Getting started with the STM32Cube 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 BlueMS app, to view real-time motion and environmental (such as temperature, relative humidity, carbon monoxide) sensor data.

This package also enables advanced functions such as the sensor data fusion and accelerometer-based real-time activity recognition.

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.

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
- Compatible with BlueMS applications for Android/iOS, to perform sensor data reading, motion algorithm features demo and firmware update (FOTA)
- Example implementation available for the X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A1), P-NUCLEO-IKA02A1 and X-NUCLEO-IDB05A1 (or X- NUCLEO-IDB04A1) connected to a NUCLEO-F401RE or NUCLEO-L476RG or NUCLEO-L053R8 board
- 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
   - CO gas concentration, when the P-NUCLEO-IKA02A1 is present
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 bi-directional 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 when LSM6DS3 is mounted on to DIL24 in X-NUCLEO-IKS01A1 expansion boards, or for LSM6DSL on to X-NUCLEO-IKS01A2:
   - 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 HTS221, LPS25H, LSM6DS0 (or LSM6DS3) and LIS3MDL devices available when an X-NUCLEO-IKS01A1 expansion board is mounted on the STM32 Nucleo board

- the temperature, humidity, pressure, CO gas concentration and motion sensor drivers for the HTS221, LPS22HB, TGS5141, LSM6DSL and LSM303AGR, devices available when an X-NUCLEO-IKS01A2 expansion board plus a P-NUCLEO-IKA02A1 expansion board are mounted on the STM32 Nucleo board
- the CO gas concentration drivers for TGS5141 available when only a P-NUCLEO-IKA02A1 expansion board is mounted on the STM32 Nucleo board

This package is compatible with the BlueMS Android/iOS (Ver. 3.3.0 or higher) application available at the respective Play/iTunes stores, which can be used for displaying information sent via the Bluetooth low energy protocol. BlueMS Version 3.0.0 and above is required for Over-The-Air firmware updates (for X-NUCLEO-IDB05A1 Bluetooth low energy expansion board only and excluding the NUCLEO-L053R8 board).

BlueMS Version 3.6.0 or higher is required to be able to show the Multi Events detection.

BlueMS Version 3.8.0 or higher is required to be able to show the plot for CO gas concentration.

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-MS, sensor expansion board and middleware components for communication with other Bluetooth low energy devices and for sensor data fusion.

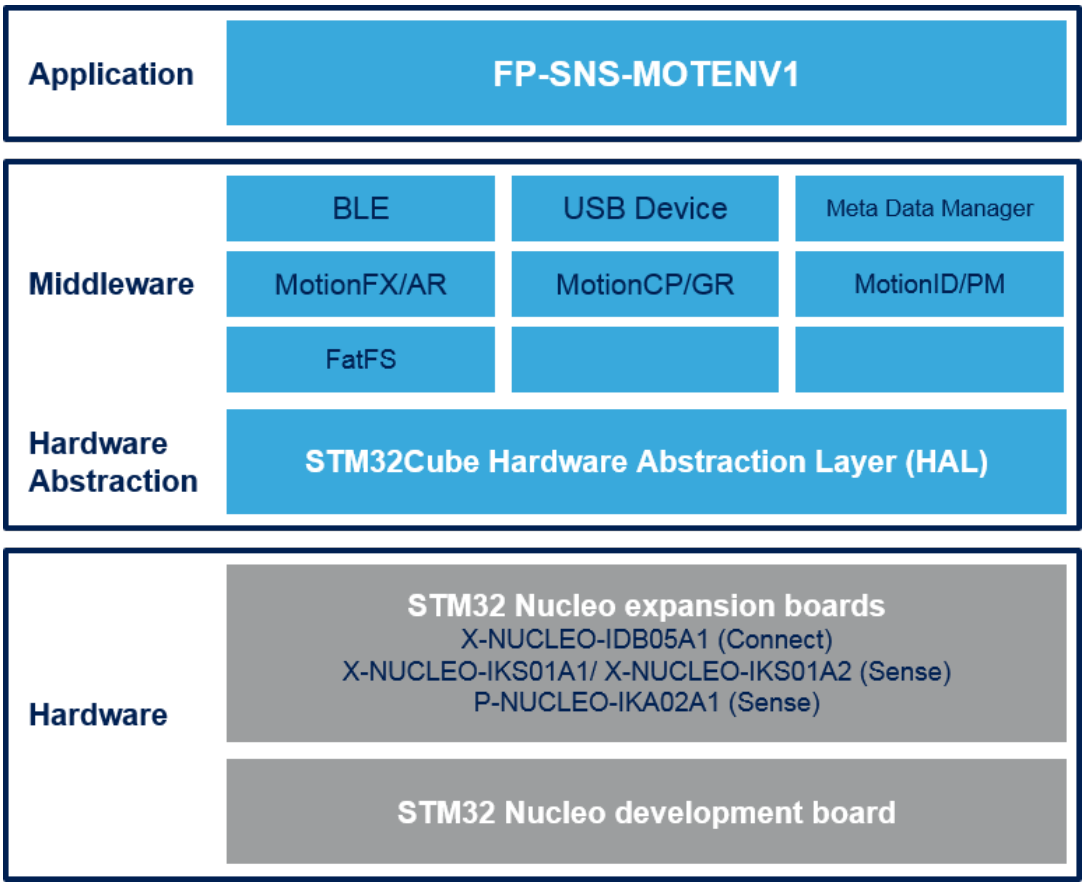
The implementation makes use of low power consumption strategies suitable for this field of application, compliant with the Bluetooth specifications core 4.0 (X-NUCLEO-IDB04A1) or 4.2 (X-NUCLEO-IDB05A1) for STM32 Nucleo boards.

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.

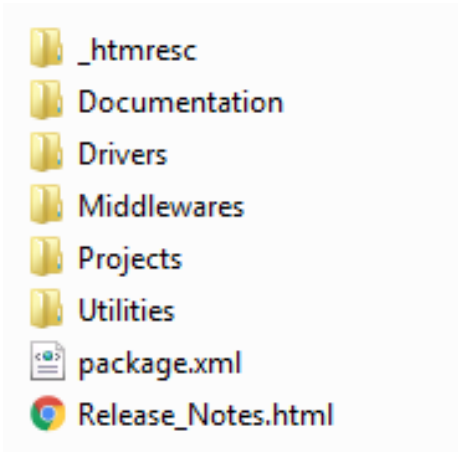
Figure 1. **FP-SNS-MOTENV1 software architecture**



1.3 Folder structure

This section provides an overview of the package 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 Bluetooth low energy, USB Device Library, Generic FAT File System Module (FatFs), the Meta Data Manager, MotionFX (iNEMOEngine PRO) sensors fusion library, MotionAR (iNEMOEngine PRO) activity-recognition library, MotionCP (iNEMOEngine PRO) carry-position recognition library, MotionGR (iNEMOEngine PRO) gesture recognition library, MotionPM real-time pedometer (iNEMOEngine PRO) library, MotionID (iNEMOEngine PRO) 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 platforms through the IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM) and System Workbench for STM32 development environments.
- **Utilities**: contains the boot loader binary ready to be flashed for STM32F401RE and STM32L476RG.

1.4 Flash management

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

1. save the magnetometer calibration values in the Meta Data Manager
2. allow the Firmware-Over-The-Air update

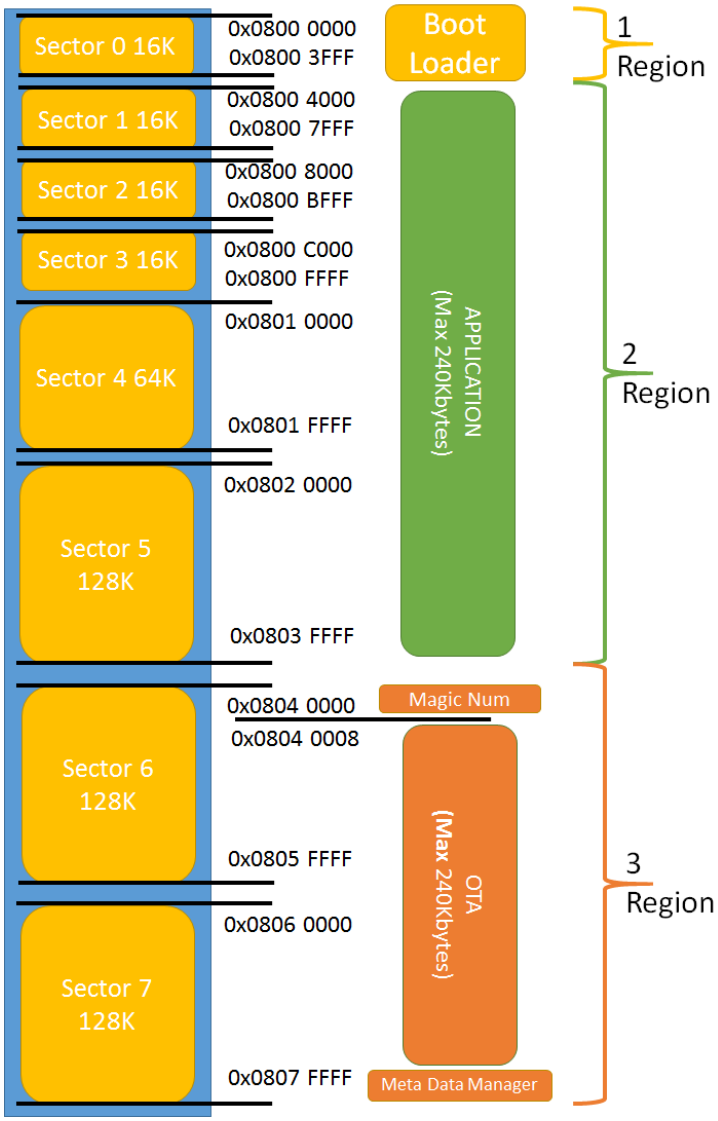
To enable these features, the whole Flash is divided into the following distinct regions:

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**



1.5 The Boot process

Excluding the NUCLEO-L053R8, 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:

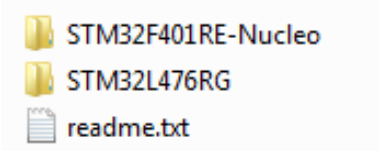
```
define symbol __ICFEDIT_intvec_start__ = 0x08004000;
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__ = 0x08004000;
define symbol __ICFEDIT_region_ROM_end__ = 0x0803FFFF;
define symbol __ICFEDIT_region_RAM_start__ = 0x20000000;
define symbol __ICFEDIT_region_RAM_end__ = 0x20017FFF;
/*-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.

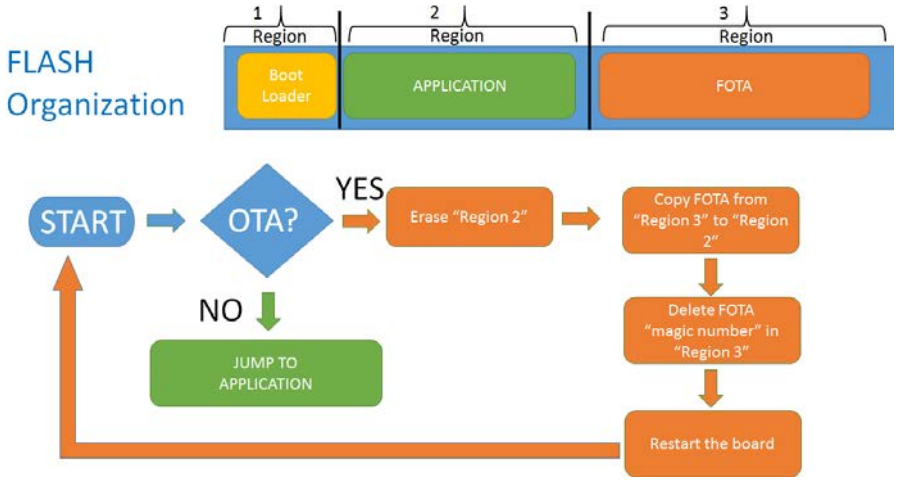
The boot loader checks whether a FOTA is available: the check is based on the presence of a "magic number" at the beginning of the third Flash region.

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. MOTENV1 boot process



1.6 The installation process

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

- pre-compiled application firmware that may be flashed to the correct memory address (0x08004000) of a NUCLEO-F401RE or NUCLEO-L476RG board, by using ST-LINK

Note: This pre-compiled binary is compatible with the FOTA update procedure

- pre-compiled application plus BootLoader firmware that may be flashed to a NUCLEO-F401RE or NUCLEO-L476RG board, by using ST-LINK or drag and drop

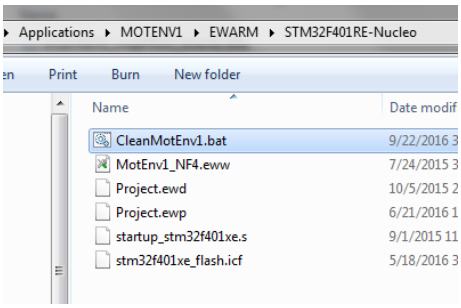
Note: This pre-compiled binary is not compatible with the FOTA update procedure

- pre-compiled application firmware that may be directly flashed to a NUCLEO-L053R8 board, by using ST-LINK or drag and drop

Note: The FOTA update procedure is not available for NUCLEO-L053R8

A CleanMotEnv1.bat script is supplied for each supported platform and IDE combination to simplify firmware installation by saving the FP-SNS-MOTENV1 firmware and BootLoader to the correct position:

Figure 6. Content of a project folder

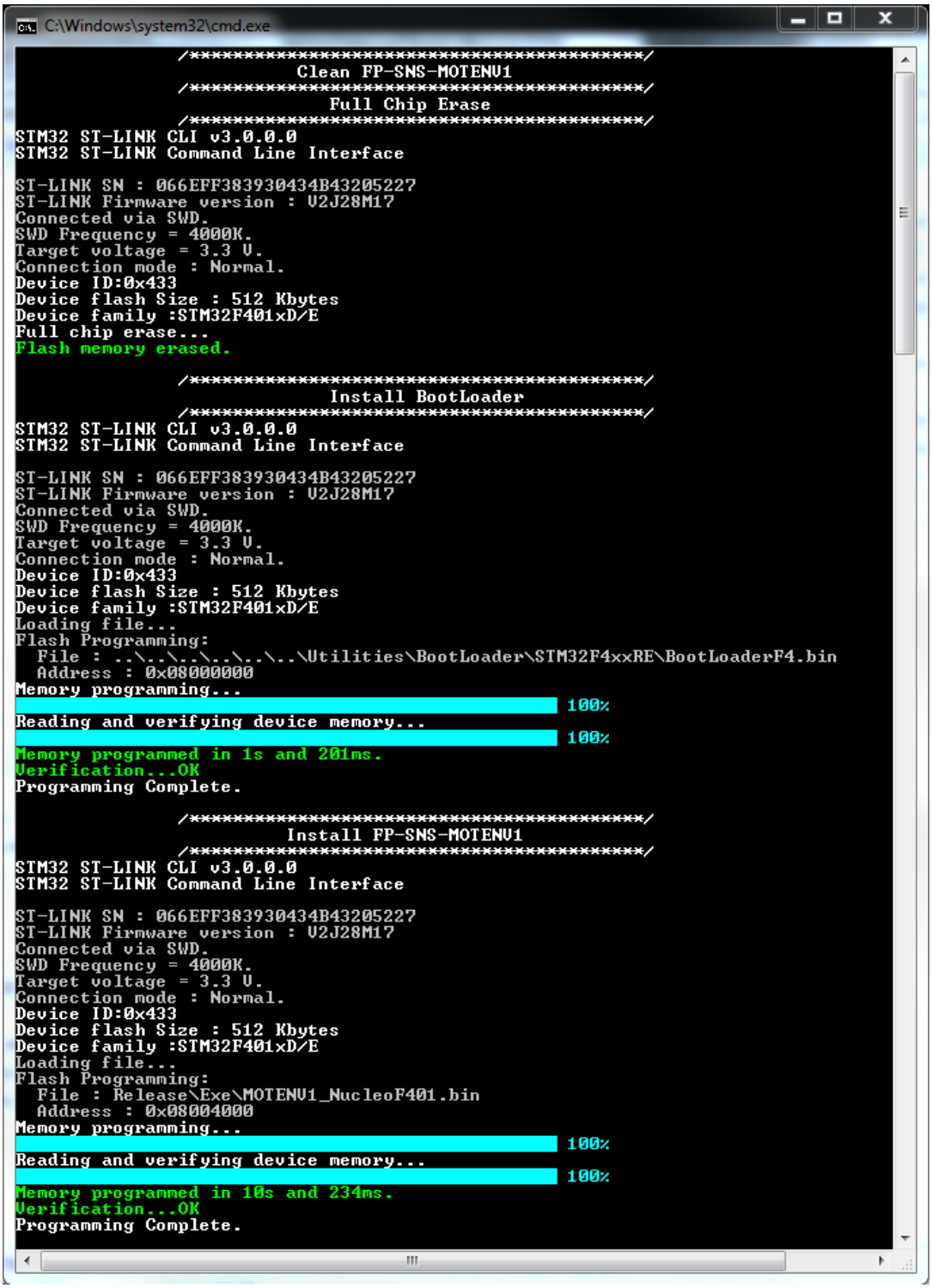


This script:

- 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

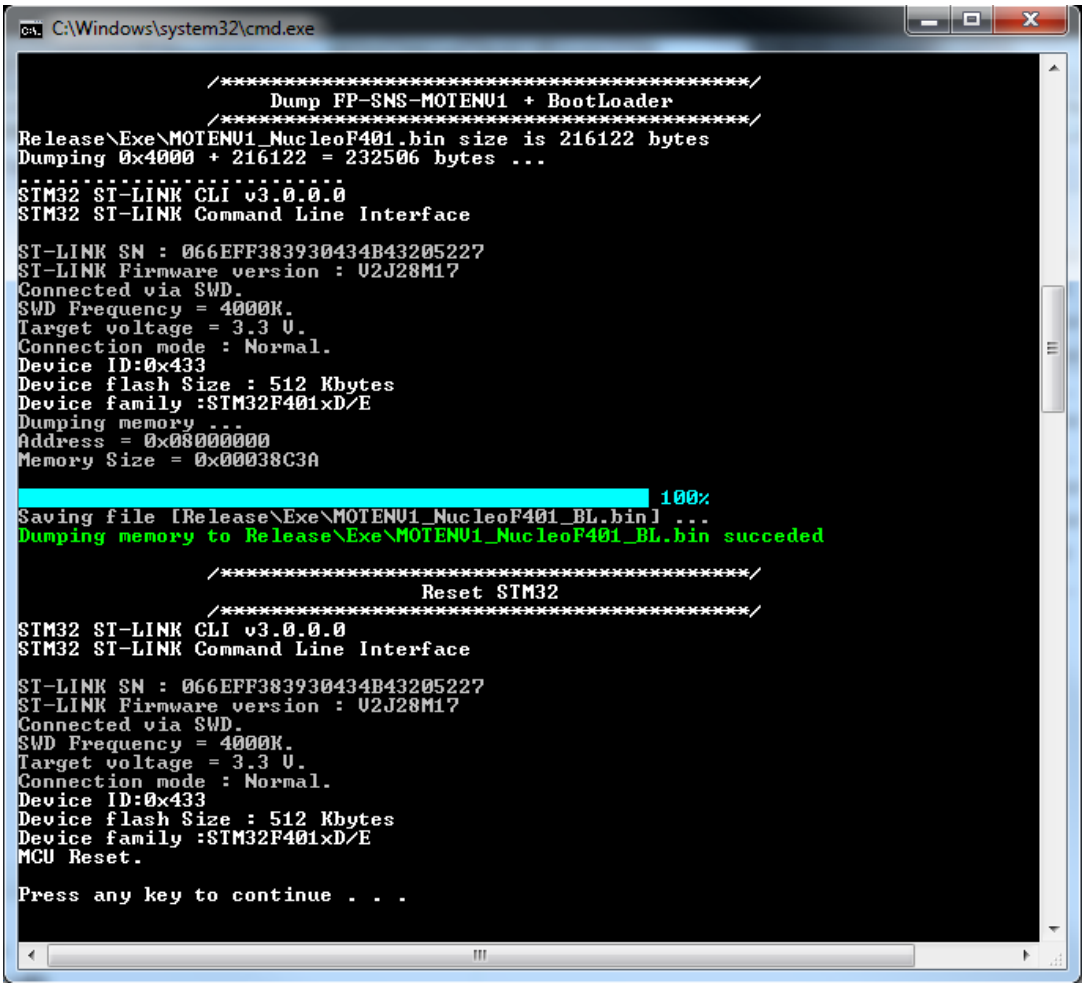
A second script (CleanMotEnv1_IKA02A1.bat) performs the same operation with the project version using the P-NUCLEO-IKA02A1.

Figure 7. **BootLoader and FP-SNS-MOTENV1 installation**



The script also dumps an image file with the BootLoader and the FP-SNS-MOTENV1 firmware. This image file can be directly flashed to the beginning of the Flash in the same way as the image in the Binary folder.

Figure 8. **BootLoader and FP-SNS-MOTENV1 dump process**



For the Linux and iOS operating systems, there is a similar script that uses OpenOCD instead of the ST-LINK command line. This CleanMotEnv1.sh script is only available with the System Workbench IDE for each platform.

In order to function, add the following to the script:

- the installation path for OpenOCD
- the installation path for STM32 OpenOCD scripts
- the Library path for OpenOCD

Below is the section of the CleanMotEnv1.sh to edit:

```
# 1) Set the Installation path for OpenOCD
# example:
#OpenOCD_DIR="C:/Ac6/SystemWorkbench/plugins/fr.ac6.mcu.externaltools.openocd.win32_1.10.0.201607261143/tools/openocd/"
OpenOCD_DIR=""

# 2) Set the installation path for stm32 OpenOCD scripts
# example:
#OpenOCD_CFC="C:/Ac6/SystemWorkbench/plugins/fr.ac6.mcu.debug_1.10.0.201607251855/resources/openocd/scripts"
OpenOCD_CFC=""

# 3) Only for Linux/iOS add openocd library path to _LIBRARY_PATH:
# For iOS example:
```

```
#export DYLD_LIBRARY_PATH=${DYLD_LIBRARY_PATH}:${OpenOCD_DIR}"lib/"

# For Linux example:
#export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${OpenOCD_DIR}"lib/"
```

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 BlueMS application (version 3.0.0 and above) available on their respective stores.

The FOTA is available for NUCLEO-F401RE/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 BlueMS 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 STM32F401/STM32L476 processor to check update integrity.

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

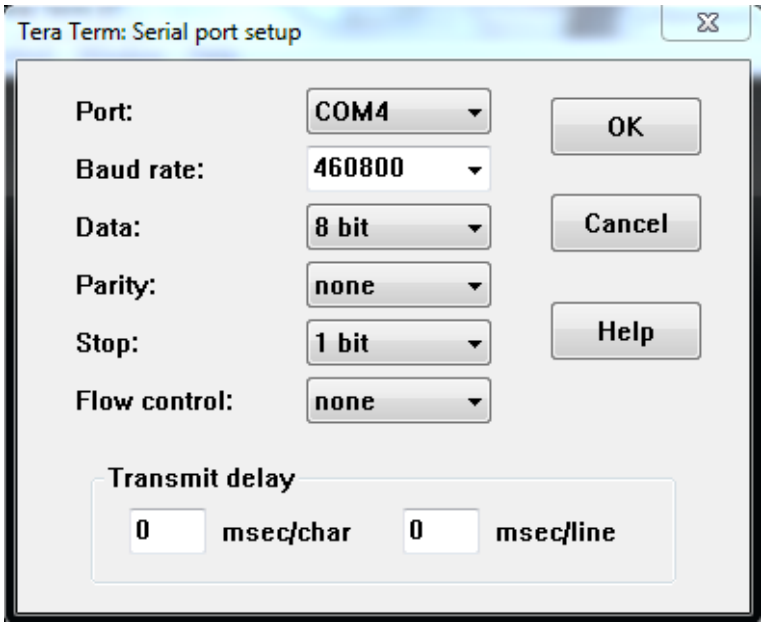
A sample application is provided in the Projects folder for:

- the X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A1) and X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1) expansion boards with the NUCLEO-F401RE or NUCLEO-L476RG or NUCLEO-L053R8
- the X-NUCLEO-IKS01A2, P-NUCLEO-IKA02A1 and X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1) expansion boards with the NUCLEO-F401RE or NUCLEO-L476RG
- the P-NUCLEO-IKA02A1 and X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1) expansion boards with the NUCLEO-F401RE or NUCLEO-L476RG

Ready to build projects are available for multiple IDEs.

With the 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 9. **Terminal setting**



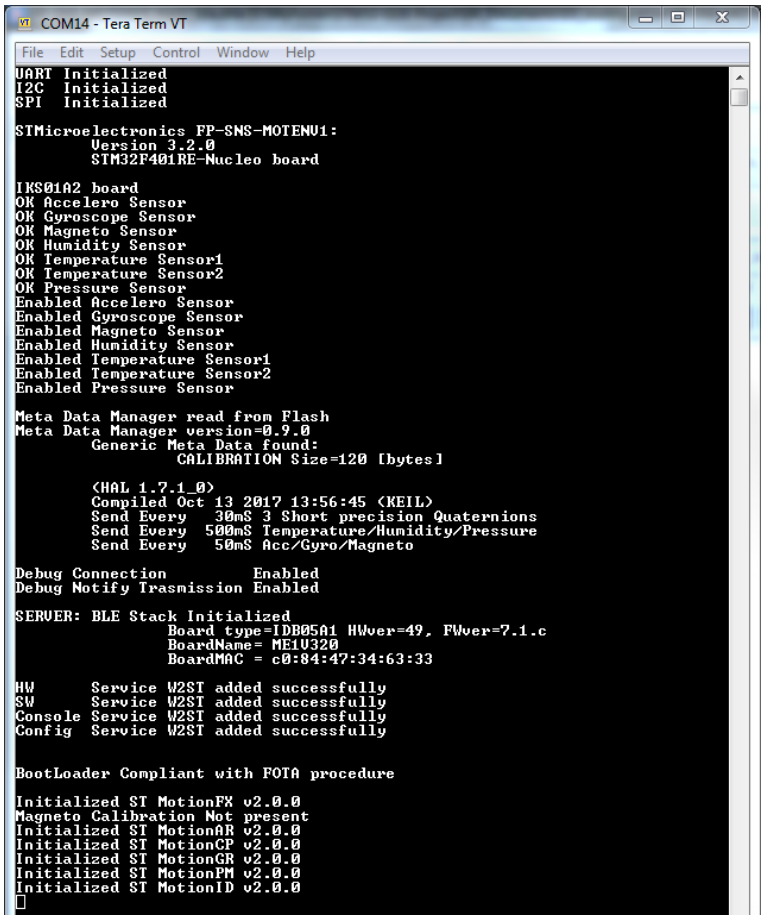
When you first press the reset button on the NUCLEO-F401RE or NUCLEO-L476RG board, the application:

- starts initializing the UART, I²C and SPI interfaces
- determines which MEMS expansion board is connected to STM32 Nucleo board.
- checks whether all the sensors are present and working
- checks whether the LSM6DS3 DIL24 extension is present
- determines which BlueNRG expansion board is connected to the STM32 Nucleo board (X-NUCLEO-IDB04A1 or X-NUCLEO-IDB05A1) and hardware and firmware version information
- indicates if the code is compiled for P-NUCLEO-IKA02A1
- creates a random BLE MAC address
- initializes the BLE hardware service, adding the temperature, humidity, pressure, 3D gyroscope, 3D magnetometer, 3D accelerometer and LED characteristics
- initializes the BLE SW service adding the MotionFX, MotionAR, MotionCP, MotionGR and MotionID libraries (if present in Flash).
- 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 LSM6DS3 DIL24 hardware features (if present).

You can set the gas sensor sensitivity by pressing the button within 5 seconds (only if the code is compiled for P-NUCLEO-IKA02A1).

If the LSM6DS3 DIL24 extension is present, or if the X-NUCLEO-IKS01A2 is mounted, 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 10. 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
- when P-NUCLEO-IKA02A1 is used, the application also sends CO gas concentration in ppm every 500 ms

This application reads the accelerometer, magnetometer and gyroscope values at 100 samples/second. The MotionFX (iNEMOEngine PRO) 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
- `MOTENV1_MOTIONID` to enable the MotionID motion intensity recognition algorithm

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

- on the 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 (iNEMOEngine PRO) library can recognize the following activities:

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

The MotionCP (iNEMOEngine PRO) 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 (iNEMOEngine PRO) library can recognize gestures like:

- pick up
- glance
- wake up in hand

The MotionPM (iNEMOEngine PRO) library counts the number of steps and computes their frequency.

The MotionID (INEMOEngine PRO) library can recognize the following activities:

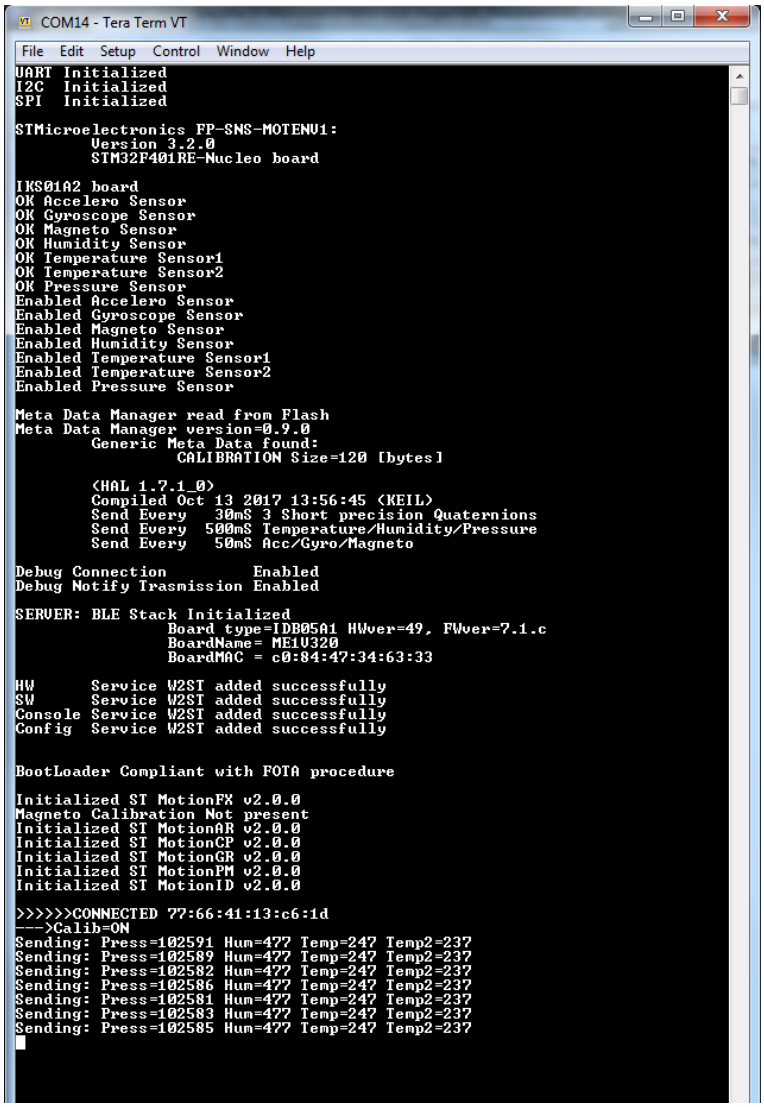
- on desk
- bed, couch
- light movement
- biking
- typing/writing
- slow walking
- walking
- fast walking
- jogging
- fast jogging
- sprinting

The MotionID is enabled via the `MOTENV_MOTIONID` define in MOTENV1_config.h.

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

Note: All the libraries described above are not used for NUCLEO-L053R8 board.

Figure 11. **UART console output when one device is connected to the board**



1.10 Android and iOS BlueMS client application

The FP-SNS-MOTENV1 software for STM32Cube is compatible with the BlueMS Android/iOS applications (ver. 3.0.0 or higher) available at the respective Play/iOS stores. Version 3.0.0 and above is required for Over-The-Air firmware updates (for X-NUCLEO-IDB05A1 Bluetooth low energy expansion boards only).

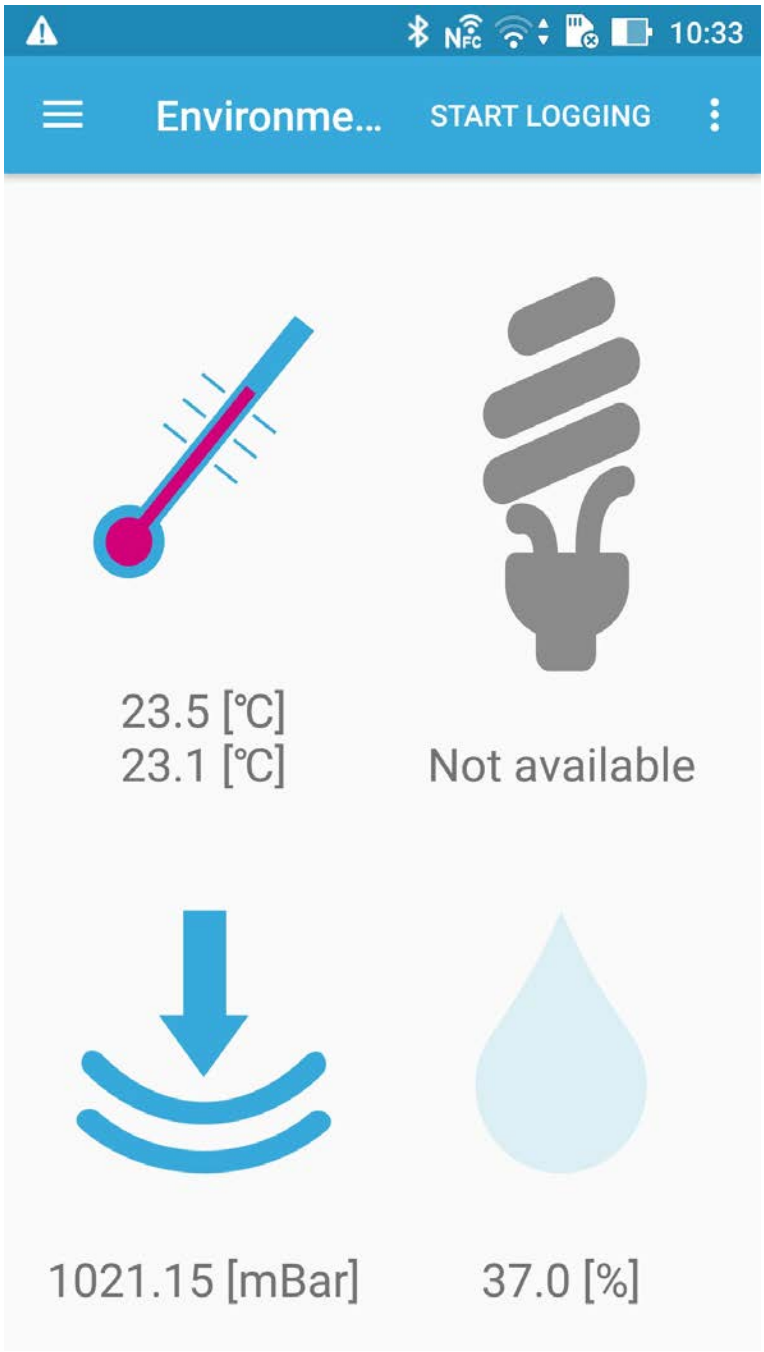
BlueMS Version 3.6.0 or higher is required to be able to show the Multi Events detection.

BlueMS Version 3.8.0 or higher is required to be able to show the CO gas concentration.

We use the Android application in this example.

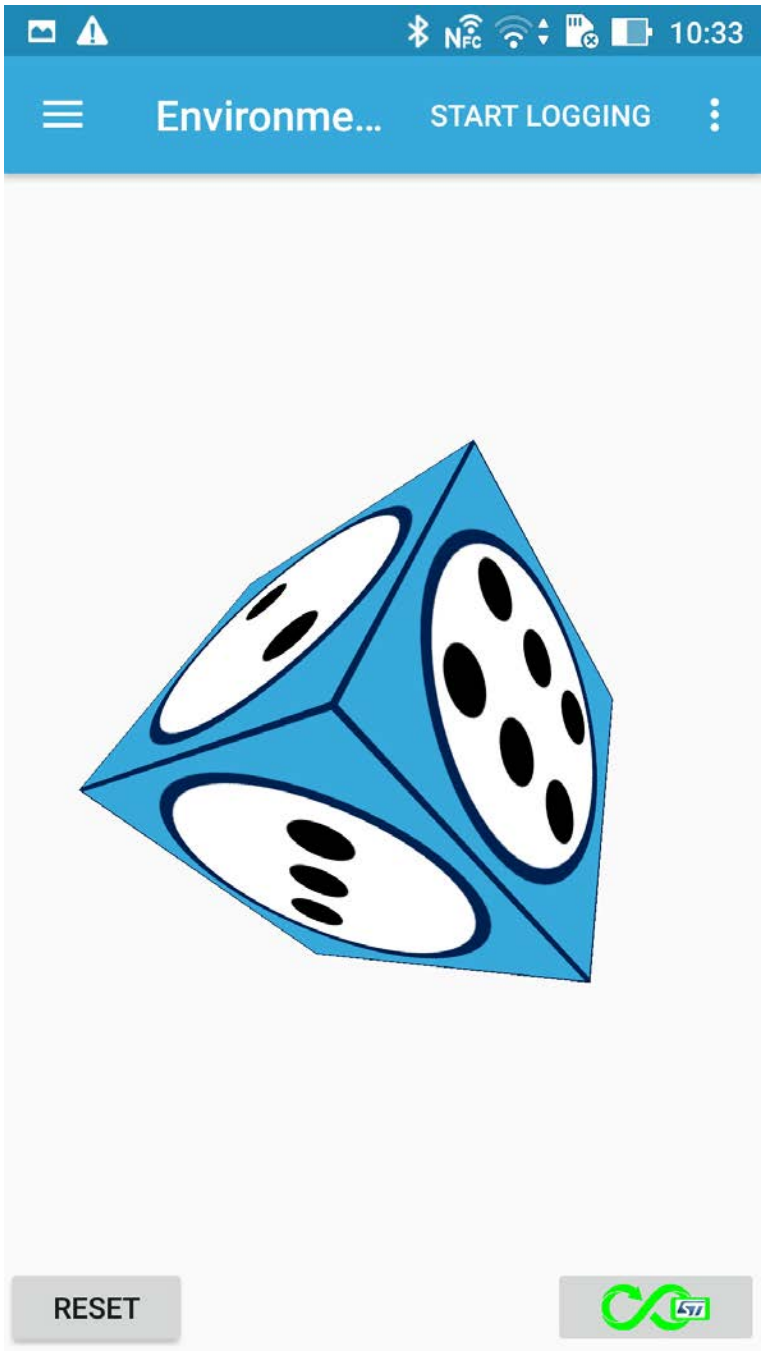
Following connection, BlueMS starts with the main page shown below, where the values of temperature, pressure and humidity are displayed.

Figure 12. BlueMS (Android version) main page following BLE connection



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

Figure 13. **BlueMS (Android version) MotionFX sensor fusion page**

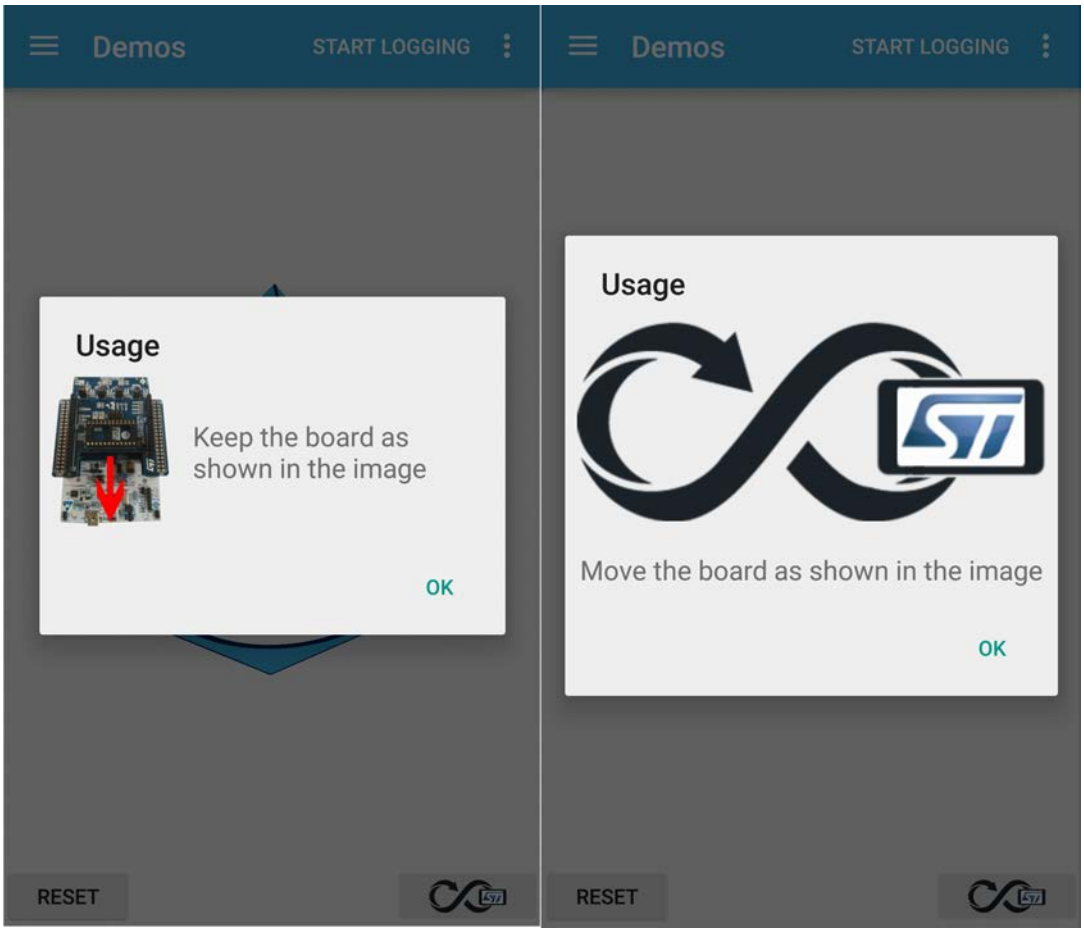


On this page, there are two buttons along the bottom:

- 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.

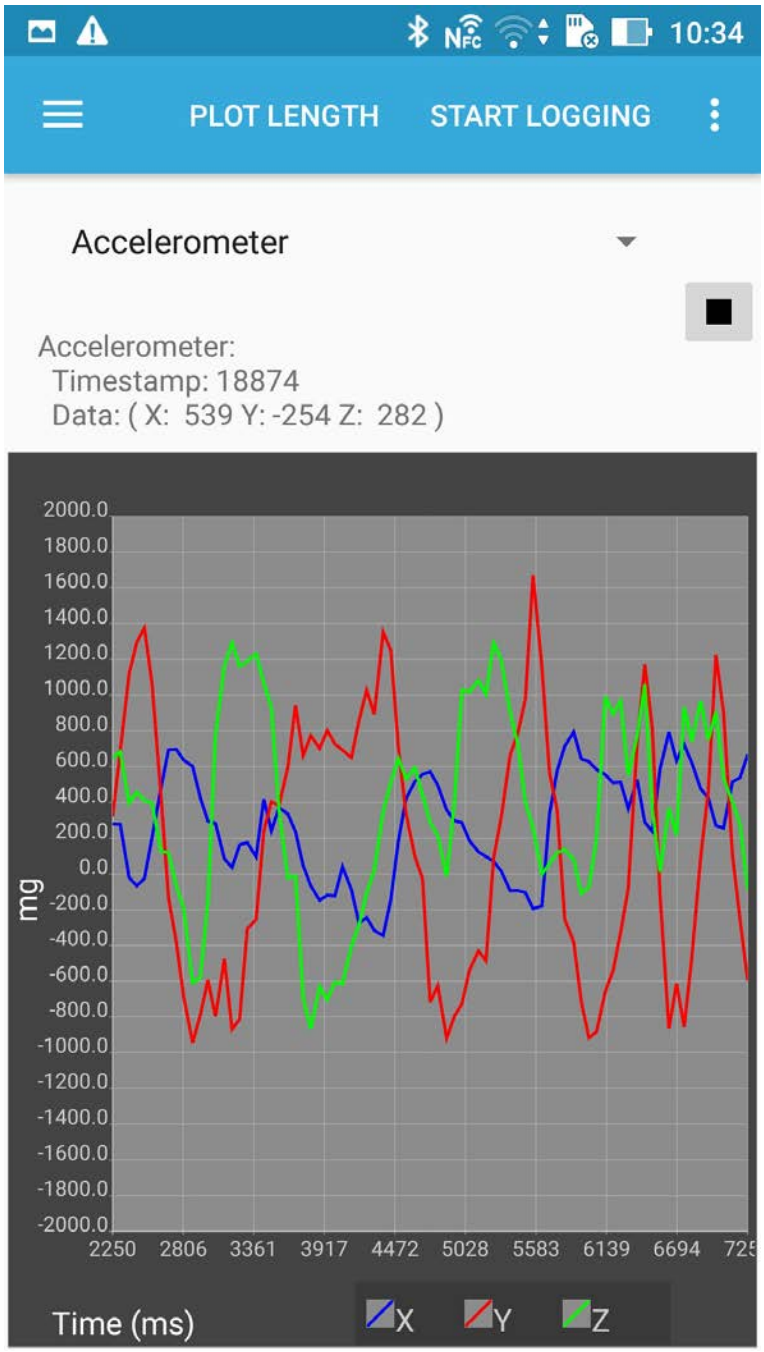
When either button is pressed, the application pops up a window describing how to position the board for correct cube rotation and how to move the board to facilitate calibration (see figure below).

Figure 14. **BlueMS (Android version) popup windows**



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

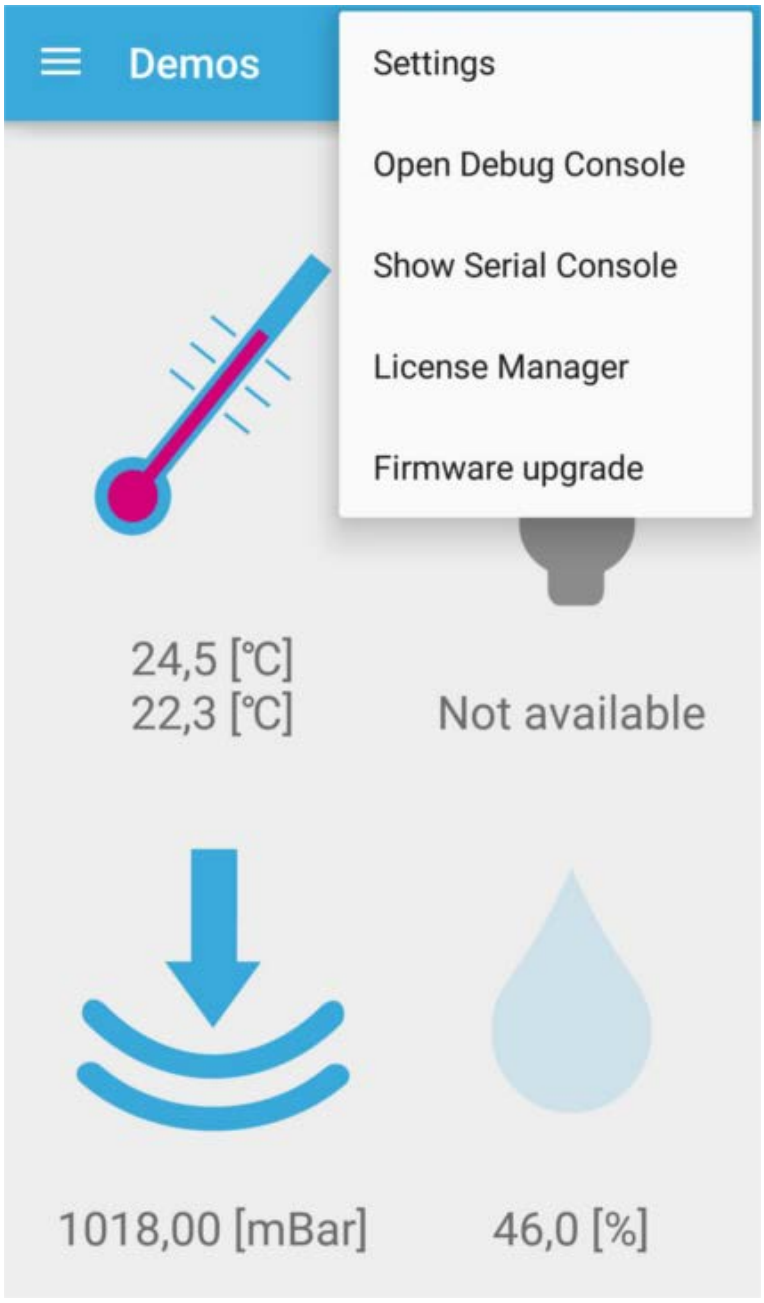
Figure 15. **BlueMS (Android version) accelerometer plot**



In the option menu below, you can open:

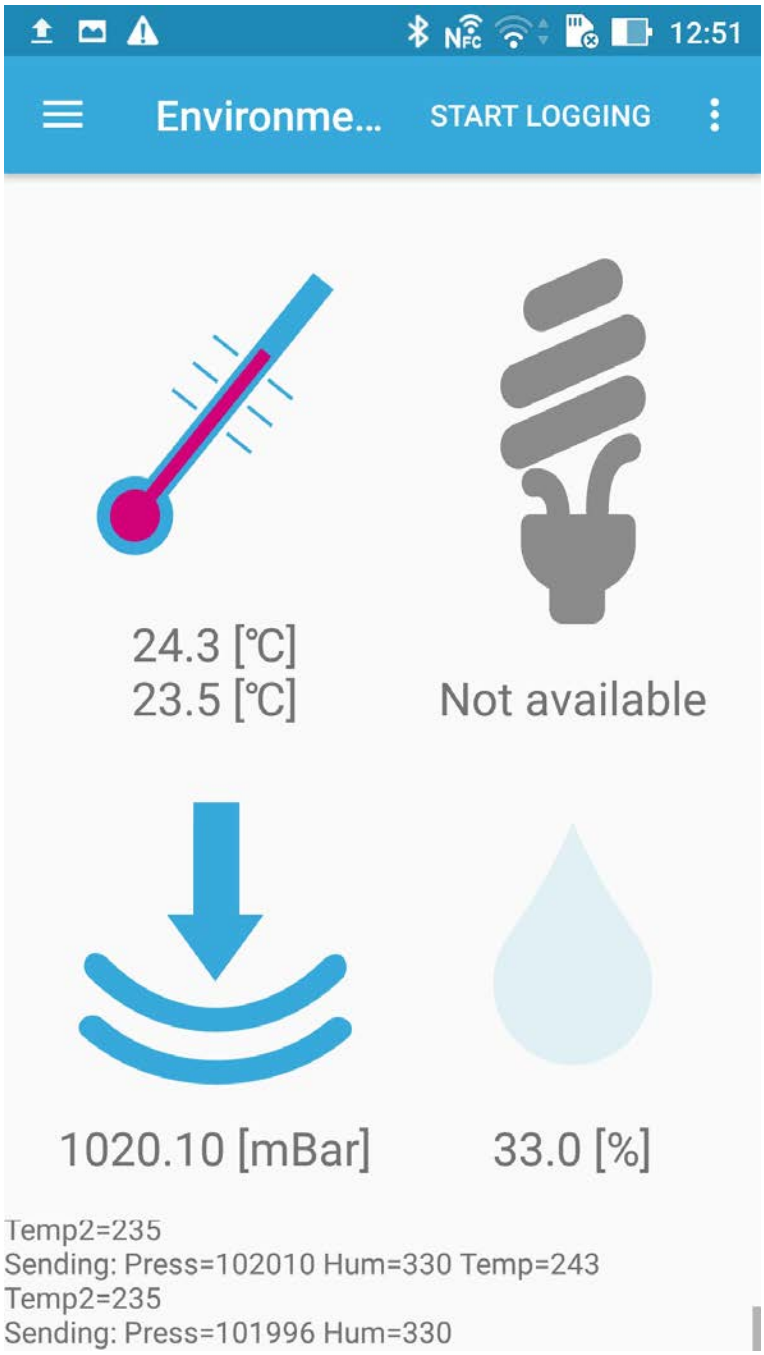
- Serial or Debug (with stdin) console
- Firmware upgrade

Figure 16. **BlueMS (Android version) menu selection**



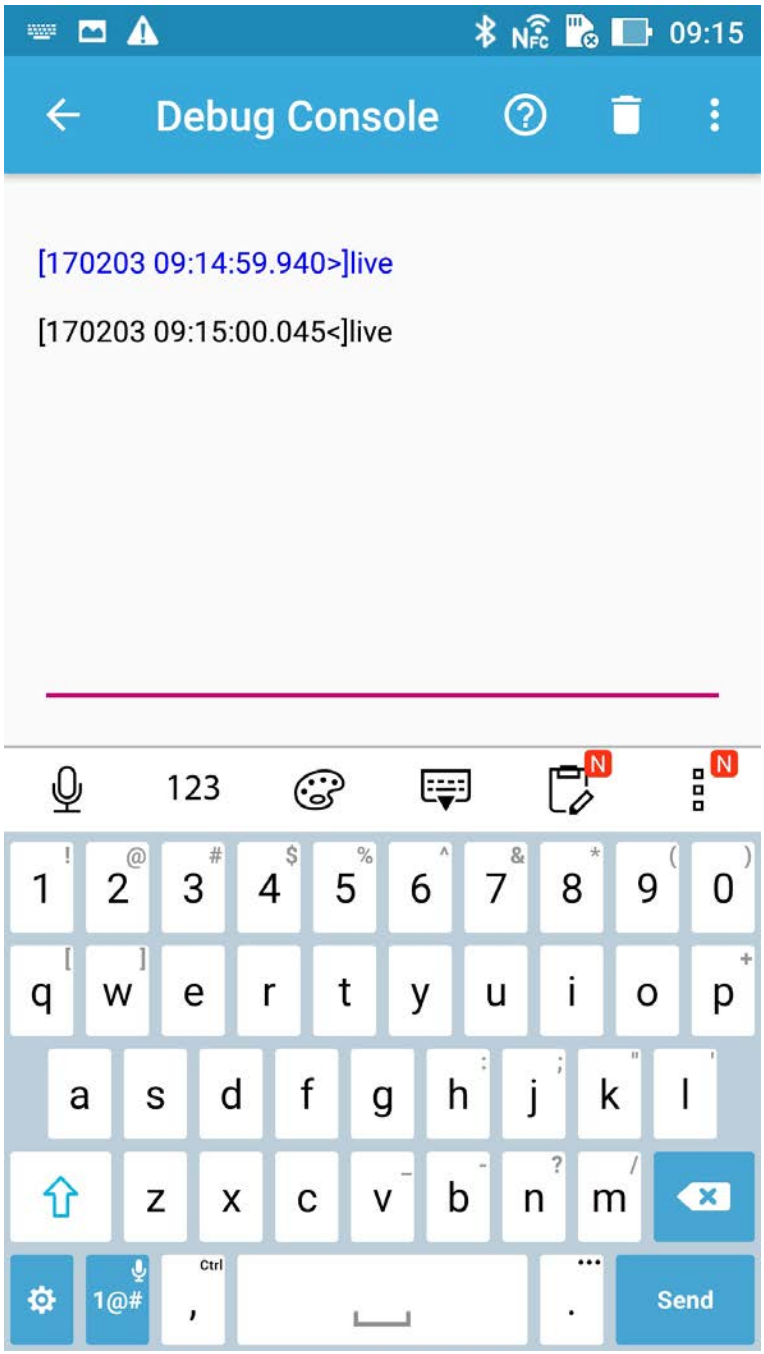
If the Serial console is enabled, stdout/stderr is displayed, as shown below.

Figure 17. **BlueMS (Android version) Serial console (stdout/stderr)**



If the Debug console is enabled, stdin is displayed and any message written in the Debug console triggers a reply with the same message, as shown below.

Figure 18. **BlueMS (Android version) Debug console (stdin/stdout/stderr)**



By the debug consol, there is the possibility to change the sensors value transmission frequency.

- Change the sensors value transmission frequency for temperature/humidity/pressure with the command:
  - @TM: the application sends enviromental data every 5 s
  - @TH: the application sends enviromental data every 1 s
  - @TL: the application sends enviromental data every 100 ms
  - @TD: the application sends enviromental data as default (500 ms).
- Change the sensors value transmission frequency for 3D accelerometer, 3D gyroscope and 3D magnetometer with the command:
  - @AM: the application sends the data every 5 s

- @AH: the application sends the data every 1 s
- @AL: the application sends the data every 100 ms
- @AD: the application sends the data as default (50 ms).

When data logging starts, the other BlueMS app functions are disabled and the data logging goes on even if the app is closed.

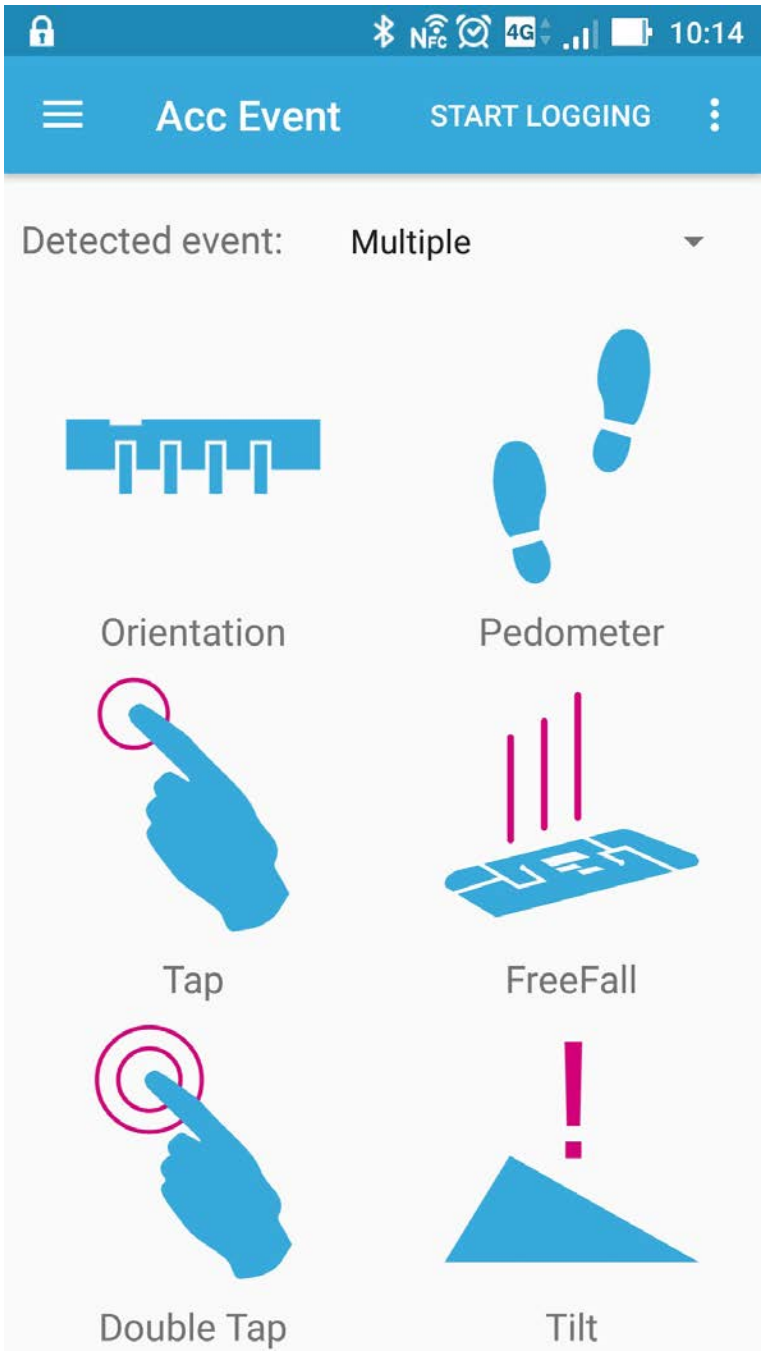
It is necessary to open the app if the data logging must be stopped.

There is another page where you can choose which hardware feature to enable and view the events on the same page from:

- LSM6DS3, if mounted on DIL24 on X-NUCLEO-IKS01A1 expansion board for NUCLEO-F401RE and NUCLEO-L476RG boards
- LSM6DSL on X-NUCLEO-IKS01A2 expansion board for NUCLEO-F401RE and NUCLEO-L476RG boards

The multiple hardware feature is the default setting.

Figure 19. **BlueMS (Android version) multiple hardware feature**



From the **Accelerometer Events** menu, a single harware feature can be selected:

Figure 20. **BlueMS (Android version) hardware feature menu**

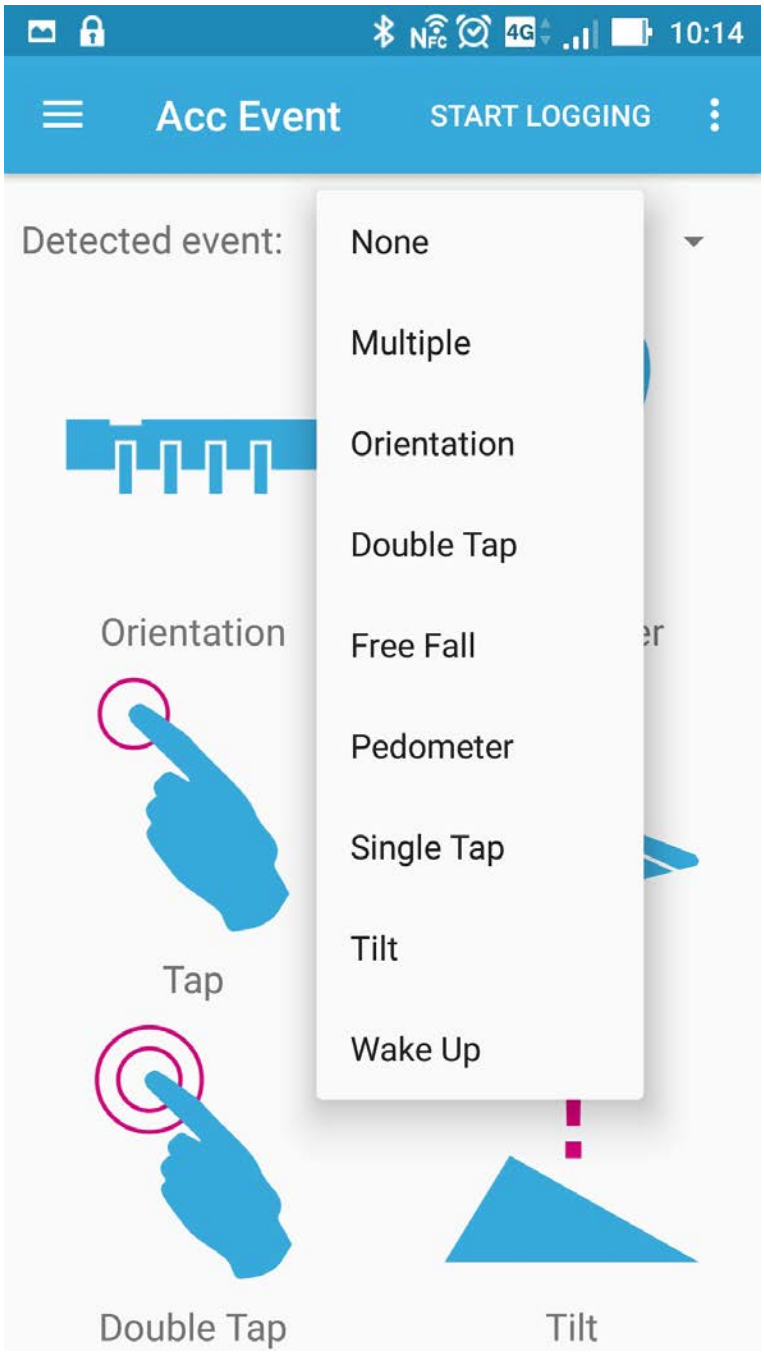
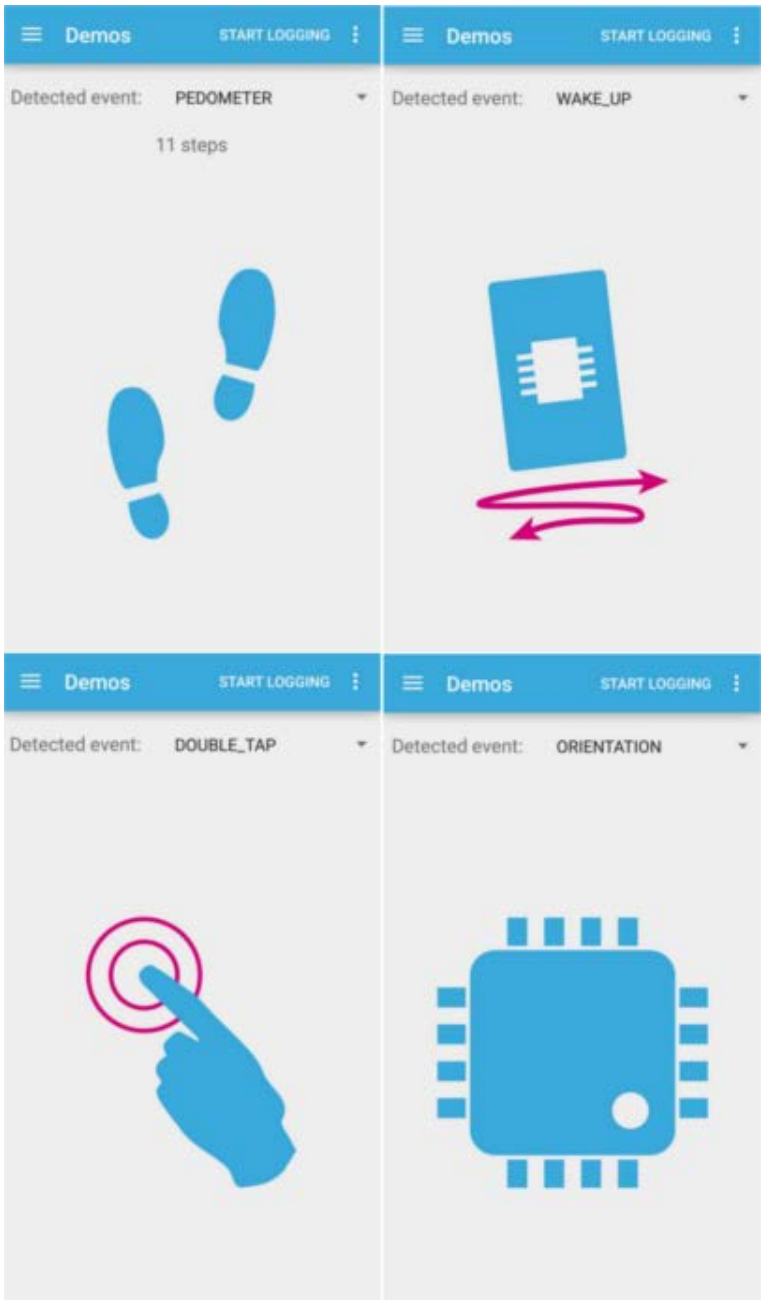


Figure 21. BlueMS (Android version) hardware feature examples: pedometer, wake up, orientation, double tap



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

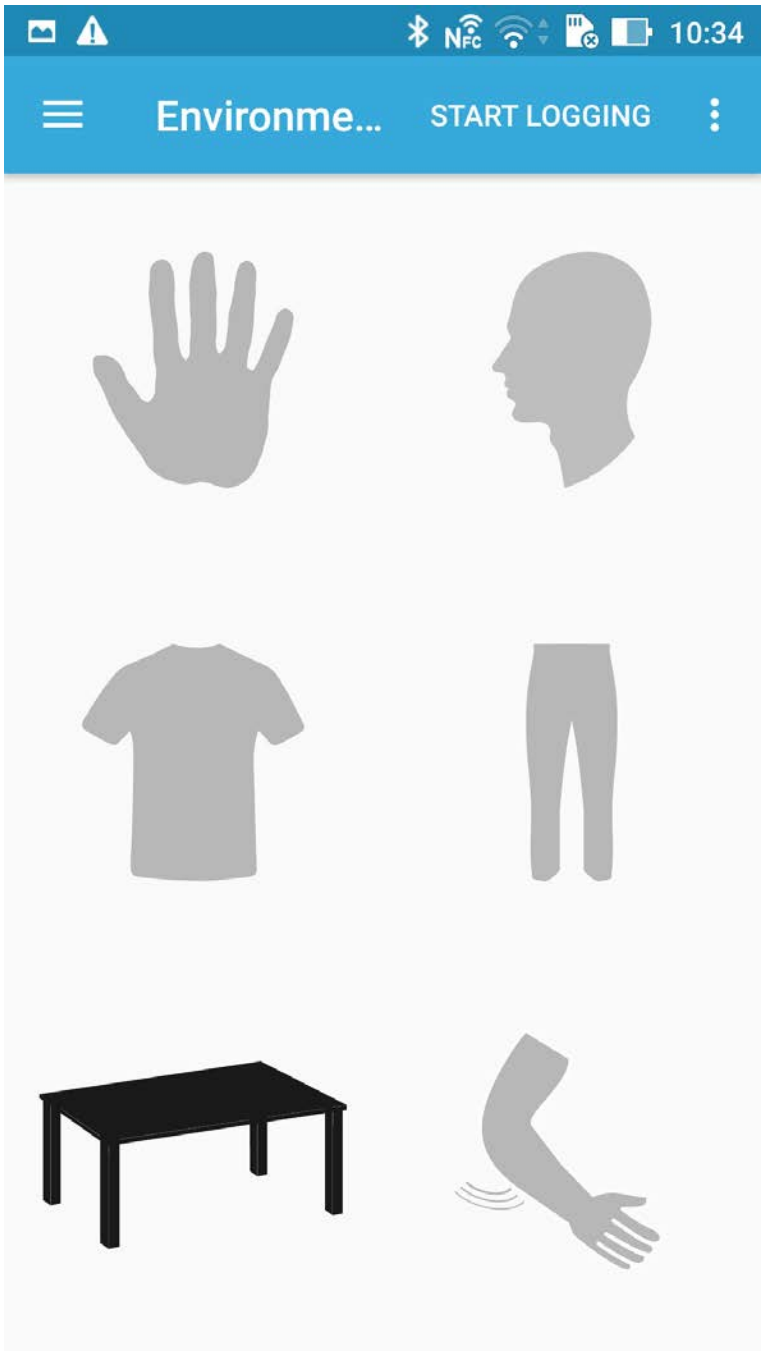
- Stationary
- Walking
- Fast walking
- Jogging
- Biking
- Driving

Figure 22. **BlueMS (Android version) MotionAR activity recognition page**

If the MotionCP algorithm is enabled, 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

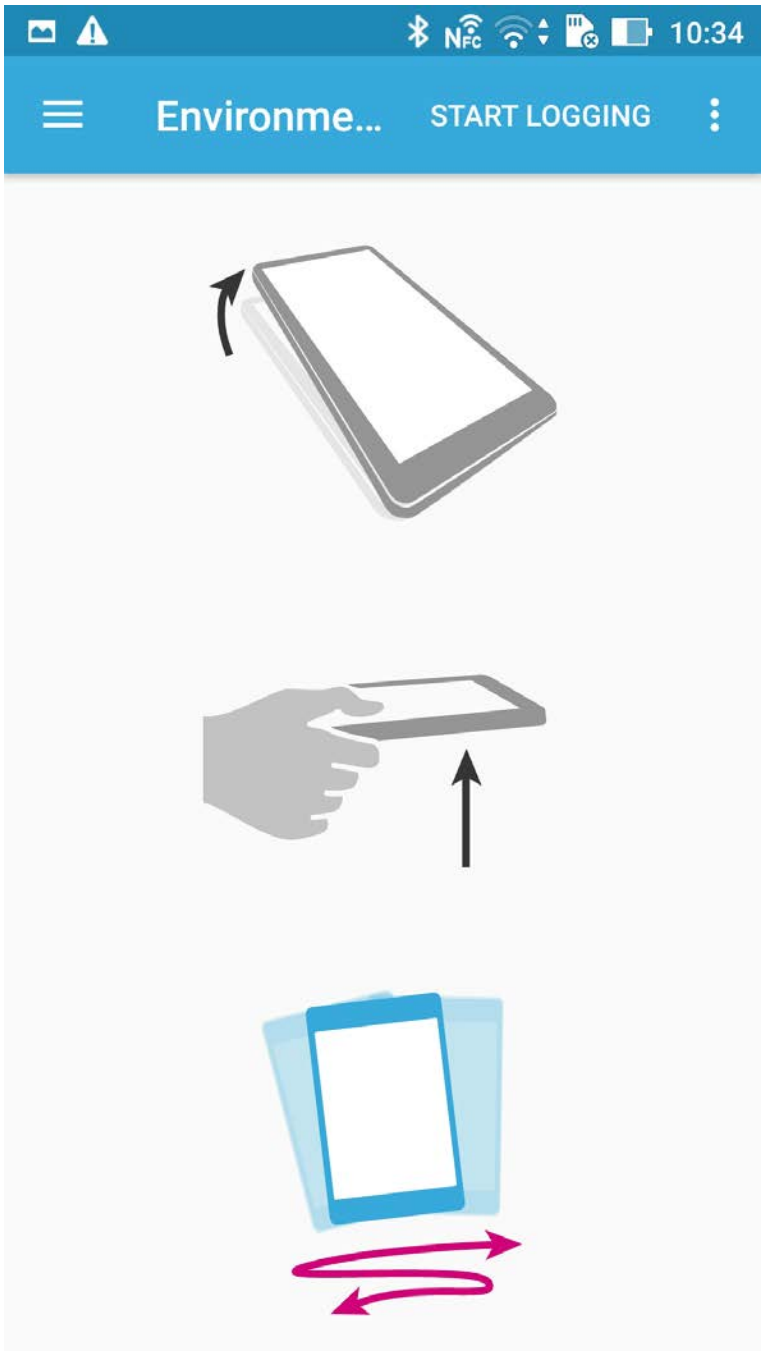
Figure 23. **BlueMS (Android version) MotionCP carry position recognition page**



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

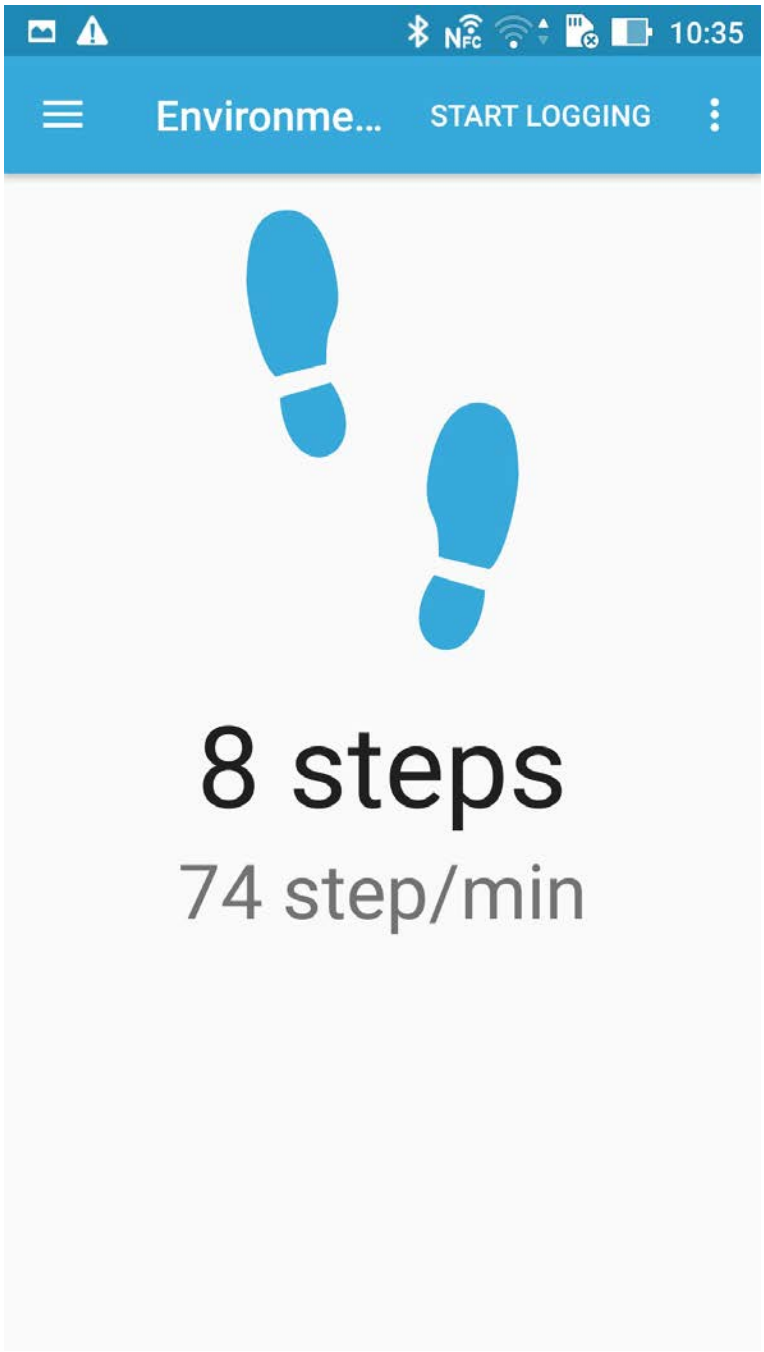
- pick up
- glance
- wake up in hand

Figure 24. **BlueMS (Android version) MotionGR gesture recognition page**



If the MotionPM algorithm is enabled, the page shown below is available with pedometer information:

Figure 25. **BlueMS (Android version) MotionPM pedometer page**

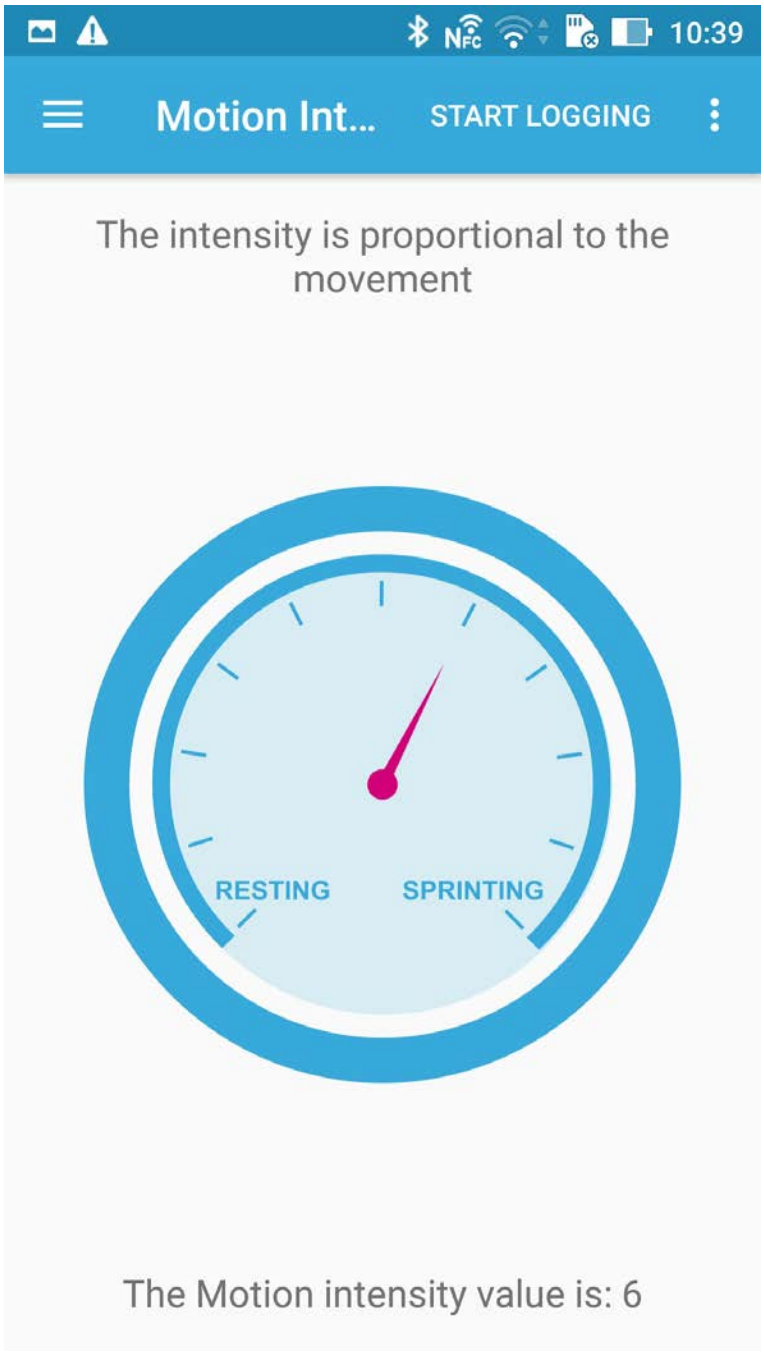


The following page shows the LED on/off control.

Figure 26. **BlueMS (Android version) board LED control**

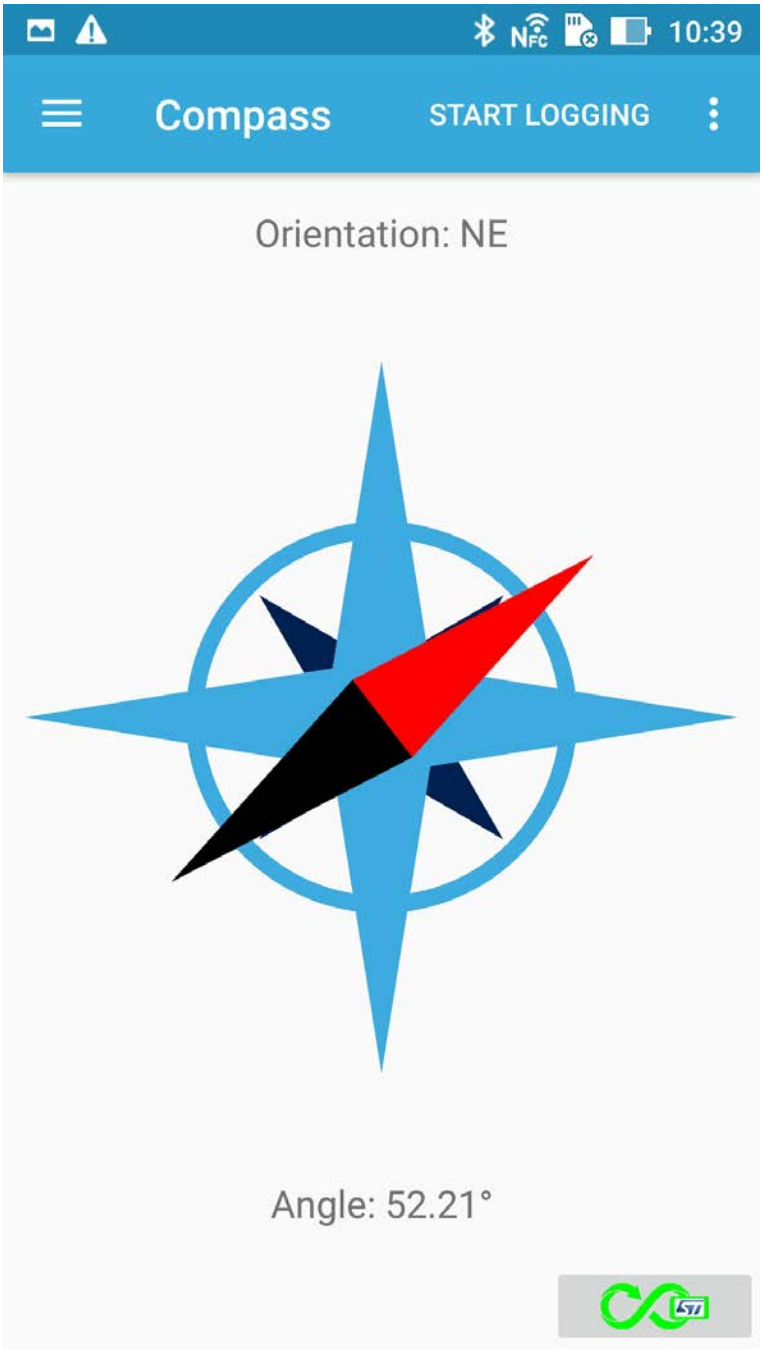
If the MotionID algorithm is enabled, the page shown below is available:

Figure 27. **BlueMS (Android version) Motion Intensity page**



If the MotionFX sensor fusion library is enabled, the following page shows a e-campass that rotates with board movement.

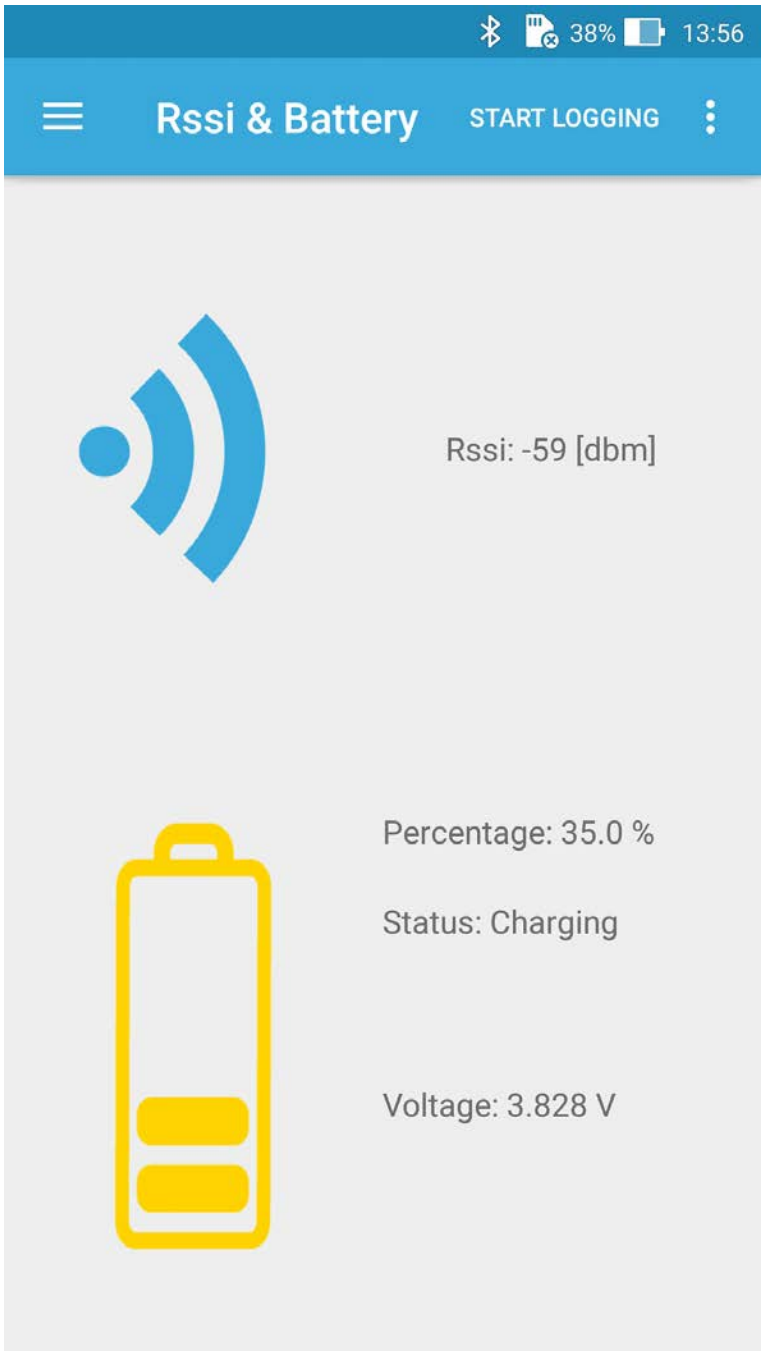
Figure 28. **BlueMS (Android version) e-comapss page**



On this page and on 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.

The last indicates the RSSI of the Bluetooth signal.

Figure 29. **BlueMS (Android version) Battery and RSSI information**

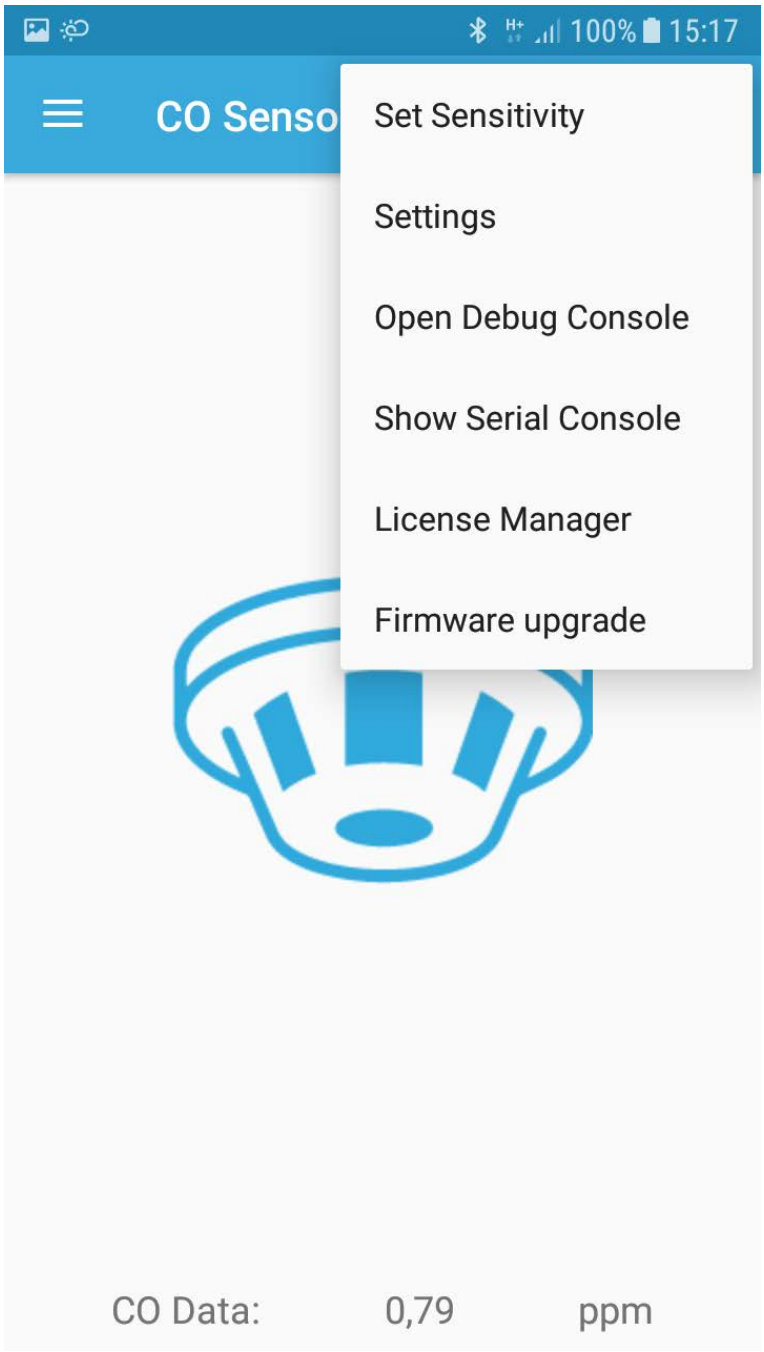


When P-Nucleo-IKA02A1 is in use, this page indicates the output of the CO (Carbon Monoxide) gas sensor in ppm (parts per million), as shown below.

Figure 30. **BlueMS (Android version) CO gas sensor information**

From the contextual menu, it is also possible to set the value of gas sensitivity in nA/ppm.

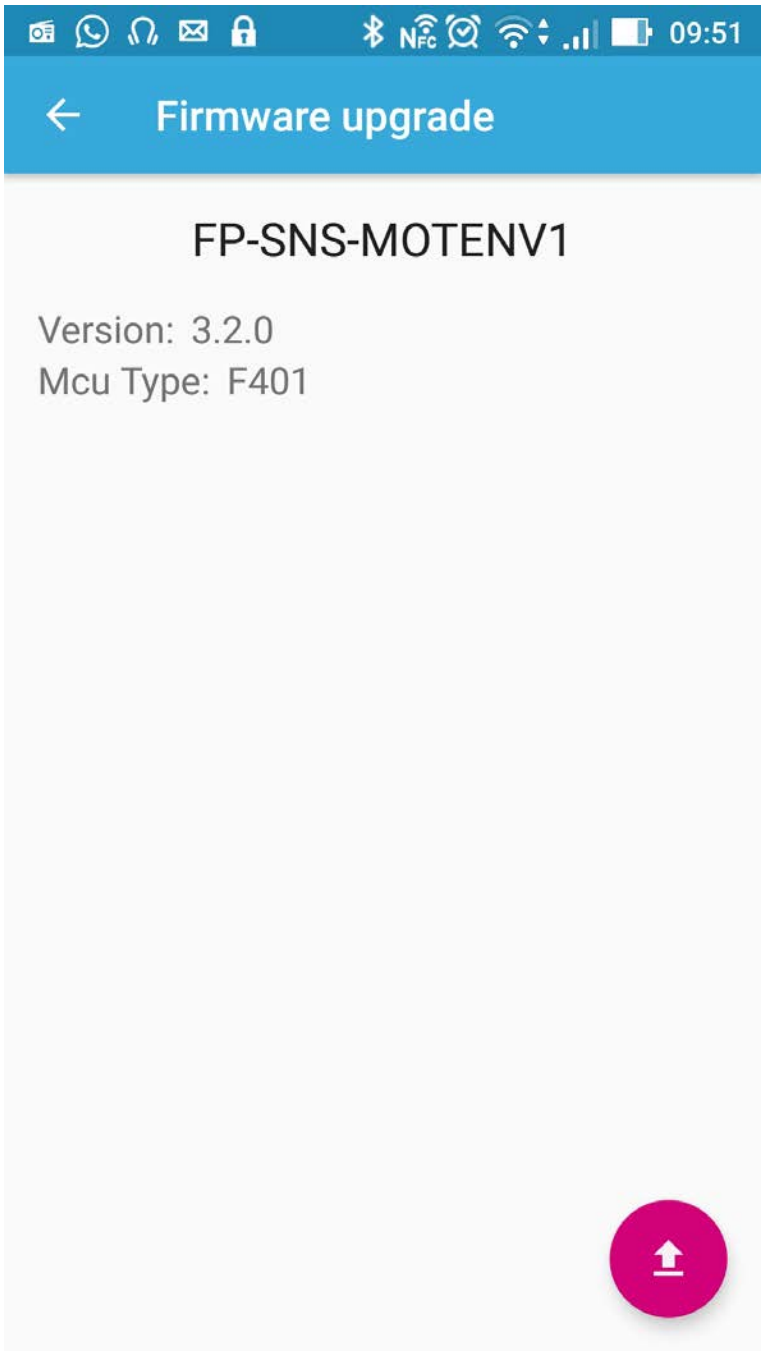
Figure 31. **BlueMS (Android version) CO gas sensor menu**



1.11 Firmware-Over-The-Air update with BlueMS

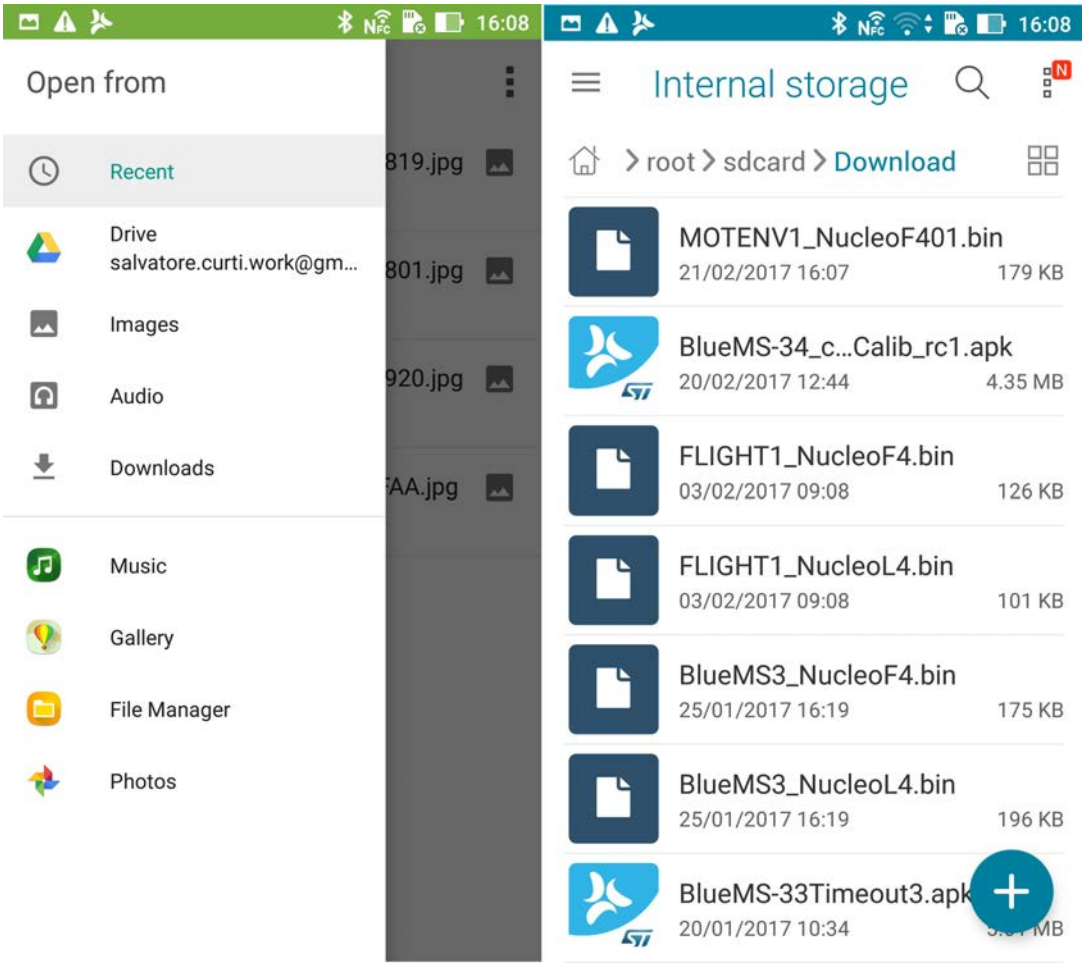
If the 'Firmware upgrade' menu option is selected in the BlueMS main application page (Figure 12. BlueMS (Android version) main page following BLE connection), the page below appears.

Figure 32. **BlueMS (Android version) firmware upgrade page**



The BlueMS 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.

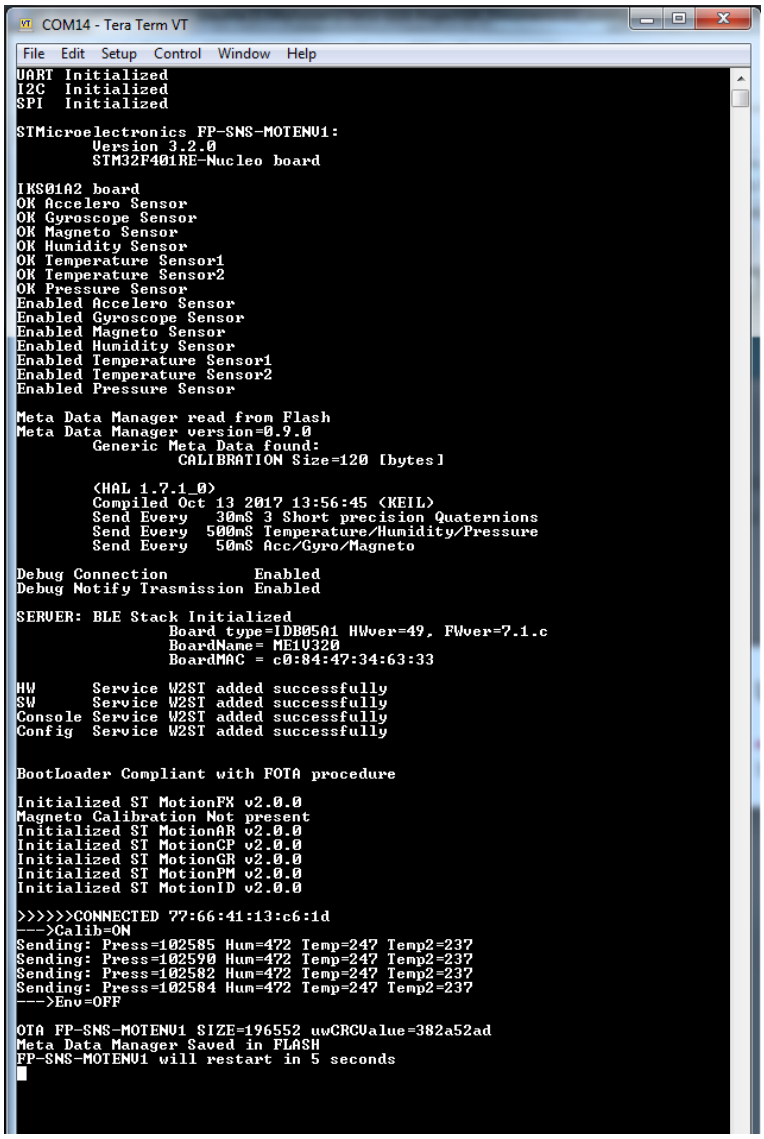
Figure 33. **BlueMS (Android version) firmware update file selection**



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

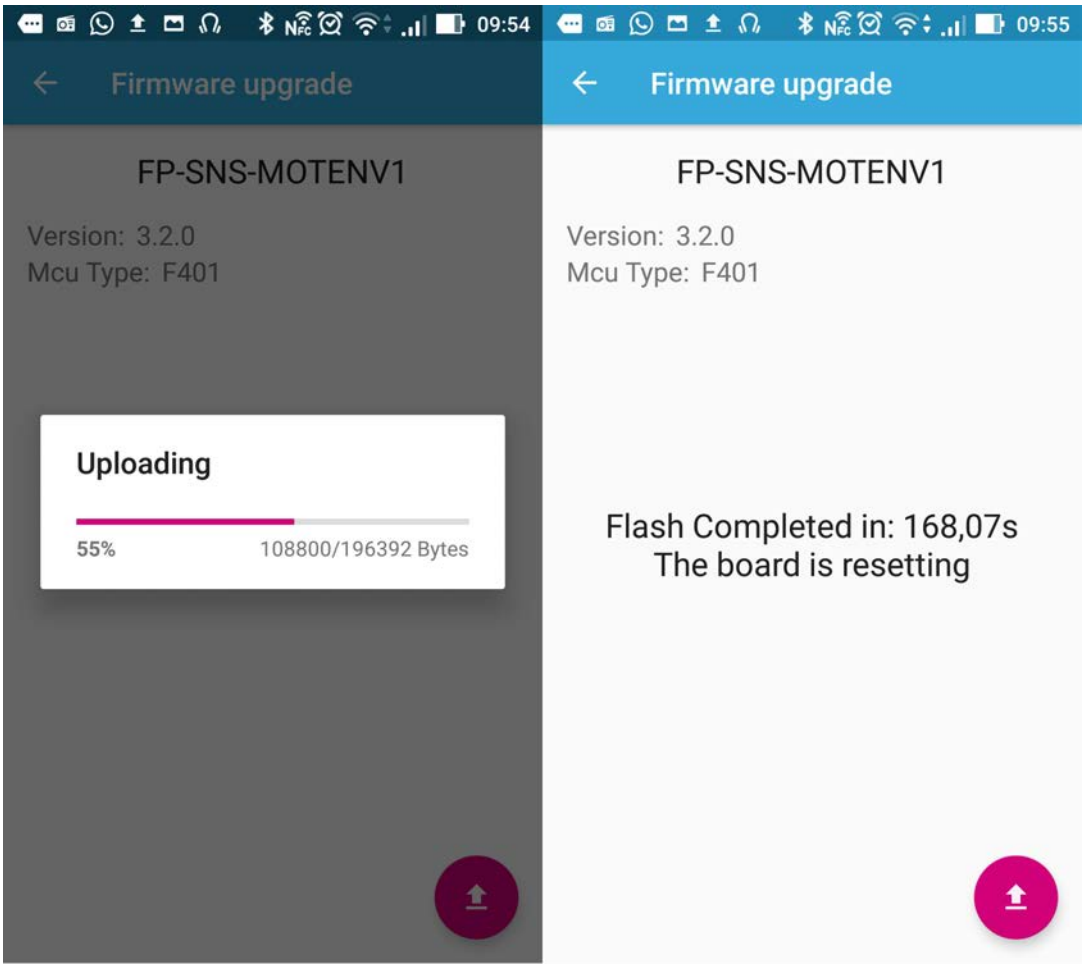
The figure below shows the terminal window with the debug information returned during FOTA for an STM32 Nucleo platform (STM32F401RE/L476RG) when we use a UART to control FP-SNS-MOTENV1 behavior.

Figure 34. **Terminal window information during FOTA**



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

Figure 35. **BlueMS (Android version) application page during FOTA and on completion**



2 System setup guide

2.1 Hardware description

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

2.1.1 STM32 Nucleo platform

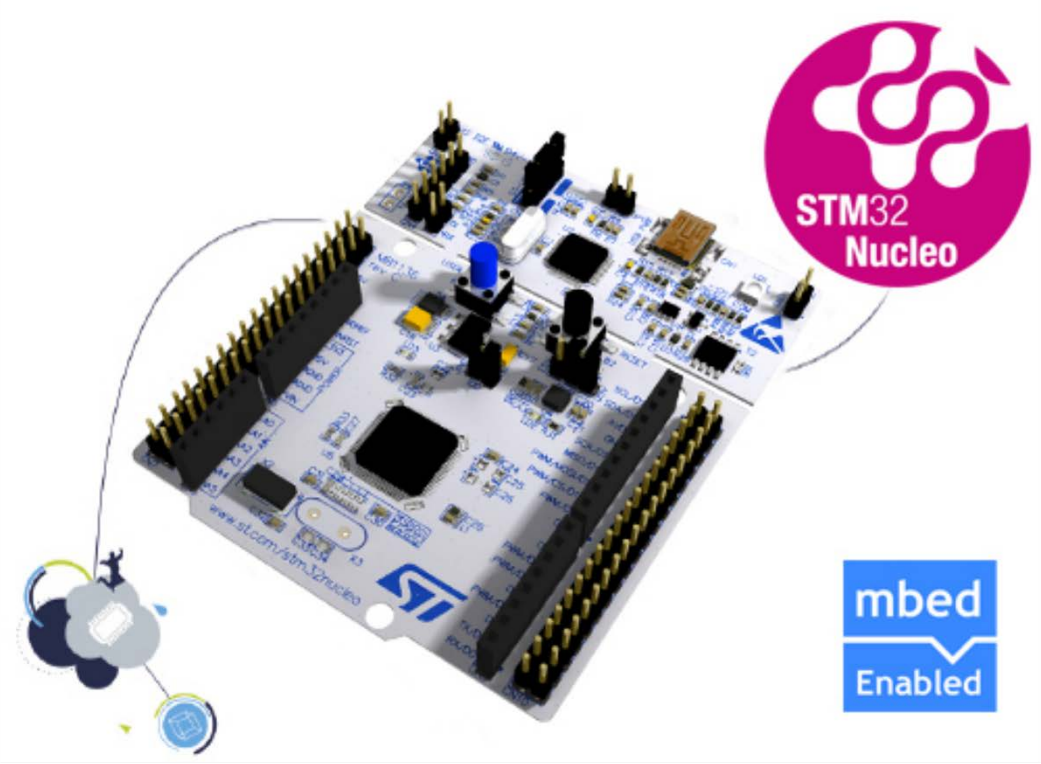
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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

The STM32 Nucleo board does not require separate probes as it integrates the ST-LINK/V2-1 debugger/ programmer.

The STM32 Nucleo board comes with the comprehensive STM32 software HAL library together with various packaged software examples.

Figure 36. STM32 Nucleo board

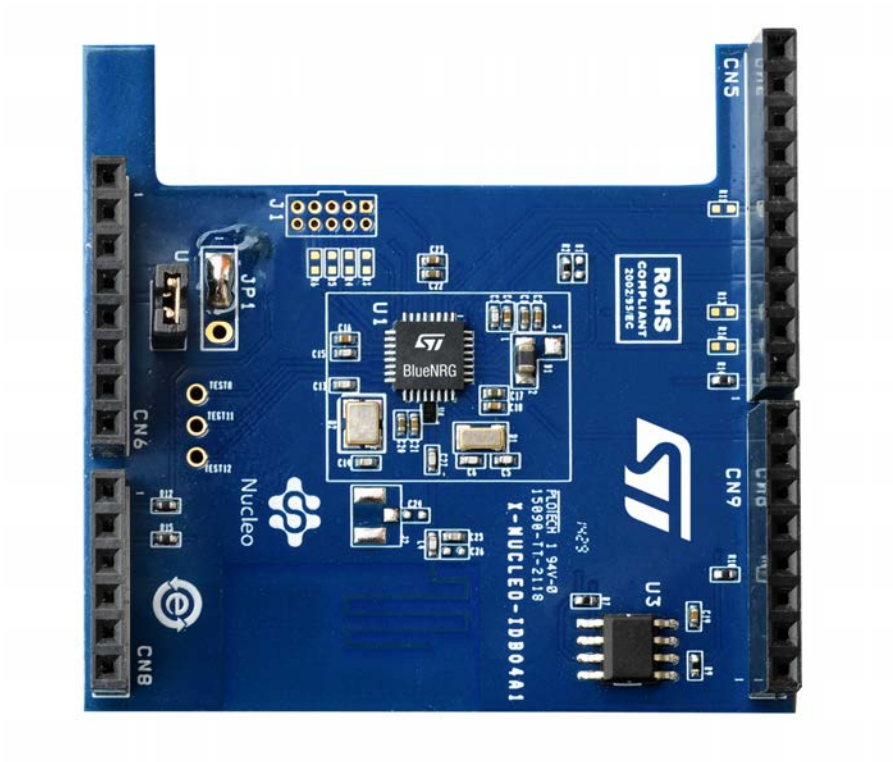


Information regarding the STM32 Nucleo board is available at www.st.com/stm32nucleo

2.1.2 X-NUCLEO-IDB04A1 expansion board

The X-NUCLEO-IDB04A1 is a Bluetooth BlueNRG expansion board usable with the STM32 Nucleo system. The BlueNRG is a very low power Bluetooth low energy (BLE) single-mode network processor, compliant with Bluetooth specifications core 4.0.

Figure 37. **X-NUCLEO-IDB04A1 expansion board**

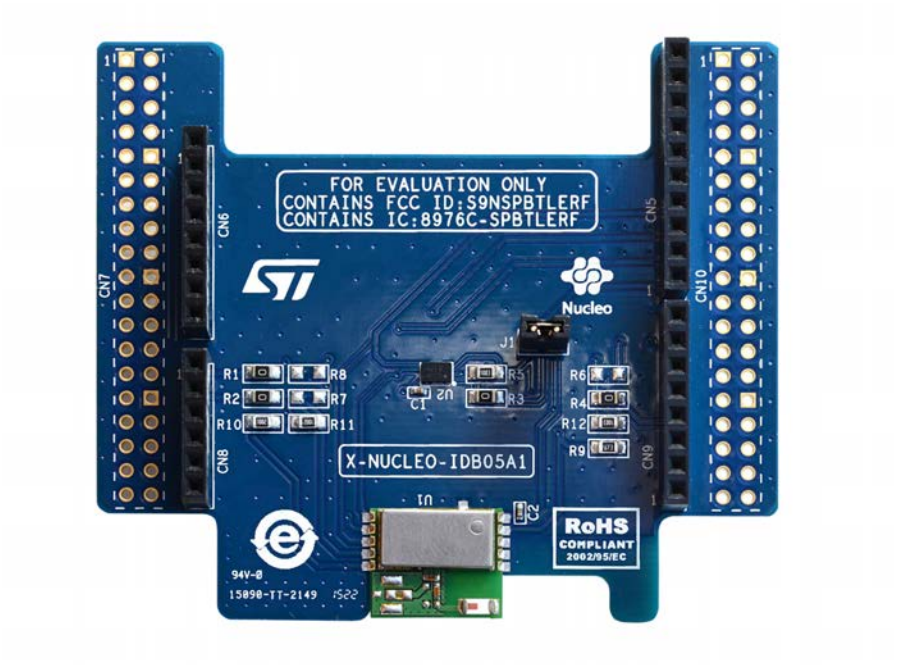


Information regarding the X-NUCLEO-IDB04A1 expansion board is available on www.st.com at http://www.st.com/x-nucleo.

2.1.3 X-NUCLEO-IDB05A1 expansion board

The X-NUCLEO-IDB05A1 is a Bluetooth low energy expansion 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 38. **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

2.1.4 X-NUCLEO-IKS01A1 expansion board

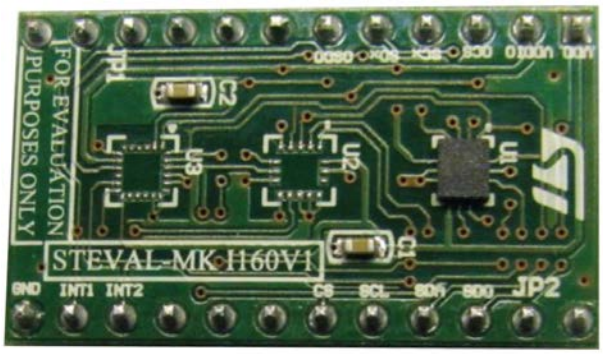
The X-NUCLEO-IKS01A1 figured below is a sensor expansion board for use with the STM32 Nucleo board. It is also compatible with the Arduino UNO R3 connector layout, and is designed around the STMicroelectronics humidity (HTS221), pressure (LPS25HB) and motion sensors (LIS3MDL and LSM6DS0). The X-NUCLEO-IKS01A1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

Figure 39. **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.

The LSM6DS3 DIL24 adapter board in the figure below can be plugged on top of the X-NUCLEO-IKS01A1 expansion board.

Figure 40. **LSM6DS3 DIL24 adapter board**



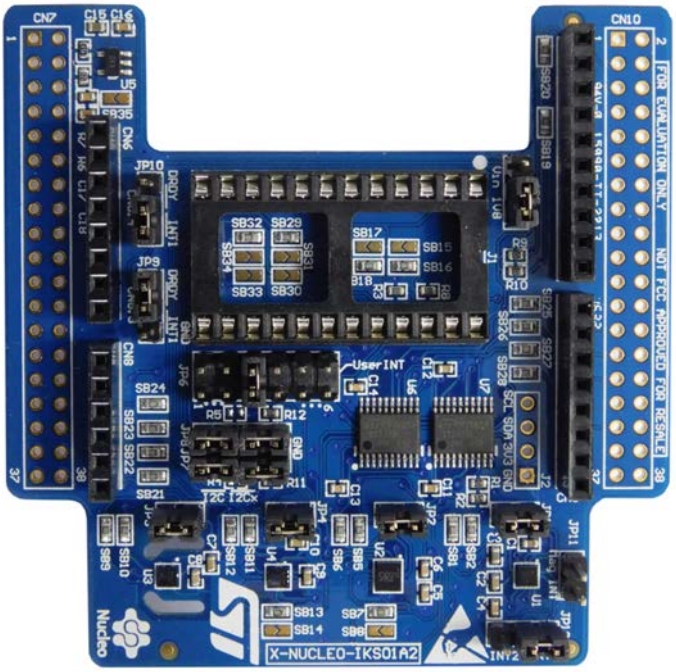
2.1.5 X-NUCLEO-IKS01A2 expansion board

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



2.1.6 P-NUCLEO-IKA02A1 evaluation pack

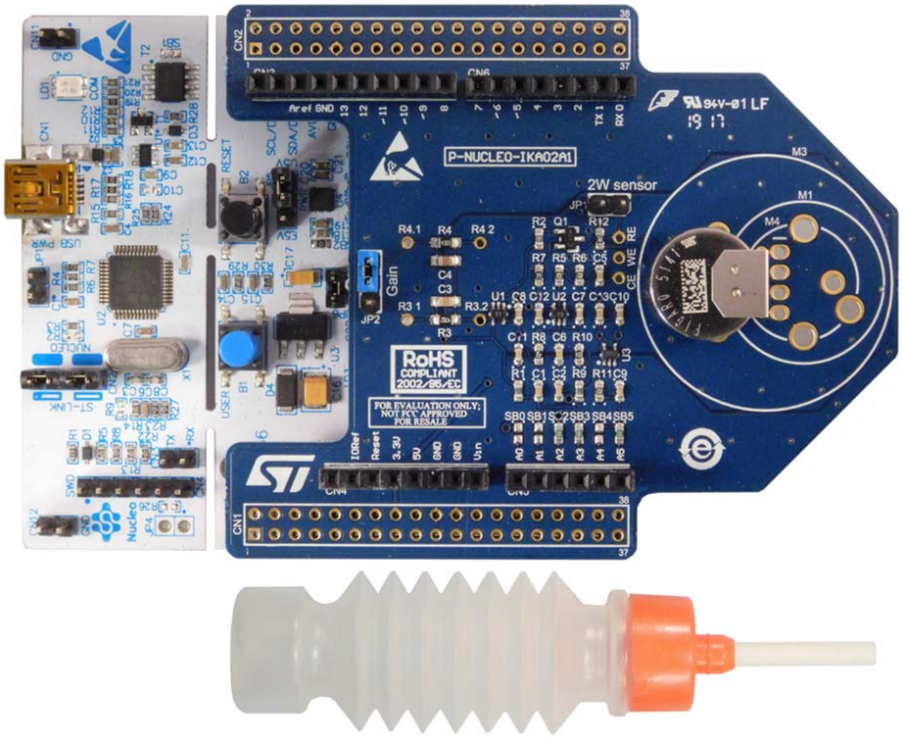
The P-NUCLEO-IKA02A1 evaluation pack provides a reference design for various electrochemical sensors.

The STM32 Nucleo gas expansion board interfaces electrochemical sensors with the MCU on the STM32 Nucleo development board. Two TSU111 operational amplifiers provide signal conditioning; they are ideal for electrochemical sensing thanks to their high precision and low power consumption. The expansion board includes an ultra-low current precision analog temperature sensor STLM20 used for compensation of gas readings.

STM32 Nucleo boards provide an affordable and flexible way for users to experiment with new ideas and build prototypes with any STM32 microcontroller line. The NUCLEO-L053R8 is designed for low power applications.

The design and componentry are optimized for battery operation and maximum battery life time.

Figure 42. **P-NUCLEO-IKA02A1 evaluation pack**



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) toolchain + ST-LINK
  - System Workbench for STM32 + ST-LINK

2.3 Hardware and software setup

This section describes the hardware and software setup procedures. It also describes the system setup needed for the above.

2.3.1 Hardware setup

The following hardware components are needed:

1. one STM32 Nucleo development platform (order code: NUCLEO-F401RE, NUCLEOL476RG or NUCLEO-L053R8)
2. one sensor expansion board (order code: X-NUCLEO-IKS01A2 or X-NUCLEO-IKS01A1)
3. one BlueNRG Bluetooth low energy expansion board (order code: X- NUCLEO-IDB05A1 or X-NUCLEO-IDB04A1)
4. one USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC

2.3.2 Software setup

This section lists the minimum requirements for the developer to set up the SDK, run the sample testing scenario based on the GUI utility and customize applications.

2.3.2.1 *Development tool-chains and compilers*

Please select 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.

The projects for NUCLEO-F401RE and NUCLEO-L476RG provide two different targets, one supporting the P-NUCLEO-IKA02A1 evaluation pack and one without it.

2.3.3 System setup guide

This section describes how to set up different hardware sections before writing and executing an application on the STM32 Nucleo board with the sensors expansion board.

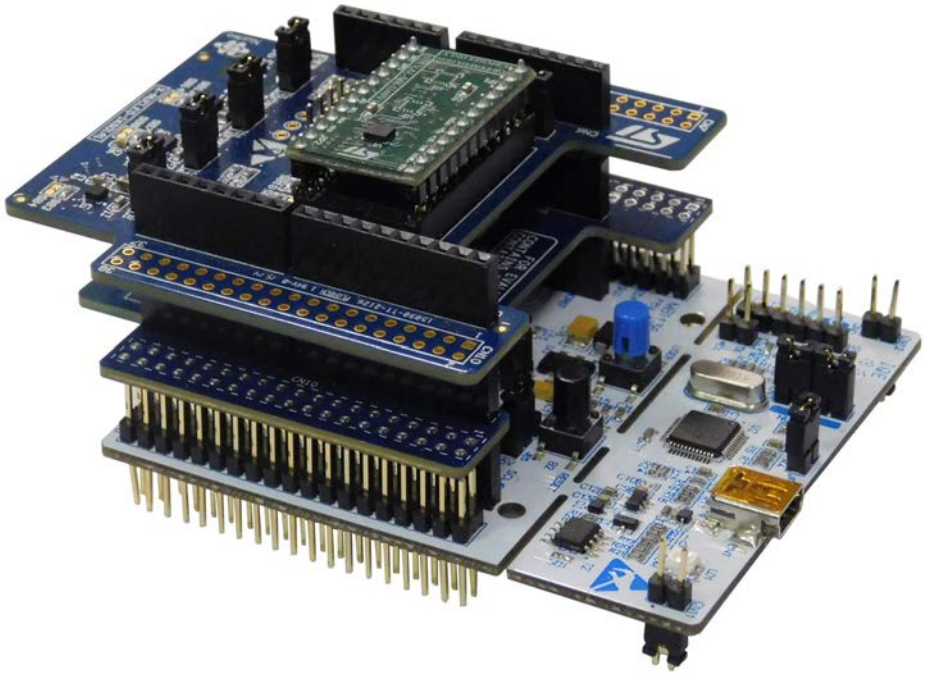
2.3.3.1 *STM32 Nucleo and sensor expansion board setup*

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 BlueNRG BLE expansion board X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1) is easily connected to the STM32 Nucleo board through the Arduino UNO R3 extension connector.

Finally the sensors board X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A1) is easily connected to the X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1) expansion board through the Arduino UNO R3 extension connector, as per the figure below.

Figure 43. X-NUCLEO-IKS01A1 sensors board connected to STM32 Nucleo Board over the X-NUCLEO-IDB05A1 expansion board



Note: To optimize the performance of the SPBTLE-RF module present on X-NUCLEO-IDB05A1 expansion board and to reduce antenna interference, stack the boards in the sequence shown above.

Optionally, a P-NUCLEO-IKA02A1 expansion board can be also connected on top of the X-NUCLEO-IKS01A2. To operate correctly, it is necessary to remove the following solder bridges on the X-NUCLEO-IKS01A2: SB21,

SB22, SB23 and SB24 (see Figure 45. Solder bridges to remove on the X-NUCLEO-IKS01A2 to use it with the P-NUCLEO-IKA02A1).

Note: The P-NUCLEO-IKA02A1 board is not compatible with the X-NUCLEO-IKS01A1.

Figure 44. Boards connected with P-NUCLEO-IKA02A1 added on top

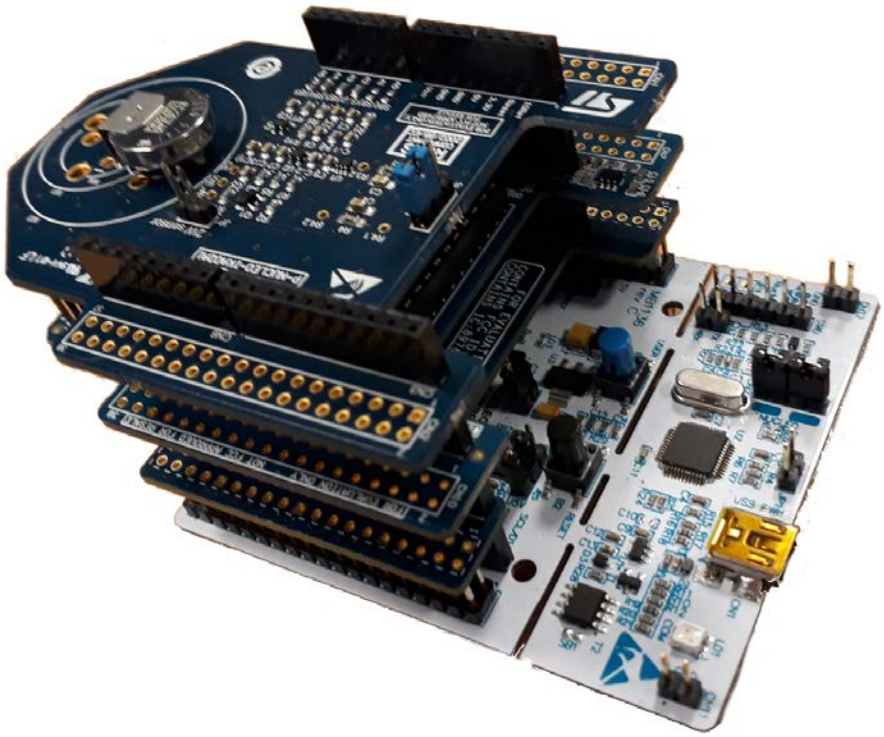
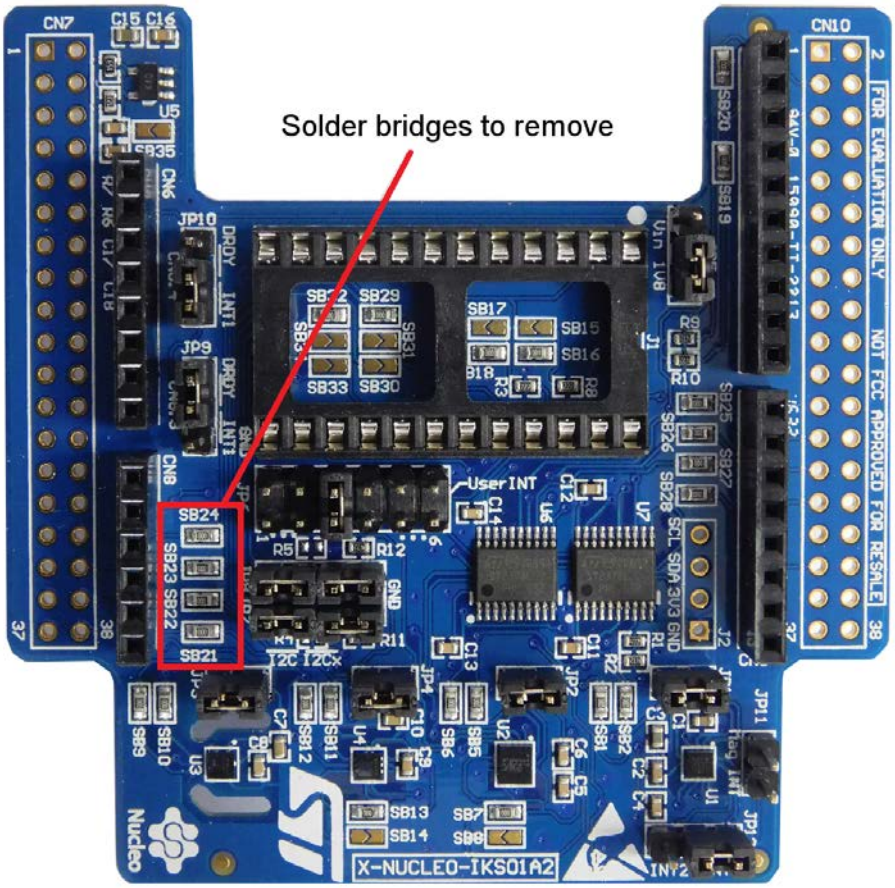


Figure 45. **Solder bridges to remove on the X-NUCLEO-IKS01A2 to use it with the P-NUCLEO-IKA02A1**



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	Updated Section ● Introduction, 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. 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.

Contents

List of figures