AN4624
Application note

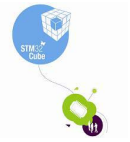
Getting started with the STM32 Nucleo and the M24SR expansion board X-NUCLEO-NFC01A1

Introduction

This document describes how to develop a M24SR based application using the STM32 Nucleo platform and the M24SR expansion board (X-NUCLEO-NFC01A1), within the STM32Cube software environment (X-CUBE-NFC1). The M24SR expansion board is a dynamic NFC tag board with a 64kbits M24SR device and 3 LEDs.

Table 1. Applicable firmware and tool

Type	Part number
Firmware	X-CUBE-NFC1
Tool	X-NUCLEO-NFC01A1



Contents

List of tables

List of figures

1 References

[1]: STMicroelectronics databrief X-CUBE-NFC1, “STM32Cube expansion software for X-NUCLEO-NFC01A1 Expansion Board”.

[2] STMicroelectronics databrief X-NUCLEO-NFC01A1, “Dynamic NFC tag expansion board based on M24SR for STM32 Nucleo”.

2 Acronyms and abbreviations

Table 2. Acronyms and abbreviations

Acronym	Description
BSP	Board support package
GUI	Graphical user interface
HAL	Hardware abstraction layer
I^2C	Inter-integrated circuit
IDE	Integrated development environment
LED	Light emitting diode

3 Getting started

3.1 Hardware description

This section describes the hardware components needed for developing a M24SR based application.

The X-NUCLEO-NFC01A1 expansion board is compatible with all the STM32 Nucleo boards. In this document the STM32F401RE Nucleo board is used as an example.

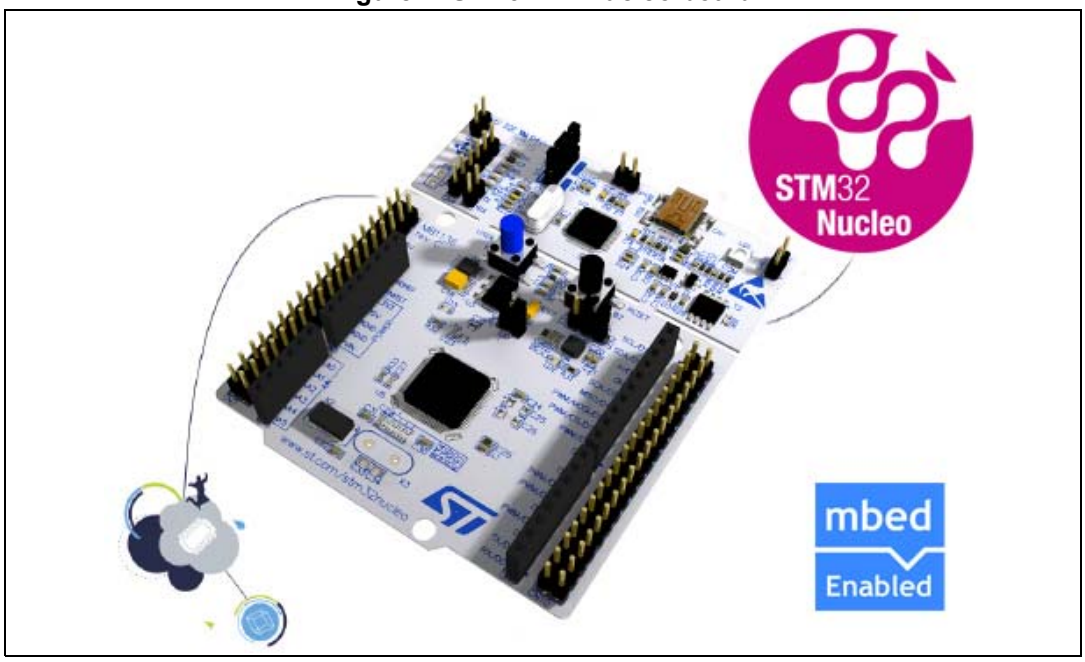
The following sub-sections describe the individual components.

3.1.1 STM32F401RE Nucleo

STM32F401RE Nucleo board belongs to STM32F401xD/xE family which is based on the high-performance ARM® Cortex®-M4 32-bit RISC core operating at a frequency of up to 84 MHz. The Cortex-M4 with FPU core features a single precision floating point unit (FPU) which supports all ARM single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory protection unit (MPU) which enhances application security. It provides 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 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.

The STM32F401RE Nucleo firmware and related documentation is available on *www.st.com*.

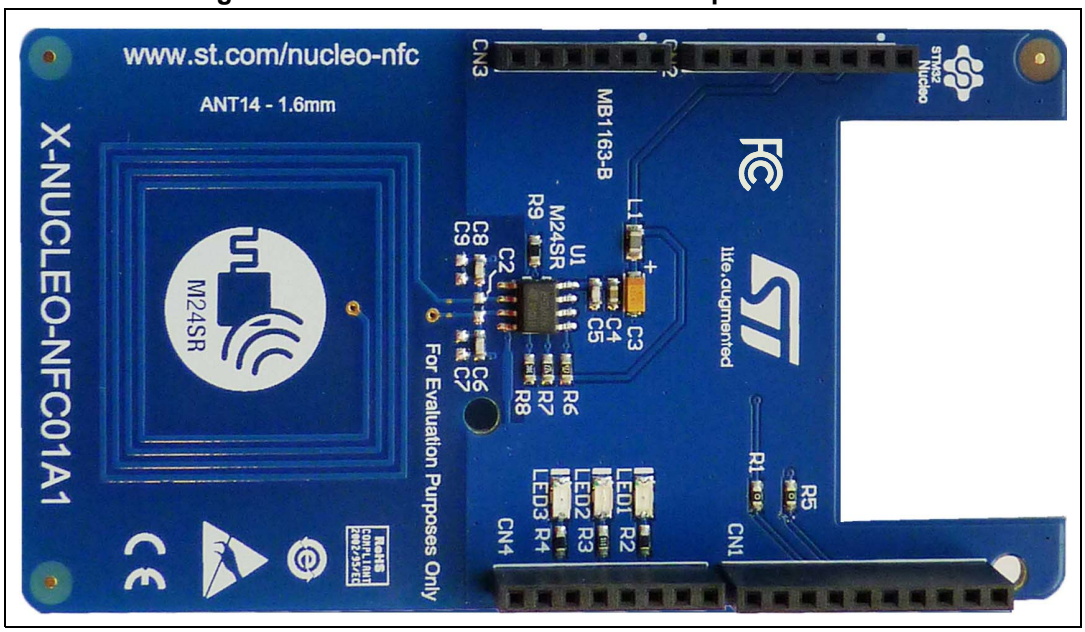
Figure 1. STM32-F4 Nucleo board



3.1.2 M24SR expansion board

The X-NUCLEO-NFC01A1 is a dynamic NFC tag expansion board usable with the STM32 Nucleo system. It is also compatible with Arduino UNO R3 connector layout, and is designed around the STMicroelectronics M24SR dynamic tag. The X-NUCLEO-NFC01A1 interfaces with the STM32 MCU via I^2C.

Figure 2. X-NUCLEO-NFC01A1 M24SR expansion board



3.2 Software description

The following software components are needed in order to setup the suitable development environment for creating an application based on the M24SR expansion board:

- STM32Cube environment and related firmware for STM32Nucleo and M24SR expansion board
- Development tool-chain and compiler. Several solutions are supported. In this document, the Keil µVision development environment is used.

3.2.1 STM32Cube

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.

The STMCube™ package is a free solution that can be downloaded from ST website at www.st.com/stm32cube.

3.2.2 Firmware for STM32 Nucleo with X-NUCLEO-NFC01A1 expansion board

- X-CUBE-NFC1 is a package that contains a firmware with the STM32 Cube framework compatible with the STM32 Nucleo and the X-NUCLEO-NFC01A1 boards. This package contains a sample application that can be used to load data to the M24SR. The same sample application is referenced in further sections of this document.
- The firmware and related documentation is available on *www.st.com*.

3.3 Hardware software and system setup

This section describes hardware, software and system setup procedure for writing an application based on the M24SR expansion board.

3.3.1 Hardware setup

The following hardware is needed to develop a M24SR based application:

- An STM32 Nucleo development platform (NUCLEO-F401RE used in this document)
- M24SR expansion board (see *Figure 2*, order code: X-NUCLEO-NFC01A1)
- USB type A to Mini-B USB cable to connect the STM32 Nucleo board to the PC

3.3.2 Software setup

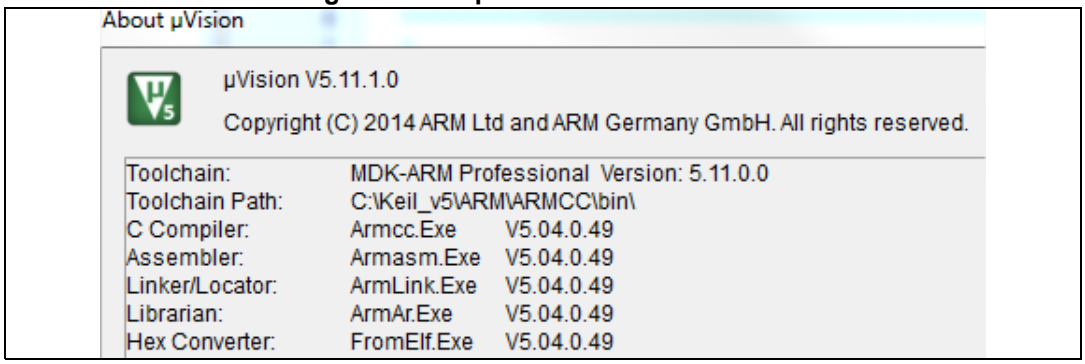
This section lists the minimum requirements in order to:

- setup the SDK
- run the sample testing scenario based on the GUI utility
- customize the application.

Development tool-chains and compilers

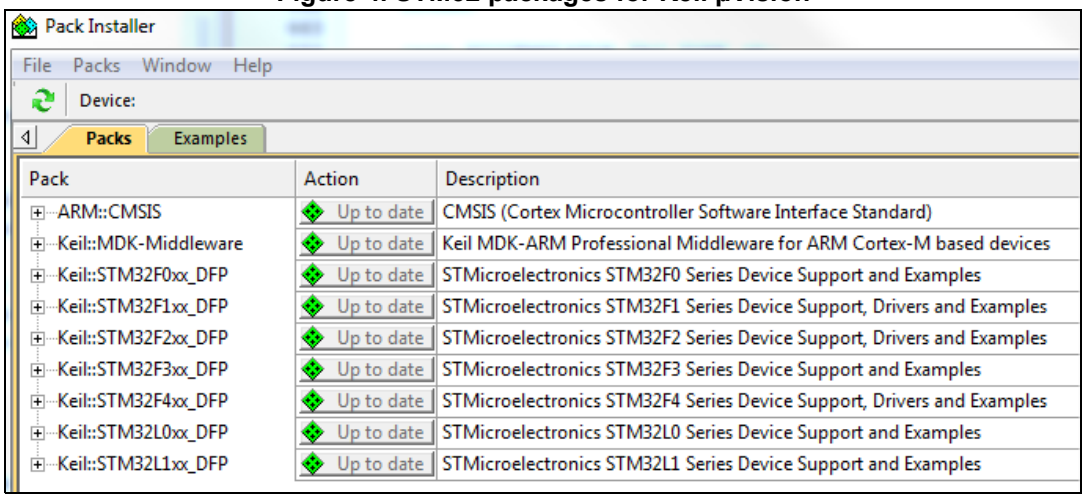
The Keil µVision has been used in this application with the following version:

Figure 3. Keil µVision installed version



In addition to Keil v5, a specific pack for STM32 microcontrollers must be installed. Here is the list of installed packages for STM32 family:

Figure 4. STM32 packages for Keil µVision



Note: The package shown above is for Keil µVision environment, packages to support IAR and TrueStudio environment are also available in the X-CUBE-NFC1 delivery.

Here are the Keil tool-chain minimum requirements for the PC:

- Available hard disk space
  - MDK: 1.4GBytes
- 1GBytes of RAM (2GB recommended)
- Pentium class PC running one of this OS
  - Windows® XP SP3
  - Windows® Vista
  - Windows® 7

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 M24SR expansion board.

STM32 Nucleo and M24SR 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 searching STSW-LINK008 (Windows Vista, Windows 7 or Windows 8) or STSW-LINK009 (Windows XP) on *www.st.com*.

The M24SR expansion board X-NUCLEO-NFC01A1 can be easily connected to the Nucleo motherboard through the Arduino UNO R3 extension connector. The communication between the M24SR expansion board and the STM32 microcontroller on the Nucleo board is using the Inter-Integrated Circuit (I^2C) transport layer.

4 Software overview

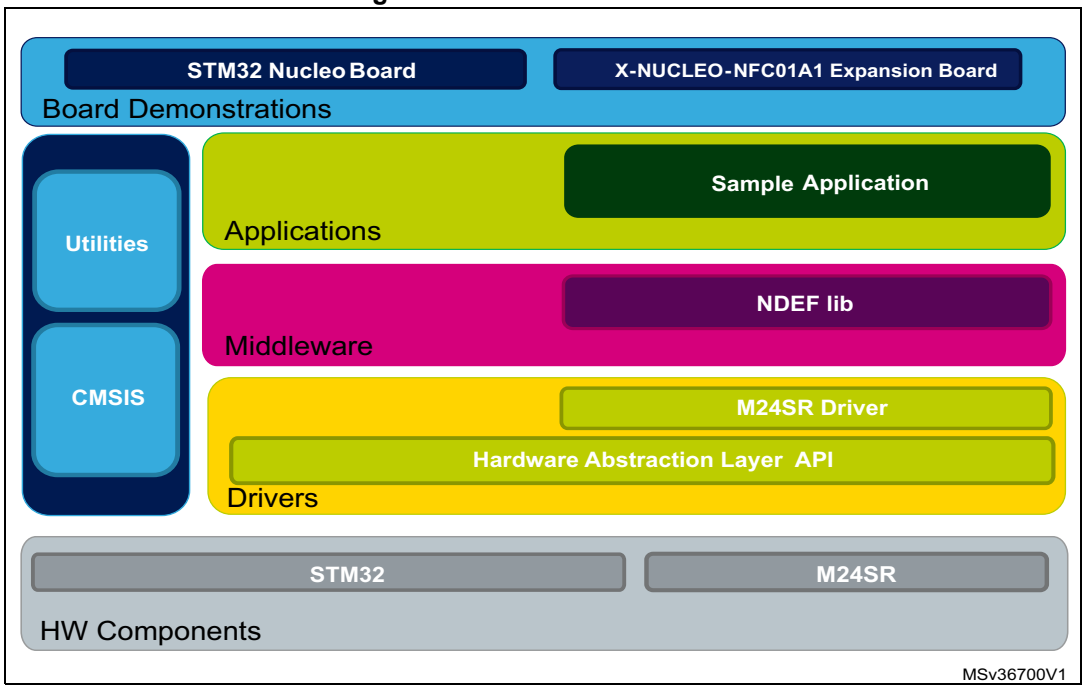
4.1 Software architecture

This section describes various software layers which are used by the application software to access and use the M24SR expansion board. These layers are:

- STM32Cube HAL layer
- Board support package (BSP) layer

Figure 5 below outlines the layering of the software architecture that comprises the STM32 Nucleo and the M24SR expansion board:

Figure 5. Software architecture



4.2 STM32Cube HAL

The STM32Cube HAL is the hardware abstraction layer for the STM32 microcontroller. The STM32Cube HAL ensures maximal portability across STM32 platforms.

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, 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 re-usability and guarantees an easy portability on other devices.

For an in-depth understanding of the STM32Cube HAL drivers API, please refer to the document “Description of STM32F4xx HAL drivers”, User manual UM1749, available from *www.st.com*.

4.3 Board support package (BSP)

The board support package is a software component that allows to control the external peripherals present on the STM32 Nucleo board (but not integrated into the MCU). 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.

In case of M24SR expansion board, it provides the driver interface for M24SR dynamic tag. It provides support for initializing and obtaining M24SR data.

4.4 Middleware

In case of M24SR expansion board, the middleware provides the NDEF interface for M24SR dynamic tag. It provides support to retrieve a NDEF message without need for the user to know how to build a NDEF message and how to store it in a type 4 tag memory like M24SR.

5 Guide for writing application

This section describes how to write an application using M24SR dynamic tag on STM32 Nucleo board equipped with M24SR expansion board.

5.1 Relevant APIs for M24SR application

This section describes the APIs available for initializing and communicating with the M24SR expansion board.

5.1.1 Initialization

For correct operation, the application needs to perform the basic initialization steps in order to configure and setup the STM32 Nucleo with M24SR expansion board hardware and software stack. This section describes the required initialization steps.

Initializing STM32Cube HAL

The STM32Cube HAL library needs to be initialized so that the necessary hardware components are correctly configured.

- `HAL_Init();`

  This API initializes the HAL library. It configures Flash memory prefetch, Flash memory preread and buffer cache. It also configures the time base source, vectored interrupt controller and low-level hardware.

Initializing Nucleo board peripherals and M24SR expansion board

Some of the Nucleo on-board peripherals, and the M24SR expansion board need to be configured before use. The initialization functions are:

- `BSP_LED_Init(Led_TypeDef Led);`

  This API configures the LED on the Nucleo board.

- `BSP_PB_Init(Button_TypeDef Button, ButtonMode_TypeDef Button_Mode);`

  This API configures the user button mode. One of two modes can be selected: the GPIO mode or the external interrupt (EXTI) mode.

- `TT4_Init()`

  This API configures the M24SR device on M24SR expansion board.

5.1.2 Writing data to M24SR expansion board

The M24SR middleware library provides the API and the functionality for programming NDEF messages to M24SR. Once the M24SR is initialized, these functions can be used to write data:

- `TT4_WriteURI(sURI_Info *pURI)`
  This API allow to store an URI message in the M24SR (previous content is deleted)

- `TT4_WriteSMS(sSMSInfo *pSMS)`
  This API allow to store a SMS message in the M24SR (previous content is deleted)

- `TT4_WriteEmail(sEmailInfo *pEmailStruct)`
  This API allow to store an eMail message in the M24SR (previous content is deleted)

- `TT4_WriteVcard(sVcardInfo *pVcard)`
  This API allow to store a vCard message in the M24SR (previous content is deleted)

- `TT4_WriteGeo(sGeoInfo *pGeo)`
  This API allow to store a geolocation message in the M24SR (previous content is deleted)

5.2 Application description

An application example using M24SR expansion board and a Nucleo board can be downloaded from *www.st.com*. The following Nucleo boards are supported: L0, L1, F0, F3, F4. This application allows to:

1. Store a URI NDEF message ("*www.st.com*") in the M24SR, using the API call `TT4_WriteURI()`
2. Bring a NFC capable smartphone near the M24SR expansion board antenna and automatically launch the smartphone browser on the *www.st.com* URL.

6 Revision history

Table 3. Document revision history

Date	Revision	Changes
18-Dec-2014	1	Initial release.