



南京凌鸥创芯电子有限公司

# *LKS32MC08x\_在线升级*

@ 2019, 版权归凌鸥创芯所有  
机密文件，未经许可不得扩散



## 目录

|     |                |   |
|-----|----------------|---|
| 1   | 概述 .....       | 1 |
| 2   | 操作说明 .....     | 2 |
| 2.1 | VTOR 寄存器 ..... | 2 |
| 2.2 | 升级范围 .....     | 2 |
| 2.3 | 升级程序编写 .....   | 2 |



## 1 概述

LKS32MC08x 在线升级功能实现。



## 2 操作说明

LKS32MC08x 在线升级功能实现，分成如下几个知识点

### 2.1 VTOR 寄存器

LKS32MC08x 系列芯片，包含了 VTOR 寄存器，该寄存器的用途是，实现中断向量表入口地址重定义。

IAP 模式，实现中断向量表的重映射。在 LKS32MC08X 系列芯片中，包含了系统寄存器 VTOR，其地址为 0xE000\_ED08。用于重新映射中断向量表入口地址。

表 2-1 IAP VTOR 寄存器描述

| 名称   | 复位值 | 偏移 | 位置     | 权限 | 说明                 |
|------|-----|----|--------|----|--------------------|
| VTOR | 0x0 |    | [31:7] | RW | 执行写入操作，写入中断向量表入口地址 |
|      |     |    | [6:0]  | -- | 默认写 0              |

默认值为 0x0，此时中断向量表入口地址为 0x0。当写入非 0 值时，中断向量表入口地址将映射到写入值对应的地址上，立即生效。

### 2.2 升级范围

因为有 VTOR 寄存器，那么可以实现对整个 FLASH 空间全部升级。此时，需要跳转到 RAM 空间（此区域需提前加载升级程序），通过对 FLASH 寄存器的配置，擦除整个 FLASH 的内容，新的 FLASH 的内容，可以通过 UART/I2C/SPI/CAN 等，传递升级数据到 RAM，通过 RAM 中的程序实现对 FLASH 的编程。

同理，我们可以实现对 FLASH 空间部分升级。此时，用户可以跳转到 FLASH 非升级区间（此区域需有在线升级的程序），通过配置 FLASH 控制寄存器实现 FLASH 升级区域的擦除，可以通过 UART/I2C/SPI/CAN 等，传递升级数据，程序实现对 FLASH 的编程。

### 2.3 升级程序编写

第二小节说明了升级程序，可以在 RAM 执行（全擦除 FLASH），也可以在 FLASH 执行（部分



擦除 FLASH)。对应的升级程序的编译环境也会不同。一般而言，选择部分擦除 FLASH 的方式。防止全擦除，在升级过程中发生掉电等事件，导致升级失败，整个 FLASH 无程序。

部分升级，一般在应用程序里面，开辟一块区间，专门用来存放在线升级函数。通过定义这个函数的属性，指定到 FLASH 某个区间。当需要在线升级的时候

- 关闭中断，配置 VTOR，选择新的中断入口地址。或者，直接关闭中断
- 跳转到在线升级函数，执行升级操作
- 完成升级，还原 VTOR，配置为正常的中断入口地址。返回到升级后的程序，开始执行

在线升级函数，指定 FLASH 存放点，如下配置

```

154 |
155 | unsigned int test_update(uint32_t addr);
156 | #pragma arm section code=".ARM.__at_0x4100"
157 | unsigned int test_update(uint32_t addr)
158 | {
159 |     volatile unsigned int dat;
160 |     volatile unsigned int cnt = 0;

```

截图声明了，test\_update(uint32\_t addr);这个在线升级函数放到 0x4100 地址（用户可以根据实际情况，自行安排）

跳转函数，如下配置：

```

5 typedef void (*iapfun)(void);
6
7 void main ()
8 {
9     iapfun jump2app;
10
11     SYS_WR_PROTECT    = PSW_PROT;
12     SYS_CLK_CFG      |= 0x00001FF;
13
14     REG32(0xE00ED08) = 0x0008000; //interrupt entry address remap
15
16
17     jump2app=(iapfun)*(UINT32 *) (0x0008000+4);
18     _set_MSP(0x0008000);
19     jump2app();
20

```

第一个红框，举例了 VTOR 寄存器的配置方法。

第二个红框，指定跳转地点和栈顶指针。