UM2341

User manual

Getting started with the FP-NET-6LPETH1 software package connecting 6LoW-PAN IoT nodes to the Internet via Ethernet networks

Description

FP-NET-6LPETH1 is an STM32 ODE function pack which lets you connect your IoT node in a 6LoWPAN wireless sensor network to the Internet via an Ethernet network.

The software, together with the suggested combination of STM32 and ST devices, can be used to develop applications such as smart city, home, building, lighting or remote monitoring.

The software runs on the STM32 microcontroller and includes drivers for SPIRIT1-based sub-1GHz RF communication modules (SPSGRF-868 or SPSGRF-915).

The software is based on STM32Cube technology and expands STM32Cube-based packages.

1 Acronyms and abbreviations

Table 1. List of acronyms

Acronym	Description
IDE	Integrated development environment
BSP	Board support package
DHCP	Dynamic host configuration protocol
HAL	Hardware abstraction layer
UDP	User datagram protocol
6LoWPAN	IPv6 over low power wireless personal area networks
RPL	Routing protocol for low power and lossy networks
MCU	Microcontroller unit
RF	Radio frequency
OS	Operating system
CoAP	Constrained application protocol
LWM2M	Lightweight machine to machine
IoT	Internet of things
MEMS	Micro electro-mechanical systems
GUI	Graphical user interface

2 FP-NET-6LPETH1 software expansion for STM32Cube

2.1 Overview

The FP-NET-6LPETH1 software package expands STM32Cube functionality.

The key features of the package are:

- Complete firmware to connect 6LoWPAN and Ethernet networks
- Middleware libraries to support Contiki OS and Contiki 6LoWPAN protocol stack 3.x
- Support for mesh networking technology via the standard RPL protocol
- Sample implementations available for X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5 expansion boards connected to a NUCLEO-F429ZI board
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

This software includes a sample application to connect 6LoWPAN network nodes to the Internet.

6LoWPAN network communication is carried out by the SPIRIT1 sub-1GHz radio (or expansion boards).

The sample application sends all the packets directed to the Internet via the Ethernet device available on the NUCLEO-F429ZI board.

Contiki OS technical details can be found at in the **Internals** section. For further information on Contiki APIs, refer to the documentation in the FP-NET-6LPETH1 package.

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 sub-1GHz RF communication expansion boards.

The drivers abstract low-level details of the hardware and allow the middleware components and applications to access sensor data in a hardware-independent manner to access and control the SPIRIT1-based sub-1GHz RF communication modules (SPSGRF-868 or SPSGRF-915).

The package includes a middleware library to support a 6LoWPAN stack, along with a sample application to send the IP packets from the 6LoWPAN network to the Ethernet network and vice versa.

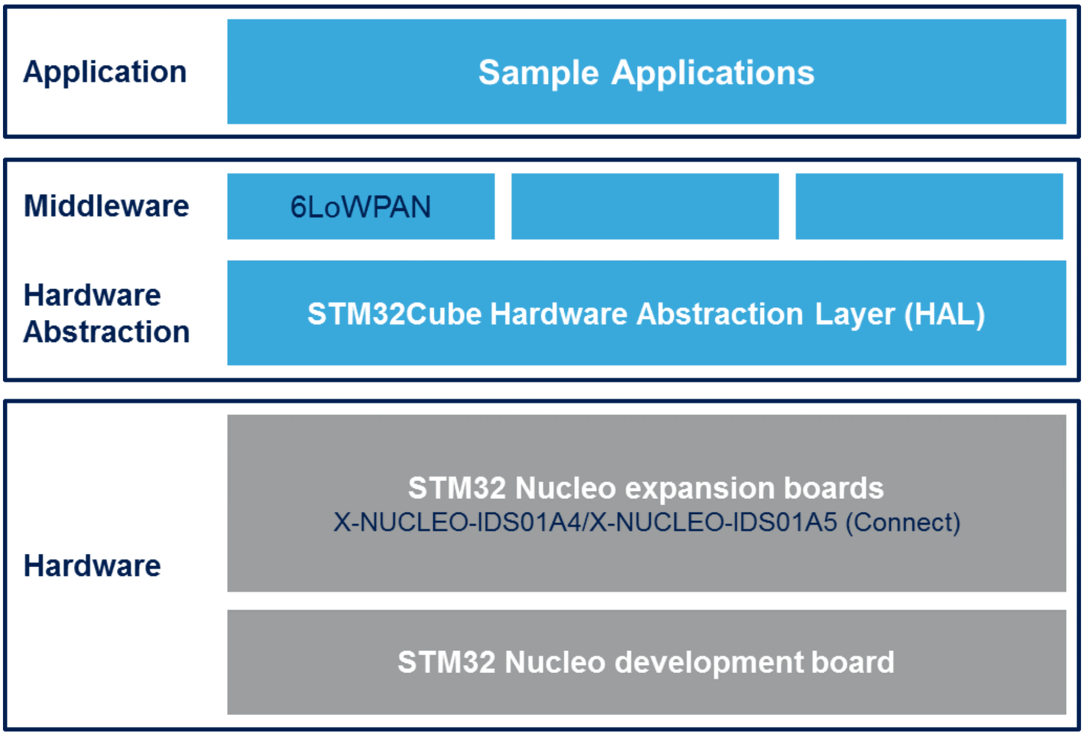
The device implements a “6to4” IP router functionality which allows converting the 6LoWPAN network IPv6 packets into the Ethernet network IPv4 packets .

The application software accesses the X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5 expansion boards via:

The **STM32Cube HAL** driver 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 facilitates portability to other devices.

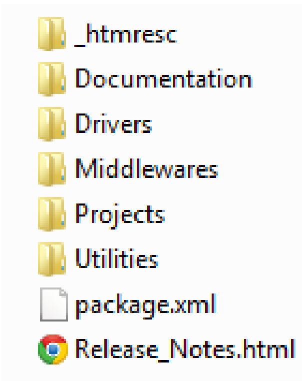
The **board support package** (BSP) layer supports all the peripherals on the STM32 Nucleo except the MCU. This limited set of APIs provides a programming interface for certain board-specific peripherals like the LED, the user button, etc. This interface also helps in identifying the specific board version.

Figure 1. **FP-NET-6LPETH1 software architecture**



2.3 Folder structure

Figure 2. **FP-NET-6LPETH1 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 for the Contiki OS with 6LoWPAN stack.
- Projects: contains a sample application to bridge a 6LoWPAN network with an Ethernet network. The projects are built for NUCLEO-F429ZI boards and the following development environments:
  1. IAR Embedded Workbench for ARM
  2. RealView Microcontroller Development Kit (MDK-ARM)

3. System Workbench for STM32 (SW4STM32).

2.4 APIs

Detailed function and parameter descriptions regarding the user APIs are available in a compiled HTML file in the software package Documentation folder.

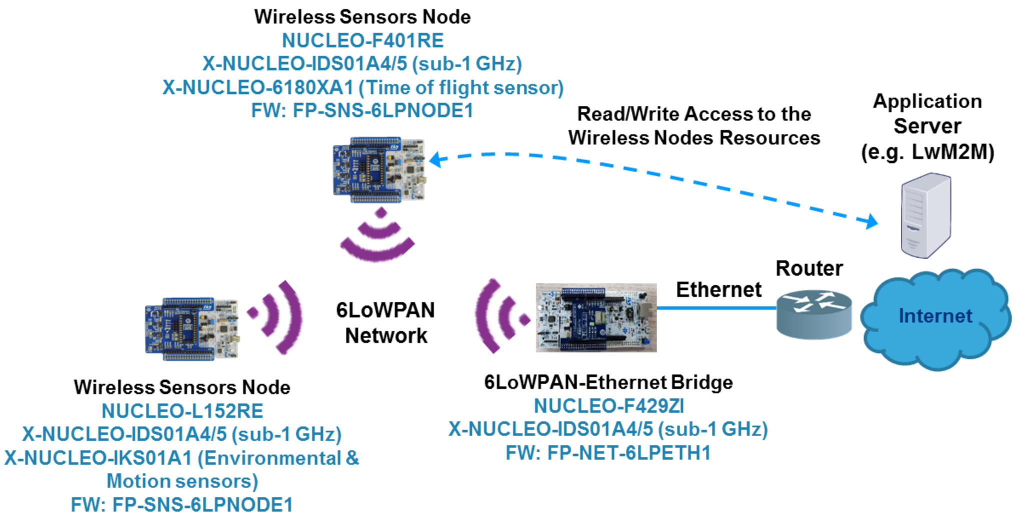
2.5 Sample application description

2.5.1 System overview

A sample application called “IP64Router”, located in the ‘Projects’directory, with ready-to-build projects for multiple IDEs, is used to:

- translate IPv6 packets to IPv4 packets through NAT64 technology;
- create a bridge between a 6LoWPAN network and the Internet using Contiki OS and the Ethernet transceiver.

Figure 3. Overall system architecture



The figure above shows:

- On the left side, a 6LoWPAN network of wireless sensor nodes, implemented using STM32Nucleo boards with sub-1GHz and sensor expansion boards. Refer to FP-SNS-6LPNODE1 for the node firmware and the related documentation.
- On the right side, an Ethernet network connected to the Internet through a standard router.

The IPv4 address for the Ethernet interface is automatically detected by the DHCP protocol.

Important: To ensure system correct functionality, the router must run a DHCP server.

The 6LoWPAN Ethernet bridge, implemented with an STM32 Nucleo board (equipped with a sub-1GHz expansion board) running the FP-NET-6LPETH1 firmware package, connects these networks.

Application servers or connected clients on the Internet can be used to access data from the wireless sensor nodes. For example, an OMA Lightweight M2M (LWM2M) server can be used to access the LWM2M client running on the nodes and to perform device management operation or sensor data reading (see 2.5.2 Wireless sensor node setup using FP-SNS-6LPNODE1 (optional)).

2.5.1.1 How to run the IP64Router application

Procedure

Step 1. Compile the project using one of the supported IDEs (see 3.2 Software requirements)

Step 2. Plug the STM32 Nucleo board to your PC

A USB mass storage location is created.

Step 3. Program the firmware on the STM32 Nucleo board by copying (or dragging and dropping) the binary file to the USB mass storage location.

- If you are using a PC with Linux, the STM32 Nucleo can be found in the media folder with the name **NUCLEO_F401RE**; for example, if the created mass storage location is **/media/NUCLEO_F401RE**, the command to program the board with a binary file named **my_firmware.bin** is `cp my_firmware.bin /media/NUCLEO_F401RE`.
- Alternatively, you can program the STM32 Nucleo board directly through one of the supported development toolchains (refer to the corresponding toolchain user manual for further information).

Step 4. Open a serial line monitor utility.

Step 5. Select the serial port name to which the board is connected and configure the parameters as shown below.

- Baud Rate = 115200
- Parity = None
- Bits= 8
- Stopbits = 1

2.5.2 Wireless sensor node setup using FP-SNS-6LPNODE1 (optional)

To enable user deployment and testing of end-to-end solutions between wireless sensor nodes and Internet servers, sample node applications are available as source code and pre-compiled, ready-to-use binaries for STM32 Nucleo platforms equipped with expansion boards.

The software package to implement wireless sensor nodes is the FP-SNS-6LPNODE1 function pack (refer to UM2100 and to the FP-SNS-6LPNODE1 quick start guide, available at www.st.com, for detailed setup information and node description).

The sample applications export different resources on the basis of the different expansion boards stacked on the STM32 Nucleo board.

The supported nodes configurations are (see UM2100, Section 3.3.1: "Hardware setup", for more details):

1. **NUCLEO-F401RE plus X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5**: the node can expose common resources like LEDs and user buttons.
2. **NUCLEO-F401RE plus X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5 plus X-NUCLEO-IKS01A1**: the node can expose resources from the MEMS sensor expansion board (temperature, humidity and acceleration sensors).
3. **NUCLEO-F401RE plus X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5 plus X-NUCLEO-6180XA1**: the node can expose resources from the FlightSense™ expansion board (proximity sensor).

The FP-SNS-6LPNODE1 function pack sample applications show how a node can connect to a remote server with the OMA lightweight M2M (LWM2M) standard; this technology uses CoAP (over UDP) to publish a node resources in a standard (LWM2M) format and make them available online.

The nodes attempt to connect to a public online server, Leshan, which is a Java implementation of an LWM2M server.

Leshan also implements a GUI that communicates with the server via a REST API (see http://www.eclipse.org/leshan/).

Note: *The Leshan public LWM2M server is used only for evaluation purposes, to show end-to-end connectivity between a wireless sensor node and an Internet connected server, thanks to the 6LoWPAN-to-Ethernet bridge using the FP-NET-6LPETH1 software package.*

2.5.3 How to connect a remote wireless node to the Internet

Procedure

Step 1. Power the STM32 Nucleo board using a Mini-B USB cable connected to the PC.

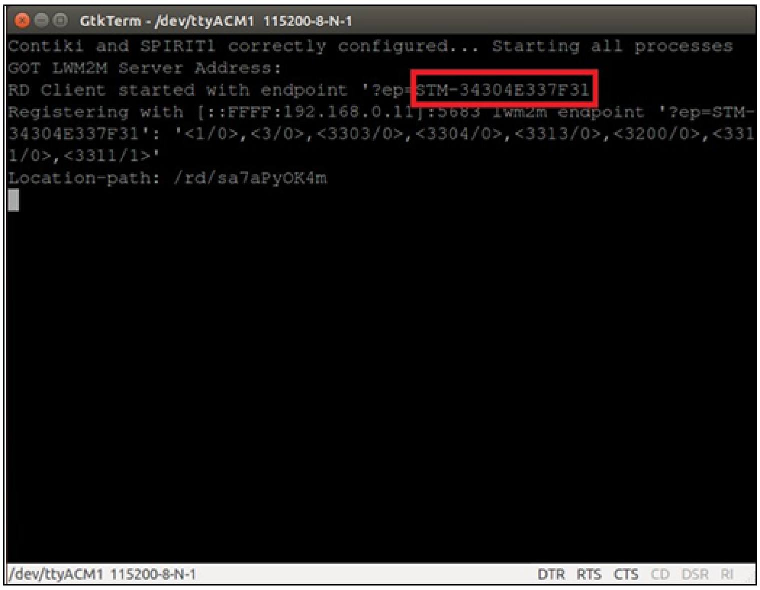
Step 2. Program the firmware on the STM32 Nucleo board:
you can copy (or drag and drop) the binary file (in the /Utility/Binary folder) to the USB mass storage that is automatically created when you connect the STM32 Nucleo board to your PC.

Step 3. Open a serial line monitor utility.

Step 4. Select the serial port name to which the board is connected and configure the parameters as shown below.
- Baud Rate = 115200
- Parity = None
- Bits= 8
- Stopbits = 1

Step 5. Press the RESET (black) button on the STM32 Nucleo board and wait for the node to complete the registration.

Figure 4. Client node registered on the remote server



Step 6. Open Leshan

Figure 5. Leshan server: homepage

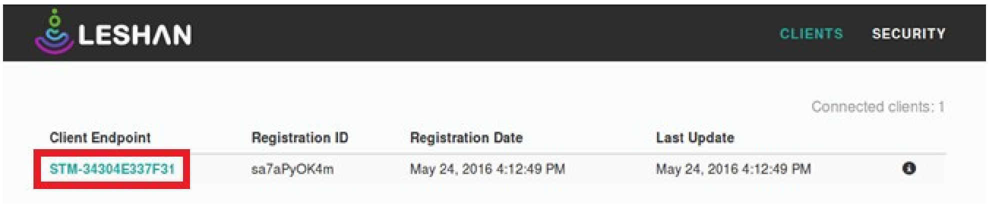
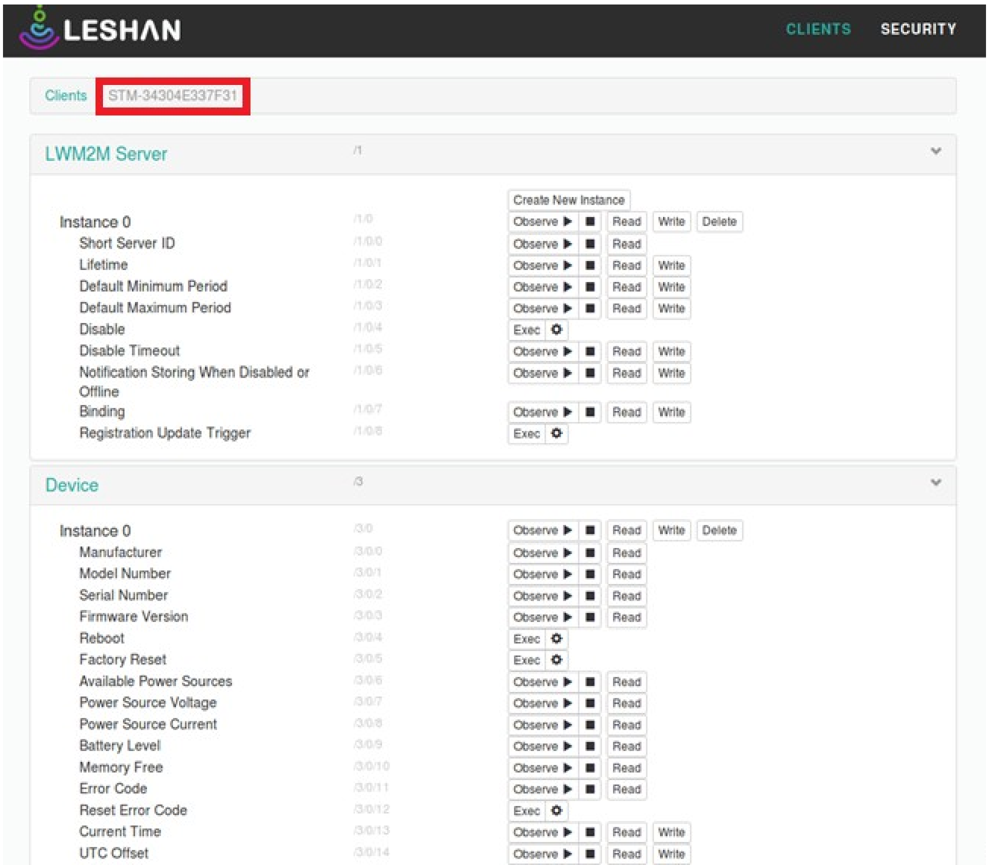
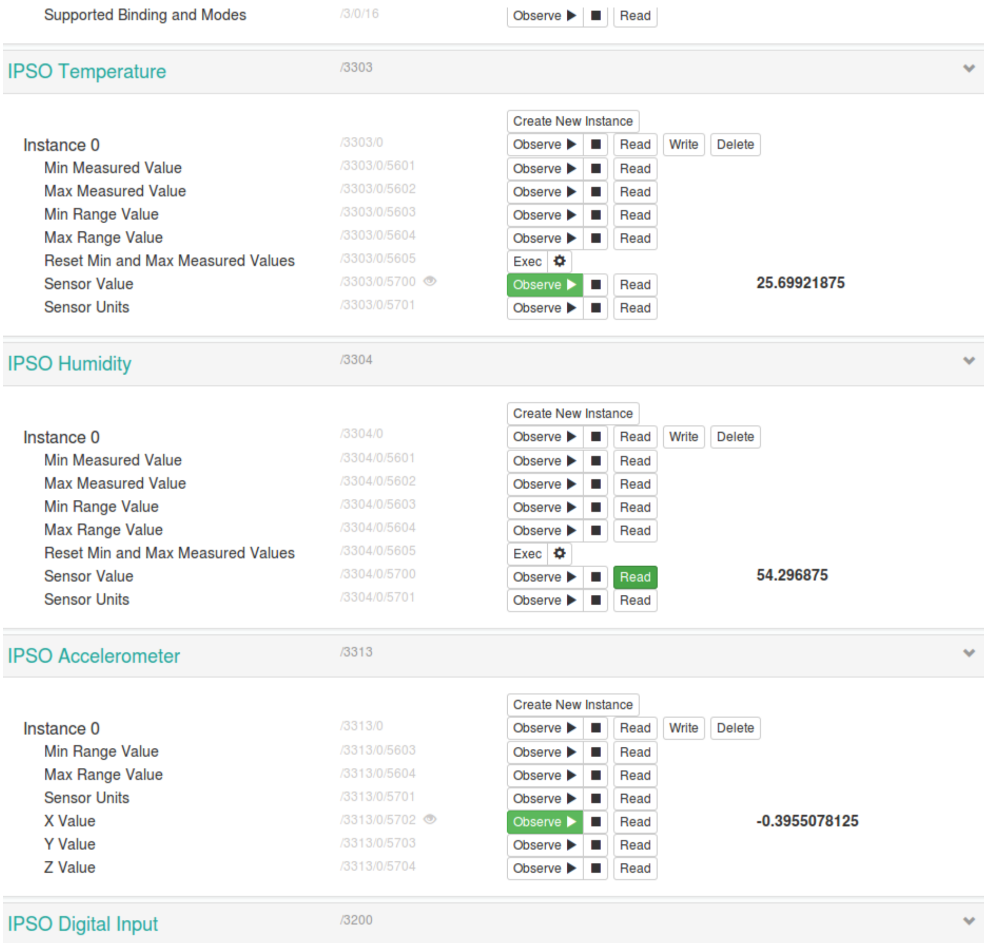


Figure 6. Leshan server: client homepage



Step 7. Observe or read resources

Figure 7. **Leshan server: supported binding and modes**



3 System setup guide

3.1 Hardware description

3.1.1 STM32 Nucleo-144 platform

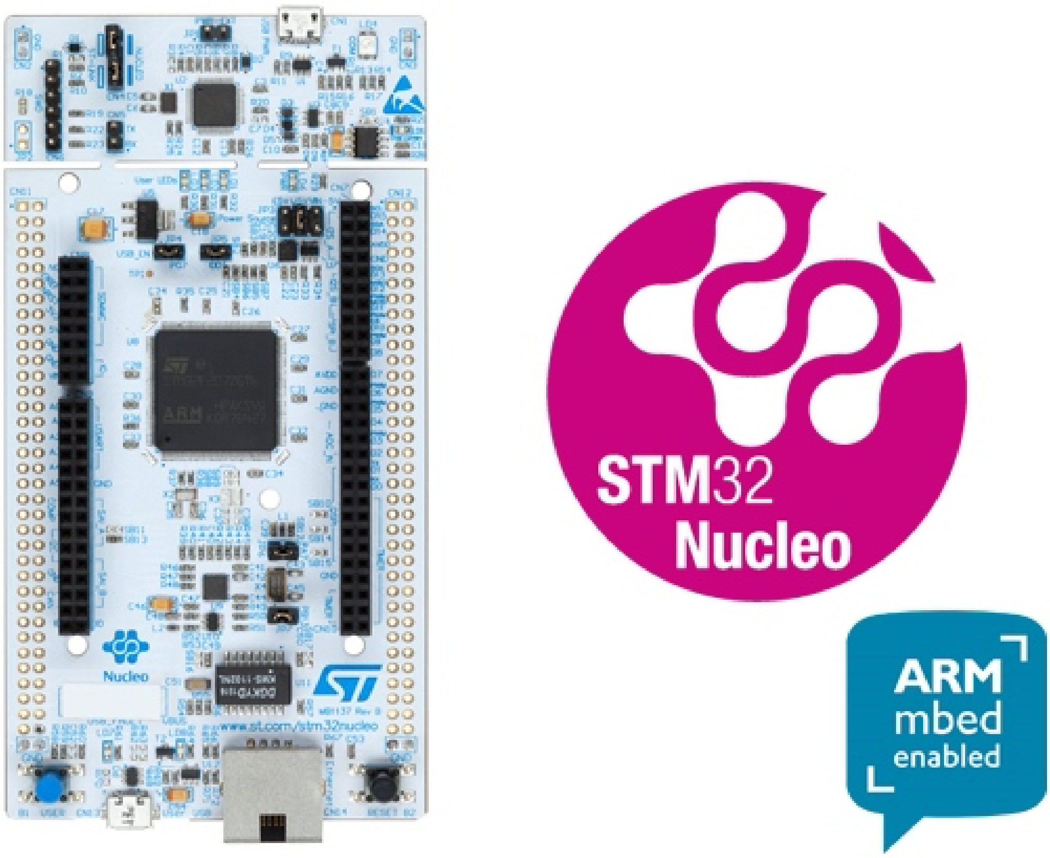
The STM32 Nucleo-144 boards provide an affordable and flexible way to build prototypes by choosing from the STM32 microcontroller various combinations of performance and power consumption.

The ST Zio connector, which extends the Arduino™ Uno V3 connectivity, and the ST morpho headers extend the Nucleo open development platform functionality with a wide choice of specialized shields.

The STM32 Nucleo-144 boards integrate the ST-LINK/V2-1 debugger/programmer and do not require any separate probes.

The STM32 Nucleo-144 boards come with the STM32 comprehensive free software libraries and samples available in the STM32Cube package.

Figure 8. STM32 Nucleo-144 board



Information regarding the STM32 Nucleo-144 boards is available at www.st.com/stm32nucleo.

3.1.2 X-NUCLEO-IDS01A4 and X-NUCLEO-IDS01A5 expansion boards

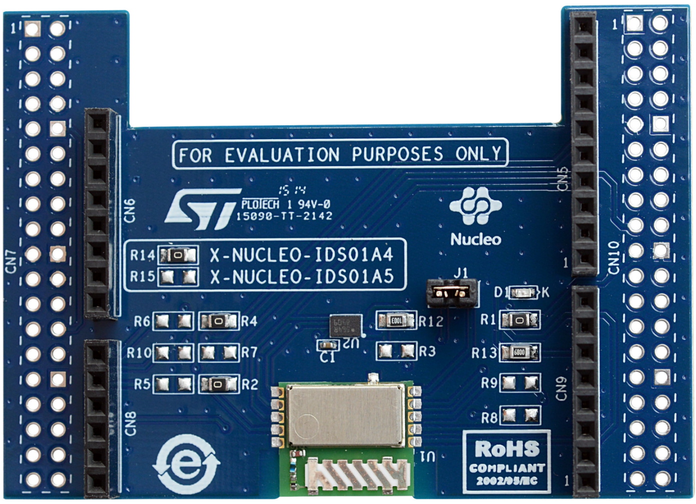
The X-NUCLEO-IDS01A4 and X-NUCLEO-IDS01A5 expansion boards provide a platform to test the features and capabilities of the SPSGRF modules, based on the SPIRIT1 low data rate, low power, sub-1 GHz transceiver device.

These expansion boards can be plugged into the Arduino UNO R3 connectors of any STM32 Nucleo board. The user can mount ST morpho connectors if required. Other expansion boards can easily be stacked to allow evaluation of different devices using sub-1 GHz communication.

The boards are equipped with the following features:

- Onboard SPSGRF module based on the SPIRIT1 sub-1 GHz transceiver device
- SPI EEPROM for saving parameters
- LED for user interface
- Jumper at 3V3 for checking the current consumption of the expansion board

Figure 9. X-NUCLEO-IDS01A4 / X-NUCLEO-IDS01A5 expansion board



3.2 Software requirements

The following software components are required to set up a suitable development environment for creating applications for the STM32 Nucleo board with RF expansion board:

- FP-NET-6LPETH1 software, available on www.st.com.
- Development tool-chain and Compiler. The STM32Cube expansion software supports the three following environments:
  – IAR Embedded Workbench for ARM® (IAR-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

3.3.1 Hardware requirements

For the implementation of the bridge, the following hardware components are required:

- One STM32 Nucleo-144 development platform (order code: NUCLEO-F429ZI)
- One SPIRIT1 expansion board (order code: X-NUCLEO-IDS01A4 (for 868 MHz), or X-NUCLEO-IDS01A5 (for 915 MHz))
- One USB type A to micro USB type B cable to connect the STM32 Nucleo to the PC

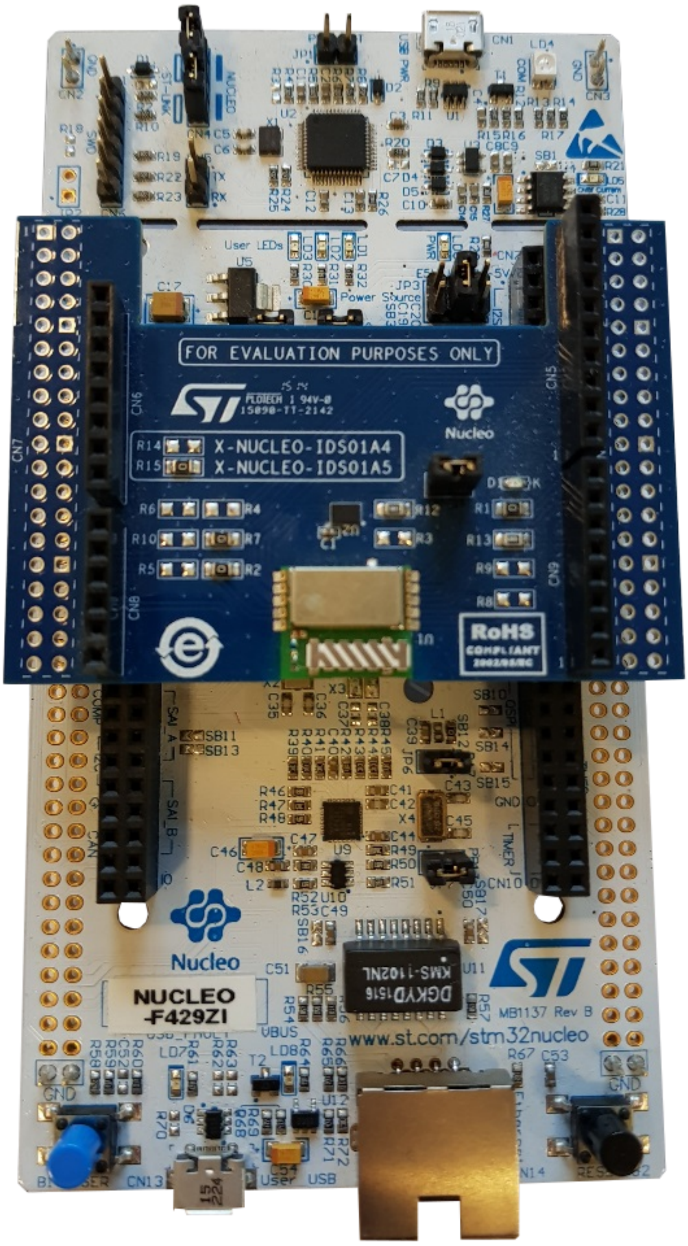
3.3.2 System setup guide

This section describes how to setup different hardware parts before writing and executing an application on the STM32 Nucleo-144 board with RF expansion board.

The STM32 Nucleo-144 board integrates the ST-LINK/V2-1 debugger/programmer; you can download the ST-LINK/V2-1 USB driver at STSW-LINK009.

The X-NUCLEO-IDS01A4 (or X-NUCLEO-IDS01A5) expansion board is easily connected to the STM32 Nucleo-144 board Arduino connector, as shown below.

Figure 10. STM32 Nucleo-144 plus the sub-1GHz expansion board



Important:

To avoid resource usage conflict between the X-NUCLEO-IDS01A4 (or X-NUCLEO-IDS01A5) expansion board and the NUCLEO-F429ZI board, the following two modifications must be done on the boards (as shown in the pictures below):

- *On the NUCLEO-F429ZI, the 0 Ω resistor at position SB121 must be moved to position SB122.*
- *On the X-NUCLEO-IDS01A4 (or X-NUCLEO-IDS01A5), the 0 Ω resistor at position R4 must be moved to R7.*

Figure 11. NUCLEO-F429ZI board (bottom view): resistor position changed

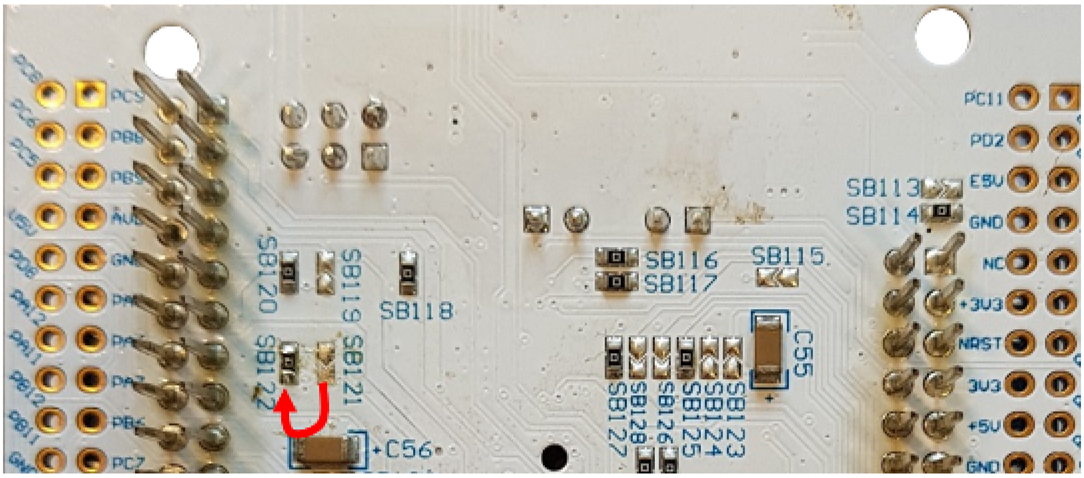


Figure 12. X-NUCLEO-IDS01A4 (or X-NUCLEO-IDS01A5) expansion board: resistor position changed

4 References

Freely available at www.st.com:

- UM2100: "Getting started with FP-SNS-6LPNODE1 software for IoT sensor node connection to 6LoWPAN networks using sub-1GHz RF"

Revision history

Table 2. Document revision history

Date	Version	Changes
21-Dec-2017	1	Initial release.

Contents

List of tables

List of figures