Getting started with the STM32Cube function pack for ultra-low power IoT node with BLE connectivity, digital microphone, environmental and motion sensors

Introduction

FP-SNS-ALLMEMS2 is an STM32Cube function pack which lets you connect your IoT node to a smartphone via BLE and use a suitable Android™ or iOS™ application, like the BlueMS app, to view real-time environmental and motion sensor data, digital microphone and battery levels.

It provides features similar to the FP-SNS-ALLMEMS1 function pack, but with a different implementation for ultra-low power consumption.

The package also enables advanced functions such as voice communication over BLE, 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, audio data logging and MEMS sensor data logging on SD card.

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, where ultra-low power consumption is a key requirement.

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-STLKT01V1 evaluation boards.

1 FP-SNS-ALLMEMS2 software description

1.1 Overview

The key features of the FP-SNS-ALLMEMS2 package are:

- Complete firmware to develop an IoT node with BLE connectivity, digital microphone, environmental and motion sensors
- Middleware libraries for sensor data fusion and accelerometer-based real-time activity recognition, acoustic source localization and beam forming, audio processing and streaming over BLE communication profile, and SD card data logging
- Compatible with BlueMS application for Android/iOS, to perform sensor data reading, audio and motion algorithm feature demo, and firmware update over the air (FOTA)
- Implementation based on the use of an RTOS, targeting ultra-low power consumption use cases with battery operated devices
- Sample implementation available for STEVAL-BCNKT01V1 and STEVAL-STLKT01V1 evaluation boards and for X-NUCLEO-CCA02M1, X-NUCLEO-IKS01A2 and X-NUCLEO-IDB05A1 connected to a NUCLEO-F446RE or NUCLEO-L476RG board
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

This software creates the Bluetooth services listed below:

1. hardware characteristics related to MEMS sensor devices:
   - temperature
   - pressure
   - humidity
   - 3D gyroscope, 3D magnetometer, 3D accelerometer
   - microphones dB noise level.
   - battery %, voltage and status (charging/discharging/low battery) for STEVAL-STLKT01V1 and STEVAL-BCNKT01V1
2. software characteristics:
   - quaternions generated by the MotionFX library
   - magnetic north direction (e-Compass)
   - recognized activity using the MotionAR algorithm
   - recognized carry position using the MotionCP algorithm (feature not available on NUCLEO-F446RE and STEVAL-BCNKT01V1)
   - recognized gesture using the MotionGR algorithm (feature not available on NUCLEO-F446RE and STEVAL-BCNKT01V1)
   - audio source localization using the AcousticSL algorithm (feature not available on the STEVAL-STLKT01V1)
   - audio beam forming using the AcousticBF algorithm (feature not available on STEVAL-STLKT01V1 and NUCLEO-L476RG)
   - voice over Bluetooth low energy using the BlueVoiceADPCM algorithm
   - SD data logging (audio and MEMS data) using Generic FAT File System middleware (feature available on the STEVAL-STLKT01V1 only)
3. Console service:
   - 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. a service to reset/transmit the calibration status and enable the following expansion hardware features for LSM6DSL on X-NUCLEO-IKS01A2 expansion board for STM32 Nucleo L4 only, or for LSM6DSM motion sensor for STEVAL-BCNKT01V1 and STEVAL-STLKT01V1:
   - 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, audio and motion sensor drivers for the HTS221, LPS22HB, MP34DT01-M, LSM6DSL and LSM303AGR devices if X-NUCLEO-IKS01A2 expansion board is mounted on the STM32 Nucleo
- the temperature, pressure, audio, motion sensor and gas gauge IC drivers for the LPS22HB, MP34DT04, LSM6DSM, LSM303AGR and STC3115 devices for STEVAL-STLKT01V1 running 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 package is compatible with the BlueMS Android/iOS (Ver. 3.0.0 or higher) applications, available at the respective store, which can be used to display information sent via BLE.

The BlueMS application allows Over-The-Air firmware update.

BlueMS Version 3.2.0 or higher is required to show battery information for STEVAL-STLKT01V1.

BlueMS Version 3.5.0 or higher is required to show the source localization and beam forming pages.

BlueMS Version 3.6.0 or higher is required to show the multi-events detection and the battery information for STEVAL-BCNKT01V1.

BlueMS Version 3.7.0 or higher is required to show the data logging setting on the SD card for the STEVAL-STLKT01V1.

1.2 Architecture

The STM32Cube function packs leverage the modularity and interoperability of STM32 Nucleo and X-NUCLEO boards with STM32Cube and X-CUBE 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 X-CUBE packages, thus enabling new functions and creating more pertinent and usable systems for developers.

STMCube™ is designed by STMicroelectronics to reduce development effort, time and cost across the entire STM32 portfolio.

STM32Cube version 1.x includes:

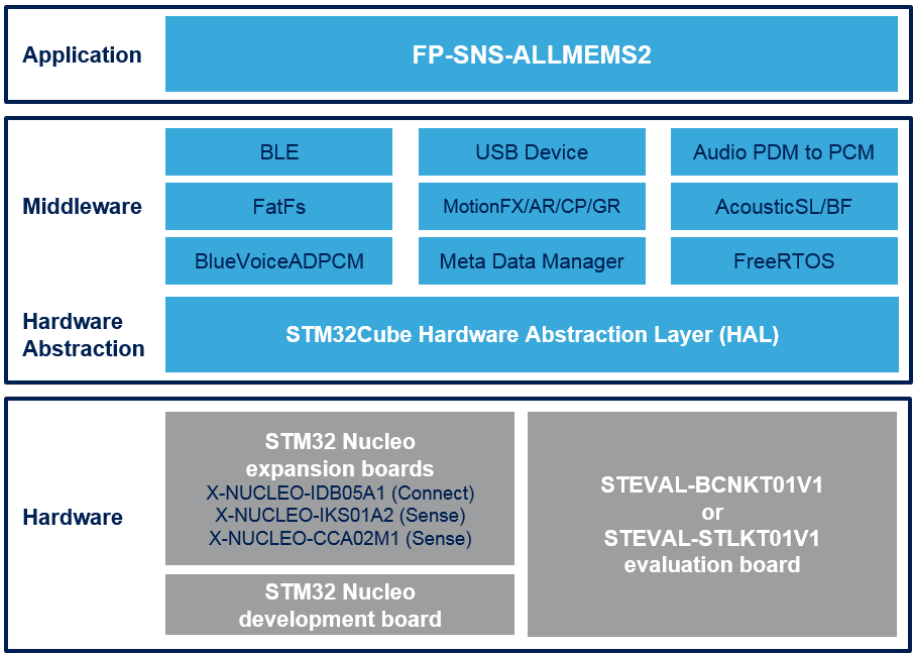
- STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards.
- A comprehensive embedded software platform specific to each series (such as the STM32Cube for the STM32 series), which includes:
  - the STM32Cube HAL embedded abstraction-layer software, ensuring maximized portability across the STM32 portfolio
  - a consistent set of middleware components such as RTOS, USB, TCP/IP and graphics
  - all embedded software utilities with a full set of examples

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.

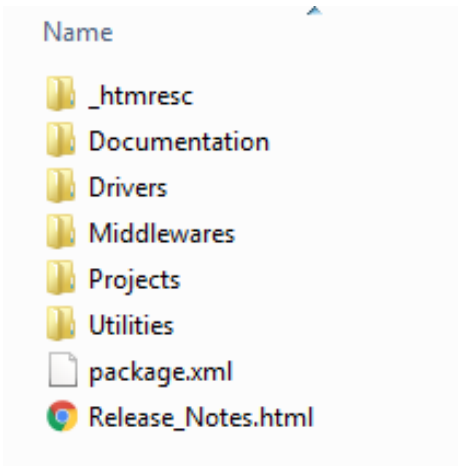
- **Board support package (BSP) layer**: supports the peripherals on the STM32 Nucleo board (except the MCU) with a limited set of APIs providing a programming interface for certain board-specific peripherals like the LED, the user button, etc., and helps determine the specific board version. For the 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-ALLMEMS2 software architecture



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, Generic FAT File System Module (FatFs), FreeRTOS real time operating system, PDM signal decoding and audio signal reconstruction when connecting an ST MEMS, the Meta Data Manager, MotionFX (iNEMOEngine PRO) sensor fusion library, MotionAR (iNEMOEngine PRO) activity-recognition library, MotionCP (iNEMOEngine PRO) carry-position recognition library, MotionGR (iNEMOEngine PRO) gesture recognition 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-STLKT01V1 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 the STM32F446RE and STM32L476RG Nucleo boards.

1.4 Flash management

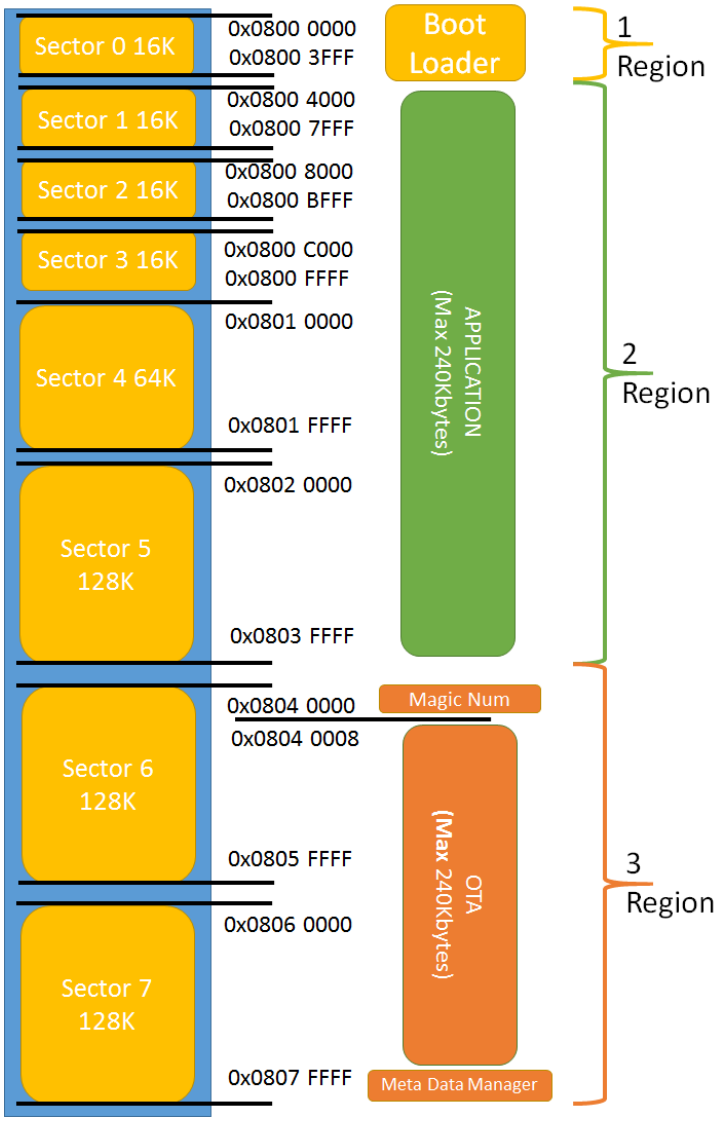
Apart from storing code, FP-SNS-ALLMEMS2 uses the Flash memory for Firmware-Over-The-Air updates. It is divided into the following regions (see figure below):

1. the first region contains a custom boot loader
2. the second region contains the FP-SNS-ALLMEMS2 firmware
3. The third region is used for storing the FOTA before the update

Even if the STM32F446RE (512 KB) and the STM32L476RG (1024 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 and 0x080FF000 for STM32L476RG). 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

Figure 3. **FP-SNS-ALLMEMS2 FLASH structure**



1.5 The boot process

The FP-SNS-ALLMEMS2 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) thus: `#define VECT_TAB_OFFSET 0x4000.`

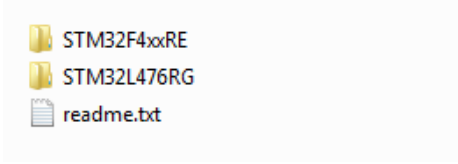
We also changed the linker script. For example, the Linker script for FP-SNS-ALLMEMS2 running on STM32F446RE and compiled using IAR Embedded Workbench for ARM is:

```
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 or STM32L476RG, found in the Utilities \BootLoader folder, to the first FLASH region (address 0x08000000).

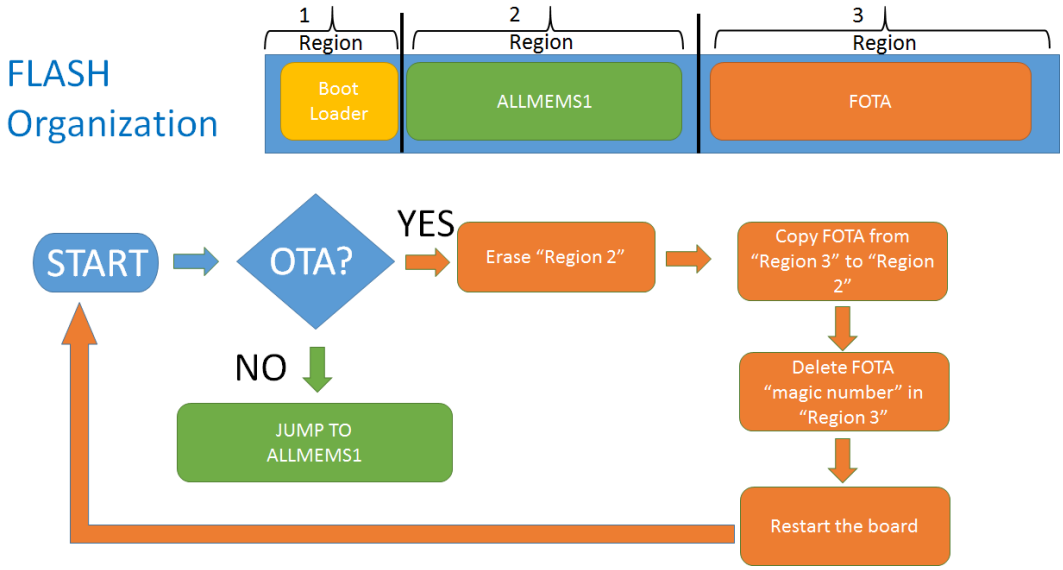
Figure 4. BootLoader folder content



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-ALLMEMS2 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-ALLMEMS2 firmware.

Figure 5. FP-SNS-ALLMEMS2 boot sequence



1.6 The installation process

The package Binary directory contains an image (in .bin and .hex format) for each platform (NUCLEO-F446RE, NUCLEO-L476RG, STEVAL-BCNKT01V1, STEVAL-STLKT01V1), including:

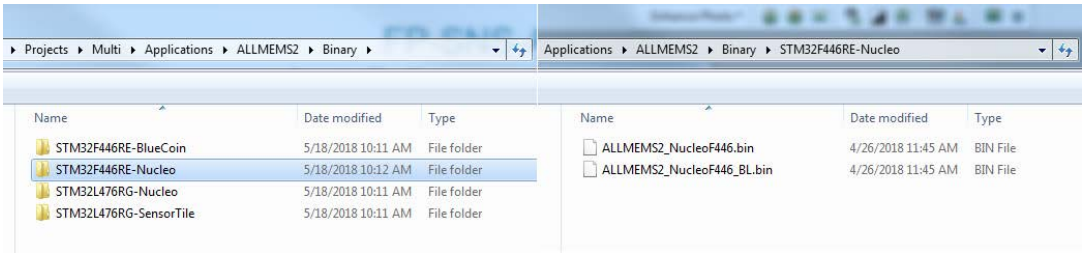
- pre-compiled ALLMEMS2 firmware that may be flashed with ST-LINK to the correct memory address (0x08004000) of a supported STM32 Nucleo development board or SensorTile board

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

- pre-compiled ALLMEMS2 plus BootLoader firmware that may be directly flashed to a supported STM32 Nucleo development board or SensorTile board with the ST-LINK or via a Drag & Drop operation (STM32 Nucleo boards only)

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

Figure 6. **Binary folder content**

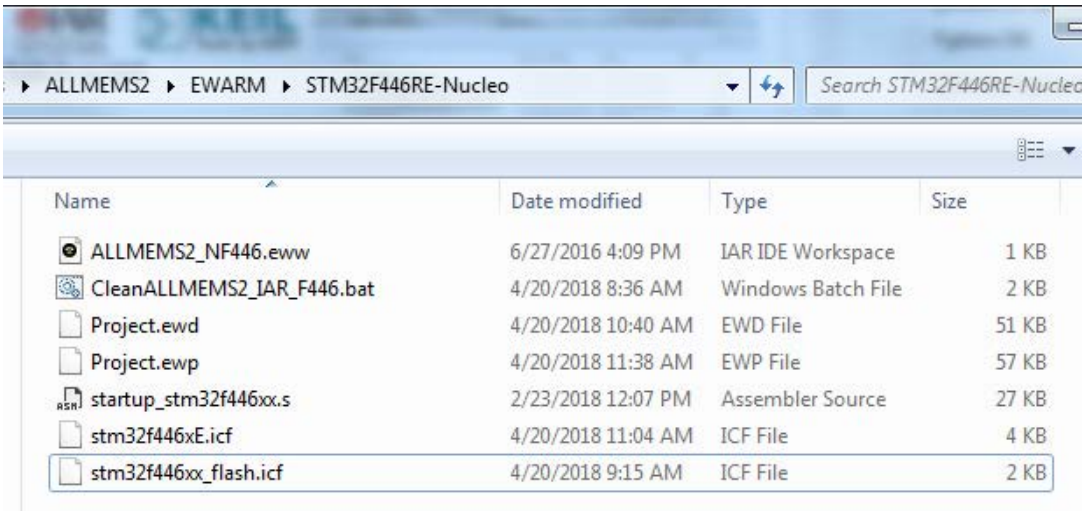


To flash modified ALLMEMS2 firmware, simply flash the compiled FP-SNS-ALLMEMS2 firmware to the correct address (0x08004000).

A batch script has been provided to simplify this operation by saving the firmware and the BootLoader to the right position; it is available for each platform (NUCLEO-F446RE, NUCLEO-L476RG, STEVAL-BCNKT01V1 and STEVAL-STLKT01V1) and for each IDE (IAR/RealView/System Workbench):

- IAR toolchain Embedded Workbench V8.20.2:
  - For Nucleo F446: CleanALLMEMS2_IAR_F446.bat
  - For Nucleo L476: CleanALLMEMS2_IAR_L476.bat
  - For STEVAL-BCNKT01V1: CleanALLMEMS2_IAR_BC.bat
  - For STEVAL-STLKT01V1: CleanALLMEMS2_IAR_ST.bat
- System Workbench for STM32 Version 2.4.0.201801120948:
  - For Nucleo F446: CleanALLMEMS2_SW4STM32_F446.bat
  - For Nucleo L476: CleanALLMEMS2_SW4STM32_L476.bat
  - For STEVAL-BCNKT01V1: CleanALLMEMS2_SW4STM32_BC.bat
  - For STEVAL-STLKT01V1: CleanALLMEMS2_SW4STM32_ST.bat
- µVision toolchain - MDK-ARM Professional Version: 5.24.2:
  - For Nucleo F446: CleanALLMEMS2_MDK_ARM_F446.bat
  - For Nucleo L476: CleanALLMEMS2_MDK_ARM_L476.bat
  - For STEVAL-BCNKT01V1: CleanALLMEMS2_MDK_ARM_BC.bat
  - For STEVAL-STLKT01V1: CleanALLMEMS2_MDK_ARM_ST.bat

Figure 7. **Content of a project folder**

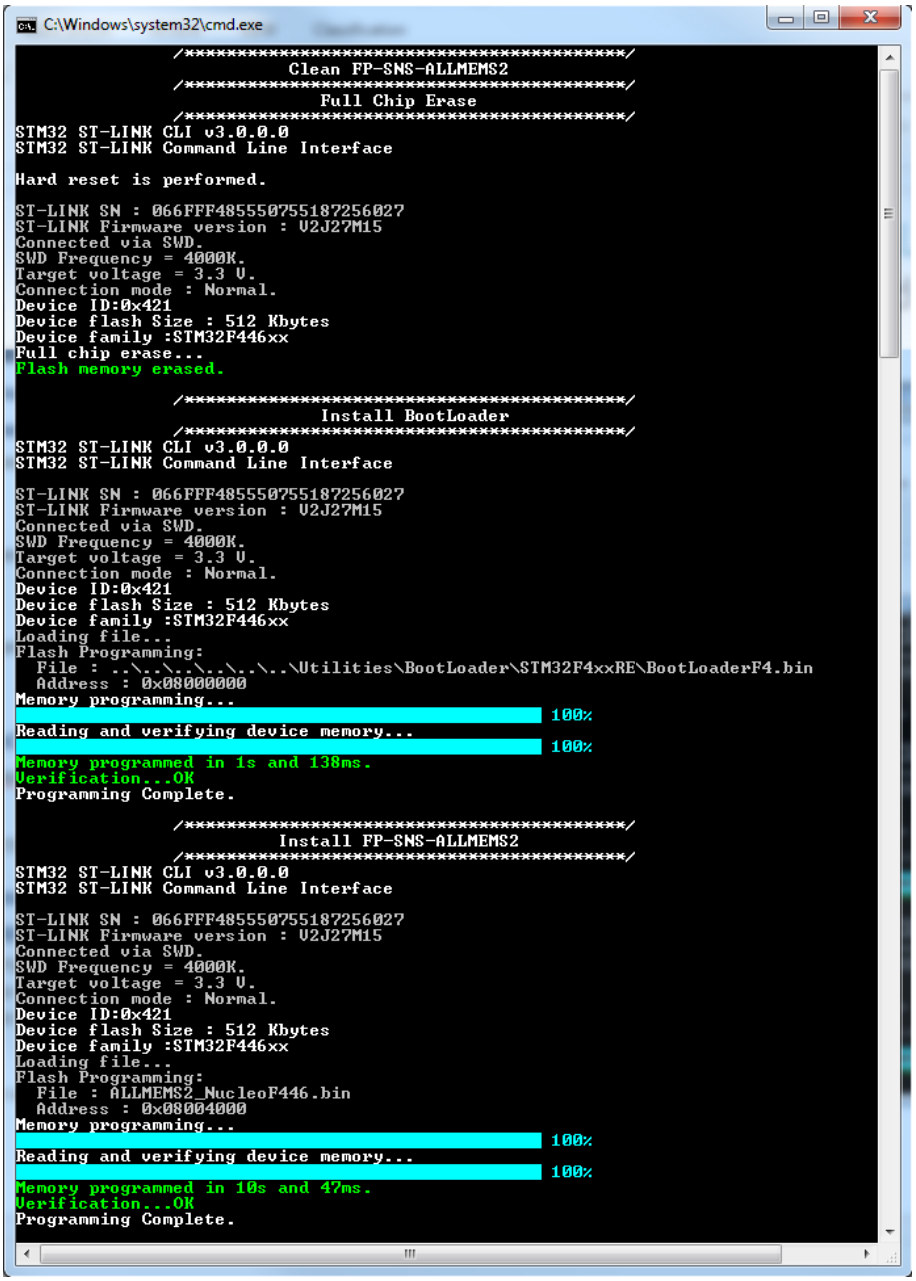


This script:

- performs a full Flash erase to start from a clean system
- flashes the BootLoader to the correct position 0x08000000

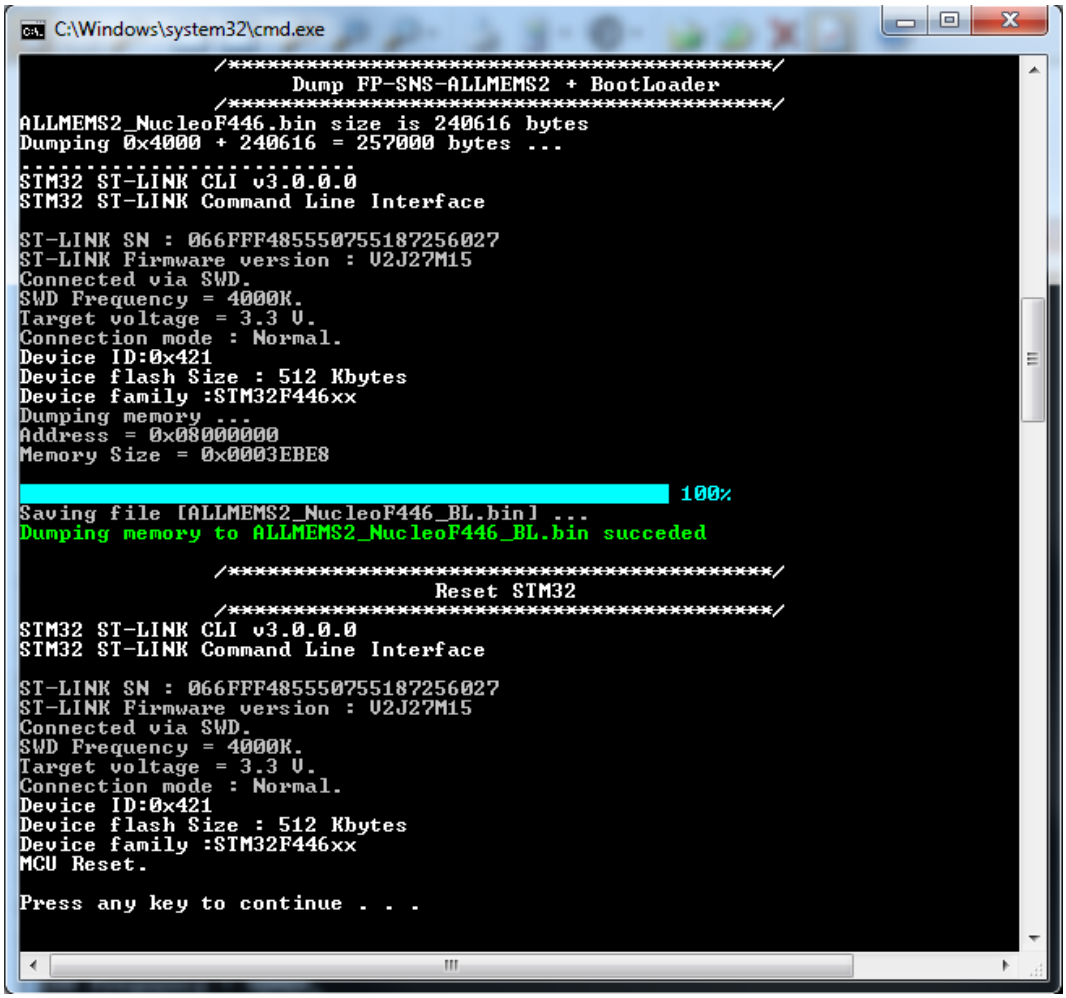
- flashes the firmware to the correct position 0x08004000

Figure 8. BootLoader and ALLMEMS2 installation



The script also dumps an image containing the BootLoader and the firmware. This image file can be directly flashed to the beginning of the Flash memory like in the same way as the image provided in the Binary folder.

Figure 9. **ALLMEMS2 Dump process**



For the Linux or iOS operating systems, there is a similar script that uses OpenOCD instead of the ST-LINK command line. The script is available for each platform, but is only included in the System Workbench IDE:

- CleanALLMEMS2_SW4STM32_F446.sh
- CleanALLMEMS2_SW4STM32_L476.sh
- CleanALLMEMS2_SW4STM32_BC.sh.
- CleanALLMEMS2_SW4STM32_ST.sh.

To function, the script must be modified with:

- The installation path for OpenOCD
- The installation path for STM32 OpenOCD scritps
- And the Library path for OpenOCD

Below is the section of the OpenOCD script to be edited:

```
# 1) Set the Installation path for OpenOCD
# example:
#OpenOCD_DIR="C:/Ac6/SystemWorkbench/plugins/fr.ac6.mcu.externaltools.openocd.win3
2_1.17.0.201801120948/tools/openocd/"
OpenOCD_DIR=""

# 2) Set the installation path for stm32 OpenOCD scritps
# example:
#OpenOCD_CFC="C:/Ac6/SystemWorkbench/plugins/fr.ac6.mcu.debug_2.1.4.201801120948/r
```

```
esources/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-ALLMEMS2 firmware may be updated Over-The-Air (FOTA) through the connected Android/iOS device via Bluetooth using the BlueMS application (ver. 3.0.0 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-ALLMEMS2 checks against the hardware cyclic redundancy check calculation unit on the STM32F446/ STM32L476 processor to ensure integrity. If the CRC calculation matches the BlueMS 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-ALLMEMS2 firmware (see Firmware-Over-The-Air update with BlueMS).

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-IKS01A2, X-NUCLEO-CCA02M1 and X-NUCLEO-IDB05A1 expansion boards with the NUCLEO-F446RE or NUCLEO-L476RG board
- the STEVAL-BCNKT01V1 evaluation board
- the STEVAL-STLKT01V1 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 10. Terminal setting) to control the initialization phase.

When the SD card data logging is not enabled, the same feature is available for the STEVAL-STLKT01V1 evaluation board when connecting the micro-USB port to a PC. However, as it is necessary to register the USB device, this is only possible when the STEVAL-STLKT01V1 starts initializing. In fact, a 10-second delay has been added to the initialization phase to allow the user to monitor its progress.

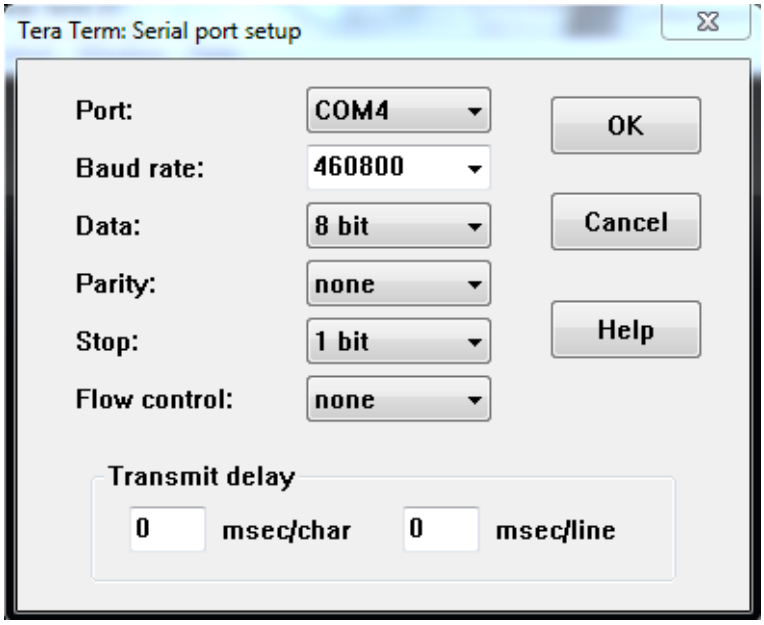
Note: You must modify the ALLMEMS2_config.h file by enabling the `//#define ALLMEMS2_ENABLE_PRINTF` line below to enable this feature for the STEVAL-STLKT01V1, as it is disabled by default.

Note: The SD card data logging is enabled by default. You must modify the ALLMEMS2_config.h file by disabling the `#define ALLMEMS2_ENABLE_SD_MEMS_RECORDING` line to disable this feature for the STEVAL-STLKT01V1.

```
#ifndef STM32_BLUECOIN
/* For enabling the printf on UART */
#ifdef STM32_SENSORTILE
  /* Enabling this define for SensorTile..
   * it will introduce a delay of 10Seconds before starting the application
   * for having time to open the Terminal
   * for looking the ALLMEMS2 Initialization phase */
  #ifndef ALLMEMS2_ENABLE_SD_MEMS_RECORDING
    //#define ALLMEMS2_ENABLE_PRINTF
```

```
  #endif /* ALLMEMS2_ENABLE_SD_MEMS_RECORDING */
#else /* STM32_SENSORTILE */
  /* For Nucleo it's enable by default */
  #define ALLMEMS2_ENABLE_PRINTF
#endif /* STM32_SENSORTILE */
#endif /* STM32_BLUECOIN */
```

Figure 10. Terminal setting

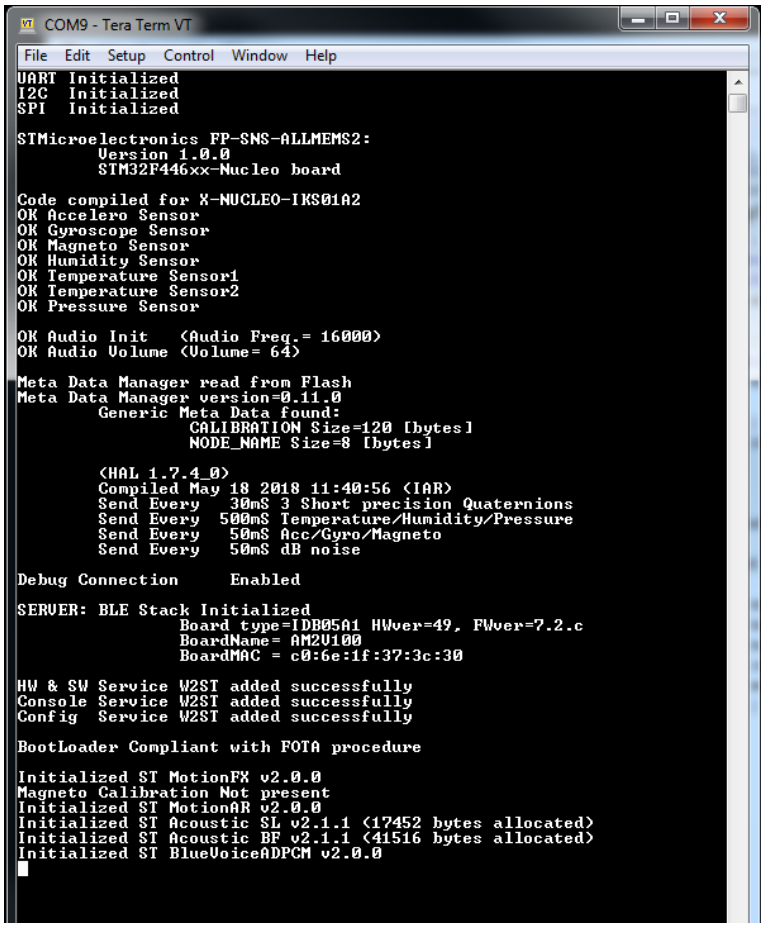


When you first press the reset button, the application:

- starts initializing the UART, I²C and SPI interfaces
- checks whether all the sensors are present and working
- checks if the BlueNRG expansion board is connected to the STM32 Nucleo board as well as the hardware and firmware version information
- creates a random BLE MAC address
- initializes the BLE hardware service (adding the temperature, humidity, pressure, 3D gyroscope, 3D magnetometer, 3D accelerometer, microphone and Gas Gauge IC characteristics) and the BLE software service (adding the MotionFX, MotionAR, MotionCP, MotionGR, 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 LSM6DSL mounted on the X-NUCLEO-IKS01A2 expansion board (for Nucleo L4 only).

For 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 11. **Initialization phase**



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 BlueMS 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-STLKT01V1 evaluation board.

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 ALLMEMS2_config.h file also defines:

- `ALLMEMS2_DEBUG_CONNECTION` and `ALLMEMS2_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-STLKT01V1 and STEVAL-BCNKT01V1 evaluation boards, this procedure can be done only through the BlueMS application.

The MotionAR (iNEMOEngine PRO) library is able to recognize the following activities:

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

The MotionCP (iNEMOEngine PRO) 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 NUCLEO-F446RE and STEVAL-BCNKT01V1):

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

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

- pick up
- glance
- wake up in hand

The AcousticSL library can localize audio sound sources using the data acquired from microphones (feature not available on the STEVAL-STLKT01V1).

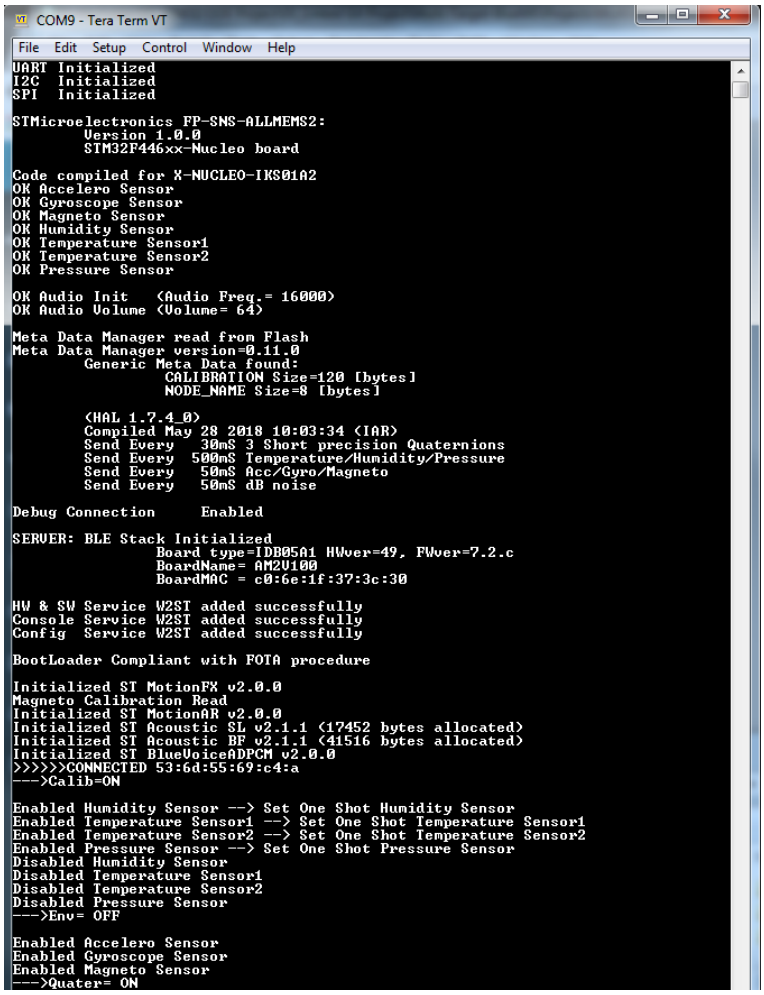
The AcousticBF library provides real-time beam forming software, using the audio signals acquired from two digital MEMS microphones, it creates a virtual directional microphone pointing to a fixed direction in space (feature not available on STEVAL-STLKT01V1 and NUCLEO-L476RG).

The BlueVoiceADPCM library implements a vendor-specific profile enabling voice communication with Bluetooth low energy.

The FatFs library provides access to the storage devices for sensor data logging (feature available for STEVAL-STLCS01V1 only).

When an Android/iOS device is connected to the NUCLEO-F446RE, NUCLEO-L476RG or STEVAL-STLKT01V1 (if the define `#define ALLMEMS2_ENABLE_PRINTF` is enabled) board, you can control data transmitted via the board.

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



1.10 Android and iOS sample client application

Based on the FP-SNS-ALLMEMS1 package and developed to reduce the power consumer, the FP-SNS-ALLMEMS2 software for STM32Cube is compatible with the BlueMS Android/iOS applications (ver. 3.0.0 or higher) available at the respective Play/iOS stores.

The ST BlueMS Android/iOS application allows Over-The-Air firmware updates (for X-NUCLEO-IDB05A1 Bluetooth low energy expansion board only) and version 3.2.0 or higher is required to display battery information (remaining charge, voltage and charge status) for the STEVAL-STLKT01V1 board.

BlueMS Version 3.5.0 or higher is required to be able to show the Source Localization and Beam Forming pages.

BlueMS Version 3.6.0 or higher is required to be able to show the Multi Events detection and the battery information for STEVAL-BCNKT01V1.

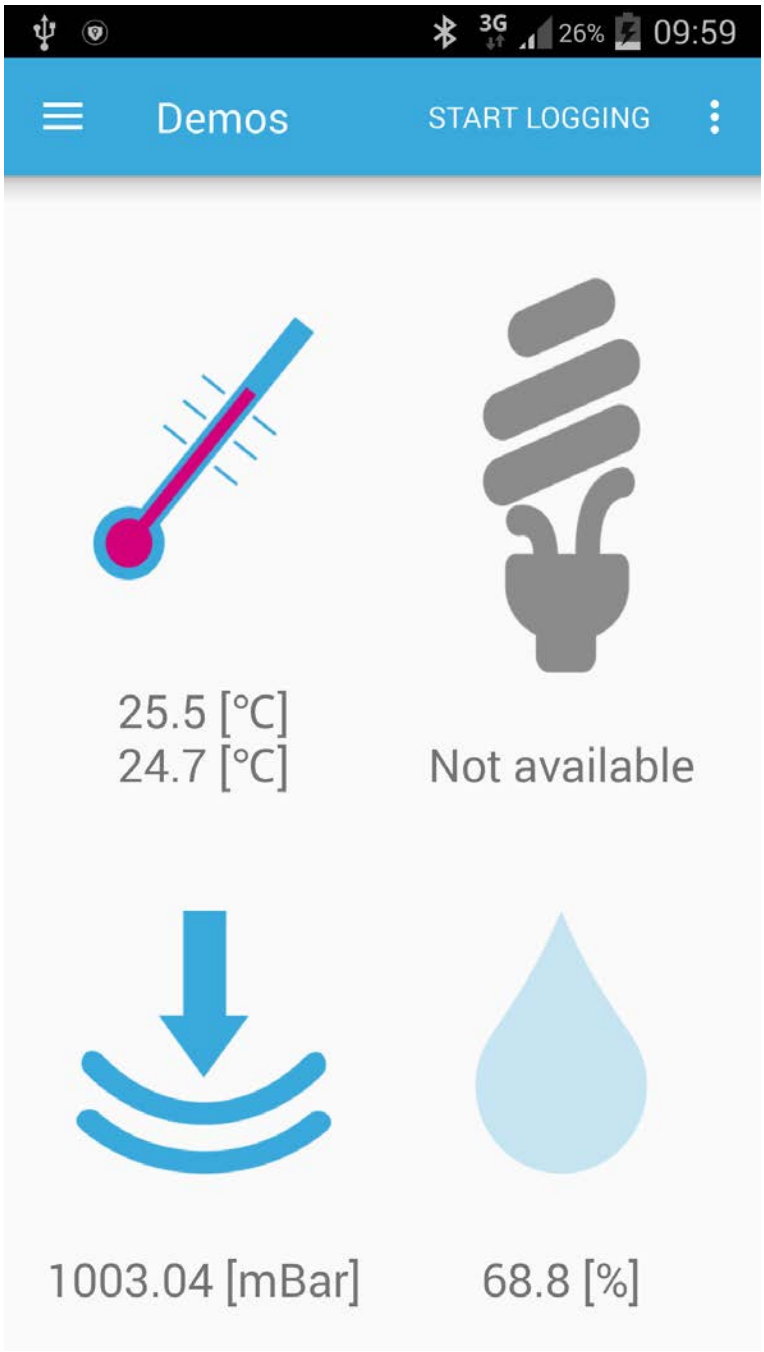
BlueMS Version 3.7.0 or higher is required to show the data logging setting on the SD card for the STEVAL-STLKT01V1.

We will use the Android application for this demonstration.

1.10.1 Main page

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

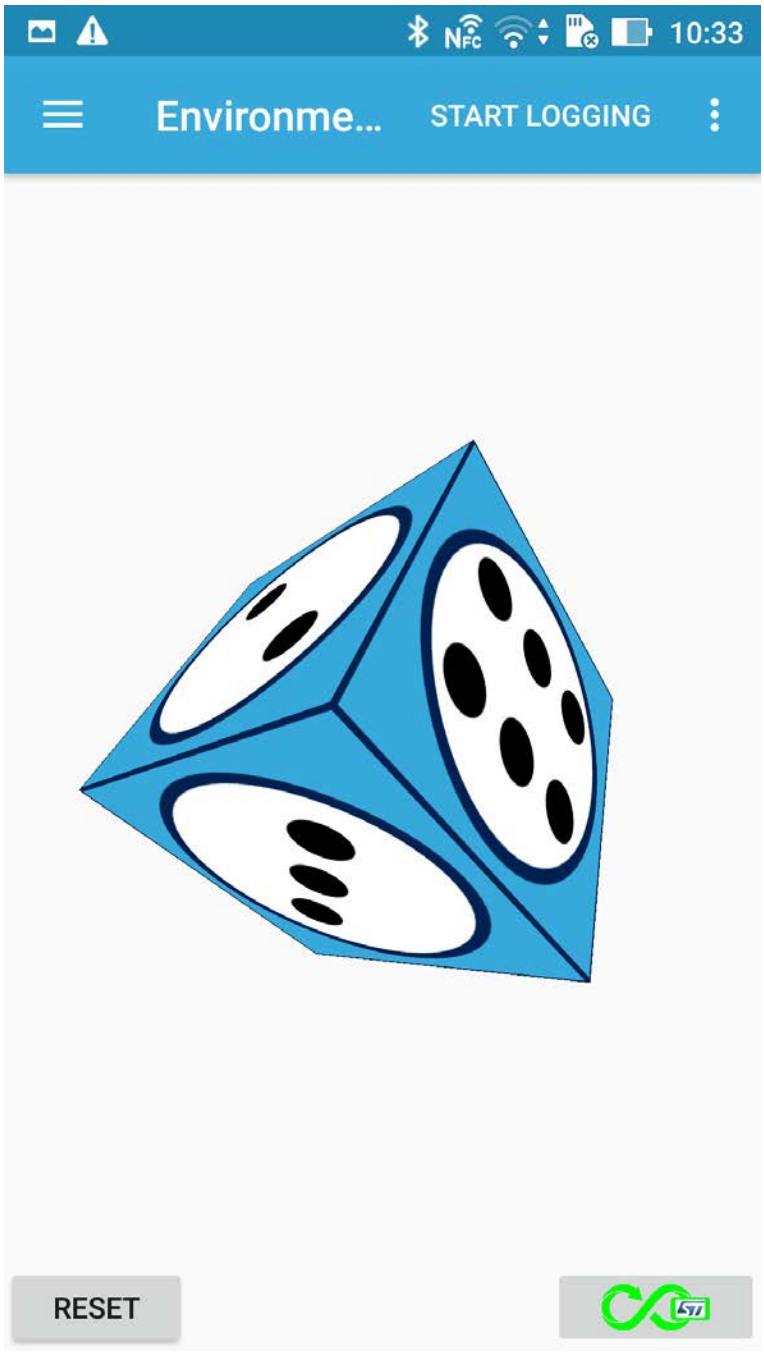
Figure 13. **BlueMS (Android version) main page following BLE connection**



1.10.2 MEMS sensor fusion

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

Figure 14. **BlueMS (Android version) MotionFX sensor fusion page**

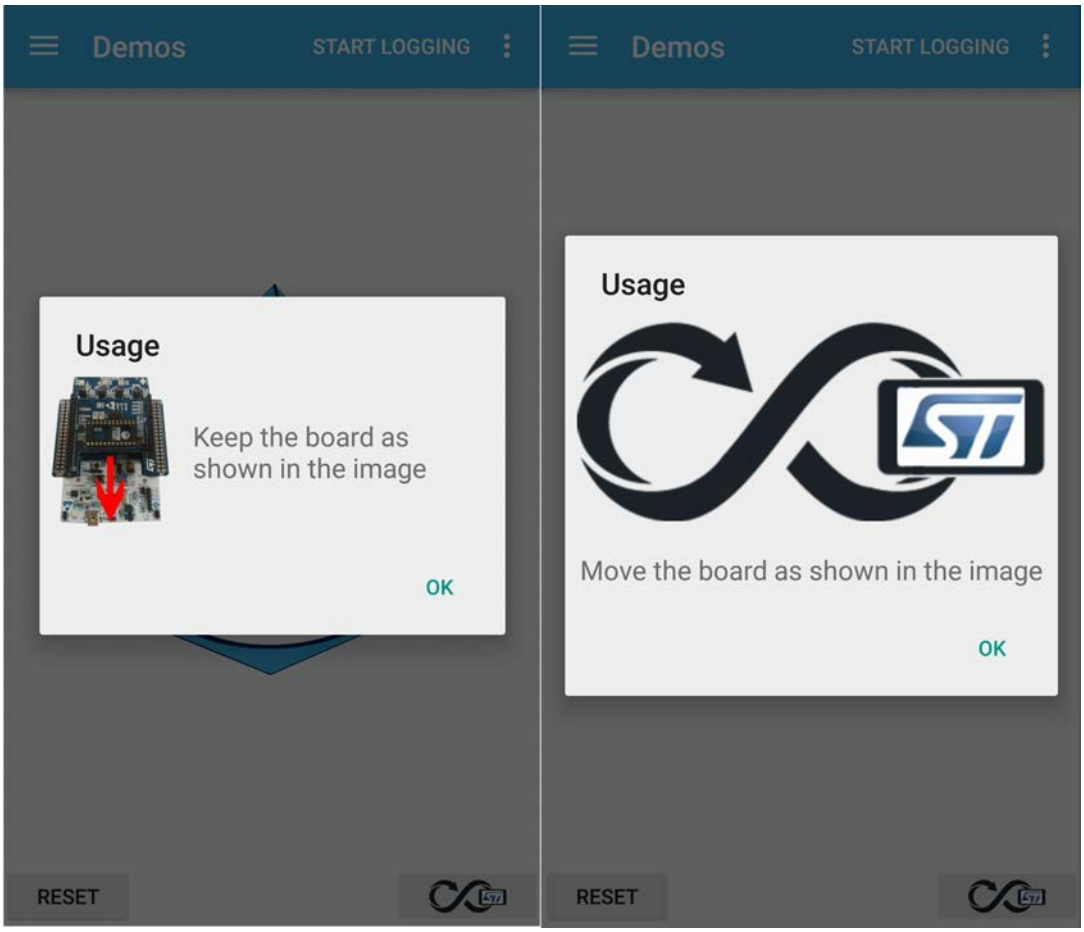


On this page, there are two buttons along the bottom:

- 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).

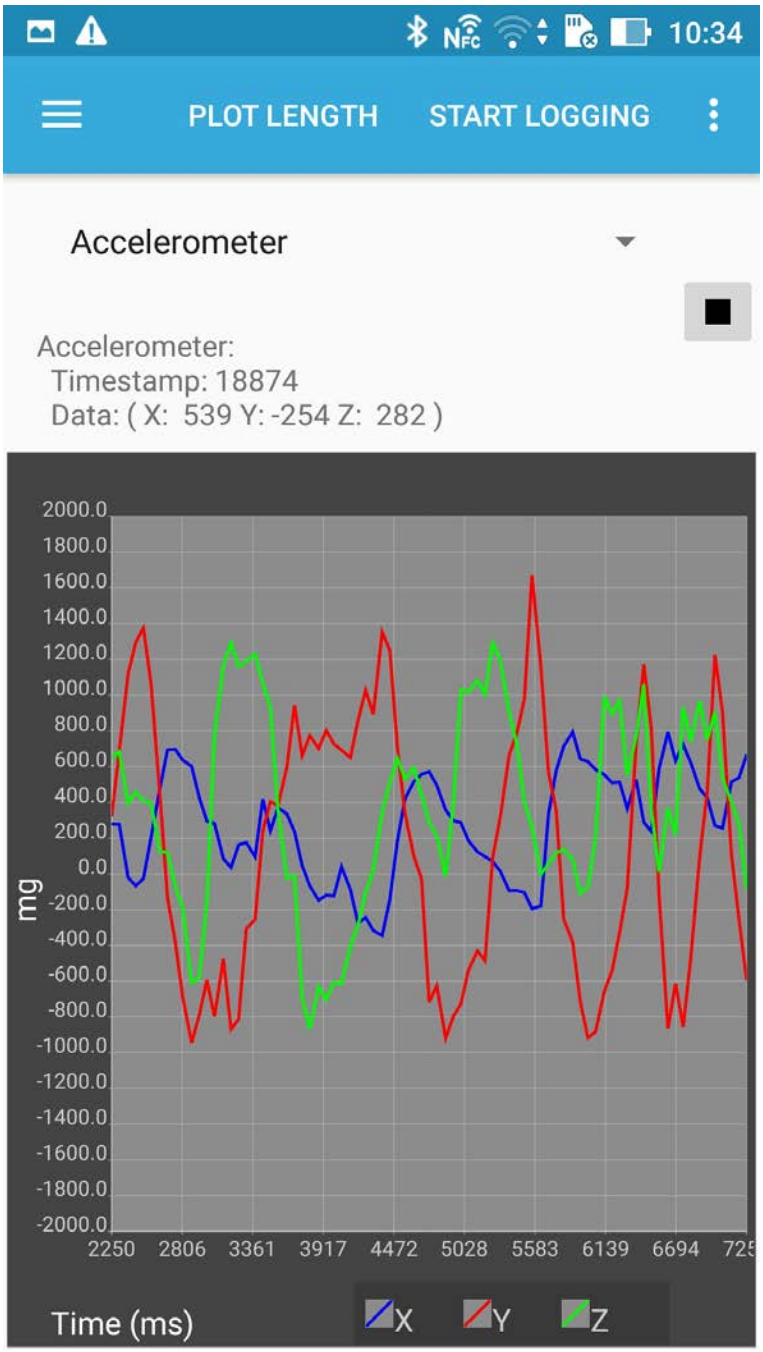
Figure 15. **BlueMS (Android version) popup windows**



1.10.3 Plot data

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

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

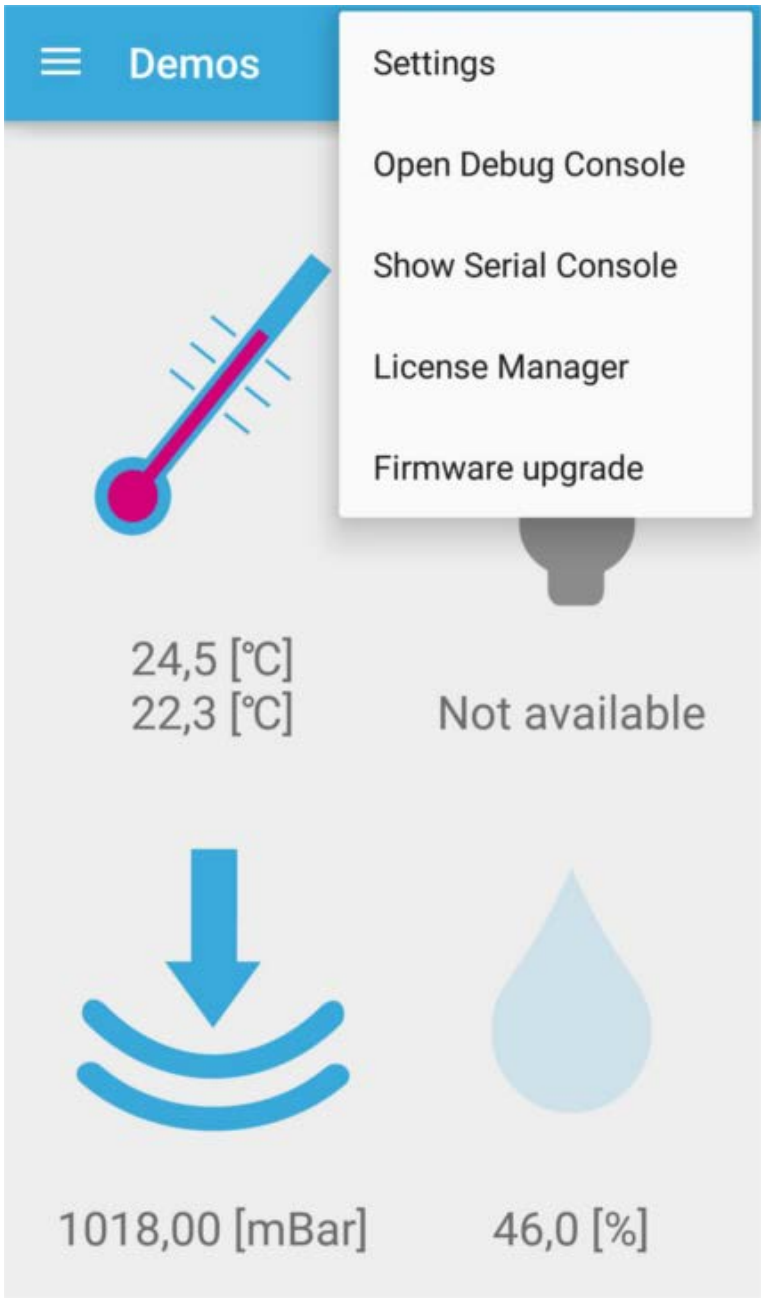


1.10.4 Settings, serial and debug console

In the option menu below, you can open:

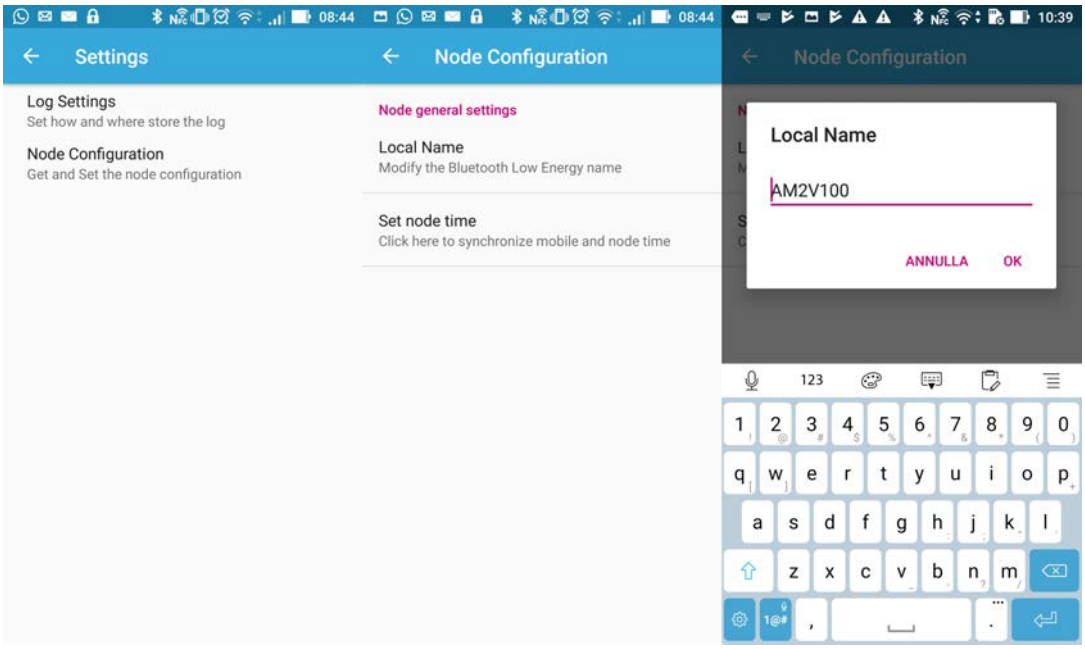
- Settings
- Serial or Debug (with stdin) console
- Firmware upgrade

Figure 17. **BlueMS (Android version) menu selection**



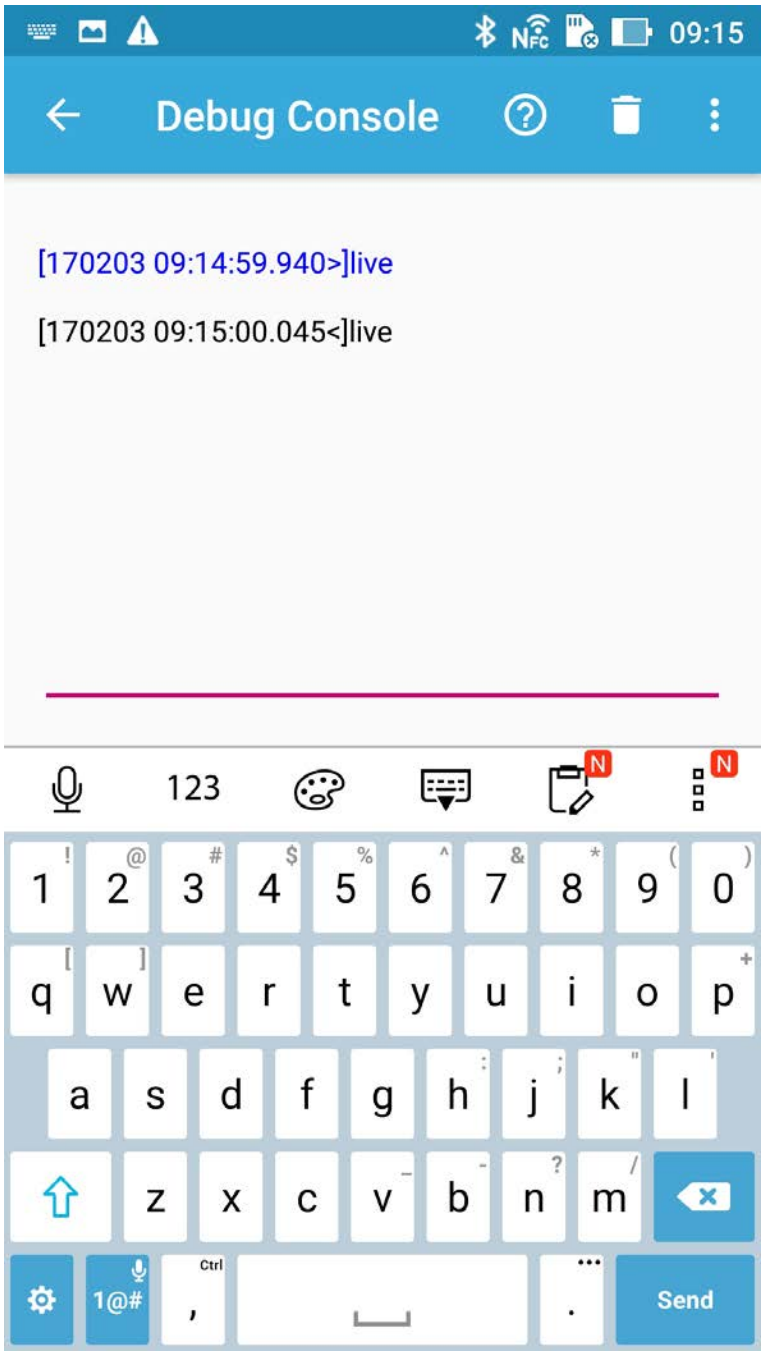
If Settings is selected, it is possible to change the node name using the node configuration, as shown below:

Figure 18. **BlueMS (Android version) Settings, Node Configuration, Local Name**



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

Figure 19. **BlueMS (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)

1.10.4.1 *SD card data logging*

SD data logging is available (only for the STEVAL-STLKT01V1 (SensorTile) and using Generic FAT File System middleware) for environmental (temperature, pressure, humidity), magnetometer, gyroscope, accelerometer and audio data.

The debug console commands to start the data logging are:

- **Start**: to start the data logging for enviromental, accelerometer, magnetometer and gyroscope data.
- **Stop**: to stop the data logging for enviromental, accelerometer, magnetometer and gyroscope data.
- **AudioStart**: to start the data logging for audio data.
- **AudioStop**: to stop the data logging for audio data.

It is not possible to start the data logging for MEMS and audio data simultaneously.

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

Note: To stop the data logging, it is necessary to open the app.

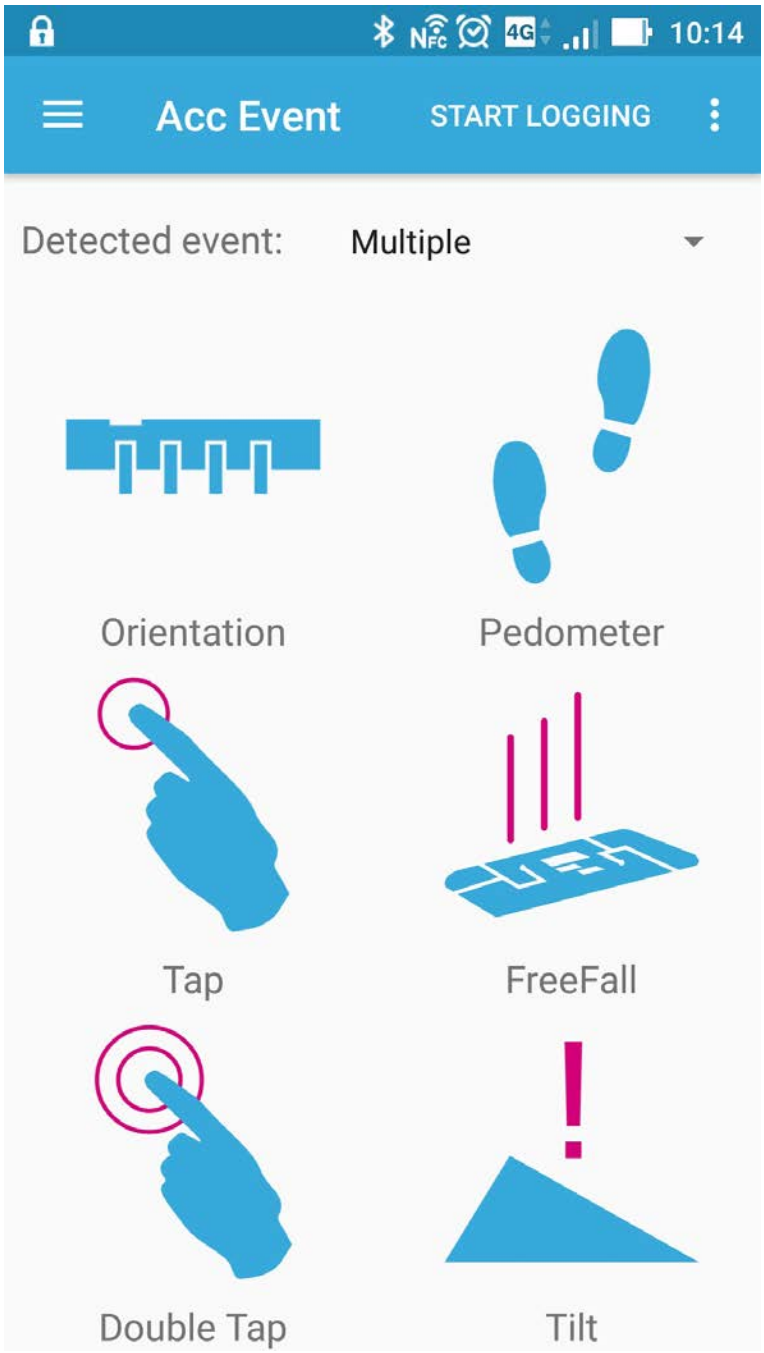
1.10.5 Enable hardware features

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:

- LSM6DSL on X-NUCLEO-IKS01A2 expansion board for NUCLEO-L476RG board only
- LSM6DSM for STEVAL-BCNKT01V1 and STEVAL-STLKT01V1 boards

The multiple hardware feature is the default setting.

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



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

Figure 21. **BlueMS (Android version) hardware feature menu**

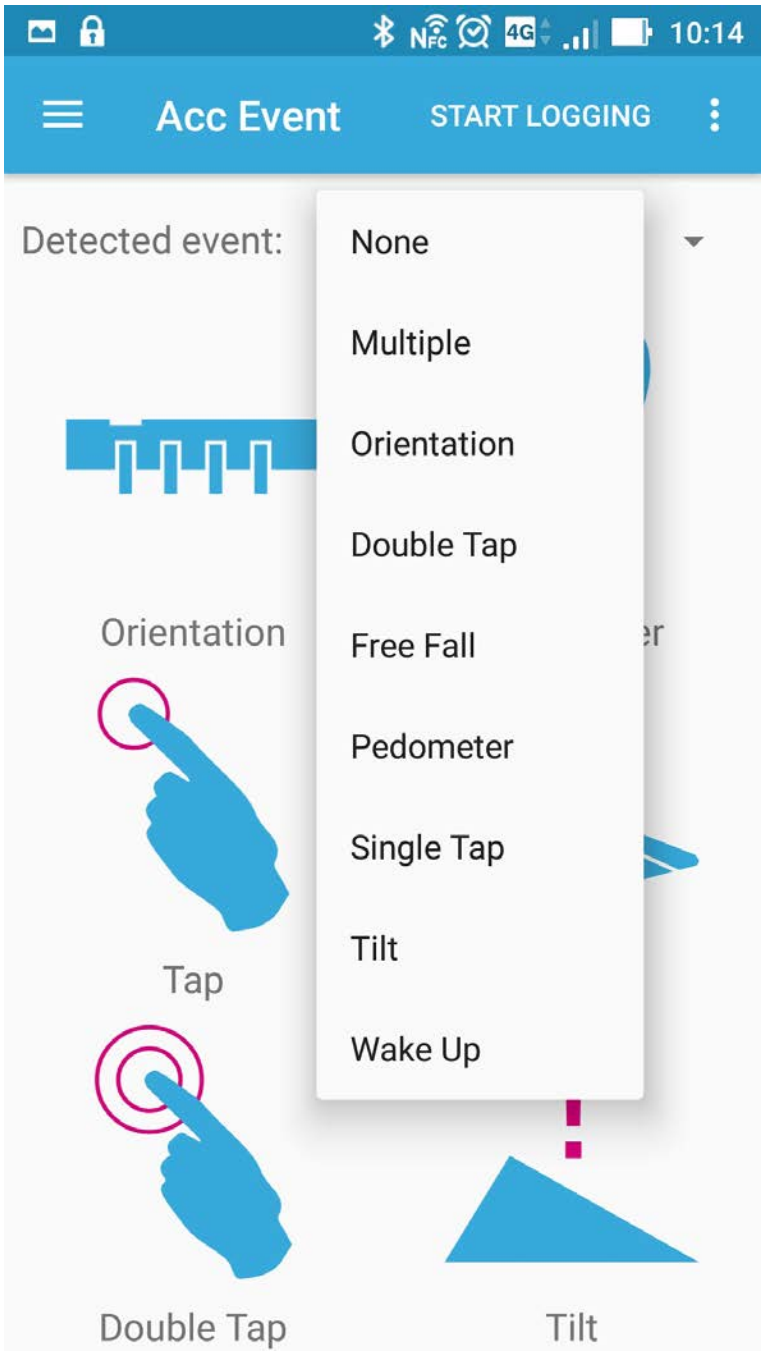
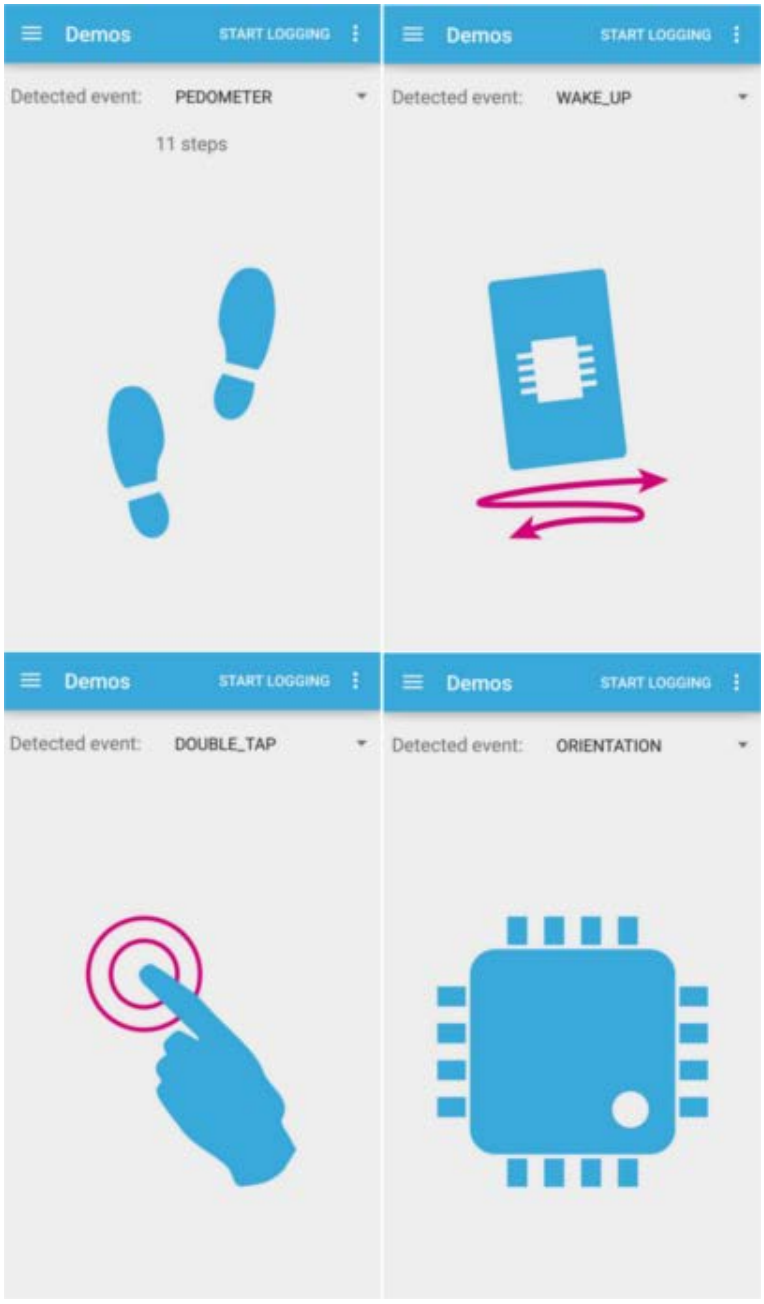


Figure 22. **BlueMS (Android version) hardware feature examples**



1.10.6 Activity recognition

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

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.

Figure 23. **BlueMS (Android version) MotionAR activity recognition page**

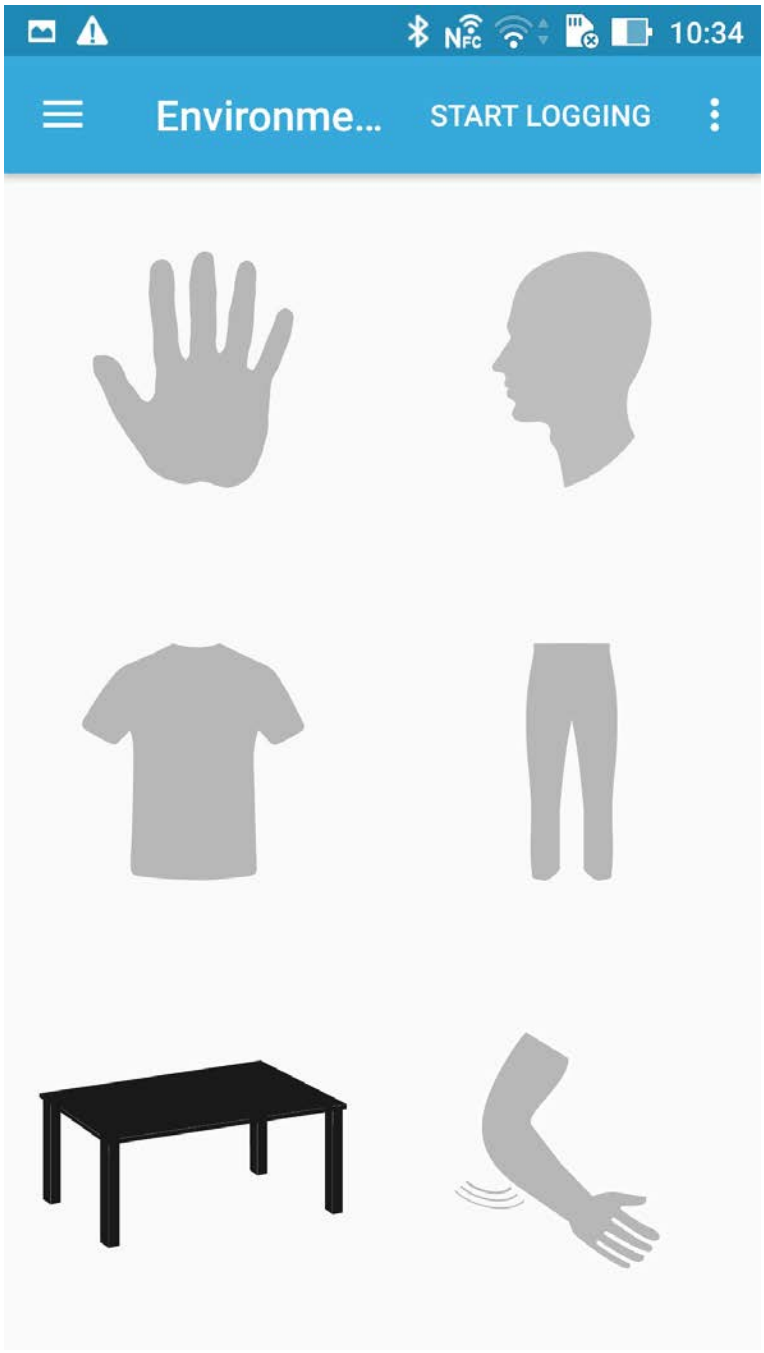
1.10.7 Carry position

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 (feature not available on NUCLEO-F446RE and STEVAL-BCNKT01V1):

- 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.

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



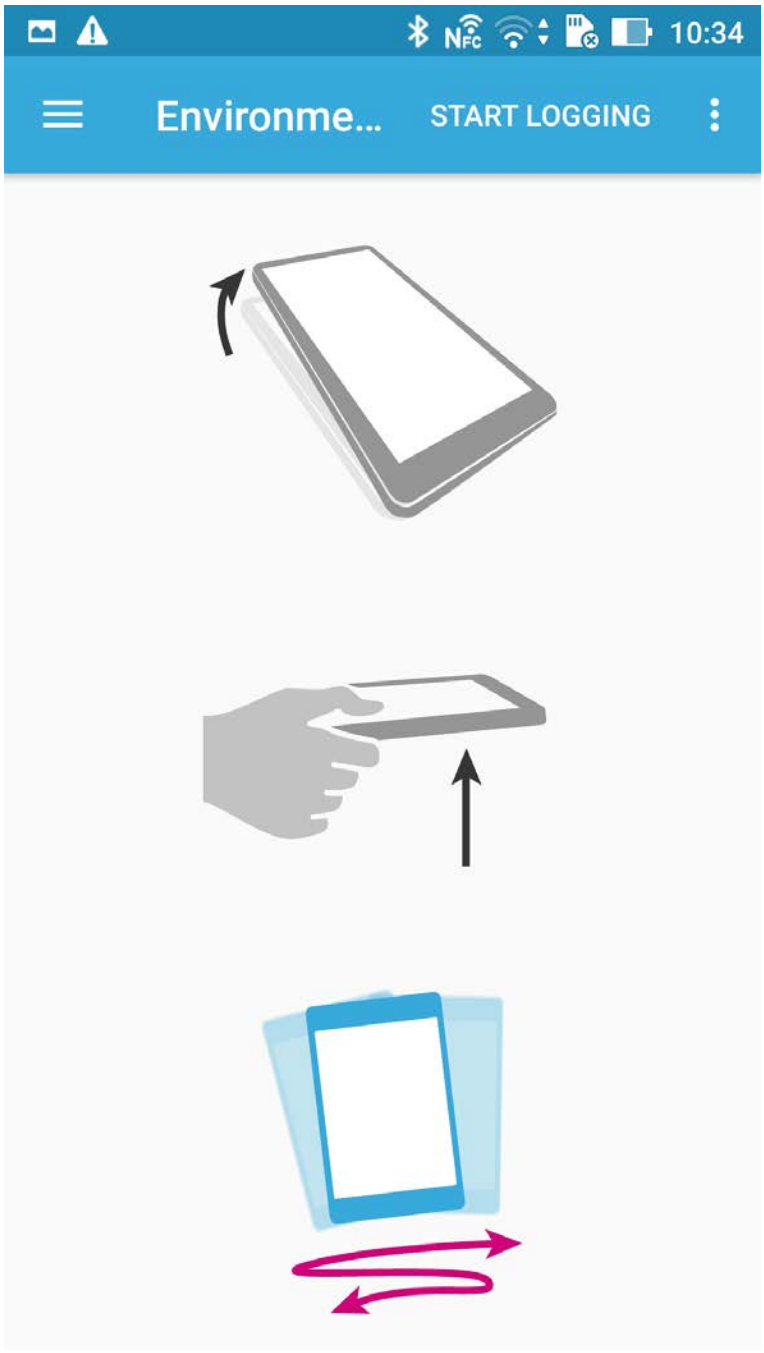
1.10.8 Gesture recognition

If the MotionGR algorithm is enabled, the page shown below is available with gesture recognition information like (feature not available on NUCLEO-F446RE and STEVAL-BCNKT01V1):

- 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.

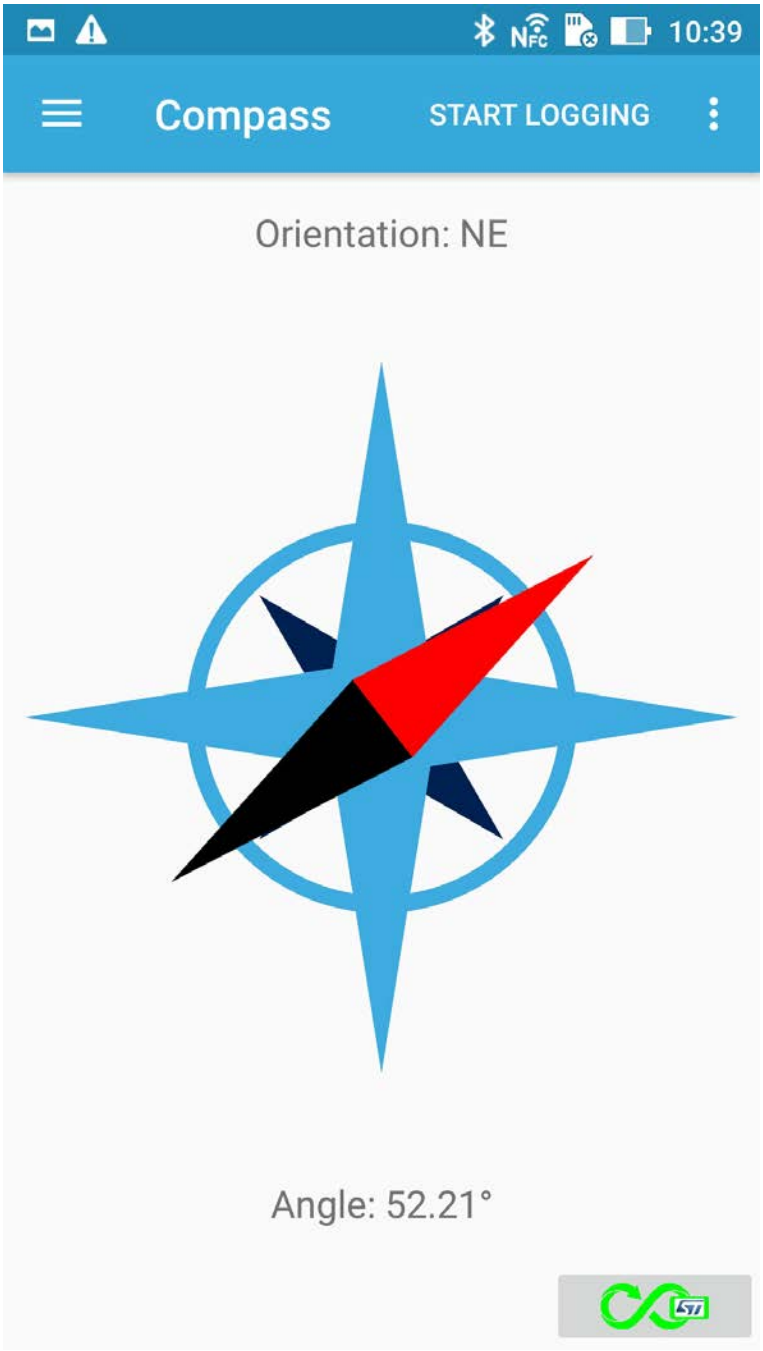
Figure 25. BlueMS (Android version) MotionGR gesture recognition page



1.10.9 E-compass

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

Figure 26. **BlueMS (Android version) MotionFX page**



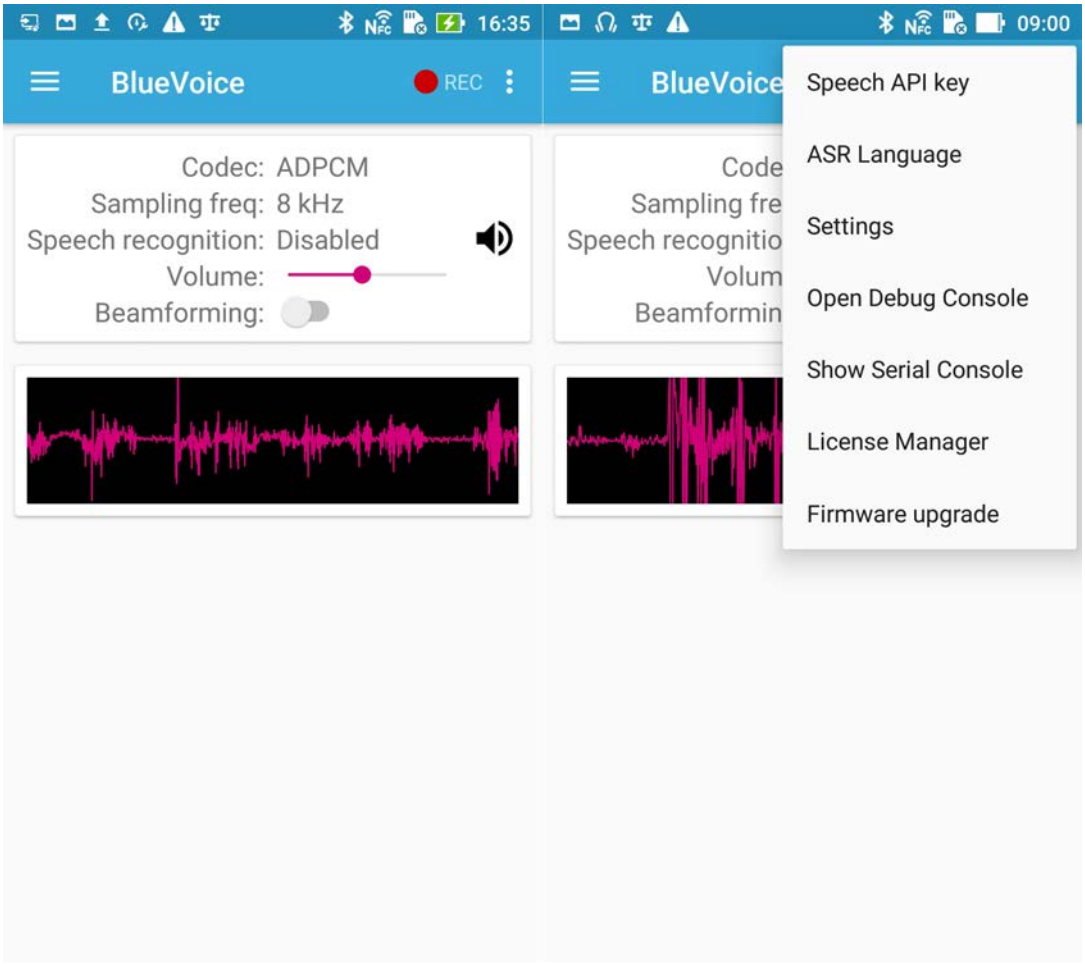
On the page bottom, the right button shows the MotionFX library calibration status (black for not calibrated, green for calibrated). Clicking it forces a magneto calibration.

1.10.10 BlueVoiceADPCM

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.
- Web-based Google ASR service.
- Web-based Chinese ASR: iFlyTek MSC service.

Figure 27. **BlueMS (Android version) BlueVoice start page**

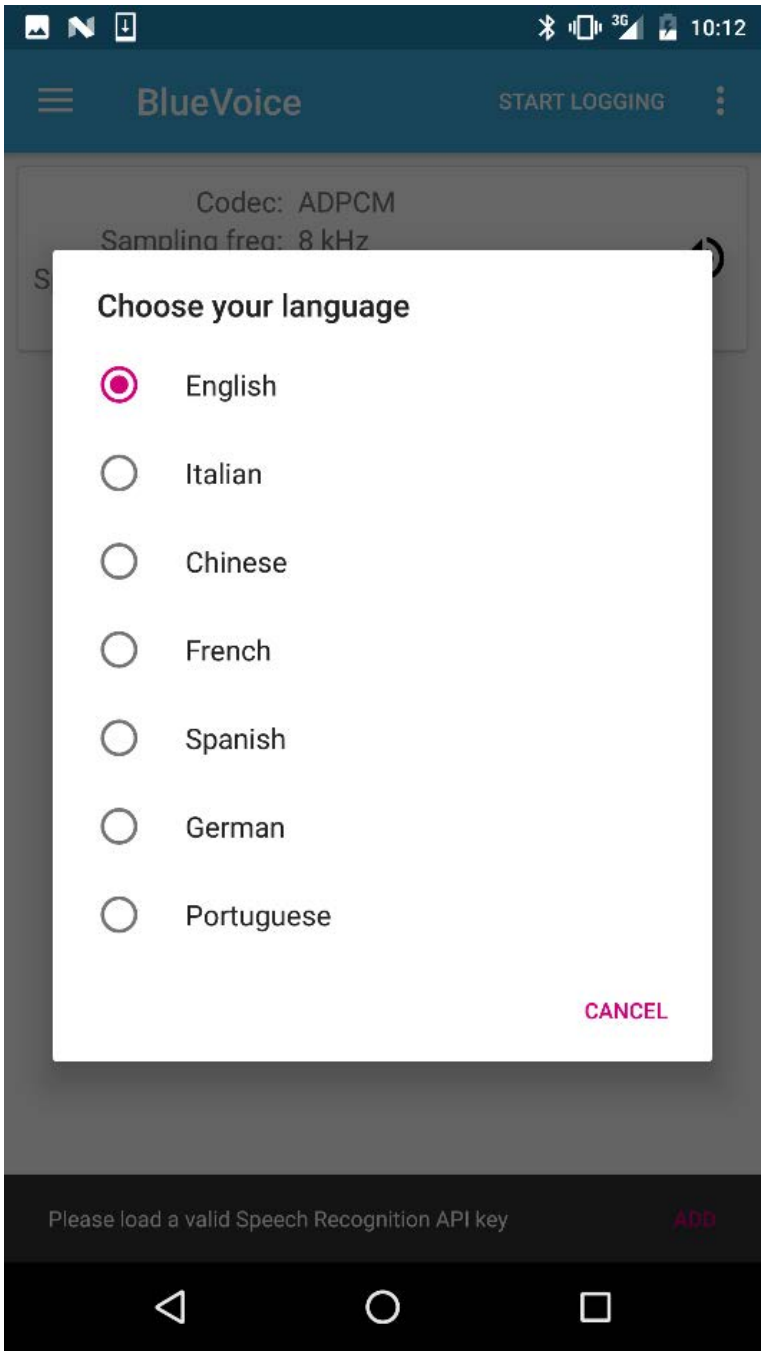


The audio playback begins as soon as the streaming from the peripheral node starts. The volume can be adjusted using the slider or muted by clicking on the speaker icon.

1.10.10.1 *ASR language selection*

Opening the ASR language menu, in the demo main menu, the application displays a popup window for ASR language selection. A specific ASR service will be configured according to the language selected.

Figure 28. **BlueMS (Android version) ASR language selection**



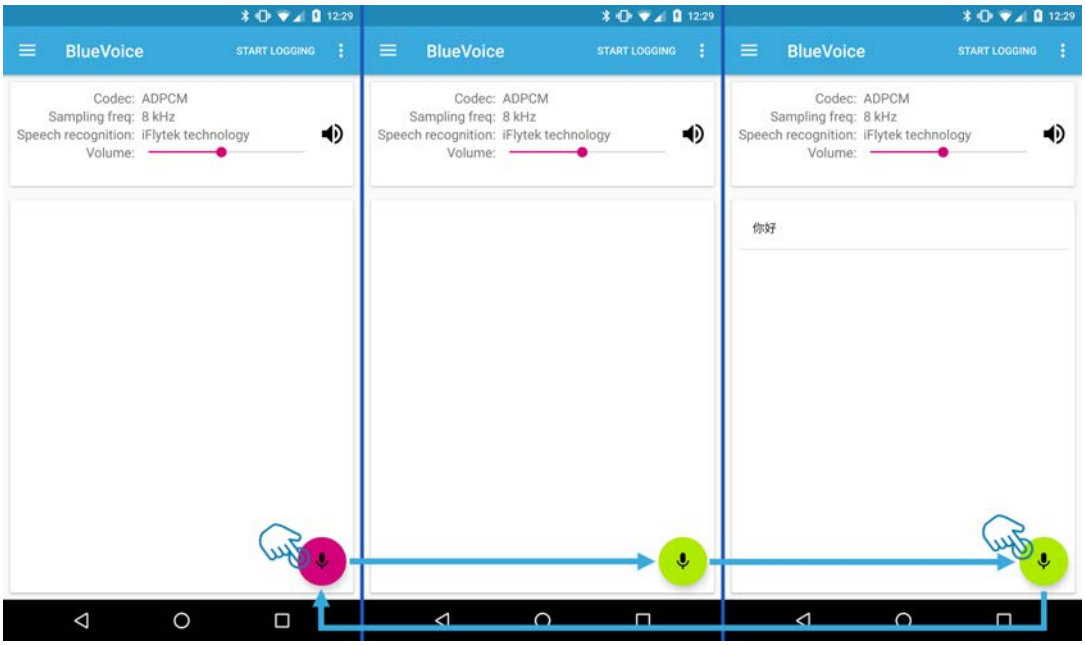
1.10.10.1.1 Chinese ASR: iFlyTek MSC service

When Chinese is selected, the ASR service provided by iFlyTek is enabled.

Pushing the button on the bottom right hand of the screen, it becomes green and the speech-to-text service starts.

The recognition is continuous and every sentence is recorded as shown below.

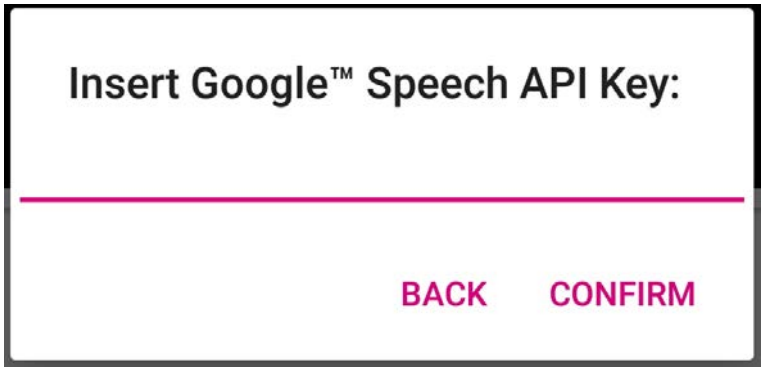
Figure 29. **BlueMS (Android version) Chinese ASR, iFlytek technology**



1.10.10.1.2 Alternative languages: Google Speech API

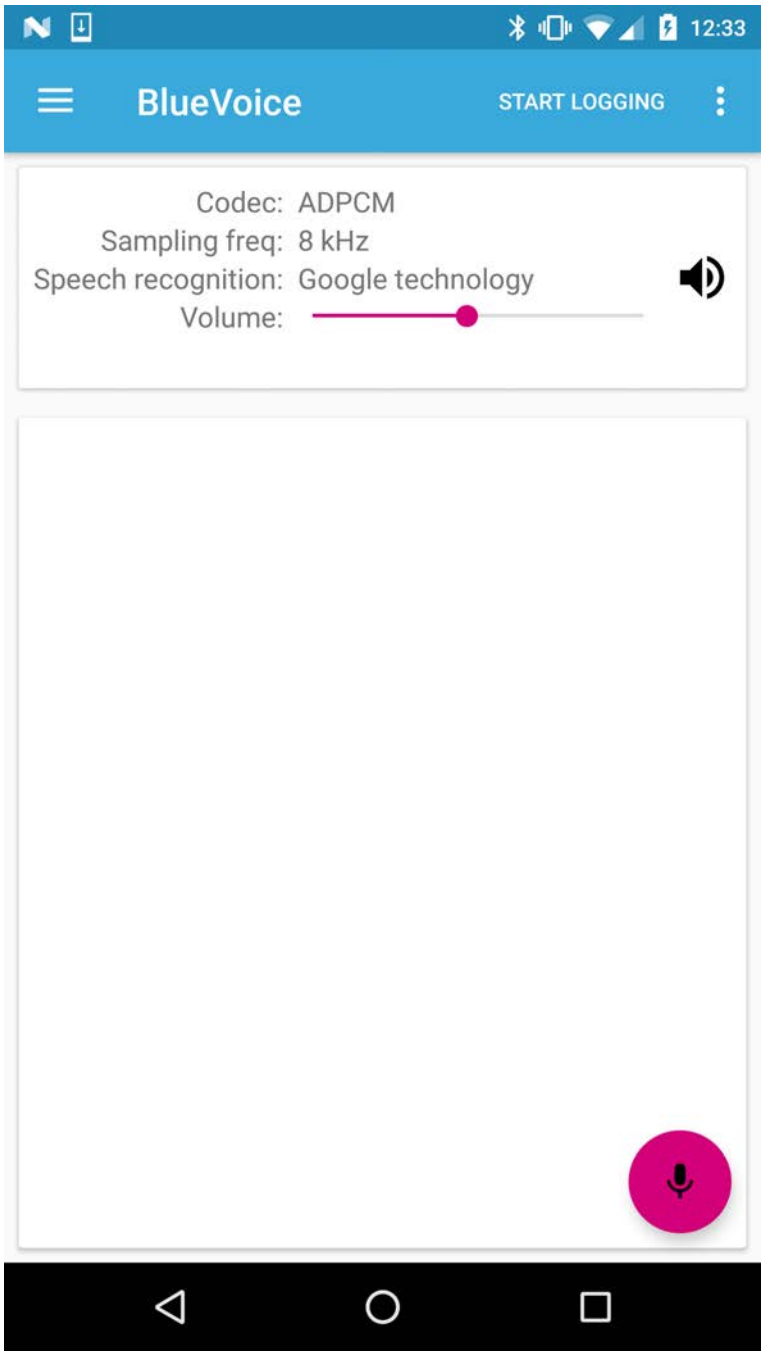
The ADD button allows the insertion of the key (see Section 1.10.10.2 Google speech ASR Key generation) to enable the ASR feature: a popup window prompts the insertion of a valid API key, followed by the ASR service activation key.

Figure 30. **BlueMS (Android version) popup API key window**



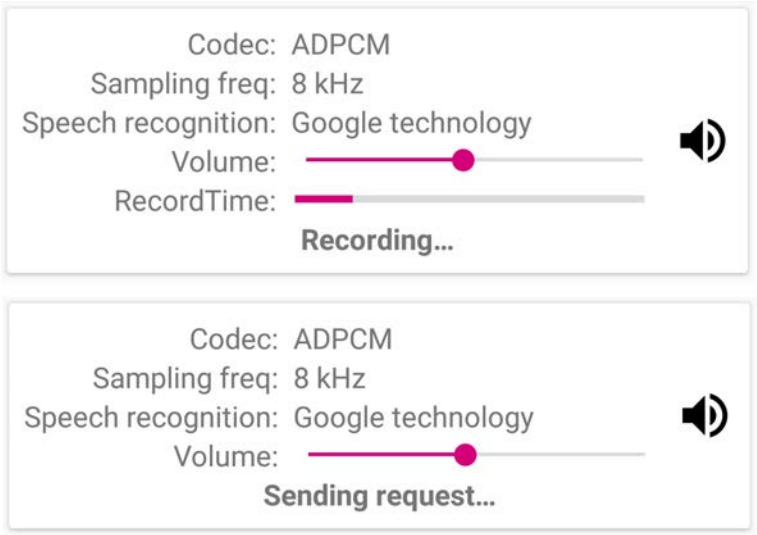
Once the key is correctly inserted, the start screen changes.

Figure 31. **BlueMS (Android version) ASR service enabled**



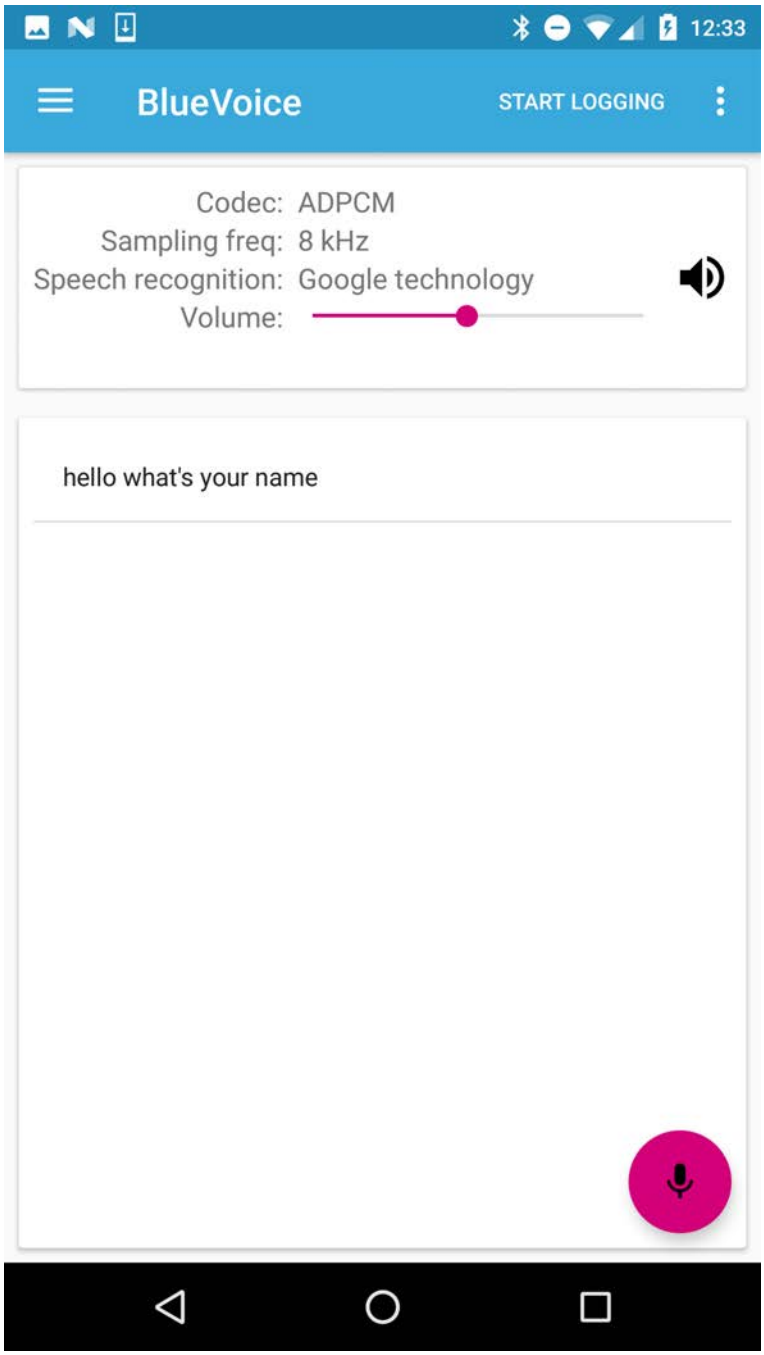
Hold the recording button to record your voice for subsequent recognition. While the button is pressed, a bar progressively indicates the elapsed recording time. When you release the button a “Sending request…” message appears.

Figure 32. **BlueMS (Android version) voice recording**



The speech recognized by the ASR service appears below the volume bar.

Figure 33. **BlueMS (Android version) recognised voice text**



Note: If the recording cannot be recognized, a "Token not recognized' message appears instead of the text.

1.10.10.2 *Google speech ASR Key generation*

The Google Speech APIs require a key to access the web-based service. You need a Google account to complete the procedure and access the service.

To generate a key:

Step 1. Login with your own Google account.

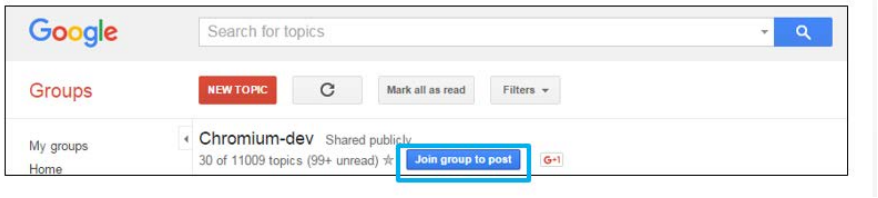
Step 2. Subscribe to Chromium-dev at https://groups.google.com/a/chromium.org/forum/?fromgroups#!forum/chromium-dev.

Step 3. Write "Chromium-dev" in the search box, and select the appropriate group.

Figure 34. **Google Chromium-dev: search group**

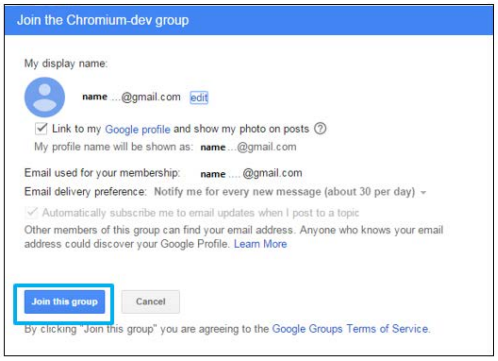
Step 4. Click on “Join group to post” button

Figure 35. **Google Chromium-dev: join group to post**



Step 5. Click on “Join this group” button to join the Chromium-dev group.

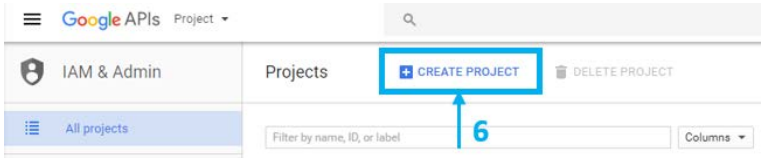
Figure 36. **Google Chromium-dev: join the group**



Step 6. Go to https://console.developers.google.com/project

Step 7. Click on “Create a project…”

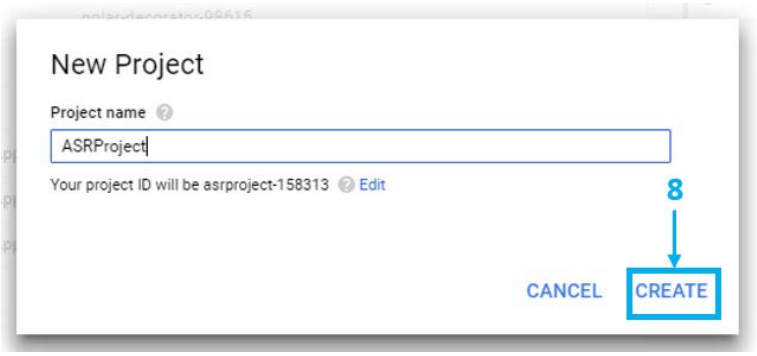
Figure 37. **Google Chromium-dev: create project**



Step 8. Choose the Project name.

Step 9. Click on “Create” button.

Figure 38. **Google Developers Console: new project**

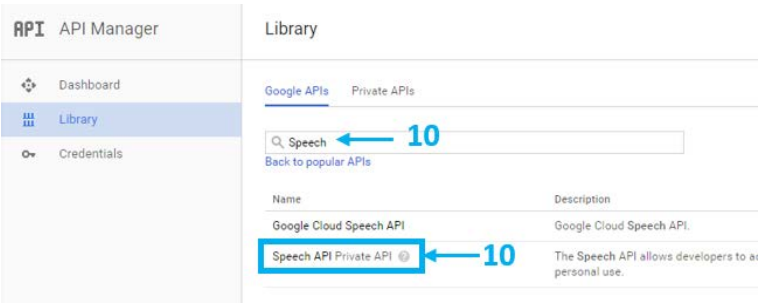


Step 10. Make sure you have selected the newly created project.

Figure 39. **Google Developers Console: ASRProject**

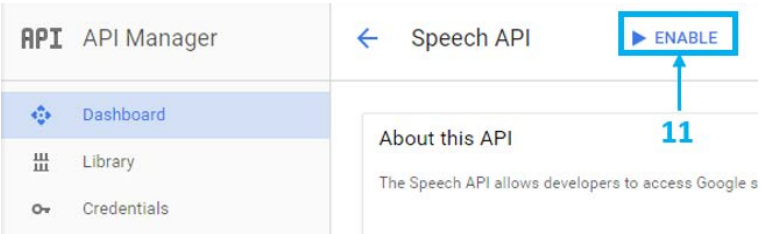
Step 11. Write "Speech API" in the search box, and select correct result.

Figure 40. **Google Developers Console: select API**



Step 12. Enable the Speech API clicking on the blue button.

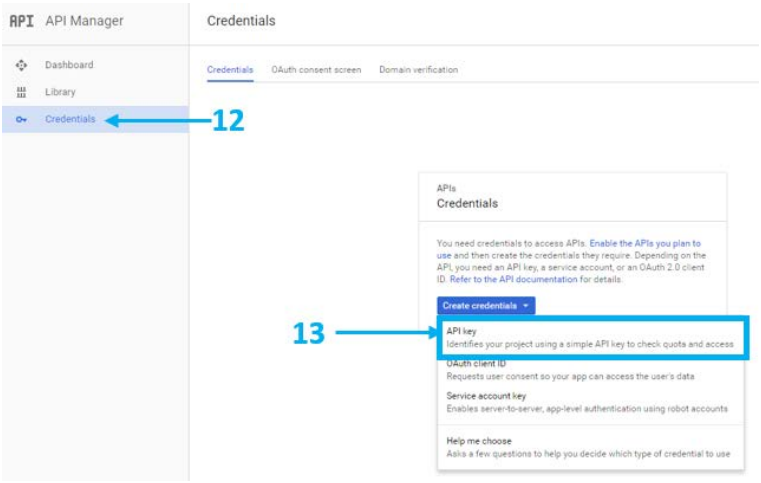
Figure 41. **Google API Manager: enable API**



Step 13. Move from the “Dashboard” tab to “Credentials” tab.

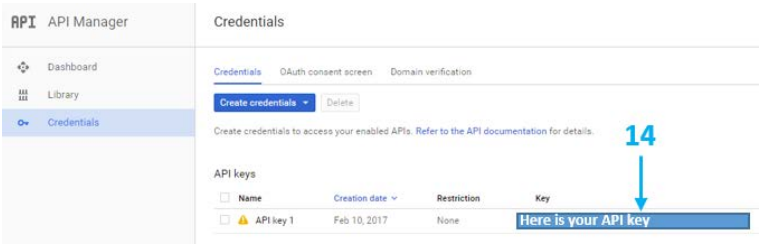
Step 14. Open the “Create credentials” menu and select “API key”.

Figure 42. **Google API Manager: create API key**



Step 15. Your API key is created. Click on Close to return to the Credentials section. Here you can see your API Key.

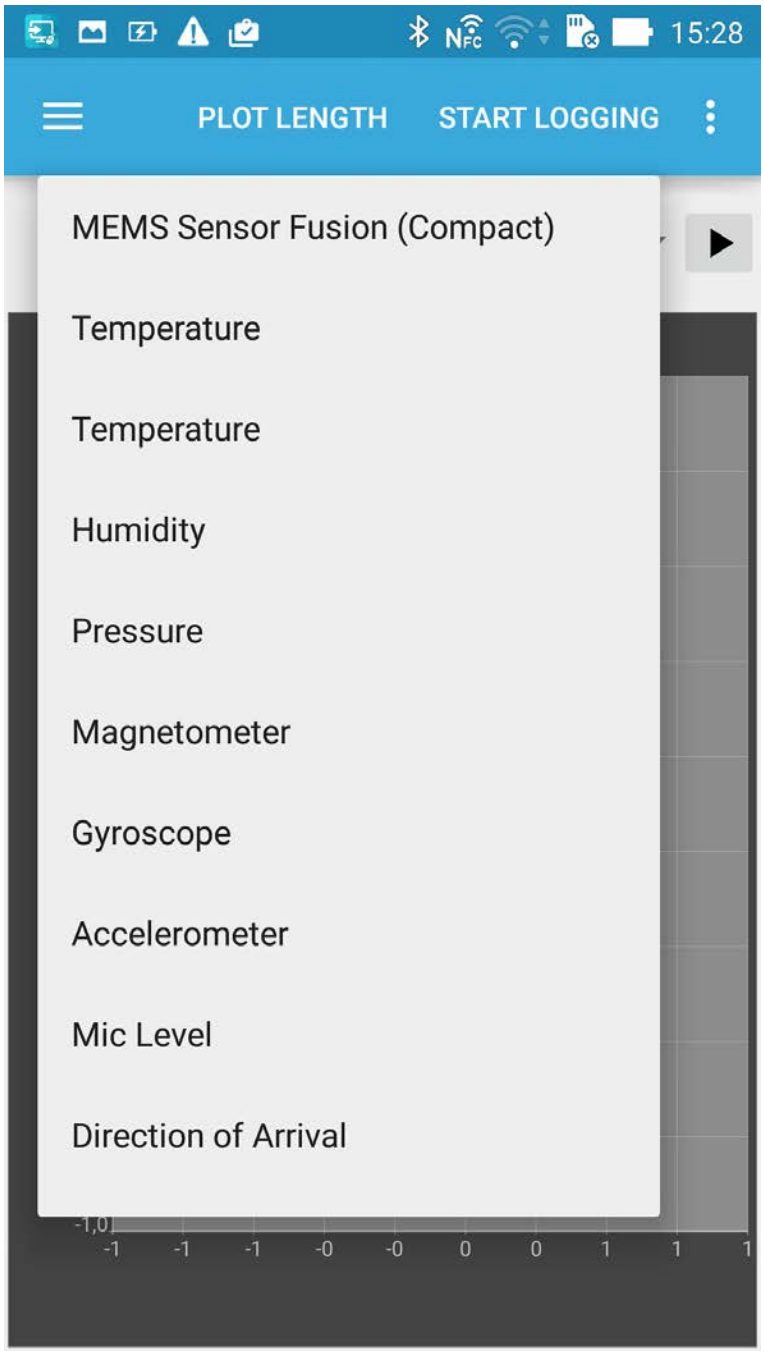
Figure 43. **Google API Manager: Android API key**



1.10.11 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-STLKT01V1).

Figure 44. **BlueMS (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 45. **BlueMS (Android version) audio source localization plot example**

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

Figure 46. **BlueMS (Android version) audio source localization BlueCoin Page**

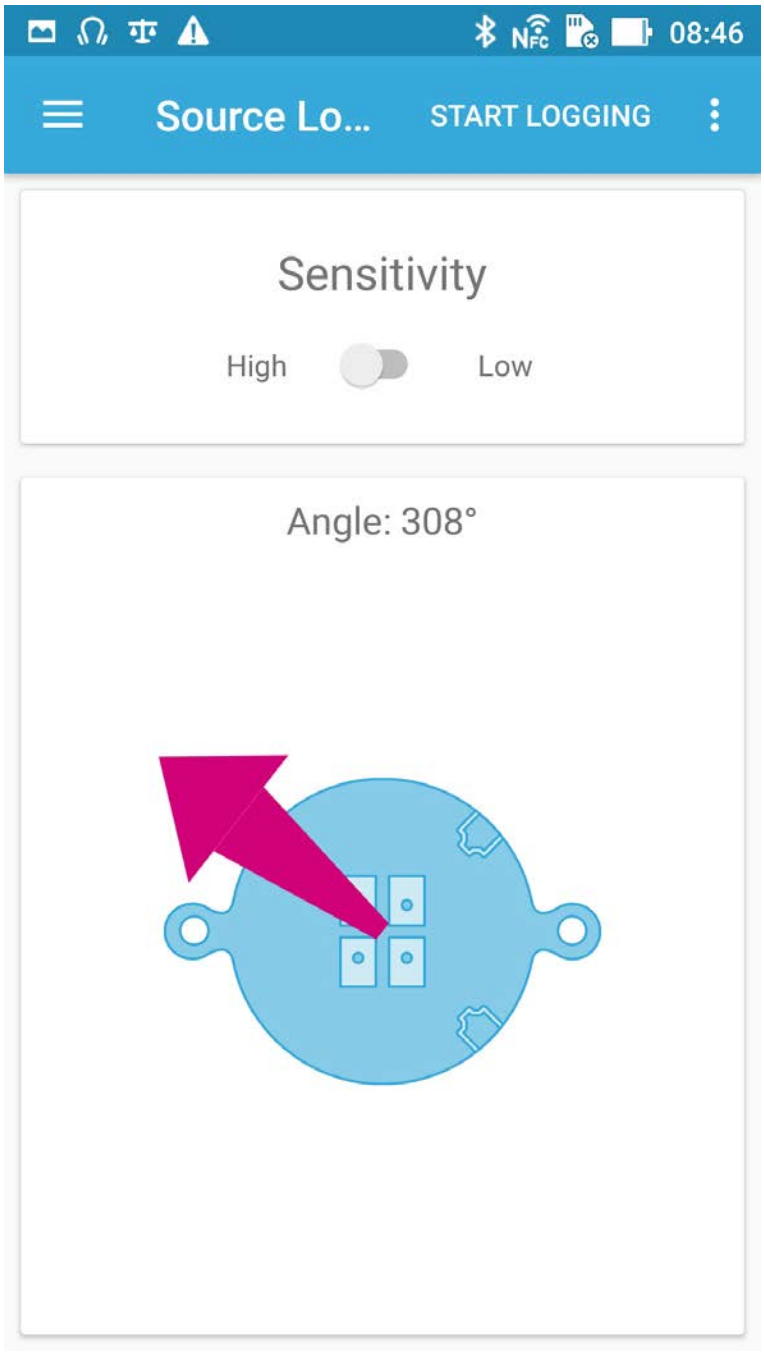
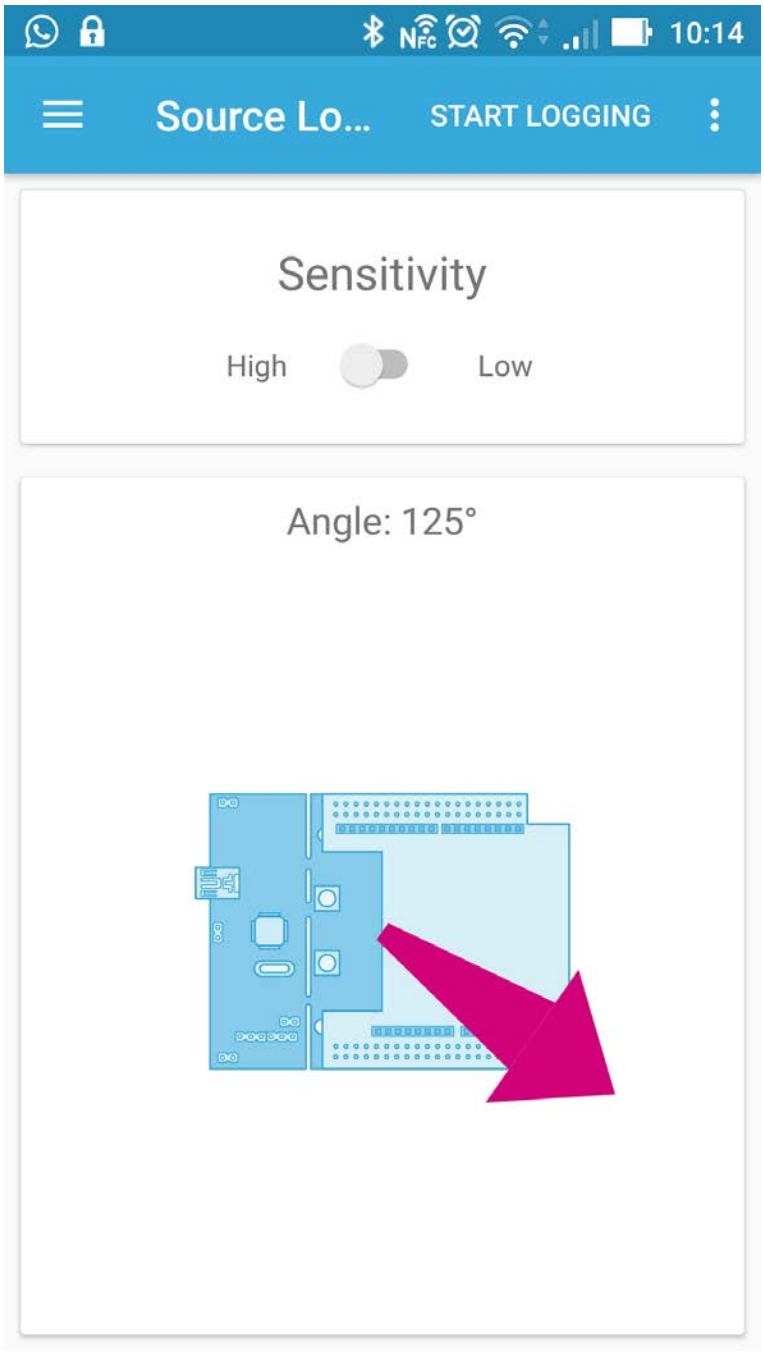


Figure 47. **BlueMS (Android version) audio source localization STM32 Nucleo Page**



In a noisy environment, use low sensitivity.

1.10.12 Beam Forming

If the AcousticBF library is enabled, the pages shown below are available (feature not available on STEVAL-STLKT01V1 and NUCLEO-L476RG):

Figure 48. **BlueMS (Android version) audio beam forming BlueCoin Page**

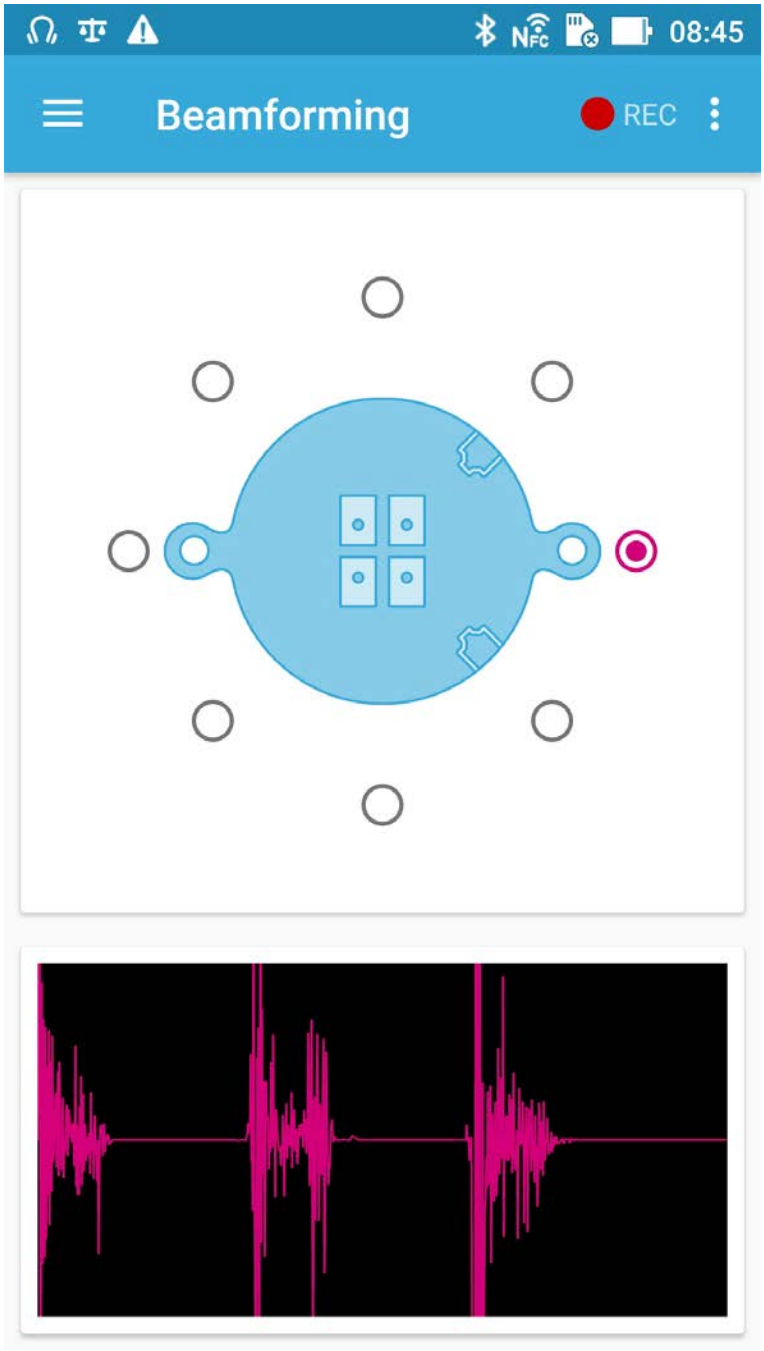
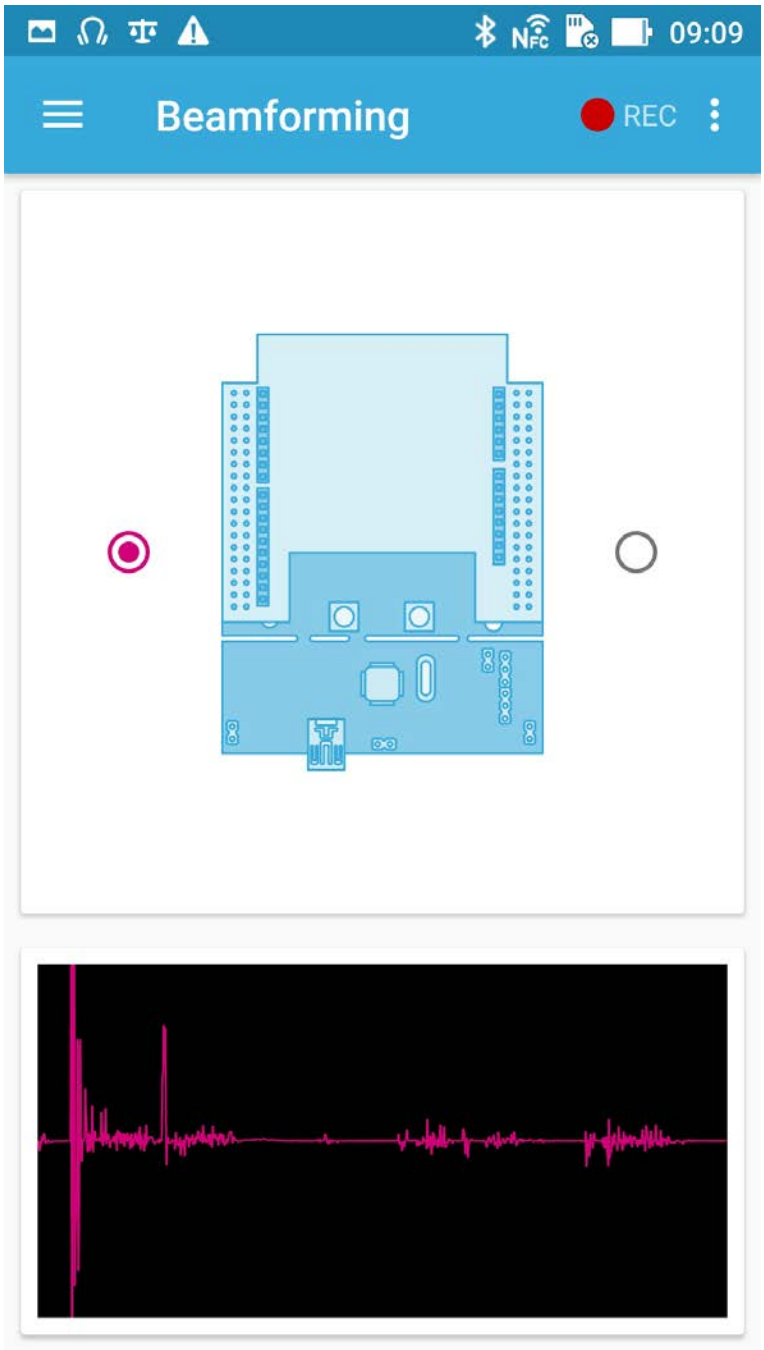


Figure 49. **BlueMS (Android version) audio beam forming STM32 Nucleo Page**



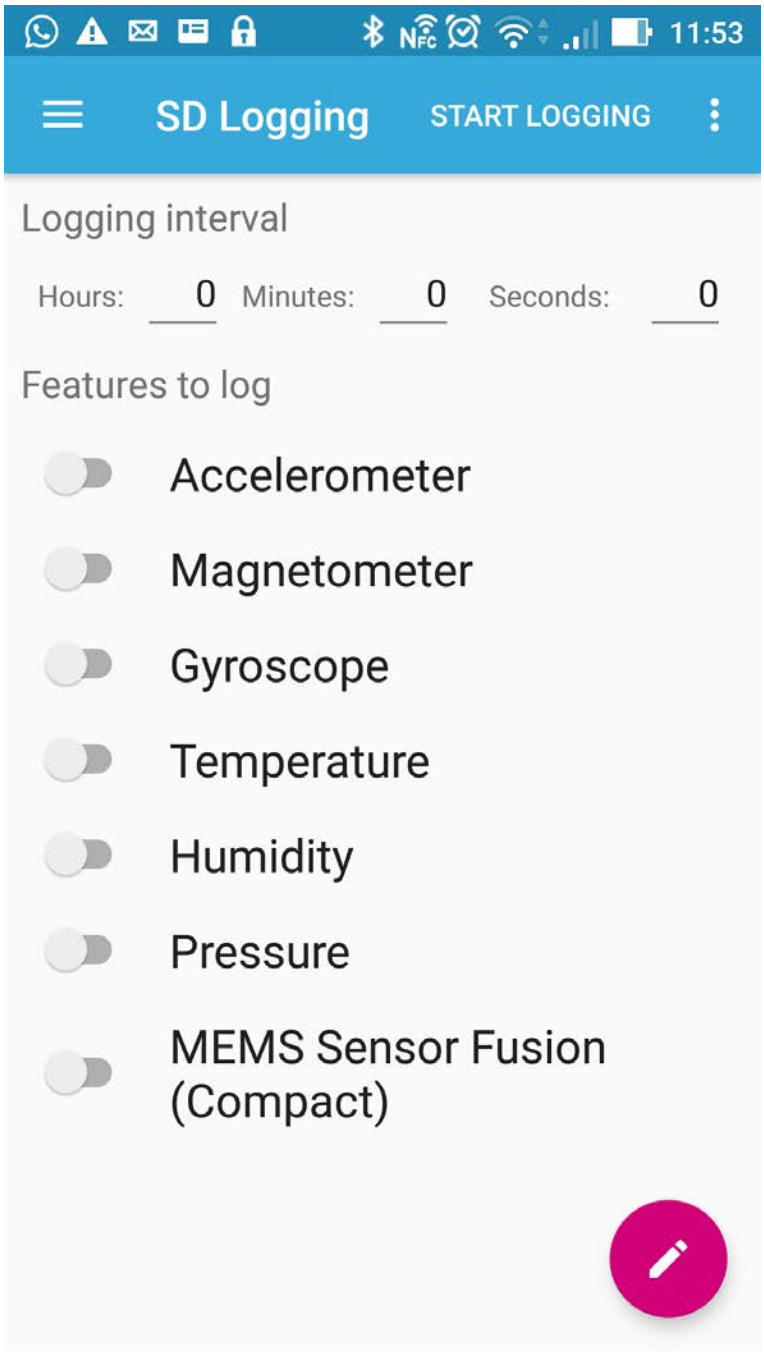
The AcousticBF provides real-time beam forming software, using the audio signals acquired from two digital MEMS microphones, it creates a virtual directional microphone pointing to a fixed direction in space.

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

1.10.13 SD Logging

This page shows SD Logging setting for STEVAL-STLKT01V1 only.

Figure 50. **BlueMS (Android version) SD Logging information**



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

If the logging interval is more then 20 seconds (when the Android/iOS device is not connected and the logging has started), the board enters in shutdown mode.

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

The RTC alarm is used to wake the board up to log the selected data with the logging interval chosen.

The accelerometer events can be selected and used to wake the board up and connect it to the Android/iOS device to stop the logging.

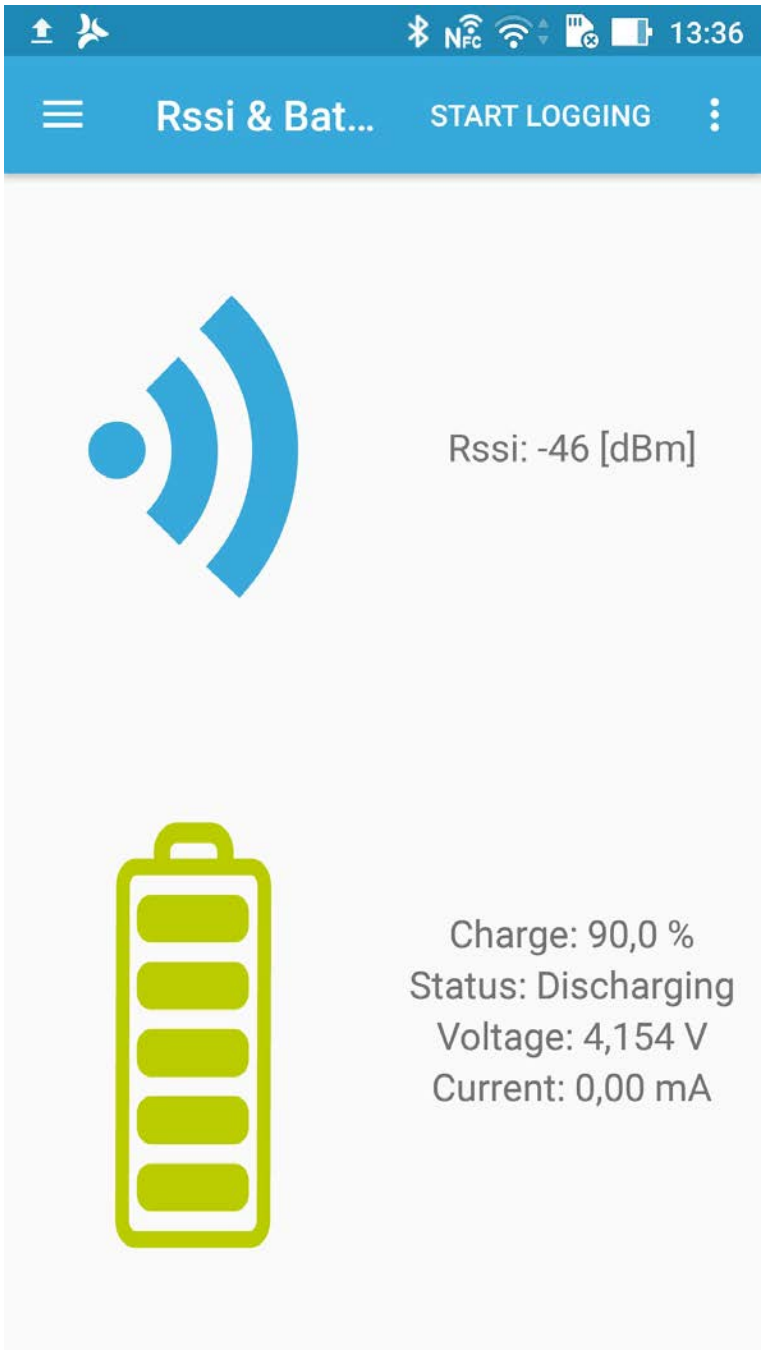
The wakeup sorce accelerometer event can be selected by setting the constant **WakeupSource** in the *main.c* file.

Note: The Double Tap event is set as default.

1.10.14 Rssi and battery

This page shows RSSI of the Bluetooth signal strength and, for STEVAL-STLKT01V1 and STEVAL-BCNKT01V1, if the battery is connected, the charge percentage, measured voltage and battery status (charging/discharging/low battery).

Figure 51. BlueMS (Android version) Battery and RSSI information



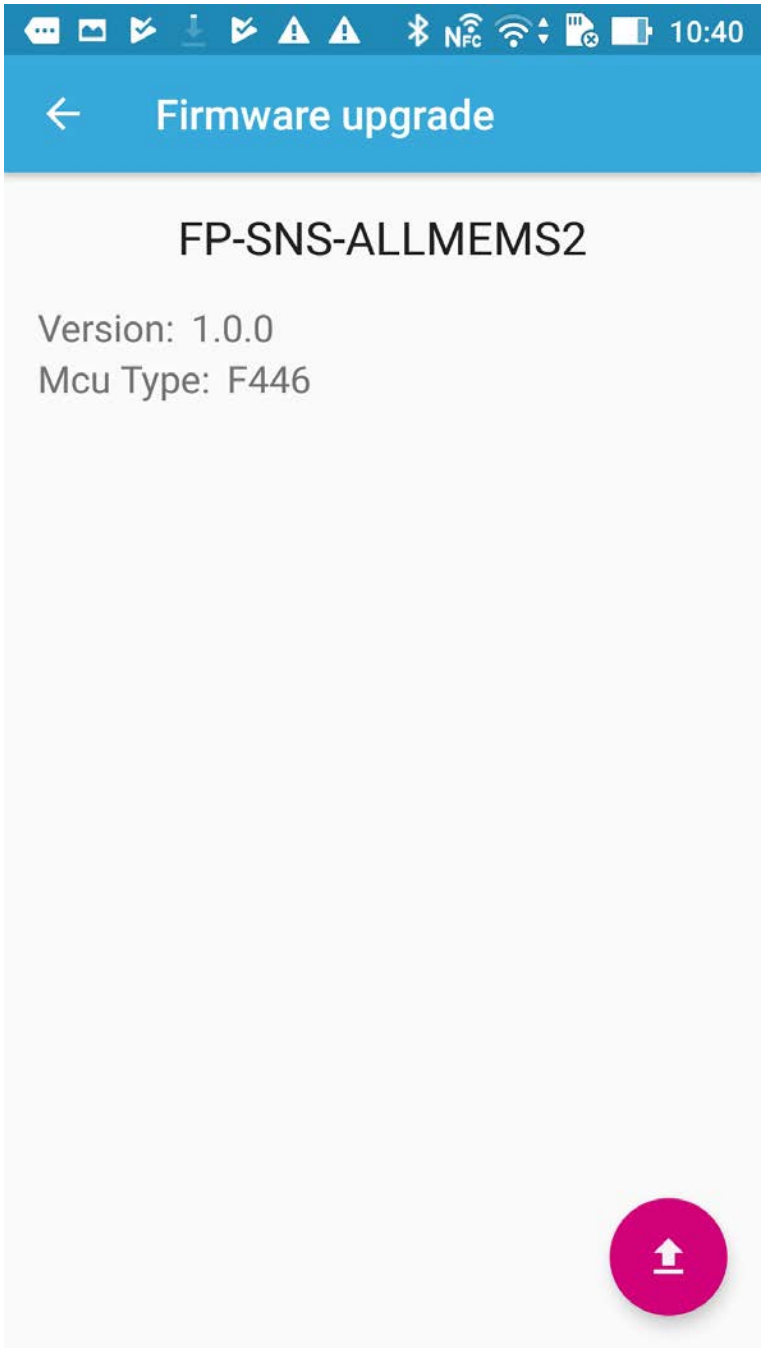
The RSSI value is updated every 0.5 seconds.

For STEVAL-BCNKT01V1 the current value is not available.

1.11 Firmware-Over-The-Air update with BlueMS

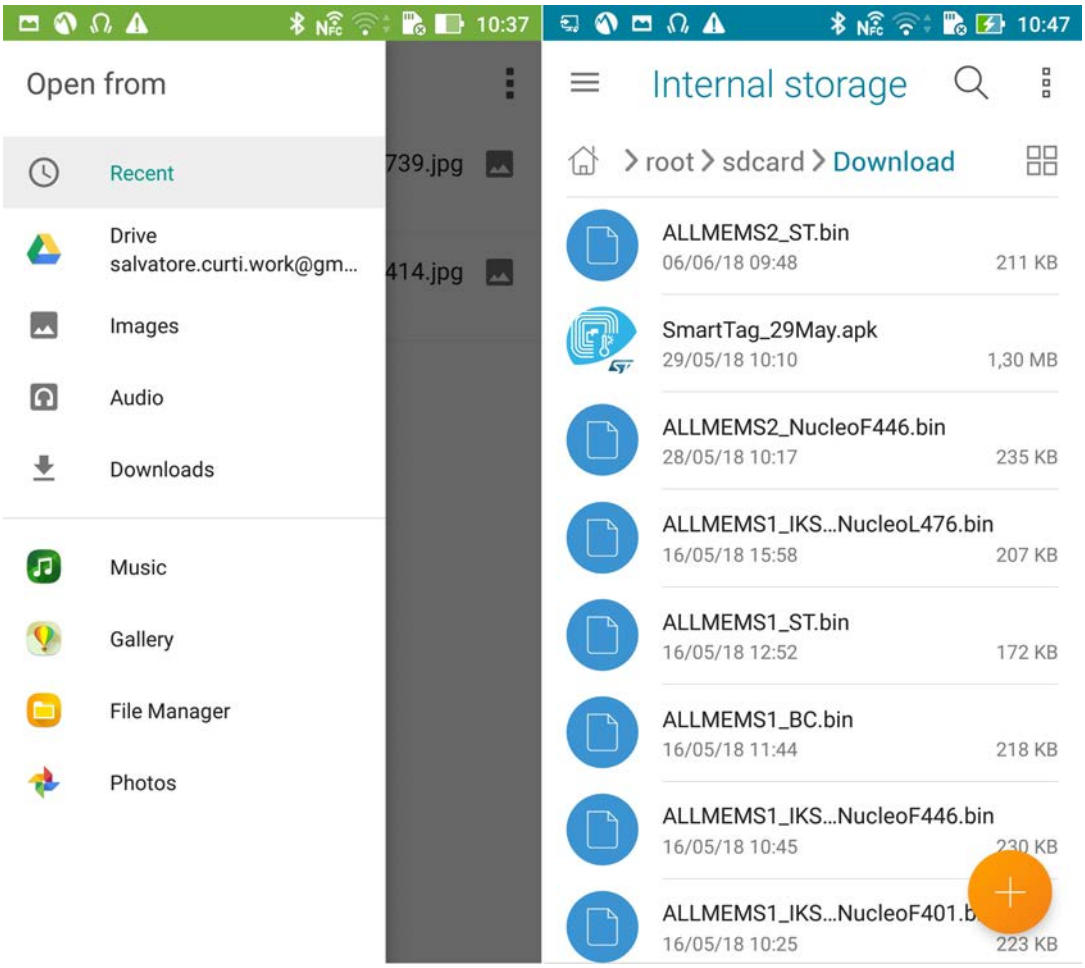
If the 'Firmware upgrade' option menu is selected, the following page appears.

Figure 52. BlueMS (Android version) firmware upgrade page



The BlueMS application shows which version of the FP-SNS-ALLMEMS2 software is running and the board type. To apply an update, press the red button and choose the appropriate file.

Figure 53. **BlueMS (Android version) firmware update file selection**

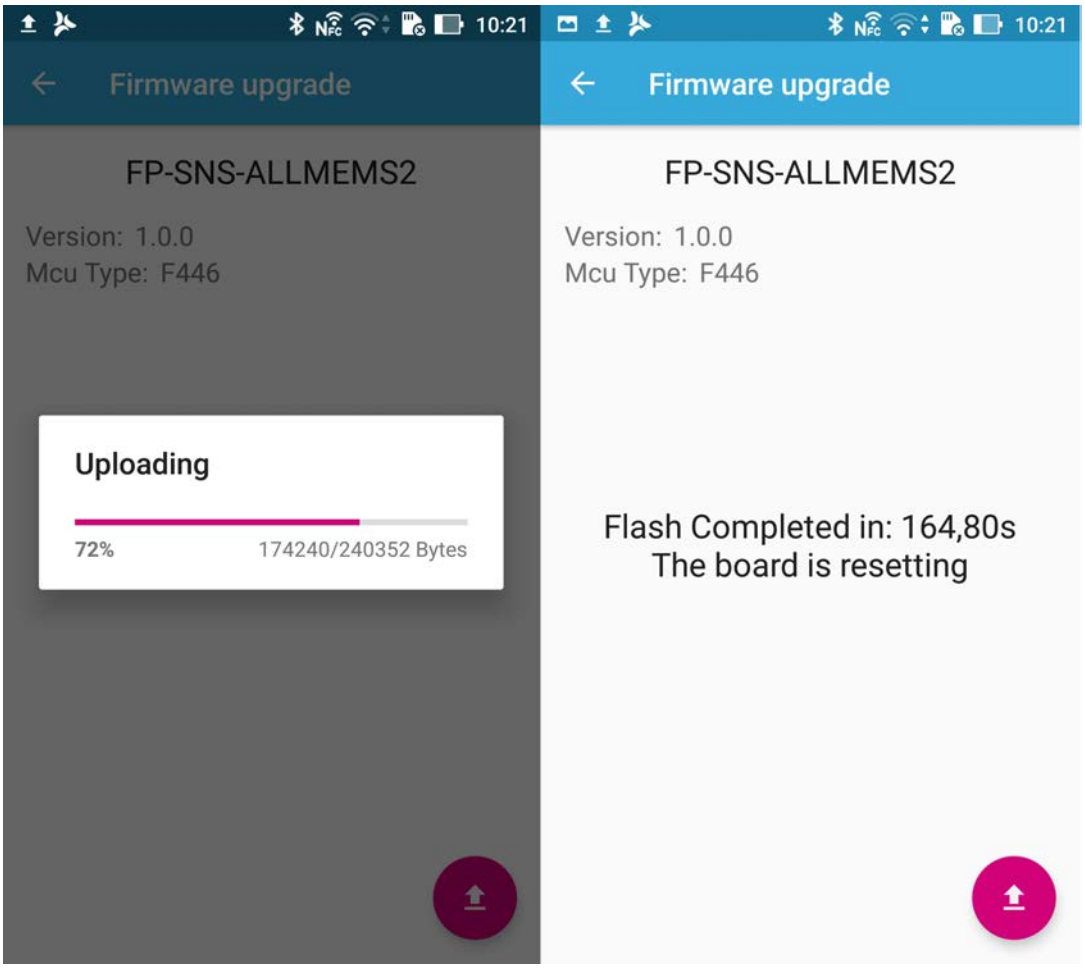


BlueMS sends the FP-SNS-ALLMEMS2 a command communicating that it is going to send an update of a certain byte size and corresponding CRC value.

Figure 4 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-ALLMEMS2 behavior.

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

Figure 54. **BlueMS (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 BlueMS application, FP-SNS-ALLMEMS2 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-ALLMEMS2 firmware.

Figure 55. **Terminal window feedback during FOTA**

1.12 System setup guide

1.12.1 Hardware description

1.12.1.1 *STM32 Nucleo platform*

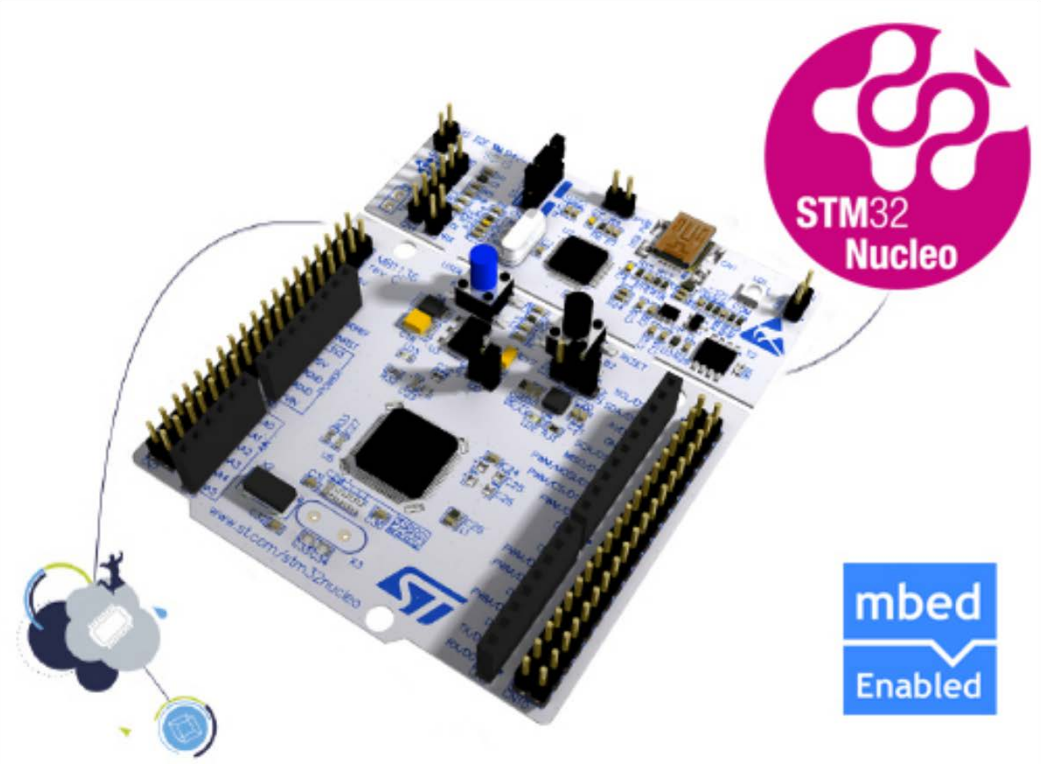
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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 56. **STM32 Nucleo board**



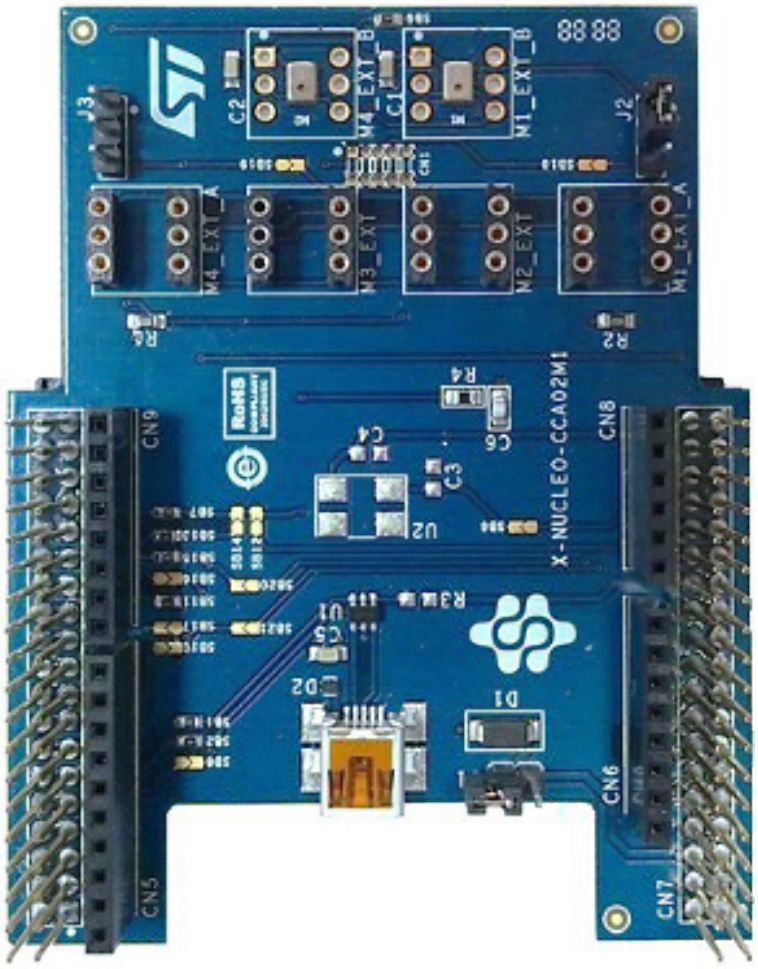
Information regarding the STM32 Nucleo board is available at www.st.com/stm32nucleo

1.12.1.2 *X-NUCLEO-CCA02M1 expansion board*

The X-NUCLEO-CCA02M1 is an expansion board based on digital MEMS microphones. It is compatible with the morpho connector layout, and is designed around STMicroelectronics MP34DT01-M digital microphones. There are two microphones soldered onto board and it offers the possibility to plug in additional microphones using MP32DT01 (or MP34DT01-M) based coupon evaluation board STEVAL-MKI129V3 (or STEVAL-MKI155V3).

The X-NUCLEO-CCA02M1 allows the acquisition of up to two microphones using the I²S bus and up to four coupon microphones using I²S and SPI together. In addition, it offers a USB output for the STM32 Nucleo board. It represents a fast and easy solution for the development of microphone-based applications as well as a starting point for audio algorithm implementation.

Figure 57. **X-NUCLEO-CCA02M1 expansion board**

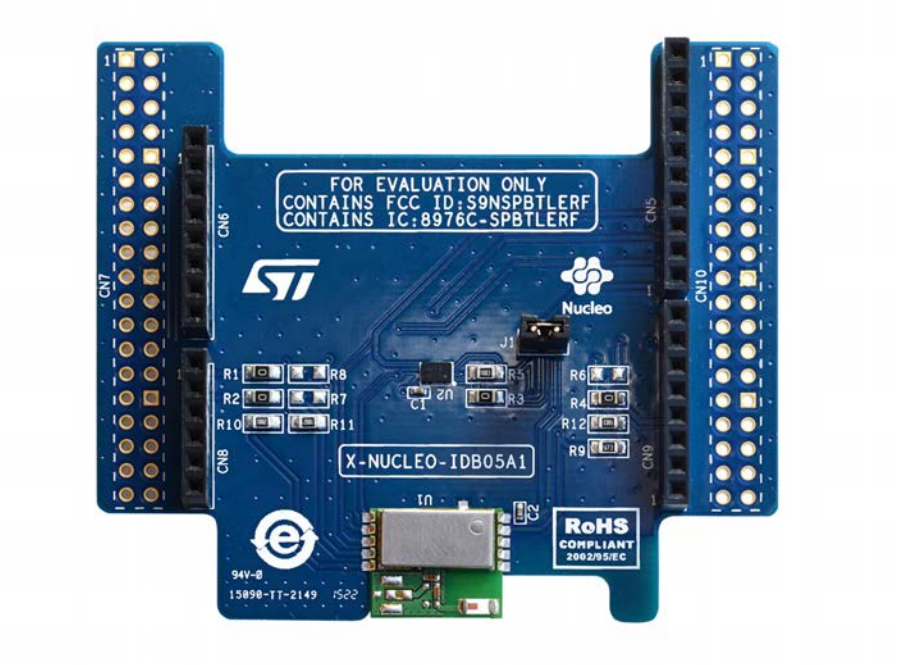


Information regarding the X-NUCLEO-CCA02M1 expansion board is available on www.st.com at http://www.st.com/x-nucleo.

1.12.1.3 *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 58. **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

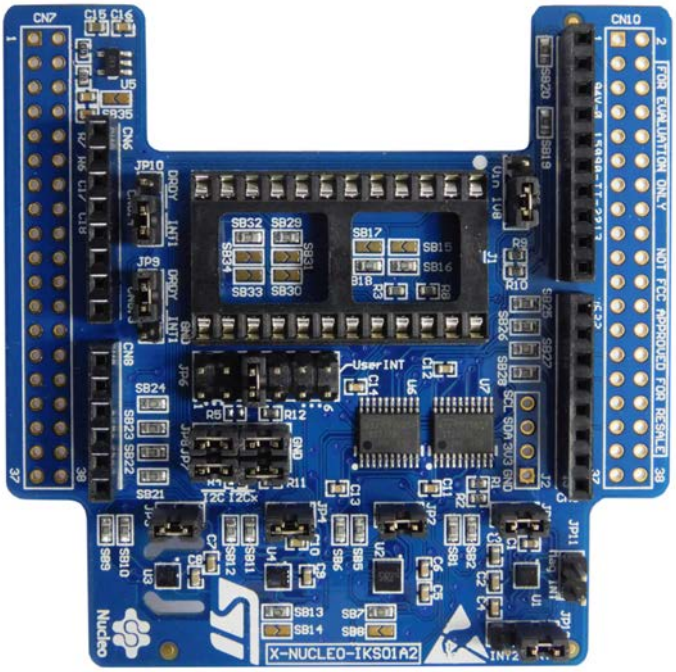
1.12.1.4 *X-NUCLEO-IKS01A2 expansion board*

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



1.12.1.5 *STEVAL-BCNKT01V1 BlueCoin development kit*

1.12.1.5.1 Description

The STEVAL-BCNKT01V1 integrated development and prototyping platform for augmented acoustic and motion sensing for IoT applications builds on the listening and balancing capabilities of the human ear.

With the expanded capabilities of its starter kit, BlueCoin lets you explore advanced sensor fusion and signal processing functions for robotics and automation applications with a 4 digital MEMS microphone array, a high-performance 9-axis inertial and environmental sensor unit and time-of-flight ranging sensors.

A high-performance STM32F446 180 MHz MCU enables real-time implementation of the very advanced sensor fusion algorithms like adaptive beamforming and sound source localization, with ready-to-use, royalty-free building blocks.

The BlueCoin can connect via the on-board BLE link to any IoT and smart industry wireless sensor network.

To upload new firmware onto the BlueCoin an external SWD debugger (not included in the starter-kit) is needed. It is recommended to use the ST-Link V2.1 found on any "STM32 Nucleo-64" development board.

1.12.1.5.2 Features

- Contains FCC ID: S9NBCOIN01
- Contains module IC 8976C-BCOIN01 certified with PMN: STEVAL-BCNKT01V1; HVIN: STEVAL-BCNCS01V1; HMN: STEVAL-BCNCR01V1; FVIN: bluenrg_7_2_c_Mode_2-32MHz-XO32K_4M.img
- The development kit package includes:
  - BlueCoin module (STEVAL-BCNCS01V1) with STM32F446, LSM6DSM, LSM303AGR, LPS22HB, 4x MP34DT04-C1, BlueNRG-MS, BALF-NRG-01D3, STBC03JR
  - CoinStation (STEVAL-BCNST01V1) board
  - BlueCoin Cradle (STEVAL-BCNCR01V1)
  - 130 mAh Li-Po battery
  - Plastic box for housing the BlueCoin cradle and the battery
  - SWD programming cable
- Software libraries and tools:

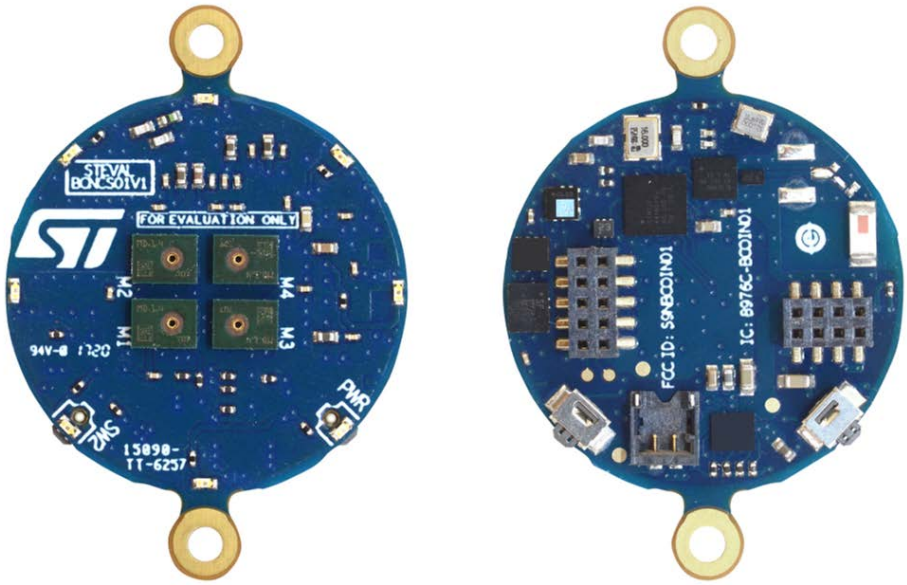
- STSW-BCNKT01 firmware package with raw sensor data streaming support via USB, data logging on SD card, audio acquisition and audio streaming, time-of-flight example and BLE protocol to interface to a smartphone app
- FP-AUD-SMARTMIC1: smart audio IN-OUT software expansion for STM32Cube
- FP-SNS-ALLMEMS1 and FP-SNS-ALLMEMS2: STM32Cube function packs for BLE and sensors
- FP-AUD-BVLINK1: BLE and microphones software expansion for STM32Cube
- BlueMS: iOS™ and Android™ demo apps
- BlueST-SDK: iOS and Android software development kit
- Compatible with STM32 ecosystem through STM32Cube support

1.12.1.5.3 Content of the starter kit

STEVAL-BCNCS01V1 - BlueCoin Core System board features

- Very compact module for motion, audio and environmental sensing and Bluetooth low energy connectivity with a complete set of firmware examples
- Main components:
  - STM32F446 – 32-bit high-performance MCU (ARM® Cortex®-M4 with FPU)
  - 4x MP34DT04-C1 – 64dB SNR Digital MEMS microphone
  - LSM6DSM – iNEMO inertial module: 3D accelerometer and 3D gyroscope
  - LSM303AGR – ultra-compact high-performance eCompass module: ultra-low power 3D accelerometer and 3D magnetometer
  - LPS22HB – MEMS nano pressure sensor: 260-1260 hPa absolute digital output barometer
  - BlueNRG-MS – Bluetooth low energy network processor
  - BALF-NRG-01D3 – 50 Ω balun with integrated harmonic filter
  - STBC03JR – linear battery charger with 150 mA LDO 3.0 V
- External interfaces: UART, SPI, SAI (Serial Audio Interface), I²C, USB OTG, ADC, GPIOs, SDIO, CAN, I2S
- SWD interface for debugging and programming capability
- The Bluetooth radio power output is set by default to 0 dBm; the FCC and IC certifications refer to this operating value. The power output can be changed up to 8 dBm by reprogramming the device firmware, but this change will require an update of the FCC and IC certifications, with additional radio emission tests to be performed.

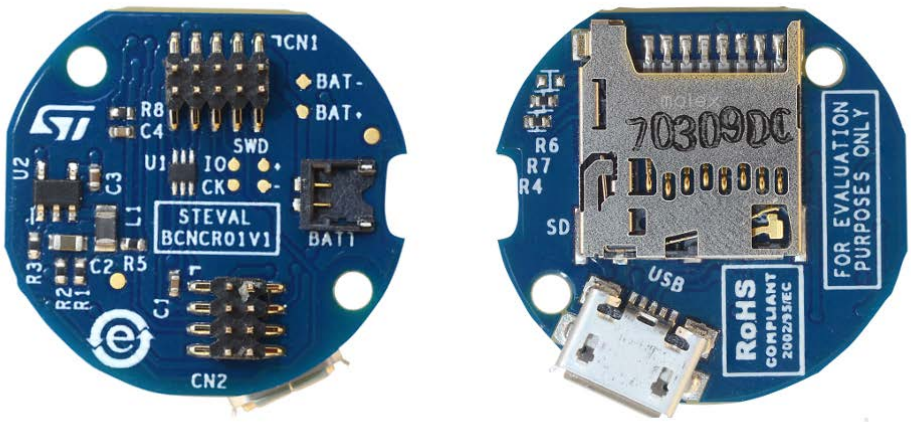
Figure 60. **STEVAL-BCNCS01V1 - BlueCoin Core System**



STEVAL-BCNCR01V1 - BlueCoin Cradle board features

- BlueCoin Cradle board with BlueCoin connectors
- ST1S12XX – 3.3 V step down DC-DC converter
- USBLC6-2P6 – very low capacitance ESD protection
- USB type A to Mini-B USB connector for power supply and communication
- microSD card socket

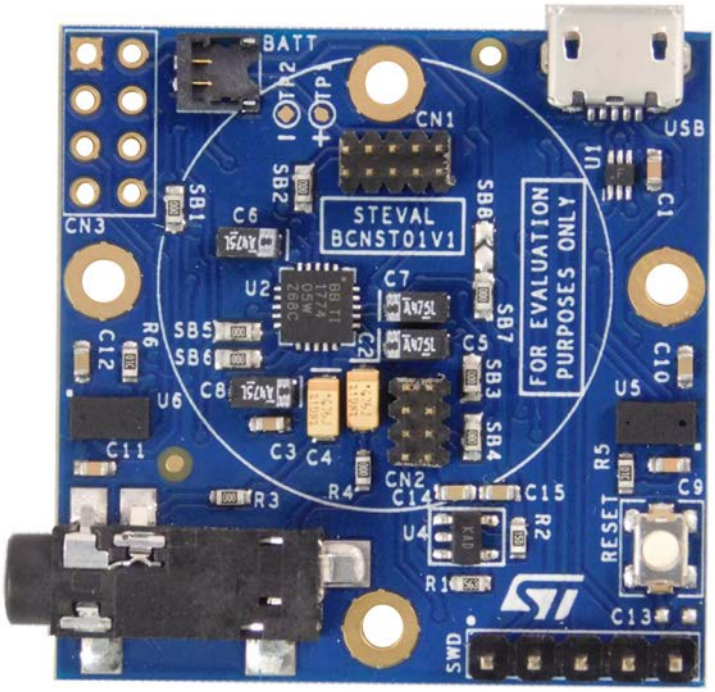
Figure 61. **STEVAL-BCNCR01V1 - BlueCoin Cradle board**



STEVAL-BCNST01V1 - CoinStation board features

- CoinStation expansion board with BlueCoin connectors
- LDK120M-R – 200 mA low quiescent current very low noise LDO
- USBLC6-2P6 – very low capacitance ESD protection for USB
- 2x VL53L0X Time-of-Flight (ToF) ranging sensor
- 16-Bit, low-power stereo audio DAC and 3.5 mm jack socket
- Micro-USB connector for power supply and communication
- Reset button
- SWD connector for programming and debugging

Figure 62. STEVAL-BCNST01V1 - CoinStation board



1.12.1.6 *STEVAL-STLKT01V1 SensorTile development kit*

1.12.1.6.1 Description

The STEVAL-STLKT01V1 is a comprehensive development kit designed to support and expand the capabilities of the SensorTile and comes with a set of cradle boards enabling hardware scalability. The development kit simplifies prototyping, evaluation and development of innovative solutions. It is complemented with software, firmware libraries and tools, including a dedicated mobile App.

The SensorTile is a tiny, square-shaped IoT module that packs powerful processing capabilities leveraging an 80 MHz STM32L476JG microcontroller and Bluetooth low energy connectivity based on BlueNRG-MS network processor as well as a wide spectrum of motion and environmental MEMS sensors, including a digital microphone.

SensorTile can fit snugly in your IoT hub or sensor network node and become the core of your solution.

To upload new firmware onto the SensorTile, an external

SWD debugger (not included in the kit) is needed. It is recommended to use ST-LINK/V2-1 found on any STM32 Nucleo-64 development board.

1.12.1.6.2 Features

- Included in the development kit package:
  - SensorTile module (STEVAL-STLCS01V1) with STM32L476, LSM6DSM, LSM303AGR, LPS22HB, MP34DT04, BlueNRG-MS, BALF-NRG-01D3 and LD39115J18R

  - SensorTile expansion Cradle board equipped with audio DAC, USB port, STM32 Nucleo, Arduino UNO R3 and SWD connector
  - SensorTile Cradle with battery charger, humidity and temperature sensor, SD memory card slot, USB port and breakaway SWD connector
  - 100 mAh Li-Ion battery
  - Plastic box
  - SWD programming cable
- Software libraries and tools
  - STSW-STLKT01: SensorTile firmware package that supports sensors raw data streaming via USB, data logging on SDCard, audio acquisition and audio streaming.
  - FP-SNS-ALLMEMS1 and FP-SNS-ALLMEMS2: STM32Cube functional packs
  - BlueMS: iOS and Android demo Apps
  - BlueST-SDK: iOS and Android Software Development Kit
- CE certified
- RoHS and China RoHS compliant
- FCC (ID: S9NSTILE01) certified
- IC (IC: 8976C-STILE01) certified with PMN: STEVAL-STLKT01V1; HVIN: STEVAL-STLCS01V1; HMN: STEVAL-STLCX01V1; FVIN: bluenrg_7_1_e_Mode_2-32MHz-XO32K_4M.img
- TYPE certified (006-000482)

1.12.1.6.3 Boards included in the kit

Figure 63. STLCS01V1 board photo

STLCS01V1 SensorTile component board features

- Very compact module for motion, audio and environmental sensing and Bluetooth low energy connectivity with a complete set of firmware examples
- Supported by the STM32Cube and the STM32Cube functional packs FP-SNS-ALLMEMS1 and FP-SNS-ALLMEMS2
- Mobile connectivity via the ST BlueMS app, available for iOS and Android
- Main components:
  - STM32L476 – 32-bit ultra-low-power MCU with CortexM4F
  - LSM6DSM – iNEMO inertial module: 3D accelerometer and 3D gyroscope

  - LSM303AGR – Ultra-compact high-performance eCompass module: ultra-low power 3D accelerometer and 3D magnetometer
  - LPS22HB – MEMS nano pressure sensor: 260-1260 hPa absolute digital output barometer
  - MP34DT04 – 64dB SNR Digital MEMS Microphone
  - BlueNRG-MS – Bluetooth low energy network processor
  - BALF-NRG-01D3 – 50 Ω balun with integrated harmonic filter
  - LD39115J18R – 150 mA low quiescent current low noise LDO 1.8 V
  - 2 V-5.5 V power supply range
  - External interfaces: UART, SPI, SAI (Serial Audio Interface), I²C, DFSDM, USB OTG, ADC, GPIOs
- Pluggable or solderable interface
- SWD interface for debugging and programming capability

STLCS01V1 SensorTile component board description

STEVAL-STLCS01V1 (SensorTile) is a highly integrated reference design that can be plugged into form-factor prototypes to add sensing and connectivity capabilities to new designs through a smart hub solution. It can also easily support development of monitoring and tracking applications as standalone sensor node connected to iOS/ Android smartphone applications.

The SensorTile comes in a very small square shape 13.5 x 13.5 mm. All the electronic components are on the top side of the pcb, while the bottom side has a small connector through which it is possible to easily plug and unplug it from a motherboard. The connector pinout is also replicated on 18 pcb pads that render the SensorTile a solderable system on module as well.

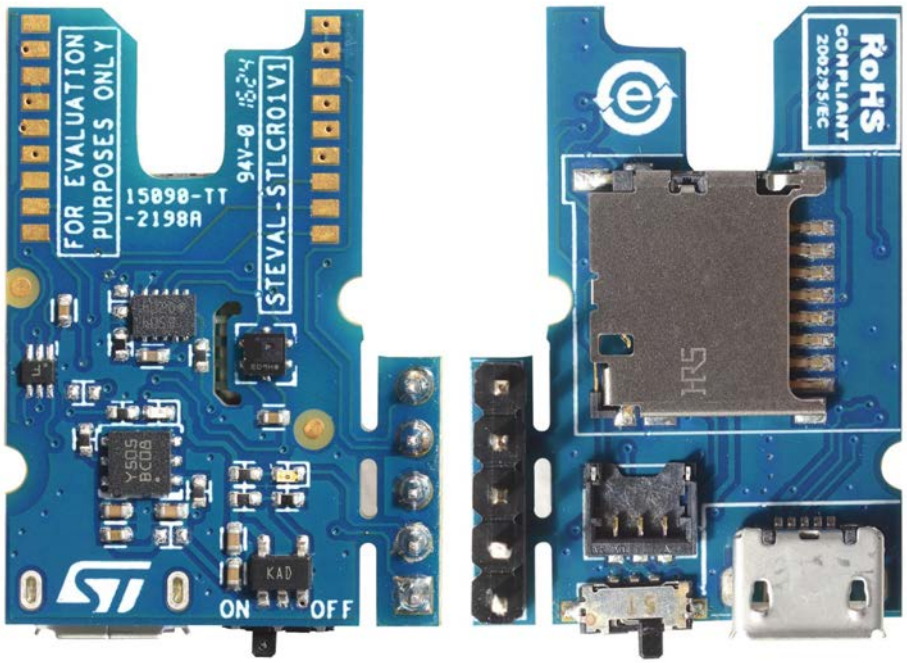
The module comes with pre-loaded FP-SNS-ALLMEMS1 (former BLUEMICROSYSTEM2) software that initializes all the sensors and the Bluetooth low energy radio. The “ST BlueMS” app, available free of charge on Apple Store™ and Google Play™, is the easiest and fastest way to start using the SensorTile board and to experience a real activity monitoring system.

The SensorTile firmware package STSW-STLKT01, built on the STM32Cube software technology, includes all the low level drivers to manage the on-board devices and system-level interfaces. It has been designed in order to be easily extended and personalized as starting point for development and customization of new dedicated applications.

All the firmware packages are freely available on www.st.com.

The Bluetooth radio power output is set by default at 0 dBm. The FCC and IC certifications refer to this operating value. The power output can be changed up to 8 dBm by reprogramming the device firmware, but the change of this operating value will require an update of the FCC and IC certifications, with additional radio emission tests to be performed.

Figure 64. **STLCR01V1 board photo**



STLCR01V1 SensorTile component board features

- Sensortile Cradle board with SensorTile footprint (solderable)
- STBC08PMR – 800 mA standalone linear Li-Ion battery charger
- HTS221 – capacitive digital sensor for relative humidity and temperature
- LDK120M-R – 200 mA low quiescent current very low noise LDO
- STC3115 – Fuel gauge IC
- USBLC6-2P6 – very low capacitance ESD protection
- USB type A to Mini-B USB connector for power supply and communication
- microSD card socket
- SWD connector for programming and debugging

Figure 65. **STLCX01V1 board photo**

STLCX01V1 SensorTile component board features

- Sensortile Cradle expansion board with SensorTile plug connector
- Compatible with STM32 Nucleo boards through Arduino UNO R3 connector
- LDK120M-R – 200 mA low quiescent current very low noise LDO
- ST2378ETTR – 8-bit dual supply 1.71 V to 5.5 V level translator
- USBLC6-2P6 – very low capacitance ESD protection
- 16-Bit, low-power stereo audio DAC
- Micro-USB connector for power supply and communication
- Reset button
- SWD connector for programming and debugging

1.12.2 Software description

The following software components are needed in order 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-STLKT01V1:

- FP-SNS-ALLMEMS2: Bluetooth low energy and sensors software for STM32Cube. FP-SNS-ALLMEMS2 firmware and related documentation is available on www.st.com.
- 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
  - System Workbench for STM32 + ST-LINK

1.12.3 Hardware and software setup

1.12.3.1 *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-CCA02M1)

  - One sensor expansion board (order code X-NUCLEO-IKS01A2)
  - One BlueNRG Bluetooth low energy expansion board (order code: X-NUCLEO-IDB05A1)
  - One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
- for STEVAL-STLKT01V1:
  - STEVAL-STLKT01V1 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-STLKT01V1 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

1.12.3.2 *Software setup*

This section describes how to set up different hardware parts before writing and executing an application:

- on the STM32 Nucleo board with the expansion boards
- on the STEVAL-STLKT01V1 development kit
- on the STEVAL-BCNKT01V1 development kit

1.12.3.2.1 Development tool-chains and compilers

Select one of the Integrated Development Environments supported by the STM32Cube expansion software and follow the system requirements and setup information provided by the selected IDE provider.

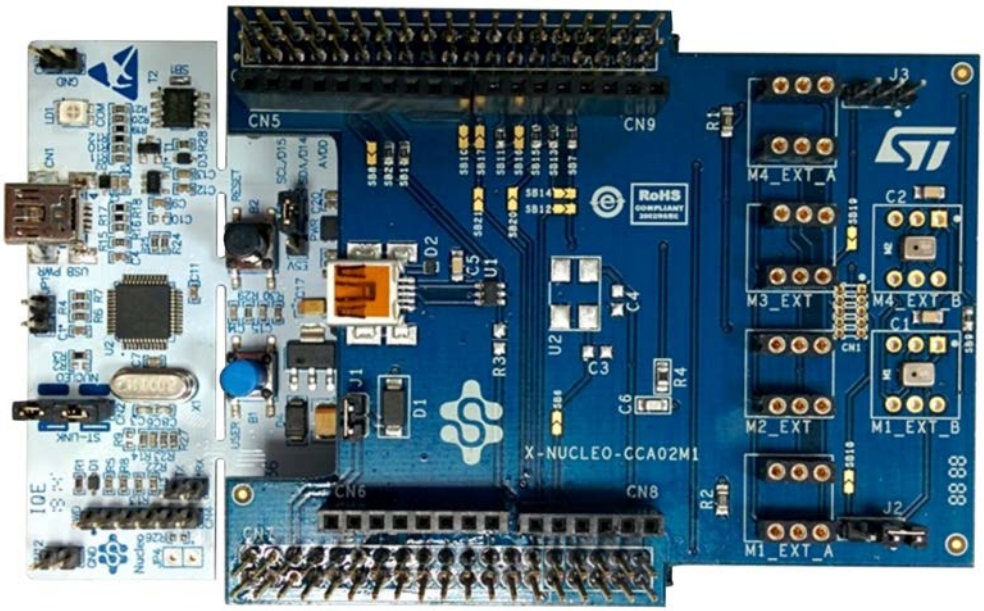
1.12.3.3 *System setup guide*

1.12.3.3.1 STM32 Nucleo and 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 at STSW-LINK008 or STSW-LINK009.

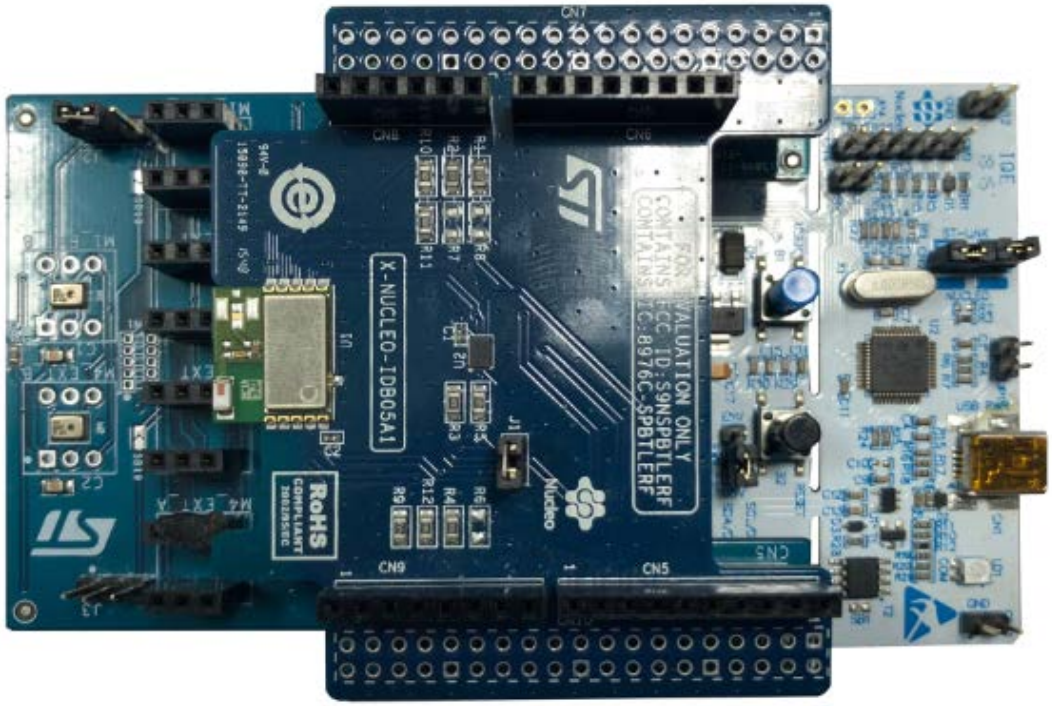
The X-NUCLEO-CCA02M1 sensor board is easily connected to the STM32 Nucleo board through the morpho connector, as shown below.

Figure 66. **STM32 Nucleo plus X-NUCLEO-CCA02M1 boards**



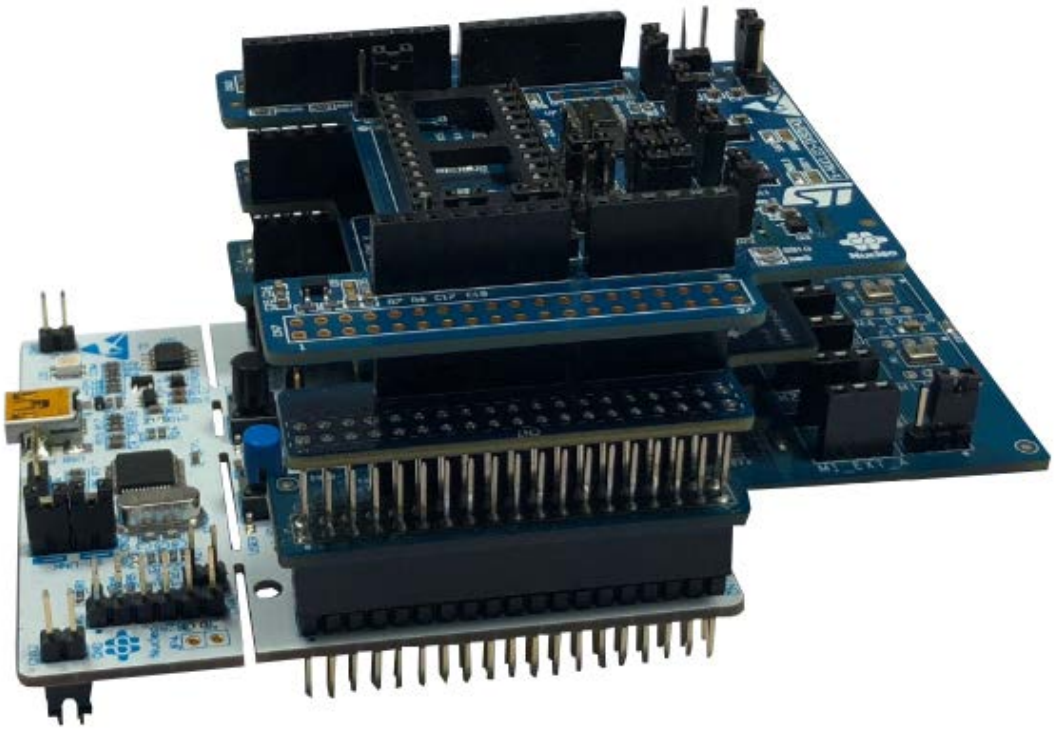
The X-NUCLEO-IDB05A1 BlueNRG BLE expansion board is easily connected to the X-NUCLEO-CCA02M1 board through the Arduino UNO R3 extension connector, as shown below.

Figure 67. **STM32 Nucleo plus X-NUCLEO-CCA02M1 plus X-NUCLEO-IDB05A1 boards**



Finally, the X-NUCLEO-IKS01A2 sensors board is easily connected to the X-NUCLEO-IDB05A1 expansion board through the Arduino UNO R3 extension connector, as shown below.

Figure 68. **STM32 Nucleo plus X-NUCLEO-CCA02M1 plus X-NUCLEO-IDB05A1 plus X-NUCLEO-IKS01A2 boards**



Note: The stacking sequence shown above is necessary to optimize the performance of the SPBTLE-RF module on the X-NUCLEO-IDB05A1 expansion board, and to reduce interference from its antenna.

1.12.3.3.2 STEVAL-STLKT01V1 setup

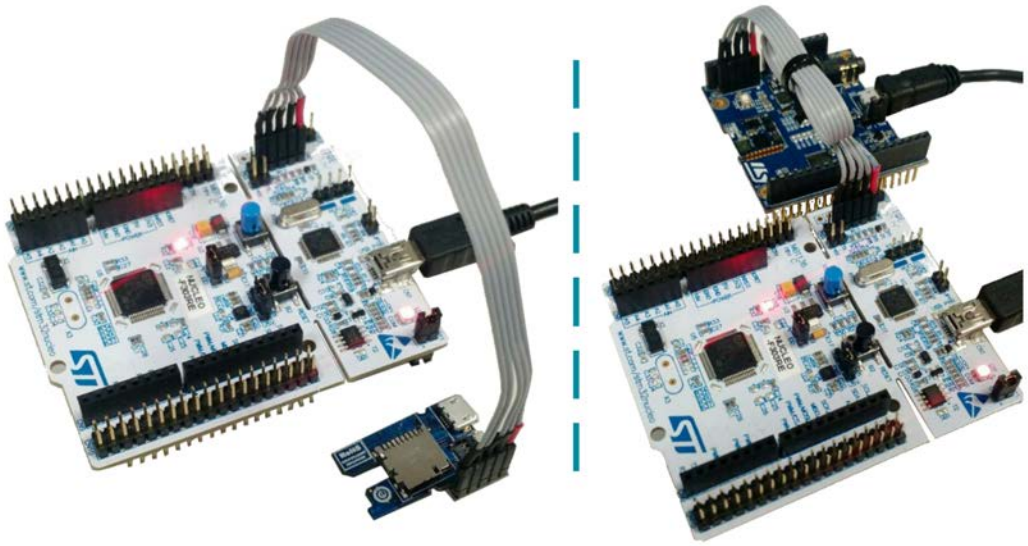
The ST-LINK/V2-1 debugger/programmer integrated on STM32 Nucleo board must be used to program the STEVAL-STLCS01V1 (SensorTile). The developer can download the relevant version of the ST-LINK/V2-1 USB driver at STSW-LINK008 or STSW-LINK009.

Connect STEVAL-STLCS01V1 (SensorTile) on the STEVAL-STLCR01V SensorTile Cradle board or on the STEVAL-STLCX01V1 Sensortile Cradle Expansion board.

Use the SWD connector to connect the Sensortile Cradle 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 SensorTile Cradle through the provided cable paying attention to the polarity of the connectors. Pin 1 can be identified by a little circle on the pcb silkscreen (STM32 Nucleo board and SensorTile Cradle Expansion) or by the square shape of the soldering pad of the connector (SensorTile Cradle).

Figure 69. **SensorTile Cradle expansion board and Sensor Tile Cradle board connected to ST-LINK/V2-1**



1.12.3.3.3 STEVAL-BCNKT01V1 setup

The ST-LINK/V2-1 debugger/programmer integrated on STM32 Nucleo board must be used to program the STEVAL-BCNCS01V1 (BlueCoin). The developer can download the relevant version of the ST-LINK/V2-1 USB driver at STSW-LINK008 or STSW-LINK009.

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 STM32Nucleo board and Coin Station PCB silkscreens or by the square shape of the soldering pad of the connector (SensorTile Cradle).

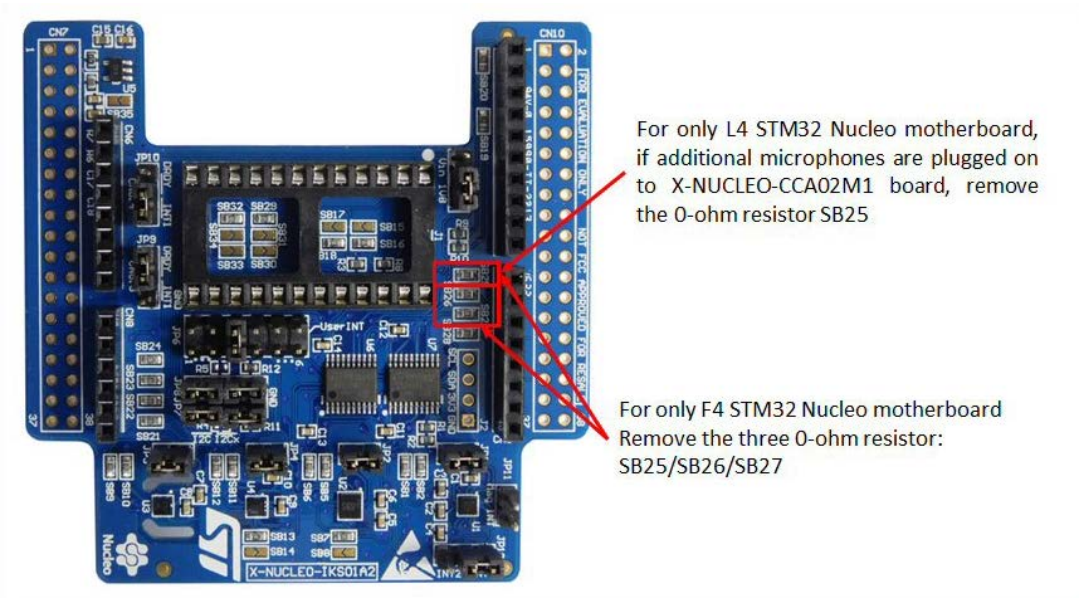
Figure 70. **BlueCoin-STM32Nucleo: SWD connections with 5-pin flat cable**

1.12.3.3.4 Important additional hardware information

For the STM32 Nucleo board: before connecting the X-NUCLEO-IKS01A2 to the X-NUCLEO-CCAM02M1 expansion board through the Arduino UNO R3 extension connector, remove these 0-Ω resistors on the X-NUCLEO-IKS01A2 board:

- for F4 STM32 Nucleo motherboard remove SB25, SB26 and SB27
- for L4 STM32 Nucleo motherboard remove SB25 if additional microphones are plugged on to X-NUCLEO-CCA02M1 board.

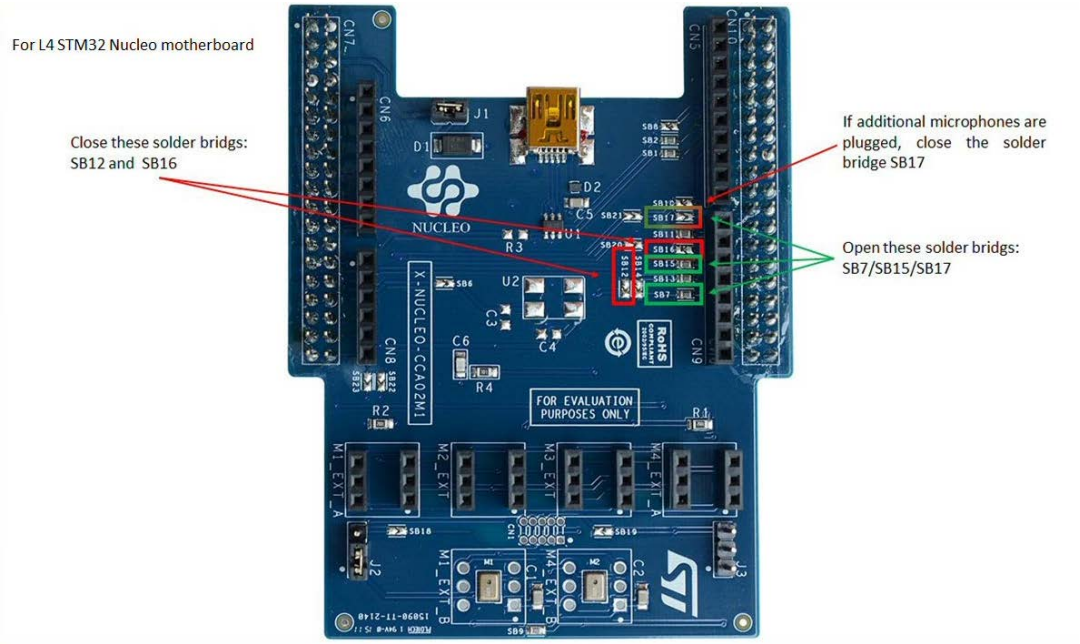
Figure 71. **X-NUCLEO-IKS01A2 solder bridge configuration**



For the NUCLEO-L476RG board only: before connecting the X-NUCLEO-CCA02M1 board to the STM32 Nucleo L4-series development board through the ST morpho connector layout, on the X-NUCLEO-CCA02M1 board:

- close the solder bridges SB12, SB16 and open the solder bridges SB7, SB15 and SB17
- if additional microphones are plugged, close the solder bridge SB17.

Figure 72. X-NUCLEO-CCA02M1 solder bridge configuration for the NUCLEO-L476RG board



Revision history

Table 1. **Document revision history**

Date	Version	Changes
09-Jul-2018	1	Initial release.

Contents

1 **FP-SNS-ALLMEMS2 software description** 2
1.1 Overview 2
1.2 Architecture 3
1.3 Folder structure 4
1.4 Flash management 5
1.5 The boot process 6
1.6 The installation process 7
1.7 Firmware-Over-The-Air (FOTA) update 11
1.8 APIs 11
1.9 Sample application description 11
1.10 Android and iOS sample client application 15
1.10.1 Main page 15
1.10.2 MEMS sensor fusion 16
1.10.3 Plot data 18
1.10.4 Settings, serial and debug console 19
1.10.5 Enable hardware features 23
1.10.6 Activity recognition 26
1.10.7 Carry position 27
1.10.8 Gesture recognition 28
1.10.9 E-compass 29
1.10.10 BlueVoiceADPCM 30
1.10.11 Direction of Arrival 39
1.10.12 Beam Forming 43
1.10.13 SD Logging 45
1.10.14 Rssi and battery 47
1.11 Firmware-Over-The-Air update with BlueMS 47
1.12 System setup guide 51
1.12.1 Hardware description 51
1.12.2 Software description 62
1.12.3 Hardware and software setup 62

Revision history 69

List of tables

Table 1. Document revision history 69

List of figures

Figure 1. FP-SNS-ALLMEMS2 software architecture. . . . 4
Figure 2. FP-SNS-ALLMEMS1 package folder structure . . . 4
Figure 3. FP-SNS-ALLMEMS2 FLASH structure. . . . 6
Figure 4. BootLoader folder content . . . 7
Figure 5. FP-SNS-ALLMEMS2 boot sequence . . . 7
Figure 6. Binary folder content . . . 8
Figure 7. Content of a project folder . . . 8
Figure 8. BootLoader and ALLMEMS2 installation . . . 9
Figure 9. ALLMEMS2 Dump process . . . 10
Figure 10. Terminal setting . . . 12
Figure 11. Initialization phase . . . 13
Figure 12. UART console output when a device is connected to the board. . . . 15
Figure 13. BlueMS (Android version) main page following BLE connection . . . 16
Figure 14. BlueMS (Android version) MotionFX sensor fusion page . . . 17
Figure 15. BlueMS (Android version) popup windows . . . 18
Figure 16. BlueMS (Android version) accelerometer plot . . . 19
Figure 17. BlueMS (Android version) menu selection . . . 20
Figure 18. BlueMS (Android version) Settings, Node Configuration, Local Name . . . 21
Figure 19. BlueMS (Android version) Debug console (stdin/stdout/stderr) . . . 22
Figure 20. BlueMS (Android version) multiple hardware feature . . . 24
Figure 21. BlueMS (Android version) hardware feature menu. . . . 25
Figure 22. BlueMS (Android version) hardware feature examples . . . 26
Figure 23. BlueMS (Android version) MotionAR activity recognition page. . . . 27
Figure 24. BlueMS (Android version) MotionCP carry position recognition page . . . 28
Figure 25. BlueMS (Android version) MotionGR gesture recognition page . . . 29
Figure 26. BlueMS (Android version) MotionFX page . . . 30
Figure 27. BlueMS (Android version) BlueVoice start page . . . 31
Figure 28. BlueMS (Android version) ASR language selection . . . 32
Figure 29. BlueMS (Android version) Chinese ASR, iFlytek technology . . . 33
Figure 30. BlueMS (Android version) popup API key window . . . 33
Figure 31. BlueMS (Android version) ASR service enabled . . . 34
Figure 32. BlueMS (Android version) voice recording . . . 35
Figure 33. BlueMS (Android version) recognised voice text . . . 36
Figure 34. Google Chromium-dev: search group . . . 37
Figure 35. Google Chromium-dev: join group to post . . . 37
Figure 36. Google Chromium-dev: join the group . . . 37
Figure 37. Google Chromium-dev: create project . . . 37
Figure 38. Google Developers Console: new project. . . . 38
Figure 39. Google Developers Console: ASRProject. . . . 38
Figure 40. Google Developers Console: select API. . . . 38
Figure 41. Google API Manager: enable API . . . 38
Figure 42. Google API Manager: create API key . . . 39
Figure 43. Google API Manager: Android API key . . . 39
Figure 44. BlueMS (Android version) Direction of Arrival menu selection . . . 40
Figure 45. BlueMS (Android version) audio source localization plot example . . . 41
Figure 46. BlueMS (Android version) audio source localization BlueCoin Page. . . . 42
Figure 47. BlueMS (Android version) audio source localization STM32 Nucleo Page . . . 43
Figure 48. BlueMS (Android version) audio beam forming BlueCoin Page . . . 44
Figure 49. BlueMS (Android version) audio beam forming STM32 Nucleo Page . . . 45
Figure 50. BlueMS (Android version) SD Logging information . . . 46
Figure 51. BlueMS (Android version) Battery and RSSI information . . . 47
Figure 52. BlueMS (Android version) firmware upgrade page. . . . 48

Figure 53. BlueMS (Android version) firmware update file selection . . . 49
Figure 54. BlueMS (Android application) feedback during and after FOTA transmission . . . 50
Figure 55. Terminal window feedback during FOTA . . . 51
Figure 56. STM32 Nucleo board . . . 52
Figure 57. X-NUCLEO-CCA02M1 expansion board . . . 53
Figure 58. X-NUCLEO-IDB05A1 expansion board . . . 54
Figure 59. X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board . . . 55
Figure 60. STEVAL-BCNCS01V1 - BlueCoin Core System . . . 57
Figure 61. STEVAL-BCNCR01V1 - BlueCoin Cradle board . . . 57
Figure 62. STEVAL-BCNST01V1 - CoinStation board . . . 58
Figure 63. STLCS01V1 board photo . . . 59
Figure 64. STLCR01V1 board photo . . . 61
Figure 65. STLCX01V1 board photo . . . 62
Figure 66. STM32 Nucleo plus X-NUCLEO-CCA02M1 boards . . . 64
Figure 67. STM32 Nucleo plus X-NUCLEO-CCA02M1 plus X-NUCLEO-IDB05A1 boards . . . 64
Figure 68. STM32 Nucleo plus X-NUCLEO-CCA02M1 plus X-NUCLEO-IDB05A1 plus X-NUCLEO-IKS01A2 boards . . . 65
Figure 69. SensorTile Cradle expansion board and Sensor Tile Cradle board connected to ST-LINK/V2-1 . . . 66
Figure 70. BlueCoin-STM32Nucleo: SWD connections with 5-pin flat cable . . . 67
Figure 71. X-NUCLEO-IKS01A2 solder bridge configuration . . . 67
Figure 72. X-NUCLEO-CCA02M1 solder bridge configuration for the NUCLEO-L476RG board . . . 68