

WM_W800_固件生成说明

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	更新运行区大小, 与 SDK 保持一致	Cuiyc	
V1.2	2021/5/13	增加说明:建议串口方式下载统一	Cuiyc	
		烧录 w800.fls 文件		





目录

文林	当修改	记录	2
目表	₹		4
1	引言		6
	1.1	编写目的	6
	1.2	预期读者	6
	1.3	术语定义	6
	1.4	参考资料	6
2	IMA	GE 在 QFLASH 的位置	7
	2.1	SECBOOT 参数区域	7
	2.2	SECBOOT 存放区	7
	2.3	运行 IMG 参数区	7
	2.4	运行 IMG 存放区	7
	2.5	升级 IMG 存放区	8
	2.6	升级 IMG 参数区	8
3	W80	0 的 Image 组成说明	8
	3.1	Image Header	8
		3.1.1 Image Header 各字段描述	9
		3.1.2 Image Attribute	9
	3.2	Image Body1	0
	3.3	数字签名1	1
4	IMA	GE 类型1	1

₩ Winner Micro 联盛德微电子

	4.1	SECBOOT(非压缩格式)11
	4.2	User image11
5	生产烧录	e Image(组合 Image)12
6	不同阶段	とIMAGE 文件升级13
7	IMAGE 3	文件生成
	7.1	IMAGE 的加密及签名过程(可选)14
	7.2	IMAGE 压缩(可选)14
	7.3	IMAGE 生成
	7.4	IMAGE 签名(可选)
8	FAQ	
	8.1	W800 的 IMAGE 固件空间可以调整吗?15
	8.2	首次使用 W800 模块,用户应该烧录哪个文件?15
	8.3	W800 模块没有任何响应怎么办?16
	8.4	如何烧录 W800 的工厂烧录文件?16
	8.5	W800 的用户运行区固件大小有限制吗?16
	8.6	W800 的 IMAGE 区域调整,需要做哪些工作?16



1 引言

1.1 编写目的

本文档主要用于阐述 W800 中的固件格式,存储位置及文件生成。

1.2 预期读者

该文档适用的读者包括 W800 SDK 研发人员, W800 SDK 工程开发人员等。

1.3术语定义

序号	术语/缩略语	说明/定义
1	ΟΤΑ	Over-The-Air
2	QFLASH	Quad-SPI FLASH
3	IMG	IMAGE
4	UPD	Upgrade
5	SECBOOT	Second Boot, relative to ROM
6	ROM	Read-Only Memory

1.4参考资料

无



2 IMAGE 在 QFLASH 的位置

OTA Param (4KB)	
System Param (12KB)	
User Area (>=112KB,When X>=1)	0X8AFC000
Image Run area (1087KB)可变	0x81E0000
Run Image Header(1KB)可变	-0x80D0400
Image OTA Area (768KB)	
SecBoot Image Area (55KB)	0x8010000
SecBoot Image Header (1KB) RF data (8KB)	0x8002400 0x8002000 0x8000000

图 2-1

2.1 SECBOOT 参数区域

地址空间: 0x8002000-0x80023FF, 共1KB

参数布局: 详见《WM_W800_QFLASH 布局说明》

2.2 SECBOOT 存放区

地址空间: 0x8002400-0x800FFFF, 共 55KB

2.3运行 IMG 参数区

地址空间: 0x80D0000-0x80D03FF, 共1KB

参数布局: 详见《WM_W800_QFLASH 布局说明》

2.4运行 IMG 存放区

地址空间: 0x80D0400-0x801DFFFF, 共 1087KB



2.5升级 IMG 存放区

地址空间: 0x8010000-0x80CFFFF, 共 768KB

2.6升级 IMG 参数区

地址空间: 0x8XFF000-0x8XFFFFF, 共 4KB

3 W800的 Image 组成说明

Image 由 Header, Body 和数字签名三部分组成(如图)。

Image Header	
Image Body	
Signature	
	•

3.1 Image Header

W800 Image Header 包含信息:魔术字,Image 属性,Image 启动地址,Image 长度,Image Header 头位置,Image 升级地址,Image CRC 校验,Image 解密信息,数 字签名,压缩信息。

	typedef struct IMAGE_H	EADER_PARAM{	F	typedef union {
	unsigned int	<pre>magic_no;</pre>		<pre>struct {</pre>
	Img_Attr_Type	<pre>img_attr;</pre>	►	<pre>uint32_t img_type: 4;</pre>
Ť	unsigned int	<pre>img_addr;</pre>		<pre>uint32_t code_encrypt:1;</pre>
	unsigned int	<pre>img_len;</pre>		<pre>uint32_t prikey_sel:3;</pre>
	unsigned int	<pre>img_header_addr;</pre>		<pre>uint32_t signature:1;</pre>
	unsigned int	upgrade_img_addr;		<pre>uint32_t _reserved1: 7;</pre>
header 🔶	unsigned int	org_checksum;		<pre>uint32_t zip_type: 1;</pre>
•	unsigned int	upd_no;		<pre>uint32_t psram_io: 1;</pre>
ımg	unsigned char	ver[16];		<pre>uint32_t erase_block_en: 1;</pre>
signature	unsigned int	_reserved0;		<pre>uint32_t erase_always: 1;</pre>
	unsigned int	<pre>_reserved1;</pre>		<pre>uint32_t _reserved2: 12;</pre>
	unsigned int	next;		} b;
	unsigned int	hd_checksum;		uint32_t w;
	<pre>}IMAGE_HEADER_PARAM_ST</pre>	;		} Img_Attr_Type

3.1.1 Image Header 各字段描述

字段	描述
magic_no	魔术字,固定值 0xA0FFF9F
img_attr	Img_Attr_Type, IMAGE Attribute
img_addr	Image area 在 flash 中的运行位置
img_len	Image area 的字节数长度
img_header_addr	IMAGE header 在 flash 中的位置
upgrade_img_addr	升级区地址,升级 IMAGE header 在 flash 中存放位置
org_checksum	Image body 的 crc32 结果
	升级版本号,值较大的表示版本较新;
upa_no	当版本号为 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
ver	Image 版本号,字符串
next	下一个 image header 在 flash 中的位置(可选)
hd_checksum	Image header 的以上字段的 crc32 的值

3.1.2 Image Attribute

字段	Bit	描述
	4	0x0: SECBOOT;
img_type		0x1:User Image
		OxE:ft 测试程序;
		其它值:用户自定义
code_encrypt	1	0:固件明文存储;

₩ Winner Micro 联盛德微电子

		1: 固件由客户加密后存储	
	3	芯片内置 8 组 RSA 私钥用于解密固件加密的秘钥,	
pricey_set		用户可任选一组使用,取值范围 0~7	
cianaturo	1	0:IMAGE 不包含签名部分;	
signature		1: IMAGE 包含 128bytes 签名	
	1	0:不压缩;	
zip_type		1:image area 部分为压缩档(<mark>当前仅支持 GZIP</mark>)	
reserved	1	保留	
orace block on	1	0:不支持 64KB Block 擦除;	
erase_block_en		1: 支持 Block 擦除	
		0:Sector 或 Block 擦除前检查 flash 是否全 F,	
erase_always	1	全 F 的 Sector 或 Block 不进行擦除操作;	
		1: 总是先擦后写	

3.2 Image Body

加密	压缩	Image Body 内容	用途
X	Х	原始 Image 内容	SECBOOT,User Image
х	V	原始 Image 压缩后的内容	User Image
\checkmark	Х	原始 Image 加密的内容+加密信息	SECBOOT
\checkmark	\checkmark	原始 Image 加密后压缩的内容+加密信息	User Image



3.3数字签名

如果 Image Header 的属性里的 signature 被置位了,则说明固件带有数字签名。

数字签名是针对 Image Header 和 Image Body 组成的文件。

4 IMAGE 类型

依据 Img_Attr_Type 可以区分出来不同的 Image,常用的有如下两种

4.1 SECBOOT (非压缩格式)

W800 的二级引导程序

SECBOOT Header
Next Image Header Position (optional)
Secboot Image Body
Other(optional)
Signature (header +image + other) (optional)

图 4-1

4.2 User image

用户运行区的固件,可以为压缩的或者非压缩的,压缩的采用 G-ZIP 实现。 非压缩格式:

User Image Header Next Image Header Position (optional)					
User Image Body					
Other(optional)					
Signature (header +image + other) (optional)					



图 4-2

压缩格式:



图 4-3

5 生产烧录 Image (组合 Image)

W800 生产烧录固件是把 SECBOOT 和 User.img 用工具拼接起来通过 xmodem 升级,

如下。



SECBOOT Header	
Secboot Image	
Other(optional)	
Secboot Signature (header +image + other) (optional)	
Run Image Header	
Run Area Image	
Other(optional)	
Run Image Signature (header +image + other) (optional)	
Other File Header	
File Content (optional)	
Other(optional)	
Signature (header +File + other) (optional)	

图 5-1

W800的 ROM 会依据 Header 来区分当前烧录的位置。

6 不同阶段 IMAGE 文件升级

IMAGE 类型	是否支持	是否支持	是否支持
	ROM 升级	SECBOOT 升级	OTA 升级
User Run Area	\checkmark	\checkmark	х
image			
User OTA Image	х	Х	\checkmark
W800_SECBOOT.img		Х	1
生产烧录 Image	\checkmark	Х	Х



7 IMAGE 文件生成

设定:

原始 Image 文件: w800_original.img 加密文件为: w800_original_enc.img Image 加密 Key: X,存为文件为 keyfile keyfile 的公钥加密文件为: keyencfile 公钥证书文件: capub.pem 公钥文件的 N 记为: capbu_N.dat 签名前的 Image 文件: Image_nosig.img 签名后的 Image 文件: Image_sig.img

7.1 IMAGE 的加密及签名过程(可选)

通过 openssl enc -aes-128-ecb 对 w800_original.img 进行加密(KEY 由用户自定

义), 生成原始 IMAGE 的加密文件 w800_original_enc.img。

通过 openssl rsautl -encrypt, 用公钥证书 capub.pem 对 keyfile 进行加密生成 keyencfile

把 keyencfile 追加到文件 w800_original_enc.img 的后面,生成临时文件 Temp,再把 capbu_N.dat 文件追加到 Temp 后面,生成 Image 的 Body。

7.2 IMAGE 压缩(可选)

对目标 Image(已包含了完整的 Image header+Image 内容+可选的签名+可选的加密 信息) 进行压缩, 仅支持 GZIP 压缩算法。



7.3 IMAGE 生成

Image Body 生成后, 接下来需要增加 Image Header。

使用 wm_tool 工具生成最终签名前的 Image 文件 Image_nosig.img。

7.4 IMAGE 签名(可选)

对 Image_nosig.img 做数字签名得到签名文件 sign.dat ,把 sign.dat 追加到

Image_nosig.img 后,生成最终的签名文件 Image_sig.img。

上述的整个 IMAGE 生成过程,可以参考 W800 SDK 中的

SDK/tools/w800/utilities/aft_build_project.sh

或者

rules.mk

$8 \,$ Faq

8.1 W800 的 IMAGE 固件空间可以调整吗?

可以调整,依照 QFLASH 的布局图,按照自己的需求调整即可。

8.2 首次使用 W800 模块, 用户应该烧录哪个文件?

分几种情况:

1) W800 模块仅有 ROM 固件(串口方式下载,建议都使用此固件)

烧录一个 SECBOOT 和 User Image 打包在一起的固件,通过 ROM 的串口 0 烧录。

2) W800 仅有 SECBOOT 固件

烧录 User Image



3) W800 模块有可用的用户固件

可以根据需要使用自己指定的固件

8.3 W800 模块没有任何响应怎么办?

如果 W800 模块连接上 UART0 后, 既没有进入 ROM, 也没有进入 SECBOOT 和用户固

- 件,则需要对其进行恢复操作。
 - 1) 如果模块的 BOOTMODE 脚拉低,复位模块可以进入 ROM,则按照 7.2 的方法 1) 操作即可。
 - 2) 如果 BOOTMODE 脚拉低也没用,则可以考虑硬件问题了。
 - 3) 可以考虑是否串口接反了

8.4 如何烧录 W800 的工厂烧录文件?

W800 的工厂烧录文件是一个 SECBOOT 和 User Image 文件链接在一起的文件。

工厂烧录的步骤:

- 1) BOOTMODE 脚拉低
- 2)复位芯片
- 3) 通过 UARTO 2M 方式升级

8.5 w800 的用户运行区固件大小有限制吗?

有限制的, 取决于 QFLASH 的大小, 用户的参数区大小以及用户是否需要 OTA 功能。

8.6 W800 的 IMAGE 区域调整, 需要做哪些工作?

详见:《WM_W800_参数区使用说明》的用户参数区调整规则。