UM2102
User manual

Getting started with the FP-NET-6LPBLE1 function pack for 6LoWPAN IoT node connection to a smartphone via BLE interface

Introduction

FP-NET-6LPBLE1 is an STM32 ODE function pack. This package provides application layer bridging functionality to connect a 6LoWPAN network consisting of SPIRIT1-based sub-1 GHz RF communication sensor nodes to a smartphone or tablet device in a BLE network, allowing control of the sensor nodes with an application.

This package allows jumpstarting end-to-end IoT development to save time in the integration of essential functions required for 6LoWPAN node connection to smartphones using the BLE interface.

The software runs on the STM32 microcontroller and includes drivers for the BLE module (SPBTLE-RF) and SPIRIT1-based sub-1 GHz RF communication modules (SPSGRF-868 or SPSGRF-915); the software also comes with binary firmware ready for wireless sensor nodes.

Contents

List of tables

List of figures

1 Acronyms and abbreviations

Table 1: List of acronyms

Acronym	Description
BLE	Bluetooth low energy
6LoWPAN	IPv6 over low power wireless personal area networks
IoT	internet of things
IDE	integrated development environment
BSP	board support package
HAL	hardware abstraction layer
UDP	user datagram protocol
MCU	microcontroller unit
MEMS	micro electro-mechanical systems

2 FP-NET-6LPBLE1 software expansion for STM32Cube

2.1 Overview

The key features of the package are:

- Firmware package with Contiki 6LoWPAN protocol stack 3.x and related applications running on an STM32 Nucleo board with sub-1 GHz RF, BLE connectivity and optional sensor expansion board
- Middleware library with BLE and Contiki 6LoWPAN protocol stack 3.x
- Support for mesh networking technology through the standard RPL protocol
- Sample application to connect a border router to a sensor node using 6LoWPAN mesh network technology and to display mesh network tree and sensor data on an Android™ device
- Sample implementation for X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5 and X-NUCLEO-IDB05A1 boards connected to a NUCLEO-F401RE board
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

2.2 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 X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5 and X-NUCLEO-IDB05A1 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 BLE module (SPBTLE-RF) and SPIRIT1 based sub-1 GHz RF communication modules (SPSGRF-868 or SPSGRF-915).

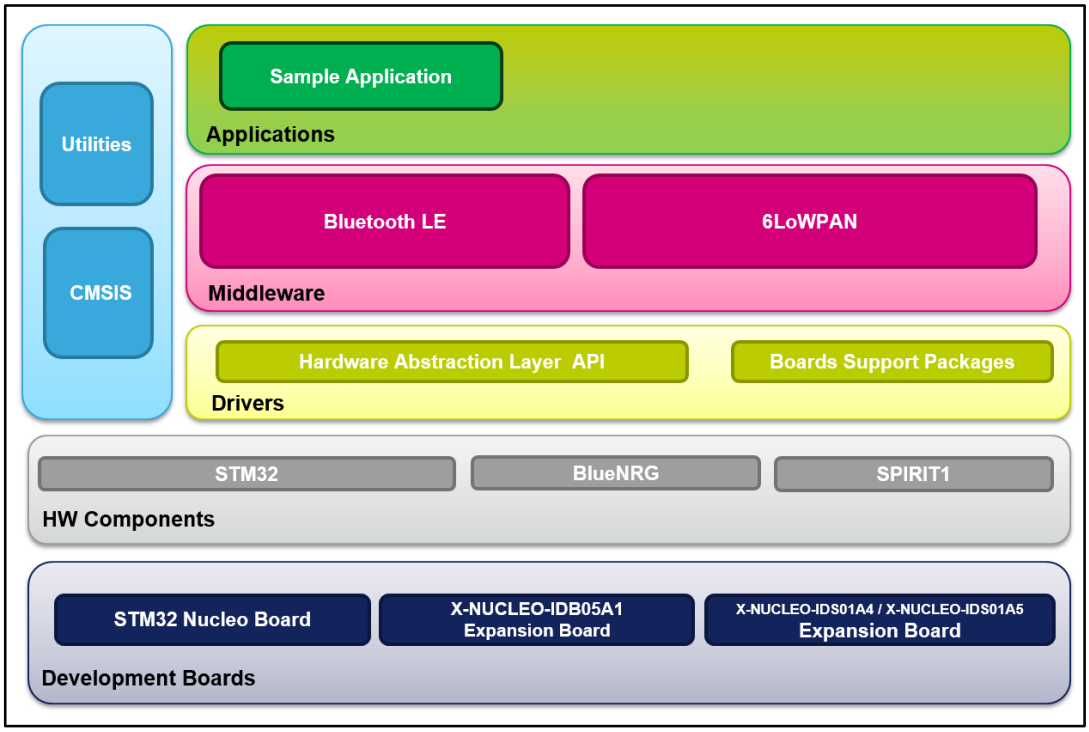
The package also includes some middleware libraries to support BLE and 6LoWPAN stacks, along with sample application for accessing sensors and actuators resources on the 6LoWPAN nodes, using Embedded Bluetooth low energy protocol stack such as GAP and GATT. Developers can use it to prototype end-to-end IoT applications.

The software layers used by the application software to access and use both the X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5 and X-NUCLEO-IDB05A1 expansion boards are the following:

STM32Cube HAL layer: consists of a simple, generic and multi-instance set of APIs (application programming interfaces) to interact with the upper application, library and stack layers. These generic and extension APIs are directly built around a generic architecture and allow layers like the middleware layer built on top of them to implement their functions without requiring specific hardware configuration information for given microcontroller units (MCU). This structure improves library code re-usability and guarantees easy portability across other devices.

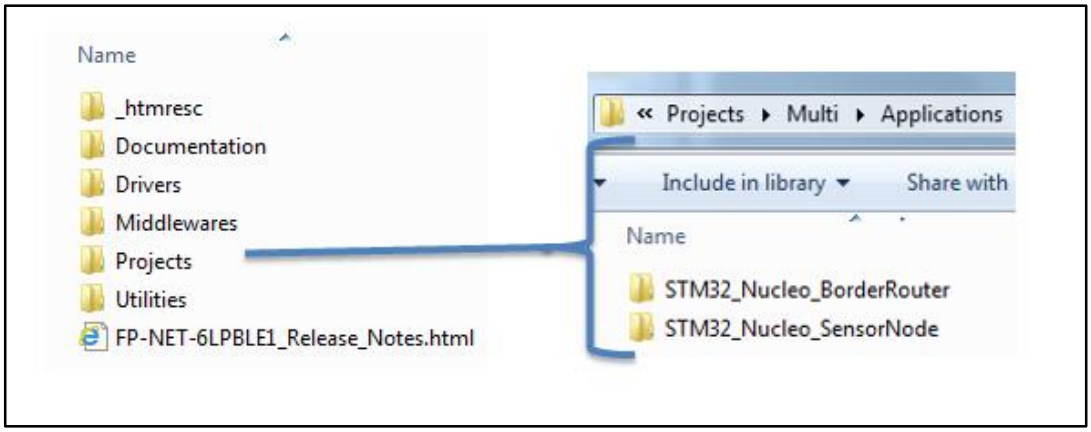
Board support package (BSP) layer: supports the peripherals on the STM32 Nucleo board, except the MCU. This limited set of APIs provides a programming interface for certain board specific peripherals like the LED and user button. This interface also helps in identifying the specific board version.

Figure 1: FP-NET-6LPBLE1 software architecture



2.3 Folder structure

Figure 2: FP-NET-6LPBLE1 package folder structure



The following folders are included in the software package:

- **Documentation**: contains a compiled HTML file generated from the source code, detailing the software components and APIs.
- **Drivers**: contains the HAL drivers, the board specific drivers for each supported board or hardware platform (including the onboard component drivers) and the CMSIS vendor-independent hardware abstraction layer for the Cortex-M processor series.
- **Middlewares**: contains libraries for the Contiki OS and the interface for the BLE expansion software.
- **Projects**: contains the border router and sensor node sample application that implements application-level functions to bridge a 6LoWPAN network with a BLE using the NUCLEO-F401RE development board.

- **Utilities**: contains the Android™-ready (.apk file) application allowing a node in the 6LoWPAN network to connect to the BLE and share its sensor and actuator resources.

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 Android application

A local user can access the nodes using Bluetooth connectivity on smartphones with an Android application to pair the phone to border router and access the node functionality.

The Android application is available in Utilities folder (\Utilities\android_apk\ Smart_Sensor_V1.1.apk)

The application has been validated for the following smartphones:

- Samsung galaxy S5 (Android V5.0)
- Nexus 5
- Nexus 7 (Android V5.1)
- Nexus 9
- Nexus 6P
- Redmi Note3 (Android V5.1.1)
- Gionee s plus

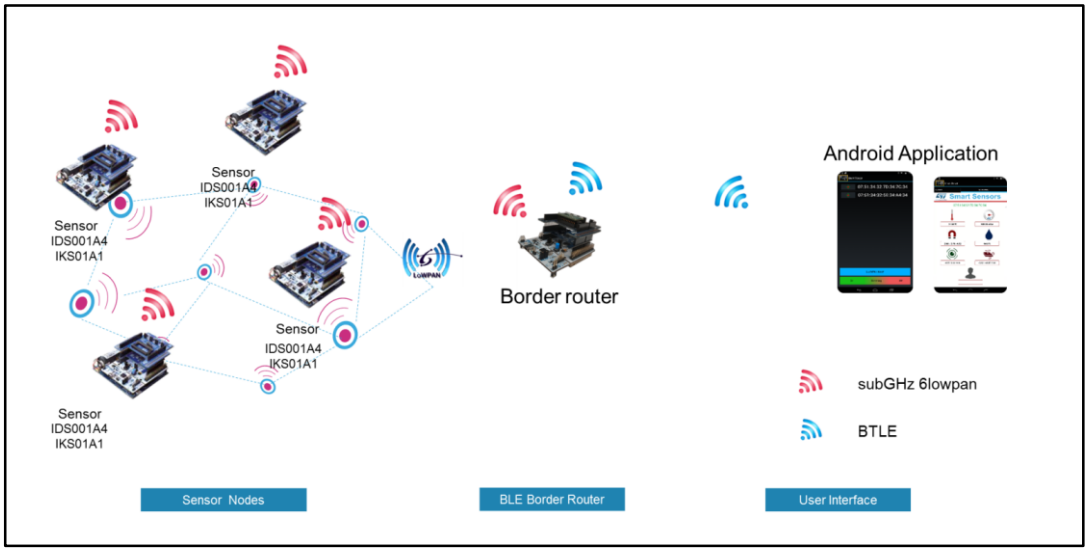
2.6 Border router sample application

A sample border router application for the X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5 and X-NUCLEO-IDB05A1 expansion boards on top of a NUCLEO-F401RE board is located in the Projects directory. Ready-to-build projects are available for multiple IDEs.

This application lets you create a bridge between a 6LoWPAN network and the BLE network using Contiki OS and the BLE module. The application supports the UDP protocol and uses NAT64 technology to translate IPv6 to IPv4 packets. The BLE bridge seamlessly forward every packet from the node in the mesh network to your smartphone.

The following figure outlines a system connecting an application on the smartphone to nodes in a 6LoWPAN network via a 6LoWPAN to BLE bridge.

Figure 3: System connecting smartphone app to nodes via border router



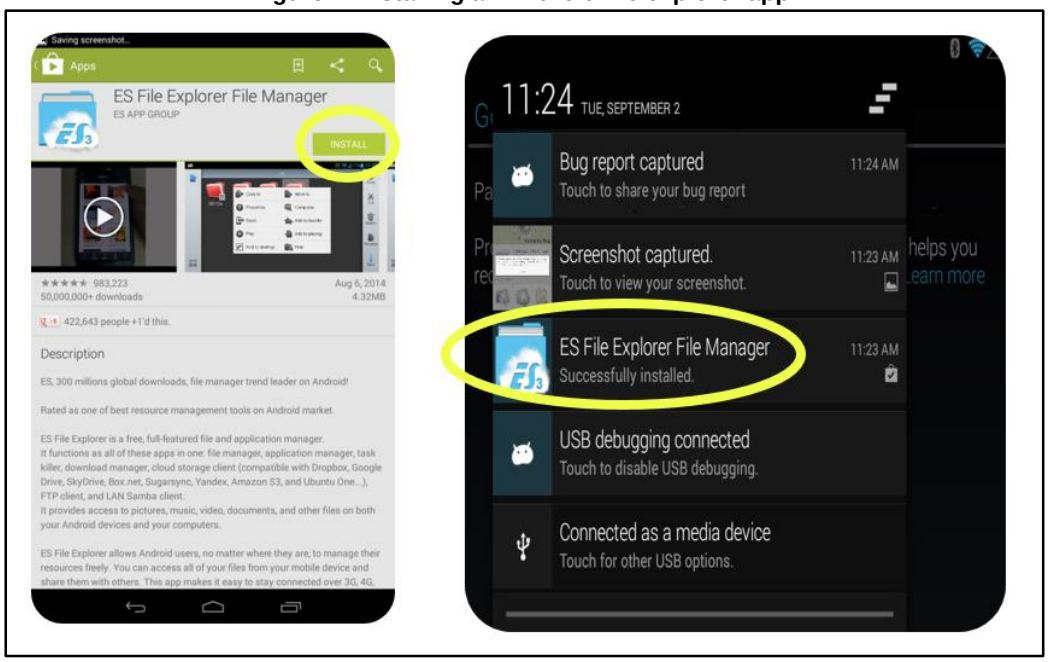
2.6.1 Connecting the border router to your smartphone

The application has been validated on the following smartphones:

- Samsung galaxy S5
- Nexus 5
- Nexus 7
- Nexus 9
- Nexus 6P
- Redmi Note3
- Gionee s plus

1 If you do not have one, download and install an appropriate file explorer app from Google Play™ store.

Figure 4: Installing an Android file explorer app



2 Copy Smart_Sensor.apk to your Android smartphone memory.

3 Use the File Explorer app to locate Smart_Sensor.apk and install it on your phone.

4 Launch the application.

2.7 Sensor node sample application

The sensor node sample application is a pre-compiled and ready-to-use binary file, which implements a bridge between a 6LoWPAN network and a BLE network so you can deploy and test your end-to-end application involving wireless sensor nodes.

The sensor node system (X-NUCLEO-IDS01A4 + X-NUCLEO-IDS01A1 + NUCLEO-F401RE) is connected to the border router system (X-NUCLEO-IDB05A1 + X-NUCLEO-IDS01A4 + NUCLEO-F401RE) via 6LoWPAN network.

The node transmits data at an interval of 2 (default) to 180 seconds, which is configurable from the smartphone device; the border router receives node data passes it to the smartphone via BLE interface.

Users have access to:

- node IPv6 addresses
- sensor values
- the mesh network tree

2.7.1 Connecting the sensor node to the smartphone

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

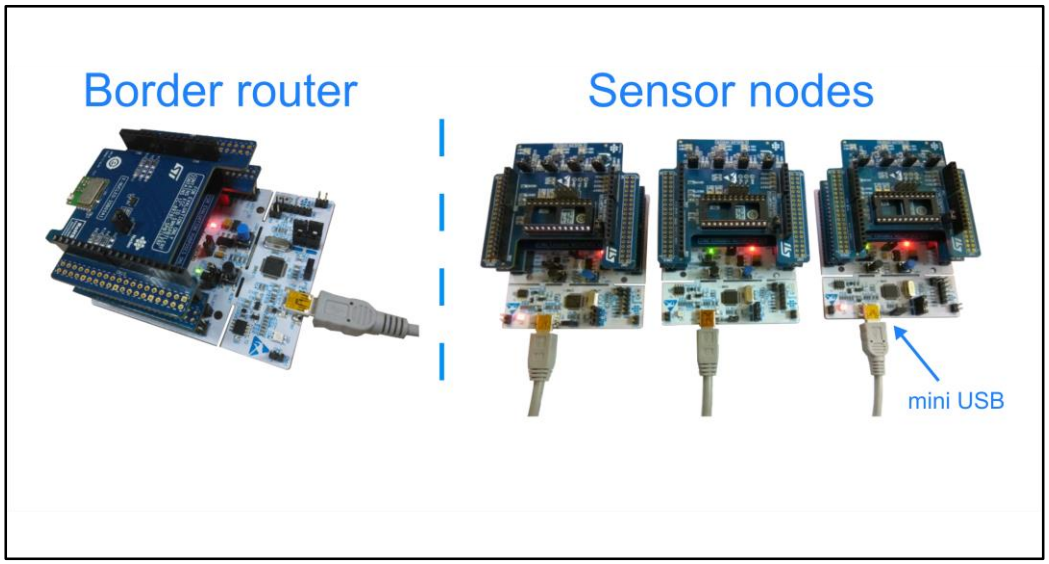
2 Program the firmware onto the STM32 Nucleo board by copying (drag and drop) the binary file created in the USB mass storage when you plug the STM32 Nucleo board to a PC.

The binaries can be found in:

- Border router:
  Applications/STM32_Nucleo_BorderRouter/Binary/STM32F401RE-Nucleo/STM32F401RE-Nucleo_BorderRouter.bin
- Sensor node:
  Applications/STM32_Nucleo_SensorNode/Binary/STM32F401RE-Nucleo/STM32F401RE-Nucleo_SensorNode.bin

3 Reset the MCU via the black button on the STM32 Nucleo board.

Figure 5: Connecting nodes to PC via Mini-USB



3 System setup guide

3.1 Hardware description

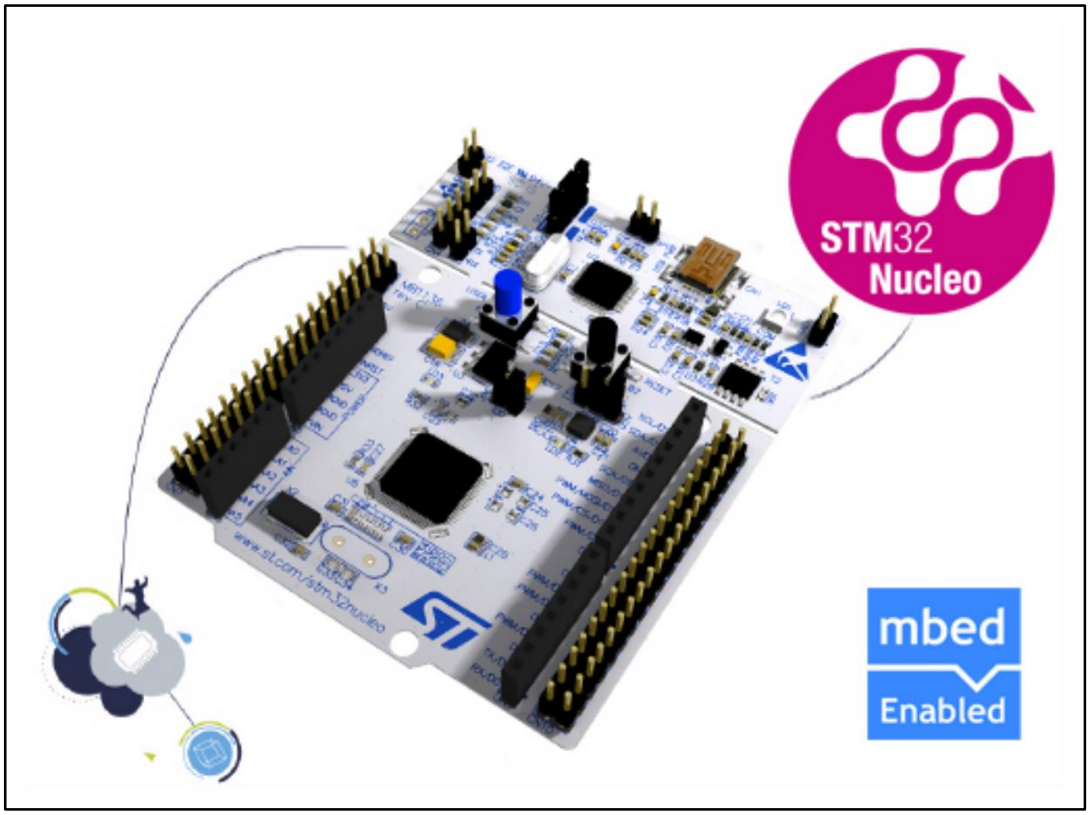
This section describes the hardware components required to develop a sensor-based application with sensor nodes and border routers.

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 6: STM32 Nucleo board

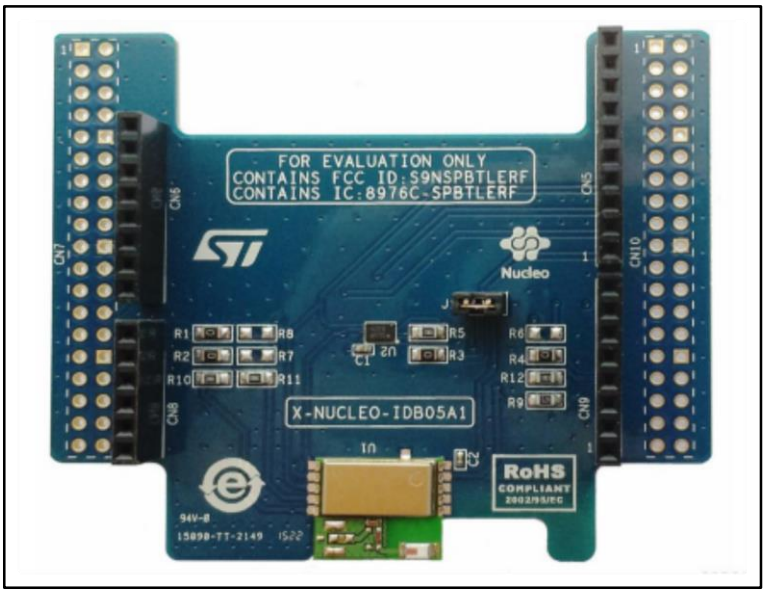


3.1.2 X-NUCLEO-IDB05A1 expansion board

The X-NUCLEO-IDB05A1 is a Bluetooth low energy evaluation board based on the SPBTLE-RF BlueNRG-MS RF module to allow expansion of the STM32 Nucleo boards. The SPBTLE-RF module is FCC (FCC ID: S9NSPBTLERF) and IC certified (IC: 8976C-

SPBTLERF). The BlueNRG-MS is a very low power Bluetooth low energy (BLE) single-mode network processor, compliant with Bluetooth specification v4.2. X-NUCLEO-IDB05A1 is compatible with the ST morpho and Arduino™ UNO R3 connector layout. This expansion board can be plugged into the Arduino UNO R3 connectors of any STM32 Nucleo board.

Figure 7: X-NUCLEO-IDB05A1 expansion board



Information about the X-NUCLEO-IDB05A1 expansion board is available on www.st.com at *http://www.st.com/x-nucleo*

3.1.3 X-NUCLEO-IDS01A4 and X-NUCLEO-IDS01A5 expansion board

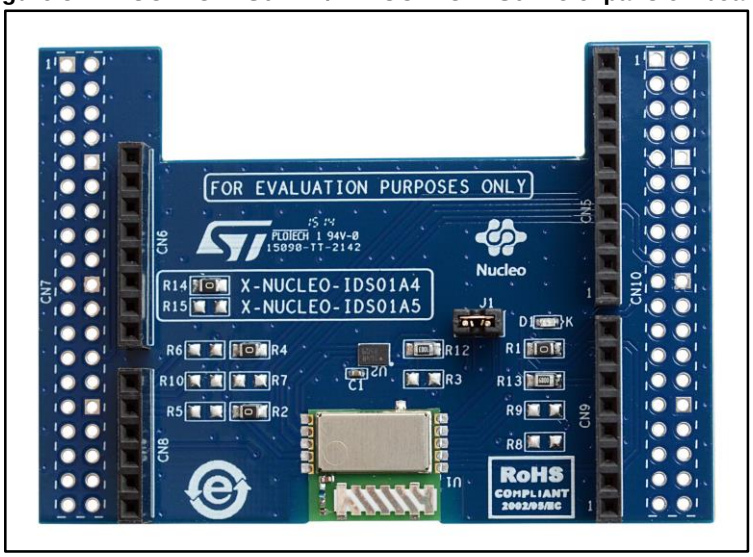
The X-NUCLEO-IDS01A4 and X-NUCLEO-IDS01A5 expansion boards provide a platform to test the features and capabilities of the SPSGRF modules, based on the SPIRIT1 low data rate, low power, sub-1 GHz transceiver device.

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 8: X-NUCLEO-IDS01A4 / X-NUCLEO-IDS01A5 expansion board

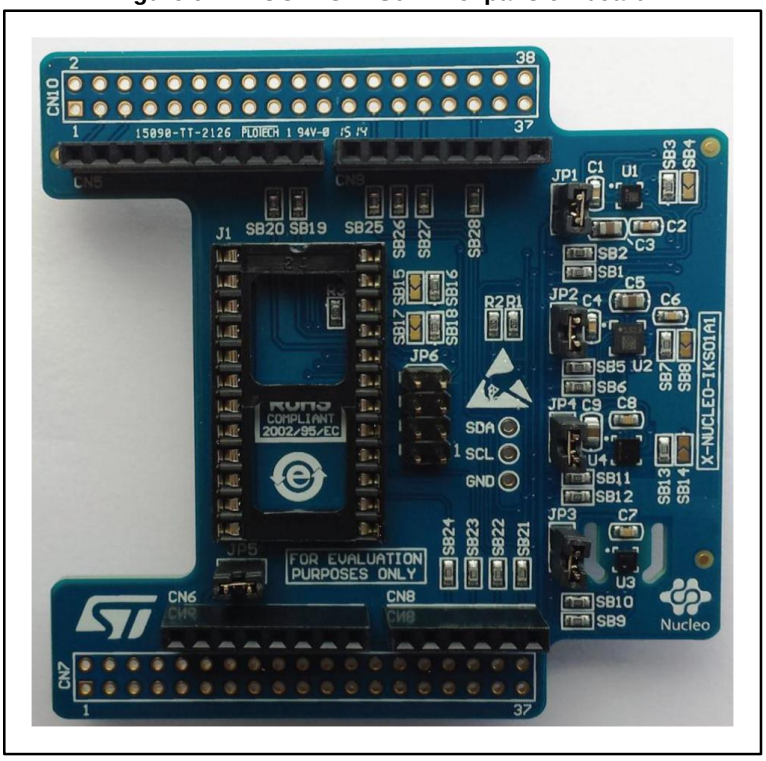


3.1.4 X-NUCLEO-IKS01A1 expansion board

The X-NUCLEO-IKS01A1 is a sensor expansion board for the STM32 Nucleo system. It is also compatible with Arduino UNO R3 connector layout and is designed around STMicroelectronics humidity (HTS221), pressure (LPS25HB) and motion (LIS3MDL and LSM6DS0) sensing devices. The X-NUCLEO-IKS01A1 interfaces with the STM32 MCU via the I²C pin, and the user can change the default I²C port and the device IRQ by changing a resistor on the evaluation board.

You can attach the LSM6DS3 DIL24 expansion component and use it instead of the one of the LSM6DS0 sensors.

Figure 9: X-NUCLEO-IKS01A1 expansion board



Information about the X-NUCLEO-IKS01A1 expansion board is available on www.st.com at http://www.st.com/x-nucleo.

3.2 Software description

To set up a suitable development environment for creating applications for the STM32 Nucleo board plus RF expansion board, the following software components are required:

1. FP-NET-6LPBLE1 software available on www.st.com.
2. 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.

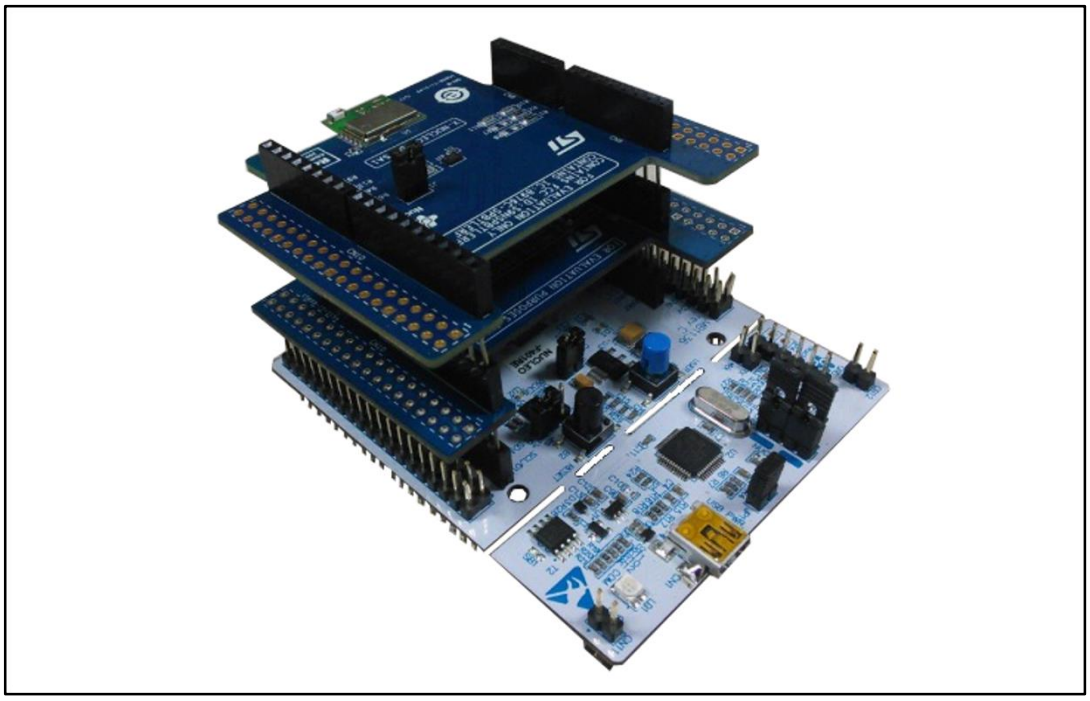
3.3.1 Hardware setup

3.3.1.1 Border router system

Hardware requirements:

1. One STM32 Nucleo development platform (order code: NUCLEO-F401RE)
2. One Bluetooth low energy expansion board (order code: X-NUCLEO-IDB05A1)
3. One Sub-1 GHz RF expansion board order code: X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5)
4. USB type A to Mini-B USB cable to connect the STM32 Nucleo board to the PC

Figure 10: STM32 Nucleo plus X-NUCLEO-IDS01A4 plus X-NUCLEO-IDB05A1



The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. You can download the ST-LINK/V2-1 USB driver by searching the STSW-LINK009 software on *www.st.com*

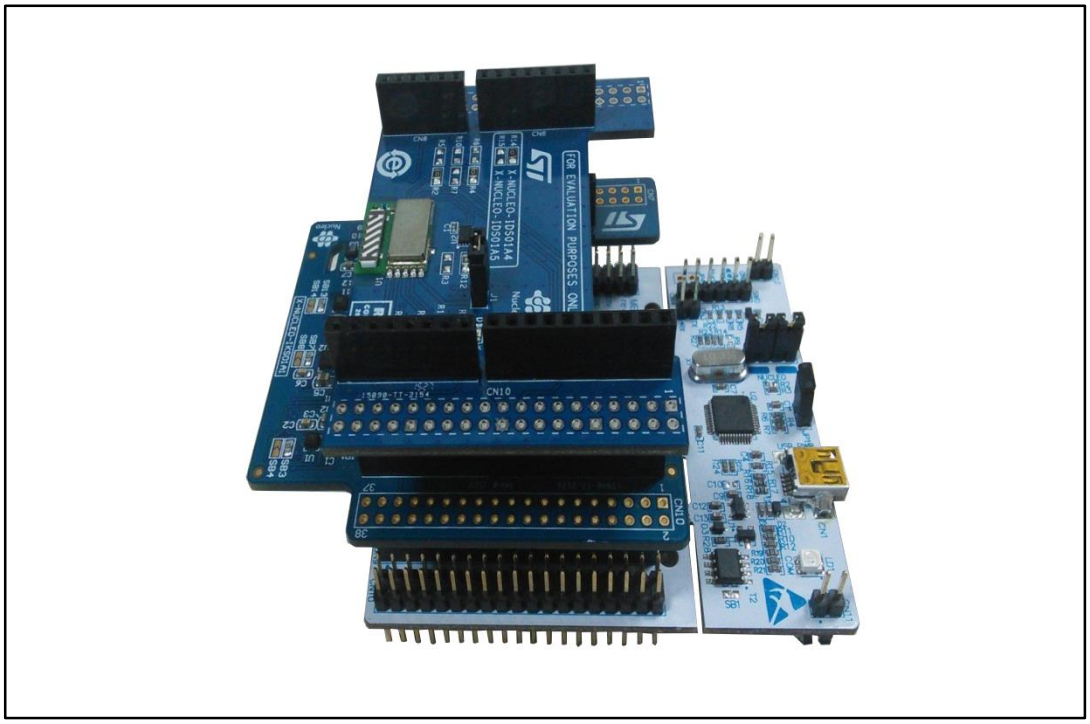
1. Connect the X-NUCLEO-IDB05A1 BLE expansion board and X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5 SPIRIT1 expansion board to the NUCLEO-F401RE development board through the Arduino UNO R3 extension connector, as shown in the figure above.
2. The path to the border router application binary file is: \Applications\STM32_Nucleo_BorderRouter\Binary\STM32F401RE-Nucleo\ STM32F401RE-Nucleo_BorderRouter.bin
3. Reset the MCU via the black button on the STM32 Nucleo board.

3.3.1.2 Sensor node system

Hardware requirements:

1. One STM32 Nucleo development platform (order code: NUCLEO-F401RE)
2. One sub-1 GHz RF expansion board (order code: X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5)
3. One sensor expansion board (order code: X-NUCLEO-IKS01A1)
4. USB type A to Mini-B USB cable to connect the STM32 Nucleo board to the PC

Figure 11: STM32 Nucleo plus X-NUCLEO-IKS01A1 plus X-NUCLEO-IDS01A4



The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. You can download the ST-LINK/V2-1 USB driver by searching the STSW-LINK009 software on *www.st.com*.

1. Connect the X-NUCLEO-IKS01A1 sensor expansion board and X-NUCLEO-IDS01A4 or X-NUCLEO-IDS01A5 SPIRIT1 expansion board to the NUCLEO-F401RE development board through the Arduino UNO R3 extension connector, as shown in the figure above

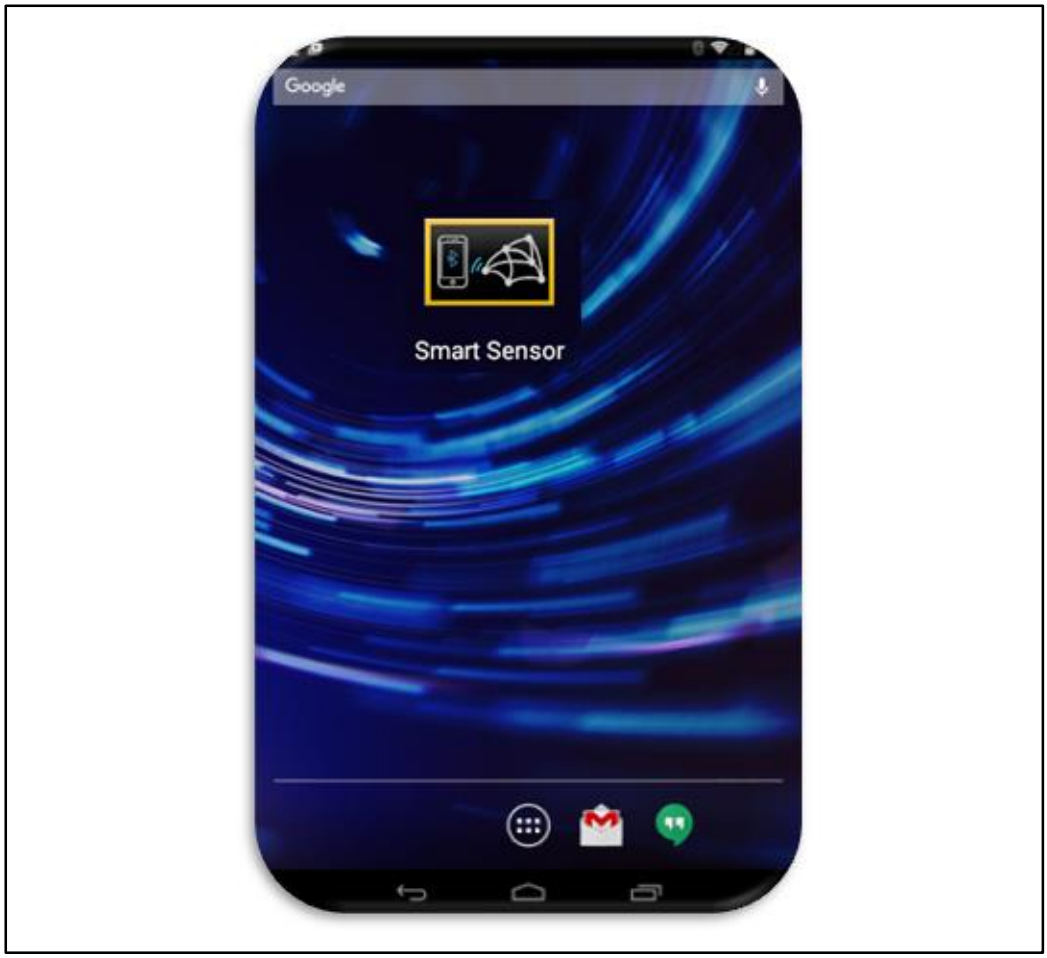
2. The path to the sensor node application binary file is:
   \Applications\STM32_Nucleo_SensorNode\Binary\STM32F401RE-Nucleo\
   STM32F401RE-Nucleo_SensorNode.bin
3. Reset the MCU via the black button on the STM32 Nucleo board.

3.3.2 Software setup

1 Use any Android™ file explore app to find and run the Smart Sensor.apk installation file.

The Smart Sensor app becomes available on your device.

Figure 12: Icon of Smart Sensor Android application

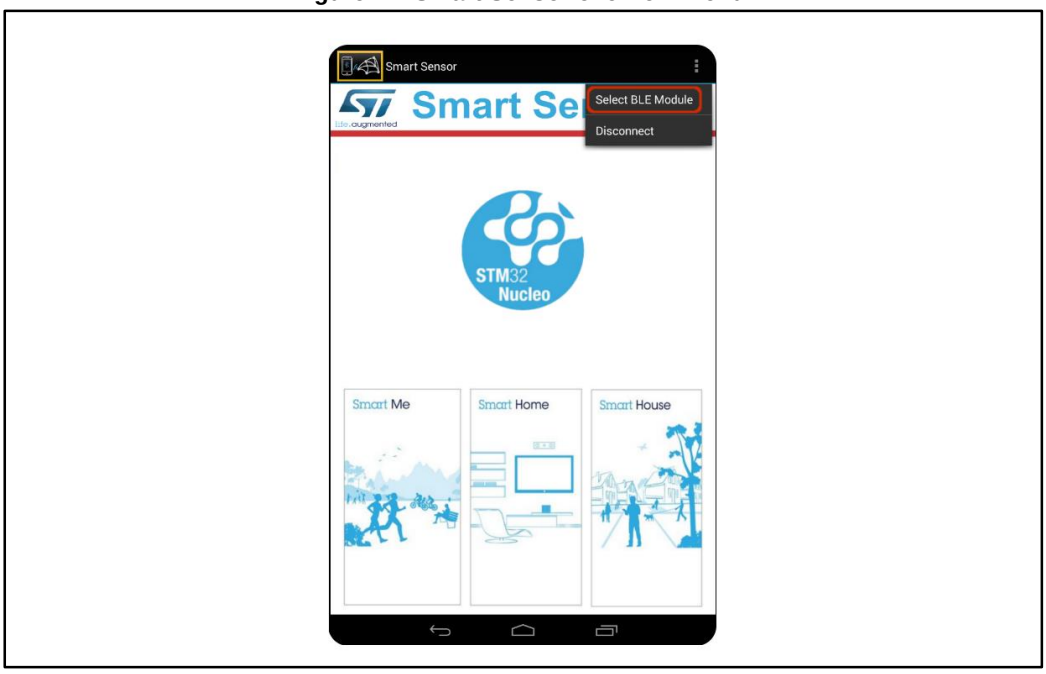


2 Launch the app to open the start screen.

Figure 13: Smart Sensor start screen

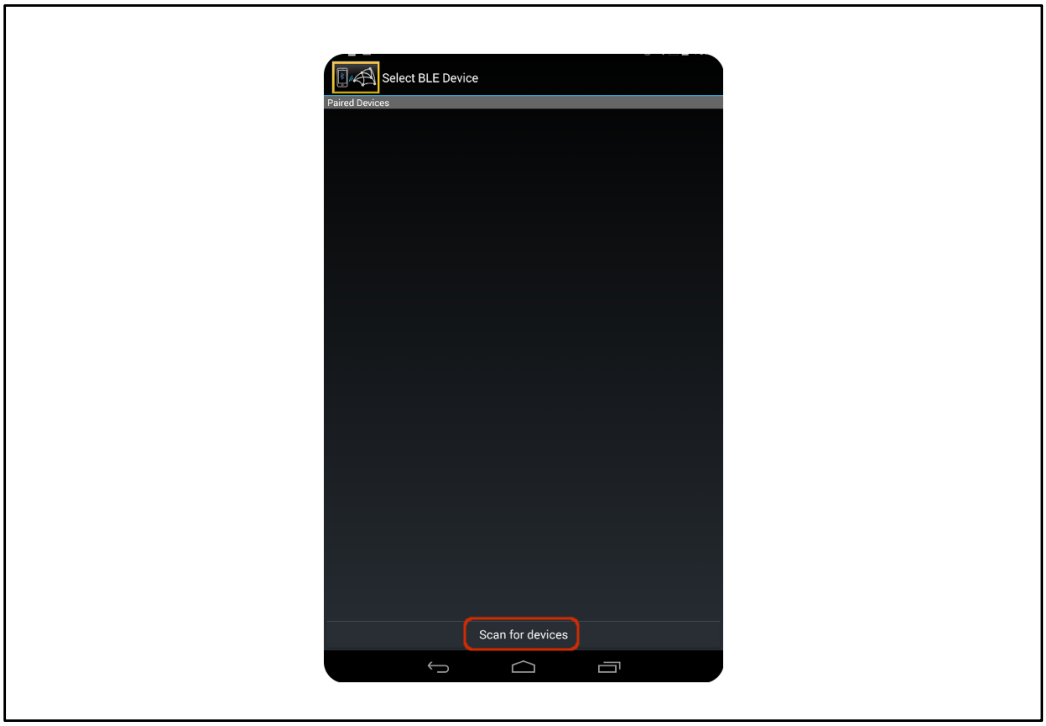
3 Click Select BLE Module on the overflow menu.

Figure 14: Smart Sensor overflow menu



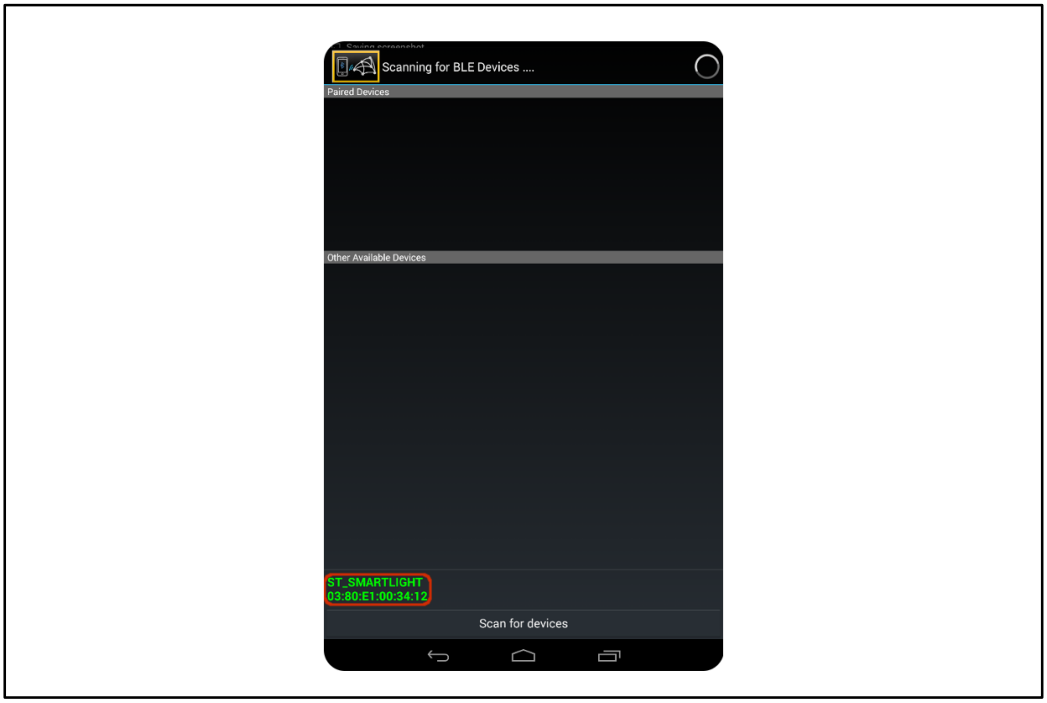
4 Click on Scan for devices in the next screen.

Figure 15: Smart Sensor device scan



5 Select ST_SMARTLIGHT to establish a connection between all nodes and smartphone.

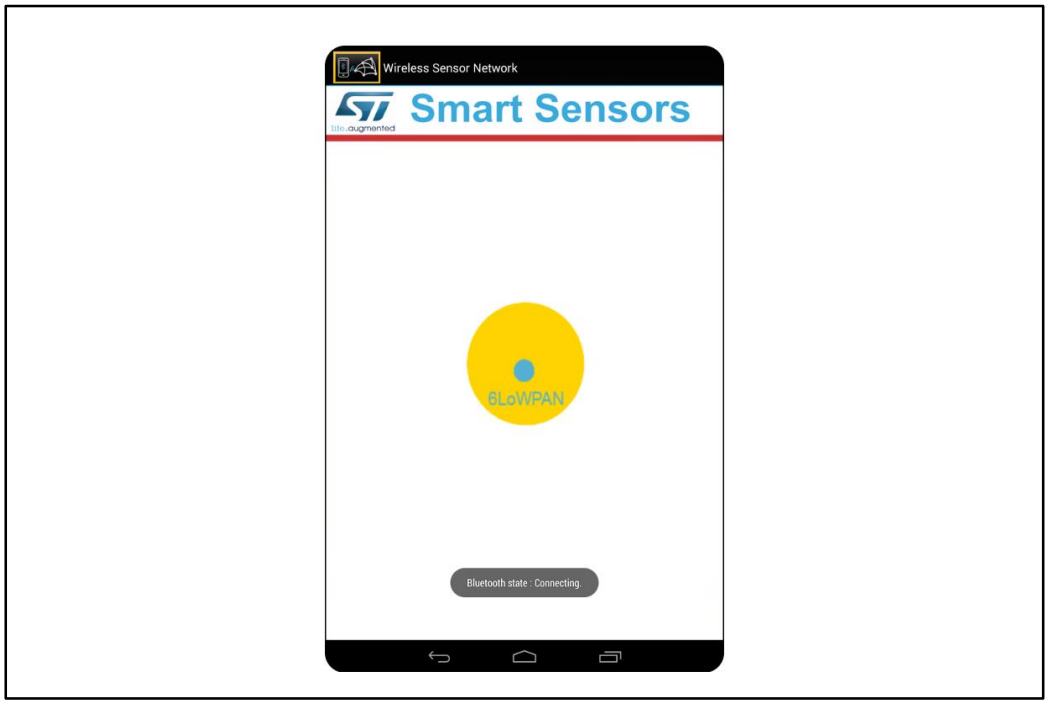
Figure 16: Smart Sensor ST_SMARTLIGHT selection



6 The first time, it will prompt you to perform pairing, click on the Pair button. When the Bluetooth module on the Smart Sensor board connects with the smartphone Bluetooth interface, LED LD2 on the STM32 Nucleo board will toggle.

On successful connection, Smart Sensor opens a new screen and the Connected popup message appears.

Figure 17: Connection between device and smartphone established

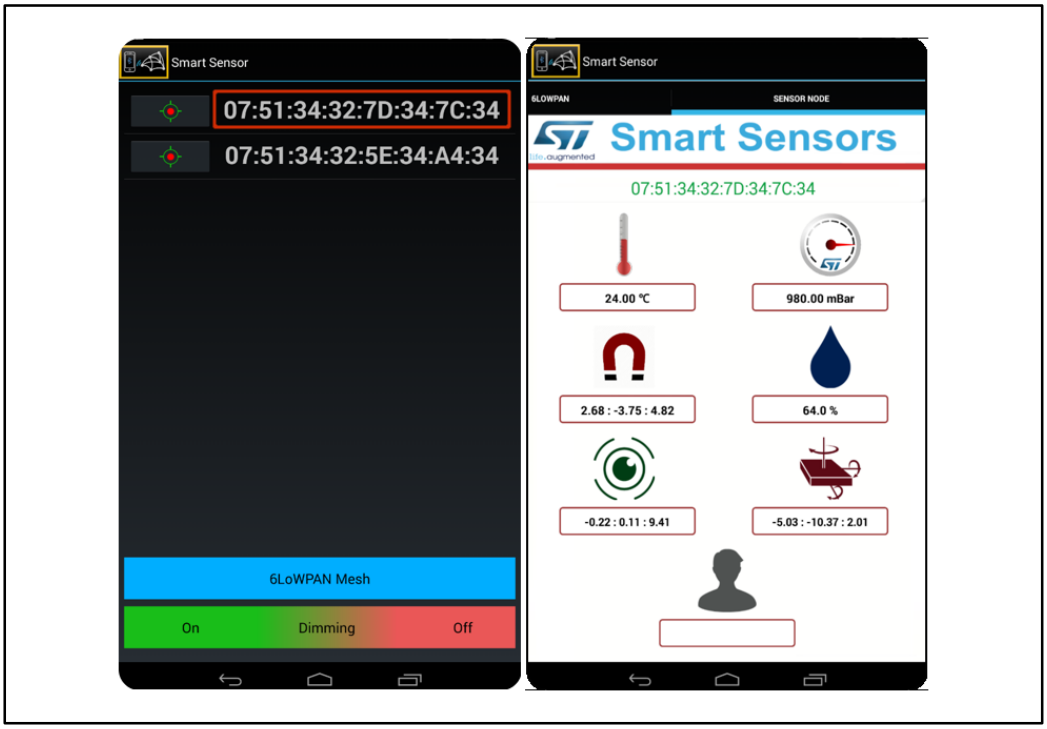


7 Select 6LoWPAN nodes to obtain a list of the nodes connected to the Smart Sensor.

Figure 18: Smart Sensor discovered sensor nodes

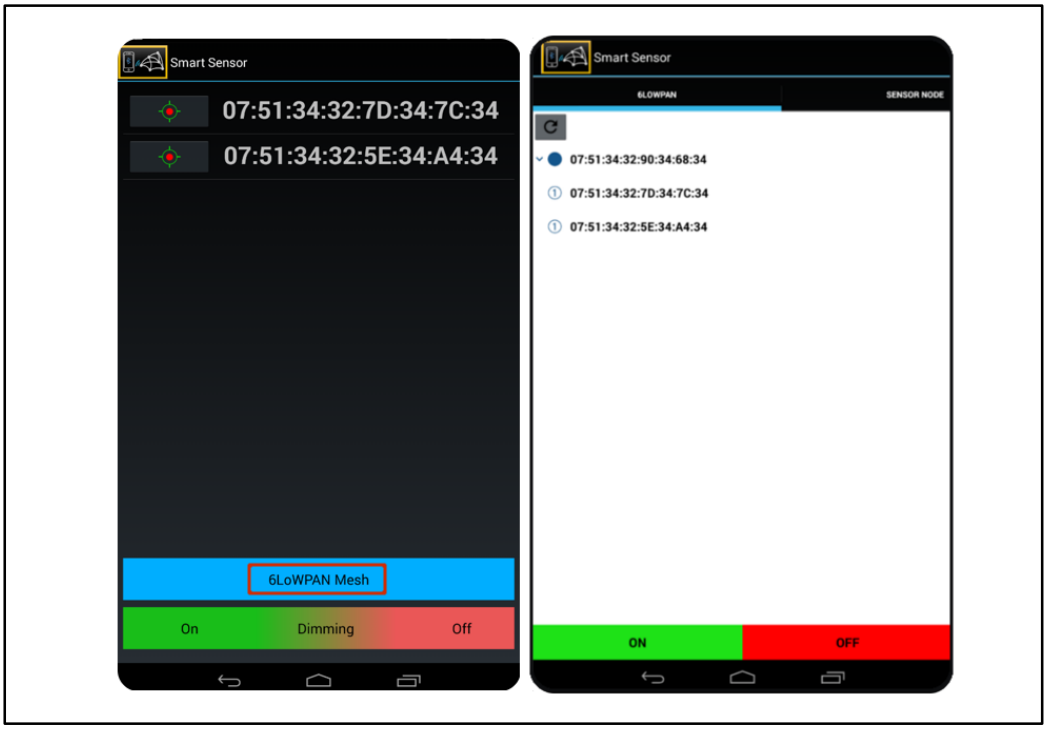
8 Select the IP address of the nodes to see their sensor values.

Figure 19: Node sensor data



9 Select 6LoWPAN Mesh option to show the mesh network tree of the connected nodes.

Figure 20: Mesh network tree



4 Revision history

Table 2: Document revision history

Date	Version	Changes
01-Sep-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.