

ME05 APP AB分区

ME05 A/B Swap

（基于ME05 MCU规格：
PFlash0地址范围0x0000_0000~0x0007_FFFF（512 KB, Block0） ,
PFlash1地址范围0x0008_0000~0x000F_FFFF（512 KB, Block1）。配置AB分区的demo。）

1. 引言

本文档针对ME05系列MCU实现硬件AB Swap功能。AB Swap允许在两个物理Flash块（Block0和Block1）之间动态切换程序执行地址，适用于固件更新、故障恢复等场景。与MD14不同，ME05的Flash容量更大（每个Block 512 KB）。

2. FLASH分区划分

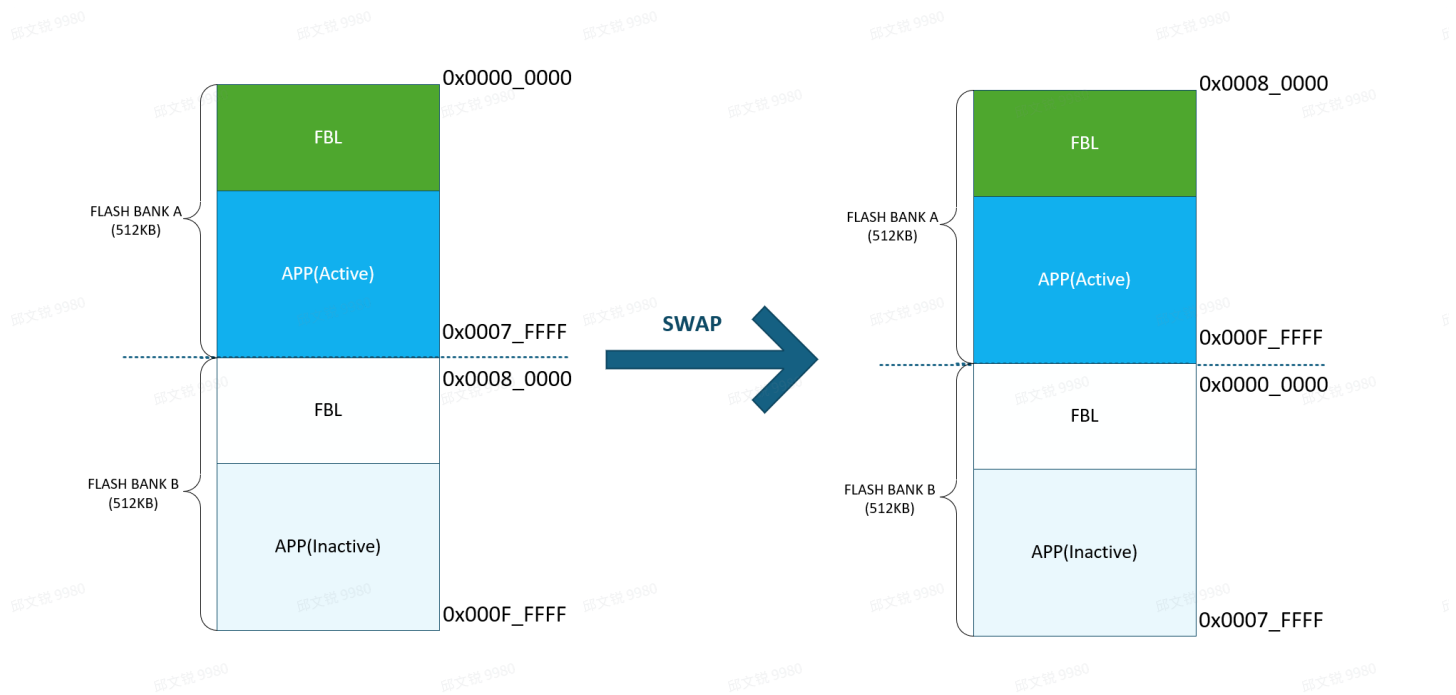
ME05的Flash分为两个独立物理Block，支持异步操作（无阻塞）：

Module	Start Address	End Address	Size	Sector Size	Program Unit	Block
PFlash0	0x0000_0000	0x0007_FFFF	512 KB	2 KB	8 B	Block0
PFlash1	0x0008_0000	0x000F_FFFF	512 KB	2 KB	8 B	Block1
DFlash	0x0010_0000	0x0013_FFFF	256 KB	1 KB	8 B	Block2
HCU_NVR	0x1000_0000	0x1000_03FF	1 KB	1 KB	8 B	Block2
OTP_NVR	0x1001_0000	0x1001_03FF	1 KB	-	8 B	Block2
CUS_NVR	0x1003_0000	0x1003_03FF	1 KB	1 KB	8 B	Block2

- **Block0 (PFlash0)**：地址范围0x0000_0000 ~ 0x0007_FFFF（512 KB），用于存储主程序或备份程序。
- **Block1 (PFlash1)**：地址范围0x0008_0000 ~ 0x000F_FFFF（512 KB），用于存储备用程序或数据。

分区优势：

- 不同Block支持异步操作，程序执行和Flash写入互不阻塞。



3. 核心要点

3.1 AB Swap原理

AB Swap通过硬件命令交换两个Block的地址映射：

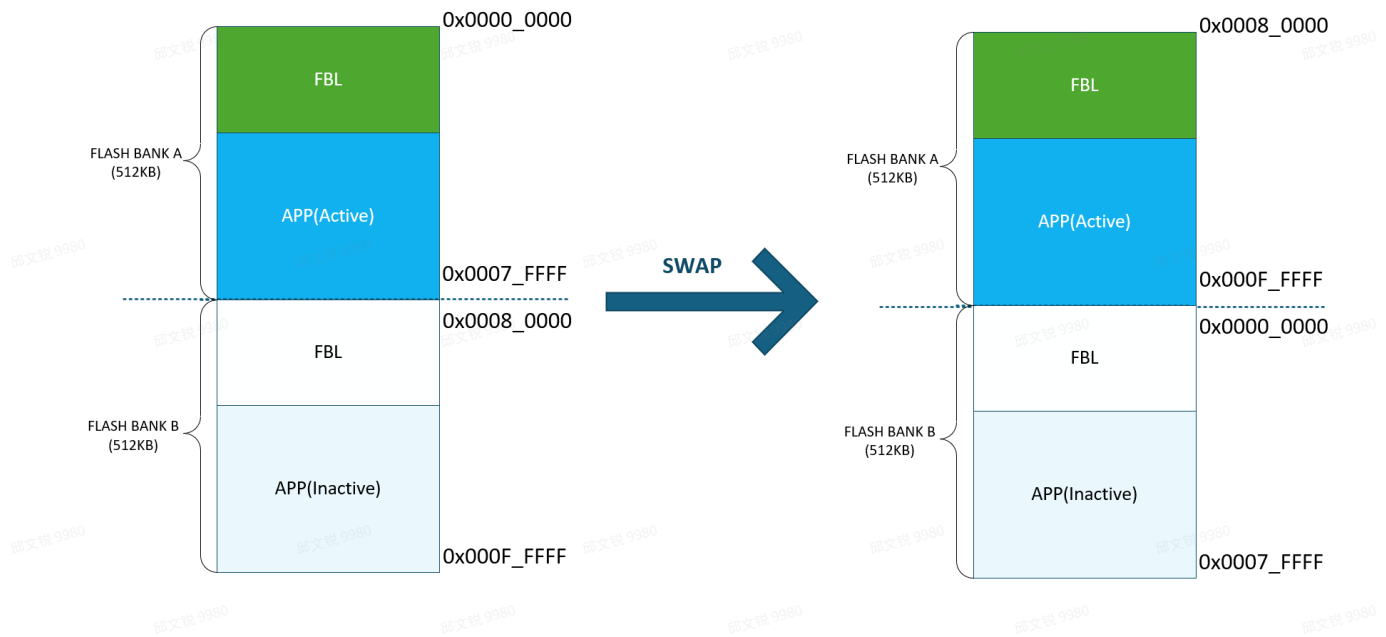
- **Swap前：**

- PFlash0地址：0x0000_0000 ~ 0x0007_FFFF（对应物理Block0）。
- PFlash1地址：0x0008_0000 ~ 0x000F_FFFF（对应物理Block1）。

- **Swap后：**

- PFlash0地址变为0x0008_0000 ~ 0x000F_FFFF（原Block1地址）。
- PFlash1地址变为0x0000_0000 ~ 0x0007_FFFF（原Block0地址）。

- **效果：**MCU复位后，程序从新映射的地址启动，实现固件切换。



- Swap命令翻转Block的基地址，不移动数据；物理存储位置不变，但逻辑地址交换。

3.2 AB Swap操作方式

通过内部Flash命令（命令码0x30）执行Swap，由SDK函数驱动：

- 关键函数： `FLASH_DRV_BootSwap()`。
- 操作流程：
 - a. 确认Flash空闲（无擦写操作）。
 - b. 发送Swap命令到Flash控制器寄存器。
 - c. 复位MCU使Swap生效。
- 注意：Swap操作需在程序控制下执行，避免中断干扰。

3.3 查询MCU当前运行位置

通过EFM（Embedded Flash Module）状态寄存器判断运行Block：

- 寄存器地址：EFM->STS（偏移量4h）。
- 关键位：
 - Bit 0：
 - 0：代码运行在Block0（原PFlash0地址）。
 - 1：代码运行在Block1（原PFlash1地址）。
- 代码示例：

代码块

```
1 status_t Fls_ReadSwap(uint8_t * DestinationAddressPtr)
```

```

2  {
3      status_t status = STATUS_SUCCESS;
4      *DestinationAddressPtr = (EFM->STS & EFM_STS_BOOT_INFO_MASK) >>
        EFM_STS_BOOT_INFO_SHIFT;
5      return status;
6  }

```

4. 应用案例及代码讲解

4.1 版本

- Config Tool Version: 2.7.6。
- ME05 SDK version: 1.3.1 HF001。

4.2 程序要求

- 实现功能：
 - 代码运行在Block0时，点亮绿灯；运行在Block1时，点亮蓝灯。
 - 串口打印当前运行位置（如"APP is locating PFLASH0"）。
- 硬件依赖：GPIO控制LED，UART串口模块。

4.3 代码讲解

4.3.1 Swap检测

- 读取EFM寄存器判断启动位置。

代码块

```

1  Fls_ReadSwap(&boot_info);
2  if(boot_info == 0U)
3  {
4      PRINTF("APP is locating PFLASH0\r\n");
5      PINS_DRV_ClearPins(LED_BLUE_PORT, (1U << LED_BLUE_PORT_INDEX));
6      PINS_DRV_WritePin(LED_GREEN_PORT, LED_GREEN_PORT_INDEX, 1U);
7  }
8  else
9  {
10     PRINTF("APP is locating PFLASH1\r\n");
11     PINS_DRV_ClearPins(LED_GREEN_PORT, (1U << LED_GREEN_PORT_INDEX));
12     INS_DRV_WritePin(LED_BLUE_PORT, LED_BLUE_PORT_INDEX, 1U);
13 }

```

4.3.2 Swap执行流程

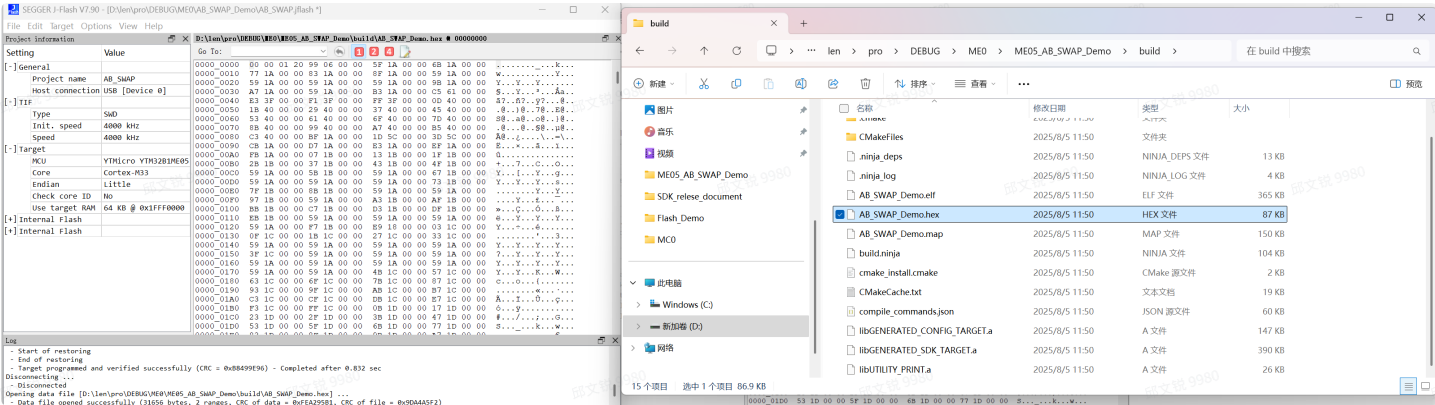
- 在main循环中检测LPTMR 1s中断标志是否置1，置1就开始Swap

代码块

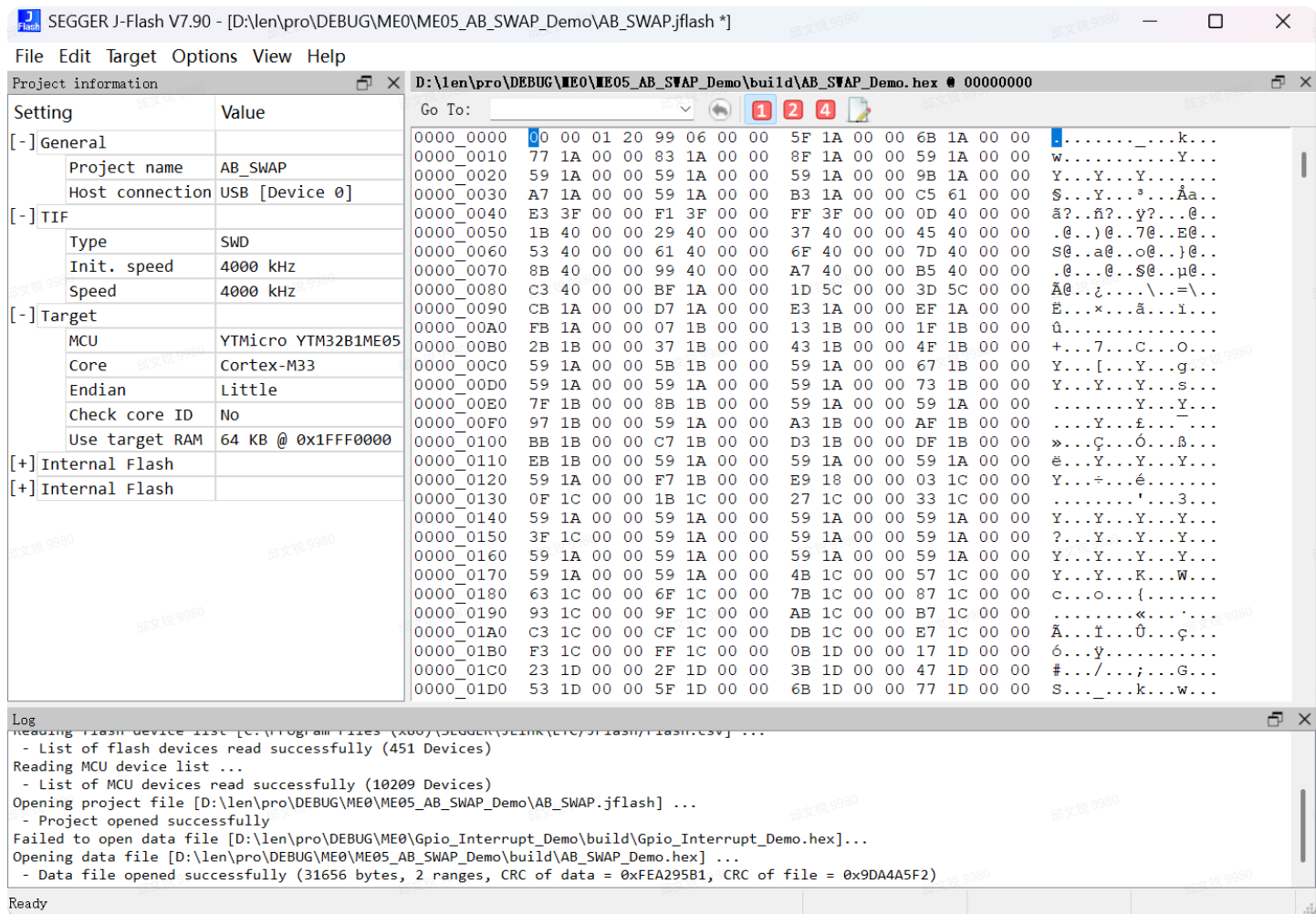
```
1  /* Infinite loop */
2  /* USER CODE BEGIN WHILE */
3  while (1)
4  {
5      /* USER CODE END WHILE */
6      /* USER CODE BEGIN 3 */
7      if(lptmr_flag == 1U)
8      {
9          lptmr_flag = 0U;
10         PRINTF("LPTMR interrupt\r\n");
11         PINS_DRV_TogglePins(LED_RED_PORT, (1U << LED_RED_PORT_INDEX));
12         FLASH_DRV_BootSwap(0);
13         SystemSoftwareReset();
14     }
15 }
16 /* USER CODE END 3 */
```

5. 测试验证

5.1 将工程程序编译的文件加载到JFLASH



5.2 擦除芯片，烧录程序（快捷键F4，F6），完成PFLASH0面烧录



5.3 偏移地址映射PFLASH1面烧录

5.3.1 设置地址偏移到0x0008 0000

SEGGER J-Flash V7.90 - [D:\len\pro\DEBUG\ME0\AB_SWAP_Demo\AB_SWAP.jflash *]

File Edit Target Options View Help

Project information

Go To: 00000000

Relocate...

Delete range...

Eliminate blank areas...

Project name AB_SWAP

Host connection USB [Device 0]

TIF

Type SWD

Init. speed 4000 kHz

Speed 4000 kHz

Target

MCU YTMicro YTM32B1ME05

Core Cortex-M33

Endian Little

Check core ID No

Use target RAM 64 KB @ 0x1FFF0000

Internal Flash

Internal Flash

Log

- Start of restoring
- End of restoring
- Target programmed and verified successfully (CRC = 0xB8499E96) - Completed after 0.832 sec

Disconnecting ...

- Disconnected

Opening data file [D:\len\pro\DEBUG\ME0\ME05_AB_SWAP_Demo\build\AB_SWAP_Demo.hex] ...

- Data file opened successfully (31656 bytes, 2 ranges, CRC of data = 0xFE295B1, CRC of file = 0x9DA4A5F2)

Relocate data by given offset

Not connected

SEGGER J-Flash V7.90 - [D:\len\pro\DEBUG\ME0\AB_SWAP_Demo\AB_SWAP.jflash *]

File Edit Target Options View Help

Project information

Go To: 80000

Setting Value

General

Project name AB_SWAP

Host connection USB [Device 0]

TIF

Type SWD

Init. speed 4000 kHz

Speed 4000 kHz

Target

MCU YTMicro YTM32B1ME05

Core Cortex-M33

Endian Little

Check core ID No

Use target RAM 64 KB @ 0x1FFF0000

Internal Flash

Internal Flash

Log

- Start of restoring
- End of restoring
- Target programmed and verified successfully (CRC = 0xB8499E96) - Completed after 0.832 sec

Disconnecting ...

- Disconnected

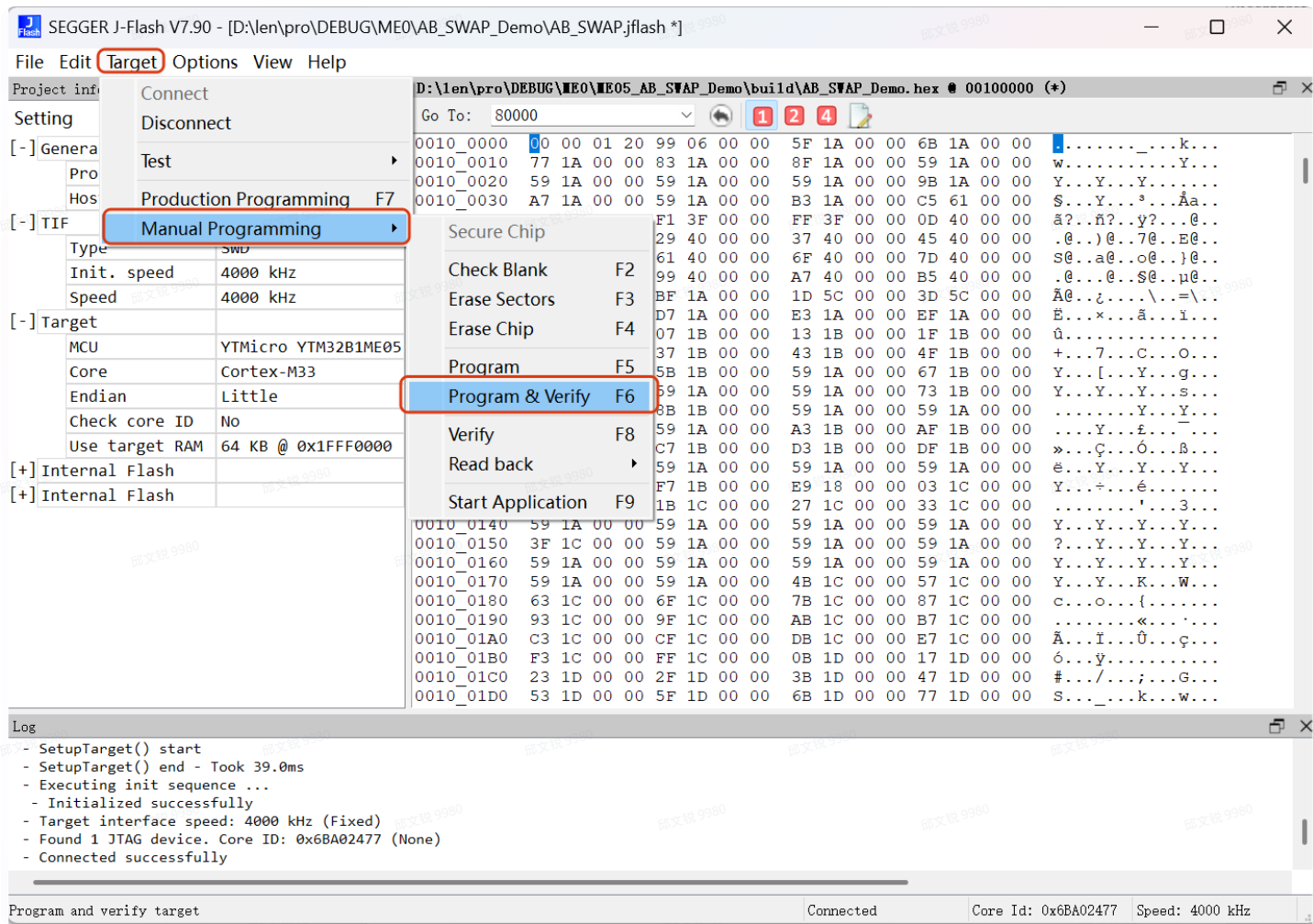
Opening data file [D:\len\pro\DEBUG\ME0\ME05_AB_SWAP_Demo\build\AB_SWAP_Demo.hex] ...

- Data file opened successfully (31656 bytes, 2 ranges, CRC of data = 0xFE295B1, CRC of file = 0x9DA4A5F2)

Ready

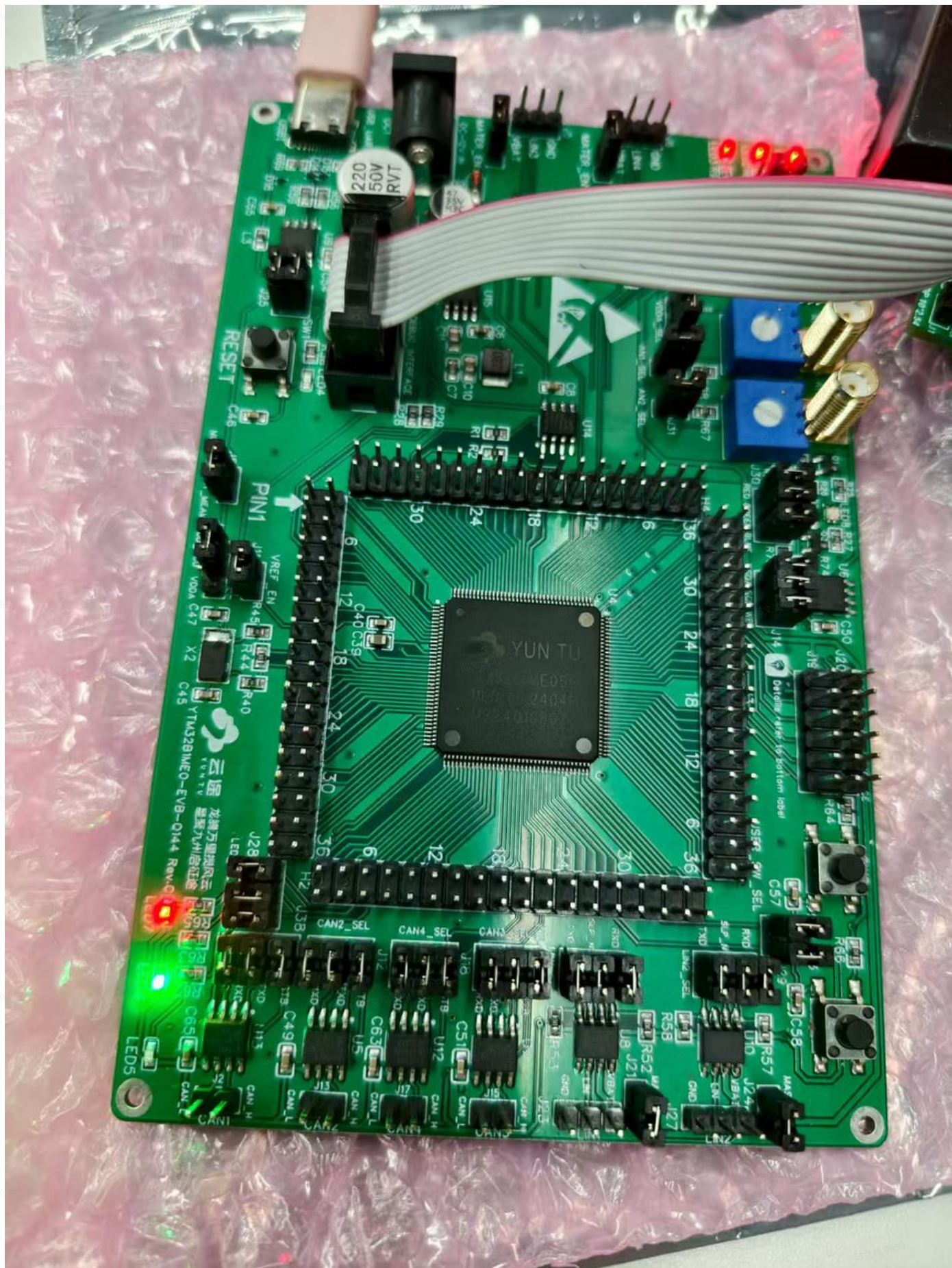
Not connected

5.3.2 烧录PFLASH1



5.4 运行测试

- 上电行为：
 - 初始运行Block0：绿灯亮，蓝灯灭，串口打印"APP is locating PFLASH0".



- 等待LPTMR 1s计时标志置起来，串口打印"LPTMR interrupt".
- 触发Swap后复位：运行Block1：绿灯灭，蓝灯亮，串口打印"APP is locating PFLASH1".



- 再次触发Swap：切换回Block0，形成循环（切换间隔为1s）。
- 串口输出：

核心函数代码：

代码块

```
1  // 交换Block地址
2  status_t FLASH_BootSwap(uint32_t instance)
3  {
4      DEV_ASSERT(instance < EFM_INSTANCE_COUNT);
5      EFM_Type *base = s_efmBase[instance];
6      flash_state_t *state = s_FlashStatePtr[instance];
7      DEV_ASSERT(state != NULL);
8      status_t status;    /* Return code variable */
9      uint32_t tmp = 0x5a5a5a5aU;
10     bool pre_state = state->async;
11
12     /* Check IDLE to verify the previous command is completed */
13     if (false == FLASH_GetIdleStatus(base))
14     {
15         status = STATUS_EFM_BUSY;
16     }
17     else
18     {
19         /* Boot swap synchronously */
20         state->async = false;
21         base->CTRL &= ~EFM_CTRL_DONEIE_MASK;
22         state->cmdParam.cmdCode = FEATURE_EFM_BOOT_SWAP_CMD_CODE;
23         state->cmdParam.pdata = &tmp;
24         state->cmdParam.word_size = 0U;
25         #if defined(FEATURE_EFM_BOOT_SWAP_TAG_ADDR)
26             state->cmdParam.dest = (uint32_t *)FEATURE_EFM_BOOT_SWAP_TAG_ADDR;
27             state->cmdParam.word_size = 1U;
28         #endif
29         /* Calling flash command sequence function to execute the command */
30         status = FLASH_LaunchCommandSequence(instance);
31         state->async = pre_state;
32         if (state->async == true)
33         {
34             base->CTRL |= EFM_CTRL_DONEIE_MASK;
35         }
36     }
37     return status;
38 }
```

注意事项

1. 地址偏移：烧录Block1时，工具需设置偏移地址0x0008_0000。

- 2. **复位必需**：Swap后必须复位MCU才能生效。
- 3. **Block保护**：避免在Swap过程中进行Flash擦写。

本方案已在ME05硬件验证，确保AB Swap功能稳定高效。用户可基于此框架扩展应用逻辑



ME05_AB_SWAP_Demo.zip

2.05MB

