

WM_W800_Firmware Generation Instructions

V1.2

Beijing Lianshengde Microelectronics Co., Ltd. (winner micro)

Address: Floor 18, Yindu Building, No. 67, Fucheng Road, Haidian District, Beijing

Tel: +86-10-62161900

Company website: www.winnermicro.com

Document modification record

Version	revision time	revision history	author	review
V0.1	2019/9/25	[C] Create document	Cuiyc	
V0.2	2020/7/8	Unified font	Cuiyc	
V1.0	2020/8/10	Upgrade version number	Cuiyc	
V1.1	2021/2/23	Update the size of the operating area to be consistent with the SDK	Cuiyc	
V1.2	2021/5/13	Add description: It is recommended to download in a unified way through the serial port Burn w800.fls file	Cuiyc	

WinnerMicro

Table of contents

Documentation Modification History	2
Table of contents.....	4
1 Introduction	6
1.1 Purpose of writing.....	6
1.2 Intended Reader	6
1.3 Definition of Terms.....	6
1.4 References.....	6
2 The position of IMAGE in QFLASH.....	7
2.1 SECBOOT parameter area.....	7
2.2 SECBOOT storage area.....	7
2.3 Run the IMG parameter area.....	7
2.4 Running the IMG repository.....	7
2.5 Upgrading the IMG storage area.....	8
2.6 Upgrade IMG parameter area.....	8
3 Image Composition Description of W800.....	8
3.1 Image Header.....	8
3.1.1 Description of each field of Image Header.....	9
3.1.2 Image Attribute	9
3.2 Image Body	10
3.3 digital signature.....	11
4 IMAGE type.....	11

4.1	SECBOOT (uncompressed format)	11
4.2	User image	11
5	Produce and burn Image (Combined Image).....	12
6	IMAGE file upgrade at different stages.....	13
7	IMAGE file generation.....	14
7.1	Encryption and signing process of IMAGE (optional).....	14
7.2	IMAGE COMPRESSION (OPTIONAL)	14
7.3	IMAGE generation	15
7.4	IMAGE signature (optional)	15
8	FAQ	15
8.1	Can IMAGE firmware space of W800 be adjusted?	15
8.2	When using the W800 module for the first time, which file should the user burn?	15
8.3	What should I do if the W800 module does not respond?	16
8.4	How to burn W800 factory burn files?	16
8.5	Is there a limit to the size of the firmware in the user runtime area of the W800?	16
8.6	W800 IMAGE area adjustment, what work needs to be done?	16

1 Introduction

1.1 Purpose of writing

This document is mainly used to explain the firmware format, storage location and file generation in W800.

1.2 Intended readers

The applicable readers of this document include W800 SDK developers, W800 SDK engineering developers, etc.

1.3 Definition of terms

Ordinal	term/abbreviation	Description/Definition
1	ORDER	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 References

none

2 The position of IMAGE in QFLASH

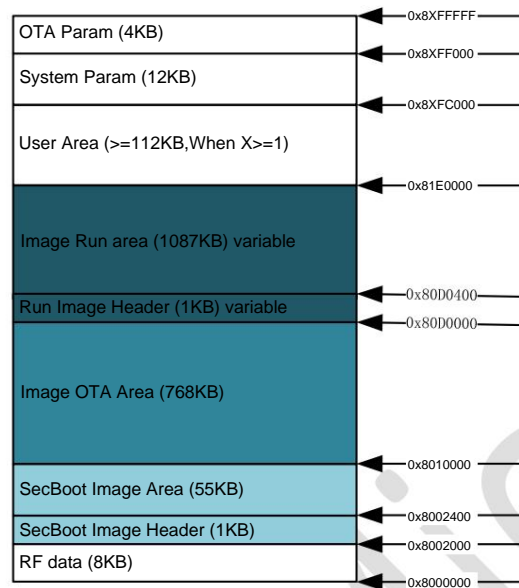


Figure 2-1

2.1 SECBOOT parameter area

Address space: 0x8002000-0x80023FF, 1KB in total

Parameter layout: see "WM_W800_QFLASH Layout Instructions" for details

2.2 SECBOOT storage area

Address space: 0x8002400-0x800FFFF, 55KB in total

2.3 Run the IMG parameter area

Address space: 0x80D0000-0x80D03FF, 1KB in total

Parameter layout: see "WM_W800_QFLASH Layout Instructions" for details

2.4 Run the IMG storage area

Address space: 0x80D0400-0x801DFFFF, a total of 1087KB

2.5 Upgrade the IMG storage area

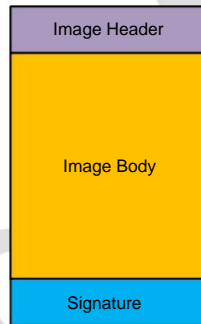
Address space: 0x8010000-0x80CFFFF, a total of 768KB

2.6 Upgrade IMG parameter area

Address space: 0x8XFF000-0x8XFFFFF, 4KB in total

3 W800 Image Composition Description

Image consists of three parts: Header, Body and digital signature (as shown in the figure).

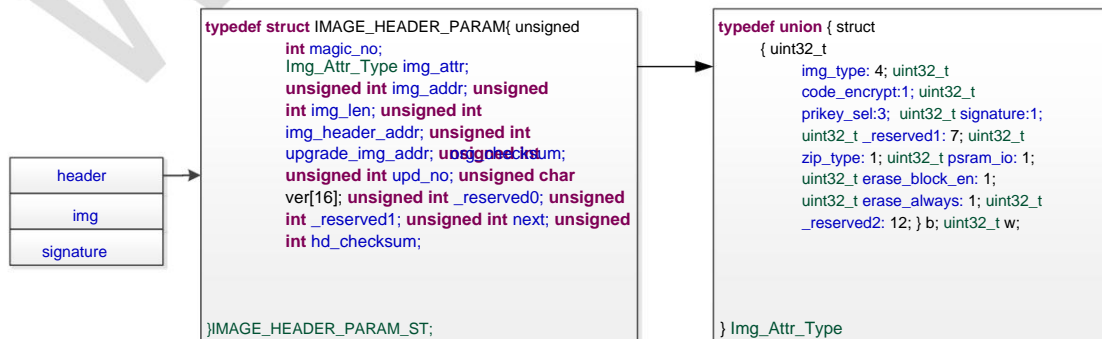


3.1 Image Header

W800 Image Header contains information: magic word, Image attribute, Image start address, Image length

degree, Image Header header position, Image upgrade address, Image CRC check, Image decryption information, data

Word signature, compressed information.



3.1.1 Description of each field of Image Header

field	describe
magic_no	magic word, fixed value 0xA0FFF9F
img_attr	Img_Attr_Type IMAGE Attribute
img_addr	The running position of Image area in flash
img_len	The length in bytes of the Image area
img_header_addr	IMAGE header position in flash
upgrade_img_addr	upgrade area address, upgrade IMAGE header storage location in flash
	The crc32 result of org_checksum Image body
upd_no	Upgrade version number, a larger value indicates a newer version; When the version number is 0xFFFFFFFF, any version number firmware can be upgraded
to see	Image version number, string
next	The position of the next image header in flash (optional)
	The crc32 value of the above field of hd_checksum Image header

3.1.2 Image Attribute

field	Bit	describe
img_type	4	0x0 SECBOOT 0x1 User Image 0xE: ft test program; Other values: user defined
code_encrypt	1	0: firmware plaintext storage;

		1: The firmware is encrypted and stored by the customer
pricey_sel	3	The chip has 8 sets of RSA private keys built-in to decrypt the secret key encrypted by the firmware. The user can choose a group to use, and the value range is 0~7
signature	1	0: IMAGE does not contain the signature part; 1: IMAGE contains 128bytes signature
zip_type	1	0: no compression; 1: The image area part is a compressed file (currently only supports GZIP)
reserved	1 reserved	
erase_block_in 1		0: Does not support 64KB Block erasure; 1: Support block erase
erase_always	1	0: Check whether the flash is all F before sector or block erase, Sector or Block with all F will not be erased; 1: Always erase before writing

3.2 Image Body

encrypted	compression	Image Body content	use
X	X	Original Image content	SECBOOTyUser Image
X	ÿ	The compressed content of the original image	User Image
ÿ	X	Original Image encrypted content + encrypted information	SECBOOT
ÿ	ÿ	Compressed content of the original image after encryption + encrypted information	User Image

3.3 Digital signature

If the signature in the Image Header attribute is set, it means that the firmware has a digital signature.

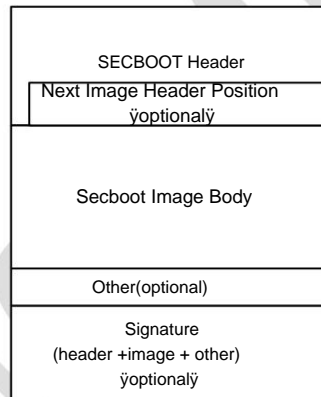
The digital signature is for the file composed of Image Header and Image Body.

4 IMAGE type

According to `Img_Attr_Type`, different images can be distinguished. There are two commonly used ones:

4.1 SECBOOT (uncompressed format)

Secondary Bootloader for W800



Pic 4-1

4.2 User image

The firmware in the user running area can be compressed or uncompressed, and the compressed one is realized by G-ZIP.

Uncompressed format:

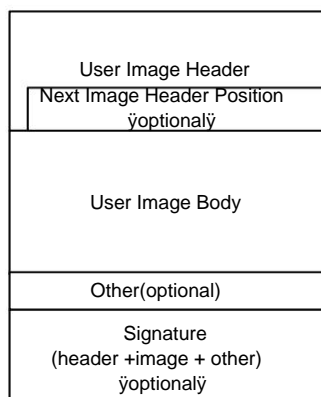


Figure 4-2

Compression format:

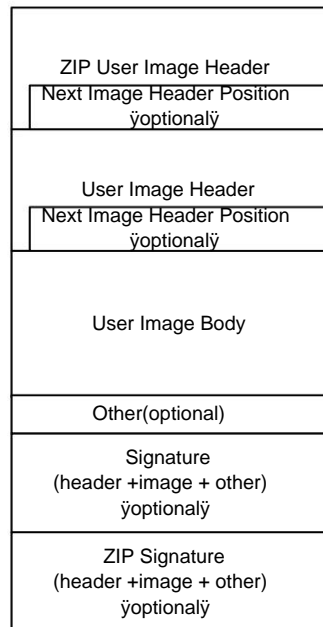


Figure 4-3

5 Produce and burn Image (combined Image)

W800 production burning firmware is to splice SECBOOT and User.img together with tools and upgrade through xmodem,

as follows.

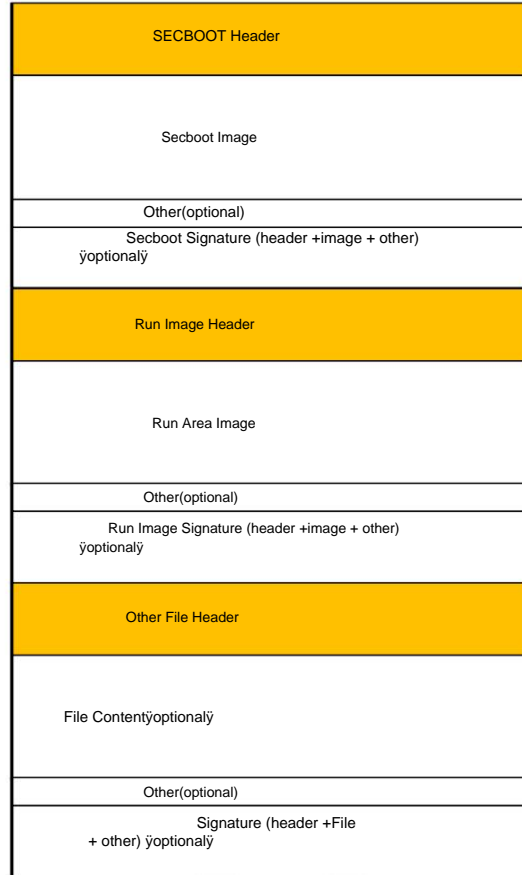


Figure 5-1

The ROM of W800 will distinguish the current burning position according to the Header.

6 IMAGE file upgrades in different stages

IMAGE type	Whether to support	Whether to support	Whether to support
	ROM upgrade	SECBOOT upgrade	OTA upgrade
User Run Area image	ÿ	ÿ	X
User OTA Image	X	X	ÿ
W800_SECBOOT.img	ÿ	X	ÿ
Produce and burn Image	ÿ	X	X

7 IMAGE file generation

set up:

Original Image file: w800_original.img

The encrypted file is: w800_original_enc.img

Image encryption Key: X, saved as a file as keyfile

The public key encryption file of keyfile is: keyencfile

Public key certificate file: capub.pem

The N of the public key file is recorded as: capbu_N.dat

Image file before signature: Image_nosig.img

Signed Image file: Image_sig.img

Signature file: sign.dat

7.1 IMAGE encryption and signature process (optional)

Encrypt w800_original.img by openssl enc -aes-128-ecb (KEY is defined by the user

definition), generate the encrypted file w800_original_enc.img of the original IMAGE.

Use openssl rsautl -encrypt to encrypt the keyfile with the public key certificate capub.pem

keyencfile

Append the keyencfile to the back of the file w800_original_enc.img, generate a temporary file Temp, and then

Append the capbu_N.dat file to Temp to generate Image Body.

7.2 IMAGE compression (optional)

For the target Image (which already contains the complete Image header + Image content + optional signature + optional encryption

information) for compression, only the GZIP compression algorithm is supported.

7.3 IMAGE generation

After the Image Body is generated, the Image Header needs to be added next.

Use the `wm_tool` tool to generate the Image file `Image_nosig.img` before the final signature.

7.4 IMAGE signature (optional)

Digitally sign `Image_nosig.img` to get the signature file `sign.dat`, and append `sign.dat` to

After `Image_nosig.img`, generate the final signature file `Image_sig.img`.

For the above-mentioned whole IMAGE generation process, you can refer to the

`SDK/tools/w800/utilities/aft_build_project.sh`

or

`rules.mk`

8 FAQ

8.1 Can the IMAGE firmware space of W800 be adjusted?

It can be adjusted, according to the layout of QFLASH, you can adjust it according to your own needs.

8.2 When using the W800 module for the first time, which file should the user burn?

There are several situations:

- 1) W800 module only has ROM **firmware** (download via serial port, it is recommended to use this firmware)

Burn a firmware packaged with SECBOOT and User Image, and burn through serial port 0 of ROM.

- 2) W800 only has SECBOOT firmware

Burn User Image

3) W800 module has user firmware available

You can use your own specified firmware as needed

8.3 What should I do if the W800 module does not respond?

If the W800 module is connected to UART0, it neither enters ROM, nor enters SECBOOT and user firmware.

file, you need to restore it.

1) If the BOOTMODE pin of the module is pulled low, the reset module can enter the ROM, then follow the method 1 of 7.2)

Just operate.

2) If it is useless to pull the BOOTMODE pin low, you can consider hardware problems.

3) You can consider whether the serial port is reversed

8.4 How to burn W800 factory burn files?

The factory programming file of W800 is a file linking SECBOOT and User Image files together.

Steps for factory programming:

1) BOOTMODE pin is pulled low

2) Reset chip

3) Upgrade via UART0 2M

8.5 Is there a limit to the size of the firmware in the user runtime area of the W800?

There are restrictions, depending on the size of QFLASH, the size of the user's parameter area and whether the user needs the OTA function.

8.6 What needs to be done to adjust the IMAGE area of W800?

For details, see: "WM_W800_Parameter Area Instructions" for user parameter area adjustment rules.