



XMC7000 OTW(Over-The-Wire)固件升级中文参考指南

Raymond Duan

2025 8 23



Table of contents

1	OTW固件升级程序原理介绍	3
2	前期环境准备	9
3	Overwrite模式固件升级步骤(I2C)	15
4	Overwrite模式固件升级步骤(UART)	25
5	Overwrite模式固件升级步骤(SPI)	37
6	Swap模式固件升级步骤	48

Table of contents

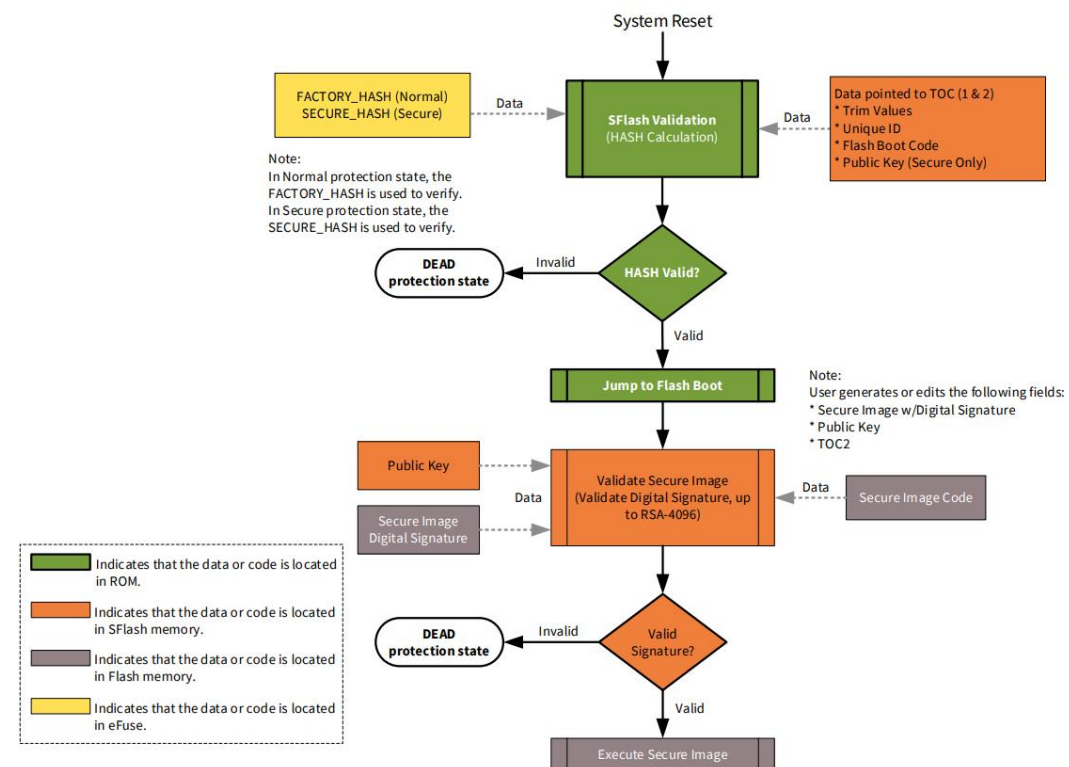
1	OTW固件升级程序原理介绍	3
2	前期环境准备	9
3	Overwrite模式固件升级步骤(I2C)	15
4	Overwrite模式固件升级步骤(UART)	25
5	Overwrite模式固件升级步骤(SPI)	37
6	Swap模式固件升级步骤	48

OTW固件升级程序原理介绍

- ModusToolBox中自带的示例工程**OTW_Firmware_Upgrade**展示了使用Edge protect bootloader程序和DFU(Device Firmware Update) GUI工具对 XMC7000 进行固件升级。它包含了两个应用程序：
 - ▣ Edge protect bootloader程序：它由运行在 CM0+ 内核上的基于 MCUboot 的bootloader程序组成。bootloader程序处理DFU 应用程序image的认证和升级。当image文件有效时，bootloader将使用primary slot的起始地址启动 DFU image。
Flash boot：它处理edge protect bootloader的image的身份验证。当image文件有效时，flash boot将启动bootloader image。
Flash boot的启动可以配置为在secure模式（安全保护状态）或normal模式（正常保护状态）下运行：
 - ✓ 正常保护状态：flash boot将启动bootloader程序而不进行身份验证
 - ✓ 安全保护状态：flash boot在认证成功后将启动bootloader程序。如果发现bootloader程序image无效或已损坏，设备将进入dead protection状态，并一直保持在dead protection状态，直到设备被重置。
 - ▣ DFU 应用程序：它由运行在 CM7 内核上的基于 DFU 中间件库的 DFU 应用程序组成。该 DFU 应用程序通过串行接口下载映像并将其存储在secondary slot（flash memory）中。
DFU 主机工具：这是一款GUI工具，可通过串行接口将image文件发送至设备。
此应用程序可以通过以下方式之一进行构建：
 - ✓ BOOT模式：应用程序Image被build并烧录到primary slot中。bootloader程序将在下次复位时直接启动该应用程序。
 - ✓ UPGRADE模式：应用程序image被build并烧录到secondary slot中。根据用户输入，引导加载程序将在下次复位时将image复制到primary slot并启动它。

OTW固件升级程序原理介绍

- 下图展示了 CM0+ 操作如何从复位开始。复位后，CM0+ 从 ROM BOOT开始执行。ROM BOOT会验证 SFlash。验证完成后，执行跳转到FLASH BOOT，并根据保护状态配置 DAP。请注意颜色编码，它描绘了数据和代码所在的内存类型。FLASH BOOT程序随后会验证 TOC2 中列出的第一个应用程序，如果验证通过则跳转到其入口点。在安全保护状态下，第一个用户应用程序是安全映像。安全映像配置硬件以保护系统后，如有需要，会验证主用户应用程序。如果发现 SFlash 或安全映像无效或已损坏，设备将进入“dead”protection state，并一直保持在“dead”protection state，直到设备被重置。



OTW固件升级程序原理介绍

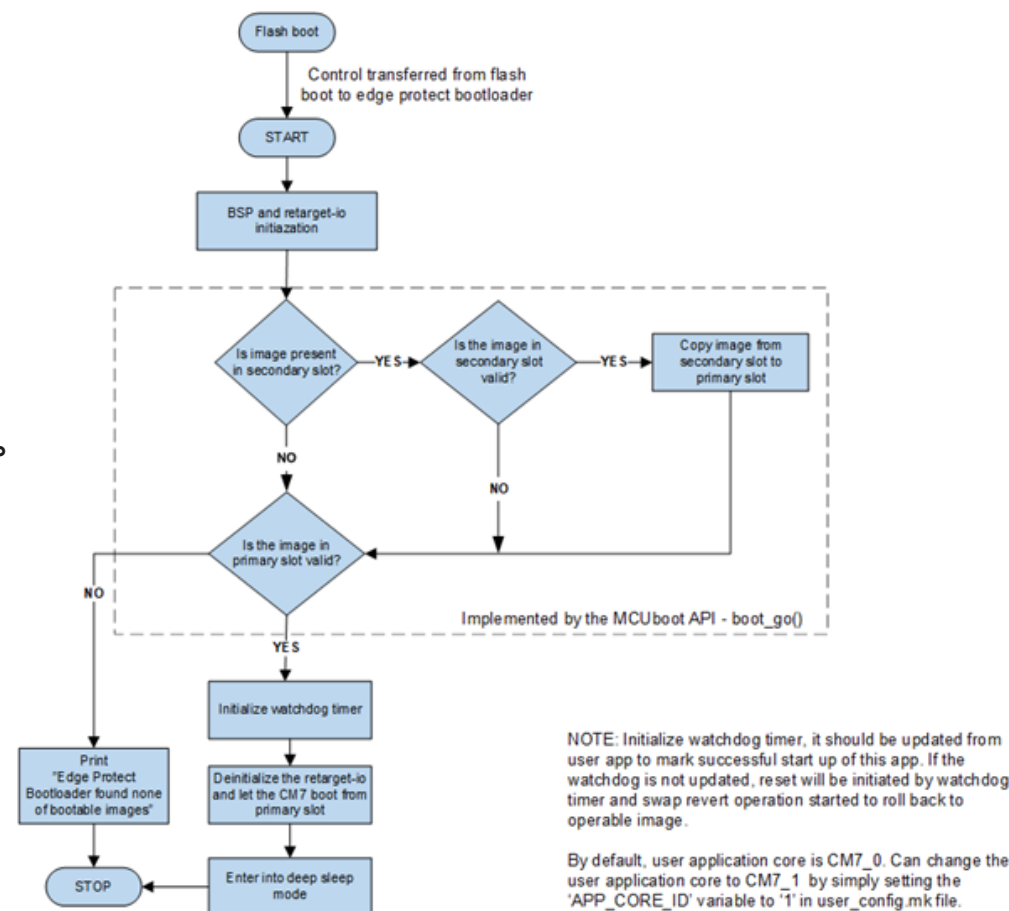
- MCUboot 的工作原理是将flash按每个image分为两个槽：primary slot和secondary slot。在生产过程中，第一个版本的应用程序被编程到primary slot中。设备中运行的固件更新 (DFU) 应用程序通过 I2C、UART、SPI 或 CAN FD 串行通信接口接收UPGRADE image，并将其放入secondary slot。这种基于槽的分区有助于对primary slot进行读/写保护，防止权限较低的应用程序访问primary slot。
- MCUboot 始终从primary slot启动，并在请求升级时将secondary slot中的image复制到primary slot中。升级可以是基于overwrite的，也可以是基于swap的。在基于overwrite的升级中，primary slot中的image将丢失，如果新image出现问题，则无法回滚。在基于swap的升级中，image在两个slot之间交换，并且可以回滚。在这种情况下，MCUboot 会利用flash中名为 "scratch area "的额外区域进行可靠的交换。XMC7000 的 MCUboot 支持基于swap的upgrade和基于overwrite的upgrade。

OTW固件升级程序原理介绍

- 对于基于swap的升级，每个image slot都包含 MCUboot 用于确定当前状态和当前boot 运行期间要采取的行动的元数据。如果是upgrade image，应用程序会更新 img_ok 字段，以便将当前image永久保存在primary slot中。MCUboot 可实现复位恢复，并在操作过程中发生复位或断电时恢复复制操作。下图显示了支持edge protection的bootloader程序的执行流程。

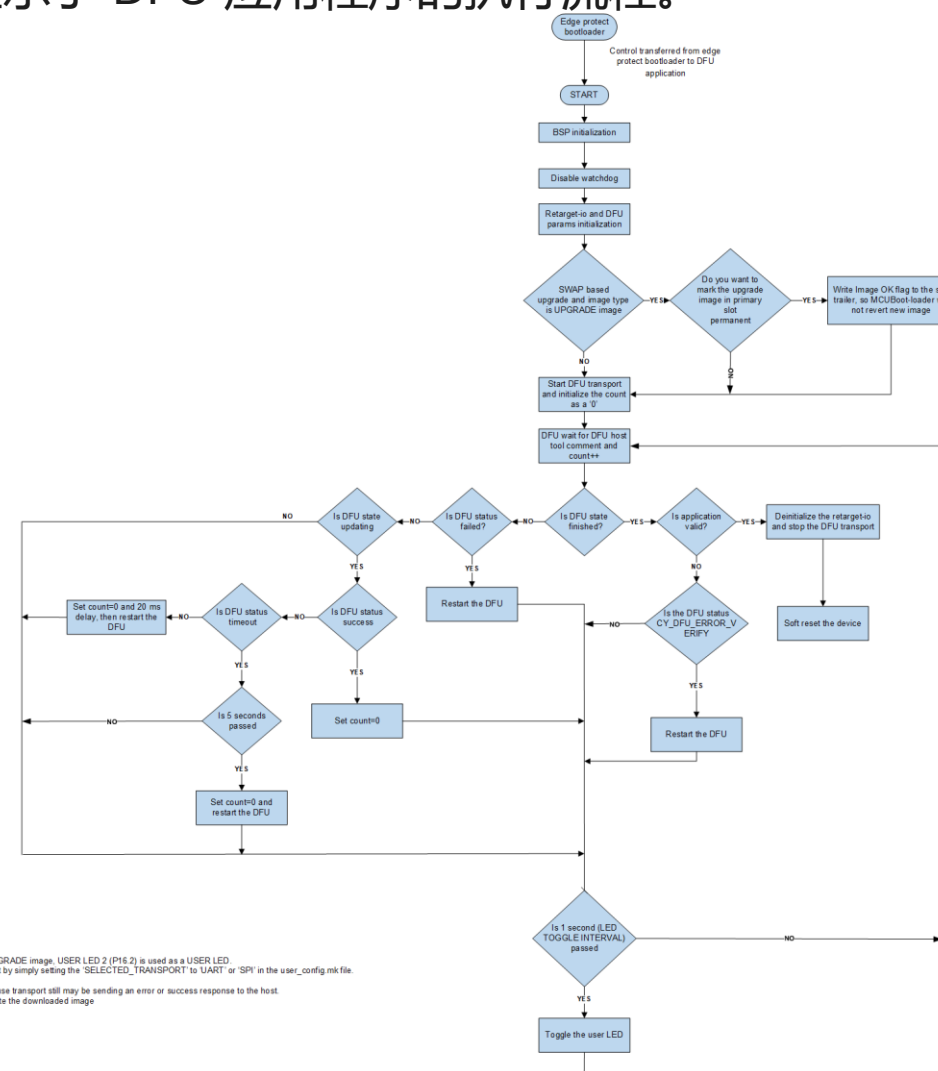
- XMC7000 设备支持两种类型的升级过程。

- 对于基于overwrite的升级，在成功验证后，secondary image将被复制 primary slot。如果secondary image无法运行，则无法恢复升级。
- 对于基于swap的升级，primary slot和secondary slot中的image会被交换。如果secondary image无法确认其操作，则可以恢复升级。



OTW固件升级程序原理介绍

- DFU 应用程序演示了基于 DFU 中间件库的 DFU 操作。下图显示了 DFU 应用程序的执行流程。
- DFU 应用程序支持与 DFU 主机工具通信的 I2C、UART、SPI 和 CAN FD 接口。你可以根据使用情况更改这些默认配置。



NOTE: For BOOT image, USER LED 1 (P16.1) is used as a USER LED. For UPGRADE image, USER LED 2 (P16.2) is used as a USER LED.
By default I2C is used as a DFU transport. Can change the DFU transport by simply setting the 'SELECTED_TRANSPORT' to 'UART' or 'SPI' in the user_config.mk file.
For UART DFU transport, if DFU updating status is unknown, Delay because transport still may be sending an error or success response to the host.
So reset the device, to give a control to edge protect bootloader. To validate the downloaded image

Table of contents

1	OTW固件升级程序原理介绍	3
2	前期环境准备	9
3	Overwrite模式固件升级步骤(I2C)	15
4	Overwrite模式固件升级步骤(UART)	25
5	Overwrite模式固件升级步骤(SPI)	37
6	Swap模式固件升级步骤	48

前期环境准备

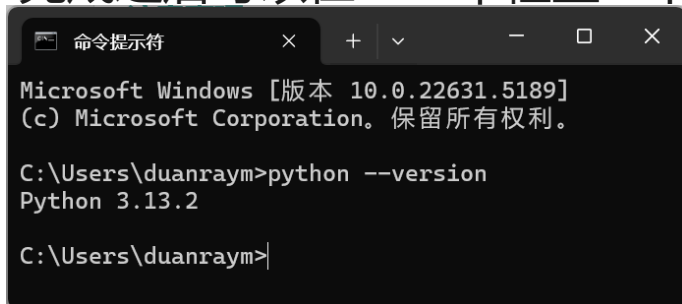
- 安装的ModusToolBox 3.4
- 安装的时候需要有**CySecureTools**这个模块。
- 需要安装**Python 3.8.10**以上版本。
 - Python下载链接如下：
<https://www.python.org/downloads/>
- **KIT_XMC72_EVK或KIT_XMC71_EVK_LITE_V1**开发板
 - 其相关资料介绍可以从以下链接获取：
https://www.infineon.com/cms/en/product/evaluation-boards/kit_xmc72_evk/?redirId=217005
https://www.infineon.com/cms/en/product/evaluation-boards/kit_xmc71_evk_lite_v1/
- **Miniprogr4**烧录调试器
 - Miniprogr4用于通过UART、SPI接口来进行固件升级，其相关资料介绍可以从以下链接获取：
<https://www.infineon.com/cms/cn/product/evaluation-boards/cy8ckit-005-a/>
- **DFU Host Tool**:一个基于图形用户界面(GUI)的工具，它会通过串行接口将需要升级的固件发送到设备端。

注意事项

- 下载好安装包之后按照指导进行安装，**但需要在下图中的步骤勾选'Add python.exe to PATH'**，如果不勾选这个选项，在后面使用Cmd语句的时候会自动搜索的Python 27的版本。其他的选项可以保持默认。



- 完成之后可以在Cmd中检查一下Python的版本



注意事项

- 安装完成之后，用Cmd定位到mtb_shared的mcuboot目录，找到scripts文件夹：

```
命令提示符
C:\Users\duanraym>cd C:\Users\duanraym\mtw_3.4\mtb_shared\mcuboot\v1.9.1-cypress\scripts
C:\Users\duanraym\mtw_3.4\mtb_shared\mcuboot\v1.9.1-cypress\scripts>
```

- 下载目录下requirement文件里对应的工具，用以下语句：

python -m pip install -r requirements.txt

如果是都满足了就会出现下图的情况，如果缺少某个工具则会自动下载。

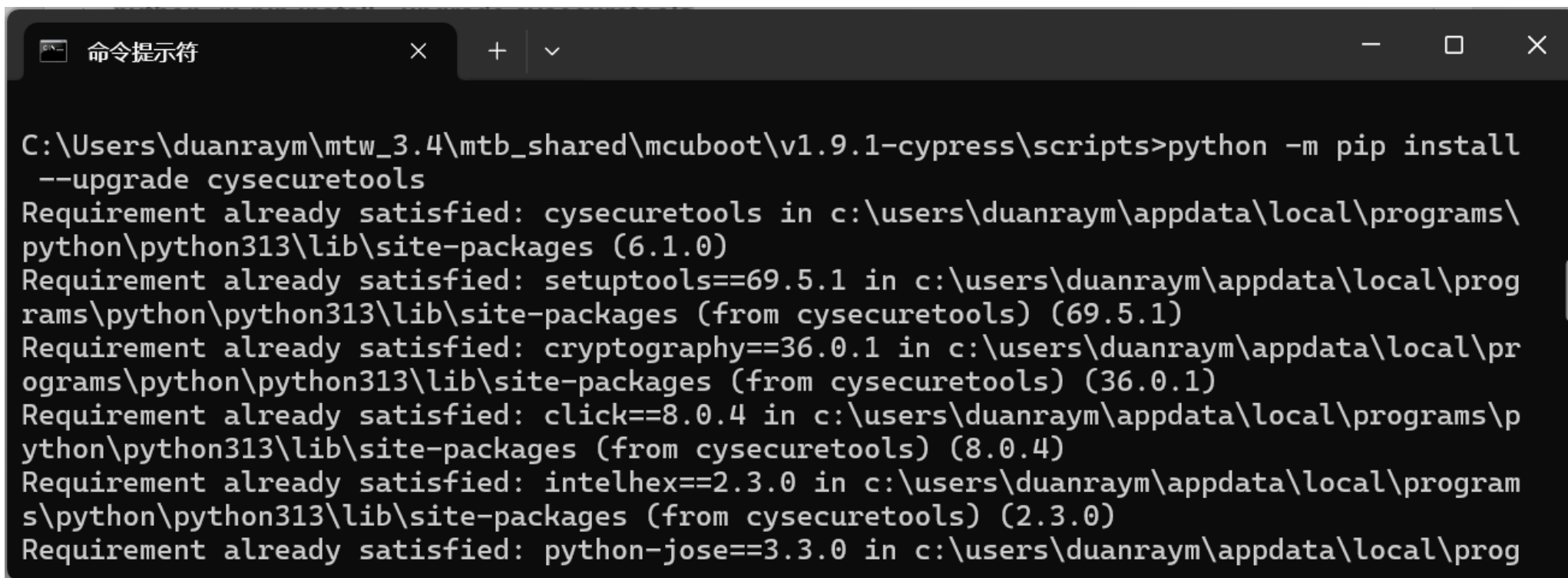
```
命令提示符
C:\Users\duanraym\mtw_3.4\mtb_shared\mcuboot\v1.9.1-cypress\scripts>python -m pip install -r requirements.txt
Requirement already satisfied: cryptography>=2.6 in c:\users\duanraym\appdata\local\programs\python\python313\lib\site-packages (from -r requirements.txt (line 1)) (36.0.1)
Requirement already satisfied: intelhex in c:\users\duanraym\appdata\local\programs\python\python313\lib\site-packages (from -r requirements.txt (line 2)) (2.3.0)
Requirement already satisfied: click in c:\users\duanraym\appdata\local\programs\python\python313\lib\site-packages (from -r requirements.txt (line 3)) (8.0.4)
Requirement already satisfied: cbor2 in c:\users\duanraym\appdata\local\programs\python\python313\lib\site-packages (from -r requirements.txt (line 4)) (5.6.5)
Requirement already satisfied: cffi>=1.12 in c:\users\duanraym\appdata\local\programs\python\python313\lib\site-packages (from cryptography>=2.6->-r requirements.txt (line 1)) (1.17.1)
```

- 安装完成后，点击new application,创建一个OTA_over_Ethernet_MQTT工程，否则，后续创建固件升级的工程可能会失败。

注意事项

- 检查CySecureTools的版本是否满足要求，在Cmd中输入以下语句：

python -m pip install --upgrade cysecuretools



```

命令提示符
C:\Users\duanraym\mtw_3.4\mtb_shared\mcuboot\v1.9.1-cypress\scripts>python -m pip install --upgrade cysecuretools
Requirement already satisfied: cysecuretools in c:\users\duanraym\appdata\local\programs\python\python313\lib\site-packages (6.1.0)
Requirement already satisfied: setuptools==69.5.1 in c:\users\duanraym\appdata\local\programs\python\python313\lib\site-packages (from cysecuretools) (69.5.1)
Requirement already satisfied: cryptography==36.0.1 in c:\users\duanraym\appdata\local\programs\python\python313\lib\site-packages (from cysecuretools) (36.0.1)
Requirement already satisfied: click==8.0.4 in c:\users\duanraym\appdata\local\programs\python\python313\lib\site-packages (from cysecuretools) (8.0.4)
Requirement already satisfied: intelhex==2.3.0 in c:\users\duanraym\appdata\local\programs\python\python313\lib\site-packages (from cysecuretools) (2.3.0)
Requirement already satisfied: python-jose==3.3.0 in c:\users\duanraym\appdata\local\prog
  
```

Demo路径

- 在new application后，选择BSP:KIT_XMC72_EVK，找到以下红色标记的工程即为固件升级demo


Select Application - Project Creator 2.30

Settings

Help

Application(s) Root Path:

C:/Users/duanraym/mtw_3.4

Browse...

Target IDE:

Eclipse IDE for ModusToolbox™

Enter filter text...

Browse for Application...

Template Application	New Application Name	New BSP Name
> Community Code Examples		
> Getting Started		
☐ Empty App		
☐ FreeRTOS Blinky		
☐ Hello World		
☐ Multicore Empty App		
☒ OTW Firmware Upgrade	OTW Firmware Upgrade 1	APP_KIT_XMC72_EVK
☐ Switching Power Modes		
☐ XMC7000 OOB Demo		

This example demonstrates a firmware upgrade for the XMC7000 using the edge protect bootloader and DFU.

For more details, see the [README on GitHub](#).

Summary:

BSP: KIT_XMC72_EVK

Template Application(s): OTW Firmware Upgrade

Application(s) Root Path: C:/Users/duanraym/mtw_3.4

Press "Create" to create the selected application(s).

< Back

Create

Close

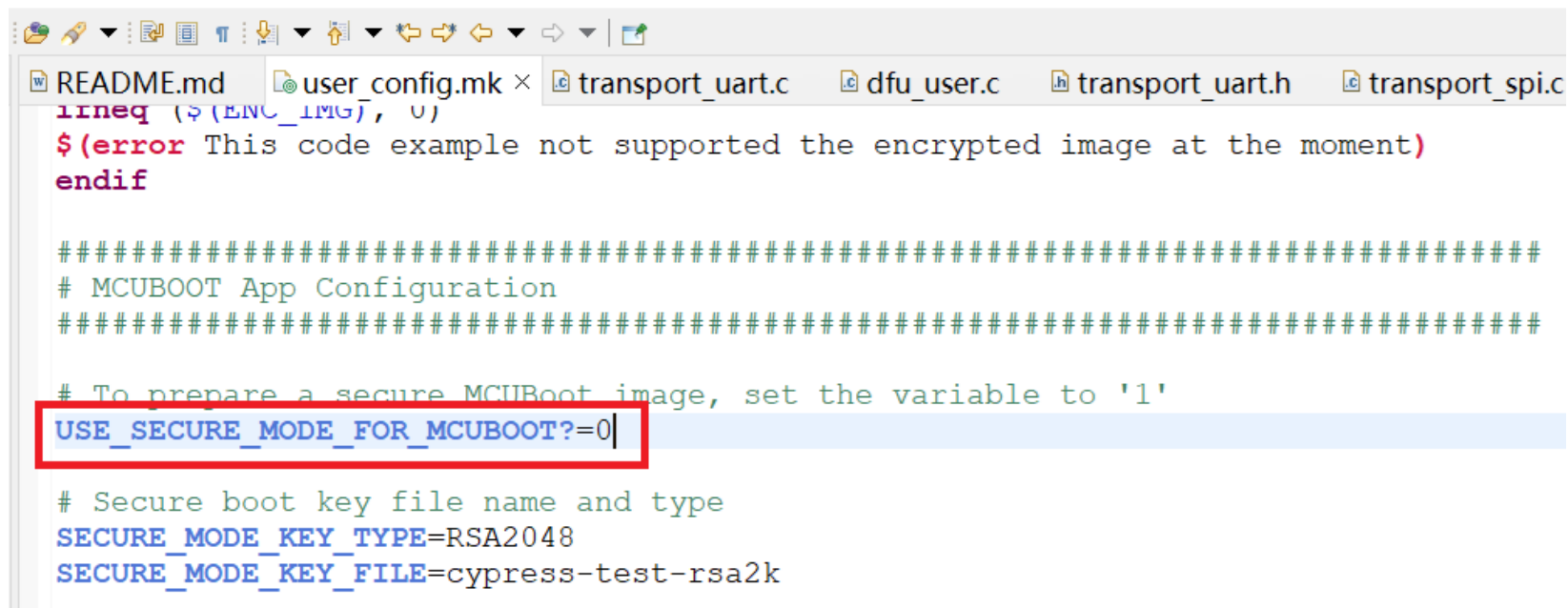
Table of contents

1	OTW固件升级程序原理介绍	3
2	前期环境准备	9
3	Overwrite模式固件升级步骤(I2C)	15
4	Overwrite模式固件升级步骤(UART)	25
5	Overwrite模式固件升级步骤(SPI)	37
6	Swap模式固件升级步骤	48

Overwrite模式固件升级步骤(I2C)

- 默认情况下，protection state被设置为Secure mode。对于身份验证，在pre-build阶段，此代码示例将使用默认的 cypress-test-rsa2k.pub 密钥自动生成 cy_si_key.c 文件。在post-build阶段，此代码示例将使用默认的 cypress-test-rsa2k.pem 密钥对映像进行签名以自动生成安全映像。默认的 cypress-test-rsa2k 密钥位于 <application>/keys 文件夹中。要将模式更改为normal mode，请在<application>/user_config.mk文件中将 USE_SECURE_MODE_FOR_MCUBOOT 变量设置为 '0'。

ModusToolbox™ 3.4



```

README.md  user_config.mk  transport_uart.c  dfu_user.c  transport_uart.h  transport_spi.c
lineq ($ (ENC_IMG), 0)
$(error This code example not supported the encrypted image at the moment)
endif

#####
# MCUBOOT App Configuration
#####

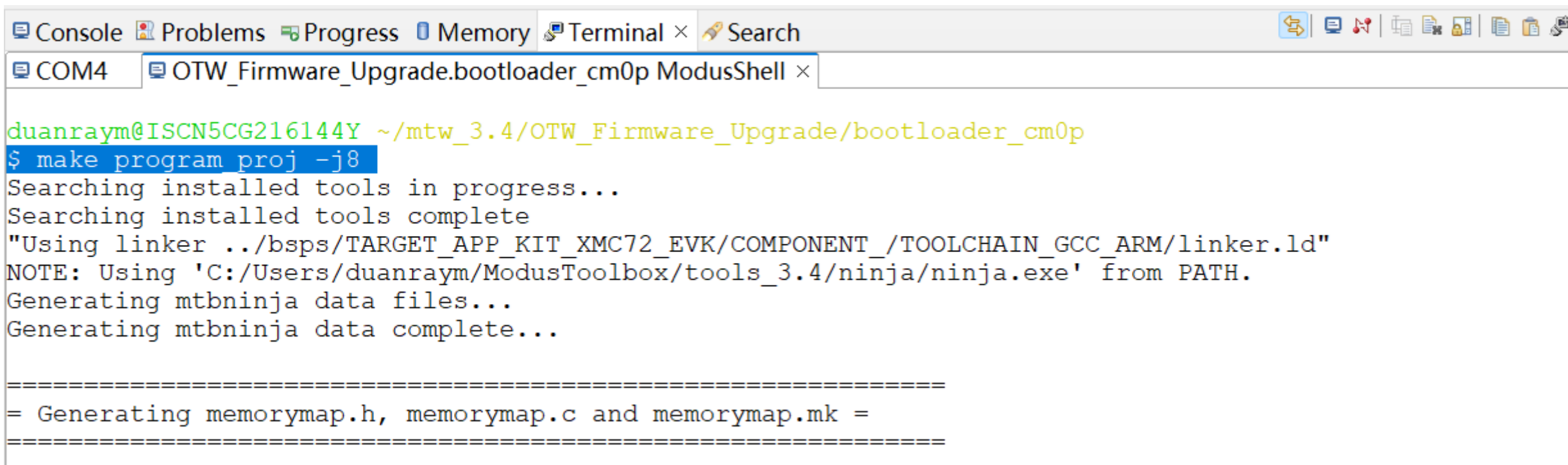
# To prepare a secure MCUBoot image, set the variable to '1'
USE_SECURE_MODE_FOR_MCUBOOT?=0

# Secure boot key file name and type
SECURE_MODE_KEY_TYPE=RSA2048
SECURE_MODE_KEY_FILE=cypress-test-rsa2k

```

Overwrite模式固件升级步骤(I2C)

- 选择工程中的"OTW_Firmware_Upgrade.bootloader_cm0p"应用，打开对应的terminal,输入以下指令
make program_proj -j8 来对bootloader工程进行编译并下载到XMC72 EVAL KIT



```

Console Problems Progress Memory Terminal × Search
COM4 OTW_Firmware_Upgrade.bootloader_cm0p ModusShell ×

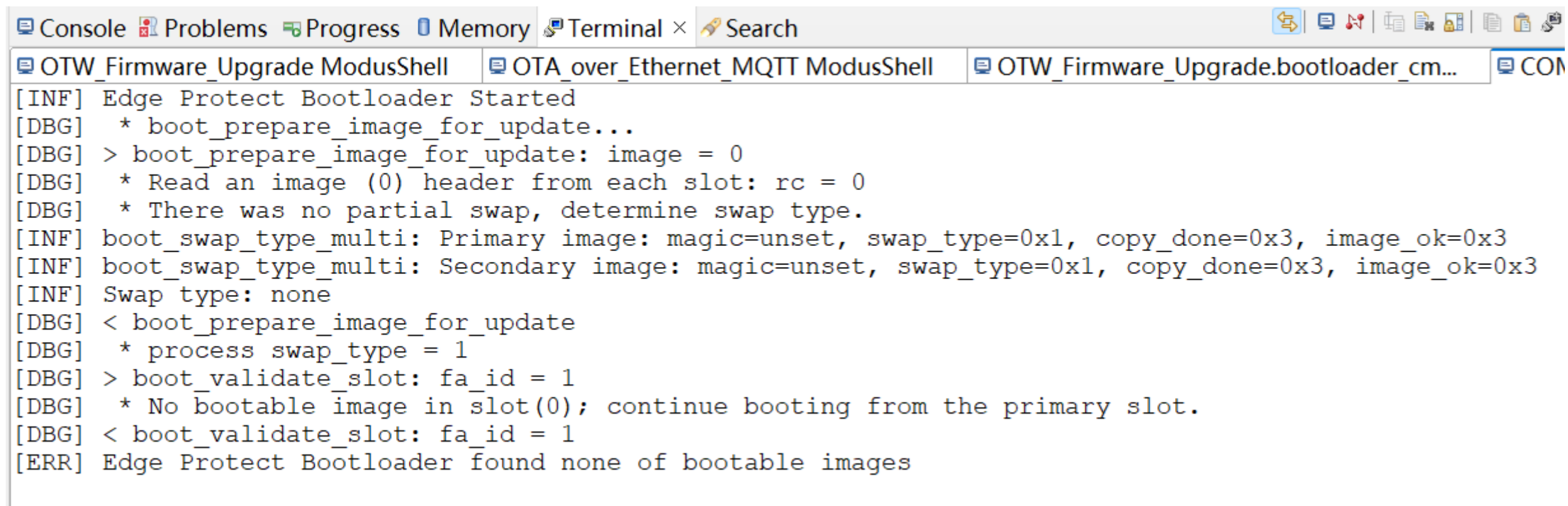
duanraym@ISCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade/bootloader_cm0p
$ make program_proj -j8
Searching installed tools in progress...
Searching installed tools complete
"Using linker ../bsps/TARGET_APP_KIT_XMC72_EVK/COMPONENT_/TOOLCHAIN_GCC_ARM/linker.ld"
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...

=====
= Generating memorymap.h, memorymap.c and memorymap.mk =
=====

```

Overwrite模式固件升级步骤(I2C)

- 烧录完成后，打开串口工具有出现如下提示信息：

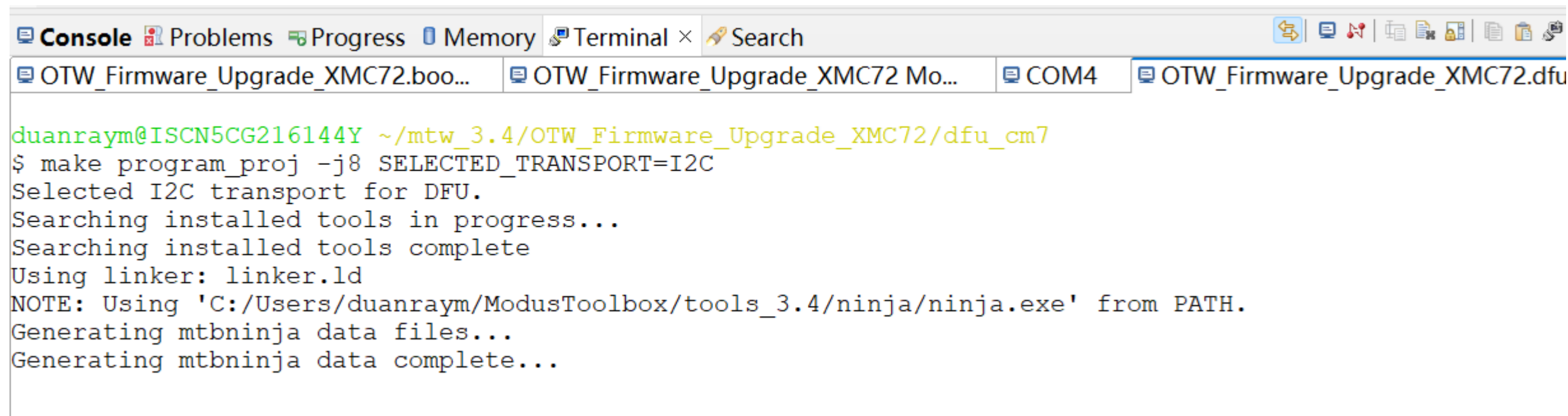


The screenshot shows a ModusShell terminal window with the following tabs: OTW_Firmware_Upgrade ModusShell, OTA_over_Ethernet_MQTT ModusShell, OTW_Firmware_Upgrade.bootloader_cm..., and COM. The terminal output is as follows:

```
[INF] Edge Protect Bootloader Started
[DBG] * boot_prepare_image_for_update...
[DBG] > boot_prepare_image_for_update: image = 0
[DBG] * Read an image (0) header from each slot: rc = 0
[DBG] * There was no partial swap, determine swap type.
[INF] boot_swap_type_multi: Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] boot_swap_type_multi: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Swap type: none
[DBG] < boot_prepare_image_for_update
[DBG] * process swap_type = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] * No bootable image in slot(0); continue booting from the primary slot.
[DBG] < boot_validate_slot: fa_id = 1
[ERR] Edge Protect Bootloader found none of bootable images
```

Overwrite模式固件升级步骤(I2C)

- 选择工程中的”OTW_Firmware_Upgrade. dfu_cm7”应用，打开对应的terminal,输入以下指令
make program_proj -j8 SELECTED_TRANSPORT=I2C 来对处于BOOT mode的DFU application工程进行 编译并下载到KIT_XMC72_EVK



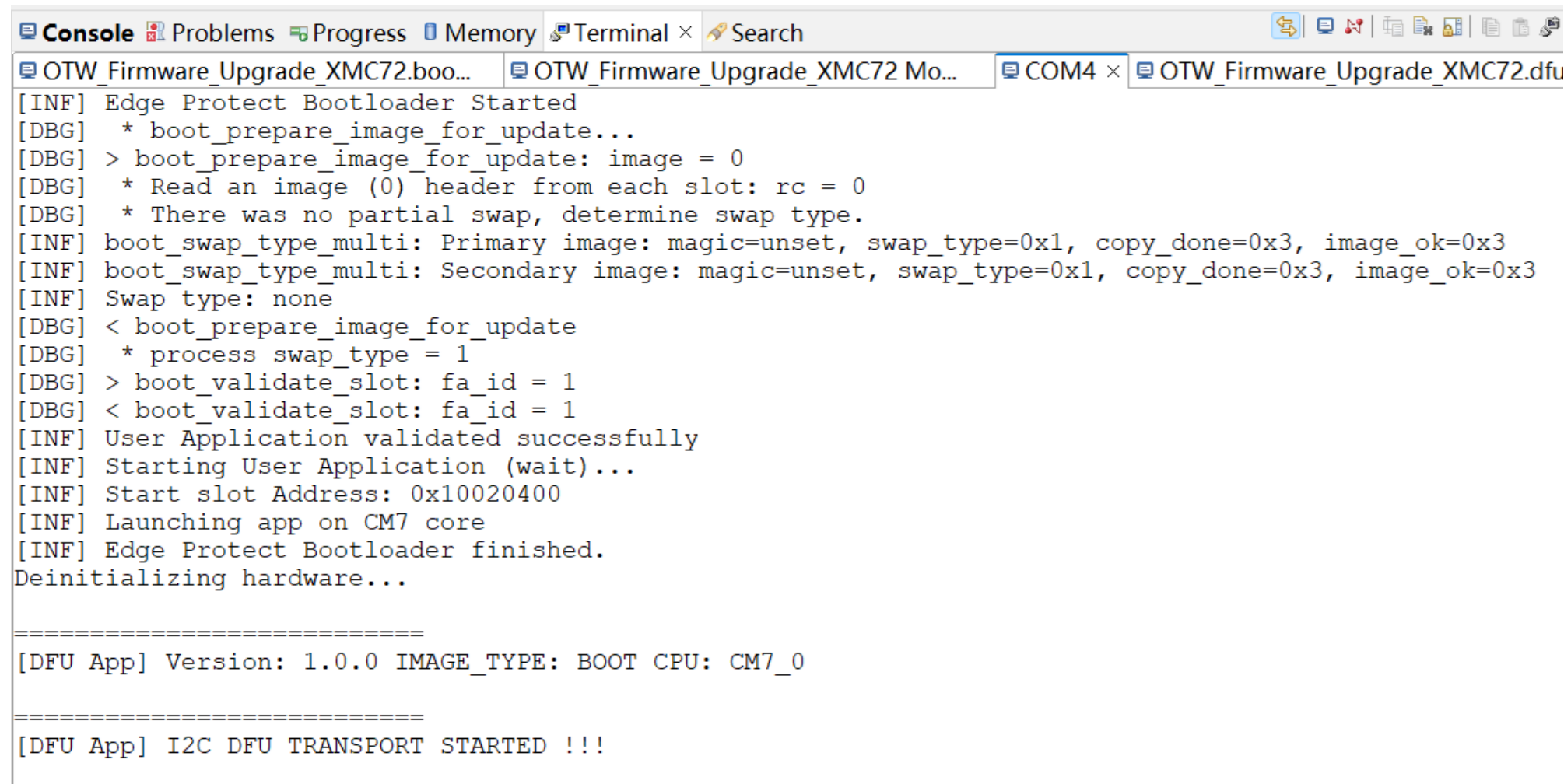
```

Console Problems Progress Memory Terminal × Search
OTW_Firmware_Upgrade_XMC72.bo... OTW_Firmware_Upgrade_XMC72 Mo... COM4 OTW_Firmware_Upgrade_XMC72.dfu

duanraym@ISCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade_XMC72/dfu_cm7
$ make program_proj -j8 SELECTED_TRANSPORT=I2C
Selected I2C transport for DFU.
Searching installed tools in progress...
Searching installed tools complete
Using linker: linker.ld
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...
  
```

Overwrite模式固件升级步骤(I2C)

- 应用下载完成后，打开串口工具，会出现以下信息：



```

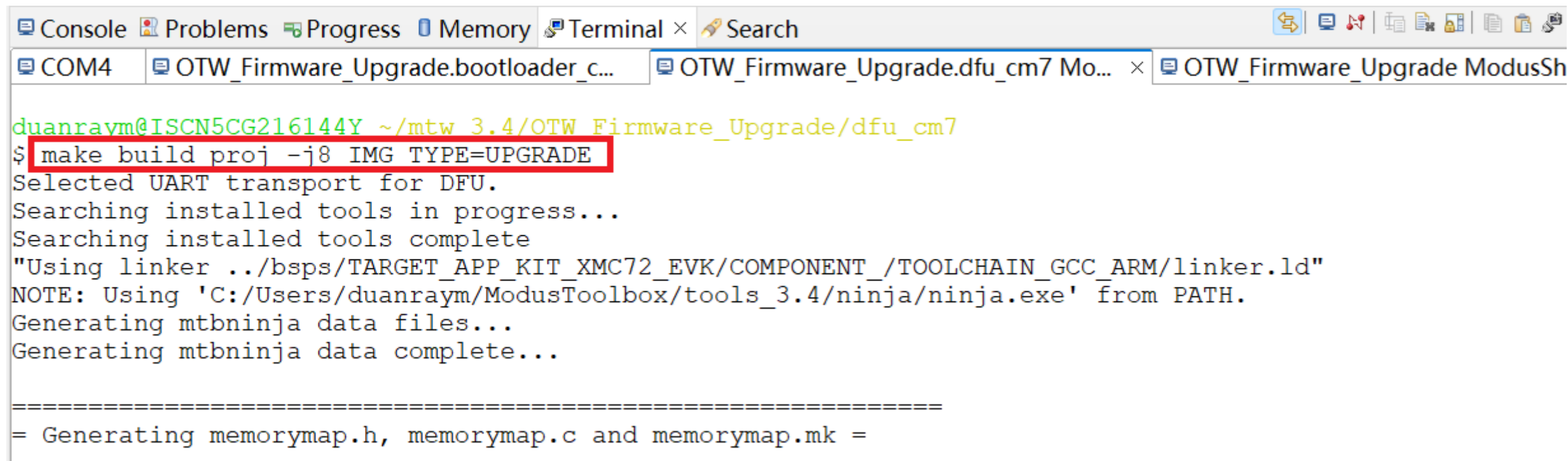
Console Problems Progress Memory Terminal × Search
OTW_Firmware_Upgrade_XMC72.bo... OTW_Firmware_Upgrade_XMC72 Mo... COM4 × OTW_Firmware_Upgrade_XMC72.dfu
[INF] Edge Protect Bootloader Started
[DBG] * boot_prepare_image_for_update...
[DBG] > boot_prepare_image_for_update: image = 0
[DBG] * Read an image (0) header from each slot: rc = 0
[DBG] * There was no partial swap, determine swap type.
[INF] boot_swap_type_multi: Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] boot_swap_type_multi: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Swap type: none
[DBG] < boot_prepare_image_for_update
[DBG] * process swap_type = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[INF] User Application validated successfully
[INF] Starting User Application (wait)...
[INF] Start slot Address: 0x10020400
[INF] Launching app on CM7 core
[INF] Edge Protect Bootloader finished.
Deinitializing hardware...

=====
[DFU App] Version: 1.0.0 IMAGE_TYPE: BOOT CPU: CM7_0

=====
[DFU App] I2C DFU TRANSPORT STARTED !!!
  
```

Overwrite模式固件升级步骤(I2C)

- 选择工程中的"OTW_Firmware_Upgrade.dfu_cm7"应用, 打开对应的terminal,输入以下指令
make build proj -j8 IMG_TYPE=UPGRADE



```

Console Problems Progress Memory Terminal × Search
COM4 OTW_Firmware_Upgrade.bootloader_c... OTW_Firmware_Upgrade.dfu_cm7 Mo... × OTW_Firmware_Upgrade ModusSh

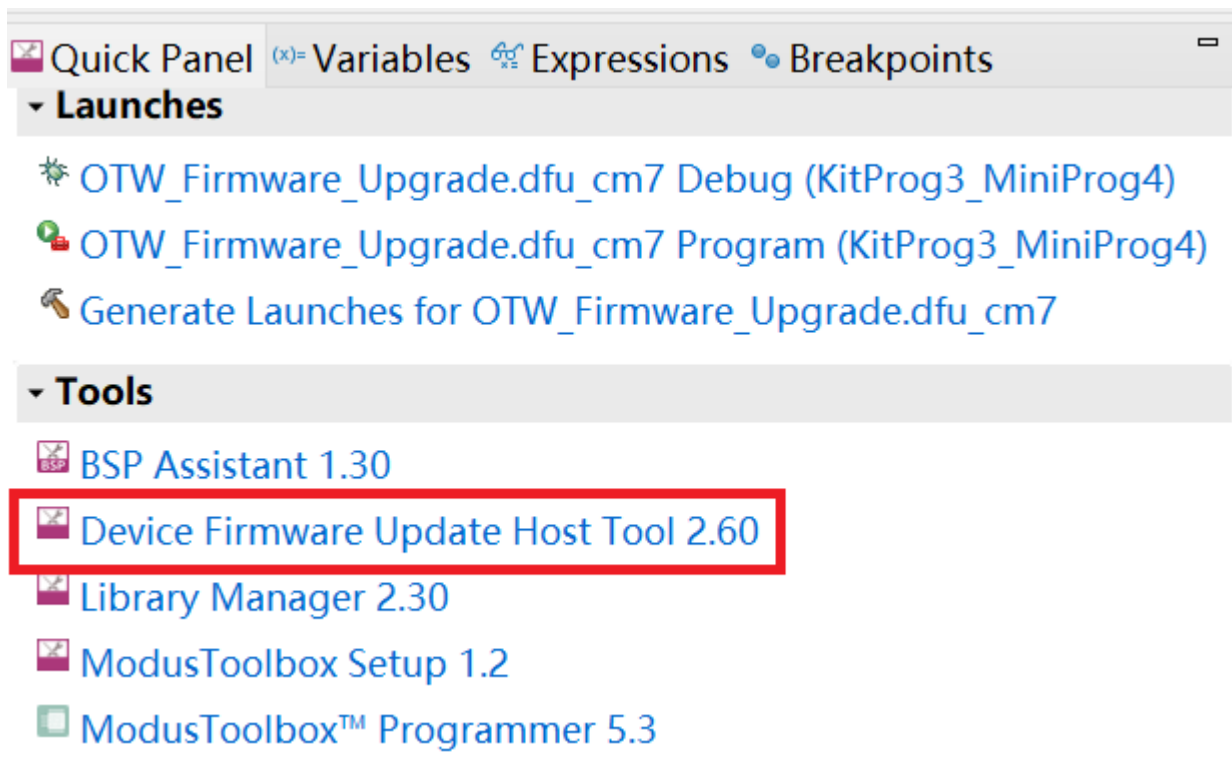
duanraym@ISCN5CG216144Y ~/mtw 3.4/OTW_Firmware_Upgrade/dfu_cm7
$ make build proj -j8 IMG_TYPE=UPGRADE
Selected UART transport for DFU.
Searching installed tools in progress...
Searching installed tools complete
"Using linker ../bsps/TARGET_APP_KIT_XMC72_EVK/COMPONENT_/TOOLCHAIN_GCC_ARM/linker.ld"
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...

=====
= Generating memorymap.h, memorymap.c and memorymap.mk =

```

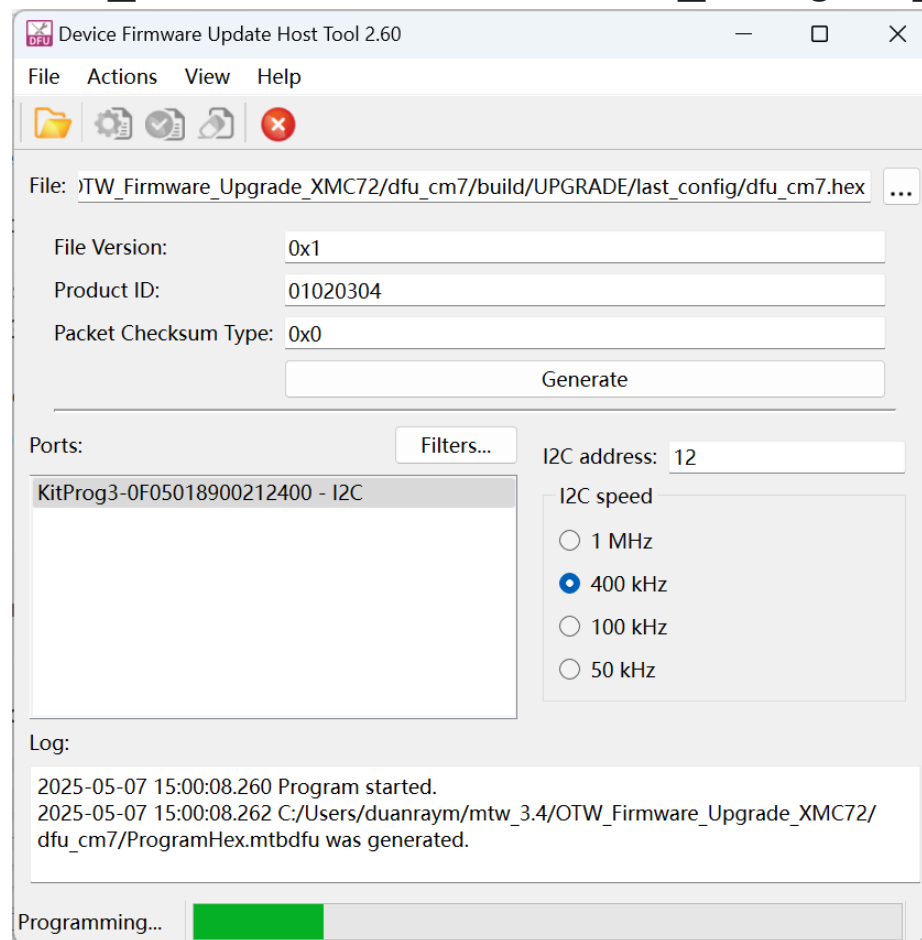
Overwrite模式固件升级步骤(I2C)

- 打开下图红色标记所示的DFU工具。



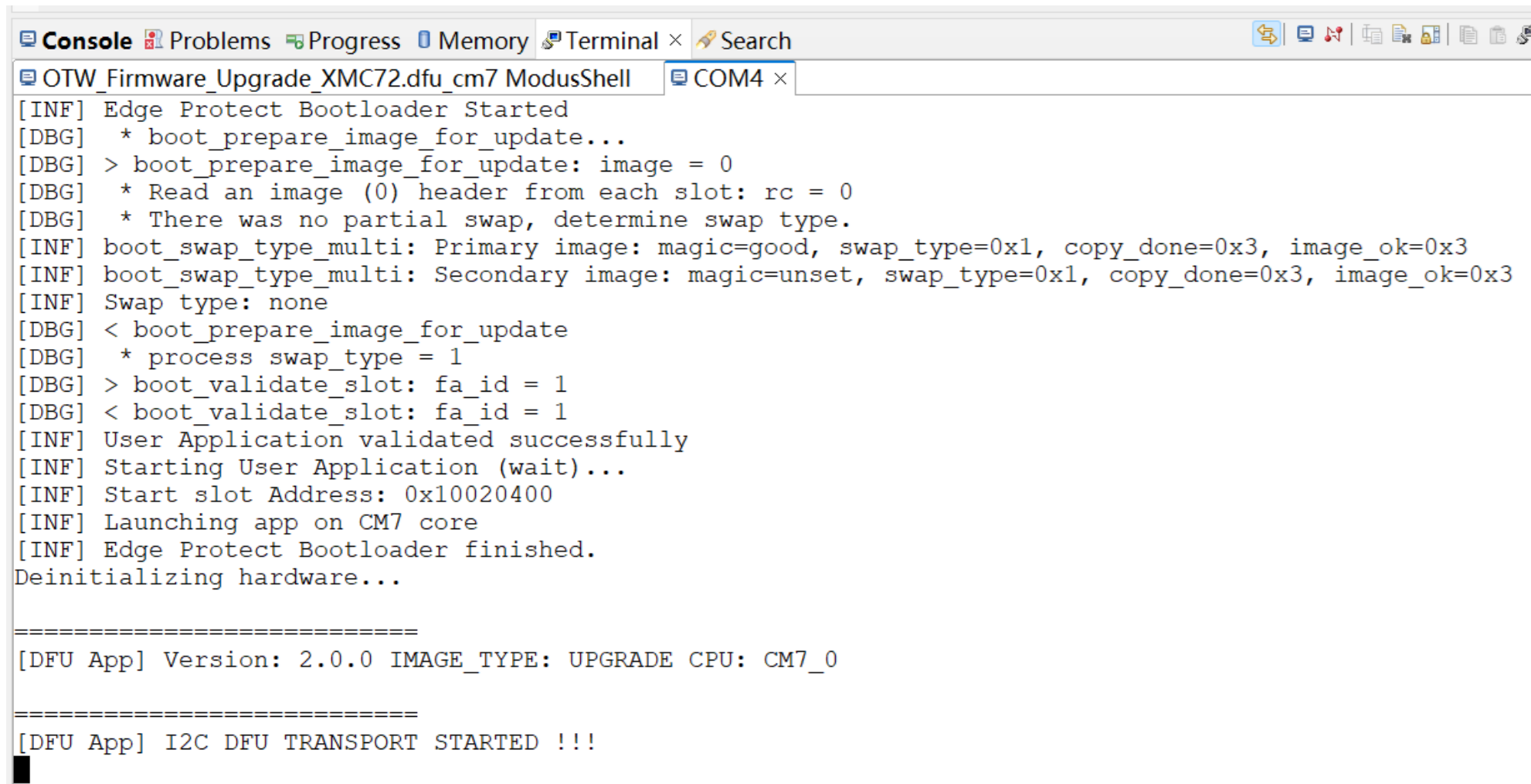
Overwrite模式固件升级步骤(I2C)

- 选择KitProg3 I2C端口，分别设置I2C Address为12， I2C speed为400KHz。hex文件路径:<OTW_Firmware_Upgrade>dfu_cm7\build\UPGRADE\last_config\dfu_cm7.hex，最后点击program进行固件升级。



Overwrite模式固件升级步骤(I2C)

- 更新完成后，打开串口工具可以看到以下打印信息：



```

Console Problems Progress Memory Terminal × Search
OTW_Firmware_Upgrade_XMC72.dfu_cm7 ModusShell COM4 ×
[INF] Edge Protect Bootloader Started
[DBG] * boot_prepare_image_for_update...
[DBG] > boot_prepare_image_for_update: image = 0
[DBG] * Read an image (0) header from each slot: rc = 0
[DBG] * There was no partial swap, determine swap type.
[INF] boot_swap_type_multi: Primary image: magic=good, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] boot_swap_type_multi: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Swap type: none
[DBG] < boot_prepare_image_for_update
[DBG] * process swap_type = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[INF] User Application validated successfully
[INF] Starting User Application (wait)...
[INF] Start slot Address: 0x10020400
[INF] Launching app on CM7 core
[INF] Edge Protect Bootloader finished.
Deinitializing hardware...

=====
[DFU App] Version: 2.0.0 IMAGE_TYPE: UPGRADE CPU: CM7_0
=====
[DFU App] I2C DFU TRANSPORT STARTED !!!

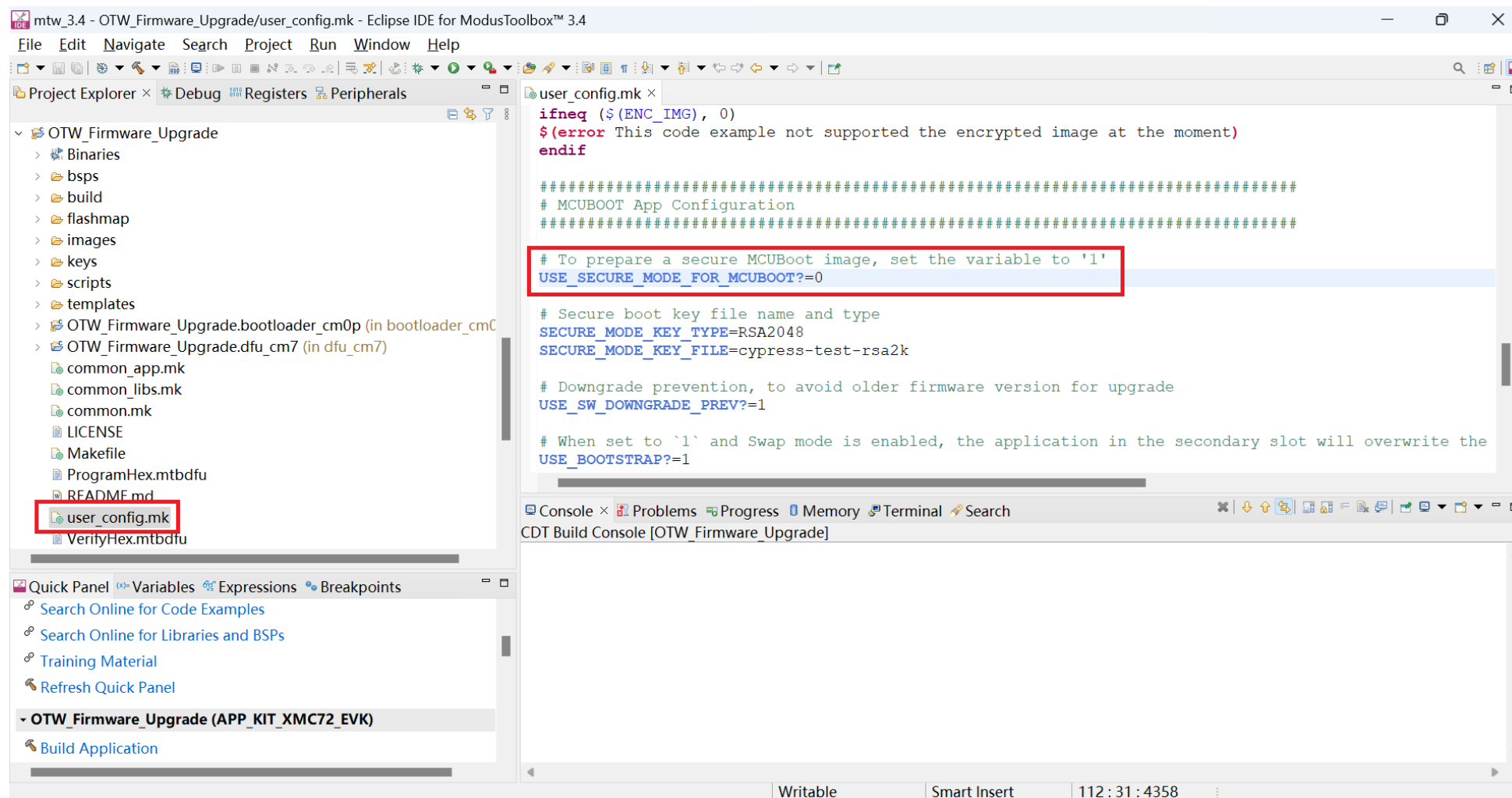
```

Table of contents

1	OTW固件升级程序原理介绍	3
2	前期环境准备	9
3	Overwrite模式固件升级步骤(I2C)	15
4	Overwrite模式固件升级步骤(UART)	25
5	Overwrite模式固件升级步骤(SPI)	37
6	Swap模式固件升级步骤	48

Overwrite模式固件升级步骤(UART)

- 打开user_config.mk文件，将USE_SECURE_MODE_FOR_MCUBOOT设置为0。



Overwrite模式固件升级步骤(UART)

● 设置SELECTED_TRANSPORT?=UART

```

README.md  user_config.mk ×  transport_uart.c  dfu_user.c  transport_uart.h
# when set to 1 and swap mode is enabled, the application in the secondary slot will overwrite the
USE_BOOTSTRAP?=1

#####
# User App Configuration
#####

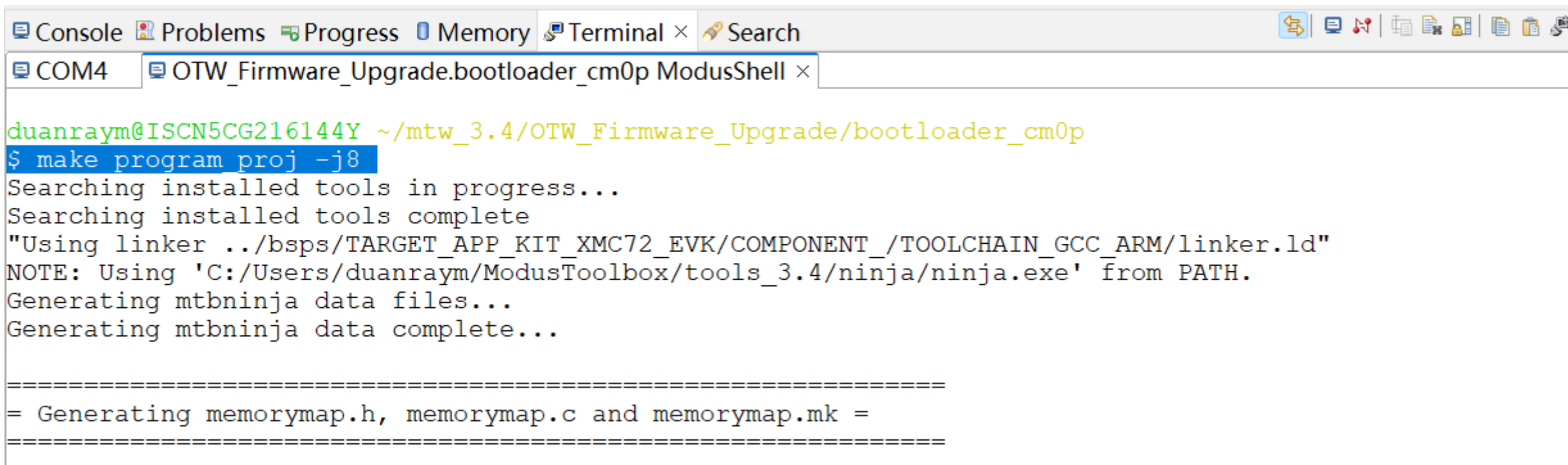
# Default DFU transport set to I2C.
#
# Supported options are I2C, SPI and UART
#
# Select transport here, as required.
SELECTED_TRANSPORT?=UART

# image type can be BOOT or UPGRADE
IMG_TYPES:=BOOT UPGRADE

```

Overwrite模式固件升级步骤(UART)

- 选择工程中的"OTW_Firmware_Upgrade.bootloader_cm0p"应用，打开对应的terminal,输入以下指令
make program_proj -j8 来对bootloader工程进行编译并下载到XMC72 EVAL KIT



```

Console Problems Progress Memory Terminal × Search
COM4 OTW_Firmware_Upgrade.bootloader_cm0p ModusShell ×

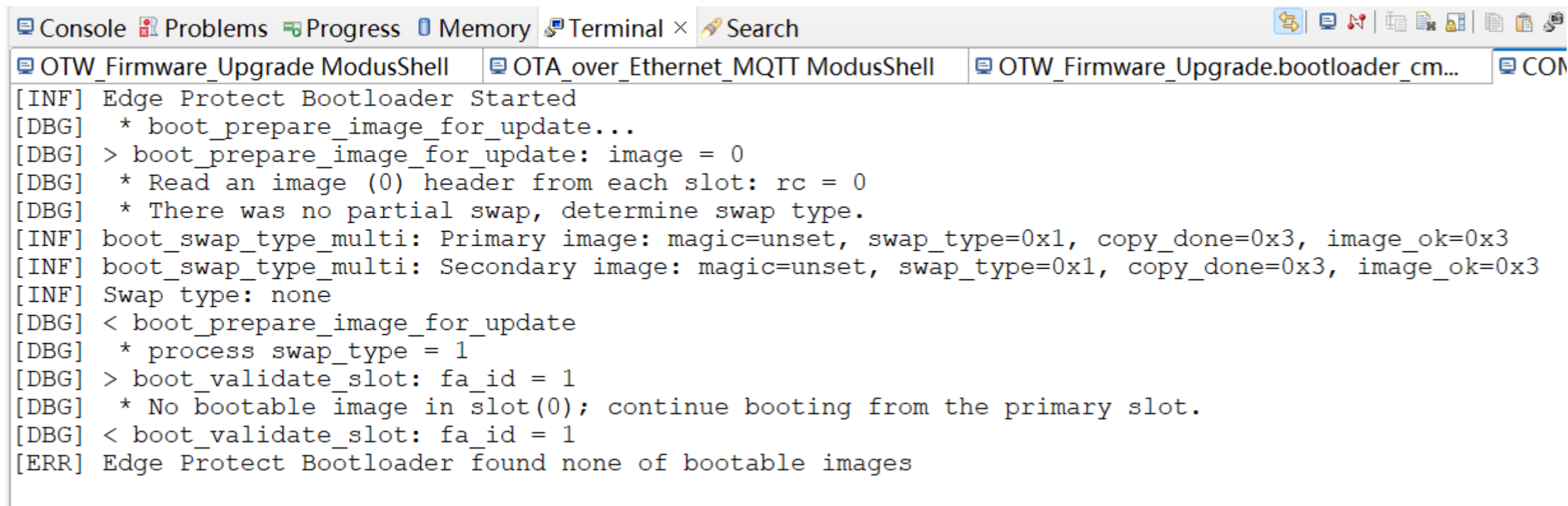
duanraym@ISCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade/bootloader_cm0p
$ make program_proj -j8
Searching installed tools in progress...
Searching installed tools complete
"Using linker ../bsps/TARGET_APP_KIT_XMC72_EVK/COMPONENT_/TOOLCHAIN_GCC_ARM/linker.ld"
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...

=====
= Generating memorymap.h, memorymap.c and memorymap.mk =
=====

```

Overwrite模式固件升级步骤(UART)

- 烧录完成后，打开串口工具有出现如下提示信息：



The screenshot shows a serial terminal window with the following tabs: Console, Problems, Progress, Memory, Terminal, and Search. The active tab is 'Terminal', which displays the output of a ModusShell session. The logs indicate that the Edge Protect Bootloader started, attempted to prepare an image for update, but found no bootable images in the slots, resulting in an error.

```

[INF] Edge Protect Bootloader Started
[DBG] * boot_prepare_image_for_update...
[DBG] > boot_prepare_image_for_update: image = 0
[DBG] * Read an image (0) header from each slot: rc = 0
[DBG] * There was no partial swap, determine swap type.
[INF] boot_swap_type_multi: Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] boot_swap_type_multi: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Swap type: none
[DBG] < boot_prepare_image_for_update
[DBG] * process swap_type = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] * No bootable image in slot(0); continue booting from the primary slot.
[DBG] < boot_validate_slot: fa_id = 1
[ERR] Edge Protect Bootloader found none of bootable images
  
```

- The screenshot displays the Eclipse IDE for ModusToolbox™ 3.4. The Project Explorer on the left shows the project structure, with `transport_uart.c` selected under `COMPONENT_DFU_UART`. The main editor shows the contents of `transport_uart.c`, which defines various UART parameters. The Console at the bottom shows the output of the `make` command, indicating that the program is being built for the `OTW_Firmware_Upgrade.dfu_cm7` target.

Project Explorer:

 - COMPONENT_CAT1
 - COMPONENT_DFU_CANFD
 - COMPONENT_DFU_EMUSB_CDC
 - COMPONENT_DFU_EMUSB_HID
 - COMPONENT_DFU_I2C
 - COMPONENT_DFU_SPI
 - COMPONENT_DFU_UART
 - transport_uart.c**
 - transport_uart.h
 - COMPONENT_DFU_USB_CDC
 - COMPONENT_DFU_USER
 - COMPONENT_CAT2
 - dfu_user.h

transport_uart.c:

```
*/
#ifndef DFU_UART_BAUD
#define DFU_UART_BAUD CYHAL_UART_DEFAULT_BAUD
#endif
#ifndef UART_BYTE_TO_BYTE_TIMEOUT_US
#define UART_BYTE_TO_BYTE_TIMEOUT_US (868U)
#endif
#ifndef DFU_UART_TX
//#define DFU_UART_TX CYDSE_DEBUG_UART_TX
#define DFU_UART_TX P20_4
#endif
#ifndef DFU_UART_RX
//#define DFU_UART_RX CYDSE_DEBUG_UART_RX
#define DFU_UART_RX P20_3
#endif
#ifndef DFU_UART_PARITY
#define DFU_UART_PARITY CYHAL_UART_PARITY_NONE
#endif
#ifndef DFU_UART_DATA_BITS
#define DFU_UART_DATA_BITS (8U)
#endif
```

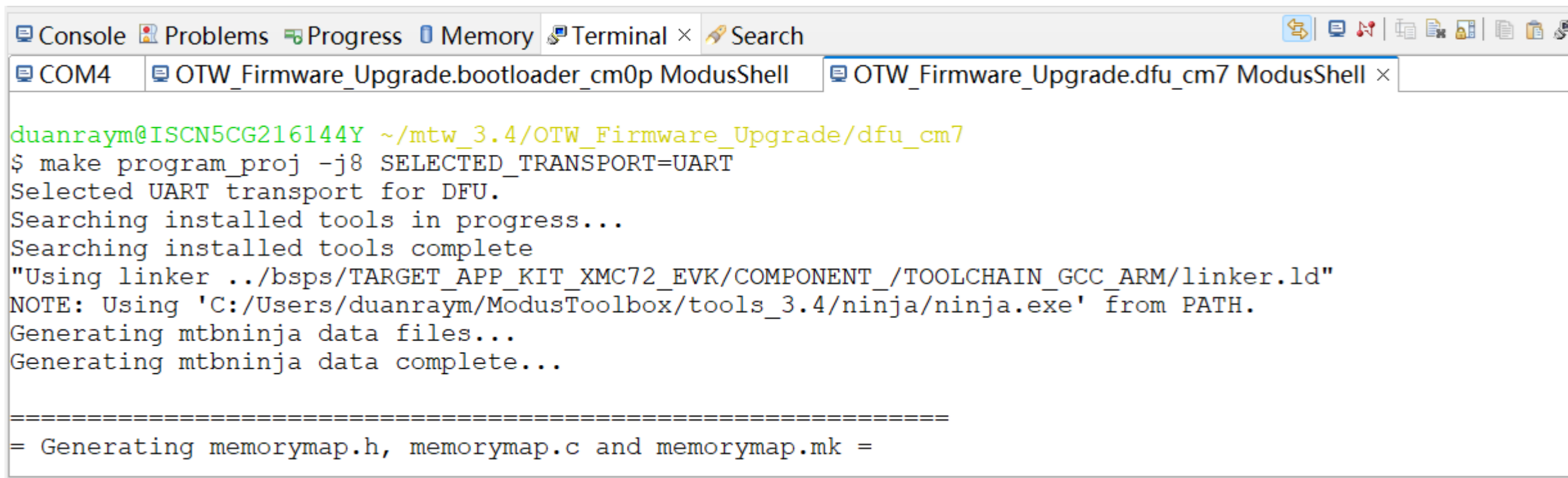
Console:

```
duanraym@ISCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade/dfu_cm7
$ make program_proj -j8 SELECTED_TRANSPORT=UART
Selected UART transport for DFU.
Searching installed tools in progress...
Searching installed tools complete
"Using linker ../bsps/TARGET_APP_KIT_XMC72_EVK/COMPONENT_/TOOLCHAIN_GCC_ARM/linker.ld"
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...

=====
= Generating memorymap.h, memorymap.c and memorymap.mk =
```


Overwrite模式固件升级步骤(UART)

- 选择工程中的"OTW_Firmware_Upgrade.dfu_cm7"应用, 打开对应的terminal, 输入以下指令
make program_proj -j8 来对处于BOOT mode的DFU application工程进行 编译并下载到KIT_XMC72_EVK



```

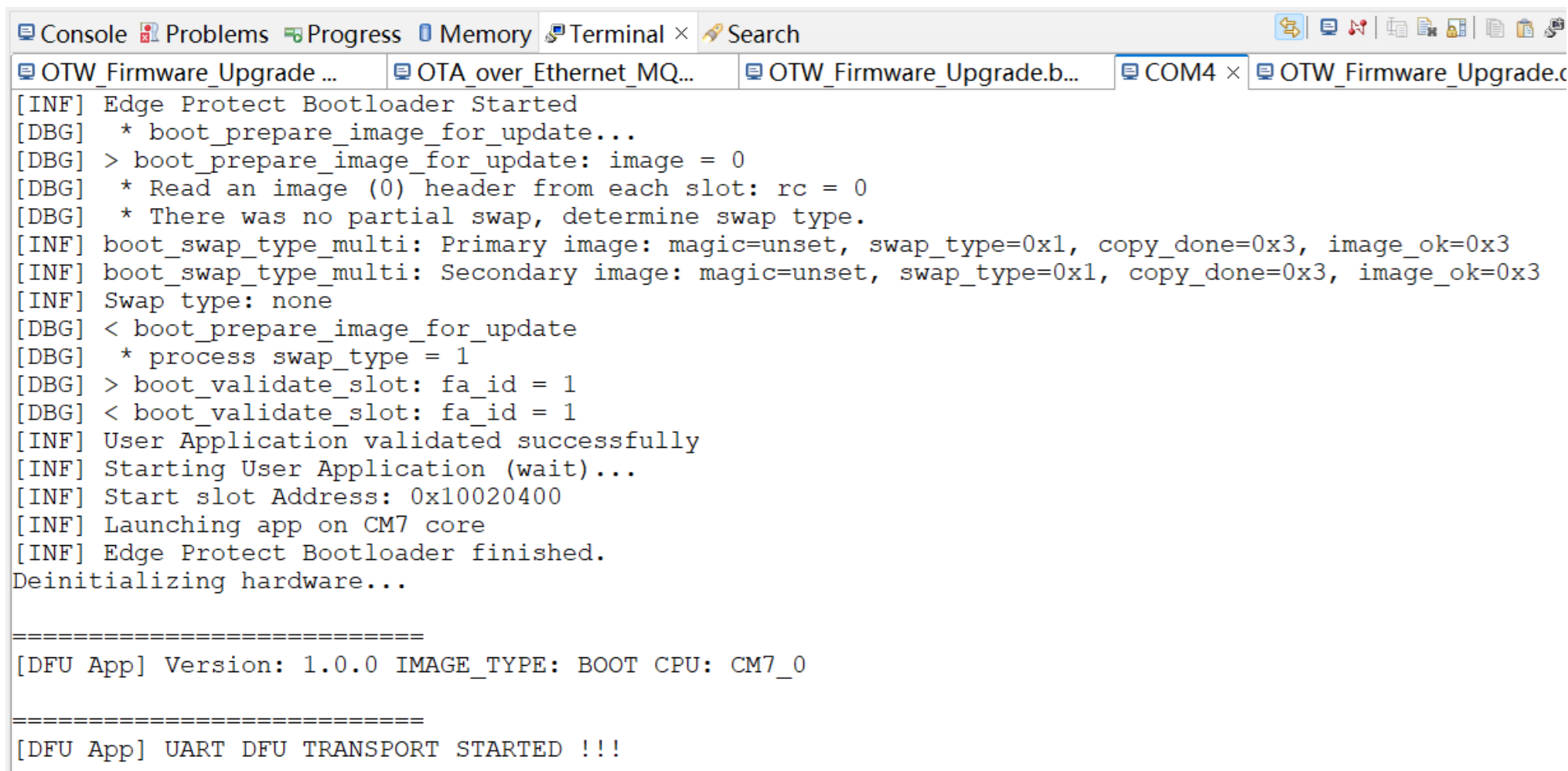
Console Problems Progress Memory Terminal × Search
COM4 OTW_Firmware_Upgrade.bootloader_cm0p ModusShell OTW_Firmware_Upgrade.dfu_cm7 ModusShell ×

duanraym@ISCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade/dfu_cm7
$ make program_proj -j8 SELECTED_TRANSPORT=UART
Selected UART transport for DFU.
Searching installed tools in progress...
Searching installed tools complete
"Using linker ../bsps/TARGET_APP_KIT_XMC72_EVK/COMPONENT_/TOOLCHAIN_GCC_ARM/linker.ld"
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...

=====
= Generating memorymap.h, memorymap.c and memorymap.mk =
  
```


Overwrite模式固件升级步骤(UART)

- 应用下载完成后，打开串口工具，会出现以下信息：



```

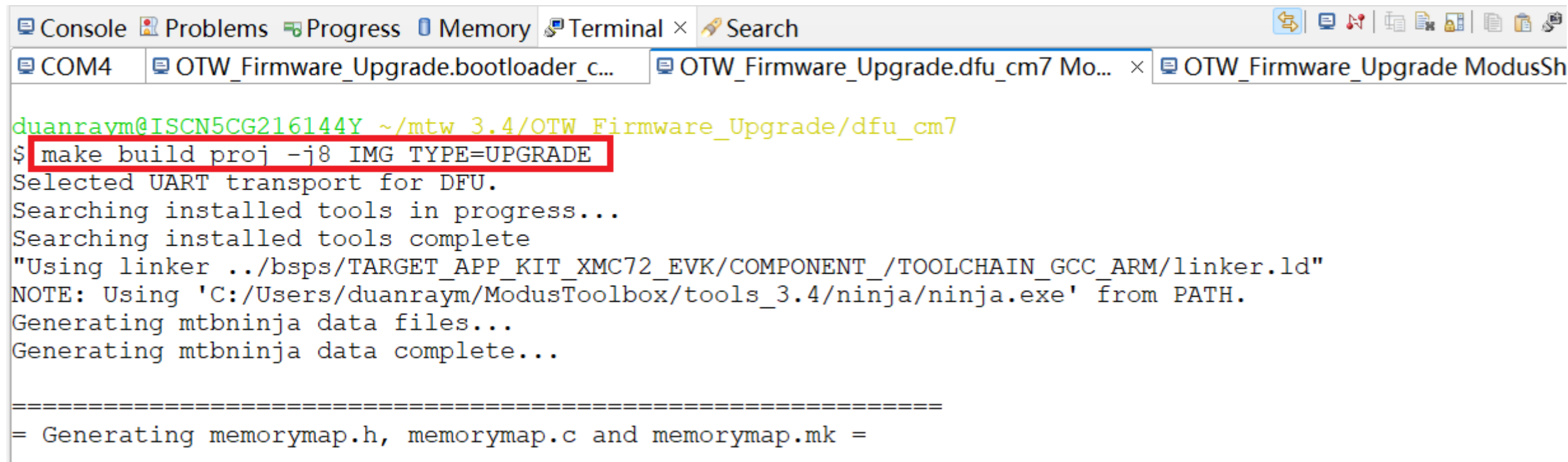
Console Problems Progress Memory Terminal x Search
OTW_Firmware_Upgrade ... OTA_over_Ethernet_MQ... OTW_Firmware_Upgrade.b... COM4 x OTW_Firmware_Upgrade.c
[INF] Edge Protect Bootloader Started
[DBG] * boot_prepare_image_for_update...
[DBG] > boot_prepare_image_for_update: image = 0
[DBG] * Read an image (0) header from each slot: rc = 0
[DBG] * There was no partial swap, determine swap type.
[INF] boot_swap_type_multi: Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] boot_swap_type_multi: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Swap type: none
[DBG] < boot_prepare_image_for_update
[DBG] * process swap_type = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[INF] User Application validated successfully
[INF] Starting User Application (wait)...
[INF] Start slot Address: 0x10020400
[INF] Launching app on CM7 core
[INF] Edge Protect Bootloader finished.
Deinitializing hardware...

=====
[DFU App] Version: 1.0.0 IMAGE_TYPE: BOOT CPU: CM7_0

=====
[DFU App] UART DFU TRANSPORT STARTED !!!
  
```

Overwrite模式固件升级步骤(UART)

- 选择工程中的"OTW_Firmware_Upgrade.dfu_cm7"应用，打开对应的terminal,输入以下指令
make build proj -j8 IMG_TYPE=UPGRADE 来编译需要升级的固件。



```

Console Problems Progress Memory Terminal × Search
COM4 OTW_Firmware_Upgrade.bootloader_c... OTW_Firmware_Upgrade.dfu_cm7 Mo... × OTW_Firmware_Upgrade ModusSh

duanraym@ISCN5CG216144Y ~/mtw 3.4/OTW_Firmware_Upgrade/dfu_cm7
$ make build proj -j8 IMG_TYPE=UPGRADE
Selected UART transport for DFU.
Searching installed tools in progress...
Searching installed tools complete
"Using linker ../bsps/TARGET_APP_KIT_XMC72_EVK/COMPONENT_/TOOLCHAIN_GCC_ARM/linker.ld"
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...

=====
= Generating memorymap.h, memorymap.c and memorymap.mk =
  
```

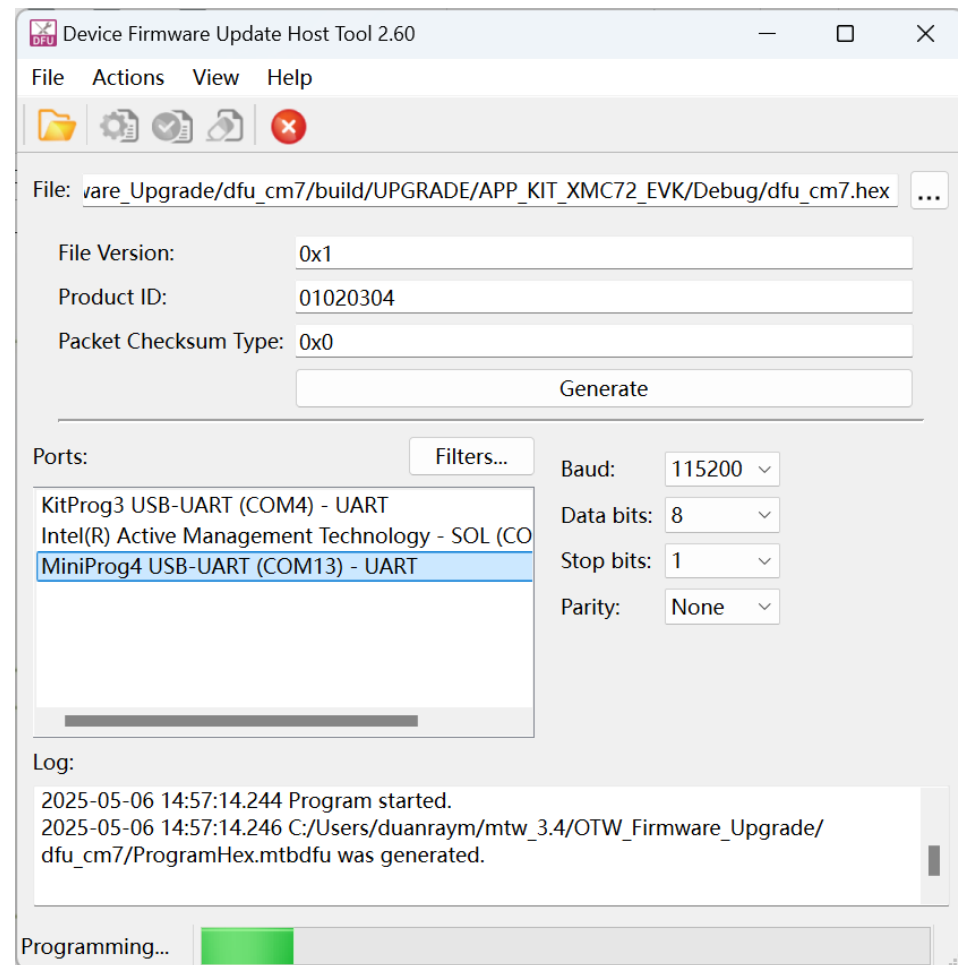
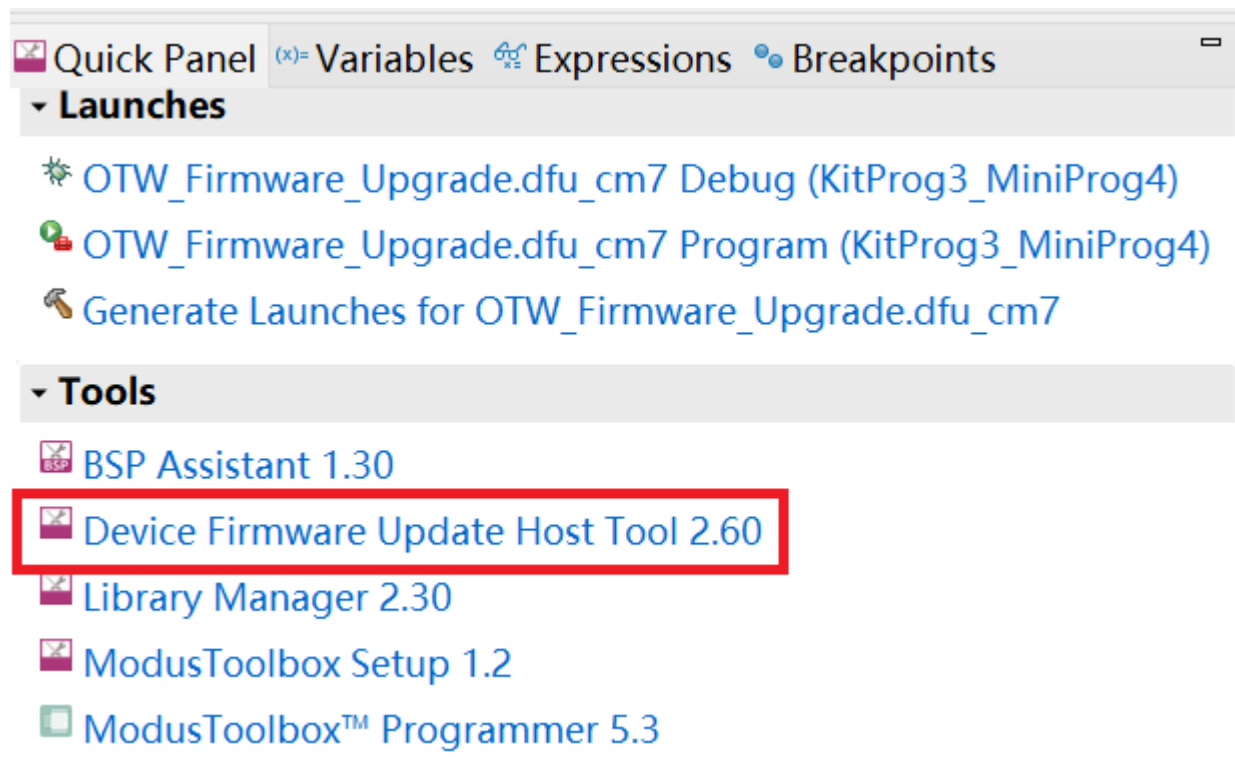
Overwrite模式固件升级步骤(UART)

- 将MiniProg4和KIT_XMC72_EVK的以下几个PIN分别进行连接

MiniProg4	KIT_XMC72_EVK
RX	P20.4
TX	P20.3
VTARG	3.3V
GND	GND

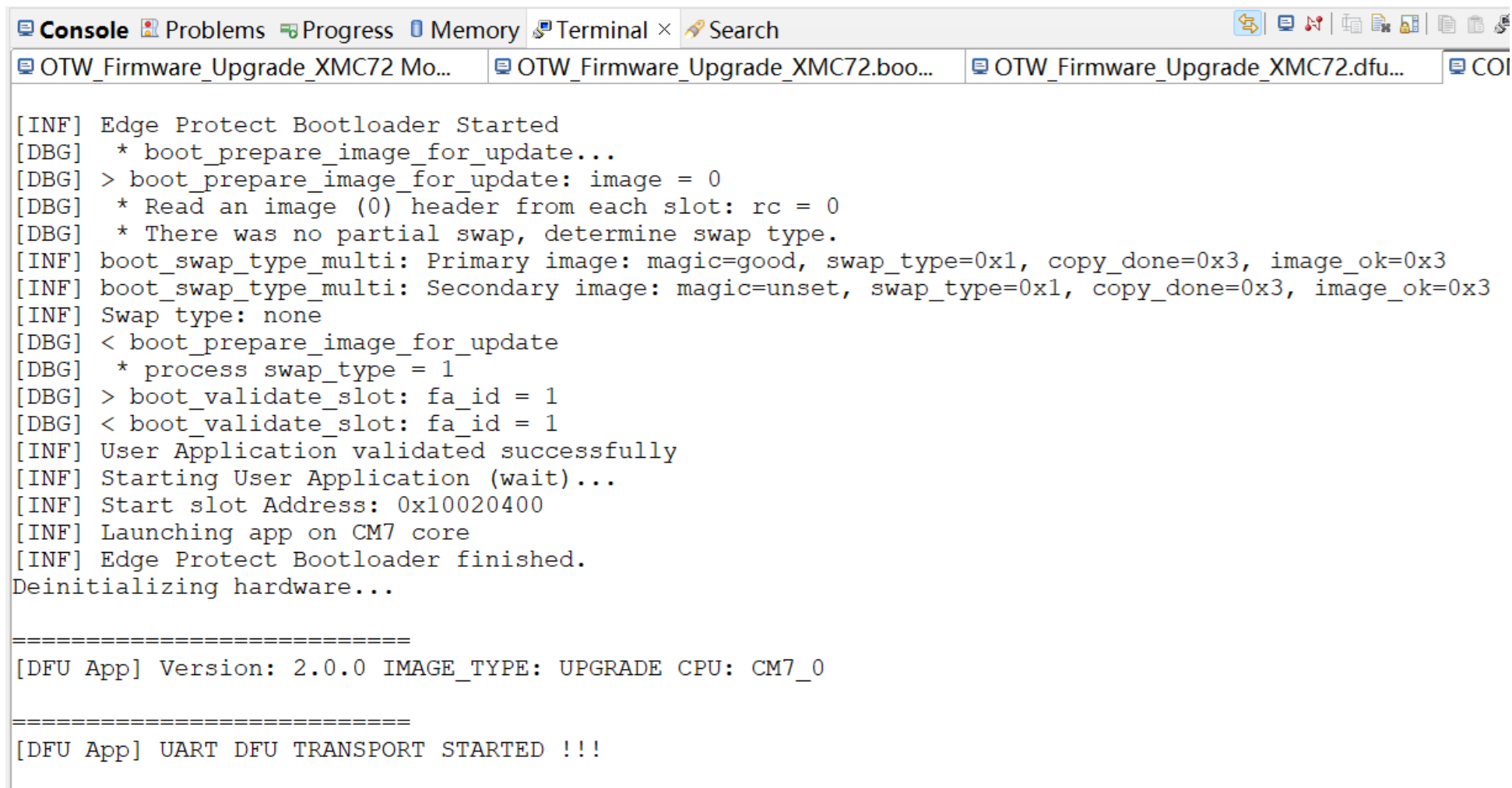
Overwrite模式固件升级步骤(UART)

- 打开下图红色标记所示的DFU工具。然后，选择MiniProg4 USB-UART端口， hex文件路径:<OTW_Firmware_Upgrade>dfu_cm7\build\UPGRADE\last_config\dfu_cm7.hex，最后点击program进行烧录。



Overwrite模式固件升级步骤(UART)

- 升级完成后，打开串口工具，可以观察到以下信息：



```

Console Problems Progress Memory Terminal × Search
OTW_Firmware_Upgrade_XMC72 Mo... OTW_Firmware_Upgrade_XMC72.bo... OTW_Firmware_Upgrade_XMC72.dfu... COI

[INF] Edge Protect Bootloader Started
[DBG] * boot_prepare_image_for_update...
[DBG] > boot_prepare_image_for_update: image = 0
[DBG] * Read an image (0) header from each slot: rc = 0
[DBG] * There was no partial swap, determine swap type.
[INF] boot_swap_type_multi: Primary image: magic=good, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] boot_swap_type_multi: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Swap type: none
[DBG] < boot_prepare_image_for_update
[DBG] * process swap_type = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[INF] User Application validated successfully
[INF] Starting User Application (wait)...
[INF] Start slot Address: 0x10020400
[INF] Launching app on CM7 core
[INF] Edge Protect Bootloader finished.
Deinitializing hardware...

=====
[DFU App] Version: 2.0.0 IMAGE_TYPE: UPGRADE CPU: CM7_0

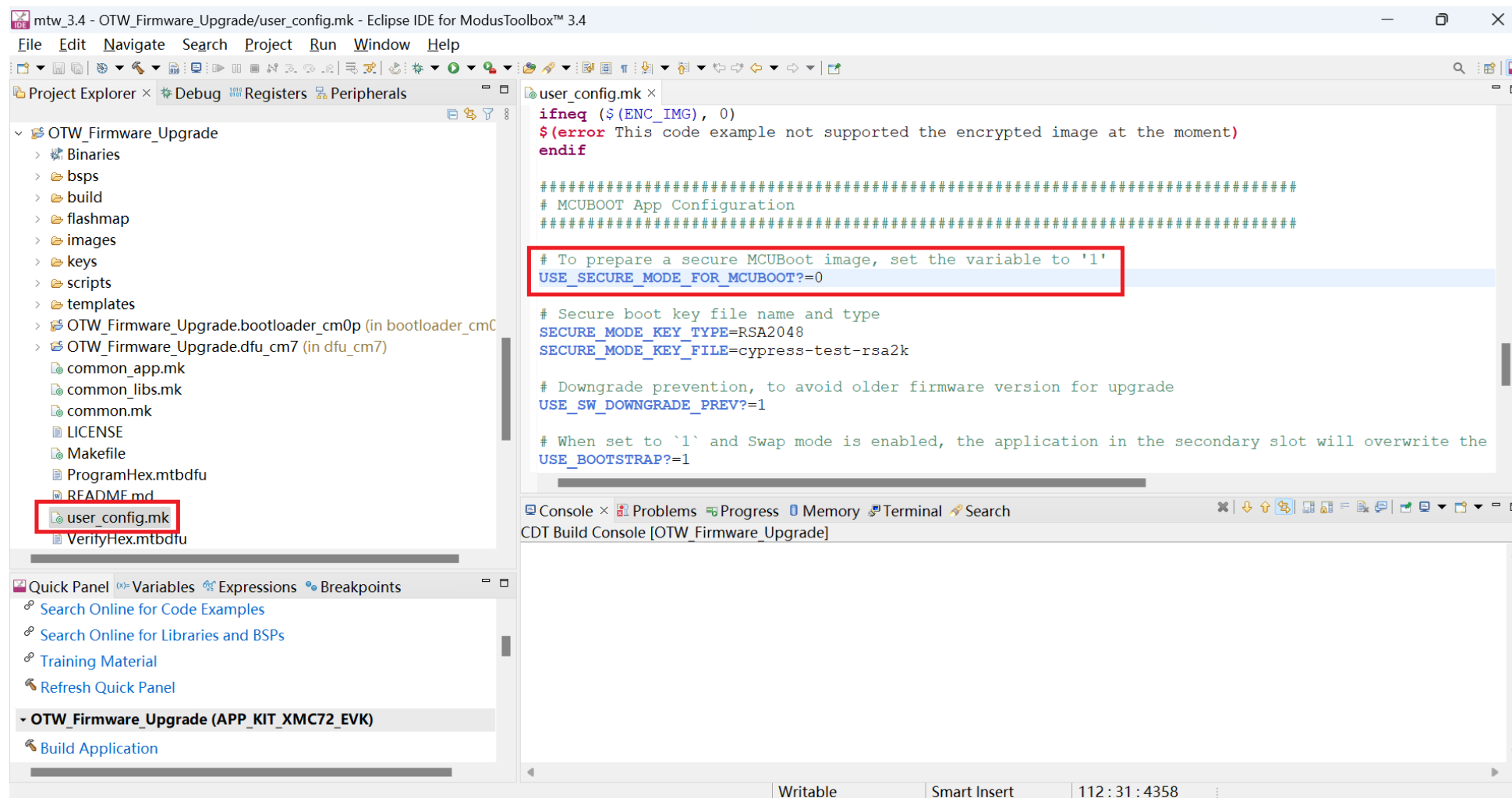
=====
[DFU App] UART DFU TRANSPORT STARTED !!!
  
```

Table of contents

1	OTW固件升级程序原理介绍	3
2	前期环境准备	9
3	Overwrite模式固件升级步骤(I2C)	15
4	Overwrite模式固件升级步骤(UART)	25
5	Overwrite模式固件升级步骤(SPI)	37
6	Swap模式固件升级步骤	48

Overwrite模式固件升级步骤(SPI)

- 打开user_config.mk文件，将USE_SECURE_MODE_FOR_MCUBOOT设置为0。



Overwrite模式固件升级步骤(SPI)

● 设置SELECTED_TRANSPORT?=SPI

```

README.md  user_config.mk ×  transport_uart.c  dfu_user.c  transport_uart.h  transport_spi.c  transport_spi.h
# when set to 1 and swap mode is enabled, the application in the secondary slot will overwrite the
USE_BOOTSTRAP?=1

#####
# User App Configuration
#####

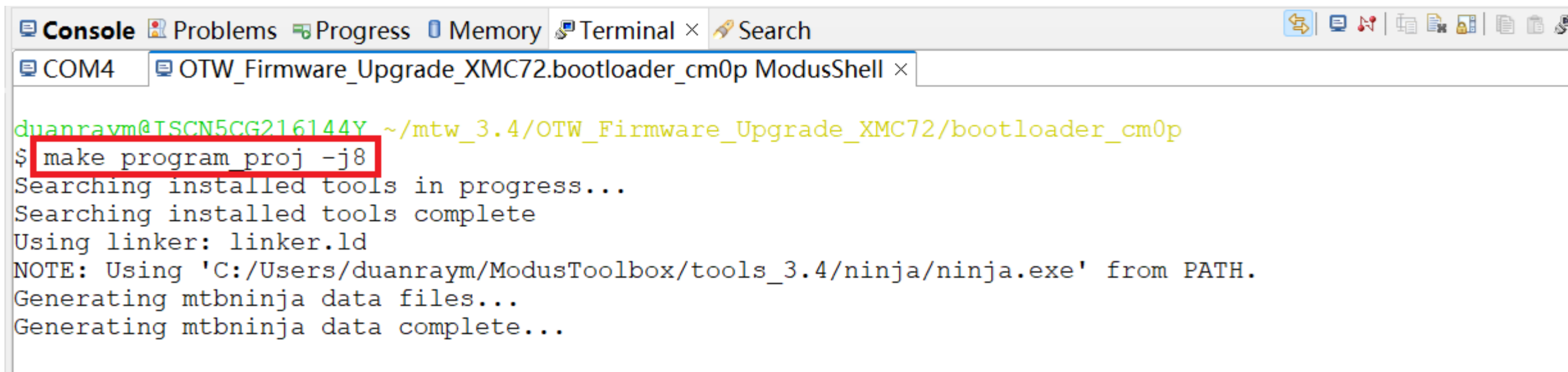
# Default DFU transport set to I2C.
#
# Supported options are I2C, SPI and UART
#
# Select transport here, as required.
SELECTED_TRANSPORT?=SPI

# image type can be BOOT or UPGRADE
IMG_TYPES:=BOOT UPGRADE

```

Overwrite模式固件升级步骤(SPI)

- 选择工程中的"OTW_Firmware_Upgrade.bootloader_cm0p"应用, 打开对应的terminal,输入以下指令
make program_proj -j8 来对bootloader工程进行编译并下载到XMC72 EVAL KIT

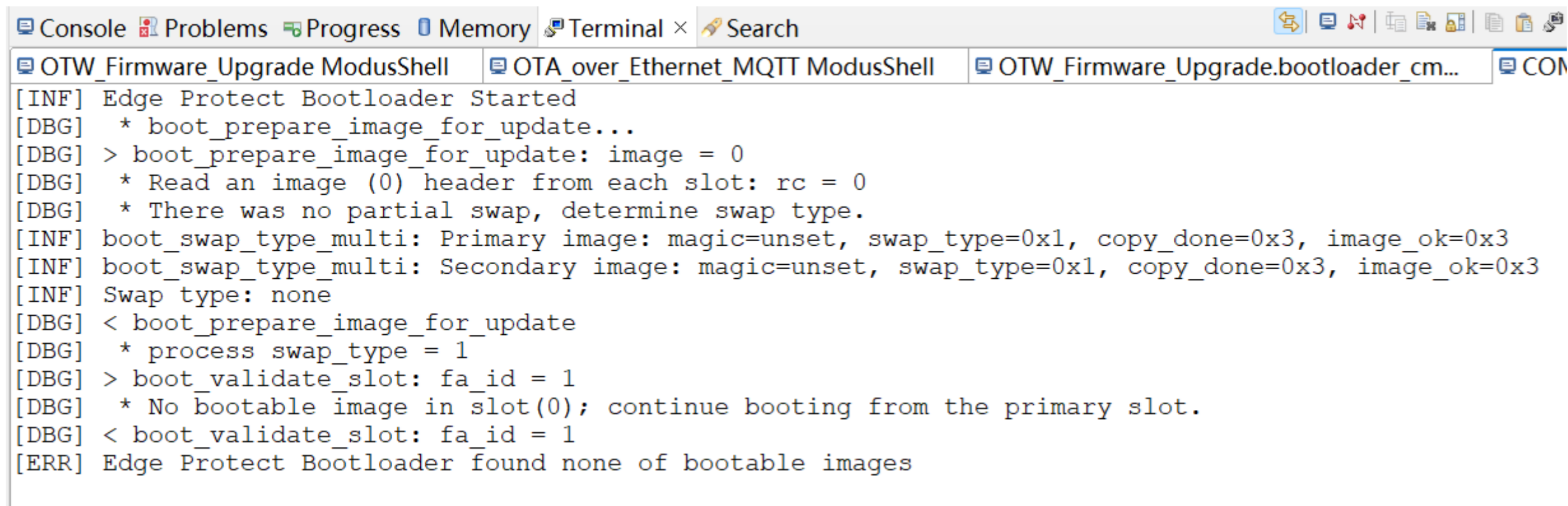


The screenshot shows an IDE interface with a terminal window open. The terminal title is "COM4 OTW_Firmware_Upgrade_XMC72.bootloader_cm0p ModusShell x". The prompt is "duanraym@TSCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade_XMC72/bootloader_cm0p". The command "make program_proj -j8" is entered and highlighted with a red box. The output shows the build process progress, including searching for tools, using linker.ld, and generating mtbninja data files.

```
duanraym@TSCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade_XMC72/bootloader_cm0p
$ make program_proj -j8
Searching installed tools in progress...
Searching installed tools complete
Using linker: linker.ld
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...
```

Overwrite模式固件升级步骤(SPI)

- 烧录完成后，打开串口工具有出现如下提示信息：

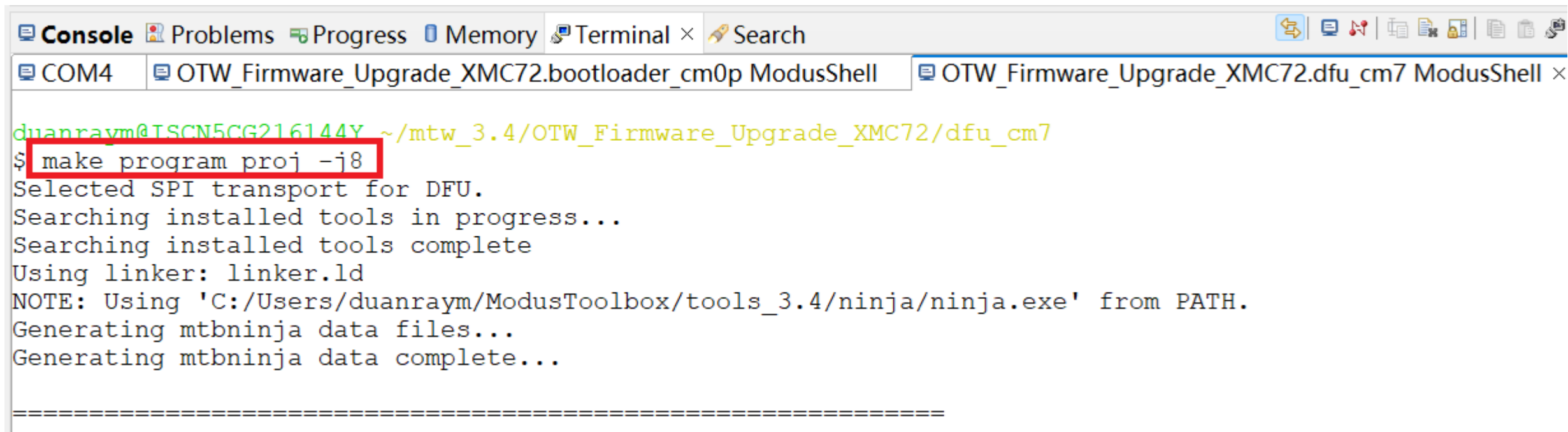


The screenshot shows a serial terminal window with the following tabs: Console, Problems, Progress, Memory, Terminal, and Search. The active tab is 'Console', which displays the output of a boot loader. The logs indicate that the boot loader started, prepared the image for update, read the header from each slot, and determined the swap type. It then attempted to validate the slot but found no bootable images, resulting in an error message.

```
[INF] Edge Protect Bootloader Started
[DBG] * boot_prepare_image_for_update...
[DBG] > boot_prepare_image_for_update: image = 0
[DBG] * Read an image (0) header from each slot: rc = 0
[DBG] * There was no partial swap, determine swap type.
[INF] boot_swap_type_multi: Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] boot_swap_type_multi: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Swap type: none
[DBG] < boot_prepare_image_for_update
[DBG] * process swap_type = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] * No bootable image in slot(0); continue booting from the primary slot.
[DBG] < boot_validate_slot: fa_id = 1
[ERR] Edge Protect Bootloader found none of bootable images
```

Overwrite模式固件升级步骤(SPI)

- 选择工程中的"OTW_Firmware_Upgrade. dfu_cm7"应用, 打开对应的terminal,输入以下指令
make program_proj -j8 来对处于BOOT mode的DFU application工程进行 编译并下载到KIT_XMC72_EVK



The screenshot shows an IDE interface with a terminal window open. The terminal title is "OTW_Firmware_Upgrade_XMC72.dfu_cm7 ModusShell". The user prompt is "duanraym@TSCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade_XMC72/dfu_cm7". The command "make program_proj -j8" is entered and highlighted with a red box. The output shows the build process for the DFU application using SPI transport.

```

duanraym@TSCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade_XMC72/dfu_cm7
$ make program_proj -j8
Selected SPI transport for DFU.
Searching installed tools in progress...
Searching installed tools complete
Using linker: linker.ld
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...
=====

```

Overwrite模式固件升级步骤(SPI)

- 应用下载完成后，打开串口工具，会出现以下信息：

```

Console Problems Progress Memory Terminal x Search
COM4 x OTW_Firmware_Upgrade_XMC72.dfu_cm7 ModusShell OTW_Firmware_Upgrade_XMC72.bootloader_cm0p ModusShell
[INF] Edge Protect Bootloader Started
[DBG] * boot_prepare_image_for_update...
[DBG] > boot_prepare_image_for_update: image = 0
[DBG] * Read an image (0) header from each slot: rc = 0
[DBG] * There was no partial swap, determine swap type.
[INF] boot_swap_type_multi: Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] boot_swap_type_multi: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Swap type: none
[DBG] < boot_prepare_image_for_update
[DBG] * process swap_type = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[INF] User Application validated successfully
[INF] Starting User Application (wait)...
[INF] Start slot Address: 0x10020400
[INF] Launching app on CM7 core
[INF] Edge Protect Bootloader finished.
Deinitializing hardware...

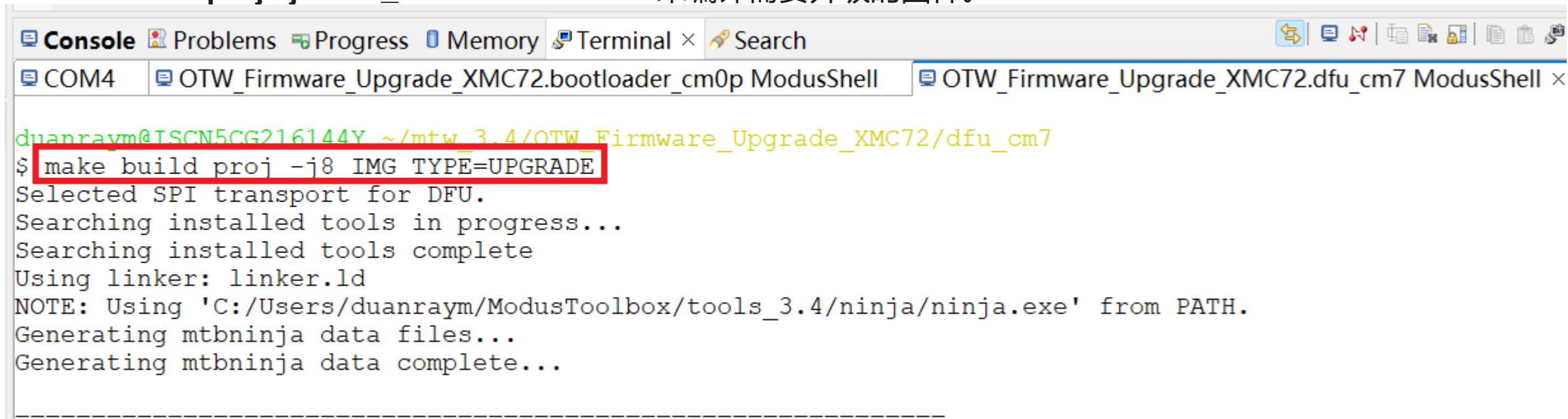
=====
[DFU App] Version: 1.0.0 IMAGE_TYPE: BOOT CPU: CM7_0

=====
[DFU App] SPI DFU TRANSPORT STARTED !!!

```

Overwrite模式固件升级步骤(SPI)

- 选择工程中的"OTW_Firmware_Upgrade. dfu_cm7"应用, 打开对应的terminal, 输入以下指令
make build proj -j8 IMG_TYPE=UPGRADE 来编译需要升级的固件。



```

Console Problems Progress Memory Terminal × Search
COM4 OTW_Firmware_Upgrade_XMC72.bootloader_cm0p ModusShell OTW_Firmware_Upgrade_XMC72.dfu_cm7 ModusShell ×

duanraym@ISCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade_XMC72/dfu_cm7
$ make build proj -j8 IMG_TYPE=UPGRADE
Selected SPI transport for DFU.
Searching installed tools in progress...
Searching installed tools complete
Using linker: linker.ld
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...
=====
  
```

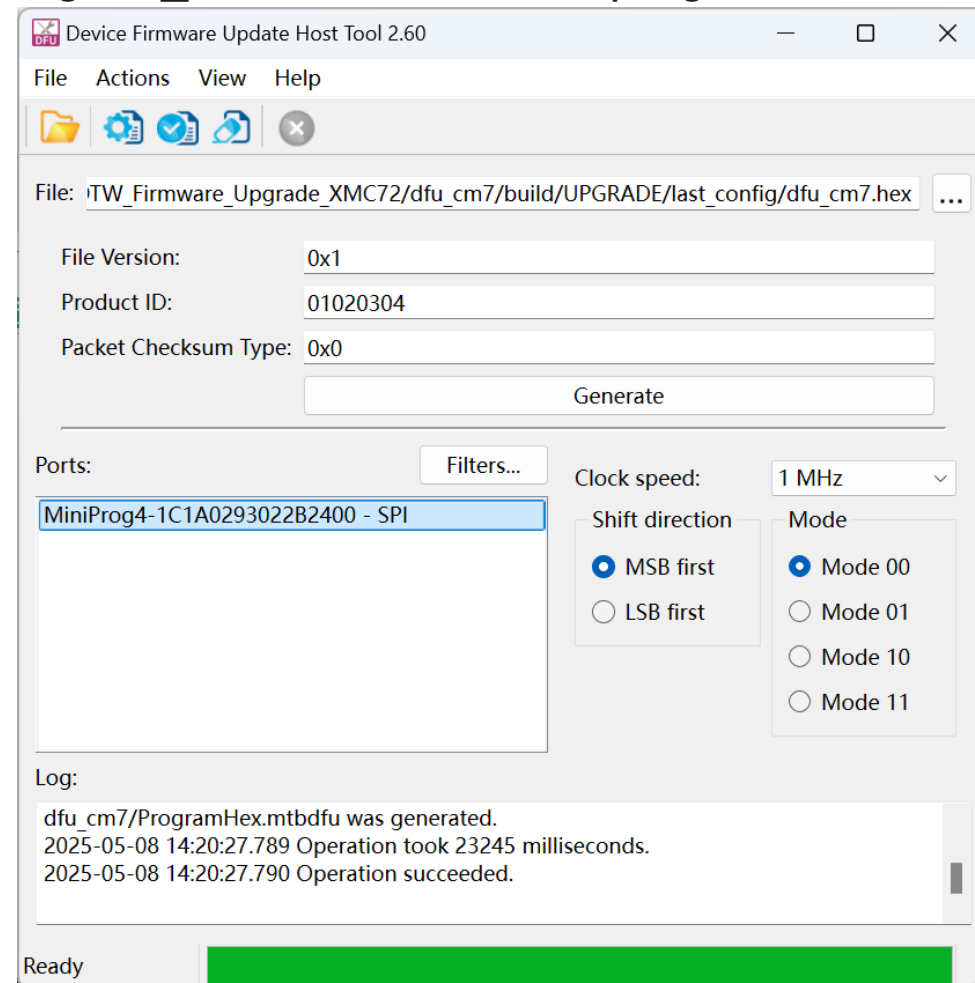
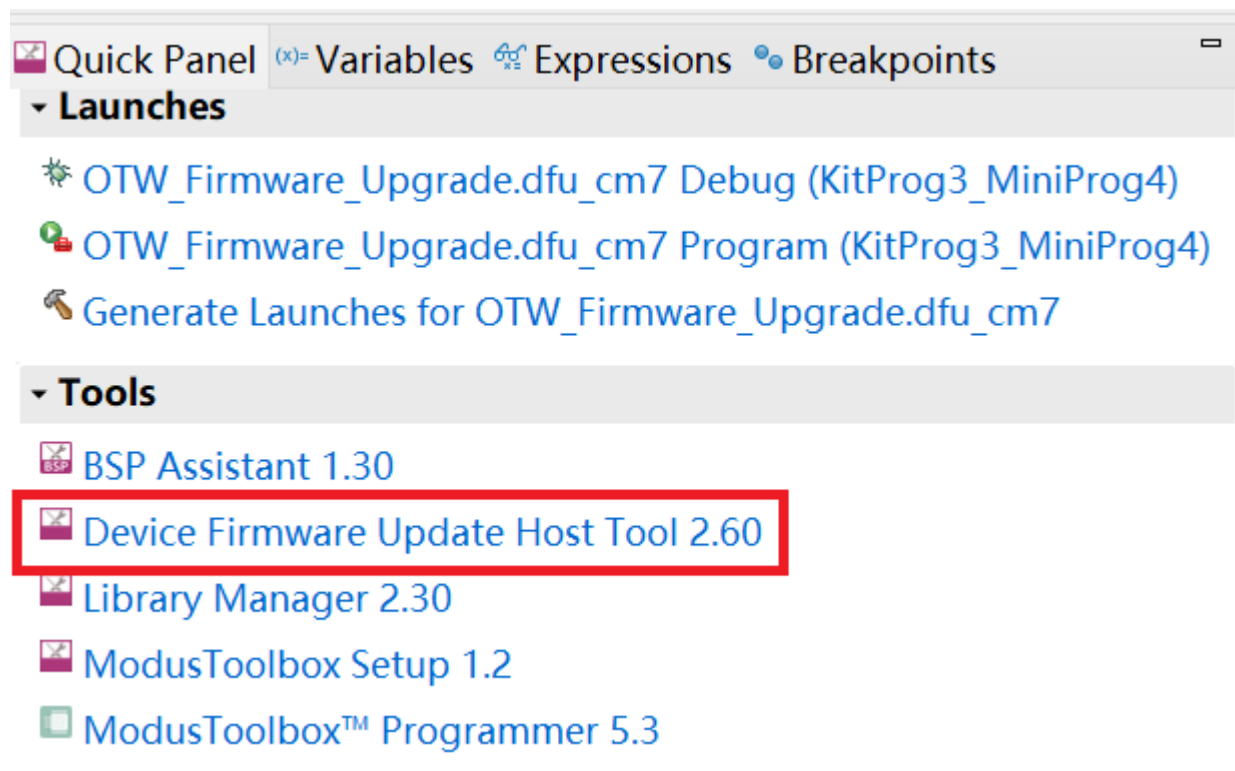
Overwrite模式固件升级步骤(SPI)

- 将MiniProg4和KIT_XMC72_EVK的以下几个PIN分别进行连接

MiniProg4	KIT_XMC72_EVK
SCLK (SPI CLK)	P10.2
SSEL (SPI CS)	P10.3
MOSI	P10.1
MISO	P10.0
VTARG	3.3V
GND	GND

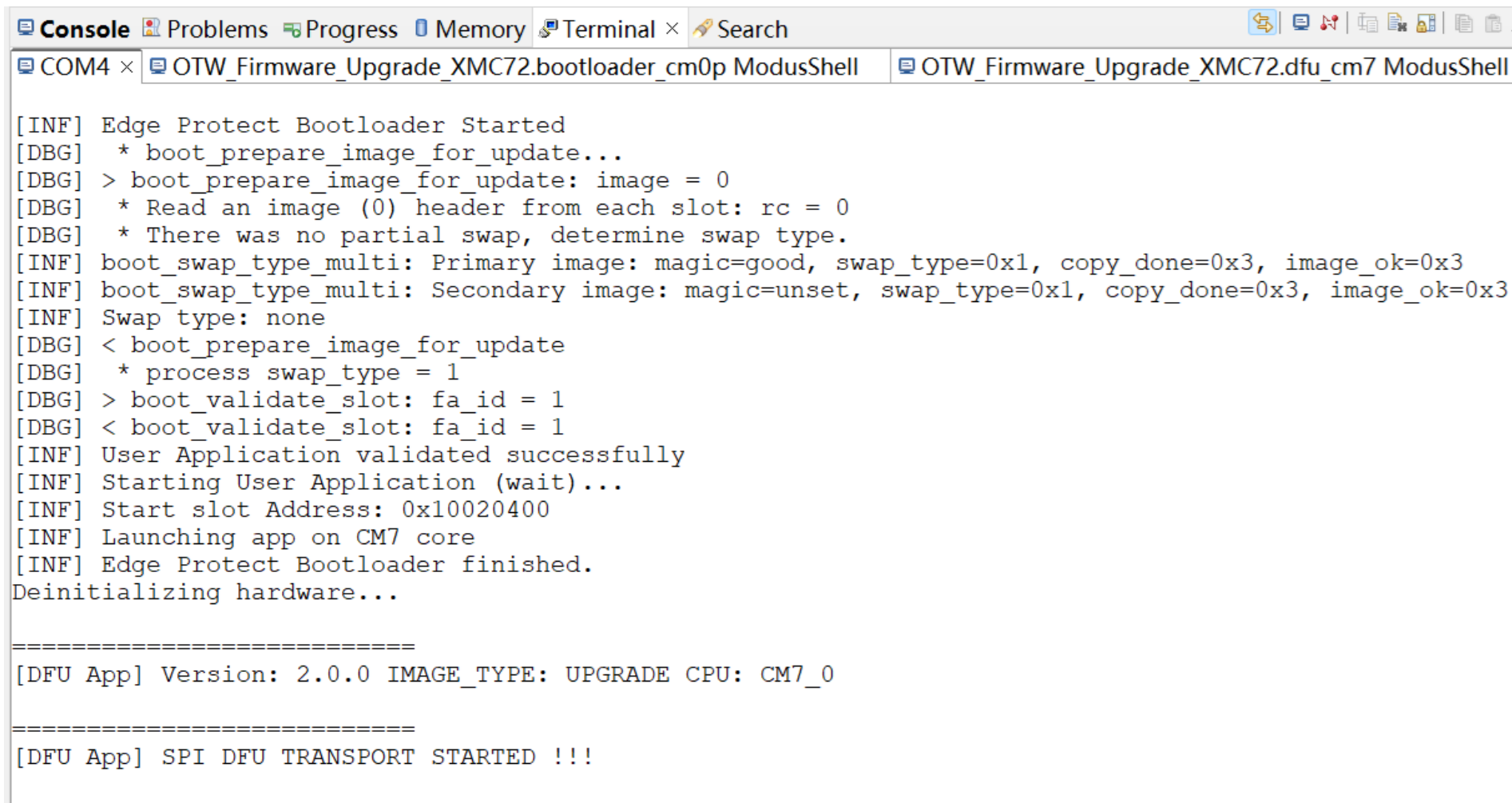
Overwrite模式固件升级步骤(SPI)

- 打开下图红色标记所示的DFU工具。然后，选择MiniProg4 SPI端口，其他配置按下图所示，hex文件路径:<OTW_Firmware_Upgrade>dfu_cm7\build\UPGRADE\last_config\dfu_cm7.hex，最后点击program进行烧录。



Overwrite模式固件升级步骤(SPI)

- 升级完成后，打开串口工具，可以观察到以下信息：



```

Console Problems Progress Memory Terminal × Search
COM4 × OTW_Firmware_Upgrade_XMC72.bootloader_cm0p ModusShell OTW_Firmware_Upgrade_XMC72.dfu_cm7 ModusShell

[INF] Edge Protect Bootloader Started
[DBG] * boot_prepare_image_for_update...
[DBG] > boot_prepare_image_for_update: image = 0
[DBG] * Read an image (0) header from each slot: rc = 0
[DBG] * There was no partial swap, determine swap type.
[INF] boot_swap_type_multi: Primary image: magic=good, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] boot_swap_type_multi: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Swap type: none
[DBG] < boot_prepare_image_for_update
[DBG] * process swap_type = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[INF] User Application validated successfully
[INF] Starting User Application (wait)...
[INF] Start slot Address: 0x10020400
[INF] Launching app on CM7 core
[INF] Edge Protect Bootloader finished.
Deinitializing hardware...

=====
[DFU App] Version: 2.0.0 IMAGE_TYPE: UPGRADE CPU: CM7_0

=====
[DFU App] SPI DFU TRANSPORT STARTED !!!
  
```

Table of contents

1	OTW固件升级程序原理介绍	3
2	前期环境准备	9
3	Overwrite模式固件升级步骤(I2C)	15
4	Overwrite模式固件升级步骤(UART)	25
5	Overwrite模式固件升级步骤(SPI)	37
6	Swap模式固件升级步骤	48

Swap模式固件升级步骤

- 打开user_config.mk文件，按照下图所示修改FLASH_MAP， USE_SECURE_MODE_FOR_MCUBOOT 变量设置为 '0' 。

```
user_config.mk ×
#####
# Shared Configuration
#####

# Include the common make file
include ../common.mk

# Flashmap JSON file name
#FLASH_MAP?=xmc7000 overwrite single.json
FLASH_MAP?=xmc7000_swap_single.json

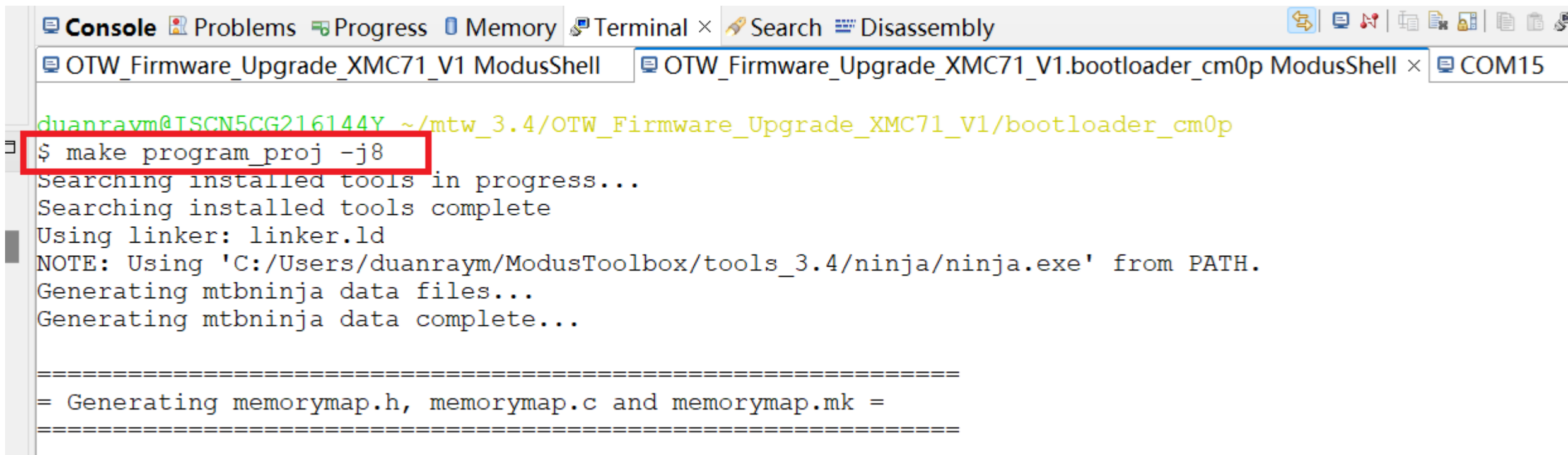
# Device family name. Ex: PSOC6, XMC7000
FAMILY=XMC7000

#####
# MCUBOOT App Configuration
#####

# To prepare a secure MCUBoot image, set the variable to '1'
USE_SECURE_MODE_FOR_MCUBOOT?=0
```

Swap模式固件升级步骤

- 选择工程中的"OTW_Firmware_Upgrade.bootloader_cm0p"应用，打开对应的terminal,输入以下指令
make program_proj -j8 来对bootloader工程进行编译并下载到EVAL KIT



```

Console Problems Progress Memory Terminal × Search Disassembly
OTW_Firmware_Upgrade_XMC71_V1 ModusShell OTW_Firmware_Upgrade_XMC71_V1.bootloader_cm0p ModusShell × COM15

duanraym@TSCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade_XMC71_V1/bootloader_cm0p
$ make program_proj -j8
Searching installed tools in progress...
Searching installed tools complete
Using linker: linker.ld
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...

=====
= Generating memorymap.h, memorymap.c and memorymap.mk =
=====
  
```

Swap模式固件升级步骤

- 烧录完成后，打开串口工具有出现如下提示信息：

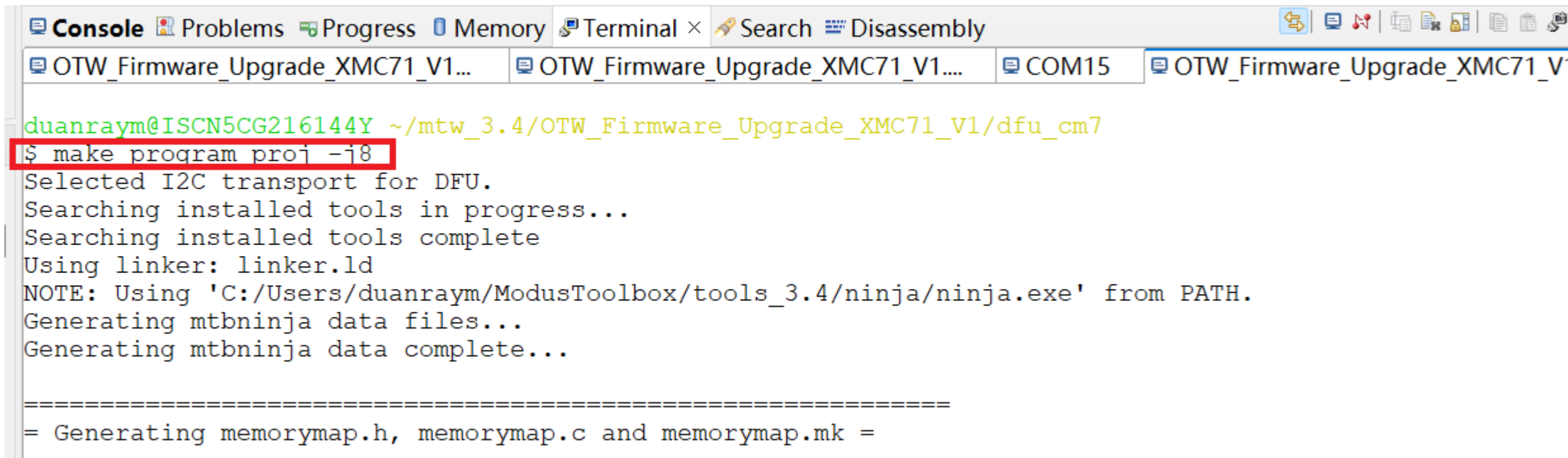
```

Console Problems Progress Memory Terminal × Search Disassembly
OTW_Firmware_Upgrade_XMC71_V1 ModusShell OTW_Firmware_Upgrade_XMC71_V1.bootloader_cm0p ModusShell COM15 ×
[DBG] * Read an image (0) header from each slot: rc = 0
[DBG] * selected SCRATCH area, copy_done = 3
[INF] Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Scratch: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Boot source: primary slot
[DBG] > STATUS: swap_read_status_bytes: fa_id = 1
[DBG] * re-read image(0) headers: rc = 0.
[DBG] * There was no partial swap, determine swap type.
[INF] boot_swap_type_multi: Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] boot_swap_type_multi: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Swap type: none
[DBG] > boot_validate_slot: fa_id = 1
[DBG] * Fix the secondary slot when image is invalid.
[DBG] * No bootable image in slot(0); continue booting from the primary slot.
[DBG] < boot_validate_slot: fa_id = 1
[DBG] > boot_validate_slot: fa_id = 2
[DBG] * Fix the secondary slot when image is invalid.
[DBG] * Erase secondary image trailer.
[INF] Erasing trailer; fa_id=2
[DBG] * No bootable image in slot(1); continue booting from the primary slot.
[DBG] < boot_validate_slot: fa_id = 2
[DBG] < boot_prepare_image_for_update
[DBG] * process swap_type = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] * Fix the secondary slot when image is invalid.
[DBG] * No bootable image in slot(0); continue booting from the primary slot.
[DBG] < boot_validate_slot: fa_id = 1
[ERR] Edge Protect Bootloader found none of bootable images

```

Swap模式固件升级步骤

- 选择工程中的"OTW_Firmware_Upgrade. dfu_cm7"应用, 打开对应的terminal, 输入以下指令
make program_proj -j8 来对处于BOOT mode的DFU application工程进行 编译并下载到EVAL KIT



```

Console Problems Progress Memory Terminal × Search Disassembly
OTW_Firmware_Upgrade_XMC71_V1... OTW_Firmware_Upgrade_XMC71_V1... COM15 OTW_Firmware_Upgrade_XMC71_V1...

duanraym@ISCN5CG216144Y ~/mtw_3.4/OTW_Firmware_Upgrade_XMC71_V1/dfu_cm7
$ make program_proj -j8
Selected I2C transport for DFU.
Searching installed tools in progress...
Searching installed tools complete
Using linker: linker.ld
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...

=====
= Generating memorymap.h, memorymap.c and memorymap.mk =
  
```

Swap模式固件升级步骤

- 应用下载完成后，打开串口工具，会出现以下信息：

```

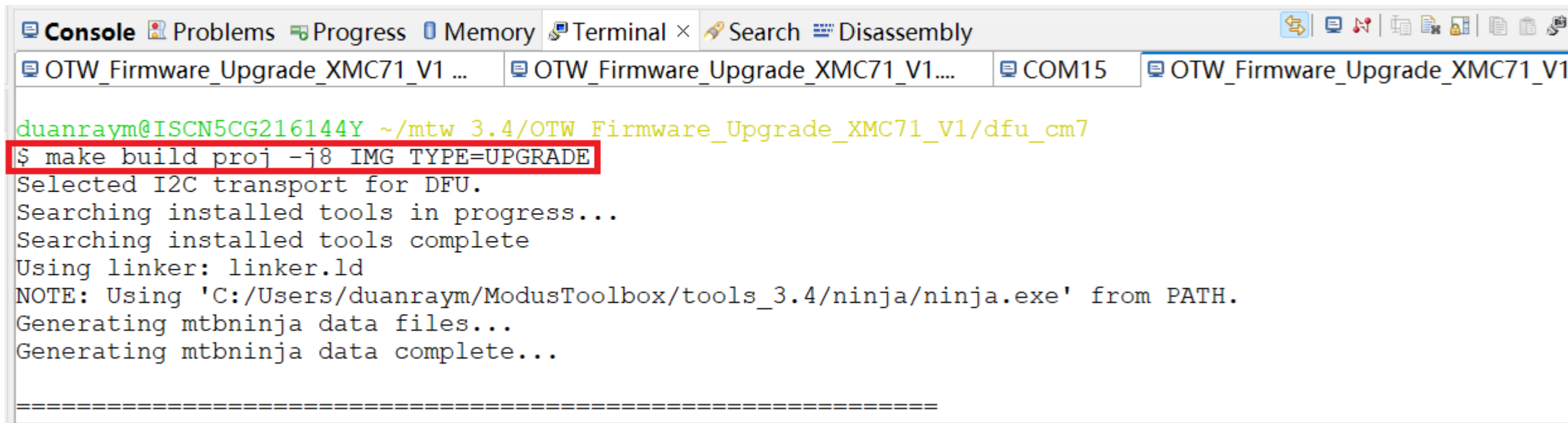
Console Problems Progress Memory Terminal × Search Disassembly
OTW_Firmware_Upgrade_XMC71_V1... OTW_Firmware_Upgrade_XMC71_V1... COM15 × OTW_Firmware_Upgrade_XMC71_V1
[INF] Edge Protect Bootloader Started
[DBG] * boot_prepare_image_for_update...
[DBG] > boot_prepare_image_for_update: image = 0
[DBG] * Read an image (0) header from each slot: rc = 0
[DBG] Slot 0 firmware + tlvs size = 44127, slot size = 131072, write_size = 32768, write_size * sect_nu
m - write_size = 98304
[DBG] * selected SCRATCH area, copy_done = 3
[INF] Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Scratch: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Boot source: primary slot
[DBG] > STATUS: swap_read_status_bytes: fa_id = 1
[DBG] * re-read image(0) headers: rc = 0.
[DBG] * There was no partial swap, determine swap type.
[INF] boot_swap_type_multi: Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] boot_swap_type_multi: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
[INF] Swap type: none
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[DBG] < boot_prepare_image_for_update
[DBG] * process swap_type = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[INF] User Application validated successfully
[INF] Starting User Application (wait)...
[INF] Start slot Address: 0x10020400
[INF] Launching app on CM7 core
[INF] Edge Protect Bootloader finished.
Deinitializing hardware...

=====
[DFU App] Version: 1.0.0 IMAGE_TYPE: BOOT CPU: CM7_0
=====

```


Swap模式固件升级步骤

- 选择工程中的"OTW_Firmware_Upgrade. dfu_cm7"应用，打开对应的terminal,输入以下指令
make build proj -j8 IMG_TYPE=UPGRADE



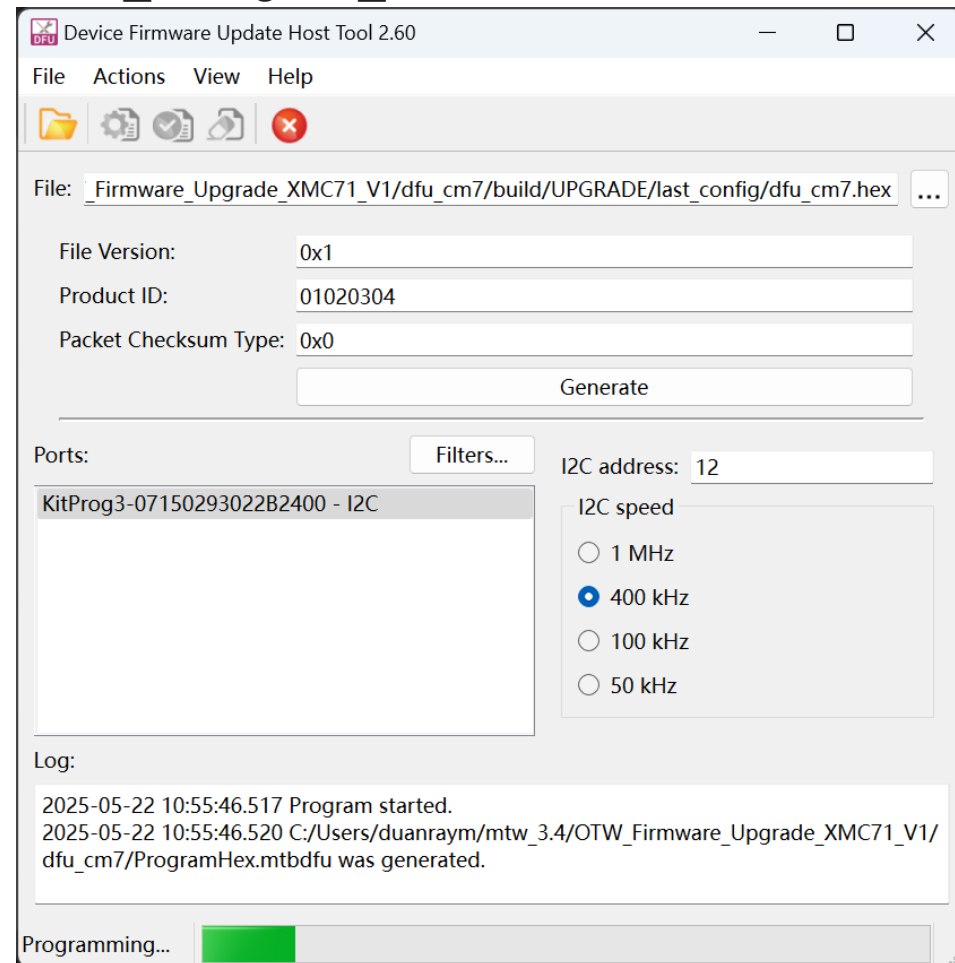
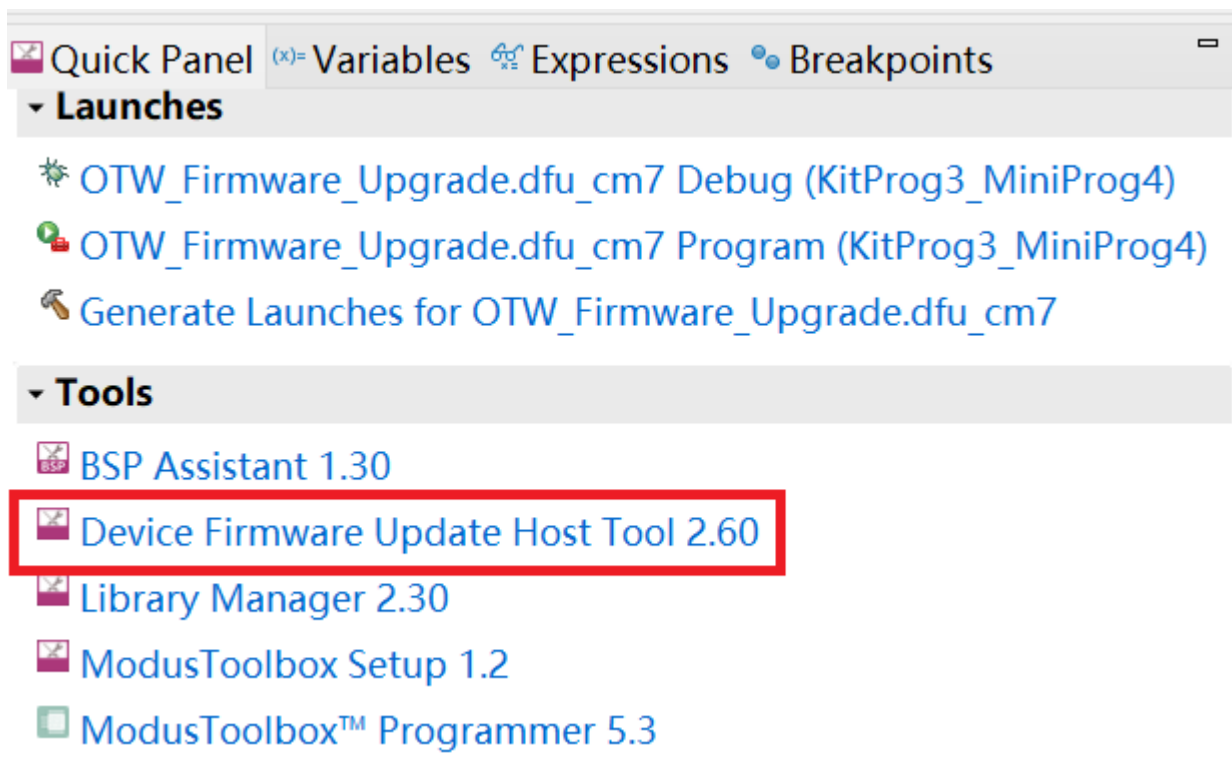
The screenshot shows an IDE interface with a terminal window open. The terminal displays the following output:

```

duanraym@ISCN5CG216144Y ~/mtw 3.4/OTW_Firmware_Upgrade_XMC71_V1/dfu_cm7
$ make build proj -j8 IMG_TYPE=UPGRADE
Selected I2C transport for DFU.
Searching installed tools in progress...
Searching installed tools complete
Using linker: linker.ld
NOTE: Using 'C:/Users/duanraym/ModusToolbox/tools_3.4/ninja/ninja.exe' from PATH.
Generating mtbninja data files...
Generating mtbninja data complete...
=====
  
```

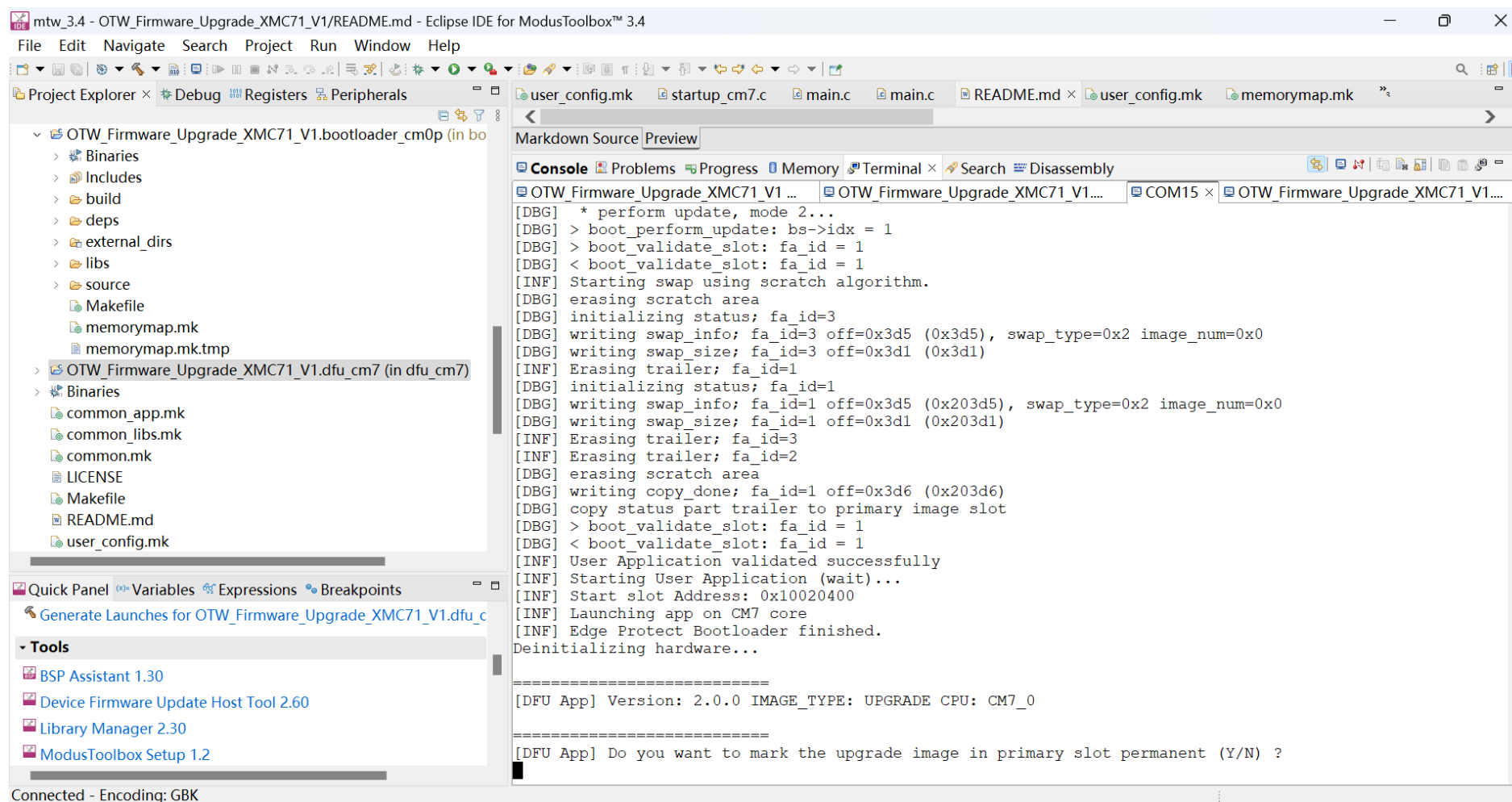

Swap模式固件升级步骤

- 打开下图红色标记所示的DFU工具，选择KitProg3 I2C端口，分别设置I2C Address为12， I2C speed为400KHz。hex文件路径:<OTW_Firmware_Upgrade>dfu_cm7\build\UPGRADE\last_config\dfu_cm7.hex，最后点击program进行固件升级。



Swap模式固件升级步骤

- 更新完成后，打开串口工具可以看到以下打印信息：



```

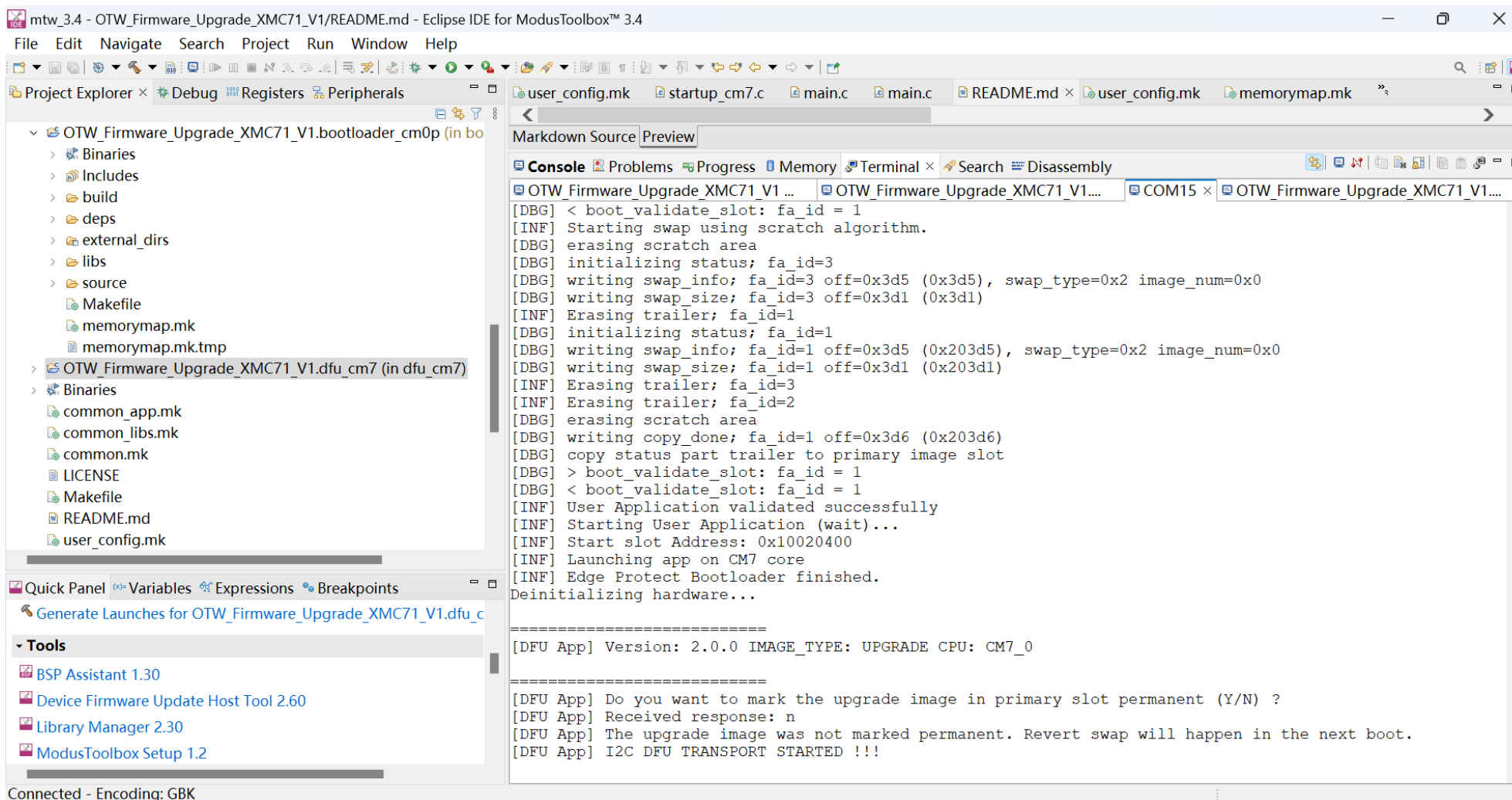
mtw_3.4 - OTW_Firmware_Upgrade_XMC71_V1/README.md - Eclipse IDE for ModusToolbox™ 3.4
File Edit Navigate Search Project Run Window Help
Project Explorer x Debug Registers Peripherals
OTW_Firmware_Upgrade_XMC71_V1.bootloader_cm0p (in bo
  Binaries
  Includes
  build
  deps
  external_dirs
  libs
  source
  Makefile
  memorymap.mk
  memorymap.mk.tmp
OTW_Firmware_Upgrade_XMC71_V1.dfu_cm7 (in dfu_cm7)
  Binaries
  common_app.mk
  common_libs.mk
  common.mk
  LICENSE
  Makefile
  README.md
  user_config.mk
Quick Panel (x) Variables Expressions Breakpoints
Generate Launches for OTW_Firmware_Upgrade_XMC71_V1.dfu_c
Tools
  BSP Assistant 1.30
  Device Firmware Update Host Tool 2.60
  Library Manager 2.30
  ModusToolbox Setup 1.2
Connected - Encoding: GBK

Markdown Source Preview
Console Problems Progress Memory Terminal x Search Disassembly
OTW_Firmware_Upgrade_XMC71_V1... OTW_Firmware_Upgrade_XMC71_V1... COM15 x OTW_Firmware_Upgrade_XMC71_V1...
[DBG] * perform update, mode 2...
[DBG] > boot_perform_update: bs->idx = 1
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[INF] Starting swap using scratch algorithm.
[DBG] erasing scratch area
[DBG] initializing status; fa_id=3
[DBG] writing swap_info; fa_id=3 off=0x3d5 (0x3d5), swap_type=0x2 image_num=0x0
[DBG] writing swap_size; fa_id=3 off=0x3d1 (0x3d1)
[INF] Erasing trailer; fa_id=1
[DBG] initializing status; fa_id=1
[DBG] writing swap_info; fa_id=1 off=0x3d5 (0x203d5), swap_type=0x2 image_num=0x0
[DBG] writing swap_size; fa_id=1 off=0x3d1 (0x203d1)
[INF] Erasing trailer; fa_id=3
[INF] Erasing trailer; fa_id=2
[DBG] erasing scratch area
[DBG] writing copy_done; fa_id=1 off=0x3d6 (0x203d6)
[DBG] copy status part trailer to primary image slot
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[INF] User Application validated successfully
[INF] Starting User Application (wait)...
[INF] Start slot Address: 0x10020400
[INF] Launching app on CM7 core
[INF] Edge Protect Bootloader finished.
Deinitializing hardware...

=====
[DFU App] Version: 2.0.0 IMAGE_TYPE: UPGRADE CPU: CM7_0
=====
[DFU App] Do you want to mark the upgrade image in primary slot permanent (Y/N) ?
  
```

Swap模式固件升级步骤

- 当输入回复N时，MCU会在下次启动后回退到bootloader模式，串口打印信息如下：



```

mtw_3.4 - OTW_Firmware_Upgrade_XMC71_V1/README.md - Eclipse IDE for ModusToolbox™ 3.4
File Edit Navigate Search Project Run Window Help
Project Explorer × Debug Registers Peripherals
  OTW_Firmware_Upgrade_XMC71_V1.bootloader_cm0p (in bo
    Binaries
    Includes
    build
    deps
    external_dirs
    libs
    source
    Makefile
    memorymap.mk
    memorymap.mk.tmp
  OTW_Firmware_Upgrade_XMC71_V1.dfu_cm7 (in dfu_cm7)
  Binaries
  common_app.mk
  common_libs.mk
  common.mk
  LICENSE
  Makefile
  README.md
  user_config.mk
Quick Panel (v)= Variables Expressions Breakpoints
  Generate Launches for OTW_Firmware_Upgrade_XMC71_V1.dfu_c
  Tools
    BSP Assistant 1.30
    Device Firmware Update Host Tool 2.60
    Library Manager 2.30
    ModusToolbox Setup 1.2
Connected - Encoding: GBK
Console Problems Progress Memory Terminal × Search Disassembly
  OTW_Firmware_Upgrade_XMC71_V1 ... OTW_Firmware_Upgrade_XMC71_V1... COM15 × OTW_Firmware_Upgrade_XMC71_V1...
[DBG] < boot_validate_slot: fa_id = 1
[INF] Starting swap using scratch algorithm.
[DBG] erasing scratch area
[DBG] initializing status; fa_id=3
[DBG] writing swap_info; fa_id=3 off=0x3d5 (0x3d5), swap_type=0x2 image_num=0x0
[DBG] writing swap_size; fa_id=3 off=0x3d1 (0x3d1)
[INF] Erasing trailer; fa_id=1
[DBG] initializing status; fa_id=1
[DBG] writing swap_info; fa_id=1 off=0x3d5 (0x203d5), swap_type=0x2 image_num=0x0
[DBG] writing swap_size; fa_id=1 off=0x3d1 (0x203d1)
[INF] Erasing trailer; fa_id=3
[INF] Erasing trailer; fa_id=2
[DBG] erasing scratch area
[DBG] writing copy_done; fa_id=1 off=0x3d6 (0x203d6)
[DBG] copy status part trailer to primary image slot
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[INF] User Application validated successfully
[INF] Starting User Application (wait)...
[INF] Start slot Address: 0x10020400
[INF] Launching app on CM7 core
[INF] Edge Protect Bootloader finished.
[INF] Deinitializing hardware...

=====
[DFU App] Version: 2.0.0 IMAGE_TYPE: UPGRADE CPU: CM7_0

=====
[DFU App] Do you want to mark the upgrade image in primary slot permanent (Y/N) ?
[DFU App] Received response: n
[DFU App] The upgrade image was not marked permanent. Revert swap will happen in the next boot.
[DFU App] I2C DFU TRANSPORT STARTED !!!

```

Swap模式固件升级步骤

- 当输入回复Y时，UPGRADE image文件作为永久的主image，串口打印信息如下：

```

Console Problems Progress Memory Terminal × Search Disassembly
OTW_Firmware_Upgrade_XMC71_V1... OTW_Firmware_Upgrade_XMC71_V1... COM15 × OTW_Firmware_Upgrade_XMC71_V1...
[DBG] < boot_validate_slot: fa_id = 1
[INF] Starting swap using scratch algorithm.
[DBG] erasing scratch area
[DBG] initializing status; fa_id=3
[DBG] writing swap_info; fa_id=3 off=0x3d5 (0x3d5), swap_type=0x2 image_num=0x0
[DBG] writing swap_size; fa_id=3 off=0x3d1 (0x3d1)
[INF] Erasing trailer; fa_id=1
[DBG] initializing status; fa_id=1
[DBG] writing swap_info; fa_id=1 off=0x3d5 (0x203d5), swap_type=0x2 image_num=0x0
[DBG] writing swap_size; fa_id=1 off=0x3d1 (0x203d1)
[INF] Erasing trailer; fa_id=3
[INF] Erasing trailer; fa_id=2
[DBG] erasing scratch area
[DBG] writing copy_done; fa_id=1 off=0x3d6 (0x203d6)
[DBG] copy status part trailer to primary image slot
[DBG] > boot_validate_slot: fa_id = 1
[DBG] < boot_validate_slot: fa_id = 1
[INF] User Application validated successfully
[INF] Starting User Application (wait)...
[INF] Start slot Address: 0x10020400
[INF] Launching app on CM7 core
[INF] Edge Protect Bootloader finished.
Deinitializing hardware...

=====
[DFU App] Version: 2.0.0 IMAGE_TYPE: UPGRADE CPU: CM7_0

=====
[DFU App] Do you want to mark the upgrade image in primary slot permanent (Y/N) ?
[DFU App] Received response: y
[DFU App] SWAP Status : Image OK was set at 0x1003ffe8.
[DFU App] I2C DFU TRANSPORT STARTED !!!

```

