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 STBLESensor 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, STEVAL-MKSBOX1V1 and STEVAL-STLKT01V1 evaluation boards.

RELATED LINKS

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

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 STBLESensor 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, STEVAL-MKSBOX1V1 and STEVAL-STLKT01V1 evaluation boards and for X-NUCLEO-CCA02M2, X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A3) and X-NUCLEO-BNRG2A1 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, STEVAL-MKSBOX1V1 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 and STEVAL-MKSBOX1V1)
   - audio beam forming using the AcousticBF algorithm (feature available only forSTEVAL-BCNKT01V1)
   - voice over Bluetooth low energy using the BlueVoiceADPCM algorithm (feature not available on NUCLEO-F446RE, NUCLEO-L476RG and STEVAL-MKSBOX1V1)
   - SD data logging (audio and MEMS data) using Generic FAT File System middleware (feature available on the STEVAL-STLKT01V1 and STEVAL-MKSBOX1V1)
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 LSM6DSO expansion board (or for LSM6DSO on X-NUCLEO-IKS01A3) for STM32 Nucleo L4 only, or for LSM6DSM motion sensor for STEVAL-BCNKT01V1 and STEVAL-STLKT01V1 or for LSM6DSOX motion sensor for STEVAL-MKSBOX1V1:
   - pedometer (feature 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 (feature not available on STEVAL-MKSBOX1V1)

This software gathers:

- the temperature, humidity, pressure, audio and motion sensor drivers for the HTS221, LPS22HB, MP34DT06J, LSM6DSL and LSM303AGR devices if X-NUCLEO-IKS01A2 expansion board is mounted on the STM32 Nucleo
- the temperature, humidity, pressure, audio and motion sensor drivers for the HTS221, LPS22HH, MP34DT06J, LSM6DSO and LIS2MDL devices if X-NUCLEO-IKS01A3 expansion board is mounted on the STM32 Nucleo
- the temperature, pressure, audio, motion sensor and gas gauge IC drivers for the LPS22HB, LSM6DSM, LSM303AGR and STC3115 devices for STEVAL-STLKT01V1 running on STM32 Nucleo
- the temperature, pressure, audio and motion sensor for the LPS22HB, LSM6DSM and LSM303AGR devices for STEVAL-BCNKT01V1 running on STM32 Nucleo

- the temperature, humidity, pressure, audio and motion sensor for the HTS221, LPS22HB, MP23ABS1, LSM6DSOX and LIS2MDL devices for STEVAL-MKSBOX1V1 running on STM32 Nucleo

The package is compatible with the STBLESensor Android/iOS (Ver. 4.0.0 or higher) applications, available at the respective store, which can be used to display information sent via BLE.

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.

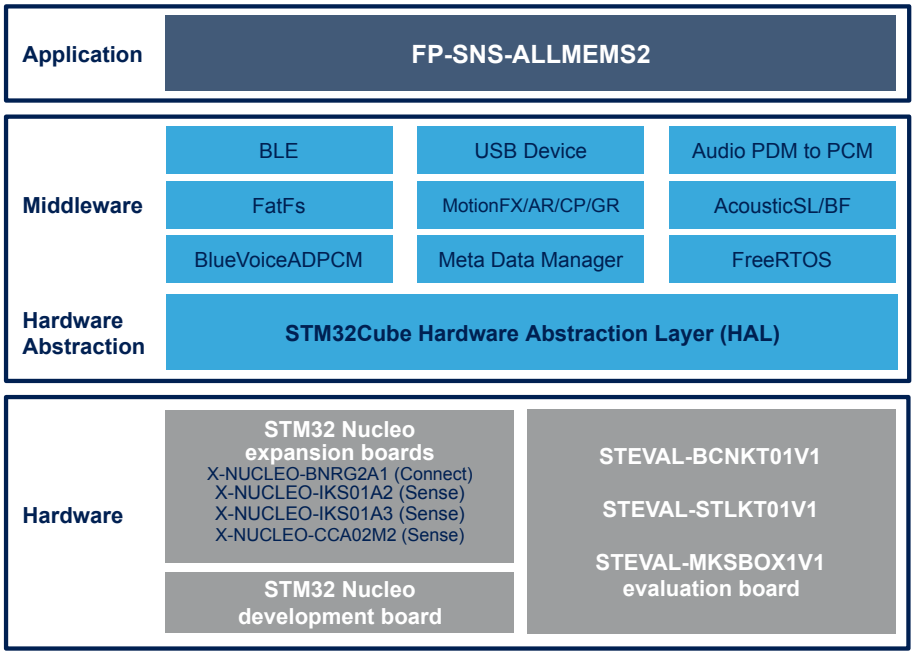
These software function packs are designed to fully exploit the underlying STM32 ODE hardware and software components to best satisfy the final user application requirements.

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:

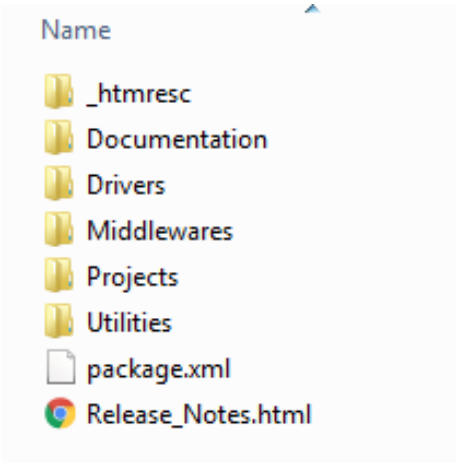
- **STM32Cube HAL layer**: provides a simple, generic and multi-instance set of generic and extension APIs (application programming interfaces) to interact with the upper layer application, libraries and stacks. It is directly based on a generic architecture and allows the layers that are built on it, 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-ALLMEMS2 software architecture



1.3 Folder structure

Figure 2. **FP-SNS-ALLMEMS2 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 activity-recognition library, MotionCP carry-position recognition library, MotionGR 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, STEVAL-MKSBOX1V1 and STEVAL-STLKT01V1 platforms through the IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM-STR) and Integrated Development Environment for STM32 (STM32CubeIDE).
- **Utilities**: contains the boot loader binary ready to be flashed for the STM32F446RE, STM32L4R9ZI and STM32L476RG Nucleo boards.

2 Flash management

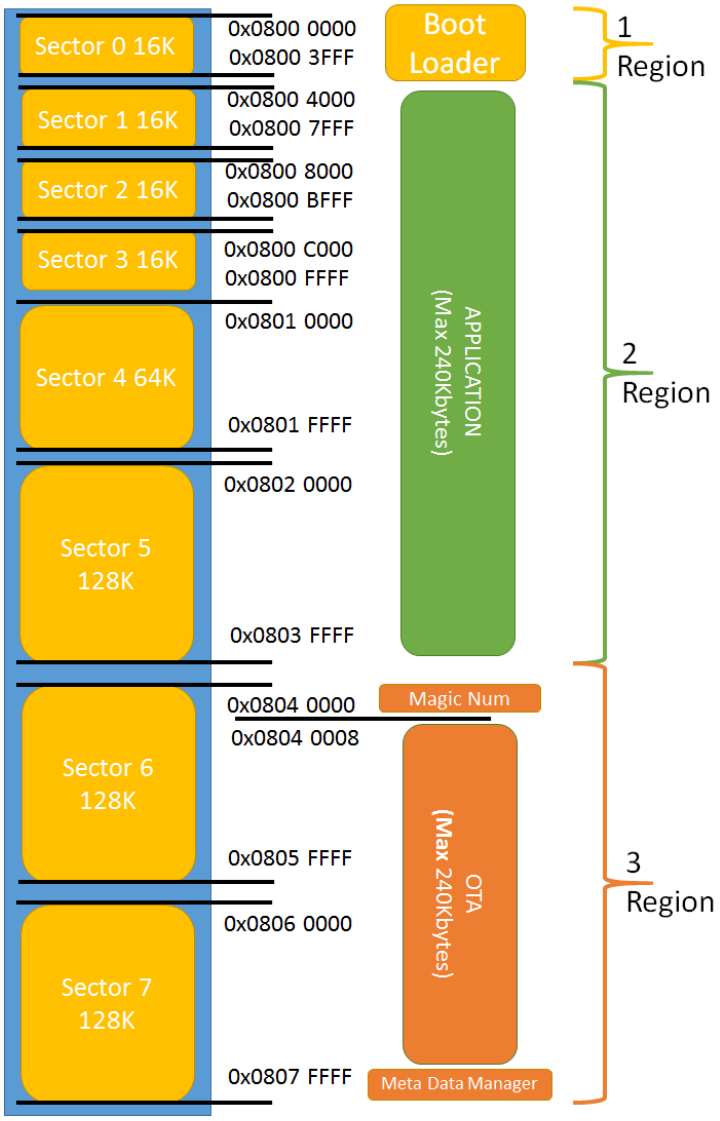
Apart from storing code, FP-SNS-ALLMEMS2 uses the Flash memory for Firmware-Over-The-Air updates. It is divided into the following regions (see figure below):

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), STM32L4R9ZI (2048KB) 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 (0x0807F000 for STM32F446RE, 0x081FF000 for STM32L4R9ZI 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
- *RM0432* Reference manual - STM32L4Rxxx and STM32L4Sxxx advanced ARM®-based 32-bit MCUs

Figure 3. FP-SNS-ALLMEMS2 FLASH structure for STM32F446RE



3 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 and STM32L4R9) thus: `#define VECT_TAB_OFFSET 0x4000`.

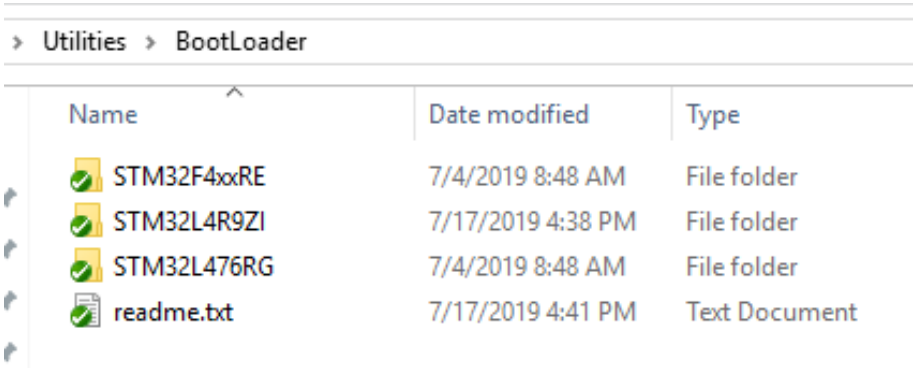
We also changed the linker script. For example, the Linker script for FP-SNS-ALLMEMS2 running on STM32F446RE and compiled using IAR Embedded Workbench for ARM is:

```
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, STM32L4R9ZI or STM32L476RG, found in the Utilities\BootLoader folder, to the first FLASH region (address 0x08000000).

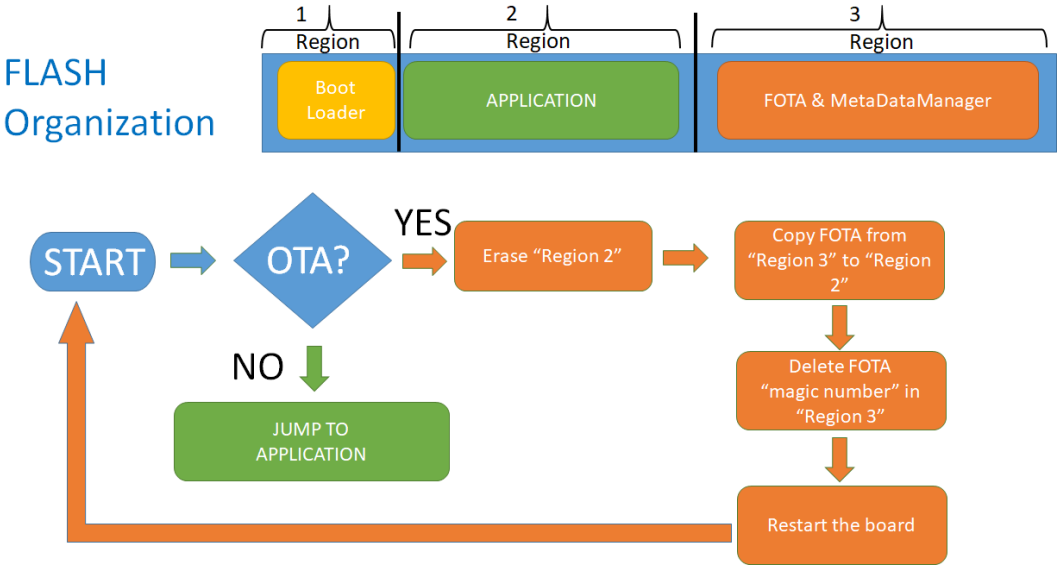
Figure 4. BootLoader folder content



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**



4 The installation process

For each platform (NUCLEO-F446RE, NUCLEO-L476RG, STEVAL-BCNKT01V1, STEVAL-STLKT01V1, STEVAL-MKSBOX1V1), the package Binary directory contains an image in .bin format, 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 evaluation 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 evaluation board with the ST-LINK or via a Drag & Drop operation

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

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, STEVAL-MKSBOX1V1 and STEVAL-STLKT01V1) and for each IDE (IAR/RealView/STM32CubeIDE):

- IAR toolchain Embedded Workbench V8.32.3:
  - For NUCLEO-F446RE:
    - CleanALLMEMS2_IAR_IKS01A2_F446.bat
    - CleanALLMEMS2_IAR_IKS01A3_F446.bat
  - For NUCLEO-L476RG:
    - CleanALLMEMS2_IAR_IKS01A2_L476.bat
    - CleanALLMEMS2_IAR_IKS01A3_L476.bat
  - For STEVAL-BCNKT01V1: CleanALLMEMS2_IAR_BC.bat
  - For STEVAL-MKSBOX1V1: CleanALLMEMS2_IAR_ST.box.bat
  - For STEVAL-STLKT01V1: CleanALLMEMS2_IAR_ST.bat
- Integrated Development Environment for STM32 (STM32CubeIDE) version 1.3.1:
  - For NUCLEO-F446RE:
    - CleanALLMEMS2_STM32CubeIDE_IKS01A2_F446.bat
    - CleanALLMEMS2_STM32CubeIDE_IKS01A3_F446.bat
  - For NUCLEO-L476RG:
    - CleanALLMEMS2_STM32CubeIDE_IKS01A2_L476.bat
    - CleanALLMEMS2_STM32CubeIDE_IKS01A3_L476.bat
  - For STEVAL-BCNKT01V1: CleanALLMEMS2_STM32CubeIDE_BC.bat
  - For STEVAL-MKSBOX1V1: CleanALLMEMS2_STM32CubeIDE_ST.box.bat
  - For STEVAL-STLKT01V1: CleanALLMEMS2_STM32CubeIDE_ST.bat
- µVision toolchain - MDK-ARM Professional Version: 5.29.0:
  - For NUCLEO-F446RE:
    - CleanALLMEMS2_MDK_ARM_IKS01A2_F446.bat
    - CleanALLMEMS2_MDK_ARM_IKS01A3_F446.bat
  - For NUCLEO-L476RG:
    - CleanALLMEMS2_MDK_ARM_IKS01A2_L476.bat
    - CleanALLMEMS2_MDK_ARM_IKS01A3_L476.bat
  - For STEVAL-BCNKT01V1: CleanALLMEMS2_MDK_ARM_BC.bat
  - For STEVAL-MKSBOX1V1: CleanALLMEMS2_MDK_ARM_ST.box.bat
  - For STEVAL-STLKT01V1: CleanALLMEMS2_MDK_ARM_ST.bat

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 6. Content of a project folder

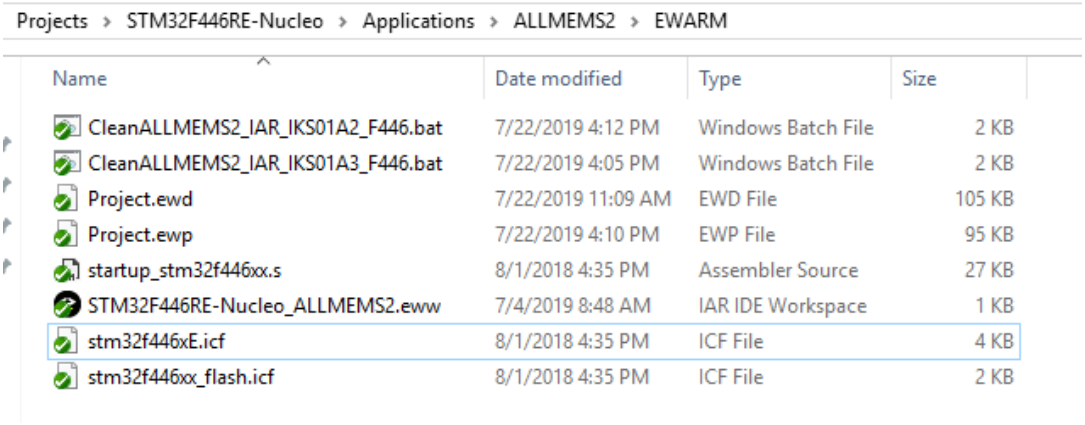
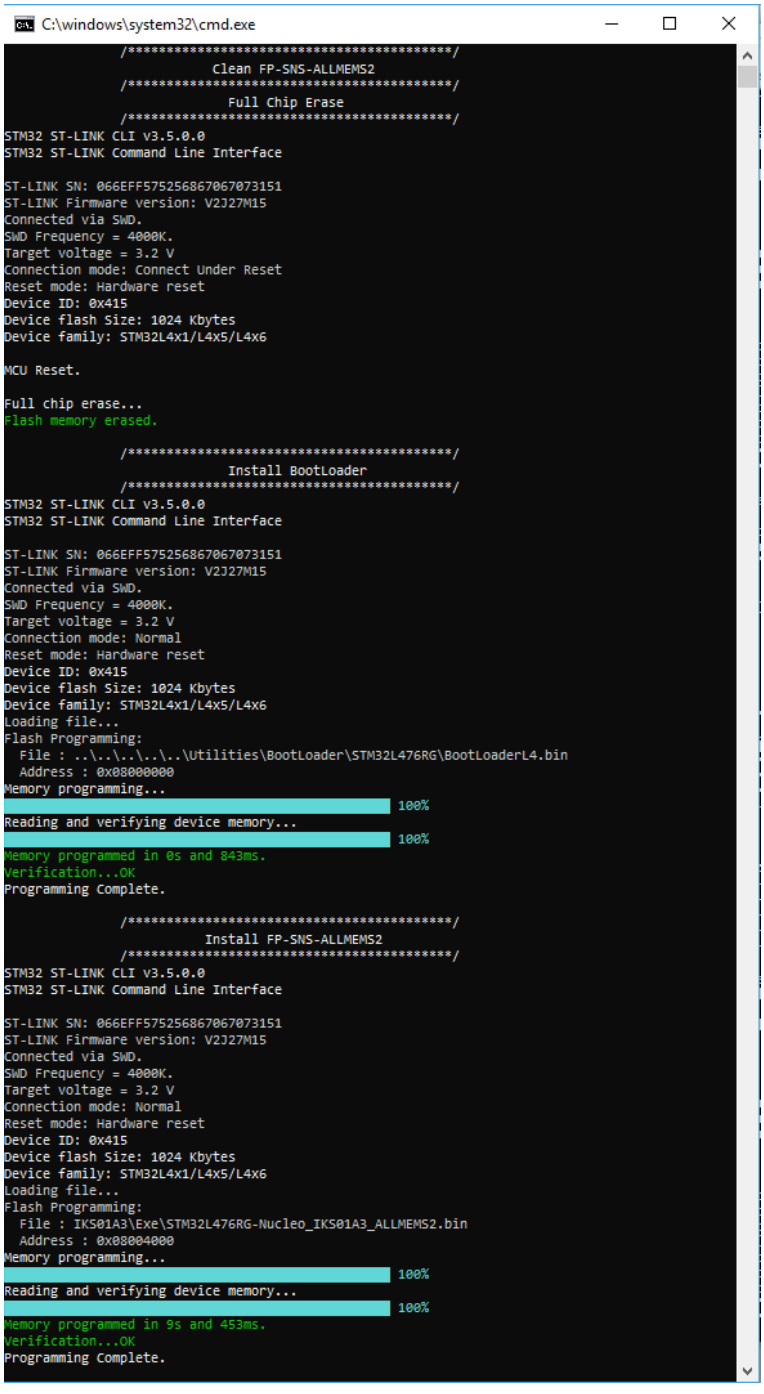
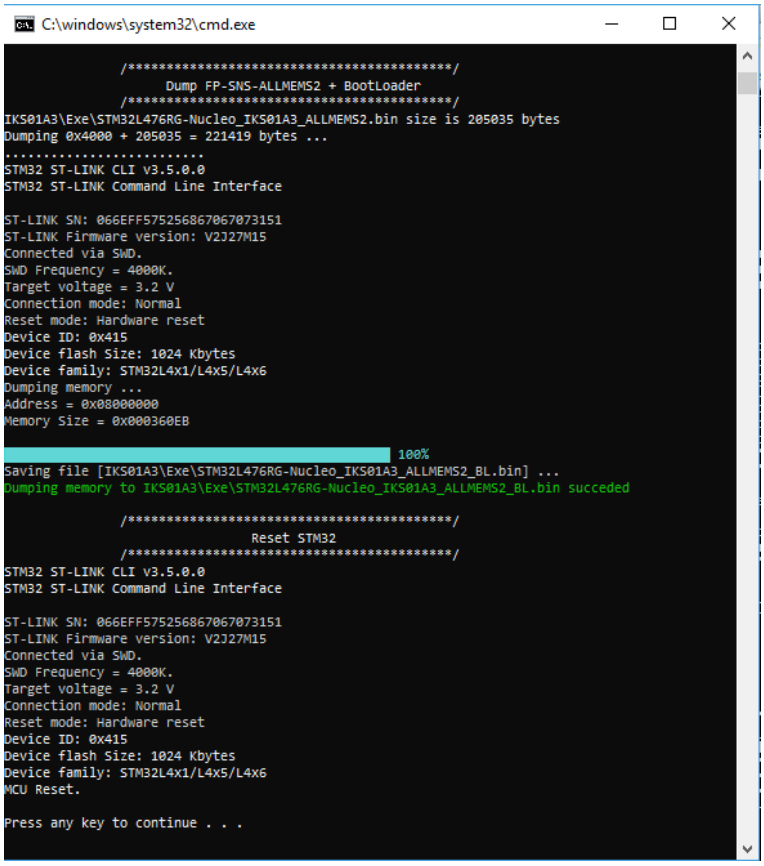


Figure 7. BootLoader and ALLMEMS2 installation



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 8. **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 (except for STEVAL-MKSBOX1V1), but is only included in the Integrated Development Environment for STM32 (STM32CubeIDE):

- CleanALLMEMS2_STM32CubeIDE_F446.sh
- CleanALLMEMS2_STM32CubeIDE_L476.sh
- CleanALLMEMS2_STM32CubeIDE_BC.sh.
- CleanALLMEMS2_STM32CubeIDE_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

The OpenOCD script section to be edited is the following one.

```
# 1) Set the Installation path for OpenOCD
# example:
#OpenOCD_DIR="C:/ST/STM32CubeIDE_1.3.0/STM32CubeIDE/plugins/com.st.stm32cube.ide.mcu.external
tools.openocd.win32_1.3.0.202002181050/tools"
OpenOCD_DIR=""

# 2) Set the installation path for stm32 OpenOCD scritps
# example:
#OpenOCD_CFC="C:/ST/STM32CubeIDE_1.3.0/STM32CubeIDE/plugins/com.st.stm32cube.ide.mcu.debug.op
enocd_1.3.0.202002181050/resources/openocd/st_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/"
```

5 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 STBLESensor application (ver. 4.0.0 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/STM32L4R9 processor to ensure integrity. If the CRC calculation matches the STBLESensor 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 Section 9 Firmware-Over-The-Air update with STBLESensor).

6 APIs

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

7 Sample application description

A sample application using:

- the X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A3), X-NUCLEO-CCA02M2 and X-NUCLEO-BNRG2A1 expansion boards with the NUCLEO-F446RE or NUCLEO-L476RG board
- the STEVAL-BCNKT01V1 evaluation kit
- the STEVAL-MKSBOX1V1 evaluation kit
- the STEVAL-STLKT01V1 evaluation kit

Ready to build projects are available for multiple IDEs.

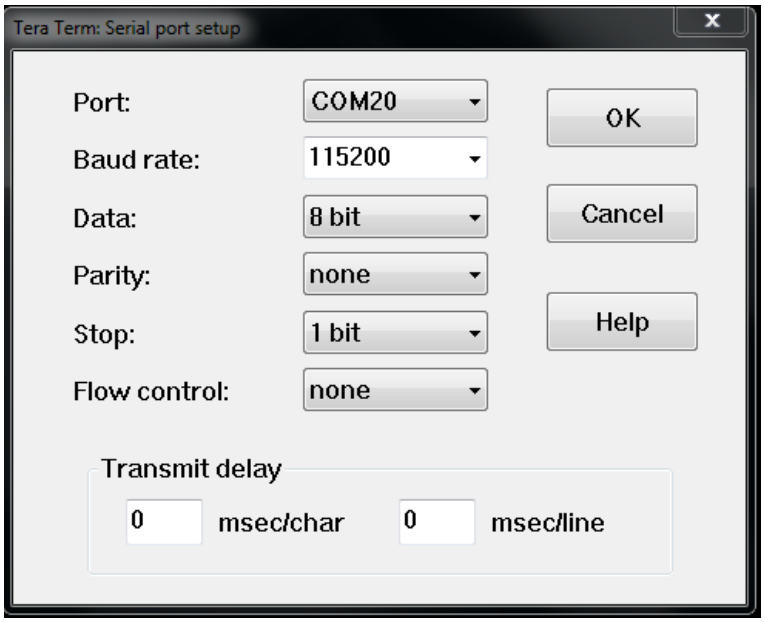
With the NUCLEO-F446RE, NUCLEO-L476RG and STEVAL-MKSBOX1V1, you can set up a terminal window for the appropriate UART communication port (as per Figure 9. Terminal setting) to control the initialization phase.

When 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 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.

Figure 9. Terminal setting



When you first press the reset button, the application:

- starts initializing the UART
- checks whether all the sensors are present and working
- reads the Meta Data Manager from flash memory
- checks if the BlueNRG expansion board is connected to the STM32 Nucleo board as well as the hardware and firmware version information
- shows board name and 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 or for LSM6DSO mounted on the X-NUCLEO-IKS01A3 (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 10. 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 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-BCNKT01V1, STEVAL-STLKT01V1 and STEVAL-MKSBOX1V1 evaluation kits.

These definitions in allmems2_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, STEVAL-MKSBOX1V1 and STEVAL-BCNKT01V1 evaluation kits, 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 NUCLEO-F446RE and STEVAL-BCNKT01V1):

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

The MotionGR 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 for the STEVAL-STLKT01V1 and STEVAL-MKSBOX1V1).

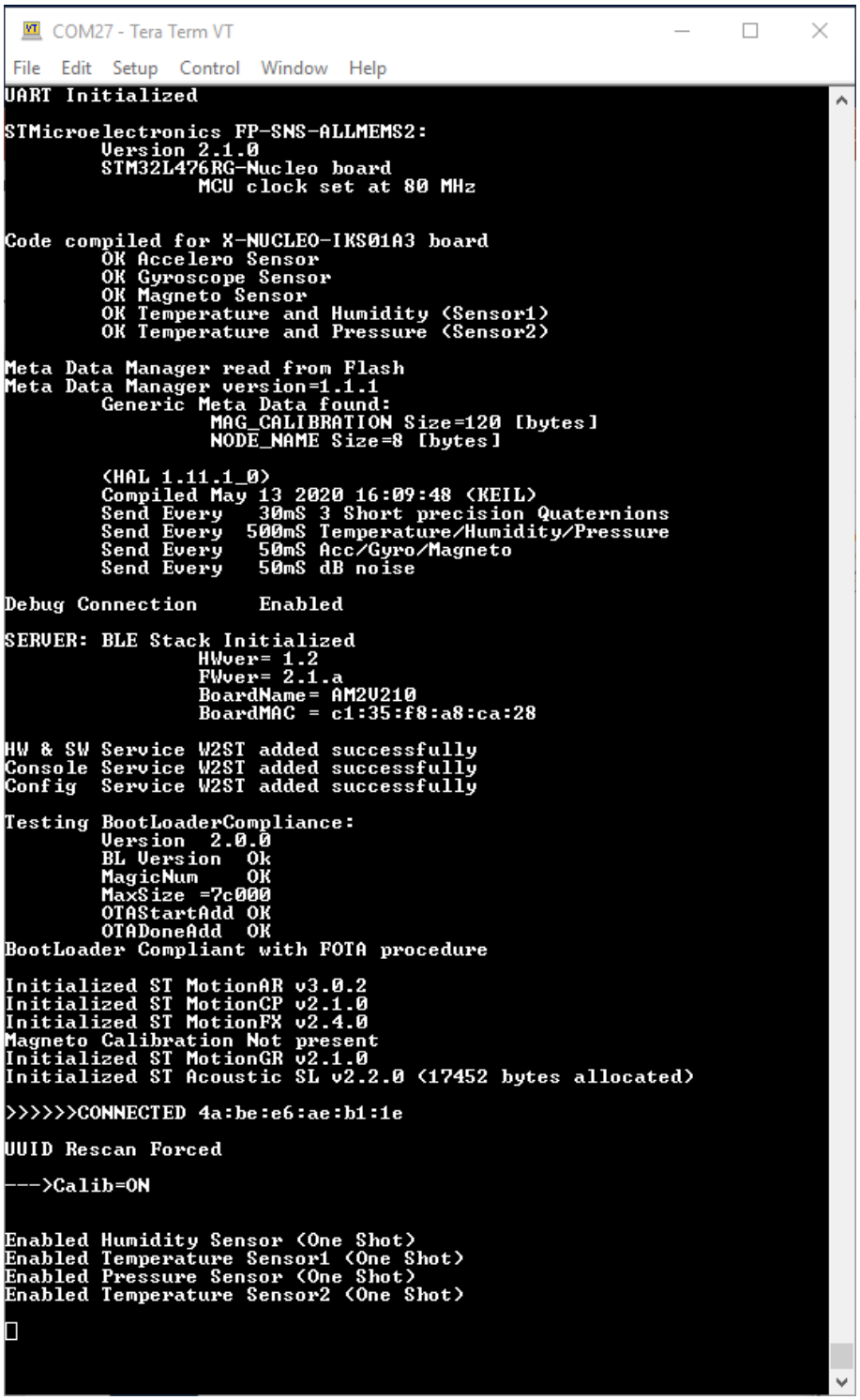
The AcousticBF library provides real-time beam forming software, using the audio signals acquired from two digital MEMS microphones, it creates a virtual directional microphone pointing to a fixed direction in space (feature available only for STEVAL-BCNKT01V1).

The BlueVoiceADPCM library implements a vendor-specific profile enabling voice communication with Bluetooth low energy (feature available for STEVAL-STLKT01V1 and STEVAL-BCNKT01V1).

The FatFs library provides access to the storage devices for sensor data logging (feature available for STEVAL-STLKT01V1 and STEVAL-MKSBOX1V1).

When an Android/iOS device is connected to the NUCLEO-F446RE, NUCLEO-L476RG, STEVAL-MKSBOX1V1 or STEVAL-STLKT01V1 (if the define `#define ALLMEMS2_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



8 Android and iOS sample client application

Based on the FP-SNS-ALLMEMS1 package and developed to reduce the power consumption, the FP-SNS-ALLMEMS2 software for STM32Cube is compatible with the STBLESensor Android/iOS applications (ver. 4.0.0 or higher) available at the respective Play/iOS stores.

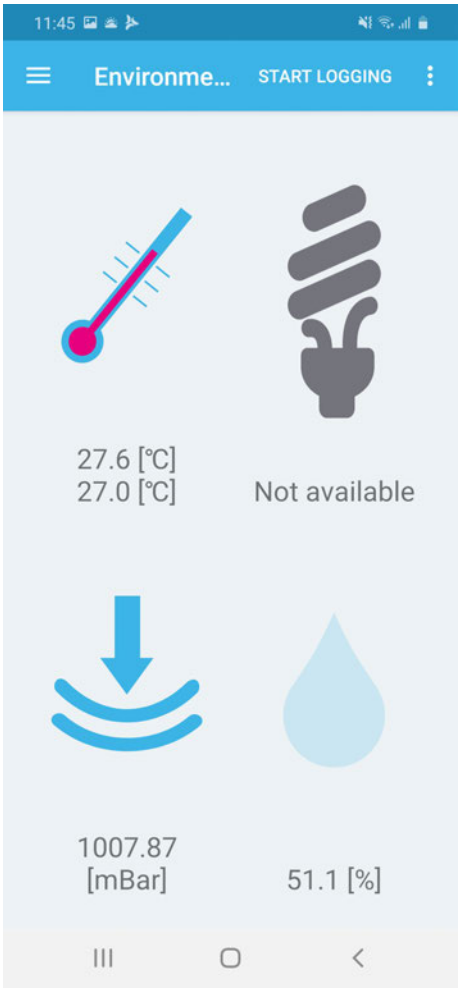
The STBLESensor application allows FOTA updates.

We will use the Android version for this demonstration.

8.1 Main page

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

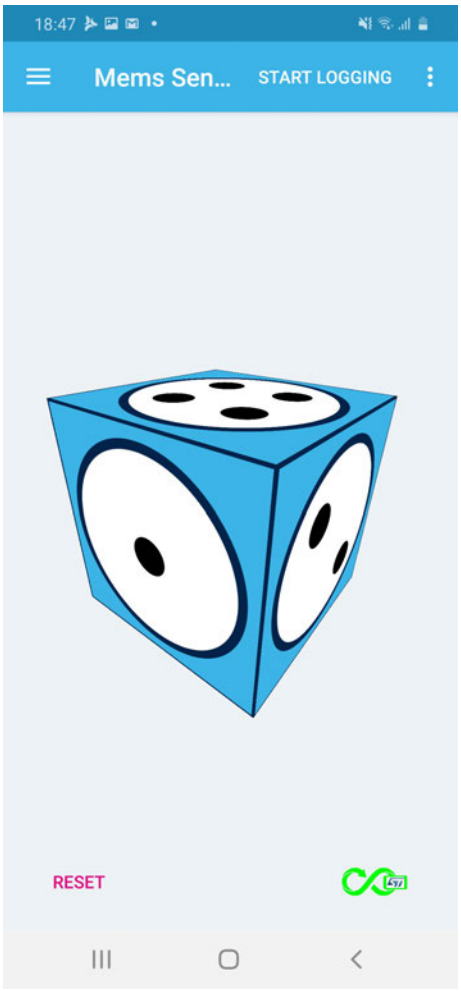
Figure 12. STBLESensor (Android version) main page following BLE connection



8.2 MEMS sensor fusion

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

Figure 13. STBLESensor (Android version) MotionFX sensor fusion page

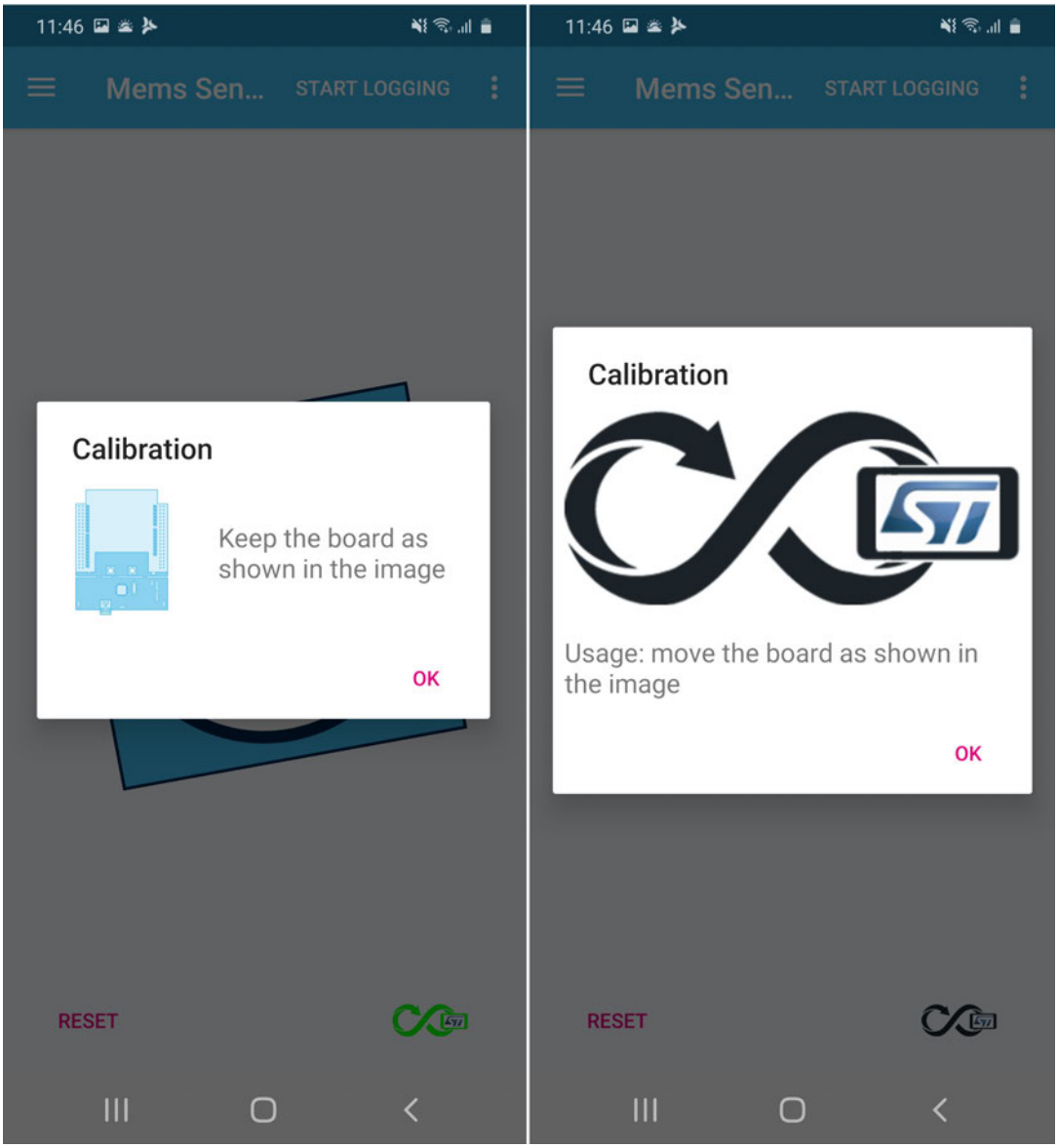


On this page, there are two buttons along the bottom:

- 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).

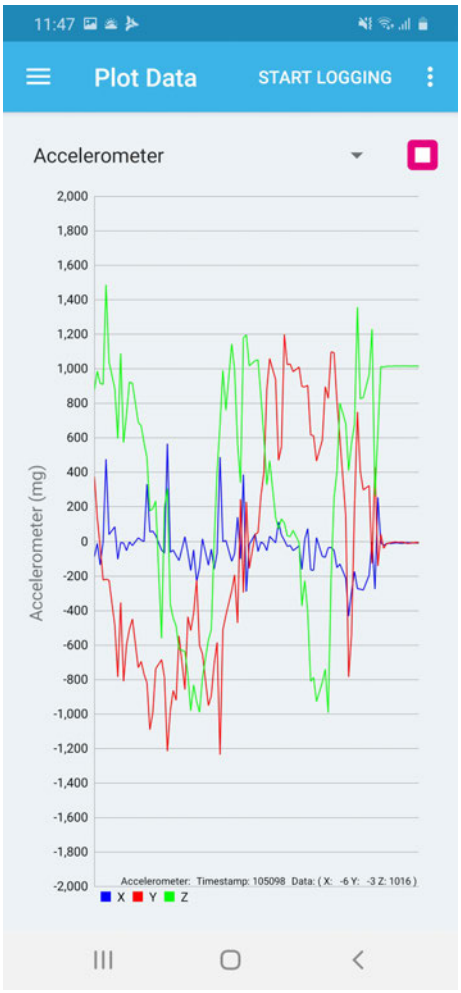
Figure 14. **STBLESensor (Android version) popup windows**



8.3 Plot data

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

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

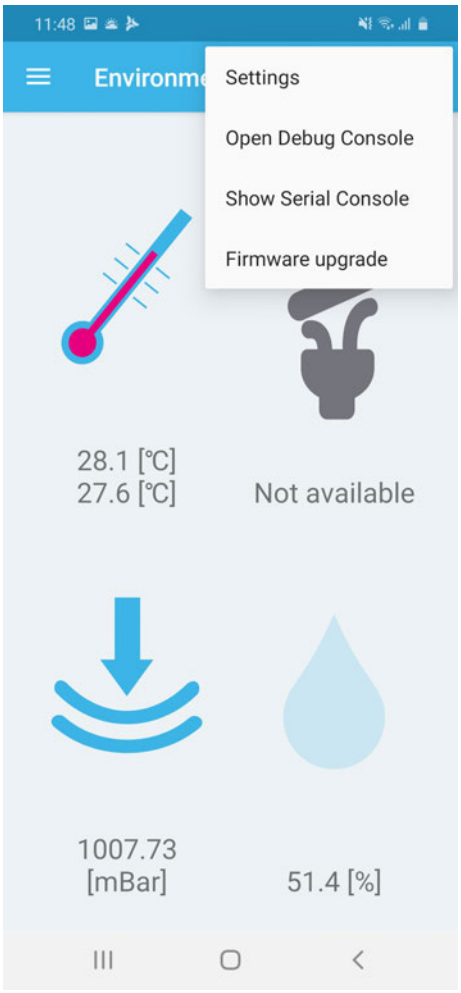


8.4 Settings, serial and debug console

In the option menu below, you can open:

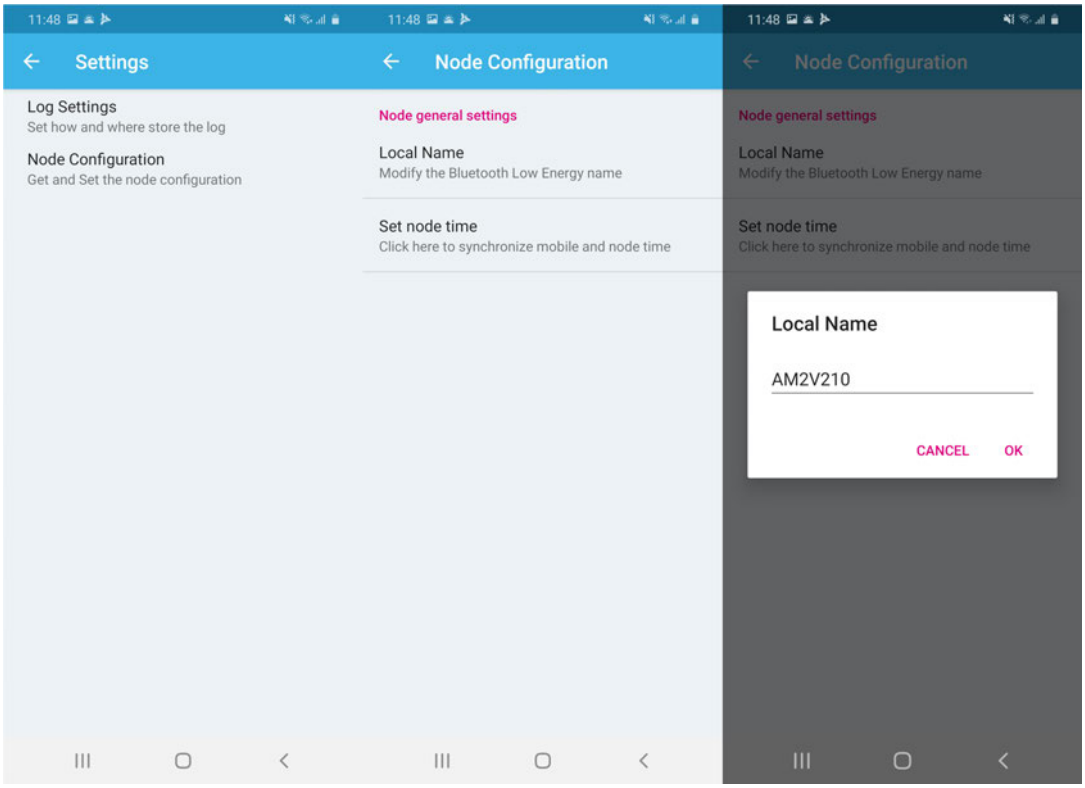
- Settings
- Serial or Debug (with stdin) console
- Firmware upgrade

Figure 16. STBLESensor (Android version) menu selection



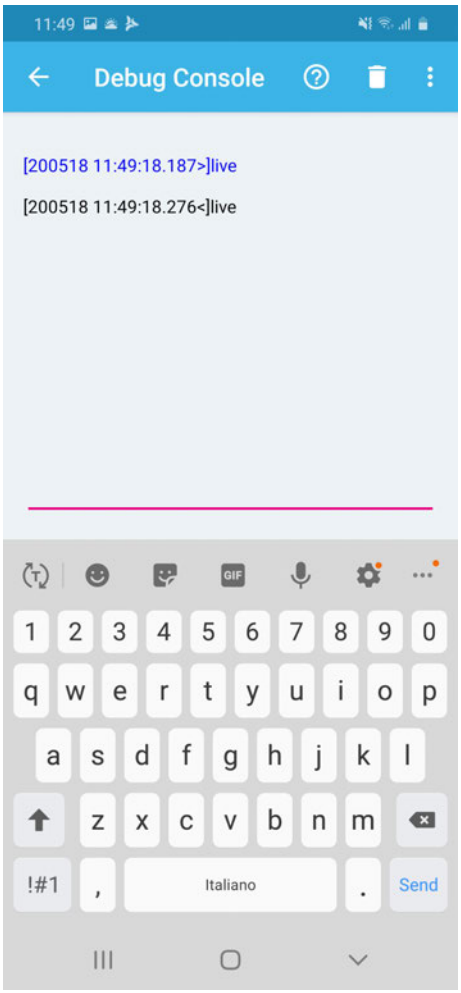
If Settings is selected, it is possible to change the node name using the node configuration, as shown below:

Figure 17. **STBLESensor (Android version) Settings, Node Configuration, Local Name**



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

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)

8.4.1 SD card data logging

SD data logging is available (for the STEVAL-STLKT01V1 (SensorTile) and STEVAL-MKSBOX1V1 (SensorTile.box) 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 STBLESensor 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.

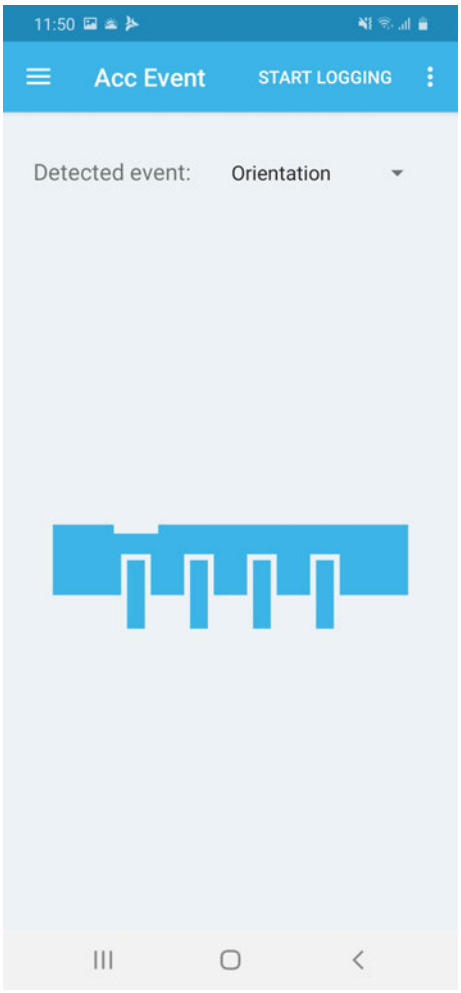
8.5 Enable hardware features

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
- LSM6DSO on X-NUCLEO-IKS01A3 expansion board for NUCLEO-L476RG board only
- LSM6DSM for STEVAL-BCNKT01V1 and STEVAL-STLKT01V1 kits
- LSM6DSOX for STEVAL-MKSBOX1V1 kit

The orientation hardware feature is the default setting.

Figure 19. STBLESensor (Android version) orientation hardware feature



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

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

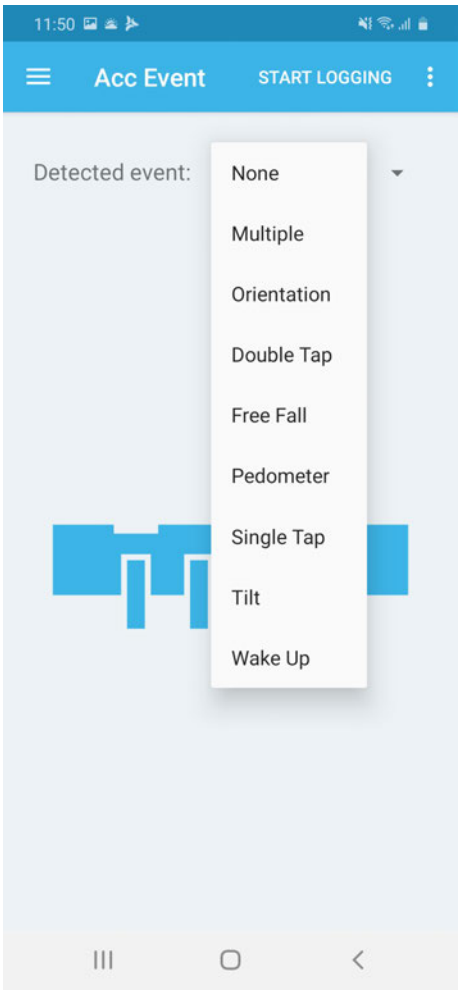
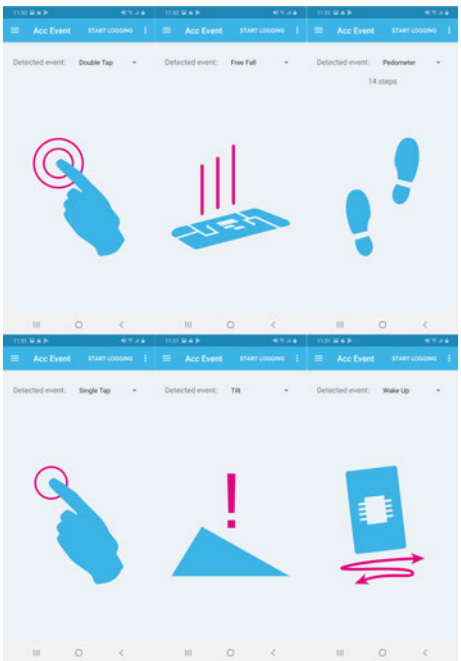


Figure 21. **STBLESensor (Android version) hardware feature examples**



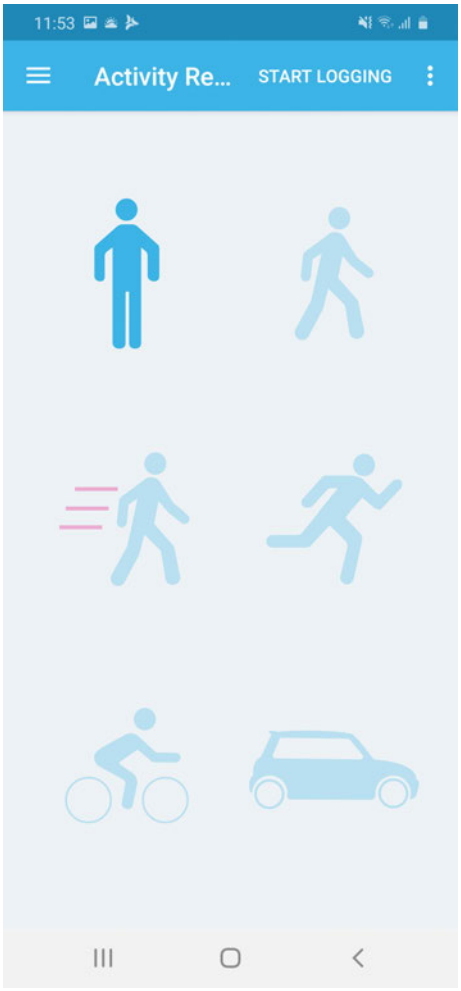
8.6 Activity recognition

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 22. STBLESensor (Android version) MotionAR activity recognition page



8.7 Carry position

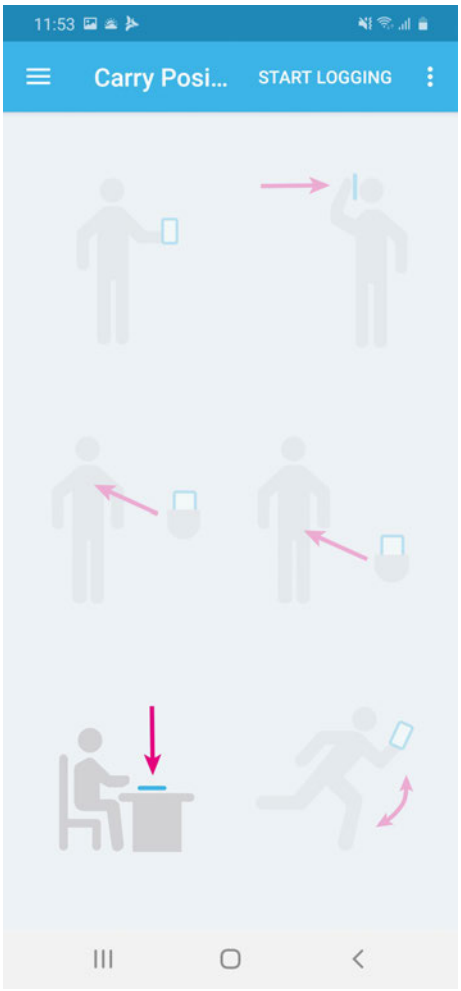
If the MotionCP algorithm is enabled, the page shown below is available, with information about how the user is carrying the board, which equates to phone carry positions (feature not available on NUCLEO-F446RE and STEVAL-BCNKT01V1):

- 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 23. STBLESensor (Android version) MotionCP carry position recognition page



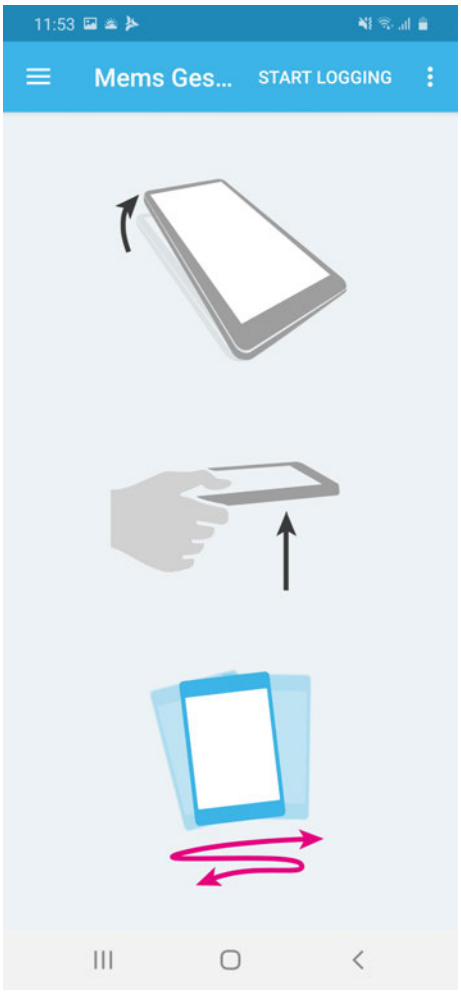
8.8 Gesture recognition

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 24. **STBLESensor (Android version) MotionGR gesture recognition page**



8.9 E-compass

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

Figure 25. **STBLESensor (Android version) MotionFX page**

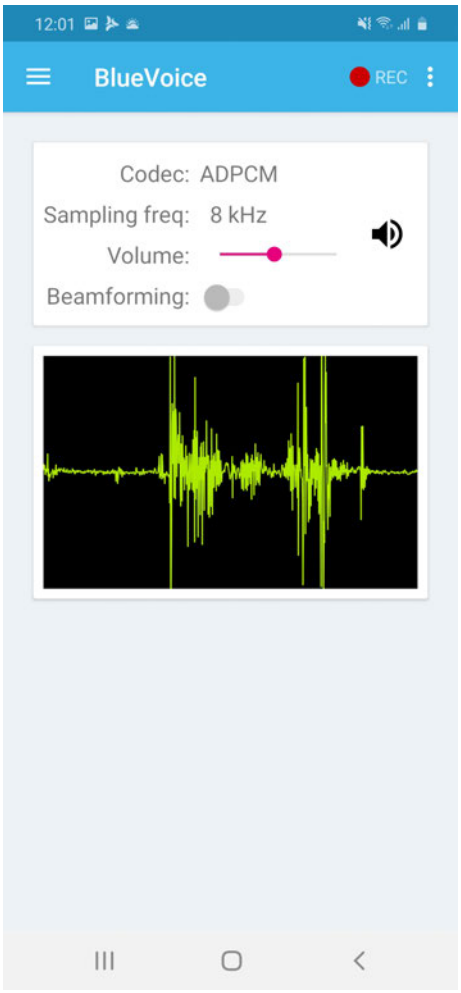
On the page bottom, the right button shows the MotionFX library calibration status (black for not calibrated, green for calibrated). Clicking it forces a magneto calibration.

8.10 BlueVoiceADPCM voice

If the BlueVoiceADPCM voice over BLE library is enabled (feature available on STEVAL-BCNKT01V1 and STEVAL-STLKT01V1 evaluation boards), 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 26. **STBLESensor (Android version) BlueVoice page**

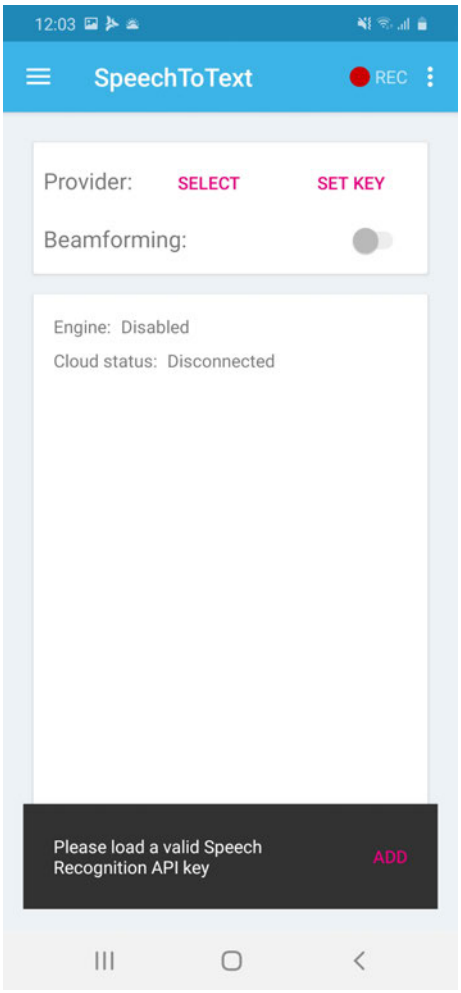


The audio playback begins as soon as the streaming from the peripheral node starts. The volume can be adjusted using the slider or muted by clicking on the speaker icon.

8.10.1 Speech to text

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

Figure 27. **STBLESensor (Android version) Speech to text page**



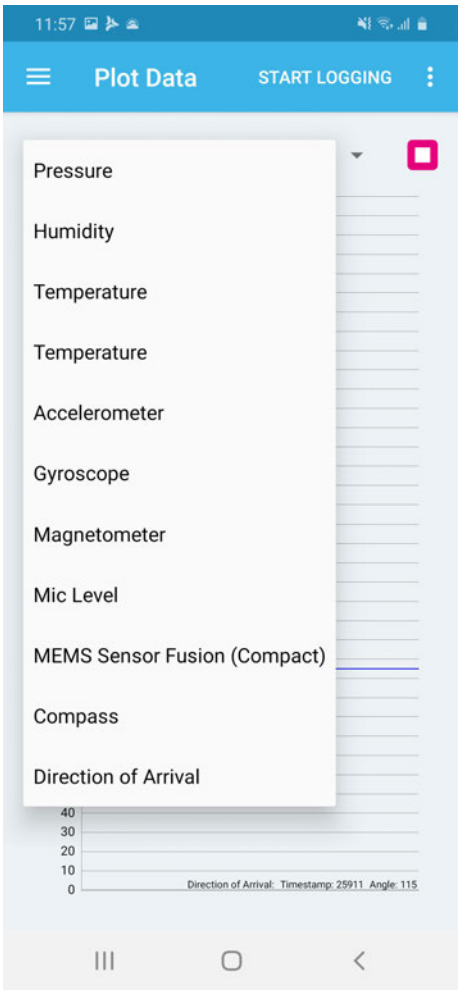
RELATED LINKS

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

8.11 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 and STEVAL-MKSBOX1V1).

Figure 28. **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 29. 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 30. **STBLESensor (Android version) audio source localization BlueCoin Page**

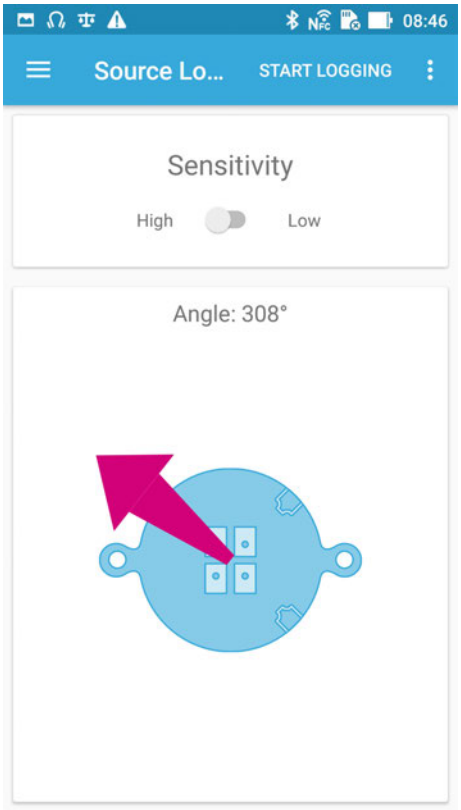
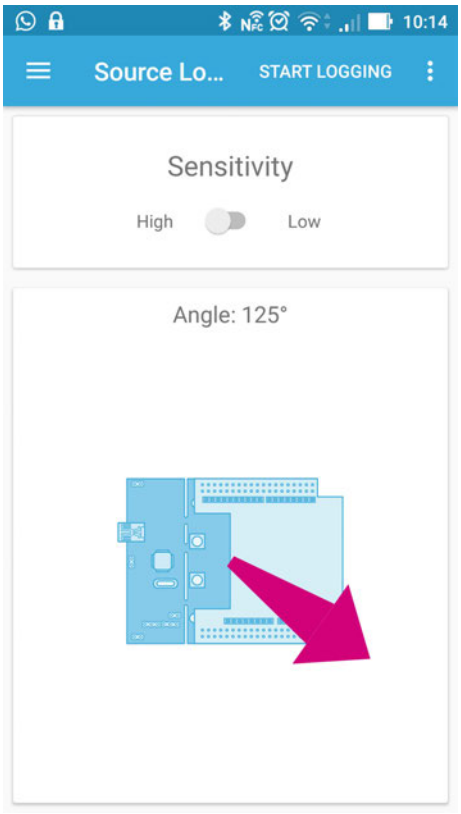


Figure 31. **STBLESensor (Android version) audio source localization STM32 Nucleo Page**

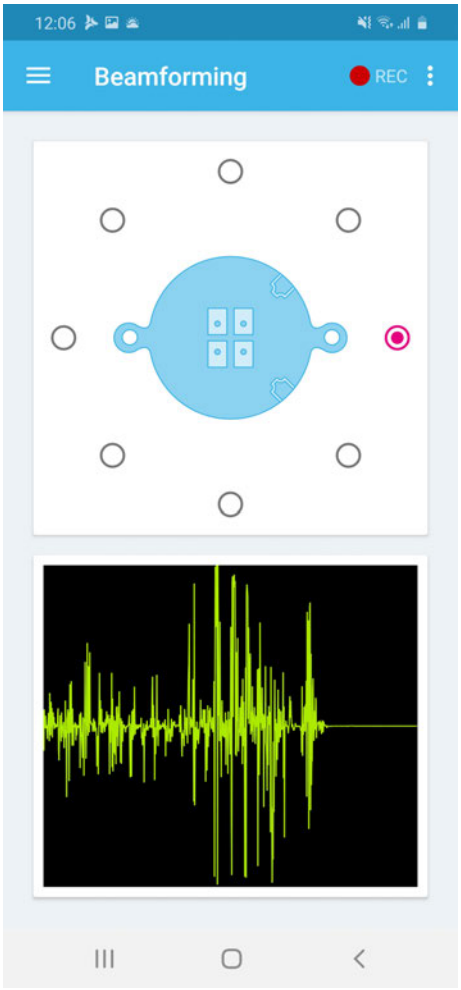


In a noisy environment, use low sensitivity.

8.12 Beam Forming

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

Figure 32. STBLESensor (Android version) audio beam forming BlueCoin Page



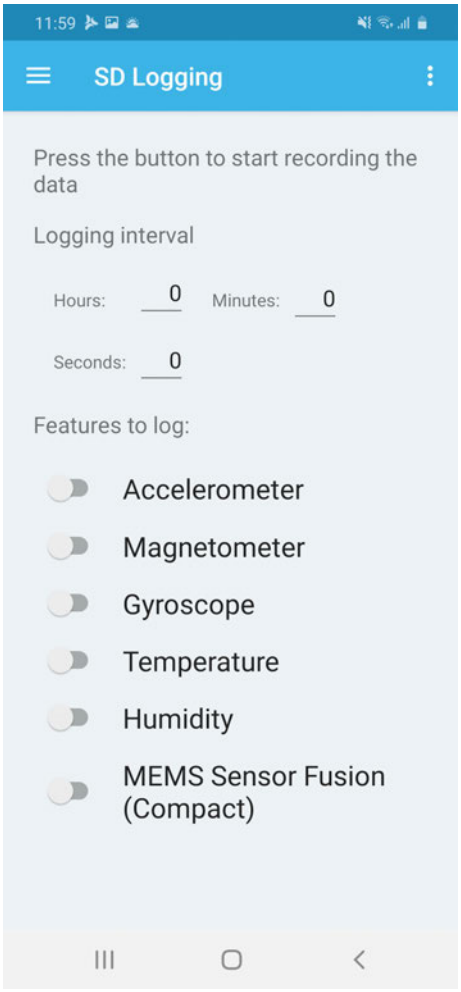
The AcousticBF provides real-time beam forming software, using the audio signals acquired from two digital MEMS microphones, it creates a virtual directional microphone pointing to a fixed direction in space.

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

8.13 SD Logging

This page shows SD Logging setting for STEVAL-STLKT01V1 and STEVAL-MKSBOX1V1 only.

Figure 33. **STBLESensor (Android version) SD Logging information**

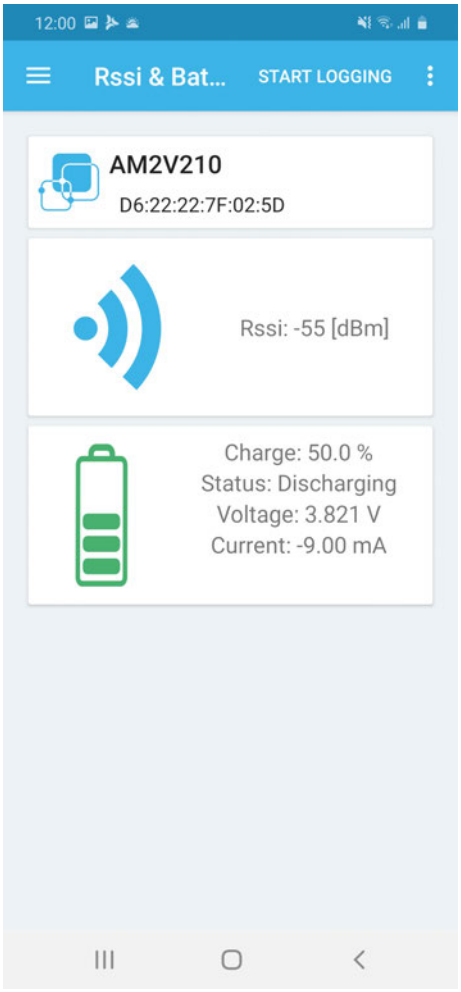


When the data logging starts, the other STBLESensor app functions are disabled and the data logging goes on even if the app is closed.

8.14 Rssi and battery

This page shows RSSI of the Bluetooth signal strength and, for STEVAL-STLKT01V1, STEVAL-MKSBOX1V1 and STEVAL-BCNKT01V1, if the battery is connected, the charge percentage, measured voltage and battery status (charging/discharging/low battery).

Figure 34. **STBLESensor (Android version) Battery and RSSI information**

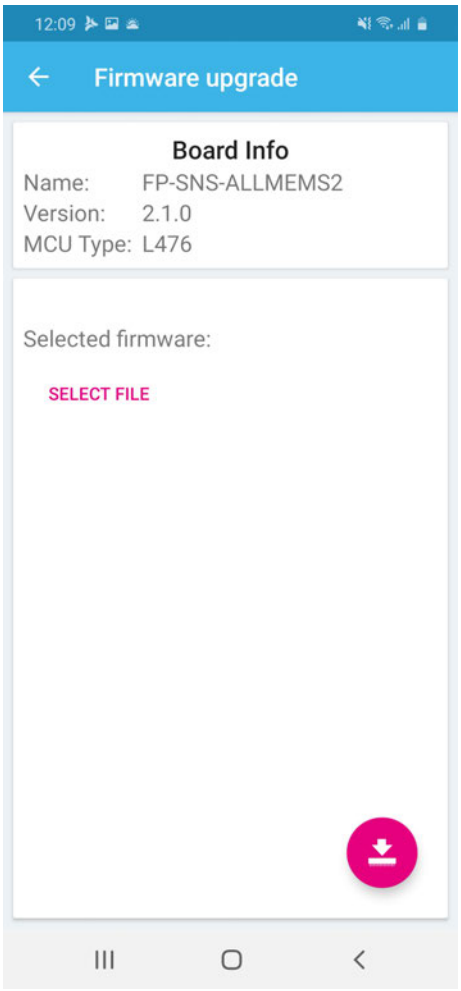


The RSSI value is updated every 0.5 seconds.
For STEVAL-BCNKT01V1 and STEVAL-MKSBOX1V1 the current value is not available.

9 Firmware-Over-The-Air update with STBLESensor

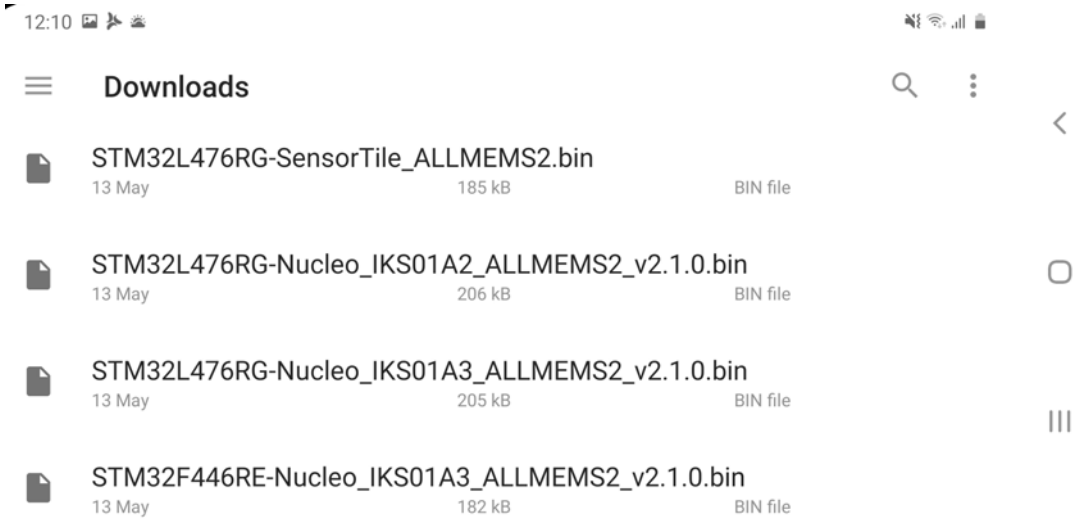
If the 'Firmware upgrade' option menu is selected, the following page appears.

Figure 35. STBLESensor (Android version) firmware upgrade page



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

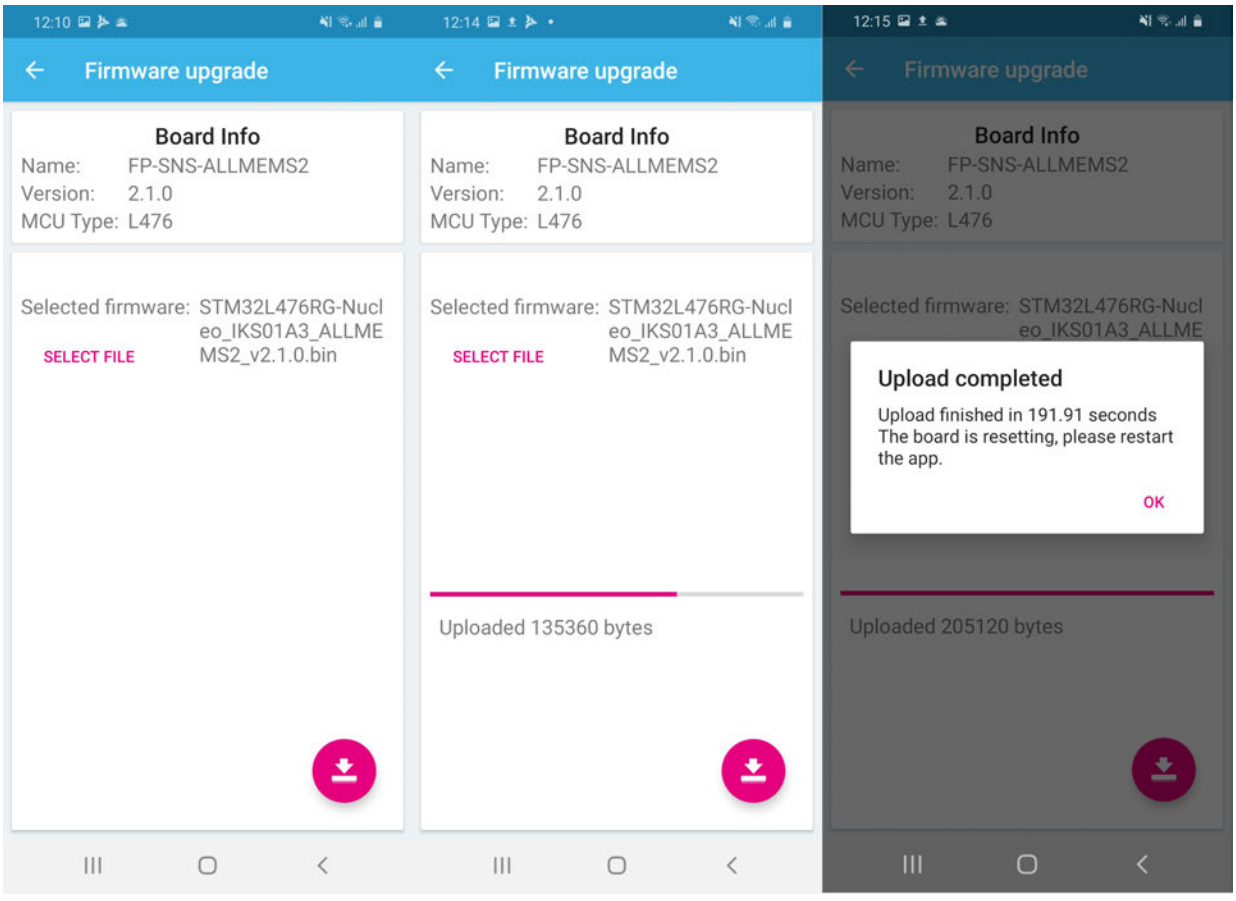


STBLESensor sends the FP-SNS-ALLMEMS2 a command communicating that it is going to send an update of a certain byte size and corresponding CRC value.

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.

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

Figure 37. **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-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 38. Terminal window feedback during FOTA

10 System setup guide

10.1 Hardware description

10.1.1 STM32 Nucleo

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

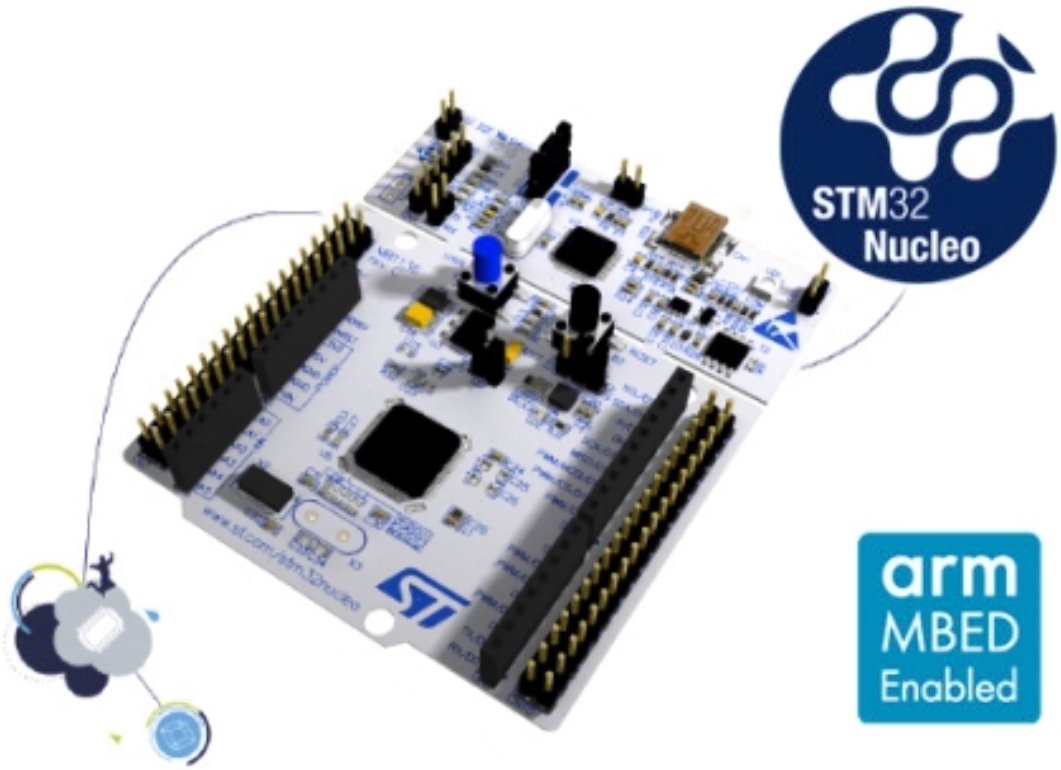
The Arduino™ connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

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

The STM32 Nucleo board comes with the comprehensive STM32 software HAL library together with various packaged software examples for different IDEs (IAR EWARM, Keil MDK-ARM, STM32CubeIDE, mbed and GCC/ LLVM).

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

Figure 39. STM32 Nucleo board



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

10.1.2 X-NUCLEO-CCA02M2 expansion board

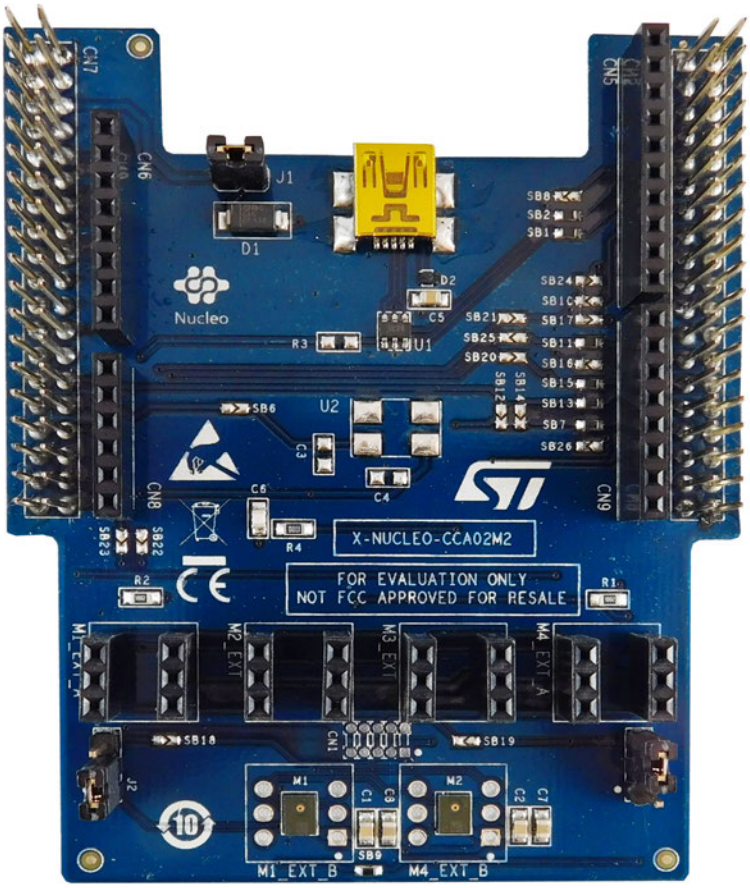
The X-NUCLEO-CCA02M2 expansion board has been designed around MP34DT06J digital MEMS microphone.

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 40. X-NUCLEO-CCA02M2 expansion board



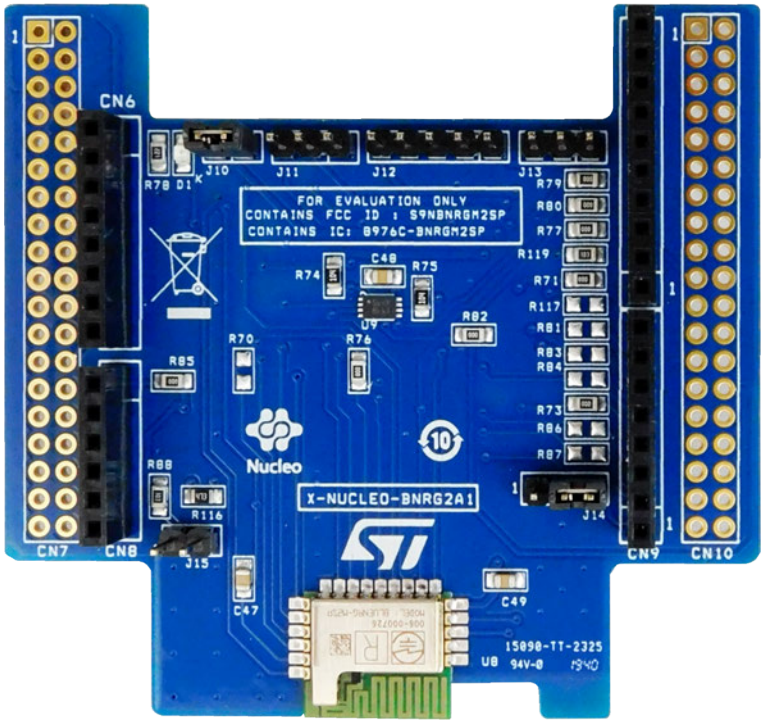
10.1.3 X-NUCLEO-BNRG2A1 expansion board

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

The expansion board features the Bluetooth® v5.0 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, 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 by the hardware.

Figure 41. **X-NUCLEO-BNRG2A1 BLE expansion board**



Information about the X-NUCLEO-BNRG2A1 expansion board is available at http://www.st.com/x-nucleo.

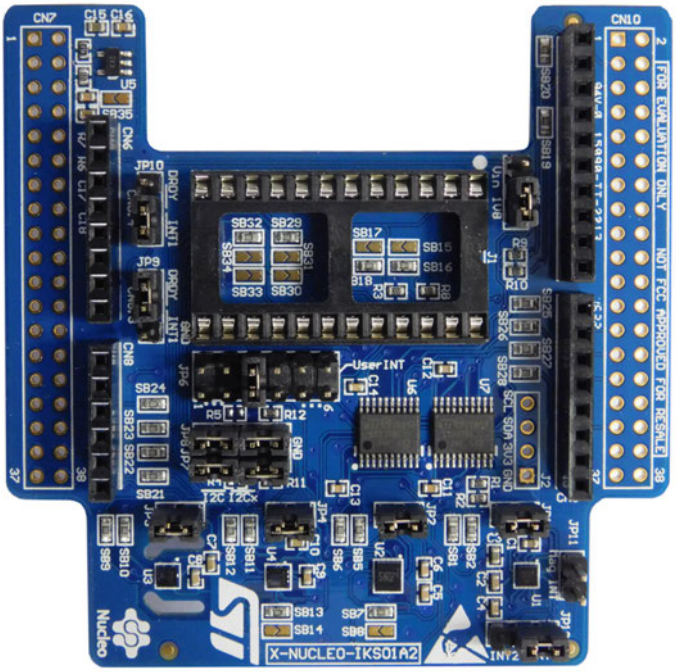
10.1.4 X-NUCLEO-IKS01A2 expansion board

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



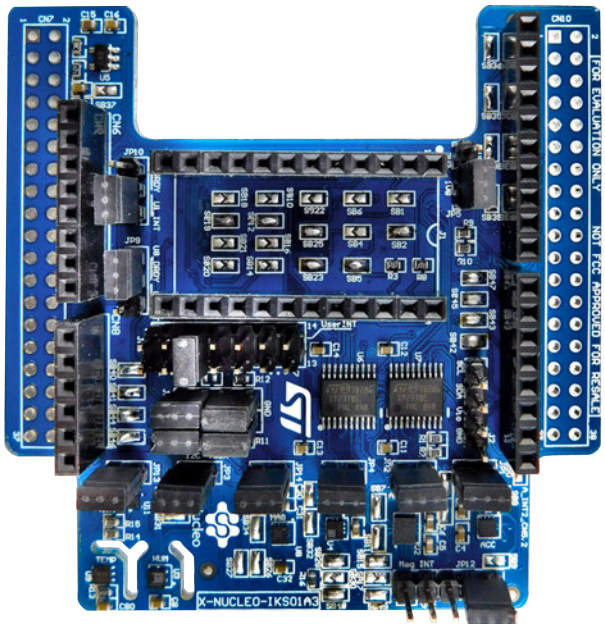
10.1.5 X-NUCLEO-IKS01A3 expansion board

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 43. **X-NUCLEO-IKS01A3 MEMS and environmental sensor expansion board**



10.1.6 STEVAL-BCNKT01V1 BlueCoin development kit

10.1.6.1 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.

10.1.6.2 Features

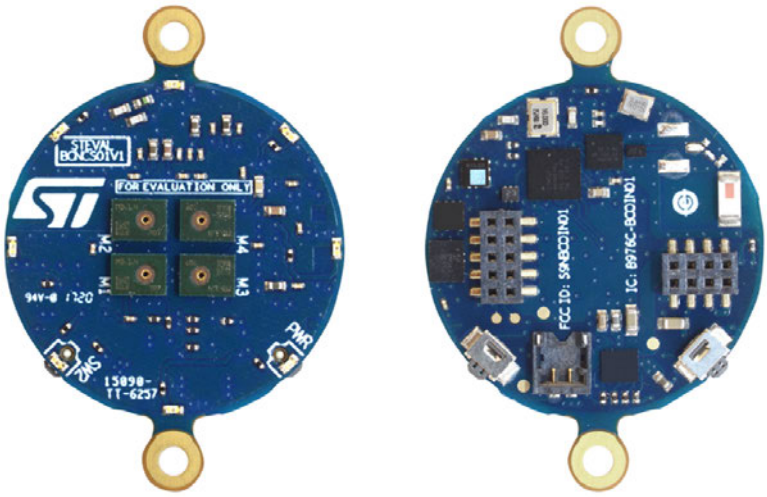
- Contains FCC ID: S9NBCOIN01
- Contains module IC 8976C-BCOIN01 certified with PMN: STEVAL-BCNKT01V1; HVIN: STEVAL-BCNCS01V1; HMN: STEVAL-BCNCR01V1; FVIN: bluenrg_7_2_c_Mode_2-32MHz-XO32K_4M.img
- The development kit package includes:
  - BlueCoin module (STEVAL-BCNCS01V1) with STM32F446, LSM6DSM, LSM303AGR, LPS22HB, 4x MP34DT06J, 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

10.1.6.3 *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 MP34DT06JTR – 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.

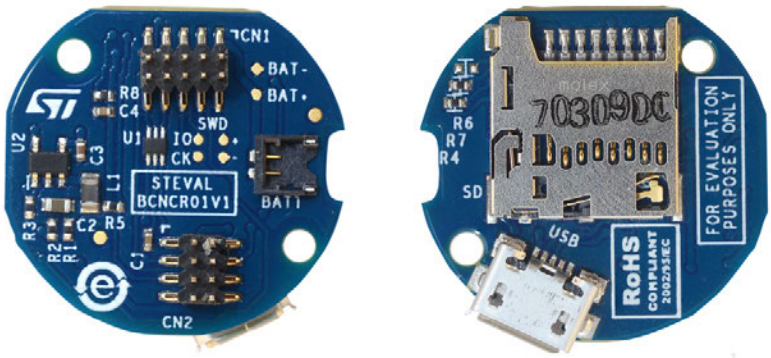
Figure 44. STEVAL-BCNCS01V1 - BlueCoin Core System



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

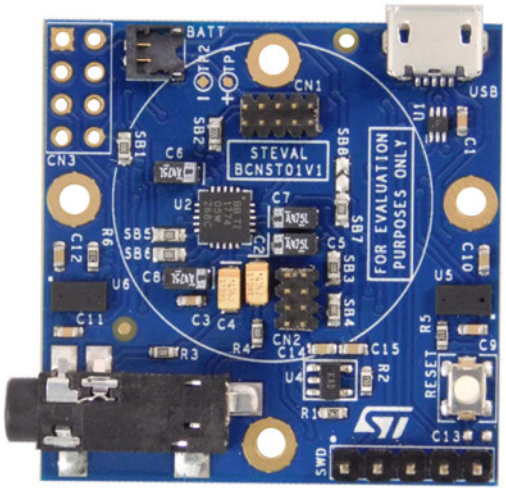
Figure 45. **STEVAL-BCNCR01V1 - BlueCoin Cradle board**



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 46. **STEVAL-BCNST01V1 - CoinStation board**



10.1.7 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 47. STEVAL-MKSBOX1V1 evaluation kit

10.1.8 STEVAL-STLKT01V1 SensorTile development kit

10.1.8.1 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.

10.1.8.2 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

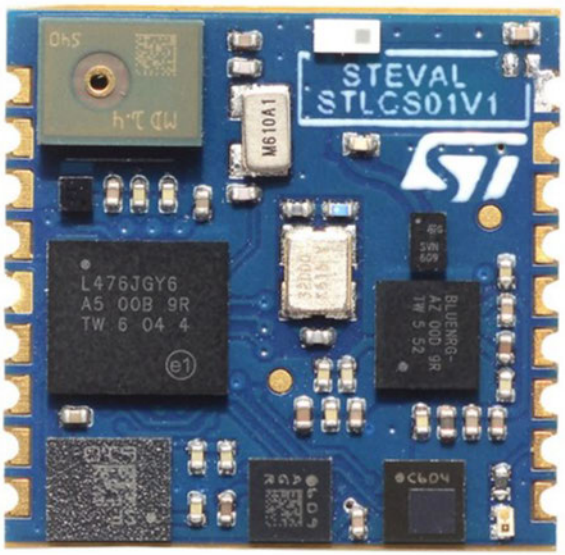
- Software libraries and tools
  - STSW-STLKT01: SensorTile firmware package that supports sensors raw data streaming via USB, data logging on SDCard, audio acquisition and audio streaming.
  - FP-SNS-ALLMEMS1 and FP-SNS-ALLMEMS2: STM32Cube functional packs
  - BlueMS: iOS and Android demo Apps
  - BlueST-SDK: 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)

10.1.8.3 *Boards included in the kit*

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

Figure 48. **STLCS01V1 board**



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.

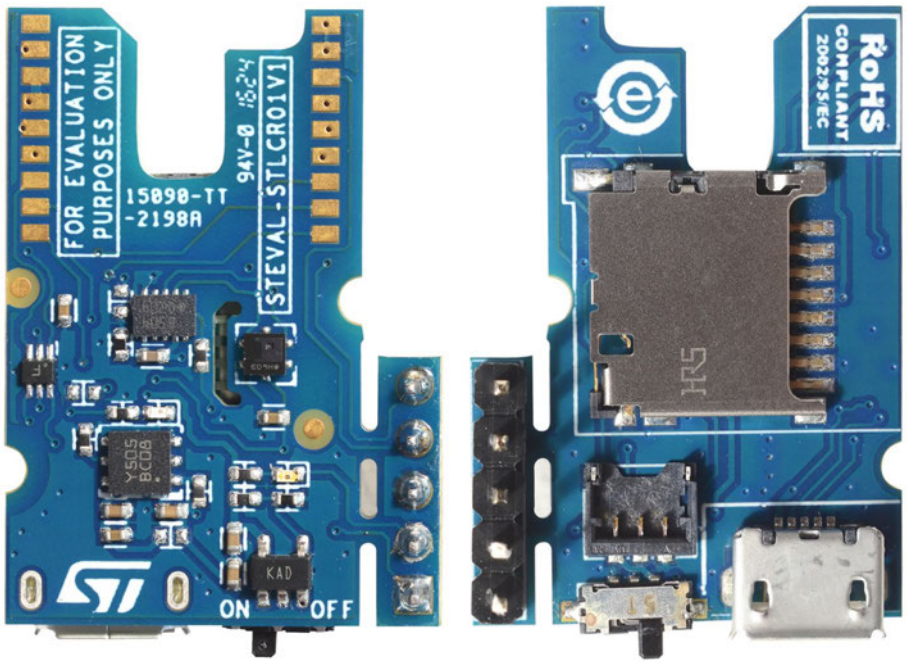
All the firmware packages are freely available on www.st.com.

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.

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

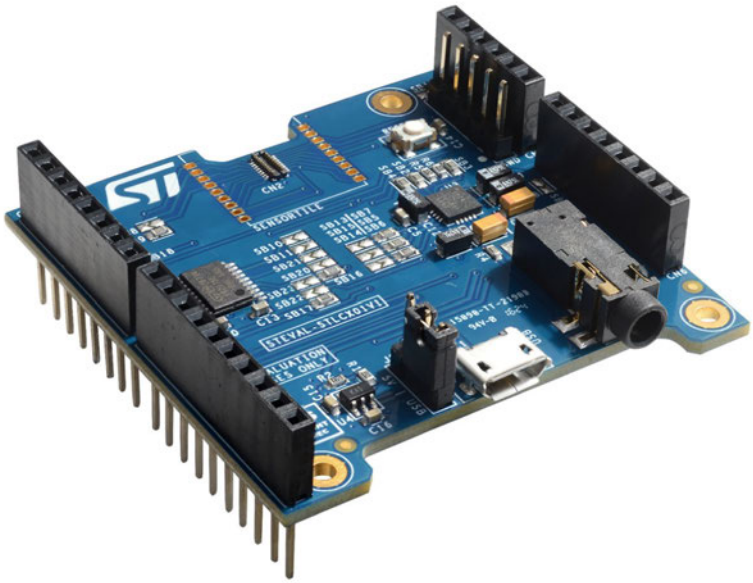
Figure 49. **STLCR01V1 board**



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

Figure 50. **STLCX01V1 board photo**



10.2 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-IKS01A2 or X-NUCLEO-IKS01A3)
  - 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-STLKT01V1:
  - STEVAL-STLKT01V1 development kit
  - ST-LINK/V2-1 debugger/programmer integrated in the STM32 Nucleo board
  - One USB type A to Mini-B USB cable to connect theSTM32 Nucleo to the PC
  - One USB type A to Micro-B USB cable to connect the STEVAL-STLKT01V1 to the PC
- for STEVAL-MKSBOX1V1:
  - STEVAL-MKSBOX1V1 development kit
  - ST-LINK/V2-1 in-circuit debugger/programmer for STM32
  - One USB type A to Mini-B USB cable to connect the ST-LINK/V2-1 to the PC
  - One USB type A to Micro-B USB cable to connect the STEVAL-MKSBOX1V1 to the PC
- for STEVAL-BCNKT01V1:
  - STEVAL-BCNKT01V1 development kit
  - ST-LINK/V2-1 debugger/programmer integrated in the 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

10.3 Software setup

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, STEVAL-MKSBOX1V1 and STEVAL-BCNKT01V1:

- 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-STR) toolchain + ST-LINK
  - Integrated Development Environment for STM32 (STM32CubeIDE) + ST-LINK

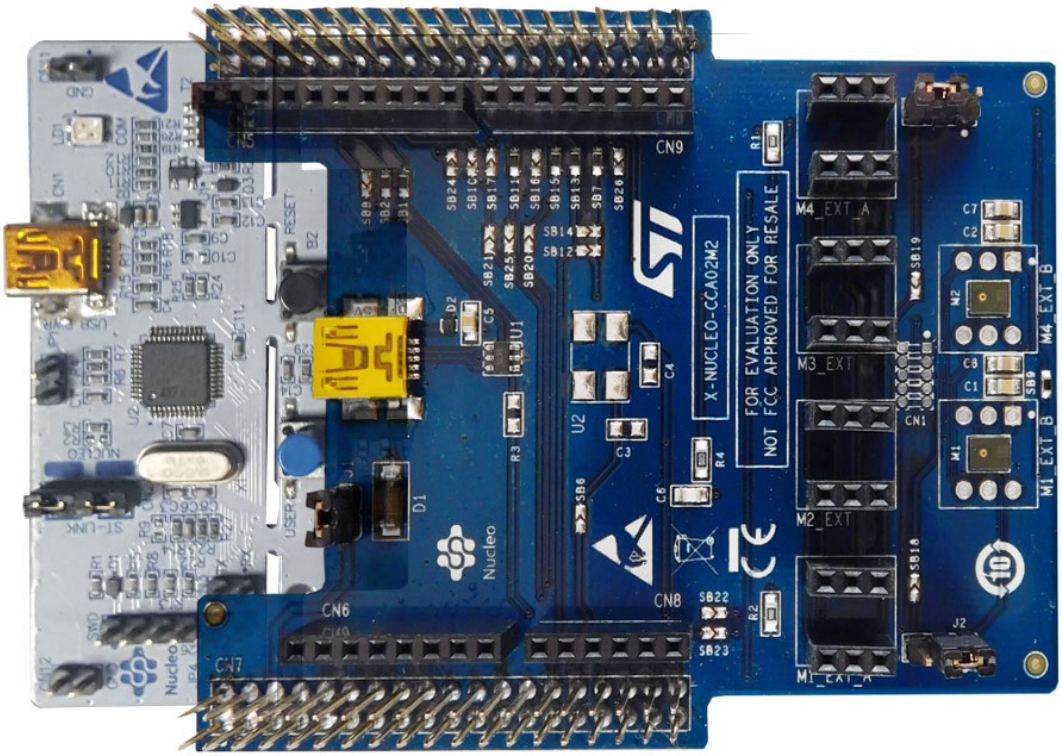
10.4 System setup

10.4.1 STM32 Nucleo and expansion board setup

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

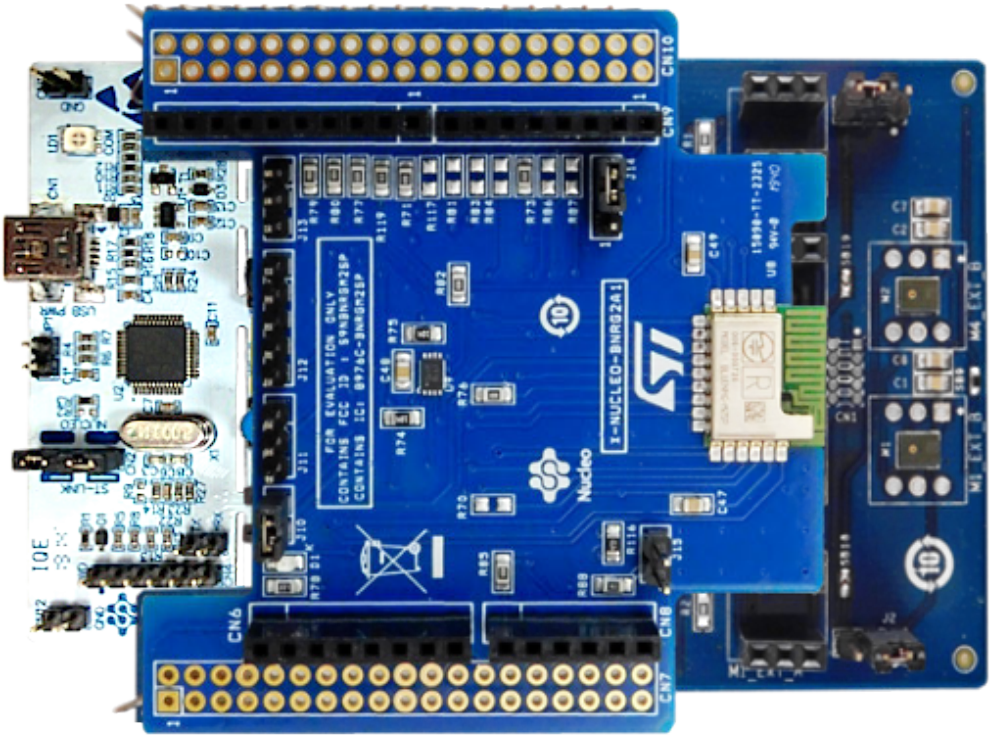
The X-NUCLEO-CCA02M2 sensor board is easily connected to the STM32 Nucleo development board through the morpho connectors, as shown below.

Figure 51. STM32 Nucleo plus X-NUCLEO-CCA02M2 boards



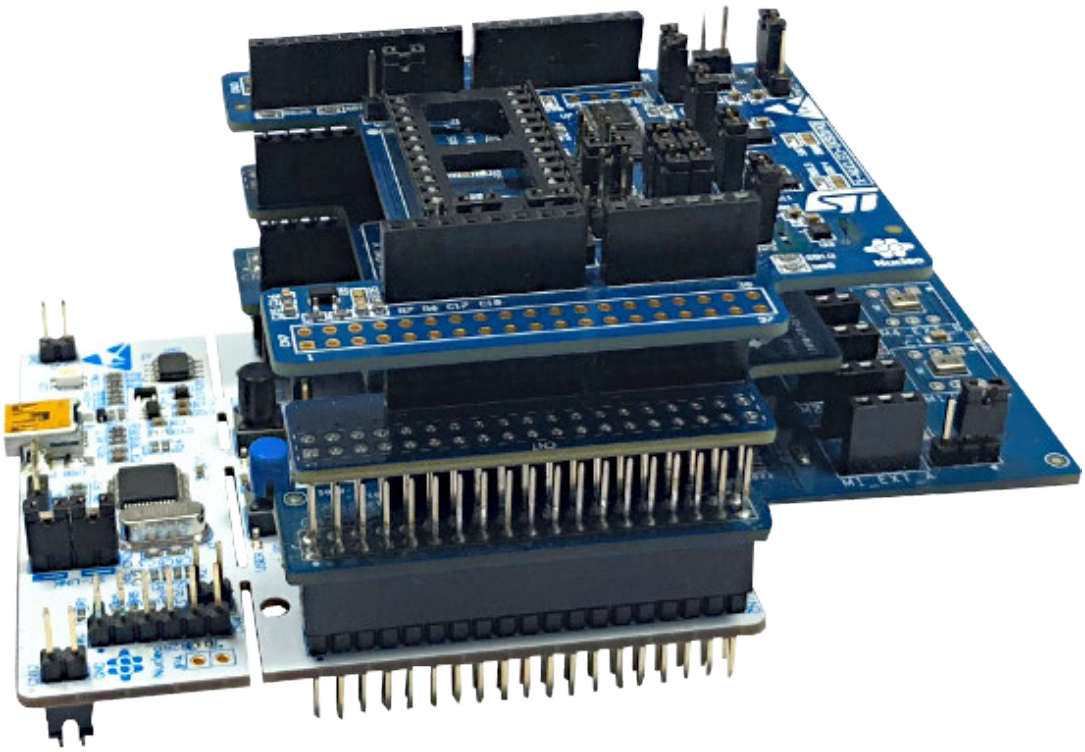
The X-NUCLEO-BNRG2A1 expansion board is connected to the X-NUCLEO-CCA02M2 board through the Arduino UNO R3 extension connector, as shown below.

Figure 52. **STM32 Nucleo plus X-NUCLEO-CCA02M2 plus X-NUCLEO-BNRG2A1 boards**



The X-NUCLEO-IKS01A2 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 53. **STM32 Nucleo plus X-NUCLEO-CCA02M2 plus X-NUCLEO-BNRG2A1 plus X-NUCLEO-IKS01A2 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.

10.4.2 STEVAL-STLKT01V1 setup

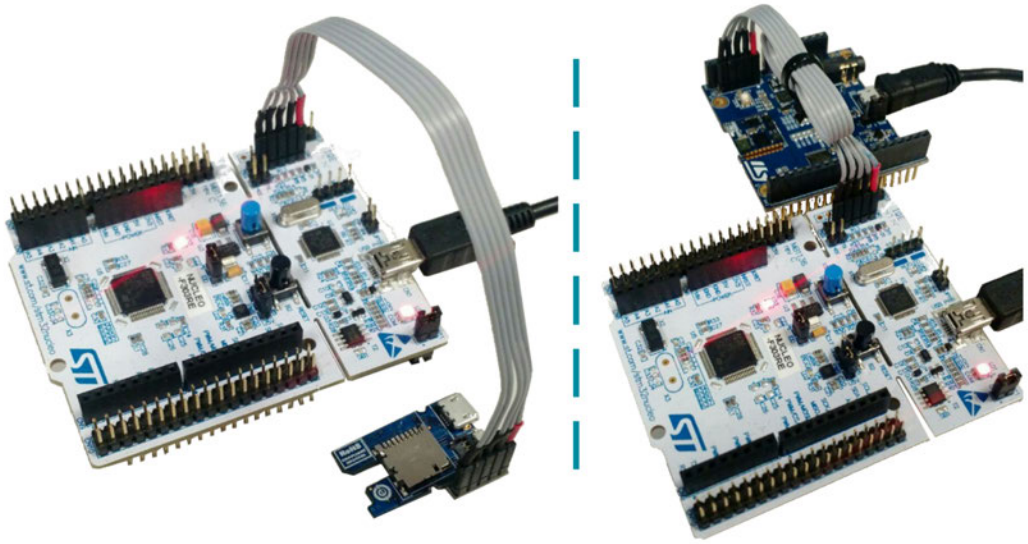
The ST-LINK/V2-1 debugger/programmer integrated in the STM32 Nucleo board must be used to program the STEVAL-STLCS01V1 (SensorTile).

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



10.4.3 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

10.4.4 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 55. **BlueCoin-STM32 Nucleo: SWD connections with 5-pin flat cable**

10.4.5 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 BLE module.

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

Step 2. Update the X-NUCLEO-BNRG2A1 BLE 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 56. Connection between X-NUCLEO-BNRG2A1 expansion board and ST-LINK/V2-1

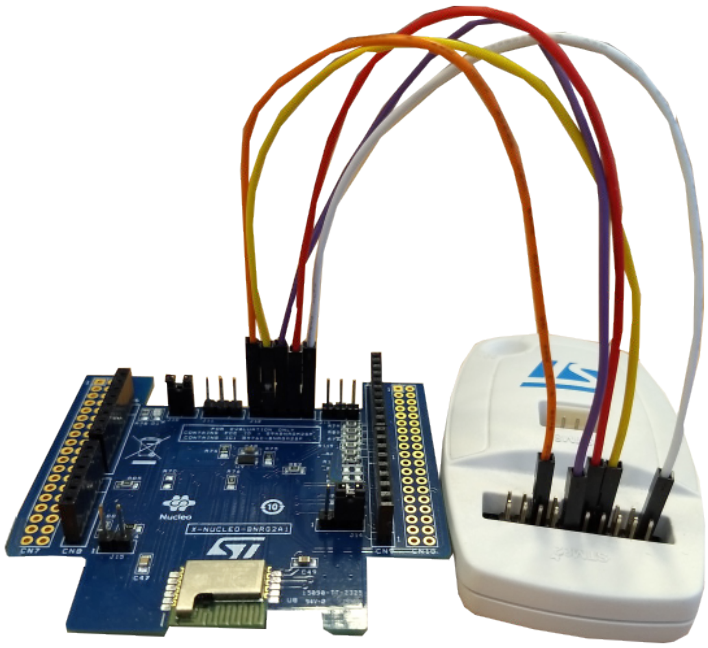


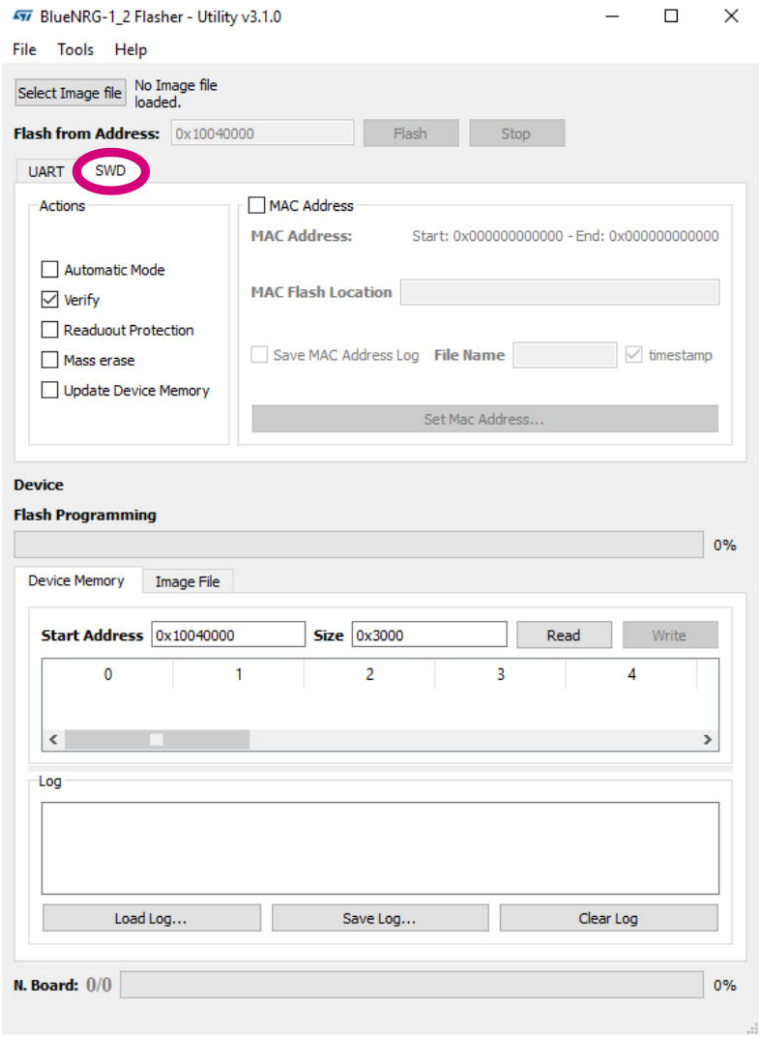
Table 1. X-NUCLEO-BNRG2A1 and ST-LINK/V2-1 pin connections

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.

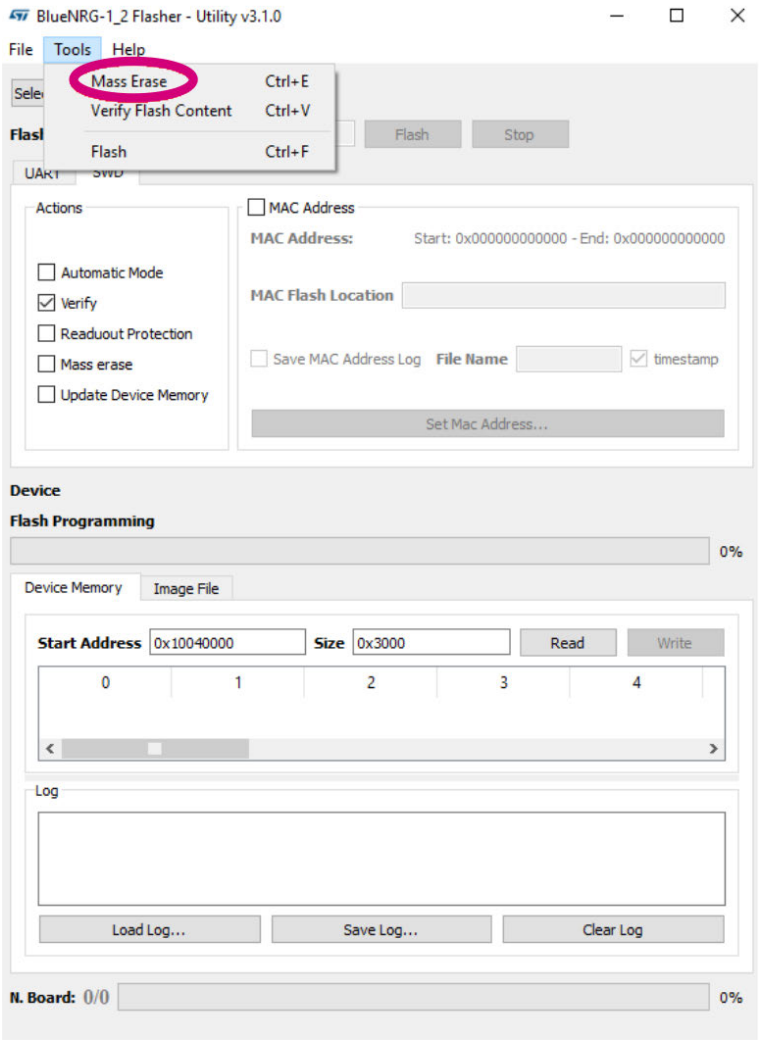
Step 5. Select the SWD tab.

Figure 57. STSW-BNRGFLASHER - SWD tab



Step 6. Erase the BlueNRG-2 chip Flash memory.

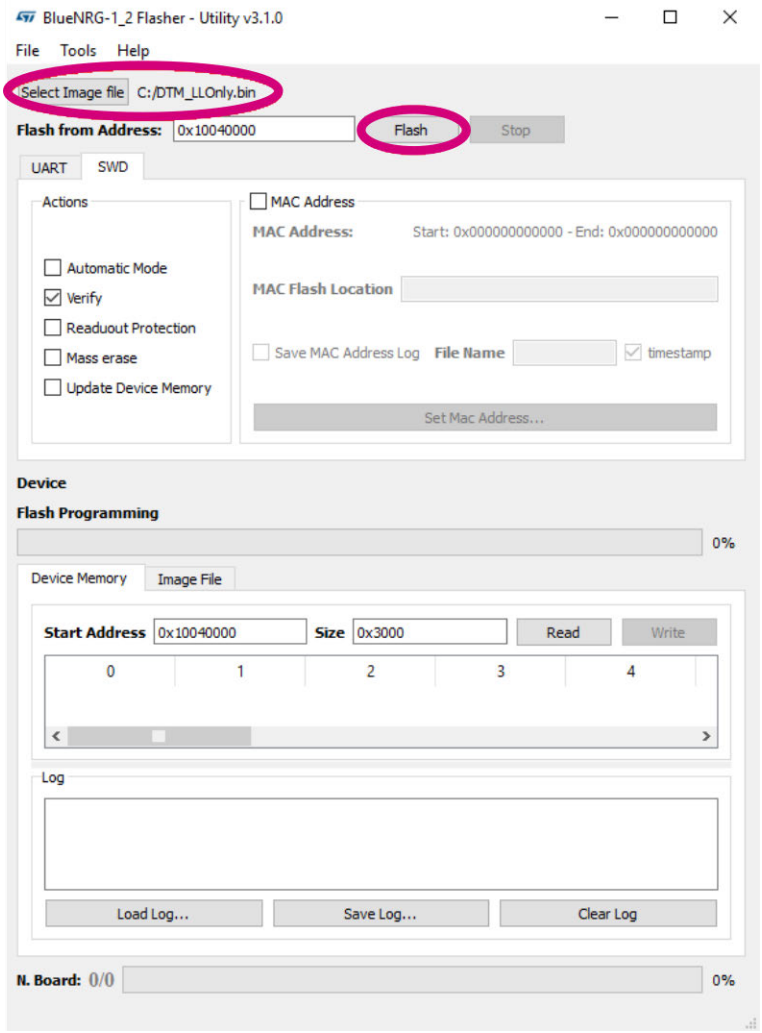
Figure 58. STSW-BNRGFLASHER - mass erase



Step 7. Download the Link Layer Only firmware for the BLE module.

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

Figure 59. STSW-BNRGFLASHER - Flash button



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

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

10.4.6 Important additional hardware information

For F4xx STM32 Nucleo development board, there is an hardware conflict among the X-NUCLEO-IKS01A2/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-IKS01A2:
  - in pins 5 (SB27) and 6 (SB26) there are the interrupts INT1 and INT2 for LSM6DSL (used for the featured hardware)
  - in pin 7 (SB25) there is the interrupt for LPS22HB 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).

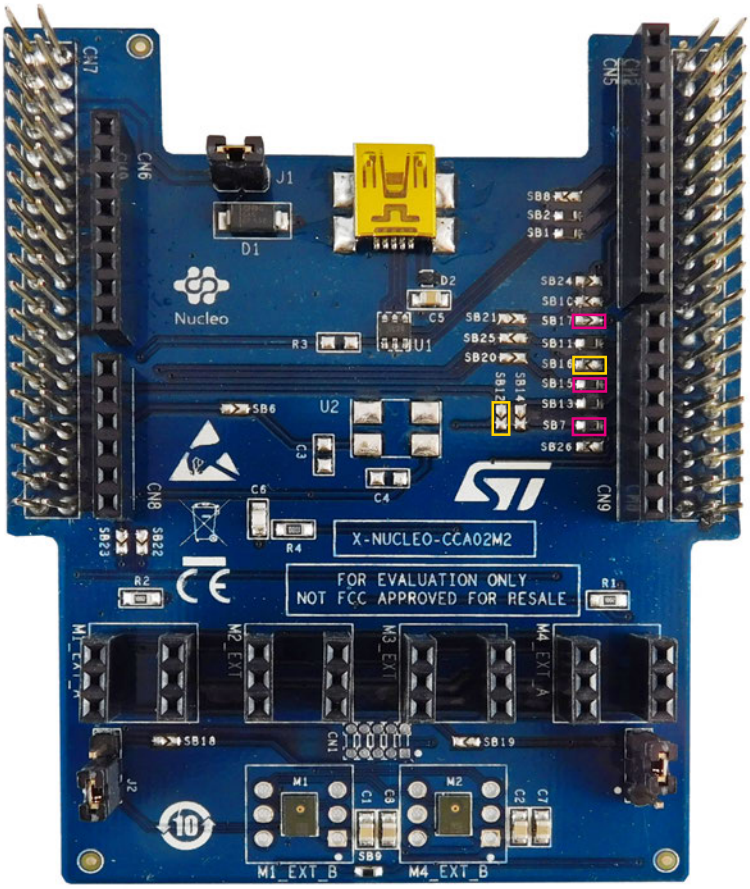
For NUCLEO-L476RG:

- 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 the solder bridge SB17
- before connecting the X-NUCLEO-IKS01A2 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 60. **X-NUCLEO-CCA02M2 solder bridge configuration for the NUCLEO-L476RG board**



Revision history

Table 2. **Document revision history**

Date	Version	Changes
09-Jul-2018	1	Initial release.
22-Aug-2019	2	Updated Introduction, Section 1.1 Overview, Figure 1. FP-SNS-ALLMEMS2 software architecture, Section 1.3 Folder structure, Section 2 Flash management, Section 3 The boot process, Section 4 The installation process, Section 5 Firmware-Over-The-Air (FOTA) update, Section 7 Sample application description, Section 8 Android and iOS sample client application, Section 8.5 Enable hardware features, Section 8.13 SD Logging, Section 8.14 Rssi and battery, Section 9 Firmware-Over-The-Air update with STBLESensor, Section 10.2 Software description and Section 10.3.3.5 Important additional hardware information. Added Section 10.1.5 X-NUCLEO-IKS01A3 expansion board, Section 10.1.7 STEVALMKSBOX1V1 evaluation kit and Section 10.3.3.3 STEVAL-MKSBOX1V1 setup.
26-May-2020	3	Updated Section 1.1 Overview, Section 1.2 Architecture, Section 1.3 Folder structure, Section 4 The installation process, Section 7 Sample application description, Figure 12. STBLESensor (Android version) main page following BLE connection, Section 8.2 MEMS sensor fusion, Section 8.3 Plot data, Section 8.4 Settings, serial and debug console, Section 8.5 Enable hardware features, Figure 22. STBLESensor (Android version) MotionAR activity recognition page, Figure 23. STBLESensor (Android version) MotionCP carry position recognition page, Figure 25. STBLESensor (Android version) MotionFX page, Section 8.10 BlueVoiceADPCM voice, Section 8.11 Direction of Arrival, Section 8.12 Beam Forming, Figure 33. STBLESensor (Android version) SD Logging information, Figure 34. STBLESensor (Android version) Battery and RSSI information, Section 9 Firmware-Over-The-Air update with STBLESensor, Section 10.2 Hardware setup, Section 10.3 Software setup, Section 10.4.1 STM32 Nucleo and expansion board setup and Section 10.4.6 Important additional hardware information. Added Section 8.10.1 Speech to text, Section 10.1.2 X-NUCLEO-CCA02M2 expansion board, Section 10.1.3 X-NUCLEO-BNRG2A1 expansion board and Section 10.4.5 X-NUCLEO-BNRG2A1 setup.

Contents

List of figures