
Getting started with the STM32Cube function pack for IoT node with Dynamic NFC Tag, environmental and motion sensors

Introduction

FP-SNS-SMARTAG1 is an [STM32Cube](#) function pack which allows you to read the motion and environmental sensor data on your IoT node via an NFC enabled reader such as a mobile phone or a tablet. The package supports energy harvesting (enabled by NFC) and battery operated use cases.

This software, together with the suggested combination of STM32 and ST devices can be used, for example, to develop tracking, cold chain, medical, smart sensing, and smart home, city and building applications.

The software runs on an ultra-low power [STM32L0](#) microcontroller and includes drivers for the Dynamic NFC tag and for the motion and environmental sensors.

1 FP-SNS-SMARTAG1 software description

1.1 Overview

The key features of the [FP-SNS-SMARTAG1](#) package are:

- Complete firmware to access data from an IoT node with dynamic NFC tag, environmental and motion sensors
- Ultra-low power operations, with the support of energy harvesting and battery operated use cases
- Compatible with the ST SmarTag application for Android/iOS for reading and displaying sensor data
- Sample implementation available for the [STEVAL-SMARTAG1](#) evaluation board, and for the [X-NUCLEO-IKS01A2](#) and [X-NUCLEO-NFC04A1](#) expansion boards connected to a [NUCLEO-L053R8](#) development board
- Easy portability across different MCU families, thanks to [STM32Cube](#)
- Free, user-friendly license terms

This software tracks temperature, pressure, humidity and vibration values in a fixed time range and sends them via NFC. Using an Android or iOS device the logged data can be monitored/displayed.

The software gathers:

- the temperature, humidity, pressure, and vibration sensor drivers for the [HTS221](#), [LPS22HB](#), [LSM6DSL](#), [LSM303AGR](#) and [ST25DV04K](#) devices for the [X-NUCLEO-IKS01A2](#) and [X-NUCLEO-NFC04A1](#) expansion boards connected to a [NUCLEO-L053R8](#) running on STM32.
- the temperature, humidity, pressure, and vibration sensor for the [HTS221](#), [LPS22HB](#), [LIS2DW12](#) and [ST25DV64K](#) devices for the [STEVAL-SMARTAG1](#) evaluation board running on STM32.

This package is compatible with the ST SmarTag Android/iOS (Ver. 1.0.0 or higher) application available at the respective GooglePlay/iTunes store, which can be used to display information sent via dynamic NFC/RFID tag protocol.

1.2 Architecture

The [STM32Cube](#) function packs leverage the modularity and interoperability of STM32 Nucleo and X-NUCLEO boards, and STM32Cube and X-CUBE software, to create function examples, embodying some of the most common use cases, for each application area.

These software function packs are designed to exploit as much as possible the underlying [STM32 ODE](#) hardware and software components to best fit the requirements of final users' applications.

Moreover, function packs may include additional libraries and frameworks which do not present the original X-CUBE packages, thus enabling new functionalities and creating a real and usable system for developers.

STM32Cube™ is designed by STMicroelectronics to reduce development effort, time and cost across the entire 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 STM32Cube for the STM32 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

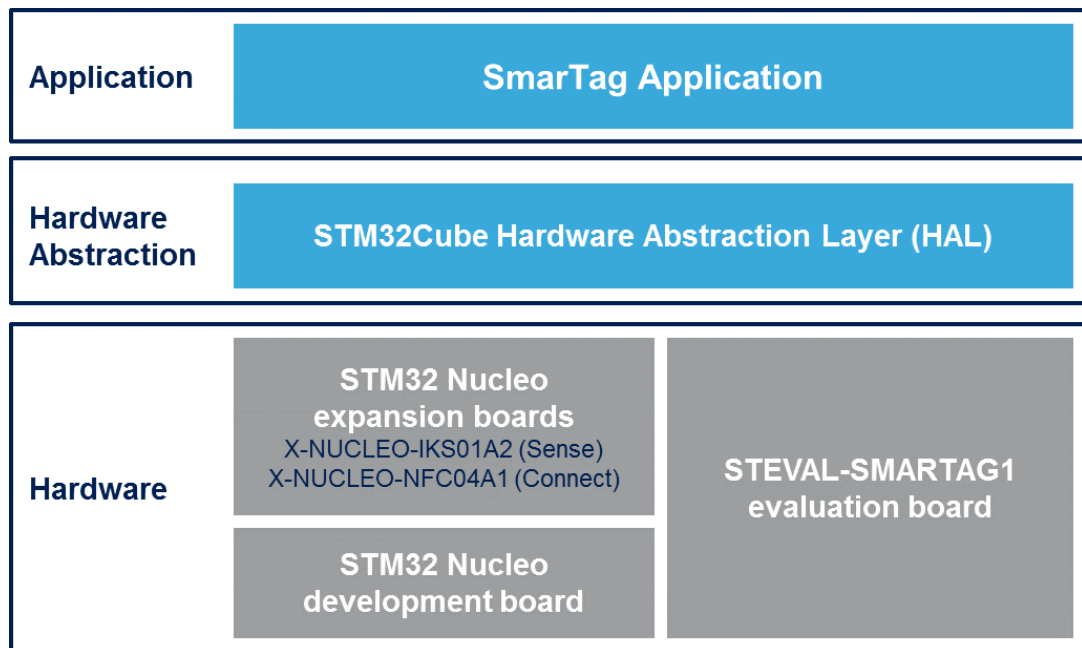
To access and use the sensor expansion board, the application software uses:

- **STM32Cube HAL layer:** provides a simple, generic and multi-instance set of generic and extension APIs (application programming interfaces) to interact with the upper layer application, libraries and stacks. It is directly based on a generic architecture and allows the layers that are built on it, such as the middleware

layer, to implement their functions without requiring the specific hardware configuration for a given microcontroller unit (MCU). This structure improves library code reusability and guarantees easy portability across other devices.

- **Board support package (BSP) layer:** supports the peripherals on the STM32 Nucleo board (except the MCU) with a limited set of APIs providing a programming interface for certain board-specific peripherals like the LED, the user button, etc., and helps determine the specific board version. For the sensor expansion board, it provides the programming interface for various inertial and environmental sensors and support for initializing and reading sensor data.

Figure 1. FP-SNS-SMARTAG1 software architecture

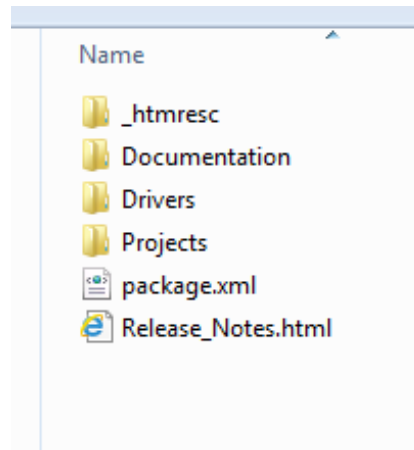


This software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller. The package extends [STM32Cube](#) by providing a board support package (BSP) for the Dynamic NFC/RFID tag IC expansion board based on [ST25DV](#), for the environmental and motion MEMS sensors expansion board, and for the [STEVAL-SMARTAG1](#) evaluation board. The drivers abstract low-level details of the hardware and allow the sample applications to leverage NFC communication and access sensor data in a hardware-independent manner. The package supports different use cases featuring ultra-low power modes of operation. An advanced, one-shot mode leverages the energy harvesting feature of the dynamic NFC tag to provide enough energy to power the reading of sensor data (available for the [STEVAL-SMARTAG1](#) board only).

A battery operated data-logger mode enables continuous sensor data reading, which can be viewed on a mobile device using the ST SmarTag application. Developers can use this package to prototype ultra-low power IoT applications requiring sensor data reading through NFC communication.

1.3 Folder structure

Figure 2. FP-SNS-SMARTAG1 package folder structure



The following folders are included in the software package:

- **Documentation:** contains a compiled HTML file generated from the source code, which details 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.
- **Projects:** contains a sample application used for reading the motion and environmental sensor data on your IoT node via an NFC enabled reader, such as a mobile phone or a tablet. It is available for the [NUCLEO-L053R8](#) and [STEVAL-SMARTAG1](#) boards and compatible with the IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM) and System Workbench for STM32 development environments.

1.4 APIs

Detailed user-API technical information with full function and parameter descriptions is available in a compiled HTML file in the package "Documentation" folder.

1.5 Sample application description

A sample application is available for:

- the [X-NUCLEO-IKS01A2](#) and [X-NUCLEO-NFC04A1](#) expansion boards with the [NUCLEO-L053R8](#) board
- the [STEVAL-SMARTAG1](#) evaluation board

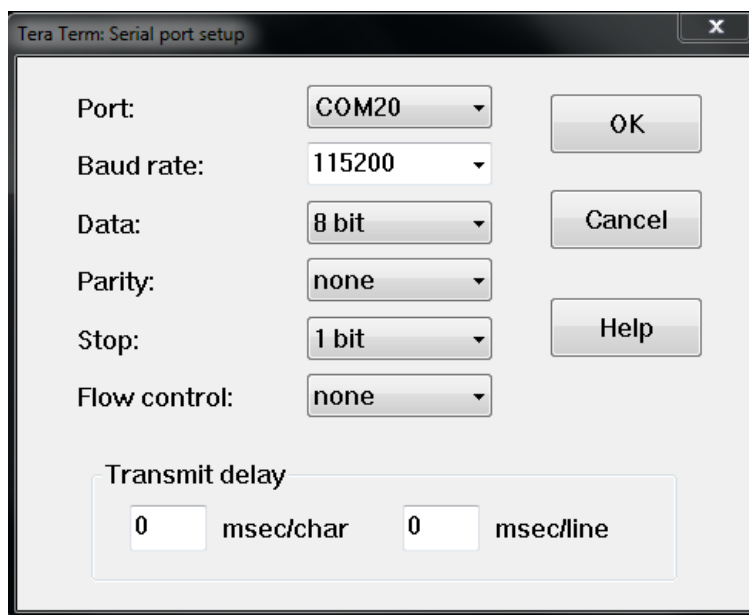
Ready to build projects are available for multiple IDEs.

You can set up a terminal window for the appropriate UART communication port in order to control the initialization phase, as shown in the figure below. The following Baud Rate must be used:

- 115200 for NUCLEO-L053R8
- 9600 for STEVAL-SMARTAG1 (due to flash size constraints, a few messages have been disabled by default - you can enable/disable the messages in the code according to your preference)

Note: *UART communication is not available for the STEVAL-SMARTAG1 using the binary file generated by System Workbench for STM32 (due to flash size constraints).*

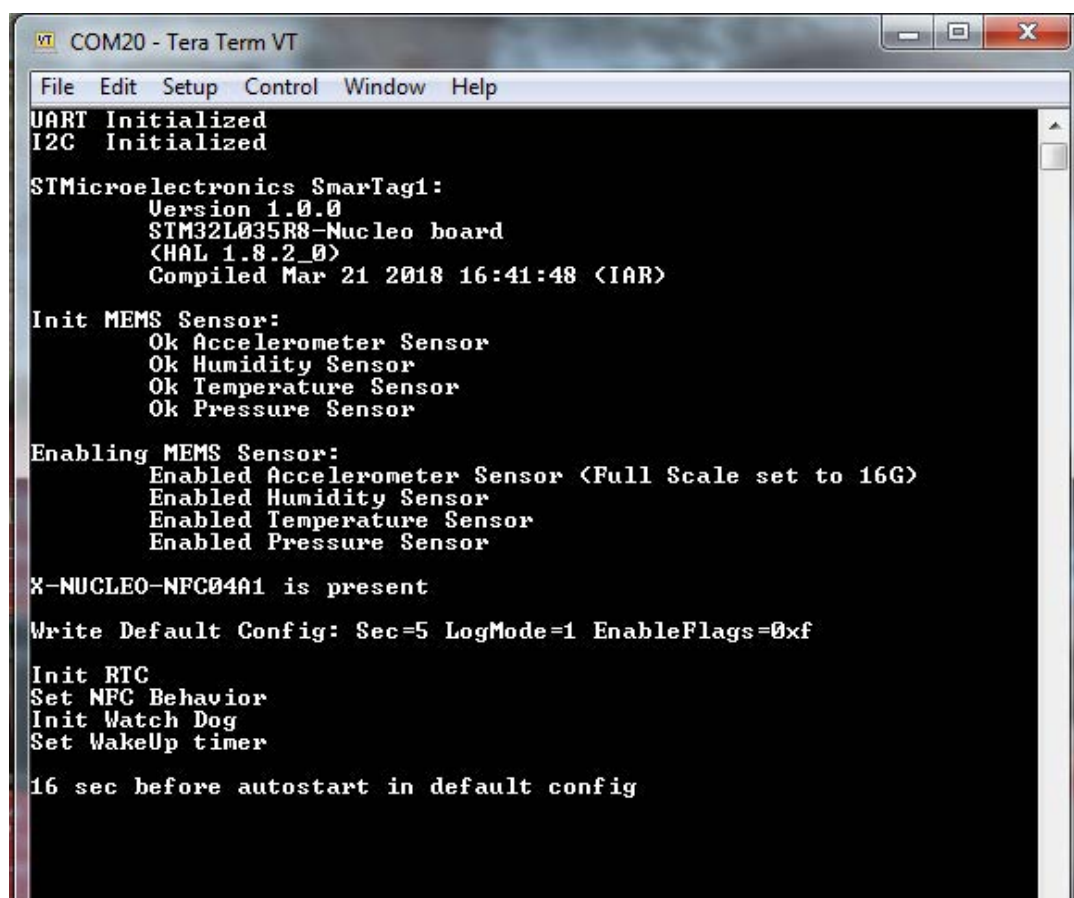
Figure 3. Tera Term setup



When you first press the reset button, the application:

- starts initializing the UART and I²C interfaces
- checks whether all the sensors are present and working
- shows the accelerometer full scale value
- checks whether the X-NUCLEO-NFC04A1 expansion board is present
- shows the default values of the SmarTag config (LogMode, RangeTime, EnableFlags)
- initializes the RTC
- sets the NFC behavior
- initializes the Watch Dog
- sets the wake up timer

Figure 4. Initialization phase



```

COM20 - Tera Term VT
File Edit Setup Control Window Help
UART Initialized
I2C Initialized
STMicroelectronics SmarTag1:
  Version 1.0.0
  STM32L035R8-Nucleo board
  <HAL 1.8.2_0>
  Compiled Mar 21 2018 16:41:48 <IAR>
Init MEMS Sensor:
  Ok Accelerometer Sensor
  Ok Humidity Sensor
  Ok Temperature Sensor
  Ok Pressure Sensor
Enabling MEMS Sensor:
  Enabled Accelerometer Sensor <Full Scale set to 16G>
  Enabled Humidity Sensor
  Enabled Temperature Sensor
  Enabled Pressure Sensor
X-NUCLEO-NFC04A1 is present
Write Default Config: Sec=5 LogMode=1 EnableFlags=0xf
Init RTC
Set NFC Behavior
Init Watch Dog
Set WakeUp timer
16 sec before autostart in default config

```

After the auto-start range time, the samples are logged using the default SmarTag config.

Figure 5. UART console output after auto-start range time

```

VT COM20 - Tera Term VT
File Edit Setup Control Window Help
UART Initialized
I2C Initialized

STMicroelectronics SmarTag1:
  Version 1.0.0
  STM32L035R8-Nucleo board
  <HAL 1.8.2_0>
  Compiled Mar 21 2018 16:41:48 <IAR>

Init MEMS Sensor:
  Ok Accelerometer Sensor
  Ok Humidity Sensor
  Ok Temperature Sensor
  Ok Pressure Sensor

Enabling MEMS Sensor:
  Enabled Accelerometer Sensor <Full Scale set to 16G>
  Enabled Humidity Sensor
  Enabled Temperature Sensor
  Enabled Pressure Sensor

X-NUCLEO-NFC04A1 is present

Write Default Config: Sec=5 LogMode=1 EnableFlags=0xf

Init RTC
Set NFC Behavior
Init Watch Dog
Set WakeUp timer

16 sec before autostart in default config

AutoStart

Read Sensor
->TS 3:26:42 Sample #1

Read Sensor
->TS 3:26:47 Sample #2

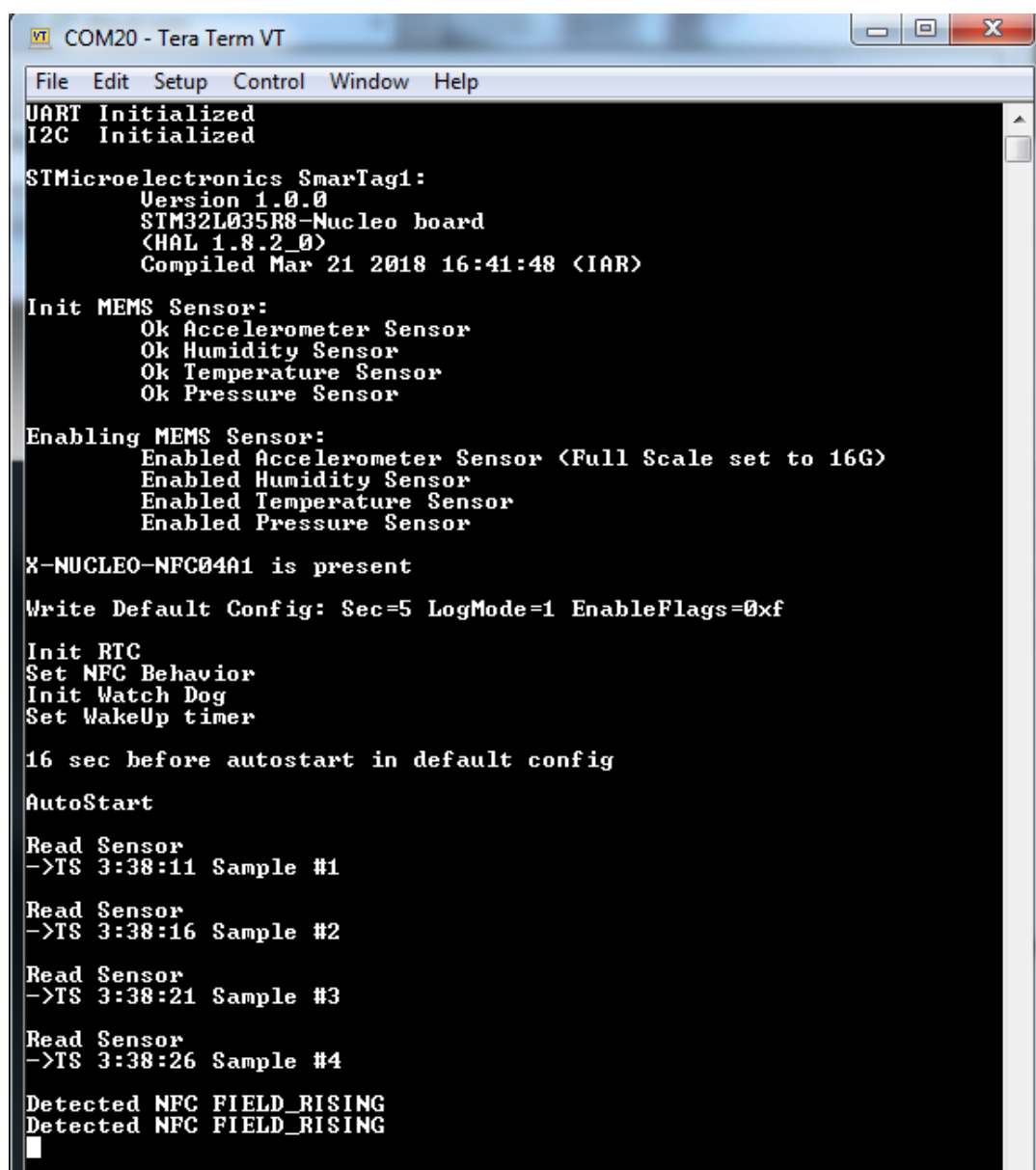
Read Sensor
->TS 3:26:52 Sample #3

Read Sensor
->TS 3:26:57 Sample #4

```

When the smartphone is close to the NFC tag, the message "Detected NFC FIELD_RISING" appears.

Figure 6. UART console output NFC Rising



```

COM20 - Tera Term VT
File Edit Setup Control Window Help
UART Initialized
I2C Initialized

STMicroelectronics SmarTag1:
  Version 1.0.0
  STM32L035R8-Nucleo board
  <HAL 1.8.2_0>
  Compiled Mar 21 2018 16:41:48 <IAR>

Init MEMS Sensor:
  Ok Accelerometer Sensor
  Ok Humidity Sensor
  Ok Temperature Sensor
  Ok Pressure Sensor

Enabling MEMS Sensor:
  Enabled Accelerometer Sensor <Full Scale set to 16G>
  Enabled Humidity Sensor
  Enabled Temperature Sensor
  Enabled Pressure Sensor

X-NUCLEO-NFC04A1 is present

Write Default Config: Sec=5 LogMode=1 EnableFlags=0xf

Init RTC
Set NFC Behavior
Init Watch Dog
Set WakeUp timer

16 sec before autostart in default config

AutoStart

Read Sensor
->TS 3:38:11 Sample #1

Read Sensor
->TS 3:38:16 Sample #2

Read Sensor
->TS 3:38:21 Sample #3

Read Sensor
->TS 3:38:26 Sample #4

Detected NFC FIELD_RISING
Detected NFC FIELD_RISING

```

When the smartphone is kept distant from the NFC tag, the message "Detected NFC FIELD FALLING" appears together with the new config if a new one is detected.

In particular, if the new logging mode has a threshold, the messages also highlight the enabled accelerometer events.

Figure 7. UART console output NFC Falling

```

COM20 - Tera Term VT
File Edit Setup Control Window Help
UART Initialized
I2C Initialized
STMicroelectronics SmarTag1:
  Version 1.0.0
  STM32L035R8-Nucleo board
  <HAL 1.8.2_0>
  Compiled Mar 21 2018 16:41:48 <IAR>

Init MEMS Sensor:
  Ok Accelerometer Sensor
  Ok Humidity Sensor
  Ok Temperature Sensor
  Ok Pressure Sensor

Enabling MEMS Sensor:
  Enabled Accelerometer Sensor <Full Scale set to 16G>
  Enabled Humidity Sensor
  Enabled Temperature Sensor
  Enabled Pressure Sensor

X-NUCLEO-NFC04A1 is present
Write Default Config: Sec=5 LogMode=1 EnableFlags=0xf

Init RTC
Set NFC Behavior
Init Watch Dog
Set WakeUp timer

16 sec before autostart in default config
AutoStart

Read Sensor
->TS 3:44:40 Sample #1

Read Sensor
->TS 3:44:45 Sample #2

Read Sensor
->TS 3:44:50 Sample #3

Read Sensor
->TS 3:44:55 Sample #4

Detected NFC FIELD_RISING
Detected NFC FIELD_RISING
Detected NFC FIELD_FALLING

Read New Config Sec=5 LogMode=1 EnableFlags=0xf

Disabled 6D Orientation Detection
Disabled Wake Up Detection
Set FIFO Bypass mode
Accelerometer Full Scale set to 16G

Read Sensor
->TS 8:45:54 Sample #1

```

1.6 Android and iOS sample client application

The **FP-SNS-SMARTAG1** software for STM32Cube is compatible with the ST SmarTag Android/iOS application (ver. 1.0.0 or higher) available at the respective GooglePlay/iOS store.

The next sections show some use cases enabled by the Android application.

1.6.1

Settings

When the smartphone is close to the NFC tag, the application is opened as shown in [Figure 8. ST SmarTag \(Android version\) setting page 1](#).

From this page, you can select the data to be logged (pressure, temperature, humidity and vibration) as well as the time interval.

After saving the settings, when the smartphone is distant from the NFC tag, the logging starts in default mode.

You can choose other two different logging modes:

1. **Log only out of range [min, max] and accelerometer events:**
 - a. sync events: the selected data are logged only if a minimum or maximum threshold value is reached.
 - b. async events: logging only if a wakeup or change of orientation events occur (if selected, as shown in [Figure 9. ST SmarTag \(Android version\) setting page 2](#))
2. **Force logging of one sample:** the current value of the selected data is logged; then, the data logging restarts in the previous mode.

Figure 8. ST SmarTag (Android version) setting page 1

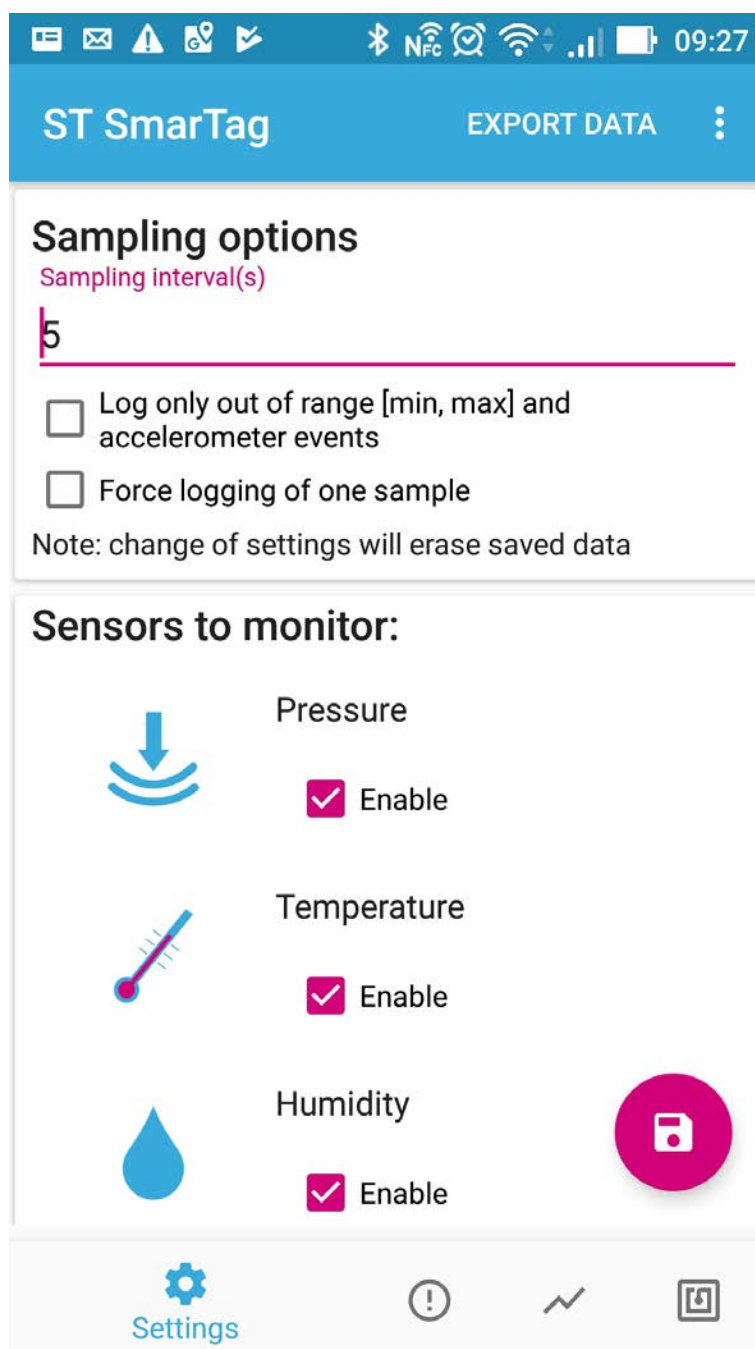
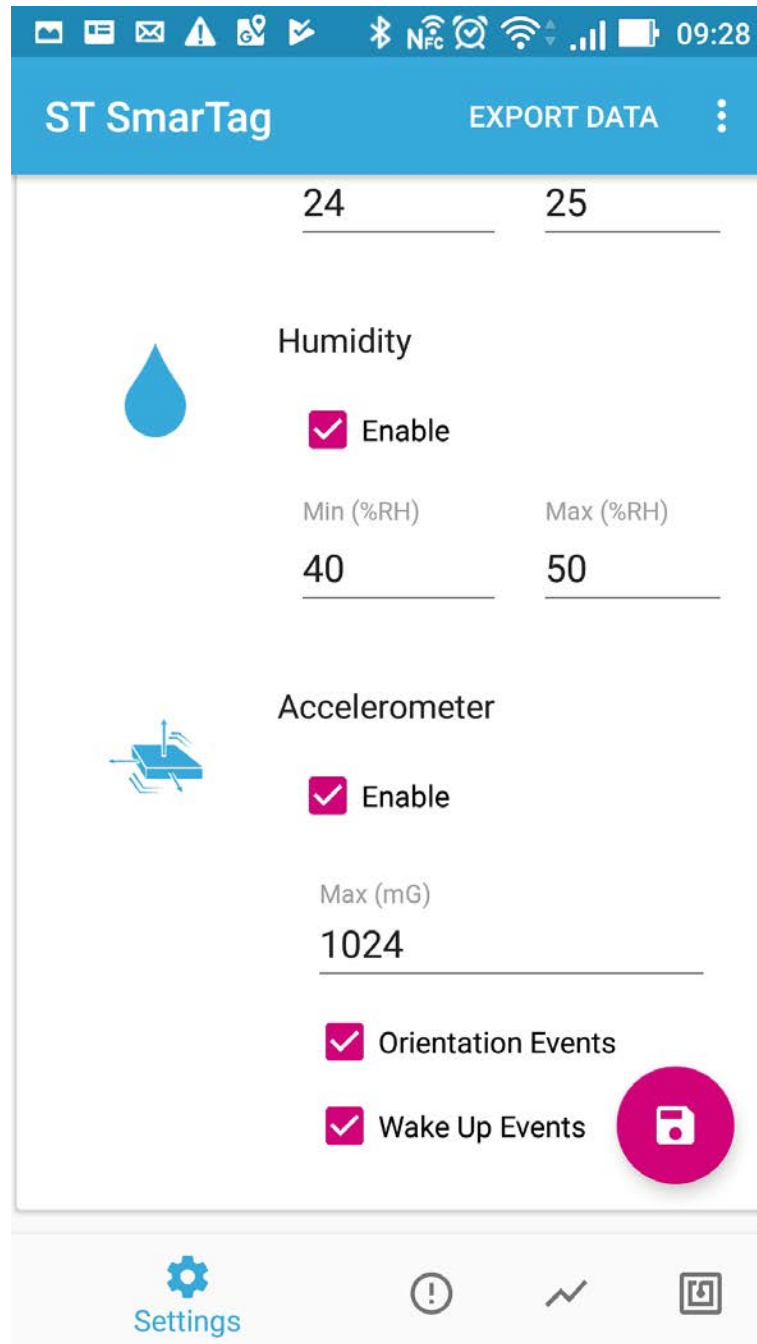


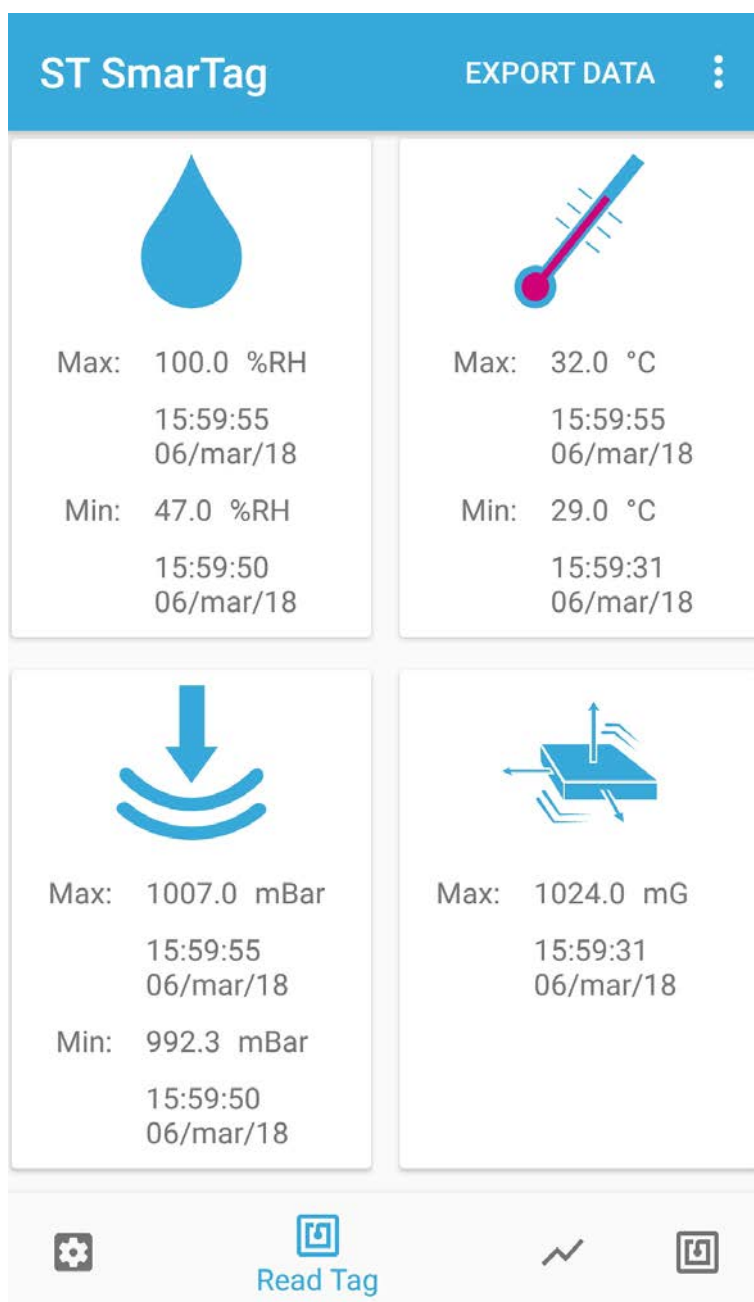
Figure 9. ST SmarTag (Android version) setting page 2



1.6.2 Read tag

This page shows the maximum and minimum values obtained during the selected data logging.

Figure 10. ST SmarTag (Android version) maximum and minimum value page

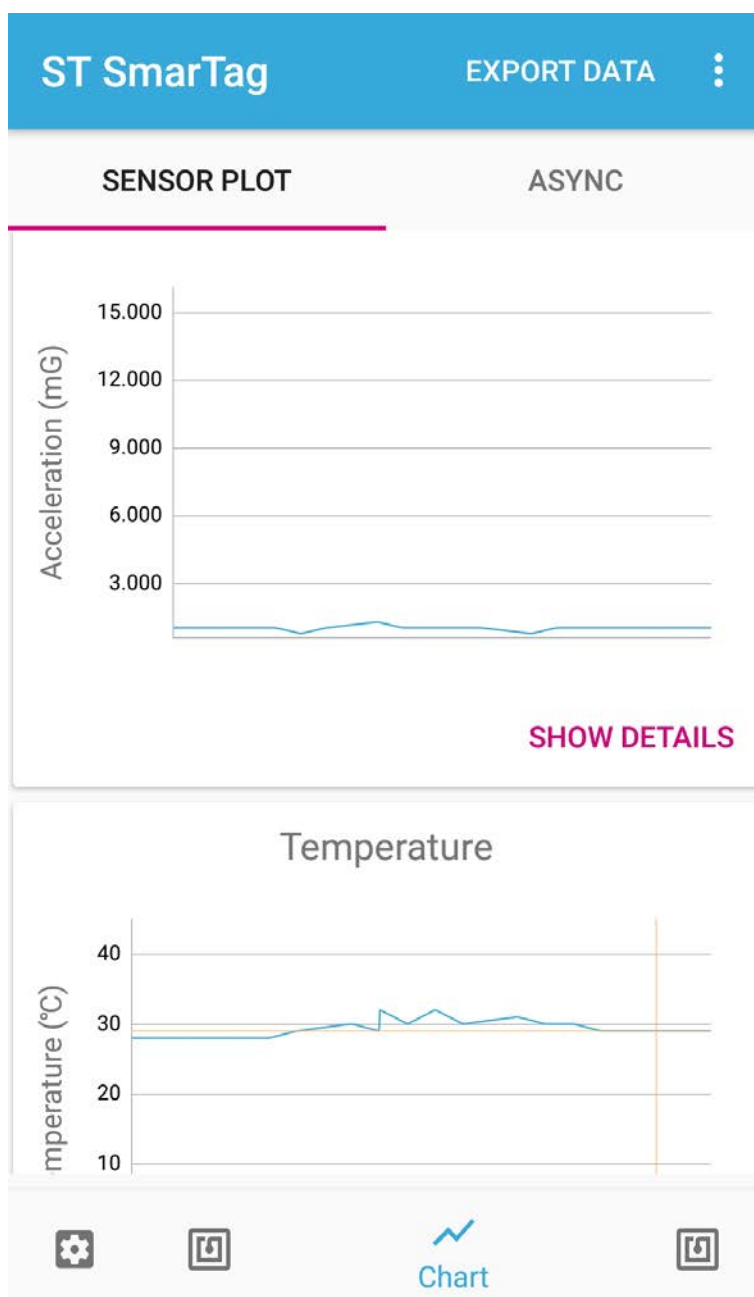


1.6.3

Chart

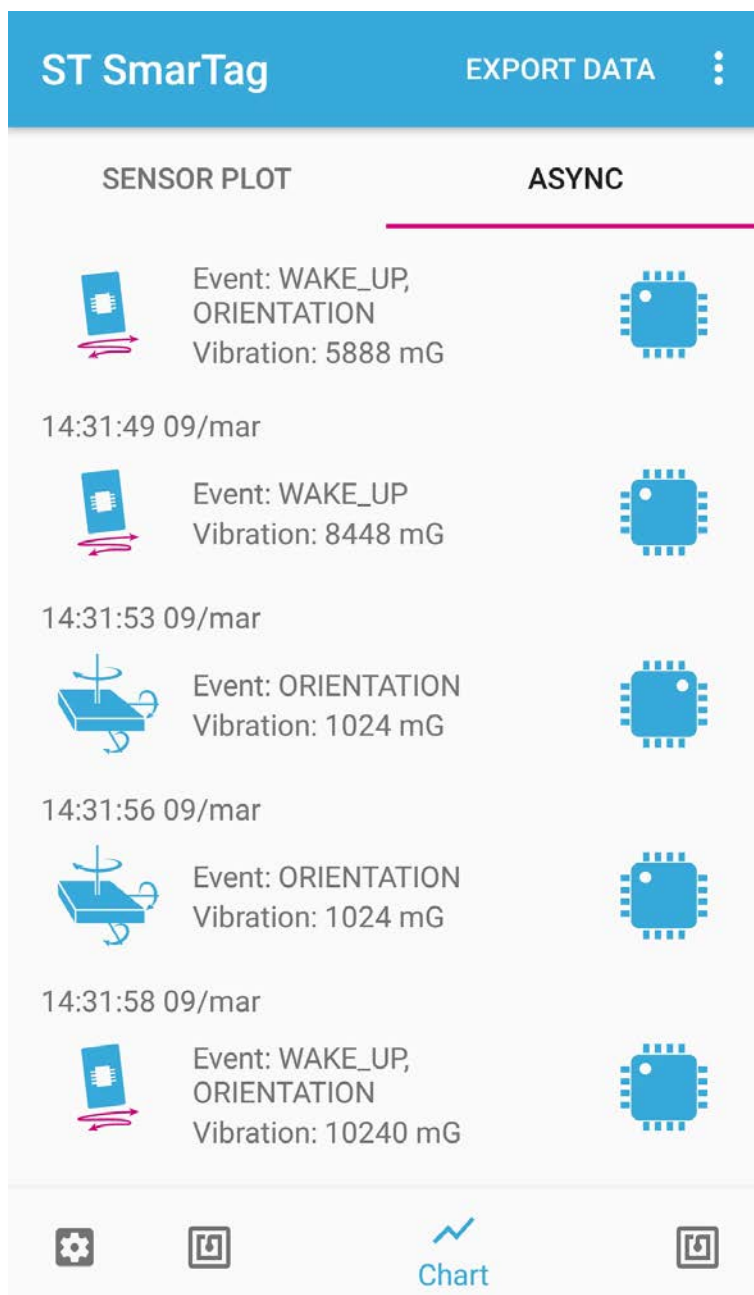
This page shows the plots of the data selected for the logging, when the sensor plot feature is selected.

Figure 11. ST SmarTag (Android version) chart sensor plot page



If the "Log with Threshold" has been chosen and Async is selected, the accelerometer events are shown.

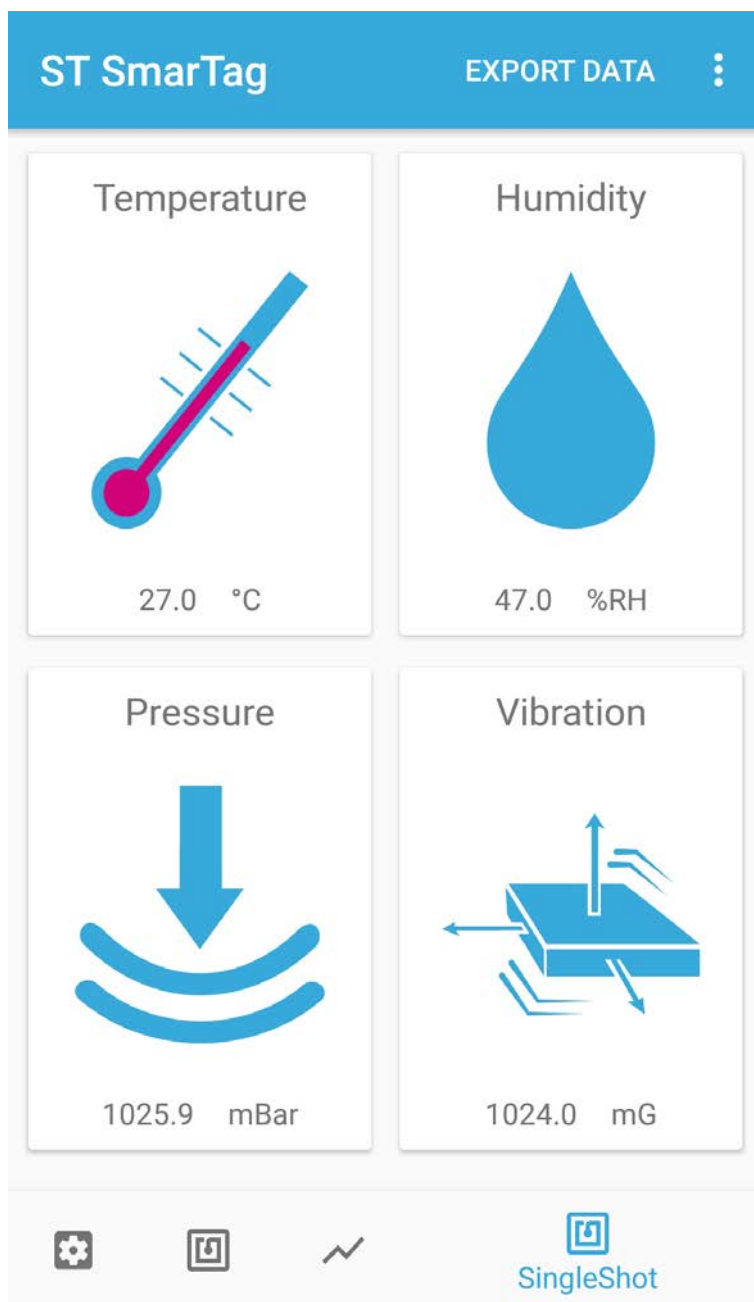
Figure 12. ST SmarTag (Android version) chart async page



1.6.4 Single shot

If this page is selected, the current values of the selected data are read from the tag and then displayed.

Figure 13. ST SmarTag (Android version) single shot page



2 System setup guide

2.1 Hardware description

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

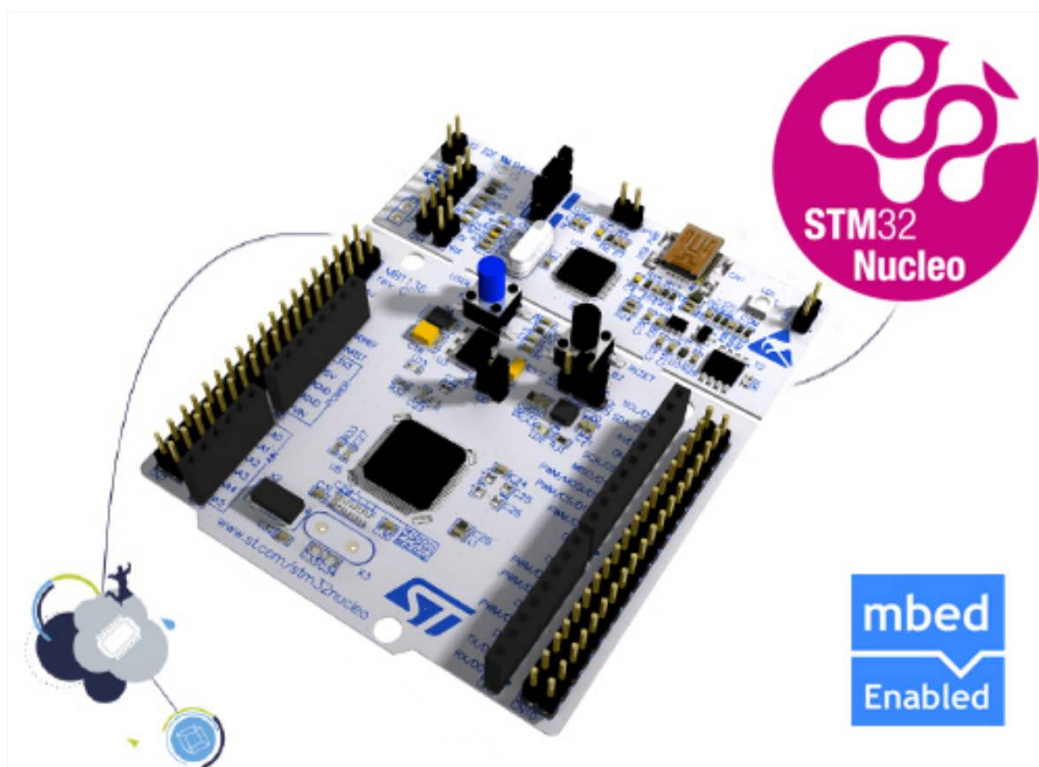
2.1.1 STM32 Nucleo platform

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 14. STM32 Nucleo board



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

2.1.2 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 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 15. X-NUCLEO-NFC04A1 expansion board



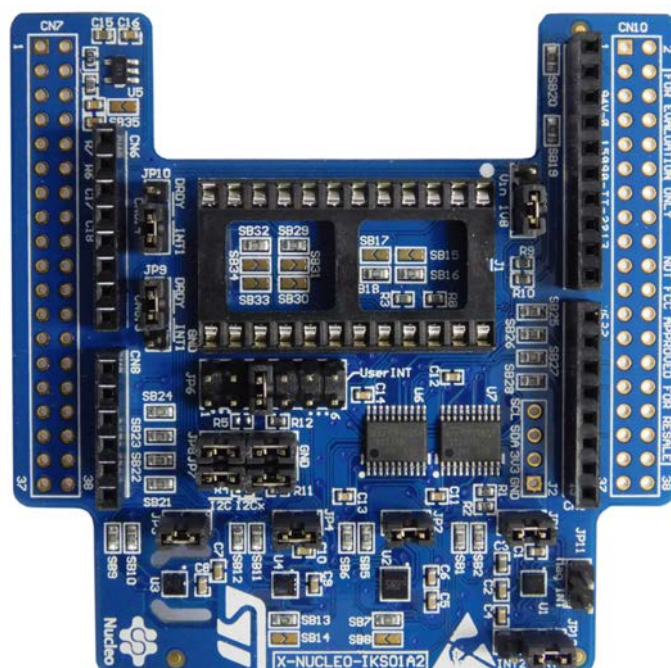
2.1.3 X-NUCLEO-IKS01A2 expansion board

The [X-NUCLEO-IKS01A2](#) is a motion MEMS and environmental sensor expansion board for STM32 Nucleo.

It is compatible with the Arduino UNO R3 connector layout, and is designed around the [LSM6DSL](#) 3D accelerometer and 3D gyroscope, the [LSM303AGR](#) 3D accelerometer and 3D magnetometer, the [HTS221](#) humidity and temperature sensor and the [LPS22HB](#) pressure sensor.

The X-NUCLEO-IKS01A2 interfaces with the STM32 microcontroller via the I²C pin, and it is possible to change the default I²C port.

Figure 16. X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board



2.1.4 STEVAL-SMARTAG1 evaluation board

This smart and flexible NFC Tracker evaluation board with sensors includes a comprehensive software library and a sample application to monitor and log sensor data over NFC from an Android or iOS device.

The ultra-low power sensor node evaluation board mounts an ST25DV NFC Tag, an STM32L0 ARM Cortex M0+, environment sensors (temperature, humidity and pressure) and motion (accelerometer) sensor.

The evaluation board features NFC harvesting to supply power and a battery cradle for a CR2032 battery.

Figure 17. STEVAL-SMARTAG1 evaluation board



The STEVAL-SMARTAG1 evaluation board key features are:

- ST25DV64K dynamic NFC tag solution based on 64K-bit (8K-Byte) EEPROM and with I²C interface, Fast Transfer Mode and Energy Harvesting features
- STM32L031K6 ultra-low-power ARM Cortex-M0+ MCU running at 32 MHz with 32-Kbytes Flash and 8-Kbytes RAM
- LIS2DW12 ultra-low-power high-performance three-axis linear accelerometer

- LPS22HB ultra-compact piezo-resistive absolute pressure sensor which functions as a digital output barometer: 260-1260 hPa
- HTS221 capacitive digital sensor for relative humidity and temperature
- STLQ015 low drop linear regulator power management
- CR2032 Battery powered (not included)
- STM32Cube function pack ([FP-SNS-SMARTAG1](#))
- Android (Google Play) and iOS demo apps (ST SmarTag)
- Suitable for the following applications:
 - Internet of Things
 - Supply Chain and Cold-Chain Management
 - Smart building, home and city
 - Retail and apparel
 - Smart packaging
 - Medical and pharmaceutical
 - Batteryless sensing
 - Smart agriculture (soil control, animal tracking, etc.)

2.2 Software description

The following software components are needed to set up a suitable development environment to create applications for the STM32 Nucleo equipped with the sensors and NFC tag expansion boards and for the [STEVAL-SMARTAG1](#) evaluation board:

- [FP-SNS-SMARTAG1](#): complete firmware to access data from an IoT node with dynamic NFC tag, environmental and motion sensors. The package provides easy portability across different MCU families, thanks to [STM32Cube](#). The FP-SNS-SMARTAG1 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 + ST-LINK

2.3 Hardware and software setup

2.3.1 Hardware setup

The following hardware components are required:

- for hardware configuration based on STM32 Nucleo expansion boards:
 - One STM32 Nucleo board (order code: [NUCLEO-L053R8](#))
 - One sensor expansion board (order code: [X-NUCLEO-IKS01A2](#))
 - One dynamic NFC/RFID tag IC expansion board (order code: [X-NUCLEO-NFC04A1](#))
 - One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
- for hardware configuration based on the [STEVAL-SMARTAG1](#) evaluation board:
 - One STEVAL-SMARTAG1 evaluation board
 - One ST-LINK/V2-1 debugger/programmer integrated in the STM32 Nucleo board
 - One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
 - One CR 2032 battery

2.3.2 Software setup

This section describes how to set up different hardware components before writing and executing an application:

- on the [STM32 Nucleo](#) board connected to the expansion boards

- on the [STEVAL-SMARTAG1](#) evaluation board

2.3.2.1 Development tool-chains and compilers

Select one of the Integrated Development Environments supported by the [STM32Cube](#) expansion software and follow the system requirements and setup information provided by the selected IDE provider.

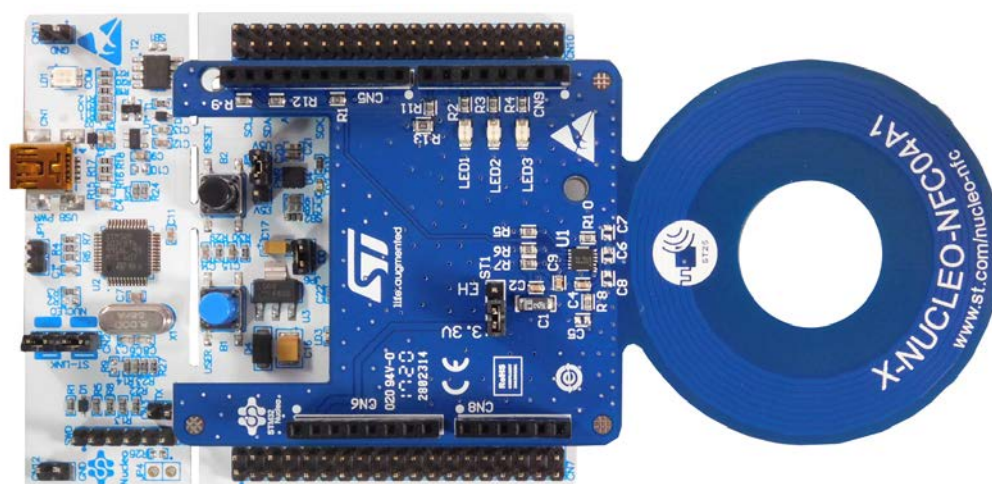
2.3.3 System setup guide

2.3.3.1 STM32 Nucleo and 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 clicking [STSW-LINK008](#) or [STSW-LINK009](#).

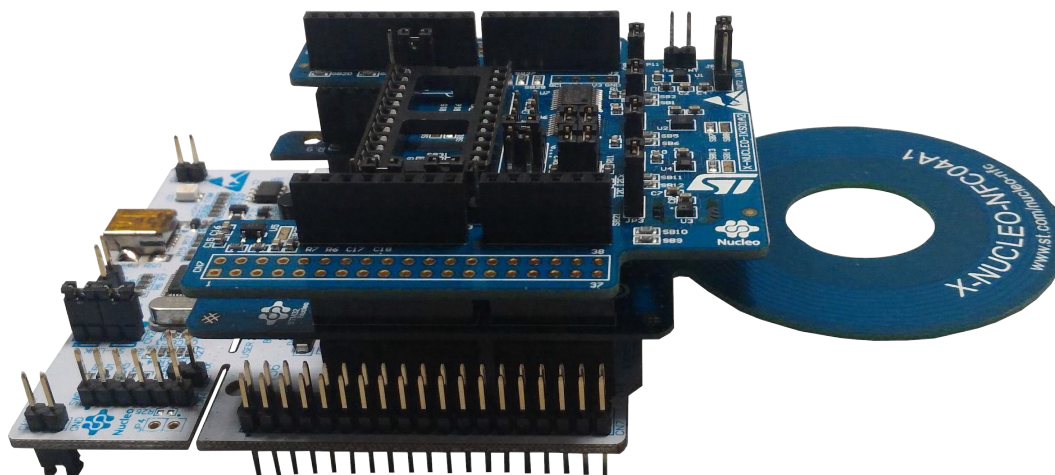
The [X-NUCLEO-NFC04A1](#) dynamic NFC/RFID tag IC board is easily connected to the STM32 Nucleo board via the Arduino UNO R3 extension connectors, as shown below.

Figure 18. STM32 Nucleo plus X-NUCLEO-NFC04A1 board



The [X-NUCLEO-IKS01A2](#) sensor expansion board is easily connected to the X-NUCLEO-NFC04A1 expansion board through the Arduino UNO R3 extension connectors, as shown below.

Figure 19. STM32 Nucleo plus X-NUCLEO-NFC04A1 plus X-NUCLEO-IKS01A2 boards



2.3.3.2 STEVAL-SMARTAG1 evaluation board setup

The ST-LINK/V2-1 debugger/programmer integrated in the STM32 Nucleo board must be used to program the STEVAL-SMARTAG1 evaluation board. The developer can download the relevant version of the ST-LINK/V2-1 USB driver by clicking [STSW-LINK008](#) or [STSW-LINK009](#).

- Step 1.** Connect the STEVAL-SMARTAG1 to the ST-LINK/V2-1 debugger/programmer integrated on the STM32 Nucleo board via the SWD connector to start programming.
- Step 2.** Connect your STM32 Nucleo board to the STEVAL-SMARTAG1, after ensuring CN2 jumpers are OFF.

Important:

Pay attention to the polarity of the connectors: pin 1 can be identified by a little circle on the PCB silkscreen (on the STM32 Nucleo and the STEVAL-SMARTAG1 boards).

Figure 20. STEVAL-SMARTAG1 evaluation board connected to ST-LINK/V2-1 via SWD

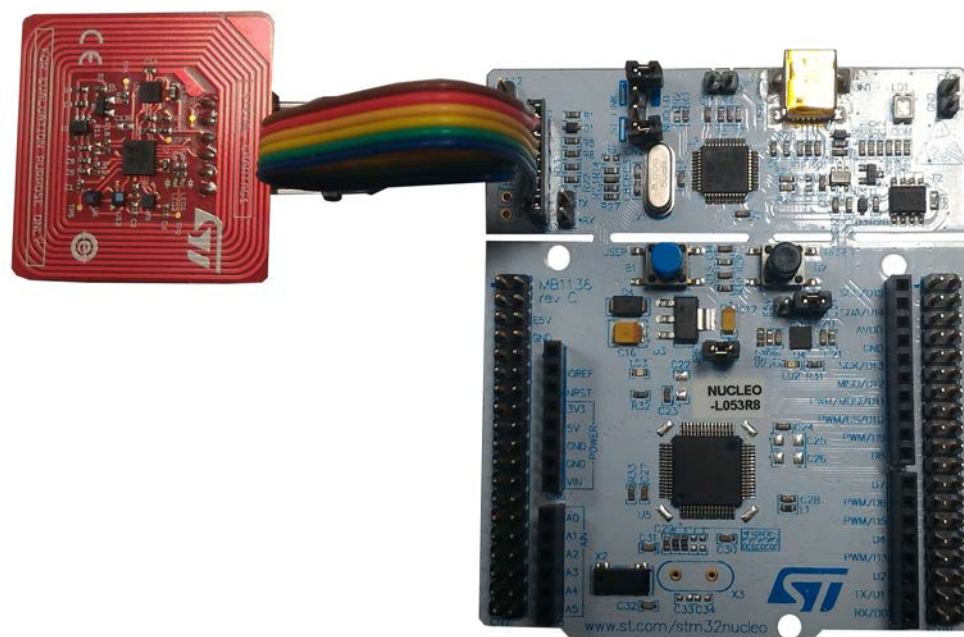
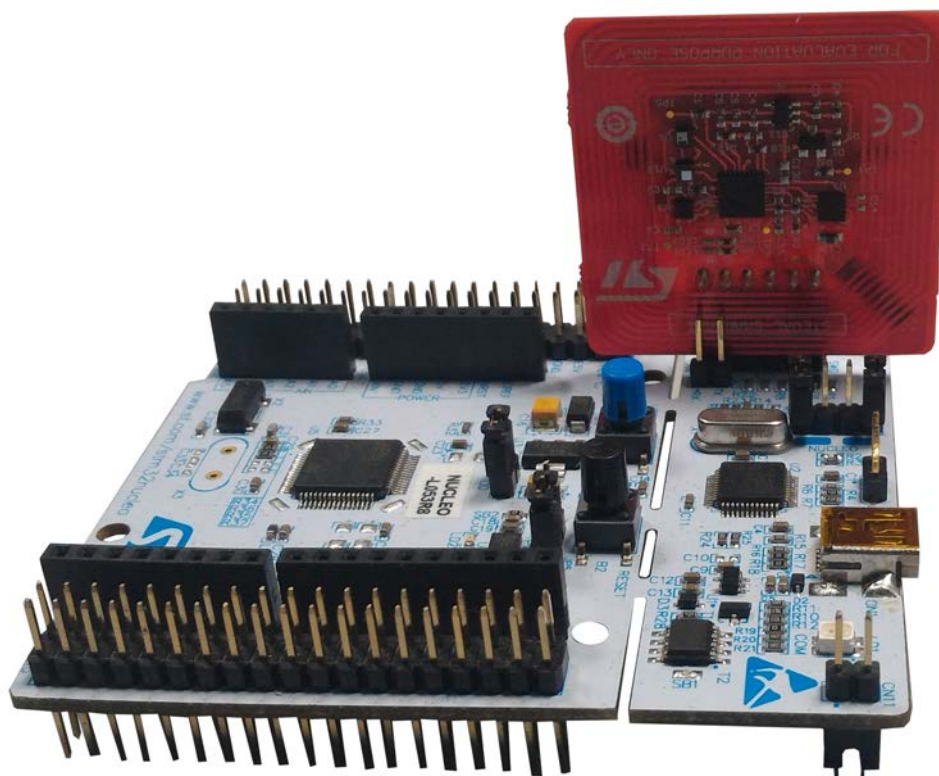


Figure 21. STEVAL-SMARTAG1 evaluation board directly connected to ST-LINK/V2-1



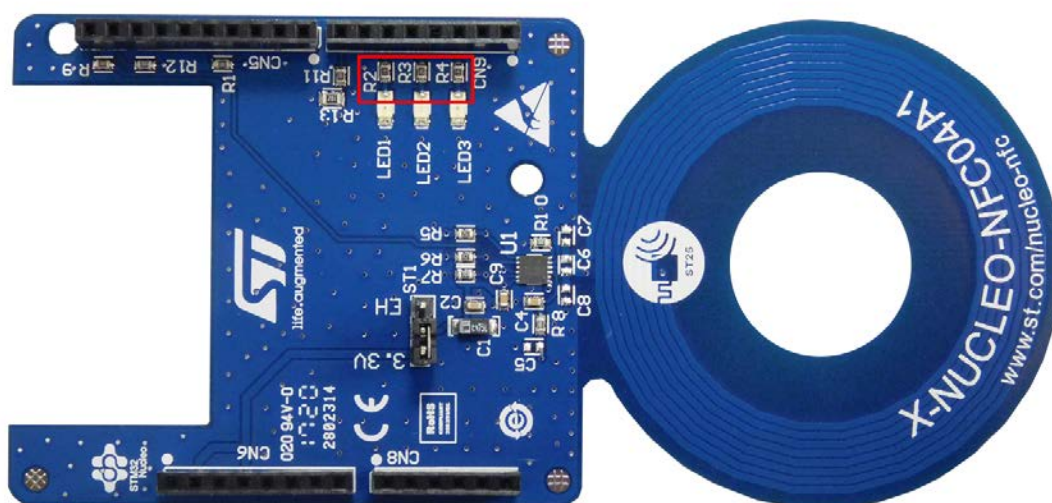
2.3.3.3

Important additional hardware information

For any STM32 Nucleo board configuration: before connecting the X-NUCLEO-NFC04A1 board to the X-NUCLEO-IKS01A2 expansion board through the Arduino UNO R3 connector, remove these resistors on the X-NUCLEO-NFC04A1 board:

- R2 (180 Ω)
- R3 (1 K Ω)
- R4 (560 Ω)

Figure 22. X-NUCLEO-NFC04A1 resistors configuration



Revision history

Table 1. Document revision history

Date	Version	Changes
26-Mar-2018	1	Initial release.

Contents

1	FP-SNS-SMARTAG1 software description	2
1.1	Overview	2
1.2	Architecture	2
1.3	Folder structure	3
1.4	APIs	4
1.5	Sample application description	4
1.6	Android and iOS sample client application	9
1.6.1	Settings	9
1.6.2	Read tag	12
1.6.3	Chart	13
1.6.4	Single shot	15
2	System setup guide	17
2.1	Hardware description	17
2.1.1	STM32 Nucleo platform	17
2.1.2	X-NUCLEO-NFC04A1 expansion board	17
2.1.3	X-NUCLEO-IKS01A2 expansion board	18
2.1.4	STEVAl-SMARTAG1 evaluation board	19
2.2	Software description	20
2.3	Hardware and software setup	20
2.3.1	Hardware setup	20
2.3.2	Software setup	20
2.3.3	System setup guide	21
	Revision history	24

List of figures

Figure 1.	FP-SNS-SMARTAG1 software architecture	3
Figure 2.	FP-SNS-SMARTAG1 package folder structure	4
Figure 3.	Tera Term setup	5
Figure 4.	Initialization phase	6
Figure 5.	UART console output after auto-start range time	7
Figure 6.	UART console output NFC Rising	8
Figure 7.	UART console output NFC Falling.	9
Figure 8.	ST SmarTag (Android version) setting page 1	11
Figure 9.	ST SmarTag (Android version) setting page 2	12
Figure 10.	ST SmarTag (Android version) maximum and minimum value page.	13
Figure 11.	ST SmarTag (Android version) chart sensor plot page	14
Figure 12.	ST SmarTag (Android version) chart async page.	15
Figure 13.	ST SmarTag (Android version) single shot page	16
Figure 14.	STM32 Nucleo board	17
Figure 15.	X-NUCLEO-NFC04A1 expansion board.	18
Figure 16.	X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board	19
Figure 17.	STEVAL-SMARTAG1 evaluation board	19
Figure 18.	STM32 Nucleo plus X-NUCLEO-NFC04A1 board	21
Figure 19.	STM32 Nucleo plus X-NUCLEO-NFC04A1 plus X-NUCLEO-IKS01A2 boards	22
Figure 20.	STEVAL-SMARTAG1 evaluation board connected to ST-LINK/V2-1 via SWD.	22
Figure 21.	STEVAL-SMARTAG1 evaluation board directly connected to ST-LINK/V2-1.. . . .	23
Figure 22.	X-NUCLEO-NFC04A1 resistors configuration	23

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.

© 2018 STMicroelectronics – All rights reserved