

# Simplify the Integration of Sensors and Bluetooth Low Energy (BLE) Connectivity Using the BlueTile Eval Kit

IoT Systems Development - Connectivity



WiFi Network: **STMicr**  
Password: **STMTT19**



**Technology Tour 2019**

Vancouver, BC | September 24





## BlueTile (STEVAL-BCN002V1)

- Training Material Check/Installation Help
- ARM® Keil MDK Installation
- Introduction to Bluetooth® Low Energy
- BlueTile Development Kit
- ARM® Keil MDK License Installation
- Lab 1: Getting Started with BlueTile “Serial Terminal Test”
- Lab 2: Connecting to the ST BLE Sensor app
- Lab 3: LED characteristic
- Lab 4: Accelerometer embedded events detection
- Lab 5: 9-axis Acc+Gyro+Mag Sensor Data Fusion
- Lab 6: Cloud data logging on IBM Watson
- Lab 7: Bonus – Voice over BLE

# Agenda

2

WiFi Network: **STMicro**  
Password: **STMTT19**

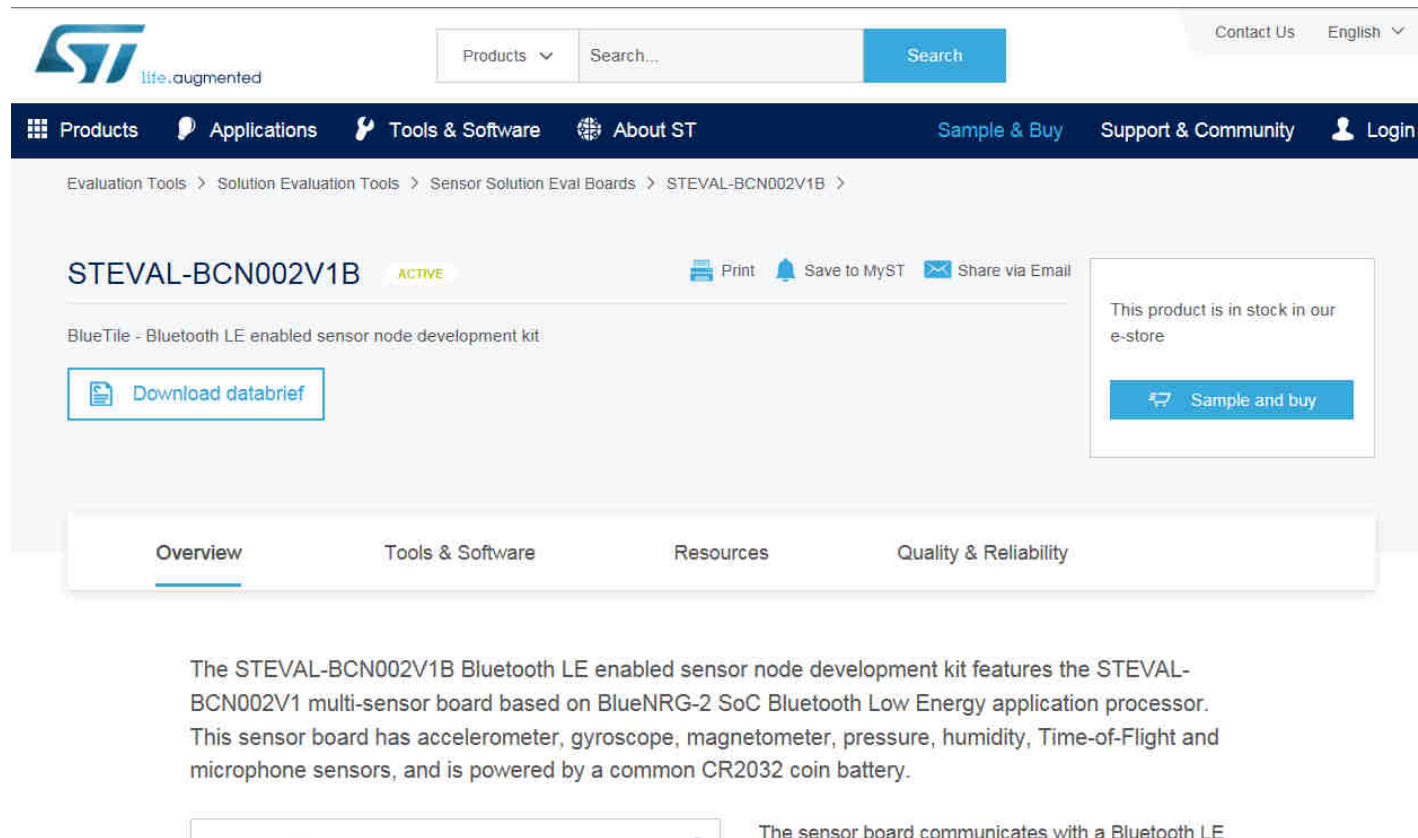
## 3



On St.com

4

<http://www.st.com/bluetile>



The screenshot displays the ST.com website interface for the STEVAL-BCN002V1B product. At the top, the ST logo with the tagline "life.augmented" is visible on the left, and navigation links for "Products", "Applications", "Tools & Software", and "About ST" are in the center. On the right, there are links for "Sample & Buy", "Support & Community", and "Login". A search bar with a "Search" button is also present. Below the navigation bar, a breadcrumb trail reads: "Evaluation Tools > Solution Evaluation Tools > Sensor Solution Eval Boards > STEVAL-BCN002V1B". The main product title "STEVAL-BCN002V1B" is followed by a yellow "ACTIVE" badge. To the right of the title are icons for "Print", "Save to MyST", and "Share via Email". Below the title, the description "BlueTile - Bluetooth LE enabled sensor node development kit" is shown, along with a "Download databrief" button. A callout box on the right states "This product is in stock in our e-store" and includes a "Sample and buy" button. A horizontal tab bar at the bottom of the product section contains "Overview" (selected), "Tools & Software", "Resources", and "Quality & Reliability". The main content area describes the product: "The STEVAL-BCN002V1B Bluetooth LE enabled sensor node development kit features the STEVAL-BCN002V1 multi-sensor board based on BlueNRG-2 SoC Bluetooth Low Energy application processor. This sensor board has accelerometer, gyroscope, magnetometer, pressure, humidity, Time-of-Flight and microphone sensors, and is powered by a common CR2032 coin battery." Below this text is a partially visible image of the sensor board and the text "The sensor board communicates with a Bluetooth LE".

ST life.augmented

Products Applications Tools & Software About ST Sample & Buy Support & Community Login

Evaluation Tools > Solution Evaluation Tools > Sensor Solution Eval Boards > STEVAL-BCN002V1B

STEVAL-BCN002V1B ACTIVE

Print Save to MyST Share via Email

BlueTile - Bluetooth LE enabled sensor node development kit

Download databrief

This product is in stock in our e-store

Sample and buy

Overview Tools & Software Resources Quality & Reliability

The STEVAL-BCN002V1B Bluetooth LE enabled sensor node development kit features the STEVAL-BCN002V1 multi-sensor board based on BlueNRG-2 SoC Bluetooth Low Energy application processor. This sensor board has accelerometer, gyroscope, magnetometer, pressure, humidity, Time-of-Flight and microphone sensors, and is powered by a common CR2032 coin battery.

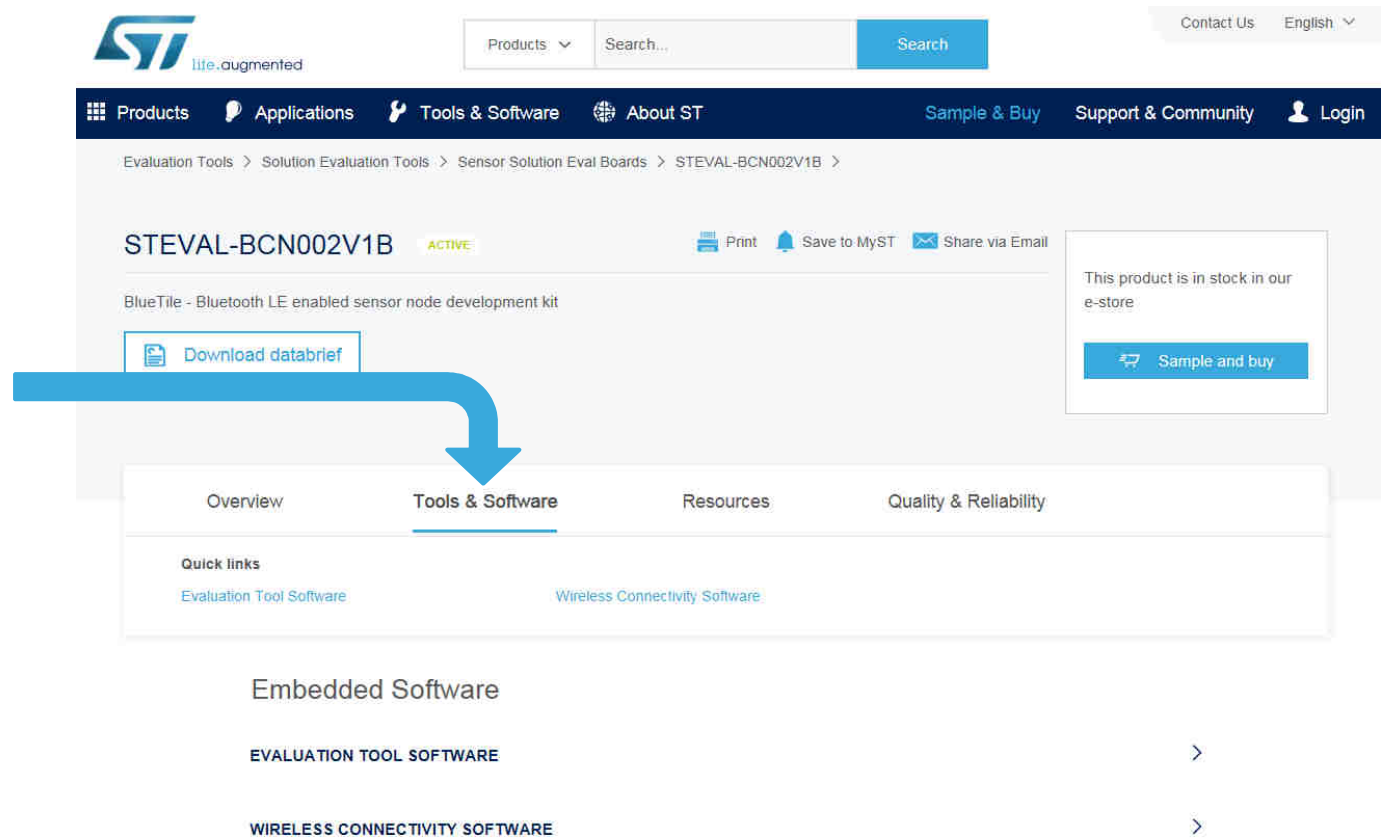
The sensor board communicates with a Bluetooth LE

# BlueTile SDK

5

<http://www.st.com/bluetile>

## Software Development Kit



The screenshot shows the STMicroelectronics website interface. At the top, there is a navigation bar with the ST logo and 'life.augmented' tagline. Below this is a search bar and a navigation menu with links: Products, Applications, Tools & Software, About ST, Sample & Buy, Support & Community, and Login. The main content area displays the product page for STEVAL-BCN002V1B, which is marked as 'ACTIVE'. The product description is 'BlueTile - Bluetooth LE enabled sensor node development kit'. A blue arrow points from the 'Download databrief' button to the 'Tools & Software' tab. Below the tabs, there is a 'Quick links' section with links to 'Evaluation Tool Software' and 'Wireless Connectivity Software'. At the bottom, there is a list of software categories: 'Embedded Software', 'EVALUATION TOOL SOFTWARE', and 'WIRELESS CONNECTIVITY SOFTWARE', each with a right-pointing arrow.

ST life.augmented

Products Applications Tools & Software About ST Sample & Buy Support & Community Login

Evaluation Tools > Solution Evaluation Tools > Sensor Solution Eval Boards > STEVAL-BCN002V1B >

STEVAL-BCN002V1B **ACTIVE**

Print Save to MyST Share via Email

BlueTile - Bluetooth LE enabled sensor node development kit

Download databrief

This product is in stock in our e-store

Sample and buy

Overview **Tools & Software** Resources Quality & Reliability

Quick links

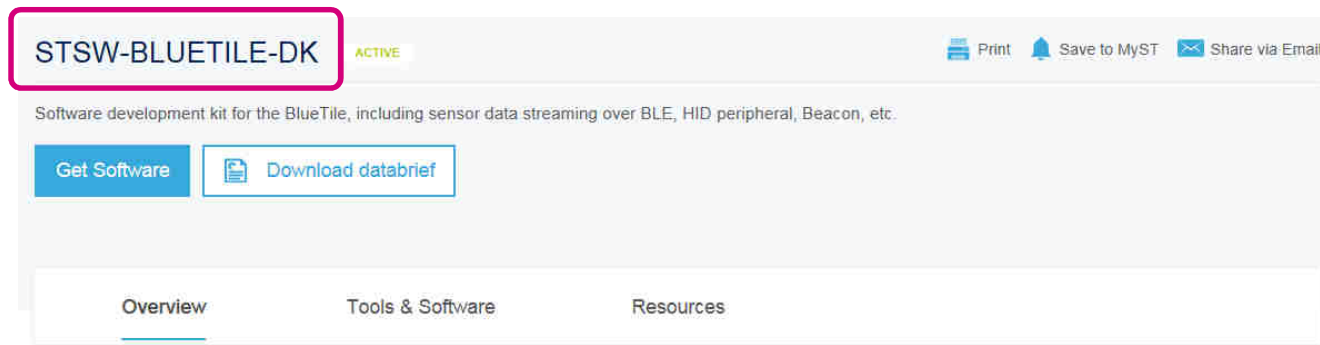
Evaluation Tool Software Wireless Connectivity Software

Embedded Software

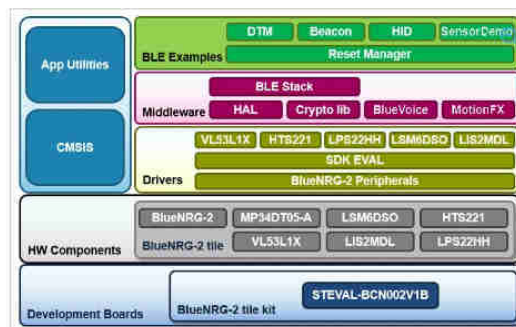
EVALUATION TOOL SOFTWARE >

WIRELESS CONNECTIVITY SOFTWARE >

- <https://www.st.com/en/embedded-software/stsw-bluetile-dk.html>



The STSW-BLUTILE-DK is based on STSW-BLUENRG1-DK evaluation SW package.



The STSW-BLUENRG1-DK package supports the BlueNRG-1 and BlueNRG-2 Bluetooth Low-Energy system-on-chip. This package includes a Wizard PC application to automatically generate the configuration header file needed for the BLE radio initialization. This package also includes BlueNRG-1 and BlueNRG-2 peripheral drivers and related examples, the BLE firmware stack together with the HAL (Hardware Abstraction Layer) and CryptoLib (Cryptographic Library).

The STSW-BLUTILE-DK supports the BlueTile platform. This package extends STSW-BLUENRG1-DK.

On St.com

7

<http://www.st.com/bluetile>

Schematic  
Gerbers  
BOM

The screenshot shows the ST.com website interface. At the top, there is a navigation bar with the ST logo, a search bar, and links for 'Products', 'Applications', 'Tools & Software', 'About ST', 'Sample & Buy', 'Support & Community', and 'Login'. Below the navigation bar, the breadcrumb trail reads: 'Evaluation Tools > Solution Evaluation Tools > Sensor Solution Eval Boards > STEVAL-BCN002V1B >'. The product name 'STEVAL-BCN002V1B' is displayed with a status of 'ACTIVE'. To the right of the product name are links for 'Print', 'Save to MyST', and 'Share via Email'. Below the product name is a 'Download databrief' button. A blue arrow points from the 'Schematic Gerbers BOM' text to the 'Resources' tab in the 'Quick links' section. The 'Resources' tab contains links for 'Product Specifications', 'Presentations', 'Board Manufacturing Specifications', 'Bill of Materials', 'Schematics', and 'Brochures'. A 'Sample and buy' button is also visible on the right side of the page.

# Companion ST BLE Sensor App

8



ON YOUR PHONE

- Look for “**ST BLE Sensor**” on the App Store or Google Play





# Where to Get It

9

## Sample & Buy

Part Number ▲	Marketing Status ▾	Budgetary Price (US\$) * ▾	Quantity ▾	Core Product ▾	ECCN (US) ▾	Country of Origin ▾	Order from Distributors ▾	Order from ST ▾
STEVAL- BCN002V1B	ACTIVE	50.0	1	BlueNRG-2, BALF-NRG- 02D3, LSM6DSO, LIS2MDL, VL53L1X, MP34DT05TR-A, LPS22HH, HTS221	5A992C	ITALY	 Check Availability	 Buy direct

# Where to Get It 10

[Contact Mouser \(USA\) \(800\) 346-6873](#) | [Feedback](#)Change Location English US

[Products](#) [Manufacturers](#) [Technical Resources](#) [Services & Tools](#) [Help](#) [Order History](#) [Log In](#) [Register](#)

All

☐ In Stock ☐ RoHS

[All Products](#) > [Embedded Solutions](#) > [Engineering Tools](#) > [Sensor Development Tools](#) > [Multiple Function Sensor Development Tools](#) > [STMicroelectronics STEVAL-BCN002V1B](#) [See an Error?](#)

**STEVAL-BCN002V1B**

[Enlarge](#)  
Images are for reference only  
See Product Specifications  
[Share](#)

<b>Mouser #:</b>	511-STEVAL-BCN002V1B
<b>Mfr. #:</b>	STEVAL-BCN002V1B
<b>Mfr.:</b>	<a href="#">STMicroelectronics</a>
<b>Customer #:</b>	<input type="text" value="Customer #"/>
<b>Description:</b>	Multiple Function Sensor Development Tools BlueNRG-Tile - Bluetooth LE enabled sensor node development kit
<b>Lifecycle:</b>	<b>New Product:</b> New from this manufacturer.
<b>Datasheet:</b>	<a href="#">STEVAL-BCN002V1B Datasheet</a>
<b>More Information:</b>	<a href="#">Learn more about STMicroelectronics STEVAL-BCN002V1B</a>
<b>Shipping Alert:</b>	This product may require additional documentation to export from the United States.

**In Stock: 49**

<b>Stock:</b>	49 Can Ship Immediately
<b>On Order:</b>	30 <a href="#">View Delivery Dates</a>
<b>Factory Lead-Time:</b>	11 Weeks
<b>Enter Quantity:</b>	Minimum: 1 Multiples: 1 <input type="text"/> 

**Pricing (USD)**

Qty.	Unit Price	Ext. Price
1	\$50.00	\$50.00


**NEWEST PRODUCTS**  
**STMICROELECTRONICS**

**STEVAL-BCN002V1B BlueTile Development Kit**

# Online Support

11

<https://my.st.com/ols>



Products ▾ Search... Search

Contact Us English ▾

Products Applications Tools & Software About ST

Sample & Buy Support & Community Login

Welcome back!

Enter your e-mail address and password to login your myST user.

E-mail address

Password

☐ Remember me on this computer. ⓘ

Login

[Forgot password?](#)

New user?

myST brings you a set of personalized features:

- Participate to ST Events
- Stay informed with ST eNewsletters
- Get help with ST Online Support
- Discuss on the ST Community
- Benefit from our Online Design Tools
- Download Software
- Order free samples
- Manage your weekly product updates
- Buy ST Products & Tools

Create Account

Support

# Online Support

12



Service Portal **New Request** RAF RIVA Logout

Case Id	Subject	Status	Creation Date	Last Update On
---------	---------	--------	---------------	----------------

No Cases found. Click on 'New Request' to create one.

Insert the Part Number

**STEVAL-BCN002V1B**



Service Portal New Request RAF RIVA Logout

## Case Details

<b>Urgency*</b> Medium	<b>Project Name</b> Project Name
<b>Category*</b>	<b>Production Forecast (Qty)</b>
<b>Notify To</b> Please specify email addresses in the 'Description' or 'Comments' boxes to add people in copy of this request. This will be added by the Support Agent.	

## Case Description

<b>Product Category or ST Part Number</b> Enter atleast 3 characters.
<b>Subject*</b> Subject
<b>Description*</b> Type something



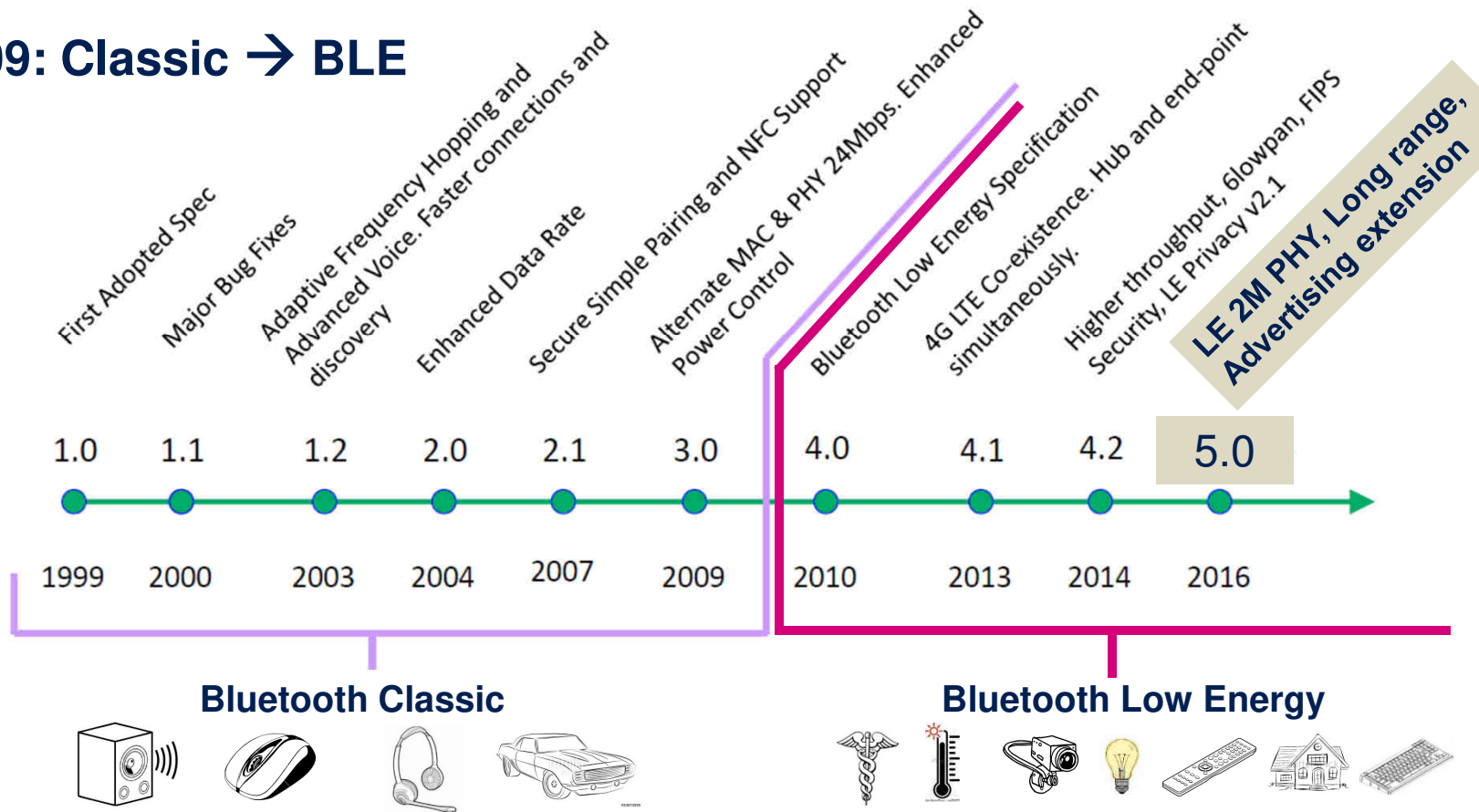


# Introduction to Bluetooth Low Energy

# Bluetooth® Evolution

14

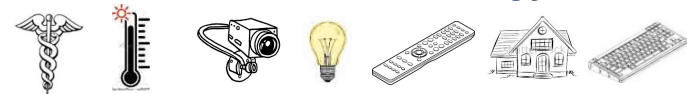
- 20 Years in the market!
- 2009: Classic → BLE



## Bluetooth Classic



## Bluetooth Low Energy



# Bluetooth Low Energy (LE)

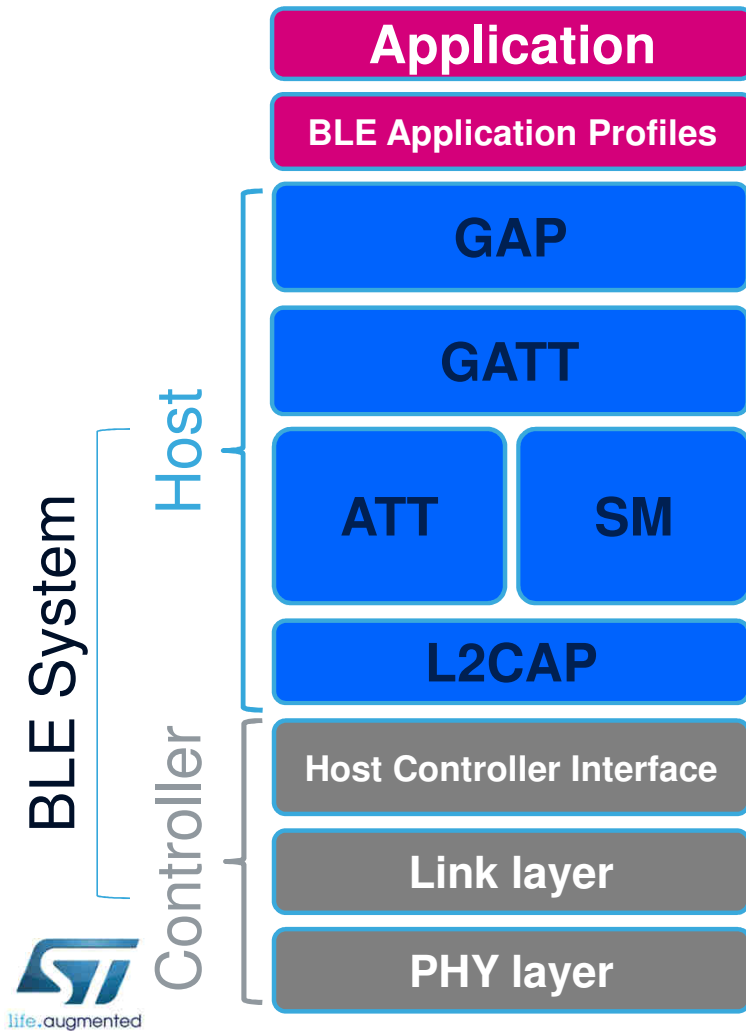
## Designed for Success

15

- Lower power than Bluetooth “Classic” (or Bluetooth 3.0)
  - Lower duty cycle
  - Faster connection
  - Lower peak power (relaxed RF parameters)
- Compatible with all major platforms (iOS, Android, Windows, Linux)
- Multiple network topologies:
  - Point to point: single master connects to single slave (and each slave can connect to 2 master)
  - Star: Multiple slaves connected to a single master
  - Mesh: introduced in the 2017
- State of the Art encryption, security including privacy/authentication

# Protocol Stack

16



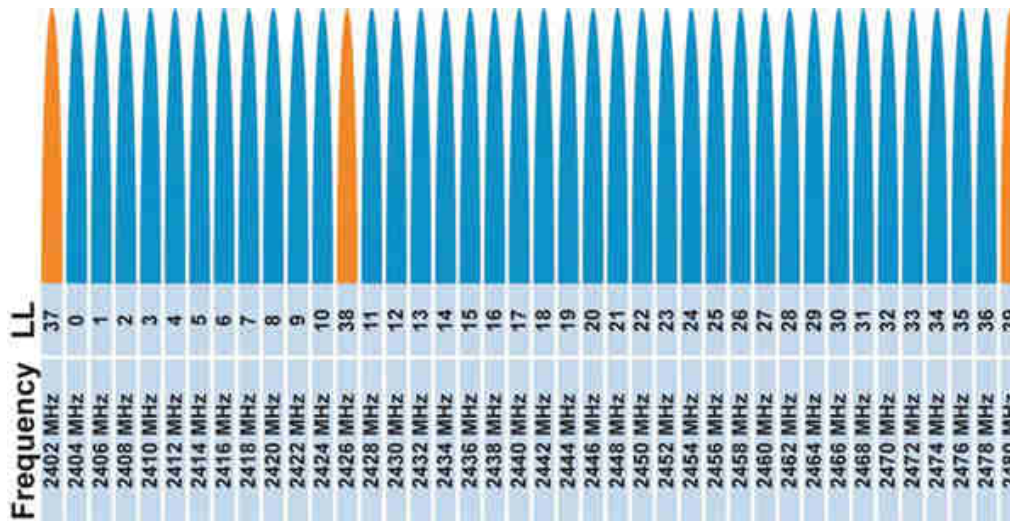
- **Generic Access Profile**
  - How a device can discover and connect with one another
- **Generic Attribute Profile**
  - How services, characteristics can be discovered and then used
- **Attribute Protocol**
  - Protocol for discovering, reading, and writing attributes on a peer
- **Security Manager**
  - Handles the secure communication
- **L2CAP**
  - Protocol multiplexer. Segmentation and reassembly of packets
- **HCI**
  - Interface between Host and Controller
- **Link Layer**
  - Handles packets, channels, advertising, scanning & connections
- **Physical Layer**
  - Transmits/receives bits



# Protocol Stack: PHY

17

- A BLE Radio is a 2.4 GHz ISM Band Transceiver
- 40 RF Channels, 2 MHz Channel Spacing. Two types of channels:
  - Advertising channels (3 - orange) for Advertising Packets. Used for Discoverability and for Broadcasting/Observing
  - Data Channels (37 - blue) for Data Packets. Used to send application data in Connection



Source : Bluetooth® SIG

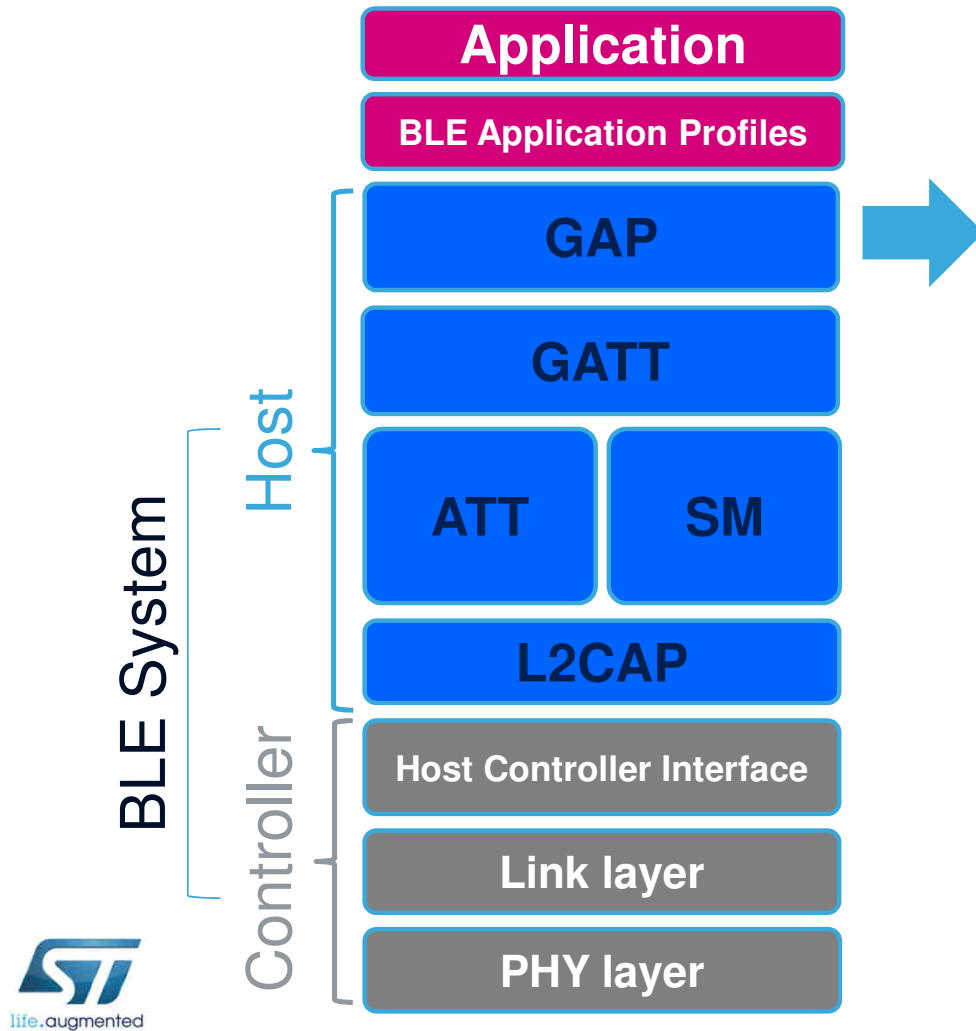
## • GFSK Modulation

- $BT = 0.5$
- Modulation Index = 0.5
- “pulse shaping” Gaussian filter “smooths” transitions from zero to one reduces spectral width

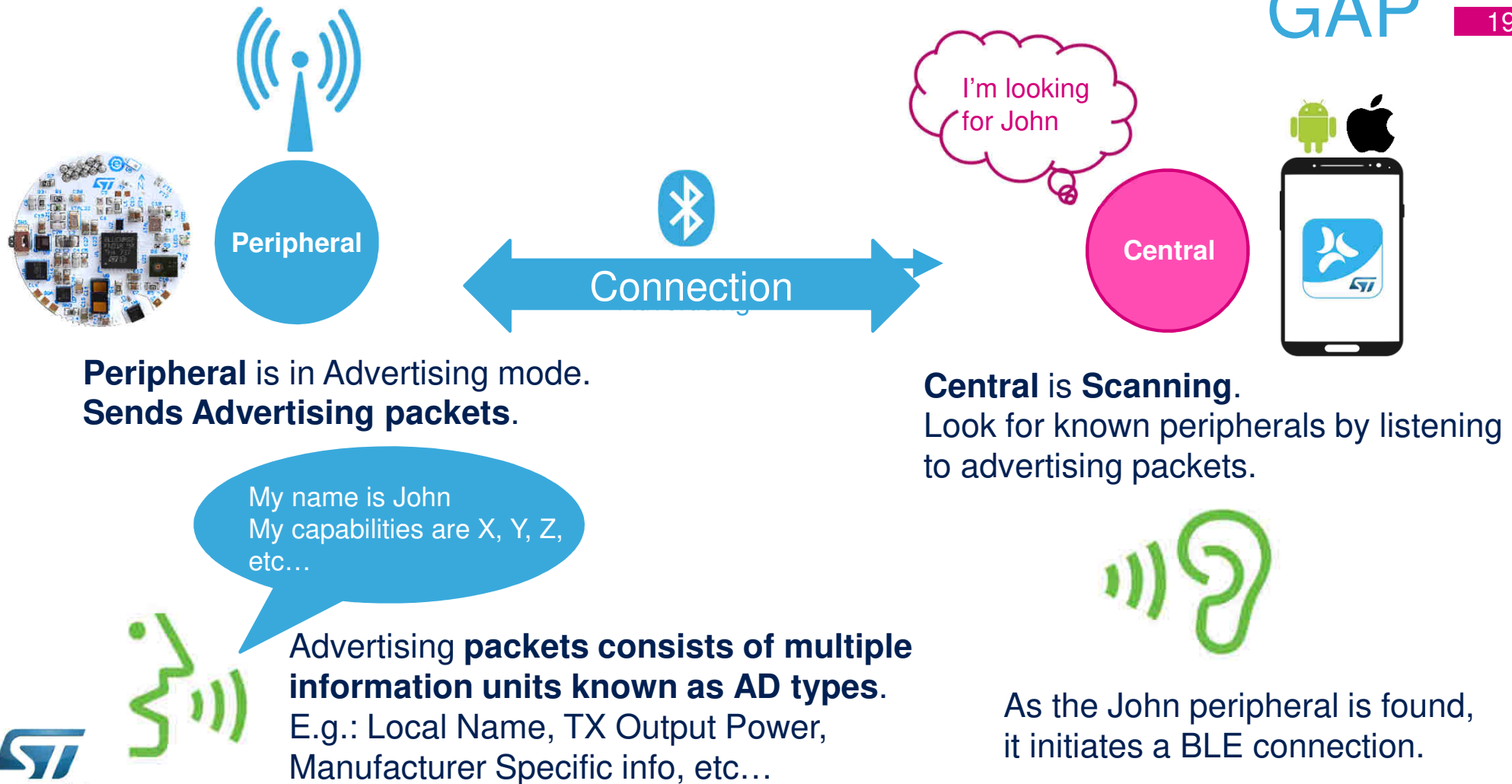


# Protocol Stack - GAP

18

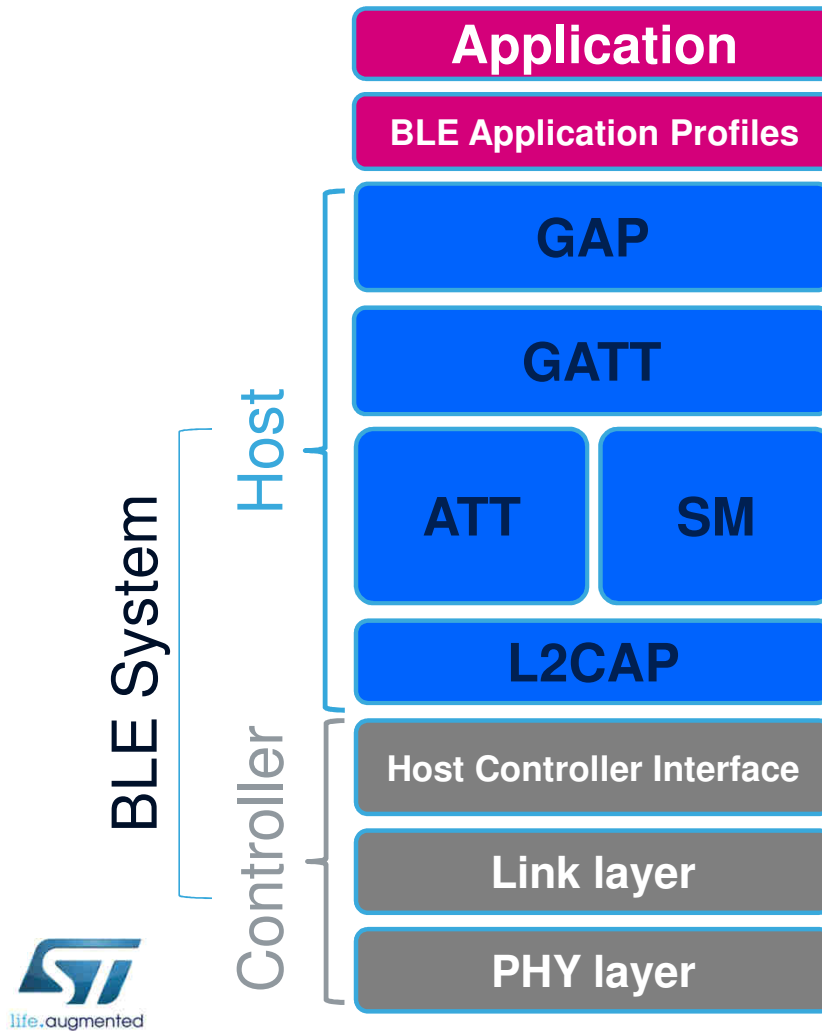


- Who controls the radio network?
  - **GAP (Generic Access Profile)**
  - Defines 4 roles
    - **Broadcaster**
    - **Observer**
    - **Peripheral**
    - **Central**

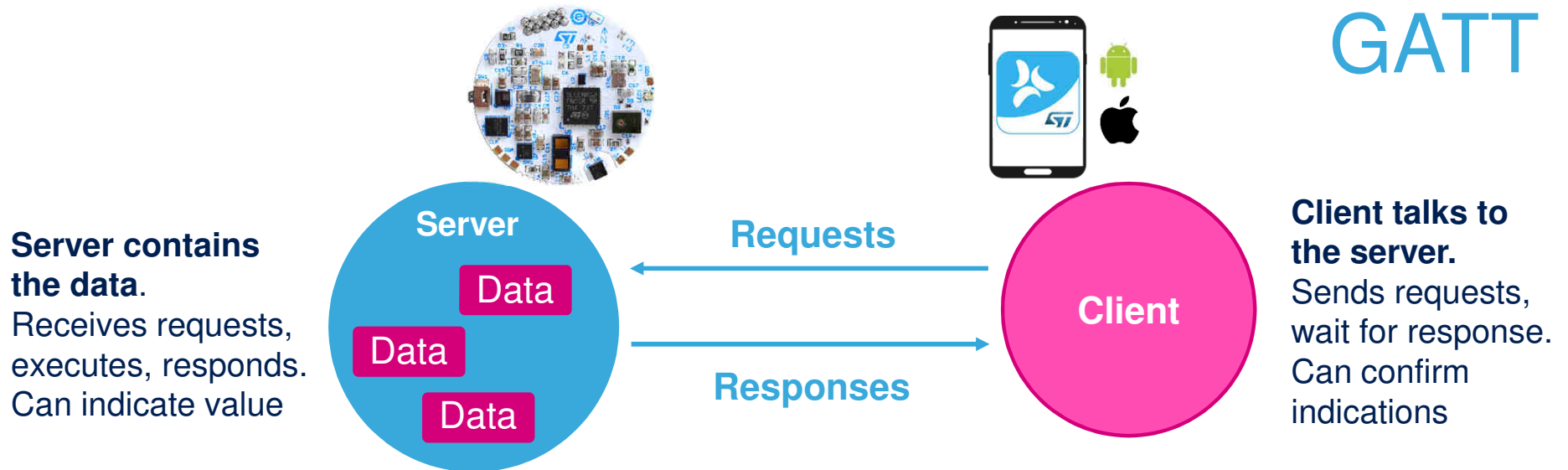


# Protocol Stack - GATT

20



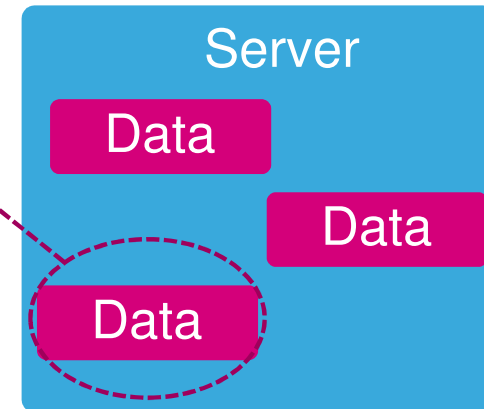
- Who controls the data flow?
  - **GATT (Generic Attribute Profile)**
  - Defines 2 roles
    - **Server**
    - **Client**



Protocol PDU Type	Sent by	Description
Request	Client	Client requests something from server (it always causes a response)
Response	Server	Server sends response to a request from a client
Command	Client	Client commands something to server (no response)
Notification	Server	Server notifies client of new value (no confirmation)
Indication	Server	Server indicates to client new value (it always causes a confirmation)
Confirmation	Client	Confirmation to an indication

- **Data are organized** and exposed as **attributes**

Each **Data** element in the Server is called **Attribute**



- Each attribute has:

- A **handle**: it identifies the attribute on the server
- A **type** (defined by Universal Unique ID - **UUID**): what the value means
  - 16-bit UUID: pre-defined by Bluetooth SIG
  - 128-bit UUID: Vendor Specific identifiers
- A **value** (0 to 512 octets)

- Example

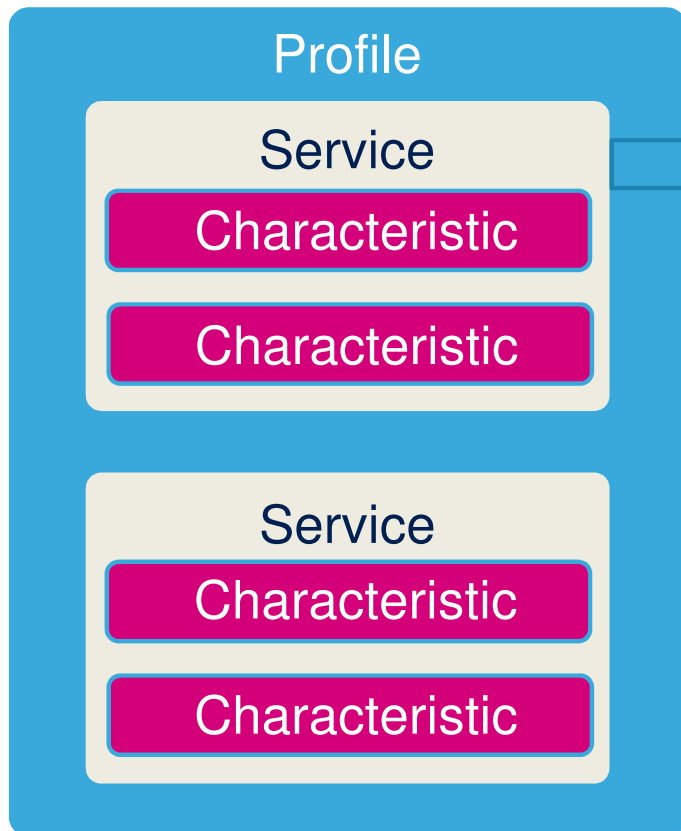


Handle	Type	Value
0x0009	«Device Name»	“Temperature Sensor”
0x0022	«Battery State»	0x04
0x0098	«Temperature»	0x0802

Source : Bluetooth® SIG

# GATT Profile

23



- A GATT Profile defines how attributes are organized and how the application can access them.
- Attributes are organized in Services and Characteristics

a typical example:

1 service: "ARG" (Angular Rate and Gravity)

2 characteristics: "Gyro", "Acc"

Values: [0,-1,+2], [-10,+15,+950]



# ST BLE Devices

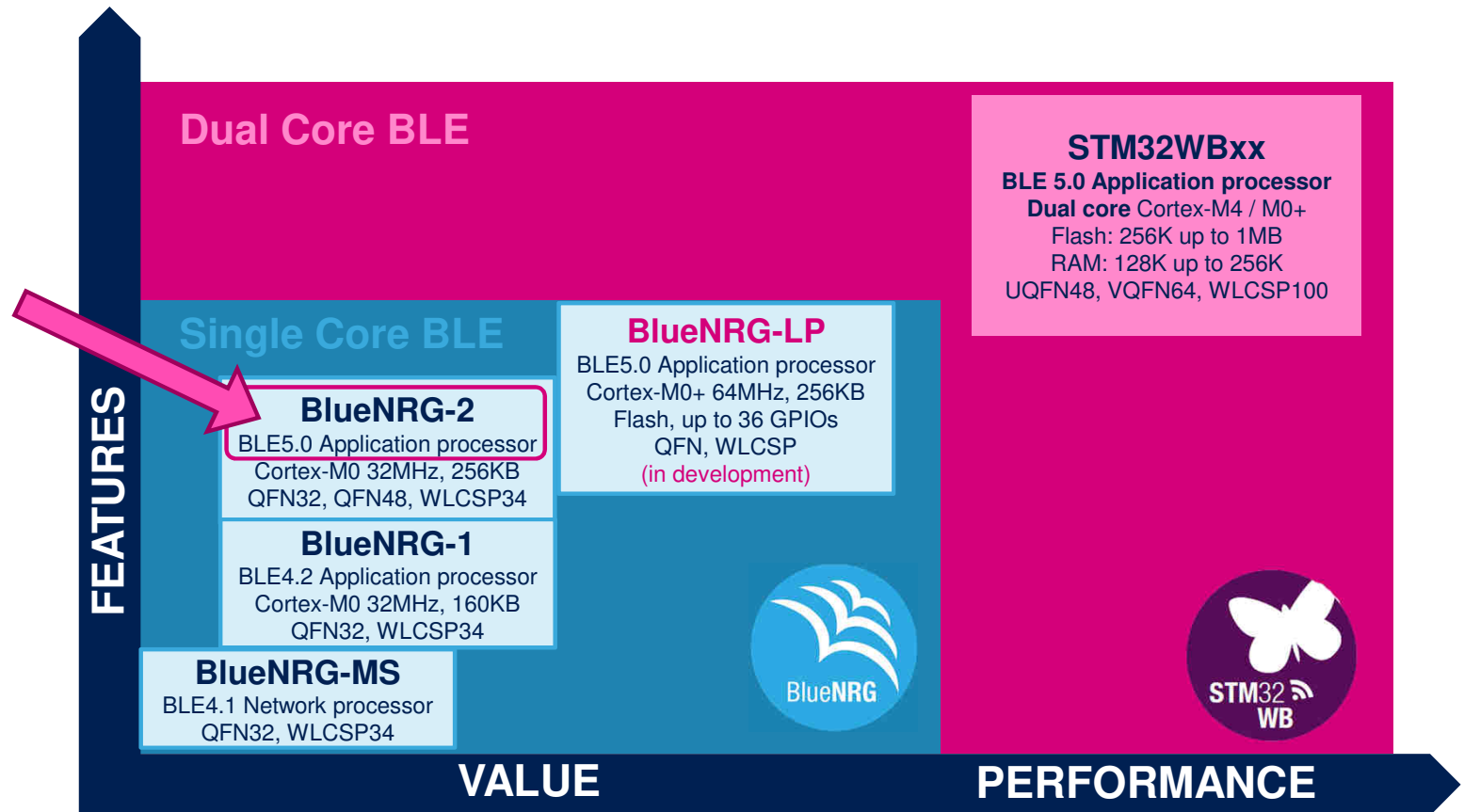









# ST BLE Portfolio

25

ST BlueTile



# BlueNRG-2 SoC at a Glance

<p><b>The lowest power consumption</b></p>  <p><b>&gt;3 years lifetime on CR2032<sup>(*)</sup></b> 59 <math>\mu</math>A/MHz 0.9 <math>\mu</math>A sleep</p>	<p><b>Processing power on demand</b></p>  <p><b>Low-power architecture, Cortex-M0 @ 32 MHz</b></p>	<p><b>Flexible memory architecture</b></p>  <p><b>256 KB eFLASH 24 KB ULL SRAM</b> (with full SRAM data retention)</p>	<p><b>Optimized BLE Radio stack</b></p>  <p><b>70 kB FLASH 8 kB RAM</b> 0.9<math>\mu</math>A with full RAM retention</p>	<p><b>Maximum security</b></p>  <p><b>ECC-256 AES-128</b> Factory UID Secure KEY</p>
--	---	--	---	---

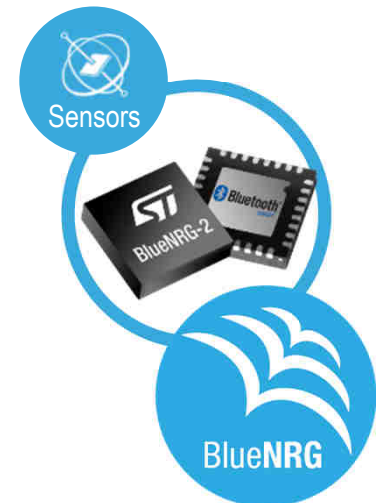
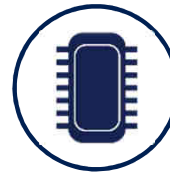
Seamless connection with SENSORS



Robust and Reliable BLE Link



Small form factor



(\*) Based on the average current consumption in connection mode (7.059  $\mu$ A, connection interval 1000 ms)



**Bluetooth 5.0 certified**



**Privacy 1.2 and secure connection 4.2**



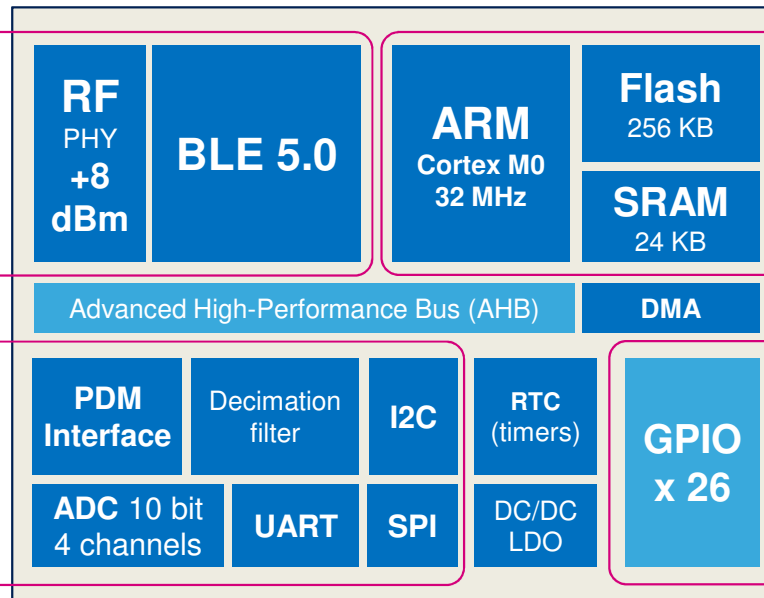
**Faster and more reliable data transfer**

# Bluetooth LE Programmable Processor

27



Bluetooth LE  
5.0  
ULP radio



PROGRAMMABLE  
MCU  
with eFLASH  
to host application code

Application and Profiles

BLE Protocol stack

PROGRAMMABLE  
PERIPHERALS

PROGRAMMABLE  
Digital I/O



STEVAL-IDB008V2

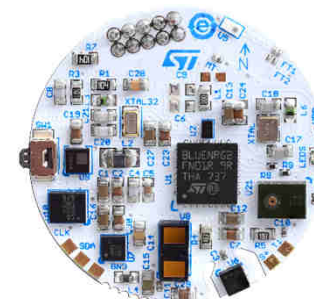


STEVAL-BCN002V1B



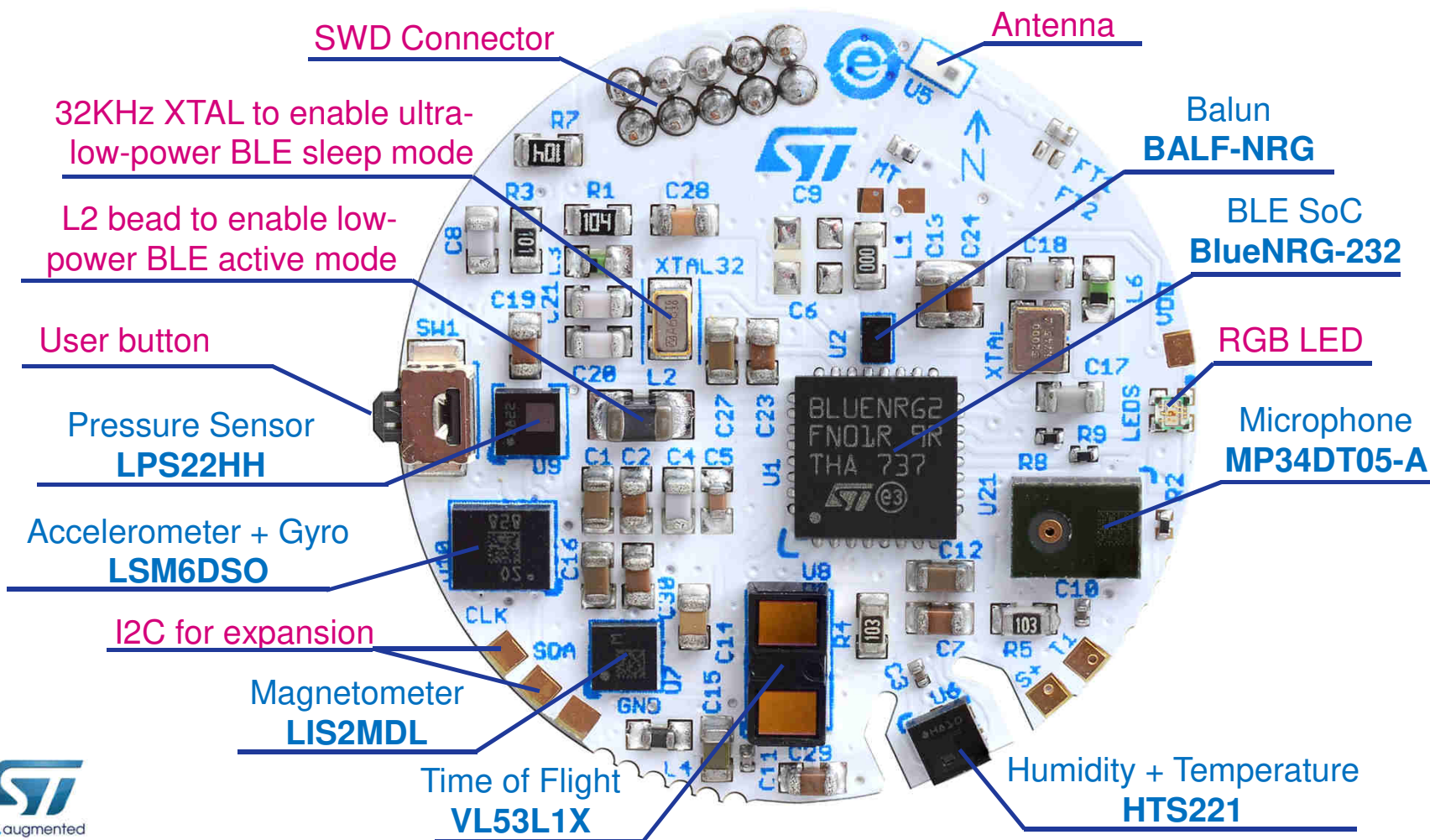


# BlueTile Development Kit

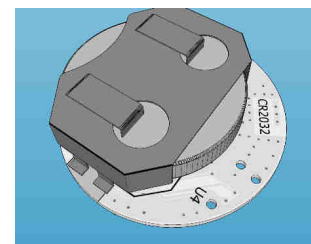


# STEVAL-BCN002V1

29



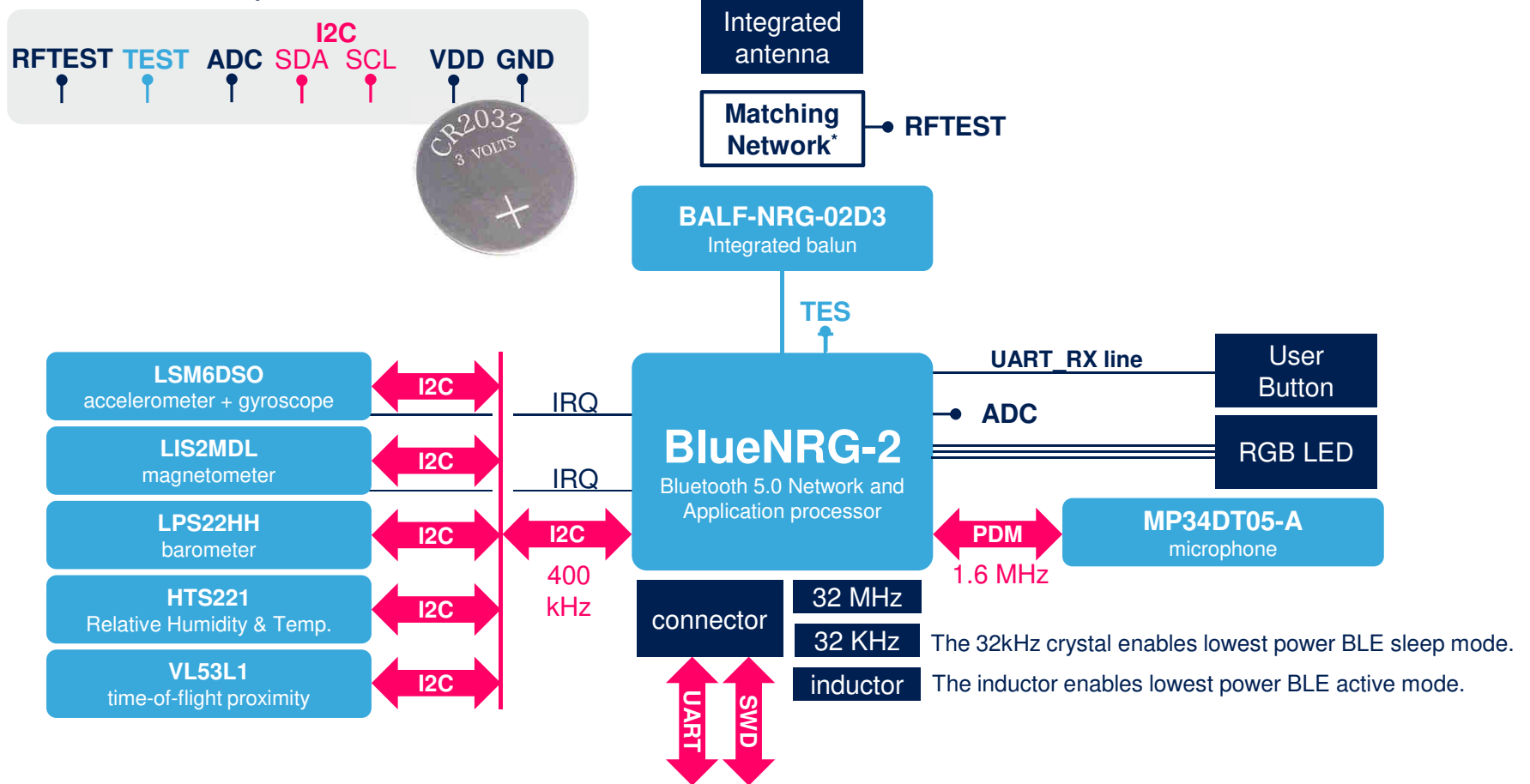
Bottom view



# STEVAL-BCN002V1 Block Diagram

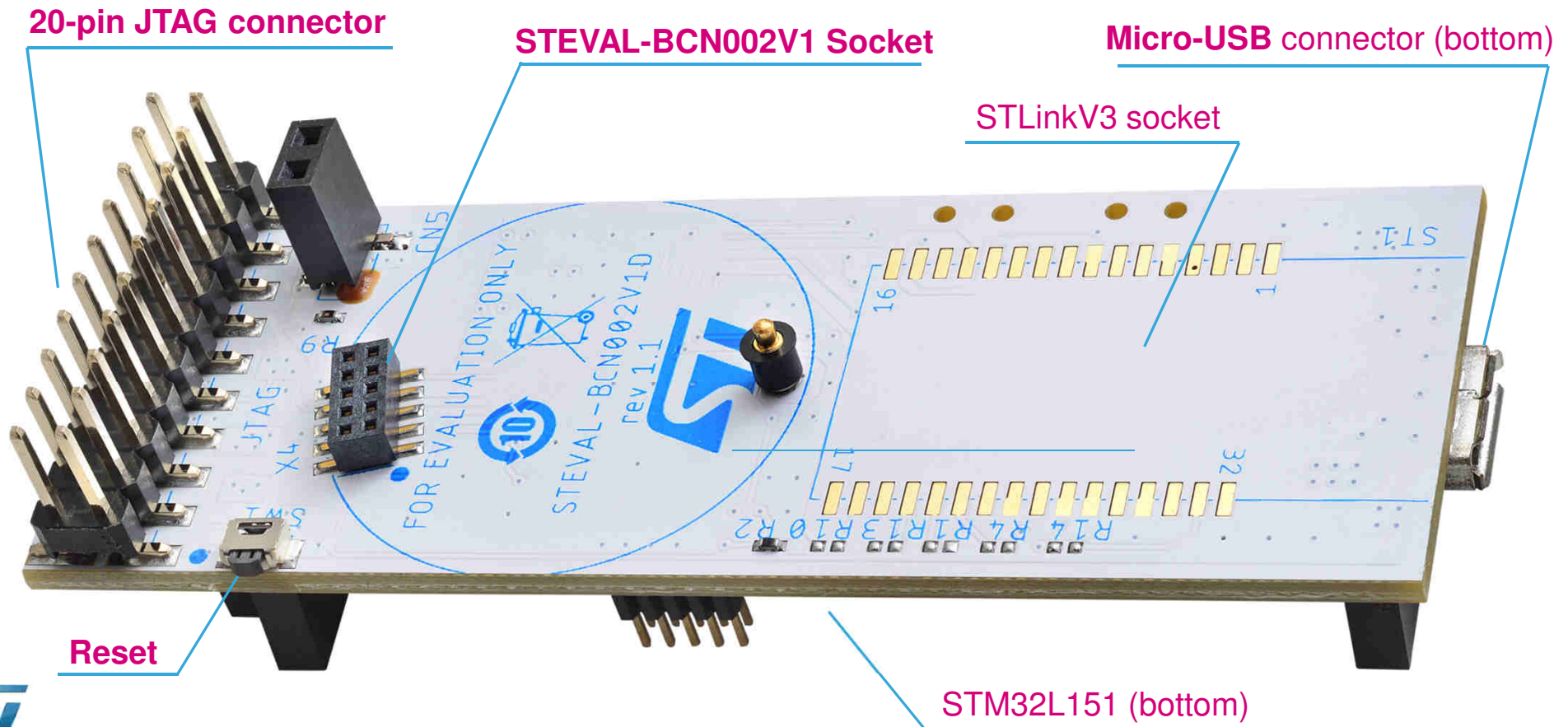
30

## Test points



# STEVAL-BCN002V1D

31

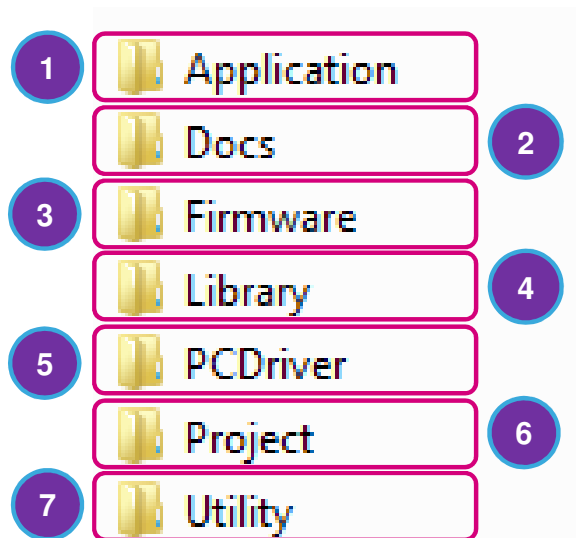




# BlueTile SDK Overview

32

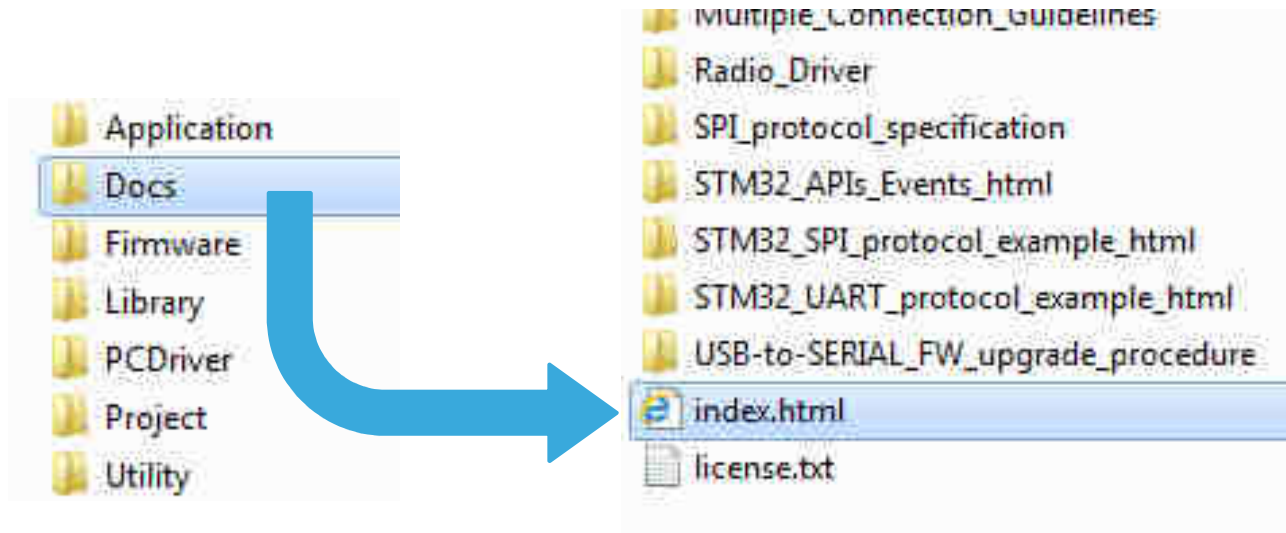
<https://www.st.com/en/embedded-software/stsw-bluetile-dk.html>



1. PC GUI Application to initialize the Radio params
2. Doxygen Documentation
3. Pre-built FW examples/images
4. Low level drivers and BLE stack/profile libraries
5. Virtual COM port drivers
6. **Reference examples in source code**
7. Utility section: IAR BlueNRG-2 Flasher



- Open the **Docs** folder



- Double click on **index.html**

- Device Datasheet
- BLE stack documentation
  - Release notes
  - APIs and events
  - Programming manual
- Device (and kits) documentation
  - PCB design guidelines
  - Bring up guide
  - Getting started



# Hands on Overview

# Labs Overview

35

- **Lab 1:** Getting started with STEVAL-BCN002V1 “Serial Terminal Test”
- **Lab 2:** Connecting to the ST BLE Sensor app
- **Lab 3:** LED characteristic
- **Lab 4:** Accelerometer embedded events detection
- **Lab 5:** 9-axis Acc+Gyro+Mag Sensor Data Fusion
- **Lab 6:** Cloud data logging on IBM Watson
- **Lab 7:** Voice over BLE

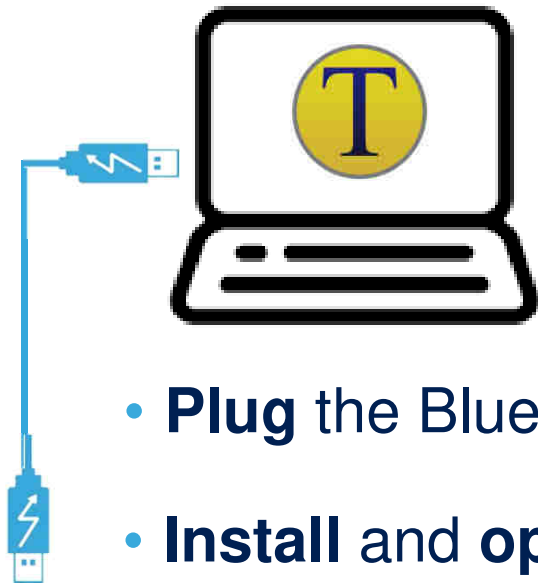


# Lab 1

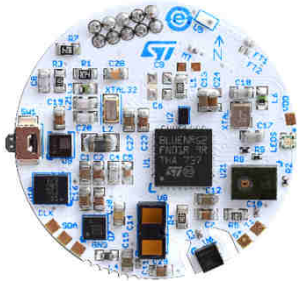
## Getting Started with STEVAL-BCN002V1B “Serial Terminal Test”

# “Serial Terminal Test”

37

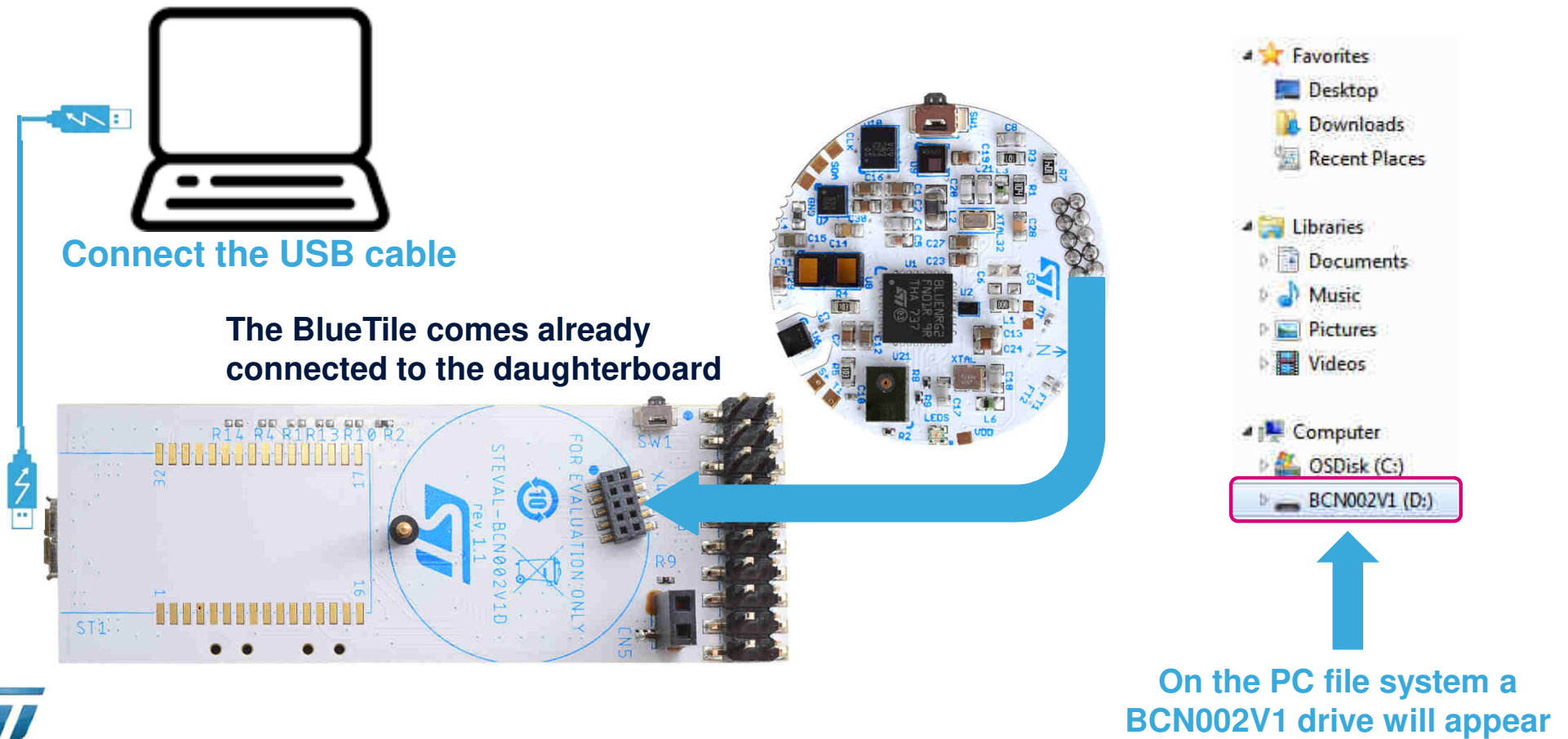


- **Plug** the BlueTile to the PC using the USB cable
- **Install** and **open Tera Term** and **configure** serial terminal



# Connect Your STEVAL-BCN002V1 B to the PC Using USB

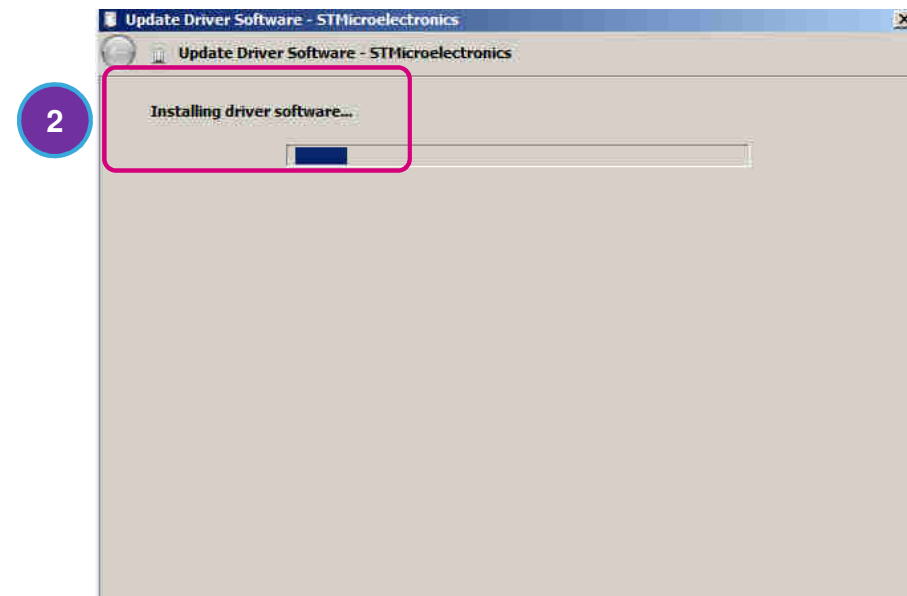
38



# Windows7: Allow the Driver Installation

39

1. Click on **Install driver software anyway**
2. Installation starts



# Virtual COM Port Driver

40

If you have issue with the **STMicroelectronics Virtual COM Port** device driver installation here the **instructions** for installing the Virtual COM port driver:

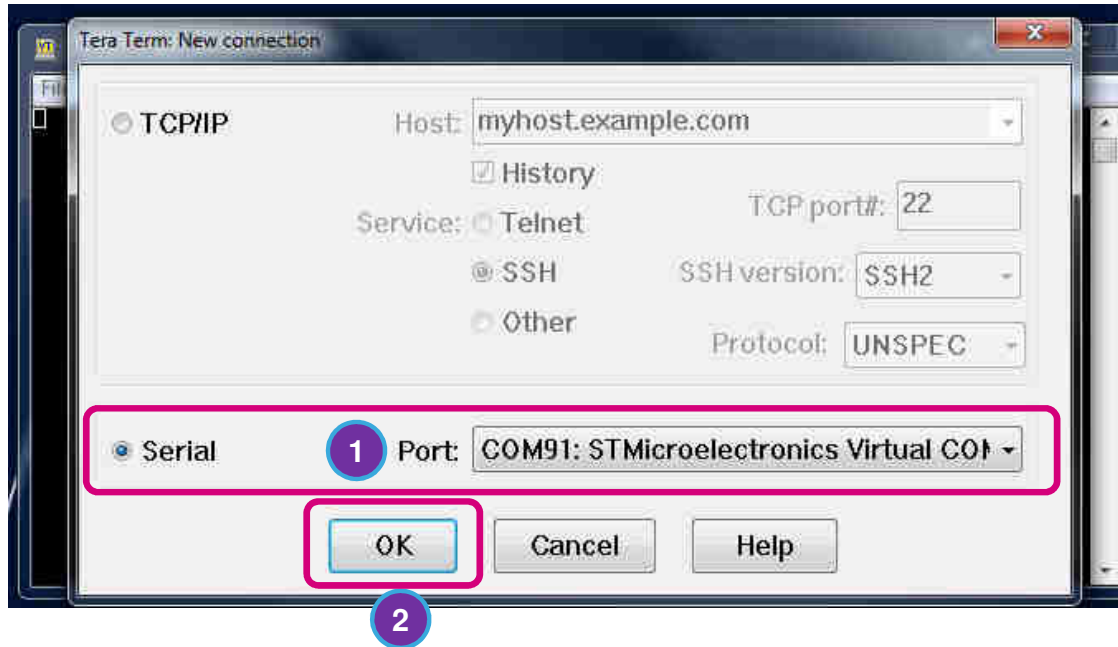
**Win7**

**Win10**



# Tera Term Configuration 1/4

41



## 1. Select the **STMicroelectronics Virtual COM Port**

- **NOTE:** on **Win10 PC** the serial port is labeled just as “**COMxx**”

## 2. Click **OK**

# Tera Term Configuration 2/4

42

1. In **Setup** -> **Serial port...**

**Set the following:**

**Baud rate : 115200**

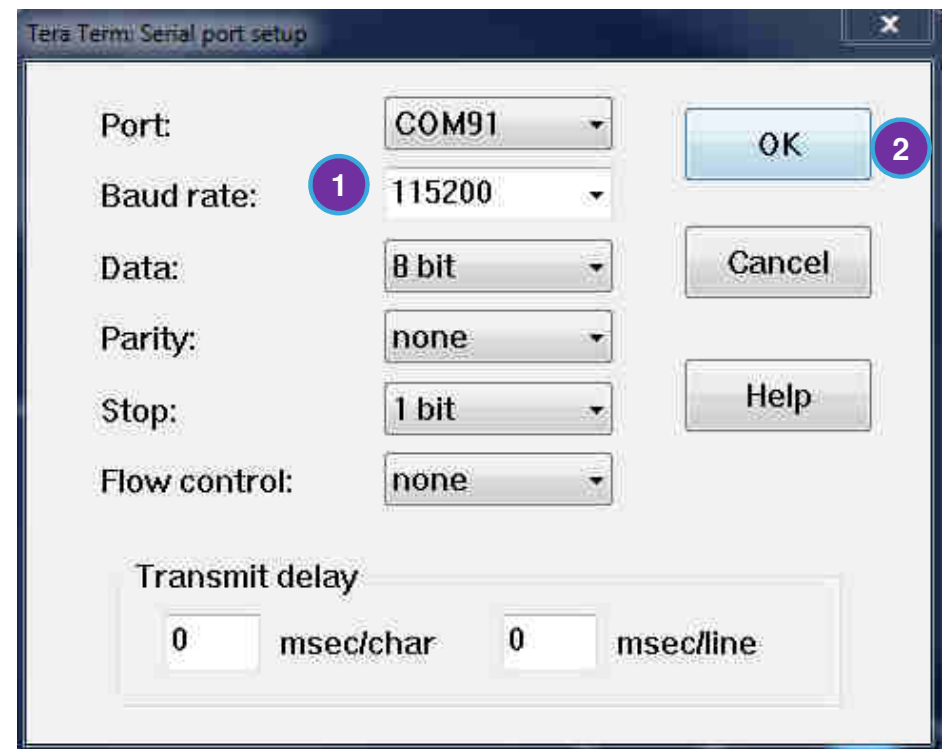
**Data : 8 bit**

**Parity : none**

**Stop : 1 bit**

**Flow control : none**

2. Click **OK**

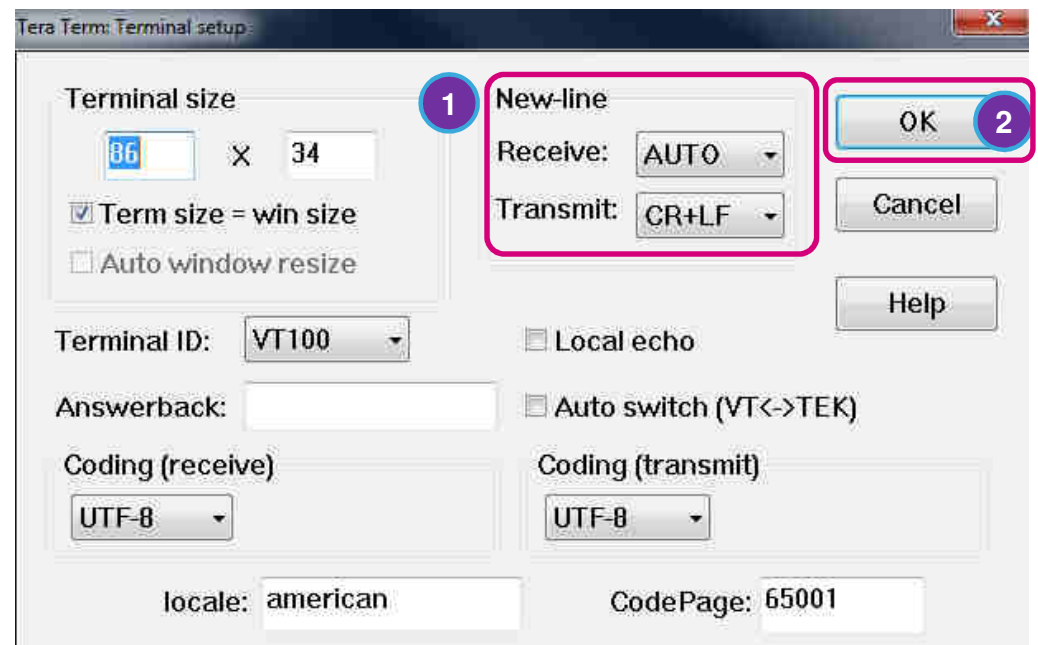


# Tera Term Configuration 3/4

43

1. In **Setup** -> **Terminal...** set  
**New-line Receive:**  
**AUTO**

2. Click **OK**



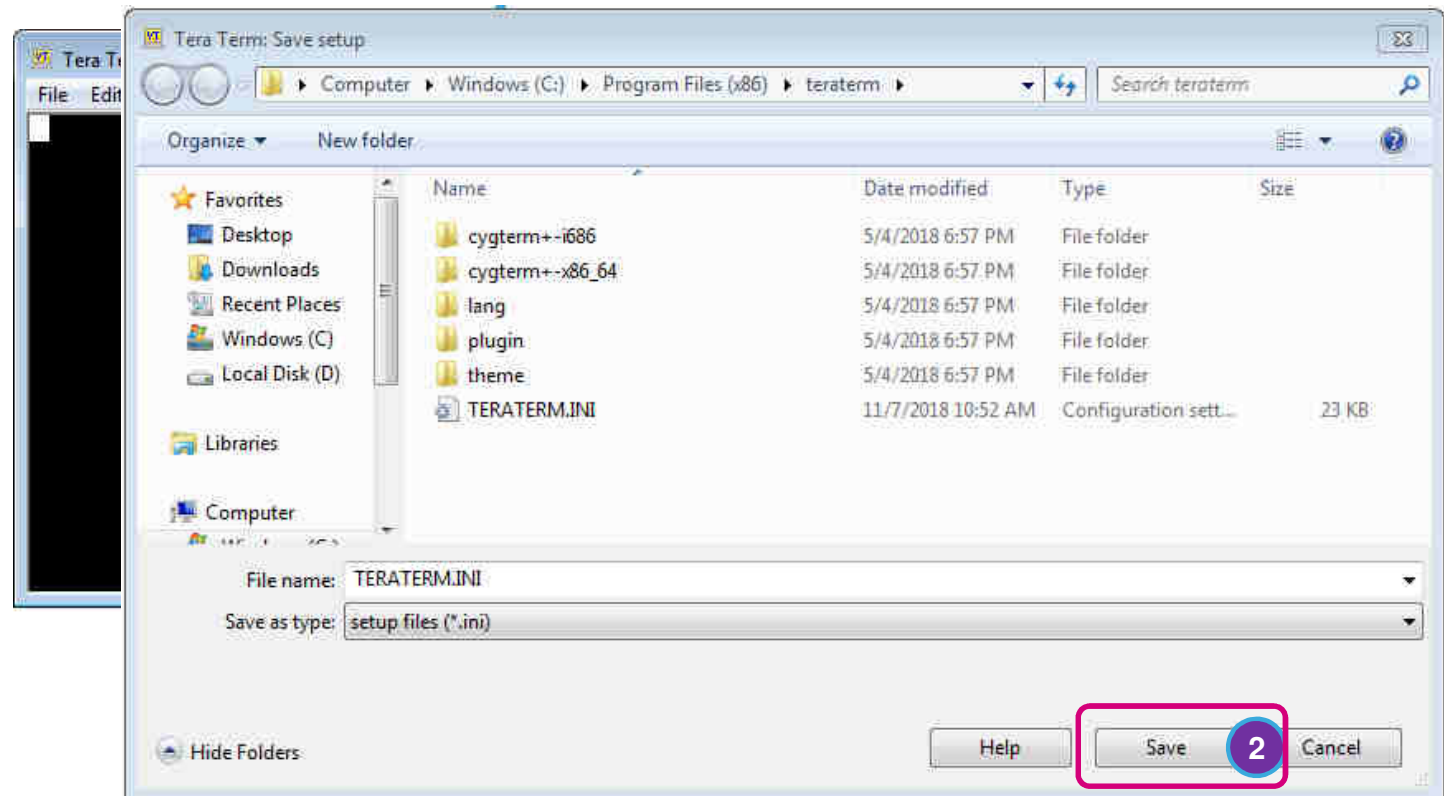
# Tera Term Configuration 4/4

44

1. Click **Setup->Save setup...**

2. Click **Save**

1





- If you see the output above, **Tera Term** is now properly configured

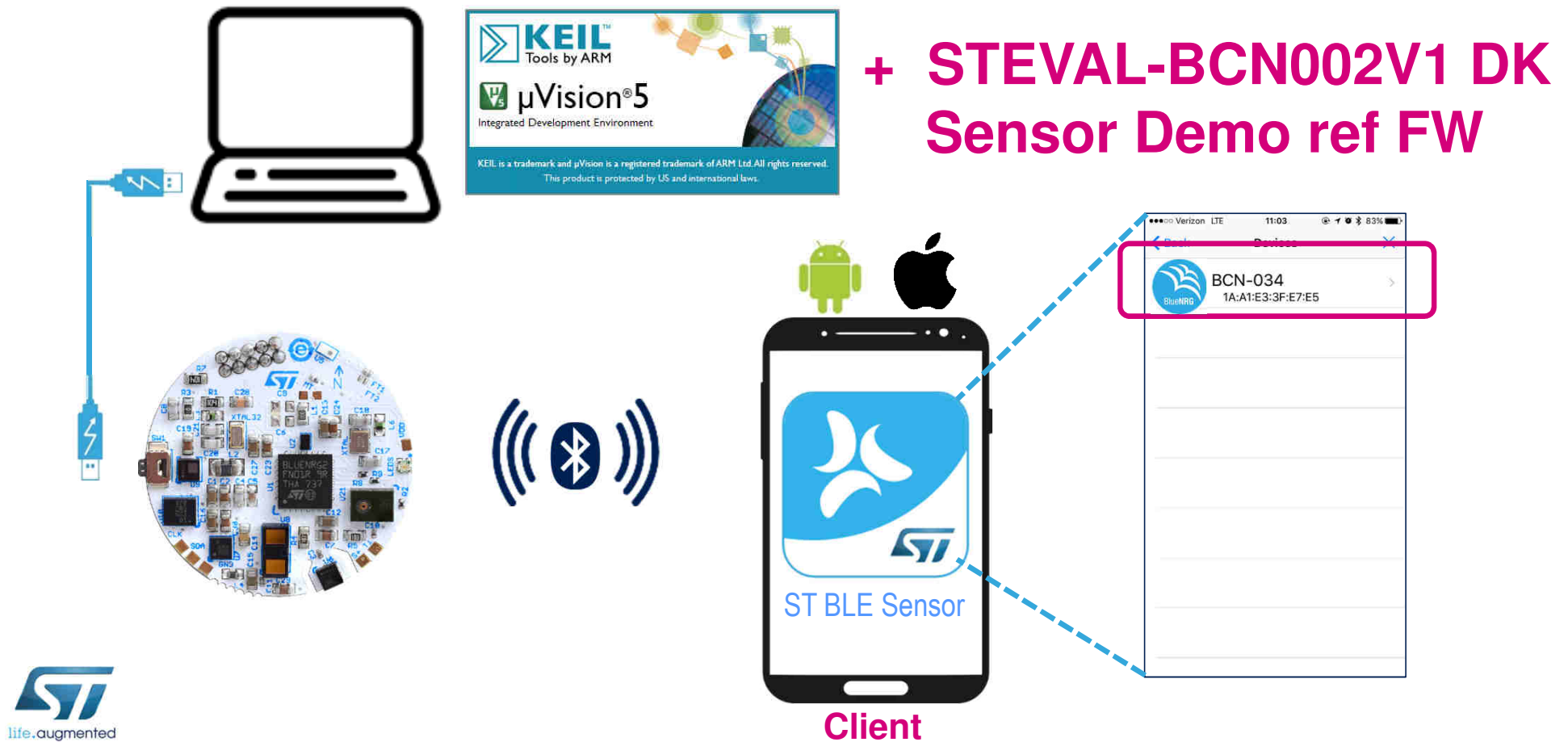


# Lab 2

Connect to the ST BLE Sensor App

# Customize YOUR STEVAL-BCN002V1

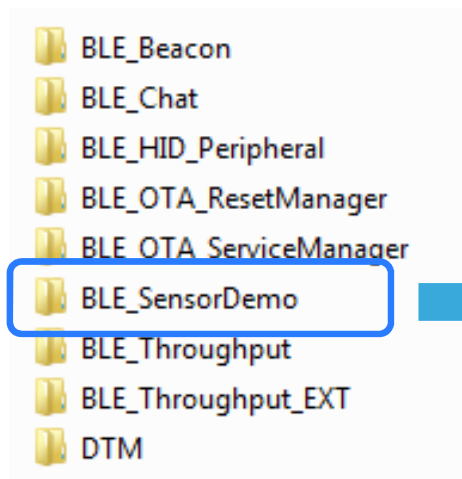
47



# BLE\_SensorDemo Application

48

1. In the BlueTile SDK browse the following path: STEVAL-BCN002V1 DK 1.1.0\Project\BLE\_Examples



**The BLE\_SensorDemo  
embedded application is the  
main reference**



# Integrated Development Environment

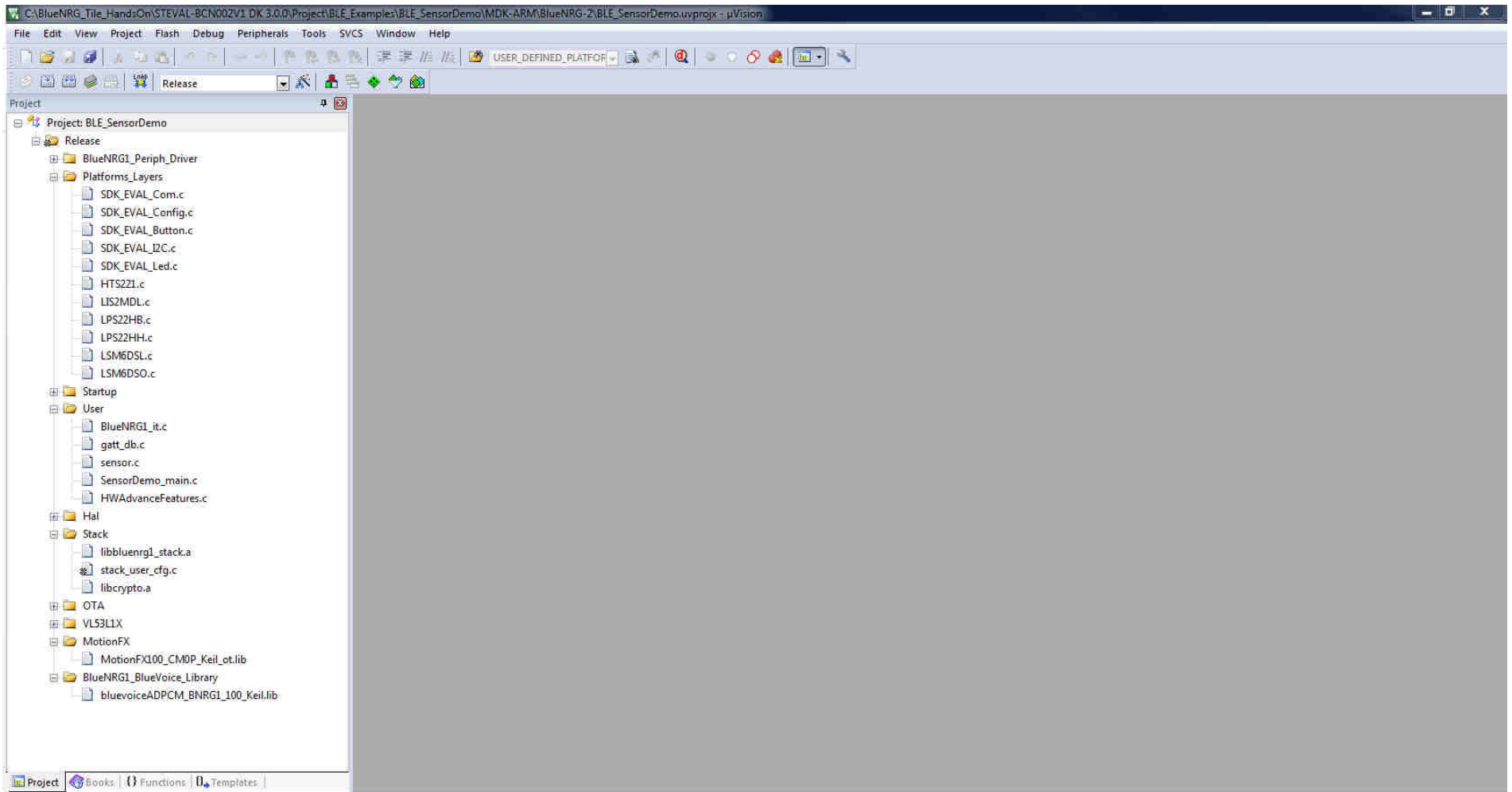
## Arm Keil MDK

49

- **MDK: Microcontroller Development Kit**
  - Complete software development environment from Arm for a wide range of Arm Cortex-M based microcontroller devices.
  - MDK includes the **µVision IDE**, **debugger**, and **Arm C/C++ compiler**
- A **FREE** license is available for ST BlueNRG-2
  - Go **HERE** to get the free license!

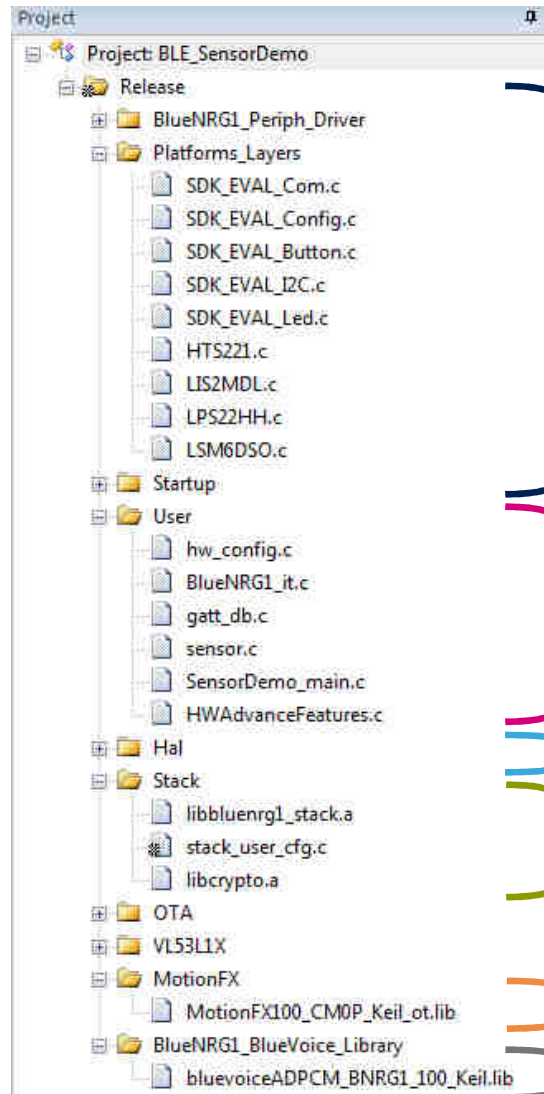
# BLE\_SensorDemo Application

50



# Application Structure

51



**HW peripherals drivers and platform layer files**

**Application source code**

- Main
- ATTRIBUTE Database
- Application

**Hardware Abstraction Layer and sleep management**

**BLE Stack library – provided in binary format**

**Sensor Data Fusion library**

**Voice over BLE library**

# A Look at the Main Application

52

```
int main(void) {
```

```
    SystemInit();
```

**Remap the vector table and configure all the interrupts priority**

```
    SdkEvalIdentification();
```

**Identifies STEVAL or custom PCB**

```
    PlatformInit();
```

**HW peripherals initialization**

```
    BlueNRG_Stack_Initialization(&BlueNRG_Stack_Init_params);
```

**BLE stack initialization**

```
    Sensor_DeviceInit();
```

**Sensors initialization**

```
    Set_DeviceConnectable();
```

**Set device in advertising**

```
    while(1){
```

**Start of while loop**

```
        BTLE_StackTick();
```

**Advances the stack FSM**

```
        User_AppTick();
```

**Advances the application FSM. THIS IS DEVELOPERS USER SPACE!**

```
    } // end while(1)  
}
```

# BLE Flow on the LAB

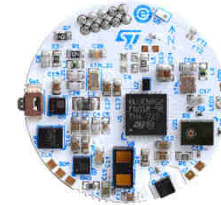
53



Central



Peripheral



## Step 2: Scan

Master is in discovery mode looking for a **specific** slave to connect to

## Step 3: Connection request

## Step 4: Services & Chars discovery

**Master (Client) starts** the BLE Service and Characteristics discovery procedure to understand the Server ATT DB

Advertise

Advertise

Advertise

Connection request

Connection created

Service & Characteristics discovery

Notification

100msecs

Notification

## Step 1: Advertising

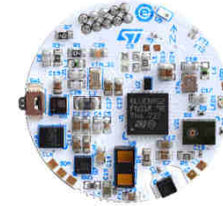
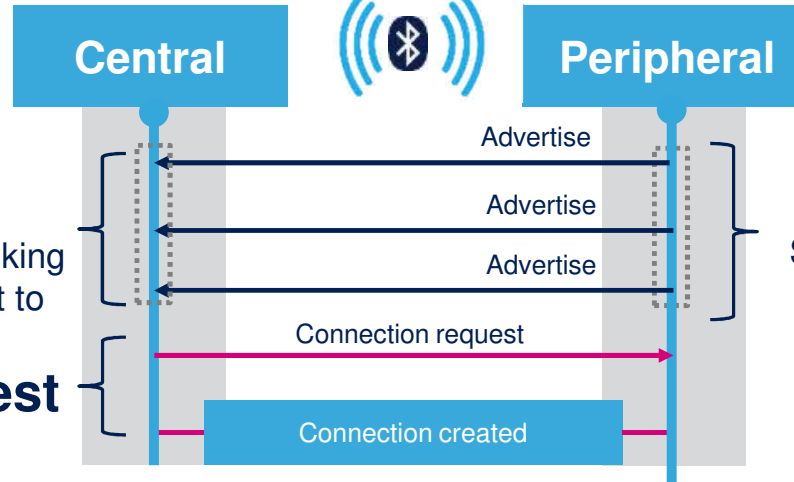
Slave is in Peripheral mode and sends ADV\_IND PDU

## Step 5: Data flow

**Slave (Server) starts** sending periodically (100 ms) to the master, **notifications packets** of sensors values (acc&gyro and pressure)

# Advertising and Scanning

54



## Step 1: Advertising

Slave is in Peripheral mode and sends ADV\_IND PDU

## Step 2: Scan

Master is in discovery mode looking for a specific slave to connect to

## Step 3: Connection request

## Master:

needs an app for discovering the slave device in advertising

Off-the-shelf app: e.g.



LightBlue® Explorer

Custom app: e.g.



<https://itunes.apple.com/us/app/lightblue-explorer/id557428110?mt=8>

ST15



## Slide 54

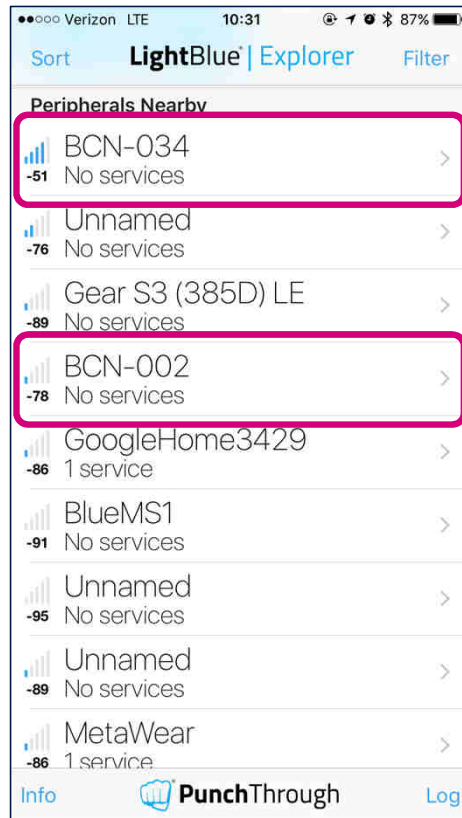
---

**ST15** Possible ST LOGO CRASH:  
Please rearrange slide layout.  
If not possible, hide ST logo.  
Corporate Com', 9/5/2019

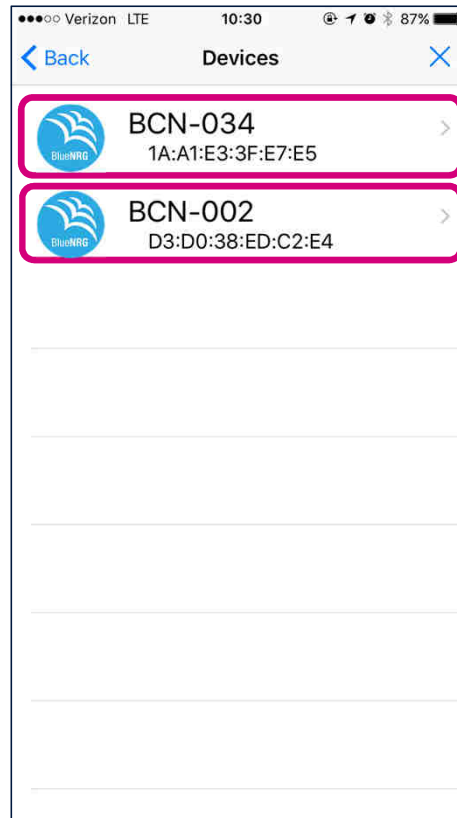
# Scan Results

55

## LightBlue scan results



## ST BLE Sensor scan results



Why some devices are not present in the ST BLE Sensor app?

How devices will pop up in the ST BLE Sensor scan list?

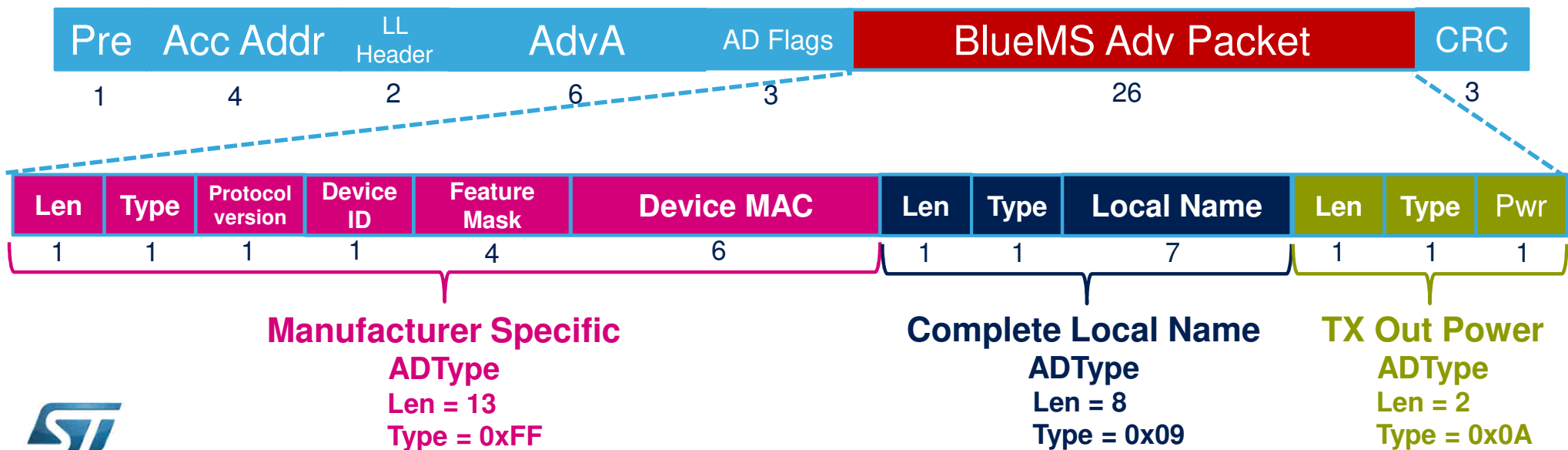




# ST BlueMS Protocol

56

- In order to be connected to the **ST BLE Sensor** app, a BLE peripheral **SHALL comply** with a specific advertising packet format
- ST BlueMS protocol** specifies a **26-byte packet format** composed of Advertisement Types – **ADTypes** - compliant with BT SIG definitions



# BlueST SDK for Android and IOS

57

## Android

[https://github.com/stmicroelectronics-centrallabs/bluestsdk\\_android](https://github.com/stmicroelectronics-centrallabs/bluestsdk_android)

## Source code for iOS and Android

available online and maintained on GitHub repos

## iOS

[https://github.com/stmicroelectronics-centrallabs/bluestsdk\\_ios](https://github.com/stmicroelectronics-centrallabs/bluestsdk_ios)



**GitHub** This repository Search Explore Features Enterprise Pricing

STMicroelectronics-Centrallabs / BlueSTSDK\_Android Watch 3

Bluetooth low energy Sensors Technology Software Development Kit (Android version)

2 commits 2 branches 0 releases 1 contributor

Branch: master BlueSTSDK\_Android / +

GiovanniVisentini Update README Latest commit 573ce4 10 days ago

BlueSTExample	Initial commit	14 days ago
BlueSTSDK	Initial commit	14 days ago
.gitignore	Initial commit	14 days ago
LICENSE	Initial commit	14 days ago
README.md	Update README	10 days ago
build.gradle	Initial commit	14 days ago
settings.gradle	Initial commit	14 days ago

STMicroelectronics-Centrallabs / BlueSTSDK\_iOS Watch 3

Bluetooth low energy Sensors Technology Software Development Kit (iOSVersion)

2 commits 2 branches 0 releases 1 contributor

Branch: master BlueSTSDK\_iOS / +

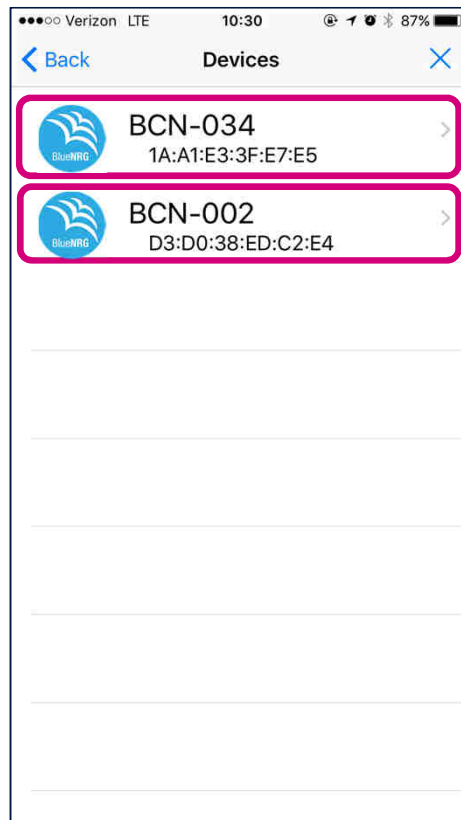
GiovanniVisentini fix parse error with multiple feature in the same characteristics Latest commit c0e8d1 2 days ago

BlueSTSDK	fix parse error with multiple feature in the same characteristics	2 days ago
BlueSTSDKExample	fix parse error with multiple feature in the same characteristics	2 days ago
LICENSE	Initial commit	9 days ago
Readme.md	Initial commit	9 days ago

# ST BLE Sensor Scan Results

58

Len	Type	Protocol version	Device ID	Feature Mask	Device MAC	Len	Type	Local Name
-----	------	------------------	-----------	--------------	------------	-----	------	------------



BCN-034	1A:A1:E3:3F:E7:E5
BCN-002	D3:D0:38:ED:C2:E4

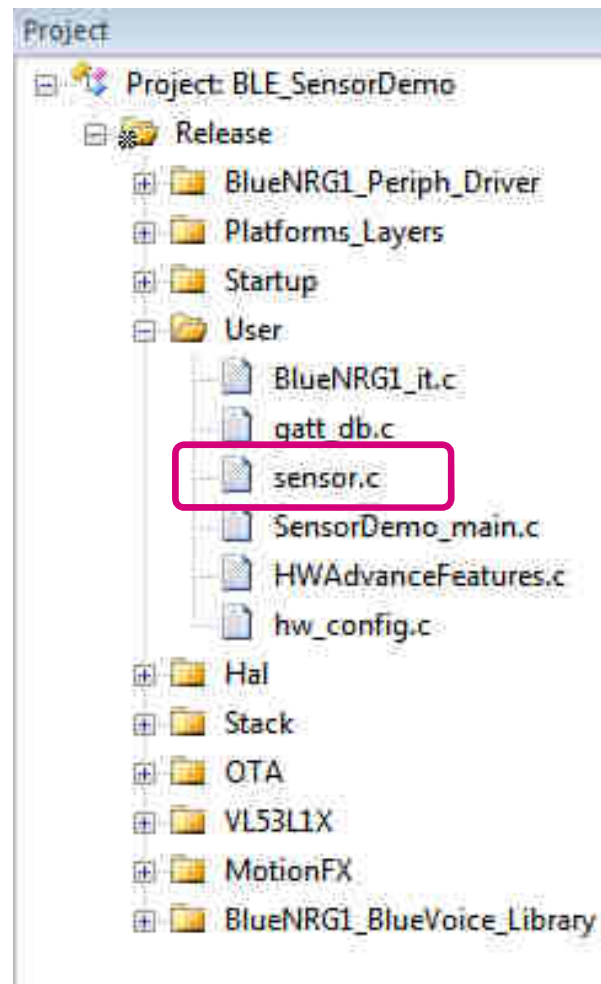
Each ST Platform is recognized by means of a  
**Device ID**  
(for STEVAL-BCN002V1 it is equal to **0x05**)

Users can distinguish their node from  
the **Local Name**



# Customize Your BlueTile

59



# Customize Your BlueTile

60

Modify the **local name** in the advertisement payload

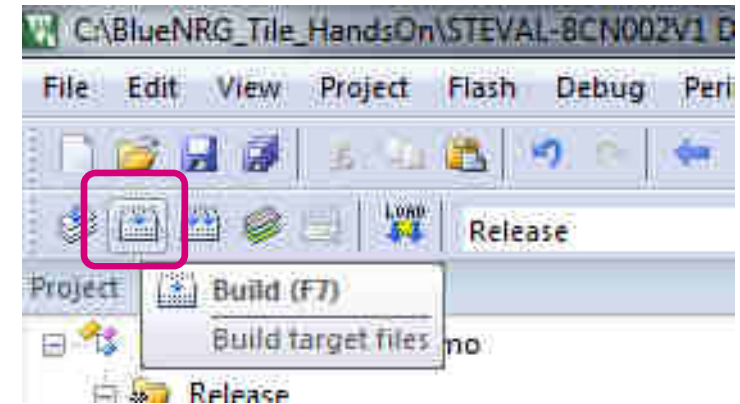
1. In the file **sensor.c** go to **line 52**
2. “**BCN-002**” is the default local name value
3. You can modify it as you prefer with a 7-characters string

```
51  
52 #define DEFAULT_NAME_ALLMEMS 'B','C','N','-','0','0','2'  
53
```

# Build the New Code

61

1. Click on the **Build button** (top left corner) or hit **F7** on your keyboard
2. In the **Build Output** window (bottom) wait for the build to be completed.
  - **BLE\_SensorDemo.bin** created
  - “**0 Error(s), 0 Warning(s)**” message appear



```
Build Output
compiling v15311_wait.c...
compiling v15311_register_funcs.c...
compiling v15311_platform.c...
linking...
Program Size: Code=121908 RO-data=1428 RW-data=1136 ZI-data=21252
FromELF: creating hex file...
After Build - User command #1: fromelf.exe --bin ..\..\..\..\HandsOn\BLE_SensorDemo_PrjOutput\BLE_SensorDemo.axf --output ..\..\..\..\HandsOn\BLE_SensorDemo_PrjOutput\BLE_SensorDemo.bin
"..\..\..\..\HandsOn\BLE_SensorDemo_PrjOutput\BLE_SensorDemo.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:12
```

# Programming Embedded Flash

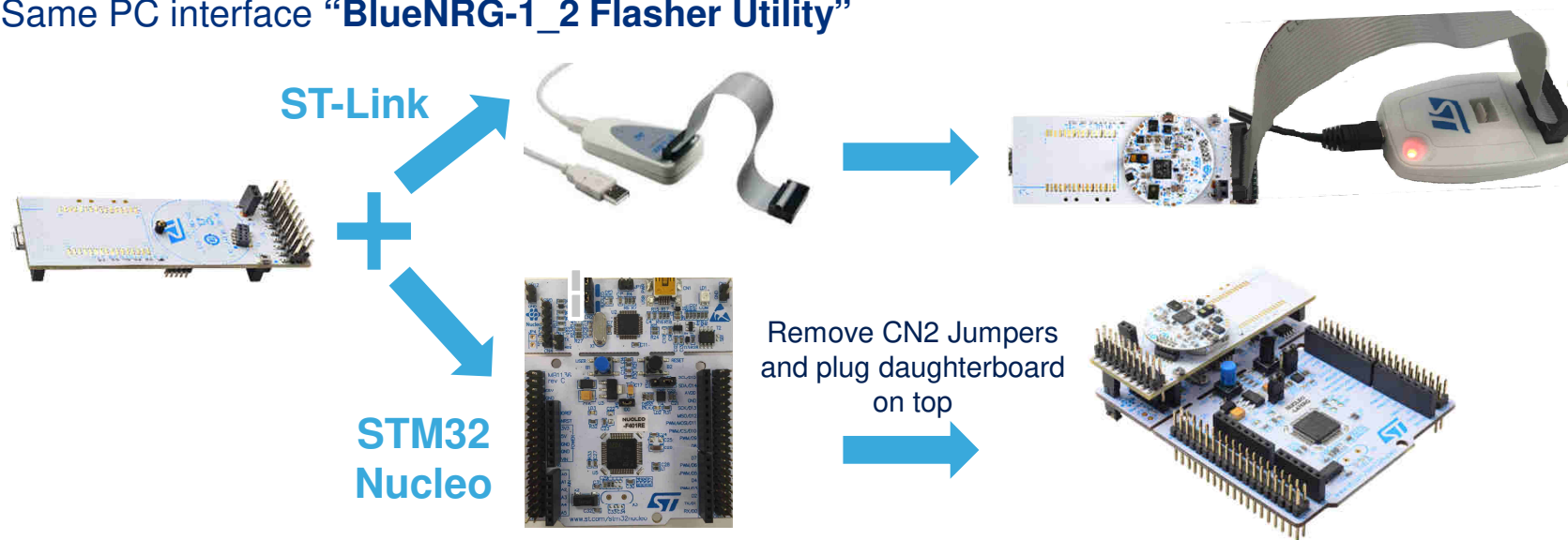
62

## 1. **UART** Bootloader

- **ROM bootloader**. HW activation through dedicate pin (**DIO7**) configured for Boot
- **PC interface** named “**BlueNRG-1\_2 Flasher Utility**” available in the SW package

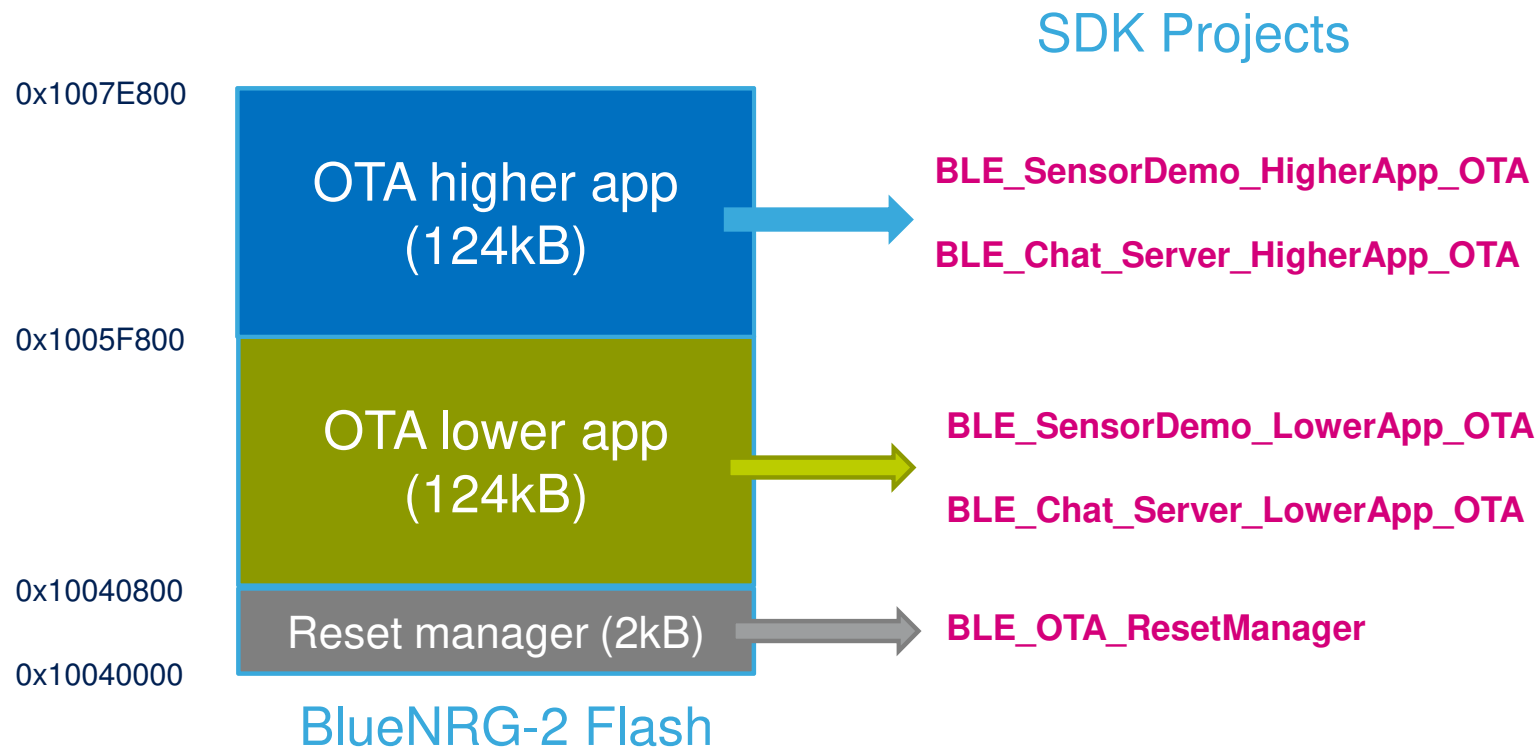
## 2. **SWD** interface + ST-LINK

- Interface with the STEVAL-BCN002V1 through the **20-pin JTAG connector**
- Same PC interface “**BlueNRG-1\_2 Flasher Utility**”



# Flash Memory Layout 1/2

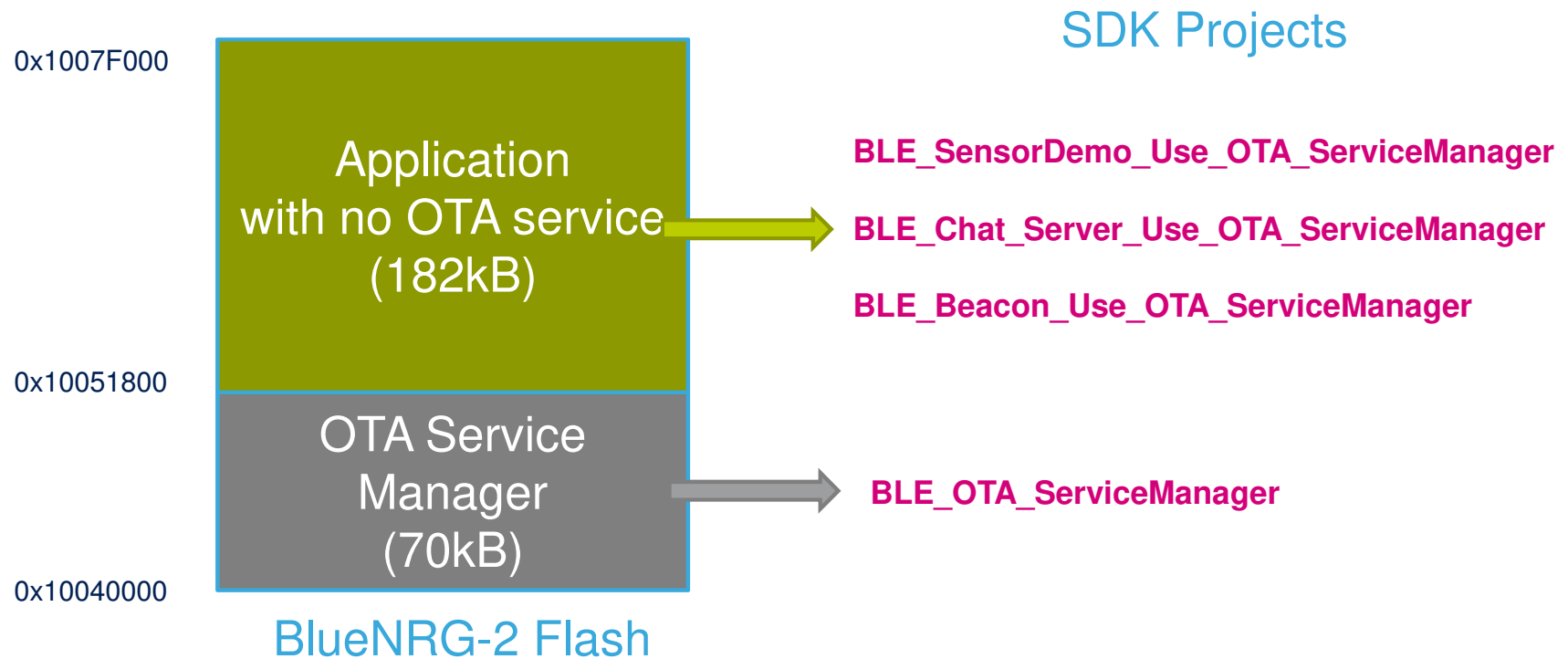
63





# Flash Memory Layout 2/2

64



# UART Bootloader: Use the BlueNRG Flasher Utility

65

- On st.com at this link:

<https://www.st.com/en/embedded-software/stsw-bnrgflasher.html>



The STSW-BNRGFLASHER is a standalone PC application which allows the BlueNRG-1, BlueNRG-2 devices Flash to be read, mass erased, written and programmed.

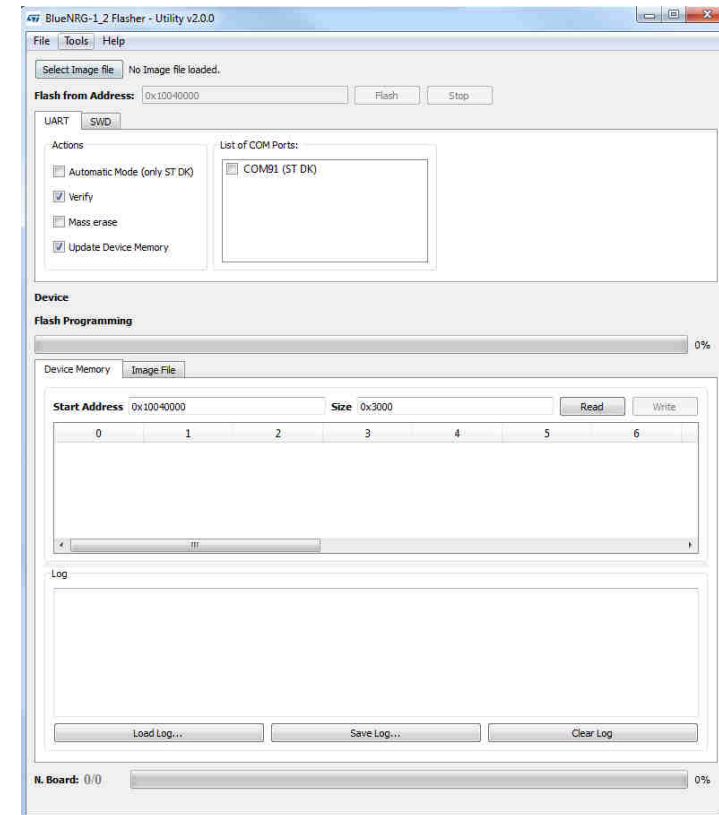
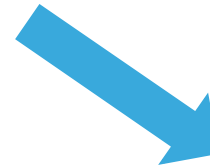
# Open the BlueNRG Flasher Utility

66

## 1. Go to

C:\Program Files (x86)\STMicroelectronics\BlueNRG-1\_2 Flasher Utility 3.0.0\Application

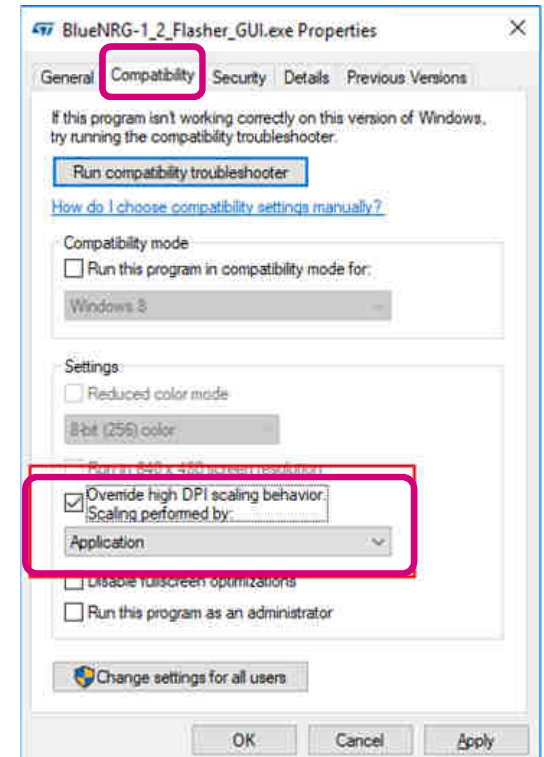
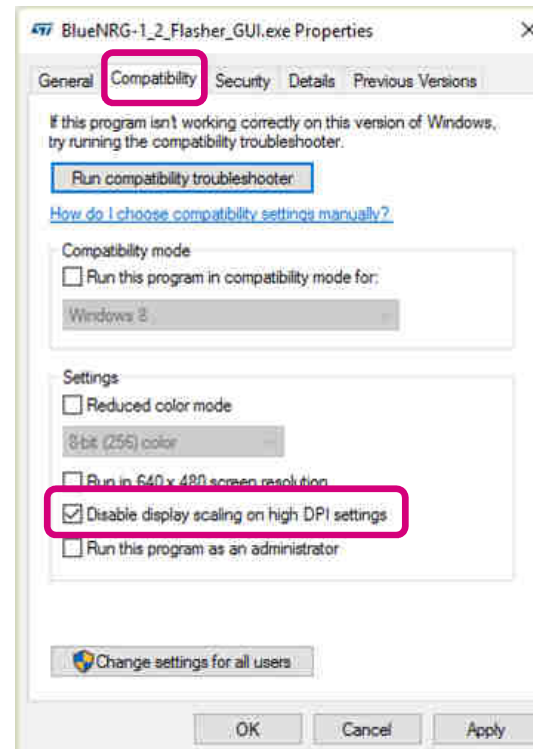
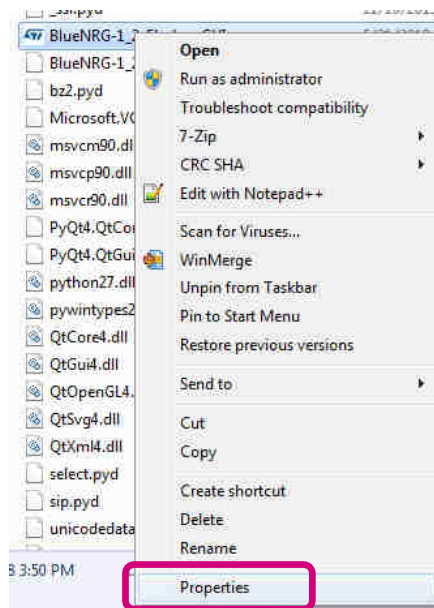
## 1. Double click on BlueNRG-1\_Flasher\_GUI.exe



# For HD Screens...If You've Screen Resolution problems

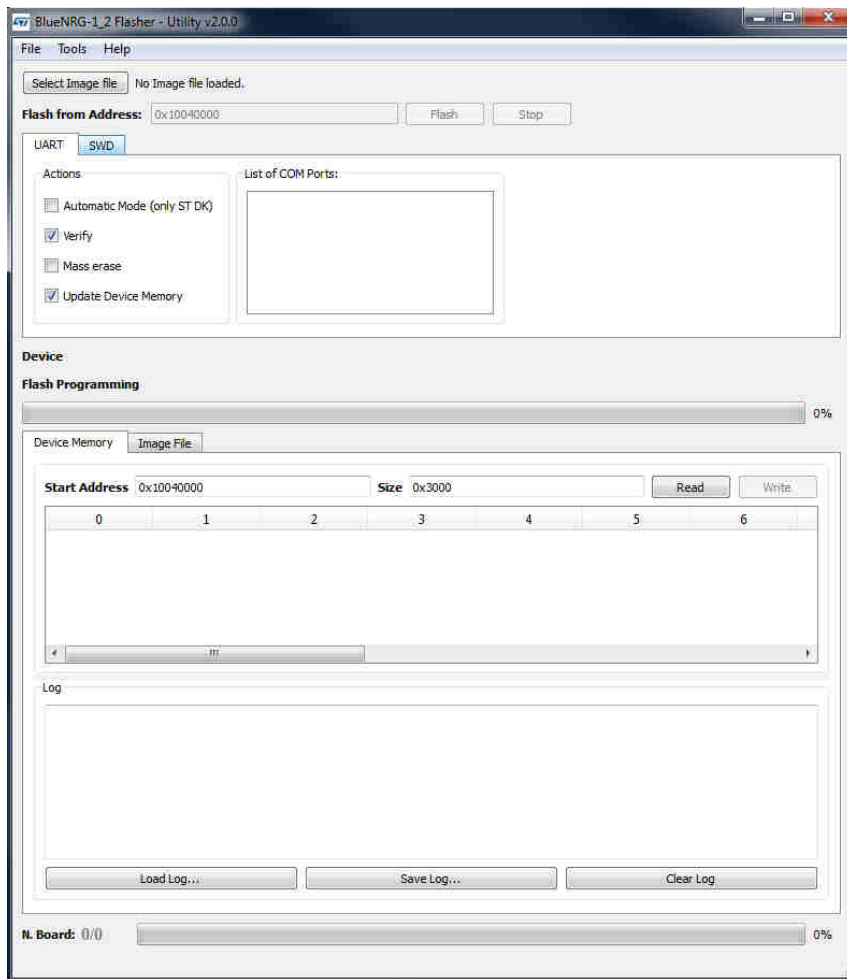
67

- **Right Click** on the .exe file and **select Properties**
- In the **Compatibility** tab **select** either **Disable display scaling...** or **Override high DPI scaling...** → **Application**



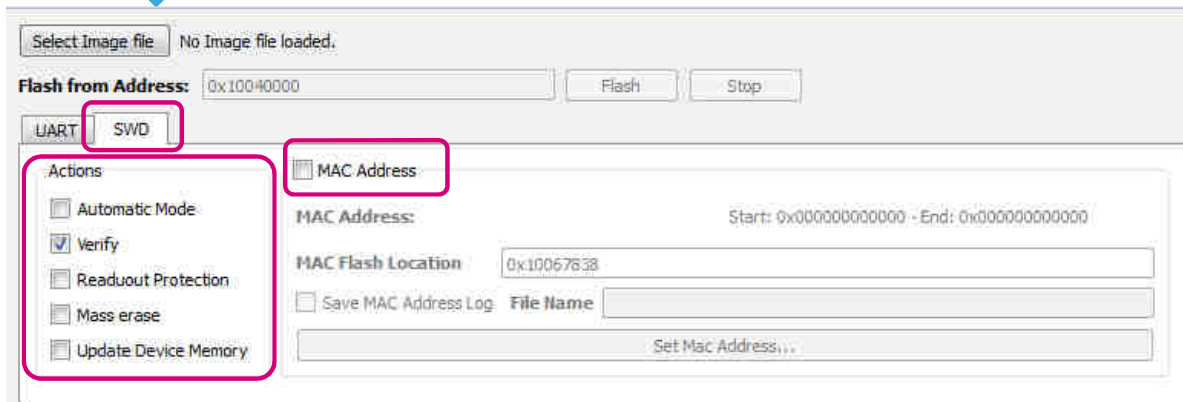
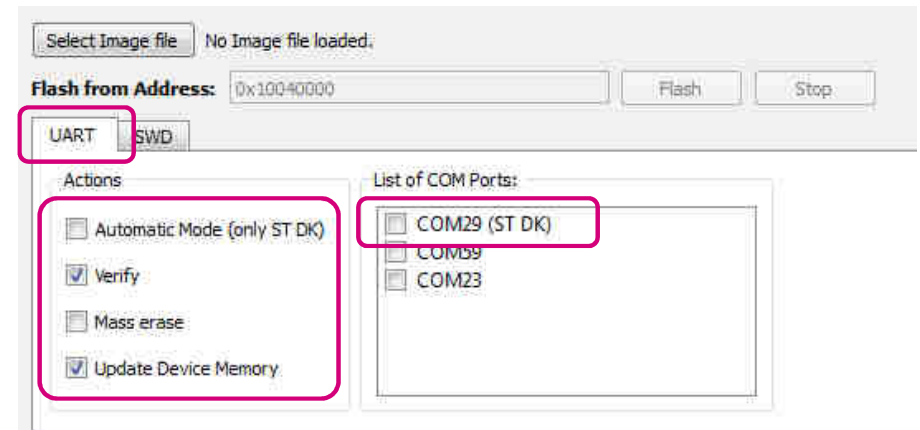
# BlueNRG Flasher Utility

68



UART

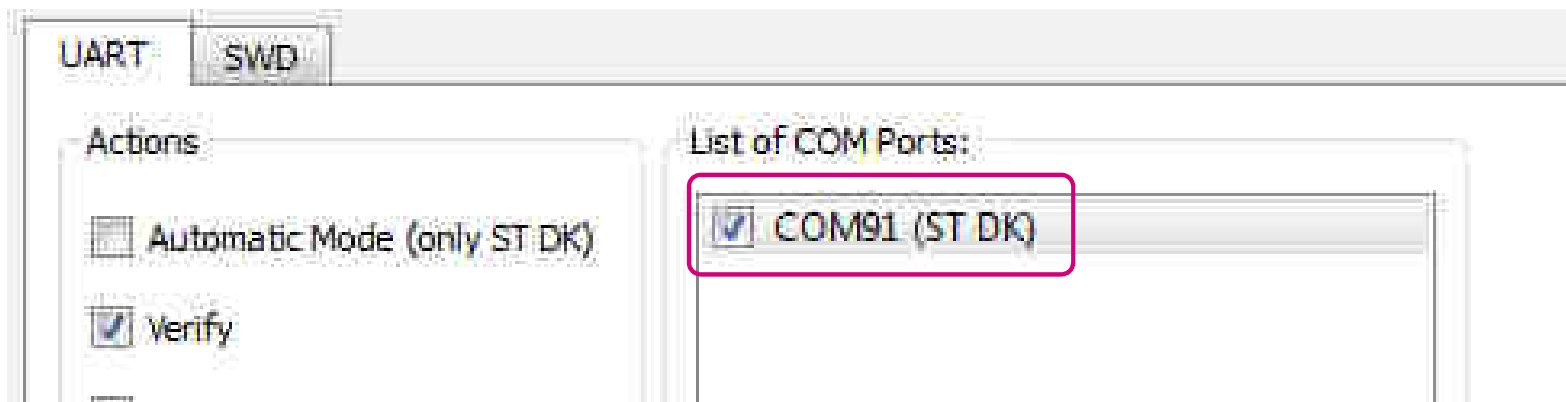
SWD



# Flash the BlueNRG-2 1/6

69

## 1. Select the COM port labeled (ST DK)



# Flash the BlueNRG-2 2/6

70

## 2. Device Memory will populate with data

Device: BlueNRG-2 (max flash address: 0x1007ffff)

Reading Memory COM91

100%

Device Memory COM91 Image File

Start Address 0x10040000 Size 0x3000 Read Write

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ASCII
0x10040000	00	60	00	20	D9	07	04	10	35	06	04	10	3D	06	04	10	" U 5 =
0x10040010	AA	55	55	AA	00	00	00	00	00	00	00	00	00	00	00	00	#UU#
0x10040020	00	00	00	00	00	00	00	00	00	00	00	00	45	06	04	10	E
0x10040030	00	00	00	00	00	00	00	00	4F	06	04	10	59	06	04	10	O Y

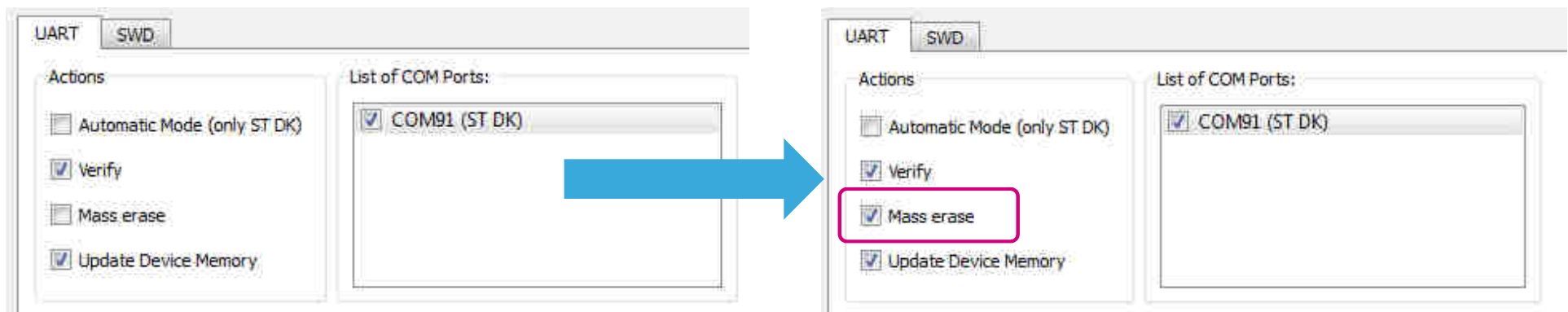
Log

```
16:10:30.632: Device COM91 -> Read Board Started
16:10:31.701: Device COM91 -> Read Board End
```

# Flash the BlueNRG-2 3/6

71

## 3. Select Mass Erase

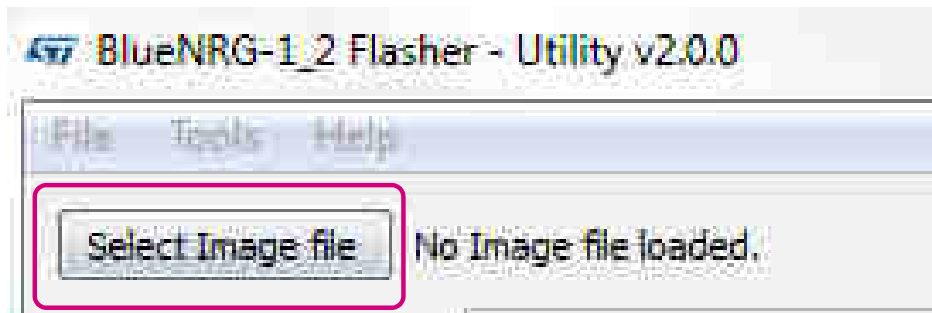




# Flash the BlueNRG-2 4/6

72

4. Click on the **Select Image file** button



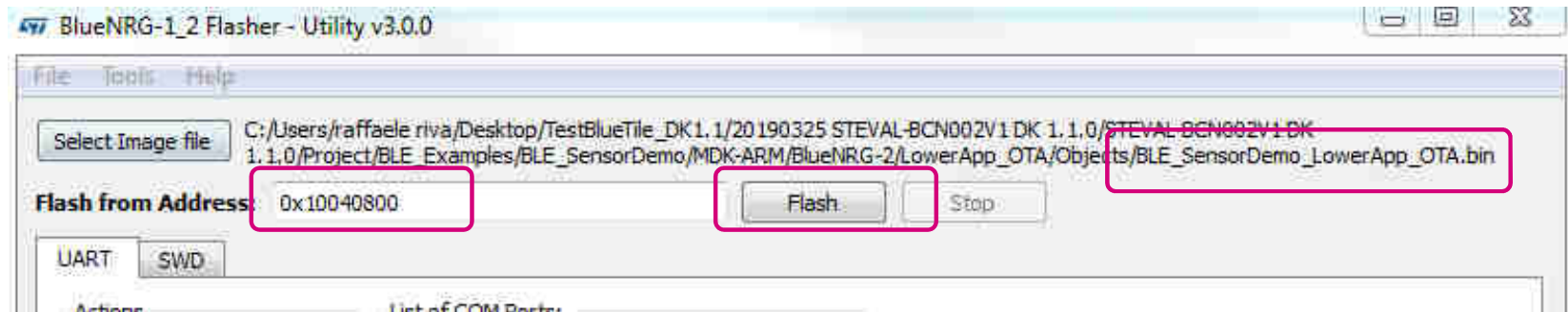
5. And **browse** the following **path**

STEVAL-BCN002V1 DK1.1.0 ▶ Project ▶ BLE\_Examples ▶ BLE\_SensorDemo ▶ MDK-ARM ▶ BlueNRG-2 ▶ LowerApp\_OTA ▶ Objects

# Flash the BlueNRG-2 5/6

73

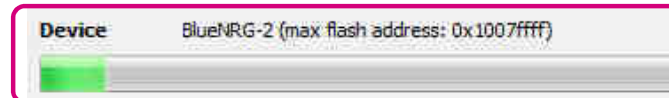
6. **Select** the binary file (e.g. BLE\_SensorDemo\_LowerApp\_OTA.bin) and **click Open**
7. Insert the correct **start address** (e.g. 0x10040800 for Lower App)
8. **Click** on the **Flash** button



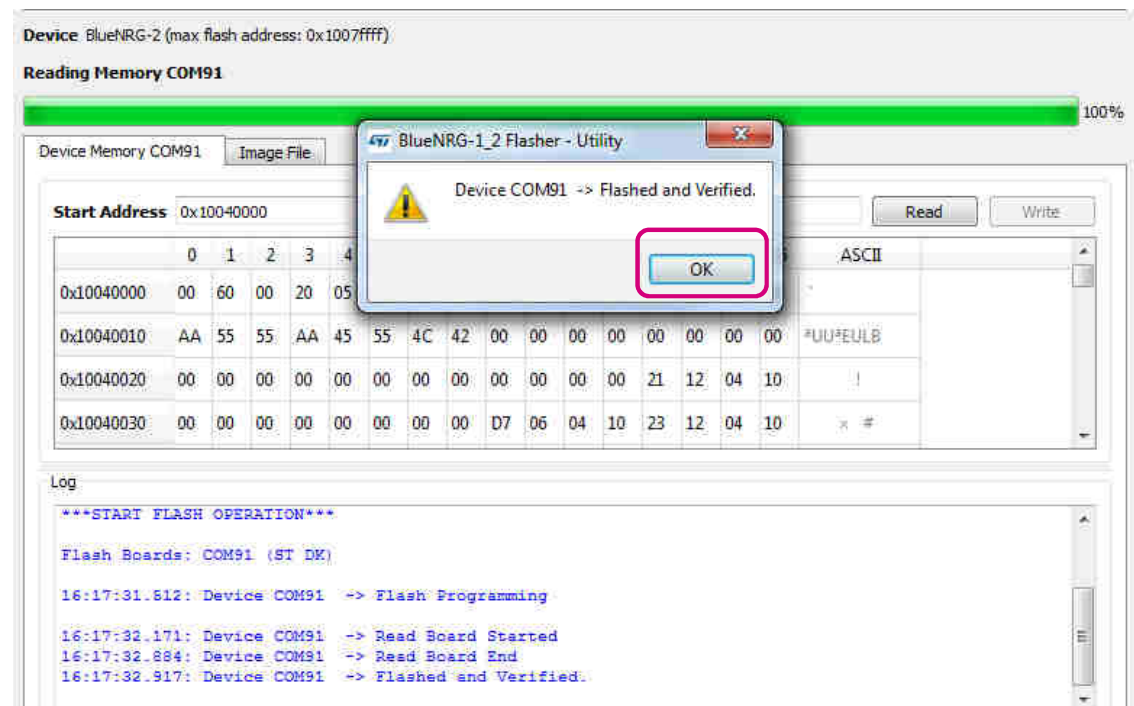
# Flash the BlueNRG-2 6/6

74

9. Flashing starts: **green bar** proceeding



10. Wait for the **pop-up** window and **click** on **OK**



# SWD: Use BlueNRG ST-LINK Utility

75

- On st.com at this link:

<https://www.st.com/en/embedded-software/stsw-bnrg1stlink.html>

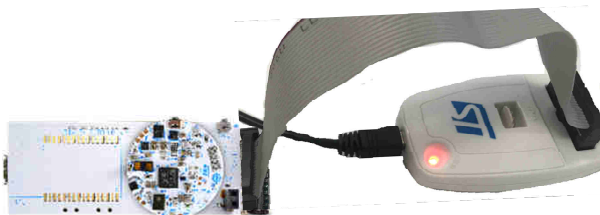


The BlueNRG-1 ST-LINK utility is a full-featured software interface for programming BlueNRG-1 and BlueNRG-2 devices.

# Programming Embedded Flash

76

## 1. SWD interface + ST-LINK



BlueNRG-1 ST-LINK Utility

File Edit View Target ST-LINK Help

Memory display

Address: 0x10040800 Size: 0x16FA8 Data Width: 32 bits

Device: BlueNRG-2  
Version ID: Ver 0  
Revision ID: Rev 0  
Flash size: 256KBytes

Device Memory @ 0x10040800 : Binary File

Target memory, Address range: [0x10040800 0x100577A8]

Address	0	4	8	C	ASCII
0x10040800	20006000	1004764D	10047931	10047933	.. Mv...1y...3y..
0x10040810	AA5555AA	424C5545	00000000	00000000	*U U *E U L B.....
0x10040820	00000000	00000000	00000000	10047935	.....5y..
0x10040830	00000000	00000000	100474AF	10047939	.....t..9y..
0x10040840	10047959	100474B1	00000000	00000000	Yy..t.....
0x10040850	100479B5	100474B3	100479C9	100474BF	py..t..Éy..gt..
0x10040860	00000000	00000000	00000000	00000000	.....
0x10040870	00000000	100474C1	100474C3	100474C5	...Ät..Ät..Ät..
0x10040880	00000000	100474B5	100474B7	100474B9	...pt...t...t..

03:43:03 : ST-LINK SN : 57FF70065088495348330487  
03:43:03 : ST-LINK Firmware version : V2J31S7  
03:43:03 : Connected via SWD.  
03:43:03 : SWD Frequency = 4,0 MHz.  
03:43:03 : Connection mode : Normal.  
03:43:03 : Debug in Low Power mode enabled.  
03:43:03 : Device : BlueNRG-2  
03:43:03 : Device family :BlueNRG-2  
03:43:03 : Device flash Size : 256KBytes

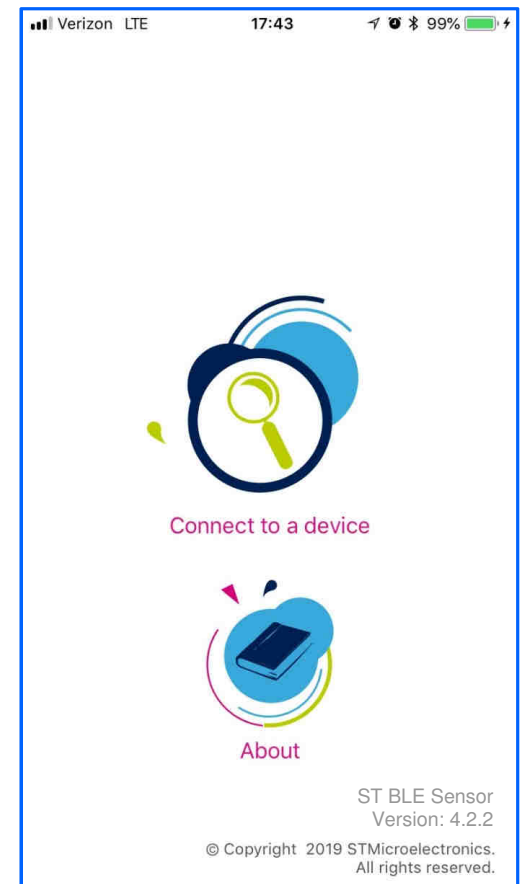
Debug in Low Power mode enabled. Device : BlueNRG-2 Core State : LiveUpdate Disabled

# Open the ST BLE Sensor App 77



Launch the **ST BLE Sensor** app previously installed

Note: in the following slides all the pictures are referred to the iOS version of the ST BLE Sensor app. The Android version is slightly different

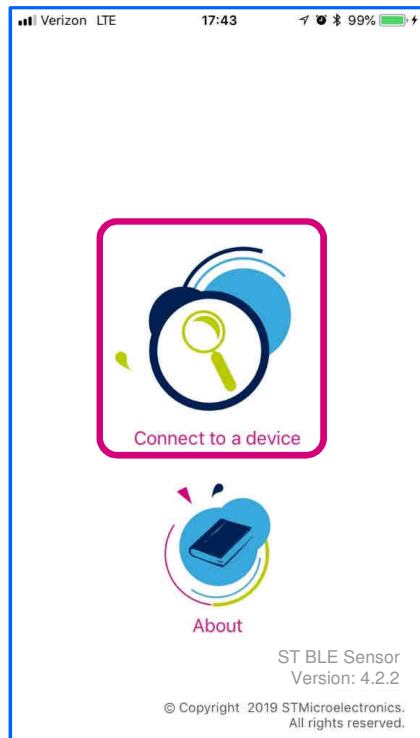


# Connect Using the ST BLE Sensor App

78

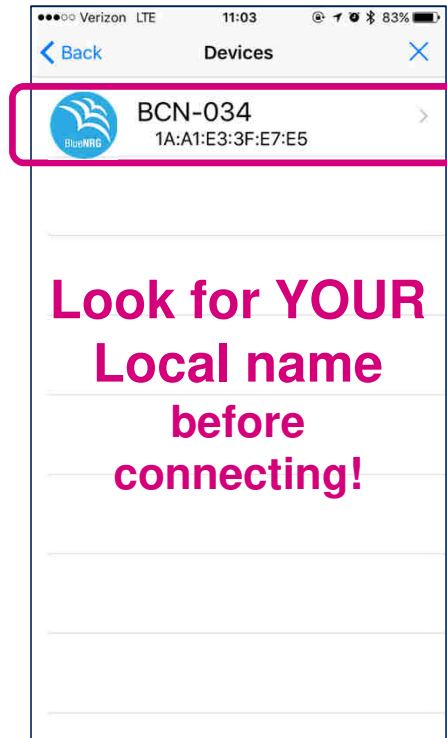
1

**Tap  
“Connect to a device”**



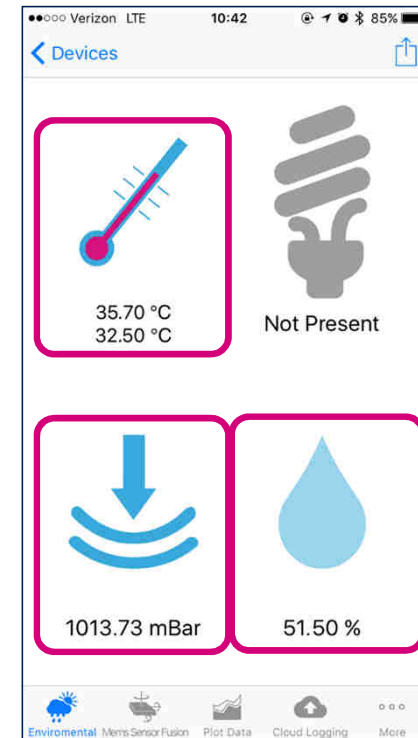
2

**Select your  
BlueTile**



3

**You are  
connected**

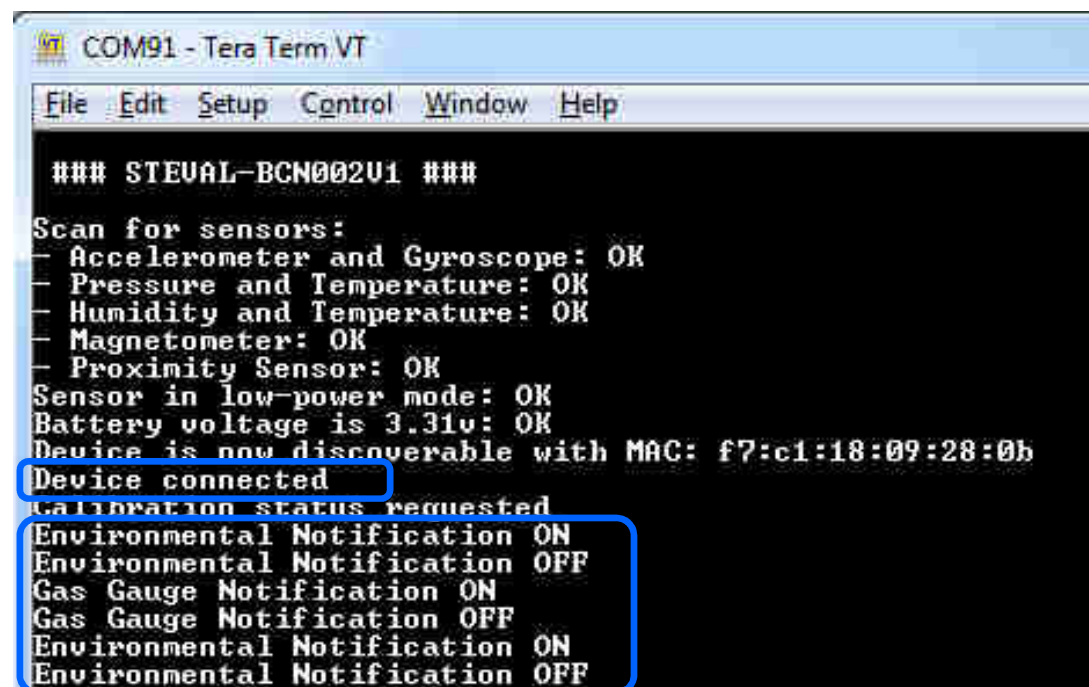


**Blow air on  
the  
Temperature  
sensor to  
change the  
temperature  
and humidity  
values**

# Tera Term Output

79

- **“Device connected”** will appear as connection is created
- Each time user moves on different tabs/screens in the app:
  - **BLE notifications** on different BLE characteristics **are enabled**.
  - **Environmental and Gas Gauge/Battery** characteristics are involved



```
COM91 - Tera Term VT
File Edit Setup Control Window Help

### STEVAL-BCN002U1 ###
Scan for sensors:
- Accelerometer and Gyroscope: OK
- Pressure and Temperature: OK
- Humidity and Temperature: OK
- Magnetometer: OK
- Proximity Sensor: OK
Sensor in low-power mode: OK
Battery voltage is 3.31v: OK
Device is now discoverable with MAC: f7:c1:18:09:28:0b
Device connected
Calibration status requested
Environmental Notification ON
Environmental Notification OFF
Gas Gauge Notification ON
Gas Gauge Notification OFF
Environmental Notification ON
Environmental Notification OFF
```



# ST BlueMS Protocol 80

In file **sensor.c**

Len	Type	Protocol version	Device ID	Feature Mask	Device MAC	Len	Type	Local Name	Len	Type	Pwr
-----	------	------------------	-----------	--------------	------------	-----	------	------------	-----	------	-----

```

389  uint8_t manuf_data[26] = { 2, /* length*/0x00, 0x00 /* 0 dBm */ // Transmission Power
390  0, /* Length*/0x00, DEFAULT_NAME_ADNEMS, // Complete Name
391  13, /* Length*/0xFF, 0x01, /* SKD version */
392  0x05, /* 0x05 BlueNRG-Tile Board */
393  #if ENABLE_BLUEVOICE
394      0x6A, /* ADPCM Sync + ADPCM Audio + Led + Prox */
395  #else
396      0x22, /* Led + Prox */
397  #endif
398      0xFE, /* Acc + Gyro + Mag + Press + Hum + Temp + Batt*/
399  #if ENABLE_MOTIONFX
400      0x05, /* AccEvents + iNemo Compact */
401      0x40, /* eCompass */
402  #else
403      0x04, /* AccEvents */
404      0x00,
405  #endif
406      0x00, /* BLE MAC start */
407      0x00, 0x00, 0x00, 0x00, 0x00, /* BLE MAC stop */
408  };

```

3 bytes TX output power

9 bytes Local Name

13 bytes Manufacturer Specific



# BlueMS Protocol

## Complete Feature Mask

81

4 Bytes

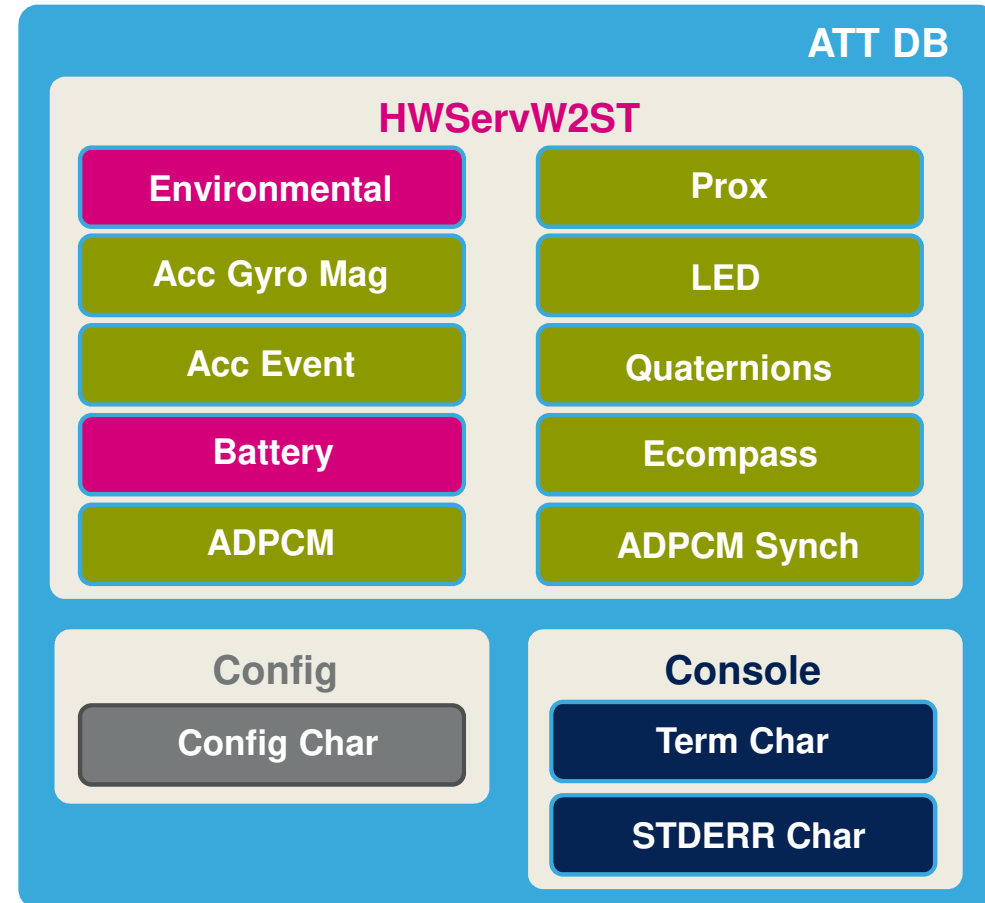
Len	Type	Protocol version	Device ID	Feature Mask	Device MAC	Len	Type	Local Name	Len	Type	Pwr
-----	------	------------------	-----------	--------------	------------	-----	------	------------	-----	------	-----

31	30	29	28	27	26	25	24	0x6A
RFU	ADPCM	Switch	DoA	ADPC	MicLevel	Proximity	Lux	
23	22	21	20	19	18	17	16	0xFE
Acc	Gyro	Mag	Pressure	Humidity	Temperat	Battery	2nd Temp	
15	14	13	12	11	10	9	8	0x05
RFU	RFU	RFU	RFU	Beamform	AccEvent	FreeFall	SensFusC	
7	6	5	4	3	2	1	0	0x40
SensFus	Compass	MotionInt	Activity	Carry Pos	MemsGes	ProxGes	Pedo	

# How Feature Mask is Mapped onto ATT DB?

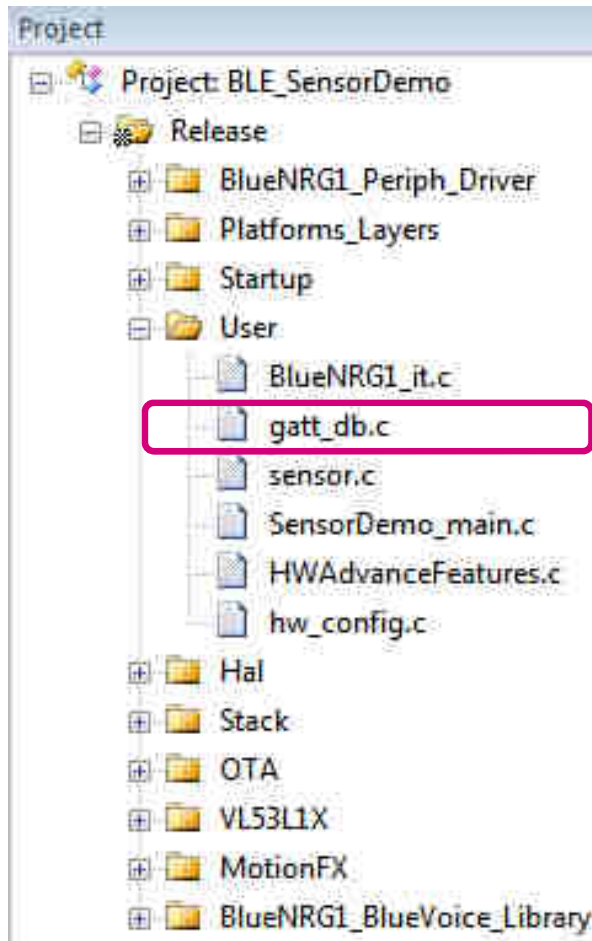
82

- **3 services:** HWServW2ST, Config, and Console
- **Each bit** of the 4-byte Feature Mask in the Advertising packet **corresponds to a HW/SW feature**
- In the Server ATT DB a **BLE Characteristic** has to be added for each bit of the Advertising Feature Mask



# BLE\_SensorDemo ATT DB

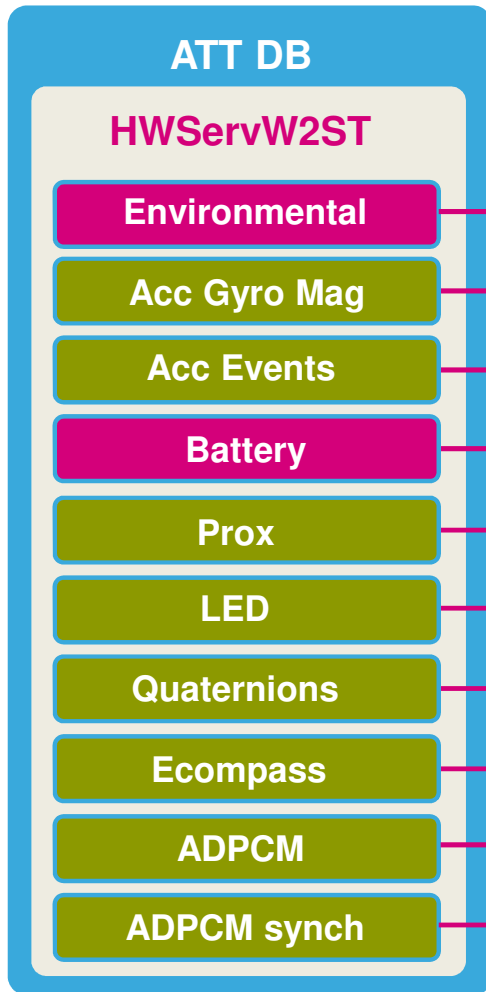
83



The **ATT DB** is defined and created in the file **gatt\_db.c**

# BLE\_SensorDemo ATT DB

84



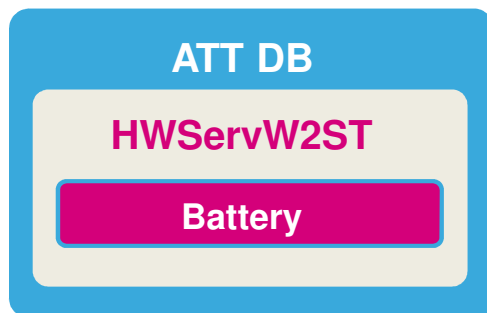
Len	PROPERTIES	UUID TYPE	UUID VALUE
12	N,R	128-bit	001D0000000111e1ac360002a5d5c51b
20	N	128-bit	00E00000000111e1ac360002a5d5c51b
5	N,R	128-bit	00000400000111e1ac360002a5d5c51b
9	N,R	128-bit	00020000000111e1ac360002a5d5c51b
4	N,R	128-bit	02000000000111e1ac360002a5d5c51b
3	N,R	128-bit	20000000000111e1ac360002a5d5c51b
8	N	128-bit	00000100000111e1ac360002a5d5c51b
4	N	128-bit	00000040000111e1ac360002a5d5c51b
20	N	128-bit	08000000000111e1ac360002a5d5c51b
6	N	128-bit	40000000000111e1ac360002a5d5c51b

NOTE: two additional bytes for a timestamp for each char

# Example: Battery Characteristic

85

31	30	29	28	27	26	25	24	0x00
RFU	ADPCM	Switch	DoA	ADPC	MicLevel	Proximity	Lux	
23	22	21	20	19	18	17	16	0x02
Acc	Gyro	Mag	Pressure	Humidity	Temperat	Battery	2nd Temp	
15	14	13	12	11	10	9	8	0x00
RFU	RFU	RFU	RFU	Beamform	AccEvent	FreeFall	SensFusC	
7	6	5	4	3	2	1	0	0x00
SensFus	Compass	MotionInt	Activity	Carry Pos	MemsGes	ProxGes	Pedo	

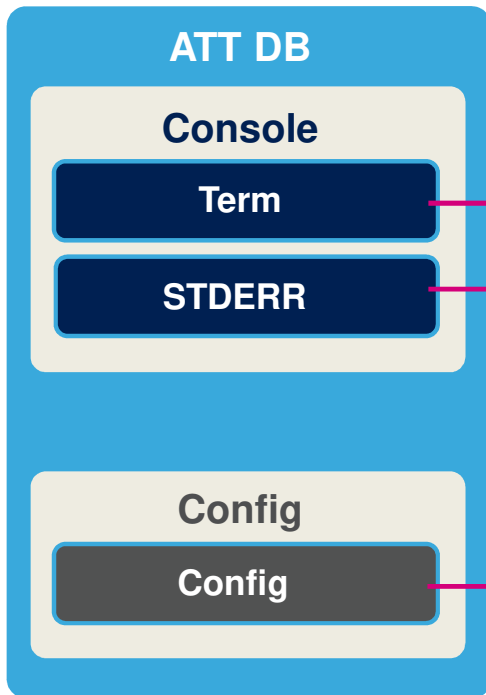


UUID VALUE
00020000000111e1ac360002a5d5c51b

The **UUID values** are linked to the Feature Mask in advertising

# BLE\_SensorDemo ATT DB

86



Each characteristic is defined by the following key parameters

Len	PROPERTIES	UUID TYPE	UUID VALUE
20	N,R	128-bit	00000001000E11e1ac360002a5d5c51b
20	N,R	128-bit	00000002000E11e1ac360002a5d5c51b

This byte differentiates the **Service**

Len	PROPERTIES	UUID TYPE	UUID VALUE
20	N, Ww/oR, R,W	128-bit	00000002000F11e1ac360002a5d5c51b

# BLE\_SensorDemo ATT DB

87

```
tBleStatus Add_HWServW2ST_Service(void)
```

```
aci_gatt_add_service(UUID_TYPE, UUID, ..., MAX_NB_ATTRIBUTES, &ServHandle)
```

```
aci_gatt_add_char(ServHandle, UUID_TYPE, UUID, Length, PROPERTIES, PERMISSIONS, ..., &CharHandle)
```



→ **HWServW2STHandle** = 0x000C

→ **EnvironmentalCharHandle** = 0x000D

→ **AccGyroMagCharHandle** = 0x0010

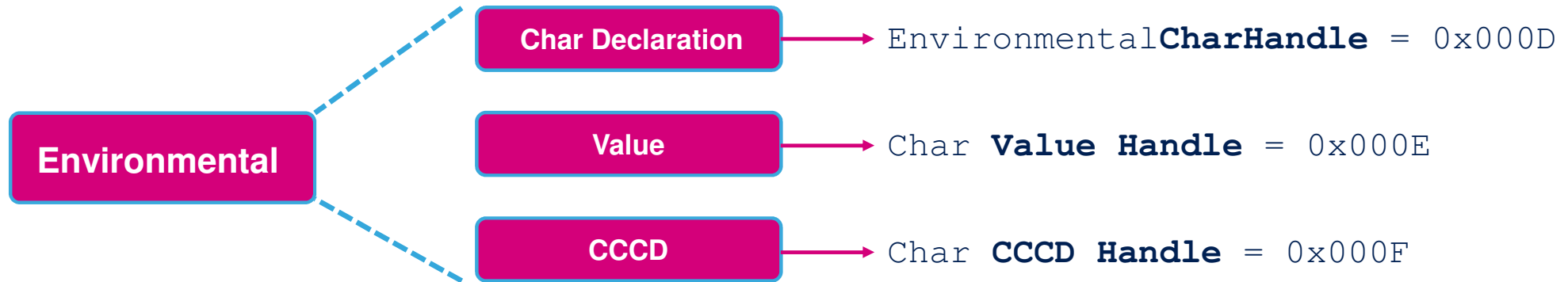
→ **ConfigW2STHandle** = 0x002B

→ **ConfigCharHandle** = 0x002C



# BLE\_SensorDemo ATT DB

88



- **Declaration Handle:** used by the application to access the Characteristic
- **Characteristic Value Handle:** used by the Client for Read/Write operations
- **Client Characteristic Configurator Descriptor (CCCD):** a GATT descriptor is added by default by the stack, if **char** has **Notify/Indicate property**. Used by Client to enable notifications/indications on char value.

# Central: Services and Characteristics

## Discovery Procedure

89

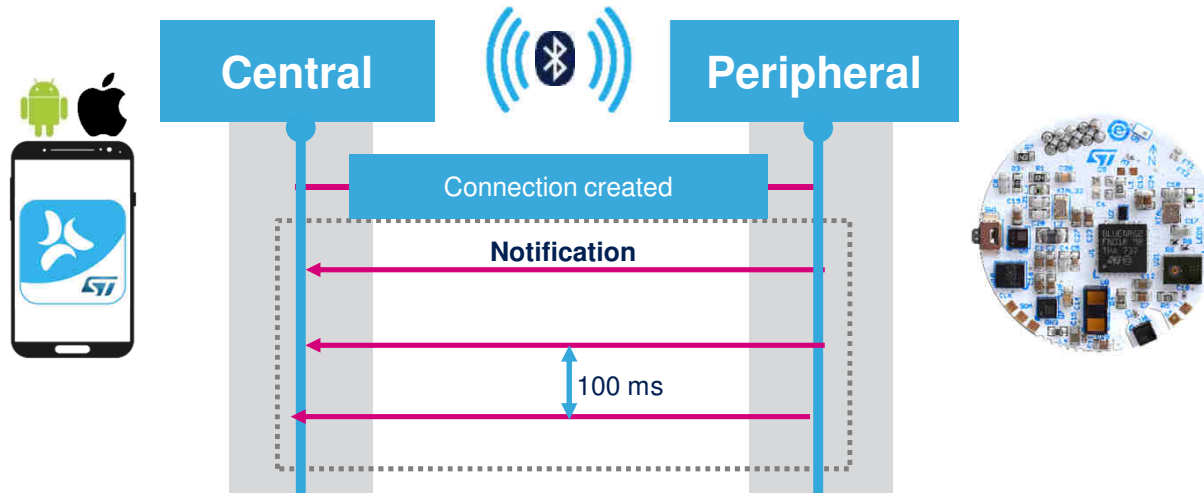
UUIDs Handles

Name	Handle	Property	Value	Values	Value Length
Service User Defined (0x00000000000011E19AB40002A5D5C51B)	0x000C				
Characteristic User Defined (0x001D0000000011E1AC360002A5D5C51B)	0x000D	Read, Notify	0x011F0111023C00018B9CA605	0x0C	
Client Characteristic Configuration (0x2902)	0x000F		0x0000	0x02	
Characteristic User Defined (0x00E00000000011E1AC360002A5D5C51B)	0x0010	Notify			
Client Characteristic Configuration (0x2902)	0x0012		0x0000	0x02	
Characteristic User Defined (0x00000400000011E1AC360002A5D5C51B)	0x0013	Read, Notify	0x0000002405	0x05	
Client Characteristic Configuration (0x2902)	0x0015		0x0000	0x02	
Characteristic User Defined (0x02000000000011E1AC360002A5D5C51B)	0x0016	Read, Notify	0x88A1232E	0x04	
Client Characteristic Configuration (0x2902)	0x0018		0x0000	0x02	
Characteristic User Defined (0x20000000000011E1AC360002A5D5C51B)	0x0019	Read, Notify	0x0024E6	0x03	
Client Characteristic Configuration (0x2902)	0x001B		0x0000	0x02	
Characteristic User Defined (0x00020000000011E1AC360002A5D5C51B)	0x001C	Read, Notify	0x0180000CBA03E8AB63	0x09	
Client Characteristic Configuration (0x2902)	0x001E		0x0000	0x02	
Characteristic User Defined (0x00000100000011E1AC360002A5D5C51B)	0x001F	Notify			
Client Characteristic Configuration (0x2902)	0x0021		0x0000	0x02	
Characteristic User Defined (0x00000040000011E1AC360002A5D5C51B)	0x0022	Notify			
Client Characteristic Configuration (0x2902)	0x0024		0x0000	0x02	
Characteristic User Defined (0x08000000000011E1AC360002A5D5C51B)	0x0025	Notify			
Client Characteristic Configuration (0x2902)	0x0027		0x0000	0x02	
Characteristic User Defined (0x40000000000011E1AC360002A5D5C51B)	0x0028	Notify			
Client Characteristic Configuration (0x2902)	0x002A		0x0000	0x02	
Service User Defined (0x000000000000F11E19AB40002A5D5C51B)	0x002B				
Characteristic User Defined (0x00000002000F11E1AC360002A5D5C51B)	0x002C	Write w/o resp, Notify			
Client Characteristic Configuration (0x2902)	0x002E		0x0000	0x02	
Service User Defined (0x000000000000E11E19AB40002A5D5C51B)	0x002F				
Characteristic User Defined (0x00000001000E11E1AC360002A5D5C51B)	0x0030	Read, Write w/o resp, Write, Notify	0x0	0x00	
Client Characteristic Configuration (0x2902)	0x0032		0x0000	0x02	
Characteristic User Defined (0x00000002000E11E1AC360002A5D5C51B)	0x0033	Read, Notify	0x0	0x00	
Client Characteristic Configuration (0x2902)	0x0035		0x0000	0x02	

This is how  
ATT DB  
looks like  
from the  
Central

# BLE Sensor – Characteristic Update

90

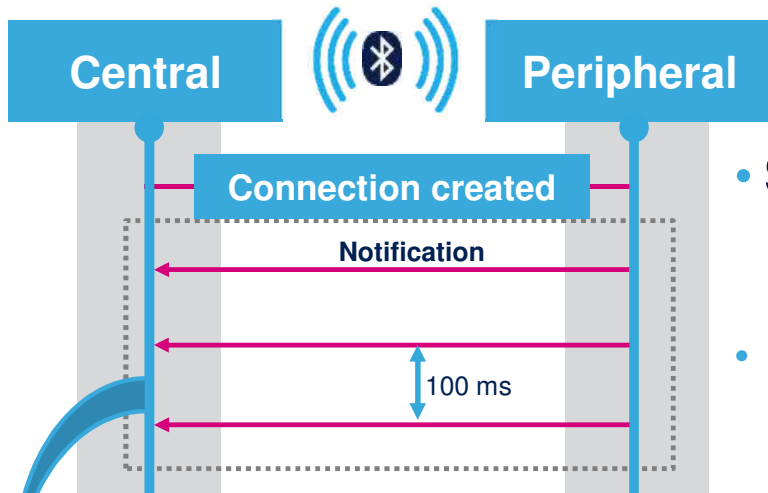


**aci\_gatt\_update\_char\_value** (ServiceHandle, 0x000C, //HWserv handle  
CharHandle, 0x000D, //EnvChar handle  
Offset, 0x00,  
Length, 0x08,  
Value) Data) //EnvChar Value

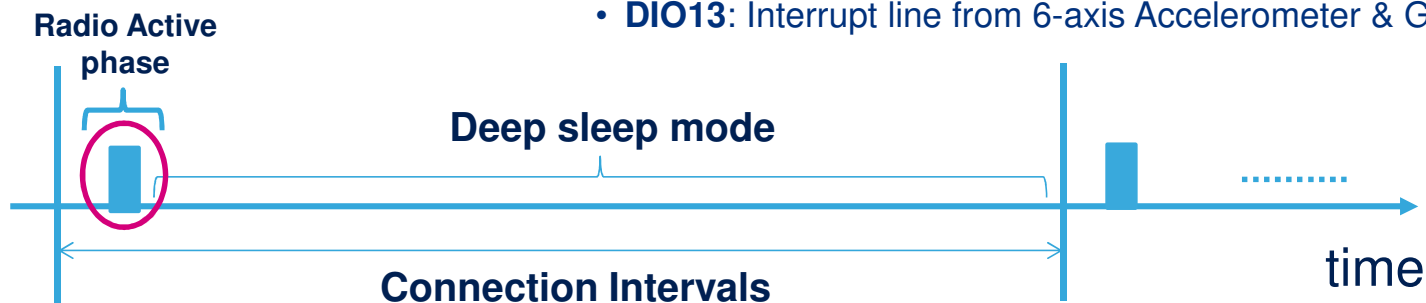
Description	Where
If notifications (or indications) are enabled on the characteristic, this API sends a notification (or indication) to the client.	GATT server

# Cortex-M0 Sleep Management

91

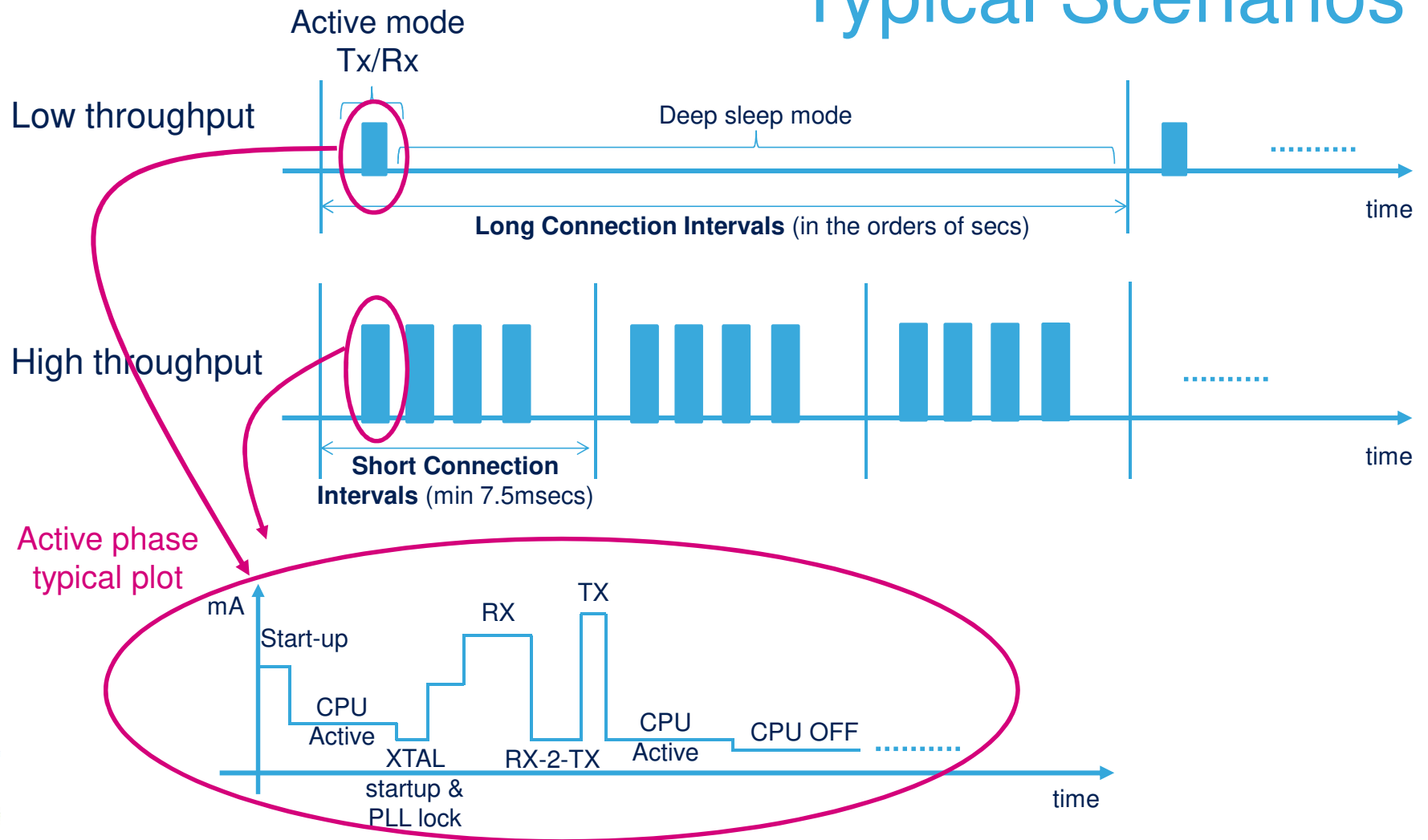


- **Sleep mode** managed through specific a **library** and **API**
  - `BlueNRG_Sleep(sleepMode, gpioWakeBitMask, gpioWakeLevelMask)`
- **4 wake-up sources: timers, DIO11, DIO12 and DIO13**
  - **Timers:** 4 virtual timers mapped on 2 physical timers managed by the radio IP
  - **DIO11:** Interrupt line from UART RX or SW1 button
  - **DIO12:** Interrupt line from Magnetometer
  - **DIO13:** Interrupt line from 6-axis Accelerometer & Gyroscope



# Typical Scenarios

92



# Low Power Modes

93

- **Deep sleep mode** can represent most of the application time.
- Efficient management of sleep mode significantly lowers the avg power consumption.

## BlueNRG-2 flexible low power architecture

Sleeping Mode	Consumption	Notes
RUNNING	1.9mA	Core running
CPU HALT	1.5mA	WFI instruction
WAKETIMER	900nA	GPIOs and Timer Wakeup
NOTIMER	500nA	GPIOs only Wakeup

Highest pwr

Low Power library combines requests coming from the application with the radio operating mode

Lowest pwr

# Context Save/Restore

94

- BlueNRG-2 Low Power Library **handles autonomously entering and exiting to/from the sleep mode.**
- **The library saves peripherals configurations and application context** before entering deep sleep, and **restores the context** on exiting from low power state:
  - CPU returns to execute **the next instruction after the low power function call**
  - No need to re-initialize **peripheral and radio stack**
  - RAM retention **is guaranteed**

**NO need for the application to worry about sleep management and RAM retention!**

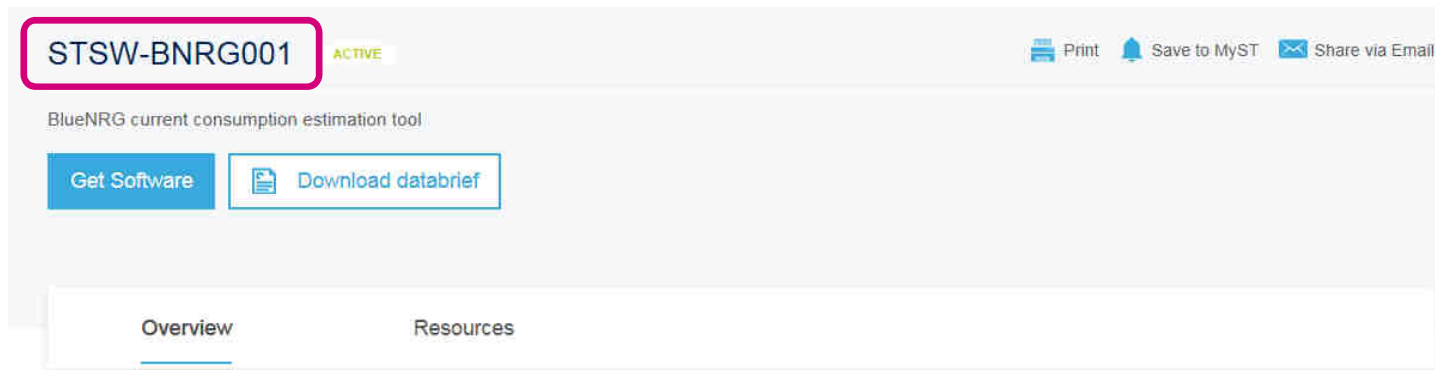
# BlueNRG

## Current Consumption Estimation Tool

95

- On st.com at this link:

<https://www.st.com/en/embedded-software/stsw-bnrg001.html>



The BlueNRG current consumption estimation tool can predict the current consumption under different conditions, as defined by the Bluetooth low energy specification, for the BlueNRG and BlueNRG-MS Bluetooth® low energy wireless network processors as well as the BlueNRG-1 and BlueNRG-2 Bluetooth® low energy systems-on-chip.



## Current Consumption Estimation Tool

- **Accurate estimate** of average **current consumption** and **battery lifetime**

The screenshot shows the BlueNRG Current Consumption Estimation Tool v.1.4 interface. The tool is divided into two main sections: 'Calculate Consumption' on the left and 'Performance Summary' on the right. The 'Calculate Consumption' section has a 'General' tab selected, which contains various configuration options. The 'Performance Summary' section displays the results of the calculation.

Numbered steps and callouts:

1. **Select the device**: Points to the 'Type of Device' dropdown menu, which is set to 'BlueNRG-2'.
2. **Select the type of event**: Points to the 'Events' dropdown menu, which is set to 'Advertising'.
3. **Configure the HW operating conditions**: Points to the 'Supply Voltage (V)' field, which is set to '3.0'.
4. **Configure the application use case scenario**: Points to the 'High Power mode' checkbox, which is checked.
5. **Specify the Battery capacity**: Points to the 'Battery capacity (mAh)' field in the 'Battery lifetime' section, which is set to '230'.
6. **Read the average current and the estimated battery lifetime**: Points to the 'Total average current' and 'Battery lifetime' fields in the 'Performance Summary' section.

Performance Summary:

- Time of active phase: 2.87 ms
- Average current during the active phase: 4.9 mA
- Total average current: **7.93 uA**
- Payload data rate: NA
- Battery lifetime:
  - Battery capacity (mAh): 230
  - Battery lifetime: **3 year(s), 3 month(s), 23 day(s)**

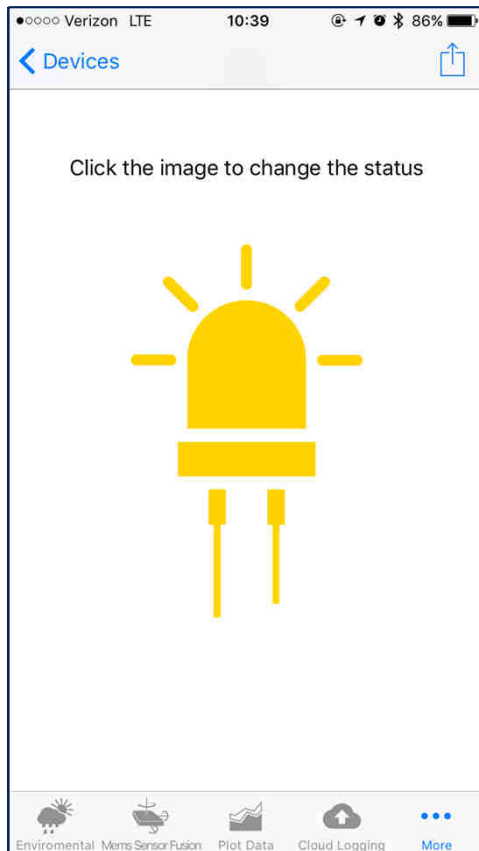


# Lab 3

## LED Characteristic

# Enable LED Toggling

98

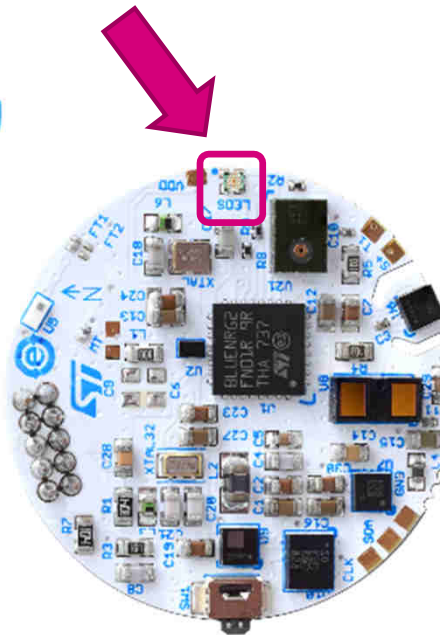


**Write  
Packet**

'1' LED ON  
'0' LED OFF

**Notification  
Packet**

'1' LED ON  
'0' LED OFF



1. **Tap** the image on the screen
2. Send **BLE notification packet**
3. Image changes color and the **RED LED toggles**

1. Client **enables LED characteristic notifications**
2. Client **Writes in Config Char**
3. Parse **write command** from the client and **send BLE notification** on LED status from the server

4 Bytes

Len	Type	Protocol version	Device ID	Feature Mask	Device MAC	Len	Type	Local Name	Len	Type	Pwr
-----	------	------------------	-----------	--------------	------------	-----	------	------------	-----	------	-----

31	30	29	28	27	26	25	24	0x20
RFU	ADPCM	Switch	DoA	ADPC	MicLevel	Proximity	Lux	
23	22	21	20	19	18	17	16	0x1E
Acc	Gyro	Mag	Pressure	Humidity	Temperat	Battery	2nd Temp	
15	14	13	12	11	10	9	8	0x00
RFU	RFU	RFU	RFU	Beamform	AccEvent	FreeFall	SensFusC	
7	6	5	4	3	2	1	0	0x00
SensFus	Compass	MotionInt	Activity	Carry Pos	MemsGes	ProxGes	Pedo	

# Client Enables LED Characteristic Notifications

101

1. In the file **gatt\_db.c** go to the **line 670**
2. **Client writes** in the **LED Characteristics Client Configurator Descriptor (CCCD)** and **Server enables notifications** through the **xFeatureNotification** structure in **line 673** by **xFeatureNotification.LedNotification = true;**

```
670 } else if (attr_handle == (LedCharHandle + 2)) {  
671  
672     if (att_data[0] == 01) {  
673         xFeatureNotification.LedNotification = true;  
674         PRINTF("Led ON\n\r");  
675     } else if (att_data[0] == 0) {  
676         xFeatureNotification.LedNotification = false;  
677         PRINTF("Led OFF\n\r");  
678     }  
679 }
```

# Client Writes in Config Char

102

1. In the file **gatt\_db.c** go to the **line 806**
2. Client **writes** in the **Config Char** and **Server parses** the command through the function **ConfigCommandParsing(att\_data, data\_length)**

```
806 } else if (attr_handle == ConfigCharHandle + 1) {  
807  
808     /* Received one write command from Client on Configuration characteristic */  
809     ConfigCommandParsing(att_data, data_length);  
810  
811 }
```

# Parse Write Command And Send BLE Notification on LED Status

103

1. In file **Sensor.c** at line **1040**
2. **Check** on the **FeatureMask** (line **1046**)
3. If it's the LED feature mask (line **1152**)
4. **Check** on the command value. If **0x01** (case1)
5. **Turn ON LED3**
6. **Send Notification** on the Config char on the command received and parsed
7. **Send Notification** on the **LED** char with the **LED status**

```
1040 uint32_t ConfigCommandParsing(uint8_t  
1041     FeatureMask, uint8_t* att_data) {  
1042     uint8_t Command = att_data[4];  
1043     uint8_t Data = att_data[5];  
1044     uint32_t SendItBack = 1;  
1045  
1046     switch (FeatureMask)
```

```
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172
```

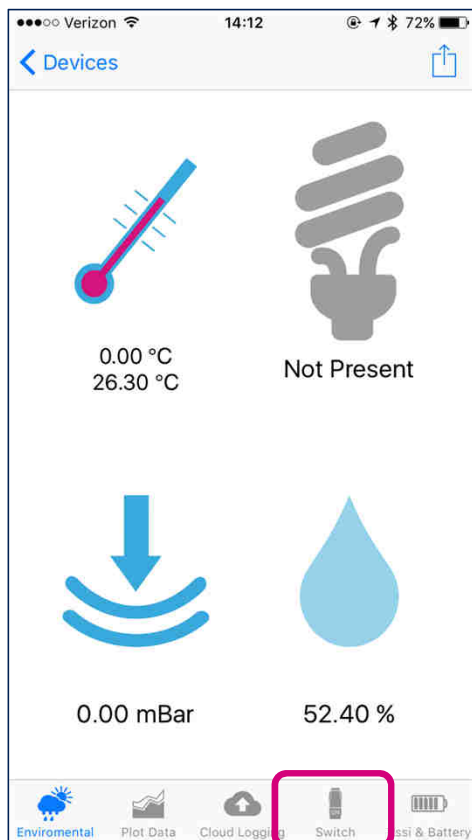
```
case FEATURE_MASK_LED:  
    switch (Command) {  
        case 1:  
            SdkEvalLedOn(LED1);  
            PRINTF("Enabled: LED\n\r");  
            Config_Notify(FEATURE_MASK_LED, Command, Data);  
            break;  
        case 0:  
            SdkEvalLedOff(LED1);  
            PRINTF("Disabled: LED\n\r");  
            Config_Notify(FEATURE_MASK_LED, Command, Data);  
            break;  
    }  
  
    if (xFeatureNotification.LedNotification) {  
        if (SdkEvalLedGetState(LED1))  
            Led_Update(ENABLE);  
        else  
            Led_Update(DISABLE);  
    }  
    break;
```



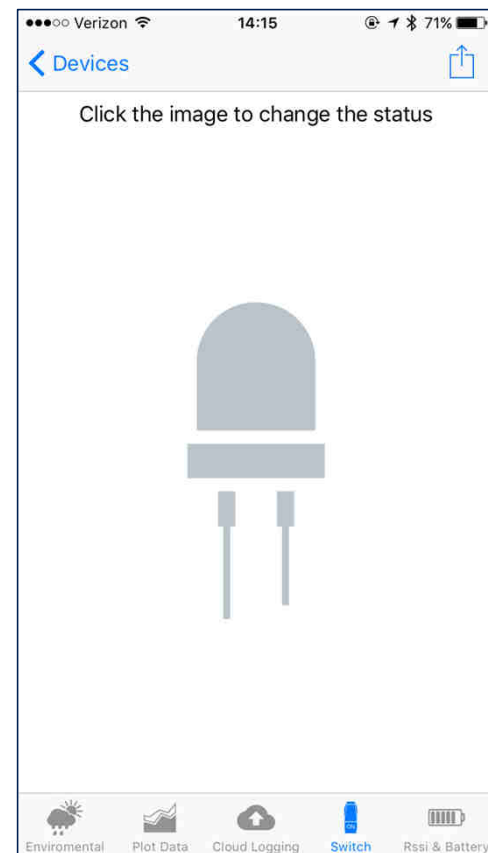
# Enabling LED Toggling

104

Click on “Switch”

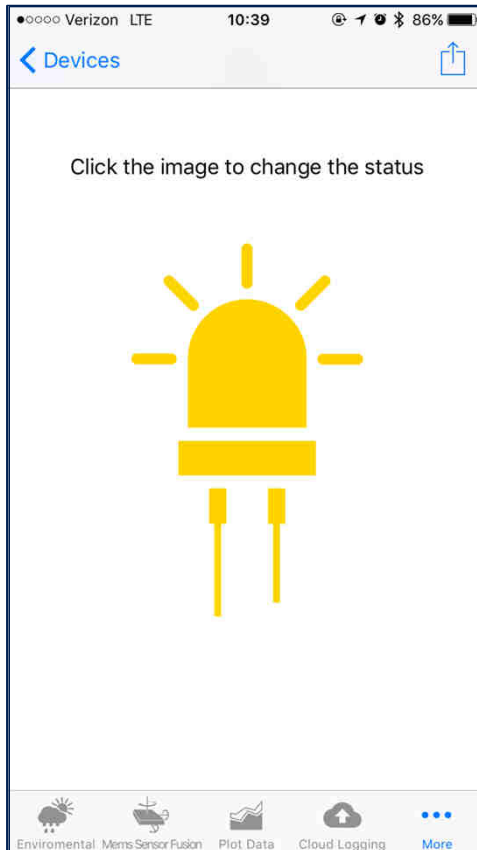


A new page will appear



# Enable LED Toggling

105

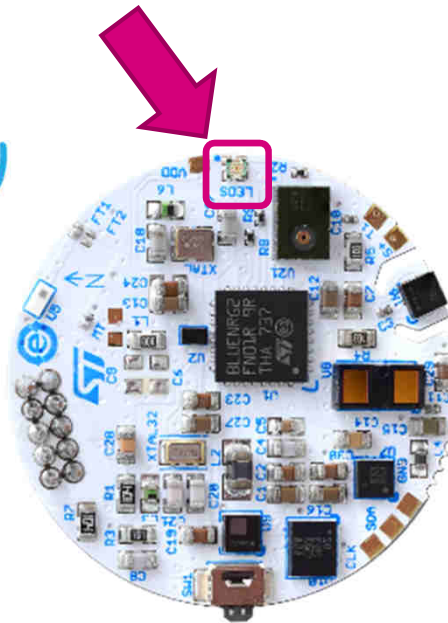


**Write  
Packet**

'1' LED ON  
'0' LED OFF

**Notification  
Packet**

'1' LED ON  
'0' LED OFF

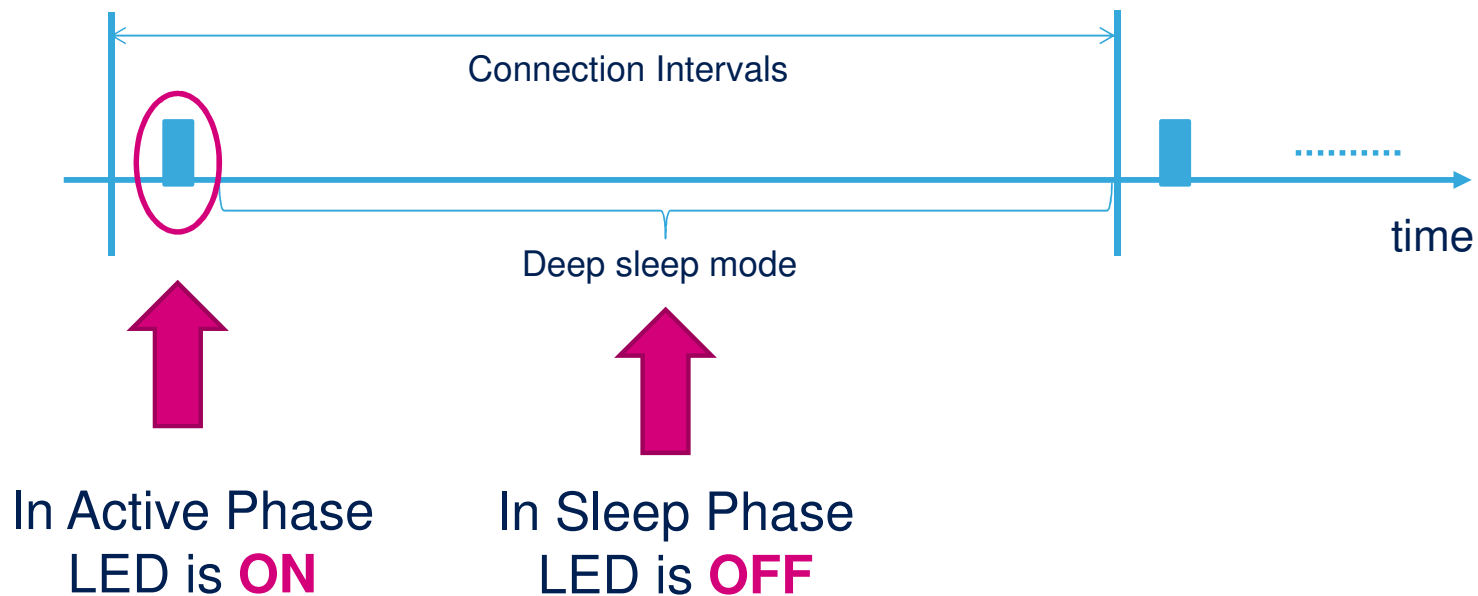


1. Tap the image on the screen
2. Send **BLE notification packet**
3. Image changes color and **RED LED toggles**

# LED Fast Blinking Due to Sleep Mode

106

- **Red LED** fast toggling shows the entering/exiting to/from Sleep mode



# Tera Term Output

107

- String “**LED Notification ON**” will appear as the app tab is enabled
- Each time user **click on the LED icon** in the **BLE Sensor app**, the **Write command is sent** to the board and the **LED toggles** accordingly (“Enabled: RGB led” = ON, “Disabled: RGB led” = OFF)

```
Device is now discoverable with MAC: 3f:2c:f6:eb:da:d8
Sensor activated: OK
Device connected
Environmental Notification ON
Environmental Notification OFF
Led Notification ON
Enabled: RGB led
Disabled: RGB led
Enabled: RGB led
Disabled: RGB led
Enabled: RGB led
Disabled: RGB led

```



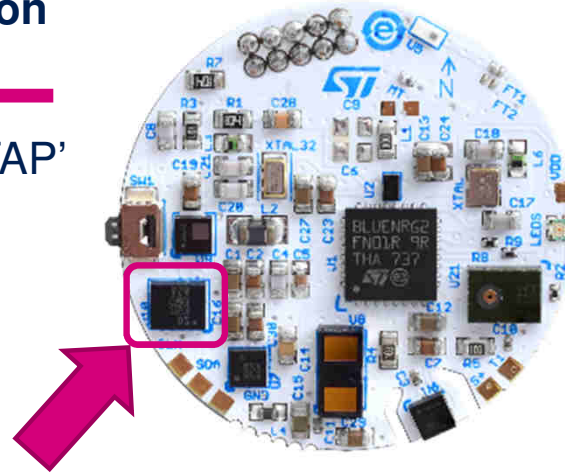
# Lab 4

## Accelerometer Embedded Events Detection

# Example - Single Tap

109

1. On the board **perform a Single Tap**
2. Send **BLE notification packet**



**NOTE:** this is just an example for SingleTap. Other events will be displayed later in the lab!

4 Bytes

Len	Type	Protocol version	Device ID	Feature Mask	Device MAC	Len	Type	Local Name	Len	Type	Pwr
-----	------	------------------	-----------	--------------	------------	-----	------	------------	-----	------	-----

31	30	29	28	27	26	25	24	0x20
RFU	ADPCM	Switch	DoA	ADPC	MicLevel	Proximity	Lux	
23	22	21	20	19	18	17	16	0x9E
Acc	Gyro	Mag	Pressure	Humidity	Temperat	Battery	2nd Temp	
15	14	13	12	11	10	9	8	0x04
RFU	RFU	RFU	RFU	Beamform	AccEvent	FreeFall	SensFusC	
7	6	5	4	3	2	1	0	0x00
SensFus	Compass	MotionInt	Activity	Carry Pos	MemsGes	ProxGes	Pedo	



# LSM6DSO at a Glance

111

## 1. PERFORMANCE AND FEATURE

- **Accuracy:** Tango and Daydream compliance
- **0.55mA** current consumption is **HP combo** -15% vs. LSM6DSL/M @ same performance
- New **ultra low power mode:** **14uA (@100Hz ODR) Axl only**



## 3. NEW STANDARD

### • I3C



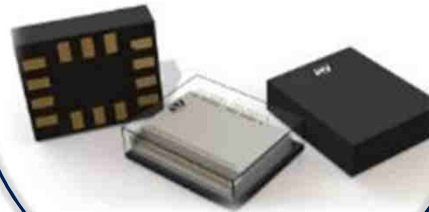
- FIFO tag

## 2. FLEXIBILITY: algorithm @ silicon level



- **Pedometer 2.x**  
WeChat Precision

## LSM6DSO



2.5x3x0.86 mm



- **FSM** build custom sensors for XL and Gyro

## 4. INNOVATION

- MEMS: **Axl T-structure** for mechanical shock robustness
- **Compressed FIFO (3x)**
- **10 patents** protecting LSM6DSO innovation



# Client Enables Accelerometer Events

## Characteristic Notifications

112

1. In the file **gatt\_db.c** go to the **line 703**
2. **Client writes** in the **Acc Event Characteristic Client Configurator Descriptor (CCCD)** and **Server enables notifications** through the **xFeatureNotification** structure in **line 714** by **xFeatureNotification.AccEventNotification = true**

```
705 } else if (attr_handle == (AccEventCharHandle + 2)) {  
706     if (xFeatureNotification.MotionNotification == false && xFeatureNotification.iNemoEngineNotification == false) {  
707         if (att_data[0] == 01) {  
708             lsm6dso_xl_data_rate_set(0, LSM6DSO_XL_ODR_52Hz);  
709             GPIO_writeBit(GPIO_Pin_7, Bit_RESET); // Proximity OFF  
710             EnableHWMultipleEvents();  
711             ResetHWPedometer();  
712             GPIO_EXTICmd(GPIO_Pin_13, ENABLE);  
713             Config_Notify(FEATURE_MASK_ACC_EVENTS, 'm', 1);  
714             AccEvent_Notify(0, 0);  
715             xFeatureNotification.AccEventNotification = true;  
716             PRINTF("AccEvent Notification ON\n\r");  
717         } else if (att_data[0] == 0) {  
718             lsm6dso_xl_data_rate_set(0, LSM6DSO_XL_ODR_OFF);  
719             xFeatureNotification.AccEventNotification = false;  
720             GPIO_EXTICmd(GPIO_Pin_13, DISABLE);  
721         }  
722     }  
723 }
```

# Read Event Status and Send BLE Notification

113

```
1299 void MEMSCallback(void) {
1300
1301     lsm6dso_all_sources_get(0, &all_source);

1334 /* Check if the interrupt is due to Single Tap */
1335 if (xHardwareFeaturePresence.HwSingleTAP || xHardwareFeaturePresence.MultipleEvent) {
1336     if (all_source.reg.all_int_src.single_tap)
1337         SdkEvalLedOn(LED1);
1338         SdkEvalLedOn(LED3);
1339         PRINTF("Event: Single Tap\n\r");
1340         AccEvent_Notify(ACC_SINGLE_TAP, 2)
1341     }
1342 }
1343
```

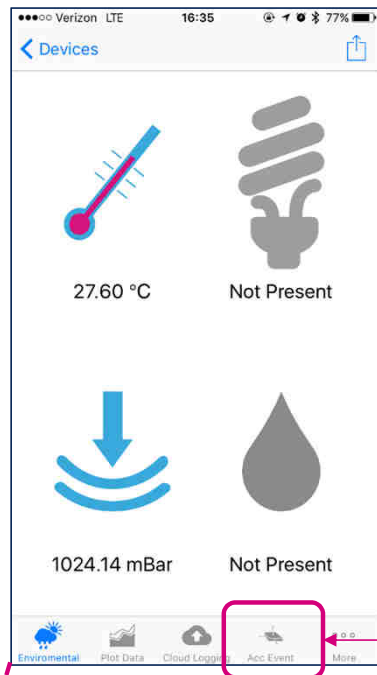
In the file **sensor.c**

1. **line 1299 void MEMSCallback(void)**  
Callback triggered by IO13
2. **line 1301 lsm6dso\_all\_sources\_get**  
Read accelerometer status registers
3. **line 1336 if (all\_source.reg.all\_int\_src.single\_tap)**  
Check vs. single tap event
4. **line 1340 AccEvent\_Notify**  
Send BLE notification

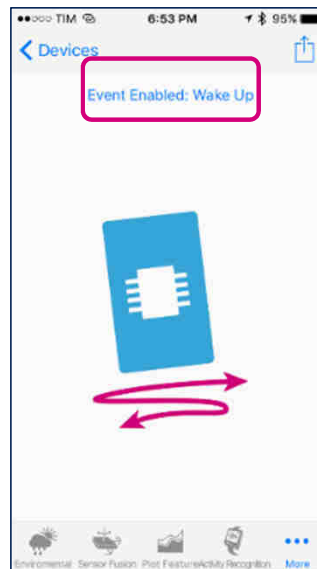
# Visualize Single Tap Event in ST BLE Sensor

114

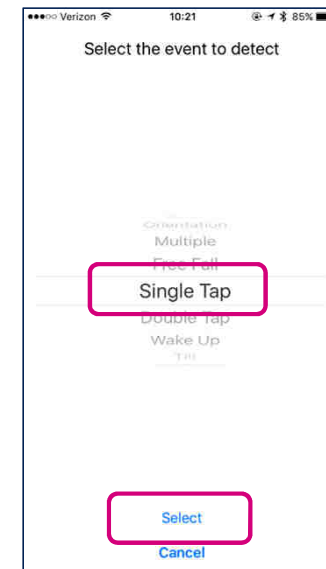
Select  
“Acc Event”



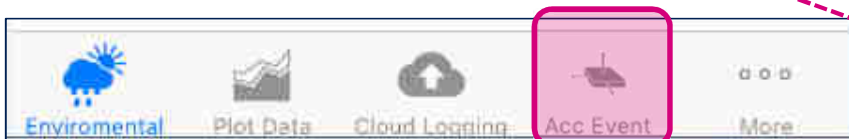
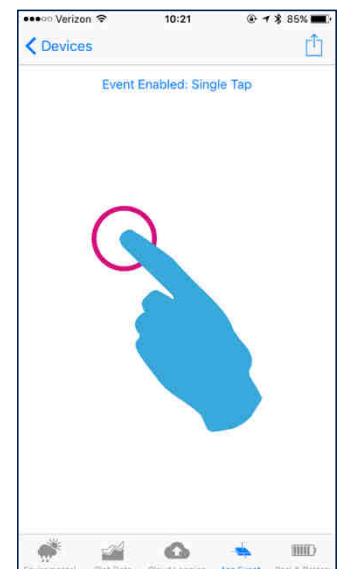
Tap  
“Event Enabled”



Select  
“Single Tap”  
and hit “Select”



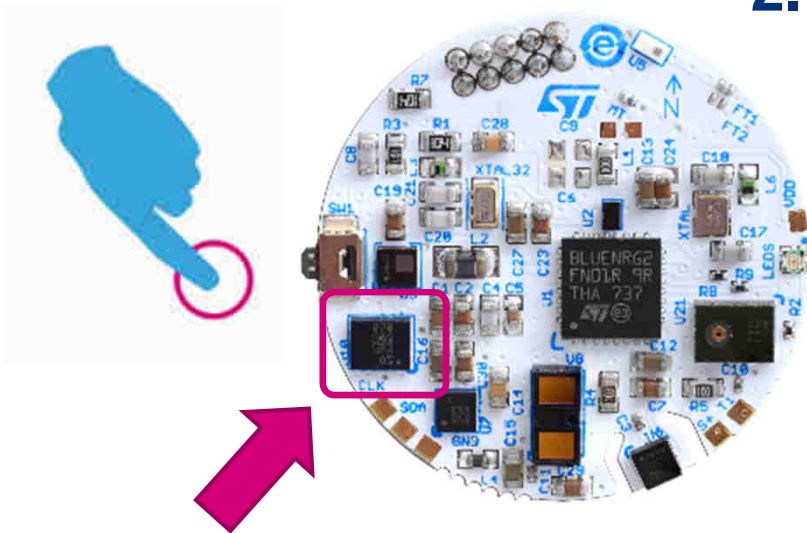
Enabled  
“Single Tap”



# Event Detection

115

1. On the board **perform a Single Tap**
2. Send **BLE notification packet**



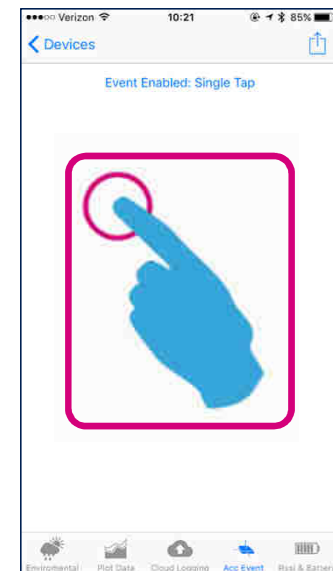
Tap gently on the board



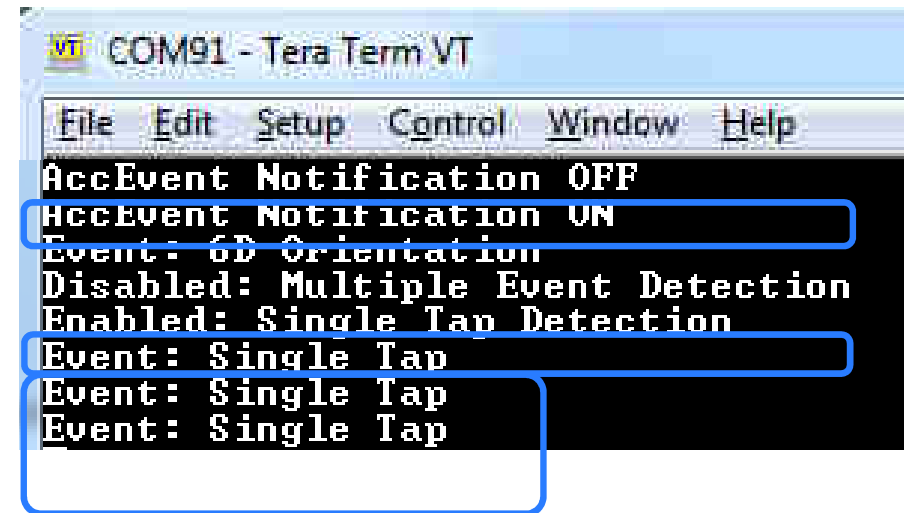
Notification  
Packet

→  
'SINGLE TAP'

## Single Tap



- **“AccEvent Notification ON”**: enable notifications on the Accelerometer Event characteristic
- **“Enabled Single Tap Detection”**: user enables Single Tap event detection
- **“Event: Single Tap”**: the actual BLE notification packet is sent upon detection of the Single Tap event.



The screenshot shows the Tera Term VT interface with the title bar 'COM91 - Tera Term VT'. The menu bar includes File, Edit, Setup, Control, Window, and Help. The output text is as follows:

```
AccEvent Notification OFF
AccEvent Notification ON
Event: 6D Orientation
Disabled: Multiple Event Detection
Enabled: Single Tap Detection
Event: Single Tap
Event: Single Tap
Event: Single Tap
```

Blue rectangular boxes highlight the following lines in the output: 'AccEvent Notification ON', 'Event: 6D Orientation', 'Enabled: Single Tap Detection', and the three 'Event: Single Tap' lines.

NOTE: if the phone display enters sleep mode, the notifications are not sent

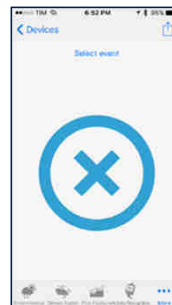
# LSM6DSO Embedded Events

117

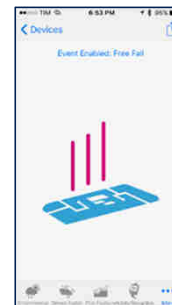
Other possible  
embedded events



No event



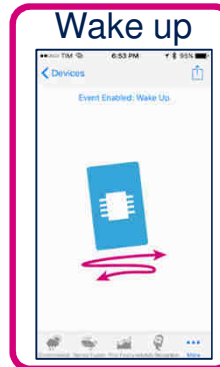
Free fall



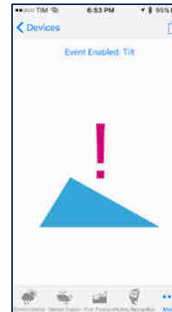
Single tap



Wake up



Tilt



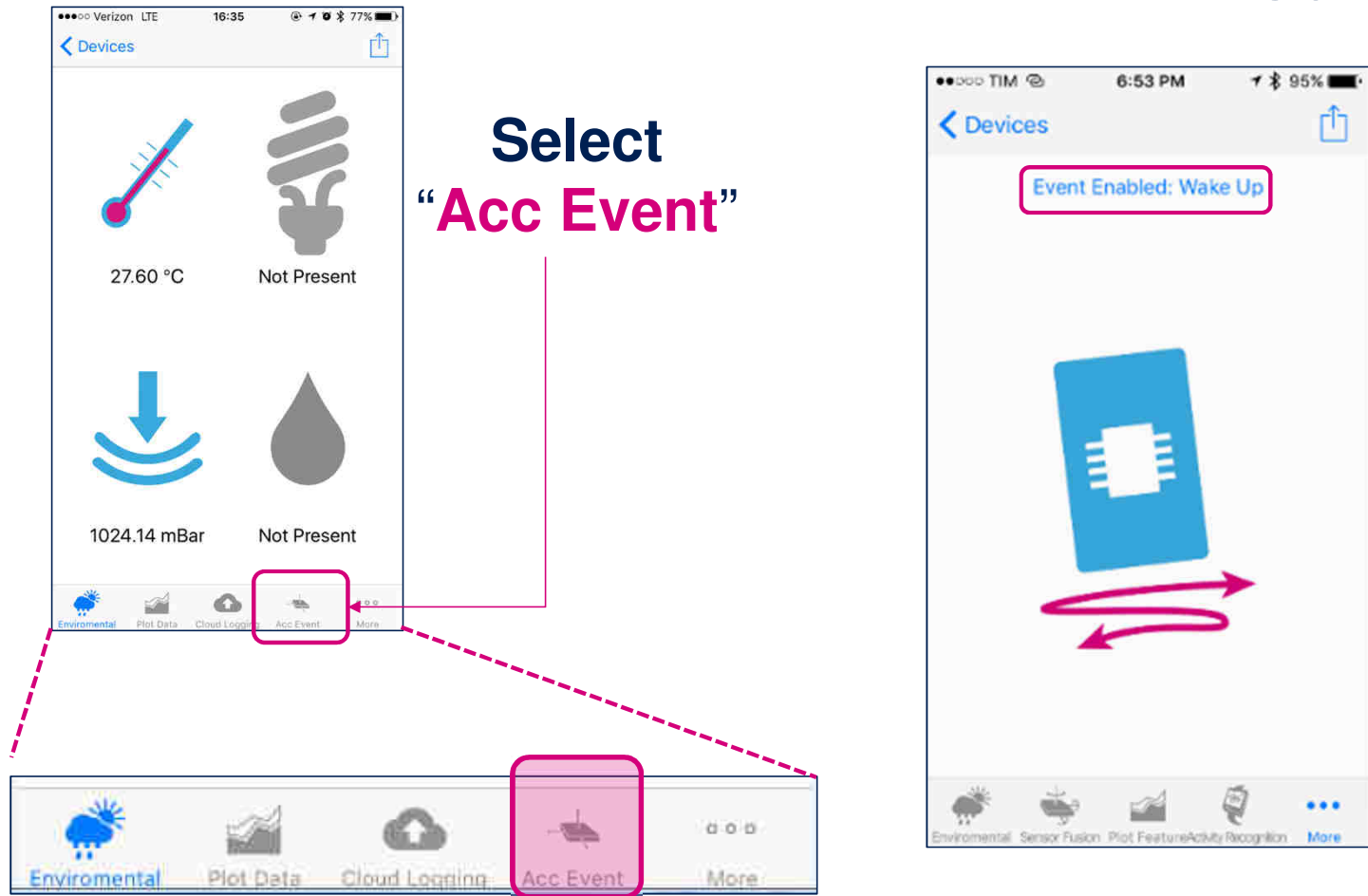
Pedometer



WakeUp  
example

# Visualize Hardware Wakeup Event in BLE Sensor

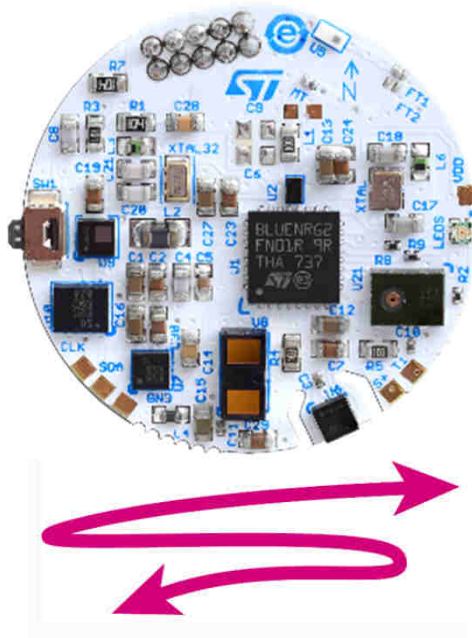
118



# Event Detection

119

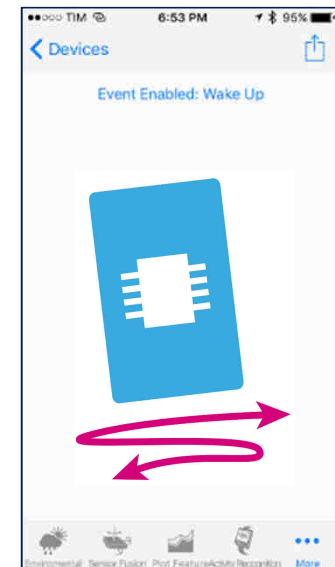
1. **Leave** the board **still** for a few seconds
2. **Shake** the board
3. Send **BLE notification packet**



Notification  
Packet

→  
'WAKE UP'

## Wake Up

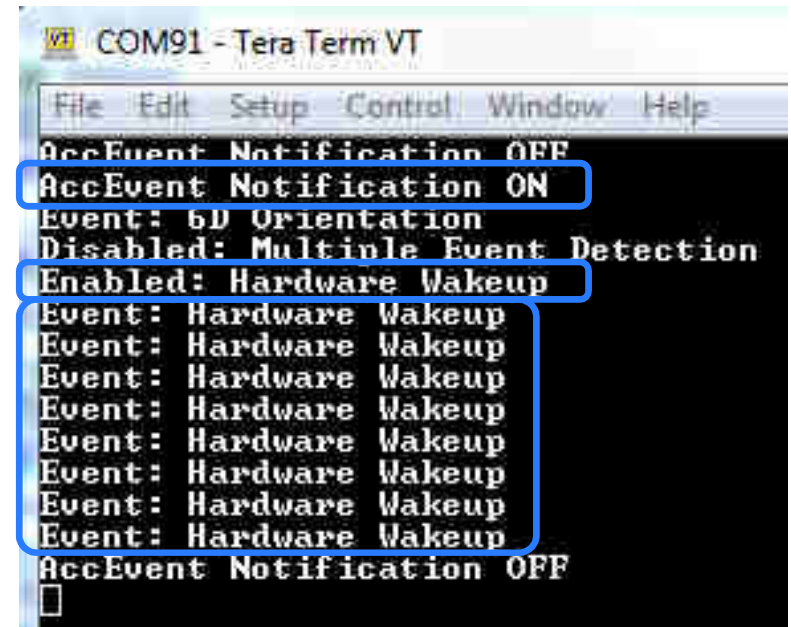




# Tera Term Output

120

- **“AccEvent Notification ON”**: enable notifications on the Accelerometer Event characteristic
- **“Enabled Hardware Wakeup”**: user enable hardware wakeup event detection
- **“Event: Hardware Wakeup”**: the actual BLE notification packet sent upon detection of the Hardware Wakeup event.



The screenshot shows a Tera Term window titled 'COM91 - Tera Term VT'. The output text is as follows:

```
AccEvent Notification OFF
AccEvent Notification ON
Event: 6D Orientation
Disabled: Multiple Event Detection
Enabled: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
AccEvent Notification OFF
```

Blue rectangular boxes highlight the following lines in the output:

- AccEvent Notification ON
- Enabled: Hardware Wakeup
- The entire block of eight 'Event: Hardware Wakeup' lines.

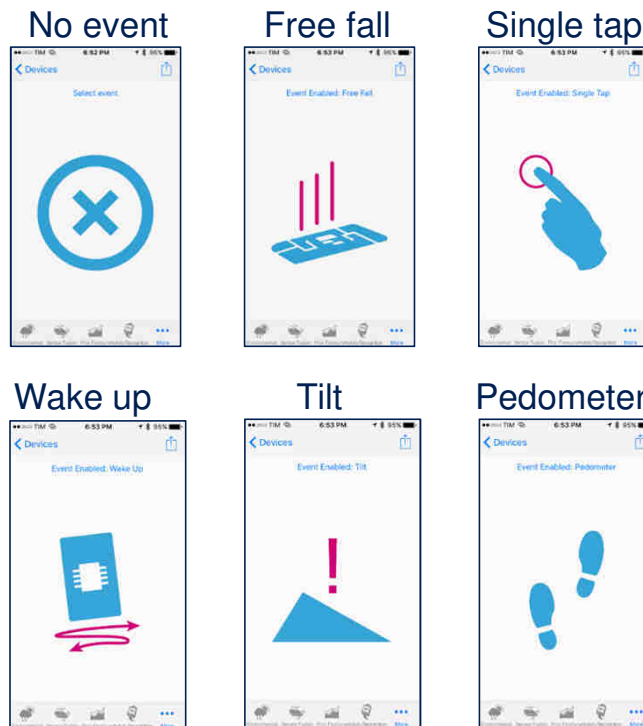
NOTE: if the phone display enters sleep mode, notifications are not sent

# LSM6DSO Embedded Events

121

You can go ahead later and test other events...

Other possible  
embedded events





# Lab 5

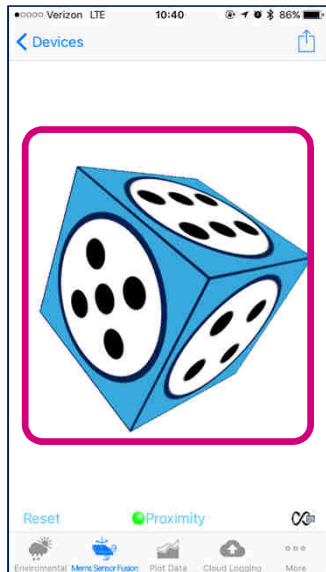
## 9-axis Acc+Gyro+Mag Sensor Data Fusion

- In this example we are going to demonstrate how to:
  - **Enable** the embedded sensor data fusion library
    - Input: raw Acc+Gyro+Mag sensor data
    - Output: quaternions
  - **Scale the quaternions value** by a scaling factor proportional to proximity detection
  - **Send** scaled data fusion information through **BLE notifications packets** to the ST BLE Sensor app client

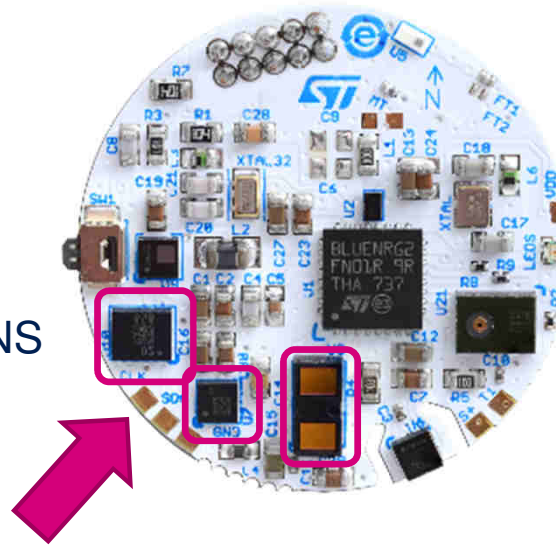
# Sensor Data Fusion

124

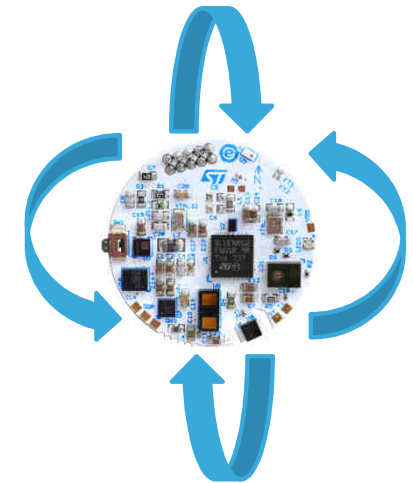
1. **Enable** Sensor Fusion library
2. **Send** quaternions information through **BLE notification packet**



  
**Notification  
Packet**  
←  
QUATERNIONS



**Sensors used by data fusion:**  
Acc+Gyro+Mag and Proximity



# Code Modifications

125

1. **Enable Sensor Data fusion library**
2. **Enable proximity sensor**
3. **Send quaternions data** – scaled by proximity sensor data - through **BLE notifications packets**

4 Bytes

Len	Type	Protocol version	Device ID	Feature Mask	Device MAC	Len	Type	Local Name	Len	Type	Pwr
-----	------	------------------	-----------	--------------	------------	-----	------	------------	-----	------	-----

31	30	29	28	27	26	25	24
RFU	ADPCM	Switch	DoA	ADPC	MicLevel	Proximity	Lux
23	22	21	20	19	18	17	16
Acc	Gyro	Mag	Pressure	Humidity	Temperat	Battery	2nd Temp
15	14	13	12	11	10	9	8
RFU	RFU	RFU	RFU	Beamform	AccEvent	FreeFall	SensFusC
7	6	5	4	3	2	1	0
SensFus	Compass	MotionInt	Activity	Carry Pos	MemsGes	ProxGes	Pedo

0x22

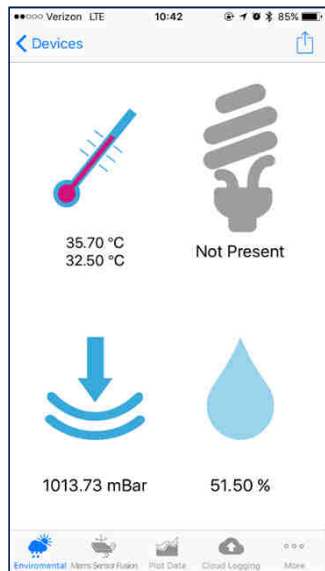
0xFE

0x05

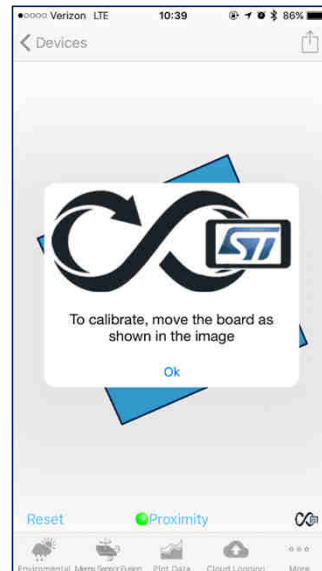
0x40

# Sensor Data Fusion 127

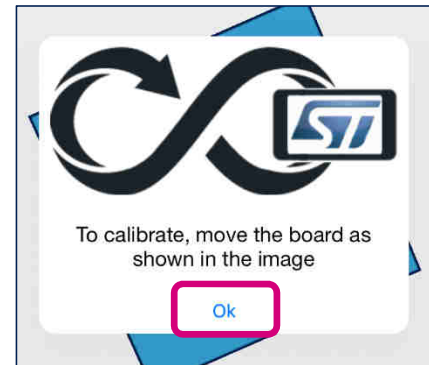
**Swipe left** to **view** the **Mems Sensor Fusion**



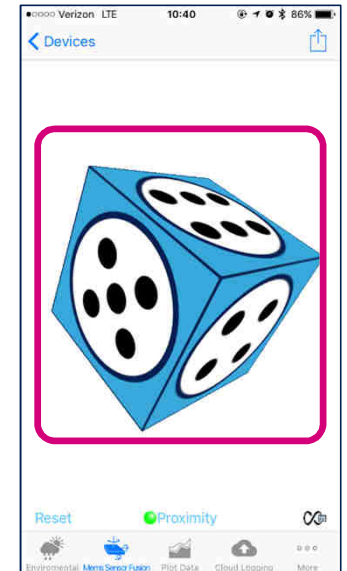
**“Mems Sensor Fusion”** tab



**Click on**  
**OK**



Sensor Fusion  
**enabled**

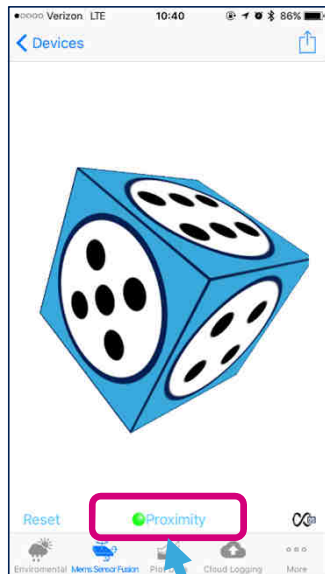




# Proximity Sensor

128

## Enable Proximity Sensor

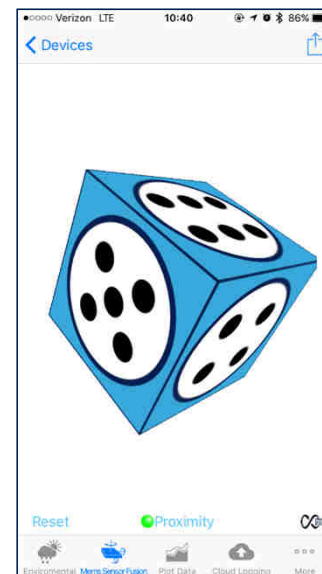


Move your hand far and close to the BlueNRG-Tile



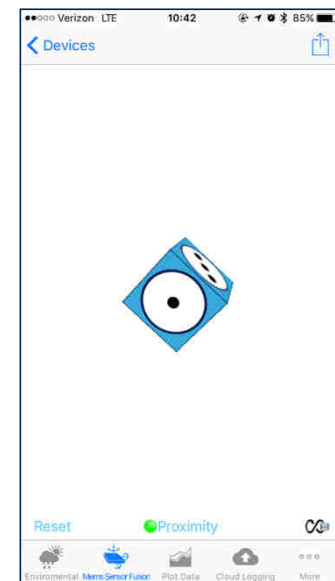
1

Far from BlueTile



2

Near to BlueTile



Note: ONLY for rendering purposes, the cube is scaled proportional to a clipped 30 cm distance



# Proximity Sensor - VL53L1X

129

3rd gen **ToF sensor** with lens for **long distance** ranging & ROI selection



## Product highlights

OLGA: 4.9 x 2.5 x 1.56 mm

FoV : 27°

Compatible footprint with VL53L0X

### Enhanced performances:

- Full FoV ranging : **400cm+** (white target, no IR)
- **Programmable FOV**: SPAD zone selection – full-screen to 4x4

### Cutting-edge module and silicon :

- **Fastest miniature ToF product in the market** up to 50Hz
- **Integrated lens** for longer range and better ambient light immunity
- **Programmable settings** to best fit customer's application:
  - Low power with interrupts for user / object detection
  - Long distance ranging
  - High accuracy for small movement detection

## Applications

### Presence user detection

- Autonomous mode with interrupts
- Low-power
- Long distance 400cm+
- PC, tablets, IoT, portable handsets, security



### Obstacle detection:

- Robots: Obstacle avoidance
- Vacuum cleaners: Wall following, cliff detection
- Drones: Take-off and landing, Ceiling detection

### Accurate objects distance scanning

- Vending machines: control of objects in racks
- Coins dispensers: coins counting
- Smart shelves: Consumer scanning



# Full-range Proximity Real-time Data Plot

130

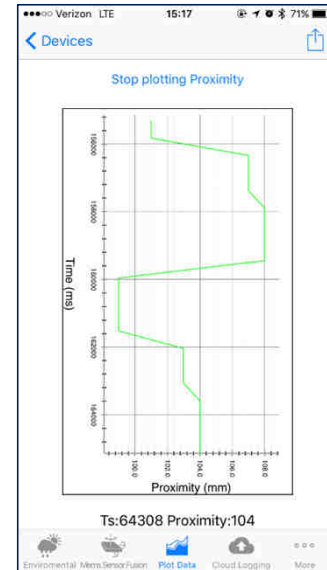
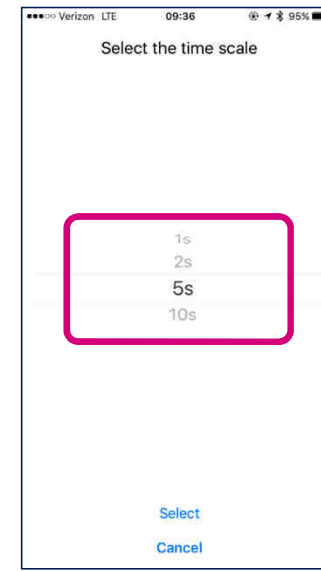
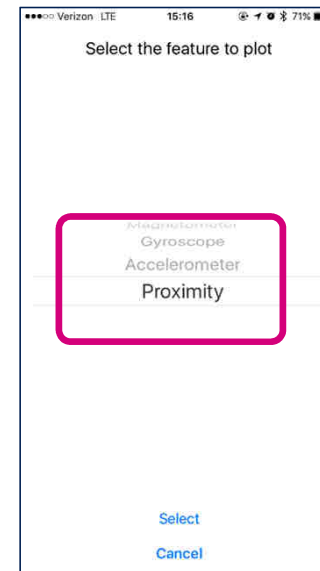
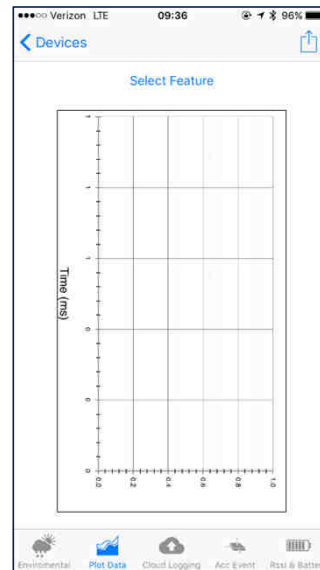
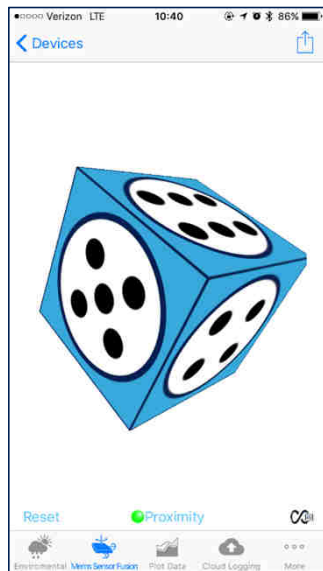
**Swipe left** to **view** the real-time data plot

**“Plot Feature”** tab

**Select Proximity**

**Select the time frame**

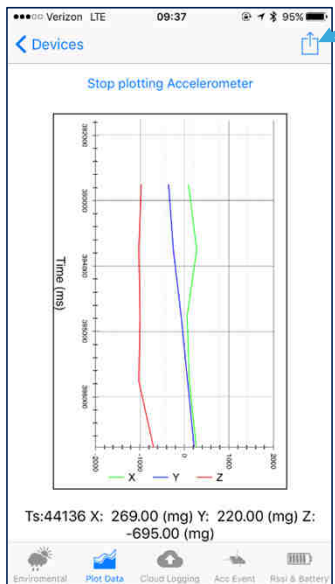
**View the real-time data plot**



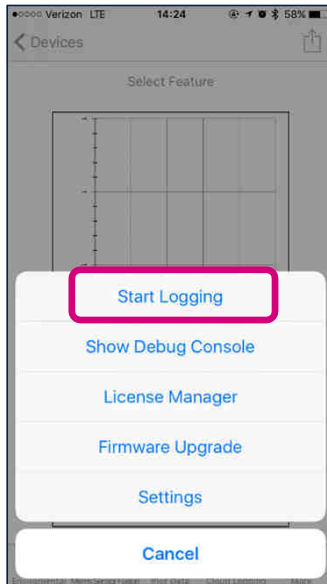
# Logging Real-time Data Plot

131

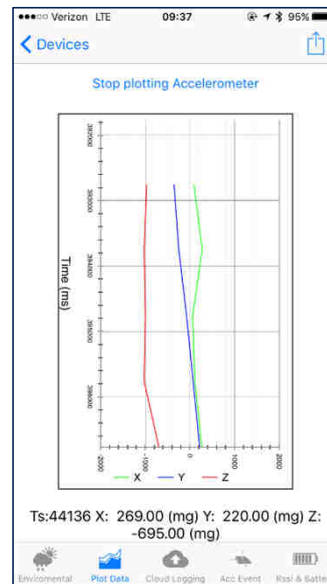
Click on  
top right corner



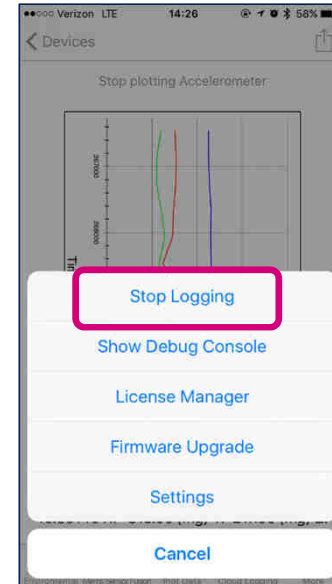
1  
Select  
“Start logging”



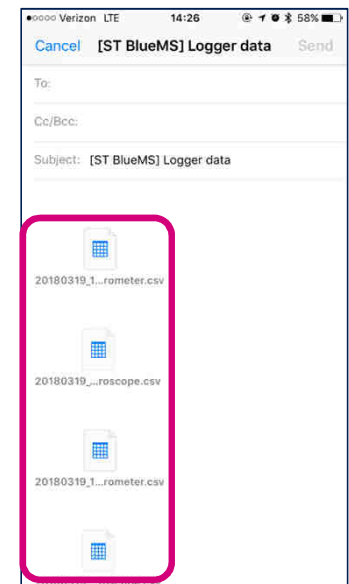
2  
After some time  
re-click on the  
right-top corner



3  
Select  
“Stop logging”



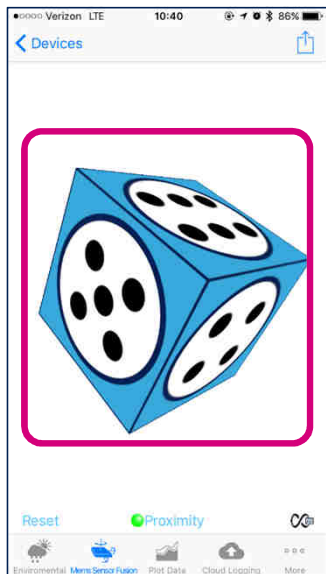
4  
App opens your  
email client



.csv files attached  
to the email

# Sensor Data Fusion 132

Sensor Fusion  
**enabled**



Try **static** position: no drift, the cube is perfectly still

- In steady static position the **acc** and **mag** enable an accurate orientation estimate
- acc vibrations and mag bias and interferences are **rejected or compensated by the motionFX library**

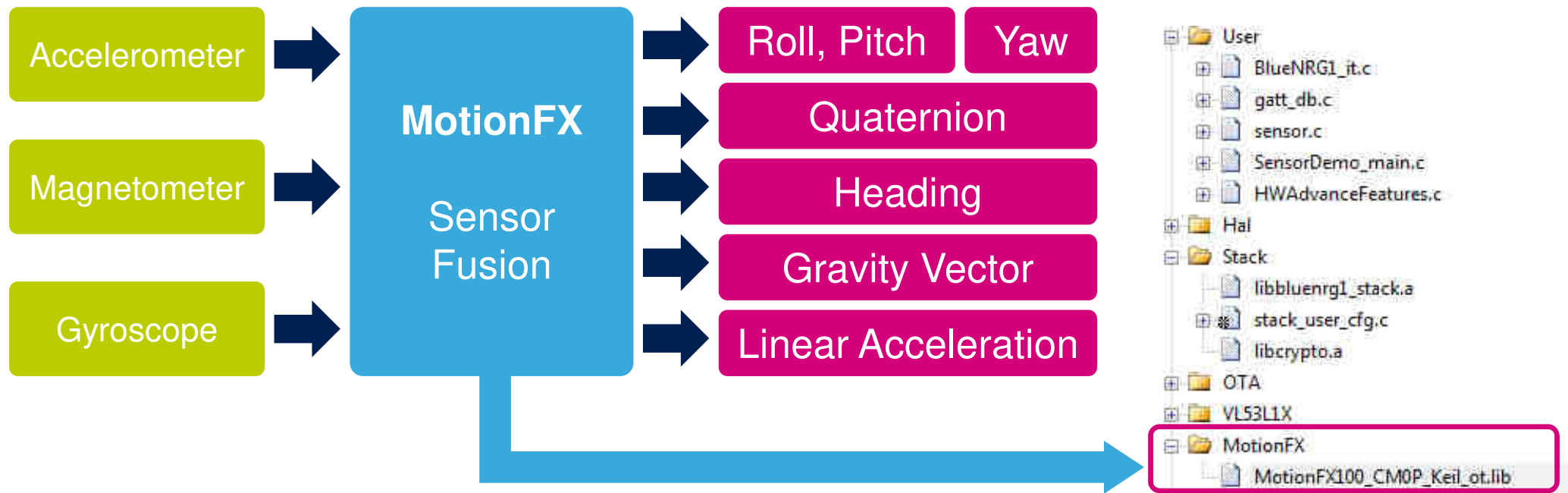
Try highly **dynamic** motion: perfect tracking

- In high dynamic motion the acc cannot be used, therefore the **gyro** is used to update the orientation
- gyro bias is **estimated and compensated at runtime** by the **motionFX library**

# Sensor Data Fusion

## MotionFX Library

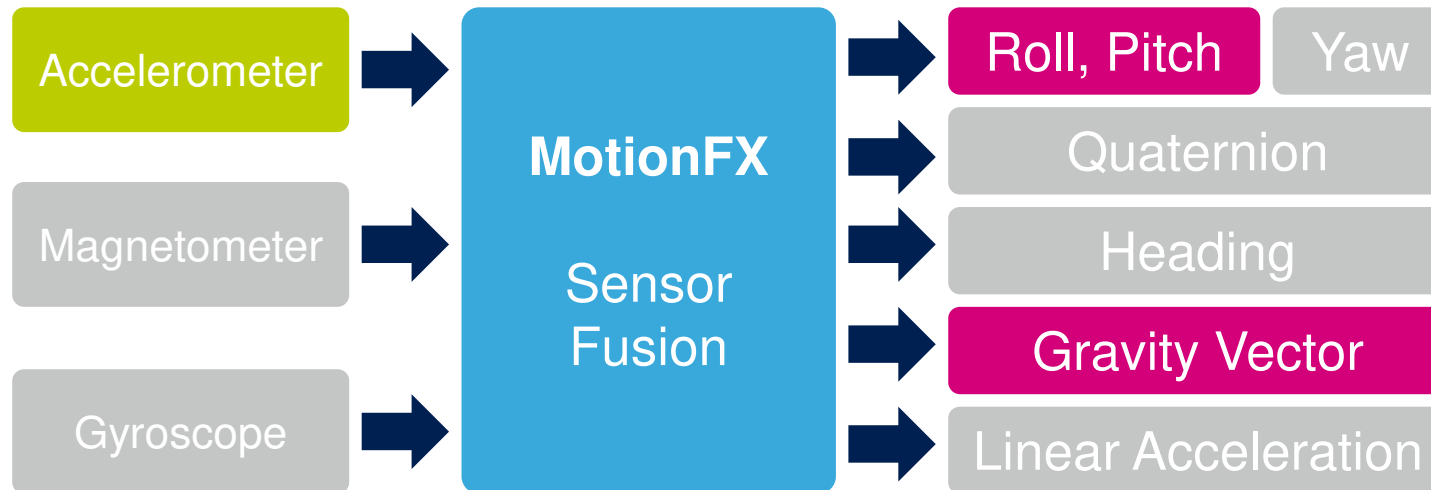
133



# Sensor Data Fusion

## MotionFX Library

134

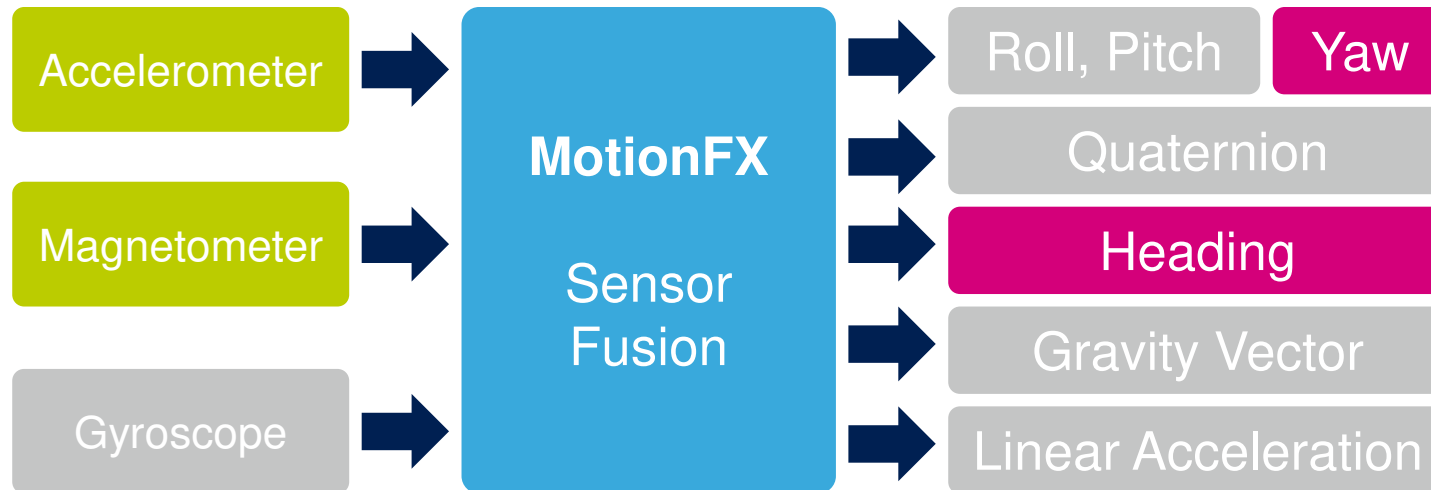


- **Accelerometer gives roll and pitch angles and the gravity vector...but only in static conditions! (or low dynamics)**

# Sensor Data Fusion

## MotionFX Library

135



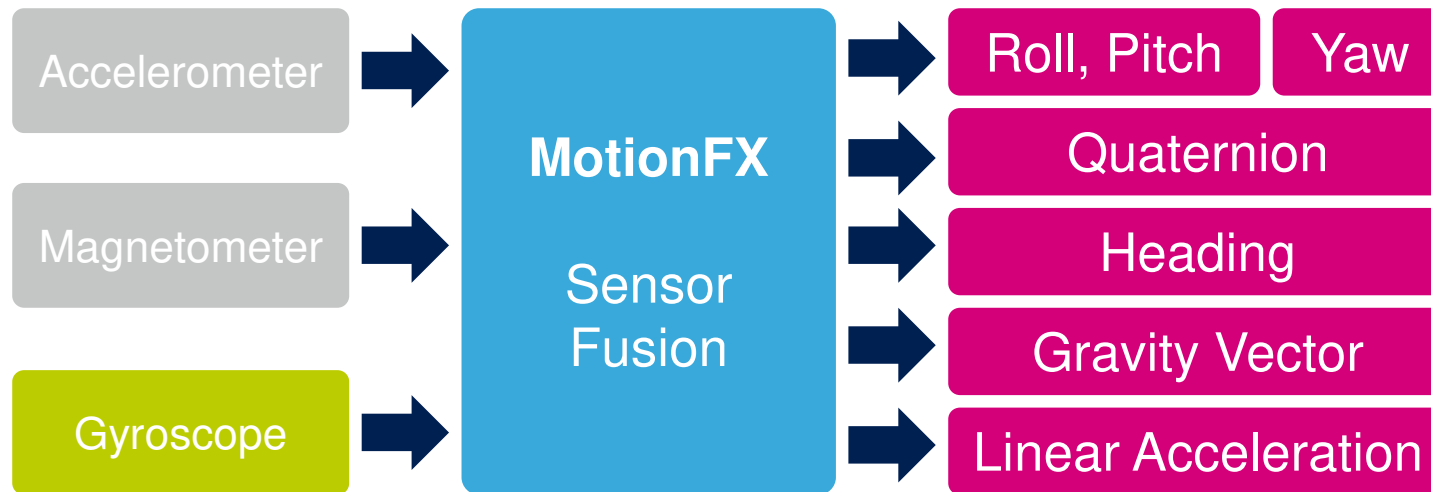
- **Magnetometer gives yaw angle and heading**  
...but **only if tilt is compensated**: the accelerometer is needed!  
...and mag bias (hard-iron) **must be compensated!**



# Sensor Data Fusion

## MotionFX Library

136

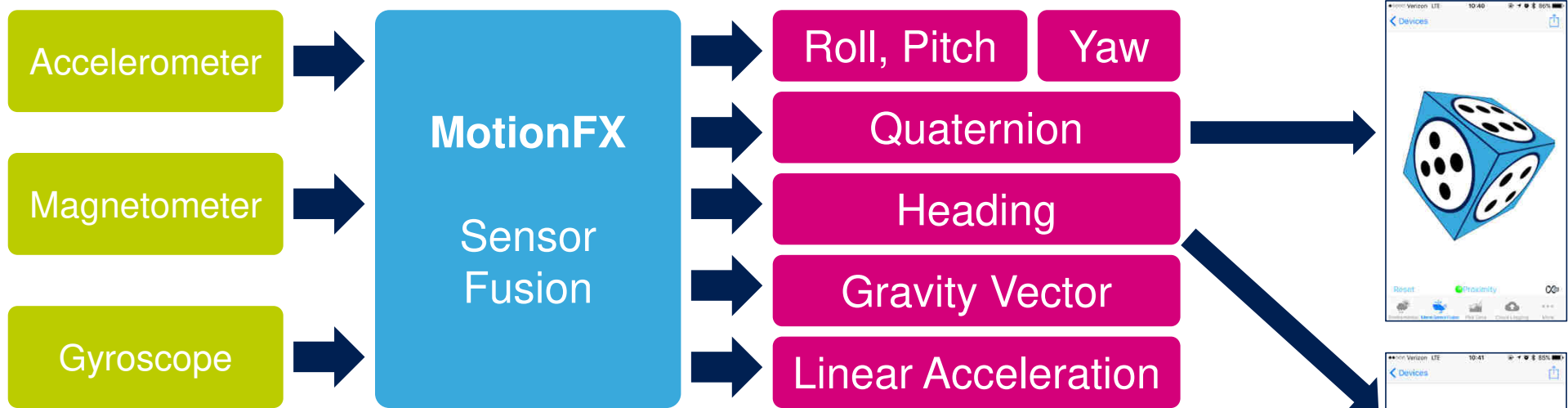


- **Gyroscope gives the new orientation** based on previous orientation ...but gyro bias offset needs to be compensated for accuracy!

# Sensor Data Fusion

## MotionFX Library

137



**Motion FX library provides orientation estimation plus:**

- **Magnetometer hard-iron offset** compensation.
- **Accelerometer vibrations** rejection.
- **Gyroscope bias** offset compensation.

**The gyroscope is the most important sensor in the system**  
 ...works in highly dynamic conditions when the Acc cannot be used  
 ...works with magnetic anomalies when the Mag cannot be used

# Gyroscope

138

**LSM6DSO** is an inertial module with **Acc** and **Gyro**

- **Exceptional noise performance** for maximum accuracy in orientation estimation
  - 90 ug/sqrtHz acc noise density
  - 3.8 mdps/sqrtHz gyro noise density
- **Extremely low-power**
  - **650 uA peak power** at ODR 6.66kS/sec with Acc+Gyro in high performance mode!


life.augmented

**LSM6DSO**

iNEMO inertial module:  
always-on 3D accelerometer and 3D gyroscope

Data brief



LGA-54L  
(2.5 x 3 x 0.83 mm) typ.

**Features**

- Power consumption: 0.55 mA in combo high-performance mode
- "Always-on" experience with low power consumption for both accelerometer and gyroscope
- Smart FIFO up to 9 kbyte
- Android compliant
- $\pm 2/\pm 4/\pm 8/\pm 16$  g full scale
- $\pm 125/\pm 250/\pm 500/\pm 1000/\pm 2000$  dps full scale
- Analog supply voltage: 1.71 V to 3.6 V
- Independent IO supply (1.62 V)
- Compact footprint: 2.5 mm x 3 mm x 0.83 mm
- SPI / I<sup>2</sup>C & MIPI (CSI<sup>SM</sup>) serial interface with main processor data synchronization
- Auxiliary SPI for OIS data output for gyroscope and accelerometer
- Advanced pedometer, step detector and step counter
- Significant Motion Detection, Tilt detection
- Standard interrupts: free-fall, wakeup, 6D/4D orientation, click and double-click
- Programmable finite state machine: accelerometer, gyroscope and external sensors
- Embedded temperature sensor
- ECOMPACT<sup>®</sup>, RoHS and "Green" compliant

**Description**

The LSM6DSO is a system-in-package featuring a 3D digital accelerometer and a 3D digital gyroscope boosting performance at 0.55 mA in high-performance mode and enabling always-on low-power features for an optimal motion experience for the consumer.

The LSM6DSO supports main OS requirements, offering real, virtual and batch sensors with 9 kbytes for dynamic data batching. ST's family of MEMS sensor modules leverages the robust and mature manufacturing processes already used for the production of micromachined accelerometers and gyroscopes. The various sensing elements are manufactured using specialized micromachining processes, while the IC interfaces are developed using CMOS technology that allows the design of a dedicated circuit which is trimmed to better match the characteristics of the sensing element.

The LSM6DSO has a full-scale acceleration range of  $\pm 2/\pm 4/\pm 8/\pm 16$  g and an angular rate range of  $\pm 125/\pm 250/\pm 500/\pm 1000/\pm 2000$  dps.


The LSM6DSO fully supports EIS and OIS applications as the module includes a dedicated configurable signal processing path for OIS and auxiliary SPI, configurable for both the gyroscope and accelerometer.

High robustness to mechanical shock makes the LSM6DSO the preferred choice of system designers for the creation and manufacturing of reliable products. The LSM6DSO is available in a plastic land grid array (LGA) package.

Table 1. Device summary

## LIS2MDL is digital Mag


- **Exceptional noise performance and dynamic range** for maximum accuracy in orientation estimation
  - 50 Gauss dynamic range
  - 3 mgauss RMS noise
- **Embedded offset compensation**
  - **Intrinsic offset is estimated and compensated automatically**
  - Extrinsic offset (hard-iron) must be estimated by host and can be compensated internally


**LIS2MDL**

---

Digital output magnetic sensor:  
ultra-low-power, high-performance 3-axis magnetometer

Datasheet - production data



LGA-12 (2.0x2.0x0.7 mm)

### Description

The LIS2MDL is an ultra-low-power, high-performance 3-axis digital magnetic sensor. The LIS2MDL has a magnetic field dynamic range of  $\pm 50$  gauss.

The LIS2MDL includes an I<sup>2</sup>C serial bus interface that supports standard, fast mode, fast mode plus, and high-speed (100 kHz, 400 kHz, 1 MHz, and 3.4 MHz) and an SPI serial standard interface.

The device can be configured to generate an interrupt signal for magnetic field detection.

The LIS2MDL is available in a plastic land grid array package (LGA) and is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

### Features

- 3 magnetic field channels
- $\pm 50$  gauss magnetic dynamic range
- 16-bit data output
- SPI/I<sup>2</sup>C serial interfaces
- Analog supply voltage 1.71 V to 3.6 V
- Selectable power mode/resolution
- Single measurement mode
- Programmable interrupt generator
- Embedded self-test
- Embedded temperature sensor
- ECOPACK®, RoHS and "Green" compliant

### Applications

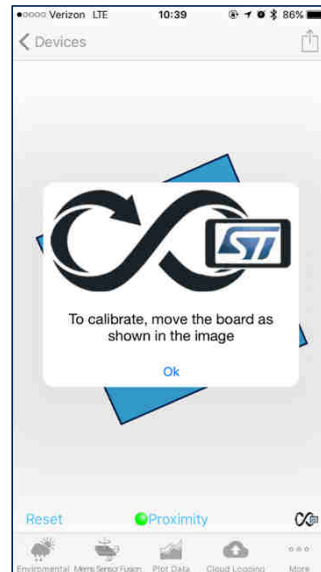
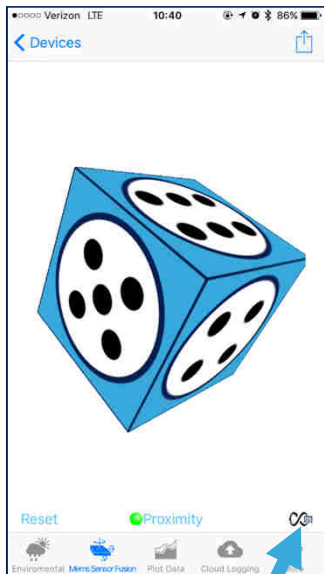
- Tilt-compensated compasses
- Map rotation
- Intelligent power saving for handheld devices
- Gaming and virtual reality input devices

### Table 1. Device summary

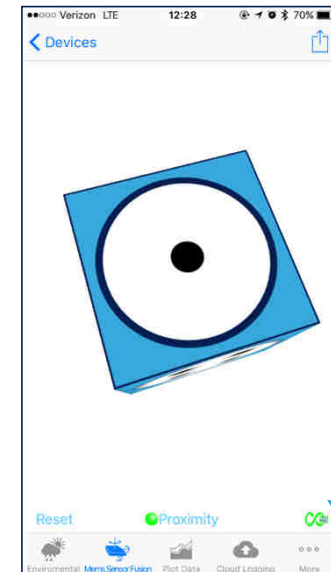
Part number	Temp. range [°C]	Package	Packaging
LIS2MDL	-40 to +85	LGA-12	Tray
LIS2MDLTR	-40 to +85	LGA-12	Tape and reel

# Magnetometer Calibration

140



**Move** the BlueTile with the “**8 pattern**” shown in the figure to calibrate the magnetometer

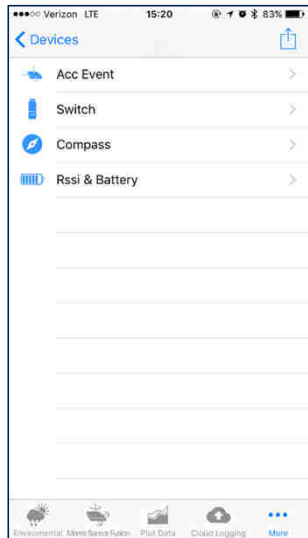


Calibration is **Completed** when the icon becomes **green**.

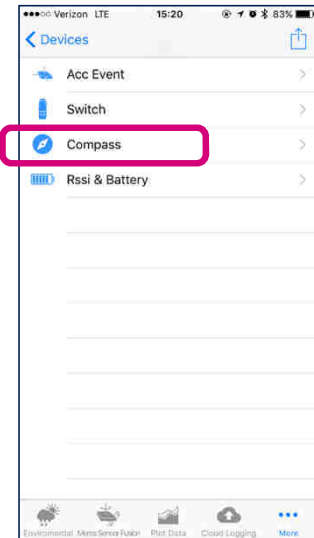


# Magnetometer ECompass

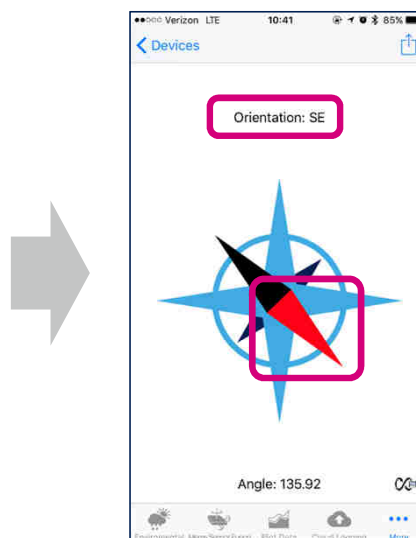
141



Tap

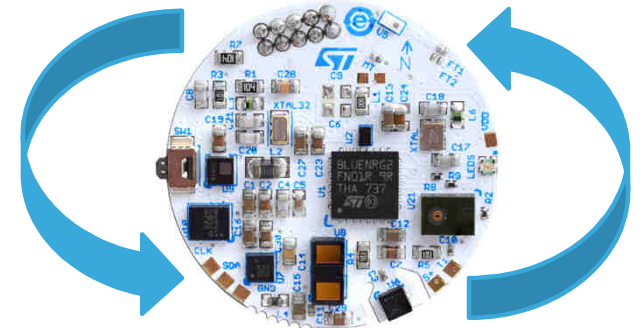


Select  
“Compass”

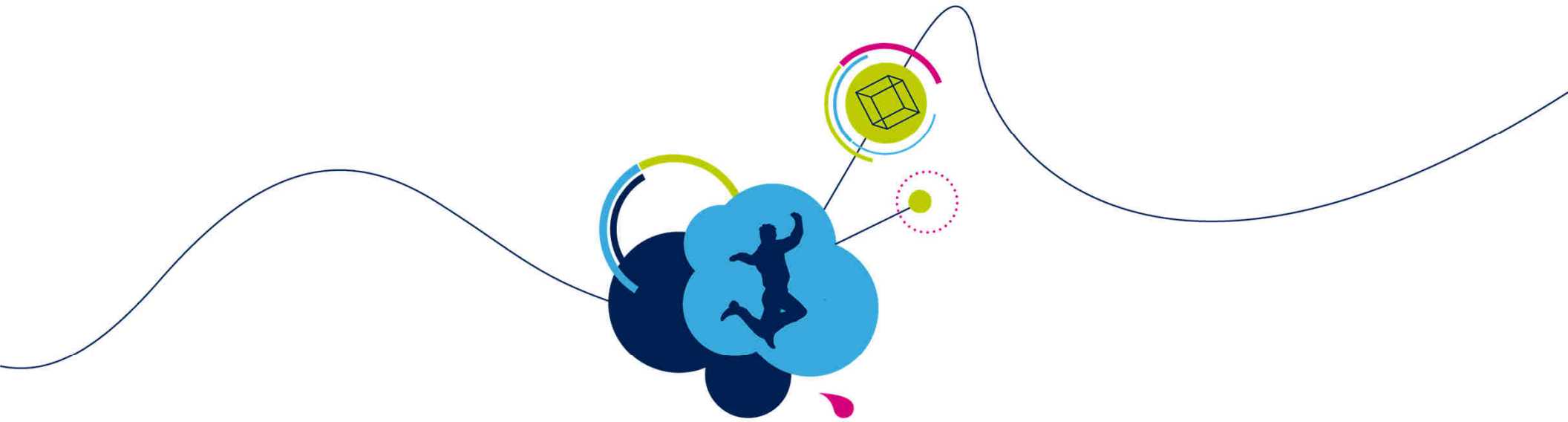


Red arrow is related to  
the current orientation  
of the BlueTile

Rotate the BlueTile



(you can check against the  
phone eCompass)



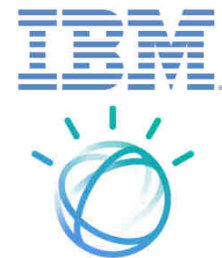
# Lab 6

## Cloud Data Logging on IBM Watson

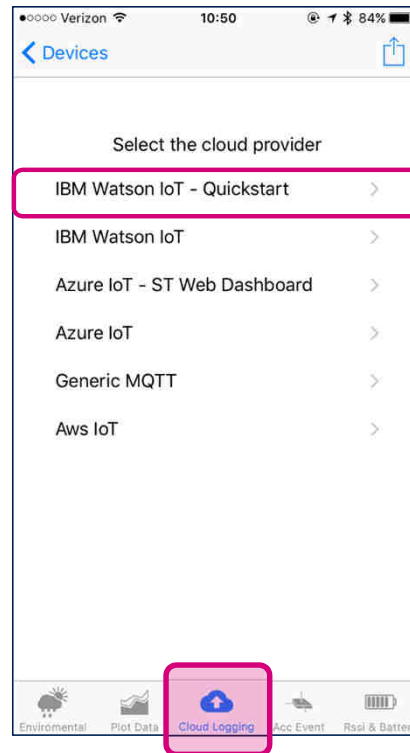
# Cloud Logging

143

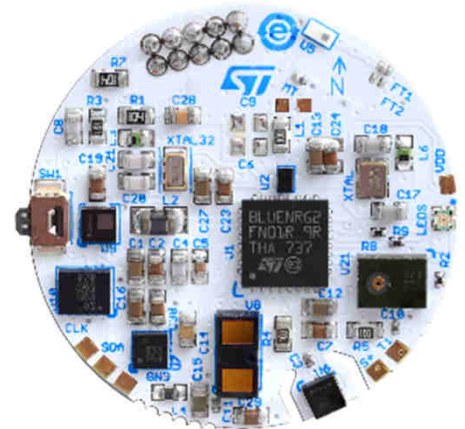
1. Enable **IBM Watson IoT – Cloud Logging**
2. Send **BLE** notification packets on Sensor status
3. **Visualize** the data



IBM Watson



Notification  
Packets

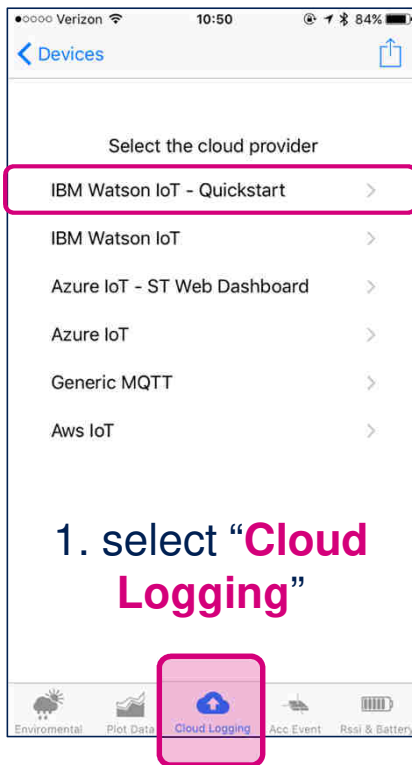




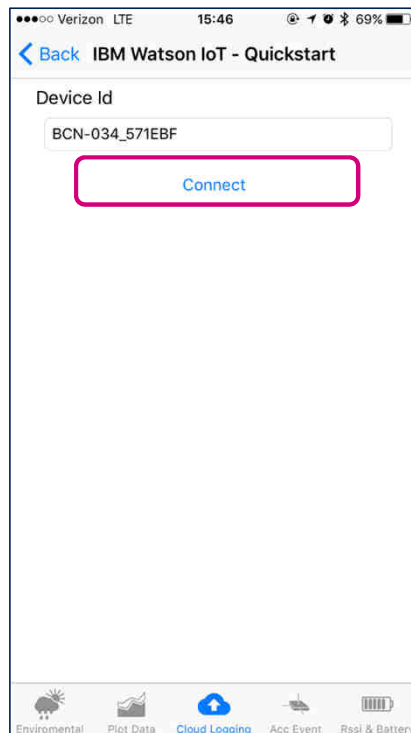
# IBM Watson IoT Quickstart

144

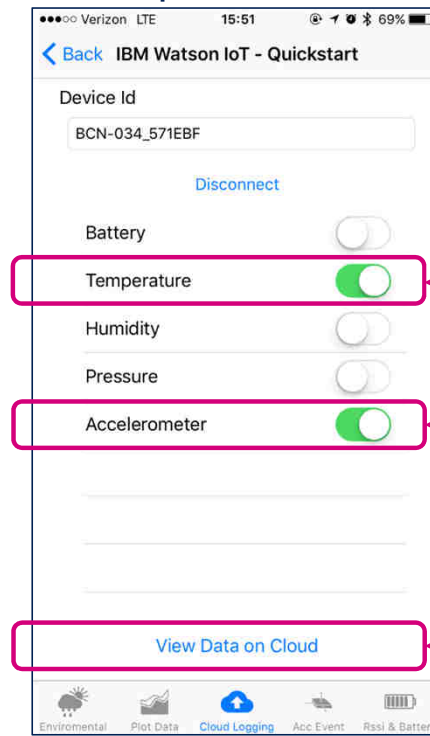
## 2. Select "IBM Watson IoT - Quickstart"



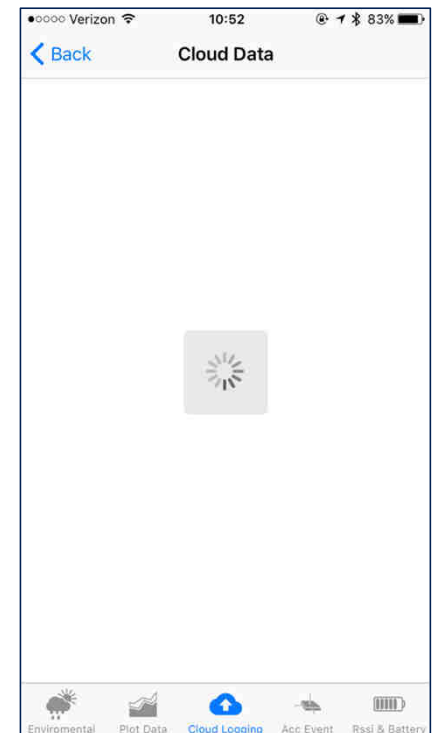
## Click "Connect"



## 1. Select one or multiple features



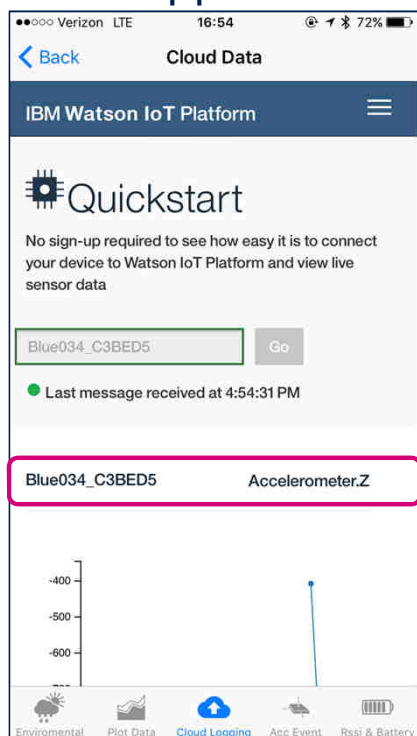
## Wait a few seconds



# IBM Watson IoT Quickstart

145

Quickstart will appear



Scroll down to see your selected sensor plot.

You will see the Plot of selected feature

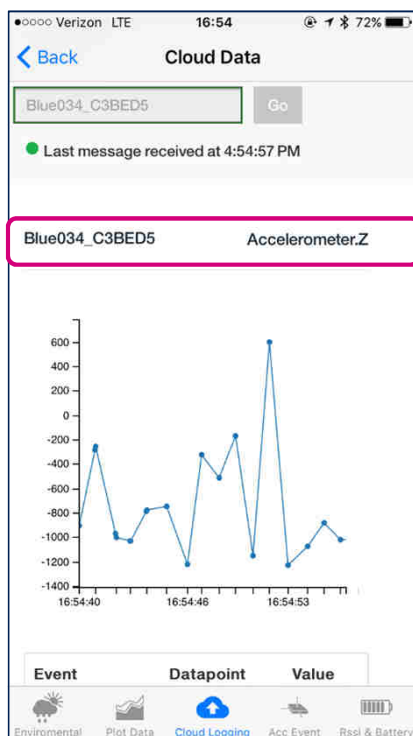
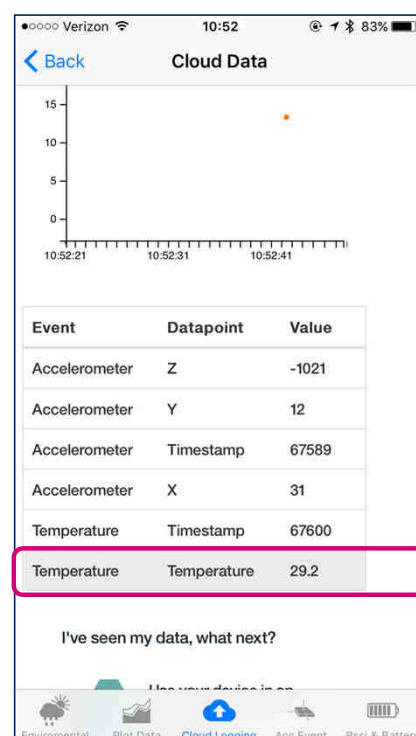
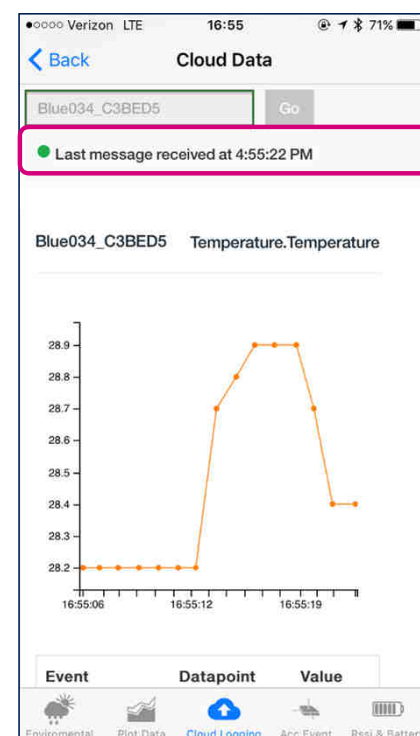


Table of available features



Scroll down again to change sensor data or axes



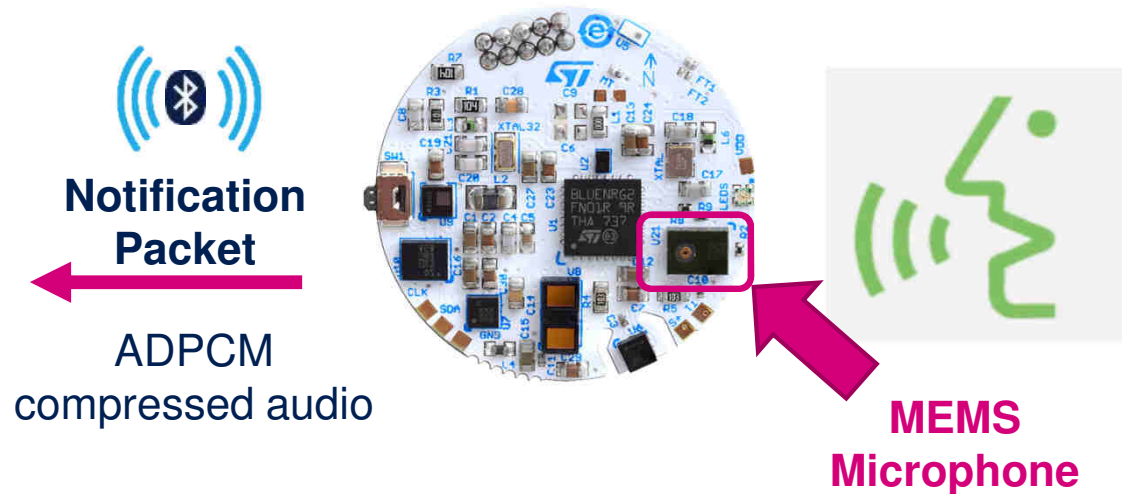
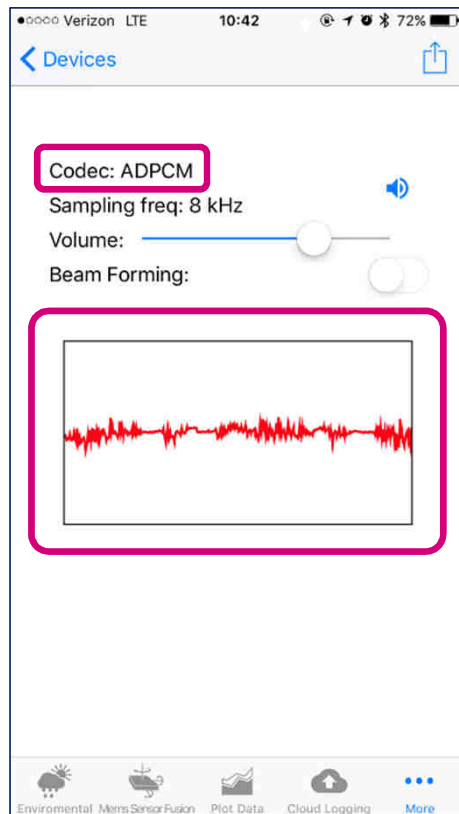


# Lab 7

## Voice over BLE

- In this example we are going to demonstrate how to:
  - **Enable the BlueVoice library** for Voice Over BLE streaming
    - Input: raw data from the MEMS microphone
    - Output: ADPCM data streaming @8kHz
  - **Send ADPCM data** through **BLE notifications packets** to the ST BLE Sensor app

1. Enable **BlueVoice** library
2. Send **voice** to the ST BLE Sensor app client through **BLE notification packet**



4 Bytes

Len	Type	Protocol version	Device ID	Feature Mask	Device MAC	Len	Type	Local Name	Len	Type	Pwr
-----	------	------------------	-----------	--------------	------------	-----	------	------------	-----	------	-----

31	30	29	28	27	26	25	24
RFU	ADPCM	Switch	DoA	ADPC	MicLevel	Proximity	Lux
23	22	21	20	19	18	17	16
Acc	Gyro	Mag	Pressure	Humidity	Temperat	Battery	2nd Temp
15	14	13	12	11	10	9	8
RFU	RFU	RFU	RFU	Beamform	AccEvent	FreeFall	SensFusC
7	6	5	4	3	2	1	0
SensFus	Compass	MotionInt	Activity	Carry Pos	MemsGes	ProxGes	Pedo

0x6A

0xFE

0x05

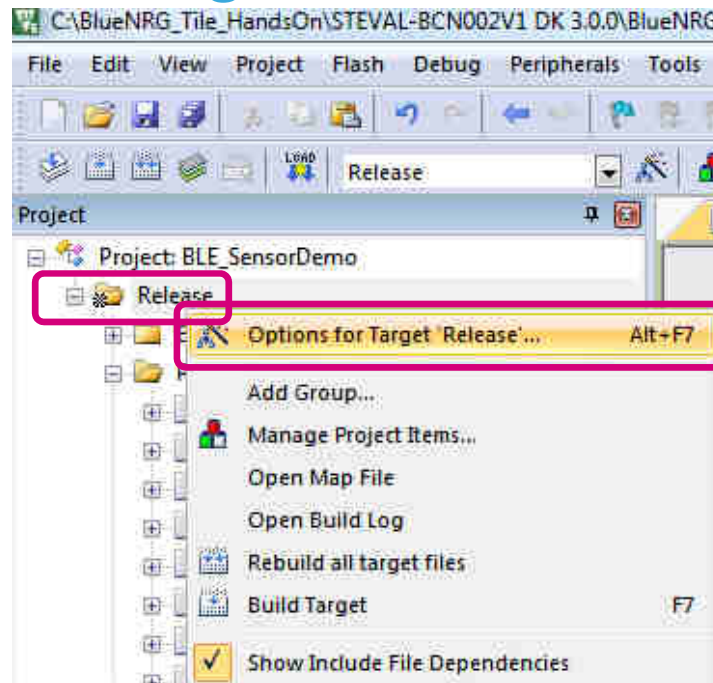
0x40

# Enable BlueVoice Library

150

Add the preprocessor symbol **ENABLE\_AUDIO** to the project

1. Right-click on **Release**
2. Select “Options for Target ‘Release’...”

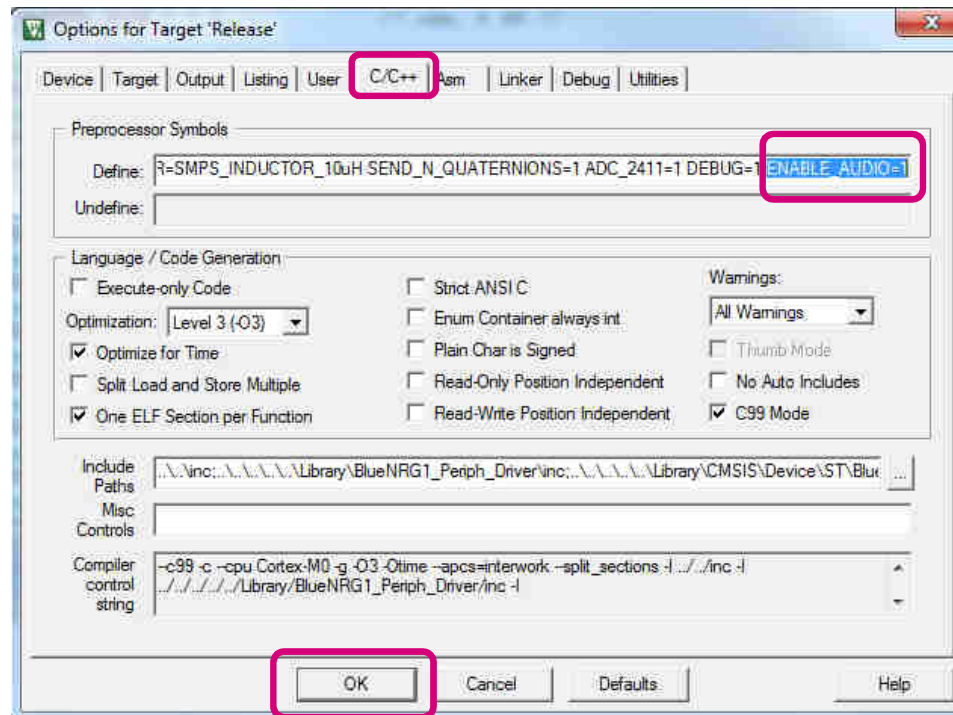


# Enable BlueVoice Library

151

In the project options:

1. Go to the tab **C/C++** and then in the **Preprocessor Symbols**
2. Set the symbol **ENABLE\_AUDIO=1** and then click on **OK**

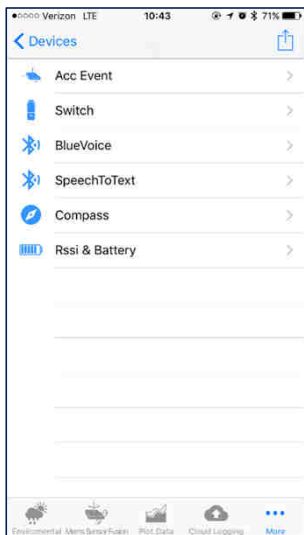




# BlueVoice: Voice over Bluetooth LE

152

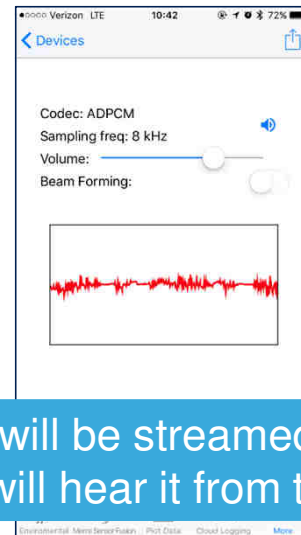
Select  
“BlueVoice”



Tap

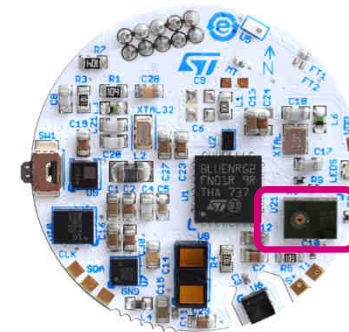


Speak close to  
the BlueNRG-Tile



Voice will be streamed over BLE  
You will hear it from the phone

Do not silence your phone,  
must NOT be vibration only!

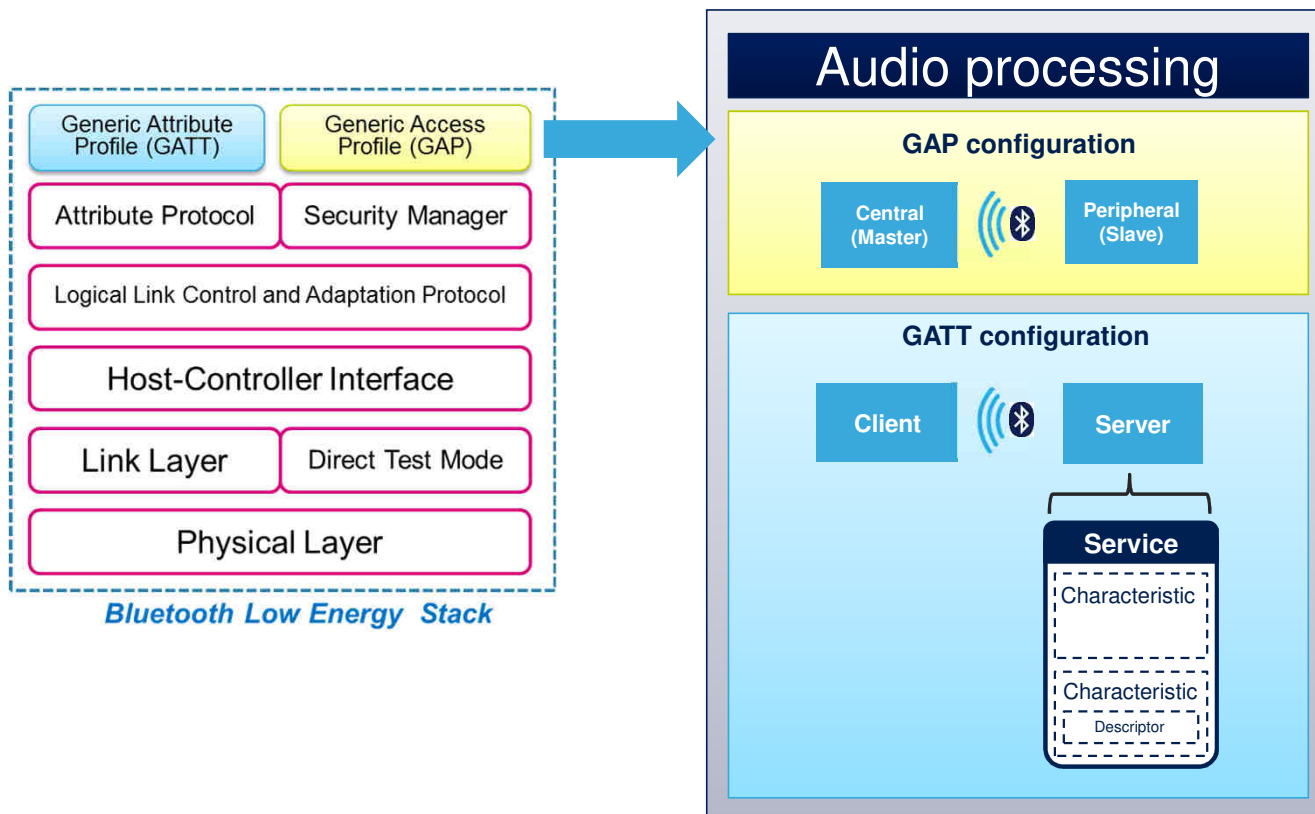


**MEMS  
Microphone**

(if the mic captures the audio from  
the phone speaker, a very high  
pitch sound can happen!)

# Voice over Bluetooth LE

153



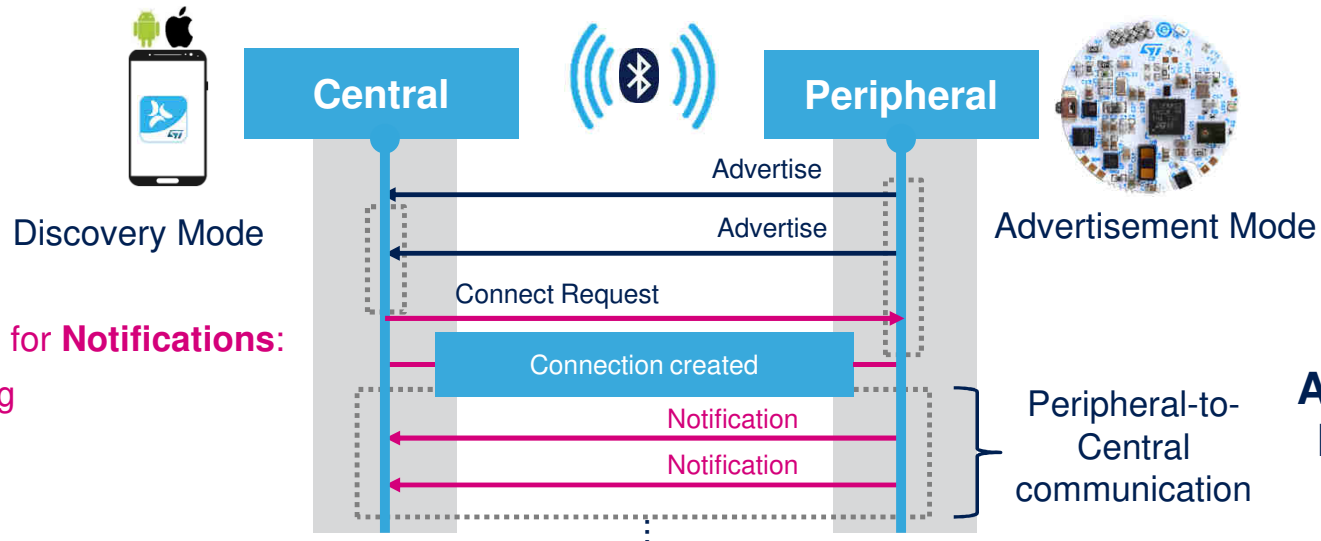
It's a Vendor Specific Service!

Audio: exported by the Server through **2 dedicated BLE characteristics**

Voice streaming over BLE:  
Audio @8kHz  
Codec: ADPCM  
Bitrate: 32kbps

# BlueVoice ADPCM - Audio 8kHz

154



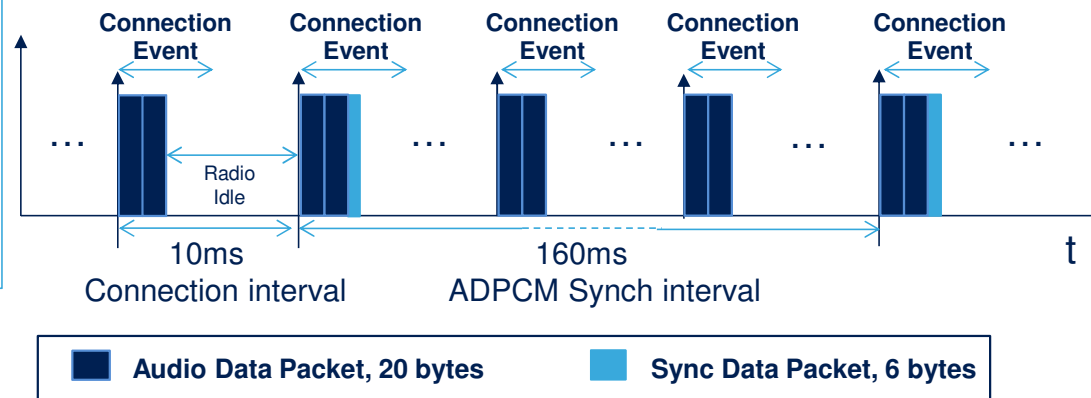
No acknowledge required for **Notifications**:

- OK for Audio streaming
- **Minimal Latency**
- **Optimal Bandwidth**

**Audio can be half-duplex**

## Voice Streaming

- Audio Format: **PCM 16 bit @ 8 kHz**
- **ADPCM Compression @ 32 Kbps**: Low latency and low complexity
- Side Information at low data rate enhances error resilience



**2 packets/event**  
**20 bytes/packets**  
**100 events/sec**

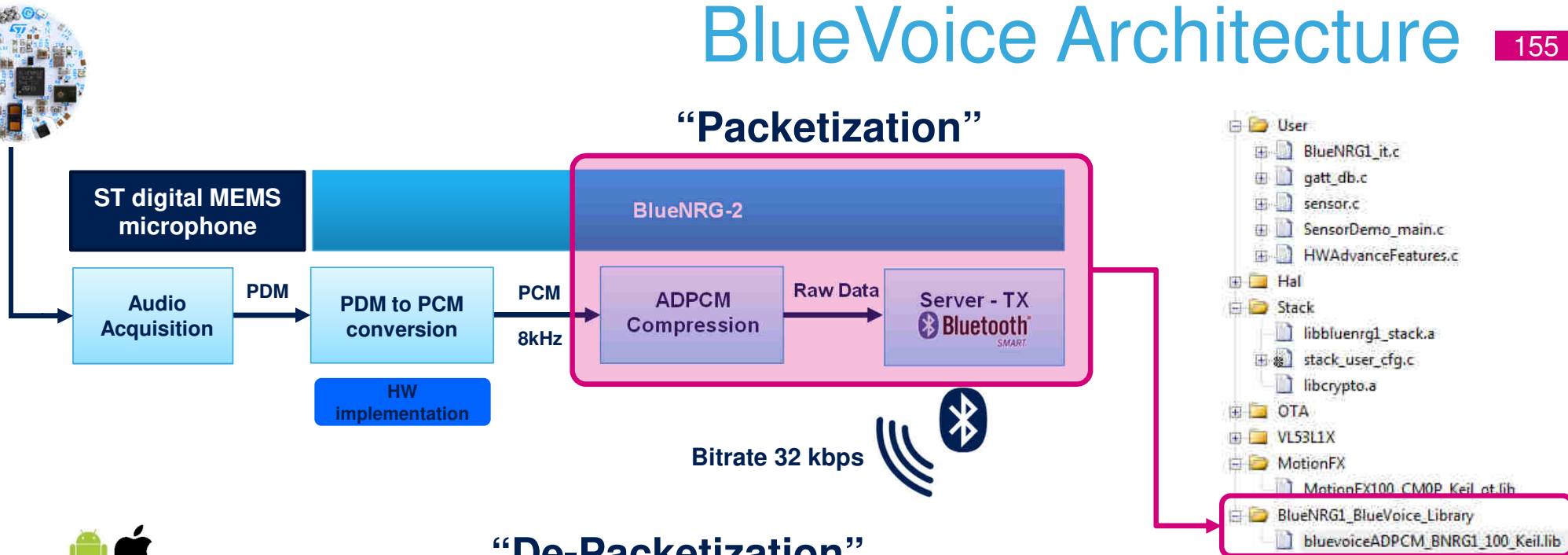
**=**

**32kbps**

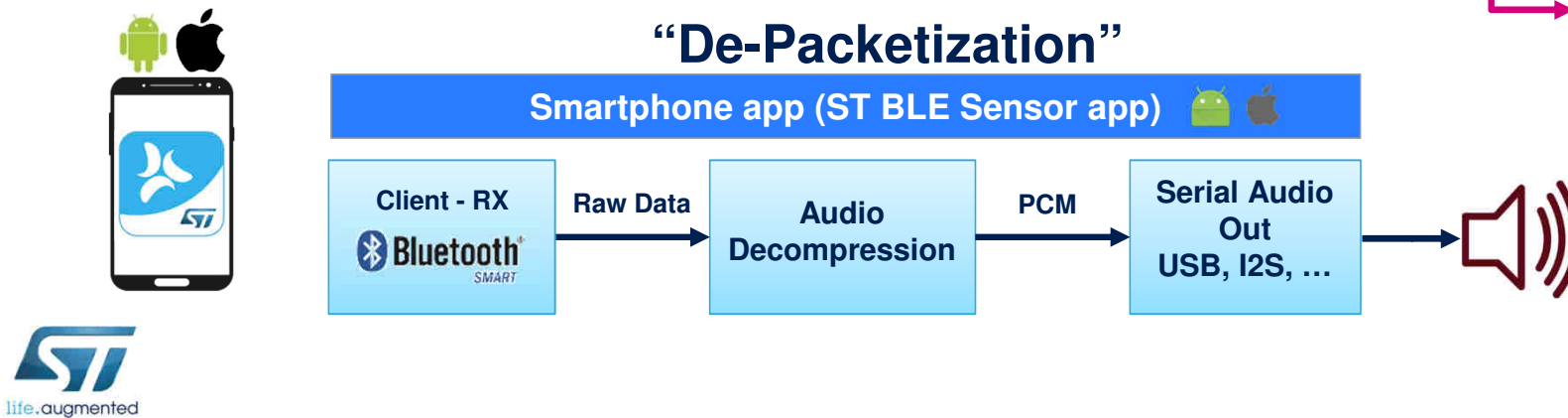
# BlueVoice Architecture

155

## “Packetization”



## “De-Packetization”



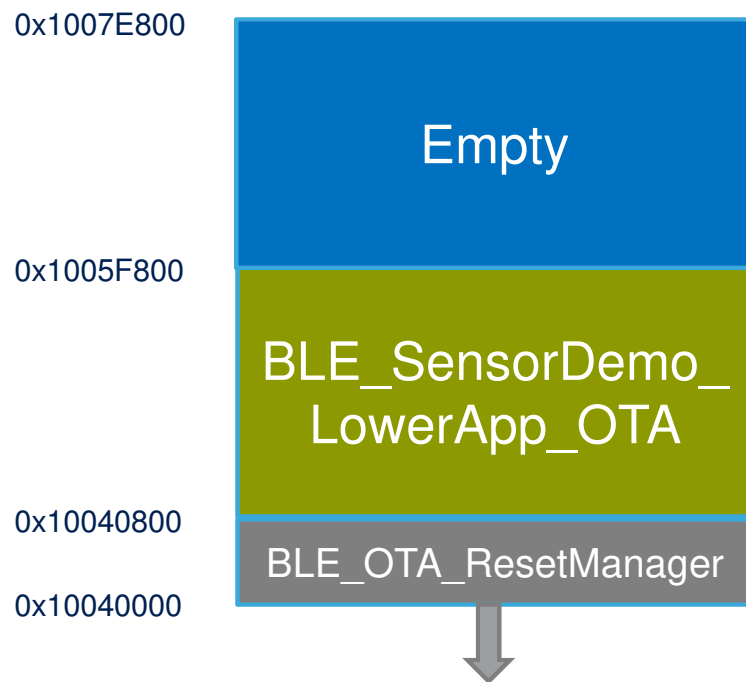


# Lab 8

## Over-The-Air (OTA) FW Upgrade

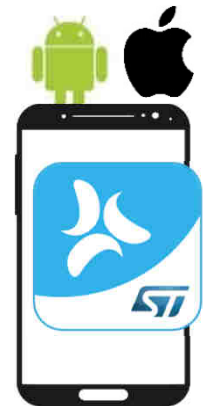
# Flash Memory Layout

157



**OTA**

**BLE\_Chat\_Server\_HigherApp\_OTA.bin**



**REMEMBER TO LOAD THE BLE OTA Reset manager!!!**

# Once OTA is Completed...

158



0x1007E800

BLE\_Chat\_Server\_  
HigherApp\_OTA

0x1005F800

Empty

0x10040800

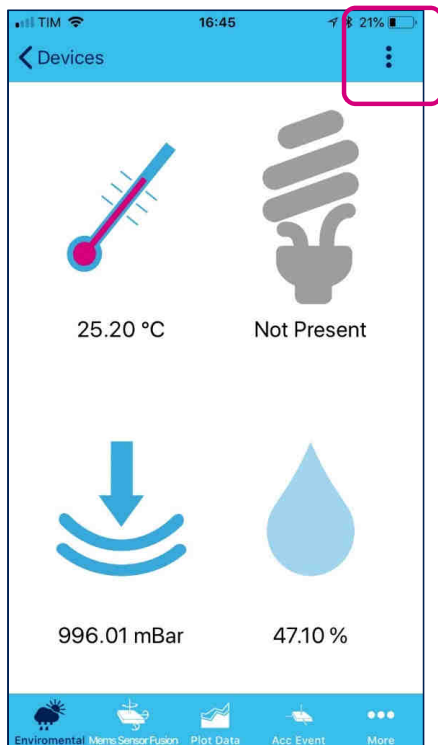
BLE\_OTA\_ResetManager

0x10040000

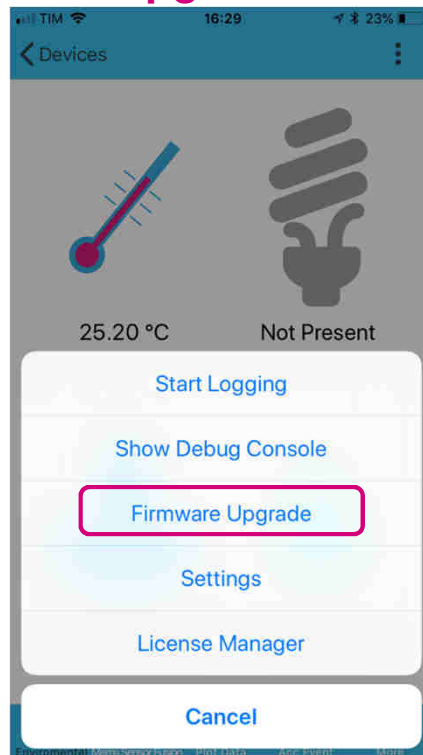
# OTA FW Upgrade (1/2)

159

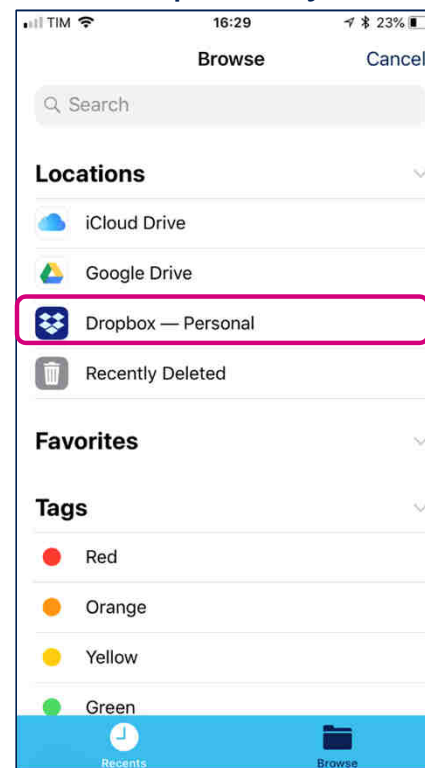
Click on the top right corner



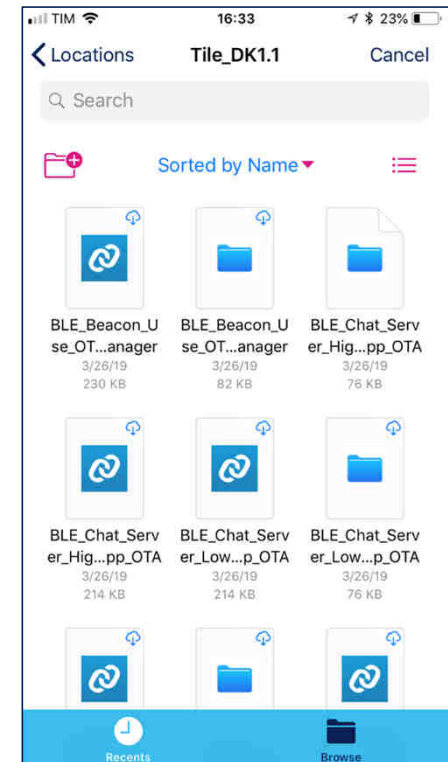
Select "Firmware Upgrade"



Select your remote repository



Select the binary file

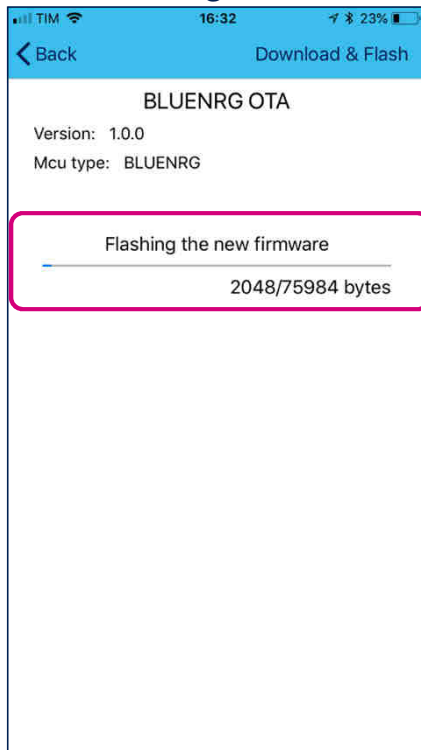




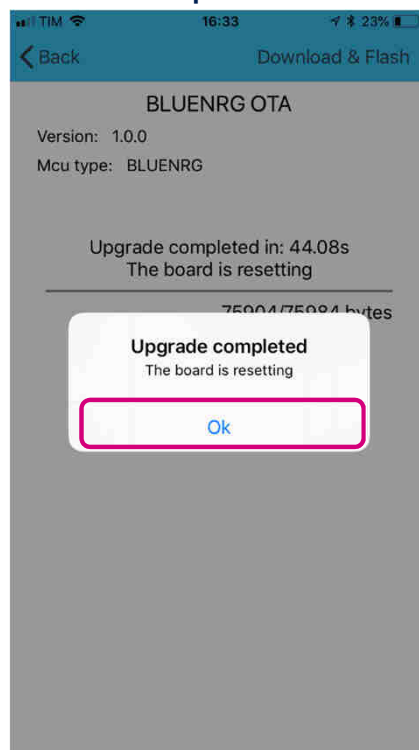
# OTA FW Upgrade (2/2)

160

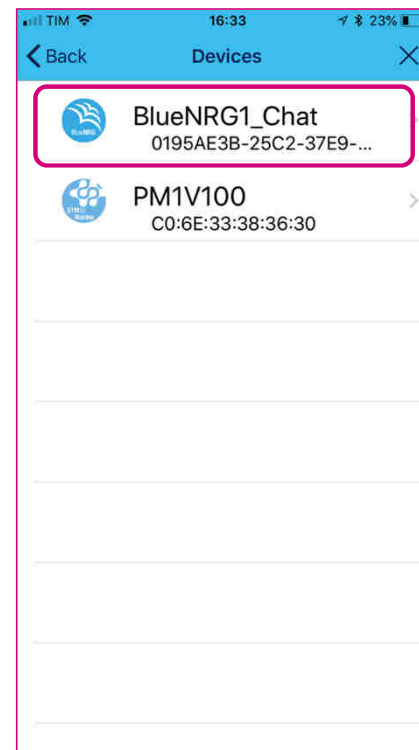
**OTA data transfer begins**



**Click OK as completed**



**Re-Connect** to BlueTile, now running a new application

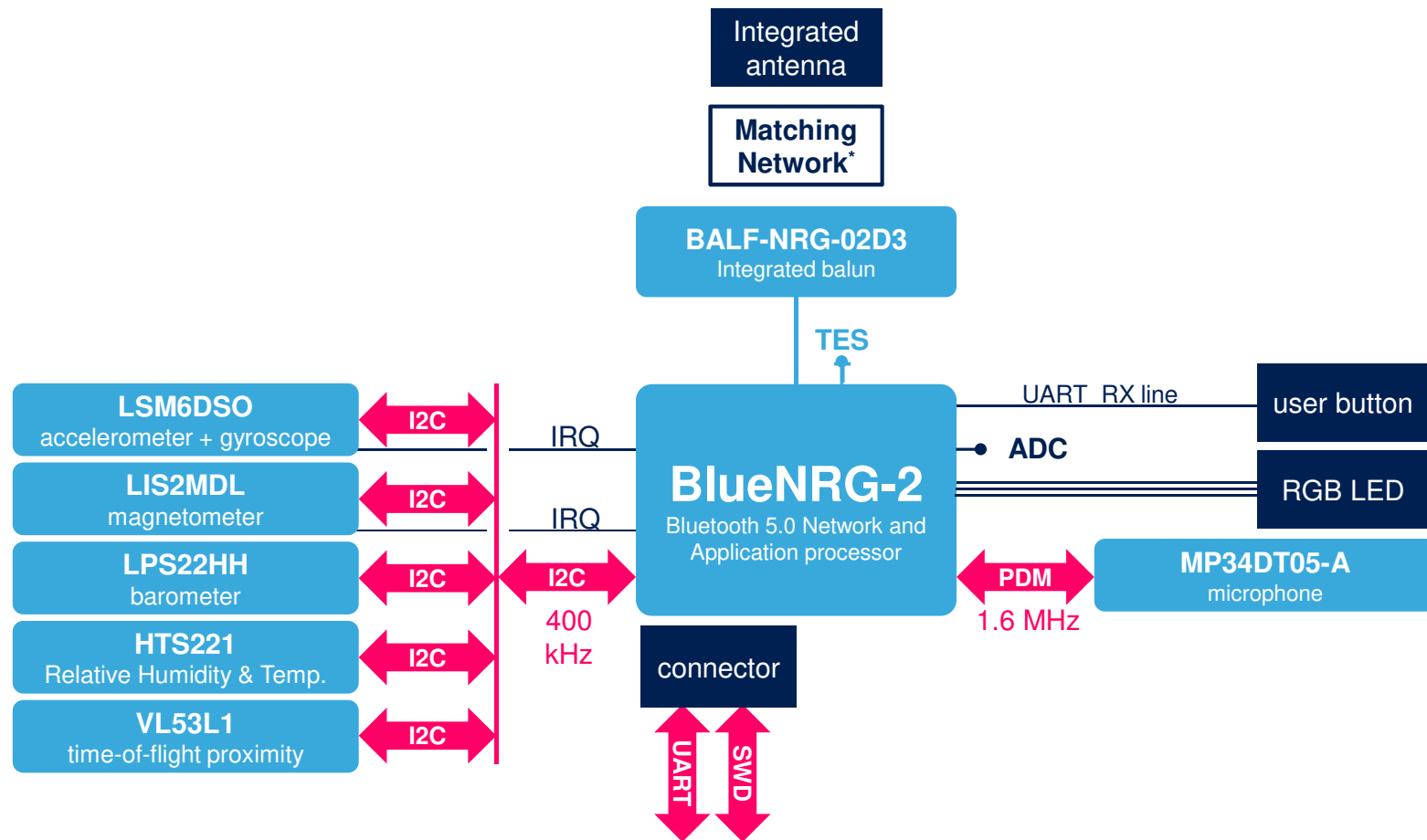




# Customizing Your Design

# STEVAL-BCN002V1 B Block Diagram

162



# Enable/Disable Sensors & Libraries

- **Dedicated structure “FeaturePresence”** for enabling/disabling sensors & libraries individually. File **sensor.h**

```
typedef struct {
    bool AccelerometerGyroscopePresence;
    bool MagnetometerPresence;
    bool HumidityTemperaturePresence;
    bool PressurePresence;
    bool ProximityLightPresence;
    bool iNemoEngine;
    bool Pedometer;
} FeaturePresence;
```

- **SensorScan() function** set each field of the structure to **false** or **true**. File **sensor.c**

Disable the unneeded sensor for optimizing the power consumption!

```
// Check sensor list
SensorsScan();

// Configure discovered sensors
if (xFeaturePresence.PressurePresence)
    Init_Pressure_Temperature_Sensor();
if (xFeaturePresence.HumidityTemperaturePresence)
    Init_Humidity_Sensor();
if (xFeaturePresence.MagnetometerPresence)
    Init_Magnetometer();
if (xFeaturePresence.AccelerometerGyroscopePresence)
    Init_Accelerometer_Gyroscope();
if (xFeaturePresence.ProximityLightPresence)
    Init_Proximity_Sensor();

// Configure sensors in low power mode
SensorsLowPower();
```

- **Advertising intervals:**

- Dedicated API `aci_gap_set_discoverable(Advertising_Type, Advertising_Interval_Min, Advertising_Interval_Max, ... )`
- In file **sensor.c**

- **Connection intervals:**

- Dictated by the Central device. Peripheral has no full control on this.

- **Notifications frequency**

- Dedicated Virtual Timers (mapped on HW physical timers) for different functionalities
- Timeouts defined in **sensor.h**

```
#define BATTERY_UPDATE_RATE      1000    // Fixed ODR @ 1 Hz
#define ENV_SENSOR_UPDATE_RATE   100     // Fixed ODR @ 10 Hz
#define MOTION_SENSOR_UPDATE_RATE 40      // Fixed ODR @ 25 Hz
```

- In this case redesign is of course necessary
- **Schematics and Gerbers** files:
  - online at [www.st.com/bluetile](http://www.st.com/bluetile)
- **BlueNRG-2 pin mapping**
  - Check **BlueNRG-2 DS** at **Table 129**

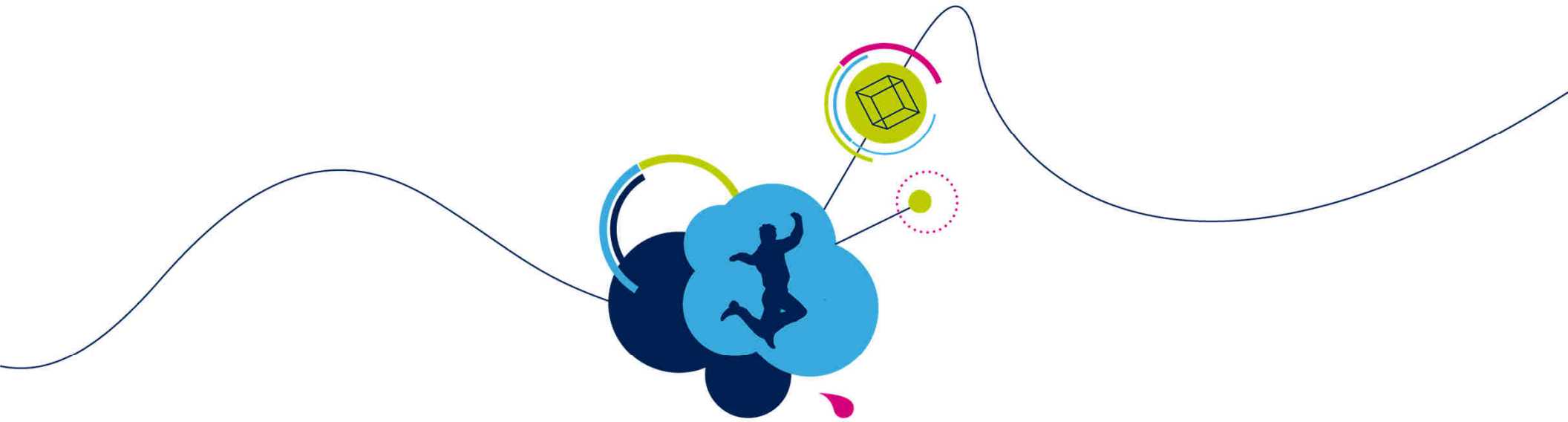
3.12.2

## Functional description

The table below shows the GPIO configuration table where each IO pin is associated with related functions.

**Table 129. IO functional map**

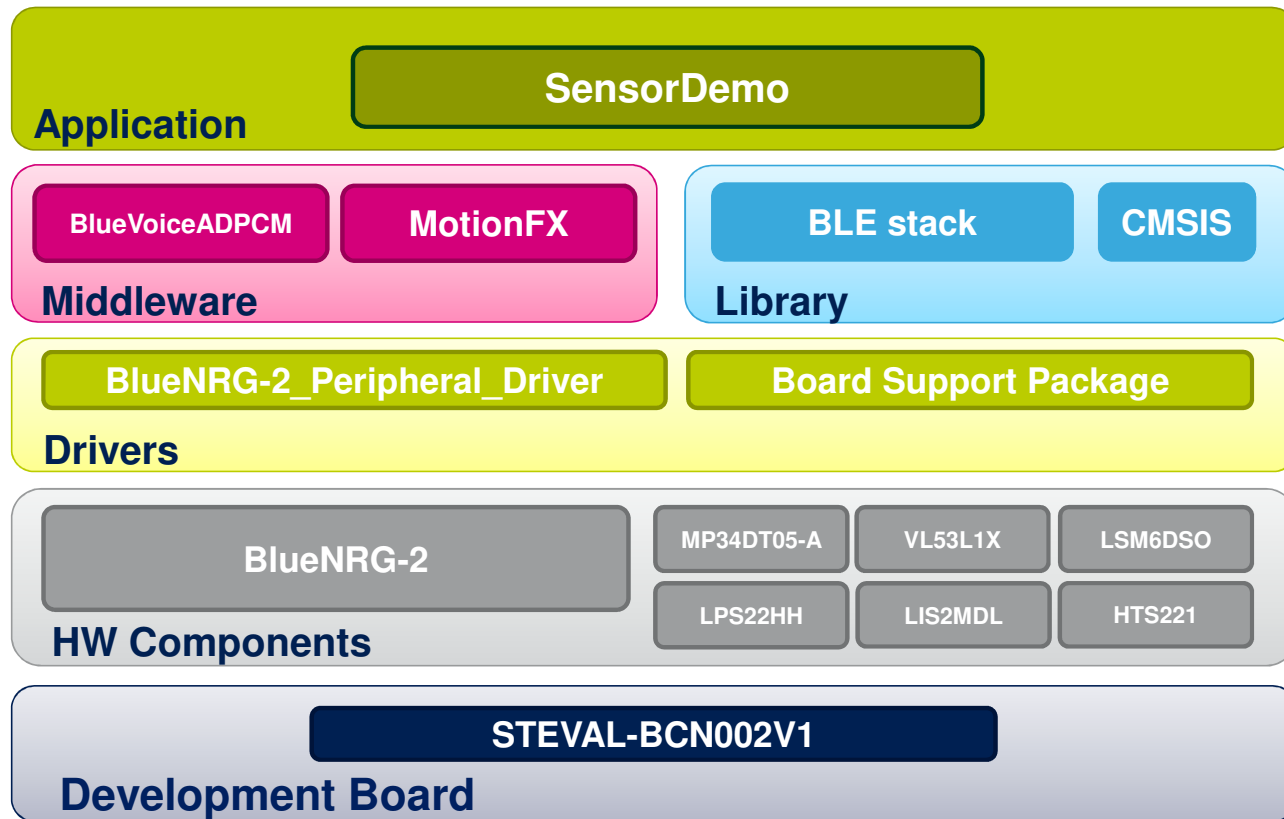
Pin name <sup>(1)</sup>	GPIO mode "000"		Serial1 mode "001"		Serial0 mode "100"		Serial2 mode "101"	
	Type	Signal	Type	Signal	Type	Signal	Type	Signal
IO0	I/O	GPIO 0	I	UART_CTS	I/O	SPI_CLK	O	CPUCCLK
IO1	I/O	GPIO 1	O	UART_RTS	I/O	SPI_CS1	I	PDM_DATA
IO2	I/O	GPIO 2	O	PWM0	O	SPI_OUT	O	PDM_CLK
IO3	I/O	GPIO 3	O	PWM1	I	SPI_IN	-	-
IO4	I/O	GPIO 4	I	UART_RXD	I/O	I2C2_CLK	O	PWM0
IO5	I/O	GPIO 5	O	UART_TXD	I/O	I2C2_DAT	O	PWM1
IO6	I/O	GPIO 6	O	UART_RTS	I/O	I2C2_CLK	I	PDM_DATA
IO7	I/O	GPIO 7	I	UART_CTS	I/O	I2C2_DAT	O	PDM_CLK
IO8	I/O	GPIO 8	O	UART_TXD	I/O	SPI_CLK	I	PDM_DATA
IO9	I/O	GPIO 9	I	SWCLK	I	SPI_IN	O	XO16/32M
IO10	I/O	GPIO 10	I	SWDIO	O	SPI_OUT	O	CLK_32K
IO11	I/O	GPIO 11	I	UART_RXD	I/O	SPI_CS1	O	CLK_32K
IO12	OD	GPI 12 <sup>(2)</sup>		-	I/O	I2C1_CLK	-	-
IO13	OD	GPI 13 <sup>(2)</sup>	I	UART_CTS	I/O	I2C1_DAT	-	-
IO14	I/O	GPIO 14	I/O	I2C1_CLK	I/O	SPI_CLK	-	-



# Quick Recap

# DK 3.0.0 SW Architecture

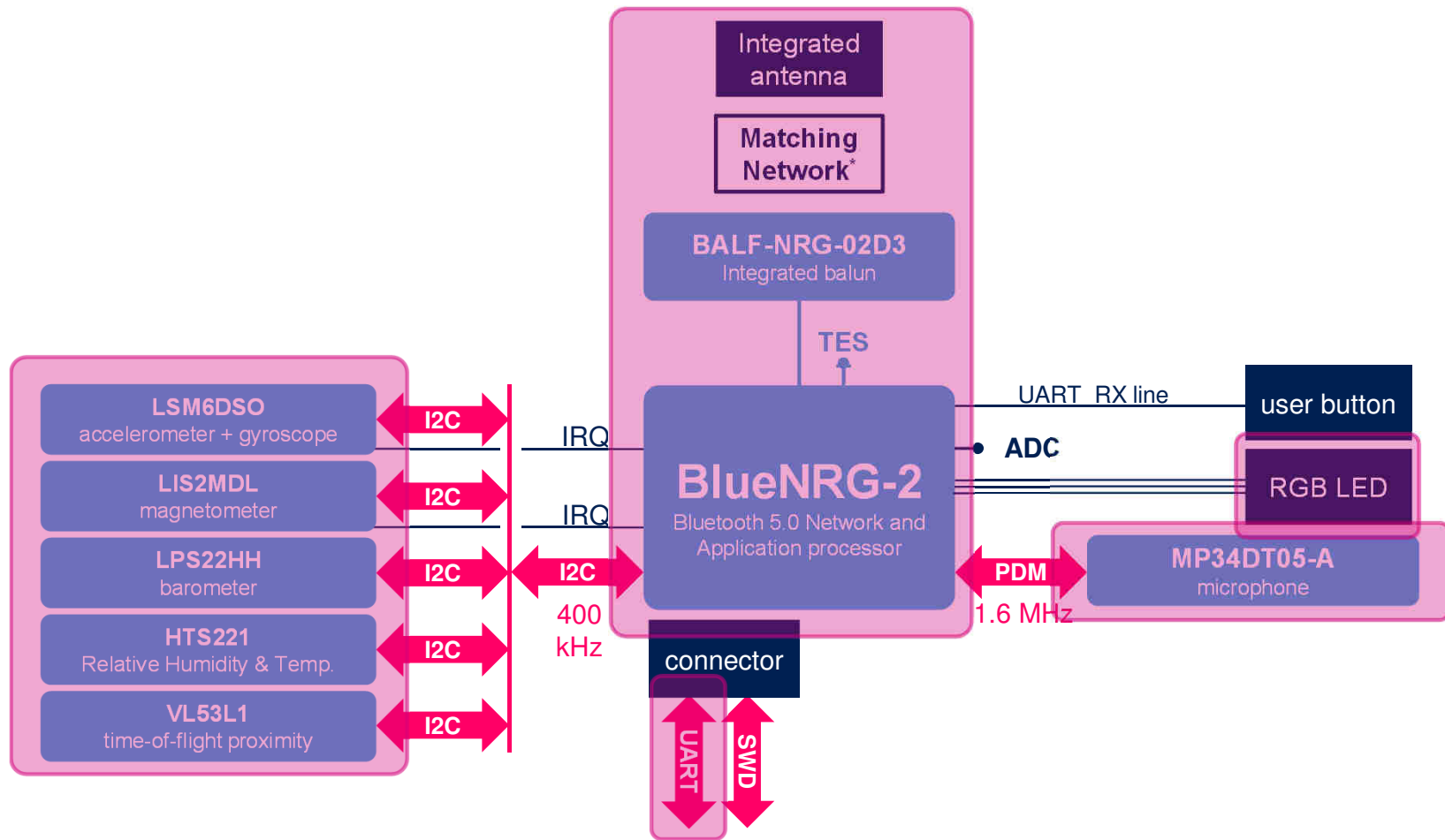
167





# STEVAL-BCN002V1 B Block Diagram

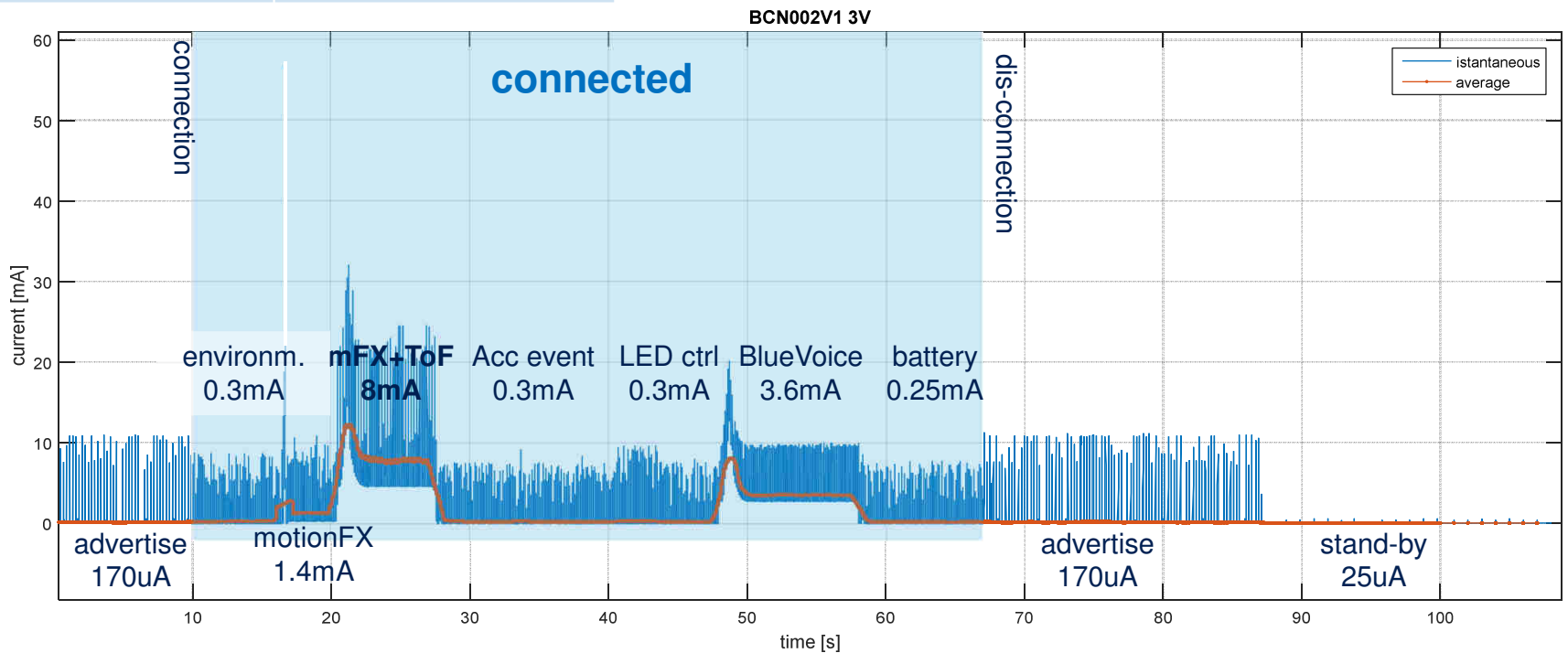
168



# Power Consumption

169

State	Avg power cons
stand-by	25uA
advertise 250ms	170uA
Battery notification	0.25mA
Environmental / AccEvents / LED control	0.3mA
Motion FX (Inertial)	1.4mA
BlueVoice	3.6mA
Motion FX plus Time Of Flight	8mA



# Battery Lifetime: CR2032

170

State	Avg Power Cons	Battery lifetime
stand-by	25uA	8800 hrs / 367 days
advertise 250ms	170uA	1294 hrs / 54 days
Battery notification	0.25mA	880 hrs / 37 days
Environmental / AccEvents / LED control	0.3mA	733 hrs / 30 days
Motion FX (Inertial)	1.4mA	157 hrs / 6.5 days
BlueVoice	3.6mA	61.1 hrs / 2.5 days
Motion FX plus Time Of Flight	8mA	27.5 hrs



The End: Q&A

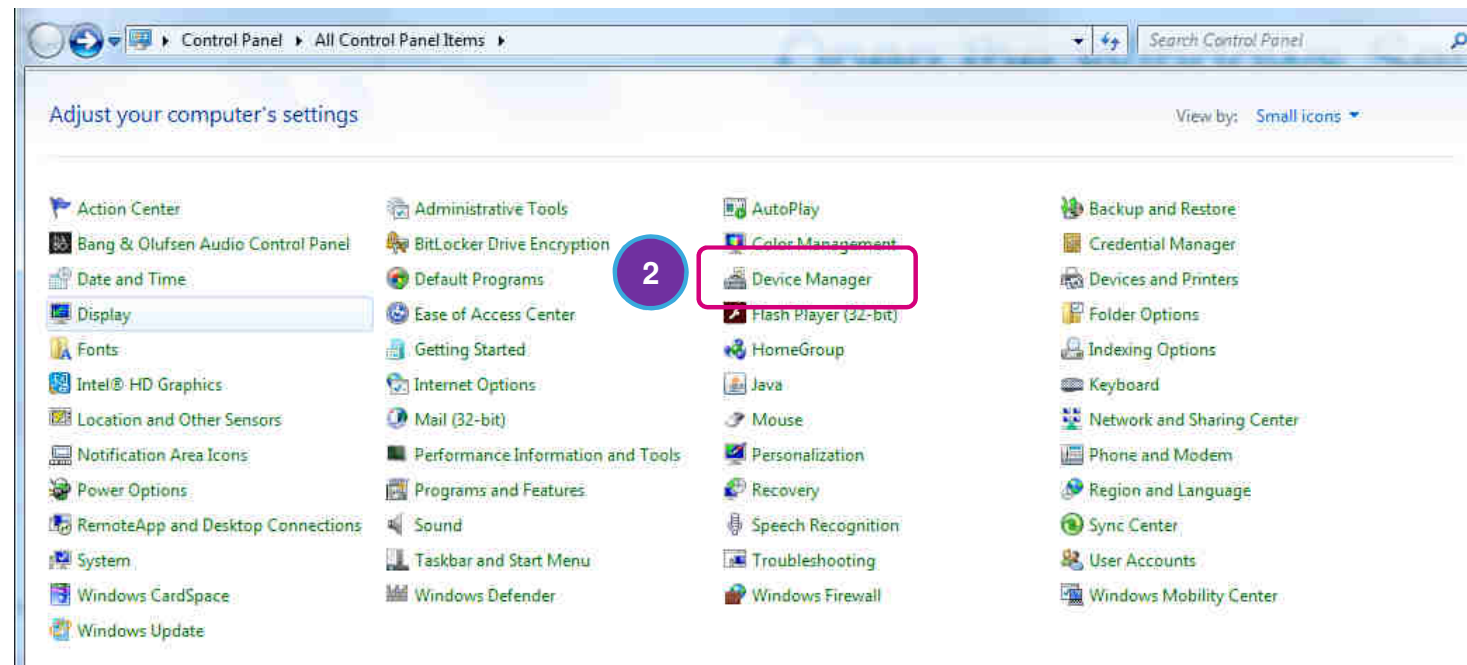


# Virtual COM Driver Installation Win7

# Open the Windows Control Panel

173

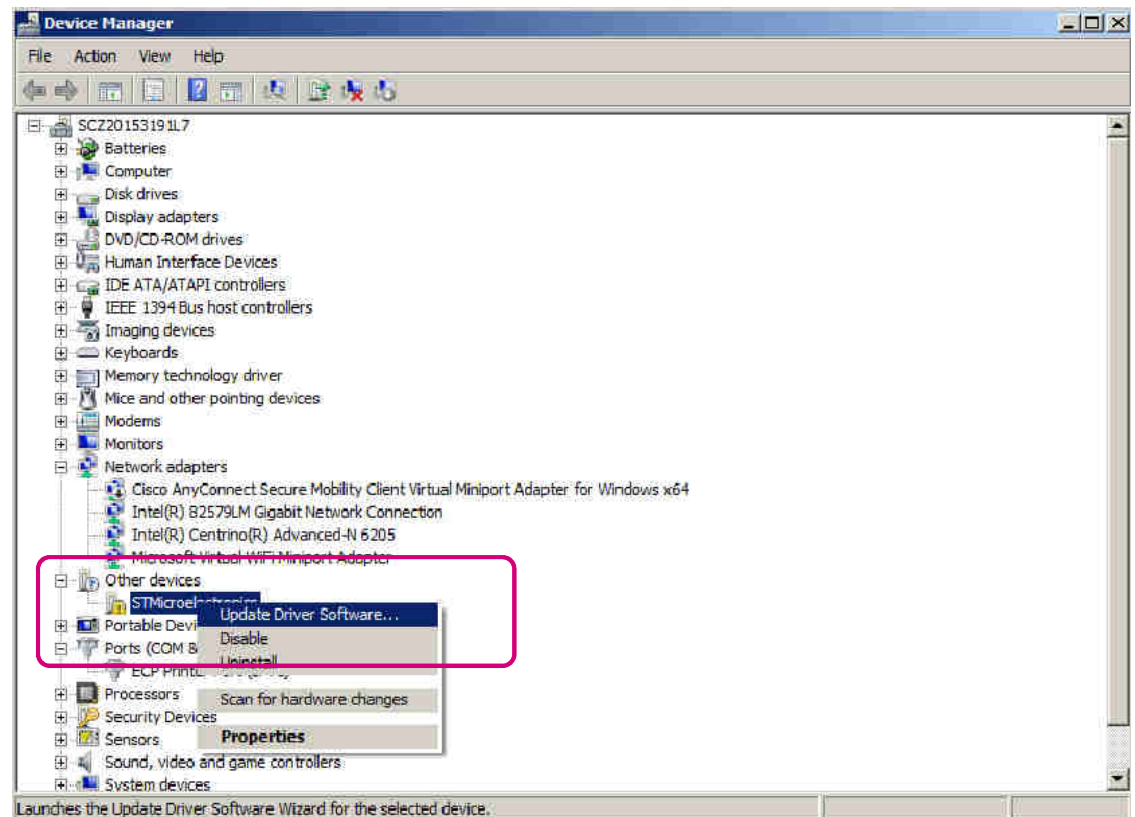
1. From **Start Menu** select **Control Panel**
2. Select **Device Manager**



# Open the Device Manager

174

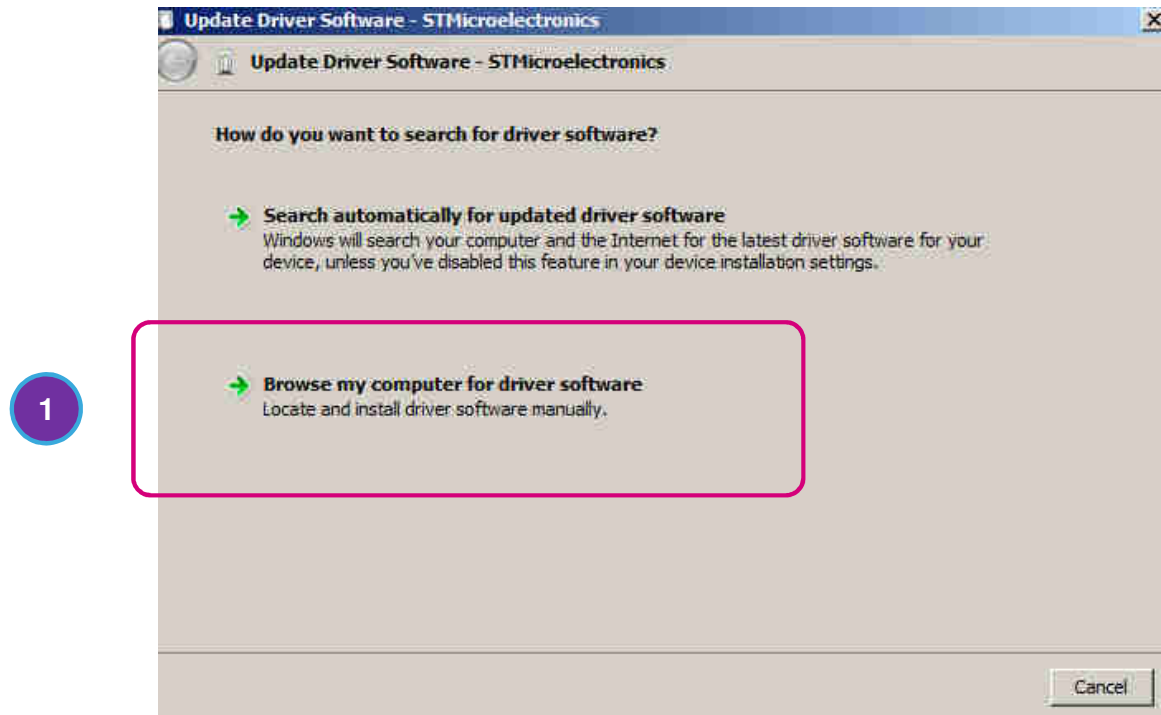
1. Look for **Other devices** and right click and then select **Update Driver Software...**



# Look for the VCOM Driver

175

## 1. Select **Browse my computer for driver software**

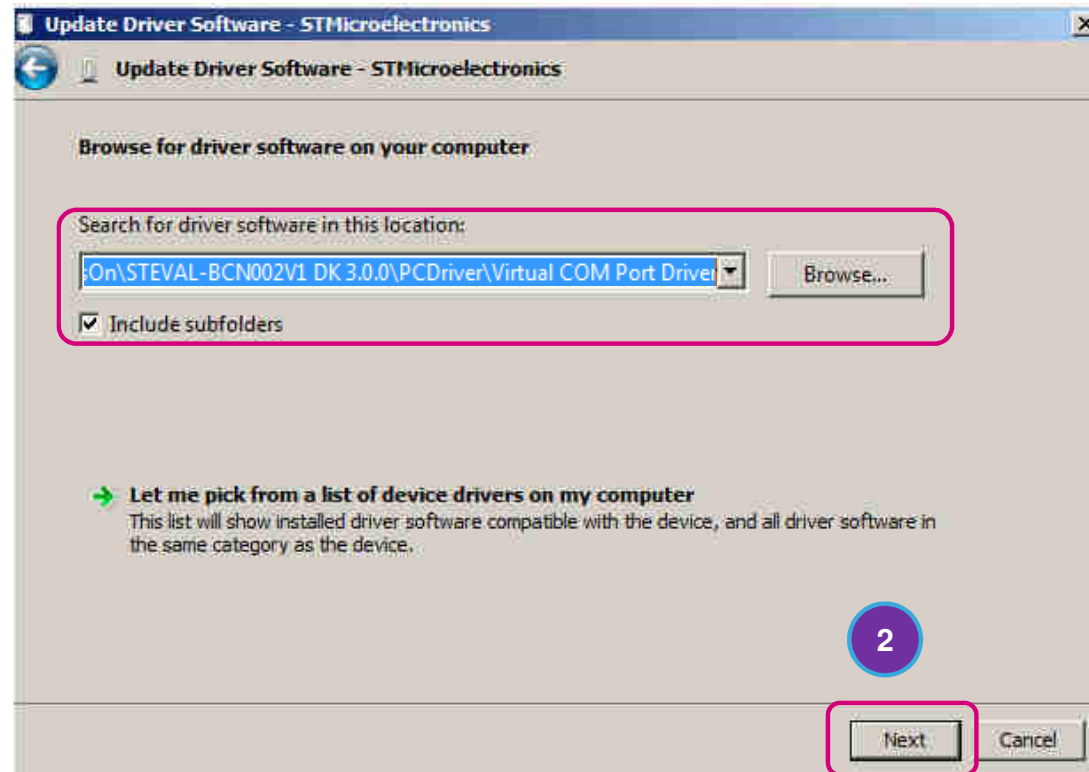




# Look for the VCOM Driver

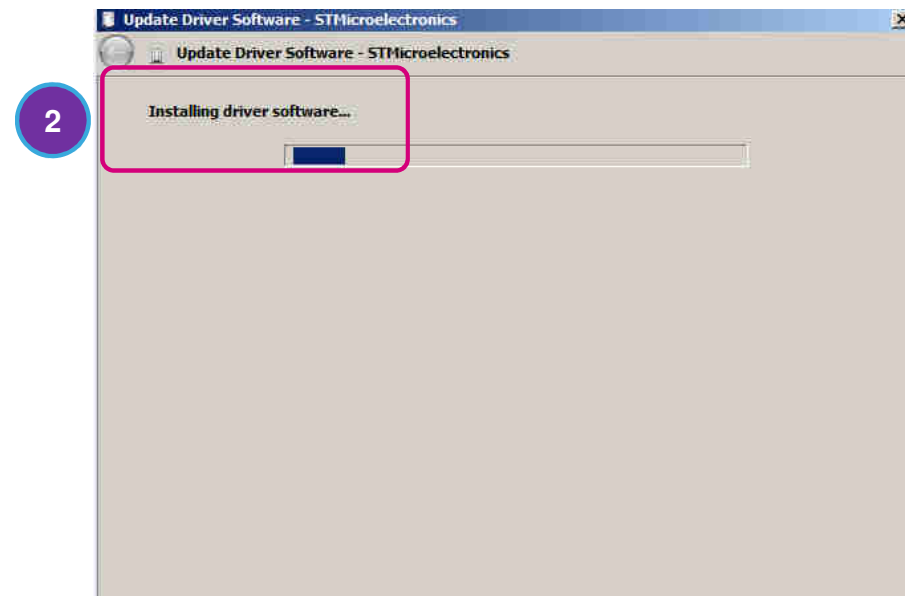
176

1. Click on **Browse** button and go to the folder “\STEVAL-BCN002V1 DK 1.0.9\PCDriver\Virtual COM Port Driver”
2. Click on **Include subfolder** and then on **Next**

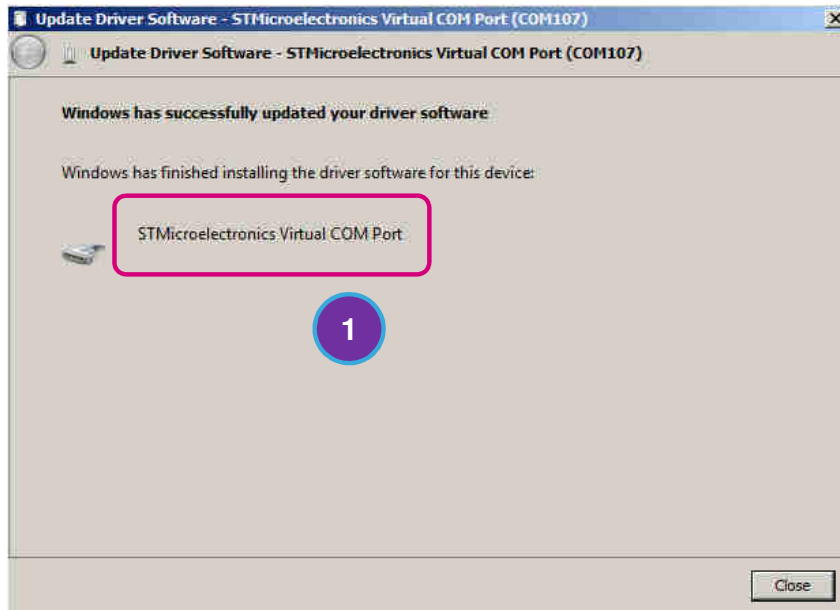


# Allow the Driver Installation 177

1. Click on **Install driver software anyway**
2. Installation starts



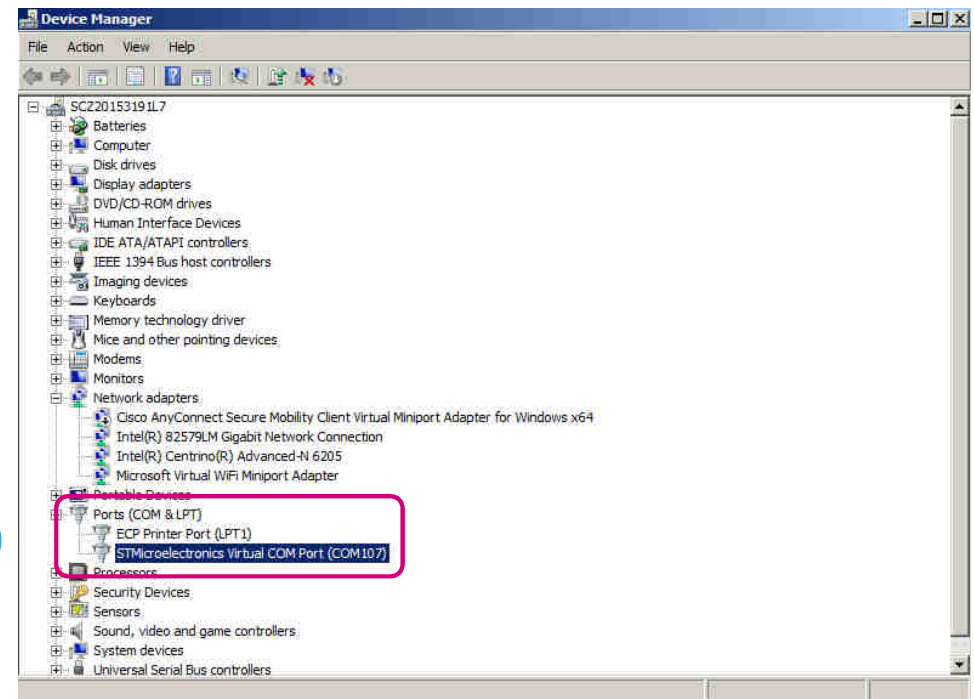
1. Installation completed
2. Device is in the **COM Ports** list



Done! Now please go back from here!



2





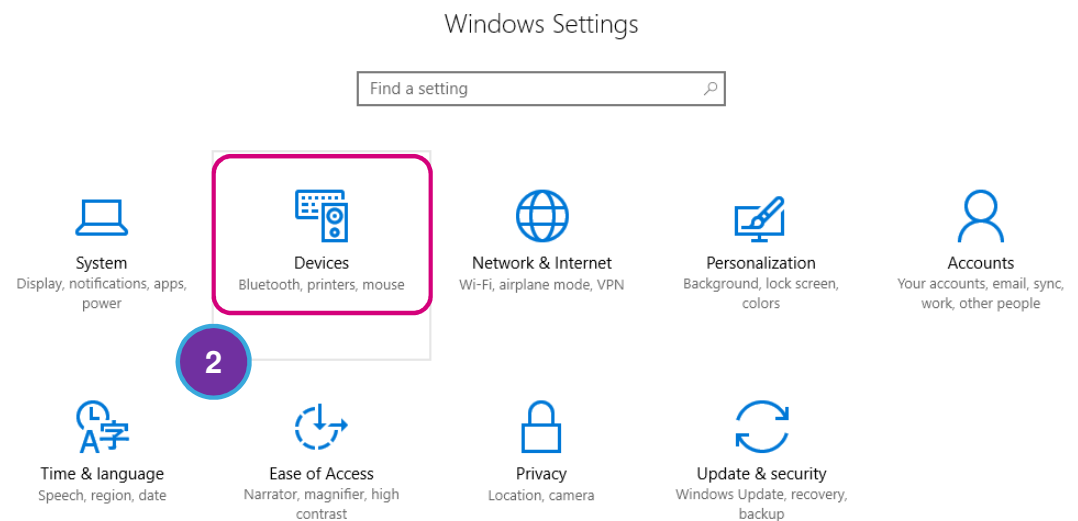
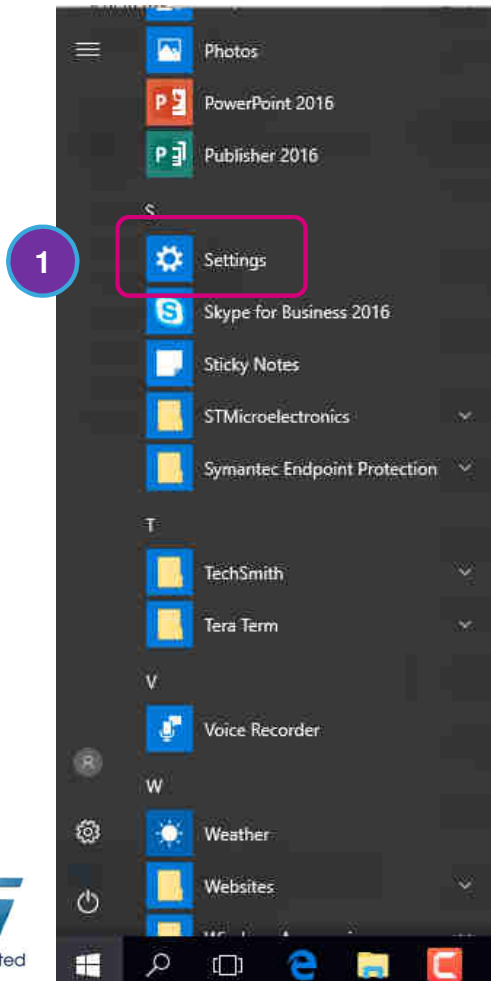
# Virtual COM Driver Installation Win10

# Open the Windows Settings

180

1. From **Start Menu** select **Settings**

2. Select **Devices**

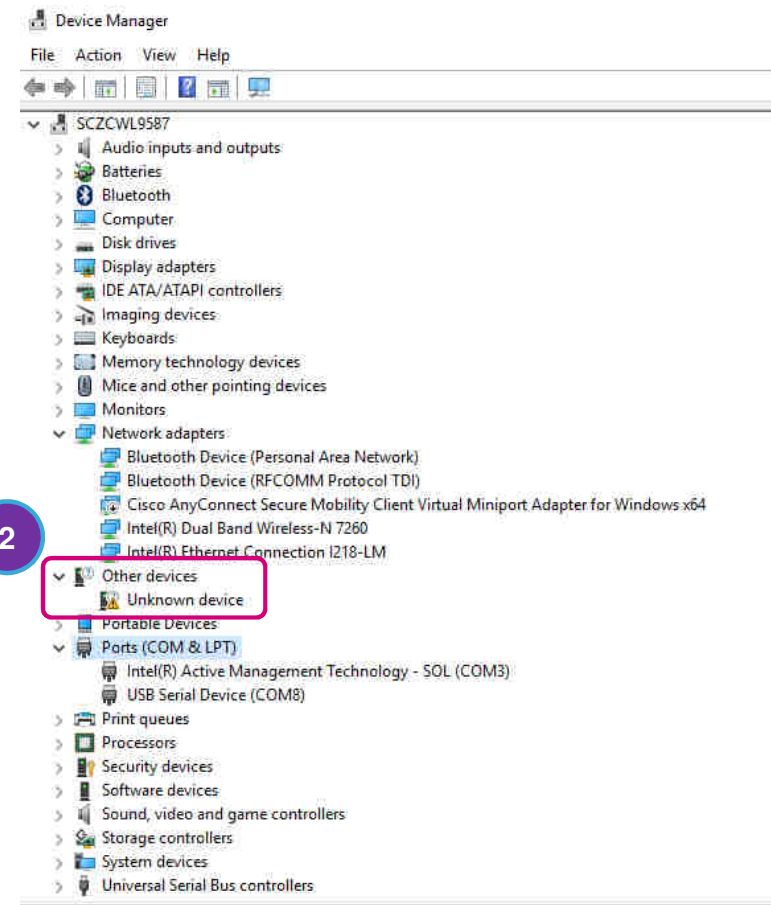
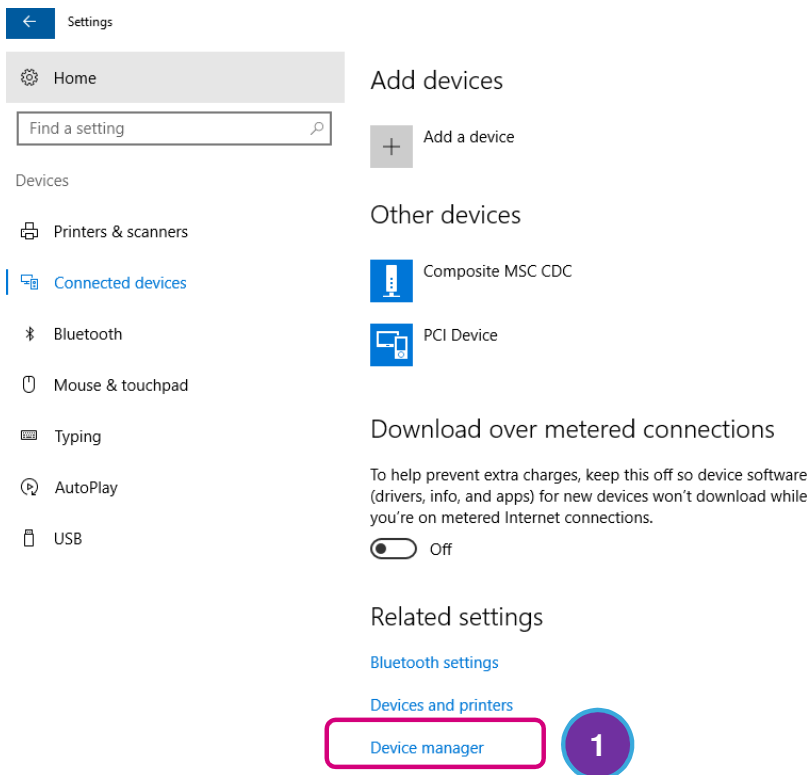


# Open the Device Manager

181

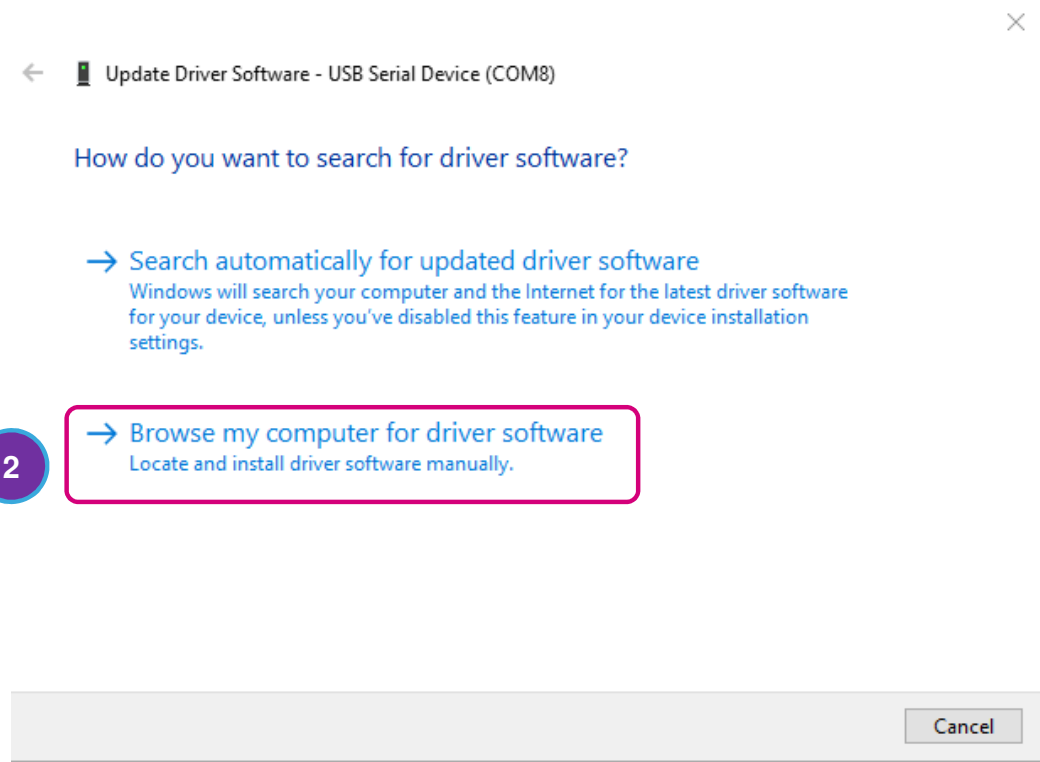
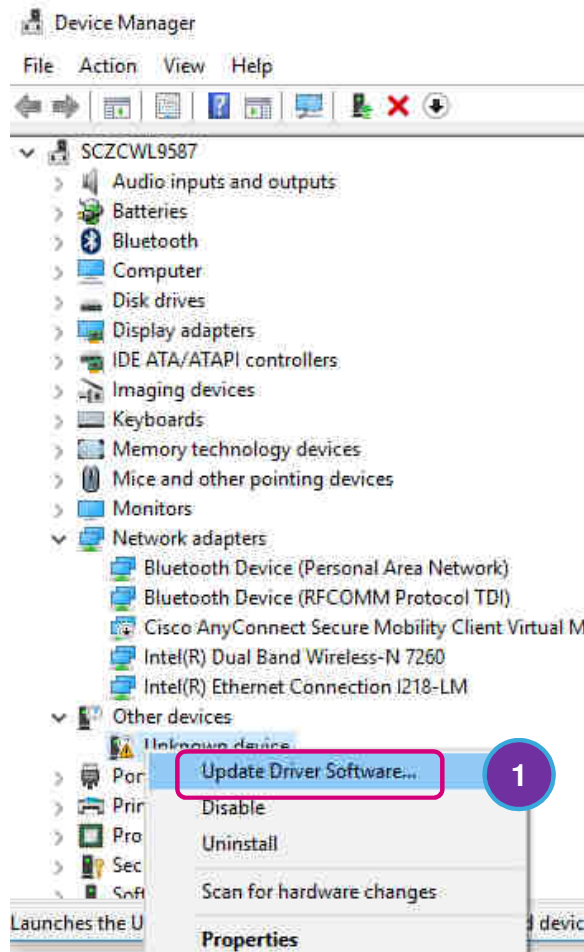
## 1. Select Device manager

## 2. Select Other Devices and Unknown device



# Update Driver Software 182

1. Right click and then select **Update Driver Software...**
2. Select **Browse my computer for driver software**



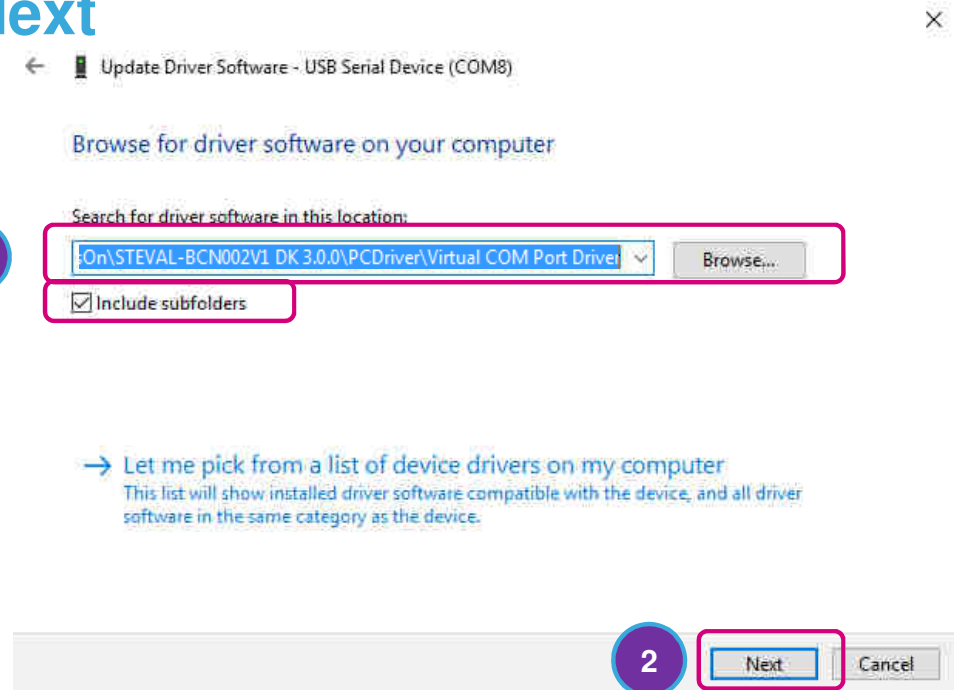
# Look for the VCOM Driver

183

1. Click on **Browse** button and **go to** the folder

“\STEVAL-BCN002V1 DK 1.0.9\PCDriver\Virtual COM Port Driver”

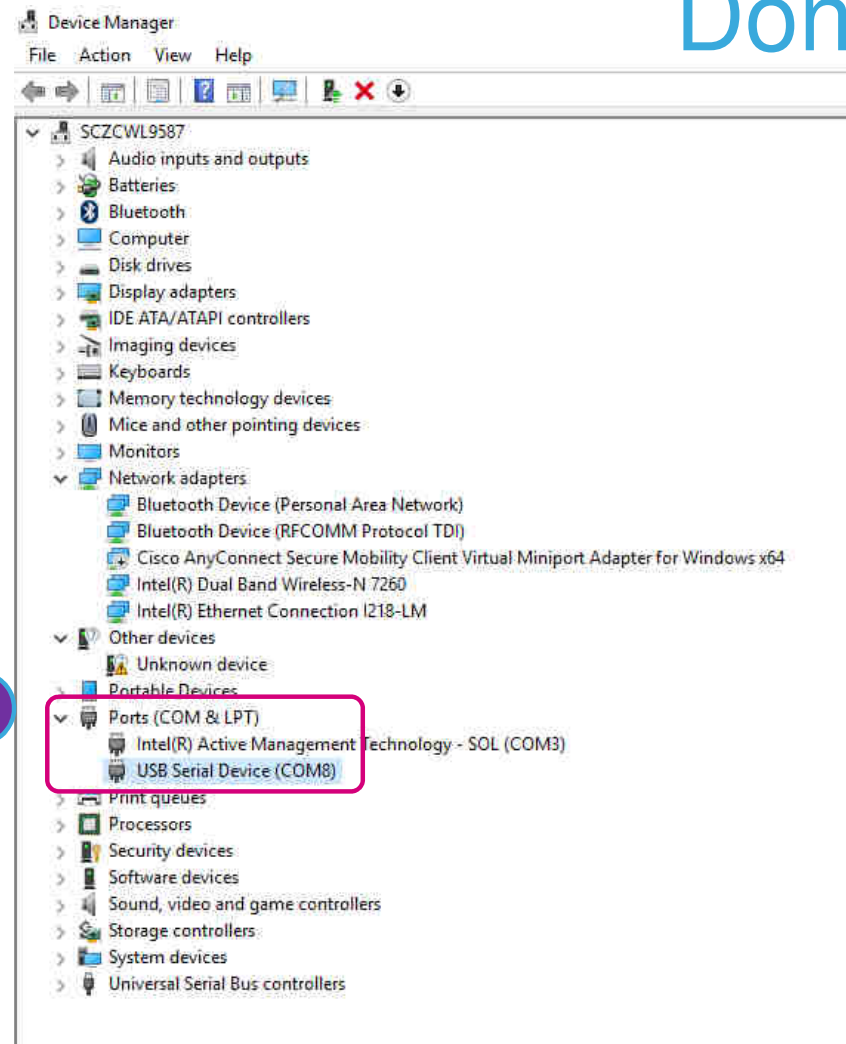
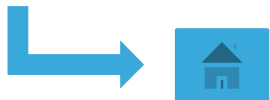
2. Click on **Include subfolder** and then on **Next**





1. Installation completed. Device is in the **COM Ports** list

Done! Now please go back from here!



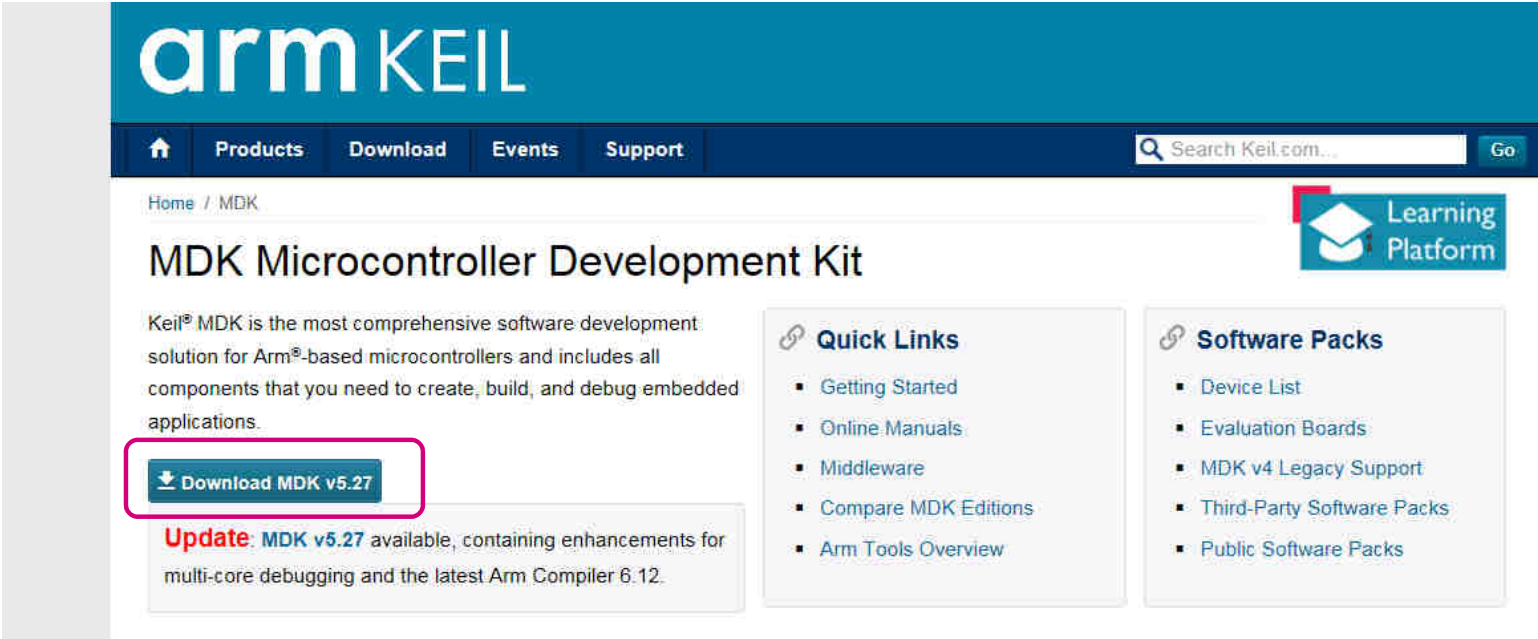


# ARM Keil MDK Installation

# Keil ARM MDK Installation

186

- Download and install the tool from the following webpage:
  - <http://www2.keil.com/mdk5/>



The screenshot shows the Keil ARM MDK website. The header features the 'arm KEIL' logo. Below the header is a navigation bar with links for Home, Products, Download, Events, and Support. A search bar is located on the right side of the navigation bar. The main content area is titled 'MDK Microcontroller Development Kit'. It includes a description of the tool and a prominent 'Download MDK v5.27' button, which is highlighted with a red rectangle. To the right of the main content are two sidebars: 'Quick Links' and 'Software Packs'. The 'Quick Links' sidebar contains links to Getting Started, Online Manuals, Middleware, Compare MDK Editions, and Arm Tools Overview. The 'Software Packs' sidebar contains links to Device List, Evaluation Boards, MDK v4 Legacy Support, Third-Party Software Packs, and Public Software Packs. An update notice at the bottom of the main content area states: 'Update: MDK v5.27 available, containing enhancements for multi-core debugging and the latest Arm Compiler 6.12.'

arm KEIL

Home / MDK

## MDK Microcontroller Development Kit

Keil® MDK is the most comprehensive software development solution for Arm®-based microcontrollers and includes all components that you need to create, build, and debug embedded applications.

[Download MDK v5.27](#)

**Update:** MDK v5.27 available, containing enhancements for multi-core debugging and the latest Arm Compiler 6.12.

### Quick Links

- Getting Started
- Online Manuals
- Middleware
- Compare MDK Editions
- Arm Tools Overview

### Software Packs

- Device List
- Evaluation Boards
- MDK v4 Legacy Support
- Third-Party Software Packs
- Public Software Packs

Learning Platform



# Arm Keil MDK Configuration

# Keil ARM MDK Configuration

188

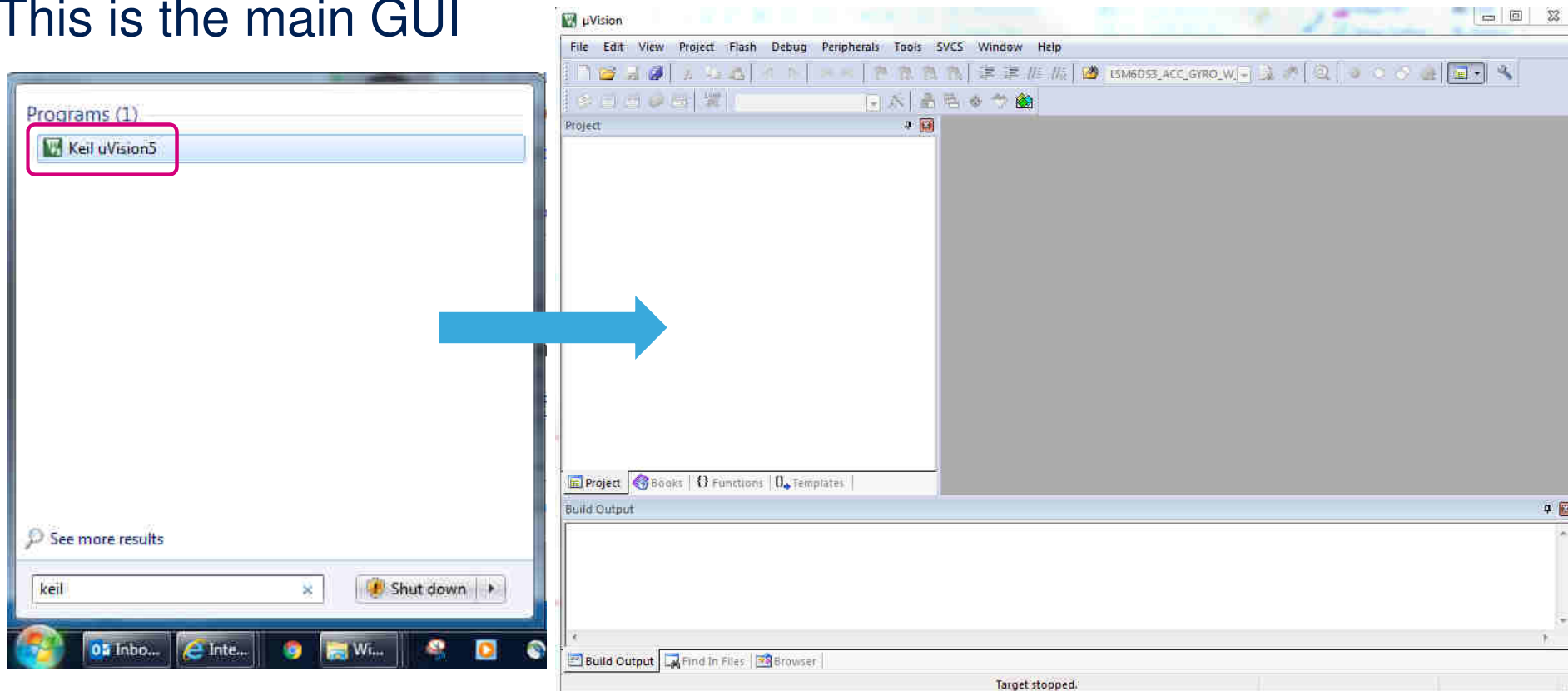
**Download and save on your PC** the .zip file “**Keil.STBlueNRG-2\_DFP.1.0.1.zip**” from the following webpage: [https://www.keil.com/dd2/stmicroelectronics/bluenrg\\_2/](https://www.keil.com/dd2/stmicroelectronics/bluenrg_2/)

The screenshot shows the Keil ARM MDK website. The header features the 'armKEIL' logo and a navigation bar with links: Home, Products, Download, Events, Support, and Videos. A search bar is also present. The left sidebar contains sections for Product Information, Technical Support, Software Downloads, and Other Information. The main content area displays the 'STMicroelectronics BlueNRG-2' product page. It includes a breadcrumb trail: Home / MDK5 Device List / BlueNRG-2. The product name 'STMicroelectronics BlueNRG-2' is prominently displayed. Below it, a box lists 'Hz', 256 kB ROM, 24 kB RAM. A detailed description follows, mentioning 'High performance, ultra-low power ARM Cortex-M0 32-bit based architecture core - Upgradable BLE stack (stored in embedded Flash memory, via SPI) - AES security co-processor - Low power modes - 16 or 32 MHz crystal oscillator - 12 MHz ring oscillator - 32 kHz crystal oscillator - 32 kHz ring oscillator - Compliant with the following radio frequency regulations: ETSI EN 300 328, EN 300 440, FCC CFR47 Part 15, ARIB STD - Operating temperature range: -40 to 85 Celsius'. Below this, there are expandable sections for 'Core' (Hz) and 'Memory' (24 kB RAM, 256 kB ROM). On the right, a 'Quick Links' section lists Board List, Software Packs, MDK Version 5, Legacy Support, and Feedback. Below that, the 'Silicon Supplier' section identifies STMicroelectronics with the ST logo. The 'Device Family Pack' section notes that support is contained in the 'STMicroelectronics BlueNRG-2 Series Device Support' and includes a 'Download' button, which is highlighted with a red rectangular box.

# Keil ARM MDK Configuration

189

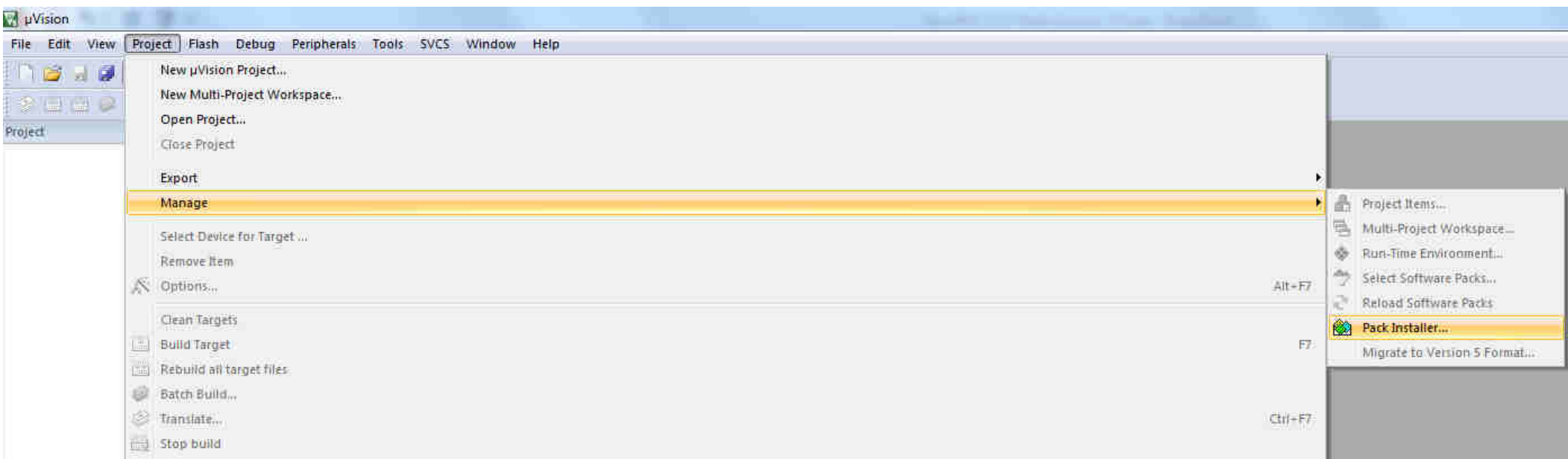
- Open Keil uVision5 IDE
- This is the main GUI



# Keil ARM MDK Configuration

190

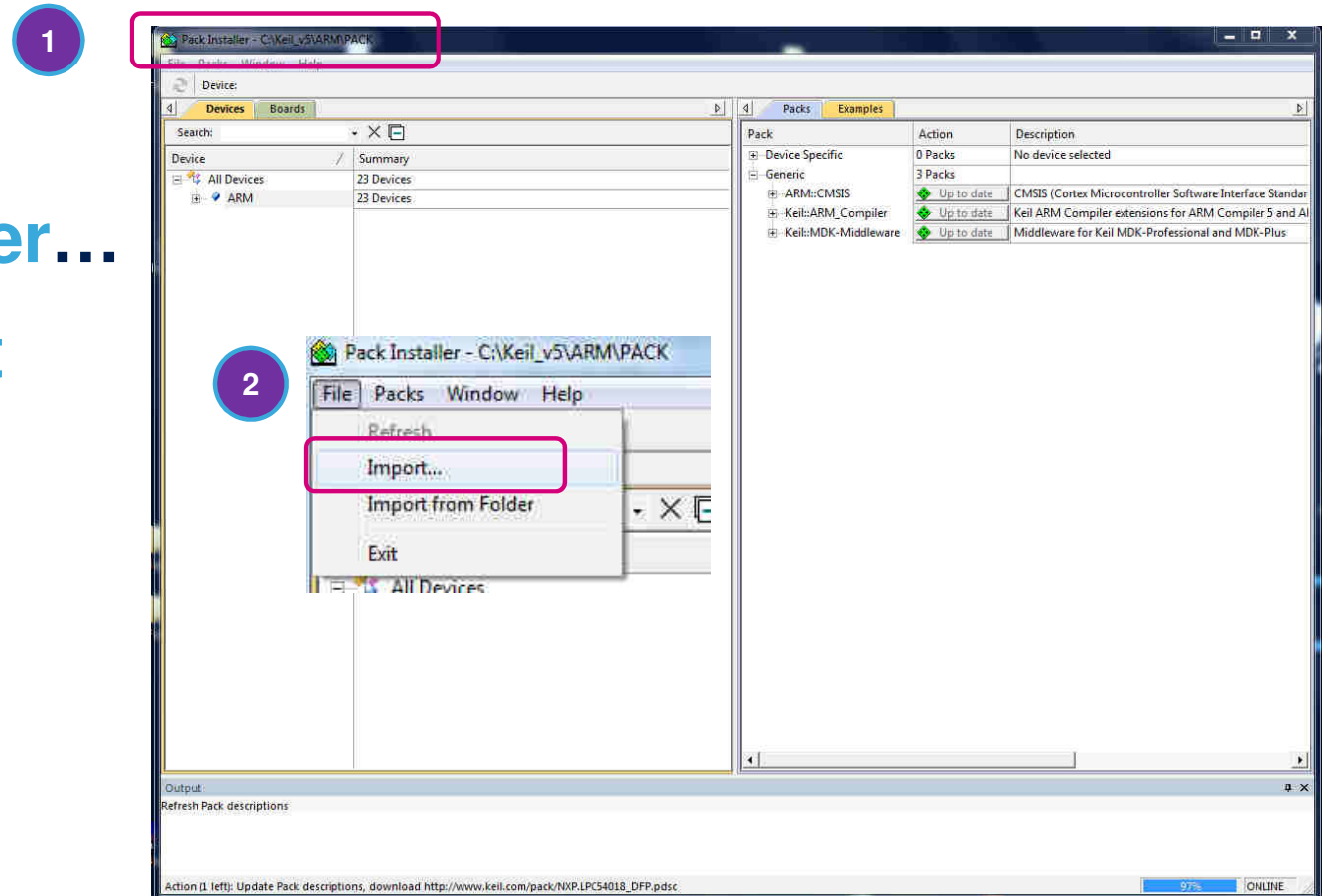
- Open the **Pack Installer**
  - Go to **Project->Manage->Pack Installer**



# Keil ARM MDK Configuration

191

1. In the **Pack Installer...**
2. Go to **File->Import**

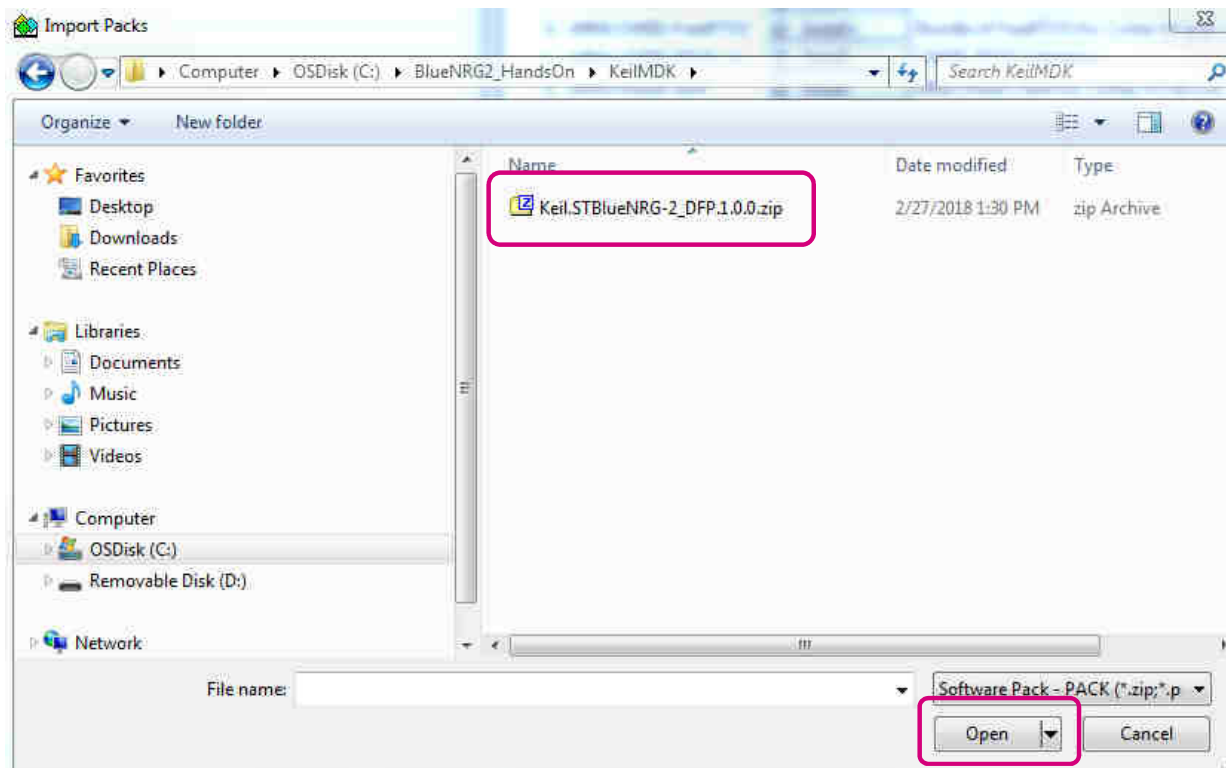




# Keil ARM MDK Configuration

192

3. Open the previously saved .zip file “**Keil.STBlueNRG-2\_DFP.1.0.1.zip**”



# Keil ARM MDK Configuration

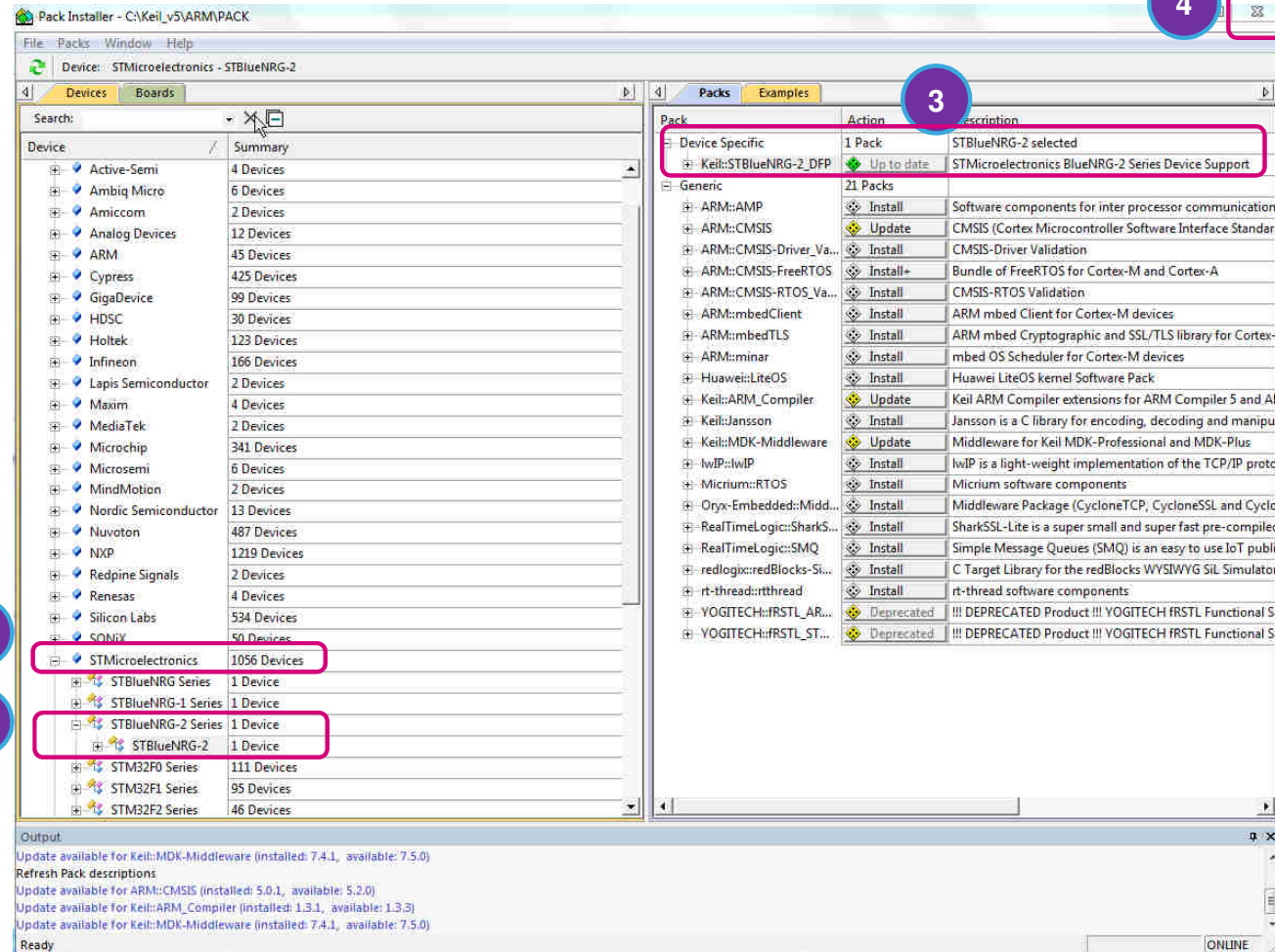
193

4. Once the Device Support pack has been installed, on the **Devices** list in the **left column** select the vendor **STMicroelectronics**

5. Go to **STBlueNRG-2 Series** and select **STBlueNRG-2**

6. Now it will show in green **“Up to date”**

7. Close the **Pack Installer**





# BlueNRG-2 Free License for Keil ARM MDK

# BlueNRG-2 MDK License Activation

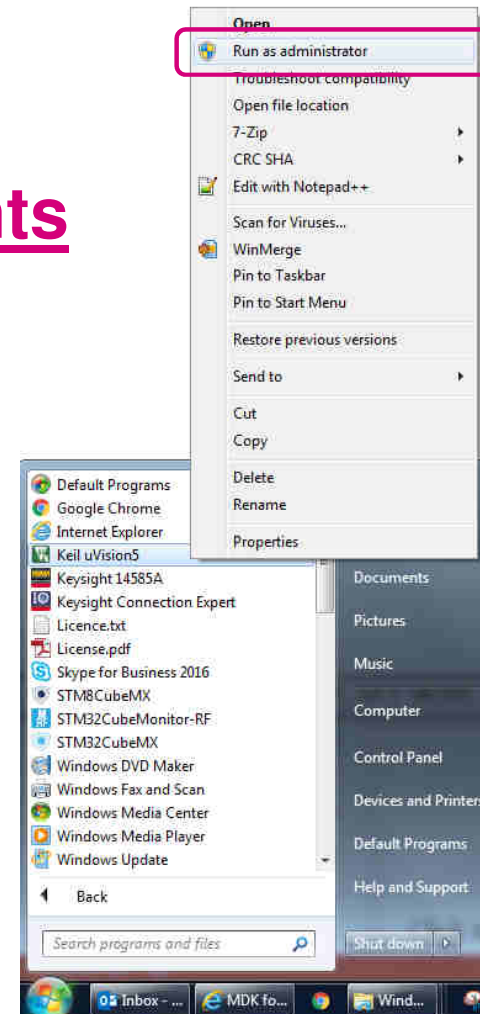
195

- You can refer to the following webpage:
  - <http://www2.keil.com/stmicroelectronics-stm32/mdk>
- Note: it is mentioned STM32L0 and STM32F0, but the same procedure applies to BlueNRG-2 device

# BlueNRG-2 License Installation 1/6

196

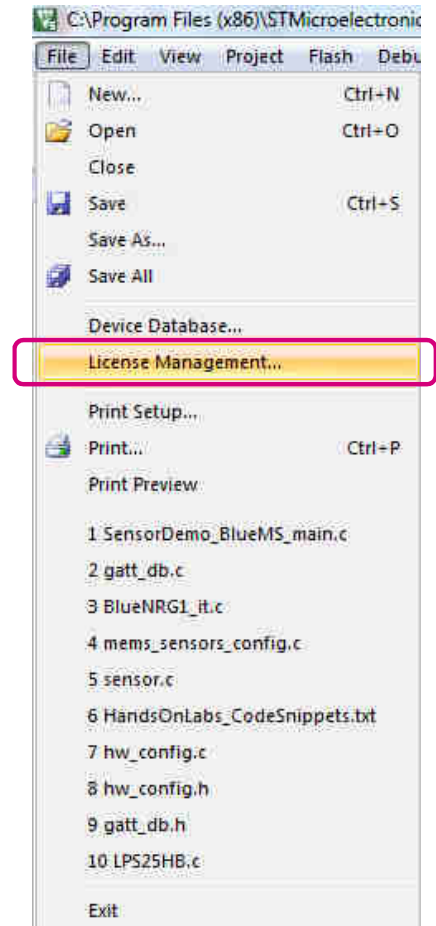
1. **Login** with an account that has **administration rights**
2. Right-click the  $\mu$ Vision icon and select **Run as Administrator...** from the context menu.



# BlueNRG-2 License Installation 2/6

197

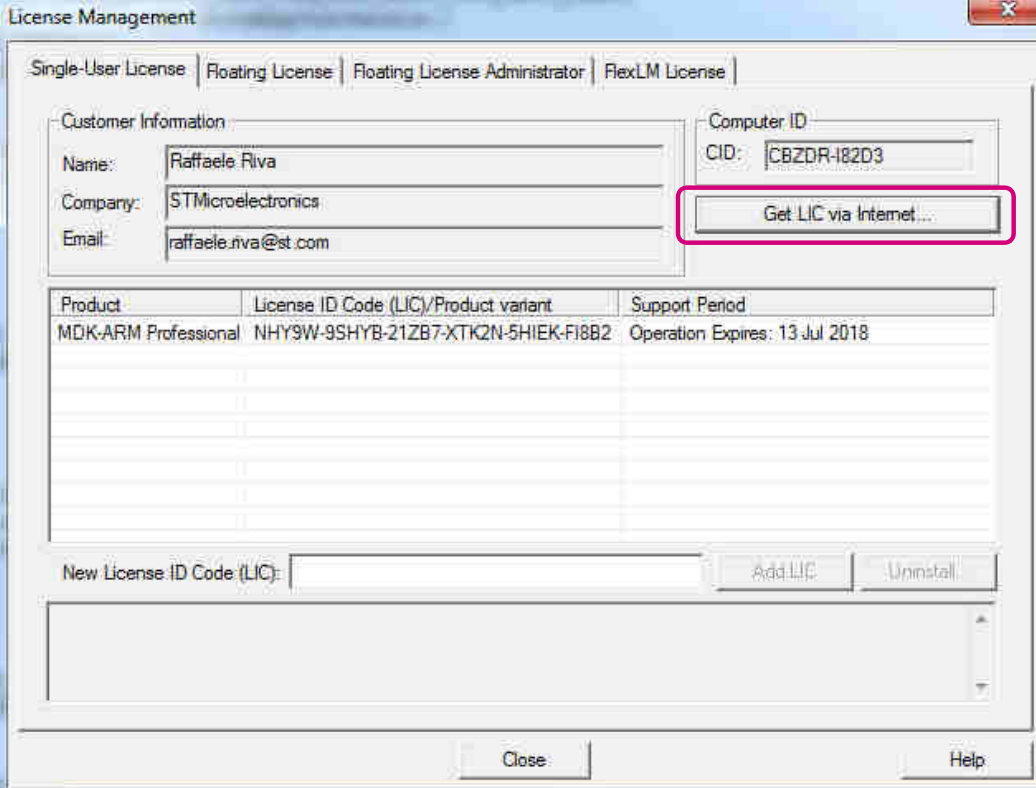
3. Open the dialog **File — License Management...**



# BlueNRG-2 License Installation 3/6

198

4. Select the **Single-User License** tab and click the button **Get LIC via Internet...**



License Management

Single-User License | Floating License | Floating License Administrator | FlexLM License

Customer Information

Name: Raffaele Riva

Company: STMicroelectronics

Email: raffaele.riva@st.com

Computer ID

CID: CBZDR-182D3

Get LIC via Internet...

Product	License ID Code (LIC)/Product variant	Support Period
MDK-ARM Professional	NHY9W-9SHYB-21ZB7-XTK2N-5HIEK-F18B2	Operation Expires: 13 Jul 2018

New License ID Code (LIC):

Add LIC Uninstall

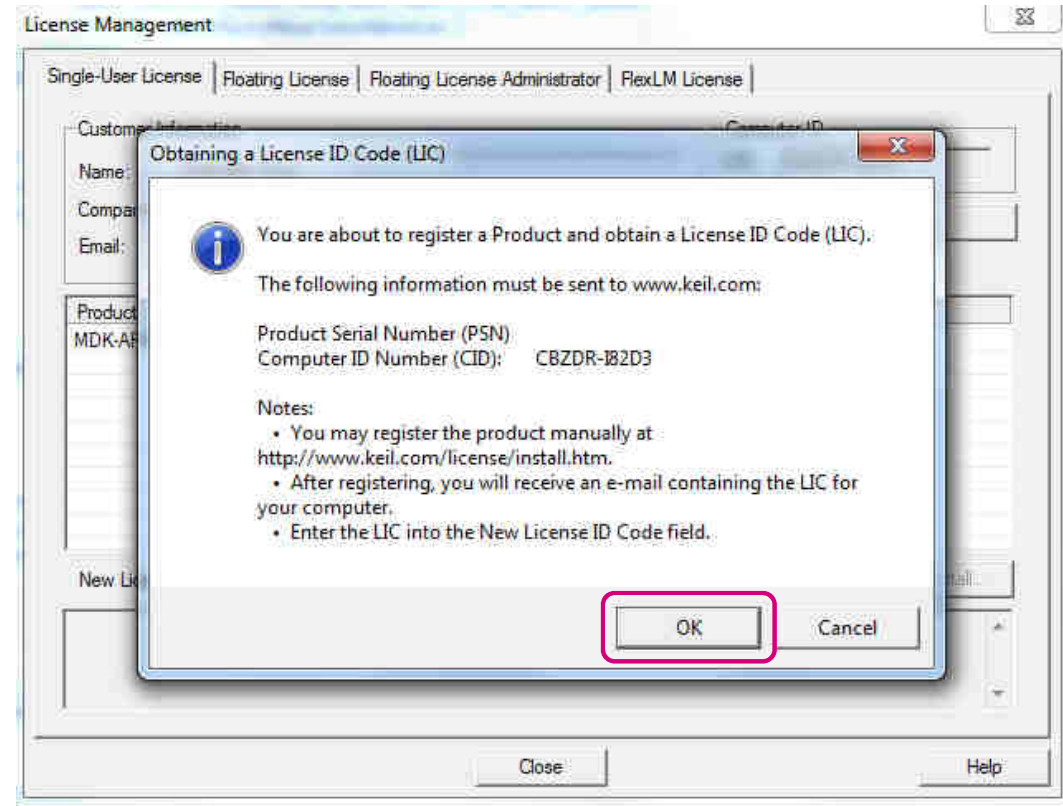
Close Help

# BlueNRG-2 License Installation 4/6

199

5. Click the button **OK** to register the product.

This action opens the License Management page on the Keil web site.





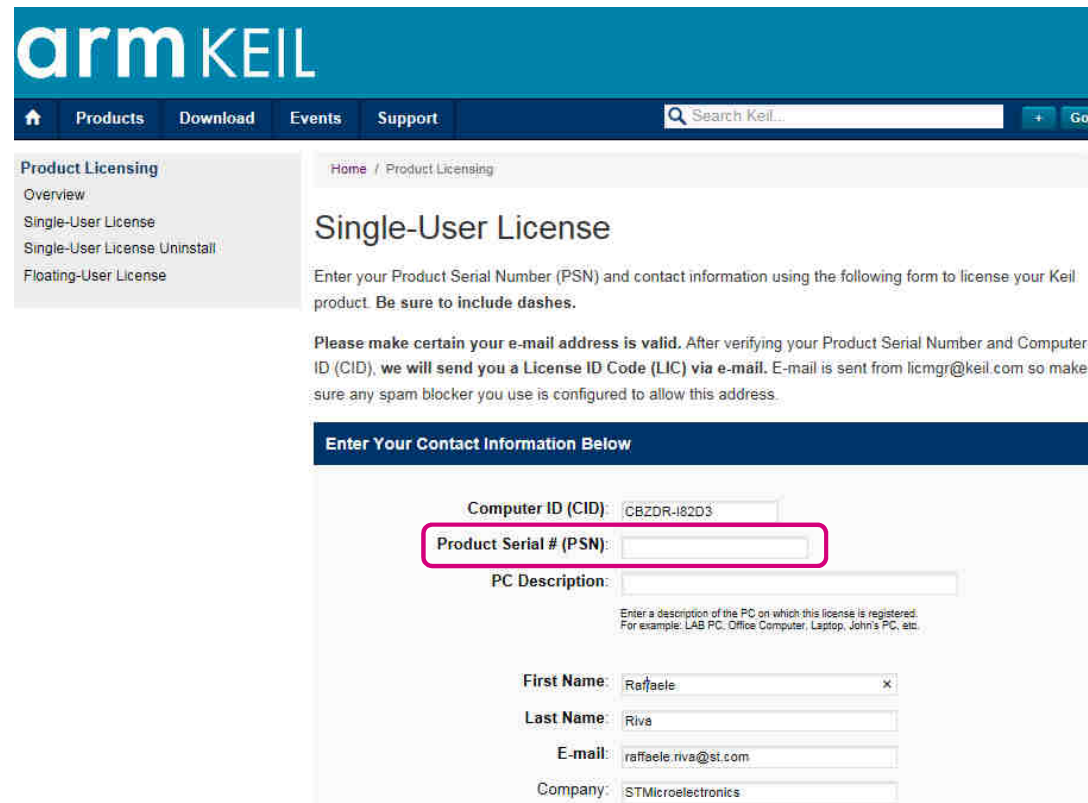
# BlueNRG-2 License Installation 5/6

200

6. For the latest **Product Serial Number** please check on the Keil website at this URL: <http://www2.keil.com/stmicroelectronics-stm32/mdk>

7. Then enter the PSN in this webpage along with your contact information and click the button **Submit**.

8. An e-mail is sent back with the **License ID Code (LIC)** within a few minutes.



The screenshot shows the ARM KEIL website's 'Single-User License' page. The page has a blue header with the 'armKEIL' logo and navigation links: Home, Products, Download, Events, Support, and a search bar. A left sidebar lists 'Product Licensing' options: Overview, Single-User License (selected), Single-User License Uninstall, and Floating-User License. The main content area is titled 'Single-User License' and contains instructions to enter the Product Serial Number (PSN) and contact information. A form titled 'Enter Your Contact Information Below' includes fields for Computer ID (CID), Product Serial # (PSN), PC Description, First Name, Last Name, E-mail, and Company. The PSN field is highlighted with a red rectangle. The form is pre-filled with the following data: CID: CBZDR-I82D3, PSN: (empty), PC Description: (empty), First Name: Raffaele, Last Name: Riva, E-mail: raffaele.riva@st.com, and Company: STMicroelectronics.

armKEIL

Home / Product Licensing

### Single-User License

Enter your Product Serial Number (PSN) and contact information using the following form to license your Keil product. Be sure to include dashes.

Please make certain your e-mail address is valid. After verifying your Product Serial Number and Computer ID (CID), we will send you a License ID Code (LIC) via e-mail. E-mail is sent from licmgr@keil.com so make sure any spam blocker you use is configured to allow this address.

**Enter Your Contact Information Below**

Computer ID (CID): CBZDR-I82D3

Product Serial # (PSN):

PC Description:

Enter a description of the PC on which this license is registered.  
For example: LAB PC, Office Computer, Laptop, John's PC, etc.

First Name: Raffaele

Last Name: Riva

E-mail: raffaele.riva@st.com

Company: STMicroelectronics

# BlueNRG-2 License Installation 6/6

201

9. To activate the Software Product, enter the **LIC** you received by email in the field **New License ID Code (LIC)** and click **Add LIC**.

10. Then click on **Close**

The screenshot shows the 'License Management' window with the following sections:

- Single-User License** | Floating License | Floating License Administrator | FlexLM License
- Customer Information:**
  - Name: Raffaele Riva
  - Company: STMicroelectronics
  - Email: raffaele.riva@st.com
- Computer ID:**
  - CID: CFW23-ATCIR
  - Get LIC via Internet...
- Table of Licenses:**

Product	License ID Code (LIC)/Product variant	Support Period
MDK-ARM Cortex-M0/M0+ 256K for ST	QXJMN-YH8DA-N1Z6J-7RXNJ-VUYX7-SU4RH	Expires: Jan 2019
- New License ID Code (LIC):** QXJMN-YH8DA-N1Z6J-7RXNJ-VUYX7-SU4RH **Add LIC** **Uninstall...**
- Close** **Help**