FEE 速度的优化方法

Peter Chen（Peter_Chen@wtmec.com）

1. 现状

S32K3 芯片内并不带有 EEPROM 模拟硬件。为解决客户对于 EEPROM 数据存储的需求，S32K3 使用 AUTOSAR 的 FEE 模块，使用 PFLASH 或者 DFLASH 来模拟 EEPROM。由于 S32K3 的 PFLASH 的写入寿命只能保证 1000 次，而 DFLASH 的写入寿命可以达到 10 万次，所以推荐使用 DFLASH 的 Sector 来模拟 EEPROM。FEE 模块是 BSW（基础软件）中的一部分，客户会向第三方 AUTOSAR 软件供应商购买。为了能让客户在不购买 BSW 的前提下也能有相应的功能，NXP 在 RTD 中提供了一个免费的实现。客户在使用过程中反馈 FEE 在初始化阶段的耗时太长，导致无法在规定的时间内完成初始化。为了研究 FEE 的代码，我将 FEE 样例程序移植到了 Windows 平台，请参考 https://github.com/chenzch/Fee_On_Windows。运行了以后发现速度慢的原因主要是默认的配置对于 Flash 的读取是按照 1 字节来配置的，初始化的时候如果需要读取 60 字节的头部数据，就需要连续进入 60 次 MainFunction 周期。本文以 S32DS for S32 Platform 3.5 + SW32K3_S32M27x_RTD_R21-11_4.0.0_HF02_D2407 作为演示，来提升 FEE 初始化的速度。

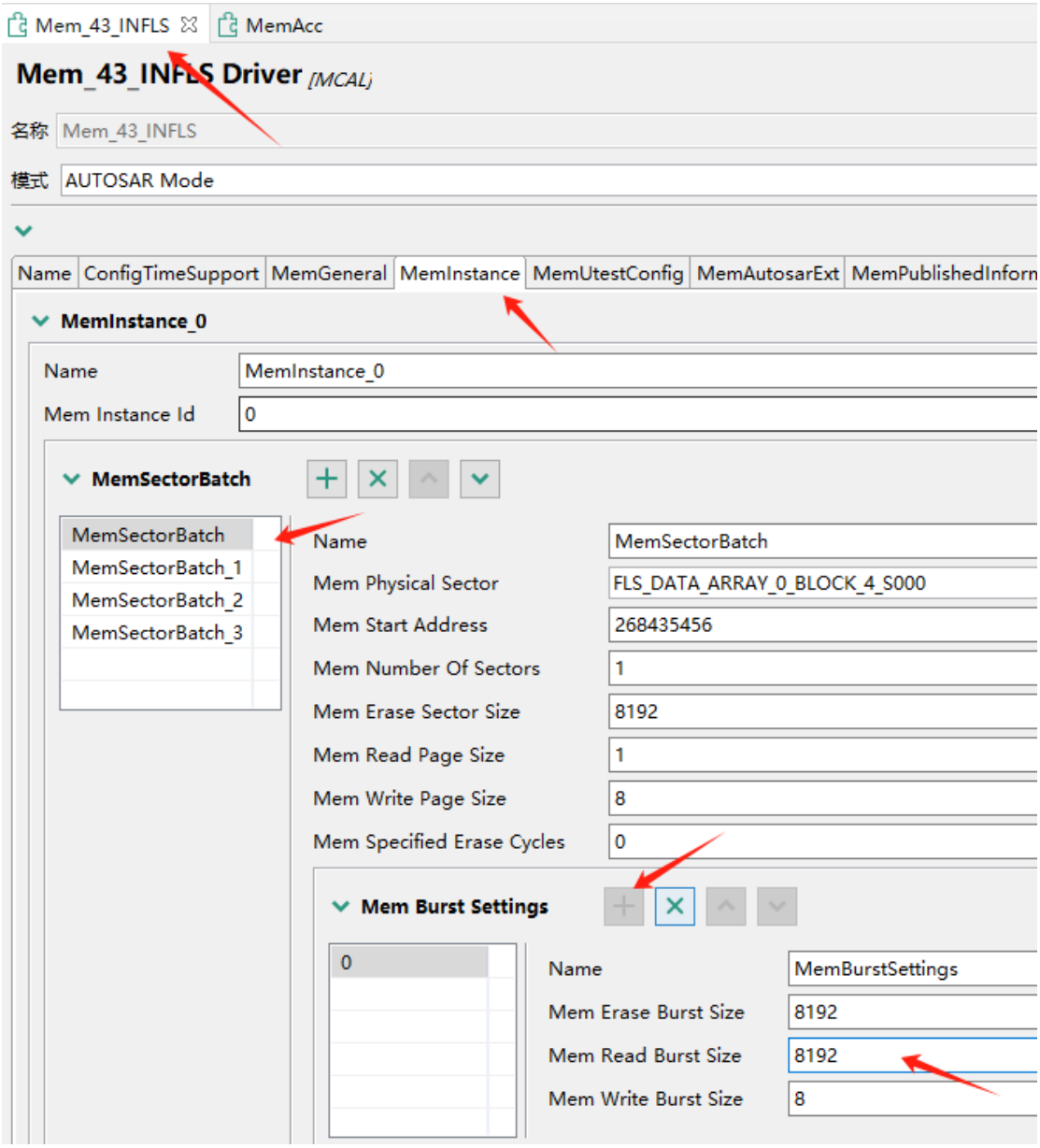
2. 修改配置

演示的样例使用 RTD 自带的 Fee_Example_S32K344。MemAcc_PBcfg.c 文件中列出了 MemAcc 模块的配置信息，样例中 4 个 Sector 的 ReadPageSize 和 ReadPageBurstSize 都被设置成了 1，如下图。这样 FEE 每次最多只能从 Flash 读取 1 个字节，初始化的时候就需要进入多次读取周期才能将头部信息读完。

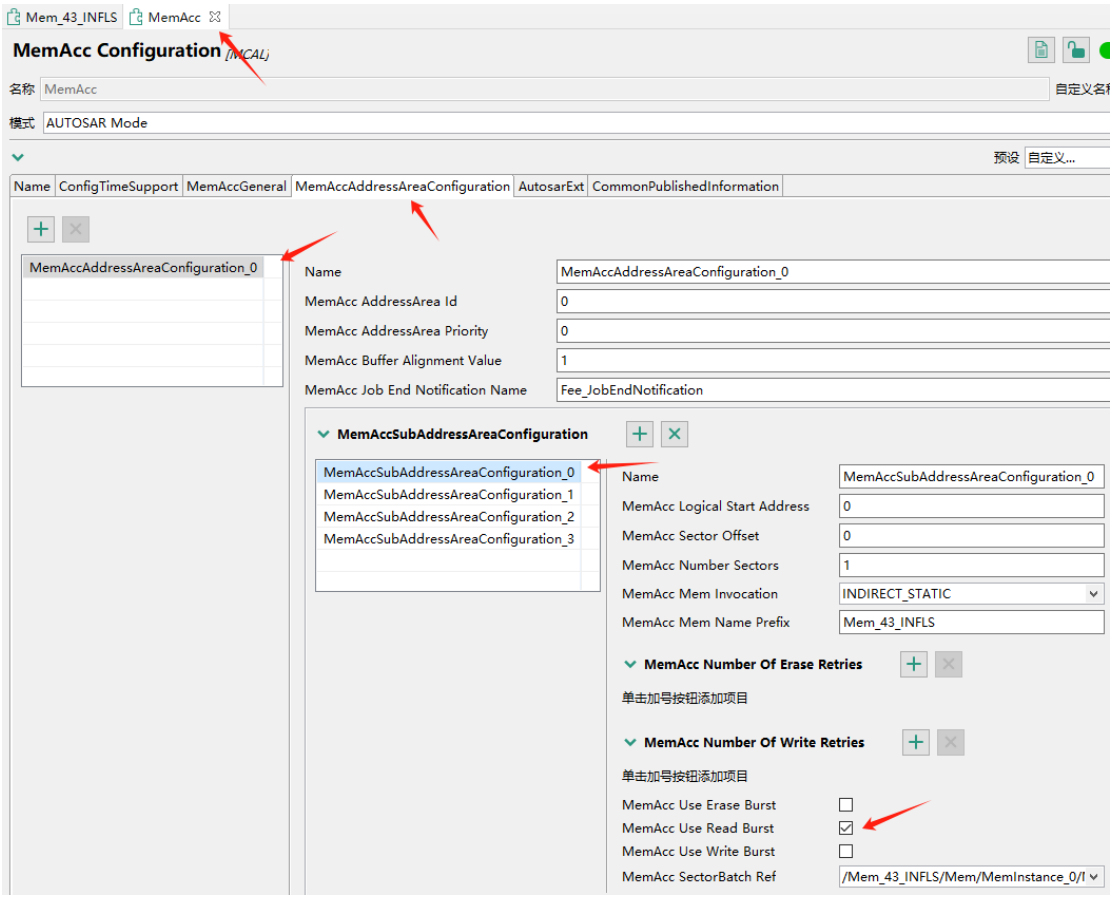
```
181 static const MemAcc_SubAddressAreaType MemAcc_MemAccAddressAreaConfiguration_0_SubAreas[4U] =
182 {
183     {
184         /* MemAccSubAddressAreaConfiguration_0 */
185         0U,                                                    /* LogicalStartAddress */
186         268435456U,                                            /* PhysicalStartAddress */
187         8192U,                                                 /* Length */
188         0U,                                                    /* NumOfEraseRetries */
189         0U,                                                    /* NumOfWriteRetries */
190         0U,                                                    /* BurstSettings */
191         MEMACC_MEM_INDIRECT_STATIC,                            /* MemInvocation */
192         &(MemAcc_MemApis[0]),                                  /* MemApi */
193         0U,                                                    /* MemInstanceId */
194         {
195             8192U,                                             /* SectorEraseSize          */
196             1U,                                                /* ReadPageSize             */
197             8U,                                                /* WritePageSize            */
198             8192U,                                             /* SectorEraseBurstSize     */
199             1U,                                                /* ReadPageBurstSize        */
200             8U                                                 /* WritePageBurstSize       */
```

修改方法是将 ReadPageBurstSize 改成 8192，然后在 MemAcc 中打开 Burst 读取。由于 EEPROM 在访问的时候是会有单字节访问的情况出现的，所以 ReadPageSize 只能设置为 1。如果 PageSize 设置成了其他值，那么对于这个 Sector 的读取就必须按照这个 PageSize 进行对齐。其实当你打开 Burst 读取的时候，这个 PageSize 就只有验证对齐的功能，对于读取速度没有任何影响，所以可以保留为 1。

第一步是要将 Mem_43_INFLS 中 MemInstance 里的每一个 MemSector 都添加 Burst 设置，并将 Read Burst Size 设置成 8192。



第二步是将 MemAcc 组件中 MemAccAddressAreaConfiguration 中的每一个 MemAccSubAddressAreaConfiguration 的 Use Read Burst 打开。



这样生成的代码中就会打开 BurstRead 功能，每次最大读取量就变成了 8192 字节。

```
181 static const MemAcc_SubAddressAreaType MemAcc_MemAccAddressArea
182 {
183     {
184         /* MemAccSubAddressAreaConfiguration_0 */
185         0U,
186         268435456U,
187         8192U,
188         0U,
189         0U,
190         0U | MEMACC_BURST_READ,
191         MEMACC_MEM_INDIRECT_STATIC,
192         &(MemAcc_MemApis[0]),
193         0U,
194         {
195             8192U,
196             1U,
197             8U,
198             8192U,
199             8192U,
200             8U
201         },
```

```
181 static const MemAcc_SubAddressArea
182 {
183     {
184         /* MemAccSubAddressAreaConf
185         0U,
186         268435456U,
187         8192U,
188         0U,
189         0U,
190         0U,
191         MEMACC_MEM_INDIRECT_STATIC,
192         &(MemAcc_MemApis[0]),
193         0U,
194         {
195             8192U,
196             1U,
197             8U,
198             8192U,
199             1U,
200             8U
201         },
```

3. 修改驱动

按照以上方法操作以后，初始化速度应该会有 4 倍左右的提升，如果速度还不能满足需求，就要按照以下方法来修改 C40 的驱动代码。MemAcc 对于 Flash 的读、写、擦除操作都是通过 C40 这个驱动来完成的。SW32K3_S32M27x_RTD_R21-11_4.0.0_HF02_D2407 这个版本中的 C40 驱动（C40_Ip.c : 799 行）只实现了 1、2、4 字节的读取功能。也就是说，我们将 Burst 读取设置了 8192 个字节，那么在一个 MainFunction 周期内最多可以读取 8192 个字节，但是是拆分成很多个 4 字节的读取操作来完成的，对于 S32K3 来说，最经济的读取操作是按照 8 字节对齐进行读取。这样就需要修改 C40 的驱动代码，使得最小的读取单位可以扩展成 8 个字节。当然也可以扩展成无限的字节数，但是一个 MainFunction 中的多次调用读取函数的开销，在没有设置回调函数的情况下与在一次读取函数中多次实际访问差别并不大。考虑到 FEE 实际对于 FLASH 的访问强度在初始化以外的情况下并不会太大，所以没有必要冒风险增加任意字节的访问方法。一共需要修改 C40_Ip_Types.h 和 C40_Ip.c 这 2 个文件，修改的内容请参考以下 2 个链接。

https://github.com/chenzch/Fee_On_Windows/blob/main/C40_Ip_Types.h

https://github.com/chenzch/Fee_On_Windows/blob/main/C40_Ip.c