Application Note

SPC11X8/SPD11X8 Flash 存储寿命延长使用指南

2019 年 7 月 – 版本 1

目录

表格列表

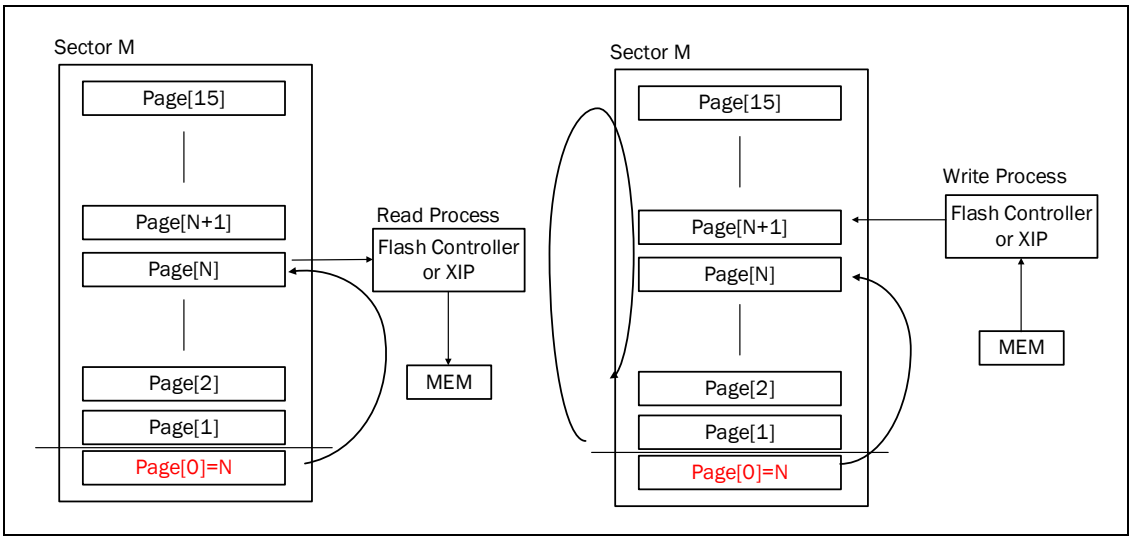
图片列表

1 引言

SPC11X8/SPD11X8 内部集成了 Flash，数据空间最大可达 128KB，每个 Sector(512 byte)可经历 100K 次的擦写，透过 Spintrol 提供之 data_manage 档案管理，可将档案存储寿命延长为 15 倍，亦即最少可支持 1500K(150 万次)次的擦写。

本技巧支持 256 byte 以内的数据量，使用 Sector 内的 16 个 Page(每个 page 256 byte)作为依序存储空间，减少擦写次数，使用原理如下。

图 1-1: 轮序存储表示图

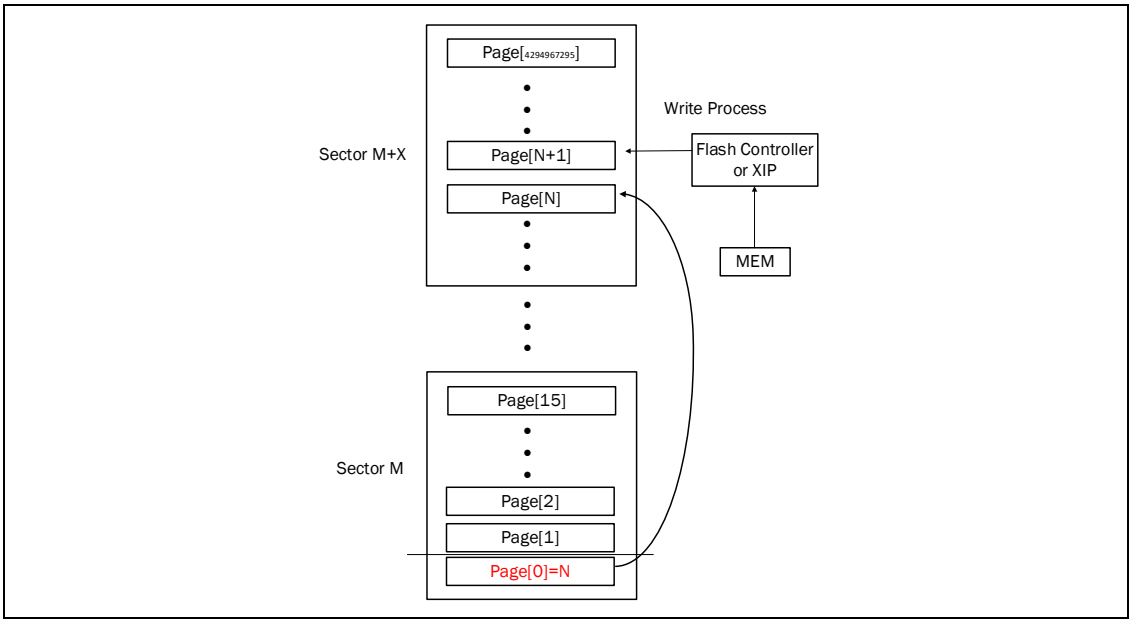


如上图，Page0 内会记录目前使用的 page 指针，每次存储前，会将此指标+1(由此 Page 内 1 翻 0 的个数代表指标数值，如 0xff…f8 代表 N=3，代表此 Sector 目前 Page[1]~Page[3]被写，下一次写入操作将会写入到 Page[4]，Bit 位翻转方式亦可大幅减少操作 Page[0]的抹除数)，随后将数据存入指标+1 所对应的 page 内。读取前，亦会读取此指标，按照指标的指示读取对应 Page，即，每次读取的数值都是最新一个 Page 存储的数值。按照此策略，只有当一个 Sector 的所有 Page 都写满时，才进行一次 Sector 的擦除操作，如此一来便可以大幅减少 Flash 的擦写次数。

Spintrol 所提供之 Data manage 代码不保证数据写入前后的正确性，若使用者有需求，请自行加入 checksum 机制确保数据之正确性。

以上策略能够保证 Flash 擦除次数为 150 万次，若欲突破 150 万次的擦写次数上限，可以将多个物理 Sector 合并成一个看待，在 SPC11X8/SPD11X8 中，Page0 存储的指标位数是 32 位，理论上支持 2^32=最多可支持计数 4294967295 次，也即，理论上此策略可支持每个 Sector 可包含 4294967295 个 Page，且能使 Flash 最高抹除次数可提升至 400 亿次左右，如下图所示。

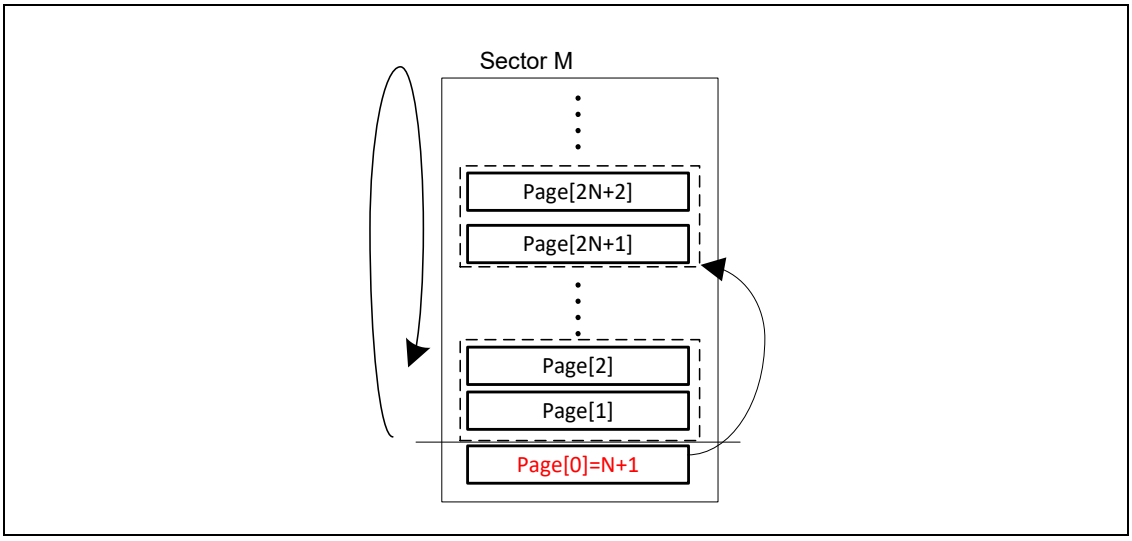
图 1-2:　提升擦写次数表示图



注意：SPC11X8/SPD11X8 相关 demo 例程里，若要求每 Page 256 bytes，那么每个 Sector 只有两个 Pagae，为了使每个 Sector 拥有 16 个 Page，合并了 8 个 Sector 当做一个 Sector 看待，具体细节可参考例程代码。

以上描述的存储单位是 256bytes 的 Page，若想单一存储量大于 256 byte，可以整并两个以上的 Page 作为一个存储单元，用户可自行修改代码。如下图所示

图 1-3:　提升单位存储量表示图



2 调用 API 进行存储

2.1 Data Manage 存储功能

打开 data_manage.h，先设定好 Sector 的位置，请注意起始位置必须与 Sector 的物理地址对齐，地址结尾必须是 00，需要注意的是，Flash 的此 Sector 内在这之前不能存有有效数值，否则将被复写丢失，也不能挪作他用。

图 2-1: 设定存储位置

```
#define OffsetADDRInFlash 0x10000
#define FlashBaseADDR 0x10000000
#define OperationADDR FlashBaseADDR + OffsetADDRInFlash
```

存储的使用范例如下图，其中 ai16myData 可以是任一种数据结构之指针(pointer)，而注意此结构大小不得超过 256 byte。

```
/* Put the input value in gsVSP.i16Input */
gsVSP.i16Input = i16VSP;
i16VSPFiltered = AvgFilter_ReadFilterValue(&gsVSP);
ai16myData[0] = i16VSP;
ai16myData[1] = i16VSPFiltered;
```

先将 ADC 读到的数值(i16VSP)存入先定义好的 array 中，再呼叫以下函数，将数据存入预先划定好的 sector 内。Page 指标自动+1，超过 15 自动折回到 1，均由内部代码自行处理，使用者即可使用。

```
printf("Data stored...\n\n");
DATA_ByteSeriesSave(DATA_FLASH_ACCESS_PRIMARY_ADDR,  /* Sector start address     */
                    (uint8_t*)ai16myData,            /* pointer to array stored  */
                    sizeof(ai16myData));             /* number of bytes of array */
```

2.2 Data Manage 读取功能

读取功能，参考代码如下

```
int16_t ai16myData[2];
DATA_ByteSeriesRead(DATA_FLASH_ACCESS_PRIMARY_ADDR,  /* Sector start address     */
                    (uint8_t*)ai16myData,            /* pointer to array stored  */
                     sizeof(ai16myData));            /* number of bytes of array */
printf("Data sread...\n");
printf("Task_C4 = %d sec\n"   ,u16VTimerC[2]);
printf("VSP = %dmV,  "        ,ai16myData[0]);
printf("Filtered VSP = %dmV\n",ai16myData[1]);
```

只要呼叫此程序，代码会由 Page0 的纪录自动判断上一次存储时的 Page 指标，并从中读取最新的存储信息。

3 修订记录

表 3-1: 文档修订记录

日期	版本	修改内容
2019-07-25	1	初始版本