UM2059

User manual

Getting started with the STM32Cube function pack for IoT node with Bluetooth® Low Energy connectivity, digital microphone, environmental, and motion sensors

Introduction

FP-SNS-ALLMEMS1 is an STM32Cube function pack, which lets you connect your IoT node to a smartphone via Bluetooth® Low Energy and use a suitable Android™ or iOS™ application, like the STBLESensor or STBLESensClassic app, to view real-time environmental sensor data, motion sensor data, digital microphone levels, and battery level.

The package also enables advanced functions such as voice communication over Bluetooth® Low Energy, sound source localization, and acoustic beam forming using inputs from multiple microphones, as well as sensor data fusion and accelerometer-based real-time activity recognition, carry position, gesture recognition, motion intensity recognition and audio data logging.

Moreover, it provides real-time information about the user current pose based on data from a device, working mode (sitting/standing desk position), the device tilt angles, the repetitions of various fitness activities performed and the vertical movement.

This package, together with the suggested combination of STM32 and ST devices, can be used to develop specific wearable 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, as well as on the STEVAL-BCNKT01V1 and STEVAL-MKSBOX1V1.

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

RELATED LINKS

Visit the STM32Cube ecosystem web page on www.st.com for further information

1 FP-SNS-ALLMEMS1 software description

1.1 Overview

The key features of the FP-SNS-ALLMEMS1 package are:

- Complete firmware to develop an IoT node with Bluetooth® Low Energy connectivity, digital microphone, environmental, and motion sensors
- Middleware libraries for sensor data fusion, accelerometer-based real-time activity recognition, carry position, gesture recognition, motion intensity recognition, user current pose recognition, working mode recognition, tilt angles evaluation, vertical movement detection, fitness activities quantity repetition, acoustic source localization and beam forming, audio processing and streaming over Bluetooth® Low Energy communication profile, SD card data logging
- Compatible with STBLESensor or STBLESensClassic application for Android/iOS, to perform sensor data reading, audio and motion algorithm feature demo, and firmware update over the air (FOTA)
- Sample implementation available for STEVAL-BCNKT01V1 and STEVAL-MKSBOX1V1 boards and for X-NUCLEO-CCA02M2, X-NUCLEO-IKS4A1 (or X-NUCLEO-IKS01A3) and X-NUCLEO-BNRG2A1 connected to a NUCLEO-F446RE or NUCLEO-L476RG boards
- 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 and software characteristics:
   - HW characteristics related to MEMS sensor devices:
     - Temperature
     - Pressure
     - Humidity
     - 3-axis gyroscope, 3-axis magnetometer, 3-axis accelerometer
     - Microphones dB noise level.
     - Battery %, voltage, and status (charging/discharging/low battery) for STEVAL-MKSBOX1V1 and STEVAL-BCNKT01V1
   - SW characteristics:
     - quaternions generated by the MotionFX library in short precision
     - magnetic North direction (e-Compass)
     - recognized activity using the MotionAR algorithm
     - features not available for STEVAL-BCNKT01V1 and NUCLEO-F446RE:
       - recognized carry position using the MotionCP algorithm
       - count of repetitions for various fitness activities performed by a user using the MotionFA algorithm
       - recognized gesture using the MotionGR algorithm
       - recognized motion intensity using the MotionID algorithm
       - recognized user current pose using the MotionPE algorithm
       - recognized user working mode using the MotionSD algorithm
       - tilt angles evaluation of the user device using the MotionTL algorithm
       - recognized type of vertical movement using the MotionVC algorithm
     - audio source localization using the AcousticSL algorithm (feature not available on the STEVAL-MKSBOX1V1)
     - audio beam forming using the AcousticBF algorithm (feature available only on STEVAL-BCNKT01V1)
     - voice over Bluetooth® Low Energy using the BlueVoiceADPCM algorithm (feature available on STEVAL-BCNKT01V1)
2. The second 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
3. The last service is for transmitting/resetting the calibration status and enabling the following expansion hardware features for LSM6DSV16X on an X-NUCLEO-IKS4A1 expansion board for STM32 Nucleo L4 only, for LSM6DSO on an X-NUCLEO-IKS01A3 expansion board for STM32 Nucleo L4 only, for LSM6DSM on STEVAL-BCNKT01V1 or for LSM6DSOX for STEVAL-MKSBOX1V1:
   - Pedometer (not available on STEVAL-MKSBOX1V1)
   - 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) - not available on STEVAL-MKSBOX1V1

This software gathers:

- the temperature, humidity, pressure, audio, and motion sensor drivers for the STTS22H, SHT40AD1B, LPS22DF, MP34DT01-M and LSM6DSV16X devices if X-NUCLEO-IKS4A1 expansion board is mounted on STM32 Nucleo
- the temperature, humidity, pressure, audio, and motion sensor drivers for the HTS221, LPS22HH, MP34DT01-M, LSM6DSO and LIS2MDL devices if X-NUCLEO-IKS01A3 expansion board is mounted on STM32 Nucleo
- the temperature, pressure, audio, and motion sensor for the LPS22HB, MP34DT04-C1, and LSM6DSM, LSM303AGR devices for STEVAL-BCNKT01V1 running on STM32 Nucleo.
- the temperature, pressure, humidity, audio, and motion sensor for the HTS221, LPS22HH, MP23ABS1, and LSM6DSOX, LIS2MDL devices for STEVAL-MKSBOX1V1 running on STM32 Nucleo.

This package is compatible with the STBLESensor and STBLESensClassic Android/iOS application, available at the respective Play/iTunes stores, which can be used for displaying information sent via the Bluetooth® Low Energy protocol and for Over-The-Air firmware updates.

1.2 Architecture

The STM32Cube function packs leverage the modularity and interoperability of STM32 Nucleo and expansion boards with STM32Cube and expansion software to create functional examples representing some of the most common use cases in each sphere of application.

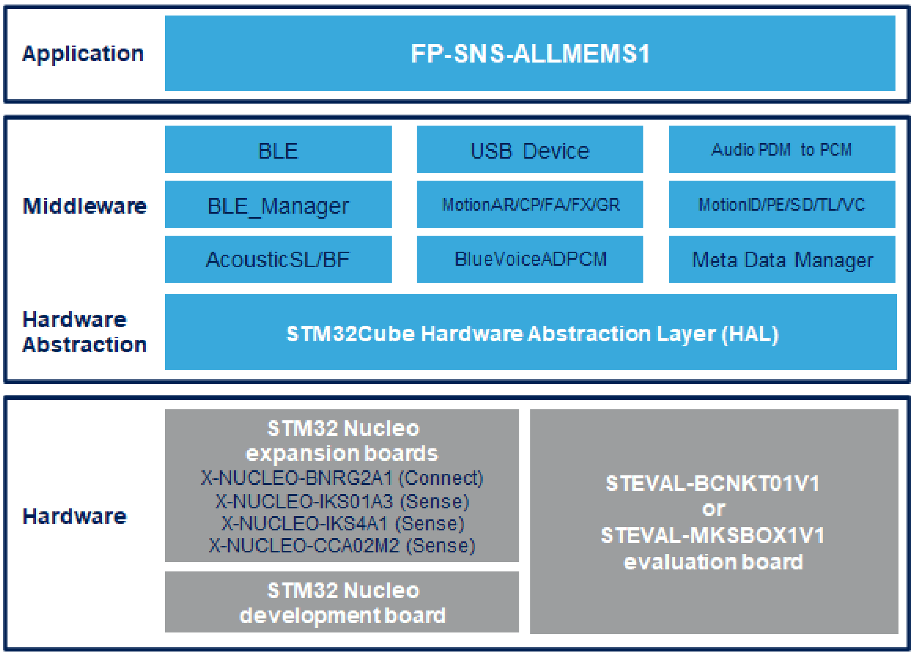
These software function packs are designed to fully exploit the underlying STM32 ODE hardware and software components to best satisfy the final user application requirements.

Function packs may also include additional libraries and frameworks not present in the original expansion software packages, thus enabling new functions and creating more pertinent and usable systems for developers.

To access and use the sensor expansion board, the application software uses:

- **STM32Cube HAL layer**: provides a simple, generic and multi-instance set of generic and extension APIs (application programming interfaces) to interact with the upper layer application, libraries and stacks. It is directly based on a generic architecture and allows the layers that are built on it, such as the middleware layer, to implement their functions without requiring the specific hardware configuration for a given microcontroller unit (MCU). This structure improves library code reusability and guarantees easy portability across other devices.
- **Board support package (BSP) layer**: supports the peripherals on the STM32 Nucleo board (except the MCU) with a limited set of APIs providing a programming interface for certain board-specific peripherals like the LED, the user button, etc., and helps determine the specific board version. For the sensor expansion board, it provides the programming interface for various inertial and environmental sensors and support for initializing and reading sensor data.

Figure 1. FP-SNS-ALLMEMS1 software architecture



This software is based on the STM32CubeHAL. It extends STM32Cube by providing a board support package (BSP) for the BlueNRG-MS and BlueNRG-2, sensor expansion board, and middleware components for communication with other Bluetooth® Low Energy devices, for sensor data fusion, real-time audio library and voice communication over Bluetooth® Low Energy.

The package contains audio libraries (AcousticBF, AcousticSL, and BlueVoiceADPCM) and motion sensor libraries (MotionAR, MotionCP, MotionFA, FusionFX, MotionGR, MotionID, MotionPE, MotionSD, MotionTL, MotionVC) useful for sensing applications based on Bluetooth® Low Energy communication.

The motion algorithms are managed through special software designed for mobile and wearable applications and are strictly limited to work with accelerometer and pressure data only, to facilitate low power consumption strategies commonly required in these applications, in compliance with Bluetooth specifications.

The provided drivers abstract low-level hardware details, so middleware components and applications can access the sensors in a hardware-independent manner.

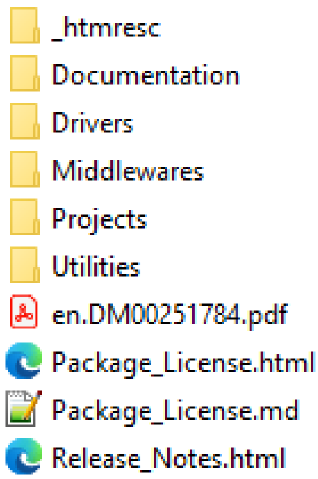
The package includes a sample application to transmit the values read from all the sensors to a Bluetooth low energy-enabled device such as an Android™ or iOS™.

The STBLESensor or STBLESensClassic Android/iOS application, available on the respective application stores, displays the values read from the sensors. The application also allows firmware update over the air as well as displaying battery information.

The STEVAL-MKSBOX1V1 board goes in shutdown mode if it is not connected to an Android/iOS device for a period longer than a fixed range time.

1.3 Folder structure

Figure 2. **FP-SNS-ALLMEMS1 package folder structure**



The following folders are included in the software package:

- **Documentation**: contains a compiled HTML file generated from the source code, which details the software components and APIs.
- **Drivers**: contains the HAL drivers, 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 Cortex-M processor series.
- **Middlewares**: contains libraries and protocols for BlueNRG Bluetooth low energy, USB Device Library, PDM signal decoding and audio signal reconstruction when connecting an ST MEMS, the Meta Data Manager, MotionAR activity-recognition library, MotionCP carry-position recognition library, MotionFA fitness activity library, MotionFX (iNEMOEngine PRO) sensor fusion library, MotionGR gesture recognition library, MotionID motion intensity detection library, MotionPE real-time pose estimation library, MotionSD standing vs sitting desk detection library, MotionTL tilt measurement library, MotionVC vertical context library, AcousticSL sound source localization library, AcousticBF beam forming and BlueVoiceADPCM half-duplex voice-over-Bluetooth low energy communication profile.
- **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, MotionCP carry-position, MotionGR gesture recognition, AcousticSL sound source localization, AcousticBF beam forming and BlueVoiceADPCM over Bluetooth low energy libraries by using the Bluetooth low energy protocol provided for the NUCLEO-F446RE/NUCLEO-L476RG, STEVAL-BCNKT01V1 and STEVAL-MKSBOX1V1 platforms through the IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM) and Integrated Development Environment for STM32 (STM32CubeIDE).
- **Utilities**: contains the boot loader binary ready to be flashed for the STM32F446RE, STM32L4R9ZI and STM32L476RG Nucleo boards.

1.4 Flash management

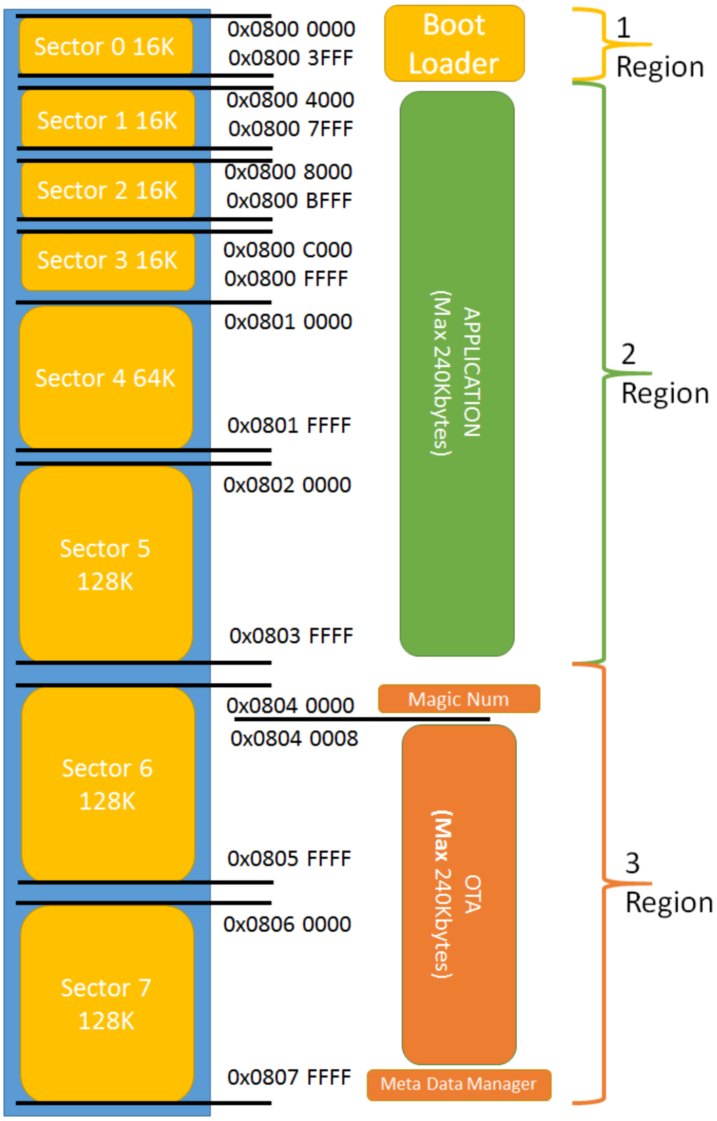
Apart from storing code, FP-SNS-ALLMEMS1 uses the flash memory for Firmware-Over-The-Air updates. It is divided into the following regions (see figure below):

1. the first region contains a custom boot loader
2. the second region contains the FP-SNS-ALLMEMS1 firmware
3. The third region is used for storing the FOTA before the update

Even if the STM32F446RE (512 KB), the STM32L476RG (1024 KB), STM32L4R9ZI (2048 KB) cache sizes and arrangements differ, we have used the same FLASH arrangement for both. The Meta Data Manager is placed at the end of the flash (0x08007000 for STM32F446RE, 0x080FF000 for STM32L476RG and 0x081FF000 for STM32L4R9ZI). For more information, refer to:

- *RM0390* Reference manual STM32F446xx advanced ARM®-based 32-bit MCUs
- *RM0351* Reference manual STM32L4x6 advanced ARM®-based 32-bit MCUs
- *RM0432* Reference manual STM32L4Rxxx advanced ARM®-based 32-bit MCUs

Figure 3. **FP-SNS-ALLMEMS1 flash structure (STM32F446)**



1.5 The boot process

The FP-SNS-ALLMEMS1 cannot not be flashed at the beginning of the flash (address 0x08000000), and is therefore compiled to run from the beginning of the second flash region, at 0x08004000.

To enable this behavior, we set the vector table offset in Src/system_stm32f4xx.c (for STM32F446) and Src/system_stm32l4xx.c (for STM32L476 and STM32R9ZI) thus: `#define VECT_TAB_OFFSET 0x4000`.

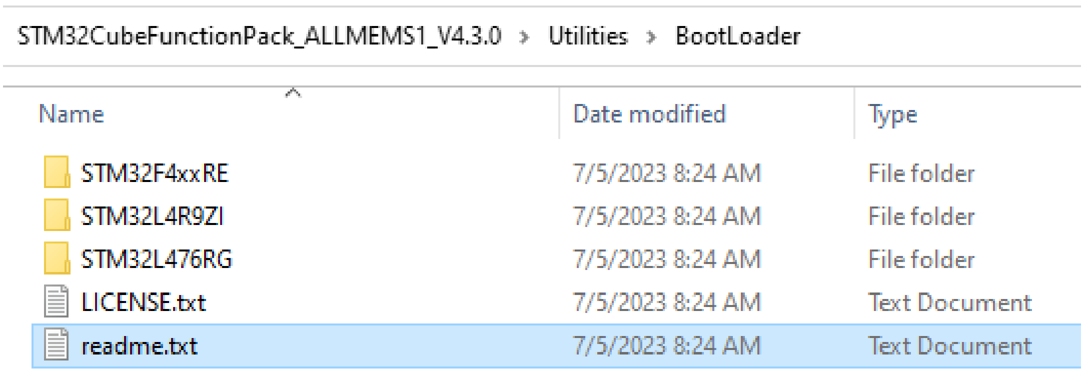
We also changed the linker script. For example, the Linker script for FP-SNS-ALLMEMS1 running on STM32F401RE and compiled using IAR Embedded Workbench for ARM is:

```
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 KB.

You must flash the appropriate bootloader binary for STM32F446RE, STM32L476RG or STM32L4R9ZI, found in the Utilities\BootLoader folder, to the first flash region (address 0x08000000).

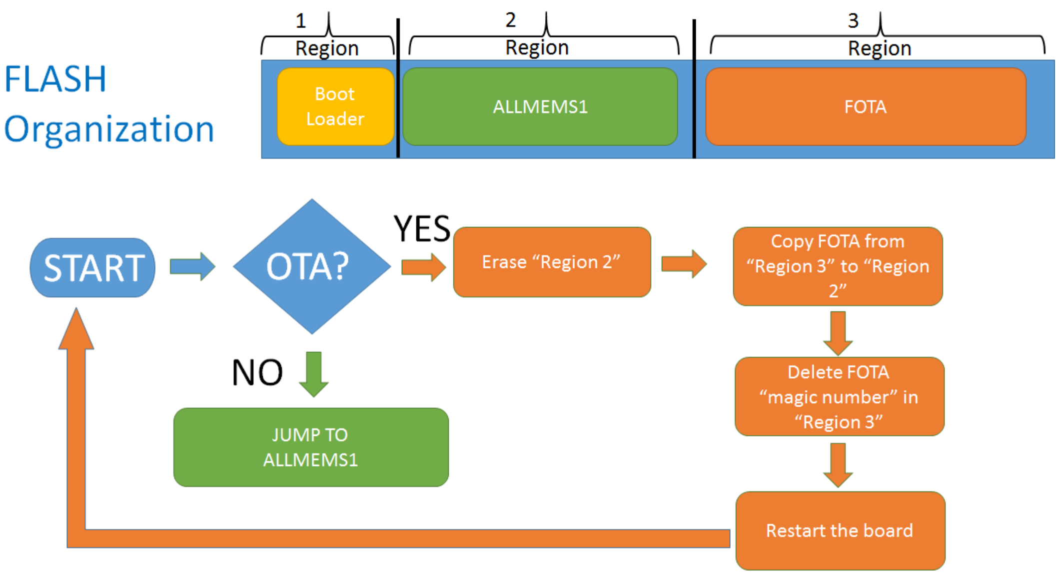
Figure 4. BootLoader folder content



On any board reset:

- If there is a FOTA in the third Flash region, the boot loader overwrites the second Flash region (with FP-SNS-ALLMEMS1 firmware) and replaces its content with the FOTA and restarts the board.
- If there is no FOTA, the boot loader jumps to the FP-SNS-ALLMEMS1 firmware.

Figure 5. FP-SNS-ALLMEMS1 boot sequence



1.6 The installation process

For each platform (NUCLEO-F446RE, NUCLEO-L476RG, STEVAL-BCNKT01V1, STEVAL-MKSBOX1V1), there is a package binary directory that contains:

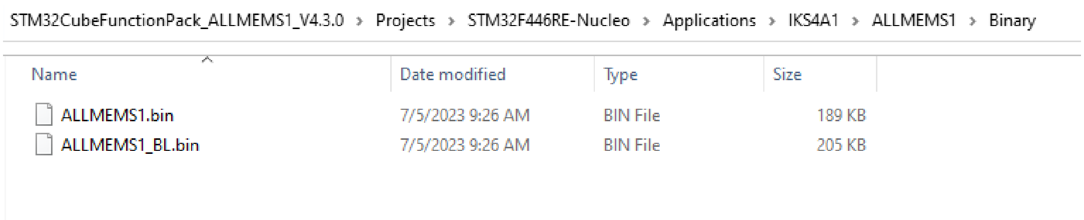
- precompiled ALLMEMS1 firmware that may be flashed with ST-LINK to the correct memory address (0x08004000) of a supported STM32 Nucleo development board or evaluation boards

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

- precompiled ALLMEMS1 plus bootloader firmware that may be directly flashed to a supported STM32 Nucleo development board or evaluation board with the ST-LINK or via a drag & drop operation (STM32 Nucleo development boards only)

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

Figure 6. **Binary folder content for NUCLEO-F446RE**



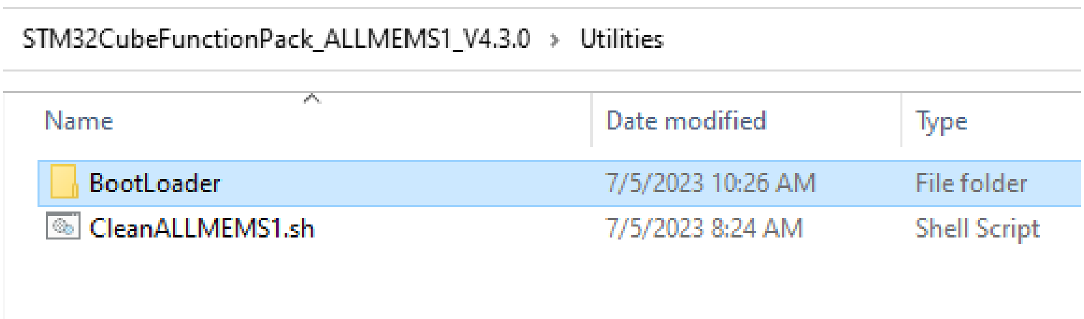
In the folder Utilities there is a scripts *.sh that makes the following operations:

- 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 7. **Utilities folder content**



This script:

- performs a full flash memory erase to start from a clean system
- flashes the bootloader to the correct position (0x08000000)
- flashes the firmware to the correct position (0x08004000)

Figure 8. **Bootloader and ALLMEMS1 installation**

1.7 Firmware-Over-The-Air (FOTA) update

The FP-SNS-ALLMEMS1 firmware can be updated Over-The-Air (FOTA) through an Android/iOS device via Bluetooth using the STBLESensor (or STBLESensClassic) application (ver. 4.20.0 and above) available on their respective application market stores.

The application sends the update and associated CRC (cyclic-redundancy-check) value, which the FP-SNS-ALLMEMS1 checks against the hardware cyclic redundancy check calculation unit on the STM32F446RE/STM32L476RG/STM32L4R9ZI processor to ensure integrity. If the CRC calculation matches the STBLESensor (or STBLESensClassic) CRC value, the new firmware is written to the beginning of the third Flash region. A “magic number” setting signals the boot loader that a Firmware update has been received and checked, and is ready to replace the current FP-SNS-ALLMEMS1 firmware.

1.8 APIs

Detailed user-API technical information with full function and parameter descriptions is available in a compiled HTML file in the package “Documentation” folder.

1.9 Sample application description

A sample application using:

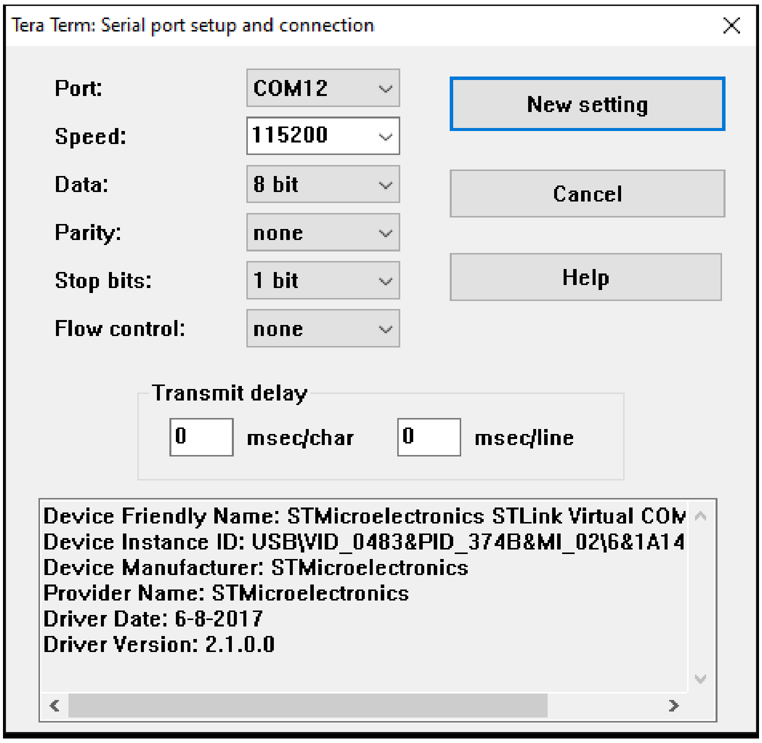
- the X-NUCLEO-IKS01A3 or X-NUCLEO-IKS4A1, X-NUCLEO-CCA02M2 and X-NUCLEO-BNRG2A1 expansion boards with the NUCLEO-F446RE or NUCLEO-L476RG board
- the STEVAL-BCNKT01V1 evaluation board
- the STEVAL-MKSBOX1V1 evaluation board

Ready to build projects are available for multiple IDEs.

With the NUCLEO-F446RE and NUCLEO-L476RG boards, you can set up a terminal window for the appropriate UART communication port (as per Figure 9. Terminal setting) to control the initialization phase.

When connecting the micro-USB port to a PC, with STEVAL-MKSBOX1V1 evaluation boards, you can set up a terminal window for the appropriate UART communication port (as per Figure 9. Terminal setting) to control the initialization phase. However, as it is necessary to register the USB device, this is only possible when the STEVAL-MKSBOX1V1 starts initializing. In fact, a 5-second (10-second) delay has been added to the initialization phase to allow the user to monitor its progress.

Figure 9. Terminal setting



When you first press the reset button, the application:

- starts initializing the UART interface
- writes whether the firmware is compiled for X-NUCLEO-IKS01A3 or X-NUCLEO-IKS4A1
- checks whether all the sensors are present and working
- determines if the BlueNRG expansion board (X-NUCLEO-BNRG2A1) is connected to the STM32 Nucleo board as well as the hardware and firmware version information
- shows a random BLE MAC address
- initializes the BLE hardware service (adding the temperature, humidity, pressure, 3D gyroscope, 3D magnetometer, 3D accelerometer and microphone characteristics) and the BLE software service (adding the MotionFX, MotionAR, MotionCP, MotionFA, MotionGR, MotionID, MotionPE, MotionSD, MotionVC, AcousticSL, AcousticBF and BlueVoiceADPCM)
- initializes the BLE console service adding the stdin/stdout and stderr characteristics
- initializes the BLE configuration service to enable the hardware features:
  - for LSM6DSV16X mounted on the X-NUCLEO-IKS4A1 expansion board (for Nucleo L4 only)
  - for LSM6DSO mounted on the X-NUCLEO-IKS01A3 expansion board (for Nucleo L4 only)

If the X-NUCLEO-IKS01A3 or X-NUCLEO-IKS4A1 is mounted and Nucleo L4 only, it can generate an interrupt due to free fall, tilt, wake up, single tap, double tap, 6D position or pedometer events, transmitted over Bluetooth to the connected Android™/iOS™ device.

Figure 10. Initialization phase

```
COM14 - Tera Term VT
File  Edit  Setup  Control  Window  Help
UART Initialized

STMicroelectronics FP-SNS-ALLMEMS1:
        Version 4.3.0
        STM32L476RG-Nucleo board
                MCU clock set at 80 MHz

Code compiled for X-NUCLEO-IKS4A1 board
        OK Accelero Sensor
        OK Gyroscope Sensor
        OK Magneto Sensor
        OK Temperature Sensor
        OK Humidity Sensor
        OK Temperature and Pressure
        Enabled Accelero Sensor
        Enabled Gyroscope Sensor
        Enabled Magneto Sensor
        Enabled Temperature Sensor
        Enabled Humidity Sensor
        Enabled Temperature Sensor
        Enabled Pressure Sensor

Meta Data Manager read from Flash (Address: 0x80ff000)

        (HAL 1.13.4_0)
        Compiled Jul  5 2023 08:59:45 (IAR)
        Send Every   30mS 3 Short precision Quaternions
        Send Every  500mS Temperature/Humidity/Pressure
        Send Every   50mS Acc/Gyro/Magneto
        Send Every   50mS dB noise

Debug Connection         Enabled
Debug Notify Trasmission Enabled


Node name read from FLASH (AM1V430)

SERVER: BLE Stack Initialized
                BoardName= AM1V430
                BoardMAC = f5:b4:5d:de:fa:3f
                BlueNRG-2 HW ver1.2
                BlueNRG-2 FW ver2.1.b

BlueST-SDK V2
Config  Service added successfully
Console Service added successfully
Warning: Read request HW Accelerometer Event function not defined
BLE HW Accelerometer Event features ok
BLE Audio Level features ok
BLE Environmental features ok
BLE Inertial features ok
Warning: Read request Activity Recognition function not defined
BLE Activity Recognition features ok
BLE Audio Source Localization features ok
BLE E-Compass features ok
BLE Fitness Activities features ok
Warning: Read request Gesture Recognition function not defined
BLE Gesture Recognition features ok
BLE Motion Algorithms features ok
Warning: Read request Motion Intensity function not defined
BLE Motion Intensity features ok
BLE Sensor Fusion features ok
Warning: Read request Tilt Sensing function not defined
BLE Tilt Sensing features ok
Features Service added successfully (Status= 0x0)

Testing BootLoaderCompliance:
        Version  2.0.0
        BL Version  OK
        MagicNum    OK
        MaxSize =7c000
        OTAStartAdd OK
        OTADoneAdd  OK
BootLoader Compliant with FOTA procedure

Initialized ST MotionFX v2.7.1
Magneto Calibration Not present
Initialized ST MotionAR v3.2.1
Initialized ST MotionFA v1.3.1
Initialized ST MotionGR v2.2.1
Initialized ST MotionID v2.4.1
Initialized ST MotionPE v2.2.1
Initialized ST MotionSD v1.2.1
Initialized ST MotionTL v1.5.0
            --> Set Accelerometer Calibration Failed
            --> Accellerometer Calibration will be eresed in FLASH
Updating the Generic Meta Data type=ACC_CALIBRATION
            --> Accelerometer Calibration Not present
Initialized ST MotionVC v1.2.1
Initialized ST Acoustic SL v2.2.0 (17452 bytes allocated)
Meta Data Manager erased in FLASH
Meta Data Manager Saved in FLASH
Call to SetConnectableFunction
aci_gap_update_adv_data OK
```

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

- 3 short precision quaternions every 30 ms
- temperature/humidity/pressure data every 500 ms
- 3D accelerometer, 3D gyroscope and 3D magnetometer data every 50 ms
- signal noise microphone levels every 50 ms

Note: You can change the transmission frequency of the sensor data via the STBLESensor Android/iOS application.

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.

The above also applies to the STEVAL-MKSBOX1V1 and STEVAL-BCNKT01V1 evaluation boards.

These definitions in ALLMEMS1_config.h control the number of quaternions sent by the application 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 number of quaternions sent to each Bluetooth package.

By default, the application sends 3 quaternions every 30 ms.

The same allmems1_config.h file also defines:

- `ALLMEMS1_DEBUG_CONNECTION` and `ALLMEMS1_DEBUG_NOTIFY_TRAMISSION` to enable some debugging information for BLE communication

The MotionFX (INEMOEngine PRO) library has an auto-calibrating procedure and the calibration status is transmitted via BLE to the client:

- on the NUCLEO-F446RE or NUCLEO-L476RG boards, you can press the user button to reset the library calibration status and force a new auto-calibration procedure.
- for the STEVAL-MKSBOX1V1 and STEVAL-BCNKT01V1 evaluation boards, this procedure can be done only through the STBLESensor application.

The MotionAR library is able to recognize the following activities:

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

The MotionCP library recognizes and provides real-time information about how the user is carrying the board, which equates to the phone carry position (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE):

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

The MotionFA library is able to count the number of bicep curls, squats and push-ups performed by a user (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE)

Important:
For NUCLEO-L476RG board, due to flash size constraints, the MotionCP and MotionFA features are not available together. As default the MotionFA is enabled.

You must modify the ALLMEMS1_config.h file by disabling the `#define ALLMEMS1_MOTIONFA` to disable the MotionFA and enable MotionCP.

The MotionGR library is able to recognize gestures like (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE):

- pick up
- glance
- wake up in hand

The MotionID library is able to distinguish motion intensity in a range from 0 to 10 (according to the library indexes) corresponding to the following activities: on desk, hand on bed/couch/cushion, light movements, biking, typing/writing, high intensity typing/slow walking, washing hands/walking, fast walking/jogging, running/brushing teeth, sprinting (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE).

The MotionPE library is able to distinguish the poses like (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE):

- sitting
- standing
- lying down.

The MotionSD library is able to distinguish the user working modes like (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE):

- sitting at the desk
- standing desk

The MotionTL library is able to provide real-time information about the tilt angles of the user device and it is also able to perform accelerometer 6-position calibration (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE).

The MotionVC library is able to distinguish the type of vertical movement like (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE):

- on floor
- up/down
- stairs
- elevator
- escalator

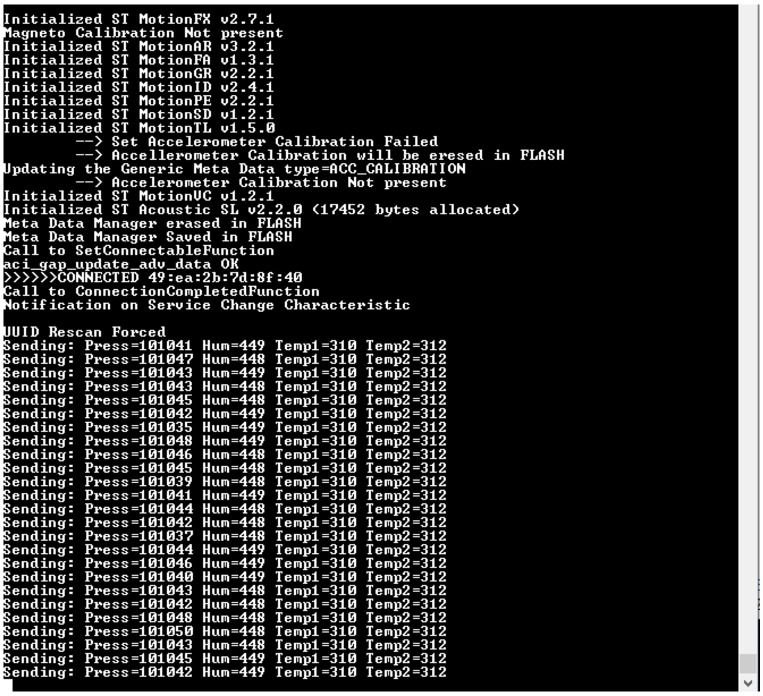
The AcousticSL library can localize audio sound sources using the data acquired from microphones (feature not available on the STEVAL-MKSBOX1V1).

The AcousticBF library provides real-time beam forming software, using the audio signals acquired from two digital MEMS microphones, it creates a virtual directional microphone pointing to a fixed direction in space (feature available only on the STEVAL-BCNKT01V1).

The BlueVoiceADPCM library implements a vendor-specific profile enabling voice communication with Bluetooth low energy (feature available on the STEVAL-BCNKT01V1).

When an Android/iOS device is connected to the NUCLEO-F446RE, NUCLEO-L476RG, or STEVAL-MKSBOX1V1 (if the define `#define ALLMEMS1_ENABLE_PRINTF` is enabled) board, you can control data transmitted via the board.

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



For the STEVAL-MKSBOX1V1, when the Android/iOS device is not connected for more than 20 seconds, the board enters in shutdown mode, which can be enabled or disabled by the macro `ENABLE_SHUT_DOWN_MODE`.

The accelerometer events can be selected and used to wake the board up to connect it to Android/iOS again via the constant **WakeupSource** in the file *main.c*. The **Double Tap** event is set as default.

Through the define `#define RANGE_TIME_WITHOUT_CONNECTED` in the *main.h* file, it is possible to modify this time value.

1.10 Android and iOS sample client application

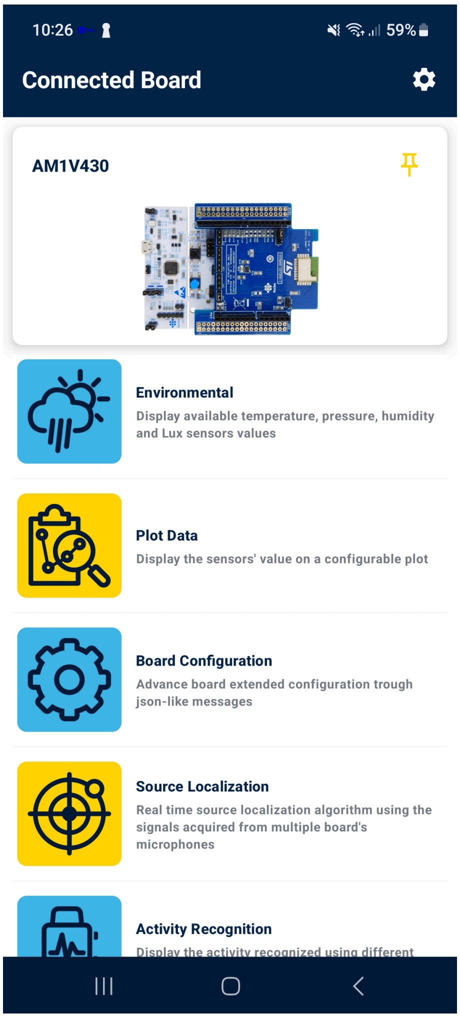
The FP-SNS-ALLMEMS1 software for STM32Cube is compatible with STBLESensor (or STBLESensClassic) Android/iOS applications (ver. 4.20.0 or higher) available at the respective Play/iOS stores.

The STBLESensor allows Over-The-Air firmware updates.

1.10.1 Main page

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

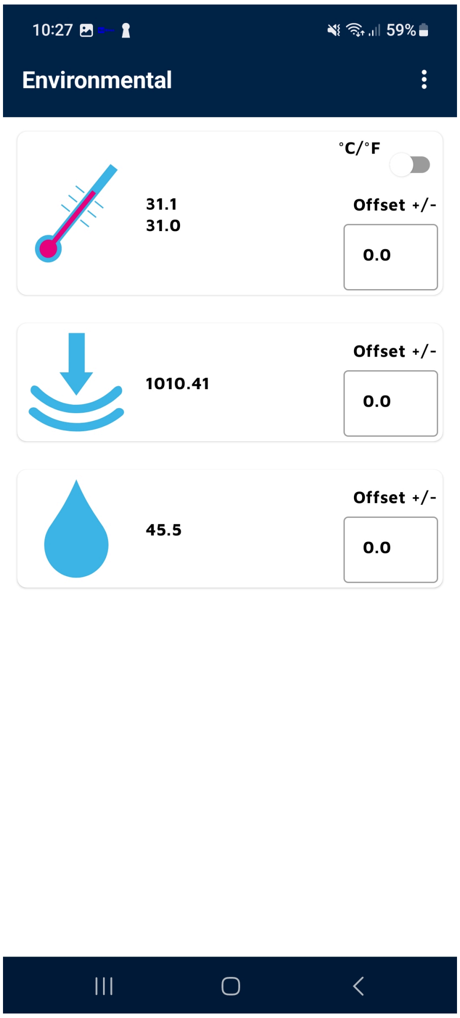
Figure 12. ST BLE Sensor Connected Board



1.10.2 Environmental

The following page shows the value for the tempetature, pressure and Humidity.

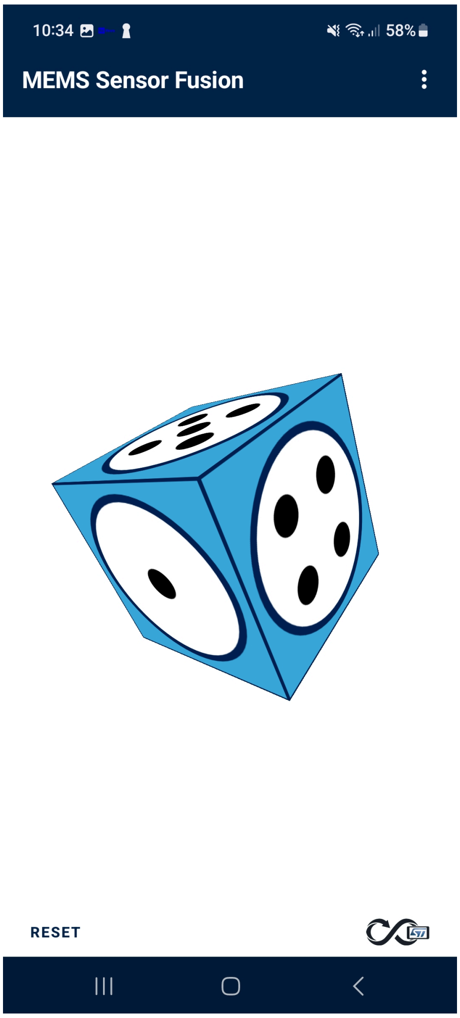
Figure 13. **STBLESensor (Android version) environmental page**



1.10.3 MEMS sensor fusion

The following page shows a cube that rotates with board movement.

Figure 14. STBLESensor (Android version) MotionFX sensor fusion page

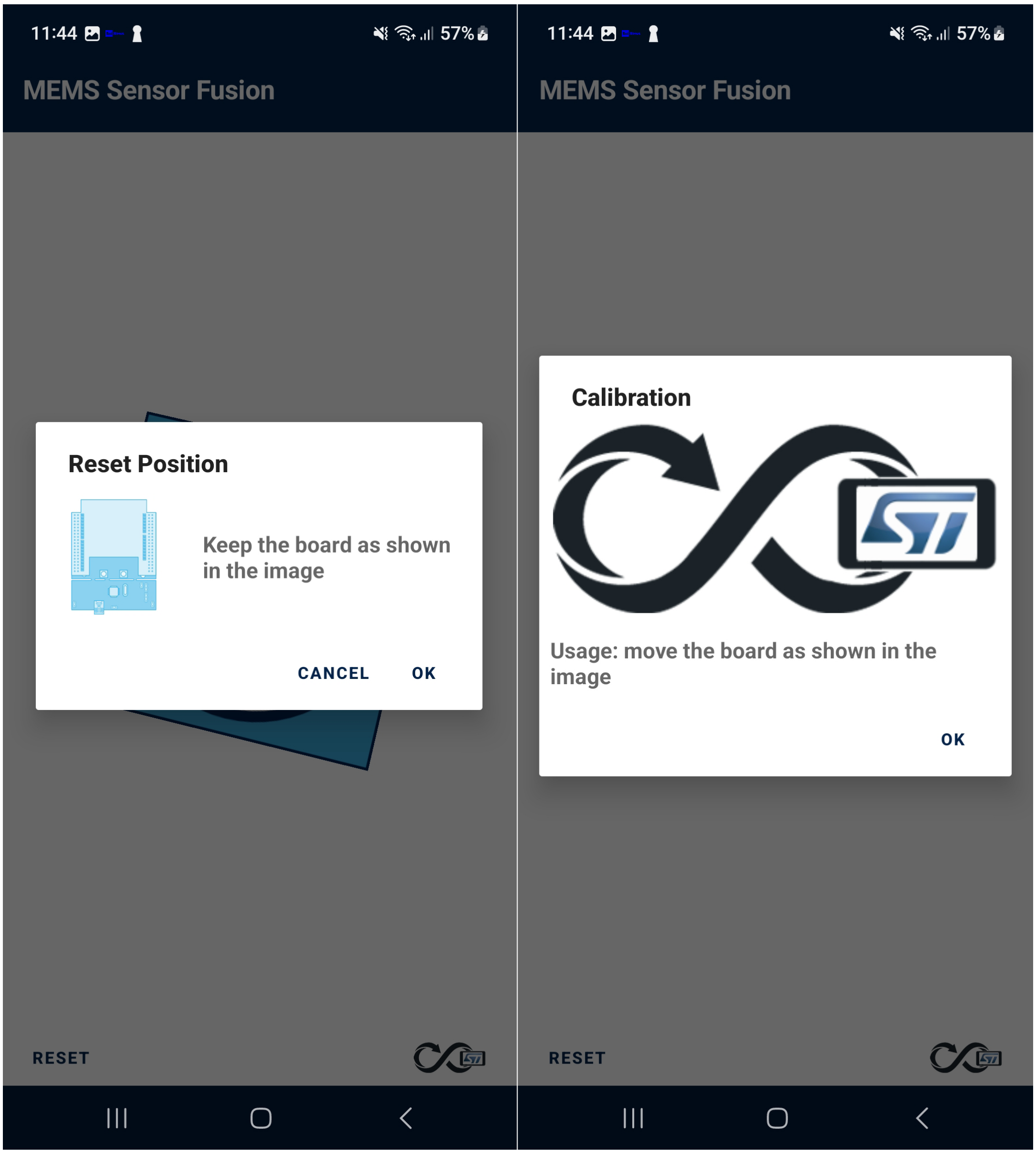


On this page, there are two buttons along the bottom:

- the left one is for resetting the cube position.
- the right one 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).

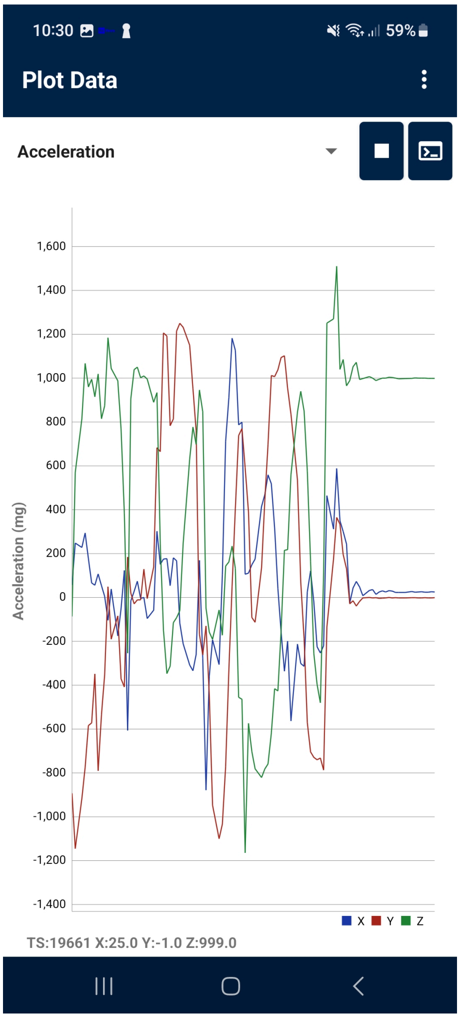
Figure 15. STBLESensor (Android version) popup windows



1.10.4 Plot data

You can plot any value from the sensor expansion boards.

Figure 16. **STBLESensor (Android version) accelerometer plot**



1.10.5 Debug console

In the option menu below, you can open:

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

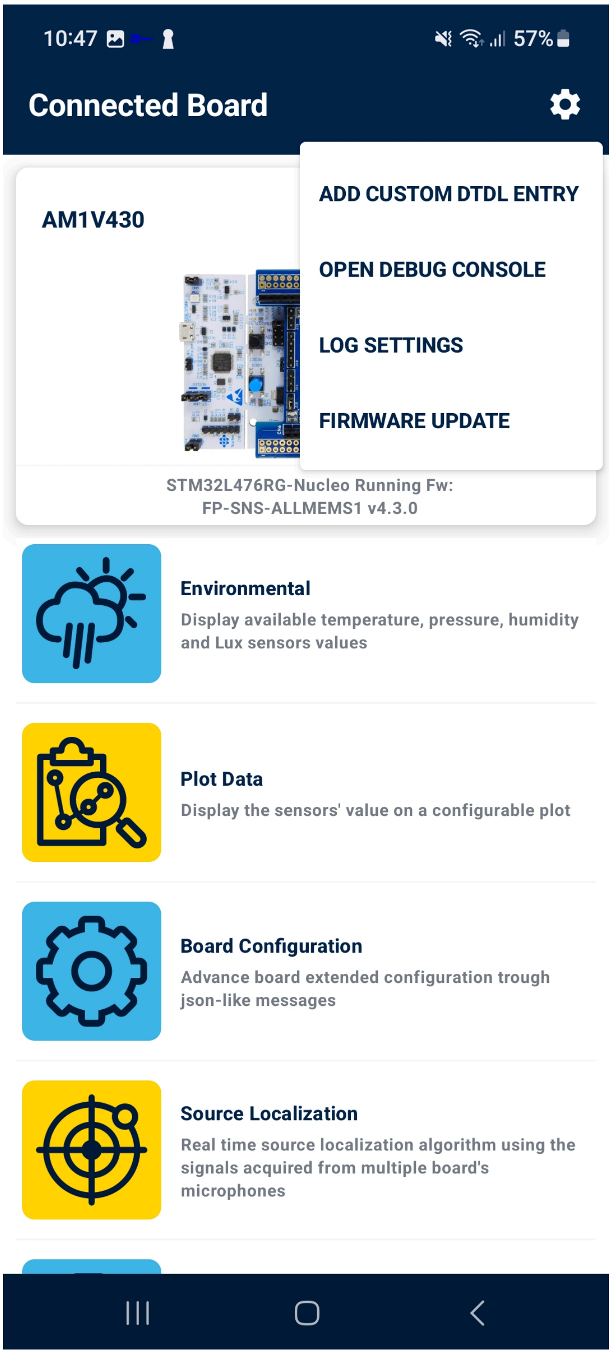
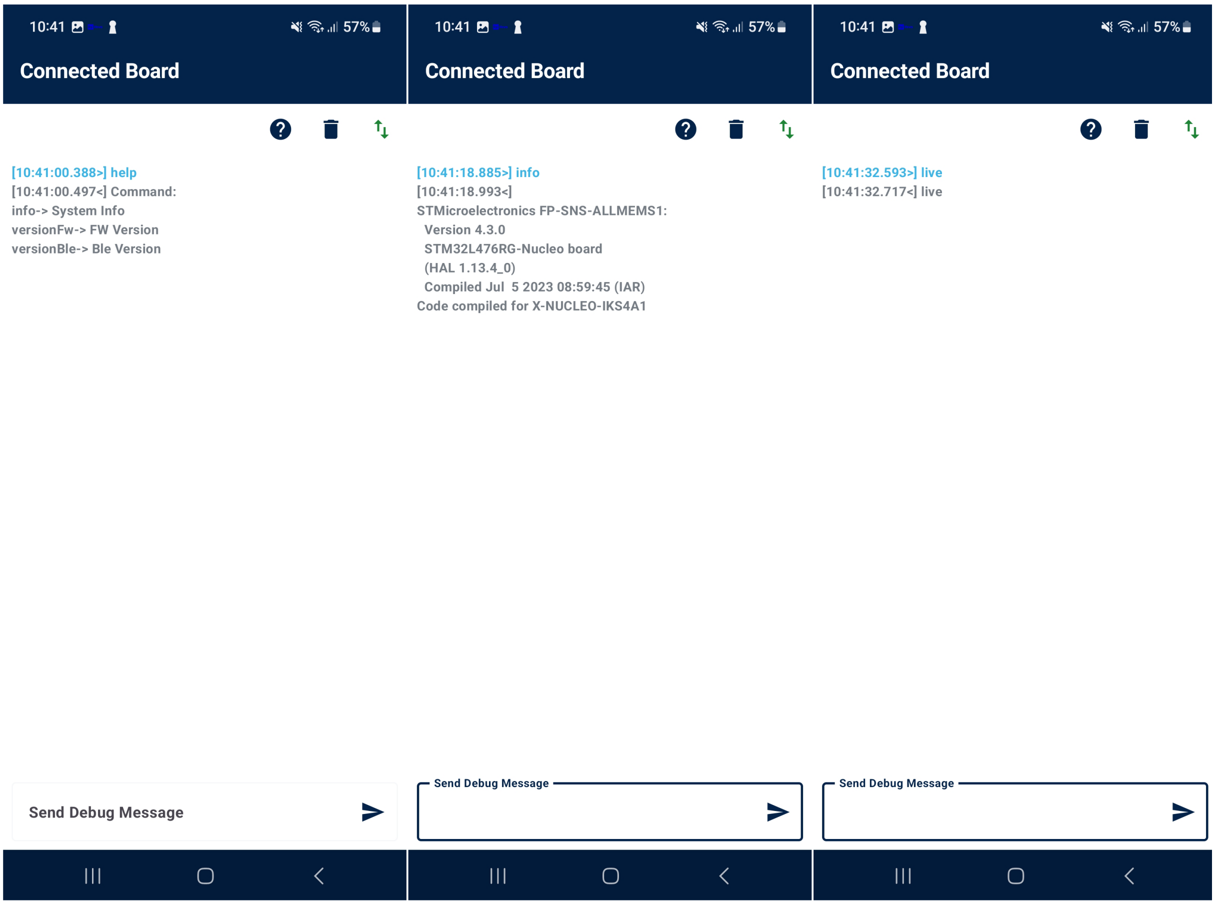


Figure 17. STBLESensor (Android version) options selection

If the Debug Console is selected.

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



You can change the transmission frequency of the sensor values through the debug console:

- for temperature/humidity/pressure with the command:
  - @TM: the application sends environmental data every 5 s
  - @TH: the application sends environmental data every 1 s
  - @TL: the application sends environmental data every 100 ms
  - @TD: the application sends environmental data at the default rate (500 ms)
- 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 at the default rate (50 ms)
- for signal noise microphone levels with the command:
  - @MM: the application sends the data every 5 s
  - @MH: the application sends the data every 1 s
  - @ML: the application sends the data every 100 ms
  - @MD: the application sends the data at the default rate (50 ms)

1.10.6 Enable hardware features

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:

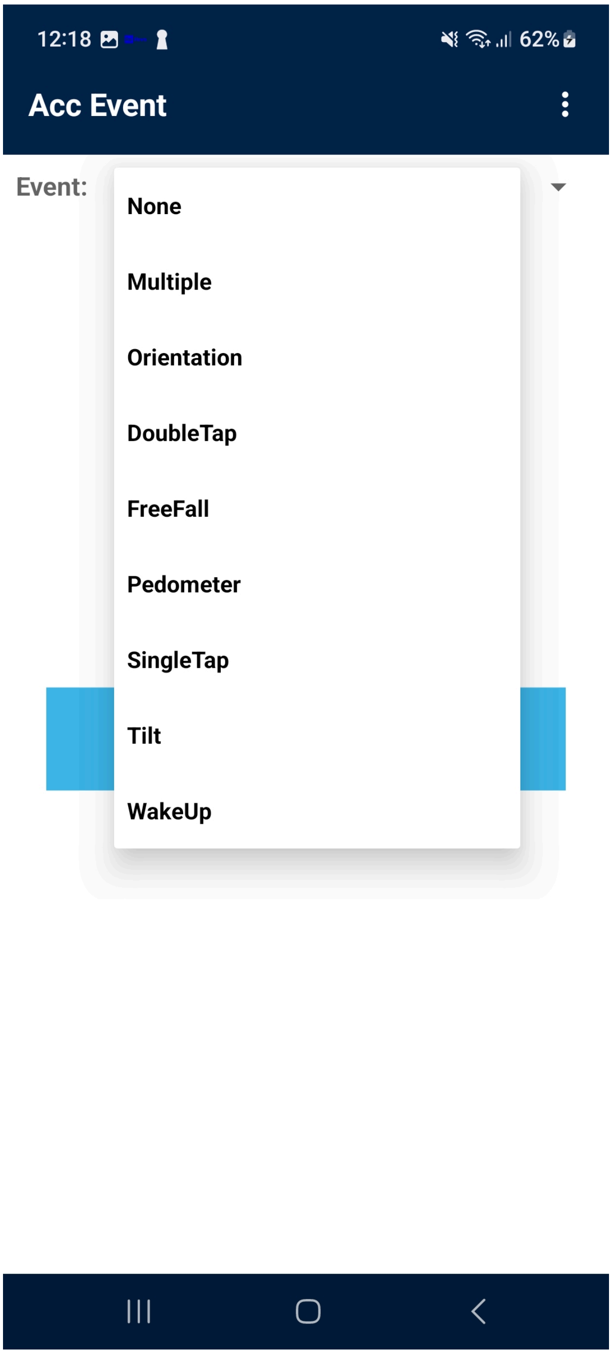
- LSM6DSV16X on X-NUCLEO-IKS4A1 expansion board for NUCLEO-L476RG board only
- LSM6DSO on X-NUCLEO-IKS01A3 expansion board for NUCLEO-L476RG board only
- LSM6DSM for STEVAL-BCNKT01V1
- LSM6DSOX for STEVAL-MKSBOX1V1 evaluation board

The orientation hardware feature is the default setting.

Figure 19. STBLESensor (Android version) orientation hardware feature

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

Figure 20. STBLESensor (Android version) hardware feature menu



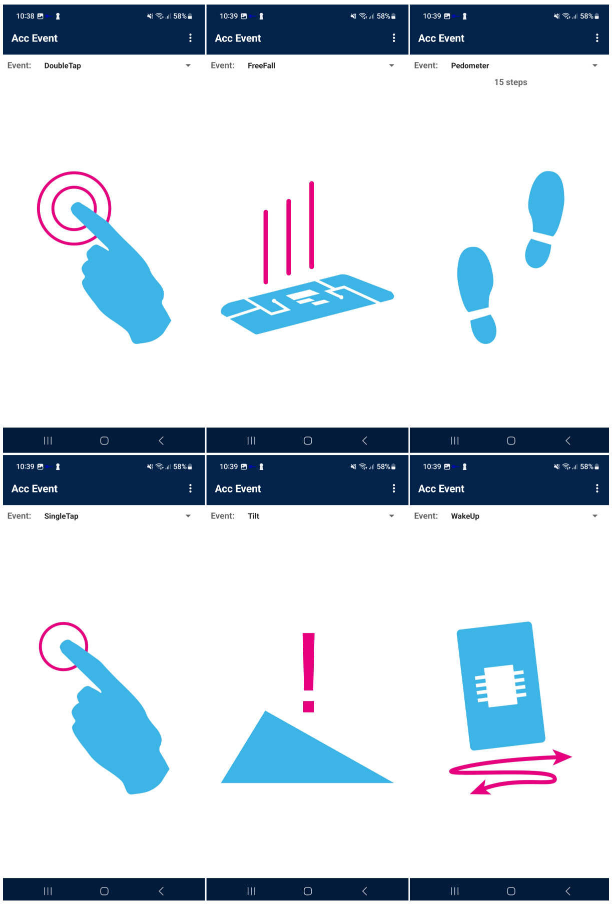
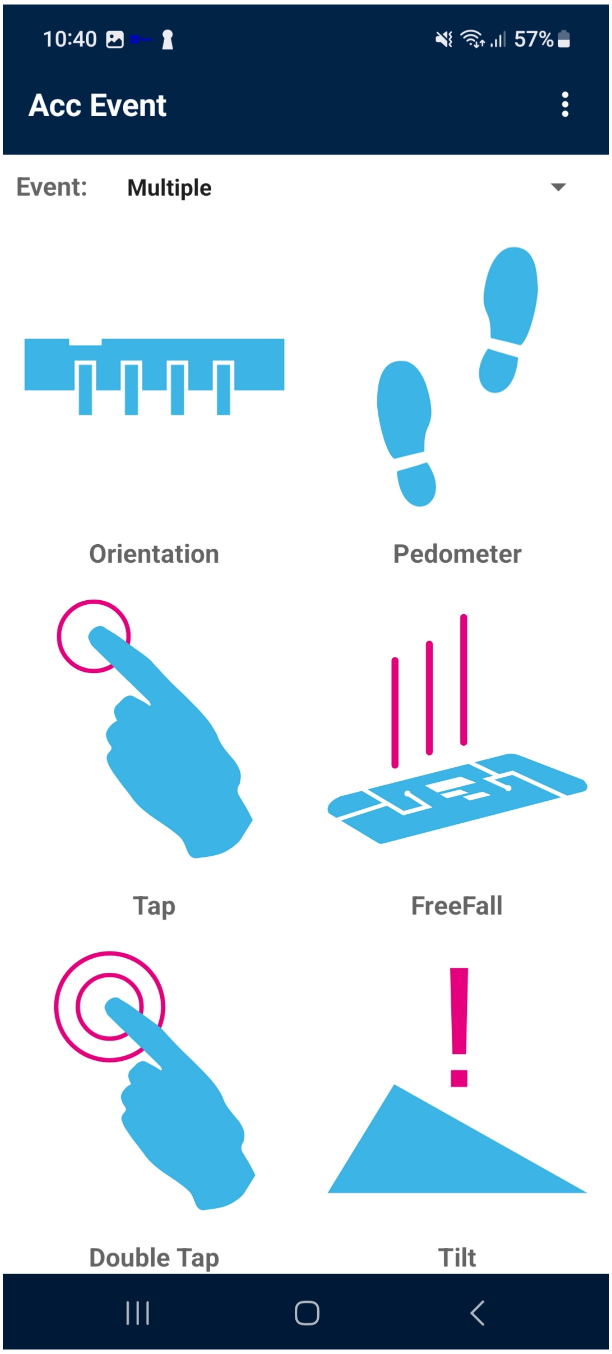


Figure 21. STBLESensor (Android version) hardware feature examples

Figure 22. **STBLESensor (Android version) multiple hardware feature**



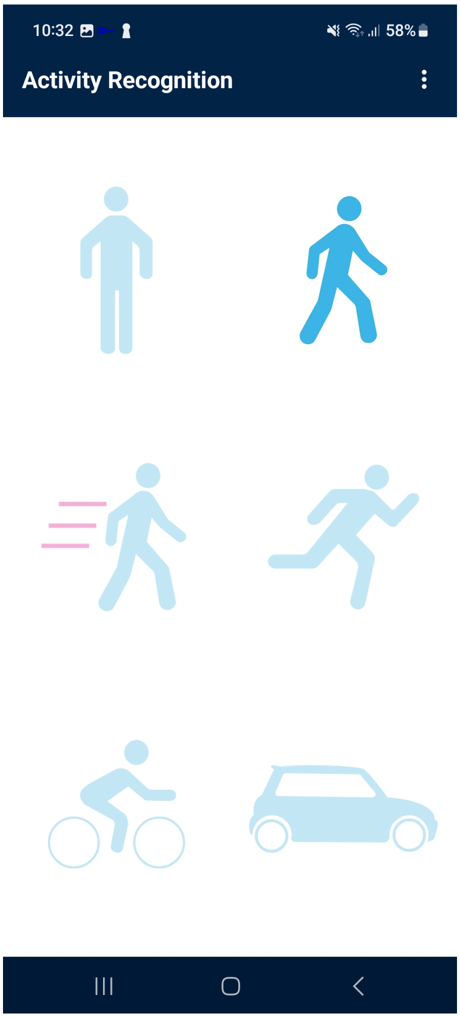
1.10.7 Activity recognition

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

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

Note: As the algorithm has to collect data before recognizing any activity, all the images are shown in a lighter blue for few seconds after the demo starts.

Figure 23. STBLESensor (Android version) MotionAR activity recognition page



1.10.8 Carry position

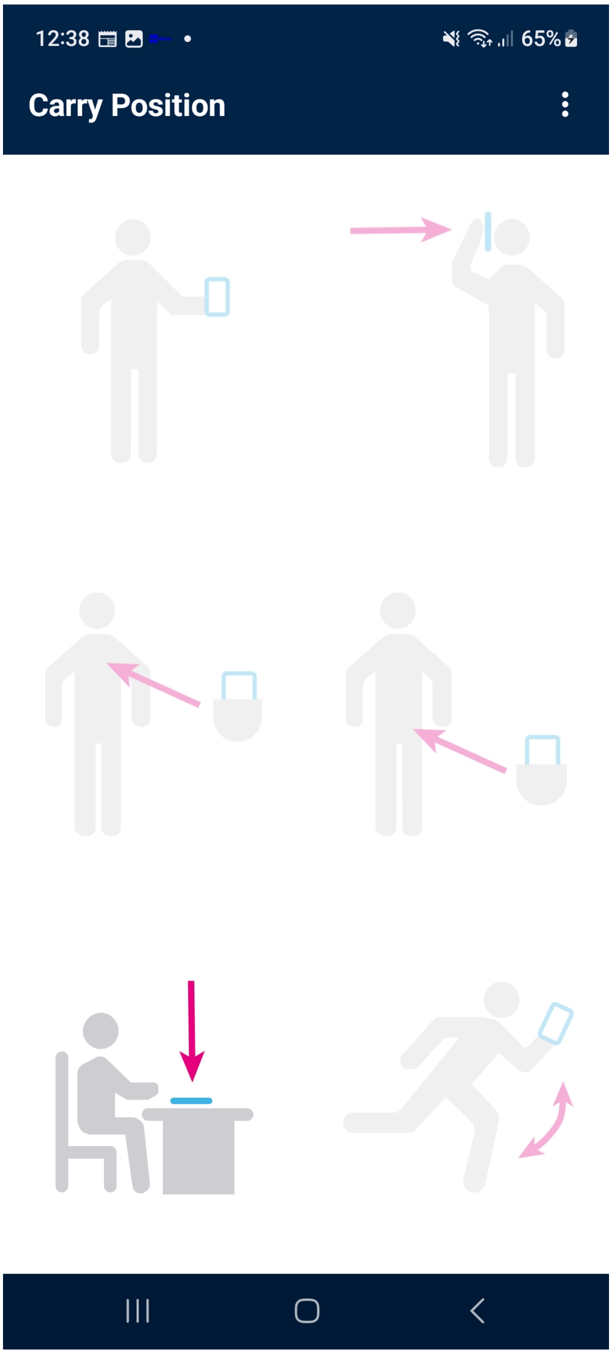
For the MotionCP algorithm, the page below is shown, with information about how the user is carrying the board, which equates to phone carry positions (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE:

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

Note: As the algorithm has to collect data before recognizing any activity, all the images are shown in grayscale for few seconds after the demo starts.

Important:
For NUCLEO-L476RG board, due to flash size constraints, the MotionCP and MotionFA features are not available together. As default the MotionFA is enabled.
You must modify the ALLMEMS1_config.h file by disabling the `#define ALLMEMS1_MOTIONFA` to disable the MotionFA and enable MotionCP.

Figure 24. **STBLESensor (Android version) MotionCP carry position recognition page**



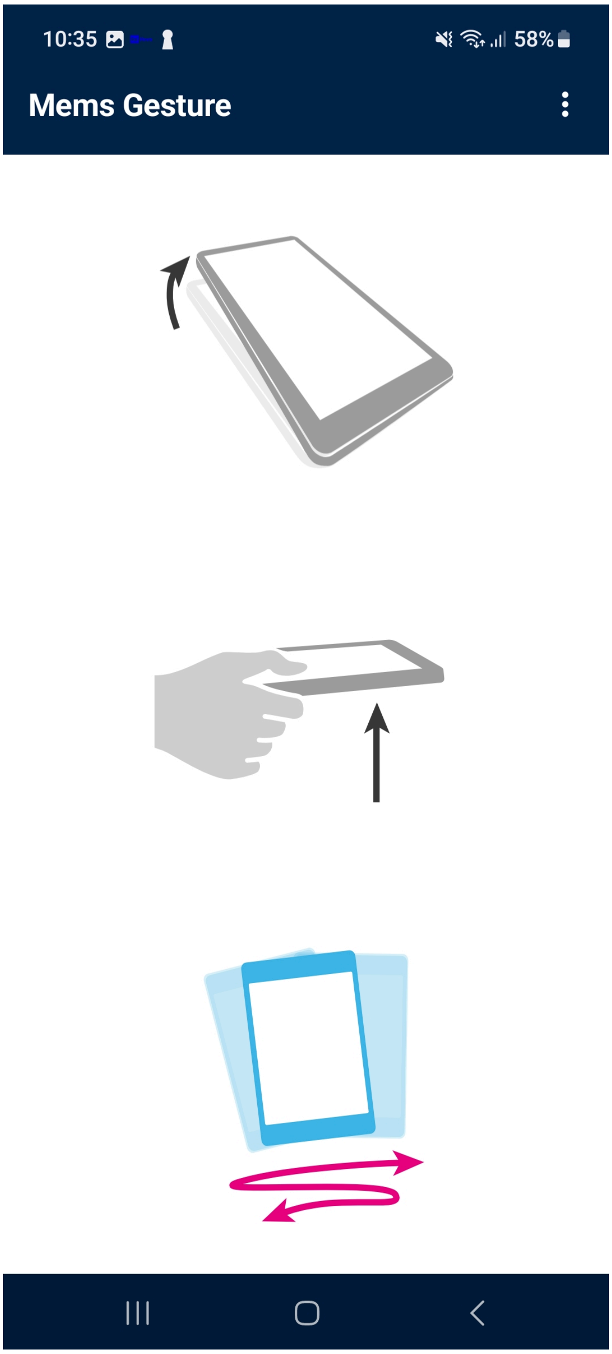
1.10.9 Gesture recognition

The page shown below is available with gesture recognition information like (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE):

- glance: the user moves the device to look at the display (in our example, to look at the sensor)
- pick up: the user picks up the device
- wake up: the user shakes the device

Each time an event is detected, the icon animates and becomes colored. After three seconds, or when a new event arrives, the icon goes gray again.

Figure 25. **STBLESensor (Android version) MotionGR gesture recognition page**



1.10.10 Intensity detection

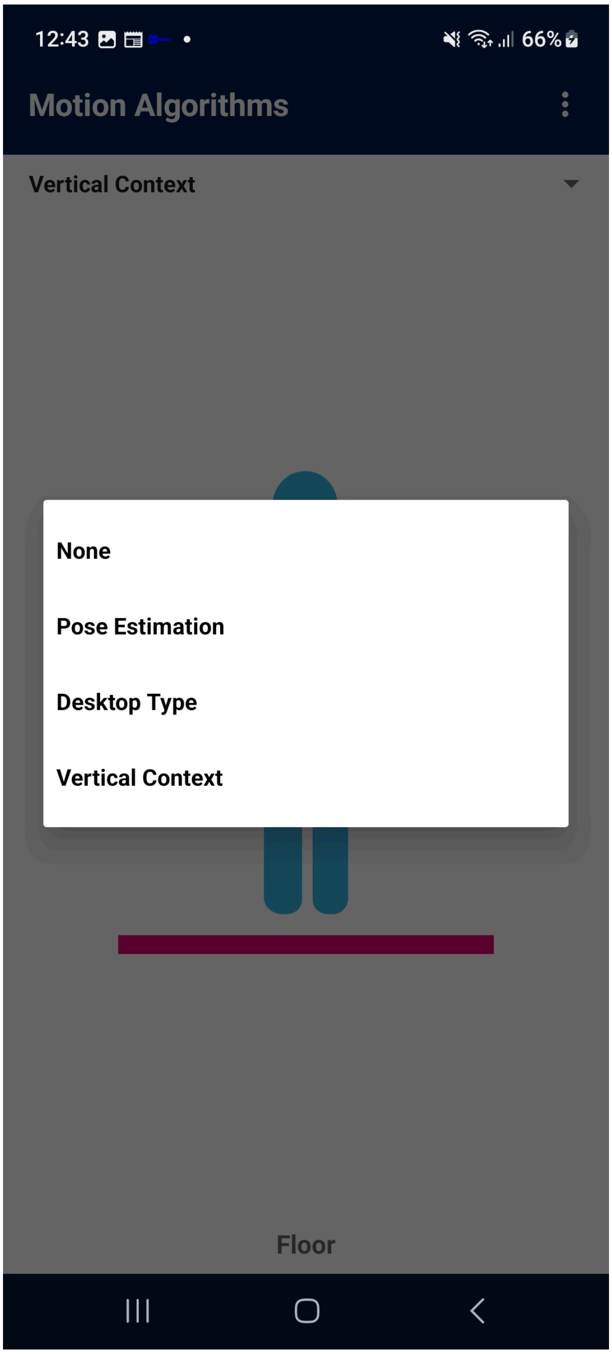
For the MotionID algorithm , the page below is shown (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE).

Figure 26. STBLESensor (Android version) Motion Intensity page

1.10.11 Motion algorithms

For the MotionPE, MotionSD and MotionVC algorithms, the Motion Algorithms page below is shown (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE).

Figure 27. STBLESensor (Android version) Motion Algorithms page - menu



The MotionPE algorithm provides real-time information about the user current pose based on data from a device. It is able to distinguish the following poses:

- sitting
- standing
- lying down

Figure 28. STBLESensor (Android version) Motion Algorithms page - MotionPE algorithm

The MotionSD algorithm provides real-time information about the user working mode:

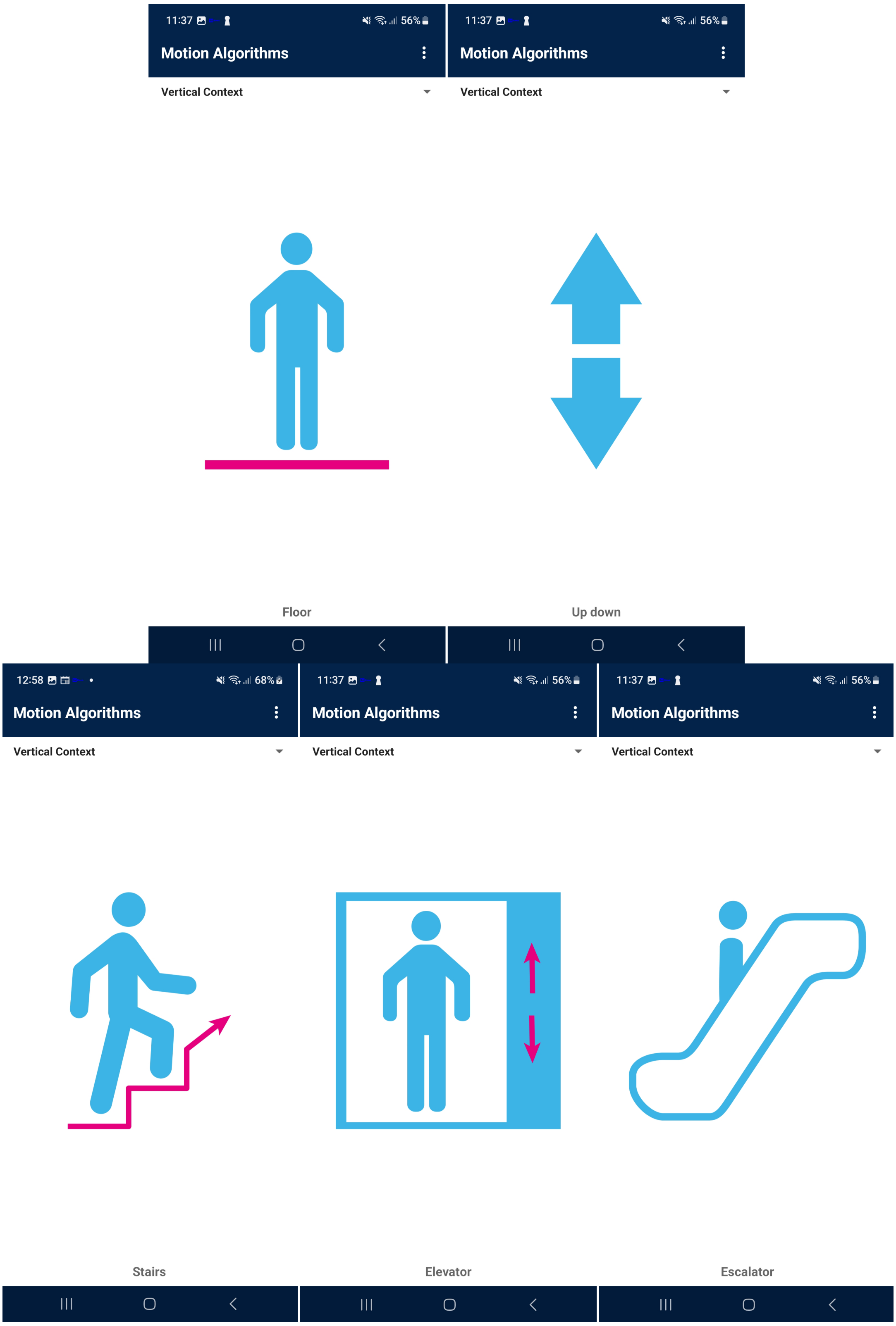
- sitting at the desk
- standing desk position

Figure 29. STBLESensor (Android version) Motion Algorithms page - MotionSD algorithm

The MotionVC algorithm provides real-time information about vertical movement. The library is able to detect a change of altitude and distinguish the type of vertical movement:

- floor
- up/down
- stairs
- elevator
- escalator

Figure 30. STBLESensor (Android version) Motion Algorithms page - MotionVC algorithm



1.10.12 Fitness activities

For the MotionFA algorithm, the page below is shown (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE).

It provides real-time information about the repetition quantity of various fitness activities performed by a user. It is able to count the number of bicep curls, squats and push-ups.

Important:
For NUCLEO-L476RG board, due to flash size constraints, the MotionCP and MotionFA features are not available together. As default the MotionFA is enabled.
You must modify the ALLMEMS1_config.h file by disabling the `#define ALLMEMS1_MOTIONFA` to disable the MotionFA and enable MotionCP.

Figure 31. ST BLE Sensor (Android version) Fitness Activities page - menu

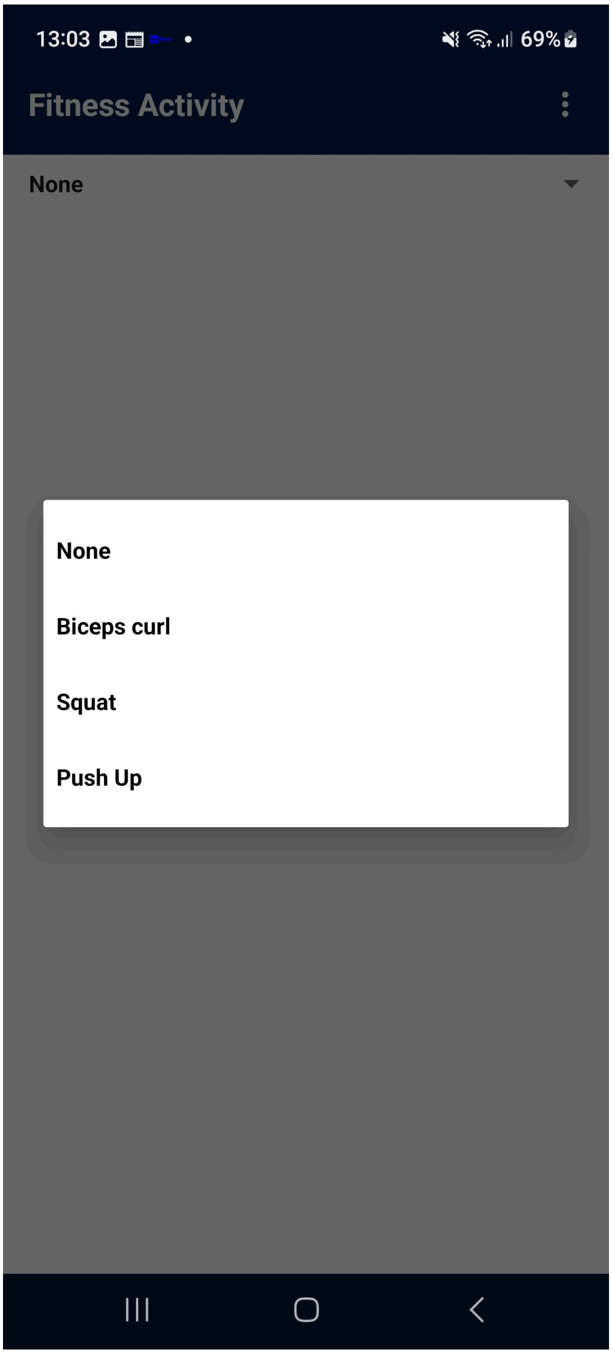


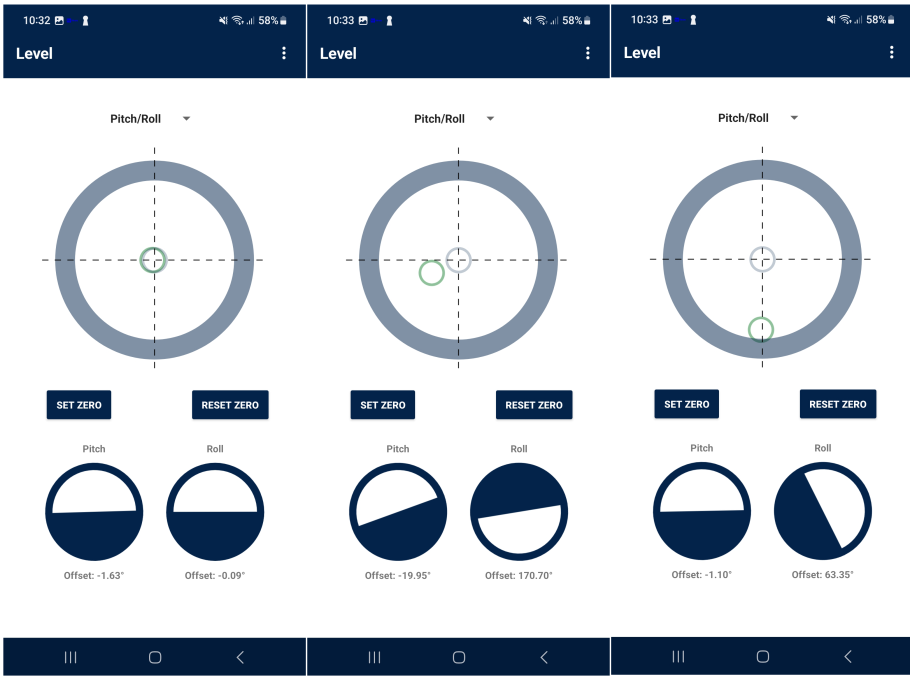
Figure 32. ST BLE Sensor (Android version) Fitness Activities page - biceps curl, squat, push up

1.10.13 Tilt measurement

For the MotionTL algorithm, the page below is shown (feature not available on STEVAL-BCNKT01V1 and NUCLEO-F446RE).

It provides real-time information about the tilt angles of the user device.

Figure 33. STBLESensor (Android version) level page



The library is also able to perform accelerometer 6-position calibration. In this case, the debug console must be used and send these commands:

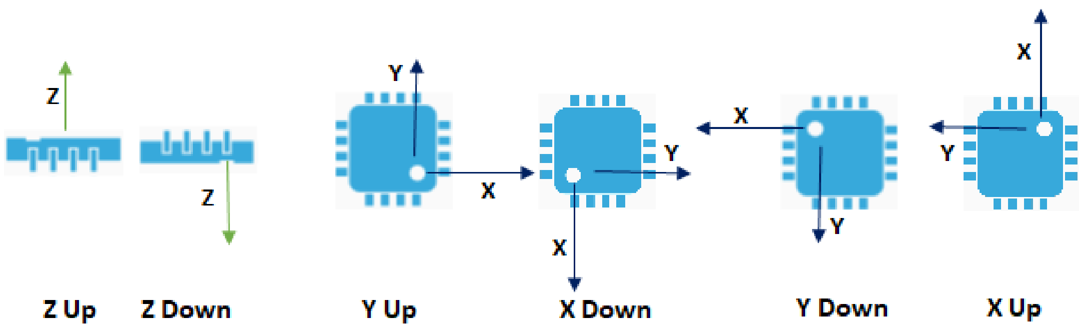
- TLcalibstart: to start the accelerometer 6-position calibration
- TLcalibstop: to stop the calibration (the process fails)

When the command to start the calibration is sent, you must place the board at position 1 and send the command "next" and so on for all the six positions.

In the end, when the calibration of position 6 is executed, the firmware gives a feedback whether the calibration has been performed or failed.

To obtain the accelerometer calibration you have to put the board in the correct sequence as shown below.

Figure 34. **Accelerometer calibration sequence**



1.10.14 E-compass

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

Figure 35. STBLESensor (Android version) MotionFX page

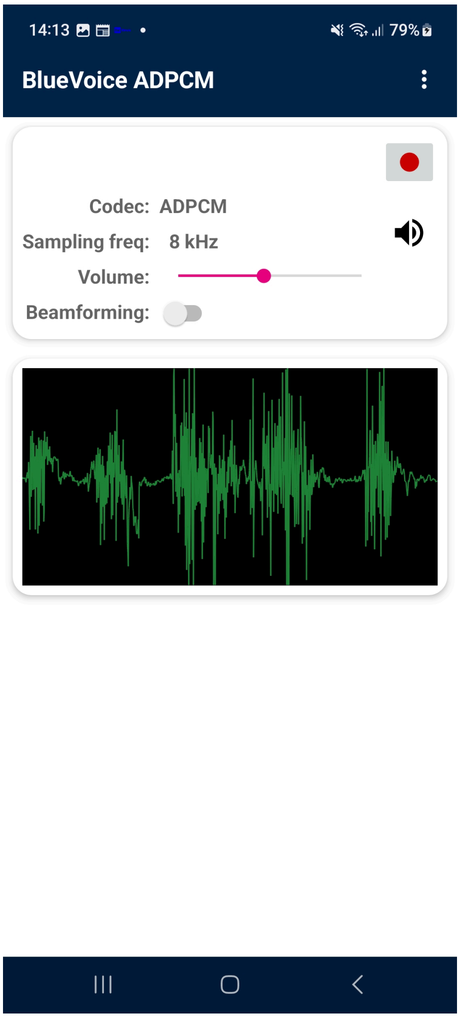
On the page bottom, the right button shows the MotionFX library calibration status (black for not calibrated, green for calibrated). Clicking it forces a magneto calibration.

1.10.15 BlueVoiceADPCM voice

If the BlueVoiceADPCM voice over BLE library is enabled the following page is available with the following functions:

- Play back the audio stream received from the ST device
- Audio playback recording
- Enable/disable beamforming (if available)

Figure 36. STBLESensor (Android version) BlueVoice page

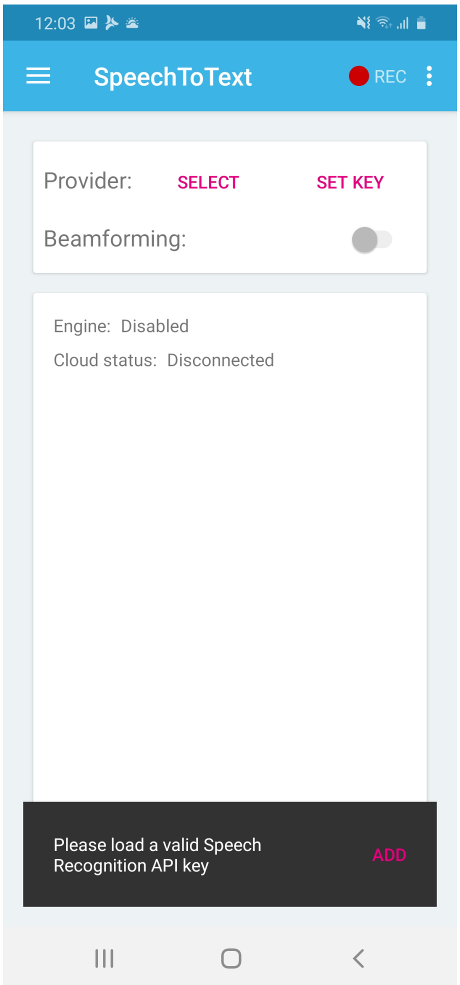


The audio playback begins as soon as the streaming from the peripheral node starts. The volume can be adjusted using the slider or muted by clicking on the speaker icon.

1.10.15.1 Speech to text (available STBLESensor Classic mobile application)

If the BlueVoice library is enabled, the following page is also available. This page is dedicated to the cloud speech to text services.

Figure 37. STBLESensClassic (Android version) Speech to text page



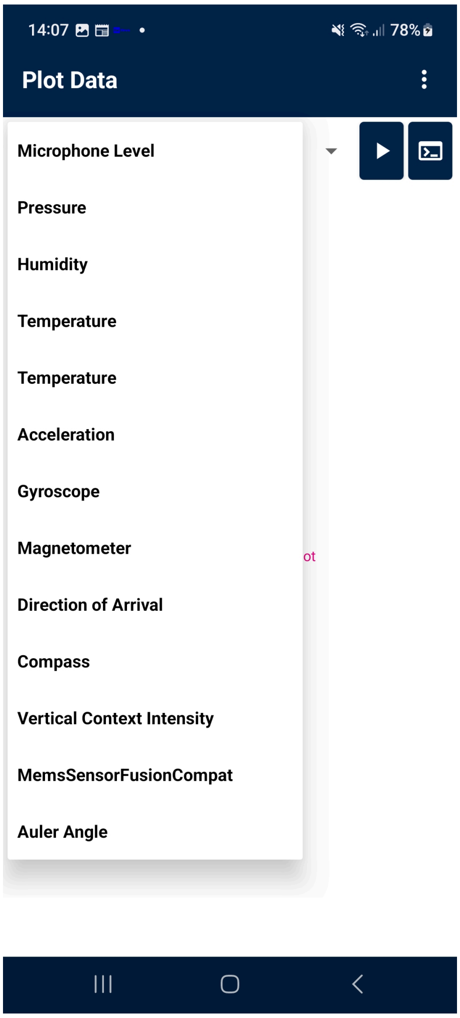
RELATED LINKS

Visit STBLESensor applicaton web page on www.st.com for further information

1.10.16 Direction of Arrival

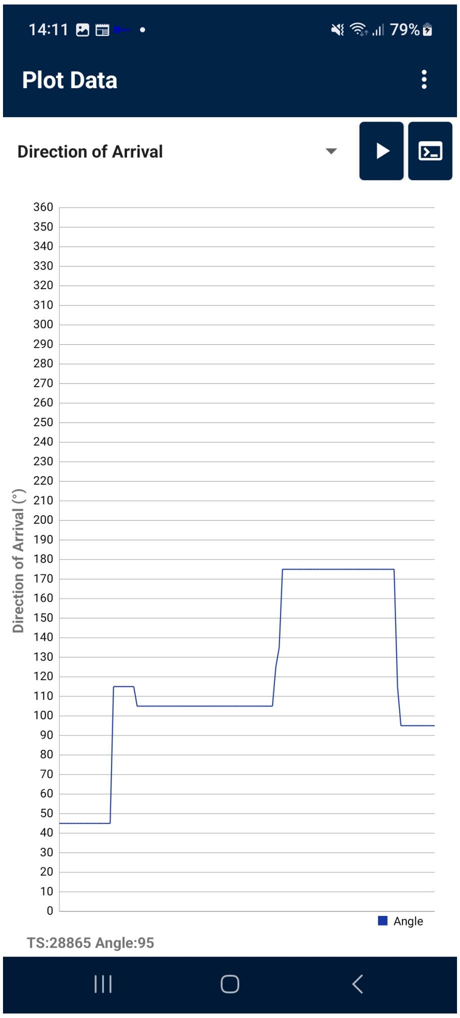
If the AcousticSL library is enabled, Direction of Arrival item is shown in the plot length menu (feature not available on the STEVAL-MKSBOX1V1).

Figure 38. STBLESensor (Android version) Direction of Arrival menu selection



If the Direction of Arrival menu item is selected, the audio sound source localization algorithm is activated and the associated plot is shown.

Figure 39. STBLESensor (Android version) audio source localization plot example



Again, if the audio sound source localization algorithm is activated, the pages shown below are available.

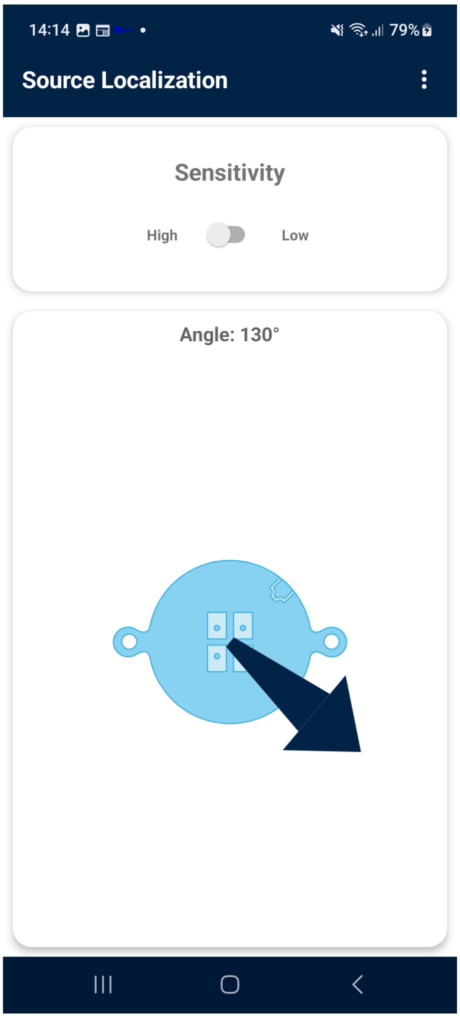
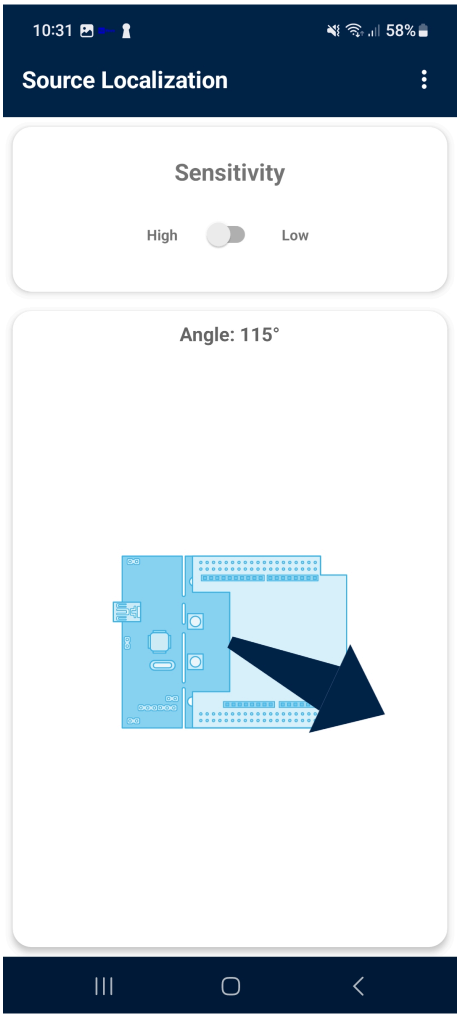


Figure 40. STBLESensor (Android version) audio source localization BlueCoin Page

Figure 41. STBLESensor (Android version) audio source localization STM32 Nucleo Page

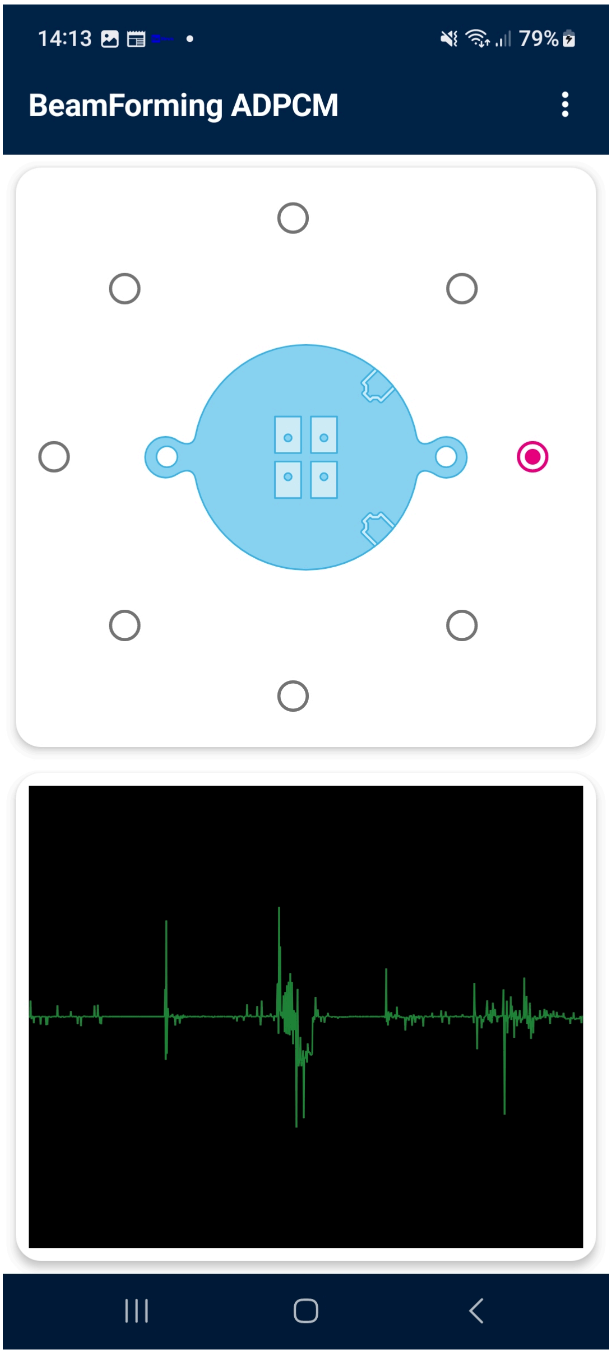


In a noisy environment, use low sensitivity.

1.10.17 Beam Forming

If the AcousticBF library is enabled, the pages shown below are available (feature available only on STEVAL-BCNKT01V1):

Figure 42. STBLESensor (Android version) audio beam forming BlueCoin Page



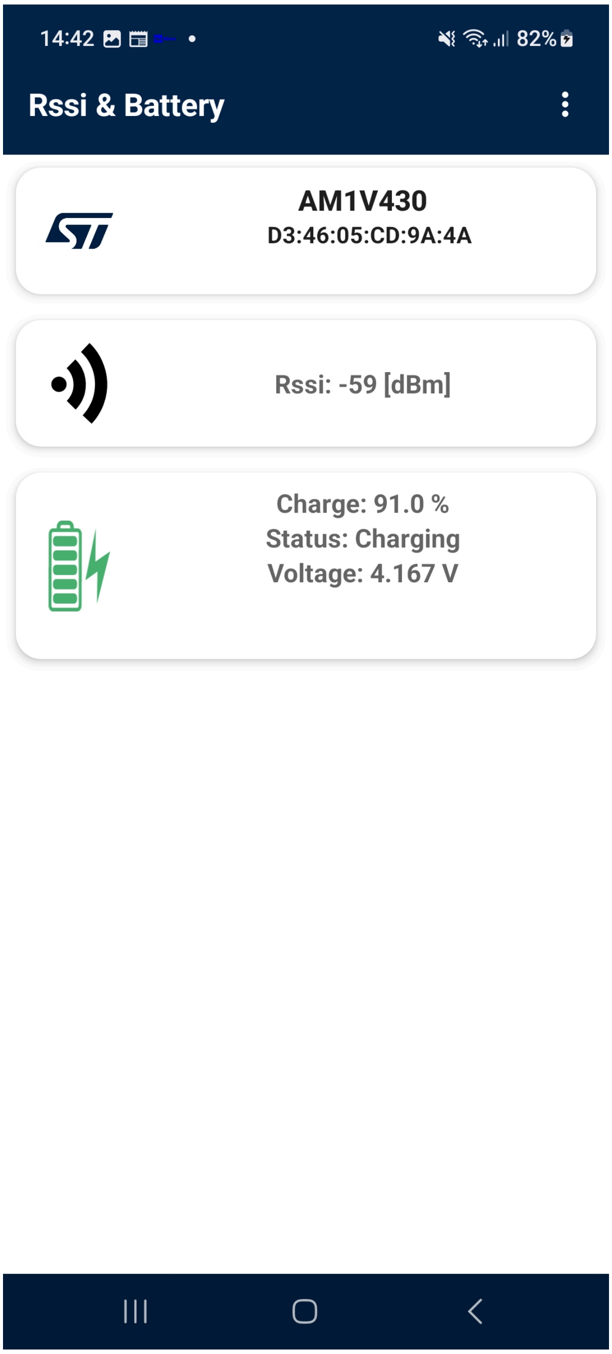
The AcousticBF provides real-time beam forming software, using the audio signals acquired from two digital MEMS microphones, it creates a virtual directional microphone pointing to a fixed direction in space.

From the STBLESensor beam forming page, you can set the direction in space to create microphone with a virtual direction.

1.10.18 Rssi and battery

This page shows RSSI of the Bluetooth signal strength and, for STEVAL-MKSBOX1V1 and STEVAL-BCNKT01V1, if the battery is connected, the charge percentage, measured voltage and battery status (charging/discharging/low battery).

Figure 43. STBLESensor (Android version) Battery and RSSI information



The RSSI value is updated every 0.5 seconds.

1.10.19 Board configuration

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

Figure 44. STBLESensor (Android version) Board Configuration

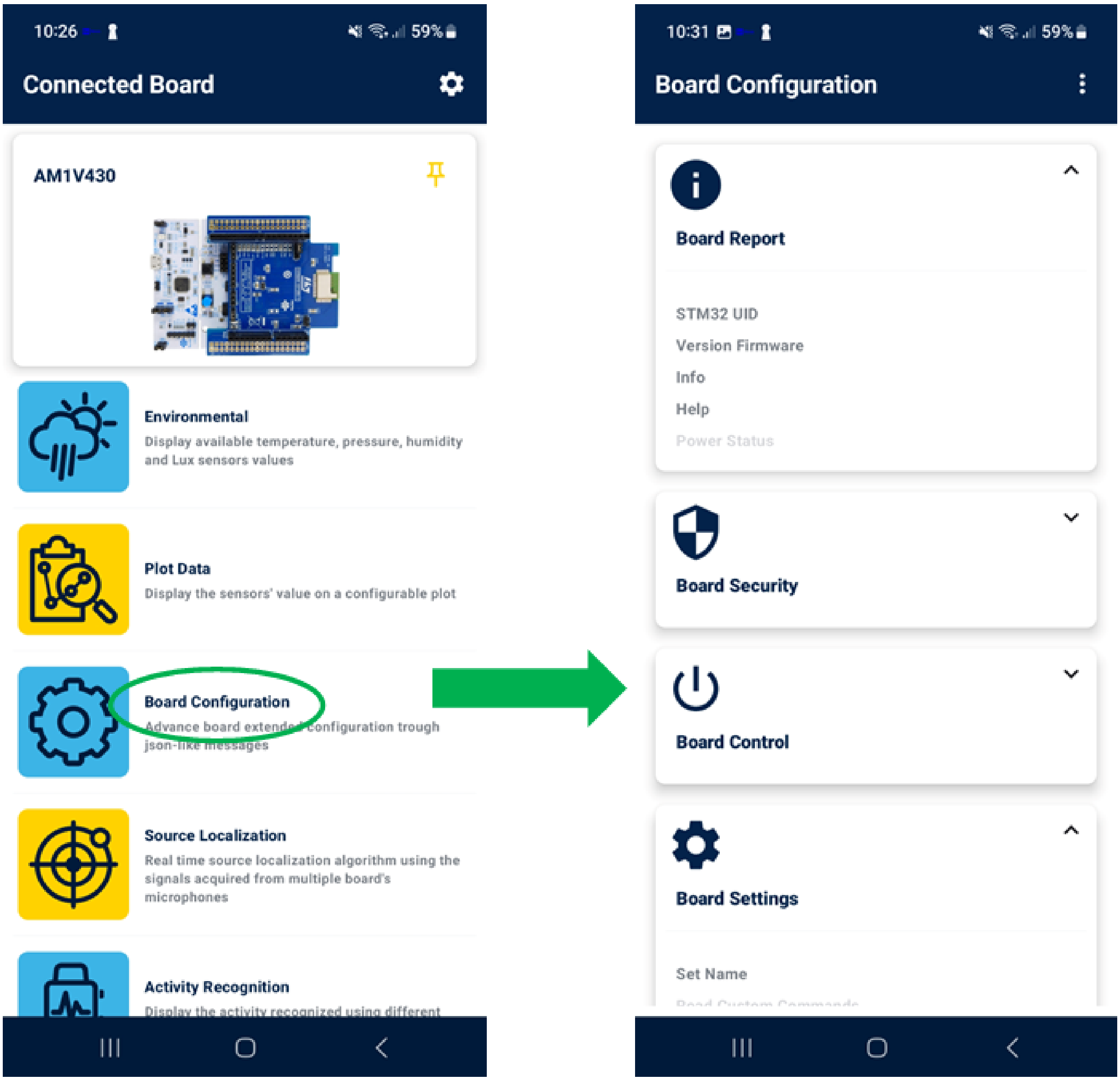
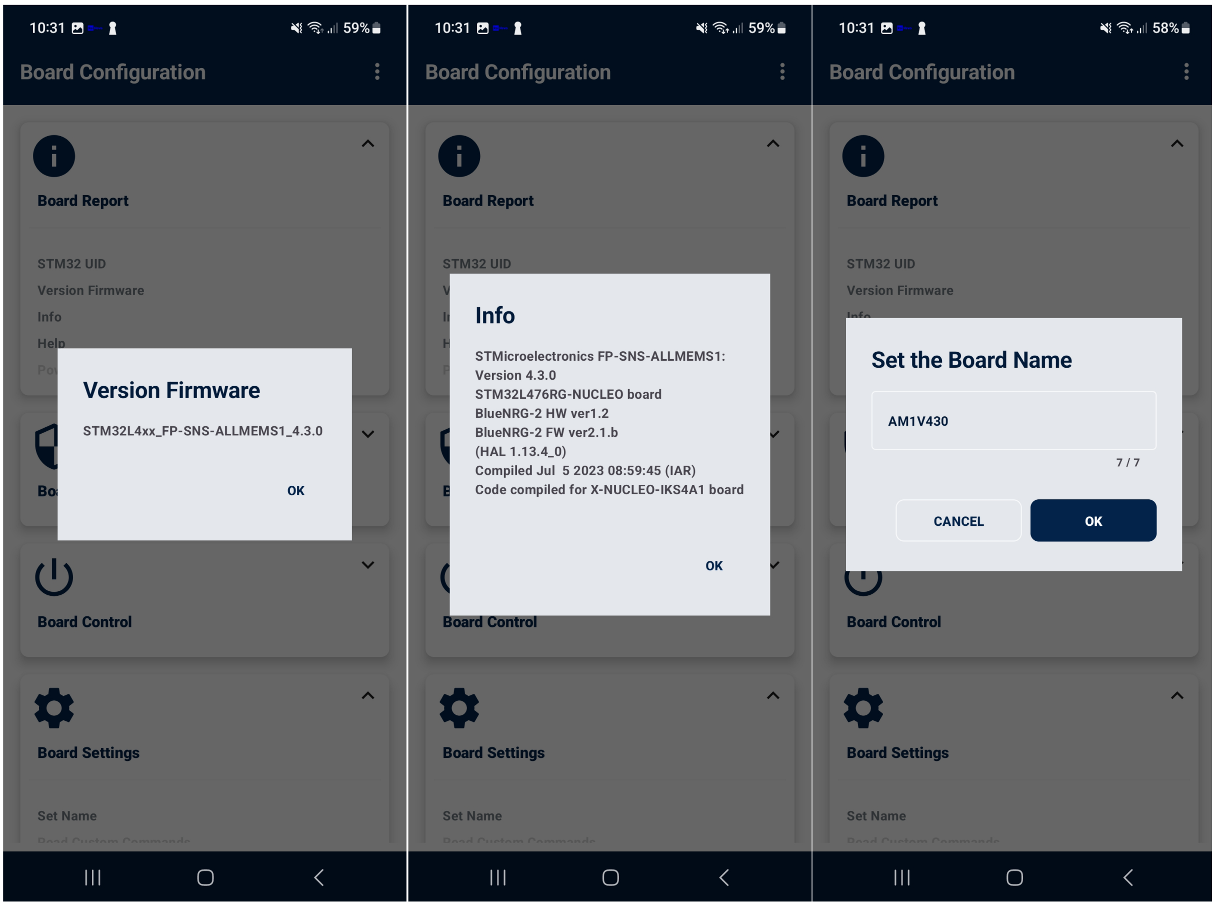


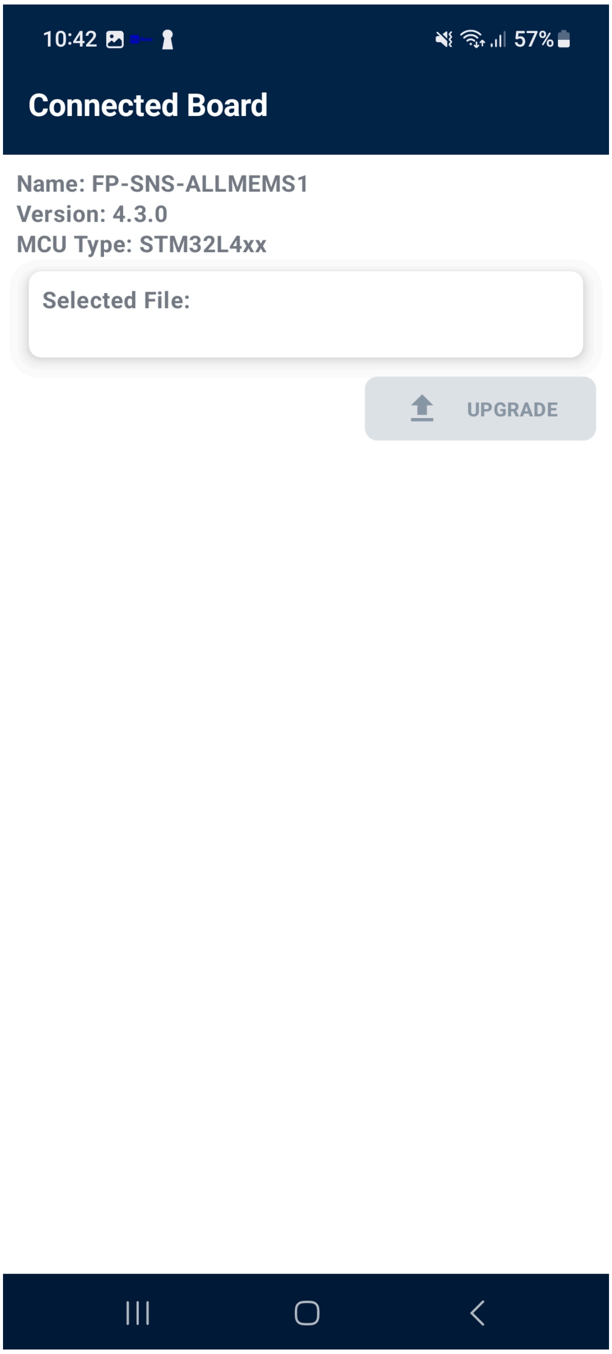
Figure 45. **STBLESensor (Android version) Board Configuration - Commad Version Firmware, Info, Set Board Name**



1.11 Firmware-Over-The-Air update with STBLESensor

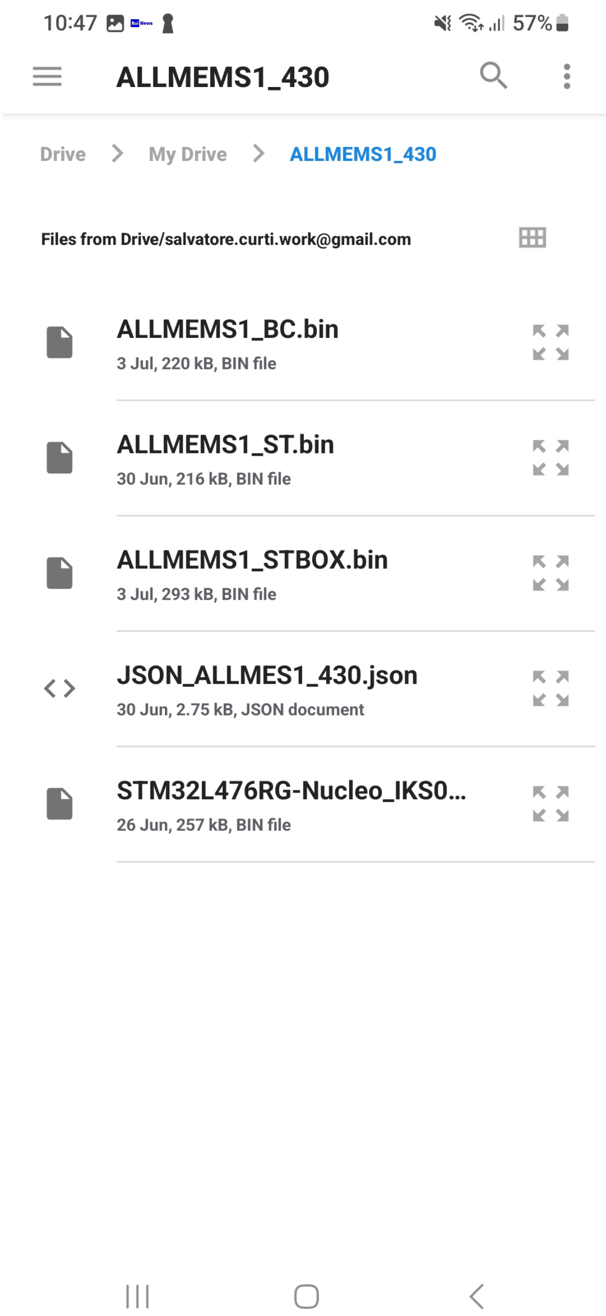
If the 'Firmware upgrade' option menu (see Figure 17. STBLESensor (Android version) options selection) is selected, the following page appears.

Figure 46. STBLESensor (Android version) firmware upgrade page



The STBLESensor application shows which version of the FP-SNS-ALLMEMS1 software is running and the board type. To apply an update, choose the appropriate file and press UPGRADE button.

Figure 47. STBLESensor (Android version) firmware update file selection

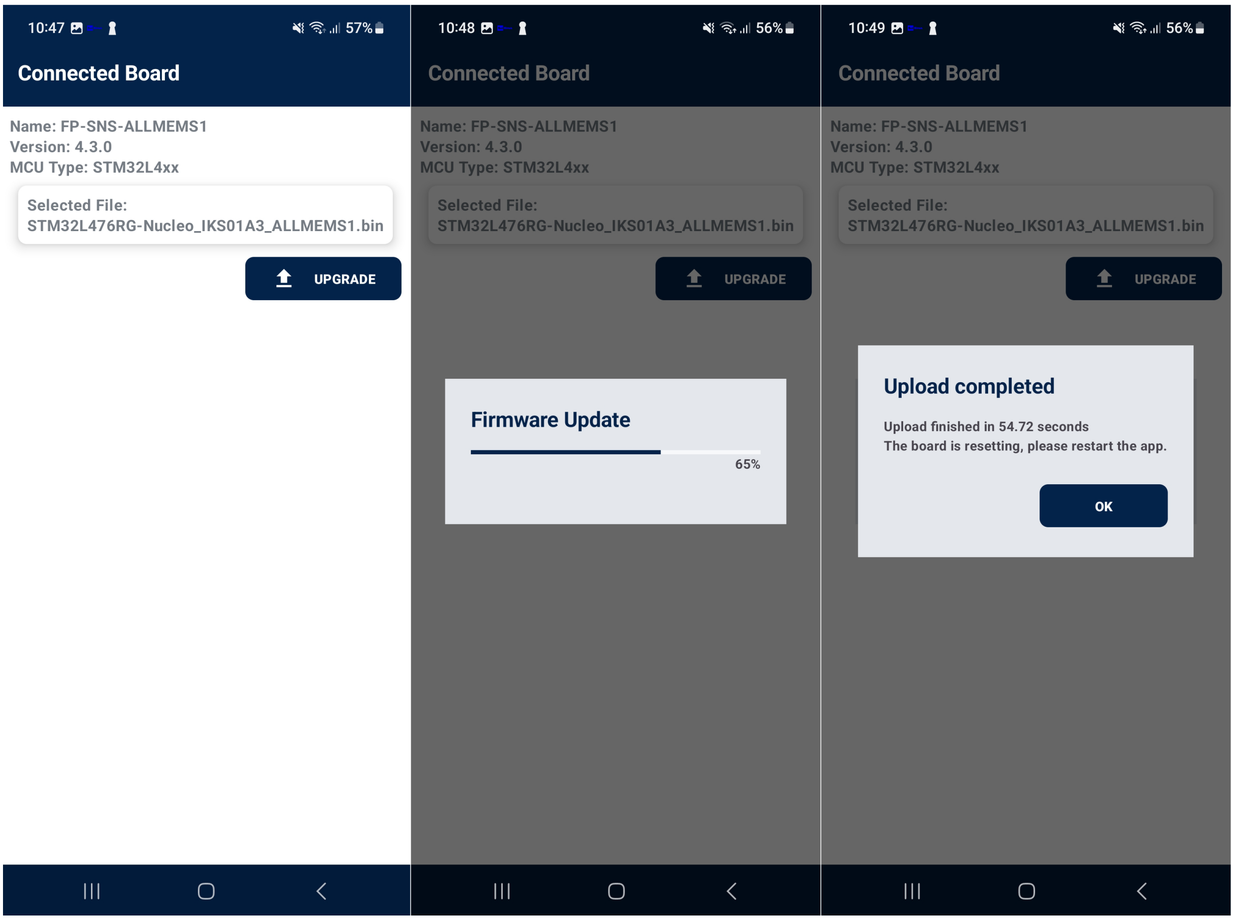


STBLESensor sends the FP-SNS-ALLMEMS1 a command communicating that it is going to send an update of a certain byte size and corresponding CRC value.

Figure 49. Terminal window feedback during FOTA shows the terminal window with the debug information returned during FOTA for an STM32 Nucleo platform when we use a UART to control FP-SNS-ALLMEMS1 behavior.

STBLESensor displays a progress bar during the FOTA procedure, followed by the total upload time on completion.

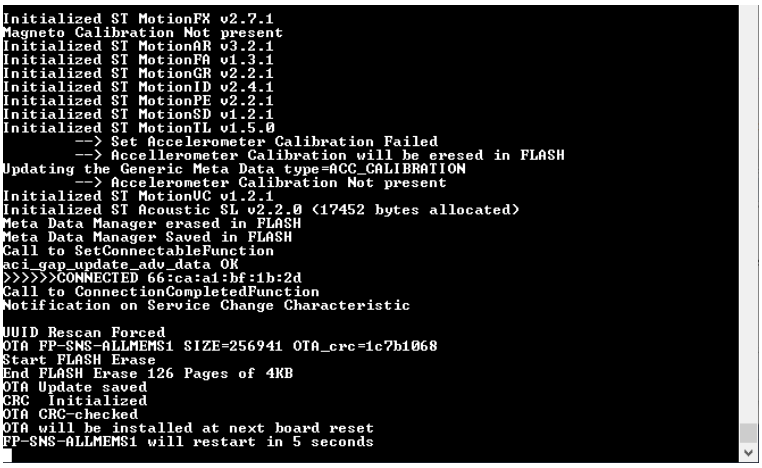
Figure 48. **STBLESensor (Android application) feedback during and after FOTA transmission**



On completion of FOTA transmission, the STM32 uses the CRC hardware unit to compute the CRC value for the FOTA received. If this CRC matches the expected CRC previously sent by the STBLESensor application, FP-SNS-ALLMEMS1 writes a code number to signal the BootLoader there is an OTA ready to be applied.

As the following figure shows, the BootLoader applies the OTA at the next board reboot and executes the new FP-SNS-ALLMEMS1 firmware.

Figure 49. **Terminal window feedback during FOTA**



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.

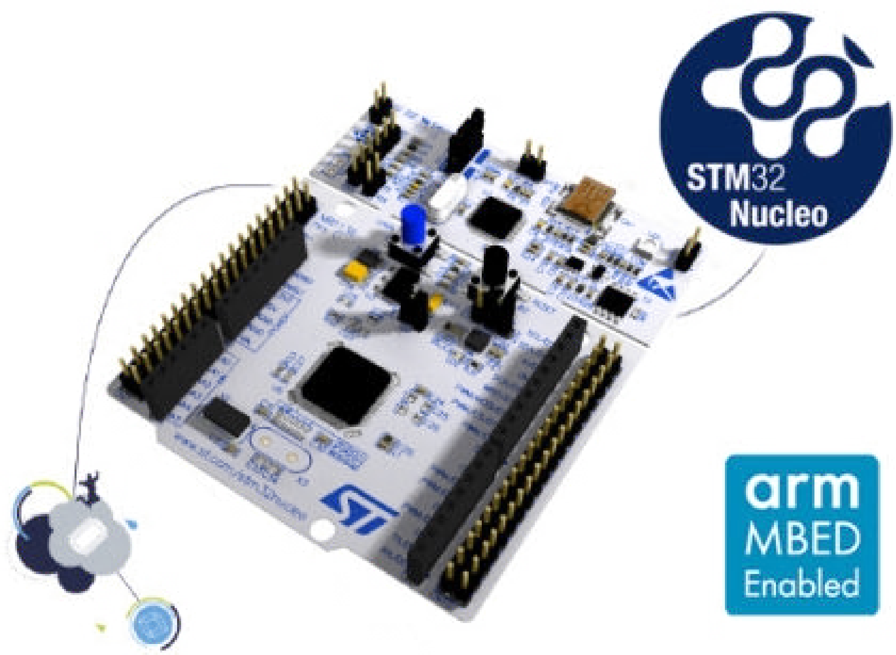
The Arduino connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

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 50. STM32 Nucleo board



2.1.2 X-NUCLEO-CCA02M2 expansion board

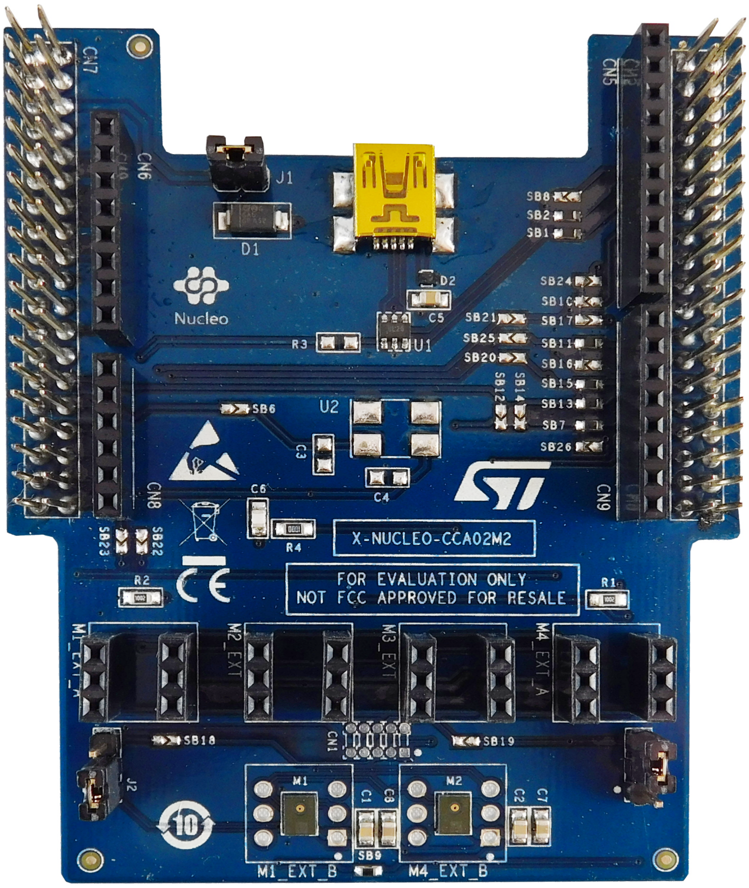
The X-NUCLEO-CCA02M2 expansion board has been designed around MP34DT06J digital MEMS microphone.

It is compatible with the ST morpho connector layout and with digital microphone coupon boards such as STEVAL-MIC001V1, STEVAL-MIC002V1 and STEVAL-MIC003V1.

The X-NUCLEO-CCA02M2 embeds two MP34DT06J microphones and allows synchronized acquisition and streaming of up to 4 microphones through I²S, SPI, DFSDM or SAI peripherals.

It represents a quick and easy solution for the development of microphone-based applications as well as a starting point for audio algorithm implementation.

Figure 51. **X-NUCLEO-CCA02M2 expansion board**



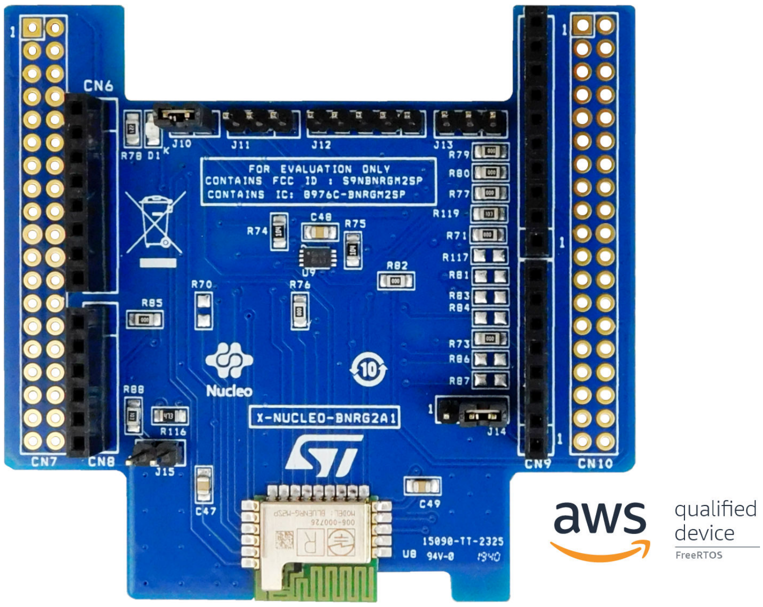
2.1.3 X-NUCLEO-BNRG2A1 expansion board

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 52. **X-NUCLEO-BNRG2A1 BLE expansion board**



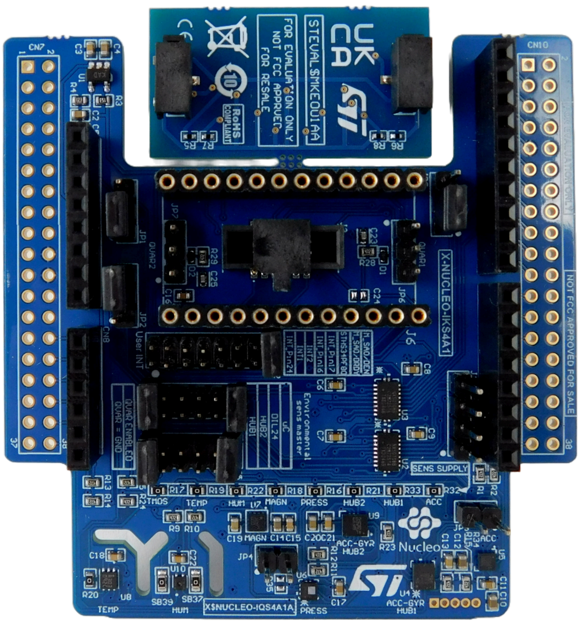
2.1.4 X-NUCLEO-IKS4A1 expansion board

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 3-axis digital accelerometer and a 3-axis digital gyroscope, the LIS2MDL ultra-low-power and high-performance 3-axis digital magnetic sensor, the STTS22H temperature sensor, SHT40AD1B humidity, 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 53. **X-NUCLEO-IKS4A1 expansion board**



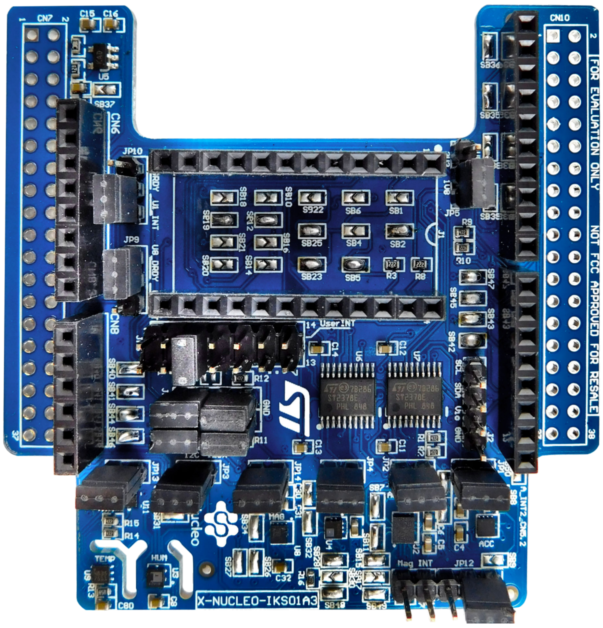
2.1.5 X-NUCLEO-IKS01A3 expansion board

The X-NUCLEO-IKS01A3 is a motion MEMS and environmental sensor evaluation board system.

It is compatible with the Arduino UNO R3 connector layout and features the LSM6DSO 3-axis accelerometer + 3-axis gyroscope, the LIS2MDL 3-axis magnetometer, the LIS2DW12 3-axis accelerometer, the HTS221 humidity and temperature sensor, the LPS22HH pressure sensor, and the STTS751 temperature sensor.

The X-NUCLEO-IKS01A3 interfaces with the STM32 microcontroller via the I²C pin, and it is possible to change the default I²C port.

Figure 54. X-NUCLEO-IKS01A3 MEMS and environmental sensor expansion board



2.1.6 STEVAL-MKSBOX1V1 evaluation kit

The STEVAL-MKSBOX1V1 (SensorTile.box) is a ready-to-use box kit with wireless IoT and wearable sensor platform to help you use and develop apps based on remote motion and environmental sensor data, regardless of your level of expertise.

The SensorTile.box board fits into a small plastic box with a long-life rechargeable battery, and the ST BLE Sensor app on your smartphone connects via Bluetooth to the board and allows you to immediately begin using the wide range of default IoT and wearable sensor applications.

In Expert Mode, you can build customs apps from your selection of SensorTile.box sensors, operating parameters, data and output types, and special functions and algorithms available. This multi sensor kit therefore allows you to design wireless IoT and wearable sensor applications quickly and easily, without performing any programming.

SensorTile.box includes a firmware programming and debugging interface that allows professional developers to engage in more complex firmware code development using the STM32 Open Development Environment (STM32 ODE), which includes a sensing AI function pack with neural network libraries.

Figure 55. STEVAL-MKSBOX1V1 evaluation kit

2.2 Hardware setup

The following hardware components are required:

- for STM32 Nucleo expansion boards:
  - One STM32 Nucleo board (order code: NUCLEO-F446RE or NUCLEO-L476RG)
  - One microphone expansion board (order code: X-NUCLEO-CCA02M2)
  - One sensor expansion board (order code: X-NUCLEO-IKS01A3 or X-NUCLEO-IKS4A1)
  - One BlueNRG Bluetooth low energy expansion board (order code: X-NUCLEO-BNRG2A1)
  - One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
- for STEVAL-BCNKT01V1:
  - STEVAL-BCNKT01V1 development kit
  - ST-LINK/V2-1 debugger/programmer integrated onto STM32 Nucleo board
  - One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
  - One USB type A to Micro-B USB cable to connect the STEVAL-BCNKT01V1 to the PC
- for STEVAL-MKSBOX1V1:
  - STEVAL-MKSBOX1V1 development kit
  - ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32
  - One USB type A to Mini-B USB cable to connect the ST-LINK/V2 in-circuit debugger/programmer to the PC
  - One USB type A to Micro-B USB cable to connect the STEVAL-MKSBOX1V1 to the PC

2.3 Software setup

The following software components are needed to set up a suitable development environment for creating applications for the STM32 Nucleo equipped with the sensors, microphones and Bluetooth low energy expansion boards and for STEVAL-MKSBOX1V1 and STEVAL-BCNKT01V1:

- FP-SNS-ALLMEMS1: Bluetooth low energy and sensors software for STM32Cube.
- Development tool-chain and Compiler. The STM32Cube expansion software supports the three following environments:
  - IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
  - STM32CubeIDE for STM32 + ST-LINK

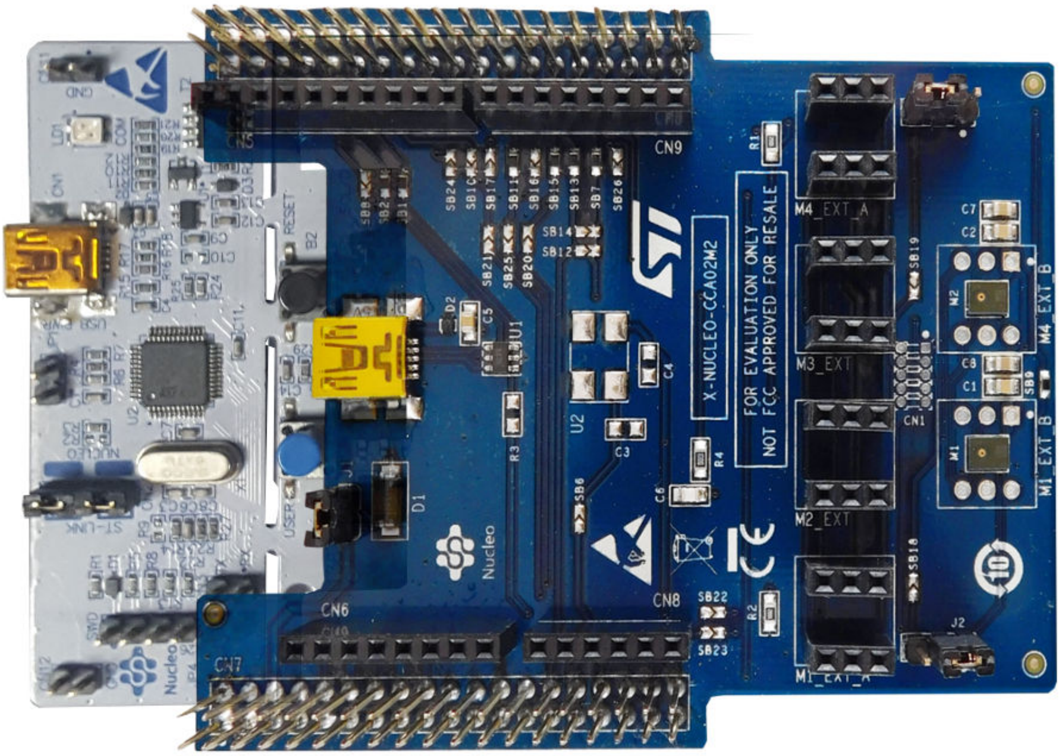
2.4 System setup

2.4.1 STM32 Nucleo and expansion board setup

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

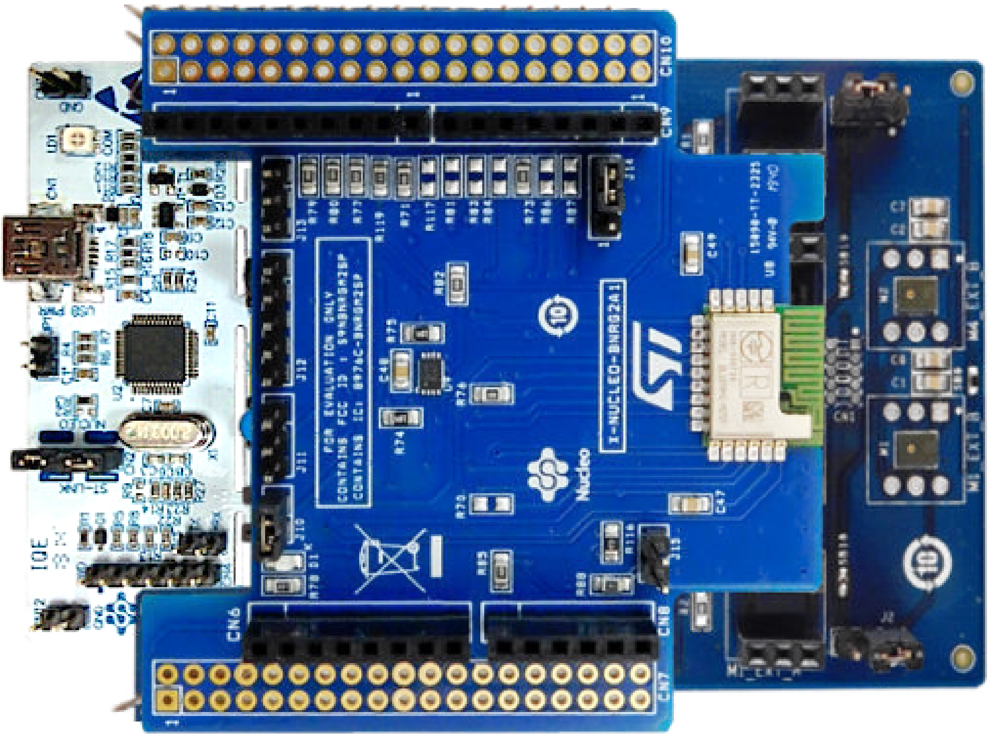
The X-NUCLEO-CCA02M2 sensor board is easily connected to the STM32 Nucleo development board through the morpho connectors, as shown below.

Figure 56. STM32 Nucleo plus X-NUCLEO-CCA02M2 boards



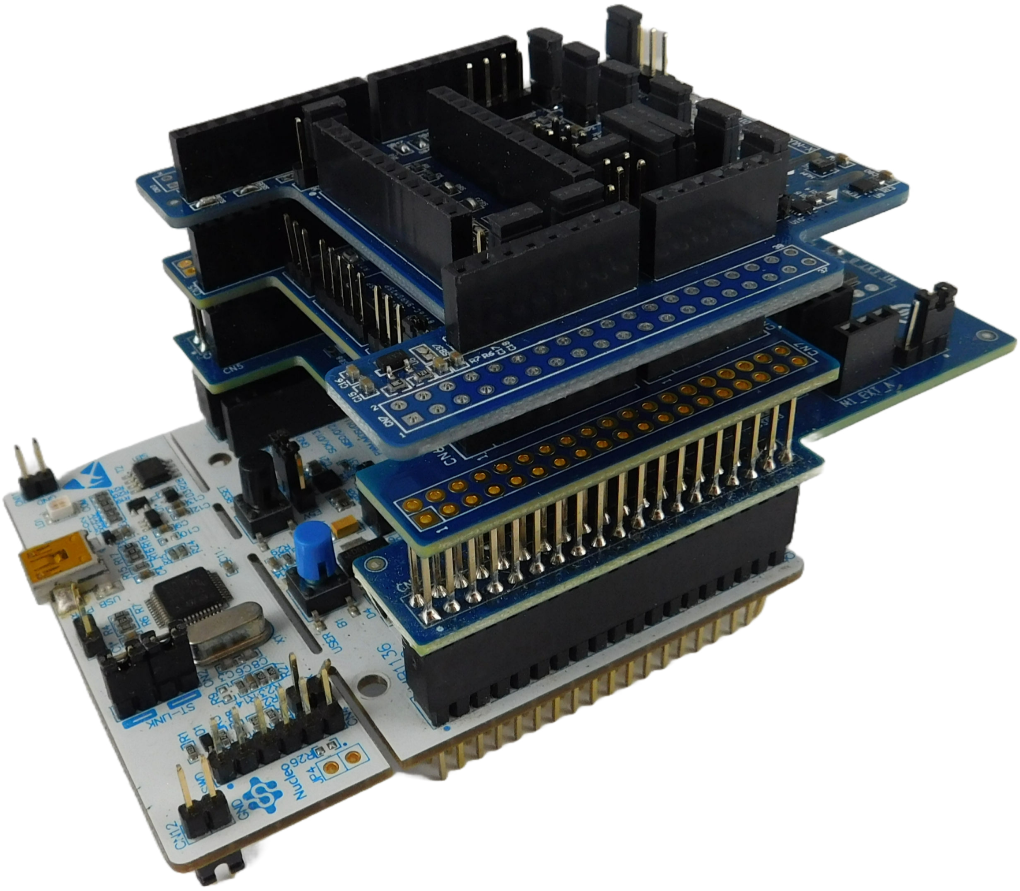
The X-NUCLEO-BNRG2A1 expansion board is connected to the X-NUCLEO-CCA02M2 board through the Arduino UNO R3 extension connector, as shown below.

Figure 57. STM32 Nucleo plus X-NUCLEO-CCA02M2 plus X-NUCLEO-BNRG2A1 boards



The X-NUCLEO-IKS4A1 or X-NUCLEO-IKS01A3 sensor board is then connected to the X-NUCLEO-BNRG2A1 expansion board through the Arduino UNO R3 extension connector, as shown below.

Figure 58. STM32 Nucleo plus X-NUCLEO-CCA02M2 plus X-NUCLEO-BNRG2A1 plus X-NUCLEO-IKS01A3 boards



Note: The stacking sequence indicated above helps to optimize the performance of the BlueNRG-M2SP module on the X-NUCLEO-BNRG2A1 expansion board, reducing interference from its antenna.

2.4.2 STEVAL-BCNKT01V1 setup

The ST-LINK/V2-1 debugger/programmer integrated in the STM32 Nucleo board must be used to program the STEVAL-BCNCS01V1 (BlueCoin).

To program the board, connect STEVAL-BCNCS01V1 (BlueCoin) on the STEVAL-STLCX01V1 BlueCoin Coinstation board.

Use the SWD connector (a 5-pin flat cable is provided in the BlueCoin Kit package) to connect the BlueCoin Coinstation board to ST-LINK/V2-1 debugger/programmer integrated on the STM32 Nucleo board for programming.

Be sure that CN2 Jumpers are OFF and connect your STM32 Nucleo board to the BlueCoin Coinstation through the provided cable paying attention to the polarity of the connectors. Pin 1 can be identified by a small circle on the STM32 Nucleo board and Coin Station PCB silkscreens or by the square shape of the soldering pad of the connector (SensorTile Cradle).

Figure 59. BlueCoin-STM32 Nucleo: SWD connections with 5-pin flat cable

2.4.3 STEVAL-MKSBOX1V1 setup

To program the STEVAL-MKSBOX1V1 (SensorTile.box) you need an ST-LINK/V2-1 debugger/programmer (integrated in the B-L475E-IOT01AIoT Discovery kit node board).

Step 1. Connect the STEVAL-MKSBOX1V1 to ST-LINK/V2-1 using the flat cable

Step 2. Connect the ST-LINK/V2-1 to the PC using one USB type A to Mini-B USB cable

Step 3. Power the STEVAL-MKSBOX1V1 by plugging the battery or using a USB type A Micro-B USB cable connected to the PC

2.4.4 X-NUCLEO-BNRG2A1 setup

The procedure described below is required to make the BlueNRG-2 library correctly work with the firmware embedded in the X-NUCLEO-BNRG2A1 Bluetooth® Low Energy module.

Step 1. Solder a 0 Ohm resistor to R117 on the X-NUCLEO-BNRG2A1 expansion board.

Step 2. Update the X-NUCLEO-BNRG2A1 Bluetooth® Low Energy module through the ST-LINK/V2-1 and the STSW-BNRGFLASHER flasher utility.

Step 3. Connect the X-NUCLEO-BNRG2A1 J12 pins to the ST-LINK/V2-1 pins through the 5-wire cable included as shown in the picture below and detailed in Table 1.

Figure 60. **Connection between X-NUCLEO-BNRG2A1 expansion board and ST-LINK/V2-1**

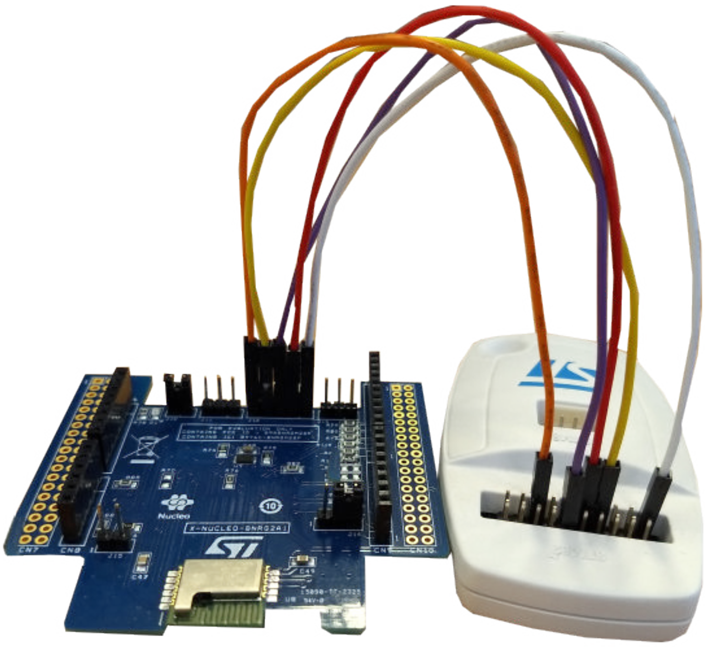


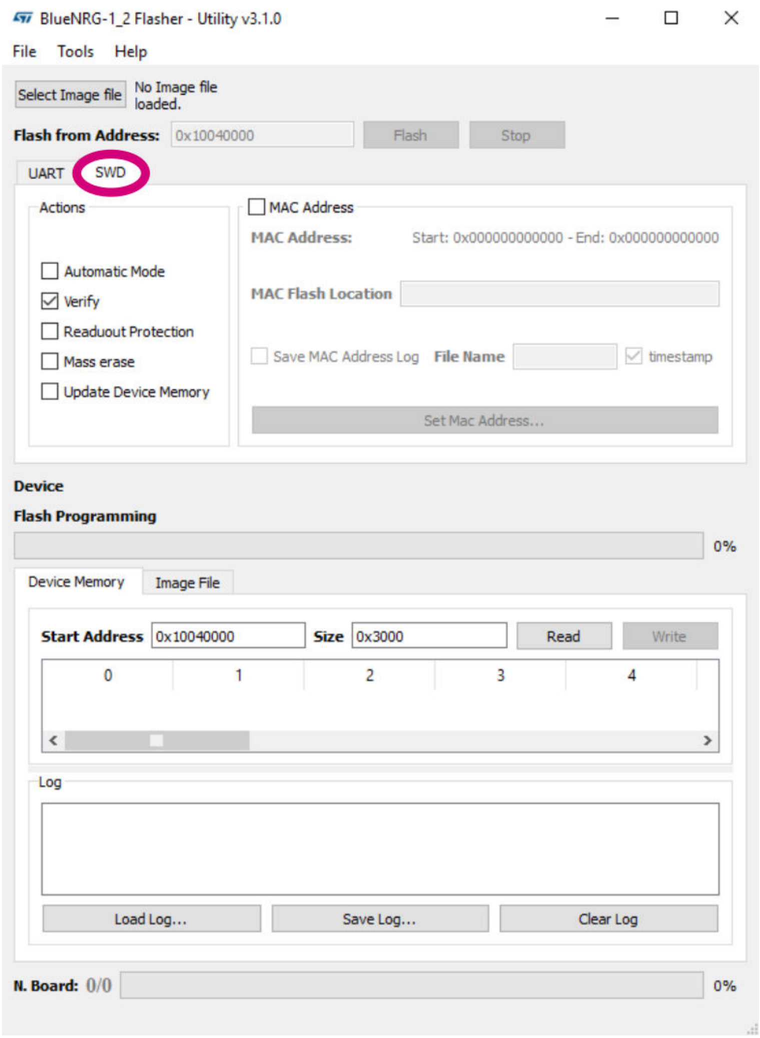
Table 1. **X-NUCLEO-BNRG2A1 and ST-LINK/V2-1 pin connections**

X-NUCLEO-BNRG2A1 J12 pin no.	ST-LINK/V2-1 pin no.
1	1
2	9
3	12
4	7
5	15

Step 4. Install the STSW-BNRGFLASHER and open it.

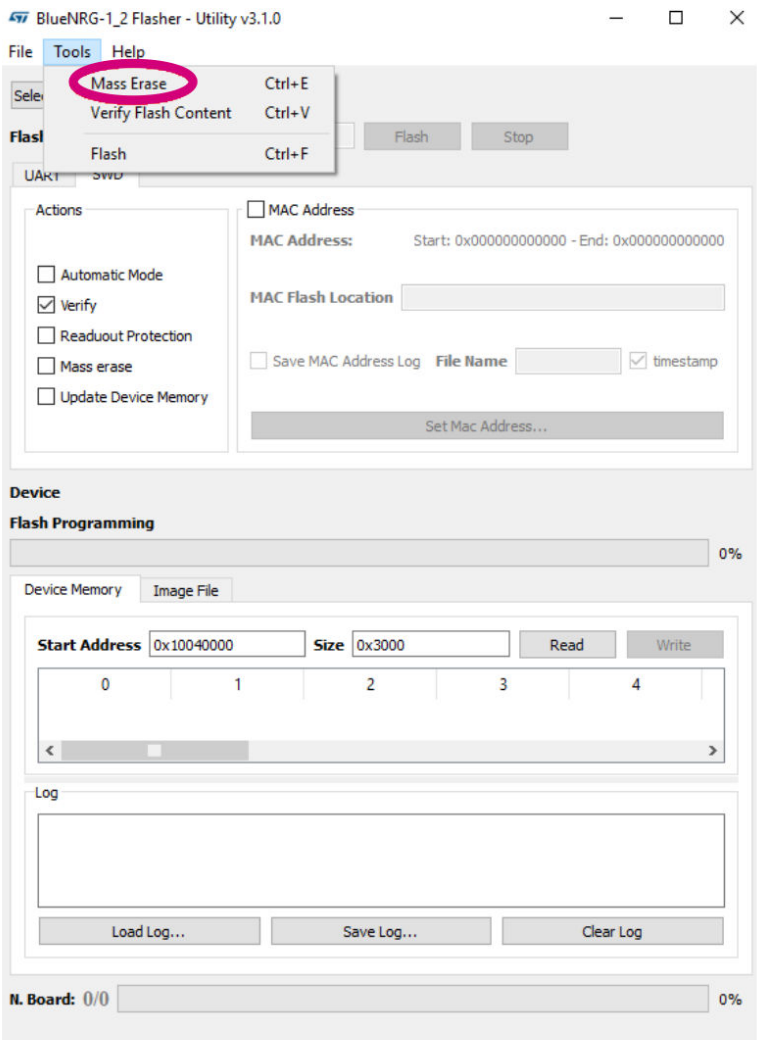
Step 5. Select the SWD tab.

Figure 61. STSW-BNRGFLASHER - SWD tab



Step 6. Erase the BlueNRG-2 flash memory.

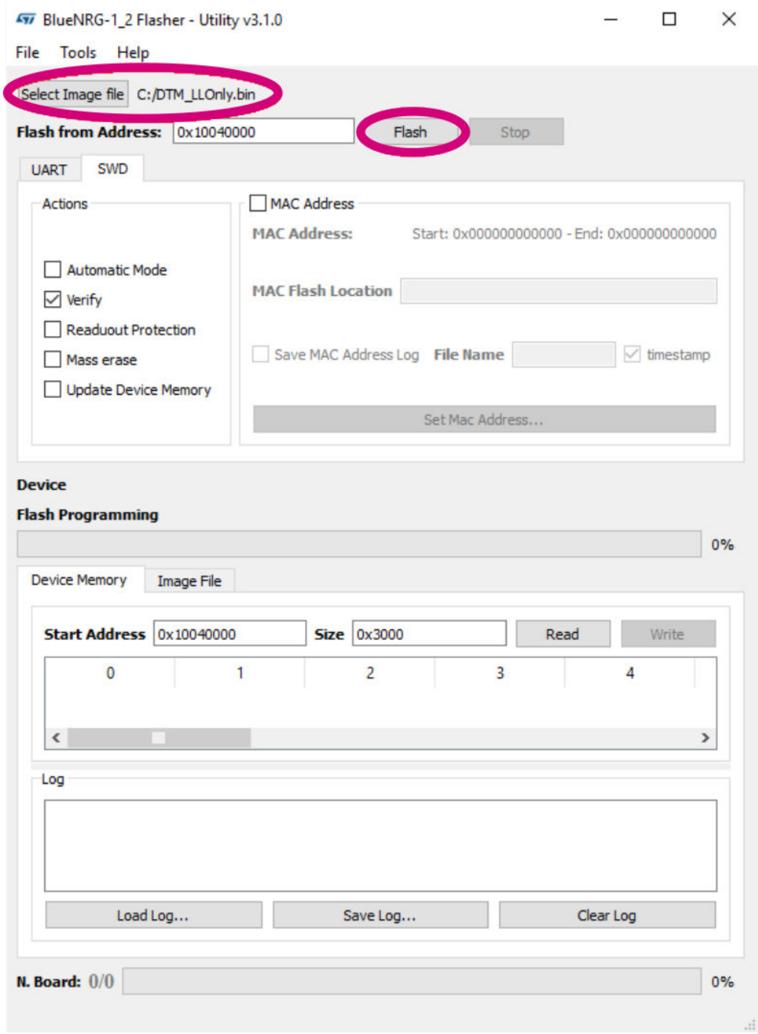
Figure 62. STSW-BNRGFLASHER - mass erase



Step 7. Download the Link Layer Only firmware for the Bluetooth® Low Energy module.

Step 8. Load the Link Layer Only firmware into the STSW-BNRGFLASHER and press the [**Flash**] button.

Figure 63. STSW-BNRGFLASHER - Flash button



Step 9. To restore the X-NUCLEO-BNRG2A1 Bluetooth® Low Energy module embedded firmware, repeat the procedure using the following firmware image: DTM_Full.bin.

In case of issues during the update process, close the X-NUCLEO-BNRG2A1 J15 jumper before repeating the procedure.

2.4.5 Important additional hardware information

For F4xx STM32 Nucleo development board, there is an hardware conflict among the X-NUCLEO-IKS4A1/X-NUCLEO-IKS01A3 expansion boards and the X-NUCLEO-CCA02M2 expansion board through the Arduino UNO R3 extension connector.

The hardware conflict is on pins 5, 6 and 7 of the CN9 Arduino connector.

- For X-NUCLEO-IKS4A1:
  - In the pin 5 (SB8) and 6 (SB10) there are the interrupts INT2 and INT1 for LSM6DSV16X component (used for the feature hardware)
  - In the pin 7 (SB11) there is the interrupt for LPS22DF component
- For X-NUCLEO-IKS01A3:
  - in pins 5 (SB43) and 6 (SB45) there are the interrupts INT1 and INT2 for LSM6DSO (used for the featured hardware)
  - in pin 7 (SB47) there is the interrupt for LPS22HB component
- For X-NUCLEO-CCA02M2:
  - in pin 5 (SB7) there is the clock for the microphones
  - in pin 6 (SB15) there is the clock x2 for the microphones
  - in pin 7 (SB17) there is the connection for the microphone (solder bridge open as default)

For this reason the hardware features are not available for NUCLEO-F446RE (the related GPIO pin is not configured).

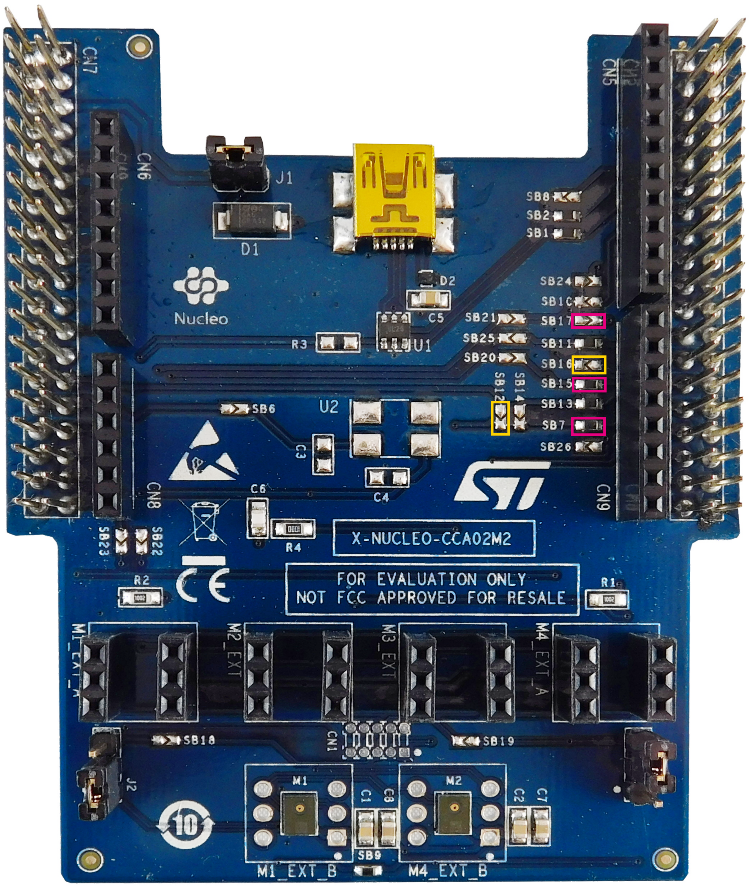
For NUCLEO-L476RG:

- before connecting the X-NUCLEO-CCA02M2 to the STM32 Nucleo L4-series development board through the ST morpho connector layout, on the X-NUCLEO-CCA02M2 board::
  - close the solder bridges SB12, SB16 and open the solder bridges SB7, SB15 and SB17 (to enable the signal clock from STM32 Nucleo L4-series development board)
  - if additional microphones are plugged to the board, close solder bridge SB17
- before connecting the X-NUCLEO-IKS4A1 expansion board to the X-NUCLEO-CCA02M2 expansion board:
  - remove the solder bridge SB25 if additional microphones are plugged to the X-NUCLEO-CCA02M2 board
- before connecting the X-NUCLEO-IKS01A3 expansion board to the X-NUCLEO-CCA02M2 expansion board:
  - remove solder bridge SB47 if additional microphones are plugged to the X-NUCLEO-CCA02M2 board

Note: For the NUCLEO-L476RG board only: before connecting the X-NUCLEO-CCA02M2 board to the STM32 Nucleo L4-series development board through the ST morpho connector layout, on the X-NUCLEO-CCA02M2 board:

- *close SB12 and SB16 solder bridges (in yellow in the figure below)*
- *open SB7, SB15 and SB17 solder bridges (in pink in the figure below)*

Figure 64. X-NUCLEO-CCA02M2 solder bridge configuration for the NUCLEO-L476RG board



Revision history

Table 2. Document revision history

Date	Version	Changes
18-Apr-2016	1	Initial release.
02-Aug-2016	2	Text and formatting changes throughout document Added STEVAL-STLKT01V1 board use details Added Section 1.4: "Flash management" Added Section 1.5: "The boot process" Added Section 1.6: "Firmware-Over-The-Air (FOTA) update" Added Section 1.10: "Firmware-Over-The-Air update with BlueMS" Added Section 2.1.6: "STEVAL-STLKT01V1 SensorTile development kit" Added Section 2.3.3.2: "STEVAL-STLKT01V1 setup"
18-Oct-2016	3	Throughout document: - text, formatting and graphics enhancements - added STEVAL-STLKT01V1 voice communication information - added STEVAL-STLKT01V1 Gas Gauge information Updated Section "Introduction" Updated Section 1.1: "Overview" Updated Figure 1: "FP-SNS-ALLMEMS1 software architecture" Added Section 1.6: "The Installation process"
21-Dec-2016	4	Throughout document: - minor text and formatting changes Added: Section 2.1.6: "X-NUCLEO-IKS01A2 expansion board" and Figure 41: "X-NUCLEO-IKS01A2 solder bridge configuration" Updated: Figure 1: "FP-SNS-ALLMEMS1 software architecture", Figure 6: "Binary folder content", Figure 7: "Content of a project folder", Figure 8: "BootLoader and ALLMEMS1 installation", Figure 9: "ALLMEMS1 Dump process", Figure 11: "Initialization phase", Figure 12: "UART console output when a device is connected to the board", Figure 22: "BlueMS (Android version) firmware upgrade page", Figure 23: "BlueMS (Android version) firmware update file selection", Figure 24: "BlueMS (Android application) feedback during and after FOTA transmission", Figure 25: "Terminal window feedback during FOTA", Figure 40: "XNUCLEO- IKS01A1 solder bridge configuration", and Figure 42:

Date	Version	Changes
		"X-NUCLEO-CCA02M1 solder bridge configuration for the NUCLEO-L476RG board"
08-May-2017	5	Added references to STM32F446 processor throughout document. Updated: title, Section "Introduction", Section 1.1: "Overview", Section 1.2: "Architecture", Section 1.3: "Folder structure", Section 1.4: "Flash management", Section 1.6: "The installation process", Section 1.9: "Sample application description", Section 1.10.1: "Main page", Section 1.10.3: "Plot data", Section 1.10.4: "Serial and debug console", Section 1.10.5: "Enable hardware features", Section 1.10.13: "Rssi and battery" and Section 1.11: "Firmware-Over-The-Air update with BlueMS". Added: Section 1.10.2: "MEMS sensor fusion", Section 1.10.6: "Activity recognition", Section 1.10.7: "Carry position", Section 1.10.8: "Gesture recognition", Section 1.10.9: "E-compass", Section 1.10.11: "Direction of Arrival", Section 1.10.10: "BlueVoiceADPCM", Section 1.10.10.1: "ASR language selection", Section 1.10.10.1.1: "Chinese ASR: iFlyTek MSC service", Section 1.10.10.1.2: "Alternative languages: Google Speech API" and Section 1.10.10.2: "Google speech ASR Key generation".
26-Jul-2017	6	In Section 1.1: "Overview": - added BlueMS version information associated with Source Localization and Beam Forming. In Section 1.4: "Flash management": - added STM32F446RE Flash address information for the Meta Data Manager Updated Figure 4: "BootLoader folder content" Updated Figure 46: "BlueMS (Android version) audio source localization STM32 Nucleo Page"
19-Oct-2017	7	Updated Introduction, Section 1.1 Overview, Section 1.2 Architecture, Section 1.3 Folder structure, Section 1.9 Sample application description, Section 1.10 Android and iOS sample client application, Section 1.10.14 Rssi and battery, Section 1.11 Firmware-Over-The-Air update with BlueMS and Section 1.10.5 Enable hardware features. Added Section 1.10.4.1 SD card data logging
01-Feb-2018	8	Updated Section 1.1 Overview, Section 1.2 Architecture, Section 1.9 Sample application description, Section 1.10 Android and iOS sample client application and Section 1.10.4 Settings, serial and debug console. Added Section 1.10.17 SD Logging.
03-Oct-2019	9	Updated Introduction, Section 1.1 Overview, Section 1.2 Architecture, Section 1.3 Folder structure, Section 1.4 Flash management, Section 1.6 The installation process, Section 1.9 Sample application description, Section 1.10 Android and iOS sample client application, Section 1.10.5 Enable hardware features, Section 2.2 Software description, Section 2.3.1 Hardware setup, Section 2.3.2 Software setup, Section 2.3.3.1 STM32 Nucleo and expansion boards setup and Section 2.3.3.5 Important additional hardware information. Added Section 1.10.9 Intensity detection, Section 1.10.10 Motion algorithms, Section 1.10.11 Fitness activities, Section 1.10.12 Tilt measurement, Section 2.1.5 X-NUCLEO-IKS01A3 expansion board and Section 2.1.7 STEVAL-MKSBOX1V1 evaluation kit.

Date	Version	Changes
09-Jun-2020	10	Updated Section 1.1 Overview, Section 1.2 Architecture, Section 1.6 The installation process, Section 1.9 Sample application description, Figure 13. STBLESensor (Android version) main page following BLE connection, Section 1.10.2 MEMS sensor fusion, Section 1.10.3 Plot data, Section 1.10.4 Settings, serial and debug console, Section 1.10.5 Enable hardware features, Figure 24. STBLESensor (Android version) MotionAR activity recognition page, Section 1.10.7 Carry position , Figure 26. STBLESensor (Android version) MotionGR gesture recognition page, Figure 34. STBLESensor (Android version) level page, Figure 36. STBLESensor (Android version) MotionFX page, Section 1.10.14 BlueVoiceADPCM voice, Section 1.10.15 Direction of Arrival, Section 1.10.16 Beam Forming, Figure 45. STBLESensor (Android version) Battery and RSSI information, Section 1.11 Firmware-Over-The-Air update with STBLESensor, Section 2.4.1 STM32 Nucleo and expansion board setup, and Section 2.4.6 Important additional hardware information. Added Section 1.10.14.1 Speech to text, Section 2.1.2 XNUCLEO-CCA02M2 expansion board, Section 2.1.3 XNUCLEO-BNRG2A1 expansion board and Section 2.4.5 XNUCLEO-BNRG2A1 setup.
07-Mar-2022	11	Updated introduction, *Section 1.1 Overview, Section 1.6 The installation process, Section 1.9 Sample application description, Section 1.10.4.1 SD card data logging,* and *Section 1.10.17 SD Logging.*
20-Sep-2023	12	Updated introduction in cover page, Section 1.1 Overview, Section 1.2 Architecture, Section 1 FP-SNS-ALLMEMS1 software description, Section 2.1 Hardware description. Updated Section 2.1 Hardware description, Section 2.3 Software setup, Section 2.4 System setup and Section 2.4.5 Important additional hardware information. Removed section "X-NUCLEO-IKS01A2 expansion board", section "STEVAL-STLKT01V1 SensorTile development kit", and section "STEVAL-STLKT01V1 setup". Minor text changes.

Contents

List of tables

List of figures