Getting started with the STM32Cube function pack for multi-sensor nodes with signal processing to enable predictive maintenance

Introduction

FP-IND-PREDMNT1 is an STM32Cube function pack including dedicated algorithms for advanced time and frequency domain signal processing and analysis of 3D digital accelerometers with flat bandwidth up to 6 kHz.

The package includes pressure, relative humidity and temperature sensor monitoring, and audio algorithms for acoustic emission (AE), up to 20 kHz, and ultrasound emission analysis up to 80 kHz.

According to the designer's needs for connectivity and sensors, different options are available and selectable by changing parameter settings.

With the NUCLEO-F446RE development board and the STEVAL-STWINKT1B or STEVAL-STWINKT1 kit with BLE connectivity, you can monitor and log the algorithm output and sensor data using the STBLESensor app.

The STEVAL-BFA001V2B evaluation kit allows connecting a PC via USB to monitor and log the algorithm output, sensor data and equipment status.

By using the STEVAL-STWINKT1B or STEVAL-STWINKT1 kit with Wi-Fi connectivity, you can connect the device to the dedicated DSH-PREDMNT web-based dashboard to monitor and log the algorithm output, sensor data and equipment status.

The FP-IND-PREDMNT1, together with the suggested combination of STM32 and ST devices, can be used to develop specific industrial predictive maintenance applications for early detection of warning signs of potential failure.

The software runs on the STM32 microcontroller and includes all the necessary drivers for the STM32 Nucleo development board and expansion boards, as well as for the STEVAL-BFA001V2B and STEVAL-STWINKT1B or STEVAL-STWINKT1 evaluation kits.

Note: In the NUCLEO-F446RE platform, the ISM330DLC ODR is limited to 1.6 kHz when it is mounted on the X-NUCLEO-IKS01A3 and running with MotionSP library in I^2C mode.

RELATED LINKS

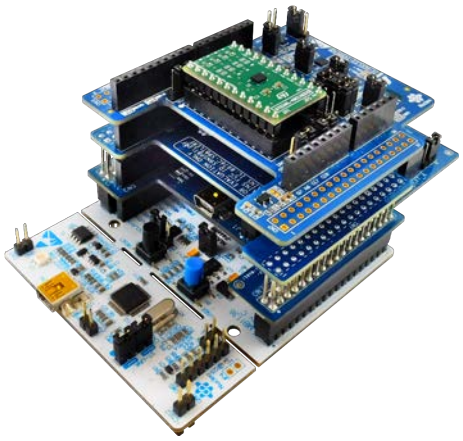
Visit the STM32Cube ecosystem web page on www.st.com for further information

1 FP-IND-PREDMNT software expansion for STM32Cube

1.1 Overview

1.1.1 Compatible hardware platforms

Table 1. Compatible hardware platform overview

Features	Platform		
	STM32 Nucleo	STEVAL-BFA001V2B	STEVAL-STWINKT1B (or STEVAL-STWINKT1)
Image			
Size (cm)	12x7x6	6.3x1x1	5x5x1.5
Vibration analysis	Up to 3.3 kHz	Up to 3.3 kHz	Up to 5 kHz
Sound analysis	Available up to 20 kHz		
Ultrasound analysis	N.A.	N.A.	Available up to 80 kHz
Temperature	Available		
Humidity			
Pressure			
Connectivity	BLE	Serial UART	BLE/Wi-Fi
Datalog and GUI	Mobile app→STBLESensor	User Terminal	Mobile app→STBLESensor
			Mobile Cloud→Web application

RELATED LINKS

2 STM32 Nucleo platform on page 11
3 STEVAL-BFA001V2B platform on page 49
4 STEVAL-STWINKT1 platform on page 59

1.1.2 Firmware

FP-IND-PREDMNT1 key features are:

- Complete firmware to develop a sensor node for predictive maintenance applications, featuring digital or analog microphone, environmental and motion sensors, and performing real-time monitoring of parameters and equipment status via:
  - user terminal by UART (available for the STEVAL-BFA001V2B kit)
  - Wi-Fi connectivity (available for the STEVAL-STWINKT1B or STEVAL-STWINKT1 with the STEVAL-STWINWFV1 Wi-Fi board)
  - BLE connectivity (available for the NUCLEO-F446RE development board and the STEVAL-STWINKT1B or STEVAL-STWINKT1 kit)
- Signal processing (MotionSP) middleware for vibration analysis in time domain (speed RMS and acceleration peak) and frequency domain (FFT with programmable size, averaging, overlapping and windowing)
- PDM to PCM and sound pressure level (SPL) middleware for digital microphones (for NUCLEO-F446RE and STEVAL-BFA001V2B)
- Acoustic FFT Analysis for NUCLEO-F446RE and STEVAL-BFA001V2B
- Ultrasound FFT Analysis for the STEVAL-STWINKT1B or STEVAL-STWINKT1
- Configurable alarm and warning thresholds for key parameters
- Data logging capability through mobile app or PC serial terminal
- Compatible with STBLESensor application for Android/iOS, to perform sensor data reading, audio and motion algorithm feature demo, and firmware update over the air (FOTA) (feature available only when using NUCLEO-F446RE and STEVAL-STWINKT1B or STEVAL-STWINKT1 with BLE connectivity)
- Compatible with DSH-PREDMNT web-based predictive maintenance dashboard for monitoring sensor data and device status (feature available only when using the STEVAL-STWINKT1B or STEVAL-STWINKT1 with the STEVAL-STWINWFV1 Wi-Fi adapter)
- Sample implementation available for the STEVAL-BFA001V2B and STEVAL-STWINKT1B or STEVAL-STWINKT1 kits
- Sample implementation available for X-NUCLEO-CCA02M2, X-NUCLEO-IKS01A3, STEVAL-MKI182V2 (DIL24 based on ISM330DLC) and X-NUCLEO-BNRG2A1 connected to a NUCLEO-F446RE board
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

The software gathers:

- the temperature, humidity, pressure, audio and motion sensor drivers for the HTS221, LPS22HB, MP34DT06J, ISM330DLC and LSM303AGR devices for STM32 Nucleo
- the temperature, humidity, pressure, audio, 3-axis digital vibration sensor drivers for the HTS221, LPS22HB, MP34DT05-A and IIS3DWB devices for the STEVAL-BFA001V2B running on STM32
- the temperature, humidity, pressure, audio, 3-axis digital vibration, motion sensor drivers for the HTS221, LPS22HH, IMP34DT05 and IMP23ABSU (or MP23ABS1), IIS3DWB, ISM330DHCX devices for the STEVAL-STWINKT1B (or STEVAL-STWINKT1)

For NUCLEO-F446RE and STEVAL-STWINKT1B (or STEVAL-STWINKT1) boards, the package is compatible with the STBLESensor Android/iOS (Ver. 4.2.1 or higher) applications, available at the respective store, which can be used to display information sent via BLE and for Over-The-Air firmware update.

1.2 Architecture

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

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

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

STM32Cube is designed by STMicroelectronics to reduce development effort, time and cost across the entire STM32 portfolio.

STM32Cube version 1.x includes:

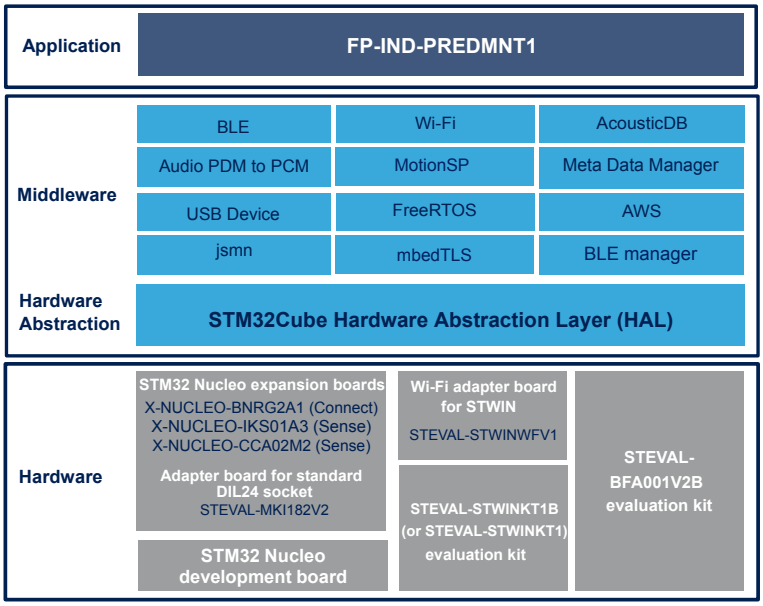
- STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards.
- A comprehensive embedded software platform specific to each series (such as the STM32Cube for the STM32 series), which includes:
  - the STM32Cube HAL embedded abstraction-layer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

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

The **STM32Cube HAL** driver layer, which provides a simple, generic, multi-instance set of application programming interfaces (APIs) to interact with the upper application, library and stack layers. It has generic and extension APIs and is directly built around a generic architecture and allows successive layers like the middleware layer to implement functions without requiring specific hardware configurations for a given microcontroller unit (MCU). This structure improves library code reusability and facilitates portability to other devices.

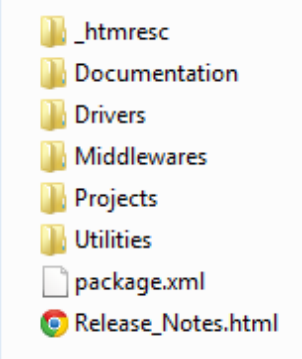
The **board support package** (BSP) layer supports all the peripherals on the STM32 Nucleo except the MCU. This limited set of APIs provides a programming interface for certain board-specific peripherals like the LED, the user button, etc. This interface also helps in identifying the specific board version.

Figure 1. FP-IND-PREDMNT1 software architecture



1.3 Folder structure

Figure 2. **FP-IND-PREDMNT1 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 and 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 ARM Cortex-M processor series.
- **Middlewares**: contains libraries and protocols for BlueNRG-2 Bluetooth low energy, AcousticDB and Audio PDM/PCM software library, MotionSP software library, Meta Data Manager to save meta data in the Flash memory, FreeRTOS real-time operating system, USB Device Library and STM32 Connect Library which provides API to access network services on STM32 devices, Parson library, AWS library
- **Projects**: contains sample applications used for transmitting the output of the sensor data and the analysis results for all the platforms in different folders:
  - STM32F446RE-Nucleo: motion signal processing and audio over BLE protocol
  - STM32F469AI-BFA001V2: motion signal processing and audio using user terminal by UART
  - STM32L4R9ZI-STWIN: motion processing and ultrasound analysis over BLE and Wi-Fi protocol

  All projects are available for IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM-STR) and Integrated Development Environment for STM32 (STM32CubeIDE).
- **Utilities**: contains the boot loader binary ready to be flashed for the NUCLEO-F446RE development board and the STEVAL-STWINKT1B (or STEVAL-STWINKT1) evaluation kit.

1.4 Flash management

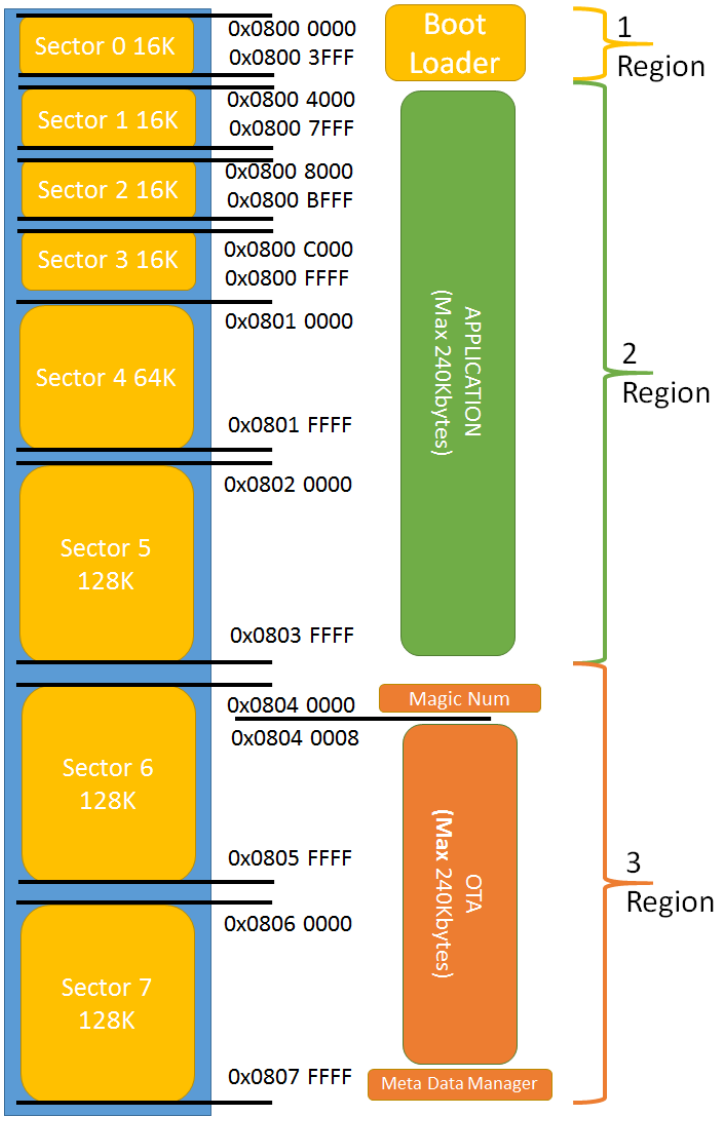
For NUCLEO-F446RE and STEVAL-STWINKT1B (or STEVAL-STWINKT1), the FP-IND-PREDMNT1 also uses the Flash memory for Firmware-Over-The-Air updates.

As shown in the figure below, it is divided into:

1. the first region, containing a custom boot loader
2. the second region, containing the FP-IND-PREDMNT1 firmware
3. the third region, used for storing the FOTA before the update

The Meta Data Manager is placed at the end of the Flash memory (0x0807FFFF for NUCLEO-F446RE and 0x081FF000 for STEVAL-STWINKT1B (or STEVAL-STWINKT1)).

Figure 3. **FP-IND-PREDMNT1 Flash structure for NUCLEO-F446RE**



1.5 The boot process

The FP-IND-PREDMNT1 cannot be flashed at the beginning of the Flash memory (address 0x08000000), and is therefore compiled to run from the beginning of the second Flash region, at 0x08004000.

To enable this behavior, the vector table offset in:

- *Src/system_stm32f4xx.c* (for STM32F446RE)
- *Src/system_stm32l4xx.c* (for STM32L4R9ZI)

has been set by `#define VECT_TAB_OFFSET 0x4000`.

The linker script has been changed too.

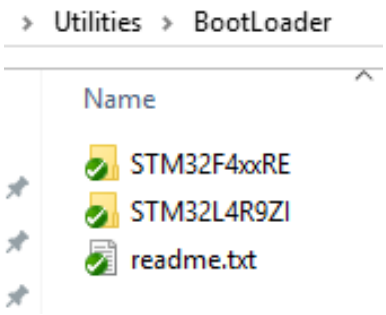
For example, for IAR Embedded Workbench for ARM for NUCLEO-F446RE, the linker script is:

```
define symbol   ICFEDIT_intvec_start_  = 0x08004000;
/*-Memory Regions-*/
define symbol   ICFEDIT_region_ROM_start = 0x08004000;
define symbol   ICFEDIT_region_ROM_end   = 0x0803FFFF;
```

The maximum usable code size is fixed at 240 KB for NUCLEO-F446RE and 1024 KB for STEVAL-STWINKT1B (or STEVAL-STWINKT1).

You must flash the appropriate bootloader binary for STM32F446RE and STM32L4R9ZI, found in the Utilities \BootLoader folder, to the first Flash region (address0x08000000).

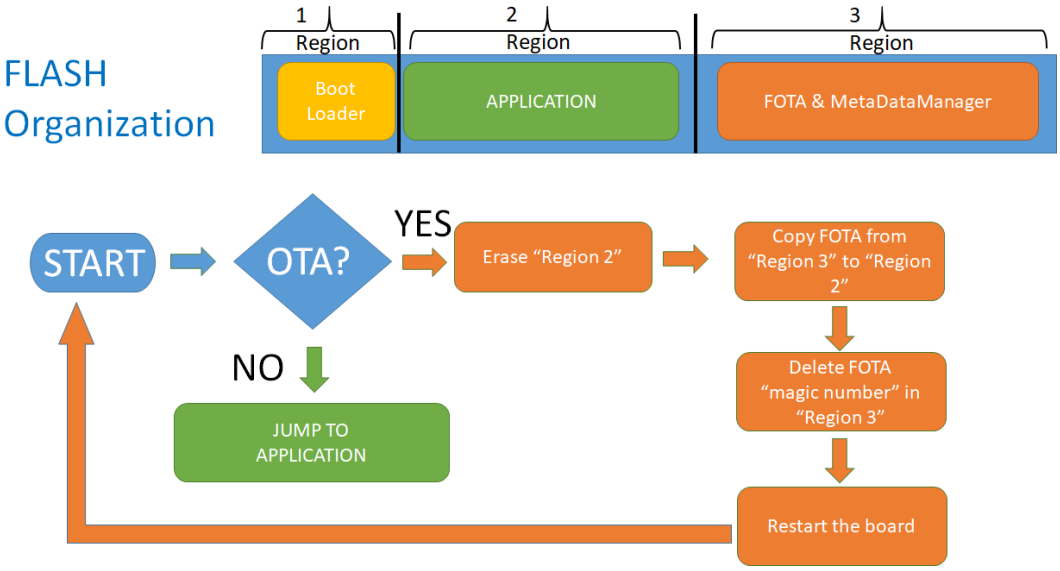
Figure 4. Bootloader folder content



At any board reset:

- if there is a FOTA in the third Flash region, the bootloader overwrites the second Flash region (with FP-IND-PREDMNT1 firmware) and replaces its content with the FOTA, restarting the board;
- if there is no FOTA, the bootloader jumps to the FP-IND-PREDMNT1 firmware.

Figure 5. FP-IND-PREDMNT1 boot sequence



For each IDE (IAR/RealView/STM32CubeIDE) and demonstration, a batch script has been provided to simplify this operation by saving the firmware and the Bootloader to the right position:

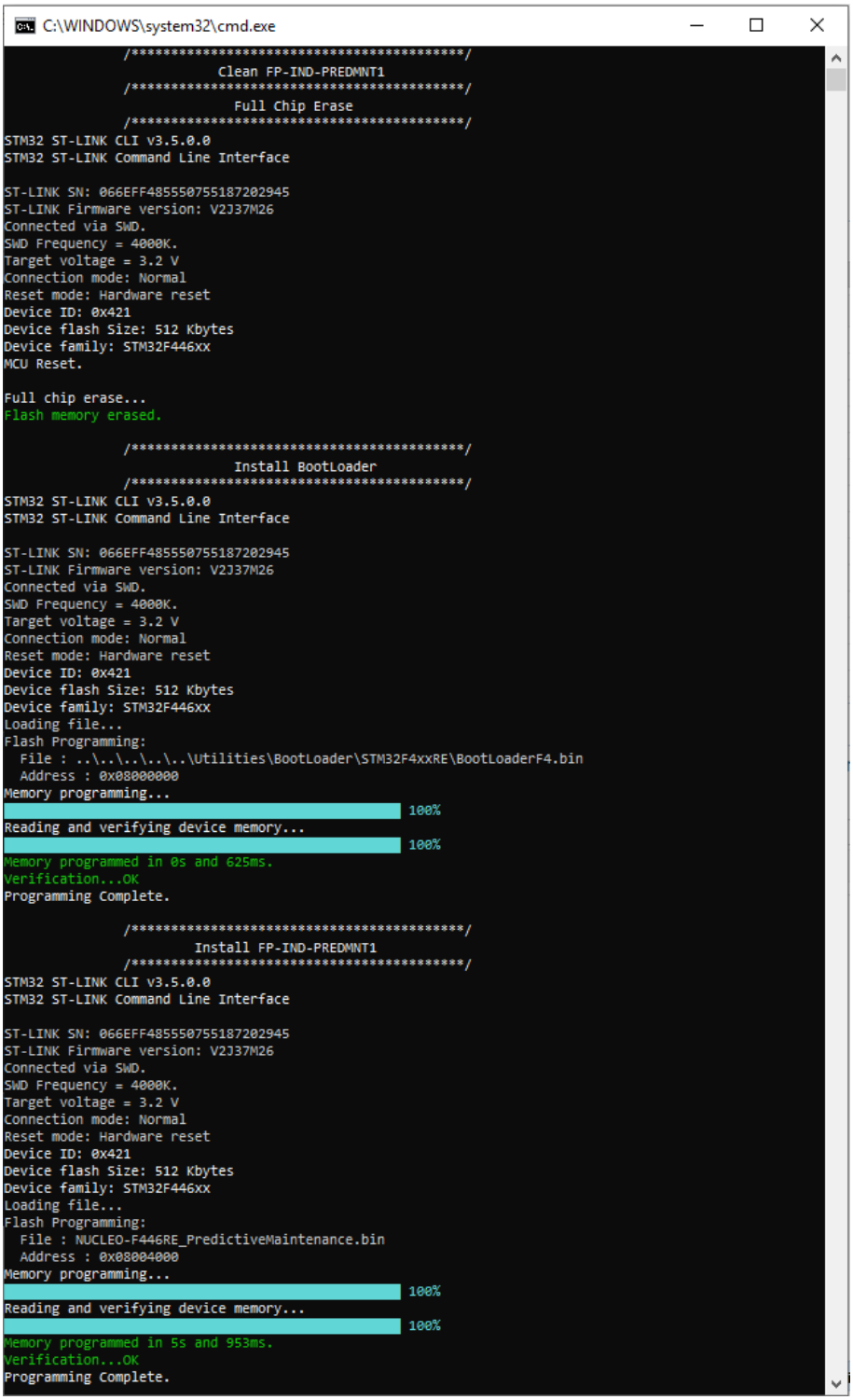
- NUCLEO-F446RE
  - `CleanPREDMNT1_IAR_F446`
  - `CleanPREDMNT1_MDK-ARM_F446.bat`
  - `CleanPREDMNT1_SW4STM32_F446 .bat`
- STEVAL-STWINKT1 (for BLE project only)
  - `CleanPREDMNT1_IAR_STWIN.bat`
  - `CleanPREDMNT1_MDK-ARM_STWIN.bat`

1.5.1 Batch script

The batch script:

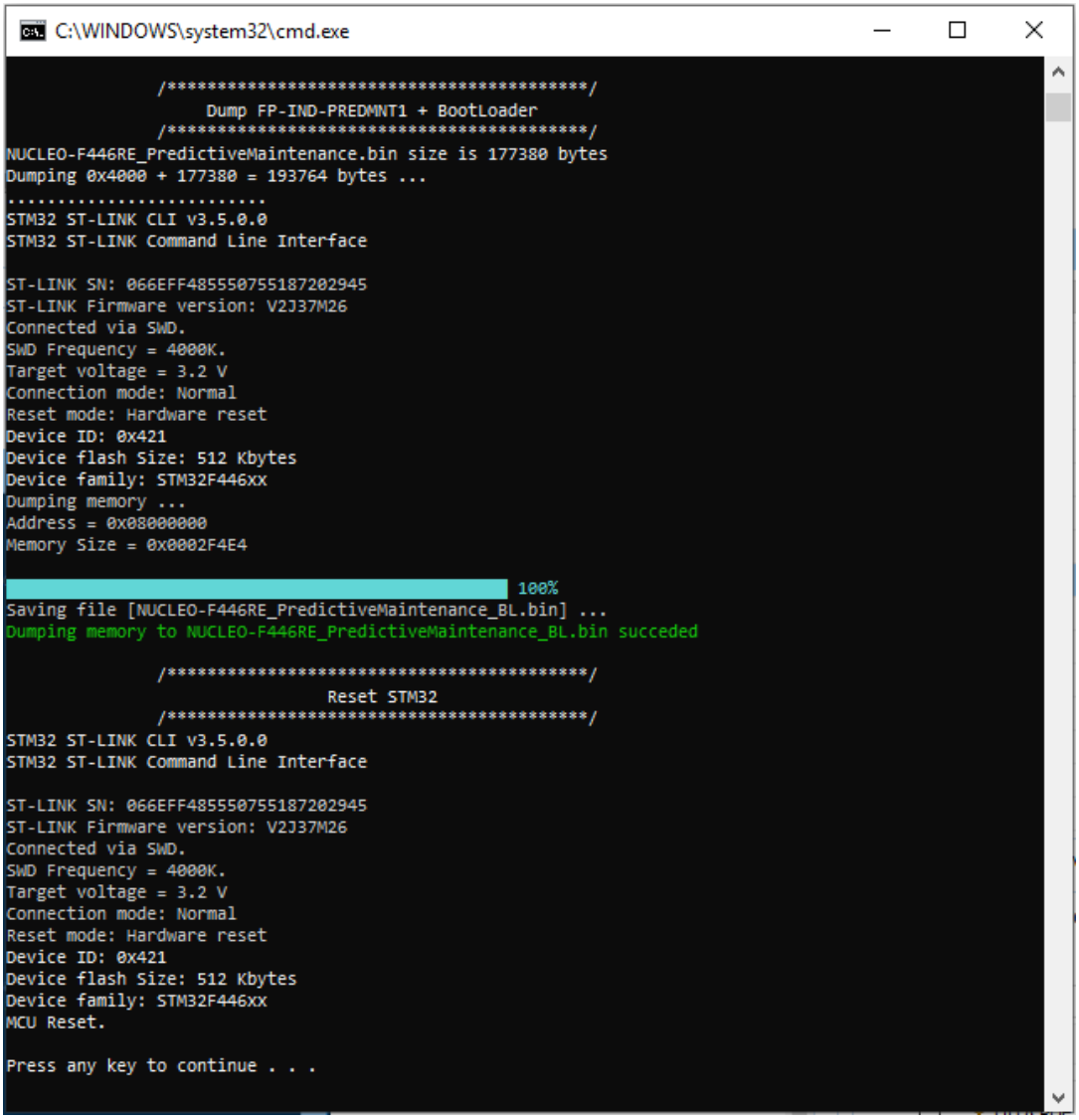
- performs a full Flash erase to start from a clean system
- flashes the Bootloader to the correct position (0x08000000)
- flashes the firmware to the correct position (0x08004000)

Figure 6. Bootloader and FP-IND-PREDMNT1 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 as the image provided in the Binary folder.

Figure 7. **FP-IND-PREDMNT1 dump process**



For Linux or iOS operating systems, there is a similar script that uses OpenOCD instead of the ST-LINK command line and is included in STM32CubeIDE only (*CleanPREDMNT1_STM32CubeIDE_F446.sh*).

To function, the script must be modified with:

- the installation path for OpenOCD
- the installation path for STM32 OpenOCD scripts
- the library path for OpenOCD

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

```
######## Modify this Section:
# 1) Set the Installation path for OpenOCD
# example:
#OpenOCD_DIR="C:/ST/STM32CubeIDE_1.4.0/STM32CubeIDE/plugins/
com.st.stm32cube.ide.mcu.externaltools.openocd.win32_1.4.0.202007081208/tools"
OpenOCD_DIR=""

# 2) Set the installation path for stm32 OpenOCD scritps
# example:
#OpenOCD_CFC="C:/ST/STM32CubeIDE_1.4.0/STM32CubeIDE/plugins/
com.st.stm32cube.ide.mcu.debug.openocd_1.4.2.202008061104/resources/openocd/st_scripts"
OpenOCD_CFC=""

# 3) Only for Linux/iOS add openocd library path to _LIBRARY_PATH:
# For iOS example:
#export DYLD_LIBRARY_PATH=${DYLD_LIBRARY_PATH}:${OpenOCD_DIR}"lib/"

# For Linux example:
#export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${OpenOCD_DIR}"lib/"
```

1.6 Firware-Over-The-Air (FOTA) update

The FP-IND-PREDMNT1 firmware can be updated Over-The-Air (FOTA) through an Android/iOS device via Bluetooth using the STBLESensor application (ver. 4.2.1 and above).

The application sends the update and associated cyclic redundancy check (CRC) value, which the FP-IND-PREDMNT1 checks against the hardware cyclic redundancy check calculation unit on the STM32F446REor STM32L4R9ZI processor to ensure integrity. If the CRC calculation matches the STBLESensor CRC value, the new firmware is written at the beginning of the third Flash region. A “magic number” signals the bootloader that a firmware update has been received and checked, and is ready to replace the current FP-IND-PREDMNT1 firmware (refer to Section 2.10.8 Firmware-Over-The-Air update with STBLESensor).

1.7 APIs

Detailed technical information with full user API function and parameter description are in a compiled HTML file in the “Documentation” folder.

2 STM32 Nucleo platform

The STM32 Nucleo platform compatible with the FP-IND-PREDMNT1 function pack consists of:

- one STM32 Nucleo development board (order code: NUCLEO-F446RE)
- one microphone expansion board (order code: X-NUCLEO-CCA02M2)
- one sensor expansion board (order code: X-NUCLEO-IKS01A3)
- one DIL24 based on ISM330DLC (order code: STEVAL-MKI182V2)
- one BlueNRG Bluetooth low energy expansion board (order code: X-NUCLEO-BNRG2A1)
- one USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC

RELATED LINKS

1.1.1 Compatible hardware platforms on page 2

2.1 STM32 Nucleo

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

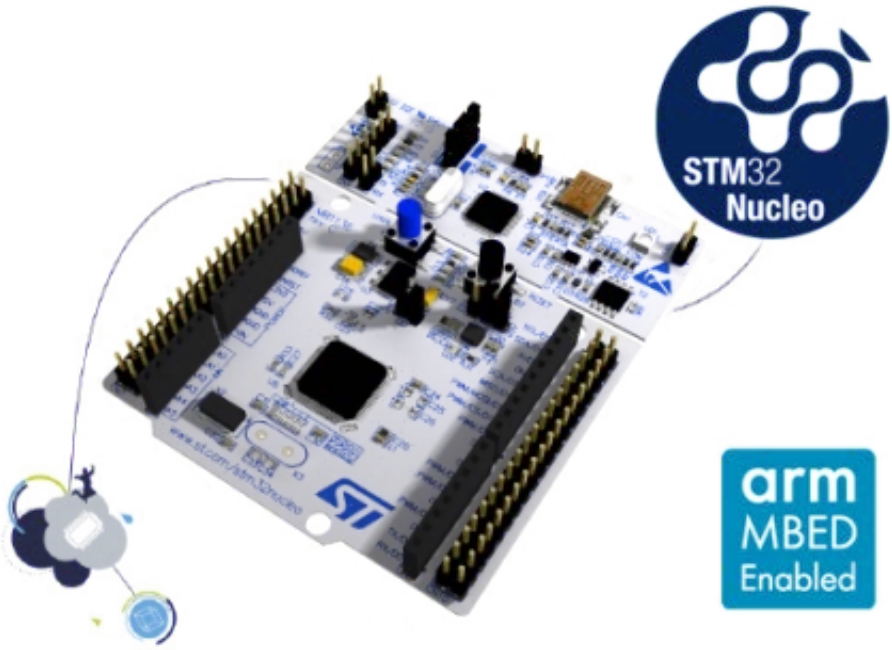
The Arduino connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

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

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

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

Figure 8. STM32 Nucleo board



2.2 X-NUCLEO-CCA02M2 expansion board

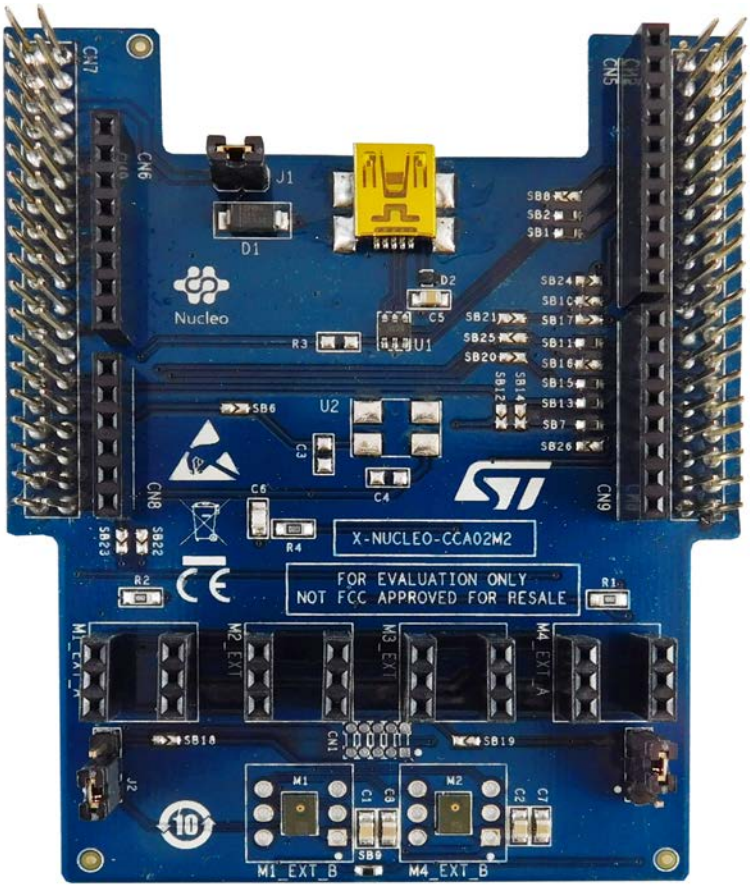
The X-NUCLEO-CCA02M2 expansion board has been designed around MP34DT06J digital MEMS microphone.

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

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

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

Figure 9. X-NUCLEO-CCA02M2 expansion board



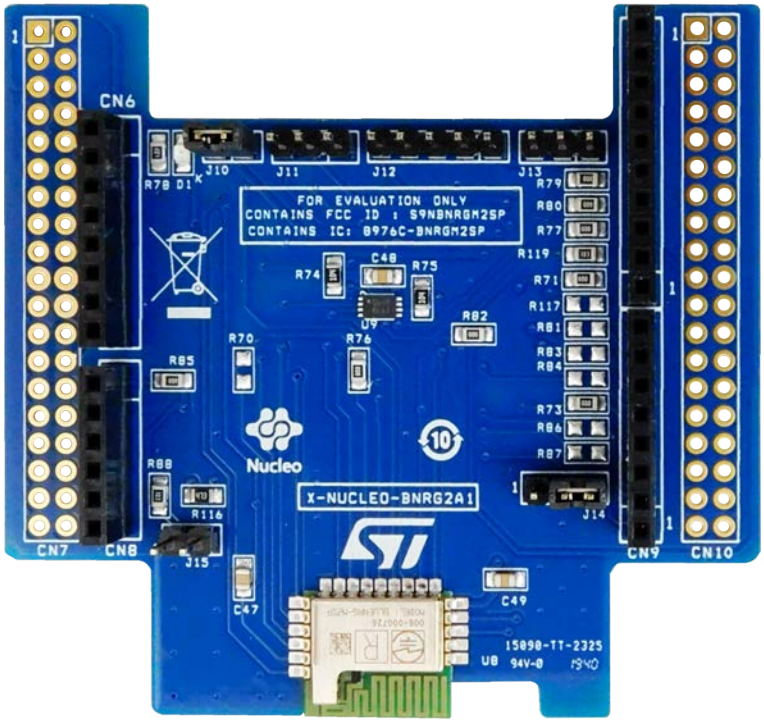
2.3 X-NUCLEO-BNRG2A1 expansion board

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

The expansion board features the Bluetooth® v5.2 compliant and FCC certified BlueNRG-M2SP application processor module based on the ST BlueNRG-2 System-on-Chip. This SoC manages the complete Bluetooth low energy stack and protocols on its Cortex-M0 core and programmable Flash, which can accommodate custom applications developed using the SDK. The BlueNRG-M2SP module supports master and slave modes, increased transfer rates with data length extension (DLE), and AES-128 security encryption.

The X-NUCLEO-BNRG2A1 interfaces with the STM32 Nucleo microcontroller via SPI connections and GPIO pins, some of which can be configured by the hardware.

Figure 10. **X-NUCLEO-BNRG2A1 BLE expansion board**



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

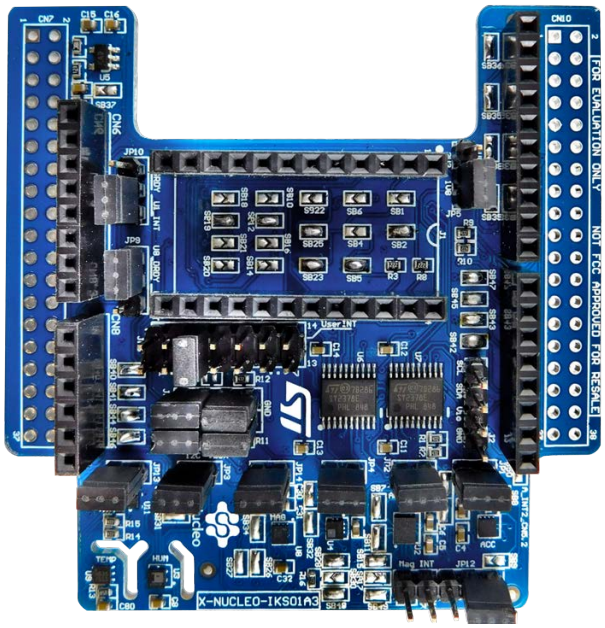
2.4 X-NUCLEO-IKS01A3 expansion board

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

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

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

Figure 11. **X-NUCLEO-IKS01A3 MEMS and environmental sensor expansion board**



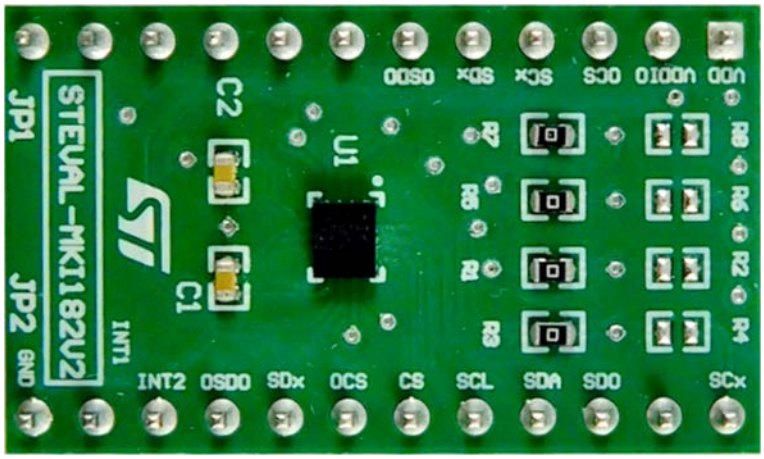
2.5 STEVAL-MKI182V2 evaluation board

The STEVAL-MKI182V2 is an adapter board designed to facilitate the evaluation of MEMS devices in the ISM330DLC product family. The board offers an effective solution for fast system prototyping and device evaluation directly within the user's own application.

The STEVAL-MKI182V2 can be plugged into a standard DIL 24 socket. The adapter provides the complete ISM330DLC pin-out and comes ready-to-use with the required decoupling capacitors on the VDD power supply line.

This adapter is supported by the STEVAL-MKI109V2 and STEVAL-MKI109V3 motherboards, which include a high performance 32-bit microcontroller functioning as a bridge between the sensor and a PC, on which it is possible to use the downloadable graphical user interface (Unico GUI), or dedicated software routines for customized applications.

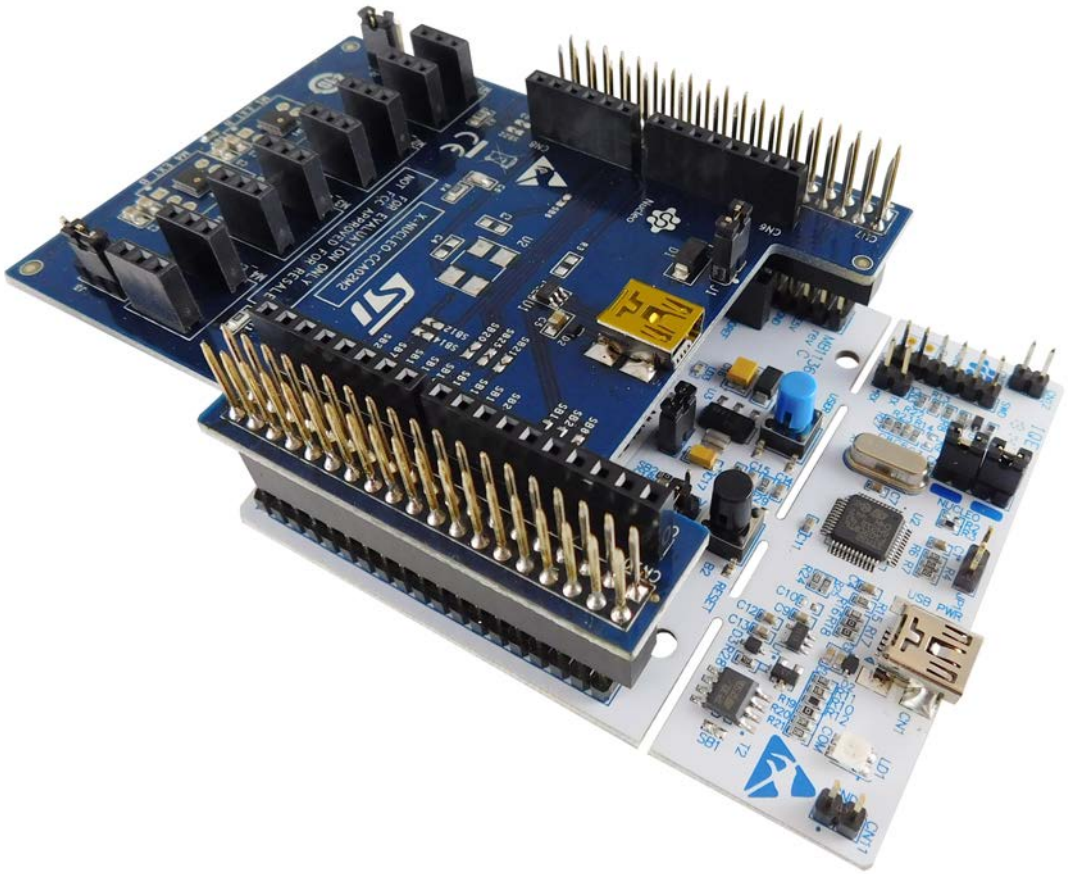
Figure 12. **STEVAL-MKI182V2 evaluation board**



2.6 Hardware setup

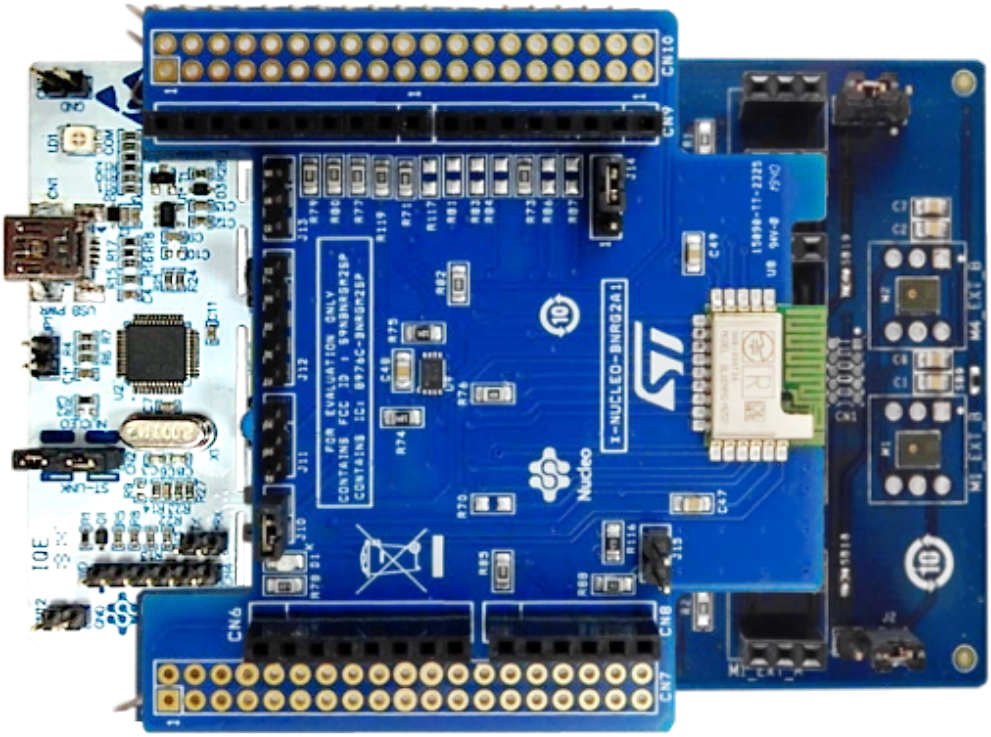
The X-NUCLEO-CCA02M2 sensor board is easily connected to the STM32 Nucleo board through the morpho connector, as shown below.

Figure 13. STM32 Nucleo plus X-NUCLEO-CCA02M2 expansion board



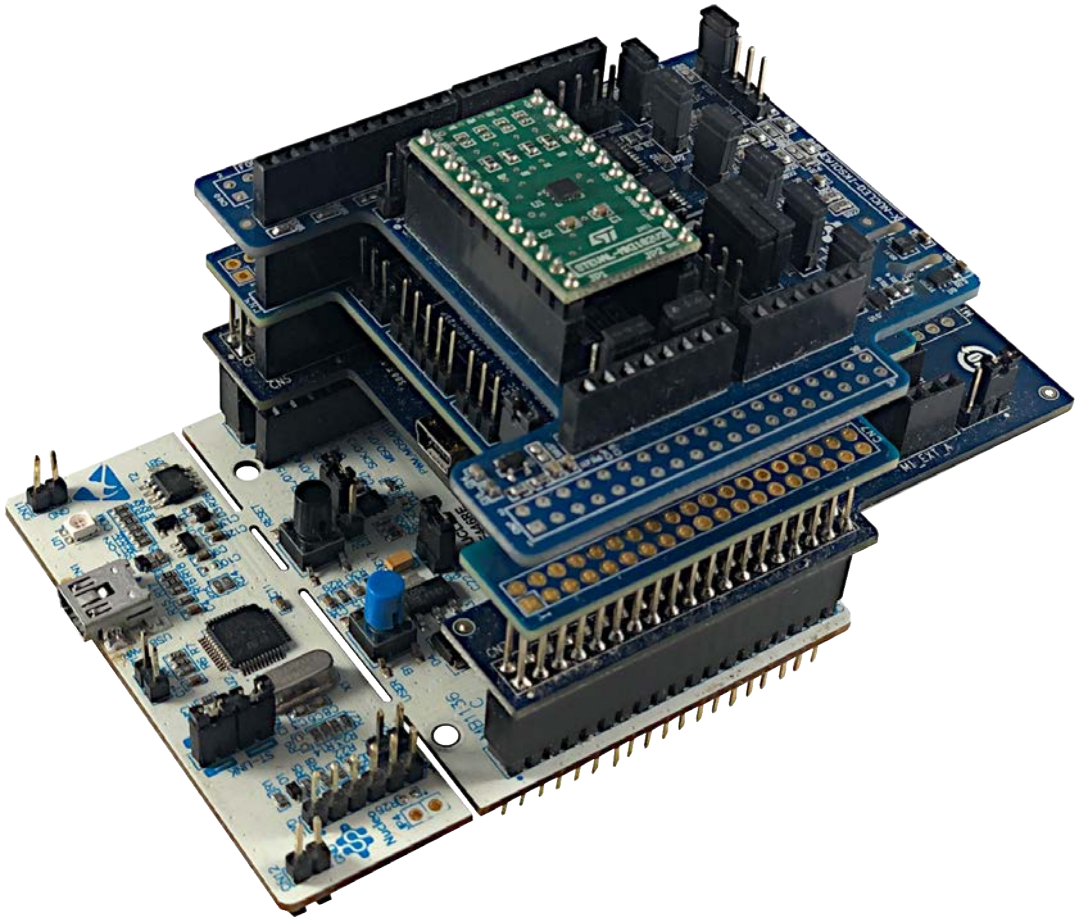
The X-NUCLEO-BNRG2A1 BLE expansion board is easily connected to the X-NUCLEO-CCA02M2 board through the Arduino UNO R3 extension connector, as shown below.

Figure 14. **STM32 Nucleo plus X-NUCLEO-CCA02M2 and X-NUCLEO-BNRG2A1 expansion boards**



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

Figure 15. **STM32 Nucleo plus X-NUCLEO-CCA02M2, X-NUCLEO-BNRG2A1 and X-NUCLEO-IKS01A3 expansion boards plus STEVAL-MKI182V2 evaluation board**



2.6.1 X-NUCLEO-IKS01A3 setup

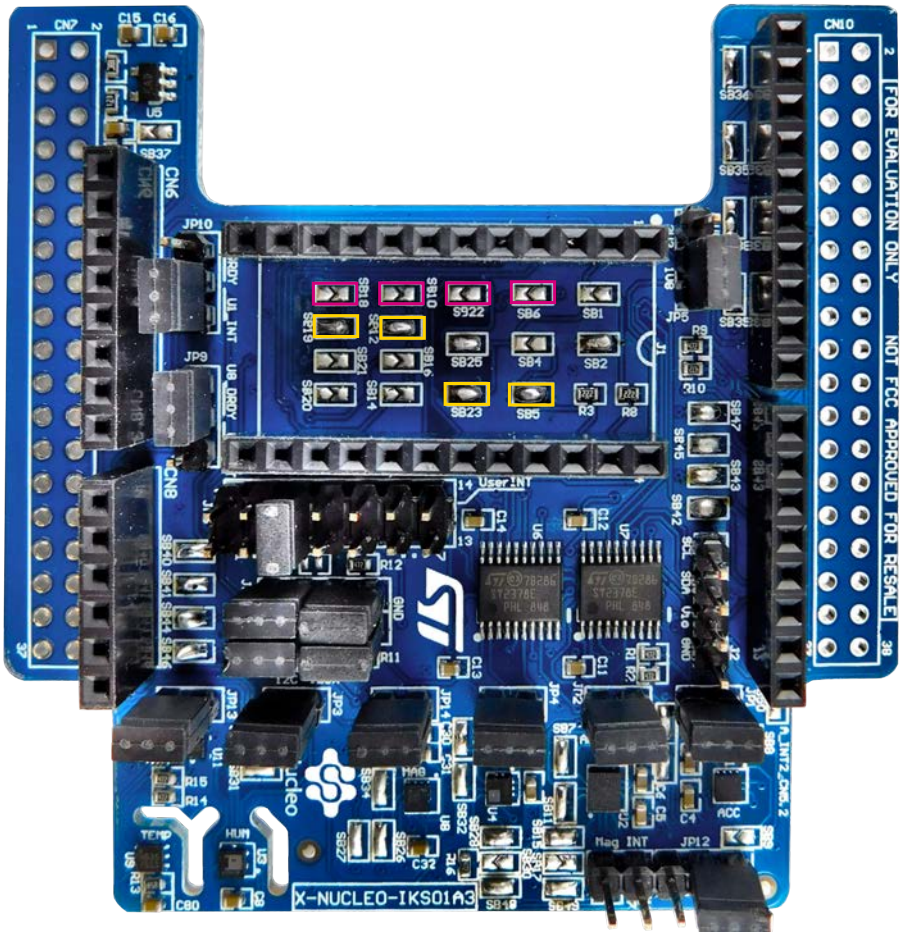
Only when using the X-NUCLEO-IKS01A3 expansion board, you can select I^2C or SPI connection mode for the DIL24 socket by modifying the `iks01a3_conf.h USE_SPI_FOR_DIL24` define with:

- 0 → for I^2C connection
- 1 → for SPI connection

To enable the SPI support on the DIL24 socket, on the X-NUCLEO-IKS01A3:

- open SB5, SB12, SB19 and SB23 solder bridges
- close SB6, SB10, SB18 and SB22 solder bridges

Figure 16. **X-NUCLEO-IKS01A3 expansion board - solder bridges to open (in yellow) and to close (in pink)**



2.6.2 X-NUCLEO-BNRG2A1 setup

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

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

Step 2. Update the X-NUCLEO-BNRG2A1 BLE module through the ST-LINK/V2-1 and the STSW-BNRGFLASHER Flasher utility.

Step 3. Connect the X-NUCLEO-BNRG2A1 J12 pins to the ST-LINK/V2-1 pins through the 5-wire cable included as shown in the picture below and detailed in Table 2.

Figure 17. Connection between X-NUCLEO-BNRG2A1 expansion board and ST-LINK/V2-1

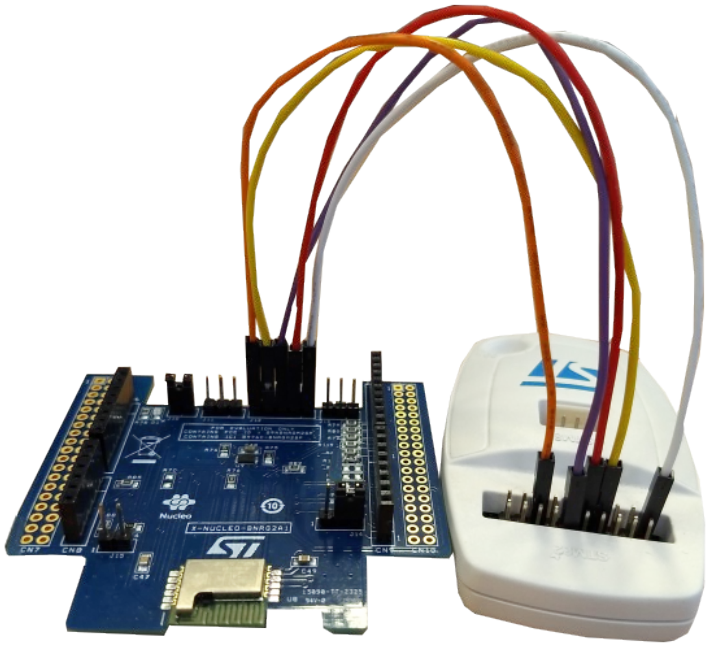


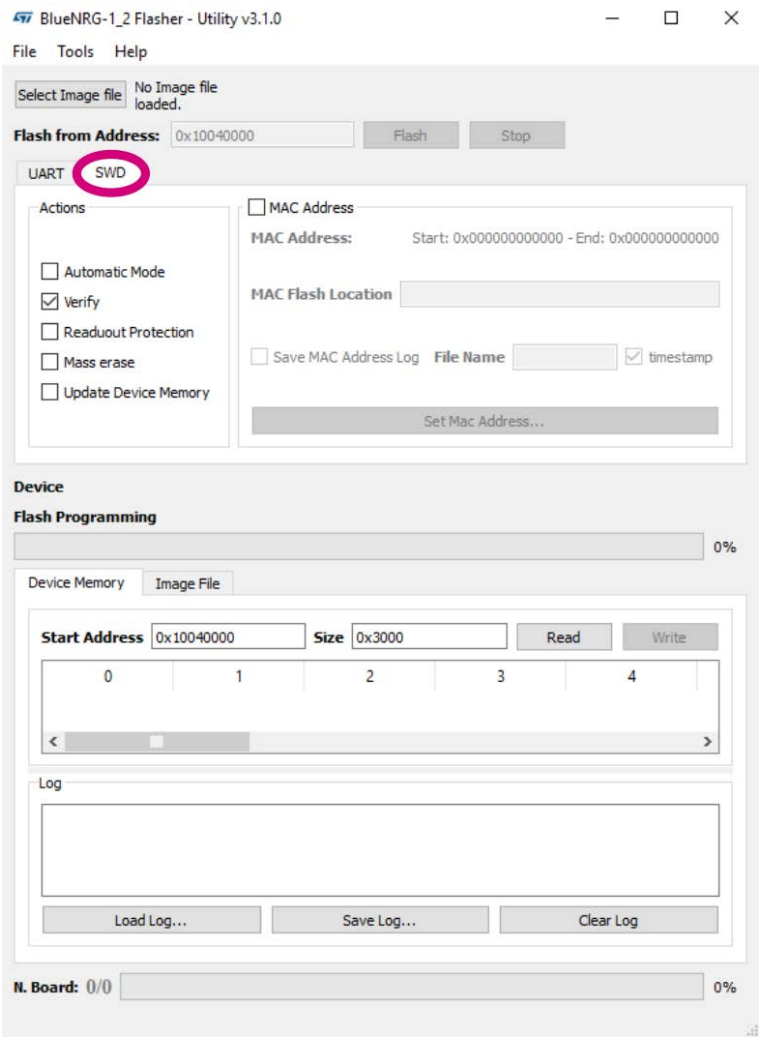
Table 2. X-NUCLEO-BNRG2A1 and ST-LINK/V2-1 pin connections

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

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

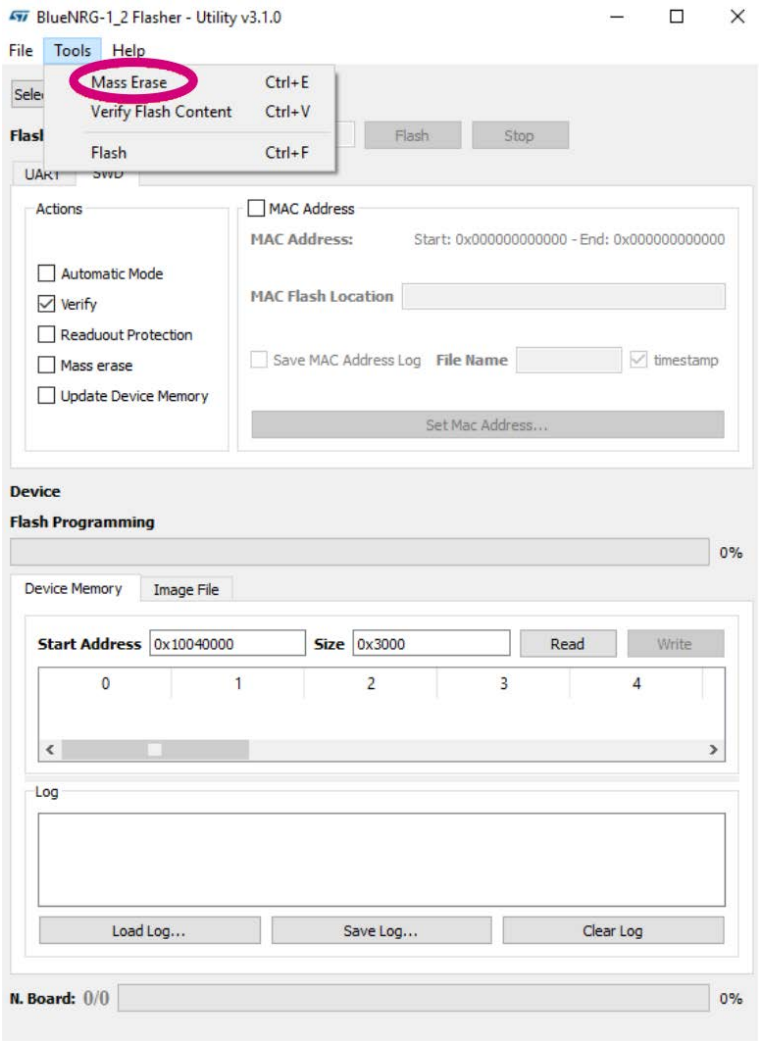
Step 5. Select the SWD tab.

Figure 18. STSW-BNRGFLASHER - SWD tab



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

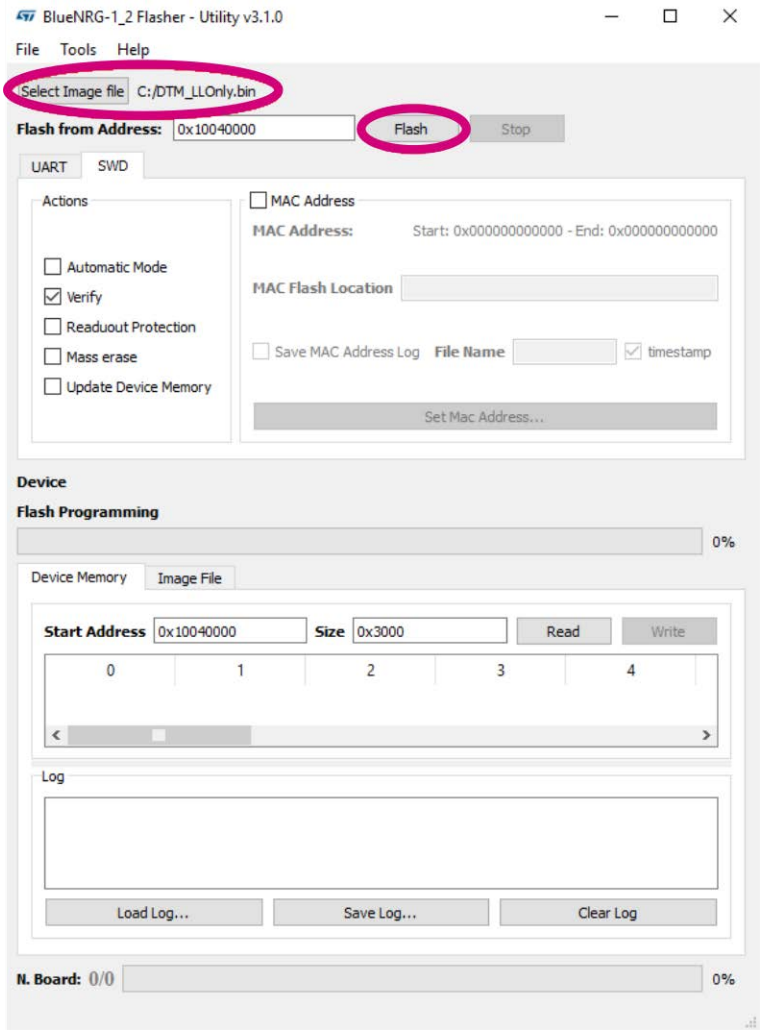
Figure 19. STSW-BNRGFLASHER - mass erase



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

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

Figure 20. STSW-BNRGFLASHER - Flash button



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

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

2.7 Software setup

The following software components are required for the setup of a suitable development environment to create applications for the STM32 Nucleo board with the sensors, microphones, Bluetooth low energy expansion boards:

- FP-IND-PREDMNT1 firmware, available on www.st.com.
- Development tool-chain and Compiler. The STM32Cube expansion software supports the following environments to select from:
  - IAR Embedded Workbench for ARM® (EWARM release 8.50.5) toolchain plus ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM release 5.31.0) toolchain plus ST-LINK
  - Integrated Development Environment for STM32 (STM32CubeIDE release 1.4.2) plus ST-LINK

2.8 The installation process

For each application folder there is a package binary directory that contains:

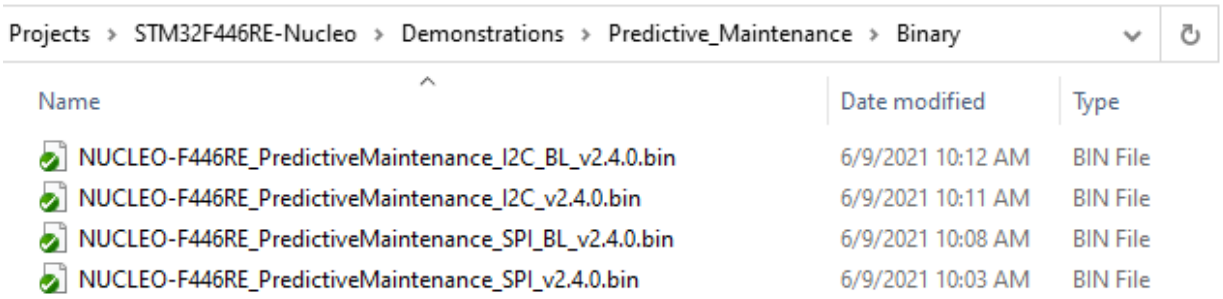
- pre-compiled PREDMNT1 firmware that can be flashed with ST-LINK to the correct memory address (0x08004000) of a supported STM32 Nucleo development board.

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

- pre-compiled PREDMNT1 plus BootLoader firmware that can be directly flashed to a supported STM32 Nucleo development board with the ST-LINK or by drag and drop.

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

Figure 21. Binary folder container



RELATED LINKS

1.5.1 Batch script on page 7
1.6 Firware-Over-The-Air (FOTA) update on page 10

2.9 Sample application description

Several sample applications are provided in the *Project* directory for:

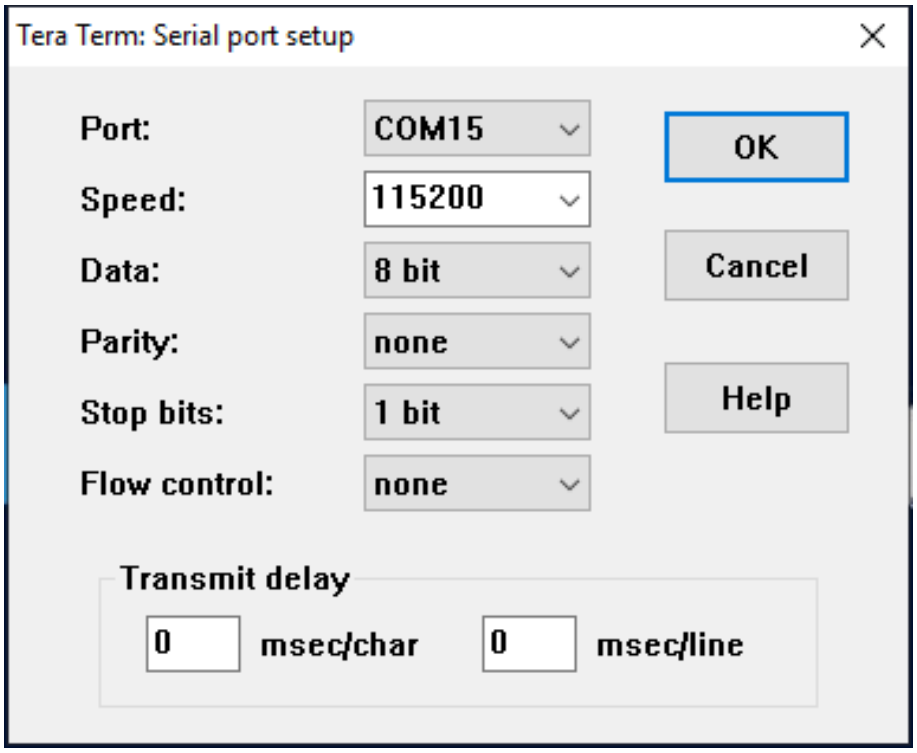
- the X-NUCLEO-IKS01A3, STEVAL-MKI182V2 (DIL24 based on ISM330DLC), X-NUCLEO-CCA02M2 and X-NUCLEO-BNRG2A1 expansion boards connected to the NUCLEO-F446RE development board

Ready to be built projects are available for multiple IDEs (IAR 8.50.5, Keil 5.31.0 and STM32CubeIDE 1.4.2).

2.9.1 Terminal window setup

For the NUCLEO-F446RE development board you can set a terminal window up for the appropriate UART communication port used to control the initialization phase and the execution phases.

Figure 22. **NUCLEO-F446RE terminal setup**



2.9.2 Sample application startup

For the NUCLEO-F446RE and for each demonstration, when you first press the reset button, the application:

- starts initializing the UART
- checks whether all the sensors are present and working
- reads the Meta Data Manager
- checks if the BlueNRG expansion board is connected to the STM32 Nucleo board as well as the hardware and firmware version information
- initializes the BLE stack
- initializes the BLE hardware service (adding the temperature, humidity, pressure, 3D gyroscope,3D magnetometer, 3D accelerometer and microphone characteristics) and the BLE software service
- initializes the BLE console service adding the stdin/stdout and stderr characteristics
- shows the vibration parameter values present in the Flash memory (for Predictive_Maintenance Demonstrations) and sets the accelerometer parameters (for Predictive_Maintenance Demonstrations only).

Figure 23. **Initialization phase**

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

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

When an Android/iOS device is connected to the NUCLEO-F446RE board, you can control data transmitted through the board.

Figure 24. UART console output when a device is connected to the board

2.10 Android and iOS STBLEsensor client application

Developed to reduce power consumption, the FP-IND-PREDMNT1 software for STM32Cube is compatible with the STBLESensor Android/iOS applications (ver. 4.2.1or higher).

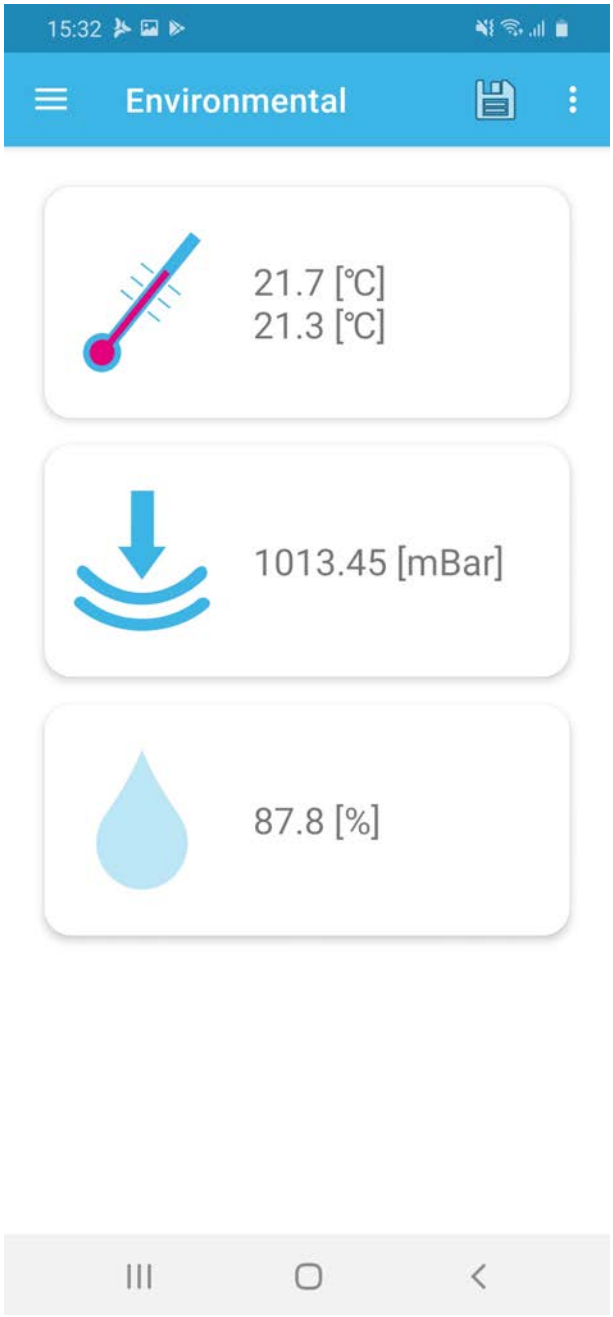
The STBLESensor application allows Over-The-Air firmware updates.

The Android version is used for this demonstration.

2.10.1 Main page

After connection, STBLESensor opens the main page shown below, where the values of temperature, pressure and humidity are displayed.

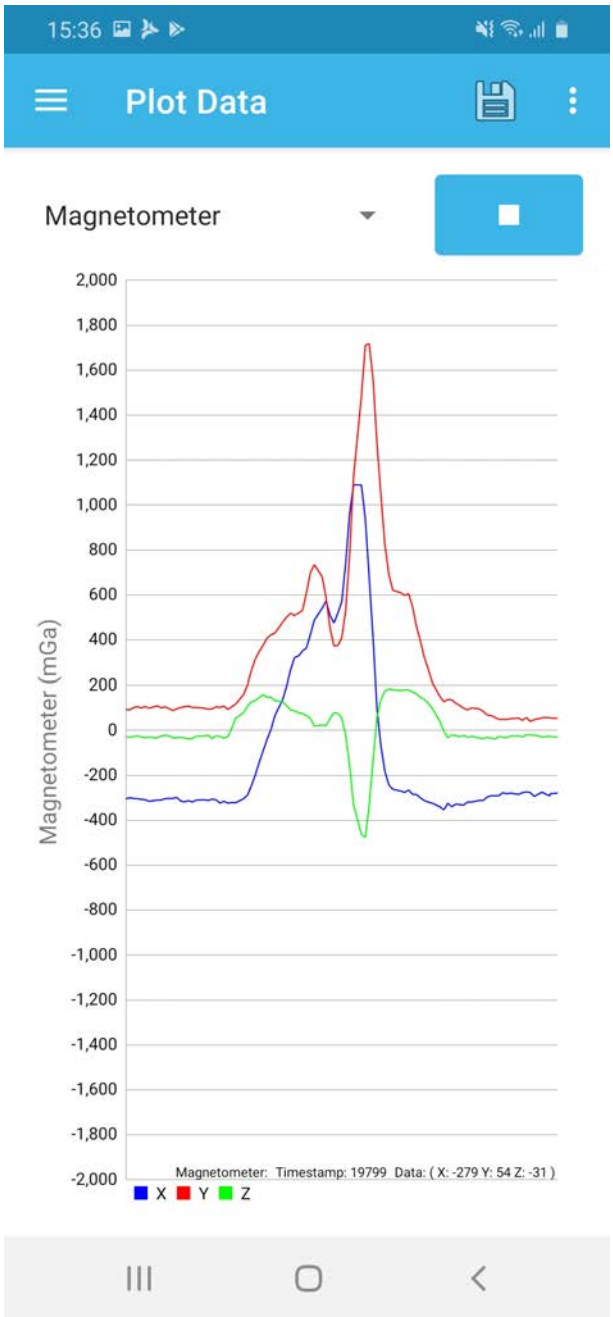
Figure 25. STBLESensor (Android version) main page following BLE connection



2.10.2 Plot data

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

Figure 26. STBLESensor (Android version) magnetometer plot

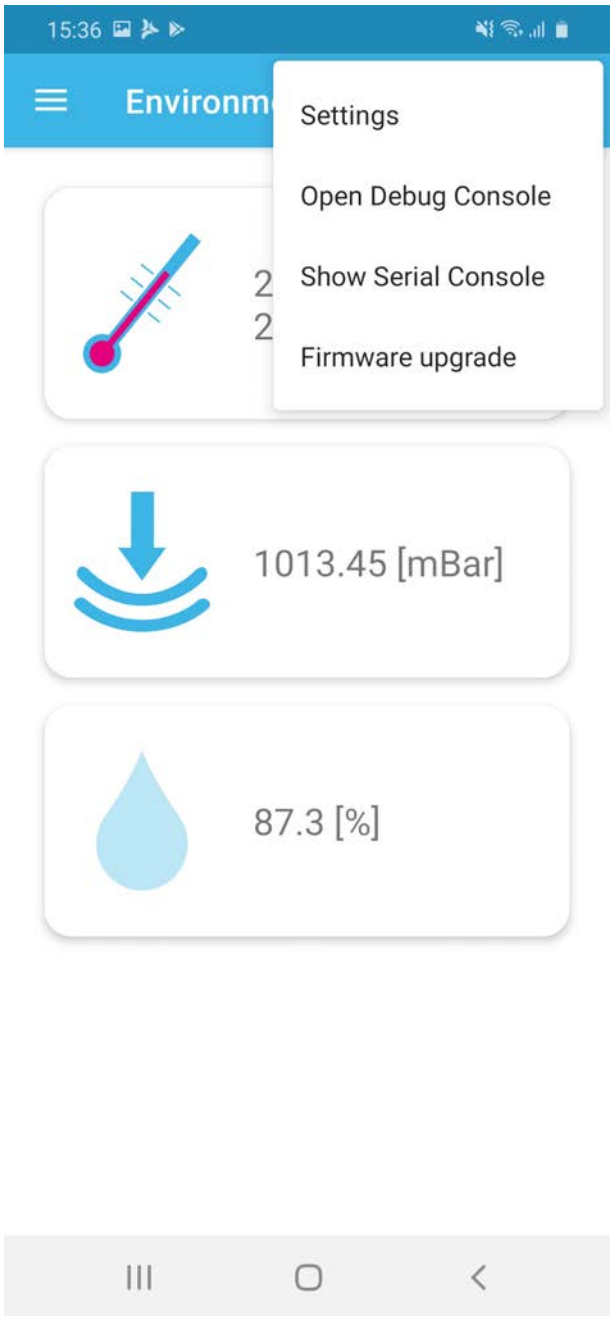


2.10.3 Settings, serial and debug console

In the option menu below, you can open:

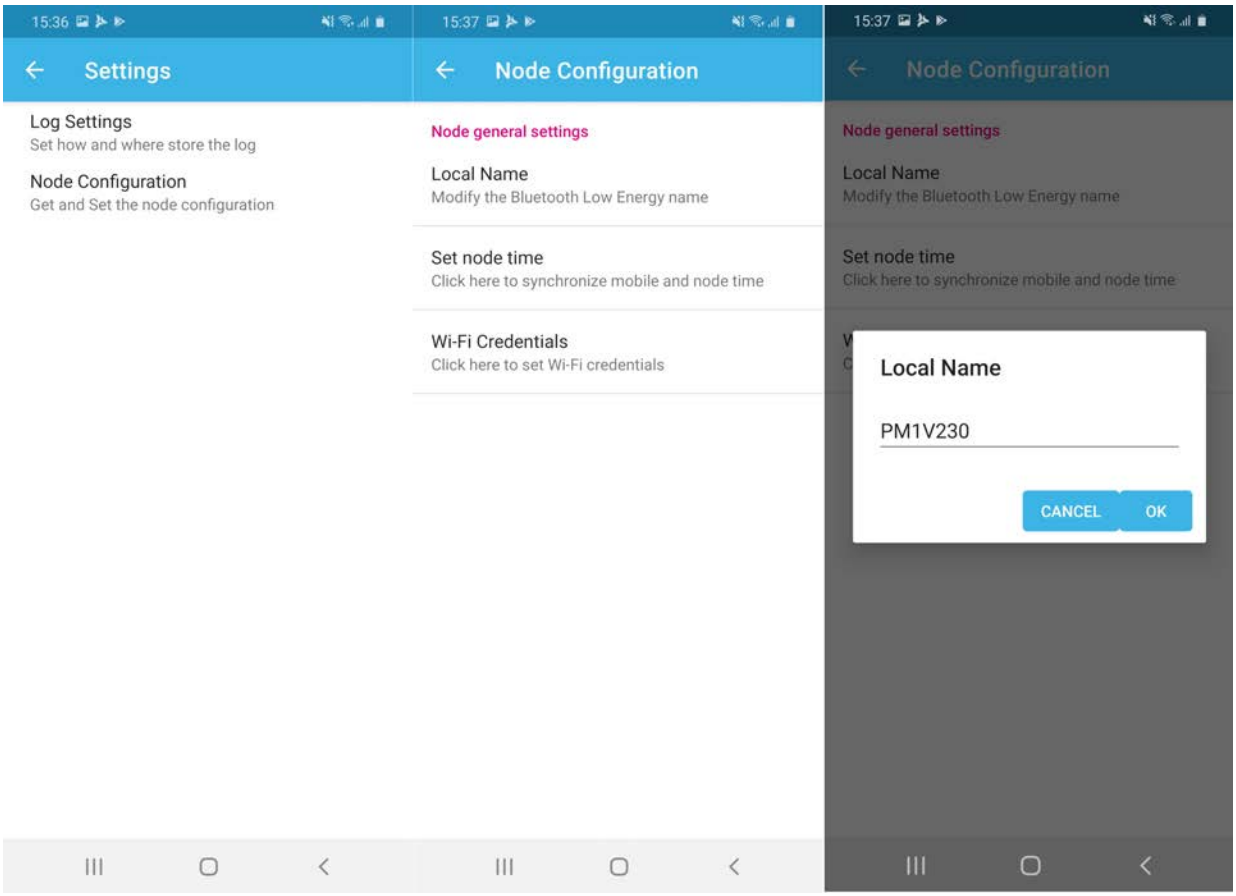
- Settings
- Serial or Debug (with stdin) console
- Firmware upgrade

Figure 27. **STBLESensor (Android version) menu selection**



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

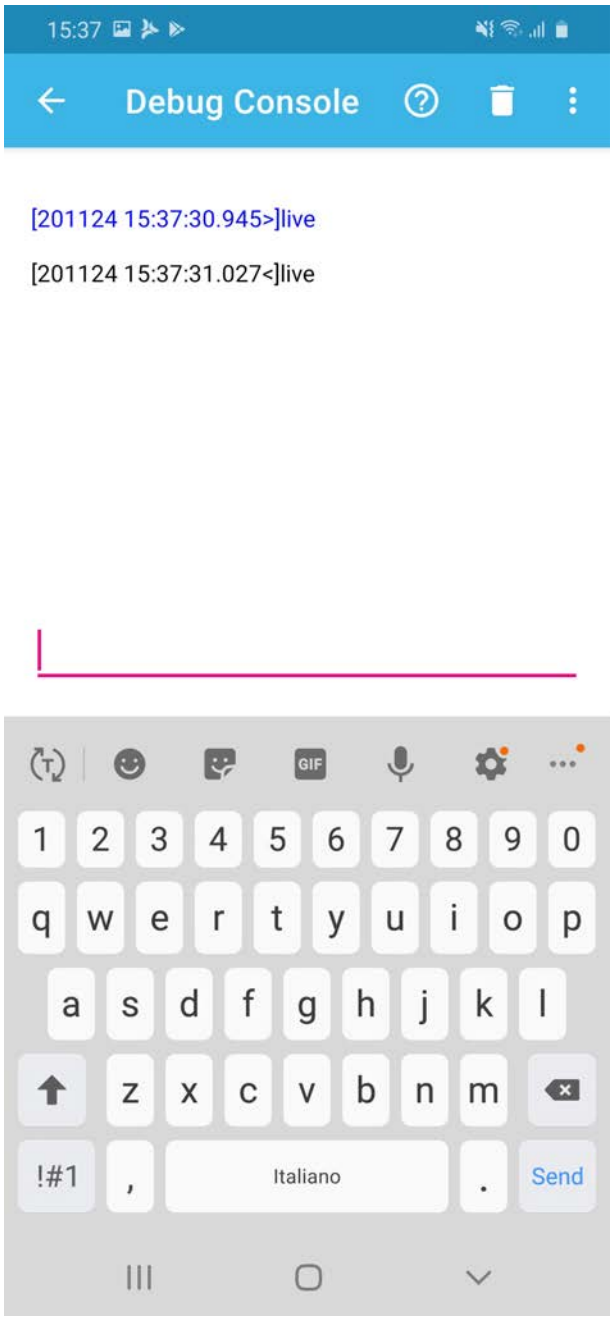
Figure 28. **STBLESensor (Android version) node configuration: local name**



If the Debug console is enabled, stdin is displayed and any message written triggers a reply with the same message, if the command line is not recognized, as shown below.

Note: Node configuration is not available if the board configuration feature is enabled.

Figure 29. **STBLESensor (Android version) Debug console (stdin/stdout/stderr)**



The [**help**] command shows the list of the recognized commands.

Figure 30. **STBLESensor (Android version) Debug console (stdin/stdout/stderr): Help Command**

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

1. for temperature/humidity/pressure with the command:
   a. @TM: the application sends environmental data every 5 s
   b. @TH: the application sends environmental data every 1 s
   c. @TL: the application sends environmental data every 100 ms
   d. @TD: the application sends environmental data at the default rate (500 ms)
2. for 3D accelerometer, 3D gyroscope and 3D magnetometer with the command:
   a. @AM: the application sends environmental data every 5 s
   b. @AH: the application sends environmental data every 1 s
   c. @AL: the application sends environmental data every 100 ms
   d. @AD: the application sends environmental data at the default rate (50 ms)
3. for signal noise microphone levels with the command:
   a. @MM: the application sends environmental data every 5 s
   b. @MH: the application sends environmental data every 1 s
   c. @ML: the application sends environmental data every 100 ms
   d. @MD: the application sends environmental data at the default rate (50 ms)

2.10.4 FTT amplitude

The “FFT Amplitude” page can be used to monitor the vibration frequency plot from MotionSP algorithms provided with FP-IND-PREDMNT1.

Figure 31. STBLESensor (Android version) FFT amplitude page for predictive maintenance (plot, details and settings

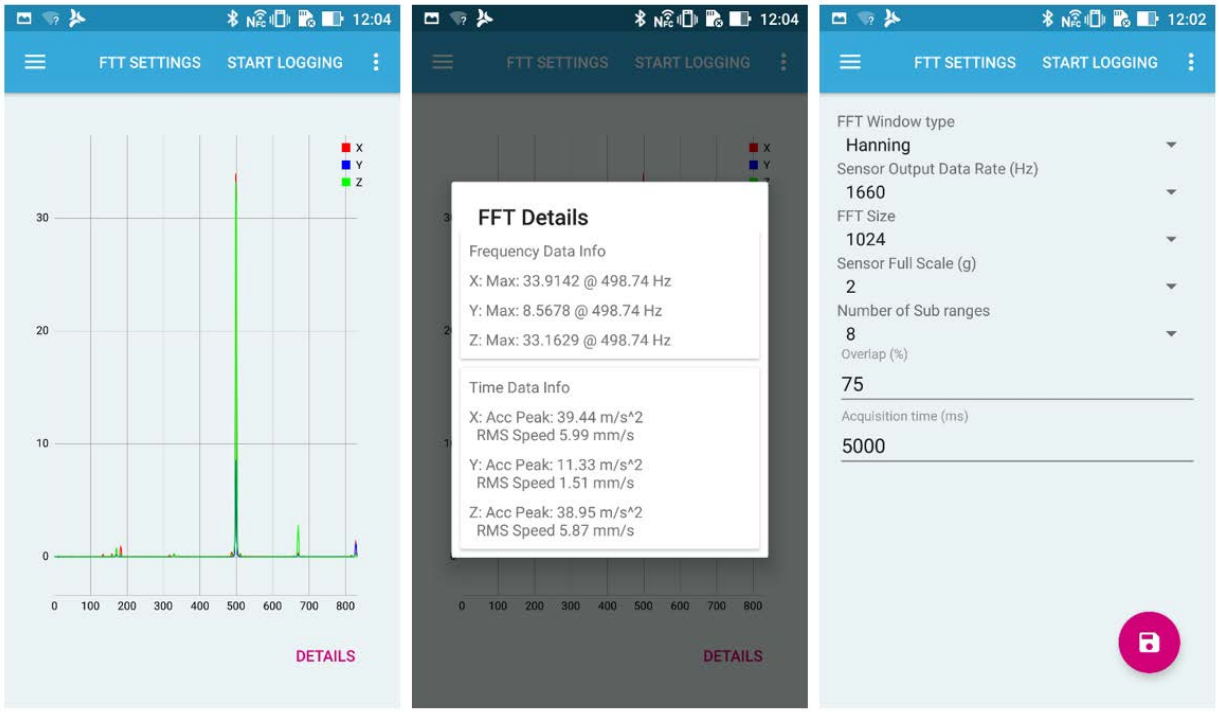
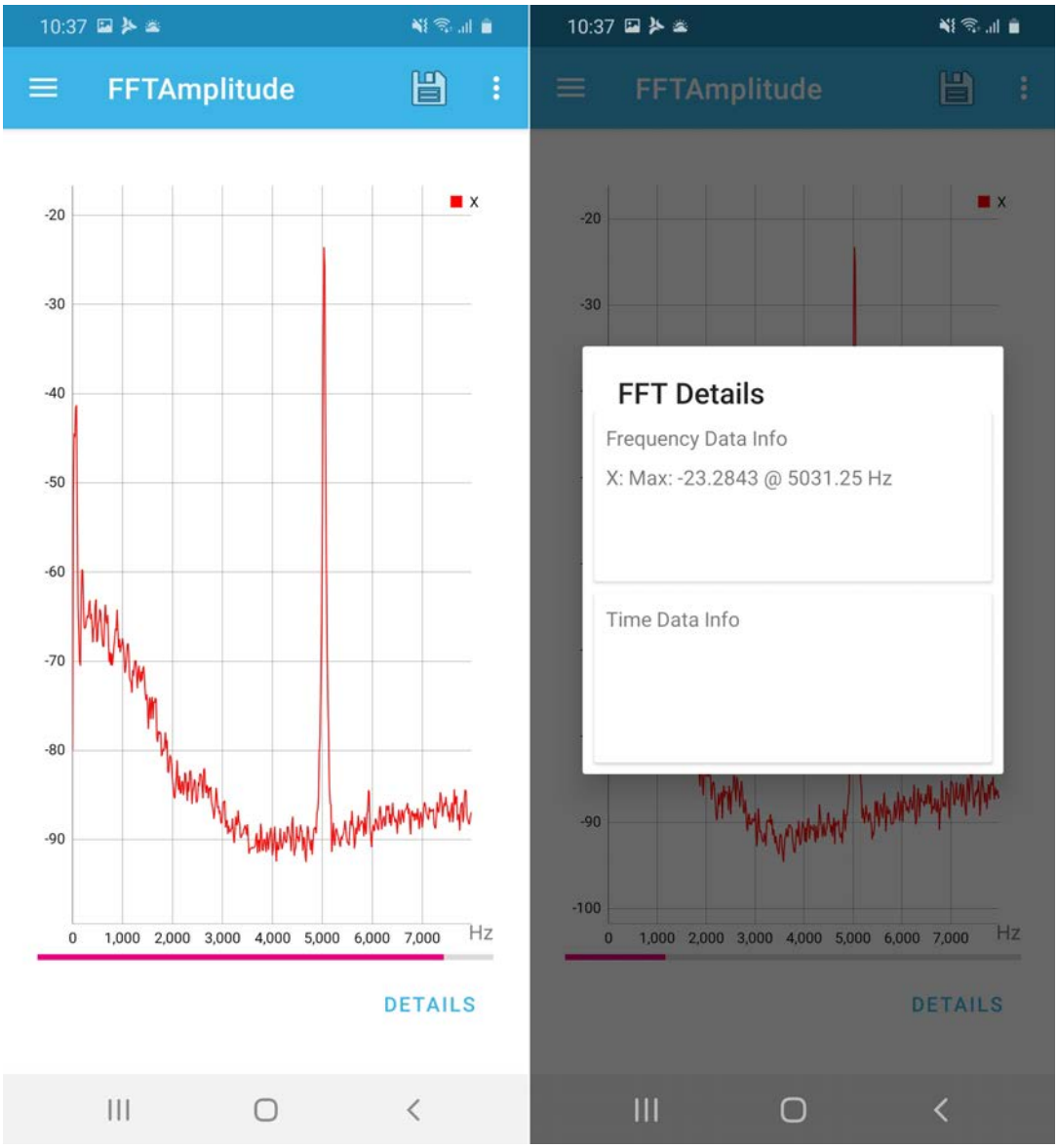


Figure 32. **STBLESensor (Android version) FFT amplitude page for acoustic analysis (plot and details)**



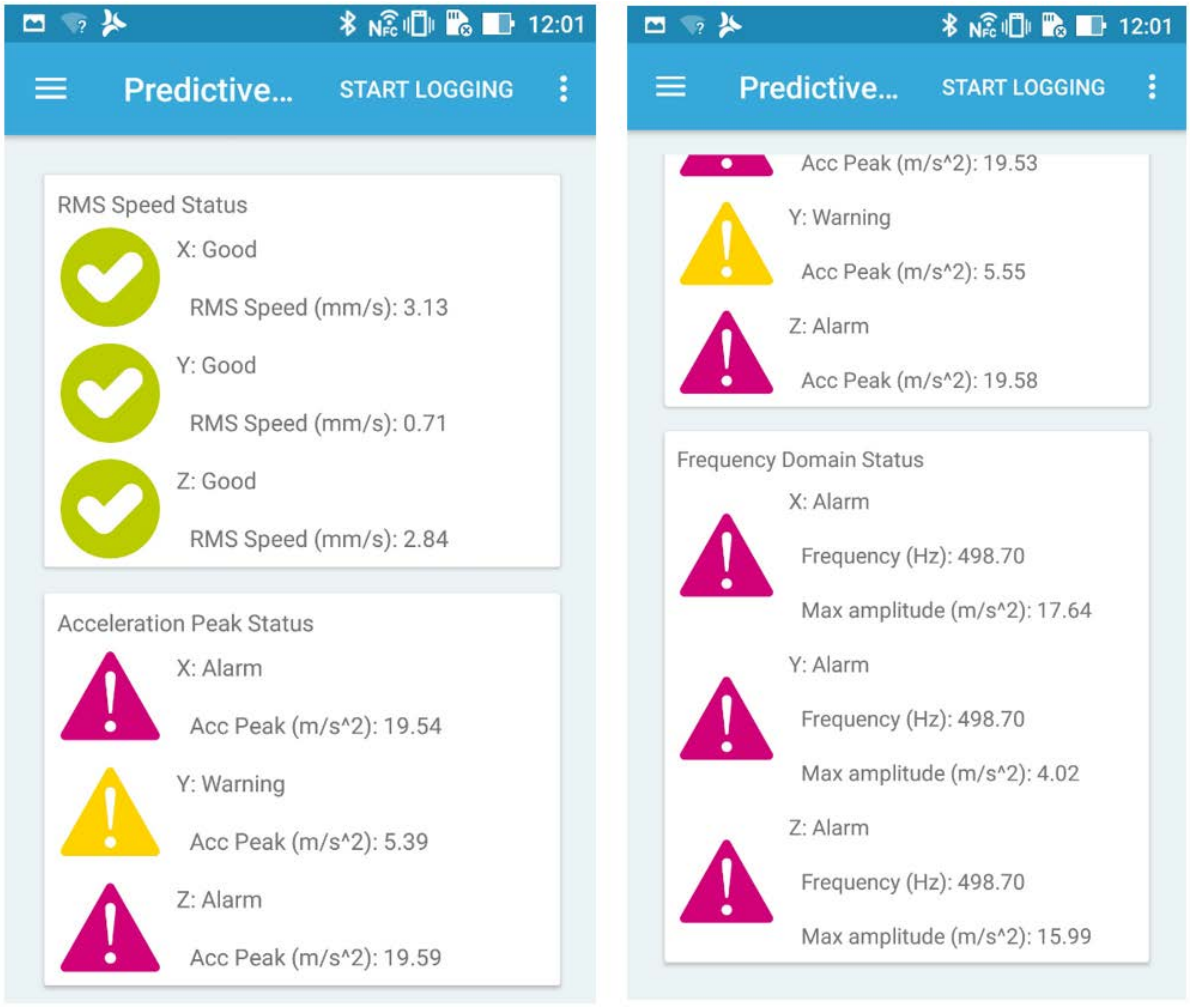
2.10.5 Predictive maintenance

The predictive maintenance page, based on MotionSP algorithms, can be used to monitor key parameters and the equipment status based on:

- RMS speed
- acceleration peak
- accelerometer frequency domain

The parameter status results are based on configurable alarm and warning thresholds you can modify for speed RMS, accelerometer peak and spectral band, precompiling the **MotionSP_Threshold.h** file included in *\Projects\STM32F446RE-Nucleo\Demonstrations\Predictive_Maintenance\Inc*.

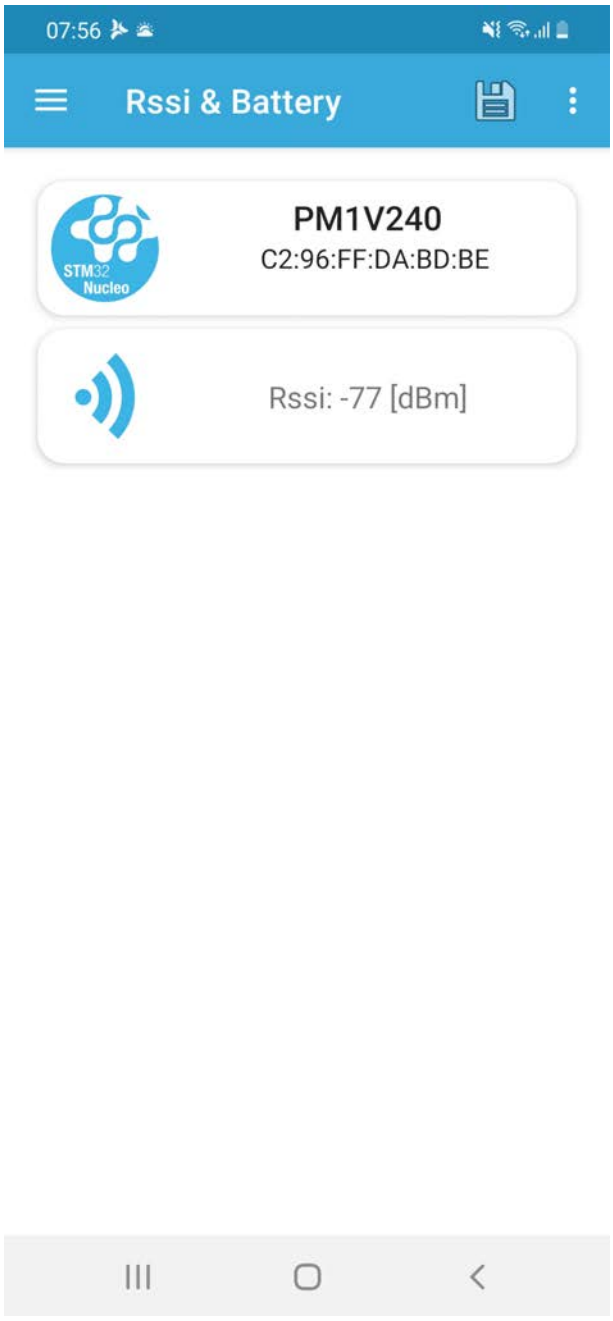
Figure 33. **STBLESensor (Android version) predictive maintenance page**



2.10.6 RSSI and battery

This page shows RSSI of the Bluetooth signal strength.

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



The RSSI value is updated every 0.5 seconds.

2.10.7 Board configuration

If board configuration feature is enabled, more simply can be obtained the board report and node name setting.

Figure 35. **Board configuration selection**

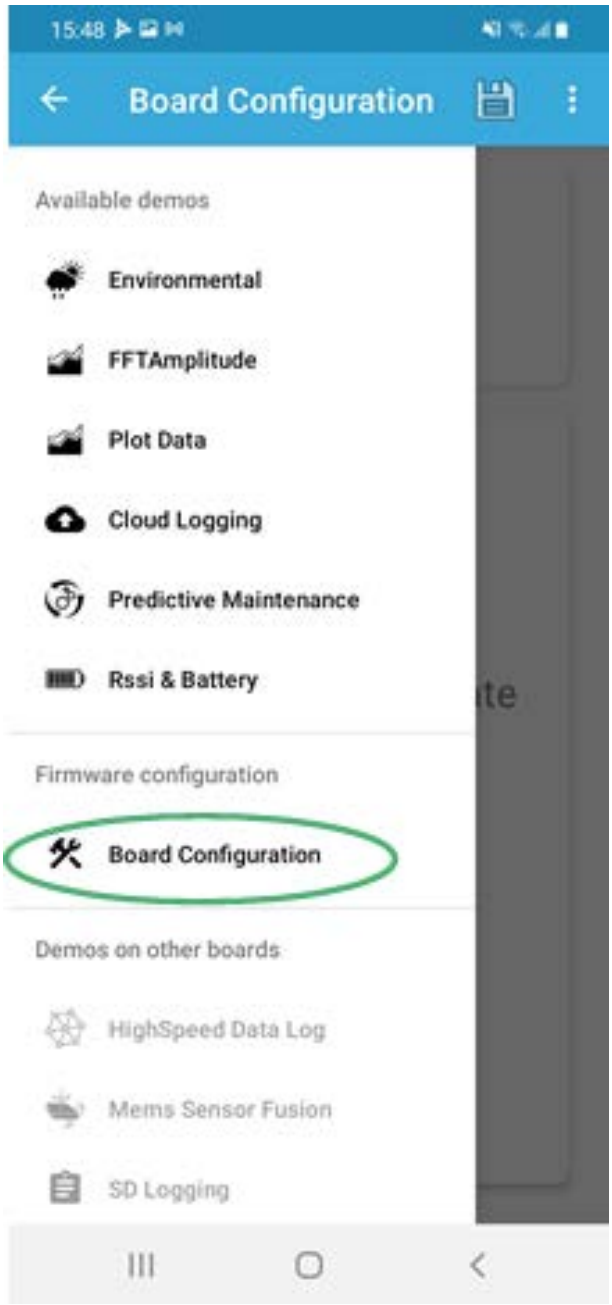


Figure 36. **Board configuration options**

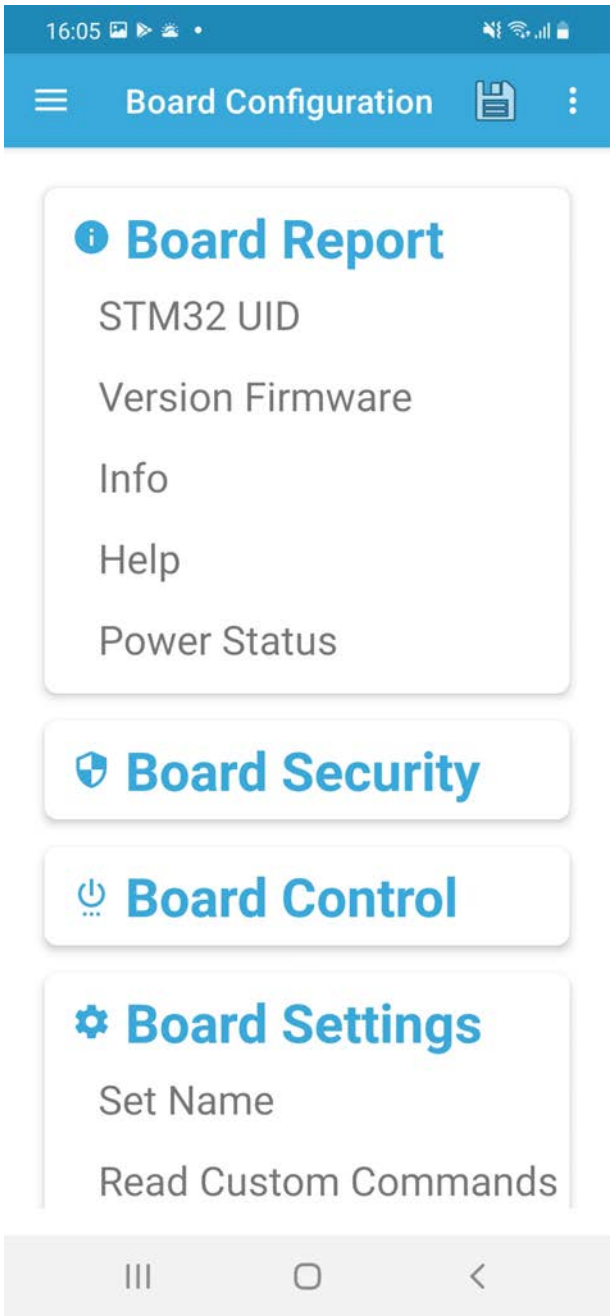


Figure 37. **Board firmware version**

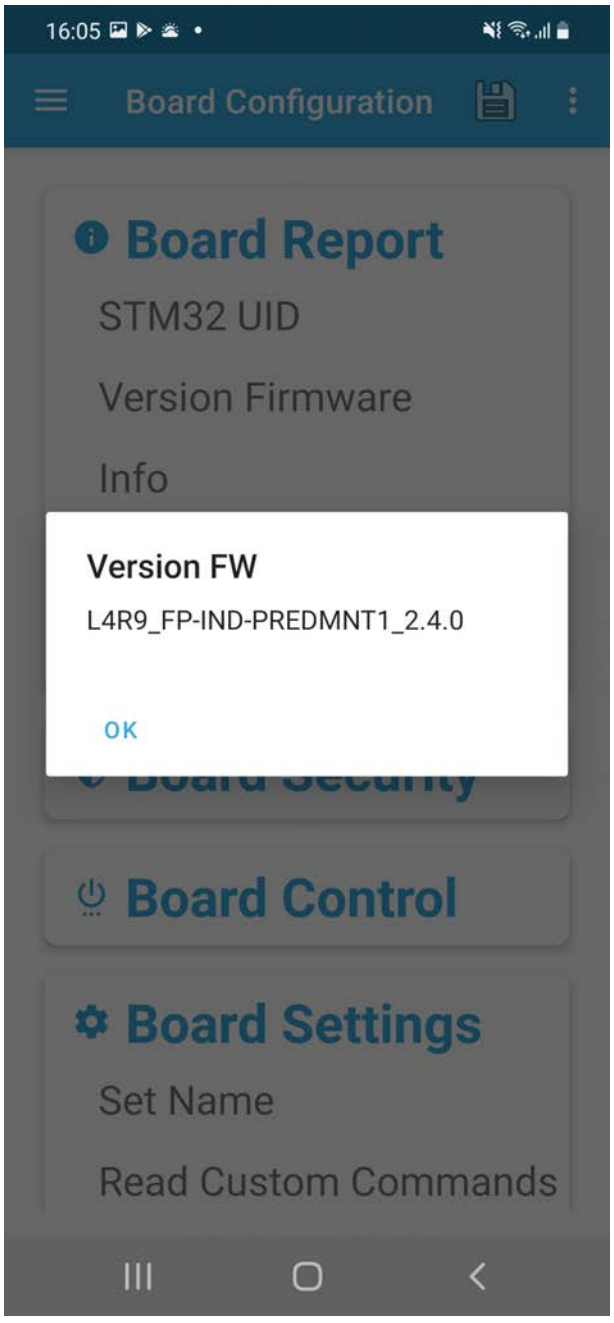


Figure 38. **Hardware and firmware information**

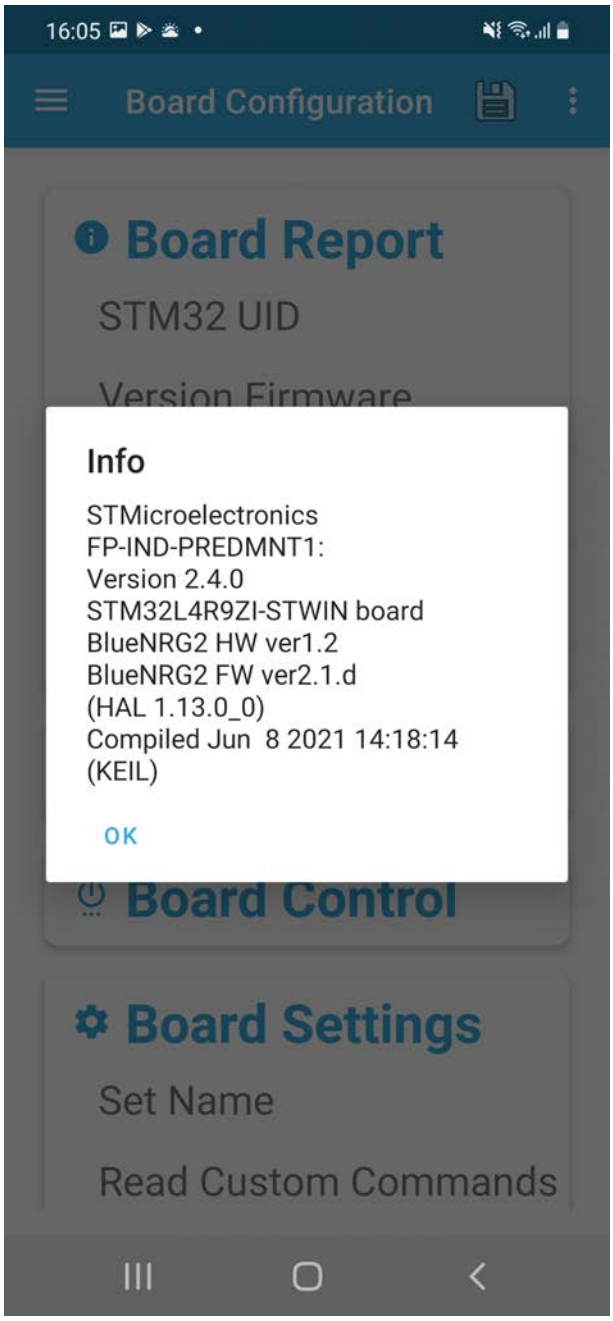
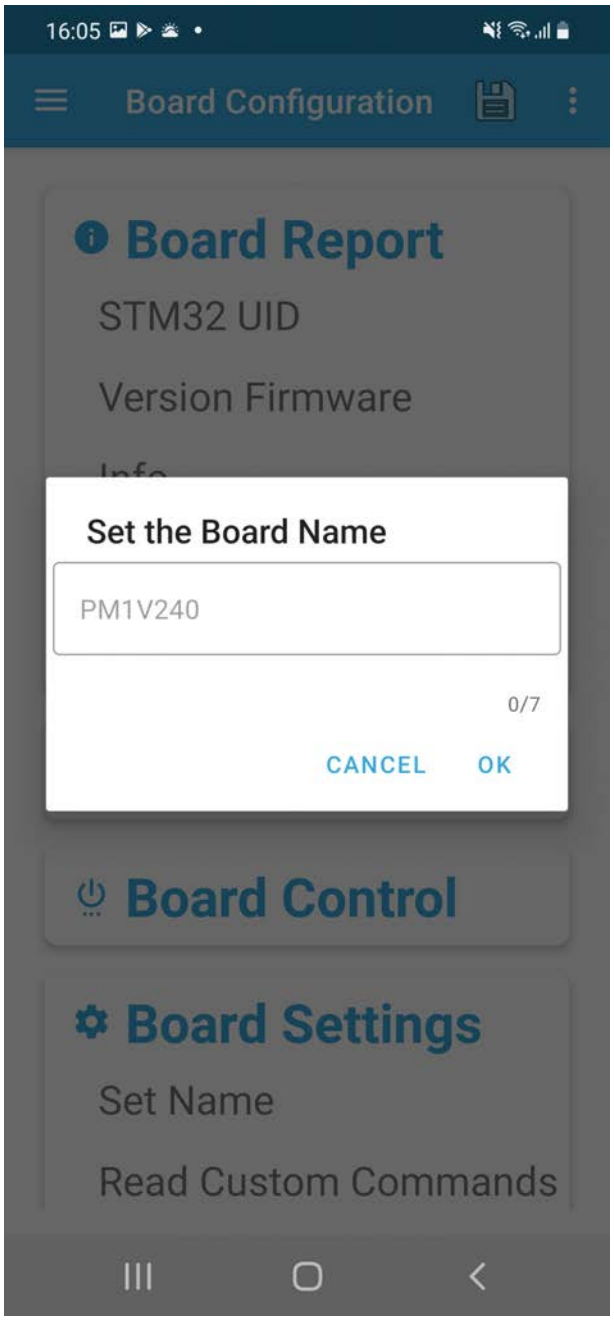


Figure 39. **Setting the board name**



The firmware parameters can be changed by selecting "Read Custom Commands".

Figure 40. **Read Custom Commands**

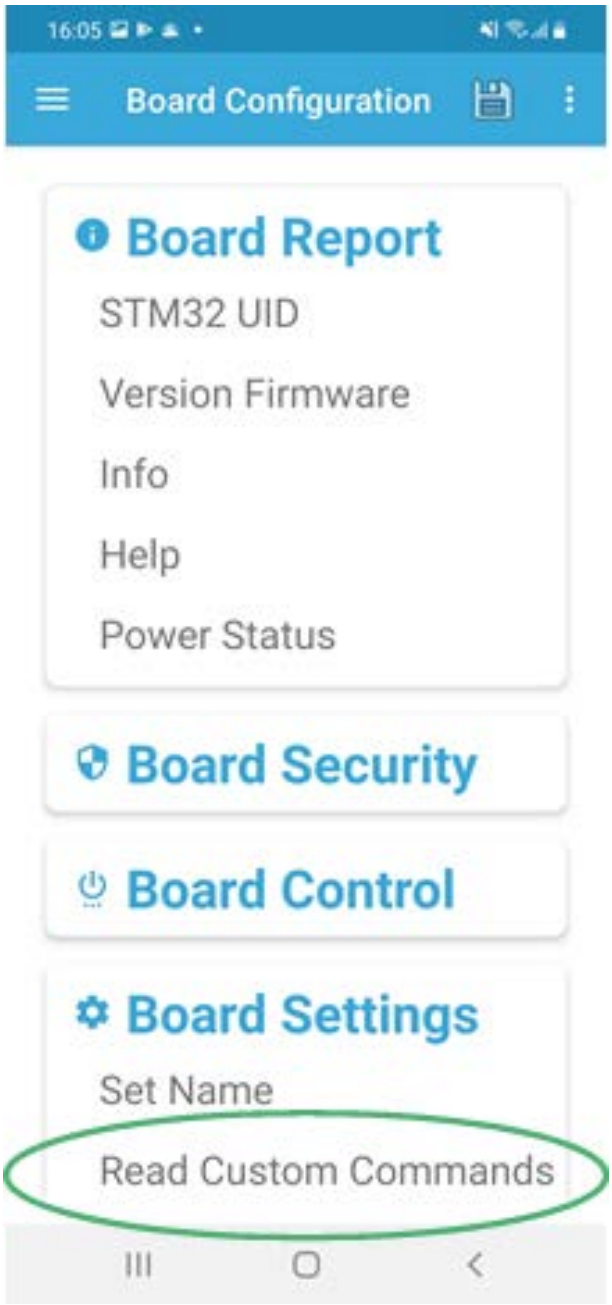


Figure 41. **List of custom commands**

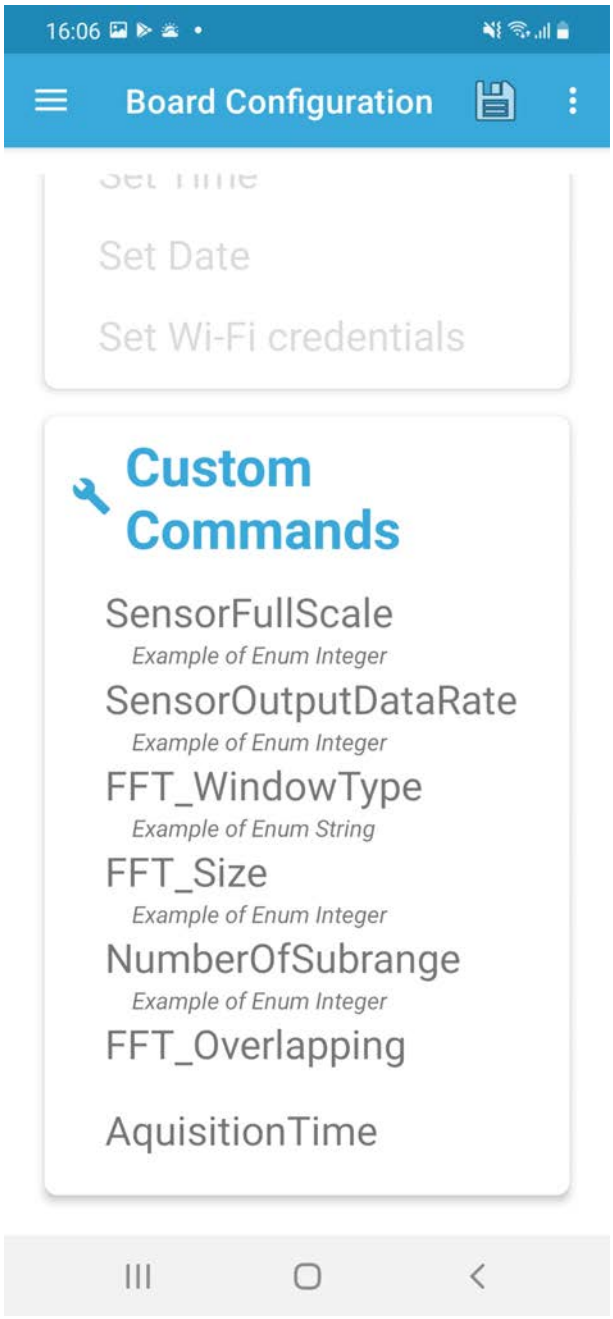
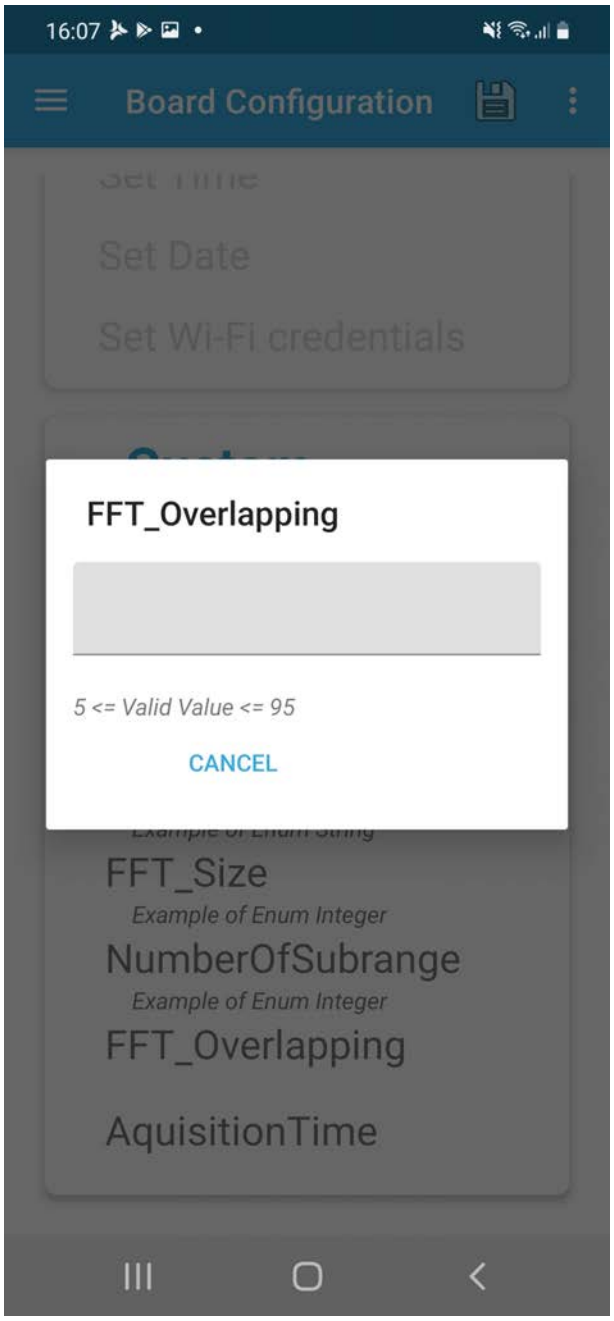


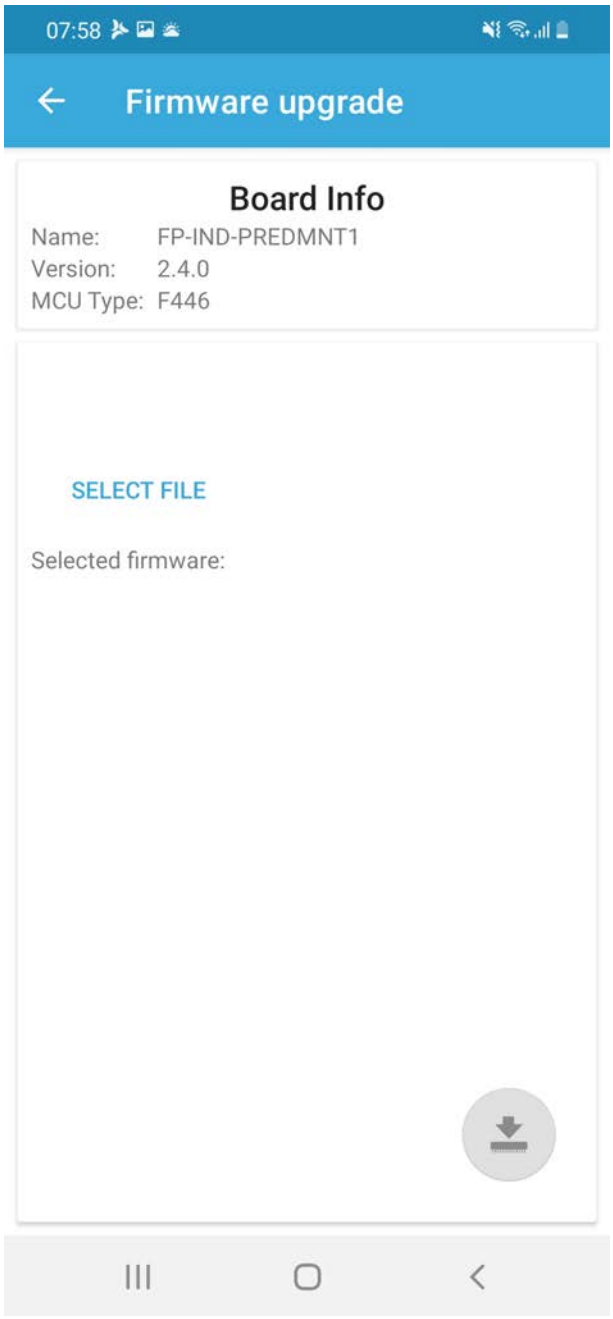
Figure 42. **Custom commands - example (FFT_Overlapping)**



2.10.8 Firmware-Over-The-Air update with STBLESensor

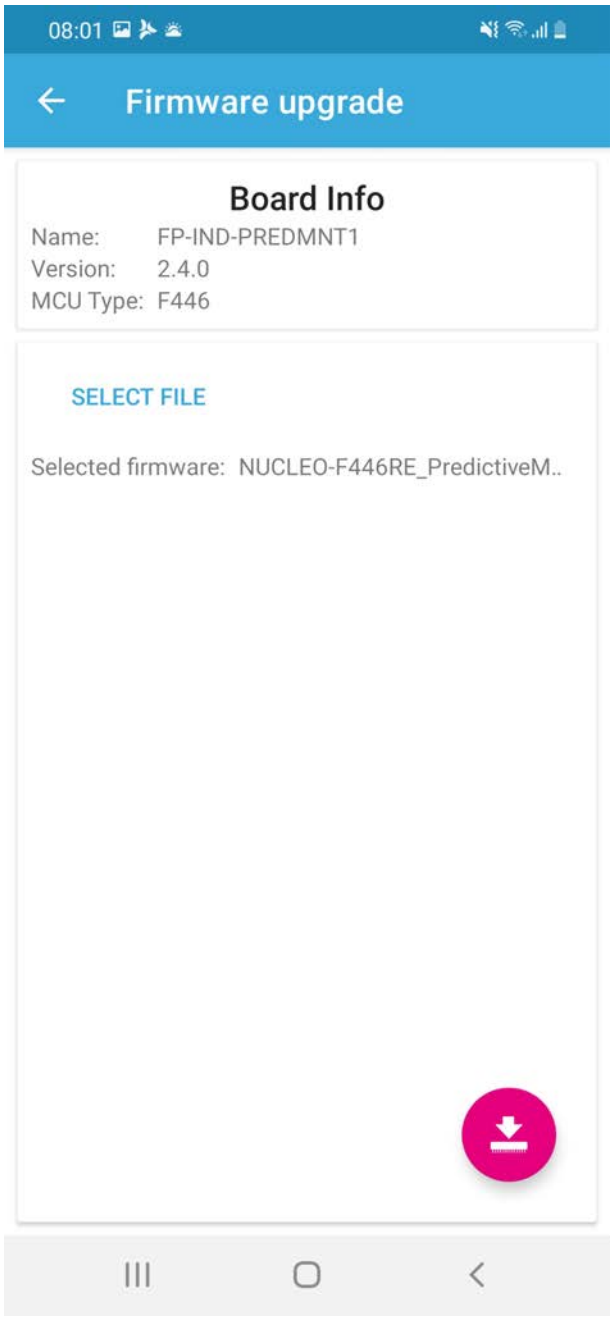
If the Firmware upgrade option menu is selected, the following page appears.

Figure 43. **STBLESensor (Android version) firmware upgrade page**



The STBLESensor application shows which version of the FP-IND-PREDMNT1 software is running and the board type. To apply an update, press the pink button and choose the appropriate file.

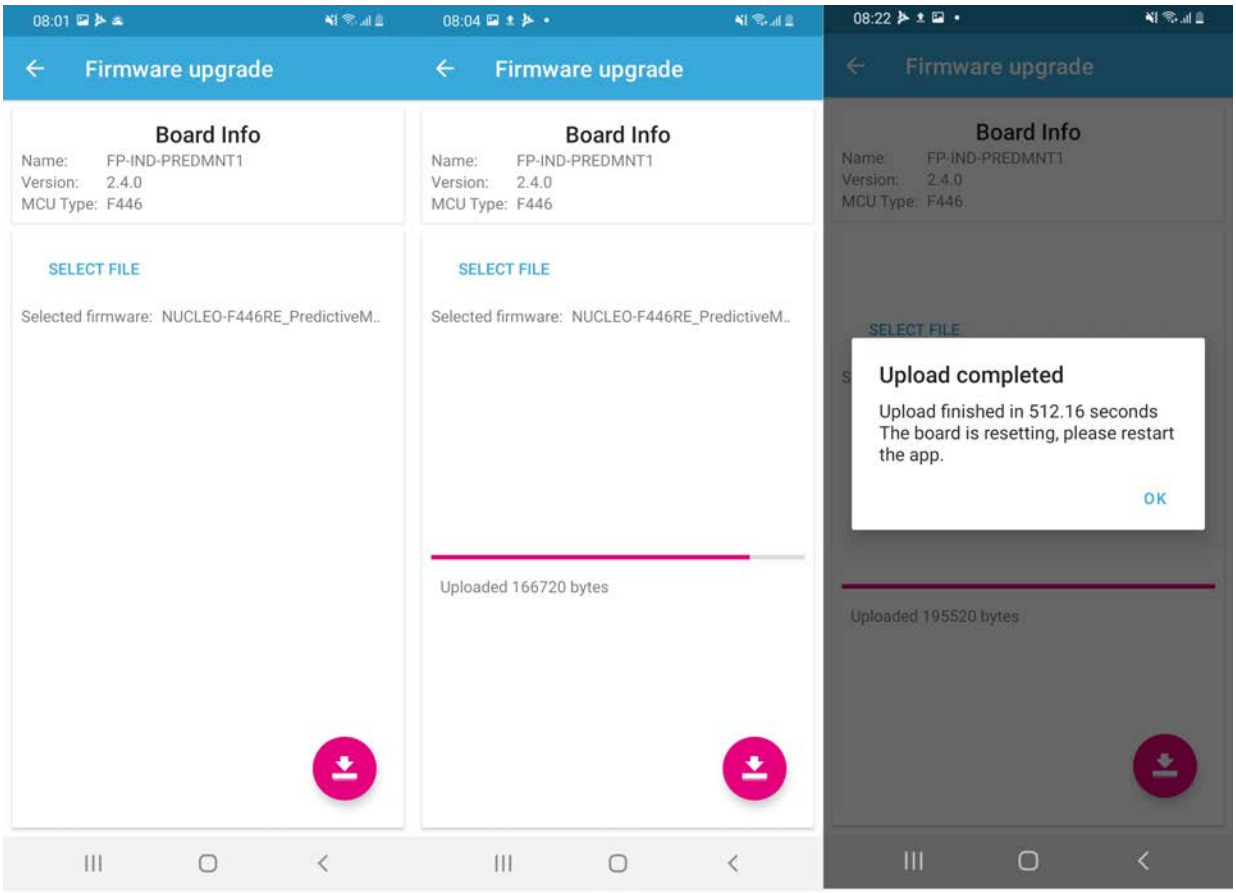
Figure 44. **STBLESensor (Android version) firmware update file selection**



STBLESensor sends the FP-IND-PREDMNT1 a command communicating that it is going to send an update of a certain byte size and corresponding CRC value.

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

Figure 45. **STBLESensor (Android version) feedback during and after FOTA transmission**



On completion of FOTA transmission, the STM32 uses the CRC hardware unit to compute the CRC value for the FOTA received. If this CRC matches the expected CRC previously sent by the application, FP-IND-PREDMNT1 writes a code number to signal the Bootloader there is an OTA ready to be applied.

As the following figure shows, the Bootloader applies the OTA at the next board reboot and executes the new FP-IND-PREDMNT1 firmware.

Figure 46. **STBLESensor (Android version) UART console after the FOTA and before update**

3 STEVAL-BFA001V2B platform

The STEVAL-BFA001V2B is an industrial reference design kit for condition monitoring (CM) and predictive maintenance (PdM).

The hardware consists of a highly compact (50 x 9 x 9 mm) industrial sensor board specifically designed for real industrial applications, and the necessary debugging tools, cables, plugs and adapters for an industrial communications scenario. The connection is managed using a standard multipolar cable with one wire used for IO-Link data.

The layout is designed to meet IEC61000-4-2/4 and EN60947 requirements for industrial applications.

The STSW-BFA001V2 firmware package (freely available on www.st.com) includes dedicated algorithms for advanced time and frequency domain signal processing and analysis of the high bandwidth 3D digital accelerometer for vibration monitoring. The package includes pressure, relative humidity and temperature sensor monitoring samples as well as audio algorithms for acoustic emission (AE).

The firmware runs on the high performance STM32F469AI, ARM® Cortex®-M4, 32-bit microcontroller. The sensor data analysis results can be displayed on a user PC terminal emulator via wired connectivity or the related IO-Link master board interface.

Figure 47. STEVAL-BFA001V2B predictive maintenance reference kit

RELATED LINKS

1.1.1 Compatible hardware platforms on page 2

3.1 System setup

For the STEVAL-BFA001V2B kit setup, the STLINK-V3MINI must be used to program the STEVAL-IDP005V2 and monitor data. You can download the relevant version of the STLINK-V3MINI USB driver at STSW-LINK007.

3.2 Hardware setup

Step 1. Plug the STEVAL-UKI001V2 on the STLINK-V3MINI to overlap the connectors with the same identification.

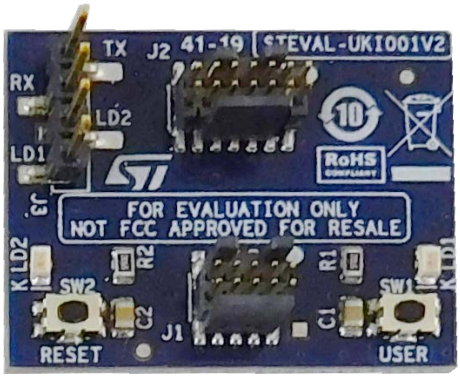
Step 2. Connect the STLINK-V3MINI to the PC through the USB micro-B cable.

Step 3. Respecting the polarity, connect the 10-pin flat IDC cable to STEVAL-UKI001V2.

Step 4. Connect the STEVAL-UKI001V2 to STLINK-V3MINI using the STDC14 pin flat cable.

Step 5. Use the 4-wire cable with free ends and an M12 A-coded 4-pin female connector.

Figure 48. STEVAL-UKI001V2 top view



Step 6. Connect the M12 A-coded 4-pin female connector of the cable to JP1 (IO-Link connector) to the STEVAL-IDP005V2.

Step 7. Connect wire 1 (VIN) and wire 3 (GND) of the cable to a power supply able to provide 18 to 32 V_{DC}.

Step 8. Respecting the polarity, connect the free end of the 10-pin flat IDC wire cable to STEVAL-IDP005V2 J1 (SWD connector).

Figure 49. Connection between ST-LINK-V3MINI and STEVAL-UKI001V2

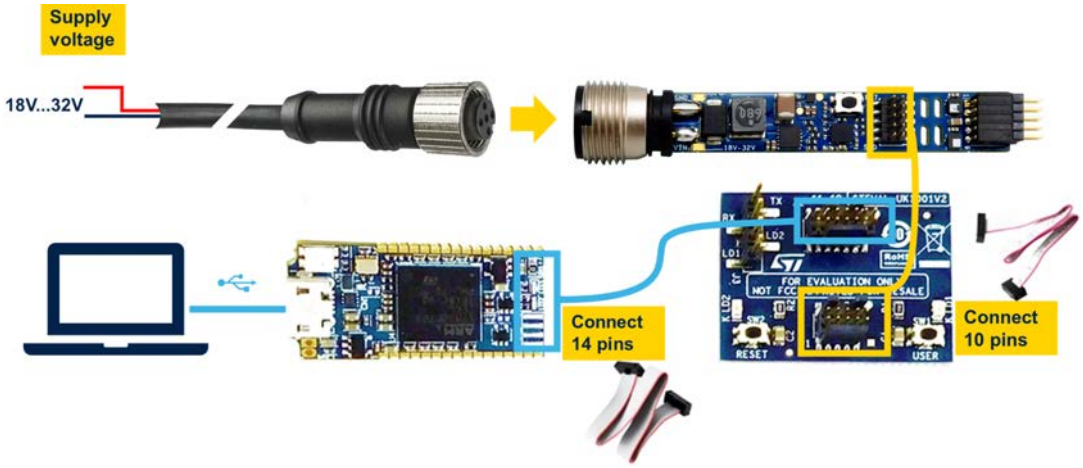
Figure 50. Power supply connection (without IO-Link master board)

– Pin 1 (Brown) as positive
– Pin 3 (Blue) as negative

The STEVAL-IDP005V2 is ready to be programmed with new firmware or to monitor data.

Step 9. To display data, open a terminal emulator, configured with baud rate 230400, 8N1 parameters, and run it to monitor and log application data on the PC.

Figure 51. STEVAL-BFA001V2B complete setup kit connection in stand-alone mode



3.3 Software setup

The following software components are required for the setup of a suitable development environment to create applications for the STEVAL-BFA001V2B kit setup:

- FP-IND-PREDMNT1 firmware, available on www.st.com.
- Standard user terminal software, as Putty or Tera Term (v. 4.97 or higher)
- Development tool-chain and Compiler. The STM32Cube expansion software supports the following environments to select from:
  - IAR Embedded Workbench for ARM® (EWARM release 8.50.5) toolchain plus ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM release 5.31.0) toolchain plus ST-LINK
  - Integrated Development Environment for STM32 (STM32CubeIDE release 1.4.2) plus ST-LINK

3.4 The installation process

The package Binary directory, in each application/demonstration folder (i.e Projects\STM32F469AI-BFA001V2\Demonstrations\Predictive_Maintenance_SRV\Binary), contains an image for each application/demonstration, i.e. pre-compiled binaries that can be flashed via ST-LINK to the correct memory address (0x08000000) of a STEVAL-IDP005V2 board (in the STEVAL-BFA001V2B kit):

- STEVAL-BFA001V2_AcousticAnalysis_v2.3.0.bin
- STEVAL-BFA001V2_EnvMon_v2.3.0.bin
- STEVAL-BFA001V2_VibrationAnalysis_v2.3.0.bin
- STEVAL-BFA001V2_PredictiveMaintenance_v2.3.0.bin

3.5 Sample application description

3.5.1 Terminal window setup

For the STEVAL-BFA001V2B development kit, you can set a terminal window up for the appropriate UART communication port, in order to provide a tool for monitor and data logging.

3.5.2 Sample application startup

To run a sample application with the STEVAL-BFA001V2B, follow the procedure below.

Step 1. Plug the STEVAL-UKI001V2 to an STM32 Nucleo-64 board

Step 2. Connect the STEVAL-UKI001V2 plus the STM32 Nucleo board assembly to the STEVAL-IDP005V2 trough the 10-pin flat IDC wire cable

Step 3. Supply power (refer to Section 3.2 Hardware setup)

Step 4. Connect the STLINK-V3MINI to the PC through the USB Type-A Maleto Type-B mini cable

Step 5. Open and configure your terminal emulator (such as PuTTY or Tera Term)

Step 6. Set the following parameters:

- Name: COM Port name
- Baud Rate: 230400
- Data: 8
- Parity: None
- Stop Bit: One
- Flow Control: None

Step 7. Push the [**Reset**] button on the STEVAL-UKI001V2 (or STEVAL-IDP005V2)

Step 8. Follow the terminal instruction to monitor data

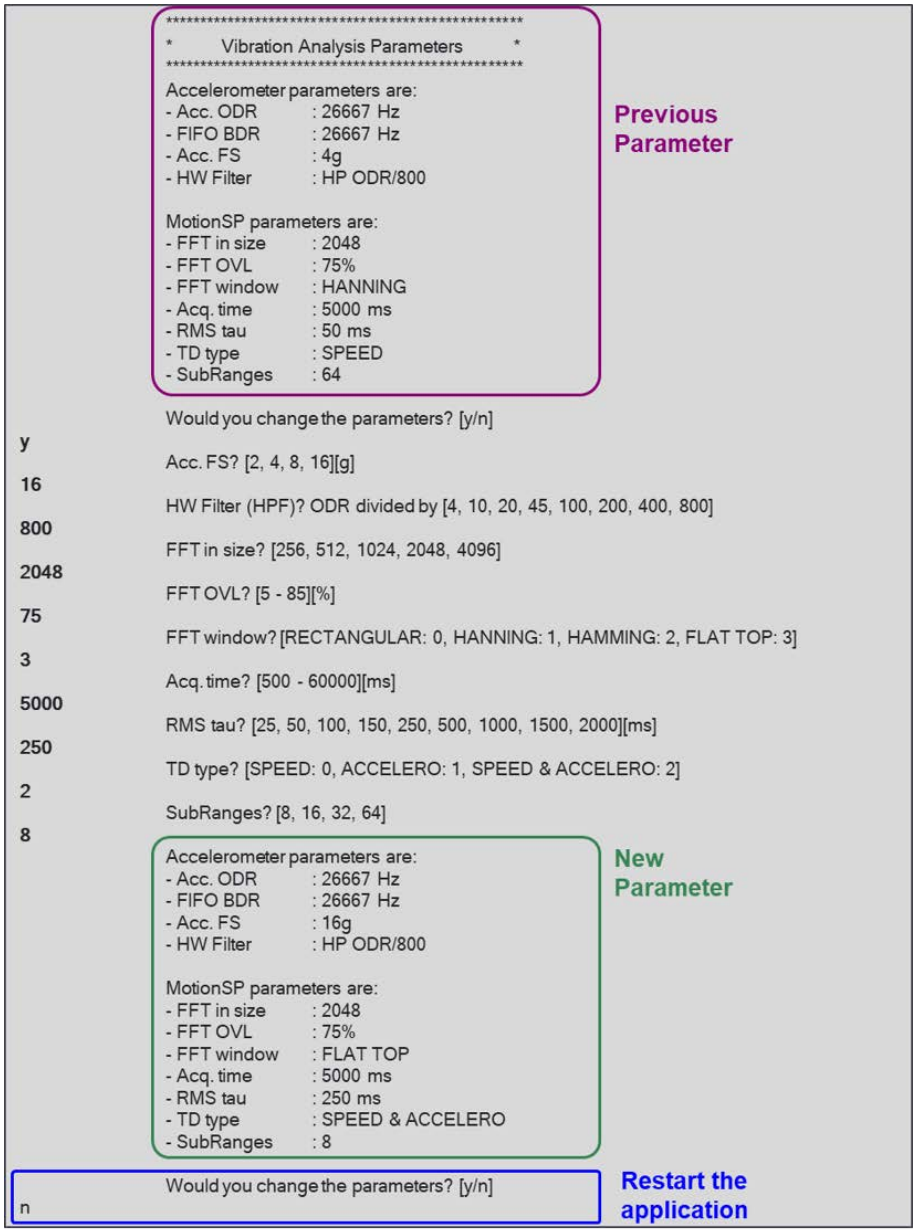
For example, if you want to run a PREDICTIVE MAINTENANCE application inside the package folder Projects\STM32F469AI-BFA001V2\Demonstration\Predictive_Maintenance_SRV\Binary, after a board reset, the screen shows data similar to the figure below.

Figure 52. STEVAL-BFA001V2B serial terminal output after reset

Step 9. Type ‘y’ to change or confirm each configuration parameters as explained in the next figure.

Pay attention to every parameter inserted via user terminal: ensure it overwrites the previous line in the window, so the following figure is simply a summary obtained through the "log" functionality.

Figure 53. STEVAL-BFA001V2B parameter settings and configuration



The new parameters written by the user terminal are now fixed and ready to use from the library, that configures everything before starting the application, measuring time domain (1st array) and frequency domain (2nd array).

Figure 54. **STEVAL-BFA001V2B parameter overview and analysis with time domain array**

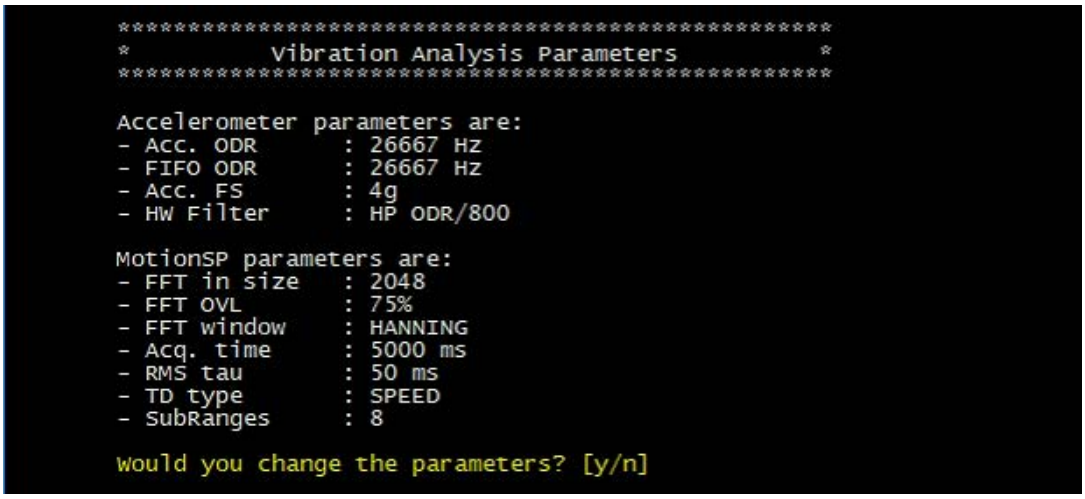
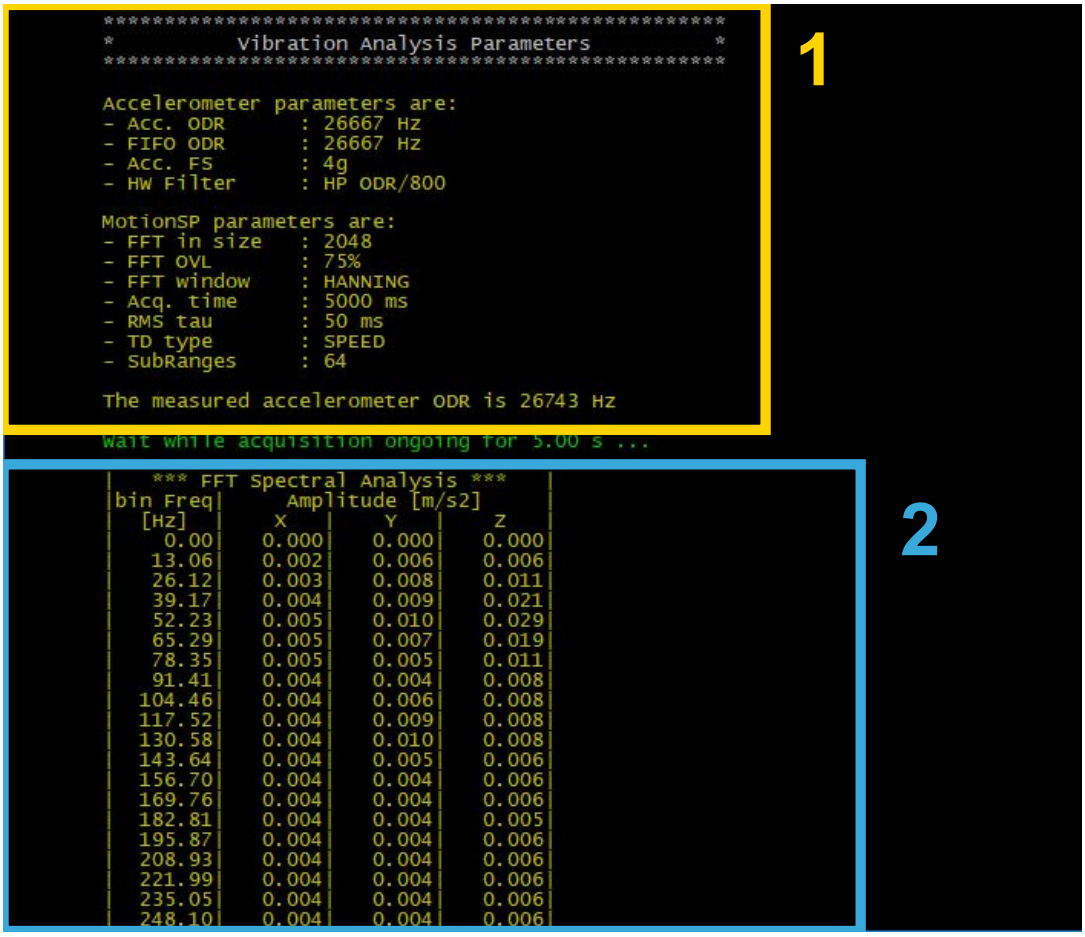


Figure 55. **STEVAL-BFA001V2B 2nd analysis with frequency domain array**

1. Parameters and initialization summary
2. Analysis start with time domain array



The analysis results shown for this application are:

– Frequency domain values about the FFT average numbers and the absolute maximum values detected in the whole power spectrum

– Frequency domain threshold status for each subrange analized

– General status for time and frequency domain

Figure 56. STEVAL-BFA001V2B time domain predictive maintenance results

1. FFT arrays
2. Time domain results and threshold status
3. FFT average numbers and results: max. amplitude per frequency

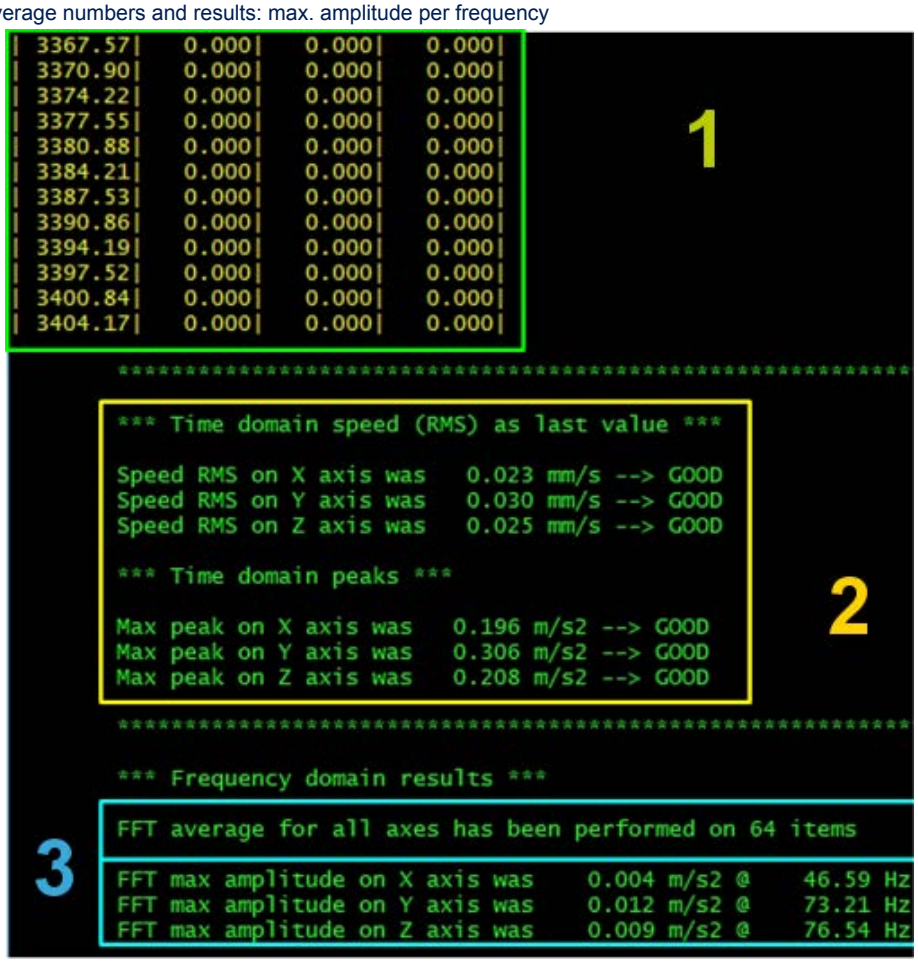
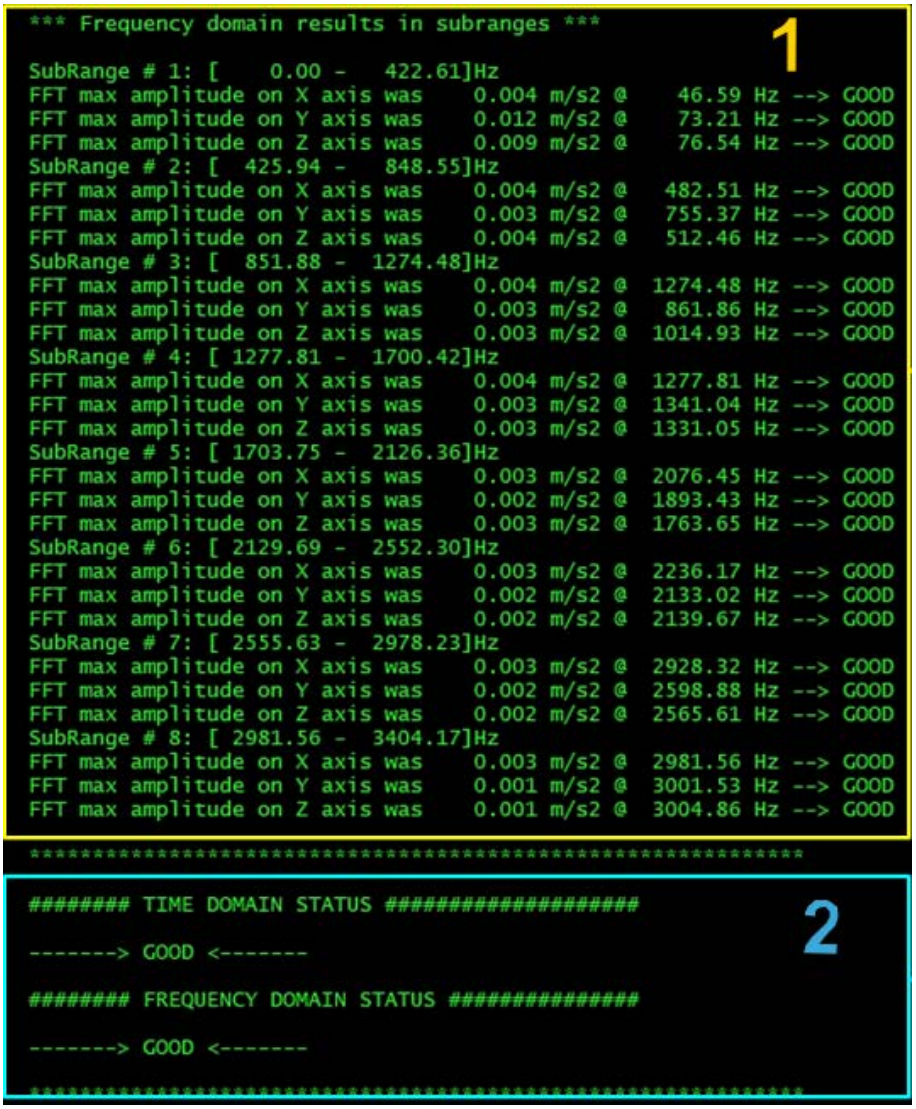


Figure 57. STEVAL-BFA001V2B frequency domain sub-range status and all status results

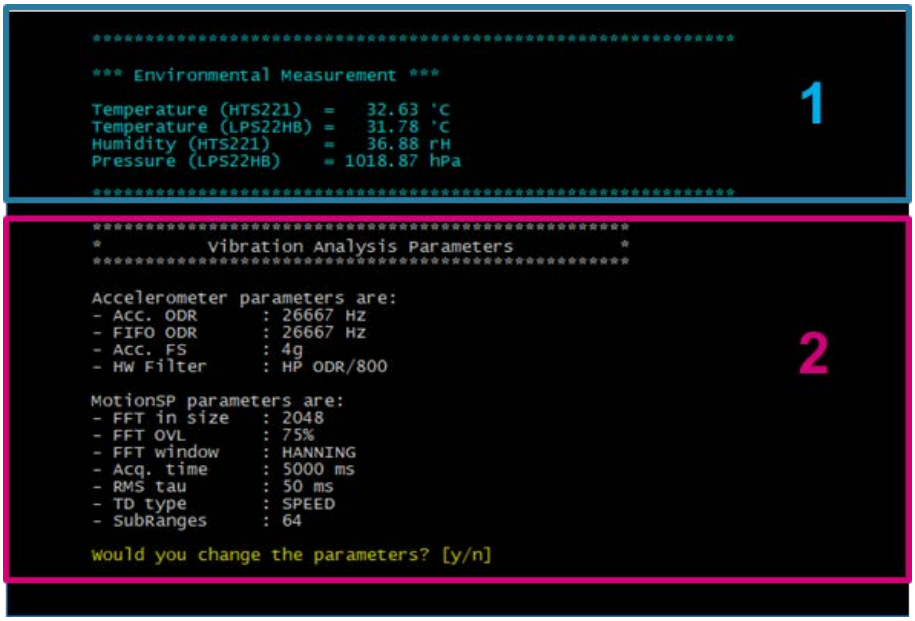
1. FFT sub-range analysis, max. amplitude per frequency status
2. Time domain and frequency domain: all threshold status



The figure below shows the environmental measurements and parameters already configured with the opportunity of changing them before starting next analysis.

Figure 58. STEVAL-BFA001V2B environmental measurement and parameter list

1. Environmental measurements
2. Parameter overview for next analysis



4 STEVAL-STWINKT1 platform

The STWIN SensorTile wireless industrial node (STEVAL-STWINKT1) is a development kit and reference design that simplifies prototyping and testing of advanced industrial IoT applications such as condition monitoring and predictive maintenance.

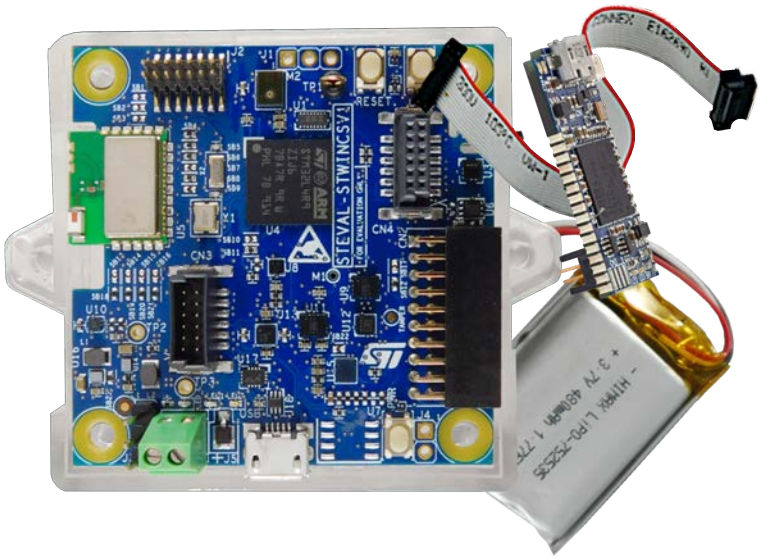
The kit features a core system board with a range of embedded industrial-grade sensors and an ultra-low-power microcontroller for vibration analysis of 9-DoF motion sensing data across a wide range of vibration frequencies, including very high frequency audio and ultrasound spectra, and high precision local temperature and environmental monitoring.

The development kit is complemented with a rich set of software packages and optimized firmware libraries, as well as a cloud dashboard application, all provided to help speed up design cycles for end-to-end solutions.

The kit supports BLE wireless connectivity through an on-board module, and Wi-Fi connectivity through a special plugin expansion board (STEVAL-STWINWFV1). Wired connectivity is also supported via an on-board RS485 transceiver. The core system board also includes an STMOD+ connector for compatible, low cost, small form factor daughter boards associated with the STM32 family, such as the LTE Cell pack.

Apart from the core system board, the kit is provided complete with a 480 mAh Li-Po battery, an STLINK-V3MINI debugger and a plastic box.

Figure 59. STEVAL-STWINKT1 devlopment kit



RELATED LINKS

1.1.1 Compatible hardware platforms on page 2

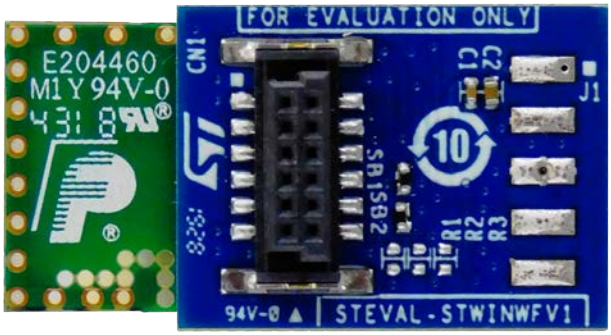
4.1 STEVAL-STWINWFV1

The STEVAL-STWINWFV1 expansion board adds 2.4 GHz Wi-Fi connectivity to the SensorTile Wireless Industrial Network (STWIN) kit (STEVAL-STWINKT1).

It features:

- Wi-Fi adapter for STEVAL-STWINKT1B (and STEVAL-STWINKT1)
- Plugs into STWIN core system board through dedicated 12-pin connector
- Single 3.3 V power supply input
- ISM43362-M3G-L44-E Wi-Fi module:
  - 802.11 b/g/n Compatible based on Broadcom MAC/Baseband/Radio device
  - Fully contained TCP/IP stack
  - Host interface: SPI up to 25 MHz
  - Network features ICMP (Ping), ARP, DHCP,TCP, UDP
  - Low power operation (3.3 V supply) with built-in low power modes
  - Secure Wi-Fi authentication WEP-128, WPA-PSK (TKIP), WPA2-PSK
  - CE certified
  - FCC certified
  - IC certified
- RoHS and China RoHS compliant
- WEEE compliant

Figure 60. STEVAL-STWINWFV1 expansion board



4.2 Hardware setup

The following hardware components are needed:

- One STWIN Core System board (order code: STEVAL-STWINKT1)
- One STLINK-V3MINI debugger
- One USB Micro-B cable
- One STEVAL-STWINWFV1 Wi-Fi expansion board (only for predictive maintenance cloud application)

4.3 Software setup

The following software components are required for the setup of a suitable development environment to create applications for the STEVAL-STWINKT1 kit:

- FP-IND-PREDMNT1 firmware, available on www.st.com.
- A standard user terminal as Putty or Tera Term (v. 4.97 or higher)
- STBLESensor app (only for BLE application)
- An account for the DSH-PREDMNT predictive maintenance dashboard (only for predictive maintenance cloud application)
- Development tool-chain and Compiler. The STM32Cube expansion software supports the three following environments to select from:
  - IAR Embedded Workbench for ARM® (EWARM release 8.50.5) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM release 5.31.0) toolchain + ST-LINK
  - Integrated Development Environment for STM32 (STM32CubeIDE release 1.4.2) + ST-LINK

4.4 Sample application description

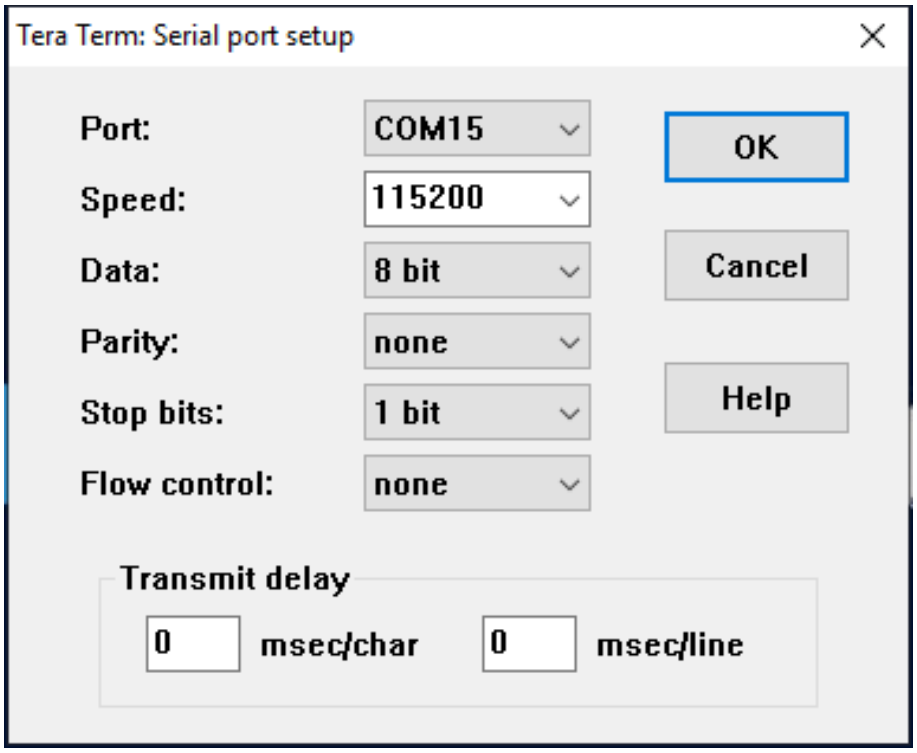
4.4.1 BLE application

A sample application is provided in the *Project* directory for the STEVAL-STWINKT1 evaluation kit and ready to be built projects are available for multiple IDEs (IAR 8.50.5, Keil 5.31.0 and STM32CubeIDE 1.4.2).

4.4.1.1 Terminal window setup

For the STEVAL-STWINKT1 development kit you can set a terminal window up for the appropriate UART communication port, used to control the initialization and the execution phases.

Figure 61. STEVAL-STWINKT1 terminal setup



4.4.1.2 Sample application startup

When you first press the reset button, the application:

- checks whether all the sensors are working
- reads the Meta Data Manager
- initializes the BLE stack connectivity
- initializes the BLE hardware service (adding the temperature, humidity, pressure, 3Dgyroscope, 3D magnetometer, 3D accelerometer and microphone characteristics) and the BLE software service
- initializes the BLE console service adding the stdin/stdout and stderr characteristics
- shows the vibration parameter values present in the Flash memory and sets the accelerometer parameters

Figure 62. **Initialization phase**

When an Android/iOS device is connected to the STWIN Core System board, you can control data transmitted through the board.

Figure 63. **UART console output after a device has been connected to the STEVAL-STWINKT1**

Like the NUCLEO-F446RE demonstrations, this sample is compatible with the STBLESensor Android/iOS applications (ver. 4.2.1or higher) and allows Over-The-Air firmware updates.

4.4.2 Predictive maintenance cloud application

The predictive maintenance cloud application is a software example for STWIN (STEVAL-STWINKT1 plus STEVAL-STWINWFV1) that collects sensor data and performs spectral analysis for the IIS3DWB accelerometer (up to 5 kHz) and the MP23ABS1 analog microphone (up to 80 kHz) signals.

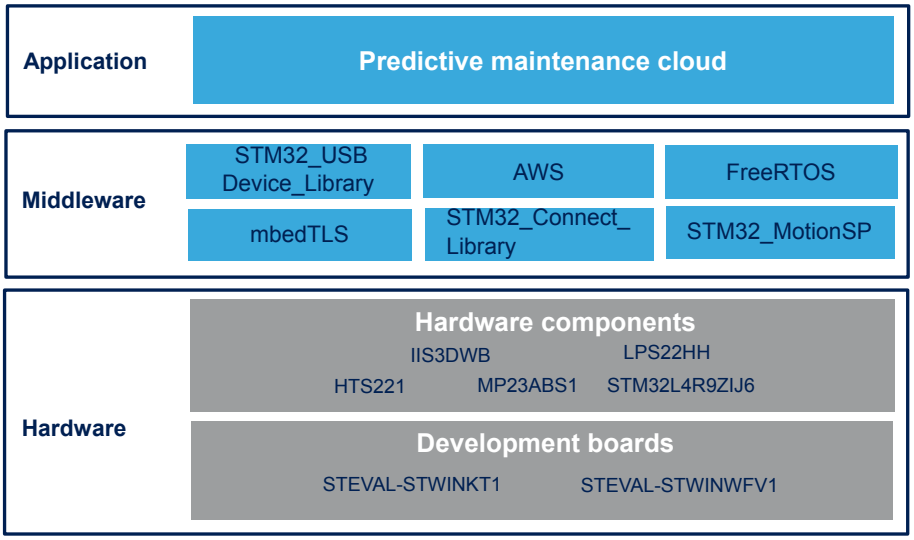
The result of spectral analysis and raw environmental data are sent to the DSH-PREDMNT dashboard exploiting AWS (Amazon Web Service) SDK and MQTT protocol.

You can set the predictive maintenance cloud application parameters through the dashboard.

The application main functions are:

- perform a secure connection to DSH-PREDMNT
- compute the spectrum of audio signal from analog microphone in ultrasound bandwidth up to 80 kHz
- analyze vibrations by computing the spectrum of vibrometer data. Up to 5 KHz
- get the environmental sensor data
- receive configuration parameters from the dashboard

Figure 64. Predictive maintenance cloud application overview

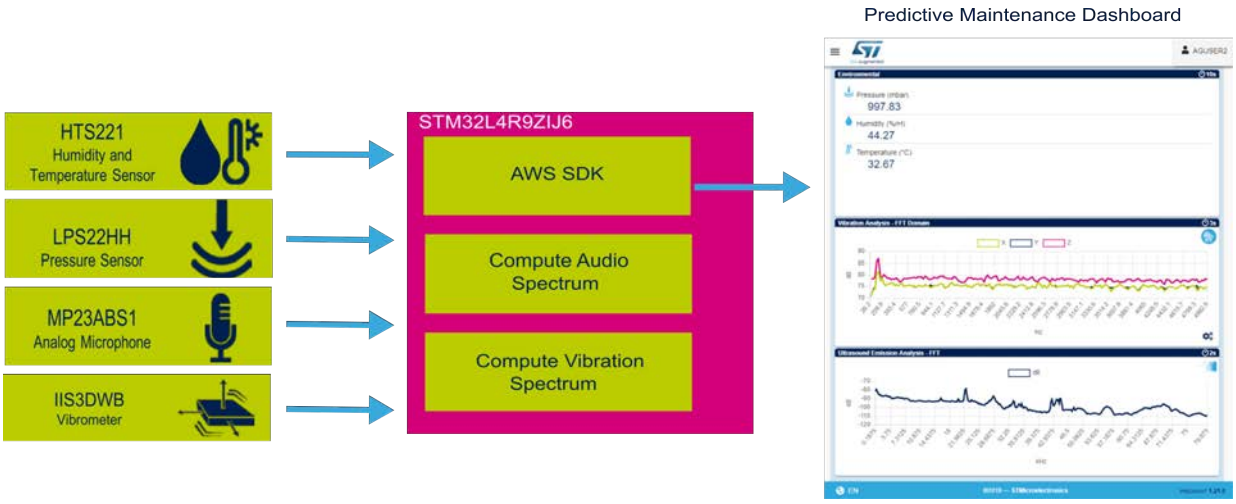


4.4.2.1 *Functional blocks*

The predictive maintenance cloud application connects to the DSH-PREDMNT dashboard exploiting the underlying AWS IoT Core API and defines the following AWS MQTT topics:

- environmental data topic:
  - temperature and humidity (HTS221)
  - pressure (LPS22HH)
- ultrasound analysis topic:
  - audio spectral analysis in ultrasound bandwidth exploiting MP23ABS1 analog microphone
- vibration analysis topic:
  - vibration analysis up to 5 kHz of bandwidth exploiting IIS3DWB MEMS vibrometer

Figure 65. Predictive maintenance cloud application functional blocks



Data are collected and processed in three independent logical tasks:

- **Audio Process Task**
  - acquires signal from the analog microphone:
    - sampling frequency = 192 kHz
    - audio frame = 1 s
  - performs the FFT:
    - window= Hann window
    - FFT size = 1024
  - computes the magnitude of the result of FFT
- **Vibration Process Task**
  - acquires wide band accelerometer signals:
    - sampling frequency = 26.7 kHz
    - vibration frame = 1 s
  - performs the FFT:
    - window= Hann window
    - FFT size = 1024
  - computes the magnitude of the result of FFT
- **Environmental Process Task**
  - acquires signal from environmental sensors:
    - humidity and temperature sampling frequency = 12.5 Hz
    - pressure sampling frequency = 25 Hz

4.4.2.2 *Configuration*

To program the STWIN core system you have to download the relevant version of the STLINK-V3MINI USB driver and follow the procedure below.

Step 1. Plug the STLINK-V3MINI to the STWIN core board through the programming cable available in the STEVAL-STWINKT1.

Step 2. Connect the STLINK-V3MINI to the PC through the USB USB Micro-B male cable.

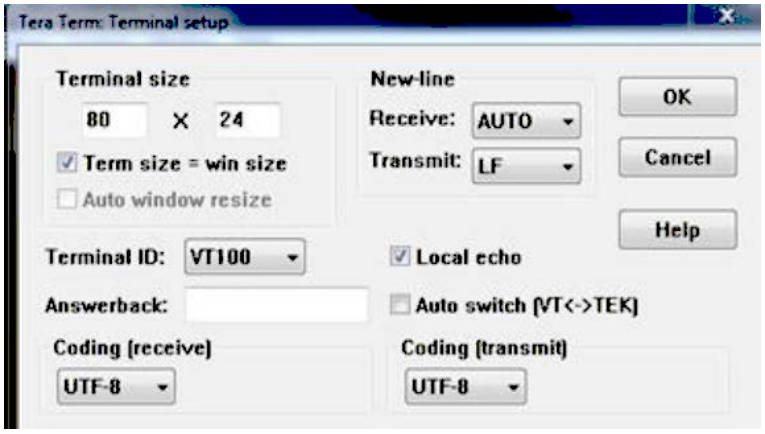
The board is ready to be programmed with the application in *Projects\STM32L4R9ZI-STWIN\Demonstrations\Predictive_Maintenance_WIFI\AWS\Binary\STM32L4R9ZI-STWIN_PredictiveMaintenance_WIFI_v2.0.0.bin*.

Step 3. Once the firmware has been loaded, press the [**Reset**] black button on the STWIN core system.

Step 4. Open Tera Term.

Step 5. Set the parameters as shown below.

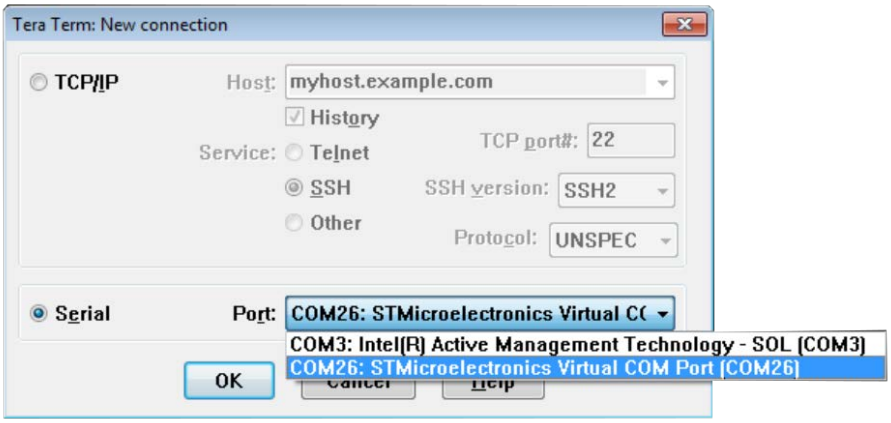
Figure 66. Tera Term terminal setup



Step 6. Click on [**File**]>[**New Connection**].

Step 7. Select the right COM port.

Figure 67. Tera Term COM port selection



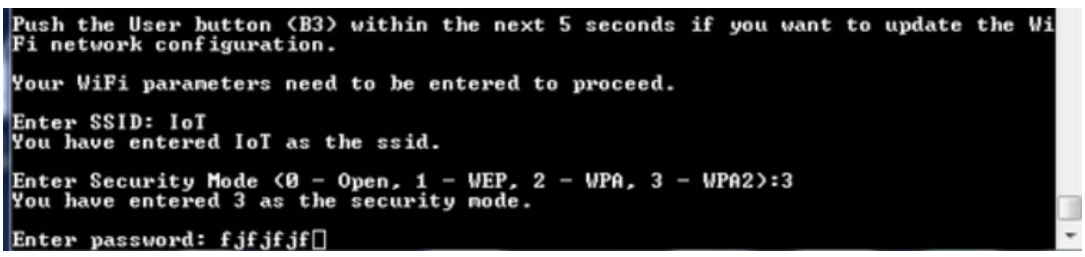
Step 8. Press the [**Reset**] button on the board.

Step 9. Set up Wi-Fi and AWS credentials (see also Section 4.6 Predictive maintenance dashboard (DSH-PREDMNT)).

If the board has never been configured before, the system automatically asks for the needed parameters. Otherwise, you can change the configuration by pressing the USR button within 5 seconds.

Step 10. Insert the SSID, security mode and password of your Wi-Fi network.

Figure 68. Wi-Fi credentials



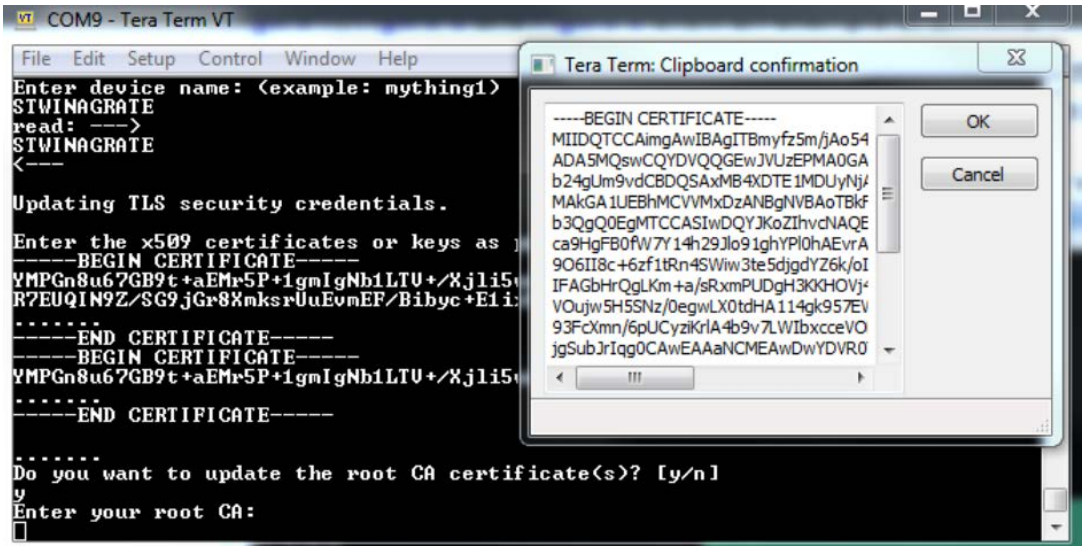
Step 11. Press the USR button again within 5 seconds and insert the AWS IoT Core endpoint.

Figure 69. AWS credentials

Step 12. Enter the security certificates.
You can either drag and drop the file or copy and paste the text.

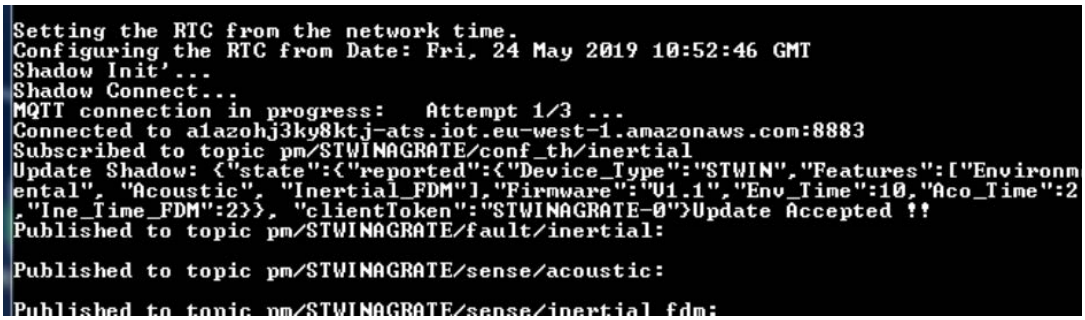
Step 13. Enter root certificate (press Enter Key twice) and public and private keys (press Enter Key once).

Figure 70. AWS security certificates



At the end of the configuration, data are published via MQTT protocol and available in the dashboard page.

Figure 71. Data published via MQTT



Step 14. If the board is battery plugged, use the PWR button to power the board on/off.

- For power on, press the PWR button until the red led turns off (~1 sec)
- For power off, press the PWR button shortly

Figure 72. STWIN core system power button

4.5 Android and iOS STBLESensor client application

RELATED LINKS

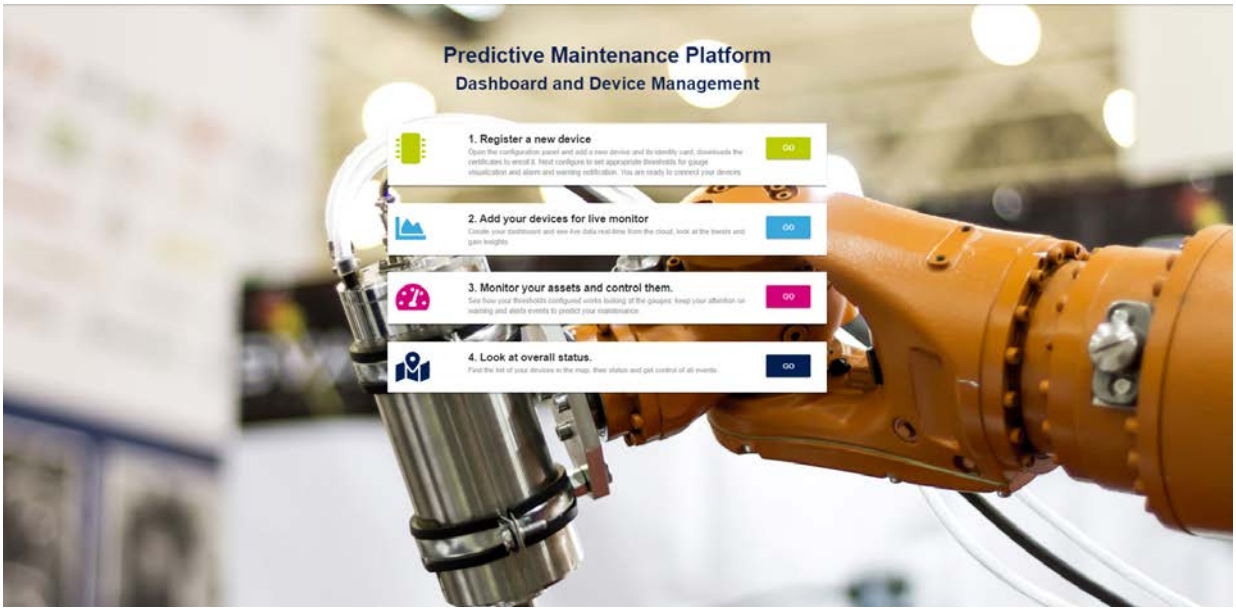
2.10 Android and iOS STBLEsensor client application on page 27

4.6 Predictive maintenance dashboard (DSH-PREDMNT)

The DSH-PREDMNT predictive maintenance dashboard is a cloud application based on AWS services. It provides a highly functional and intuitive interface that is tailored for the collection, visualization and analysis of condition monitoring data from motion and acoustic vibration sensing elements, as well as temperature and other environmental data. You can use the dashboard to plot and graph real-time and historical data, monitor critical operating conditions such as running temperature, and set thresholds for automatic warnings when key parameters exceed acceptable limits.

Figure 73. **DSH-PREDMNT dashboard overview**

- The green button is for:
  - user login and data segregation
  - device and AWS Green Grass Edge
  - registration and configuration:
    - provisioning
    - associations to assets
    - streaming time
    - geolocalization
- The light blue button is for:
  - live data visualization
  - add a device to live monitoring
- The pink button is for:
  - asset health monitoring:
    - data collection
    - historical trend analysis
    - failure thresholds settings for alerts and warnings
- The dark blue button is for asset mapping



To create your virtual device follow the procedure below.

Step 1. Log on to my.st.com.

Step 2. Register a new device by clicking on the green button.

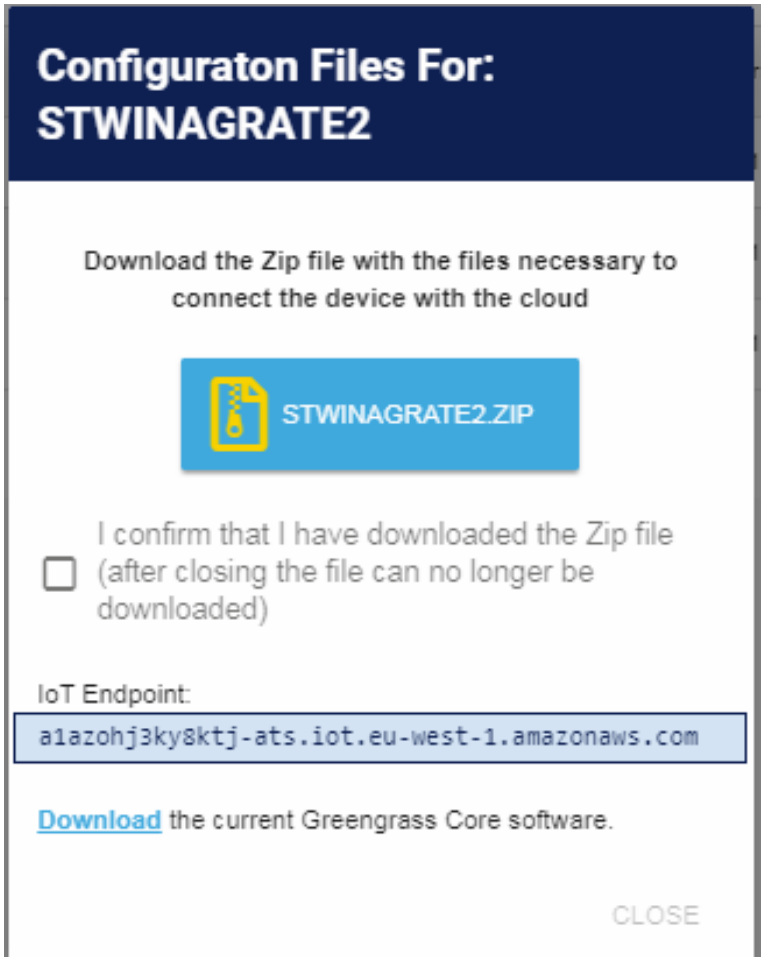
Step 3. Click [**Devices**] and then on the [**+**] button.

Step 4. Fill the form with device information and click [**Register**].

The zip file just downloaded includes the certificates to authenticate your device with the predictive maintenance cloud application to the DSH-PREDMNT dashboard.

Important:
The link highlighted in the figure below refers to the AWS IoT Core endpoint, so you must copy it as it is necessary for the configuration.

Figure 74. DSH-PREDMNT configuration files to connect your device to the cloud



After the configuration, you can see the data streaming coming from your device.

Step 5. Click on the [**Dashboard**] tab drop-down list and select your virtual device.

Revision history

Table 3. **Document revision history**

Date	Version	Changes
27-Mar-2019	1	Initial release.
30-Jul-2019	2	Removed Section 1.1 What is STM32Cube? and Section 1.2 STM32Cube architecture and replaced them with a link in the Introduction. Updated Section 1.1.2 Firmware, Figure 1. FP-IND-PREDMNT1 software architecture, Section 1.3 Folder structure, Section 1.4 Flash management, Section 1.5 The boot process, Section 2 STM32 Nucleo platform, Section 2.8 Software setup, Section 2.9 The installation process, Section 2.10.1 Terminal window setup, Section 2.10 Sample application description, Section 2.10.2 Sample application startup, Section 2.11 Android and iOS STBLEsensor client application, Figure 23. STBLESensor (Android version) menu selection, Section 2.11.5 Predictive maintenance, Figure 29. STBLESensor (Android version) firmware upgrade page, Figure 30. STBLESensor (Android version) firmware update file section, Figure 31. STBLESensor (Android version) feedback during and after FOTA transmission, Figure 32. STBLESensor (Android version) UART console after the FOTA and before update and Section 3 STEVAL-BFA001V1B platform. Added Section 1.1.1 Compatible hardware platforms, System setup, Section 1.5.1 Batch script, Section 3.1 System setup, Section 3.3 Software setup, Section 3.4 The installation process, Section 3.5.1 Terminal window setup, Section 3.5.2 Sample application startup, Section 4 STEVAL-STWINKT1 platform, Section 4.1 STEVAL-STWINWFV1, System setup, Section 4.2 Hardware setup, Section 4.3 Software setup, Section 4.4.1 BLE application, Section 4.4.1.1 Terminal window setup, Section 4.4.1.2 Sample application startup, Section 4.4.2 Predictive maintenance cloud application, Section 4.4.2.2 Configuration, Section 4.4.2.1 Functional blocks, Section 4.5 Android and iOS STBLESensor client application and Section 4.6 Predictive maintenance dashboard (DSH-PREDMNT).
18-May-2020	3	Updated Section 1.1.2 Firmware, Figure 1. FP-INDPREDMNT1 software architecture, Section 1.3 Folder structure, Section 1.5.1 Batch script, Section 2 STM32 Nucleo platform, Section 2.8 Software setup, Section 2.9 The installation process, Section 2.10 Sample application description, Section 2.10.2 Sample application startup, Section 2.11.2 Plot data, Section 2.11.7 Firmware-Over-The-Air update with STBLESensor, Section 3.3 Software setup, Section 4.4.1 BLE application and Section 4.4.1.2 Sample application startup. Added Section 2.2 X-NUCLEO-CCA02M2 expansion board, Section 2.3 X-NUCLEO-BNRG2A1 expansion board, Section 2.7.1 X-NUCLEO-IKS01A3 setup and Section 2.7.2 X-NUCLEO-BNRG2A1 setup.
09-Dec-2020	4	Added STEVAL-BFA001V2B compatibility information.

Date	Version	Changes
		Updated Section 1.5.1 Batch script, Section 2.6 Hardware setup, Section 2.7 Software setup, Section 2.8 The installation process, Section 2.9 Sample application description, Section 2.9.1 Terminal window setup, Section 2.9.2 Sample application startup, Section 2.10.1 Main page, Section 2.10.2 Plot data, Section 2.10.3 Settings, serial and debug console, Section 2.10.4 FTT amplitude, Section 2.10.6 RSSI and battery, Section 2.10.7 Firmware-Over-The-Air update with STBLESensor, Section 3.1 System setup, Section 3.2 Hardware setup, Section 3.3 Software setup, Section 3.4 The installation process, Section 3.5.2 Sample application startup, Section 4.3 Software setup, Section 4.4.1 BLE application, Section 4.4.1.1 Terminal window setup and Section 4.4.1.2 Sample application startup. Added Section 3 STEVAL-BFA001V2B platform.
07-Jul-2021	5	Updated Section 1.2 Architecture, Section 2.8 The installation process, Section 2.9.2 Sample application startup, Section 2.10.3 Settings, serial and debug console, Section 2.10.6 RSSI and battery, Section 2.10.8 Firmware-Over-The-Air update with STBLESensor and Section 4.4.1.2 Sample application startup. Added Section 2.10.7 Board configuration.

Contents

1 **FP-IND-PREDMNT software expansion for STM32Cube 2**
1.1 Overview 2
1.1.1 Compatible hardware platforms 2
1.1.2 Firmware 3
1.2 Architecture 3
1.3 Folder structure 5
1.4 Flash management 5
1.5 The boot process 6
1.5.1 Batch script 7
1.6 Firware-Over-The-Air (FOTA) update 10
1.7 APIs 10
2 **STM32 Nucleo platform 11**
2.1 STM32 Nucleo 11
2.2 X-NUCLEO-CCA02M2 expansion board 12
2.3 X-NUCLEO-BNRG2A1 expansion board 12
2.4 X-NUCLEO-IKS01A3 expansion board 13
2.5 STEVAL-MKI182V2 evaluation board 14
2.6 Hardware setup 15
2.6.1 X-NUCLEO-IKS01A3 setup 17
2.6.2 X-NUCLEO-BNRG2A1 setup 18
2.7 Software setup 22
2.8 The installation process 22
2.9 Sample application description 23
2.9.1 Terminal window setup 23
2.9.2 Sample application startup 24
2.10 Android and iOS STBLEsensor client application 27
2.10.1 Main page 27
2.10.2 Plot data 28
2.10.3 Settings, serial and debug console 28
2.10.4 FTT amplitude 33

2.10.5 Predictive maintenance . . . 34
2.10.6 RSSI and battery . . . 35
2.10.7 Board configuration . . . 36
2.10.8 Firmware-Over-The-Air update with STBLESensor . . . 44
3 STEVAL-BFA001V2B platform . . . 49
3.1 System setup . . . 49
3.2 Hardware setup . . . 49
3.3 Software setup . . . 51
3.4 The installation process . . . 51
3.5 Sample application description . . . 51
3.5.1 Terminal window setup . . . 51
3.5.2 Sample application startup . . . 51
4 STEVAL-STWINKT1 platform . . . 59
4.1 STEVAL-STWINWFV1 . . . 59
4.2 Hardware setup . . . 60
4.3 Software setup . . . 60
4.4 Sample application description . . . 61
4.4.1 BLE application . . . 61
4.4.2 Predictive maintenance cloud application . . . 64
4.5 Android and iOS STBLESensor client application . . . 68
4.6 Predictive maintenance dashboard (DSH-PREDMNT) . . . 68
Revision history . . . 71

List of figures

Figure 1. FP-IND-PREDMNT1 software architecture . . . 4
Figure 2. FP-IND-PREDMNT1 package folder structure . . . 5
Figure 3. FP-IND-PREDMNT1 Flash structure for NUCLEO-F446RE . . . 6
Figure 4. Bootloader folder content . . . 7
Figure 5. FP-IND-PREDMNT1 boot sequence . . . 7
Figure 6. Bootloader and FP-IND-PREDMNT1 installation . . . 8
Figure 7. FP-IND-PREDMNT1 dump process . . . 9
Figure 8. STM32 Nucleo board . . . 11
Figure 9. X-NUCLEO-CCA02M2 expansion board . . . 12
Figure 10. X-NUCLEO-BNRG2A1 BLE expansion board . . . 13
Figure 11. X-NUCLEO-IKS01A3 MEMS and environmental sensor expansion board . . . 14
Figure 12. STEVAL-MKI182V2 evaluation board . . . 14
Figure 13. STM32 Nucleo plus X-NUCLEO-CCA02M2 expansion board . . . 15
Figure 14. STM32 Nucleo plus X-NUCLEO-CCA02M2 and X-NUCLEO-BNRG2A1 expansion boards . . . 16
Figure 15. STM32 Nucleo plus X-NUCLEO-CCA02M2, X-NUCLEO-BNRG2A1 and X-NUCLEO-IKS01A3 expansion boards plus STEVAL-MKI182V2 evaluation board . . . 17
Figure 16. X-NUCLEO-IKS01A3 expansion board - solder bridges to open (in yellow) and to close (in pink) . . . 18
Figure 17. Connection between X-NUCLEO-BNRG2A1 expansion board and ST-LINK/V2-1 . . . 19
Figure 18. STSW-BNRGFLASHER - SWD tab . . . 20
Figure 19. STSW-BNRGFLASHER - mass erase . . . 21
Figure 20. STSW-BNRGFLASHER - Flash button . . . 22
Figure 21. Binary folder container . . . 23
Figure 22. NUCLEO-F446RE terminal setup . . . 24
Figure 23. Initialization phase . . . 25
Figure 24. UART console output when a device is connected to the board . . . 26
Figure 25. STBLESensor (Android version) main page following BLE connection . . . 27
Figure 26. STBLESensor (Android version) magnetometer plot . . . 28
Figure 27. STBLESensor (Android version) menu selection . . . 29
Figure 28. STBLESensor (Android version) node configuration: local name . . . 30
Figure 29. STBLESensor (Android version) Debug console (stdin/stdout/stderr) . . . 31
Figure 30. STBLESensor (Android version) Debug console (stdin/stdout/stderr): Help Command . . . 32
Figure 31. STBLESensor (Android version) FFT amplitude page for predictive maintenance (plot, details and settings . . . 33
Figure 32. STBLESensor (Android version) FFT amplitude page for acoustic analysis (plot and details) . . . 34
Figure 33. STBLESensor (Android version) predictive maintenance page . . . 35
Figure 34. STBLESensor (Android version) battery and RSSI information . . . 36
Figure 35. Board configuration selection . . . 37
Figure 36. Board configuration options . . . 38
Figure 37. Board firmware version . . . 39
Figure 38. Hardware and firmware information . . . 40
Figure 39. Setting the board name . . . 41
Figure 40. Read Custom Commands . . . 42
Figure 41. List of custom commands . . . 43
Figure 42. Custom commands - example (FFT_Overlapping) . . . 44
Figure 43. STBLESensor (Android version) firmware upgrade page . . . 45
Figure 44. STBLESensor (Android version) firmware update file selection . . . 46
Figure 45. STBLESensor (Android version) feedback during and after FOTA transmission . . . 47
Figure 46. STBLESensor (Android version) UART console after the FOTA and before update . . . 48
Figure 47. STEVAL-BFA001V2B predictive maintenance reference kit . . . 49
Figure 48. STEVAL-UKI001V2 top view . . . 50
Figure 49. Connection between ST-LINK-V3MINI and STEVAL-UKI001V2 . . . 50
Figure 50. Power supply connection (without IO-Link master board) . . . 50
Figure 51. STEVAL-BFA001V2B complete setup kit connection in stand-alone mode . . . 51

Figure 52. STEVAL-BFA001V2B serial terminal output after reset . . . 53
Figure 53. STEVAL-BFA001V2B parameter settings and configuration . . . 54
Figure 54. STEVAL-BFA001V2B parameter overview and analysis with time domain array . . . 55
Figure 55. STEVAL-BFA001V2B 2nd analysis with frequency domain array . . . 55
Figure 56. STEVAL-BFA001V2B time domain predictive maintenance results . . . 56
Figure 57. STEVAL-BFA001V2B frequency domain sub-range status and all status results . . . 57
Figure 58. STEVAL-BFA001V2B environmental measurement and parameter list . . . 58
Figure 59. STEVAL-STWINKT1 devlopment kit . . . 59
Figure 60. STEVAL-STWINWFV1 expansion board . . . 60
Figure 61. STEVAL-STWINKT1 terminal setup . . . 61
Figure 62. Initialization phase . . . 62
Figure 63. UART console output after a device has been connected to the STEVAL-STWINKT1 . . . 63
Figure 64. Predictive maintenance cloud application overview . . . 64
Figure 65. Predictive maintenance cloud application functional blocks . . . 65
Figure 66. Tera Term terminal setup . . . 66
Figure 67. Tera Term COM port selection . . . 66
Figure 68. Wi-Fi credentials . . . 66
Figure 69. AWS credentials . . . 67
Figure 70. AWS security certificates . . . 67
Figure 71. Data published via MQTT . . . 67
Figure 72. STWIN core system power button . . . 68
Figure 73. DSH-PREDMNT dashboard overview . . . 69
Figure 74. DSH-PREDMNT configuration files to connect your device to the cloud . . . 70

List of tables

Table 1. Compatible hardware platform overview . . . 2
Table 2. X-NUCLEO-BNRG2A1 and ST-LINK/V2-1 pin connections . . . 19
Table 3. Document revision history . . . 71

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. For additional information about ST trademarks, please refer to www.st.com/trademarks. 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.

© 2021 STMicroelectronics – All rights reserved