



# Session III – parte B

## Sensori e relativi sistemi di sviluppo

**Adriano Basile**  
**Francesco Sudano**  
**Maria Celvisia Virzì**

System LAB  
STMicroelectronics

## Libreria firmware per i sensori MEMS

[http://www.st.com/content/st\\_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/x-cube-mems1.html](http://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/x-cube-mems1.html)

## Libreria per Bluetooth

[http://www.st.com/content/st\\_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/x-cube-ble1.html](http://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/x-cube-ble1.html)

## Function Pack STM32 ODE per Internet of Thing

[http://www.st.com/content/st\\_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32-ode-function-pack-sw/fp-sns-motenv1.html](http://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32-ode-function-pack-sw/fp-sns-motenv1.html)

## STM32 ST-LINK utility

<http://www.st.com/en/development-tools/stsw-link004.html>

## Interfaccia PC

[http://www.st.com/content/st\\_com/en/products/embedded-software/evaluation-tool-software/unicleo-gui.html](http://www.st.com/content/st_com/en/products/embedded-software/evaluation-tool-software/unicleo-gui.html)

## App per Smartphone

[http://www.st.com/content/st\\_com/en/products/embedded-software/wireless-connectivity-software/bluems.html](http://www.st.com/content/st_com/en/products/embedded-software/wireless-connectivity-software/bluems.html)



**Educational part:** What is a Sensor? What are the MEMS? Overview of ST Sensors?

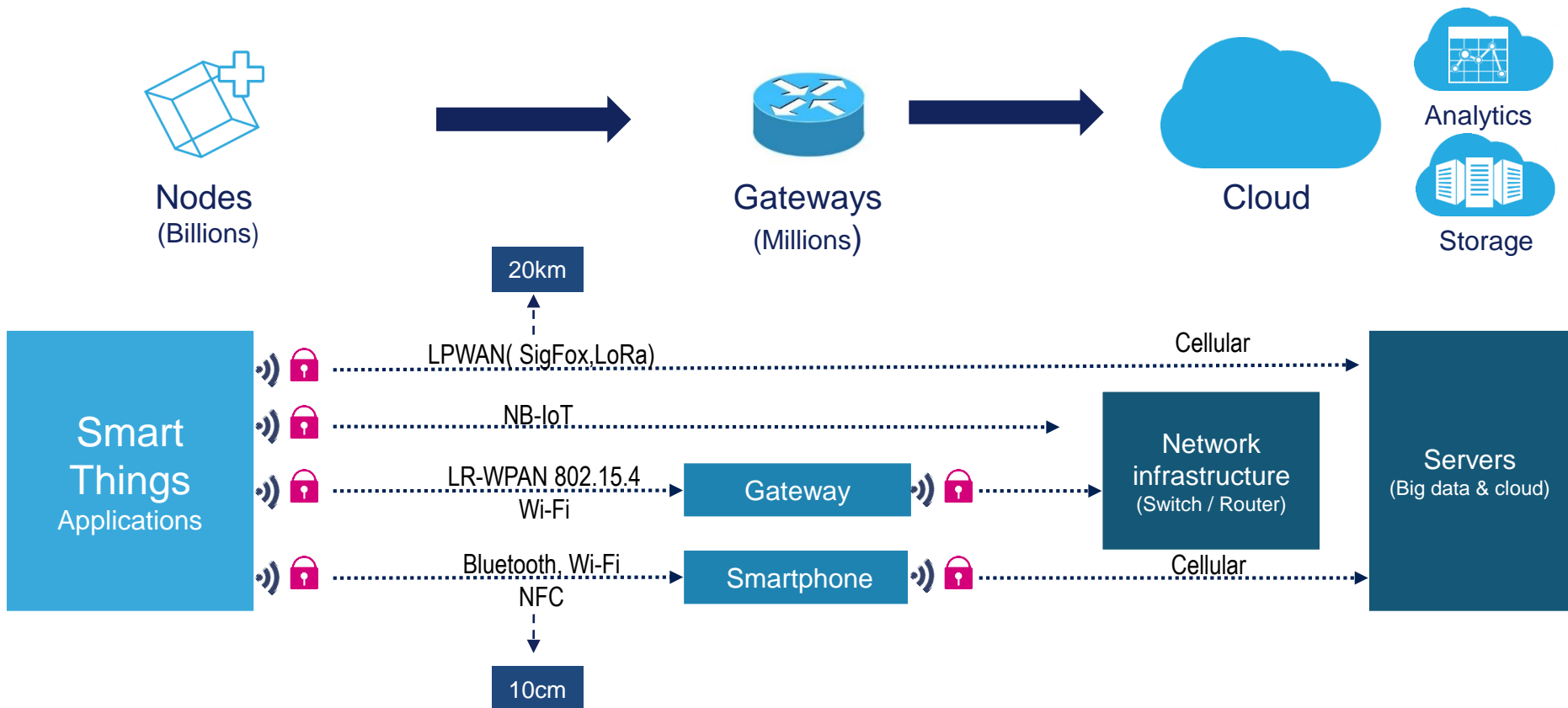
**Lab:** How to connect a Sensor Board on Nucleo? How to acquire Data on PC?

**Break**

**Education part:** ST portfolio on Connectivity. What is a Bluetooth Low Energy?

**Lab:** How to connect Bluetooth over the Sensors? How to retrieve data on Smartphone?

Any system able to leverage the Internet and its ecosystem



# What is Bluetooth® SMART (Low Energy)

Bluetooth® SMART is the latest enhancement of Bluetooth standard (V4.0), ultra-low power technology.

- Bluetooth® **SMART** enables **devices with coin cell batteries to be wirelessly connected**
- Bluetooth® **SMART** devices are used in a wide range of sensor applications transmitting small amounts of data.
  - Automotive
  - Sport and fitness
  - Healthcare
  - Entertainment
  - Home automation
  - Security and proximity



# The Bluetooth® SMART Marks Overview



- Ultra low power consumption being a pure low energy implementation
- Months to years of lifetime on a standard coin cell battery



- Classic Bluetooth + Bluetooth low energy on a single chip (small price delta)
- These are the hub devices of the Bluetooth ecosystem

# Bluetooth<sup>®</sup> SMART vs Classic Bluetooth<sup>®</sup>

Feature	Bluetooth <sup>®</sup> Classic Mode	Bluetooth <sup>®</sup> Low energy Mode
Power Consumption Range	Between 1mA and 30mA	Between 1µA and 15mA
Over the air data rate	1–3 Mbit/s	1 Mbit/s
Range (typical)	30 m	50 m
Max TX power	+20 dBm (class 1) +4 dBm (class 2)	+10 dBm
RF Channels	79	40
Connection Time	100 ms	3 ms
Max packet Size	2875 µs = 1021 Bytes	328 µs = 27 Bytes
Encryption	Safer+	AES-128

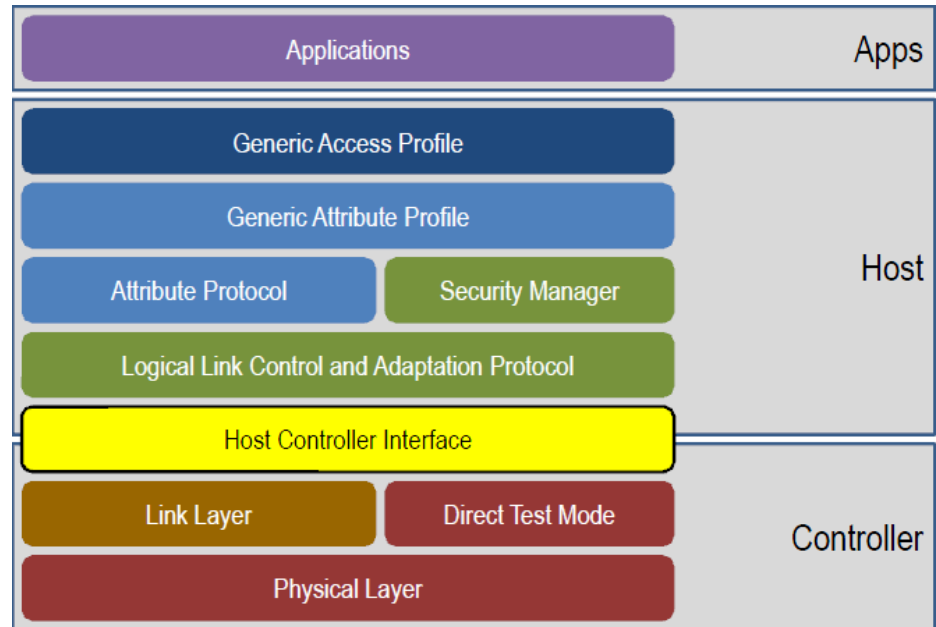
***Lower data-rate + Shorter connection time + less channels + smaller packet size***

***→ Bluetooth Smart<sup>®</sup> aims at saving current consumption !***

# Bluetooth Low Energy - Summary

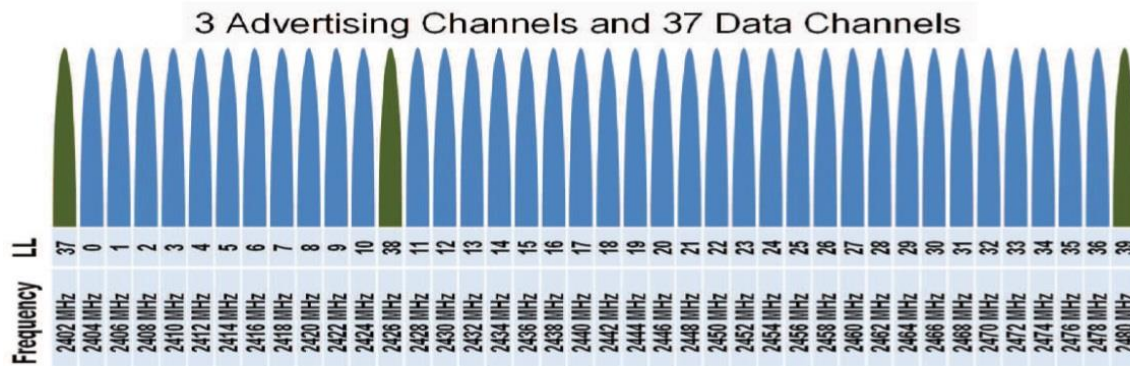
- BLE technology

- Short range wireless ISM 2.4 GHz
- Optimized for ultra low power
  - <15 mA peak current
  - <1 uA average current
- Fast connection procedure
- Client server architecture
- Low data throughput application



- Security including privacy/authentication/authorization

- Based on encryption AES128





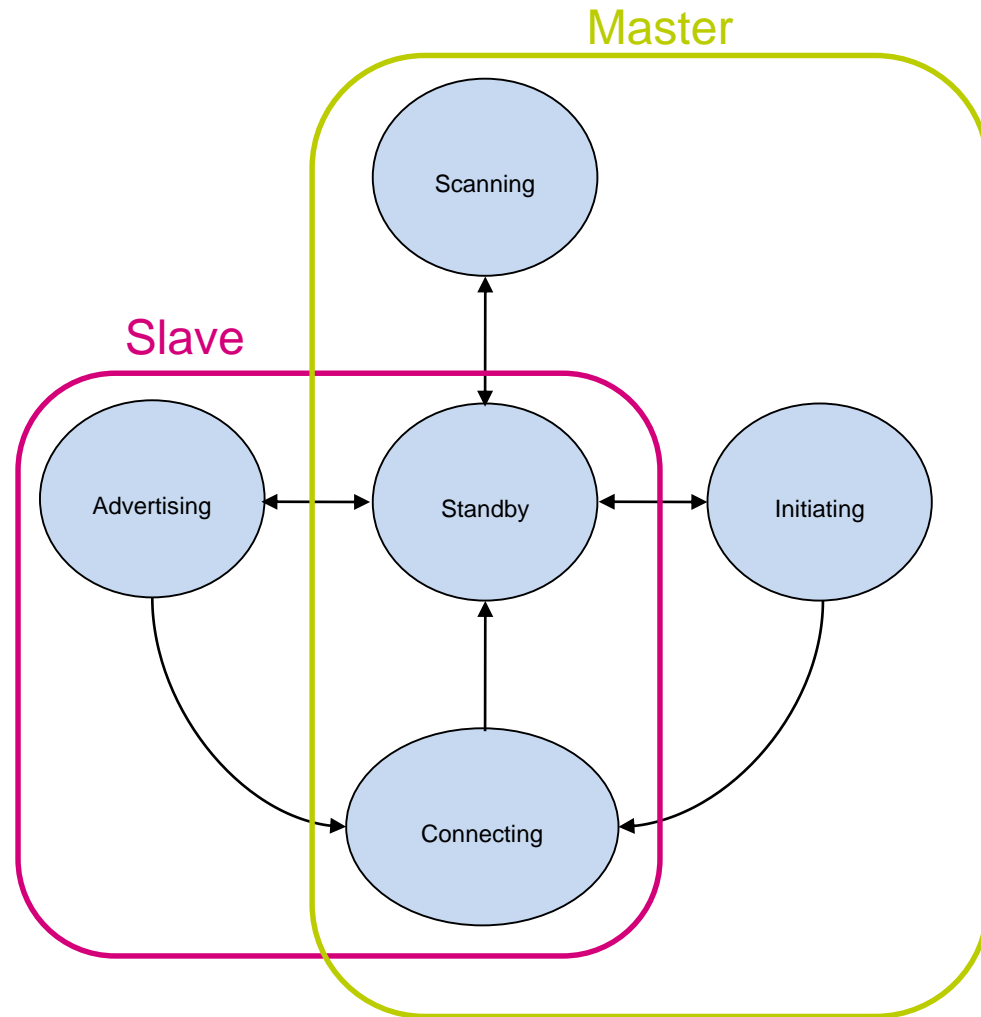
# Bluetooth Smart Protocol Stack

- Two types of channel

- Advertising channels (3)
  - Used for Discoverability
  - Used for Broadcasting/Observing
- Data Channels (37)
  - Data channel Packets
  - Used to send application data in

- Modes of operations

- Standby :  
Does not transmit or receive packets
- Advertising :  
Broadcasts advertisements in advertising channels
- Scanning :  
Looks for advertisers
- Initiating :  
Initiates connection to advertiser
- Connection
  - Initiator device will be in Master Role
    - Communicates with device in the Slave role, defines timings of transmissions
  - Advertiser device will be in Slave Role
    - Communicates with single device in Master Role



# ST Bluetooth® SMART profiles

Time

Alert notification

Running speed & cadence

Proximity

Blood Pressure

Glucose Meter

Find Me

HOGP (HID)

Cycling speed & cadence

Heart Rate

Scan Parameters

Cycling Power

Phone Alert

Health Thermometer

Location and Navigation

# Bluetooth Low Energy Expansion Board Hardware Overview

## X-NUCLEO-IDB05A1 Hardware Description

- The X-NUCLEO-IDB05A1 is a Bluetooth Low Energy (BLE) evaluation and development board system, designed around ST's SPBTLE-RF Bluetooth Low Energy module based on BlueNRG-MS.
- The BlueNRG-MS processor hosted in the SPBTLE-RF module communicates with the STM32 Nucleo developer board host microcontroller through an SPI link available on the Arduino UNO R3 connector.

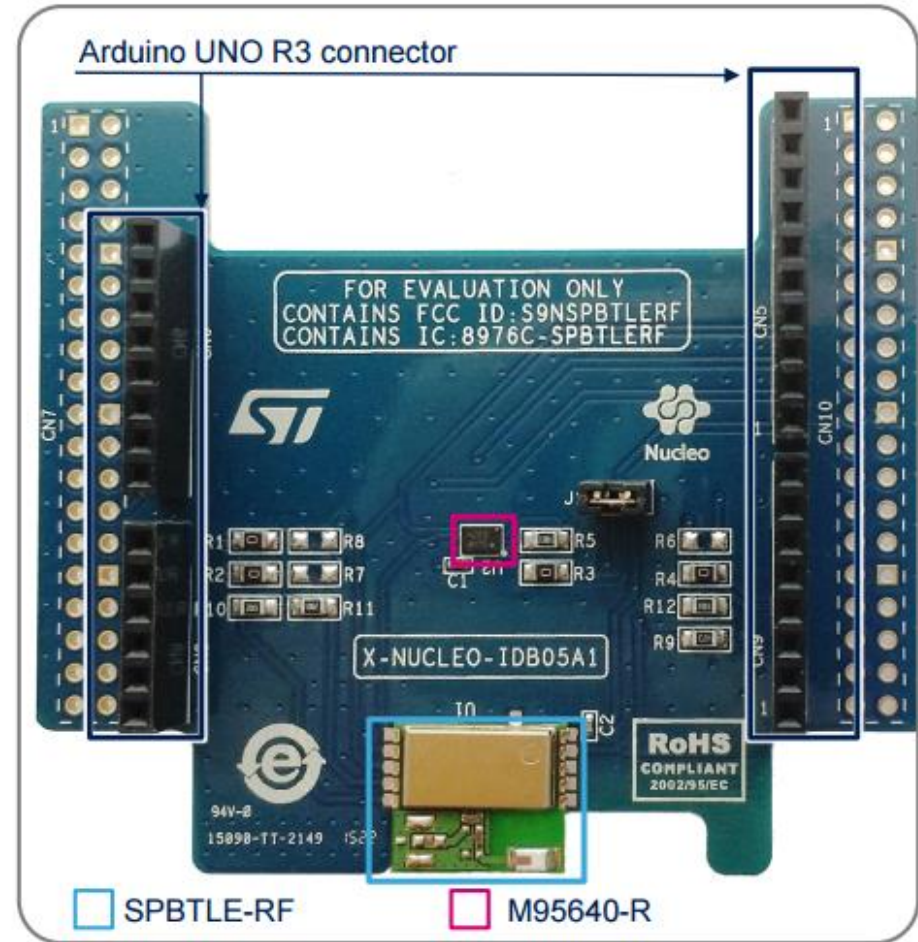
### Key Products on board

#### SPBTLE-RF

Bluetooth Low Energy, FCC and IC certified, module based on Bluetooth® Low Energy wireless network processor BlueNRG-MS, BLE4.1 compliant. SPBTLE-RF integrates a BALF-NRG-01D3 balun and a chip antenna. It embeds 32 MHz and 32.768 kHz crystal oscillators for the BlueNRG-MS.

#### M95640-R

64-Kbit serial SPI bus EEPROM with high-speed clock interface



Latest info available at [www.st.com](http://www.st.com)  
**X-NUCLEO-IDB05A1**

## Libreria per Bluetooth

[http://www.st.com/content/st\\_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/x-cube-ble1.html](http://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/x-cube-ble1.html)

## Function Pack STM32 ODE per Internet of Thing

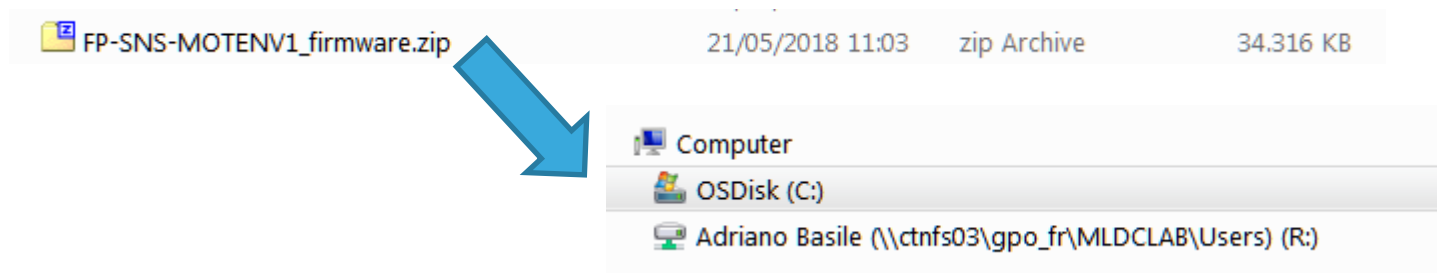
[http://www.st.com/content/st\\_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32-ode-function-pack-sw/fp-sns-motenv1.html](http://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32-ode-function-pack-sw/fp-sns-motenv1.html)

## STM32 ST-LINK utility

<http://www.st.com/en/development-tools/stsw-link004.html>

## App per Smartphone

[http://www.st.com/content/st\\_com/en/products/embedded-software/wireless-connectivity-software/bluems.html](http://www.st.com/content/st_com/en/products/embedded-software/wireless-connectivity-software/bluems.html)



# FP-SNS-MOTENV1

## Software Overview

### Software Description

FP-SNS-MOTENV1 is an STM32Cube function pack, which lets you connect your IoT node to a smartphone via BLE and uses a suitable Android™ or iOS™ application, such as the BlueMS app, to view real-time motion and environmental (such as temperature, relative humidity, carbon monoxide) sensor data, and gas gauge level.

This package also enables advanced functions such as the sensor data fusion and accelerometer-based real-time activity recognition and MEMS sensor data logging on SD card.

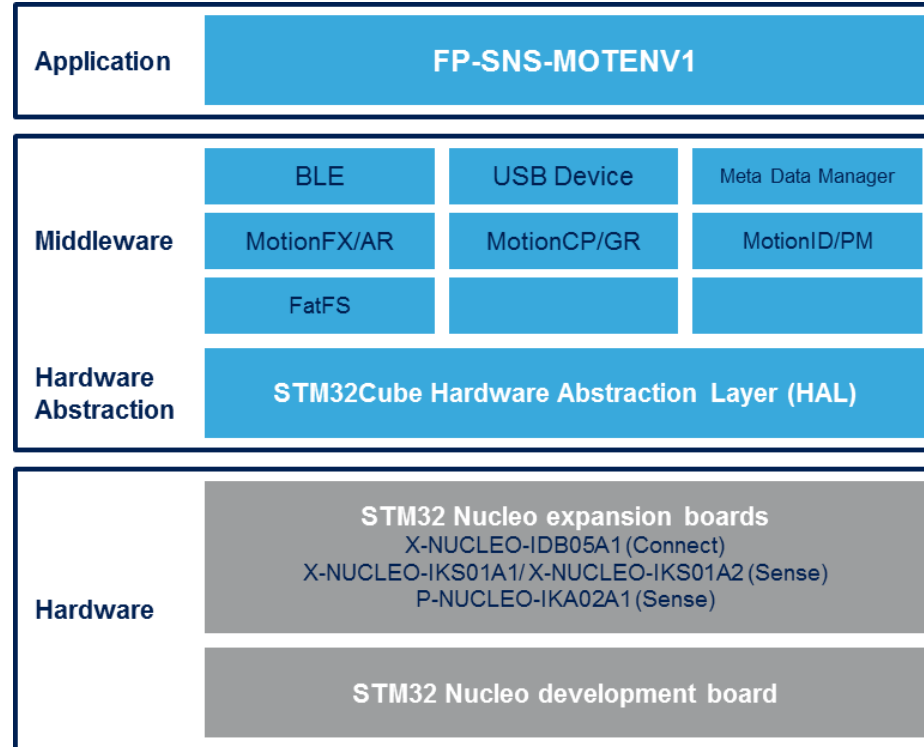
Together with the suggested combination of STM32 and ST devices, it can be used to develop specific wearable and environmental monitoring applications, or smart things applications in general.

The software runs on the STM32 microcontroller and includes all the necessary drivers to recognize the devices on the STM32 Nucleo development board and expansion boards.

### Key features

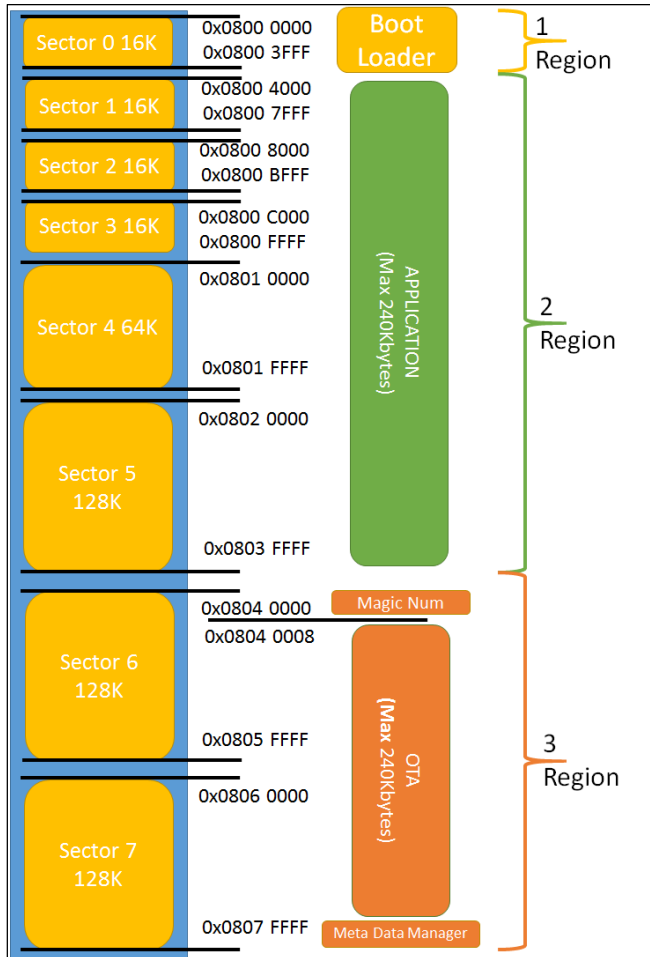
- Complete firmware to develop an IoT node with BLE connectivity, environmental and motion sensors
- Middleware libraries for sensor data fusion and accelerometer-based real-time activity recognition and SD card data logging
- Compatible with BlueMS applications for Android/iOS, to perform sensor data reading, motion algorithm features demo and firmware update (FOTA)
- Example implementation available for the X-NUCLEO-IKS01A2 (or X-NUCLEO-IKS01A1), P-NUCLEO-IKA02A1, and X-NUCLEO-IDB05A1 (or X-NUCLEO-IDB04A1) connected to a NUCLEO-F401RE or NUCLEO-L476RG or NUCLEO-L053R8 board
- Easy portability across different MCU families, thanks to the STM32Cube
- Free, user-friendly license terms

### Overall Software Architecture



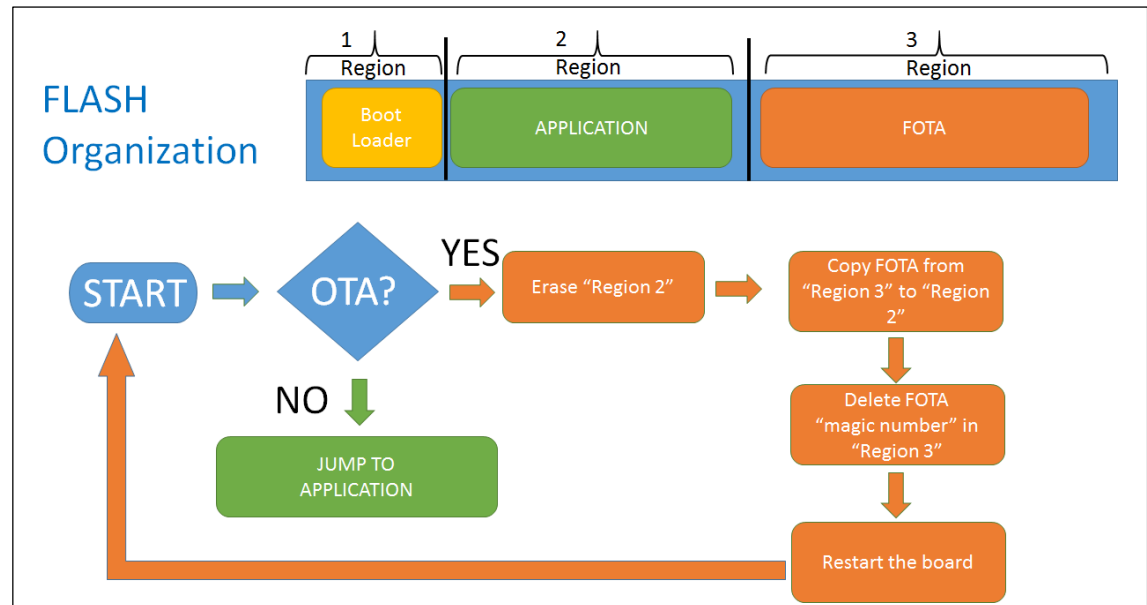
Latest info available at [www.st.com](http://www.st.com)  
**FP-SNS-MOTENV1**



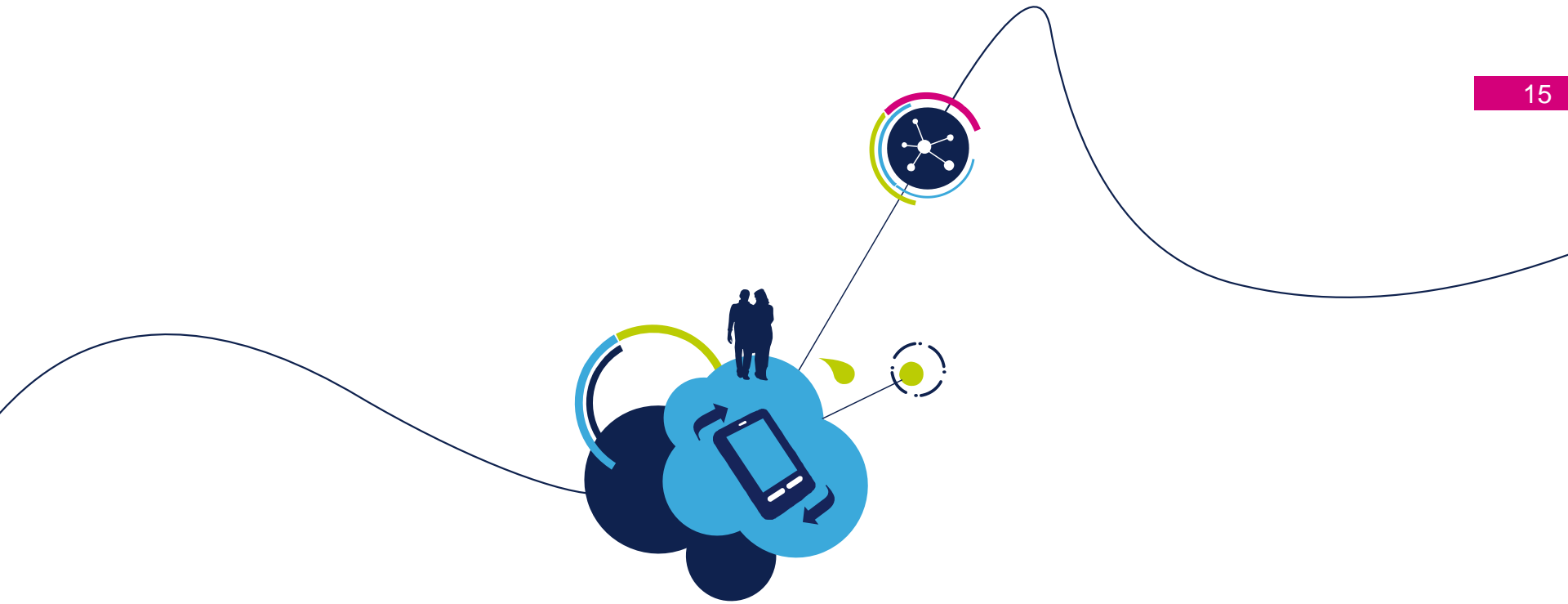


FP-SNS-MOTENV1 Flash structure

➔ For NUCLEO-F401RE and NUCLEO-L476RG











FP-SNS-MOTENV1 boot sequence



# Go Live!

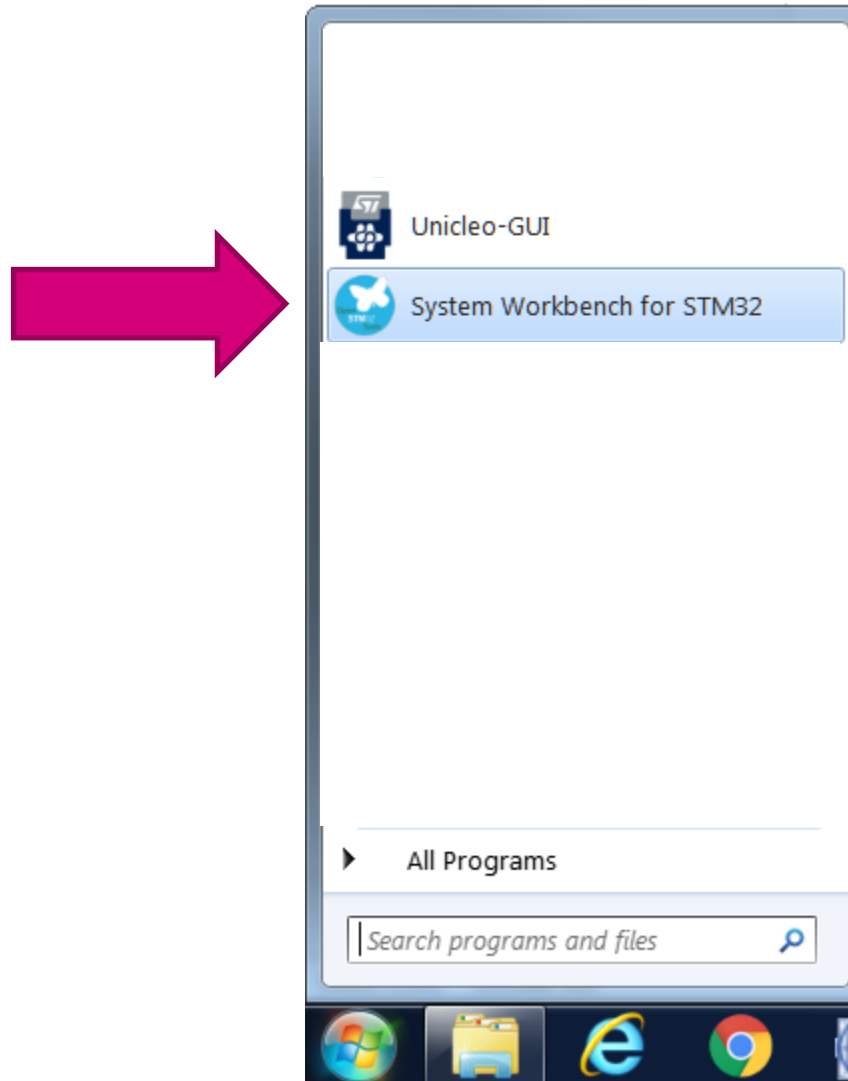
# FP-SNS-MOTENV1 package structure

C:\STM32CubeFunctionPack\_MOTENV1\_V3.3.0

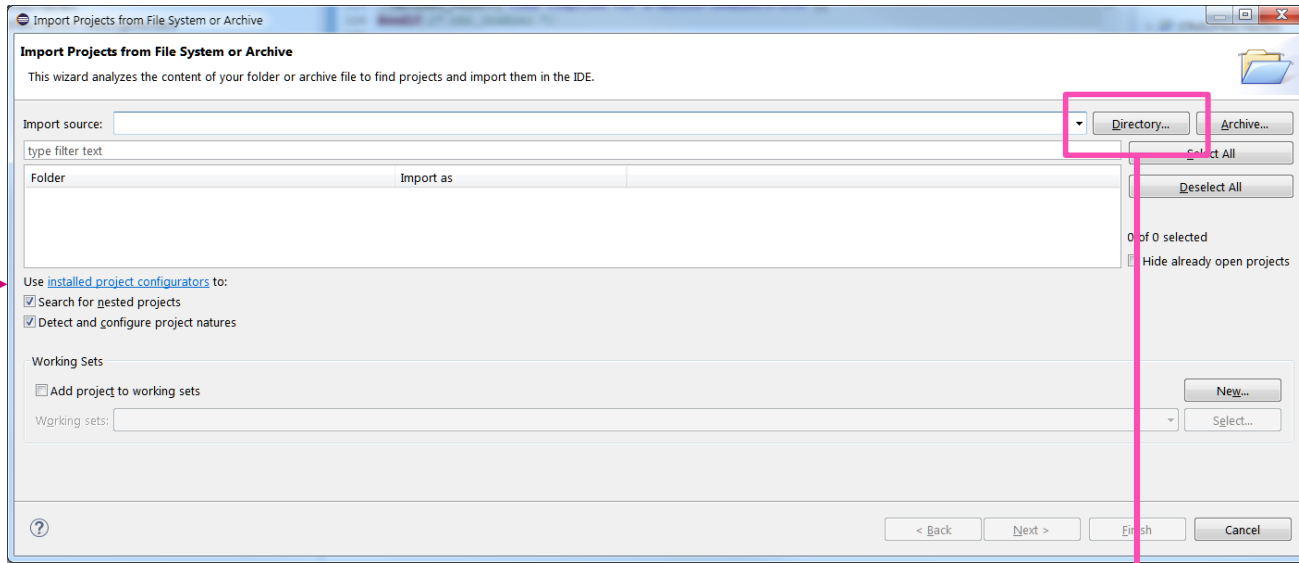
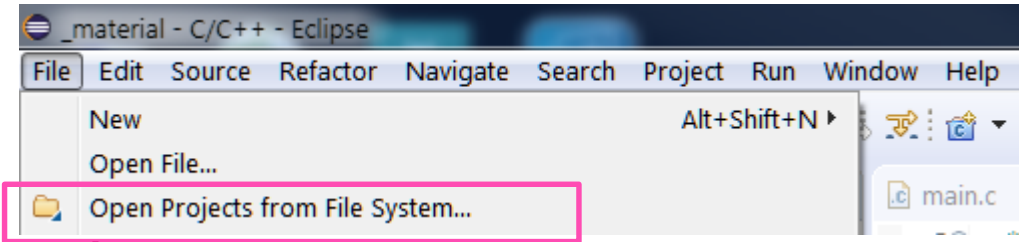
Name	
 _htmresc	
 Documentation	← Docs
 Drivers	← BSP, HAL and drivers
 Middlewares	← BlueNRG, Motion Libs, FatFs
 Projects	← Application example
 Utilities	← Boot loader binary
 package.xml	
 Release_Notes.html	



# Launch System Workbench for STM32

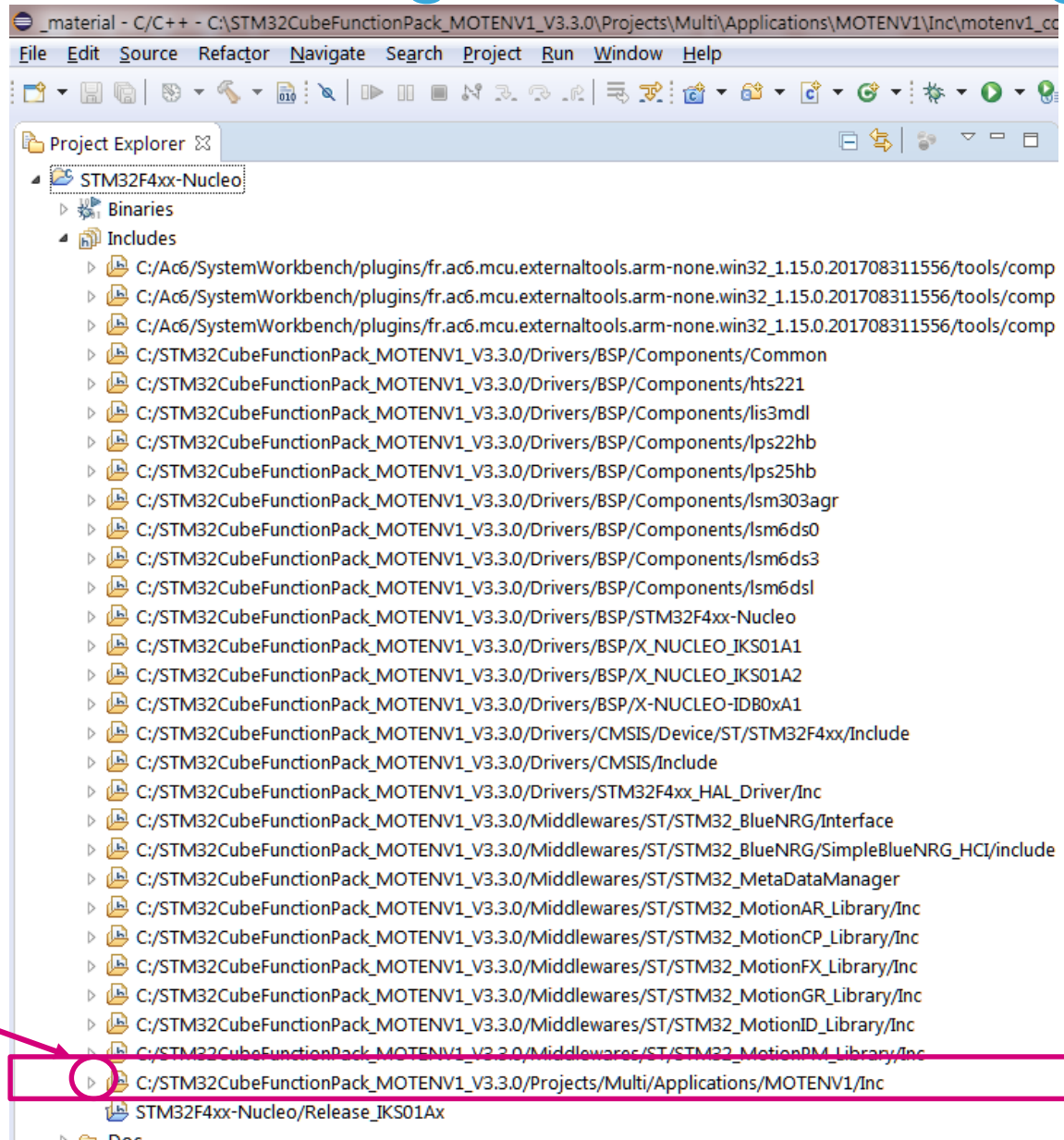


# Open the MOTENV1 Project...



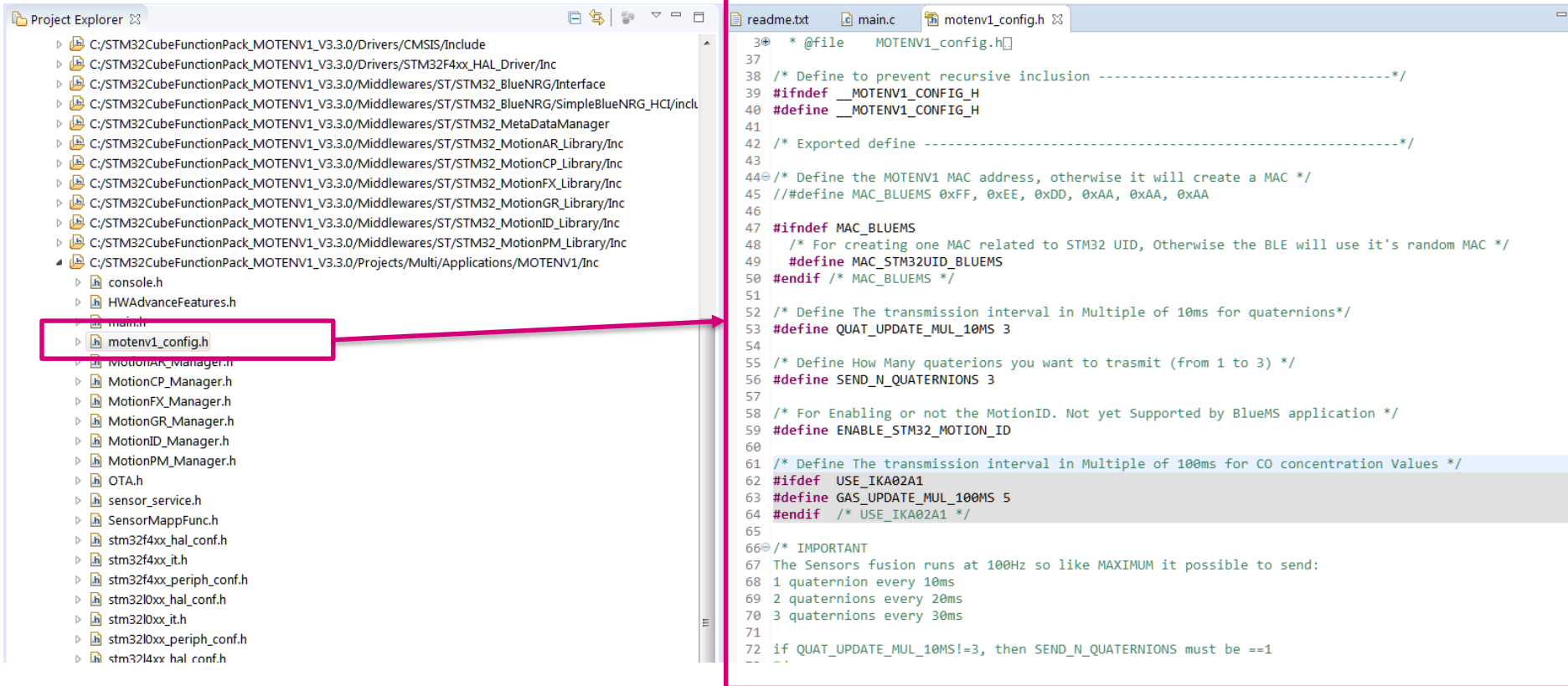
.\Projects\Multi\Applications\MOTENV1\SW4STM32\STM32F401RE-Nucleo\STM32F4xx-Nucleo

# Change Board Naming (1/3)



Click here

# Change Board Naming (2/3)



The image shows a screenshot of an IDE with two windows. The left window is the Project Explorer, showing a file tree for a project. The file 'motenv1\_config.h' is highlighted with a red box. The right window is the Code Editor, showing the contents of 'motenv1\_config.h'. The code defines various macros for the MOTENV1 board, including MAC address, quaternions, and gas updates.

```
37 * @file MOTENV1_config.h
38 /* Define to prevent recursive inclusion -----*/
39 #ifndef __MOTENV1_CONFIG_H
40 #define __MOTENV1_CONFIG_H
41
42 /* Exported define -----*/
43
44 /* Define the MOTENV1 MAC address, otherwise it will create a MAC */
45 // #define MAC_BLUEMS 0xFF, 0xEE, 0xDD, 0xAA, 0xAA, 0xAA
46
47 #ifndef MAC_BLUEMS
48 /* For creating one MAC related to STM32 UID, Otherwise the BLE will use it's random MAC */
49 #define MAC_STM32UID_BLUEMS
50 #endif /* MAC_BLUEMS */
51
52 /* Define The transmission interval in Multiple of 10ms for quaternions*/
53 #define QUAT_UPDATE_MUL_10MS 3
54
55 /* Define How Many quaternions you want to trasmit (from 1 to 3) */
56 #define SEND_N_QUATERNIONS 3
57
58 /* For Enabling or not the MotionID. Not yet Supported by BlueMS application */
59 #define ENABLE_STM32_MOTION_ID
60
61 /* Define The transmission interval in Multiple of 100ms for CO concentration Values */
62 #ifndef USE_IKA02A1
63 #define GAS_UPDATE_MUL_100MS 5
64 #endif /* USE_IKA02A1 */
65
66 /* IMPORTANT
67 The Sensors fusion runs at 100Hz so like MAXIMUM it possible to send:
68 1 quaternion every 10ms
69 2 quaternions every 20ms
70 3 quaternions every 30ms
71
72 if QUAT_UPDATE_MUL_10MS!=3, then SEND_N_QUATERNIONS must be ==1
73 ..
```

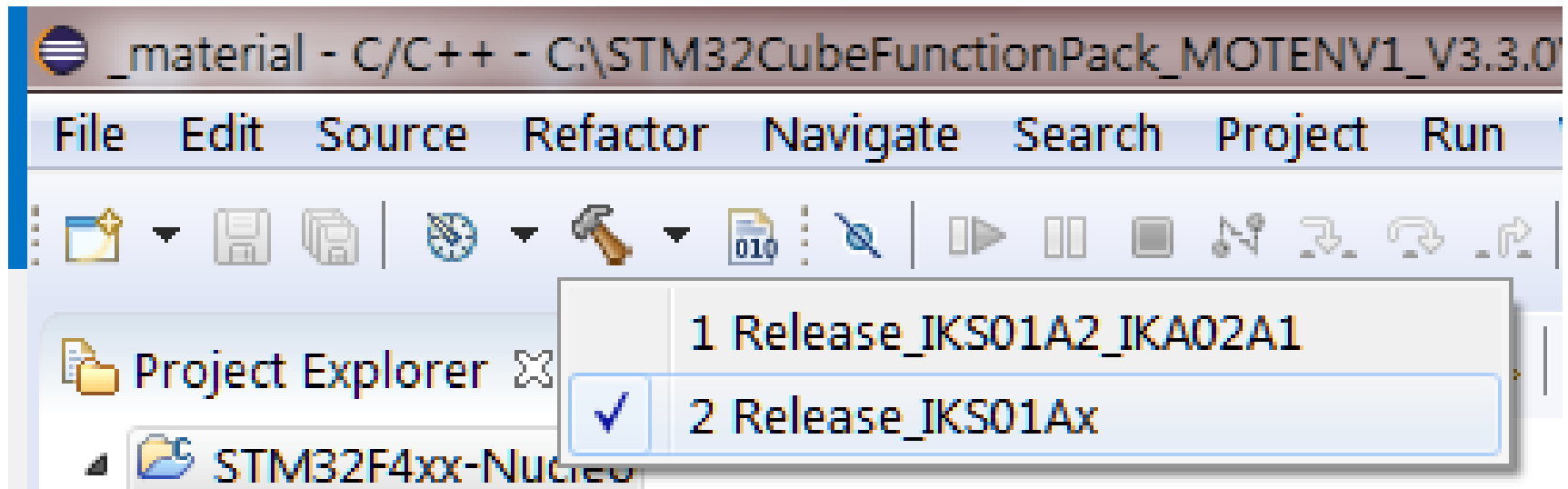
# Change Board Naming (3/3)

21

```
Projects\Multi\Applications\MOTENV1\Inc\motenv1_config.h - Eclipse
Window Help
readme.txt main.c *motenv1_config.h
69 2 quaternions every 20ms
70 3 quaternions every 30ms
71
72 if QUAT_UPDATE_MUL_10MS!=3, then SEND_N_QUATERNIONS must be ==1
73 */
74
75 /***** Debug Defines *****/
76 #ifndef USE_STM32L0XX_NUCLEO
77 /* For Nucleo F401RE/L476RG it's enable by default */
78 #define MOTENV1_ENABLE_PRINTF
79 #endif /* USE_STM32L0XX_NUCLEO */
80
81
82 /* For enabling connection and notification subscriptions debug */
83 #define MOTENV1_DEBUG_CONNECTION
84
85 /* For enabling trasmission for notified services (except for quaternions) */
86 #define MOTENV1_DEBUG_NOTIFY_TRAMISSION
87
88 /***** Don't Change the following defines *****/
89
90 /* Package Version only numbers 0->9 */
91 #define MOTENV1_VERSION_MAJOR '3'
92 #define MOTENV1_VERSION_MINOR '3'
93 #define MOTENV1_VERSION_PATCH '0'
94
95 /* Define the MOTENV1 Name MUST be 7 char long */
96 #define NAME_BLUEMS 'M','E','1','V',MOTENV1_VERSION_MAJOR,MOTENV1_VERSION_MINOR,MOTENV1_VERSION_PATCH
97
98 /* Package Name */
99 #define MOTENV1_PACKAGENAME "FP-SNS-MOTENV1"
100
101 #ifdef MOTENV1_ENABLE_PRINTF
102 #define MOTENV1_PRINTF(...) printf(__VA_ARGS__)
103 #else /* MOTENV1_ENABLE_PRINTF */
104 #define MOTENV1_PRINTF(...)
105 #endif /* MOTENV1_ENABLE_PRINTF */
106 |
107 /* STM32 Unique ID */
108 #ifdef USE_STM32F4XX_NUCLEO
```

Replace: 'M','E','1','V' with .... And SAVE!

# Check Release



# If Build does not start, force it ...

The screenshot shows the Eclipse IDE interface. The 'Project' menu is open, and the 'Build All' option (with the keyboard shortcut Ctrl+B) is highlighted with a red box. The background editor shows the 'motenv1\_config.h' file with the following code:

```
82 /* For enabling connection and notification subscriptions debug */
83 #define MOTENV1_DEBUG_CONNECTION
84
85 /* For enabling trasmission for notified services (except for quaternions) */
86 #define MOTENV1_DEBUG_NOTIFY_TRANMISSION
87
88 /****** Don't Change the following defines *****/
89
90 /* Package Version only numbers 0->9 */
91 #define MOTENV1_VERSION_MAJOR '3'
92 #define MOTENV1_VERSION_MINOR '3'
93 #define MOTENV1_VERSION_PATCH '0'
94
95 /* Define the MOTENV1 Name MUST be 7 char long */
96 #define NAME_BLUEMS 'B','B','1','V',MOTENV1_VERSION_MAJOR,MOTENV1_VERSION_MINOR,MOTENV1_VERSION_PATCH
97
98 /* Package Name */
99 #define MOTENV1_PACKAGENAME "FP-SNS-MOTENV1"
100
101 #ifndef MOTENV1_ENABLE_PRINTF
102 #define MOTENV1_PRINTF(...) printf(__VA_ARGS__)
103 #else /* MOTENV1_ENABLE_PRINTF */
104 #define MOTENV1_PRINTF(...)
105 #endif /* MOTENV1_ENABLE_PRINTF */
106
107 /* STM32 Unique ID */
108 #ifndef USE_STM32F4XX_NUCLEO
```

The console at the bottom shows the following output:

```
CDT Build Console [STM32F4xx-Nucleo-DataLogExtended]
15:09:30 **** Incremental Build of configuration Debug for project STM32F4xx-Nucleo-DataLogExtended ****
```

# Check Generated Binary File

File Explorer window showing the contents of the directory: Multi > Applications > MOTENV1 > SW4STM32 > STM32F401RE-Nucleo > STM32F4xx-Nucleo > Release\_IKS01Ax.

The search bar contains: Search Release\_IKS01Ax

The main pane displays a list of 11 items:

Name	Date modified	Type	Size
Drivers	21/05/2018 11:52	File folder	
FP-SNS-MOTENV1_NF401	21/05/2018 11:52	File folder	
Middlewares	21/05/2018 11:52	File folder	
makefile	21/05/2018 12:15	File	3 KB
MOTENV1_IKS01Ax_NucleoF401.bin	21/05/2018 12:12	BIN File	206 KB
MOTENV1_IKS01Ax_NucleoF401.elf	21/05/2018 12:12	ELF File	3.029 KB
MOTENV1_IKS01Ax_NucleoF401_BL.bin	21/05/2018 12:19	BIN File	222 KB
objects.list	21/05/2018 12:15	LIST File	5 KB
objects.mk	21/05/2018 12:10	MK File	1 KB
output.map	21/05/2018 12:12	Text Document	1.741 KB
sources.mk	21/05/2018 12:14	MK File	1 KB

The status bar at the bottom shows: 11 items



# How Install the code after compiling the project (1/2)

25

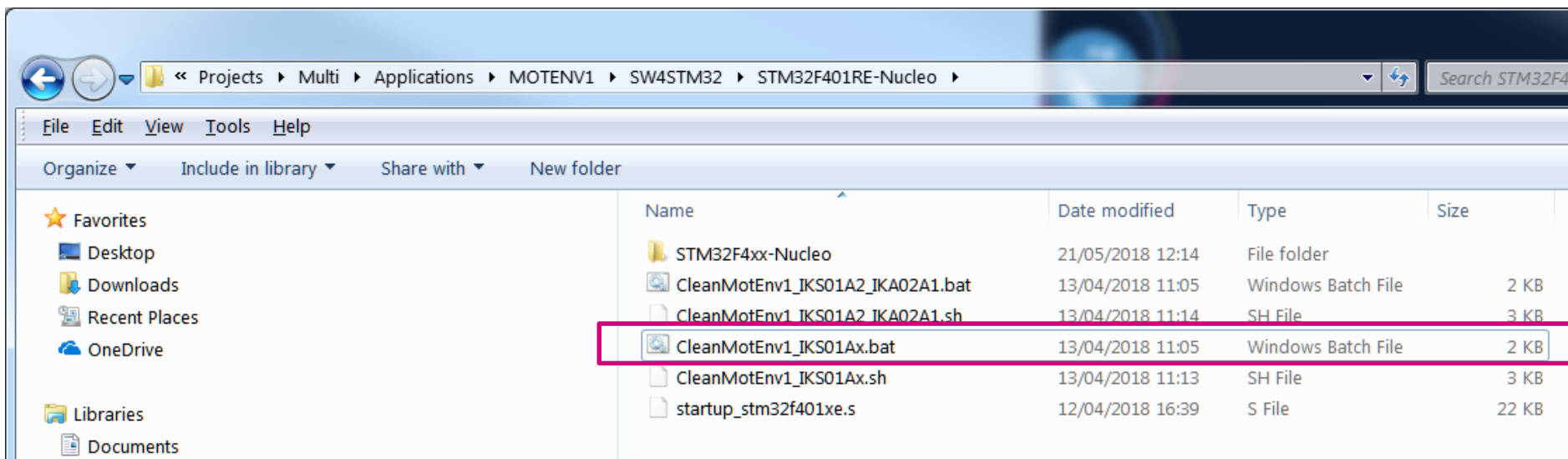
- After compile the project with your preferred IDE



- On Windows: for each IDE and for each platform there is one script called **“CleanMotEnv1.bat”**

- For Linux/iOS: only for Openstm32 IDE and for each platform there is one script called **“CleanMotEnv1.sh”**

*It's necessary to edit this file for setting the right installation and Library path*



# How Install the code after compiling the project (2/2)

• These scripts perform the following steps:

1. Full Flash Erase
2. Flash the right BootLoader at the right position (0x08000000)
3. Flash the MOTENV1 firmware at the right position (0x08004000)  
This is the firmware that was compiled with the IDE  
This firmware is compatible with the FOTA update procedure
4. Save a complete **Binary** FW that includes both MOTENV1 and the BootLoader

This **Binary** can be directly flashed to a supported STM32 Nucleo or SensorTile board using the ST-Link or by doing “Drag & Drop” (the latter only for STM32 Nucleo boards)

*Important Note:* this additional pre-compiled binary is not compatible with the FOTA update procedure

```
ca: C:\Windows\system32\cmd.exe
/***** Clean BlueMicrosystem1 *****/
/***** Full Chip Erase *****/
/*****
STM32 ST-LINK CLI v2.3.0
STM32 ST-LINK Command Line Interface

ST-LINK SN : 0676FF50528767145452
ST-LINK Firmware version : U2J24M11
Connected via SWD.
SWD Frequency = 4000K.
Target voltage = 3.2 U.
Connection mode : Normal.
Device ID:0x415
Device flash Size : 1024 Kbytes
Device family :STM32L47x/L48x
Full chip erase...
Flash memory erased.

*****/
/***** Install BootLoader *****/
/*****
STM32 ST-LINK CLI v2.3.0
STM32 ST-LINK Command Line Interface

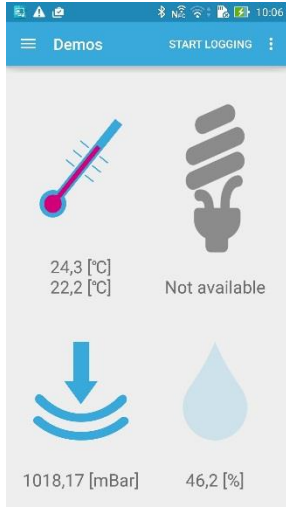
ST-LINK SN : 0676FF50528767145452
ST-LINK Firmware version : U2J24M11
Connected via SWD.
SWD Frequency = 4000K.
Target voltage = 3.2 U.
Connection mode : Normal.
Device ID:0x415
Device flash Size : 1024 Kbytes
Device family :STM32L47x/L48x
Loading file...
Flash Programming:
File : ..\..\..\..\..\Utilities\BootLoader\STM32L476RG\BootLoaderL4.bin
Address : 0x08000000
Memory programming... 100%
Reading and verifying device memory... 100%
Memory programmed in 0s and 468ms.
Verification...OK
Programming Complete.

*****/
/***** Install BlueMicrosystem1 *****/
/*****
STM32 ST-LINK CLI v2.3.0
STM32 ST-LINK Command Line Interface

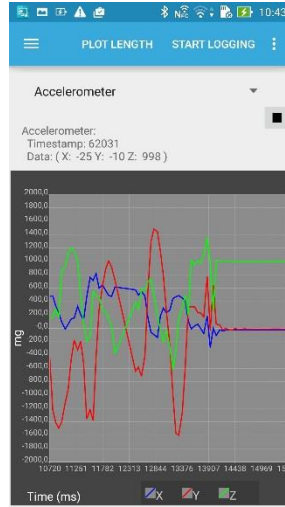
ST-LINK SN : 0676FF50528767145452
ST-LINK Firmware version : U2J24M11
Connected via SWD.
SWD Frequency = 4000K.
Target voltage = 3.2 U.
Connection mode : Normal.
Device ID:0x415
Device flash Size : 1024 Kbytes
Device family :STM32L47x/L48x
Loading file...
Flash Programming:
File : STM32L476RG_SENSOR TILE\Exe\BlueMS1_ST.bin
Address : 0x08004000
Memory programming... 100%
Reading and verifying device memory... 100%
Memory programmed in 7s and 582ms.
Verification...OK
Programming Complete.
```

# Launch BlueMS Application for Android/iOS (1/4)

Hardware Features – Android Version



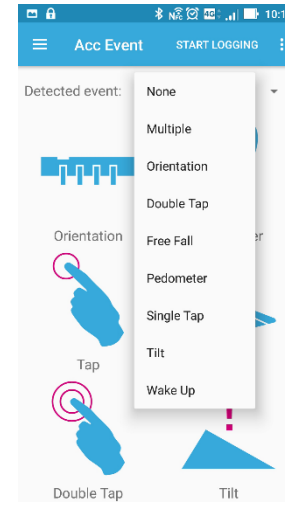
Environmental page



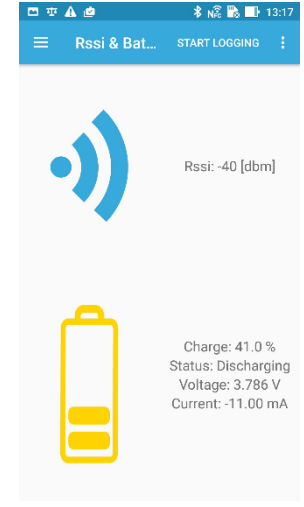
Accelerometer plot



Led Status

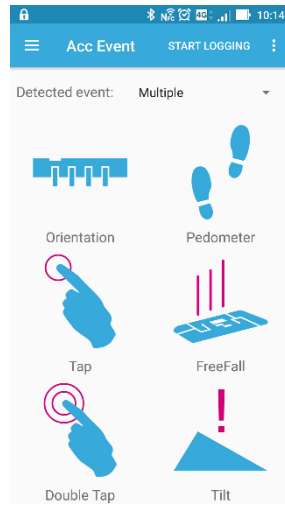


DS3/DSM/DSL Menu Events

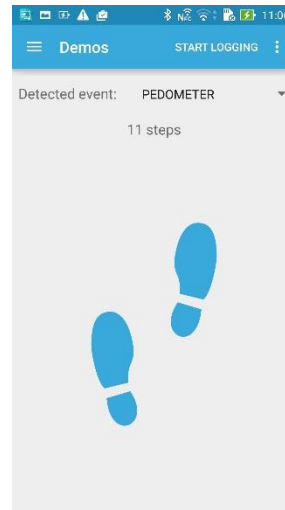


RSS & Battery Page

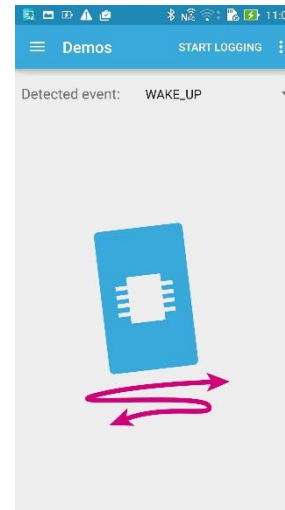
DS3/DSM/DSL Event



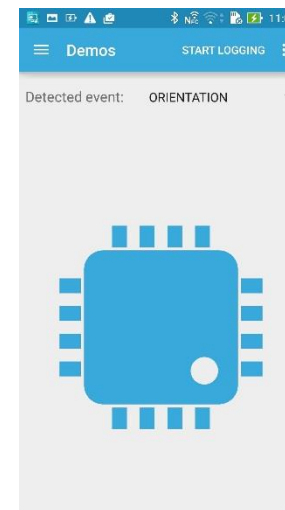
Multiple events page



Pedometer

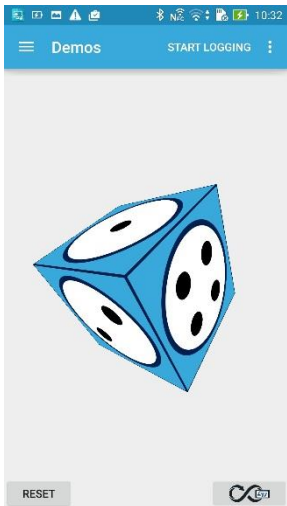


Wake Up

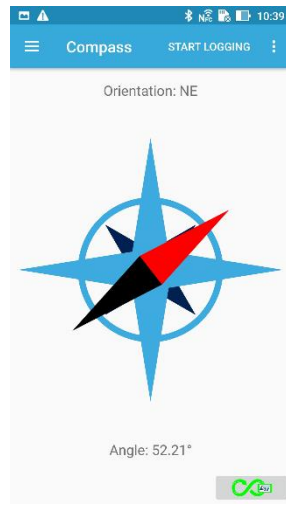


Orientation

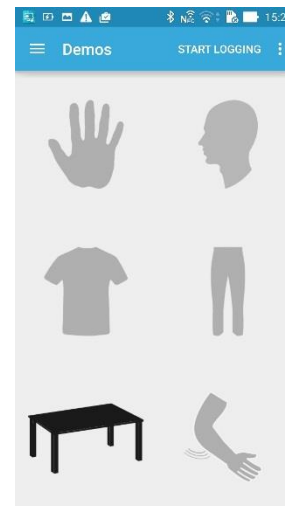
# BlueMS Application for Android/iOS (2/4)



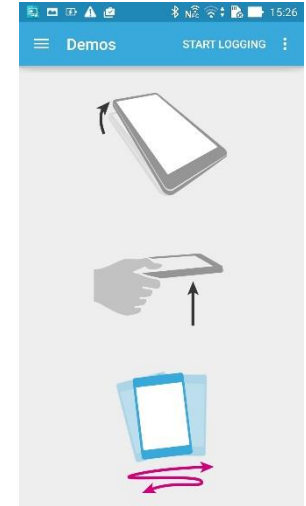
MotionFX sensor fusion page



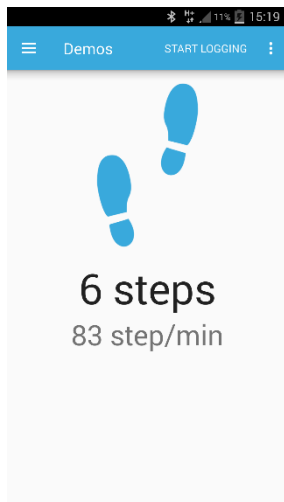
MotionFX - ecompass



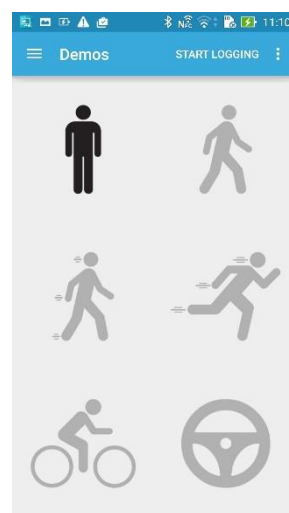
MotionCP carry position recognition page



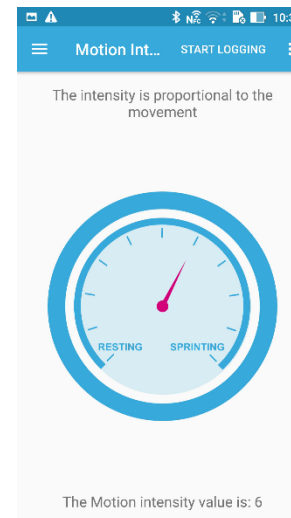
MotionGR gesture recognition page



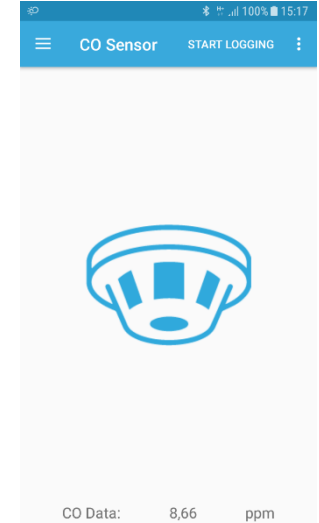
MotionPM Pedometer page



MotionAR activity recognition page



MotionID motion intensity page

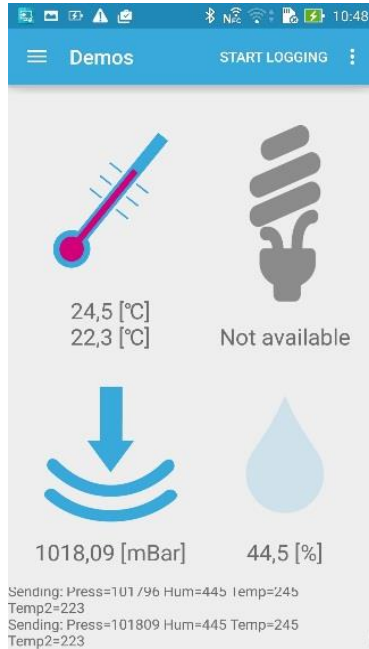


Gas concentration page

# BlueMS Application for Android/iOS (3/4)



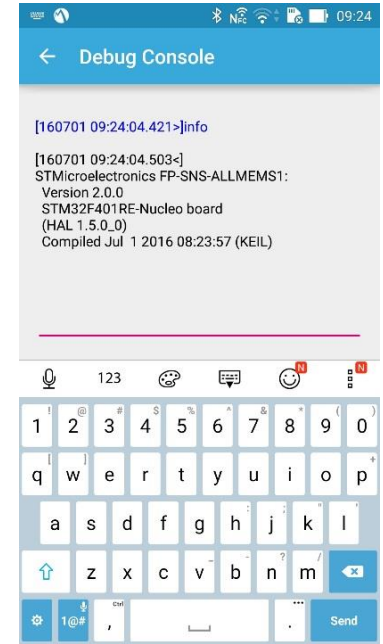
BlueMS: menu option



Serial Console (stdout/stderr)



BlueMS: menu option



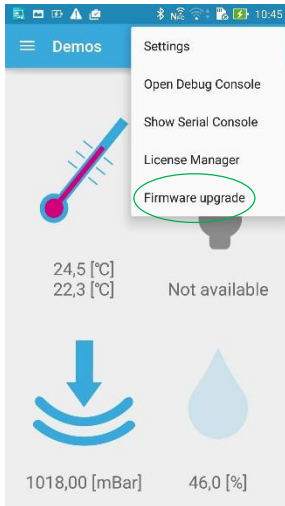
Debug Console (stdin/stdout/stderr)



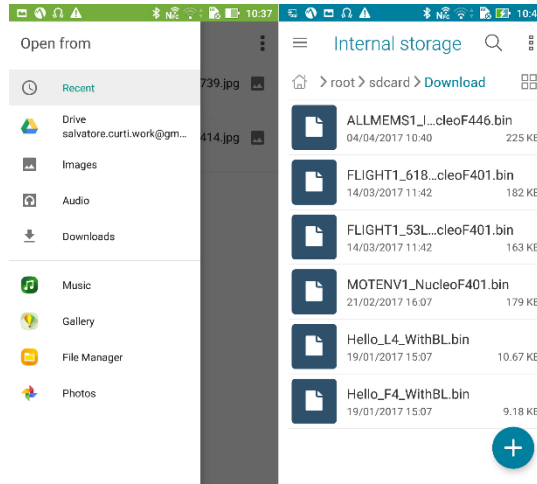
Note: data logging feature on SD card is not available with STM32 Nucleo boards

# BlueMS Application for Android/iOS (4/4)

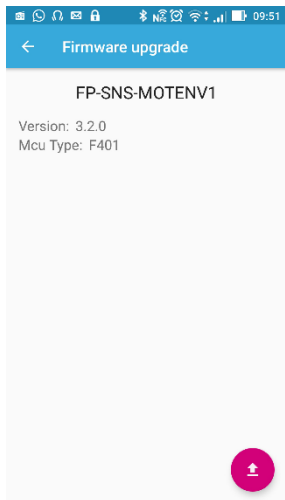
Firmware Upgrade – Android version



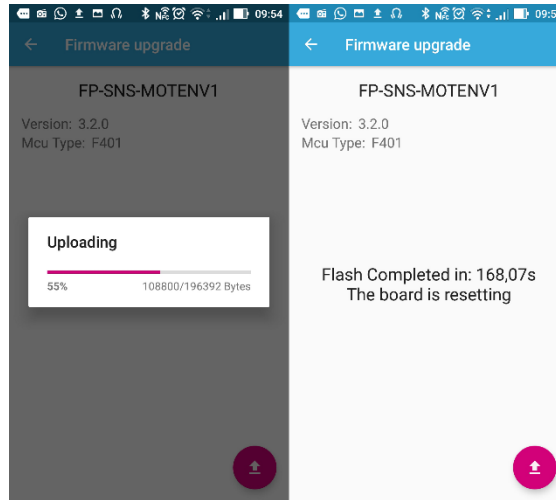
BlueMS: menu option



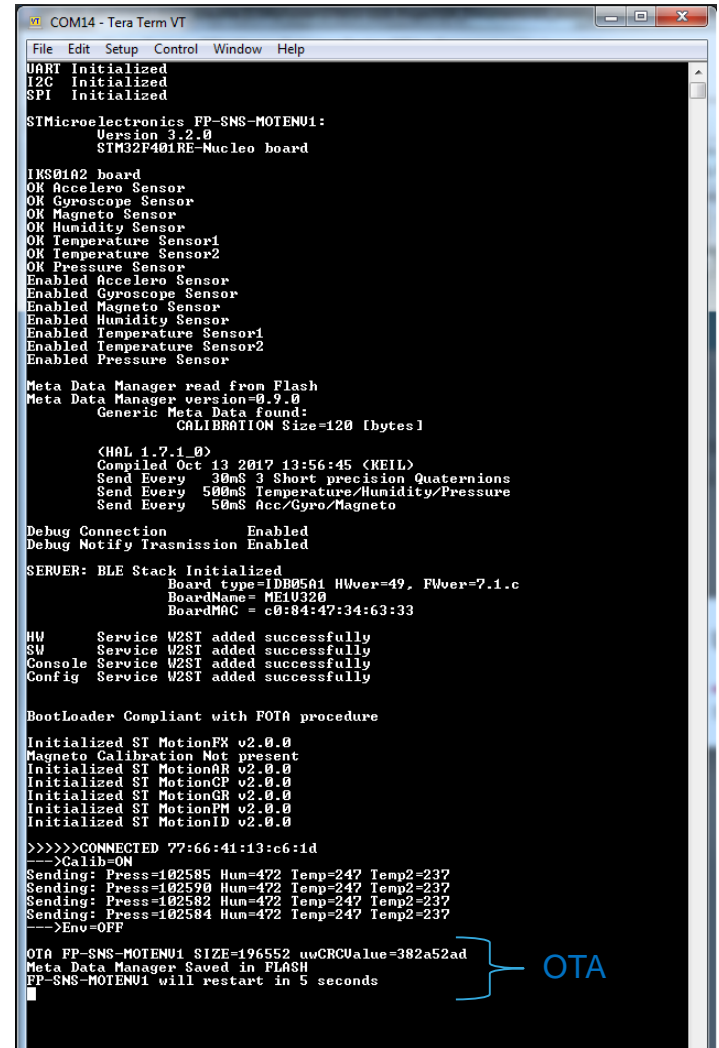
BlueMS: Firmware update file selection



BlueMS: Firmware upgrade page



BlueMS: application page during FOTA and on completion



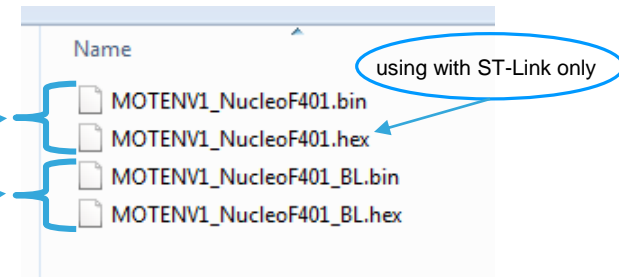
For only STM32 Nucleo F4/L4 - Terminal window information during FOTA

## • How to install the pre-compiled binary:

- There is inside the package one folder called “Binary”

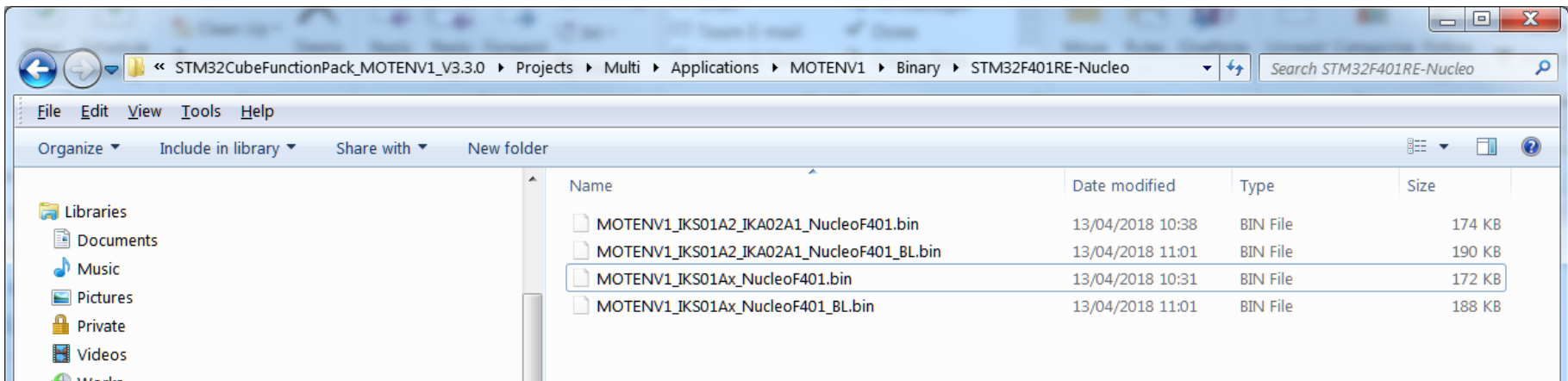
- For NUCLEO-F401RE, NUCLEO-L476RG there are 2 pre-compiled binaries:

- MOTENV1 that could be not directly flashed to Nucleo.  
(THIS COULD BE USED FOR FOTA Update)
- MOTENV1+BootLoader FW that could be directly flashed to a supported STM32 Nucleo Board using the ST-Link or by doing “Drag & Drop” (the latter only for STM32 Nucleo boards)  
(this could not be used for FOTA Update)



- For NUCLEO-F401RE and NUCLEO-L476RG two other binaries are provided for use with P-NUCLEO-IKA02A1

- For NUCLEO-L053R8 it contains a pre-compiled MOTENV FW that could be directly flashed to a STM32 Nucleo using the ST-Link or by doing “Drag & Drop”



# Documents & Related Resources (1/2)

32

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

## X-NUCLEO-IKS01A2:

- **Gerber files, BOM, Schematics**
- **DB3009:** Motion MEMS and environmental sensor expansion board for STM32 Nucleo – **Data brief**
- **UM2121:** Getting started with motion MEMS and environmental sensor expansion board for STM32 Nucleo – **User manual**

## X-CUBE-MEMS1:

- **DB2442:** Motion MEMS and environmental sensor software expansion for STM32Cube – **Data brief**
- **UM1859:** Getting started with the X-CUBE-MEMS1 motion MEMS and environmental sensor software expansion for STM32Cube – **User manual**
- Software Setup File

## X-NUCLEO-IKA02A1 (not provided):

- **DB3274:** STM32 Nucleo pack: electrochemical toxic gas sensor expansion board with CO sensor – **data brief**
- **UM2247:** Getting started with the P-NUCLEO-IKA02A1 STM32 Nucleo pack for electrochemical toxic gas sensor expansion board with CO sensor – **user manual**



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

## FP-SNS-MOTENV1

- **DB2915:** STM32 ODE function pack for IoT node with BLE connectivity and environmental and motion sensors – **data brief**
- **UM2015:** Getting started with the FP-SNS-MOTENV1 Bluetooth low energy and sensors software expansion for STM32Cube – **user manual**
- **Software setup file**

## X-NUCLEO-IDB05A1

- Gerber files, BOM, Schematic
- **DB2592:** Bluetooth Low Energy expansion board based on SPBTLE-RF module for STM32 Nucleo – **data brief**
- **UM1912:** Getting started with X-NUCLEO-IDB05A1 Bluetooth low energy expansion board based on SPBTLE-RF module for STM32 Nucleo – **user manual**