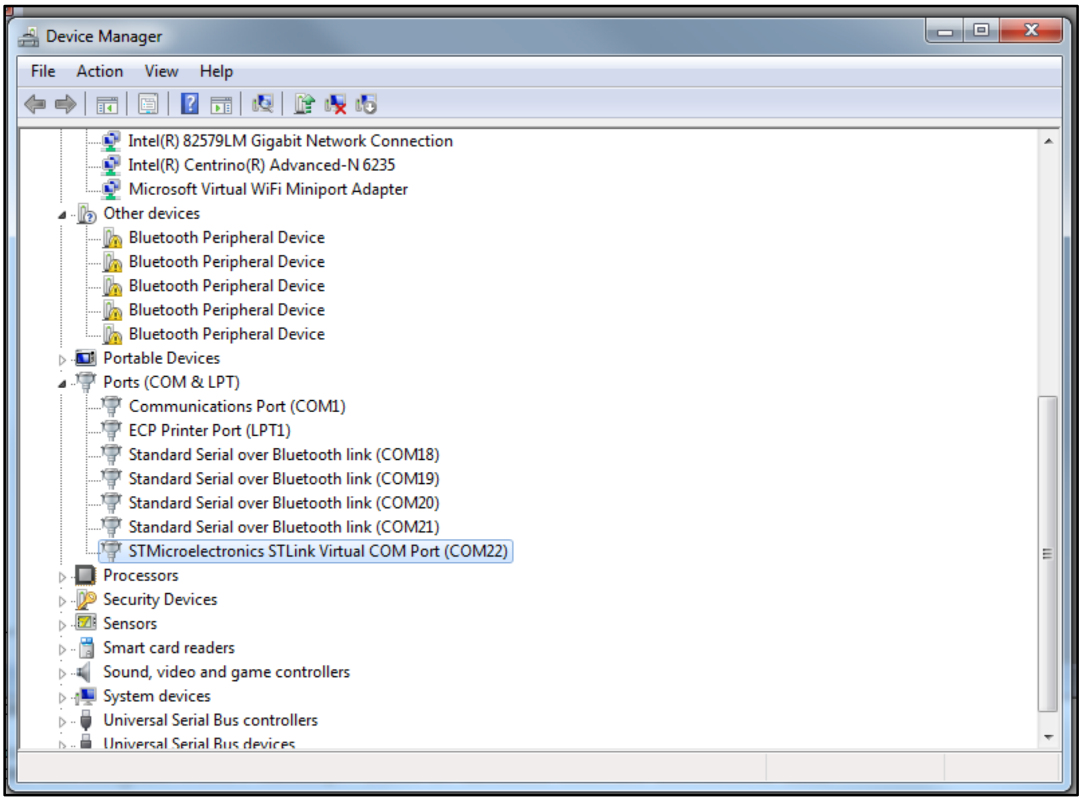
Authentication Type:
WPA/WPA2 PSK

Write tag

Write successful !

Open a Hyperterminal-like application (Hercules_3-2-5 or Tera Term) and create a COM connection with the settings shown in *serialPORTsetup*. Choose the COM port according to the port assignment in device manager; in our example, it’s COM22.

Figure 10: Com port selection from the device manager



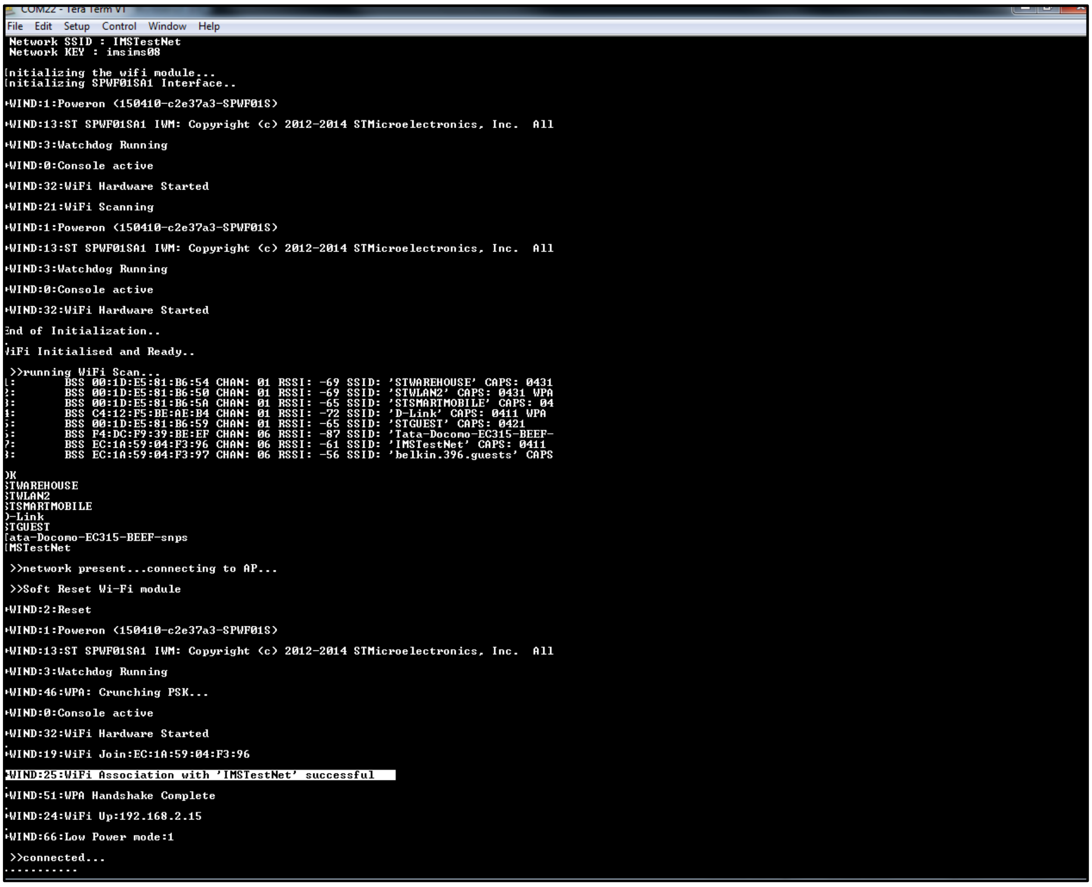
Intel(R) 82579LM Gigabit Network Connection
Intel(R) Centrino(R) Advanced-N 6235
Microsoft Virtual WiFi Miniport Adapter
Other devices
Bluetooth Peripheral Device
Bluetooth Peripheral Device
Bluetooth Peripheral Device
Bluetooth Peripheral Device
Bluetooth Peripheral Device
Portable Devices
Ports (COM & LPT)
Communications Port (COM1)
ECP Printer Port (LPT1)
Standard Serial over Bluetooth link (COM18)
Standard Serial over Bluetooth link (COM19)
Standard Serial over Bluetooth link (COM20)
Standard Serial over Bluetooth link (COM21)
STMicroelectronics STLink Virtual COM Port (COM22)
Processors
Security Devices
Sensors
Smart card readers
Sound, video and game controllers
System devices
Universal Serial Bus controllers
Universal Serial Bus devices

Once this is done, reset the board by pressing the power on button.

On successful connection with the network or hotspot, you will see the log messages as in the image below. Here the SSID is **ImsTestNet** and SSID Key is **imsims08**.

You can connect to any network or hotspot whose tag information you have written via the Android application (SSID and SSID Key).

Figure 11: sample successful connection log



```
 Network SSID : IMSTestNet
 Network KEY : imsims08

Initializing the wifi module...
Initializing SPWF01SA1 Interface..

+WIND:1:Poweron (150410-c2e37a3-SPWF01S)

+WIND:13:ST SPWF01SA1 IWM: Copyright (c) 2012-2014 STMicroelectronics, Inc.  All

+WIND:3:Watchdog Running

+WIND:0:Console active

+WIND:32:WiFi Hardware Started

+WIND:21:WiFi Scanning

+WIND:1:Poweron (150410-c2e37a3-SPWF01S)

+WIND:13:ST SPWF01SA1 IWM: Copyright (c) 2012-2014 STMicroelectronics, Inc.  All

+WIND:3:Watchdog Running

+WIND:0:Console active

+WIND:32:WiFi Hardware Started

End of Initialization..

WiFi Initialised and Ready..

 >>running WiFi Scan...
1:       BSS 00:1D:E5:81:B6:54 CHAN: 01 RSSI: -69 SSID: 'STWAREHOUSE' CAPS: 0431
2:       BSS 00:1D:E5:81:B6:50 CHAN: 01 RSSI: -69 SSID: 'STWLAN2' CAPS: 0431 WPA
3:       BSS 00:1D:E5:81:B6:5A CHAN: 01 RSSI: -65 SSID: 'STSMARTMOBILE' CAPS: 04
4:       BSS C4:12:F5:BE:AE:B4 CHAN: 01 RSSI: -72 SSID: 'D-Link' CAPS: 0411 WPA
5:       BSS 00:1D:E5:81:B6:59 CHAN: 01 RSSI: -65 SSID: 'STGUEST' CAPS: 0421
6:       BSS F4:DC:F9:39:BE:EF CHAN: 06 RSSI: -87 SSID: 'Tata-Docomo-EC315-BEEF-
7:       BSS EC:1A:59:04:F3:96 CHAN: 06 RSSI: -61 SSID: 'IMSTestNet' CAPS: 0411
8:       BSS EC:1A:59:04:F3:97 CHAN: 06 RSSI: -56 SSID: 'belkin.396.guests' CAPS

OK
STWAREHOUSE
STWLAN2
STSMARTMOBILE
D-Link
STGUEST
Tata-Docomo-EC315-BEEF-snps
IMSTestNet

 >>network present...connecting to AP...

 >>Soft Reset Wi-Fi module

+WIND:2:Reset

+WIND:1:Poweron (150410-c2e37a3-SPWF01S)

+WIND:13:ST SPWF01SA1 IWM: Copyright (c) 2012-2014 STMicroelectronics, Inc.  All

+WIND:3:Watchdog Running

+WIND:46:WPA: Crunching PSK...

+WIND:0:Console active

+WIND:32:WiFi Hardware Started

+WIND:19:WiFi Join:EC:1A:59:04:F3:96

+WIND:25:WiFi Association with 'IMSTestNet' successful

+WIND:51:WPA Handshake Complete

+WIND:24:WiFi Up:192.168.2.15

+WIND:66:Low Power mode:1

 >>connected...
...........
```

3 System setup guide

3.1 Hardware description

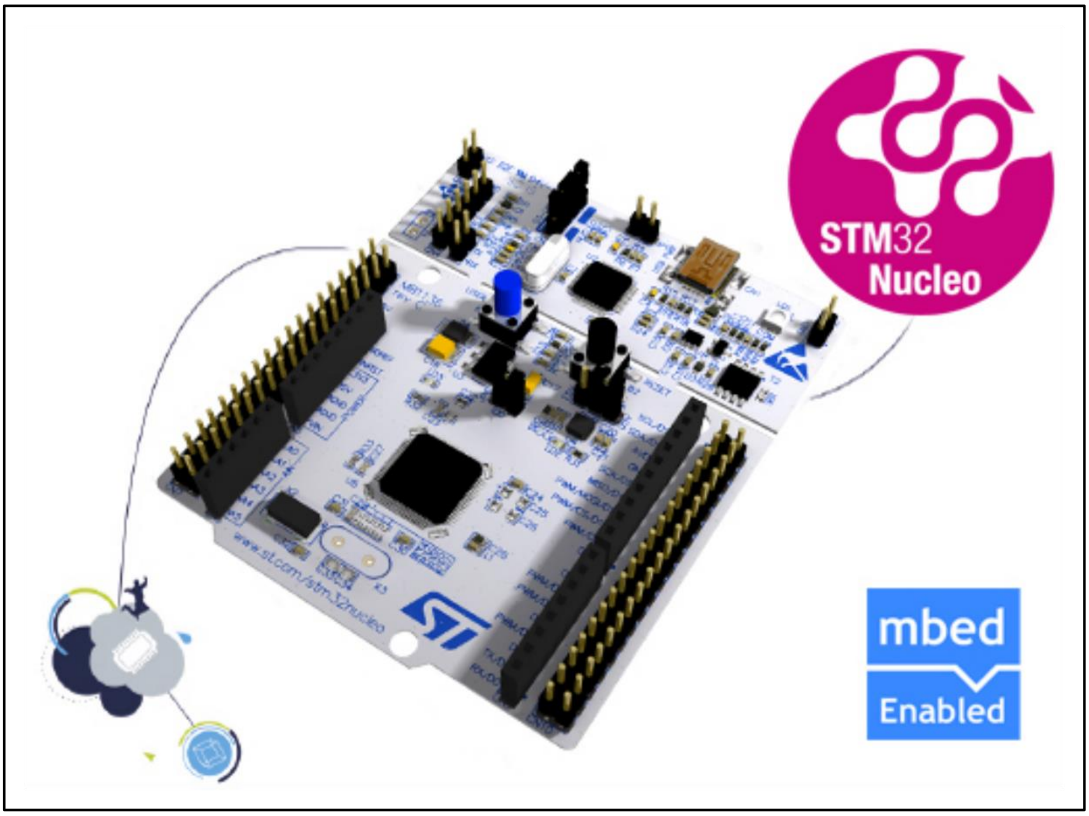
This section describes the hardware components needed for the development of Wi-Fi and NFC module-based applications.

3.1.1 STM32 Nucleo platform

The STM32 Nucleo boards provide an affordable and flexible way for users to try out 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 range of specialized expansion boards to choose from. 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 comprehensive STM32 software HAL library together with various packaged software examples.

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

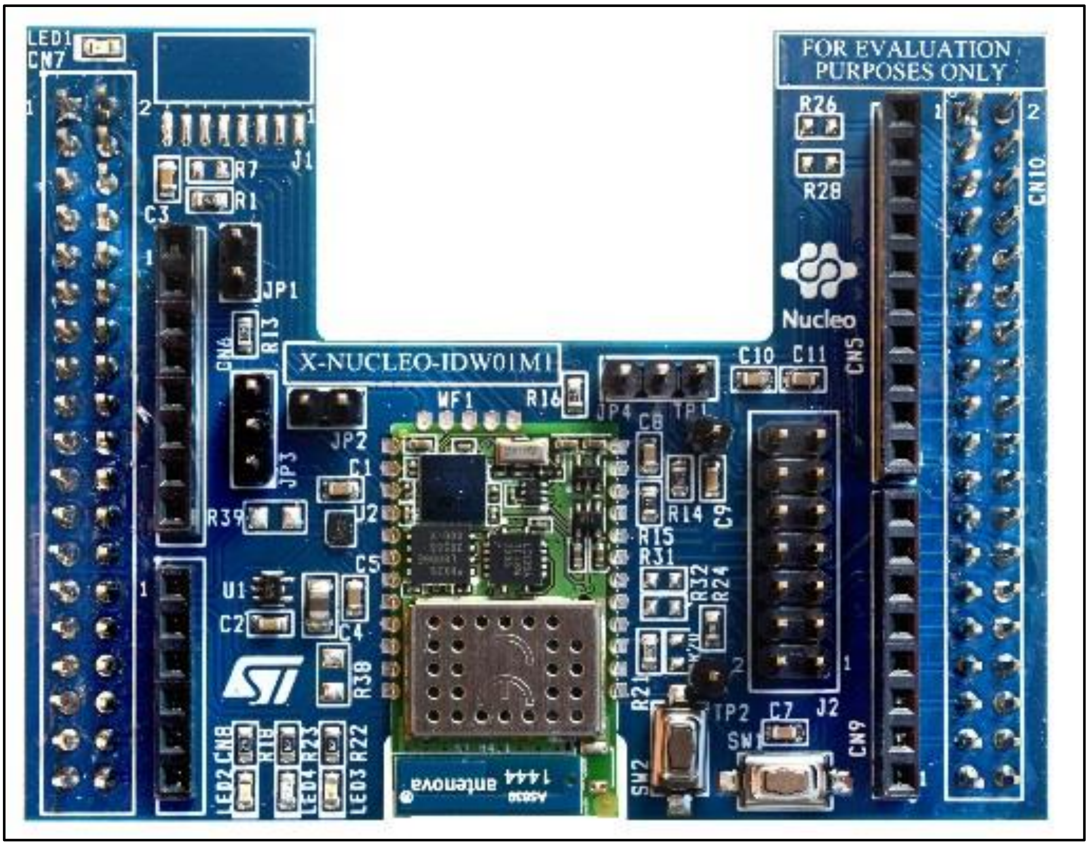
Figure 12: STM32 Nucleo board



3.1.2 X-NUCLEO-IDW01M1 expansion board

The X-NUCLEO-IDW01M1 is an expansion board based on the SPWF01SA device.

Figure 13: X-NUCLEO-IDW01M1 Wi-Fi expansion board



Information regarding the expansion board is available on www.st.com at *http://www.st.com/x-nucleo*.

3.1.3 X-NUCLEO-NFC01A1 expansion board

The X-NUCLEO-NFC01A1 is an expansion board based on the M24SR64-Y device. This expansion board can be plugged on the Arduino UNO R3 connectors of any STM32 Nucleo board.

The M24SR64-Y device is a dynamic NFC/RFID tag IC with a dual interface. It embeds 64 Kbit EEPROM memory, and can be operated from:

- an I²C interface
- a 13.56 MHz RFID reader or a NFC phone.

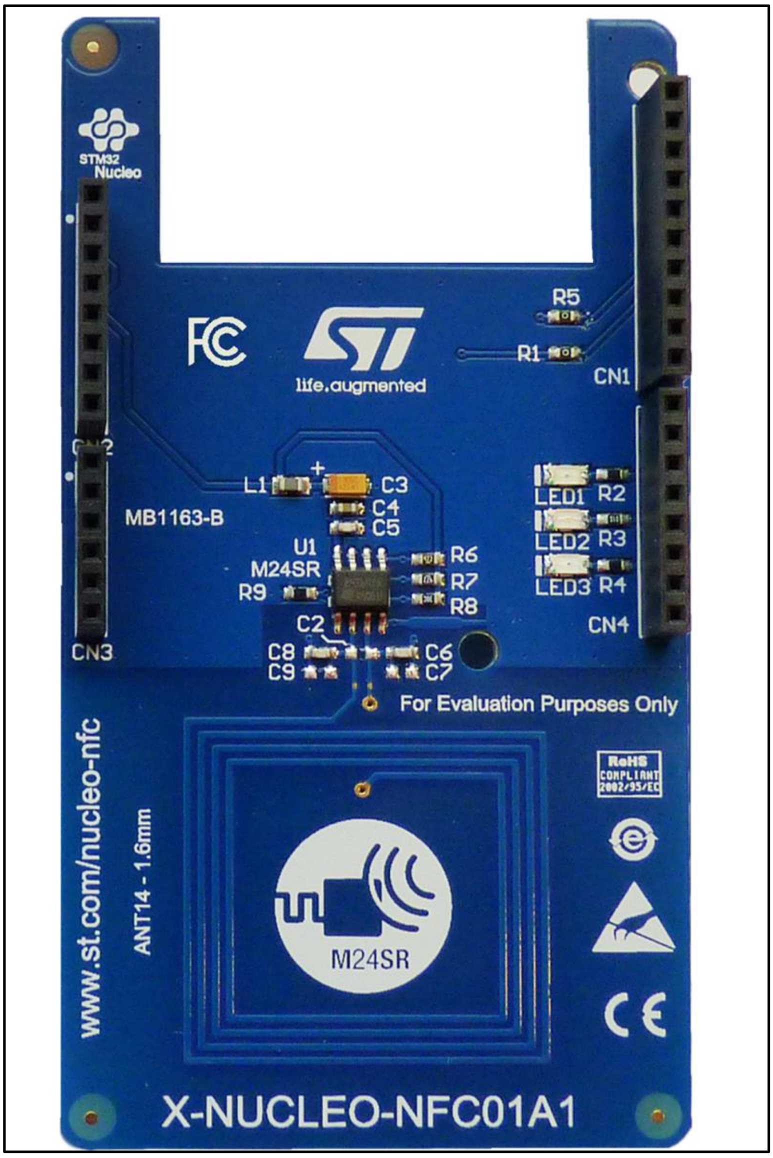
The I²C interface uses a two-wire serial interface, consisting of a bidirectional data line and a clock line. It behaves as a slave with respect to the I²C protocol.

The RF protocol is compatible with:

- ISO/IEC 14443 Type A
- NFC Forum Type 4 Tag.

The board is powered through the Arduino UNO R3 connectors and includes three general purpose LEDs.

Figure 14: X-NUCLEO-NFC01A1 M24SR64-Y dynamic NFC tag expansion board



Information regarding the X-NUCLEO-NFC01A1 expansion board is available on www.st.com at *http://www.st.com/x-nucleo*

3.2 Software description

The following software components are required in order to set up a suitable development environment for creating applications for the STM32 Nucleo board equipped with the NFC and Wi-Fi expansion boards:

- The FP-SEC-WIFINFC1 Wi-Fi and Dynamic NFC tag software for STM32Cube; the FP-SEC-WIFINFC1 firmware and related documentation is available on www.st.com
- 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 (SW4STM32) + ST-LINK

3.3 Hardware and software setup

This section describes the hardware and software setup procedures. It also describes the system setup needed for the above.

3.3.1 Hardware setup

The following hardware components are needed:

One STM32 Nucleo development platform (order code: NUCLEO-F401RE)

One NFC expansion board (order code: X-NUCLEO-NFC01A1)

One Wi-Fi expansion board (order code: X-NUCLEO-IDW01M1)

One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC

3.3.2 Software setup

This section lists the minimum requirements for the developer to set up the SDK, run the sample testing scenario based on the GUI utility and customize applications.

3.3.2.1 Development tool-chains and compilers

Select one of the integrated development environments supported by the STM32Cube expansion software. Currently, IAR 7.40 is used for this setup.

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

3.3.2.2 PC utility

Tera Term or Hercules to view logs.

3.3.3 System setup guide

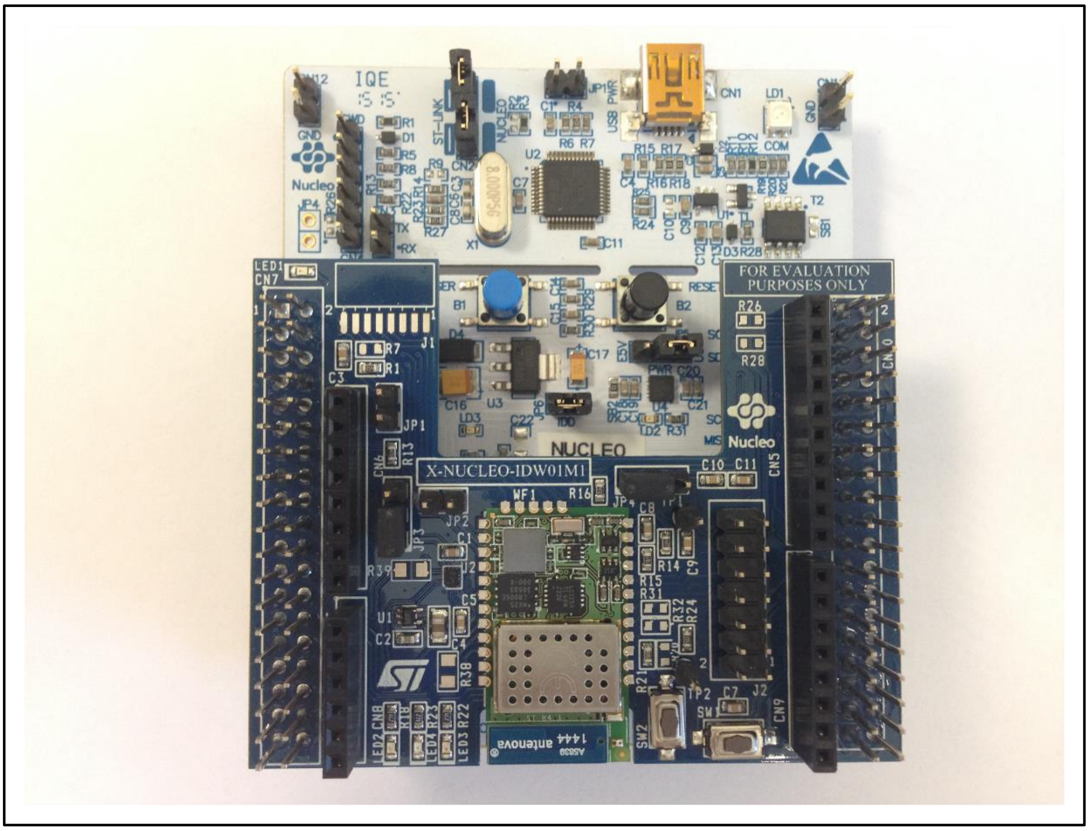
This section describes how to setup different hardware parts before writing and executing an application on the STM32 Nucleo board with the NFC and Wi-Fi expansion board.

3.3.3.1 STM32 Nucleo and Wi-Fi and NCF expansion boards 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 looking STSW-LINK008 or STSW-LINK009 on www.st.com (depending on your your Windows version).

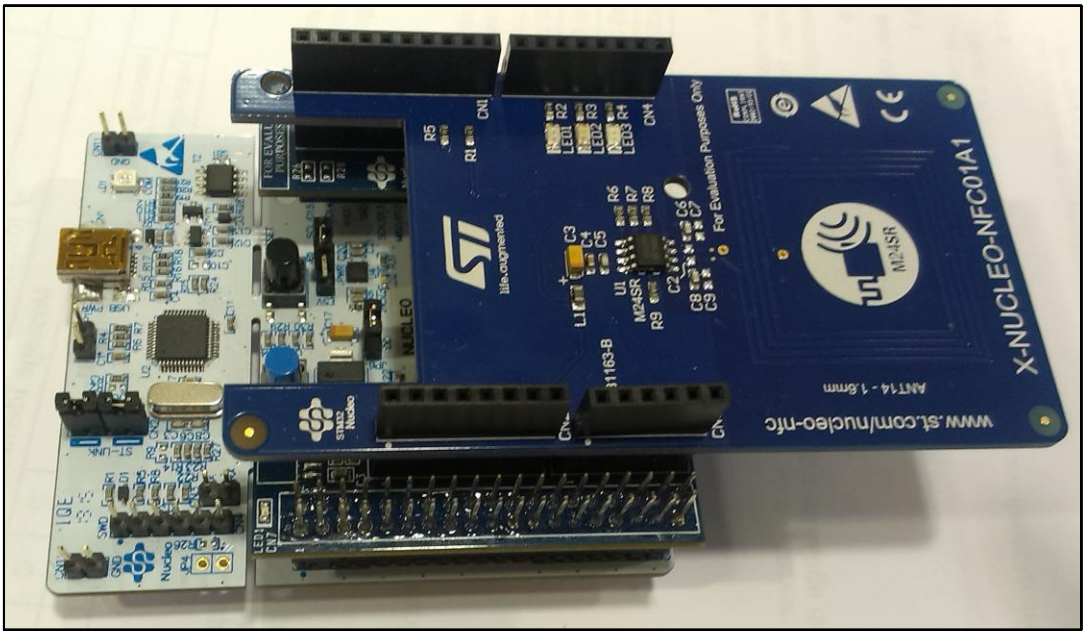
The X-NUCLEO-IDW01M1 expansion board is easily connected to the STM32 Nucleo board through the Arduino UNO R3 extension, as per the figure below.

Figure 15: X-NUCLEO-IDW01M1 expansion board on STM32 Nucleo board



The X-NUCLEO-NFC01A1 board is then plugged on top of the X-NUCLEO-IDW01M1 expansion board, as shown below.

Figure 16: X-NUCLEO-NFC01A1 expansion board on X-NUCLEO-IDW01M1 expansion board on STM32 Nucleo board



4 Revision history

Table 3: Document revision history

Date	Version	Changes
02-Mar-2016	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.