Application Note

MSPM0 NONMAIN Implement

ABSTRACT

The MSPM0 NONMAIN is a specific memory region, NONMAIN can configure chip startup related parameters and extended function selection. Due to the particularity of NONMAIN, this article will provide CCS/Keil/IAR guidance and instructions for NONMAIN related operations.

Table of Contents

1 Overview of NONMAIN Features

The NONMAIN is a dedicated region of flash memory which stores the configuration data used by the Boot Configuration Routine(BCR) configuration and Bootstrap loader(BSL) configuration to boot the device. The region is not used for any other purpose. The BCR and BSL both have configuration policies which can be left at their default values (as is typical during development and evaluation), or modified for specific purposes (as is typical during production programming) by altering the values programmed into the NONMAIN flash region.

1.1 Terminology

Bootcode (BCR) - Startup Routine that runs after BOOTRST, to configure the device for executing application.

Bootloader (BSL) - Boot routine used to load data to the device memory.

BCR configuration - Configuration structure that contains all user configurable parameters for Bootcode, which resides in Non-main flash memory.

BSL configuration - Configuration structure that contains all user configurable parameters for Bootloader, which resides in Non-main flash memory.

1.2 NONMAIN Function

NONMAIN has two main functions, BCR configuration and BSL configuration.

BCR configuration - Configuration structure that contains all user configurable parameters for Bootcode, which resides in Non-main flash memory，NONMAIN can mainly achive these functions: Debug Security Policy Configuration，SWD Mass Erase and Factory Reset Configuration and Flash Memory Static Write Protection (SWP) Configuration.

BSL configuration - Configuration structure that contains all user configurable parameters for Bootloader, which resides in Non-main flash memory, NONMAIN can mainly achieve the configuration of BSL-related parameters including invoke pin，Communication interface(I2C or UART) , plugin configuration and Alternate BSL configuration.

Note

- Memory address ranges from 41C00000h ~ 41C00157h
- MSPM0 will solidify the default NONMAIN configuration at the factory
- If NONMAIN Flash is erased and user do NOT load new data to NONMAIN, then the device will fail to start, and can NOT connect anymore.

---

2 Nomain Architecture

This chapter provides a detailed introduction to the functions, uses, and applications of various options for NONMAIN.

2.1 MSPM0 family NONMAIN Function Resource Comparison

Table 2-1. MSPM0 NONMAIN function resource

NONMAIN Function			MSPM0Gx	MSPM0L1x	MSPM0Lx22x	MSPM0Cx
BCR	Debug Security Policy Configuration	Enable Debug	√	√	√	√
		Debug Password	√	√	√	×
		Debug Hold	×	×	√	×
		Enable TI Failure Analysis	√	√	√	×
	SWD Mass Erase and Factory Reset Configuration	Enable Mass Erase and Factory Reset	√	√	√	√
		Mass Erase and Factory Reset Password	√	√	√	×
	Flash Memory Static Write Protection (SWP) Configuration		√	√	√	√
	Other	Enable CSC Policy	×	×	√	×
		Enable Flash Bank Swap Policy	×	×	√	×
		Enable Fast Boot Mode	√	√	√	×
		Enable BSL	√	√	√	×
BSL	BSL Access		√	√	√	×
	BSL GPIO Invoke Configuration		√	√	√	
	BSL UART Pin Configuration		√	√	√	
	BSL I2C Pin Configuration		√	√	√	
	BSL Plugin Configuration		√	√	√	
	Alternate BSL Configuration		√	√	√	
	BSL Read Out Enable		√	√	√	
	BSL Security Alert Configuration		√	√	√	

2.2 Memory

The address ranges for the NONMAIN data structures are given in below table.

NONMAIN Section	Start Address	End Address
BCR Configuration	41C0.0000h	41C0.005Bh
BCR Configuration CRC	41C0.005Ch	41C0.005Fh
BSL Configuration	41C0.0100h	41C0.0153h
BSL Configuration CRC	41C0.0154h	41C0.0157h

2.3 BCR Function

The boot configuration routine is the first firmware to run on the device after a BOOTRST. The BCR manages the following at boot time:

- Configuring the debug interface security policy
- Optionally executing a mass erase
- Optionally executing a factory reset
- Configuring the flash memory static write protection policy
- Optionally verifying the integrity of some or all of the application firmware (with a 32-bit CRC)

• Optionally starting the bootstrap loader (BSL)

MSPM0 devices support three generic security levels: no restrictions (Level 0), custom restrictions (Level 1), and fully restricted (Level 2). Below table shows the three generic security levels, from least restrictive to most restrictive with the difference between each Level.

Level	Scenario	SW-DP Policy	App Debug Policy	Mass Erase Policy	Factory Reset Policy	TI FA Policy
0	No restrictions	EN	EN	EN	EN	EN
1	Custom restrictions	EN	EN, EN with PW, DIS	EN, EN with PW, DIS	EN, EN with PW, DIS	EN, DIS
2	Fully restricted	DIS	Don't care (access not possible with SW-DP disabled) (1)			

2.3.1 SWD Encryption and Decryption

The serial wire debug related policies configure the functionality which is available through the device's physical debug interface (SWD). By default, MSPM0 devices come from TI in an unrestricted state. This state allows for easy production programming, evaluation, and development. However, this unrestricted state is not recommended for mass production, as it leaves a large attack surface present. To accommodate a variety of needs while keeping the configuration process simple,

There are 4 main uses of the SWD interface for which protection needs to be considered:

• Application debug access – Debugging in the IDE tool

• Mass erase access - Erase the MAIN memory region

• Factory reset access - erase the MAIN memory region and reset the NONMAIN device configuration memory to TI factory defaults (Level 0)

• TI failure analysis access - Ability for TI to initiate a failure analysis

SWD supports the customization of four sets of passwords, which can be unlocked via CCS Scripts PasswordAuthentication or Factory Reset function, as well as via the BSL Host for SWD Encryption.

SWD Encryption is mainly used to protect chips from malicious connections, tampering, or memory reading during the mass production stage. Once SWD Encryption is set, the debugger will not be able to connect to the chip without the correct password configuration.

2.3.2 SWD Mass Erase and Factory Reset

The BCR provides mass erase and factory reset functionality through commands sent to the device over SWD from a debug probe using the debug subsystem mailbox (DSSM). These commands are not available in SWD security level 2, but they are optionally available in security level 0 and 1. When the device is not configured for SWD security level 2, the mass erase and factory reset commands can be individually configured to be enabled, enabled with a unique 128-bit password, or disabled. By default, both commands are enabled.

The SWD mass erase and factory reset DSSM commands superseded any static write protection policies. For example, if SWD factory reset is configured to be enabled or enabled with password, the BCR configuration data can be reset even if it is statically write protected.

SWD Mass Erase

A SWD mass erase is an erase of the MAIN flash regions only, which typically includes the user application. The BCR and BSL policies stored in the NONMAIN flash region are not affected by a mass erase.

A mass erase is useful for erasing all application code and data while leaving the device configuration itself intact.

SWD Factory Reset

A SWD factory reset is an erase of the MAIN flash regions followed by a reset of the NONMAIN flash region to default values.

Such an erase is useful for completely resetting the BCR and BSL device boot policies while also erasing the application code and data.

SWD Mass Erase and Factory Reset support the configuration of four sets of passwords respectively. After configuring the password, Mass Erase and Factory Reset cannot be performed through the Scripts in CCS. However, Factory Reset operation can be carried out through the online Factory Reset tool.

2.3.3 Flash Write Static Protection

The flash memory protection and integrity policies specify which sectors of flash memory are locked from modification, as well as which sectors are to be checked for integrity during the boot process before the user application is started.

Key features of flash protection:

- Locking the Application (MAIN) Flash Memory
  By configuring FLASHSWP0(Low 32kB) and FLASHSWP1(High 32kB) fields in the NONMAIN memory, This will can protect the corresponding area of the FLASH.

Note

One bit corresponds to one sector with the LSB being Sector 0. Setting a bit to 0 enables write protection, and setting a bit to 1 disables write protection.For example, on MSPM0 device with 32kB of flash or more and a sector size of 1kB, the FLASHSWP0 setting defines the protection of the lower 32kB.A value of 0x7FFFFFFE (bits 31 and 0 cleared) would make all sectors writable, except for sector 0 (0x0000_0000-0x0000_003FF) and sector 31 (0x0000_7C00-0x0000_7FFF). Likewise， FLASHSWP1 controls the write protection of the upper 32kB of FLASH.

- Static Write Protection NONMAIN Fields
  When configured to be protected, the entire NONMAIN region will be write-locked and will be functionally immutable when the boot configuration routine transfers execution to either the bootstrap loader or the user application code in MAIN flash. Any attempt to program or erase the NONMAIN by the application code or the bootstrap loader will result in a hardware flash operation error, and the sector will not be modified.

Note

Factory reset commands sent to the BCR via the debug sub system mailbox (DSSM) will override the specified static write protection policy. If this behavior isn’t desired, configure the mass erase and factory reset commands to be enabled with password or disabled. Note that mass erase and factory reset commands sent to BSL will respect the specified static write protection policy (the BSL has the same permissions as application code).

2.4 BSL Function

The bootstrap loader (BSL) provides a means to program and verify the device memory through a standard serial interface (UART or I2C), as opposed to the serial wire debug interface. The BSL has its own configuration policy, but the BCR determines if the BSL is enabled to be invoked, or if it is to be disabled (non-invokable).Since the BSL presents an additional attack surface, if it is not used in an application it may be disabled in the user-specified boot security policies. If the BSL is used in an application, then the BSL security settings (including the BSL access password) are managed in the BSL configuration policy. When the BSL is disabled, it is not possible to enter the bootloader through any invocation mechanism.

The bootstrap loader (BSL) provides a method to program and/or verify the device memory through a standard UART or I2C serial interface. Key features of the BSL which are accessible through the serial interface include:

- Programming and erase of flash memory
- Ability to return a firmware version number through a pointer to the main flash
- Ability to specify a hardware invoke GPIO
- Ability to enable code/data read-out (disabled by default)
- Ability to return a 32-bit CRC of a code/data region (1KB minimum region size) to verify programming
- Access is always protected with a 256-bit password
- Configurable security alert handling for resisting brute force attacks
- Support for MAIN flash plug-ins to enable additional interfaces beyond UART and I2C

2.4.1 Invoke Pin

The bootloader supports hardware invoking after a BOOTRST through the use of a GPIO. The BSL configuration in the NONMAIN flash memory contains the pad, pin, and polarity definition for the GPIO invocation. Devices come configured from TI for a specific GPIO and polarity, but software can change this default by modifying the GPIO pin configuration in the BSL configuration in NONMAIN flash memory.

2.4.2 Conmunication Interface

MSPM0 support configurate I2C or UART as the serial wire debug interface. Users can customize the UART or I2C pins. The default pin configuration for the device can be found in the Signal Description section of the datasheet. The following figure shows the default pin configuration for the BSL communication interface in the MSPM0L series.

BSL	BSL_invoke	22	21	14	14	14	10	I	Input pin used to invoke bootloader
BSL (I^2C)	BSLSCL	2	5	1	5	5	4	I/O	Default I^2C BSL clock
	BSLSDA	1	4	24	4	4	3	I/O	Default I^2C BSL data
BSL (UART)	BSLRX	26	25	18	17	1	13	I	Default UART BSL receive
	BSLTX	27	26	19	18	2	14	O	Default UART BSL transmit

Note

In terms of configuring UART and I2C, BSL only supports the use of UART0 and I2C0.

2.4.3 BSL Access Password

Access to the BSL is always protected by a 256-bit user-specified password. There is no option to disable the password. The password must be provided to the BSL after invocation for access to most BSL functions to be granted. When the password is not provided, the only BSL commands allowed are Get Identity and Start Application.If a wrong password is provided to the BSL, the BSL halts for 2 seconds, after which an additional attempt can be made to send the correct password. After three failed password attempts, the security alert function is activated.

2.4.4 BSL Plugin

The BSL supports MAIN flash plugins to enable additional interfaces beyond UART and I2C. When this feature is enabled, the BSL core will call the corresponding plugin functions (Init, Receive, Transmit and De-Init).

BSL Plugin can be used to custom BSL special function(Mainly including UART or I2C communication layer) defined by customer. The MSPM0 SDK includes code examples illustrating the implementation of plugins.

2.4.5 Alternate BSL Configuration

The MSPM0 bootcode can jump to an alternate BSL if enabled, allowing developers to implement custom bootloaders in MAIN flash. The specified address of the alternate BSL must be valid.

Alternate BSL allowed customer use a custom BSL instead of ROM BSL. The MSPM0 SDK includes code examples illustrating the implementation of alternate bootloaders.

As for MSPM0 BSL detail application, please refer to MSPM0 Bootloader User’s Guide**.**

3 NONMAIN Configuration by SysConfig

This chapter mainly introduces how to configure the various functional options of the MSPM0 NONMAIN region through the graphical configuration tool Sysconfig.

3.1 SysConfig Introduction

SysConfig is an intuitive and comprehensive collection of graphical utilities for configuring pins, peripherals, radios, subsystems, and other components.
SysConfig helps you manage, expose and resolve conflicts visually so that you have more time to create differentiated applications.

The tool's output includes C header and code files that can be used with software development kit (SDK) examples or used to configure custom software.
The SysConfig tool automatically selects the pinmux settings that satisfy the entered requirements.

The SysConfig tool is delivered as a standalone installer, integrated in CCS, it can be manually integrated into IAR and Keil, or can be used via the dev.ti.com cloud tools portal.

In Sysconfig, you can click on the "?" at the end of the configuration item to view the description and explanation of the configuration item.

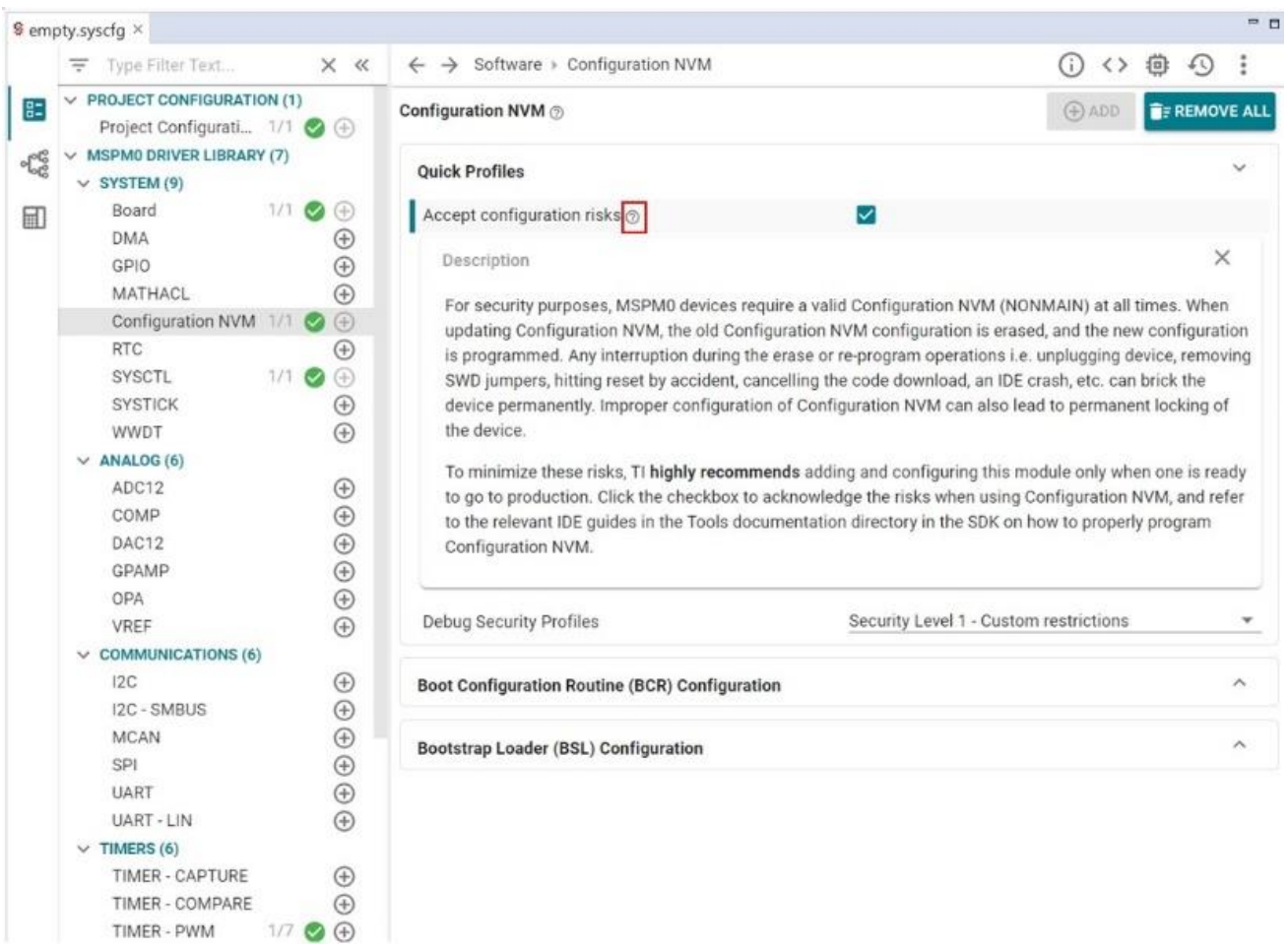

Figure 3-1. Sysconfig page

After completing the relevant configuration in sysconfig and compiled, the main program will automatically generate the SYSCFG_DL_init() function, which contains all the initialization code for the configurations completed in sysconfig. If NONMAIN was configured, boot_config.c and boot_config.h will automatically generated in the Sysconfig folder.

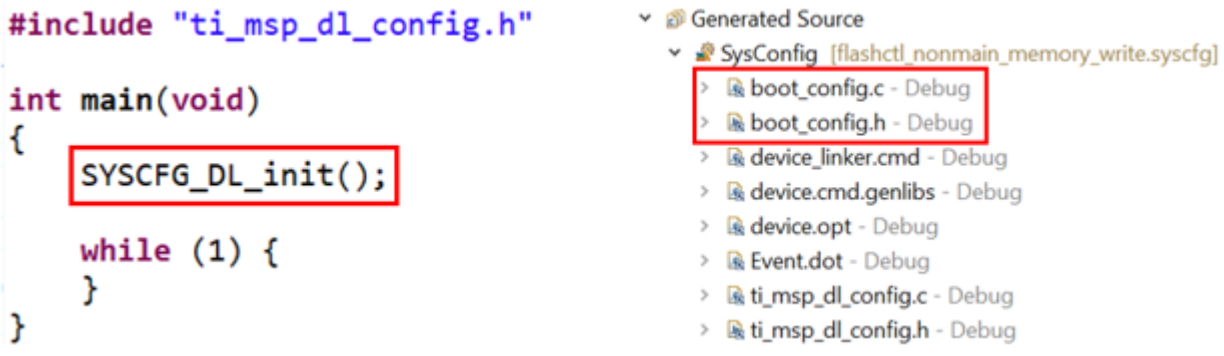

Figure 3-2. Sysconfig generated source

3.2 BCR Configuration

3.2.1 SWD Encryption and decryption Operation

In the Debug Security Policy Configuration, there are mainly three configurable items:

- Enable or disable SW-DP controls the activation and deactivation of the SWD interface.
- Enable or disable SWD Access and configure it to enable with a password (customizable with four sets of 32-bit SWD Password).

- Ability to enable or disable TI Failure Analysis, allowing TI to initiate a failure analysis return flow through SWD.

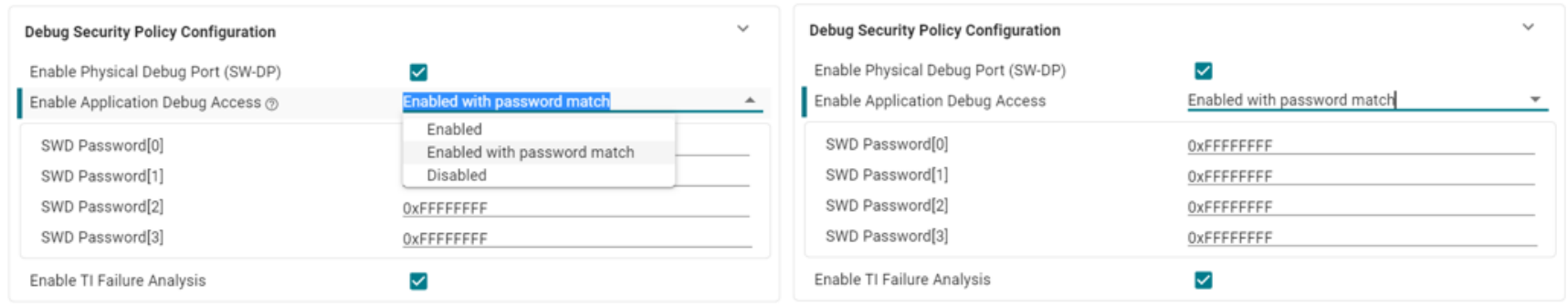

Figure 3-3. Debug Security Policy Configuration

3.2.2 SWD Mass Erase and Factory Reset Operation

SWD Mass Erase and Factory Reset Operation can enable or disable the chip's Mass Erase and Factory Reset functions, and configure them to enable with a password (both Mass erase and Factory reset can be customized with four sets of 32-bit SWD Password).

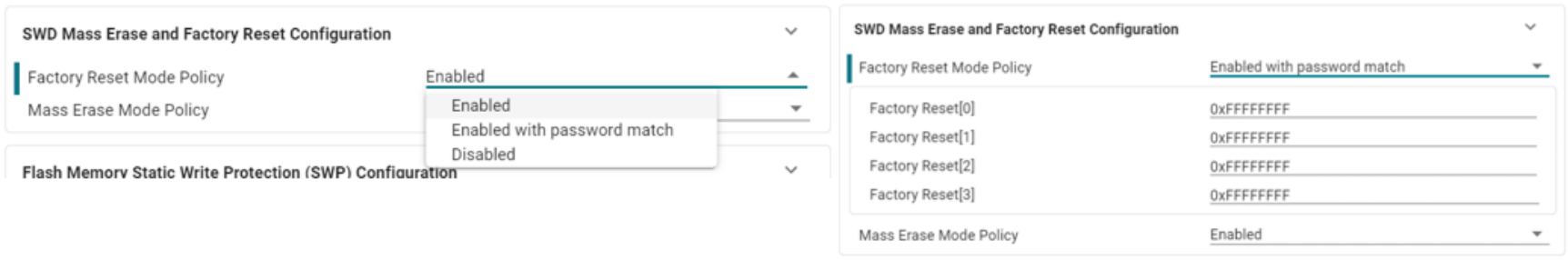

Figure 3-4. SWD Mass Erase and Factory Reset Configuration

3.2.3 Flash Write Protection Operation

Flash Write Protection Operation has three items can be configured:

- MAIN SWP(Lower Sectors) - Defines the protection of the lower 32kB
- MAIN SWP(Remaining Sectors) - Defines the protection of memory above 32kB
- NONMAIN Static Write Protection – Lock of NONMAIN region

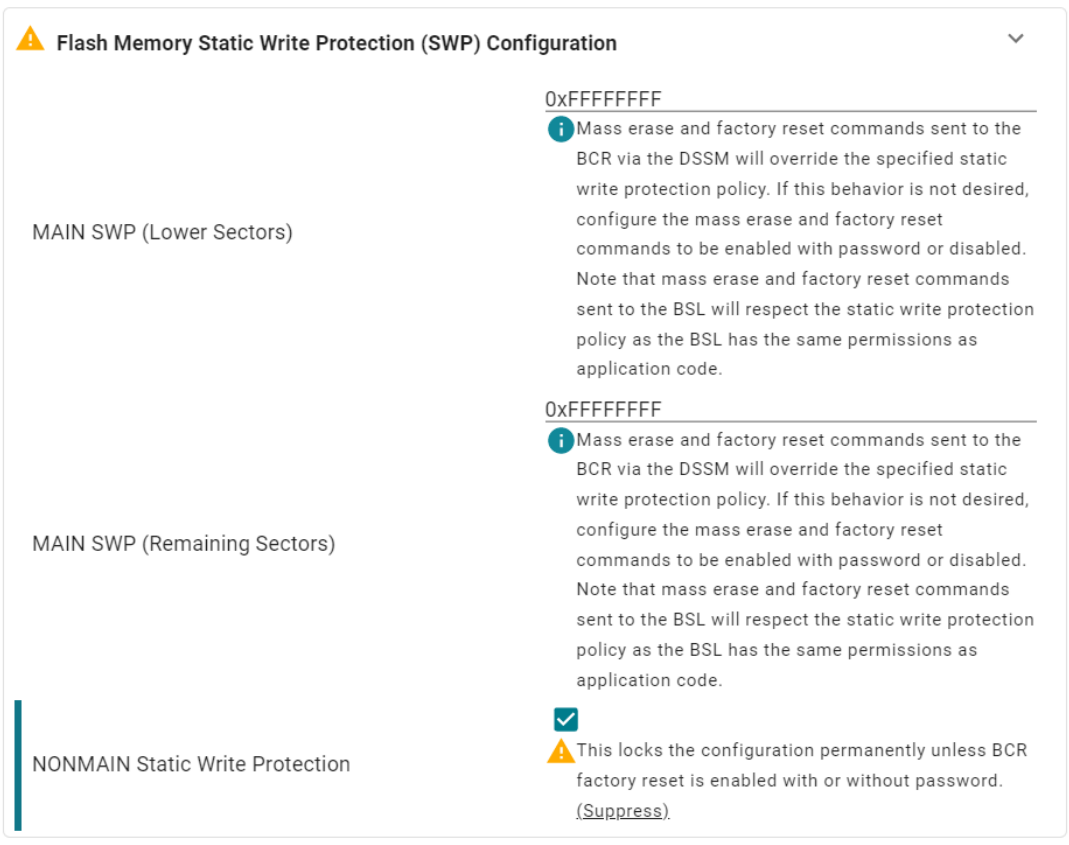

Figure 3-5. Flash SWP Configuration

Note

- Mass erase and factory reset commands sent to the BCR via the DSSM will override the specified MAIN SWP policy. If this behavior is not desired, configure the mass erase and factory reset commands to be enabled with password or disabled.
- NONMAIN SWP locks the configuration permanently unless BCR factory reset is enabled with or without password.

3.2.4 Other Configuration Options

Other configuration of BCR:

- Enable Fast Boot Mode - Fast boot, will bypass CRC check
- BCR Configuration ID - default parameter
- Expected BCR Configuration CRC - default parameter, If the stored CRC of the BCR configuration doesn't match the calculated CRC during boot, the result is a boot error
- Enable BSL - Define BSL Enable or Disable

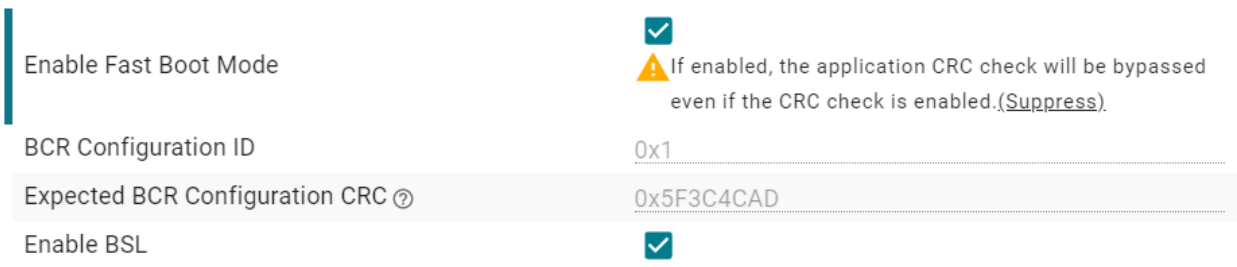

Figure 3-6. Rest of BCR Configuration

3.3 BSL Configuration

3.3.1 BSL Access Configuration

There are eight groups 32bit BSL Access Password, Eight goups of passwords are independent of each other and collectively protect the access permission of the BSL.

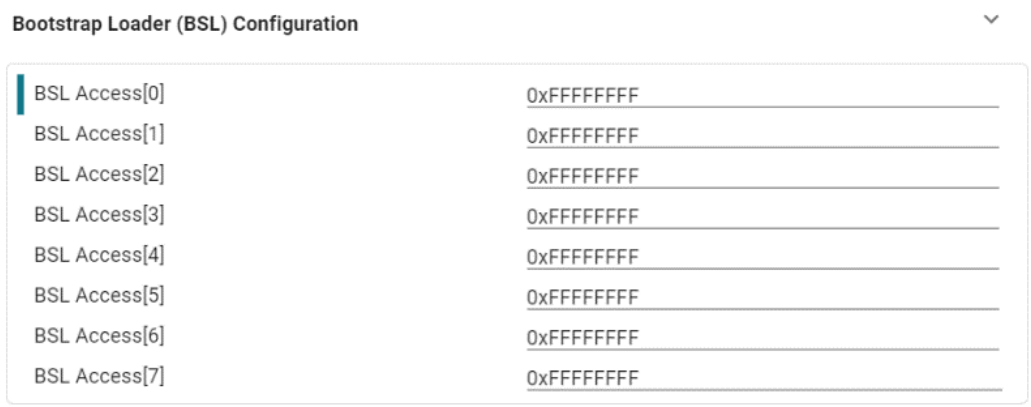

Figure 3-7. BSL Access Configuration

3.3.2 BSL Invoke Pin Configure

In Sysconfig, you can enable or disable the BSL Invoke Pin and choose to use the default Invoke Pin or a custom one. Additionally, you can configure the active level of the Invoke Pin.

Figure 3-8. BSL GPIO Invoke Configuration

3.3.3 BSL Communication Interface Pin Configuration

MSPM0 BSL supports two communication modes: UART and I2C. The BSL will automatically detect whether the host computer is using UART or I2C, so you only need to define the respective pins for each communication mode in Sysconfig.

Figure 3-9. BSL Comunication Interface Configuration

3.3.4 BSL Plugin Configuration

BSL Plugin Configuration defined that BSL provides an option to add custom interface implementation to the ROM BSL core as a Flash Plug-in. This gives an advantage of customizing the interface, without reimplementing the complete BSL core. More details and usage can see MSPM0 Bootloader User's Guide.

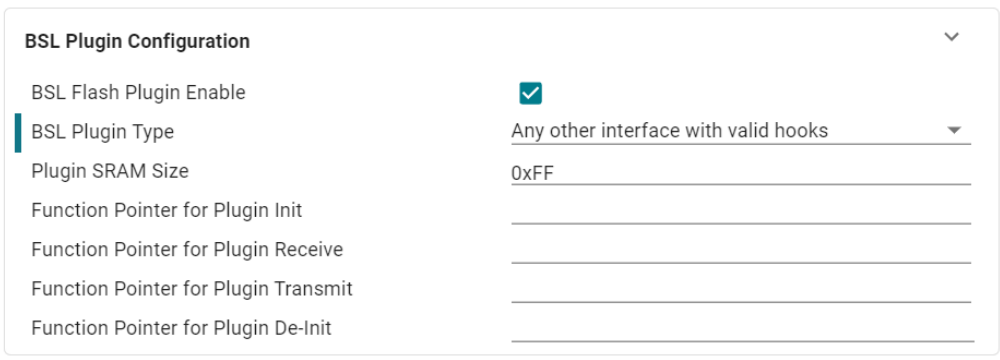

Figure 3-10. BSL Plugin Configuration

3.3.5 BSL Alternate Configuration

sss The alternate BSL address should be the address of the first function to get executed in the secondary BSL code. This function should be placed in a fixed location. Refer to the secondary_bsl SDK example inside the SDK for more details.

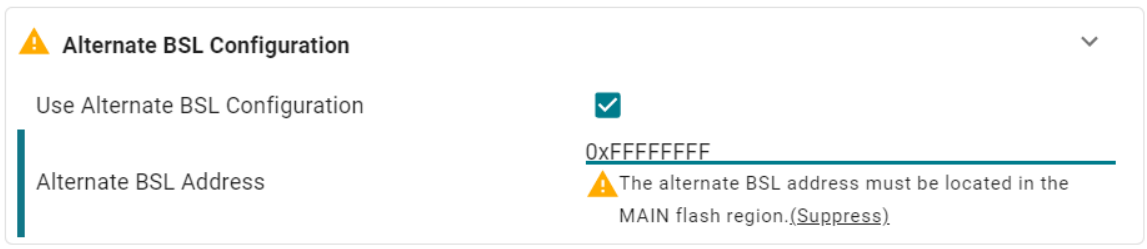

Figure 3-11. Alternate BSL Configuration

3.3.6 Other Configuration Options

Other configuration of BSL:

- BSL Configuration ID - default parameter
- BSL App Version - Custom version number, the BSL supports returning an application version number through the BSL serial interface
- BSL Read Out Enable - The BSL can be configured to allow read-out of memory locations through the BSL serial interface
- BSL Security Alert Configuration - The BSL can be configured to respond to the security alert in one of the three ways: Ignore / Disable BSL / Trigger Reset

- Expected BSL Configuration CRC - default parameter

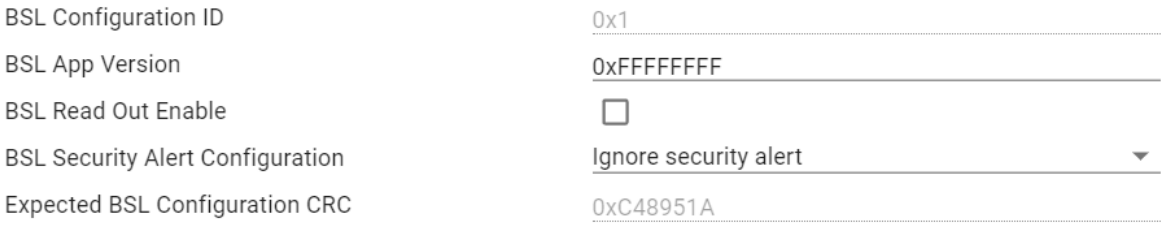

Figure 3-12. BSL Comunication Interface Configuration

4 NONMAIN Operation on CCS

This chapter mainly introduces the operational process of configuring the various function options of the MSPM0 NONMAIN region in CCS, as well as the usage case of key functions.

4.1 NONMAIN Operation Project Implement

The steps to configure and program the NONMAIN region in CCS are as follows:

- Complete the configuration of various items in Sysconfig, save and compile.

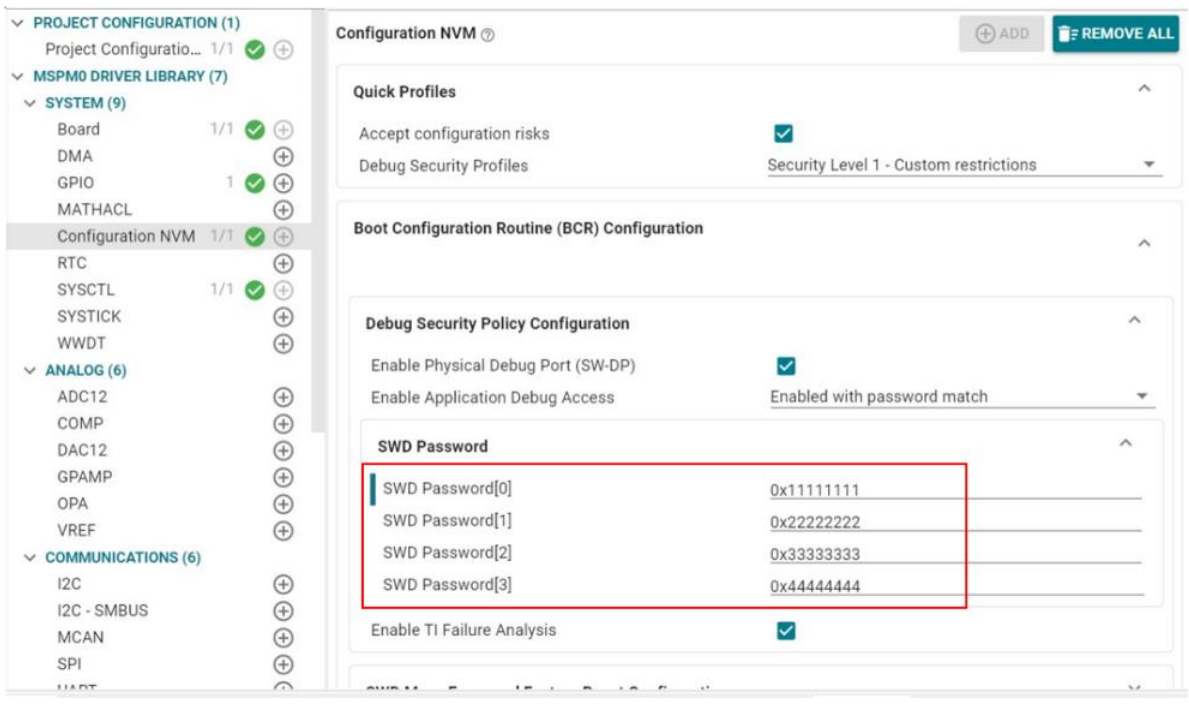

Figure 4-1. SWD Password Configuration

- Two NONMAIN configuration-related files, boot_config.c and boot_config.h, will be generated in the Sysconfig directory.

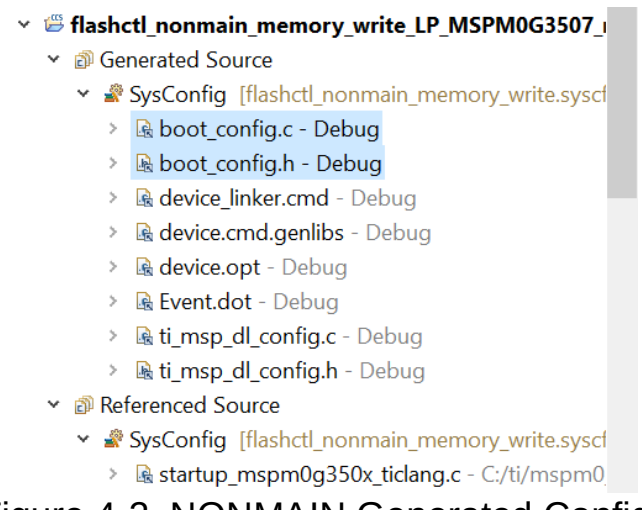

Figure 4-2. NONMAIN Generated Configuration

- Modify the Flash erase region setting in the project properties.

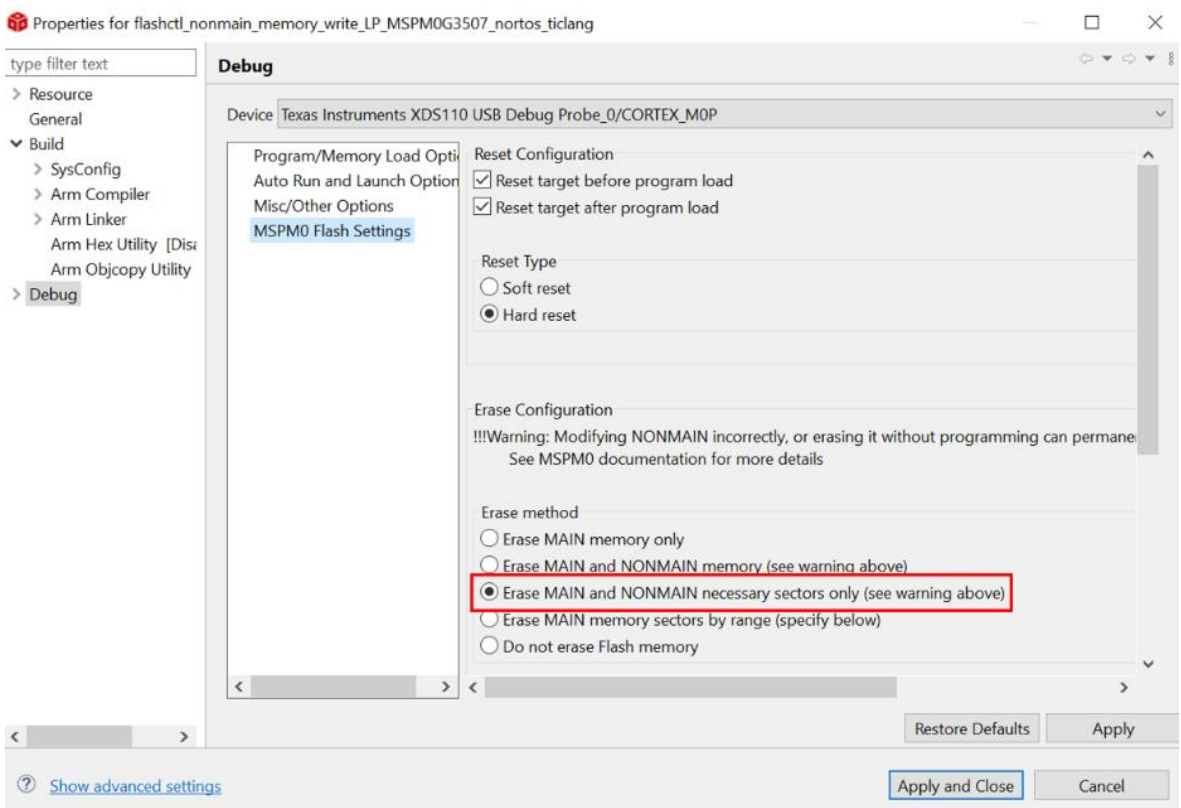

Figure 4-3. Flash Setting Configuration

- Execute the NONMAIN programming steps in the main program:
  - First of all, unlock the NONMAIN region and then erase it.
  - Unlock the NONMAIN region and write the configuration data from boot_config.c to the BCR region.
  - Unlock the NONMAIN region and write the configuration data from boot_config.c to the BSL region.

Note

- After each programming process, the Flash write protection will automatically reactivate, so it needs to be disabled again before the next programming.
- For a demonstration of NONMAIN operation, refer to flashctl_NONMAIN_memory_write project in the MSPM0 SDK

4.2 BCR Configuration

4.2.1 SWD Encryption and decryption Operation

4.2.1.1 XDS110 Operation

SWD Encryption Procedure:

- Configure four sets of SWD Password in Sysconfig.
- After compiling the project, burn it into the chip.
- When the device is disconnected and powered on again, attempt to burn the chip which will prompt the following message indicating successful encryption:

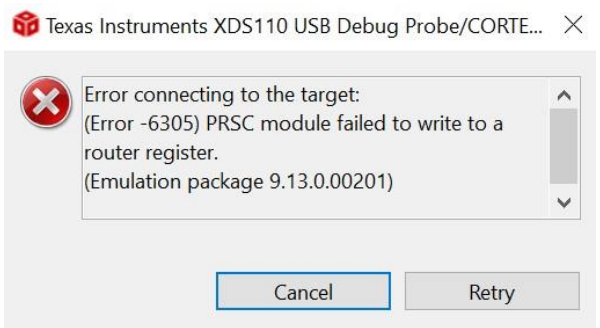

Figure 4-4. SWD Encryption Block Window

SWD Decryption Procedure:

- Double-click on the ccxml file in the project to open it.

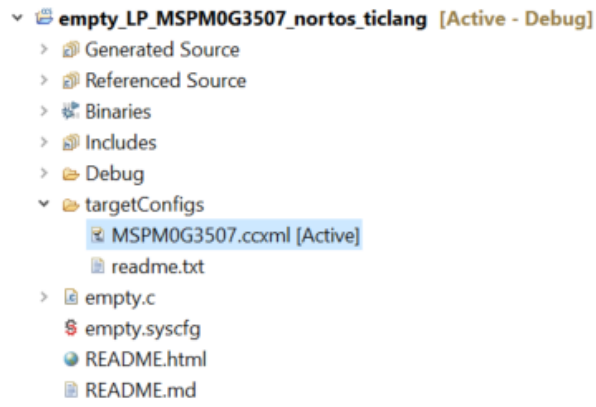

Figure 4-5. ccxml Configuration File

- Go to Target Configuration.

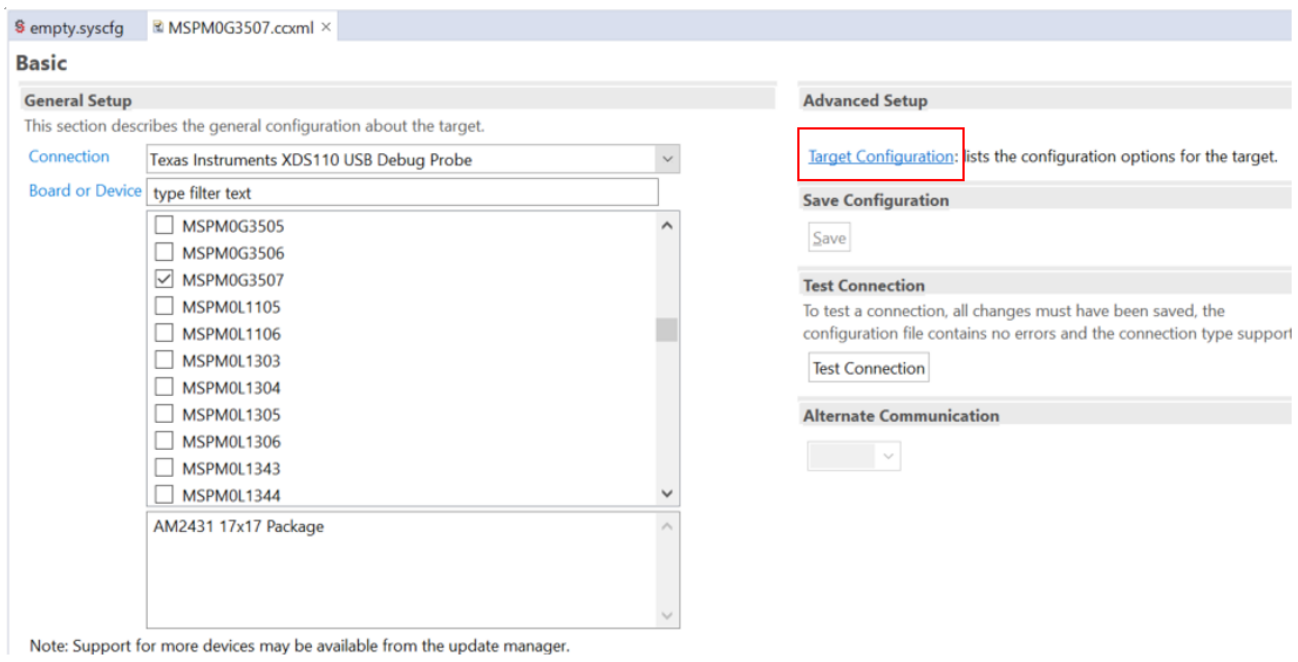

Figure 4-6. ccxml Target Configuration

- Click on the device model MSPM0xxx, and enter the four sets of SWD Password one by one.

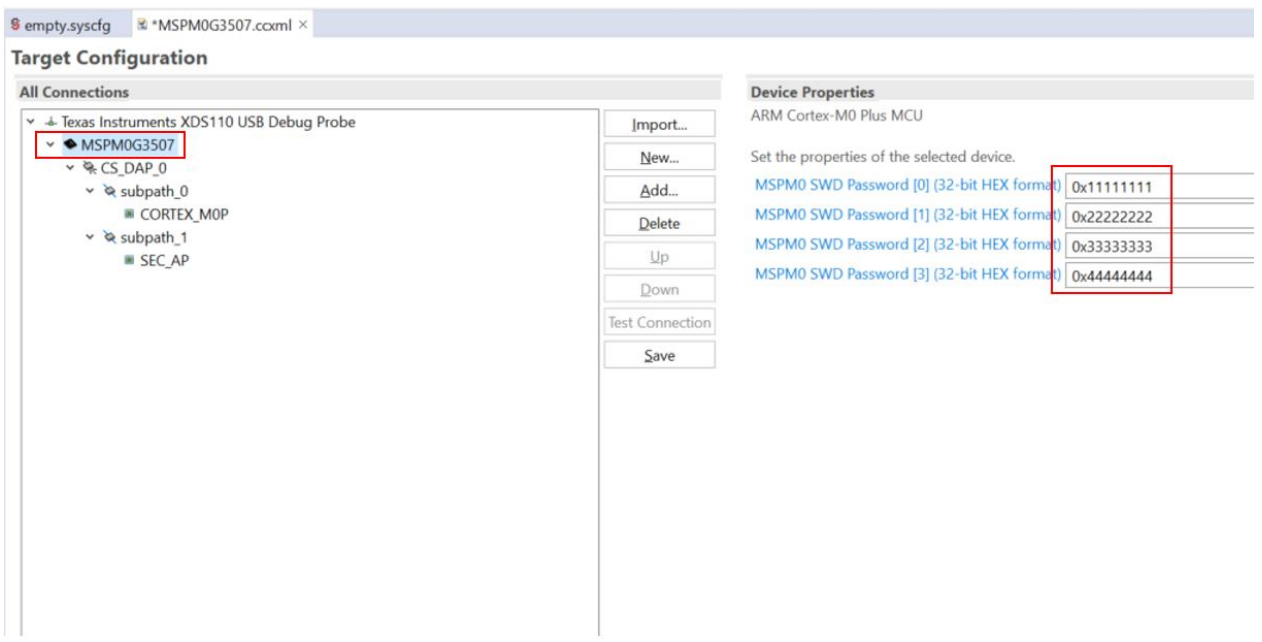

Figure 4-7. SWD Password Input Page

- In Target Configuration, find the corresponding ccxml file of the project, right-click, and select Launch.

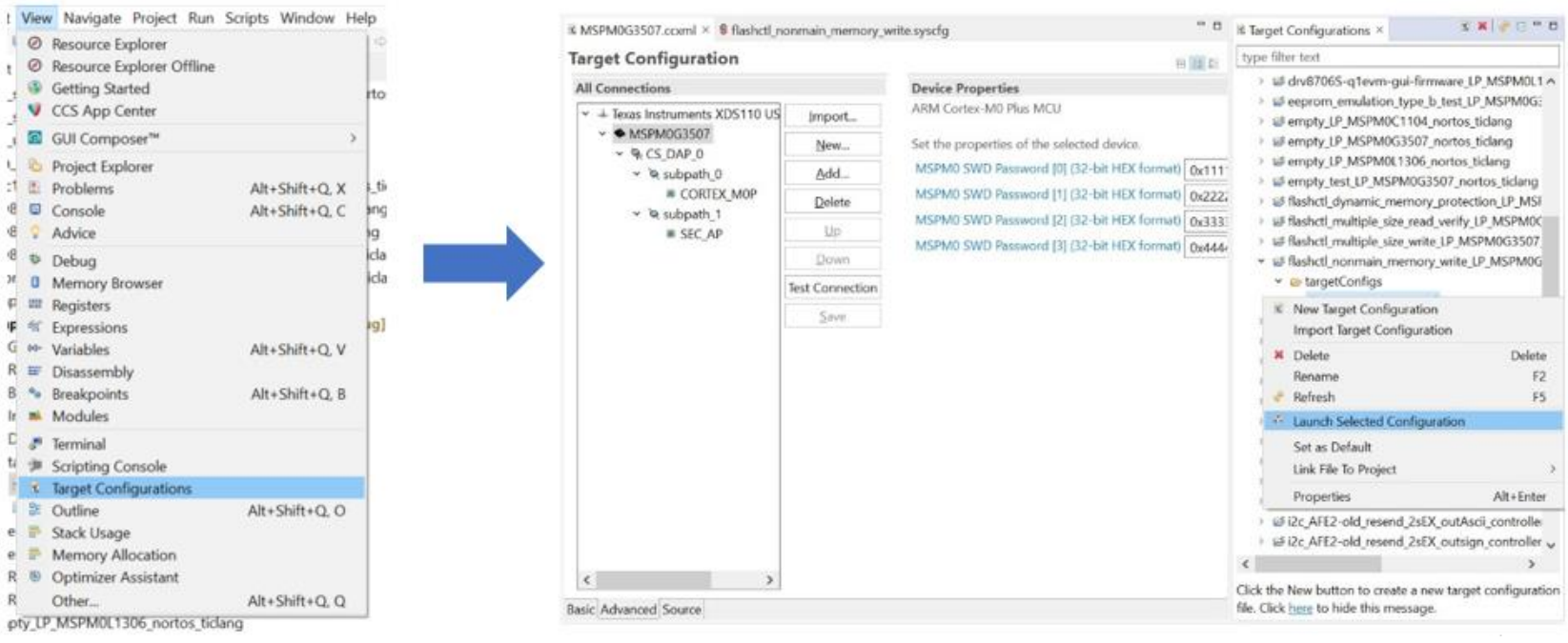

Figure 4-8. Launch ccxml File

- Choose MSPM0_Maibox_PasswordAuthentication_Auto from the Scripts in the navigation bar.

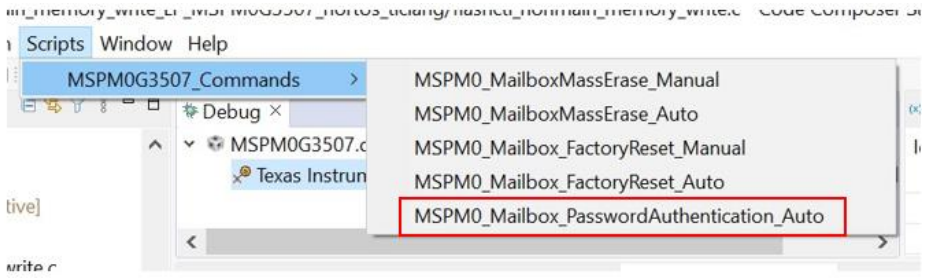

Figure 4-9. CCS Scripts Selection

- The following picture shows the successful unlocking of SWD

Figure 4-10. CCS SWD Unlock Succeed Blog

In addition to using the PasswordAuthentication command in the Scripts, SWD can also be unlocked using Factory Reset and BSL Host.

4.2.1.2 J-Link Operation

SWD Encryption phenomenon:

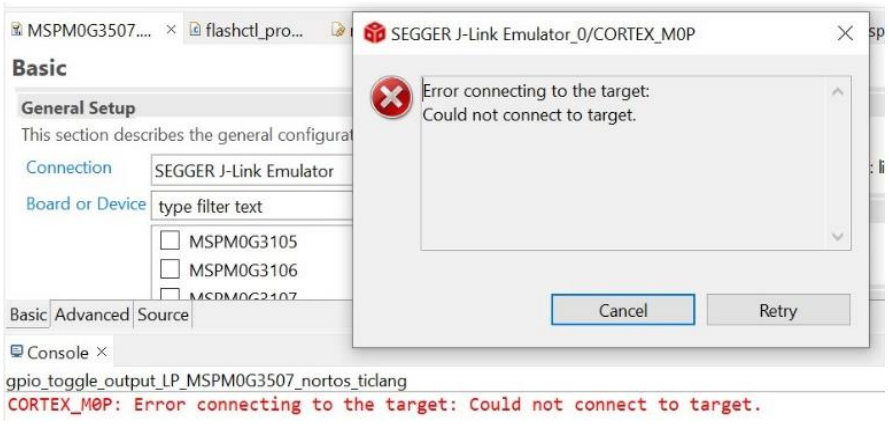

Figure 4-11. J-Link SWD Encryption Block Window

For J-Link users, it is possible to use the XDS110 with the CCS Scripts mentioned in Chapter4.2.2 or the Factory Reset online tool to unlock SWD and remove FLASH Protect. In addition, unlocking can also be done using the BSL Host Implement mentioned in Chapter4.3.1. Furthermore, TI will gradually support the use of Scripts with J-Link.

Note of SWD Encryption:

- J-Link cannot use the Scripts command function in CCS.
- SWD PSW need cut out the power and then take effect
- After unlocking with a password using SWD in CCS, the chip will remain unlocked as long as it is not reset.
- After unlocking with a password using SWD in CCS, if the NONMAIN password configuration is not disabled, the password will still be active after the chip is reset.
- If the compiler's Flash erase range setting is not modified, CCS will display an error after programming the NONMAIN operation code.

Figure 4-12. Flash Erase Range Error Blog

4.2.2 SWD Mass Erase and Factory Reset Operation

SWD Mass Erase and Factory Reset Procedure

- Hardware connection:
  Connect the USB interface to PC, For MSPM0 Launchpad with default configuration, There are UART, NRST and BSL trigger pin jumper interface reserved:

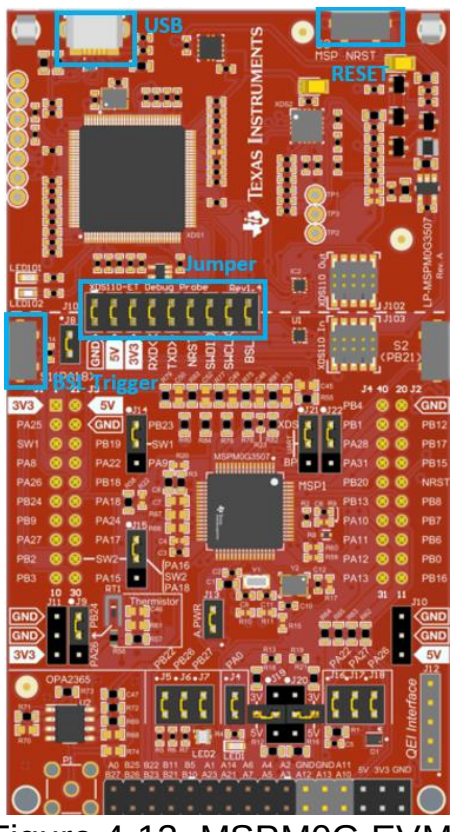

Figure 4-13. MSPM0G EVM Board

- Launch the project target configuration refer to the previous content
- Right-click on the XDS110 USB Debug Probe in the Debug window, then click on "Show all cores".

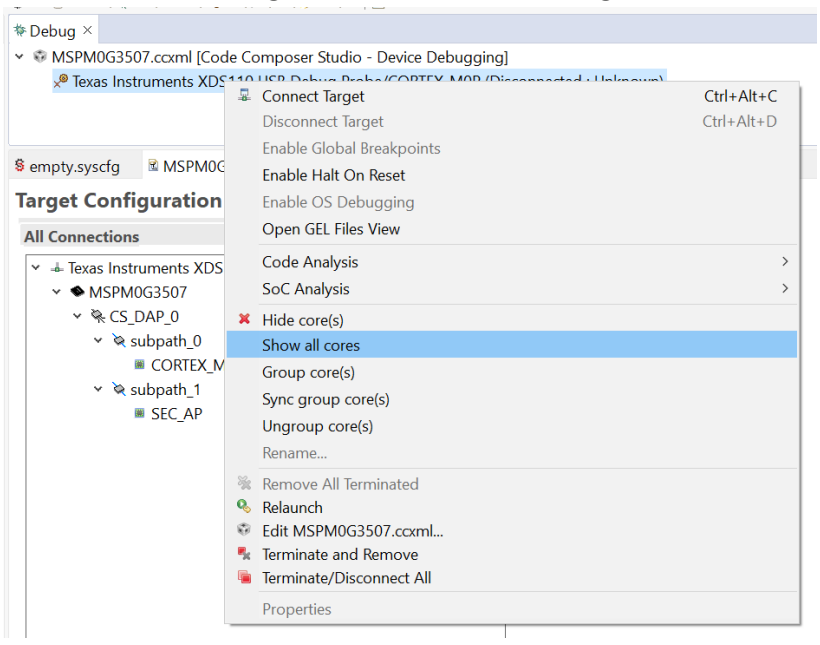

Figure 4-14. Show All Core in Debug Mode

- Choose CS_DAP，Connect Target

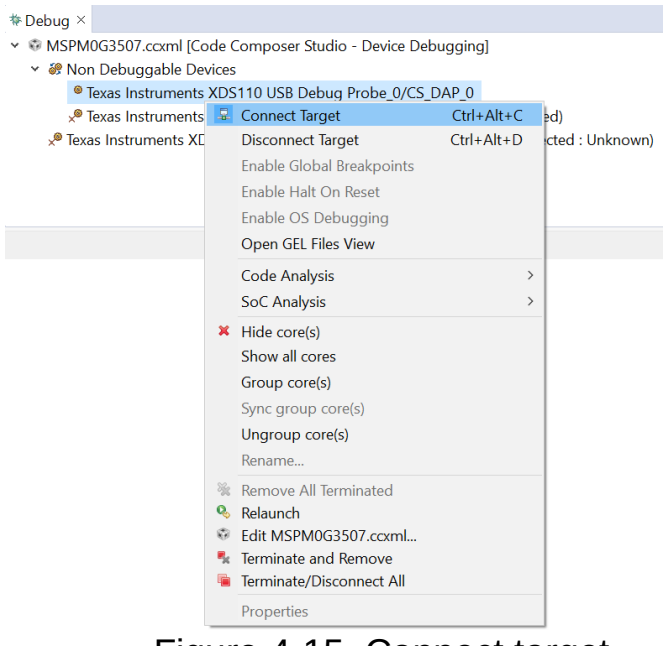

Figure 4-15. Connect target

- Choose Scripts Mass Erase or Factory Reset

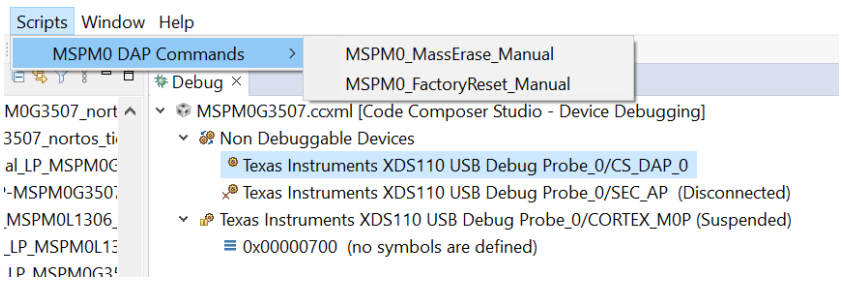

Figure 4-16. Scripts to Factory Reset

- Pull NRST pin low when appears this windows

Figure 4-17. Waiting for Reset

- The successful execution interface is as below

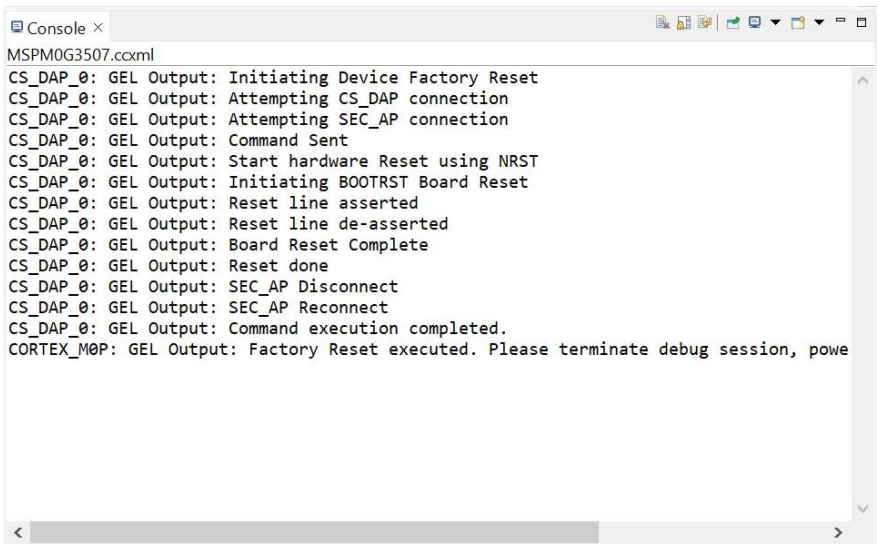

Figure 4-18. Fctory Reset Succeed Blog

In addition, TI provides an online tool for performing a Factory Reset operation on the chip. You can find this tool in dev.ti.com/gallery/view/TIMSPGC/MSPM0_Factory_Reset_Tool. This toolkit consists of two parts shown as below Figure 4.4, Info and DSSM. Info provides the usage instructions for the toolkit, while DSSM is the graphical user interface (GUI) for the toolkit.

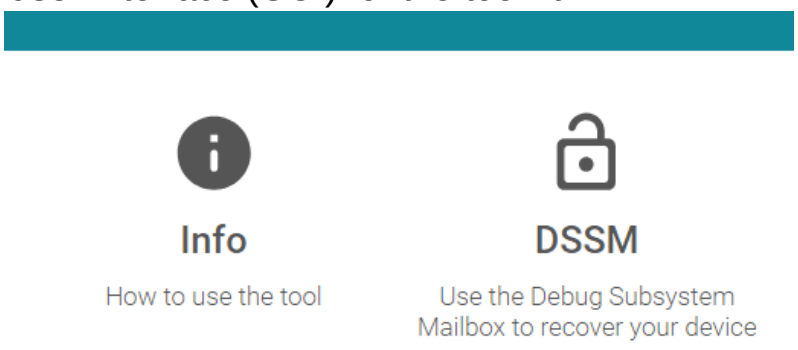

Figure 4-19. Factory Reset Tool Online

4.2.3 Flash Protection Operation

Flash SWP Configuration Procedure:

- Configurate two SWP in the Sysconfig, SWP(Lower Sector) = 0xFFFFFFFB will lock the third sector(0x800 ~ 0xBFF)

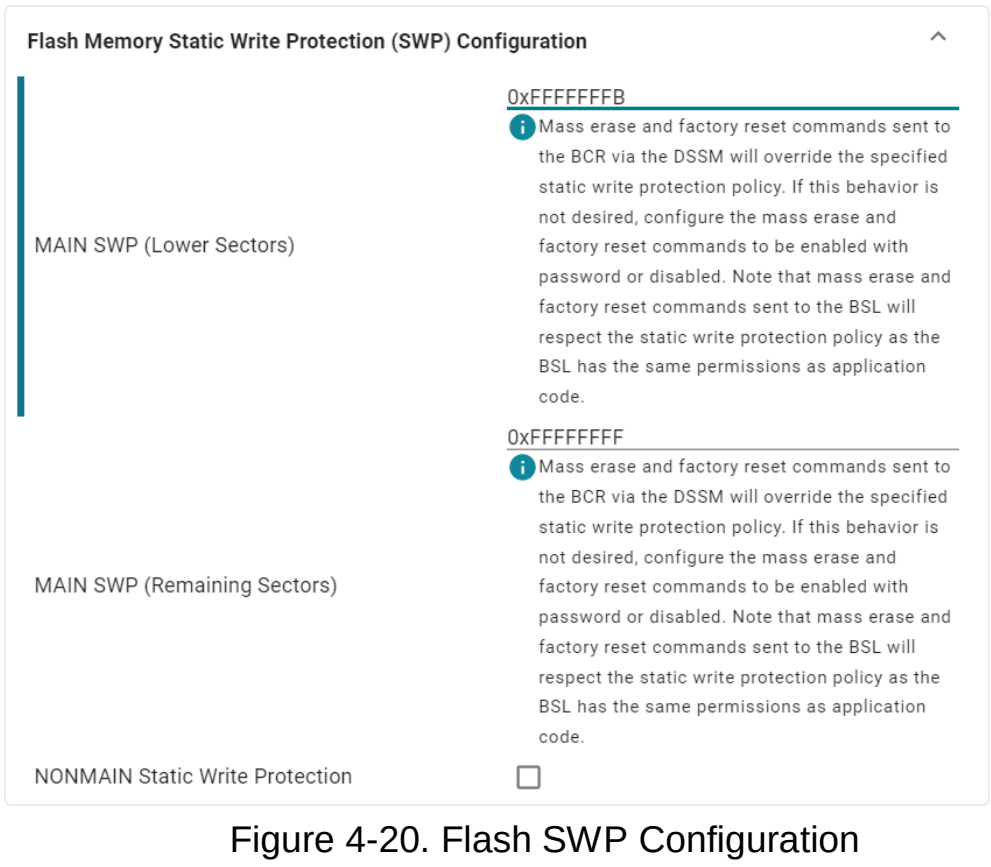

Figure 4-20. Flash SWP Configuration

- Download this configuration to device, and the third sector(0x800 ~ 0xBFF) has been locked

The resulting phenomenon:

- In another CCS project, attempt to write 0x11 to address 0x800, then burn and execute the program.

```
#define MAIN_BASE_ADDRESS       (0x00000800)

uint8_t gData8 = 0x11;

DL_FlashCTL_unprotectSector(
    FLASHCTL, MAIN_BASE_ADDRESS, DL_FLASHCTL_REGION_SELECT_MAIN);
gCmdStatus = DL_FlashCTL_programMemoryFromRAM8WithECCGenerated(
    FLASHCTL, MAIN_BASE_ADDRESS, &gData8);
```

Figure 4-21. Try to Write Protected Address

- In CCS Debug mode, check Memory Browser and confirm that address 0x800 was not successfully written with 0x11, indicating successful Flash SWP configuration.

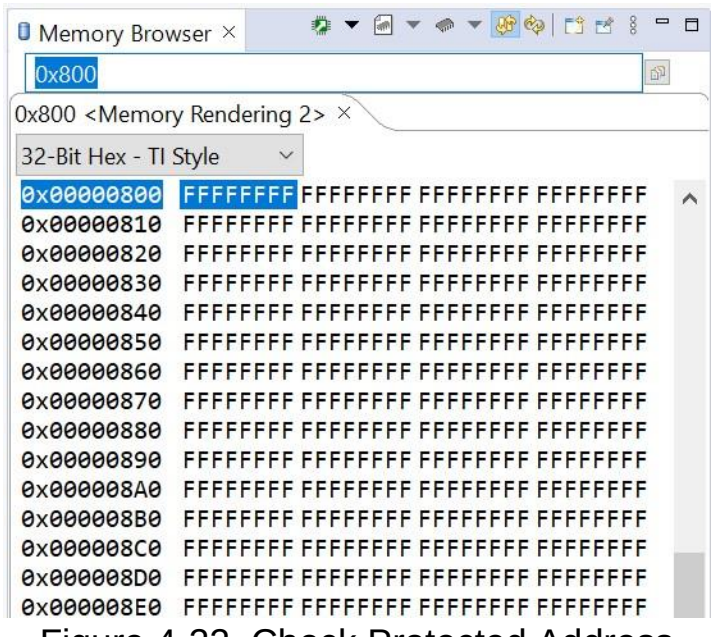

Figure 4-22. Check Protected Address

- When attempting to download a program to a protected FLASH region, the following prompt indicates a download failure:

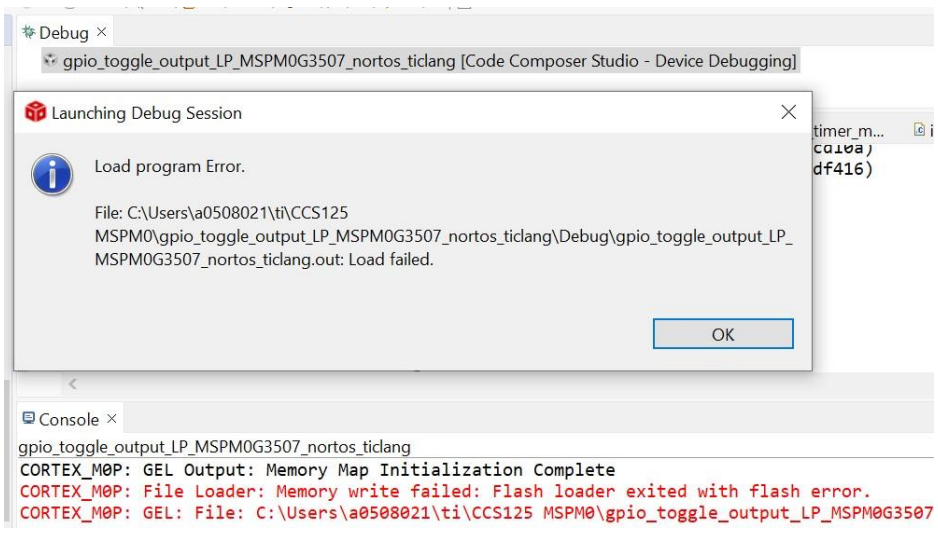

Figure 4-23. Write Protected Address Error Window

Note

Flash static write protection can be unlock by using Flash Reset or BSL Host.

4.3 BSL configuration

The MSPM0 BSL configuration mainly involves the software and hardware setup for the Bootstrap Loader. The specific implementation process and usage guidance for the BSL Host tool in the MSPM0 SDK are outlined below:

- Choose "Enable BSL" in BCR Configuration.

Figure 4-24. Enable BSL

- Configure eight sets of BSL Access passwords in BSL Configuration.
- Enable the Invoke pin and select the default pin or customize the required Pin Mux (if using software invoke, enable it).
- Configure the Communication Interface pin, using UART as an example.

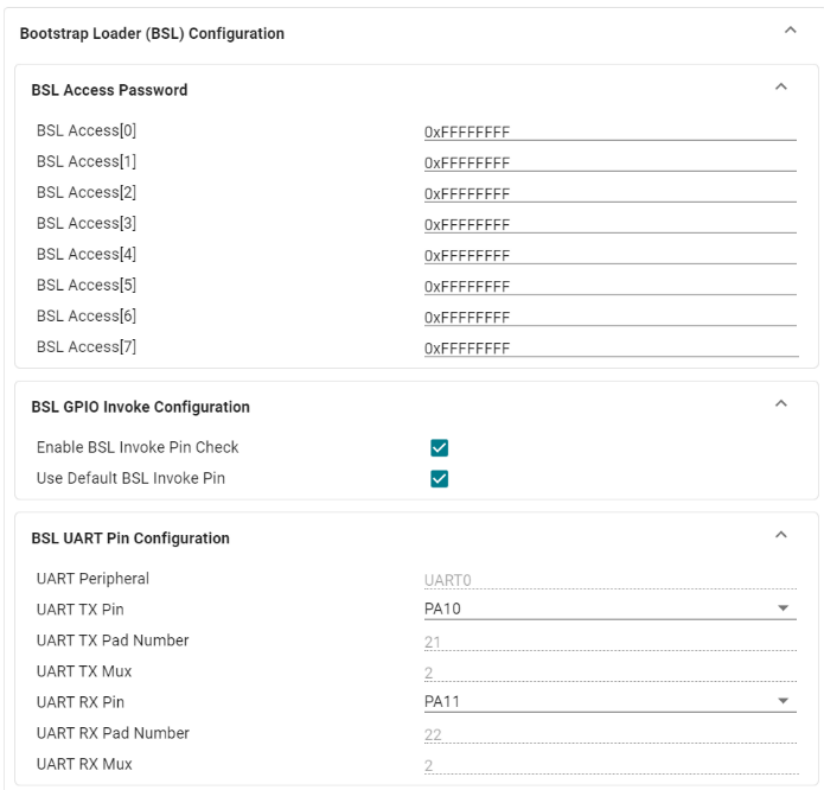

Figure 4-25. BSL Communication Interface Configuration

- After configuring, compile and download the program to the device.
- TI provides a BSL Host plugin in the MSPM0 SDK, and the following will explain the BSL Host communication implementation.
  - Connect the XDS110 and the chip via the UART interface pin configurated in the Sysconfig

- Open BSL host in the MSPM0 SDK

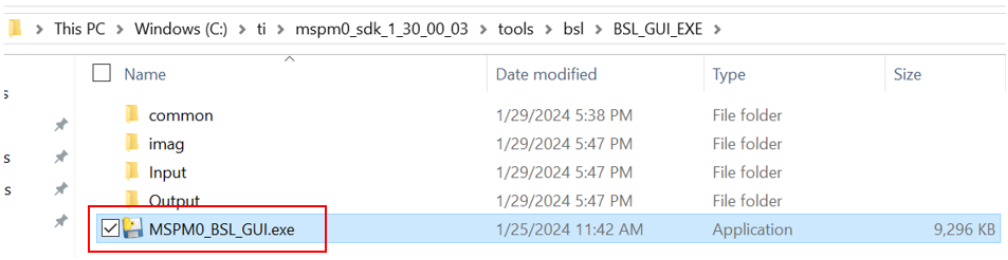

Figure 4-26. Open BSL Host GUI

- Following below steps to finish firmware update
  a) Choose the TI-TXT format image file that need to download(There are two demo images in the folder named input)
  b) Choose theTI-TXT format password file(There is a default file is in the input folder)
  c) Choose hardware bridge(If using standalone XDS110 debugger, choose “standalone XDS110”)
  d) Set the Invoke pin level and then pull down NRST pin, after that click Download button

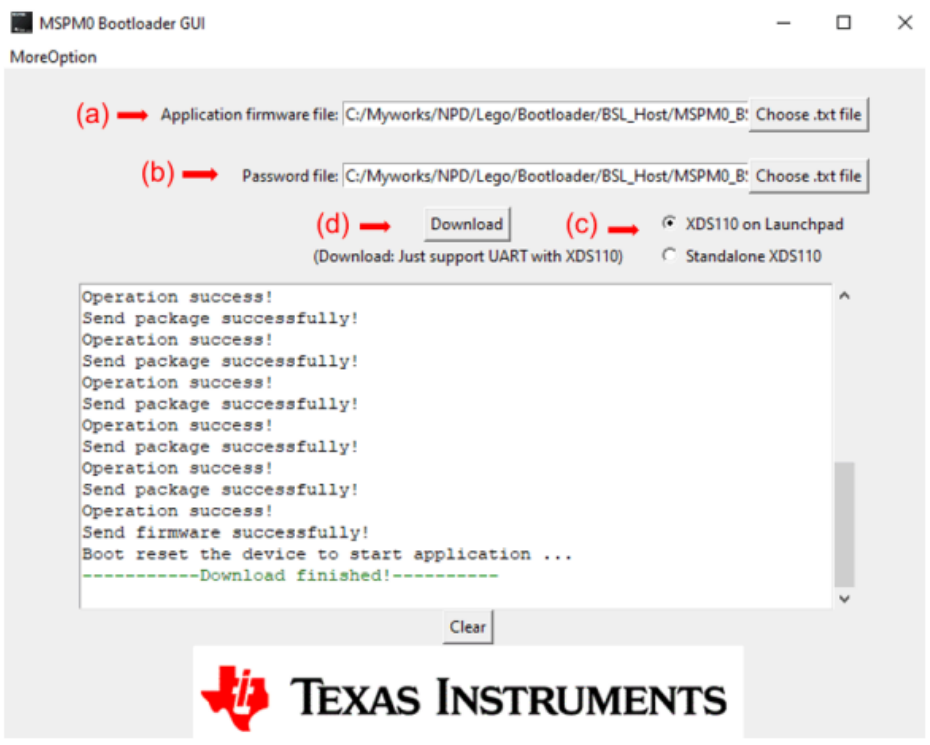

Figure 4-27. BSL GUI

Note

- J-Link cannot use the Scripts command function in CCS.
- SWD PSW need cut out the power and then take effect
- SWD unlock just one time , unless change NONMAIN configuration
- SWD PSW will not support to Mass Erase and Factory Erase
- Do not erase the NONMAIN region if won’t load new configuration

---

5 NONMAIN Operation on Keil

Keil support sysconfig same with CCS, so just refer to CCS for NONMAIN configuration, but Keil does not support scripts function. This chapter will focus on introducing the programming procedure for NONMAIN.

5.1 NONMAIN Operation Demo Project Implement

The steps to configure and program NONMAIN in Keil are as follows:

- First of all, double-click to open the .sysconfig file in the project. Go to Tools -> Sysconfig to open the Sysconfig configuration interface.

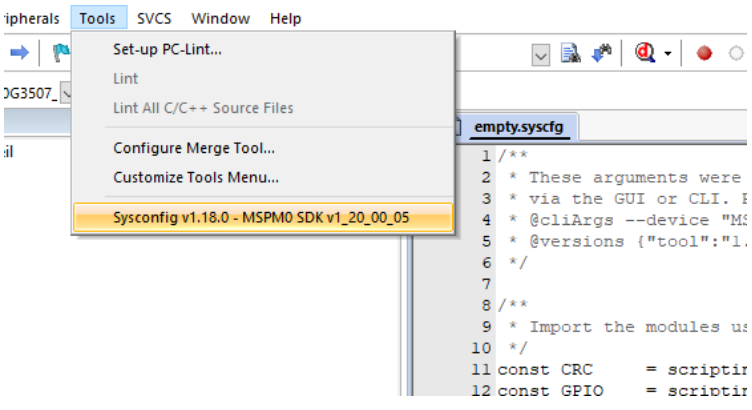

Figure 5-1. Open Sysconfig Tool

- Refer to Chapter 3 for the Sysconfig configuration process to complete the necessary configuration for NONMAIN. Save the manually generated boot_config.c and boot_config.h files and copy them into the project (Keil will not automatically import them into the project).

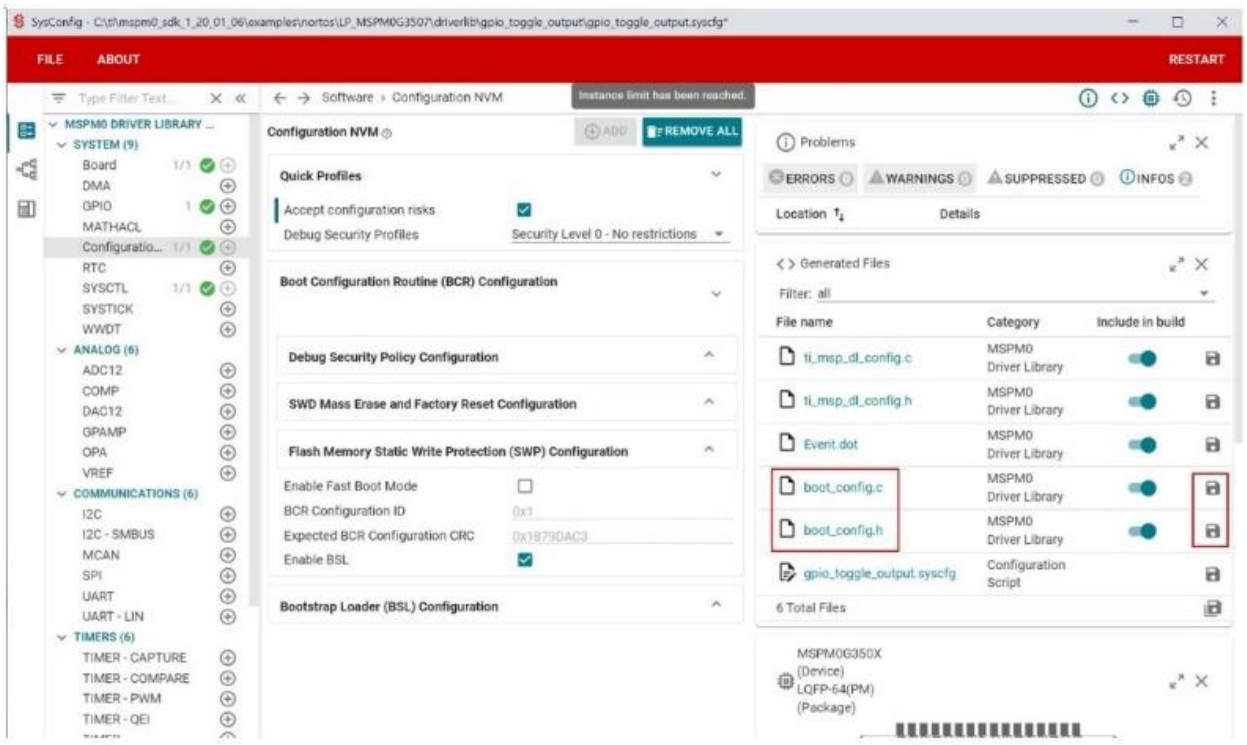

Figure 5-2. NONMAIN Generated File

- Select the correct debugger (XDS110 should choose CMSIS-DAP), and go to Setting.

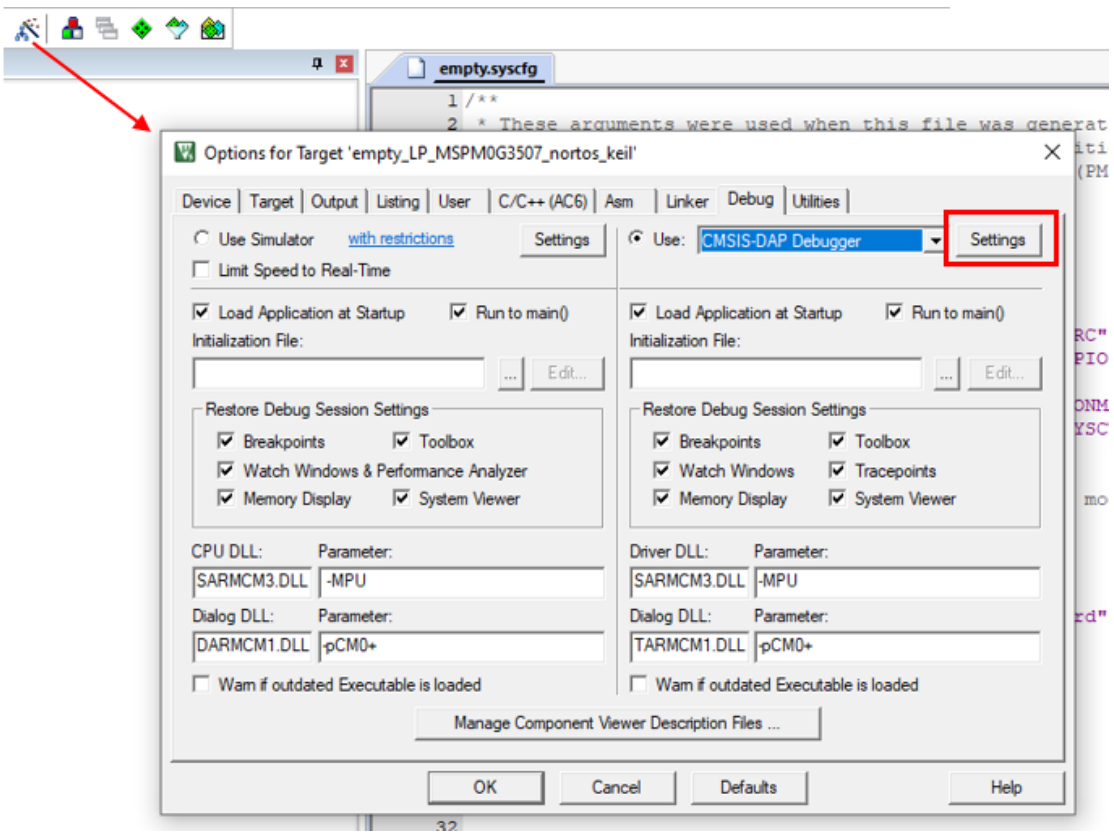

Figure 5-3. Debugger Selection

- Add NON-MAIN Programming Algorithm.

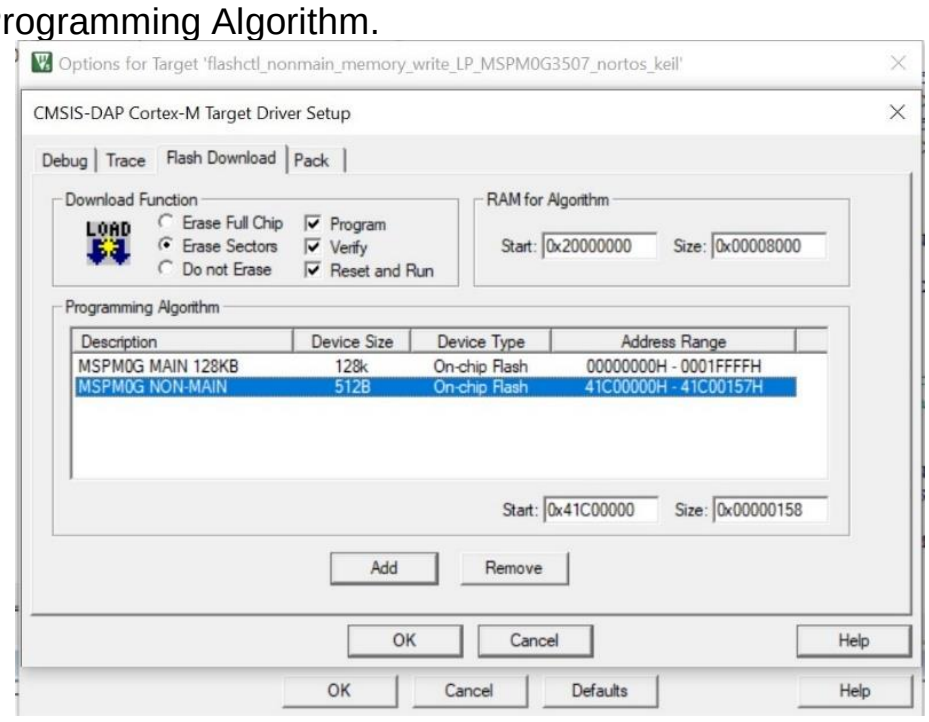

Figure 5-4. Add NONMAIN Algorithm

- After completing the relevant NONMAIN operations similar to CCS in the main program, compile the project and burn it into the device. NONMAIN will take effect after power cycling the device.

5.2 NONMAIN Operation Phenomenon

5.2.1 Resulting phenomenon of SWD Password

After configuring the SWD Password for NONMAIN, attempting to reprogram resulted in a failed programming attempt with the following error message:

Figure 5-5. SWD Password Block Window

5.2.2 Resulting phenomenon of FLASH Protect

After configuring the FLASH Protect for NONMAIN, attempting to reprogram the corresponding address resulted in a failed programming attempt with the following error message:

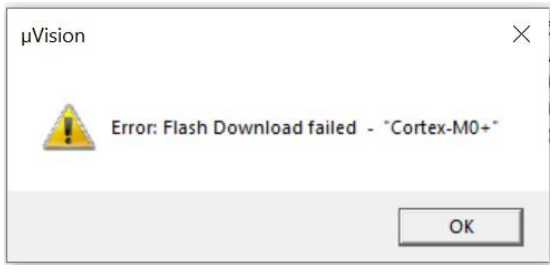

Figure 5-6. Flash Protected Block Window

As Keil does not yet support unlocking chips with Scripts, you can use the CCS Scripts or Factory Reset online tool mentioned in Chapter4.2.2, as well as the BSL Host mentioned in Chapter4.3.1 to unlock SWD locks or remove FLASH Protect.

5.2.2.1 J-Link Operation

First of all, refer to the configuration process for Sysconfig in Chapter 3 to complete the required configuration for NONMAIN.

SWD Encryption phenomenon:

There won't detect any device, and an error message will be prompted during the attempt to burn:

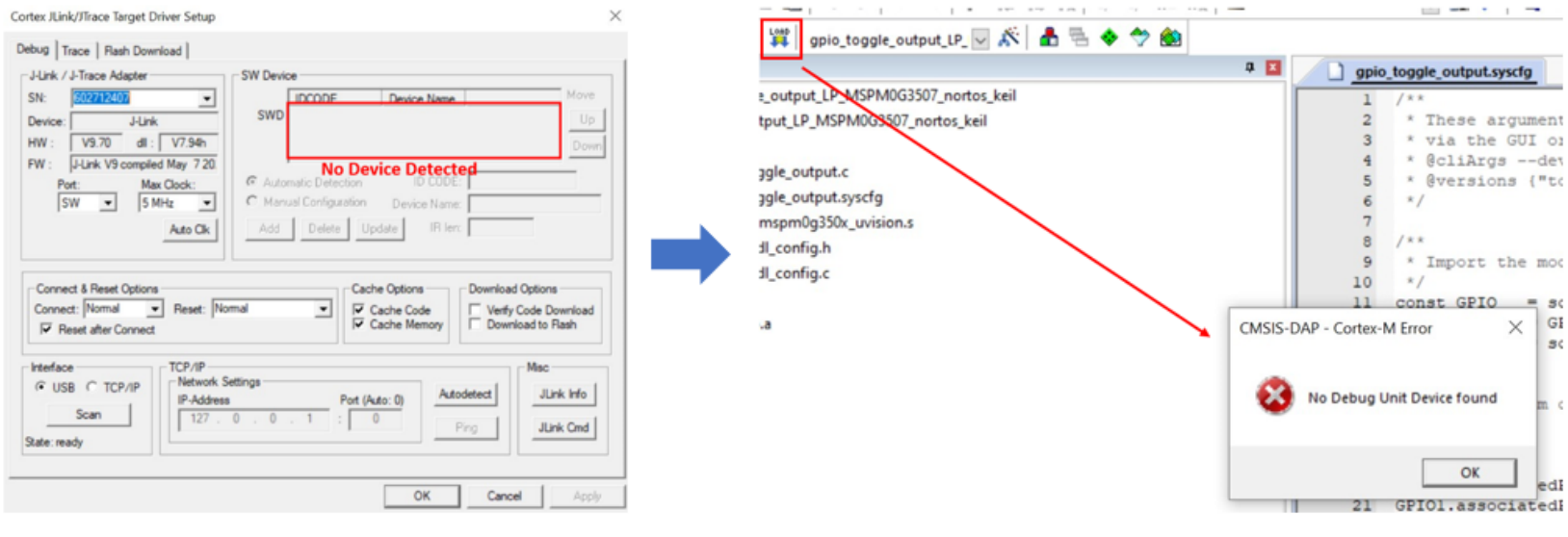

Figure 5-7. SWD Encryption Block Window

Similarly, you can utilize XDS110 with CCS Scripts or the Factory Reset online tool to remove SWD locks and FLASH Protect, and use BSL Host to unlock. In addition, TI will gradually support J-Link for using Scripts.

Note

- Sysconfig tool needs to be launched successfully when the .sysconfig file is open.
- J-Link requires the selection of SWD interface to communicate when used in Keil.
- Keil does not support Scripts for performing Mass Erase, Factory Reset, and other operations on a device.

For more details for how to start Keil development with MSPM0, you can refer to Keil IDE Guide.

6 NONMAIN Operation on IAR

IAR support sysconfig same with CCS, so just refer to CCS for NONMAIN configuration, but IAR does not support scripts function. This chapter will focus on introducing the programming procedure for NONMAIN.

6.1 NONMAIN Operation Demo Project Implement

The process of completing NONMAIN operations in IAR is as follows:

- Double-clicking the .sysconfig file in the project opens the Sysconfig tool.
- After completing the required configurations, save and close the Sysconfig interface. This will automatically generate Boot_config.c and Boot_config.h in the project.
- Select the correct debugger.

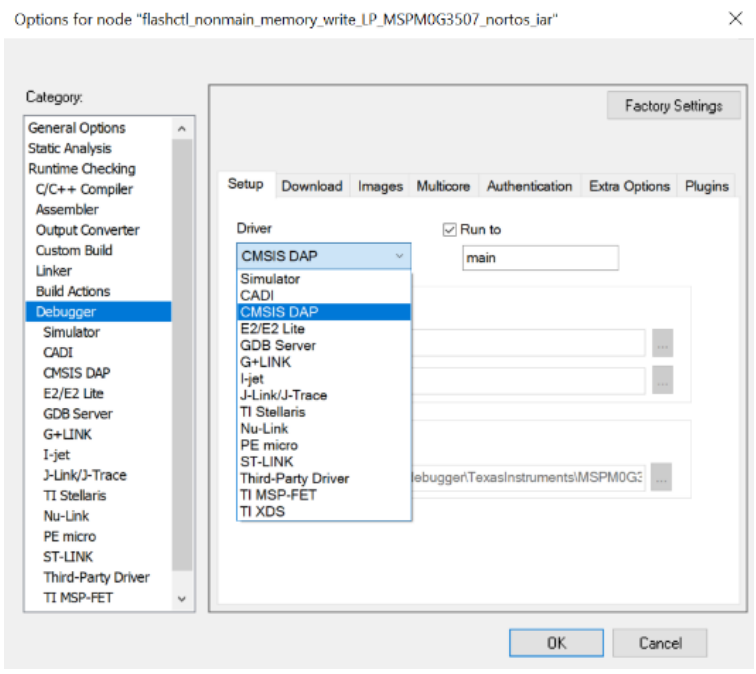

Figure 6-1. Debugger Selection

- In the Download options, add and edit the settings related to NONMAIN erasure as follows.

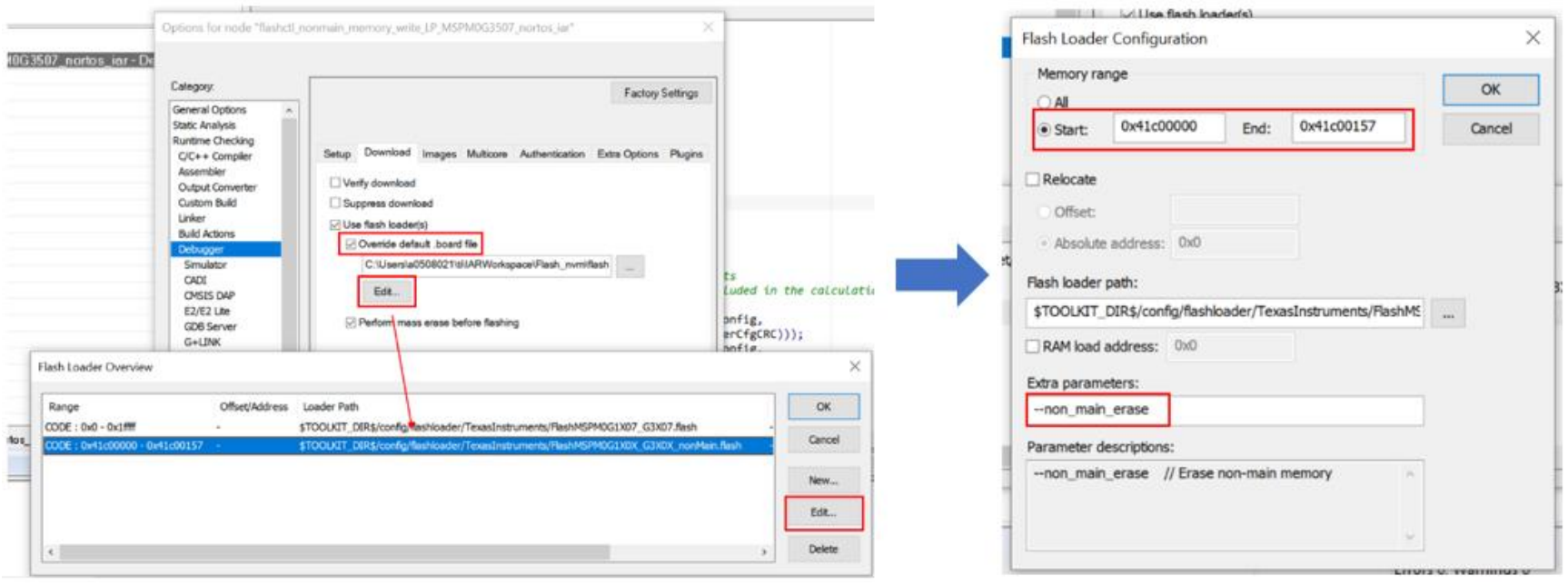

Figure 6-2. Add NONMAIN Related Settings

- After completing the same NONMAIN-related operations as in CCS in the main program, compile the project and burn it to the device. After power cycle, NONMAIN will take effect.

6.2 NONMAIN Operation Phenomenon

6.2.1 Resulting phenomenon of SWD Password

After configuring the SWD Password for NONMAIN, when attempting to re-burn, the burning failed and the following error appeared.

Figure 6-3. SWD Encryption Block Window

6.2.2 Resulting phenomenon of FLASH Protect

After configuring FLASH Protect for NONMAIN, trying to re-burn the program to the corresponding address failed, with the same error as with SWD.

Figure 6-4. Flash Protected Block Window

Similar to the Keil IDE, you can use CCS Scripts mentioned in the previous Chapter4.2.2 or the Factory Reset online tool to unlock SWD and FLASH Protect, or use the BSL Host Implement mentioned in Chapter4.3.1 to unlock it.

Note

- Same with Keil, there is no Scripts functionality to perform Mass Erase and Factory Reset operations on the Device.

For more details for how to start Keil development with MSPM0, you can refer to IAR IDE Guide.

7 NONMAIN operation by programmer tool

This chapter mainly demonstrates the programming of MSPM0-related burn-in tools Uniflash and J-Flash in NONMAIN.

7.1 Uniflash

For the operation of MSPM0 NONMAIN, TI's accompanying programming tool Uniflash comes with a standalone Command Line Interface (CLI), which can achieve memory file export, program burning, and custom command parameters for chip erasure.

The process of using Uniflash to program the NONMAIN operation of the device is as follows:

- Modify the Erase Range setting to ensure that NONMAIN has been erased before writing to it.

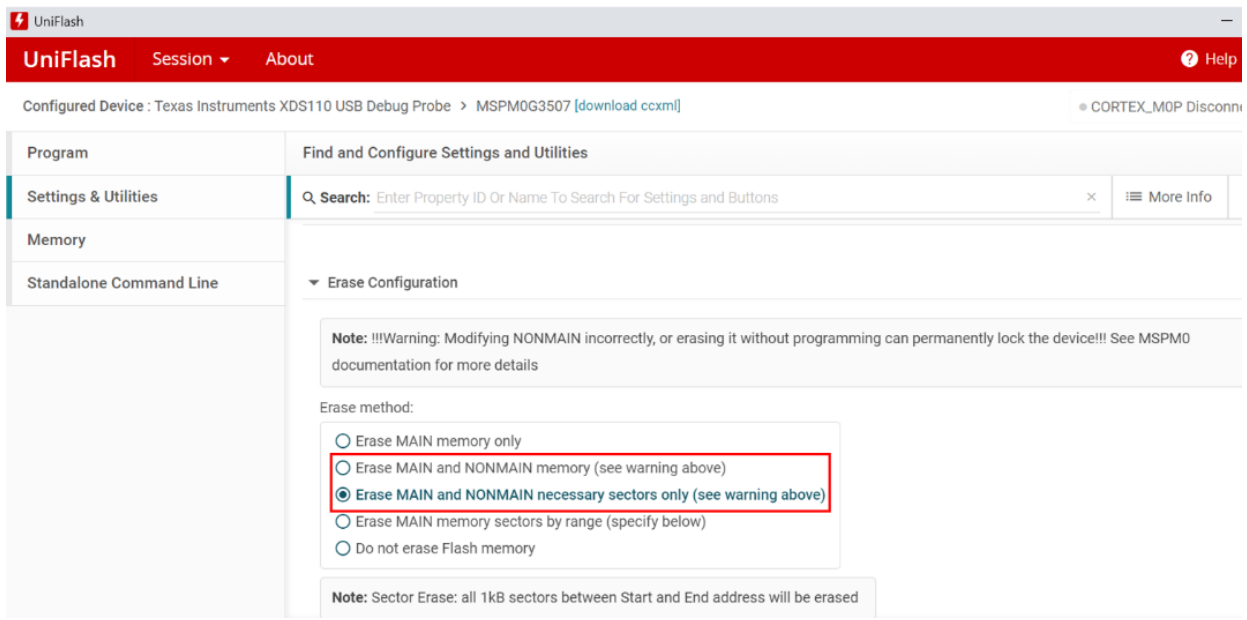

Figure 7-1. Erase Range Setting

- Import the firmware to be programmed and complete the firmware burning using Load Image.

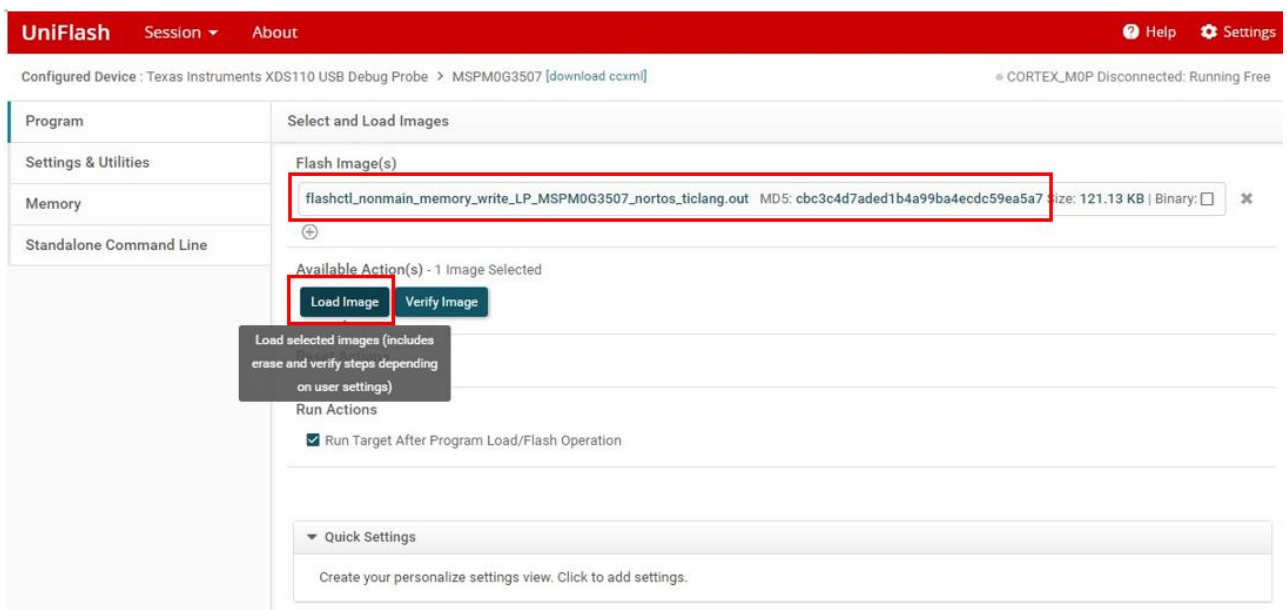

Figure 7-2. Load Image

SWD lock or NONMAIN destoried wrongly phenomenon:

When try Load Image or Read Memory for Device you will see below error

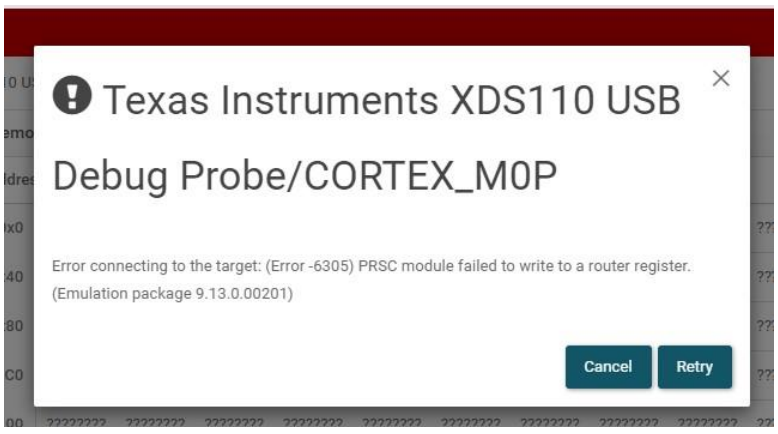

Figure 7-3. SWD Encryption Block Window

FLASH protect phenomenon:

When try programming device you will see below errors

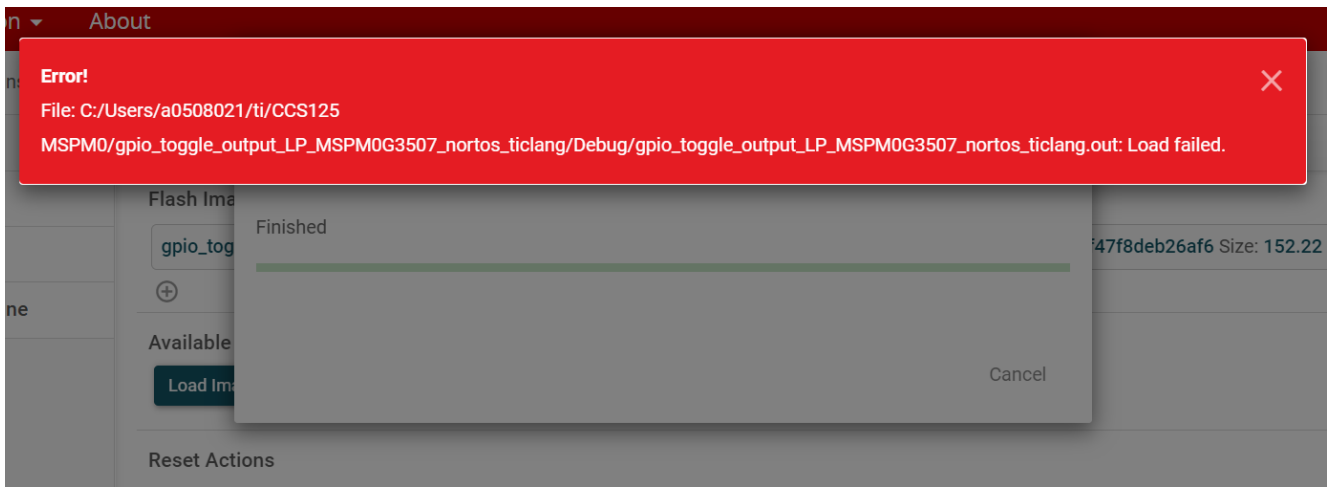

Figure 7-4. Flash Protected Block Window

7.2 J-Flash

J-Flash is a programming tool compatible with the J-Link debugger. The following is the operation procedure for burning the NONMAIN program using J-Flash with the J-Link debugger:

- After importing the Hex firmware file into J-Flash, modify the project settings for the SWD interface and Erase Range.

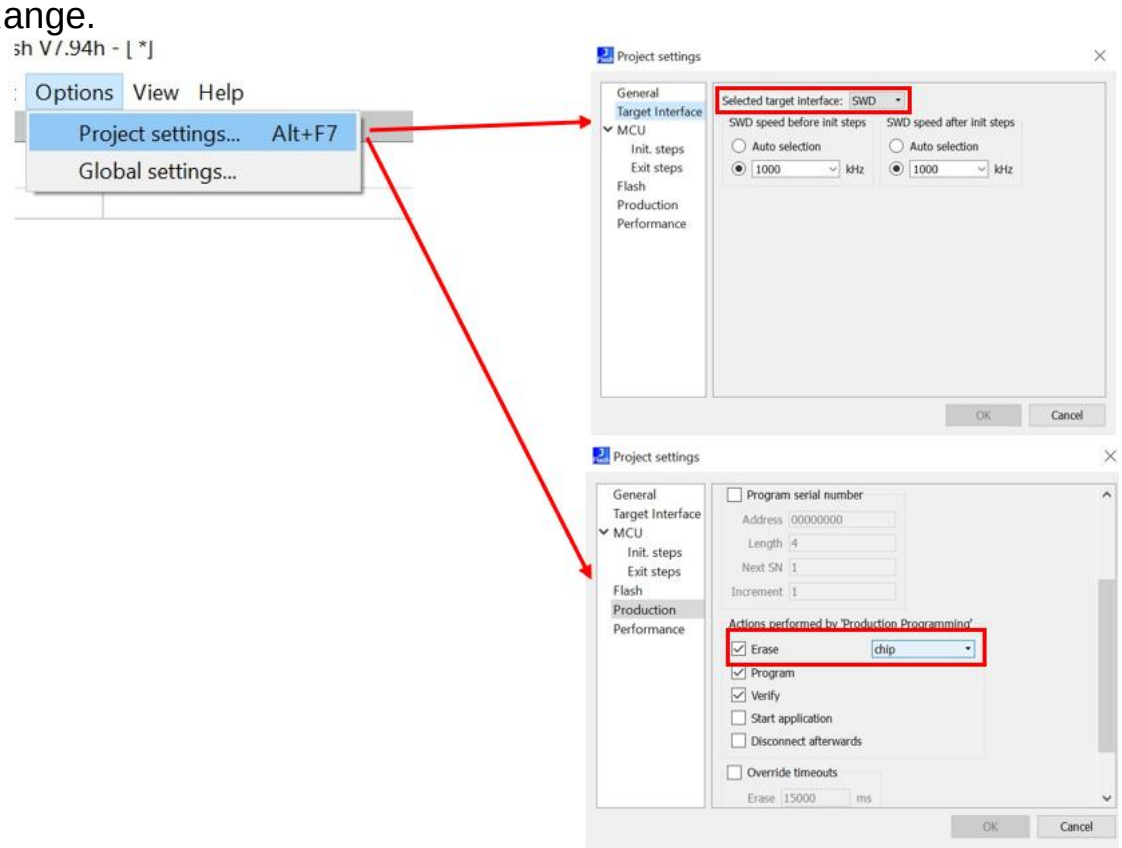

Figure 7-5. Project Setting

- Test the connection status of the Device to ensure it is working properly.

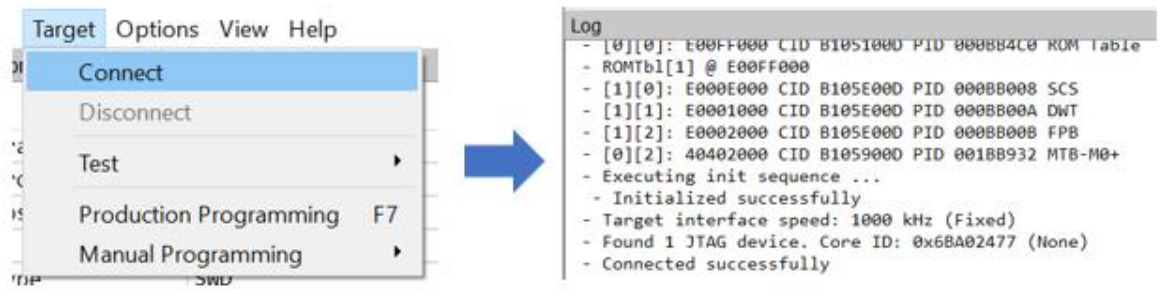

Figure 7-6. Connect to Target

- Download the firmware to the Device, choosing between Production (Auto) or Manual Programming.

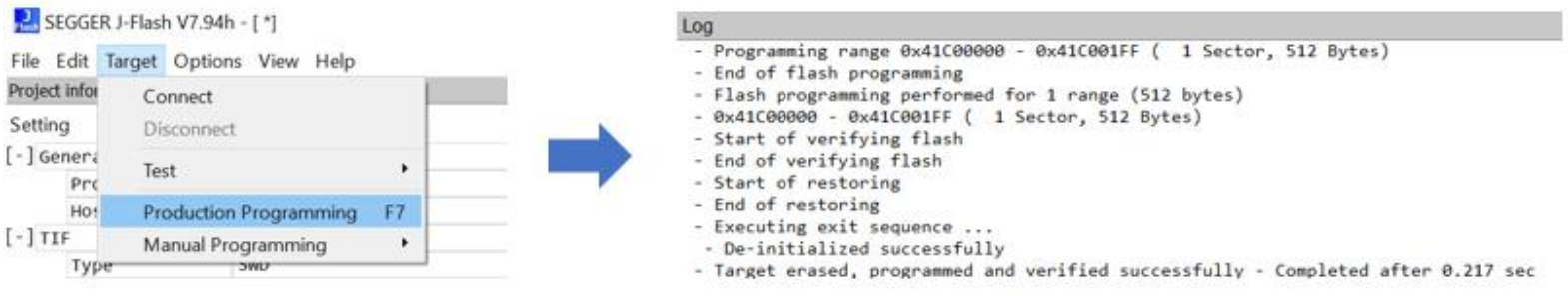

Figure 7-7. Programming

SWD lock or NONMAIN destoried wrongly phenomenon:

When try Connect or Programming you can will below errors

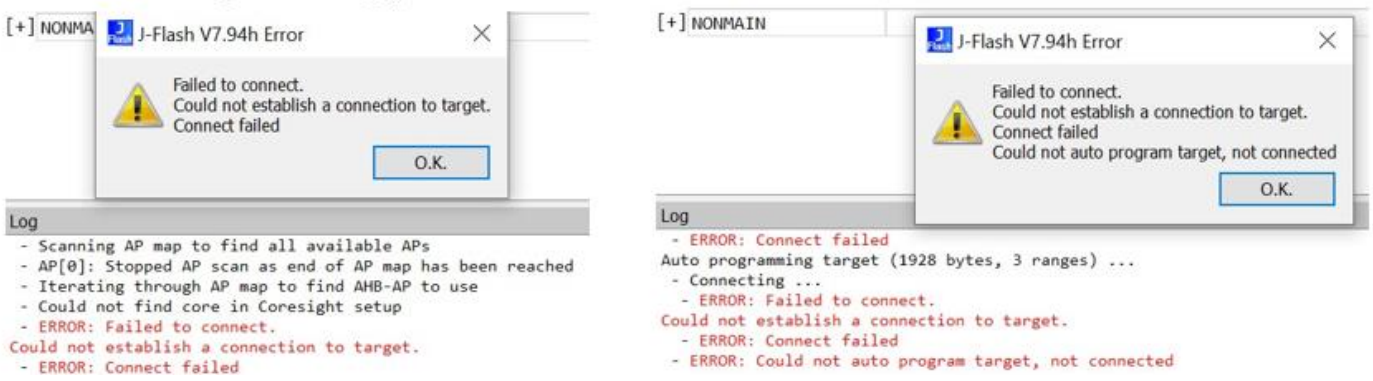

Figure 7-8. SWD Encryption Block Window

FLASH protect phenomenon:

When try Programming you can will below errors

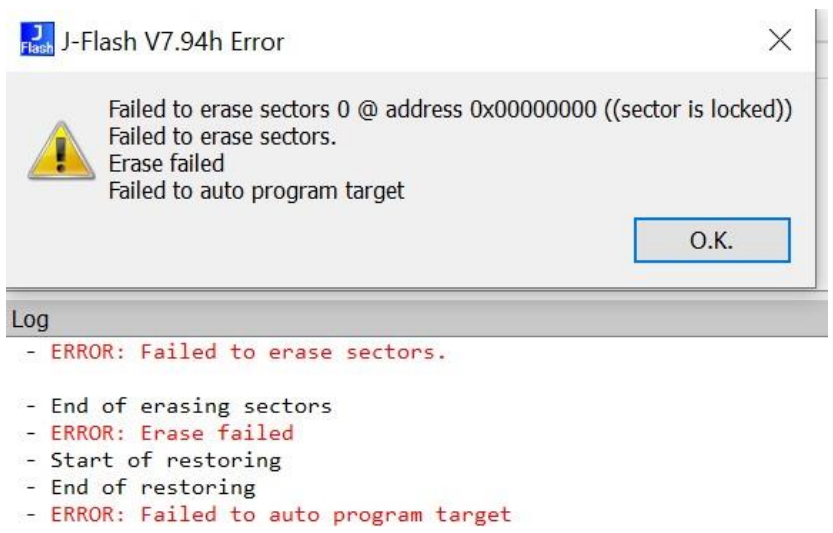

Figure 7-9. Flash Protected Block Window

Note

- During development, it is recommended to burn NONMAIN and MAIN code separately. The APP code should be burned first, followed by the NONMAIN configuration code.
- It is advisable to keep the BSL default Invoke pin externally pulled down to prevent the chip from incorrectly entering BSL when powered up.

---

8 Common Questions

8.1 Dynamically modifying NONMAIN configuration

During the development process, it may be necessary to dynamically modify NONMAIN parameters in the APP program. Please refer to the “Execute the NONMAIN programming steps in the main program” in the chapter 4.1.

The mainly operation involve erasing the NONMAIN area first, then updating the NONMAIN configuration by writing it, Finally, you need to reset the device to execute the new NONMAIN configuration.

8.2 Create a separate NONMAIN configuration project

- Firstly, refer to the above content to complete the NONMAIN configuration in Sysconfig.
- Next, modify the Flash programming region to include NONMAIN.

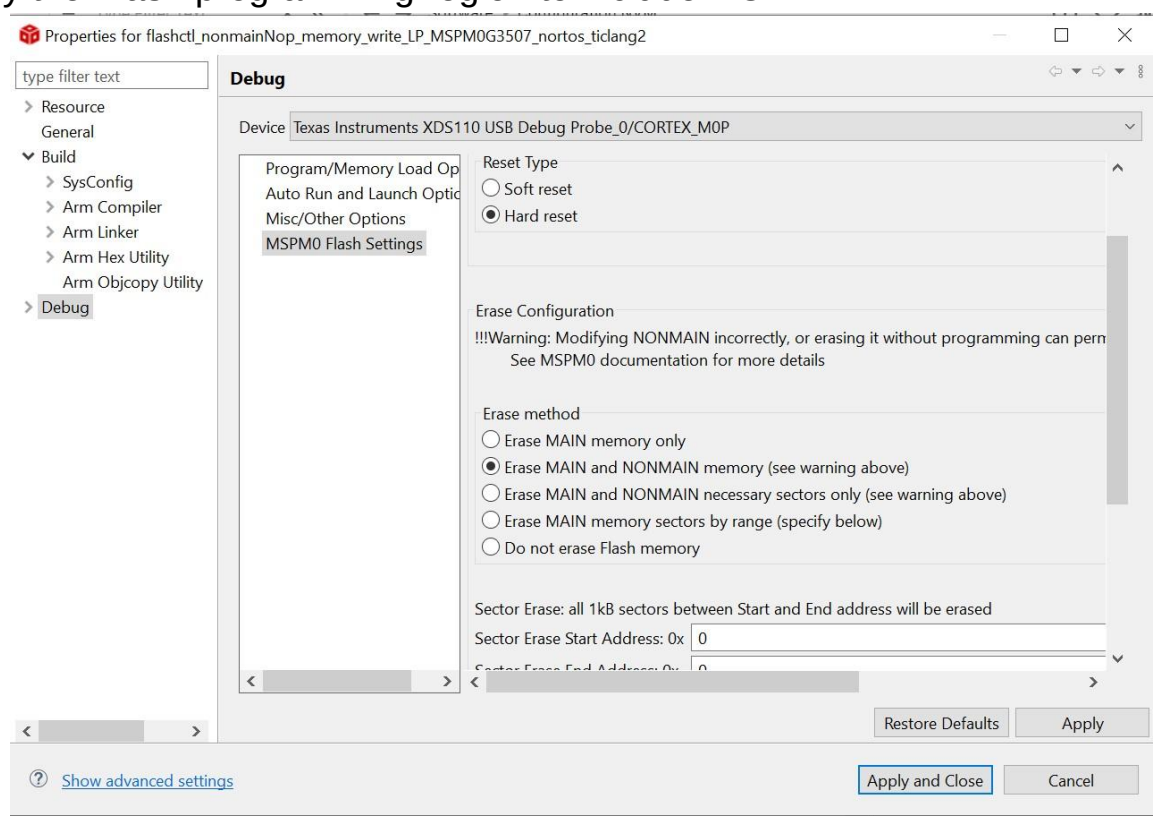

Figure 8-1. Erase Range Setting

- Save and compile the project, then burn it to the device.

9 Reference

1. Texas Instruments: MSPM0 G-Series 80-MHz Microcontrollers Technical Reference Manual
2. Texas Instruments: MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual
3. Texas Instruments: MSPM0 C-Series 24-MHz Microcontrollers Technical Reference Manual
4. Texas Instruments: MSPM0 Bootloader User's Guide
5. Texas Instruments: MSPM0 Bootloader (BSL) Implementation

10 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

DATA	REVERSION	NOTES
Apr 2024	-	Initial Release