Application Signature Tools

&

CANoe Flashing Tools

UserManual

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Data | Author | Status | Remarks |
| V0.0.1 | 2022.03.28 | Chaoqiang | Init | Init Version |
| V0.0.2 | 2022.05.17 | Mingfen XIAO | Modified | 1. Chapter 5,added some comments 2. Add Chapter 8 |
| V0.0.3 | 2022.05.20 | Mingfen XIAO | Modified | 1. Ajust the structure 2. Added Chapter 3 |
| V0.0.4 | 2022.09.21 | Chaoqiang Yan | Modified | 1. Add Chapter 1.5 2. Add Chapter 1.6 3. Add Chapter 3 Download Application BootloaderV4.0 |
| V0.0.5 | 2022.10.28 | Chaoqiang Yan | Modified | 1. Add Chapter 1.7 2. Add Chapter 4 Download Application BootloaderV5.0 |
|  |  |  |  |  |

**Notes: Test Engineer only Focus in Chapter2.**

Contents

[1. Compile and Integration 3](#_Toc114671850)

[1.1 Configure Application A and B bank 3](#_Toc114671851)

[1.2 Update TRICORE-tasking.ldscript file 3](#_Toc114671852)

[1.3 Compile code 4](#_Toc114671853)

[1.4 BootloaderV1.0&V2.0&V3.0 Generate Bootloader format file 4](#_Toc114671854)

[1.5 Bootloader V4.0 Generate Bootloader format file 5](#_Toc114671855)

[1.6 Bootloader update from V2.0 or V3.0 to V4.0 7](#_Toc114671856)

[2. Download Application 8](#_Toc114671857)

[2.1 Place Bootloader format in RPG\_CANoe12\data 8](#_Toc114671858)

[2.2 Download Bootloader hex by Lauterach or miniwiggler 9](#_Toc114671859)

[2.3 Download New Bootloader by G-Puls(kavsar) 9](#_Toc114671860)

[2.4 Downloading Application Code 9](#_Toc114671861)

[3. Debug 12](#_Toc114671862)

[4. Set the Application Bank 12](#_Toc114671863)

[4.2 Removed the generated ld script 13](#_Toc114671864)

[4.3 Compiling Code 13](#_Toc114671865)

[4.4 Downloading NonSignature bootloader 13](#_Toc114671866)

[4.5 Downloading application elf/hex 13](#_Toc114671867)

[4.6 Clean data flash 13](#_Toc114671868)

# Compile and Integration

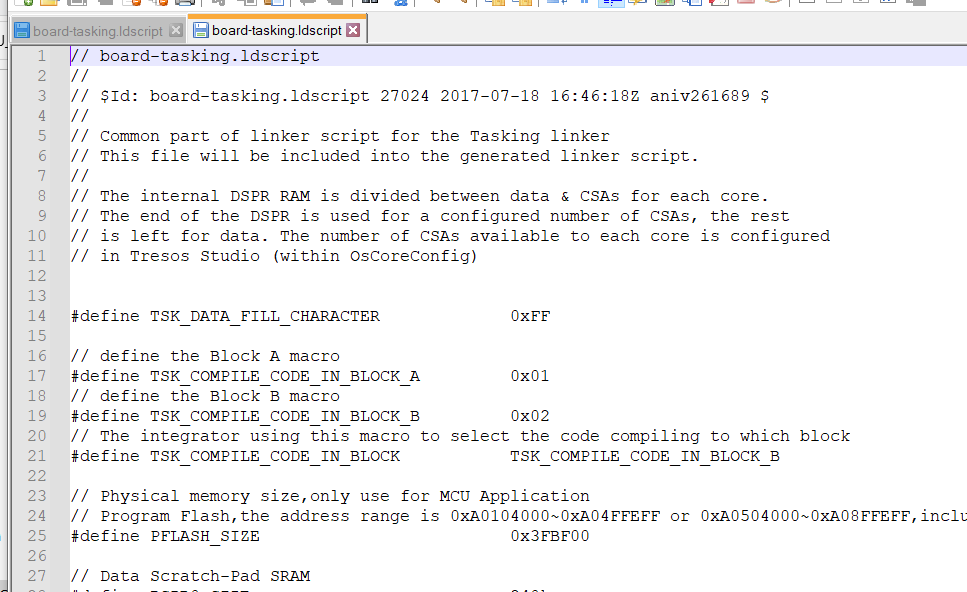
## Configure Application A and B bank

We need to configure application A and B bank in the ld file.

The path is **\01\_AsrConfig\source\boards\TriboardTC387TH\_DPM\board-tasking.ldscript**

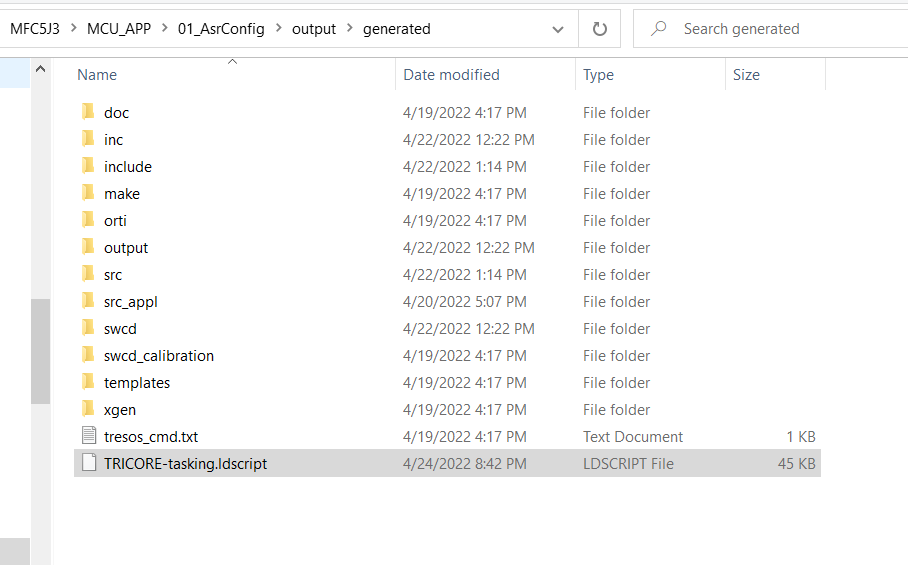
A Bank-> TSK\_COMPILE\_CODE\_IN\_BLOCK\_A

B Bank-> TSK\_COMPILE\_CODE\_IN\_BLOCK\_B



## Update TRICORE-tasking.ldscript file

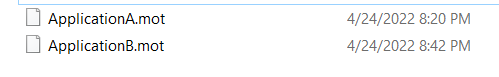
When define A or B bank , need to delete TRICORE-tasking.ldscript file by manual. This file is located in **\01\_AsrConfig\output\generated\TRICORE-tasking.ldscript**



## Compile code

A and B application need to be compiled separately, and rename the .mot or .hex file to

ApplicationA.mot and ApplicationB.mot because of Boot Generate tool only identity this format name.

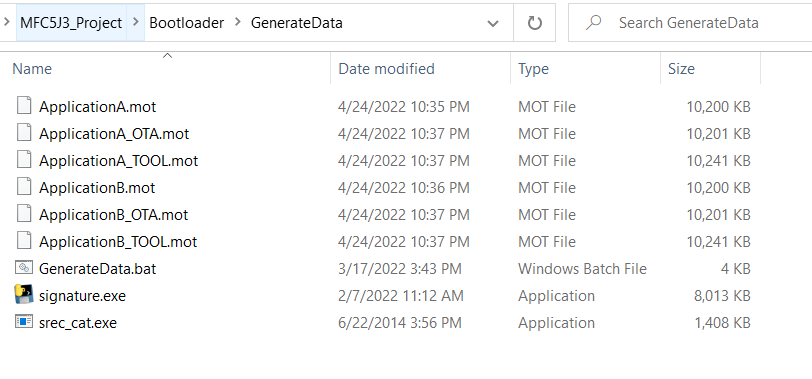


## BootloaderV1.0&V2.0&V3.0 Generate Bootloader format file

Place application origin code ApplicationA.mot and ApplicationB.mot in SignatureTools folder which the location in **\06\_Tools\01\_SignatureTool**

Click GenerateData.bat, it will generate Bootloader format file ApplicationA\_OTA.mot and ApplicationA\_TOOL.mot , ApplicationB\_OTA.mot and ApplicationB\_TOOL.mot.

ApplicationA\_TOOL.mot and ApplicationB\_TOOL.mot can be downloaded by Lauterbach or miniwiggler, the original code can’t be used any more.



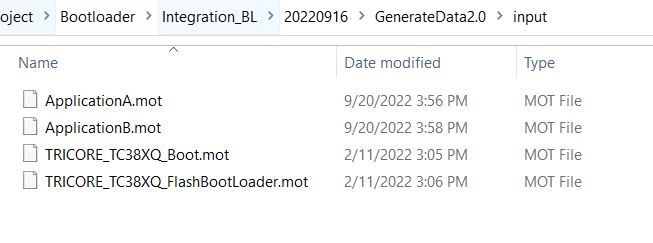
Origin code

Generate code

* ApplicationA.mot/ApplicationB.mot is origin code,gerenation from TASKING;
* ApplicationA\_OTA.mot/ApplicationB\_OTA.mot is added signature file,gerenation from signature tools,use for FBL updated/Flashing;
* ApplicationA\_TOOL.mot/ApplicationB\_OTA.mot is gerenation form signature tools,use for Debug tools(Lauterach Trace32)

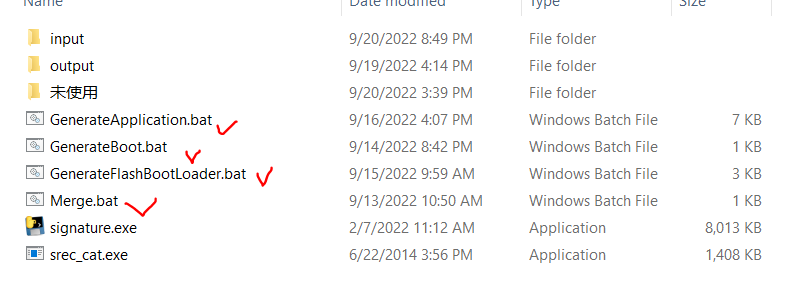
## Bootloader V4.1 Generate Bootloader format file

* Place App Bank A and B in GenerateData2.0\input file
* Place TRICORE\_TC38XQ\_Boot.mot and TRICORE\_TC38XQ\_FlashBootLoader.mot in GenerateData2.0\input file

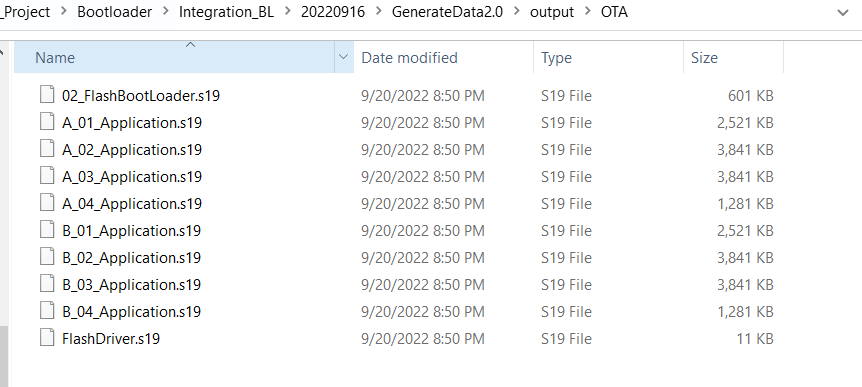


Perform Below .bat file.

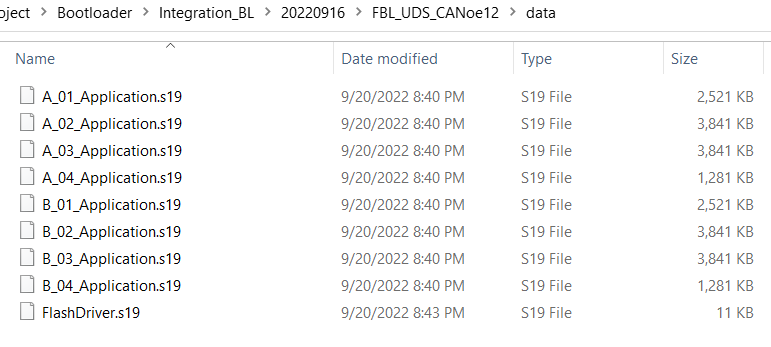
* GenerateApplication.bat
* GenerateBoot.bat
* GenerateFlashBootLoader.bat
* Merge.bat



In GenerateData2.0\Output file\OTA ,it will make 10 items .s19 file.Copy below files to FBL\_UDS\_CANoe12\data folder



Copy to CanOe flash Tool



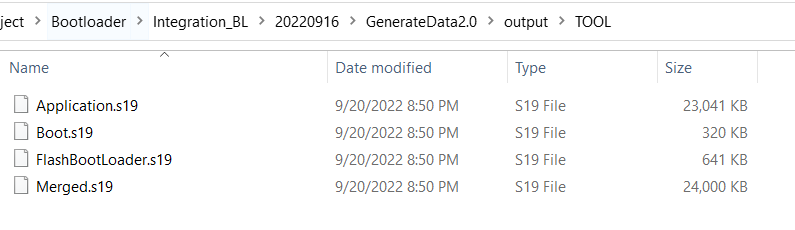
Copy from OTA

In GenerateData2.0\Output\Tool folder, which will make 4 items .s19 file.

* Application.s19 is App file.
* Boot.s19 is first level Boot.
* FlashBootloader.s19 is second level Boot.
* Merged.s19 is that merge Application+Boot+FlashBootLoader file.

Need to programming Boot.s19+FlashBootLoader.s19 as Bootloader parts with miniwiggler tool,

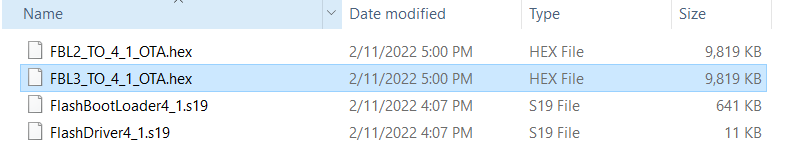
Or programming Merged.s19 as Bootloader+App with miniwiggler tool.



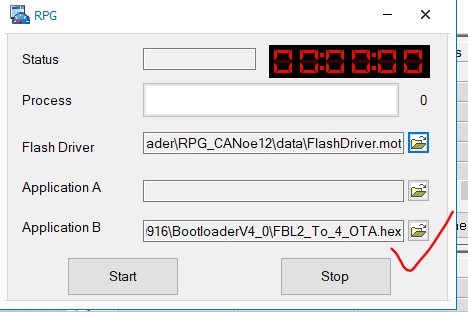
## Bootloader update from V2.0 or V3.0 to V4.1

If MFC5J3 or S202 has used Bootloader V2.0, need to use FBL2\_To\_4\_1\_OTA.hex.

If MFC5J3 or S202 has used Bootloader V3.0, need to use FBL3\_To\_4\_1\_OTA.hex.



Open CANOE tool , choose Application B folder address, choose FBL2\_To\_4\_OTA.hex or FBL3\_To\_4\_1\_OTA.hex

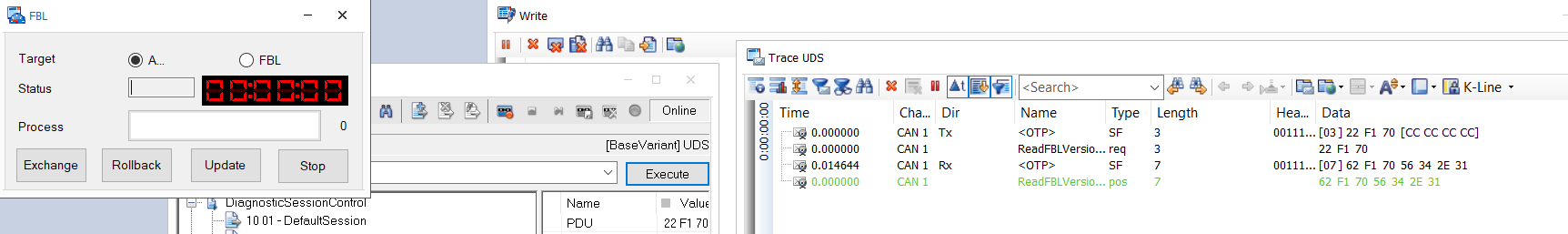


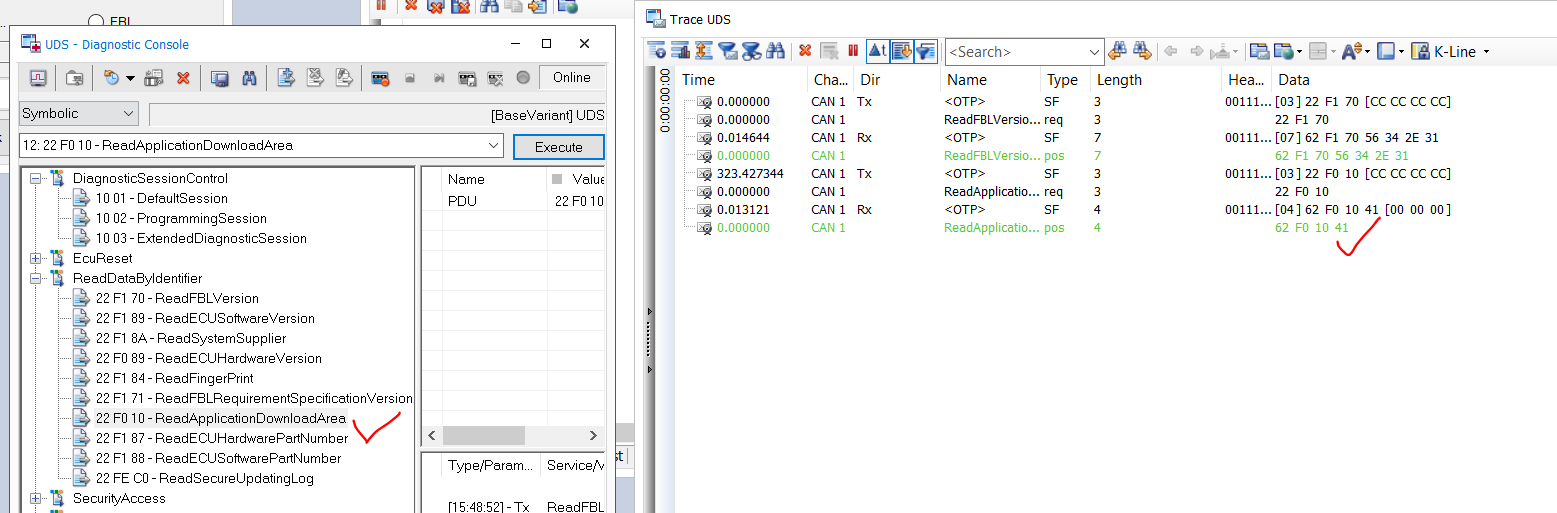
注意：

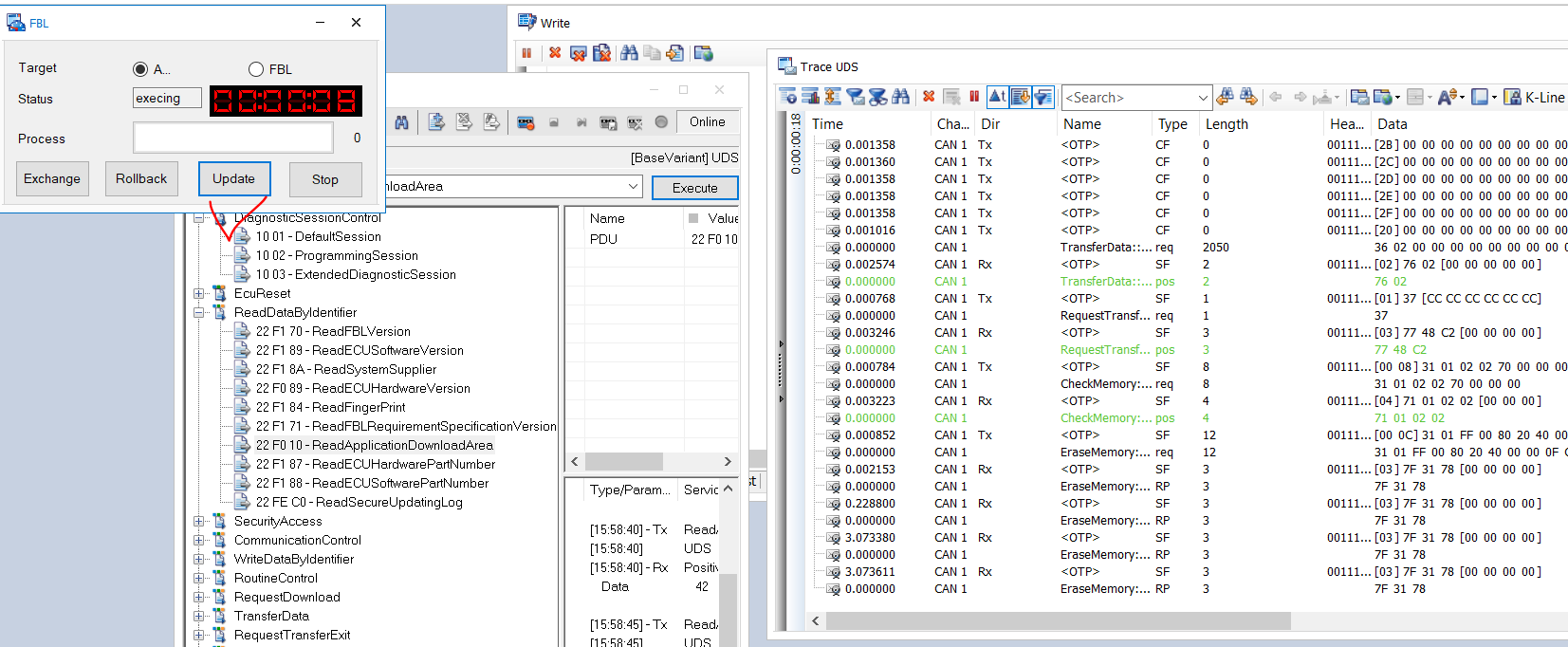
* canoe升级到Bootloader4.0时，需要将App切换到A面，进行刷写B面，否则刷写失败。
* 先将BootloaderV2.0升级到BootloaderV4.0.
* 将BootloaderV3.0升级到BootloaderV4.0.
* 刷写完成之后，停在刷写页面大概30s, 等待B面copy 到Bootloader地址页。
* 使用BootloaderV4.0升级App。

## Bootloader update from V4.1 to V5.0

* 打开CANOE 升级脚本工具FBL\_UDS\_CANoe12\_V4.1toV5.0，先读Bootloader软件版本V4.1
* 确认当前程序运行在A面，通过22 F010,确认当前下载区。
* 点Update 进行Bootloader升级。



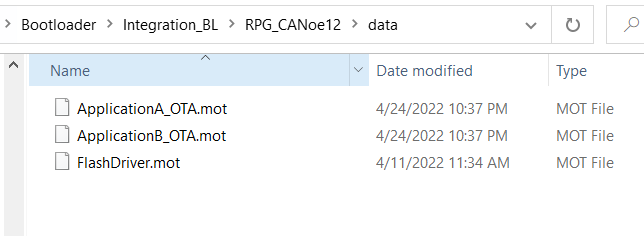




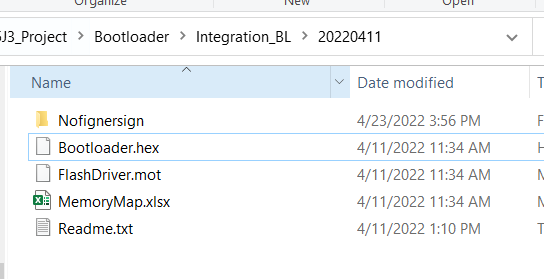
# Download Application with BootloaderV2.0&V3.0

## Place Bootloader format in RPG\_CANoe12\data

Place ApplicationA\_OTA.mot ,ApplicationB\_OTA.mot, FlashDriver.mot in RPG\_CANoe12\data



## Download Bootloader hex by Lauterach or miniwiggler



And the UCB start address is 0x80040080.

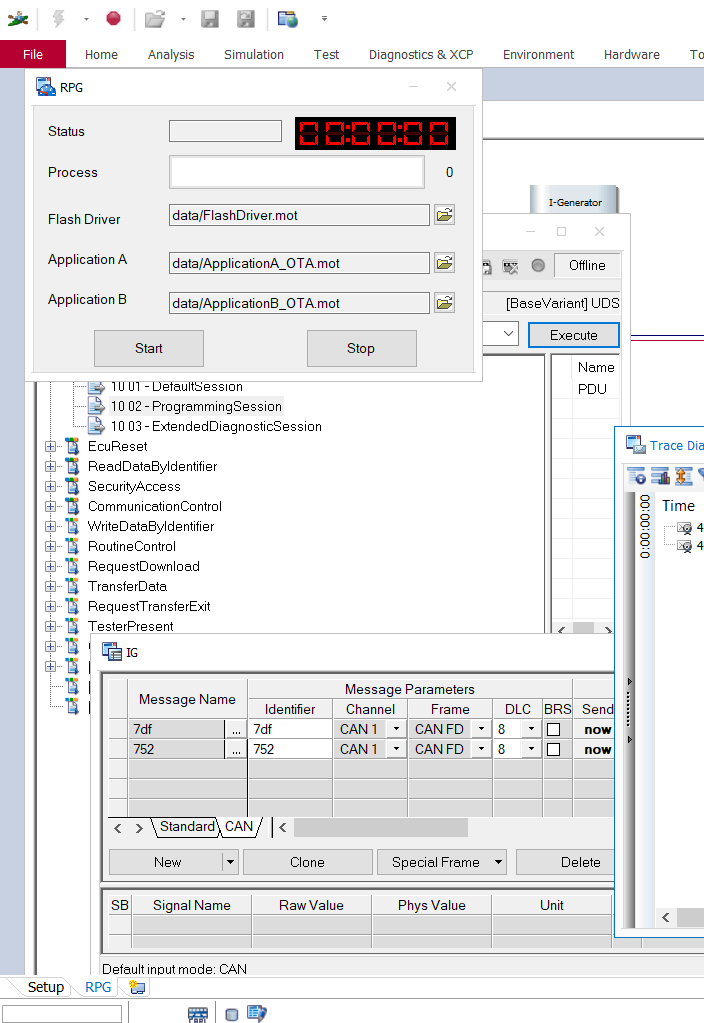
## Download New Bootloader by G-Puls(kavsar)

If the board have G-Puls bootloader, and not have Lauterarch or Minwiggler,we can use the kavsar to download New Bootloader.

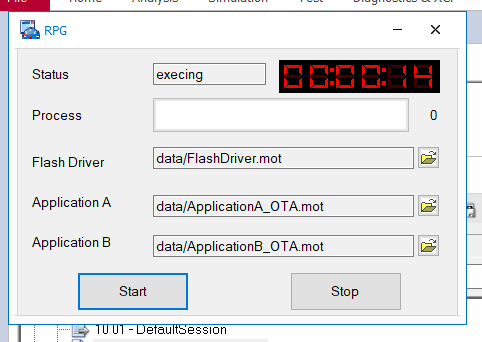
The download method same of the flashing MFC5J3 application, If download fail, you can download application again, then download new bootloader again.

## 2.4 Downloading Application Code

Open Canoe project ->RPG\_CANoe12.cfg , CANOE version need be bigger than CANOE12.



Click Start , it will download A bank and B bank code in order.



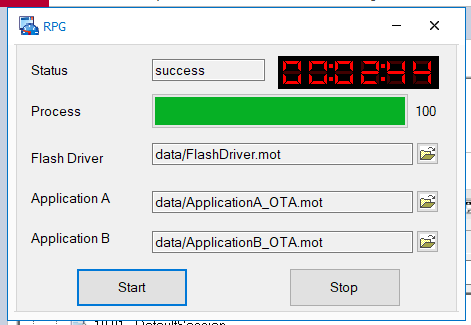
Download A Bank code

Machine generated alternative text:
Write 
Source 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
R PG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
RPG 
C I i ent 
CI i ent 
C I i ent 
CI i ent 
C I i ent 
CI i ent 
C I i ent 
CI i ent 
C I i ent 
CI i ent 
C I i ent 
CI i ent 
C I i ent 
CI i ent 
C I i ent 
CI i ent 
C I i ent 
CI i ent 
C I i ent 
CI i ent 
C I i ent 
CI i ent 
Message 
parserMotFiIe [data/ FlashDriver . mot] 
Mot Parser finish. Address [Ox 70000000] Blocksize [OXOOOOIOOO] DataSize [OXOOOOIOOO] S' 
segment : Address [0>00000000] size [Ox00001000] Offset [Oxoooooooo] CRC[OXOOOOOOOO] 
Segment DataTop[82008201] 
Segment DataEnd[OOOOOOOO] 
parserMotFiIe [data/ ApplicationA OTA.mot] 
Mot Parser finish. Address [OxA01040 
Segment Address [OxA0104000] Size 
Segment DataQop[OOOOOOOO] 
Segment DataEnd[411ÄS85Ä] 
Change to step ] 
Progress [0/4182016] 
Received positive response for service 
Change to step [17] 
Progress [0/4182016] 
Received pos It I ve response for service 
wMaxNumbe rOfBI ockLength [ 2 0 5 0 ] 
Change to step [18] 
Progress [0/4182016] 
Received positive response for service 
Change to step 18] 
Progress [2048/4182016] 
Size [Ox003FCOOO] DataSize [Ox003FCOOO] S' 
3FCOOO] Offset [Ox00001000] CRC[OXOOOOOOOO] 
Ox€e 
ox74 
ox7€ 

Download B Bank code

Machine generated alternative text:
Source 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
O 
R PG 
RPG 
R PG 
RPG 
R PG 
RPG 
R PG 
RPG 
R PG 
RPG 
R PG 
RPG 
R PG 
RPG 
R PG 
RPG 
R PG 
RPG 
R PG 
RPG 
R PG 
RPG 
C I i ent 
CI ent 
C I i ent 
CI ent 
C I i ent 
CI ent 
C I i ent 
CI ent 
C I i ent 
CI ent 
C I i ent 
CI ent 
C I i ent 
CI ent 
C I i ent 
CI ent 
C I i ent 
CI ent 
C I i ent 
CI ent 
C I i ent 
CI ent 
Message 
Received positive response for service Ox62 
Change to step [15] 
Received positive response for service Ox62 
parserMotFiIe [ data/ FlashDri ver . mot ] 
Mot Parser finish. Address [Ox 70000000] Blocksize [OXOOOOIOOO] DataSize [OXOOOOIOOO] S' 
segment : Address [0>00000000] size [Ox00001000] Offset [Oxoooooooo] CRC[OXOOOOOOOO] 
Segment DataQop[82008201] 
Segment DataEnd [00000000] 
parserMotFiIe [data/ÄppIicationB OTA. mot] 
Mot Parser finish. Address [OxÄ0504 
ockSize [Ox003FCOOO] DataSize Ox003FCOOO] S' 
Overview System 
CAPL / 
.NET 
Test 
Segment [O] : Address [OxA0504000] Size [Ox003FCOOO] Offset [OXOOOOIOOO] CRC[OXOOOOOOOO] 
Segment DataTop[OOOOOOOO] 
Segment O] : DataEnd 4A2EEIFA] 
Change to step [16] 
Progress [0/4182016] 
Received positive response for service Ox€e 
Change to step 17 ] 
Progress [0/4182016] 
Received positive response for service Ox 74 
wMaxNumherOfBI ockLength [ 20 5 0 ] 
Change to step 18] 
Progress [0/4182016] 
RPG aent 

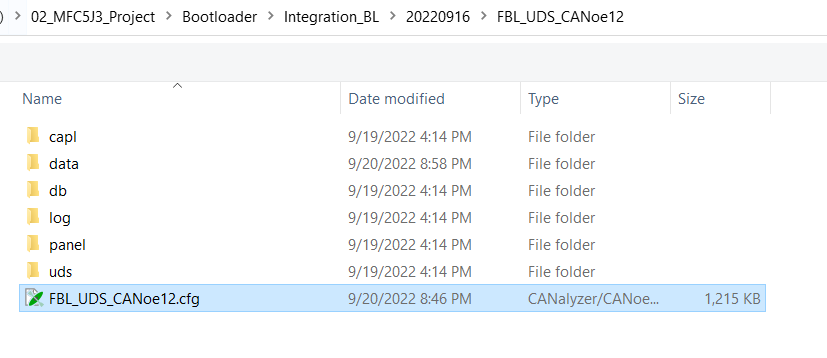
After 2min44s, it will present success, otherwise it is failure.

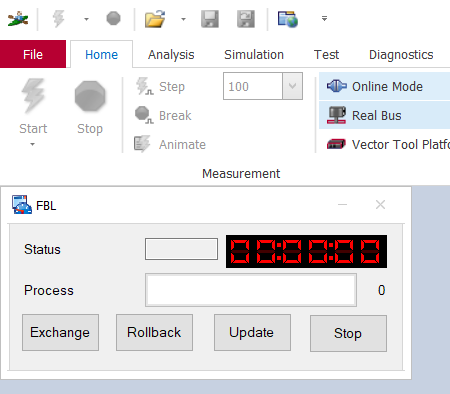


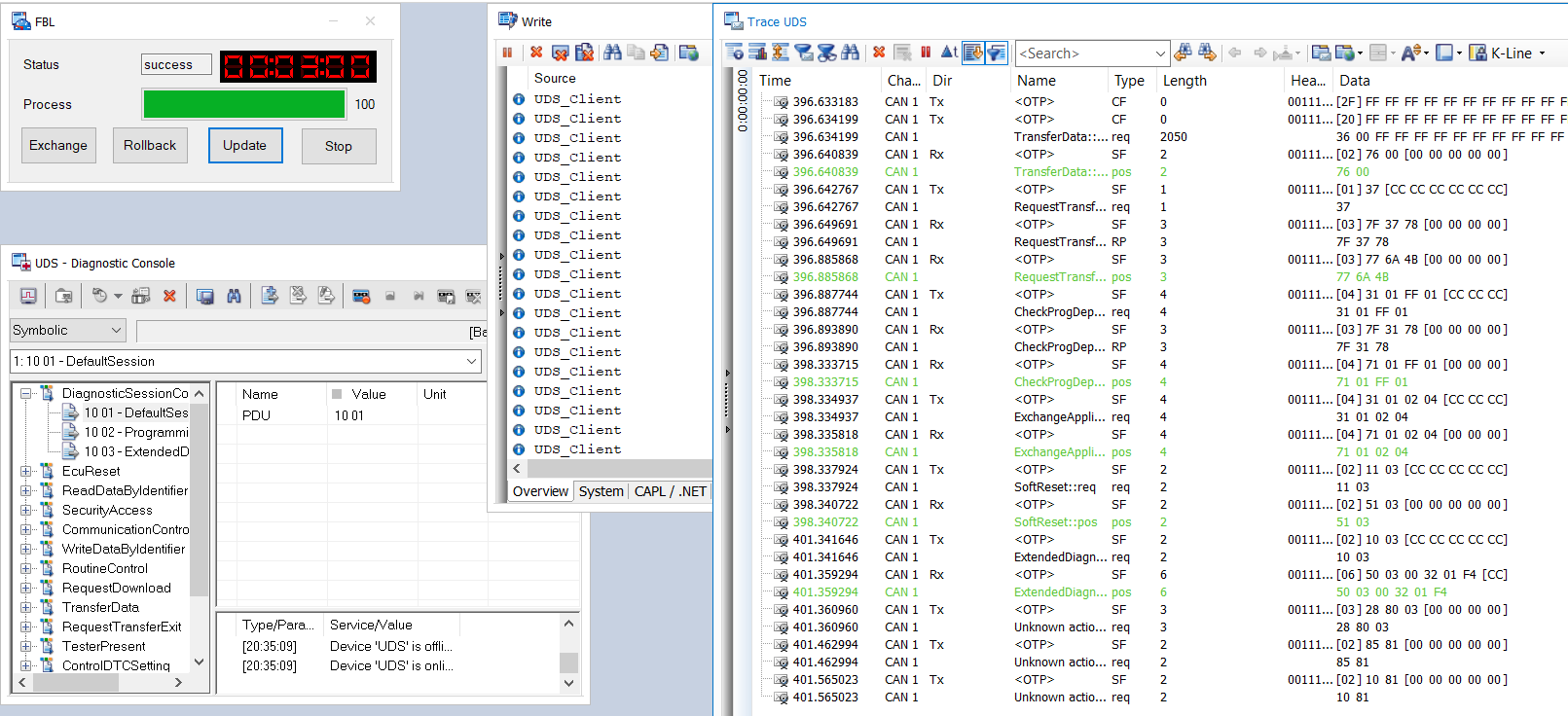
# Download Application with BootloaderV4.1

Use CANOE open FBL\_UDS\_CANoe12.cfg,perform update to downloader Application,A or B,

We need to update twice because it has Bank A and Bank B.





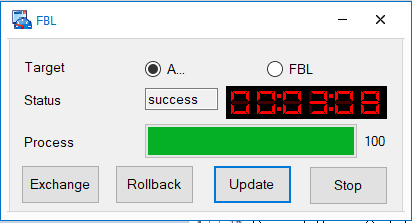


# Download Application with BootloaderV5.0

Use CANOE open FBL\_UDS\_CANoe12\_V5.0,perform update to downloader Application,A or B,

We need to update twice because it has Bank A and Bank B.

默认选择Application

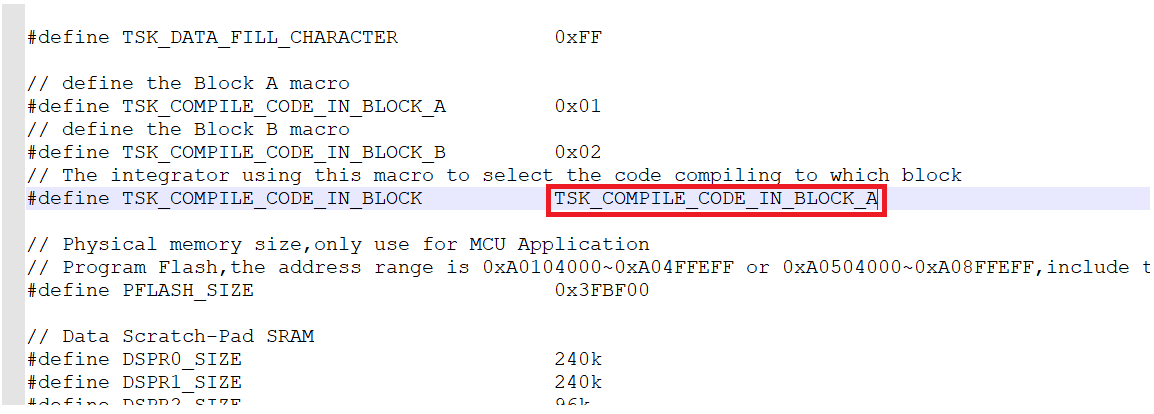


# Debug

The chapter describes how to use Lauterbach debugging base this bootloader.

# Set the Application Bank

We need to configure application location in **bank A** for the ld file**( suggestion)**

The path: **\01\_AsrConfig\source\boards\TriboardTC387TH\_DPM\board-tasking.ldscript**

## Removed the generated ld script

Removed the generated ld script which path : **\01\_AsrConfig\output\generated\TRICORE-tasking.ldscript**

## Compiling Code

Using the cmd command of make -j to compiling code

## Downloading NonSignature bootloader

Using Lauterach downloading non-signature version bootloader which the non-signature location in **\10\_Bootloader\01\_Bootloader\02\_Debug\_NonSignature\ Bootloader.hex**

## Downloading application elf/hex

Using Lauterach downloading application elf or hex file.

## Clean data flash

If the bootloader not jump to Bank A to perform application code,you shall clean the data flash or use the routine control to switch the bank.