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.firebbs.cn



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



修改选项字节的过程

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	
SPR MOD	DB1M	Rese	erved		nWRP[11:0]											
rw	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	
	RDP[7:0]									WDG_ SW	BFB2	BOR_LEV		OPTST RT	OPTLO CK	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rs	rs	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	Reserv	vod.			nWRP[11:0]											
	Reserv	eu		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															



修改选项字节的过程

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	
SPR MOD	DB1M	Rese	erved		nWRP[11:0]											
rw	rw			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	
	RDP[7:0]								nRST_ STOP	WDG_ SW	BFB2	BOR_LEV		OPTST RT	OPTLO CK	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rs	rs	

3	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	
1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
								Res	erved								



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
 OPTKEY1 = 0x0819 2A3B;接着在 Flash 选项密钥寄存器
 (FLASH_OPTKEYR) 中写入 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。



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。

```
2 #define FLASH OPT KEY1
                                     ((uint32 t) 0x08192A3B)
 3 #define FLASH OPT KEY2
                                     ((uint32 t) 0x4C5D6E7F)
     * @brief Unlocks the FLASH Option Control Registers access.
     * @param None
     * @retval None
10 void FLASH OB Unlock(void)
    if((FLASH->OPTCR & FLASH_OPTCR OPTLOCK) != RESET)
       /* Authorizes the Option Byte register programming */
       FLASH->OPTKEYR = FLASH OPT KEY1;
16
       FLASH->OPTKEYR = FLASH OPT KEY2;
17
18 }
19
20 /**
     * @brief Locks the FLASH Option Control Registers access.
     * @retval None
25 void FLASH OB Lock (void)
 26 {
      /* Set the OPTLOCK Bit to lock the FLASH Option Byte Registers access
 28
      FLASH->OPTCR |= FLASH OPTCR OPTLOCK;
 29
```



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:

```
* @brief Sets the read protection level.
     * @param OB RDP: specifies the read protection level.
                This parameter can be one of the following values:
                   @arg OB_RDP_Level_0: No protection
                   @arg OB RDP Level 1: Read protection of the memory
                   @arg OB RDP Level 2: Full chip protection
     * /!\ 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 */
     assert param(IS OB RDP(OB RDP));
19
     status = FLASH WaitForLastOperation();
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。



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)
     FLASH Status status = FLASH COMPLETE;
10
11
     /* Set the OPTSTRT bit in OPTCR register */
     *( IO uint8 t *)OPTCR BYTEO ADDRESS |= FLASH OPTCR OPTSTRT;
12
13
14
     /* Wait for last operation to be completed */
     status = FLASH WaitForLastOperation();
15
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





论坛: www.firebbs.cn

淘宝: firestm32.taobao.com



扫描进入淘宝店铺