

Hands-on #1

Build basic **p2pServer** application and connect

Agenda



1 Hands-on presentation

4 Step 3 : Code generation and user application code

2 Step 1: STM32CubeMX/STM32CubeIDE initialization for STM32WBA Nucleo board

5 “bonus track” : Adding logs

3 Step2 : Advertising and BLE application configuration and explanation



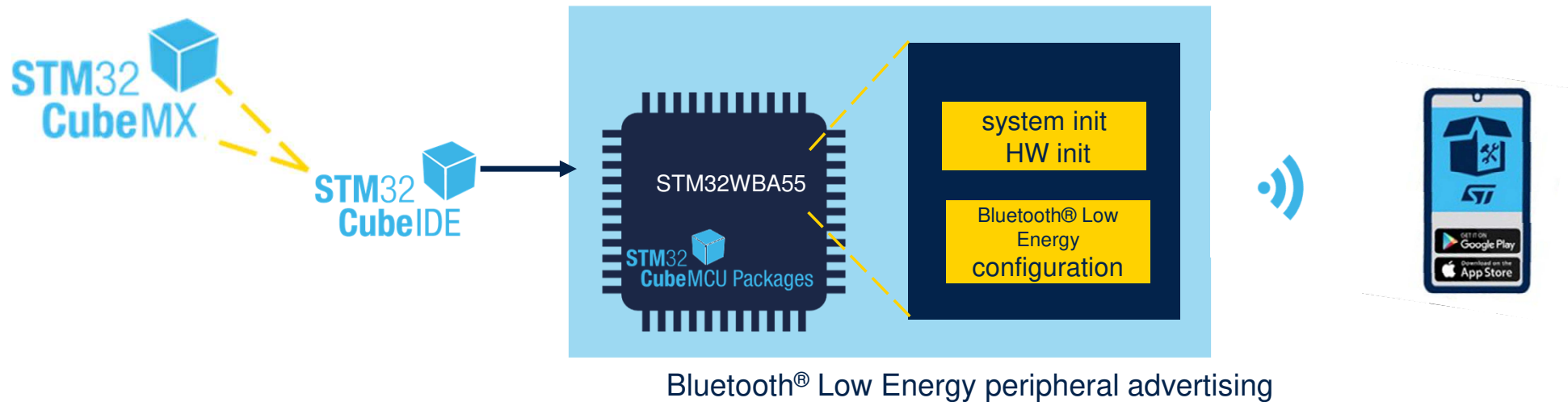
life.augmented

Hands-on presentation



Purpose

- The purpose is to start from WBA55 chipset level and build a basic server (**p2pServer**) application using STM32CubeMX/STM32CubeIDE
- In this first part, focus is to get device **visible and connectable** from my smartphone



Unpack NUCLEO-WBA55, plug to laptop,
install your favorite ST BLE ToolBox App and Let's start !



Source

Hands-on based on

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

step by step guideline to build a Bluetooth® Low Energy peripheral application



Bluetooth® Low Energy peripheral exposing data to central device



Legend

- Slides including following symbol are purely theoretical ones



- Source code for development is included inside blue boxes

```
HAL_Delay(500);
```

Step 1 : STM32CubeMX initialization for STM32WBA Nucleo board



STM32CubeMX capabilities



STM32CubeMX : “Standalone version” or “integrated version” into STM32CubeIDE 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_WBA55)

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 can be standalone application but also part of STM32CubeIDE





STM32CubeMX design from chipset level complete journey

STM32CubeMx initialisation for STM32WBA Nucleo board

STM32WBA IPs & peripherals configuration

Clock Tree configuration

BLE configuration : Advertising, Service, Characteristic

STM32
CubeIDE

Code generation & application code management over CubeIDE

STM32
CubeIDE

Hands-on
Focus





STM32CubeMx design from chipset level

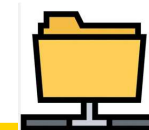
Hands-on focus (1/2)

3

Chipset level

require to configure your HW (PCB) & your application

To ease Hands-on session use [Hands-on_WS_WBA55.ioc](#)
All HW IPs & required peripheral to use RF are already initialized : NVIC, RNG, RCC,...
Thanks to [Hands-on_WS_WBA55.ioc](#) let's focus on BLE application design



Copy [Hands-on_WS_WBA55.ioc](#) on your local repository :

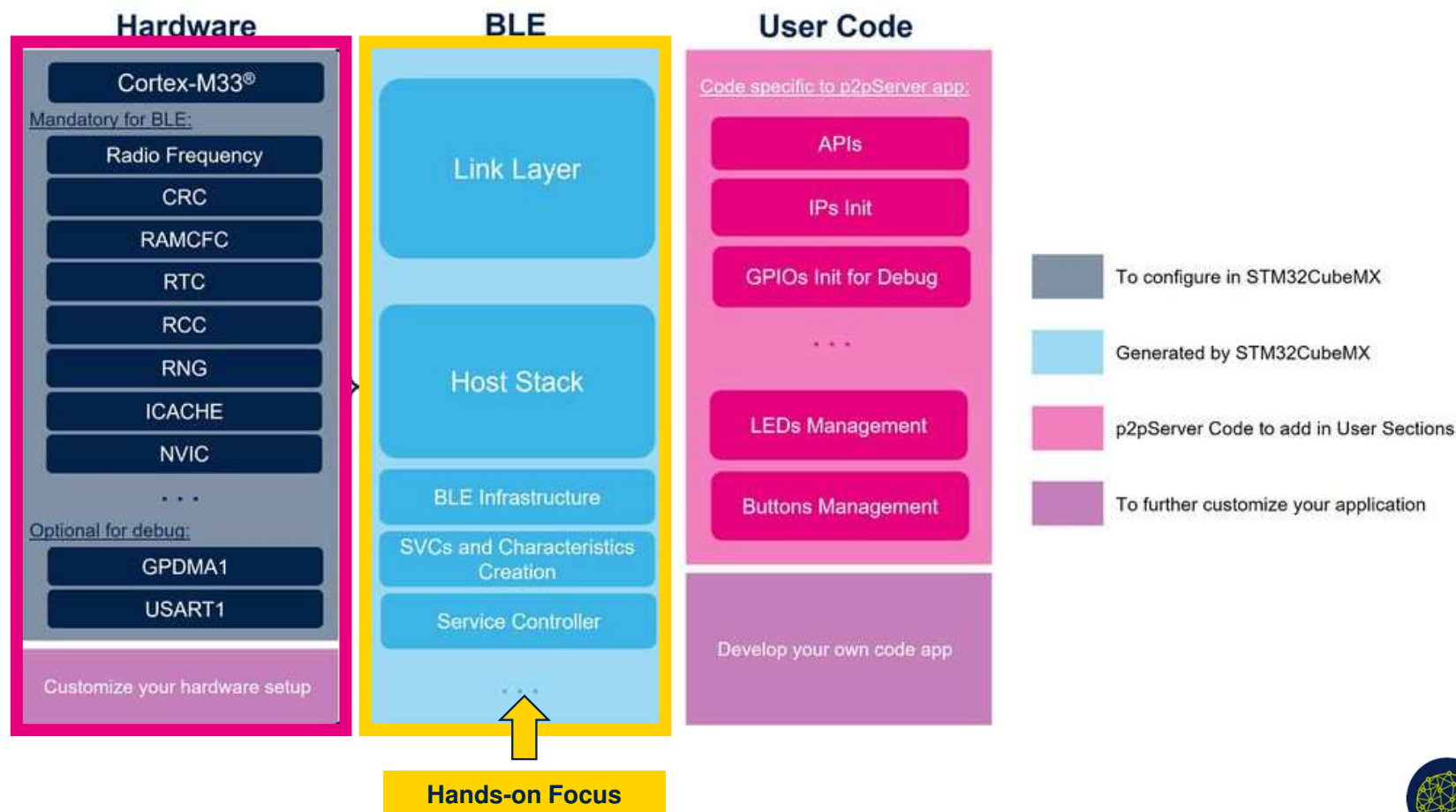
example : C:\users\...\STM32WBA_WS\project



STM32CubeMx design from chipset level

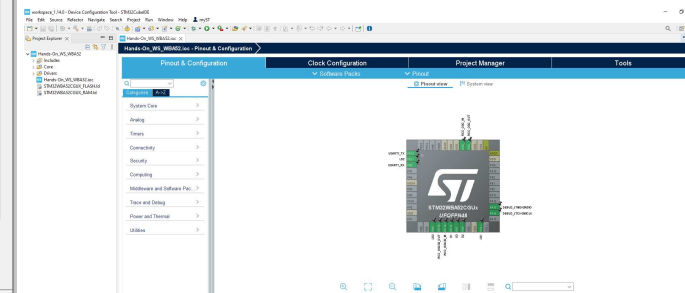
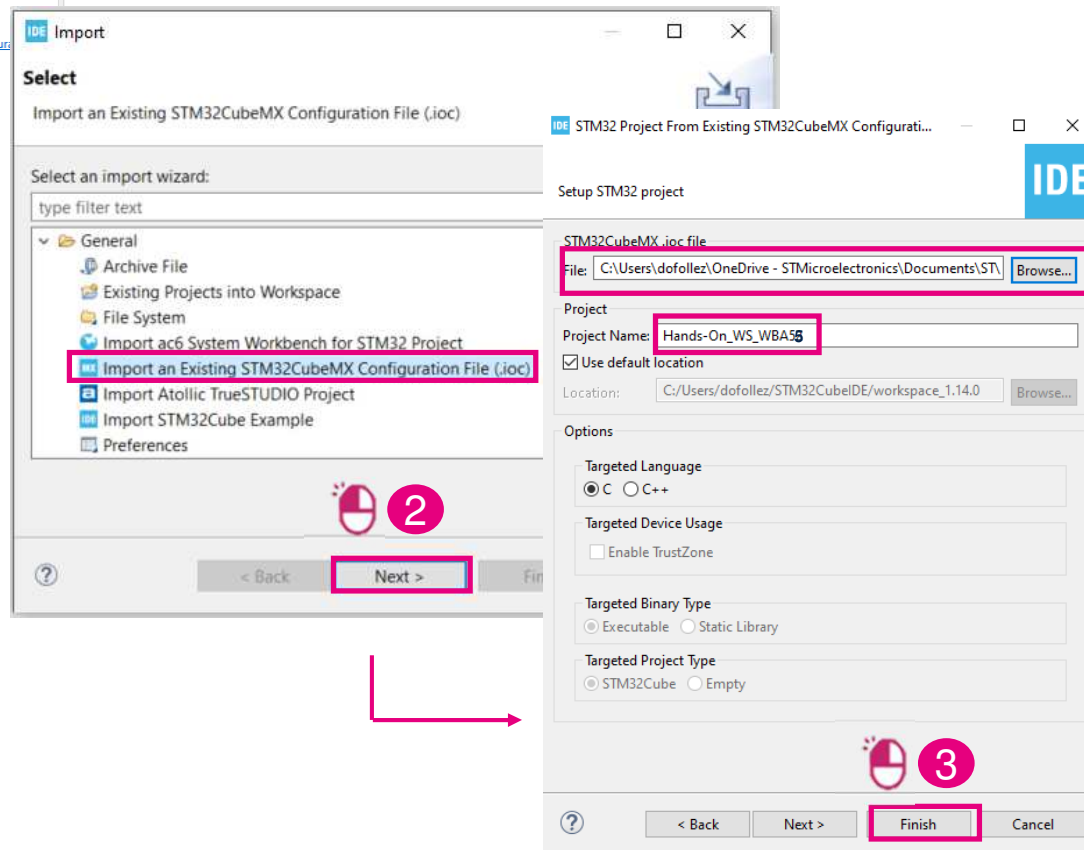
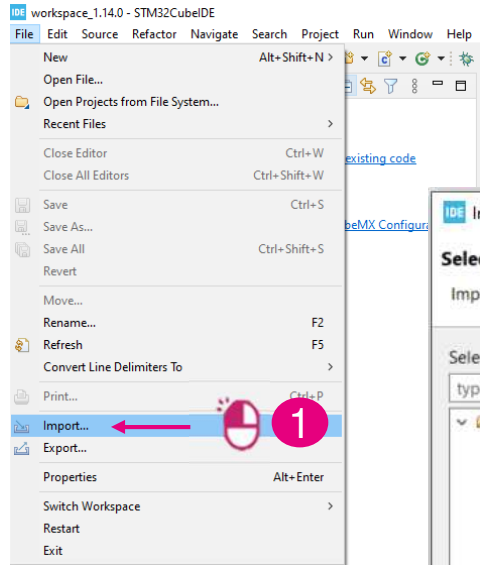
Hands-on focus (2/2)

Hands-On_WS_WBA55.ioc



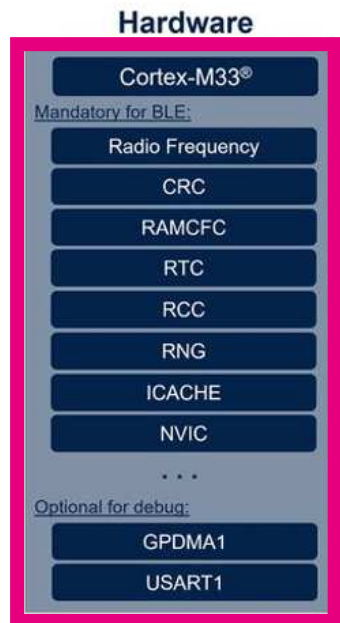


Open and Start STM32CubeIDE

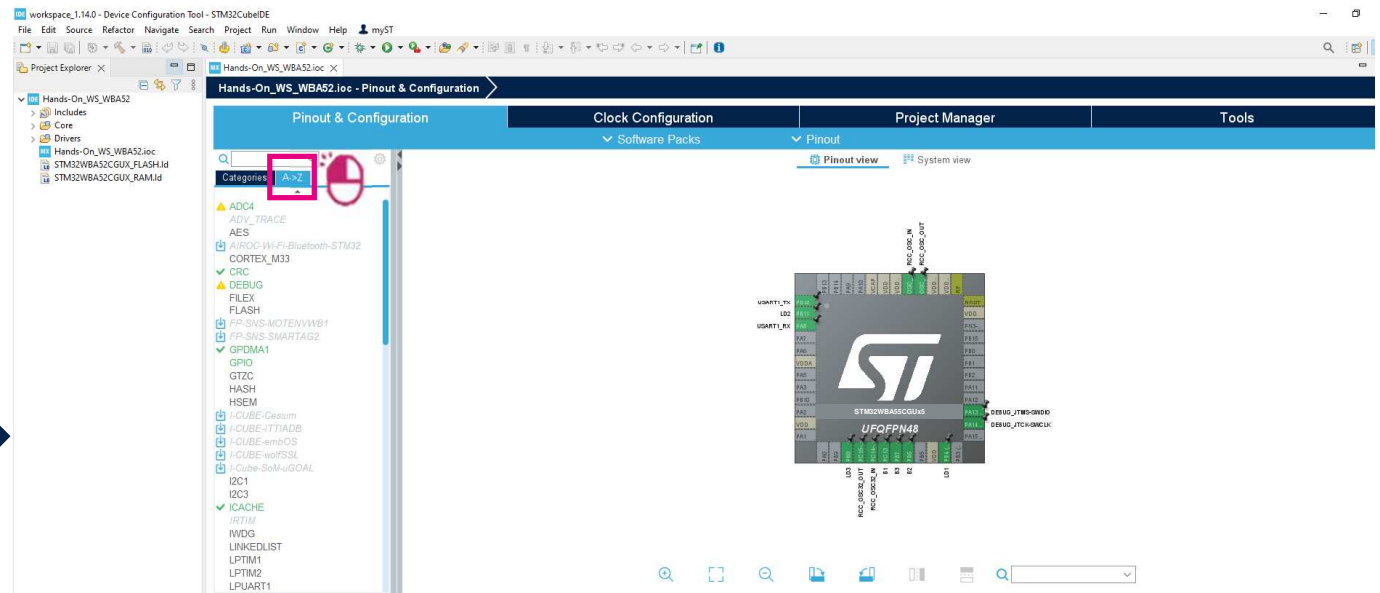




Peripherals in place to start BLE configuration !



Hands_On_WBA55.ioc



Hands_On_WBA55.ioc

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



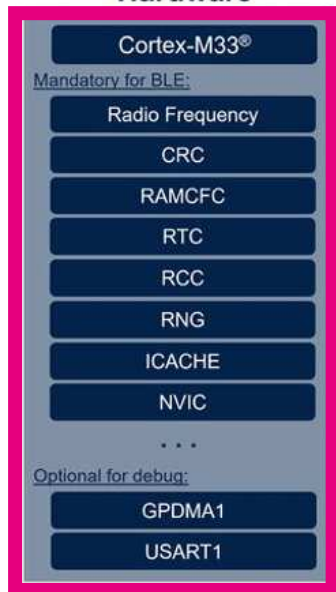
Peripherals in place to start BLE configuration !

Wiki explanations



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

Hardware



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	The Radio system is mandatory for a BLE project.
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.

Step2 : Advertising and Bluetooth® Low Energy GAP/GATT custom application configuration



Enabling Bluetooth® Low Energy

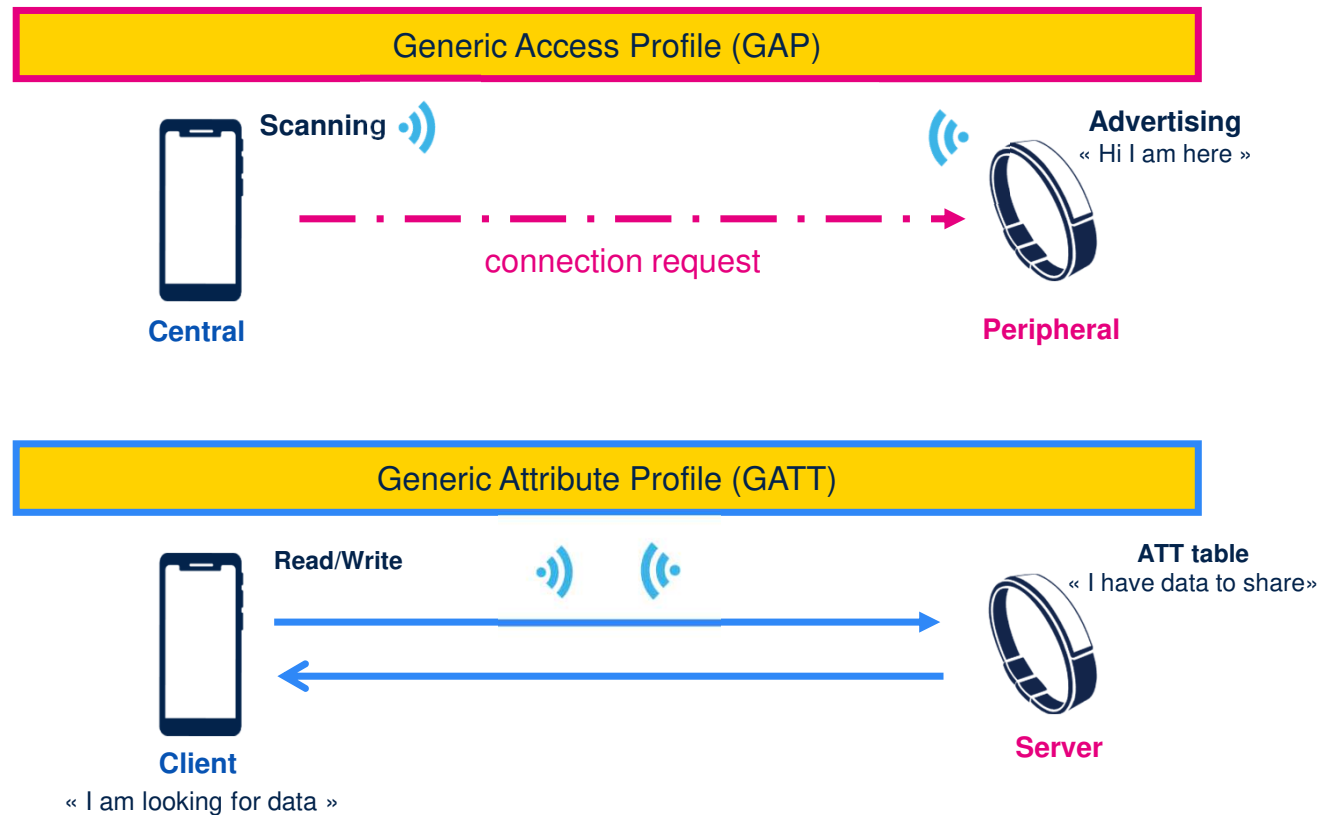
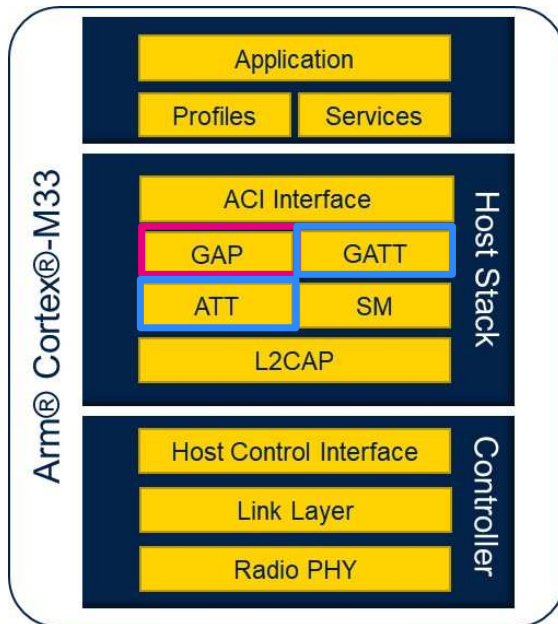
The screenshot shows the STM32CubeMX Pinout & Configuration tab. On the left, a list of components is shown with checkboxes. A red circle with the number 1 points to the 'STM32_WPAN' checkbox, which is checked. A red circle with the number 2 points to the 'A->Z' button in the 'Categories' section. On the right, the 'STM32_WPAN Mode and Configuration' section is shown. A red box highlights the 'BLE' mode selection, which is set to 'Create your Peripheral & GATT server application'. Below this, the 'THREAD' and 'ZIGBEE' modes are set to 'Disable', and the 'MAC 802.15.4' checkbox is unchecked.

Our Application is a peripheral server application.





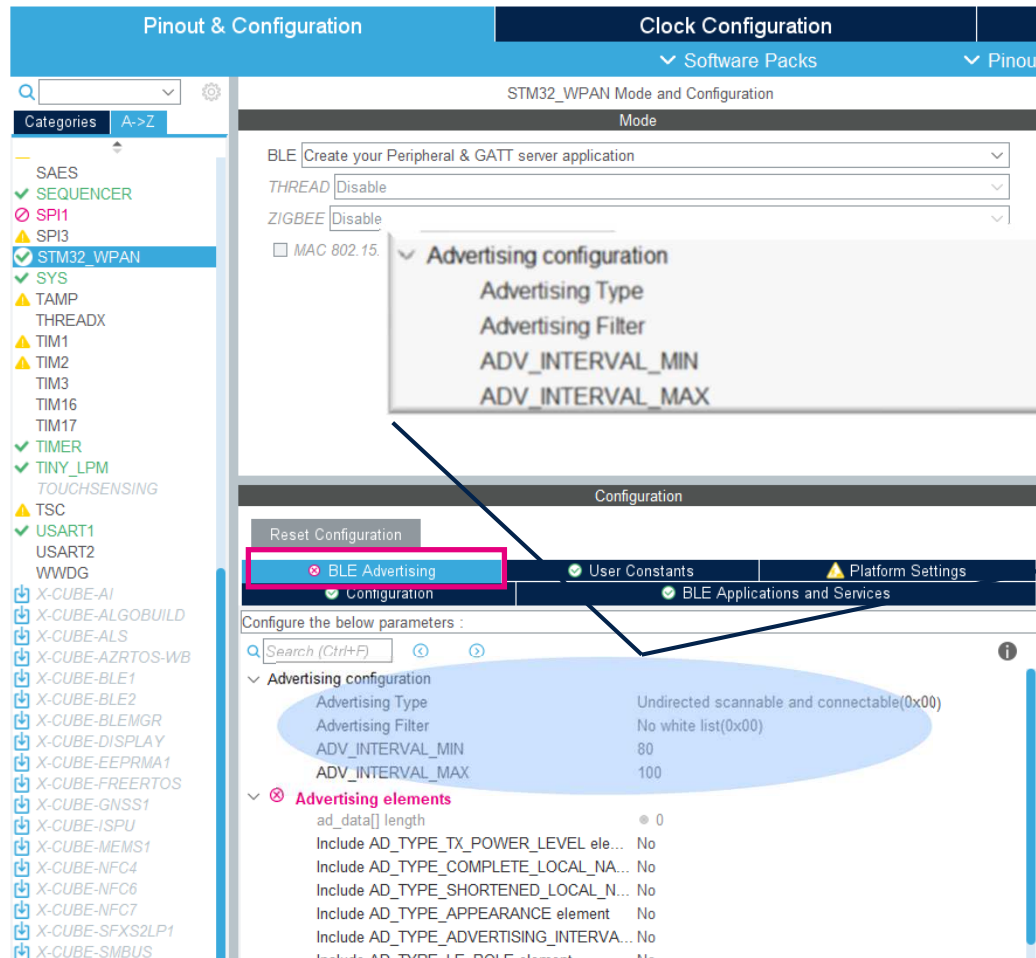
Bluetooth® Low Energy Connection roles vs. Data roles



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



Advertising Configuration



Advertising Type
accept connection requests from any peer device

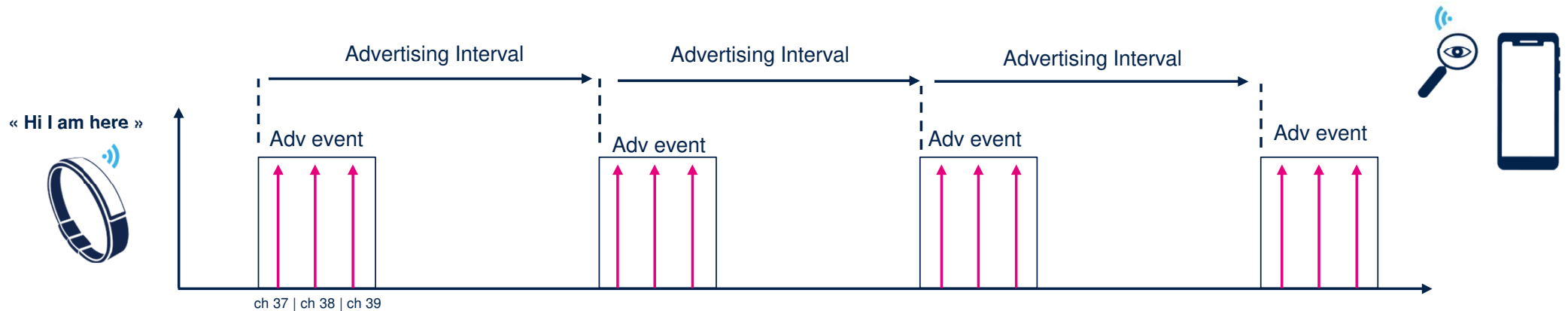
Advertising Filter
In general, used in case of Privacy.

Advertising Interval
Advertising set = $(\text{MIN} + \text{MAX}) / 2$
Min & Max used in case of multi connections
Units : ms



Advertising Configuration

Legacy Advertising Interval

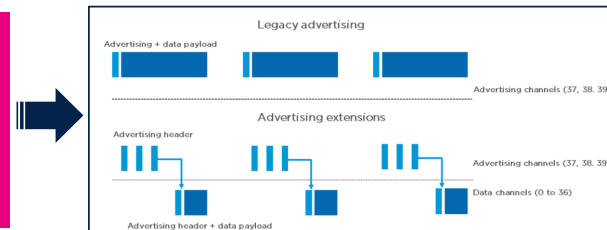


- The advertising interval value ranges all the way from **20** milliseconds up to **10.24** seconds in small increments of **625** microseconds.
- The advertising interval greatly impacts battery life and should be chosen carefully.

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)

WBA5x supporting advertising extension to increase your advertising data
Thanks to adv extension, Periodic advertising supported





Advertising Elements Local Name

Pinout & Configuration

Clock Configuration

Software Packs

Pinout

STM32_WPAN Mode and Configuration

Mode

BLE Create your Peripheral & GATT server application

THREAD Disable

ZIGBEE Disable

MAC 802.15.4

Configuration

Reset Configuration

Configuration

BLE Applications and Services

BLE Advertising

User Constants

Platform Settings

Configure the below parameters :

Search (Ctrl+F)

Advertising elements

ad_data[] length	25
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	130 00
Number of user defined data item(s)	12
User defined data 1	00
User defined data 2	00

Local Name length = Local name + 1

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.

Local Name length must be < 11
CubeMx constraints

If not ST Toolbox potential crash

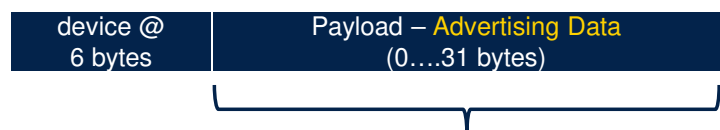


life.augmented

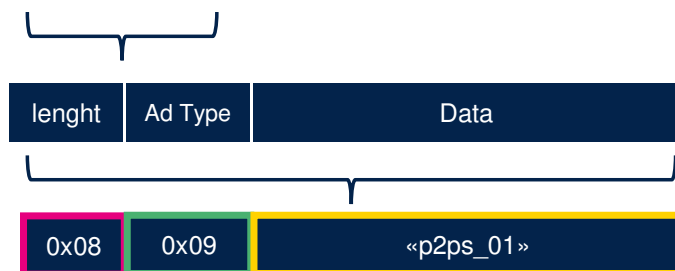


Advertising Elements

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

Raw data
0809703270535F30310FFF3000000000000000
00000000000000000000000000000000

Length	Type	Value
8	0x09	0x703270535F3031
15	0xFF	0x30000000000000000000000000000000
2	0x01	0x06

11 complete local name Ad Type «my_device»

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)



A0

Shall we open for scan rsp ? (adding complexity

Author; 2023-08-09T16:02:28.992



Advertising Elements Manufacturer Data

Pinout & Configuration | Clock Configuration | P

Software Packs | Pinout

STM32_WPAN Mode and Configuration

Mode

BLE Create your Peripheral & GATT server application

THREAD Disable

ZIGBEE Disable

☐ MAC 802.15.4

Configuration

Reset Configuration

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

Configure the below parameters :

Search (Ctrl+F)

Advertising elements

ad_data[] length	25
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
User defined data 1	00
Comment data 1	
User defined data 2	00

Manufacturer Ad Type , with company ID 0x30 (STMicroelectronics)



Allow to detect device as an ST device
and to connect as P2P profile





Customize Device Name

Pinout & Configuration | Clock Configuration | Project Manager

Software Packs | Pinout

STM32_WPAN Mode and Configuration

Mode

BLE Create your Peripheral & GATT server application

THREAD Disable

ZIGBEE Disable

MAC 802.15.4

Configuration

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

Configure the below parameters :

Search (Ctrl+F)

- BLE_OPTIONS_DEV_NAME_READ_ONLY	with service changed description
- BLE_OPTIONS_EXTENDED_ADV	Device name is read/write
- BLE_OPTIONS_REDUCED_DB_IN_NVM	Extended advertising is not supported
- BLE_OPTIONS_GATT_CACHING	Full GATT DB in NVM
- BLE_OPTIONS_POWER_CLASS_1	GATT caching is not used
- BLE_OPTIONS_APPEARANCE_WRITABLE	Power Class 2-3
- BLE_OPTIONS_APPEARANCE_READ_ONLY	Appearance is read only

Application configuration - Application parameters

CFG_TX_POWER	-0.3 dBm (0x19)
CFG_BD_ADDRESS	0x0008E12A1234
Address Type	Public address(0)
CFG_PRIVACY	Disabled
PAIRING_PARAMETERS	OFF
CFG_GAP_APPEARANCE	GAP_APPEARANCE_UNKNOWN
CFG_PHY_PREF	Preference on both TX and RX PHY
CFG_PHY_PREF_TX	2M_PREF
CFG_PHY_PREF_RX	2M_PREF
CFG_IO_CAPABILITY	Display Yes No (0x01)
CFG_MITM_PROTECTION	MITM protection required (0x01)
CFG_BLE_IR	12, 34, 56, 78, 9A, BC, DE, F0, 12, 34, 56, 78, 9A, BC, DE, F0
CFG_BLE_ER	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

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)



Platform Settings

Trace & Logs: BSP settings

Pinout & Configuration | Clock Configuration | Project Manager

▼ Software Packs | ▼ Pinout

STM32_WPAN Mode and Configuration

Mode

BLE Create your Peripheral & GATT server application

THREAD Disable

ZIGBEE Disable

☐ MAC 802.15.4

Configuration

Reset Configuration

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

Platform proposal

BSP

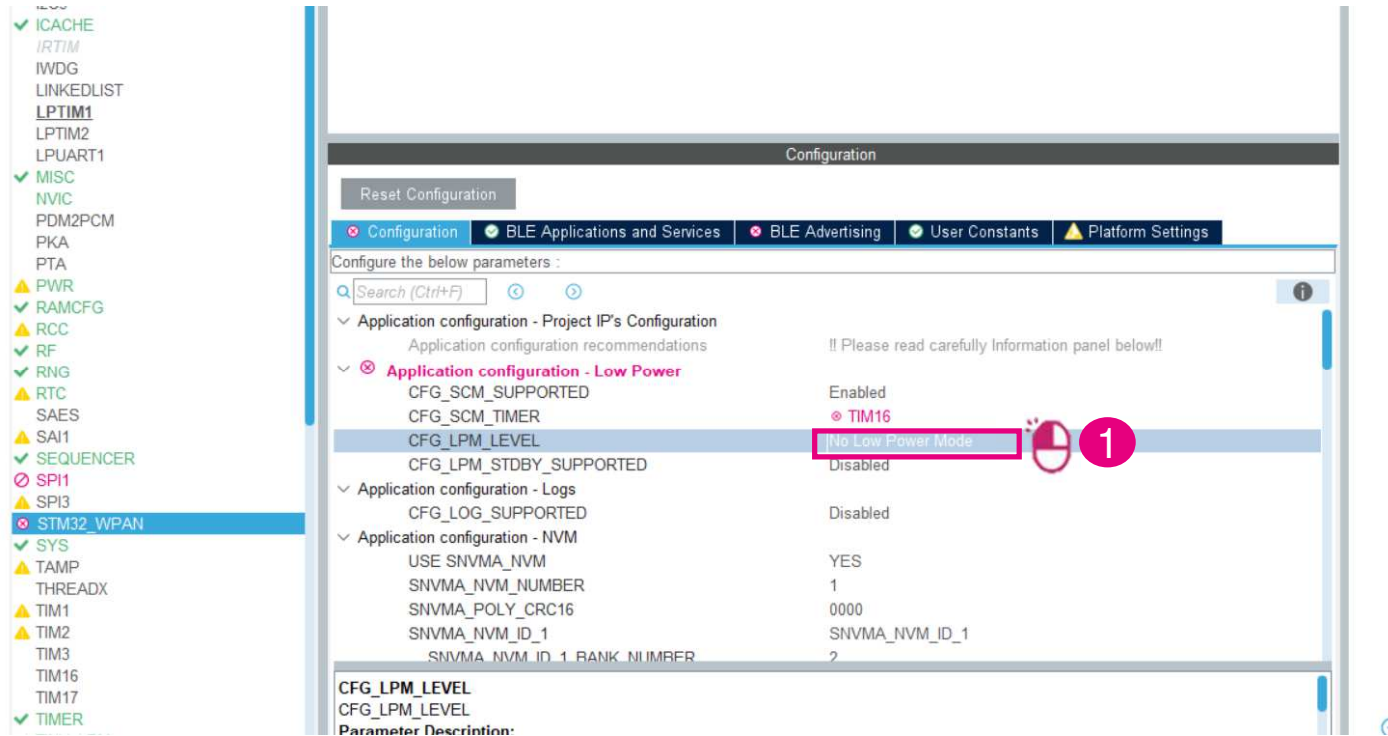
Name	IPs or Components	Found Solutions	BSP API
Serial Link for Logs	USART:Asynchronous	USART1	Unknown

Anticipate Logs activation

Logs activation would require application configuration changes



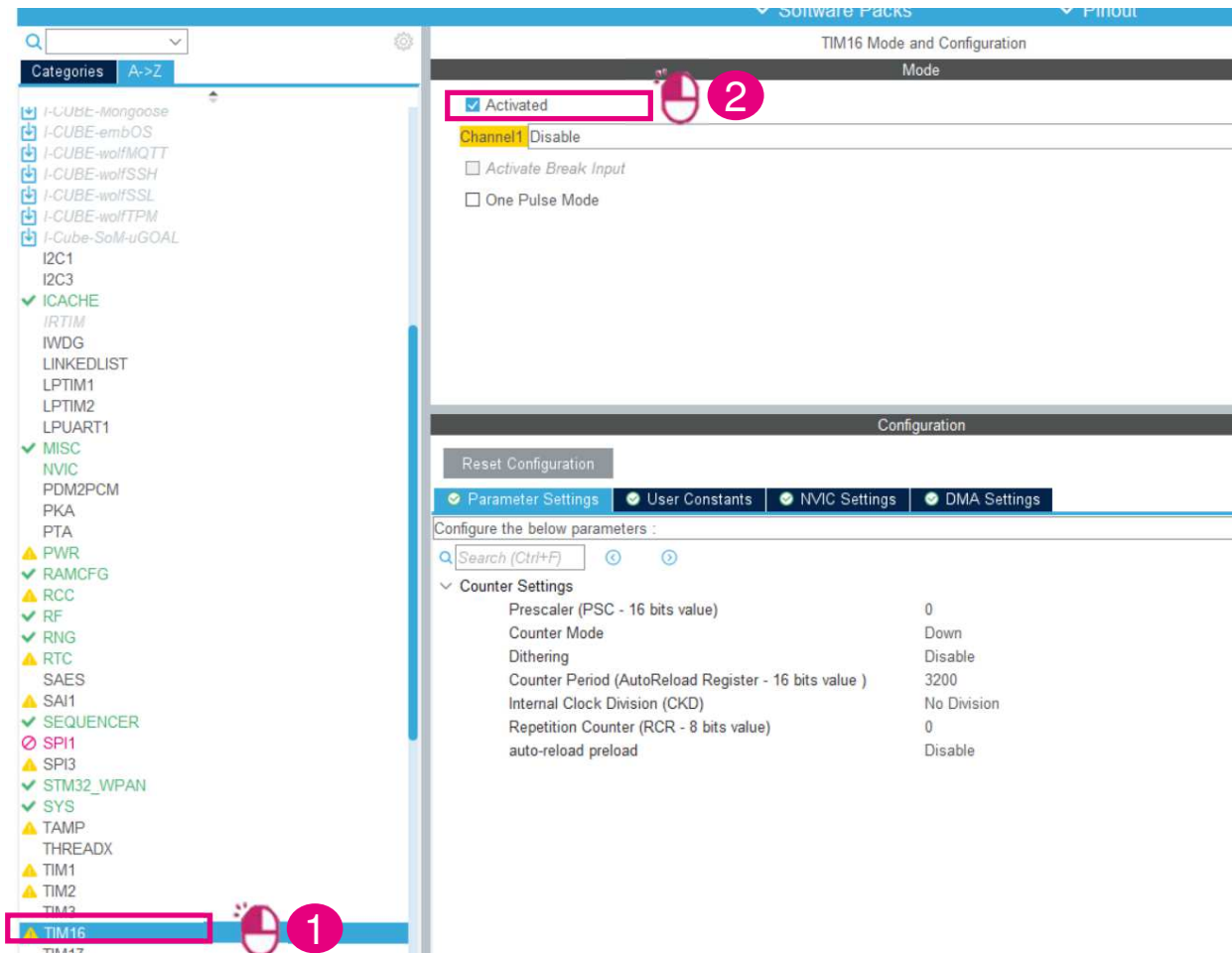
Configuration Disable: Low Power



As sanity check to avoid debugger connection



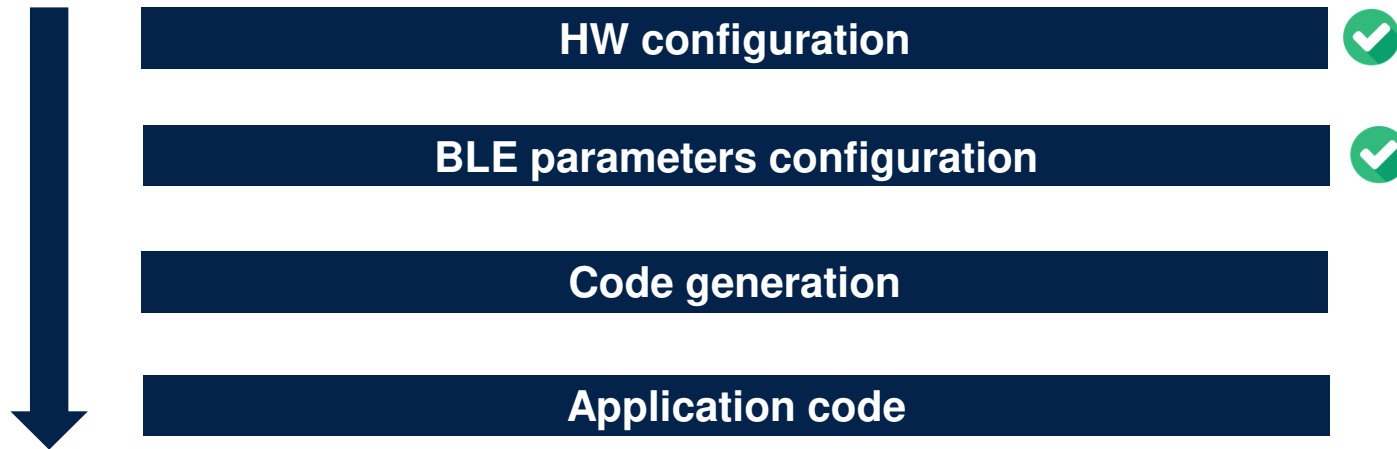
Configuration Enable: TIM16





Configuration completed

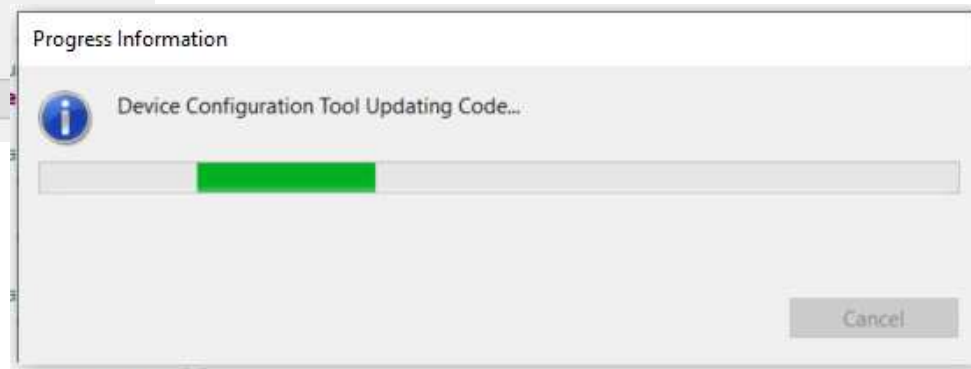
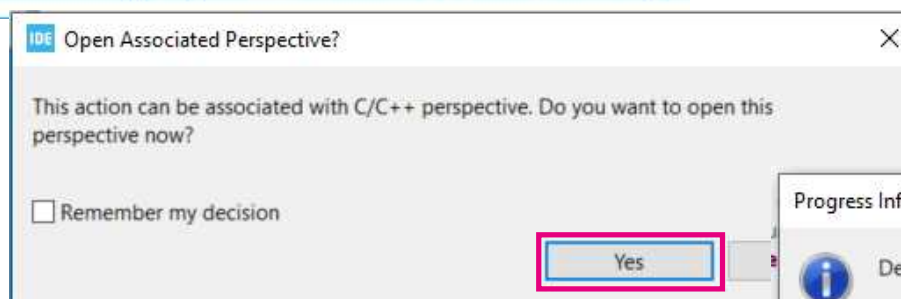
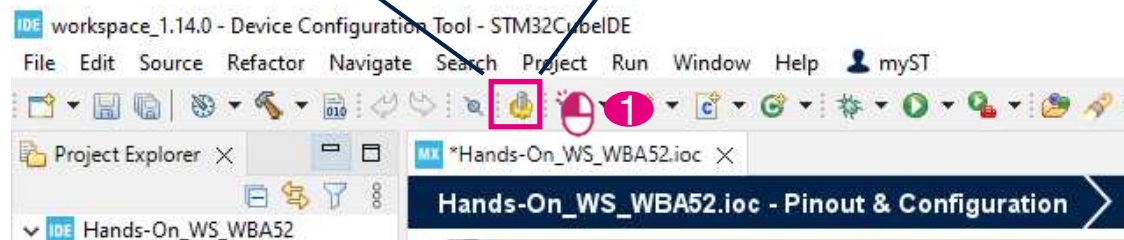
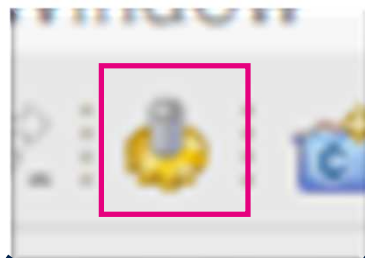
What's next : code generation ?



Step 3 : Code generation and user application code



Code Generation



Slide 29

A0

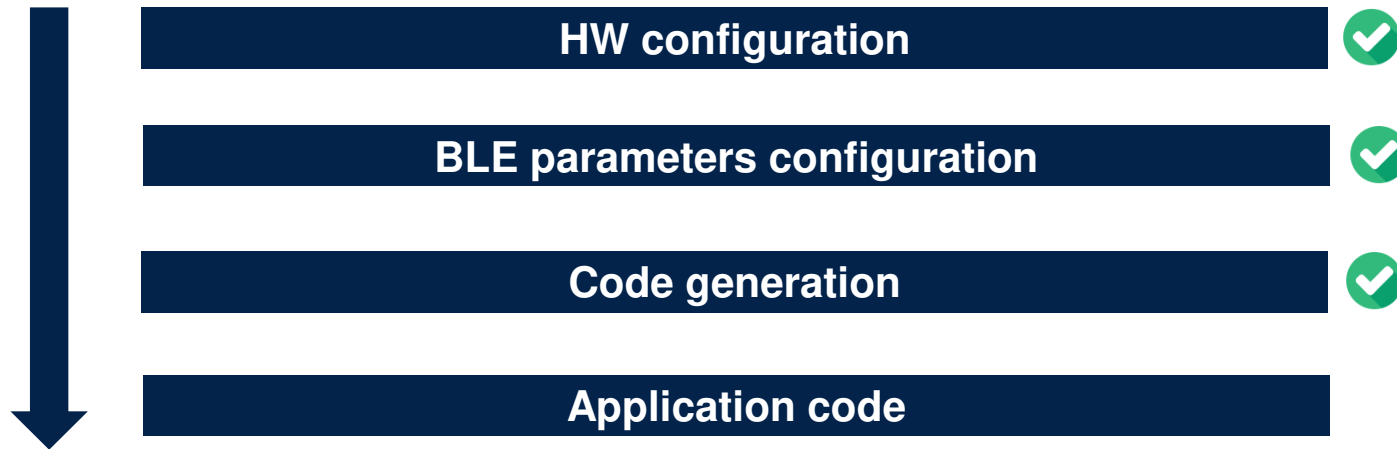
Put a small comment also on the first two tabs

Author; 2023-08-03T11:59:22.255



Configuration completed

What's next : code generation ?





Here are our ADV data

workspace_1.14.0 - Hands-On_WS_WBA52/STM32_WPAN/App/app_ble.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help myST

Project Explorer

- Hands-On_WS_WBA52
 - Includes
 - Core
 - Drivers
 - Middlewares
 - STM32_WPAN
 - App
 - app_ble.c
 - app_ble.h
 - ble_conf.h
 - ble_dbg_conf.h
 - Target
 - System
 - Utilities
 - Hands-On_WS_WBA52.ioc
 - STM32WBA52CGUX_FLASH.Id
 - STM32WBA52CGUX_RAM.Id

```

163 (uint8_t)((CFG_BD_ADDRESS & 0x00000000FF00) >> 8),
164 (uint8_t)((CFG_BD_ADDRESS & 0x000000FF0000) >> 16),
165 (uint8_t)((CFG_BD_ADDRESS & 0x0000FF000000) >> 24),
166 (uint8_t)((CFG_BD_ADDRESS & 0x00FF00000000) >> 32),
167 (uint8_t)((CFG_BD_ADDRESS & 0xFF0000000000) >> 40)
168 };
169
170 static uint8_t a_BdAddrUdn[BD_ADDR_SIZE];
171
172 /* Identity root key used to derive IRK and DHK(Legacy) */
173 static const uint8_t a_BLE_CfgIrValue[16] = CFG_BLE_IR;
174
175 /* Encryption root key used to derive LTK(Legacy) and CSRK */
176 static const uint8_t a_BLE_CfgErValue[16] = CFG_BLE_ER;
177 static BleApplicationContext_t bleAppContext;
178
179 static const char a_GapDeviceName[] = { 'p', '2', 'p', 's', '_', '0', '1' }; /* Gap Device Name */
180
181 /* Advertising Data */
182 uint8_t a_AdvData[25] =
183 {
184     8, AD_TYPE_COMPLETE_LOCAL_NAME, 'p', '2', 'p', 's', '_', '0', '1', /* Complete name */
185     15, AD_TYPE_MANUFACTURER_SPECIFIC_DATA, 0x30, 0x00, 0x00 /* */, 0x00 /* */, 0x00 /* */, 0x00 /* */, 0x00 /* */, 0x00 /* */, 0x00 /* */, 0x00 /* */, 0x00 /* */, 0x00 /* */, 0x00 /* */, 0x00 /* */,
186 };
187 uint8_t buffer_nvmm[CFG_BLEPLAT_NVMM_MAX_SIZE] = {0};
188
189 static APP_VirtualMemoryCallbackFunction_t APP_BLE_ResumeFlowProcessCb;
190
191 /* Host stack init variables */
192 static uint32_t buffer[DIVC(BLE_DYN_ALLOC_SIZE, 4)];
193 static uint32_t gatt_buffer[DIVC(BLE_GATT_BUF_SIZE, 4)];
194 static BleStack_init_t pInitParams;
195
196 /* USER CODE BEGIN PV */
197
198 /* USER CODE END PV */
199
200 /* Global variables -----*/
201

```

Slide 31

A0

[Mention was removed] as agreed I let you to adjust it 😊

Author; 2023-09-21T14:02:26.277

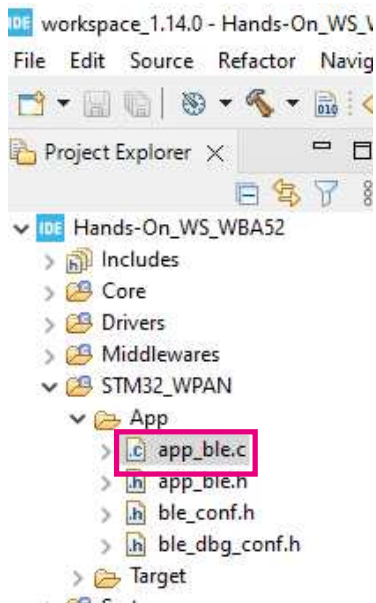


Open Project

Add application code to move to discoverable (1/2)

Set device discoverable at init :

In app_ble.c > function APP_BLE_Init()



```
/* USER CODE BEGIN APP_BLE_Init_2 */
tBleStatus status;
status = aci_gap_set_discoverable(ADV_TYPE, ADV_INTERVAL_MIN, ADV_INTERVAL_MAX,
                                CFG_BD_ADDRESS_TYPE,
                                ADV_FILTER,
                                0, 0, 0, 0, 0, 0);
if (status != BLE_STATUS_SUCCESS) {
    return;
}

status = aci_gap_delete_ad_type(AD_TYPE_TX_POWER_LEVEL);
if (status != BLE_STATUS_SUCCESS) {
    return;
}

status = aci_gap_update_adv_data(sizeof(a_AdvData), (uint8_t*) a_AdvData);
if (status != BLE_STATUS_SUCCESS) {
    return;
}
/* USER CODE END APP_BLE_Init_2 */
```

To accommodate the Advertising payload, remove the Tx power Adv Type set by stack

🔍 Search for "APP_BLE_Init_2"

Open Project

Add application code to move to discoverable

Build & Flash



Please refer to cheatsheet for copy/paste



Open Project

Add application code to move to discoverable (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 */
tBleStatus status;
status = aci_gap_set_discoverable(ADV_TYPE, ADV_INTERVAL_MIN, ADV_INTERVAL_MAX,
                                CFG_BD_ADDRESS_TYPE,
                                ADV_FILTER,
                                0, 0, 0, 0, 0, 0);
if (status != BLE_STATUS_SUCCESS) {
    LOG_INFO_APP("==>> aci_gap_set_discoverable - fail, result: 0x%02X\n", status);
}

status = aci_gap_delete_ad_type(AD_TYPE_TX_POWER_LEVEL);
if (status != BLE_STATUS_SUCCESS) {
    LOG_INFO_APP("==>> delete tx power level - fail, result: 0x%02X\n", status);
}

status = aci_gap_update_adv_data(sizeof(a_AdvData), (uint8_t*) a_AdvData);
if (status != BLE_STATUS_SUCCESS) {
    LOG_INFO_APP("==>> Start Advertising Failed, result: 0x%02X\n", status);
}
/* USER CODE END EVT_DISCONN_COMPLETE */
```

Search for "EVT_DISCONN_COMPLETE"

1 Build

2 Flash

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

Open Project

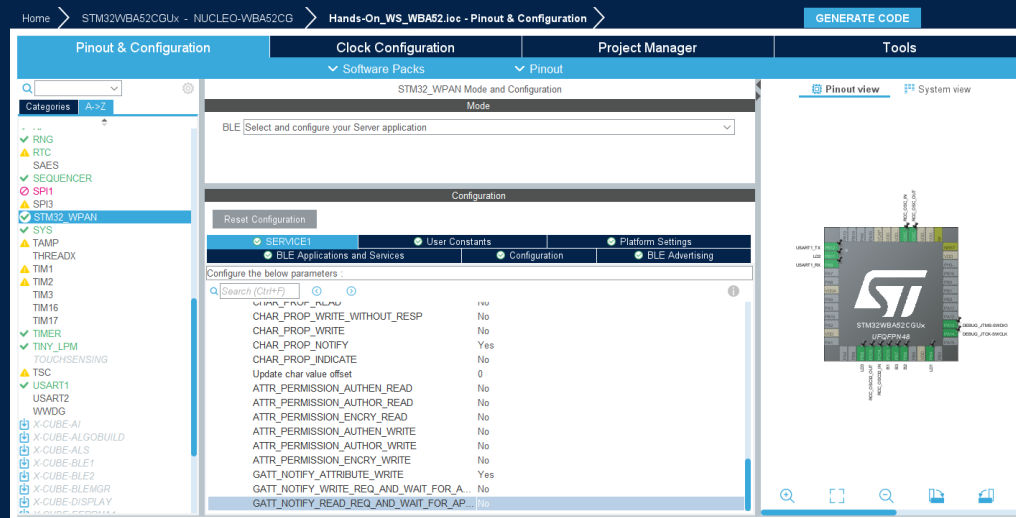
Add application code to move to discoverable

Build & Flash

Please refer to cheatsheet for copy/paste

Bonus : Add debug capabilities

Move back to
STM32CubeIDE/STM32CubeMX





Application configuration

Traces & logs

Pinout & Configuration | Clock Configuration | Project Manager

Software Packs | Pinout

STM32_WPAN Mode and Configuration

Mode

BLE Create your Peripheral & GATT server application

THREAD Disable

ZIGBEE Disable

☐ MAC 802.15.4

Configuration

Reset Configuration

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

Configure the below parameters :

Search (Ctrl+F)

Application configuration - Low Power

CFG_LPM_LEVEL LPM enabled

CFG_LPM_STDBY_SUPPORTED Disabled

Application configuration - Logs

CFG_LOG_SUPPORTED Enabled

* CFG_LOG_INSERT_TIME_STAMP_INSIDE_THE_TRACE Disabled

* CFG_LOG_INSERT_COLOR_INSIDE_THE_TRACE Disabled

* CFG_LOG_INSERT_EOL_INSIDE_THE_TRACE Disabled

* CFG_LOG_VERBOSE_LEVEL LOG_VERBOSE_INFO

Application configuration - NVM

USE_SNVM_NVM YES

SNVM_NVM_NUMBER 1

SNVM_NVM_POLY_CRC16 0000

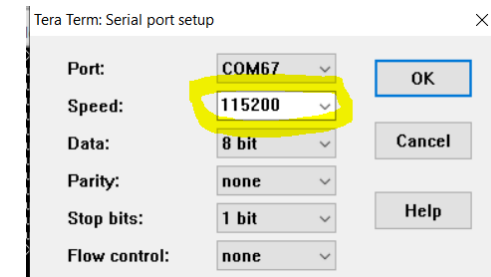
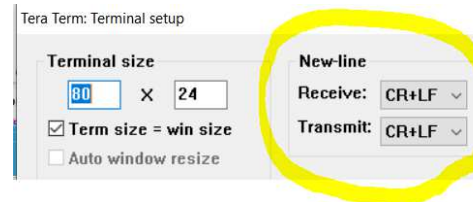
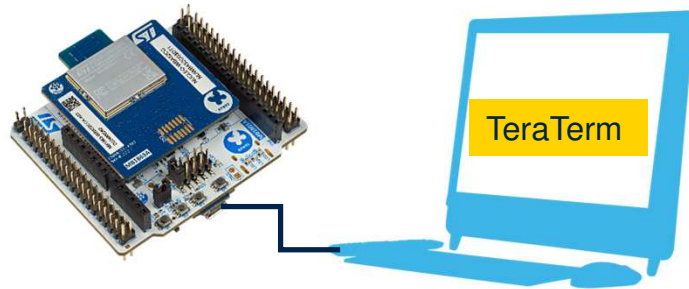
SNVM_NVM_ID_1 SNVM_NVM_ID_1

SNVM_NVM_ID_1_BANK_NUMBER 2

SNVM_NVM_ID_1 Number of Buffer 1



Open your App and Connect



1

reset device



```
COM67 - Tera Term VT
File Edit Setup Control Window Help
Success: aci_hal_write_config_data command - CONFIG_DATA_PUBADDR_OFFSET
Public Bluetooth Address: 00:80:ea: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_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 : SWITCH_C
End of Services and Characteristics creation
==> aci_gap_set_discoverable - Success
==> Success: Start Advertising
```

2

Connect



```
COM67 - Tera Term VT
File Edit Setup Control Window Help
>>== HCI_LE_CONNECTION_COMPLETE SUBEVENT CODE - Connection handle: 0x0001
- Connection established with 8:77:1c:a8:d6:d9:5a
- Connection Interval: 10 ms
- Connection latency: 0
- Supervision Timeout: 720 ms
```



Takeaways

What's next



Hands-on#1 – Basic Bluetooth® Low Energy advertising device

Inherit of STM32 ecosystem and build a Bluetooth® Low Energy advertising device application in few steps

save .ioc project file



Hands-on#2 – Add Bluetooth® Low Energy profile application code

Extend existing application code to enable proprietary profile (P2P_Server)

Thank you

© STMicroelectronics - All rights reserved.

The STMicroelectronics corporate logo is a registered trademark of the STMicroelectronics group of companies. All other names are the property of their respective owners.



life.augmented