

# WM\_W600\_参数区使用说明

## V1.1

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](http://www.winnermicro.com)

## 文档修改记录

版本	修订时间	修订记录	作者	审核
1.0	2018/9/6	[C]创建文档	崔艳昌	
1.1	2018/10/12	增加图形编号	崔艳昌	

## 目录

1	引言 .....	3
1.1	编写目的 .....	3
1.2	预期读者 .....	3
1.3	术语定义 .....	3
1.4	参考资料 .....	3
2	QFLASH 参数区布局 .....	4
2.1	物理层参数区 .....	4
2.2	QFLASH 参数区域 .....	4
2.3	用户参数区 .....	5
2.4	系统参数区域 .....	5
3	物理层参数区 .....	6
3.1	物理层参数介绍 .....	6
3.2	物理层参数写入阶段 .....	6
3.3	物理层参数的使用 .....	6
4	QFLASH 参数区 .....	7
4.1	QFLASH 参数区介绍 .....	7
4.2	QFLASH 参数区写入阶段 .....	7
4.2.1	QFLASH 参数区使用 .....	7
5	系统参数区 .....	8
5.1	系统参数介绍 .....	8
5.2	系统参数的使用 .....	8
5.2.1	初始化阶段 .....	8
5.2.2	参数使用阶段 .....	10
6	用户参数区 .....	11
6.1	用户参数 .....	11
6.2	用户区使用 .....	11
6.2.1	用户参数区的操作 .....	11
6.2.2	用户参数区的调整规则 .....	11
6.2.3	用户参数区的双备份机制 .....	14

## 1 引言

### 1.1 编写目的

本文档主要用于阐述 W600 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

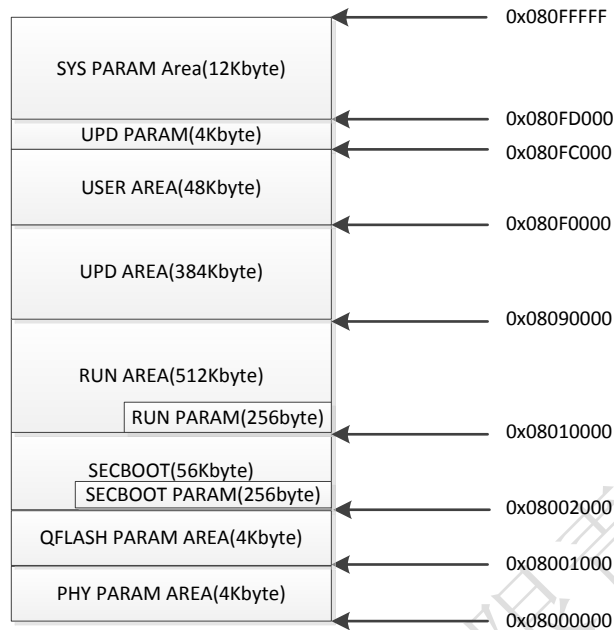


图 2-1

### 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:

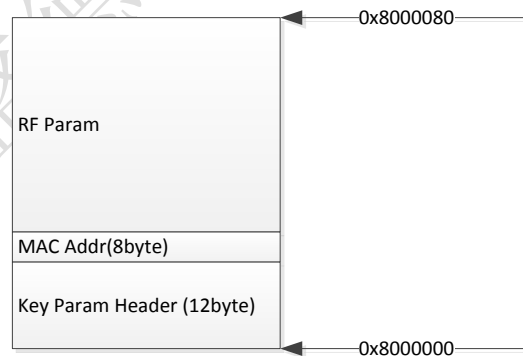


图 2-2

### 2.2 QFLASH 参数区域

地址空间: 0x8001000-0x8001FFF, 共 4kbyte

参数内容:

QFLASH 参数头、Security Level 和 CHIP ID 以及 QFLASH 参数。

参数布局:

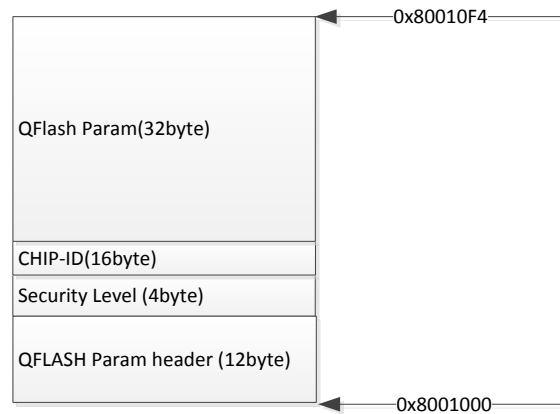


图 2-3

### 2.3 用户参数区

地址空间: 0x80F0000-0x80FBFFF, 共 48Kbyte

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

### 2.4 系统参数区域

地址空间: 0x80FD000-0x80FFFFFF, 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: 0x80FD000-0x80FDFFF
- 2) 系统参数 2 区: 0x80FE000-0x80FEFFF
- 3) 系统参数 3 区: 0x80FF000-0x80FFFFFF

### 3 物理层参数区

#### 3.1 物理层参数介绍

W600 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

W600 芯片或者模块生产时写入

#### 3.3 物理层参数的使用

W600 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 QFLASH 参数区

### 4.1 QFLASH 参数区介绍

W600 芯片 ROM 的安全级别设置，芯片 ID 以及 QFLASH 工作参数的存储

### 4.2 QFLASH 参数区写入阶段

W600 芯片生产阶段。

#### 4.2.1 QFLASH 参数区使用

W600 启动时使用，获取安全级别或者芯片 ID，或者针对 QFLASH 的某些操作时使用，具有备份。



## 5 系统参数区

### 5.1 系统参数介绍

系统参数是指 W600 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, SDIO, HSPI 模式配置)
- 4) 其他参数 (WEB)

### 5.2 系统参数的使用

#### 5.2.1 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。

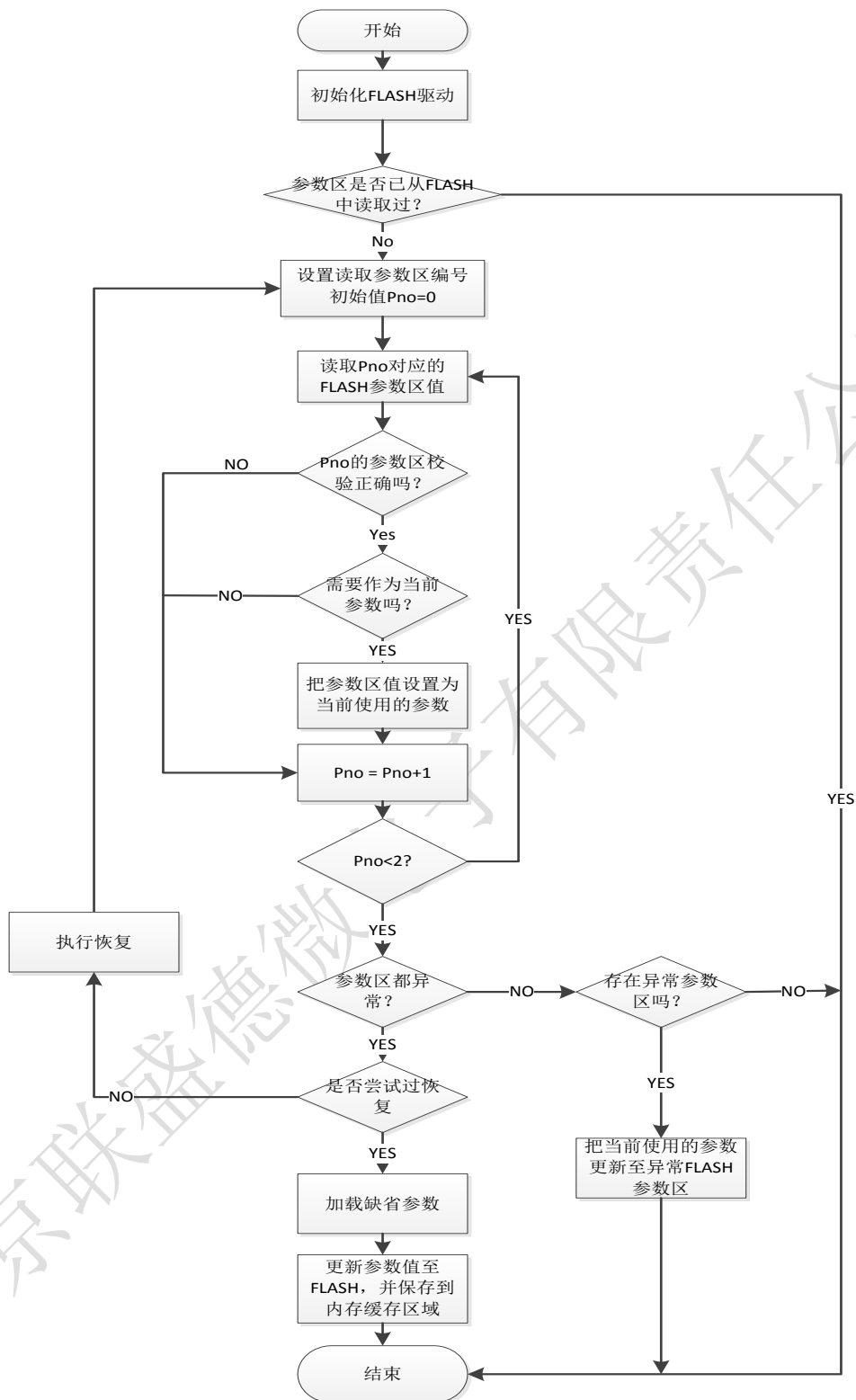


图 5-1

### 5.2.2 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

## 6 用户参数区

### 6.1 用户参数

W600 使用者期望存储自定义的参数或者运行日志。

### 6.2 用户区使用

#### 6.2.1 用户参数区的操作

W600 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。

#### 6.2.2 用户参数区的调整规则

W600 的默认 QFLASH 的布局所能提供给用户的区域为 48Kbyte。但是，当前的 W600 用户参数区设置是依据代码区最大化来设计的。

##### 6.2.2.1 用户参数区的调整规则：

- 1) 依据用户编译的 WM\_W600\_SEC.img 确定的所用运行区空间

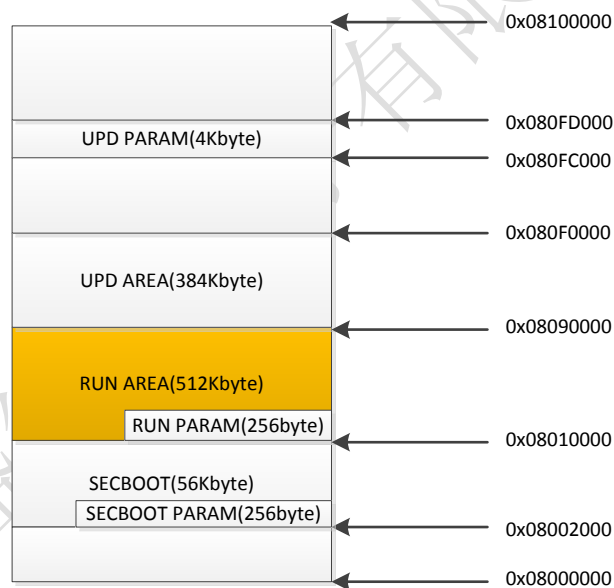


图 6-1

- 2) 依据用户编译的 WM\_W600\_GZ.img 确定的所用升级区空间

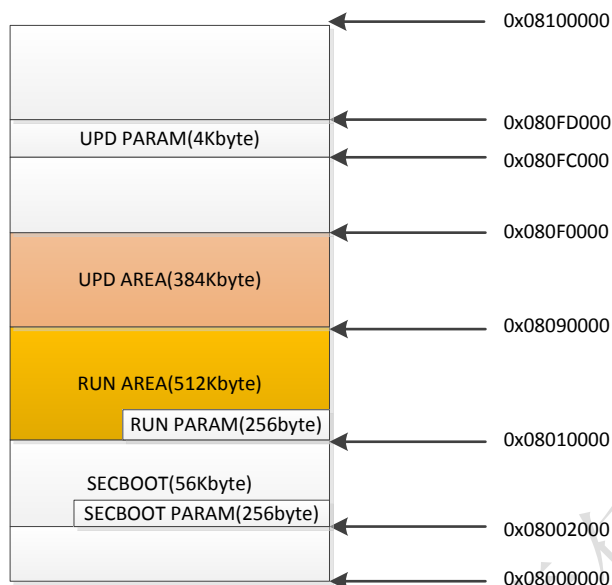


图 6-2

- 3) 依据 WM\_W600\_SEC.img 和 WM\_W600\_GZ.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 W600 SDK 的宏定义确定新的用户空间起始地址

```

/**Upgrade image header area & System parameter area */
#define CODE_UPD_HEADER_ADDR (FLASH_BASE_ADDR + 0xFC000)
#define TLS_FLASH_PARAM1_ADDR (FLASH_BASE_ADDR + 0xFD000)
#define TLS_FLASH_PARAM2_ADDR (FLASH_BASE_ADDR + 0xFE000)
#define TLS_FLASH_PARAM_RESTORE_ADDR (FLASH_BASE_ADDR + 0xFF000)
#define TLS_FLASH_END_ADDR (FLASH_BASE_ADDR + 0xFFFFF)

#define FLASH_KEY_PARAM_AREA_LEN (0x2000)

#define SECBOOT_HEADER_ADDR (FLASH_BASE_ADDR + FLASH_KEY_PARAM_AREA_LEN)
#define SECBOOT_HEADER_AREA_LEN (0x100)

#define SECBOOT_AREA_ADDR (SECBOOT_HEADER_ADDR + SECBOOT_HEADER_AREA_LEN)
#define SECBOOT_AREA_LEN (0x10000 - FLASH_KEY_PARAM_AREA_LEN - SECBOOT_HEADER_AREA_LEN)

/**Run-time image header area*/
#define CODE_RUN_HEADER_ADDR (SECBOOT_AREA_ADDR + SECBOOT_AREA_LEN)
#define CODE_RUN_HEADER_AREA_LEN (0x100)

/**Run-time image area*/
#define CODE_RUN_START_ADDR (CODE_RUN_HEADER_ADDR + CODE_RUN_HEADER_AREA_LEN)
#define CODE_RUN_AREA_LEN (512*1024 - CODE_RUN_HEADER_AREA_LEN)

/**Upgrade image area*/
#define CODE_UPD_START_ADDR (CODE_RUN_START_ADDR + CODE_RUN_AREA_LEN)
#define CODE_UPD_AREA_LEN (384*1024)

/**Area can be used by User*/
#define USER_ADDR_START (CODE_UPD_START_ADDR + CODE_UPD_AREA_LEN)
#define TLS_FLASH_PARAM_DEFAULT (USER_ADDR_START)
#define USER_AREA_LEN (CODE_UPD_HEADER_ADDR - USER_ADDR_START)

```

- 6) 依据确定的升级区间的起始位置调整 IMAGE 生成的参数 (红色字体)

wm\_gzip.exe "..\Bin\WM\_W600.bin"

makeimg.exe "..\Bin\WM\_W600.bin.gz" "..\Bin\WM\_W600\_GZ.img" 0 1 "..\Bin\version.txt" 90000 10100 "..\Bin\WM\_W600.bin"

makeimg.exe "..\Bin\WM\_W600.bin" "..\Bin\WM\_W600\_SEC.img" 0 0 "..\Bin\version.txt" 90000 10100

makeimg.exe "..\Bin\WM\_W600.bin" "..\Bin\WM\_W600.img" 0 0 "..\Bin\version.txt" 90000 10100

makeimg\_all.exe "..\Bin\secboot.img" "..\Bin\WM\_W600.img" "..\Bin\WM\_W600.FLS"

### 6.2.2.2 举例

如果用户编译的 IMAGE 大小为

WM\_W600\_SEC.img: 311Kbyte

WM\_W600\_GZ.img: 222Kbyte

把 IMAGE 的大小向上取 64Kbyte 的整数倍 (重要), 则

运行区空间: 320Kbyte

升级区空间: 256Kbyte

配置步骤如下:

1) 用户的新空间如图黄色部分

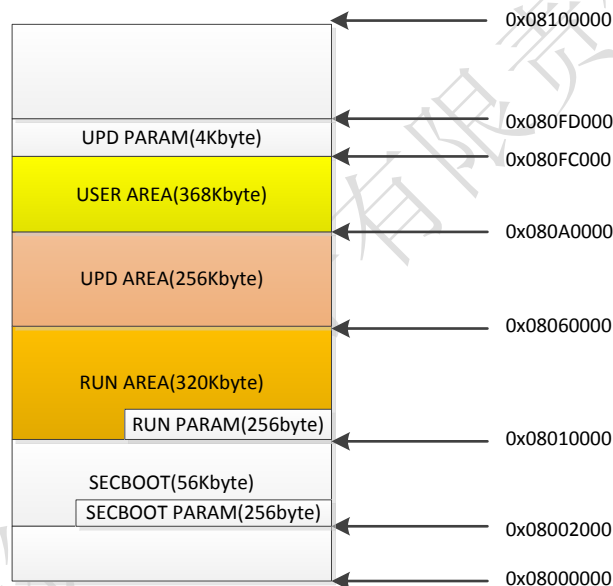


图 6-3

2) 新的代码空间调整为:

```
#define SECBOOT_HEADER_ADDR (FLASH_BASE_ADDR + FLASH_KEY_PARAM_AREA_LEN)
#define SECBOOT_HEADER_AREA_LEN (0x100)

#define SECBOOT_AREA_ADDR (SECBOOT_HEADER_ADDR + SECBOOT_HEADER_AREA_LEN)
#define SECBOOT_AREA_LEN (0x10000 - FLASH_KEY_PARAM_AREA_LEN - SECBOOT_HEADER_AREA_LEN)

/**Run-time image header area*/
#define CODE_RUN_HEADER_ADDR (SECBOOT_AREA_ADDR + SECBOOT_AREA_LEN)
#define CODE_RUN_HEADER_AREA_LEN (0x100)

/**Run-time image area*/
#define CODE_RUN_START_ADDR (CODE_RUN_HEADER_ADDR + CODE_RUN_HEADER_AREA_LEN)
#define CODE_RUN_AREA_LEN (320*1024 - CODE_RUN_HEADER_AREA_LEN)

/**Upgrade image area*/
#define CODE_UPD_START_ADDR (CODE_RUN_START_ADDR + CODE_RUN_AREA_LEN)
#define CODE_UPD_AREA_LEN (256*1024)

/**Area can be used by User*/
#define USER_ADDR_START (CODE_UPD_START_ADDR + CODE_UPD_AREA_LEN)
#define TLS_FLASH_PARAM_DEFAULT (USER_ADDR_START)
#define USER_AREA_LEN (CODE_UPD_HEADER_ADDR - USER_ADDR_START)
```

3) IMAGE 生成修改为:

```
wm_gzip.exe "..\Bin\WM_W600.bin"
```

```
makeimg.exe "..\Bin\WM_W600.bin.gz" "..\Bin\WM_W600_GZ.img" 0 1 "..\Bin\version.txt" 60000 10100 "..\Bin\WM_W600.bin"
```

```
makeimg.exe "..\Bin\WM_W600.bin" "..\Bin\WM_W600_SEC.img" 0 0 "..\Bin\version.txt" 60000 10100
```

```
makeimg.exe "..\Bin\WM_W600.bin" "..\Bin\WM_W600.img" 0 0 "..\Bin\version.txt" 60000 10100
```

```
makeimg_all.exe "..\Bin\secboot.img" "..\Bin\WM_W600.img" "..\Bin\WM_W600.FLS"
```

- 4) 重新编译烧录 WM\_W600\_GZ.img 文件，模块启动后，用户参数区即变为新的设定值。

### 6.2.3 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。