FAQ0048
Frequently Asked Questions

Flash sector erase issue

Questions: How to perform Flash sector erase when they have different sector size?

Answer:
For 256K or above Flash memory, each sector is 2K, while for less than 256K, each sector size is 1K. When it comes to sector erase operation, they show some differences:

1. Sector erase procedure: unlock Flash memory-→ erase Flash sector -→ lock Flash memory
2. Erasing the start address of Flash memory (sector_address) or any address within this sector will erase the entire sector
3. For Flash memory less than 512K, sector erase is done through FLASH->CTRL/FLASH->ADDR;
   For 512K or above Flash memory, sector erase is done through FLASH->CTRL2/FLASH->ADDR2.
4. When migrating from the erase code of 1K-sector Flash to 2K-sector Flash, if the users need to erase several sectors continuously, the incremental length of the start address should be changed to 2K (0x800) accordingly. If it is still 1K (0x400), the 2K Flash sector will have to be erased twice.

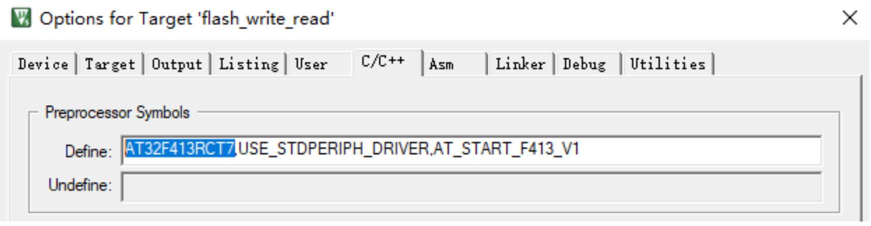
   For example, when upgrading APP through IAP, its general operation is to erase one sector and then write one sector. If the 2K sector is erased in the way of 1K sector erase operation, for instance, to erase a Flash sector with start address 0x08001000, this sector is erased in the first erase operation,and 1K data is written. In the second erase operation, the erase length is increased by ony 1K, the start address of write operation is 0x08001400, then this sector will still be erased, erasing the previously written 1K data.
5. When migrating from the erase code of 2K-sector Flash to 1K-sector Flash, if the users need to erase several sectors continuously, the incremental length of the start address should be changed to 1K (0x400) accordingly. If it is still 2K (0x800), then every other sector will be missed.

   For example, when upgrading APP through IAP, its general operation is to erase one sector and then write one sector. If 1K-sector is erased in the way of 2K sector erase operation, for instance, to erase the sector with start address 0x08001000, the 1K sector is erased and 2K data is written. In this case, the sector where the start address is 0x08001400 is written with data without being erased, causing that the written data is incorrect.

Because of the above information, for AT32F413/AT32F415 with Flash memory less than 256K, attention should be paid to the fact that BSP program is desgined based on the maximum size of 256K, direct compiling and download may trigger issues such as download failure or re-download failure due to bank erase failure.

Solutions:

1. Refer to *AT32F4xx firmware BSP&Pack user guide* stored in BSP document, change AT32 MCU part number and Flash algorithm document.
2. *Options→C/C++-→Preprocessor Symbols→ Define,* change AT32 MCU part number



Options for Target 'flash_write_read'

Preprocessor Symbols

Define: AT32F413RCT7,USE_STDPERIPH_DRIVER,AT_START_F413_V1

Undefine:

3. If you are using Flash read/write example code, the SECTOR_SIZE in the flash.c/flash.h file should be changed.

```
#define SECTOR_SIZE 2048
```

Erase example code:

```
flash_unlock();
flash_sector_erase(sector_address);
flash_lock();
```

For Flash memory within 512 K size, its Flash sector erase code in the flash_sector_erase(); function is described as follows:

```
#define FLASH_BANK1_START_ADDR        ((uint32_t)0x08000000) /*!< flash start address of bank1 */
#define FLASH_BANK1_END_ADDR          ((uint32_t)0x0807FFFF) /*!< flash end address of bank1 */
if((sector_address >= FLASH_BANK1_START_ADDR) && (sector_address <=
FLASH_BANK1_END_ADDR))
{
  /* wait for last operation to be completed */
  status = flash_bank1_operation_wait_for(ERASE_TIMEOUT);

  if(status == FLASH_OPERATE_DONE)
  {
    /* if the previous operation is completed, continue to erase the sector */
    FLASH->ctrl_bit.secers = TRUE;
    FLASH->addr = sector_address;
    FLASH->ctrl_bit.erstr = TRUE;

    /* wait for operation to be completed */
    status = flash_bank1_operation_wait_for(ERASE_TIMEOUT);

    /* disable the secers bit */
    FLASH->ctrl_bit.secers = FALSE;
  }
}
```

For 512K and above Flash memory, its sector erase code is described as follows:

```
#define FLASH_BANK2_START_ADDR      ((uint32_t)0x08080000) /*!< flash start address of bank2 */
#define FLASH_BANK2_END_ADDR        ((uint32_t)0x080FFFFF) /*!< flash end address of bank2 */
if((sector_address >= FLASH_BANK2_START_ADDR) && (sector_address <=
FLASH_BANK2_END_ADDR))
{
  /* wait for last operation to be completed */
  status = flash_bank2_operation_wait_for(ERASE_TIMEOUT);

  if(status == FLASH_OPERATE_DONE)
  {
    /* if the previous operation is completed, continue to erase the sector */
    FLASH->ctrl2_bit.secers = TRUE;
    FLASH->addr2 = sector_address;
    FLASH->ctrl2_bit.erstr = TRUE;

    /* wait for operation to be completed */
    status = flash_bank2_operation_wait_for(ERASE_TIMEOUT);

    /* disable the secers bit */
    FLASH->ctrl2_bit.secers = FALSE;
  }
}
```

Type: MCU

Applicable products: AT32F4xx

Main function: Flash

Minor function: None

Document revision history

Date	Revision	Changes
2022.2.16	2.0.0	Initial release

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services, and ARTERY assumes no liability whatsoever relating to the choice, selection or use of the ARTERY products and services described herein.

No license, express or implied, to any intellectual property rights is granted under this document. If any part of this document deals with any third party products or services, it shall not be deemed a license grant by ARTERY for the use of such third party products or services, or any intellectual property contained therein, or considered as a warranty regarding the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

Unless otherwise specified in ARTERY's terms and conditions of sale, ARTERY provides no warranties, express or implied, regarding the use and/or sale of ARTERY products, including but not limited to any implied warranties of merchantability, fitness for a particular purpose (and their equivalents under the laws of any jurisdiction), or infringement of any patent, copyright or other intellectual property right.

Purchasers hereby agrees that ARTERY's products are not designed or authorized for use in: (A) any application with special requirements of safety such as life support and active implantable device, or system with functional safety requirements; (B) any air craft application; (C) any automotive application or environment; (D) any space application or environment, and/or (E) any weapon application. Purchasers' unauthorized use of them in the aforementioned applications, even if with a written notice, is solely at purchasers' risk, and is solely responsible for meeting all legal and regulatory requirement in such use.

Resale of ARTERY products with provisions different from the statements and/or technical features stated in this document shall immediately void any warranty grant by ARTERY for ARTERY products or services described herein and shall not create or expand in any manner whatsoever, any liability of ARTERY.