

零死角玩转STM32



设置FLASH的读写保护及解除

淘宝：firestm32.taobao.com

论坛：www.chuxue123.com



扫描进入淘宝店铺

主讲内容

01

选项字节与读写保护

02

修改选项字节的过程

03

操作选项字节的库函数

04

实验：设置读写保护及解除

参考资料:《零死角玩转STM32》

“设置FLASH的读写保护及解除” 章节

修改选项字节的内容可修改各种配置，但是，当应用程序运行时，无法直接通过选项字节的地址改写它们的内容，例如，接使用指针操作地址0x1FFFC0000的修改是无效的。要改写其内容时必须设置寄存器FLASH_OPTCR及FLASH_OPTCR1中的对应数据位，寄存器的与选项字节对应位置如下：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				nWRP[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

默认情况下，FLASH_OPTCR寄存器中的第0位OPTLOCK值为1，它表示选项字节被上锁，需要解锁后才能进行修改，当寄存器的值设置完成后，对FLASH_OPTCR寄存器中的第1位OPTSTRT值设置为1，硬件就会擦除选项字节扇区的内容，并把FLASH_OPTCR/1寄存器中包含的值写入到选项字节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				nWRP[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

设置FLASH的读写保护及解除



修改选项字节的过程

综上所述，修改选项字节的配置步骤如下：

- 解锁，在 Flash 选项密钥寄存器 (FLASH_OPTKEYR) 中写入 $\text{OPTKEY1} = 0x0819\ 2A3B$ ；接着在 Flash 选项密钥寄存器 (FLASH_OPTKEYR) 中写入 $\text{OPTKEY2} = 0x4C5D\ 6E7F$ 。
- 检查 FLASH_SR 寄存器中的 BSY 位，以确认当前未执行其它 Flash 操作。
- 在 FLASH_OPTCR 和/或 FLASH_OPTCR1 寄存器中写入选项字节值。
- 将 FLASH_OPTCR 寄存器中的选项启动位 (OPTSTRT) 置 1。
- 等待 BSY 位清零，即写入完成。

设置FLASH的读写保护及解除



操作选项字节的库函数

为简化编程，STM32标准库提供了一些库函数，它们封装了修改选项字节时操作寄存器的过程。

1. 选项字节解锁、上锁函数

对选项字节解锁、上锁的函数：

解锁的时候，它对

FLASH_OPTCR寄存器写入两个解

锁参数，上锁的时候，对FLASH_

OPTCR寄存器的

FLASH_OPTCR_OPTLOCK位置1。

```
1
2 #define FLASH_OPT_KEY1          ((uint32_t)0x08192A3B)
3 #define FLASH_OPT_KEY2          ((uint32_t)0x4C5D6E7F)
4
5 /**
6  * @brief  Unlocks the FLASH Option Control Registers access.
7  * @param  None
8  * @retval None
9  */
10 void FLASH_OB_Unlock(void)
11 {
12     if((FLASH->OPTCR & FLASH_OPTCR_OPTLOCK) != RESET)
13     {
14         /* Authorizes the Option Byte register programming */
15         FLASH->OPTKEYR = FLASH_OPT_KEY1;
16         FLASH->OPTKEYR = FLASH_OPT_KEY2;
17     }
18 }
19
20 /**
21  * @brief  Locks the FLASH Option Control Registers access.
22  * @param  None
23  * @retval None
24  */
25 void FLASH_OB_Lock(void)
26 {
27     /* Set the OPTLOCK Bit to lock the FLASH Option Byte Registers access
28     */
29     FLASH->OPTCR |= FLASH_OPTCR_OPTLOCK;
```


设置FLASH的读写保护及解除

2. 设置读保护级别

解锁后设置选项字节寄存器的RDP位可调用FLASH_OB_RDPConfig完成:

```
1 /**
2  * @brief Sets the read protection level.
3  * @param OB_RDP: specifies the read protection level.
4  *           This parameter can be one of the following values:
5  *           @arg OB_RDP_Level_0: No protection
6  *           @arg OB_RDP_Level_1: Read protection of the memory
7  *           @arg OB_RDP_Level_2: Full chip protection
8  *
9  * /\ Warning /\ When enabling OB_RDP level 2 it's no more possible to go back to level 1 or 0
10 *
11 * @retval None
12 */
13 void FLASH_OB_RDPConfig(uint8_t OB_RDP)
14 {
15     FLASH_Status status = FLASH_COMPLETE;
16
17     /* Check the parameters */
18     assert_param(IS_OB_RDP(OB_RDP));
19
20     status = FLASH_WaitForLastOperation();
21
22     if(status == FLASH_COMPLETE)
23     {
24         *(__IO uint8_t*)OPTCR_BYTE1_ADDRESS = OB_RDP;
25     }
26 }
27 }
```

该函数根据输入参数设置RDP寄存器位为相应的级别，其注释警告了若配置成OB_RDP_Level_2会无法恢复。类似地，配置其它选项时也有相应的库函数，如FLASH_OB_PCROP1Config、FLASH_OB_WRP1Config分别用于设置要进行PCROP保护或WRP保护(写保护)的扇区。

设置FLASH的读写保护及解除



2.设置读保护级别

调用上一步骤中的函数配置寄存器后，还要调用FLASH_OB_Launch函数把寄存器的内容写入到选项字节中。

```
7 FLASH_Status FLASH_OB_Launch(void)
8 {
9     FLASH_Status status = FLASH_COMPLETE;
10
11     /* Set the OPTSTRT bit in OPTCR register */
12     *(__IO uint8_t *)OPTCR_BYTE0_ADDRESS |= FLASH_OPTCR_OPTSTRT;
13
14     /* Wait for last operation to be completed */
15     status = FLASH_WaitForLastOperation();
16
17     return status;
18 }
```

该函数设置FLASH_OPTCR_OPTSTRT位后调用了FLASH_WaitForLastOperation函数等待写入完成，并返回写入状态，若操作正常，它会返回FLASH_COMPLETE。

零死角玩转STM32



THANKS

论坛：www.chuxue123.com

淘宝：firestm32.taobao.com



扫描进入淘宝店铺