

WM_W800_参数区使用说明

V1.2

北京联盛德微电子有限责任公司 (winner micro)

地址：北京市海淀区阜成路 67 号银都大厦 18 层

电话：+86-10-62161900

公司网址：www.winnermicro.com

文档修改记录

版本	修订时间	修订记录	作者	审核
V0.1	2019/9/25	[C]创建文档	Cuiyc	
V0.2	2020/7/8	统一字体	Cuiyc	
V1.0	2020/8/10	升级版本号	Cuiyc	
V1.1	2021/2/23	更新用户区大小, 与 SDK 保持一致	Cuiyc	
V1.2	2021/5/13	更新用户区变化的编译脚本调整	Cuiyc	

目录

文档修改记录	2
目录	3
1 引言	5
1.1 编写目的	5
1.2 预期读者	5
1.3 术语定义	5
1.4 参考资料	5
2 QFLASH 参数区布局	6
2.1 物理层参数区	6
2.2 用户参数区	7
2.3 系统参数区域	7
3 物理层参数区	8
3.1 物理层参数介绍	8
3.2 物理层参数写入阶段	8
3.3 物理层参数的使用	8
4 系统参数区	9
4.1 系统参数介绍	9
4.2 系统参数的使用	9
4.2.1 初始化阶段	9
4.2.2 参数使用阶段	11
5 用户参数区	12

5.1	用户参数	12
5.2	用户区使用	12
5.2.1	用户参数区的操作.....	12
5.2.2	用户参数区的调整规则.....	12
5.2.3	用户参数区的双备份机制.....	17

WinnerMicro

1 引言

1.1 编写目的

本文档主要用于阐述 W800 中的 QFLASH 布局，关键参数区和系统参数区使用以及用户参数区处理。

1.2 预期读者

该文档适用的读者包括研发人员、测试人员、架构师等。

1.3 术语定义

序号	术语/缩略语	说明/定义
1	QFLASH	Quad-SPI FLASH
2	SECBOOT	Second Boot, relative to ROM
3	ROM	Read-Only Memory

1.4 参考资料

无

2 QFLASH 参数区布局

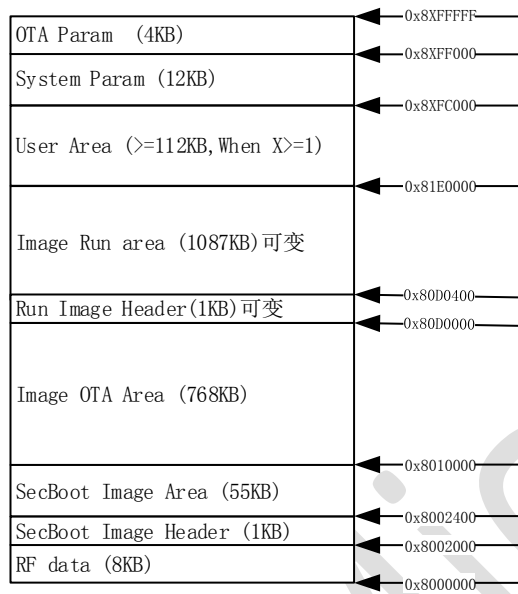


图 2-1

本文档以 X=1 为例，即 Flash 容量为 2MByte。

2.1 物理层参数区

地址空间：0x8000000-0x8000FFF，共 4KByte

参数内容：

MAC 地址和 RF 参数。

参数布局：

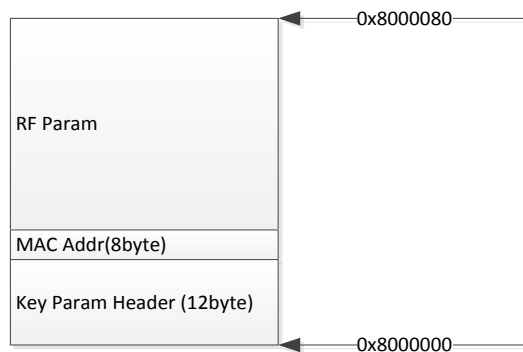


图 2-2

2.2 用户参数区

地址空间：0x81E0000-0x81FBFFF，共 112KByte

参数内容：

用于用户存放自定义参数时使用。

参数布局：

用户自定义

2.3 系统参数区域

地址空间：0x81FC000-0x81FEFFF，共 12KByte

参数内容：

系统运行时所需的相关参数

参数布局：

MAGIC Number:4Byte	
PARTITION_NUM:2Byte	MODIFY_CNT:2Byte
RESERVED:4Byte	
RESERVED:2Byte	Length:2Byte(整个参数的大小,包含 CRC 值, 由系统参数决定)
Data Content(系统参数决定)	
CRC Value:4Byte (CRC 之前的内容的值)	

- 1) 系统参数 1 区：0x81FC000-0x81FCFFF
- 2) 系统参数 2 区：0x81FD000-0x81FDFFF
- 3) 系统参数 3 区：0x81FE000-0x81FEFFF

3 物理层参数区

3.1 物理层参数介绍

W800 模块工作所需要的 MAC 地址，以及 Wi-Fi 收发机工作所需要的 RF 校准参数

3.2 物理层参数写入阶段

W800 芯片或者模块生产时写入

3.3 物理层参数的使用

W800 模块启动时会从关键参数区把所需参数读取出来使用。

物理层参数具有备份机制。

4 系统参数区

4.1 系统参数介绍

系统参数是指 W800 模块运行时所需要的联网，接口配置，模式配置等的参数，具体如下：

- 1) Wi-Fi 相关 (SSID, BSSID, KEY, 信道列表, 节电标志, 速率设置, 区域码, 工作模式)
- 2) IP 信息 (静态 IP, DHCP 使能信息, NTP 服务器, DNS 服务器)
- 3) 接口配置 (UART 配置)
- 4) BT 参数
- 5) 其他参数 (WEB)

4.2 系统参数的使用

4.2.1 初始化阶段

系统参数区具有备份机制，通过 CRC 和 MODIFY_CNT 校验值确定使用哪个参数区的内容作为系统运行时使用的参数，具体机制为：

- 1) 参数区 CRC 均正确的情况下，依据 MODIFY_CNT 选取使用的当前参数
- 2) 参数区 CRC 只有一个正确的情况下，选择 CRC 正确的参数区作为当前参数，另外一个参数区更新为当前参数区的值
- 3) 参数区 CRC 都不正确的情况下，首先尝试参数恢复，如果尝试恢复后，参数依然都不正确，则使用默认参数值作为运行时使用参数，同时，更新参数区的内容为默认参数。

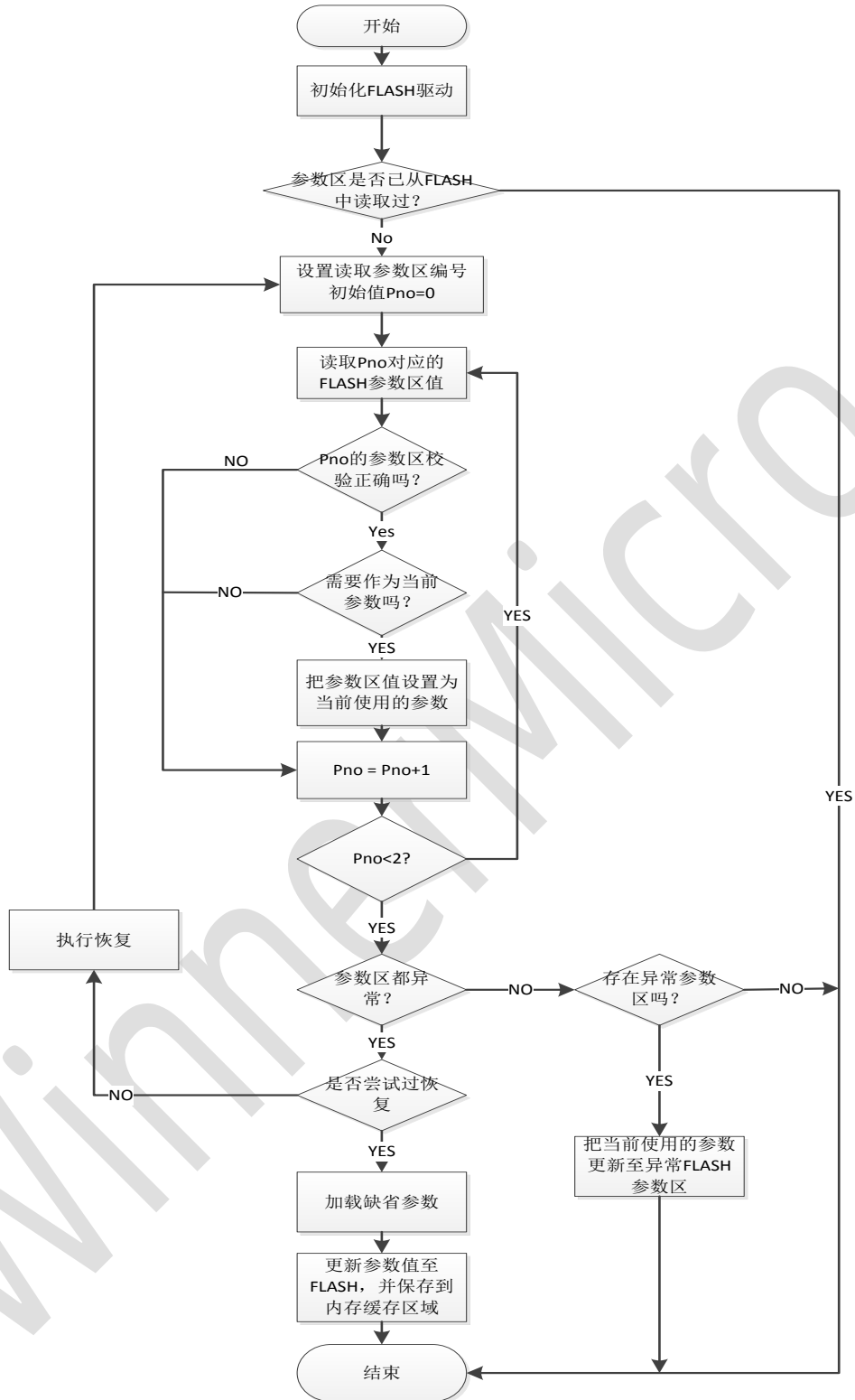


图 5-1

4.2.2 参数使用阶段

1) 参数获取

系统参数区除了存放于 QFLASH 的两个区域外，还会在初始化的时候在内存中备份一份，以便于运行时的使用，防止频繁访问 QFLASH。

2) 参数写入

(1) 系统启动时，第一次初始化或者参数区有破坏，会写参数区

(2) 运行中，系统参数更新，会写参数区

5 用户参数区

5.1 用户参数

W800 使用者期望存储自定义的参数或者运行日志。

5.2 用户区使用

5.2.1 用户参数区的操作

W800 的 SDK 会增加针对用户参数区的操作机制,保证用户针对参数区的操作仅使用相对地址(相对 USER_ADDR_START)即可实现。

5.2.2 用户参数区的调整规则

W800 的默认 QFLASH 的布局所能提供给用户的区域为 240KByte。但是,当前的 W800 用户参数区设置是依据尽可能大的代码区来设计的。

5.2.2.1 用户参数区的调整规则:

1) 依据用户编译的 w800.img 确定的所用运行区空间

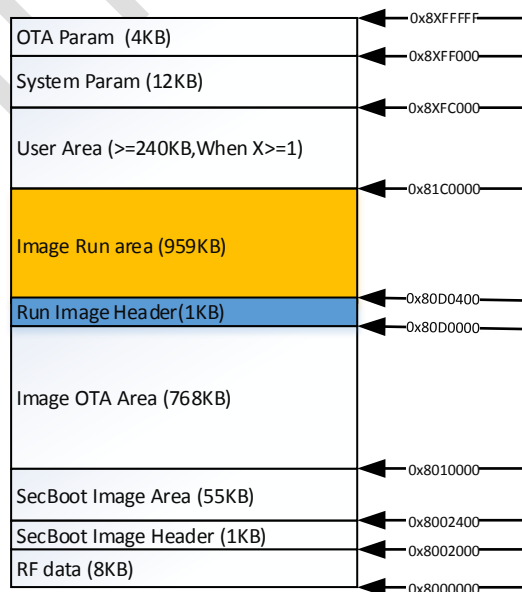


图 5-1

2) 依据 w800.img 的压缩比来确定所用的升级区空间

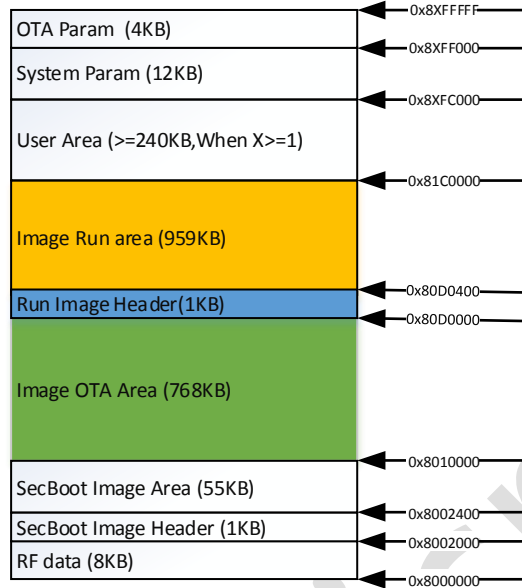


图 5-2

- 3) 依据 w800.img 的大小按照 QFLASH 的 BLOCK(64Kbyte) 区间向上取整划分(需要重点关注)。
- 4) 依据 IMAGE 的划分结果重新确定用户空间的起始地址。
- 5) 根据新划分的空间调整 W800 SDK 的宏定义确定新的用户空间起始地址

```

: /**FLASH MAP**/
:
: /**Flash Base Address */
: #define FLASH_BASE_ADDR (0x8000000UL)
:
: /**Upgrade image area*/
: #define CODE_UPD_START_ADDR (0x8010000UL)
:
: /**Run-time image header area*/
: #define CODE_RUN_START_ADDR (0x80D0000UL)
:
: /**Area can be used by User*/
: #define USER_ADDR_START (0x81C0000UL)

```

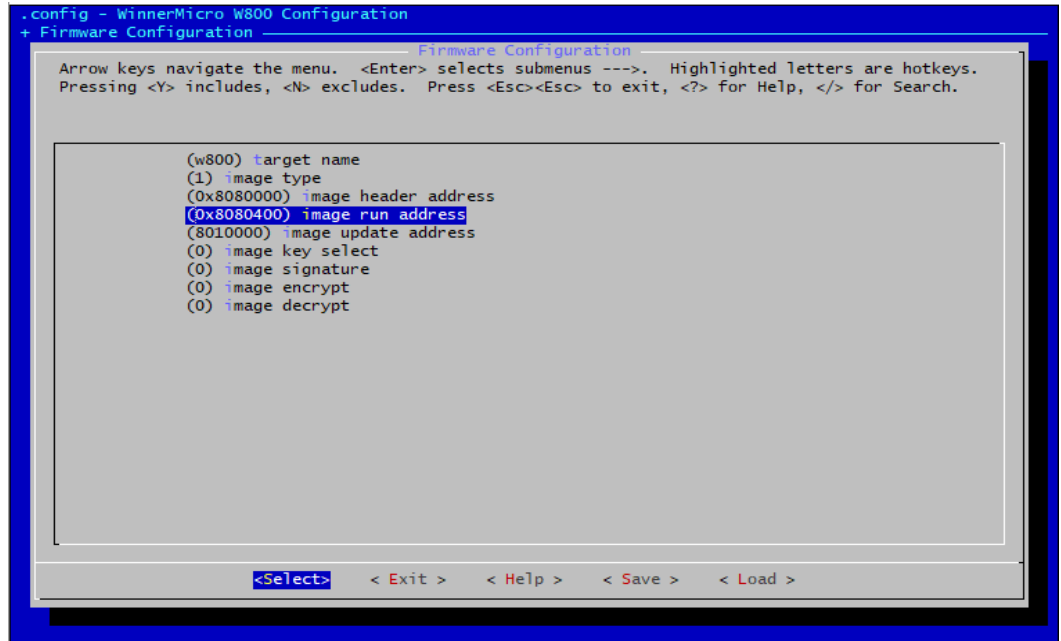
- 6) 调整用户空间后，还需要更新配置文件和链接文件

配置文件调整：

- (1) 针对使用 linux 或者 cygwin 方式使用命令编译的用户，需要调整文件：

- 使用 make menuconfig 进入界面调整 Image Headeraddress 和

Image run header address 的位置 (如下图)



- 如果不想使用上述界面，可以直接修改 tools\w800 目录下的.config 文件下面两个值：

CONFIG_W800_IMAGE_HEADER=8080000

CONFIG_W800_RUN_ADDRESS=8080400

- (2) 针对使用 CDK 工程的用户，需要修改：tools\w800\utilities 下 aft_build_project.sh 里的两个变量的值：

run_img_header=8080000

run_img_pos=8080400

链接文件调整：

修改 ld\w800 目录下 gcc_csky.ld，修改 I-SRAM 的 ORIGIN 值 (如下)：

I-SRAM : ORIGIN = 0x08080400 , LENGTH = 0x120000 /* I-SRAM */

7) 重新 make 编译烧录 w800.flc 文件，模块启动后，用户参数区即变为新的设定值。

注意：

如果客户使用的 V1.00.02 版本之前的 SDK，需要同步更新一下 V1.00.02 版本相关的

修改，才可使用。

5.2.2.2 举例

如果用户编译的 IMAGE 大小为

W800.img: 560KByte

压缩后 img: 400KByte

把 IMAGE 的大小向上取 64KByte 的整数倍 (重要), 则

运行区空间: 576KByte

升级区空间: 448KByte

配置步骤如下:

1) 用户的新空间如图黄色部分

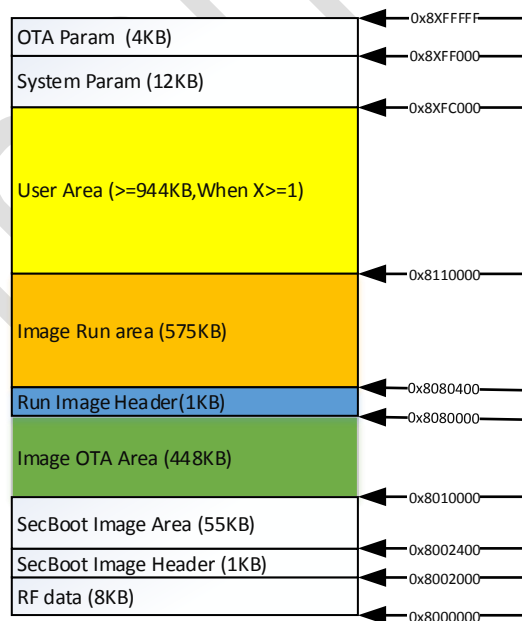


图 6-3

2) 新的代码空间调整为:

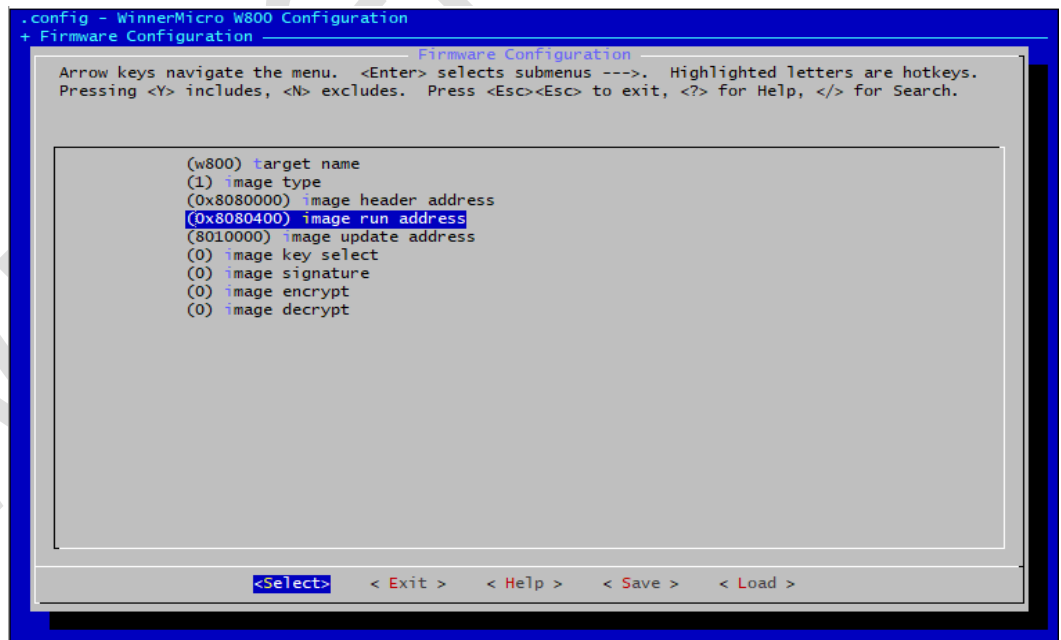
```
/**Flash Base Address */  
#define FLASH_BASE_ADDR (0x8000000UL)  
  
/**Upgrade image area*/  
#define CODE_UPD_START_ADDR (0x8010000UL)  
  
/**Run-time image header area*/  
#define CODE_RUN_START_ADDR (0x8080000UL)  
  
/**Area can be used by User*/  
#define USER_ADDR_START (0x8110000UL)
```

3)调整用户空间后，还需要更新配置文件和链接文件

配置文件调整：

(3) 针对使用 linux 或者 cygwin 方式使用命令编译的用户，需要调整文件：

- 使用 make menuconfig 进入界面调整 Image Headeraddress 和 Image run header address 的位置



- 如果不想使用上述界面，可以直接修改 tools\w800 目录下的.config 文件下面两个值：

CONFIG_W800_IMAGE_HEADER=8080000

`CONFIG_W800_RUN_ADDRESS=8080400`

- (4) 针对 CDK 用户, 需要修改: tools\w800\utilities 下 aft_build_project.sh 里的两个变量的值:

`run_img_header=8080000`

`run_img_pos=8080400`

链接文件调整:

修改 ld\w800 目录下 gcc_csky.ld, 修改 I-SRAM 的 ORIGIN 值 (如下):

`I-SRAM : ORIGIN = 0x08080400 , LENGTH = 0x120000 /* I-SRAM */`

- 4) 重新 make 编译烧录 w800.flc 文件, 模块启动后, 用户参数区即变为新的设定值。

说明:

如果客户使用的 V1.00.02 版本之前的 SDK, 需要同步更新一下 V1.00.02 版本相关的修改, 才可使用。

5.2.3 用户参数区的双备份机制

如果用户参数区会记录关键信息, 建议用户实现双备份机制, 主区和备区按照 4Kbyte 间隔划分。