MSPM0G3519 Flash Bank Swap Guide

MSP HSM Apps

Bank Swap Function Description

- G3519 will include two code bank:

Table 6-11. Bank Address Swap Translation

Bank and Region	Address Space Before Swap	Address Space After Swap
BANK0 MAIN	0x0000.0000 – 0x0003.FFFF	0x0004.0000 – 0x0007.FFFF
BANK1 MAIN	0x0004.0000 – 0x0007.FFFF	0x0000.0000 – 0x0003.FFFF

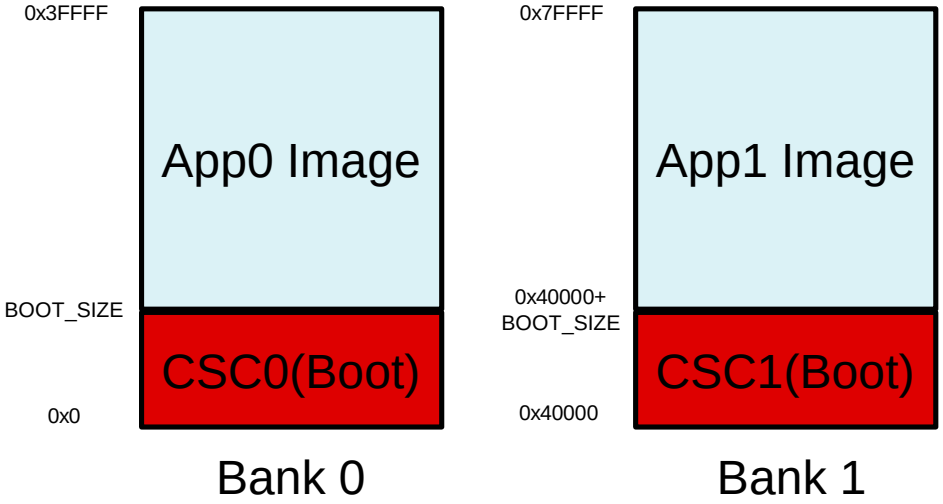
- MCU will run in BANK0 default after POR/BOR
- Bank swap function need to enable by change NONMAIN region
  - Enable CSC function
  - INITDONE will not auto release
- Swap bank from bank0 to bank1
  - Set swap bit:
    DL_SYSCTL_executeFromUpperFlashBank(); // set swap bank to bank1
  - Issue INITDONE(This will trigger System Reset, and enable bank swap):
    DL_SYSCTL_issueINITDONE(); // Issue INITDOEN to trigger System Reset -> swap to bank1
- Swap bank from bank1 to bank0
  - Trigger BOOT Reset:
    DL_SYSCTL_resetDevice(DL_SYSCTL_RESET_BOOT); // trigger BOOT Reset -> swap to bank0

Bank Swap Function Description

- G3519 will include two code bank:

Table 6-11. Bank Address Swap Translation

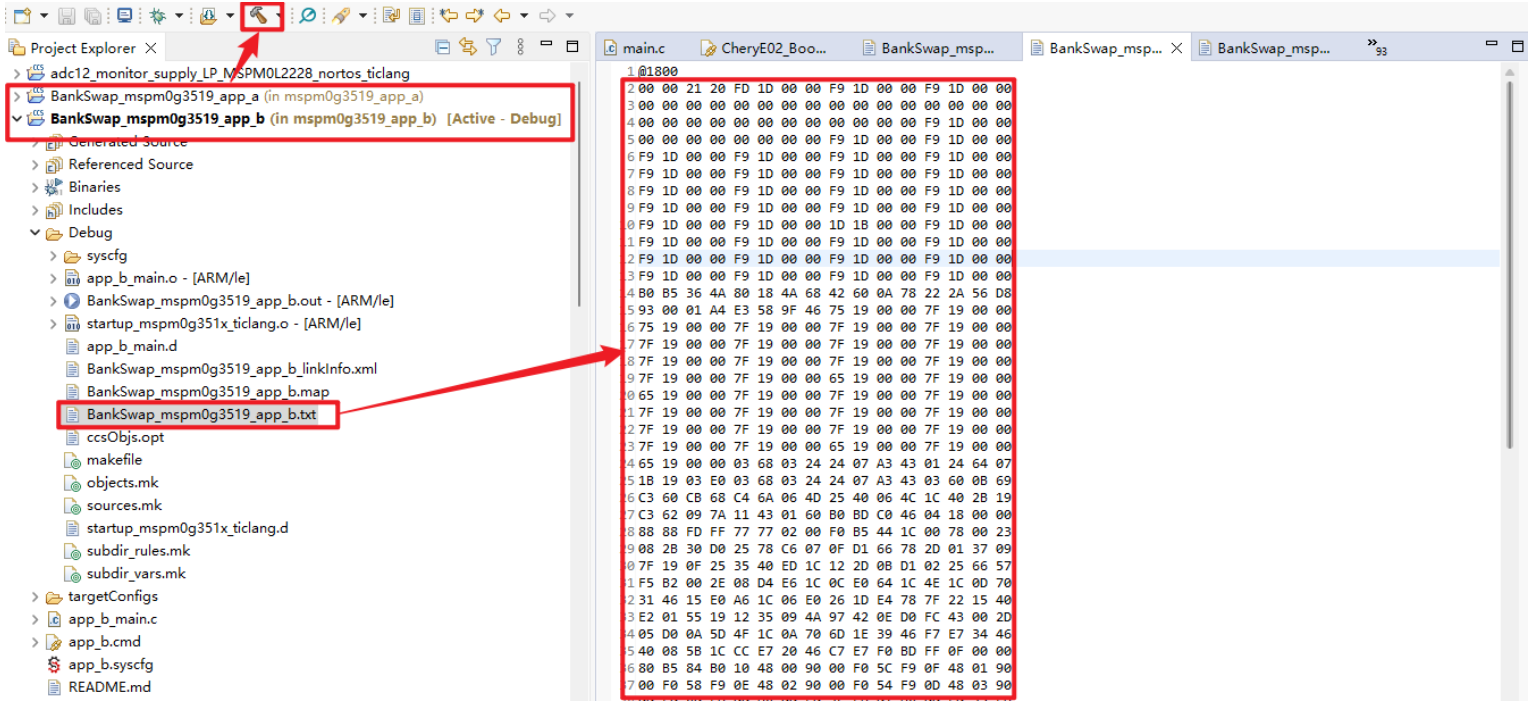
Bank and Region	Address Space Before Swap	Address Space After Swap
BANK0 MAIN	0x0000.0000 – 0x0003.FFFF	0x0004.0000 – 0x0007.FFFF
BANK1 MAIN	0x0004.0000 – 0x0007.FFFF	0x0000.0000 – 0x0003.FFFF



- In a bank swap situation, there are four parts of code in Flash MAIN region:
  - App0 image in bank0
  - App1 image in bank1
  - CSC0 code in bank0 root address, which determine bank swap policy and release INITDONE.
  - CSC1 code in bank1 root address, which has the same code with CSC0 and will directly jump to App1 code.

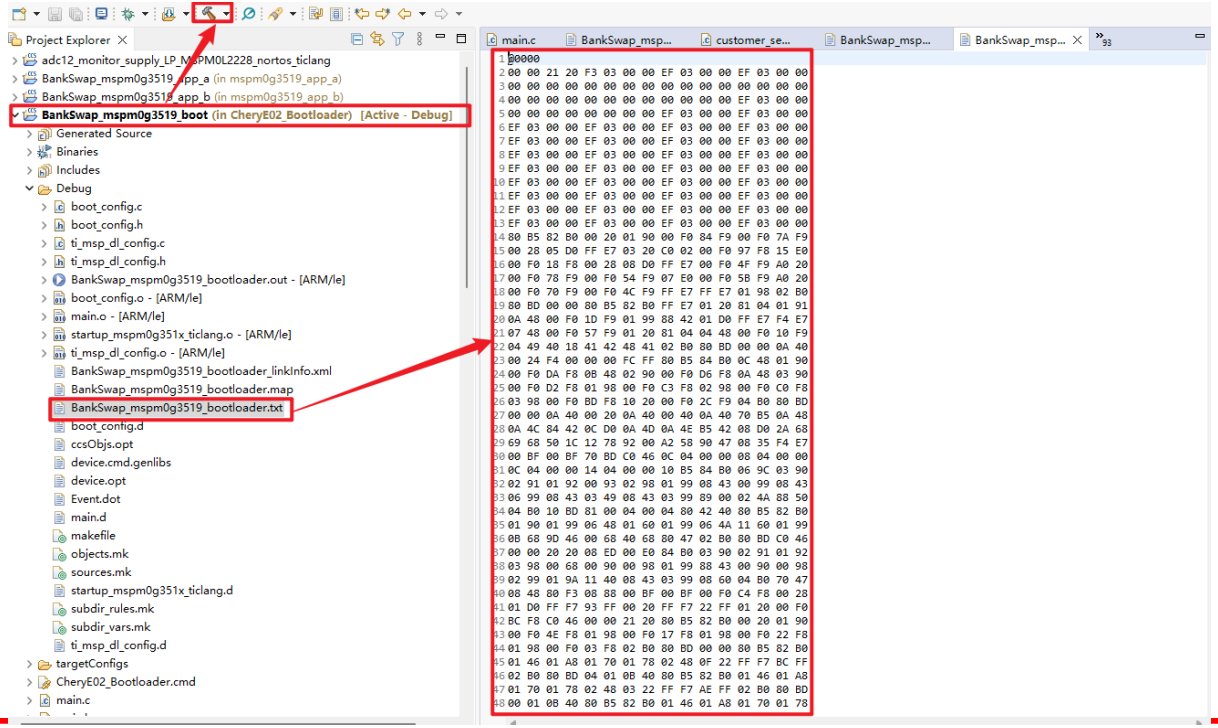
Hands on

Step 1: Built the application example of bank0 & bank1, and generate .txt output file.



Hands on

Step 2: Built the application example of boot, the generated .txt data will be place at both 0x0000 address and 0x40000 address.

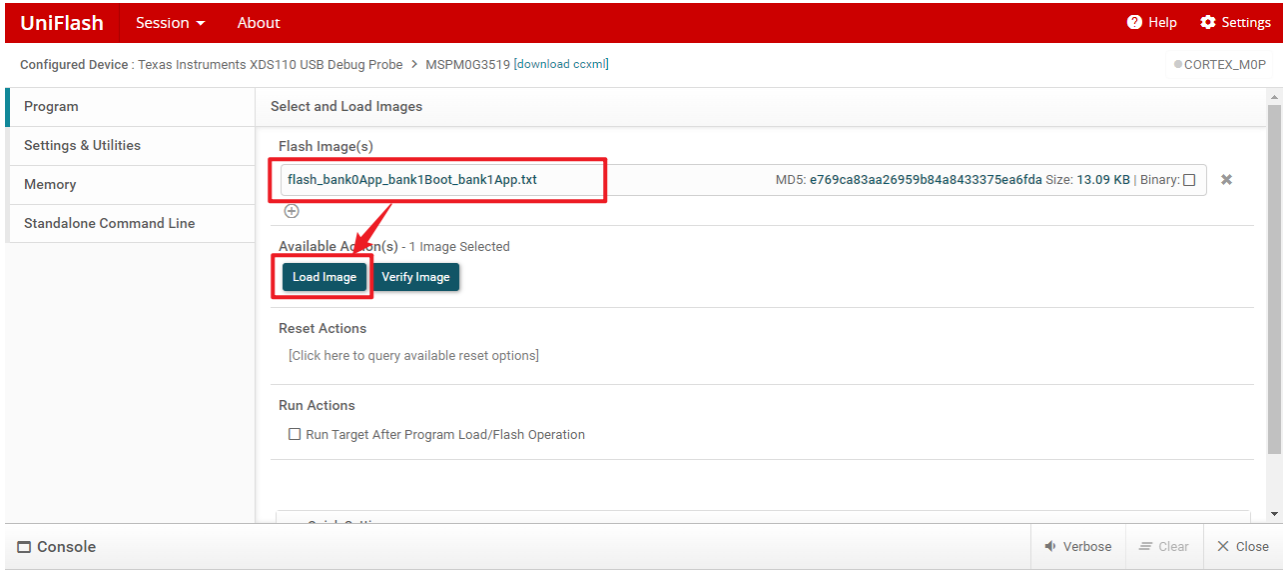


Hands on

Step 3: Generate the image file of App0 + App1 + CSC1. (flash_bank0App_bank1Boot_bank1App.txt)

Hands on

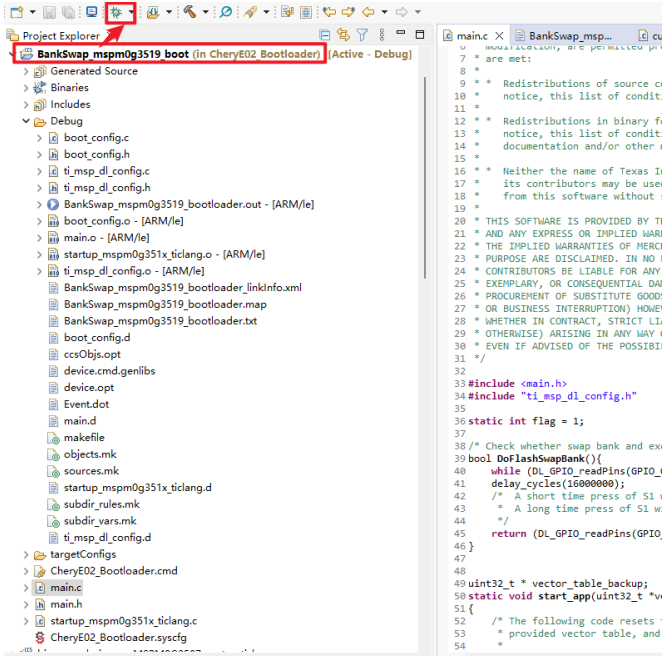
Step 4: Download the image by Uniflash . (May need repower MCU after downloading)

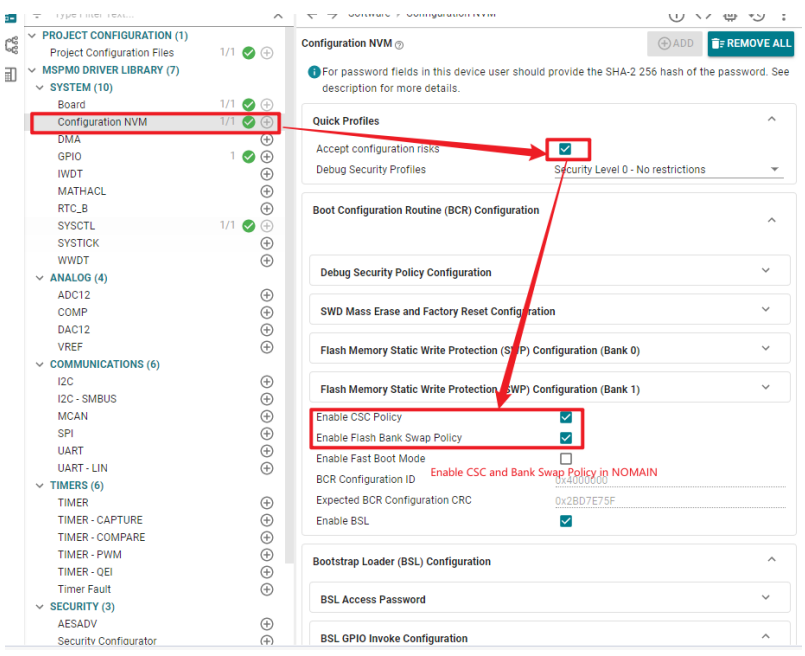


Hands on

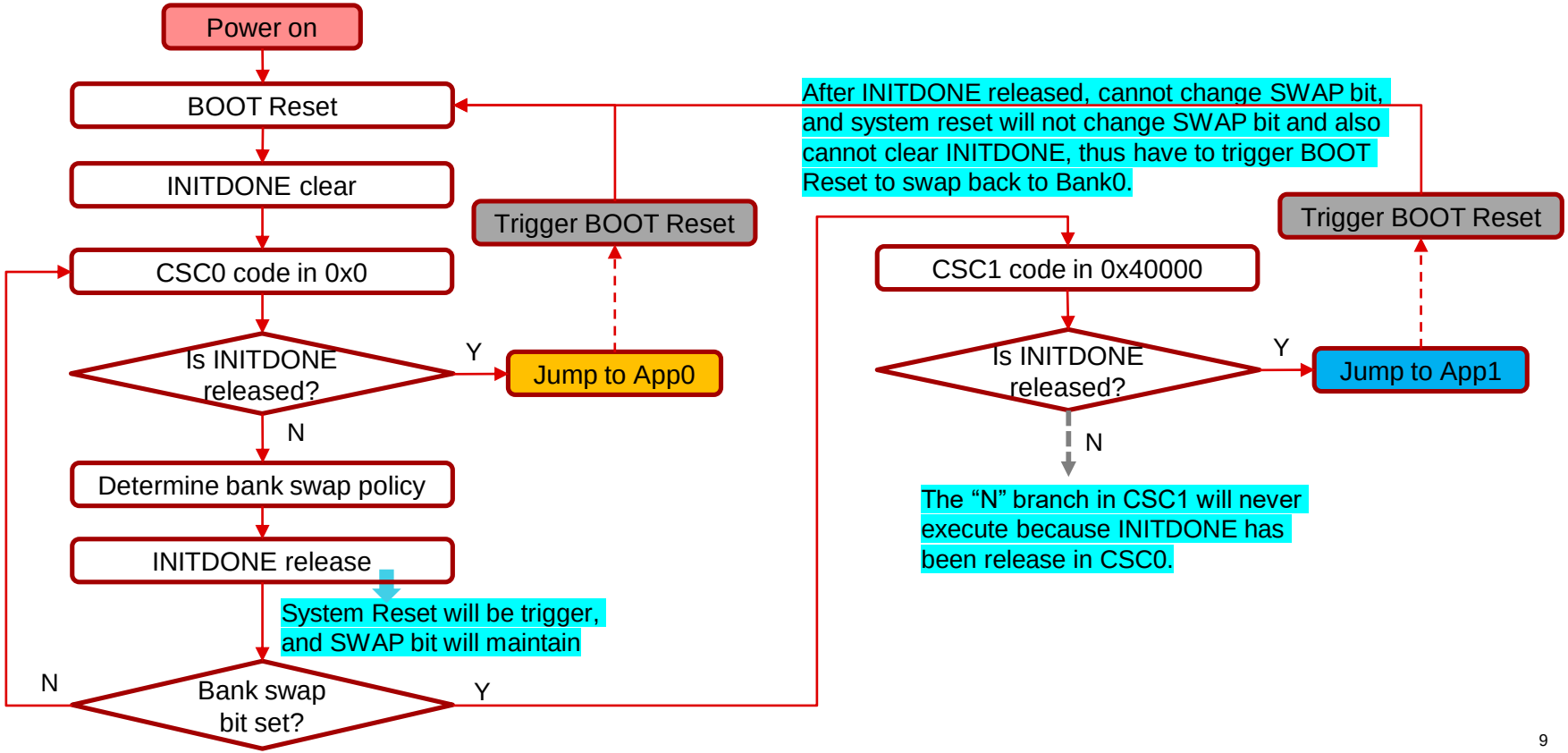
Step 5: Download and debug CSC0, which including the change of NONMAIN.

- Press S1 less than 0.5s -> run in bank0, blue led fast flashing
- Press S1 more than 0.5s -> run in bank1, green led slow flashing



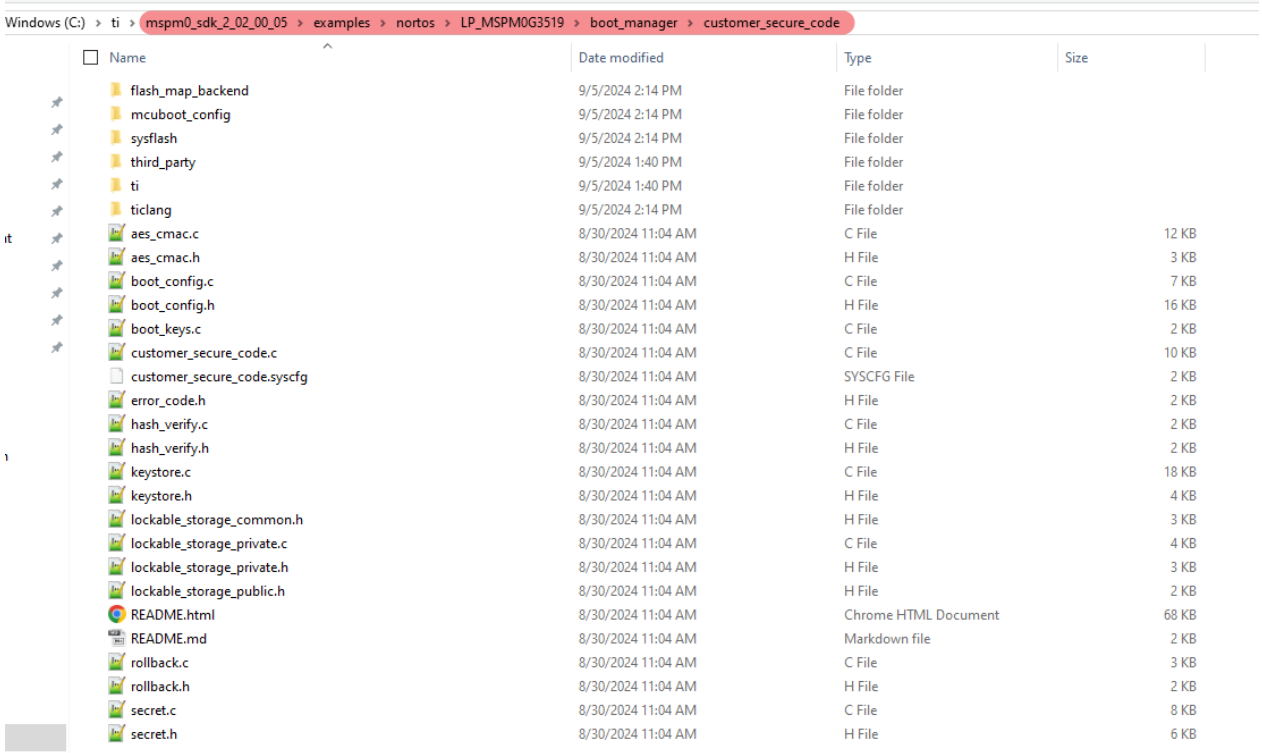


Swap Flow



More Resource

Secure Boot Example, will include Bank swap feature



Name	Date modified	Type	Size
flash_map_backend	9/5/2024 2:14 PM	File folder	
mcuboot_config	9/5/2024 2:14 PM	File folder	
sysflash	9/5/2024 2:14 PM	File folder	
third_party	9/5/2024 1:40 PM	File folder	
ti	9/5/2024 1:40 PM	File folder	
ticlang	9/5/2024 2:14 PM	File folder	
aes_cmac.c	8/30/2024 11:04 AM	C File	12 KB
aes_cmac.h	8/30/2024 11:04 AM	H File	3 KB
boot_config.c	8/30/2024 11:04 AM	C File	7 KB
boot_config.h	8/30/2024 11:04 AM	H File	16 KB
boot_keys.c	8/30/2024 11:04 AM	C File	2 KB
customer_secure_code.c	8/30/2024 11:04 AM	C File	10 KB
customer_secure_code.syscfg	8/30/2024 11:04 AM	SYSCFG File	2 KB
error_code.h	8/30/2024 11:04 AM	H File	2 KB
hash_verify.c	8/30/2024 11:04 AM	C File	2 KB
hash_verify.h	8/30/2024 11:04 AM	H File	2 KB
keystore.c	8/30/2024 11:04 AM	C File	18 KB
keystore.h	8/30/2024 11:04 AM	H File	4 KB
lockable_storage_common.h	8/30/2024 11:04 AM	H File	3 KB
lockable_storage_private.c	8/30/2024 11:04 AM	C File	4 KB
lockable_storage_private.h	8/30/2024 11:04 AM	H File	3 KB
lockable_storage_public.h	8/30/2024 11:04 AM	H File	2 KB
README.html	8/30/2024 11:04 AM	Chrome HTML Document	68 KB
README.md	8/30/2024 11:04 AM	Markdown file	2 KB
rollback.c	8/30/2024 11:04 AM	C File	3 KB
rollback.h	8/30/2024 11:04 AM	H File	2 KB
secret.c	8/30/2024 11:04 AM	C File	8 KB
secret.h	8/30/2024 11:04 AM	H File	6 KB