从 STM32WL55JC 到 STM32WLE5CC 的 LoRaWAN_FUOTA 移植

关键字：LoRaWAN FUOTA， STM32WL

1. 需求分析

最近有些客户需要在 STM32WL55 实现 LoRaWAN FUOTA 功能，LoRaWAN_FUOTA 默认是基于双核 STM32WL55JC， 由于成本考虑客户需要用 STM32WLE5CC，客户咨询如何移植?

2. 工程分析

以下移植是基于 IAR IDE 和 STM32Cube_FW_WL_V1.3.0，复制 STM32Cube_FW_WL_V1.3.0\Projects\NUCLEO-WL55JC\Applications 下的 LoRaWAN_FUOTA 为 LoRaWAN_FUOTA_WLE5CC。

先分析基于 STM32WL55JC 的 LoRaWAN_FUOTA 。

LoRaWAN_FUOTA 下的 1_Image_KMS_Blob 和 1_Image_SECoreBin 仅有 icf 文件，没有.s 文件。

LoRaWAN_FUOTA 下的 1_Image_BFU 和 LoRaWAN_End_Node 都有.icf 和.s 文件。

2.1. 对比.s 文件(两个都一样)

LoRaWAN_FUOTA\LoRaWAN_End_Node\EWARM\startup_stm32wl55xx_cm4.s 和 LoRaWAN_FUOTA\1_Image_BFU\EWARM\startup_stm32wl55xx_cm4.s 是一样的!

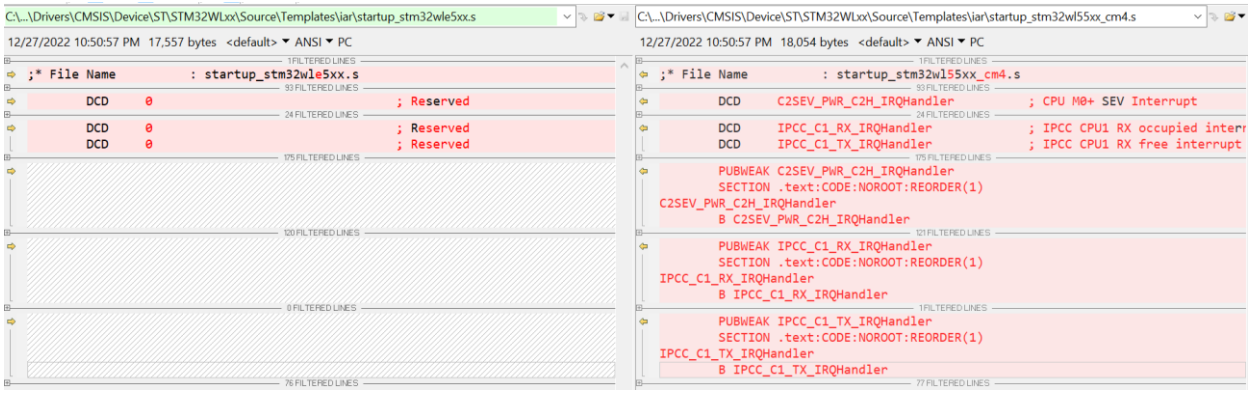
LoRaWAN_FUOTA\LoRaWAN_End_Node 和 LoRaWAN\LoRaWAN_End_Node 的 startup_stm32wl55xx_cm4.s 是相同的!

EWARM\startup_stm32wl55xx_cm4.s 和 STM32Cube_FW_WL_V1.3.0 \Drivers\CMSIS\Device\ST\STM32WLxx\Source\Templates\iar\startup_stm32wl55xx_cm4.s 是一样的!

所以 STM32WLE5 的.s 可使用 STM32Cube_FW_WL_V1.3.0 \Drivers\CMSIS\Device\ST\STM32WLxx\Source\Templates\iar\startup_stm32wle5xx.s

所以，可复制 STM32Cube_FW_WL_V1.3.0 \Drivers\CMSIS\Device\ST\STM32WLxx\Source\Templates\iar\startup_stm32wle5xx.s 到 LoRaWAN_FUOTA_WLE5CC\LoRaWAN_End_Node\EWARM 目录下

注意：startup_stm32wle5xx.s 和 startup_stm32wl55xx_cm4.s 不一样，startup_stm32wle5xx.s 中没有 C2SEV_PWR_C2H_IRQHandler，IPCC_C1_RX_IRQHandler，IPCC_C1_TX_IRQHandler 相关的配置。



2.2. 对比.icf 文件(四个都不一样)

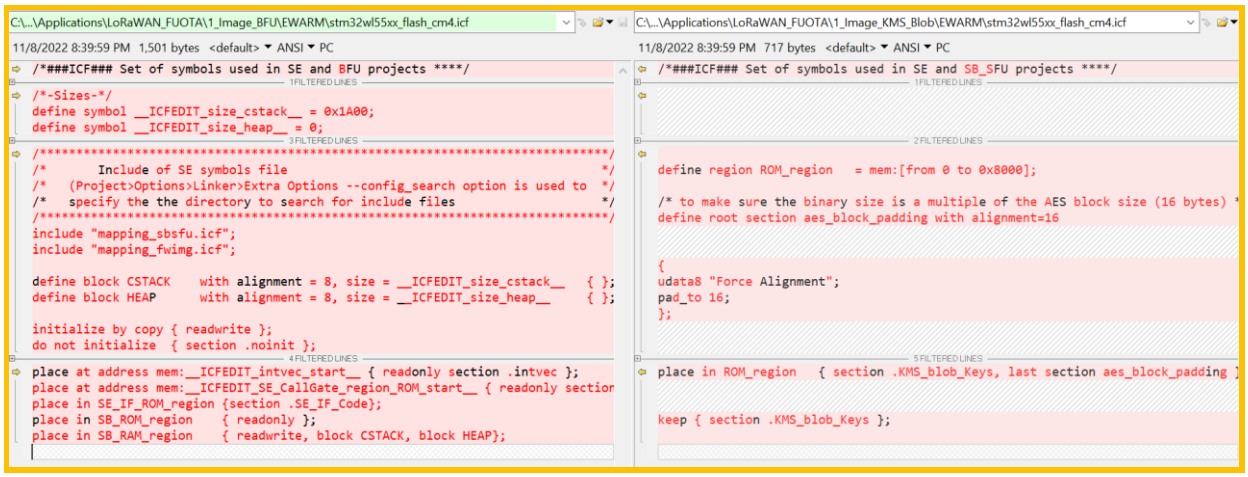
LoRaWAN_FUOTA\1_Image_BFU\EWARM\stm32wl55xx_flash_cm4.icf

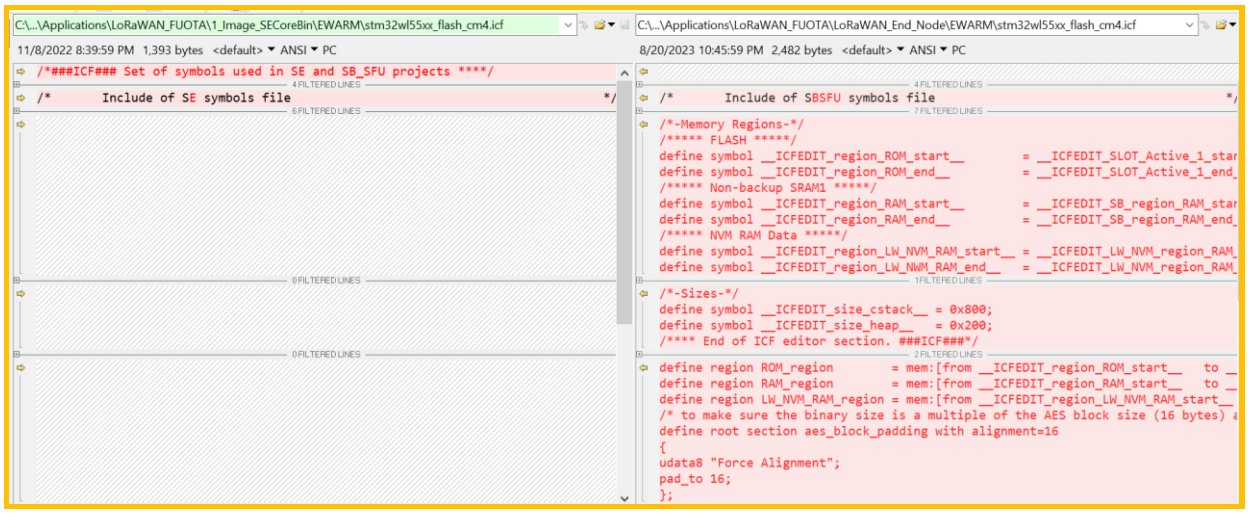
LoRaWAN_FUOTA\1_Image_KMS_Blob\EWARM\stm32wl55xx_flash_cm4.icf

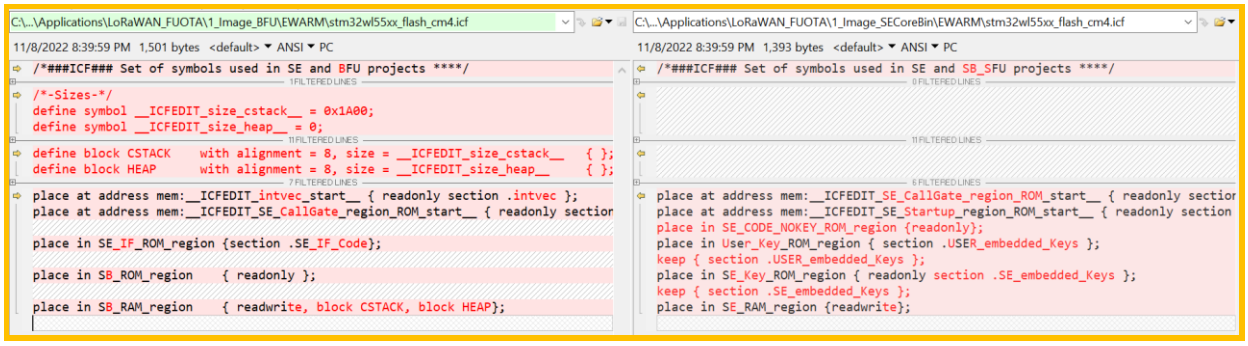
LoRaWAN_FUOTA\1_Image_SECoreBin\EWARM\stm32wl55xx_flash_cm4.icf

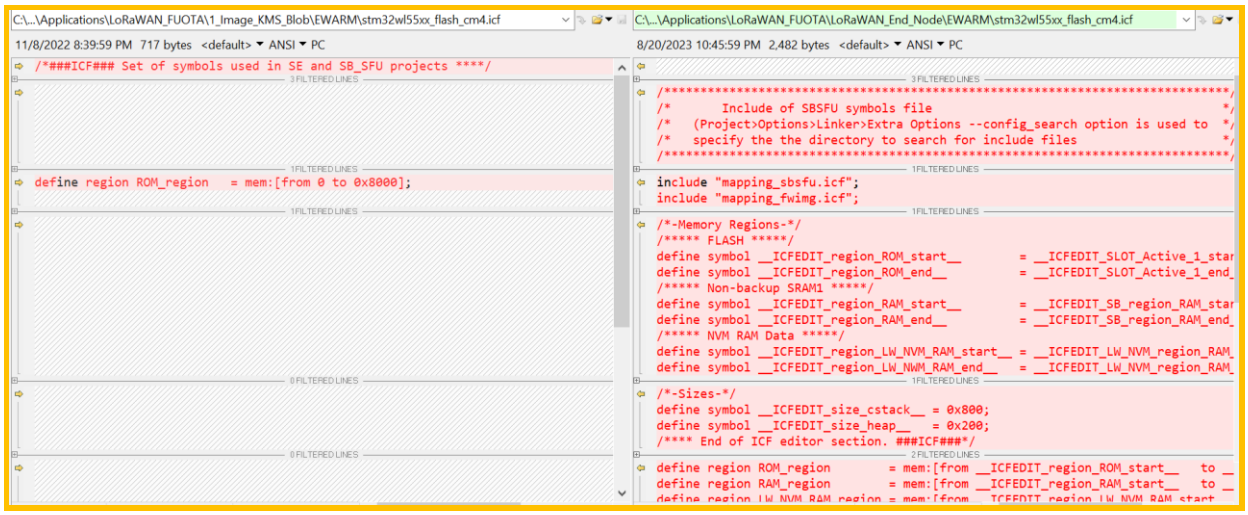
LoRaWAN_FUOTA\LoRaWAN_End_Node\EWARM\stm32wl55xx_flash_cm4.icf

这四个.icf 文件都不一样!

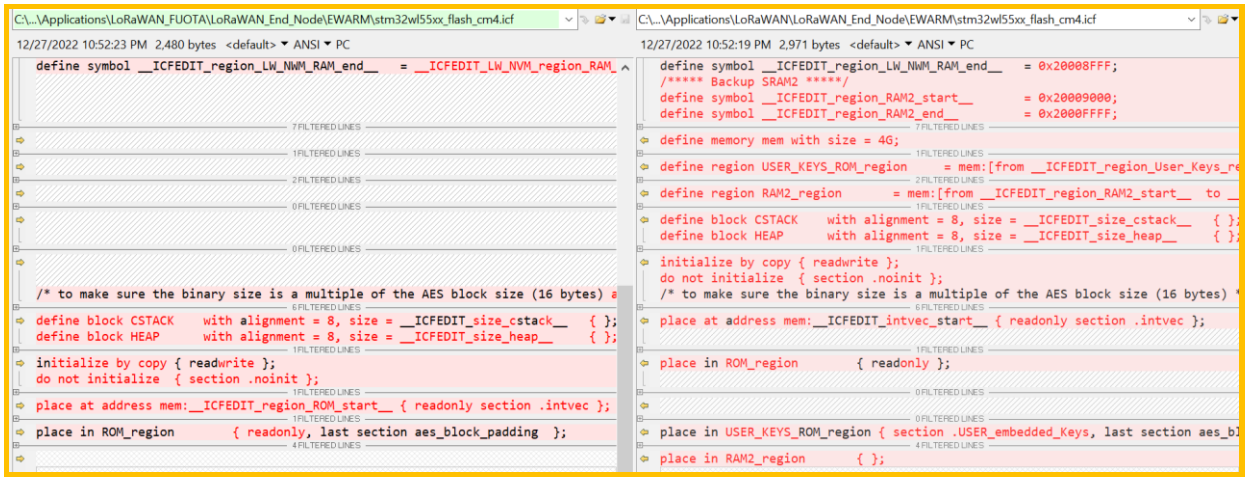

C:\...\Applications\LoRaWAN_FUOTA\1_Image_BFU\EWARM\stm32wl55xx_flash_cm4.icf
11/8/2022 8:39:59 PM 1,501 bytes <default> ANSI PC
/*###ICF### Set of symbols used in SE and BFU projects ****/
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x1A00;
define symbol __ICFEDIT_size_heap__ = 0;
/***/
/* Include of SE symbols file */
/* (Project>Options>Linker>Extra Options --config_search option is used to */
/* specify the the directory to search for include files */
/***/
include "mapping_sbsfu.icf";
include "mapping_fwimg.icf";
define block CSTACK with alignment = 8, size = __ICFEDIT_size_cstack__ { };
define block HEAP with alignment = 8, size = __ICFEDIT_size_heap__ { };
initialize by copy { readwrite };
do not initialize { section .noinit };
place at address mem:__ICFEDIT_intvec_start__ { readonly section .intvec };
place at address mem:__ICFEDIT_SE_CallGate_region_ROM_start__ { readonly section
place in SE_IF_ROM_region {section .SE_IF_Code};
place in SB_ROM_region { readonly };
place in SB_RAM_region { readwrite, block CSTACK, block HEAP};
C:\...\Applications\LoRaWAN_FUOTA\1_Image_KMS_Blob\EWARM\stm32wl55xx_flash_cm4.icf
11/8/2022 8:39:59 PM 717 bytes <default> ANSI PC
/*###ICF### Set of symbols used in SE and SB_SFU projects ****/
define region ROM_region = mem:[from 0 to 0x8000];
/* to make sure the binary size is a multiple of the AES block size (16 bytes) *
define root section aes_block_padding with alignment=16
{
udata8 "Force Alignment";
pad_to 16;
};
place in ROM_region { section .KMS_blob_Keys, last section aes_block_padding }
keep { section .KMS_blob_Keys };


C:\...\Applications\LoRaWAN_FUOTA\1_Image_SECoreBin\EWARM\stm32wl55xx_flash_cm4.icf
11/8/2022 8:39:59 PM 1,393 bytes <default> ANSI PC
/*###ICF### Set of symbols used in SE and SB_SFU projects ****/
/* Include of SE symbols file */
C:\...\Applications\LoRaWAN_FUOTA\LoRaWAN_End_Node\EWARM\stm32wl55xx_flash_cm4.icf
8/20/2023 10:45:59 PM 2,482 bytes <default> ANSI PC
/* Include of SBSFU symbols file */
/*-Memory Regions-*/
/***** FLASH *****/
define symbol __ICFEDIT_region_ROM_start__ = __ICFEDIT_SLOT_Active_1_star
define symbol __ICFEDIT_region_ROM_end__ = __ICFEDIT_SLOT_Active_1_end_
/***** Non-backup SRAM1 *****/
define symbol __ICFEDIT_region_RAM_start__ = __ICFEDIT_SB_region_RAM_star
define symbol __ICFEDIT_region_RAM_end__ = __ICFEDIT_SB_region_RAM_end_
/***** NVM RAM Data *****/
define symbol __ICFEDIT_region_LW_NVM_RAM_start__ = __ICFEDIT_LW_NVM_region_RAM_
define symbol __ICFEDIT_region_LW_NWM_RAM_end__ = __ICFEDIT_LW_NVM_region_RAM_
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x800;
define symbol __ICFEDIT_size_heap__ = 0x200;
/**** End of ICF editor section. ###ICF###*/
define region ROM_region = mem:[from __ICFEDIT_region_ROM_start__ to __
define region RAM_region = mem:[from __ICFEDIT_region_RAM_start__ to __
define region LW_NVM_RAM_region = mem:[from __ICFEDIT_region_LW_NVM_RAM_start__
/* to make sure the binary size is a multiple of the AES block size (16 bytes) a
define root section aes_block_padding with alignment=16
{
udata8 "Force Alignment";
pad_to 16;
};


C:\...\Applications\LoRaWAN_FUOTA\1_Image_BFU\EWARM\stm32wl55xx_flash_cm4.icf
11/8/2022 8:39:59 PM 1,501 bytes <default> ANSI PC
/*###ICF### Set of symbols used in SE and BFU projects ****/
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x1A00;
define symbol __ICFEDIT_size_heap__ = 0;
define block CSTACK with alignment = 8, size = __ICFEDIT_size_cstack__ { };
define block HEAP with alignment = 8, size = __ICFEDIT_size_heap__ { };
place at address mem:__ICFEDIT_intvec_start__ { readonly section .intvec };
place at address mem:__ICFEDIT_SE_CallGate_region_ROM_start__ { readonly section
place in SE_IF_ROM_region {section .SE_IF_Code};
place in SB_ROM_region { readonly };
place in SB_RAM_region { readwrite, block CSTACK, block HEAP};
C:\...\Applications\LoRaWAN_FUOTA\1_Image_SECoreBin\EWARM\stm32wl55xx_flash_cm4.icf
11/8/2022 8:39:59 PM 1,393 bytes <default> ANSI PC
/*###ICF### Set of symbols used in SE and SB_SFU projects ****/
place at address mem:__ICFEDIT_SE_CallGate_region_ROM_start__ { readonly section
place at address mem:__ICFEDIT_SE_Startup_region_ROM_start__ { readonly section
place in SE_CODE_NOKEY_ROM_region {readonly};
place in User_Key_ROM_region { section .USER_embedded_Keys };
keep { section .USER_embedded_Keys };
place in SE_Key_ROM_region { readonly section .SE_embedded_Keys };
keep { section .SE_embedded_Keys };
place in SE_RAM_region {readwrite};



LoRaWAN_FUOTA\LoRaWAN_End_Node\EWARM\stm32wl55xx_flash_cm4.icf 和 LoRaWAN\LoRaWAN_End_Node\EWARM\stm32wl55xx_flash_cm4.icf 也不一样!



因此保留原有的.icf 文件

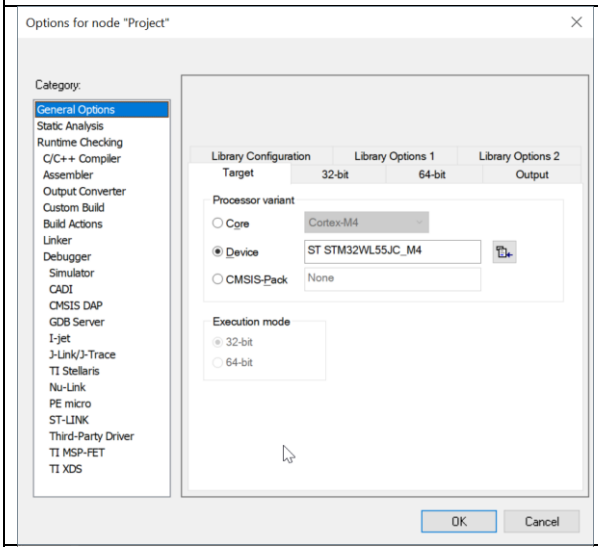
3. 移植

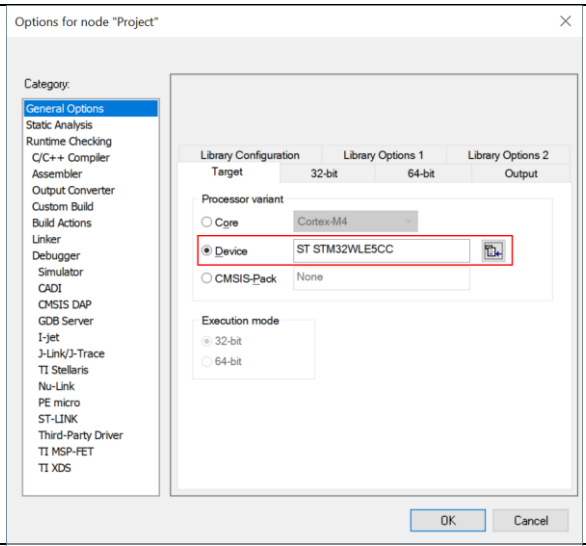
以下移植是基于 IAR IDE 和 STM32Cube_FW_WL_V1.3.0，复制 STM32Cube_FW_WL_V1.3.0\Projects\NUCLEO-WL55JC\Applications 下的 LoRaWAN_FUOTA 为 LoRaWAN_FUOTA_WLE5CC。

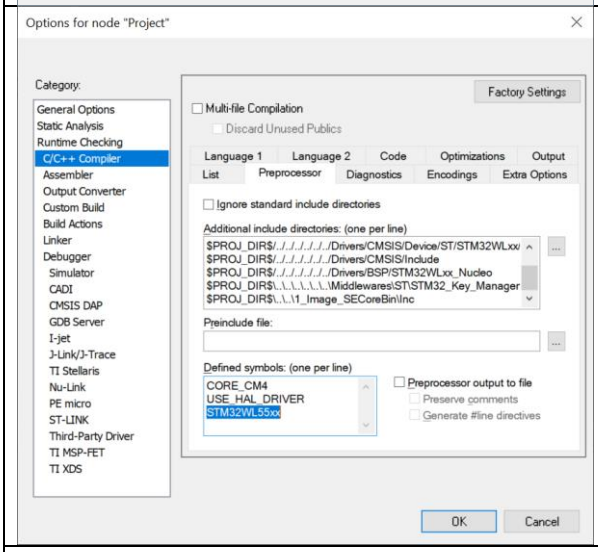
根据编译顺序依次移植 LoRaWAN_FUOTA_WLE5CC 下的 1_Image_KMS_Blob, 1_Image_SECoreBin, 1_Image_BFU, LoRaWAN_End_Node

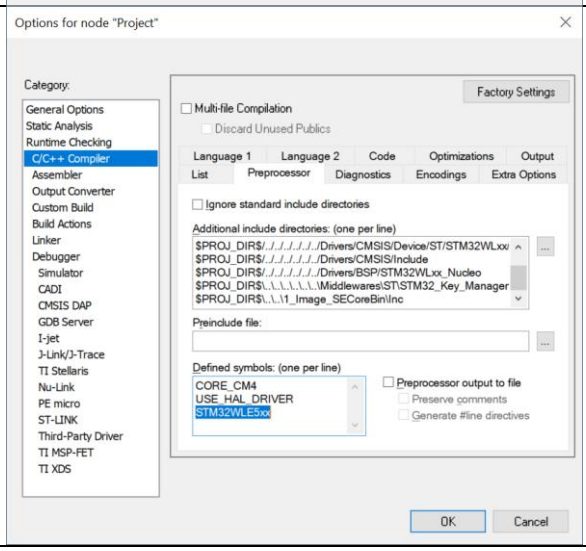
3.1. 移植 1_Image_KMS_Blob

打开，选中 Project-STM32WL55JC_Nucleo_1_Image_KMS_Blob， 按 ALT+F7 打开 Options 配置

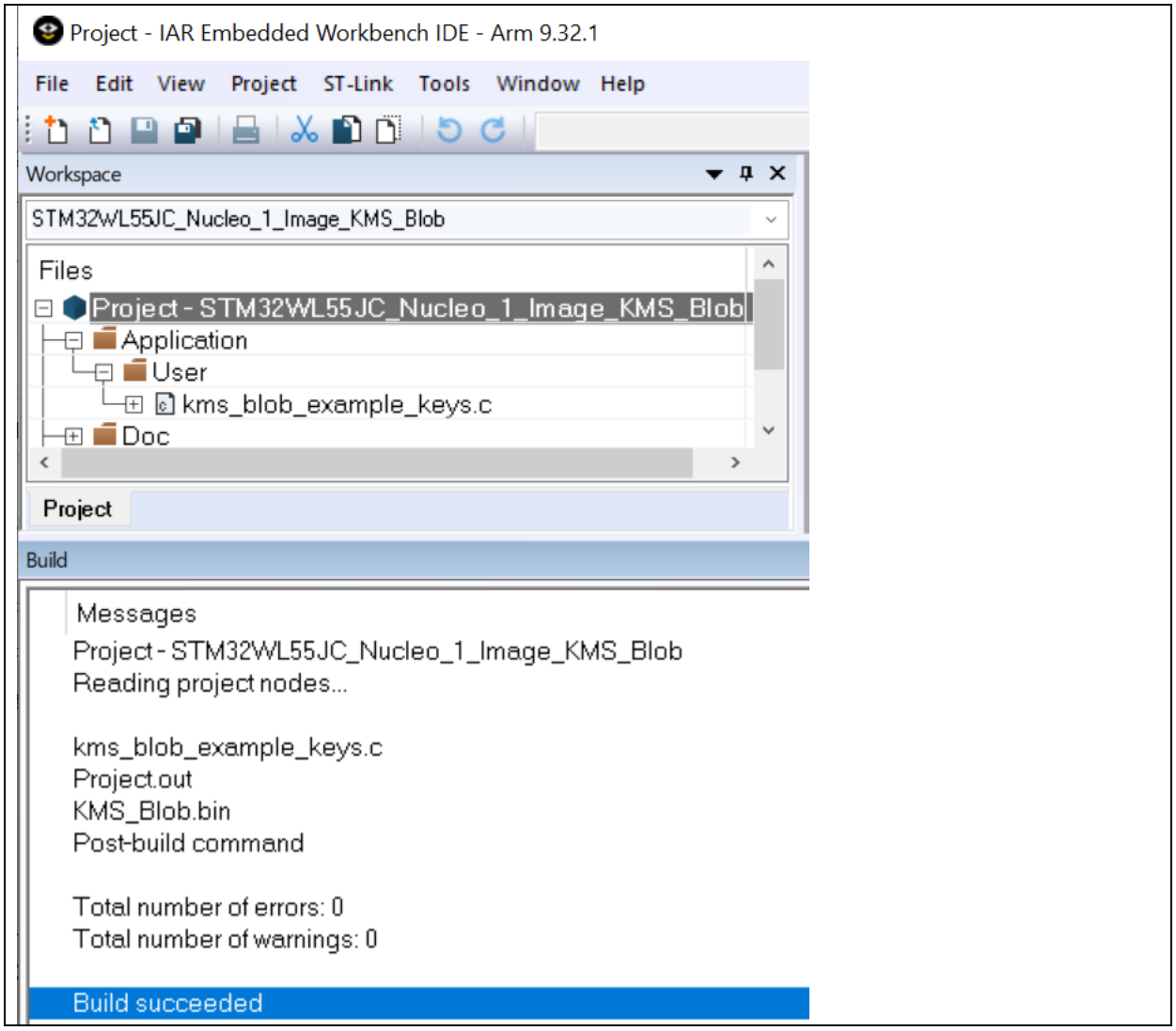






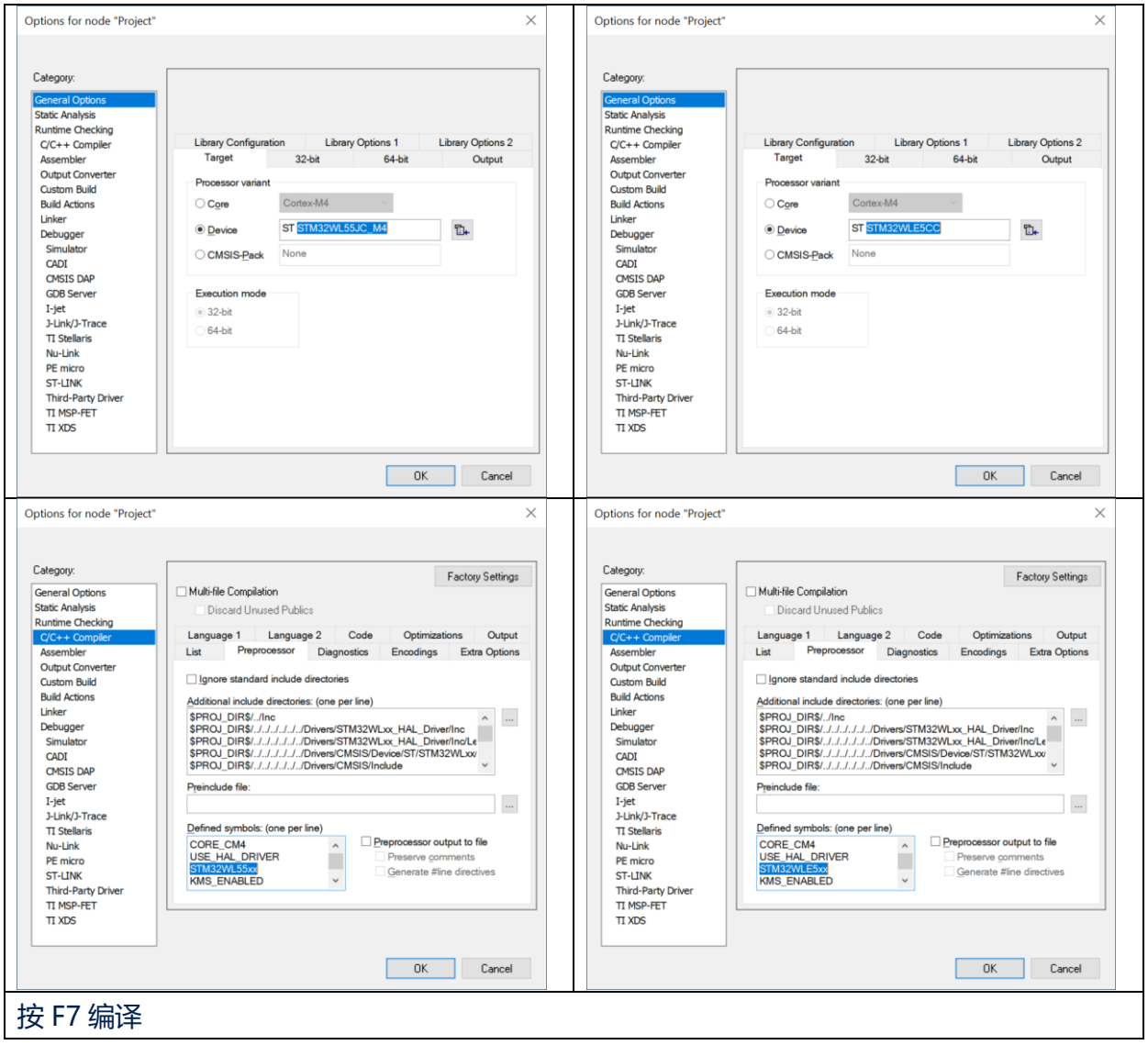


按 F7 编译

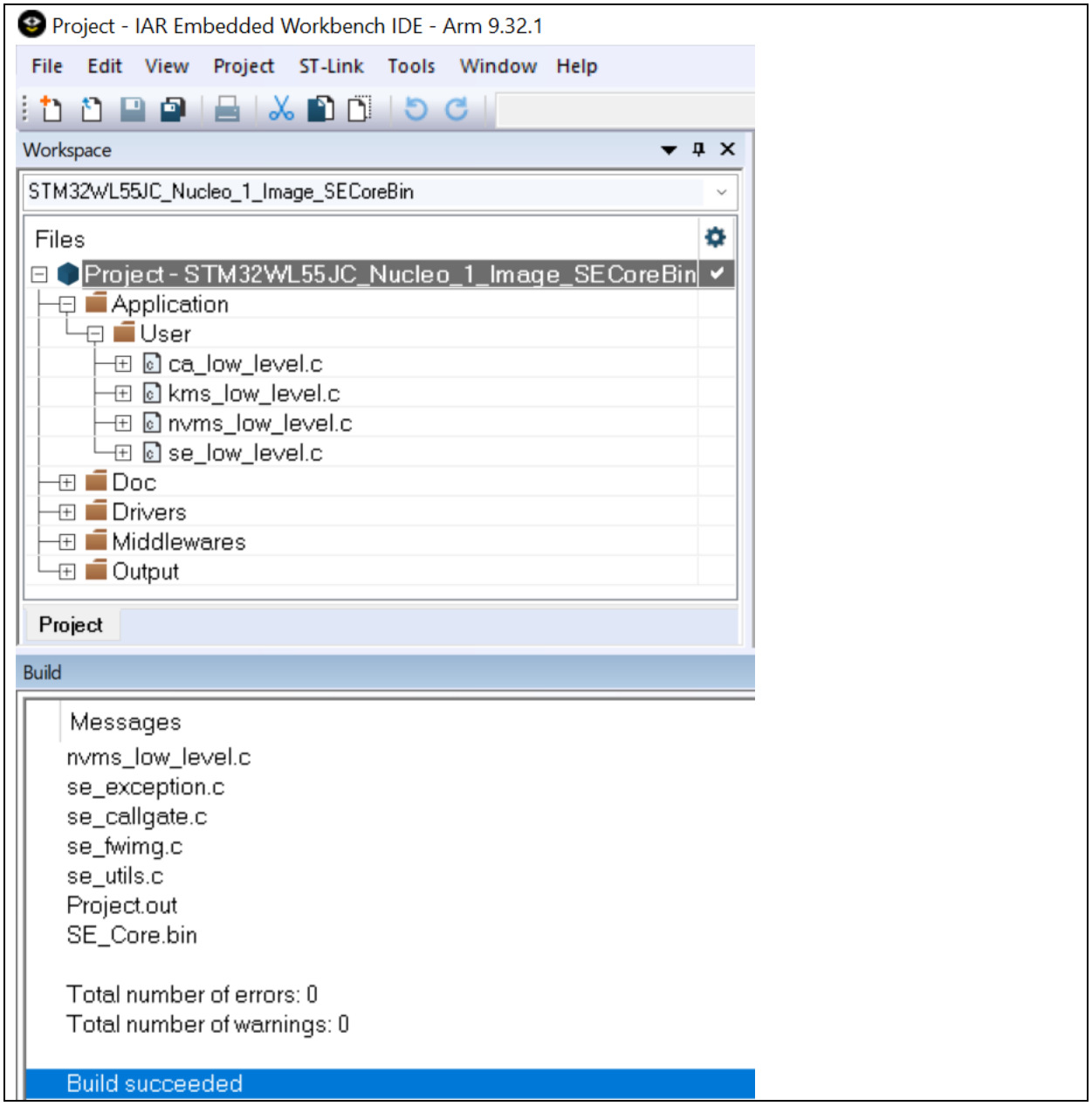


3.2. 移植 1_Image_SECoreBin

打开，选中 Project-STM32WL55JC_Nucleo_1_Image_SECoreBin， 按 ALT+F7 打开 Options 配置


Options for node "Project"
Category:
General Options
Static Analysis
Runtime Checking
C/C++ Compiler
Assembler
Output Converter
Custom Build
Build Actions
Linker
Debugger
Simulator
CADI
CMSIS DAP
GDB Server
I-jet
J-Link/J-Trace
TI Stellaris
Nu-Link
PE micro
ST-LINK
Third-Party Driver
TI MSP-FET
TI XDS
Library Configuration
Library Options 1
Library Options 2
Target
32-bit
64-bit
Output
Processor variant
Core
Cortex-M4
Device
ST STM32WL55JC_M4
CMSIS-Pack
None
Execution mode
32-bit
64-bit
OK
Cancel
Options for node "Project"
Category:
General Options
Static Analysis
Runtime Checking
C/C++ Compiler
Assembler
Output Converter
Custom Build
Build Actions
Linker
Debugger
Simulator
CADI
CMSIS DAP
GDB Server
I-jet
J-Link/J-Trace
TI Stellaris
Nu-Link
PE micro
ST-LINK
Third-Party Driver
TI MSP-FET
TI XDS
Library Configuration
Library Options 1
Library Options 2
Target
32-bit
64-bit
Output
Processor variant
Core
Cortex-M4
Device
ST STM32WLE5CC
CMSIS-Pack
None
Execution mode
32-bit
64-bit
OK
Cancel
Options for node "Project"
Category:
General Options
Static Analysis
Runtime Checking
C/C++ Compiler
Assembler
Output Converter
Custom Build
Build Actions
Linker
Debugger
Simulator
CADI
CMSIS DAP
GDB Server
I-jet
J-Link/J-Trace
TI Stellaris
Nu-Link
PE micro
ST-LINK
Third-Party Driver
TI MSP-FET
TI XDS
Factory Settings
Multi-file Compilation
Discard Unused Publics
Language 1
Language 2
Code
Optimizations
Output
List
Preprocessor
Diagnostics
Encodings
Extra Options
Ignore standard include directories
Additional include directories: (one per line)
$PROJ_DIR$/../Inc
$PROJ_DIR$/../../../../../../Drivers/STM32WLxx_HAL_Driver/Inc
$PROJ_DIR$/../../../../../../Drivers/STM32WLxx_HAL_Driver/Inc/Le
$PROJ_DIR$/../../../../../../Drivers/CMSIS/Device/ST/STM32WLxx/
$PROJ_DIR$/../../../../../../Drivers/CMSIS/Include
Preinclude file:
Defined symbols: (one per line)
CORE_CM4
USE_HAL_DRIVER
STM32WL55xx
KMS_ENABLED
Preprocessor output to file
Preserve comments
Generate #line directives
OK
Cancel
Options for node "Project"
Category:
General Options
Static Analysis
Runtime Checking
C/C++ Compiler
Assembler
Output Converter
Custom Build
Build Actions
Linker
Debugger
Simulator
CADI
CMSIS DAP
GDB Server
I-jet
J-Link/J-Trace
TI Stellaris
Nu-Link
PE micro
ST-LINK
Third-Party Driver
TI MSP-FET
TI XDS
Factory Settings
Multi-file Compilation
Discard Unused Publics
Language 1
Language 2
Code
Optimizations
Output
List
Preprocessor
Diagnostics
Encodings
Extra Options
Ignore standard include directories
Additional include directories: (one per line)
$PROJ_DIR$/../Inc
$PROJ_DIR$/../../../../../../Drivers/STM32WLxx_HAL_Driver/Inc
$PROJ_DIR$/../../../../../../Drivers/STM32WLxx_HAL_Driver/Inc/Le
$PROJ_DIR$/../../../../../../Drivers/CMSIS/Device/ST/STM32WLxx/
$PROJ_DIR$/../../../../../../Drivers/CMSIS/Include
Preinclude file:
Defined symbols: (one per line)
CORE_CM4
USE_HAL_DRIVER
STM32WLE5xx
KMS_ENABLED
Preprocessor output to file
Preserve comments
Generate #line directives
OK
Cancel

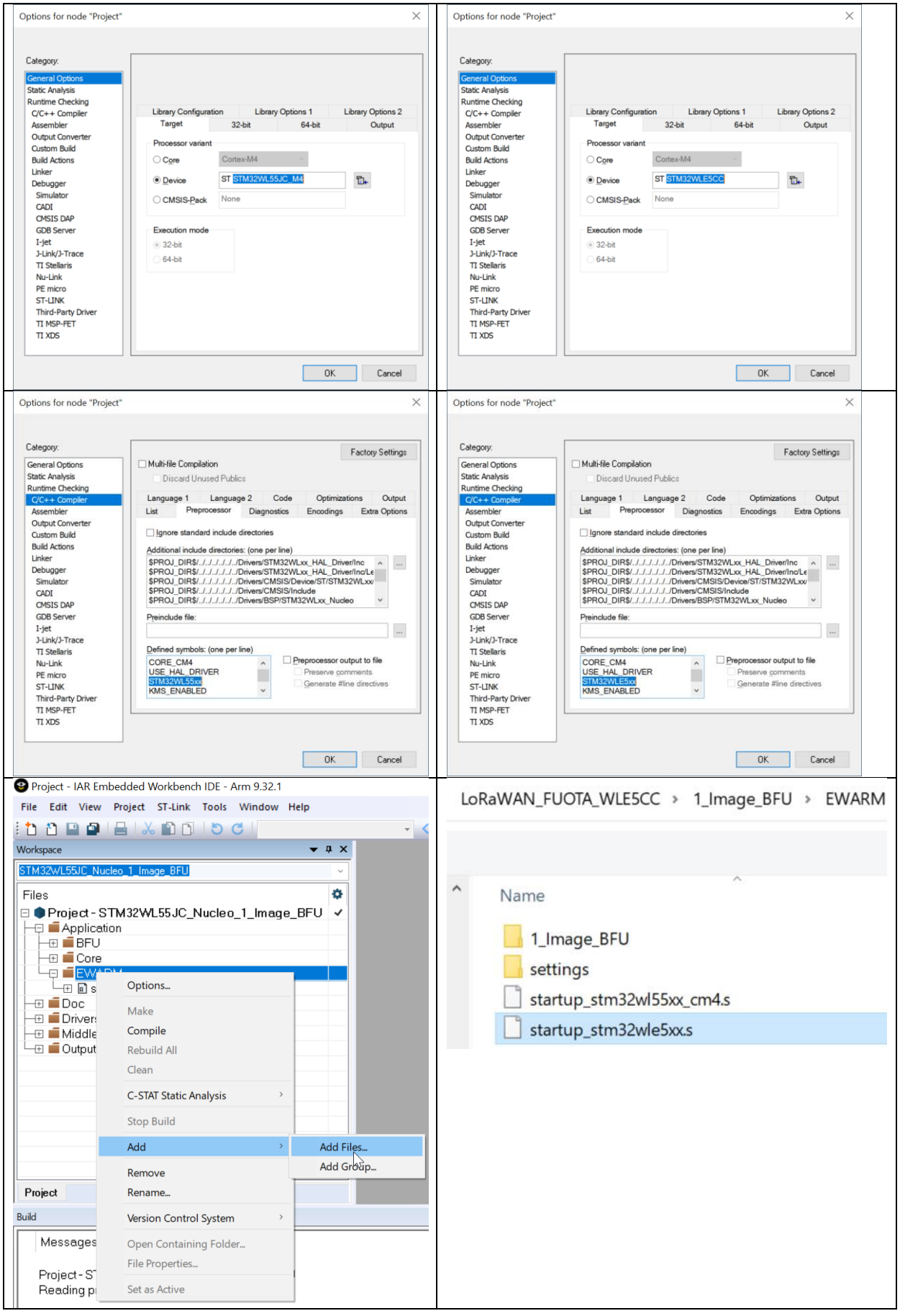
按 F7 编译

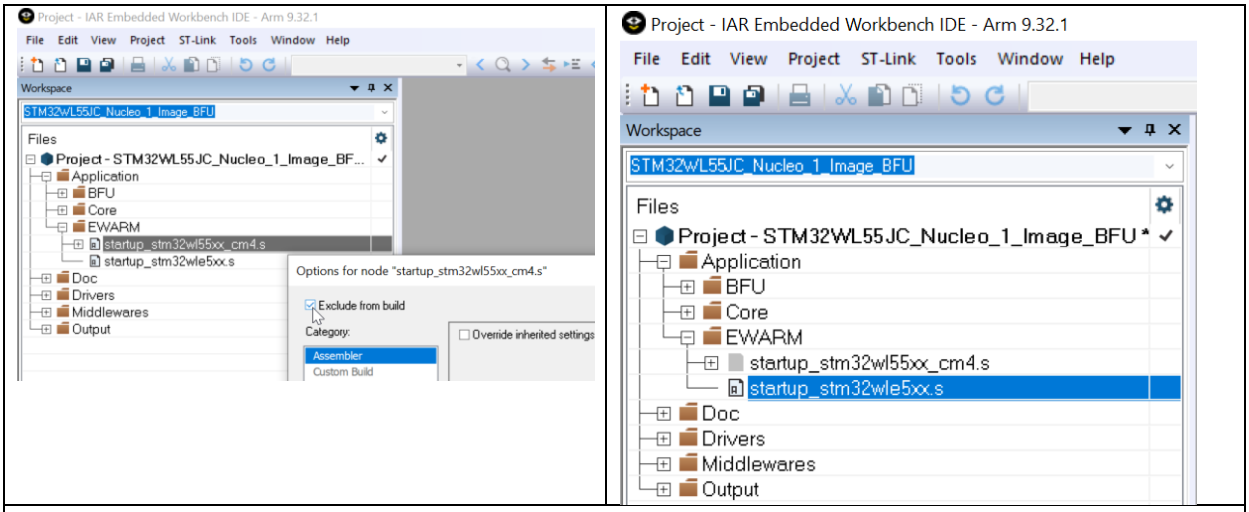


3.3. 移植 1_Image_BFU

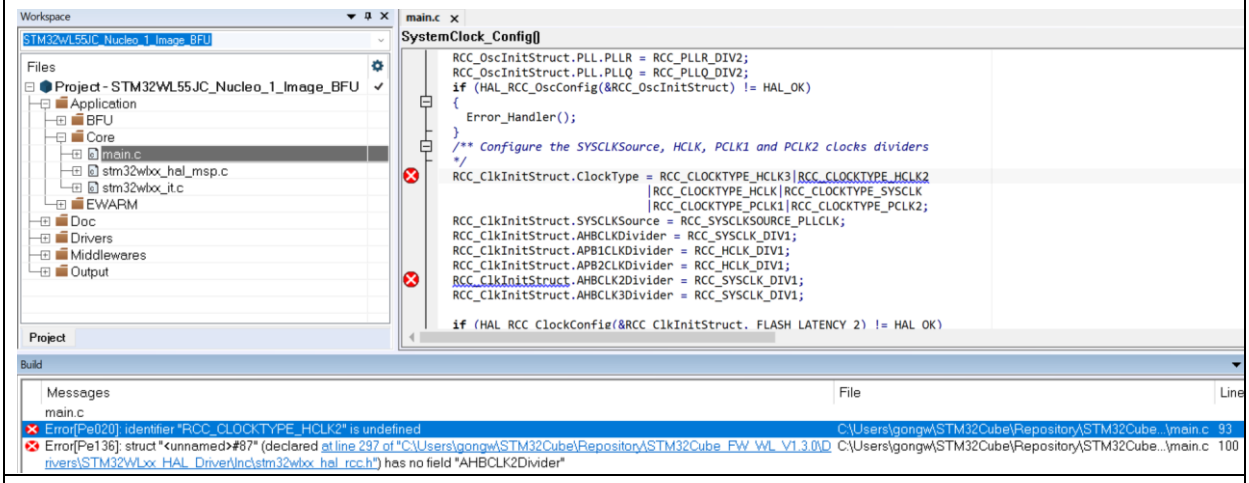
复制 STM32Cube_FW_WL_V1.3.0 \Drivers\CMSIS\Device\ST\STM32WLxx\Source\Templates\iar\startup_stm32wle5xx.s 到 LoRaWAN_FUOTA_WLE5CC\1_Image_BFU\EWARM 目录下

打开，选中 Project-STM32WL55JC_Nucleo_1_Image_SECoreBin， 按 ALT+F7 打开 Options 配置


Options for node "Project"
Category:
General Options
Static Analysis
Runtime Checking
C/C++ Compiler
Assembler
Output Converter
Custom Build
Build Actions
Linker
Debugger
Simulator
CADI
CMSIS DAP
GDB Server
I-jet
J-Link/J-Trace
TI Stellaris
Nu-Link
PE micro
ST-LINK
Third-Party Driver
TI MSP-FET
TI XDS
Library Configuration
Library Options 1
Library Options 2
Target
32-bit
64-bit
Output
Processor variant
Core
Cortex-M4
Device
ST STM32WL55JC_M4
CMSIS-Pack
None
Execution mode
32-bit
64-bit
OK
Cancel
Options for node "Project"
Device
ST STM32WLE5CC
Factory Settings
Multi-file Compilation
Discard Unused Publics
Language 1
Language 2
Code
Optimizations
Output
List
Preprocessor
Diagnostics
Encodings
Extra Options
Ignore standard include directories
Additional include directories: (one per line)
$PROJ_DIR$/../../../../../Drivers/STM32WLxx_HAL_Driver/Inc
$PROJ_DIR$/../../../../../Drivers/STM32WLxx_HAL_Driver/Inc/Le
$PROJ_DIR$/../../../../../Drivers/CMSIS/Device/ST/STM32WLxx/
$PROJ_DIR$/../../../../../Drivers/CMSIS/Include
$PROJ_DIR$/../../../../../Drivers/BSP/STM32WLxx_Nucleo
Preinclude file:
Defined symbols: (one per line)
CORE_CM4
USE_HAL_DRIVER
STM32WL55xx
STM32WLE5xx
KMS_ENABLED
Preprocessor output to file
Preserve comments
Generate #line directives
Project - IAR Embedded Workbench IDE - Arm 9.32.1
File Edit View Project ST-Link Tools Window Help
Workspace
STM32WL55JC_Nucleo_1_Image_BFU
Files
Project - STM32WL55JC_Nucleo_1_Image_BFU
Application
BFU
Core
Doc
Options...
Make
Compile
Rebuild All
Clean
C-STAT Static Analysis
Stop Build
Add
Add Files...
Add Group...
Remove
Rename...
Version Control System
Open Containing Folder...
File Properties...
Set as Active
Project
Build
Messages
LoRaWAN_FUOTA_WLE5CC › 1_Image_BFU › EWARM
Name
1_Image_BFU
settings
startup_stm32wl55xx_cm4.s
startup_stm32wle5xx.s



按 F7 编译



main.c

Error[Pe020]: identifier "RCC_CLOCKTYPE_HCLK2" is undefined

C:\Users\gongw\STM32Cube\Repository\CubeWL\STM32Cube_FW_WL_V1.3.0_FUOTA\Projects\NUCLEO-WL55JC\Applications\LoRaWAN_FUOTA_WLE5CC\1_Image_BFU\Core\Src\main.c93

Error[Pe136]: struct "<unnamed>#87" (declared Hat line 297 of "C:\Users\gongw\STM32Cube\Repository\CubeWL\STM32Cube_FW_WL_V1.3.0_FUOTA\Drivers\STM32WLxx_HAL_Driver\Inc\stm32wlxx_hal_rcc.h"h) has no field "AHBCLK2Divider"

C:\Users\gongw\STM32Cube\Repository\CubeWL\STM32Cube_FW_WL_V1.3.0_FUOTA\Projects\NUCLEO-WL55JC\Applications\LoRaWAN_FUOTA_WLE5CC\1_Image_BFU\Core\Src\main.c100

修改代码如下

```
/** Configure the SYSCLKSource, HCLK, PCLK1 and PCLK2 clocks dividers
  */
```

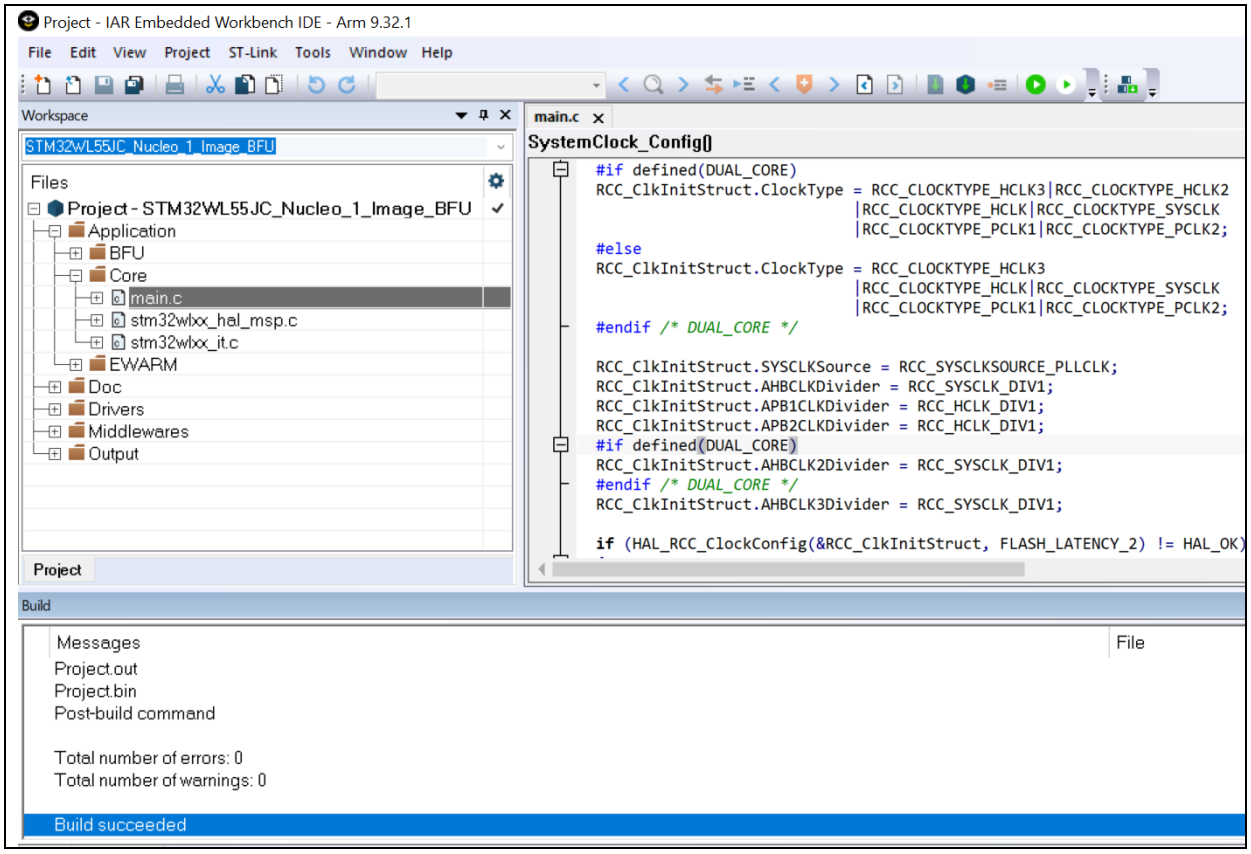
```
#if defined(DUAL_CORE)
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK3|RCC_CLOCKTYPE_HCLK2
                              |RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
#else
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK3
                              |RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  #endif /* DUAL_CORE */

  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
#if defined(DUAL_CORE)
  RCC_ClkInitStruct.AHBCLK2Divider = RCC_SYSCLK_DIV1;
#endif /* DUAL_CORE */

  RCC_ClkInitStruct.AHBCLK3Divider = RCC_SYSCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
```

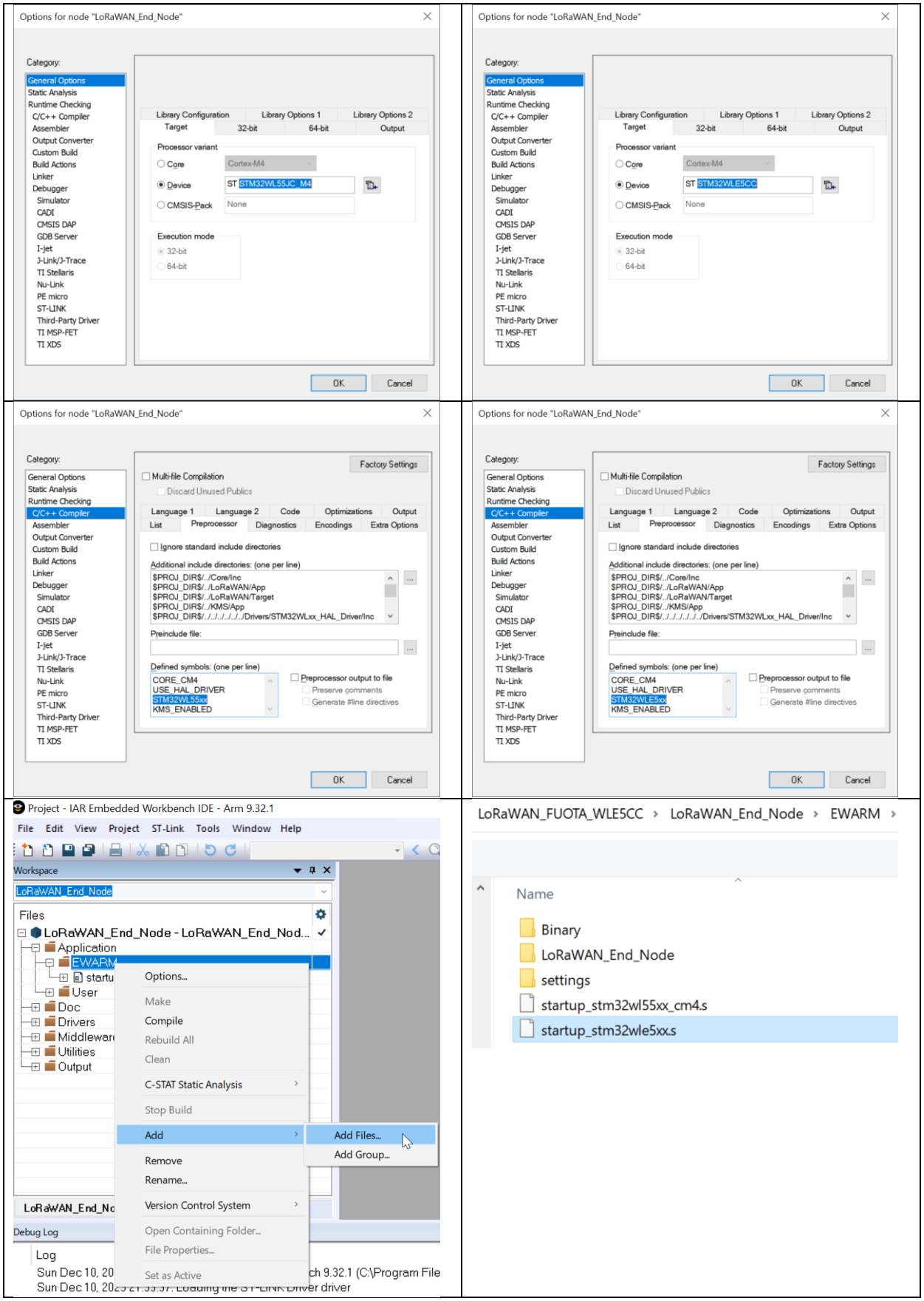
按 F7 编译

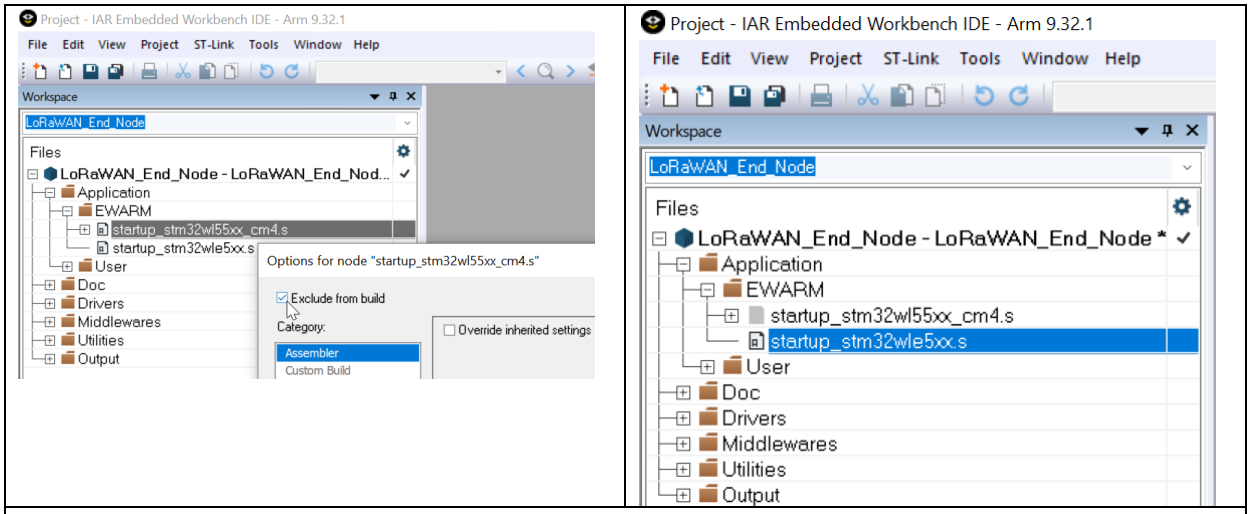


3.4. 移植 LoRaWAN_End_Node

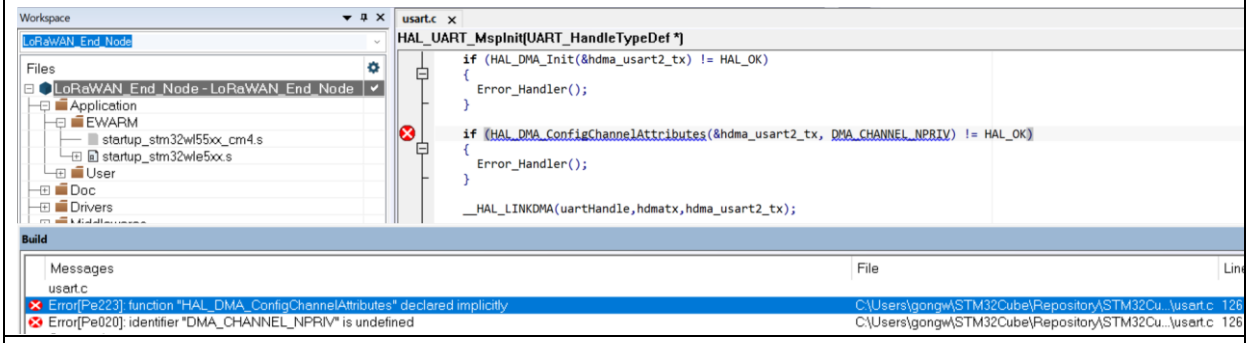
复制 STM32Cube_FW_WL_V1.3.0 \Drivers\CMSIS\Device\ST\STM32WLxx\Source\Templates\iar\startup_stm32wle5xx.s 到 LoRaWAN_FUOTA_WLE5CC\LoRaWAN_End_Node\EWARM 目录下

打开，选中 Project-STM32WL55JC_Nucleo_1_Image_SECoreBin， 按 ALT+F7 打开 Options 配置


Options for node "LoRaWAN_End_Node"
Category:
General Options
Static Analysis
Runtime Checking
C/C++ Compiler
Assembler
Output Converter
Custom Build
Build Actions
Linker
Debugger
Simulator
CADI
CMSIS DAP
GDB Server
I-jet
J-Link/J-Trace
TI Stellaris
Nu-Link
PE micro
ST-LINK
Third-Party Driver
TI MSP-FET
TI XDS
Library Configuration
Library Options 1
Library Options 2
Target
32-bit
64-bit
Output
Processor variant
Core
Cortex-M4
Device
ST STM32WL55JC_M4
CMSIS-Pack
None
Execution mode
32-bit
64-bit
OK
Cancel
Options for node "LoRaWAN_End_Node"
Category:
General Options
Static Analysis
Runtime Checking
C/C++ Compiler
Assembler
Output Converter
Custom Build
Build Actions
Linker
Debugger
Simulator
CADI
CMSIS DAP
GDB Server
I-jet
J-Link/J-Trace
TI Stellaris
Nu-Link
PE micro
ST-LINK
Third-Party Driver
TI MSP-FET
TI XDS
Library Configuration
Library Options 1
Library Options 2
Target
32-bit
64-bit
Output
Processor variant
Core
Cortex-M4
Device
ST STM32WLE5CC
CMSIS-Pack
None
Execution mode
32-bit
64-bit
OK
Cancel
Options for node "LoRaWAN_End_Node"
Category:
General Options
Static Analysis
Runtime Checking
C/C++ Compiler
Assembler
Output Converter
Custom Build
Build Actions
Linker
Debugger
Simulator
CADI
CMSIS DAP
GDB Server
I-jet
J-Link/J-Trace
TI Stellaris
Nu-Link
PE micro
ST-LINK
Third-Party Driver
TI MSP-FET
TI XDS
Factory Settings
Multi-file Compilation
Discard Unused Publics
Language 1
Language 2
Code
Optimizations
Output
List
Preprocessor
Diagnostics
Encodings
Extra Options
Ignore standard include directories
Additional include directories: (one per line)
$PROJ_DIR$/../Core/Inc
$PROJ_DIR$/../LoRaWAN/App
$PROJ_DIR$/../LoRaWAN/Target
$PROJ_DIR$/../KMS/App
$PROJ_DIR$/../../../../../../Drivers/STM32WLxx_HAL_Driver/Inc
Preinclude file:
Defined symbols: (one per line)
CORE_CM4
USE_HAL_DRIVER
STM32WL55xx
KMS_ENABLED
Preprocessor output to file
Preserve comments
Generate #line directives
OK
Cancel
Options for node "LoRaWAN_End_Node"
Category:
General Options
Static Analysis
Runtime Checking
C/C++ Compiler
Assembler
Output Converter
Custom Build
Build Actions
Linker
Debugger
Simulator
CADI
CMSIS DAP
GDB Server
I-jet
J-Link/J-Trace
TI Stellaris
Nu-Link
PE micro
ST-LINK
Third-Party Driver
TI MSP-FET
TI XDS
Factory Settings
Multi-file Compilation
Discard Unused Publics
Language 1
Language 2
Code
Optimizations
Output
List
Preprocessor
Diagnostics
Encodings
Extra Options
Ignore standard include directories
Additional include directories: (one per line)
$PROJ_DIR$/../Core/Inc
$PROJ_DIR$/../LoRaWAN/App
$PROJ_DIR$/../LoRaWAN/Target
$PROJ_DIR$/../KMS/App
$PROJ_DIR$/../../../../../../Drivers/STM32WLxx_HAL_Driver/Inc
Preinclude file:
Defined symbols: (one per line)
CORE_CM4
USE_HAL_DRIVER
STM32WLE5xx
KMS_ENABLED
Preprocessor output to file
Preserve comments
Generate #line directives
OK
Cancel
Project - IAR Embedded Workbench IDE - Arm 9.32.1
File Edit View Project ST-Link Tools Window Help
Workspace
LoRaWAN_End_Node
Files
LoRaWAN_End_Node - LoRaWAN_End_Nod...
Application
EWARM
User
Doc
Drivers
Middleware
Utilities
Output
Options...
Make
Compile
Rebuild All
Clean
C-STAT Static Analysis
Stop Build
Add
Add Files...
Add Group...
Remove
Rename...
Version Control System
Open Containing Folder...
File Properties...
Set as Active
Debug Log
Log
Sun Dec 10, 20
ch 9.32.1 (C:\Program File
Sun Dec 10, 20
LoRaWAN_FUOTA_WLE5CC › LoRaWAN_End_Node › EWARM ›
Name
Binary
LoRaWAN_End_Node
settings
startup_stm32wl55xx_cm4.s
startup_stm32wle5xx.s



按 F7 编译

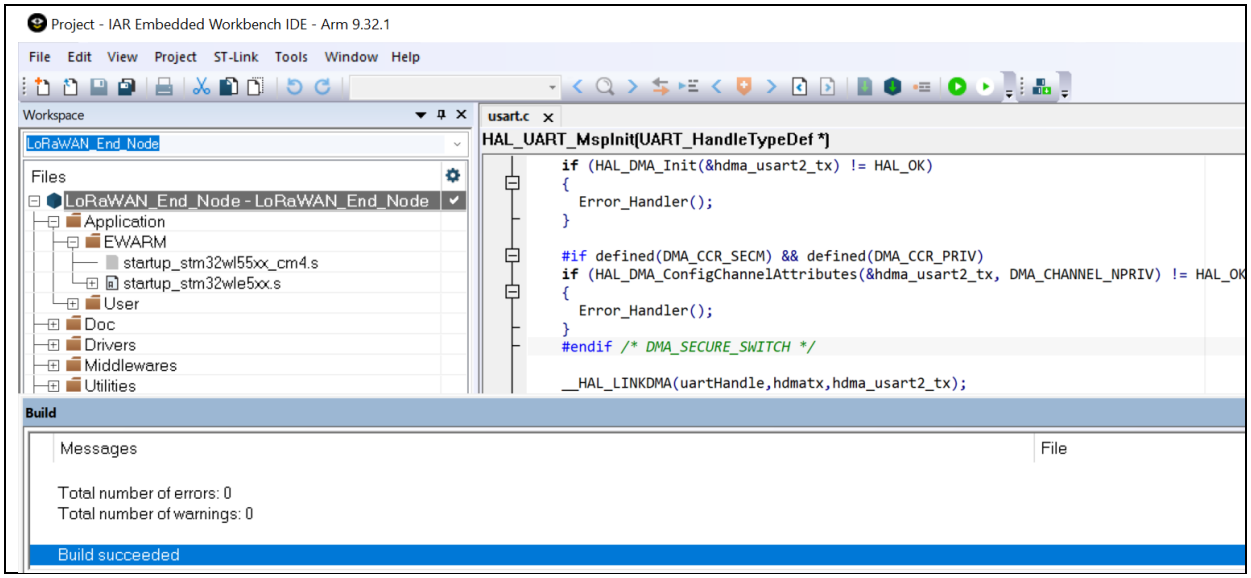


修改代码如下

```
  #if defined(DMA_CCR_SECM) && defined(DMA_CCR_PRIV)
  if (HAL_DMA_ConfigChannelAttributes(&hdma_usart2_tx,
DMA_CHANNEL_NPRIV) != HAL_OK)
  {
    Error_Handler();
  }
  #endif /* DMA_SECURE_SWITCH */

  __HAL_LINKDMA(uartHandle,hdmatx,hdma_usart2_tx);
```

按 F7 编译



3.5. 移植 disable_security.bat

STM32WLE5CC 的 OptionBytes 中没 C2BOOT_LOCK，HDPAD，SUBGHSPISD，SNBRSA，SBRSA，SBRV，因此注释掉 LoRaWAN_FUOTA_WLE5CC\Scripts\disable_security.bat 这些 Option 的设置。

```
echo IWDG: Independent watchdog counter frozen in Stop/Standby modes
call :write_ob IWGD_STDBY 0x0 IWDG_STOP 0x0 || goto :eof

echo BOOT: CPU1 CM4 Boot lock disabled
call :write_ob BOOT_LOCK 0x0 || goto :eof

REM echo BOOT: CPU1+CPU2 CM0+ Boot lock disabled
REM call :write_ob BOOT_LOCK 0x0 C2BOOT_LOCK 0x0 || goto :eof

REM echo ------ Security Configuration ------
REM echo HDPAD: User Flash hide protection area access disabled
REM call :write_ob HDPAD 0x1 || goto :eof

REM echo SPISD: SPI3 security disabled
REM call :write_ob SUBGHSPISD 0x1 || goto :eof

REM echo SBRSA: Reset default value of SRAM Start address secure
```

```
REM call :write_ob SNBRSA 0x1F SBRSA 0x1F || goto :eof

REM echo SBRV: Reset default value of CPU2 Boot start address
REM call :write_ob SBRV 0x8000 || goto :eof

exit /B 0
```

3.6. 配置 setenv.bat

在 LoRaWAN_FUOTA_WLE5CC\Scripts\setenv.bat 中配置 CUBEPROG_EXE，EWARM_EXE，MDK_EXE，CUBEIDE_EXE 的正确路径

4. 编译

可在 LoRaWAN_FUOTA_WLE5CC\Scripts\EWARM\build.bat 编译 LoRaWAN_End_Node 之后加个 “pause” 命令，查看编译 log

```
if %compile_all_projects% equ 1 (
  echo ###########################################
  echo # 1- Compile all BFU projects
  echo ###########################################
  call :compile_func %current_dir%..\..\1_Image_KMS_Blob\EWARM\Project.ewp || goto :eof
  call :compile_func %current_dir%..\..\1_Image_SECoreBin\EWARM\Project.ewp || goto :eof
  call :compile_func %current_dir%..\..\1_Image_BFU\EWARM\Project.ewp || goto :eof
)

echo ###########################################
echo # 2- Compile Application project
echo ###########################################
call :compile_func %current_dir%..\..\LoRaWAN_End_Node\EWARM\LoRaWAN_End_Node.ewp || goto :eof

pause

exit /B 0
```

```
C:\WINDOWS\system32\cmd.exe
LoRaMac.c
LoRaWAN_End_Node.out
LoRaWAN_End_Node.bin
Post-build command
Postbuild with windows executable

Total number of errors: 0
Total number of warnings: 0

Build succeeded
Press any key to continue . . .
```

还可将 rebuild 改为 0，这样再次运行 build.bat 时，不会 clean，编译时间会短；否认会 clean 完全重新编译，时间会长!

```
echo ##############################
echo # 0- Set all global variables
echo ##############################
call %current_dir%..\setenv.bat
set compile_all_projects=1
set rebuild=1
```

=>

```
echo ##############################
echo # 0- Set all global variables
echo ##############################
call %current_dir%..\setenv.bat
set compile_all_projects=1
set rebuild=0
```

```
Select C:\WINDOWS\system32\cmd.exe
########################################
# 0- Set all global variables
########################################
########################################
# 1- Compile all BFU projects
########################################

     IAR Command Line Build Utility V9.1.5.10207
     Copyright 2002-2022 IAR Systems AB.


Project - STM32WL55JC_Nucleo_1_Image_KMS_Blob
Reading project nodes...


     IAR Command Line Build Utility V9.1.5.10207
     Copyright 2002-2022 IAR Systems AB.


Project - STM32WL55JC_Nucleo_1_Image_SECoreBin
Reading project nodes...

Pre-build command
Prebuild with windows executable
       1 file(s) copied.
       1 file(s) copied.
       1 file(s) copied.

Total number of errors: 0
Total number of warnings: 0

Build succeeded

     IAR Command Line Build Utility V9.1.5.10207
     Copyright 2002-2022 IAR Systems AB.


Project - STM32WL55JC_Nucleo_1_Image_BFU
Reading project nodes...

########################################
# 2- Compile Application project
########################################

     IAR Command Line Build Utility V9.1.5.10207
     Copyright 2002-2022 IAR Systems AB.


LoRaWAN_End_Node - LoRaWAN_End_Node
Reading project nodes...

Press any key to continue . . .
```

5. 下载

通过 STLink 连接 STM32WLE5CC 的板子到 PC 上，运行 program.bat 即可下载 LoRaWAN_End_Node\EWARM\Binary\BFU_LoRaWAN_End_Node.bin 到 flash 的 0x08000000 处。

6. 运行

下载完 BFU_LoRaWAN_End_Node.bin 之后，将 STM32WLE5CC 的板子重新断电再上电！至此 STM32WLE5CC 上可运行 LoRaWAN_FUOTA 了。

7. 小结

移植要点：

1. 使用 STM32Cube_FW_WL_V1.3.0 \Drivers\CMSIS\Device\ST\STM32WLxx\Source\Templates\iar\startup_stm32wl55xx_cm4，
2. 保持原有的.icf 文件
3. 通过 IDE 修改 Device 由 ST STM32WL55JC 改为 ST STM32WLE5CC，将宏定义
4. STM32WL55XX 改为 STM32WLE5XX
5. 注释掉或删除 Scripts\disable_security.bat 中的 C2BOOT_LOCK，HDPAD，SUBGHSPISD，SNBRSA，SBRSA，SBRV 的 OtionBytes 配置。

文档中所用到的工具及版本

STM32Cube_FW_WL_V1.3.0

版本历史

日期	版本	变更
2024 年 05 月 16 日	1.0	首版发布

重要通知 - 请仔细阅读

意法半导体公司及其子公司 （“ST”）保留随时对 ST 产品和 / 或本文档进行变更的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。 ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。若需 ST 商标的更多信息，请参考 *www.st.com/trademarks*。所有其他产品或服务名称均为其 各自所有者的财产。

本文档是 ST 中国本地团队的技术性文章，旨在交流与分享，并期望借此给予客户产品应用上足够的帮助或提醒。若文中内容存有局限或与 ST 官网资料不一致，请以实际应用验证结果和 ST 官网最新发布的内容为准。您拥有完全自主权是否采纳本文档（包括代码，电路图等）信息，我们也不承担因使用或采纳本文档内容而导致的任何风险。

本文档中的信息取代本文档所有早期版本中提供的信息。