Getting started with the STM32 ODE function pack for IoT node with BLE connectivity and environmental and motion sensors

Introduction

FP-SNS-MOTENV1 is an STM32Cube function pack, which lets you connect your IoT node to a smartphone via BLE and uses a suitable Android™ or iOS™ application, such as the STBLESensor app, to view real-time motion and environmental (such as temperature and relative humidity) sensor data.

This package also enables advanced functions such as the sensor data fusion and accelerometer-based real-time activity recognition, carry position, gesture recognition, motion intensity recognition, and real-time information about the number of steps and cadence which the user just performed with the device, that is, a cell phone.

Together with the suggested combination of STM32 and ST devices, it can be used to develop specific wearable and environmental applications, or smart things applications in general.

The software runs on the STM32 microcontroller and includes all the necessary drivers to recognize the devices on the STM32 Nucleo development board and expansion boards.

The software is available also on GitHub, where the users can signal bugs and propose new ideas through [Issues] and [Pull Requests] tabs.

1 FP-SNS-MOTENV1 software description

1.1 Overview

The key features of the FP-SNS-MOTENV1 package are:

- Complete firmware to develop an IoT node with BLE connectivity, environmental and motion sensors
- Middleware libraries for sensor data fusion and accelerometer-based real-time activity recognition, carry position, gesture recognition, motion intensity recognition and pedometer
- Compatible with STBLESensor applications for Android/iOS, to perform sensor data reading, motion algorithm features demo and firmware update (FOTA)
- Sample implementations available for the X-NUCLEO-IKS4A1 and X-NUCLEO-BNRG2A1 connected to a NUCLEO-U575ZI-Q or NUCLEO-F401RE or NUCLEO-L476RG or NUCLEO-L053R8 board
- Compatible with STM32CubeMX, can be downloaded from st.com and installed directly into STM32CubeMX
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

This software creates the following Bluetooth services:

1. The first service exposes all the hardware features with the following characteristics:
   - Temperature
   - Pressure
   - Humidity
   - 3D gyroscope, 3D magnetometer, 3D accelerometer
   - LED status
2. The second service exposes the software characteristics (excluding the NUCLEO-L053R8 board):
   - quaternions generated by the MotionFX library in short precision
   - magnetic North direction (e-Compass)
   - recognized activity using the MotionAR algorithm
   - recognized carry position using the MotionCP algorithm
   - recognized gesture using the MotionGR algorithm
   - Number of steps and frequency using the MotionPM algorithm
   - recognized motion intensity using the MotionID
3. The third service exposes the console service with:
   - stdin/stdout for bidirectional communication between client and server
   - stderr for a mono-directional channel from the STM32 Nucleo board to an Android/iOS device
4. The last service is for transmitting/resetting the calibration status (excluding the NUCLEO-L053R8 board), for switching the LED on/off and enabling the following expansion hardware features for STM32 Nucleo boards for LSM6DSV16X on to X-NUCLEO-IKS4A1 expansion board:
   - Pedometer
   - Free fall detection
   - Single tap detection
   - Double tap detection
   - Wake up detection
   - Tilt detection
   - 3D orientation
   - Multi Events detection (3D orientation, pedometer, single tap, double tap, free fall and tilt detection)

This software gathers:

- the temperature, humidity, pressure, and motion sensor drivers for the STTS22H, SHT40AD1B, LPS22DF, LIS2MDL and LSM6DSV16X, devices available when an X-NUCLEO-IKS4A1 expansion board is mounted on the STM32 Nucleo board

This package is compatible with the STBLESensor Android (version 5.0.0 or higher) or iOS (version 5.0.0 or higher) application available at the Google Play or iTunes stores, which can be used for displaying information sent via the Bluetooth low energy protocol and for Over-The-Air firmware updates (excluding the NUCLEO-L053R8 board).

1.2 Architecture

The proposed software is based on the STM32CubeHAL, the package extends STM32Cube by providing a board support package (BSP) for the BlueNRG-2 network processor (embedded in the BlueNRG-M2SP module), sensor expansion board and middleware components for communication with other Bluetooth low energy devices and for sensor data fusion.

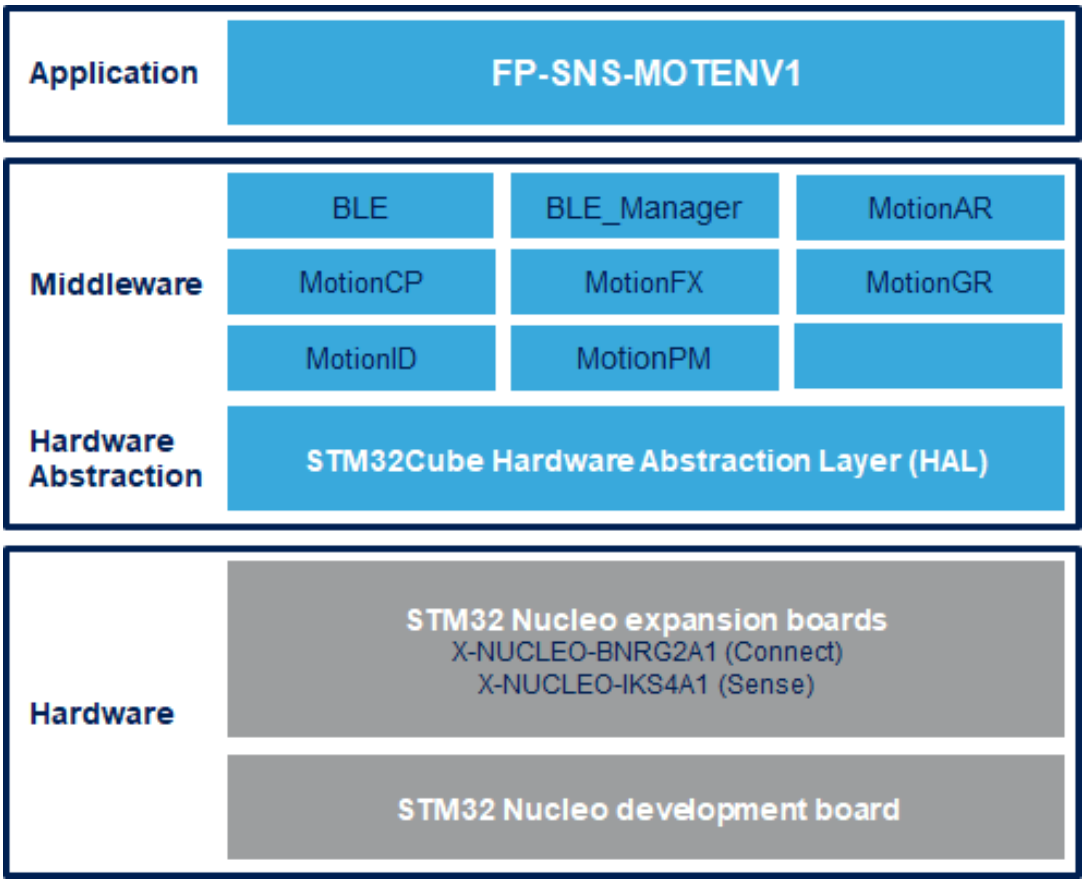
The implementation makes use of low power consumption strategies suitable for this field of application, compliant with the Bluetooth specifications core 5.2 (X-NUCLEO-BNRG2A1) for STM32 Nucleo boards.

The provided drivers abstract low-level hardware details, so middleware components and applications can access the sensors in a hardware-independent manner; the package includes a sample application to transmit the values read from all the sensors (temperature, humidity, pressure, accelerometer, magnetometer, gyroscope) to a Bluetooth low energy-enabled device such as an Android™ or iOS™-based smartphone.

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

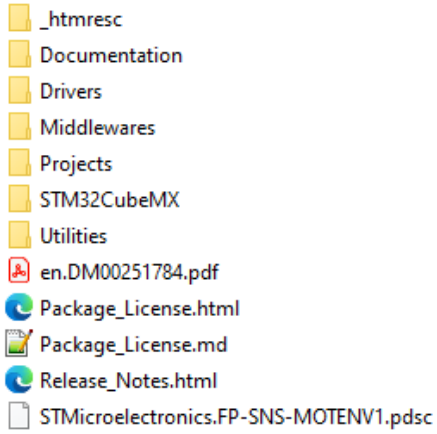
- **STM32Cube HAL layer**: The HAL driver layer provides a generic multi instance simple set of APIs (application programming interfaces) to interact with the upper layers (application, libraries and stacks). It is composed of generic and extension APIs. It is directly built around a generic architecture 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 reusability and guarantees an easy portability on other devices.
- **Board support package (BSP) layer**: The software package needs to support the peripherals on the STM32 Nucleo board apart from the MCU. This software is included in the board support package (BSP). 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.

Figure 1. FP-SNS-MOTENV1 software architecture



1.3 Folder structure

Figure 2. FP-SNS-MOTENV1 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 on-board components and the CMSIS vendor-independent hardware abstraction layer for the ARM Cortex-M processor series.
- **Middlewares**: contains libraries and protocols for BlueNRG-2 Bluetooth low energy, the Meta Data Manager, MotionFX (iNEMOEngine PRO) sensors fusion library, MotionAR activity-recognition library, MotionCP carry-position recognition library, MotionGR gesture recognition library, MotionPM real-time pedometer library, MotionID motion intensity recognition library.
- **Projects** contains:
  - a sample application used for transmitting the output of the sensor data and of the MotionFX sensor fusion and e-Compass, MotionAR activity-recognition, MotionID motion-intensity-recognition, MotionCP carry-position, MotionGR gesture recognition, and MotionPM pedometer libraries by using the Bluetooth low energy protocol provided for the NUCLEO-F401RE/NUCLEO-L476RG/NUCLEO-U575ZI-Q platforms through the IAR Embedded Workbench for ARM, RealView microcontroller Development Kit (MDK-ARM), and Integrated Development Environment for STM32 (STM32CubeIDE);
  - a sample application used for transmitting the output of the sensor data by using the Bluetooth low energy protocol provided for the NUCLEO-L053R8 platforms through the IAR Embedded Workbench for ARM, RealView microcontroller Development Kit (MDK-ARM-STR) and Integrated Development Environment for STM32 (STM32CubeIDE).
- **Utilities**: contains the bootloader binary ready to be flashed for STM32F401RE and STM32L476RG.

1.4 Flash management

For NUCLEO-F401RE, NUCLEO-L476RG and NUCLEO-U575ZI-Q the FP-SNS-MOTENV1 uses the Flash memory to:

1. save firmware information (node name, firmware ID, etc.) and the magnetometer calibration values in the dedicate flash address
2. allow the Firmware-Over-The-Air update

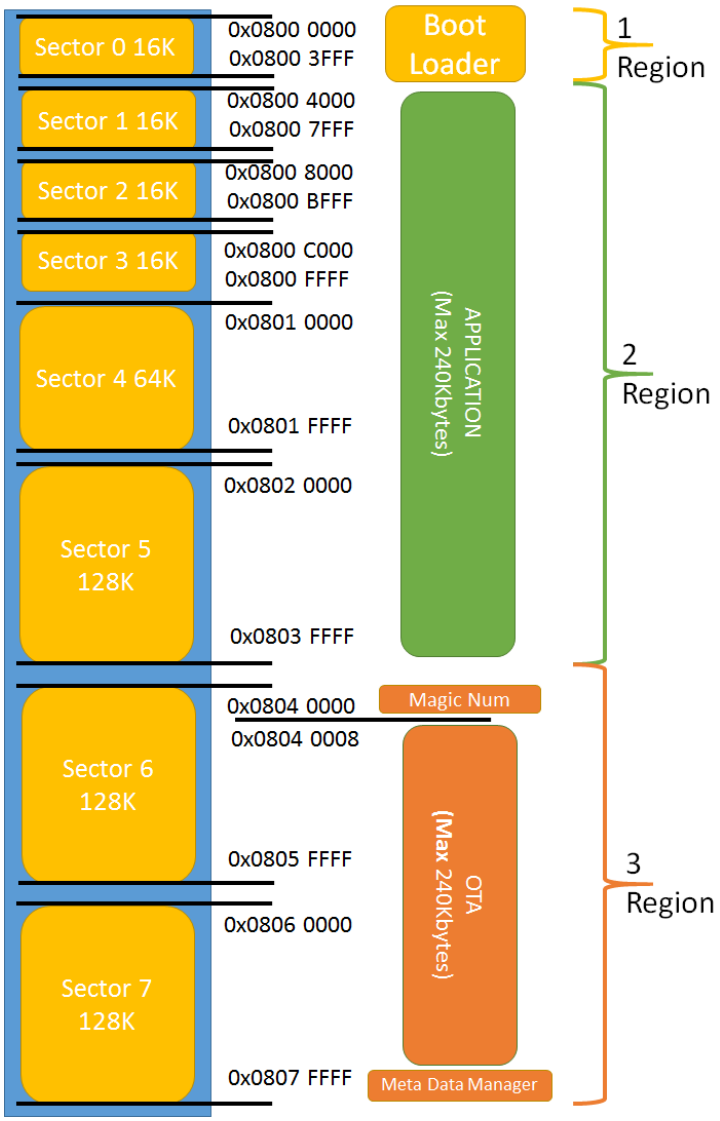
For NUCLEO-F401RE and NUCLEO-L476RG, to enable these features, the whole Flash is divided into the following distinct regions:

1. contains a custom boot loader
2. contains the FP-SNS-MOTENV1 firmware
3. used to store FOTA before the update

The same Flash management applies to both the STM32F401RE and the STM32L476RG boards, even if they have different cache sizes (respectively 512 and 1024 Kbytes), and two different configurations. For further info on Flash configuration please refer to:

- *RM0368* Reference manual STM32F401xB/C and STM32F401xD/E advanced ARM®-based 32-bit MCUs
- *RM0351* Reference manual STM32L4x6 advanced ARM®-based 32-bit MCUs

Figure 3. FP-SNS-MOTENV1 Flash management on STM32F401RE



For NUCLEO-U575ZI-Q the dual bank flash features to allow Firmware-Over-the-Air update of a running program is used.

For further info on Flash configuration please refer to:

- RM0456 STM32U5 Series Arm®-based 32-bit MCUs

1.5 The Boot process

1.5.1 NUCLEO-F401RE and NUCLEO-L476RG Boot Process

The FP-SNS-MOTENV1 cannot be flashed to the beginning of the Flash (address 0x08000000), and is therefore compiled to run from the beginning of the second Flash region (address 0x08004000).

To enable this behavior, we set the vector table offset with respect to the default value by modifying the Src/system_stm32f4xx.c (for STM32F401) and the Src/system_stm32l4xx.c (for STM32L476) files, thus: `#define VECT_TAB_OFFSET 0x4000`.

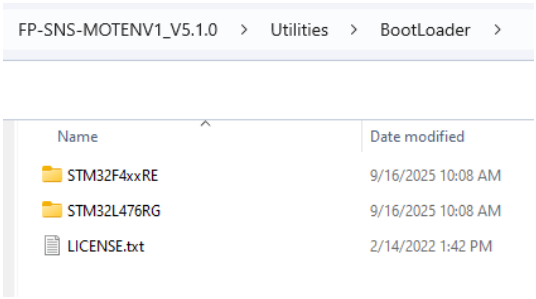
We also changed the linker script. For example, the linker script used for FP-SNS-MOTENV1 running on STM32F401RE and compiled using IAR Embedded Workbench for ARM is:

```
define symbol __ICFEDIT_intvec_start__ = 0x08004000;
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__ = 0x08004000;
define symbol __ICFEDIT_region_ROM_end__ = 0x0803FFFF;
define symbol __ICFEDIT_region_RAM_start__ = 0x20000000;
define symbol __ICFEDIT_region_RAM_end__ = 0x20017FFF;
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x8000;
define symbol __ICFEDIT_size_heap__ = 0x800;
```

Using the above linker script, the maximum usable code size is fixed at 240 Kbytes.

To use FP-SNS-MOTENV1, flash the appropriate bootloader binary for STM32F401RE or STM32L476RG in the Utilities\BootLoader folder to the first Flash region (address 0x08000000).

Figure 4. Bootloader utility content



On any board reset, the board starts executing the boot loader.

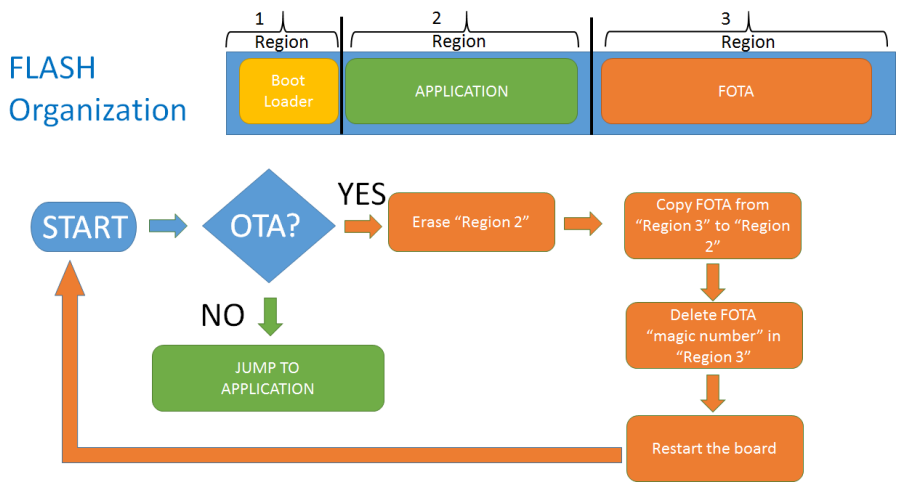
The boot loader checks whether a FOTA is available: the check is based on the presence of a "magic number" at the beginning of the third Flash region.

If there is a FOTA available, the bootloader:

1. erases the second Flash region (containing the FP-SNS-MOTENV1 firmware)
2. replaces its content with the FOTA
3. erases the “magic number” used to check FOTA presence
4. restarts the board

If there is no FOTA available, the boot loader jumps directly to the FP-SNS-MOTENV1 firmware.

Figure 5. FP-SNS-MOTENV1 boot process

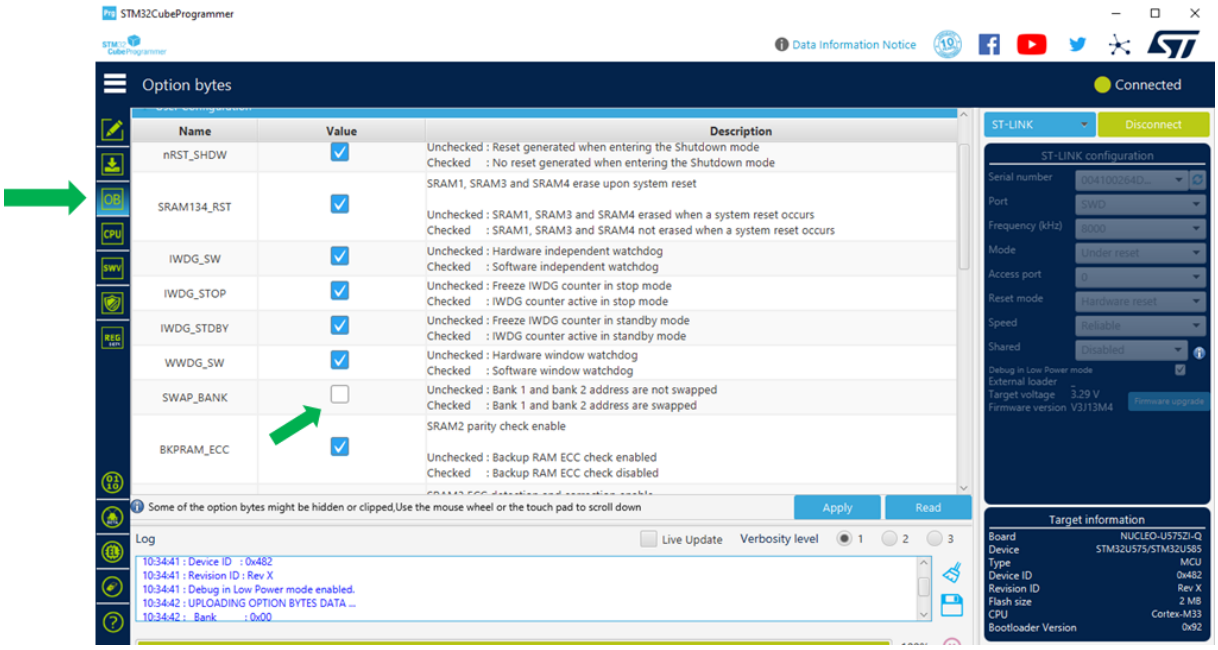


1.5.2 NUCLEO-U575ZI-Q boot process

The MOTENV1 firmware uses the dual bank flash features to allow firmware over-the-air update of a running program without using the bootloader.

For the first installation, after the full flash erases (suggest procedure), use the STM32CubeProgrammer to set STM32 MCU user byte settings to use the bank 1 for flash the firmware and starts the application, as shown in the figure below:

Figure 6. STM32CubeProgrammer option bytes



After the boot (using the reset button on the board) the MOTENV1 firmware receives the new firmware from the STBLESensor application, saves it on one flash bank (either bank1 or bank2) and performs a reboot executing the new code saved on the other flash bank.

A program related to a specific region can run in that region only.

The MOTENV1 application, however, can swap among different flash banks and each program can run in any flash memory bank.

As the application does not erase the previous version of the code after the update, it allows the rollback to the previous program in case of a hardware fault.

Note: The FOTA update procedure and meta data in flash memory are not available for NUCLEO-L053R8.

1.6 The installation process for NUCLEO-F401RE or NUCLEO-L476RG board

The package contains in the Binary directory an image that includes:

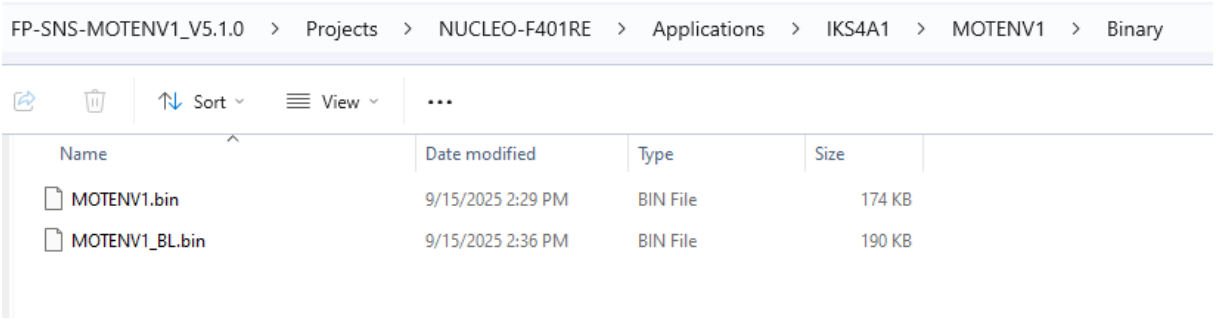
- precompiled application firmware that may be flashed to the correct memory address (0x08004000) by using STM32CubeProgrammer

Note: This precompiled binary is compatible with the FOTA update procedure

- precompiled application plus bootloader firmware that may be flashed to a NUCLEO-F401RE or NUCLEO-L476RG board, by using STM32CubeProgrammer or drag and drop

Note: This precompiled binary is not compatible with the FOTA update procedure

Figure 7. Binary folder content for NUCLEO-F401RE

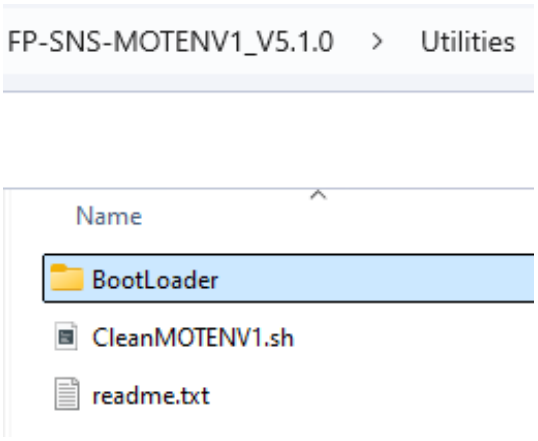


In the folder Utilities there is a scripts *.sh that makes the following operations:

- full Flash Erase
- load the BootLoader on the rigth flash region
- load the Program (after the compilation) on the rigth flash region (This could be used for a FOTA)
- dump back one single binary that contain BootLoader+Program that could be flashed at the flash beginning (address 0x08000000) (This COULD BE NOT used for FOTA)
- reset the board

Before to execute the *.sh script, it is necessary to edit it to set the installation path for STM32CubeProgrammer. BootLoaderPath/<BootLoader file name> and BinaryPath as input are required when execute *.sh script.

Figure 8. Utilities folder content



This script:

- performs a full Flash erase to start from a clean system
- flashes the bootloader to the right position 0x08000000
- flashes the FP-SNS-MOTENV1 firmware on the right position 0x08004000

Figure 9. **Bootloader and ALLMEMS1 installation**

1.7 Firmware-over-the-air (FOTA) update

The FP-SNS-MOTENV1 firmware can be updated in FOTA through the Bluetooth low energy protocol, communicating with an Android/iOS device, via the STBLESensor application (version 5.0.0 and above) available on their respective stores.

The FOTA is available for NUCLEO-U575ZI-Q, NUCLEO-F401RE and NUCLEO-L476RG, but not for NUCLEO-L053R8 as the latter has 64 Kbyte Flash memory, which is not sufficient to store the Bootloader, the firmware and the FOTA.

To update the firmware, the STBLESensor application sends the update size (bytes) and its associated CRC (cyclic redundancy check) value to the FP-SNS-MOTENV1. Once the update has been received, the FP-SNS-MOTENV1 uses the hardware CRC calculation unit included in the processor to check update integrity.

For NUCLEO-F401RE and NUCLEO-L476RG, if the CRC computed matched the CRC expected, the FP-SNS-MOTENV1 writes the “magic number”at the beginning of the third Flash region, just before the saved FOTA, to signal the boot loader a Firmware update has been received and checked, and is ready to update the FP-SNS-MOTENV1.

1.8 APIs

Detailed technical information about the APIs available to the user can be found in a compiled HTML file located inside the “Documentation” folder of the software package where all the functions and parameters are fully described.

1.9 Sample application description

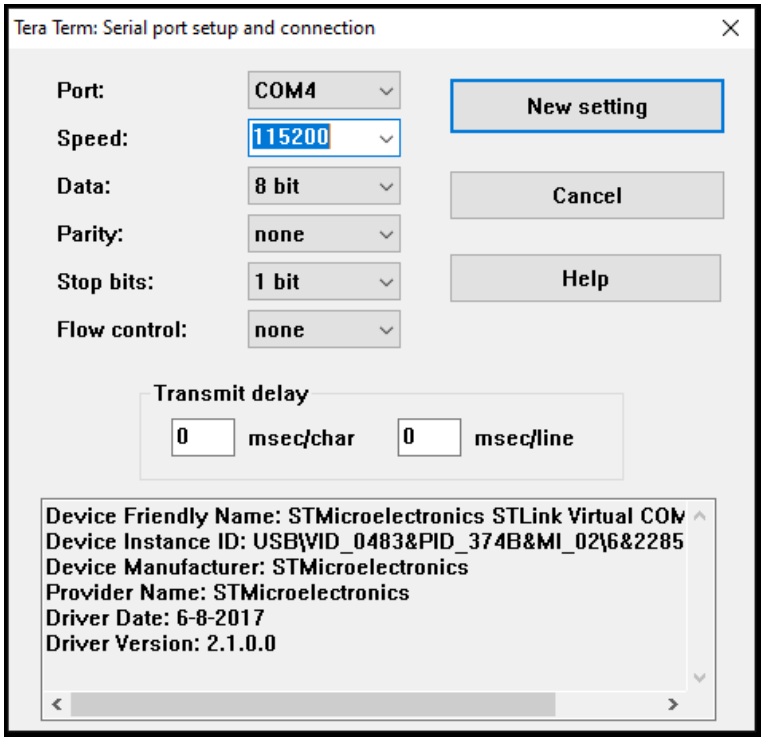
A sample application is provided in the Projects folder for:

- the X-NUCLEO-IKS4A1 and X-NUCLEO-BNRG2A1 expansion boards with the NUCLEO-U575ZI-Q or NUCLEO-F401RE or NUCLEO-L476RG or NUCLEO-L053R8

Ready to build projects are available for multiple IDEs.

With the NUCLEO-U575ZI-Q, NUCLEO-F401RE, and NUCLEO-L476RG boards, you can set up a terminal window for the appropriate UART communication port (use the baud, data, parity and stop settings below) to control the initialization phase.

Figure 10. **Terminal setting**



When you first press the reset button on the NUCLEO-U575ZI-Q or NUCLEO-F401RE or NUCLEO-L476RG board, the application:

- starts initializing the UART
- determines which MEMS expansion board is connected to STM32 Nucleo board
- checks whether all the sensors are present and working
- determines which BlueNRG expansion board is connected to the STM32 Nucleo board (X-NUCLEO-BNRG2A1) and hardware and firmware version information
- shows a random BLE MAC address
- initializes the BLE feature service (temperature, humidity, pressure, 3D gyroscope, 3D magnetometer, 3D accelerometer, LED characteristics and MotionFX, MotionAR, MotionCP, MotionGR, MotionPM, and MotionID libraries)
- initializes the BLE console service adding the stdin/stdout and stderr characteristics
- initializes the BLE config service transmitting/resetting the calibration status and enabling the hardware features

It can generate an interrupt signaling a free fall, tilt, wake up, single tap, double tap, 6D position or pedometer event, which is transmitted over Bluetooth to the attached Android™/iOS™ device.

Figure 11. **Initialization phase**

As shown in the console output above, the application sends:

- 3 short precision quaternions every 30 ms
- Temperature, humidity and pressure data every 500 ms
- 3D accelerometer, gyroscope and magnetometer data every 50 ms

This application reads the accelerometer, magnetometer and gyroscope values at 100 samples/second. The MotionFX library combines these sensor values to produce and transmit 100 quaternions/second to the client connected via Bluetooth low energy to reflect real motion using a vendor-specific BLE service.

These definitions in MOTENV1_config.h control the quantity of quaternions the application sends to the Bluetooth client:

- `QUAT_UPDATE_MUL_10MS`: defines the transmission rate for each set of quaternions by multiple of 10 ms.
- `SEND_N_QUATERNIONS`: defines the quantity of quaternions sent to each Bluetooth package.

By default, the application sends three quaternions every 30 ms.

The same MOTENV1_config.h file also defines:

- `MOTENV1_DEBUG_CONNECTION` and `MOTENV1_DEBUG_NOTIFY_TRAMISSION` to enable some debugging information for BLE communication

The MotionFX library includes an e-Compass auto-calibrating procedure whose status is transmitted via BLE to the client:

- on the NUCLEO-U575ZI-Q or NUCLEO-F401RE or NUCLEO-L476RG boards, you can press the user button to reset the library calibration status and force a new auto-calibration procedure.

The MotionAR library can recognize the following activities:

- stationary
- walking
- fast walking
- jogging
- biking
- driving

The MotionCP library recognizes and provides real-time information about the way the user is carrying the board, which equates to the phone carry position:

- on desk
- in hand
- near head
- shirt pocket
- trouser pocket
- arm swing

The MotionGR library can recognize gestures like:

- pick up
- glance
- wake up in hand

The MotionPM library counts the number of steps and computes their frequency.

The MotionID library can recognize the following activities:

- on desk
- bed, couch
- light movement
- biking
- typing/writing
- slow walking
- walking
- fast walking
- jogging
- fast jogging
- sprinting

When an Android/iOS device is connected to the NUCLEO-U575ZI-Q or NUCLEO-F401RE or NUCLEO-L476RG board, it is possible to control data transmitted by the board (see the following figure).

Note: Due to constraints flash size, all the libraries described above are not used for NUCLEO-L053R8 board.

Figure 12. UART console output when one device is connected to the board

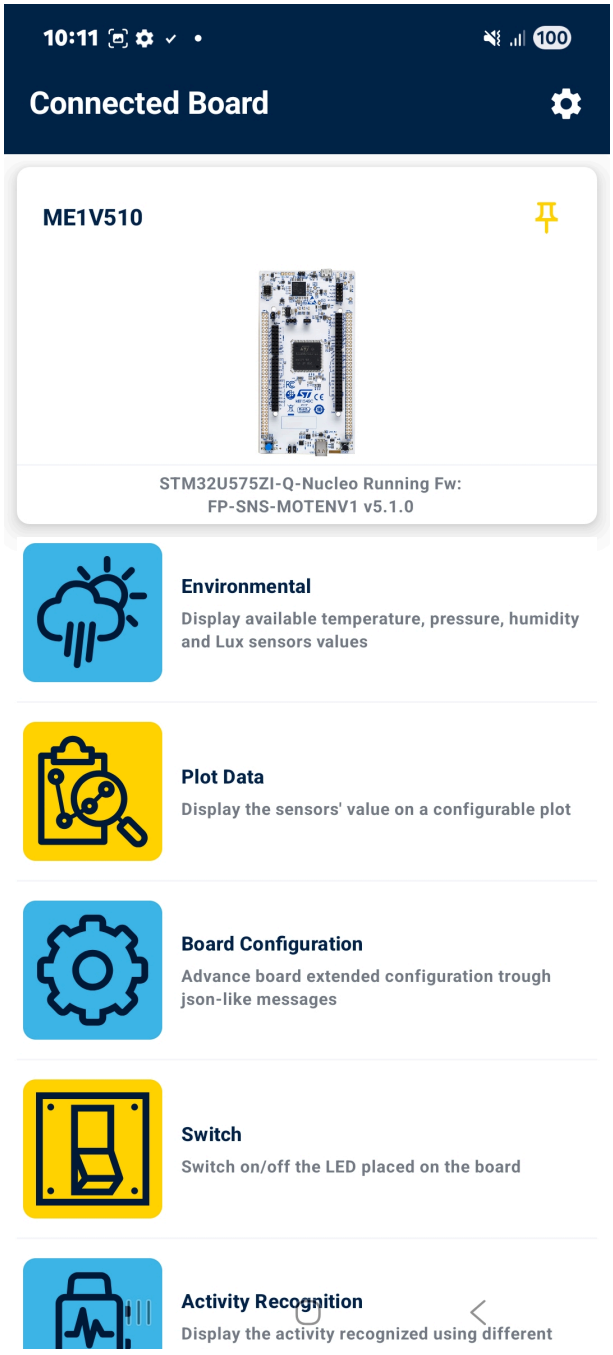
1.10 Android and iOS STBLESensor client application

The FP-SNS-MOTENV1 software for STM32Cube is compatible with the STBLESensor Android (version 5.0.0 or higher) or iOS (version 5.0.0 or higher) application available at Google Play or iOS stores.

We use the Android application in this example.

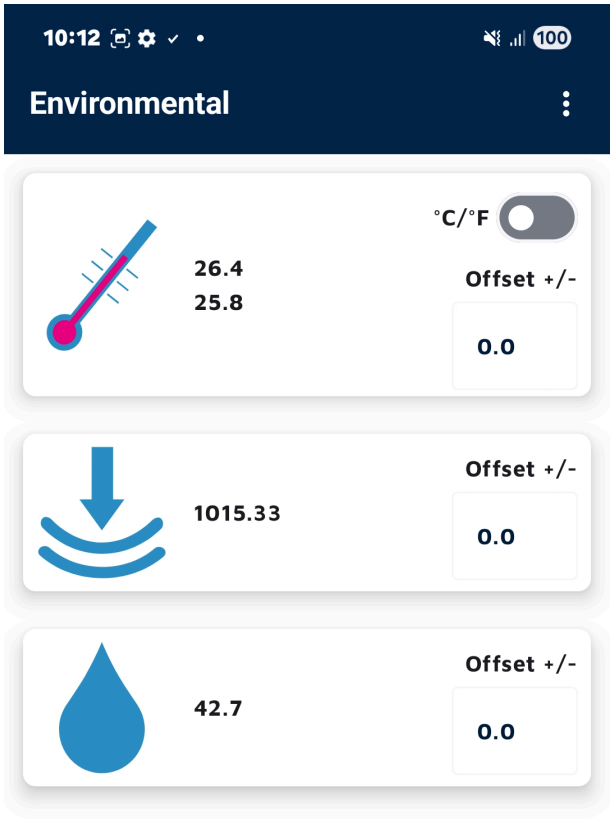
Following connection, STBLESensor starts with the main page shown below, where there are a list of the available features.

Figure 13. **ST BLE Sensor Connected Board**



The following page shows the value for the temperature, pressure and Humidity

Figure 14. **STBLESensor (Android version) main page following BLE connection**



For the MotionFX sensor fusion library, the following page shows a cube that rotates with board movement.

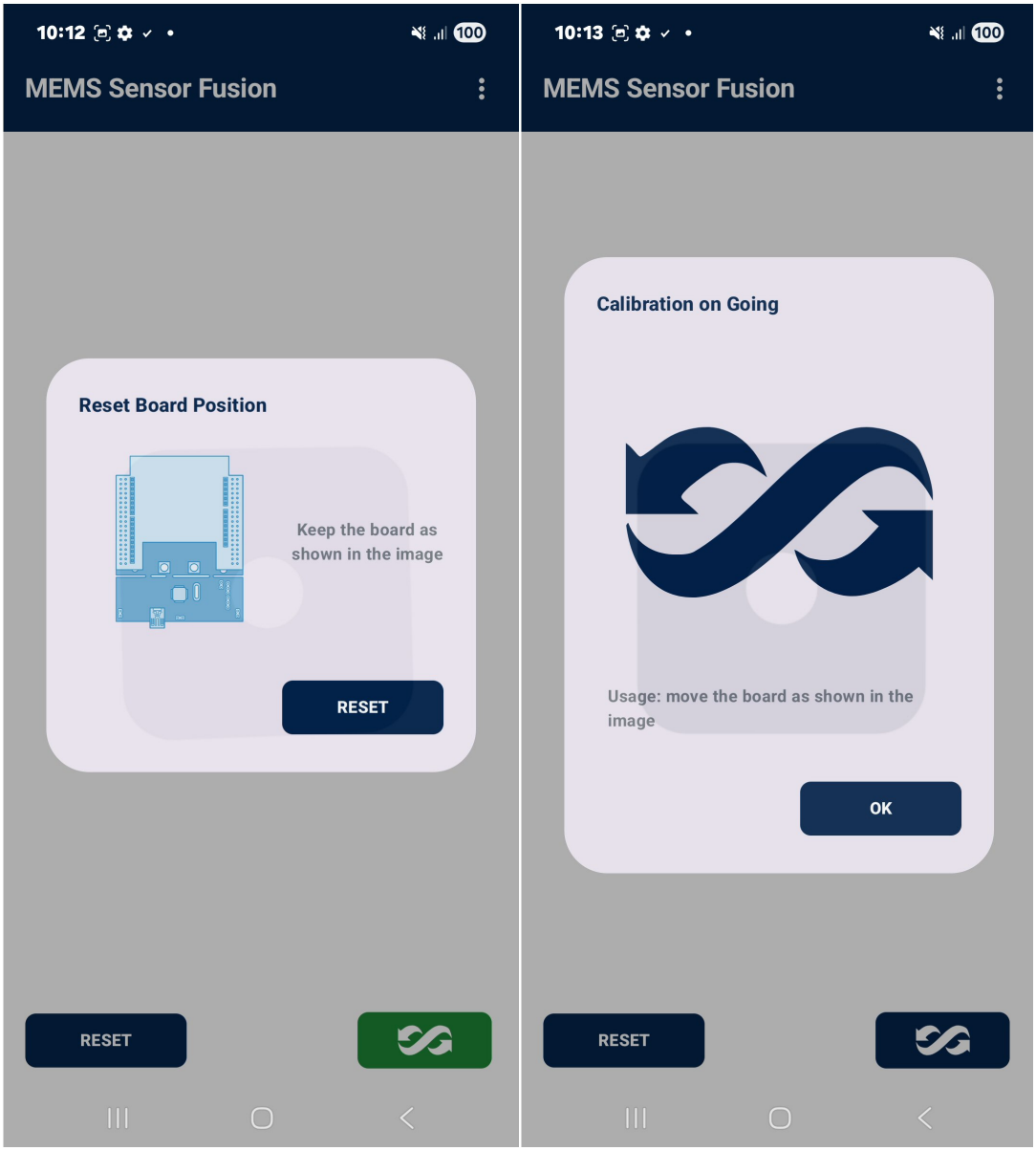
Figure 15. **STBLESensor (Android version) MotionFX sensor fusion page**

On this page, there are two buttons along the bottom:

- the left is for resetting the cube position.
- the right shows the calibration status of the MotionFX library (black for not calibrated, green for calibrated). Clicking it forces a magneto calibration.

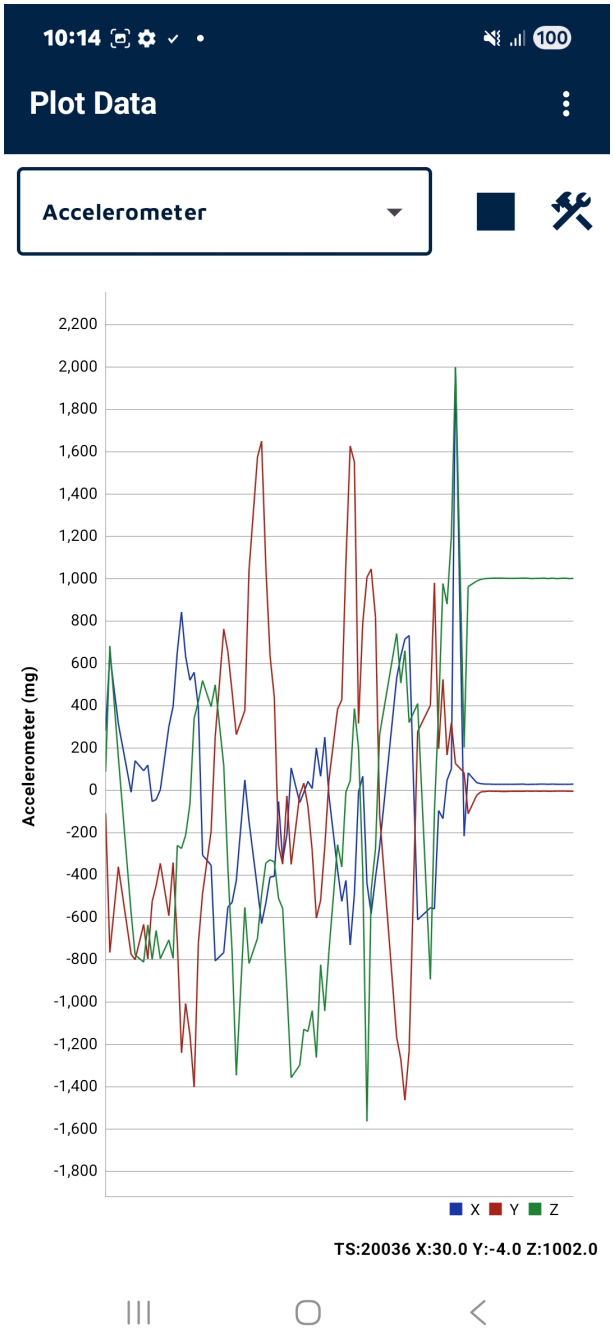
When either button is pressed, the application pops up a window describing how to position the board for correct cube rotation and how to move the board to facilitate calibration (see figure below).

Figure 16. **STBLESensor (Android version) popup windows**



On the next page to the left, you can plot any value from the sensor expansion boards.

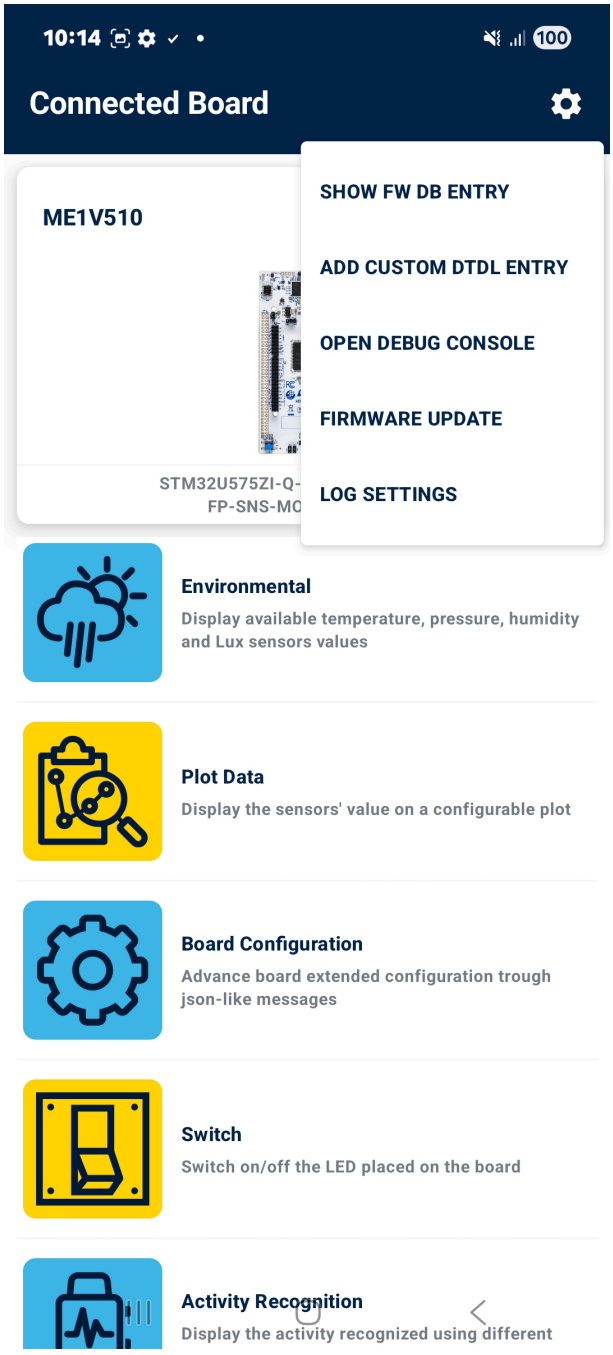
Figure 17. **STBLESensor (Android version) accelerometer plot**



In the option menu below, you can open:

- Show FW DB Entry
- Add Custom DTDL Entry
- Open Debug Console
- Firmware Update
- Log Settings

Figure 18. **STBLESensor (Android version) options selection**

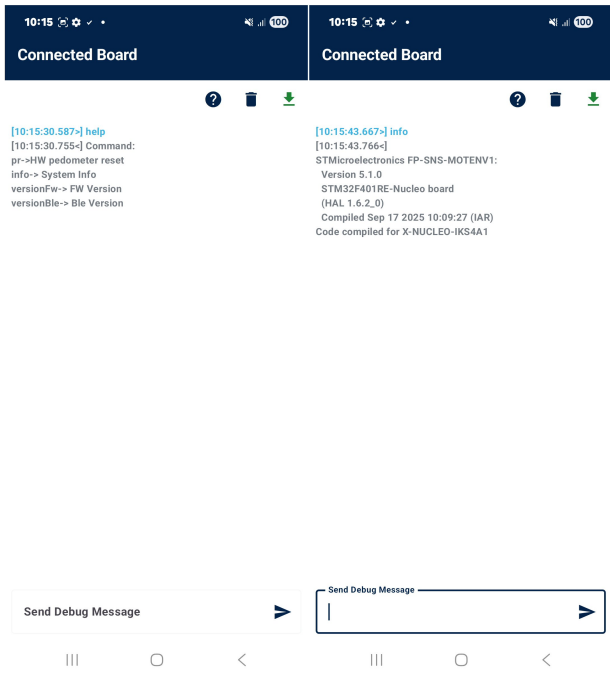


If the Debug console is selected, stdin is displayed and any message written in the Debug console triggers a reply with the same message if it is not implemented command, as shown below.

Figure 19. **STBLESensor (Android version) Debug console (stdin/stdout/stderr) no command**

With the command "Help" and "Info".

Figure 20. STBLESensor (Android version) Debug console (stdin/stdout/stderr) command Help and Info

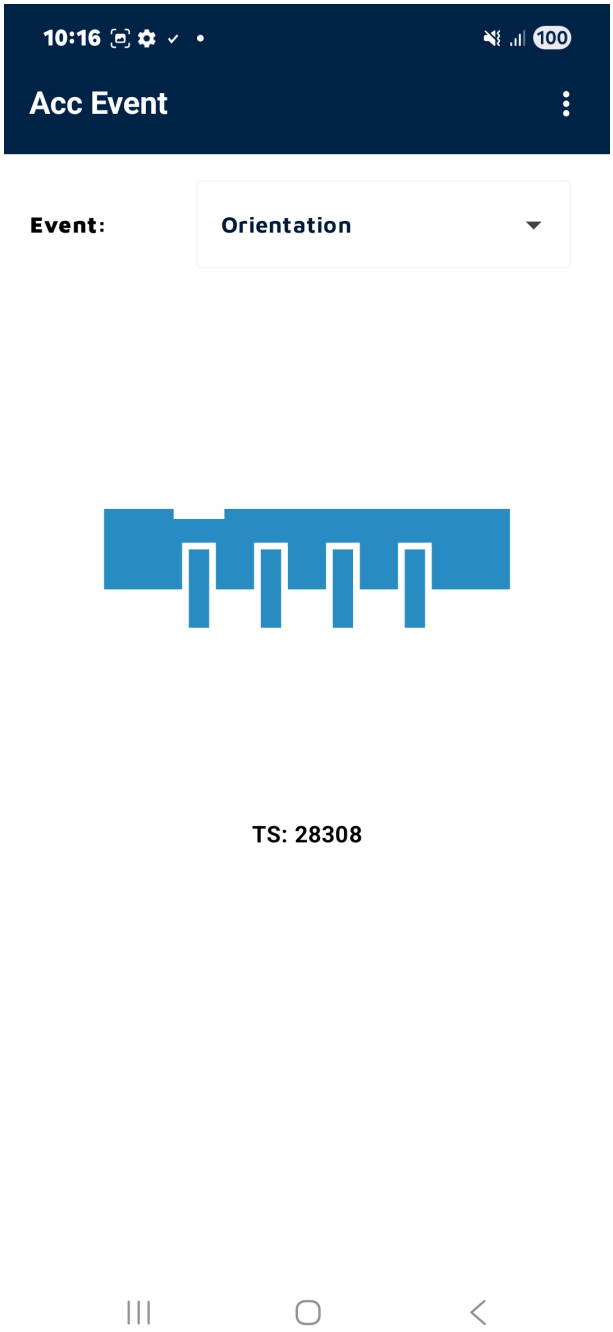


There is another page where you can choose which hardware feature to enable (one at the time) and view the events (see following figures) on the same page from:

- LSM6DSV16X on X-NUCLEO-IKS4A1 expansion board and for each nucleo board

The orientation hardware feature is the default setting.

Figure 21. STBLESensor (Android version) Orientation hardware feature



From the **Accelerometer Events** menu, a single hardware feature can be selected.

Figure 22. STBLESensor (Android version) hardware feature menu

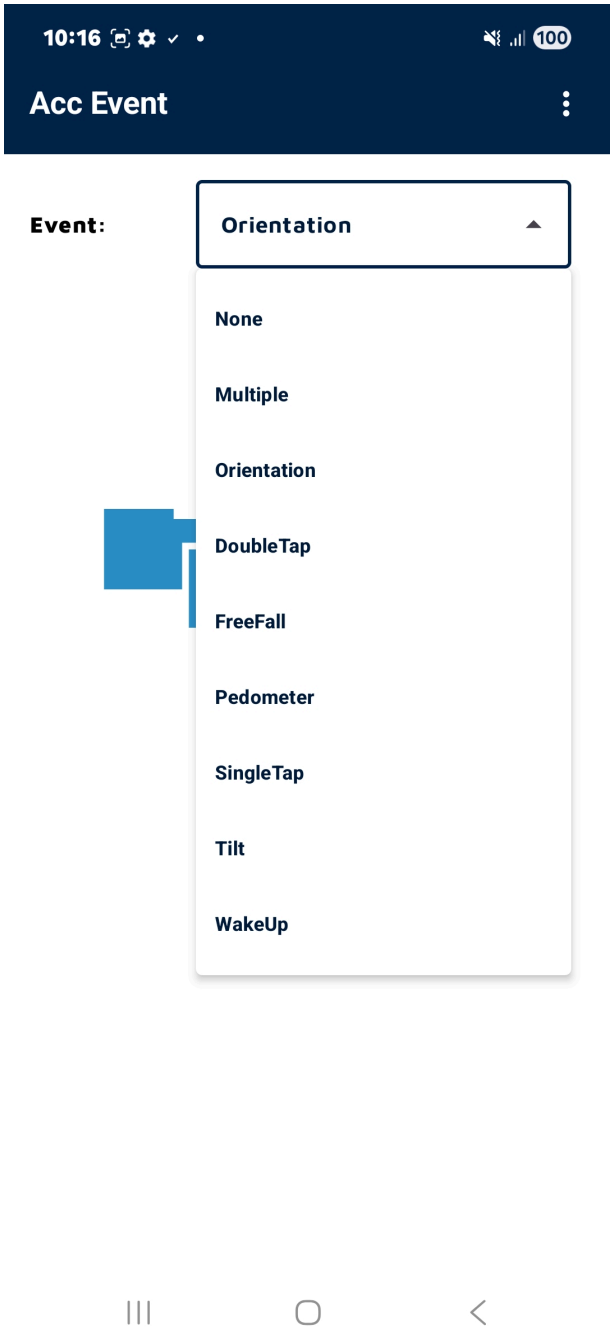


Figure 23. STBLESensor (Android version) hardware feature examples: double tap, Free fall, pedometer, single tap, tilt, wake up

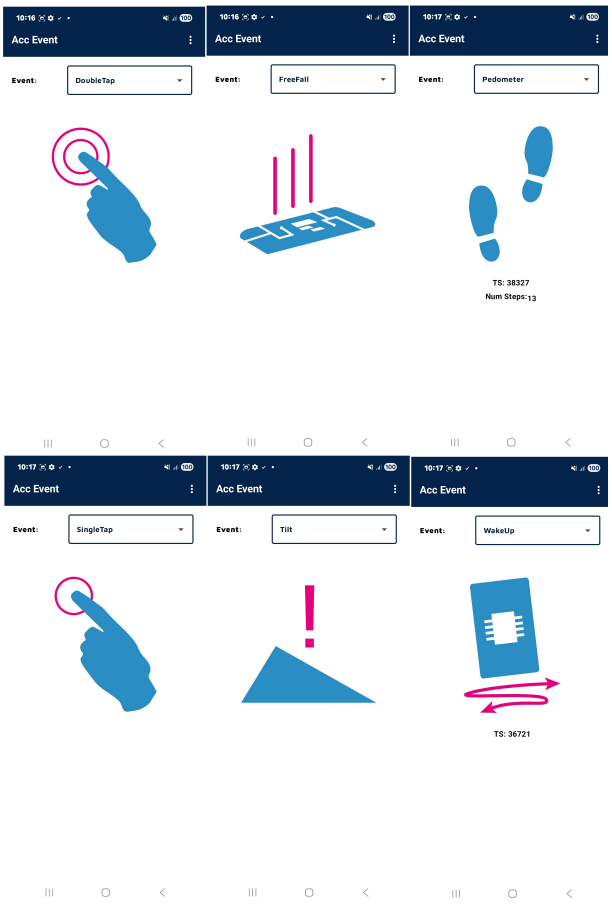
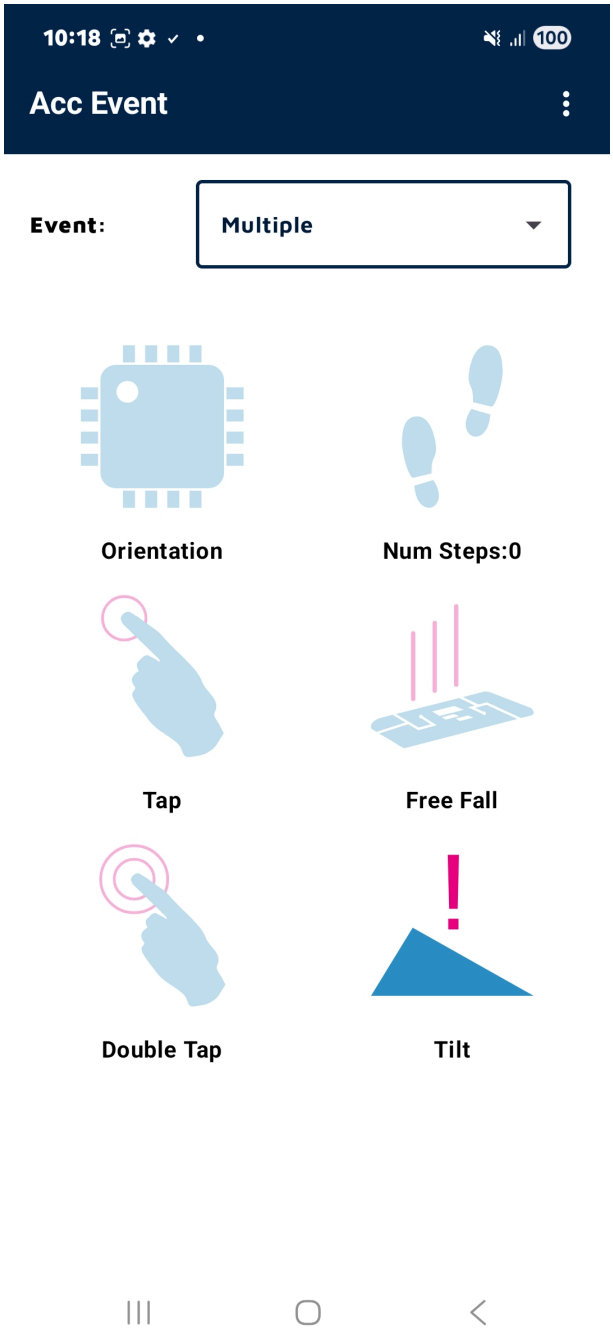


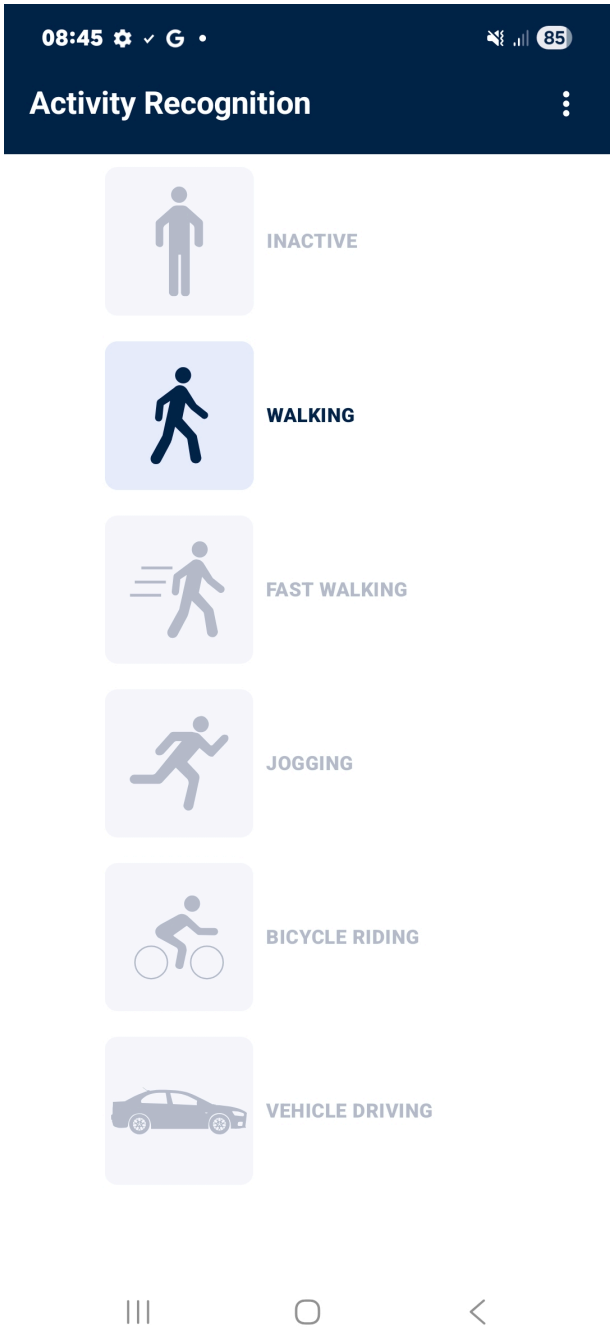
Figure 24. **STBLESensor (Android version) multiple hardware feature**



For the MotionAR algorithm, the page shown below is available, signaling one of the following recognized activities:

- Stationary
- Walking
- Fast walking
- Jogging
- Biking
- Driving

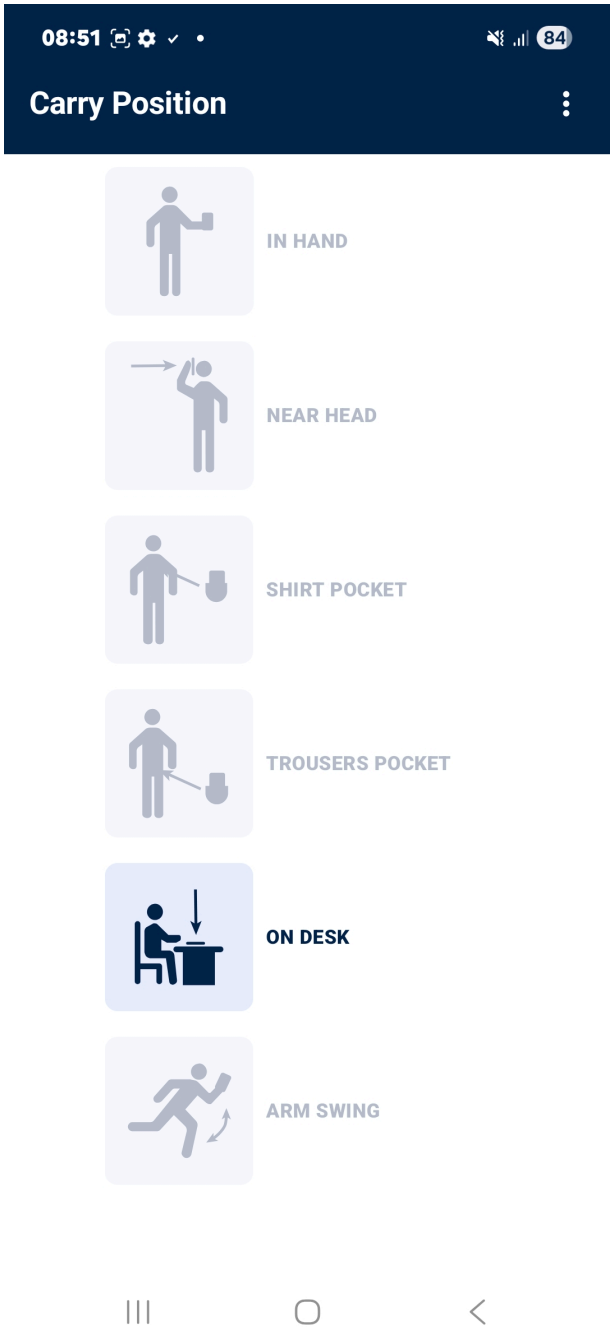
Figure 25. **STBLESensor (Android version) MotionAR activity recognition page**



For the MotionCP algorithm, the page shown below is available, with information about how the user is carrying the board, which equates to phone carry positions:

- on desk
- in hand
- near head
- shirt pocket
- trousers pocket
- arm swing

Figure 26. **STBLESensor (Android version) MotionCP carry position recognition page**



For the MotionGR algorithm, the page shown below is available with gesture recognition information like:

- pick up
- glance
- wake up in hand

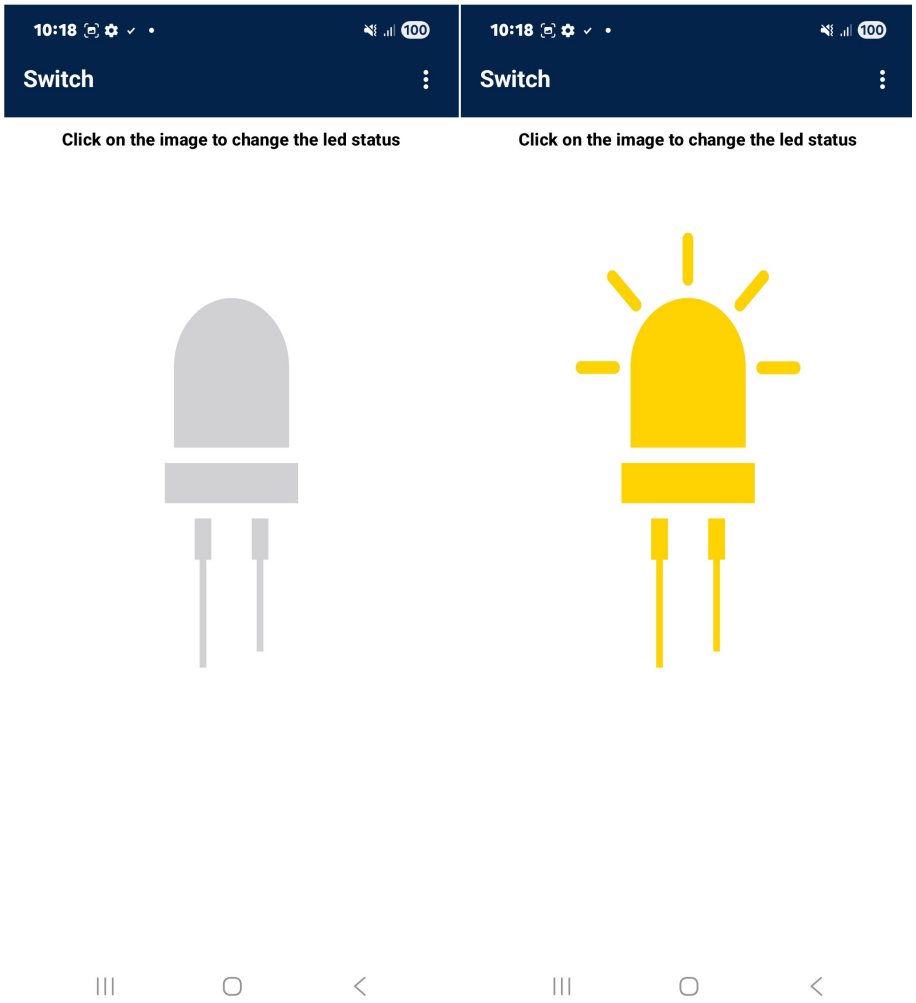
Figure 27. **STBLESensor (Android version) MotionGR gesture recognition page**

For the MotionPM algorithm, the page shown below is available with pedometer information.

Figure 28. **STBLESensor (Android version) MotionPM pedometer page**

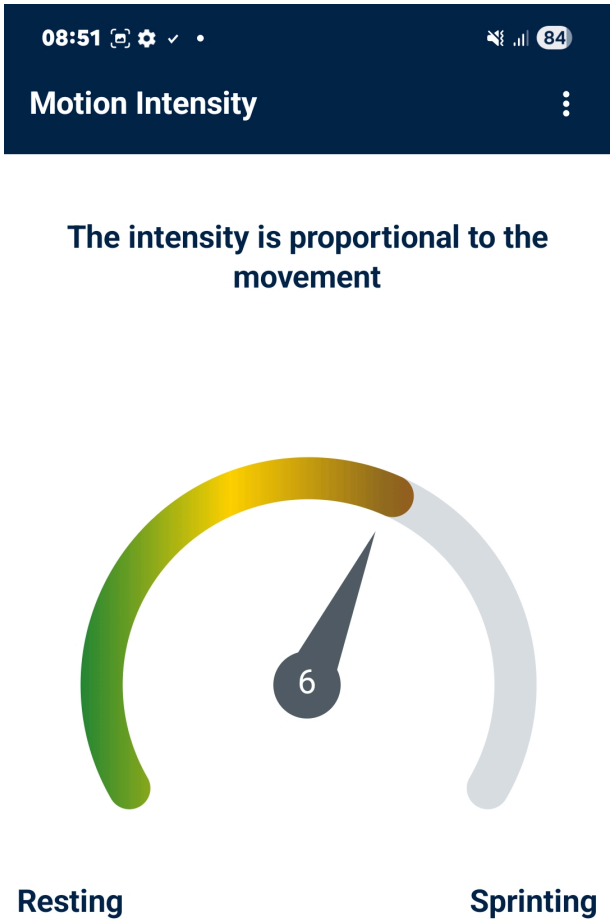
The following page shows the LED on/off control.

Figure 29. **STBLESensor (Android version) board LED control**



For the MotionID algorithm, the page shown below is available.

Figure 30. **STBLESensor (Android version) Motion Intensity page**



For the MotionFX sensor fusion library, the following page shows a e-compass that rotates with board movement.

Figure 31. **STBLESensor (Android version) e-compass page**

At the bottom, the right button shows the calibration status of the MotionFX library (black for not calibrated, green for calibrated). Clicking it forces a magneto calibration.

With the board configuration page a few firmware details can be shown. The image below shows the available commands:

Figure 32. **STBLESensor (Android version) Board Configuration**

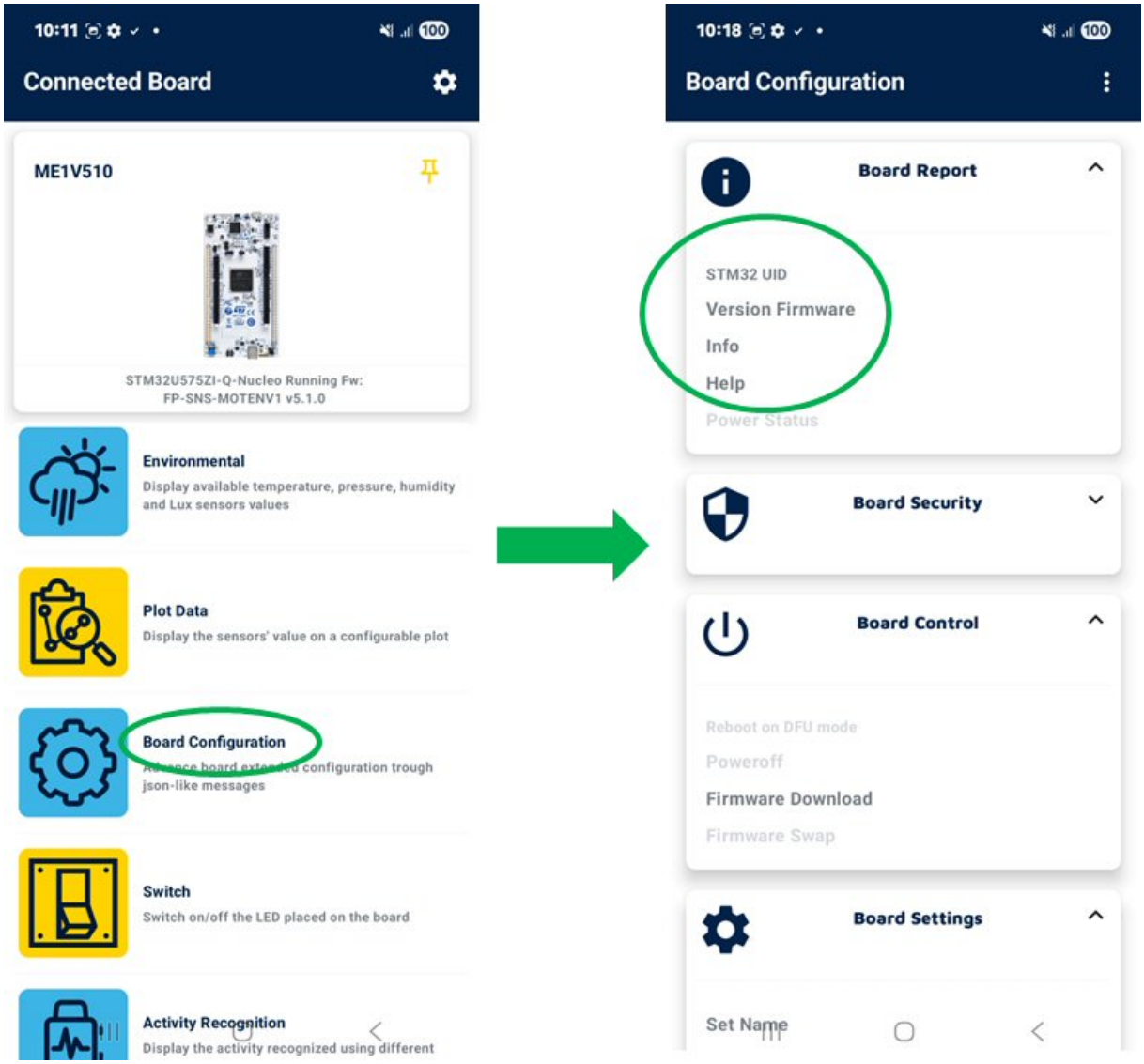
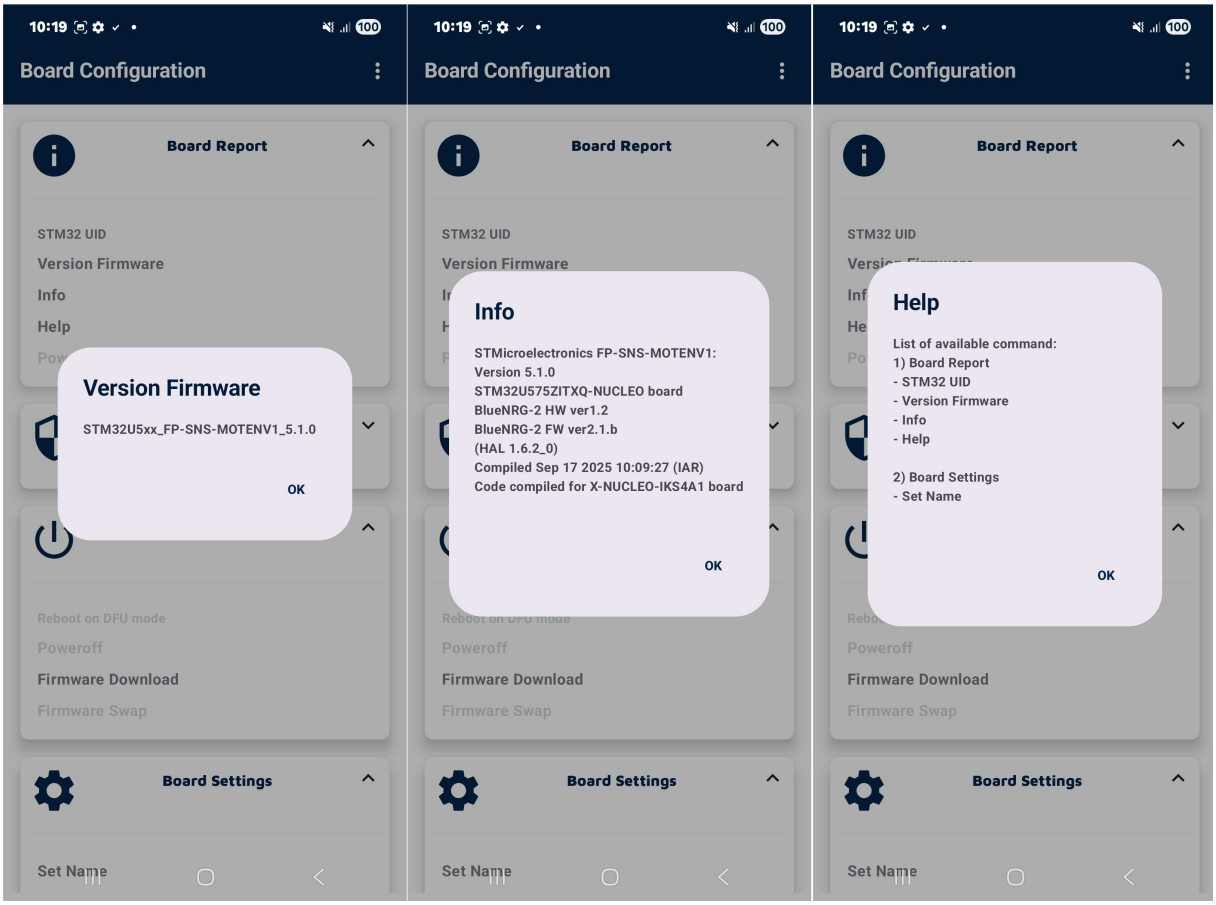
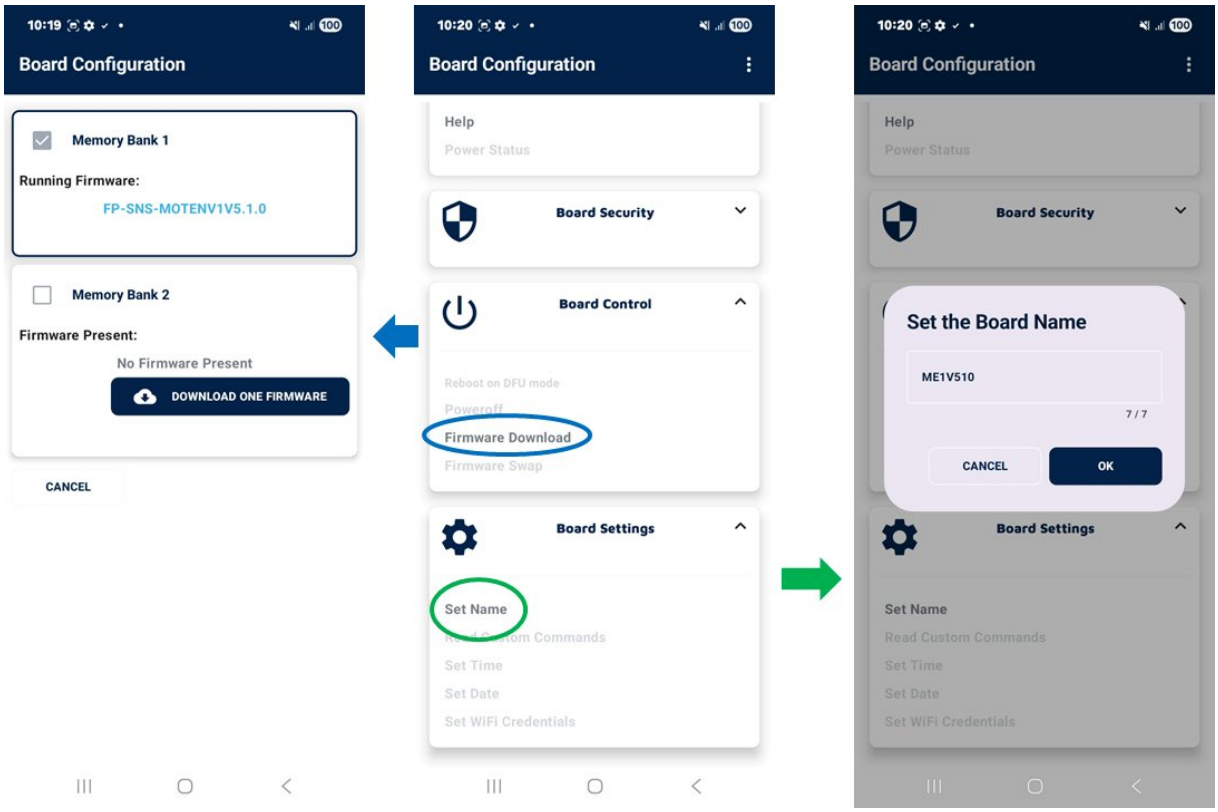


Figure 33. **STBLESensor (Android version) Board Configuration - Commad Version Firmware, Info, Help**



A new IoT node name can selected and firmware download or swap memory bank can be do, too

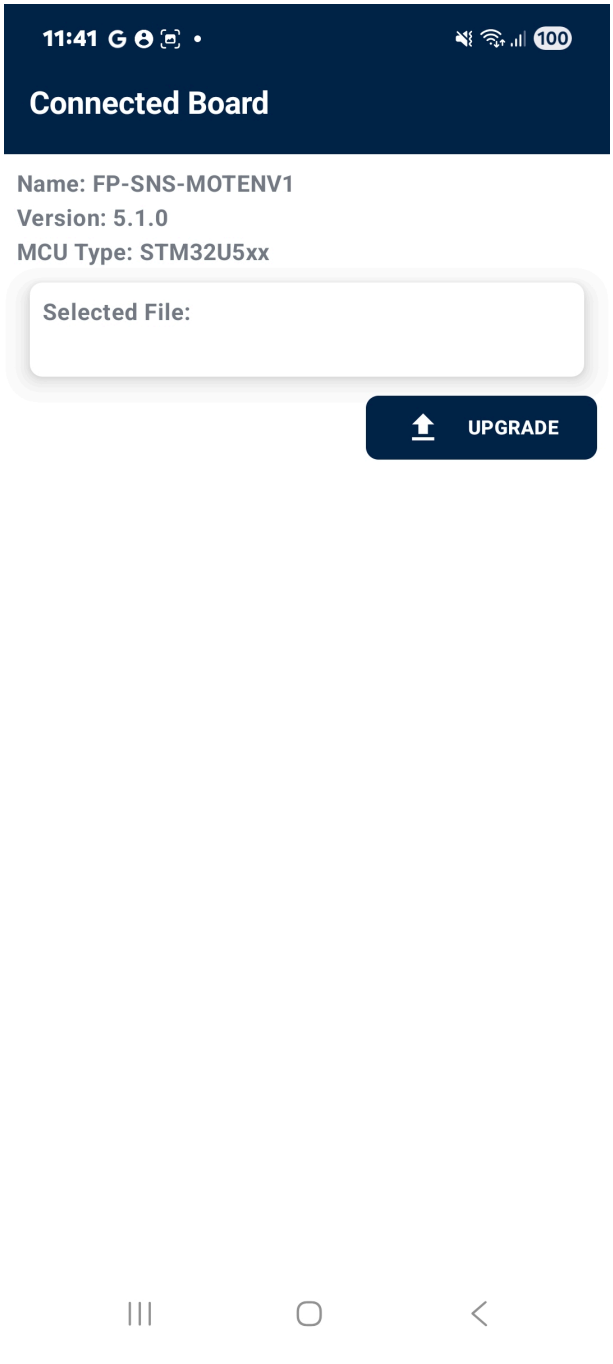
Figure 34. STBLESensor (Android version) Board Configuration - Commad Firmware Download, Swap Bank



1.11 Firmware-over-the-air update with STBLESensor

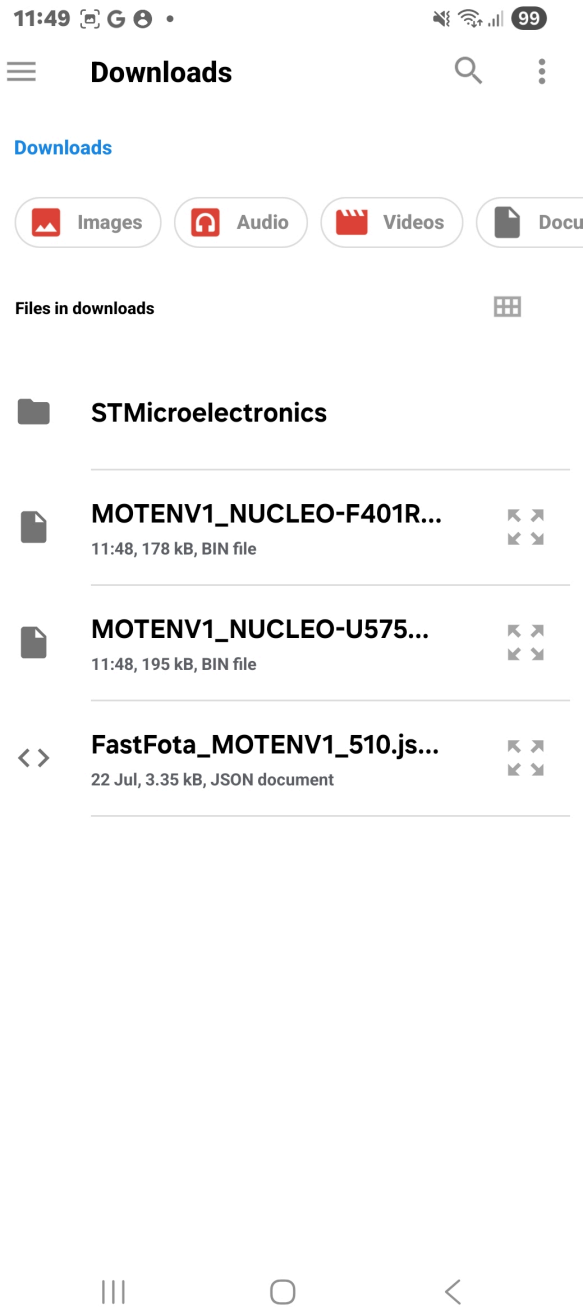
If the 'Firmware upgrade' menu option is selected in the STBLESensor main application page (Figure 14. STBLESensor (Android version) main page following BLE connection), the page below appears.

Figure 35. STBLESensor (Android version) firmware upgrade page



The STBLESensor application shows which version of the FP-SNS-MOTENV1 software is running and the board type. To apply an update, press the red button and select the appropriate update file.

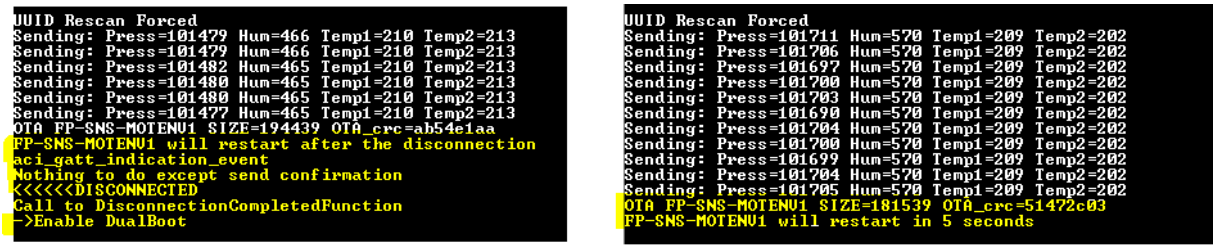
Figure 36. **STBLESensor (Android version) firmware update file selection**



STBLESensor sends to FP-SNS-MOTENV1 an update of a certain byte size and corresponding CRC value.

The figure below shows the terminal window with the debug information returned during FOTA for an STM32 Nucleo platform (STM32F401RE/L476RG) when we use a UART to control FP-SNS-MOTENV1 behavior.

Figure 37. Terminal window information during FOTA

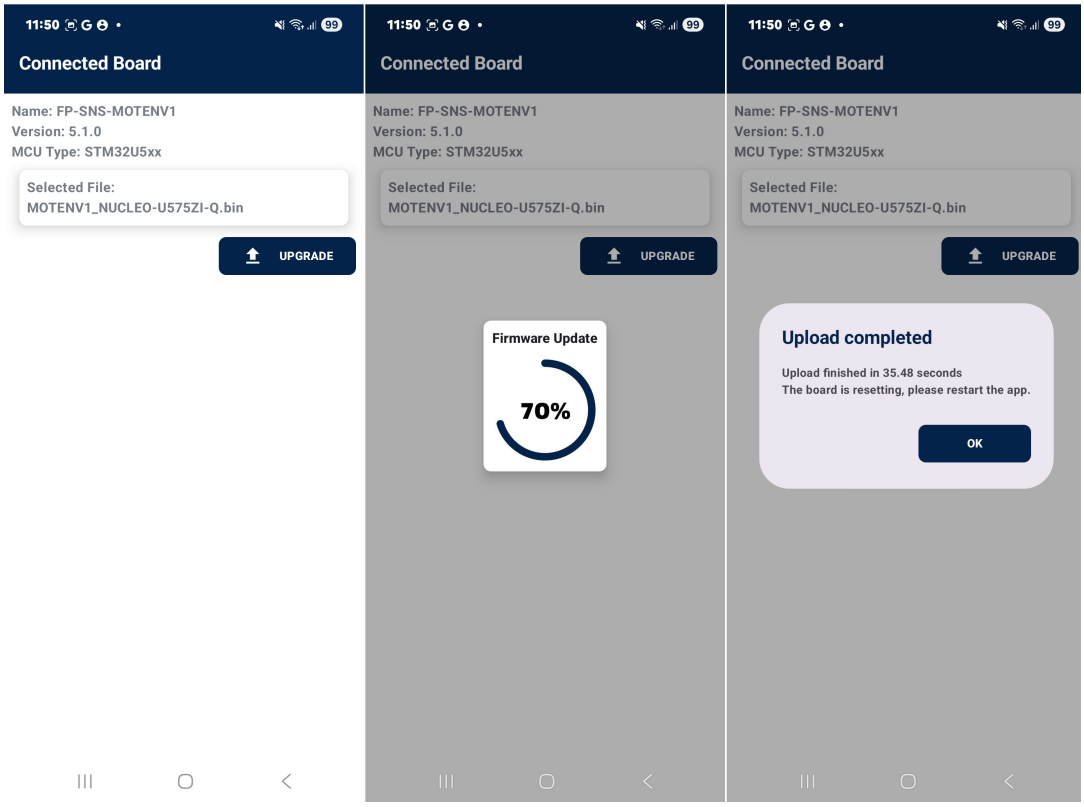


NUCLEO-U575ZI-Q

NUCLEO-F401RE and NUCLEO-L476RG

During the FOTA procedure, the STBLESensor application shows the remaining packets to be sent, and the total update time when the procedure has finished.

Figure 38. STBLESensor (Android version) application feedback during and after FOTA transmission



2 System setup guide

2.1 Hardware description

2.1.1 STM32 Nucleo

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

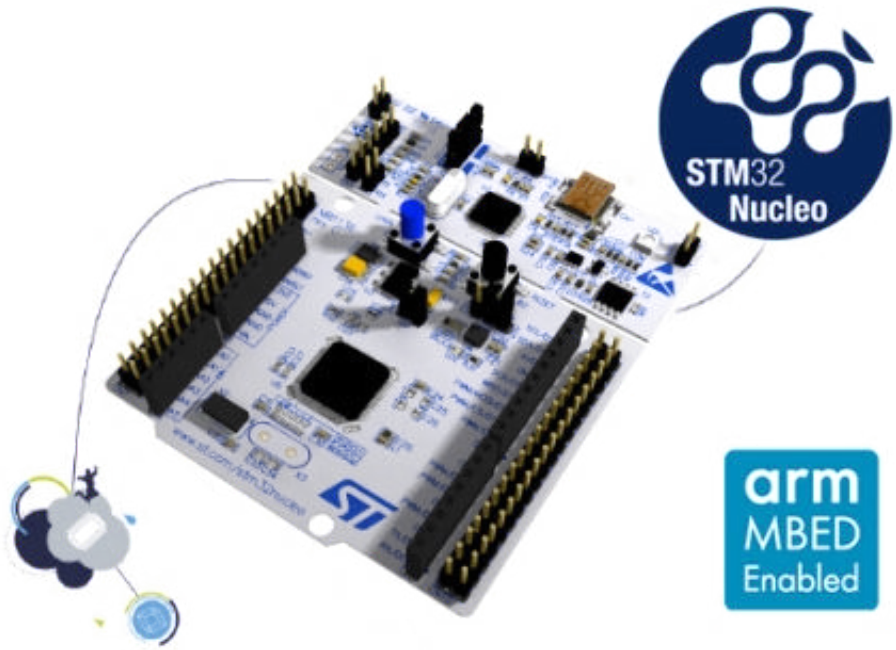
The Arduino connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

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 for different IDEs (IAR EWARM, Keil MDK-ARM, STM32CubeIDE, mbed and GCC/ LLVM).

All STM32 Nucleo users have free access to the mbed online resources (compiler, C/C++ SDK and developer community) at www.mbed.org to easily build complete applications.

Figure 39. STM32 Nucleo board



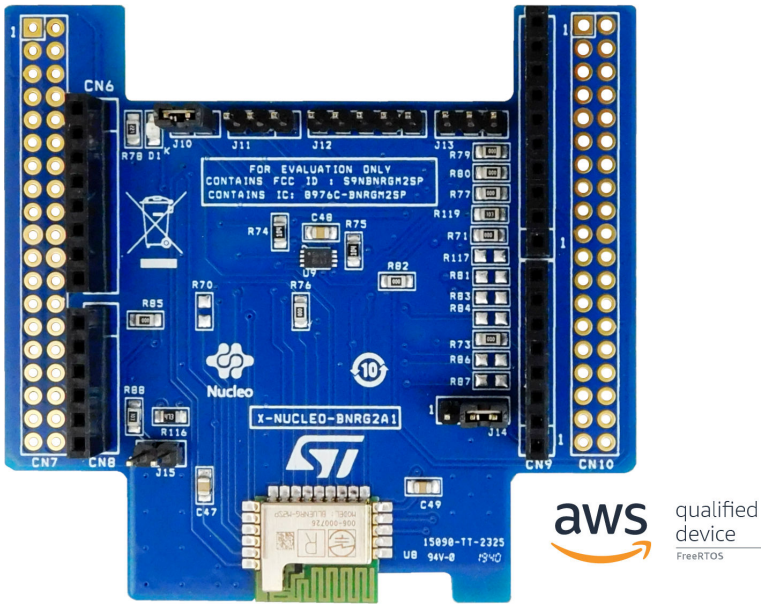
2.1.2 X-NUCLEO-BNRG2A1 expansion board

The X-NUCLEO-BNRG2A1 expansion board provides Bluetooth® Low Energy connectivity for developer applications and can be plugged onto an STM32 Nucleo development board (for example, NUCLEO-L476RG with an ultra-low power STM32 microcontroller) through its Arduino UNO R3 connectors.

The expansion board features the Bluetooth® v5.2 compliant and FCC certified BlueNRG-M2SP application processor module based on the ST BlueNRG-2 System-on-Chip. This SoC manages the complete Bluetooth® Low Energy stack and protocols on its Cortex-M0 core and programmable flash memory, which can accommodate custom applications developed using the SDK. The BlueNRG-M2SP module supports master and slave modes, increased transfer rates with data length extension (DLE), and AES-128 security encryption.

The X-NUCLEO-BNRG2A1 interfaces with the STM32 Nucleo microcontroller via SPI connections and GPIO pins, some of which can be configured through the hardware.

Figure 40. X-NUCLEO-BNRG2A1 expansion board



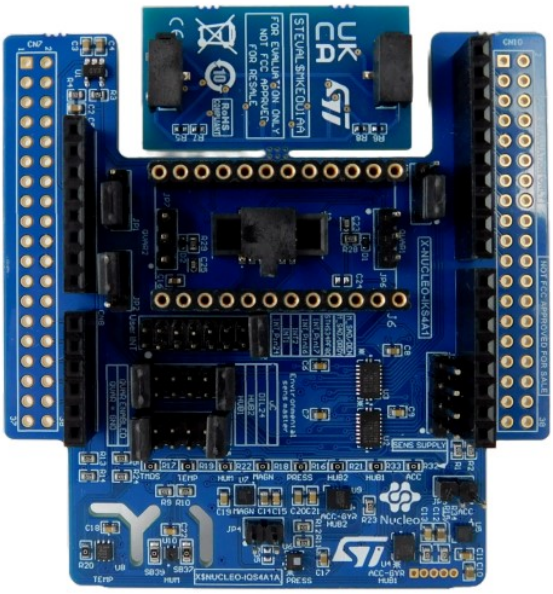
2.1.3 X-NUCLEO-IKS4A1 expansion board

The X-NUCLEO-IKS4A1 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 LSM6DSV16X 3D accelerometer and 3D gyroscope, the LSM6DSO16IS 3D accelerometer and 3D gyroscope with ISPU, the LIS2DUXS12 3D accelerometer, the LIS2MDL 3D magnetometer, the STTS22H temperature sensor and the LPS22DF pressure sensor.

The X-NUCLEO-IKS4A1 interfaces with the STM32 microcontroller via the I²C pin, and it is possible to change the default I²C port.

Figure 41. X-NUCLEO-IKS4A1 MEMS and environmental sensor expansion board



2.2 Software description

The following software components are needed to set up a suitable development environment for creating applications for the STM32 Nucleo equipped with the sensors, and BlueNRG expansion boards:

- FP-SNS-MOTENV1: the Bluetooth low energy and sensors software for STM32Cube. The FP-SNS-MOTENV1 firmware and relative 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-STR) toolchain + ST-LINK
  - Integrated Development Environment for STM32 (STM32CubeIDE) +ST-LINK

After choosing one of the integrated development environments supported by the STM32Cube expansion software, follow the system requirements and setup information provided by the selected IDE provider.

2.3 Hardware setup

The following hardware components are needed:

1. one STM32 Nucleo development platform (order code: NUCLEO-U575ZI-Q, NUCLEO-F401RE, NUCLEO-L476RG or NUCLEO-L053R8)
2. one sensor expansion board (order code: X-NUCLEO-IKS4A1
3. one BlueNRG Bluetooth low energy expansion board (order code: X-NUCLEO-BNRG2A1)
4. one USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC for NUCLEO-F401RE or NUCLEO-L476RG or NUCLEO-L053R8
5. one USB type A to Micro-B USB cable to connect the STM32 Nucleo to the PC for NUCLEO-U575ZI-Q

2.4 System setup

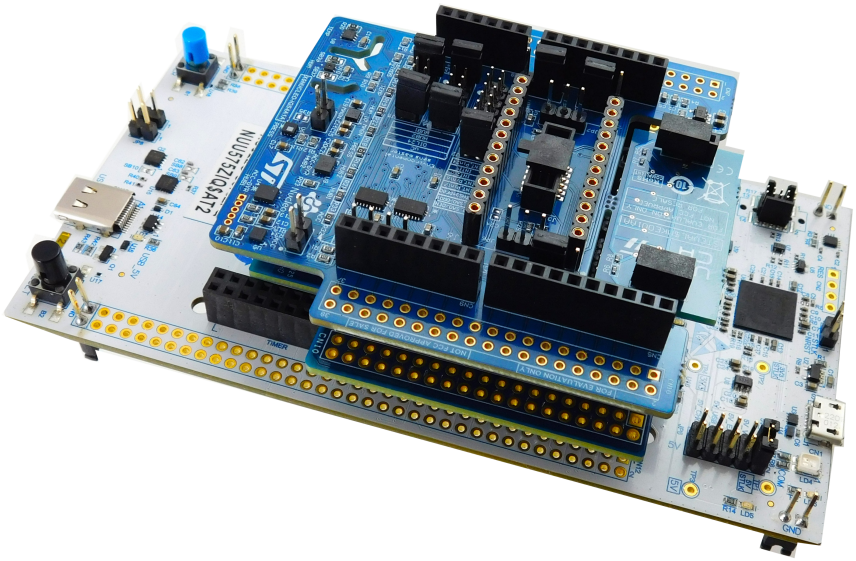
2.4.1 STM32 Nucleo and sensor expansion board setup

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 or STSW-LINK009 on www.st.com (depending on your Windows version).

The X-NUCLEO-BNRG2A1 BlueNRG BLE expansion board is easily connected to theSTM32 Nucleo through the Arduino UNO R3 extension connector.

The sensor board X-NUCLEO-IKS4A1 is easily connected to the X-NUCLEO-BNRG2A1 expansion board through the Arduino UNO R3 extension connector, as shown in the figure below.

Figure 42. X-NUCLEO-IKS4A1 sensor board connected to STM32 Nucleo Board over the X-NUCLEO-BNRG2A1 expansion board



Note: To optimize the performance of the BlueNRG-M2SP module embedded in the X-NUCLEO-BNRG2A1 expansion board and to reduce antenna interference, stack the boards as per the sequence shown above.

Revision history

Table 1. **Document revision history**

Date	Version	Changes
18-Feb-2016	1	Initial release.
13-Apr-2016	2	Throughout document: - minor text changes - added reference to NUCLEO-L053R8 board compatibility - added reference to Nucleo LED status feature Updated Figure 14: "BlueMS (Android version) LSM6DS3/ LSM6DSM hardware features" Added Figure 15: "BlueMS (Android version) LED status"
28-Jul-2016	3	Throughout document: - minor text changes - added reference to STEVAL-STLKT01V1 board compatibility - added Section 2.4:"Flash management", Section 2.5: "The Boot process", Section 2.6: "Firmware over the air (FOTA) update", Section 2.9.1: "Firmware over the air (FOTA) update through BlueMS application", Section 3.1.5: "STEVAL-STLKT01V1 platform", Section 3.3.2: "STEVAL-STLKT01V1 board setup" -changed Figure 11: "BlueMS (Android version) option menu, "Figure 12: "BlueMS (Android version) serial console (stdout/ stderr)"
19-Oct-2016	4	Throughout document: - minor text changes - added reference to the STEVAL-STLKT01V1 VCOM - added reference to the Gas Gauge for STEVAL-STLKT01V1 Added Section : The installation process Updated Section: Android and iOS sample client application
12-Dec-2016	5	Updated Introduction, , Figure 1: "FP-SNS-MOTENV1 software architecture", Figure 10: "Initialization phase", Figure 11: "UART console output when one device is connected to the board" and Section 1.9: "Sample application description" Added Section 2.1.5: "X-NUCLEO-IKS01A2 expansion board" Added X-NUCLEO-IKS01A2 compatibility information throughout document
01-Mar-2017	6	Updated Title, Introduction, Section 1.1: "Overview ", Section 1.2: "Architecture", Section 1.3: "Folder structure", Section 1.4: "Flash management", Section 1.9: "Sample application description", Section 1.10: "Android and iOS BlueMS client application", Section 2.1.6.2: "Features" and Section 2.1.6.3: "Boards included in the kit". Added Section 1.11: "Firmware-Over-The-Air update with BlueMS"
27-Oct-2017	7	Updated Introduction, Section 1.1 Overview , Section 1.2 Architecture, Section 1.3 Folder structure, Section 1.9 Sample application description and Section 1.10 Android and iOS BlueMS client application.
15-Mar-2018	8	Section 1.1 Overview , Section 1.2 Architecture, Section 1.6 The installation process, Section 1.9 Sample application description, Section 1.10 Android and iOS BlueMS client application and Section 2.3.3.1 STM32 Nucleo and sensor expansion board setup.

Date	Version	Changes
		Added Section 2.1.6 P-NUCLEO-IKA02A1 evaluation pack. Throughout the document: - removed references to the STEVAL-STLKT01V1 evaluation board; - added references to the P-NUCLEO-IKA02A1 evaluation pack.
12-Dec-2019	9	Updated Introduction, Section 1.1 Overview, Section 1.2 Architecture, Section 1.3 Folder structure, Figure 4. Bootloader utility content, Section 1.6 The installation process, Section 1.9 Sample application description, Section 1.10 Android and iOS STBLESensor client application, Section 1.11 Firmware-Over-The-Air update with STBLESensor, Section 2.3.2.1 Development toolchains and compilers and Section 2.3.3.1 STM32 Nucleo and sensor expansion board setup. Added Section 2.3.3.2 Important additional hardware information.
11-Jun-2020	10	Updated Section 1.1 Overview, Section 1.2 Architecture, Section 1.3 Folder structure, Section 1.6 The installation process, Section 1.9 Sample application description, Section 1.10 Android and iOS STBLESensor client application, Section 1.11 Firmware-Over-The-Air update with STBLESensor, and Section 2.2 Software description. Added Section 2.1.2 X-NUCLEO-IDB05A2 expansion board.
15-Nov-2021	11	Updated Introduction, *Section 1.1 Overview,* *Section 1.2 Architecture, Section 1.3 Folder structure,* *Section 1.6 The installation process, Section 1.9 Sample application description, Section 1.10 Android and iOS STBLESensor client application, Section 1.11 Firmware-overthe-air update with STBLESensor, Section 2.2 Software description, Section 2.3 Hardware setup*, and Section 2.4.1 STM32 Nucleo and sensor expansion board setup. Removed Section 2.1.2 X-NUCLEO-IDB05A2 expansion board, Section 2.1.5 P-NUCLEO-IKA02A1 evaluation pack, and 2.4.2 Important additional hardware information. Added *Section 2.1.2 X-NUCLEO-BNRG2A1 expansion board.*
08-Jun-2023	12	Updated Introduction and *Section 1.1 Overview*. Added reference to GitHub and STM32CubeMX.
21-Mar-2024	13	Updated Section 1.1: Overview , Section 1.2: Architecture, Section 1.3: Folder structure, Section 1.4: Flash management, Section 1.5: The Boot process, Section 1.6: The installation process for NUCLEO-F401RE or NUCLEO-L476RG board, Section 1.7: Firmware-over-the-air (FOTA) update, Section 1.9: Sample application description, Section 1.10: Android and iOS STBLESensor client application, Section 1.11: Firmware-over-the-air update with STBLESensor, Section 2.2: Software description, Section 2.3: Hardware setupand Section 2.4.1: STM32 Nucleo and sensor expansion board setup. Added Section 1.5: The Boot process and Section 1.5.2: NUCLEO-U575ZI-Q boot process. Removed Section 2.1.3 X-NUCLEO-IKS01A2 expansion board.
27-Oct-2025	14	Updated Section 1.1: Overview , Section 1.2: Architecture, Section 1.2: Architecture, Section 1.5.1: NUCLEO-F401RE and NUCLEO-L476RG Boot Process, Section 1.6: The installation process for NUCLEO-F401RE or NUCLEO-L476RG board, Section 1.9: Sample application description, Section 1.10: Android and iOS STBLESensor client application,

Date	Version	Changes
		Section 1.11: Firmware-over-the-air update with STBLESensor, Section 2.2: Software description, Section 2.3: Hardware setup and Section 2.4.1: STM32 Nucleo and sensor expansion board setup. Removed 2.1.4 X-NUCLEO-IKS01A3 expansion board.

Contents

List of figures

IMPORTANT NOTICE – 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.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The 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 acknowledgment.

The 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 the 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.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. 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.