UM1863
User manual

Getting started with the software package for BLE and sensors in BLUEMICROSYSTEM1

Introduction

The BLUEMICROSYSTEM1 expansion software package for STM32Cube lets you read and display real-time inertial and environmental sensor data through an Android™/iOS™ app.

The package implements application level functions based on the Bluetooth low energy protocol and enables communication with Android/iOS devices. Developers can use it to prototype applications with Android/iOS services, and transmit real-time sensor data with ease.

The software runs on the STM32 microcontroller and includes drivers that recognize the Bluetooth low energy (BlueNRG) hardware and sensor devices. The expansion is built on the STM32Cube framework for STM32 microcontrollers and comes with a sample implementation to kick-start development.

Contents

1 BLUEMICROSYSTEM1 software description 5

1.1 Overview 5
1.2 Architecture 6
1.3 Folder structure 7
1.4 Flash management 8
1.5 The boot process 9
1.6 The installation process 10
1.7 Firmware-Over-The-Air (FOTA) update 14
1.8 APIs 14
1.9 Sample application description 14
1.10 Open.MEMS licenses 18
1.11 Android and iOS sample client application 19
1.12 Firmware-Over-The-Air update with BlueMS 34
1.13 Open.MEMS license activation 37
1.13.1 Open.MEMS license activation with osxLicenseWizard 46

2 System setup guide 49

2.1 Hardware description 49
2.1.1 STM32 Nucleo platform 49
2.1.2 X-NUCLEO-IDB04A1 expansion board 50
2.1.3 X-NUCLEO-IDB05A1 expansion board 51
2.1.4 X-NUCLEO-IKS01A1 expansion board 52
2.1.5 X-NUCLEO-IKS01A2 expansion board 53
2.1.6 STEVAL-STLKT01V1 SensorTile development kit 53
2.2 Software description 57
2.3 Hardware and software setup 57
2.3.1 Hardware setup 58
2.3.2 Software setup 58
2.3.3 System setup guide 58

3 Revision history 62

List of figures

Figure 1: BLUEMICROSYSTEM1 overall system architecture 7
Figure 2: BLUEMICROSYSTEM1 package folder structure....7
Figure 3: BLUEMICROSYSTEM1 Flash structure 9
Figure 4: BootLoader folder content 10
Figure 5: BLUEMICROSYSTEM1 boot sequence....10
Figure 6: Content of one project folder 11
Figure 7: BootLoader and BLUEMICROSYSTEM1 installation....12
Figure 8: BootLoader and BLUEMICROSYSTEM1 dump process....13
Figure 9: Terminal setting 15
Figure 10: Initialization phase 16
Figure 11: UART console output when one device is connected to the board....18
Figure 12: BlueMS (Android version) main page following BLE connection 19
Figure 13: BlueMS (Android version) osxMotionFX sensor fusion page....20
Figure 14: BlueMS (Android version) popup windows....21
Figure 15: BlueMS (Android version) accelerometer plot....22
Figure 16: BlueMS (Android version) menu selection 23
Figure 17: BlueMS (Android version) Serial console (stdout/stderr) 24
Figure 18: BlueMS (Android version) Debug console (stdin/stdout/stderr) 25
Figure 19: BlueMS (Android version) hardware features 26
Figure 20: BlueMS (Android version) hardware feature examples: pedometer, wake up, orientation, double tap 27
Figure 21: BlueMS (Android version) osxMotionAR activity recognition page 28
Figure 22: BlueMS (Android version) osxMotionCP carry position recognition page....29
Figure 23: BlueMS (Android version) osxMotionGR gesture recognition page....30
Figure 24: BlueMS (Android version) osxMotionPM pedometer page 31
Figure 25: BlueMS (Android version) board LED control 32
Figure 26: BlueMS (Android version) Battery and RSSI information....33
Figure 27: BlueMS (Android version) firmware upgrade page 34
Figure 28: BlueMS (Android version) firmware update file selection....35
Figure 29: Terminal window information during FOTA 36
Figure 30: BlueMS (Android version) application page during FOTA and on completion....37
Figure 31: BlueMS (Android version) license status page....38
Figure 32: BlueMS (Android version) osxMotionCP license request....39
Figure 33: BlueMS (Android version) generated license request email 40
Figure 34: BlueMS (Android version) send request email 41
Figure 35: BlueMS (Android version) copy license details and select UPLOAD....42
Figure 36: BlueMS (Android version) paste license details 43
Figure 37: BlueMS (Android version) load license 44
Figure 38: BlueMS (Android version) osxMotionCP license enabled....45
Figure 39: osxLicenseWizard 46
Figure 40: osxLicenseWizard generated email 47
Figure 41: Received license activation email 48
Figure 42: STM32 Nucleo board....49
Figure 43: X-NUCLEO-IDB04A1 expansion board....50
Figure 44: X-NUCLEO-IDB05A1 expansion board....51
Figure 45: X-NUCLEO-IKS01A1 expansion board....52
Figure 46: LSM6DS3 DIL24 adapter board 52
Figure 47: X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board 53
Figure 48: STLCS01V1 board photo 54
Figure 49: STLCR01V1 board photo 56
Figure 50: STLCX01V1 board photo 57
Figure 51: STM32 Nucleo plus X-NUCLEO-IDB04A1 board....59

Figure 52: STM32 Nucleo plus X-NUCLEO-IDB05A1 plus X-NUCLEO-IKS01A1 plus LSM6DS3 DIL24 boards 60
Figure 53: SensorTile Cradle expansion board and Sensor Tile Cradle board connected to ST-LINK/V2-1 61

1 BLUEMICROSYSTEM1 software description

1.1 Overview

The key features of the BLUEMICROSYSTEM1 package are:

- For STM32 Nucleo expansion boards, complete middleware to build apps using:
  - temperature/humidity sensor (HTS221)
  - temperature/pressure sensors (LPS25HB/LPS22HB)
  - motion sensors (LIS3MDL/LSM303AGR and LSM6DS0/LSM6DSL)
  - compatible with the motion sensor LSM6DS3 DIL24 expansion
- For the STEVAL-STLKT01V1:
  - temperature and pressure sensor (LPS22HB)
  - motion sensors (LSM303AGR and LSM6DSM)
  - Gas Gauge (STC3115)
- Very low power BLE (BlueNRG) single-mode network processor for transmitting information to one client.
- osxMotionFX real-time motion sensor data fusion (OPEN.MEMS license) to combine the output from multiple MEMS sensors.
- Accelerometer-only real-time recognition algorithms:
  - osxMotionARactivity
  - osxMotionCP carry position
  - osxMotionGR gesture
  - osxMotionPM pedometer
  - osxMotionID motion intensity
- Free, user-friendly license terms
- BlueMS compatible application for Android/iOS (version 2.0.0 or above) for visualizing information sent via Bluetooth.
- OTA firmware update (for X-NUCLEO-IDB05A1 Bluetooth board only) using the BlueMS application (Ver. 3.0.0 or higher)
- Option to request and enable the OPEN.MEMS license using the BlueMS application (Ver. 3.0.0 and above)
- Gas Gauge STEVAL-STLKT01V1 visible using BlueMS application (Ver. 3.2.0 and above)
- Separate sample implementations for X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A1) and X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1) boards on a NUCLEO-F401RE or NUCLEO-L476RG board and for a STEVAL-STLKT01V1 board

This software creates the following Bluetooth services:

1. The first service exposes all the hardware features with the following characteristics:
   - Temperature
   - Pressure
   - Humidity
   - 3D gyroscope, 3D magnetometer, 3D accelerometer
   - LED status
   - Gas Gauge: Battery %, Voltage, and status (charging/discharging/low battery) only for STEVAL-STLKT01V1
2. The second service exposes the software characteristics:
   - quaternions generated by the osxMotionFX library in short precision
   - magnetic North direction (e-Compass)
   - recognized activity using the osxMotionAR algorithm
   - recognized carry position using the osxMotionCP algorithm

  - recognized gesture using the osxMotionGR algorithm
  - Number of steps and frequency using the osxMotionPM algorithm
  - recognized motion intensity using the osxMotionID
3. The third service exposes the console service with:
  - stdin/stdout for bi-directional communication between client and server
  - stderr for a mono-directional channel from the STM32 Nucleo board to an Android/iOS device
4. The last service is for transmitting/resetting the calibration status, for switching on/off the LED and enabling the following expansion hardware features for STM32 Nucleo boards with LSM6DSL or LSM6DS3 DIL24 boards or STEVAL-STLKT01V1 boards with LSM6DSM motion sensor:
  - Pedometer
  - Free fall detection
  - Single tap detection
  - Double tap detection
  - Wake up detection
  - Tilt detection
  - 3D orientation

This software gathers:

- the temperature, humidity, pressure, audio and motion sensor drivers for the HTS221, LPS25H/LPS22HB, LSM6DS0/LSM6DSL (or LSM6DS3) and LIS3MDL/LSM303AGR devices for STM32 Nucleo expansion boards
- the temperature, pressure and motion sensor drivers for the LPS22HB, LSM6DSM and LSM303AGR devices for STEVAL-STLKT01V1 running on STM32.

This package is compatible with the BlueMS Android/iOS (Ver. 2.2.0 or higher) application available at respective Play/iTunes stores, which can be used for displaying information sent via the Bluetooth low energy protocol. BlueMS Version 3.0.0 and above is required for Over-The-Air firmware updates (for X-NUCLEO-IDB05A1 Bluetooth low energy expansion board only) and requesting and enabling the Open.MEMS license. BlueMS Version 3.2.0 and above is required for showing the Gas Gauge information for STEVAL-STLKT01V1

1.2 Architecture

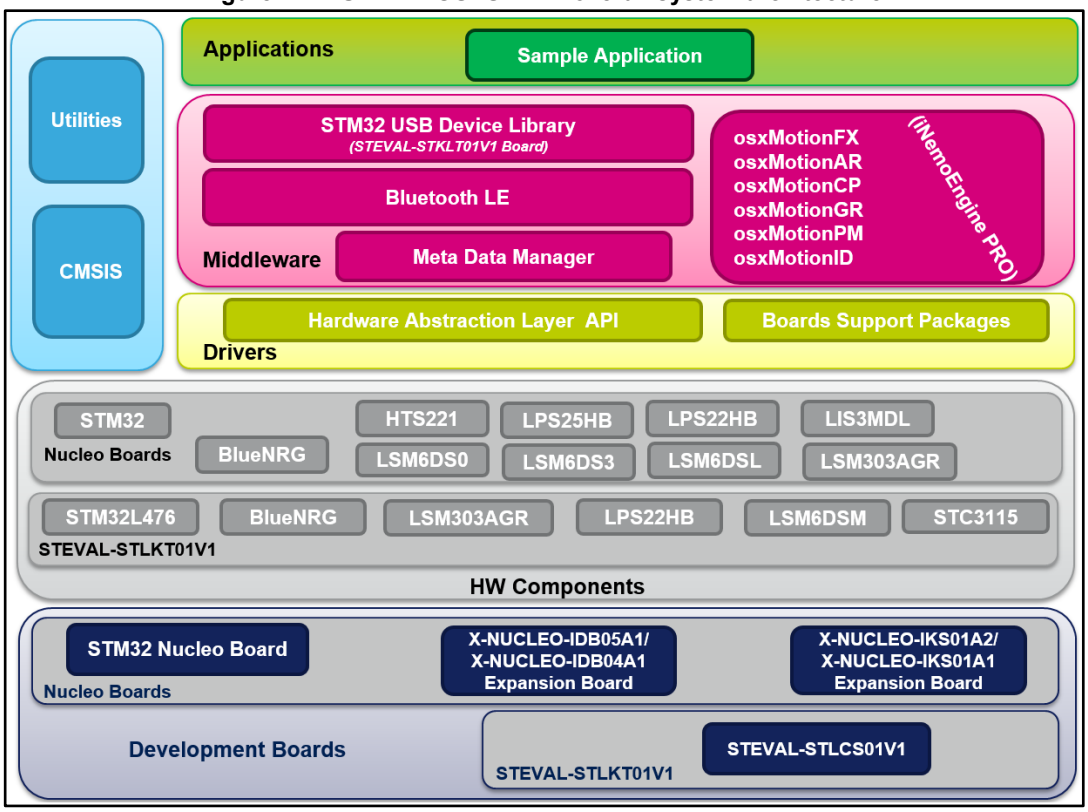
The software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller.

The package provides a Board Support Package (BSP) for the sensors expansion board and middleware components for serial communication with a PC.

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

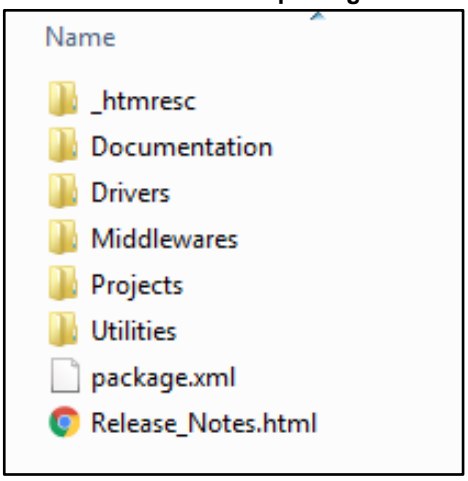
- **STM32Cube HAL layer**: provides a simple, generic and multi-instance set of generic and extension APIs (application programming interfaces) to interact with the upper layer application, libraries and stacks. It is directly based on a generic architecture and allows the layers that are built on it, 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: BLUEMICROSYSTEM1 overall system architecture



1.3 Folder structure

Figure 2: BLUEMICROSYSTEM1 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 ARM Cortex-M processor series.

- **Middlewares**: contains libraries and protocols for BlueNRG Bluetooth low energy,USB Device Library, the Meta Data Manager, osxMotionFX (iNEMOEngine PRO) sensors fusion library, osxMotionAR (iNEMOEngine PRO) activity-recognition library, osxMotionCP (iNEMOEngine PRO) carry-position recognition library, osxMotionGR (iNEMOEngine PRO) gesture recognition library, osxMotionPM real-time pedometer (iNEMOEngine PRO) library, osxMotionID (iNEMOEngine PRO) motion intensity recognition library.
- **Projects**: contains a sample application used for transmitting the output of the sensor data and of the osxMotionFX sensor fusion and e-Compass, osxMotionAR activity-recognition, osxMotionID motion-intensity-recognition, osxMotionCP carry-position, osxMotionGR gesture recognition and osxMotionPM pedometer libraries by using the Bluetooth low energy protocol provided for the NUCLEO-F401RE/NUCLEO-L476RG 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 STM32F401RE and STM32L476RG

1.4 Flash management

Apart from storing its code, BLUEMICROSYSTEM1 uses the Flash memory for:

1. saving the Open.MEMS license activated for the board inside the Meta Data Manager
2. allowing the Firmware-Over-The-Air update

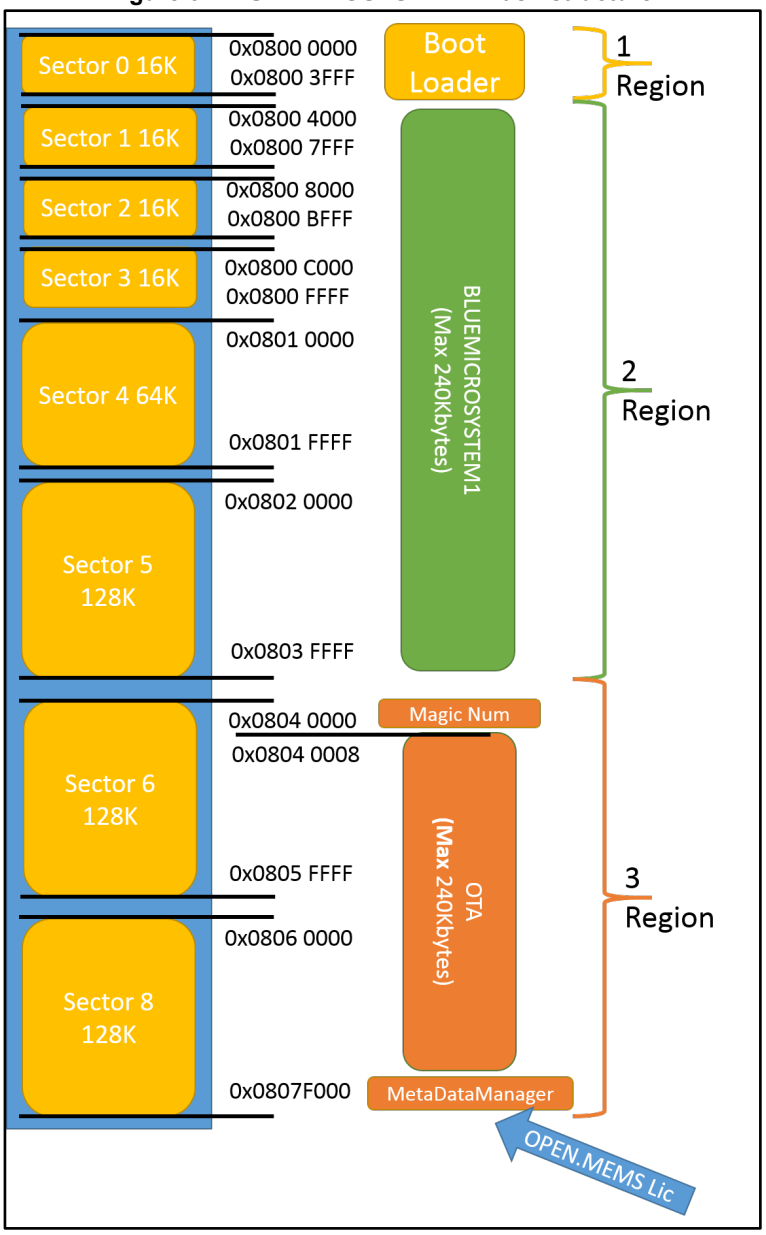
To enable these features the Flash is divided into these different regions (see *Figure 3: "BLUEMICROSYSTEM1 Flash structure"*):

1. the first region contains a custom boot loader
2. the second region contains the BLUEMICROSYSTEM1 firmware
3. The third region is used for storing the FOTA before the update and for saving the Open.MEMS license inside the Meta Data Manager

Even if the STM32F401RE (512 KB) and the STM32L476RG (1024 KB) cache sizes and arrangements differ, we have used the same Flash arrangement for both to save the FOTA. The Meta Data Manager is placed at the end of the FLASH (0x08007000 for STM32F401RE and 0x080FF000 for STM32L476RG). For more information, refer to:

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

Figure 3: BLUEMICROSYSTEM1 Flash structure



1.5 The boot process

The BLUEMICROSYSTEM1 cannot 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 STM32F401) and Src/system_stm32l4xx.c (for STM32L476) thus: `#define VECT_TAB_OFFSET 0x4000`.

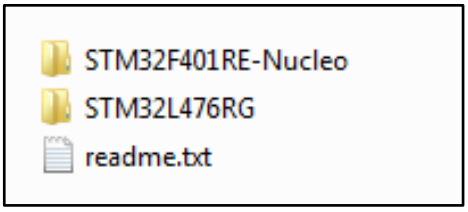
We also changed the linker script. For example, the Linker script for BLUEMICROSYSTEM1 running on STM32F401RE and compiled using IAR Embedded Workbench for ARM is:

```
define symbol   ICFEDIT intvec start   = 0x08004000;
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__ = 0x08004000;
define symbol __ICFEDIT_region_ROM_end__ = 0x0803FFFF;
define symbol   ICFEDIT region RAM start   = 0x20000000;
define symbol   ICFEDIT region RAM end   = 0x20017FFF;
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x8000;
define symbol __ICFEDIT_size_heap__ = 0x800;
```

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

You must flash the appropriate bootloader binary for STM32F401RE or STM32L476RG , found in the Utilities\BootLoader folder, to the first FLASH region (address 0x08000000).

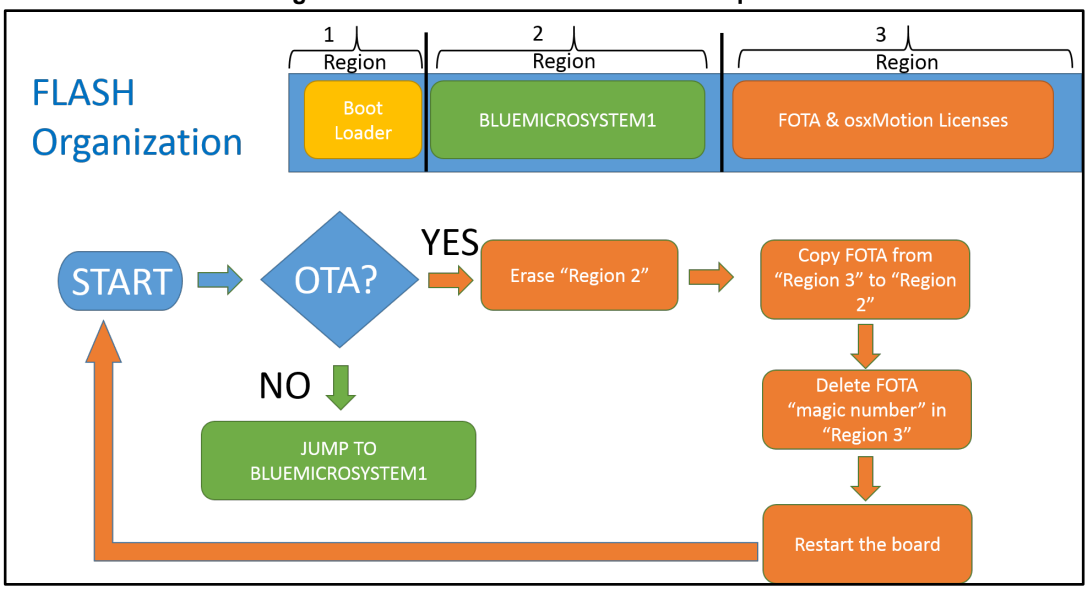
Figure 4: BootLoader folder content



On any board reset:

- If there is a FOTA in the third Flash region, the boot loader overwrites the second Flash region (with BLUEMICROSYSTEM1 firmware) and replaces its content with the FOTA and restarts the board.
- If there is no FOTA, the boot loader jumps to the BLUEMICROSYSTEM1 firmware.

Figure 5: BLUEMICROSYSTEM1 boot sequence



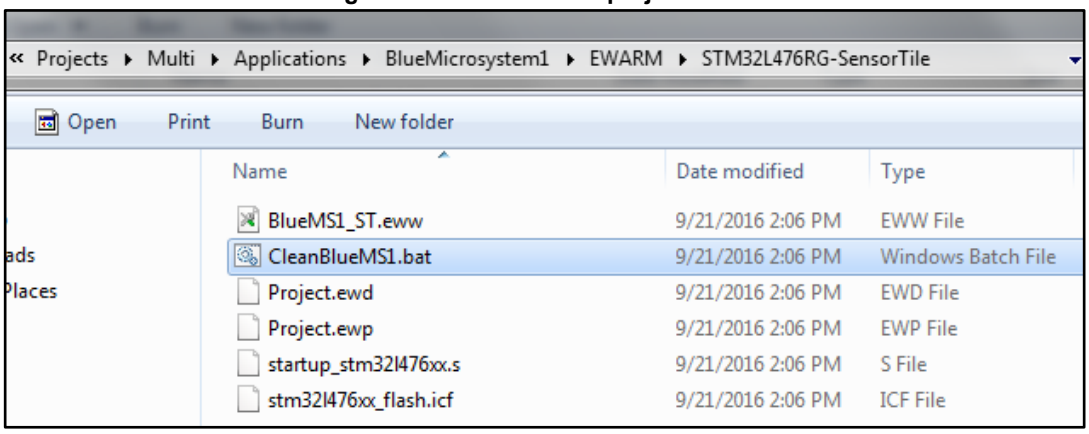
1.6 The installation process

The package contains in a single image in the Binary directory (in .bin and .hex format) that includes the BootLoader for the BLUEMICROSYTEM1 firmware for each platform (NUCLEO-F401RE, NUCLEO-L476RG and STEVAL-STLKT01V1). This image can be directly flashed on the platform.

When it is necessary to flash different BLUEMICROSYSTEM1 firmware, after having modified the code and compiled, you just need to flash the BLUMICROSYTEM1 firmware at the right position (0x08004000).

To simplify the BLUEMICROSYSTEM1 installation by saving the BLUEMICROSYSTEM1 firmware and BootLoader to the right positions for each platform (NUCLEO-F401RE, NUCLEO-L476RG, STEVAL-STLKT01V1) and for each IDE (IAR/RealView/System Workbench), there is a script called CleanBlueMS1.bat.

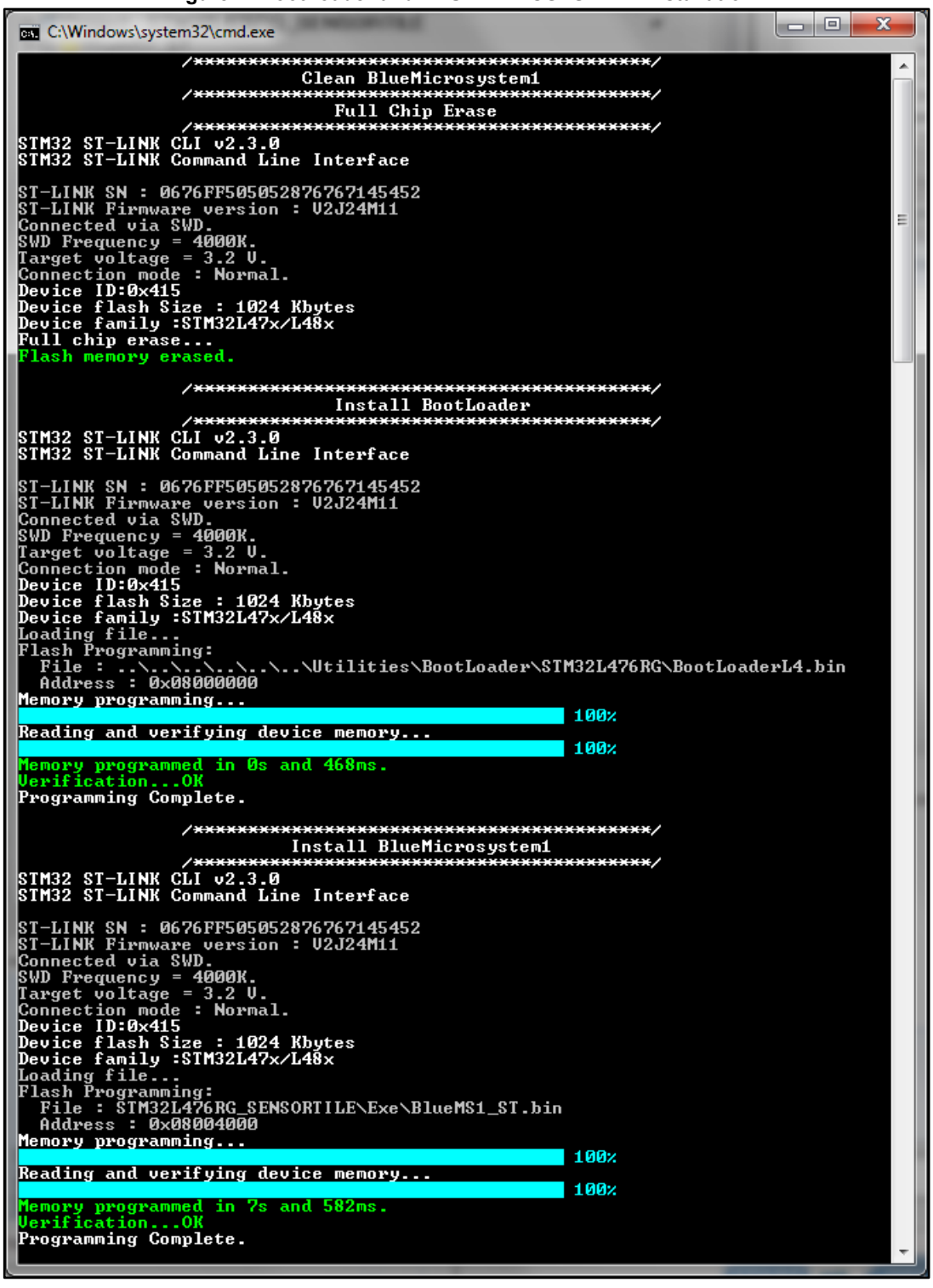
Figure 6: Content of one project folder



This script:

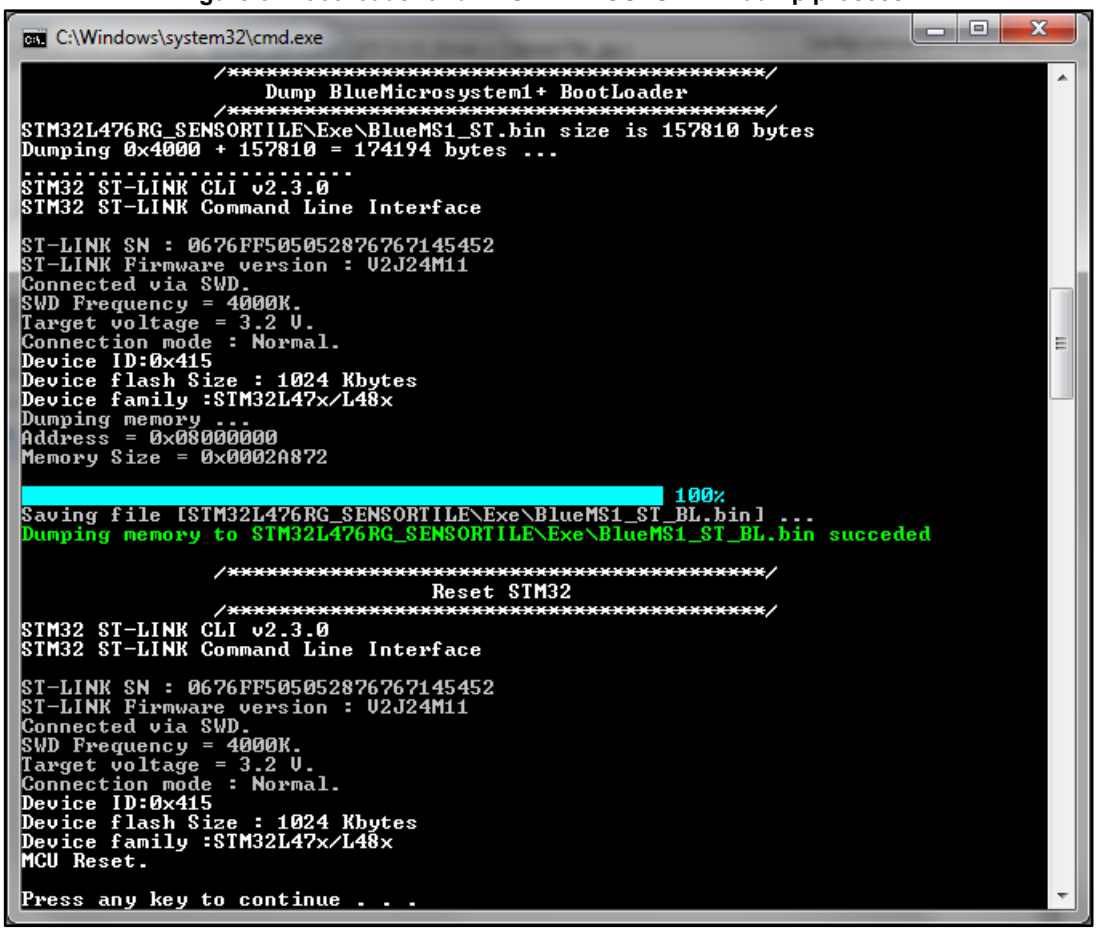
- Performs a full Flash erase
- Flashes the BootLoader at the correct position (0x08000000)
- Flashes the BLUEMICROSYSTEM1 firmware at the correct position 0x08004000

Figure 7: BootLoader and BLUEMICROSYSTEM1 installation



The same script also dumps a unique image file containing the BootLoader and the BLUEMICROSYSTEM1 firmware. This image file may be directly flashed to the beginning of the Flash memory in the same way as the image contained in the Binary folder.

Figure 8: BootLoader and BLUEMICROSYSTEM1 dump process



For the Linux/iOS operating system, there is a similar script that uses OpenOCD instead of the ST-LINK command line. The script for OpenOCD is included only inside the System Workbench IDE for each platform and is called CleanBlueMS1.sh.

To make it function, edit the section of the CleanBlueMS1.sh below by adding:

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

```
# 1) Set the Installation path for OpenOCD
# example:
#OpenOCD DIR="C:/Ac6/SystemWorkbench/plugins/fr.ac6.mcu.externaltools.openocd.win32
1.10.0.201607261143/tools/openocd/"
OpenOCD_DIR=""

# 2) Set the installation path for stm32 OpenOCD scritps
# example:
#OpenOCD CFC="C:/Ac6/SystemWorkbench/plugins/fr.ac6.mcu.debug 1.10.0.201607251855/re
sources/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

For the X-NUCLEO-IDB05A1 Bluetooth low energy expansion board only, the BLUEMICROSYSTEM1 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 BLUEMICROSYSTEM1 checks against the hardware cyclic redundancy check calculation unit on the STM32F401/STM32L476 processor to ensure the update integrity. If the CRC calculation matches the BlueMS CRC value, the new firmware is written from 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 BLUEMICROSYSTEM1 firmware (see *Section 1.12: "Firmware-Over-The-Air update with BlueMS"*).

1.8 APIs

Detailed user-API function and parameter descriptions are compiled in an HTML file in the package Documentation folder.

1.9 Sample application description

A sample application is provided in the Projects folder for:

- the X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A1) and X-NUCLEO-IDB04A1 (or X-NUCLEO-IDB05A1) expansion boards with the NUCLEO-F401RE or NUCLEO-L476RG board
- STEVAL-STLKT01V1

Ready to build projects are available for multiple IDEs.

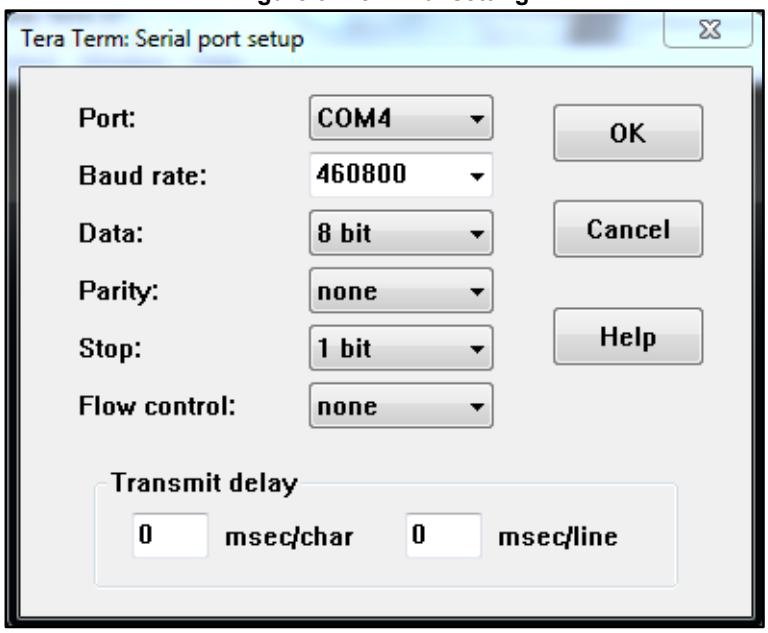
With the NUCLEO-F401RE and NUCLEO-L476RG boards, you can set up a terminal window for the appropriate UART communication port (use the baud, data, parity and stop settings below) to control the initialization phase.

The same feature is available for the STEVAL-STLKT01V1 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. In fact, a 10-second delay has been added to the initialization phase to allow you to follow its progress.

This feature for STEVAL-STLKT01V1 is disabled by default. To enable it, you must modify the osx_bms_config.h file by enabling the `//#define OSX_BMS_ENABLE_PRINTF` line below.

```
#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 BlueMicrosystem Initialization phase */
    //#define OSX_BMS_ENABLE_PRINTF
#else /* STM32 SENSORTILE */
  /* For Nucleo it's enable by default */
  #define OSX BMS ENABLE PRINTF
#endif /* STM32_SENSORTILE */
#endif /* STM32_BLUECOIN */
```

Figure 9: Terminal setting

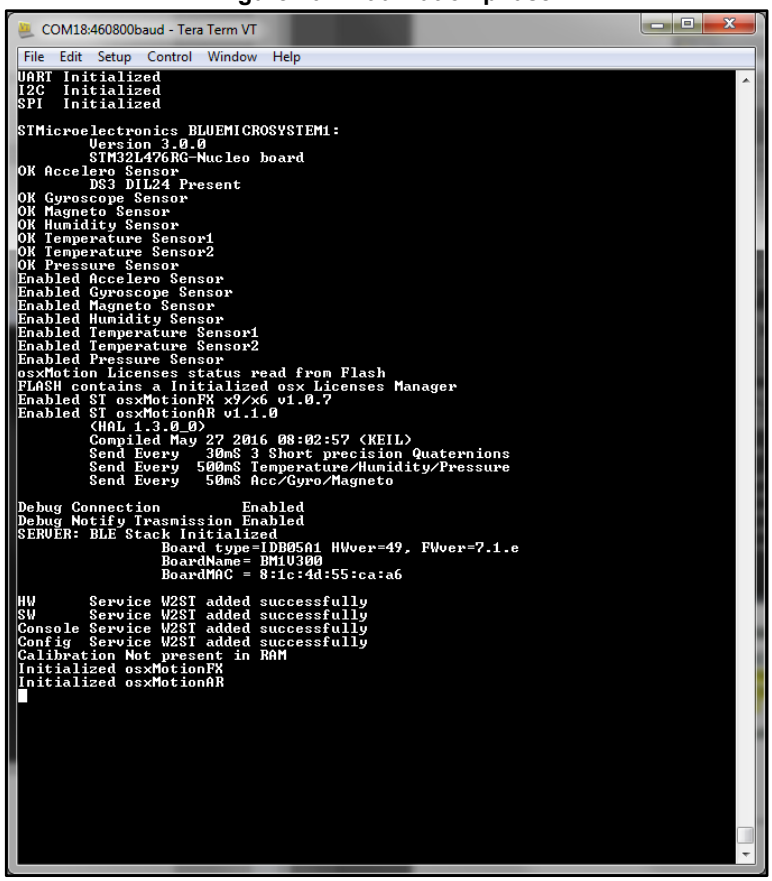


When you first press the reset button on the NUCLEO-F401RE or NUCLEO-L476RG board, the application:

- starts initializing the UART, I²C and SPI interfaces
- determines which MEMS expansion board is connected to STM32 Nucleo board.
- checks whether all the sensors are present and working
- checks whether the LSM6DS3 DIL24 extension is present
- For STEVAL-STLKT01V1 checks if the battery is present and initializes the Gas Gouge
- checks whether the osxMotionFX, osxMotionAR, osxMotionCP, osxMotionGR, osxMotionPM and osxMotionID libraries have been initialized properly with valid licenses read from Flash
- determines which BlueNRG expansion board is connected to the STM32 Nucleo board (X-NUCLEO-IDB04A1 or X-NUCLEO-IDB05A1) and hardware and firmware version information
- creates a random BLE MAC address
- initializes the BLE hardware service, adding the temperature, humidity, pressure, 3D gyroscope, 3D magnetometer, 3D accelerometer and LED characteristics
- initializes the BLE SW service adding the osxMotionFX, osxMotionAR, osxMotionCP, osxMotionGR and osxMotionID libraries (if present in Flash).
- initializes the BLE console service adding the stdin/stdout and stderr characteristics
- initializes the BLE config service transmitting/resetting the calibration status and enabling the LSM6DS3 DIL24 hardware features (if present).

If the LSM6DS3 DIL24 extension is present, or if the X-NUCLEO-IKS01A2 is mounted, it can generate an interrupt signaling a free fall, tilt, wake up, single tap, double tap, 6D position or pedometer event, which is transmitted over Bluetooth to the attached Android™/iOS™ device.

Figure 10: Initialization phase



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

- 3 short precision quaternions every 30 ms
- Temperature/Humidity/Pressure data (and Battery status if the battery is present for STEVAL-STLKT01V1) every 500 ms
- 3D Accelerometer, 3D Gyroscope and 3D Magnetometer data every 50 ms

This application reads the accelerometer, magnetometer and gyroscope values at 100 samples/second. The osxMotionFX (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.

These definitions in osx_bms_config.h control how many quaternions the application sends to the Bluetooth client:

- `QUAT_UPDATE_MUL_10MS`: defines the transmission rate for each set of quaternions by multiple of 10 ms.
- `SEND_N_QUATERNIONS`: defines how many quaternions are sent on each Bluetooth package.

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

The same osx_bms_config.h file also defines:

- `OSX_BMS_DEBUG_CONNECTION` and `OSX_BMS_DEBUG_NOTIFY_TRAMISSION` to enable some debugging information for BLE communication

- `OSX_BMS_MOTIONID` to enable the osxMotionID motion intensity recognition algorithm

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

- On the NUCLEO-F401RE or NUCLEO-L476RG boards, you can press the user button to reset the library calibration status and force a new auto-calibration procedure.
- With the STEVAL-STLKT01V1, you can only do this through the BlueMS application.

The osxMotionAR (iNEMOEngine PRO) library can recognize the following activities:

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

The osxMotionCP (iNEMOEngine PRO) library recognizes and provides real-time information about how the user is carrying the board, which equates to the phone carry position:

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

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

- pick up
- glance
- wake up in hand

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

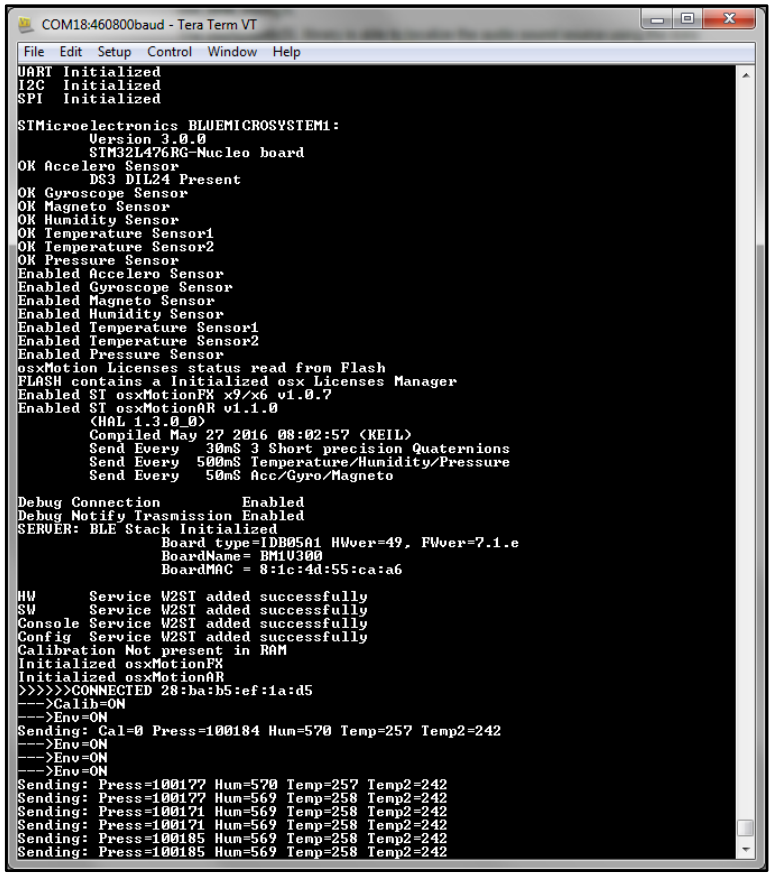
The osxMotionID (INEMOEngine PRO) library can recognize the following activities:

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

The osxMotionID is enabled via the `OSX_BMS_MOTIONID` define in osx_bms_config.h.

When an Android/iOS device is connected to the NUCLEO-F401RE or NUCLEO-L476RG board (or for STEVAL-STLKT01V1 enabling the debugging messages), it is possible to control what the board transmits (see the following figure).

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



1.10 Open.MEMS licenses

The BLUEMICROSYSTEM1 includes:

1. osxMotionFX (iNEMOEngine PRO) real-time motion sensor data fusion algorithm
2. osxMotionAR (iNEMOEngine PRO) real- time activity-recognition algorithm
3. osxMotionCP (iNEMOEngine PRO) real-time carry-position recognition algorithm
4. osxMotionGR (iNEMOEngine PRO) real- time gesture recognition algorithm
5. osxMotionPM (iNEMOEngine PRO) real time pedometer algorithm
6. osxMotionID (iNEMOEngine PRO) real time motion intensity recognition algorithm

The algorithms are released under the Open.MEMS limited license agreement (LLA) and are node-locked to a specific STM32 board; you need to obtain a license from STMicroelectronics to use them. Licenses can be requested via the Android/iOS device BlueMS application (ver. 3.0.0 and above) available on their respective application market stores, or by using the OSX License Wizard if `OSX_BMS_LICENSE_H_FILE` in Inc\osx_bms_config.h is enabled.

Licenses are saved in the last section of the third Flash region so they are not lost on board resets.

Refer to *Section 1.13: "Open.MEMS license activation"* for more information regarding license requests.

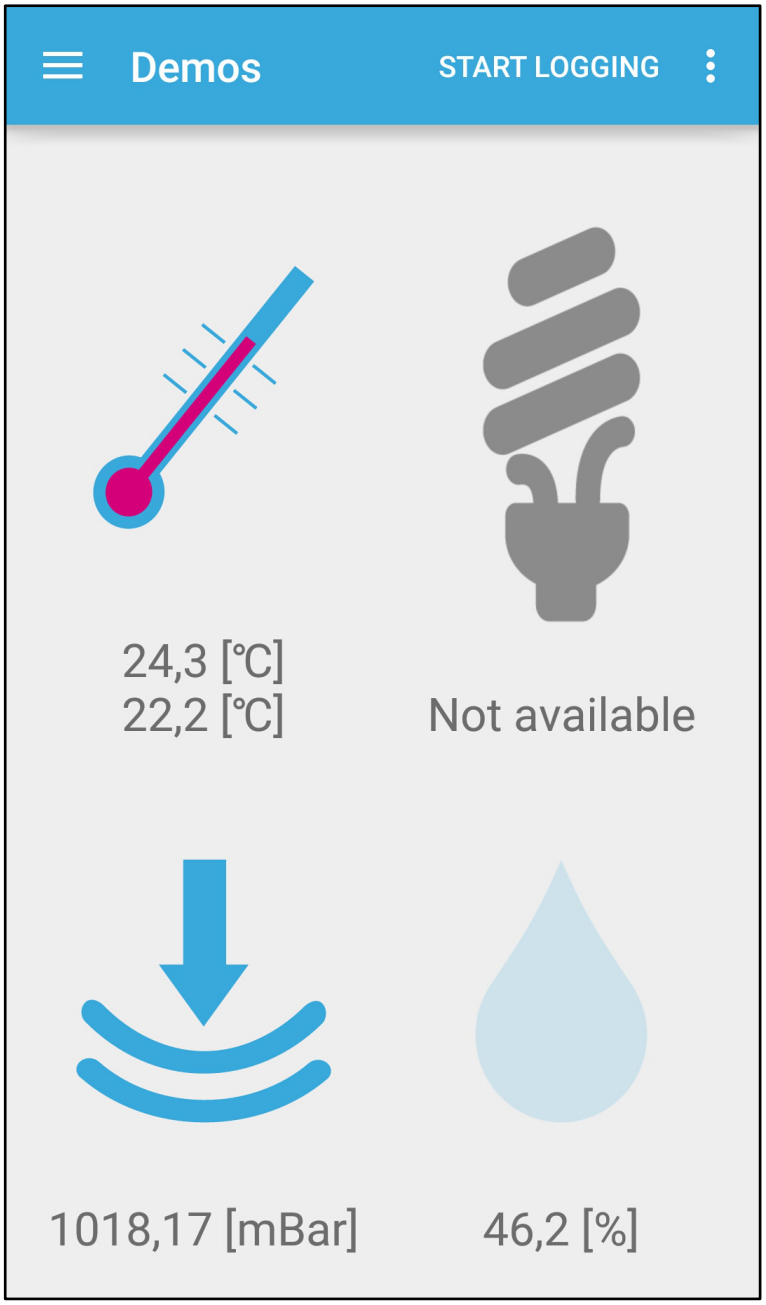
1.11 Android and iOS sample client application

The BLUEMICROSYSTEM1 software for STM32Cube is compatible with the BlueMS Android/iOS applications (ver. 2.2.0 or higher) available at the respective Play/iOS stores. Version 3.0.0 and above is required for Over-The-Air firmware updates (for X-NUCLEO-IDB05A1 Bluetooth low energy expansion boards only) and to request and enable the Open.MEMS. Version 3.2.0 and above is required to display Gas Gauge battery information (Battery %, volt, and charge status) for STEVAL-STLKT01V1 board.

We use the Android application in this example.

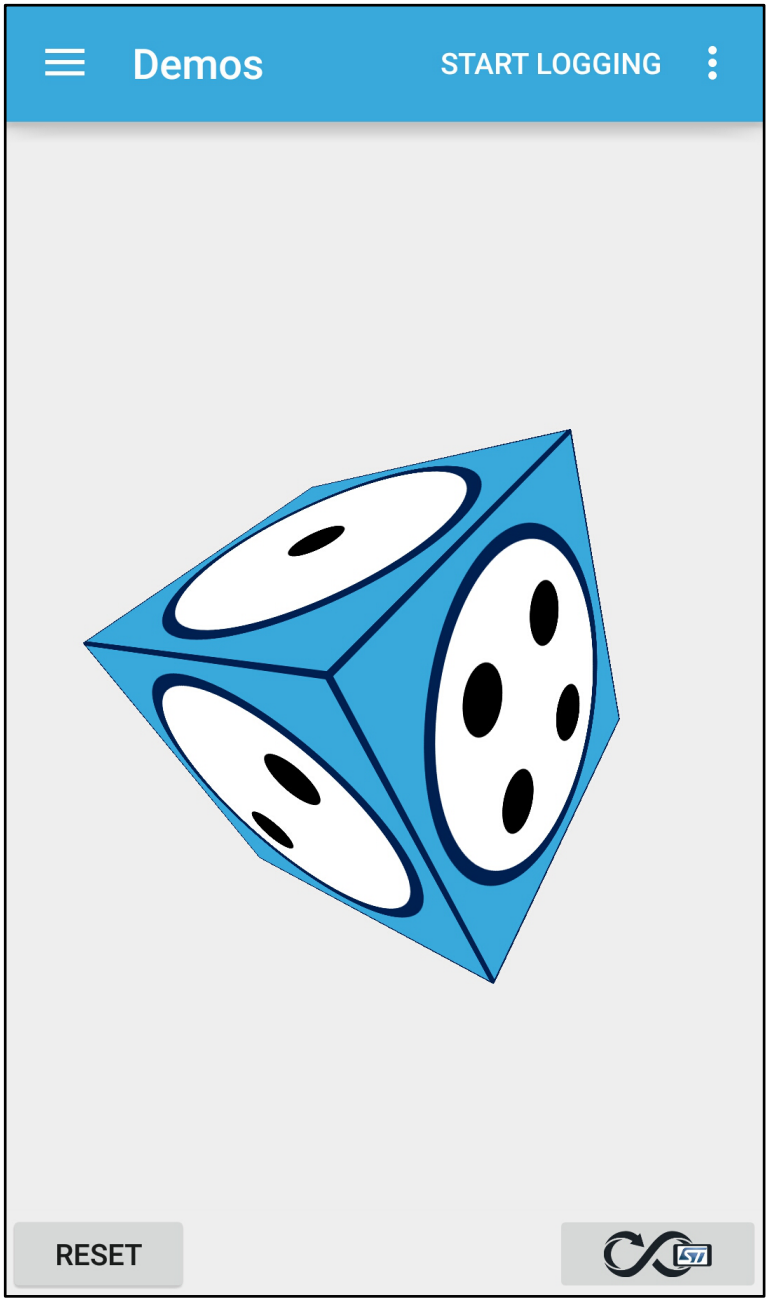
Following connection, BlueMS starts with the main page shown below, where the values of temperature, pressure and humidity are displayed.

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



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

Figure 13: BlueMS (Android version) osxMotionFX sensor fusion page

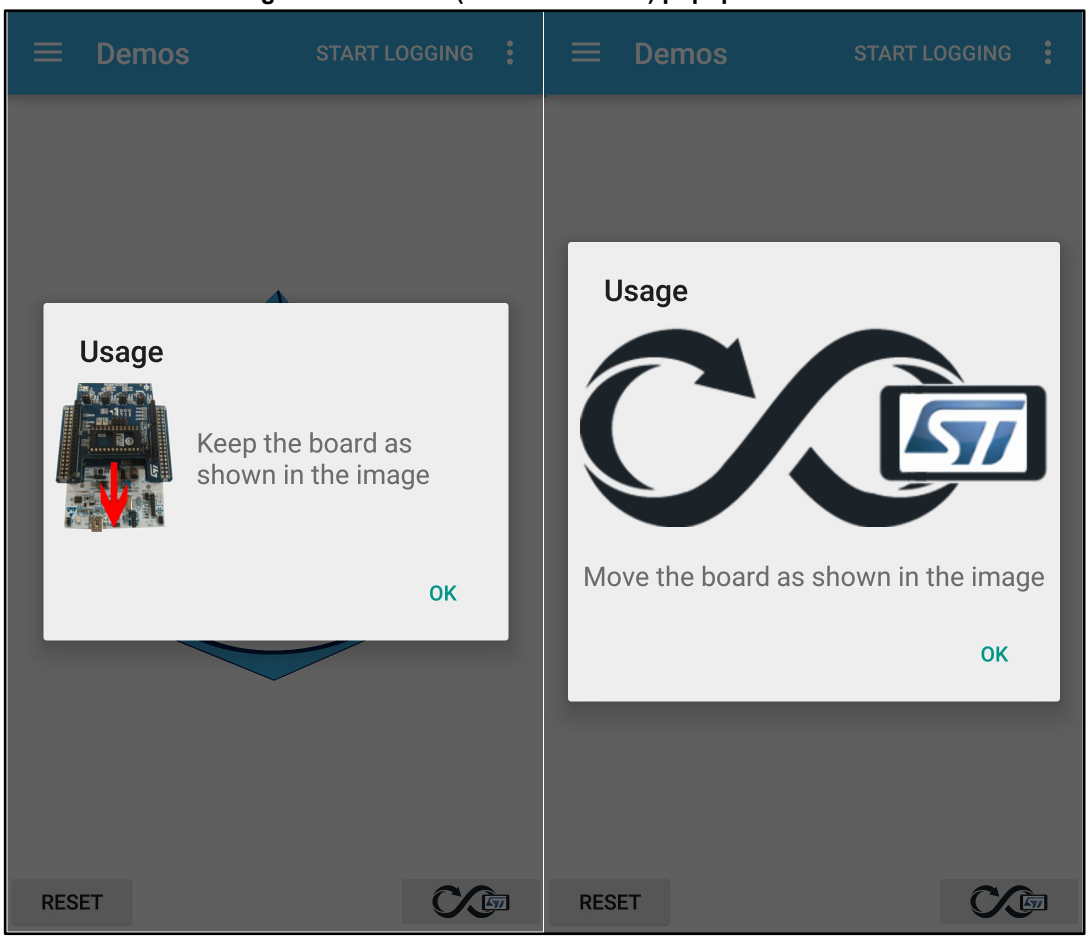


On this page, there are two buttons along the bottom:

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

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

Figure 14: BlueMS (Android version) popup windows



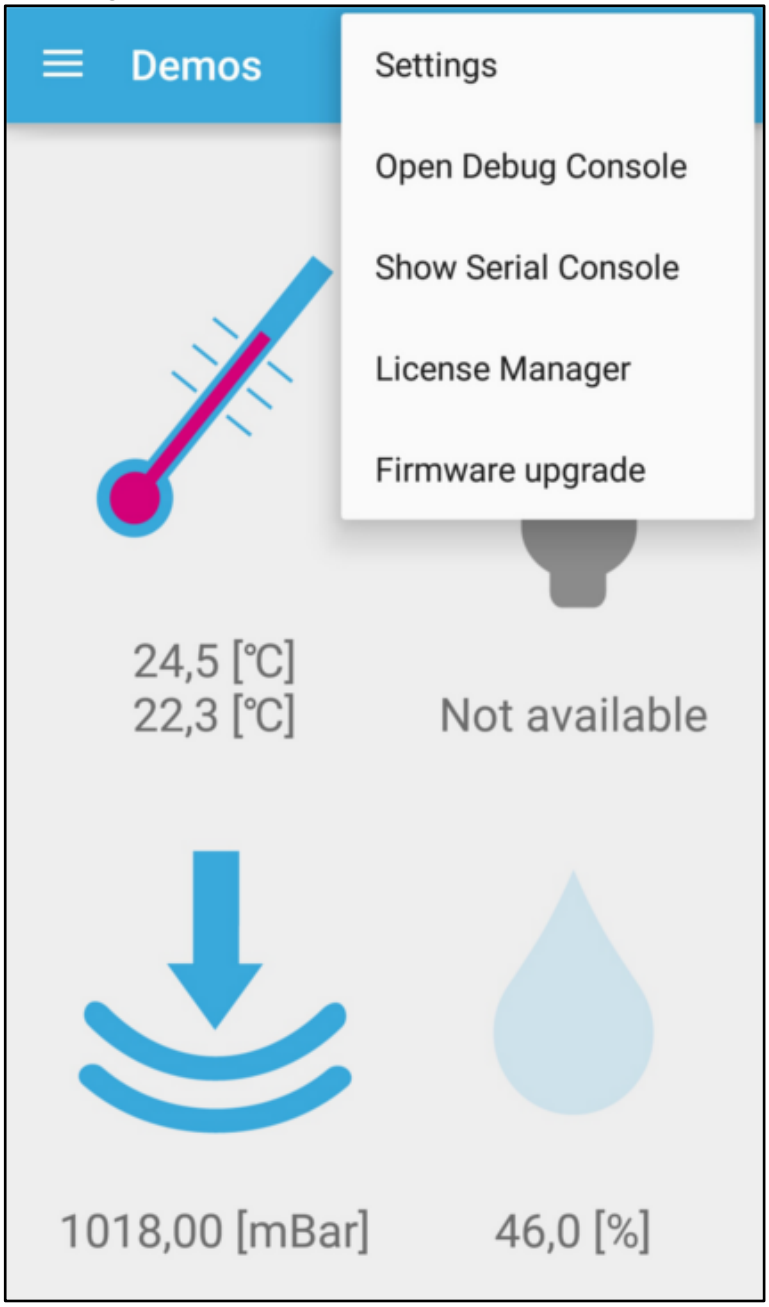
On the next page to the left, you can plot any value from the sensor expansion boards.

Figure 15: BlueMS (Android version) accelerometer plot

In the option menu below, you can open:

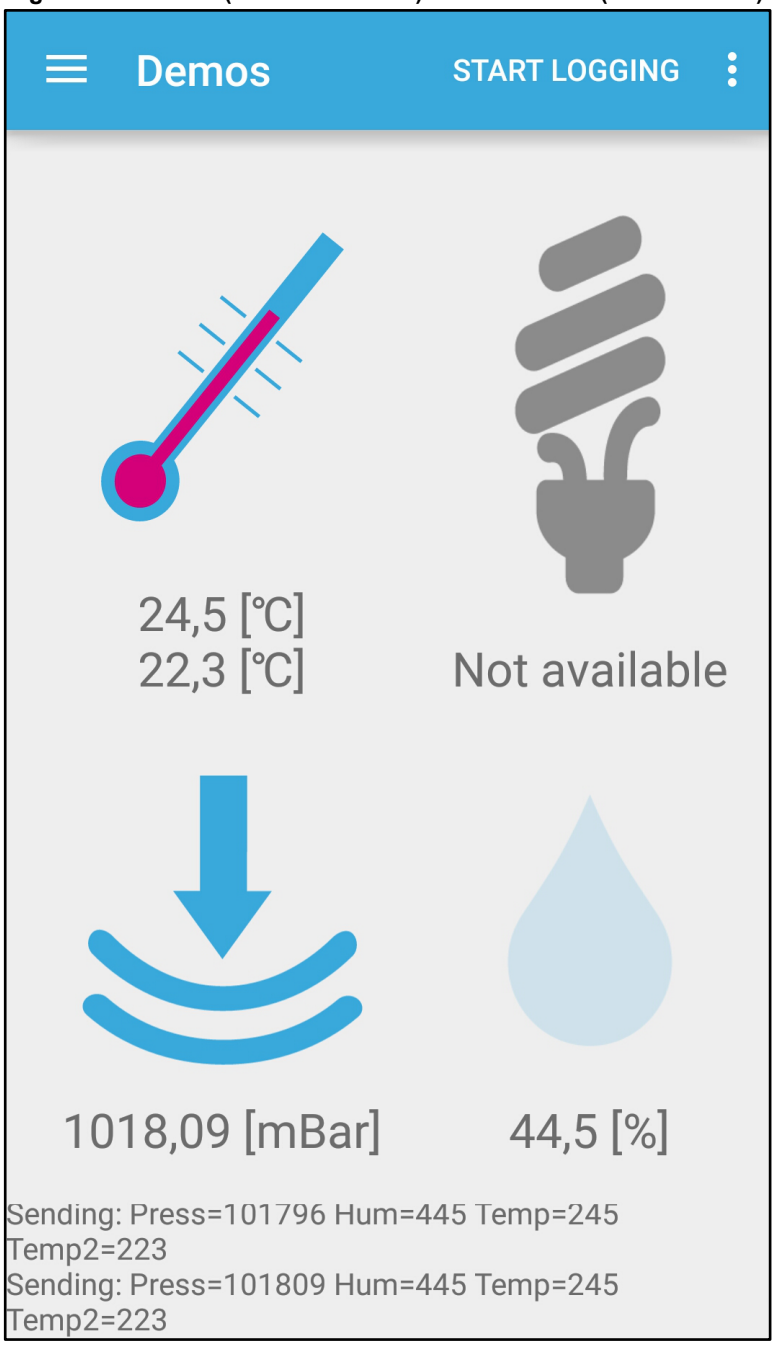
- Serial or Debug (with stdin) console
- License manager
- Firmware upgrade

Figure 16: BlueMS (Android version) menu selection



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

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



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

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

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:

- LSM6DS3 DIL24 (if present) for STM32 Nucleo expansion boards
- LSM6DSL for X-NUCLEO-ISK01A2 for STM32 Nucleo expansion boards
- LSM6DSM for STEVAL-STLKT01V1

The orientation hardware feature is the default setting.

Figure 19: BlueMS (Android version) hardware features

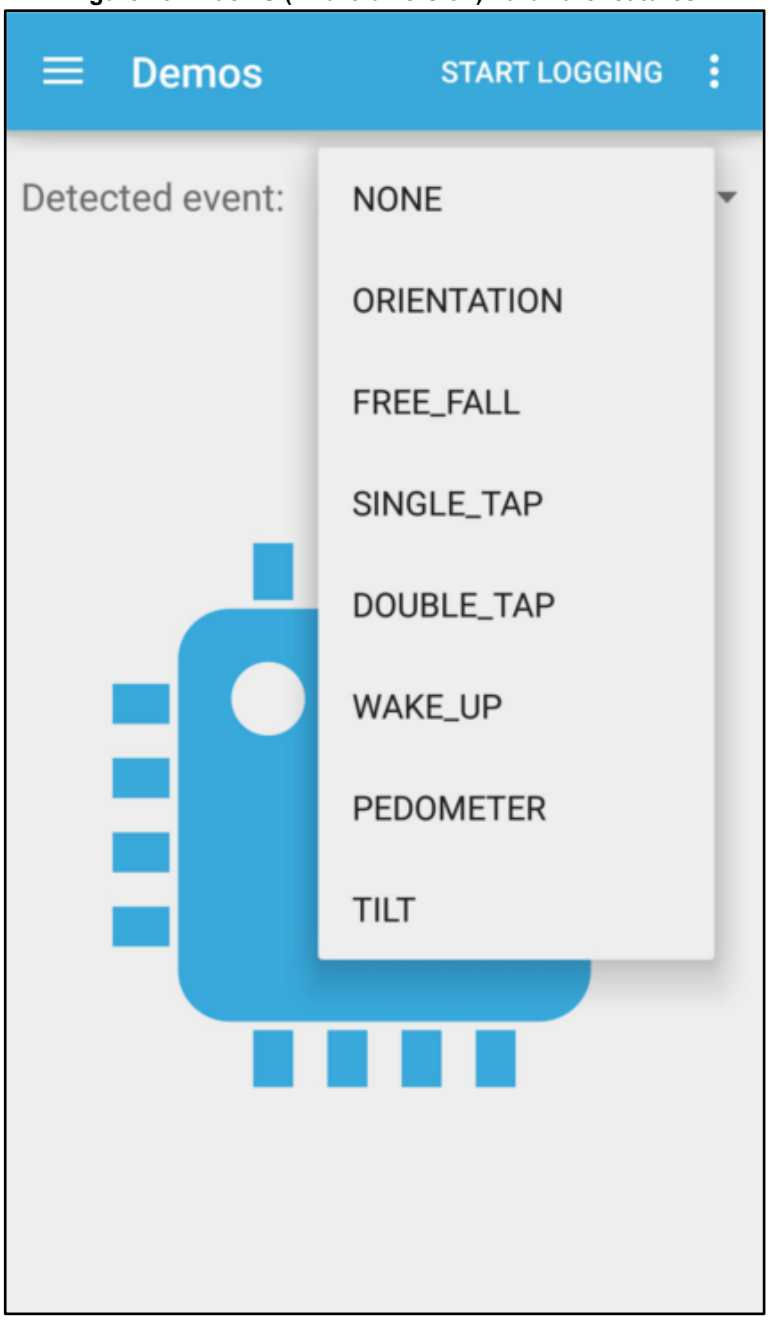


Figure 20: BlueMS (Android version) hardware feature examples: pedometer, wake up, orientation, double tap

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

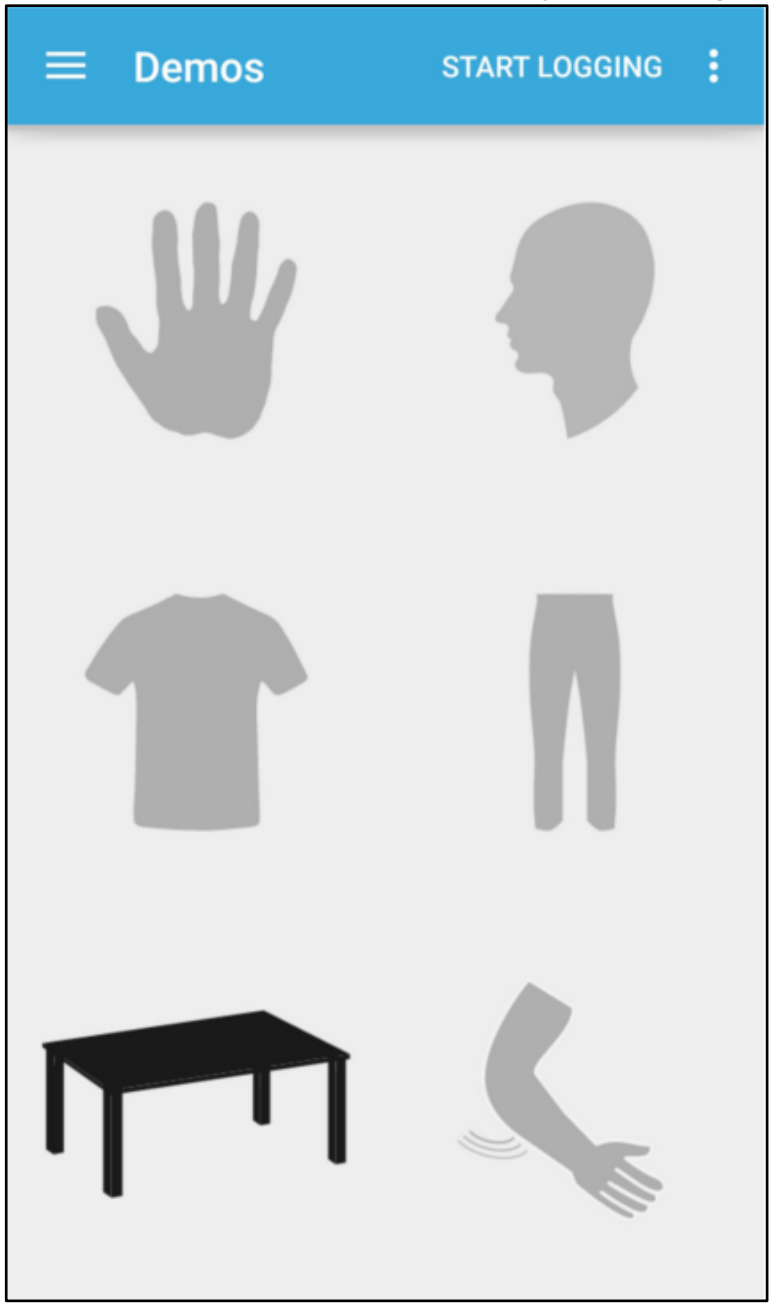
- Stationary
- Walking
- Fast walking
- Jogging
- Biking
- Driving

Figure 21: BlueMS (Android version) osxMotionAR activity recognition page

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

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

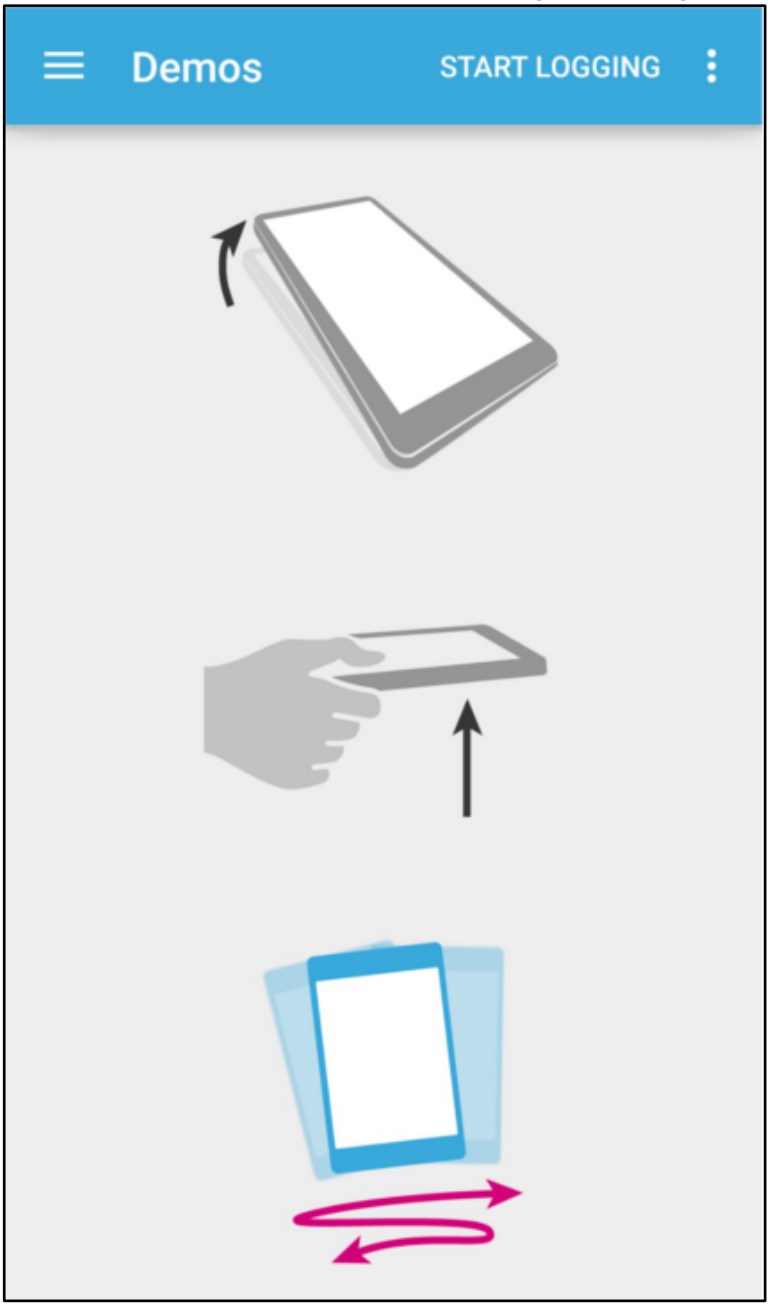
Figure 22: BlueMS (Android version) osxMotionCP carry position recognition page



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

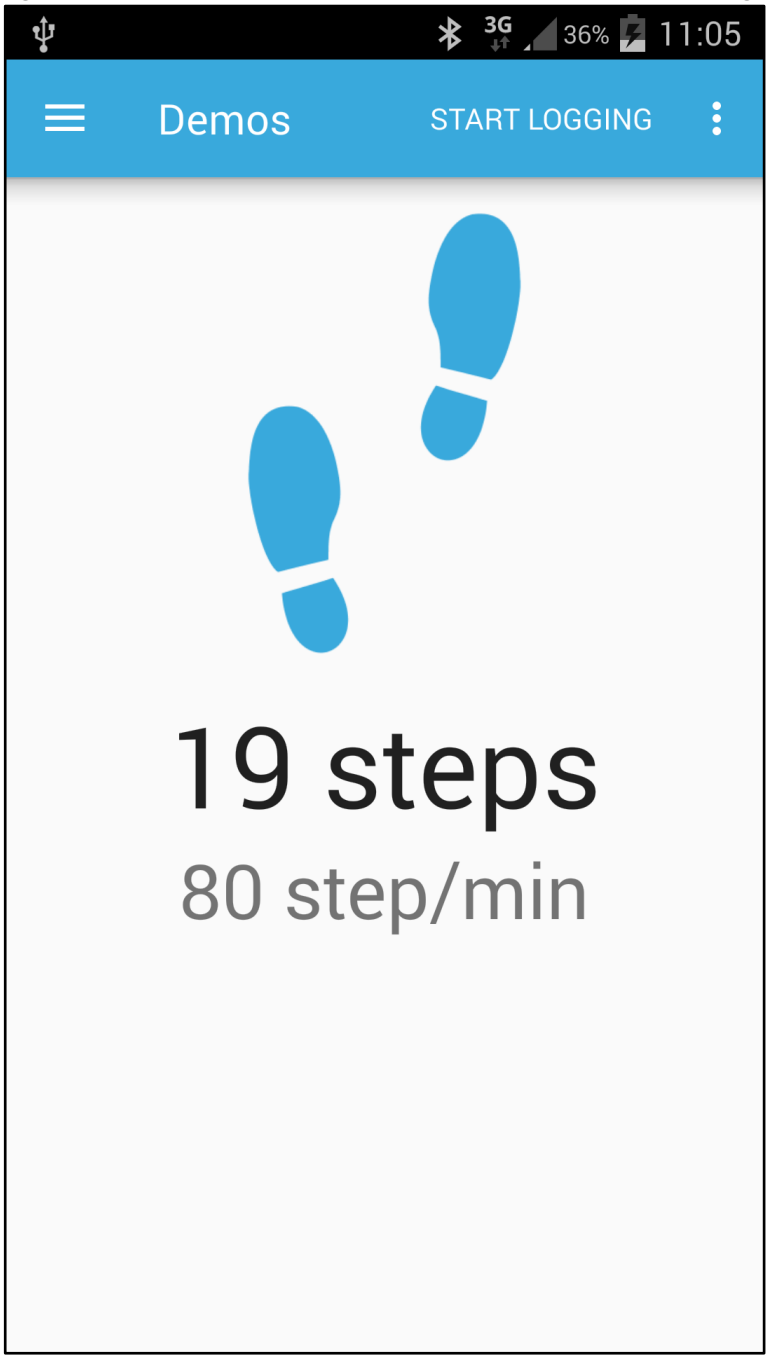
- pick up
- glance
- wake up in hand

Figure 23: BlueMS (Android version) osxMotionGR gesture recognition page



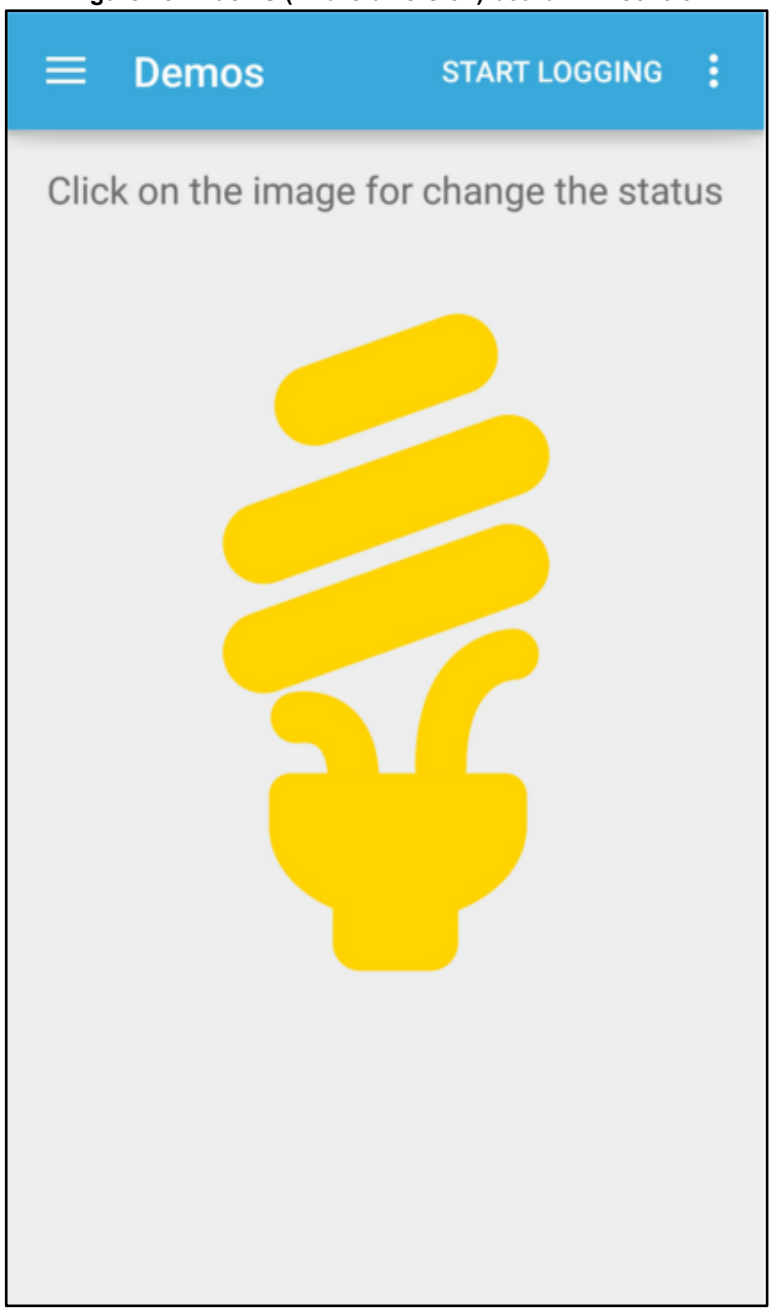
If the osxMotionPM algorithm is the page shown below is available, with pedometer information:

Figure 24: BlueMS (Android version) osxMotionPM pedometer page



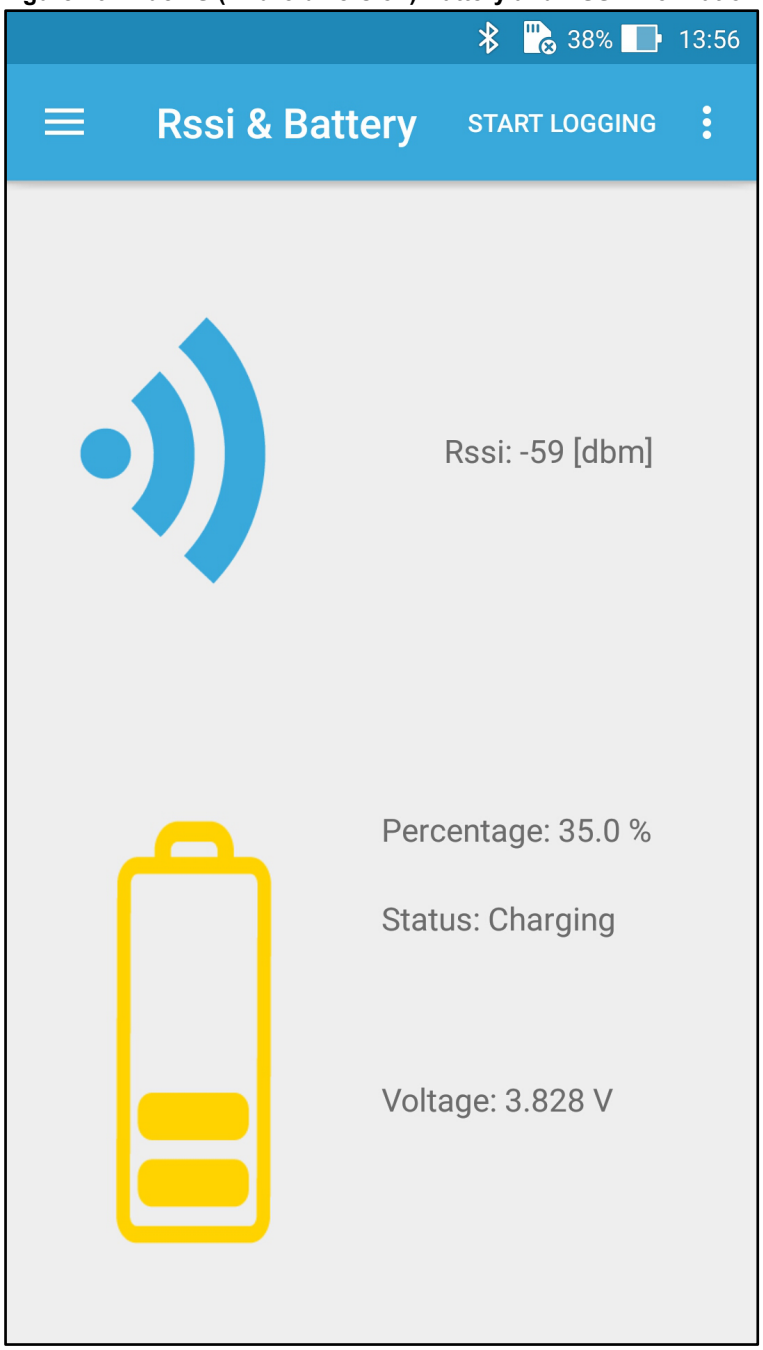
The following page shows the LED on/off control.

Figure 25: BlueMS (Android version) board LED control



The last indicates the RSSI of the Bluetooth signal and, for STEVAL-STLKT01V1 if the battery is connected, the charge percentage, measured voltage and battery status (charging/discharging/low battery).

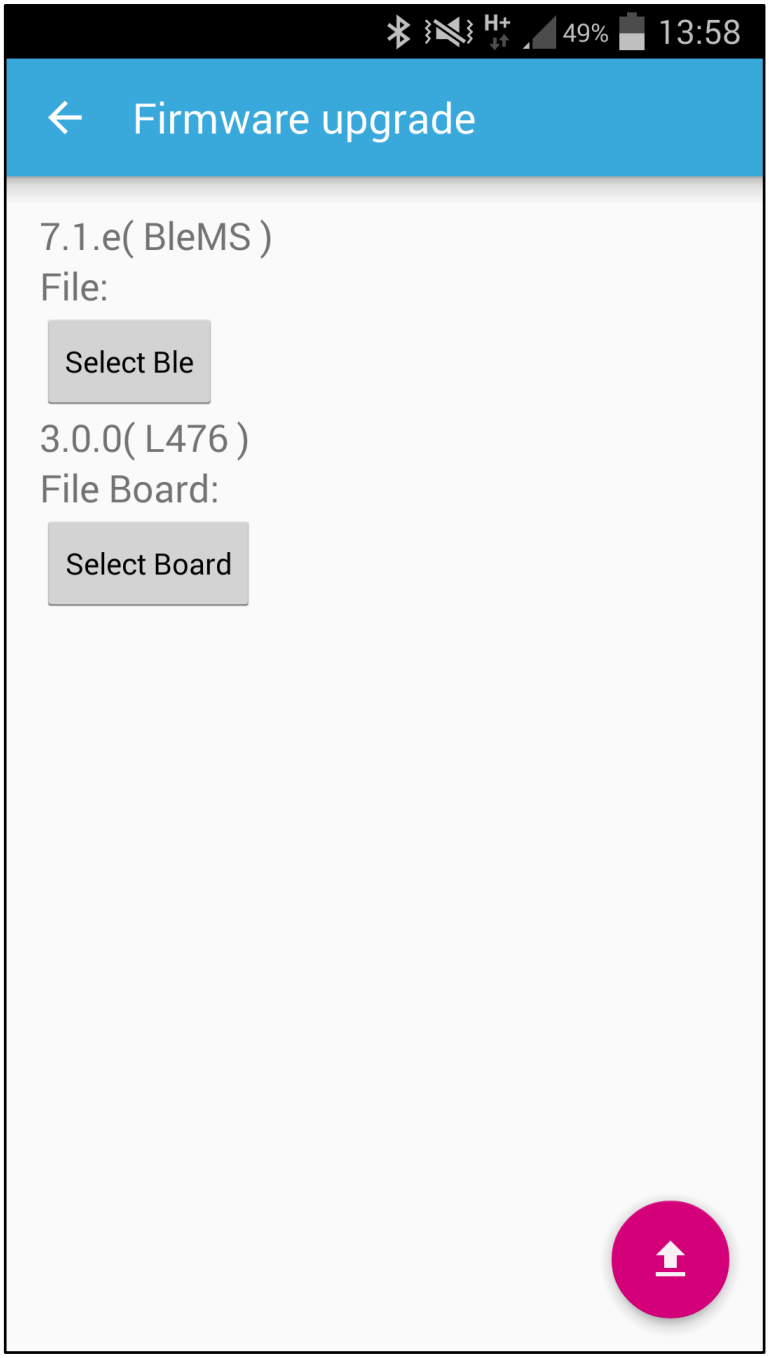
Figure 26: BlueMS (Android version) Battery and RSSI information



1.12 Firmware-Over-The-Air update with BlueMS

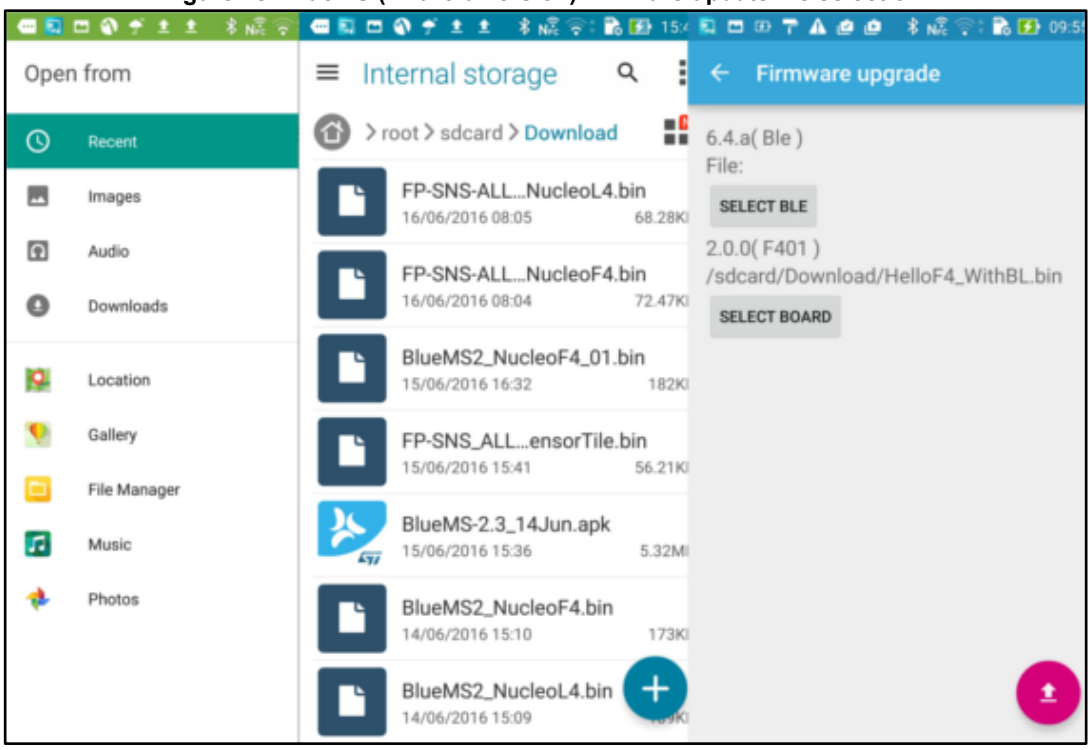
If the 'Firmware upgrade' menu option is available on the BlueMS main application page (*Figure 12: "BlueMS (Android version) main page following BLE connection"*), the page below appears.

Figure 27: BlueMS (Android version) firmware upgrade page



The BlueMS application shows which version of the BLUEMICROSYSTEM1 software is running and the board type. To apply an update, press the red button and select the appropriate update file.

Figure 28: BlueMS (Android version) firmware update file selection



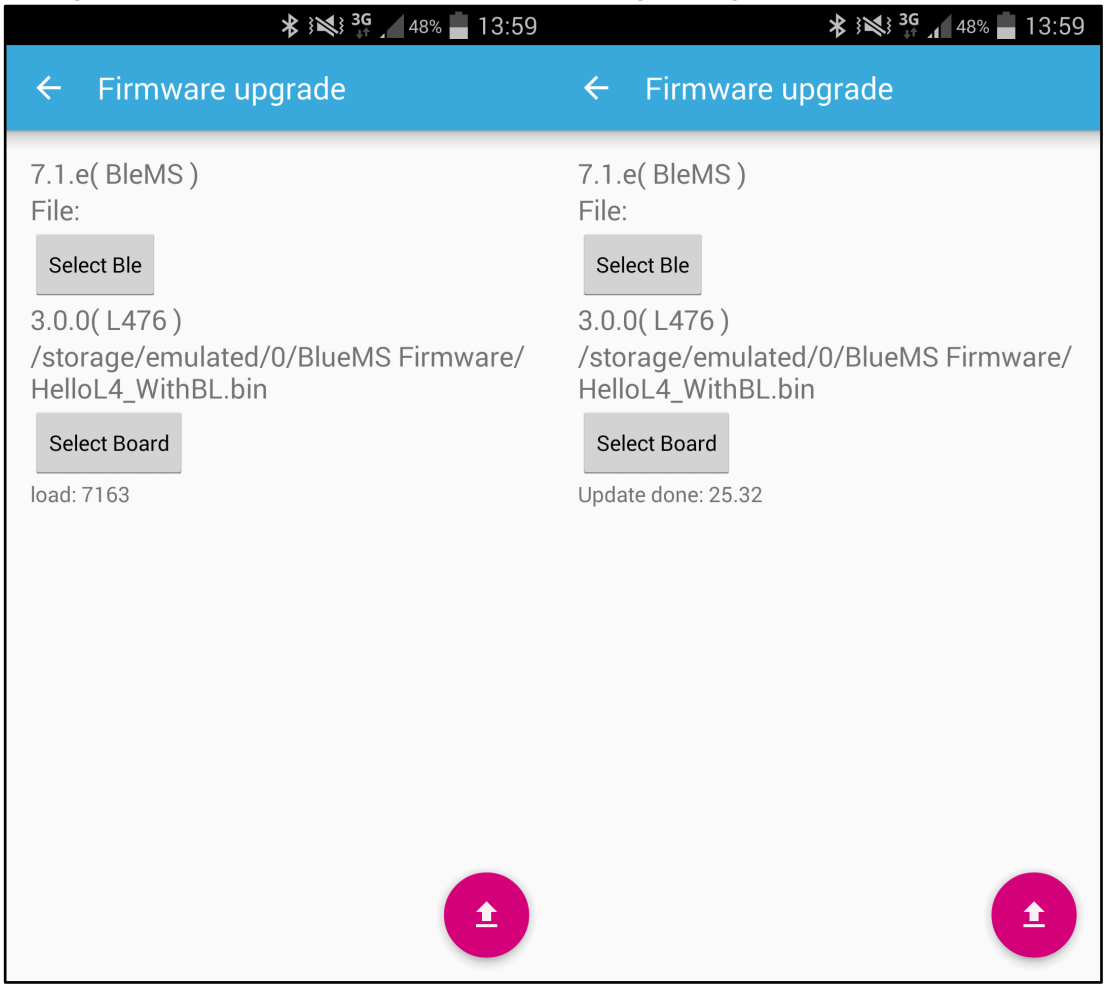
BlueMS sends the BLUEMICROSYSTEM1 a command communicating that it is going to send an update of a certain byte size and corresponding CRC value.

The figure below shows the terminal window with the debug information returned during FOTA for an STM32 Nucleo platform when we use a UART to control BLUEMICROSYSTEM1 behavior.

Figure 29: Terminal window information during FOTA

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

Figure 30: BlueMS (Android version) application page during FOTA and on completion



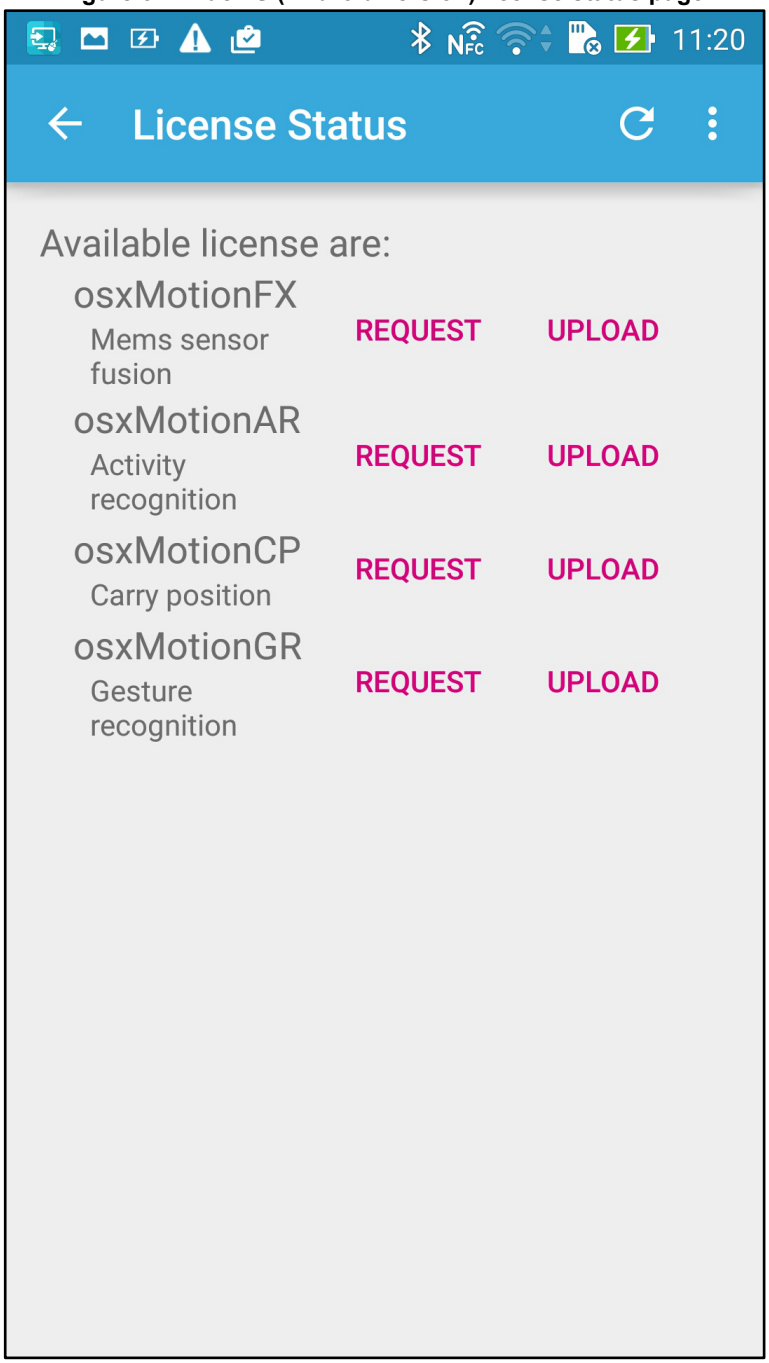
1.13 Open.MEMS license activation

If `OSX_BMS_LICENSE_H_FILE` in Inc\osx_bms_config.h is not defined, the Open.MEMS licenses must be requested and enabled through the BlueMS android/iOS application.

If no Open.MEMS licenses are active, BLUEMICROSYSTEM1 can still read and transmit sensor data values to the Android/iOS BlueMS application.

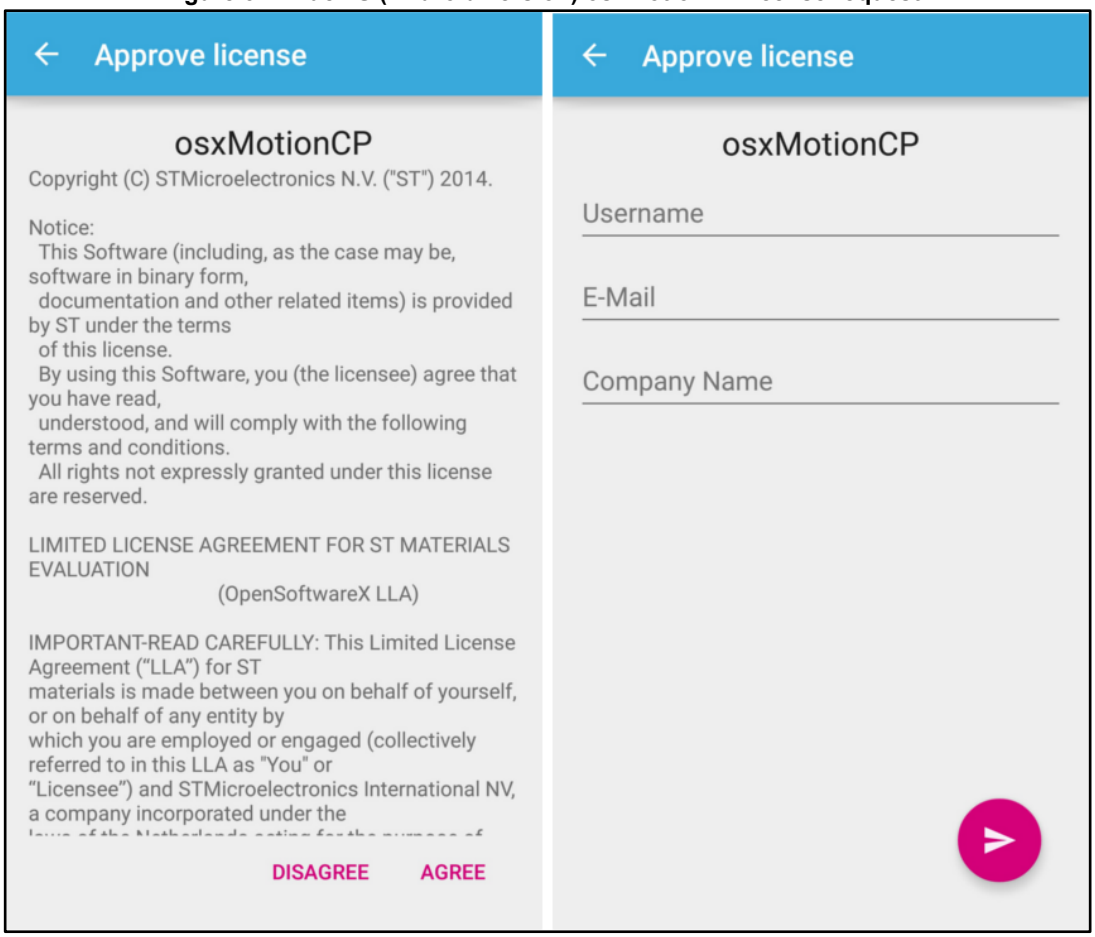
If 'License Manager' is chosen from the options menu in the BlueMS start page, the page below appears.

Figure 31: BlueMS (Android version) license status page



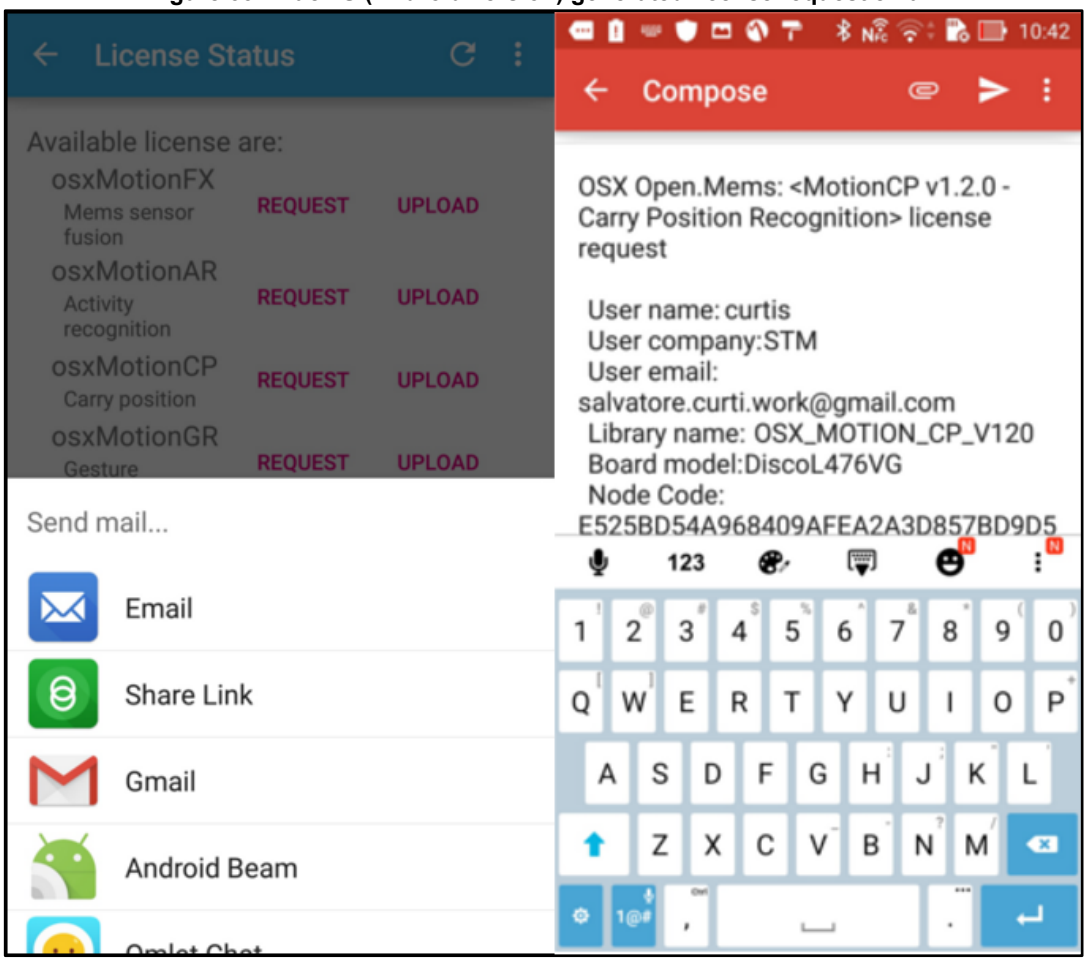
The REQUEST button opens the legal agreement page, which continues to the license request page if you accept.

Figure 32: BlueMS (Android version) osxMotionCP license request



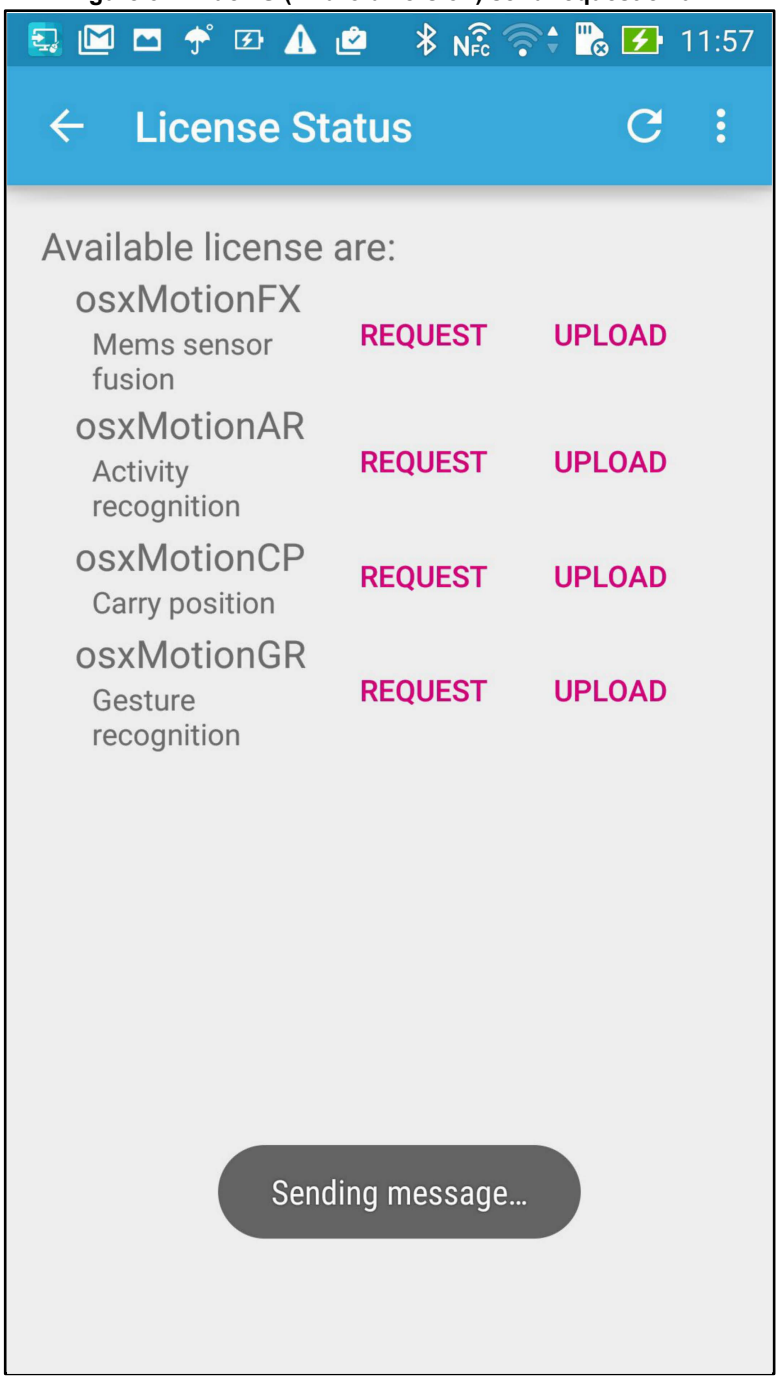
Enter a username, valid email address and company name, and click the red arrow button to generate the email request.

Figure 33: BlueMS (Android version) generated license request email



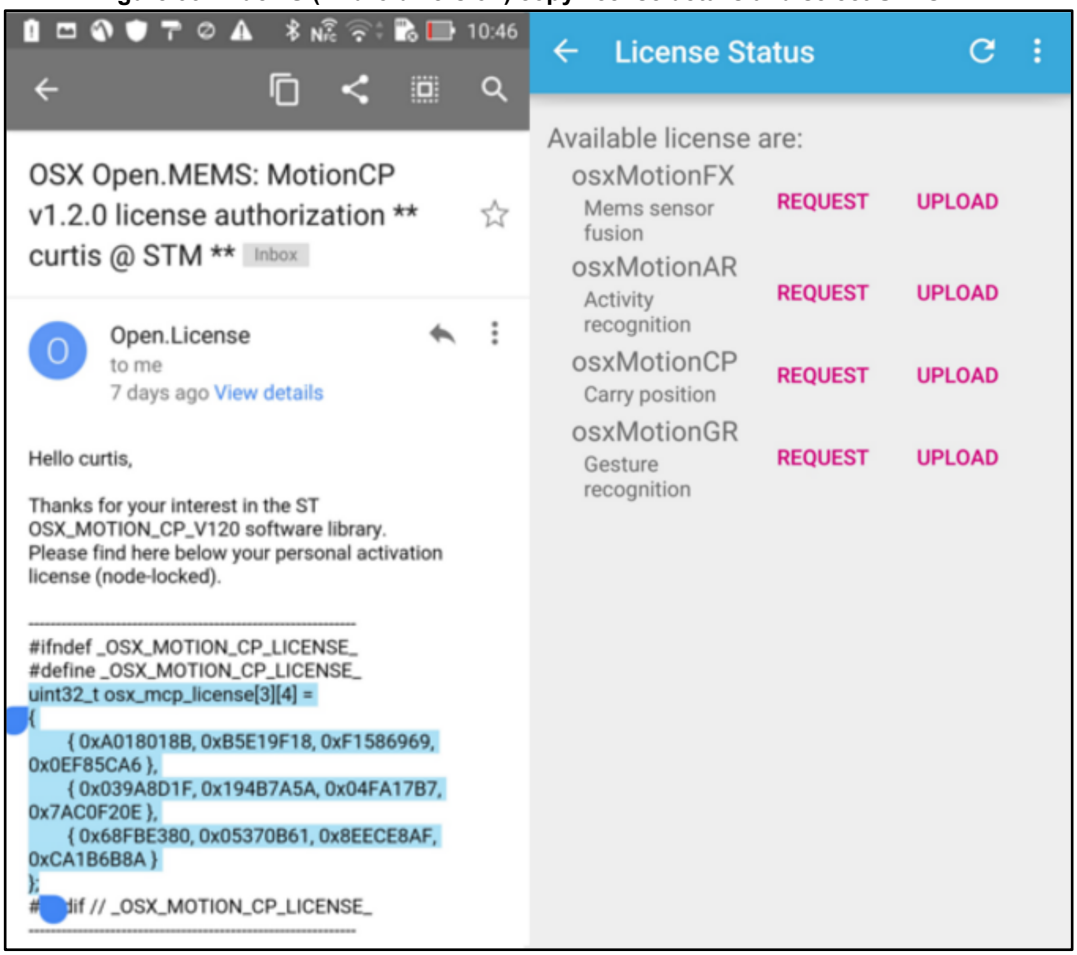
The following page is shown when the email has been sent.

Figure 34: BlueMS (Android version) send request email



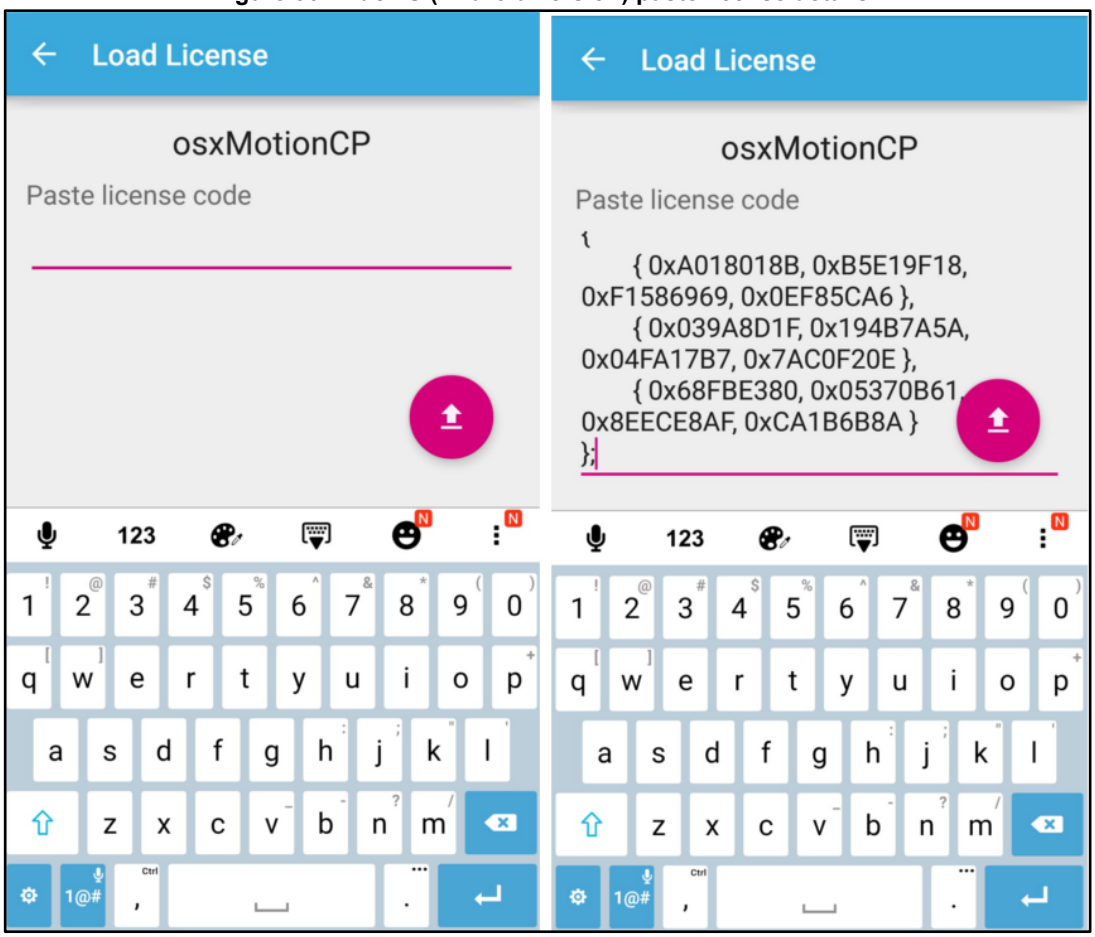
If you receive an email response with license details, select and copy the details as shown below and press UPLOAD in the License Status page.

Figure 35: BlueMS (Android version) copy license details and select UPLOAD



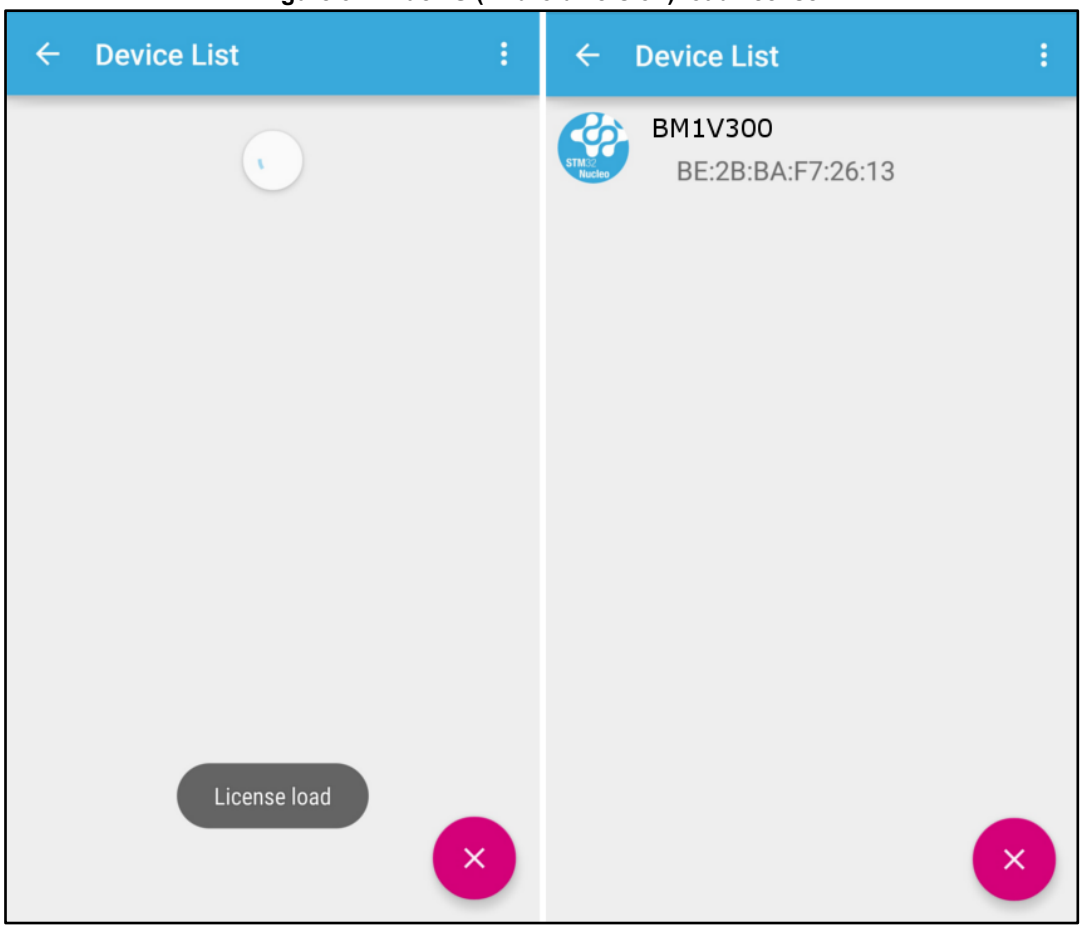
Paste the license code in the Load License page and click the red upload button.

Figure 36: BlueMS (Android version) paste license details



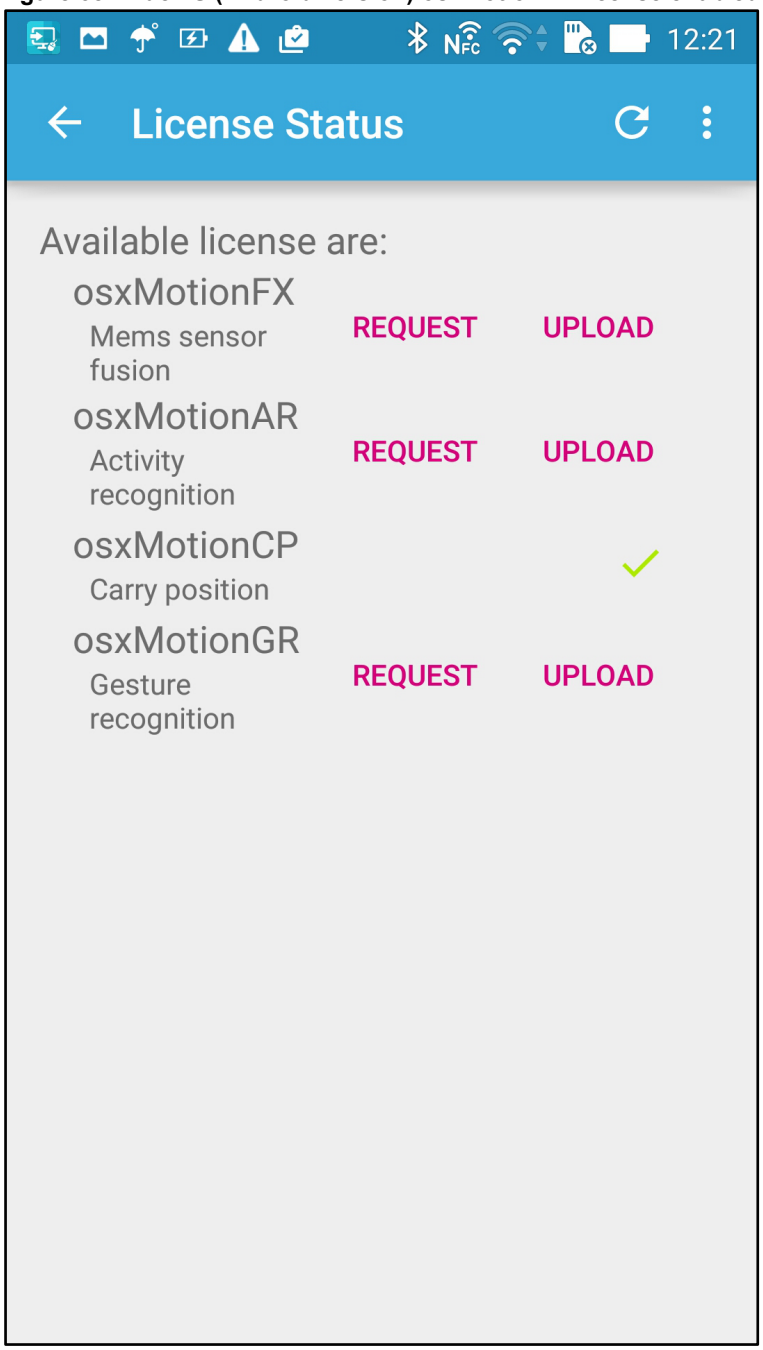
Once the correct license code is loaded, you can proceed to reconnect the board.

Figure 37: BlueMS (Android version) load license



A green tick mark indicates license acknowledgment.

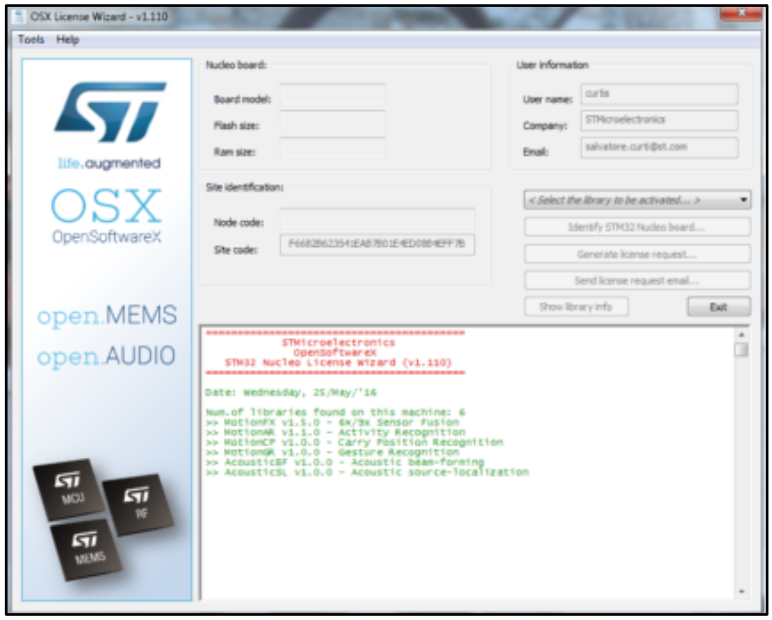
Figure 38: BlueMS (Android version) osxMotionCP license enabled



1.13.1 Open.MEMS license activation with osxLicenseWizard

If `OSX_BMS_LICENSE_H_FILE` in Inc\osx_bms_config.h is defined, the Open.MEMS licenses must be requested and enabled through the osxLicenseWizard.

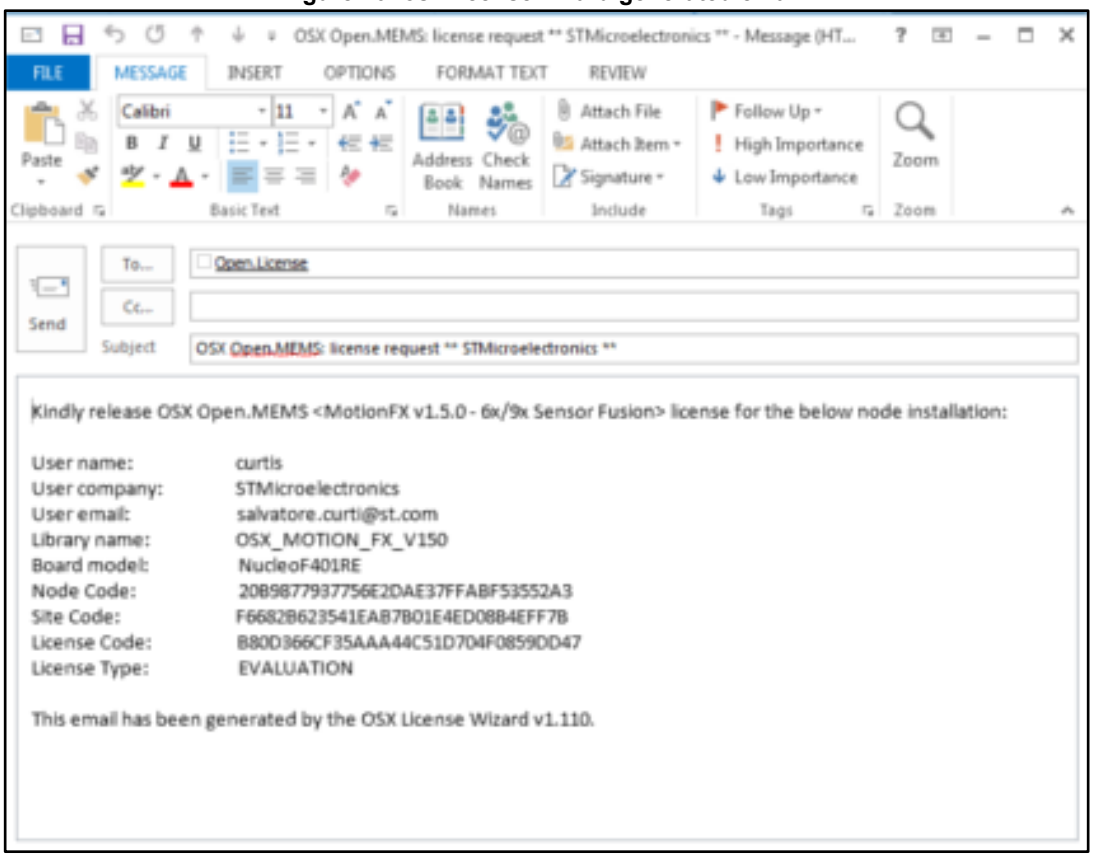
Figure 39: osxLicenseWizard



For each license:

1. Open osxLicenseWizard
2. Selected the library to be activated (osxMotionFX, osxMotionAR, osxMotionCP or osxMotionGR)
3. Click 'identify STM32 Nucleo board...'
4. Click 'generate license request...' and accept the terms in the license agreement
5. Click 'send license request email...'
6. Send the email

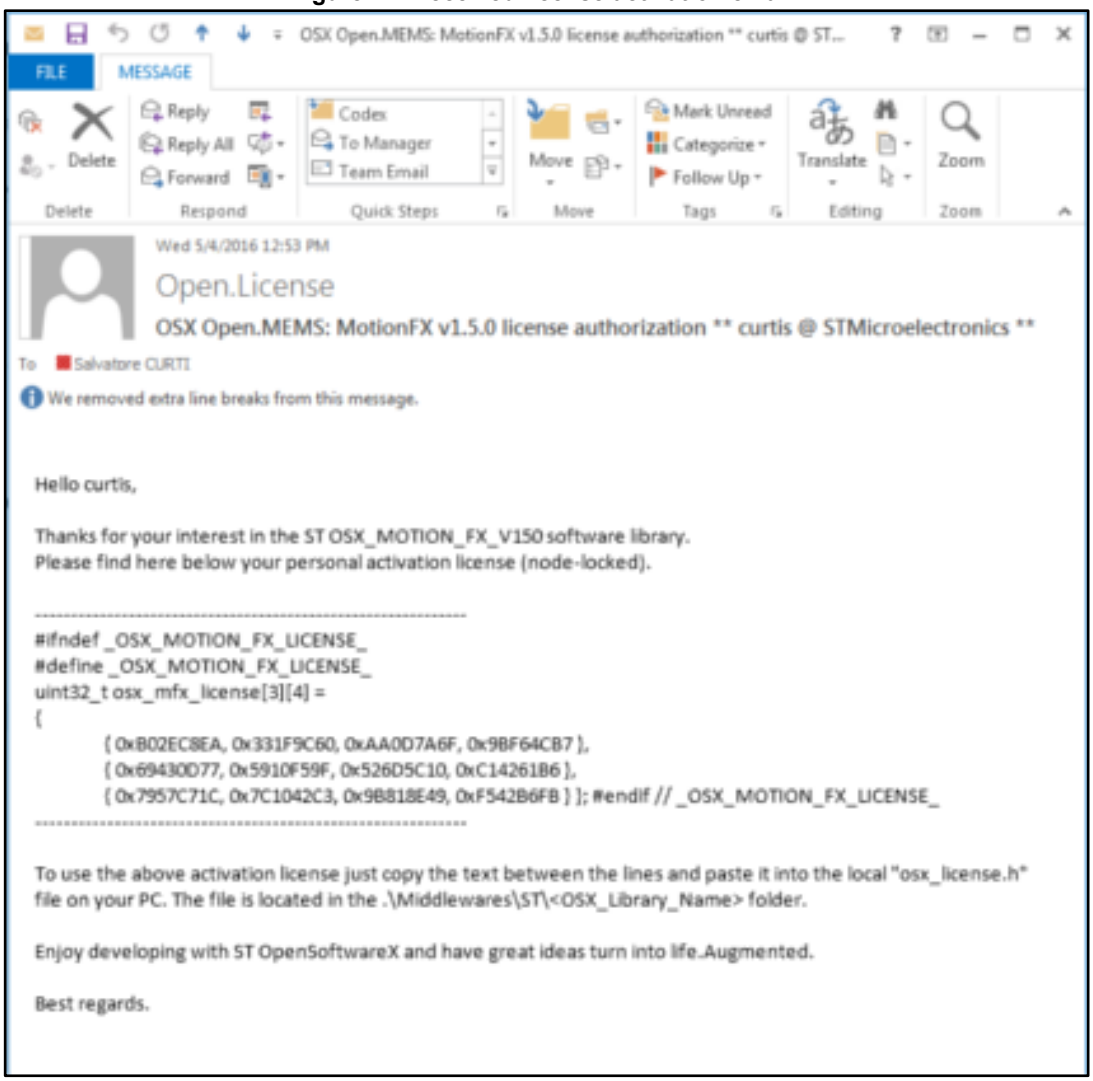
Figure 40: osxLicenseWizard generated email



Open the received email and place the license details in the respective osx_license.h files in the Middlewares directory. To enable the osxMotionFX license, for example, place the code in Middlewares\ST\STM32_OSX_MotionFX_Library\osx_license.h file.

You must then compile the BLUEMICROSYSTEM1 to enable any new licenses.

Figure 41: Received license activation email



2 System setup guide

2.1 Hardware description

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

2.1.1 STM32 Nucleo platform

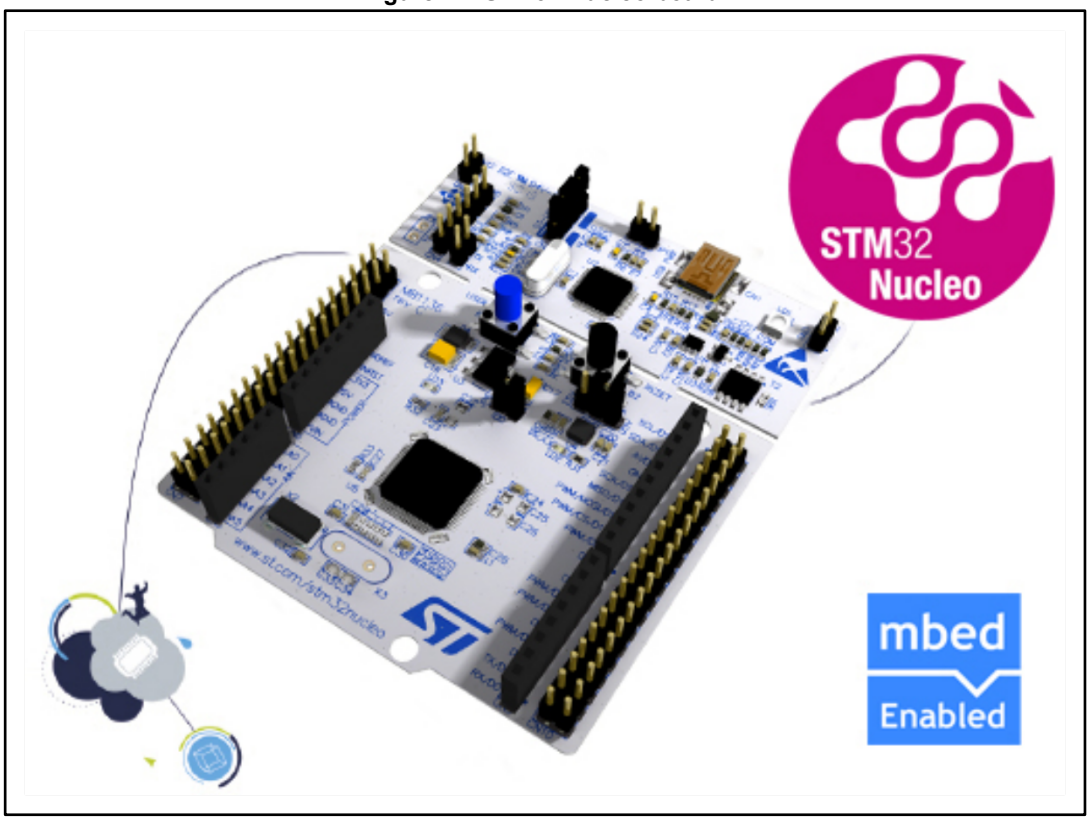
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

The Arduino™ connectivity support and ST morpho headers 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 42: STM32 Nucleo board

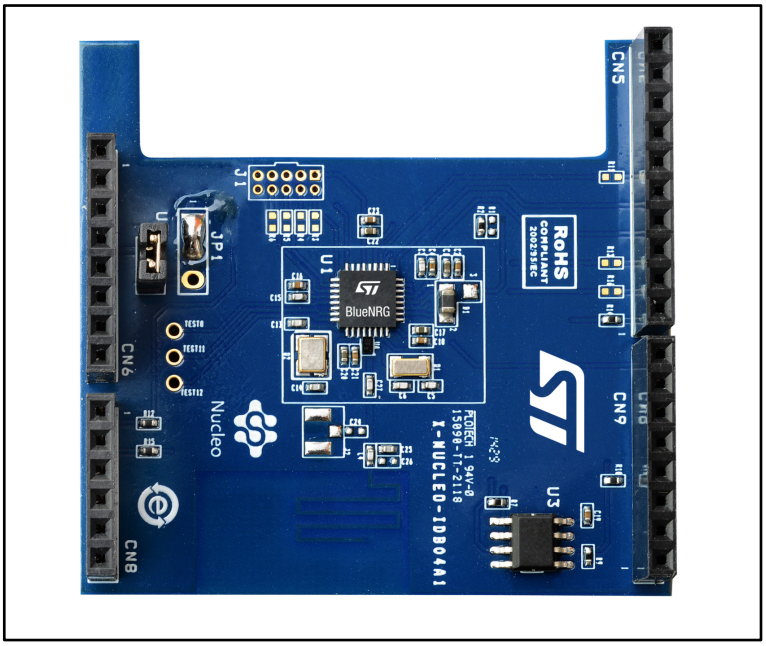


Information regarding the STM32 Nucleo board is available at *www.st.com/stm32nucleo*

2.1.2 X-NUCLEO-IDB04A1 expansion board

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

Figure 43: X-NUCLEO-IDB04A1 expansion board

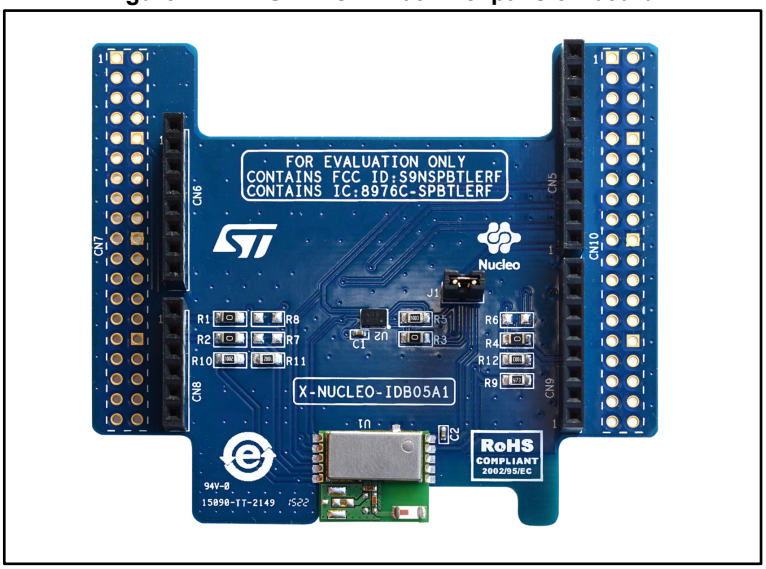


Information regarding the X-NUCLEO-IDB04A1 expansion board is available on www.st.com at *http://www.st.com/x-nucleo*.

2.1.3 X-NUCLEO-IDB05A1 expansion board

The X-NUCLEO-IDB05A1 is a Bluetooth low energy expansion board based on the SPBTLE-RF BlueNRG-MS RF module to allow expansion of the STM32 Nucleo boards. The SPBTLE-RF module is FCC (FCC ID: S9NSPBTLERF) and IC certified (IC: 8976C-SPBTLERF). The BlueNRG-MS is a very low power Bluetooth low energy (BLE) single-mode network processor, compliant with Bluetooth specification v4.2. X-NUCLEO-IDB05A1 is compatible with the ST morpho and Arduino™ UNO R3 connector layout. This expansion board can be plugged into the Arduino UNO R3 connectors of any STM32 Nucleo board.

Figure 44: X-NUCLEO-IDB05A1 expansion board

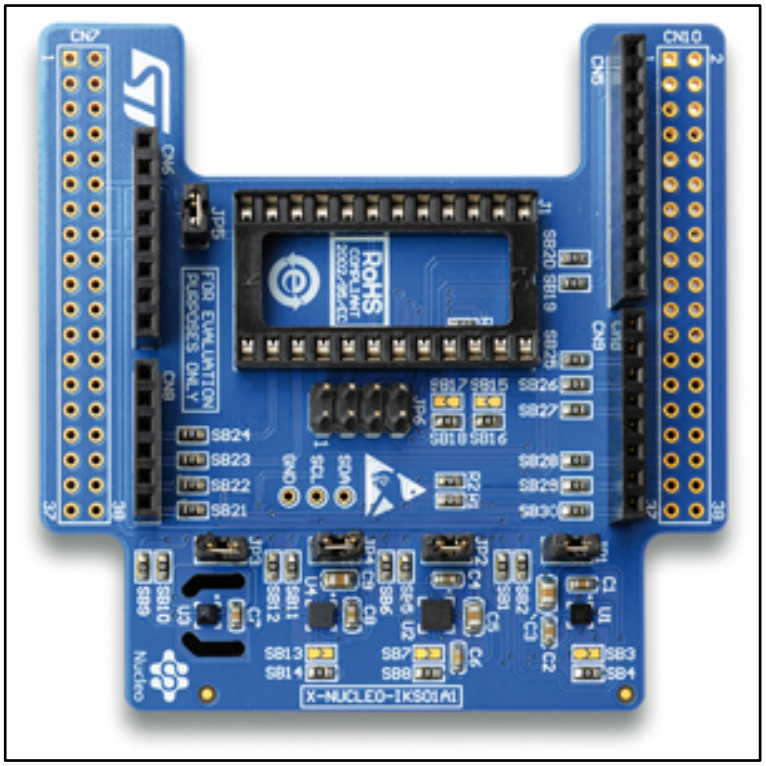


Information about the X-NUCLEO-IDB05A1 expansion board is available on www.st.com at *http://www.st.com/x-nucleo*.

2.1.4 X-NUCLEO-IKS01A1 expansion board

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

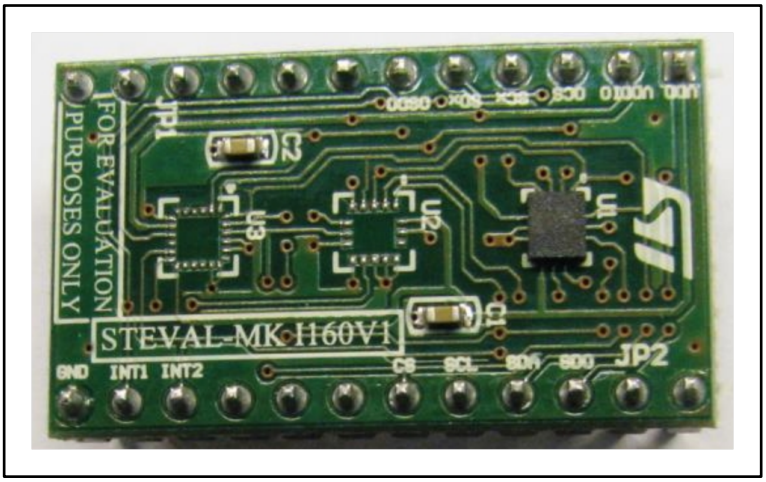
Figure 45: X-NUCLEO-IKS01A1 expansion board



Information about the X-NUCLEO-IKS01A1 expansion board is available on www.st.com at: *http://www.st.com/x-nucleo*.

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

Figure 46: LSM6DS3 DIL24 adapter board



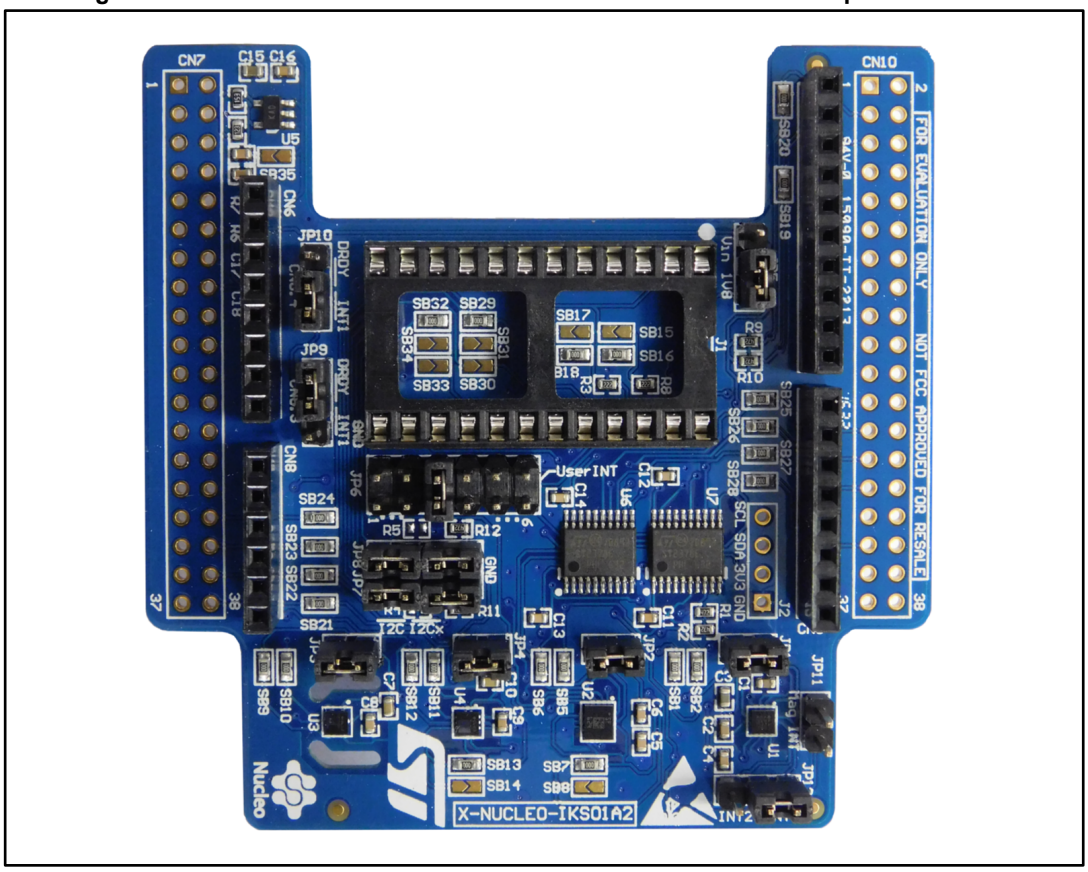
2.1.5 X-NUCLEO-IKS01A2 expansion board

The X-NUCLEO-IKS01A2 is a motion MEMS and environmental sensor expansion board for the STM32 Nucleo.

It is equipped with 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 47: X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board



2.1.6 STEVAL-STLKT01V1 SensorTile development kit

2.1.6.1 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 App.

The SensorTile is a tiny, square-shaped IoT module that packs powerful processing capabilities leveraging an 80 MHz STM32L476JGY microcontroller and Bluetooth low energy connectivity based on BlueNRG network processor as well as a wide spectrum of motion and environmental MEMS sensors, including a digital microphone.

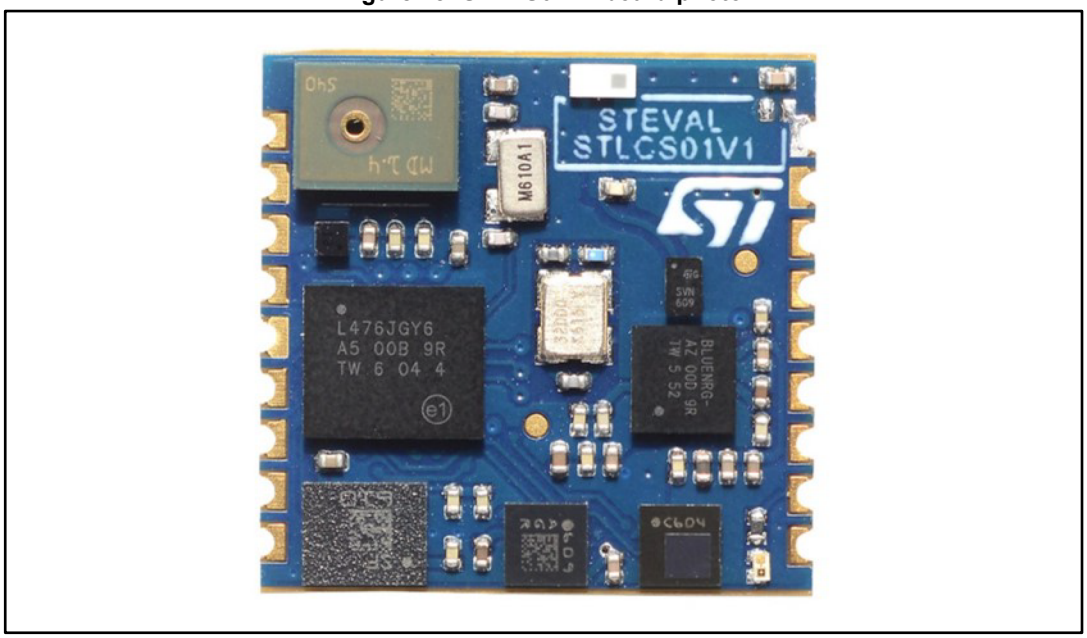
SensorTile can fit snugly in your IoT hub or sensor network node and become the core of your solution.

2.1.6.2 Features

- FCC (ID: S9NSTILE01) and IC (IC: 8976C-STILE01) certified
- Included in the development kit package:
  - SensorTile module
  - 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 for housing the SensorTile cradle and the battery
  - 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. It includes low level drivers for all the on-board devices
  - BLUEMICROSYSTEM1 and BLUEMICROSYSTEM2: STM32Cube expansion software package, supporting different algorithms tailored to the on-board sensors
  - FP-SNS-ALLMEMS1 and FP-SNS-MOTENV1: STM32 ODE functional packs
  - ST BlueMS: iOS and Android demo Apps
  - BlueST-SDK: iOS and Android Software Development Kit
  - Compatible with STM32 ecosystem through STM32Cube support

2.1.6.3 Boards included in the kit

Figure 48: 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 BLUEMICROSYSTEM1 and BLUEMICROSYSTEM2 software expansion package for STM32Cube and the STM32 ODE functional pack FP-SNS-ALLMEMS1
- 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.

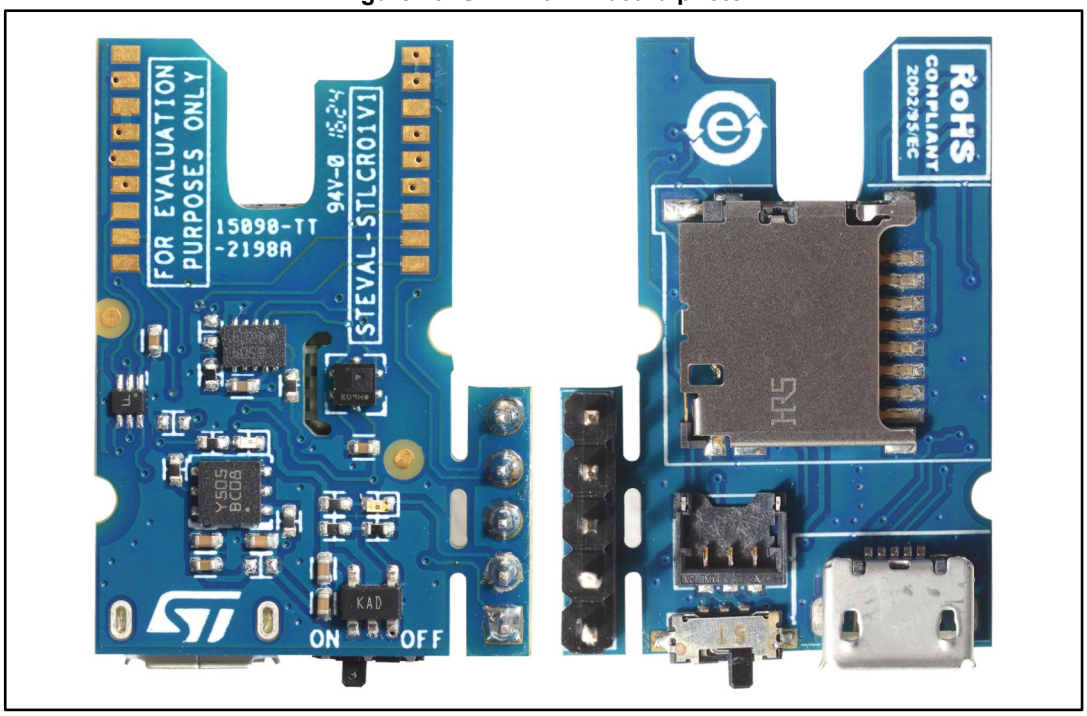
The SensorTile comes in a very small square shape 13.5 x 13.5 mm. All the electronic components are on the top side of the pcb, while the bottom side has a small connector through which it is possible to easily plug and unplug it from a motherboard. The connector pinout is also replicated on 18 pcb pads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

The module comes with pre-loaded 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.

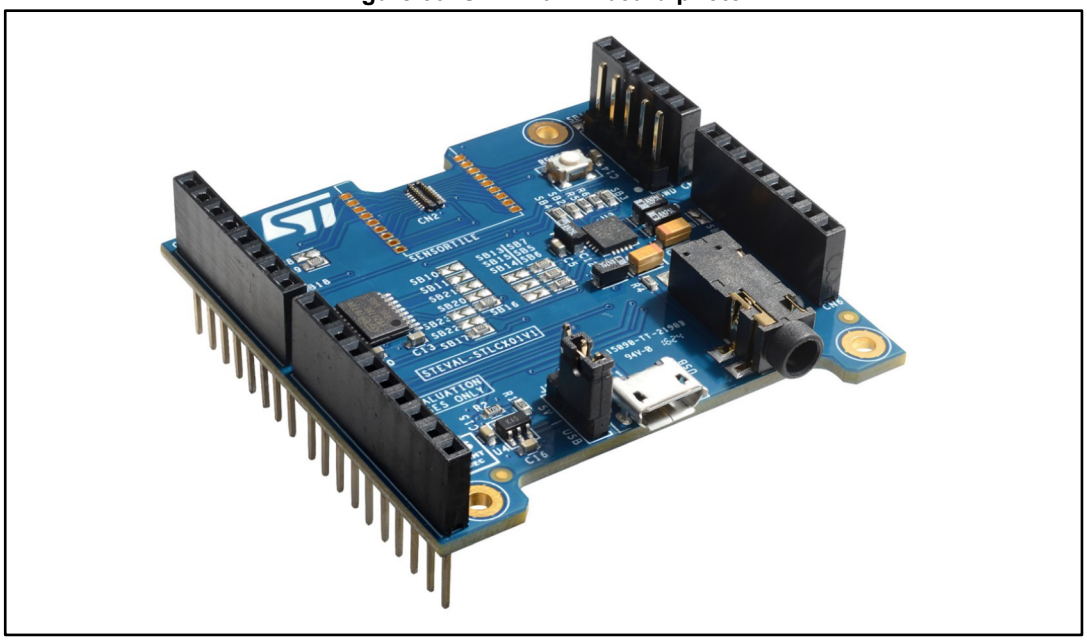
Figure 49: STLCR01V1 board photo



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 50: 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

2.2 Software description

The following software components are needed in order to set up a suitable development environment for creating applications with the STM32 Nucleo equipped with the sensors, microphones and Bluetooth low energy expansion boards and the STEVAL-STLKT01V1:

- BLUEMICROSYSTEM1: Bluetooth low energy and sensors software for STM32Cube. BLUEMICROSYSTEM1 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

2.3 Hardware and software setup

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

2.3.1 Hardware setup

The following hardware components are needed:

- for STM32 Nucleo expansion boards:
  - One STM32 Nucleo development platform (order code: NUCLEO-F401RE or NUCLEO-L476RG)
  - One sensor expansion board (order code: X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2)
  - One BlueNRG Bluetooth low energy expansion board (order code: X- NUCLEO-IDB04A1 or 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 on 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

2.3.2 Software setup

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

2.3.2.1 Development tool-chains and compilers

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.

2.3.3 System setup guide

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

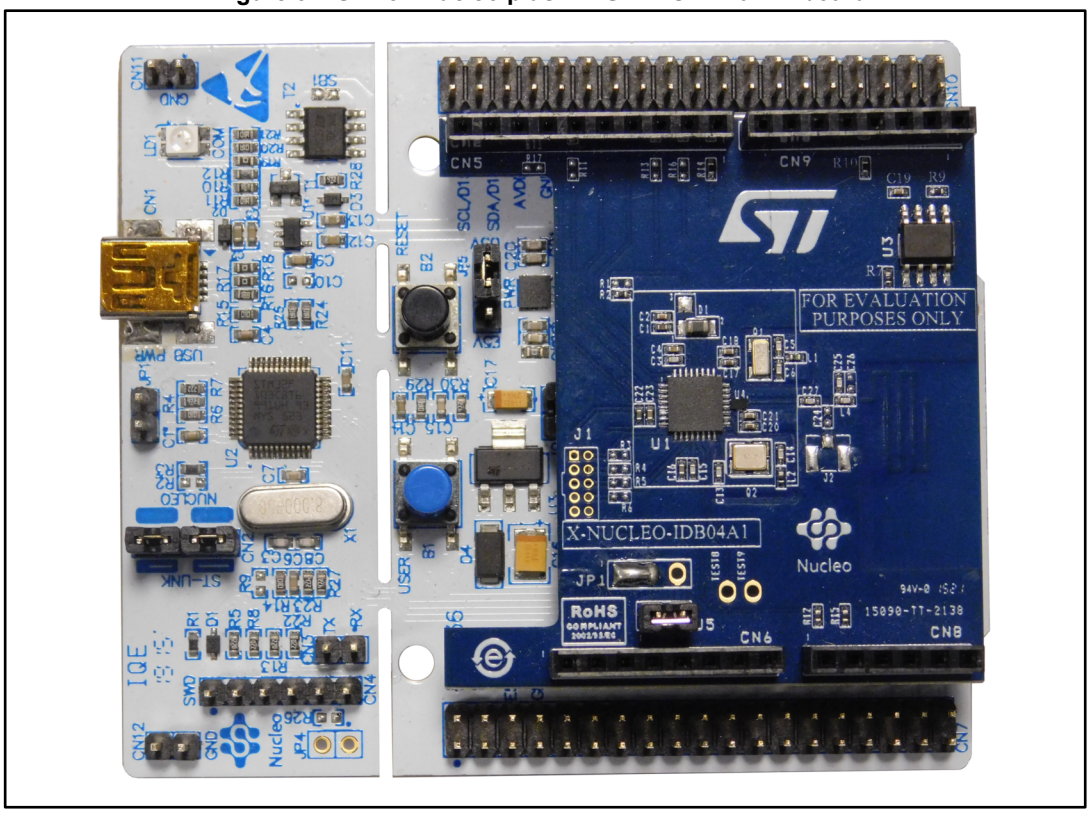
- on the STM32 Nucleo board with the expansion boards
- on STEVAL-STLKT01V1 development kit

2.3.3.1 STM32 Nucleo and expansion boards setup

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. The developer can download the relevant version of the ST-LINK/V2-1 USB driver by searching STSW-LINK008 or STSW-LINK009 on www.st.com.

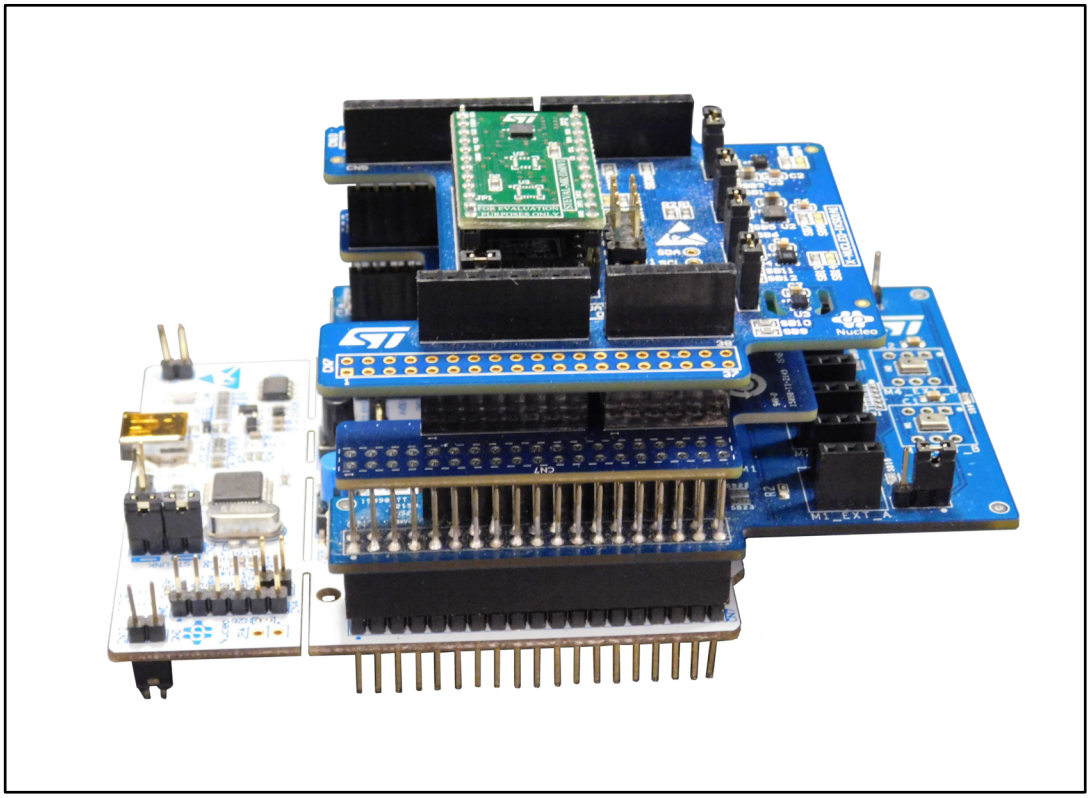
The X-NUCLEO-IDB04A1 or X-NUCLEO-IDB05A1 BlueNRG BLE expansion board is easily connected to the STM32 Nucleo board through the Arduino UNO R3 extension connector, as shown below.

Figure 51: STM32 Nucleo plus X-NUCLEO-IDB04A1 board



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

Figure 52: STM32 Nucleo plus X-NUCLEO-IDB05A1 plus X-NUCLEO-IKS01A1 plus LSM6DS3 DIL24 boards



The above stacking sequence optimizes the performance of the SPBTLE-RF module on the X-NUCLEO-IDB05A1 expansion board and reduces interference from its antenna

2.3.3.2 STEVAL-STLKT01V1 setup

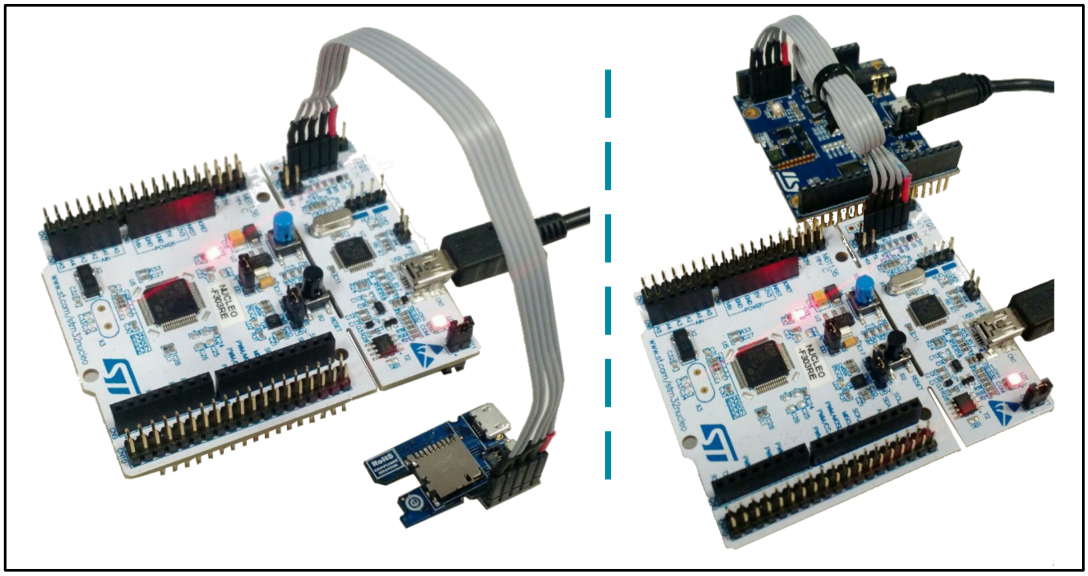
The ST-LINK/V2-1 debugger/programmer integrated on STM32 Nucleo board must be used to program the STEVAL-STLCS01V1 (SensorTile). The developer can download the relevant version of the ST-LINK/V2-1 USB driver by searching STSW-LINK008 or STSW-LINK009 on www.st.com.

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 53: SensorTile Cradle expansion board and Sensor Tile Cradle board connected to ST-LINK/V2-1



3 Revision history

Table 1: Document revision history

Date	Revision	Changes
04-Mar-2015	1	Initial release.
29-Jul-2015	2	Updated Section 1.6.
13-Nov-2015	3	Throughout document: - minor text and image edits - added reference to osxMotionAR activity and osxMotionCP carry position recognition software - all references to X-NUCLEO-IDB04A1 changed to X-NUCLEO-IDB04A1/X-NUCLEO-IDB05A1 Updated Section 1.5: "Sample application description" Updated Section 1.6: "Android and iOS sample client application" Added Section 2.1.1: "X-NUCLEO-IDB04A1 expansion board" Updated Section 2.3.5: "STM32 Nucleo and Sensor expansion boards setup"
05-Jul-2016	4	Text and formatting changes throughout document Added STEVAL-STLKT01V1 board use details Added Section 1.4: "Flash management" Added Section 1.5: "The boot process" Added Section 1.6: "Firmware-Over-The-Air (FOTA) update" Added Section 1.9: "Open.MEMS licenses" Added Section 1.11: "Firmware-Over-The-Air update with BlueMS" Added Section 1.12: "Open.MEMS license activation" Added Section 2.1.5: "STEVAL-STLKT01V1 SensorTile development kit"
27-Sep-2016	5	Throughout document: - text and formatting changes - Added serial console support for STEVAL-STLKT01V1 - Added the osxMotionPM pedometer algorithm information - Added Gas Gauge for STEVAL-STLKT01V1 information Added Section 1.6: "The installation process
06-Dec-2016	6	Updated *Section 1.1: "Overview"*, *Section 1.3: "Folder structure"*, *Section 1.4: "Flash management"*, *Figure 1: "BLUEMICROSYSTEM1 overall system architecture"*, *Figure 3: "BLUEMICROSYSTEM1 Flash structure"* and *Section 1.9: "Sample application description"* Added *Section 2.1.5: "X-NUCLEO-IKS01A2 expansion board"* Throughout document: - Added X-NUCLEO-IKS01A2 compatibility - Added osxMotionID and e-Compass references - License Manager replaced by Meta Data Manager

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2016 STMicroelectronics – All rights reserved