Getting started with the STM32Cube function pack for IoT tracker node with Sigfox connectivity and sensors

Introduction

FP-ATR-SIGFOX1 is an STM32Cube function pack which lets you read data from environmental and motion sensors and send collected data via Sigfox connectivity.

Message sending is triggered via user button, timer event or movement detection by the on-board accelerometer.

The package implements low power profiles and related transitions to ensure long battery autonomy.

This software, together with the suggested combination of STM32 and ST devices, is intended particularly to develop asset tracking applications. Low-energy device geolocation is provided by the Sigfox infrastructure.

The software runs on the STM32 microcontroller and includes drivers for the S2-LP ultra-low power RF transceiver and the motion and environmental sensors.

1 Acronyms and abbreviations

Table 1. List of acronyms

Acronym	Description
API	Application programming interface
BSP	Board support package
CMSIS	Cortex® microcontroller software interface standard
GPS	Global positioning system
HAL	Hardware abstraction layer
I2C	Inter-integrated circuit
IoT	Internet of Things
MEMS	Micro electro-mechanical systems
RCZ	Radio control zone
RF	Radio frequency
SPI	Serial peripheral interface
USB	Universal serial bus

2 What is STM32Cube?

STM32Cube™ represents the STMicroelectronics initiative to make developers' lives easier by reducing development effort, time and cost. STM32Cube covers the STM32 portfolio.

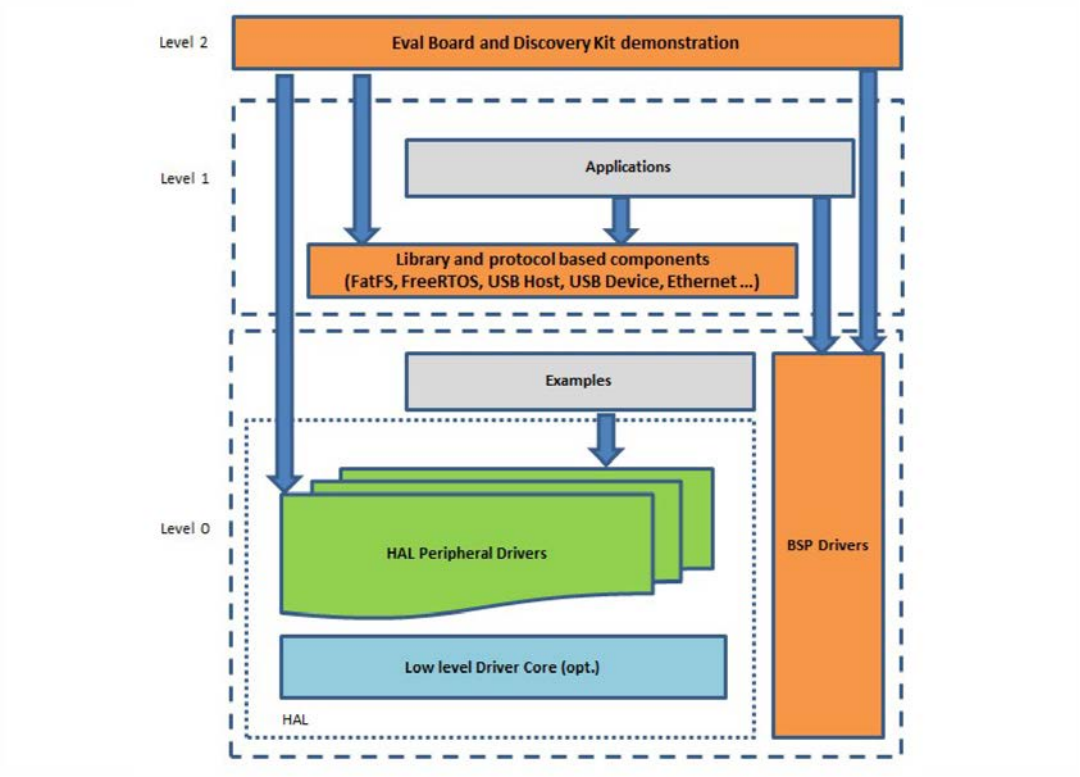
STM32Cube version 1.x includes:

- 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

2.1 STM32Cube architecture

The STM32Cube firmware solution is built around three independent levels that can easily interact with one another, as described in the diagram 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…); it is based on modular architecture allowing it to be easily ported on any hardware by just implementing the low level routines. It is 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 provides specific APIs to the external components of the BSP driver, and can be ported on any other board.
  - BSP driver: links the component driver to a specific board and provides a set of easy to use APIs. The API naming convention is BSP_FUNCT_Action(): e.g., BSP_LED_Init(), BSP_LED_On().

- 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). It provides generic, multi-instance and function-oriented APIs to help offload user application development time by providing ready to use processes. For example, for the communication peripherals (I²C, UART, etc.) it provides APIs for peripheral initialization and configuration, data transfer management based on polling, interrupt or DMA processes, and communication error management. The HAL Drivers APIs are split in two categories: generic APIs providing common, generic functions to all the STM32 series and extension APIs which provide special, customized functions for a specific family or a specific part number.
- Basic peripheral usage examples: this layer houses the examples built around the STM32 peripherals using the HAL and BSP resources only.

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 interaction among the components in this layer is performed directly by calling the feature APIs, while vertical interaction with low-level drivers is managed by specific callbacks and static macros implemented in the library system call interface. For example, FatFs implements the disk I/O driver to access a microSD drive or USB Mass Storage Class.
- Examples based on the middleware components: each middleware component comes with one or more examples (or applications) showing how to use it. Integration examples that use several middleware components are provided as well.

Level 2: This level is a single layer with a global, real-time and graphical demonstration based on the middleware service layer, the low level abstraction layer and basic peripheral usage applications for board-based functions.

3 FP-ATR-SIGFOX1 software expansion for STM32Cube

3.1 Overview

The FP-ATR-SIGFOX1 software package expands STM32Cube functionality.

The key features of the package are:

- Complete firmware to connect an IoT node to a Sigfox network, sending environmental sensor data
- Drivers for the S2-LP high performance ultra-low power RF transceiver
- Movement detection by the on-board accelerometer
- Middleware library supporting Sigfox connectivity from the X-CUBE-SFXS2LP1 software package
- Low-energy device geolocation service provided by the Sigfox infrastructure
- Sample implementation available for X-NUCLEO-S2868A1 and X-NUCLEO-IKS01A2 expansion boards connected to NUCLEO-L053R8 and NUCLEO-L476RG development boards
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

This software enables gathering environmental sensor data to transmit via Sigfox network connection. Received data can be displayed on the Sigfox backend and received by e-mail. Low energy device geolocation is provided by the Sigfox infrastructure.

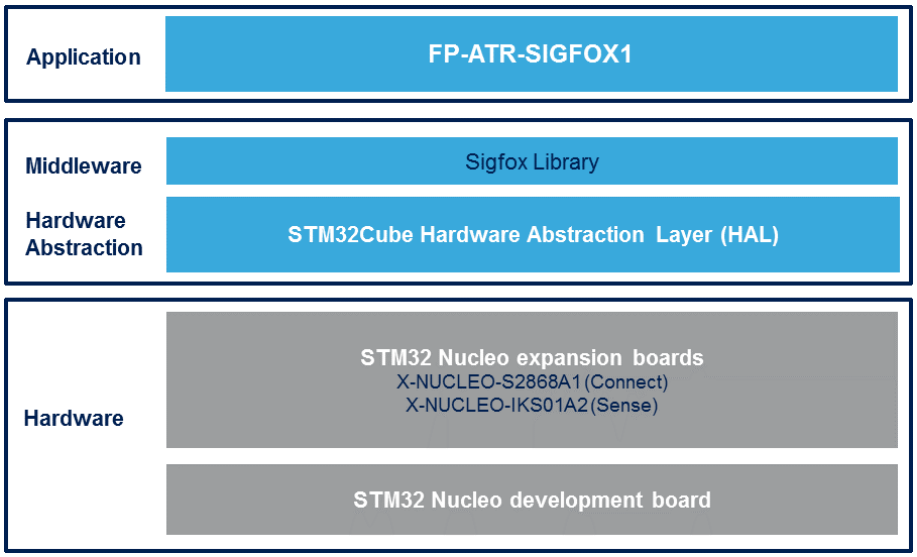
3.2 Architecture

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 sensors and S2-LP expansion boards and some middleware components for Sigfox communication.

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

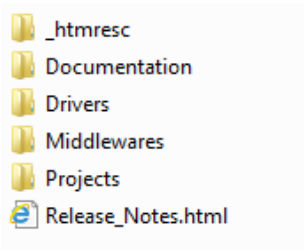
- the **STM32Cube HAL layer**, which provides a simple, generic, multi-instance set of application programming interfaces (APIs) to interact with the upper application, library and stack layers. It has generic and extension APIs and is directly built around a generic architecture and allows successive layers like the middleware layer to implement functions without requiring specific hardware configurations for a given microcontroller unit (MCU). This structure improves library code reusability and guarantees an easy portability on other devices.
- the **board support package** (BSP) layer supports all the peripherals on the STM32 Nucleo except the MCU. This limited set of APIs provides a programming interface for certain board-specific peripherals like the LED, the user button, etc. This interface also helps in identifying the specific board version.

Figure 2. **FP-ATR-SIGFOX1 software architecture**



3.3 Folder structure

Figure 3. **FP-ATR-SIGFOX1 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 and 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 ARM Cortex-M processor series.
- **Middlewares**: contains libraries and protocols related to the communication of application data with a Sigfox network.
- **Projects**: contains a sample application used to perform the Sigfox asset tracker sample application. This application is provided for the NUCLEO-L053R8 and NUCLEO-L476RG platforms with three development environments: IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM), and System Workbench for STM32 (SW4STM32).

3.4 APIs

Detailed technical information with full user API function and parameter description are in a compiled HTML file in the “Documentation” folder.

3.5 Sample application description

An example application for asset tracking using the STM32 Nucleo and the X-NUCLEO-S2868A1 and X-NUCLEO-IKS01A2 expansion boards is provided in the “Projects” directory. Ready to be built projects are available for multiple IDEs.

Pre-compiled binary executables are available for NUCLEO-L053R8 and NUCLEO-L476RG platforms and for each Sigfox radio control zone: “NUCLEO_Lx_ETSI” for RCZ1, “NUCLEO_Lx_ARIB” for RCZ3 and “NUCLEO_Lx_ALL” for all RCZs, where “Lx” can be L0 or L4.

Note: RCZ2 and RCZ4 are not supported.

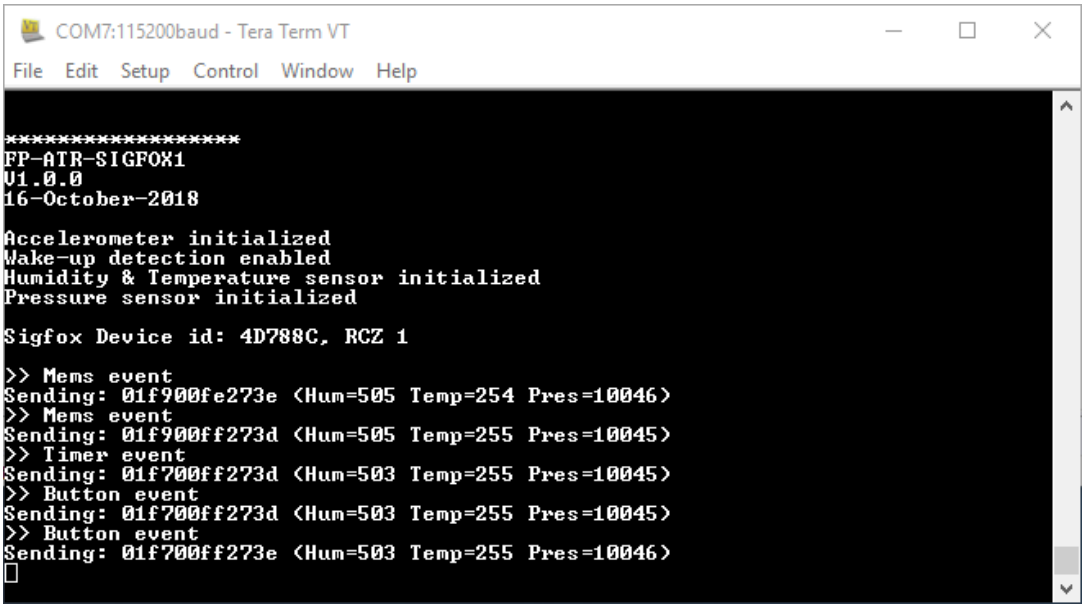
The user interface is provided via serial port, which needs to be configured with baud rate 115200, 8N1 parameters.

The application collects environmental sensor data (humidity, temperature and pressure) and sends them in a single 6-bytes Sigfox frame over the Sigfox network.

The message sending is triggered by one of the following events:

- User button pressure
- Wake-up detection by the on-board accelerometer sensor (this event can be generated by briefly shaking the boards)
- Timer expiration. By default, the timer is set to 3 minutes but it can be changed in the source code.

Figure 4. FP-ATR-SIGFOX1: example of serial interface

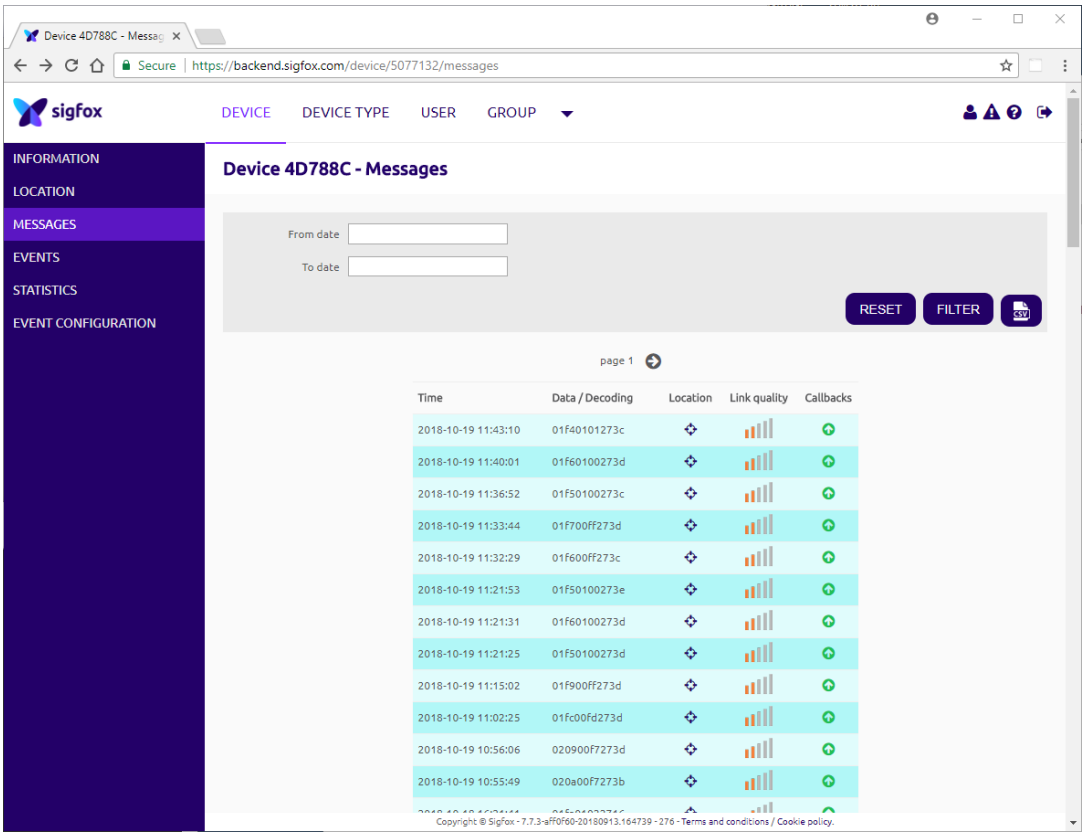


After the message is sent, the application switches to low-power state and remains in this state until the next message request is sent.

To send messages, Sigfox network coverage is needed. Alternatively, a Sigfox network emulator kit can be used.

To see the received data, connect to Sigfox backend (4.) and select the message list for your device.

Figure 5. **FP-ATR-SIGFOX1: example of message list received in Sigfox backend**



3.5.1 Sigfox geolocation

To activate the low-energy tracking provided by the Sigfox Geolocation service, you have to login on the Sigfox backend portal and follow the procedure below.

Step 1. Click on [**Device**] in the top menu bar.
A list of registered devices is displayed.

Step 2. Click on the [**Device Type**] name of the chosen device to obtain detailed information.

Step 3. Click on [**Callbacks**] in the left menu bar.
A list of service callbacks is displayed.

Step 4. Click on the [**New**] button in the top-right corner.
A list of possible callbacks is displayed.

Step 5. Click on [**Custom callback**].
A callback edition form is displayed.

3.5.2 Callback edition form

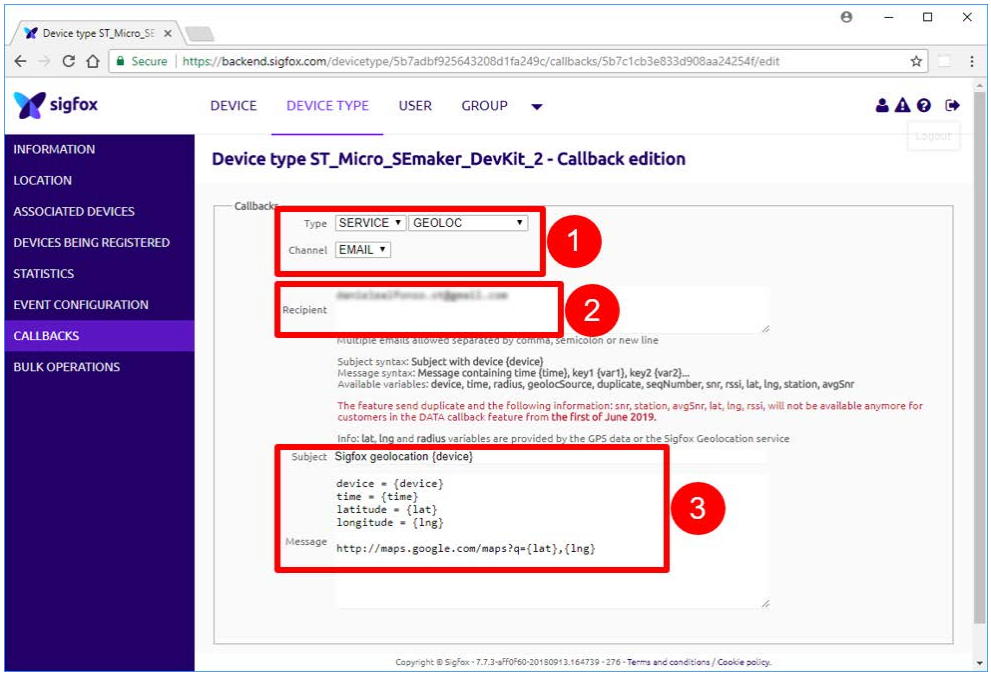
In the callback edition form, follow the procedure below.

Step 1. Select [**SERVICE**] as callback type, [**GEOLOC**] as service and [**EMAIL**] as channel.

Step 2. Provide a valid e-mail recipient.

Step 3. Provide e-mail message subject and body.

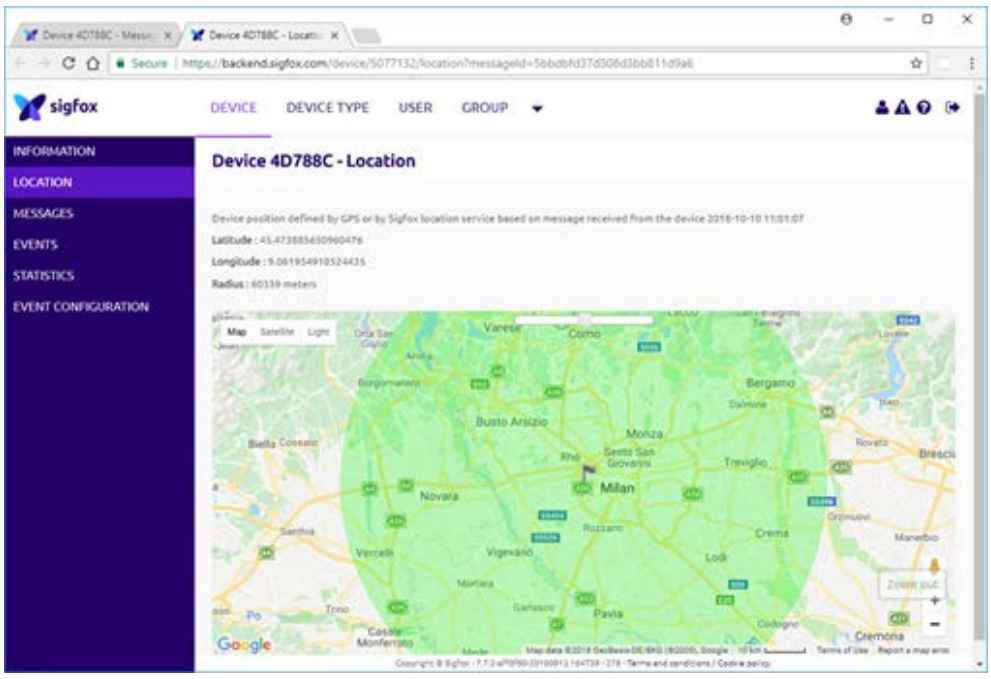
Figure 6. **Callback edition form**



After that, an e-mail message is sent to the indicated e-mail recipient with the chosen e-mail message body, including geographical coordinates and a link to Google maps, which can be clicked to open a map displaying the device location.

It is also possible to display the map directly on the Sigfox backend by selecting the device message list and then clicking on [**LOCATION**] for each message.

Figure 7. **Example of map in Sigfox backend**



3.6 Configuration and registration

Step 1. Register the Sigfox device as described in [2.].

Step 2. Follow the two steps: ST side registration and Sigfox side registration.

Step 3. Watch the tutorials on YouTube for more details:

- For ST side registration https://www.youtube.com/watch?v=JD6UE7ekRxE
- For Sigfox side registration https://www.youtube.com/watch?v=fTipdrGij7I

Step 4. Create a free account at http://backend.sigfox.com.

4 System setup guide

4.1 Hardware description

4.1.1 STM32 Nucleo platform

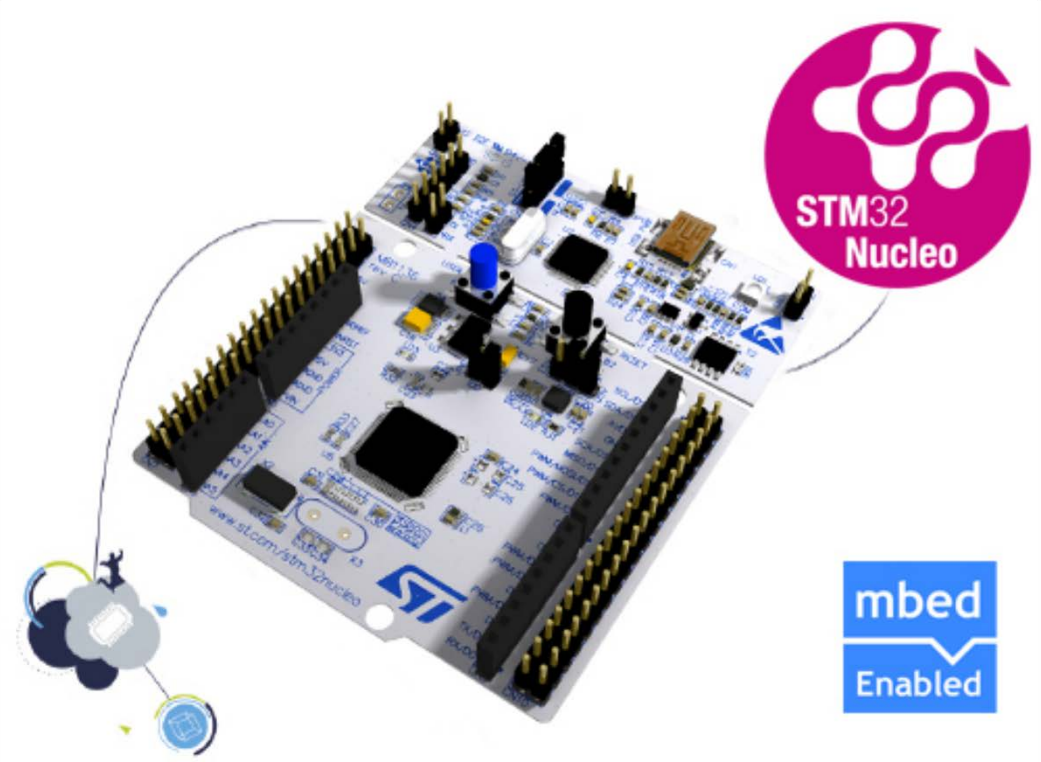
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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 8. STM32 Nucleo board



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

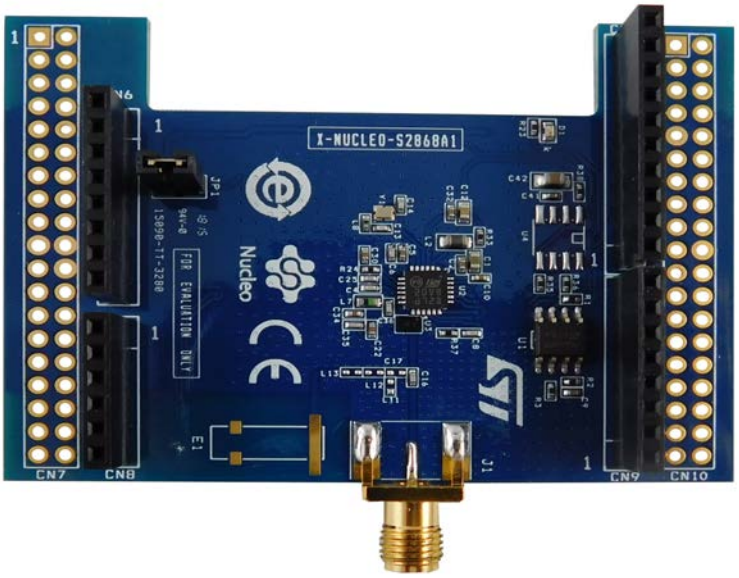
4.1.2 X-NUCLEO-S2868A1 expansion board

The X-NUCLEO-S2868A1 expansion board is based on the S2-LP radio and operates in the 868 MHz ISM frequency band.

The expansion board is compatible with ST morpho and Arduino UNO R3 connectors.

The X-NUCLEO-S2868A1 interfaces with the STM32 Nucleo microcontroller via SPI connections and GPIO pins. You can change some of the GPIOs by mounting or removing the resistors.

Figure 9. **X-NUCLEO-S2868A1 expansion board**



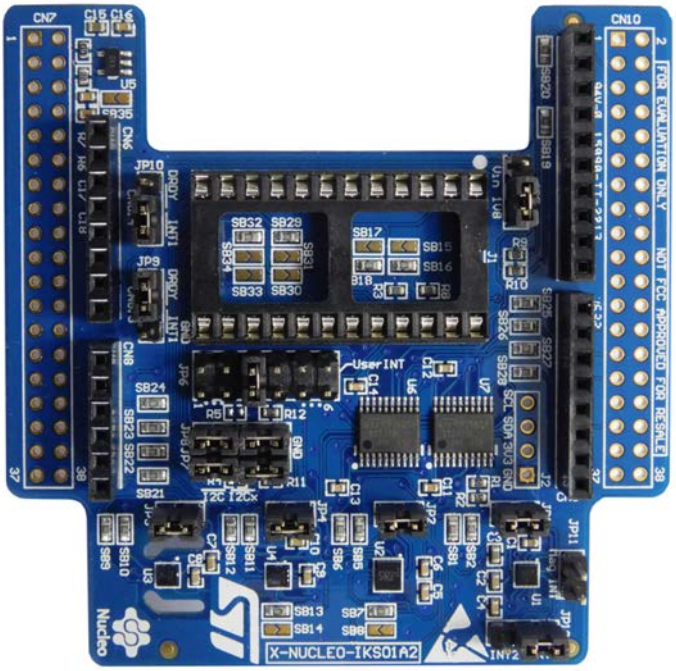
4.1.3 X-NUCLEO-IKS01A2 expansion board

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 10. **X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board**



4.2 Hardware setup

The following hardware components are needed:

1. One STM32 Nucleo development board (order code: NUCLEO-L053R8 or NUCLEO-L476RG).
2. One Sub-1 GHz 868 MHz RF expansion board (order code: X-NUCLEO-S2868A1).
3. One motion MEMS and environmental sensor expansion board (order code: X-NUCLEO-IKS01A2).
4. One USB type A to Mini USB Type B cable to connect the STM32 Nucleo board to the PC.

4.3 Software setup

The following software components are required for the setup of a suitable development environment to create applications for the STM32 Nucleo board with the sensor expansion board:

- FP-ATR-SIGFOX1: an STM32Cube function pack dedicated to asset tracking applications development. The firmware and related documentation are available on www.st.com.
- Development tool-chain and Compiler. The STM32Cube expansion software supports the three following environments to select from:
  - IAR Embedded Workbench for ARM® (IAR-EWARM) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM-STM32) toolchain + ST-LINK
  - System Workbench for STM32 (SW4STM32) + ST-LINK

4.4 System setup

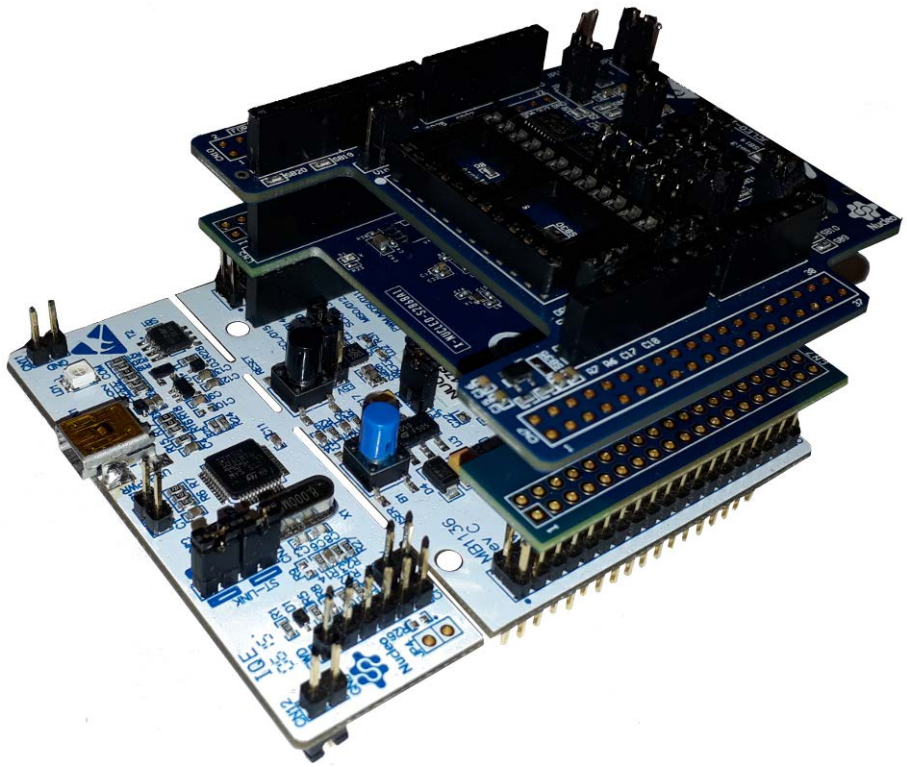
The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer.

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-S2868A1 and the X-NUCLEO-IKS01A2 expansion boards can be easily connected to the STM32 Nucleo through the Arduino UNO R3 extension connector.

The boards interfaces with the external STM32 microcontroller on STM32 Nucleo using serial peripheral interface (SPI) for the X-NUCLEO-S2868A1 and inter-integrated circuit (I²C) transport layer for the X-NUCLEO-IKS01A2.

Figure 11. STM32 Nucleo, X-NUCLEO-S2868A1 and X-NUCLEO-IKS01A2 stack



A References

1. UM2405: "Getting started with the X-NUCLEO-S2868A1 Sub-1 GHz 868 MHz RF expansion board based on S2-LP radio for STM32 Nucleo" at www.st.com
2. UM2169: "Getting started with the Sigfox S2-LP kit" at www.st.com
3. UM2121: "Getting started with the X-NUCLEO-IKS01A2 motion MEMS and environmental sensor expansion board for STM32 Nucleo" at www.st.com
4. https://backend.sigfox.com

Revision history

Table 2. **Document revision history**

Date	Version	Changes
05-Nov-2018	1	Initial release.

Contents

List of tables

List of figures