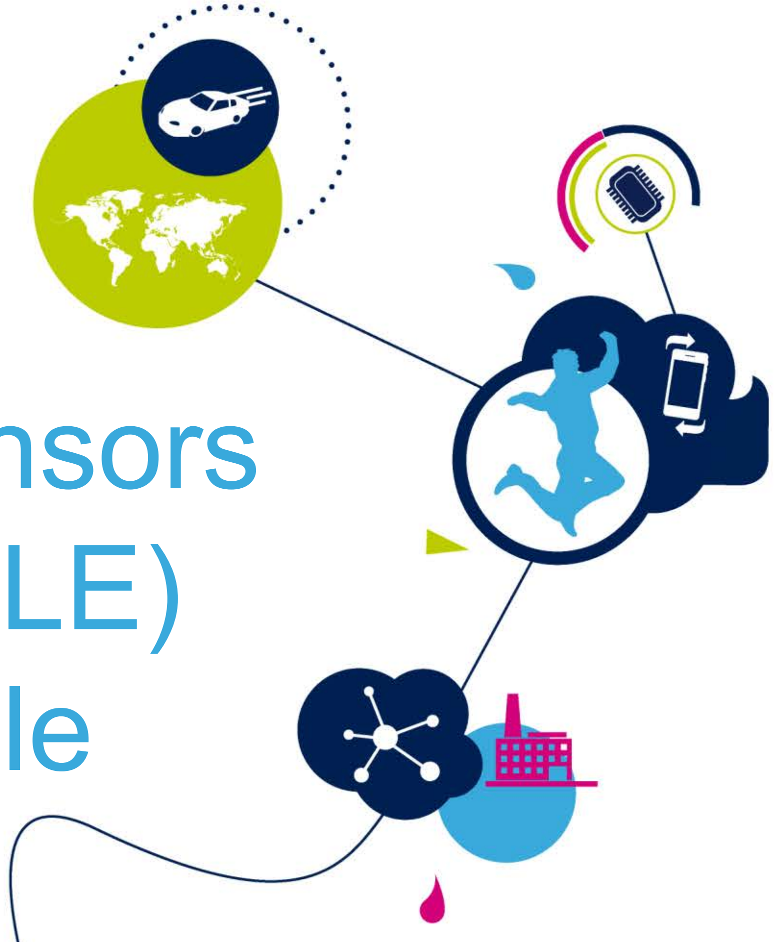
Simplify the Integration of Sensors and Bluetooth Low Energy (BLE) Connectivity Using the BlueTile Eval Kit

IoT Systems Development - Connectivity



WiFi Network: **STMicro**
Password: **STMTTT19**

life.augmented

ST Developers Conference

September 12th, 2019
Santa Clara Convention Center - Mission City Ballroom
Santa Clara, CA

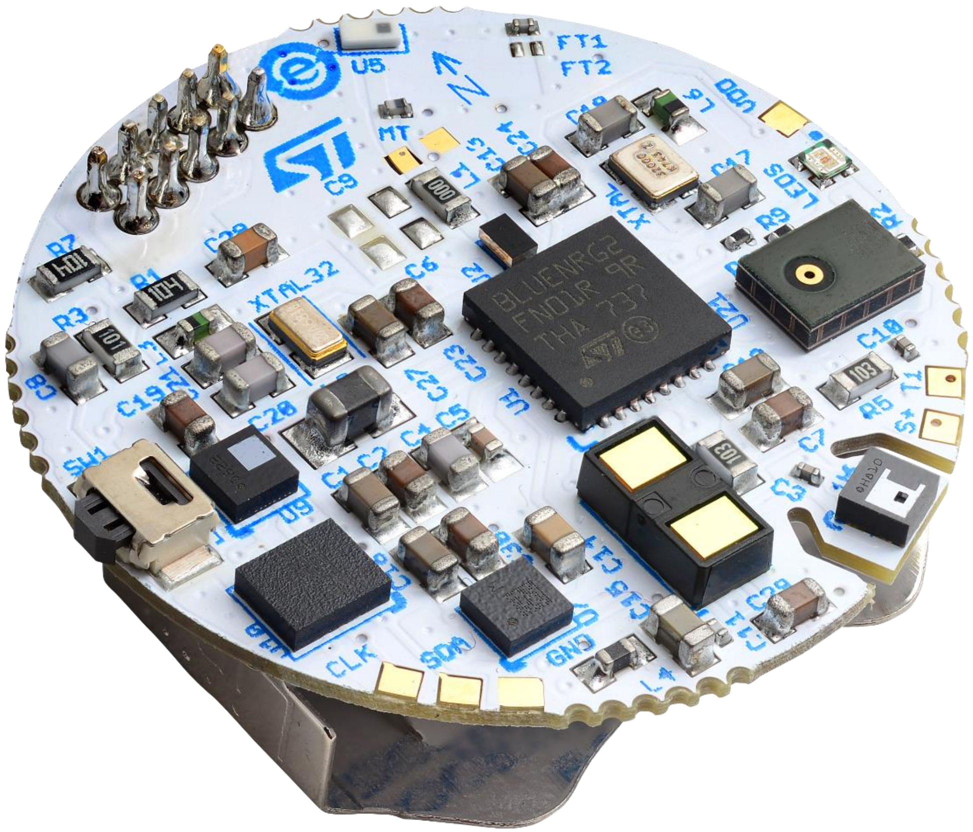
Agenda

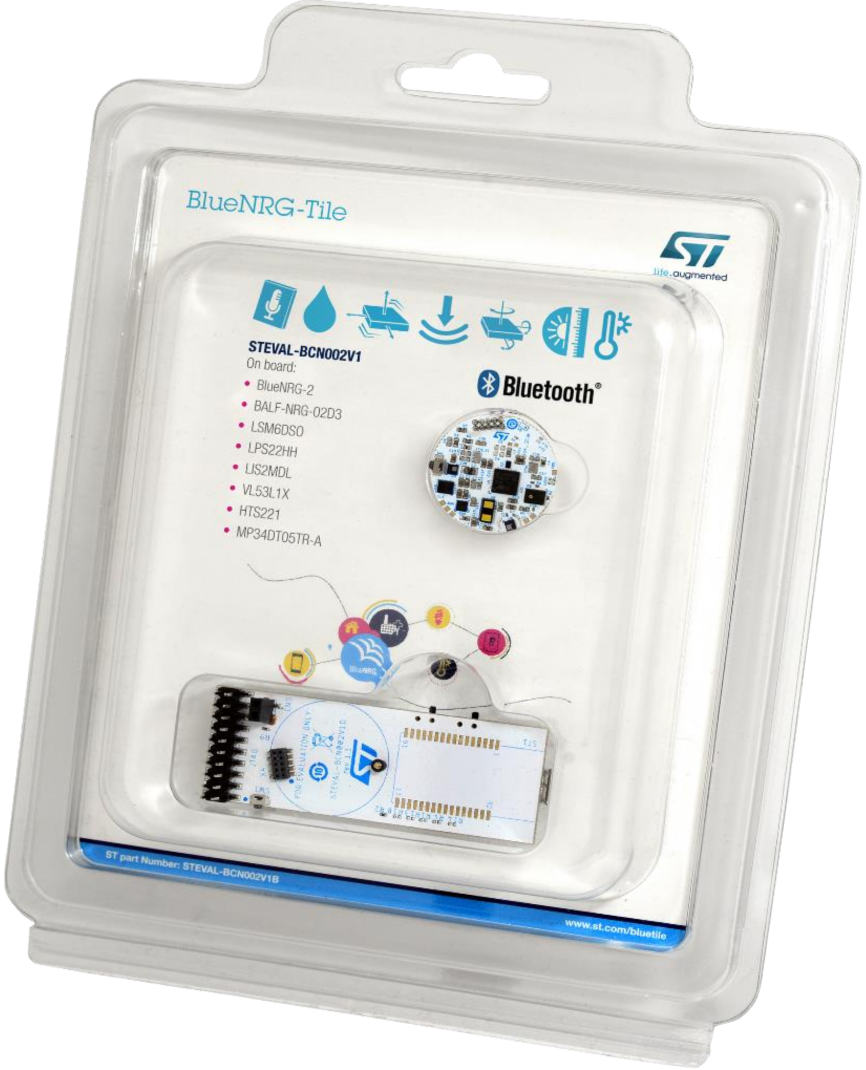
BlueTile
(STEVAL-BCN002V1)

WiFi Network: **STMicro**
Password: **STMTTT19**

- 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

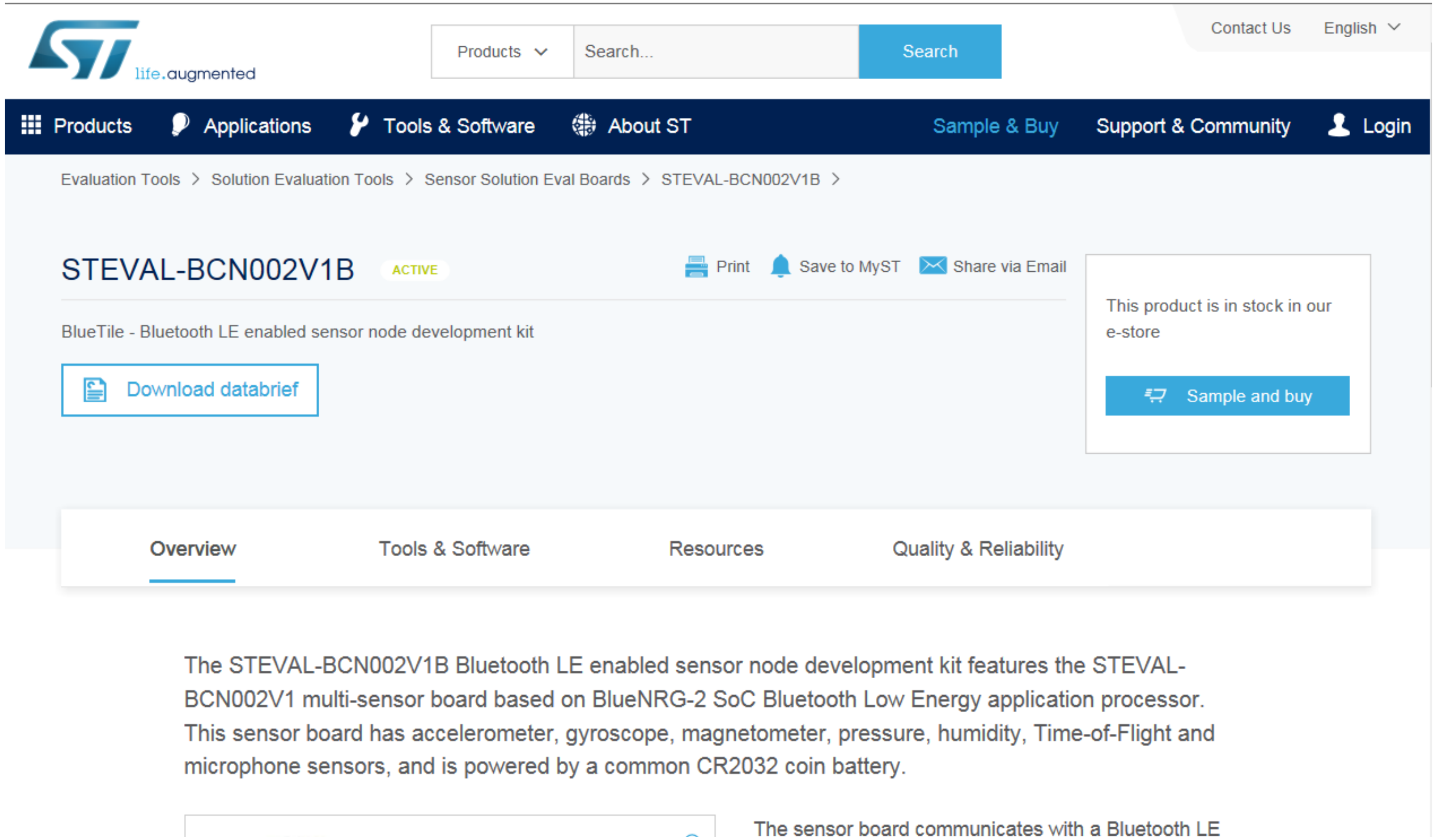
BlueTile (STEVAL-BCN002V1B)





On St.com

http://www.st.com/bluetile



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

STEVAL-BCN002V1B ACTIVE

BlueTile - Bluetooth LE enabled sensor node development kit

This product is in stock in our e-store

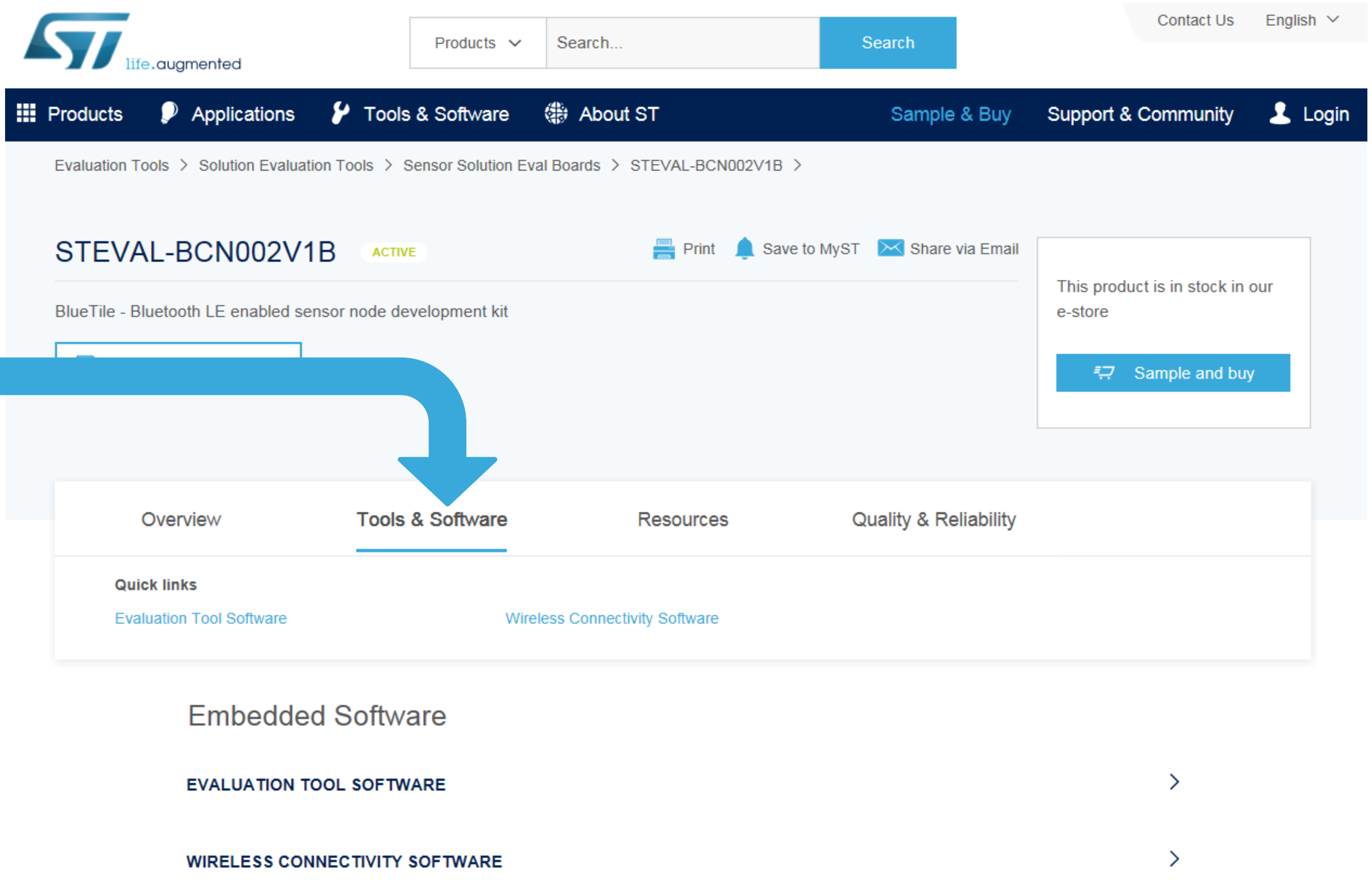
The STEVAL-BCN002V1B Bluetooth LE enabled sensor node development kit features the STEVAL-BCN002V1 multi-sensor board based on BlueNRG-2 SoC Bluetooth Low Energy application processor. This sensor board has accelerometer, gyroscope, magnetometer, pressure, humidity, Time-of-Flight and microphone sensors,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

The sensor board communicates with a Bluetooth LE

BlueTile SDK

http://www.st.com/bluetile

Software Development Kit



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

STEVAL-BCN002V1B ACTIVE

BlueTile - Bluetooth LE enabled sensor node development kit

This product is in stock in our e-store

Quick links

Evaluation Tool Software

Wireless Connectivity Software

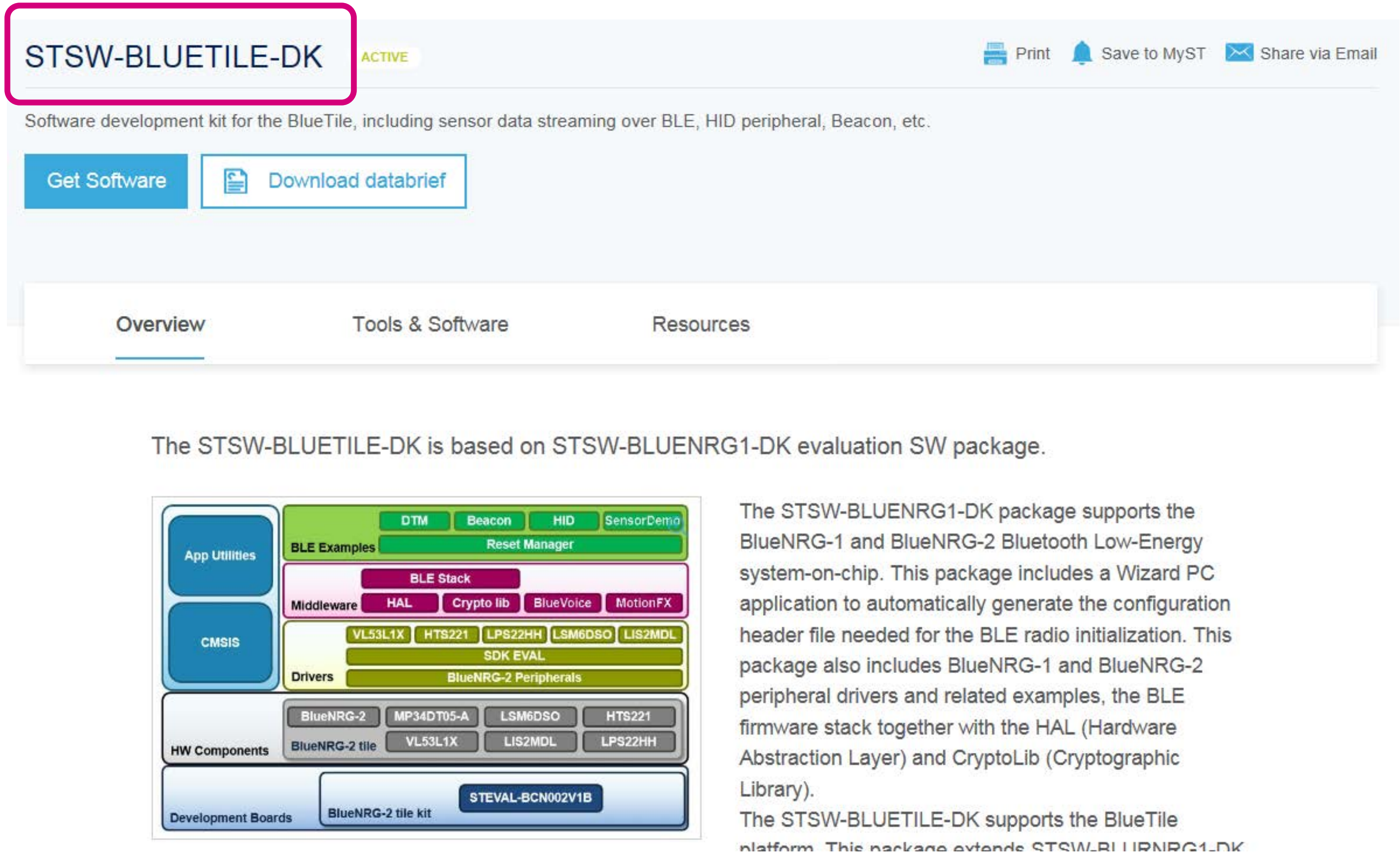
Embedded Software

EVALUATION TOOL SOFTWARE

WIRELESS CONNECTIVITY SOFTWARE

BlueTile SDK

https://www.st.com/content/st_com/en/products/embedded-software/evaluation-tool-software/stsw-bluetile-dk.html



STSW-BLUETILE-DK ACTIVE

Software development kit for the BlueTile, including sensor data streaming over BLE, HID peripheral, Beacon, etc.

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

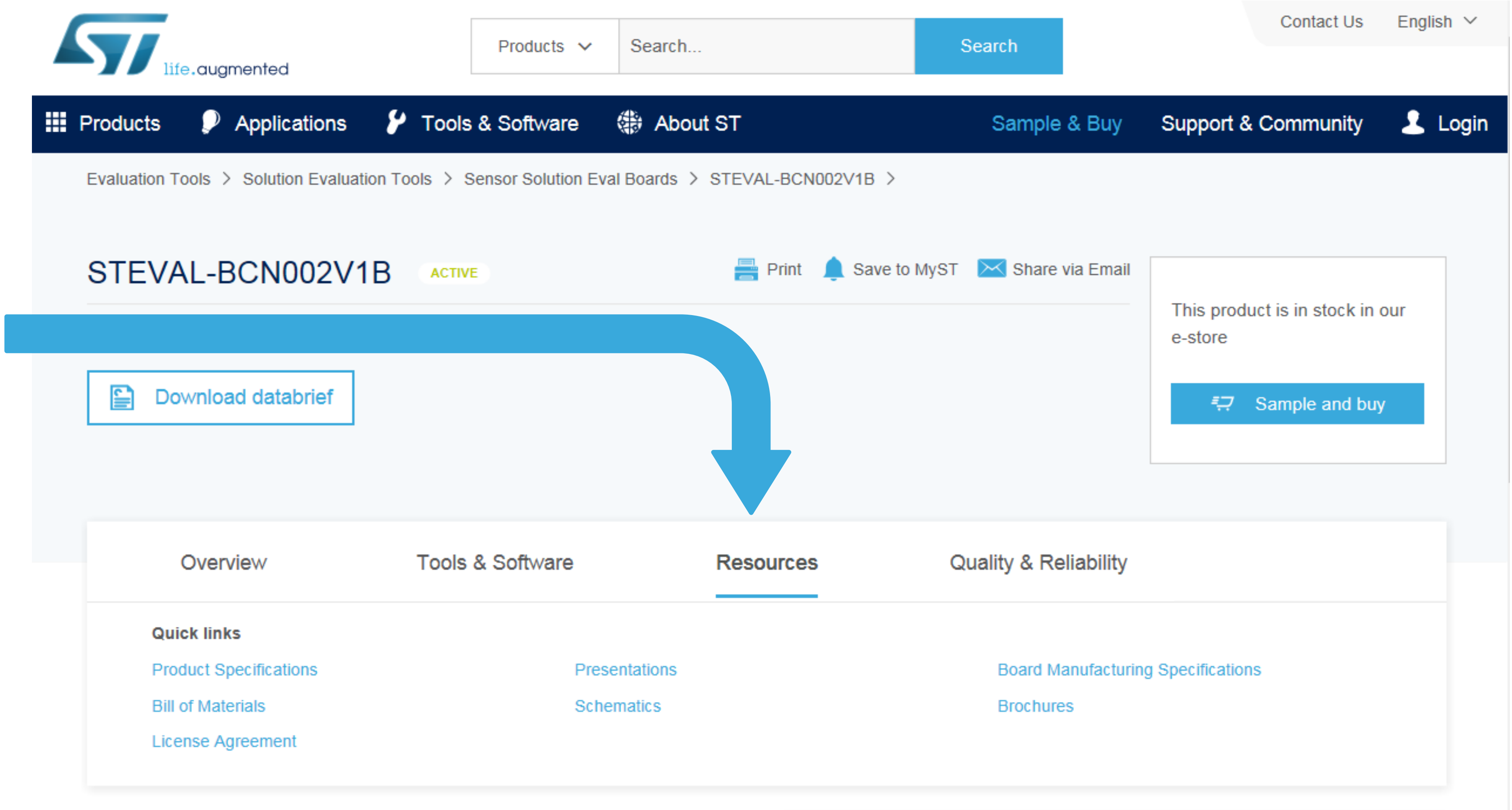
The STSW-BLUENRG1-DK package supports the BlueNRG-1 and BlueNRG-2 Bluetooth Low-Energy system-on-chip. This package includes a Wizard PC application to automatically generate the configuration header file needed for the BLE radio initialization. This package also includes BlueNRG-1 and BlueNRG-2 peripheral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

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

On St.com

http://www.st.com/bluetile

Schematic
Gerbers
BOM



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

STEVAL-BCN002V1B ACTIVE

This product is in stock in our e-store

Quick links

Product Specifications	Presentations	Board Manufacturing Specifications
Bill of Materials	Schematics	Brochures
License Agreement		

Companion ST BLE Sensor App

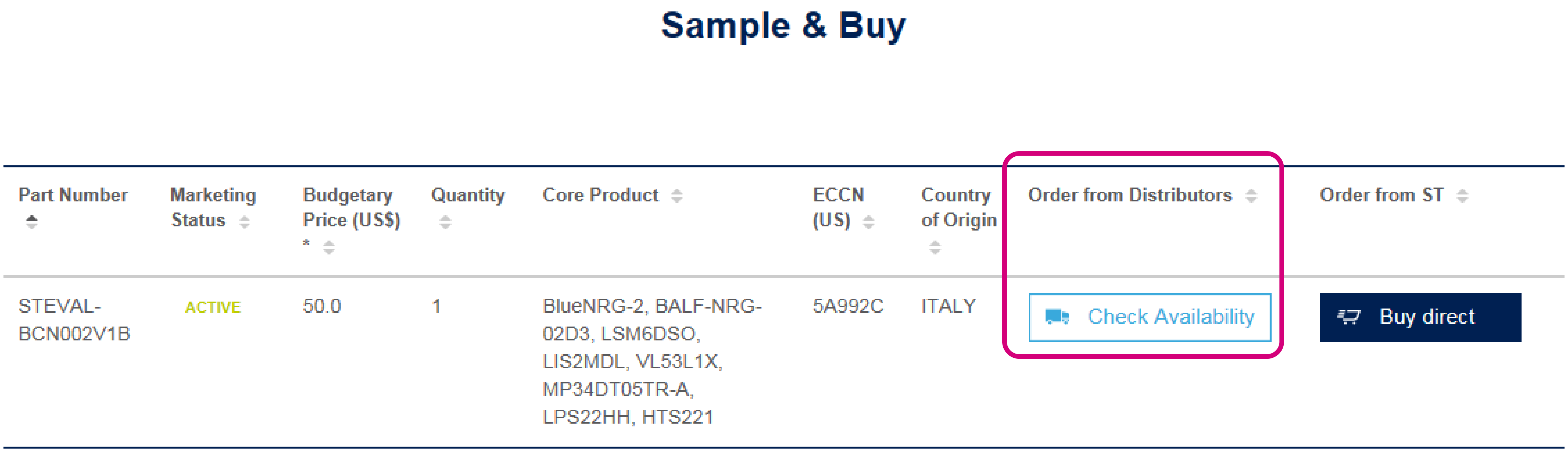
ON YOUR PHONE

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

Download on the App Store

GET IT ON Google Play

Where to Get It

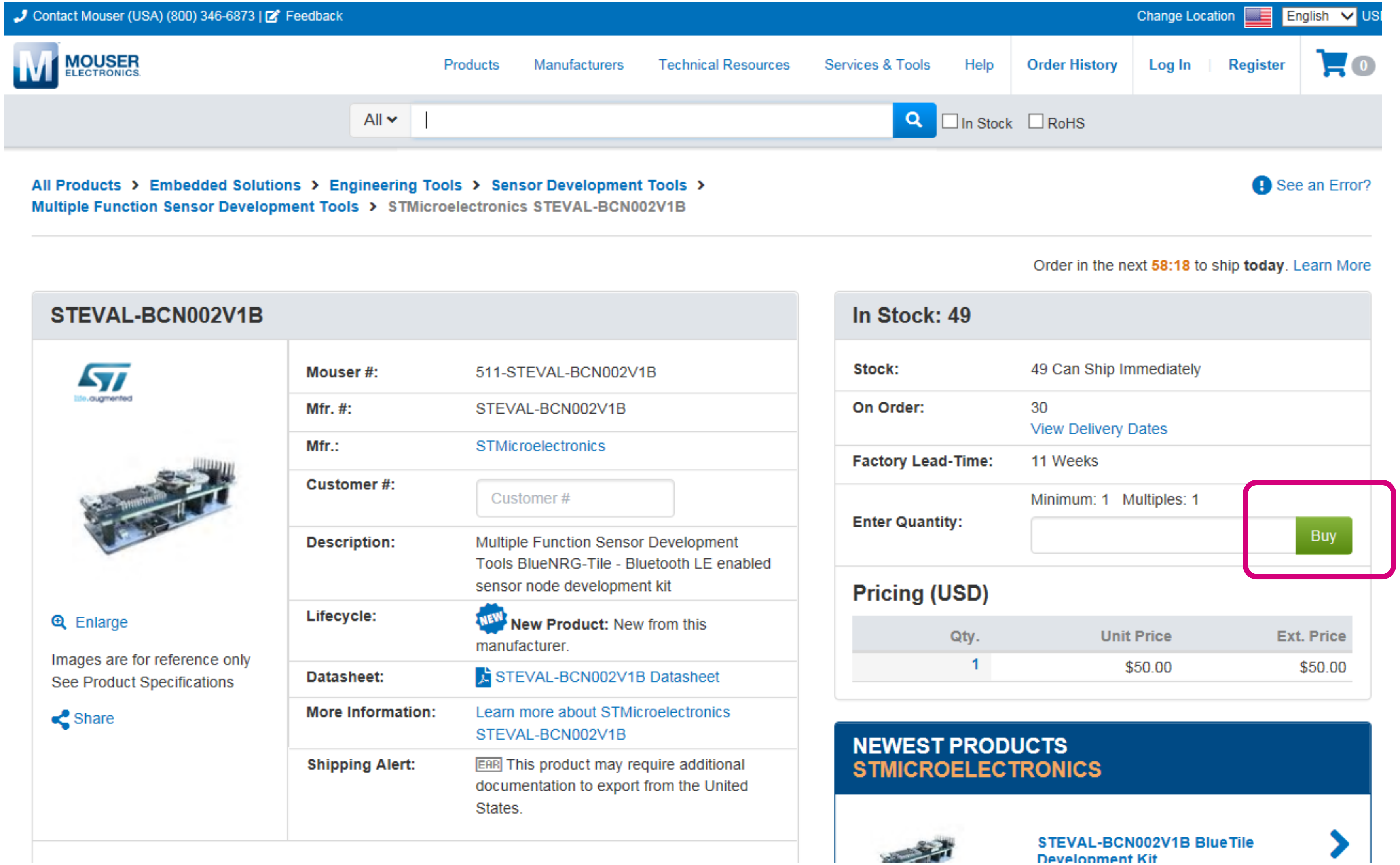


Sample & Buy

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

life.augmented

Where to Get It



All Products > Embedded Solutions > Engineering Tools > Sensor Development Tools > Multiple Function Sensor Development Tools > STMicroelectronics STEVAL-BCN002V1B

See an Error?

Order in the next 58:18 to ship today. Learn More

STEVAL-BCN002V1B

Mouser #:	511-STEVAL-BCN002V1B
Mfr. #:	STEVAL-BCN002V1B
Mfr.:	STMicroelectronics
Customer #:	Customer #
Description:	Multiple Function Sensor Development Tools BlueNRG-Tile - Bluetooth LE enabled sensor node development kit
Lifecycle:	NEW New Product: New from this manufacturer.
Datasheet:	STEVAL-BCN002V1B Datasheet
More Information:	Learn more about STMicroelectronics STEVAL-BCN002V1B
Shipping Alert:	EAR This product may require additional documentation to export from the United States.

Enlarge

Images are for reference only
See Product Specifications

Share

In Stock: 49

Stock:	49 Can Ship Immediately
On Order:	30 View Delivery Dates
Factory Lead-Time:	11 Weeks
Enter Quantity:	Minimum: 1 Multiples: 1 Buy

Pricing (USD)

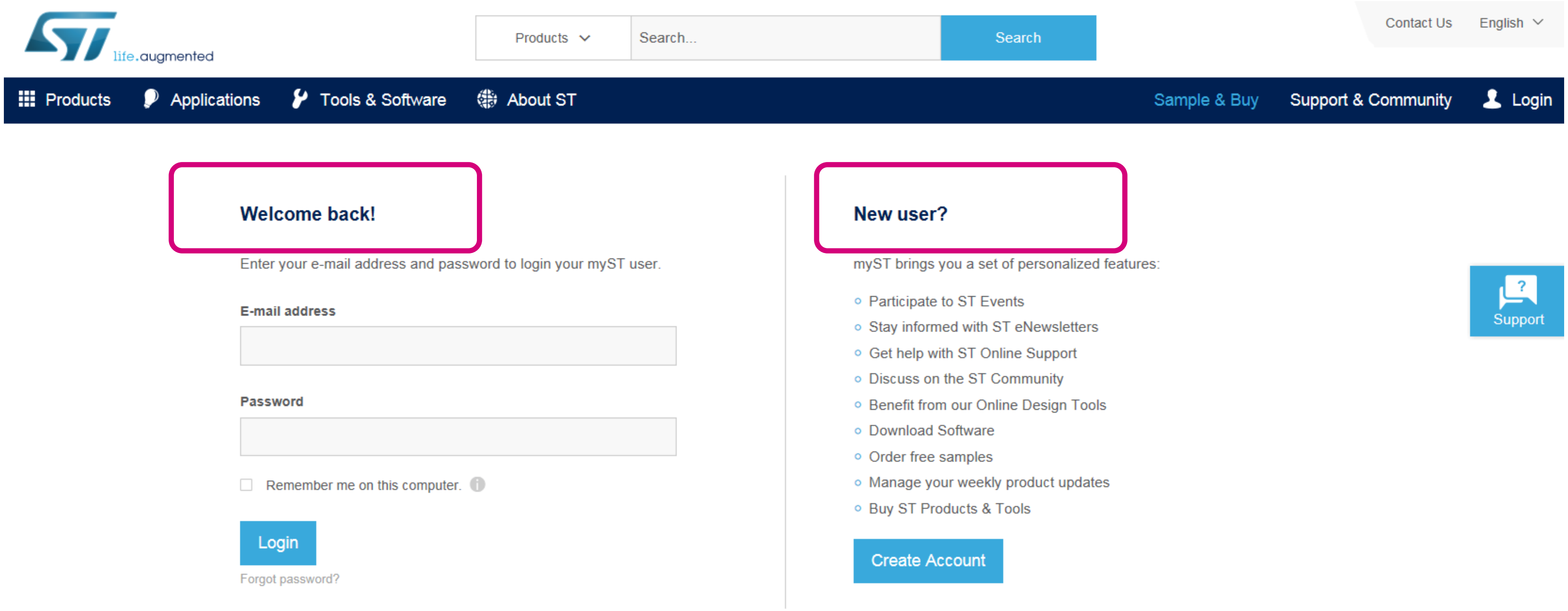
Qty.	Unit Price	Ext. Price
1	$50.00	$50.00

NEWEST PRODUCTS
STMICROELECTRONICS

STEVAL-BCN002V1B BlueTile Development Kit

Online Support

https://my.st.com/ols



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?

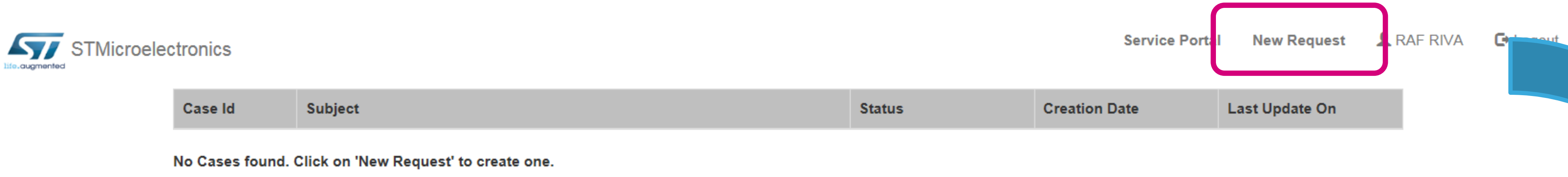
New user?

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

Online Support

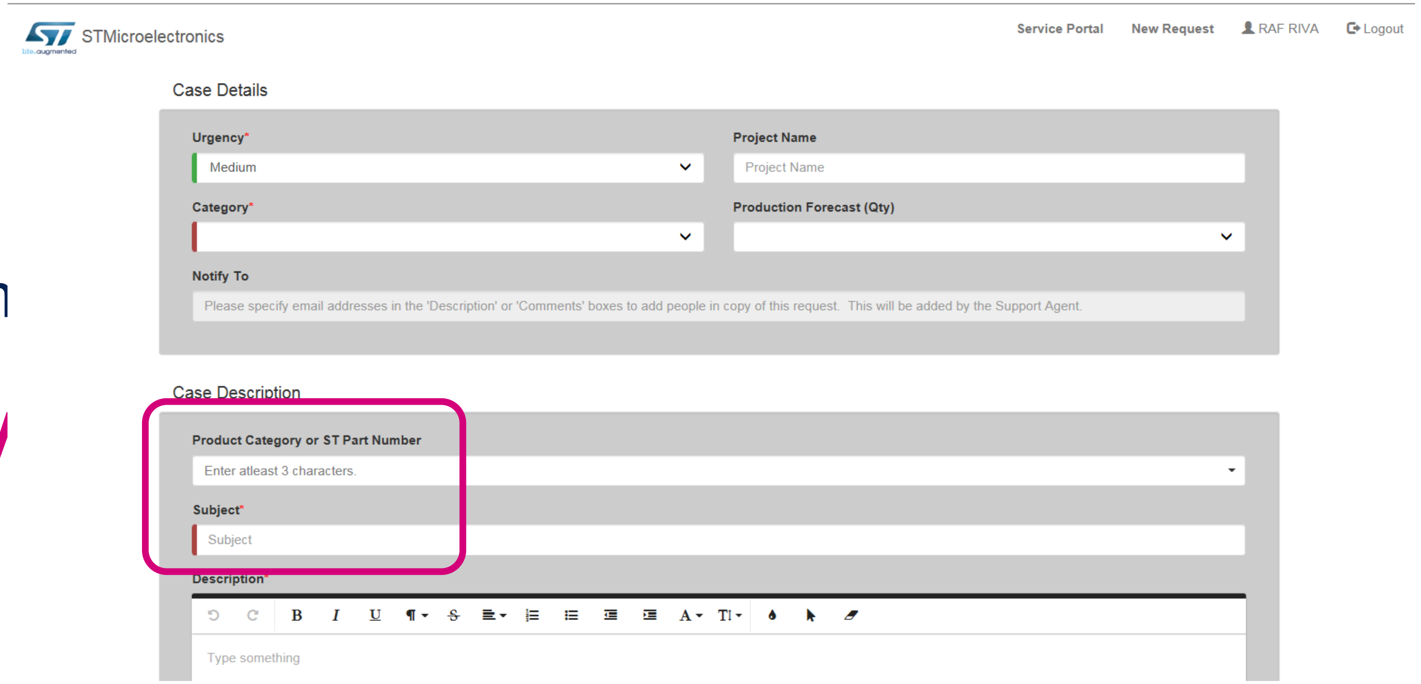

STMicroelectronics

life.augmented

Service Port l | New Request | RAF RIVA

Case Id	Subject	Status	Creation Date	Last Update On

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


STMicroelectronics

Service Portal | New Request | RAF RIVA | Logout

Case Details

Urgency*: Medium

Project Name: Project Name

Category*:

Production Forecast (Qty):

Notify To: 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

Product Category or ST Part Number: Enter atleast 3 characters.

Subject*: Subject

Description*

Type something

Insert the Part Num

STEVAL-BCN002V

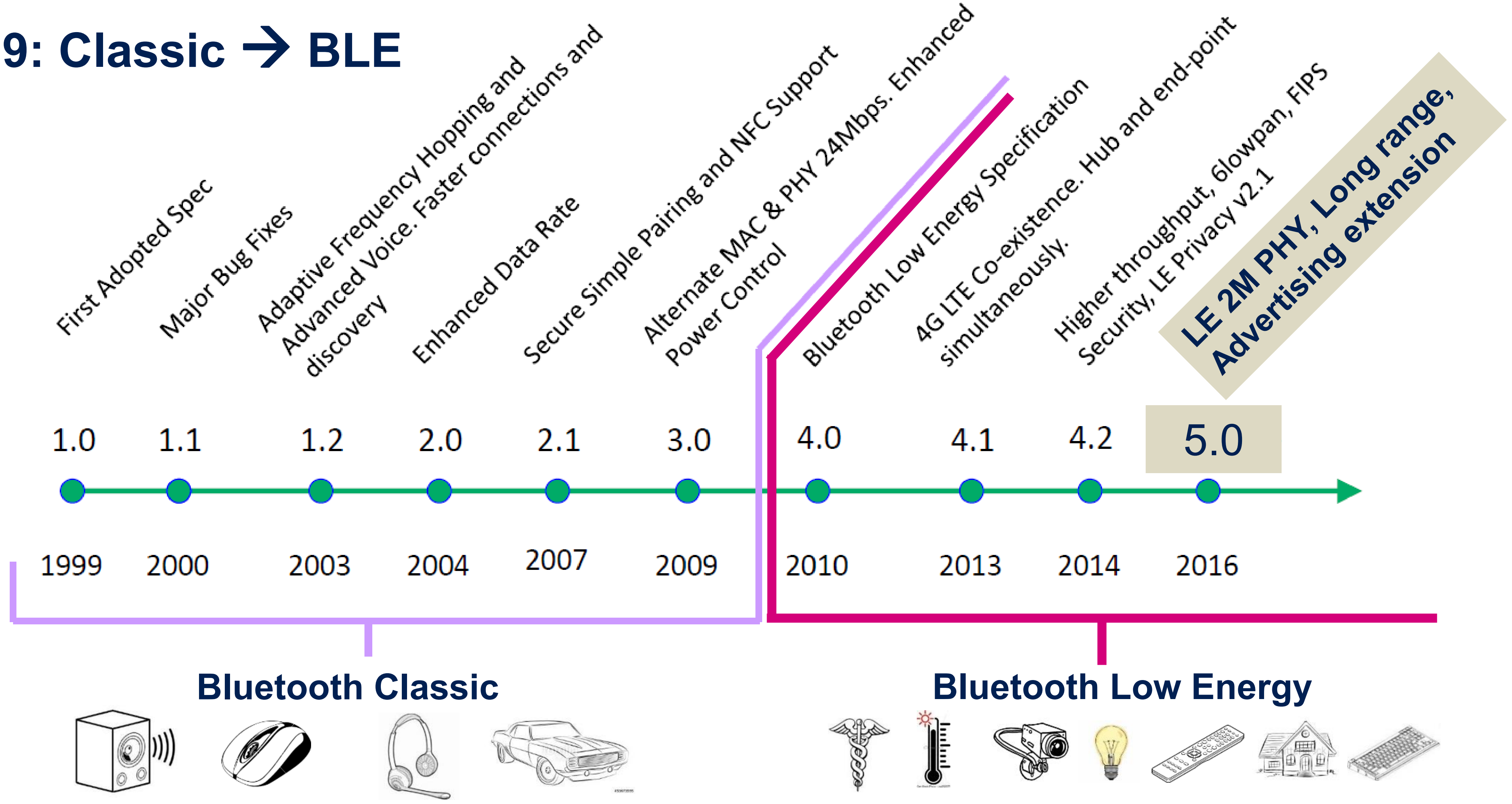
Introduction to

Bluetooth Low Energy

life.augmented

Bluetooth® Evolution

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



First Adopted Spec

Major Bug Fixes

Adaptive Frequency Hopping and Advanced Voice. Faster connections and discovery

Enhanced Data Rate

Secure Simple Pairing and NFC Support

Alternate MAC & PHY 24Mbps. Enhanced Power Control

Bluetooth Low Energy Specification

4G LTE Co-existence. Hub and end-point simultaneously.

Higher throughput, 6lowpan, FIPS Security, LE Privacy v2.1

LE 2M PHY, Long range, Advertising extension

1.0 1.1 1.2 2.0 2.1 3.0 4.0 4.1 4.2 5.0

1999 2000 2003 2004 2007 2009 2010 2013 2014 2016

Bluetooth Classic

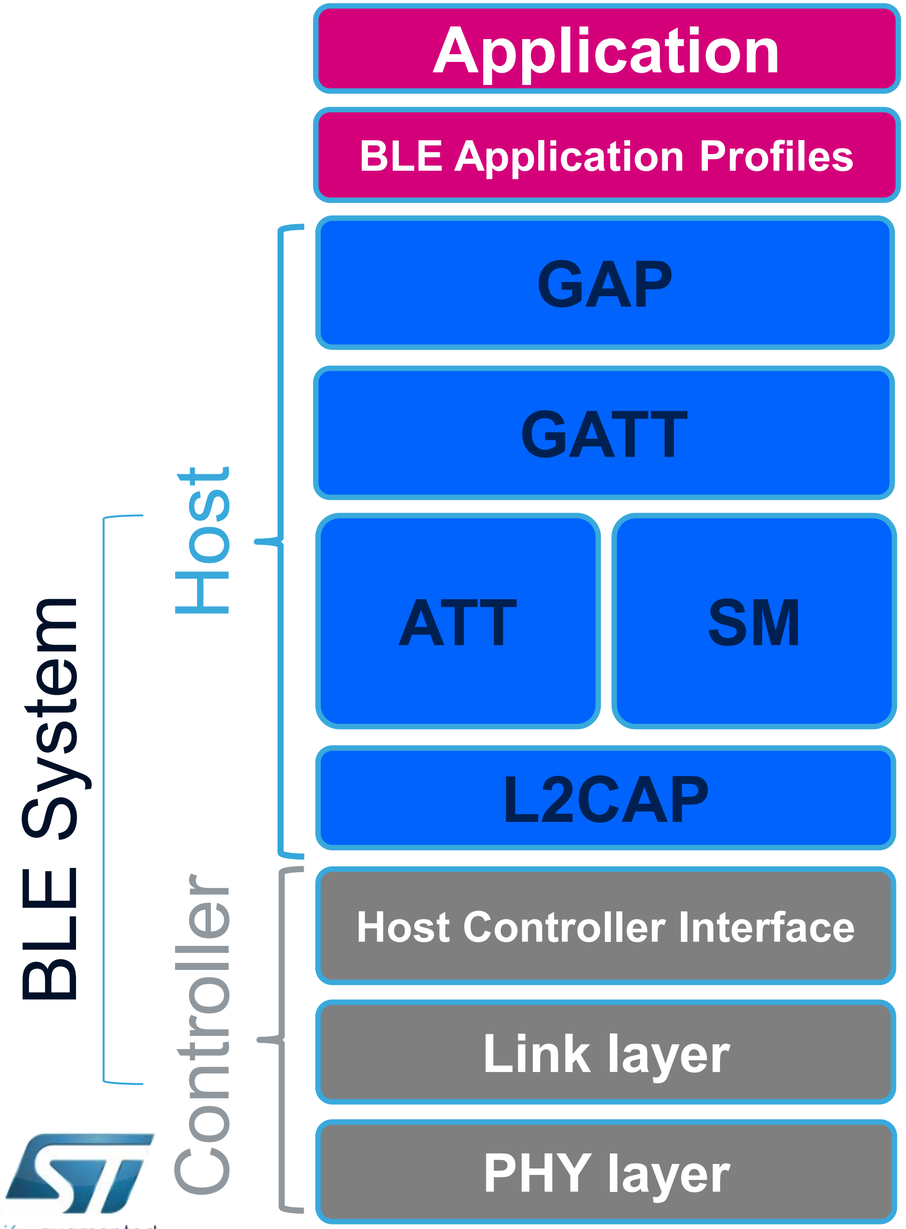
Bluetooth Low Energy

Bluetooth Low Energy (LE)

Designed for Success

- 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

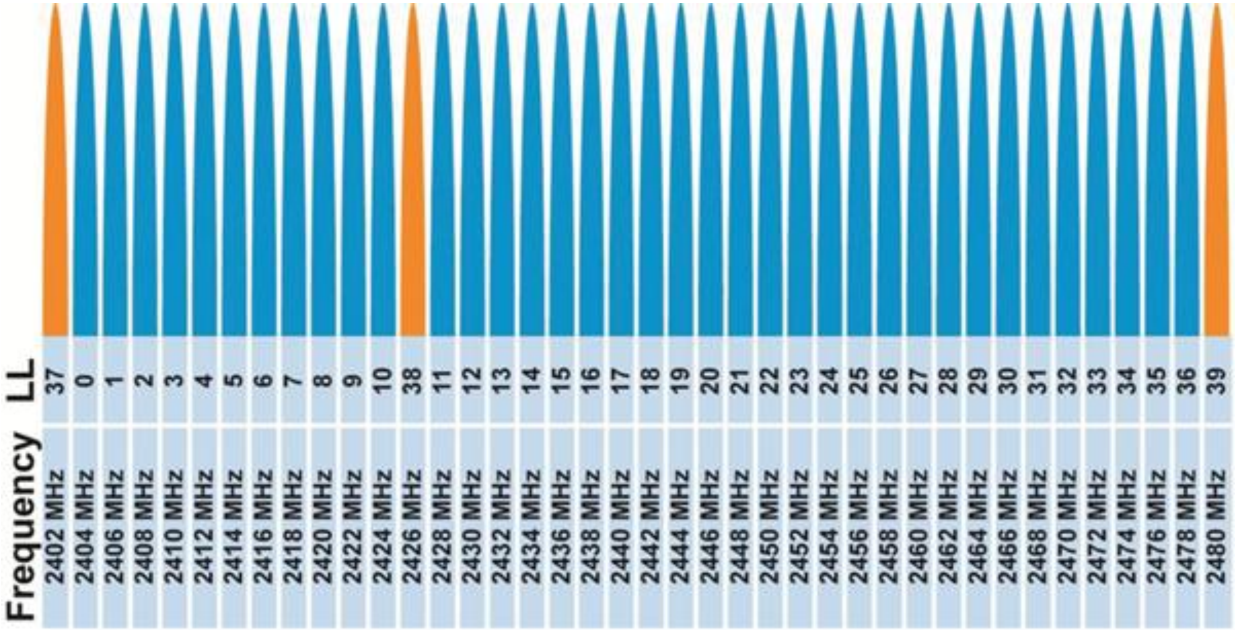
Protocol Stack



- **Generic Acces 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

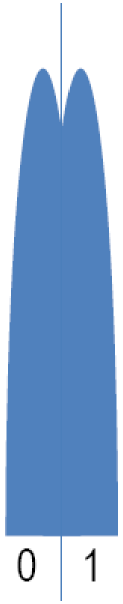
Protocol Stack: PHY

- 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



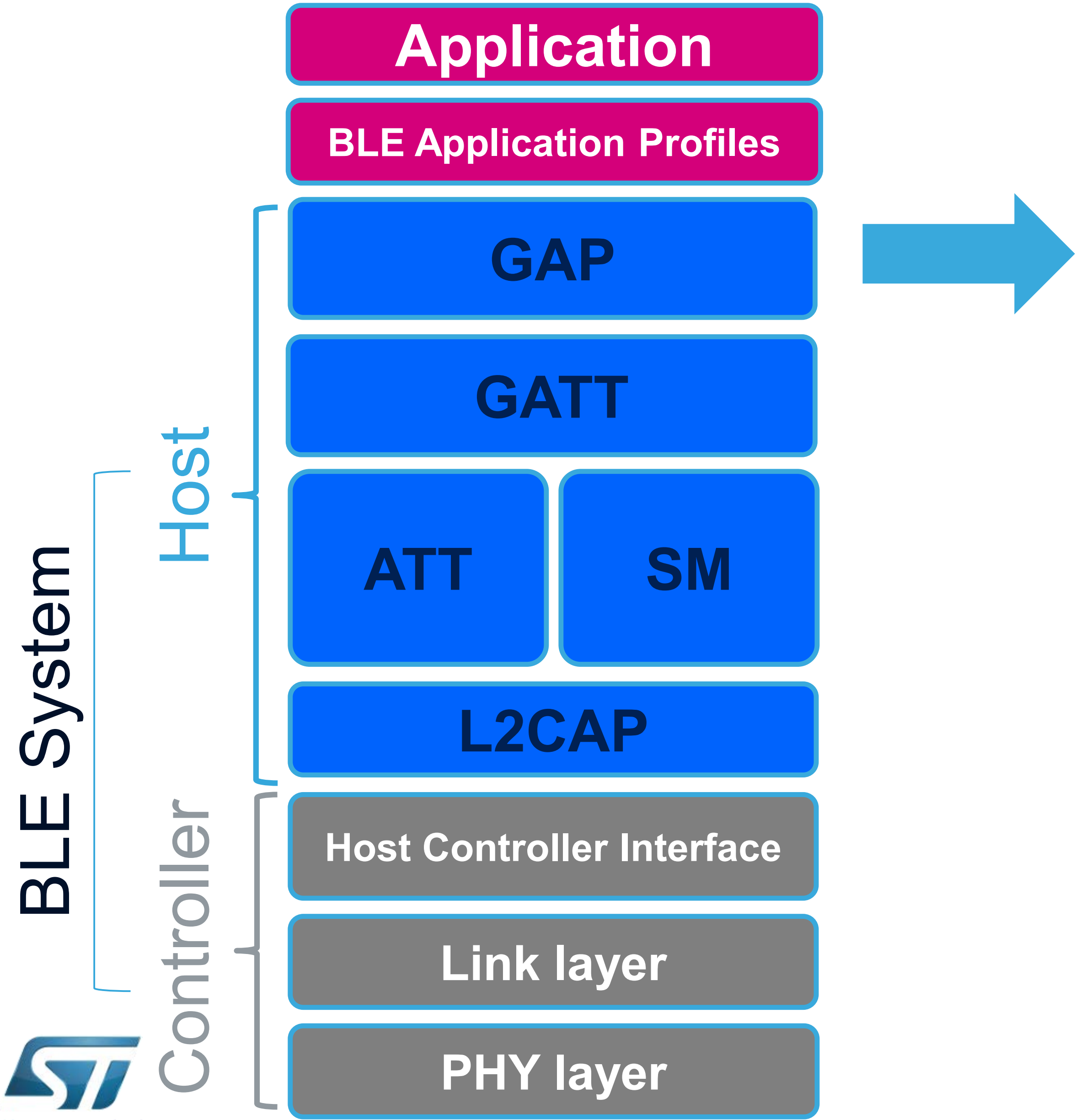
LL	37	0	1	2	3	4	5	6	7	8	9	10	38	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	39
Frequency	2402 MHz	2404 MHz	2406 MHz	2408 MHz	2410 MHz	2412 MHz	2414 MHz	2416 MHz	2418 MHz	2420 MHz	2422 MHz	2424 MHz	2426 MHz	2428 MHz	2430 MHz	2432 MHz	2434 MHz	2436 MHz	2438 MHz	2440 MHz	2442 MHz	2444 MHz	2446 MHz	2448 MHz	2450 MHz	2452 MHz	2454 MHz	2456 MHz	2458 MHz	2460 MHz	2462 MHz	2464 MHz	2466 MHz	2468 MHz	2470 MHz	2472 MHz	2474 MHz	2476 MHz	2478 MHz	2480 MHz

Source : Bluetooth® SIG



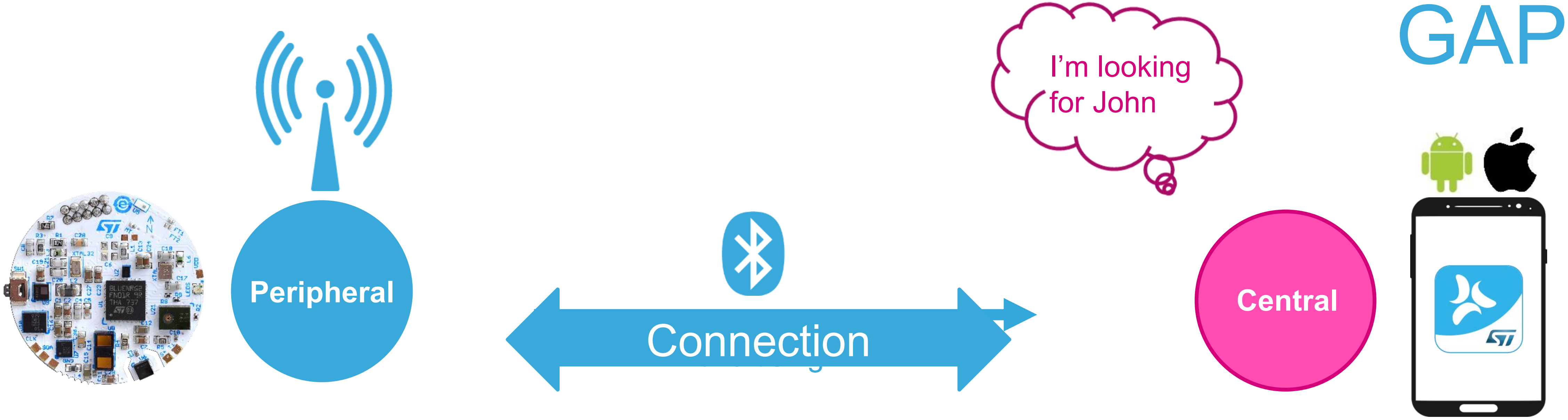
- **GFSK Modulation**
  - BT = 0.5
  - Modulation Index = 0.5
  - “pulse shaping” Gaussian filter “smooths” transitions from zero to one reduces spectral width

Protocol Stack - GAP



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

GAP



Peripheral is in Advertising mode.
Sends Advertising packets.

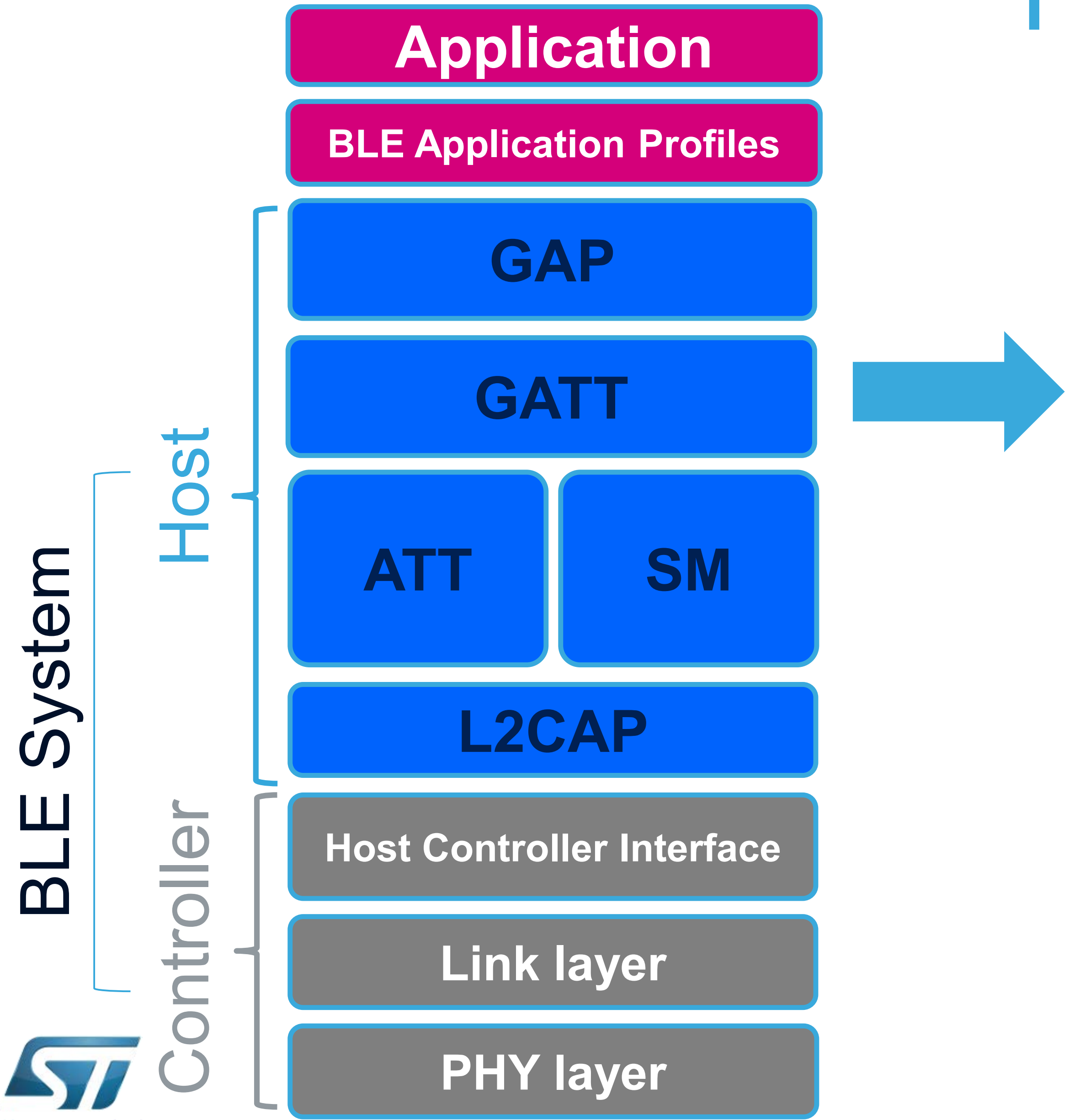
Central is **Scanning**.
Look for known peripherals by listening to advertising packets.

My name is John
My capabilities are X, Y, Z, etc…

Advertising **packets consists of multiple information units known as AD types**.
E.g.: Local Name, TX Output Power, Manufacturer Specific info, etc…

As the John peripheral is found, it initiates a BLE connection.

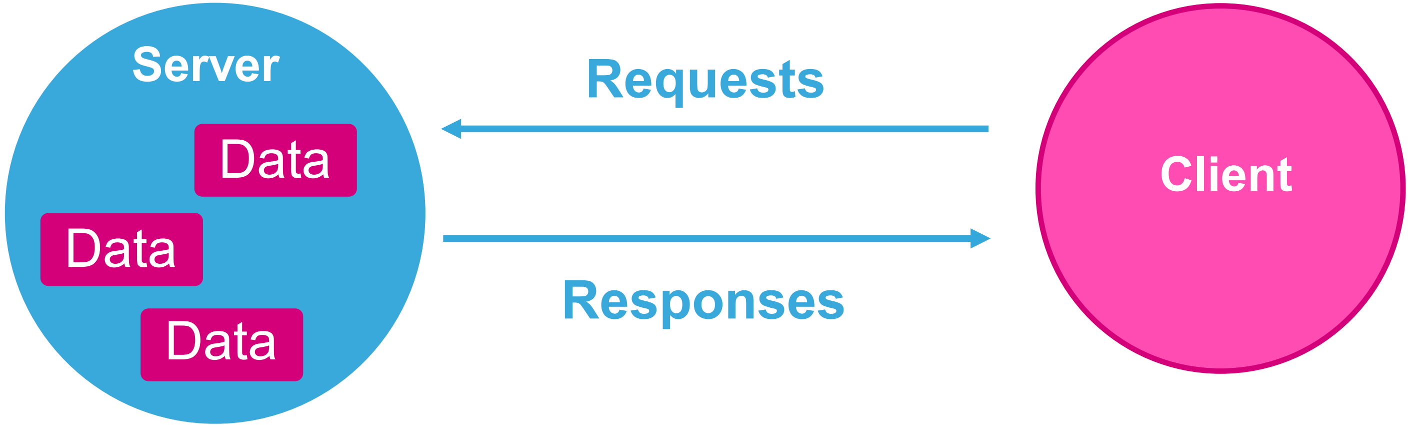
Protocol Stack - GATT



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

GATT

Server contains the data.
Receives requests, executes, responds.
Can indicate value

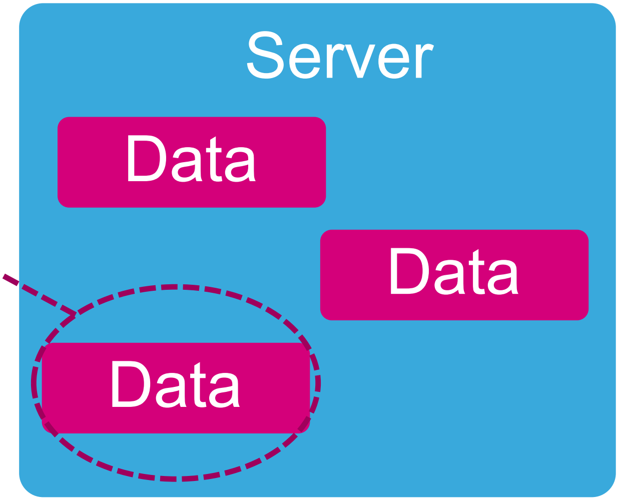


Client talks to the server.
Sends requests, wait for response.
Can confirm indications

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

Attributes

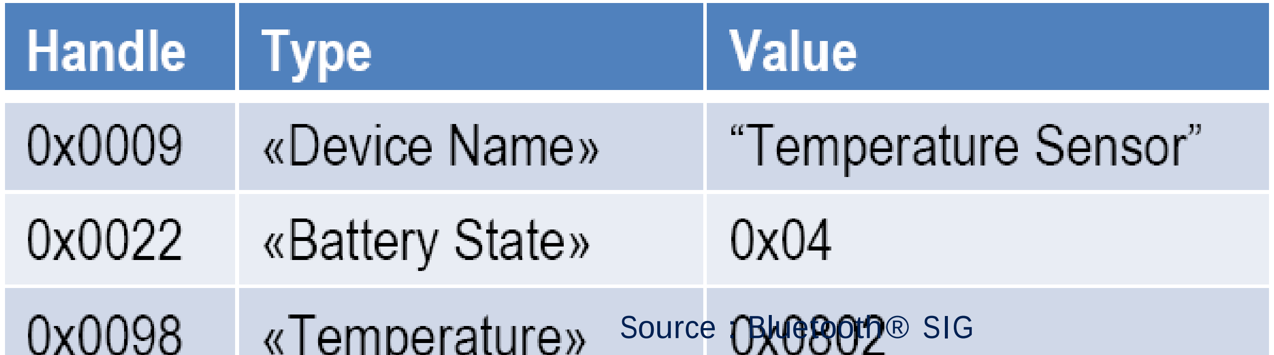
- **Data are organized** and exposed as **attributes**



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

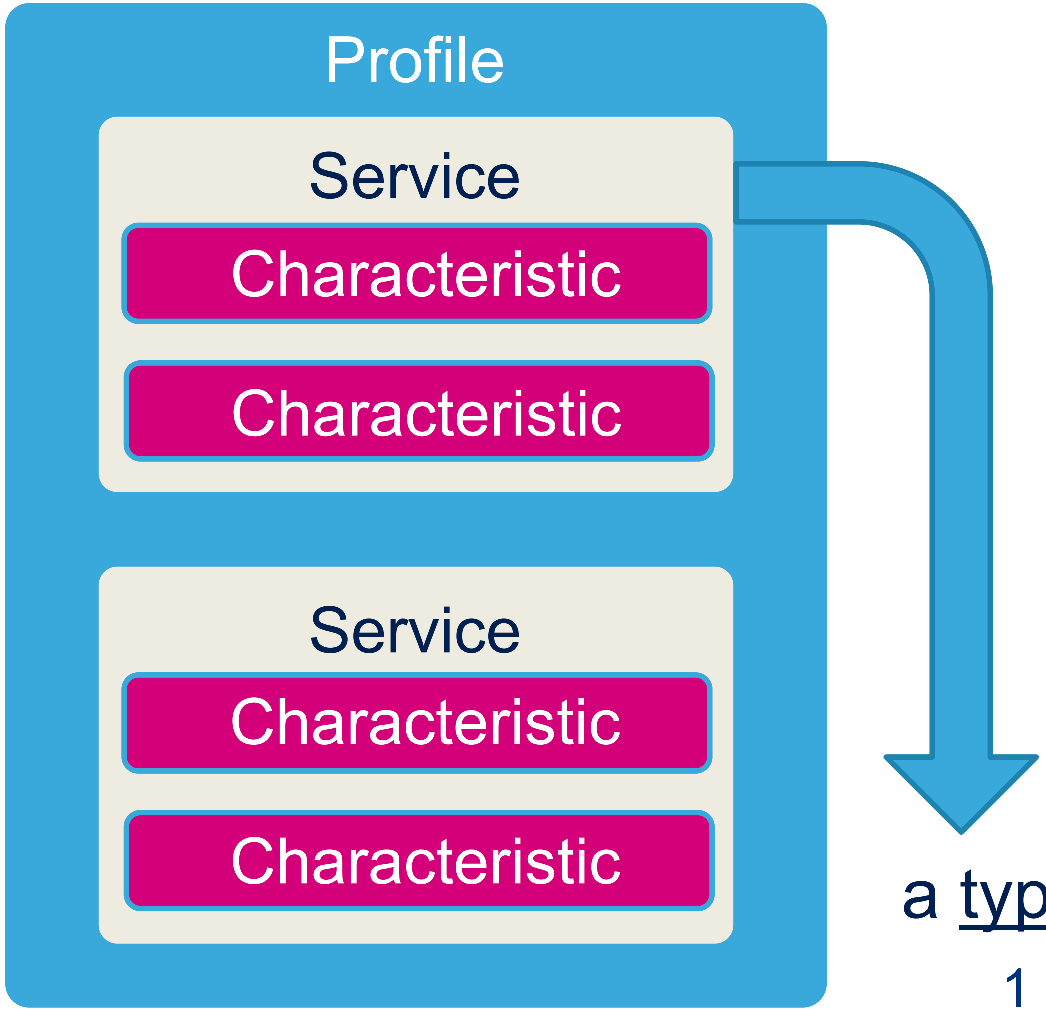


Handle	Type	Value
0x0009	«Device Name»	"Temperature Sensor"
0x0022	«Battery State»	0x04
0x0098	«Temperature»	0x0802

Source : Bluetooth® SIG

life.augmented

GATT Profile



- 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)

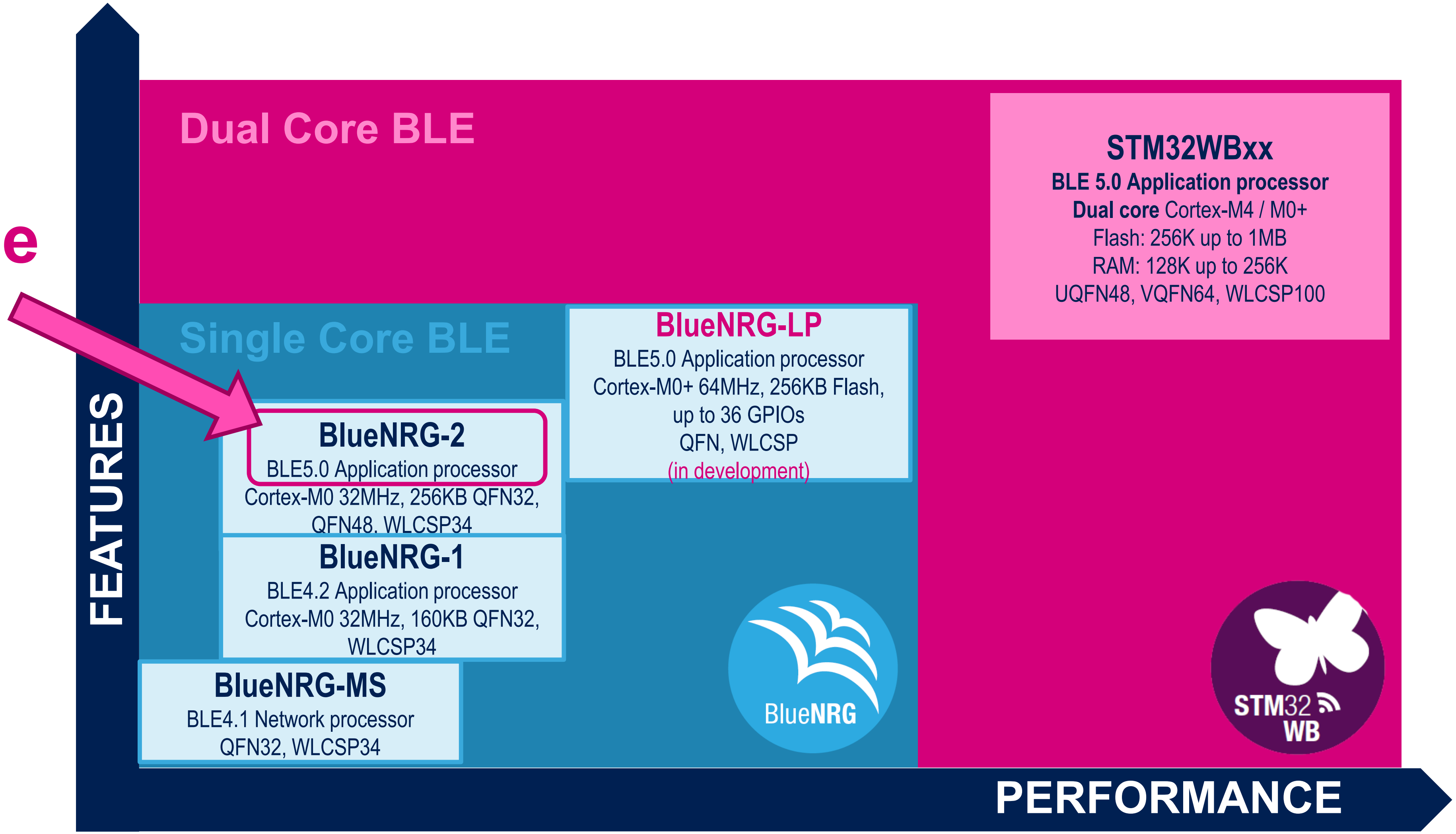
2 characteristics: “Gyro”, “Acc”

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

ST BLE Devices

ST BLE Roadmap

BlueNRG-Tile



Dual Core BLE

STM32WBxx
BLE 5.0 Application processor
Dual core Cortex-M4 / M0+
Flash: 256K up to 1MB
RAM: 128K up to 256K
UQFN48, VQFN64, WLCSP100

Single Core BLE

BlueNRG-LP
BLE5.0 Application processor
Cortex-M0+ 64MHz, 256KB Flash,
up to 36 GPIOs
QFN, WLCSP
(in development)

BlueNRG-2
BLE5.0 Application processor
Cortex-M0 32MHz, 256KB QFN32,
QFN48, WLCSP34

BlueNRG-1
BLE4.2 Application processor
Cortex-M0 32MHz, 160KB QFN32,
WLCSP34

BlueNRG-MS
BLE4.1 Network processor
QFN32, WLCSP34

life.augmented

BlueNRG-2 SoC at a Glance

The lowest power consumption

>3 years lifetime on CR2032(*)
59 µA/MHz
0.9 µA sleep

Processing power on demand

Low-power architecture, Cortex-M0 @ 32 MHz

Flexible memory architecture

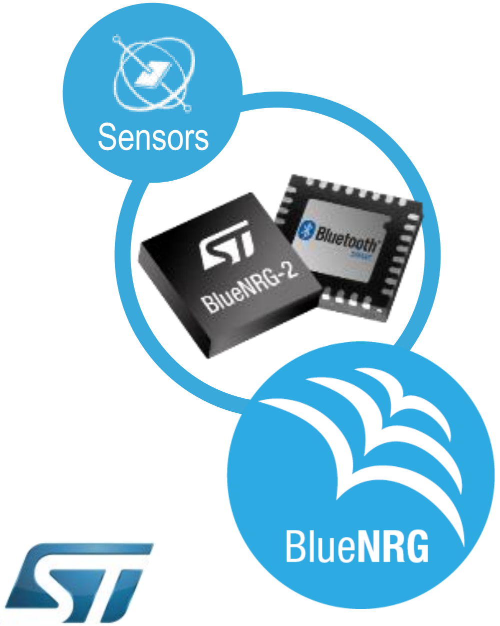
256 KB eFLASH
24 KB ULL SRAM
(with full SRAM data retention)

Optimized BLE Radio stack

70 kB FLASH
8 kB RAM
0.9uA with full RAM retention

Maximum security

ECC-256
AES-128
Factory UID
Secure KEY



Seamless connection with SENSORS

Robust and Reliable BLE Link

Small form factor

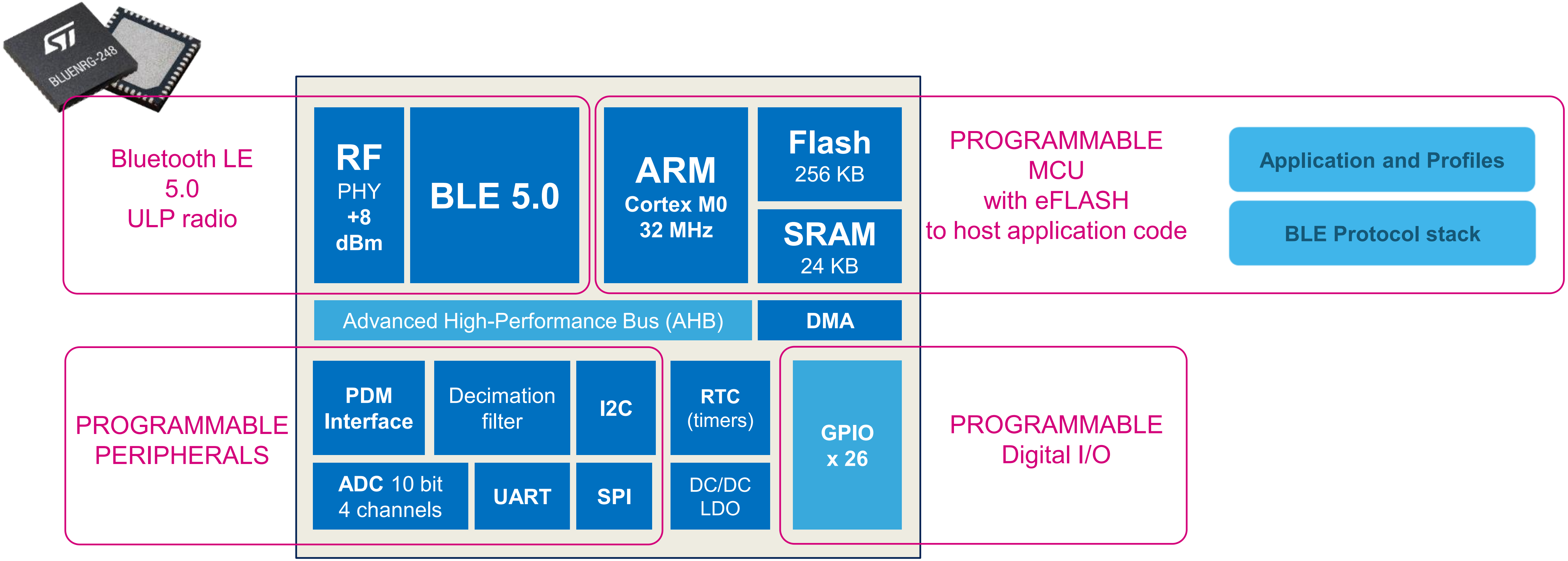
Bluetooth 5.0 certified

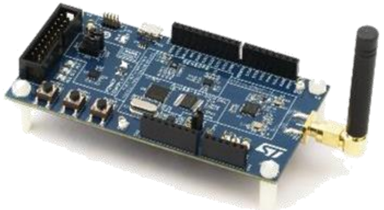
Privacy 1.2 and secure connection 4.2

Faster and more reliable data transfer

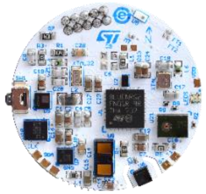
(*) Based on the average current consumption in connection mode (7.059 µA, connection interval 1000 ms)

Bluetooth LE Programmable Processor





STEVAL-IDB008V2

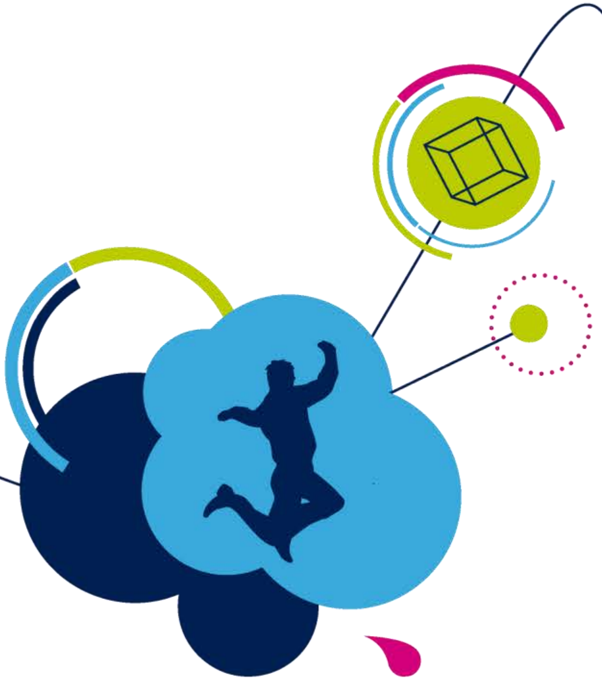


STEVAL-BCN002V1B

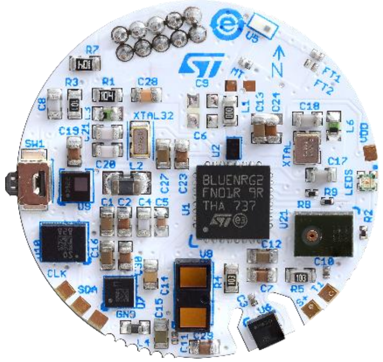
IAR SYSTEMS

KEIL Tools by ARM

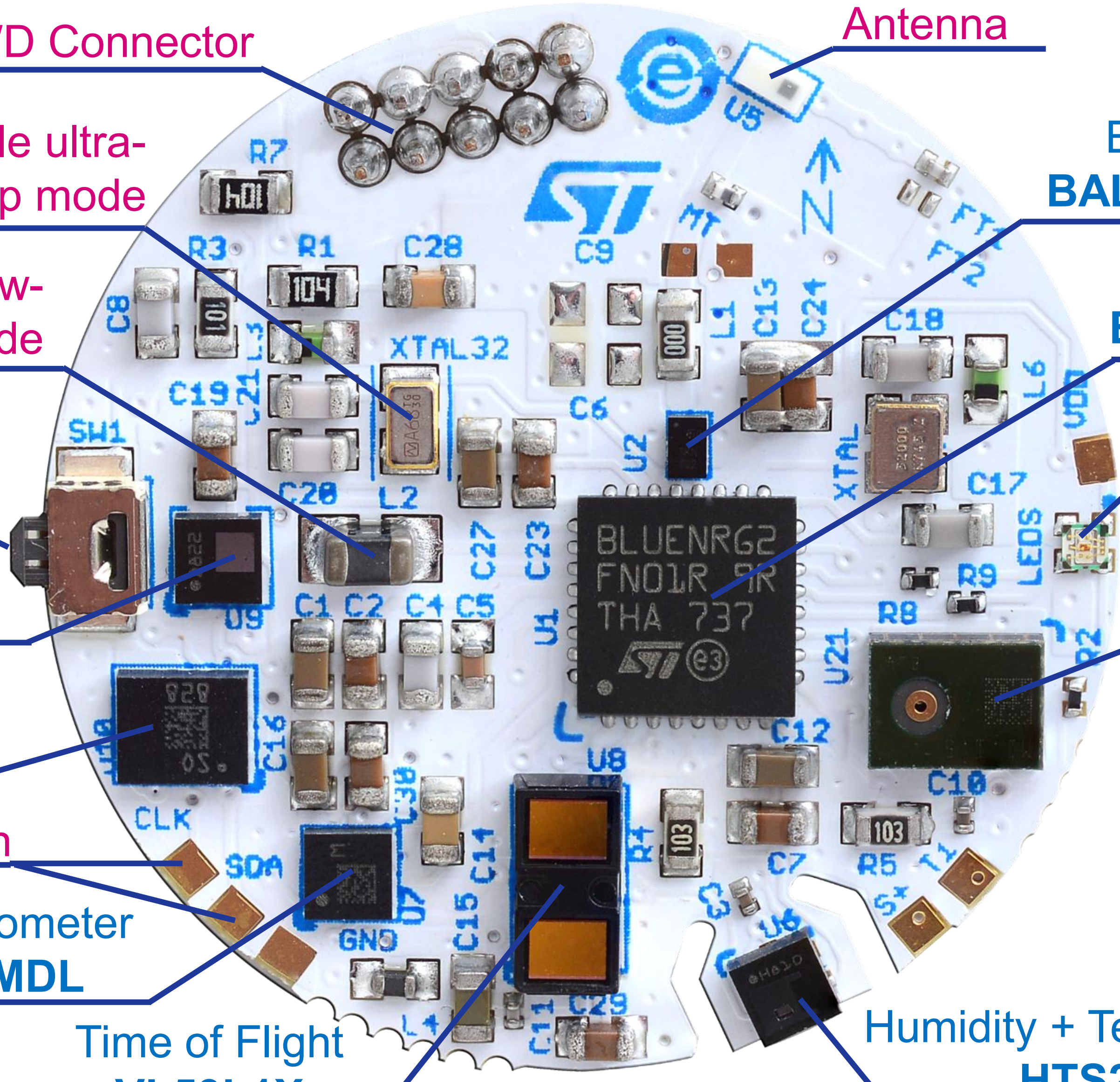
atollic



BlueTile Development Kit



STEVAL-BCN002V1



SWD Connector

32KHz XTAL to enable ultra-low-power BLE sleep mode

L2 bead to enable low-power BLE active mode

User button

Pressure Sensor
LPS22HH

Accelerometer + Gyro
LSM6DSO

I2C for expansion

Magnetometer
LIS2MDL

Time of Flight
VL53L1X

Antenna

Balun
BALF-NRG

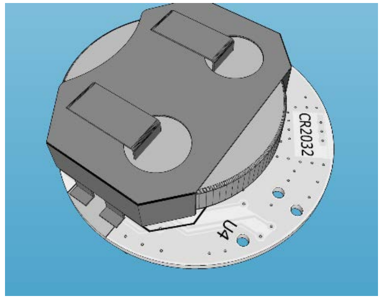
BLE SoC
BlueNRG-232

RGB LED

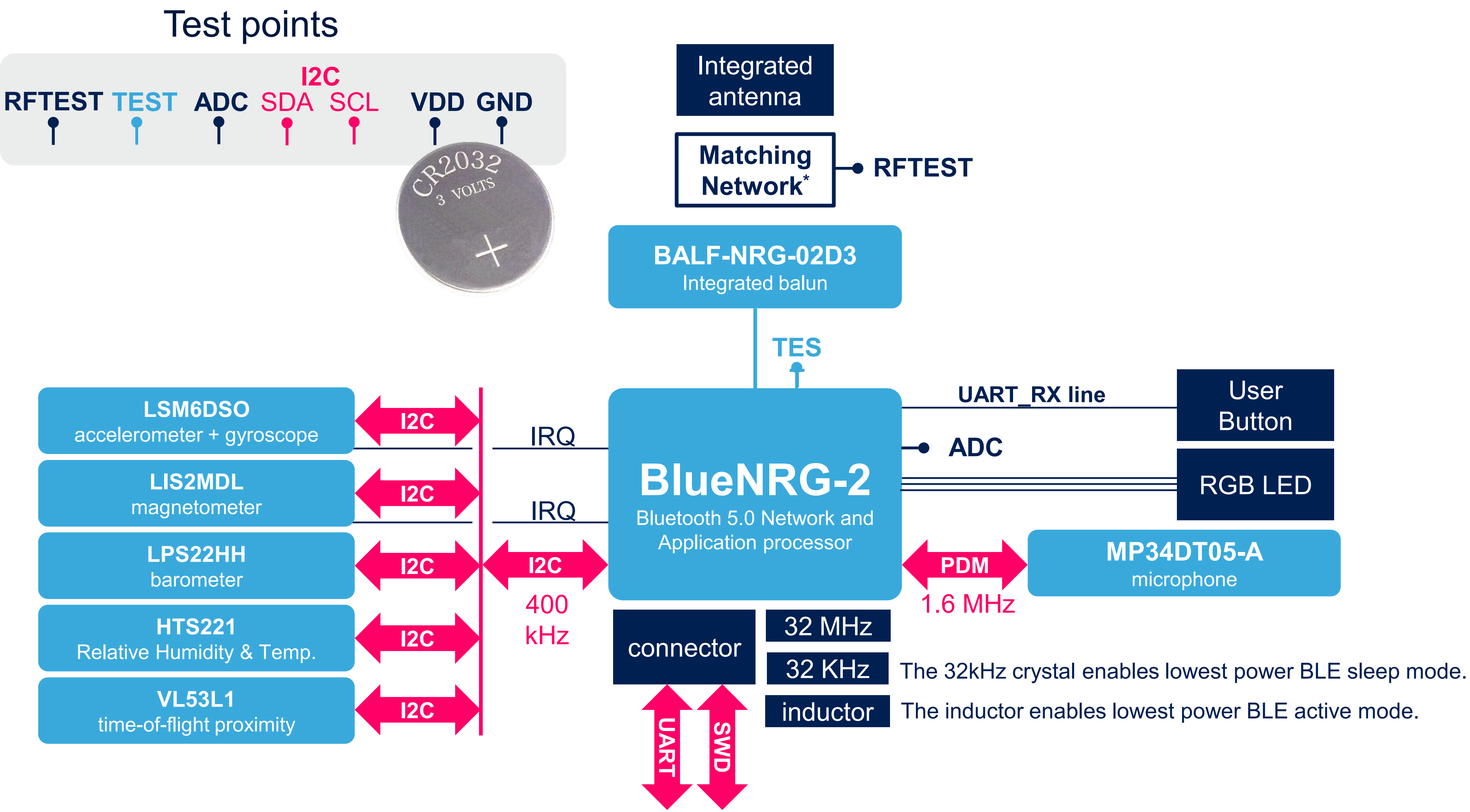
Microphone
MP34DT05-A

Humidity + Temperature
HTS221

Bottom view



STEVAL-BCN002V1 Block Diagram



Test points

I2C

RFTEST TEST ADC SDA SCL VDD GND

CR2032 3 VOLTS +

Integrated antenna

Matching Network*

RFTEST

BALF-NRG-02D3
Integrated balun

TES

LSM6DSO
accelerometer + gyroscope

LIS2MDL
magnetometer

LPS22HH
barometer

HTS221
Relative Humidity & Temp.

VL53L1
time-of-flight proximity

I2C

I2C

I2C

I2C

I2C

IRQ

IRQ

I2C
400 kHz

BlueNRG-2
Bluetooth 5.0 Network and Application processor

UART_RX line

User Button

ADC

RGB LED

PDM
1.6 MHz

MP34DT05-A
microphone

connector

UART

SWD

32 MHz

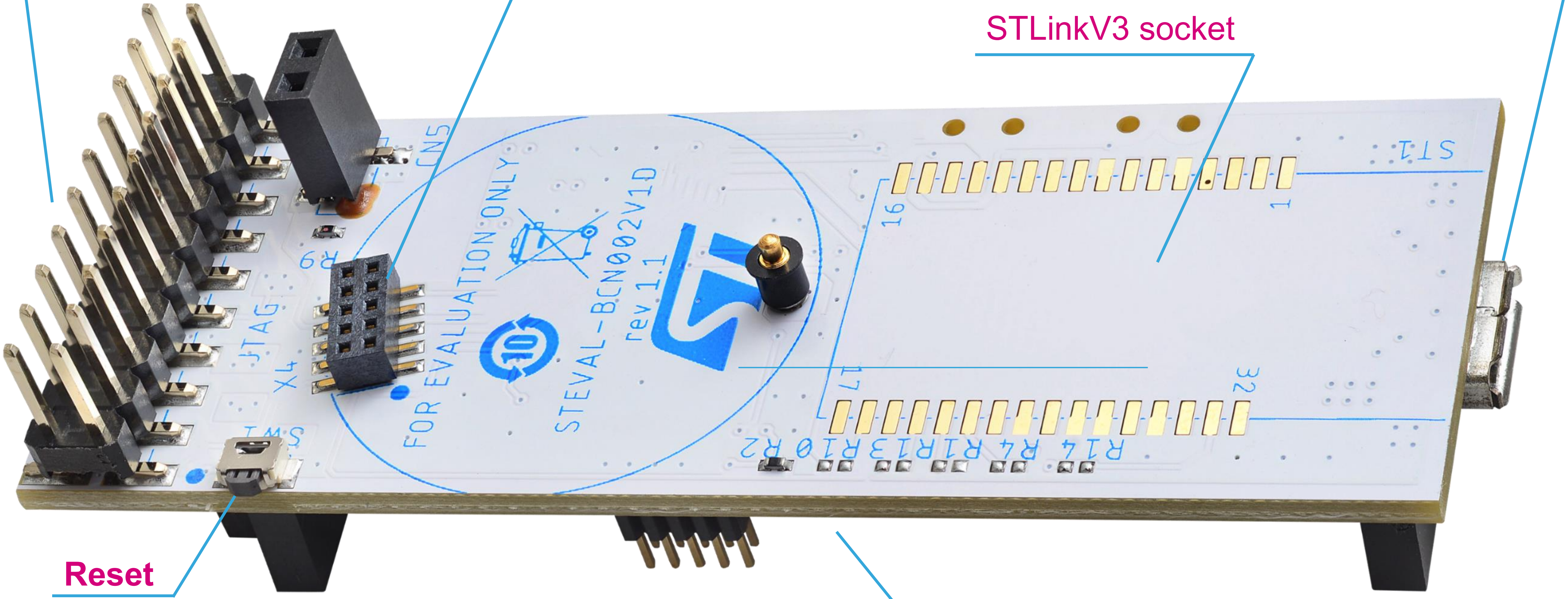
32 KHz

The 32kHz crystal enables lowest power BLE sleep mode.

inductor

The inductor enables lowest power BLE active mode.

STEVAL-BCN002V1D



20-pin JTAG connector

STEVAL-BCN002V1 Socket

Micro-USB connector (bottom)

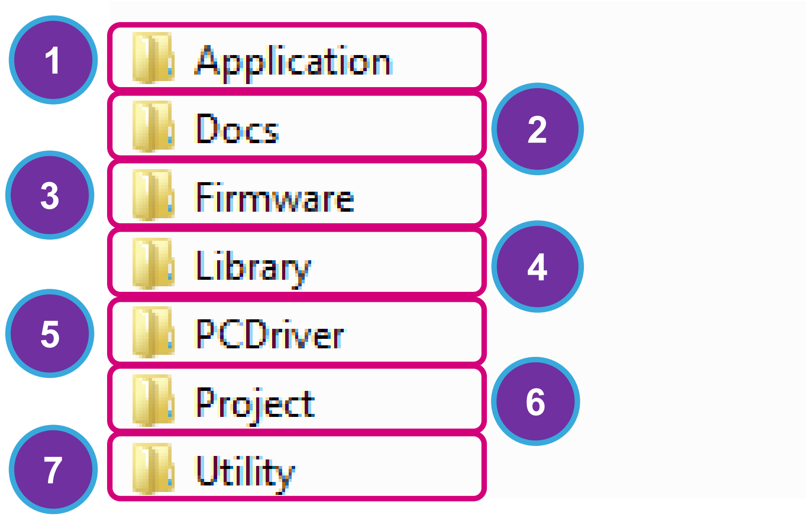
STLinkV3 socket

Reset

STM32L151 (bottom)

BlueTile SDK Overview

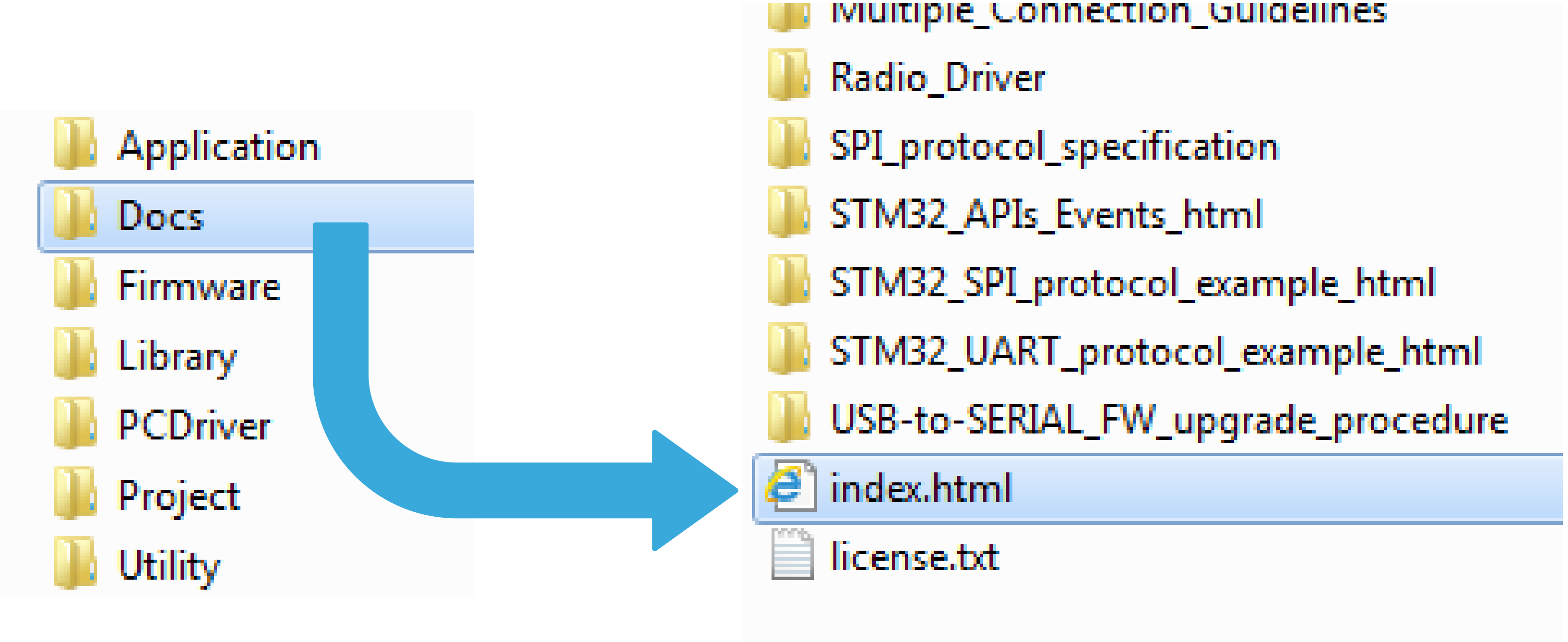
https://www.st.com/content/st_com/en/products/embedded-software/evaluation-tool-software/stsw-bluetile-dk.html



1. PC Applications
2. Documentation
3. Pre-built FW images
4. Low level drivers and BLE stack library
5. Virtual COM port drivers
6. Reference examples in source code
7. Utility section: IAR BlueNRG-2 Flasher

Documents

- Open the **Docs** folder



Application
Docs
Firmware
Library
PCDriver
Project
Utility

Multiple_Connection_Guidelines
Radio_Driver
SPI_protocol_specification
STM32_APIs_Events_html
STM32_SPI_protocol_example_html
STM32_UART_protocol_example_html
USB-to-SERIAL_FW_upgrade_procedure
index.html
license.txt

- **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

- **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”

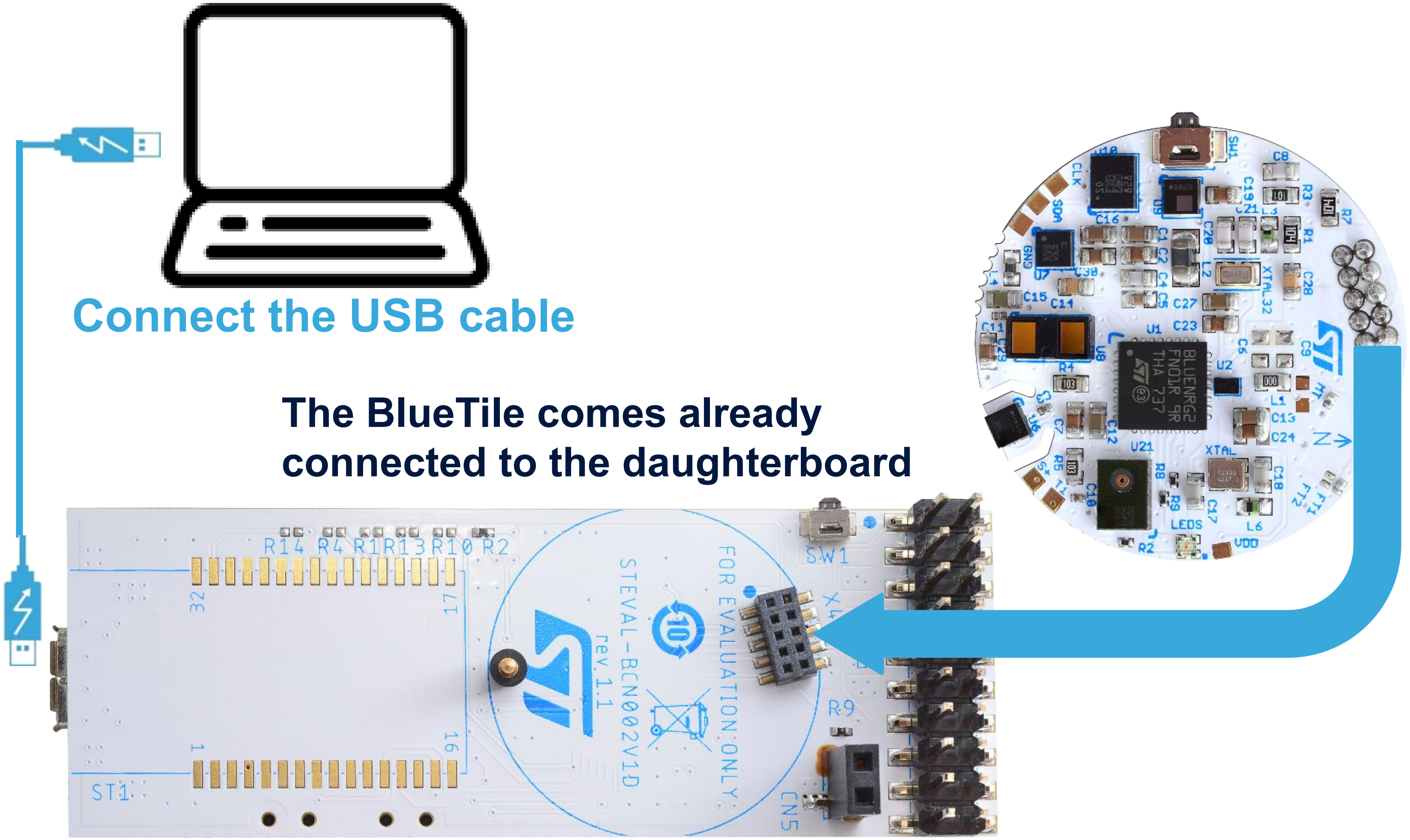
life.augmented

“Serial Terminal Test”

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

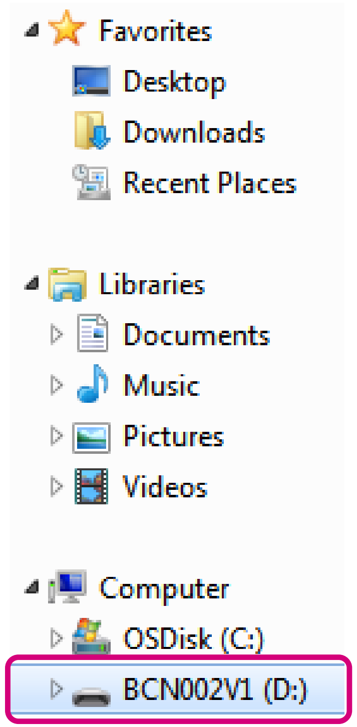
Connect Your STEVAL-BCN002V1B

to the PC Using USB



Connect the USB cable

The BlueTile comes already connected to the daughterboard



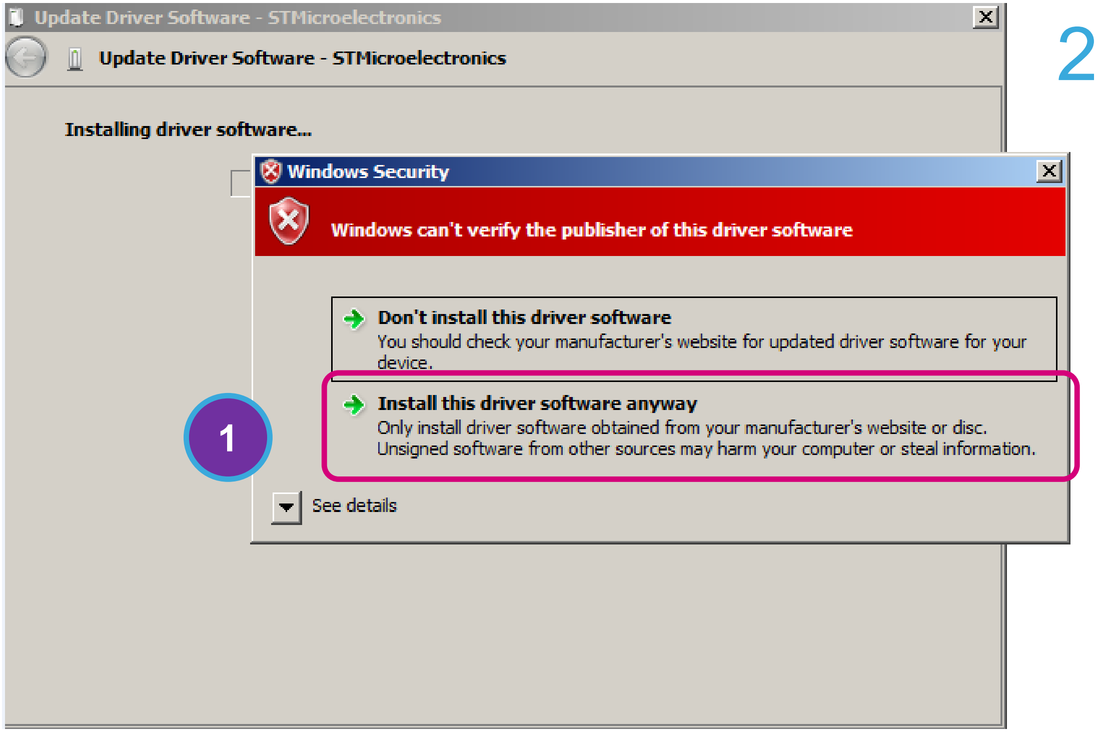
- Favorites
  - Desktop
  - Downloads
  - Recent Places
- Libraries
  - Documents
  - Music
  - Pictures
  - Videos
- Computer
  - OSDisk (C:)
  - BCN002V1 (D:)

On the PC file system a BCN002V1 drive will appear

Windows7: Allow the Driver Installation

1. Click on **Install driver software anyway**

2. Installation starts



Update Driver Software - STMicroelectronics

Update Driver Software - STMicroelectronics

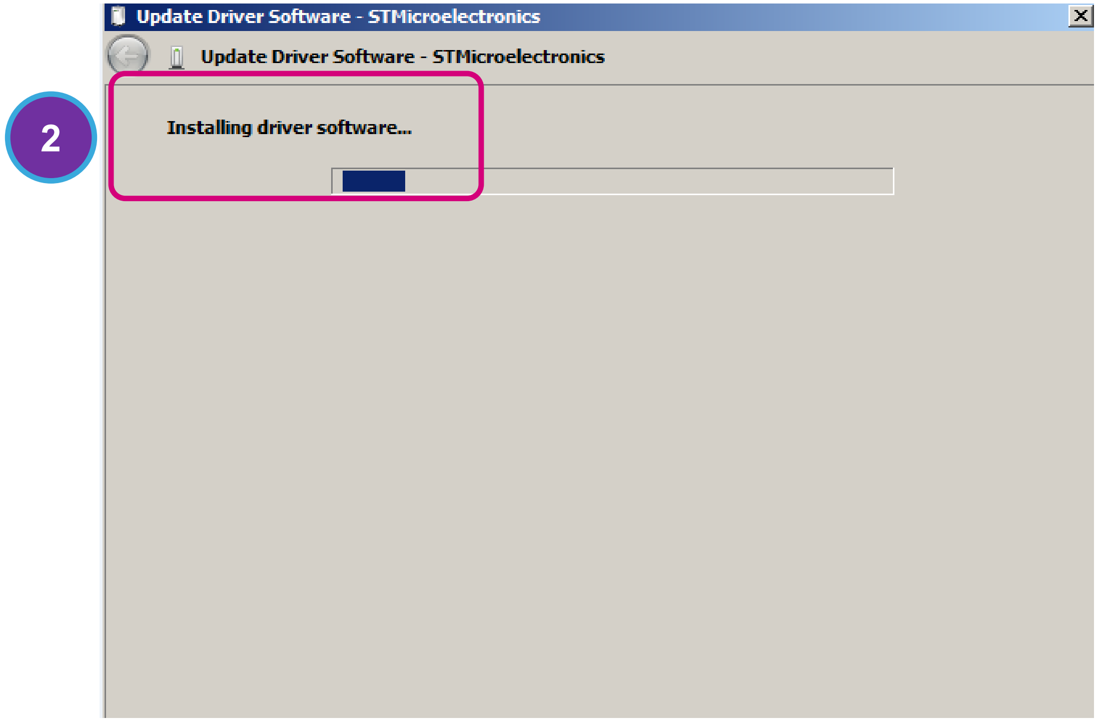
Installing driver software...

Windows Security

Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.



Update Driver Software - STMicroelectronics

Update Driver Software - STMicroelectronics

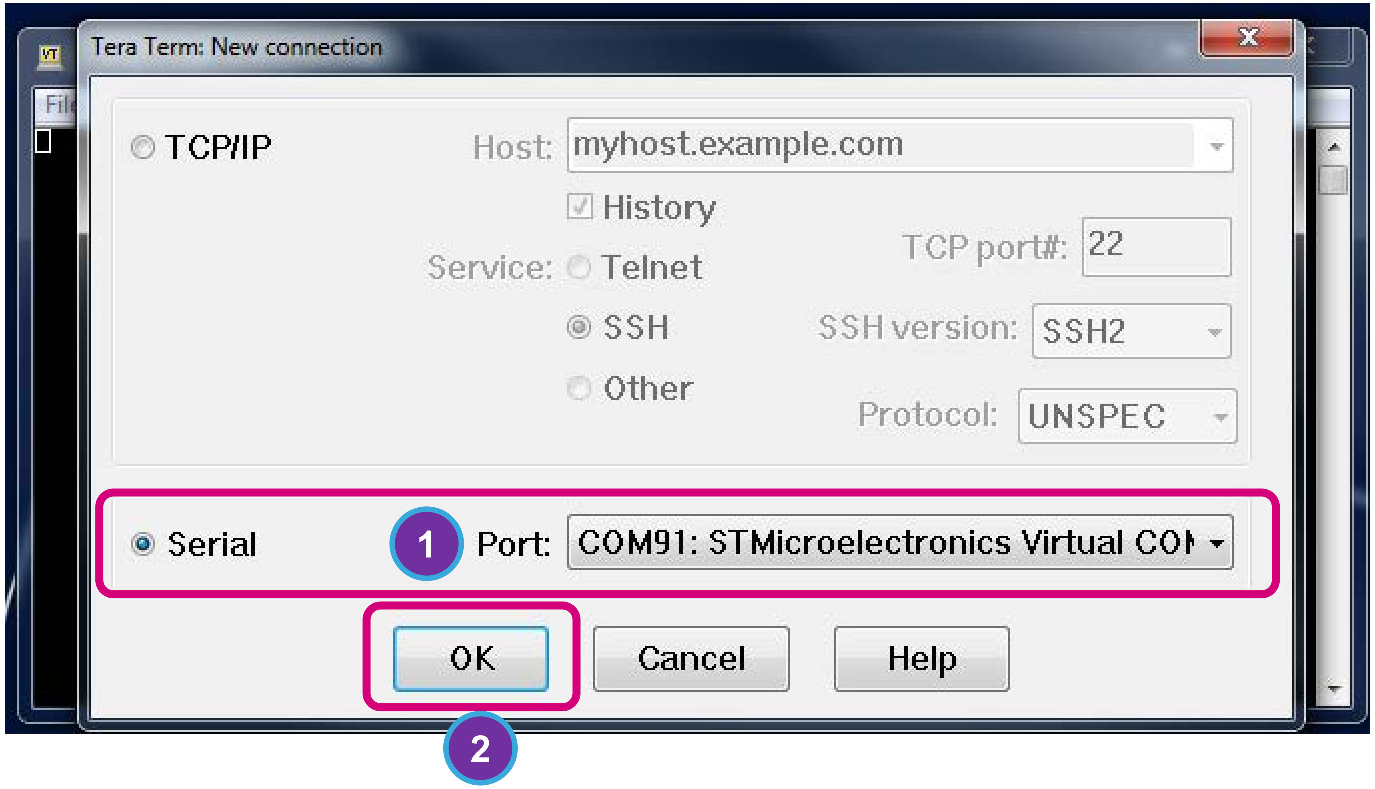
Installing driver software...

Virtual COM Port Driver

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



Tera Term: New connection

○ TCP/IP Host: myhost.example.com

☑ History

Service: ○ Telnet TCP port#: 22

● SSH SSH version: SSH2

○ Other Protocol: UNSPEC

● Serial Port: COM91: STMicroelectronics Virtual COI

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

1. In **Setup** -> **Serial port…**
   Set the following:

   Baud rate : 115200

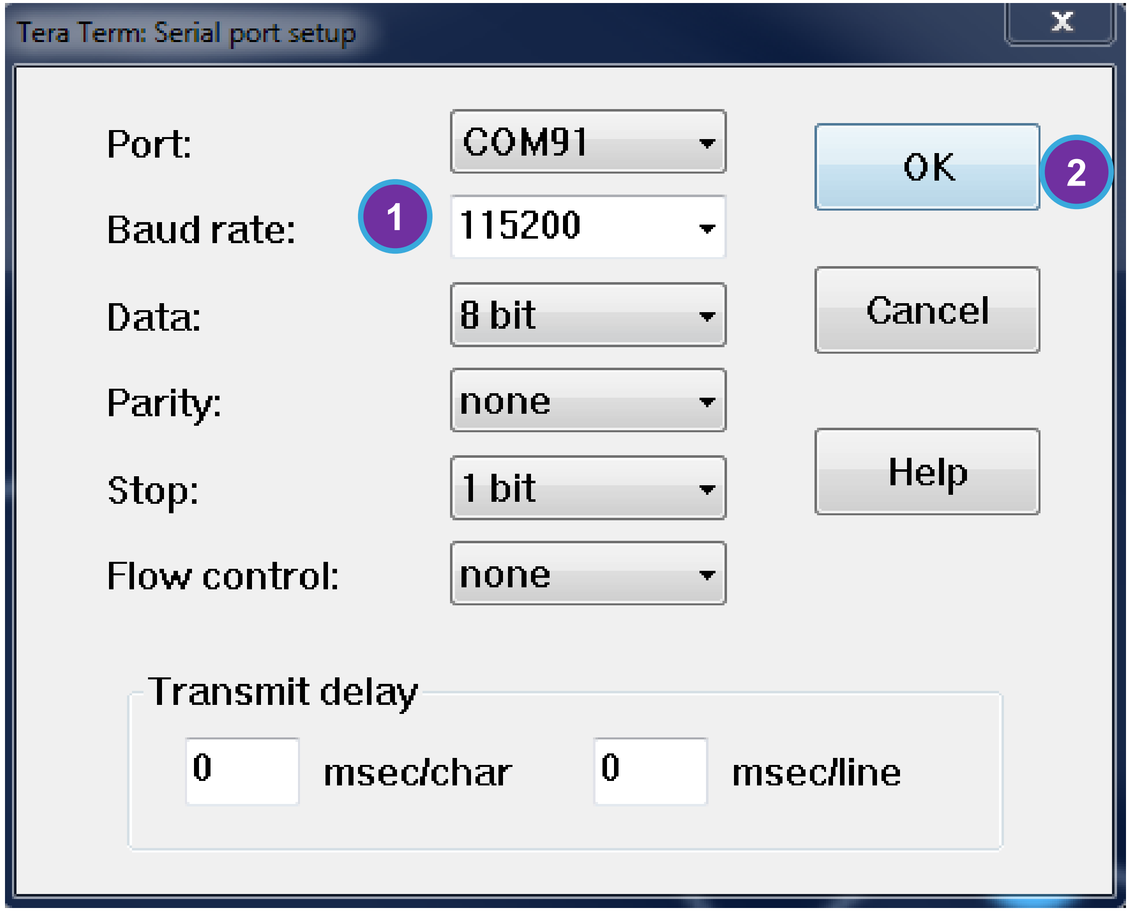
   Data : 8 bit

   Parity : none

   Stop : 1 bit

   Flow control : none

2. **Click OK**


Tera Term: Serial port setup

Port: COM91

Baud rate: 115200

Data: 8 bit

Parity: none

Stop: 1 bit

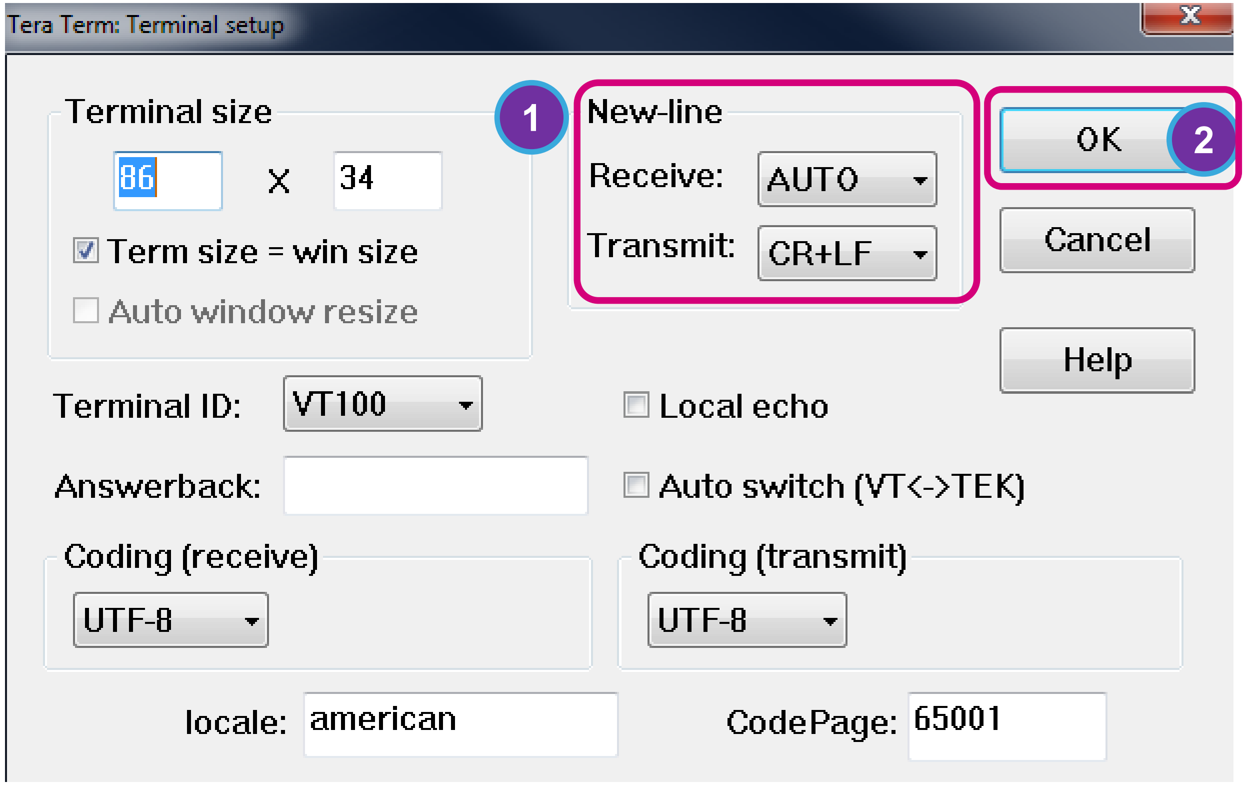
Flow control: none

Transmit delay

0 msec/char 0 msec/line

Tera Term Configuration 3/4

1. In **Setup** -> **Terminal…** set **New-line Receive : AUTO**
2. **Click OK**



Tera Term: Terminal setup

Terminal size
86 X 34
☑ Term size = win size
□ Auto window resize

New-line
Receive: AUTO
Transmit: CR+LF

Terminal ID: VT100

Answerback:

□ Local echo
□ Auto switch (VT<->TEK)

Coding (receive)
UTF-8

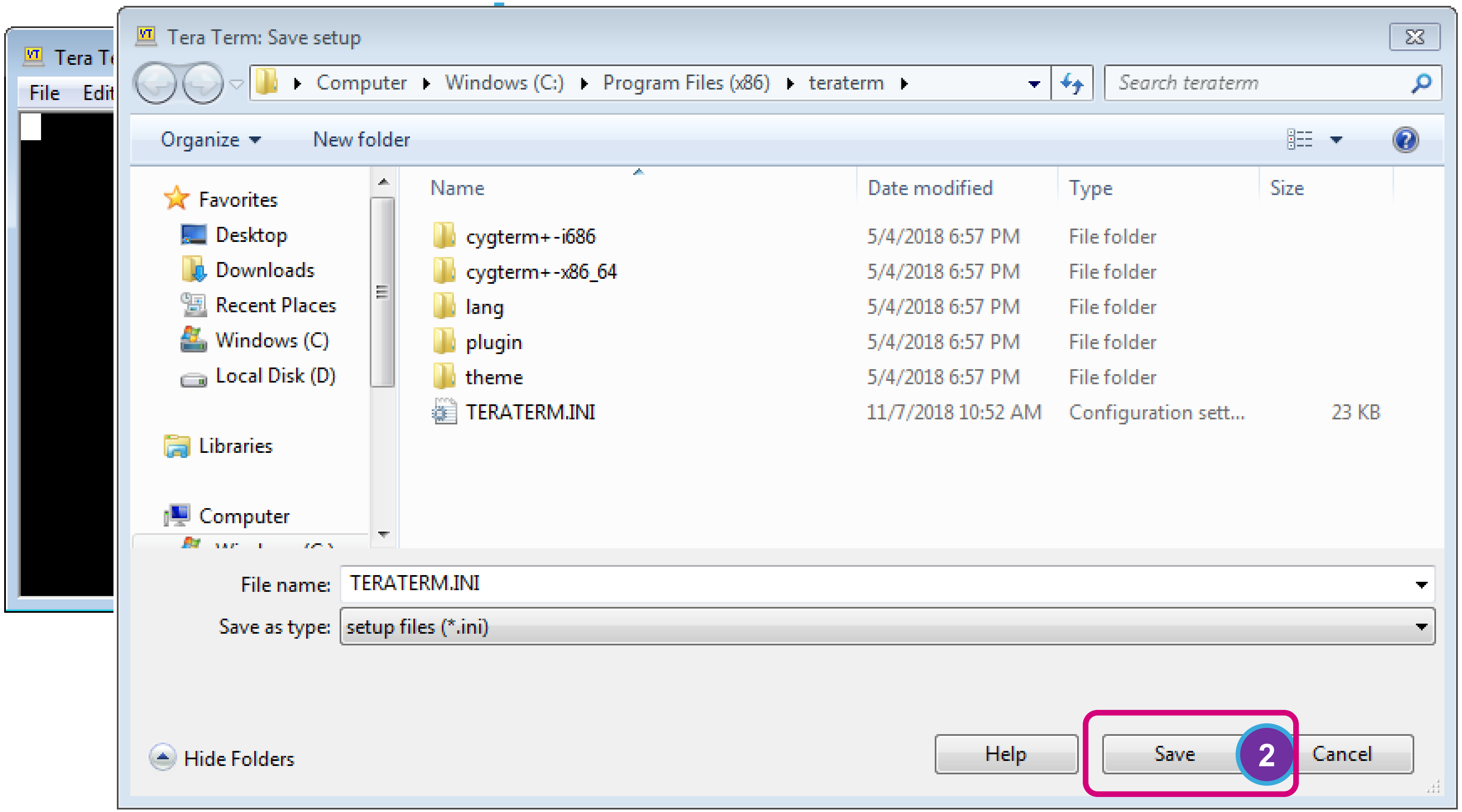
Coding (transmit)
UTF-8

locale: american

CodePage: 65001

Tera Term Configuration 4/4

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



Tera Term: Save setup

Computer ▸ Windows (C:) ▸ Program Files (x86) ▸ teraterm ▸

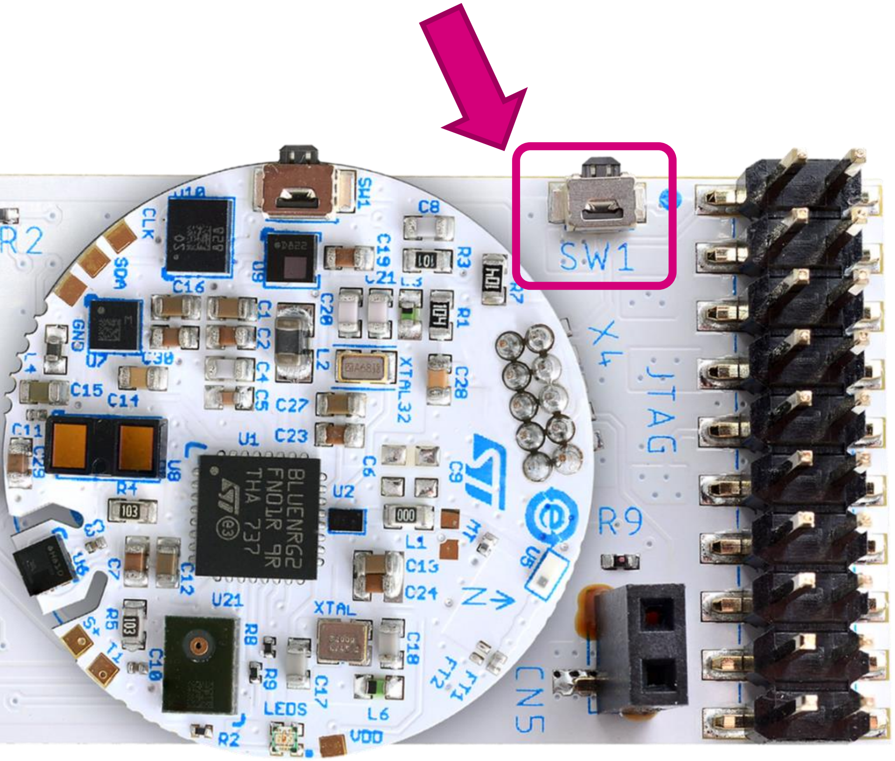
Name	Date modified	Type	Size
cygterm+-i686	5/4/2018 6:57 PM	File folder	
cygterm+-x86_64	5/4/2018 6:57 PM	File folder	
lang	5/4/2018 6:57 PM	File folder	
plugin	5/4/2018 6:57 PM	File folder	
theme	5/4/2018 6:57 PM	File folder	
TERATERM.INI	11/7/2018 10:52 AM	Configuration sett...	23 KB

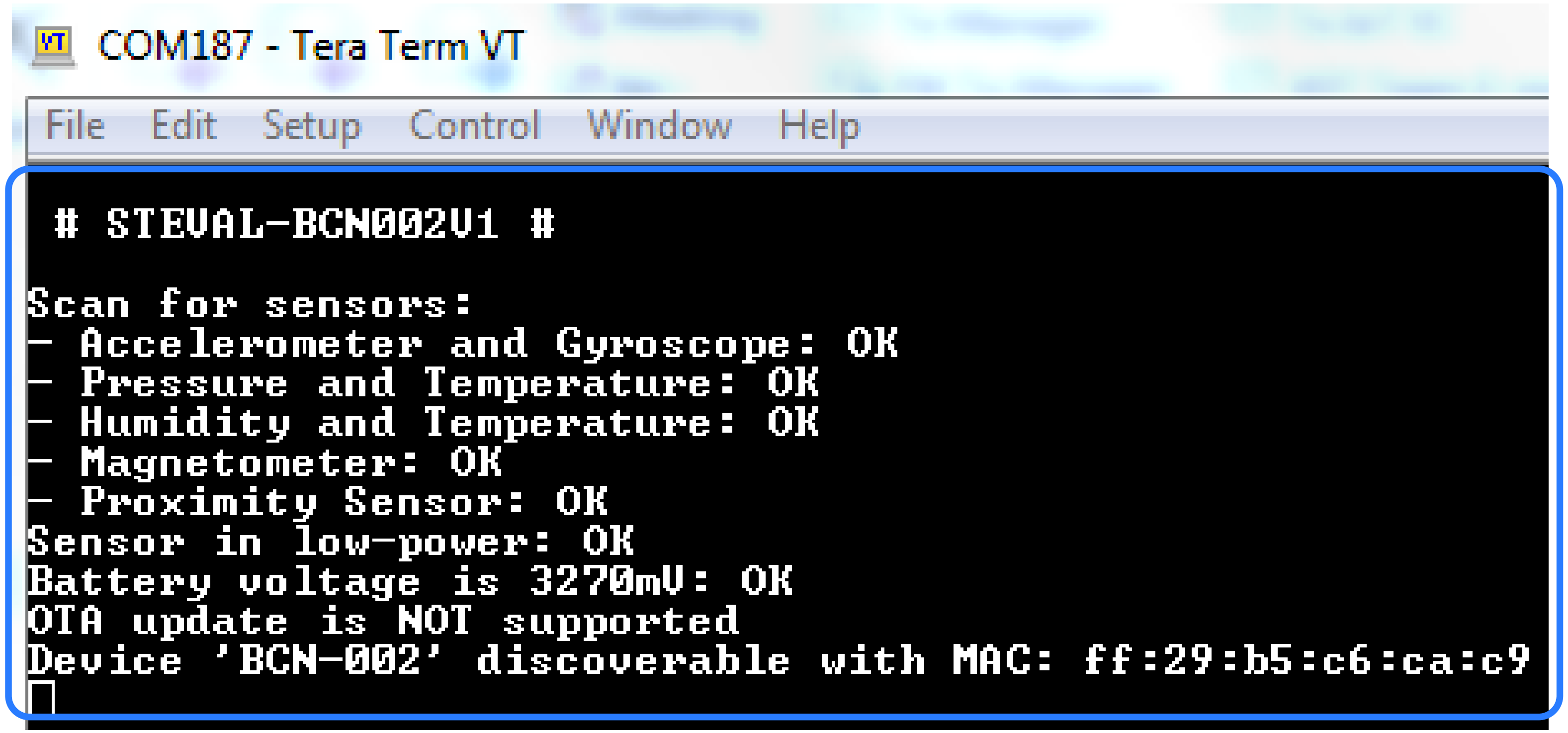
File name: TERATERM.INI

Save as type: setup files (*.ini)

…Done!

Push SW1 button on the daughterboard





```
 # STEVAL-BCN002V1 #

Scan for sensors:
- Accelerometer and Gyroscope: OK
- Pressure and Temperature: OK
- Humidity and Temperature: OK
- Magnetometer: OK
- Proximity Sensor: OK
Sensor in low-power: OK
Battery voltage is 3270mV: OK
OTA update is NOT supported
Device 'BCN-002' discoverable with MAC: ff:29:b5:c6:ca:c9
```

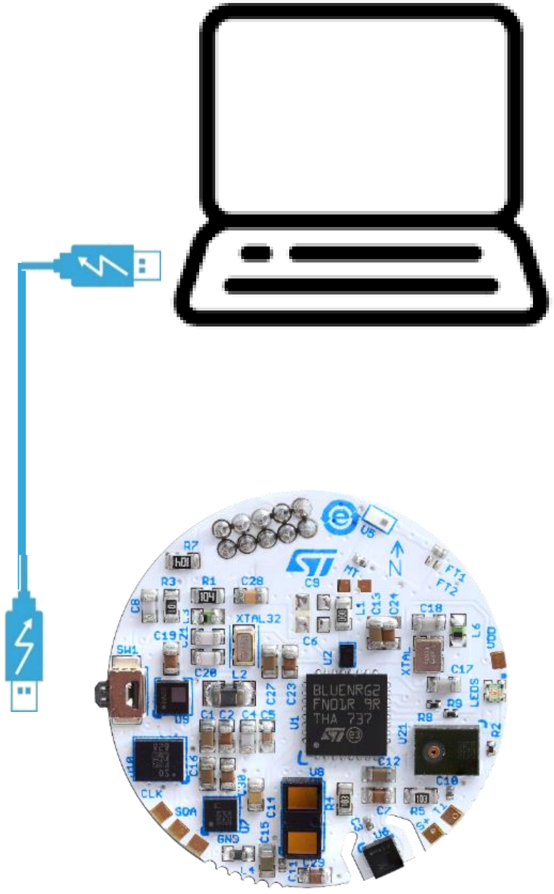
- If you see the output above, **Tera Term** is now properly configured

Lab 2

Connect to the ST BLE Sensor App

life.augmented

Customize YOUR STEVAL-BCN002V1



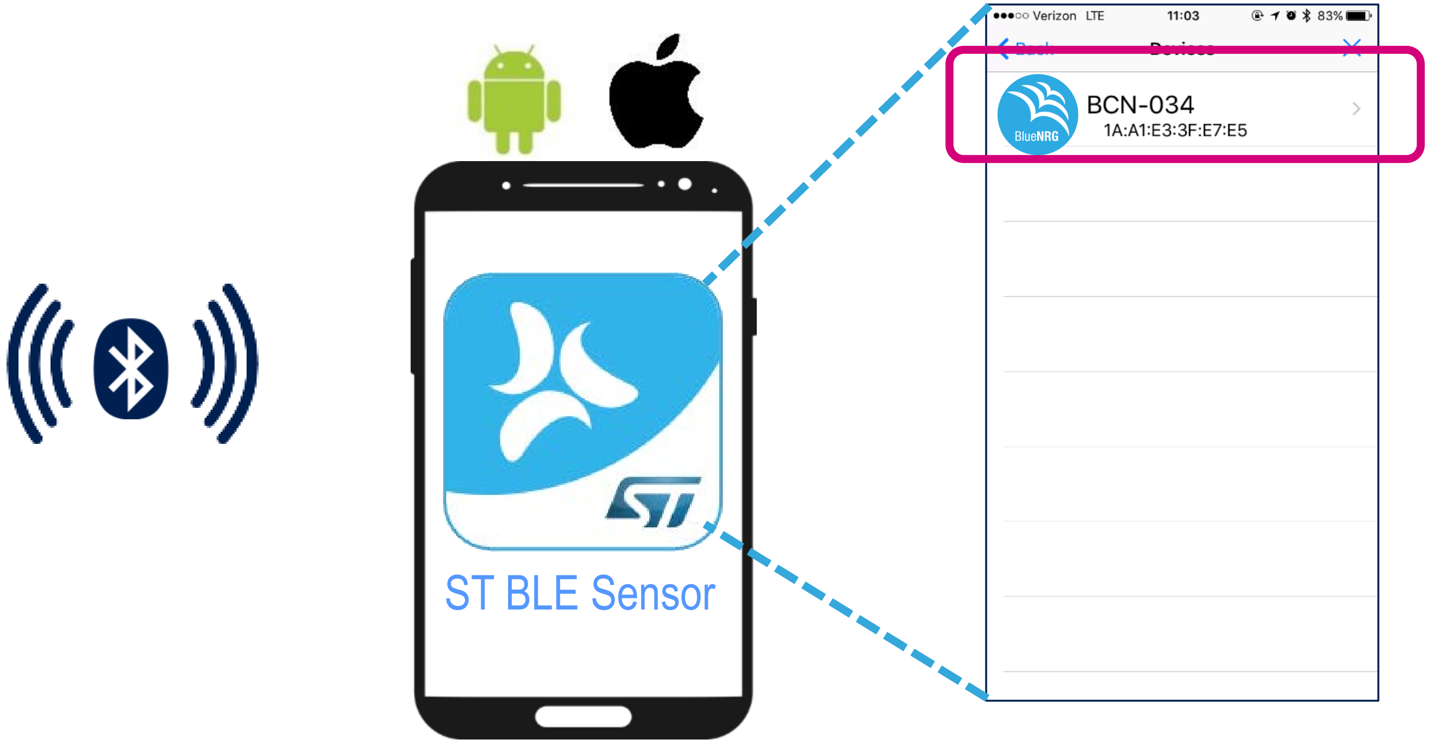
KEIL™ Tools by ARM

µVision®5

Integrated Development Environment

KEIL is a trademark and µVision is a registered trademark of ARM Ltd. All rights reserved. This product is protected by US and international laws.

+ STEVAL-BCN002V1 DK Sensor Demo ref FW



●●●○○ Verizon LTE 11:03 83%

Back Devices

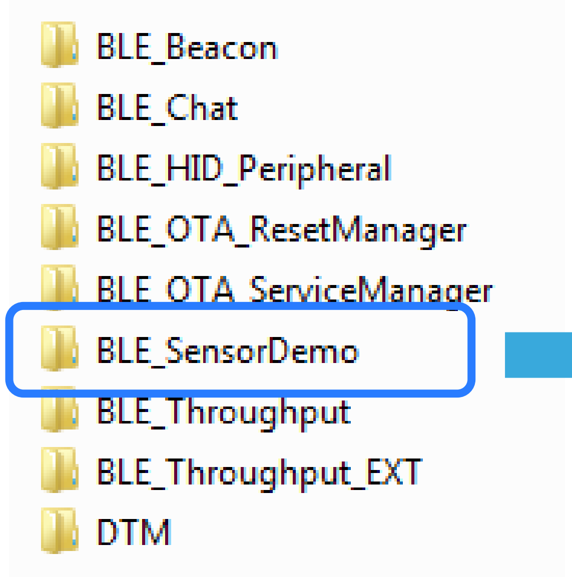
BCN-034

1A:A1:E3:3F:E7:E5

Client

BLE_SensorDemo Application

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



- BLE_Beacon
- BLE_Chat
- BLE_HID_Peripheral
- BLE_OTA_ResetManager
- BLE_OTA_ServiceManager
- BLE_SensorDemo
- BLE_Throughput
- BLE_Throughput_EXT
- DTM

The BLE_SensorDemo embedded application is the main reference

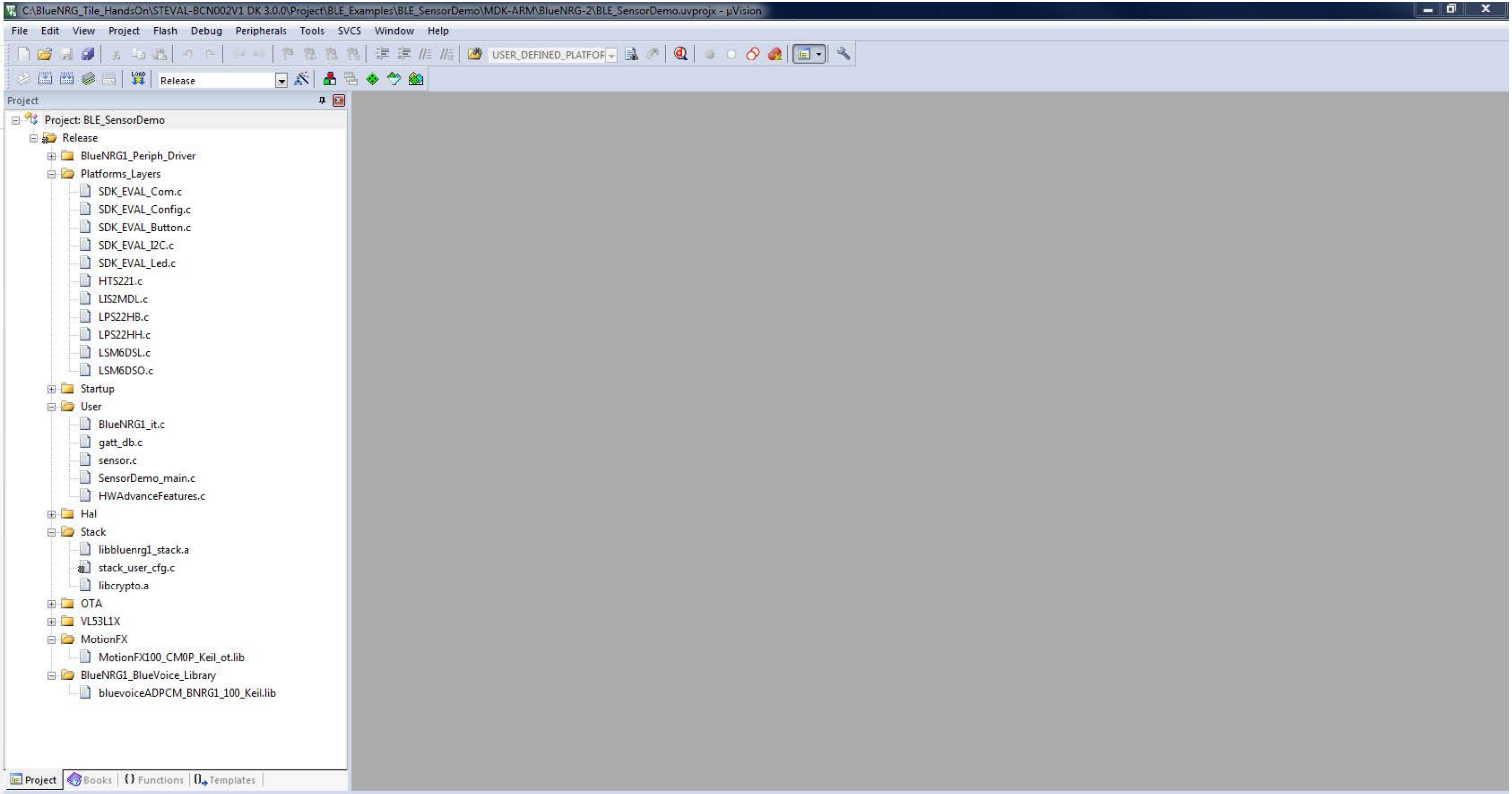
Integrated Development Environment

Arm Keil MDK

- **MDK**: **M**icrocontroller **D**evelopment **K**it
  - 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!

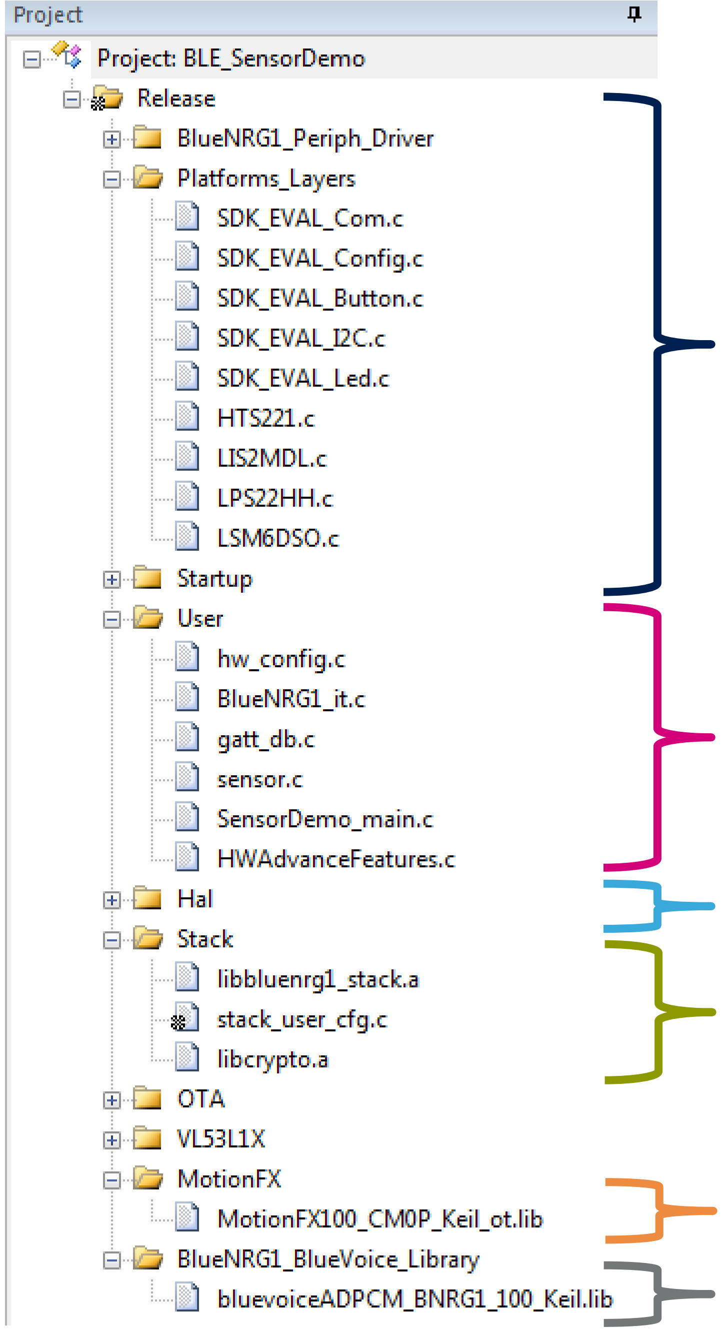
BLE_SensorDemo Application



C:\BlueNRG_Tile_HandsOn\STEVAL-BCN002V1 DK 3.0.0\Project\BLE_Examples\BLE_SensorDemo\MDK-ARM\BlueNRG-2\BLE_SensorDemo.uvprojx - µVision

- Project: BLE_SensorDemo
  - Release
    - BlueNRG1_Periph_Driver
    - Platforms_Layers
      - SDK_EVAL_Com.c
      - SDK_EVAL_Config.c
      - SDK_EVAL_Button.c
      - SDK_EVAL_I2C.c
      - SDK_EVAL_Led.c
      - HTS221.c
      - LIS2MDL.c
      - LPS22HB.c
      - LPS22HH.c
      - LSM6DSL.c
      - LSM6DSO.c
    - Startup
    - User
      - BlueNRG1_it.c
      - gatt_db.c
      - sensor.c
      - SensorDemo_main.c
      - HWAdvanceFeatures.c
    - Hal
    - Stack
      - libbluenrg1_stack.a
      - stack_user_cfg.c
      - libcrypto.a
    - OTA
    - VL53L1X
    - MotionFX
      - MotionFX100_CM0P_Keil_ot.lib
    - BlueNRG1_BlueVoice_Library
      - bluevoiceADPCM_BNRG1_100_Keil.lib

Application Structure



```
Project
⊟ Project: BLE_SensorDemo
   ⊟ Release
      ⊞ BlueNRG1_Periph_Driver
      ⊟ Platforms_Layers
            SDK_EVAL_Com.c
            SDK_EVAL_Config.c
            SDK_EVAL_Button.c
            SDK_EVAL_I2C.c
            SDK_EVAL_Led.c
            HTS221.c
            LIS2MDL.c
            LPS22HH.c
            LSM6DSO.c
      ⊞ Startup
      ⊟ User
            hw_config.c
            BlueNRG1_it.c
            gatt_db.c
            sensor.c
            SensorDemo_main.c
            HWAdvanceFeatures.c
      ⊞ Hal
      ⊟ Stack
            libbluenrg1_stack.a
            stack_user_cfg.c
            libcrypto.a
      ⊞ OTA
      ⊞ VL53L1X
      ⊟ MotionFX
            MotionFX100_CM0P_Keil_ot.lib
      ⊟ BlueNRG1_BlueVoice_Library
            bluevoiceADPCM_BNRG1_100_Keil.lib
```

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

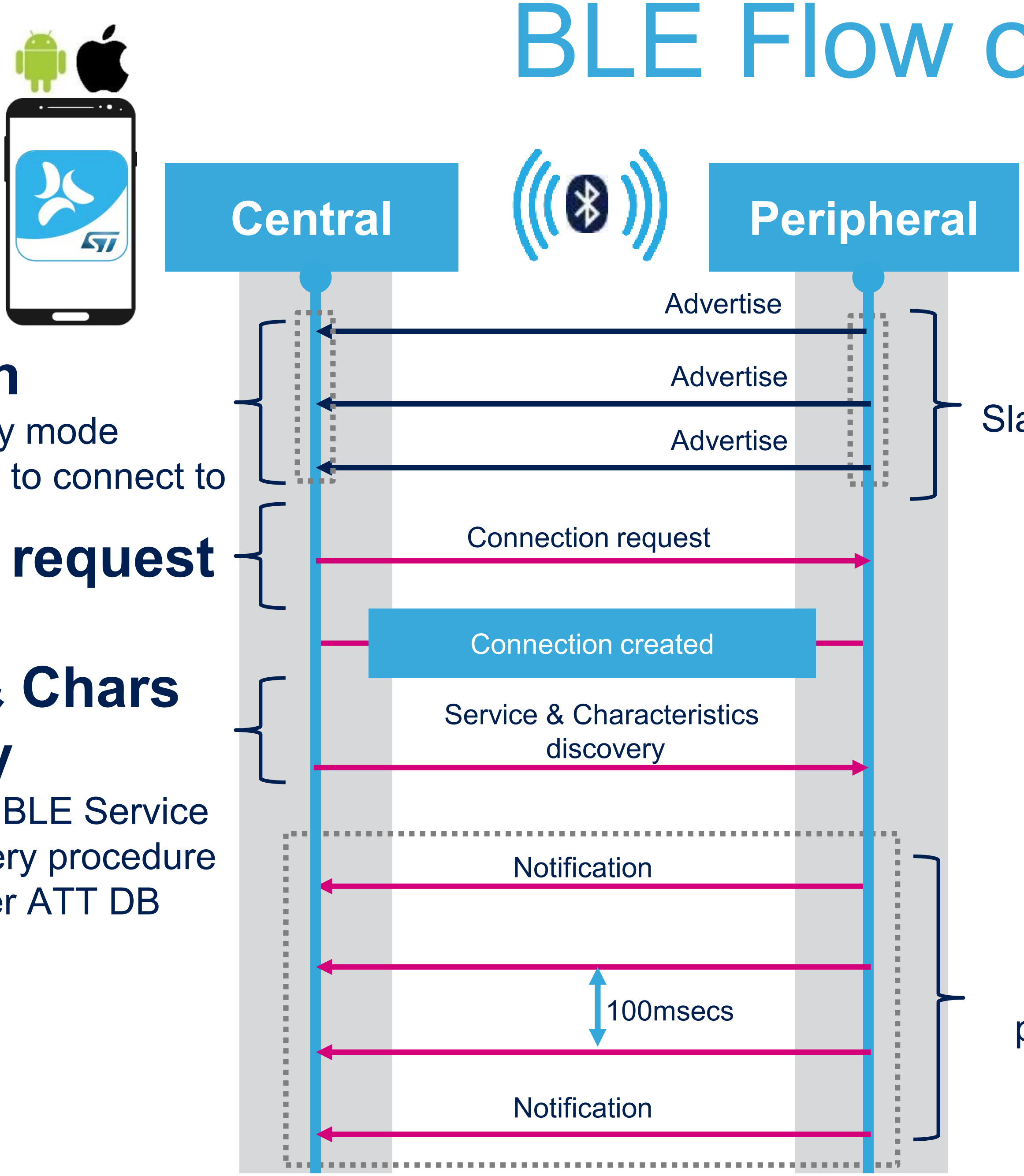
A Look at the Main Application

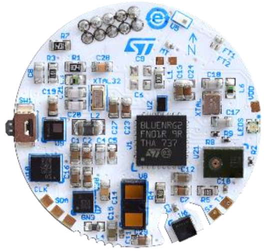
```
int main(void) {
```

`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





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**

Step 4: **Services & Chars discovery**

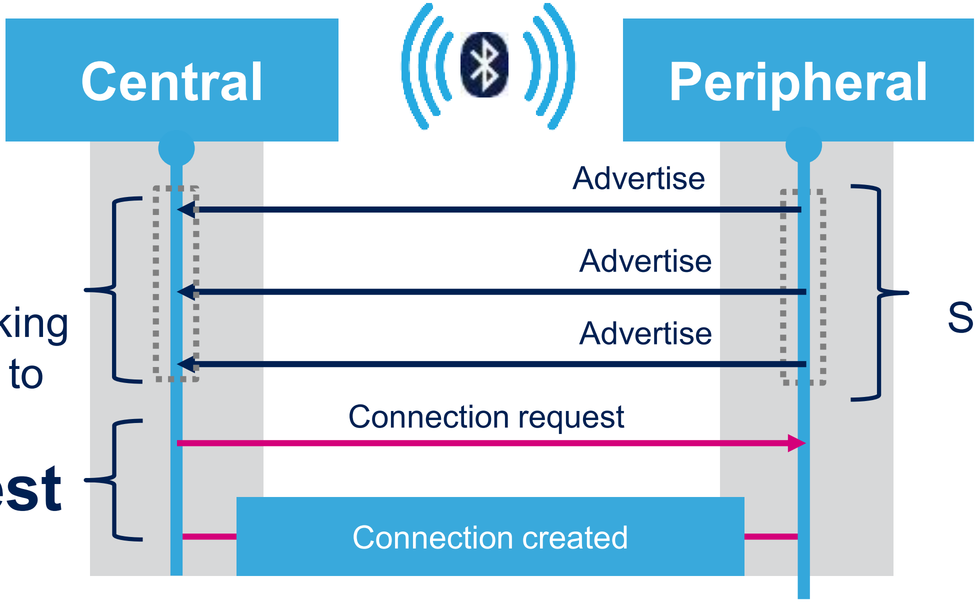
Master (Client) starts the BLE Service and Characteristics discovery procedure to understand the Server ATT DB

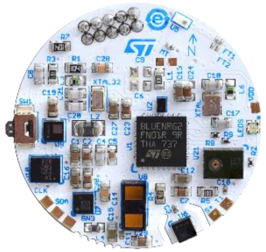
Step 5: **Data flow**

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

life.augmented

Advertising and Scanning





Step 2: **Scan**

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

Step 3: **Connection request**

Step 1: **Advertising**

Slave is in Peripheral mode and sends ADV_IND PDU

Master:

needs an app for discovering the slave device in advertising

Off-the-shelf app: e.g.

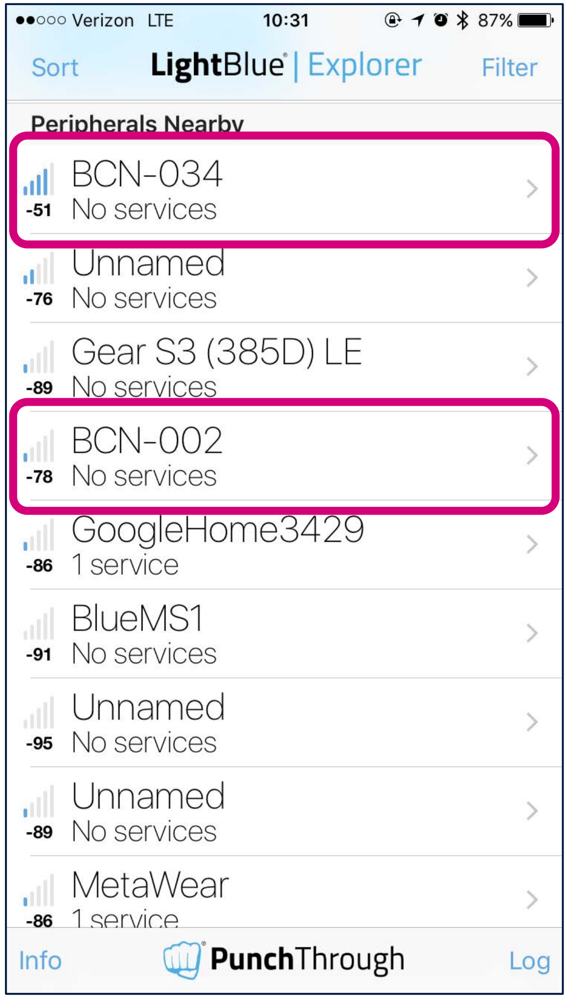
LightBlue® Explorer

Custom app: e.g.

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

Scan Results

LightBlue scan results



●●○○○ Verizon LTE 10:31 87%

Sort LightBlue | Explorer Filter

Peripherals Nearby

-51 BCN-034
No services

-76 Unnamed
No services

-89 Gear S3 (385D) LE
No services

-78 BCN-002
No services

-86 GoogleHome3429
1 service

-91 BlueMS1
No services

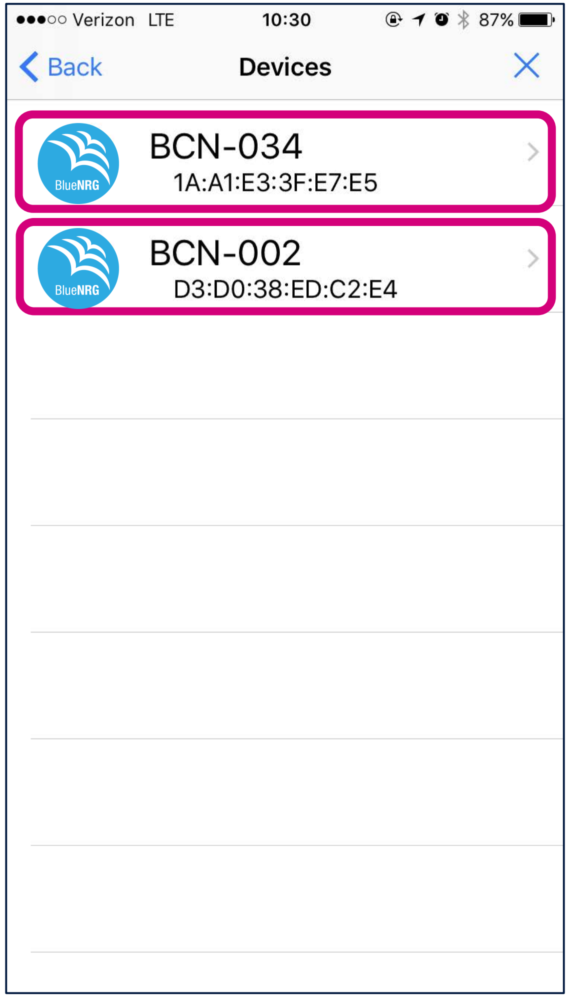
-95 Unnamed
No services

-89 Unnamed
No services

-86 MetaWear
1 service

Info PunchThrough Log

ST BLE Sensor scan results



●●●○○ Verizon LTE 10:30 87%

Back Devices

BCN-034
1A:A1:E3:3F:E7:E5

BCN-002
D3:D0:38:ED:C2:E4

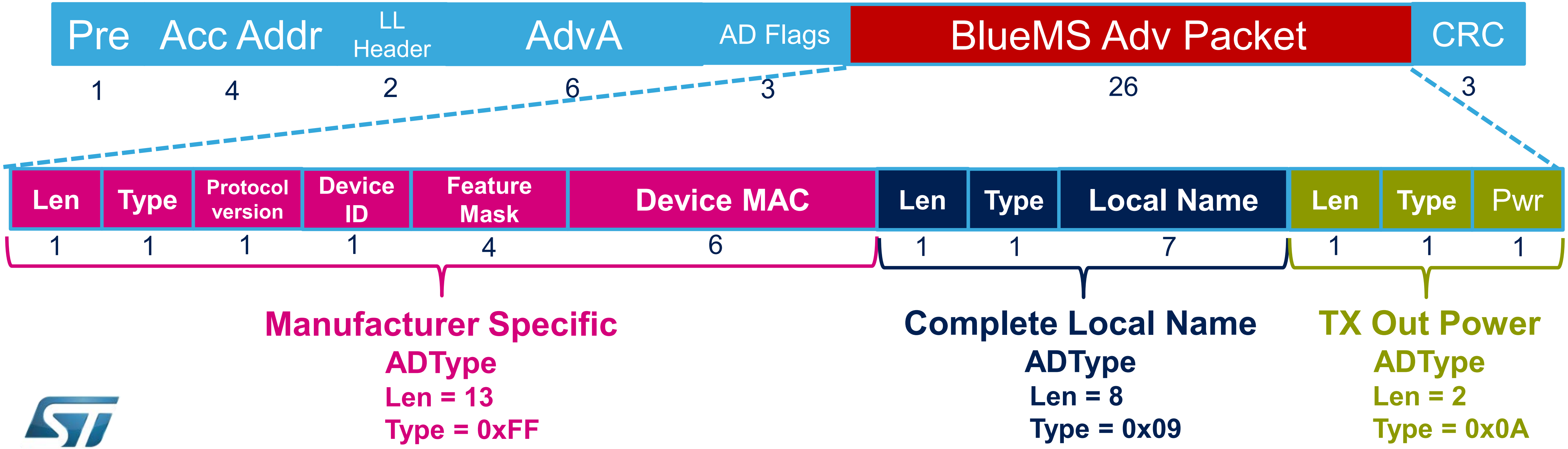
Why some devices are not present in the ST BLE Sensor app?

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

ST BlueMS Protocol

- 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

Android

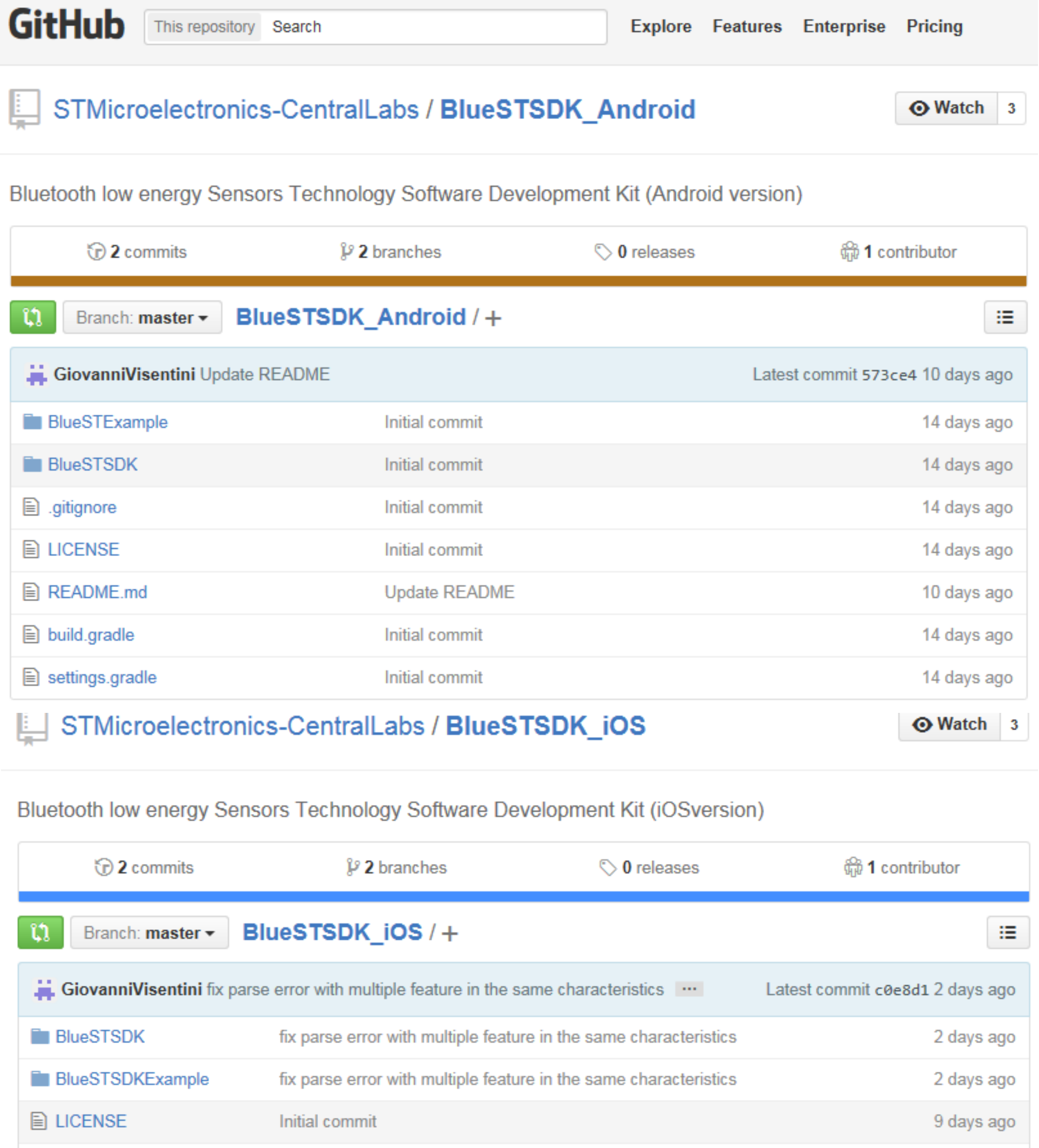
https://github.com/stmicroelectronics-centrallabs/bluestsdk_android

Source code for iOS and Android

available online and maintained on GitHub repos

iOS

https://github.com/stmicroelectronics-centrallabs/bluestsdk_ios



STMicroelectronics-CentralLabs / **BlueSTSDK_Android**

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**

Bluetooth low energy Sensors Technology Software Development Kit (iOSversion)

2 commits | **2** branches | **0** releases | **1** contributor

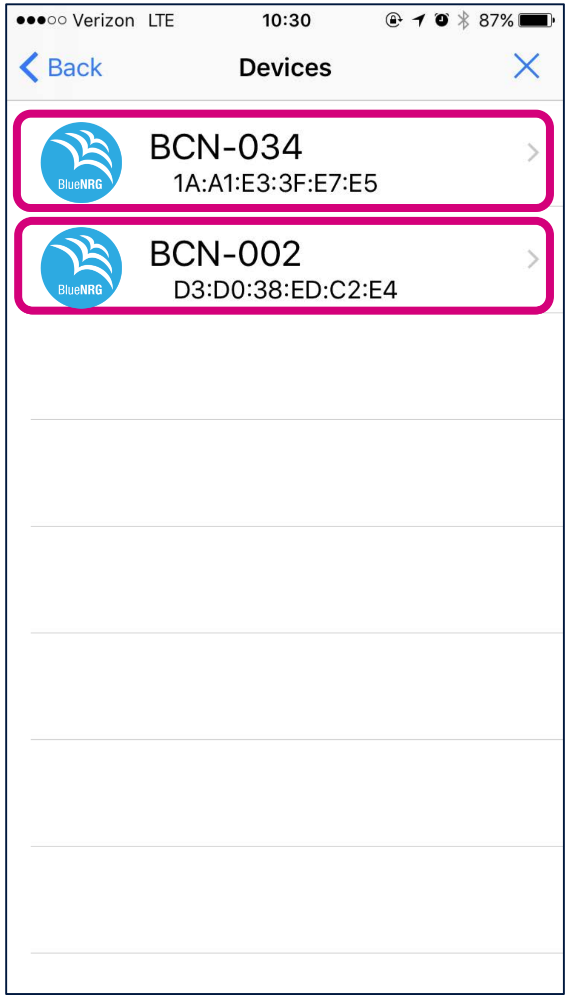
Branch: **master** | **BlueSTSDK_iOS** / +

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

ST BLE Sensor Scan Results

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



BCN-034
1A:A1:E3:3F:E7:E5

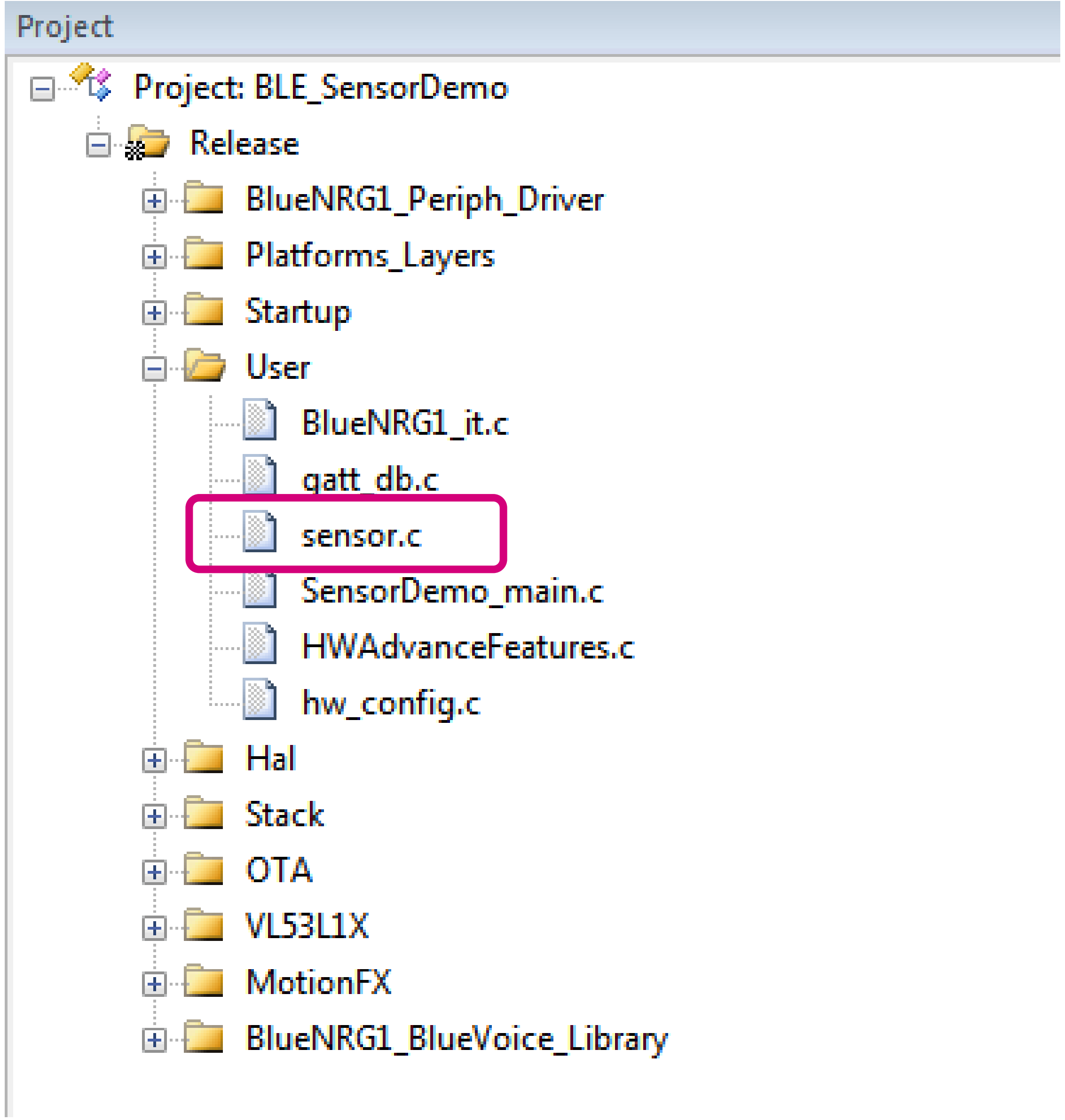
BCN-002
D3:D0:38:ED:C2:E4

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



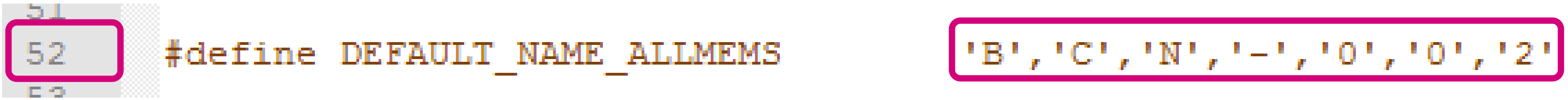
Project

- Project: BLE_SensorDemo
  - Release
    - BlueNRG1_Periph_Driver
    - Platforms_Layers
    - Startup
    - User
      - BlueNRG1_it.c
      - gatt_db.c
      - sensor.c
      - SensorDemo_main.c
      - HWAdvanceFeatures.c
      - hw_config.c
    - Hal
    - Stack
    - OTA
    - VL53L1X
    - MotionFX
    - BlueNRG1_BlueVoice_Library

Customize Your BlueTile

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

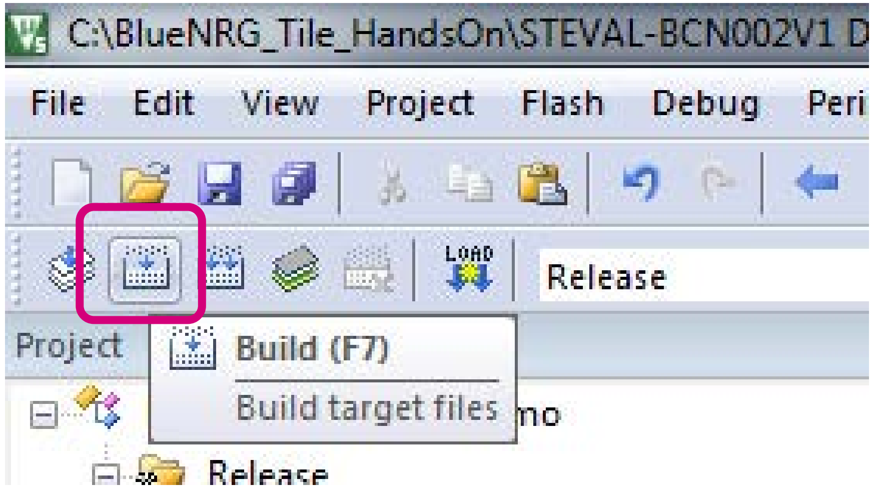
1. In the file **sensor.c** **go to** **line 52**
2. **"BCN-002" is the default local name value**
3. **You can modify it as you prefer with a 7-characters string**

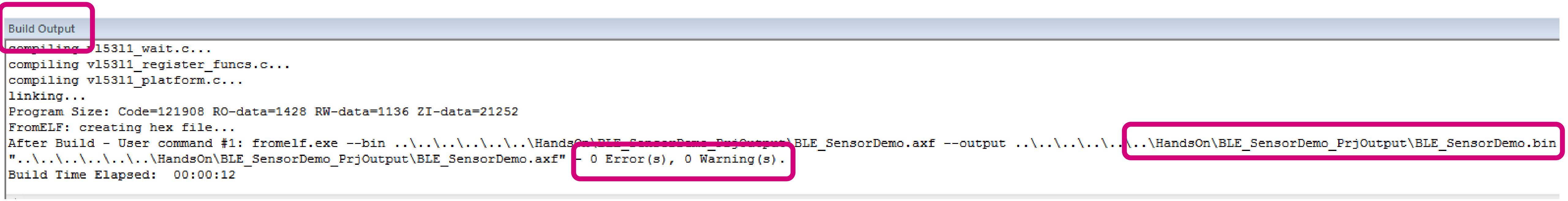


```
52    #define DEFAULT_NAME_ALLMEMS          'B','C','N','-','0','0','2'
```

Build the New Code

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





```
compiling vl5311_wait.c...
compiling vl5311_register_funcs.c...
compiling vl5311_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
```

life.augmented

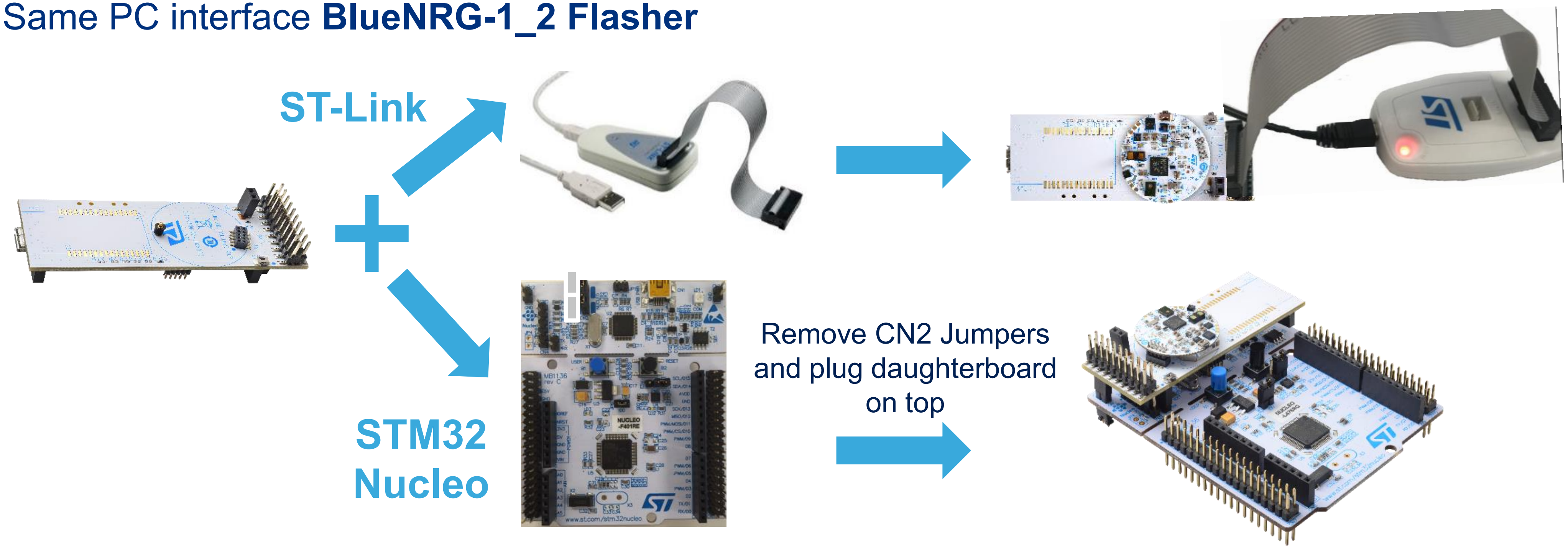
Programming Embedded Flash

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

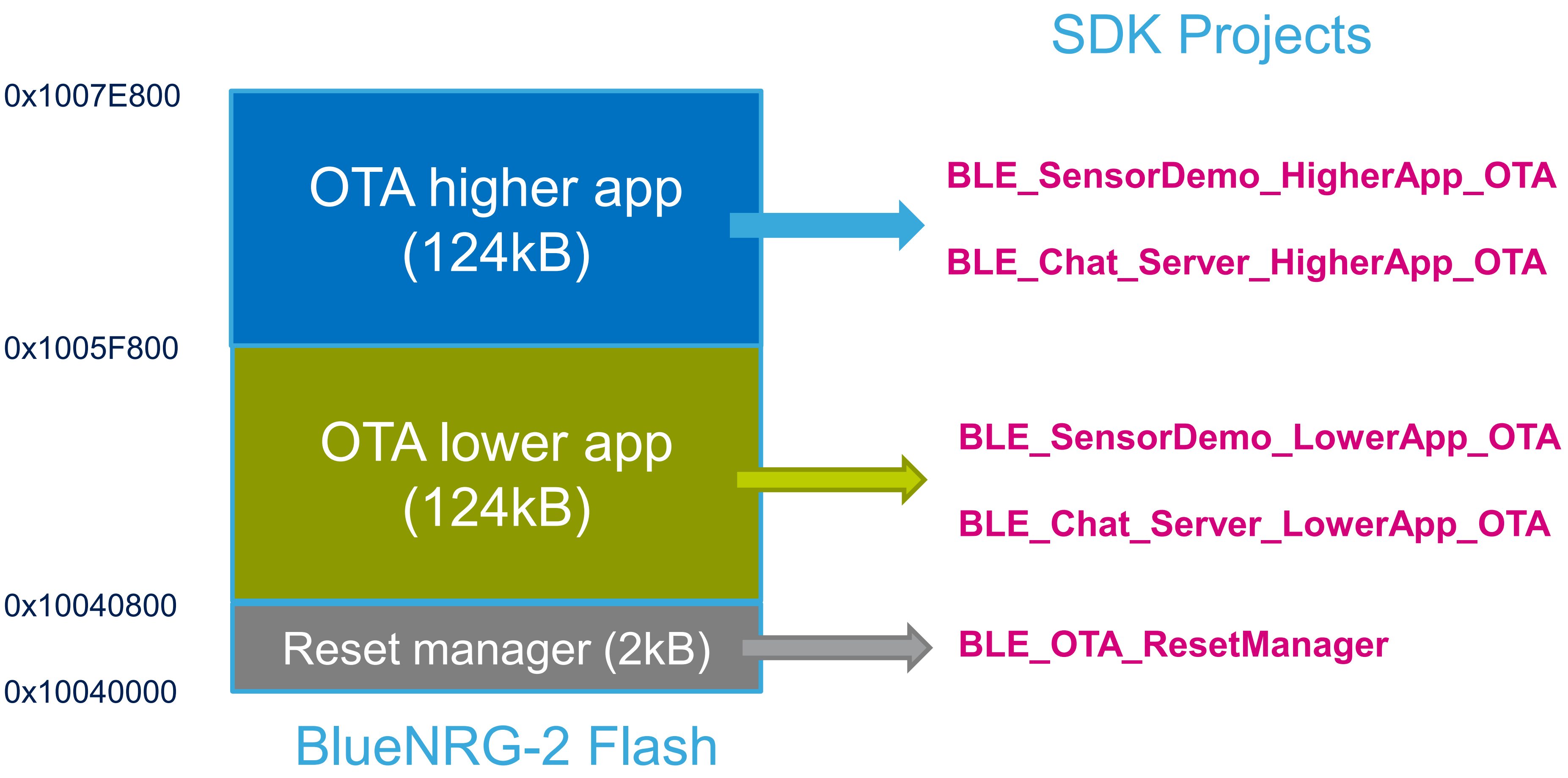
2. SWD interface + ST-LINK

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



Remove CN2 Jumpers and plug daughterboard on top

Flash Memory Layout 1/2



SDK Projects

0x1007E800

OTA higher app (124kB)

BLE_SensorDemo_HigherApp_OTA

BLE_Chat_Server_HigherApp_OTA

0x1005F800

OTA lower app (124kB)

BLE_SensorDemo_LowerApp_OTA

BLE_Chat_Server_LowerApp_OTA

0x10040800

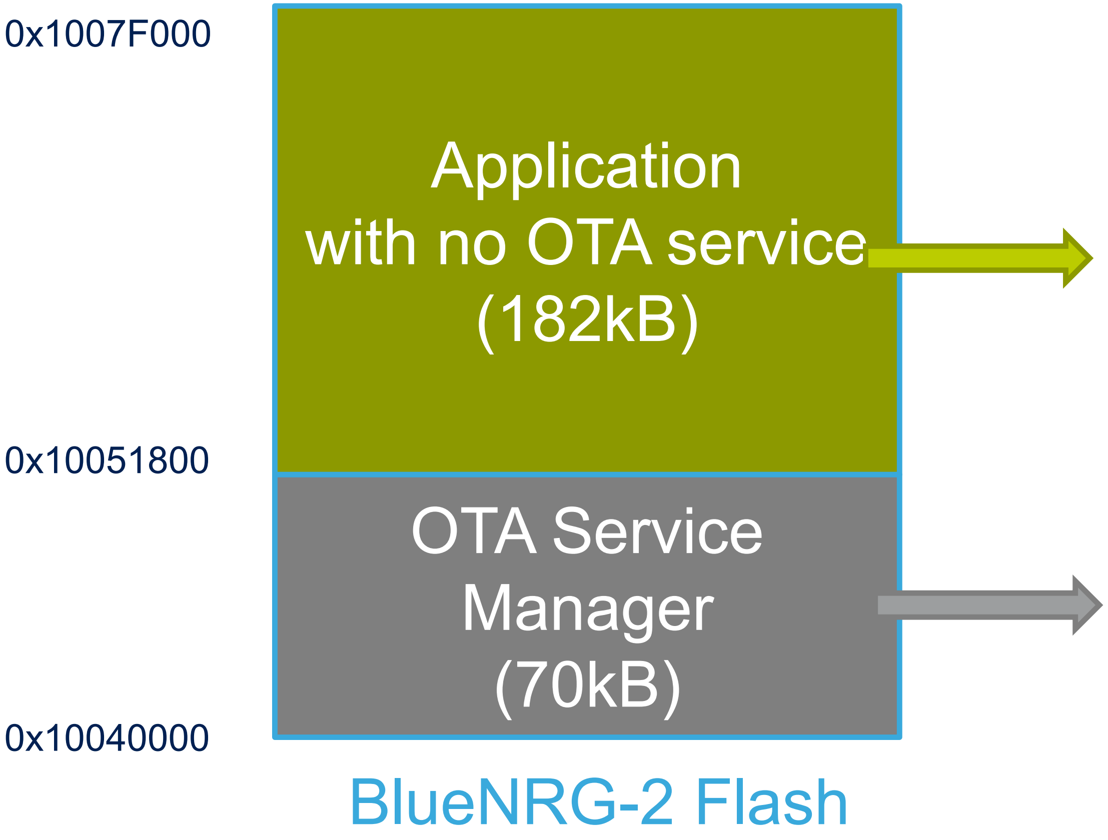
Reset manager (2kB)

BLE_OTA_ResetManager

0x10040000

BlueNRG-2 Flash

Flash Memory Layout 2/2



SDK Projects

BLE_SensorDemo_Use_OTA_ServiceManager

BLE_Chat_Server_Use_OTA_ServiceManager

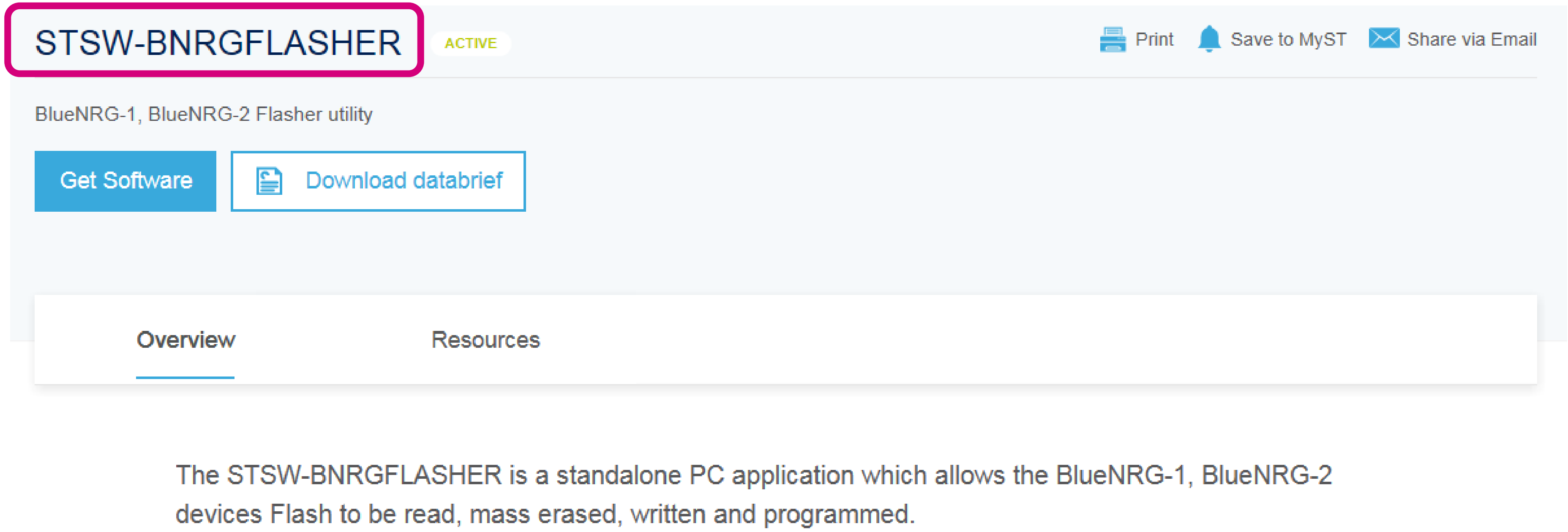
BLE_Beacon_Use_OTA_ServiceManager

BLE_OTA_ServiceManager

UART Bootloader: Use BlueNRG-2 Flasher Utility

- On st.com at this link:

  https://www.st.com/content/st_com/en/products/embedded-software/wireless-connectivity-software/stsw-bnrgflasher.html



STSW-BNRGFLASHER ACTIVE

BlueNRG-1, BlueNRG-2 Flasher utility

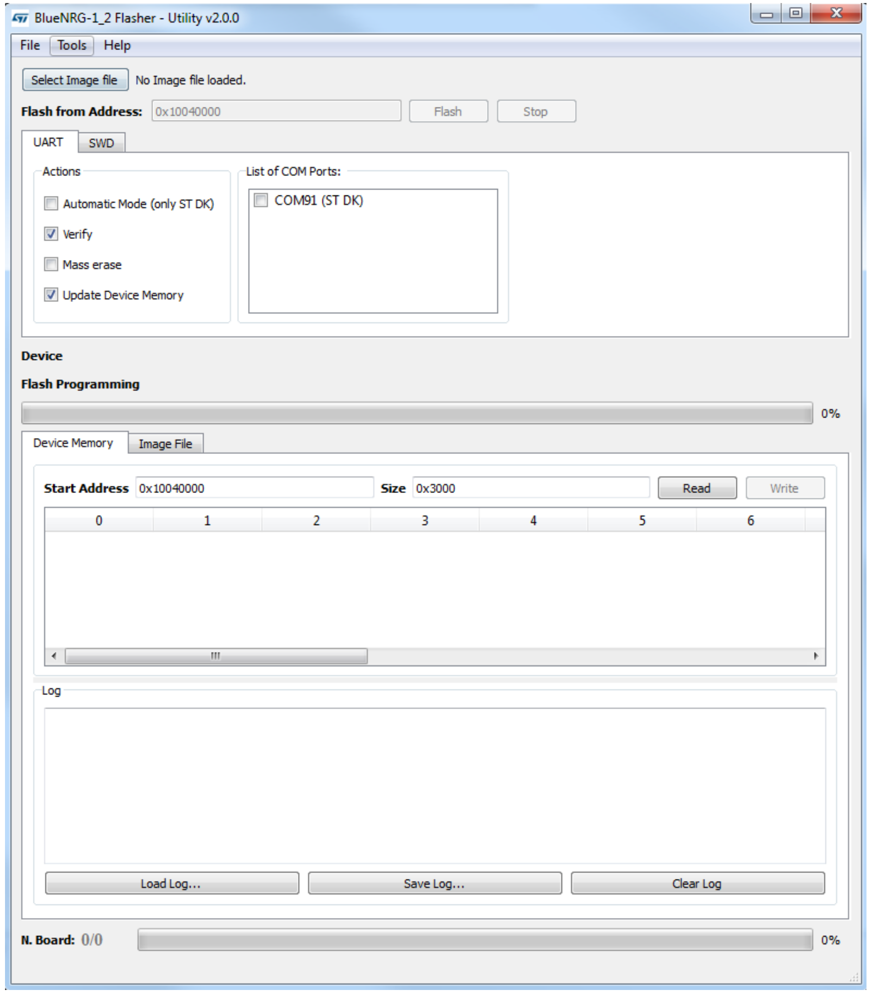
The STSW-BNRGFLASHER is a standalone PC application which allows the BlueNRG-1, BlueNRG-2 devices Flash to be read, mass erased, written and programmed.

Open the Flasher Utility

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**



BlueNRG-1_2 Flasher - Utility v2.0.0

No Image file loaded.

Flash from Address: 0x10040000

Actions

□ Automatic Mode (only ST DK)
☑ Verify
□ Mass erase
☑ Update Device Memory

List of COM Ports:

□ COM91 (ST DK)

Device

Flash Programming 0%

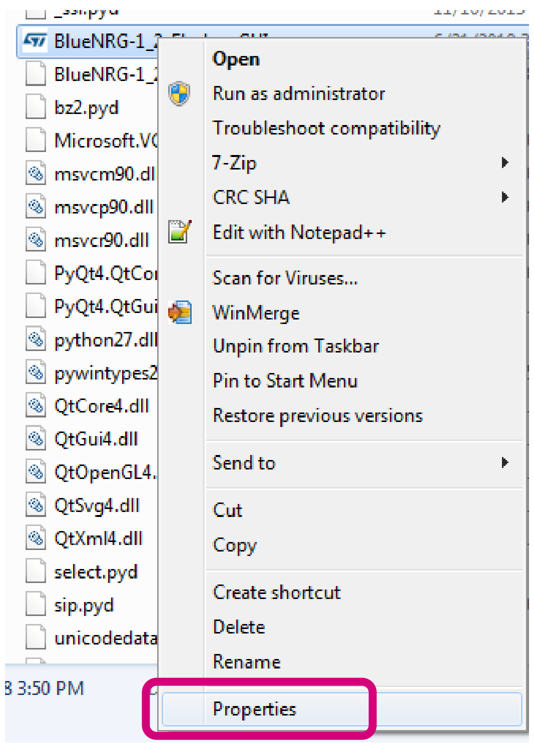
Start Address 0x10040000 Size 0x3000

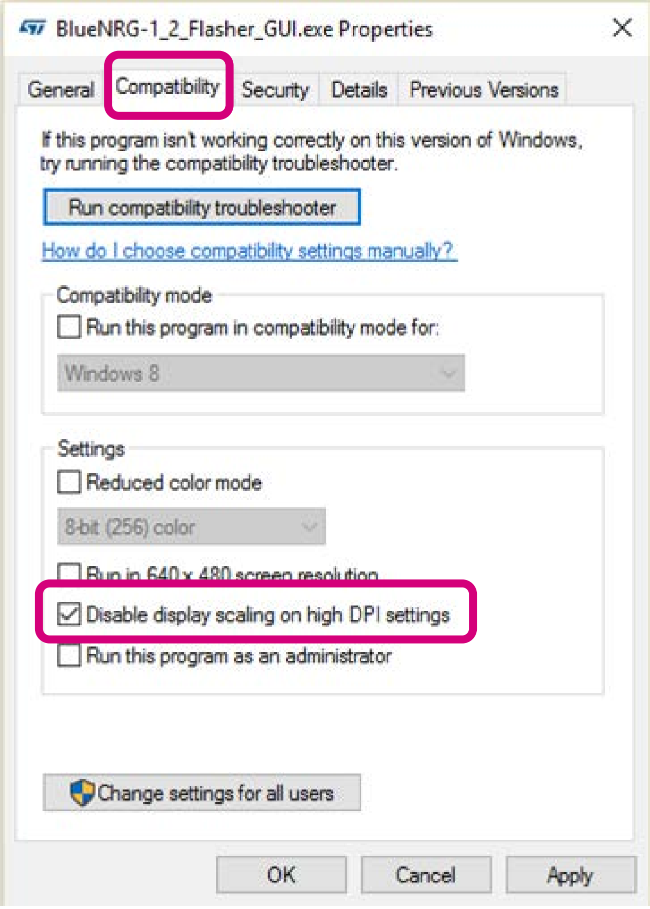
N. Board: 0/0 0%

life.augmented

For HD Screens…If You've Problem with the Resolution

- **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-1_2_Flasher_GUI.exe Properties
General
Compatibility
Security
Details
Previous Versions
Run compatibility troubleshooter
How do I choose compatibility settings manually?
Change settings for all users
OK
Cancel
Apply

If this program isn't working correctly on this version of Windows, try running the compatibility troubleshooter.

Compatibility mode

□ Run this program in compatibility mode for:

Windows 8

Settings

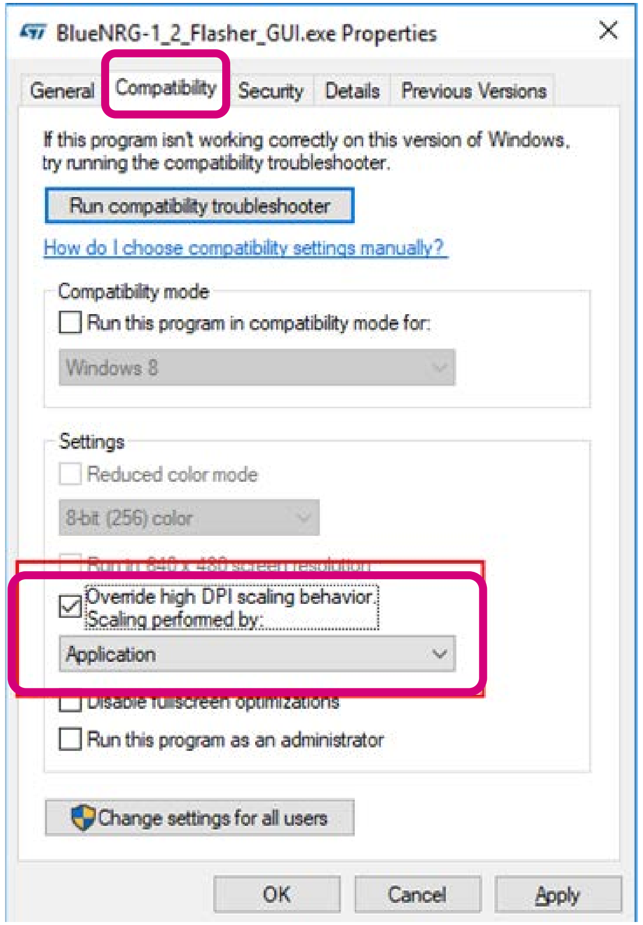
□ Reduced color mode

8-bit (256) color

□ Run in 640 x 480 screen resolution

☑ Disable display scaling on high DPI settings

□ Run this program as an administrator



BlueNRG-1_2_Flasher_GUI.exe Properties
General
Compatibility
Security
Details
Previous Versions
Run compatibility troubleshooter
How do I choose compatibility settings manually?
Change settings for all users
OK
Cancel
Apply

If this program isn't working correctly on this version of Windows, try running the compatibility troubleshooter.

Compatibility mode

□ Run this program in compatibility mode for:

Windows 8

Settings

□ Reduced color mode

8-bit (256) color

□ Run in 640 x 480 screen resolution

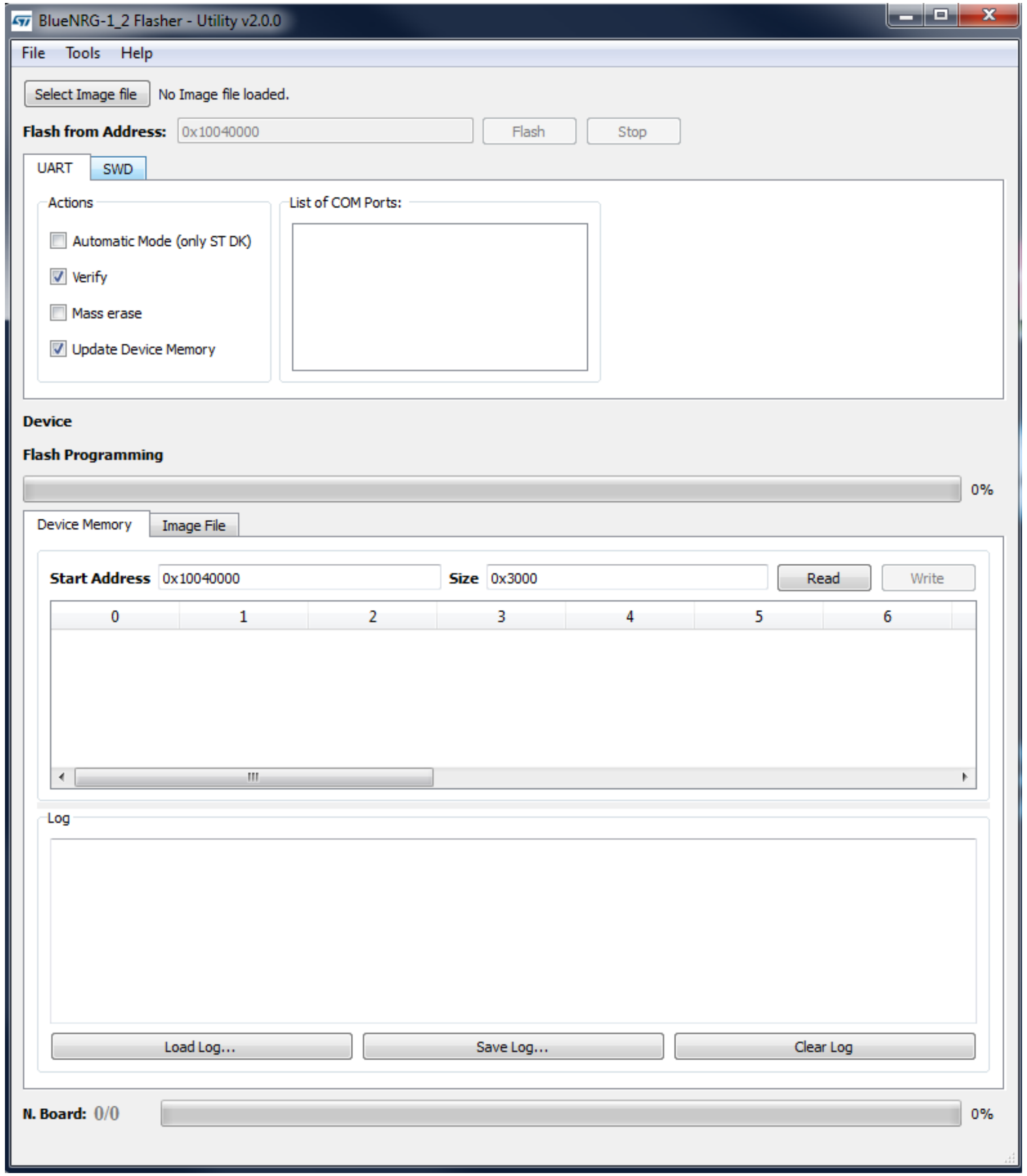
☑ Override high DPI scaling behavior. Scaling performed by:

Application

□ Disable fullscreen optimizations

□ Run this program as an administrator

BlueNRG Flasher Utility



No Image file loaded.

Flash from Address: 0x10040000

Actions

- □ Automatic Mode (only ST DK)
- ☑ Verify
- □ Mass erase
- ☑ Update Device Memory

List of COM Ports:

Device

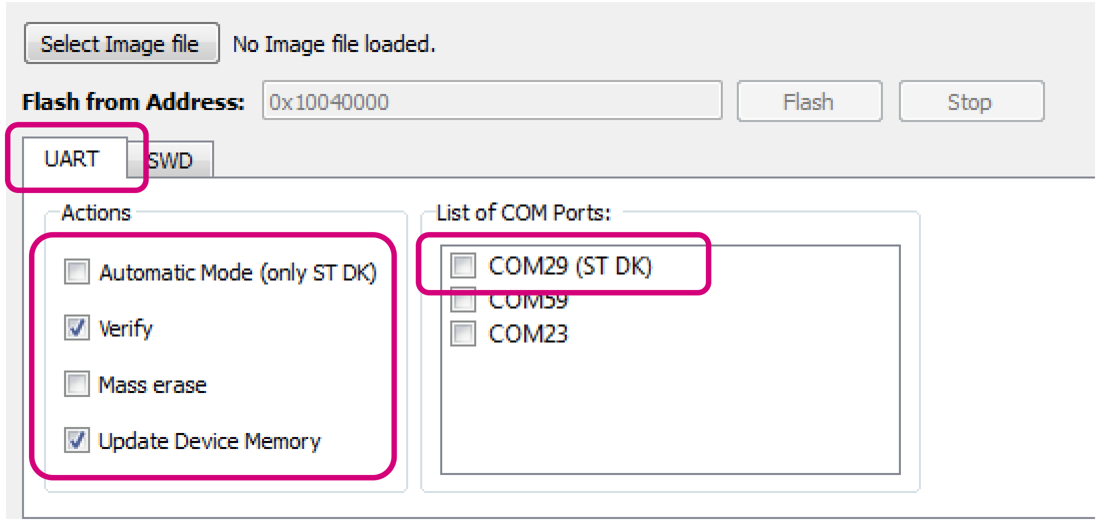
Flash Programming 0%

Start Address 0x10040000 **Size** 0x3000

Log

N. Board: 0/0 0%

UART



No Image file loaded.

Flash from Address: 0x10040000

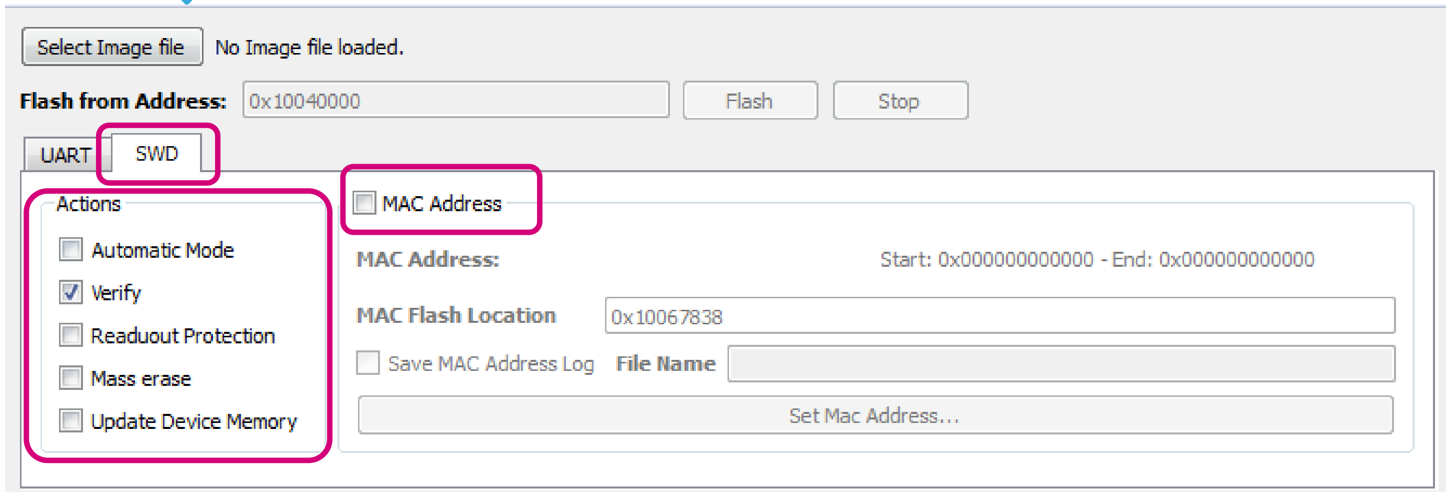
Actions

- □ Automatic Mode (only ST DK)
- ☑ Verify
- □ Mass erase
- ☑ Update Device Memory

List of COM Ports:

- □ COM29 (ST DK)
- □ COM59
- □ COM23

SWD



No Image file loaded.

Flash from Address: 0x10040000

Actions

- □ Automatic Mode
- ☑ Verify
- □ Readuout Protection
- □ Mass erase
- □ Update Device Memory

□ MAC Address

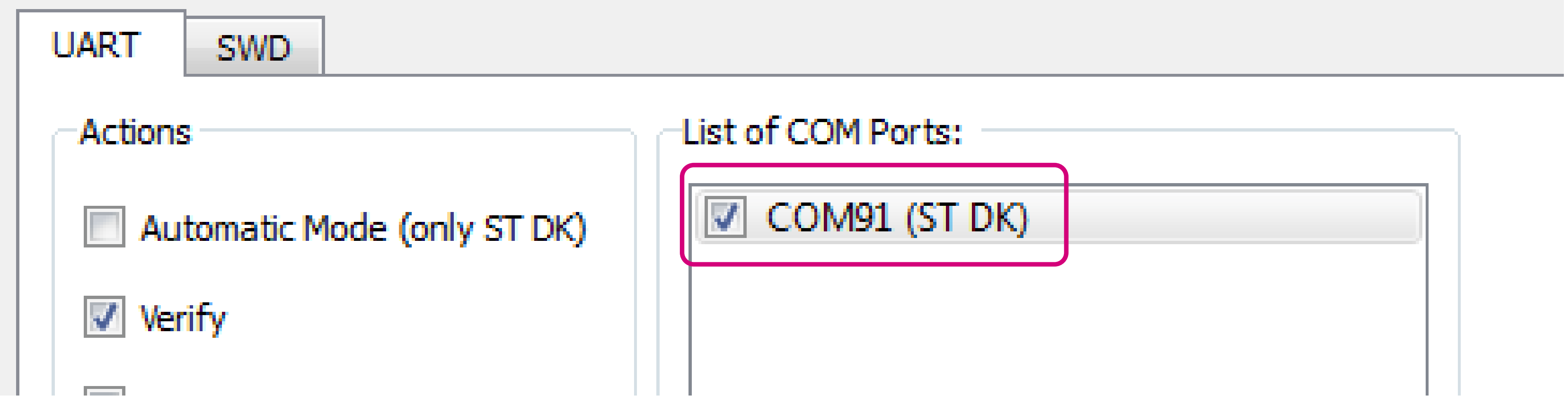
MAC Address: Start: 0x000000000000 - End: 0x000000000000

MAC Flash Location 0x10067838

□ Save MAC Address Log **File Name**

Flash the BlueNRG-2 1/6

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



Actions

☐ Automatic Mode (only ST DK)

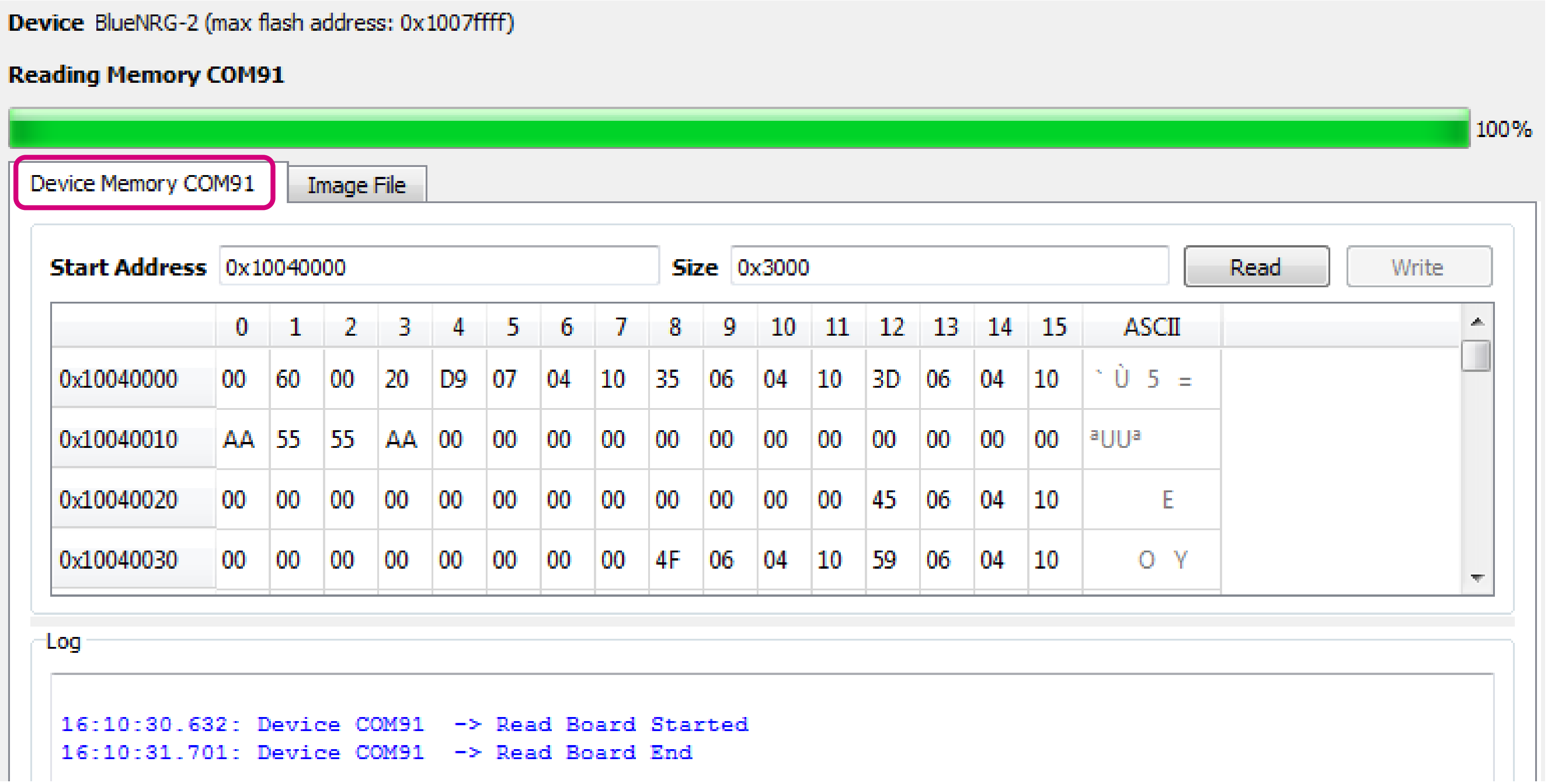
☑ Verify

List of COM Ports:

☑ COM91 (ST DK)

Flash the BlueNRG-2 2/6

1. **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

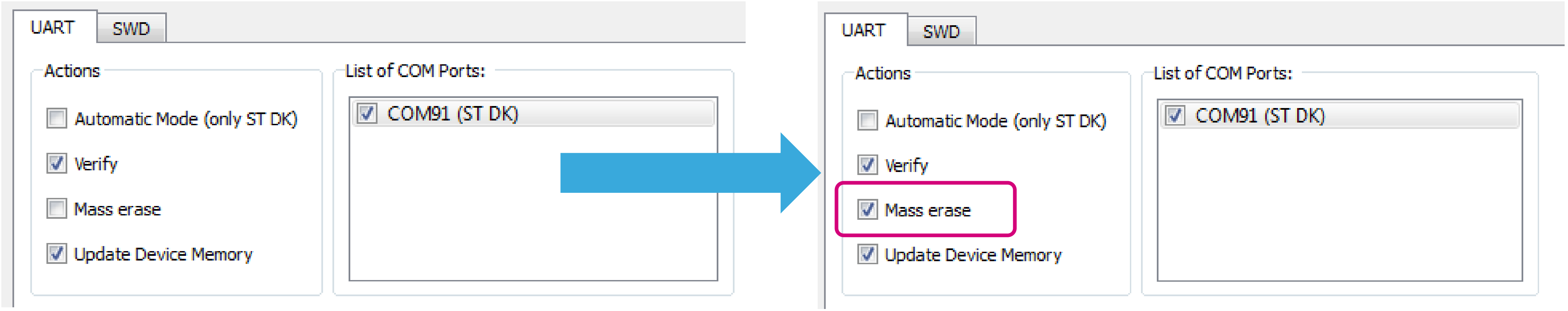
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ASCII
0x10040000	00	60	00	20	D9	07	04	10	35	06	04	10	3D	06	04	10	` Ù 5 =
0x10040010	AA	55	55	AA	00	00	00	00	00	00	00	00	00	00	00	00	ªUUª
0x10040020	00	00	00	00	00	00	00	00	00	00	00	00	45	06	04	10	E
0x10040030	00	00	00	00	00	00	00	00	4F	06	04	10	59	06	04	10	O Y

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

1. **Select Mass Erase**



Actions

- ☐ Automatic Mode (only ST DK)
- ☑ Verify
- ☐ Mass erase
- ☑ Update Device Memory

List of COM Ports:

- ☑ COM91 (ST DK)

Actions

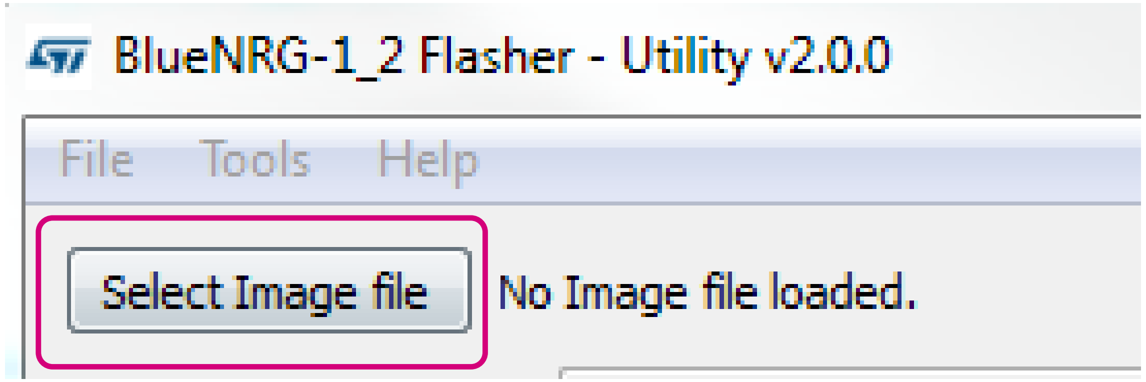
- ☐ Automatic Mode (only ST DK)
- ☑ Verify
- ☑ Mass erase
- ☑ Update Device Memory

List of COM Ports:

- ☑ COM91 (ST DK)

Flash the BlueNRG-2 4/6

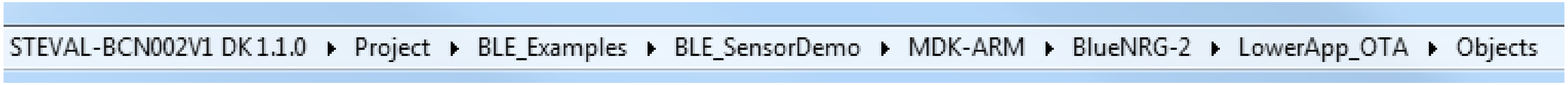
1. **Click** on the **Select Image file** button



BlueNRG-1_2 Flasher - Utility v2.0.0

No Image file loaded.

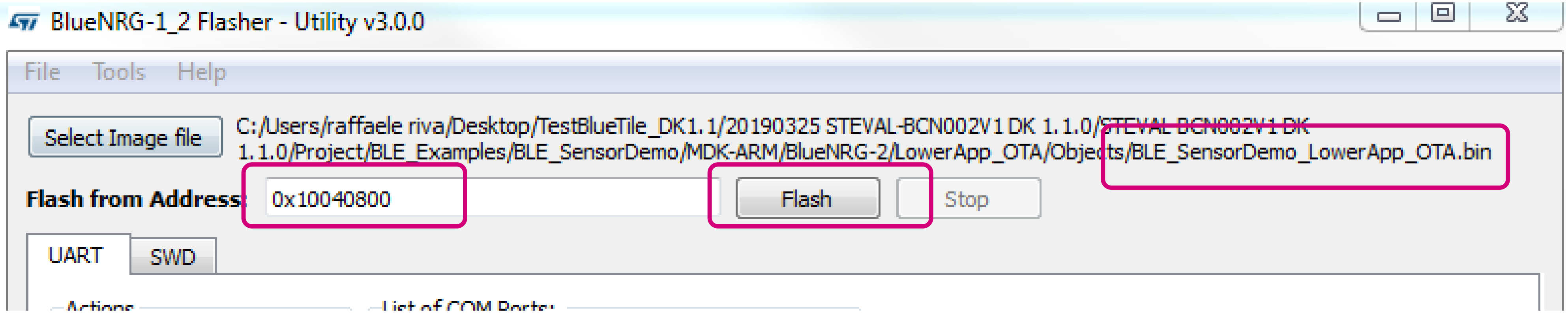
2. And **browse** the following **path**



STEVAL-BCN002V1 DK 1.1.0 ▸ Project ▸ BLE_Examples ▸ BLE_SensorDemo ▸ MDK-ARM ▸ BlueNRG-2 ▸ LowerApp_OTA ▸ Objects

Flash the BlueNRG-2 5/6

1. **Select** the binary file (e.g. BLE_SensorDemo_LowerApp_OTA.bin) and **click** **Open**
2. Insert the correct **start address** (e.g. 0x10040800 for Lower App)
3. **Click** on the **Flash** button



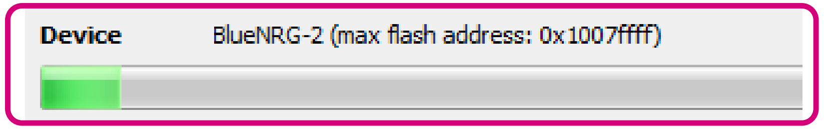
BlueNRG-1_2 Flasher - Utility v3.0.0

C:/Users/raffaele riva/Desktop/TestBlueTile_DK1.1/20190325 STEVAL-BCN002V1 DK 1.1.0/~~STEVAL-BCN002V1 DK~~ 1.1.0/Project/BLE_Examples/BLE_SensorDemo/MDK-ARM/BlueNRG-2/LowerApp_OTA/Objects/BLE_SensorDemo_LowerApp_OTA.bin

Flash from Address: 0x10040800

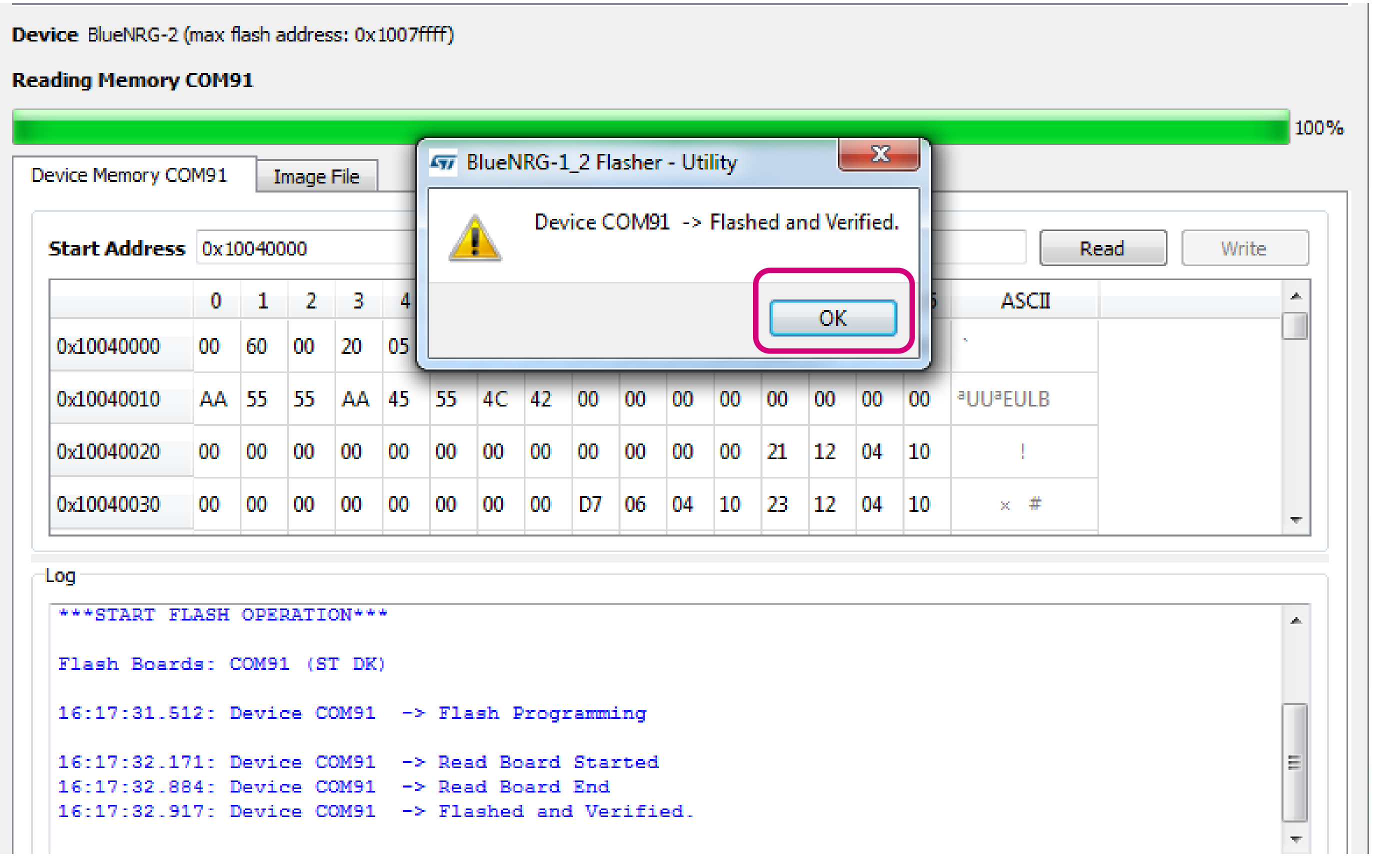
Flash the BlueNRG-2 6/6

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



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

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



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

Reading Memory COM91

100%

BlueNRG-1_2 Flasher - Utility

Device COM91 -> Flashed and Verified.

Start Address 0x10040000

	0	1	2	3	4											5	ASCII
0x10040000	00	60	00	20	05												`
0x10040010	AA	55	55	AA	45	55	4C	42	00	00	00	00	00	00	00	00	ªUUªEULB
0x10040020	00	00	00	00	00	00	00	00	00	00	00	00	21	12	04	10	!
0x10040030	00	00	00	00	00	00	00	00	D7	06	04	10	23	12	04	10	× #

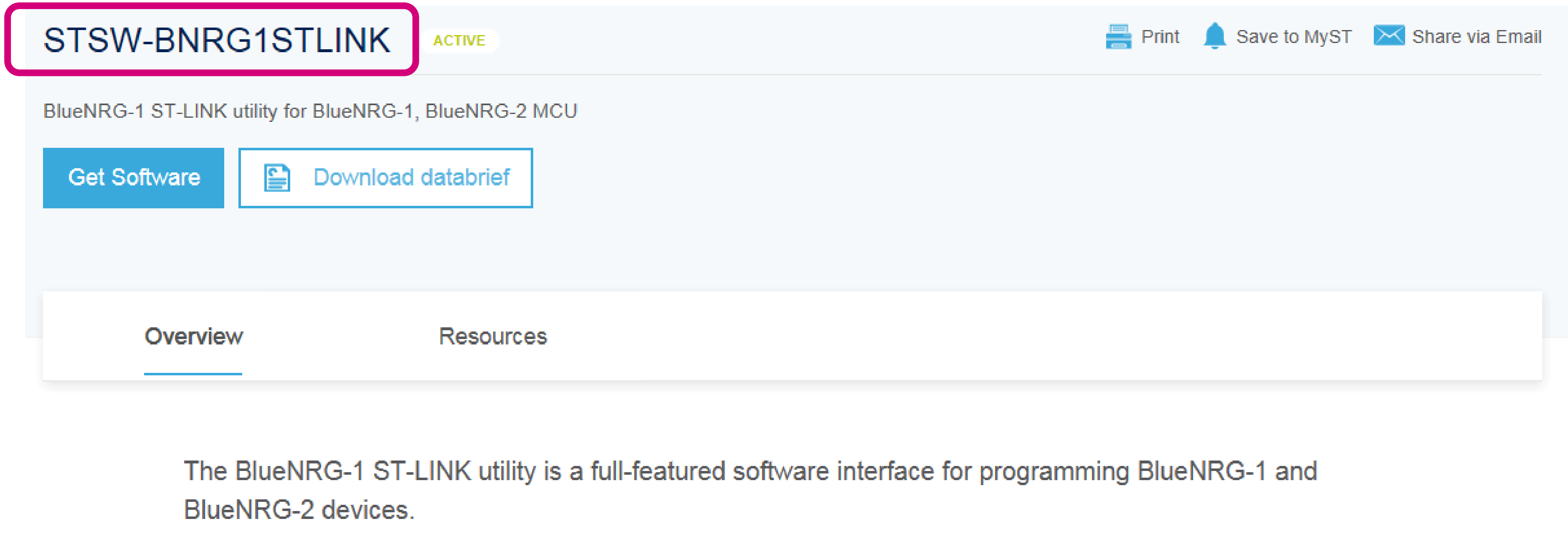
Log

```
***START FLASH OPERATION***

Flash Boards: COM91 (ST DK)

16:17:31.512: Device COM91  -> Flash Programming

16:17:32.171: Device COM91  -> Read Board Started
16:17:32.884: Device COM91  -> Read Board End
16:17:32.917: Device COM91  -> Flashed and Verified.
```

SWD: Use BlueNRG ST-LINK Utility

- On st.com at this link:

  https://www.st.com/content/st_com/en/products/embedded-software/wireless-connectivity-software/stsw-bnrg1stlink.html



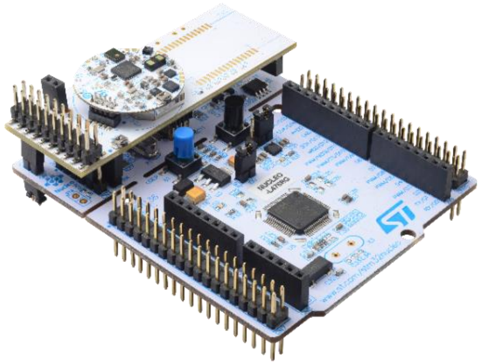
STSW-BNRG1STLINK ACTIVE

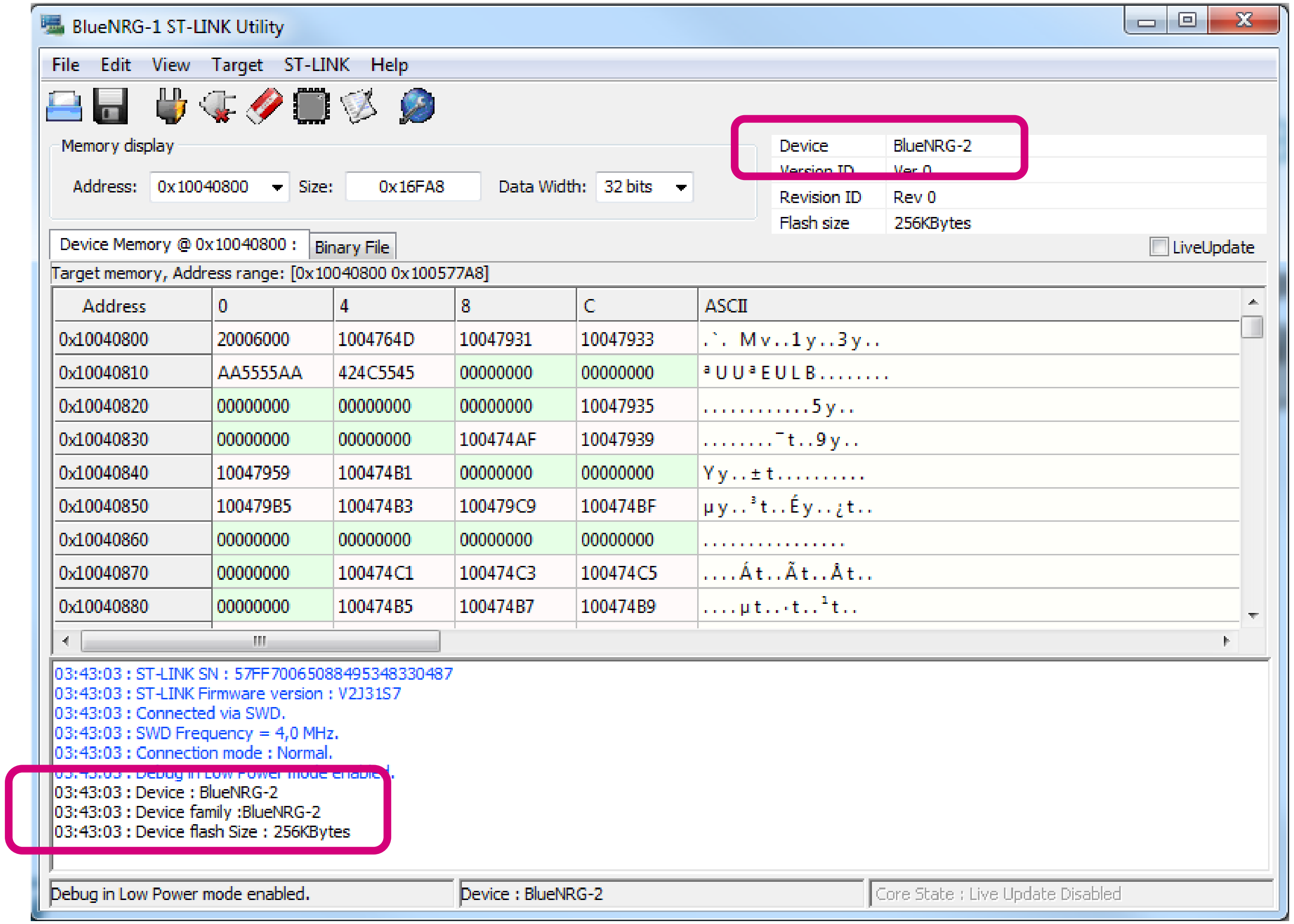
BlueNRG-1 ST-LINK utility for BlueNRG-1, BlueNRG-2 MCU

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

Programming Embedded Flash

1. SWD interface + ST-LINK




BlueNRG-1 ST-LINK Utility
File Edit View Target ST-LINK Help
Memory display
Device Memory @ 0x10040800 :
Binary File
□ LiveUpdate

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

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

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	ªUUªEULB........
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	µy..³t..Éy..¿t..
0x10040860	00000000	00000000	00000000	00000000	....
0x10040870	00000000	100474C1	100474C3	100474C5	....Át..Ãt..Åt..
0x10040880	00000000	100474B5	100474B7	100474B9	....µt..·t..¹t..

```
03:43:03 : ST-LINK SN : 57FF700650884953483330487
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
```

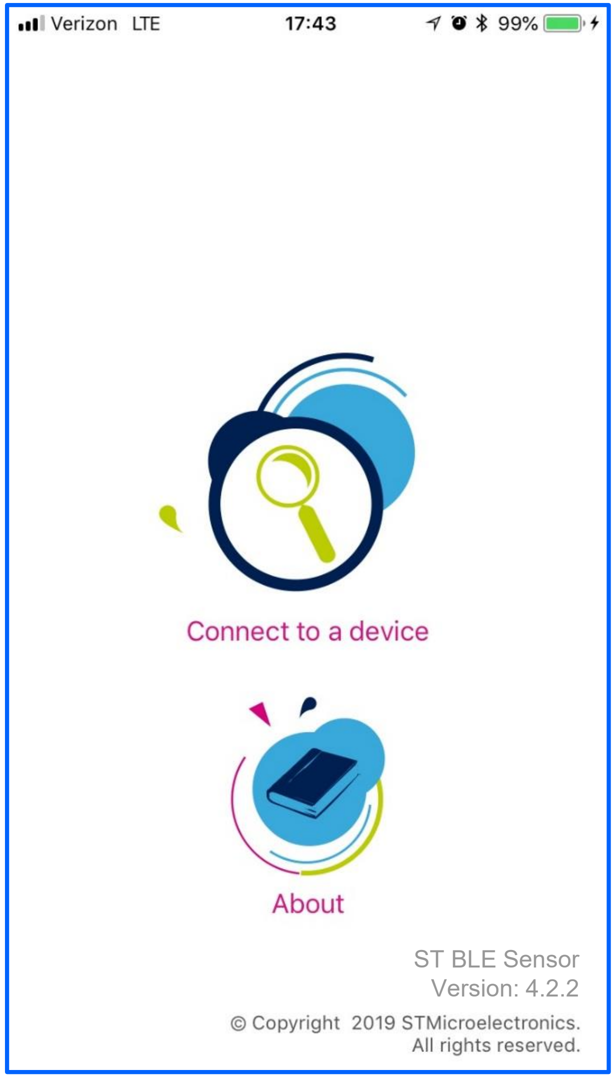
Debug in Low Power mode enabled. | Device : BlueNRG-2 | Core State : Live Update Disabled

Open the ST BLE Sensor App

Download on the App Store

GET IT ON Google Play

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



ST BLE Sensor
Version: 4.2.2

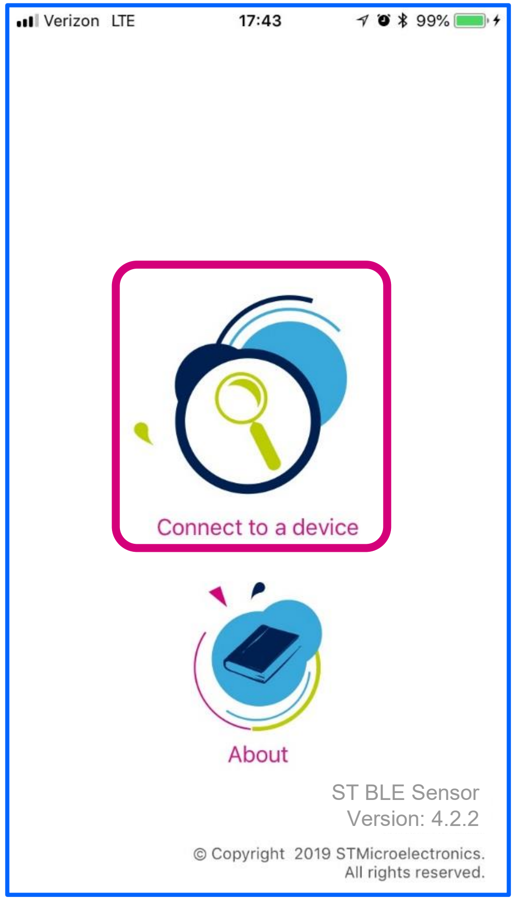
© Copyright 2019 STMicroelectronics.
All rights reserved.

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

1

Tap "Connect to a device"

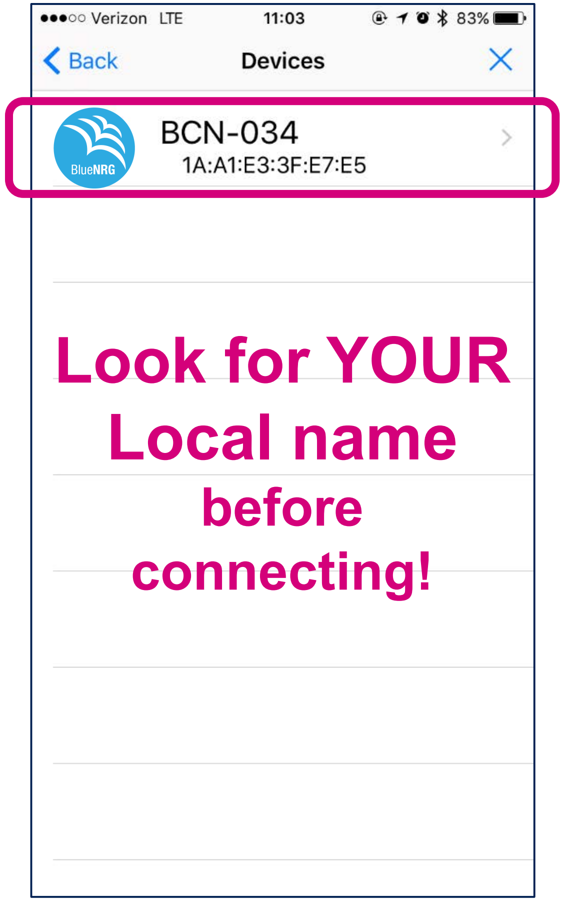


ST BLE Sensor
Version: 4.2.2

© Copyright 2019 STMicroelectronics.
All rights reserved.

2

Select your BlueTile

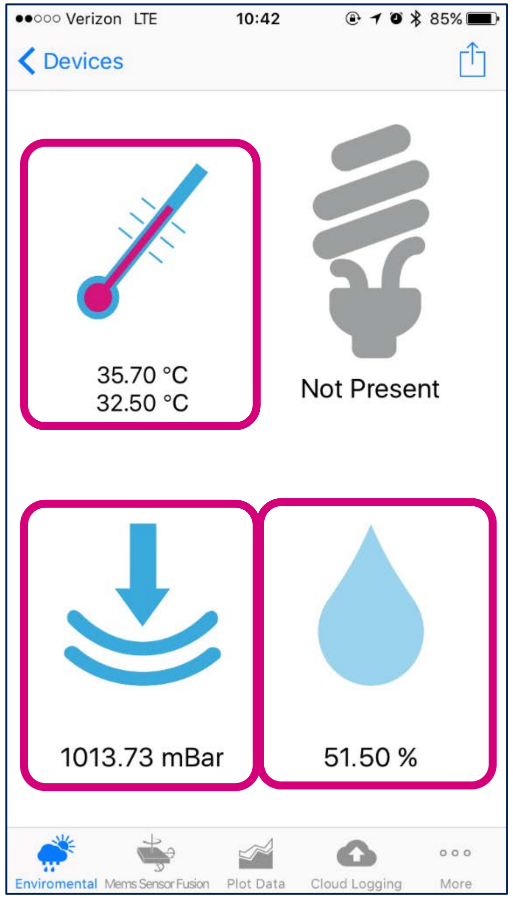


BCN-034
1A:A1:E3:3F:E7:E5

Look for YOUR Local name before connecting!

3

You are connected



35.70 °C
32.50 °C

Not Present

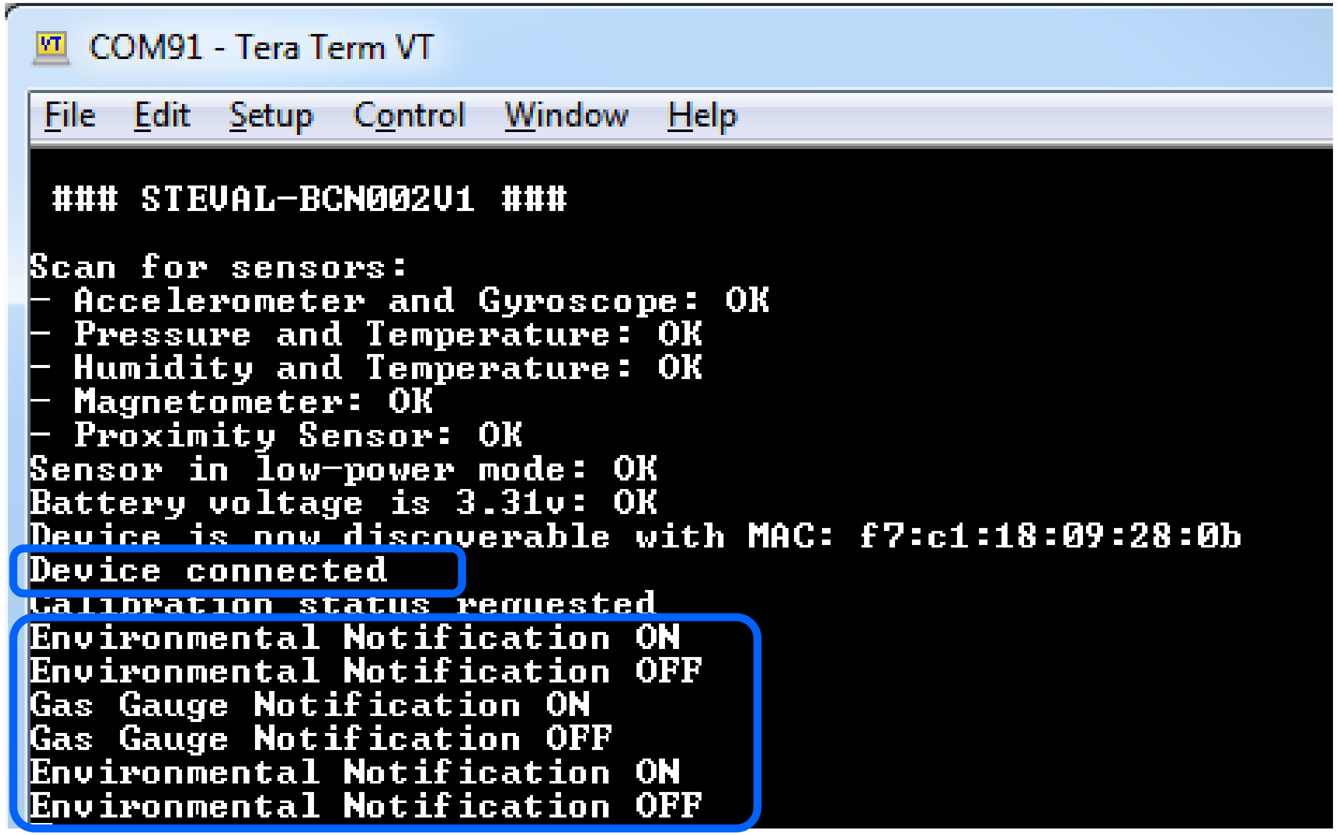
1013.73 mBar

51.50 %

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

Tera Term Output

- **“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

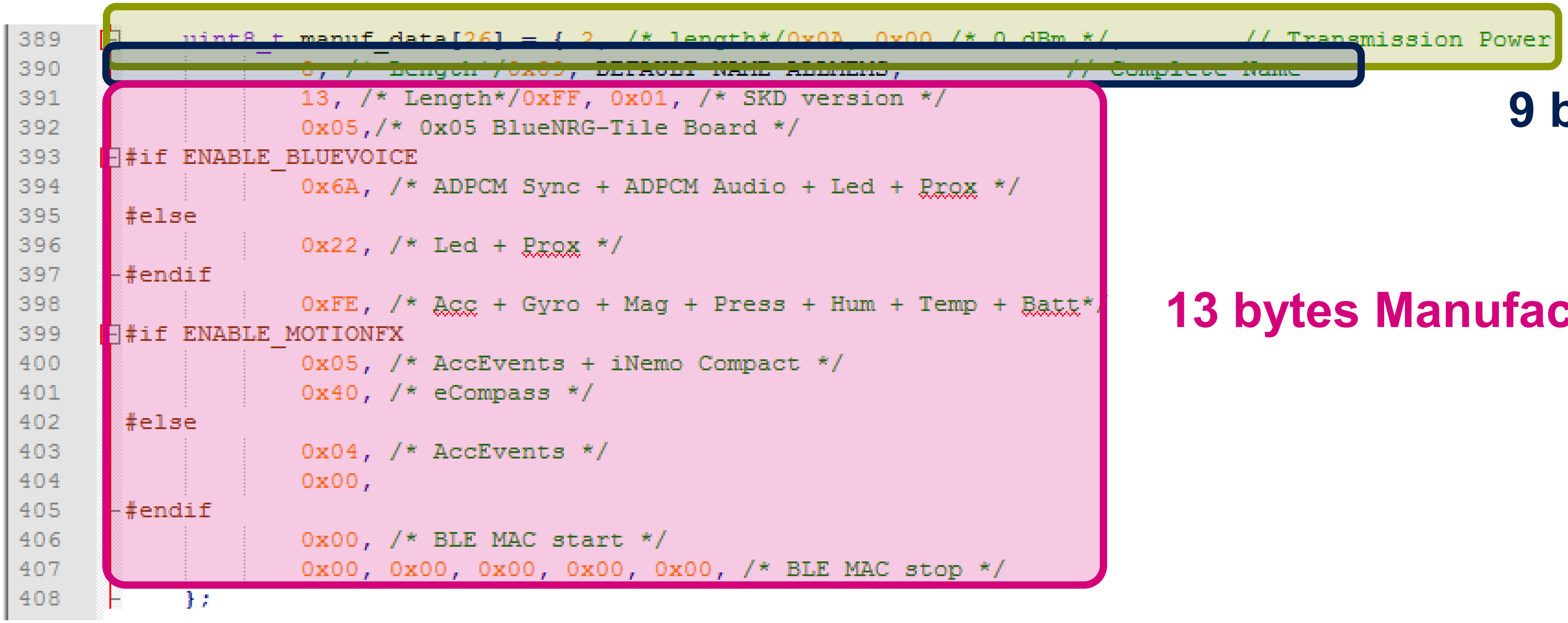


```
 ### STEVAL-BCN002V1 ###

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

In file **sensor.c**

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



```
    uint8_t manuf_data[26] = { 2, /* length*/0x0A, 0x00 /* 0 dBm */,          // Transmission Power
            8, /* Length*/0x09, DEFAULT_NAME_ALLMEMS,                 // Complete Name
            13, /* Length*/0xFF, 0x01, /* SKD version */
            0x05,/* 0x05 BlueNRG-Tile Board */
#if ENABLE_BLUEVOICE
            0x6A, /* ADPCM Sync + ADPCM Audio + Led + Prox */
#else
            0x22, /* Led + Prox */
#endif
            0xFE, /* Acc + Gyro + Mag + Press + Hum + Temp + Batt*/
#if ENABLE_MOTIONFX
            0x05, /* AccEvents + iNemo Compact */
            0x40, /* eCompass */
#else
            0x04, /* AccEvents */
            0x00,
#endif
            0x00, /* BLE MAC start */
            0x00, 0x00, 0x00, 0x00, 0x00, /* BLE MAC stop */
    };
```

3 bytes TX output power

9 bytes Local Name

13 bytes Manufacturer Specific

life.augmented

BlueMS Protocol

Complete Feature Mask

4 Bytes

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

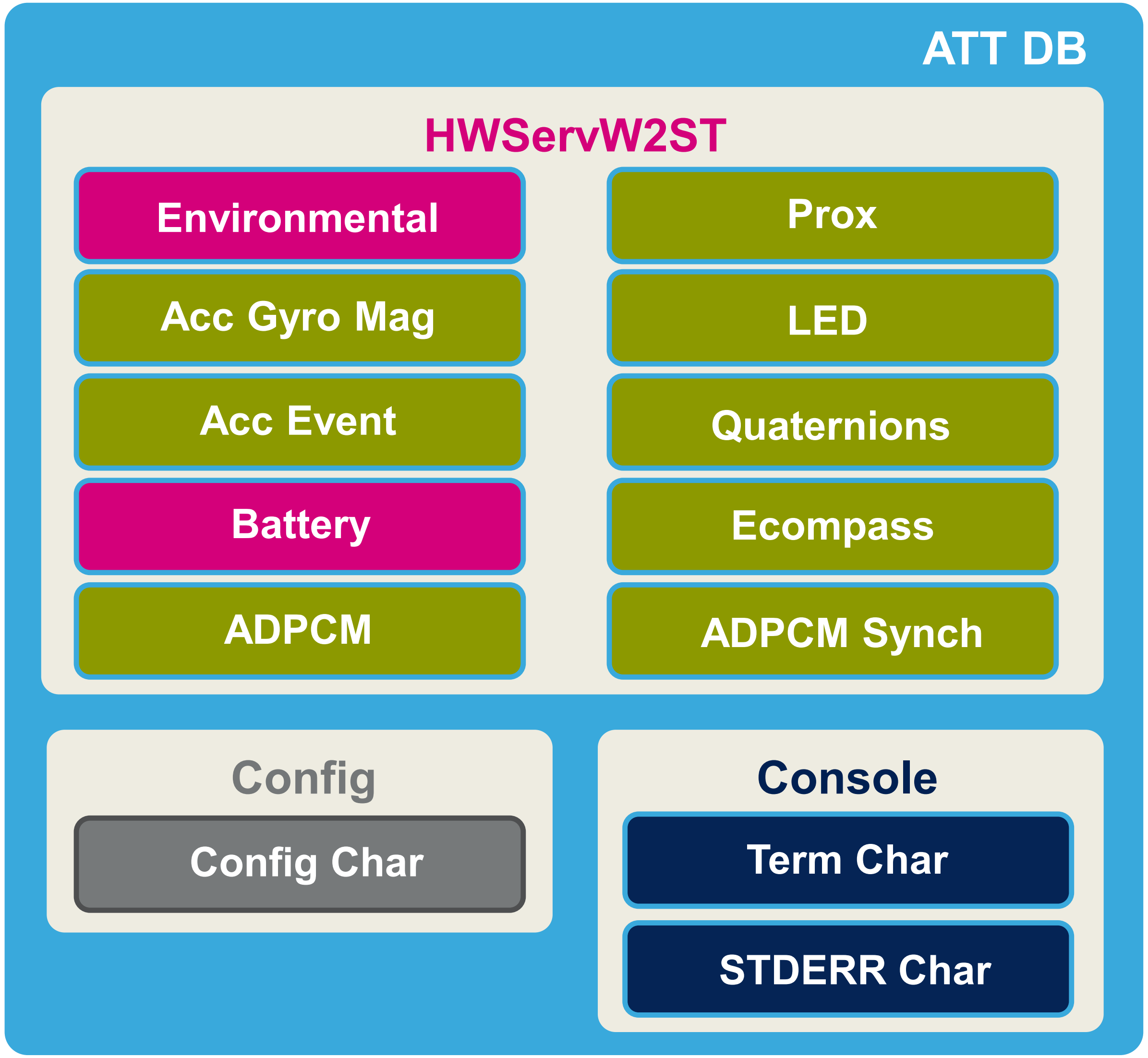
After all labs

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	

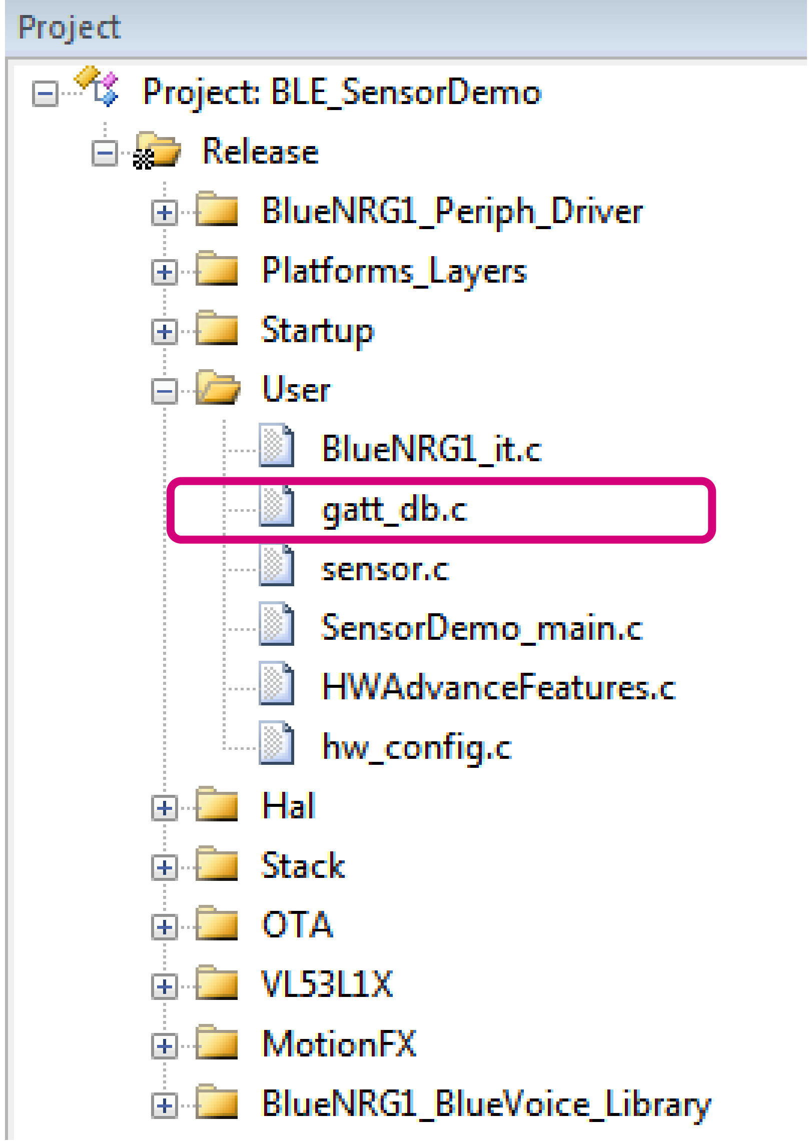
life.augmented

How Feature Mask is Mapped onto ATT DB?

- **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

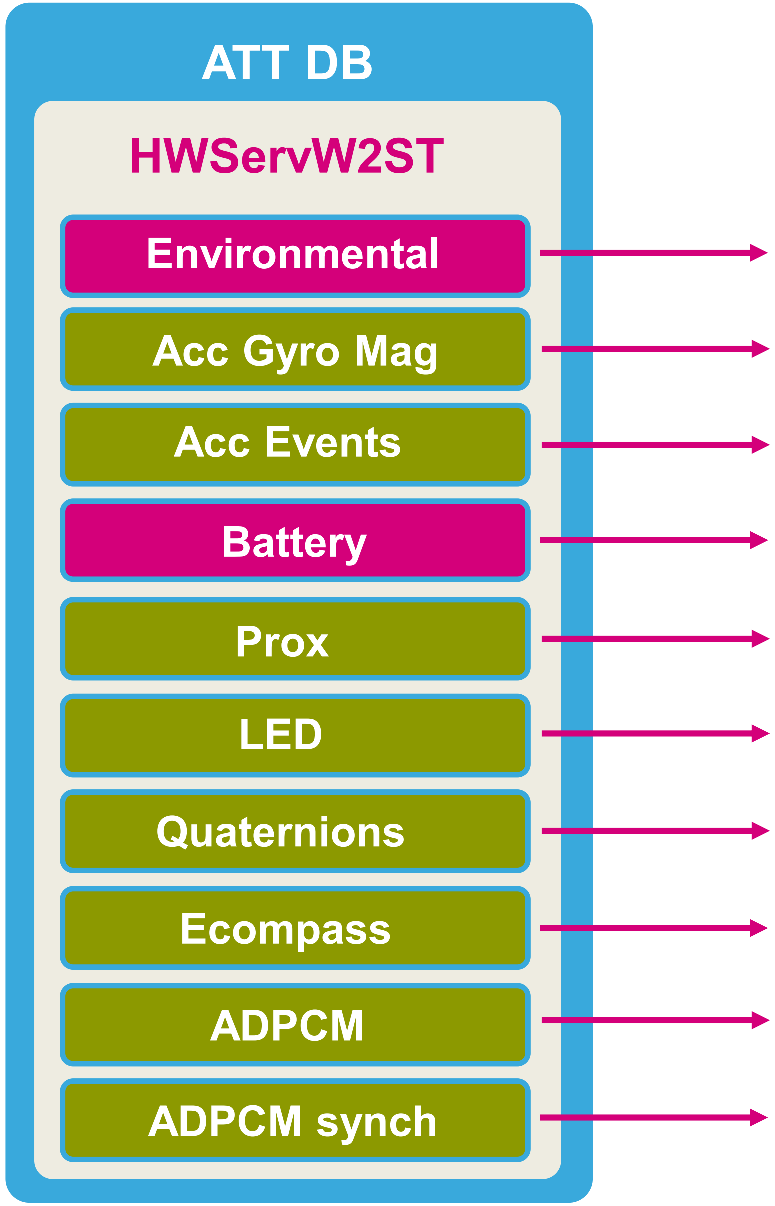


Project

- Project: BLE_SensorDemo
  - Release
    - BlueNRG1_Periph_Driver
    - Platforms_Layers
    - Startup
    - User
      - BlueNRG1_it.c
      - gatt_db.c
      - sensor.c
      - SensorDemo_main.c
      - HWAdvanceFeatures.c
      - hw_config.c
    - Hal
    - Stack
    - OTA
    - VL53L1X
    - MotionFX
    - BlueNRG1_BlueVoice_Library

The **ATT DB** is defined and created in the file **gatt_db.c**

BLE_SensorDemo ATT DB



Len	PROPERTIES	UUID TYPE	UUID VALUE
12	N,R	128-bit	001D0000000111e1ac360002a5d5c51b
20	N	128-bit	00E0000000111e1ac360002a5d5c51b
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

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	

ATT DB

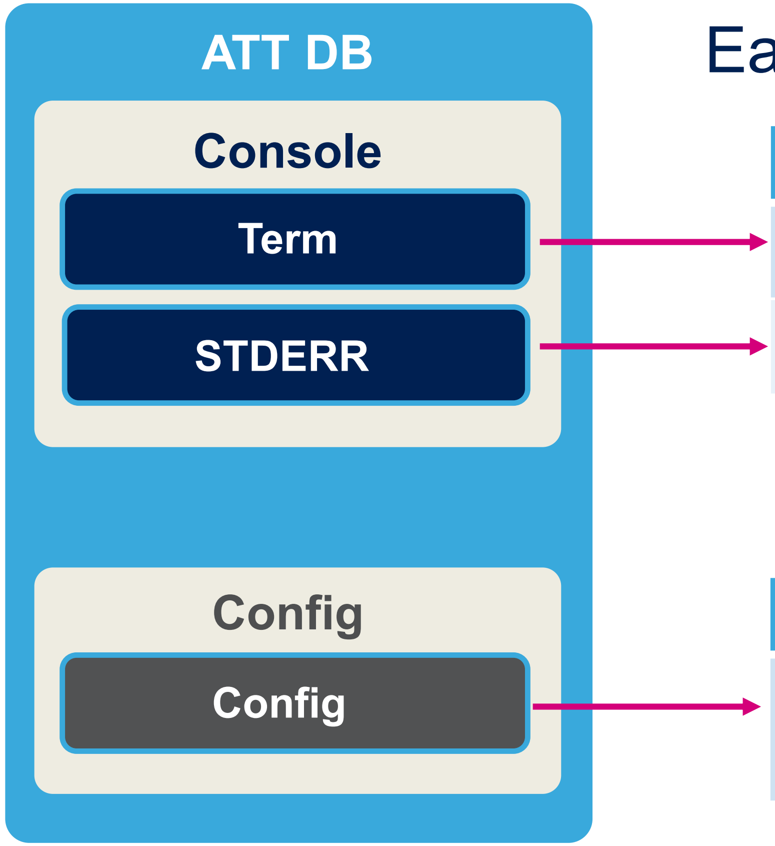
HWServW2ST

Battery

UUID VALUE
00020000000111e1ac360002a5d5c51b

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

BLE_SensorDemo ATT DB



Each characteristic is defined by the following key parameters

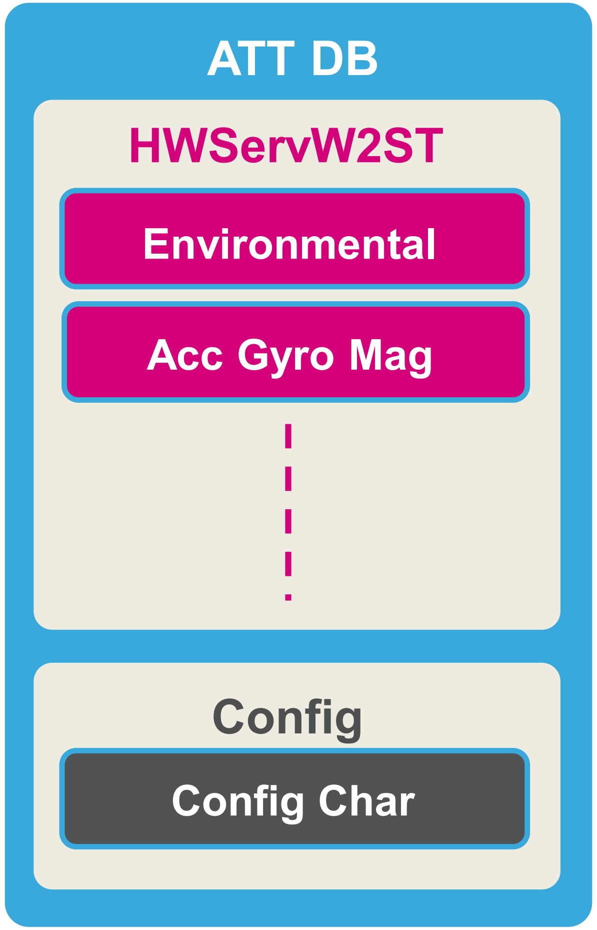
Len	PROPERTIES	UUID TYPE	UUID VALUE
20	N,R	128-bit	0000000100**0E**11e1ac360002a5d5c51b
20	N,R	128-bit	0000000200**0E**11e1ac360002a5d5c51b

This byte differentiates the **Service**

Len	PROPERTIES	UUID TYPE	UUID VALUE
20	N, Ww/oR, R,W	128-bit	0000000200**0F**11e1ac360002a5d5c51b

BLE_SensorDemo ATT DB

```
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)
```



→ HWServW2ST**Handle** = 0x000C

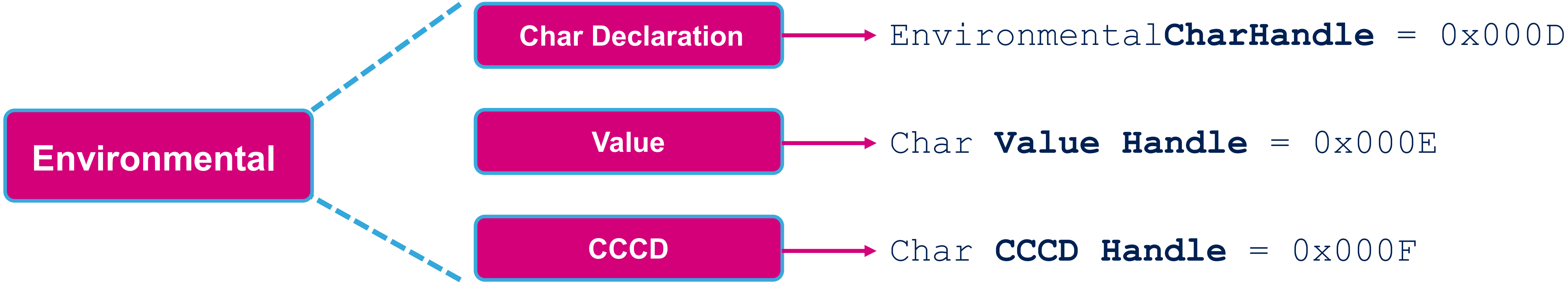
→ Environmental**CharHandle** = 0x000D

→ AccGyroMag**CharHandle** = 0x0010

→ ConfigW2ST**Handle** = 0x002B

→ Config**CharHandle** = 0x002C

BLE_SensorDemo ATT DB



Environmental

Char Declaration → Environmental**CharHandle** = 0x000D

Value → Char **Value Handle** = 0x000E

CCCD → Char **CCCD Handle** = 0x000F

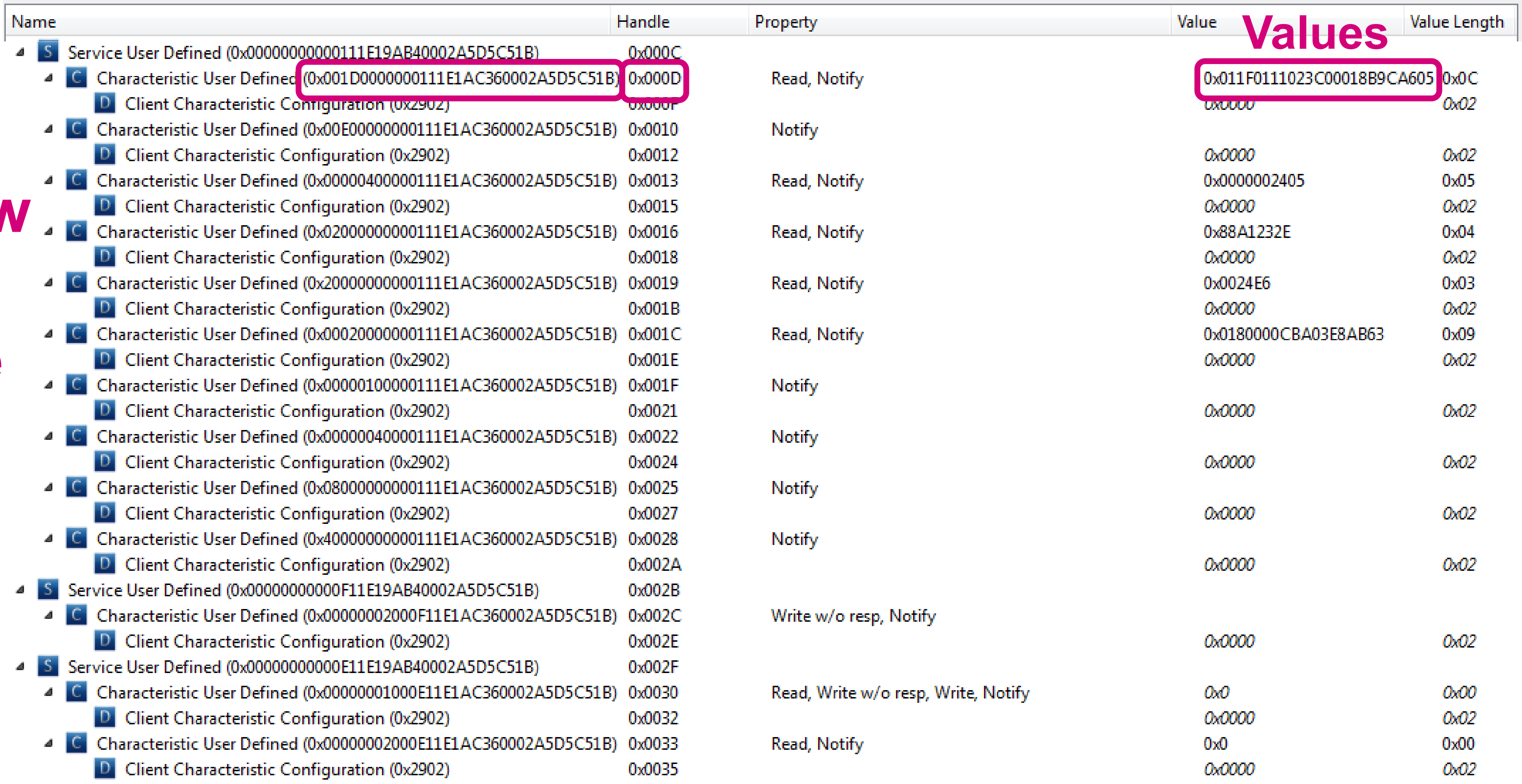
- **Declaration Handle**: used by the application to access the Characteristic
- **Characteristic Value Handle**: used by the Client for Read/Write operations
- **Client Characteristic Configurator Descriptor (CCCD)**: a GATT descriptor is added by default by the stack, if **char** has **Notify/Indicate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

Central: Services and Characteristics

Discovery Procedure

UUIDs

Handles



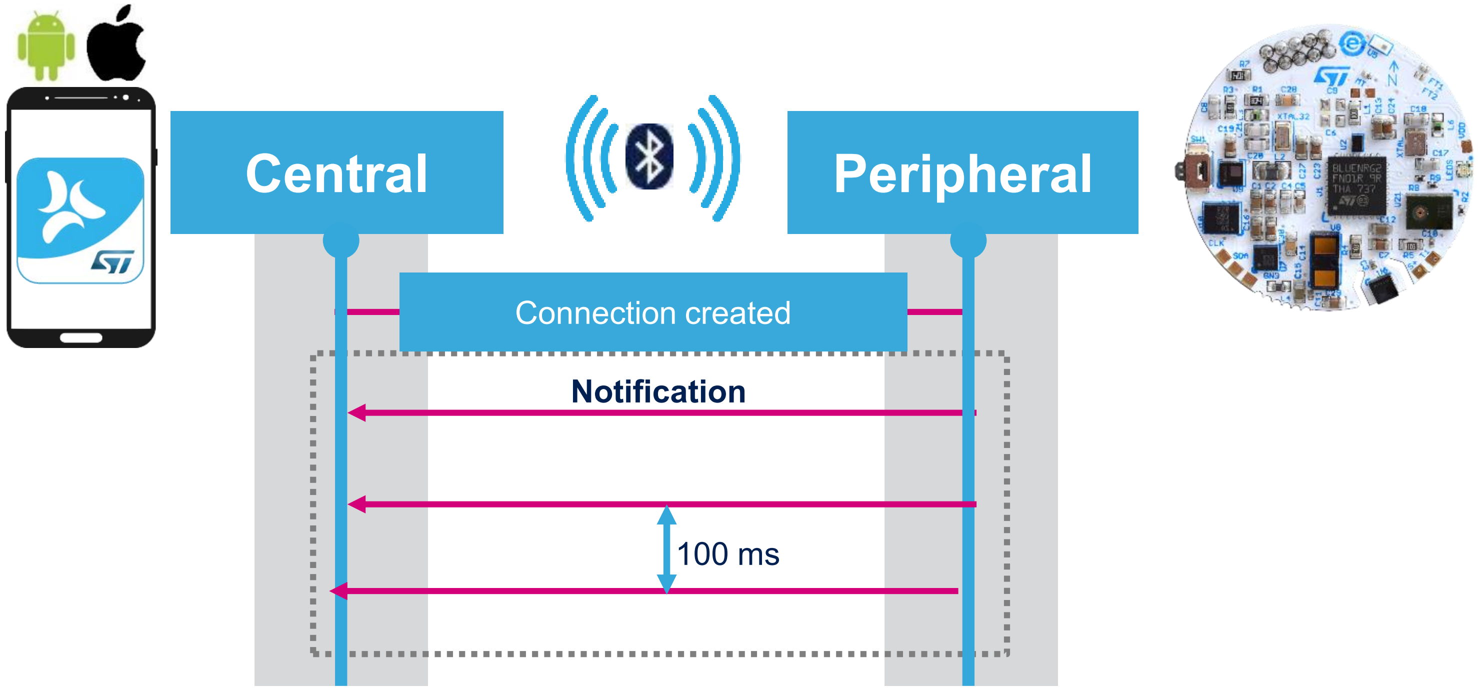
Values

Name	Handle	Property	Value	Value Length
S Service User Defined (0x00000000000111E19AB40002A5D5C51B)	0x000C			
C Characteristic User Defined (0x001D0000000111E1AC360002A5D5C51B)	0x000D	Read, Notify	0x011F0111023C00018B9CA605	0x0C
D Client Characteristic Configuration (0x2902)	0x000F		*0x0000*	*0x02*
C Characteristic User Defined (0x00E00000000111E1AC360002A5D5C51B)	0x0010	Notify		
D Client Characteristic Configuration (0x2902)	0x0012		*0x0000*	*0x02*
C Characteristic User Defined (0x00000400000111E1AC360002A5D5C51B)	0x0013	Read, Notify	0x0000002405	0x05
D Client Characteristic Configuration (0x2902)	0x0015		*0x0000*	*0x02*
C Characteristic User Defined (0x02000000000111E1AC360002A5D5C51B)	0x0016	Read, Notify	0x88A1232E	0x04
D Client Characteristic Configuration (0x2902)	0x0018		*0x0000*	*0x02*
C Characteristic User Defined (0x20000000000111E1AC360002A5D5C51B)	0x0019	Read, Notify	0x0024E6	0x03
D Client Characteristic Configuration (0x2902)	0x001B		*0x0000*	*0x02*
C Characteristic User Defined (0x00020000000111E1AC360002A5D5C51B)	0x001C	Read, Notify	0x0180000CBA03E8AB63	0x09
D Client Characteristic Configuration (0x2902)	0x001E		*0x0000*	*0x02*
C Characteristic User Defined (0x00000100000111E1AC360002A5D5C51B)	0x001F	Notify		
D Client Characteristic Configuration (0x2902)	0x0021		*0x0000*	*0x02*
C Characteristic User Defined (0x00000040000111E1AC360002A5D5C51B)	0x0022	Notify		
D Client Characteristic Configuration (0x2902)	0x0024		*0x0000*	*0x02*
C Characteristic User Defined (0x08000000000111E1AC360002A5D5C51B)	0x0025	Notify		
D Client Characteristic Configuration (0x2902)	0x0027		*0x0000*	*0x02*
C Characteristic User Defined (0x40000000000111E1AC360002A5D5C51B)	0x0028	Notify		
D Client Characteristic Configuration (0x2902)	0x002A		*0x0000*	*0x02*
S Service User Defined (0x0000000000F11E19AB40002A5D5C51B)	0x002B			
C Characteristic User Defined (0x00000002000F11E1AC360002A5D5C51B)	0x002C	Write w/o resp, Notify		
D Client Characteristic Configuration (0x2902)	0x002E		*0x0000*	*0x02*
S Service User Defined (0x0000000000E11E19AB40002A5D5C51B)	0x002F			
C Characteristic User Defined (0x00000001000E11E1AC360002A5D5C51B)	0x0030	Read, Write w/o resp, Write, Notify	*0x0*	*0x00*
D Client Characteristic Configuration (0x2902)	0x0032		*0x0000*	*0x02*
C Characteristic User Defined (0x00000002000E11E1AC360002A5D5C51B)	0x0033	Read, Notify	0x0	0x00
D Client Characteristic Configuration (0x2902)	0x0035		*0x0000*	*0x02*

This is how ATT DB looks like from the Central

life.augmented

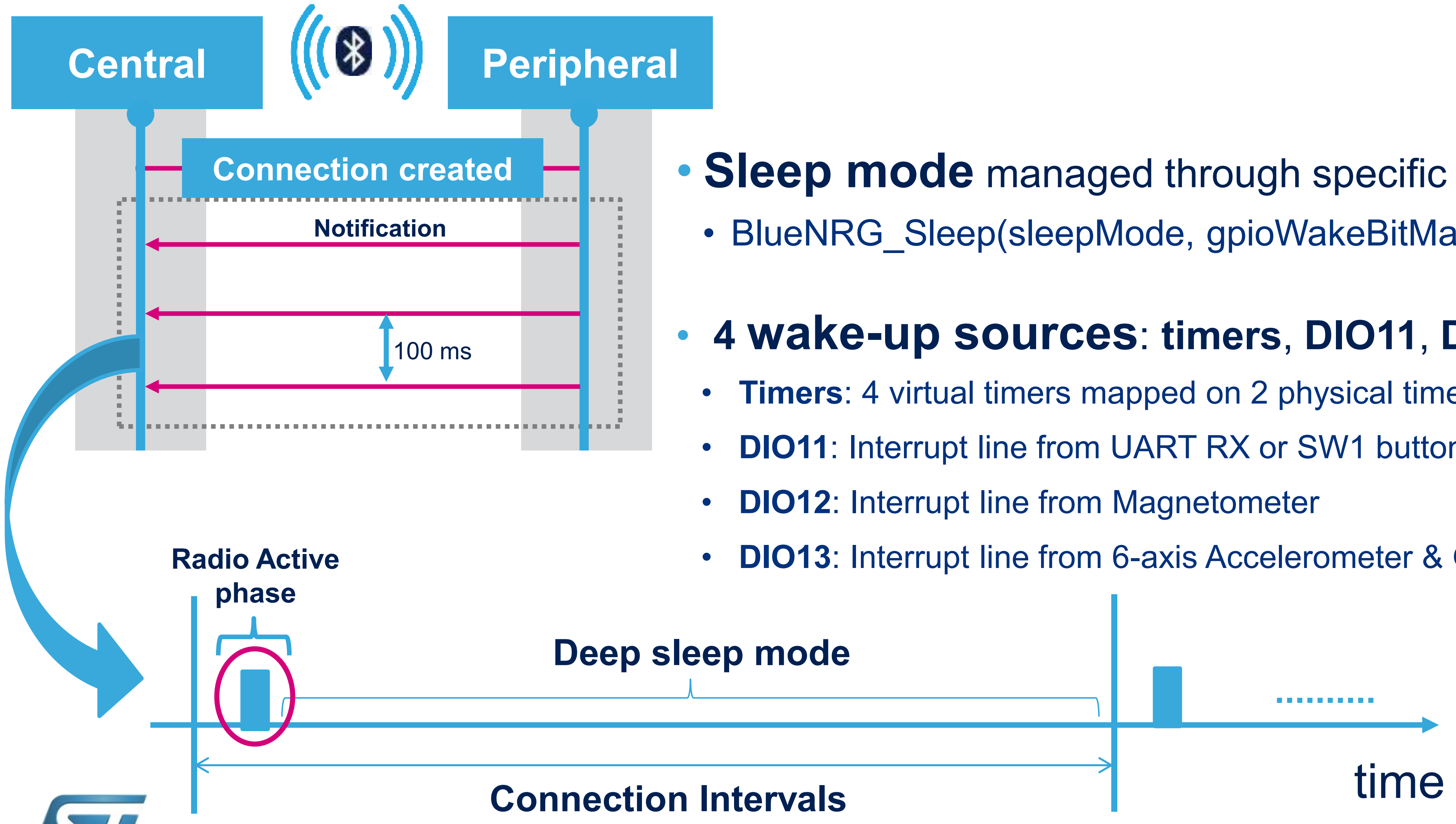
BLE Sensor – Characteristic Update



```
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

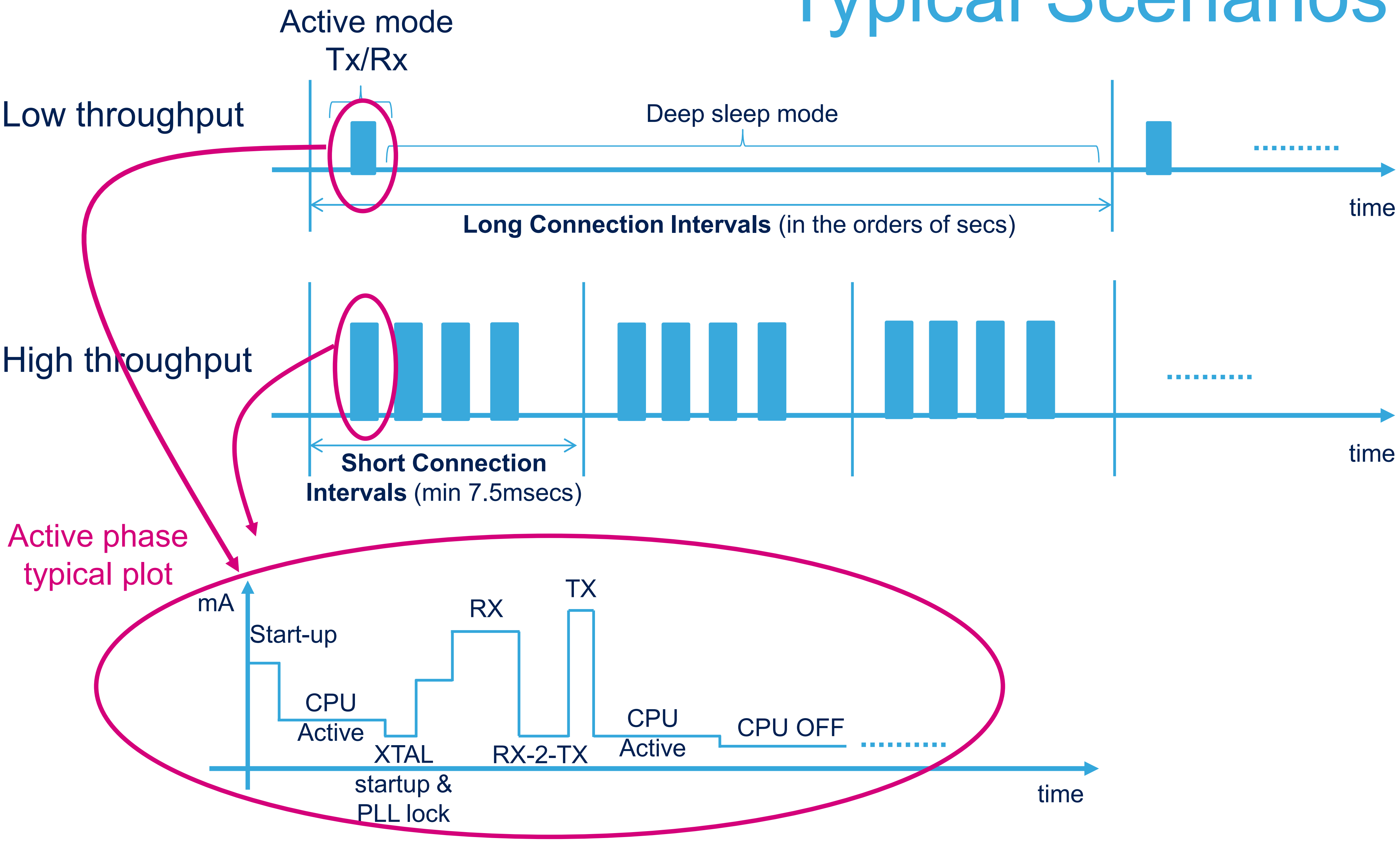
Cortex-M0 Sleep Management



- **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



Low Power Modes

- **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

- BlueNRG-2 Low Power Library **handles autonomously entering and exiting to/from the sleep mode**.

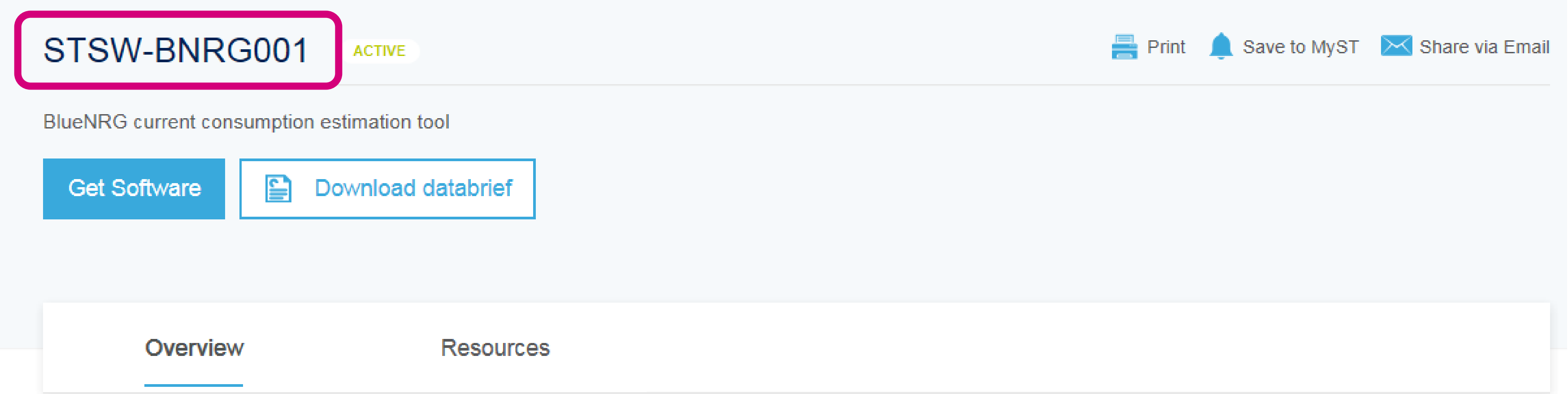
- **The library saves peripherals configurations and application context** before entering deep sleep, and **restores the context** on exiting from low power state:
  - **CPU returns to execute the next instruction after the low power function call**
  - **No need to re-initialize peripheral and radio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

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

BlueNRG
Current Consumption Estimation Tool

- On st.com at this link:

http://www.st.com/content/st_com/en/products/embedded-software/wireless-connectivity-software/stsw-bnrg001.html


STSW-BNRG001

BlueNRG current consumption estimation tool

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.

BlueNRG
Current Consumption Estimation Tool

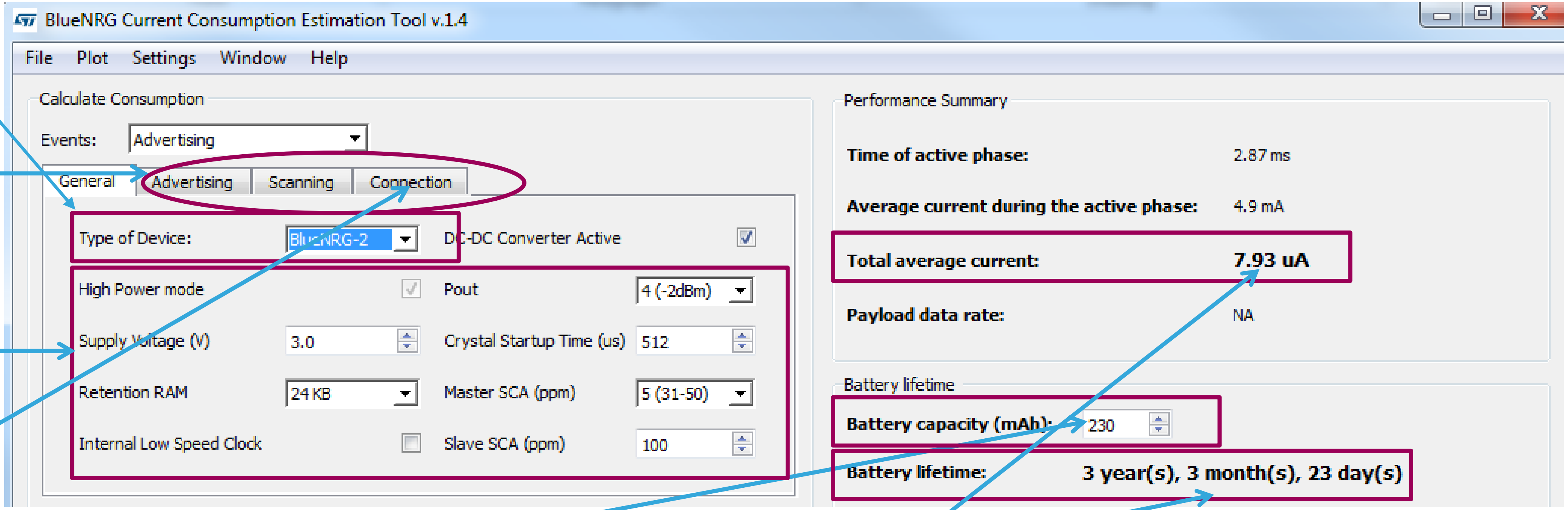
- **Accurate estimate** of average **current consumption** and **battery lifetime**

1. **Select** the device

2. **Select** the type of event

3. **Configure** the HW operating conditions

4. **Configure** the application use case scenario



BlueNRG Current Consumption Estimation Tool v.1.4

File Plot Settings Window Help

Calculate Consumption

Events: Advertising

General | Advertising | Scanning | Connection

Type of Device:	BlueNRG-2	DC-DC Converter Active	☑
High Power mode	☑	Pout	4 (-2dBm)
Supply Voltage (V)	3.0	Crystal Startup Time (us)	512
Retention RAM	24 KB	Master SCA (ppm)	5 (31-50)
Internal Low Speed Clock	□	Slave SCA (ppm)	100

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)**

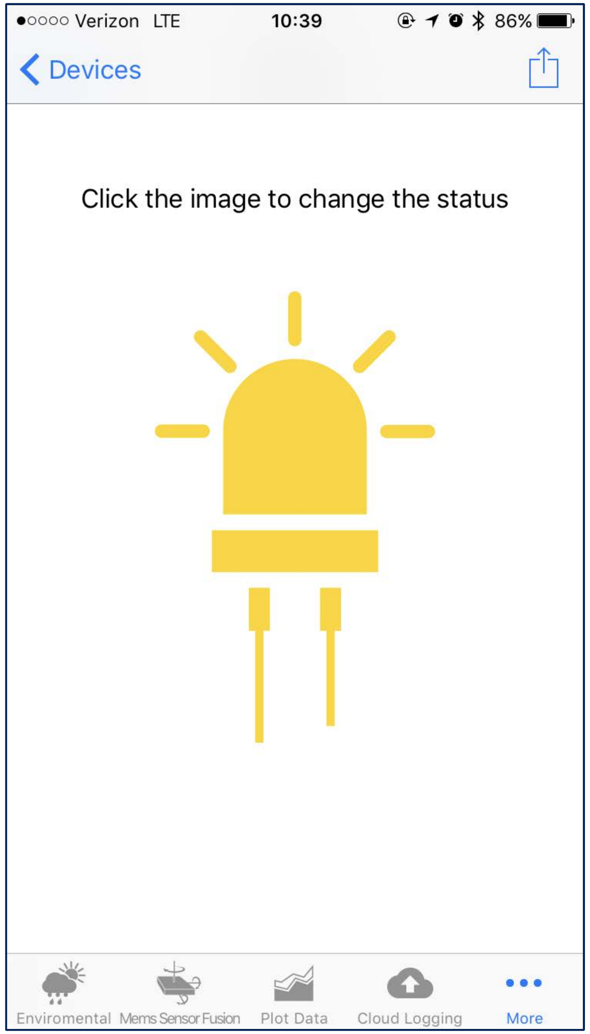
5. **Specify** the Battery capacity

6. **Read** the **average current** and the estimated **battery lifetime**

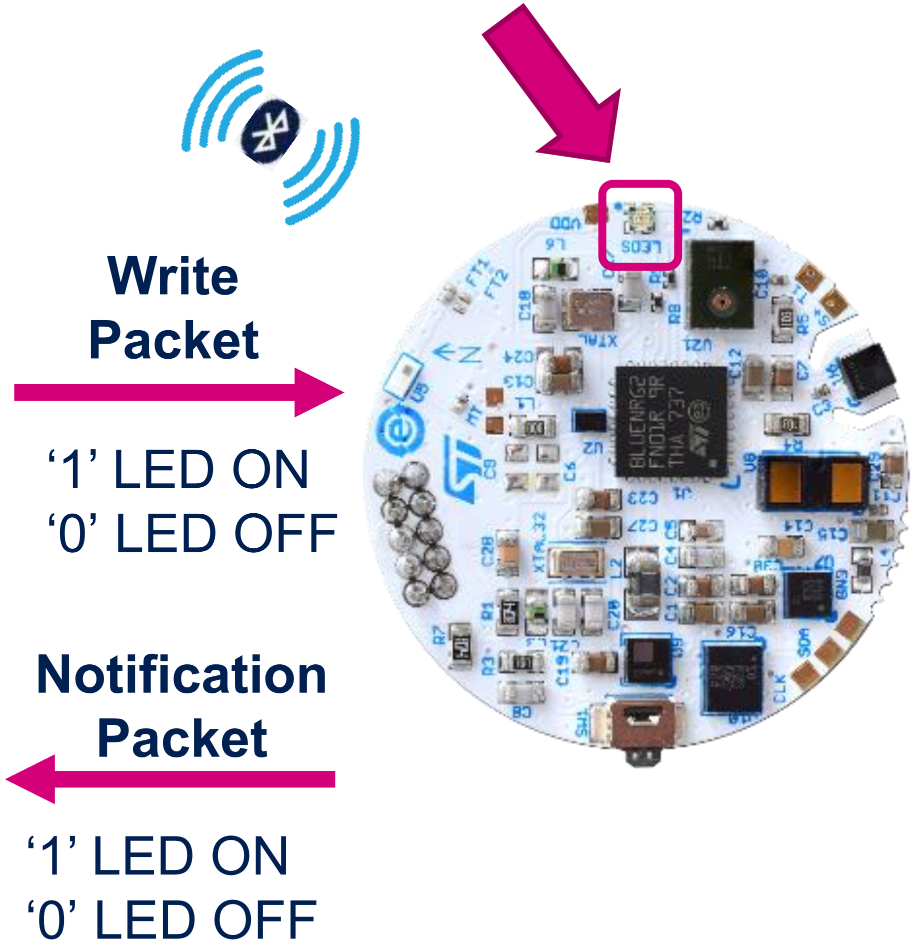
Lab 3

LED Characteristic

Enable LED Toggling



Click the image to change the status



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**

Code Flow

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

ST BlueMS Protocol

4 Bytes

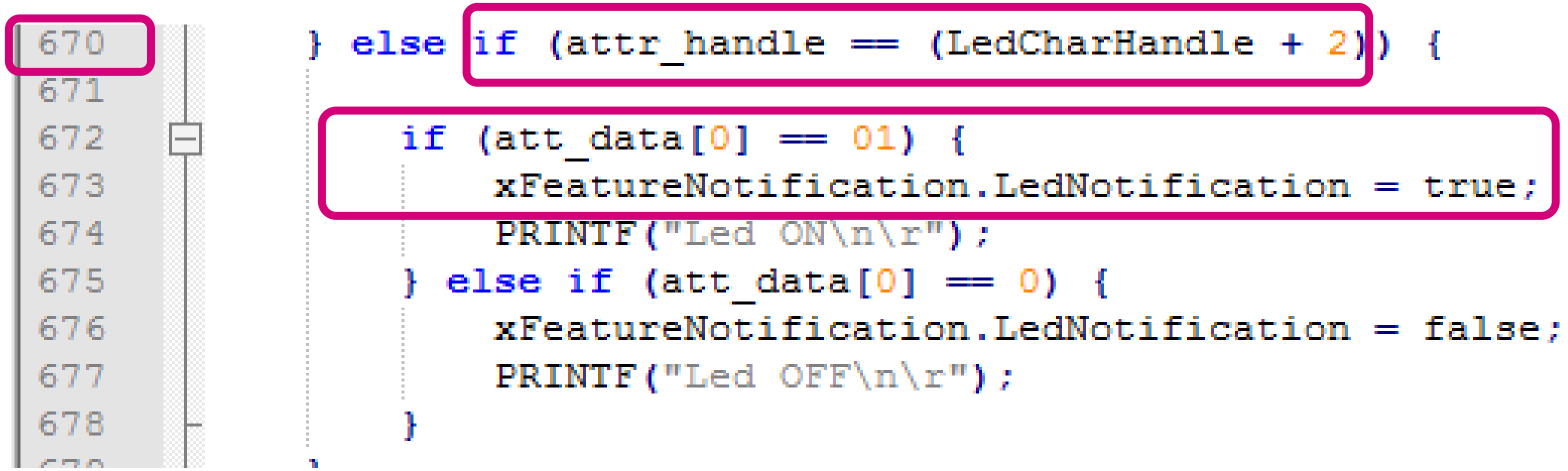
Len	Type	Protocol version	Device ID	Feature Mask	Device MAC	Len	Type	Local Name	Len	Type	Pwr

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

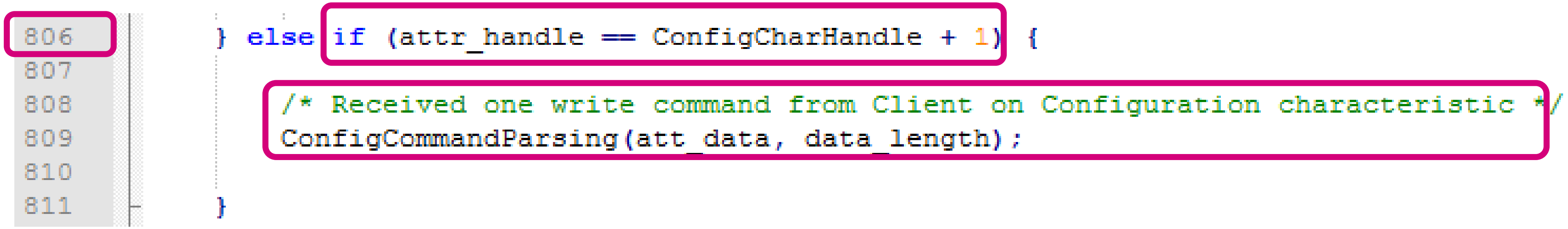
1. In the file **gatt_db.c** **go to** the **line 670**
2. **Client writes** in the **LED Characteristics Client Configurator Descriptor (CCCD)** and **Server enables notifications** through the **xFeatureNotification structure** in **line 673** by **xFeatureNotification.LedNotification = true;**



```
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         }
```

Client Writes in Config Char

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)**

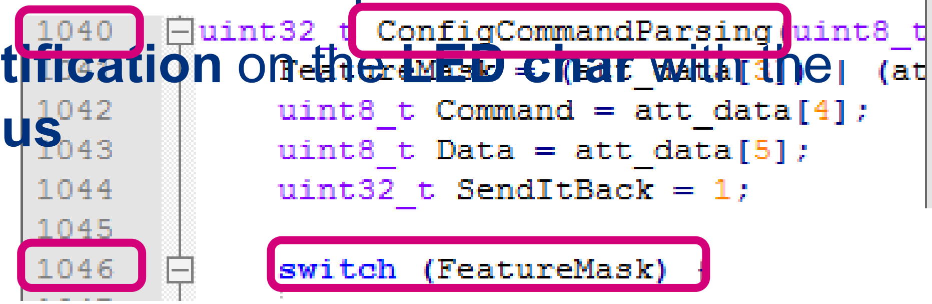


```
806     } else if (attr_handle == ConfigCharHandle + 1) {
807
808         /* Received one write command from Client on Configuration characteristic */
809         ConfigCommandParsing(att_data, data_length);
810
811     }
```

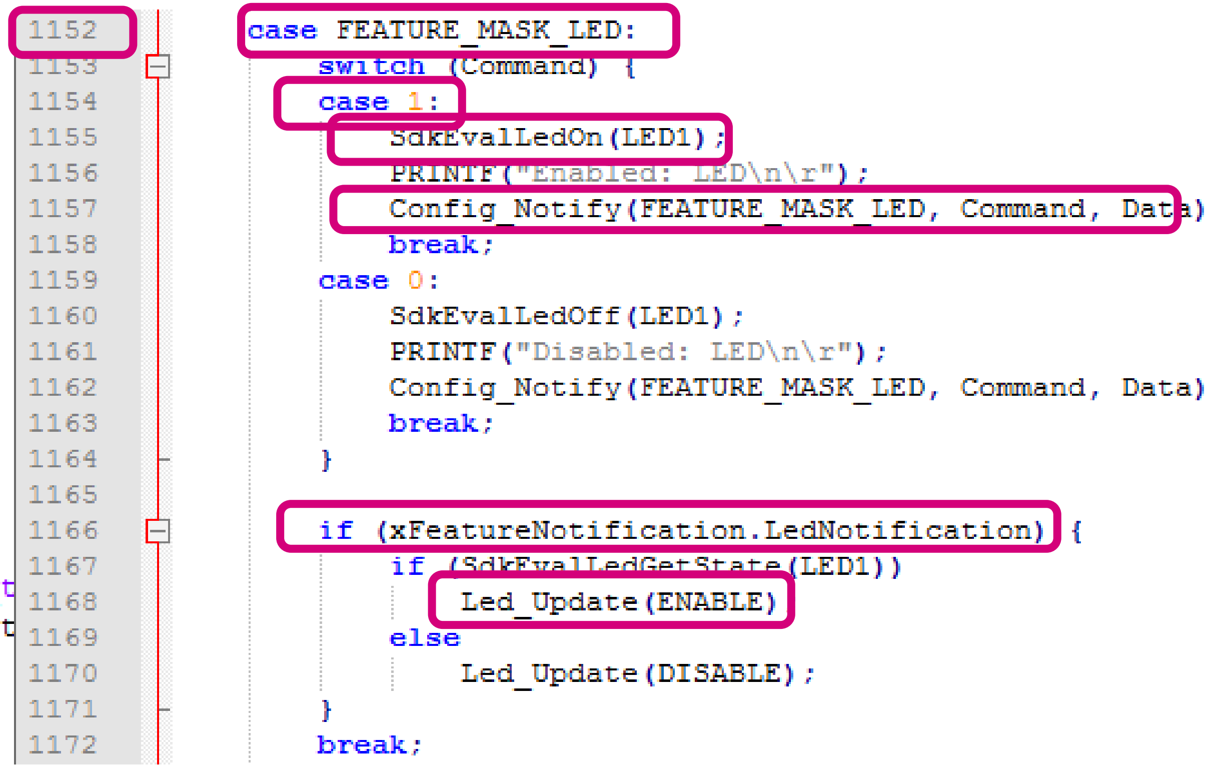
Parse Write Command And

Send BLE Notification on LED Status

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**



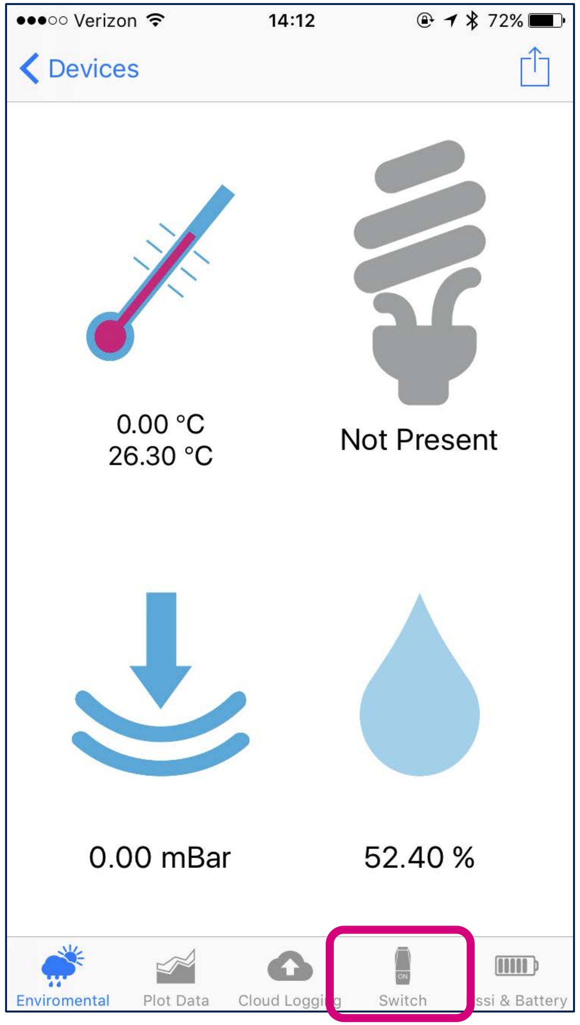
```
1040  uint32_t ConfigCommandParsing(uint8_t
1041      FeatureMask = (att_data[3]) |  (at
1042      uint8_t Command = att_data[4];
1043      uint8_t Data = att_data[5];
1044      uint32_t SendItBack = 1;
1045
1046      switch (FeatureMask) {
```



```
1152  case FEATURE_MASK_LED:
1153      switch (Command) {
1154      case 1:
1155          SdkEvalLedOn(LED1);
1156          PRINTF("Enabled: LED\n\r");
1157          Config_Notify(FEATURE_MASK_LED, Command, Data)
1158          break;
1159      case 0:
1160          SdkEvalLedOff(LED1);
1161          PRINTF("Disabled: LED\n\r");
1162          Config_Notify(FEATURE_MASK_LED, Command, Data)
1163          break;
1164      }
1165
1166      if (xFeatureNotification.LedNotification) {
1167          if (SdkEvalLedGetState(LED1))
1168              Led_Update(ENABLE)
1169          else
1170              Led_Update(DISABLE);
1171      }
1172      break;
```

Enabling LED Toggling

Click on “**Switch**”



●●●○○ Verizon 14:12 72%

Devices

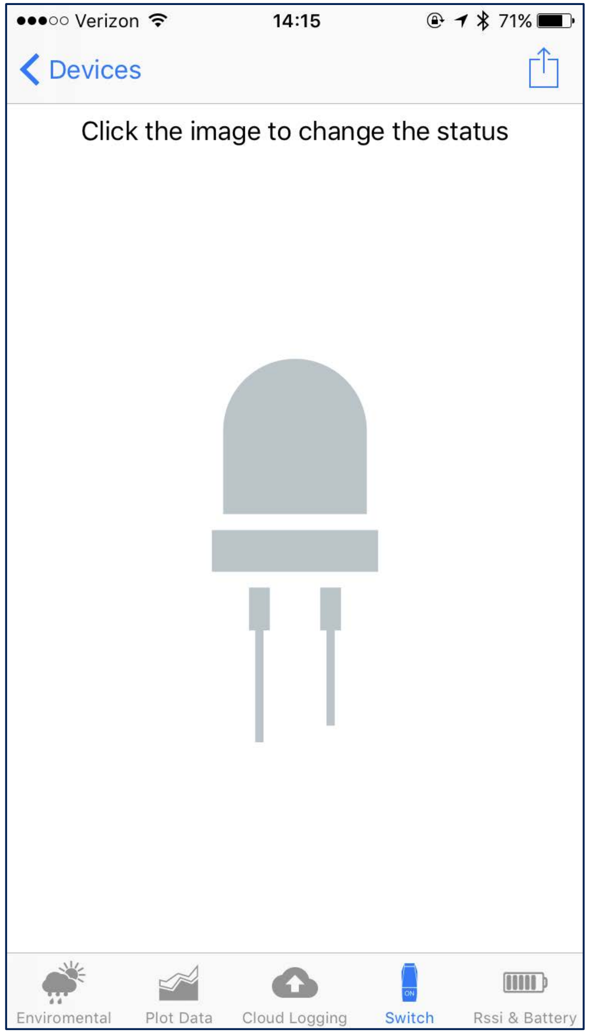
0.00 °C
26.30 °C

Not Present

0.00 mBar

52.40 %

A new page will appear

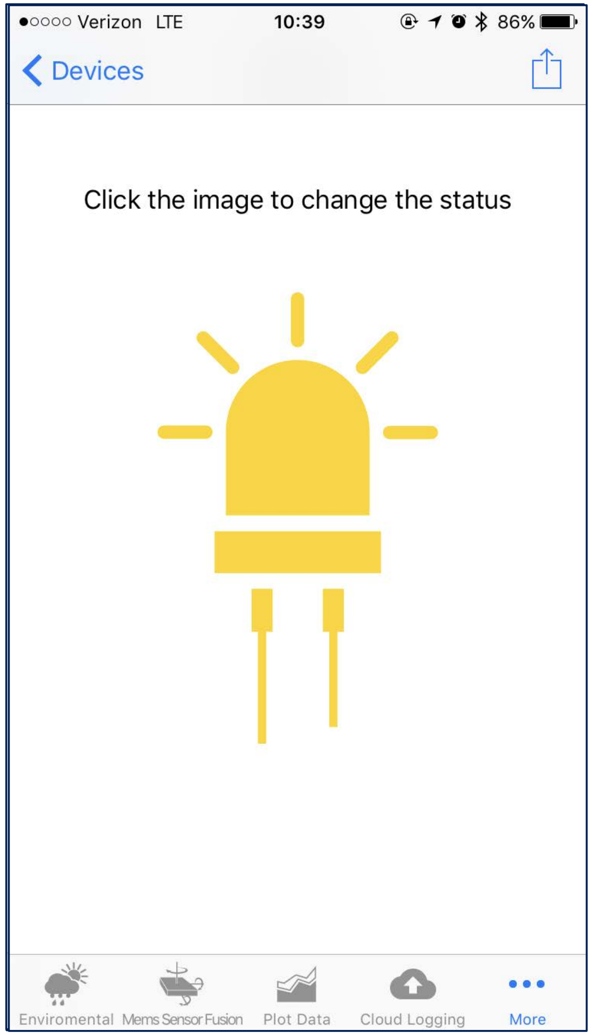


●●●○○ Verizon 14:15 71%

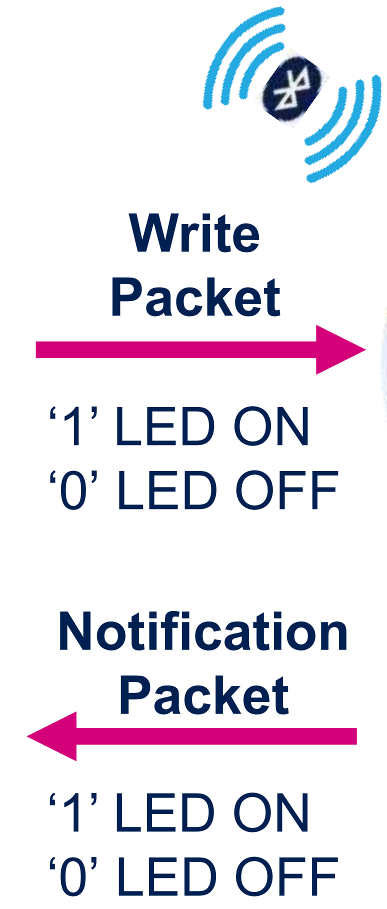
Devices

Click the image to change the status

Enable LED Toggling



Click the image to change the status

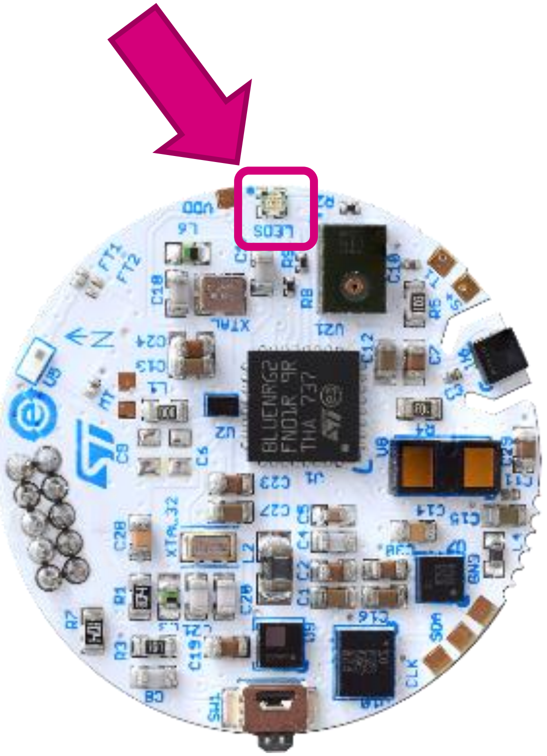


Write Packet

‘1’ LED ON
‘0’ LED OFF

Notification Packet

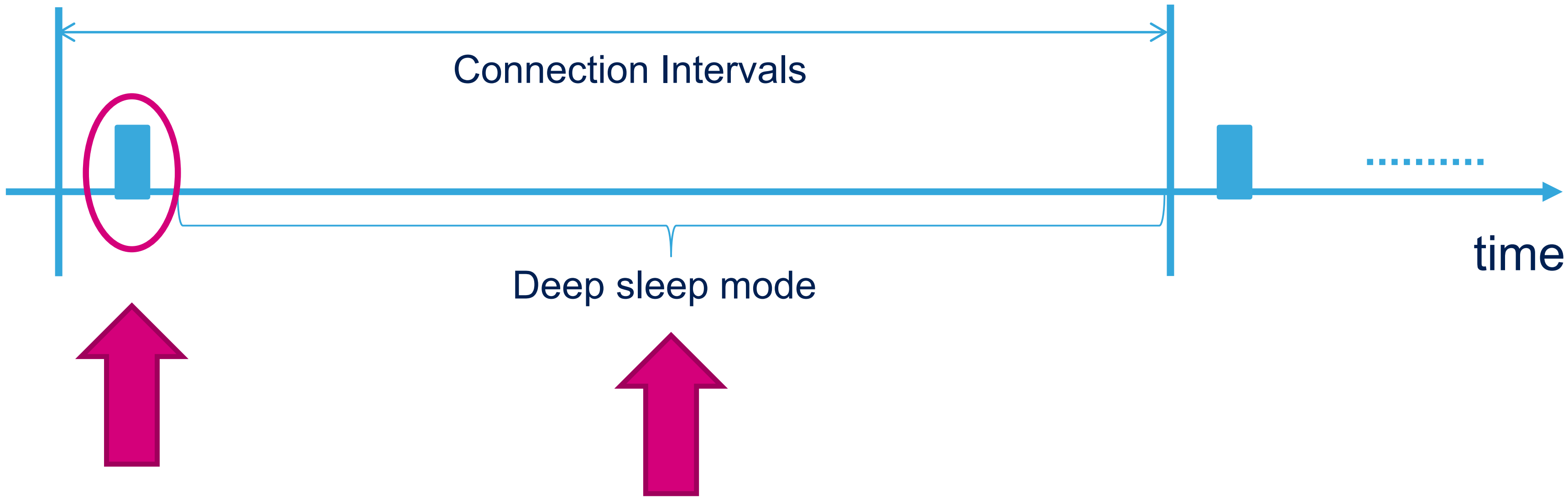
‘1’ LED ON
‘0’ LED OFF



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

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

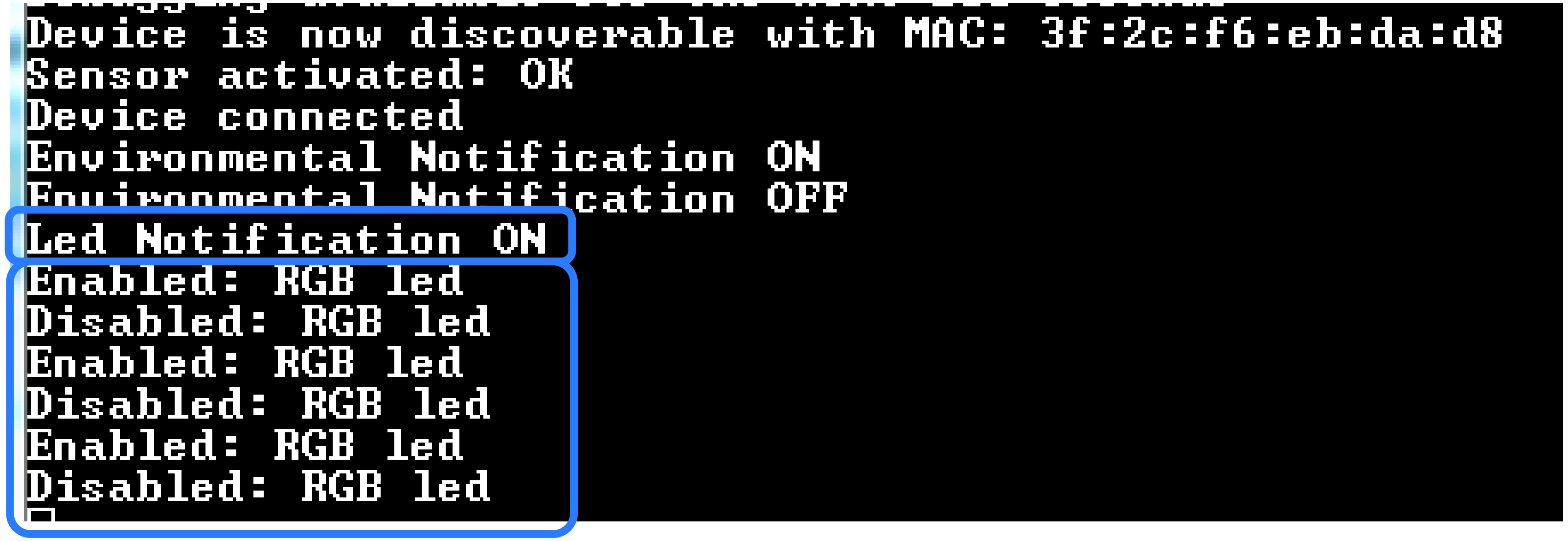


In Active Phase
LED is **ON**

In Sleep Phase
LED is **OFF**

Tera Term Output

- 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
```

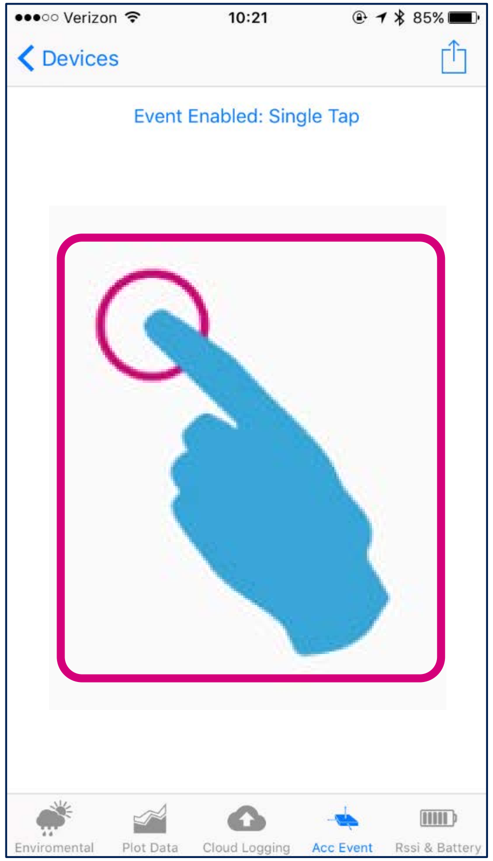
Lab 4

Accelerometer Embedded Events Detection

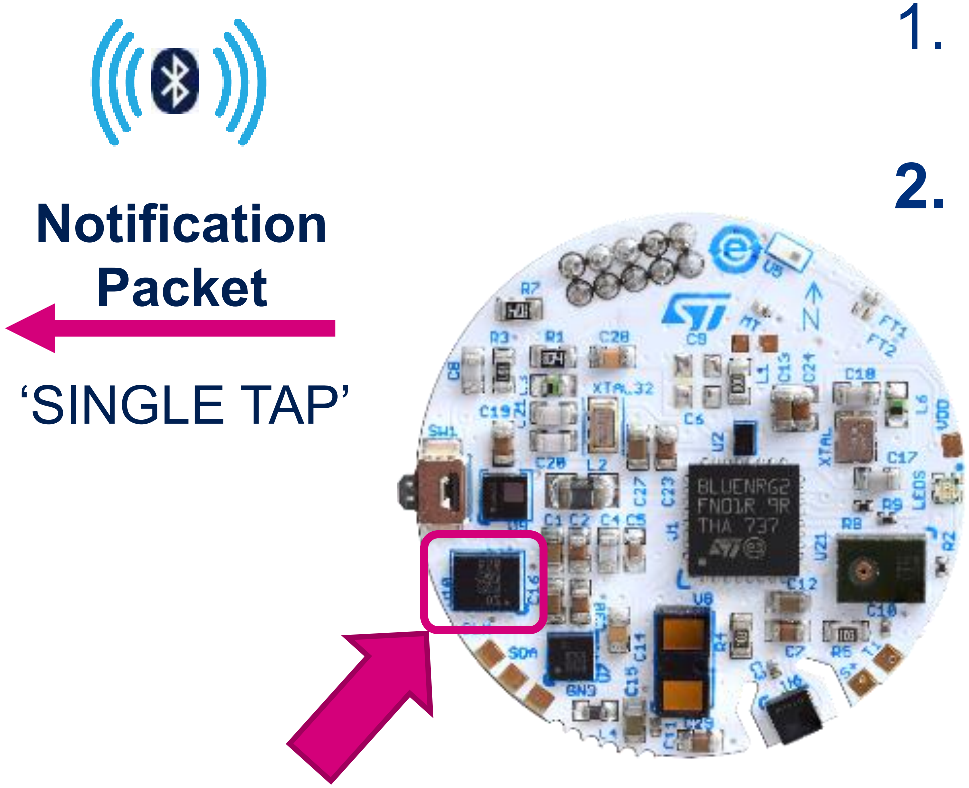
life.augmented

Example - Single Tap

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



Event Enabled: Single Tap



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

ST BlueMS Protocol

4 Bytes

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

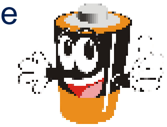
31	30	29	28	27	26	25	24	
RFU	ADPCM	Switch	DoA	ADPC	MicLevel	Proximity	Lux	0x20
23	22	21	20	19	18	17	16	
Acc	Gyro	Mag	Pressure	Humidity	Temperat	Battery	2nd Temp	0x9E
15	14	13	12	11	10	9	8	
RFU	RFU	RFU	RFU	Beamform	AccEvent	FreeFall	SensFusC	0x04
7	6	5	4	3	2	1	0	
SensFus	Compass	MotionInt	Activity	Carry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

LSM6DSO at a Glance

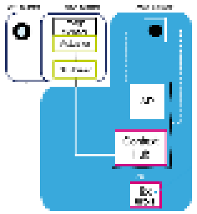
1. PERFORMANCE AND FEATURE

- **Accuracy:** Tango and Daydream compliance
- **0.55mA** current consumption is **HP combo**

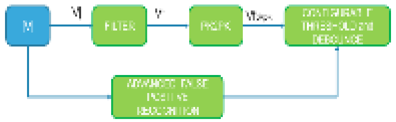
  -15% vs. LSM6DSL/M @ same performance



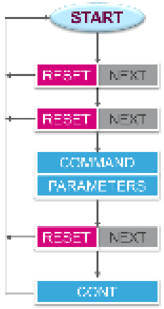
- New **ultra low power mode: 14uA (@100Hz ODR) Axl only**



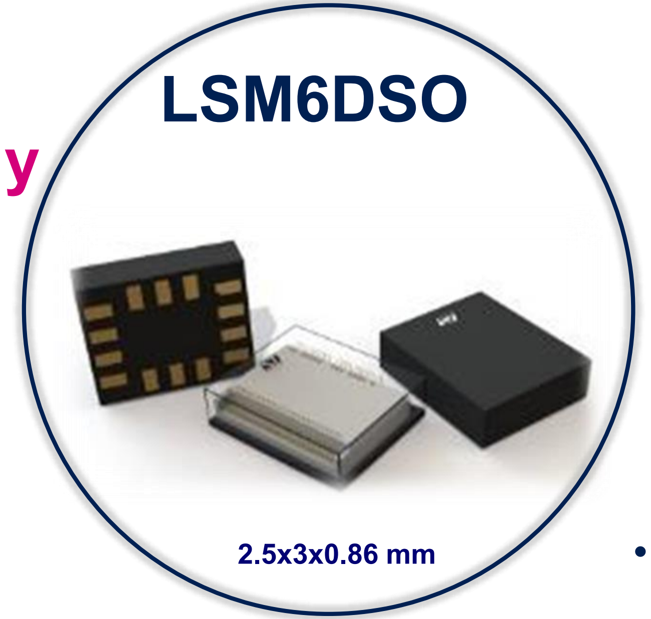
2. FLEXIBILITY: algorithm @ silicon level


M
FILTER
PEAK
CONTINUOUS THRESHOLD AND DEBOUNCE
ADVANCED FALSE POSITIVE RECOGNITION

- **Pedometer 2.x**
  WeChat Precision



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

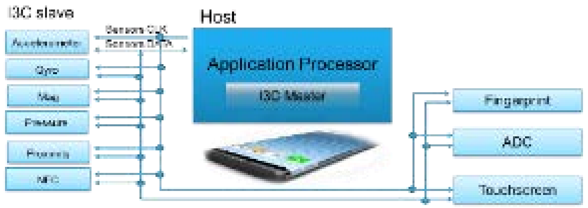


LSM6DSO

2.5x3x0.86 mm

3. NEW STANDARD

- **I3C**



- **FIFO tag**

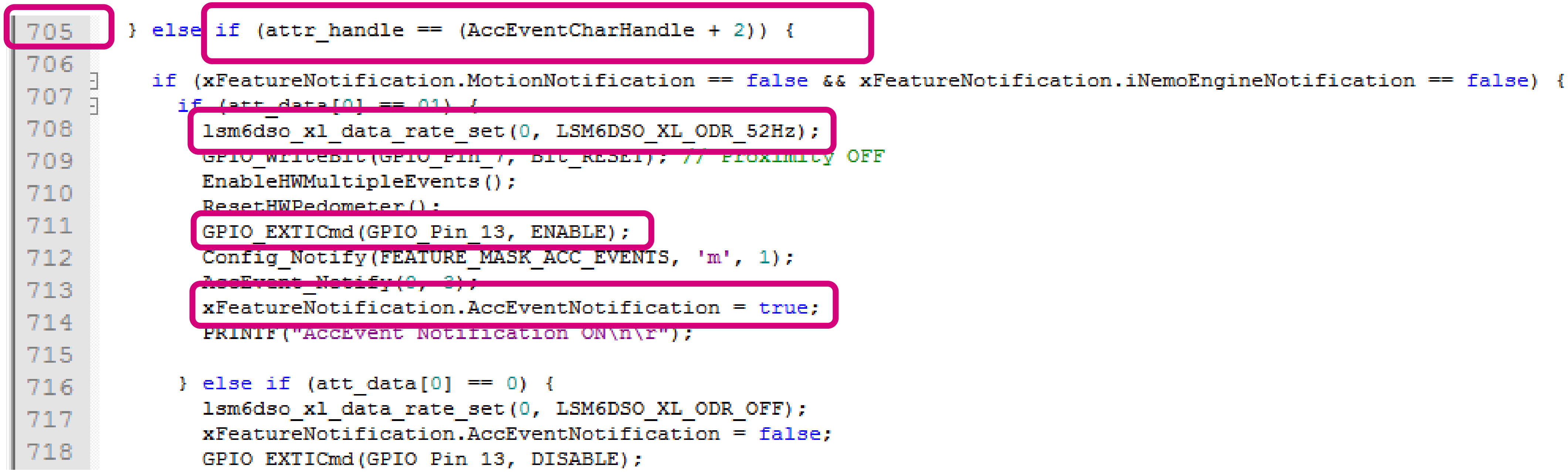
4. INNOVATION

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

Client Enables Acc Event

Characteristic Notifications

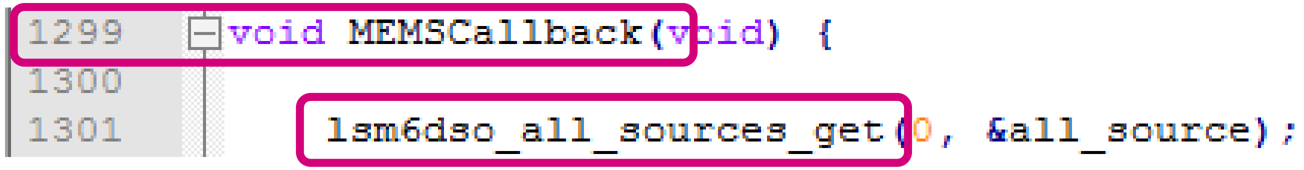
1. In the file **gatt_db.c go to** the **line 703**
2. **Client writes** in the **Acc Event Characteristic Client Configurator Descriptor (CCCD)** and **Server enables notifications** through the **xFeatureNotification structure** in **line 714** by **xFeatureNotification.AccEventNotification = true**



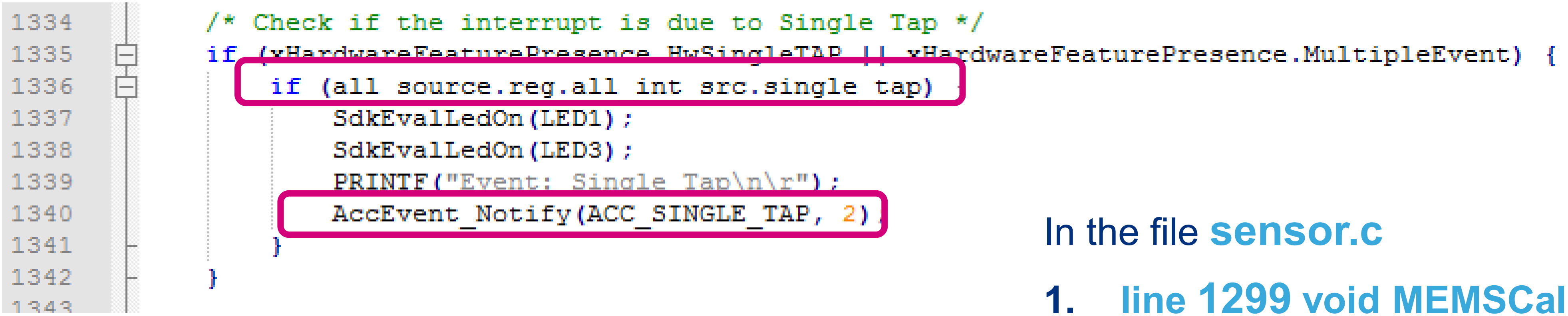
```
} else if (attr_handle == (AccEventCharHandle + 2)) {

  if (xFeatureNotification.MotionNotification == false && xFeatureNotification.iNemoEngineNotification == false) {
    if (att_data[0] == 01) {
      lsm6dso_xl_data_rate_set(0, LSM6DSO_XL_ODR_52Hz);
      GPIO_WriteBit(GPIO_Pin_7, Bit_RESET); // Proximity OFF
      EnableHWMultipleEvents();
      ResetHWPedometer();
      GPIO_EXTICmd(GPIO_Pin_13, ENABLE);
      Config_Notify(FEATURE_MASK_ACC_EVENTS, 'm', 1);
      AccEvent_Notify(0, 2);
      xFeatureNotification.AccEventNotification = true;
      PRINTF("AccEvent Notification ON\n\r");

    } else if (att_data[0] == 0) {
      lsm6dso_xl_data_rate_set(0, LSM6DSO_XL_ODR_OFF);
      xFeatureNotification.AccEventNotification = false;
      GPIO EXTICmd(GPIO Pin 13, DISABLE);
```

life.augmented

Read Event Status and
Send BLE Notification



```
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

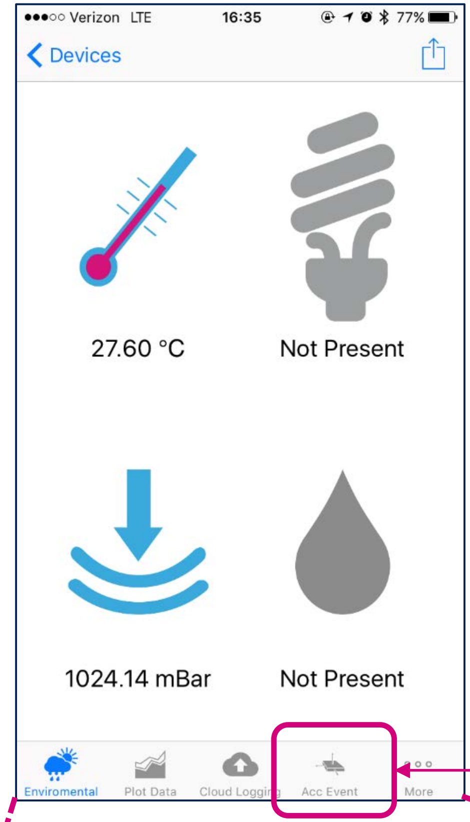
3. **line 1336 if (all_source.reg.all_int_src.single_tap)**

   Check vs. single tap event

4. **line 1340 AccEvent_Notify**

   Send BLE notification

Visualize Single Tap Event
in ST BLE Sensor

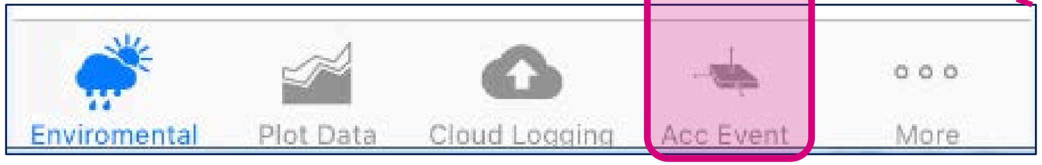


27.60 °C

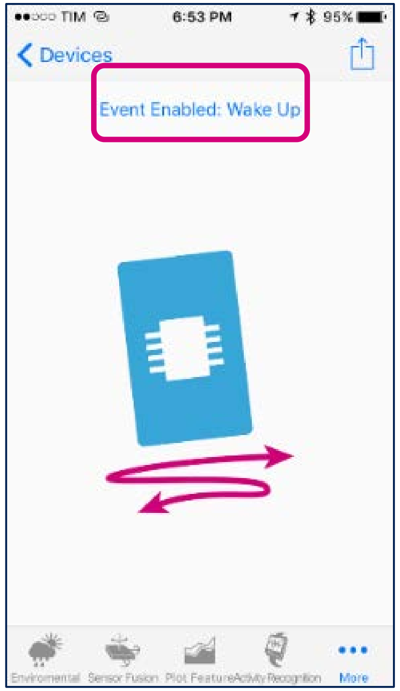
Not Present

1024.14 mBar

Not Present

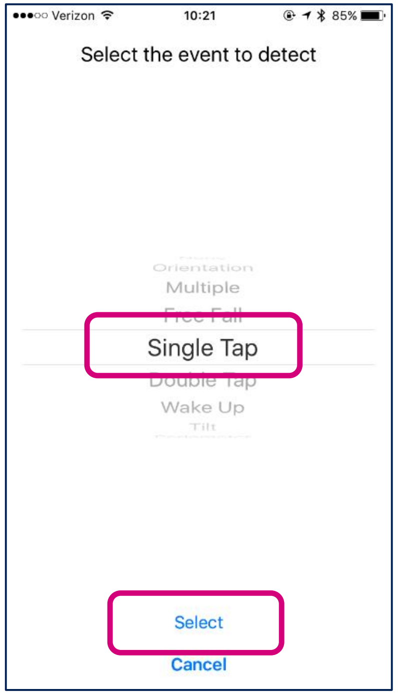


Select
"Acc Event"



Event Enabled: Wake Up

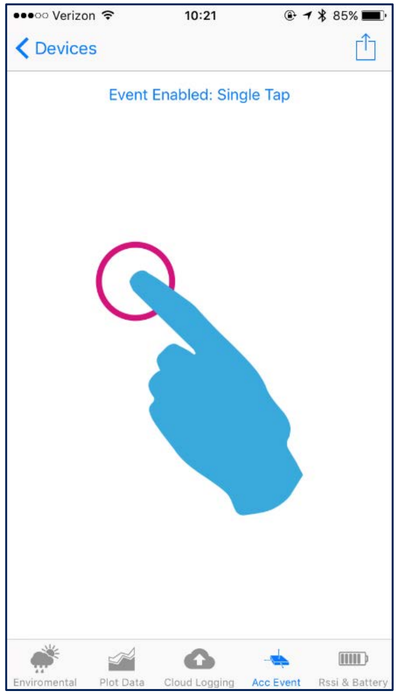
Tap
"Event Enabled"



Select the event to detect

Orientation
Multiple
Free Fall
Single Tap
Double Tap
Wake Up
Tilt

Select
"Single Tap"
and hit "**Select**"

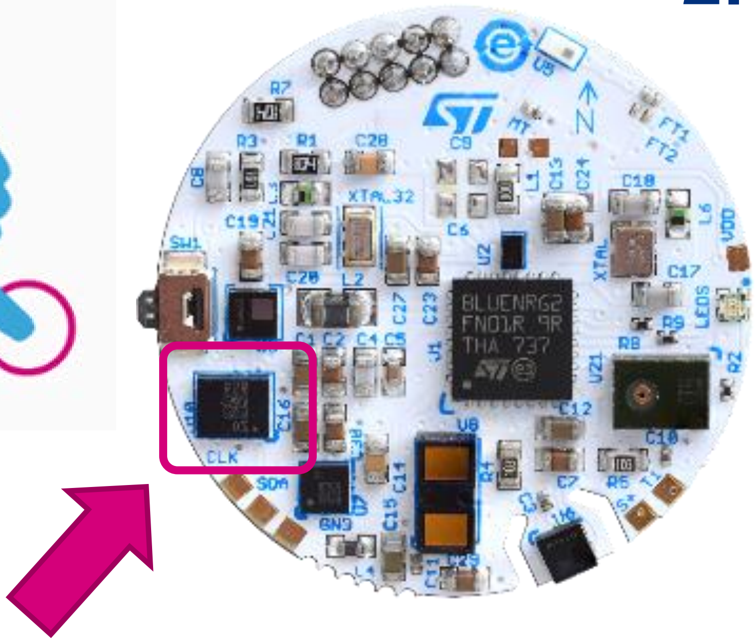


Event Enabled: Single Tap

Enabled
"Single Tap"

Event Detection

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

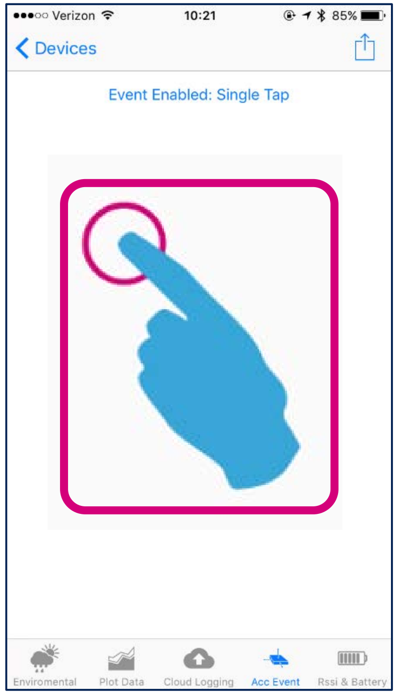


Tap **gently** on the board

Notification Packet

‘SINGLE TAP’

Single Tap

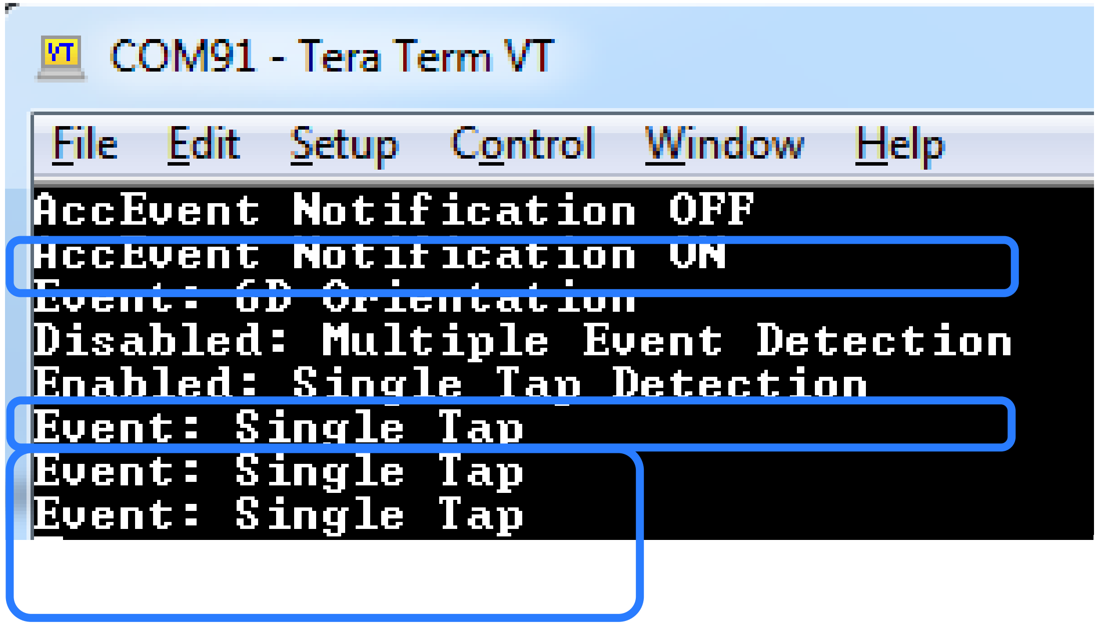


●●●○○ Verizon 10:21 85%

Event Enabled: Single Tap

Tera Term Output

- **“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.

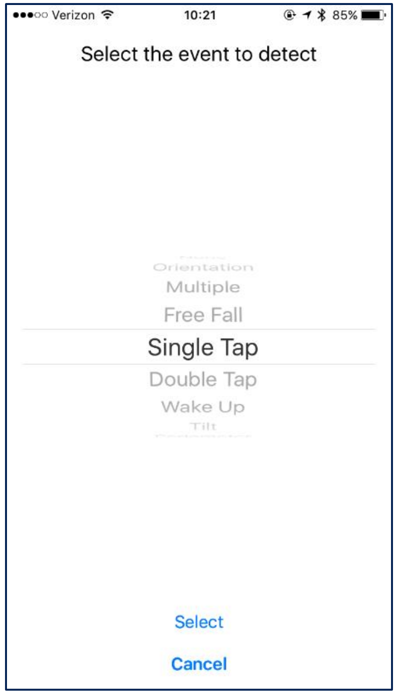


```
AccEvent Notification OFF
AccEvent Notification ON
Event: 6D Orientation
Disabled: Multiple Event Detection
Enabled: Single Tap Detection
Event: Single Tap
Event: Single Tap
Event: Single Tap
```

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

LSM6DSO Embedded Events

Other possible embedded events


●●●○○ Verizon 10:21 85%

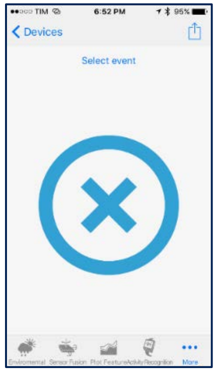
Select the event to detect

Orientation
Multiple
Free Fall
Single Tap
Double Tap
Wake Up
Tilt

Select

Cancel

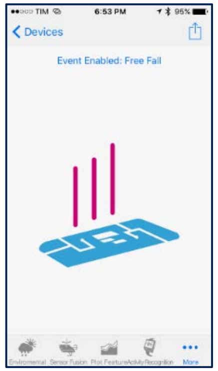
No event


●●○○○ TIM 6:52 PM 95%

Devices

Select event

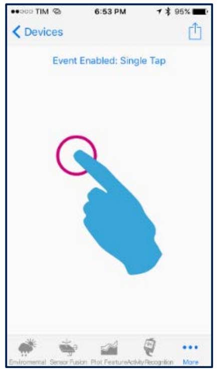
Free fall


●●○○○ TIM 6:53 PM 95%

Devices

Event Enabled: Free Fall

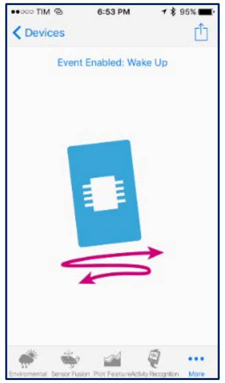
Single tap


●●○○○ TIM 6:53 PM 95%

Devices

Event Enabled: Single Tap

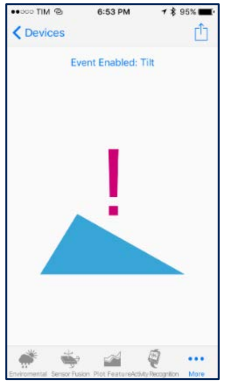
Wake up


●●○○○ TIM 6:53 PM 95%

Devices

Event Enabled: Wake Up

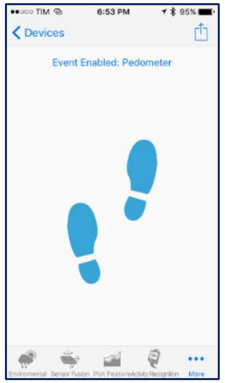
Tilt


●●○○○ TIM 6:53 PM 95%

Devices

Event Enabled: Tilt

Pedometer


●●○○○ TIM 6:53 PM 95%

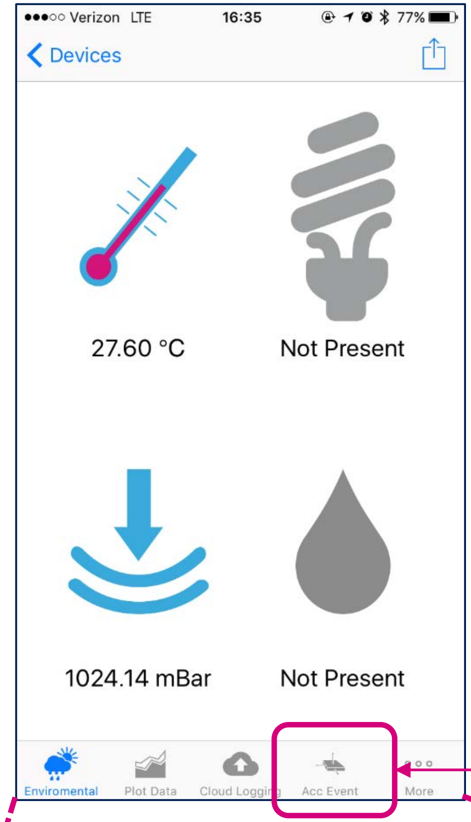
Devices

Event Enabled: Pedometer

WakeUp example

Visualize Hardware Wakeup Event

in BLE Sensor



●●●○○ Verizon LTE 16:35 77%

Devices

27.60 °C

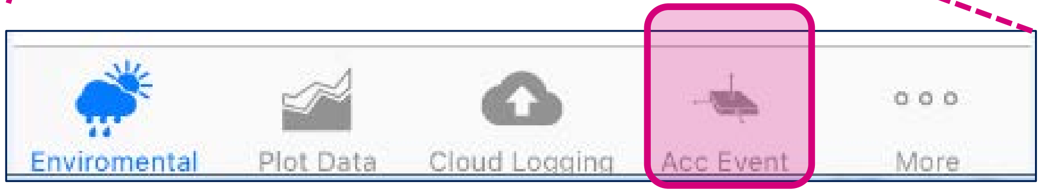
Not Present

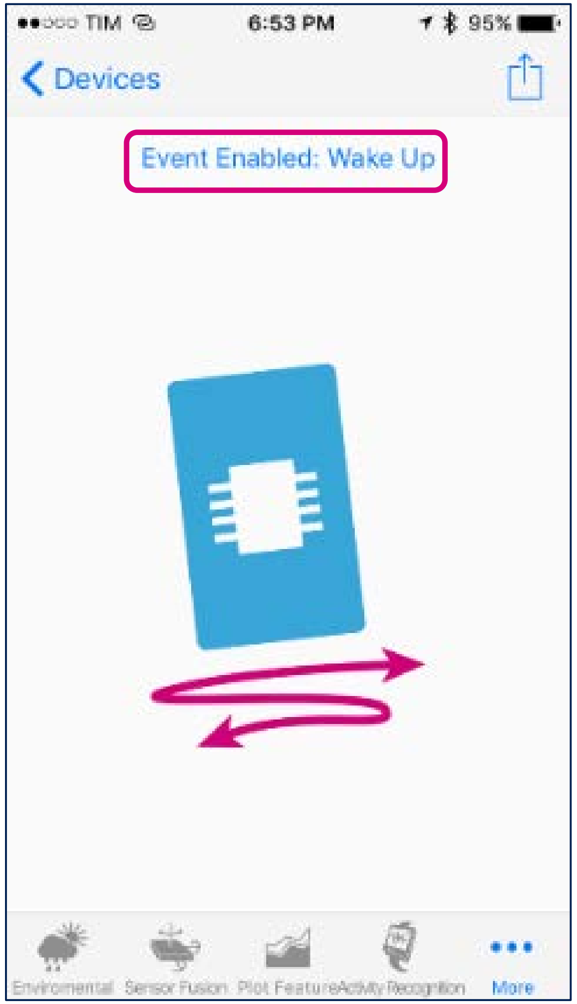
1024.14 mBar

Not Present

Select

“Acc Event”





●●○○○ TIM 6:53 PM 95%

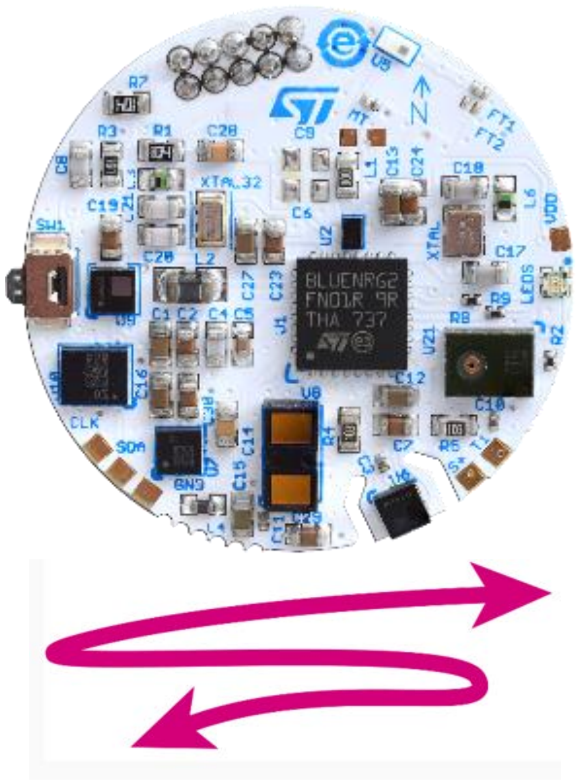
Devices

Event Enabled: Wake Up

life.augmented

Event Detection

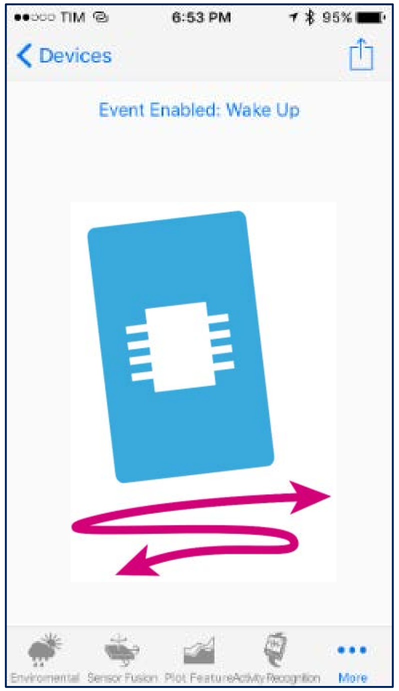
1. **Leave** the board **still** for a few seconds
2. **Shake** the board
3. **Send BLE notification packet**



Notification Packet

‘WAKE UP’

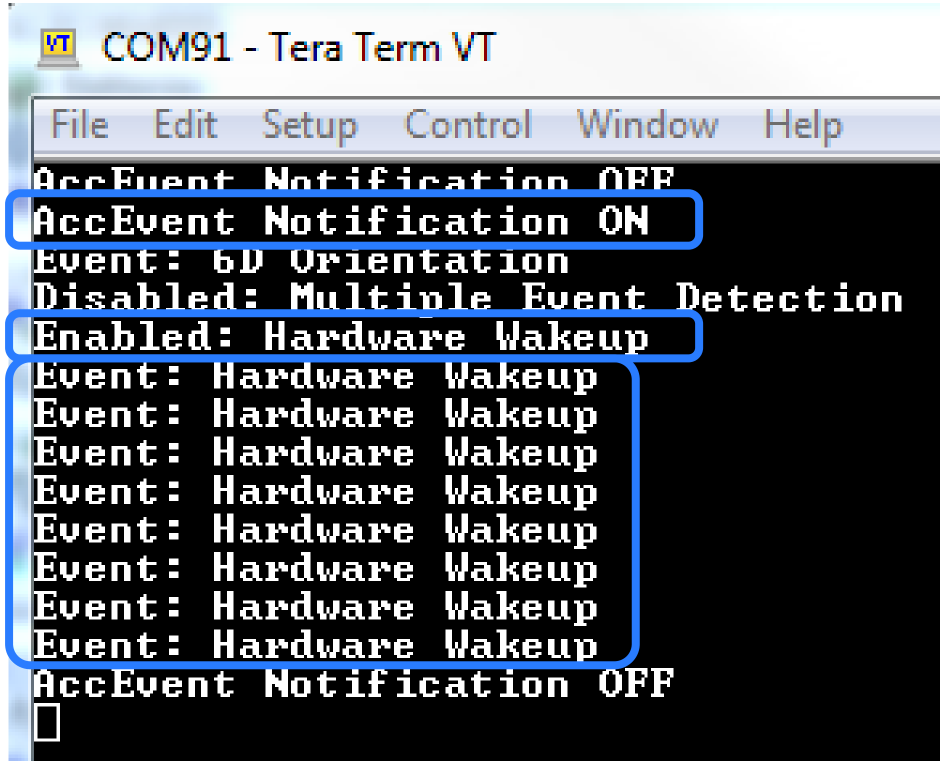
Wake Up



Event Enabled: Wake Up

Tera Term Output

- **“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.



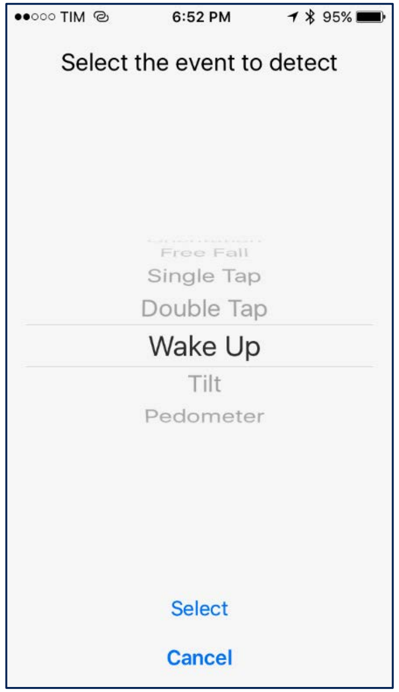
```
AccEvent Notification OFF
AccEvent Notification ON
Event: 6D Orientation
Disabled: Multiple Event Detection
Enabled: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
Event: Hardware Wakeup
Event: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

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

LSM6DSO Embedded Events

You can go ahead later and test other events…

Other possible embedded events


●●○○○ TIM 6:52 PM 95%

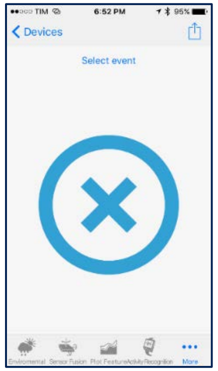
Select the event to detect

Free Fall
Single Tap
Double Tap
Wake Up
Tilt
Pedometer

Select

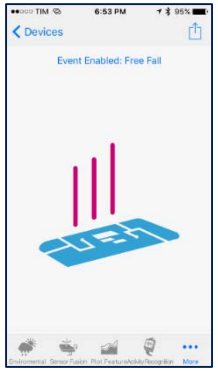
Cancel

No event


Devices

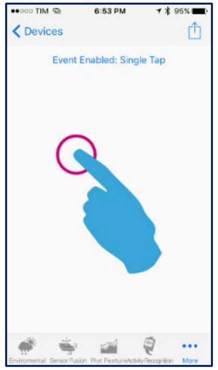
Select event

Free fall


Devices

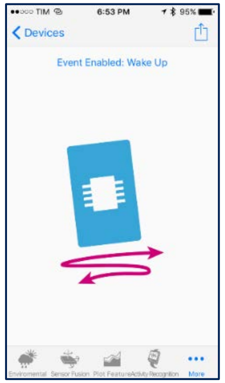
Event Enabled: Free Fall

Single tap


Devices

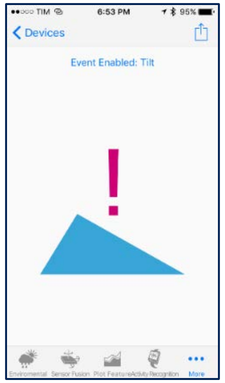
Event Enabled: Single Tap

Wake up


Devices

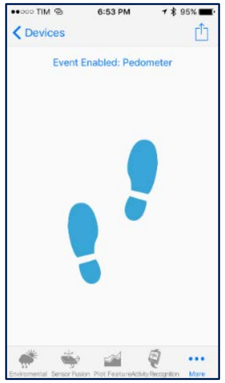
Event Enabled: Wake Up

Tilt


Devices

Event Enabled: Tilt

Pedometer


Devices

Event Enabled: Pedometer

Lab 5

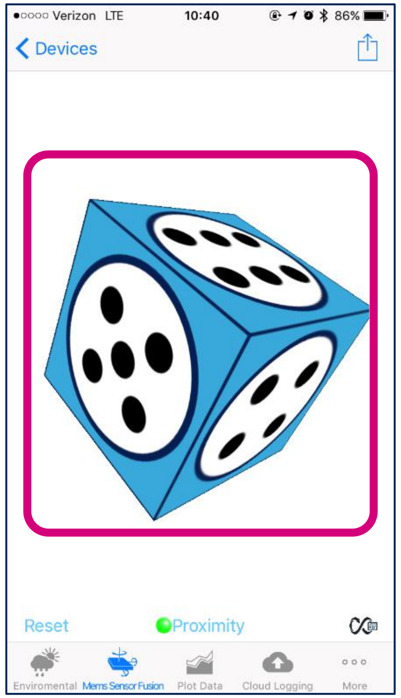
9-axis Acc+Gyro+Mag Sensor Data Fusion

Goal

- 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

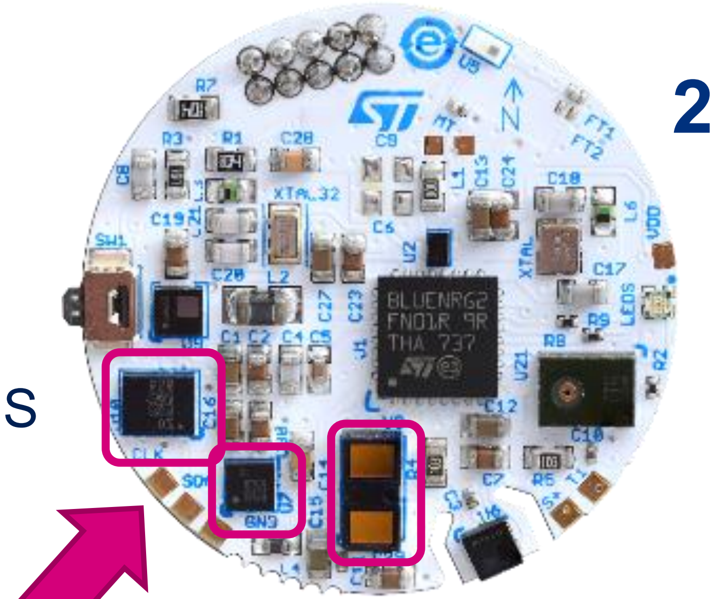
Sensor Data Fusion

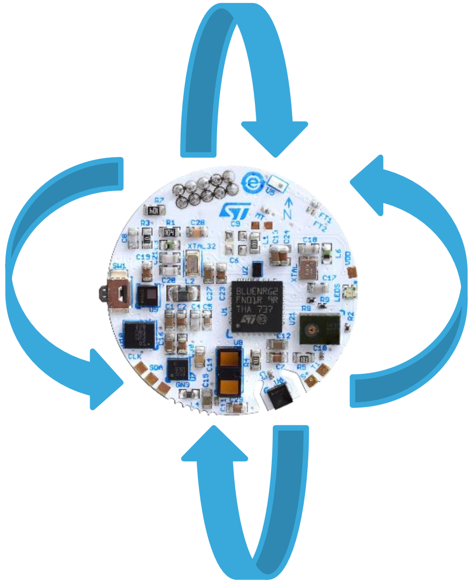
1. **Enable** Sensor Fusion library
2. **Send** quaternions information through **BLE notification packet**



Notification Packet

QUATERNIONS





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

life.augmented

Code Modifications

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

ST BlueMS Protocol

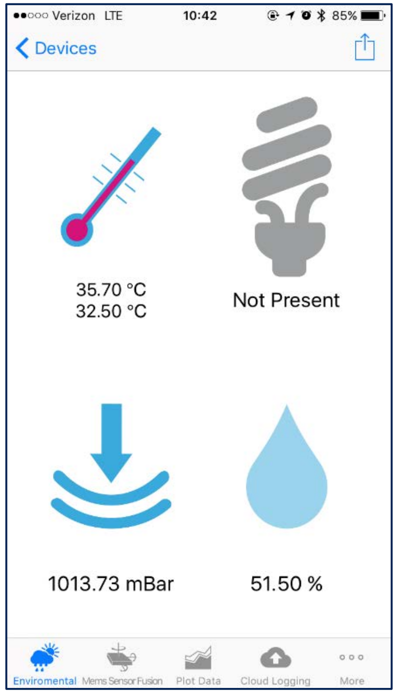
4 Bytes

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	0x22
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	

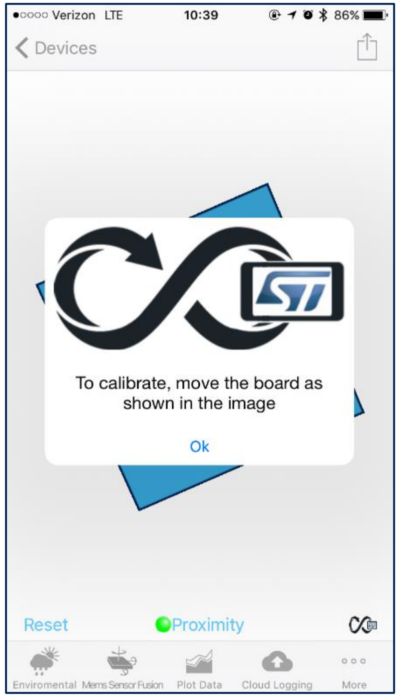
Sensor Data Fusion

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



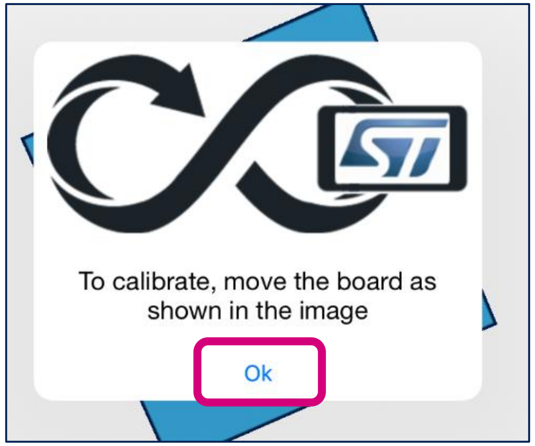
1

“**Mems Sensor Fusion**” tab


To calibrate, move the board as shown in the image

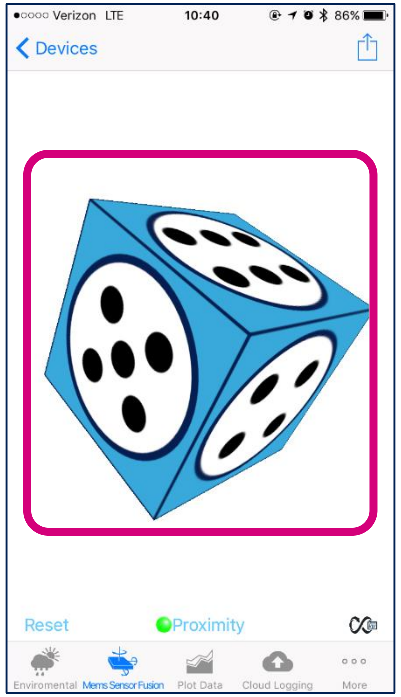
2

Click on **OK**


To calibrate, move the board as shown in the image

3

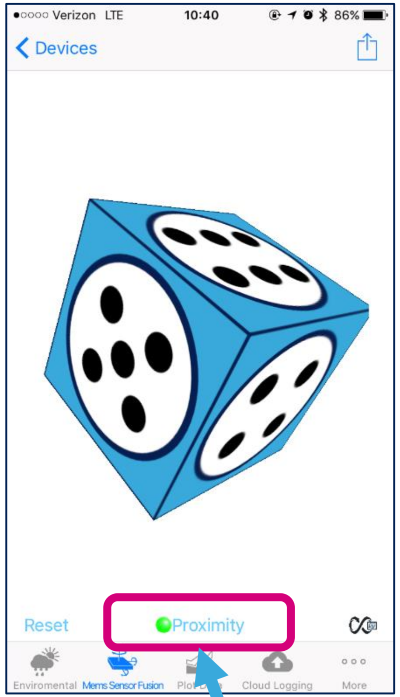
Sensor Fusion **enabled**



life.augmented

Proximity Sensor

Enable Proximity Sensor



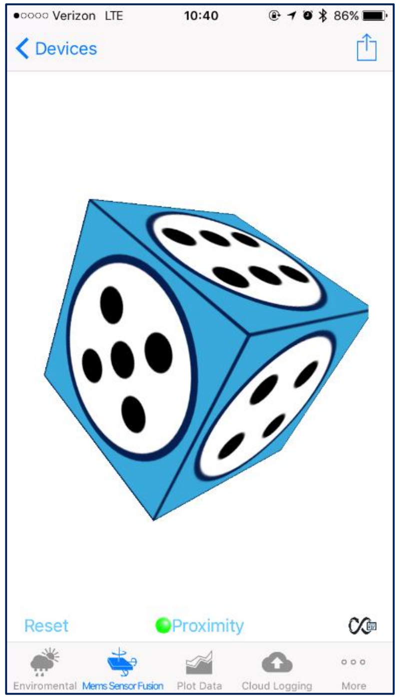
ON 🟢Proximity

OFF 🔴Proximity

Move your hand far and close to the BlueNRG-Tile

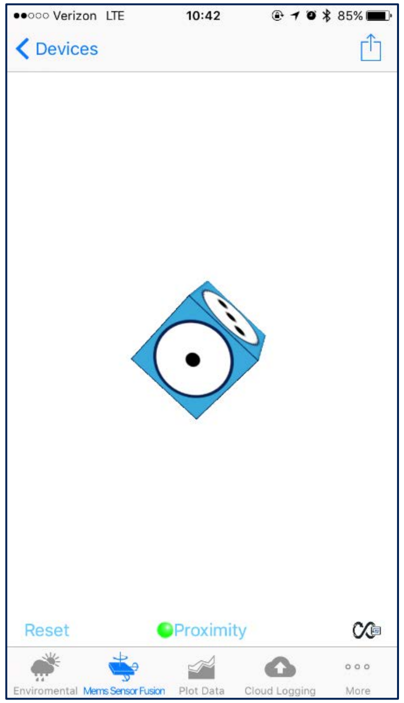
1

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

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:

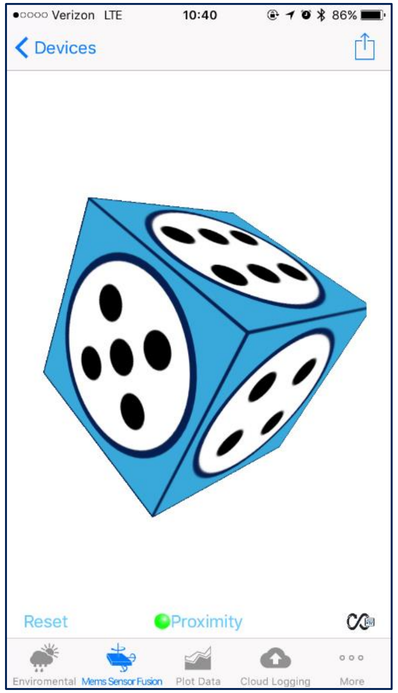
- Robots: Obstacle avoidance
- Vacuum cleaners: Wall following, cliff detection
- Drones: Take-off and landing, Ceiling detection

Accurate objects distance scanning

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

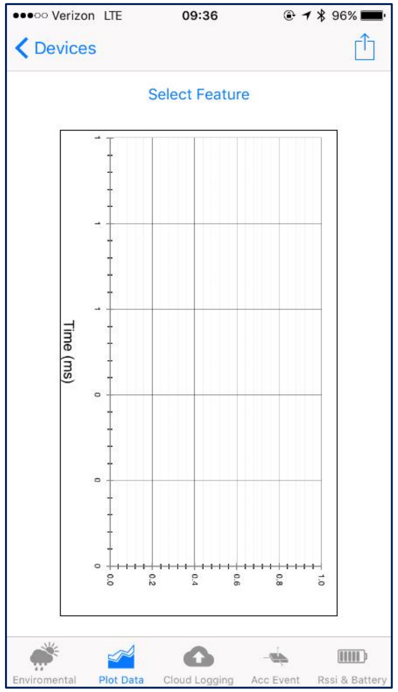
Full-range Proximity Real-time Data Plot

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



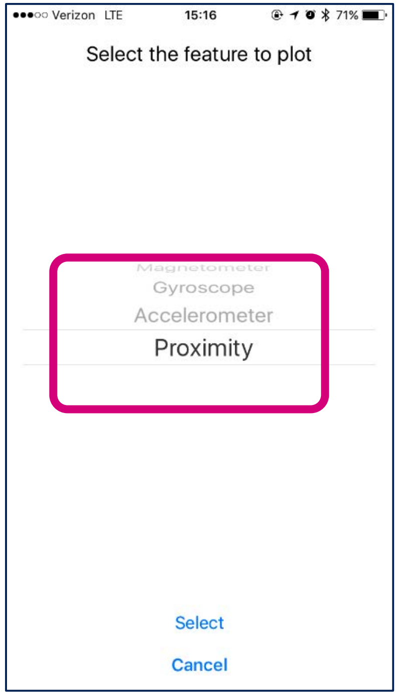
1

“Plot Feature” tab



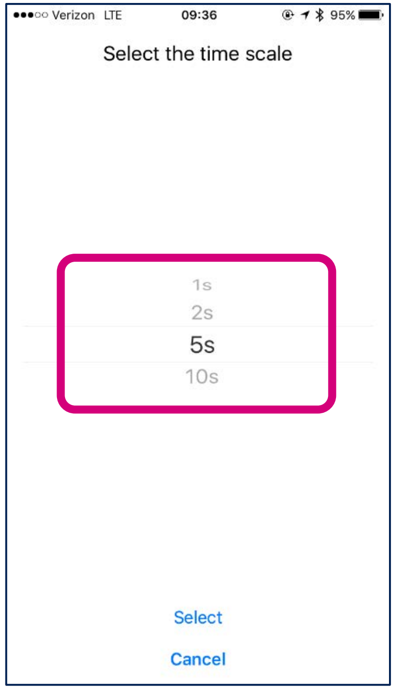
2

Select Proximity



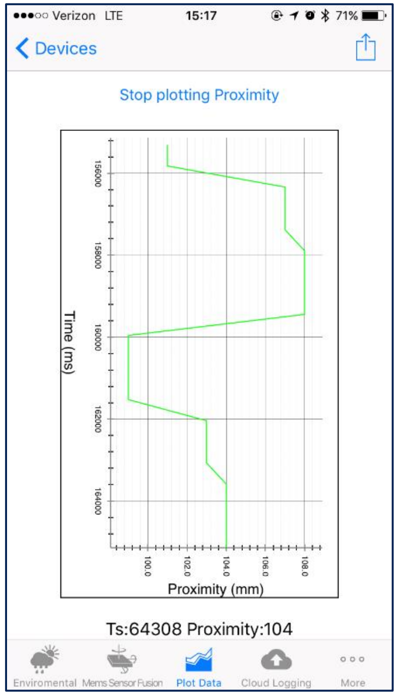
3

Select the **time frame**



4

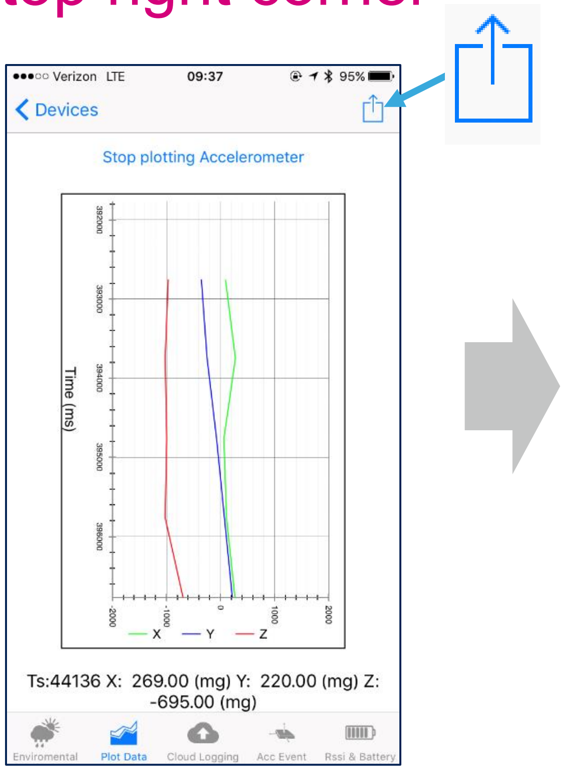
View the **real-time data plot**



Ts:64308 Proximity:104

Logging Real-time Data Plot

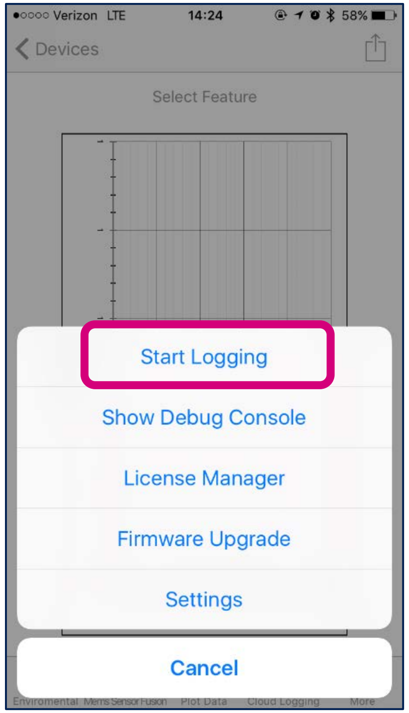
Click on
top right corner



Ts:44136 X: 269.00 (mg) Y: 220.00 (mg) Z: -695.00 (mg)

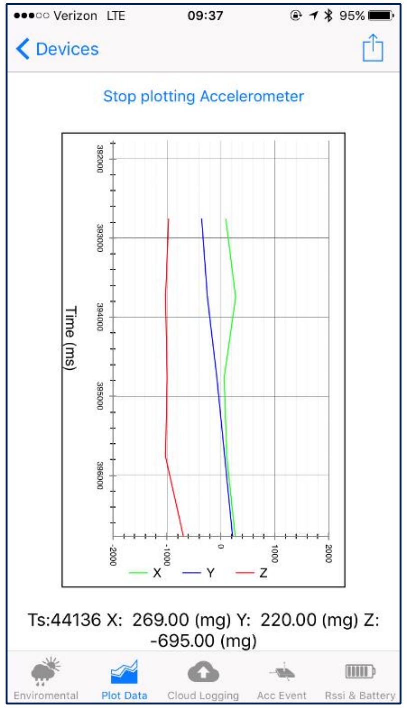
1

Select
"Start logging"



2

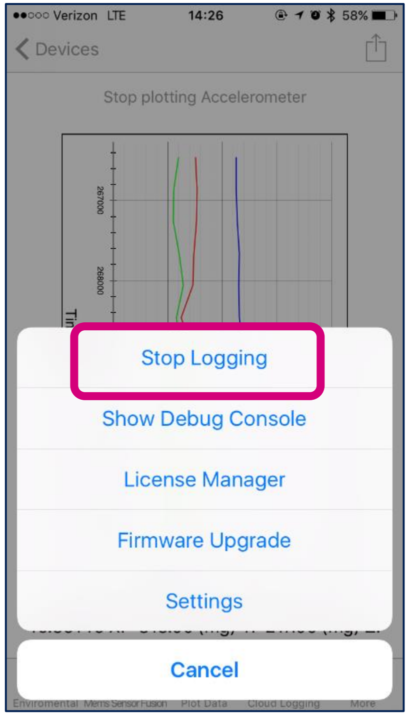
After some time
re-click on the
right-top corner



Ts:44136 X: 269.00 (mg) Y: 220.00 (mg) Z: -695.00 (mg)

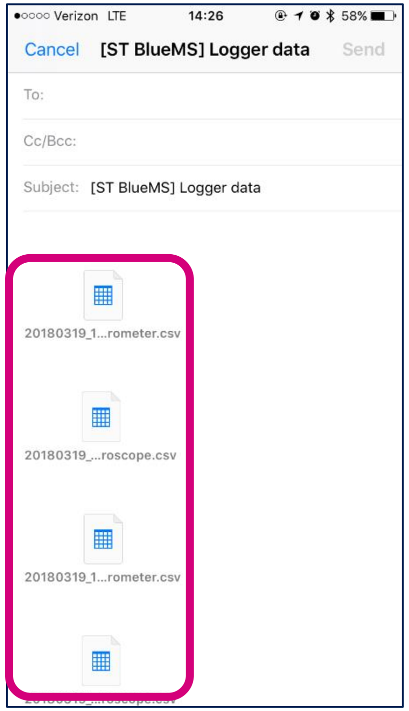
3

Select
"Stop logging"



4

App opens your
email client



To:

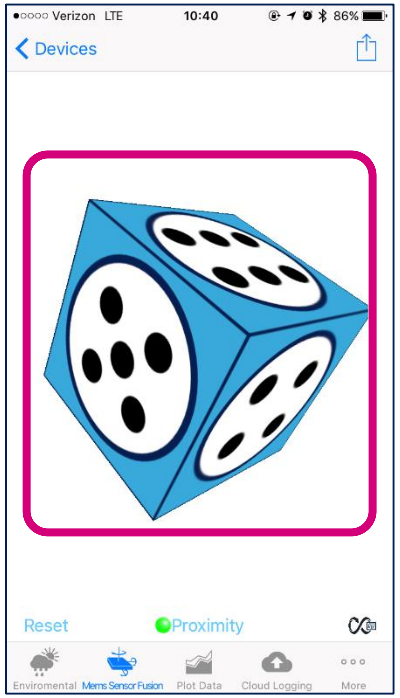
Cc/Bcc:

Subject: [ST BlueMS] Logger data

.csv files attached
to the email

Sensor Data Fusion

Sensor Fusion
enabled



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

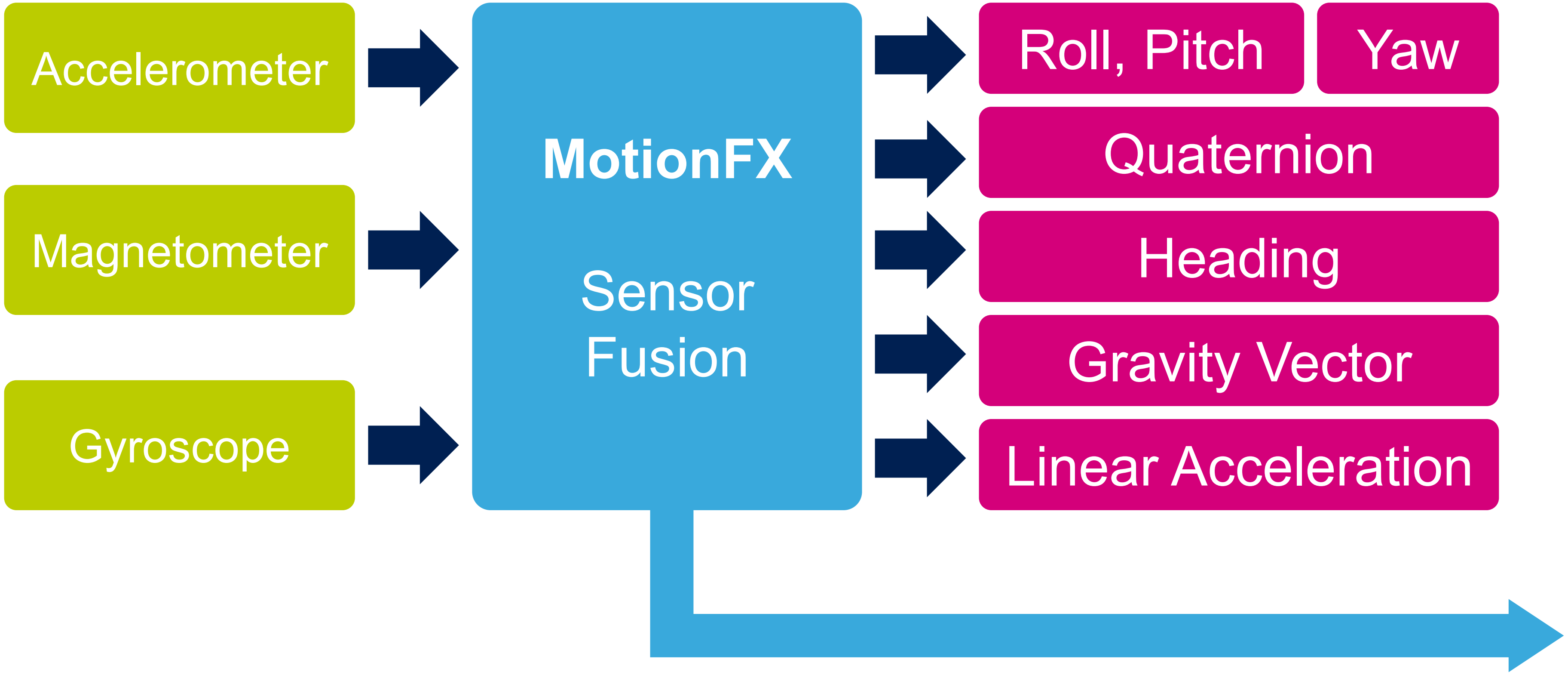
- In steady static position the **acc** and **mag** enable an accurate orientation estimate
- acc vibrations and mag bias and interferences are **rejected or compensated by the motionFX library**

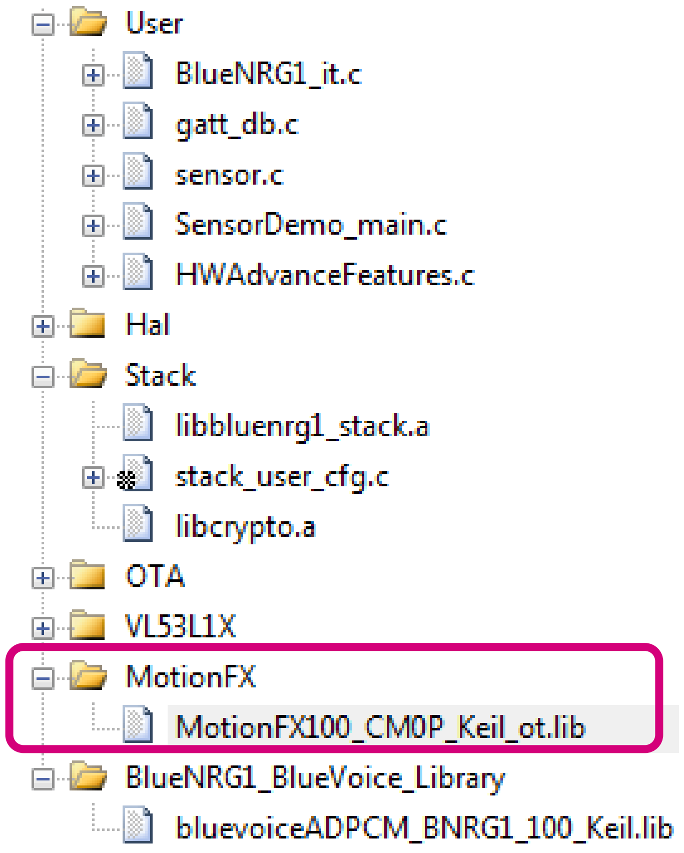
Try highly **dynamic** motion: perfect tracking

- 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

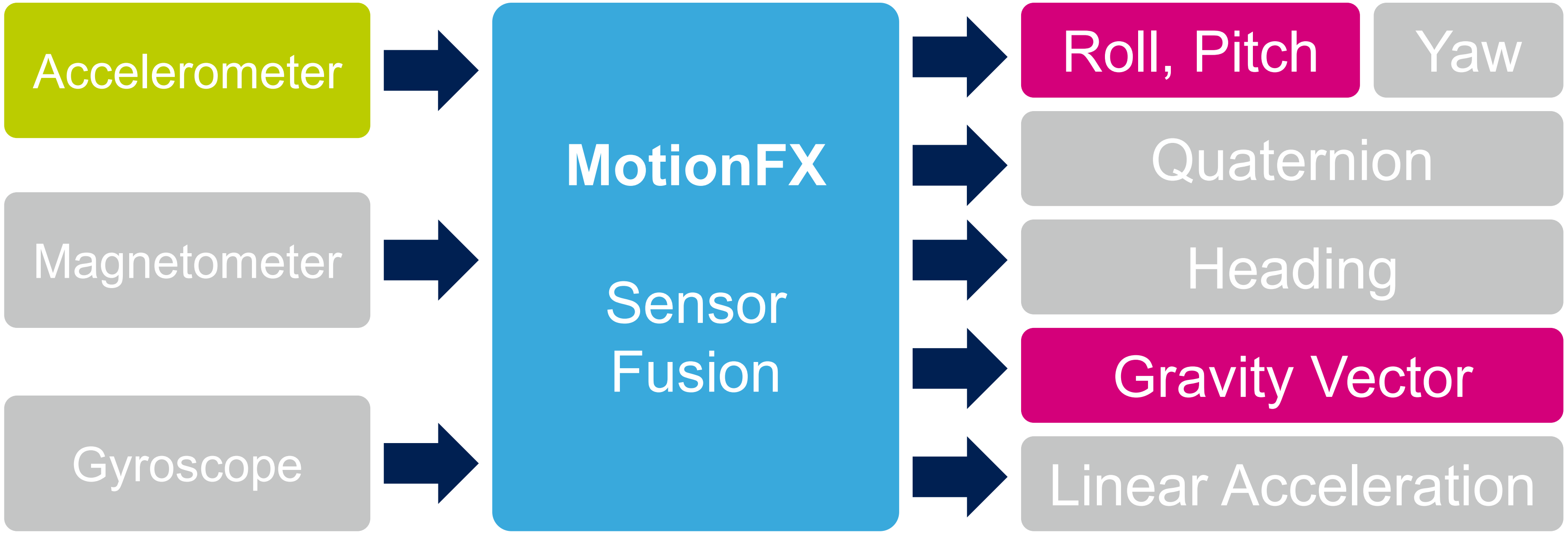




```
User
    BlueNRG1_it.c
    gatt_db.c
    sensor.c
    SensorDemo_main.c
    HWAdvanceFeatures.c
Hal
Stack
    libbluenrg1_stack.a
    stack_user_cfg.c
    libcrypto.a
OTA
VL53L1X
MotionFX
    MotionFX100_CM0P_Keil_ot.lib
BlueNRG1_BlueVoice_Library
    bluevoiceADPCM_BNRG1_100_Keil.lib
```

Sensor Data Fusion

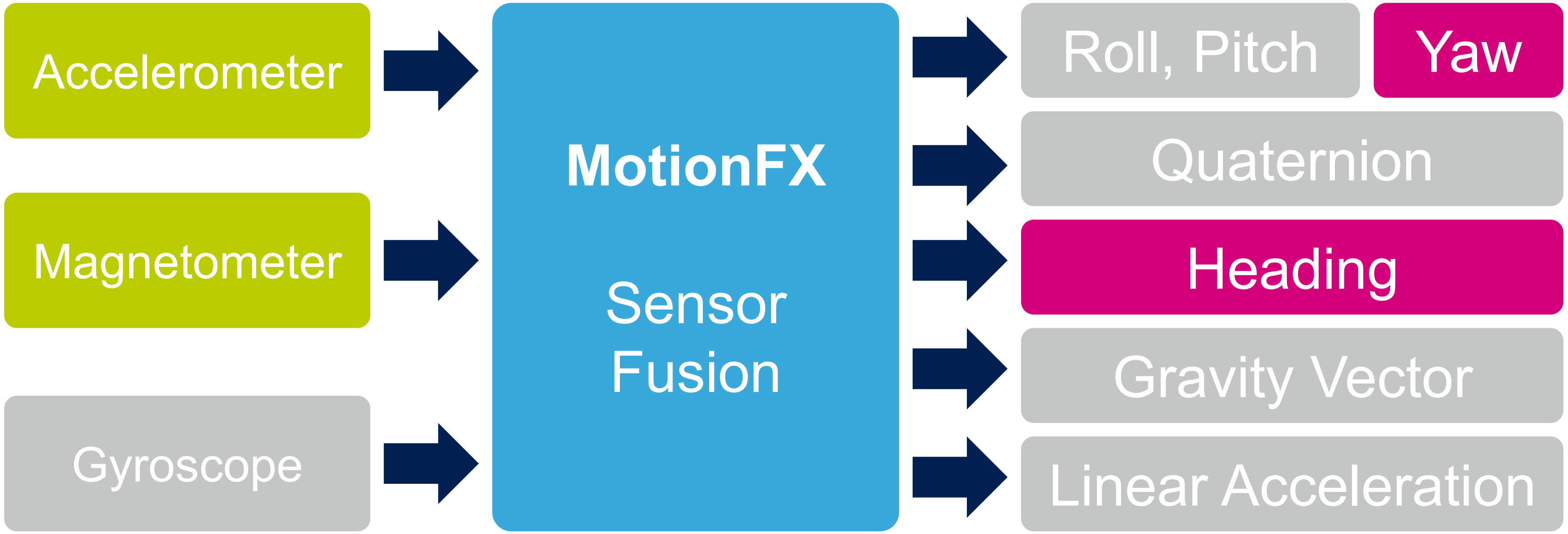
MotionFX Library



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

Sensor Data Fusion

MotionFX Library



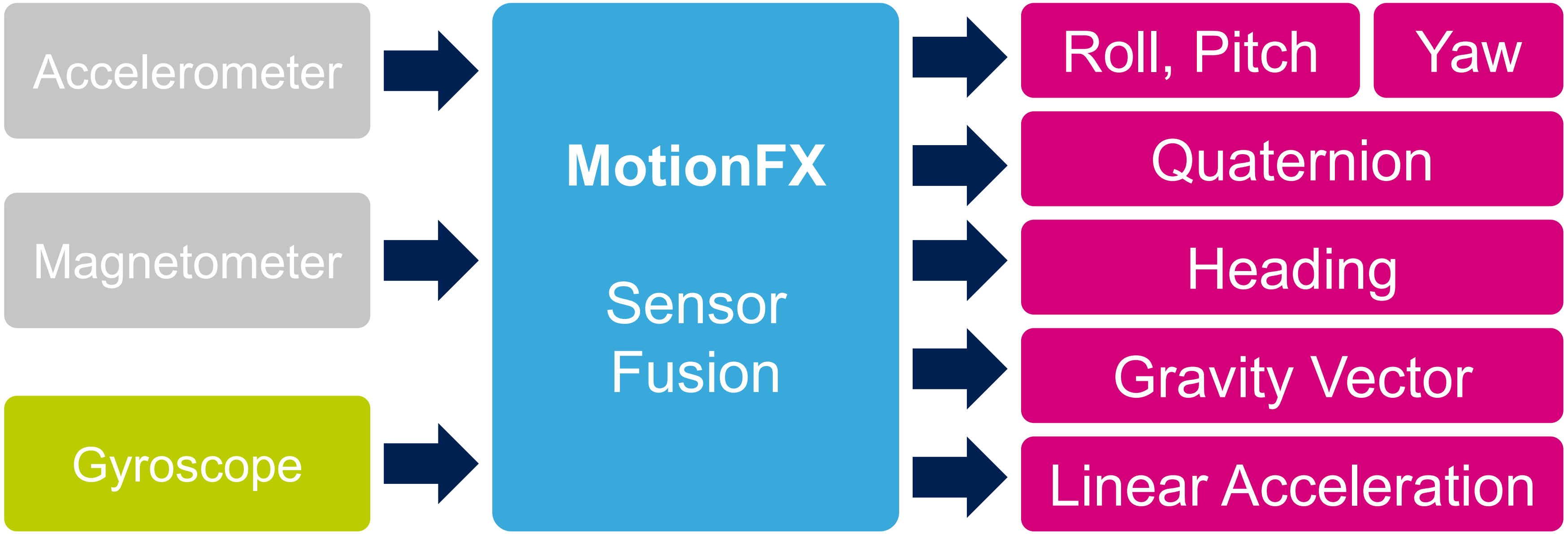
- **Magnetometer gives yaw angle** and **heading**

…but **only if tilt is compensated**: the accelerometer is needed!

…and **mag bias** (hard-iron) **must be compensated!**

Sensor Data Fusion

MotionFX Library

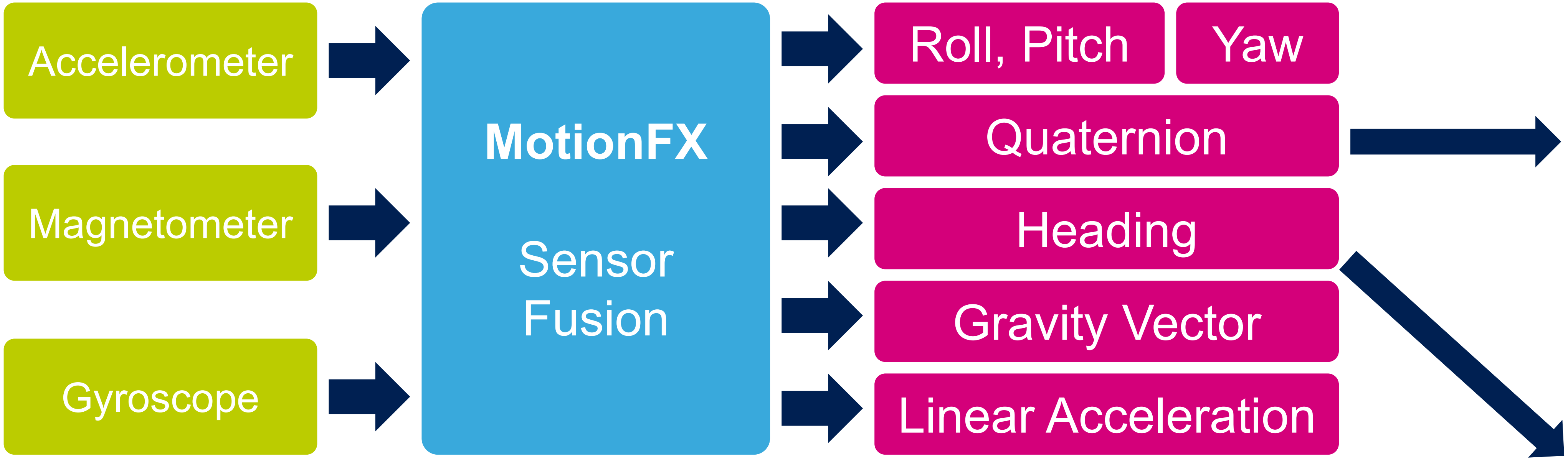


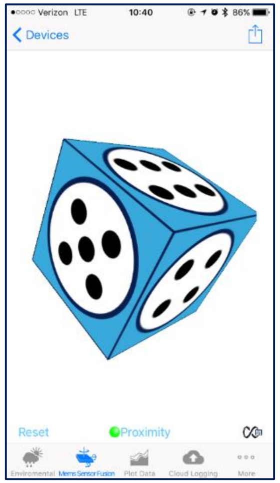
- **Gyroscope gives the new orientation** based on previous orientation

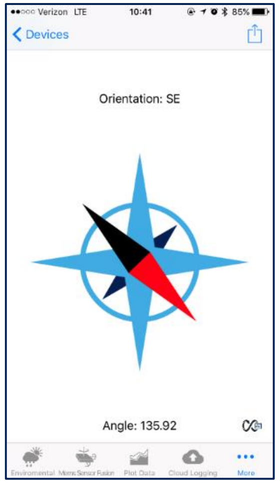
…but **gyro bias offset needs to be compensated** for accuracy!

Sensor Data Fusion

MotionFX Library







Motion FX library provides orientation estimation plus:

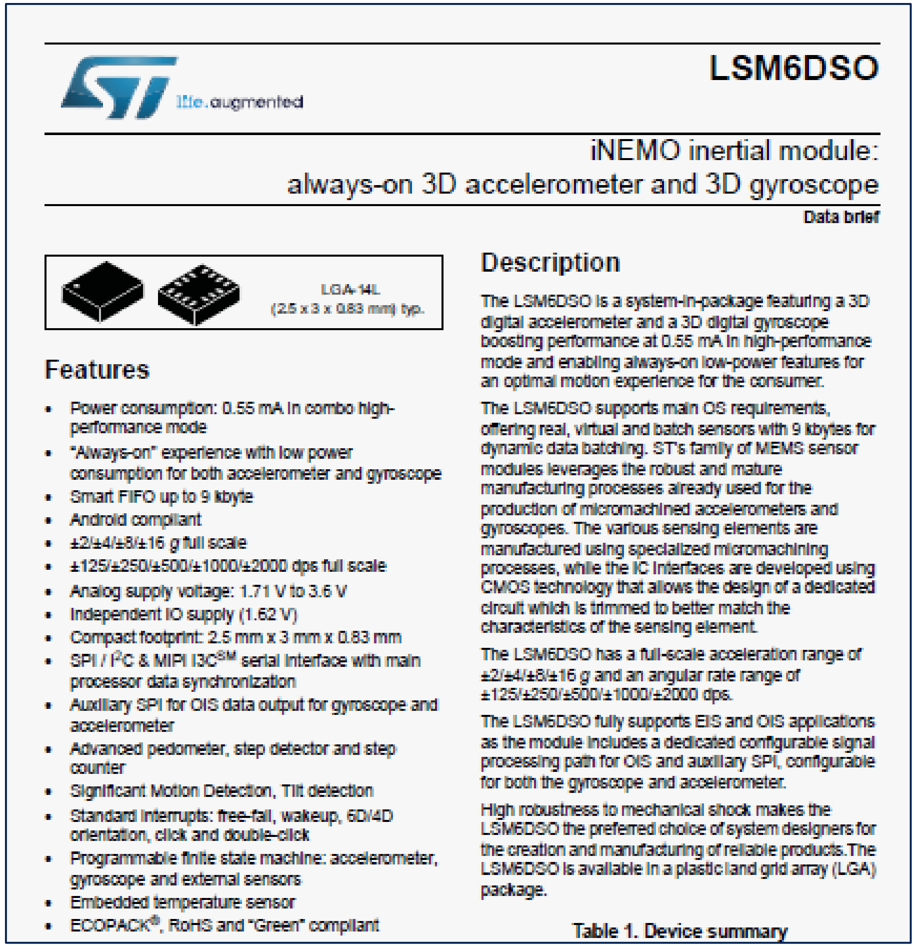
- **Magnetometer hard-iron offset** compensation.
- **Accelerometer vibrations** rejection.
- **Gyroscope bias** offset compensation.

Gyroscope

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

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-14L
(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
- ±2/±4/±8/±16 *g* full scale
- ±125/±250/±500/±1000/±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^2C & MIPI $I3C^{SM}$ 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
- ECOPACK®, 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 ±2/±4/±8/±16 *g* and an angular rate range of ±125/±250/±500/±1000/±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

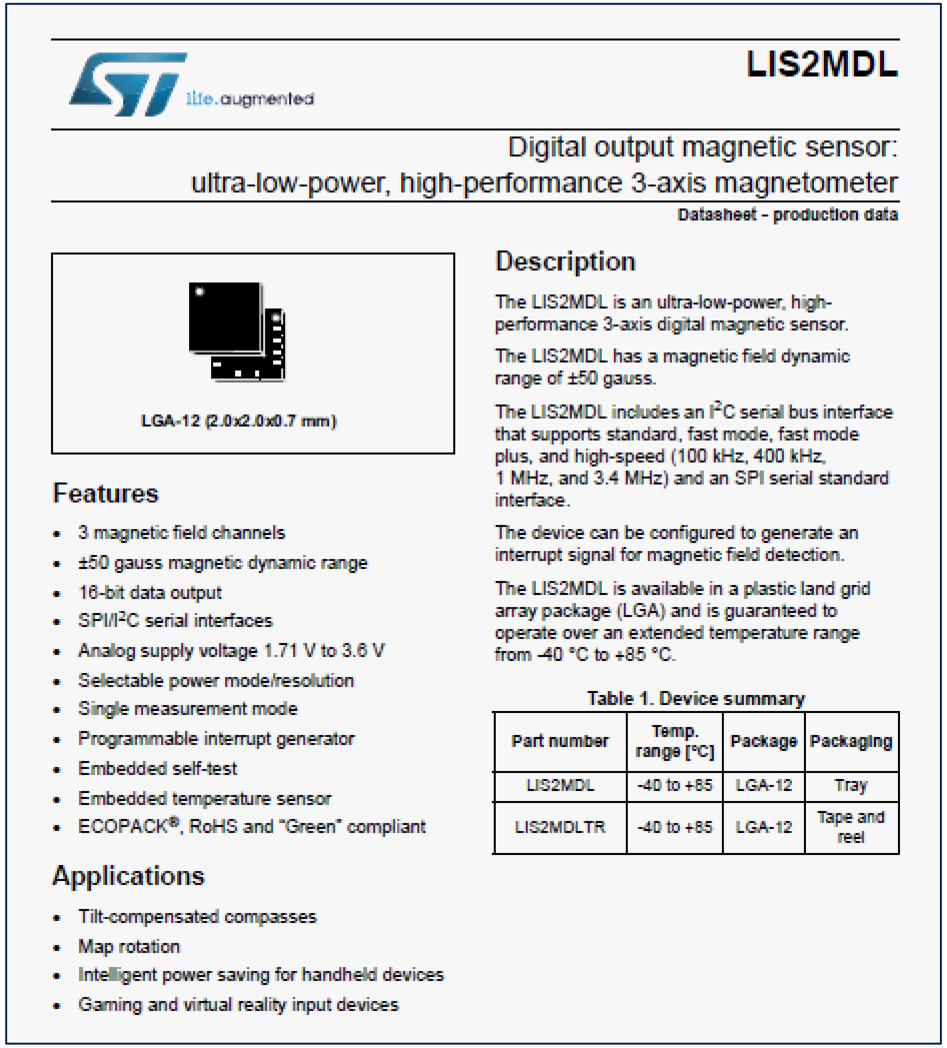
life.augmented

Magnetometer

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)

Features

- 3 magnetic field channels
- ±50 gauss magnetic dynamic range
- 16-bit data output
- SPI/I^{2}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

Description

The LIS2MDL is an ultra-low-power, high-performance 3-axis digital magnetic sensor.

The LIS2MDL has a magnetic field dynamic range of ±50 gauss.

The LIS2MDL includes an I^{2}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.

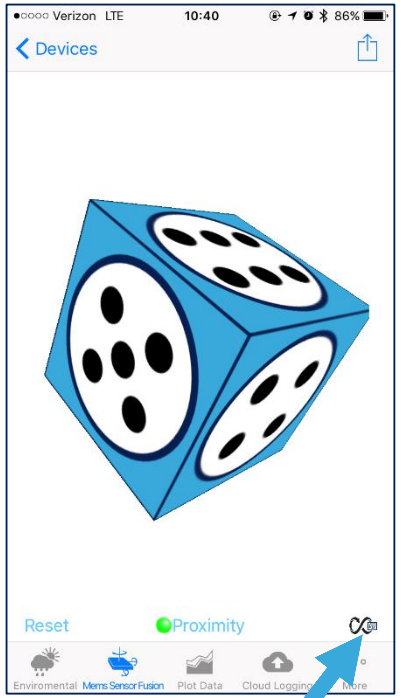
The LIS2MDL is available in a plastic land grid array package (LGA) and is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

Table 1. Device summary

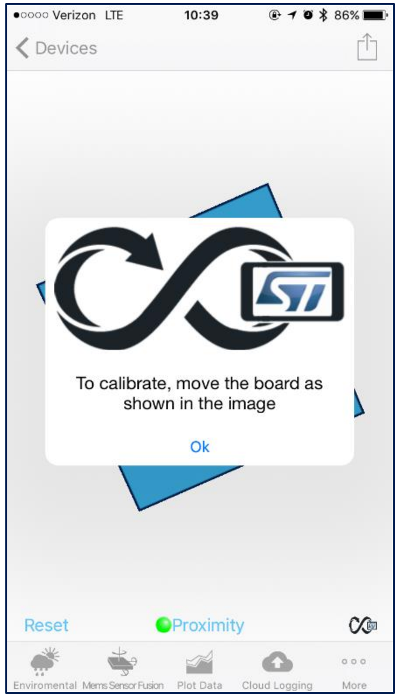
Part number	Temp. range [°C]	Package	Packaging
LIS2MDL	-40 to +85	LGA-12	Tray
LIS2MDLTR	-40 to +85	LGA-12	Tape and reel

Magnetometer

Calibration



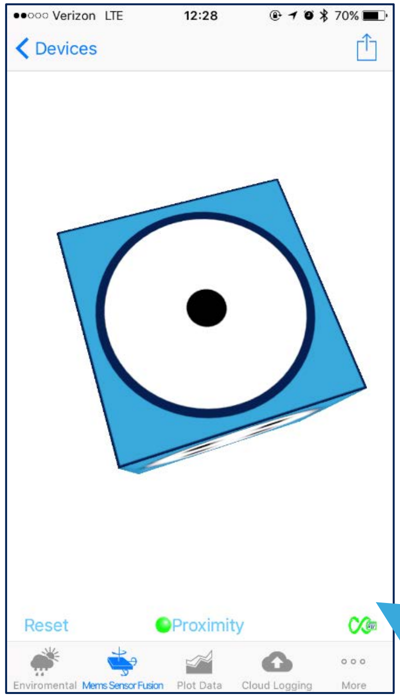
Tap



To calibrate, move the board as shown in the image

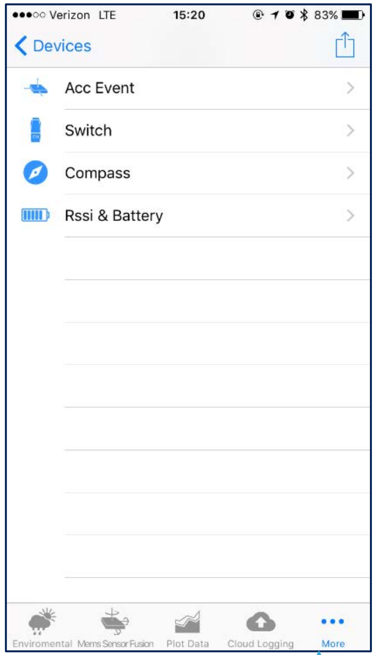
Ok

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



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

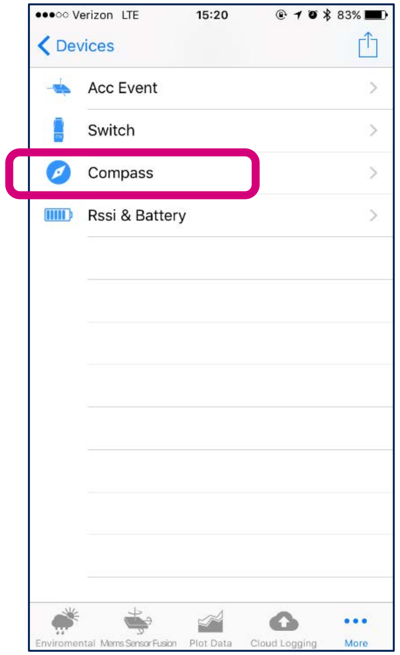
Magnetometer ECompass


●●●○○ Verizon LTE 15:20 83%

< Devices

- Acc Event
- Switch
- Compass
- Rssi & Battery

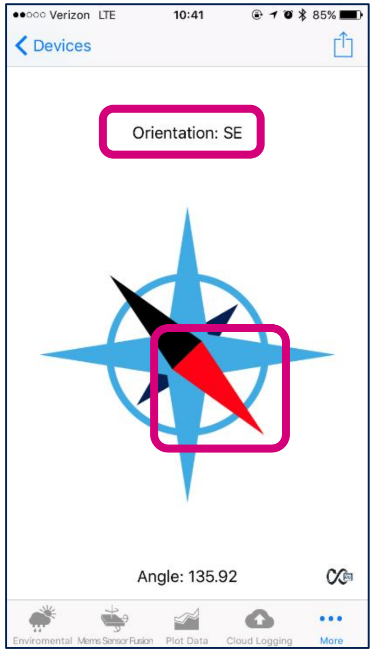
Tap ••• More


●●●○○ Verizon LTE 15:20 83%

< Devices

- Acc Event
- Switch
- Compass
- Rssi & Battery

Select “Compass”


●●○○○ Verizon LTE 10:41 85%

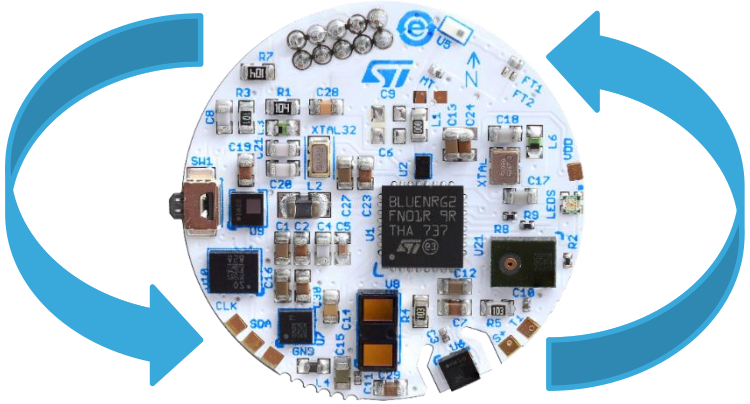
< Devices

Orientation: SE

Angle: 135.92

Red arrow is related to the current orientation of the BlueTile

Rotate the BlueTile



(you can check against the phone eCompass)

life.augmented

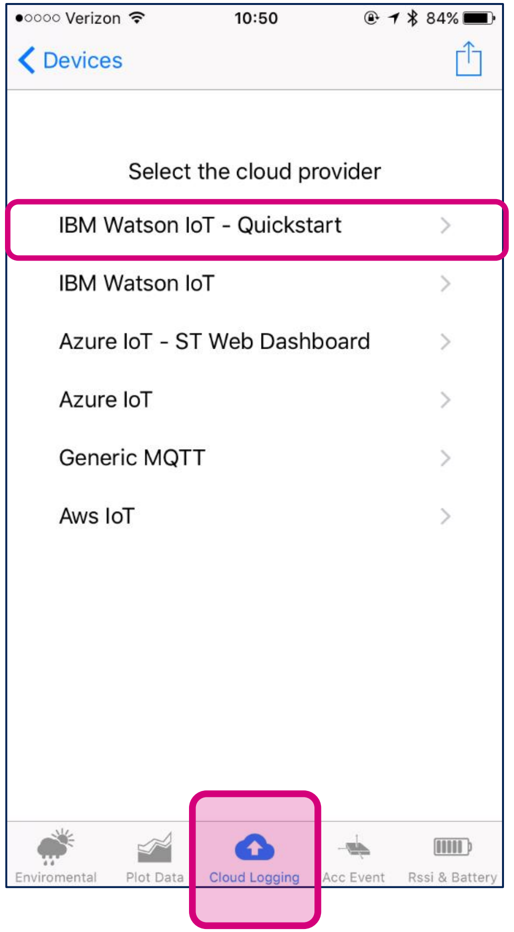
Lab 6

Cloud Data Logging on IBM Watson

Cloud Logging

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

IBM Watson



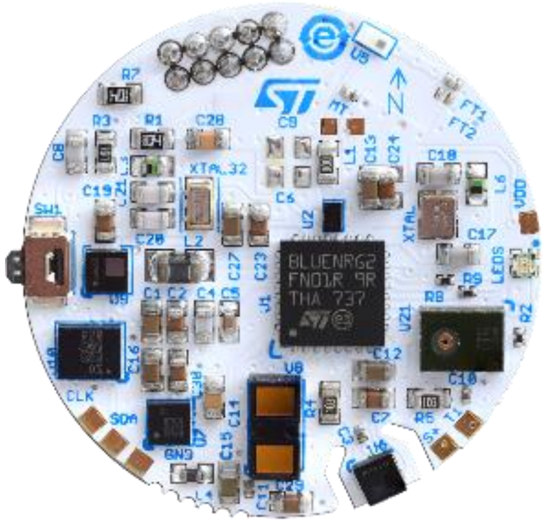
●○○○○ Verizon 10:50 84%

Devices

Select the cloud provider

- IBM Watson IoT - Quickstart
- IBM Watson IoT
- Azure IoT - ST Web Dashboard
- Azure IoT
- Generic MQTT
- Aws IoT

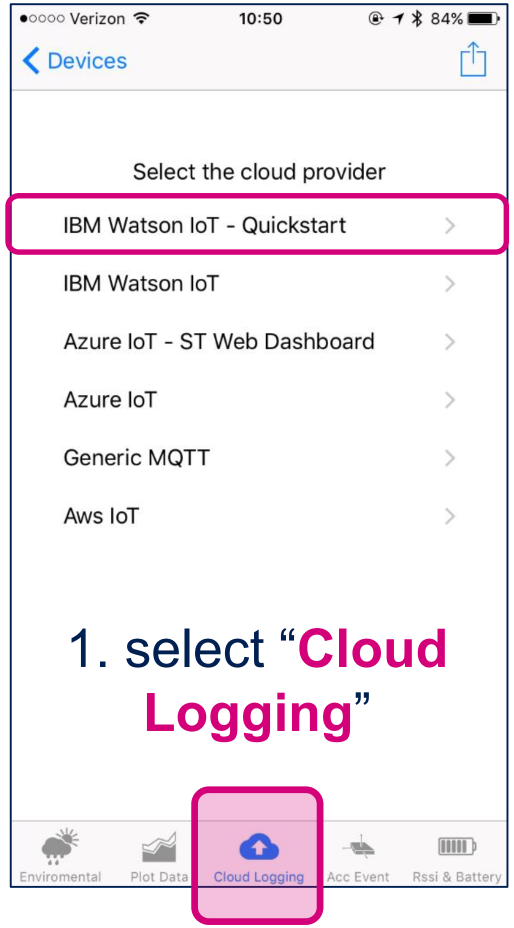
Notification Packets



life.augmented

IBM Watson IoT Quickstart

2. **Select “IBM Watson IoT - Quickstart”**



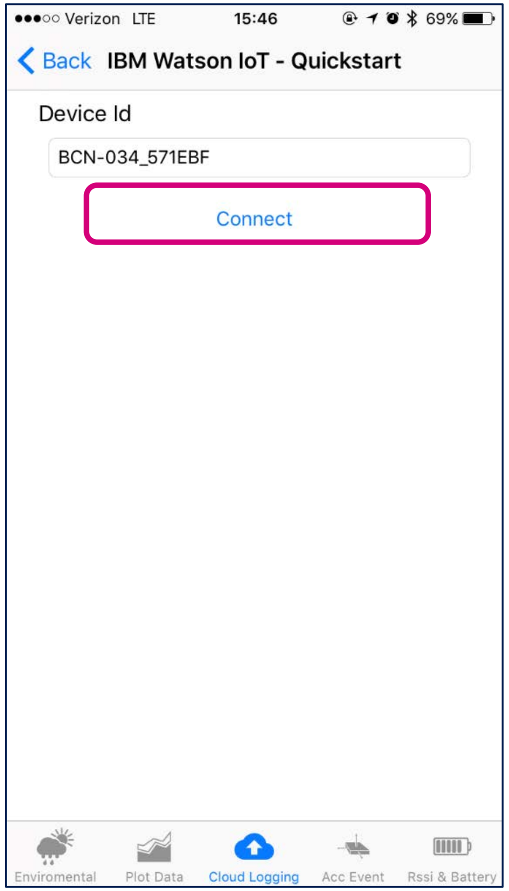
Devices

Select the cloud provider

- IBM Watson IoT - Quickstart
- IBM Watson IoT
- Azure IoT - ST Web Dashboard
- Azure IoT
- Generic MQTT
- Aws IoT

1. select “**Cloud Logging**”

Click “Connect”



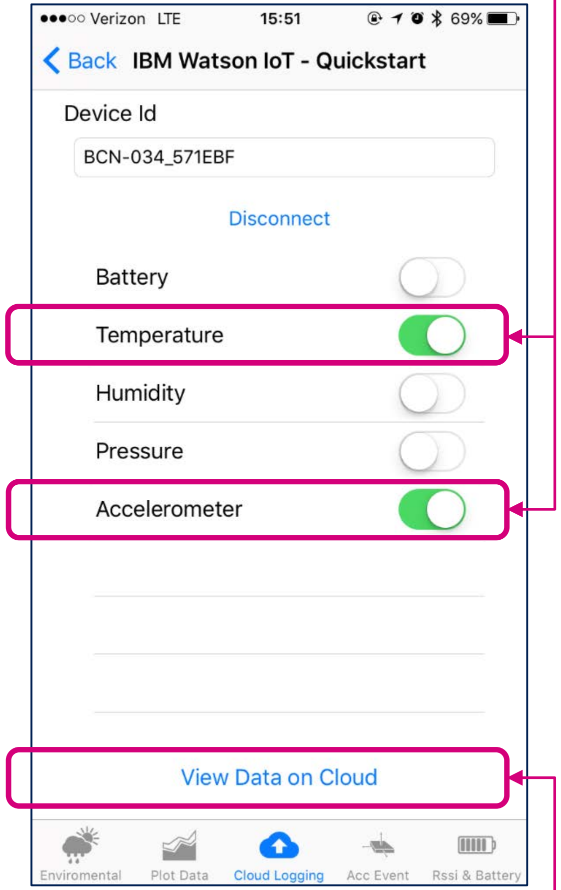
Back IBM Watson IoT - Quickstart

Device Id

BCN-034_571EBF

Connect

1. **Select** one or multiple features



Back IBM Watson IoT - Quickstart

Device Id

BCN-034_571EBF

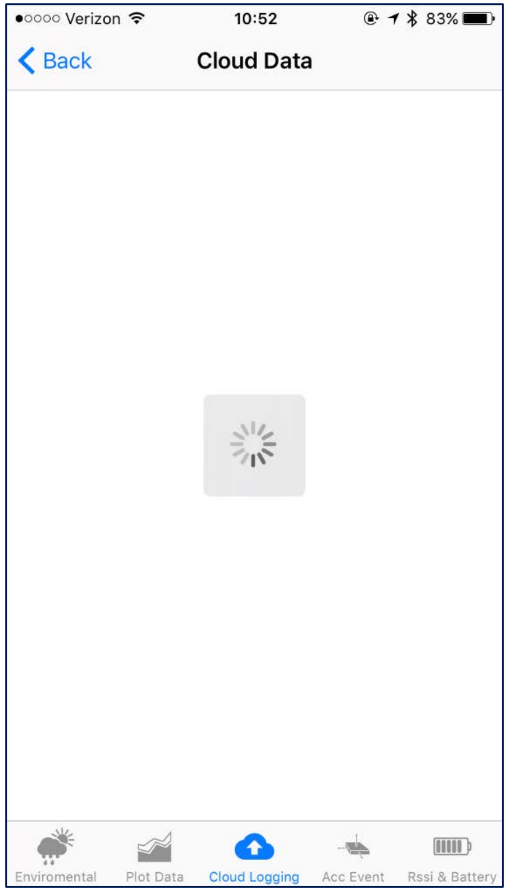
Disconnect

- Battery
- Temperature
- Humidity
- Pressure
- Accelerometer

View Data on Cloud

2. **Click “View Data on Cloud”**

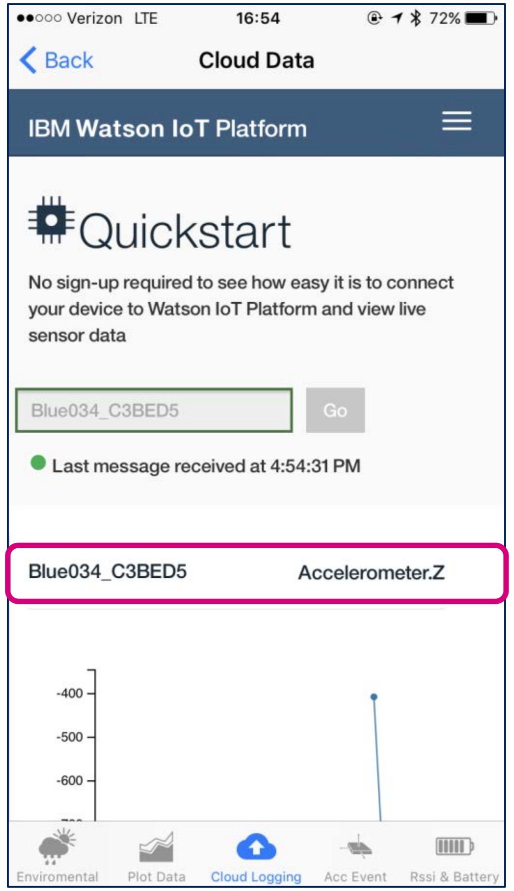
Wait a few seconds



Back Cloud Data

IBM Watson IoT Quickstart

Quickstart will appear



Quickstart

No sign-up required to see how easy it is to connect your device to Watson IoT Platform and view live sensor data

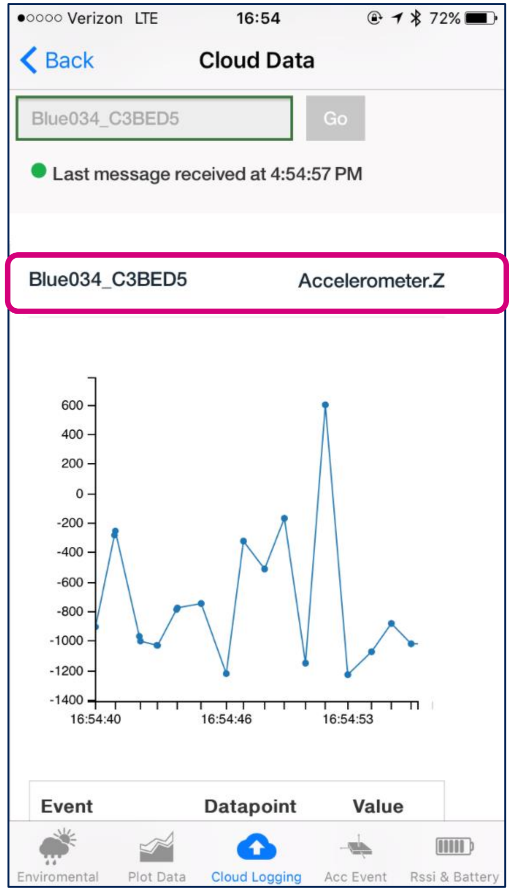
Blue034_C3BED5

● Last message received at 4:54:31 PM

Blue034_C3BED5 Accelerometer.Z

Scroll down to see your selected sensor plot.

You will see the **Plot** of selected feature



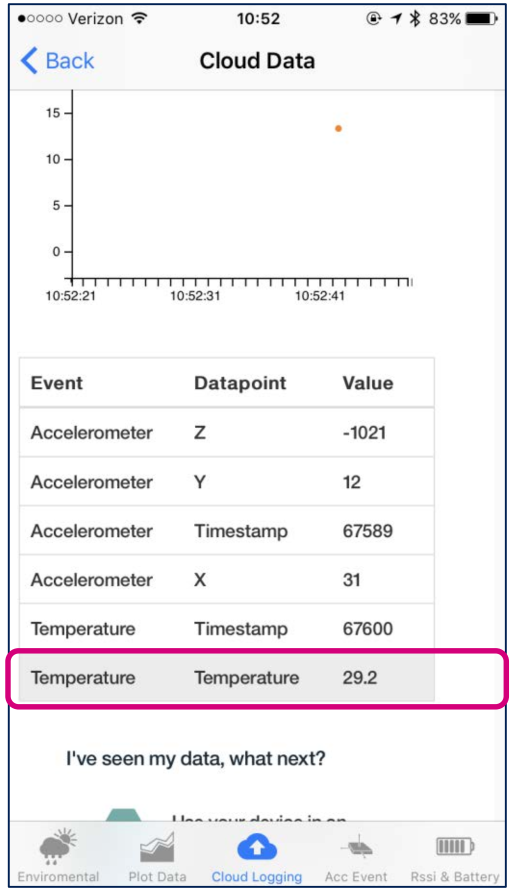
Blue034_C3BED5

● Last message received at 4:54:57 PM

Blue034_C3BED5 Accelerometer.Z

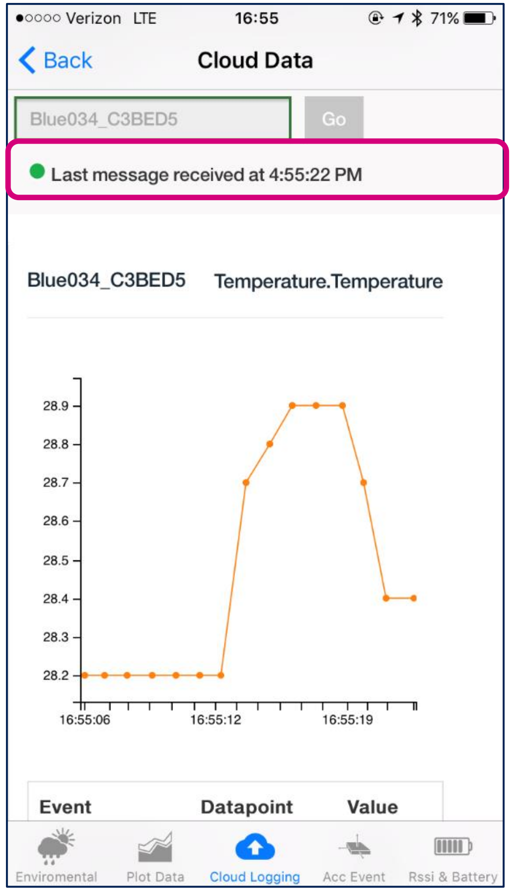
Event	Datapoint	Value

Table of **available features**



Event	Datapoint	Value
Accelerometer	Z	-1021
Accelerometer	Y	12
Accelerometer	Timestamp	67589
Accelerometer	X	31
Temperature	Timestamp	67600
Temperature	Temperature	29.2

I've seen my data, what next?



Blue034_C3BED5

● Last message received at 4:55:22 PM

Blue034_C3BED5 Temperature.Temperature

Event	Datapoint	Value

Scroll down again to change sensor data or axes

Lab 7

Voice over BLE

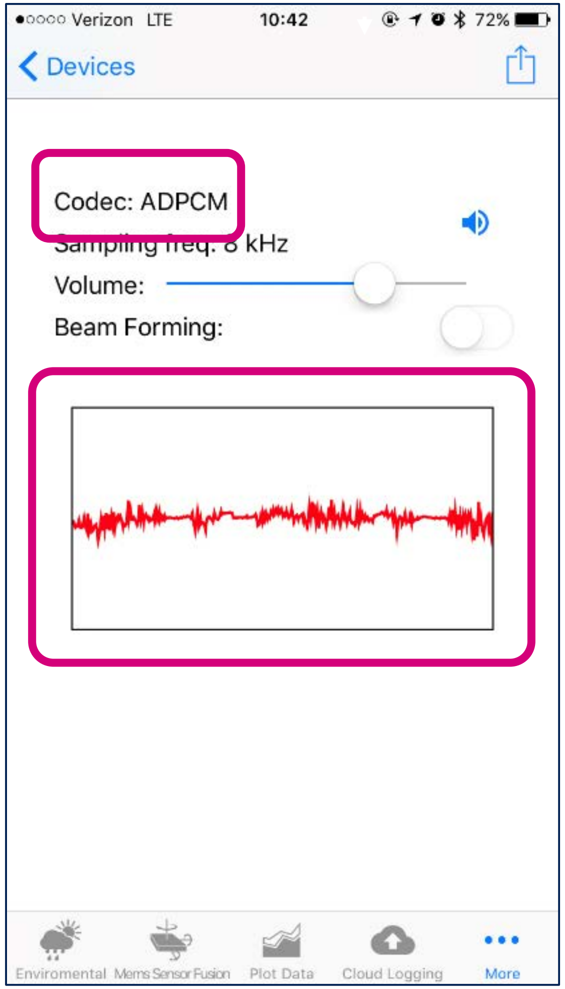
Goal

- 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

life.augmented

BlueVoice Library

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



Codec: ADPCM

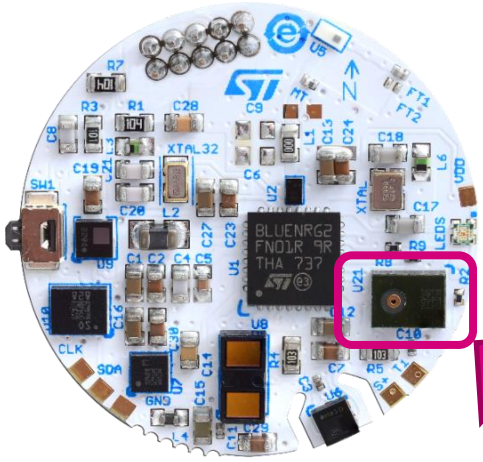
Sampling freq: 8 kHz

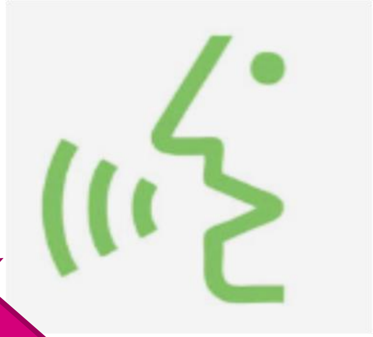
Volume:

Beam Forming:

Notification Packet

ADPCM compressed audio





MEMS Microphone

ST BlueMS Protocol

4 Bytes

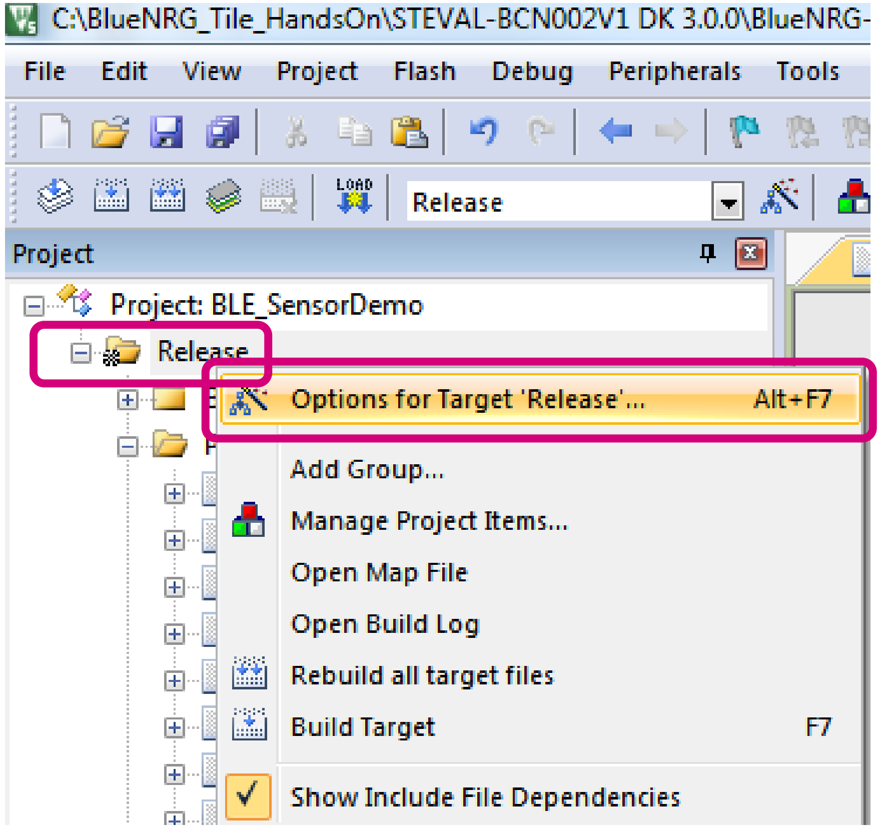
Len	Type	Protocol version	Device ID	Feature Mask	Device MAC	Len	Type	Local Name	Len	Type	Pwr

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

Enable BlueVoice Library

Add the preprocessor symbol **ENABLE_AUDIO** to the project

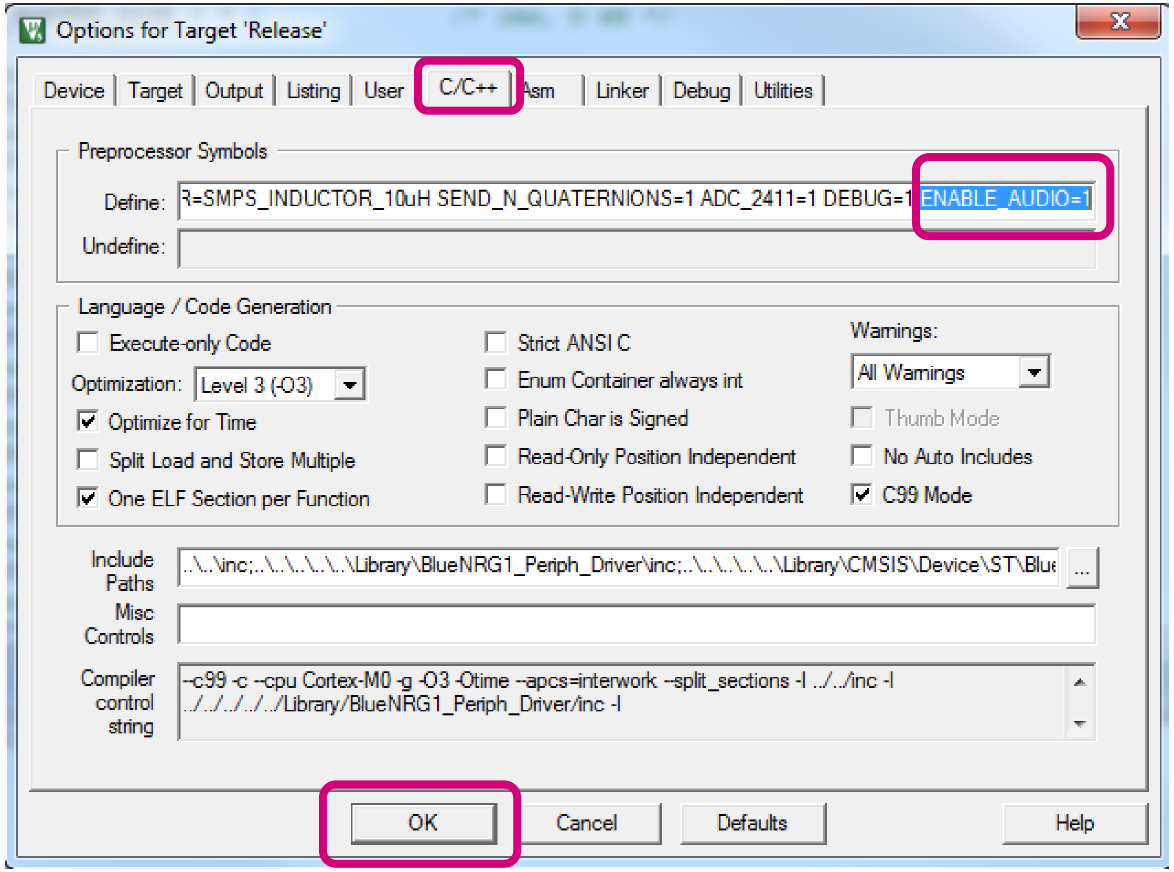
1. **Right-click** on **Release**
2. **Select “Options for Target ‘Release’…”**



Enable BlueVoice Library

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**



Options for Target 'Release'

Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities

Preprocessor Symbols

Define: R=SMPS_INDUCTOR_10uH SEND_N_QUATERNIONS=1 ADC_2411=1 DEBUG=1 ENABLE_AUDIO=1

Undefine:

Language / Code Generation

- □ Execute-only Code
- Optimization: Level 3 (-O3)
- ☑ Optimize for Time
- □ Split Load and Store Multiple
- ☑ One ELF Section per Function
- □ Strict ANSI C
- □ Enum Container always int
- □ Plain Char is Signed
- □ Read-Only Position Independent
- □ Read-Write Position Independent
- Warnings: All Warnings
- □ Thumb Mode
- □ No Auto Includes
- ☑ C99 Mode

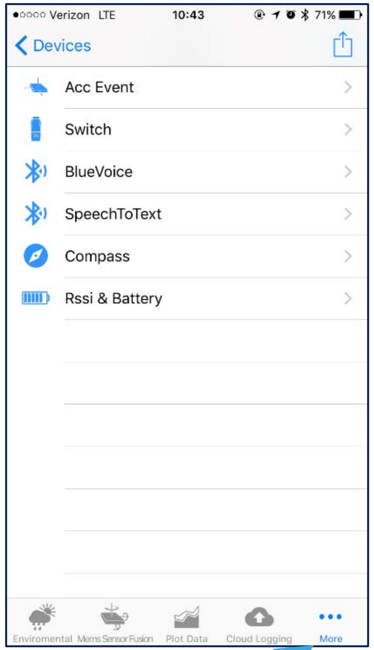
Include Paths: ..\..\inc;..\..\..\..\..\Library\BlueNRG1_Periph_Driver\inc;..\..\..\..\..\Library\CMSIS\Device\ST\Blue

Misc Controls:

Compiler control string:

```
--c99 -c --cpu Cortex-M0 -g -O3 -Otime --apcs=interwork --split_sections -I ../../inc -I
../../../../../Library/BlueNRG1_Periph_Driver/inc -I
```

BlueVoice: Voice over Bluetooth LE



●○○○○ Verizon LTE 10:43 71%

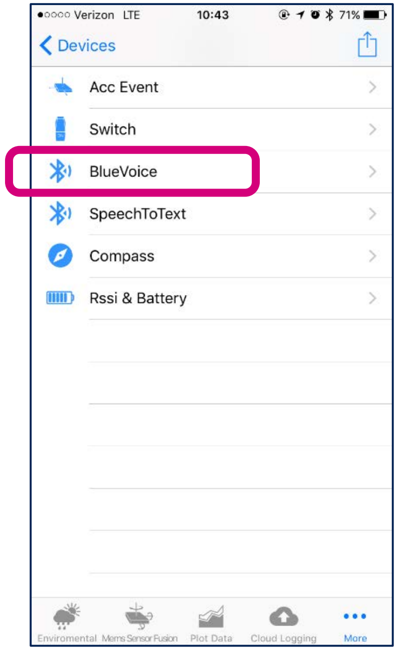
< Devices

- Acc Event
- Switch
- BlueVoice
- SpeechToText
- Compass
- Rssi & Battery

Tap ••• More

Select

“BlueVoice”

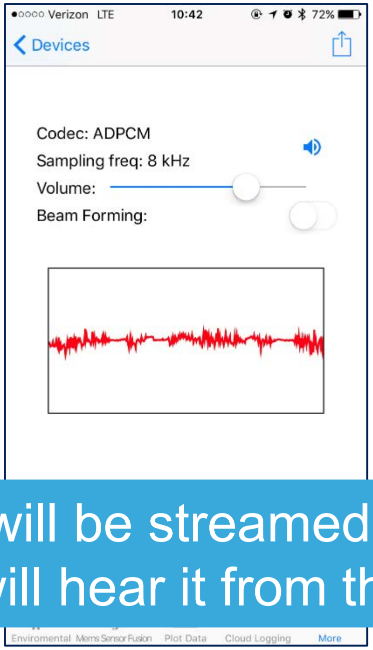


●○○○○ Verizon LTE 10:43 71%

< Devices

- Acc Event
- Switch
- BlueVoice
- SpeechToText
- Compass
- Rssi & Battery

Speak close to the BlueNRG-Tile



●○○○○ Verizon LTE 10:42 72%

< Devices

Codec: ADPCM

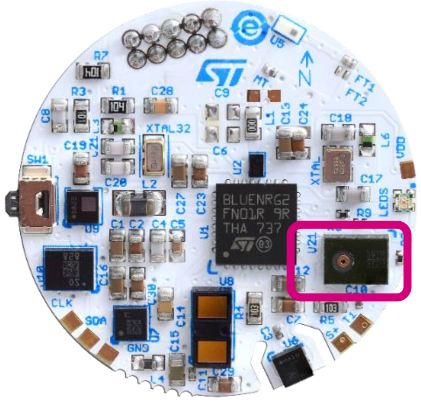
Sampling freq: 8 kHz

Volume:

Beam Forming:

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

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

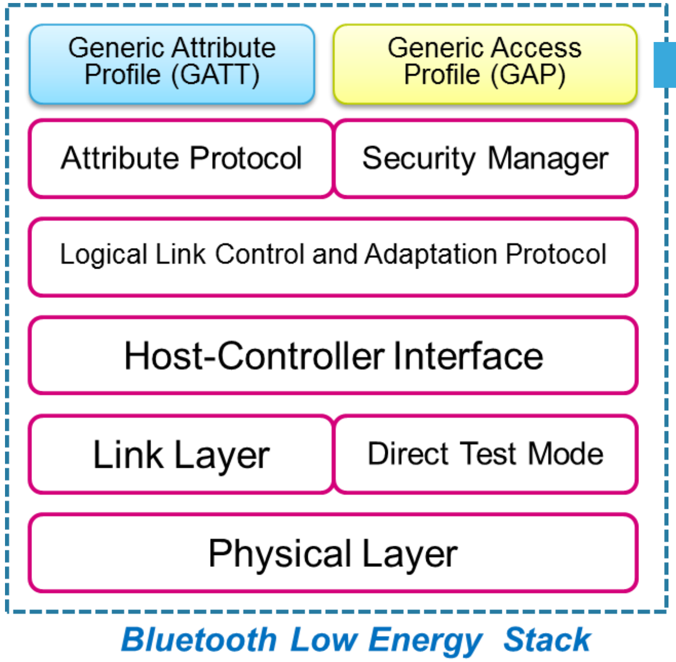


MEMS Microphone

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

life.augmented

Voice over Bluetooth LE



Generic Attribute Profile (GATT)

Generic Access Profile (GAP)

Attribute Protocol

Security Manager

Logical Link Control and Adaptation Protocol

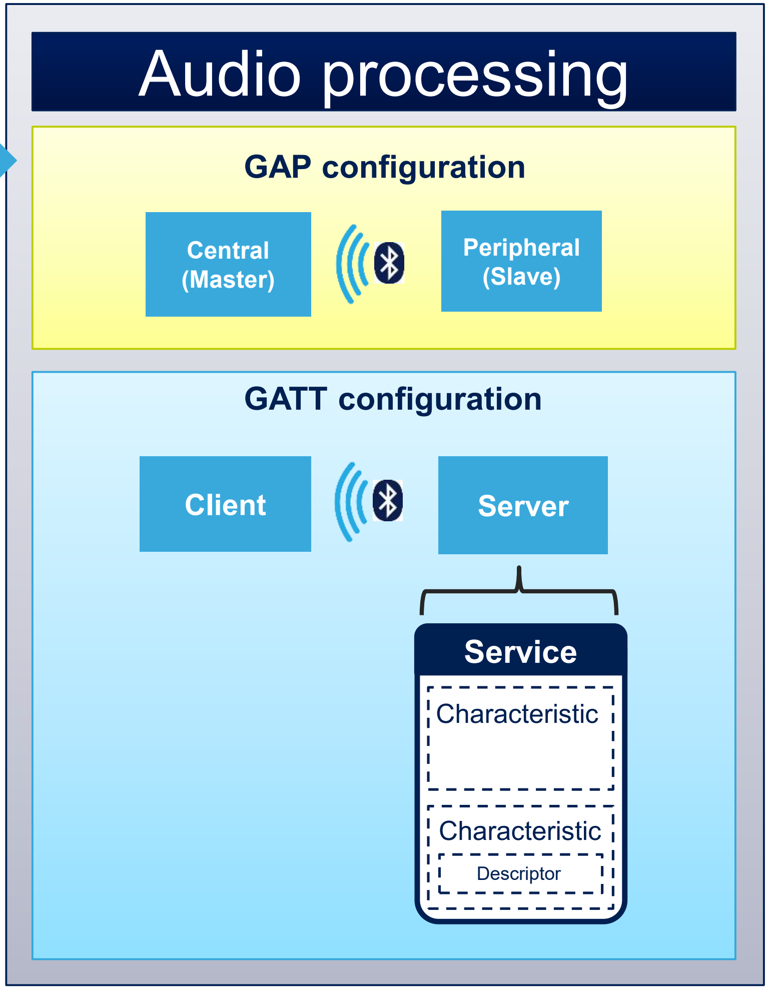
Host-Controller Interface

Link Layer

Direct Test Mode

Physical Layer

Bluetooth Low Energy Stack



Audio processing

GAP configuration

Central (Master)

Peripheral (Slave)

GATT configuration

Client

Server

Service

Characteristic

Characteristic

Descriptor

It's a Vendor Specific Service!

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

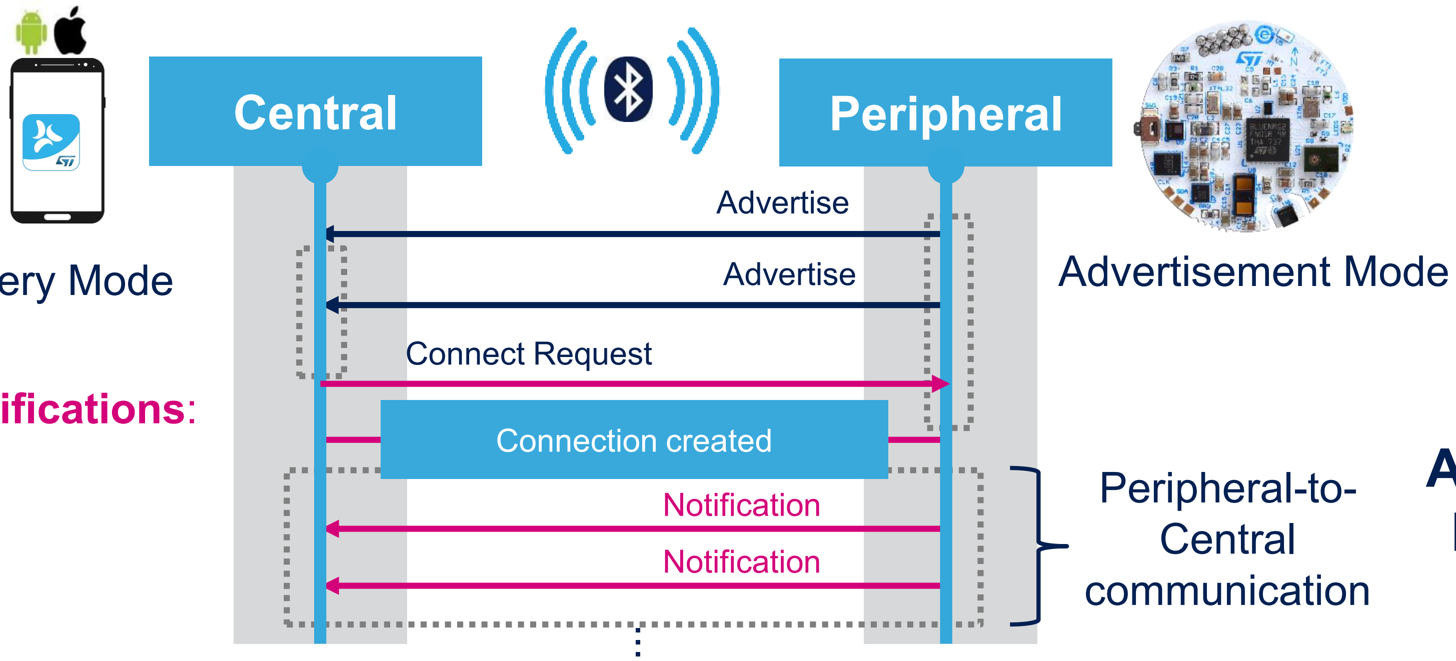
Voice streaming over BLE:

Audio @8kHz

Codec: ADPCM

Bitrate: 32kbps

BlueVoice ADPCM - Audio 8kHz



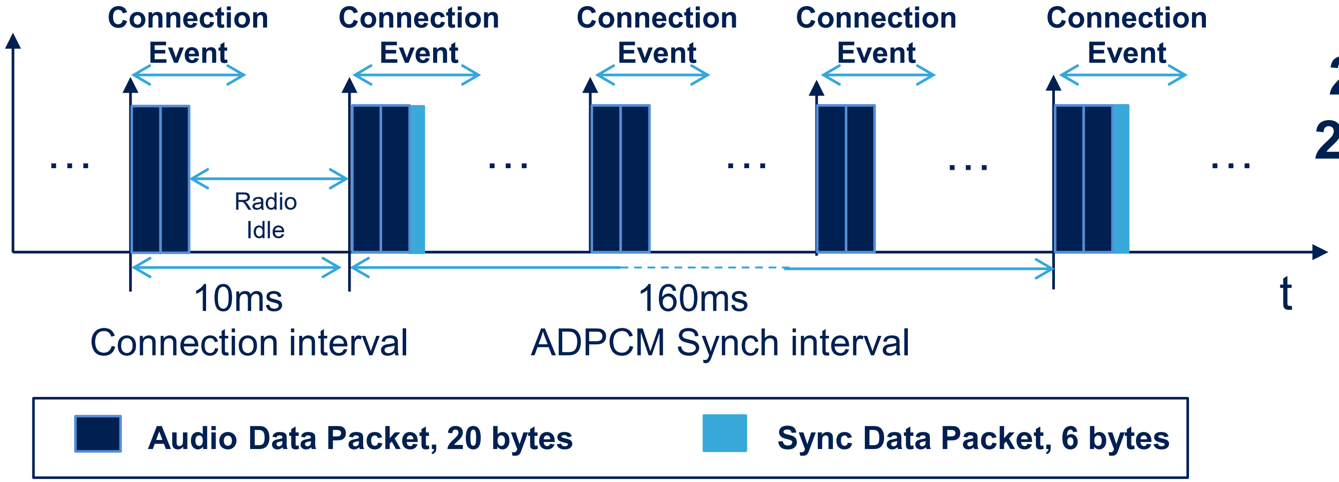
No acknowledge required for **Notifications**:

- 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

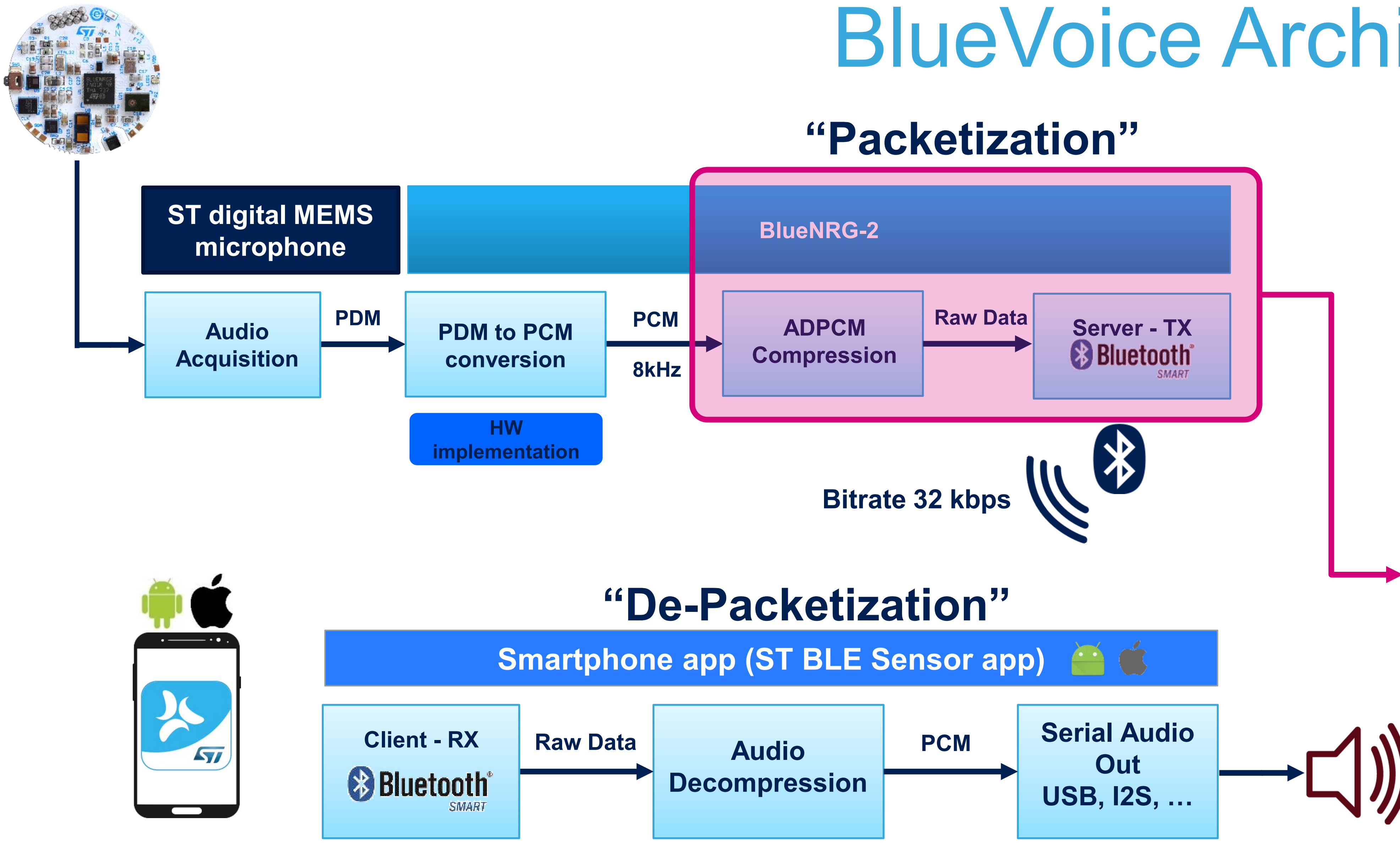


2 packets/event
20 bytes/packets
100 events/sec

=

32kbps

BlueVoice Architecture



“Packetization”

ST digital MEMS microphone

BlueNRG-2

Audio Acquisition

PDM

PDM to PCM conversion

HW implementation

PCM

8kHz

ADPCM Compression

Raw Data

Server - TX

Bluetooth SMART

Bitrate 32 kbps

“De-Packetization”

Smartphone app (ST BLE Sensor app)

Client - RX

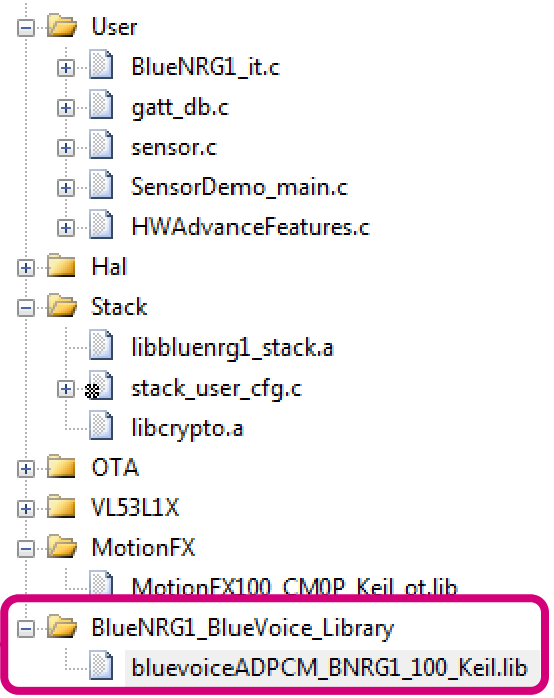
Bluetooth SMART

Raw Data

Audio Decompression

PCM

Serial Audio Out USB, I2S, …



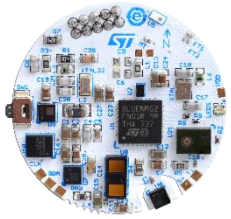
```
User
    BlueNRG1_it.c
    gatt_db.c
    sensor.c
    SensorDemo_main.c
    HWAdvanceFeatures.c
Hal
Stack
    libbluenrg1_stack.a
    stack_user_cfg.c
    libcrypto.a
OTA
VL53L1X
MotionFX
    MotionFX100_CM0P_Keil_ot.lib
BlueNRG1_BlueVoice_Library
    bluevoiceADPCM_BNRG1_100_Keil.lib
```

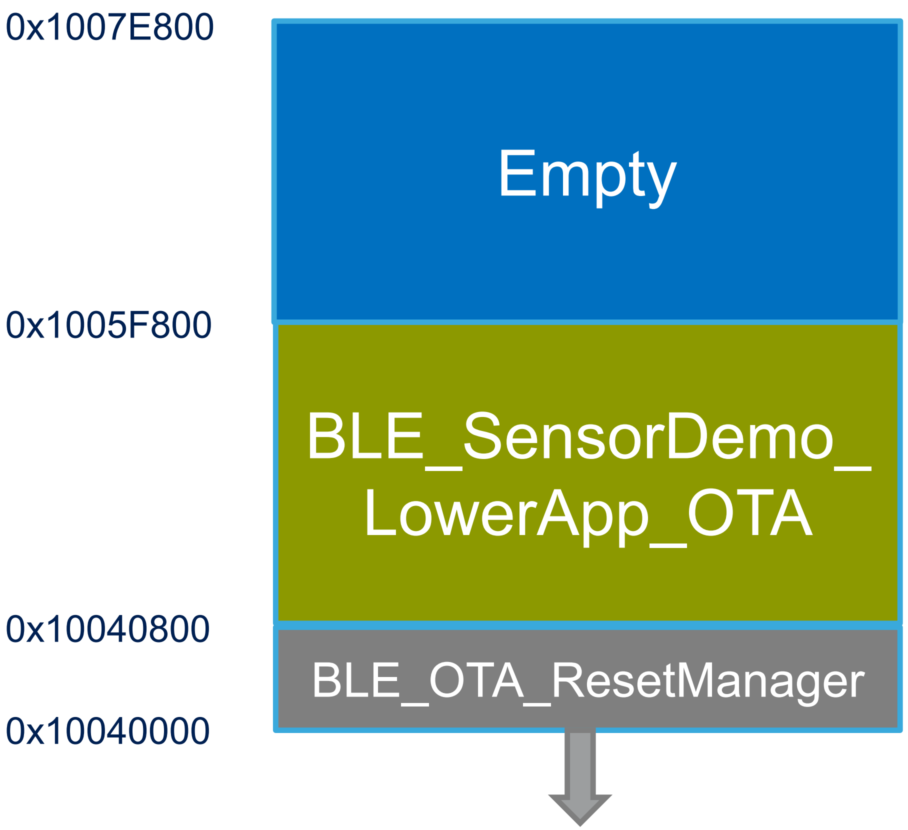
Lab 8

Over-The-Air (OTA) FW Upgrade

life.augmented

Flash Memory Layout



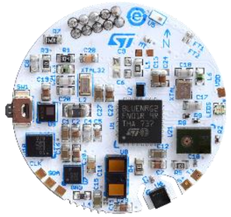


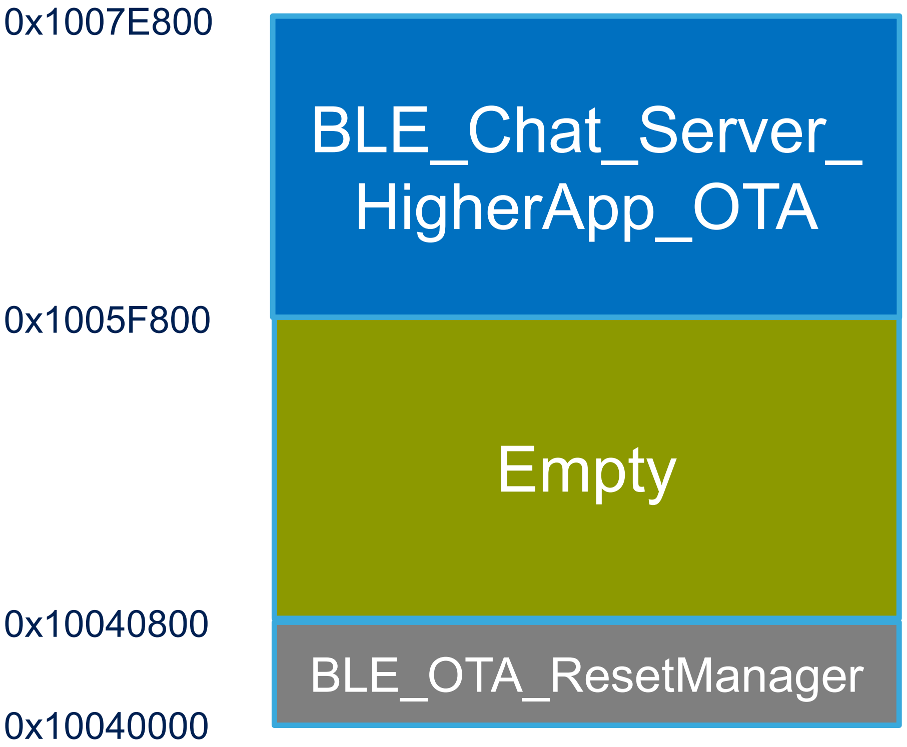
OTA

BLE_Chat_Server_HigherApp_OTA.bin

REMEMBER TO LOAD THE BLE OTA Reset manager!!!

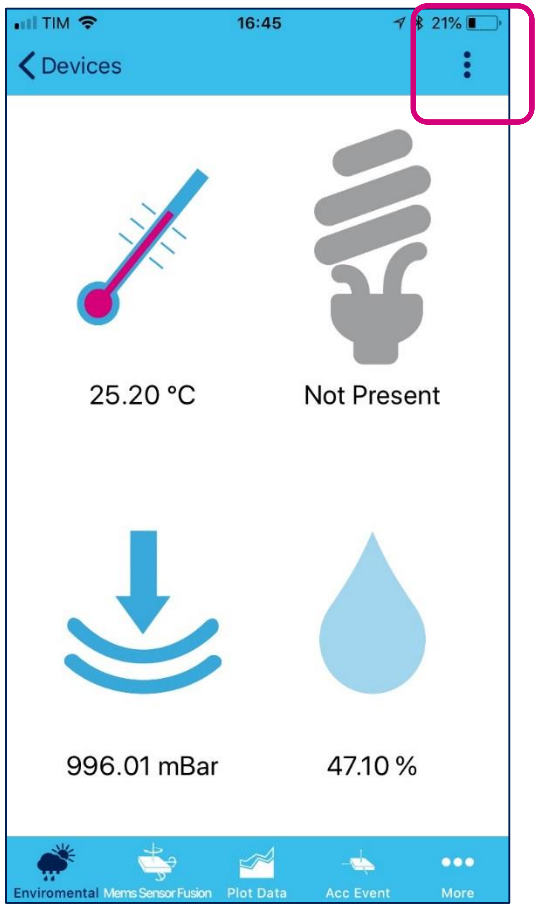
Once OTA is Completed…



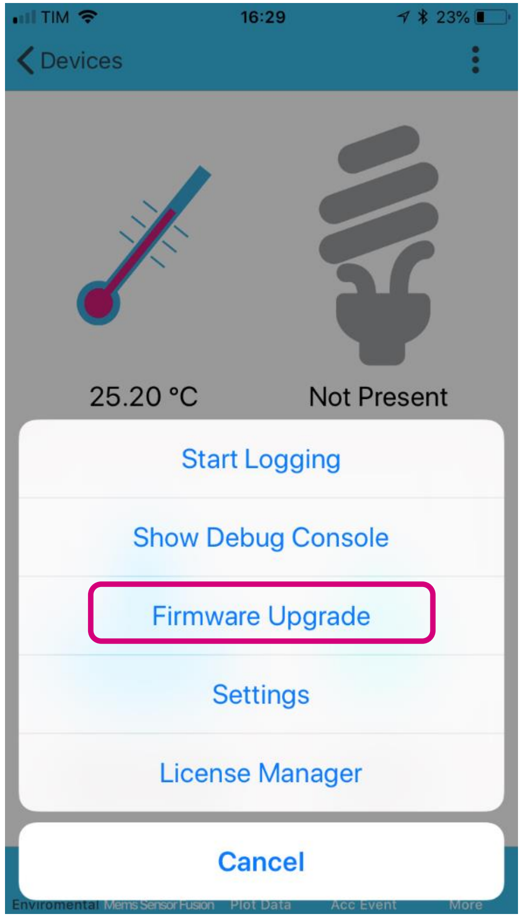


OTA FW Upgrade (1/2)

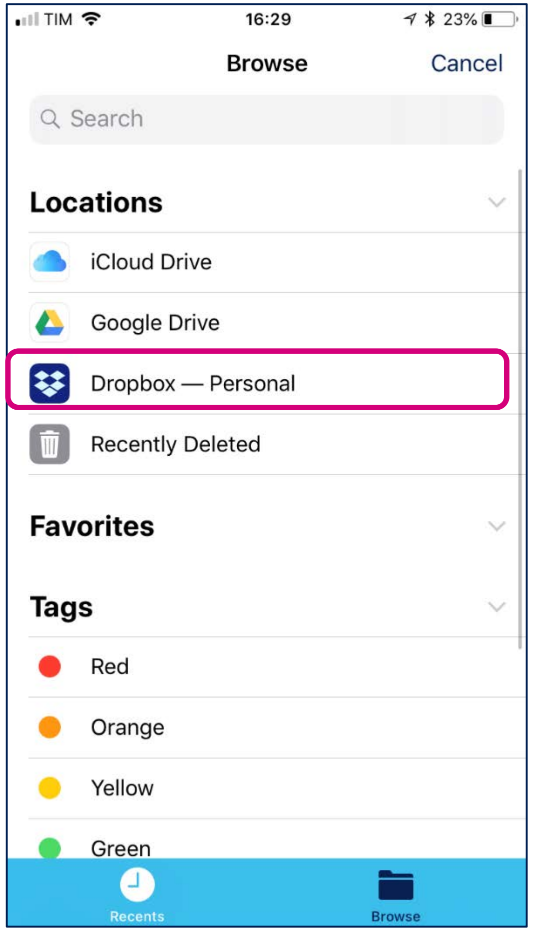
Click on the top right corner



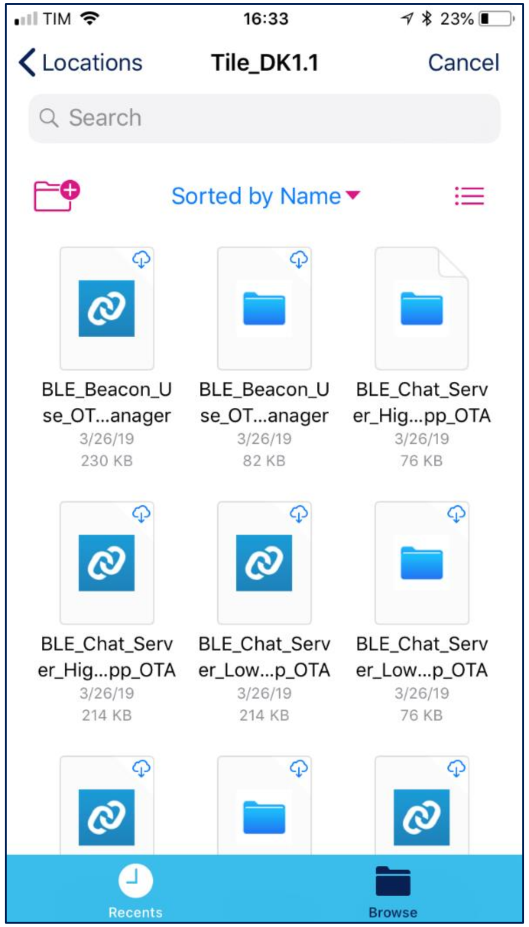
Select "Firmware Upgrade"



Select your remote repository



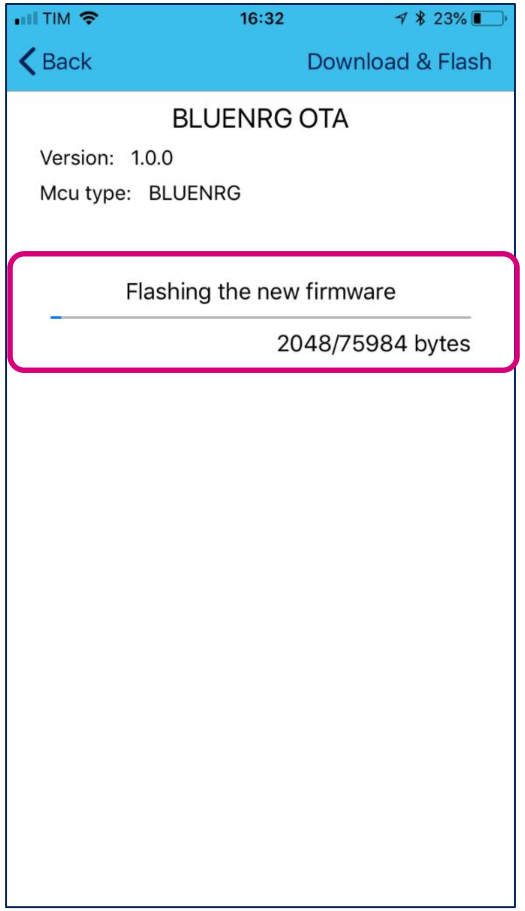
Select the binary file


TIM
16:33
23%
Locations
Tile_DK1.1
Cancel
Search
Sorted by Name
BLE_Beacon_Use_OT...anager
3/26/19
230 KB
BLE_Beacon_Use_OT...anager
3/26/19
82 KB
BLE_Chat_Server_Hig...pp_OTA
3/26/19
76 KB
BLE_Chat_Server_Hig...pp_OTA
3/26/19
214 KB
BLE_Chat_Server_Low...p_OTA
3/26/19
214 KB
BLE_Chat_Server_Low...p_OTA
3/26/19
76 KB
Recents
Browse

life.augmented

OTA FW Upgrade (2/2)

OTA data transfer begins



TIM 16:32 23%

Back Download & Flash

BLUENRG OTA

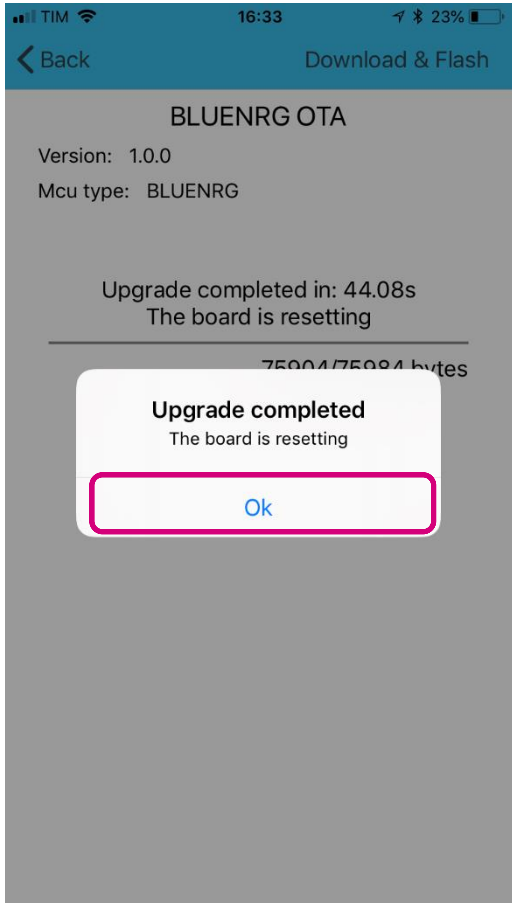
Version: 1.0.0

Mcu type: BLUENRG

Flashing the new firmware

2048/75984 bytes

Click OK as completed



TIM 16:33 23%

Back Download & Flash

BLUENRG OTA

Version: 1.0.0

Mcu type: BLUENRG

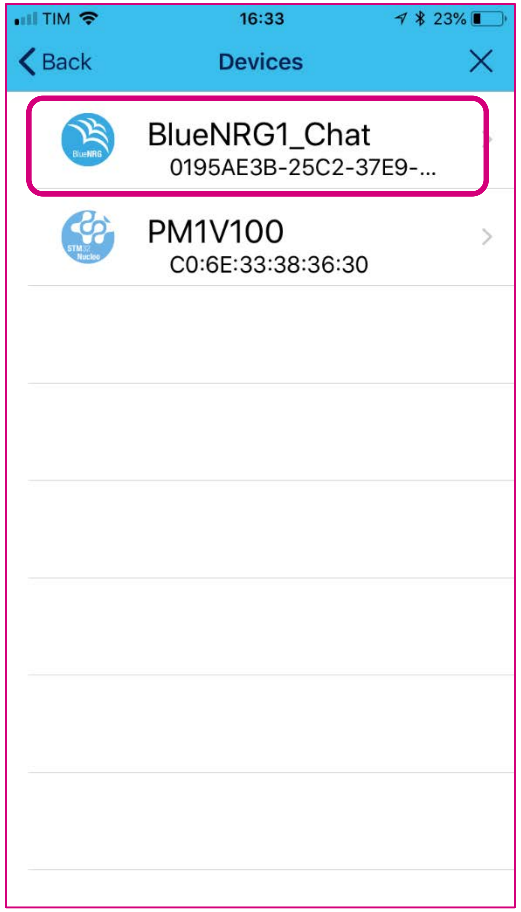
Upgrade completed in: 44.08s
The board is resetting

75984/75984 bytes

Upgrade completed

The board is resetting

Ok



TIM 16:33 23%

Back Devices ✕

BlueNRG1_Chat
0195AE3B-25C2-37E9-...

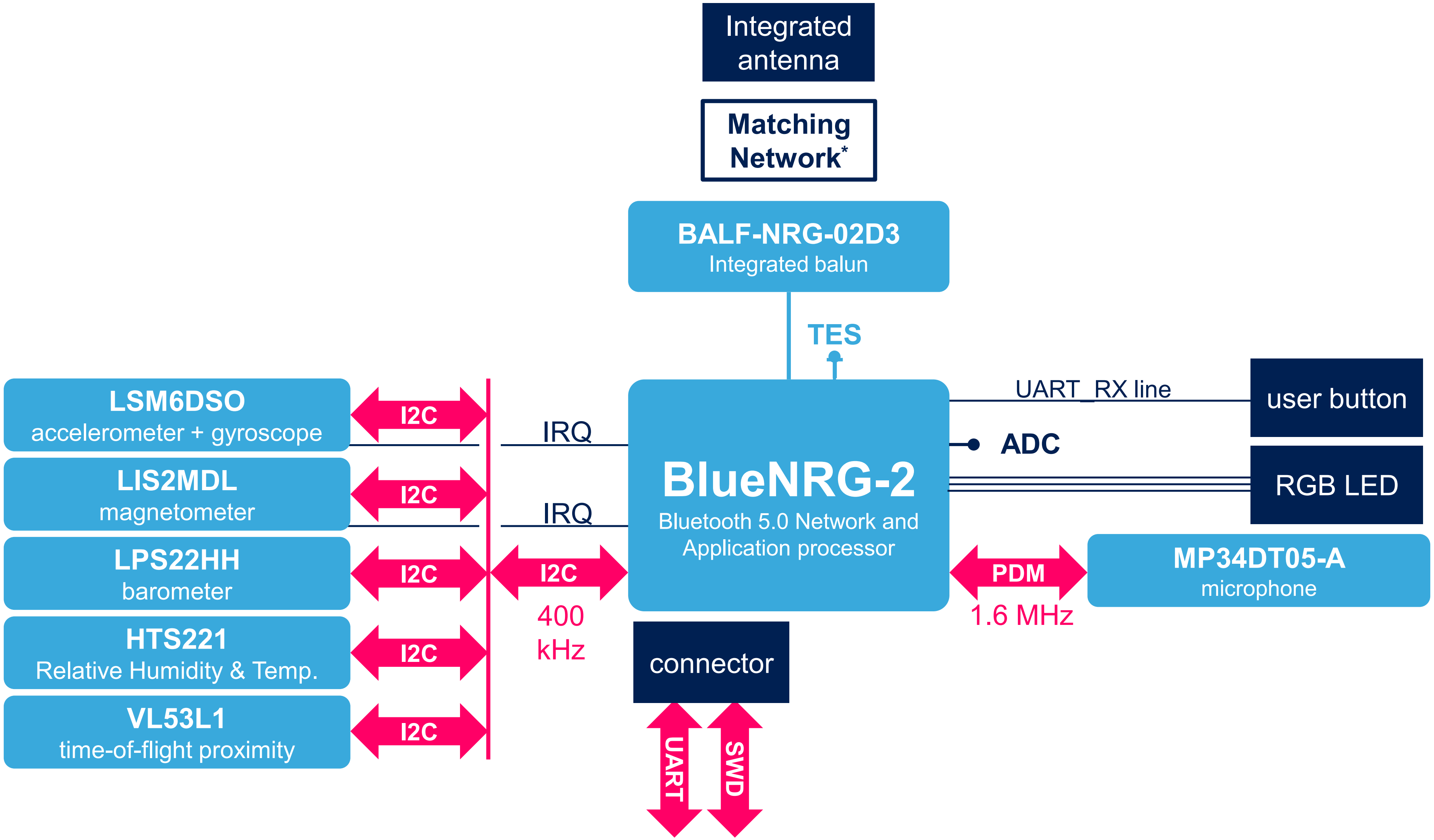
PM1V100
C0:6E:33:38:36:30

Re-Connect to BlueTile, now running a new application

Customizing Your Design

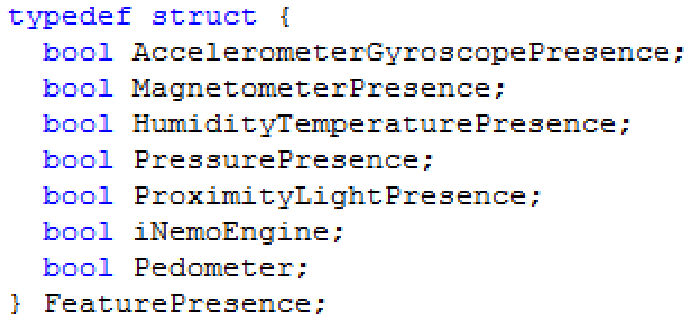
life.augmented

STEVAL-BCN002V1B Block Diagram



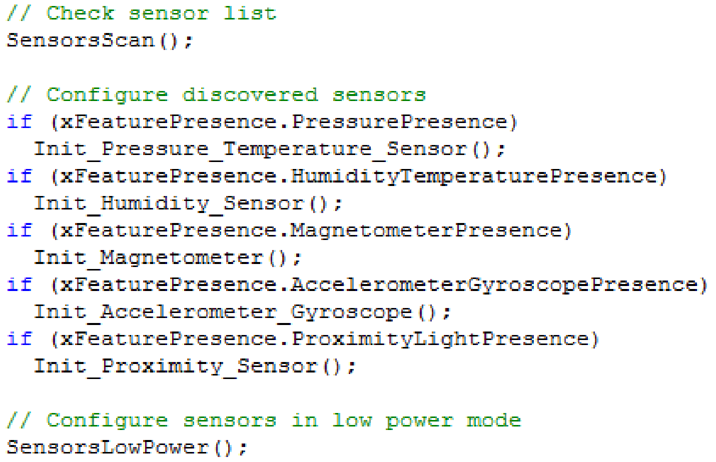
Enable/Disable Sensors & Libraries

- **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**

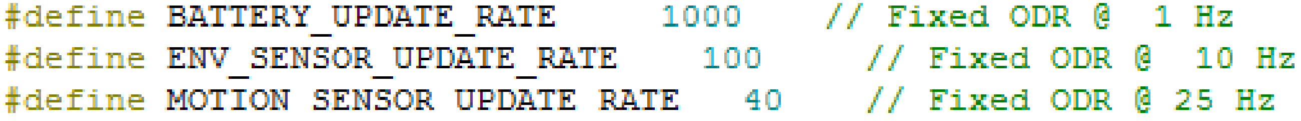


```
// Check sensor list
SensorsScan();

// 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();
```

Disable the unneeded sensor for optimizing the power consumption!

BLE Timings

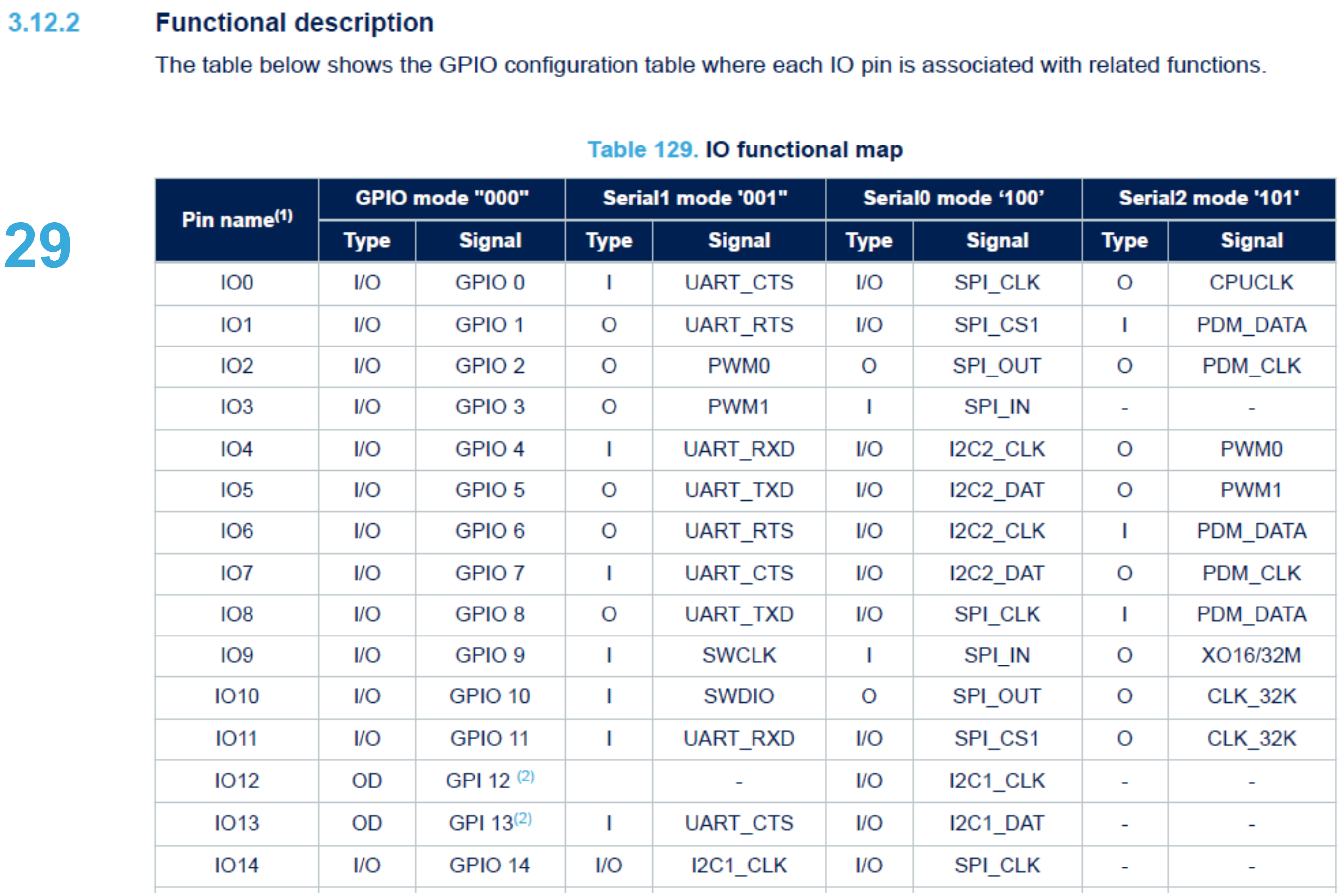
- **Advertising intervals**:
  - Dedicated API aci_gap_set_discoverable(Advertising_Type, Advertising_Interval_Min, Advertising_Interval_Max, …)
  - In file **sensor.c**
- **Connection intervals**:
  - Dictated by the Central device. Peripheral has no full control on this.
- **Notifications frequency**
  - 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
```

HW Modifications

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



3.12.2 **Functional description**

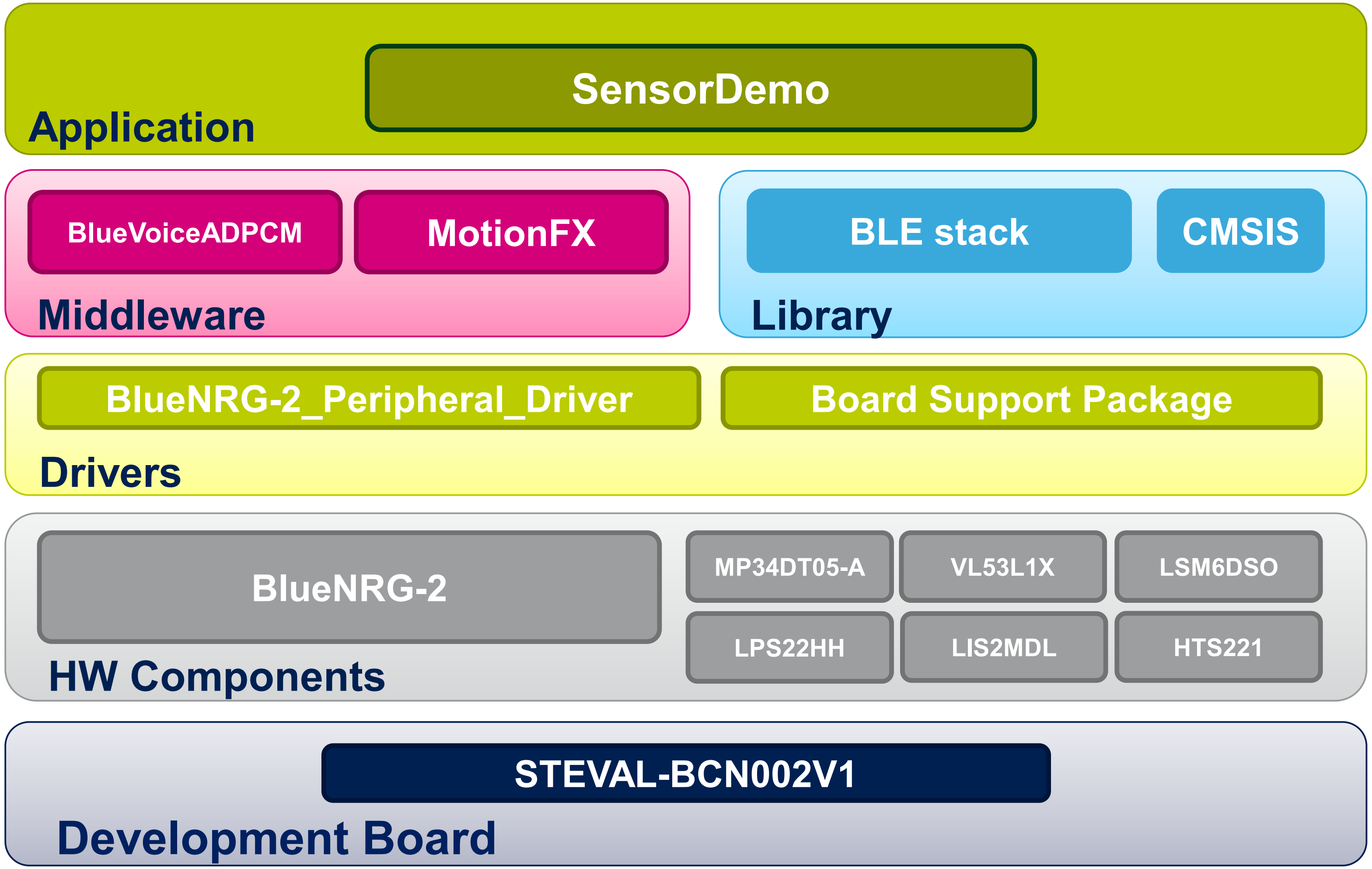
The table below shows the GPIO configuration table where each IO pin is associated with related functions.

Table 129. **IO functional map**

Pin name[1]	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	CPUCLK
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 [2]		-	I/O	I2C1_CLK	-	-
IO13	OD	GPI 13[2]	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



Application

SensorDemo

Middleware

BlueVoiceADPCM

MotionFX

Library

BLE stack

CMSIS

Drivers

BlueNRG-2_Peripheral_Driver

Board Support Package

HW Components

BlueNRG-2

MP34DT05-A

VL53L1X

LSM6DSO

LPS22HH

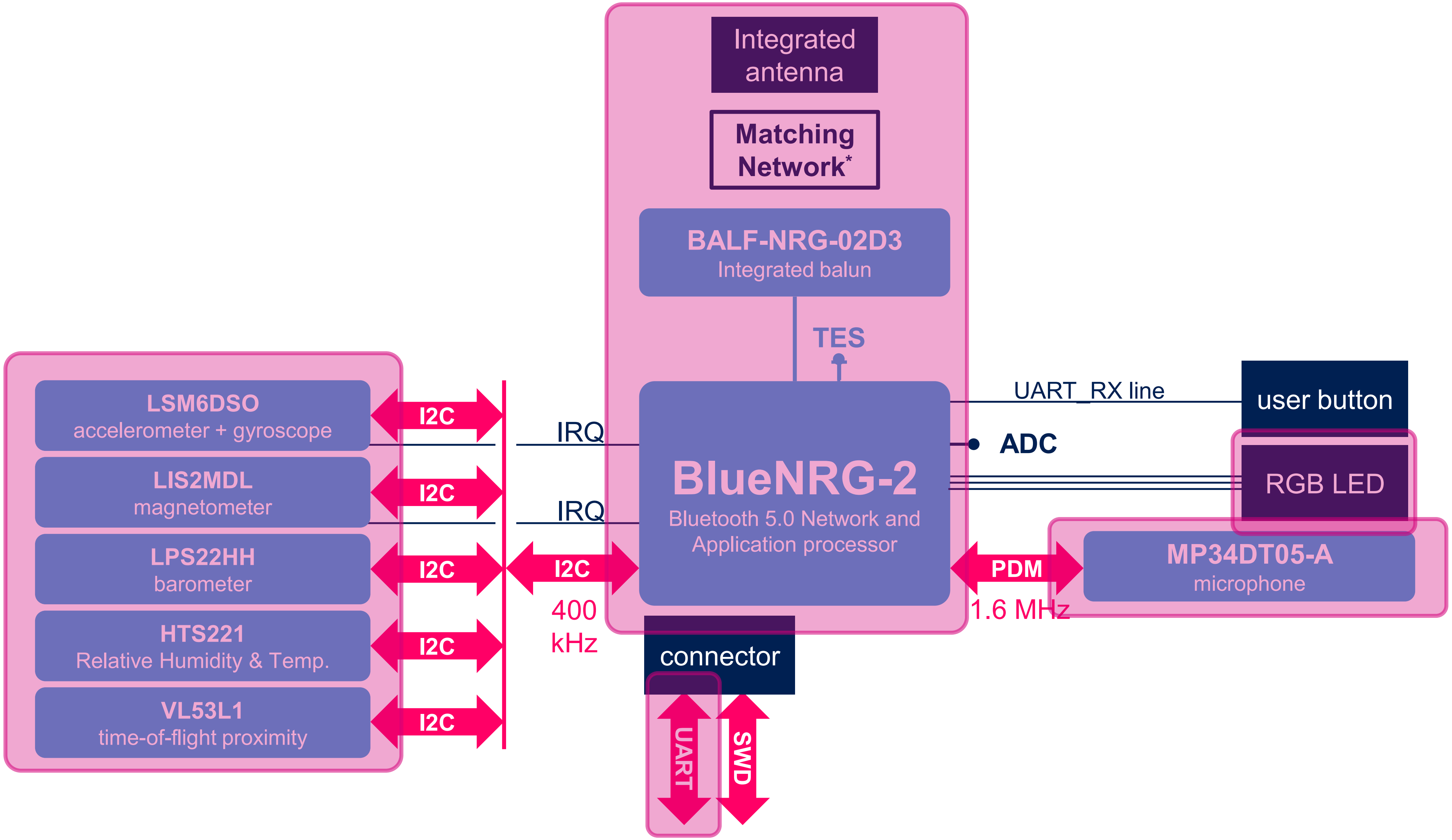
LIS2MDL

HTS221

Development Board

STEVAL-BCN002V1

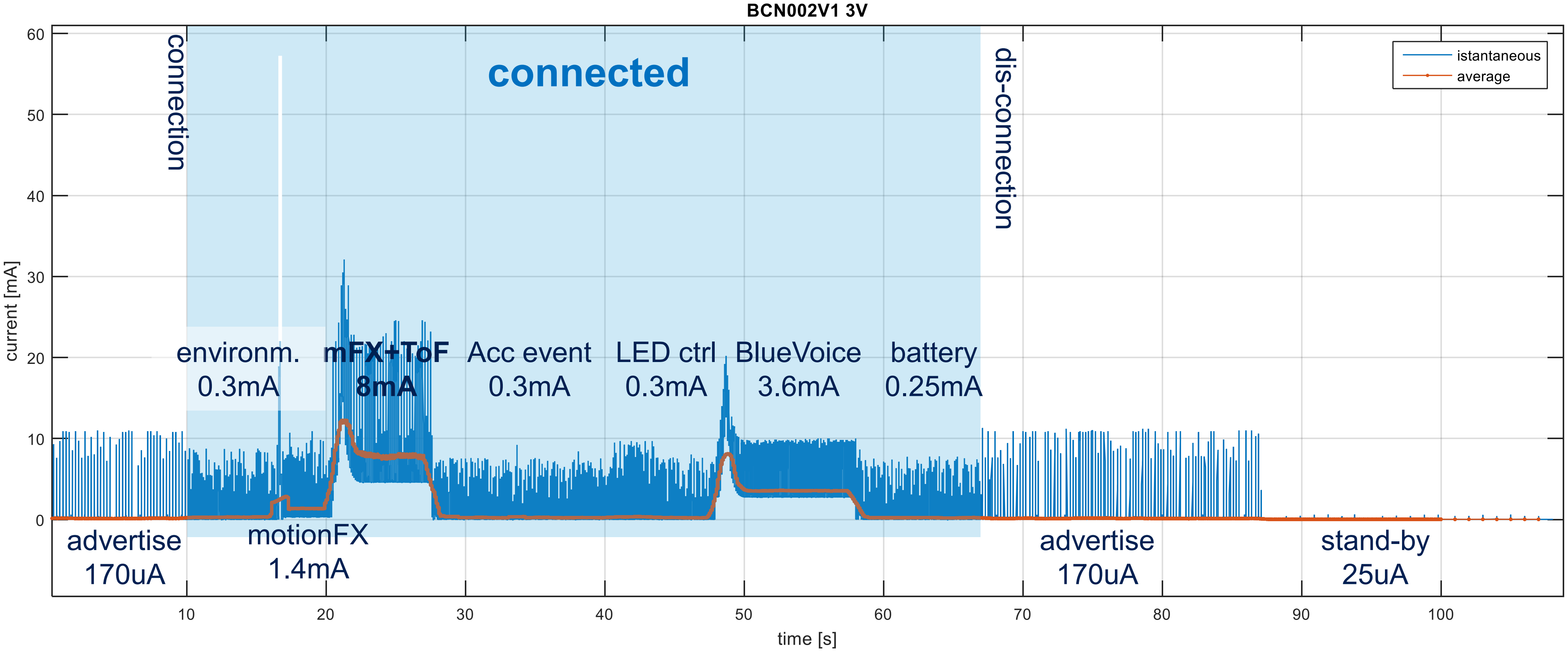
STEVAL-BCN002V1B Block Diagram



life.augmented

Power Consumption

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

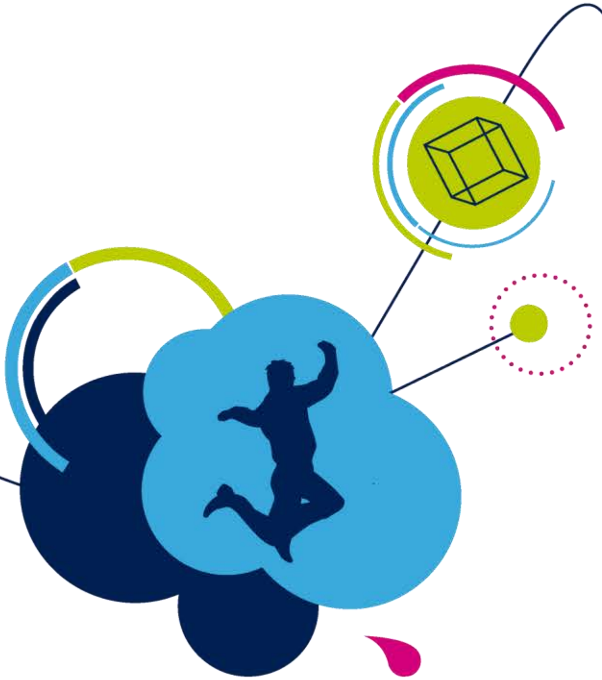


Battery Lifetime: CR2032

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

life.augmented

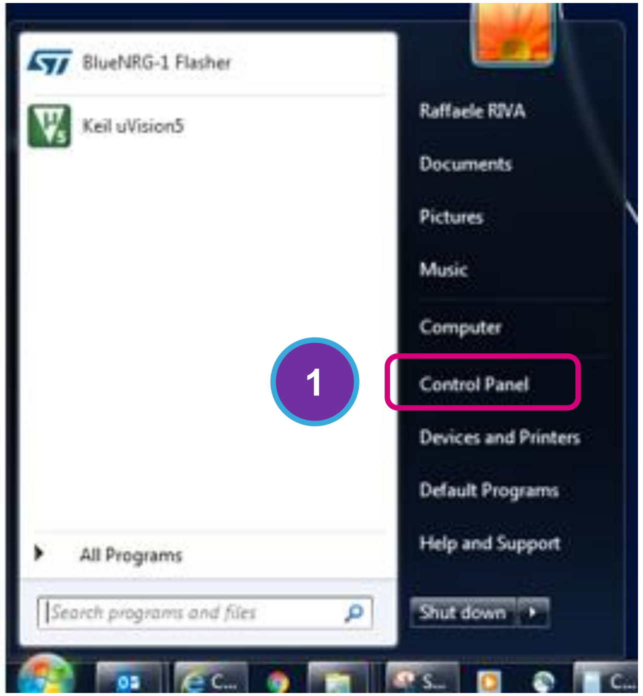


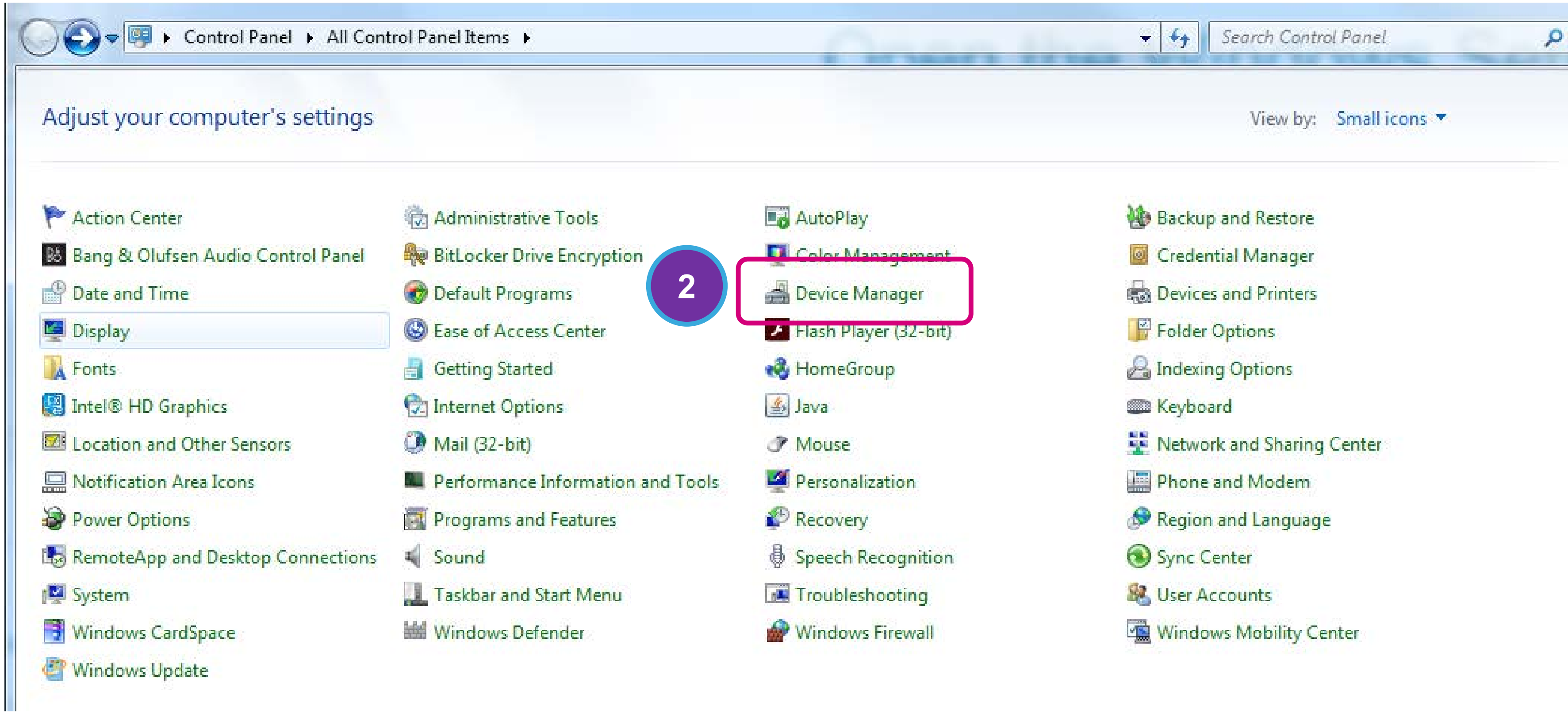
Virtual COM Driver Installation Win7

Open the Windows Control Panel

1. From **Start Menu** select **Control Panel**

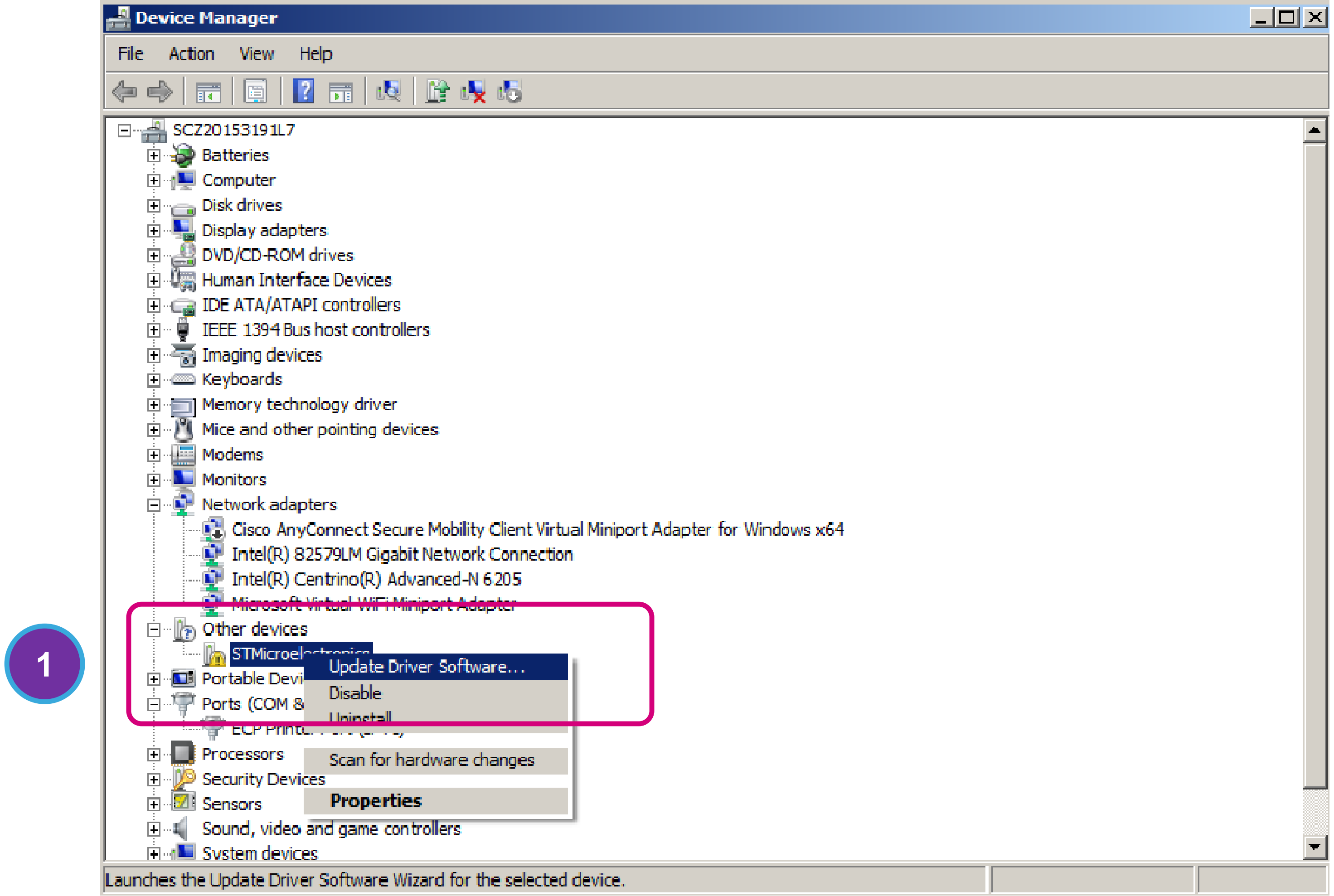
2. Select **Device Manager**





Open the Device Manager

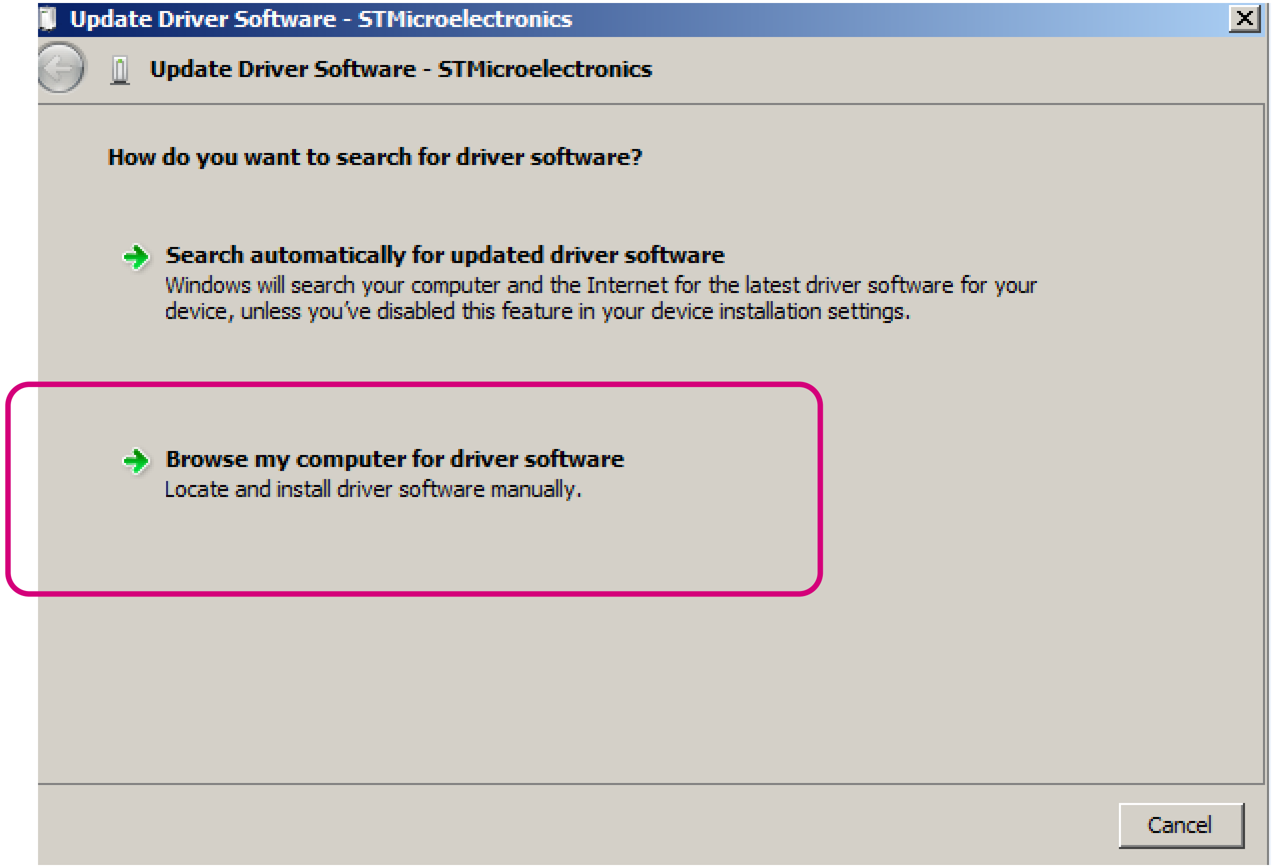
1. Look for **Other devices** and right click and then select **Update Driver Software…**



Launches the Update Driver Software Wizard for the selected device.

Look for the VCOM Driver

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



Update Driver Software - STMicroelectronics

Update Driver Software - STMicroelectronics

How do you want to search for driver software?

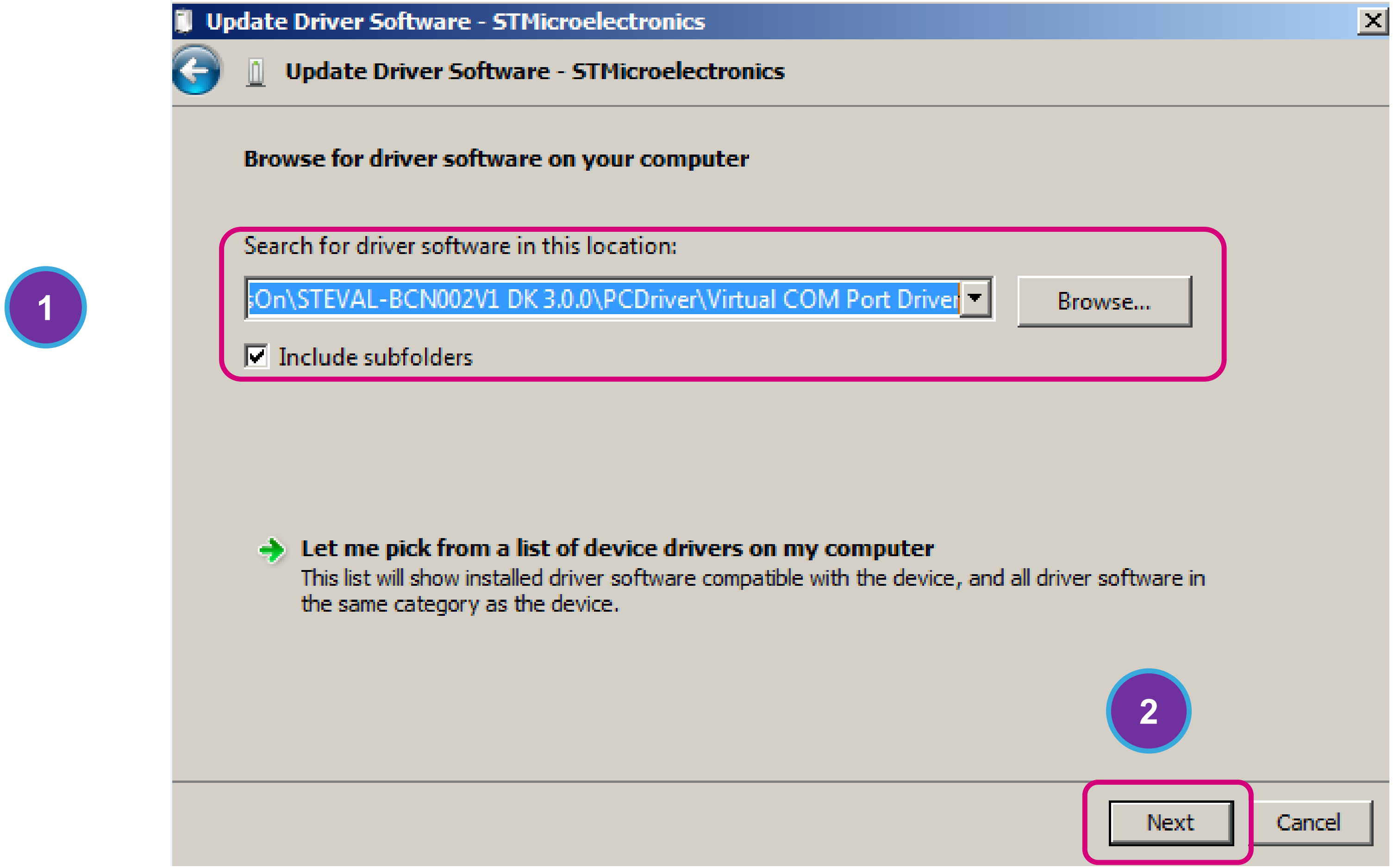
→ **Search automatically for updated driver software**
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

→ **Browse my computer for driver software**
Locate and install driver software manually.

Look for the VCOM Driver

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**



Update Driver Software - STMicroelectronics

Update Driver Software - STMicroelectronics

Browse for driver software on your computer

Search for driver software in this location:

:On\STEVAL-BCN002V1 DK 3.0.0\PCDriver\Virtual COM Port Driver

☑ Include subfolders

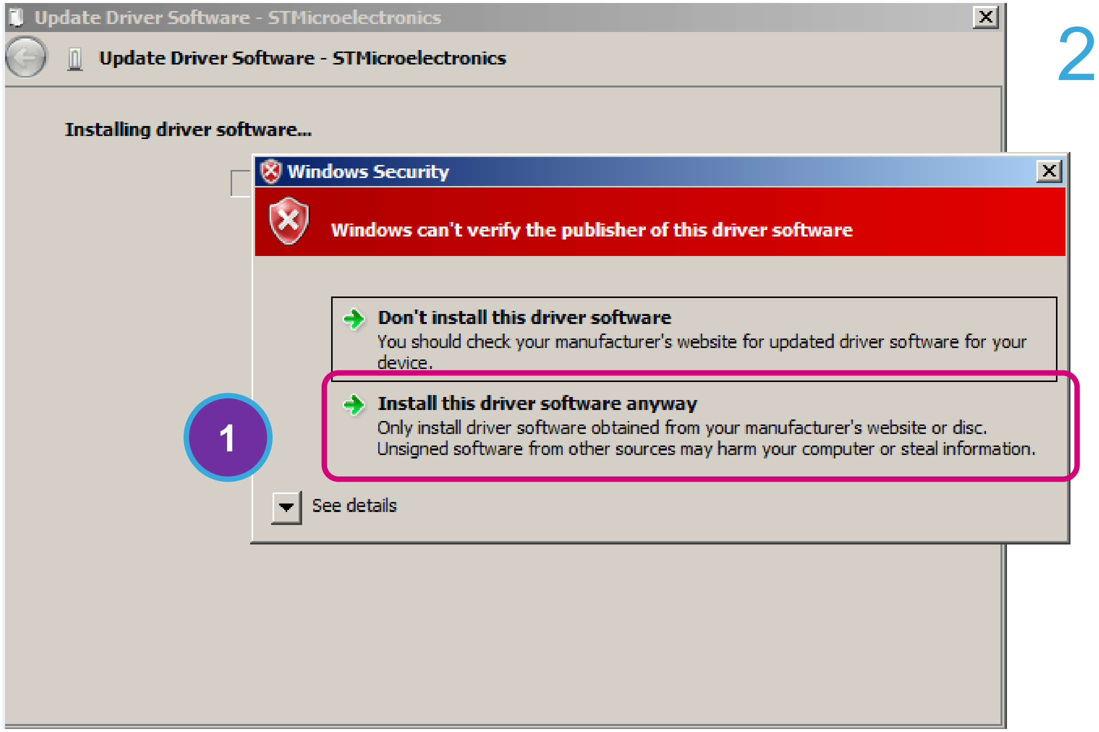
Let me pick from a list of device drivers on my computer

This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

Allow the Driver Installation

1. Click on **Install driver software anyway**

2. Installation starts



Update Driver Software - STMicroelectronics

Update Driver Software - STMicroelectronics

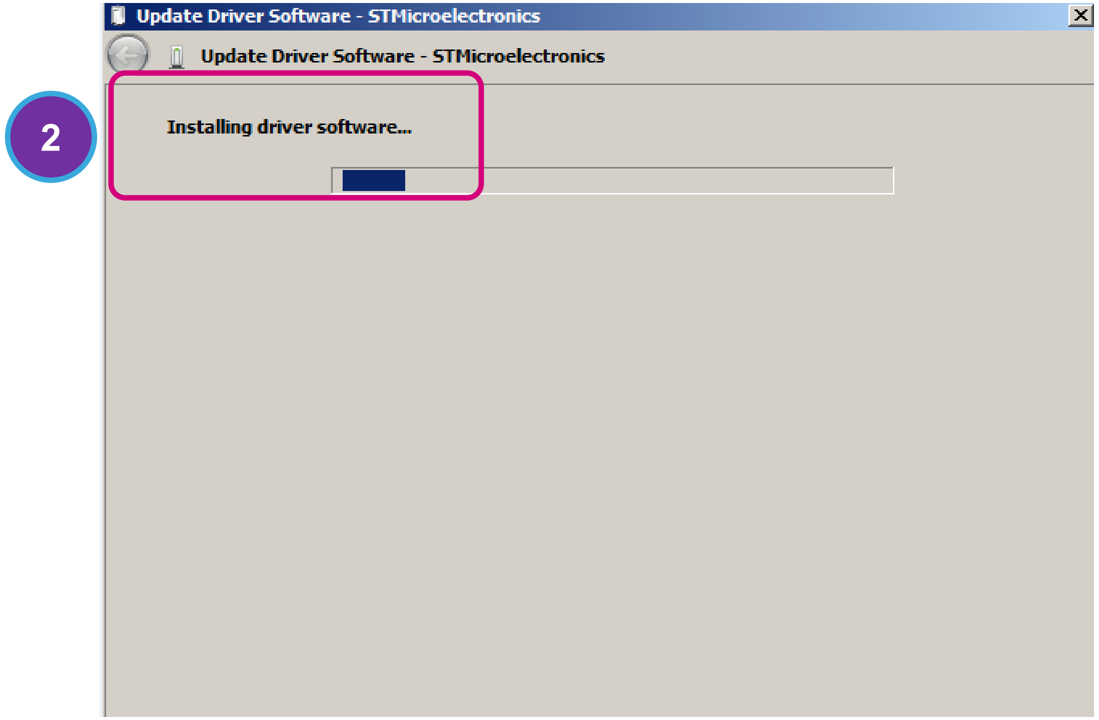
Installing driver software...

Windows Security

Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.



Update Driver Software - STMicroelectronics

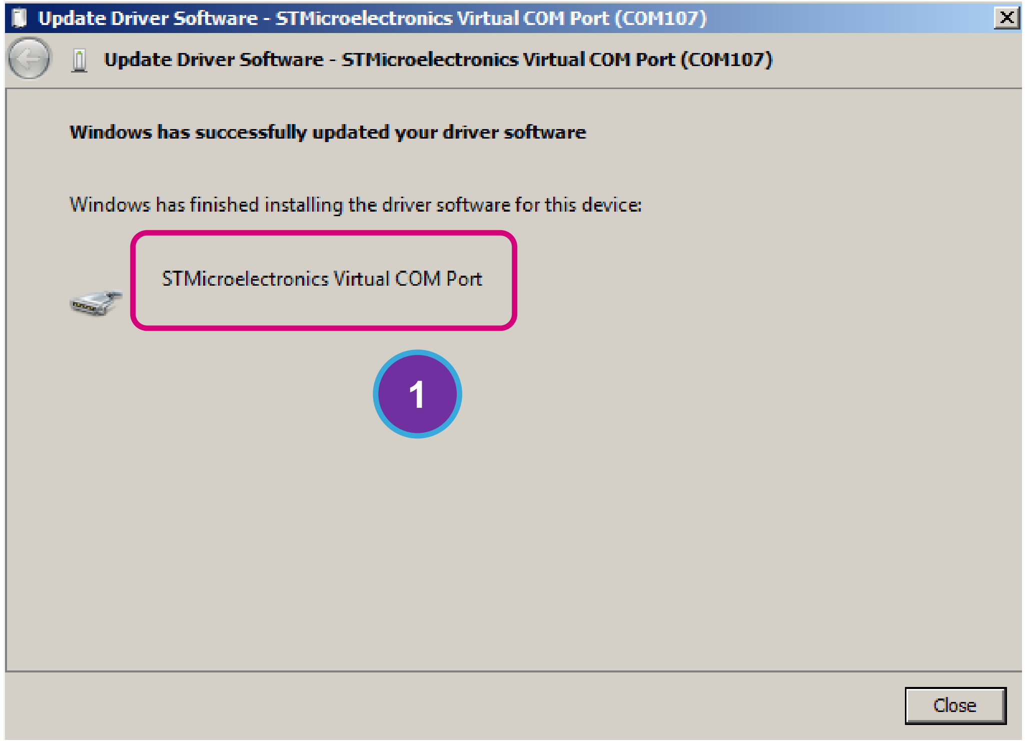
Update Driver Software - STMicroelectronics

Installing driver software...

Done

1. Installation completed

2. Device is in the **COM Ports** list



Update Driver Software - STMicroelectronics Virtual COM Port (COM107)

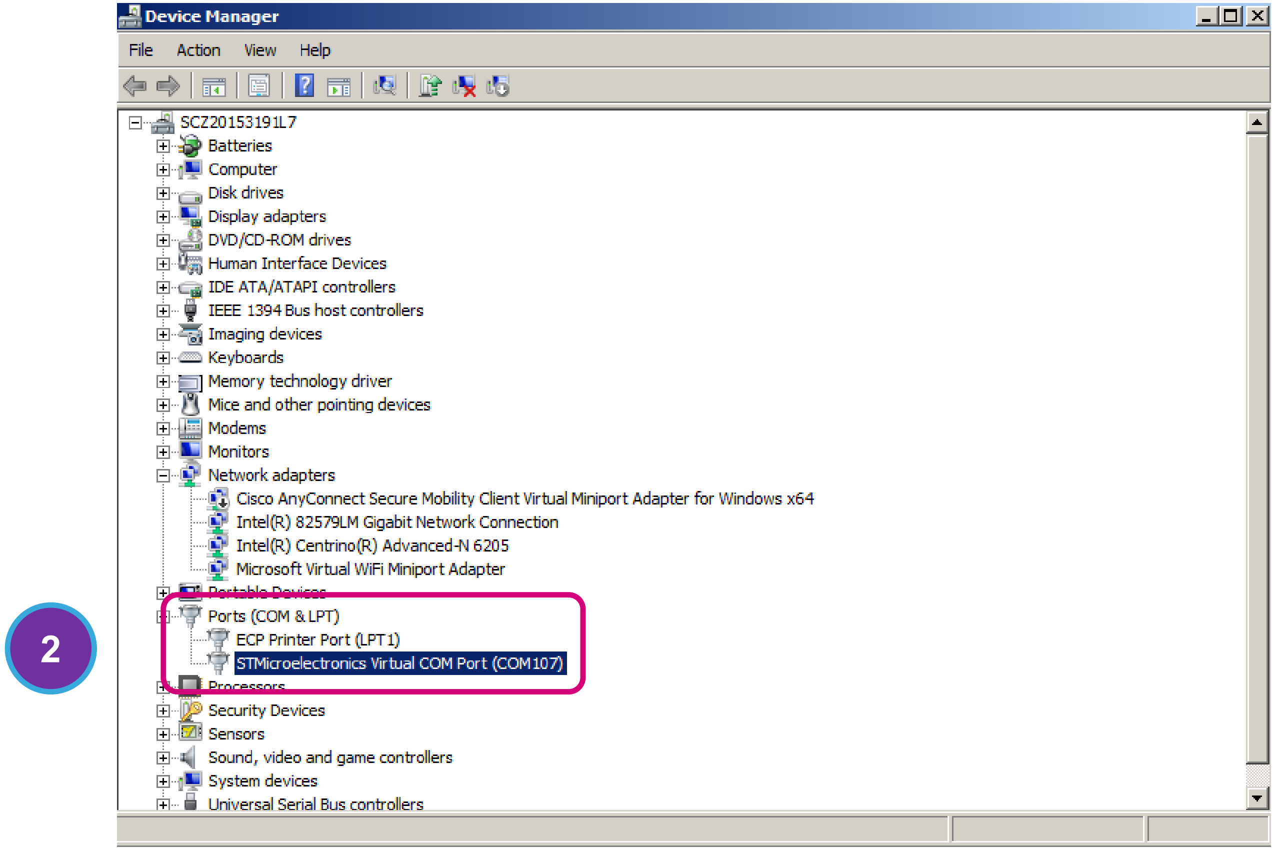
Update Driver Software - STMicroelectronics Virtual COM Port (COM107)

Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

STMicroelectronics Virtual COM Port

Done! Now please go back from here!



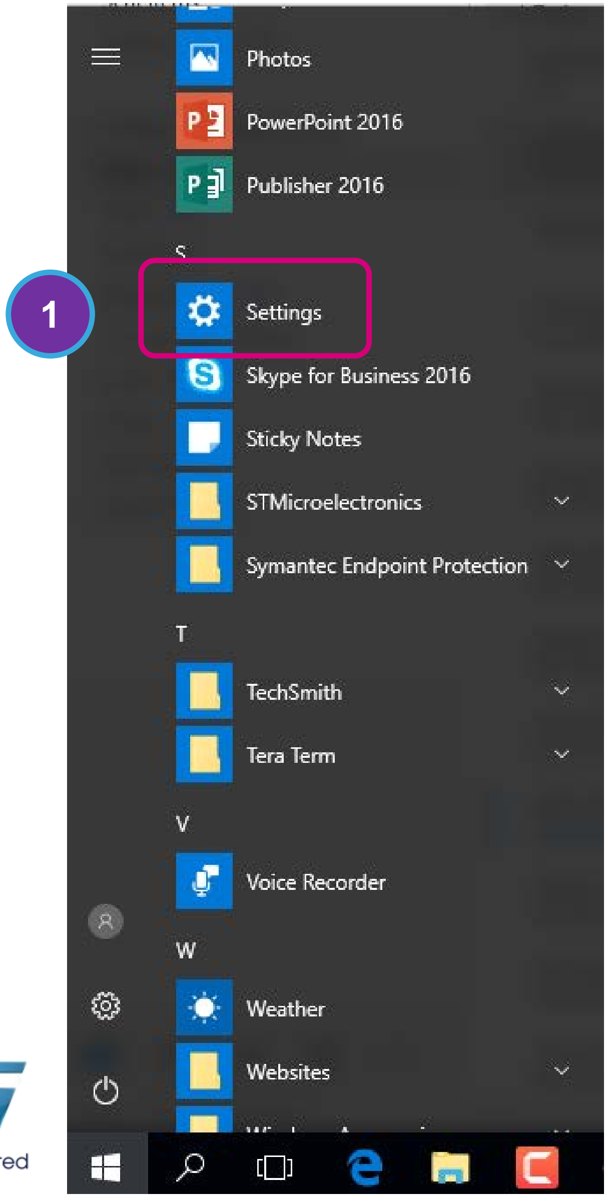
SCZ20153191L7
- Batteries
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 Bus host controllers
- Imaging devices
- Keyboards
- Memory technology driver
- Mice and other pointing devices
- Modems
- Monitors
- Network adapters
  - Cisco AnyConnect Secure Mobility Client Virtual Miniport Adapter for Windows x64
  - Intel(R) 82579LM Gigabit Network Connection
  - Intel(R) Centrino(R) Advanced-N 6205
  - Microsoft Virtual WiFi Miniport Adapter
- Portable Devices
- Ports (COM & LPT)
  - ECP Printer Port (LPT1)
  - STMicroelectronics Virtual COM Port (COM107)
- Processors
- Security Devices
- Sensors
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers

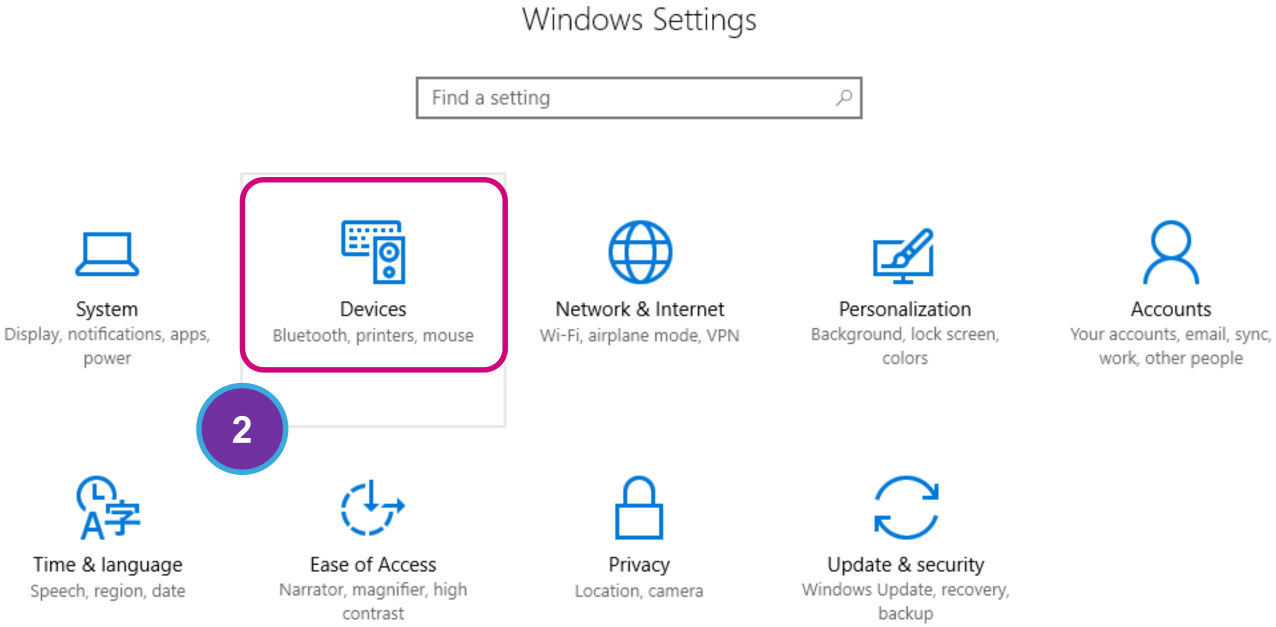
Virtual COM Driver Installation Win10

Open the Windows Settings

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

2. Select **Devices**




Windows Settings

Find a setting

System
Display, notifications, apps, power

Devices
Bluetooth, printers, mouse

Network & Internet
Wi-Fi, airplane mode, VPN

Personalization
Background, lock screen, colors

Accounts
Your accounts, email, sync, work, other people

Time & language
Speech, region, date

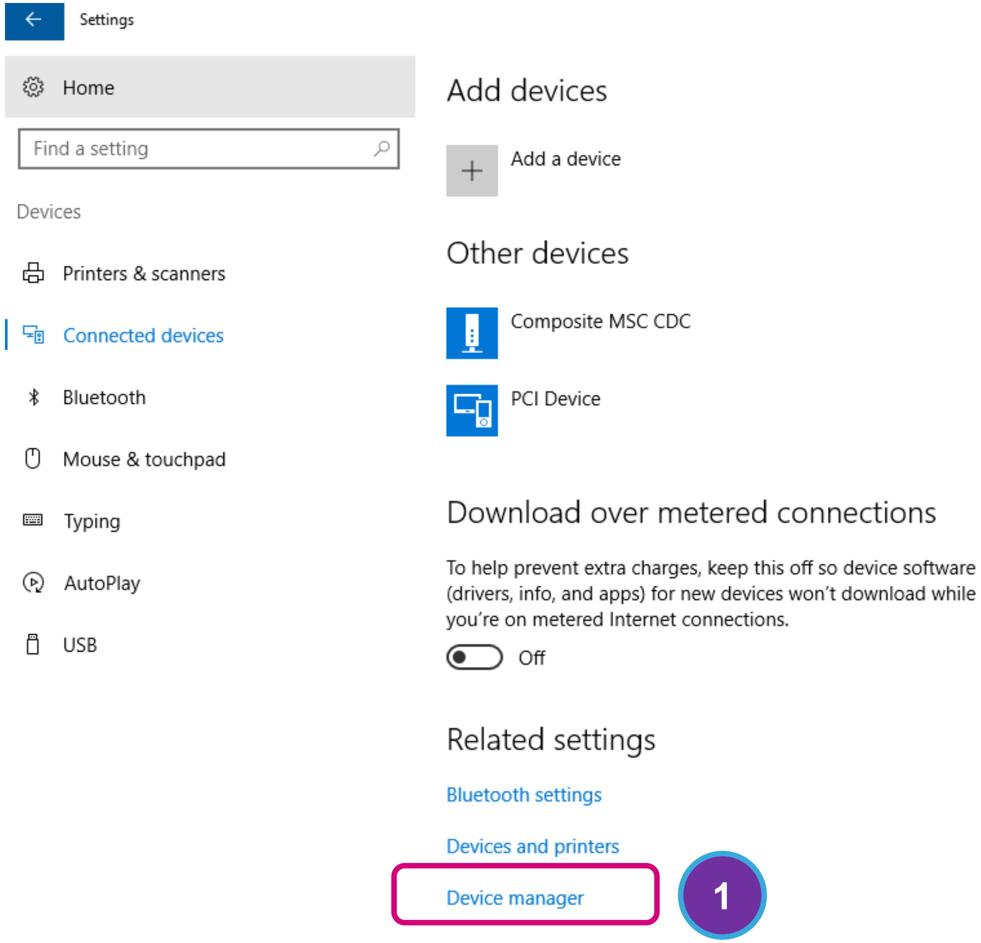
Ease of Access
Narrator, magnifier, high contrast

Privacy
Location, camera

Update & security
Windows Update, recovery, backup

Open the Device Manager

1. Select **Device manager**
2. Select **Other Devices** and **Unknown device**



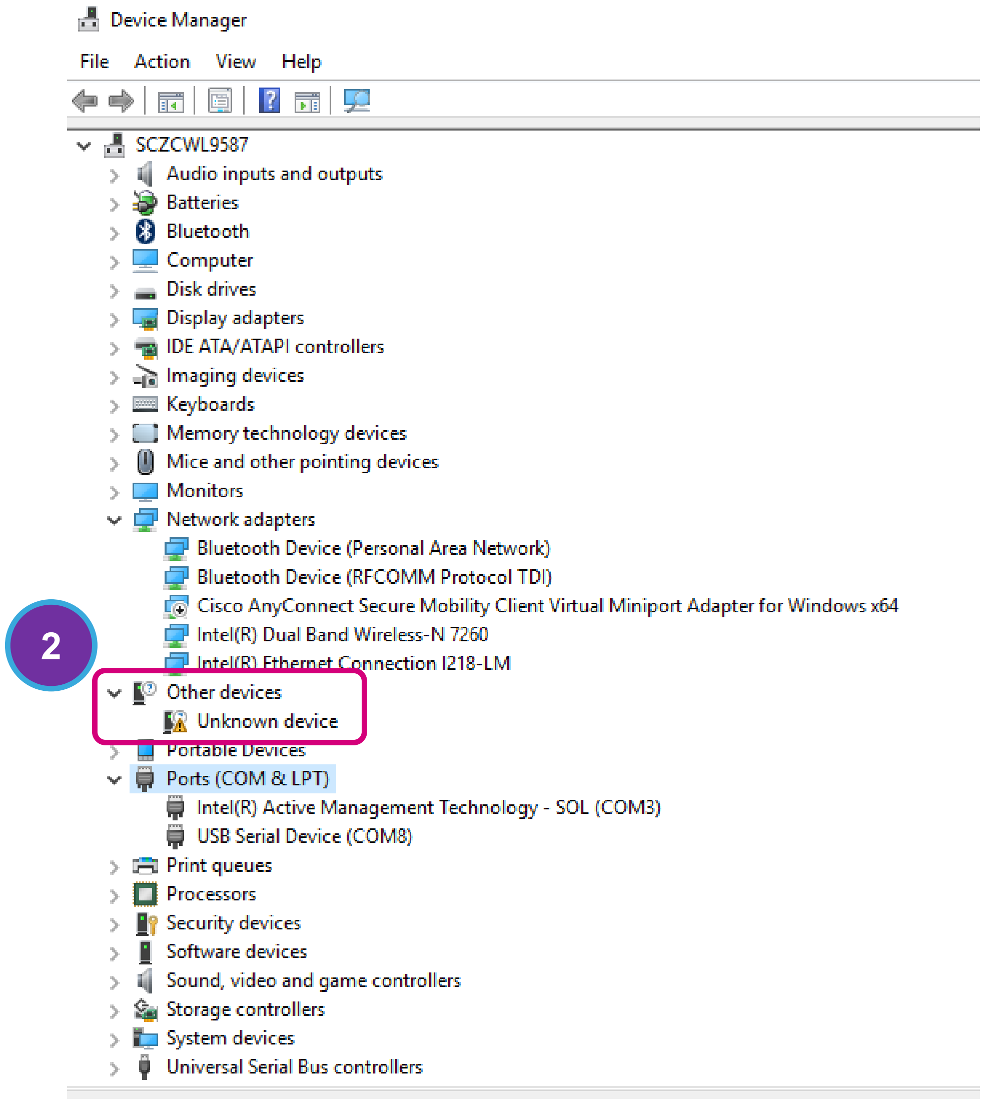
Add devices

Other devices

Download over metered connections

To help prevent extra charges, keep this off so device software (drivers, info, and apps) for new devices won't download while you're on metered Internet connections.

Related settings

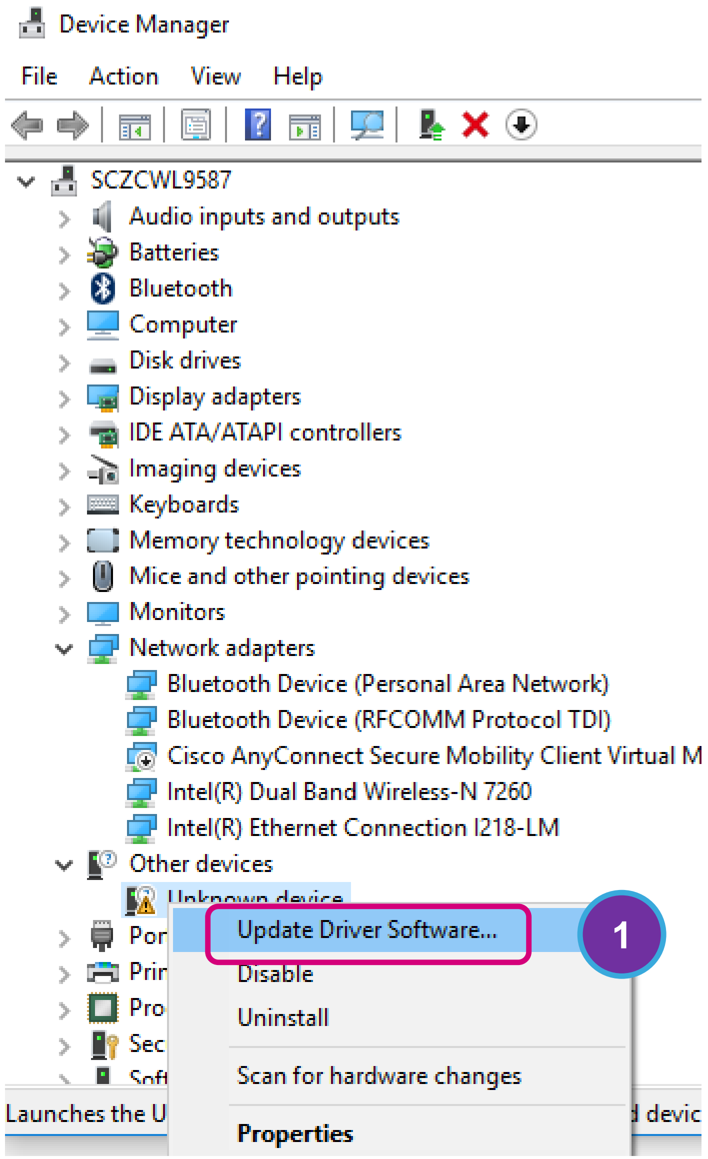


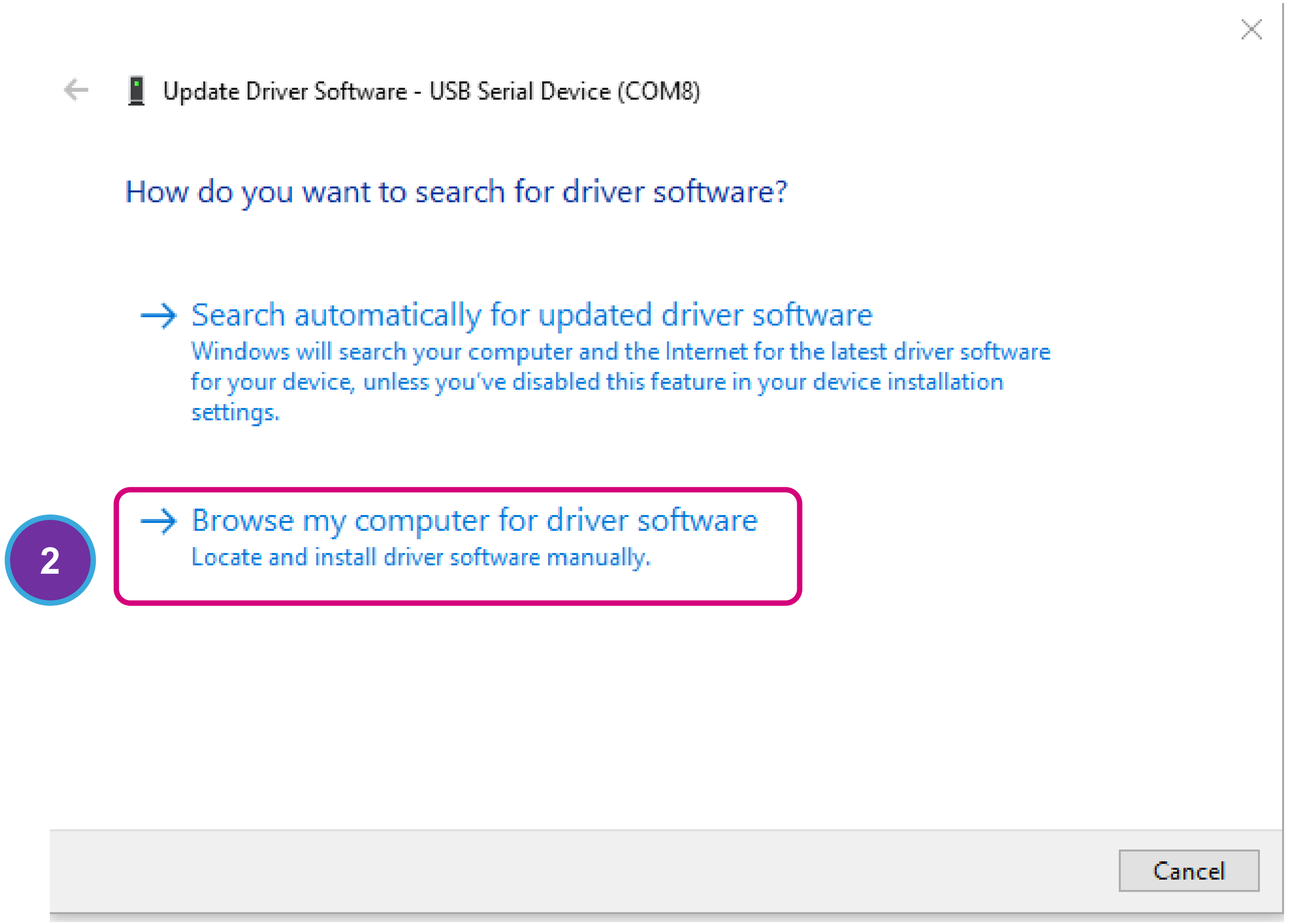
- SCZCWL9587
  - Audio inputs and outputs
  - Batteries
  - Bluetooth
  - Computer
  - Disk drives
  - Display adapters
  - IDE ATA/ATAPI controllers
  - Imaging devices
  - Keyboards
  - Memory technology devices
  - Mice and other pointing devices
  - Monitors
  - Network adapters
    - Bluetooth Device (Personal Area Network)
    - Bluetooth Device (RFCOMM Protocol TDI)
    - Cisco AnyConnect Secure Mobility Client Virtual Miniport Adapter for Windows x64
    - Intel(R) Dual Band Wireless-N 7260
    - Intel(R) Ethernet Connection I218-LM
  - Other devices
    - Unknown device
  - Portable Devices
  - Ports (COM & LPT)
    - Intel(R) Active Management Technology - SOL (COM3)
    - USB Serial Device (COM8)
  - Print queues
  - Processors
  - Security devices
  - Software devices
  - Sound, video and game controllers
  - Storage controllers
  - System devices
  - Universal Serial Bus controllers

Update Driver Software

1. Right click and then select **Update Driver Software…**

2. Select **Browse my computer for driver software**


Device Manager
File
Action
View
Help
SCZCWL9587
Audio inputs and outputs
Batteries
Bluetooth
Computer
Disk drives
Display adapters
IDE ATA/ATAPI controllers
Imaging devices
Keyboards
Memory technology devices
Mice and other pointing devices
Monitors
Network adapters
Bluetooth Device (Personal Area Network)
Bluetooth Device (RFCOMM Protocol TDI)
Cisco AnyConnect Secure Mobility Client Virtual M
Intel(R) Dual Band Wireless-N 7260
Intel(R) Ethernet Connection I218-LM
Other devices
Unknown device
Por
Prir
Pro
Sec
Soft
Update Driver Software...
Disable
Uninstall
Scan for hardware changes
Properties
1
Launches the U
d devic


Update Driver Software - USB Serial Device (COM8)

How do you want to search for driver software?

→ Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

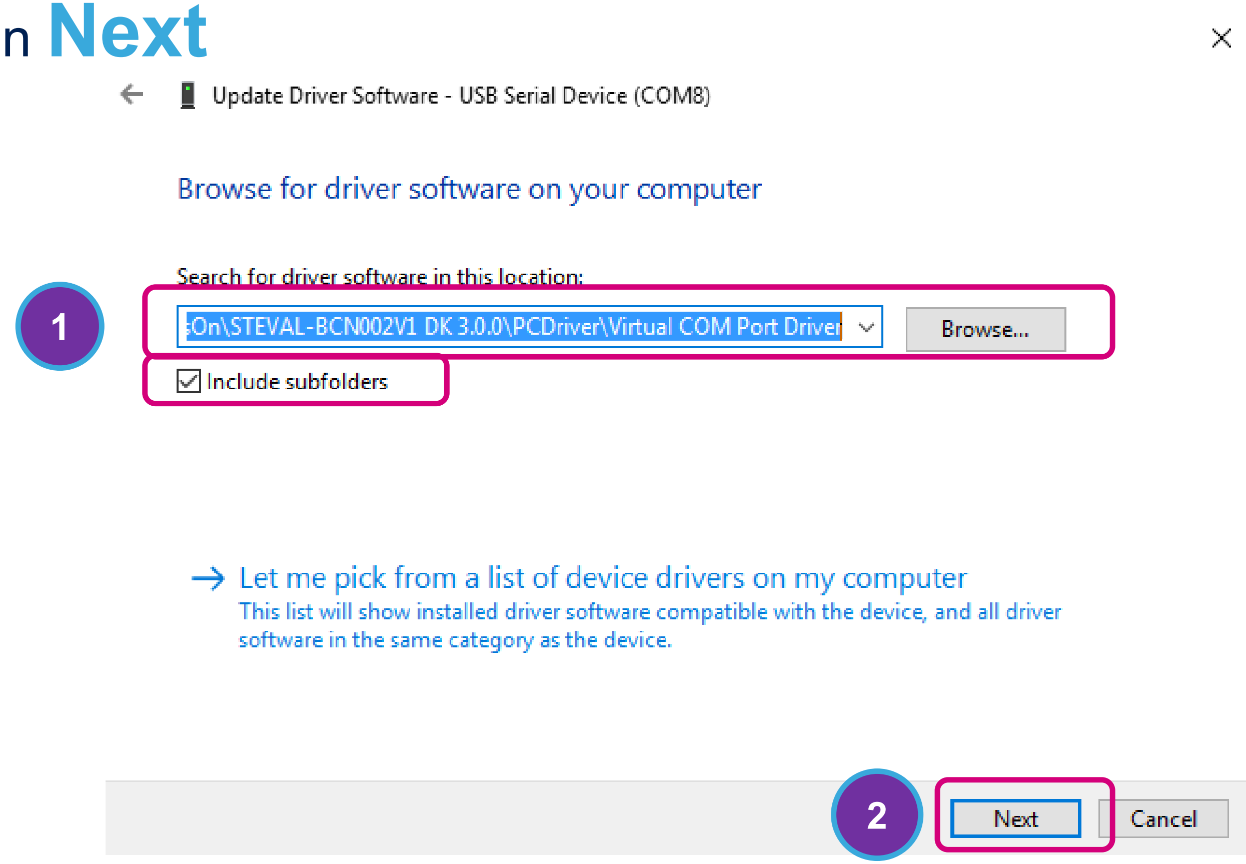
→ Browse my computer for driver software
Locate and install driver software manually.

Look for the VCOM Driver

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**



Update Driver Software - USB Serial Device (COM8)

Browse for driver software on your computer

Search for driver software in this location:

On\STEVAL-BCN002V1 DK 3.0.0\PCDriver\Virtual COM Port Driver

☑ Include subfolders

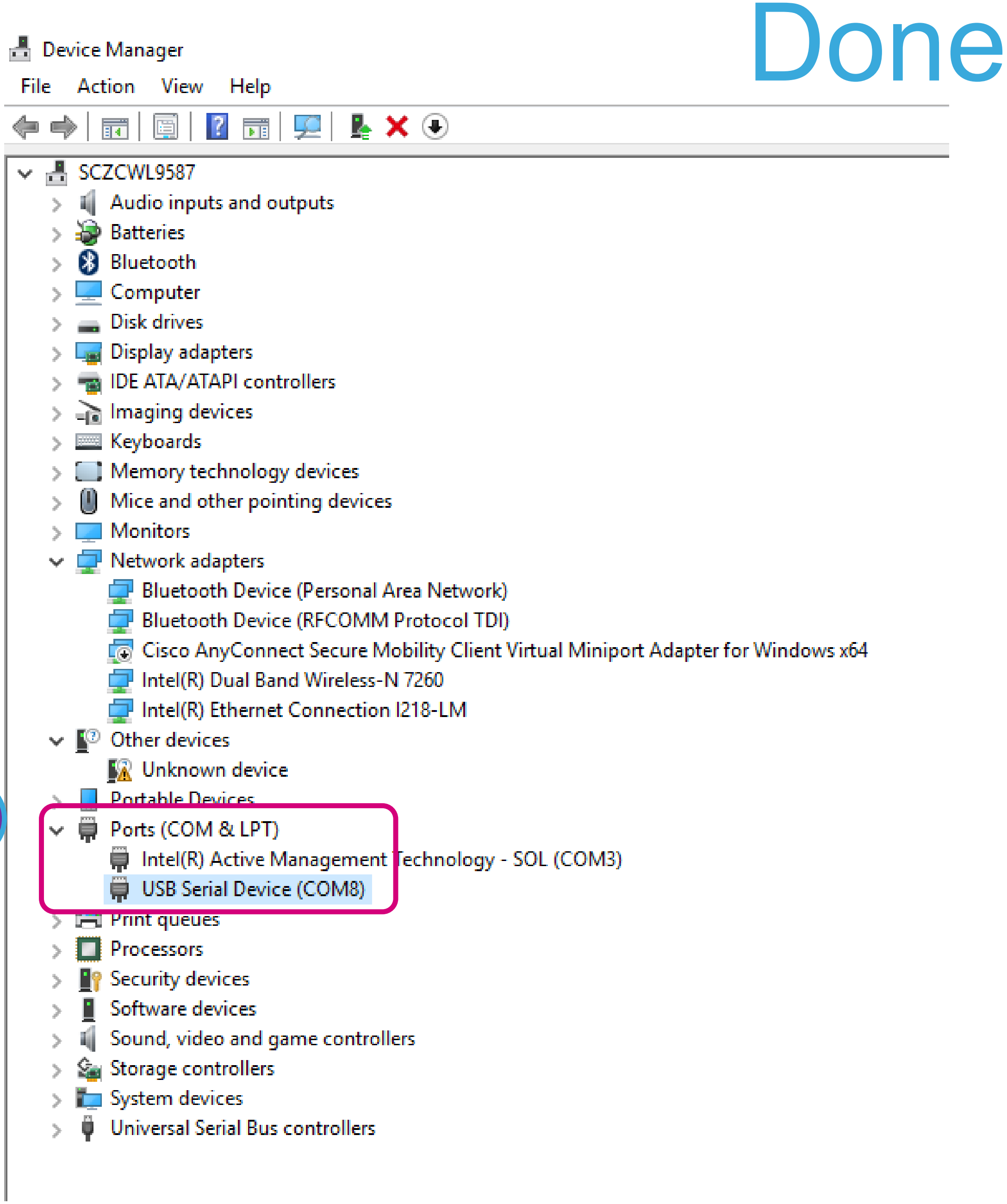
→ Let me pick from a list of device drivers on my computer

This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

Done

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

Done! Now please go back from here!



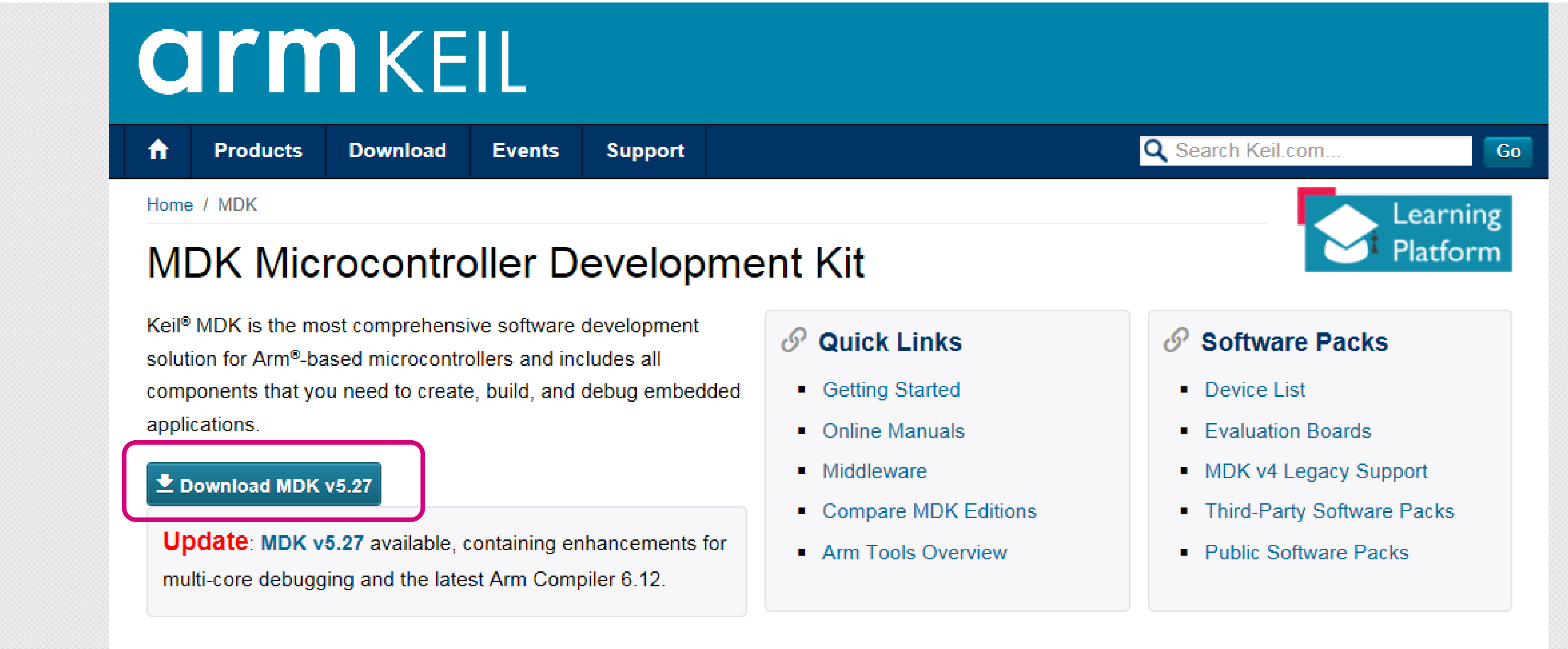
Device Manager

SCZCWL9587
- Audio inputs and outputs
- Batteries
- Bluetooth
- Computer
- Disk drives
- Display adapters
- IDE ATA/ATAPI controllers
- Imaging devices
- Keyboards
- Memory technology devices
- Mice and other pointing devices
- Monitors
- Network adapters
  - Bluetooth Device (Personal Area Network)
  - Bluetooth Device (RFCOMM Protocol TDI)
  - Cisco AnyConnect Secure Mobility Client Virtual Miniport Adapter for Windows x64
  - Intel(R) Dual Band Wireless-N 7260
  - Intel(R) Ethernet Connection I218-LM
- Other devices
  - Unknown device
- Portable Devices
- Ports (COM & LPT)
  - Intel(R) Active Management Technology - SOL (COM3)
  - USB Serial Device (COM8)
- Print queues
- Processors
- Security devices
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

ARM Keil MDK Installation

Keil ARM MDK Installation

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



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

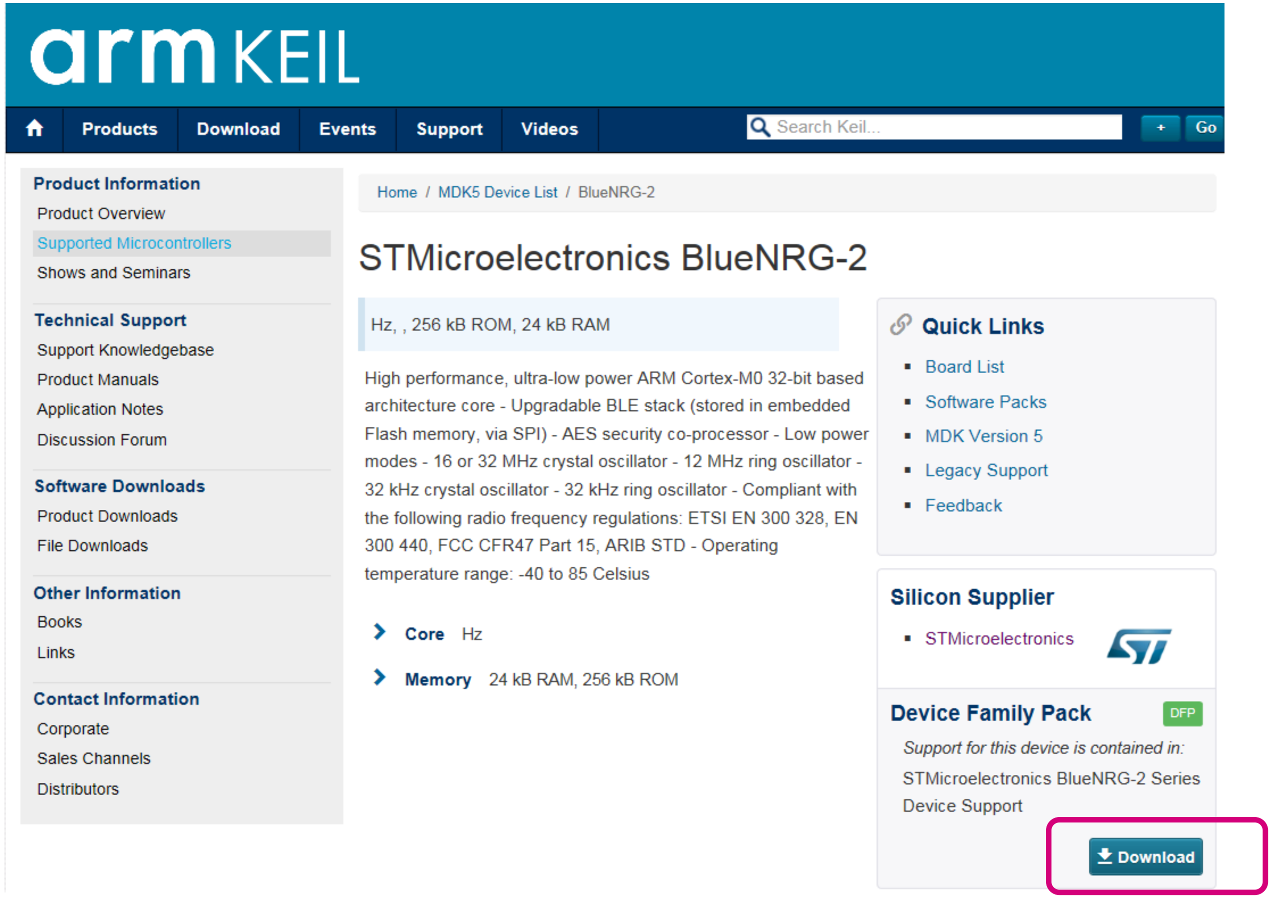
Software Packs

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

Arm Keil MDK Configuration

Keil ARM MDK Configuration

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/



Product Information
- Product Overview
- Supported Microcontrollers
- Shows and Seminars

Technical Support
- Support Knowledgebase
- Product Manuals
- Application Notes
- Discussion Forum

Software Downloads
- Product Downloads
- File Downloads

Other Information
- Books
- Links

Contact Information
- Corporate
- Sales Channels
- Distributors

Home / MDK5 Device List / BlueNRG-2

STMicroelectronics BlueNRG-2

Hz, , 256 kB ROM, 24 kB RAM

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

Core Hz

Memory 24 kB RAM, 256 kB ROM

Quick Links
- Board List
- Software Packs
- MDK Version 5
- Legacy Support
- Feedback

Silicon Supplier
- STMicroelectronics

Device Family Pack DFP

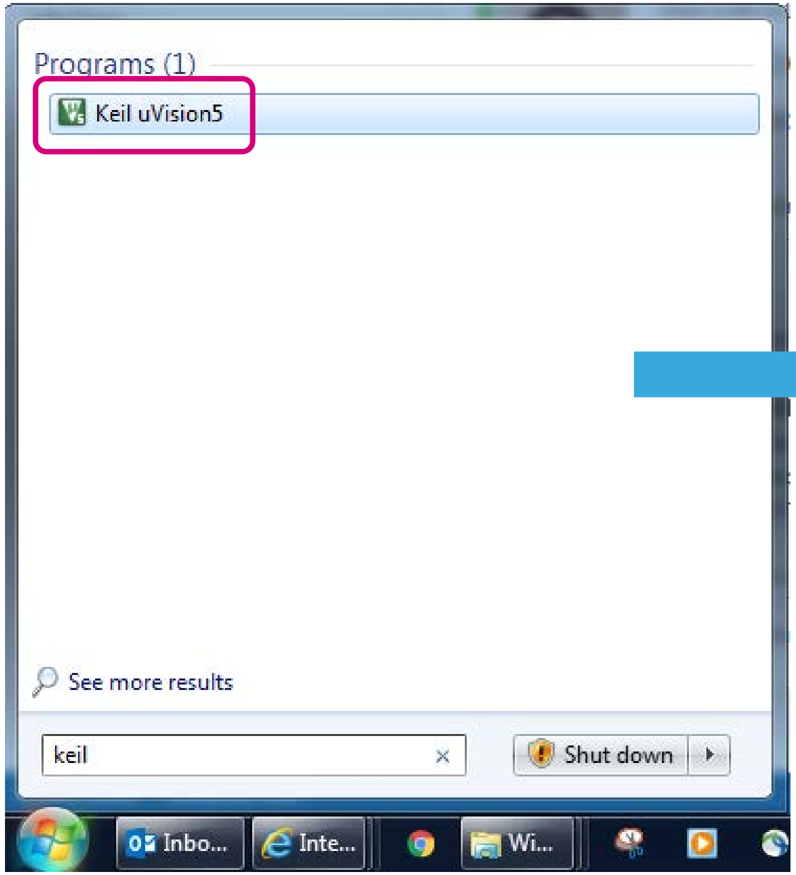
Support for this device is contained in:

STMicroelectronics BlueNRG-2 Series Device Support

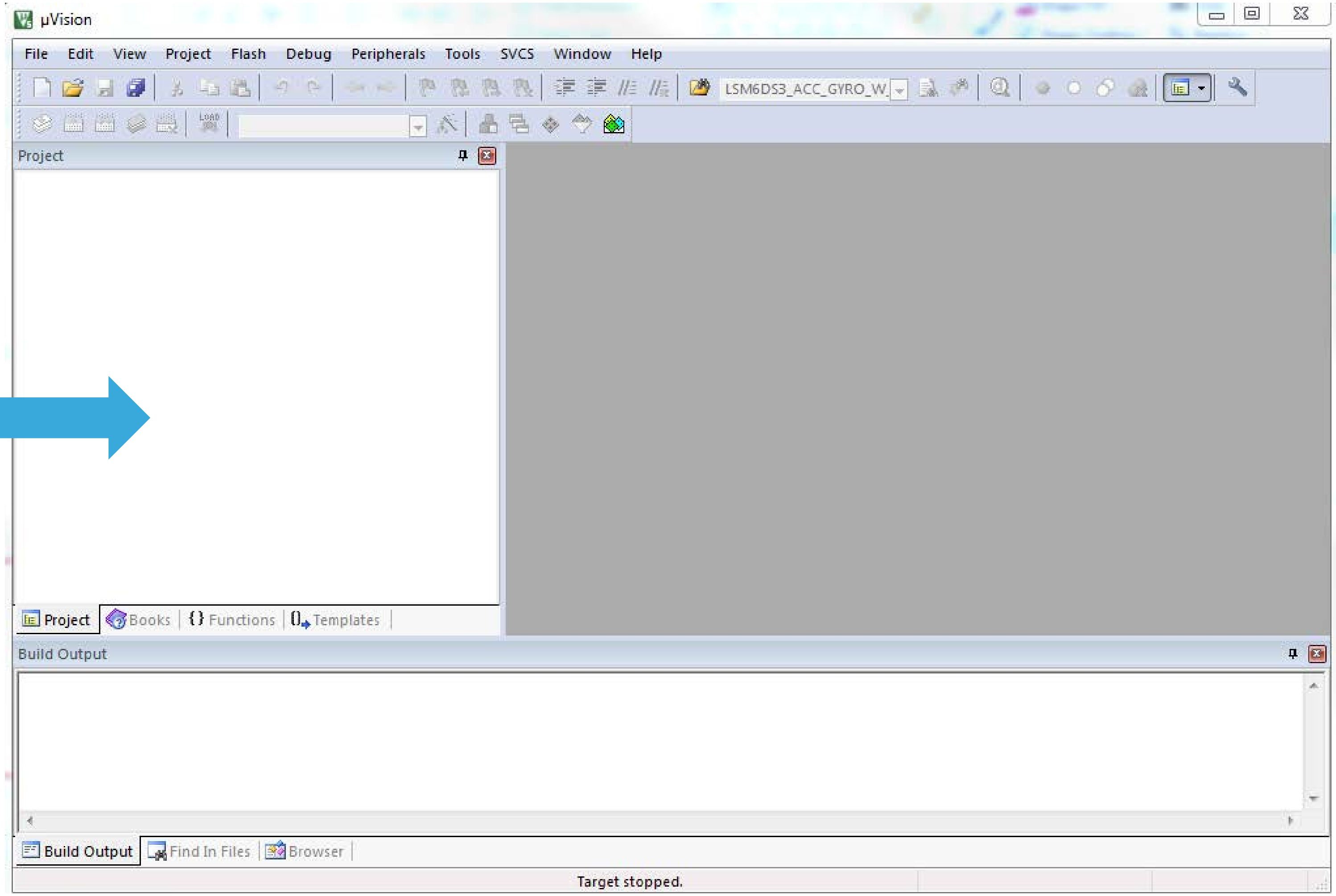
Download

Keil ARM MDK Configuration

- Open **Keil uVision5 IDE**
- This is the main GUI



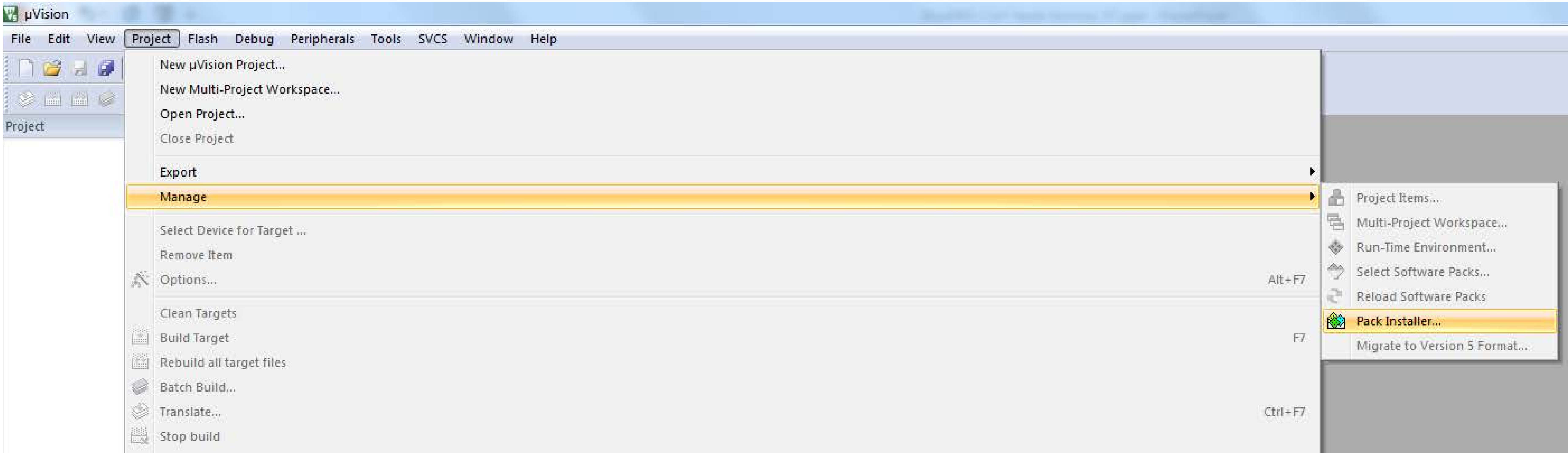
keil



Target stopped.

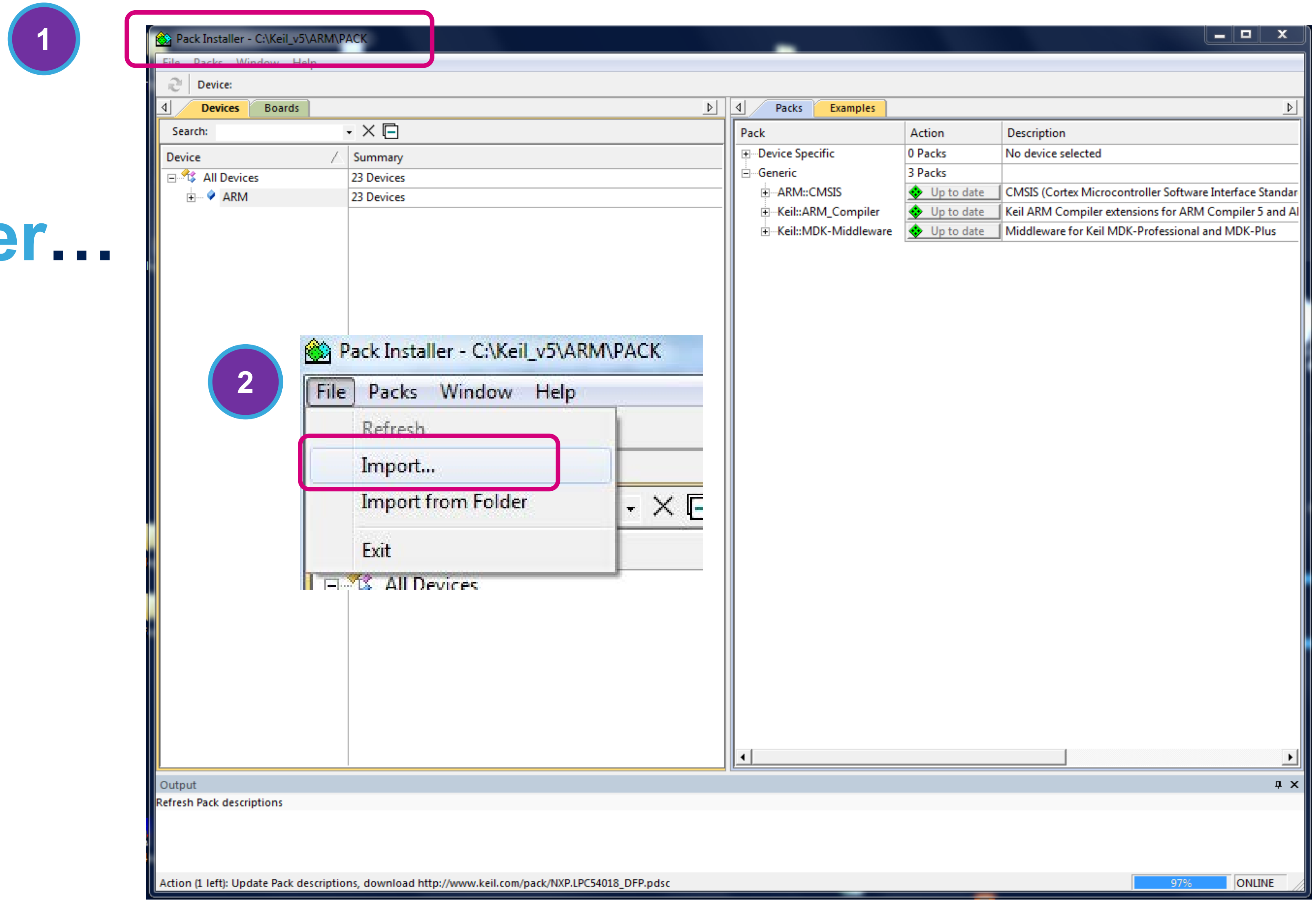
Keil ARM MDK Configuration

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



Keil ARM MDK Configuration

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



Device	Summary
All Devices	23 Devices
ARM	23 Devices

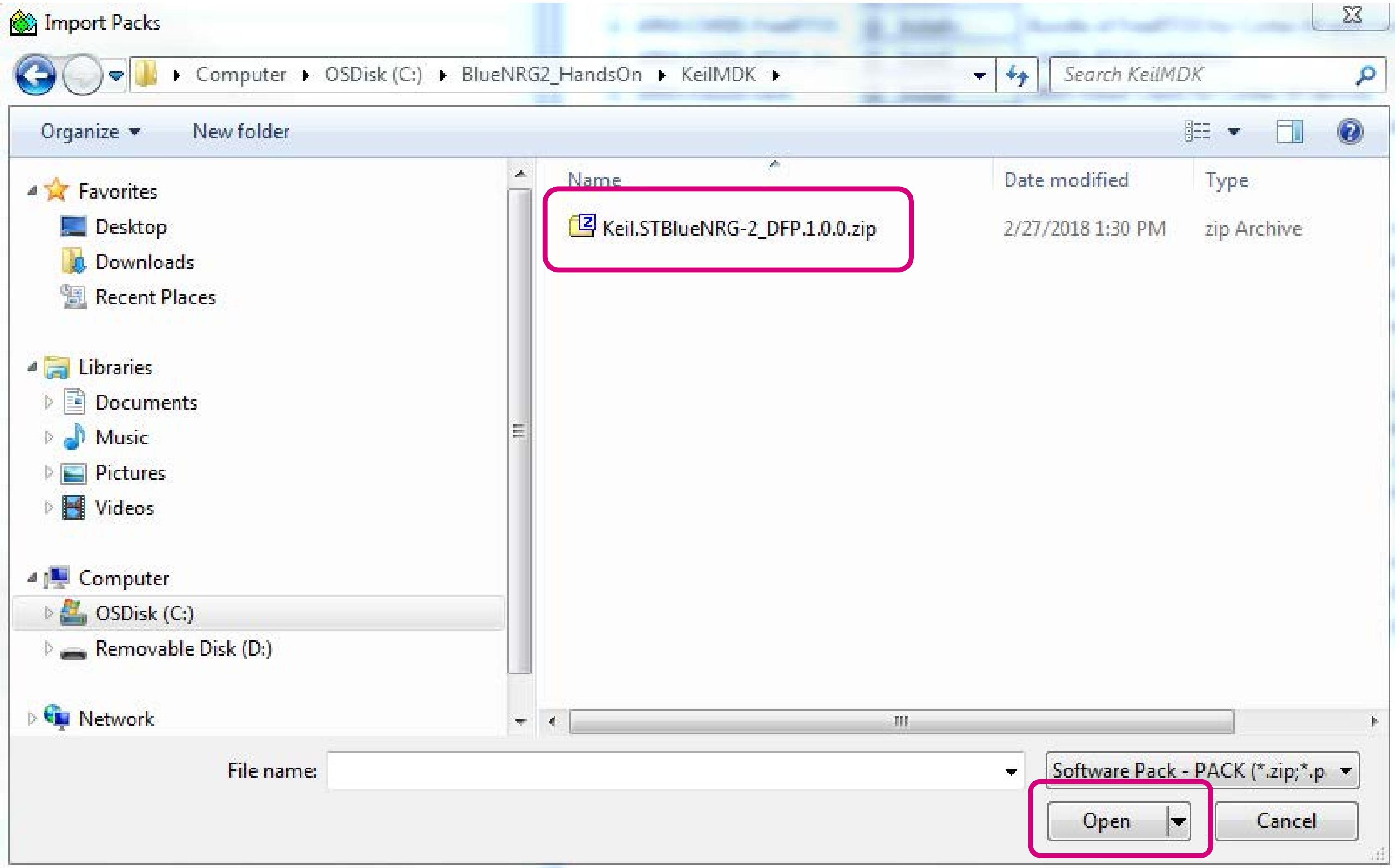
Pack	Action	Description
Device Specific	0 Packs	No device selected
Generic	3 Packs	
ARM::CMSIS	Up to date	CMSIS (Cortex Microcontroller Software Interface Standar
Keil::ARM_Compiler	Up to date	Keil ARM Compiler extensions for ARM Compiler 5 and AI
Keil::MDK-Middleware	Up to date	Middleware for Keil MDK-Professional and MDK-Plus

Refresh Pack descriptions

Action (1 left): Update Pack descriptions, download http://www.keil.com/pack/NXP.LPC54018_DFP.pdsc

Keil ARM MDK Configuration

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



Keil.STBlueNRG-2_DFP.1.0.0.zip 2/27/2018 1:30 PM zip Archive

Software Pack - PACK (*.zip;*.p

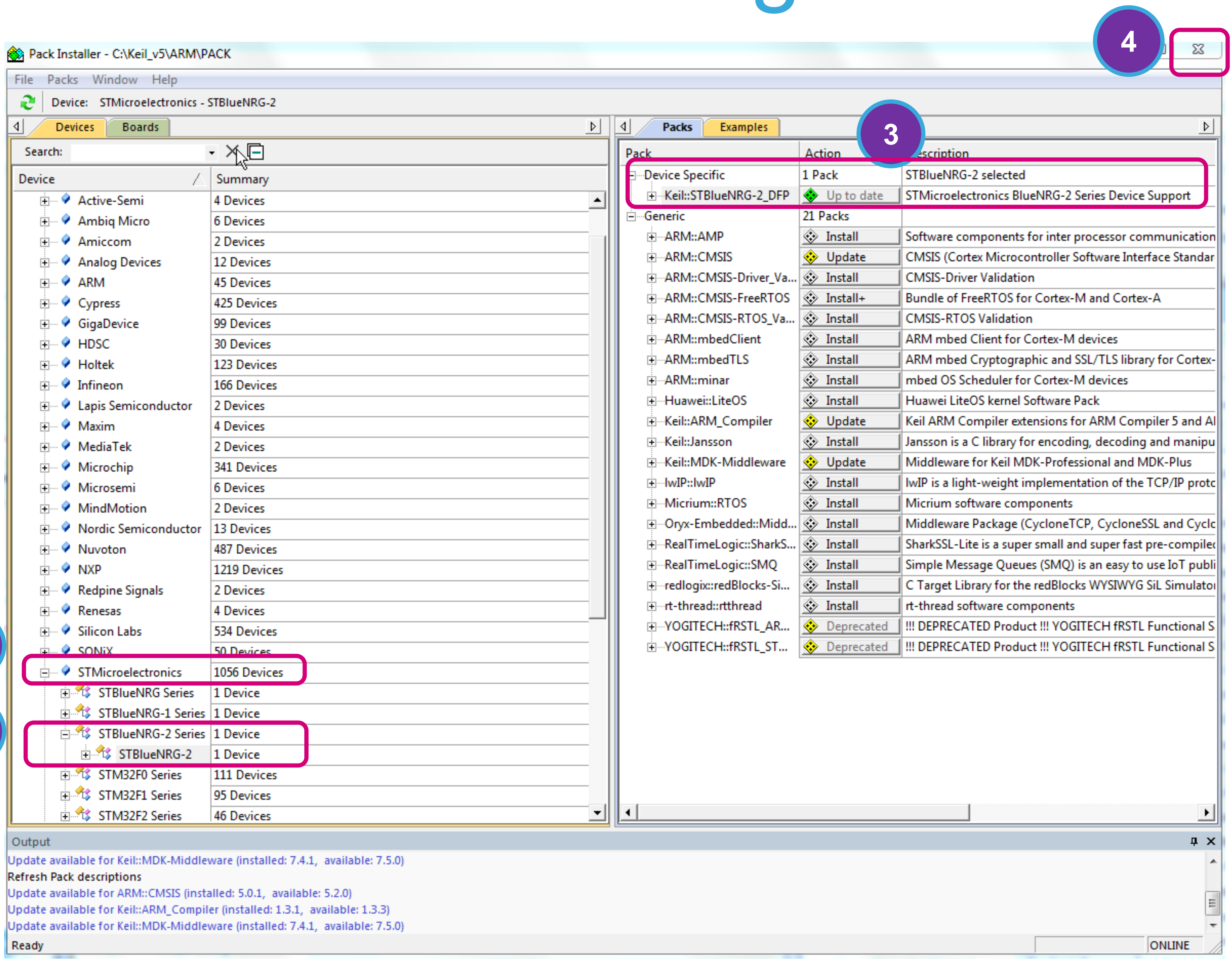
Keil ARM MDK Configuration

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

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

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

4. **Close** the **Pack Installer**



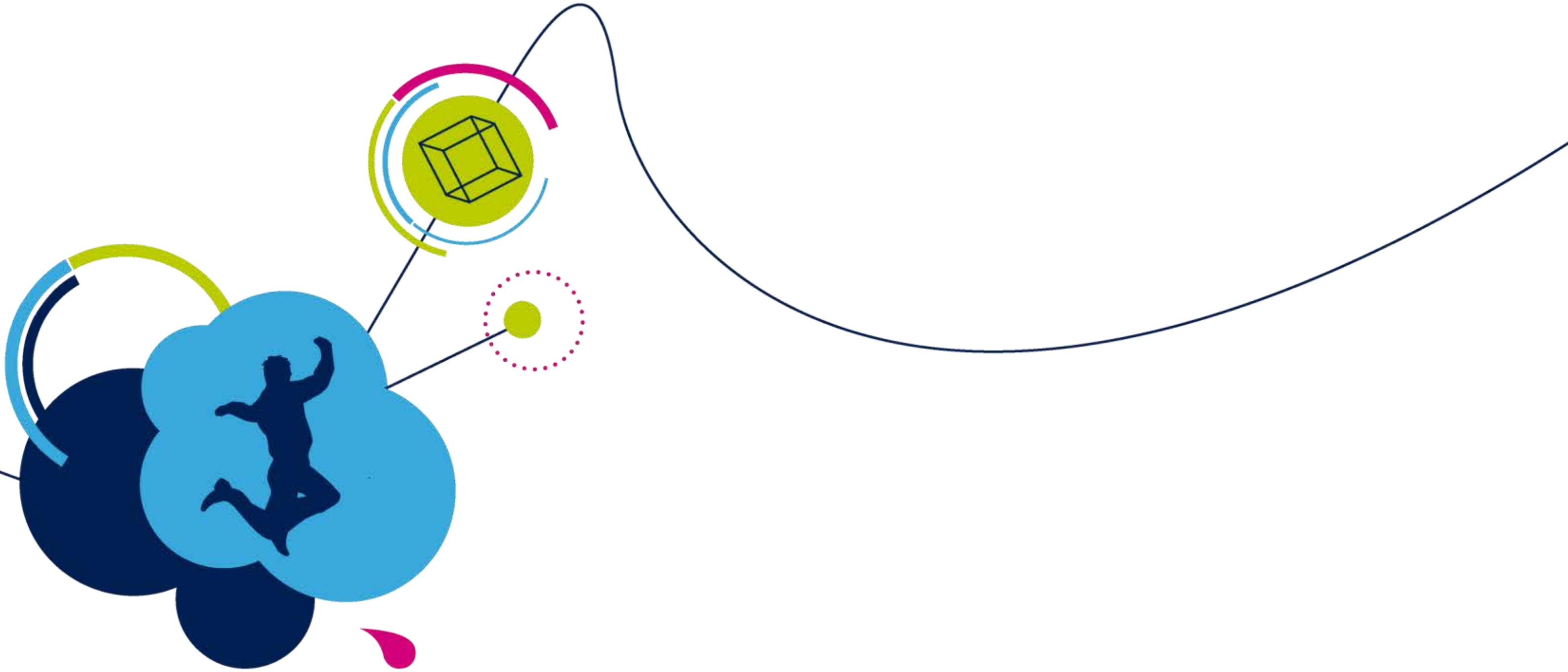
Device: STMicroelectronics - STBlueNRG-2

Device	Summary
Active-Semi	4 Devices
Ambiq Micro	6 Devices
Amiccom	2 Devices
Analog Devices	12 Devices
ARM	45 Devices
Cypress	425 Devices
GigaDevice	99 Devices
HDSC	30 Devices
Holtek	123 Devices
Infineon	166 Devices
Lapis Semiconductor	2 Devices
Maxim	4 Devices
MediaTek	2 Devices
Microchip	341 Devices
Microsemi	6 Devices
MindMotion	2 Devices
Nordic Semiconductor	13 Devices
Nuvoton	487 Devices
NXP	1219 Devices
Redpine Signals	2 Devices
Renesas	4 Devices
Silicon Labs	534 Devices
SONiX	50 Devices
STMicroelectronics	1056 Devices
STBlueNRG Series	1 Device
STBlueNRG-1 Series	1 Device
STBlueNRG-2 Series	1 Device
STBlueNRG-2	1 Device
STM32F0 Series	111 Devices
STM32F1 Series	95 Devices
STM32F2 Series	46 Devices

Pack	Action	Description
Device Specific	1 Pack	STBlueNRG-2 selected
Keil::STBlueNRG-2_DFP	Up to date	STMicroelectronics BlueNRG-2 Series Device Support
Generic	21 Packs	
ARM::AMP	Install	Software components for inter processor communication
ARM::CMSIS	Update	CMSIS (Cortex Microcontroller Software Interface Standar
ARM::CMSIS-Driver_Va...	Install	CMSIS-Driver Validation
ARM::CMSIS-FreeRTOS	Install+	Bundle of FreeRTOS for Cortex-M and Cortex-A
ARM::CMSIS-RTOS_Va...	Install	CMSIS-RTOS Validation
ARM::mbedClient	Install	ARM mbed Client for Cortex-M devices
ARM::mbedTLS	Install	ARM mbed Cryptographic and SSL/TLS library for Cortex-
ARM::minar	Install	mbed OS Scheduler for Cortex-M devices
Huawei::LiteOS	Install	Huawei LiteOS kernel Software Pack
Keil::ARM_Compiler	Update	Keil ARM Compiler extensions for ARM Compiler 5 and AI
Keil::Jansson	Install	Jansson is a C library for encoding, decoding and manipu
Keil::MDK-Middleware	Update	Middleware for Keil MDK-Professional and MDK-Plus
lwIP::lwIP	Install	lwIP is a light-weight implementation of the TCP/IP protc
Micrium::RTOS	Install	Micrium software components
Oryx-Embedded::Midd...	Install	Middleware Package (CycloneTCP, CycloneSSL and Cyclc
RealTimeLogic::SharkS...	Install	SharkSSL-Lite is a super small and super fast pre-compilec
RealTimeLogic::SMQ	Install	Simple Message Queues (SMQ) is an easy to use IoT publi
redlogix::redBlocks-Si...	Install	C Target Library for the redBlocks WYSIWYG SiL Simulatoı
rt-thread::rtthread	Install	rt-thread software components
YOGITECH::fRSTL_AR...	Deprecated	!!! DEPRECATED Product !!! YOGITECH fRSTL Functional S
YOGITECH::fRSTL_ST...	Deprecated	!!! DEPRECATED Product !!! YOGITECH fRSTL Functional S

```
Update available for Keil::MDK-Middleware (installed: 7.4.1,  available: 7.5.0)
Refresh Pack descriptions
Update available for ARM::CMSIS (installed: 5.0.1,  available: 5.2.0)
Update available for Keil::ARM_Compiler (installed: 1.3.1,  available: 1.3.3)
Update available for Keil::MDK-Middleware (installed: 7.4.1,  available: 7.5.0)
```

Ready ONLINE



BlueNRG-2 Free License for Keil ARM

MDK

life.augmented

BlueNRG-2 MDK License Activation

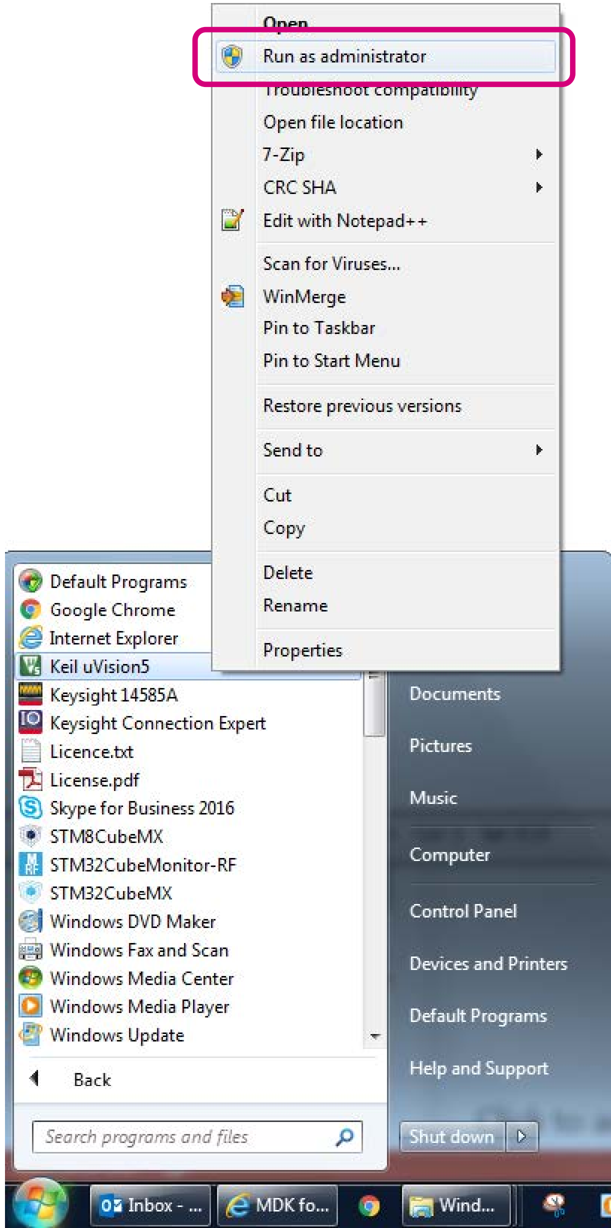
- You can refer to the following webpage:
  - http://www2.keil.com/stmicroelectronics-stm32/mdk

- Note: it is mentioned STM32L0 and STM32F0, but the same procedure applies to BlueNRG-2 device

BlueNRG-2 License Installation 1/6

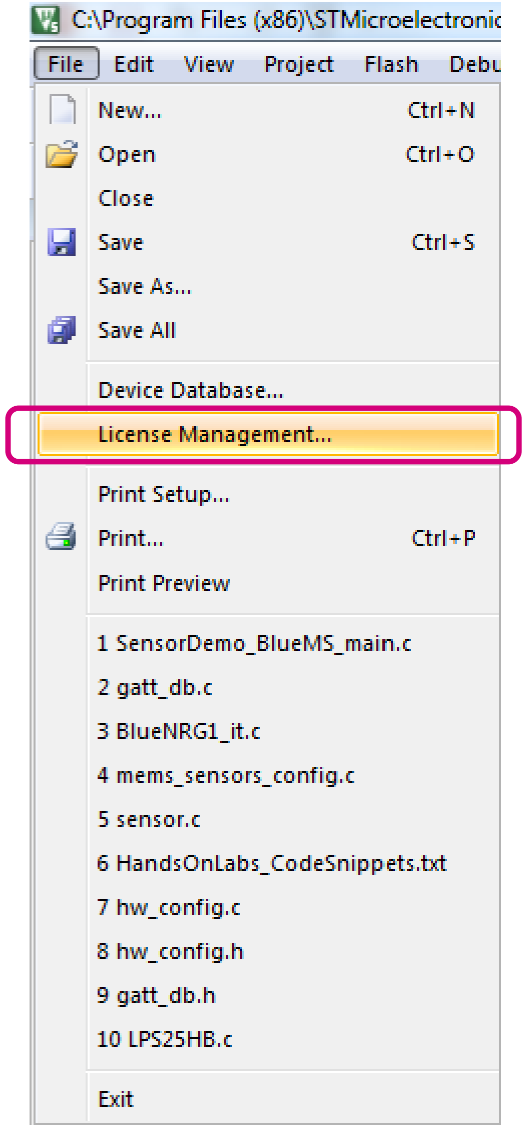
- Login with an account that has **administration rights**.

- Right-click the µVision icon and select **Run as Administrator...** from the context menu.



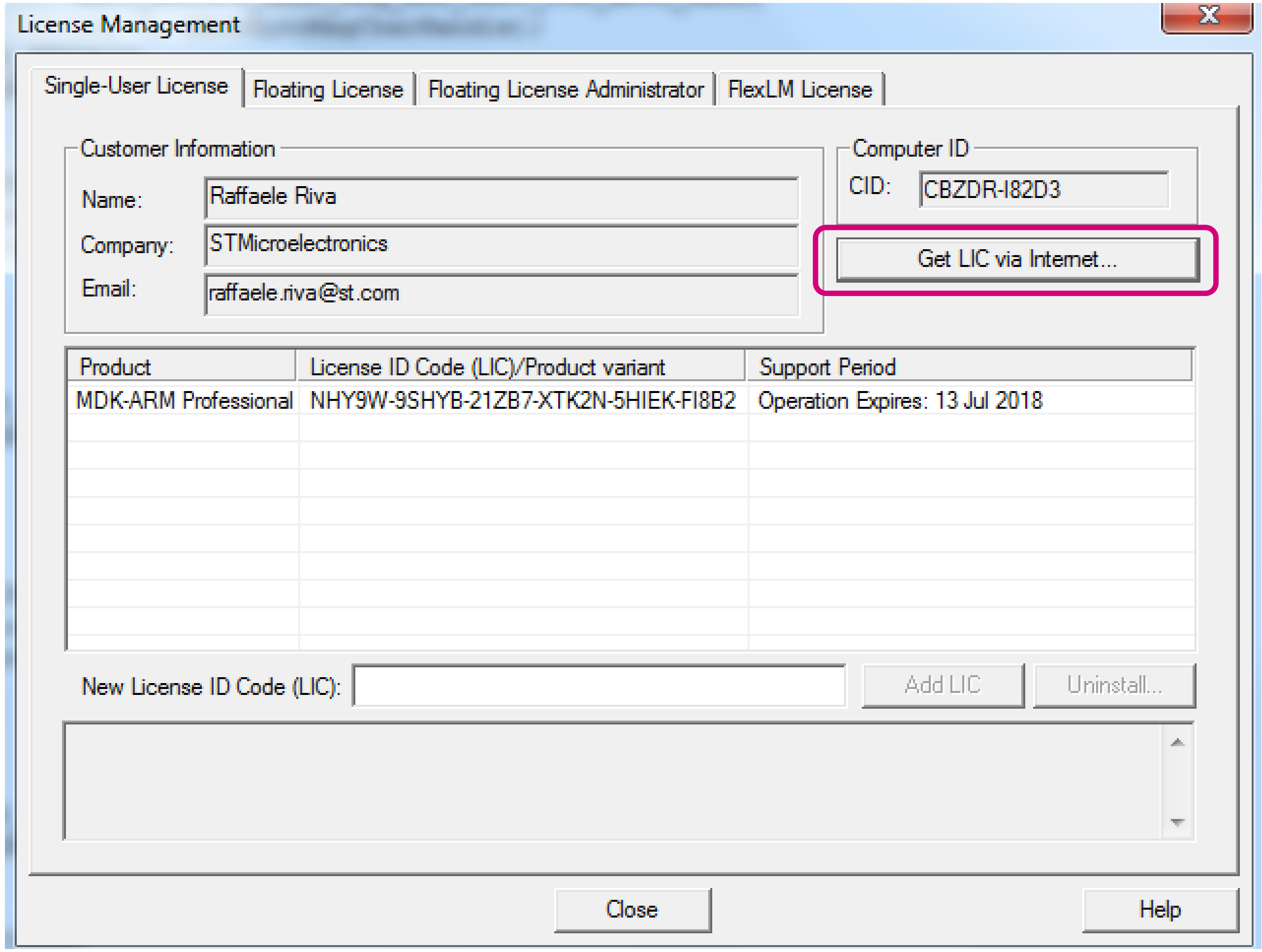
BlueNRG-2 License Installation 2/6

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



BlueNRG-2 License Installation 3/6

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-I82D3

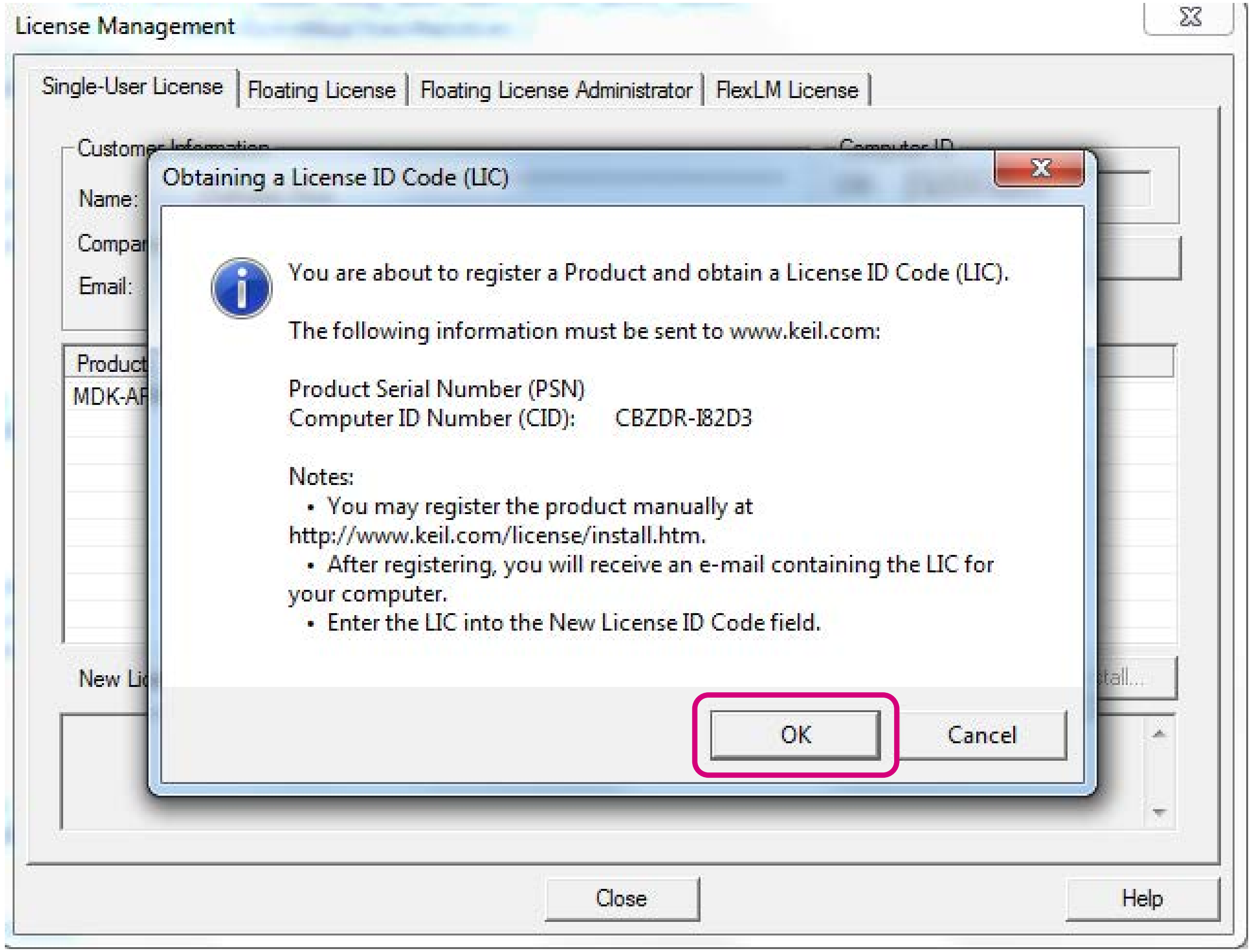
Product	License ID Code (LIC)/Product variant	Support Period
MDK-ARM Professional	NHY9W-9SHYB-21ZB7-XTK2N-5HIEK-FI8B2	Operation Expires: 13 Jul 2018

New License ID Code (LIC):

BlueNRG-2 License Installation 4/6

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

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



Obtaining a License ID Code (LIC)

You are about to register a Product and obtain a License ID Code (LIC).

The following information must be sent to www.keil.com:

Product Serial Number (PSN)
Computer ID Number (CID): CBZDR-I82D3

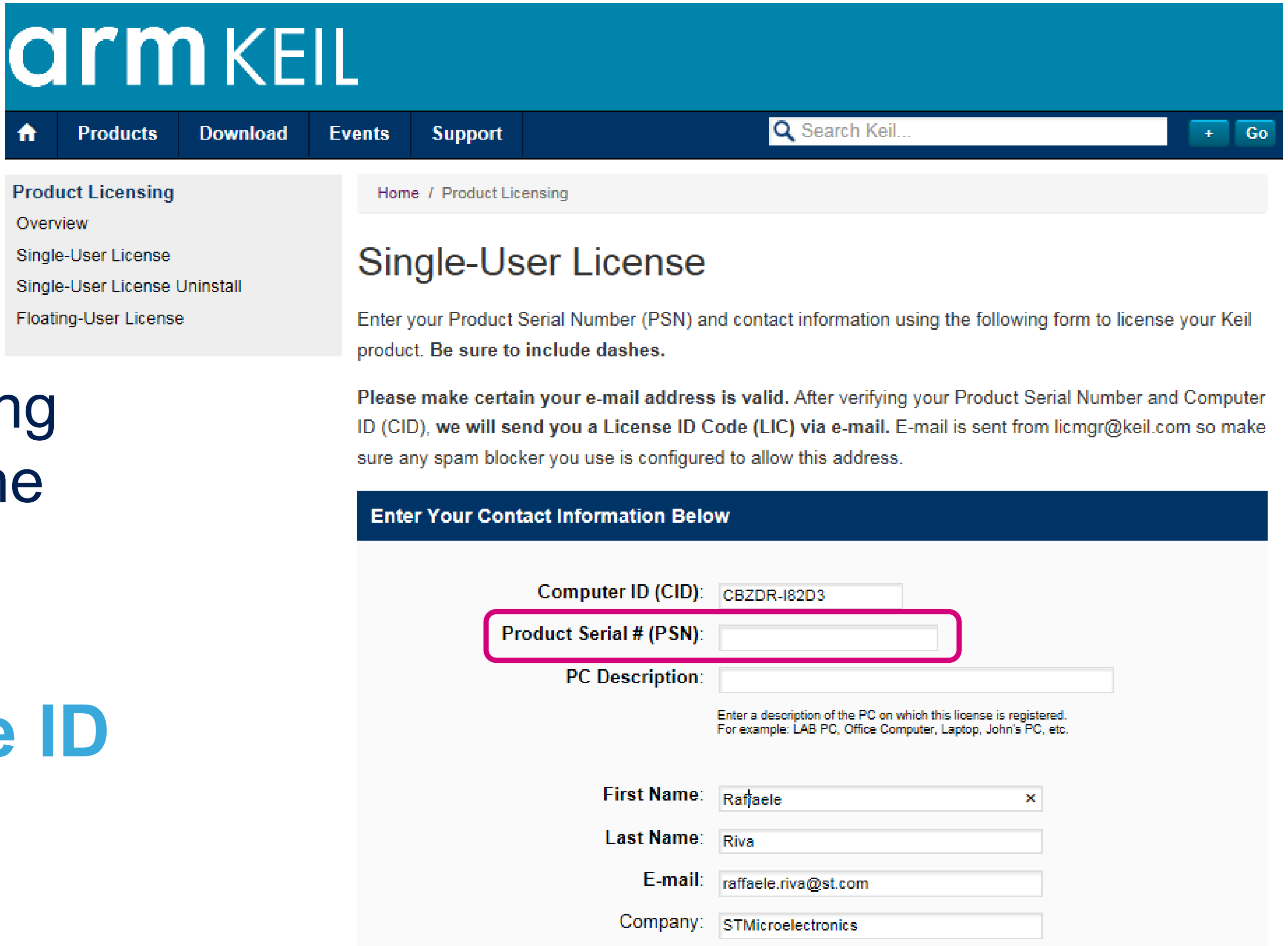
Notes:
- You may register the product manually at http://www.keil.com/license/install.htm.
- After registering, you will receive an e-mail containing the LIC for your computer.
- Enter the LIC into the New License ID Code field.

life.augmented

BlueNRG-2 License Installation 5/6

For the latest **Product Serial Number** please check on the Keil website at this URL:

http://www2.keil.com/stmicroelectronics-stm32/mdk



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

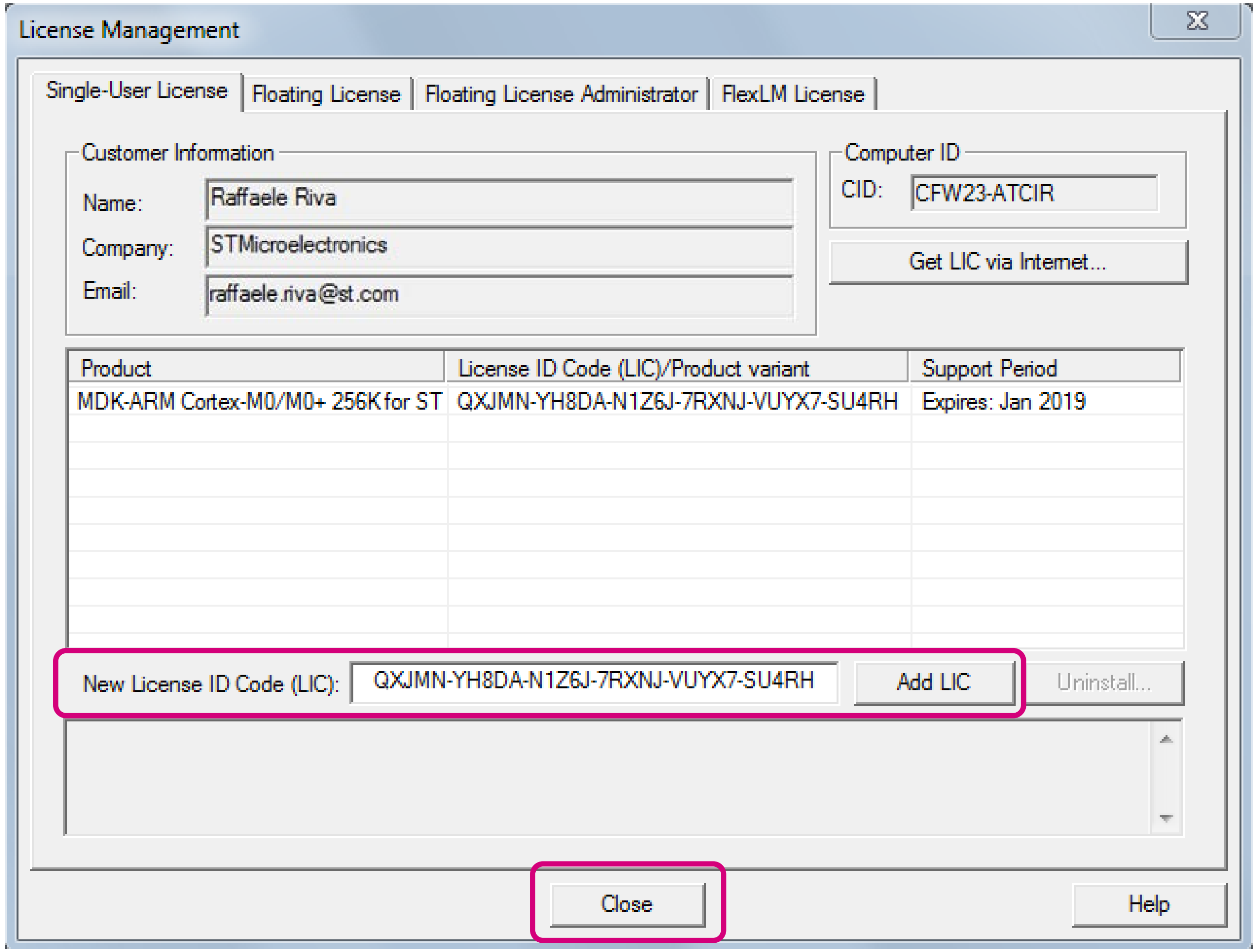
Then enter the PSN in this webpage along with your contact information and click the button **Submit**.

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

BlueNRG-2 License Installation 6/6

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

Then click on **Close**



Customer Information

Name: Raffaele Riva

Company: STMicroelectronics

Email: raffaele.riva@st.com

Computer ID

CID: CFW23-ATCIR

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

life.augmented