life.augmented

Quick Start Guide

STM32Cube function pack for IoT node with Dynamic NFC Tag, environmental, motion and ambient light sensors
(FP-SNS-SMARTAG2)

Version 1.1 (March 2, 2023)

Agenda

1 Hardware and Software overview

2 Setup & Demo Applications

3 Documents & Related Resources

4 STM32 Open Development Environment: Overview

1- Hardware and Software overview

NFC Dynamic Tag sensor node evaluation board (STEVAL-SMARTAG2)
Hardware Overview

STEVAL-SMARTAG2 Hardware Description

- STEVAL-SMARTAG2 is a flexible NFC Tracker evaluation board with sensors includes a comprehensive software library and a sample application to monitor and log sensor data over NFC from an Android or iOS device. The ultra-low power sensor node evaluation board mounts an ST25DV NFC Tag, an STM32L4 with FPU Arm Cortex-M4, environment sensors (temperature, pressure, ambient light) and motion high accuracy (accelerometer) sensor.

- The evaluation board features a battery cradle for a CR2032 battery or for rechargeable battery LIR2032 (by default standard coin cell CR2032 battery must be used).

Key Product on board

- ST25DV64K dynamic NFC tag solution based on 64K-bit (8K-Byte) EEPROM and with I²C interface, Fast Transfer Mode and Energy Harvesting features
- STM32L4P5CE Ultra-low-power with FPU Arm Cortex-M4 + MCU 120 MHz with 512 kbytes of Flash memory, USB OTG, DFSDM, CHROM-ART
- LSM6DSO32X iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope
- LPS22DF Low-power and high-precision MEMS nano pressure sensor: 260-1260 hPa absolute digital output barometer
- STTS22H Low-voltage, ultra-low-power, 0.5 °C accuracy I2C/SMBus 3.0 temperature sensor
- VD6283 Hybrid filter multispectral sensor with light flicker engine (Ambient Light Sensor)
- H3LIS331DL Low power High-g 3-axis accelerometer, SPI/I2C digital output MEMS motion sensor, user-selectable full scales of ±100g/±200g/±400g
- STLQ020 200 mA ultra-low quiescent current LDO
- STBC15 Ultra-low current consumption linear battery charger
- STSAFE-A110 Authentication, state-of-the-art security for peripherals and IoT devices
- CR2032 Battery powered (not included)
- LIR2032 rechargeable battery (not included)
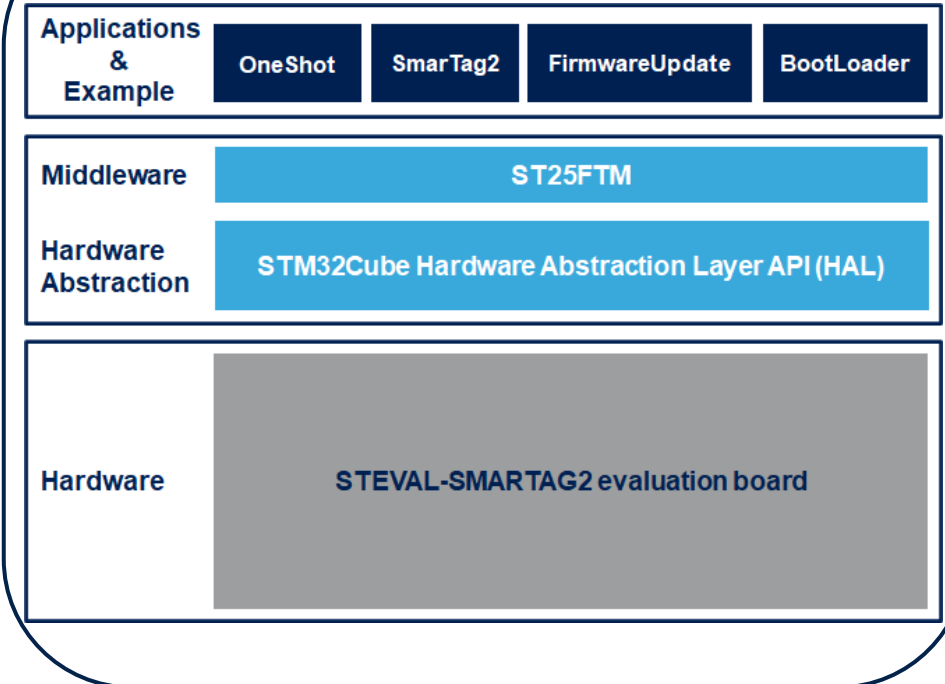
Latest info available at www.st.com
STEVAL-SMARTAG2

Software Description

- FP-SNS-SMARTAG2 is an STM32Cube function pack which allows you to read the ambient light, motion and environmental sensor data on your IoT node via an NFC enabled reader such as a mobile phone or a tablet.

- The package supports energy harvesting (enabled by NFC and available only for version B of the evaluation board STEVAL$SMARTAG2B) and battery-operated use cases.

- This software, together with the suggested combination of STM32 and ST devices can be used, for example, to develop tracking, cold chain, medical, smart sensing, and smart home, city and building applications.

- The software runs on an ultra-low power STM32L4 microcontroller and includes drivers for the Dynamic NFC tag and for the ambient light, motion and environmental sensors.

- You can register the NFC Sensor Tag node on the DSH-ASSETRACKING web application for asset tracking that store and monitors on-board sensor data as well as the geolocalization of the smartphone used to read the IoT node data.

Key features

- Complete firmware to access data from an IoT node with a dynamic NFC tag, environmental, motion, and ambient light sensors
  - Ultra-low power operations, with the support of energy harvesting (only for version B of the evaluation board STEVAL$SMARTAG2B) and battery-operated use cases
  - Compatible with the STAssetTracking application for Android/iOS. This allows data logs reading from the NFC tag and data logs sending to the DSH-ASSETRACKING cloud-based dashboard
  - Compatible with the STNFCSensor application for Android/iOS for reading and setting the data logs
- The package contains also one example that shows out to make firmware update using the fast transfer mode protocol (ST25FTM)
  - Compatible with the ST25 NFC tag application for download the firmware on board via NFC
- Sample implementation available for the STEVAL-SMARTAG2 evaluation board
- Easy portability across different MCU families, thanks to STM32Cube

Overall Software Architecture

Latest info available at www.st.com FP-SNS-SMARTAG2

2- Setup and demo applications

- **STSW-LINK009**
  - STLINK-V3SET (or STLINK-V3MINI) USB driver

- **STSW-LINK007**
  - STLINK-V3SET (or STLINK-V3MINI) firmware upgrade

- **FP-SNS-SMARTAG2**
  - Copy the .zip file content into a folder on your PC. The package will contain source code example (Keil, IAR, STM32CubeIDE) based only on **STEVAL-SMARTAG2**

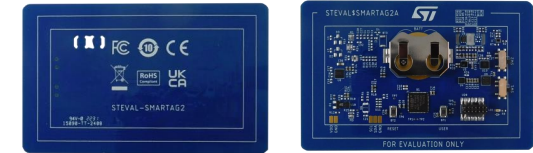- **ST Asset Tracking** and **ST NFC Sensor** applications for **Android/iOS** available from Google Store / iTunes

2.1- Setup Overview: STEVAL-SMARTAG2 evaluation boards

HW prerequisites and setup with ST NFC SensorTag

- 1x ST NFC SensorTag (**STEVAL-SMARTAG2**)

- 1x Android™ or iOS™ device with **ST Asset Tracking** and **ST NFC Sensor** apps installed

- 1x PC with Windows 10 and above

- 1x **ST-LINK-V3SET** (or **ST-LINK-V3MINI**) debugger/programmer

- 1x **USB type A to Micro-B USB cable** to connect the ST-LINK-V3SET (or ST-LINK-V3MINI) to the PC
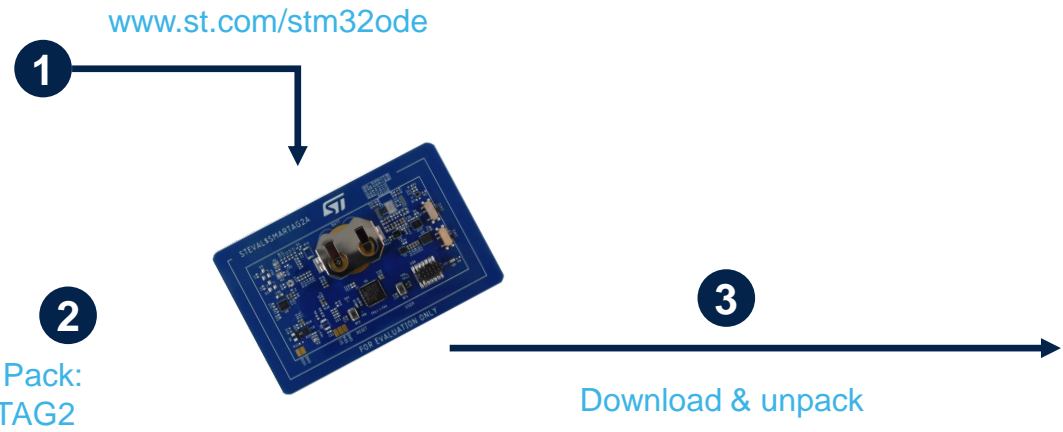
STEVAL-SMARTAG2
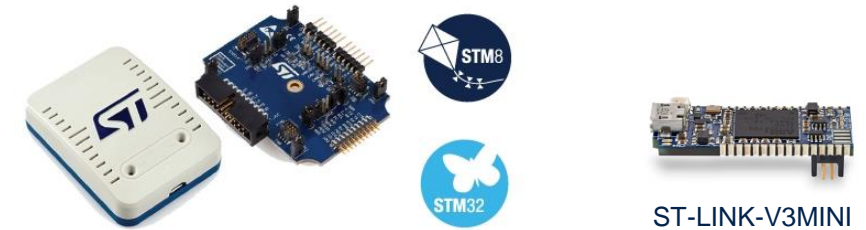
ST-LINK-V3MINI

ST-LINK-V3SET
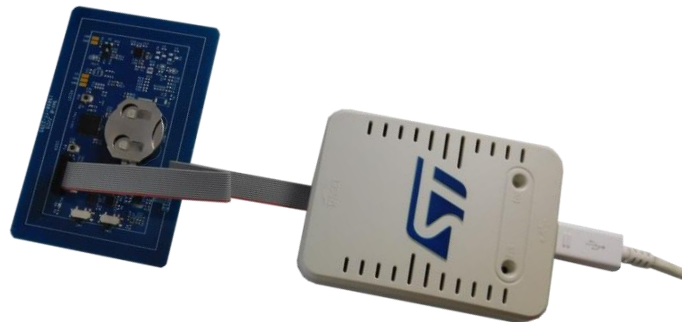
Micro USB

Start coding in just a few minutes (1/3)

1 www.st.com/stm32ode

2 Select Function Pack:
FP-SNS-SMARTAG2

3 Download & unpack

FP-SNS-SMARTAG2 package structure

Name

- _htmresc
- Documentation → Docs
- Drivers → BSP, HAL and drivers
- Middlewares → ST25FTM
- Projects → Examples/Applications for STEVAL-SMARTAG2
- DM00751078.pdf
- package.xml
- Package_License.html
- Package_License.md
- Release_Notes.html

4

. \Projects\STM32L4P5CE-SmarTag2\Examples\OneShot
. \Projects\STM32L4P5CE-SmarTag2\Examples\SmarTag2
. \Projects\STM32L4P5CE-SmarTag2\Examples\SimpleBootLoader
. \Projects\STM32L4P5CE-SmarTag2\Applications\FirmwareUpdateBL

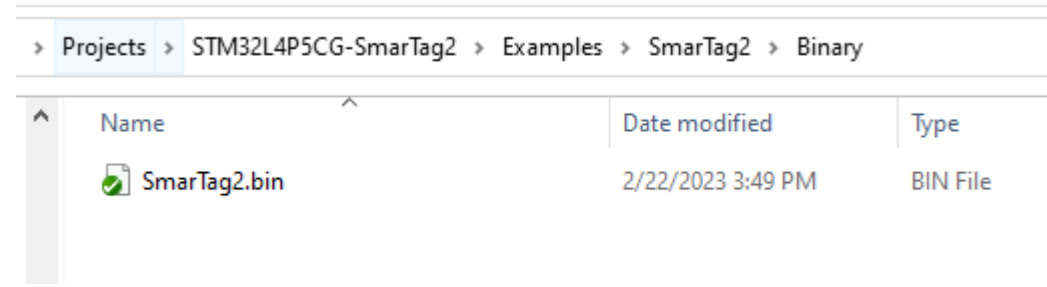Android™/iOS™ smartphone with ST Asset Tracking and ST NFC Sensor applications

6

Use the pre-compiled binaries for registering your device, or alternative re-compile the code adding your device certificate

5

STM32 CubeIDE   KEIL Tools by ARM   IAR SYSTEMS

1. How to install the pre-compiled binary:

➢ For SmarTag2, there is inside the package a folder called "Binary" with precompiled firmware that can be directly flashed with STM32CubeProgrammer to the based memory address (0x08000000)
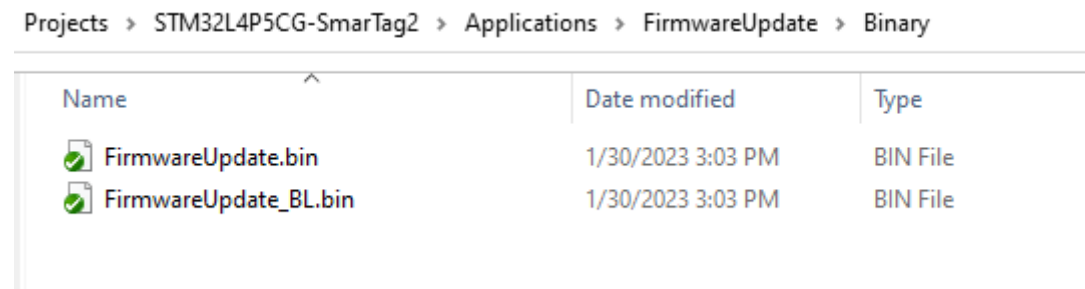
> Projects > STM32L4P5CG-SmarTag2 > Examples > OneShot > Binary

Name	Date modified	Type
OneShot.bin	2/22/2023 3:44 PM	BIN File

> Projects > STM32L4P5CG-SmarTag2 > Examples > SmarTag2 > Binary

Name	Date modified	Type
SmarTag2.bin	2/22/2023 3:49 PM	BIN File

➢ For FirmwareUpdateBL, there is inside the package a folder called "Binary" that contains:

- precompiled firmware that may be flashed with STM32CubeProgrammer to the correct memory address (0x08002000)
  - Note: This precompiled binary is compatible with the firmware updated update procedure

- precompiled firmware plus bootloader that may be directly flashed with the STM32CubeProgrammer
  - Note: This precompiled binary is not compatible with the firmware updated update procedure

Projects > STM32L4P5CG-SmarTag2 > Applications > FirmwareUpdate > Binary

Name	Date modified	Type
FirmwareUpdate.bin	1/30/2023 3:03 PM	BIN File
FirmwareUpdate_BL.bin	1/30/2023 3:03 PM	BIN File

2. How Install the code after compiling the project:

- Compile the project with your preferred IDE and download on board using the IDE
  - For only FirmwareUpdateBL, download the bootloader firmware included in this package before

- For only FirmwareUpdateBL, for each IDE a batch script for STM32CubeProgrammer has been provided to simplify the operation described before by saving the firmware and the bootloader to the right position:

  - On Windows for IAR toolchain Embedded Workbench V9.20.1:
    - CleanSMARTAG2_IAR_FirmwareUpdateBL.bat
  - On Windows for Keil µVision 5 toolchain - MDK-ARM Professional Version: 5.37.0:
    - CleanSMARTAG2_MDK-ARM_FirmwareUpdateBL.bat
  - On Windows for STM32CubeIDE Version 1.11.0:
    - CleanSMARTAG2_STM32CubeIDE_FirmwareUpdateBL.bat

  - For Linux/iOS for only for STM32CubeIDE Version 1.11.0:
    - CleanSMARTAG2_STM32CubeIDE_FirmwareUpdateBL.sh
  - NOTE: It's necessary to edit this file for setting the right installation of the STM32CubeProgrammer

These scripts perform the following steps:

1. Performs a full flash memory erase to start from a clean system

2. Flashes the bootloader to the correct position (0x08000000)

3. Flashes the firmware to the correct position (0x08002000)

Projects › STM32L4P5CG-SmarTag2 › Applications › FirmwareUpdateBL › EWARM

| Name	Date modified	Type	Size
CleanSMARTAG2_IAR_FirmwareUpdateBL.bat	8/31/2022 11:16 AM	Windows Batch File	2 KB
FirmwareUpdate.ewd	8/26/2022 8:44 AM	EWD File	54 KB
FirmwareUpdate.ewp	8/31/2022 11:15 AM	EWP File	45 KB
Project.eww	8/26/2022 8:44 AM	IAR IDE Workspace	1 KB
startup_stm32l4p5xx.s	8/26/2022 8:44 AM	S File	23 KB
stm32l4p5xx_flash.icf	8/26/2022 8:44 AM	ICF File	3 KB
stm32l4p5xx_sram.icf	8/26/2022 8:44 AM	ICF File	3 KB
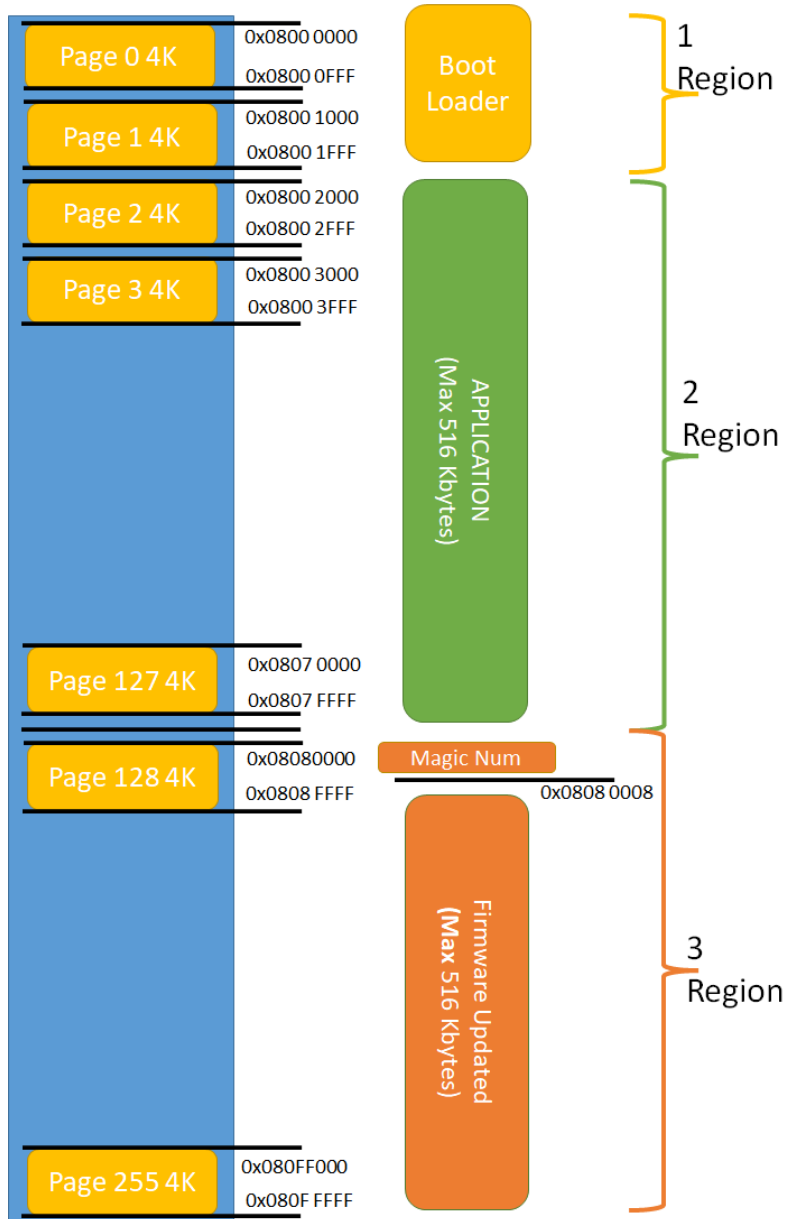
Flash Structure for STM32L4P5CE
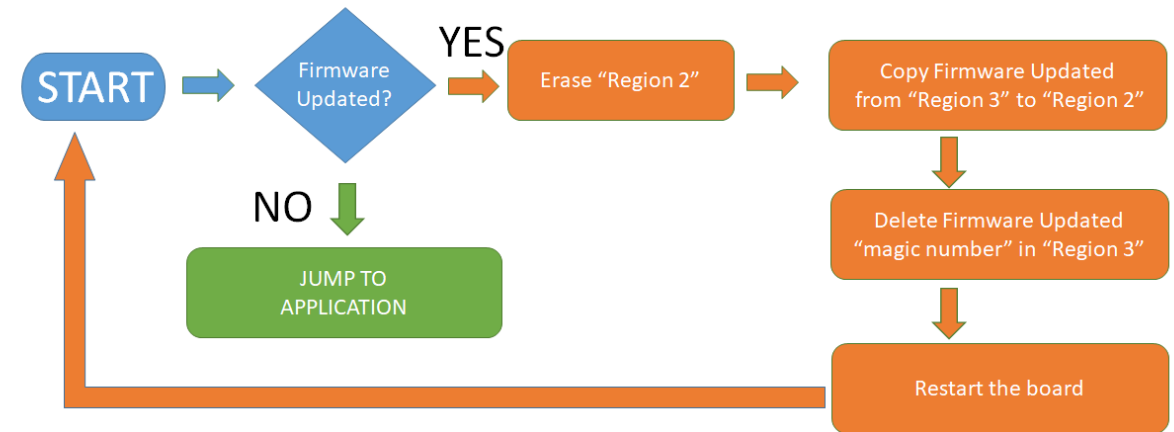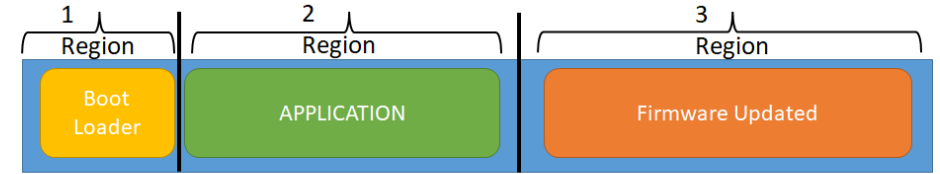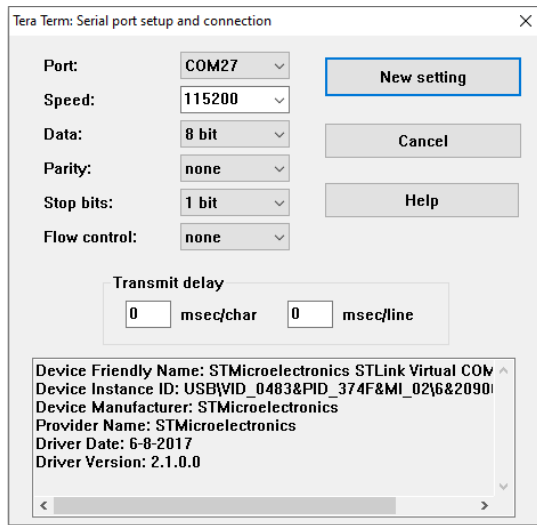
Configure the serial line monitor (speed, LF)

After the **RESET** you could see the initialization phase

The environmental sensor data are logged

```
STMicroelectronics One Shot:
        Version 1.1.0
        STEVAL_SMARTAG2 board
NFCTAG Initialized
Sleep RF
Crete NDEF record (V2.1)
SaveDefaultConfiguration
SaveVirtualSensorsConfiguration
LPS22DF: Press= 999.635498
STTS22H: Temp= 20.370001
VD6283: KLux= 94.294075
Write One Shot Done
WakeUp RF
End of Program
```
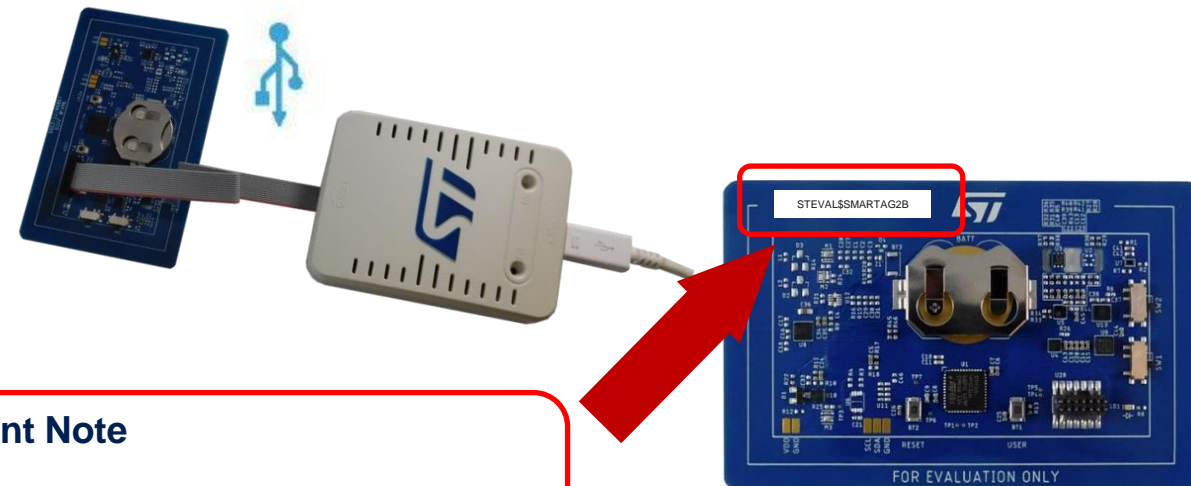
NOTE

To enable/disable this UART functionality on the STEVAL-SMARTAG2 board, you must recompile the code by uncommenting/commenting the line:

#define SMARTAG2_ENABLE_PRINTF

in the file:
Projects\ STM32L4P5CE-SmarTag2\Examples\ OneShot\Inc\SMARTAG2_config.h

STEVAL$SMARTAG2B

(*) Known Limitation / Important Note

This example/mode can be used only with the version B of the evaluation board
Please check that the label on your board is: STEVAL$SMARTAG2B

14

```
UART Initialized

STMicroelectronics FP-SNS-SMARTAG2:
        Version 1.1.0
        STEVAL_SMARTAG2 board
        (HAL 1.13.2_0)
        Compiled Jan 18 2023 11:07:29 (IAR)

Power on NFC (VDD EEP On)

NFCTAG Initialized
SmarTagUID= e0025300415acb1
Creating NDEF External record Type for saving
        ST NFC Protocol Ver 2 Rev 1

Control if there is a Valid Configuration

        Vn=3 SampleTime=60
        Found STTS22H_VS_ID:
        ThsUsageType=Int
        Th1.Ui16Value=22.000000
        Th2.Ui16Value=43.200001
        Found LPS22DF_VS_ID:
        ThsUsageType=Bigger
        Th1.Ui16Value=960.000000
        Found VD6283_LUX_VS_ID:
        ThsUsageType=Bigger
        Th1.Ui32Value=100.000000
        Configuration Present on NFC

Set RTC Date&Time
ResetMaxMinValuesAllVirtualSensors
SaveMaxMinValuesForVirtualSensors

NfcType5_SetInitialNDEFPayLoadLengthValue:
        NDEFPayLoadLength=52
        BeginAddrCompactData=96
        EndAddrCompactData=8188
        MaxSamplesNumber=1011

Set NFC Behavior
Set WakeUp timer

Wait 2 sec before autoStart
```

After the **RESET** you could see the initialization phase

```
Set NFC Behavior
Set WakeUp timer

Wait 2 sec before autoStart
AutoStart

VDD ACC Off

Sync Event:
        Powered on ambient sensors (VDD AMB On)
        Read Sensor Data
                LPS22DF:        Press= 1002.824219
        Save LPS22DF
        Save Min Value for LPS22DF
        Save Max Value for LPS22DF
                STTS22H:        Temp= 25.410000
        Save STTS22H
        Save Min Value for STTS22H
        Save Max Value for STTS22H
                VD6283:         KLux= 52.529569
        Save Max Value for VD6283

Powered off ambient sensors (VDD AMB Off)

Sync Event:
        Powered on ambient sensors (VDD AMB On)
        Read Sensor Data
                LPS22DF:        Press= 1002.845947
        Save LPS22DF
                STTS22H:        Temp= 25.330000
        Save STTS22H
        Save Min Value for STTS22H
                VD6283:         KLux= 262.553155
        Save VD6283 KLux
        Save Max Value for VD6283

Powered off ambient sensors (VDD AMB Off)

Sync Event:
        Powered on ambient sensors
        Read Sensor Data
                LPS22DF:        Pre
        Save LPS22DF
                STTS22H:        Tem
        Save STTS22H
                VD6283:         KLu
        Save VD6283 KLux

Powered off ambient sensors (VDD AM

Sync Event:
        Powered on ambient sensors
        Read Sensor Data
                LPS22DF:        Pre
        Save LPS22DF
                STTS22H:        Tem
        Save STTS22H
                VD6283:         KLu
        Save VD6283 KLux

Powered off ambient sensors (VDD AM
```

After Autostart the sensor data are logged

```
Detected NFC FIELD On

Detected NFC FIELD Off
        Check if there is a new Configuration
        Vn=2 SampleTime=60
        Found STTS22H_VS_ID:
        Found LSM6DSOX32_6D_VS_ID:
Valid Configuration present on NFC
Restart the Log
        Vn=2 SampleTime=60
        Found STTS22H_VS_ID:
        ThsUsageType=Int
        Th1.Ui16Value=22.000000
        Th2.Ui16Value=43.200001
        Found LSM6DSOX32_6D_VS_ID:
        ThsUsageType=Ext
        Th1.Ui8Value=0
        Th2.Ui8Value=0
Set RTC Date&Time
ResetMaxMinValuesAllVirtualSensors
SaveMaxMinValuesForVirtualSensors

NfcType5_SetInitialNDEFPayLoadLengthValue:
        NDEFPayLoadLength=32
        BeginAddrCompactData=76
        EndAddrCompactData=8188
        MaxSamplesNumber=1014

VDD ACC On
Init Accelerometer Events:
Init LSM6DSOX32
WakeUp Off
6D On

6D Orientation=4
6D Orientation=2
Async Event:
Save LSM6DSOX32 6D

Sync Event:
        Powered on ambient sensors (VDD AMB On)
        Read Sensor Data
                STTS22H:        Temp= 27.290001
        Save STTS22H
        Save Min Value for STTS22H
        Save Max Value for STTS22H

Powered off ambient sensors (VDD AMB Off)
```

These messages are written when getting (before) and removing (after) the smartphone from the NFC tag. In particular, the new config is written if a new one it is set. After the sensor data are logged


Configure the serial line monitor (speed, LF)

NOTE

To enable/disable this UART functionality on the STEVAL-SMARTAG2 board, you must recompile the code by uncommenting/commenting the line:

#define SMARTAG2_ENABLE_PRINTF

in the file: Projects\ STM32L4P5CE-SmarTag2\Examples\ SmarTag2\Inc\SMARTAG2_config.h

15

After the **RESET** you could see the initialization phase

After the **USER button** is pressed FTM starts

After firmware update the board restarts

Configure the serial line monitor (speed, LF)

NOTE

To enable/disable this UART functionality on the STEVAL-SMARTAG2 board, you must recompile the code by uncommenting/commenting the line:

#define SMARTAG2_ENABLE_PRINTF

in the file:
Projects\ STM32L4P5CE-SmarTag2\Applications\ FirmwareUpdateBL\Inc\Firmware_conf.h
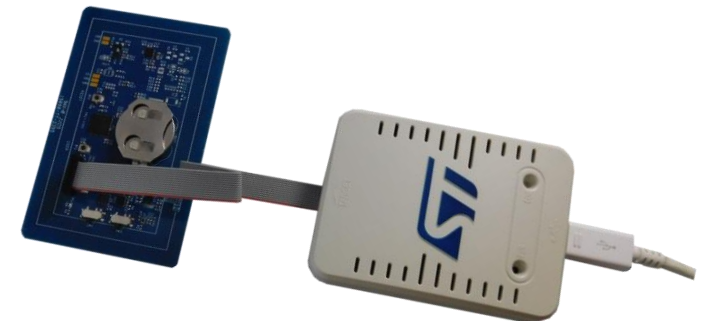
2.3- Demo Applications: ST Asset Tracking Application Overview

ST Asset Tracking Application for Android/iOS (1/4)

Sign in with MyST

If you are not sign-in or you are not registered on MyST web:

1. open the ST Asset Tracking application and click on "Sign-in with MyST"
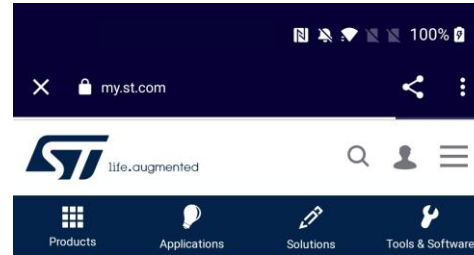
2. insert your E-mail address and password to execute the login if you are registered on MyST web before, otherwise click "New User" and follow the instruction.

ST Asset Tracking Application for Android/iOS (2/4)

Register new device
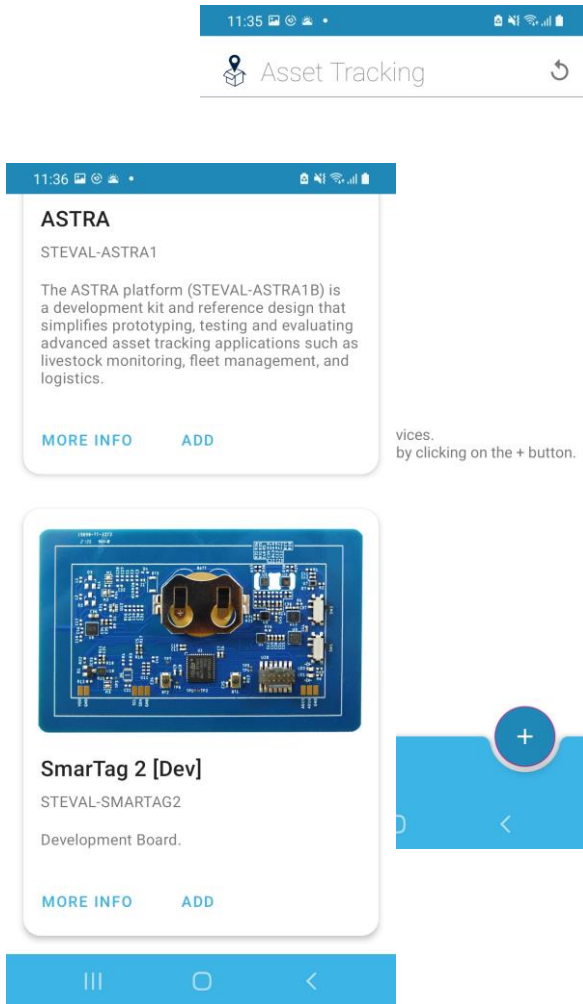
Click on the plus (+) button to register a new device
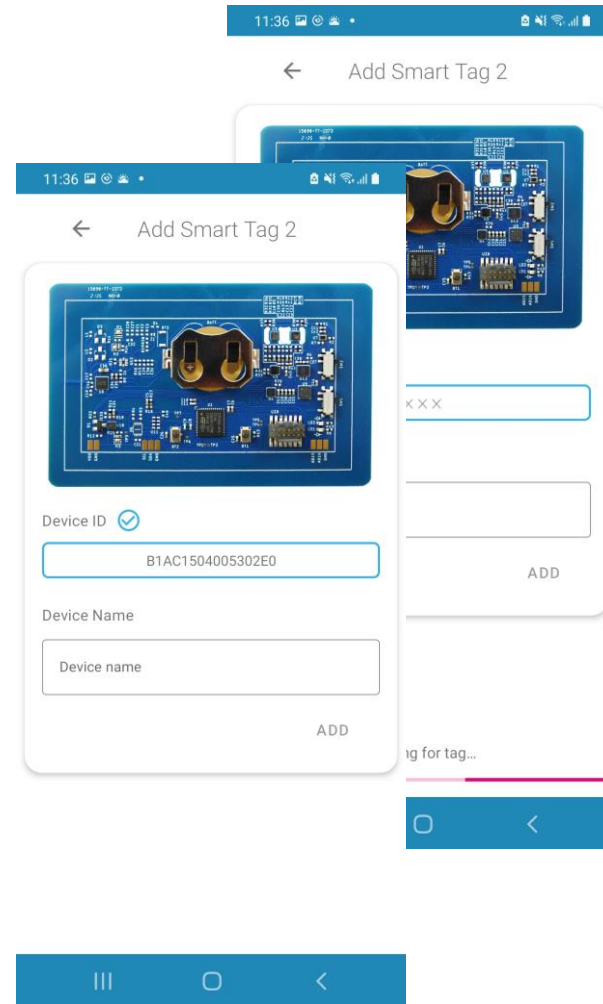
Click on the ADD button and put on the mobile phone on top the device to detect the tag by NFC
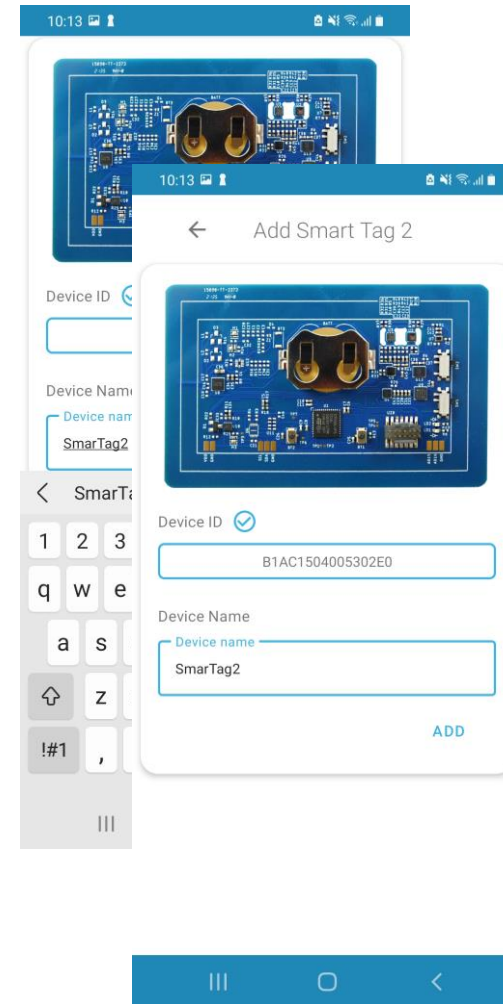
Insert the device name and click on the ADD button to complete the process
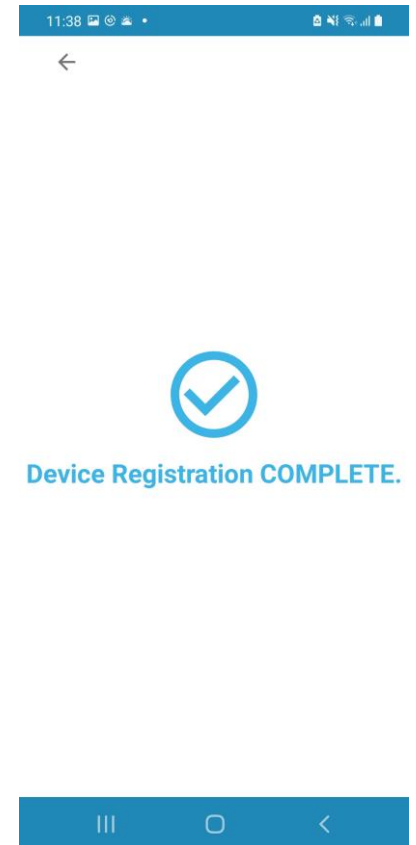
Demo Applications
ST Asset Tracking Application for Android/iOS (3/4)
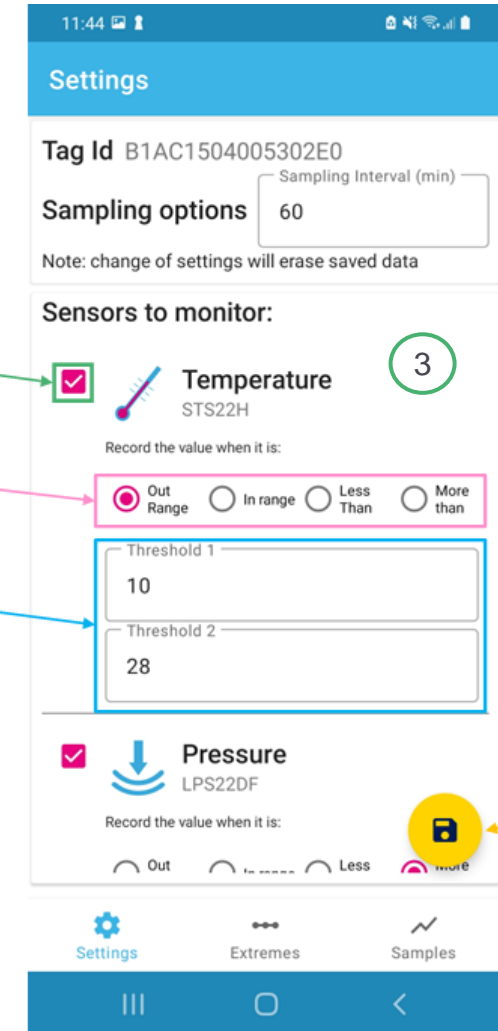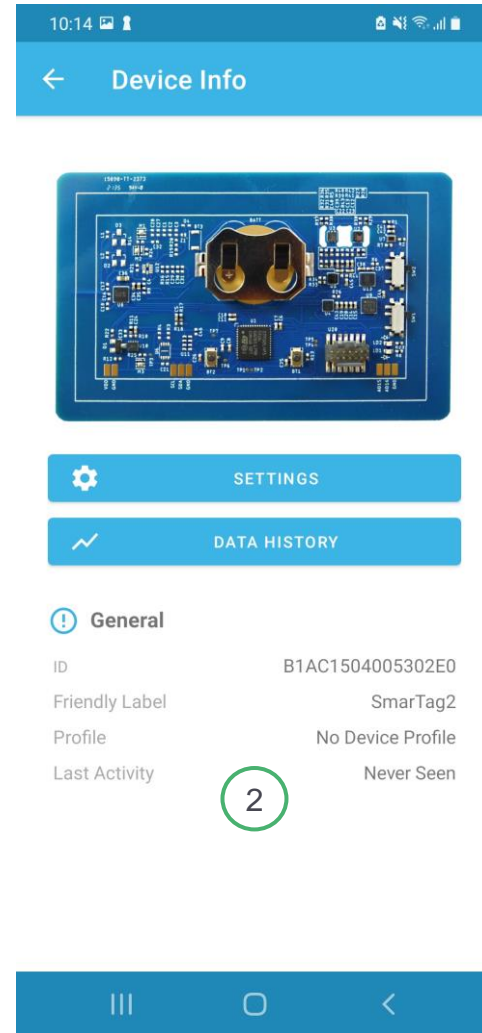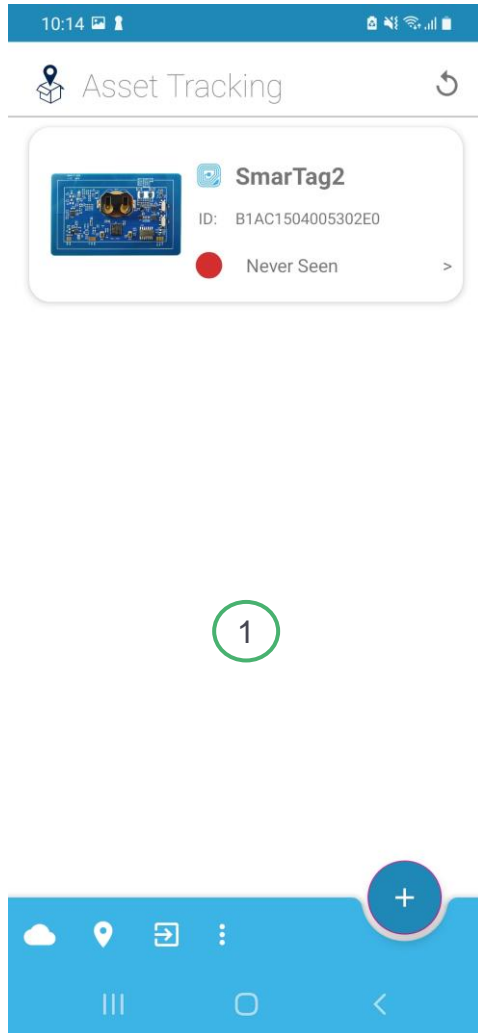
Settings

1. Start the ST Asset Tracking application and select your device from the list
2. Select settings and put on the mobile phone on top the device to detect the tag by NFC
3. Select the data to be logged with related data range

ST Asset Tracking Application for Android/iOS (4/4)

Extremes - Samples

Extremes: shows the maximum and minimum value obtained during the data logging of the selected data.
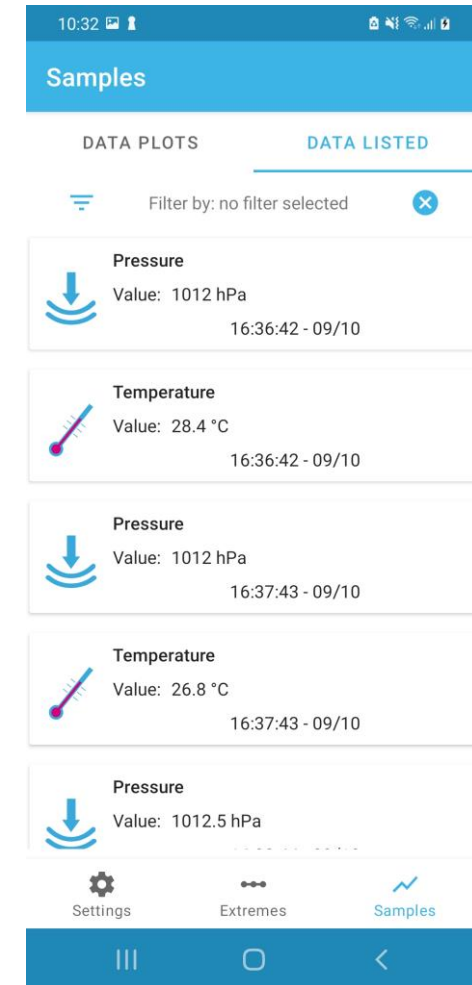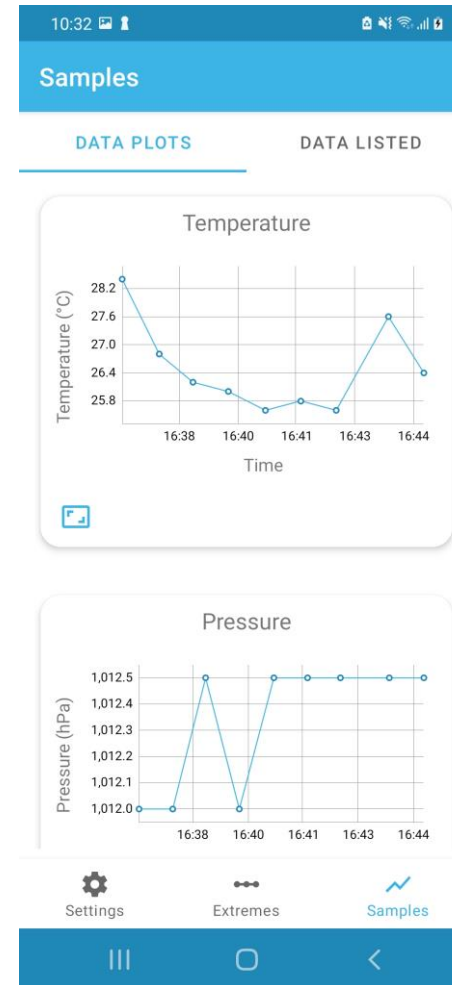
Samples: collects all data logging for the selected data with related data range

---

NOTE
For protocol used to log the data on to the NFC EEPROM read related document

21

2.4- Demo Applications: Firmware Update with Boot Loader by ST25 mobile application

Demo Applications
Firmware Update with Boot Loader by ST25 mobile application (1/2)
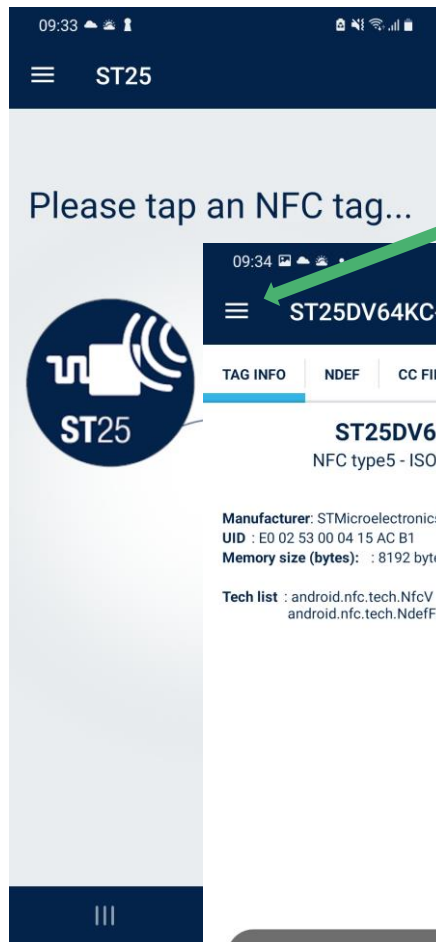
Start ST25 mobile application and put on your mobile phone onto the board.
Select menu option:
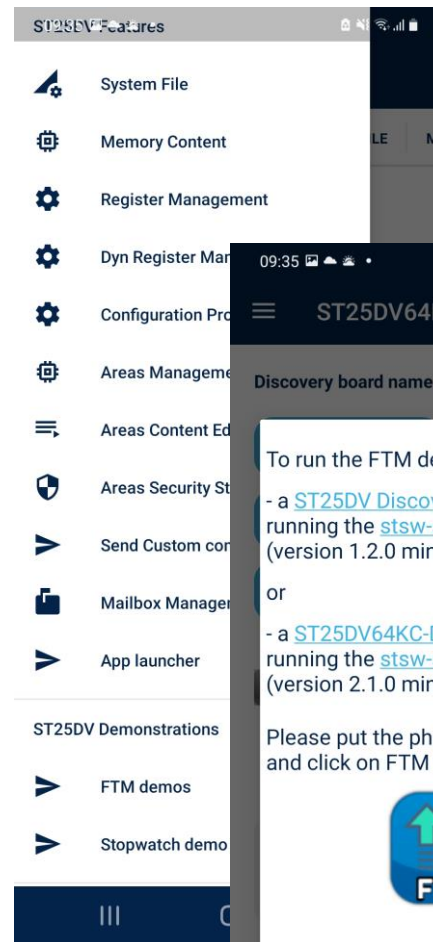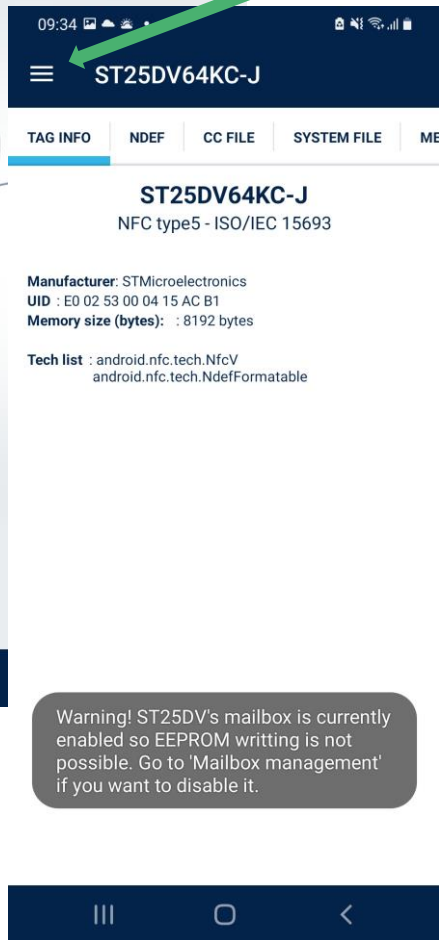
Select "FTM Demos" and click "continue"
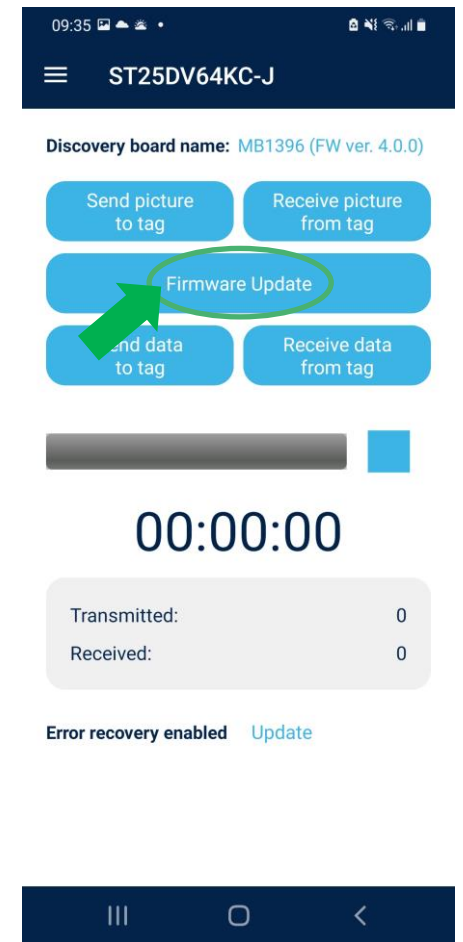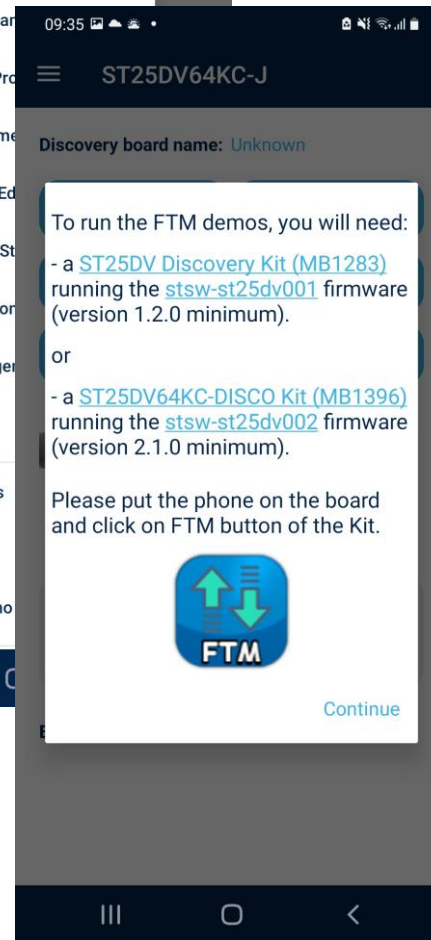
Select Firmware Update
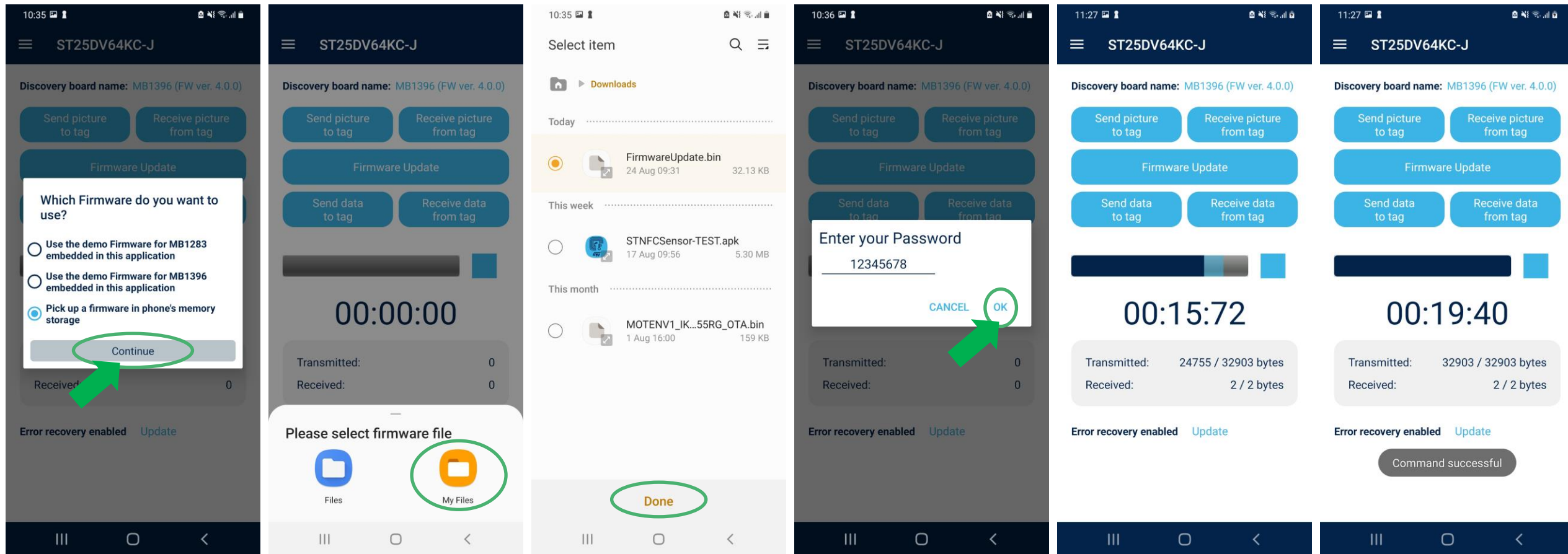
Firmware Update Done

2.5- Demo Applications: Using the Asset Tracking Web Dashboard

- Visit the home page of the DSH-ASSETRACKING dashboard on ST site for information and web dashboard URL (Link), or go Go to DSH-ASSETRACKING dashboard URL at https://dsh-assetracking.st.com/#/home

DSH-ASSETRACKING

Using the Asset Tracking Web Dashboard (2/6)

- Provide your username and password:
  - Select login and click GO button

- Adding new the device:

- Select the device you want to monitor :

- Selecting telemetry:
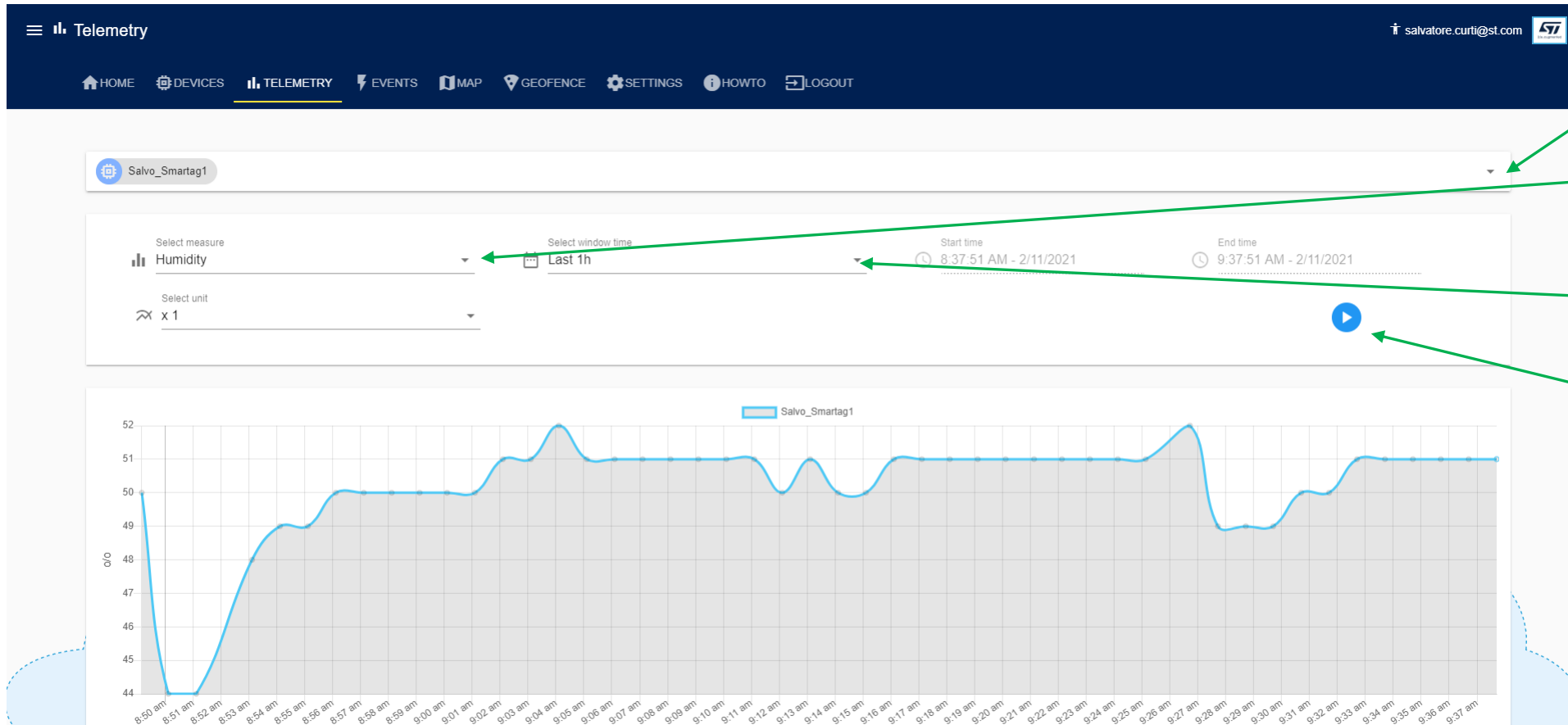
- Select the device you want to monitor.
- Select desired telemetry data to display
- Choose the amount of data to display
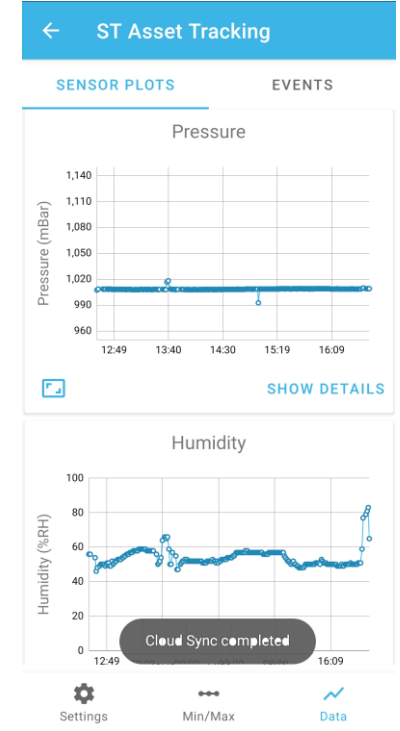- Select Play Button

Humidity data displayed

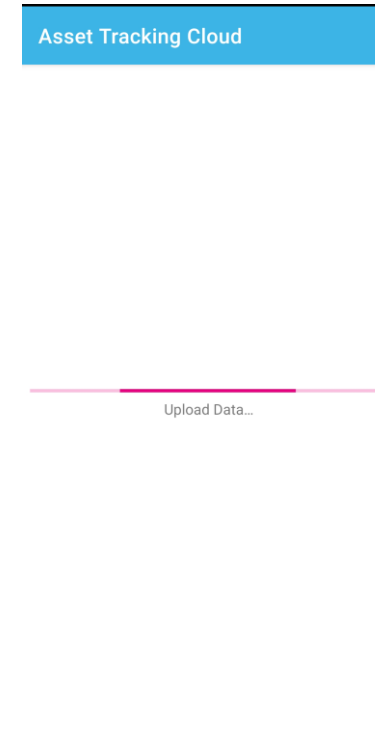## Using the Asset Tracking Web Dashboard (6/6)

- ST Asset Tracking Sign in with MyST:

31

3- Documents & Related Resources

Documents and related resources

All documents are available in the RESOURCES tab of the related products webpage

FP-SNS-SMARTAG2

- **DB4817:** STM32Cube function pack for IoT node with Dynamic NFC Tag, environmental, motion and ambient light sensors – **data brief**
- **UM3073:** Getting started with the STM32Cube function pack for IoT node with Dynamic NFC Tag, environmental, motion and ambient light sensors – **user manual**
- **Software setup file**

STEVAL-SMARTAG2

- **Gerber files, BOM, Schematic**
- **DB4770:** NFC dynamic tag sensor and processing node evaluation board – **data brief**
- **UM3034:** Getting started with the STEVAL-SMARTAG2 NFC dynamic tag sensor and processing node evaluation board – **user manual**

DSH-ASSETRACKING

- **DB4207:** Cloud Amazon-based web application for asset tracking – **data brief**

STNFCSensor

- **DB3666:** NFC Sensor TAG mobile application – **data brief**

STAssetTracking

- **DB3951:** ST Asset Tracking app to configure a Sigfox node based on the FP-ATR-SIGFOX1 function pack 3.0 – **data brief**
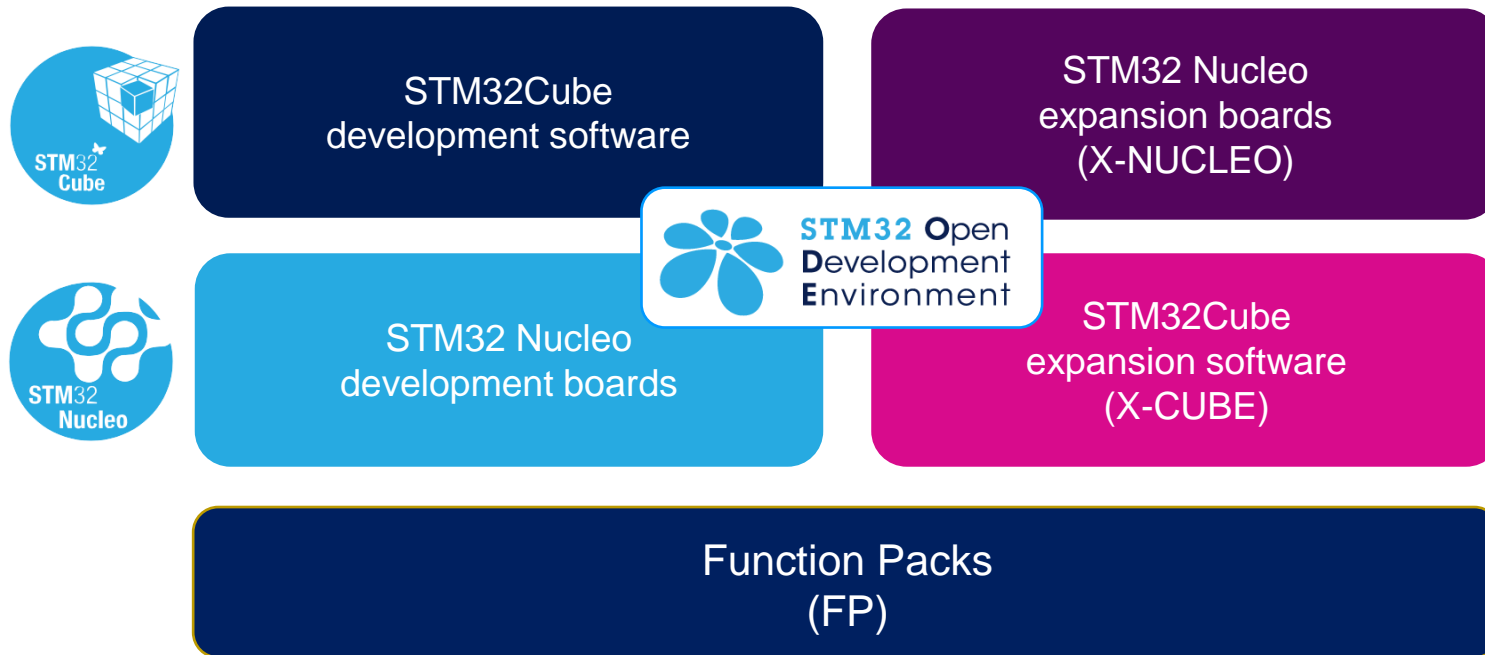
Consult www.st.com for the complete list

4- STM32 Open Development Environment: Overview

STM32 Open Development Environment
Fast, affordable Prototyping and Development

- The STM32 Open Development Environment (STM32 ODE) is an open, flexible, easy, and affordable way to develop innovative devices and applications based on the STM32 32-bit microcontroller family combined with other state-of-the-art ST components connected via expansion boards. It enables fast prototyping with leading-edge components that can quickly be transformed into final designs

STM32Cube
development software

STM32 Nucleo
expansion boards
(X-NUCLEO)

STM32 Open
Development
Environment

STM32 Nucleo
development boards

STM32Cube
expansion software
(X-CUBE)

Function Packs
(FP)

For further information, please visit www.st.com/stm32ode

Thank you

life.augmented