



STM32WBA The New and Better - Hands On

Manuel Marcias – South EMEA Field Application Engineer – Microcontroller Products

Agenda

- 1 Prerequisites
- 2 Hands On Purpose
- 3 How to start WBA evaluation ?
- 4 Hands On: Let's build P2P server with STM32Cube tools
- 5 Hands On: BLE sequences clarification
- 6 ST EMEA TOMAS support
- 7 STM32WBA Take Away

Prerequisites



Prerequisites Refresh

STM32WBA Workshop - preparation before the session

- [undefined \(rristm.github.io\)](https://rristm.github.io)

SW prerequisites

- STM32CubeWBA MCU package (v1.1.0 or up)
- IDE: STM32CubeIDE **v 1.13.1**
- A serial terminal (e.g. TeraTerm)
- ST BLE ToolBox Smartphone application

HW prerequisites

- USB A to Micro-B Cable



ST BLE Toolbox

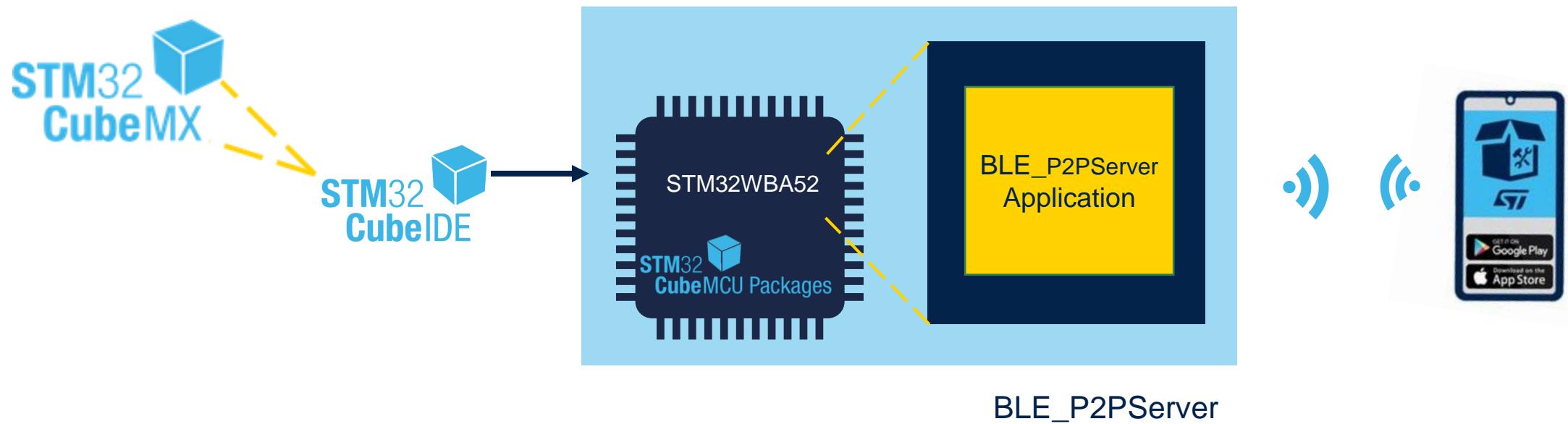


Hands On Purpose



Hands On Purpose

- The purpose of this session is to start from an existing project example **BLE_P2PServer** & modify this code example to customize advertising data (Local name).



- During this Hands-on session, we will use ST tool STM32CubeMX & STM32CubeIDE to generate associated code, flash and test over Nucleo-WBA5x board.



AVNET SILICA



How to start WBA evaluation ?

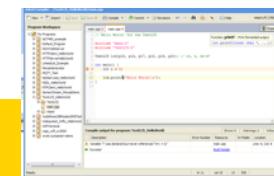


How to start with STM32WBA

Complete set of documentation
#1 understand



Powerful turnkey code examples
#2 evaluate & prototype



A complete ecosystem
#3 start design



NUCLEO-WBA52CG



A complete set of documentation Getting started

STM32WBA Wiki page - Start to learn & play

The screenshot shows a wiki page titled 'STM32CubeWBA Software Architecture'. The left sidebar contains a navigation menu with categories like Artificial Intelligence, Connectivity, and STM32WB Series. The main content area displays a table of contents for the software architecture, including sections on General software architecture, Project architecture, and Platform Resources.

- Getting started
- code architecture
- Application code explanations & tips
- certification process



https://wiki.st.com/stm32mcu/wiki/Category:Getting_started_with_STM32WB-WBA#Bluetooth-C2-AE_LE_overview

AN & HW guidance @st.com – Deep dive on product APIs

[AN5928 - How to build a short range wireless application with STM32WBA MCUs](#)

[PM0271 Guidelines for Bluetooth® Low Energy stack programming on STM32WB/ STM32WBA MCUs](#)

[AN5042 How to calibrate the HSE clock for RF applications on STM32 wireless MCUs](#)

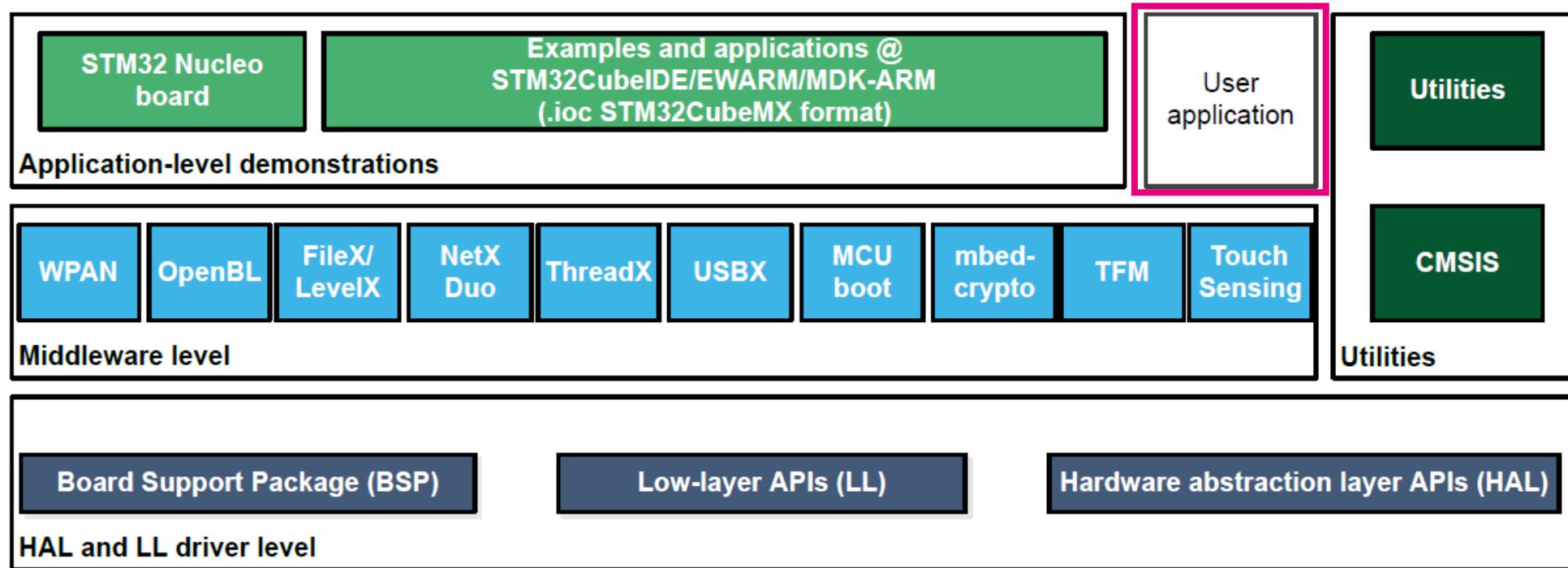




STM32CubeWBA

Evaluate and start with **existing code**

Package Components in a Nutshell





STM32CubeWBA

Evaluate and start based on existing code

STM32CubeWBA MCU Package provides software components running on STM32WBA Series MCU

- Application code : BLE , 802.15.4, Thread, Zigbee
- Peripheral code examples
- Utilities to ease application design : timer capabilities, sequencer, LCD



development environments



arm KEIL



Firmware Package for the STM32WBA series is also available on STM32CubeWBA GitHub



STM32CubeWBA

Content 1/3

STM32CubeWBA MCU folder overview



Name

📁 _htmresc
📁 Documentation
📁 Drivers
📁 Middlewares
📁 Projects
📁 Utilities
📄 package.xml
chrome Package_license.html
markdown Package_license.md
chrome Release_Notes.html

Documentation includes **STM32CubeWBA Getting Started Guide**

Driver includes BSP, CMSIS as well as HAL, LL for each peripheral

Consistent set of libraries such as Azure RTOS components such as ThreadX, FileX, LevelX, NetXDuo as well as STM32_WPAN, Touch Sensing Library, MCUboot, Crypto and TFM

STM32CubeWBA

Content 2/3

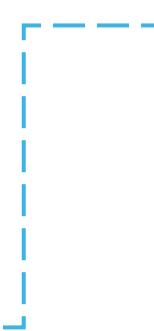
STM32CubeWBA MCU folder overview



Name
_htmresc
Documentation
Drivers
Middlewares
Projects
Utilities
package.xml
Package_license.html
Package_license.md
Release_Notes.html



various **examples** and **application** for the evaluation board



BLE application code examples

- BLE_ApplicationInstallManager
- BLE_Beacon
- BLE_DataThroughput_Client
- BLE_DataThroughput_Server
- BLE_HealthThermometer
- BLE_HeartRate
- BLE_HeartRate_ota
- BLE_HeartRateThreadX
- BLE_p2pClient
- BLE_p2pClient_Ext
- BLE_p2pRouter
- BLE_p2pServer
- BLE_p2pServer_Ext
- BLE_p2pServer_ota
- BLE_p2pServerThreadX
- BLE_SerialCom_Central
- BLE_SerialCom_Peripheral
- BLE_TransparentMode

STM32
CubelDE

arm KEIL

iar

Evaluate & understand BLE APIs
Start prototyping from existing code
(proprietary BLE profile)

STM32CubeWBA Content 3/3

STM32CubeWBA MCU folder overview



Name

- 📁 _htmresc
- 📁 Documentation
- 📁 Drivers
- 📁 Middlewares
- 📁 Projects
- 📁 Utilities
- 📄 package.xml
- chrome Package_license.html
- download Package_license.md
- chrome Release_Notes.html



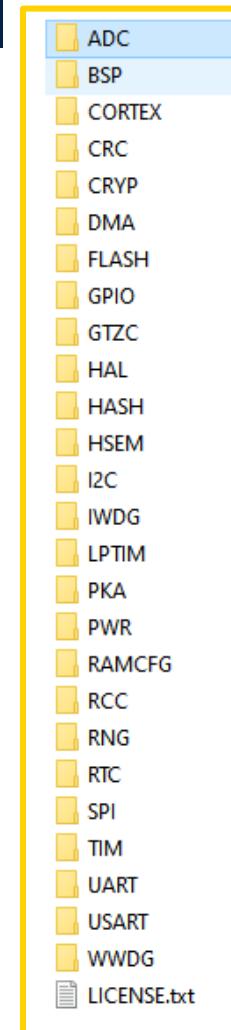
various **examples** and **application** for the evaluation board

📁 Applications

- 📁 Examples
- 📁 Examples_LL
- 📁 Examples_MIX
- 📁 Templates
- 📁 Templates_LL

STM32 CubeIDE arm KEIL iar

peripheral code examples



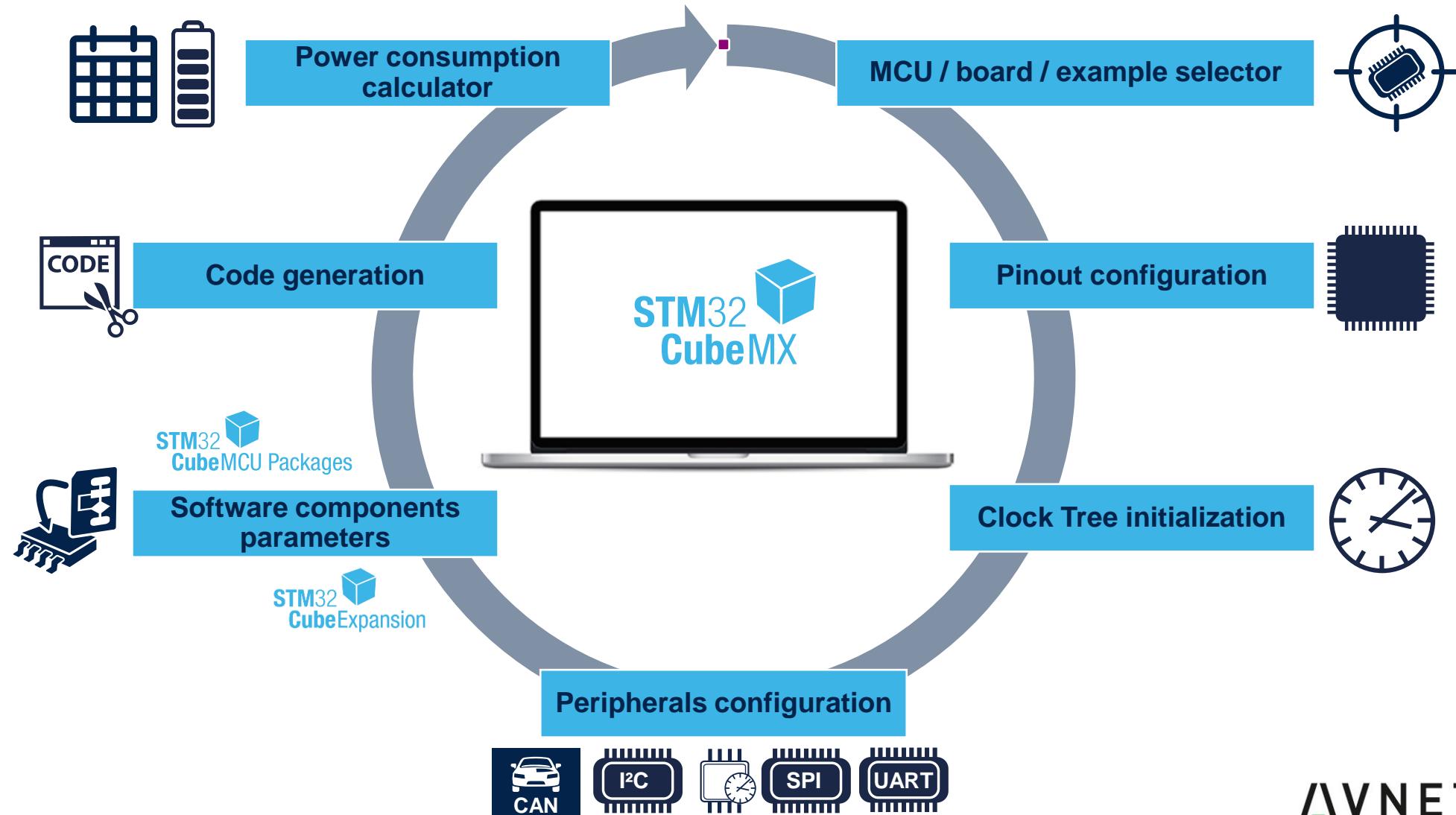
- 📁 ADC
- 📁 BSP
- 📁 CORTEX
- 📁 CRC
- 📁 CRYP
- 📁 DMA
- 📁 FLASH
- 📁 GPIO
- 📁 GTZC
- 📁 HAL
- 📁 HASH
- 📁 HSEM
- 📁 I2C
- 📁 IWDG
- 📁 LPTIM
- 📁 PKA
- 📁 PWR
- 📁 RAMCFG
- 📁 RCC
- 📁 RNG
- 📁 RTC
- 📁 SPI
- 📁 TIM
- 📁 UART
- 📁 USART
- 📁 WWDG
- 📄 LICENSE.txt

Peripheral applications running over NUCLEO HW



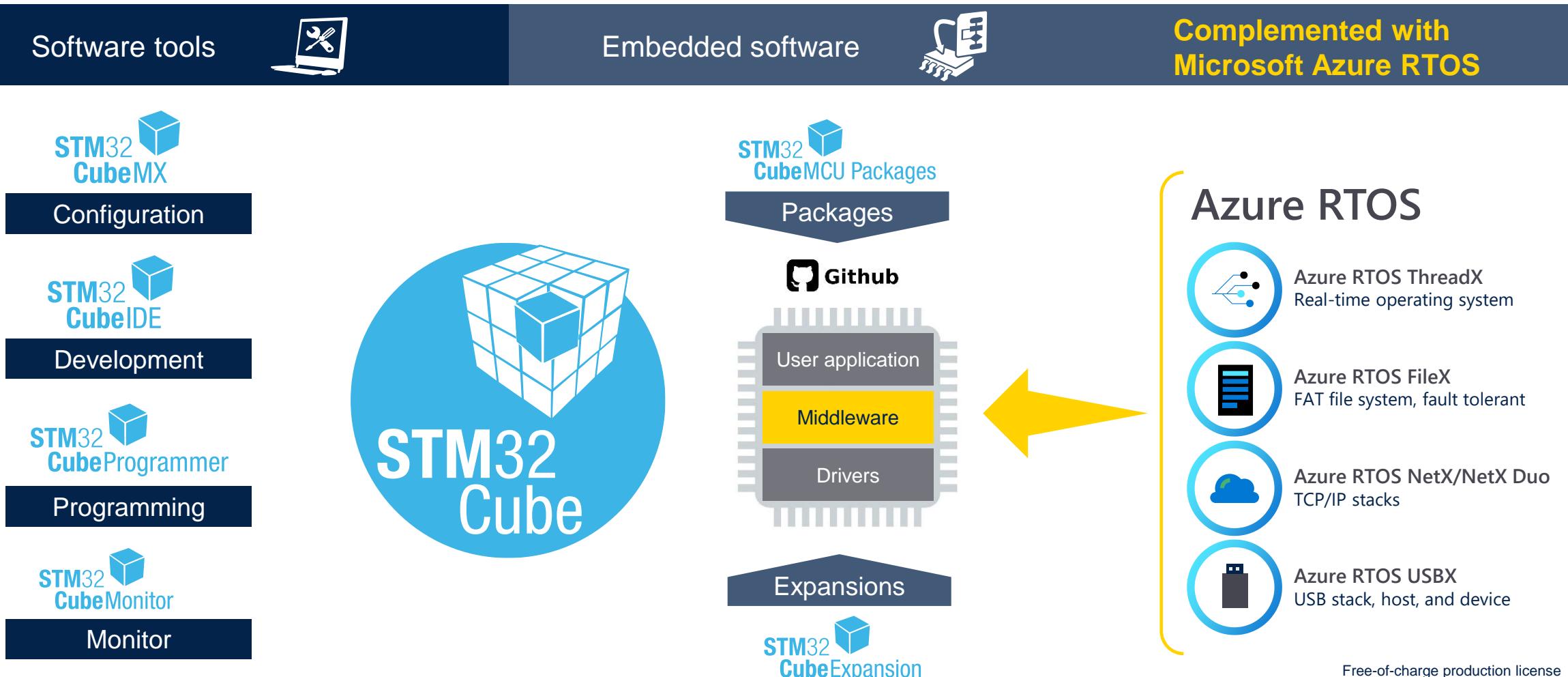
STM32CubeMX key steps

Start your Design





Leveraging STM32Cube software suite



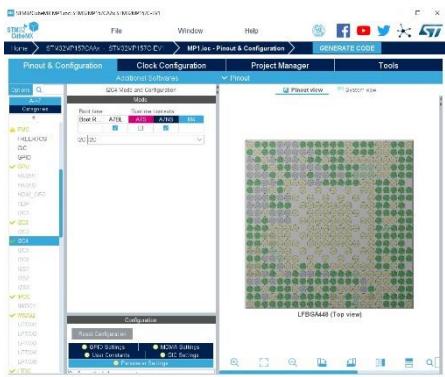


STM32WBx ecosystem

Complete set of tools to ease your design

From an Idea to a certified product

STM32
CubeMX



STM32
CubeIDE

eclipse

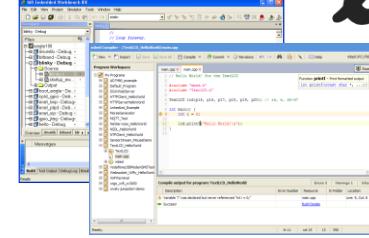
STM32CubeMX

Graphical tool
for easy configuration

- Configure and generate code
- Peripherals and middleware configuration

arm KEIL

iar



IDEs
Compile and debug

Simple,
powerful solutions

- Partners IDE (Arm® Keil®) **FREE**
- IDE based on Eclipse **FREE**
- RTOS aware debug



STM32
CubeProgrammer

STM32
CubeMonitor

STM32 programming
& monitoring tools

STM32CubeProg
STM32CubeMonitor
STM32CubeMonRF

- Device and memory configuration
- Program the application
- Monitor variables at runtime , Test RF

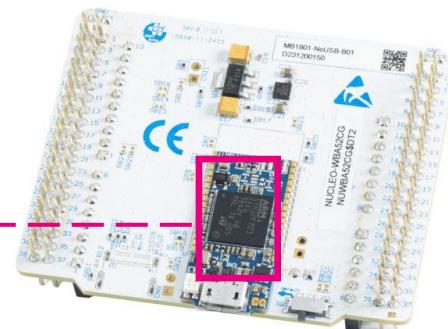
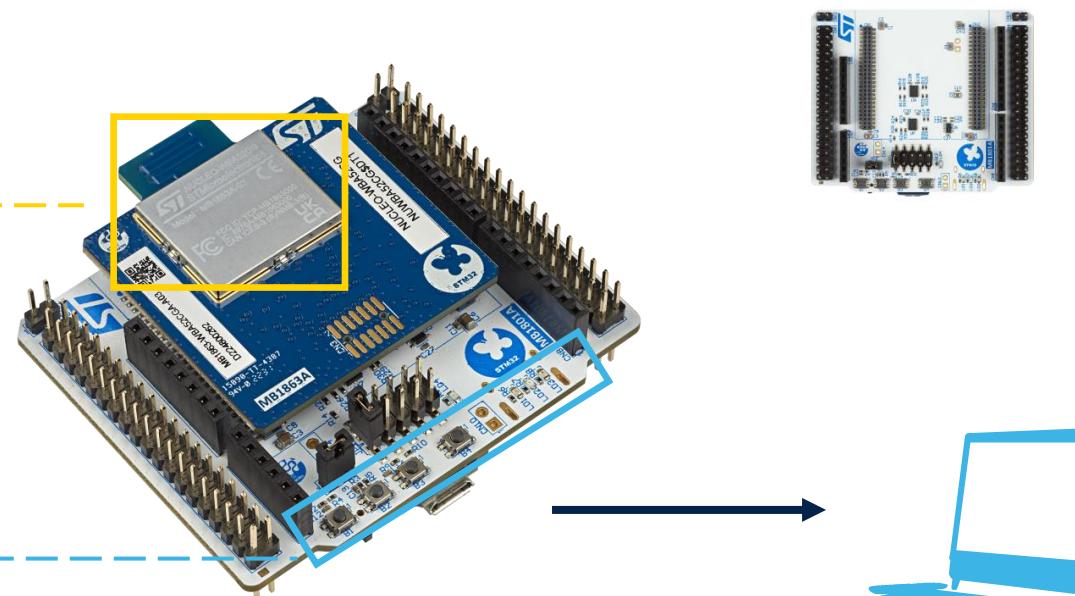
Connect the NUCLEO-WBA52CG to the PC

STM32WBA52CG

3 user LEDs
3 user buttons

STLINK-V3MODS

- Programming
- Debugging
- VCOM



Hands on session: Let's build P2P server with STM32Cube tools



What is a P2P Server?

P2P is a Generic Attribute Profile (GATT) based on Bluetooth LE defined by STM with proprietary UUIDs 128bit

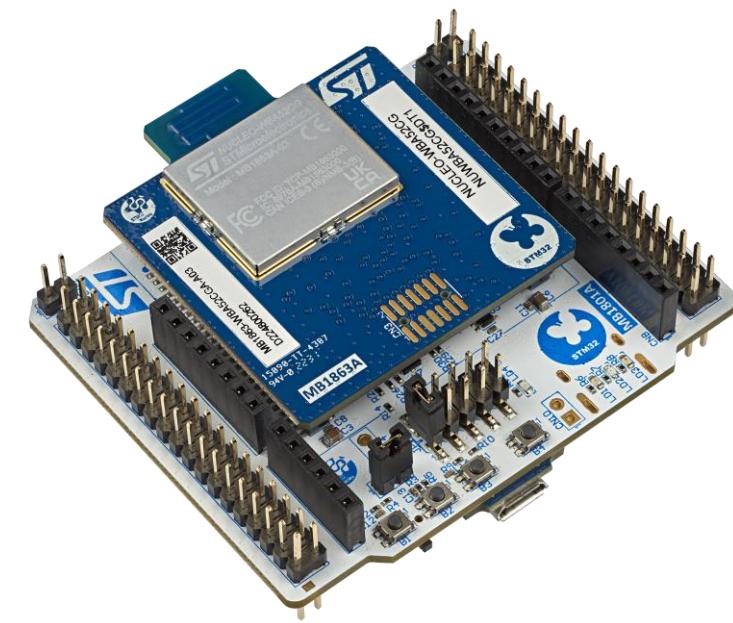


ST BLE Toolbox



GATT Client

GAP central



GATT Server

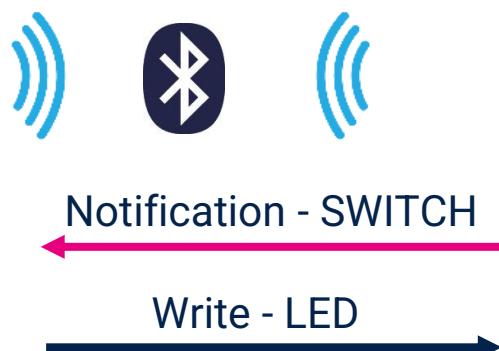
GAP peripheral





What is a P2P Server?

P2P is widely used for direct connection and defined connection between GATT Server and GATT Client



ST BLE Toolbox



GATT Client

GAP central

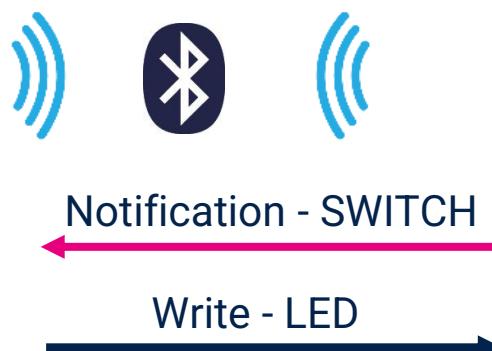
GATT Server

GAP peripheral



What is a P2P Server?

We will be able to control LED from Mobile and to get notification of LED status from Nucleo-WBA52



ST BLE Toolbox



GATT Client

GAP central

GATT Server

GAP peripheral

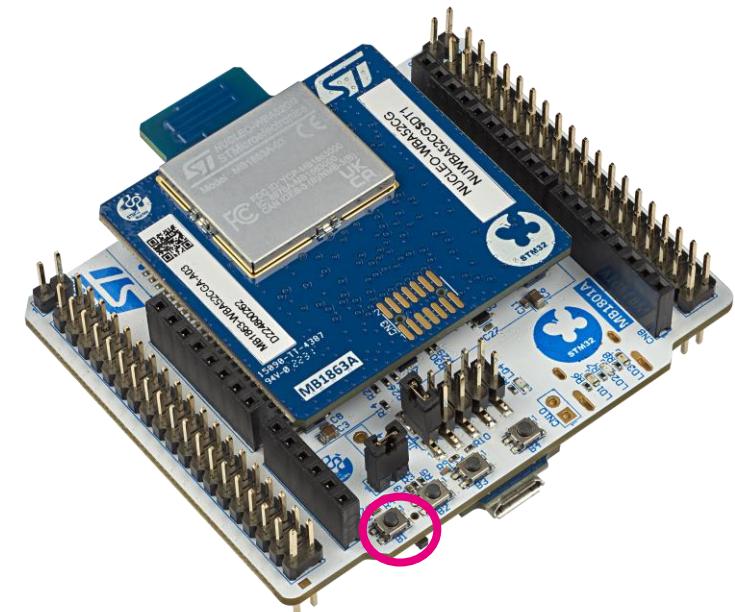


What is a P2P Server?

We will be able to have the notification on the Mobile that a button is pushed on the Nucleo-WBA52



Notification – PUSH BUTTON



ST BLE Toolbox



GATT Client

GAP central

GATT Server

GAP peripheral

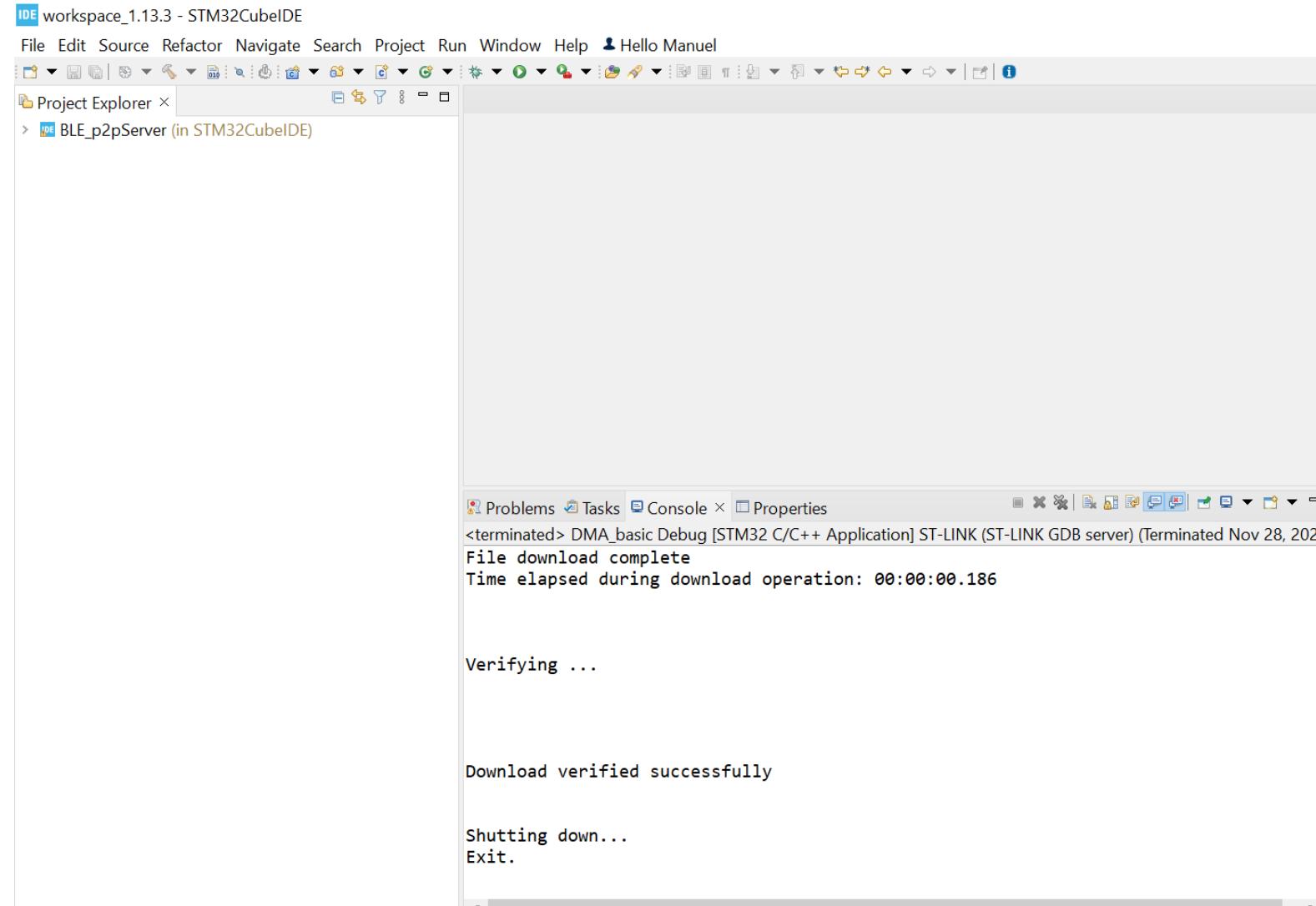
How to get started

- Open Cube IDE
 - Select File->Import->Existing project into Workspace
 - Go to the folder STM32CubeWBA in which you have p2p Server
-
- Add the path to the example
C:\Users\xxxxx\STM32Cube\Repository\STM32Cube_FW_WBA_V1.1.1\Projects\NUCLEO-WBA52CG\Applications\BLE\BLE_p2pServer
 - Click Finish



AVNET[®] SILICA

P2P Should now appear into your workspace





AVNET SILICA

Customize Local Name

The screenshot shows the STM32CubeMX interface for a project named `STM32WBA52CGUx - NUCLEO-WBA52CG`. The `BLE_p2pServer.ioc` file is selected. The **Pinout & Configuration** tab is active.

- Category Filter:** A search bar and a dropdown menu labeled "Categories" with "A-Z" selected. A red circle with the number 1 is placed above the search bar.
- Project Structure:** A sidebar on the left lists various peripheral components. A red circle with the number 2 is placed next to the `STM32_WPAN` entry, which is highlighted with a blue selection bar.
- Configuration Tab:** The configuration tabs at the top of the main area are `SERVICE1`, `User Constants`, `Platform Settings`, `BLE Applications and Services`, `Configuration`, and `BLE Advertising`. The `BLE Advertising` tab is selected, indicated by a red box and a red circle with the number 3.
- Parameter Editor:** The `AD_TYPE_COMPLETE_LOCAL_NAME` parameter is set to `MyName_01`. A red box highlights this value, and a red circle with the number 4 is placed near the parameter name. A yellow warning box displays the message: **Local Name must be < 11 CubeMx constraints**.

Pinout View: On the right side, a detailed pinout diagram for the `STM32WBA52CGUx` (UFQFPN48 package) is shown, mapping specific pins to their functions.



AVNET[®]
SILICA

Customize Device Name

The screenshot shows the STM32CubeMX software interface. The top navigation bar includes File, Window, Help, myST, and social media links. The main menu shows Home > STM32WBA52CGUx - NUCLEO-WBA52CG > Hands-On_WS_WBA52.ioc - Pinout & Configuration.

On the left, a sidebar lists categories like I2C, ICACHE, MISC, PWR, RAMCFG, RCC, RF, RNG, RTC, SEQUENCER, SPI1, SPI2, STM32_WPAN, SYS, TAMP, THREADX, TIM1, TIM2, TIM3, TIM16, TIM17, TIMER, TINY_LPM, TOUCHSENSING, TSC, USART1, USART2, WWWDG, X-CUBE-AI, X-CUBE-ALGOBUILD, X-CUBE-ALS, X-CUBE-BLE1, X-CUBE-BLE2, and X-CUBE-BLEMGR. A pink circle labeled '1' highlights the 'A-Z' sort button in the sidebar.

The central area has tabs for Pinout & Configuration, Clock Configuration, Project Manager, and Tools. The Pinout & Configuration tab is active. It shows the STM32WBA52CGUx chip with its pinouts. A pink circle labeled '2' highlights the 'STM32_WPAN' entry in the category list. Another pink circle labeled '3' highlights the 'Configuration' tab in the configuration panel. A pink circle labeled '4' highlights the 'CFG_GAP_DEVICE_NAME' parameter set to 'My_Name_01'. A yellow callout box below it states: "set same Device name = Local Name".

iOS displays Local Name (advertising data) prior to a 1st connexion.

After a 1st connexion iOS displays Device name (thanks to look up table : associates BLE MAC @ & Device Name)





AVNET[®] SILICA

Generate code

The screenshot shows the STM32CubeMX software interface with the following highlights:

- Project Name:** BLE_p2pServer (highlighted with a red box and labeled 1).
- Toolchain / IDE:** STM32CubelDE (highlighted with a red box and labeled 2). A callout text says: "Select here your favorite IDE".
- Generate Code Button:** GENERATE CODE (highlighted with a red box and labeled 3).

The Project Manager tab is active. In the background, a progress bar indicates "Generating user source code...".

Project Settings:

- Project Name: BLE_p2pServer
- Project Location: C:\Users\marciasm\STM32Cube\Example

Code Generator:

- Application Structure: Advanced
- Toolchain Folder Location: C:\Users\marciasm\STM32Cube\Example\BLE_p2pServer
- Toolchain / IDE: STM32CubelDE (selected)

Advanced Settings:

- Linker Settings:
 - Minimum Heap Size: 0x3000
 - Minimum Stack Size: 0x1000
- Thread-safe Settings:
 - CortexM33
 - Enable multi-threaded support
 - Thread-safe Locking Strategy: Default – Mapping suitable strategy depending on RTOS selection.
- Mcu and Firmware Package:
 - Mcu Reference: STM32WBA52CGUx
 - Firmware Package Name and Version: STM32Cube FW_WBA V1.1.0
 - Use Default Firmware Location
 - Firmware Relative Path: C:/Users/marciasm/STM32Cube/Repository/STM32Cube_FW_WBA_V1.1.0



Advanced Settings

The screenshot shows the STM32CubeMX software interface with the 'Advanced Settings' tab highlighted in red. The interface includes tabs for Pinout & Configuration, Clock Configuration, Project Manager, and Tools. The Project Manager tab is active, displaying driver selection for various peripherals like RCC, GPIO, GPDMA, PWR, RAMCFG, RTC, USART, ADC, and CRC, with options for HAL or LL. The Tools tab shows register callback definitions for various peripherals. A callout box points to the Project Manager table with the text: "From this tab you've selected LL/ HAL for each driver". Another callout box points to the Tools table with the text: "Enable the generation of callback define statements".

From this tab you've selected LL/ HAL for each driver

Enable the generation of callback define statements

Generate Code	Rank	Function Name	Peripheral Instance Name	<input type="checkbox"/> Do Not Generate Function Call	<input type="checkbox"/> Visibility (Static)
✓	1	SystemClock_Config	RCC	<input type="checkbox"/>	<input type="checkbox"/>
✓	2	MX_GPIO_Init	GPIO	<input type="checkbox"/>	<input type="checkbox"/>
✓	3	MX_GPDMA1_Init	GPDMA1	<input type="checkbox"/>	<input type="checkbox"/>
✓	4	SystemPower_Config	PWR	<input type="checkbox"/>	<input checked="" type="checkbox"/>
✓	5	MX_RAMCFG_Init	RAMCFG	<input type="checkbox"/>	<input type="checkbox"/>
✓	6	MX_RTC_Init	RTC	<input type="checkbox"/>	<input type="checkbox"/>
✓	7	MX_USART1_UART_Init	USART1	<input type="checkbox"/>	<input type="checkbox"/>
✓	8	APPE_Init	STM32_WPAN	<input type="checkbox"/>	<input type="checkbox"/>
✓	9	MX_ADC4_Init	ADC4	<input type="checkbox"/>	<input type="checkbox"/>
✓	10	MX_CRC_Init	CRC	<input checked="" type="checkbox"/>	<input type="checkbox"/>
✓	11	MX_RNG_Init	RNG	<input type="checkbox"/>	<input type="checkbox"/>
✓	12	MX_ICACHE_Init	ICACHE	<input type="checkbox"/>	<input type="checkbox"/>





The screenshot shows the STM32CubeMX software interface with the project `STM32CubeMX BLE_p2pServer.ioc` open. The `Project Manager` tab is selected. The `Code Generator` section contains the following configuration:

- Generated files:**
 - Generate peripheral initialization as a pair of '.c/.h' files per peripheral
 - Backup previously generated files when re-generating
 - Keep User Code when re-generating
 - Delete previously generated files when not re-generated
- HAL Settings:**
 - Set all free pins as analog (to optimize the power consumption)
 - Enable Full Assert

Two annotations highlight specific settings:

- A red box surrounds the "Generate peripheral initialization as a pair of '.c/.h' files per peripheral" checkbox with the text: "You can select how HAL .c/.h are included inside the project or backup the code that was generated previously".
- A red box surrounds the "Set all free pins as analog (to optimize the power consumption)" checkbox with the text: "Here you can enable all pins as analog to save power".

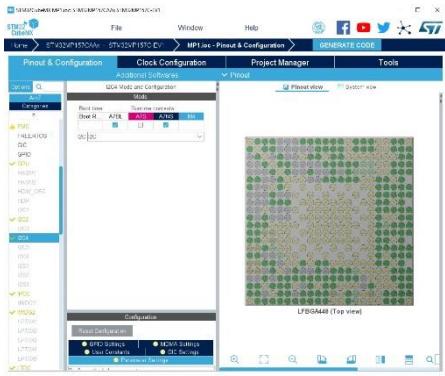


STM32WBx ecosystem

Complete set of tools to ease your design

From an Idea to a certified product

STM32
CubeMX



STM32CubeMX

Graphical tool
for easy configuration

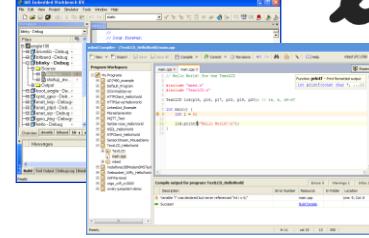
- Configure and generate code
- Peripherals and middleware configuration

STM32
CubelDE

eclipse

arm KEIL

iar



IDEs
Compile and debug

Simple,
powerful solutions

- Partners IDE (Arm® Keil®) **FREE**
- IDE based on Eclipse **FREE**
- RTOS aware debug



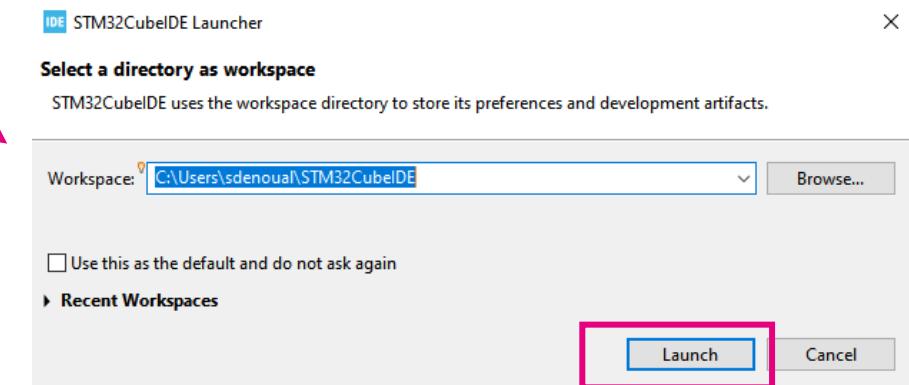
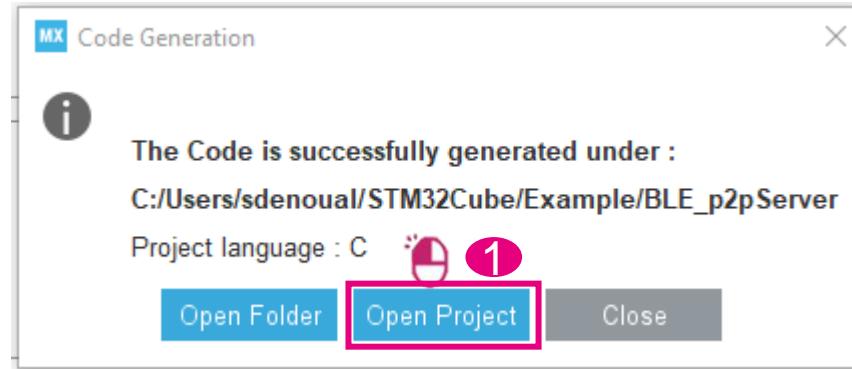
STM32 programming
& monitoring tools

STM32CubeProg
STM32CubeMonitor
STM32CubeMonRF

- Device and memory configuration
- Program the application
- Monitor variables at runtime , Test RF



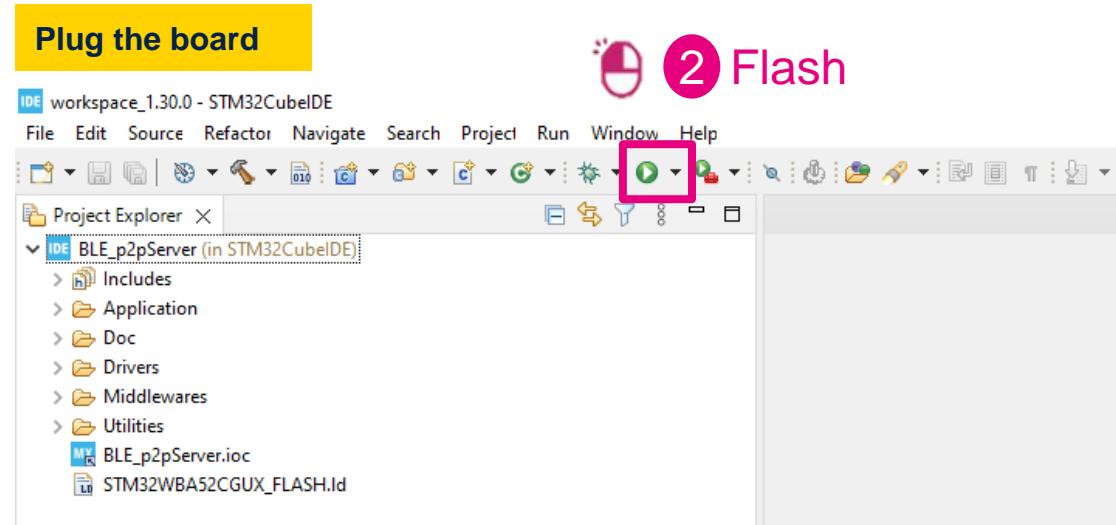
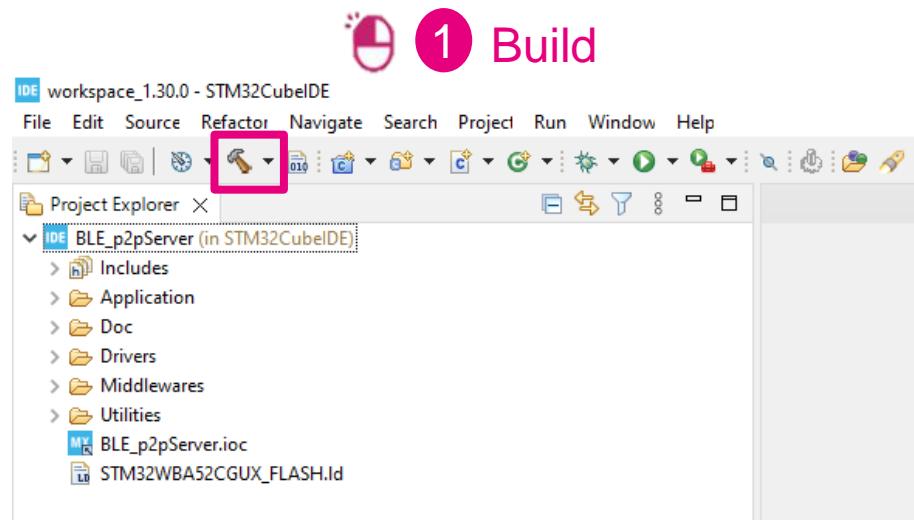
Open project





AVNET[®] SILICA

Build and flash modified project

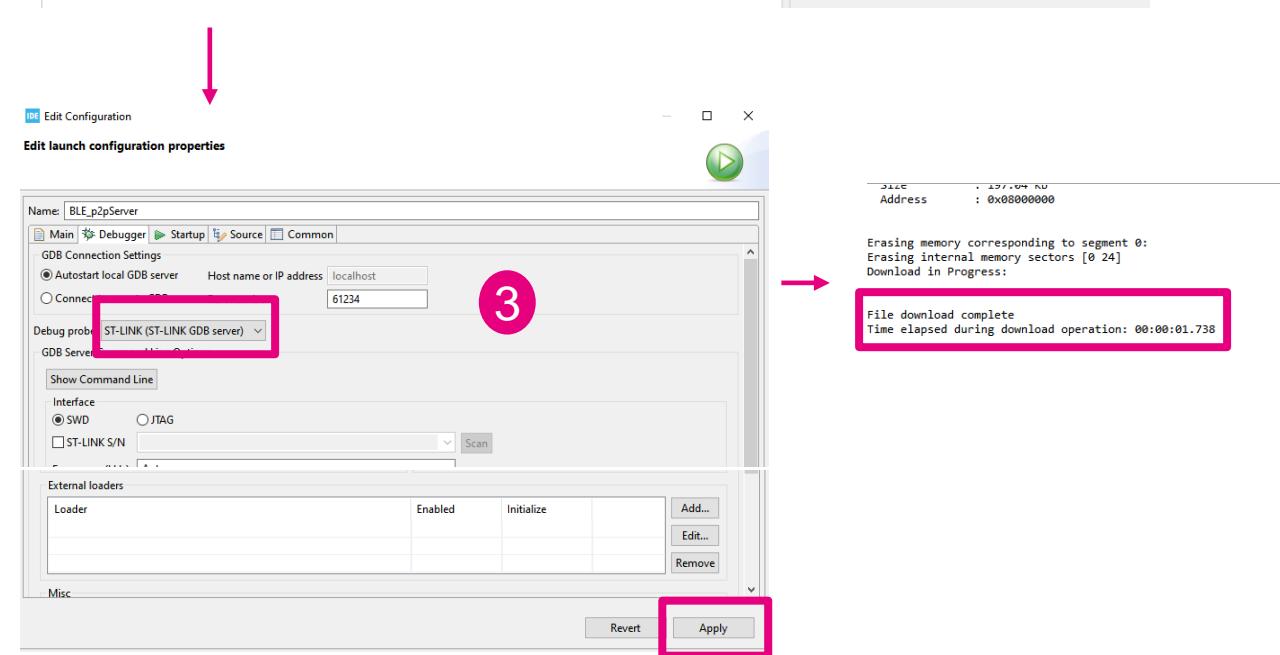


CDT Build Console [BLE_p2pServer]

```
200400 1372 56920 258692 3f284 BLE_p2pServer.elf
arm-none-eabi-objcopy -O binary BLE_p2pServer.elf "BLE_p2pServer.bin"
Finished building: default.size.stdout

Finished building: BLE_p2pServer.bin
Finished building: BLE_p2pServer.list

17:16:14 Build Finished. 0 errors, 0 warnings. (took 30s.161ms)
```





Code change with STM32 IDE

The screenshot shows the STM32CubeIDE interface with the project 'BLE_p2pServer' open. The left pane displays the Project Explorer with various source files and folders. The right pane shows the code editor with the file 'app_ble.c' open. A red box highlights a specific section of the code where the user has modified the advertising data. The original code was:

```
189 /* Advertising Data */
190 uint8_t a_AdvData[23] =
```

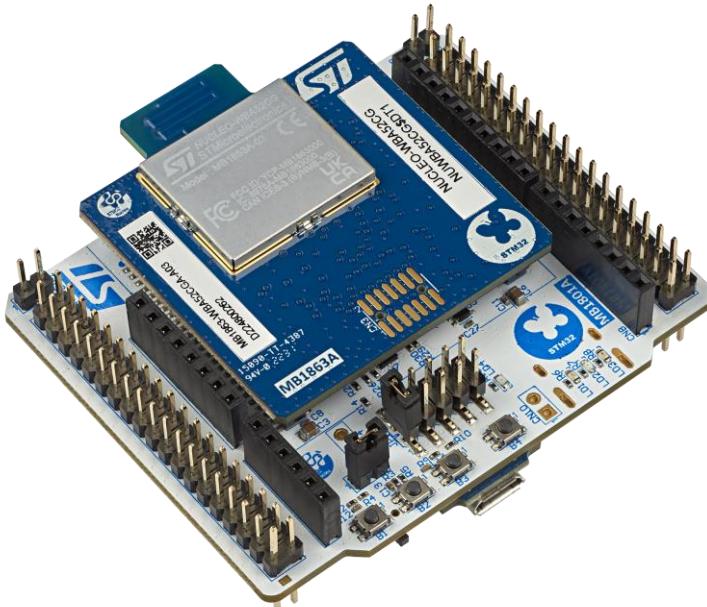
The user has added two new entries to the array:

```
92 6, AD_TYPE_COMPLETE_LOCAL_NAME, 'c', 'i', 'n', 'o', 'l', /* Complete name */
93 15, AD_TYPE_MANUFACTURER_SPECIFIC_DATA, 0x30, 0x00, 0x00 /* */, 0x00 /* */
```

The rest of the code remains the same, including comments and variable declarations.



Enjoy your first STM32WBA52 project running!



ST BLE Toolbox



BLE sequences from Advertising to PUSH BUTTON notification



UART Terminal configuration

- Terminal must be configured as defined in STM32CubeMX:

The screenshot shows the STM32CubeMX interface with the 'Parameter Settings' tab selected. On the left, a tree view lists various peripheral components. The 'USART1' node is highlighted with a blue selection bar at the bottom. The main panel displays configuration parameters for USART1, including:

Parameter	Value
Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
Data Direction	Receive and Transmit
Over Sampling	8 Samples
Single Sample	Disable
ClockPrescaler	1
Fifo Mode	Enable
Txfifo Threshold	1 eighth full configuration
Rxfifo Threshold	1 eighth full configuration
Autonomous Mode	Disable
Auto Baudrate	Disable
TX Pin Active Level Inversion	Disable
RX Pin Active Level Inversion	Disable
Data Inversion	Disable
TX and RX Pins Swapping	Disable
Others	Enable

The screenshot shows the STM32CubeMX interface with the 'Configuration' tab selected. The 'Mode' dropdown is set to 'BLE'. The 'Reset Configuration' button is visible. Below it, there are tabs for 'BLE Advertising', 'SERVICE1', 'User Constants', and 'Platform Settings'. The 'BLE Applications and Services' section shows a proposal for 'Serial Link for Traces' using 'USART-Asynchronous'. The 'BSP API' dropdown shows 'USART1'.

The screenshot shows the STM32CubeMX interface with the 'Configuration' tab selected. The 'BLE Applications and Services' tab is active. It displays a list of application configuration items under 'Project IP's Configuration' and 'Traces'. The 'Traces' section includes the following table:

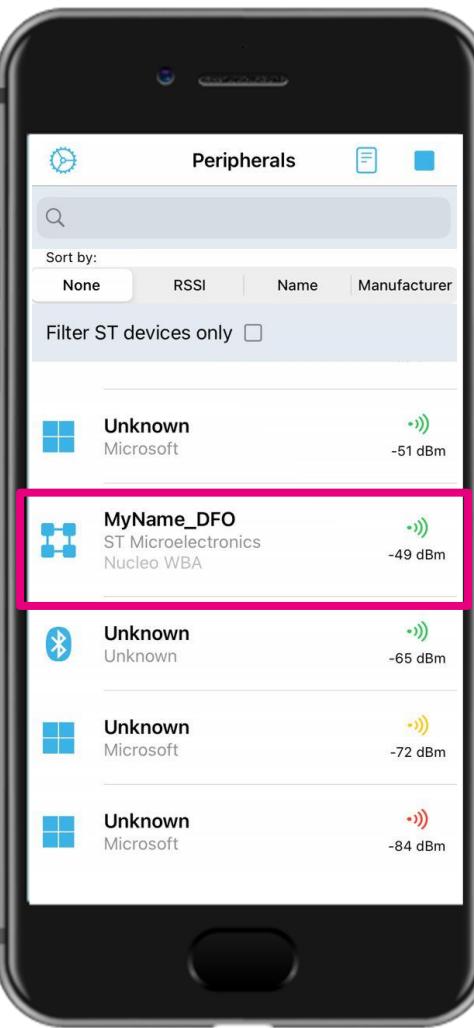
Parameter	Value
ADV_TRACE_TIMESTAMP_ENABLE	Disabled
CFG_DEBUG_APP_TRACE	Enabled
CFG_DEBUG_TRACE_LIGHT	Disabled
CFG_DEBUG_TRACE_FULL	Enabled
DBG_TRACE_USE_CIRCULAR_QUEUE	Enabled
DBG_TRACE_MSG_QUEUE_SIZE	4096
MAX_DBG_TRACE_MSG_SIZE	1024

A note at the bottom right says: "!! Please read carefully Information panel below!!".

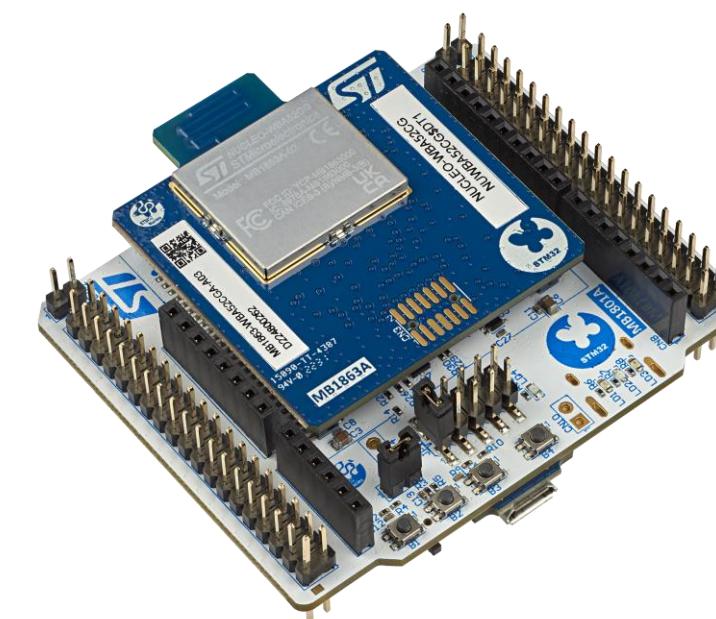
- Debug traces are also enabled:

STBLE Toolbox (Advertising)

1

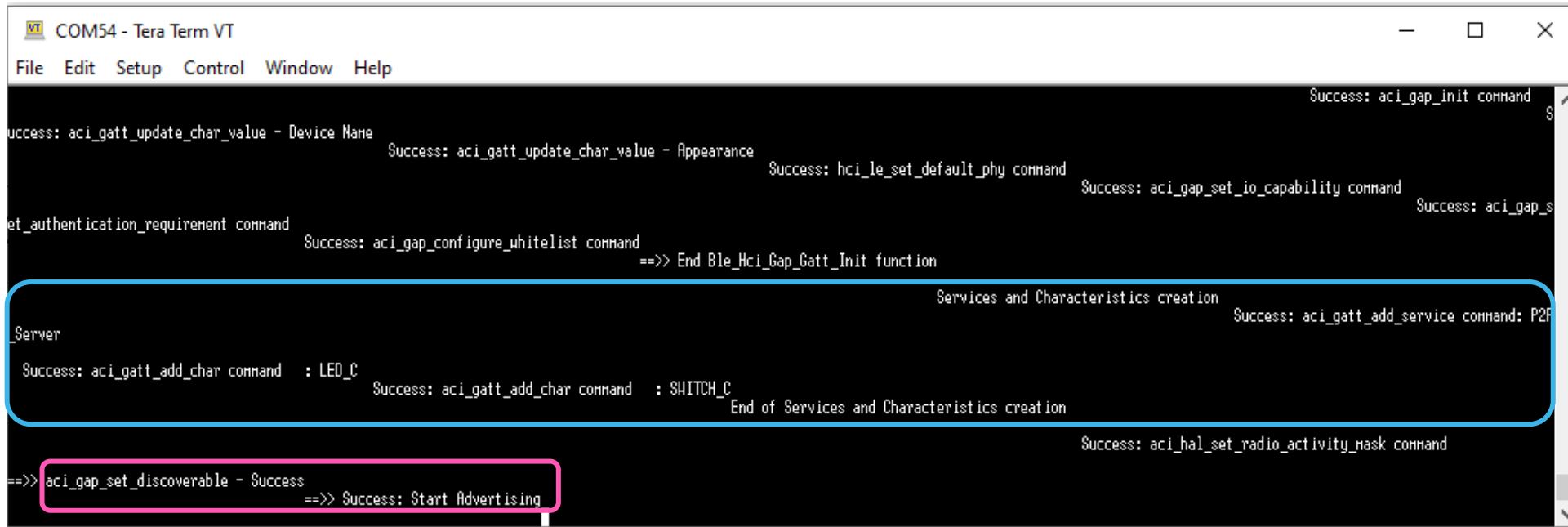


click on device



UART Terminal (Advertising)

- Via the HyperTerminal, check the “Advertising” sequence after reset:



The screenshot shows a terminal window titled "COM54 - Tera Term VT". The window has a menu bar with File, Edit, Setup, Control, Window, and Help. The main area displays a series of log messages from a BLE stack. The messages include:

- Success: aci_gatt_update_char_value - Device Name
- Success: aci_gatt_update_char_value - Appearance
- Success: hci_le_set_default_phy command
- Success: aci_gap_set_io_capability command
- Success: aci_gap_start command
- Success: aci_authentication_requirement command
- Success: aci_gap_configure_whitelist command
- ==>> End Ble_Hci_Gap_Gatt_Init function
- Services and Characteristics creation
- Success: aci_gatt_add_service command: P2P_S
- _Server
- Success: aci_gatt_add_char command : LED_C
- Success: aci_gatt_add_char command : SHITCH_C
- End of Services and Characteristics creation
- Success: aci_hal_set_radio_activity_mask command
- ==>> aci_gap_set_discoverable - Success
- ==>> Success: Start Advertising

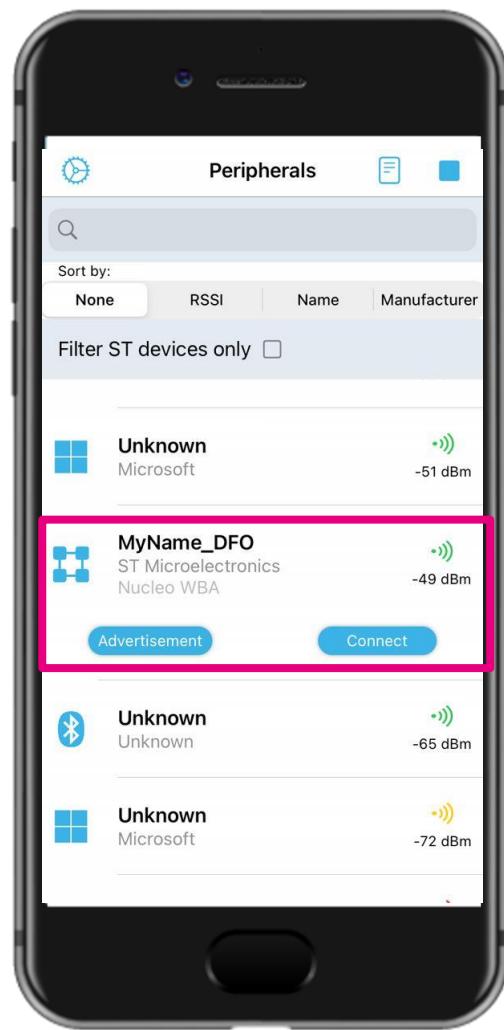
Two specific lines are highlighted with colored boxes: "Services and Characteristics creation" (blue box) and "==>> aci_gap_set_discoverable - Success" (pink box).

1. Creation of P2P service & characteristics

2. Advertising start

STBLE Toolbox (Connection)

1



2



 click on connect
life.augmented

access to profile



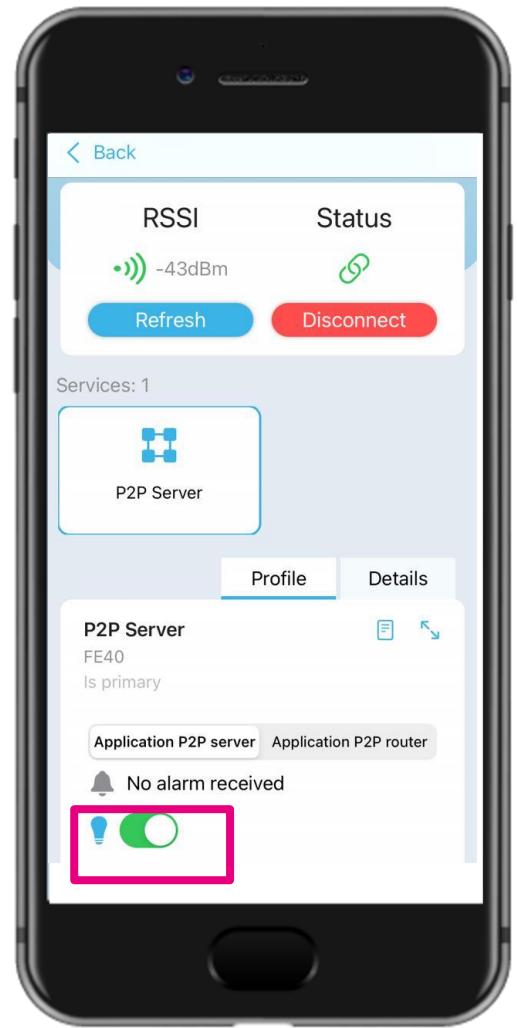


UART Terminal (Connection & P2P profile)

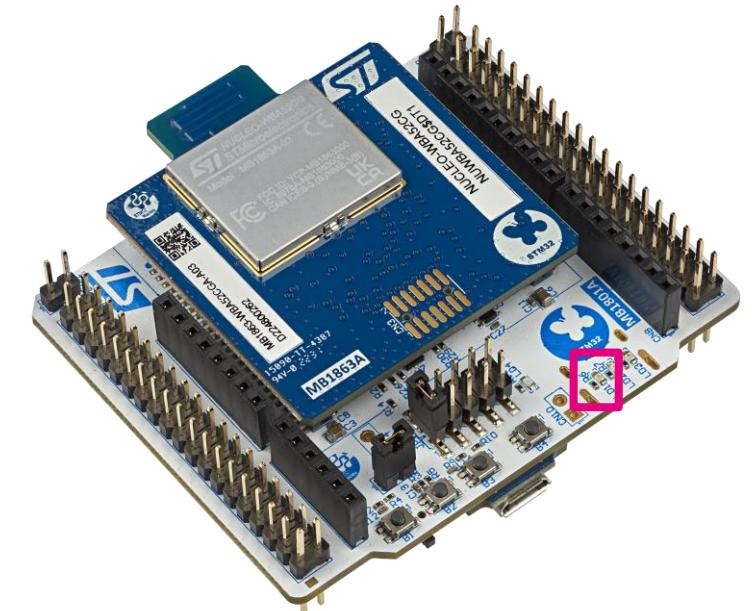
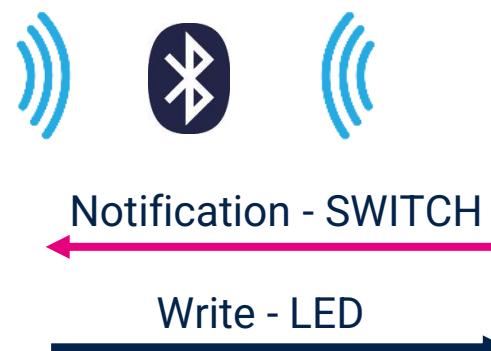
- Via the HyperTerminal, check the Connection sequence & P2P profile activation

```
=>> aci_gap_set_discoverable - Success
      ==>> Success: Start Advertising
      >>== HCI_LE_CONNECTION_COMPLETE_SUBEVT_CODE - Connection handle: 0x0001
      - Connection established with 0:6a:18:43:50:78:8d
      >>== HCI
LE_CONNECTION_UPDATE_COMPLETE_SUBEVT_CODE
      >>== HCI_LE_CONNECTION_UPDATE_COMPLETE_SUBEVT_CODE
      -- P2P APPLICATION SERVER : NOTIFICATION ENABLED
```

STBLE Toolbox (LED)



control LED status on
Nucleo





UART Terminal (LED)

- Then check the LED1 characteristic:

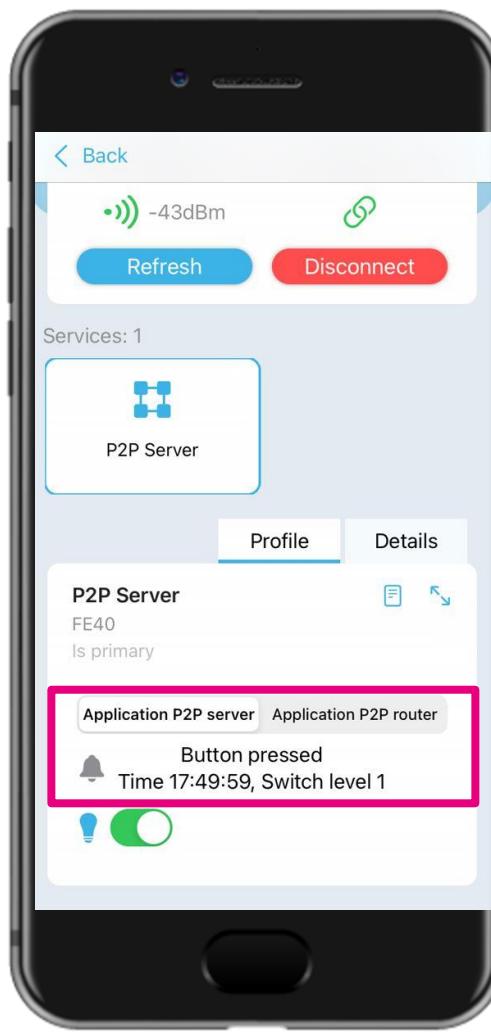
```
_LE_CONNECTION_UPDATE_COMPLETE_SUBEVT_CODE
    >>= HCI_LE_CONNECTION_UPDATE_COMPLETE_SUBEVT_CODE
        -- P2P APPLICATION SERVER : NOTIFICATION ENABLED

-- GATT : LED CONFIGURATION RECEIVED
    -- P2P APPLICATION SERVER : LED1 ON
    -- GATT : LED CONFIGURATION RECEIVED
    -- P2P APPLICATION SERVER : LED1 OFF
```



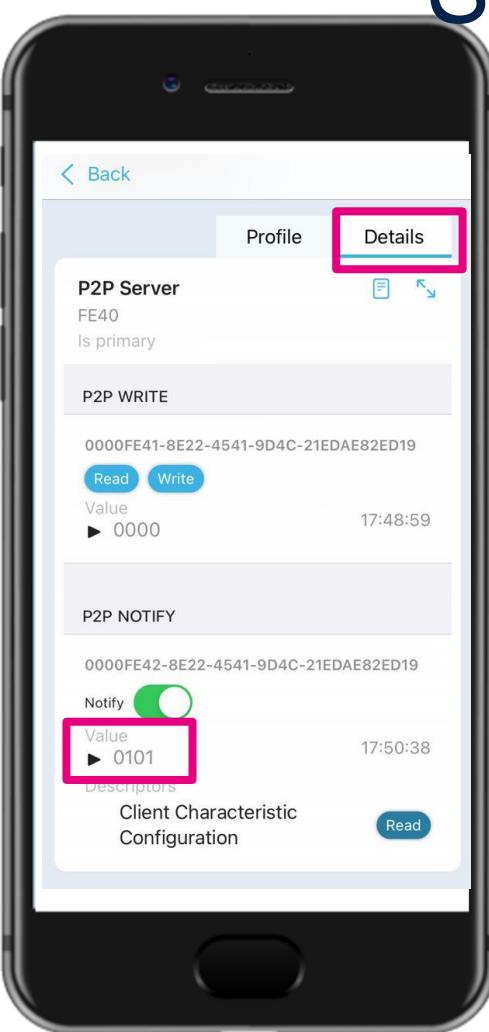
STBLE Toolbox (Push Button)

1



push button 1 and
notify device

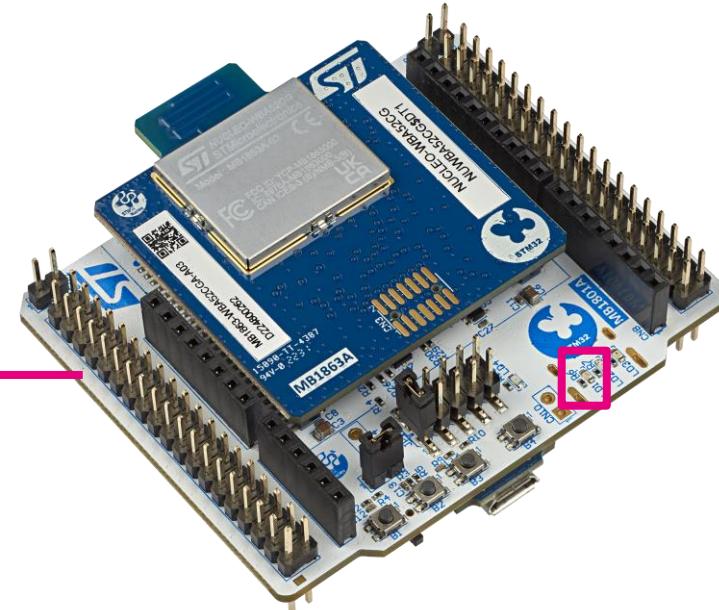
2



click on details to see
bytes sent/received



Notification - SWITCH





UART Terminal

- Connect an HyperTerminal to check application :

```
TERMINAL 1: C:\> Button 1 pressed  
-- P2P APPLICATION SERVER : INFORM CLIENT BU  
ITON 1 PUSHED Success: aci_gatt_update_char_value SWITCH_0 command
```

HW project development & certification



A complete set of documentation

AN5948 : Development of RF Hardware using STM32WBA

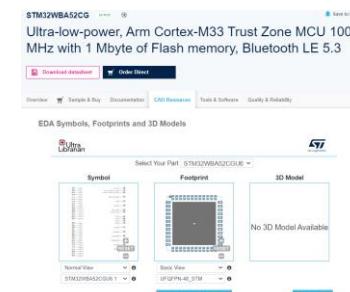


- RF basis generalities
- Schematics & components selection guidelines
- STM32WBA5x layout checklist & guidelines
- Soon on st.com

STM32WBA5x resources on st.com



MB1863
STM32WBA52/54
QFN48
PCB antenna

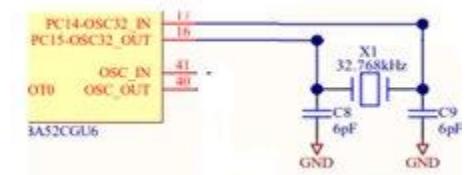
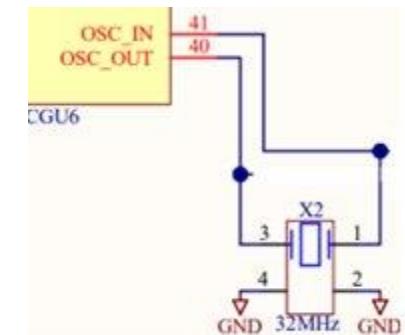


→ Multiple reference design available for various part numbers & packages

→ Download STM32WBA5x symbol
→ Download STM32WBA5x footprint

HW design with STM32WBA5x : key points

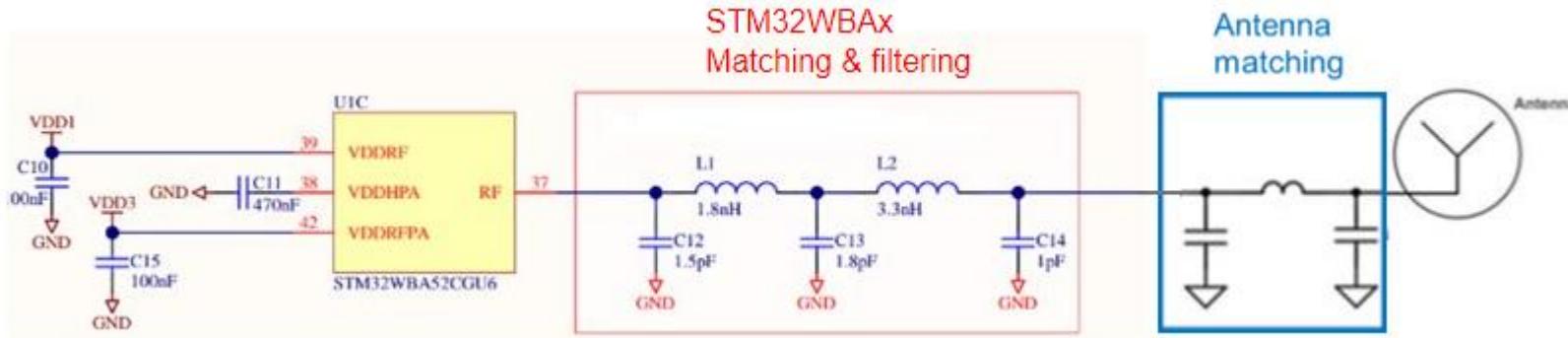
- HW design should be initiated based on documentation shown on previous slides
- Next slides are highlighting the key points you must pay attention when designing schematics and layout:
 - HSE 32MHz Xtal requirements:
 - STM32WBA5x includes internal programmable capacitances to trim the crystal frequency
 - No need external load capacitances
 - But, select 8pF load cap XTAL
 - Recommended part (or equivalent): NX1612SA-32MHZ-EXS00A-CS09166
 - Refer to AN5042 or ST team
 - LSE or LSI selection. LSE 32kHz Xtal requirements if used:
 - LSE mandatory for accurate RTC calendar application
 - BOM optimized (save 32kHz Xtal cost) with LSI (OSC32 pins can be used open or as IOs)
 - Refer to AN2867 or ST team



HW design with STM32WBA5x : key points

➤ RF matching & filtering:

- Integrated balun so single ended RF matching.
- Very limited number of discrete components for STM32WBAX matching and filtering.



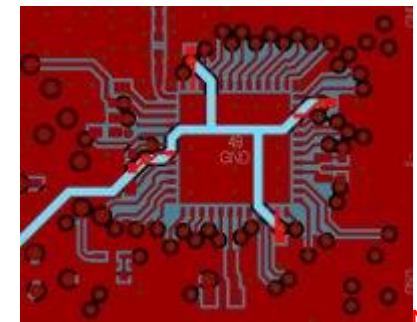
➤ Power management. SMPS implementation for STM32WBA55 (next WBA version) use case:

- STM32WBA55 embeds a SMPS that can be used to improve power efficiency.
- Apply different power supply scheme for output power target > 4dBm.

➤ Main layout recommendation:

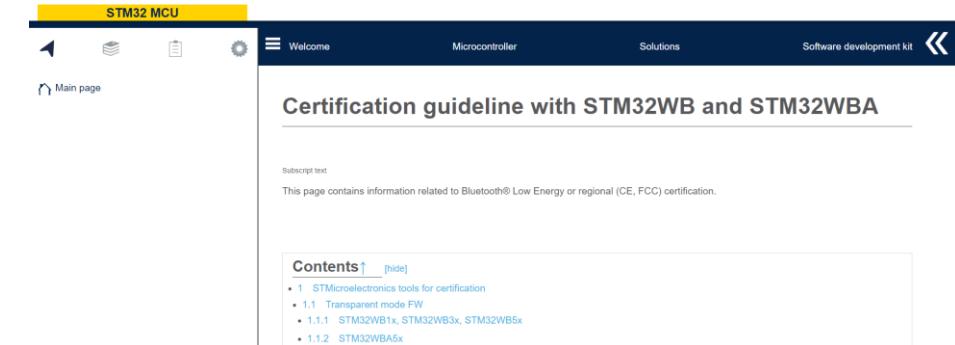


- Refer our various reference kits layout (Gerbers and Altium files available).
- 4 layers stack-up recommended but 2 layers one is possible.



STM32WBA5x : a certified solution

- STM32WBA5x is compliant in regards of regional (CE, FCC etc.) and Bluetooth requirements.
- We are providing complete set of documentation, FW and tools to certify your product.



Full set of tools and documentation

- Certification guideline on wiki
https://wiki.st.com/stm32mcu/wiki/Category:BLE_Certification
- Transparent mode FW available in [STM32CubeWBA](#) MCU Package
- [STM32CubeMonRF](#) PC tool

Bluetooth certification

- STM32WBA5x is having reference QDIDs (components and stack) at Bluetooth SIG.
- Customer must performed PHY testing and declare its product at Bluetooth SIG.

PHY QDID

Package	Part number	Cut version	RF PHY QDID
QFN48	STM32WBA52 (BLE5.4)	1.x	197135 (TCRL 2022-2)

Stack QDID

Features	Host Stack version	QDID
4.0 HCI Low Energy LL with extended advertising – ATT – GAP – GATT – L2CAP with Enhanced Connected Oriented Channel -SMP BLE 5.3	STM32Cube_WBA_BLE_HCI_STACK STM32Cube_WBA_BLE_FULL_STACK	198195 (TCRL 2022-1)

PROJECT CREATION

- Project name
- Enter QDIDs (PHY and stack)
- Select core specification layer
- Select profiles and services (if using standard)

Module

NO PHY TESTING

- Attach our modules PHY test reports (available on st.com)
- SW testing if required (*)

Chipset

PHY TESTING

- PHY testing in Test house : 9600\$ (**)
- SW testing if required (*)

DECLARATION

- Declare product name and model
- Pay Listing fees : 9600\$ (**)

(*) SW testing

- Proprietary profile
- Adopted profiles

0\$
few k\$ Pending number and adopted profile used

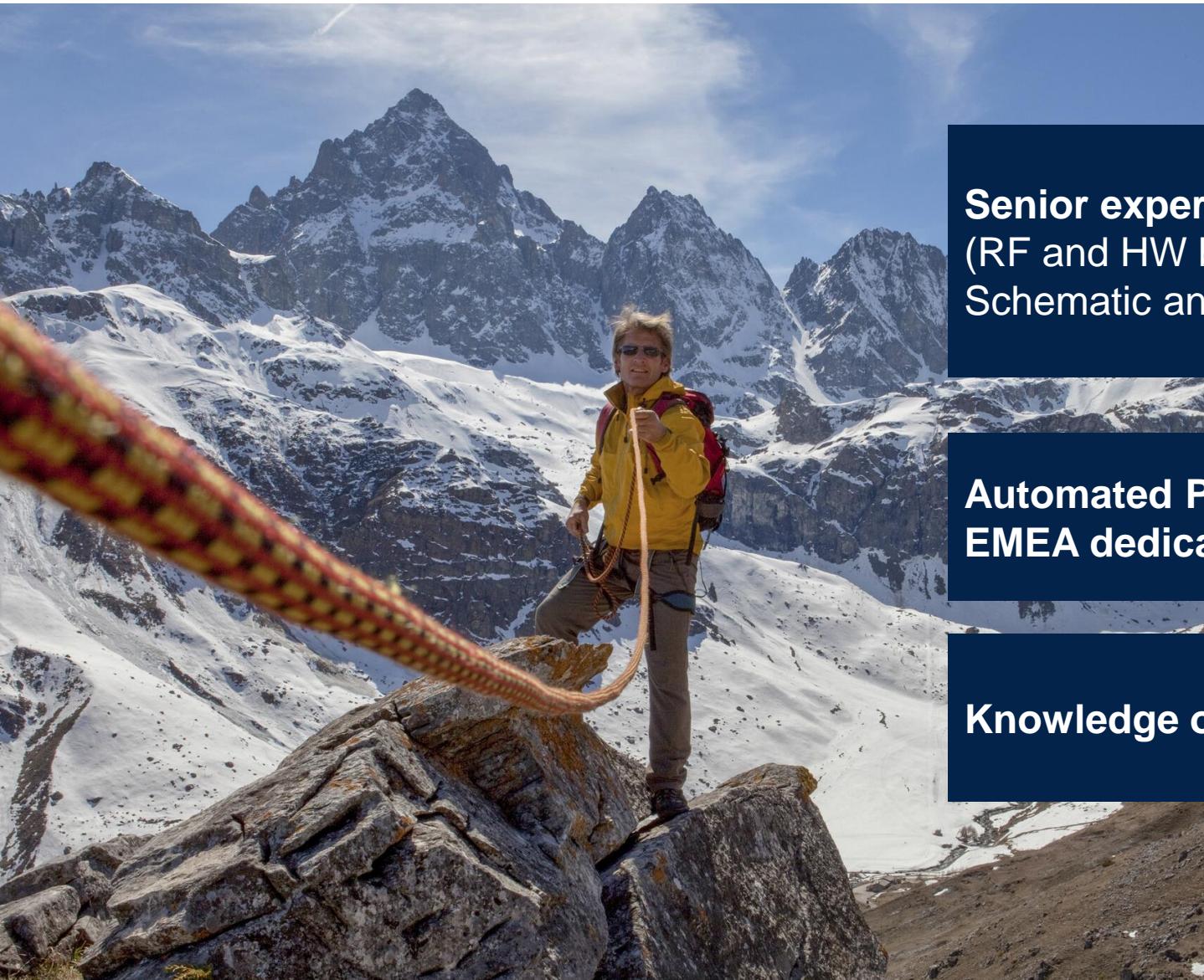
(**) Fees

- Fees are reduced if SIG member
- Fees will increase by 15% by 2024 January 1st

https://wiki.st.com/stm32mcu/wiki/Category:BLE_Certification

ST EMEA support

What you can expect from our wireless support team



Senior expertise in wireless communication
(RF and HW bring-up services, SW and HW debugging,
Schematic and layout review)

Automated Pre-certification capabilities
EMEA dedicated application labs

Knowledge on your application

... in a fast and agile way



Our customer wireless journey and how we can help

Support them to make the right choice

Make sure their choice was the right one

ST Product selection

- Training
- Benchmark
- Datasheet review

ST Product evaluation

- Eval kit bring-up
- Performance review

SW and HW Development

- Schematic review
- Layout review
- Software porting
- 1st PCB bring-up
- 1st RF test report
- Debugging

Testing and certification

- Regional RF test report
- Protocol RF tests
- Corner case debugging
- Certification process guidance

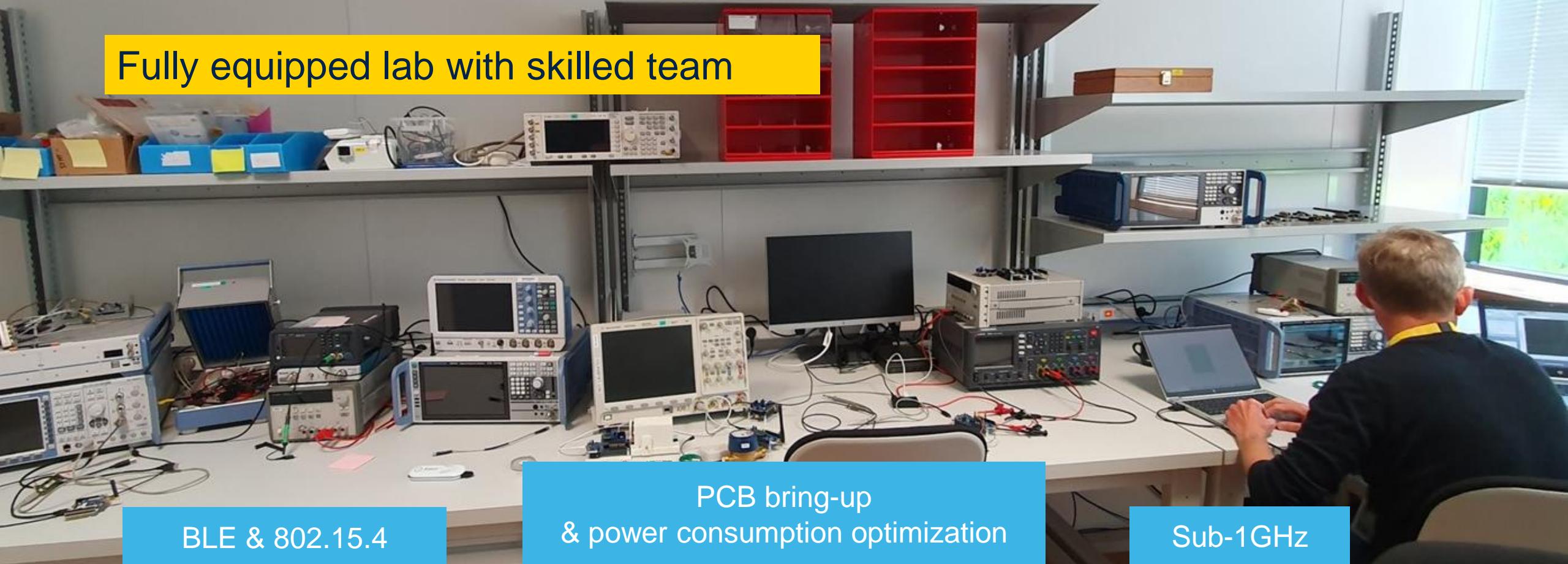
Industrialization

- RF Production tooling
- Ramp-up support

Don't wait too long to ask for any support.
The sooner the better!!

We look forward to receiving customers' PCBs!

Fully equipped lab with skilled team



BLE & 802.15.4

PCB bring-up
& power consumption optimization

Sub-1GHz



Bluetooth®



zigbee



THREAD



Wi-Fi



LoRa™



NFC



M-Bus
wireless

STM32WBA takeaway



STM32WBA takeaway 1#2



2.4Ghz Portfolio

Wide portfolio from NP to AP
from M0 to M33 – single or dual core

Wireless

Bluetooth® Low Energy 5.3 certified
Multiprotocol

Performance

Arm® Cortex®-M33 at 100 MHz
Fast wake-up time , high RF power 10dBm

Power efficiency

Extended battery lifetime
Autonomous low-power mode

Security

As a MUST to have !
TrustZone® - DPA resistant – SESIP L3

Integration

1 Mbyte of flash memory, 128 Kbytes RAM
Reduced BOM



STM32WBA takeaway 2#2



Cost Effective

new architecture , 40nm

Large Capacity

Massive investment to support large production now and onward

Free ecosystem

Faster time to market with STM32CubeMX Enhanced design journey

RF lab

Schematic & layout review to quickly go in production

Our technology starts with You



Find out more at www.st.com

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to www.st.com/trademarks.

All other product or service names are the property of their respective owners.