Getting started with the STM32Cube function pack for IoT node with NFC, BLE connectivity and environmental, motion and Time-of-Flight sensors

Introduction

The FP-SNS-FLIGHT1 is an STM32Cube function pack which let your IoT node connect to a smartphone via BLE and uses a suitable Android™ or iOS™ application like the BlueMS app to view real-time environmental sensor data, motion sensor data and proximity sensor data.

The package also enables advanced functionalities such as sensor data fusion, accelerometer-based real-time activity recognition and real-time hand gesture detection. It uses the NDEF standard for simple and secure Bluetooth pairing, storing the necessary information on the NFC tag, thus simplifying the device configuration. This package, together with the suggested combination of the STM32 and ST devices, can be used to develop wearable applications, or smart things applications in general.

The software runs on the STM32 microcontroller and includes all the necessary drivers to recognize the devices on the STM32 Nucleo development board and expansion boards as well as on the STEVAL-BCNKT01V1 evaluation board.

1 Acronyms and abbreviations

Table 1. Acronyms and abbreviations

Acronym	Description
BLE	Bluetooth low energy
NFC	Near field communication
NDEF	NFC data exchange format

2 FP-SNS-FLIGHT1 software description

2.1 Overview

The key features of the FP-SNS-FLIGHT1 package are:

- Complete firmware to develop an IoT node with NFC, BLE connectivity and environmental, motion and time-of-flight sensors
- Middleware libraries for sensor data fusion, accelerometer-based real-time activity recognition and hand gesture detection
- Compatible with BlueMS application for Android/iOS, to perform sensor data reading, motion algorithm features demo, proximity-based hand gesture detection demo and firmware update (FOTA)
- Sample implementation available for STEVAL-BCNKT01V1 evaluation board and for X-NUCLEO-NFC04A1, X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A1), X-NUCLEO-53L0A1 and X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1) connected to a NUCLEO-F401RE or NUCLEO-L476RG
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

Note: For STEVAL-BCNKT01V1 evaluation boards, the firmware package runs if the BlueCoin module (STEVAL-BCNCS01V1) is plugged to the Coin Station board (STEVAL-BCNST01V1) including two Time-of-Flight (ToF) ranging sensors (VL53L0X).

Figure 1. BlueCoin module (STEVAL-BCNCS01V1) plugged to Coin Station board (STEVAL-BCNST01V1)

This software creates the following Bluetooth services:

- the first service exposes all the hardware features and contains the following characteristics:
  - temperature
  - pressure
  - humidity
  - proximity
  - 3D gyroscope, 3D magnetometer, 3D accelerometer
- the second service exposes all the software features and contains the following characteristics:
  - quaternions generated by the MotionFX library in short precision
  - recognized activity using the MotionAR algorithm

  - recognized carry position using the MotionCP algorithm
  - recognized gesture using the MotionGR algorithm
  - the hand gesture recognized by the gesture detection middleware
- the third is the Console service that includes two characteristics:
  - stdin/stdout with bi-directional communication between client and server
  - stderr for a mono-directional channel from the STM32 Nucleo board to an Android/iOS device
- the last service is for transmitting/resetting the calibration status.

This software gathers:

- the temperature, humidity, pressure, distance and motion sensor drivers for the HTS221, LPS25H, VL53L0A1 , LSM6DS0 (or LSM6DS3) and LIS3MDL devices for STM32 Nucleo expansion boards with X-NUCLEO-IKS01A1.
- the temperature, humidity, pressure, distance and motion sensor drivers for the HTS221, LPS22H, VL53L0A1, LSM6DSL and LSM303AGR devices for STM32 Nucleo expansion boards with X-NUCLEO-IKS01A2.
- the temperature, pressure, distance and motion sensor drivers for the LPS22HB, VL53L0A1 and LSM6DSM, LSM303AGR devices for STEVAL-BCNKT01V1 running on STM32 Nucleo.

This package is compatible with the BlueMS Android/iOS (ver. 3.0.0 or higher) application, available at the respective Google Play/iTunes stores, which can be used to display information sent via the BLE

The BlueMS application allows Over-The-Air firmware updates (for X-NUCLEO-IDB05A1 Bluetooth low energy expansion board only).

2.2 Architecture

This software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a board support package (BSP) for the BlueNRG-1 sensor expansion board, proximity and ambient light sensing (for X-NUCLEO-6180XA1 only) module and the dynamic NFC tag expansion boards, and some middleware components for communication with other Bluetooth low energy devices: it also enables data exchange with an NFC-ready device using the NDEF standard.

BlueNRG is a very low power Bluetooth low energy (BLE) single-mode network processor.

The MotionFX (iNEMOEngine PRO) filtering and predictive suite uses advanced algorithms to intelligently integrate multiple MEMS sensor outputs, regardless of environmental conditions achieving an optimal performance. Real-time motion sensor data fusion is set to increase accuracy, resolution, stability and response time.

The MotionAR (iNEMOEngine PRO) real-time software acquires data from the accelerometer to recognize user activities. The software can also be combined with other human motion recognition algorithms and can significantly improve user experience in advanced motion-based applications in consumer, computer, industrial and medical fields.

The MotionCP (iNEMOEngine PRO) real-time software acquires data from the accelerometer to recognize board positions (on desk, on head, near head, shirt pocket, trousers pocket and in swinging arm).

The MotionGR (iNEMOEngine PRO) real-time software acquires data from the accelerometer and recognizes user gestures (pick up, glance and wake up).

The gesture detect software library uses the X-NUCLEO-53L0A1 (or X-NUCLEO-6180XA10) on-board sensor plus two additional satellites to detect tap and swipe (from left to right and from right to left) gestures.

Activity recognition, carry position and gesture recognition are managed through a specific software for mobile and wearable applications, and the exclusive use of the accelerometer by MotionAR, MotionCP, and MotionGR facilitates the low power consumption required in this field of application, in compliance with Bluetooth specifications core 4.0 (X-NUCLEOIDB04A1) or 4.2 (X-NUCLEO-IDB05A1) for STM32 Nucleo boards.

The drivers abstract low-level hardware details and allow the middleware components and applications to access the Dynamic NFC tag and all the other sensors in a hardware-independent manner.

The package includes a sample application that the developer can use to start experimenting with the code. The sample application was developed to enable NFC pairing and to transmit the values read from all the sensors (temperature, humidity, pressure, luminosity, proximity, accelerometer, magnetometer and gyroscope) to a Bluetooth low energy-enabled device such as an Android™ or iOS™-based smartphone.

You can use the BlueMS Android/iOS application (version 2.2.0 or higher), from the respective Google Play™ and iTunes™ stores, to visualize the results of the MotionFX, MotionAR, MotionCP, MotionGR and gesture detection algorithms and display the values read from the accelerometer, magnetometer, gyroscope, temperature, humidity, pressure, luminosity and proximity sensors.

Version 3.0.0 or higher allows over-the-air firmware update (with X-NUCLEO-IDB05A1 Bluetooth low energy expansion boards only).

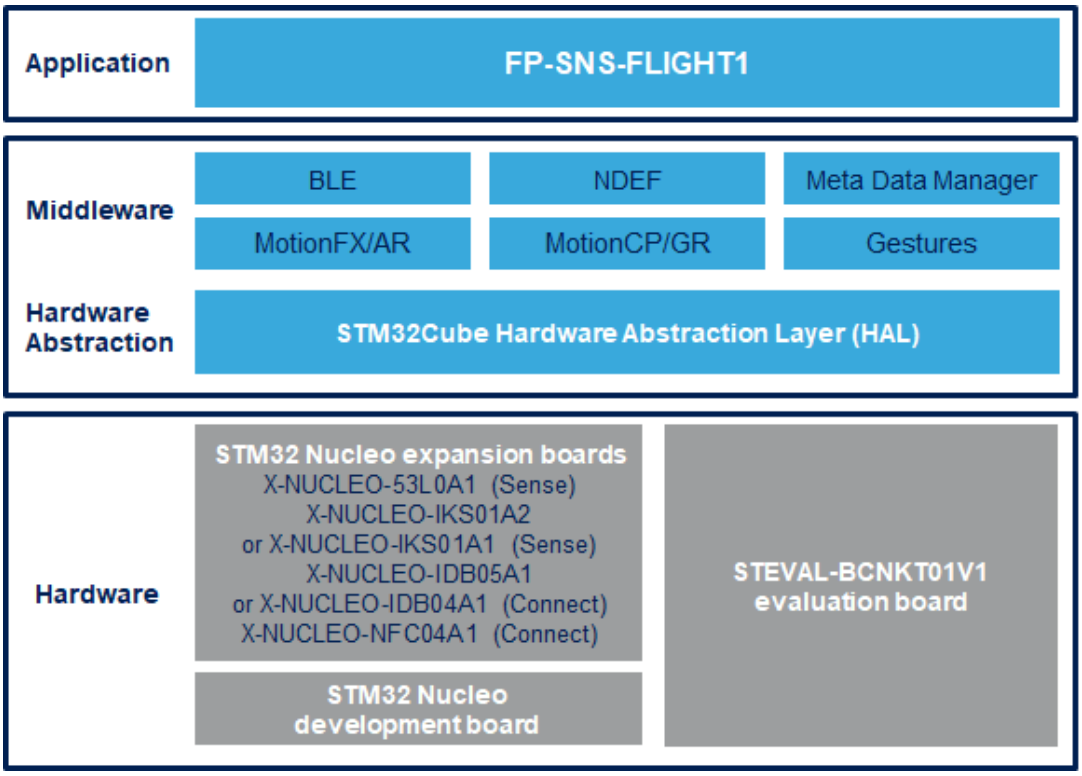
The ST BlueMS Android/iOS application was developed to enable NFC pairing prior to sensor data transmission.

The software layers used by the application software to access and use the Sensor expansion boards are:

- **STM32Cube HAL driver layer**: provides a simple, generic, multi-instance set of APIs (application programming interfaces) to interact with the upper layers (application, libraries and stacks). It is composed of generic and extension APIs and is directly built around a generic architecture and allows the layers built on top of it, such as the middleware layer, to implement their functions without dependending on the specific hardware configuration for a given microcontroller unit (MCU). This structure improves library code reusability and guarantees an easy portability across other devices.
- **Board Support Package (BSP) layer**: provides support for the peripherals (excluding the MCU) on the STM32 Nucleo board through the board support package (BSP). The BSP is a limited set of APIs which provides a programming interface for certain board-specific peripherals like the LED, the user button, etc.

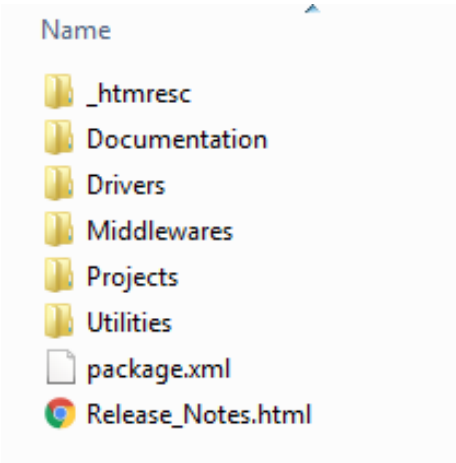
The next figure outlines the software architecture of the package:

Figure 2. FP-SNS-FLIGHT1 software architecture



2.3 Folder structure

Figure 3. **FP-SNS-FLIGHT1 package folder structure**



The following folders are included in the software package:

- **Documentation**: contains a compiled HTML file generated from the source code, detailing the software components and APIs.
- **Drivers**: contains the HAL drivers, the board specific drivers for each supported board or hardware platform, including the on-board components and the CMSIS vendor-independent hardware abstraction layer for the Cortex-M processor series.
- **Middlewares**: contains libraries and protocols for BlueNRG-1 Bluetooth low energy and NDEF devices, the Meta Data Manager, MotionFX (iNEMOEngine PRO) sensor fusion library, MotionAR (iNEMOEngine PRO) activity-recognition library, MotionCP (iNEMOEngine PRO) carry-position recognition library, MotionGR (iNEMOEngine PRO) gesture recognition library and for the gesture detection algorithm.
- **Projects**: contains a sample application used to transmit the sensor data and MotionFX sensor fusion output, MotionAR activity-recognition, MotionCP carry-position, MotionGR gesture recognition and for the gesture detection algorithm via the Bluetooth low energy protocol provided for the NUCLEO-F401RE/ NUCLEO-L476RG and STEVAL-BCNKT01V1 platforms through the IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM) and System Workbench for STM32 development environments.
- **Utilities**: contains the boot loader binary ready to be flashed for the STM32F401RE and STM32L476RG Nucleo boards.

2.4 Flash organization

Apart from storing its code, FP-SNS-FLIGHT1 uses the Flash memory to allow the Firmware-Over-The-Air update.

To enable this feature the Flash memory is divided into three different regions (see Figure 4. FP-SNS-FLIGHT1 Flash structure):

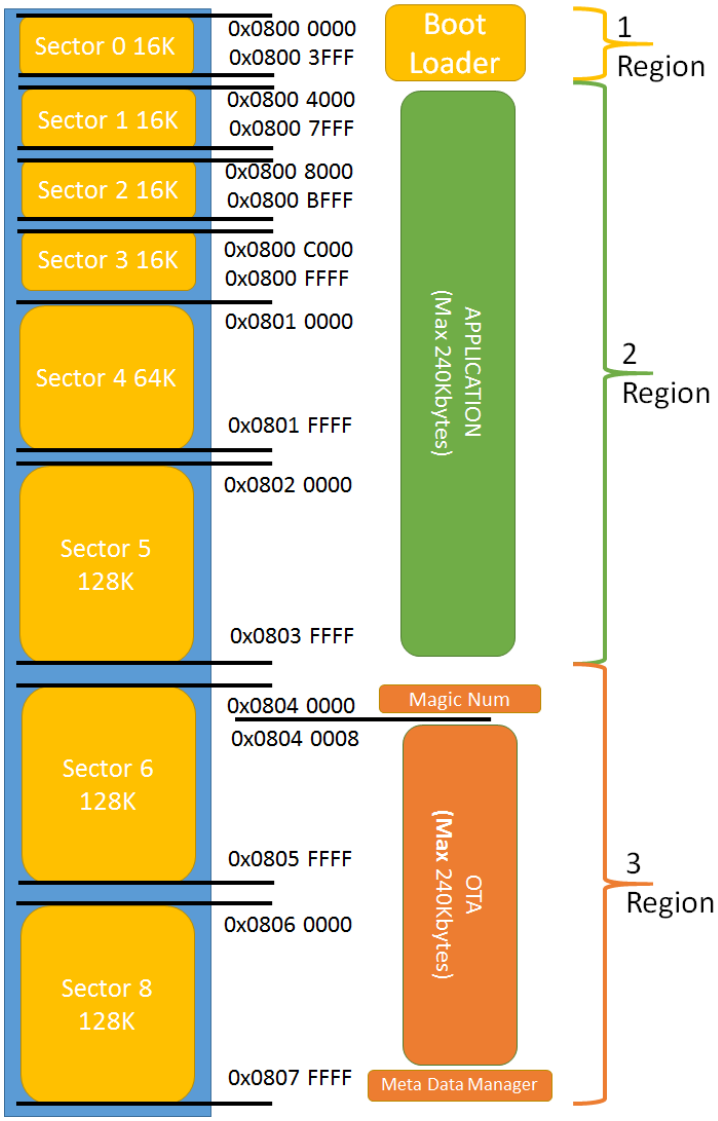
1. the first region contains a custom boot loader
2. the second region contains the FP-SNS-FLIGHT1 firmware
3. the third region is used to store the FOTA before the update

Even if the STM32F401RE (512 KB) and the STM32L476RG (1024 KB) cache sizes and arrangements differ, the same Flash arrangement has been used for both.

For further 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 4. **FP-SNS-FLIGHT1 Flash structure**



2.5 The boot process

The FP-SNS-FLIGHT1 cannot be flashed at the beginning of the Flash memory (address 0x08000000); therefore it is compiled to run from the beginning of the second Flash region (0x08004000).

To enable this behavior, the vector table offset has been set in Src/system_stm32f4xx.c (for STM32F401) and Src/system_stm32l4xx.c (for STM32L476) thus: #define VECT_TAB_OFFSET 0x4000.

The linker script has also been changed.

For example, the linker script for SP-SNS-FLIGHT1 running on STM32F401RE and compiled using IAR Embedded Workbench for ARM is:

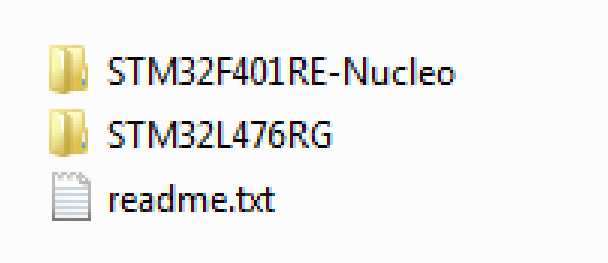
```
define symbol __ICFEDIT_intvec_start__ = 0x08004000;
      /*-Memory Regions-*/
      define symbol __ICFEDIT_region_ROM_start__ = 0x08004000;
      define symbol __ICFEDIT_region_ROM_end__ = 0x0803FFFF;
      define symbol __ICFEDIT_region_RAM_start__ = 0x20000000;
      define symbol __ICFEDIT_region_RAM_end__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

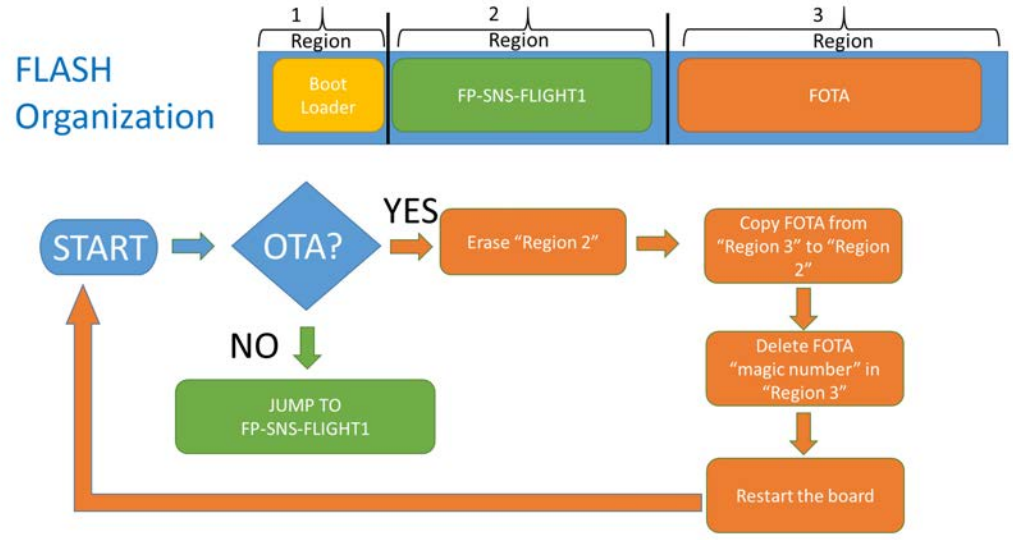
Figure 5. Bootloader folder content



At any board reset:

- if there is a FOTA in the third Flash region, the bootloader overwrites the second Flash region (with FP-SNS-FLIGHT1 firmware) and replaces its content with the FOTA restarting the board;
- if there is no FOTA, the bootloader jumps to the FP-SNS-FLIGHT1 firmware.

Figure 6. FP-SNS-FLIGHT1 Flash structure



2.6 The installation process

The package Binary directory contains an image (in .bin format) for each platform (NUCLEO-F401RE, NUCLEO-L476RG, STEVAL-BCNKT01V1), including:

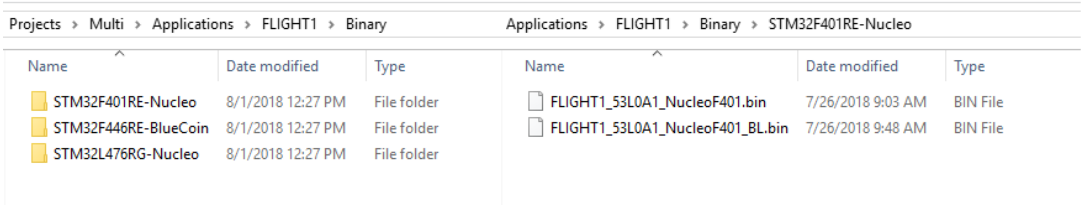
- pre-compiled FLIGHT1 firmware that may be flashed with ST-LINK to the correct memory address (0x08004000) of a supported STM32 Nucleo development board

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

- pre-compiled FLIGHT1 plus BootLoader firmware that may be directly flashed to a supported STM32 Nucleo development board with the ST-LINK or via a Drag & Drop operation (STM32 Nucleo boards only)

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

Figure 7. Binary folder content

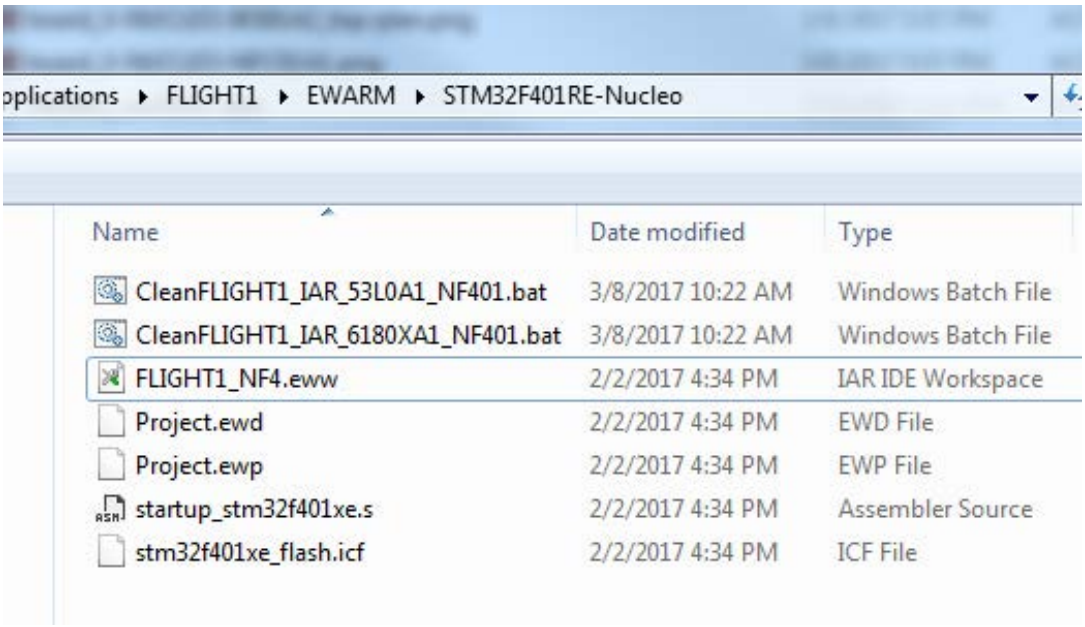


To flash modified FLIGHT1 firmware, simply flash the compiled FP-SNS-FLIGHT1 firmware to the correct address (0x08004000).

A batch script has been provided to simplify this operation by saving the firmware and the BootLoader on the right position; it is available for each platform (NUCLEO-F401RE, NUCLEO-L476RG and STEVAL-BCNKT01V1) and for each IDE (IAR/RealView/System Workbench):

- IAR toolchain Embedded Workbench V8.20.2:
  - For Nucleo F401: CleanFLIGHT1_IAR_53L0A1_NF401.bat
  - For Nucleo L4: CleanFLIGHT1_IAR_53L0A1_NL476.bat
  - STEVAL-BCNKT01V1: CleanFLIGHT1_IAR_BC.bat
- µVision toolchain - MDK-ARM Professional Version: 5.24.2:
  - For Nucleo F4: CleanFLIGHT1_MDK_ARM_53L0A1_NF401.bat
  - For Nucleo L4: CleanFLIGHT1_MDK_ARM_53L0A1_NL476.bat
  - For STEVAL-BCNKT01V1: CleanFLIGHT1_MDK_ARM_BC.bat
- System Workbench for STM32 Version 2.4.0.201801120948:
  - For Nucleo F4: CleanFLIGHT1_SW4STM32_53L0A1_NF401.bat
  - For Nucleo L4: CleanFLIGHT1_SW4STM32_53L0A1_NL476.bat
  - For STEVAL-BCNKT01V1: CleanFLIGHT1_SW4STM32_BC.bat

Figure 8. Content of a project folder



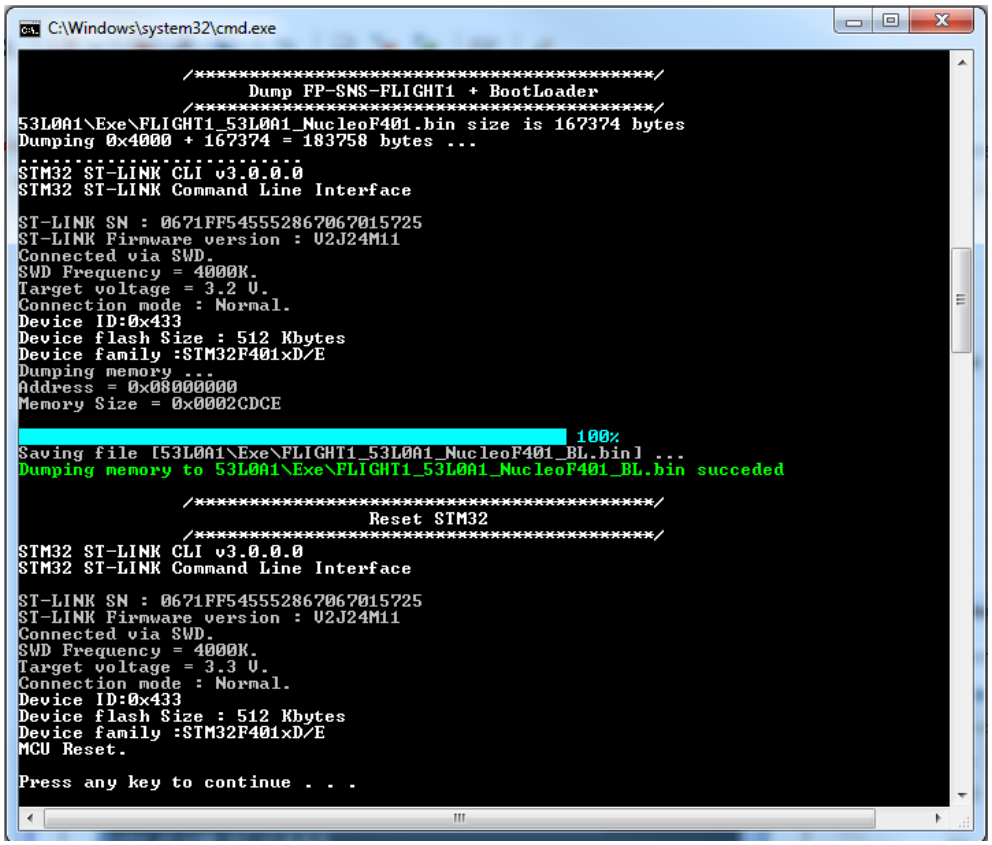
This script:

- performs a full Flash erase to start from a clean system
- flashes the BootLoader to the correct position 0x08000000
- flashes the firmware to the correct position 0x08004000

Figure 9. **BootLoader and FP-SNS-FLIGHT1 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 10. **FP-SNS-FLIGHT1 dump process**



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

- CleanFLIGHT1_SW4STM32_53L0A1_NF401.sh
- CleanFLIGHT1_SW4STM32_53L0A1_NL476.sh
- CleanFLIGHT1_SW4STM32_BC.sh

To function, the script must be modified with:

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

TheOpenOCD script section to be edited is:

```
# 1) Set the Installation path for OpenOCD
# example:
#OpenOCD_DIR="C:/Ac6/SystemWorkbench/plugins/fr.ac6.mcu.externaltools.openocd.win32_1.17.0.20
1801120948/tools/openocd/"
OpenOCD_DIR=""

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

2.7 Firmware-over-the-air (FOTA) update

For the X-NUCLEO-IDB05A1 Bluetooth low energy expansion board only, the FP-SNS-FLIGHT1 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 at their respective application web stores.

The application sends the update and associated CRC (cyclic-redundancy-check) value that the FP-SNS-FLIGHT1 checks against the hardware cyclic redundancy check calculation unit on the STM32F401/STM32L476 processor to ensure 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” prompts the boot loader that a firmware update has been received, checked and is ready to replace the current FP-SNS-FLIGHT1 firmware (see Section 2.11 Firmware over-the-air (FOTA) update with BlueMS).

2.8 APIs

Detailed technical information regarding the APIs available to the user can be found in a compiled HTML file located inside the “Documentation” folder of the software package, where all the functions and parameters are fully described.

2.9 Sample application description

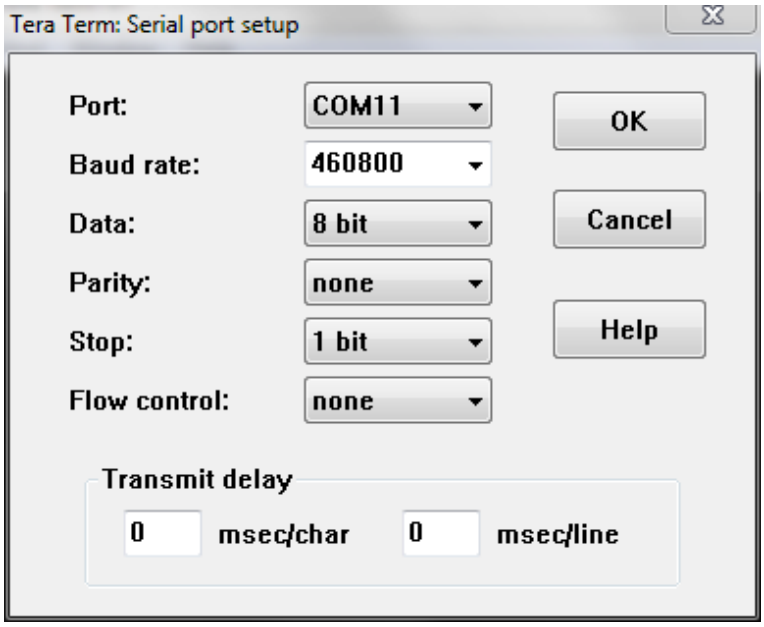
A sample application using:

- the X-NUCLEO-NFC04A1, X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A1), X-NUCLEO-53L0A1 and X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1) expansion boards with the NUCLEO-F401RE or NUCLEO-L476RG board
- the STEVAL-BCNKT01V1 evaluation board

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 (as per Figure Terminal setting) to control the initialization phase.

Figure 11. **Terminal setting**



When you first press the reset button, the application:

1. starts initializing the UART, I²C
2. declares which MEMS sensor expansion board is plugged and if all the sensors are working
3. with X-NUCLEO-IKS01A1, checks whether the LSM6DS3 DIL24 extension is present
4. with the NDEF protocol, writes the www.st.com/stm32ode URI to the M24SR dynamic NFC tag on the X-NUCLEO-NFC04A1 expansion board
5. declares whether the firmware is compiled for X-NUCLEO-53L0A1

Figure 12. **Initialization phase**

When reading the NFC tag content with an Android device, the browser automatically starts and tries to connect to that URI.

When the user presses the blue user button (see Figure 13. UART console output when the BLE services are started), the program:

1. initializes the proximity sensor and discovery of the satellites
2. initializes the SPI interface used for communicating with the BlueNRG-1 expansion board
3. determines which BlueNRG-1 expansion board (X-NUCLEO-IDB04A1 or X-NUCLEOIDB05A1) is connected to the STM32 Nucleo board and the hardware and firmware versions
4. creates the random BLE MAC address and PIN necessary for the connection
5. initializes the BLE hardware service (adding the temperature, humidity, pressure, distance, 3D gyroscope, 3D magnetometer and 3D accelerometer characteristics)
6. initializes the BLE console service adding the stdin/stdout and stderr characteristics
7. initializes the BLE configuration service
8. changes the content of the ST25DV04K dynamic NFC tag (using NDEF) in order to write all the information necessary to automatically launch the BlueMS Android application (name of application, BLE advertise data, BLE MAC address and BLE connection PIN).

Figure 13. **UART console output when the BLE services are started**

When reading the above NFC content on an Android device with the BlueMS application installed, it is possible to automatically launch the BlueMS application to connect the device with the STM32 Nucleo board, without having to scan for the board or manually insert the PIN.

iOS does not allow the use of NFC for this purpose, so you must manually run the application, scan for the STM32 Nucleo board and enter the connection PIN.

The `FLIGHT1_SECURE_CONNECTION` define in the Projects\Multi\Applications\FLIGHT1\Inc\flight1_config.h file controls whether the STM32 Nucleo board only accepts secure connections (default) or any connection (define is commented), so you do not have to enter the BLE connection PIN for a device to connect to the STM32 Nucleo board.

As the console output shows, the application sends:

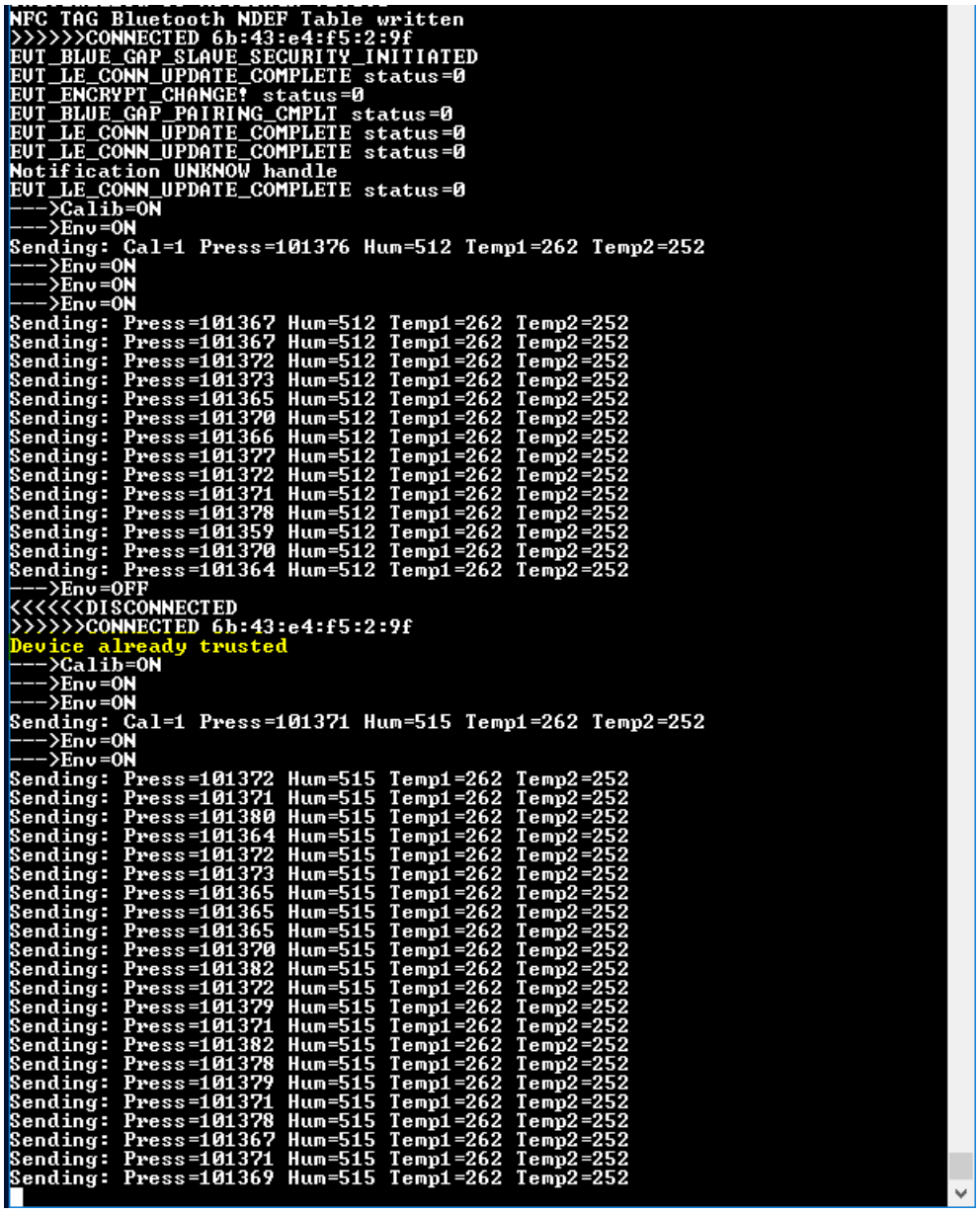
- the values of temperature, humidity and pressure every 500 ms
- the values of the 3D accelerometer, 3D gyroscope and 3D magnetometer every 100 ms
- the proximity values every 50 ms

Connection of an Android/iOS device to the STM32 Nucleo board starts with the secure pairing procedure, where ping information is sent to the stdout console BLE characteristic.

Figure 14. **UART console output when a device first connects with the board**

The application has a white list of one element, so subsequent connections with the last trusted device are automatically authenticated (see figure below).

Figure 15. **UART console output when one device are already trusted**



2.10 Android and iOS sample client application

The FP-SNS-FLIGHT1 software for STM32Cube is compatible with the BlueMS Android/iOS applications (Version 2.2.0 and above), available at the respective Google Play/iOS stores.

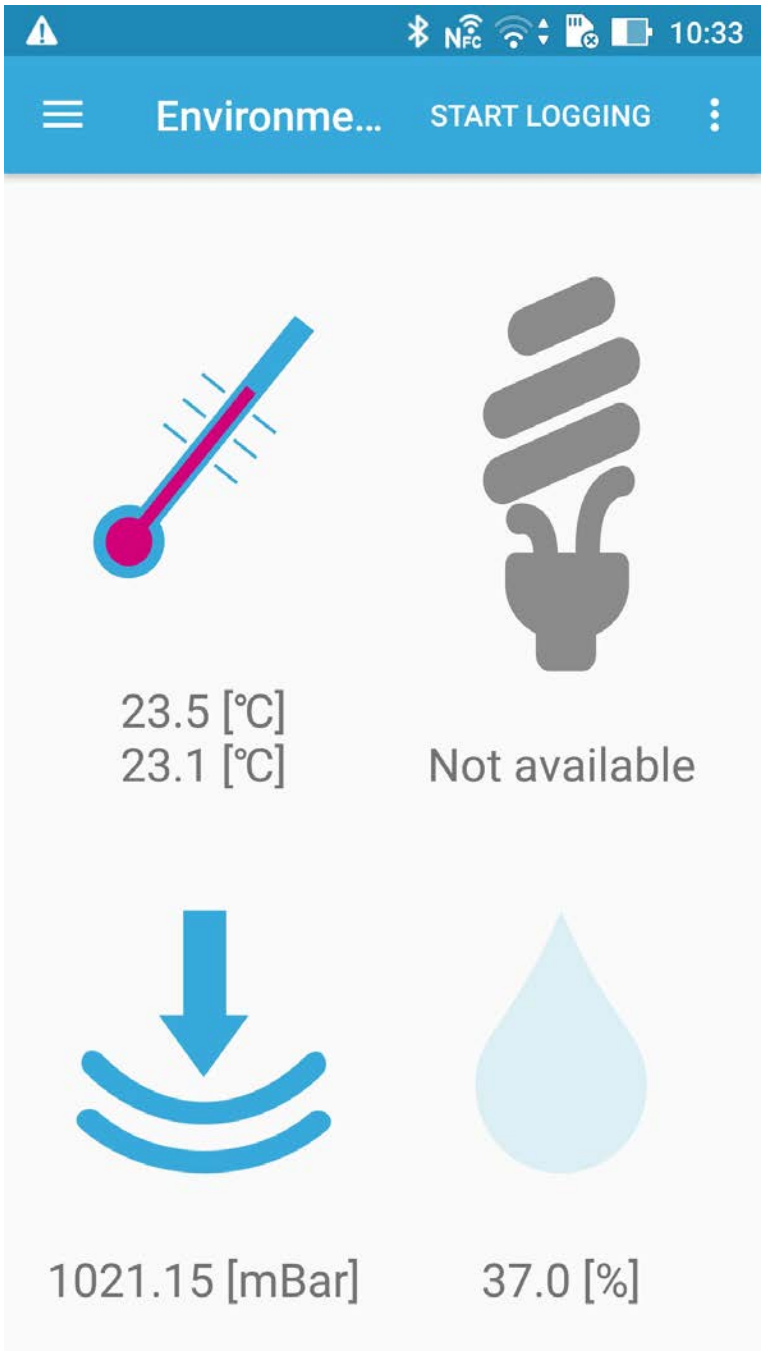
The BlueMS application allows Over-The-Air firmware updates (for X-NUCLEO-IDB05A1 Bluetooth low energy expansion boards only).

Note: For STEVAL-BCNKT01V1 evaluation boards, the firmware package runs if the BlueCoin module (STEVAL-BCNCS01V1) is plugged onto Coin Station board (STEVAL-BCNST01V1) where there are 2 Time-of-Flight (ToF) ranging sensor (VL53L0X).

The Android application is used here to show how the application works.

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

Figure 16. **BlueMS (android version) main page (after BLE connection)**



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

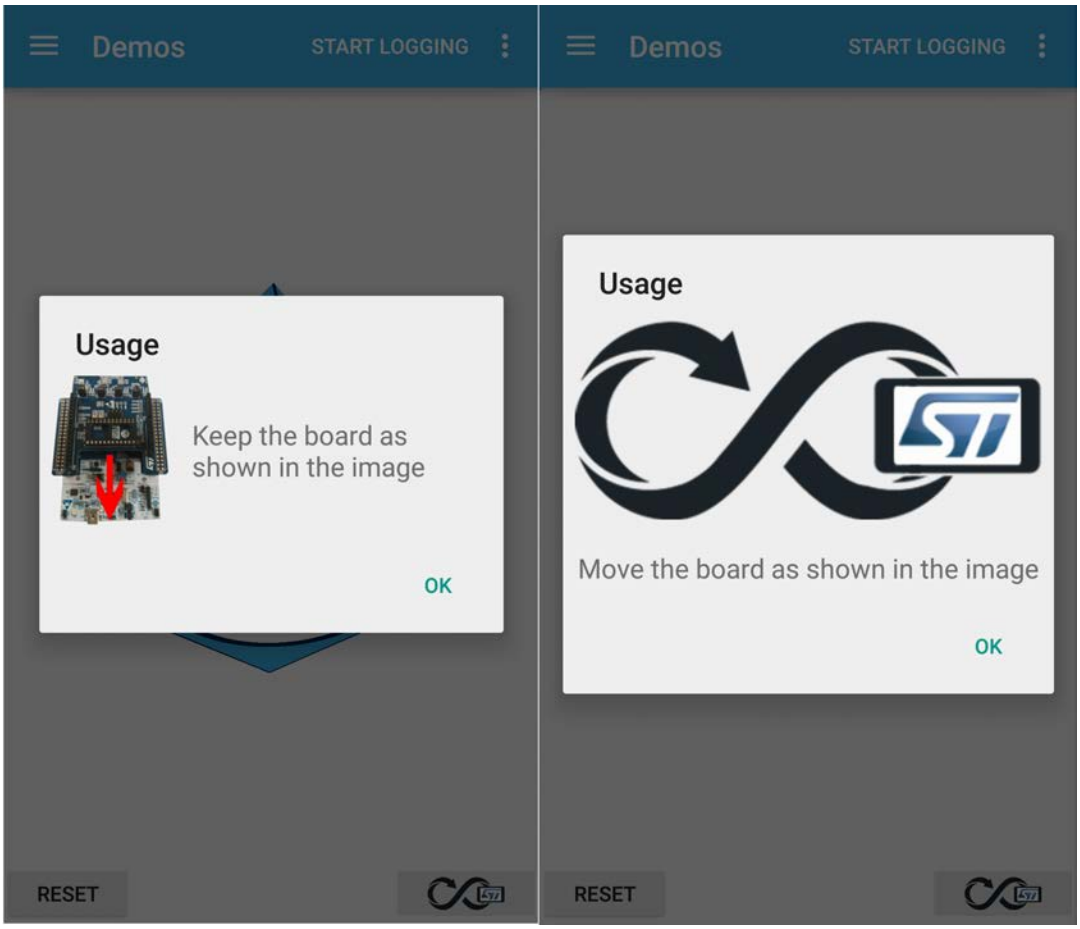
Figure 17. **BlueMS (Android version) MotionFX sensor fusion page**

In the page above:

- if the board is calibrated, the central button enables or disables the proximity sensor, which triggers the cube zooming out or in as a function of the proximity measured by the X-NUCLEO-53L0A1 expansion board;
- the left button resets the cube position;
- the right button shows the MotionFX library calibration status (black for not calibrated, green for calibrated). Clicking it forces a magneto calibration.

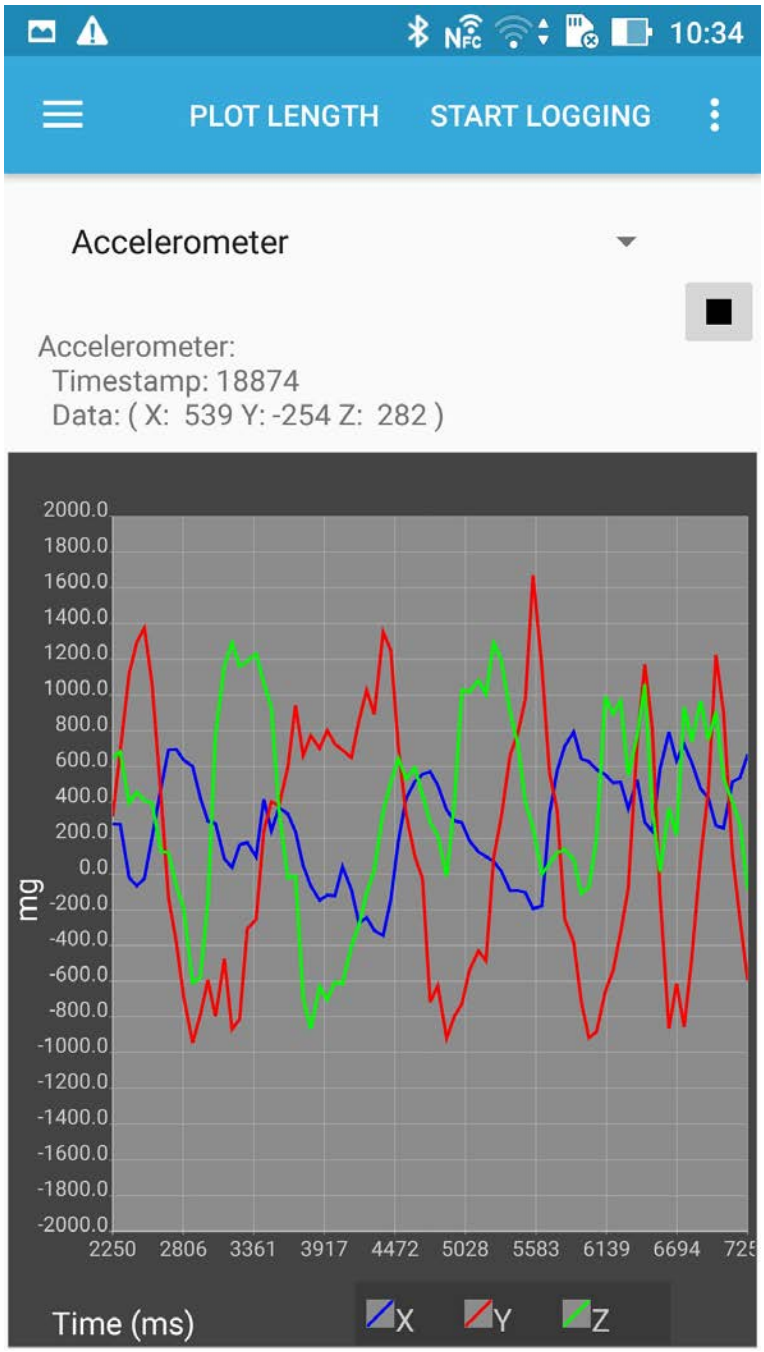
When a button is pressed, the application pops up a window describing how to position the board for a correct cube rotation and how to move the board to facilitate calibration (see figure below).

Figure 18. **BlueMS (Android version) popup windows**



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

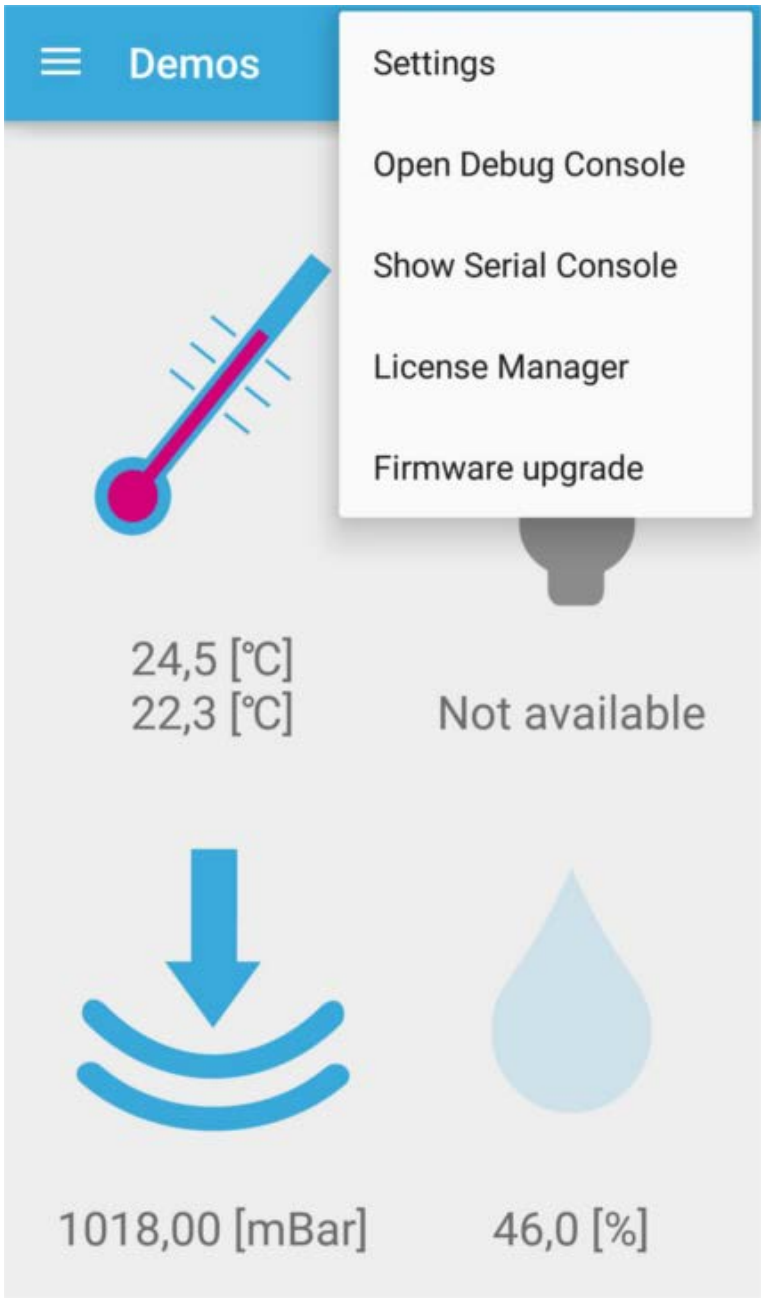
Figure 19. **BlueMS (android version) example of plot value**



In the option menu below, you can access:

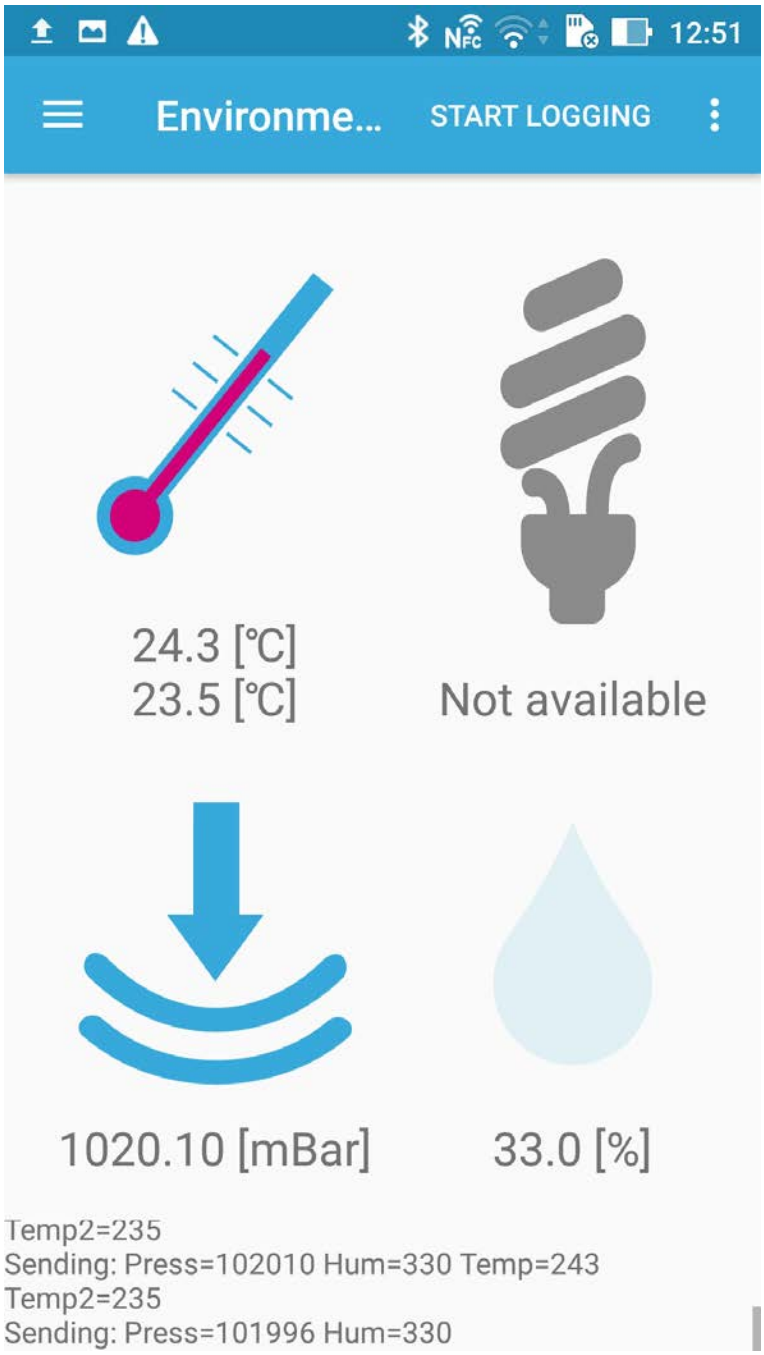
- Serial or Debug (with stdin) console
- firmware upgrade

Figure 20. **BlueMS (android version) menu selection**



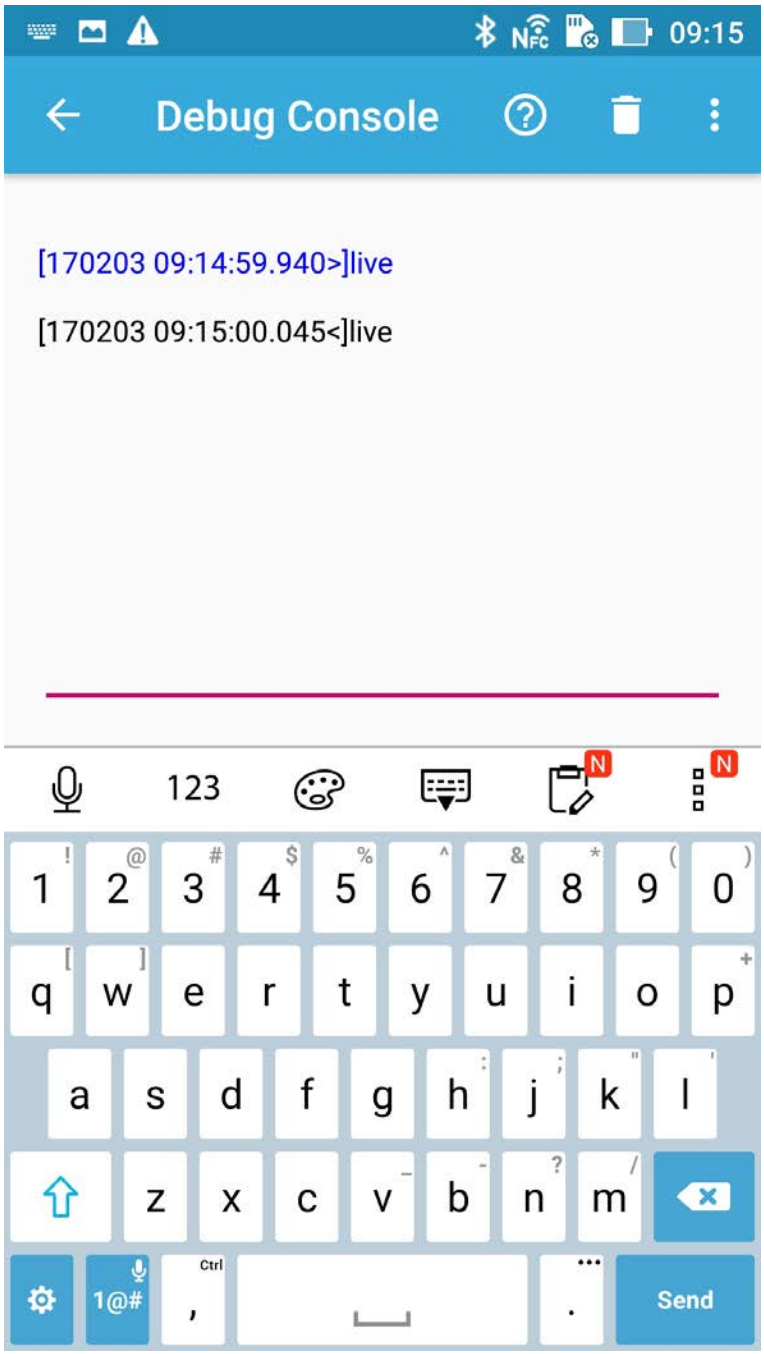
If the Serial console is enabled, stdout/stderr is displayed, as shown below.

Figure 21. **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 22. **BlueMS (android version) Debug console (stdin/stdout/stderr)**



Through the Debug console, you can change the sensor value transmission frequency.

- For temperature/humidity/pressure with the command:
  - @TL: the application sends environmental data every 5 s
  - @TM: the application sends environmental data every 1 s
  - @TH: the application sends environmental data every 100 ms
  - @TD: the application sends environmental data as default (500 ms).
- For 3D accelerometer, 3D gyroscope and 3D magnetometer with the command:
  - @AL: the application sends the data every 500 ms
  - @AM: the application sends the data every 100 ms

  - @AH: the application sends the data every 50 ms
  - @AD: the application sends the data as default (100 ms).
- For distance (proximity) with the command:
  - @PL: the application sends the data every 500 ms
  - @PM: the application sends the data every 100 ms
  - @PH: the application sends the data every 50 ms
  - @PD: the application sends the data as default (50 ms).

Figure 23. BlueMS (android version) Debug console - change transmission frequency

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

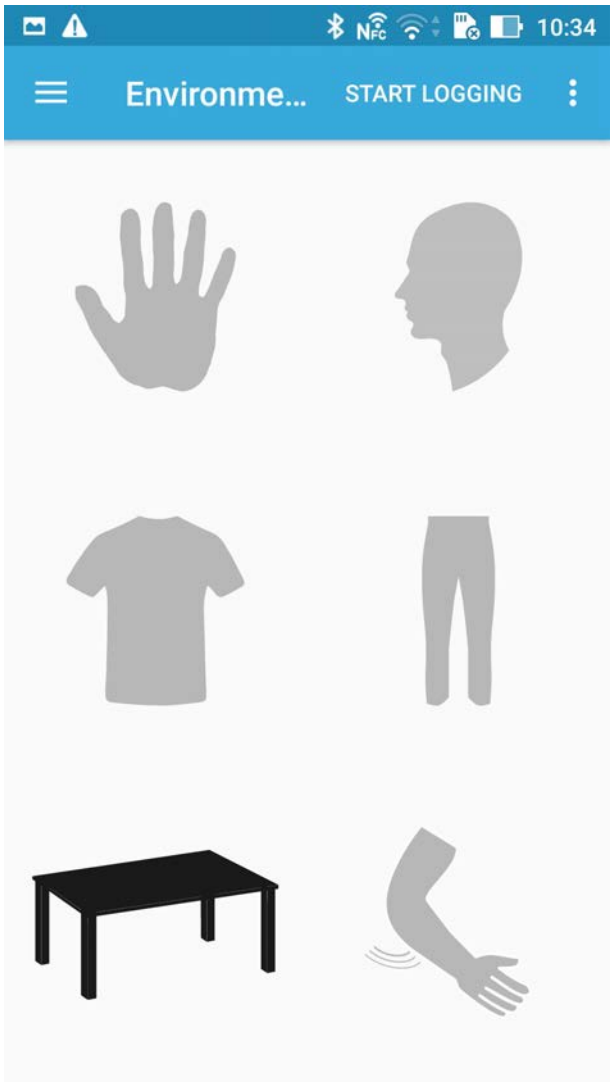
- stationary
- walking
- fast walking
- jogging
- biking
- driving

Figure 24. **BlueMS (Android version) MotionAR activity recognition page**

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

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

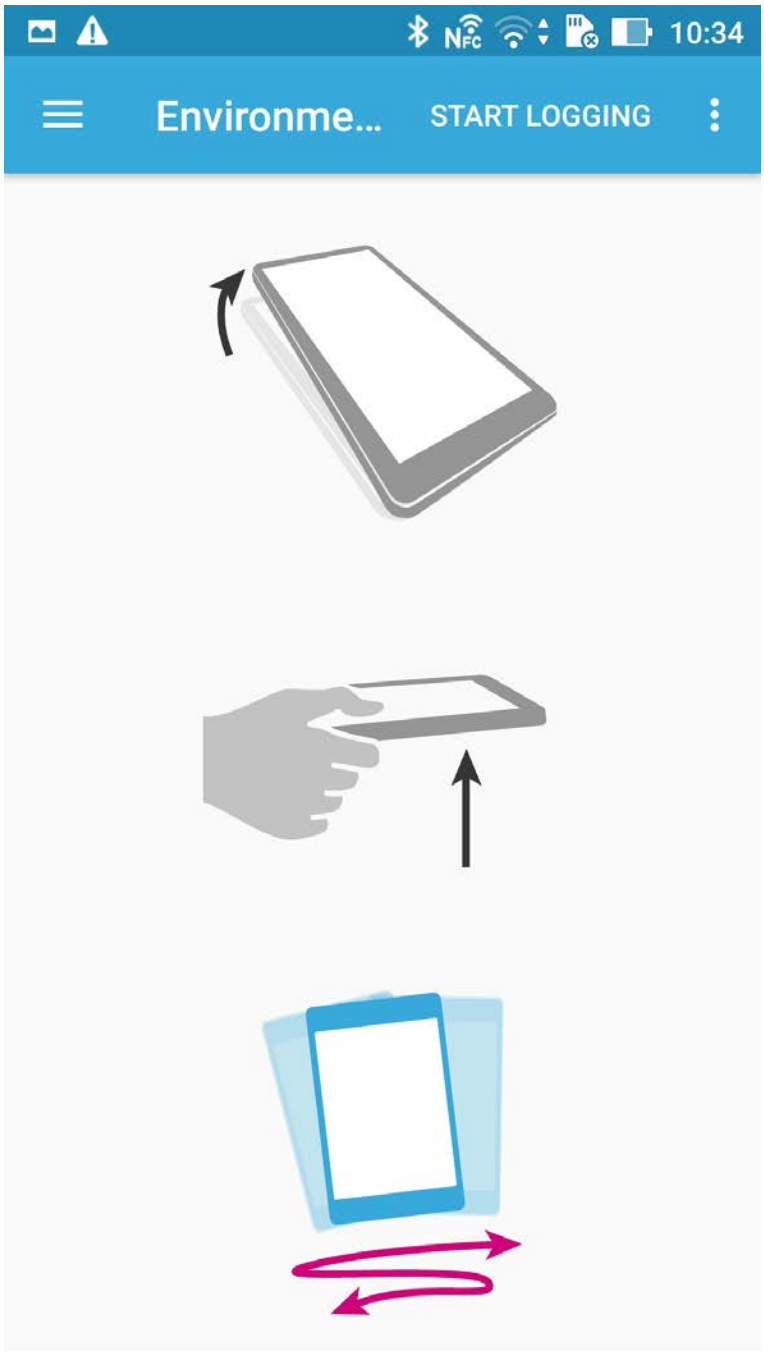
Figure 25. **BlueMS (Android version) MotionCP carry position recognition page**



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

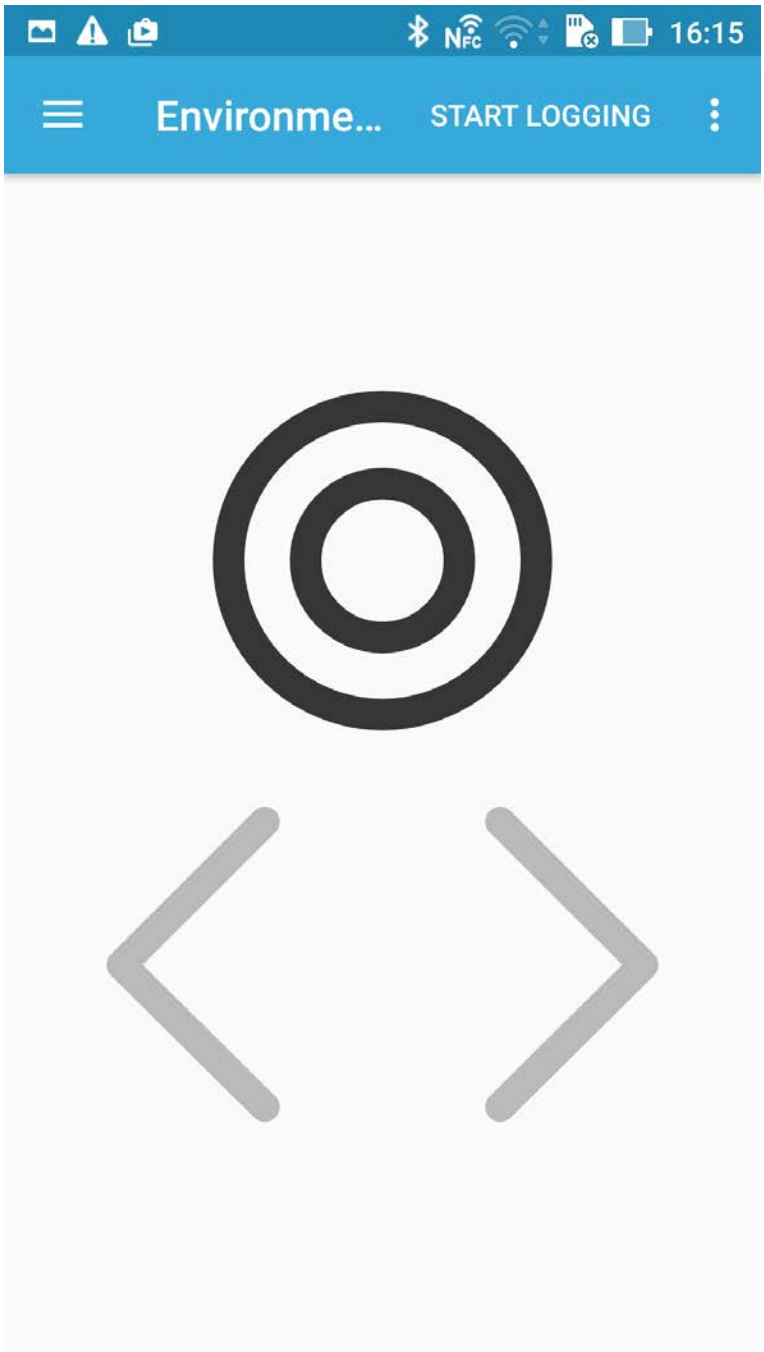
- pick up
- glance
- wake up in hand

Figure 26. **BlueMS (Android version) MotionGR gesture recognition page**



If gesture detection is enabled, the page shown below displays the results of the detected gestures, which can be a single tap (indicated by the circular symbol) or directional swipes (indicated by the double arrows).

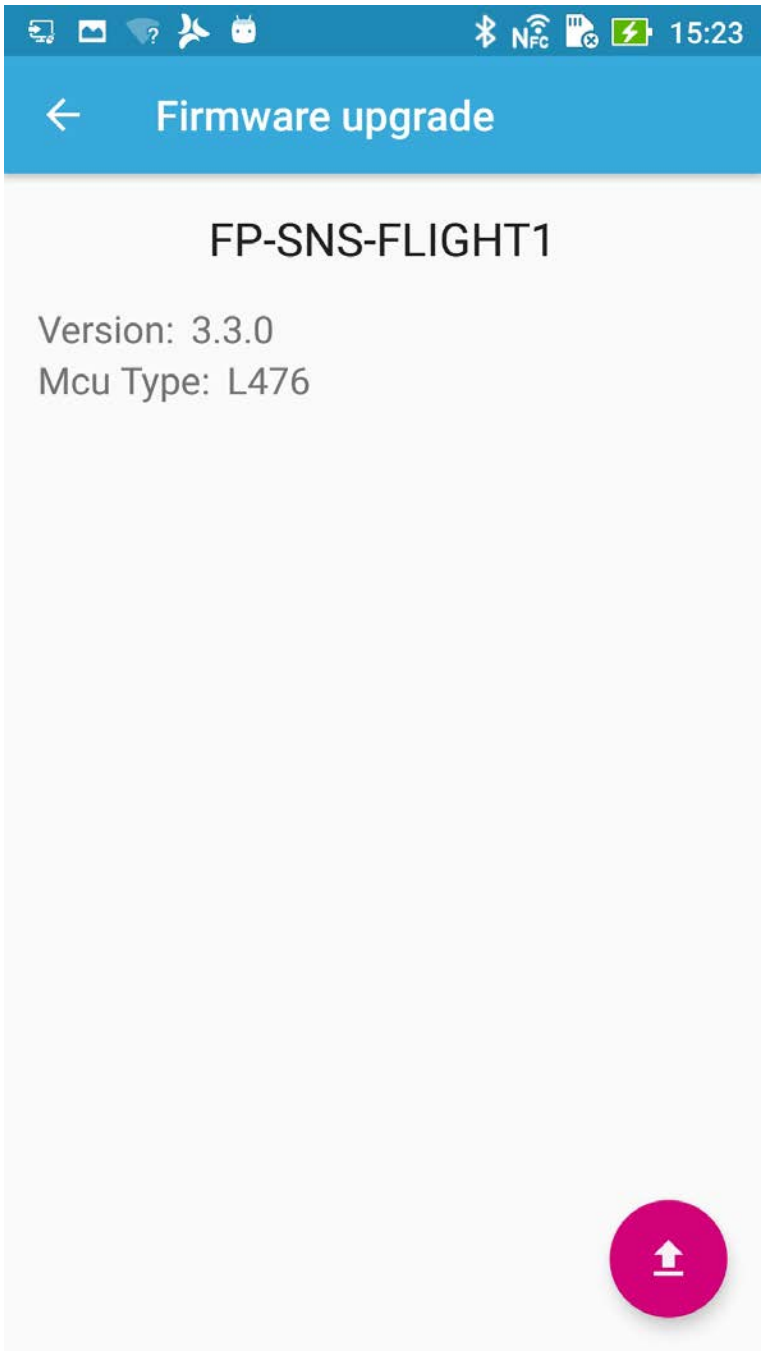
Figure 27. **BlueMS (Android version) gesture detection page**



2.11 Firmware over-the-air (FOTA) update with BlueMS

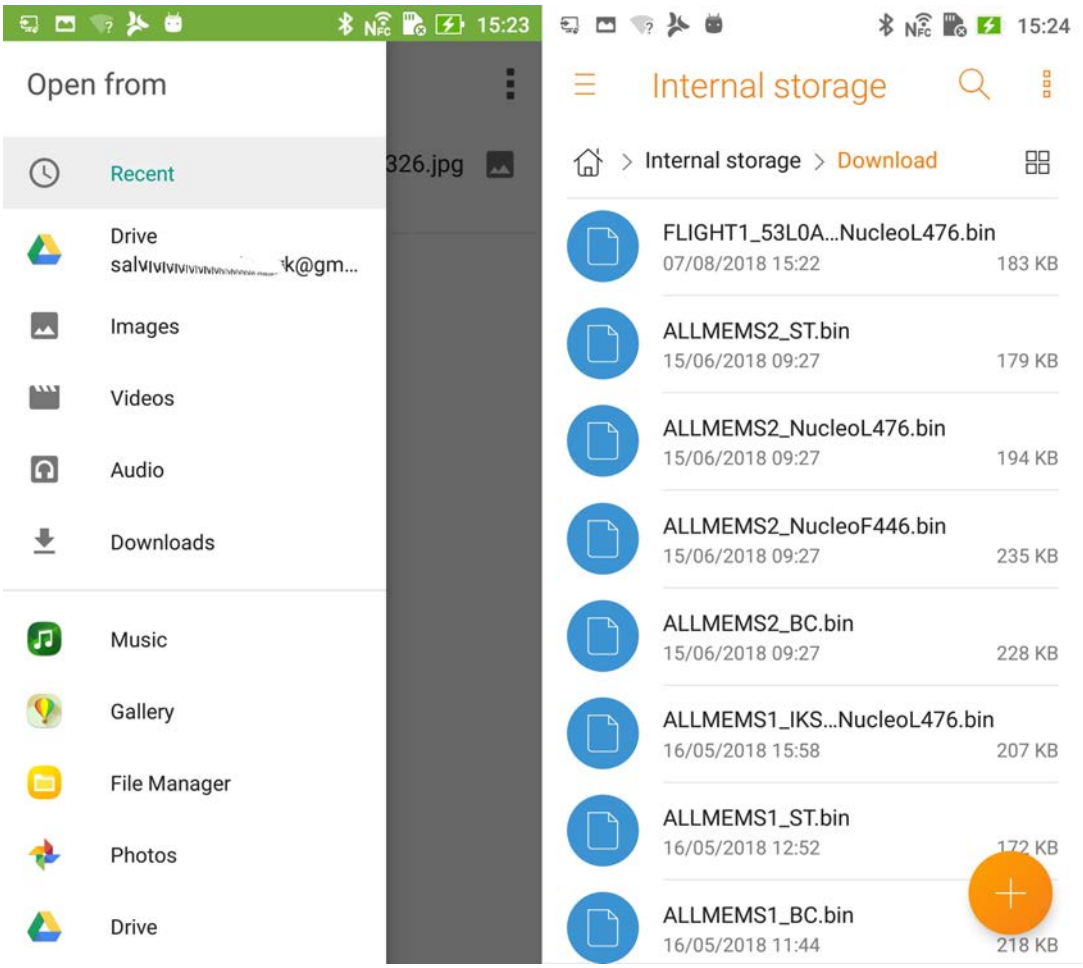
If the 'Firmware upgrade' menu option is selected in the BlueMS main application page, the following page appears:

Figure 28. **BlueMS (Android version) firmware upgrade page**



The BlueMS application shows which version of the FP-SNS-FLIGHT1 software is running and the board type. To apply an update, press the red button and select the appropriate update file.

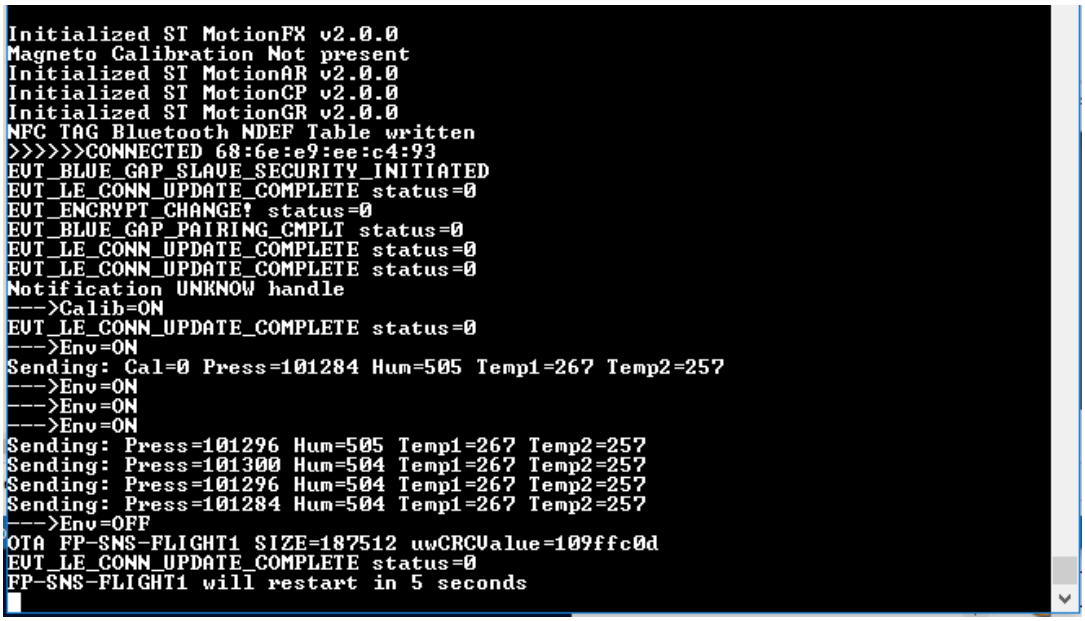
Figure 29. BlueMS (Android version) firmware update file selection



BlueMS sends to FP-SNS-FLIGHT1 an update of a certain byte size and corresponding CRC value.

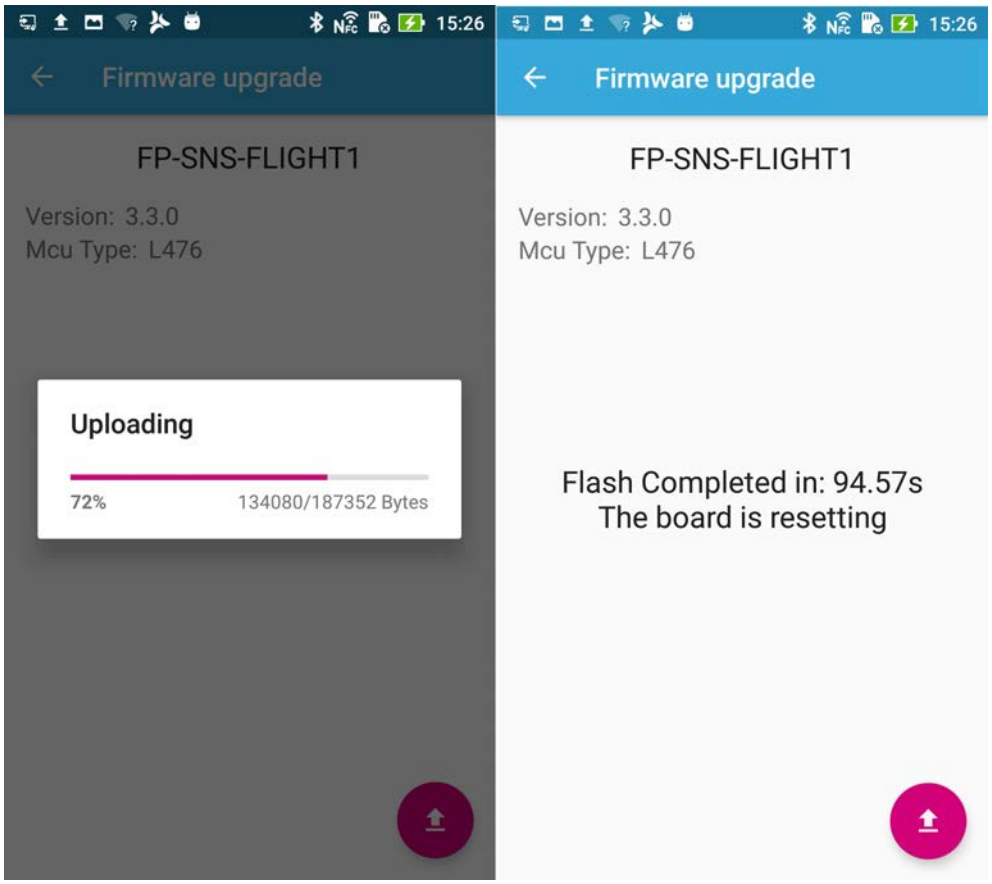
The figure below shows the terminal window with the debug information returned during FOTA for an STM32 Nucleo platform (STM32F401RE/L476RG) when we use a UART to control FP-SNS-FLIGHT1 behavior.

Figure 30. **Terminal window information during FOTA**



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

Figure 31. **BlueMS (Android version) application page during FOTA and on completion**



3 System setup guide

3.1 Hardware description

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

3.1.1 STM32 Nucleo platform

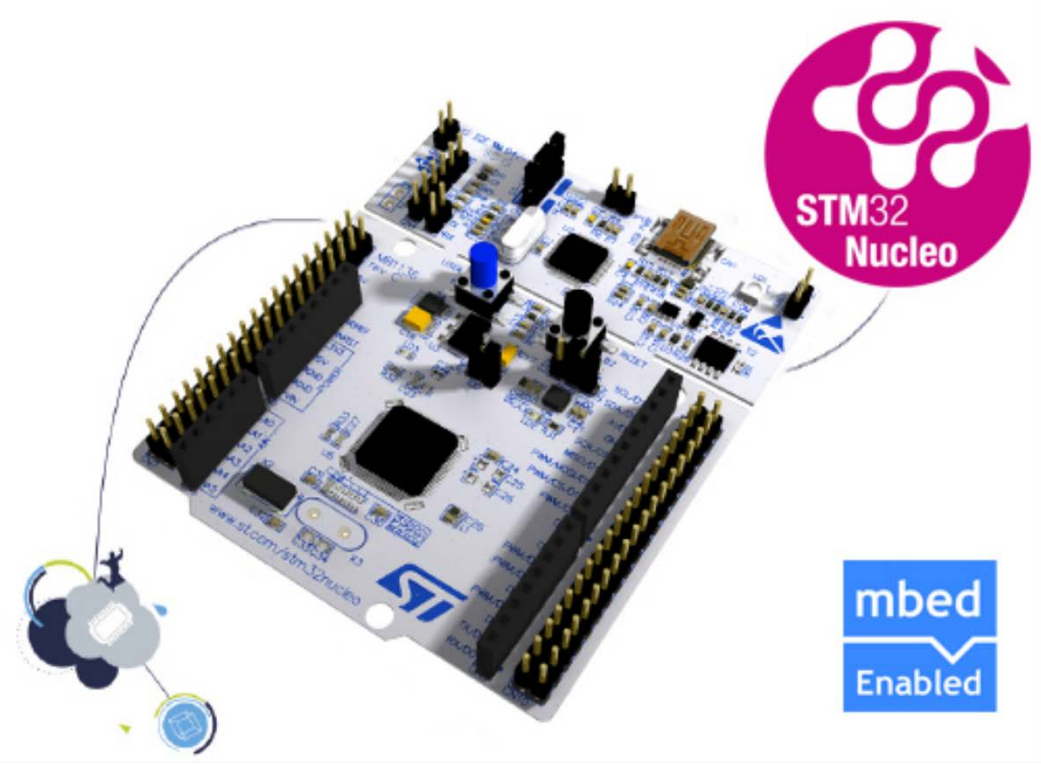
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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

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

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

Figure 32. STM32 Nucleo board

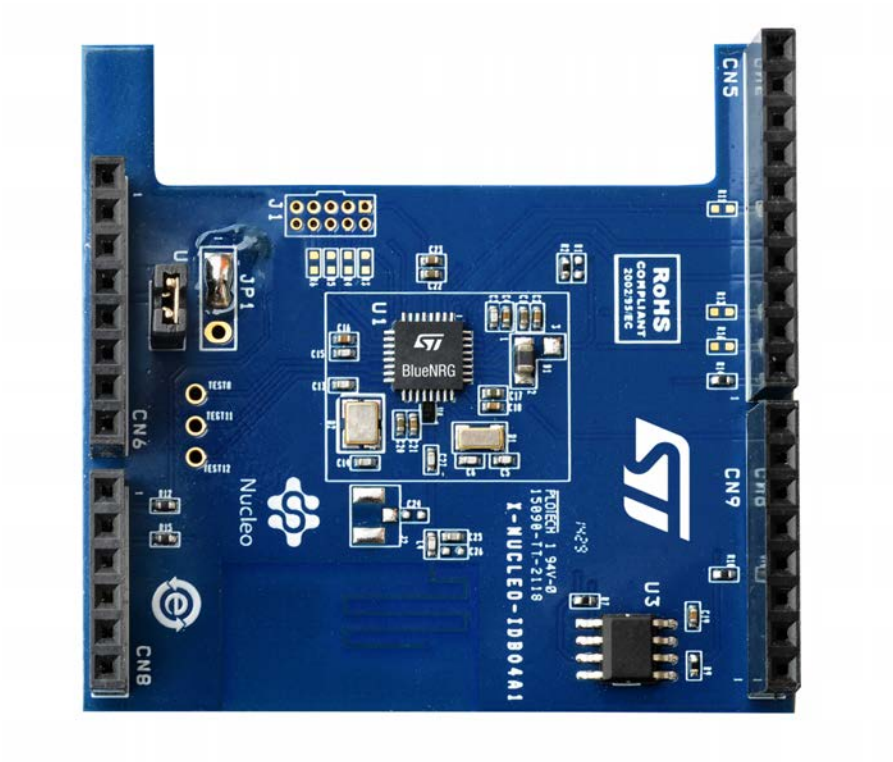


Information regarding the STM32 Nucleo board is available at www.st.com/stm32nucleo

3.1.2 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 33. **X-NUCLEO-IDB04A1 expansion board**

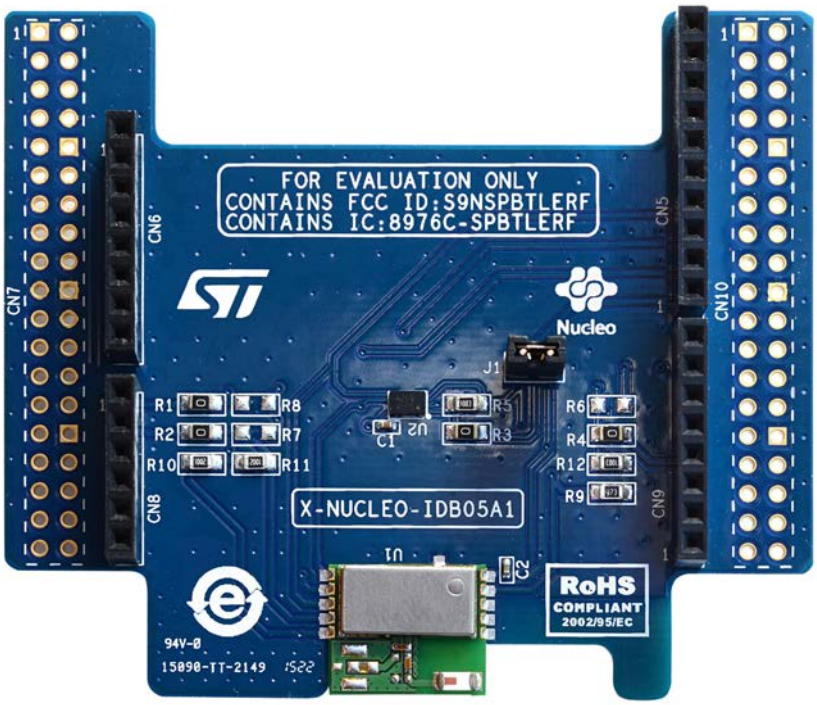


Information regarding the X-NUCLEO-IDB04A1 expansion board is available on www.st.com at http://www.st.com/x-nucleo.

3.1.3 X-NUCLEO-IDB05A1 expansion board

The X-NUCLEO-IDB05A1 is a Bluetooth low energy expansion board based on the SPBTLE-RF RF module, built around the BlueNRG-MS network processor, 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 34. **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

3.1.4 X-NUCLEO-NFC04A1 expansion board

The X-NUCLEO-NFC04A1 dynamic NFC/RFID tag IC expansion board is based on the ST25DV04K NFC Type V/ RFID tag IC with a dual interface 4 Kbits EEPROM that also features an I²C interface. It can be powered by the pin of Arduino connector or directly by the received carrier electromagnetic field.

The X-NUCLEO-NFC04A1 expansion board is compatible with the Arduino™ UNO R3 connector pin assignment and can easily be plugged onto any STM32 Nucleo board. Various expansion boards can also be stacked to evaluate different devices operating together with the dynamic NFC tag. The board also features an antenna with a 54 mm ISO 24.2 diameter, single layer, copper etched on PCB.

Figure 35. **X-NUCLEO-NFC04A1 expansion board**

3.1.5 X-NUCLEO-IKS01A1 expansion board

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

You can attach the LSM6DS3 DIL24 expansion component and use it instead of the one of the LSM6DS0 sensors.

Figure 36. **X-NUCLEO-IKS01A1 expansion board**

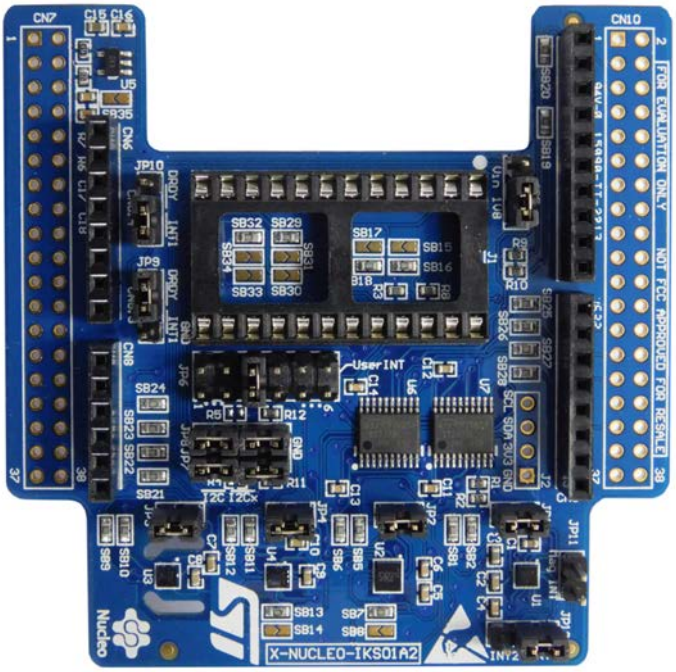
3.1.6 X-NUCLEO-IKS01A2 expansion board

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



3.1.7 X-NUCLEO-53L0A1 expansion board

The X-NUCLEO-53L0A1 is an expansion board for the STM32 Nucleo system, also compatible with Arduino UNO R3 connector layout and designed around ST VL53L0X ranging and gesture detection sensor, based on ST FlightSense™, Time-of-Flight technology.

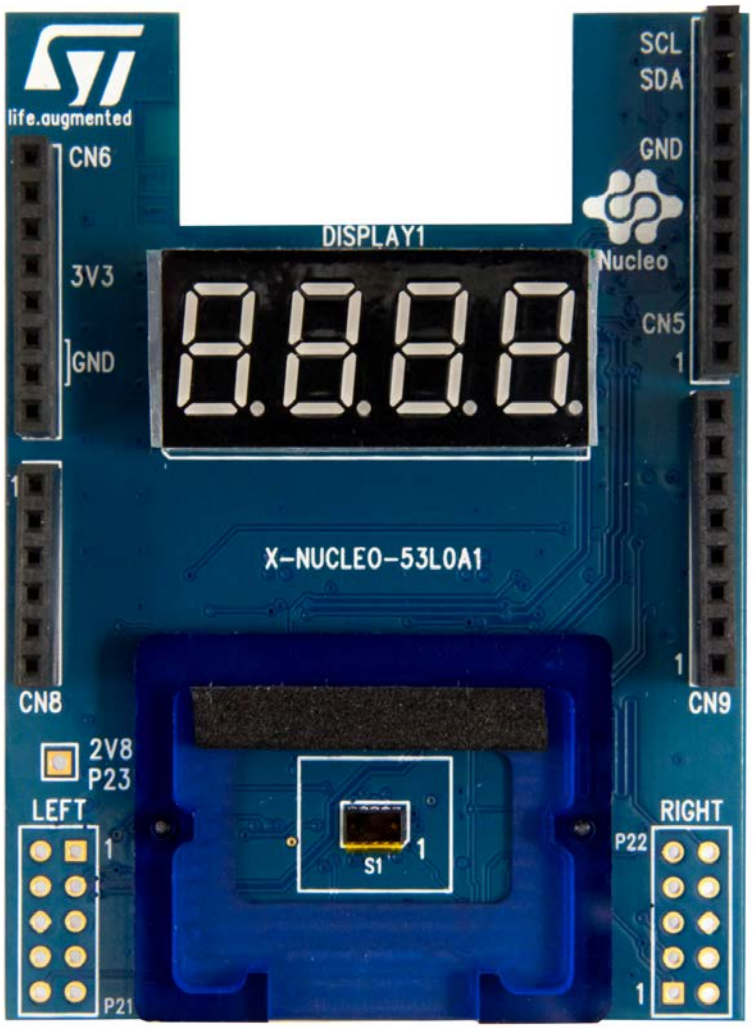
Several ST expansion boards can be superposed through the Arduino connectors, which allows, for example, to develop VL53L0X applications with Bluetooth or Wi-Fi interface.

To allow the user to quickly access the gesture recognition demonstration, the X-NUCLEO-53L0A1 expansion board is delivered with two VL53L0X satellites.

The key features are:

- VL53L0X ranging and gesture detection sensor module;
- accurate absolute ranging distance, independent of the reflectance of the target;
- a 4-digit display, displaying the distance of a target from the ranging sensor;
- two 10-pin connectors;
- a cover glass holder;
- 3 different spacers of 0.25, 0.5 and 1 mm height to be fitted below the cover glass in order to simulate various air gaps.

Figure 38. **X-NUCLEO-53L0A1 expansion board**



3.1.8 STEVAL-BCNKT01V1 BlueCoin development kit

3.1.8.1 *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.

3.1.8.2 *Features*

- Contains FCC ID: S9NBCOIN01

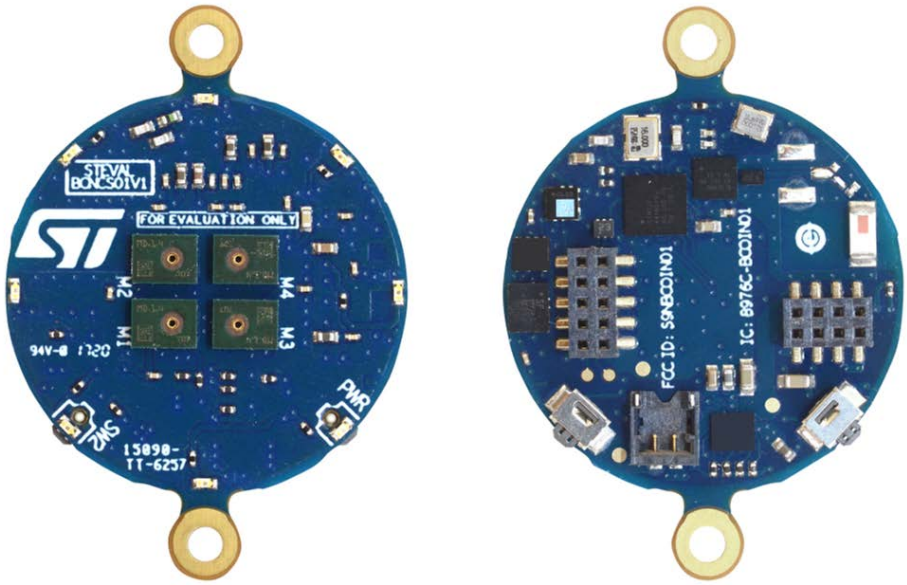
- Contains module IC 8976C-BCOIN01 certified with PMN: STEVAL-BCNKT01V1; HVIN: STEVAL-BCNCS01V1; HMN: STEVAL-BCNCR01V1; FVIN: bluenrg_7_2_c_Mode_2-32MHz-XO32K_4M.img
- The development kit package includes:
  - BlueCoin module (STEVAL-BCNCS01V1) with STM32F446, LSM6DSM, LSM303AGR, LPS22HB, 4x MP34DT04-C1, BlueNRG-MS, BALF-NRG-01D3, STBC03JR
  - CoinStation (STEVAL-BCNST01V1) board
  - BlueCoin Cradle (STEVAL-BCNCR01V1)
  - 130 mAh Li-Po battery
  - Plastic box for housing the BlueCoin cradle and the battery
  - SWD programming cable
- Software libraries and tools:
  - STSW-BCNKT01 firmware package with raw sensor data streaming support via USB, data logging on SD card, audio acquisition and audio streaming, time-of-flight example and BLE protocol to interface to a smartphone app
  - FP-AUD-SMARTMIC1: smart audio IN-OUT software expansion for STM32Cube
  - FP-SNS-ALLMEMS1 and FP-SNS-ALLMEMS2: STM32Cube function packs for BLE and sensors
  - FP-AUD-BVLINK1: BLE and microphones software expansion for STM32Cube
  - BlueMS: iOS™ and Android™ demo apps
  - BlueST-SDK: iOS and Android software development kit
  - Compatible with STM32 ecosystem through STM32Cube support

3.1.8.3 *Content of the starter kit*

STEVAL-BCNCS01V1 - BlueCoin Core System board features

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

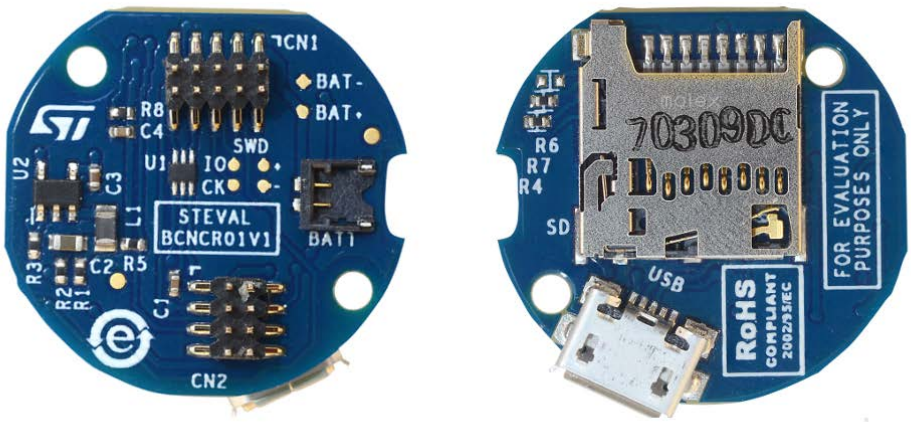
Figure 39. **STEVAL-BCNCS01V1 - BlueCoin Core System**



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

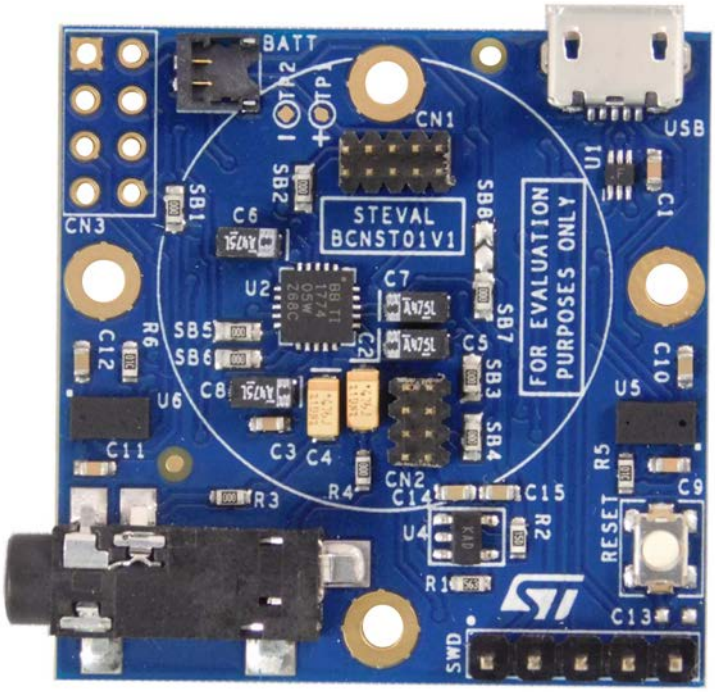
Figure 40. **STEVAL-BCNCR01V1 - BlueCoin Cradle board**



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 41. STEVAL-BCNST01V1 - CoinStation board



3.2 Software description

The following software components are required to establish a suitable development environment for creating applications for the STM32 Nucleo equipped with the NFC, sensors, FlightSense and BlueNRG expansion boards:

- FP-SNS-FLIGHT1: a Bluetooth low energy, sensors and NFC tag software for STM32Cube. The FP-SNS-FLIGHT1 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 (SW4STM32) + ST-LINK

3.3 Hardware and software setup

3.3.1 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 NFC expansion board (order code: X-NUCLEO-NFC04A1)
  - one sensor expansion board (order code: X-NUCLEO-IKS01A2 or X-NUCLEO-IKS01A1)
  - one FlightSense expansion board (order code: X-NUCLEO-53L0A1)
  - 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-BCNKT01V1:
  - STEVAL-BCNKT01V1 development kit
  - ST-LINK/V2-1 debugger/programmer integrated onto STM32 Nucleo board
  - one USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
  - one USB type A to Micro-B USB cable to connect the STEVAL-BCNKT01V1 to the PC

3.3.2 Software setup

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

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

3.3.2.1 *Development tool-chains and compilers*

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

3.3.3 System setup guide

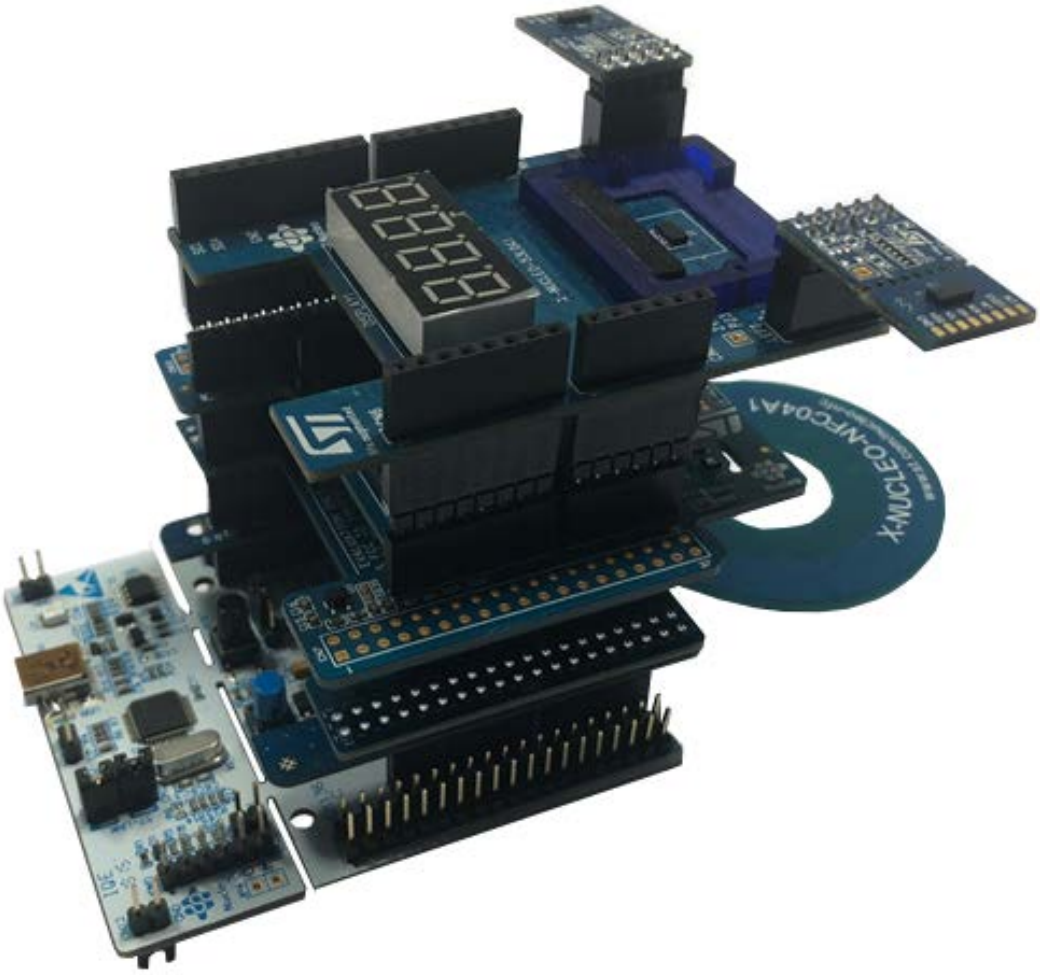
3.3.3.1 *STM32 Nucleo and sensor expansion board setup*

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

Connect the following boards through the Arduino UNO R3 extension connector:

1. The X-NUCLEO-NFC04A1 expansion board on the X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1).
2. The X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1) expansion board on the STM32 Nucleo development board.
3. The X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A1) expansion board on the X-NUCLEO-NFC04A1.
4. The X-NUCLEO-53L0A1 expansion board on the X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A1) expansion.

Figure 42. STM32 Nucleo development board plus X-NUCLEO-IDB05A1, X-NUCLEO-NFC04A1, X-NUCLEO-IKS01A2 and X-NUCLEO-53L0A1 expansion boards



Note: You must connect the boards in the sequence described above to optimize the performance of the SPBTLE-RF module on the X-NUCLEO-IDB05A1 expansion board and to reduce interference from its antenna.

3.3.3.2 *STEVAL-BCNKT01V1 setup*

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

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

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

Be sure that CN2 Jumpers are OFF and connect your STM32 Nucleo board to the BlueCoin Coinstation through the provided cable paying attention to the polarity of the connectors. Pin 1 can be identified by a small circle on the STM32Nucleo board and Coin Station PCB silkscreens or by the square shape of the soldering pad of the connector (SensorTile Cradle).

Figure 43. **BlueCoin-STM32Nucleo: SWD connections with 5-pin flat cable**

3.3.3.3 *Important additional hardware information*

Before connecting the X-NUCLEO-NFC04A1 expansion board through the Arduino UNO R3 extension connector:

- to avoid hardware conflict with the X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A1) expansion board, remove the resistors R2, R3 and R4 in the X-NUCLEO-NFC04A1 board;
- to avoid hardware conflict with the X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1) expansion board, remove the resistors R1 and R11 in the X-NUCLEO-NFC04A1 board.

Figure 44. **X-NUCLEO-NFC04A1 configuration**

Revision history

Table 2. **Document revision history**

Date	Version	Changes
25-Feb-2016	1	Initial release.
05-Dec-2016	2	Updated Section "Introduction", Section 2.1: "Overview ", Figure 4: "Initialization phase", Figure 5: "UART console output when the BLE services are started", Figure 6: "UART console output when a device first connects with the board", Figure 15: "BlueMS (android version) initial page after BLE connection", Figure 1: "FP-SNS-FLIGHT1 software architecture" and Figure 18: "BlueMS (android version) Serial console (stdout/stderr)"; Added Section 2.6: "Flash organization", Section 2.7: "The boot process", Section 2.8: "Firmware-Over-The-Air (FOTA) update" and Section 2.9: "Firmware Over The Air (FOTA) update with BlueMS"
20-Feb-2017	3	Throughout document: - minor text and formatting changes. - added IKS01A2 expansion board compatibility information Added Section 2.6: " The installation process" Updated Section 2.9: "Sample application description", Section 2.10: " Android and iOS sample client application" and Section 2.11: "Firmware-over-the-air (FOTA) update with BlueMS"
31-Mar-2017	4	Updated title, Introduction, Section 2.1: "Overview ", Section 2.2: "Architecture", Section 2.3: "Folder structure", Section 2.6: "The installation process", Section 2.9: "Sample application description", Section 2.10: "Android and iOS sample client application", Section 2.11: "Firmware over-the-air (FOTA) update with BlueMS". Added X-NUCLEO-53L0A1 expansion board compatibility information. Added Section 3.1.8: "X-NUCLEO-53L0A1 expansion board".
03-Sep-2018	5	Updated Introduction, Section 2.1 Overview, Figure 2. FP-SNS-FLIGHT1 software architecture, Section 2.6 The installation process, Section 2.9 Sample application description, Figure 16. BlueMS (android version) main page (after BLE connection), Figure 21. BlueMS (android version) Serial console (stdout/stderr), Section 3.3.1 Hardware setup and Section 3.3.3.1 STM32 Nucleo and sensor expansion board setup. Added Section 3.1.4 X-NUCLEO-NFC04A1 expansion board, Section 3.3.3.2 STEVAL-BCNKT01V1 setup and Section 3.3.3.3 Important additional hardware information.

Contents

List of tables

List of figures