life.augmented

Hands-on #1
Build basic p2pServer application and connect

STM32

Agenda

1 Hands-on presentation

2 Step 1: STM32CubeMX initialization for STM32WBA Nucleo board

3 Step2 : Advertising and BLE application configuration and explanation

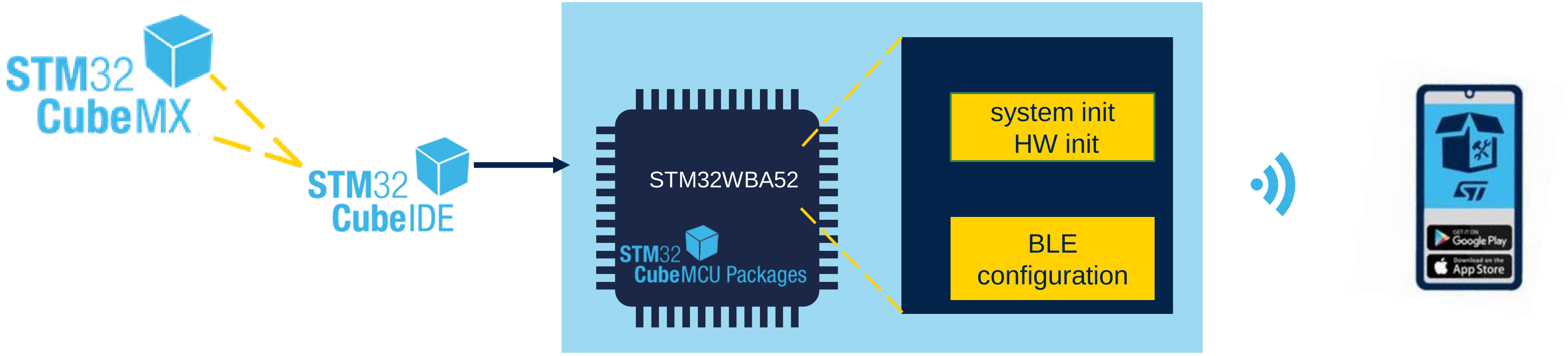
4 Step 3 : Code generation and user application code

5 Step 4 : Adding logs

Hands-On presentation

Purpose

- The purpose is to start from WB5A52 chipset level and build a basic server (p2pServer) application using STM32CubeMX and associated STM32CubeIDE
- In this first part, focus is to get device visible and connectable from my smartphone



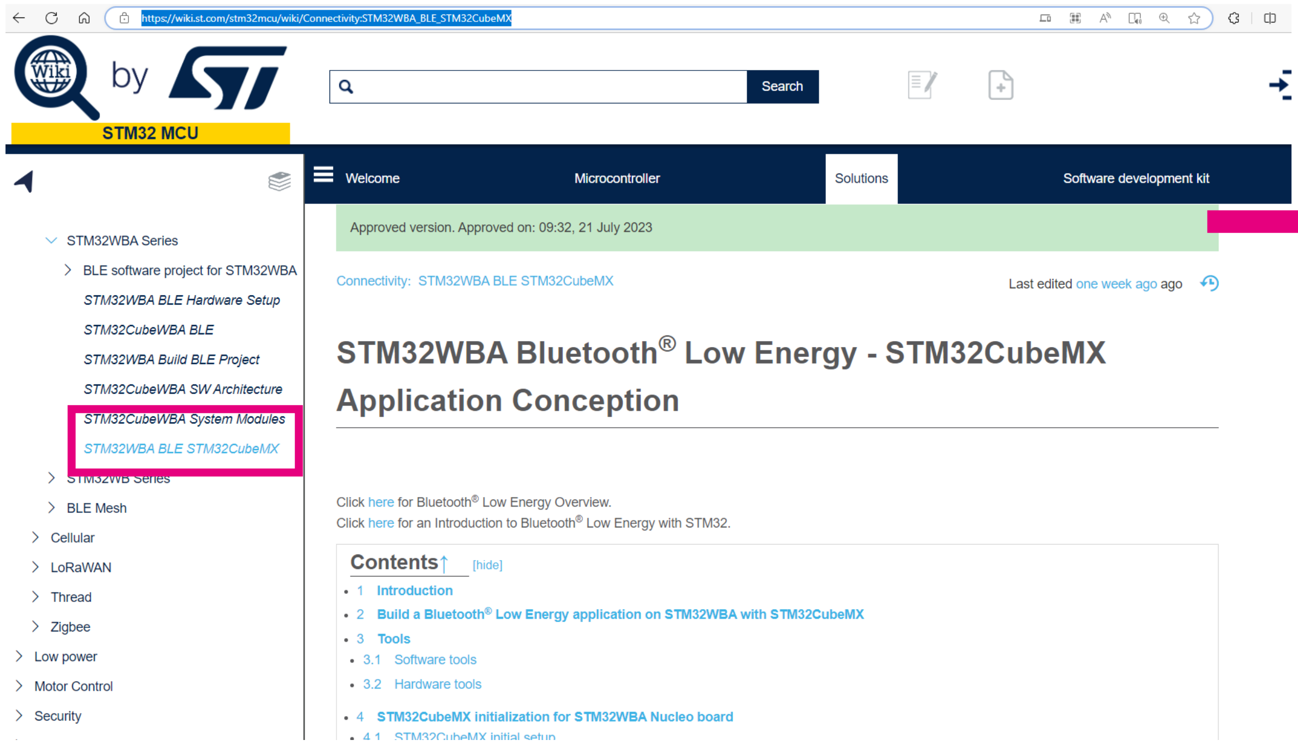
BLE server/peripheral advertising

Unpack NUCLEO-WBA52, plug to laptop,
install your favorite ST BLE ToolBox App and Let's start !

Source

STM32 MCU

Hands-On based on
https://wiki.st.com/stm32mcu/wiki/Connectivity:STM32WBA_BLE_STM32CubeMX



https://wiki.st.com/stm32mcu/wiki/Connectivity:STM32WBA_BLE_STM32CubeMX

Approved version. Approved on: 09:32, 21 July 2023

Connectivity: STM32WBA BLE STM32CubeMX

Last edited one week ago ago

STM32WBA Bluetooth® Low Energy - STM32CubeMX Application Conception

Click here for Bluetooth® Low Energy Overview.
Click here for an Introduction to Bluetooth® Low Energy with STM32.

Contents [hide]

step by step guideline to build a BLE peripheral application

BLE server/peripheral
exposing data to central device

life.augmented

Legenda

- Slides including following symbol are purely theoretical ones

- Optional steps during development in are marked with a grey bar

- Source code for development is included inside pink boxes

```
HAL_Delay(500);
```

Step 1 : STM32CubeMX initialization for STM32WBA Nucleo board

life.augmented

STM32CubeMx capabilties

STM32CubeMx allow to start design within 3 options

1	**Example application** complete application running over NUCLEO
2	**Board level** all the hardware is already configured (NUCLEO_WBA52)
3	**Chipset level** require to configure your HW (PCB) & your application

STM32WBA wiki page focus

Hands-on focus. As customer let’s build my own App

STM32CubeMx design from chispet level complete journey

STM32CubeMx initialisation for STM32WBA Nucleo board

STM32WBA IPs & peripherals configuration

Clock Tree configuration

BLE configuration : Advertising, Service, Characteristic

Code generation & application code management over CubeIDE

Hands-on Focus

STM32CubeMx design from chispet level Hands-on focus (1/2)

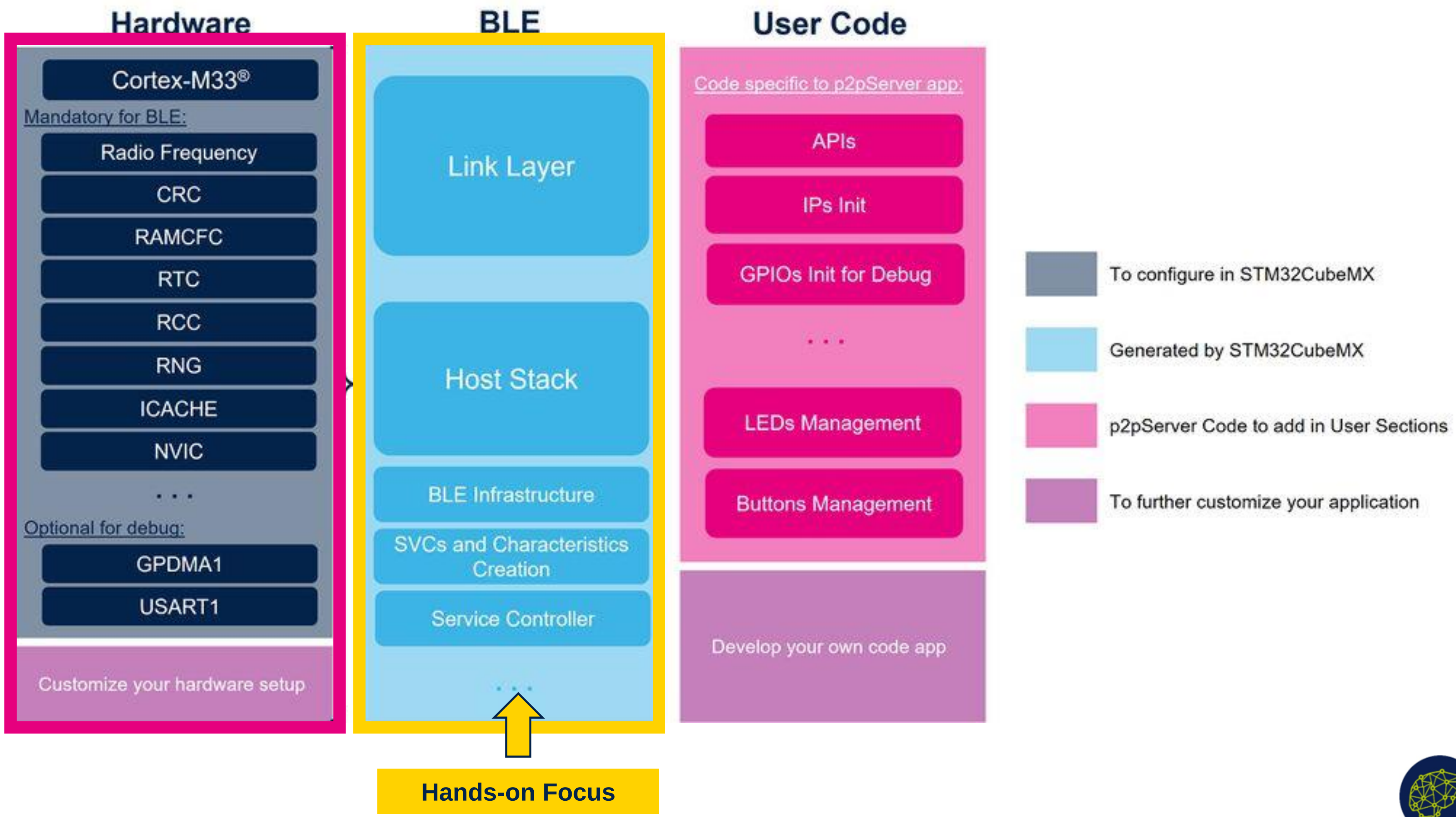
3 **Chipset level**
require to configure your HW (PCB) & your application

To ease Hands-on session use Hands-on_WS_WBA52.ioc
All HW IPs & required peripheral to use RF are already initialized : NVIC, RNG, RCC,...
Thanks to Hands-on_WS_WBA52.ioc let's focus on BLE application design

Copy Hands-on_WS_WBA52.ioc on your local repository :
example : C:\users\...\STM32WBA_WS\project

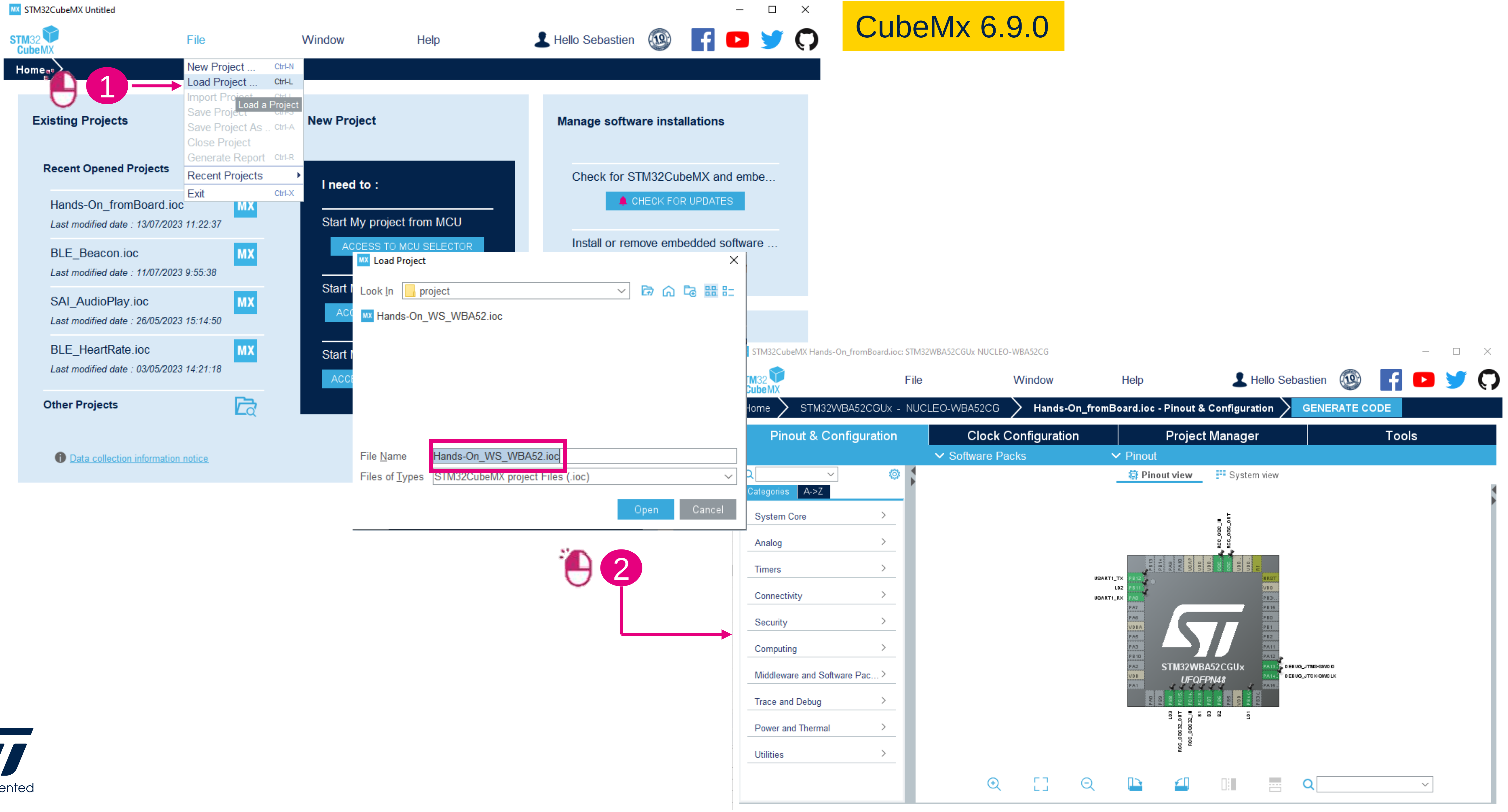
STM32CubeMx design from chispet level Hands-on focus (2/2)

Hands-On_WS_WBA52.ioc

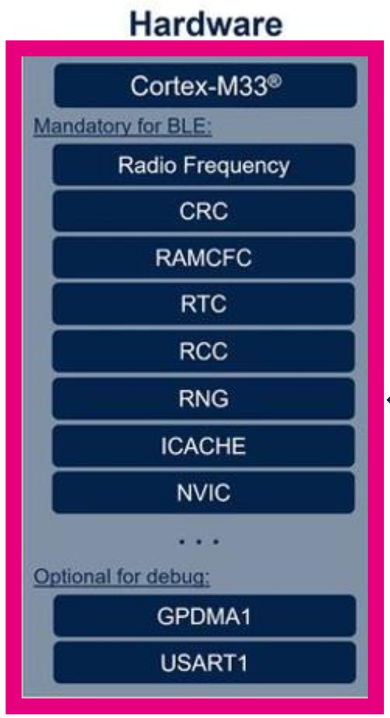


Start STM32CubeMX

CubeMx 6.9.0



Peripherals in place to start BLE configuration !



Hardware

Cortex-M33®

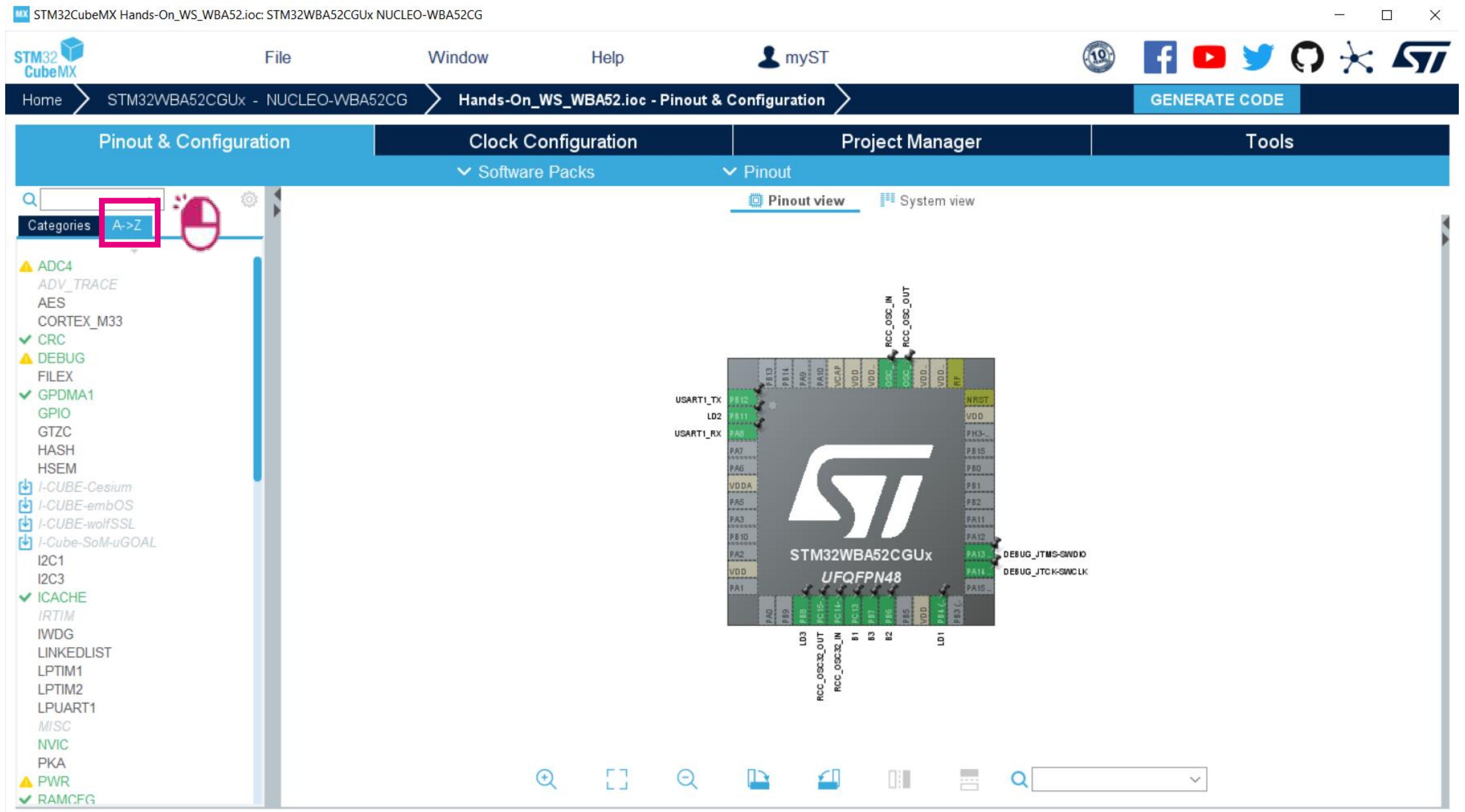
Mandatory for BLE:

- Radio Frequency
- CRC
- RAMCFC
- RTC
- RCC
- RNG
- ICACHE
- NVIC
- . . .

Optional for debug:

- GPDMA1
- USART1

Hands-On_WS_WBA52.ioc



STM32CubeMX Hands-On_WS_WBA52.ioc: STM32WBA52CGUx NUCLEO-WBA52CG

- ⚠ ADC4
- ADV_TRACE
- AES
- CORTEX_M33
- ✓ CRC
- ⚠ DEBUG
- FILEX
- ✓ GPDMA1
- GPIO
- GTZC
- HASH
- HSEM
- I-CUBE-Cesium
- I-CUBE-embOS
- I-CUBE-wolfSSL
- I-Cube-SoM-uGOAL
- I2C1
- I2C3
- ✓ ICACHE
- IRTIM
- IWDG
- LINKEDLIST
- LPTIM1
- LPTIM2
- LPUART1
- MISC
- NVIC
- PKA
- ⚠ PWR
- ✓ RAMCFG

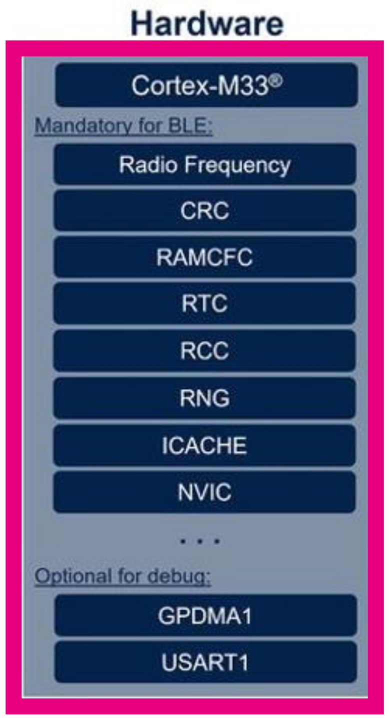
Hands-On_WS_WBA52.ioc

- HW configuraton
- enable STM32_WPAN (**BLE middleware activation**)

Peripherals in place to start BLE configuration ! Wiki explanations

by STM32 MCU

https://wiki.st.com/stm32mcu/wiki/Connectivity:STM32WBA_BLE_STM32CubeMX


Hardware

Cortex-M33®

Mandatory for BLE:

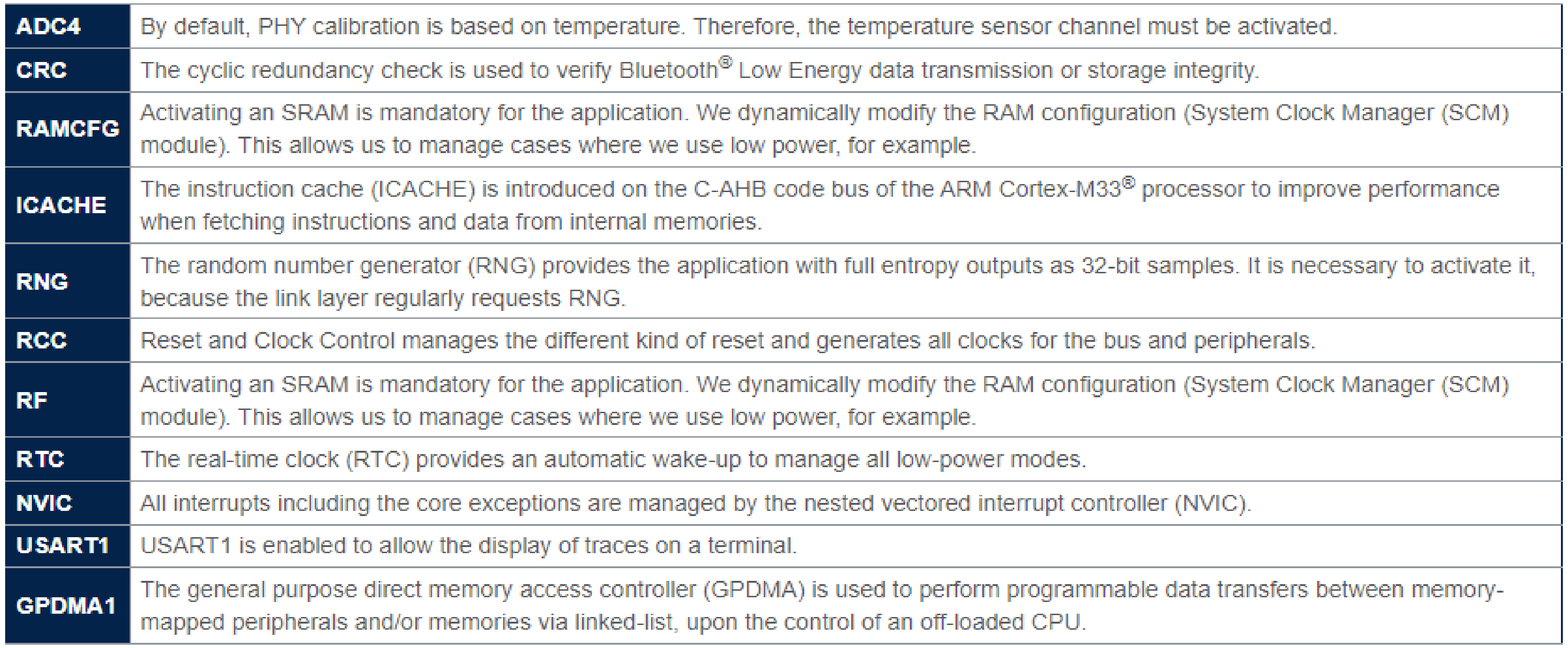
- Radio Frequency
- CRC
- RAMCFC
- RTC
- RCC
- RNG
- ICACHE
- NVIC
- . . .

Optional for debug:

- GPDMA1
- USART1

BLE activation

debug



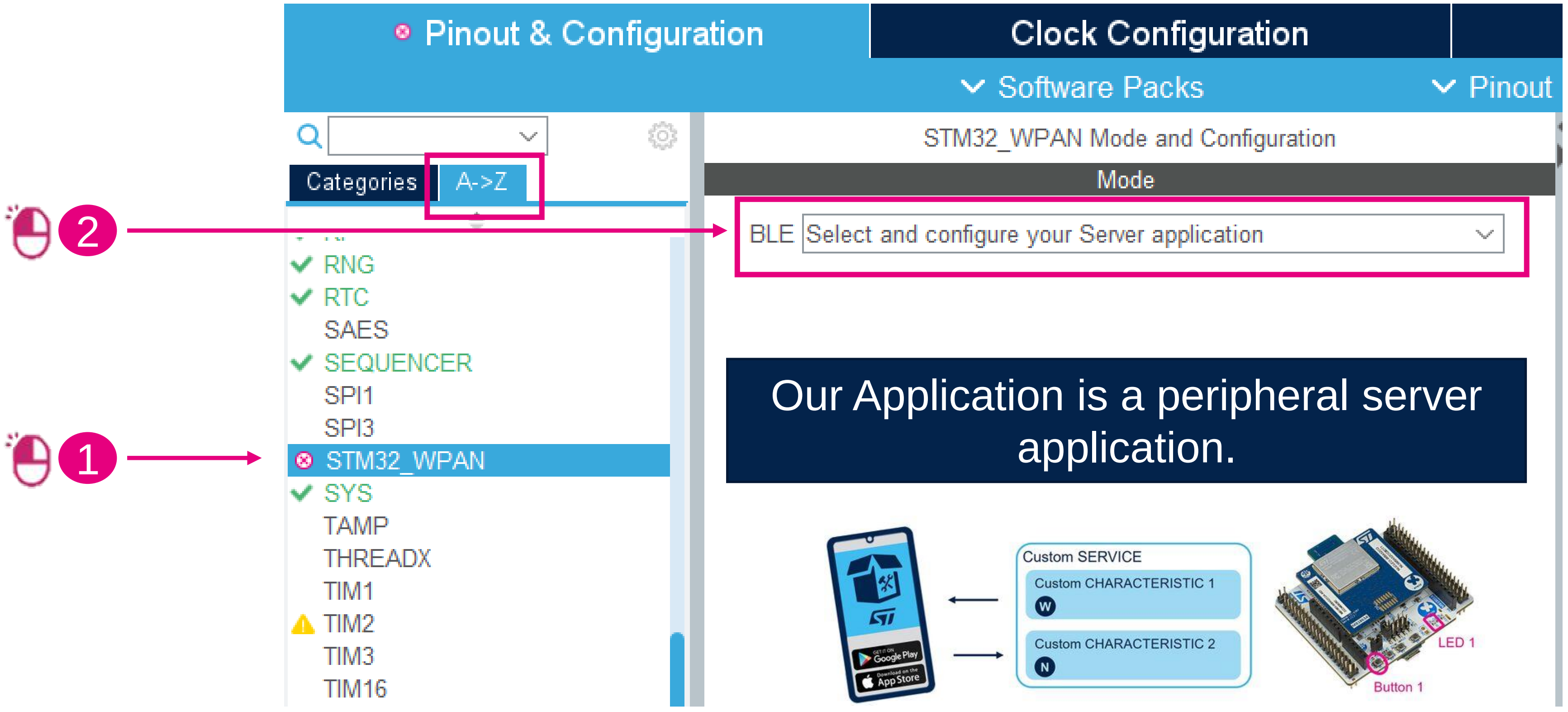
ADC4	By default, PHY calibration is based on temperature. Therefore, the temperature sensor channel must be activated.
CRC	The cyclic redundancy check is used to verify Bluetooth® Low Energy data transmission or storage integrity.
RAMCFG	Activating an SRAM is mandatory for the application. We dynamically modify the RAM configuration (System Clock Manager (SCM) module). This allows us to manage cases where we use low power, for example.
ICACHE	The instruction cache (ICACHE) is introduced on the C-AHB code bus of the ARM Cortex-M33® processor to improve performance when fetching instructions and data from internal memories.
RNG	The random number generator (RNG) provides the application with full entropy outputs as 32-bit samples. It is necessary to activate it, because the link layer regularly requests RNG.
RCC	Reset and Clock Control manages the different kind of reset and generates all clocks for the bus and peripherals.
RF	Activating an SRAM is mandatory for the application. We dynamically modify the RAM configuration (System Clock Manager (SCM) module). This allows us to manage cases where we use low power, for example.
RTC	The real-time clock (RTC) provides an automatic wake-up to manage all low-power modes.
NVIC	All interrupts including the core exceptions are managed by the nested vectored interrupt controller (NVIC).
USART1	USART1 is enabled to allow the display of traces on a terminal.
GPDMA1	The general purpose direct memory access controller (GPDMA) is used to perform programmable data transfers between memory-mapped peripherals and/or memories via linked-list, upon the control of an off-loaded CPU.

life.augmented

Step2 : Advertising and BLE GAP/GATT custom application configuration

life.augmented

Enabling Bluetooth Low Energy


Pinout & Configuration

Clock Configuration

Software Packs

Pinout

Categories

A->Z

RNG

RTC

SAES

SEQUENCER

SPI1

SPI3

STM32_WPAN

SYS

TAMP

THREADX

TIM1

TIM2

TIM3

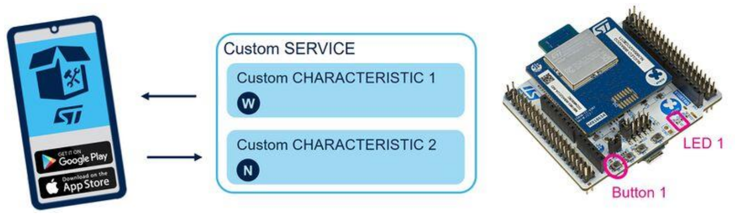
TIM16

STM32_WPAN Mode and Configuration

Mode

BLE: Select and configure your Server application

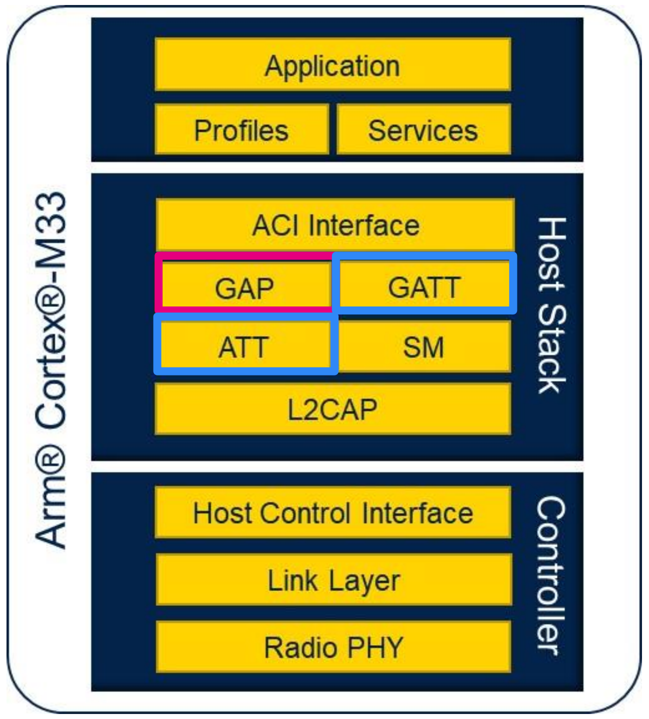
Our Application is a peripheral server application.

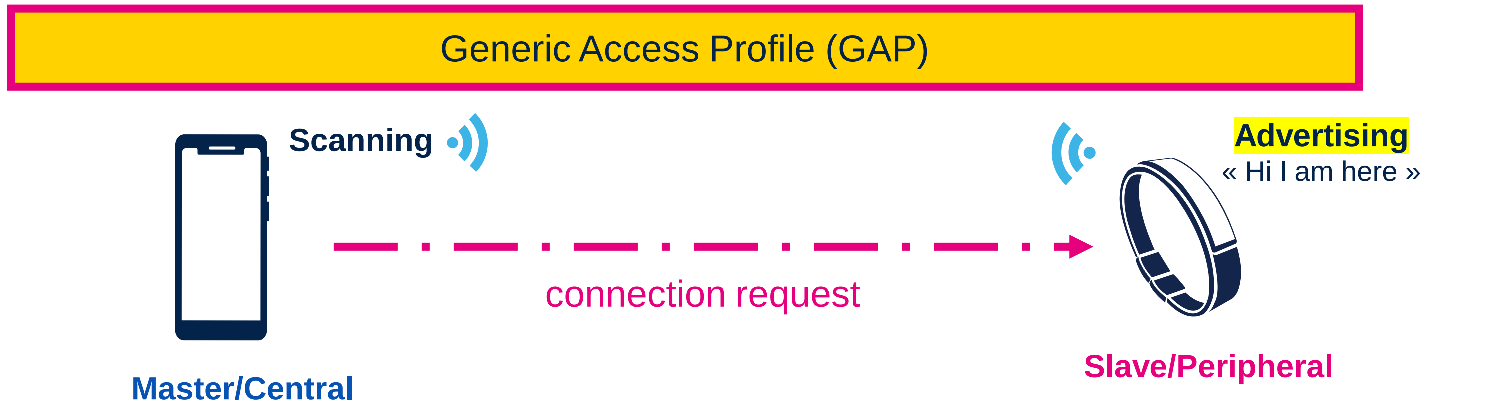


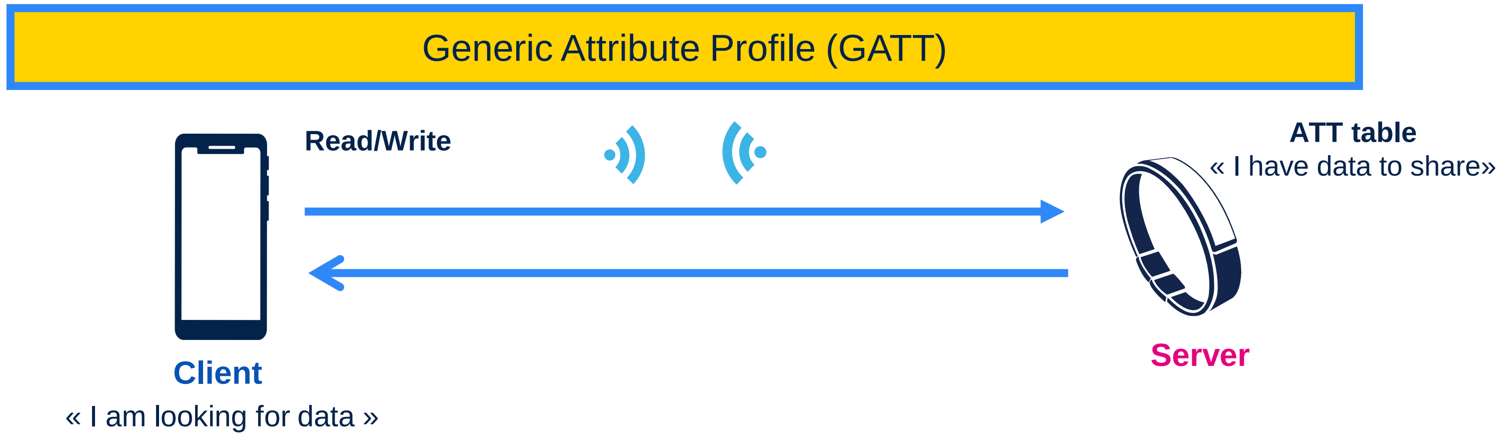
Central/Client

Peripheral/Server

Bluetooth Low Energy
Connection roles vs. Data roles

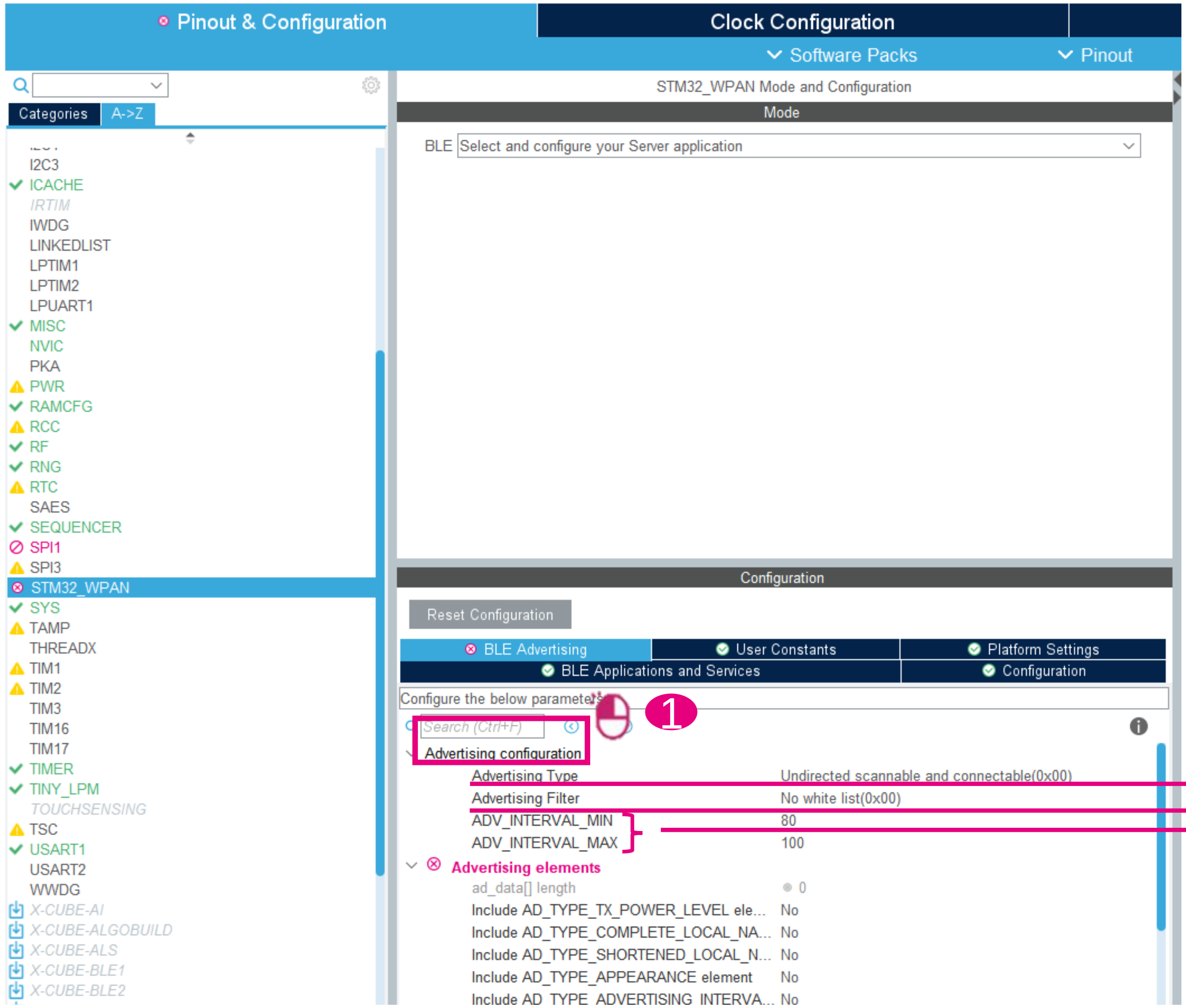






In the general run of things….
a Central is acting as GATT Client, a peripheral as a GATT server

Advertising Configuration



BLE Select and configure your Server application

Configure the below parameters :

Advertising configuration

Advertising Type	Undirected scannable and connectable(0x00)
Advertising Filter	No white list(0x00)
ADV_INTERVAL_MIN	80
ADV_INTERVAL_MAX	100

Advertising elements

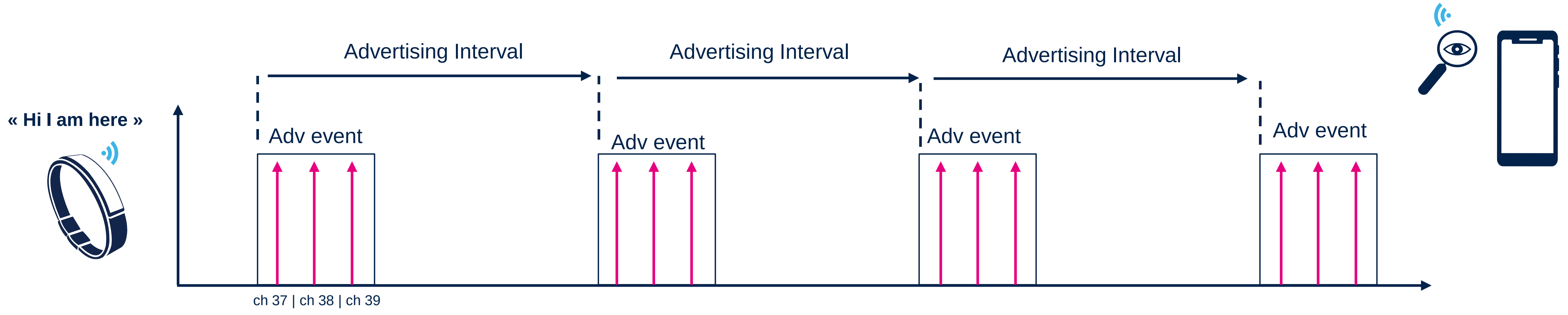
ad_data[] length	0
Include AD_TYPE_TX_POWER_LEVEL ele...	No
Include AD_TYPE_COMPLETE_LOCAL_NA...	No
Include AD_TYPE_SHORTENED_LOCAL_N...	No
Include AD_TYPE_APPEARANCE element	No
Include AD_TYPE_ADVERTISING_INTERVA...	No

Advertising Type
accept connection requests from any peer device

Advertising Filter
In general used in case of Privacy.

Advertising Interval
Advertising set = (MIN+MAX)/2
Min & Max used in case of multi connections

Advertising Configuration
Advertising Interval

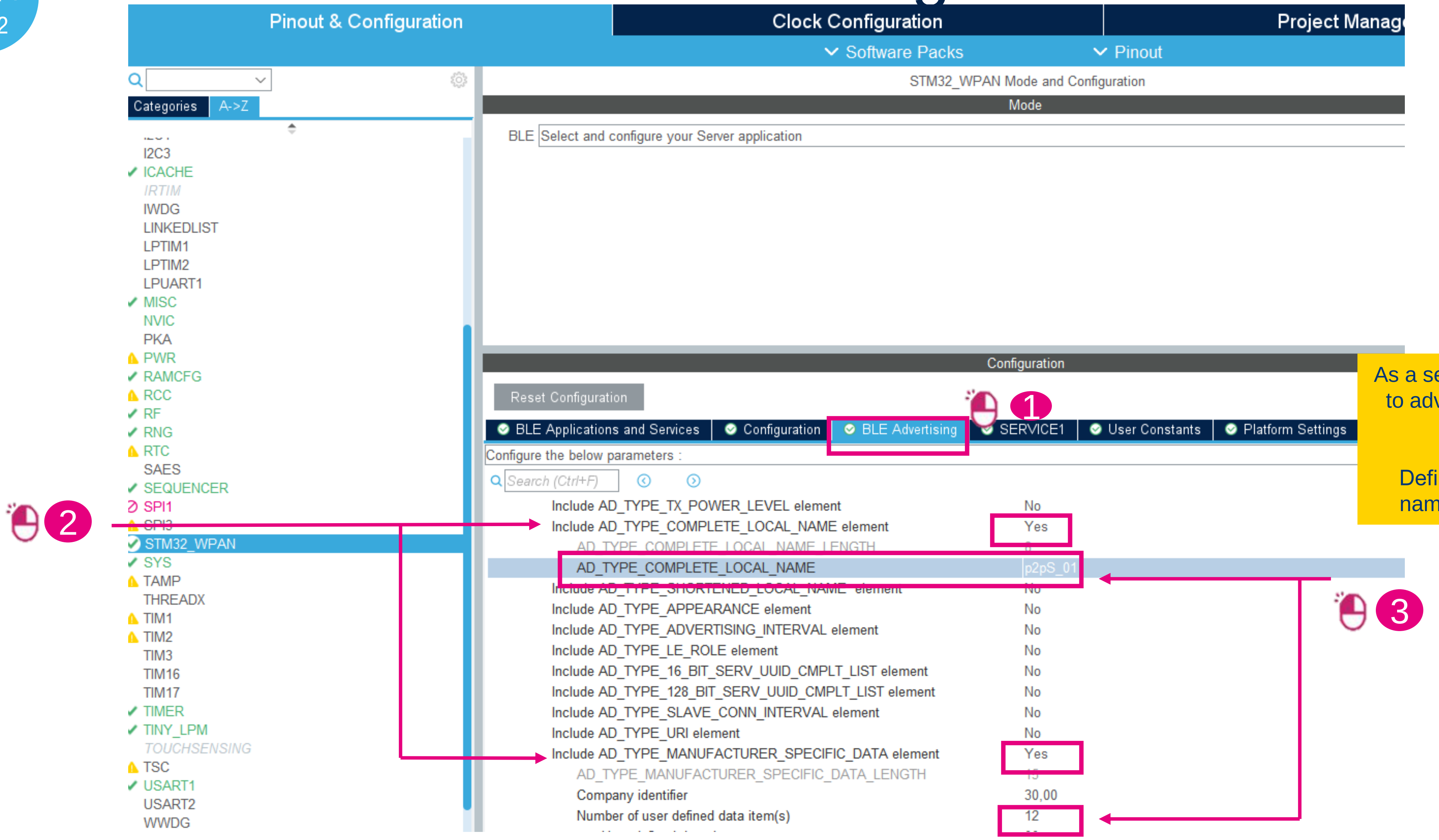


- The advertising interval value ranges all the way from **20** milliseconds up to **10.24** seconds in small increments of **625** microseconds.
- The advertising interval greatly impacts battery life and should be chosen carefully.

connectivity latency vs. power consumption efficiency

- The advertising event is the slot where peripheral will be able to push for advertising data “Hello I am here – this is my name”
- The advertising event is around **~3ms** considering legacy advertising (31 bytes)

Advertising Elements Local Name



BLE Select and configure your Server application

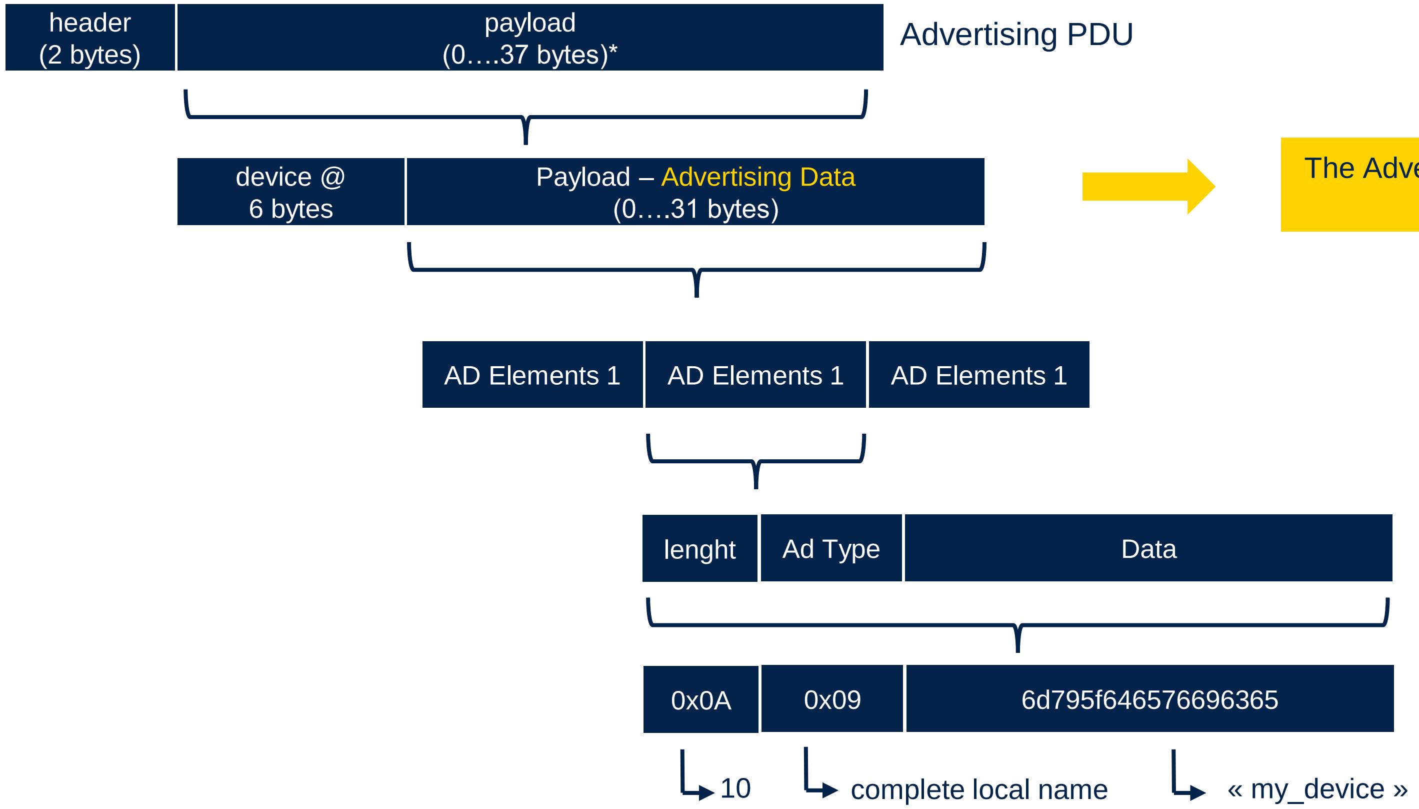
Configure the below parameters :

Include AD_TYPE_TX_POWER_LEVEL element	No
Include AD_TYPE_COMPLETE_LOCAL_NAME element	Yes
AD_TYPE_COMPLETE_LOCAL_NAME_LENGTH	8
AD_TYPE_COMPLETE_LOCAL_NAME	p2pS_01
Include AD_TYPE_SHORTENED_LOCAL_NAME element	No
Include AD_TYPE_APPEARANCE element	No
Include AD_TYPE_ADVERTISING_INTERVAL element	No
Include AD_TYPE_LE_ROLE element	No
Include AD_TYPE_16_BIT_SERV_UUID_CMPLT_LIST element	No
Include AD_TYPE_128_BIT_SERV_UUID_CMPLT_LIST element	No
Include AD_TYPE_SLAVE_CONN_INTERVAL element	No
Include AD_TYPE_URI element	No
Include AD_TYPE_MANUFACTURER_SPECIFIC_DATA element	Yes
AD_TYPE_MANUFACTURER_SPECIFIC_DATA_LENGTH	15
Company identifier	30,00
Number of user defined data item(s)	12

As a server, our application will have to advertise waiting for connection request from a client.

Define here your “custom” local name part of advertising frame.

Advertising Elements
Advertising PDU



The Advertising Data consists of one or more Advertising Data elements
AD Element/Type are listed at Bluetooth SIG website

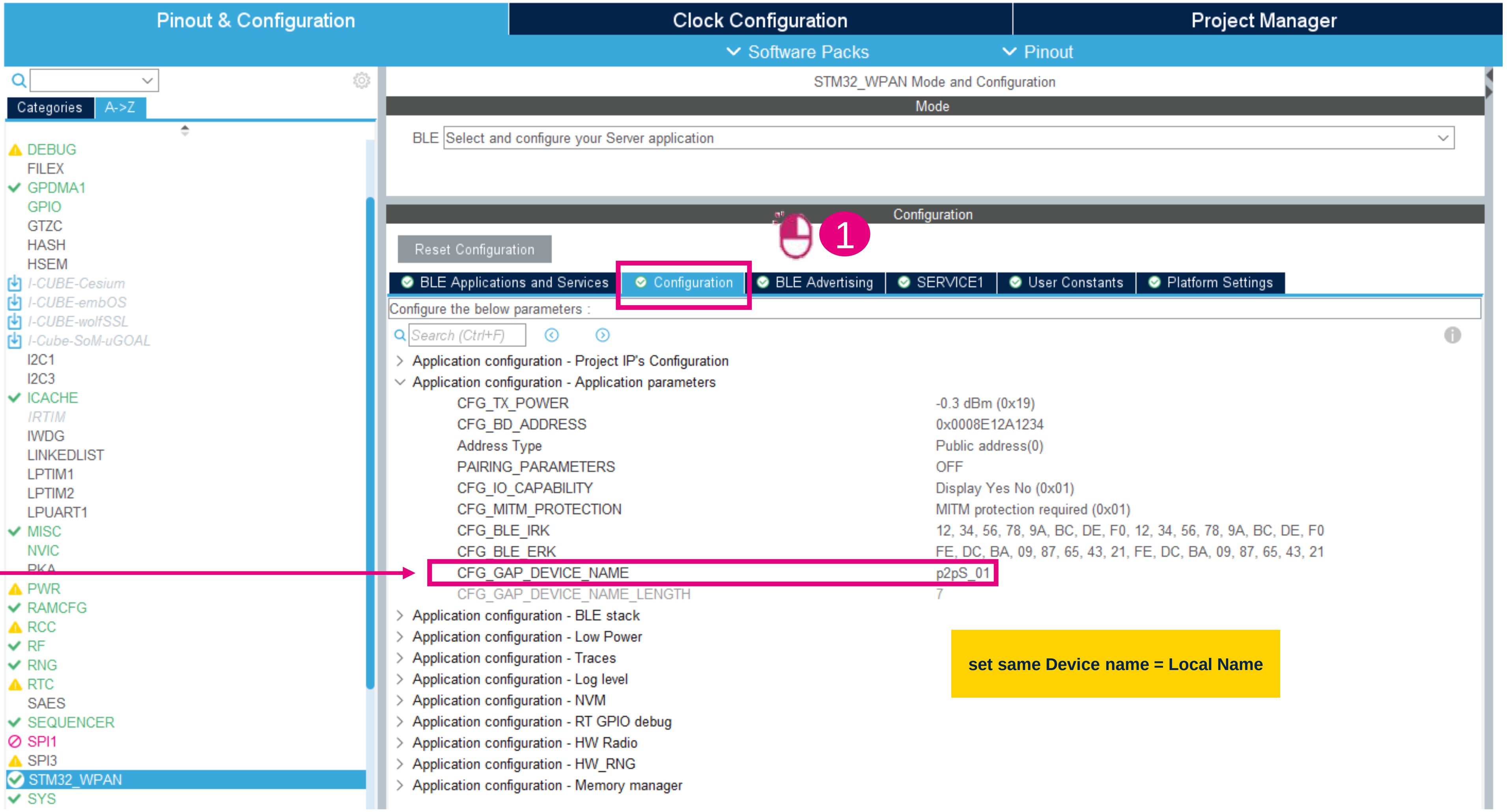
Most commonly used AD elements :
- 0x01 = Flags (**mandatory for connectable device**)
- 0x09 = Complete Local Name
- 0xFF = Manufacturer Data

added by
aci_gap_set_discoverable
command

You can push for what you want over the air ! All data need to be prefix using dedicated Ad Type

* up to 37 bytes in legacy advertising , up to 1650 in case of advertising extension (packet of 250 chained)

Customize Device Name



DEBUG
FILEX
GPDMA1
GPIO
GTZC
HASH
HSEM
I-CUBE-Cesium
I-CUBE-embOS
I-CUBE-wolfSSL
I-Cube-SoM-uGOAL
I2C1
I2C3
ICACHE
IRTIM
IWDG
LINKEDLIST
LPTIM1
LPTIM2
LPUART1
MISC
NVIC
PKA
PWR
RAMCFG
RCC
RF
RNG
RTC
SAES
SEQUENCER
SPI1
SPI3
STM32_WPAN
SYS

STM32_WPAN Mode and Configuration

Mode

BLE: Select and configure your Server application

Configuration

Reset Configuration

BLE Applications and Services | Configuration | BLE Advertising | SERVICE1 | User Constants | Platform Settings

Configure the below parameters :

Search (Ctrl+F)

- Application configuration - Project IP's Configuration
- Application configuration - Application parameters

CFG_TX_POWER	-0.3 dBm (0x19)
CFG_BD_ADDRESS	0x0008E12A1234
Address Type	Public address(0)
PAIRING_PARAMETERS	OFF
CFG_IO_CAPABILITY	Display Yes No (0x01)
CFG_MITM_PROTECTION	MITM protection required (0x01)
CFG_BLE_IRK	12, 34, 56, 78, 9A, BC, DE, F0, 12, 34, 56, 78, 9A, BC, DE, F0
CFG_BLE_ERK	FE, DC, BA, 09, 87, 65, 43, 21, FE, DC, BA, 09, 87, 65, 43, 21
CFG_GAP_DEVICE_NAME	p2pS_01
CFG_GAP_DEVICE_NAME_LENGTH	7

- Application configuration - BLE stack
- Application configuration - Low Power
- Application configuration - Traces
- Application configuration - Log level
- Application configuration - NVM
- Application configuration - RT GPIO debug
- Application configuration - HW Radio
- Application configuration - HW_RNG
- Application configuration - Memory manager

1

2

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)

Configuration completed
What’s next : code generation ?

HW configuration ✓

BLE parameters configuration ✓

In order to generate the required skeleton code to move to discoverable mode profile (service & characteristic) need to be created

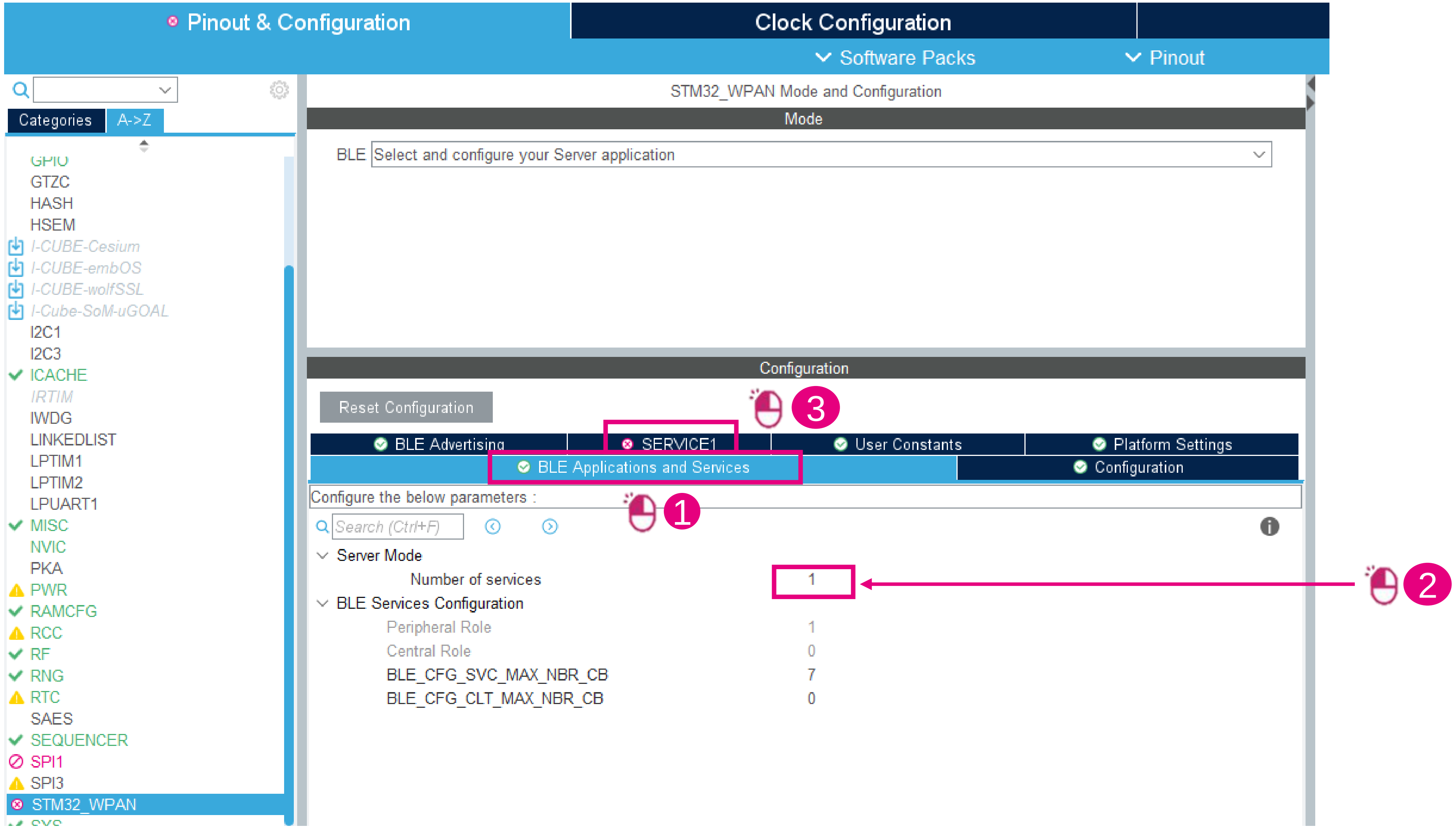
Profile creation

service and characteristics demystification will be done in **Hands-On #2**

Code generation

Application code

Profile Creation Service


Pinout & Configuration
Clock Configuration
Software Packs
Pinout

Categories | A->Z

GPIO
GTZC
HASH
HSEM
I-CUBE-Cesium
I-CUBE-embOS
I-CUBE-wolfSSL
I-Cube-SoM-uGOAL
I2C1
I2C3
ICACHE
IRTIM
IWDG
LINKEDLIST
LPTIM1
LPTIM2
LPUART1
MISC
NVIC
PKA
PWR
RAMCFG
RCC
RF
RNG
RTC
SAES
SEQUENCER
SPI1
SPI3
STM32_WPAN
SYS

STM32_WPAN Mode and Configuration

Mode

BLE: Select and configure your Server application

Configuration

Reset Configuration

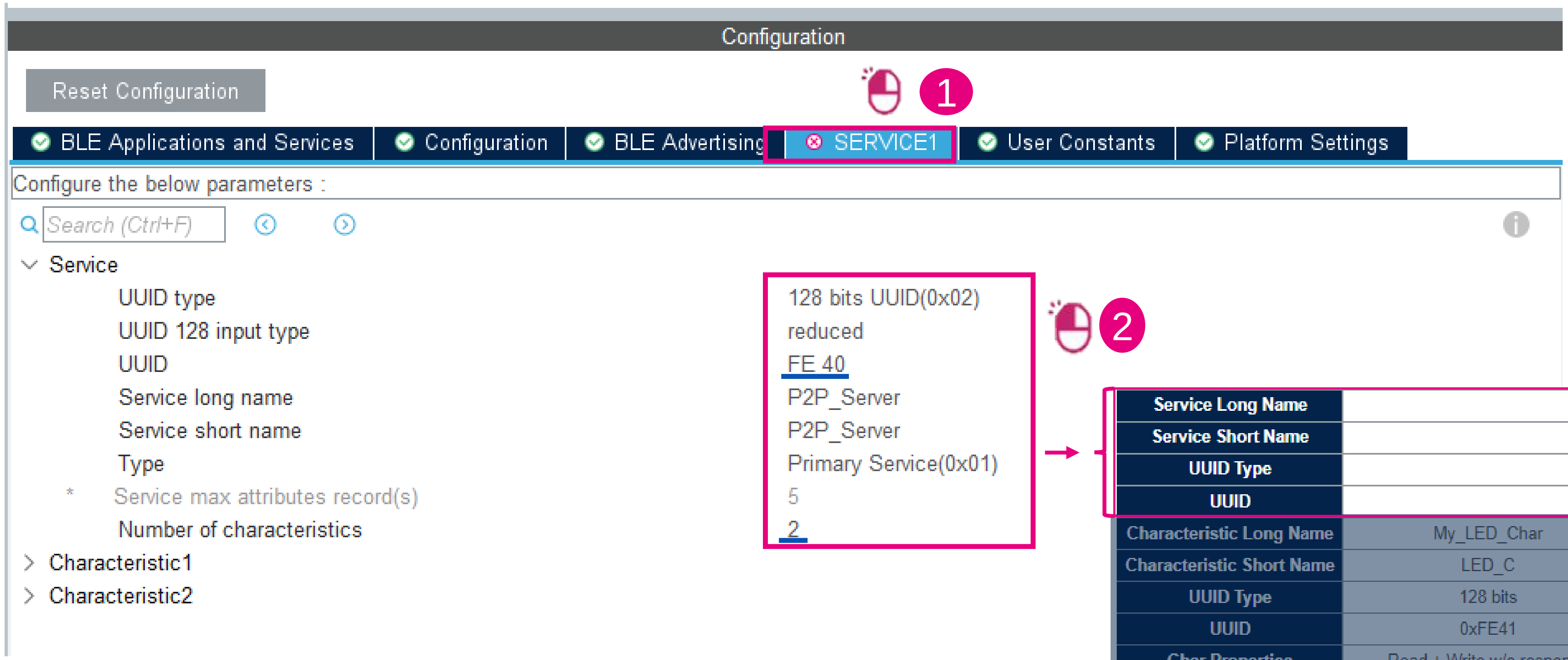
BLE Advertising | SERVICE1 | User Constants | Platform Settings
BLE Applications and Services | Configuration

Configure the below parameters :

Search (Ctrl+F)

Server Mode	
Number of services	1
BLE Services Configuration	
Peripheral Role	1
Central Role	0
BLE_CFG_SVC_MAX_NBR_CB	7
BLE_CFG_CLT_MAX_NBR_CB	0

Profile Creation
Configure my P2P Service



Service

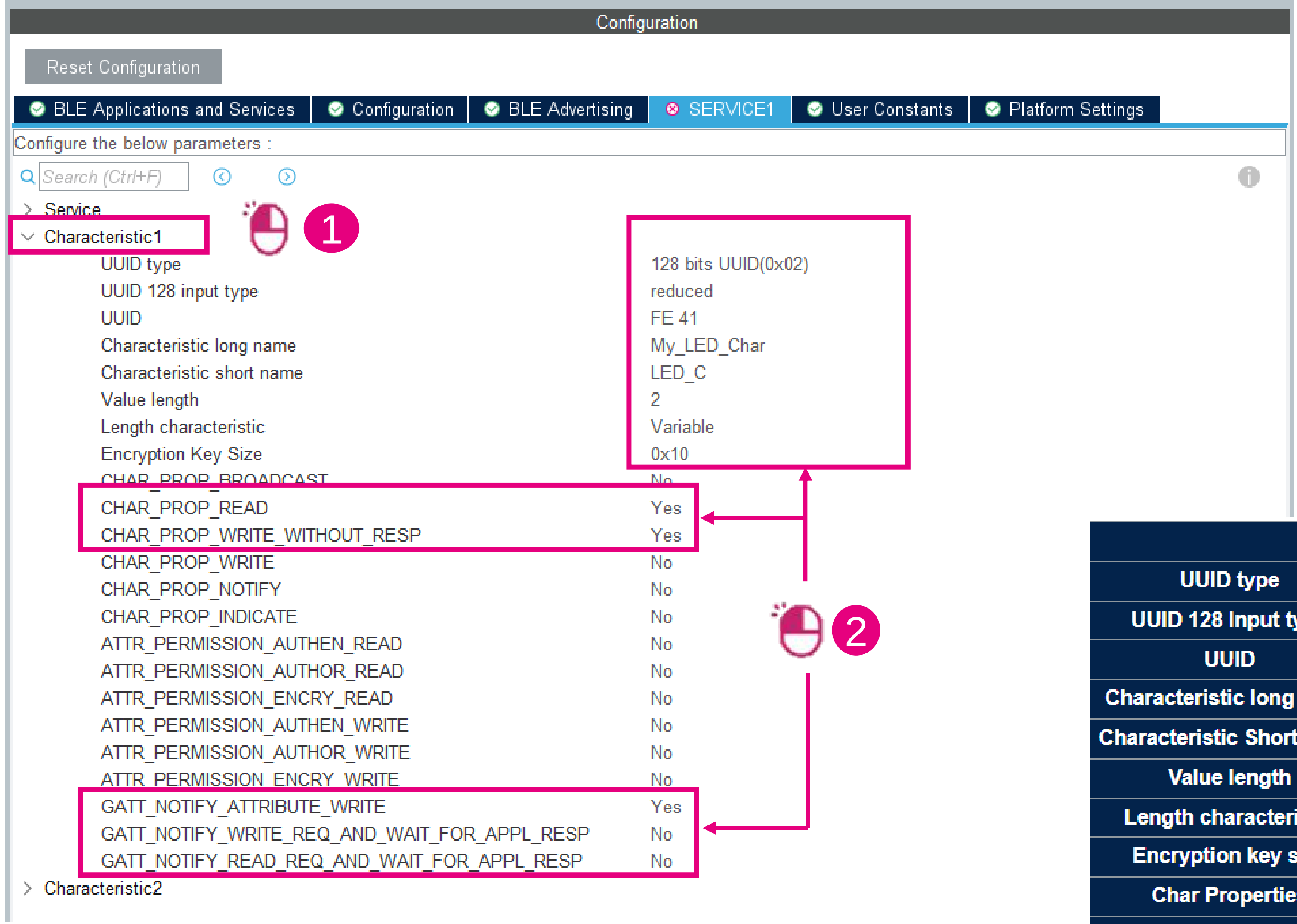
UUID type	128 bits UUID(0x02)
UUID 128 input type	reduced
UUID	FE 40
Service long name	P2P_Server
Service short name	P2P_Server
Type	Primary Service(0x01)
* Service max attributes record(s)	5
Number of characteristics	2

Characteristic1
Characteristic2

Service Long Name	My_P2P_Server	
Service Short Name	My_P2P	
UUID Type	128 bits	
UUID	0xFE40	
Characteristic Long Name	My_LED_Char	My_Switch_Char
Characteristic Short Name	LED_C	SWITCH_C
UUID Type	128 bits	128 bits
UUID	0xFE41	0xFE42
Char Properties	Read + Write w/o response	Notify
Char Permissions	None	None
Char GATT Events	GATT_NOTIFY_ATTRIBUTE_WRITE	GATT_NOTIFY_ATTRIBUTE_WRITE

life.augmented

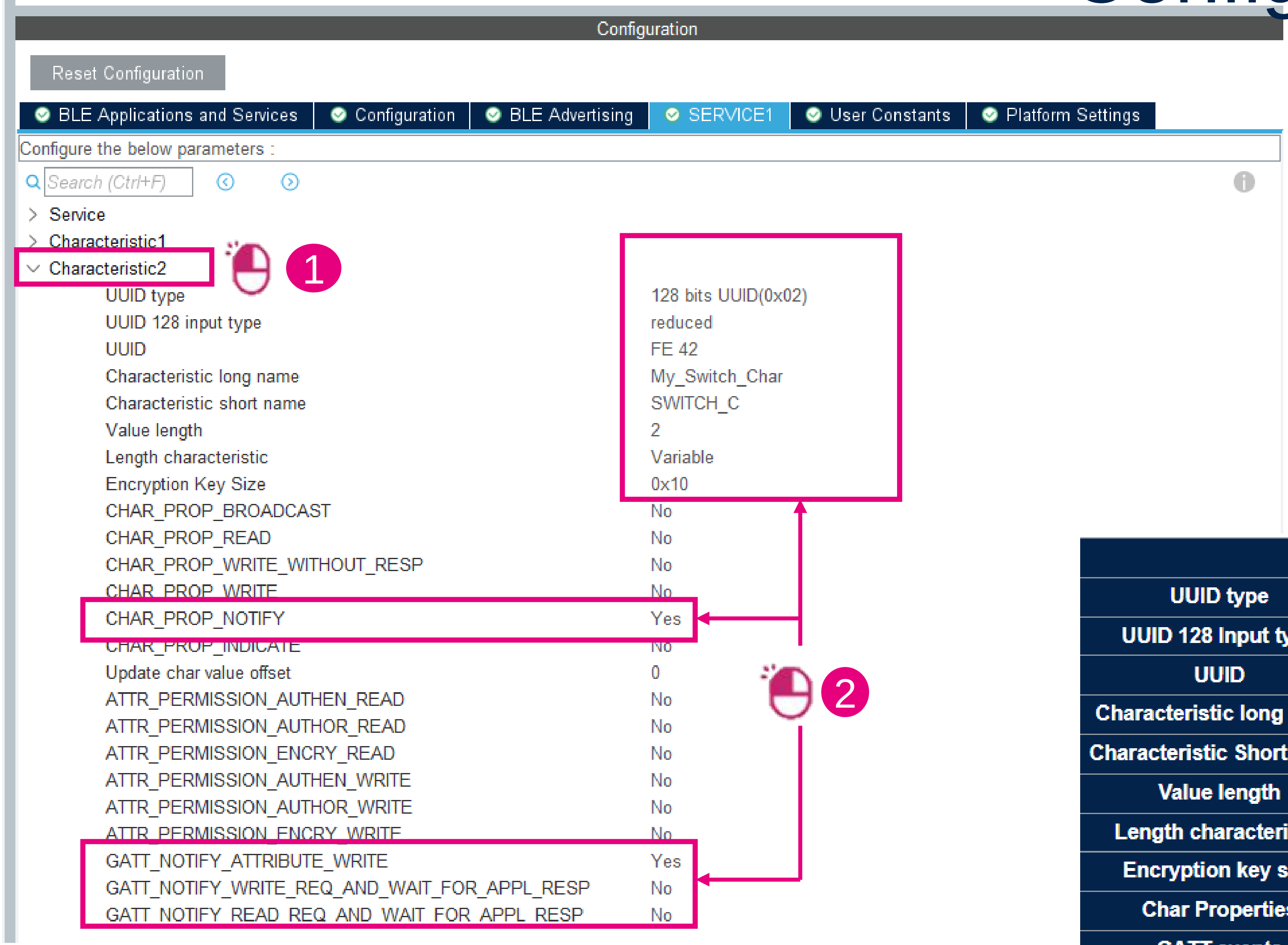
Profile Creation
Configure 1st Characteristic



UUID type	128 bits UUID(0x02)
UUID 128 input type	reduced
UUID	FE 41
Characteristic long name	My_LED_Char
Characteristic short name	LED_C
Value length	2
Length characteristic	Variable
Encryption Key Size	0x10
CHAR_PROP_BROADCAST	No
CHAR_PROP_READ	Yes
CHAR_PROP_WRITE_WITHOUT_RESP	Yes
CHAR_PROP_WRITE	No
CHAR_PROP_NOTIFY	No
CHAR_PROP_INDICATE	No
ATTR_PERMISSION_AUTHEN_READ	No
ATTR_PERMISSION_AUTHOR_READ	No
ATTR_PERMISSION_ENCRY_READ	No
ATTR_PERMISSION_AUTHEN_WRITE	No
ATTR_PERMISSION_AUTHOR_WRITE	No
ATTR_PERMISSION_ENCRY_WRITE	No
GATT_NOTIFY_ATTRIBUTE_WRITE	Yes
GATT_NOTIFY_WRITE_REQ_AND_WAIT_FOR_APPL_RESP	No
GATT_NOTIFY_READ_REQ_AND_WAIT_FOR_APPL_RESP	No

	Characteristic 1		Characteristic 2
UUID type	128 bits UUID (0x02)		128 bits UUID (0x02)
UUID 128 Input type	Reduced		Reduced
UUID	FE 41		FE 42
Characteristic long name	My_LED_Char		My_Switch_Char
Characteristic Short Name	LED_C		SWITCH_C
Value length	2		2
Length characteristic	Variable		Variable
Encryption key size	0x10		0x10
Char Properties	READ	WRITE_WITHOUT_RESP	NOTIFY
GATT events	GATT_NOTIFY_ATTRIBUTE_WRITE		GATT_NOTIFY_ATTRIBUTE_WRITE

Profile Creation
Configure 2nd Characteristic



Configure the below parameters :

Service
Characteristic1
Characteristic2

UUID type	128 bits UUID(0x02)
UUID 128 input type	reduced
UUID	FE 42
Characteristic long name	My_Switch_Char
Characteristic short name	SWITCH_C
Value length	2
Length characteristic	Variable
Encryption Key Size	0x10
CHAR_PROP_BROADCAST	No
CHAR_PROP_READ	No
CHAR_PROP_WRITE_WITHOUT_RESP	No
CHAR_PROP_WRITE	No
CHAR_PROP_NOTIFY	Yes
CHAR_PROP_INDICATE	No
Update char value offset	0
ATTR_PERMISSION_AUTHEN_READ	No
ATTR_PERMISSION_AUTHOR_READ	No
ATTR_PERMISSION_ENCRY_READ	No
ATTR_PERMISSION_AUTHEN_WRITE	No
ATTR_PERMISSION_AUTHOR_WRITE	No
ATTR_PERMISSION_ENCRY_WRITE	No
GATT_NOTIFY_ATTRIBUTE_WRITE	Yes
GATT_NOTIFY_WRITE_REQ_AND_WAIT_FOR_APPL_RESP	No
GATT_NOTIFY_READ_REQ_AND_WAIT_FOR_APPL_RESP	No

	Characteristic 1		Characteristic 2
UUID type	128 bits UUID (0x02)		128 bits UUID (0x02)
UUID 128 Input type	Reduced		Reduced
UUID	FE 41		FE 42
Characteristic long name	My_LED_Char		My_Switch_Char
Characteristic Short Name	LED_C		SWITCH_C
Value length	2		2
Length characteristic	Variable		Variable
Encryption key size	0x10		0x10
Char Properties	READ	WRITE_WITHOUT_RESP	NOTIFY
GATT events	GATT_NOTIFY_ATTRIBUTE_WRITE		GATT_NOTIFY_ATTRIBUTE_WRITE

Configuration completed
What's next - Yes code generation

HW configuration ✓

BLE parameters configuration ✓

In order to generate the required skeleton code to move to discoverable mode profile (service & characteristic) need to be created

Profile creation ✓

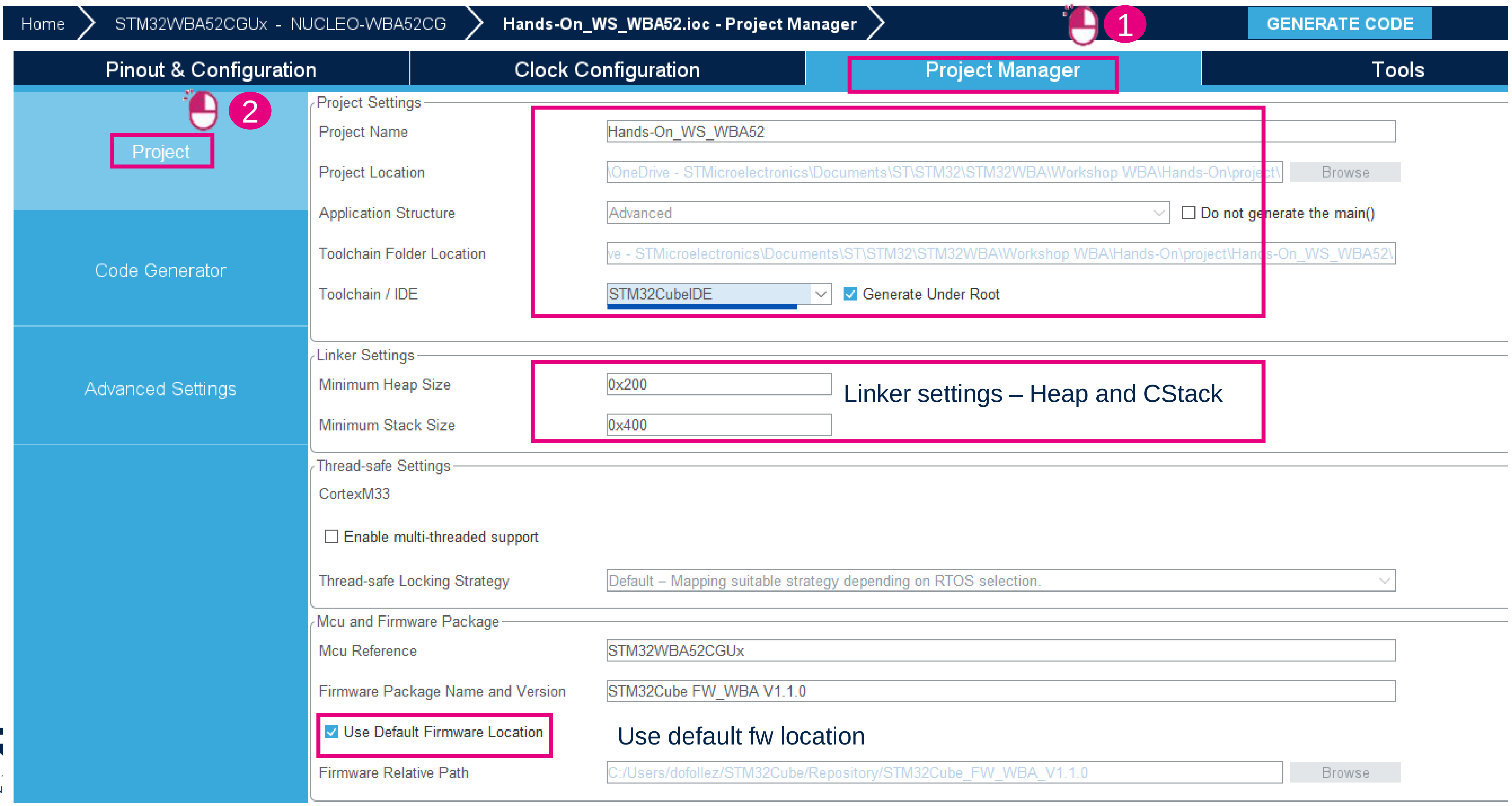
Code generation

Application code

Step 3 : Code generation and user application code

life.augmented

Project configuration



Project Settings

Project Name: Hands-On_WS_WBA52

Project Location: \OneDrive - STMicroelectronics\Documents\ST\STM32\STM32WBA\Workshop WBA\Hands-On\project\

Application Structure: Advanced ☐ Do not generate the main()

Toolchain Folder Location: ve - STMicroelectronics\Documents\ST\STM32\STM32WBA\Workshop WBA\Hands-On\project\Hands-On_WS_WBA52\

Toolchain / IDE: STM32CubeIDE ☑ Generate Under Root

Linker Settings

Minimum Heap Size: 0x200

Minimum Stack Size: 0x400

Linker settings – Heap and CStack

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

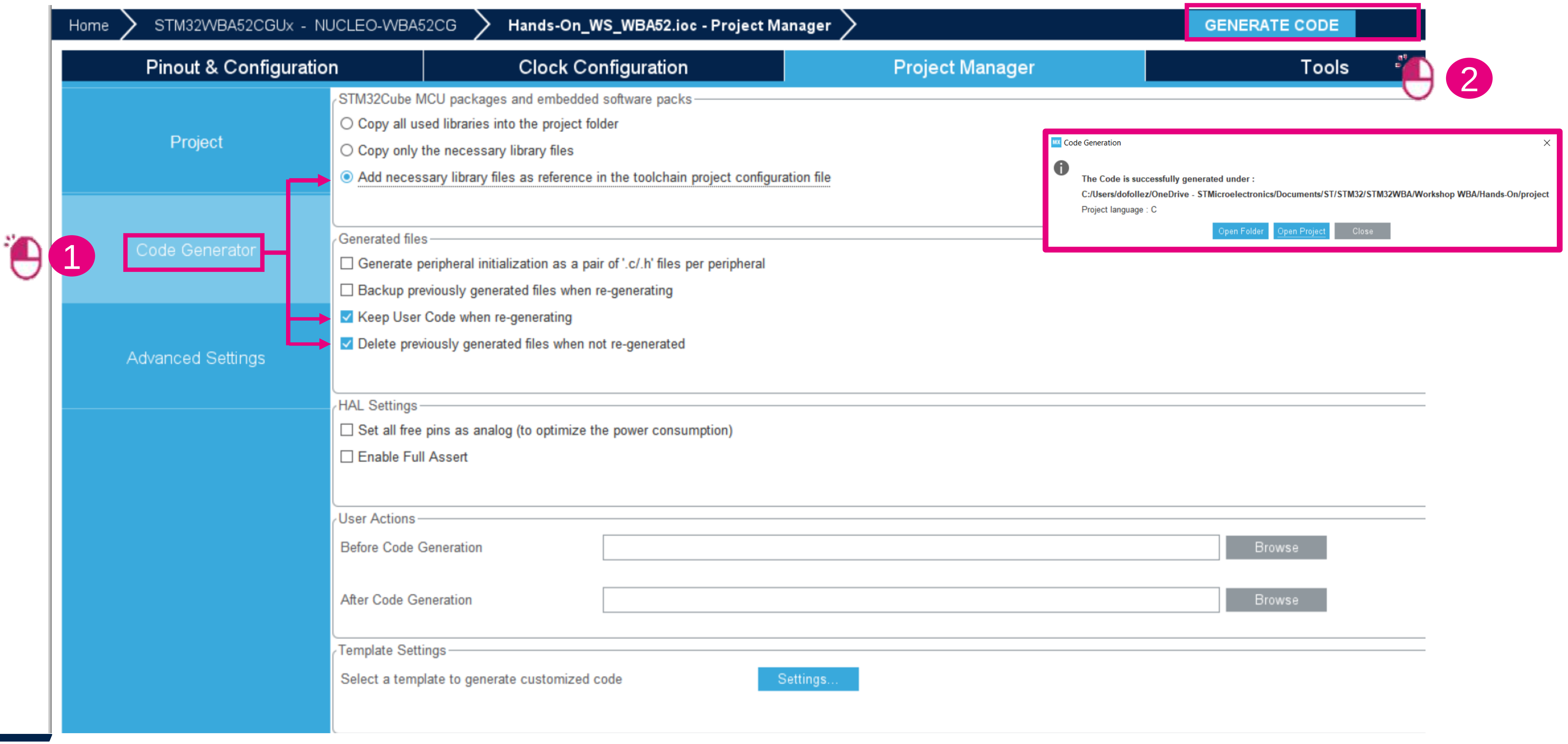
Firmware Package Name and Version: STM32Cube FW_WBA V1.1.0

☑ Use Default Firmware Location

Use default fw location

Firmware Relative Path: C:/Users/dofollez/STM32Cube/Repository/STM32Cube_FW_WBA_V1.1.0

Project configuration



STM32Cube MCU packages and embedded software packs

- ○ Copy all used libraries into the project folder
- ○ Copy only the necessary library files
- ● Add necessary library files as reference in the toolchain project configuration file

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

User Actions

Before Code Generation

After Code Generation

Template Settings

Select a template to generate customized code

Code Generation

The Code is successfully generated under :

C:/Users/dofollez/OneDrive - STMicroelectronics/Documents/ST/STM32/STM32WBA/Workshop WBA/Hands-On/project

Project language : C

Configuration completed
What's next - Yes code generation

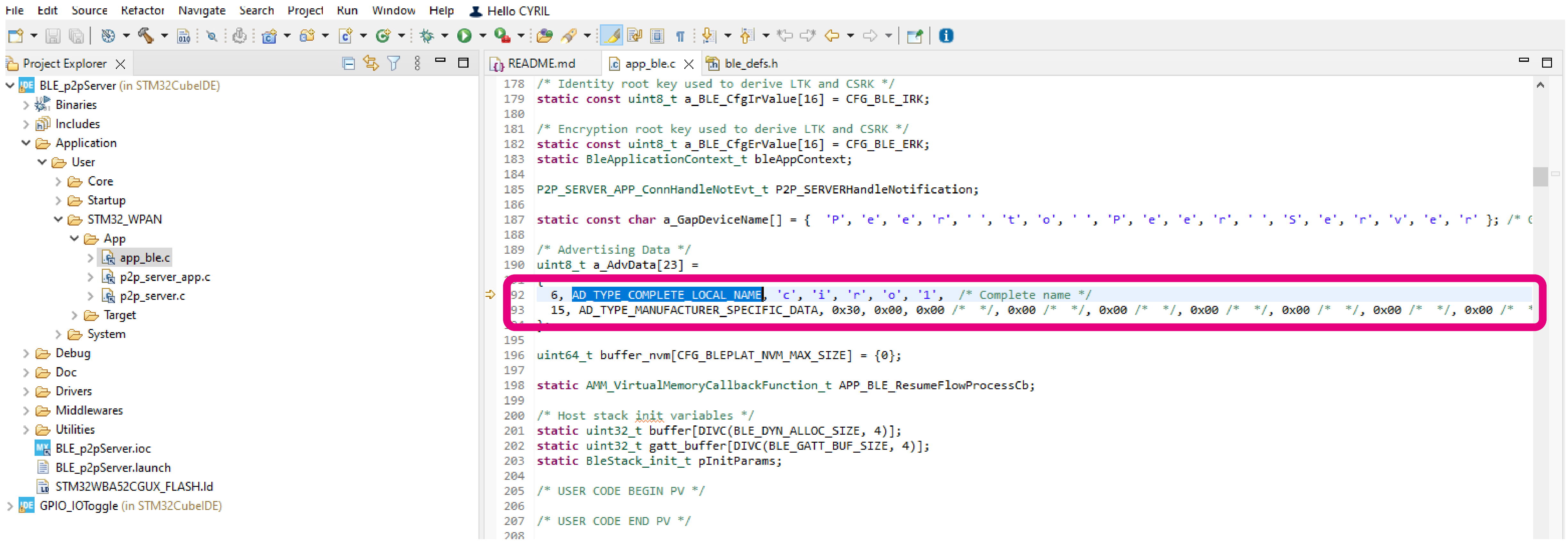
- **HW configuration** ✓
- **BLE parameters configuration** ✓

In order to generate the required skeleton code to move to discoverable mode profile (service & characteristic) need to be created

- **Profile creation** ✓
- **Code generation** ✓
- **Application code**

life.augmented

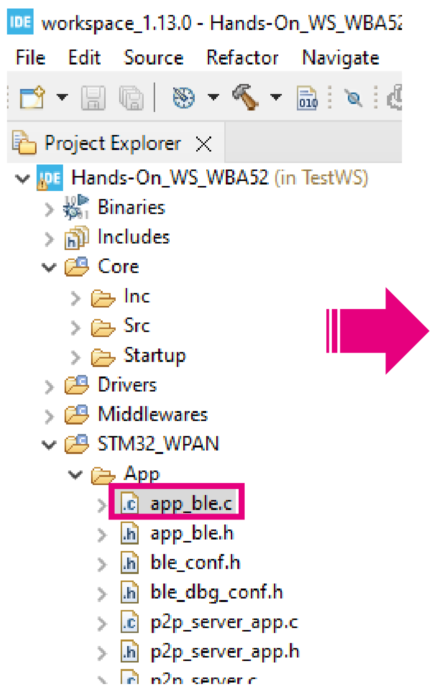
Here are our ADV data



```c
/* Identity root key used to derive LTK and CSRK */
static const uint8_t a_BLE_CfgIrValue[16] = CFG_BLE_IRK;

/* Encryption root key used to derive LTK and CSRK */
static const uint8_t a_BLE_CfgErValue[16] = CFG_BLE_ERK;
static BleApplicationContext_t bleAppContext;

P2P_SERVER_APP_ConnHandleNotEvt_t P2P_SERVERHandleNotification;

static const char a_GapDeviceName[] = {  'P', 'e', 'e', 'r', ' ', 't', 'o', ' ', 'P', 'e', 'e', 'r', ' ', 'S', 'e', 'r', 'v', 'e', 'r' }; /*

/* Advertising Data */
uint8_t a_AdvData[23] =
{
  6, AD_TYPE_COMPLETE_LOCAL_NAME, 'c', 'i', 'r', 'o', '1',  /* Complete name */
  15, AD_TYPE_MANUFACTURER_SPECIFIC_DATA, 0x30, 0x00, 0x00 /*  */, 0x00 /*  */, 0x00 /*  */, 0x00 /*  */, 0x00 /*  */, 0x00 /*  */, 0x00 /*  *
};

uint64_t buffer_nvm[CFG_BLEPLAT_NVM_MAX_SIZE] = {0};

static AMM_VirtualMemoryCallbackFunction_t APP_BLE_ResumeFlowProcessCb;

/* Host stack init variables */
static uint32_t buffer[DIVC(BLE_DYN_ALLOC_SIZE, 4)];
static uint32_t gatt_buffer[DIVC(BLE_GATT_BUF_SIZE, 4)];
static BleStack_init_t pInitParams;

/* USER CODE BEGIN PV */

/* USER CODE END PV */
```

life.augmented

Open Project
Add application code to move to discoverable



#1
: Set device discoverable at init :
In app_ble.c > function APP_BLE_Init()

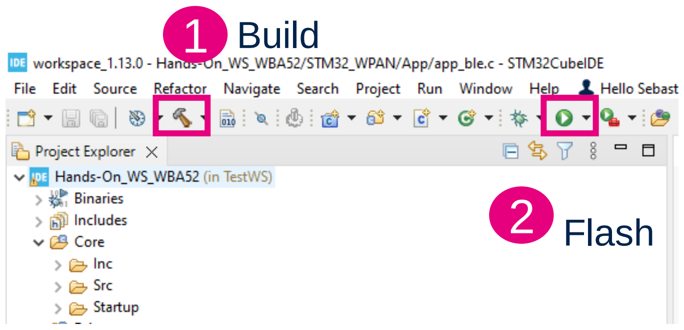
(ADV_MIN+ADV_MAX)/2

```
/* USER CODE BEGIN APP_BLE_Init_2 */
APP_BLE_Procedure_Gap_Peripheral(PROC_GAP_PERIPH_ADVERTISE_START_FAST);
 /* USER CODE END APP_BLE_Init_2 */
```

#2 : Set device discoverable at disconnection :
In app_ble.c > SVCCTL_App_Notification -
HCI_DISCONNECTION_COMPLETE_EVT_CODE

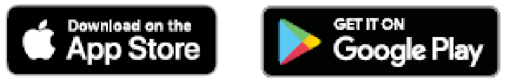
```
 /* USER CODE BEGIN EVT_DISCONN_COMPLETE */
APP_BLE_Procedure_Gap_Peripheral(PROC_GAP_PERIPH_ADVERTISE_START_FAST);
 /* USER CODE BEGIN EVT_DISCONN_COMPLETE */
```

At disconnection, stack is not moving back to advertising, this is an application decision


1 Build
2 Flash
workspace_1.13.0 - Hands-On_WS_WBA52/STM32_WPAN/App/app_ble.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help Hello Sebast
Project Explorer
Hands-On_WS_WBA52 (in TestWS)
Binaries
Includes
Core
Inc
Src
Startup

Open Project → Add application code to move to discoverable → Build& Flash

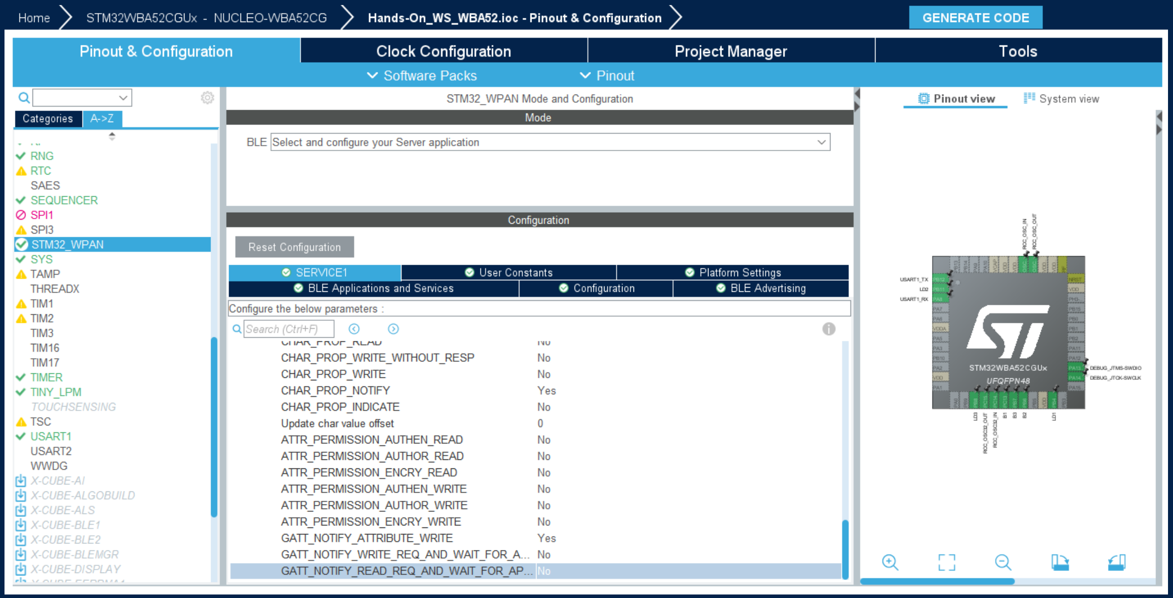
Open your App and Connect



Device should be visible and connectable

Step 4 : Add debug capabilities

Move back to CubeMx



RNG
RTC
SAES
SEQUENCER
SPI1
SPI3
STM32_WPAN
SYS
TAMP
THREADX
TIM1
TIM2
TIM3
TIM16
TIM17
TIMER
TINY_LPM
TOUCHSENSING
TSC
USART1
USART2
WWDG
X-CUBE-AI
X-CUBE-ALGOBUILD
X-CUBE-ALS
X-CUBE-BLE1
X-CUBE-BLE2
X-CUBE-BLEMGR
X-CUBE-DISPLAY

STM32_WPAN Mode and Configuration

Mode

BLE: Select and configure your Server application

Configuration

Reset Configuration

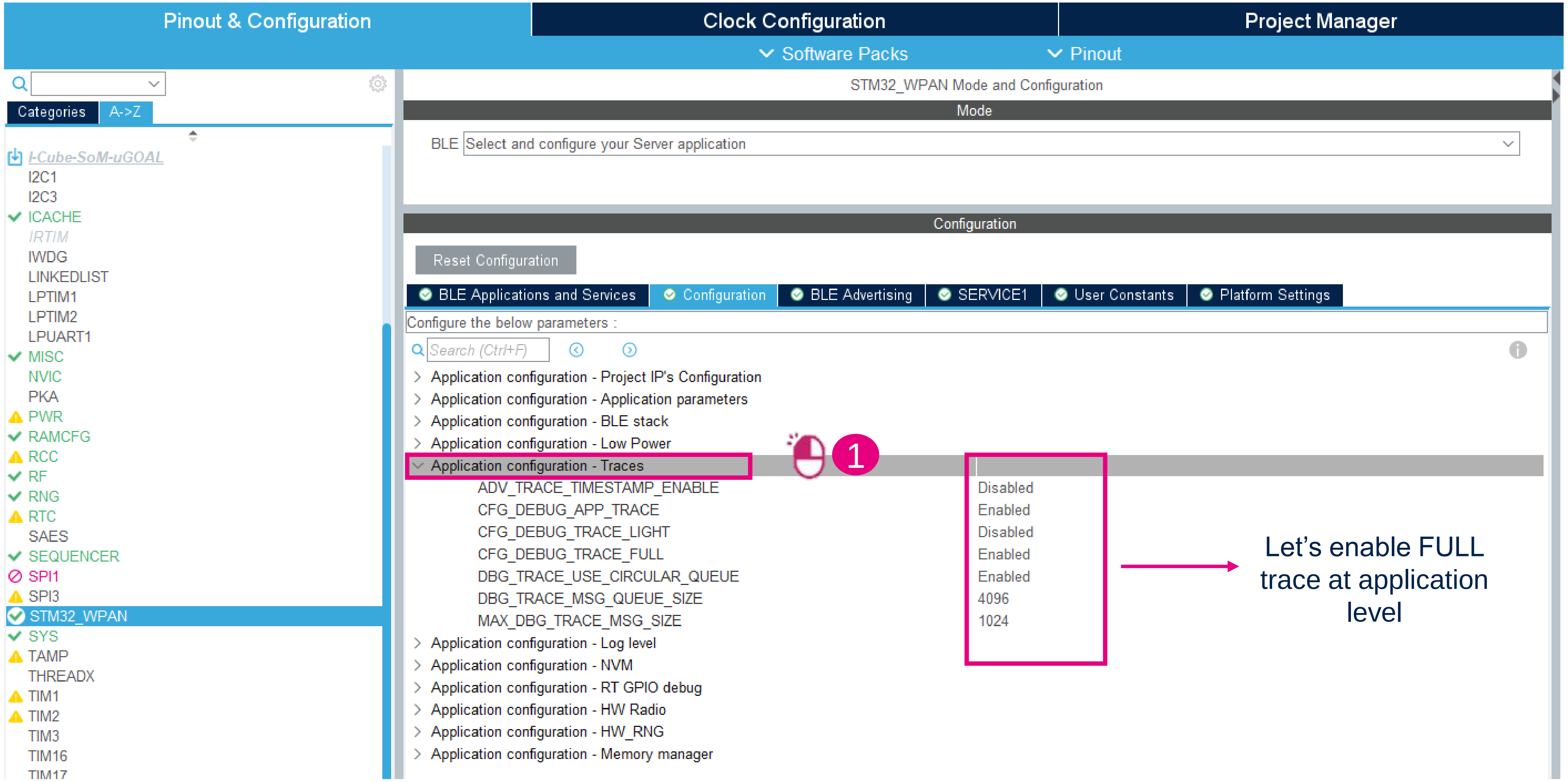
SERVICE1 | User Constants | Platform Settings | BLE Applications and Services | Configuration | BLE Advertising

Configure the below parameters :

CHAR_PROP_READ	No
CHAR_PROP_WRITE_WITHOUT_RESP	No
CHAR_PROP_WRITE	No
CHAR_PROP_NOTIFY	Yes
CHAR_PROP_INDICATE	No
Update char value offset	0
ATTR_PERMISSION_AUTHEN_READ	No
ATTR_PERMISSION_AUTHOR_READ	No
ATTR_PERMISSION_ENCRY_READ	No
ATTR_PERMISSION_AUTHEN_WRITE	No
ATTR_PERMISSION_AUTHOR_WRITE	No
ATTR_PERMISSION_ENCRY_WRITE	No
GATT_NOTIFY_ATTRIBUTE_WRITE	Yes
GATT_NOTIFY_WRITE_REQ_AND_WAIT_FOR_A...	No
GATT_NOTIFY_READ_REQ_AND_WAIT_FOR_AP...	No

life.augmented

Application configuration
Traces & logs



Pinout & Configuration | Clock Configuration | Project Manager

Software Packs | Pinout

Categories | A->Z

- I-Cube-SoM-uGOAL
- I2C1
- I2C3
- ✓ ICACHE
- IRTIM
- IWDG
- LINKEDLIST
- LPTIM1
- LPTIM2
- LPUART1
- ✓ MISC
- NVIC
- PKA
- ⚠ PWR
- ✓ RAMCFG
- ⚠ RCC
- ✓ RF
- ✓ RNG
- ⚠ RTC
- SAES
- ✓ SEQUENCER
- ⊘ SPI1
- ⚠ SPI3
- ✓ STM32_WPAN
- ✓ SYS
- ⚠ TAMP
- THREADX
- ⚠ TIM1
- ⚠ TIM2
- TIM3
- TIM16
- TIM17

STM32_WPAN Mode and Configuration

Mode

BLE: Select and configure your Server application

Configuration

Reset Configuration

BLE Applications and Services | Configuration | BLE Advertising | SERVICE1 | User Constants | Platform Settings

Configure the below parameters :

Search (Ctrl+F)

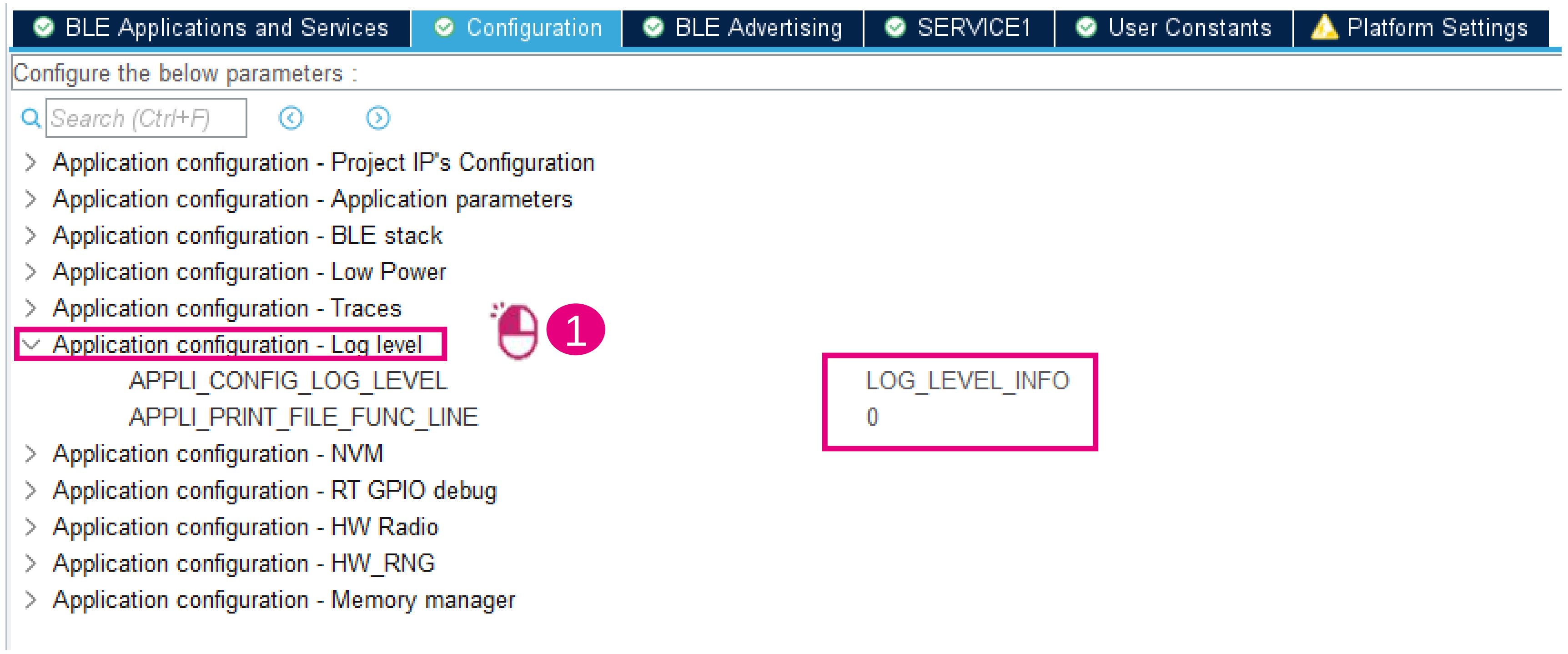
- Application configuration - Project IP's Configuration
- Application configuration - Application parameters
- Application configuration - BLE stack
- Application configuration - Low Power
- Application configuration - Traces

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

- Application configuration - Log level
- Application configuration - NVM
- Application configuration - RT GPIO debug
- Application configuration - HW Radio
- Application configuration - HW_RNG
- Application configuration - Memory manager

Let's enable FULL trace at application level

Application configuration
Trace & Logs: configure log level



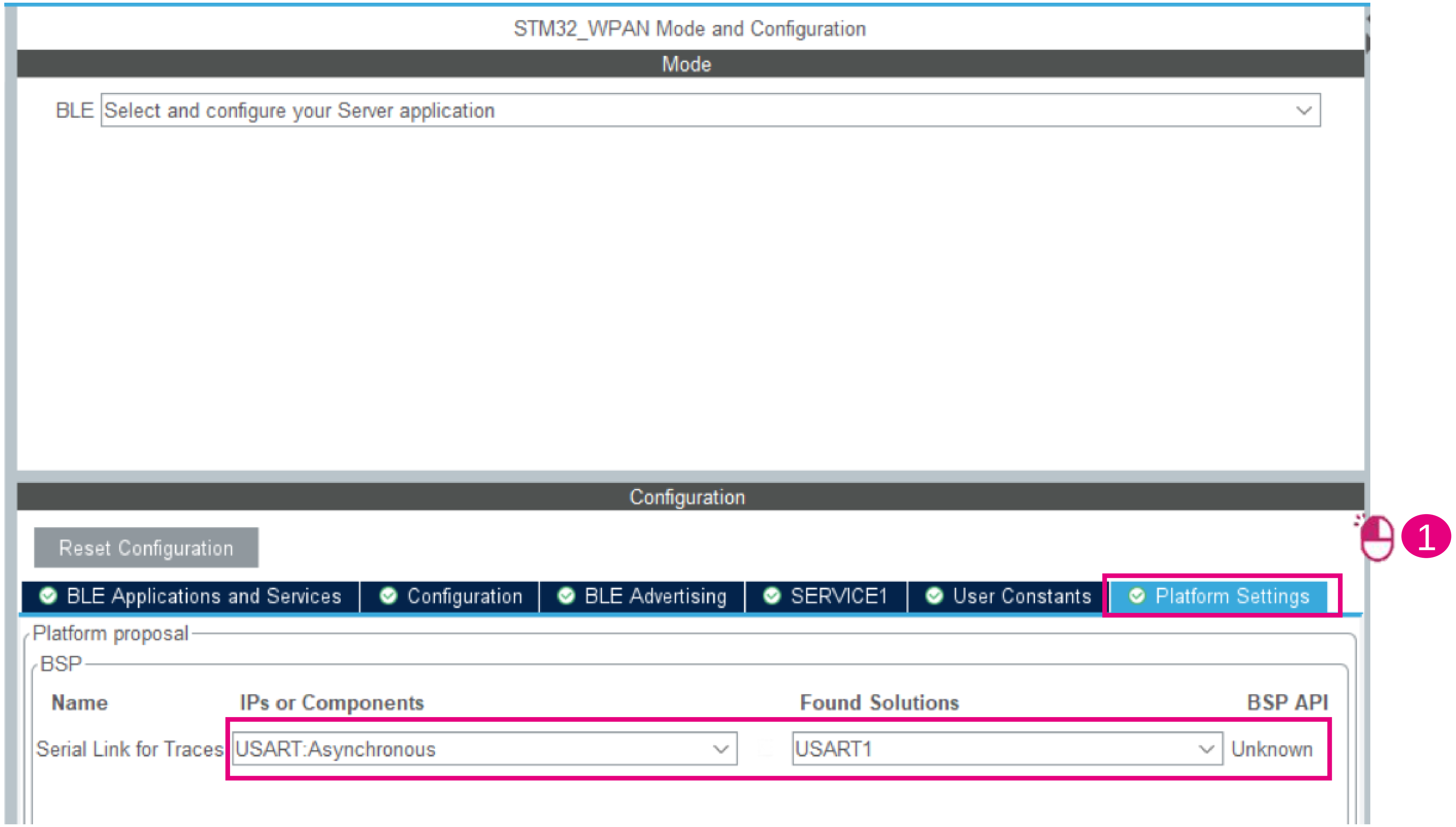
Configure the below parameters :

- Application configuration - Project IP's Configuration
- Application configuration - Application parameters
- Application configuration - BLE stack
- Application configuration - Low Power
- Application configuration - Traces
- Application configuration - Log level

APPLI_CONFIG_LOG_LEVEL	LOG_LEVEL_INFO
APPLI_PRINT_FILE_FUNC_LINE	0

- Application configuration - NVM
- Application configuration - RT GPIO debug
- Application configuration - HW Radio
- Application configuration - HW_RNG
- Application configuration - Memory manager

Platform Settings
Trace & Logs: BSP settings


STM32_WPAN Mode and Configuration

Mode

BLE: Select and configure your Server application

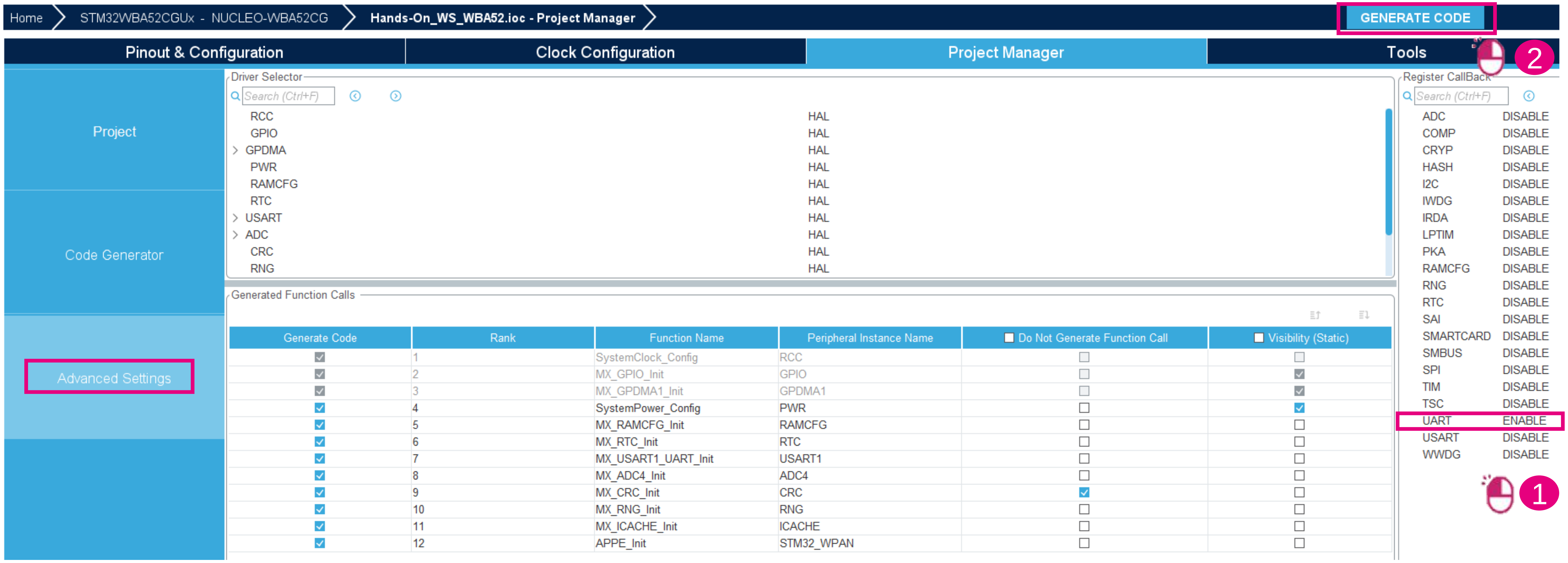
Configuration

Platform proposal

BSP

Name	IPs or Components	Found Solutions	BSP API
Serial Link for Traces	USART:Asynchronous	USART1	Unknown

Project configuration Advanced settings



Home > STM32WBA52CGUx - NUCLEO-WBA52CG > Hands-On_WS_WBA52.ioc - Project Manager

GENERATE CODE

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Project

Code Generator

Advanced Settings

Driver Selector

RCC	HAL
GPIO	HAL
GPDMA	HAL
PWR	HAL
RAMCFG	HAL
RTC	HAL
USART	HAL
ADC	HAL
CRC	HAL
RNG	HAL

Register CallBack

ADC	DISABLE
COMP	DISABLE
CRYP	DISABLE
HASH	DISABLE
I2C	DISABLE
IWDG	DISABLE
IRDA	DISABLE
LPTIM	DISABLE
PKA	DISABLE
RAMCFG	DISABLE
RNG	DISABLE
RTC	DISABLE
SAI	DISABLE
SMARTCARD	DISABLE
SMBUS	DISABLE
SPI	DISABLE
TIM	DISABLE
TSC	DISABLE
UART	ENABLE
USART	DISABLE
WWDG	DISABLE

Generated Function Calls

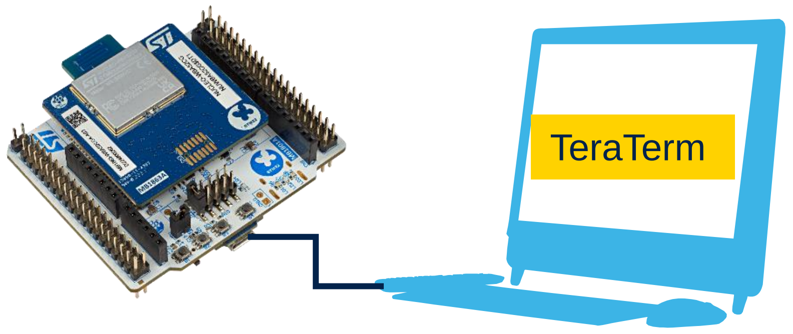
Generate Code	Rank	Function Name	Peripheral Instance Name	Do Not Generate Function Call	Visibility (Static)
✓	1	SystemClock_Config	RCC	☐	☐
✓	2	MX_GPIO_Init	GPIO	☐	✓
✓	3	MX_GPDMA1_Init	GPDMA1	☐	✓
✓	4	SystemPower_Config	PWR	☐	✓
✓	5	MX_RAMCFG_Init	RAMCFG	☐	☐
✓	6	MX_RTC_Init	RTC	☐	☐
✓	7	MX_USART1_UART_Init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

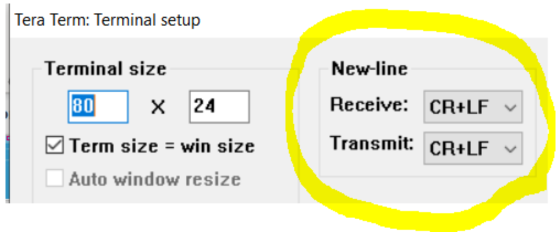
Regenerate Code

Open back existing project – refresh sources – build and flash

life.augmented

Open your App and Connect




Tera Term: Terminal setup

Terminal size: 80 X 24

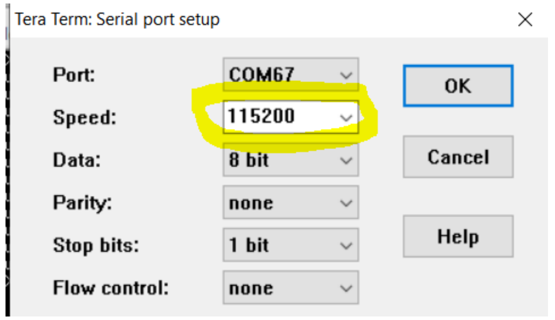
[✓] Term size = win size

[ ] Auto window resize

New-line

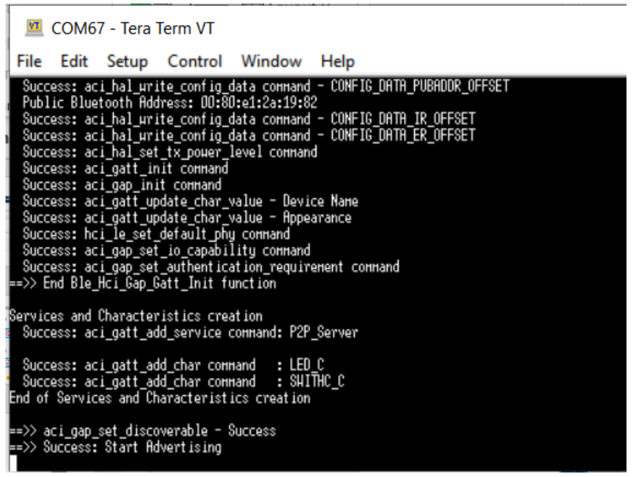
Receive: CR+LF

Transmit: CR+LF


Tera Term: Serial port setup

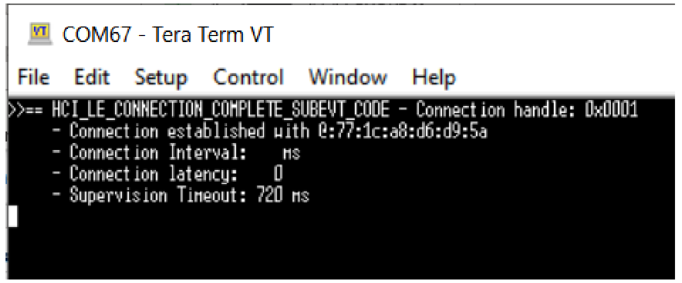
Port:	COM67
Speed:	115200
Data:	8 bit
Parity:	none
Stop bits:	1 bit
Flow control:	none

1 reset device


COM67 - Tera Term VT

```
  Success: aci_hal_write_config_data command - CONFIG_DATA_PUBADDR_OFFSET
  Public Bluetooth Address: 00:80:e1:2a:19:82
  Success: aci_hal_write_config_data command - CONFIG_DATA_IR_OFFSET
  Success: aci_hal_write_config_data command - CONFIG_DATA_ER_OFFSET
  Success: aci_hal_set_tx_power_level command
  Success: aci_gatt_init command
  Success: aci_gap_init command
  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_set_authentication_requirement command
==>> End Ble_Hci_Gap_Gatt_Init function

Services and Characteristics creation
  Success: aci_gatt_add_service command: P2P_Server

  Success: aci_gatt_add_char command   : LED_C
  Success: aci_gatt_add_char command   : SWITHC_C
End of Services and Characteristics creation

==>> aci_gap_set_discoverable - Success
==>> Success: Start Advertising
```

2 Connect


COM67 - Tera Term VT

```
>>== HCI_LE_CONNECTION_COMPLETE_SUBEVT_CODE - Connection handle: 0x0001
     - Connection established with @:77:1c:a8:d6:d9:5a
     - Connection Interval:    ms
     - Connection latency:    0
     - Supervision Timeout: 720 ms
```

Takeaways
What’s next

✓

Hands-on#1 – Basic BLE advertising device

Inherit of STM32 ecosystem and build a BLE advertising device application in few steps

save .ioc project file

STM32 CubeMX

↓

Hands-on#2 – Add BLE profile application code

Extend existing application code to enable proprietary profile (P2P_Server)

life.augmented

Thank you

life.augmented