

SYD8811 CODEB DATA 使用说明

从文章《内存和代码的分布》可知，SYD8811 的 512KB 的代码分配为了两个代码区，所以默认情况下代码最大为 $(512-8)/2=252\text{KB}$ ！当然 SYD8811 也支持 Flash data 模式，也就是说可以把 504KB 的代码区分为 176KB 的代码区和 328KB 的数据，一般而言，FLASH data 已经能够满足大多数需求！

但是有一种情况 FLASH data 还是不够，比如我一定需要 328KB 的数据和 85KB 的代码。但是这时候我还有 40K 的数据要找地方存储，这些数据接受 OTA 后数据丢失的情况，这时候 CODEB DATA 模式就发挥作用了，这里我们在 88KB 的代码备份区（除了当前运行代码的区域之外的另外一个代码区）存储 40KB 的数据，这些数据在 OTA 完成后被擦除也能够接受！

因为代码备份区前面有一些数据是非代码的，是用于存储加密的固态码。所以数据备份区可以操作的大小为本区大小减去 16KB，比如上面的代码区为 88KB，这里只能操作 72KB 的数据。

这里提供三个 API 函数：

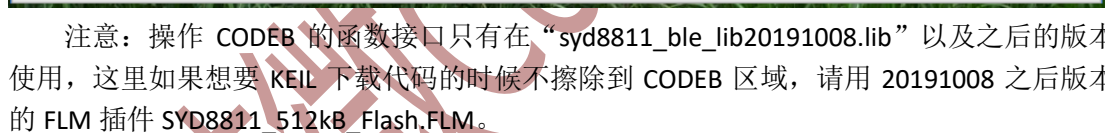
```
825  
826     extern uint8_t EraseCodeB(uint32_t addr, uint8_t sector_num);  
827     extern uint8_t ReadCodeB(uint32_t addr, uint16_t len, uint8_t *p_buf);  
828     extern uint8_t WriteCodeB(uint32_t addr, uint16_t len, uint8_t *p_buf);  
829 #endif
```

这三个函数的操作方法和 FLASH DATA 一样，这里不再累述！

这里提供一个测试 CODEB data 的例程，请看工程“SYD8811_SDK\Source Code\SYD8811_peripheral_misc\flash_internal_custom_CodeB\KEIL”，main 函数如下：

```
67 while(1)  
68 {  
69     if(GPIO_Pin_Read( U32BIT(KEY1)))  
70     {  
71         for(i=0;i<WRITE_READ_LEN;i++)  
72         {  
73             tx_buff[i]=base_data+i;  
74         }  
75         //注意ble_flash_erase函数会进行擦除操作，所以必须要保证地址是正确的  
76         EraseFlashData(FLASH_INTERNAL_DATA_ADDR+addr,1); //注意这里是标准的flash操作，在写数据之前必须要擦除，除非要写的地址区域已经被擦除过  
77         WriteFlashData(FLASH_INTERNAL_DATA_ADDR+addr,WRITE_READ_LEN, tx_buff);  
78         //注意：CODEB的函数为代码大小减去16k，比如代码区为160KB，那么CODEB的大小九尾160-16=144KB  
79         EraseCodeB(FLASH_INTERNAL_DATA_ADDR+addr,1); //注意这里是标准的flash操作，在写数据之前必须要擦除，除非要写的地址区域已经被擦除过  
80         WriteCodeB(FLASH_INTERNAL_DATA_ADDR+addr,WRITE_READ_LEN, tx_buff);  
81         dbg_printf("write addr:%04x len:%04x\r\n",FLASH_INTERNAL_DATA_ADDR+addr,WRITE_READ_LEN);  
82         DBGHEXDUMP(" ",tx_buff,WRITE_READ_LEN);  
83         base_data++;  
84         GPIO_Pin_Turn(LED1);  
85         delay_ms(600);  
86     }  
87     if(GPIO_Pin_Read( U32BIT(KEY2)))  
88     {  
89         GPIO_Pin_Turn(LED2);  
90         memset(rx_buff,0,WRITE_READ_LEN);  
91         ReadFlashData(FLASH_INTERNAL_DATA_ADDR+addr,WRITE_READ_LEN, rx_buff);  
92         //注意：CODEB的函数为代码大小减去16k，比如代码区为160KB，那么CODEB的大小九尾160-16=144KB  
93         ReadCodeB(FLASH_INTERNAL_DATA_ADDR+addr,WRITE_READ_LEN, rx_buff);  
94         dbg_printf("read addr:%04x len:%04x\r\n",FLASH_INTERNAL_DATA_ADDR+addr,WRITE_READ_LEN);  
95         DBGHEXDUMP(" ",rx_buff,WRITE_READ_LEN);  
96         if(memcmp(rx_buff,tx_buff,WRITE_READ_LEN)==0)  
97         {  
98             dbg_printf("memcmp ok!\r\n");  
99         }  
100         else  
101         {  
102             dbg_printf("memcmp failed!\r\n");  
103         }  
104         delay_ms(600);  
105     }  
106     if(GPIO_Pin_Read( U32BIT(KEY3)))  
107     {  
108         GPIO_Pin_Turn(LED3);  
109         addr+=32;  
110         dbg_printf("write read addr:%04x\r\n",FLASH_INTERNAL_DATA_ADDR+addr);  
111         delay_ms(600);  
112     }  
113 }  
114 }  
115 }
```

下载完程序后通过按下按键 1 2 3 可以实现不同的功能，具体请看代码，这里做一个简单的测试，看到的 log 如下，确实能够当成正常的 LFASH 使用：



SYD8811_512kB_Flash.FLM