Getting started with the STM32Cube function pack for the Pro Mode of wireless multi sensor development kits

Introduction

FP-SNS-STBOX1 is an STM32Cube function pack for the Pro Mode of the STWIN.box and for Sensortile.box Pro multi-sensors and wireless connectivity development kit for any intelligent IoT node. Wireless Industrial Node Development Kit, which helps you to build custom applications.

The package includes pressure, relative humidity, temperature, accelerometer, gyroscope and magnetometer sensors, as well as an analog and digital microphones, and the SPBTLE-1S Bluetooth low energy system-on-chip application processor.

With the STEVAL-STWINBX1 and STEVAL-MKBOXPRO kits with BLE connectivity, you can monitor and log the algorithm output and sensor data using the STBLESensor app.

The software runs on the STM32 microcontroller and includes all the necessary drivers for the STEVAL-STWINBX1 and STEVAL-MKBOXPRO evaluation kits.

The software is available also on GitHub, where the users can signal bugs and propose new ideas through [**Issues**] and [**Pull Requests**] tabs.

Related links

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

1 FP-SNS-STBOX1 software expansion for STM32Cube

1.1 Overview

FP-SNS-STBOX1 key features are:

- Complete samples on how to:
  - create one implementation based on transmitting the data with BLE
  - use the dual bank flash feature for rollback after a FOTA update
  - program the ISM330DHCX (for STEVAL-STWINBX1) and LSM6DSV16X (for STEVAL-MKBOXPRO) machine learning core (MLC) or theirs finite state machine (FSM), control the output of theirs registers and transmit the results via BLE
  - easily send the data via BLE
  - save the sensor data to the SD card
- Compatible with STBLESensor application for Android/iOS, to perform sensor and audio data reading, motion algorithm feature demo, and FOTA via BLE connectivity
- Sample implementation available for the STEVAL-STWINBX1 and STEVAL-MKBOXPRO kits
- Compatible with STM32CubeMX, can be downloaded from st.com and installed directly into STM32CubeMX
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

The software gathers the temperature, humidity, pressure, motion sensor drivers for the:

- IIS2DLPC, IIS2MDC, IIS3DWB, ISM330DHCX, IIS2ICLX, ILPS22QS and STTS22H the STEVAL-STWINBX1
- STTS22H, LPS22DF, LSM6DSV16X, LIS2DU12, LIS2MDL and ST25DV04K (board rev. A) or ST25DV64KC (board rev. B) for STEVAL-MKBOXPRO

1.2 Architecture

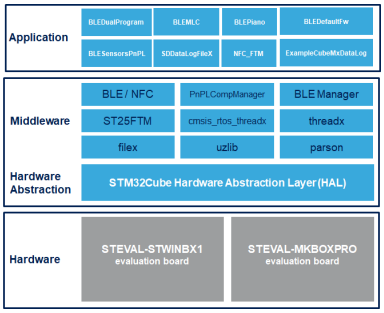
This software is based on the STM32CubeHAL. It extends STM32Cube by providing a board support package (BSP) for the BLE, sensors, microphone and middleware components for communication with other BLE devices.

It also provides some sample applications to demonstrate how to implement custom applications using the STWIN.box and Sensortile.box-Pro Pro Mode.

The software layers used by the application software to access and use the SensorTile.box, STWIN.box and Sensortile.box-Pro are:

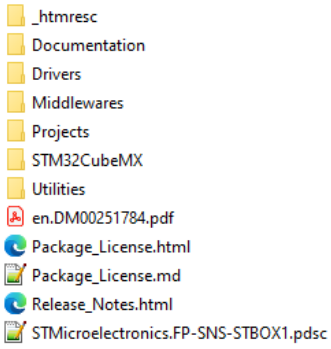
- the **STM32Cube HAL layer**, which provides a simple, generic, multi-instance set of application programming interfaces (APIs) to interact with the upper application, library and stack layers. It has generic and extension APIs and is directly built around a generic architecture and allows successive 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 guarantees an easy portability on other devices.
- the **board support package** (BSP) layer, which 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-SNS-STBOX1 software architecture



1.3 Folder structure

Figure 2. FP-SNS-STBOX1 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 and BlueNRG-LP Bluetooth low energy, ST25 fast transfer mode (ST25FTM), NDEF devices, Azure RTOS FileX and ThreadX, CMSIS-RTOS v2 wrapper implementation, PnPLCompManager used to handle PnP-like commands, BLE Manager Library for providing an APIs to manage the BLE services, Parson json library and uzlil compression/decompression.
- **Projects** contains applications and samples for creating customized Pro Mode applications for the STWIN.box - SensorTile wireless Industrial Node Development Kit (STEVAL-STWINBX1) and Sensortile.box-Pro multi-sensors and wireless connectivity development kit (STEVAL-MKBOXPRO) for any intelligent IoT node.
  The Available applications for STEVAL-STWINBX1 are:
  - **BLEMLC**: Example on how to program the ISM330DHCX Machine Learning Core (MLC) and Finite State Machine (FSM) (compatible with FoTA)
  - **BLESensorsPnPL**: Simplest way for BLE transmission of Sensor's data customizing the demos using PnPL-Like messages (compatible with FoTA)
  - **BLEDefaultFw**: Default firmware for STEVAL-STWINBX1 (compatible with FoTA)
  - **NFC_FTM**: Example on how to use ST25 Fast Transfer Memory protocol for making the firmware update
  - **SDDataLogFileX**: SD-Card data Log using FileX and ThreadX
  - **ExampleCubeMxDataLog**: Serial Data Log project for CubeMX

  The available applications for STEVAL-MKBOXPRO are
  - **BLEDualProgram**: Secure (PIN) BLE Firmware Over The Air update (FoTA)
  - **BLEMLC**: Example on how to program the LSM6DSV16X Machine Learning Core (MLC) and Finite State Machine (FSM) (compatible with FoTA)
  - **BLESensorsPnPL**: Simplest way for BLE transmission of Sensor's data customizing the demos using PnPL-Like messages (compatible with FoTA)
  - **BLEPiano**: Example of using bluetooth it is possible to play Music Notes on STEVAL-MKBOXPRO
  - **NFC_FTM**: Example on how to use ST25 Fast Transfer Memory protocol for making the firmware update
  - **SDDataLogFileX**: SD-Card data Log using FileX and ThreadX

  All projects are available for IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM-STR) and STM32CubeIDE multi-OS development tool.

1.4 APIs

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

2 Sample and application projects

FP-SNS-STBOX1 contains:

- five application projects for STEVAL-STWINBX1 in the *Projects\STEVAL-STWINBX1\Applications directory*
- six application project for STEVAL-MKBOXPRO in the *Projects\STEVAL-MKBOXPRO\Applications directory*

2.1 STEVAL-STWINBX1 application projects

2.1.1 BLEMLC application

The BLEMLC application shows how to program the ISM330DHCX machine learning core (MLC) and its finite state machine (FSM).

Default program for MLC that could be taken from: https://github.com/STMicroelectronics/STMems_Machine_Learning_Core/tree/master/

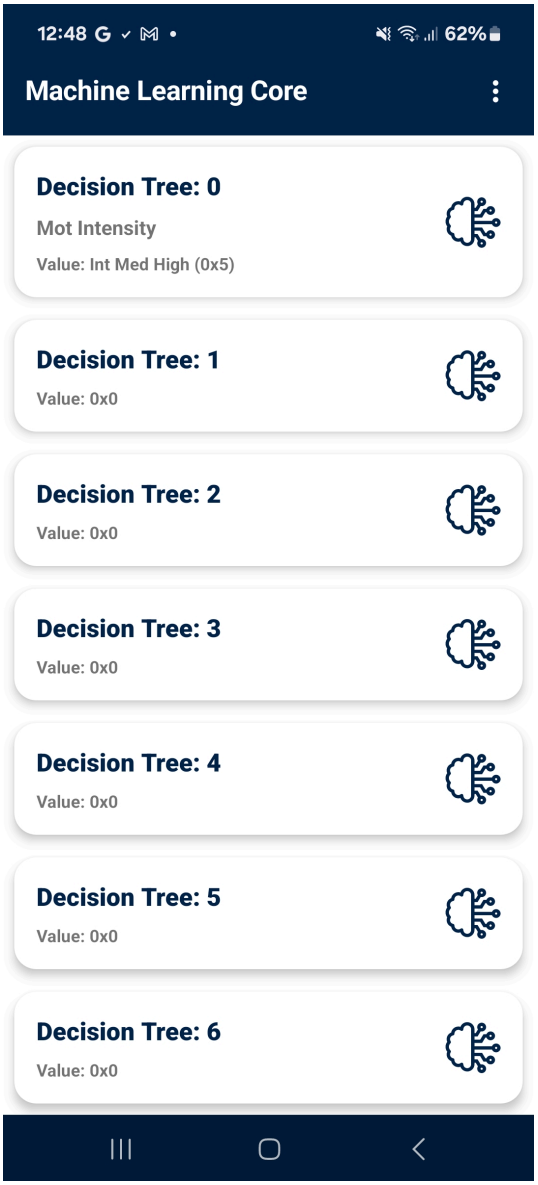
In particular, this application programs the accelerometer to run an activity recognition algorithm for the MLC, and to run a 4D position recognition algorithm for the FSM. For both, the results are sent to the STBLESensor application (8 MLC output register values and 16 FSM output register values, respectively).

The BLEMLC also enables the USBD Virtual Com Port (VCP)to visualize printf to a terminal (for example, Tera Term) to control the initialization and connection phases.

To enable VCP, it is necessary to recompile the code, enabling the compilation define called: STBOX1_ENABLE_PRINTF at the beginning of BLEMLC/Inc/STBOX1_config.h file.

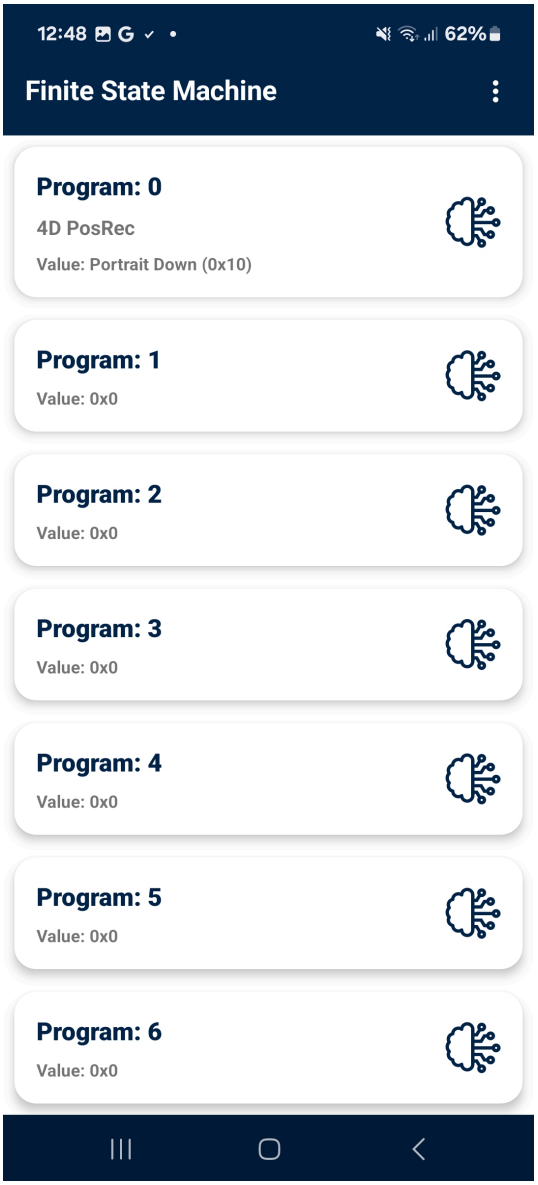
To enable the output of the MLC decision trees, you have to select the STBLESensor [Machine Learning Core] tab.

Figure 3. **ISM330DHCX MLC: values of the 8 output registers**



The [Finite State Machine] tab shows the output of the different programs running on the FSM.

Figure 4. **ISM330DHCX FSM: values of the 16 output registers**



The application is compatible with FOTA procedure.

2.1.2 BLESensorsPnPL application

The purpose of this application is to show how to is possible to customize each demo inside the STBLESensor application using the PnP-Like messages.

This code sends by default the same sensors' values used for STBLESensor application.

But the firmware model associated to this board present on the catalog published on git hub:

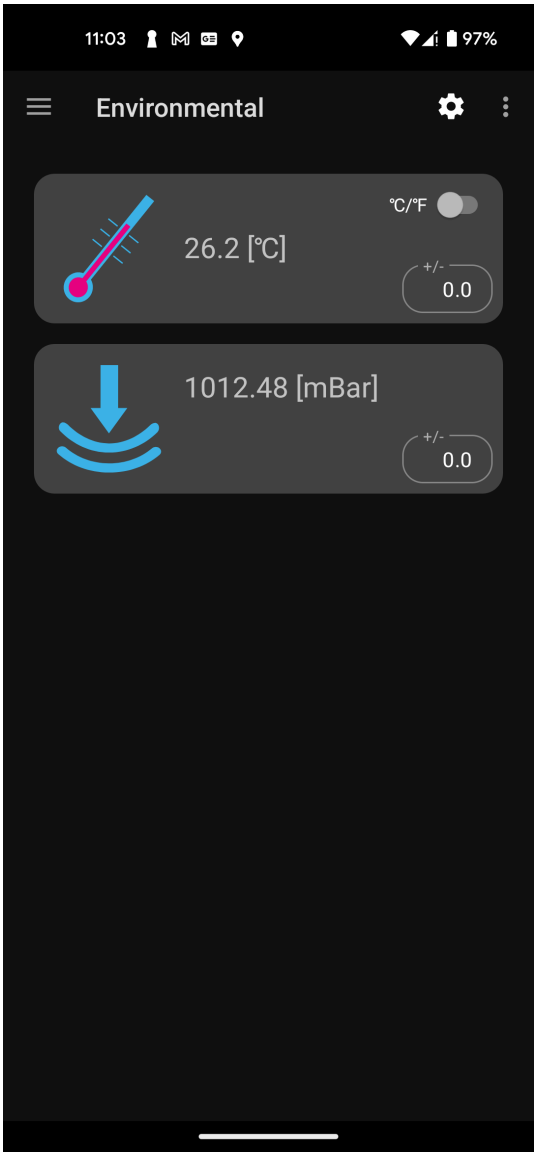
https://github.com/STMicroelectronics/appconfig contains one field:

"dtmi": "dtmi:appconfig:steval_stwinbx1:BLESensorPnPL;1",

that said to the STBLESensor Android/iOS application, that this firmware could be controlled with some PnP-Like messages and it could customize some demos present on STBLESensor Android/iOS application.

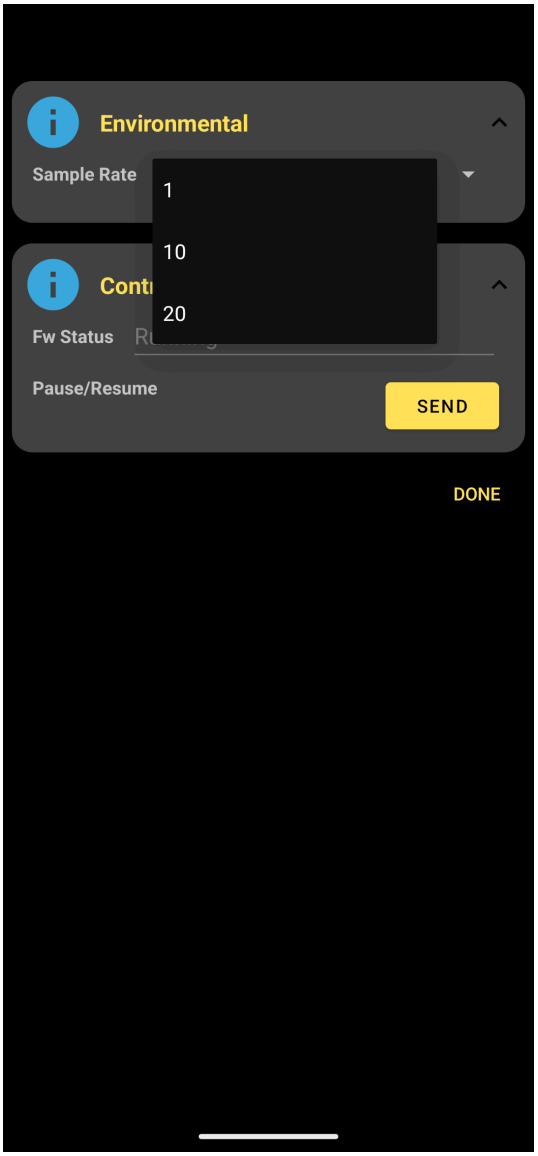
For example, for this program, the environmental demo contains a new configuration gear to the up right of the application.

Figure 5. **Environmental demo with customization gear**



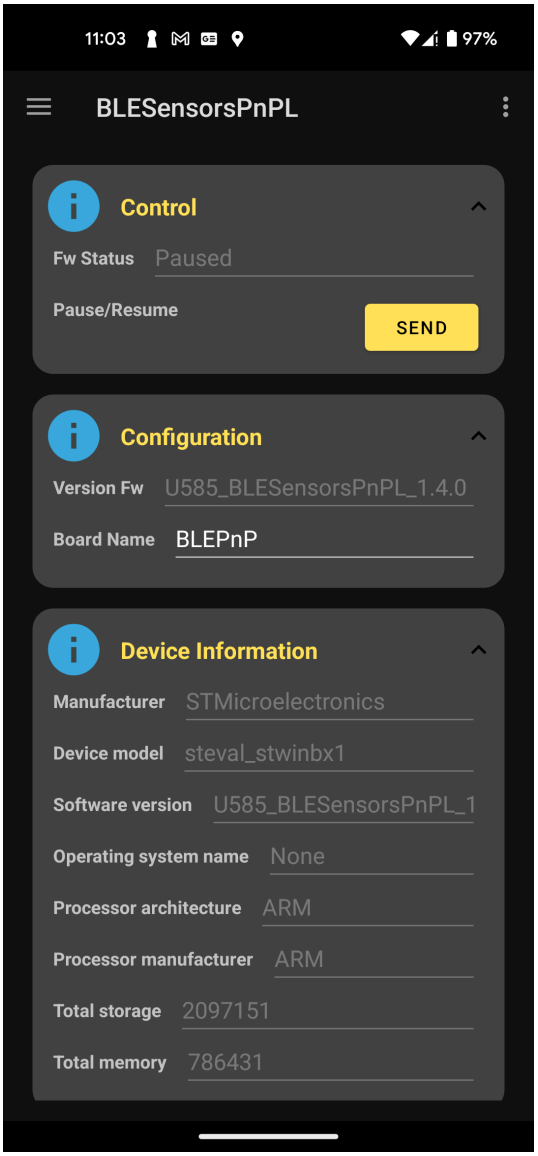
Pressing this gear is possible to open the configuration section for this demo.

Figure 6. **Environmental demo customization for BLESensorsPnPL**



Where we could change the Bluetooth® Low Energy transmission rate (from 1, to 10, to 20 Hz) and where we could decide to put in pause/resume the Bluetooth® Low Energy sensors' values trasmissions. And there is also another section enabled by default, and that has like name the program (in this case BLESensorsPnPL.

Figure 7. **BLESensorsPnPL configuration page**



Where we could also change the board name used in Bluetooth advertise.

2.1.3 BLEDefaultFw application

This is the default firmware preloaded on the STEVAL-STWINBX1 it supports:

- it is compatible with FOTA procedure
- Use the NFC for making the automatic connection with the STBLESensor Android/iOS application writing one NDEF with a deep link
- It reads the sensors values sending them to the STBLESensor Android/iOS application

2.1.4 NFC_FTM application

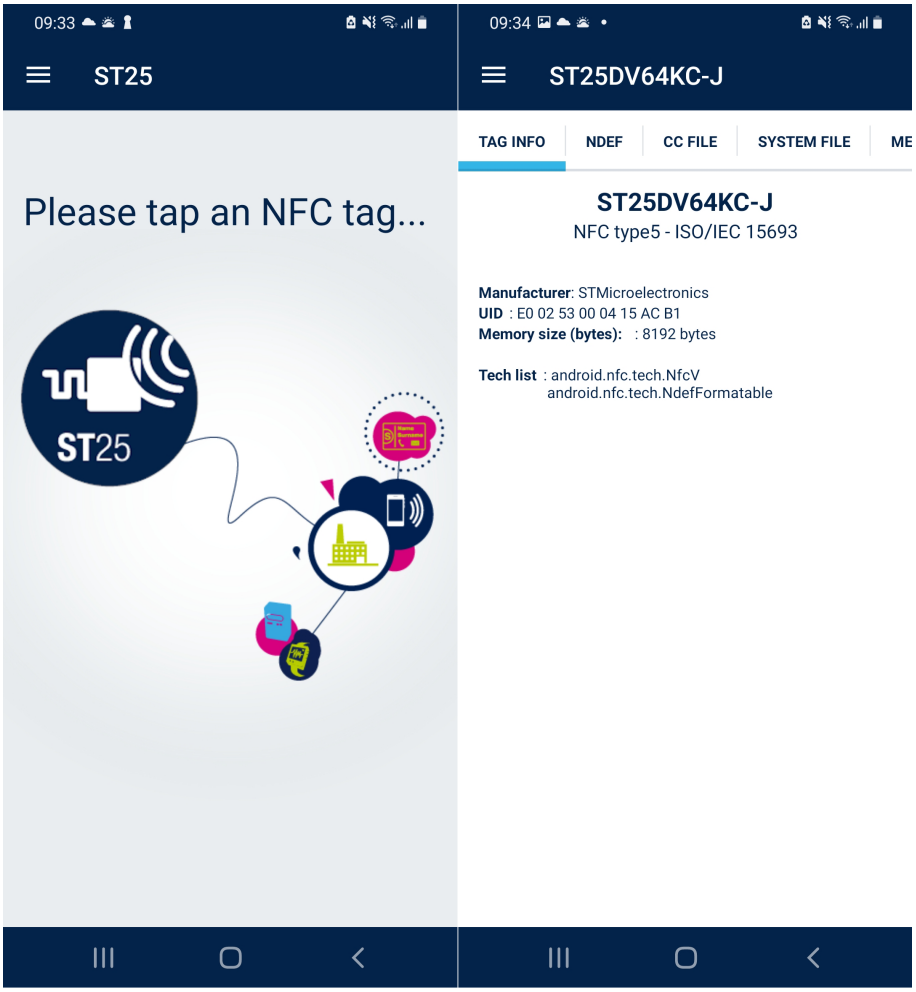
The NFC_FTM is one application that uses the ST25 Fast Transfer Memory protocol for using the NFC for making the Firmware update.

2.1.4.1 FirmwareUpdate: Android and iOS sample client application

Start the ST25 NFC tag mobile application and put your mobile phone on the board.

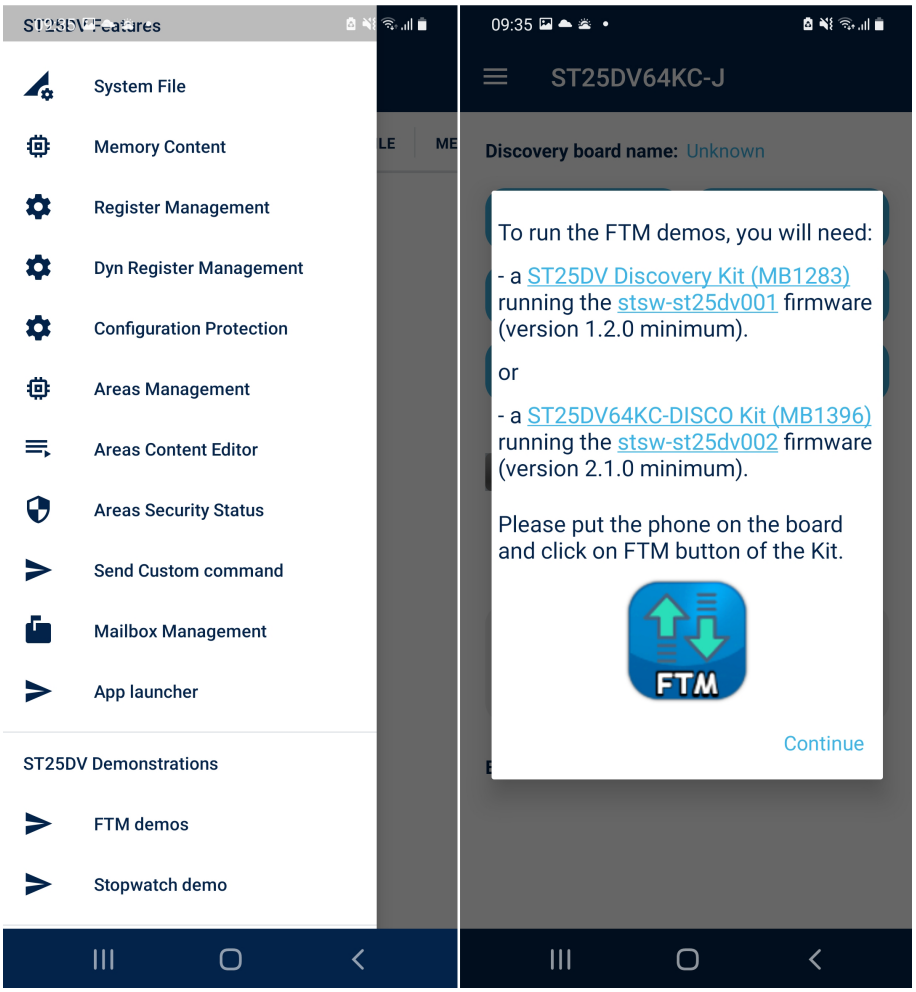
Select the menu option.

Figure 8. **ST25 NFC tag app - tag info**



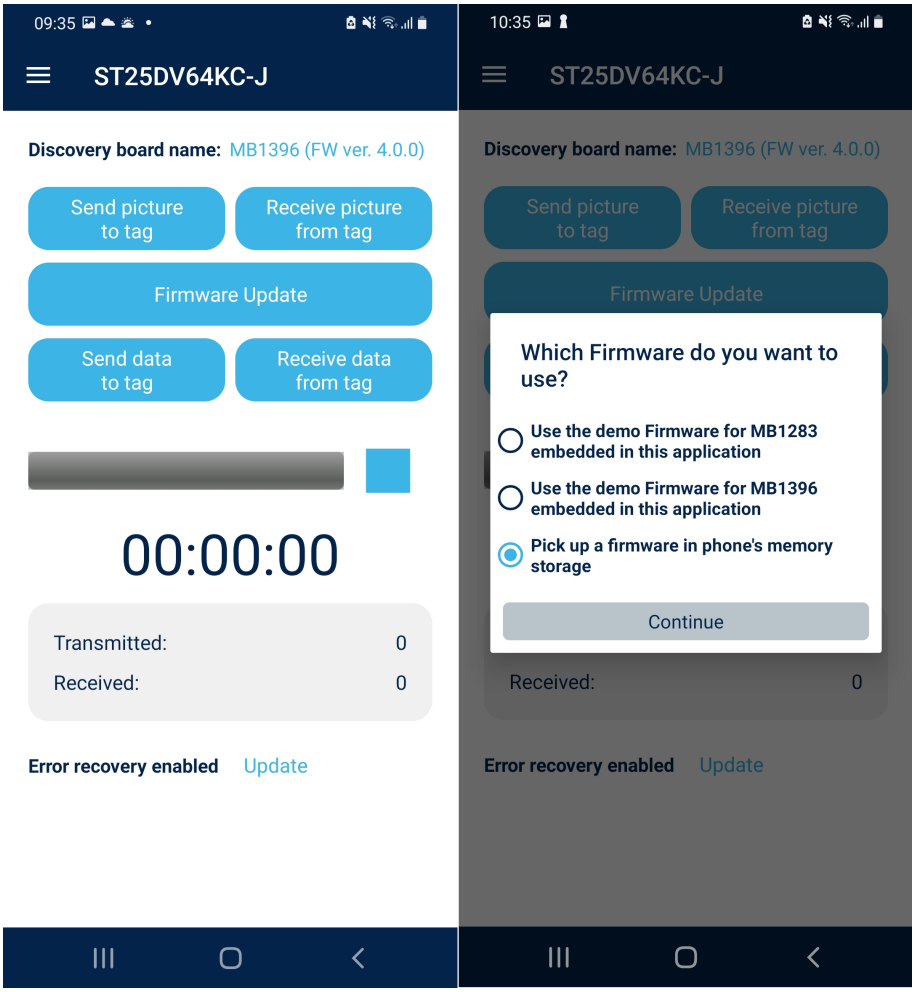
Select "FTM Demos" and click "continue".

Figure 9. **ST25 NFC tag app - FTM demos**



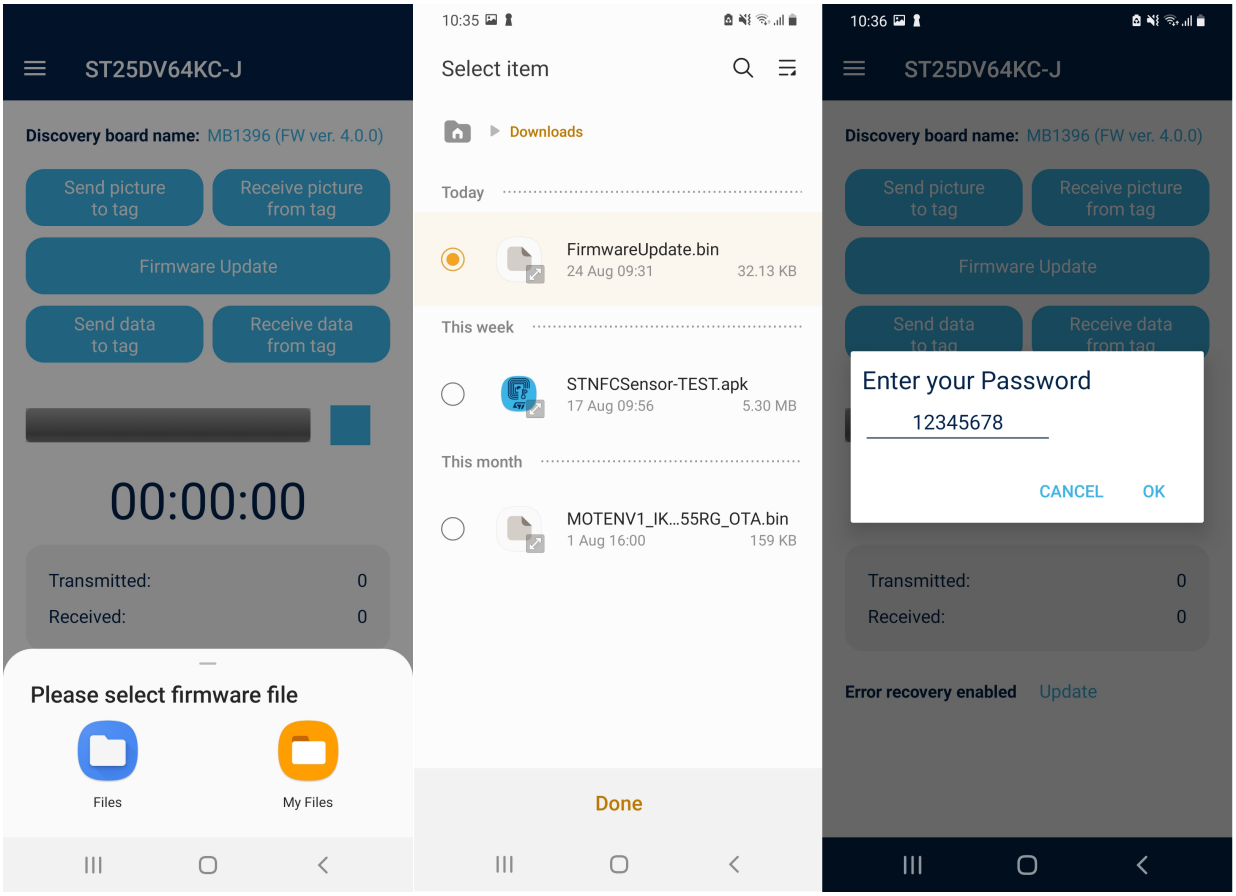
Select "Firmware Update", choose “Pick up a firmware in phone’s memory storage" and click "continue".

Figure 10. **ST25 NFC tag app - firmware update**



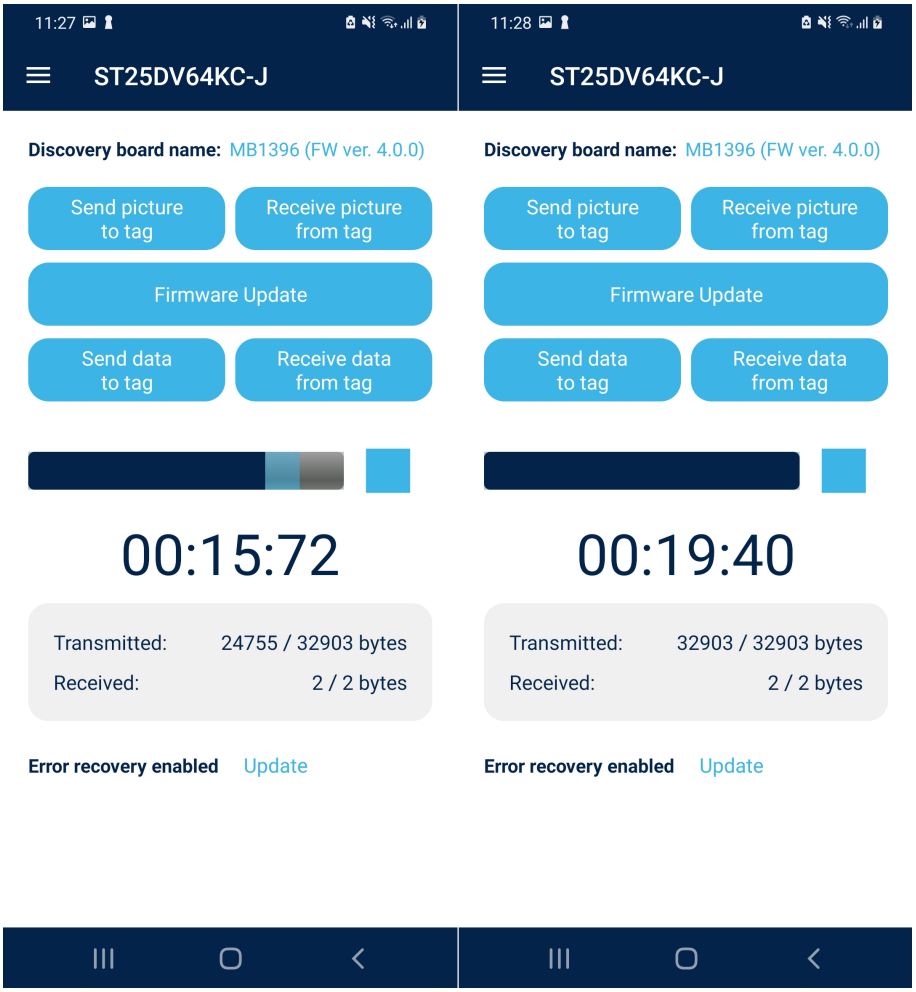
Select the firmware file and enter the password (12345678 is the default one).

Figure 11. **ST25 NFC tag app - password entering**



Wait for the firmware update to finish.

Figure 12. **ST25 NFC tag app - firmware update complete**



2.1.5 SDDataLogFileX application

The purpose of this application is to show an example about using FileX and ThreadX for saving the output of the inertial and environmental sensors present on the board, and the output of the Digital microphone, to the SD-card

After the boot sequence it is possible to press the User button for starting/stopping the SD data log.

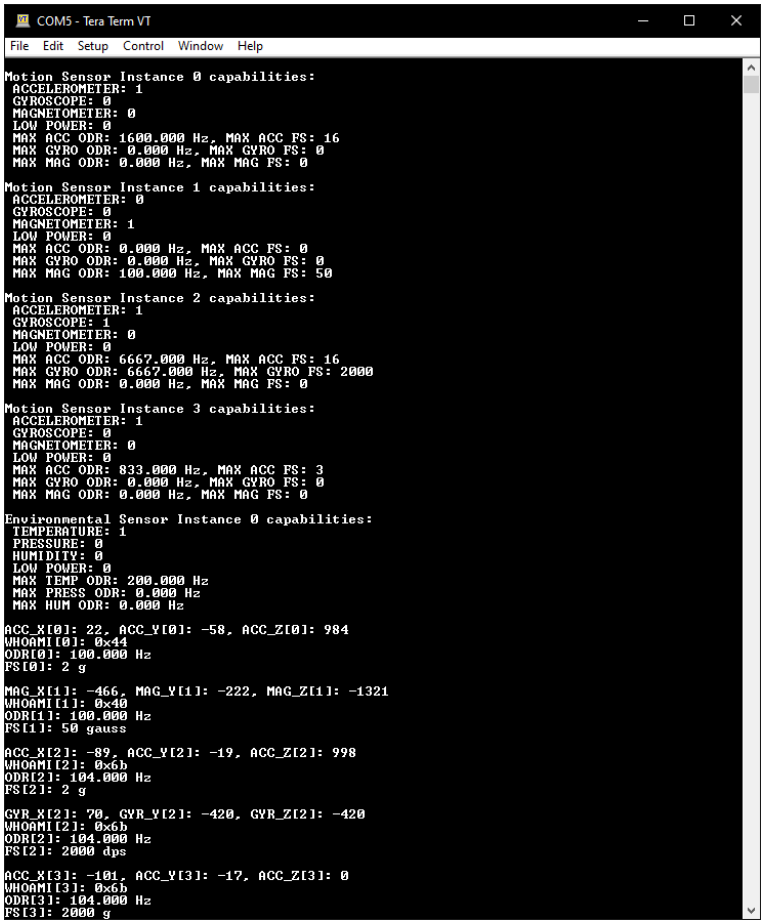
At the end of each logging there will be 2 different files saved on SD-card:

- Sens000.csv: it stores the values of Acc/Gyro/Mag/Pressure/Temperature/Humidity at 100Hz
- Mic000.wav: it stores the wave file for Analog Microphone at 48 KHz

2.1.6 ExampleCubeMxDataLog application

The ExampleCubeMxDataLog is one example of .ioc Project for CubeMX and it implement a simple Serial Data Logger that prints out the sensors values to the UART interfaces Figure 13. Serial output of ExampleCubeMxDataLog.

Figure 13. **Serial output of ExampleCubeMxDataLog**



2.2 STEVAL-MKBOXPRO application projects

2.2.1 BLEDualProgram

The BLEDualProgram application shows how to use the dual bank flash features to allow Firmware-Over-the-Air update of a running program without using the BootLoader.

Before starting the boot procedure (which can start from bank1 or bank2), it is necessary to change STM32 MCU user bytes.

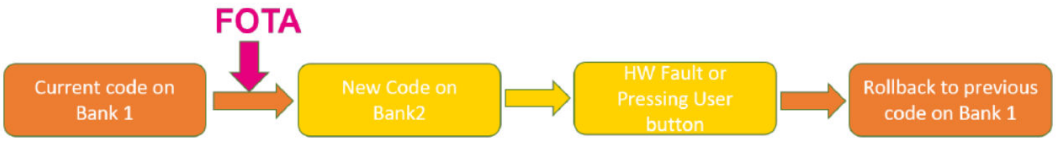
Once settings have been changed, the BLEDualProgram receives the new firmware from the STBLESensor application, saves it on one flash bank (either bank1 or bank2) and performs a reboot executing the new code saved on the other flash bank.

Even if the BootLoader allows more flexibility as you can split the flash memory into any number of regions, each program related to a specific region can run in that region only.

The BLEDualProgram application, however, can swap among different flash banks and each program can run in any flash memory bank.

BLEDualProgram does not erase the previous version of the code after the update and it allows the rollback to the previous program with the swap bank.

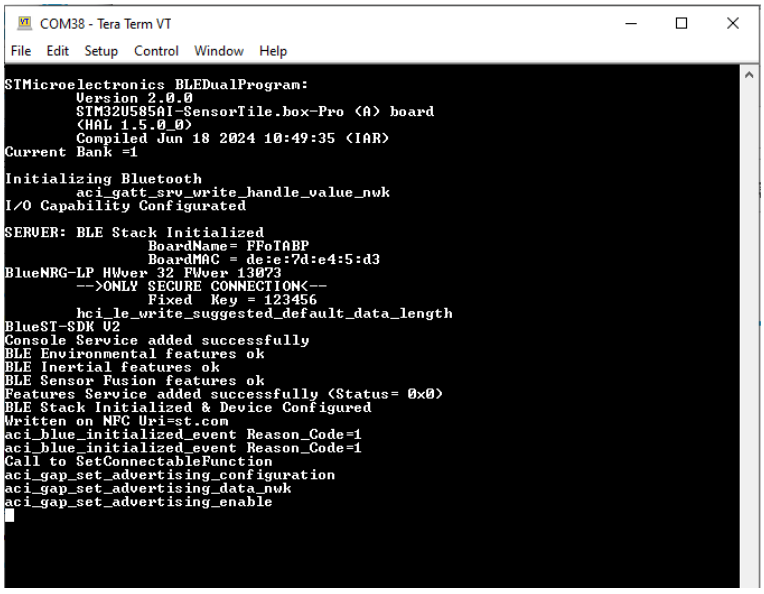
Figure 14. **Rollback function**



Because the STEVAL-MKBOXPRO has the NFC with ST25DV04K (board rev. A) component or ST25DV64KC (board rev .B) component, it use the user button for changing the information saved on NFC.

At the beginning the BLEDualProgram starts enabling all the Sensors and creating all the BLE characteristics necessary for speaking with the STBLESensor application for Android and iOS and it writes on the NFC one NDEF Uri that contains the url for www.st.com web page

Figure 15. BLEDualProgram start

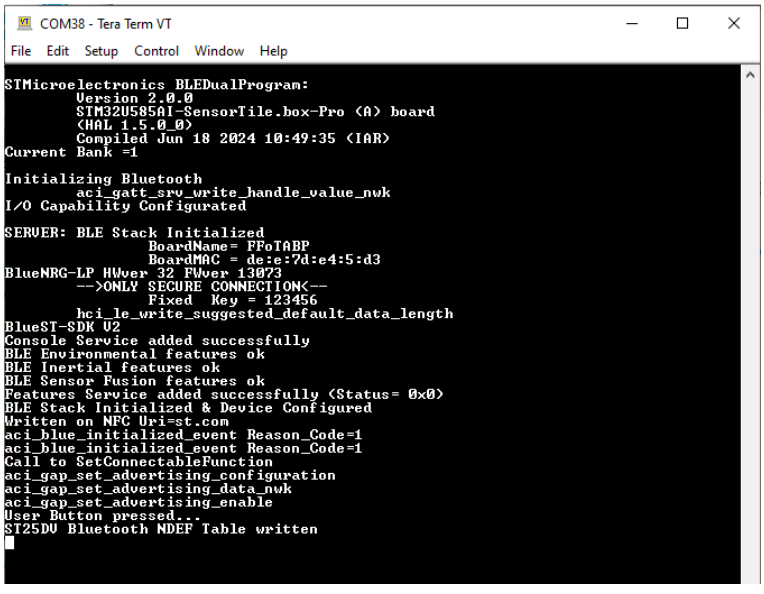


After pressing the User Button, the BLEDualProgam change the NDEF present on NFC writing a Deep Link that has the following format:

stapplication: //connect?Pin=123456&Add=AA:BB:CC:DD:EE:FF

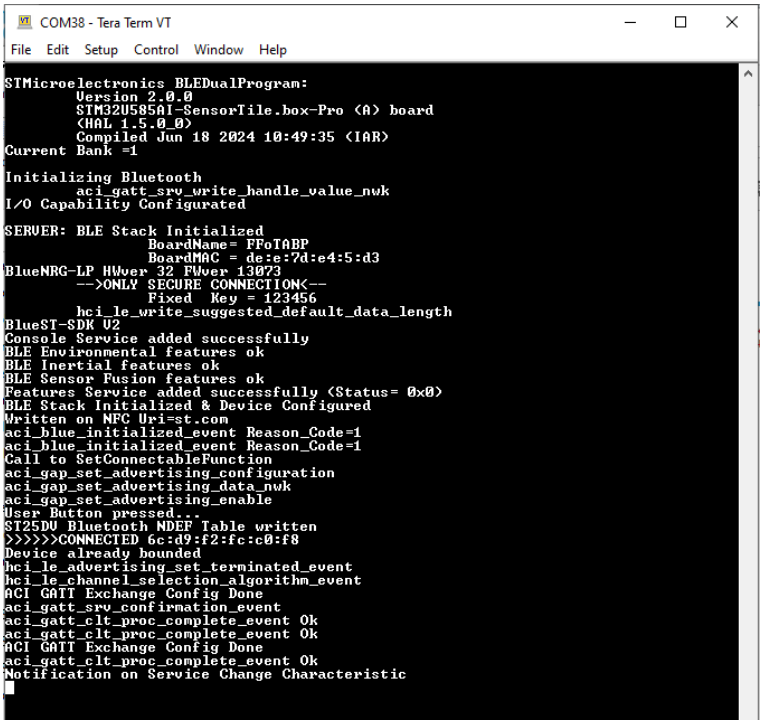
Where the 123456 will be the real pin and AA:BB:CC:DD:EE:FF will be the real BLE MAC used by the board.

Figure 16. BLEDualProgram after pressing the User Button



Reading the NFC content, the ST BLE Sensors application will make the automatic connection to the board using the right PIN.

Figure 17. **BLEDualProgram after the connection**



All the other applications for STEVAL-MKBOXPRO present on this package, and that use the Bluetooth, are compatible with the same Firmware Over the Air (FoTA) update used by BLEDualProgram.

And using the boards firmwares catalog published on:

https://github.com/STMicroelectronics/appconfig

The STBLESensor Application could propose one update of one new version of the running firmware present on the board, or it could also propose other firmwares that are compatible with the same board and way to make the FoTA, downloading them in one automatic way.

It’s necessary to go to “Board Configuration” section and under “Board Control” click the “Firmware Download” Section.

Figure 18. **STEVAL-MKBOXPRO - Board Configuration Section**

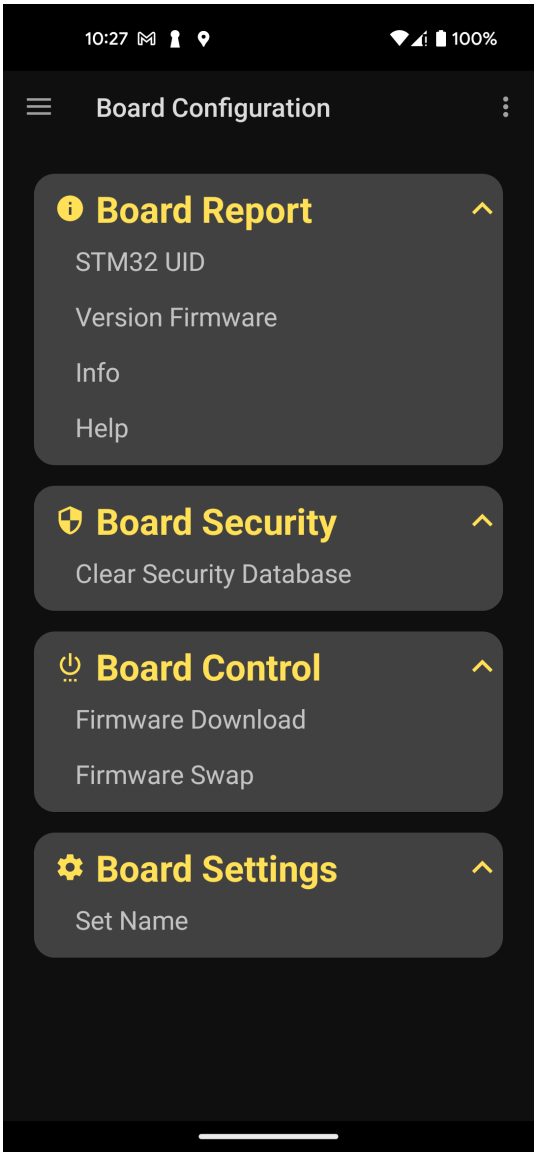


Figure 19. **Dual bank flash status**

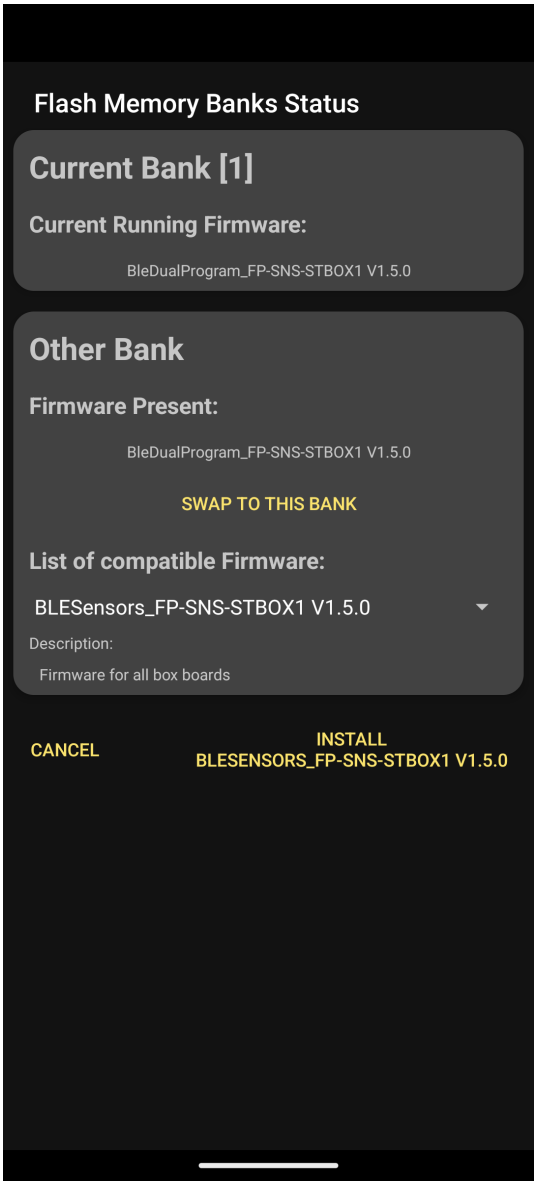
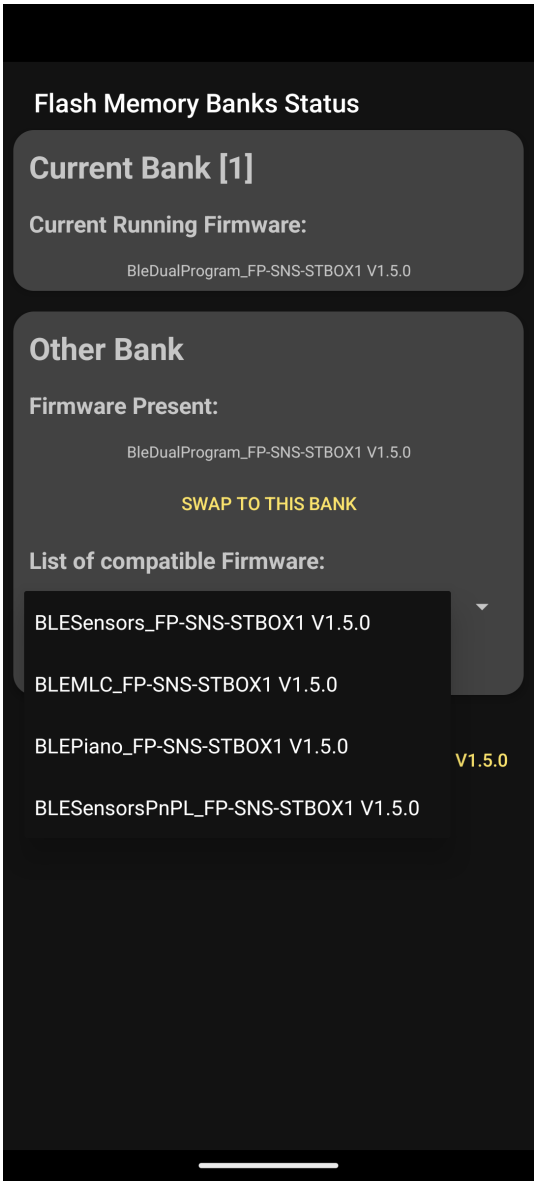


Figure 20. **Possibility to select which firmware we want to download for making the FoTA**



2.2.2 BLEMLC

BLEMLC application for STEVAL-MKBOXPRO is similar to one for STEVAL-STWINBX1 with the following differences:

- Like default program for MLC is used the Motion Intensity algorithm for LSM6DSV16X accelerometer that could be taken from: https://github.com/STMicroelectronics/STMems_Machine_Learning_Core/tree/master/application_examples/lsm6dsv16x

2.2.3 BLESensorsPnPL

The purpose of this application is to show how to is possible to customize each demo inside the ST BLE Sensors application using the PnP-Like messages. This code sends by default the same sensors' values used for STBLESensors application But the Firmware model associated to this board present on the catalog published on github:https://github.com/STMicroelectronics/appconfig

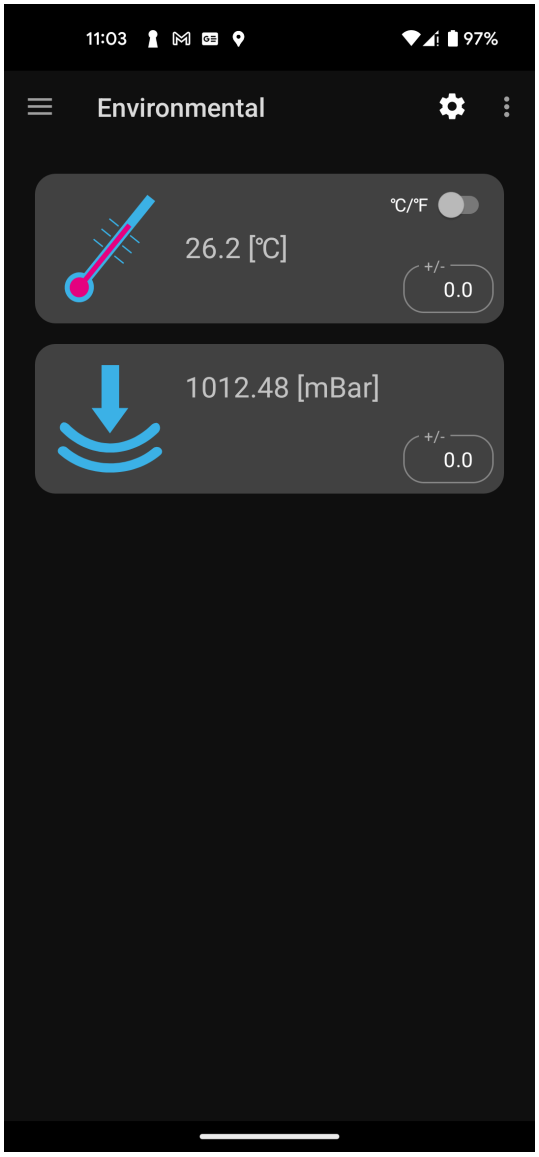
contains one field:

"dtmi": "dtmi:appconfig:steval_mkboxpro:BLESensorPnPL;1"

that said to the ST BLE Sensors Android/iOS application, that this firmware could be controlled with some PnP-Like messages and it could customize some demos present on ST BLE Sensors Android/iOS application.

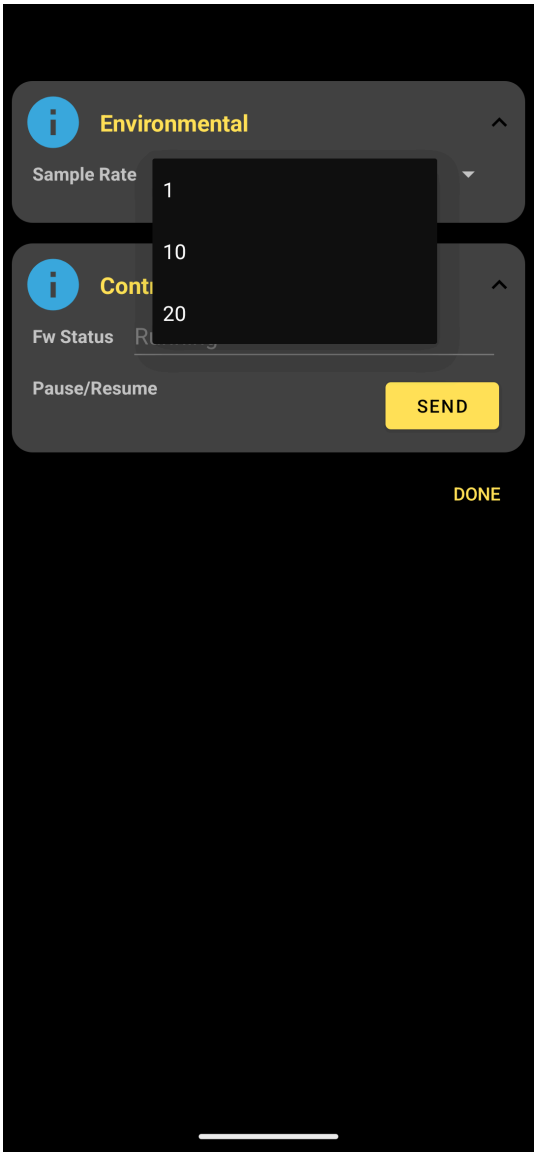
For example, for this program, the Environmental demo contains a new configuration gear to the up right of the application.

Figure 21. Environmental demo with customization gear



Pressing this gear is possible to open the configuration section for this demo.

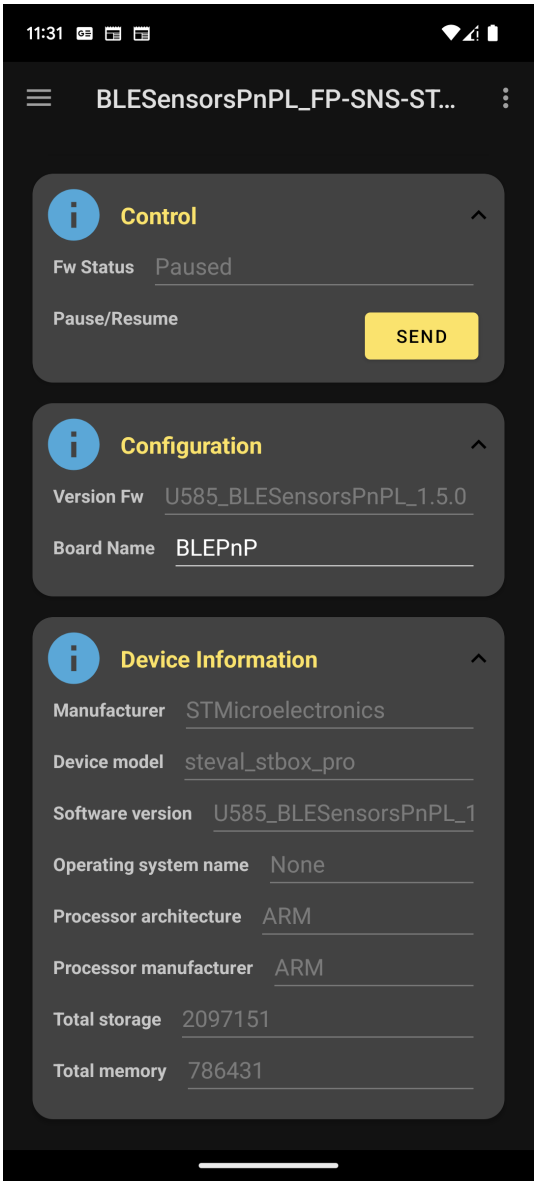
Figure 22. **Environmental demo customization for BLESensorsPnPL**



Where we could change the BLE transmission rate (from 1, to 10, to 20 Hz) and where we could decide to put in pause/resume the BLE sensors' values trasmissions.

And there is also another section enabled by default, and that has like name the Program (in this case BLESensorsPnPL).

Figure 23. **BLESensorsPnPL configuration page**

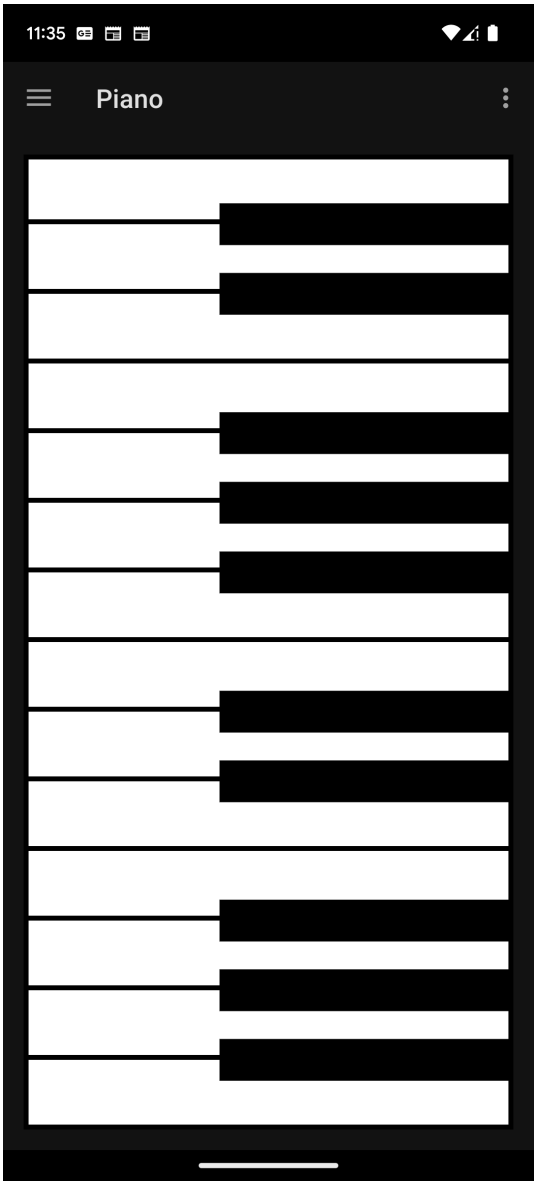


Where we could also change the board name used in Bluetooth advertise.

2.2.4 BLEPiano

This is one example that shows how it's possible to play music note using the buzzer present on STEVAL-MKBOXPRO board using the ST BLE Sensors Android/iOS application and it's compatible with FoTA procedure.

Figure 24. **BLEPiano**



2.2.5 NFC_FTM

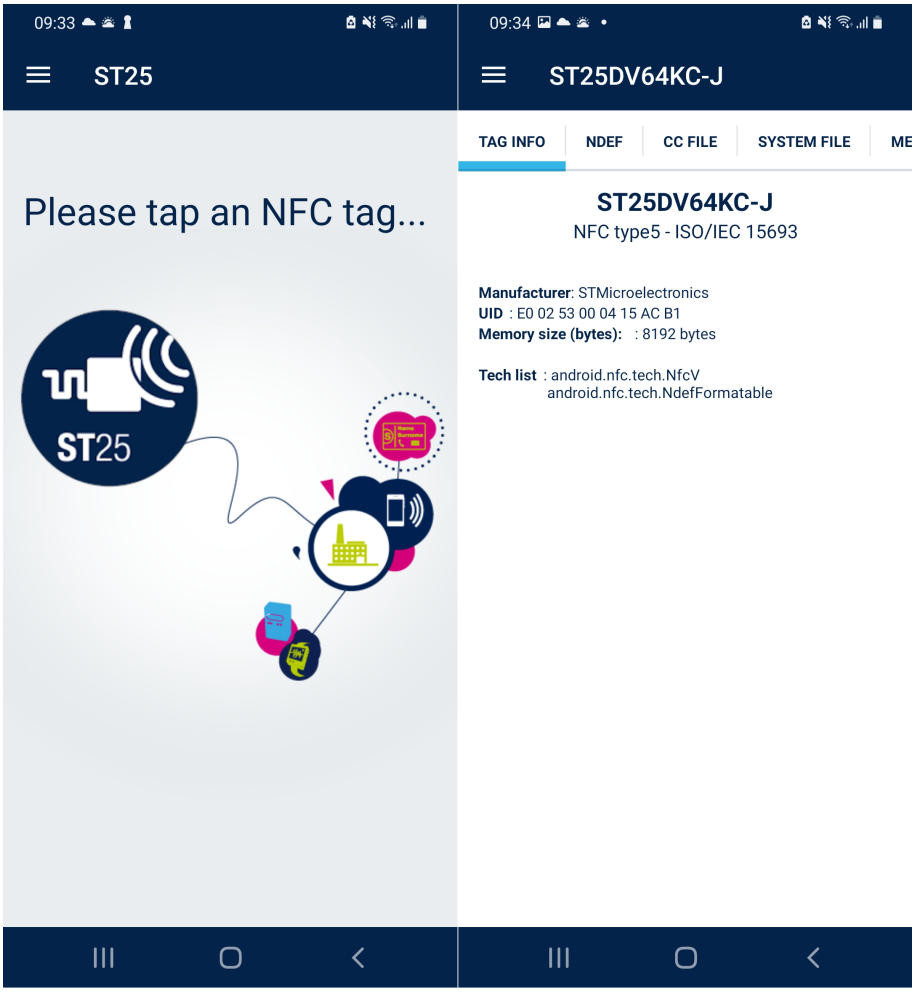
The NFC_FTM is one application that uses the ST25 Fast Transfer Memory protocol for using the NFC for making the Firmware update.

2.2.5.1 FirmwareUpdate: Android and iOS sample client application

Start the ST25 NFC tag mobile application and put your mobile phone on the board.

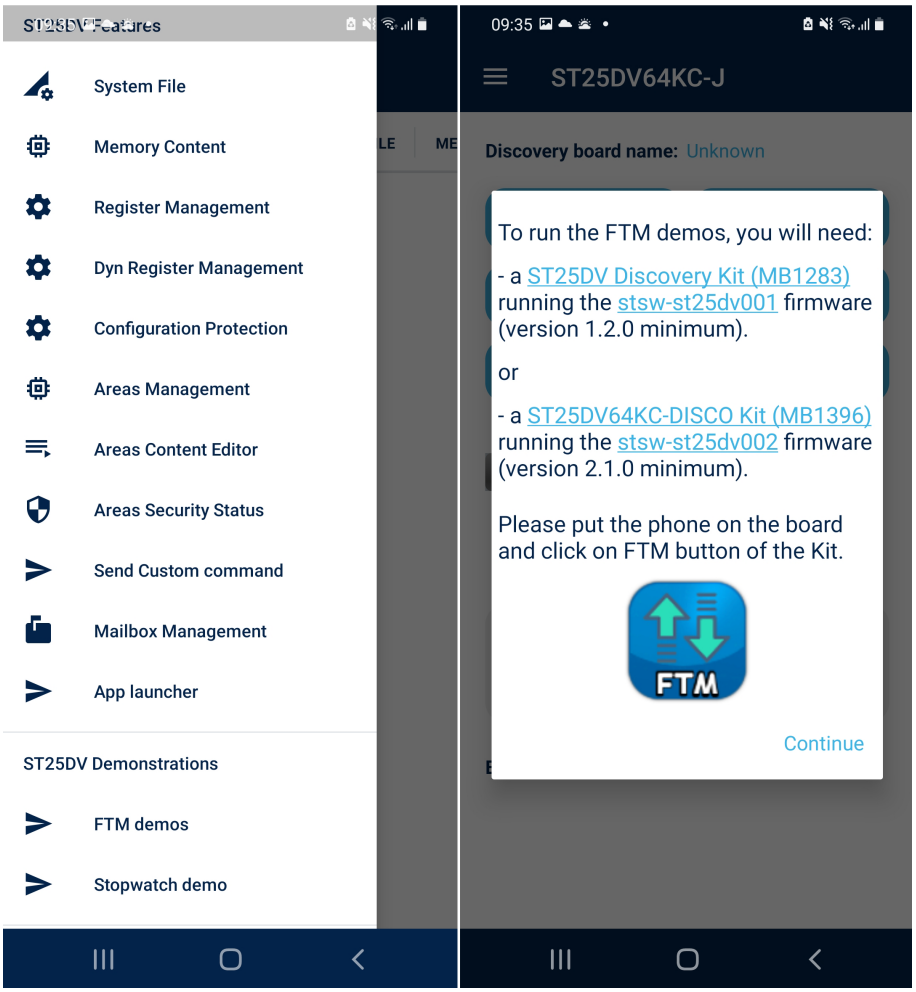
Select the menu option.

Figure 25. **ST25 NFC tag app - tag info**



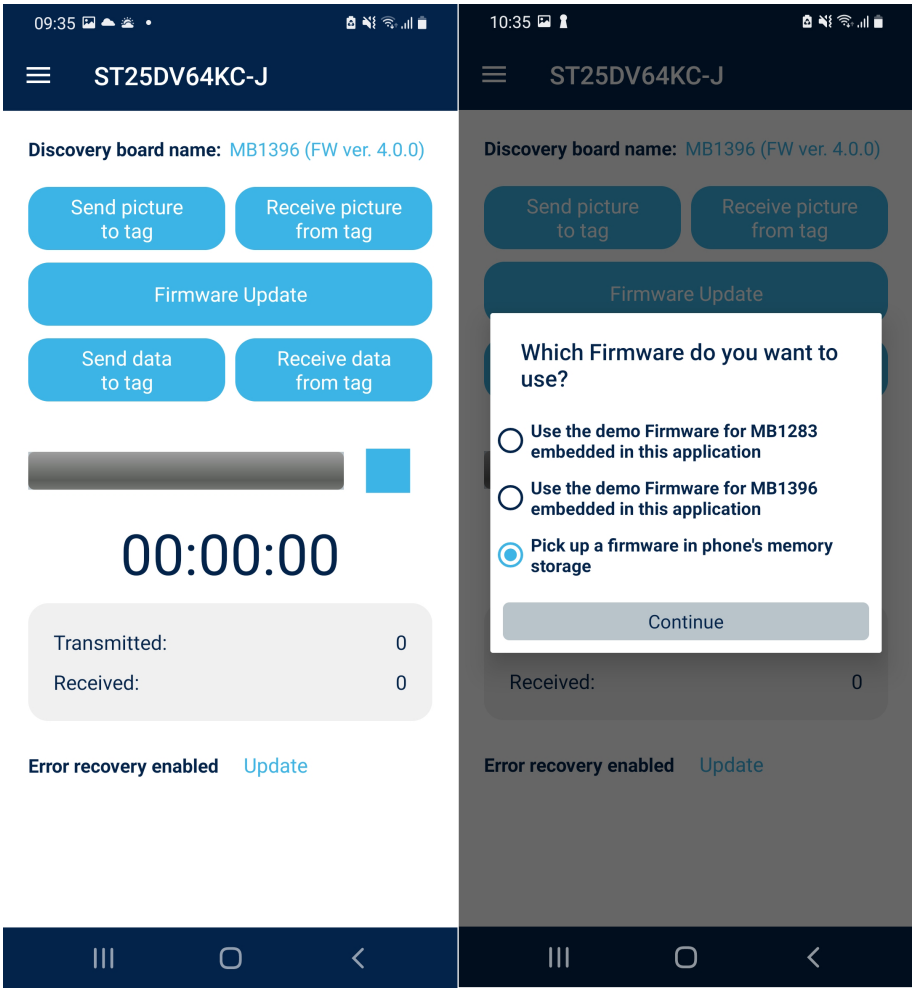
Select "FTM Demos" and click "continue".

Figure 26. **ST25 NFC tag app - FTM demos**



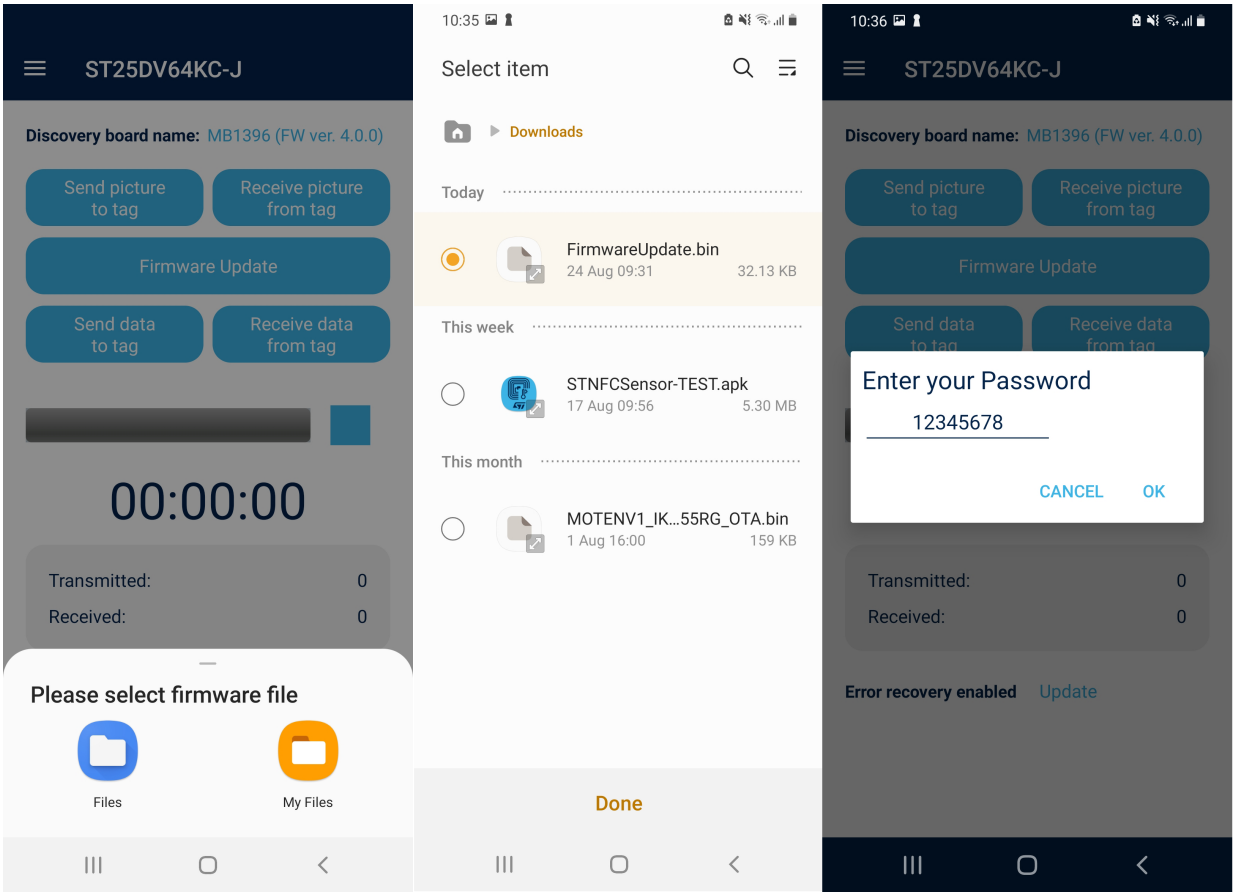
Select "Firmware Update", choose “Pick up a firmware in phone’s memory storage" and click "continue".

Figure 27. **ST25 NFC tag app - firmware update**



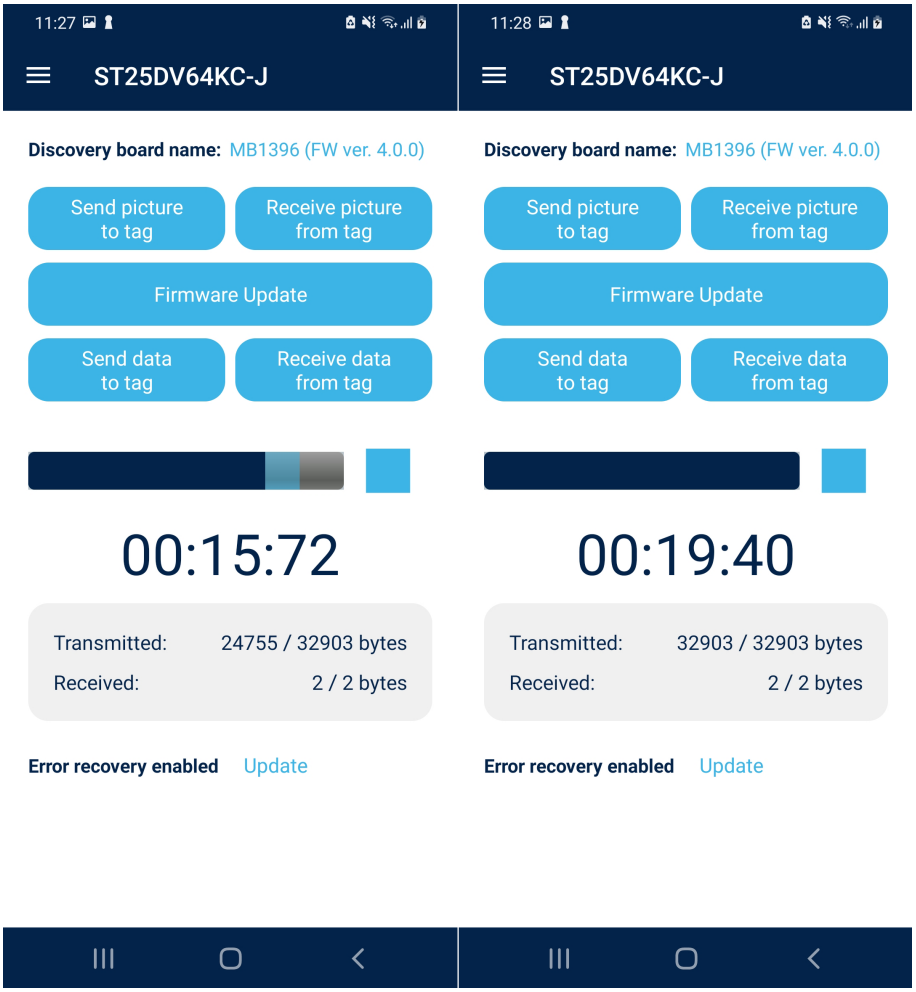
Select the firmware file and enter the password (12345678 is the default one).

Figure 28. **ST25 NFC tag app - password entering**



Wait for the firmware update to finish.

Figure 29. **ST25 NFC tag app - firmware update complete**



2.2.6 SDDataLogFileX

The purpose of this application is to show an example about using FileX and ThreadX for saving the output of the inertial and environmental sensors present on the board, and the output of the Digital microphone, to the SD-card.

After the boot sequence it is possible to press the User button for starting/stopping the SD data log.

At the end of each logging there will be 2 different files saved on SD-card:

- Sens000.csv: it stores the values of Acc/Gyro/Mag/Pressure/Temperature/Humidity at 100 Hz
- Mic000.wav: it stores the wave file for Analog Microphone at 48 KHz

3 System setup guide

3.1 Hardware description

3.1.1 STEVAL-STWINBX1 evaluation kit

The STWIN.box (STEVAL-STWINBX1) is a development kit and reference design that simplifies prototyping and testing of advanced industrial sensing applications in IoT contexts such as condition monitoring and predictive maintenance.

It is an evolution of the original STWIN kit (STEVAL-STWINKT1B) and features a higher mechanical accuracy in the measurement of vibrations, an improved robustness, an updated BoM to reflect the latest and best-in-class MCU and industrial sensors, and an easy-to-use interface for external add-ons.

The STWIN.box kit consists of an STWIN.box core system, a 480mAh LiPo battery, an adapter for the ST-LINK debugger, a plastic case, an adapter board for DIL 24 sensors and a flexible cable.

The many on-board industrial-grade sensors and the ultra-low power MCU enable applications that feature: ultra-low power, 9 DoF motion sensing, wide-bandwidth vibration analysis, audio and ultrasound acoustic inspection, very precise local temperature, and environmental monitoring.

A rich set of software packages is available in source code. Optimized firmware libraries and a complete companion cloud application help to speed up the design cycle to develop end-to-end solutions.

The kit supports a broad range of connectivity options, including the built-in RS485 transceiver, BLE, Wi-Fi, and NFC.

The STWIN.box also includes a 34-pin expansion connector for small form factor daughter boards associated with the STM32 family, such as the STEVAL-C34KAT1, STEVAL-C34KAT2 and STEVAL-PDETECT1 expansion boards.

The STWIN.box is suitable for field trials, demonstrations, and PoC for industrial IoT applications that use ST software and third-party software.

Figure 30. STEVAL-STWINBX1 evaluation kit

3.1.2 STEVAL-MKBOXPRO evaluation kit

The STEVAL-MKBOXPRO (SensorTile.box PRO) is the new ready-to-use programmable wireless box kit for developing any IoT application based on remote data gathering and evaluation, exploit the full kit potential by leveraging both motion and environmental data sensing, along with a digital microphone, and enhance the connectivity and smartness of whatever environment you find yourself into.

You can entirely enjoy the SensorTile.box PRO experience regardless of your level of expertise, the box kit could be exploited according to three different modalities:

Entry mode: run a wide range of already embedded IoT applications on your box.

You can download the free STBLESensor App on your smartphone and immediately begin commanding the board with any of the following applications that have been specifically designed to work with the board sensors:

1) Motion: Compass, Free-fall detection, Level, Pedometer, Sensor-fusion - Quaternion

2) Environmental: Barometer

3) Log: Data recorder

4) AI and MLC: Baby crying detector, Human activity recognition

5) User interface: Qtouch

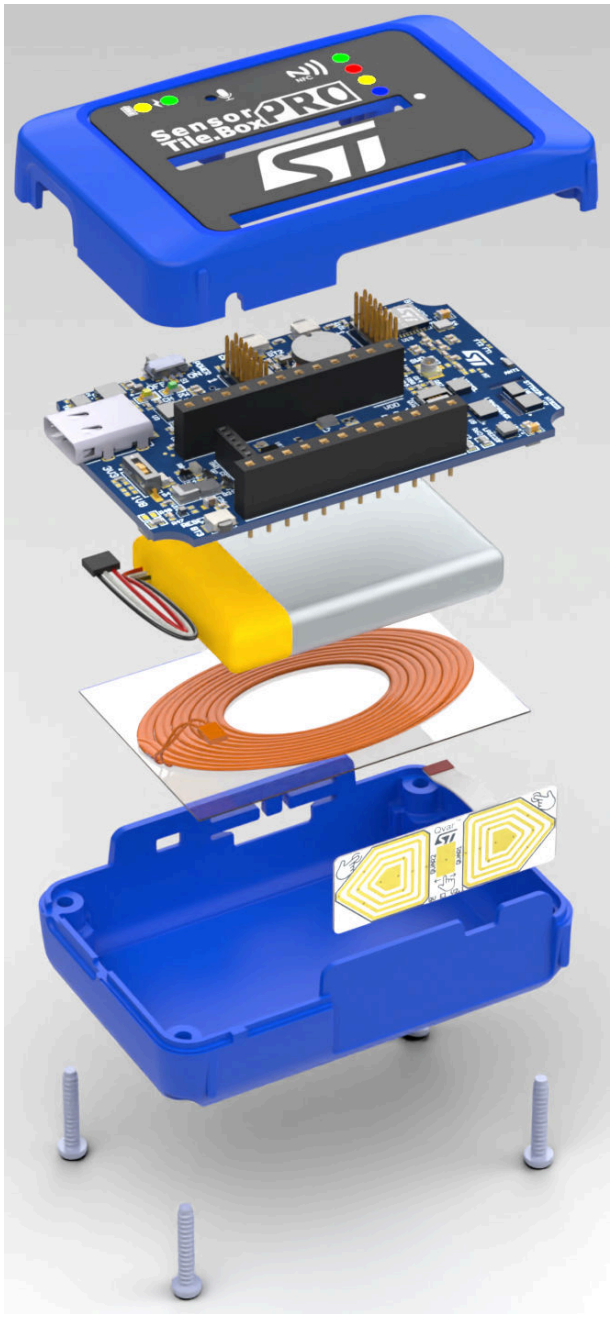
6) Connectivity: NFC Tag

Expert mode: build custom applications through the STBLESensor App by selecting specific input data and operating parameters from corresponding available in-box sensors, functions to assess/compute those data, and output types that you need, while leveraging on the available powerful algorithms.

Pro mode: develop quickly your own tailored IoT application taking advantage of STM32 open development environment (ODE) and ST function pack libraries, including sensing AI function pack with neural network libraries, without the need to perform any coding activity.

The SensorTile.box PRO board fits into a small plastic box with a long-life 480mAh rechargeable battery, for the first time leveraging also on a wireless charger and a programmable NFC tag. The board can be easily connected via Bluetooth to the ST BLE Sensor app on your smartphone, from which the box kit can be enjoyed in Entry and Expert mode. In Pro mode, professional users can exploit the firmware programming and debugging interface in the STM32 ODE for developing their firmware from scratch.

Figure 31. **STEVAL-MKBOXPRO evaluation kit**



3.1.3 Hardware setup

The following hardware components are needed for STEVAL-STWINBX1:

- One STWIN.box evaluation kit (order code: STEVAL-STWINBX1)
- One ST-LINK-V3SET (or ST-LINK-V3MINI) debugger/programmer
- One USB type A to Micro-B USB cable to connect the ST-LINK-V3SET (or ST-LINK-V3MINI) to the PC
- One type-C USB cable to connect the STEVAL-STWINBX1 to the PC

Figure 32. STEVAL-STWINBX1 setup

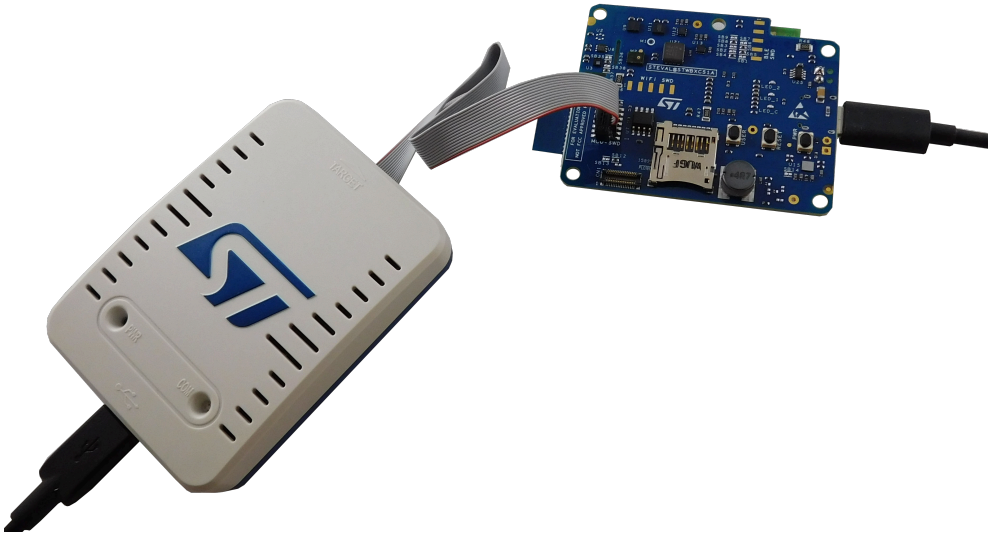
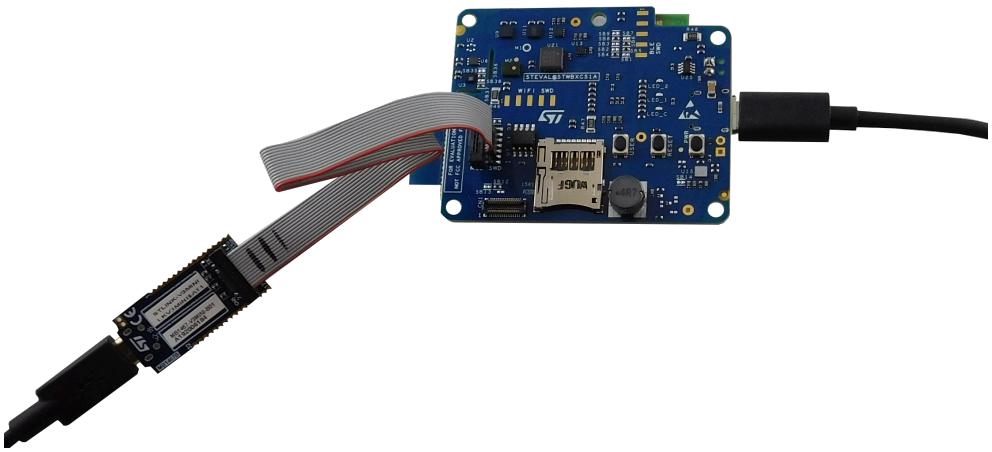


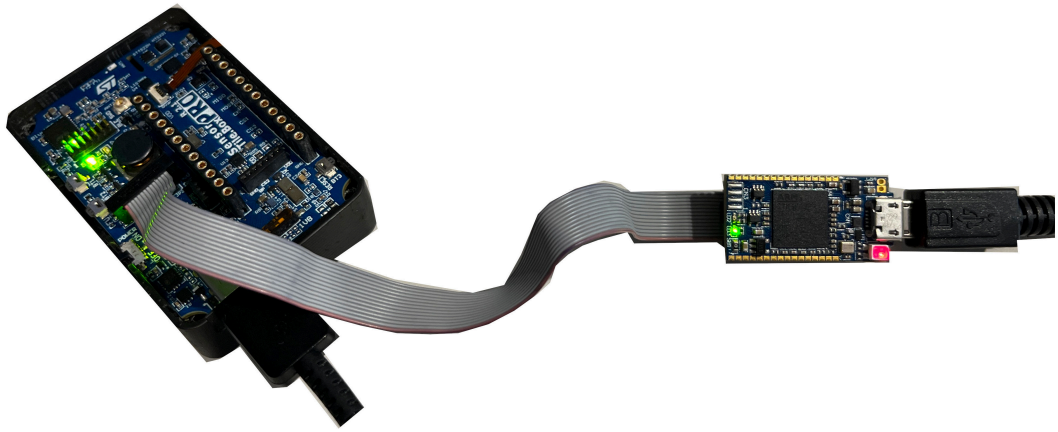
Figure 33. STWIN.box+STLinkV3_Mini setup



The following hardware components are needed for STEVAL-MKBOXPRO:

- One SensorTile.box-Pro evaluation kit (order code: STEVAL-MKBOXPRO)
- One ST-LINK-V3SET (or ST-LINK-V3MINI) debugger/programmer
- One USB type A to Micro-B USB cable to connect the ST-LINK-V3SET (or ST-LINK-V3MINI) to the PC
- One type-C USB cable to connect the STEVAL-MKBOXPRO to the PC

Figure 34. **STEVAL-MKBOXPRO setup**



3.1.4 Software setup

The following software components are required for the setup of a suitable development environment to create applications for the STEVAL-STWINBX1 and STEVAL-MKBOXPRO evaluation boards:

- STM32Cube function pack for the Pro Mode of the SensorTile.box wireless multi sensor development kitSTM32 Nucleo development board firmware
- a standard user terminal as Putty or Tera Term (v. 4.97 or higher)
- STBLESensor app (only for BLE application)
- Development tool-chain and Compiler. The STM32Cube expansion software supports the three following environments to select from:
  - IAR Embedded Workbench for ARM® toolchain + ST-LINK
  - RealView Microcontroller Development Kit toolchain + ST-LINK
  - STM32CubeIDE + ST-LINK

Revision history

Table 1. **Document revision history**

Date	Version	Changes
02-Sep-2019	1	Initial release.
23-Apr-2020	2	Updated *Section 1.1 Overview, Figure 1. FP-SNS-STBOX1 software architecture,* *Section 1.3 Folder structure, Section 2 Sample and application projects,* BootLoader and BLEFOTA applications, *Section 2.1.5 BLEMLC application* and *Section 3.1.4 Software setup*. Added *Section 2.1.1 BLEDualProgram application*.
27-Jan-2023	3	Modified the title and the introduction in cover page. Updated *Section 2 Sample and application projects*. Added *Section 2.2 STEVAL-STWINBX1 application projects*. Updated *Section 3.1.3 Hardware setup*
17-May-2023	4	Modified introduction in cover page. Updated *Section 1.1: Overview, Section 1.2: Architecture, Section 1.3: Folder structure,* *Section 2: Sample and application projects and Section 3.1.3: Hardware setup.* Added *Section 2.2: STEVAL-MKBOXPRO application projects and Section 3.1.2: STEVALMKBOXPRO* *evaluation kit.*
04-Jul-2024	5	Updated Section Introduction, Section 1.1: Overview, Figure 1. FP-SNS-STBOX1 software architecture, Section 1.3: Folder structure, Section 2: Sample and application projects, Section 2.1.1: BLEMLC application, Section 2.2.2: BLEMLC, Section 3.1.3: Hardware setup and Section 3.1.4: Software setup Removed Section 2.1 STEVAL-MKSBOX1V1 applicatin projects, Section 2.2.1 BLEDualProgram application, Section 2.2.2 BLEGPEx application, Section 2.2.4 BLESensors application for STEVAL-STWINBX1, Section 2.3.2 BLEGPEx, Section 2.3.4 BLESensors, Section 2.3.9 ExampleCubeMxDataLog and Section 3.1.1 STEVAL-MKSBOX1V1 evaluation kit.

Contents

List of figures