CubeMX 创建 WL LoRa AT_Slave 应用

关键字：STM32WL, LORA

前言

本篇 LAT 介绍使用 CubeMX 创建基于 STM32WL 的 LoRa 应用。

STM32Cube_FW_WL_V1.0.0\Projects\NUCLEO-WL55JC 中的例程都是基于 STM32WL BGA73 的，CubeMX 无法直接创建基于 STM32WL QFN48 的例程。

同时介绍如何参考 STM32Cube_FW_WL_V1.0.0\Projects\NUCLEO-WL55JC\Applications\LoRaWAN\LoRaWAN_AT_Slave 创建一个 WLEx_AT_Slave 的例子。

关于不同 MCU 间的移植可参考

UM1718 11.9 Switching to another MCU

UM1718 15 Tutorial 5: Exporting current project configuration to a compatible MCU

需要提前准备的环境：

STM32WL 官网		https://www.st.com/STM32WL https://www.stmcu.com.cn/STM32WL
硬件	HW	NUCLEO-WL55JC https://www.st.com/en/evaluation-tools/nucleo-wl55jc.html MB1389-WL55JC-highband-D04 Schematic MB1389-WL55JC-lowband-D04 Schematic
代码生成工具	CubeMX	STM32CubeMX v6.2.0 https://www.st.com/en/development-tools/stm32cubemx.html
软件源码库	CubeWL	STM32Cube_FW_WL_V1.0.0 https://www.st.com/en/embedded-software/stm32cubewl.html
集成开发环境	IAR	8.50.9 (以上版本不需要打补丁) https://netstorage.iar.com//SuppDB/Public/UPDINFO/015020/arm/doc/infocenter/readme.ENU.html

IAR v8.50.9 以下版本需要打补丁

STM32Cube_FW_WL_V1.0.0\Utilities\PC_Softwar\EWARMv8_STM32WLxx_V4.6.zip

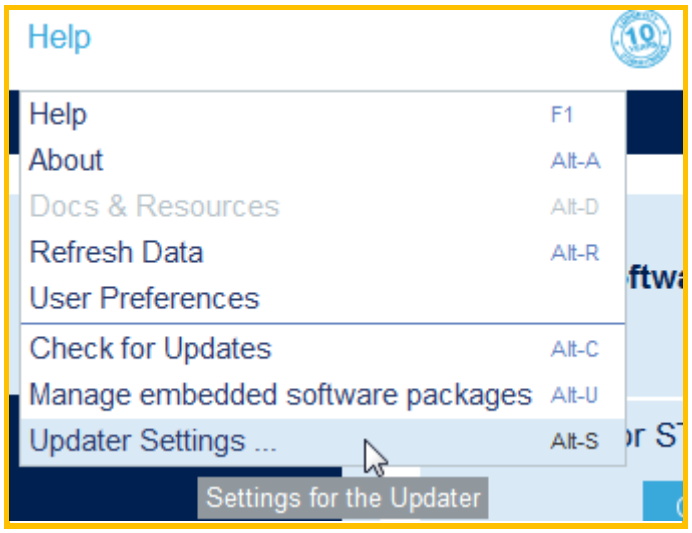
1 配置 CubeMX

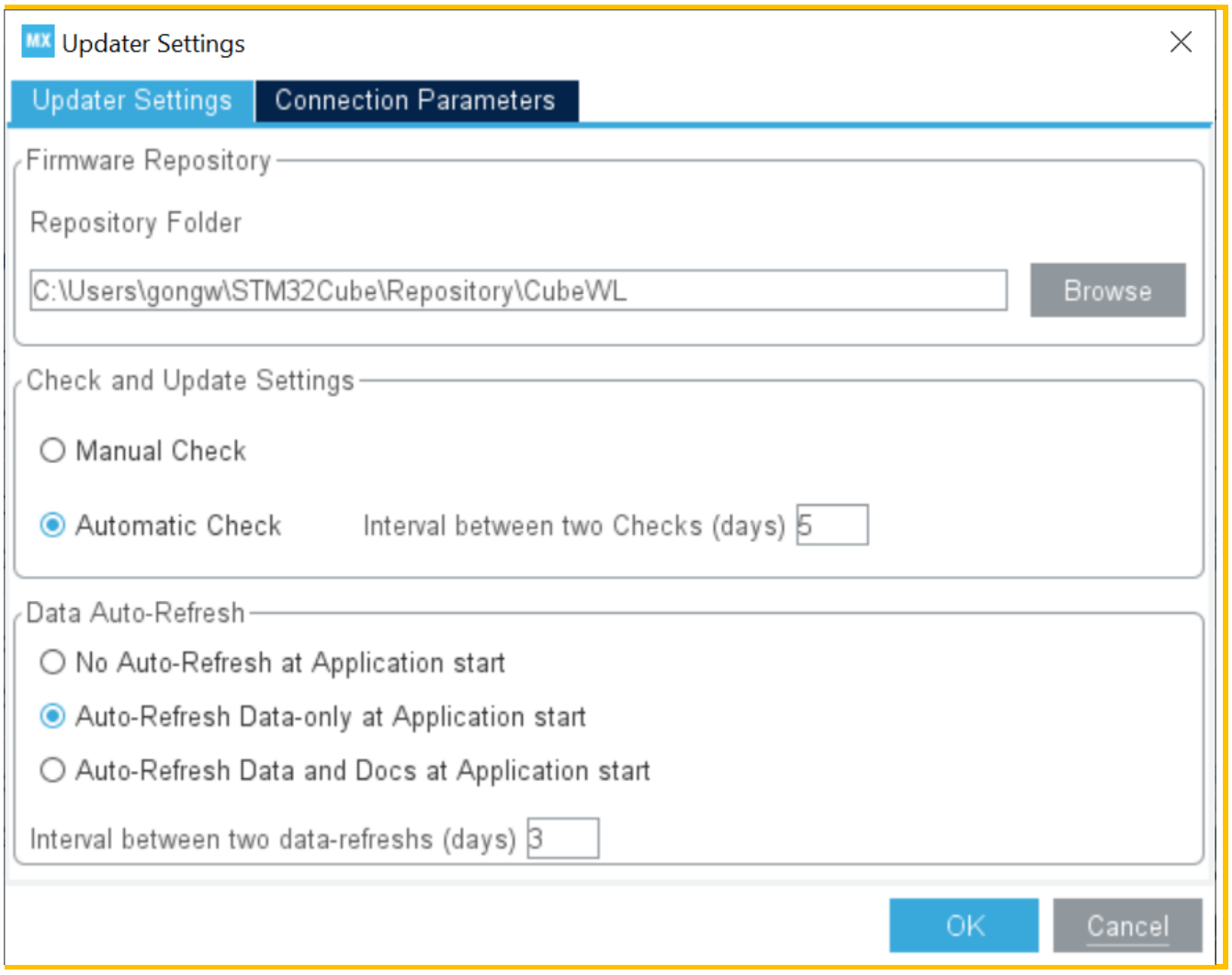
1.1 配置 CubeMX 并下载 CubeWL

可以使用 CubeMX 默认安装的仓库目录

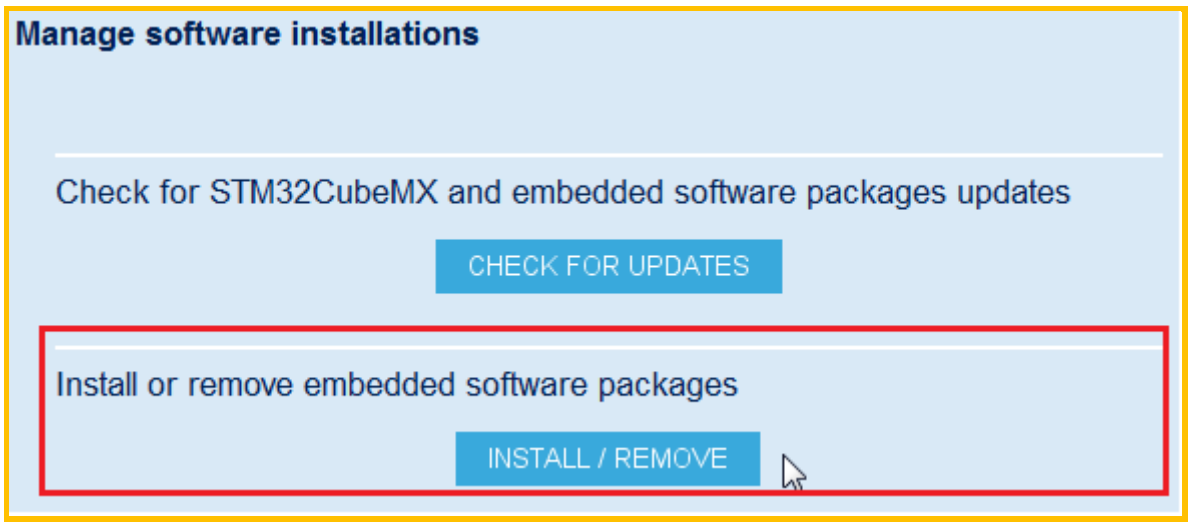
也可以自定义仓库目录，本例使用自定义的仓库目录 C:\Users\gongw\STM32Cube\Repository\CubeWL，如下

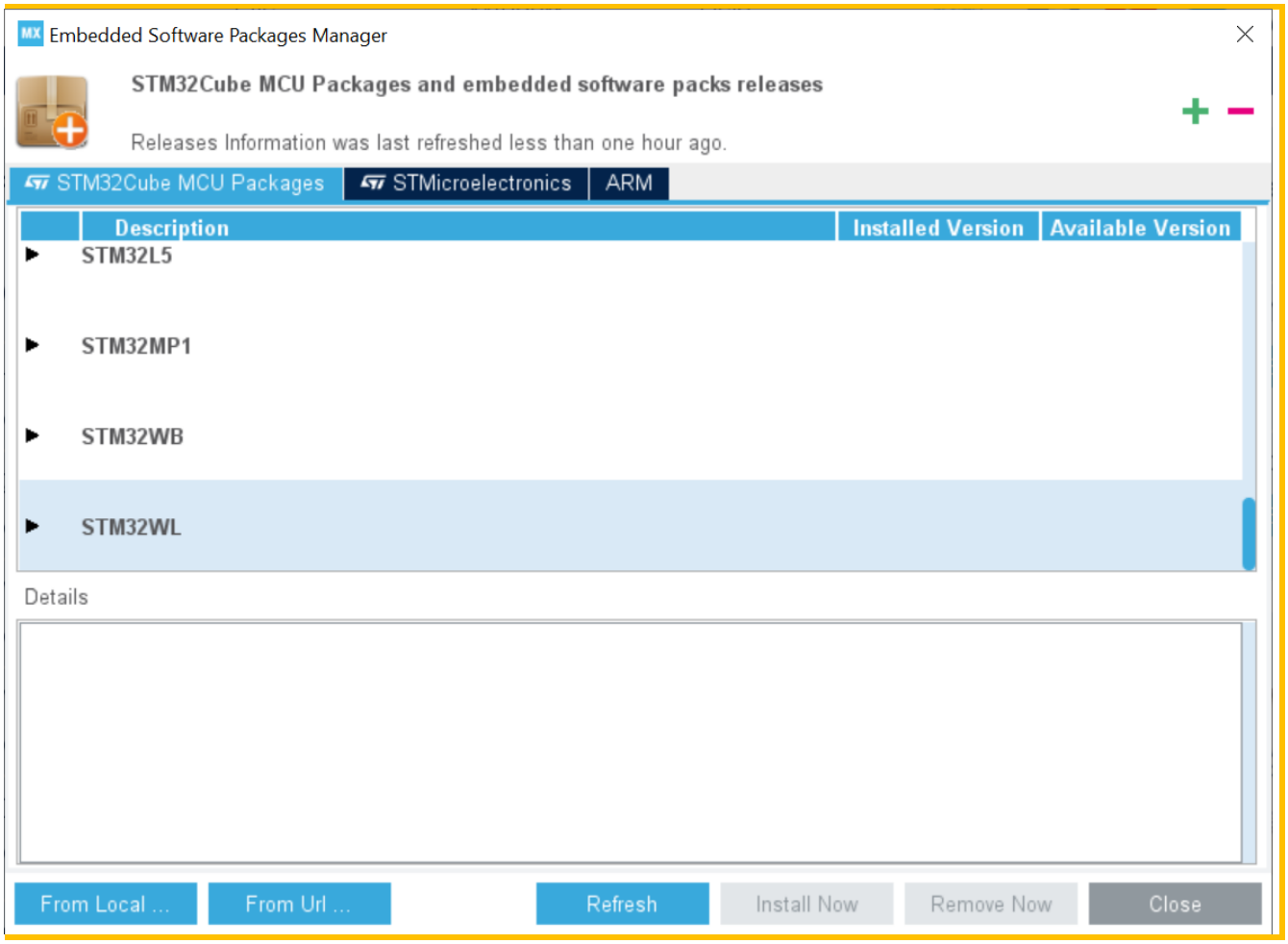
Help=>Updater Settings=>Repository Forlder=>C:\Users\gongw\STM32Cube\Repository\CubeWL=>OK



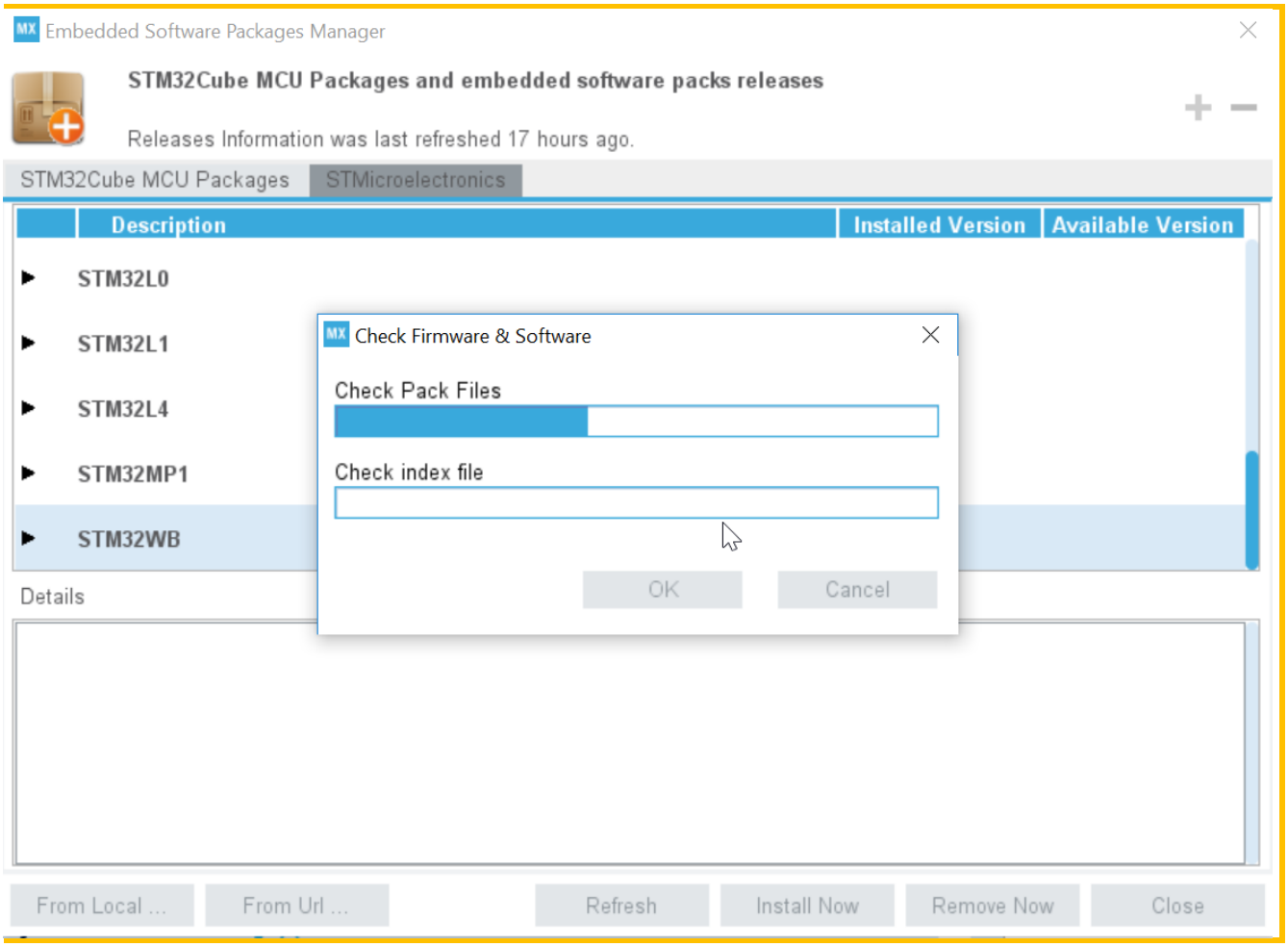


选择 Manage software installations 中的 Install or remove embedded software packages 下的 “INSTALL / REMOVE”

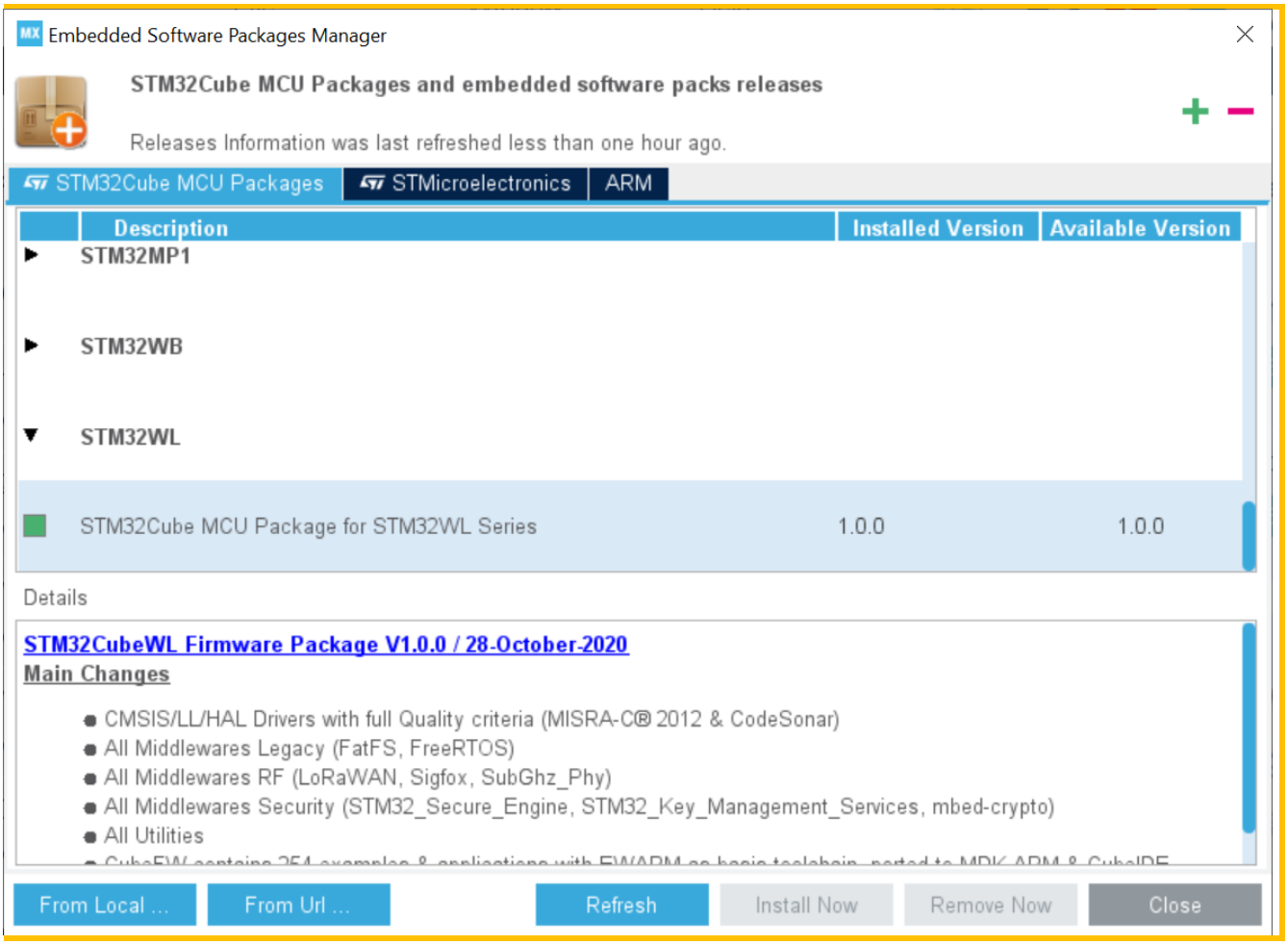




点 Refresh 更新数据库



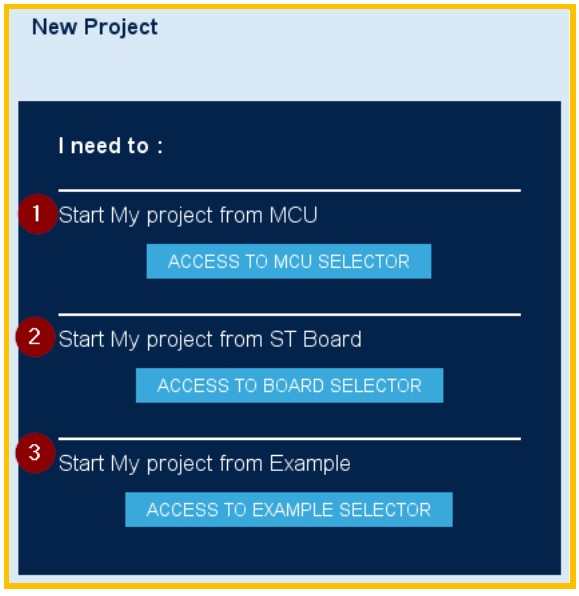
Fresh 结束后，点开 STM32WL 左侧的黑色三角，选中 STM32WL 下版本 1.0.0 的 STM32Cube MCU Package for STM32WL Series，点 Install Now



点击 Close

1.2 CubeMX 新建项目

打开 STM32CubeMX，在 New Project 下有三中选择，**请根据需求选择 New Project 的方式！**



1.2.1 Start My project from MCU

第 1 种可使用于所有的 WL 产品，只能生成和配置的用户自定义(User Defined)的 LoRaWAN/SigFox/SUBGHZ 应用，不能生成 CubeWL 中 AT_Slave/EndNode/PingPong demo

1.2.2 Start My project from BOARD

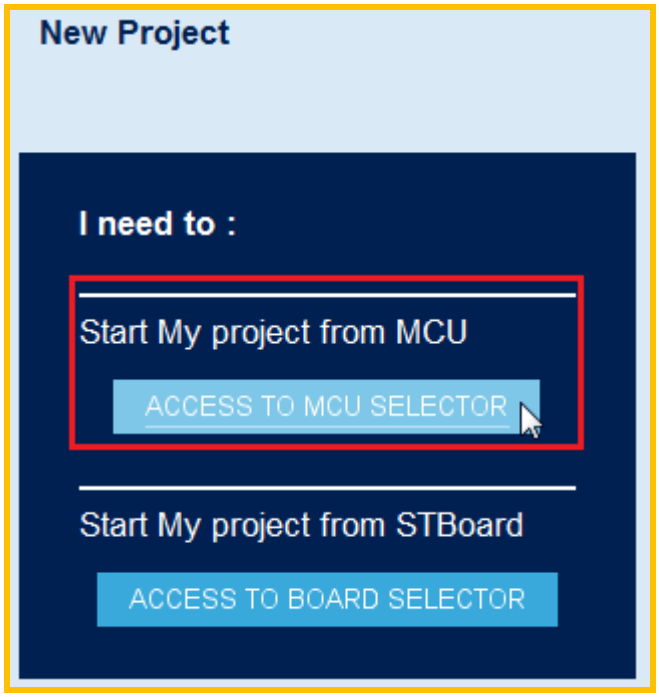
第 2 种适用于 NUCLEO-WL55JC1 和 NUCLEO-WL55JC1 两种 Demo 板，只能生成和配置的用户自定义(User Defined)的 LoRaWAN/SigFox/SUBGHZ 应用，不能生成 CubeWL 中 AT_Slave/EndNode/PingPong demo

1.2.3 Start My project from EXAMPLES

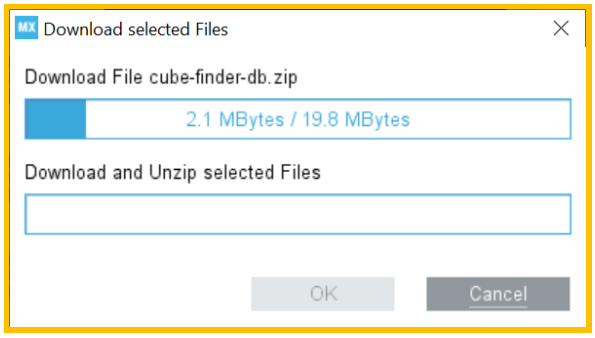
第 3 种适用于生成 CubeWL 中 NUCLEO-WL55JC1 和 NUCLEO-WL55JC1 的 Examples，可生成和配置 CubeWL 中 AT_Slave/EndNode/PingPong demo

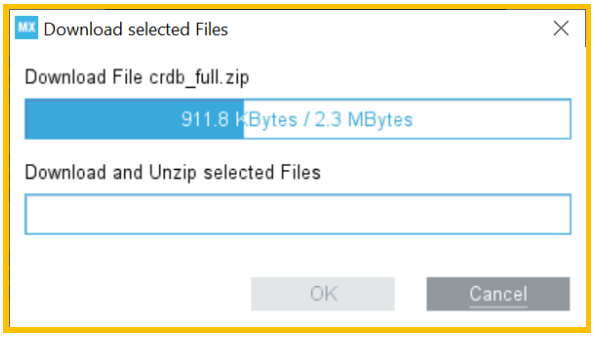
1.2.4 Start My project from MCU STM32WLEx

在 New Project 下选择，”Start My project from MCU ”下的”ACCESS TO MCU SELECTOR“

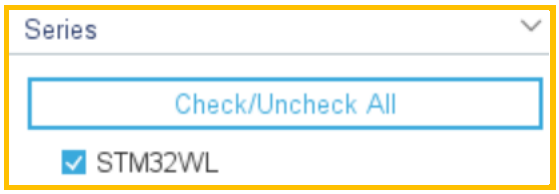


此过程中可能会更新些数据库，如

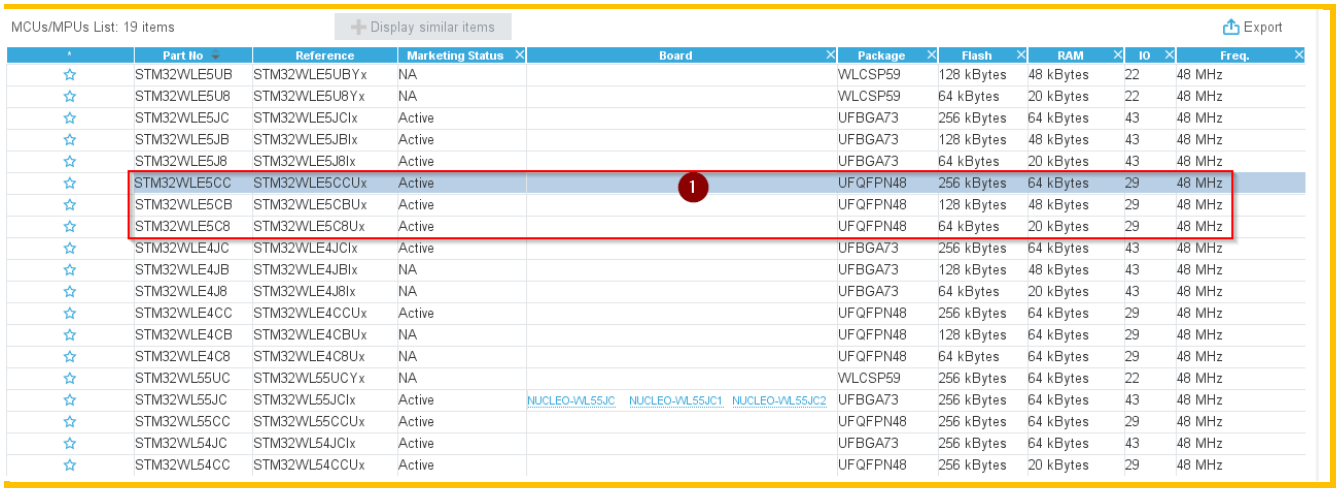




选择 Series 下的 STM32WL

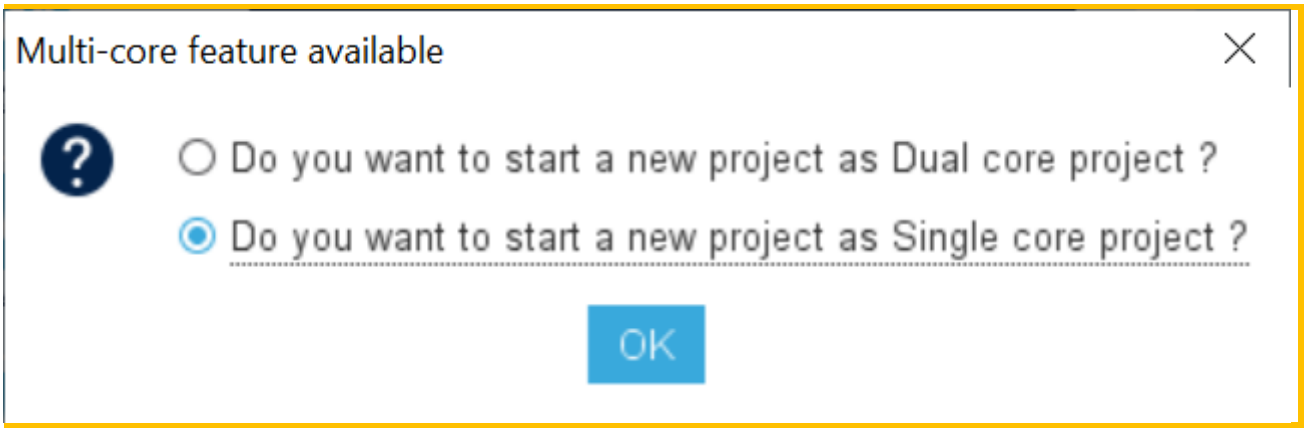


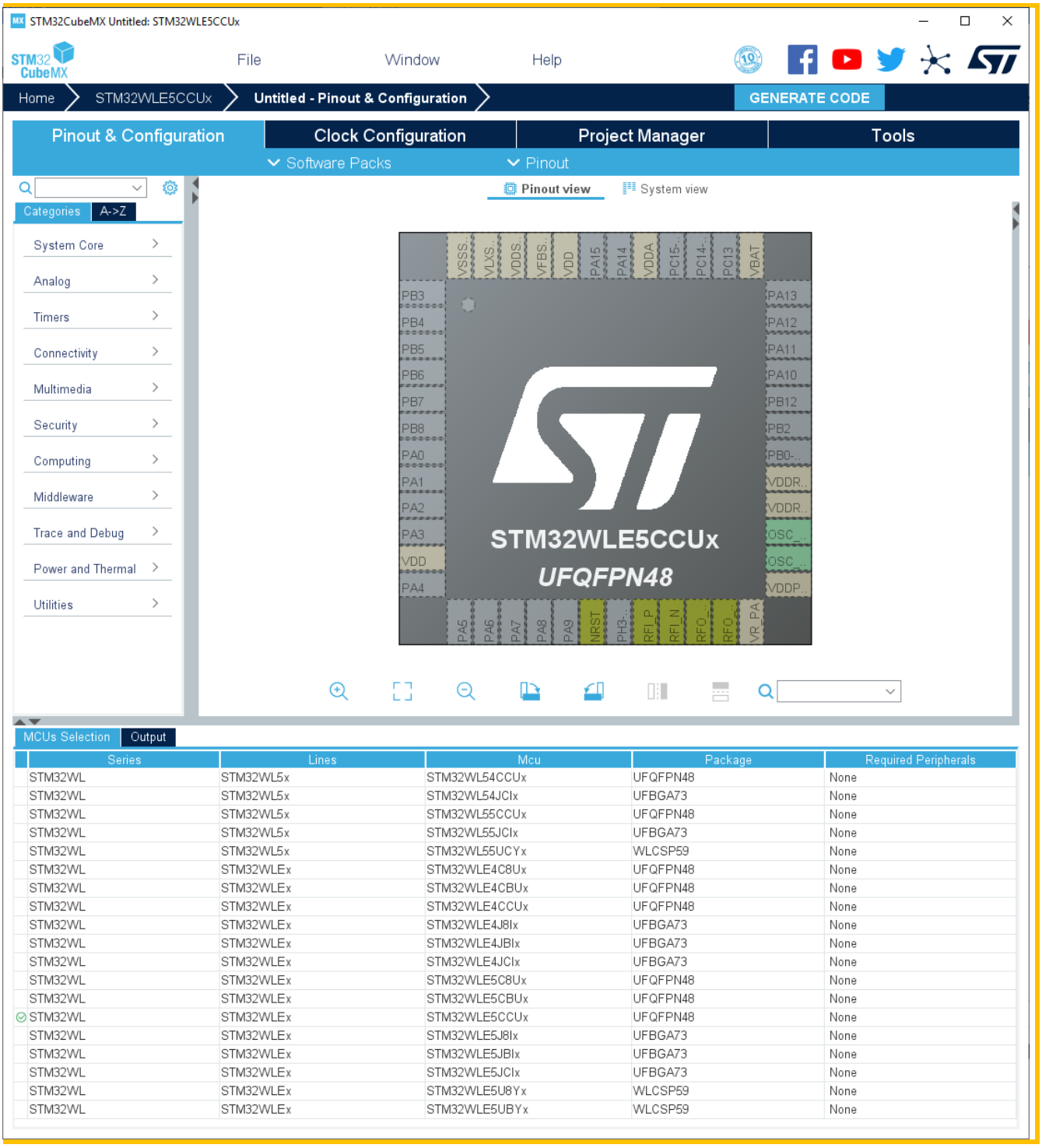
根据自己的需求选择 STM32WLE5Cx 中的任意一款



点击“Start Project”开始创建项目

选择 Single Core Project，点击 OK

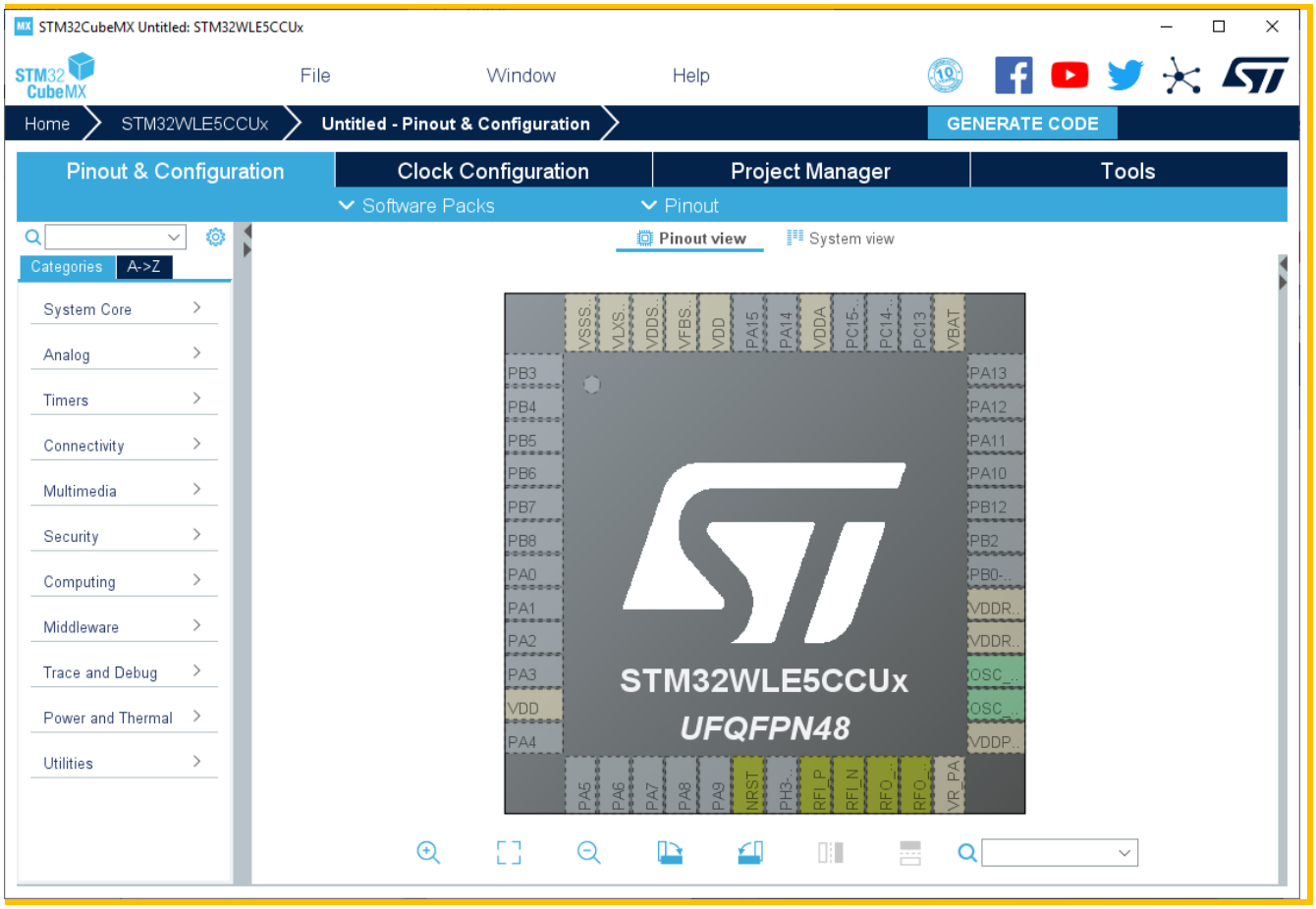




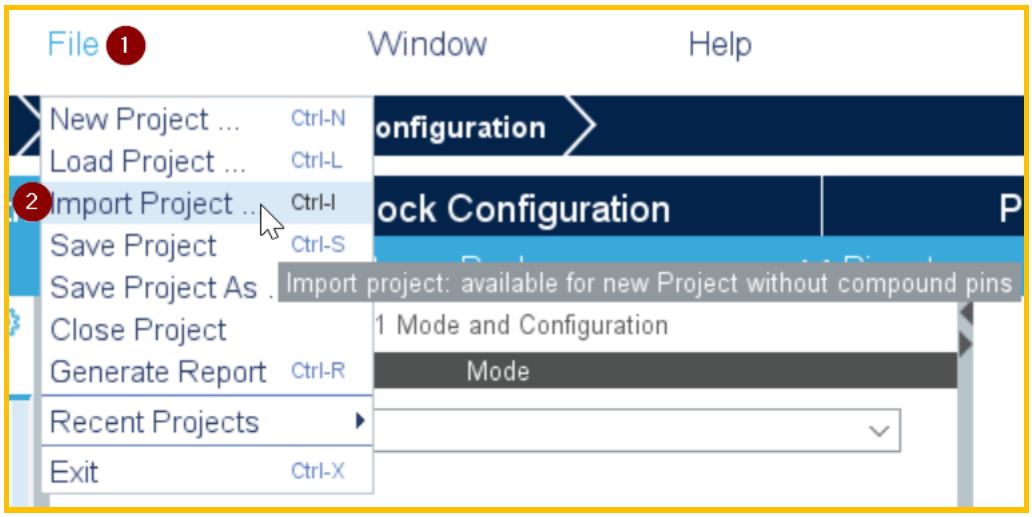
（可选）为最大化窗口，可在 Window 中取消 Output 窗口的选择

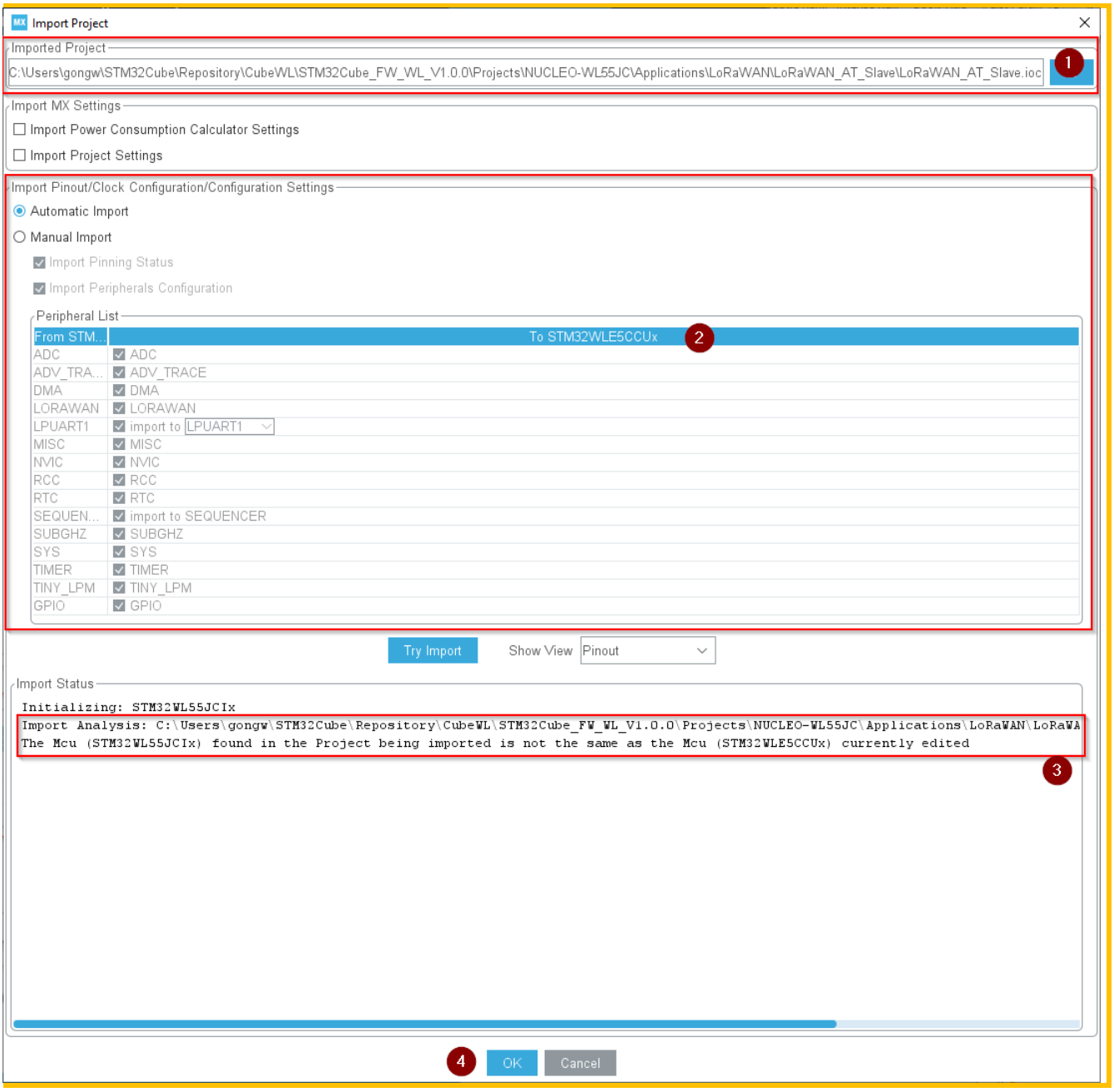
由 修改为

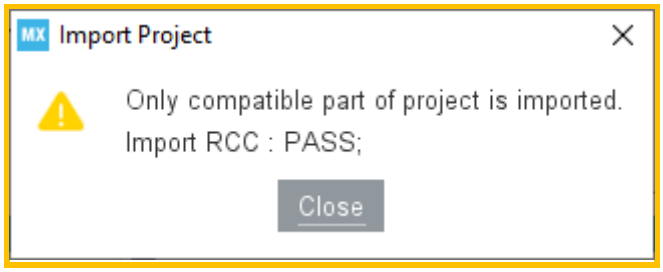
，则

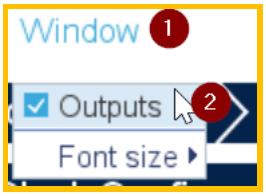


1.3 CubeMX 导入参考项目

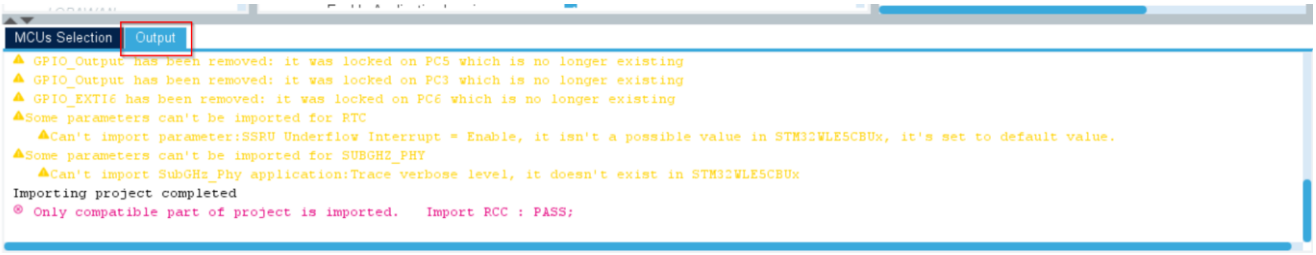








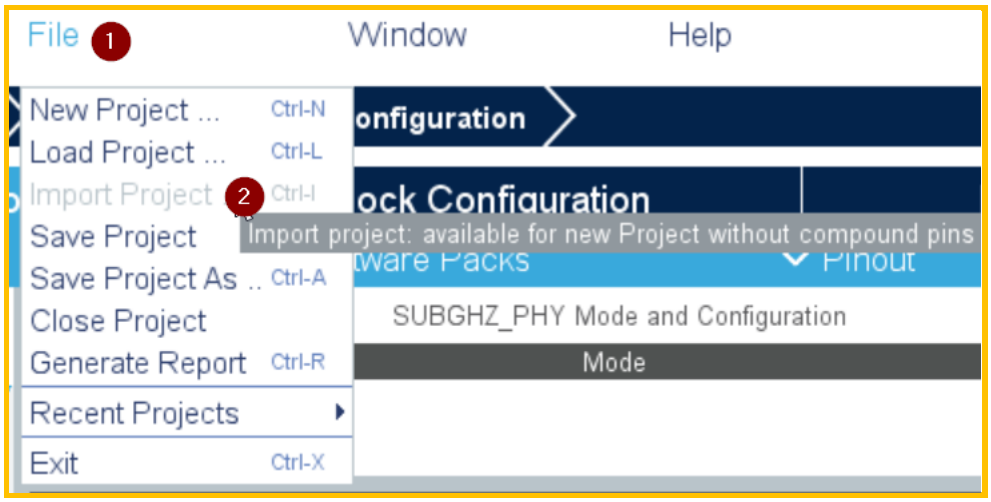
可通过 Window=>Output 切换是否查看 import 的提示 log



Creating: STM32WLE5CCUx
Initializing: STM32WLE5CCUx
Import Analysis:
C:\Users\gongw\STM32Cube\Repository\CubeWL\STM32Cube_FW_WL_V1.0.0\Projects\NUCLEO-WL55JC\Applications\LoRaWAN\LoRaWAN_AT_Slave\LoRaWAN_AT_Slave.ioc project
The Mcu (STM32WL55JCIx) found in the Project being imported is not the same as the Mcu (STM32WLE5CCUx) currently edited

Import :
import RTC partly failed
error: Alarm A:Internal Alarm A conflicts with [RTC is already used for system time base]
GPIO_Output has been removed: it was locked on PB9 which is no longer existing
GPIO_Output has been removed: it was locked on PB11 which is no longer existing
GPIO_Output has been removed: it was locked on PB10 which is no longer existing
GPIO_Output has been removed: it was locked on PB15 which is no longer existing
GPIO_Output has been removed: it was locked on PB14 which is no longer existing
GPIO_Output has been removed: it was locked on PB13 which is no longer existing
GPIO_Output has been removed: it was locked on PC4 which is no longer existing
GPIO_Output has been removed: it was locked on PC5 which is no longer existing
GPIO_Output has been removed: it was locked on PC3 which is no longer existing
GPIO_EXTI6 has been removed: it was locked on PC6 which is no longer existing
Some parameters can't be imported for LORAWAN
Can't import LoRaWAN application:Trace verbose level, it doesn't exist in STM32WLE5CCUx
Some parameters can't be imported for RTC
Can't import parameter:SSRU Underflow Interrupt = Enable, it isn't a possible value in STM32WLE5CCUx, it's set to default value.
Importing project completed
Only compatible part of project is imported. Import RCC : PASS;

此时再查看 File =>Import Project 选项，变为灰色，表示新项目已被配置，不能再 import 参考项目。



1.4 CubeMX 配置项目

1.4.1 CubeMX 配置 Project Manager

选择 Project Manager

1.4.1.1 配置 Project

选择右侧 Project 并配置如下：

Project Name: WLEx_AT_Slave
Project Location :
C:\Users\gongw\STM32Cube\Repository\CubeWL\STM32Cube_FW_WL_V1.0.0\Projects\NUCLEO-WL55JC\Applications\LoRaWAN
Application Structure : Advanced
Toolchain / IDE EWARM V8
Minimum Heap Size: 0x200
Minimum Stack Size: 0x800
CubeWL 版本: STM32Cube_FW_WL_V1.0.0

注意：

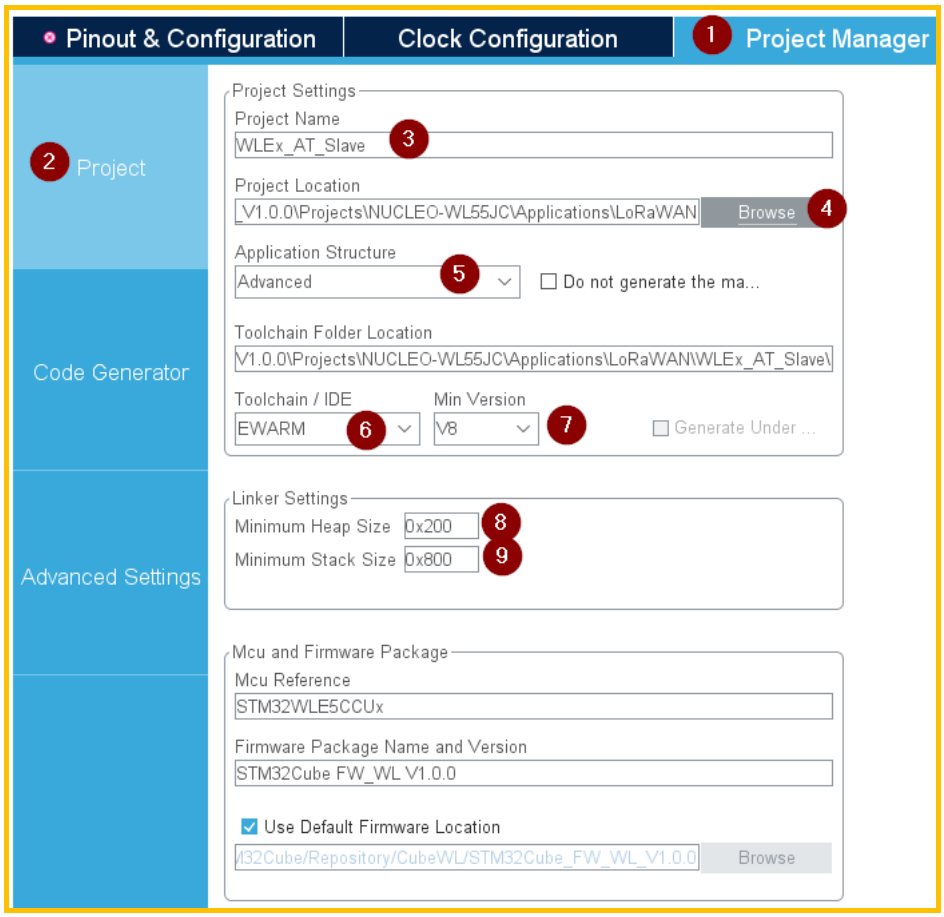
通过 Browse 选择 Project 保存的目录

- 项目名称 WLEx_AT_Slave 创建在 STM32Cube_FW_WL_V1.0.0\Projects\NUCLEO-WL55JC\Applications\LoRaWAN 下，和 LoRaWAN_End_Node 同级目录，即 C:\Users\gongw\STM32Cube\Repository\CubeWL\STM32Cube_FW_WL_V1.0.0\Projects\NUCLEO-WL55JC\Applications\LoRaWAN\WLEx_AT_Slave，以便于与 LoRaWAN_End_Node 进行比较
- 配置 Application Structure 为 Advanced 与 basic 有什么区别?

Application Structure 为 Advanced 时目录结构分类分级目录更清晰。

Application Structure 为 Basic 时，所有的*.h 都在 WLEx_AT_Slave\Inc 目录下，所有的*.c 在 WLEx_AT_Slave\Src 目录下。

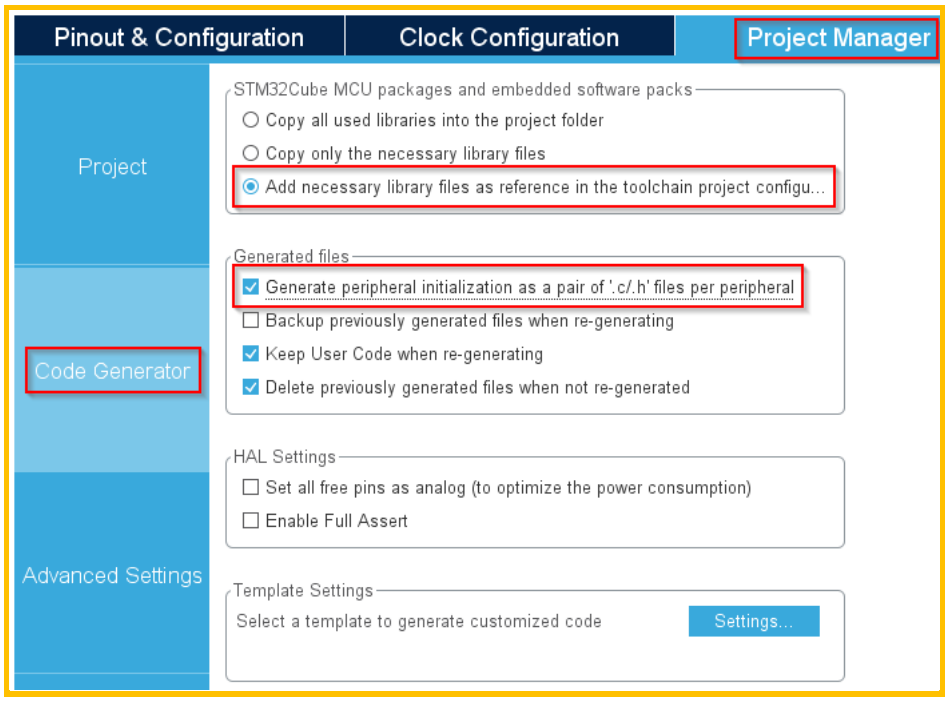
一旦生成过一次代码后目录结构就无法更改了！！！

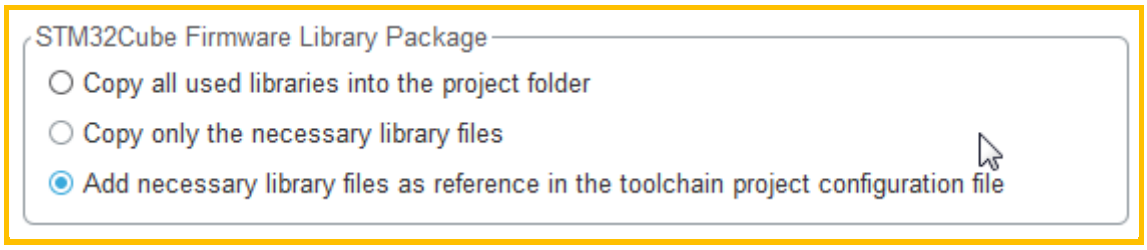


1.4.1.2 配置 Code Generator

选择右侧 Code Generator 并配置如下，选择”Add necessary library files as reference in the toolchain project configuration file “，这样将不生成 Drivers 和 Middlewares 目录。

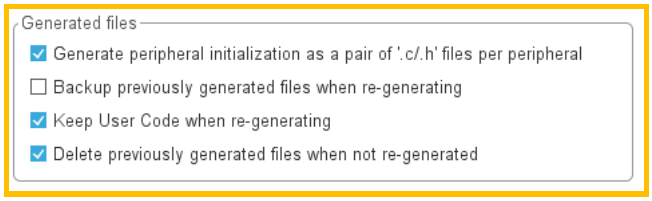
选择 Generated files 下的 “Generate peripheral initialization as a pair of ‘.c/.h’ per peripheral ”





注意:

- Copy all used library Package (如复制 Drivers\STM32WLxx_HAL_Driver 下的所有驱动)
  WLEx_AT_Slave 复制到其他目录下，还能工作
- Copy only the necessary library files(如仅复制 Drivers\STM32WLxx_HAL_Driver 下使用到的驱动)
  WLEx_AT_Slave 复制到其他目录下，还能工作
- Add necessary library files as reference in the toolchain project configuration file(不生成 Drivers 和 Middlewares 目录，仅参考 Drivers\STM32WLxx_HAL_Driver 下的所有驱动)
  WLEx_AT_Slave 复制到其他目录下，不能工作
  参考 UM1718 **4.9 Project Manager view**

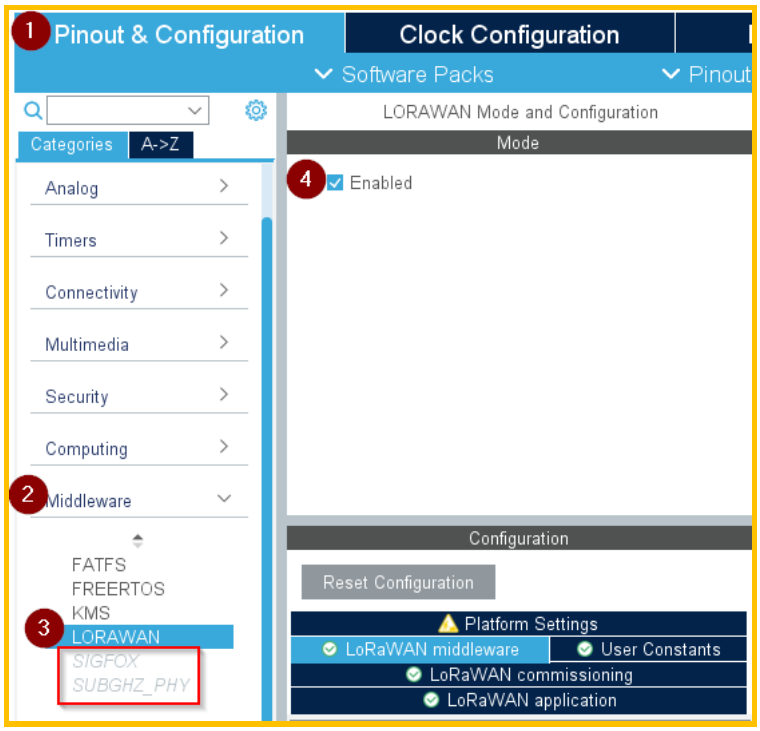


2 配置外设

2.1 使能外设

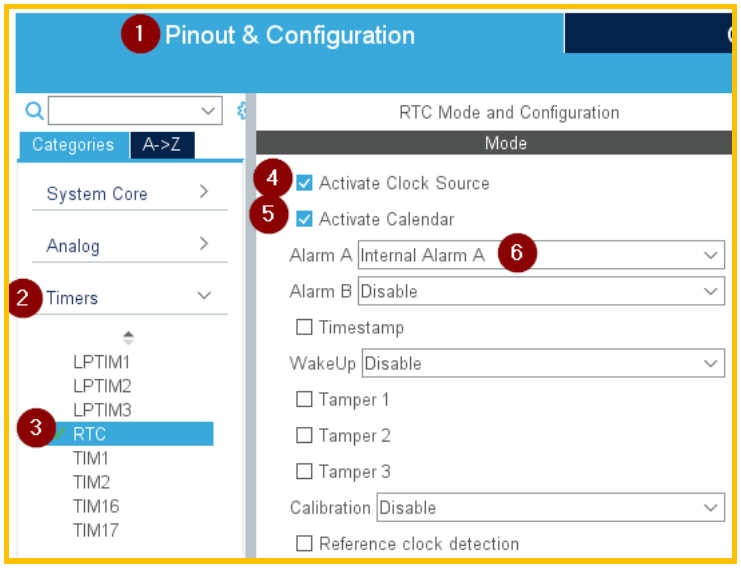
2.1.1 Middleware=>LORAWAN

Middleware=>LORAWAN=>Mode 中勾选 Enabled，LORAWAN 功能就使能了，同时 SIGFOX 和 SUBGHZ_PHY 就失能了，默认界面如下，后续再继续配置 LORAWAN 应用。



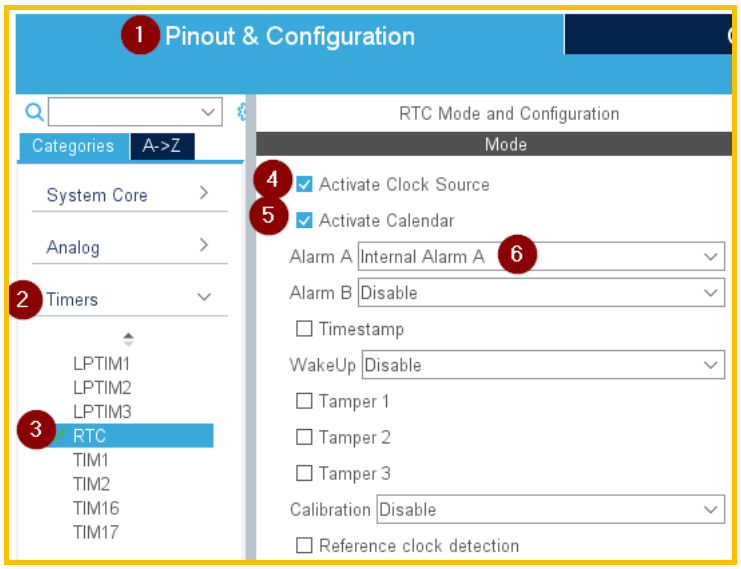
2.1.2 Timers=>RTC 使能

Timers=>RTC=>勾选 Activate Clock Source，并配置 Alarm A 为 Internal Alarm A



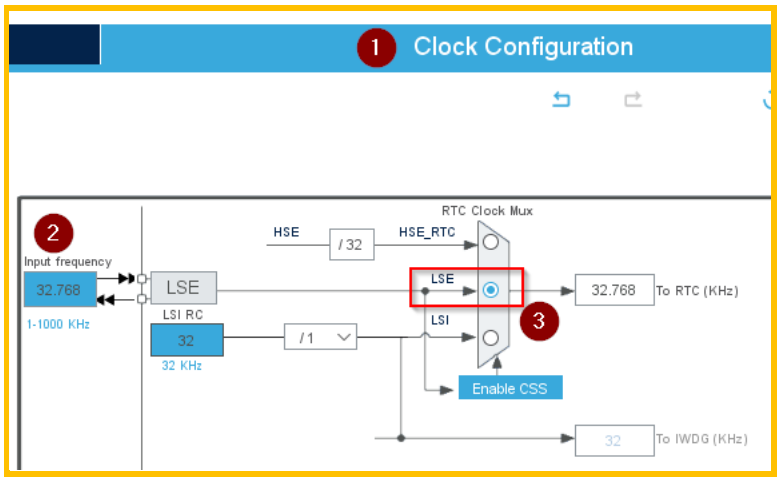
2.2 RTC

配置 Alarm A 为 Internal Alarm A

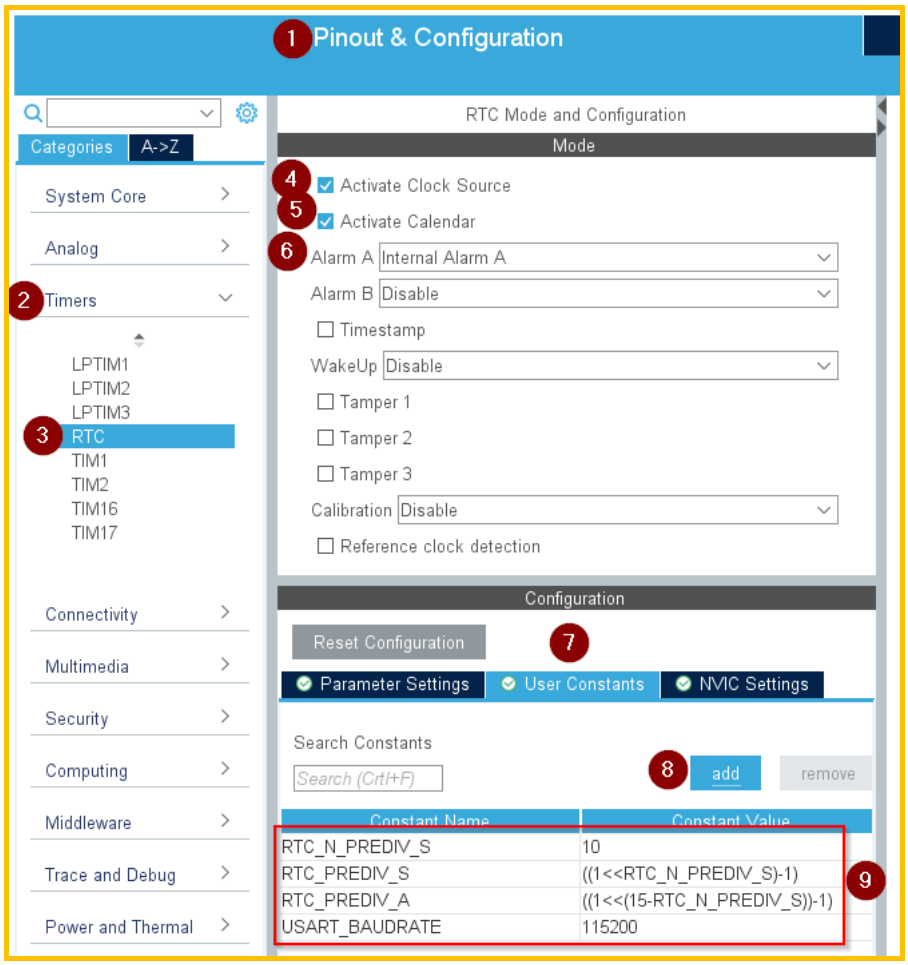


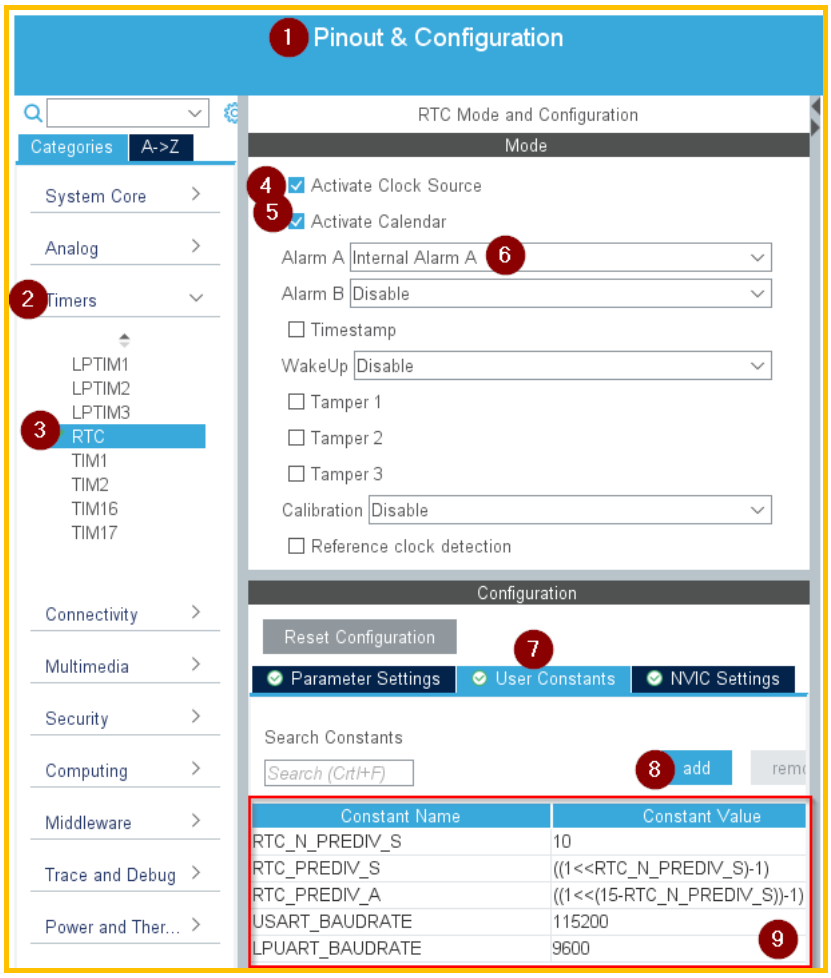
Clock Configuration => RTC Clock Mux=>LSE

在 Clock Configuration 中检查 RTC Clock Mux 是否配置为 LSE



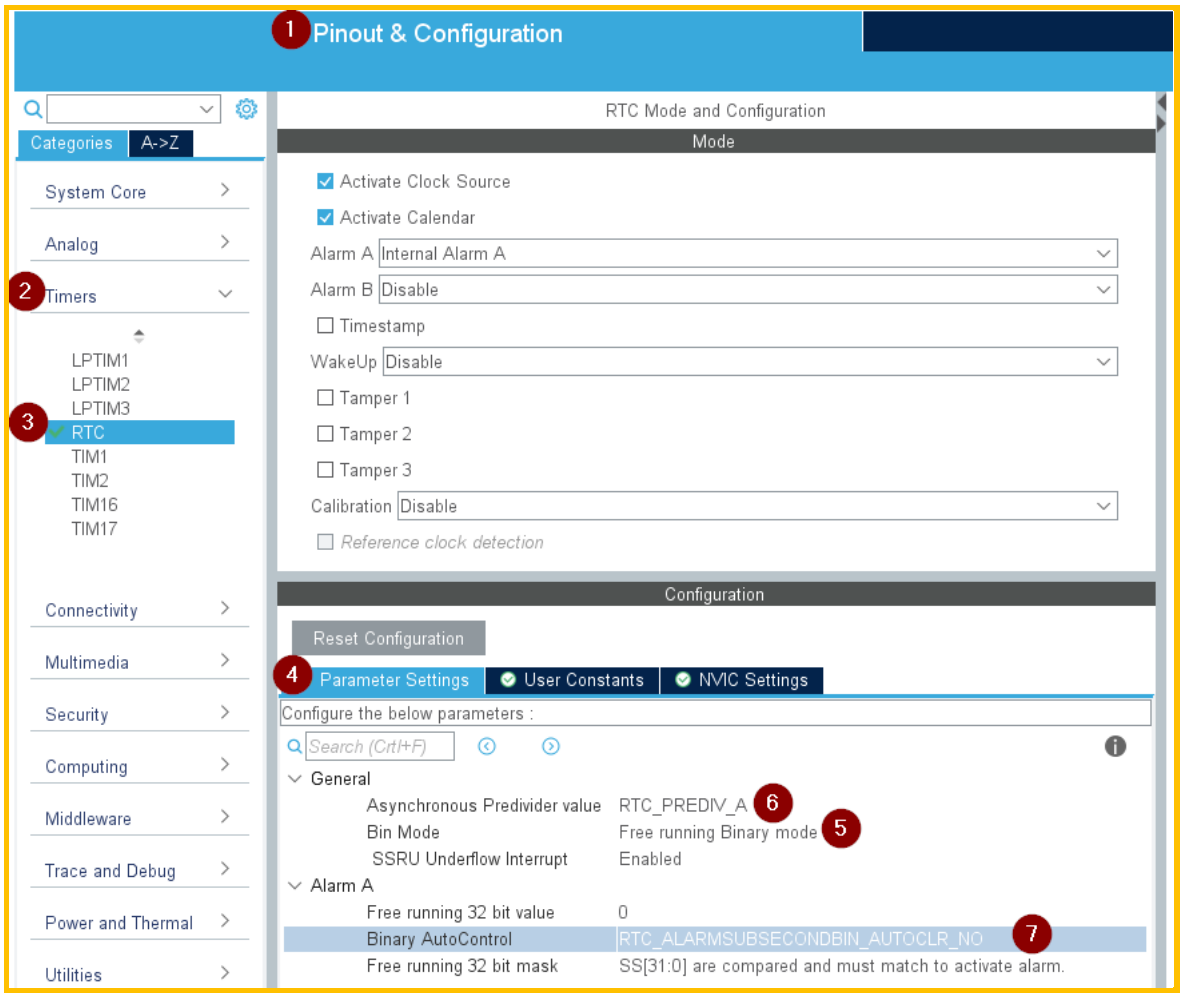
Timer =>RTC=>Users Constants


1 Pinout & Configuration
Categories
A->Z
System Core
Analog
2 Timers
LPTIM1
LPTIM2
LPTIM3
3 RTC
TIM1
TIM2
TIM16
TIM17
Connectivity
Multimedia
Security
Computing
Middleware
Trace and Debug
Power and Thermal
RTC Mode and Configuration
Mode
4 Activate Clock Source
5 Activate Calendar
6 Alarm A Internal Alarm A
Alarm B Disable
Timestamp
WakeUp Disable
Tamper 1
Tamper 2
Tamper 3
Calibration Disable
Reference clock detection
Configuration
Reset Configuration
7
Parameter Settings
User Constants
NVIC Settings
Search Constants
Search (Crtl+F)
8 add
remove
Constant Name
Constant Value
RTC_N_PREDIV_S
10
RTC_PREDIV_S
((1<<RTC_N_PREDIV_S)-1)
RTC_PREDIV_A
((1<<(15-RTC_N_PREDIV_S))-1)
USART_BAUDRATE
115200
9

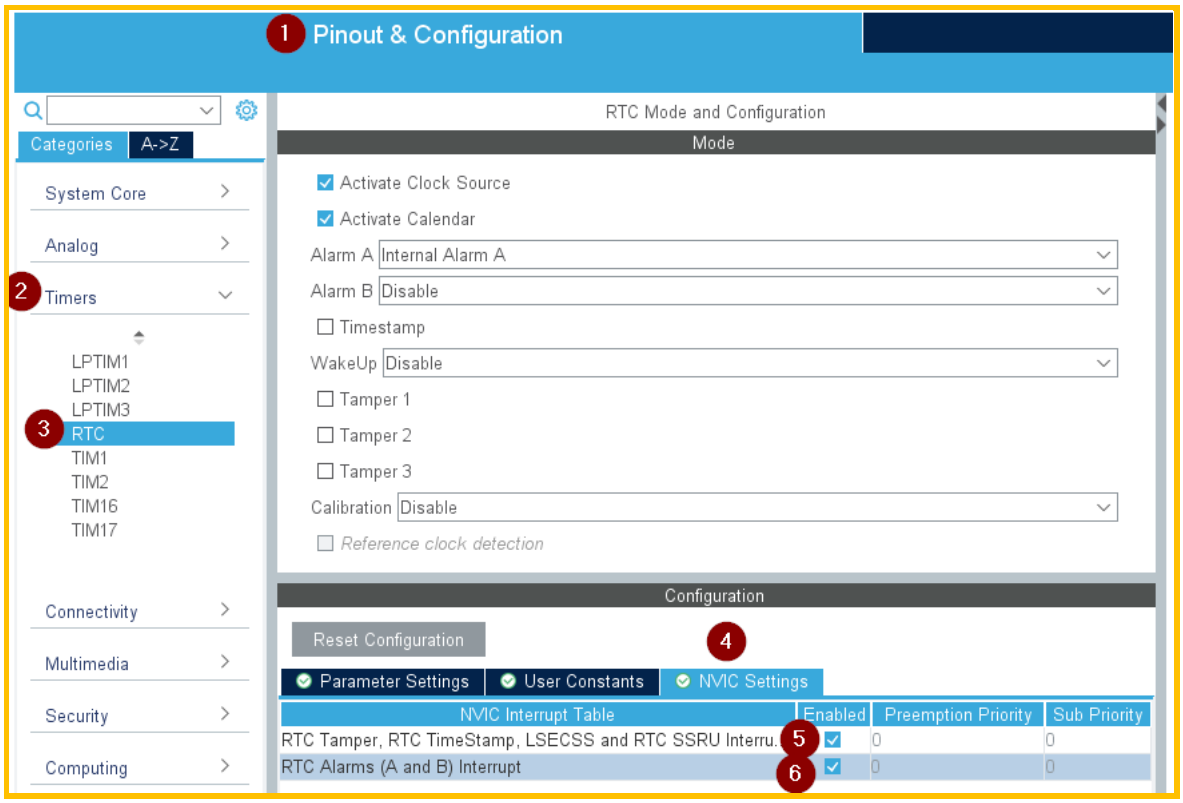


RTC_N_PREDIV_S	10
RTC_PREDIV_S	((1<<RTC_N_PREDIV_S)-1)
RTC_PREDIV_A	((1<<(15-RTC_N_PREDIV_S))-1)
USART_BAUDRATE	115200
LPUART_BAUDRATE	9600

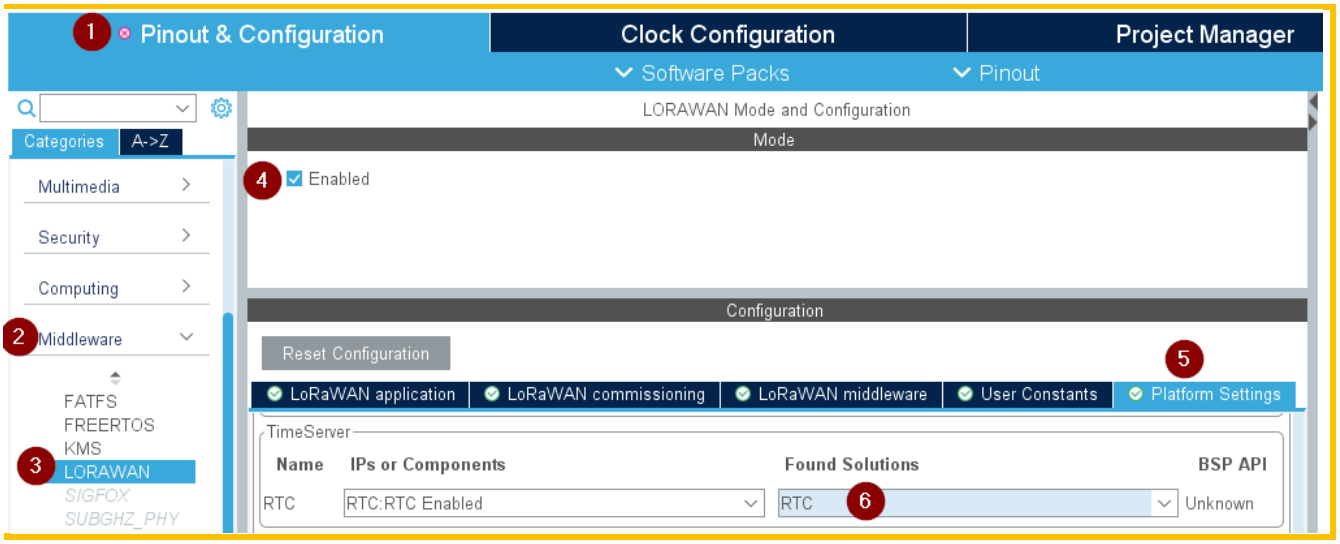
Timer =>RTC=>Parameter Settings



Timer =>RTC=>NVIC Settings

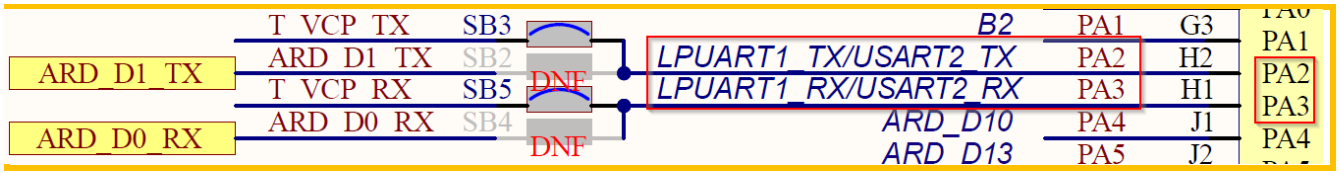


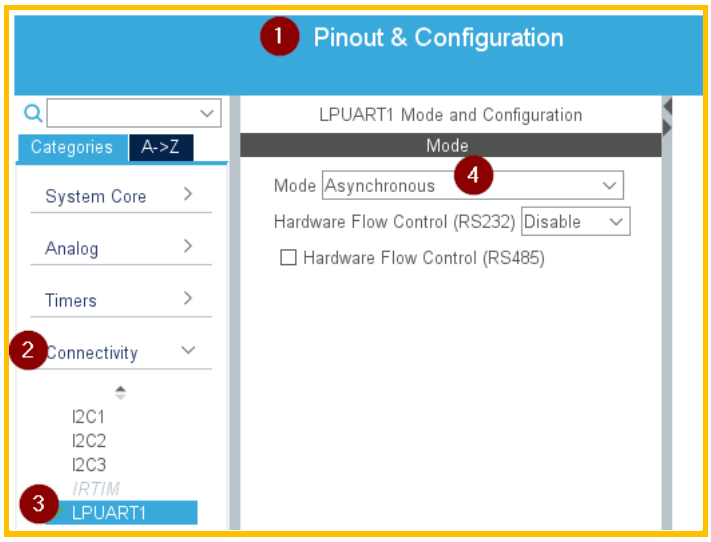
Middleware => LORAWAN => Platform Settings => Timer Server => RTC

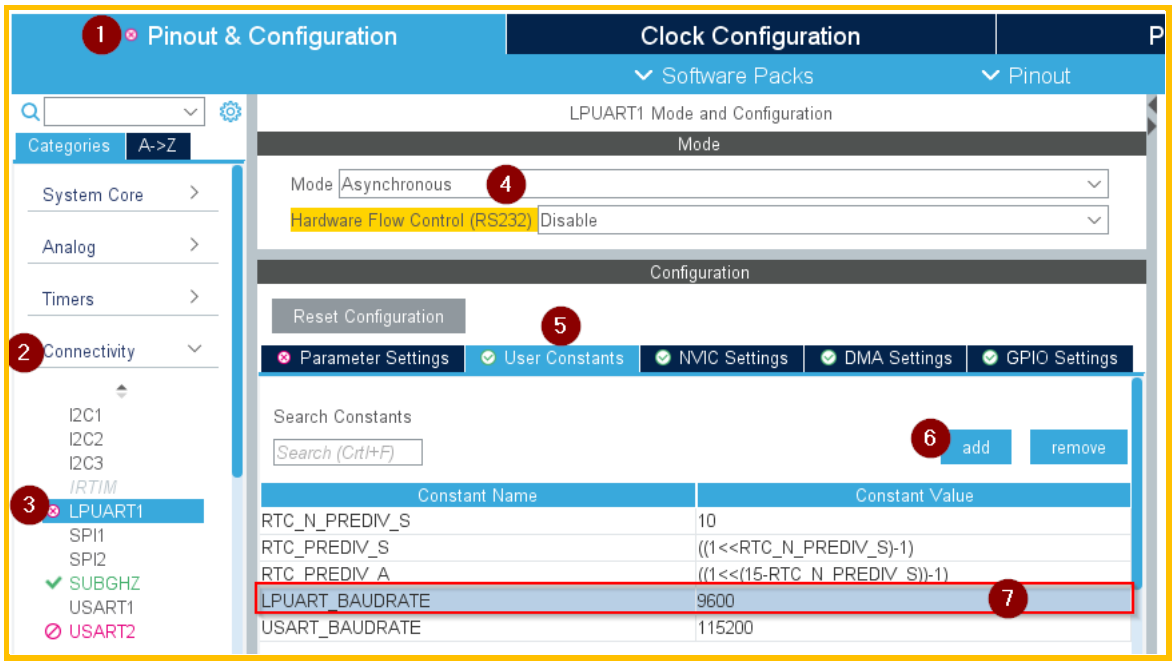


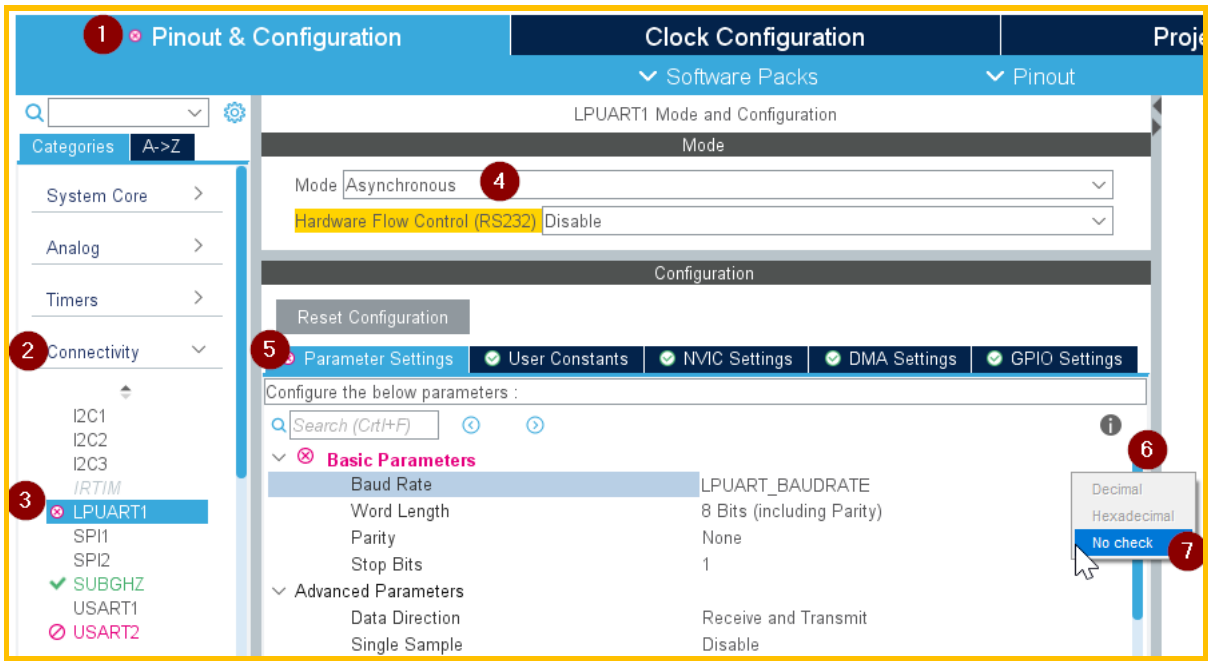
2.3 DebugLine

Connectivity=>LPUART1 (AT_Slave)

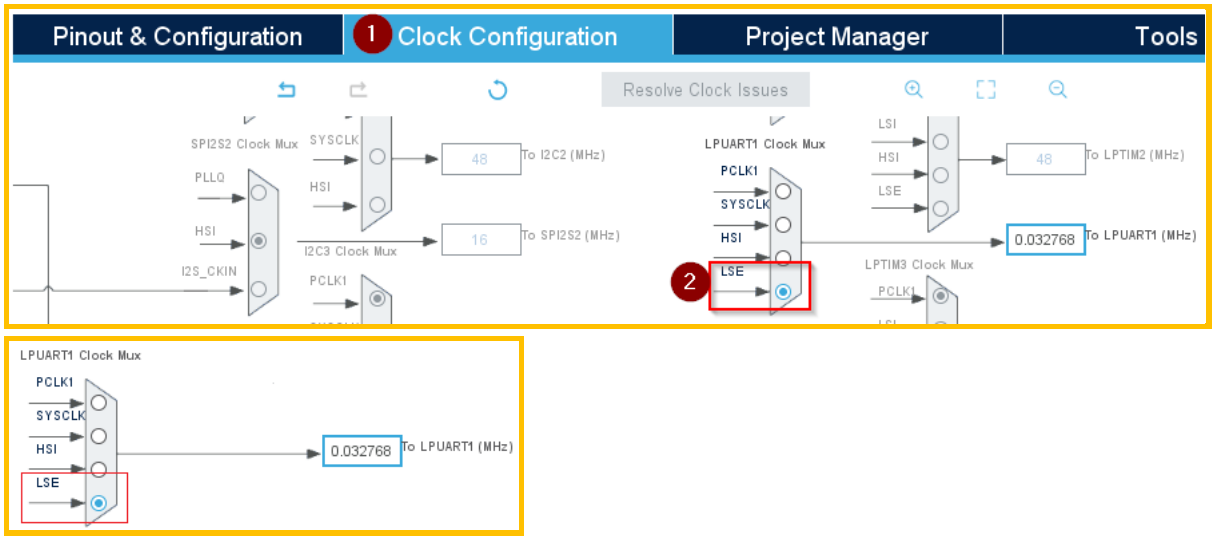



1 Pinout & Configuration
Categories
A->Z
System Core
Analog
Timers
2 Connectivity
I2C1
I2C2
I2C3
IRTIM
3 LPUART1
LPUART1 Mode and Configuration
Mode
Mode Asynchronous
4
Hardware Flow Control (RS232) Disable
Hardware Flow Control (RS485)

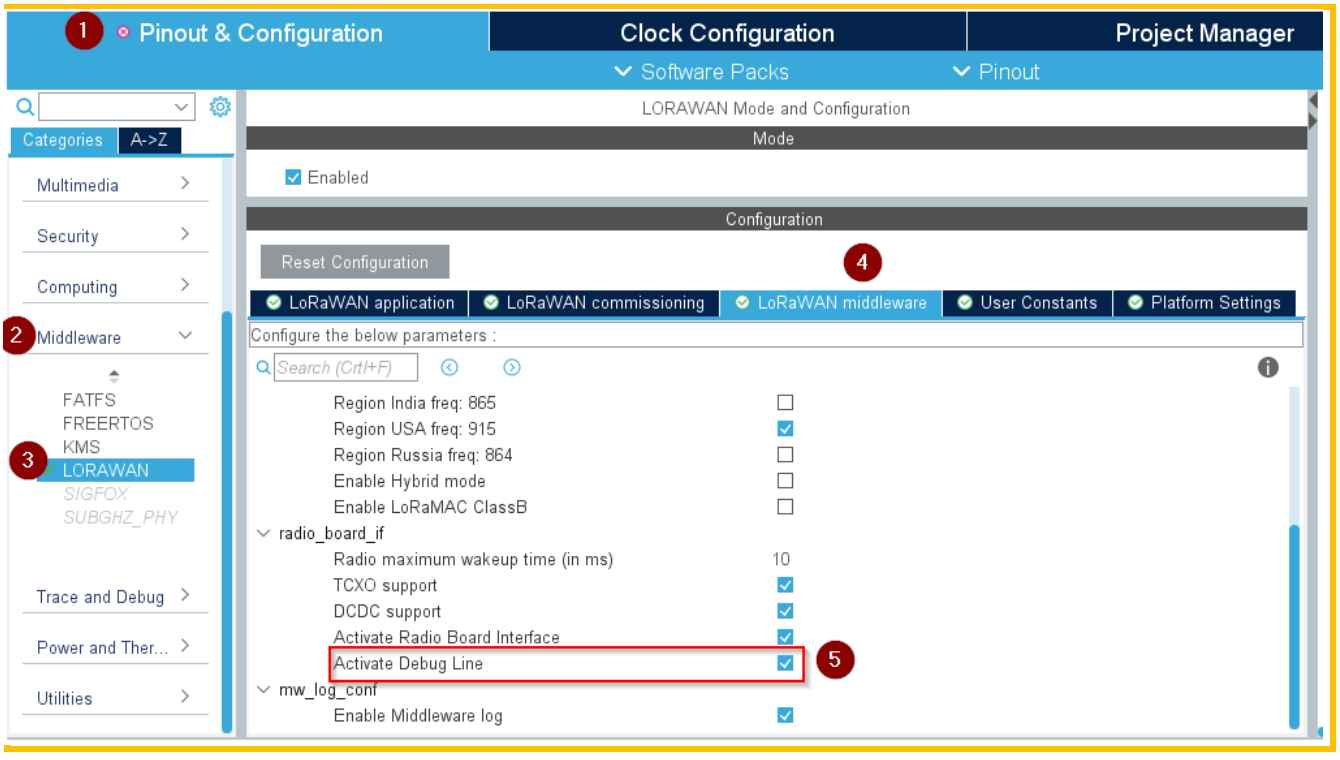

1 Pinout & Configuration
Clock Configuration
Software Packs
Pinout
LPUART1 Mode and Configuration
Mode
Mode Asynchronous
4
Hardware Flow Control (RS232) Disable
Configuration
Reset Configuration
5
Parameter Settings
User Constants
NVIC Settings
DMA Settings
GPIO Settings
Search Constants
Search (Crtl+F)
6
add
remove
Constant Name
Constant Value
RTC_N_PREDIV_S
10
RTC_PREDIV_S
((1<<RTC_N_PREDIV_S)-1)
RTC_PREDIV_A
((1<<(15-RTC_N_PREDIV_S))-1)
LPUART_BAUDRATE
9600
7
USART_BAUDRATE
115200
2 Connectivity
3 LPUART1
SPI1
SPI2
SUBGHZ
USART1
USART2


1 Pinout & Configuration
Clock Configuration
Software Packs
Pinout
LPUART1 Mode and Configuration
Mode
Mode Asynchronous
4
Hardware Flow Control (RS232) Disable
Configuration
Reset Configuration
5
Parameter Settings
User Constants
NVIC Settings
DMA Settings
GPIO Settings
Configure the below parameters :
Search (Crtl+F)
Basic Parameters
Baud Rate
LPUART_BAUDRATE
Word Length
8 Bits (including Parity)
Parity
None
Stop Bits
1
Advanced Parameters
Data Direction
Receive and Transmit
Single Sample
Disable
6
Decimal
Hexadecimal
No check
7
2 Connectivity
3 LPUART1
SPI1
SPI2
SUBGHZ
USART1
USART2

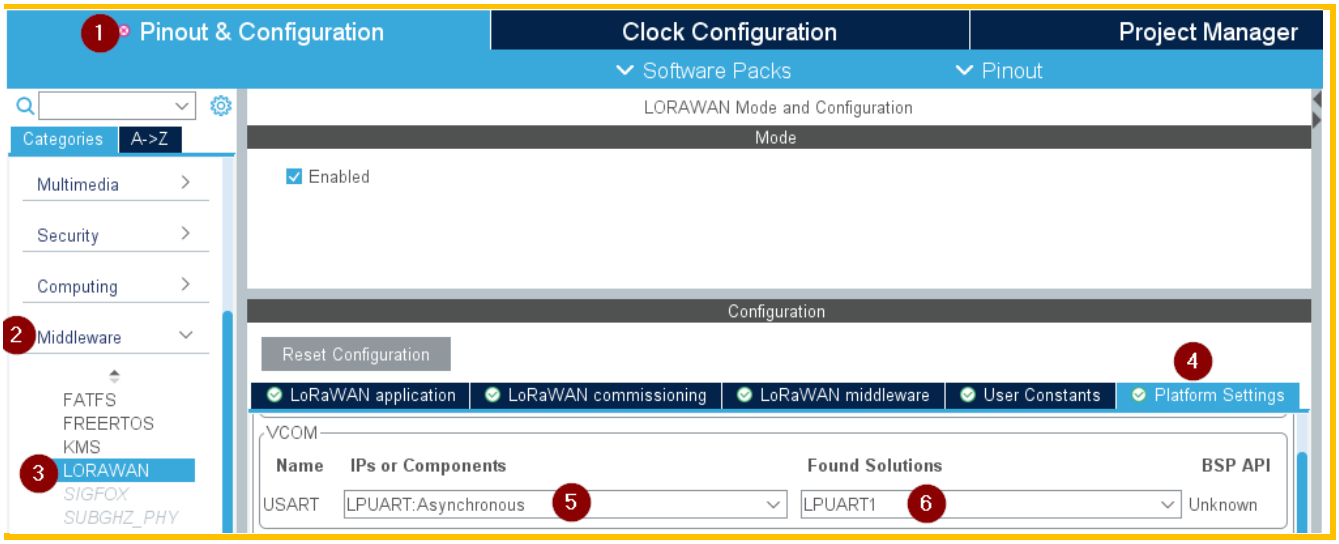
Clock Configuration => LPUART1 Clock Mux => LSE



Middleware => LORAWAN => LoRaWAN middleware => radio_board_if =>Activate Debug Line



Platform Settings => VCOM => LPUART1

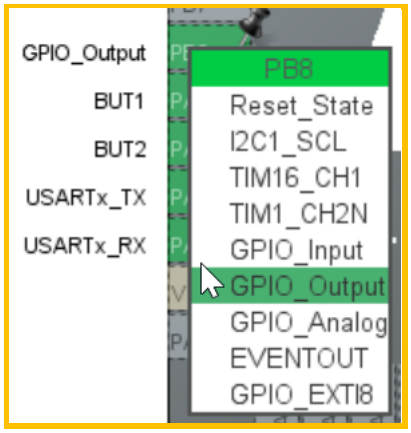


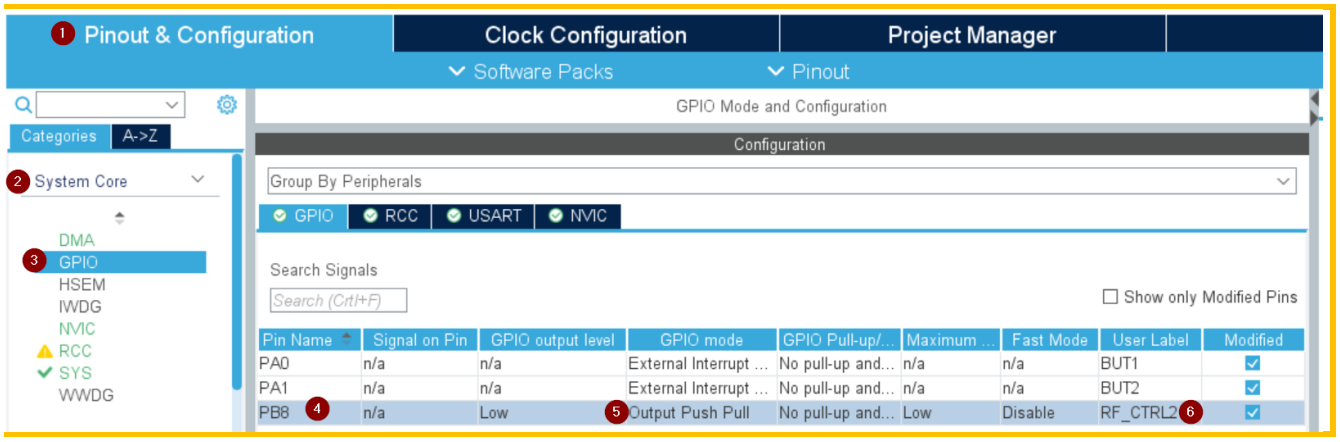
2.4 System Core => GPIO (RF SW CTRL / BUTTON / LED)

2.4.1 RF_CTRL

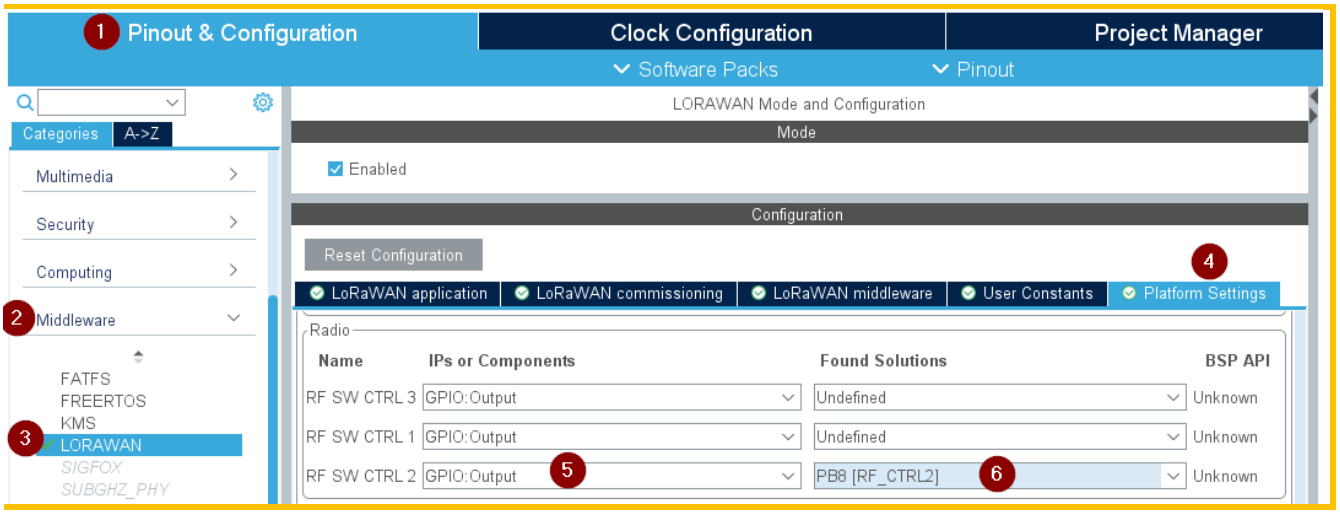
System Core => GPIO => PA8 (customizable)

RF front-end configuration	FE_CTRL1	FE_CTRL2	FE_CTRL3
Transmit high output power	Low	High	High
Transmit low output power	High	High	High
Receive	High	Low	High



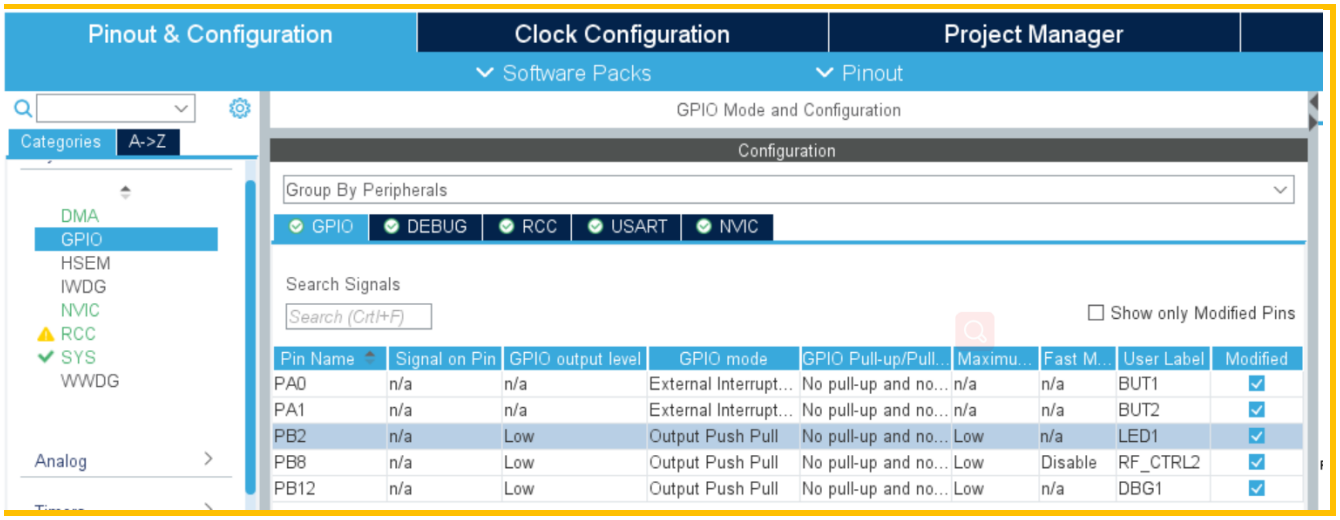


Middleware=>LORAWAN=>Platform Settings => Radio => RF SW CTRL

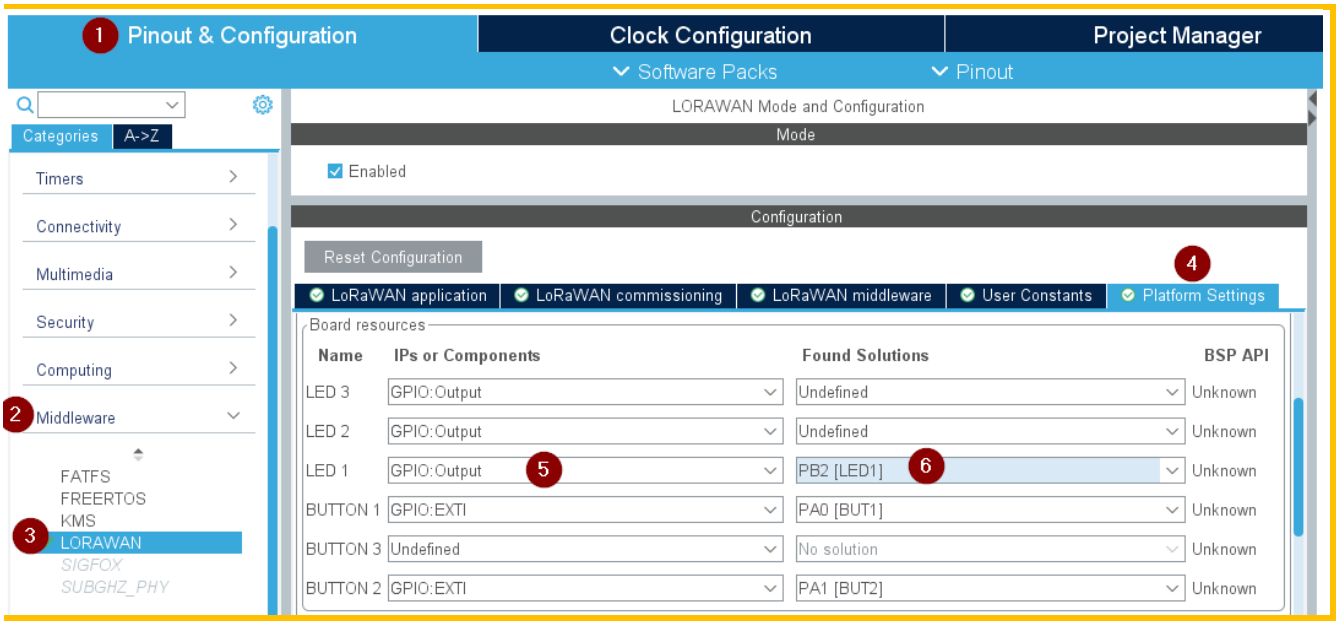


2.4.2 LED

System Core => GPIO => PB2 (customizable)



Middleware=>LORAWAN=>Platform Settings => Board resources => LED



3 STM32WL Templates Patches for CubeMX

STM32WL Templates 在如下目录，如需要可自行定制自己的 Templates

```
C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeMX\db\templates\stm32wl
```

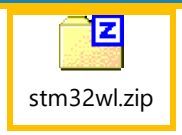
针对不同的客户定制化的板子，硬件外设配置不一致，客户可根据需要定制化自己的外设配置。

从 NUCLEO-WL55JC(BGA73) 移植到 STM32WLExCxUx(QFN48)定制化的板子上，射频开关(RF SW CTRL)，按键(BUTTON), LED 的配置会不一样，可能导致编译错误，如想生成的代码不会产生编译错误，需要编辑一下 STM32WL 的 template。

3.1 CubeMX template patches for stm32wl

请参考

```
C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeMX\db\templates\stm32wl
```



其中包含

radio_board_if_c.ftl for RF SW CTRL

radio_conf_h.ftl for DBG_GPIO_RADIO_TX

board_resources_c.ftl & board_resources_h.ftl for BUTTON

SubGHz_Phy_subghz_phy_app_c.ftl for LED

LoRaWAN_lora_app_c.ftl for LED

3.2 CubeMX Updates from v6.1.1 to v6.2.0

main.c

void SystemClock_Config(void)
{
RCC_OscInitTypeDef RCC_OscInitStruct = {0};
RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
 	
/** Configure LSE Drive Capability
*/
 	
__HAL_RCC_LSEDRIVE_CONFIG(RCC_LSEDRIVE_LOW);
/** Configure the main internal regulator output voltage
	void SystemClock_Config(void)
	{
	RCC_OscInitTypeDef RCC_OscInitStruct = {0};
	RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
	
	/** Configure LSE Drive Capability
	*/
	HAL_PWR_EnableBkUpAccess();
	
	__HAL_RCC_LSEDRIVE_CONFIG(RCC_LSEDRIVE_LOW);

/** * @brief Macro to configure the External Low Speed oscillator (LSE) drive capability. * @note As the LSE is in the Backup domain and write access is denied to * this domain after reset, you have to enable write access using * HAL_PWR_EnableBkUpAccess() function before to configure the LSE * (to be done once after reset). * @param __LSEDRIVE__ specifies the new state of the LSE drive capability. * This parameter can be one of the following values: * @arg @ref RCC_LSEDRIVE_LOW LSE oscillator low drive capability. * @arg @ref RCC_LSEDRIVE_MEDIUMLOW LSE oscillator medium low drive capability. * @arg @ref RCC_LSEDRIVE_MEDIUMHIGH LSE oscillator medium high drive capability. * @arg @ref RCC_LSEDRIVE_HIGH LSE oscillator high drive capability. * @retval None */ #define __HAL_RCC_LSEDRIVE_CONFIG(__LSEDRIVE__) LL_RCC_LSE_SetDriveCapability(__LSEDRIVE__)	

rtc.c

CubeMX v6.1.1	CubeMX v6.2.0
if (HAL_RTC_SetAlarm_IT(&hrtc, &sAlarm, RTC_FORMAT_BCD) != HAL_OK)	if (HAL_RTC_SetAlarm_IT(&hrtc, &sAlarm, 0) != HAL_OK)
#define RTC_FORMAT_BIN 0x00000000u /* This parameter will trigger a SW conversion to fit with the native BCD format of the HW Calendar. It should not be confused with the Binary mode @ref RTCEx_Binary_Mode. */ #define RTC_FORMAT_BCD 0x00000001u /* Native format of the HW Calendar.	

It should not be confused with the Binary mode @ref RTCEx_Binary_Mode. */
Ticket 96721 - [Diff Examples] WL RTC: RTC_FORMAT_BCD removed from HAL_RTC_SetAlarm_IT call **Ticket 100003** - [Diff Examples] WL RTC: RTC_FORMAT_BCD removed from HAL_RTC_SetAlarm_IT call **Ticket 90404** - [HAL_RTC_SetAlarm & HAL_RTC_SetAlarm_IT] Unnecessary assertion on parameter when Alarm Masks defined

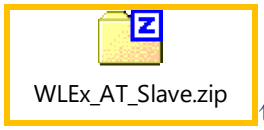
usart.c

CubeMX v6.1.1	CubeMX v6.2.0

if (HAL_DMA_ConfigChannelAttributes(&hdma_lpuart1_tx, DMA_CHANNEL_NPRIV) != HAL_OK)
{
Error_Handler();
}
	删除了

4 验证

4.0 参考工程

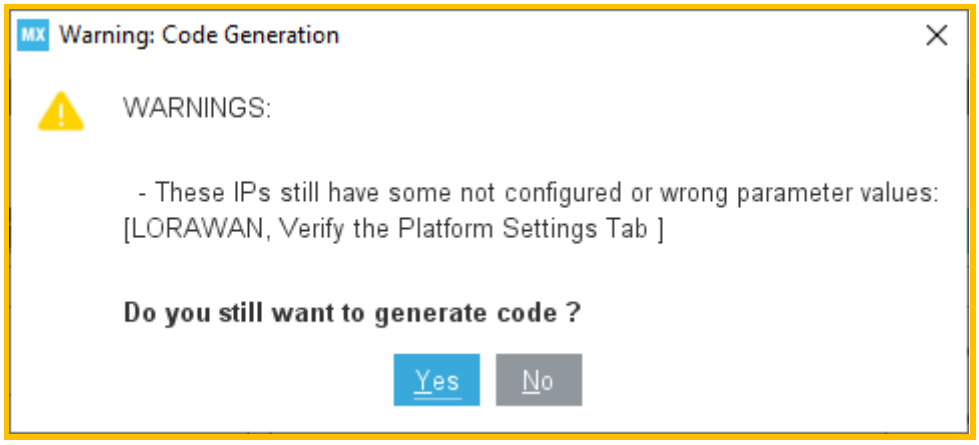


包含 WLEx_AT_Slave.ioc

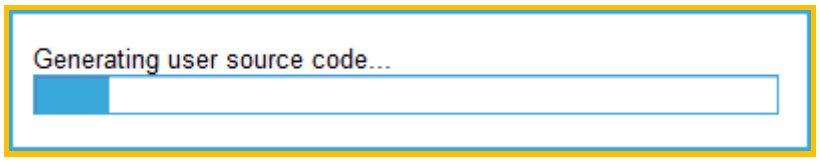
4.1 生成代码 GENERATE CODE

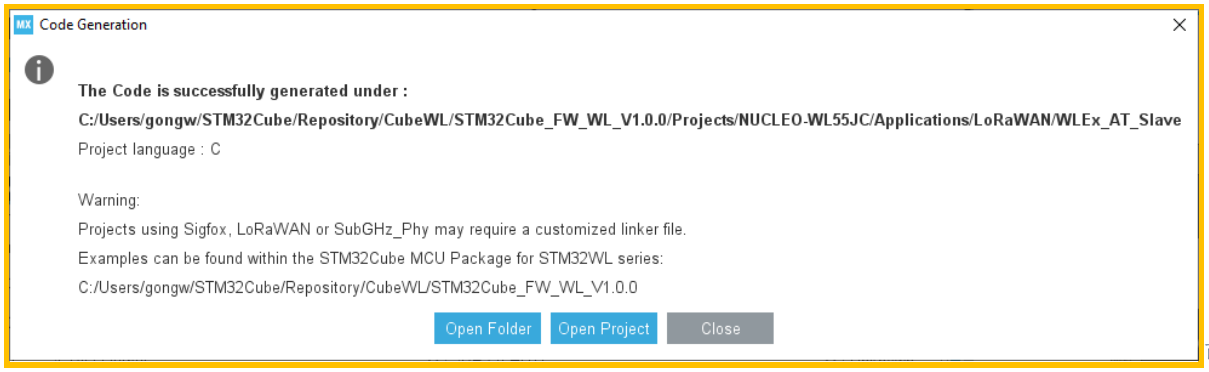
此时，可以先选择 GENERATE CODE 

生成 Project，(如需，请自己选择项目保存的目录)



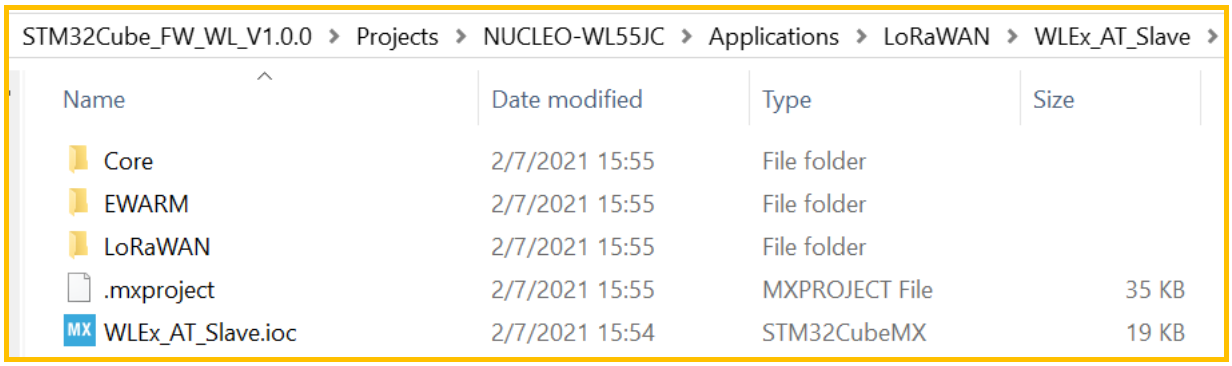
选择 Yes





可选择 Open Project 使用 IDE 直接打开 Project。

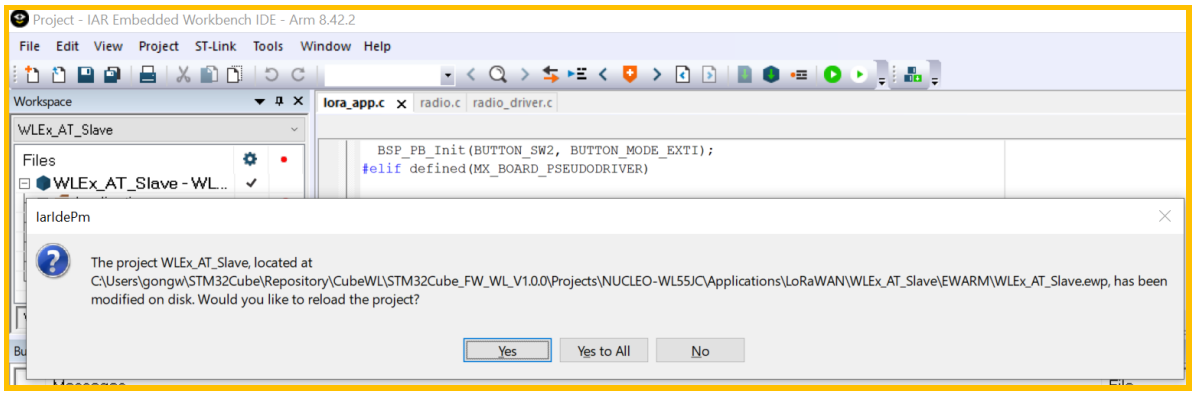
或选择 Open Folder 后，还需再选择 EWARM\Project.eww 打开 Project



4.2 编译并下载

用 IAR 打开 STM32Cube_FW_WL_V1.0.0\Projects\NUCLEO-WL55JC\Applications\LoRaWAN\WLEx_AT_Slave\EWARM\Project.eww

如果 IAR 之前已经打开了 WLEx_AT_Slave，在每次重新生成代码时都如有如下提示，选择“Yes to All”



按 F7 或选择 Project=>Make(F7)来编译工程，如果编译无错误，选择 Project=>Download=>Download Active Application 下载 WLEx_AT_Slave 应用

版本历史

日期	版本	变更
2022 年 04 月 11 日	1.0	首版发布