

WM_W60X_2M_Flash 参数区使用说明

V1.0

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
1.0	2019/03/01	[C]创建文档 注：用户要调节参数区，需关注章节 6.2	Cuiych	

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	2M FLASH 参数区布局	4
2.1	物理层参数区	4
2.2	QFLASH 参数区域	5
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	15

1 引言

1.1 编写目的

本文档主要用于阐述 W60X 芯片 2M FLASH 参数区布局, 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	SECBOOT	Second Boot, relative to ROM
2	ROM	Read-Only Memory

1.4 参考资料

无

2 2M FLASH 参数区布局

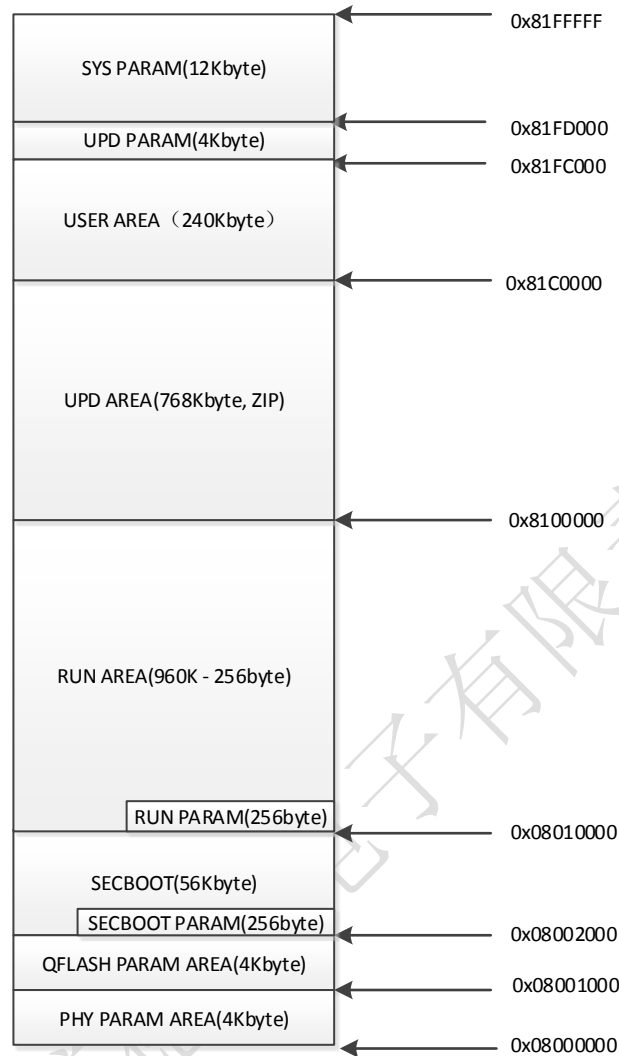


图 2-1

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:

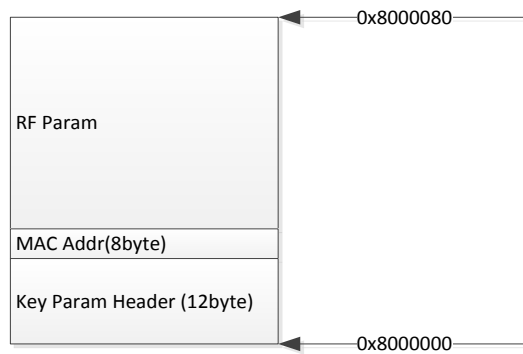


图 2-2

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:

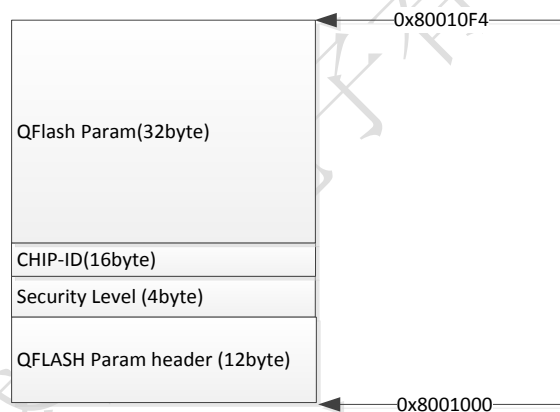


图 2-3

2.3 用户参数区

地址空间: 0x81C0000-0x81FBFFF, 共 240Kbyte

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: 0x81FD000-0x81FFFFFF, 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: 0x81FD000-0x81FDFFF
- 2) 系统参数 2 区: 0x81FE000-0x81FEFFF
- 3) 系统参数 3 区: 0x81FF000-0x81FFFFF

3 物理层参数区

3.1 物理层参数介绍

W60X 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

W60X 芯片或者模块生产时写入

3.3 物理层参数的使用

W60X 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

W60X 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

W60X 芯片生产阶段。

4.2.1 QFLASH 参数区使用

W60X 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 W60X 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。

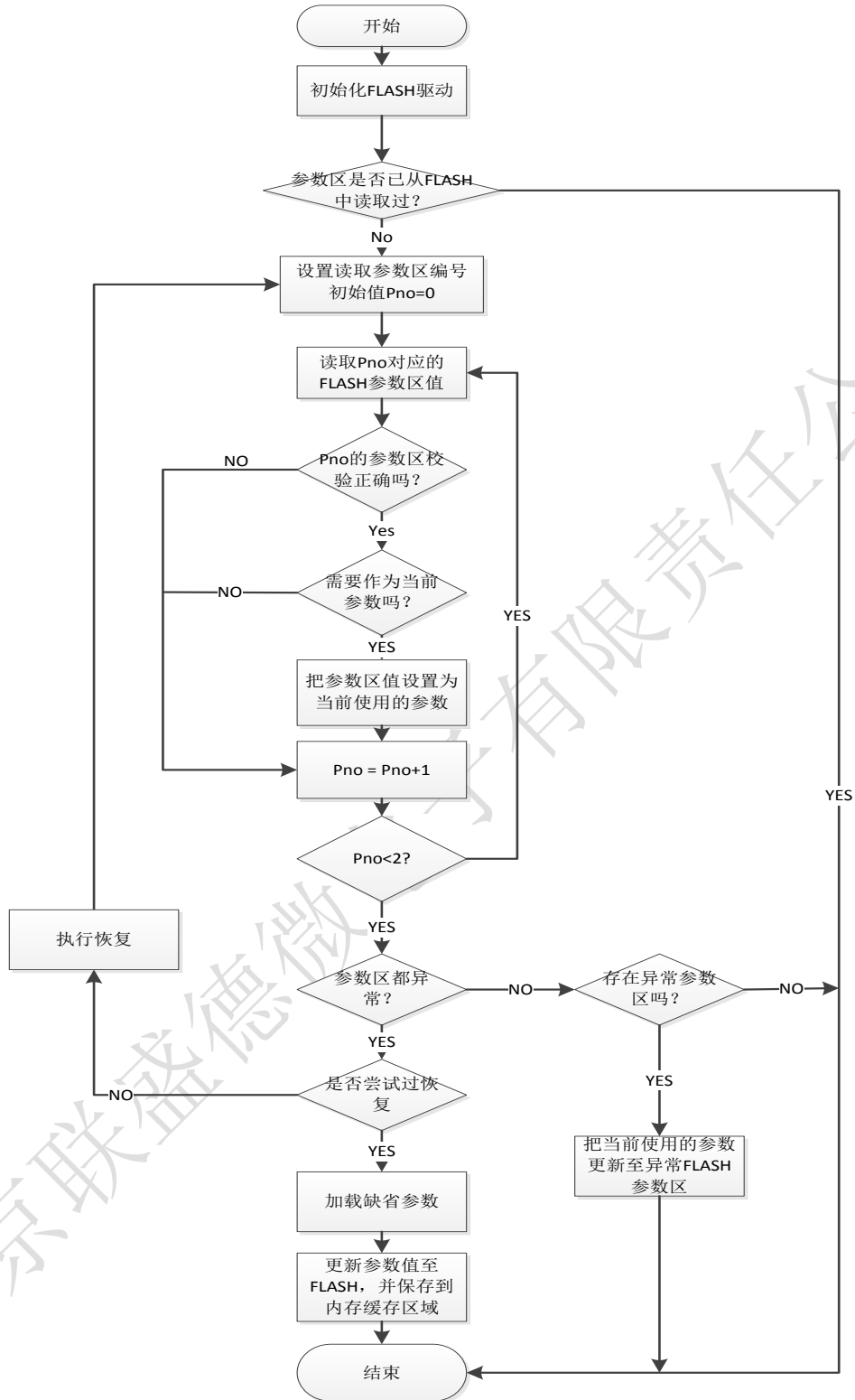


图 5-1

5.2.2 参数使用阶段

1) 参数获取

系统参数区除了存放于 FLASH 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 FLASH。

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

北京联盛德微电子有限责任公司

6 用户参数区

6.1 用户参数

W60X 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

W60X 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

W60X 的默认 FLASH 的布局所能提供给用户的区域为 240Kbyte。但是，当前的 W60X 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

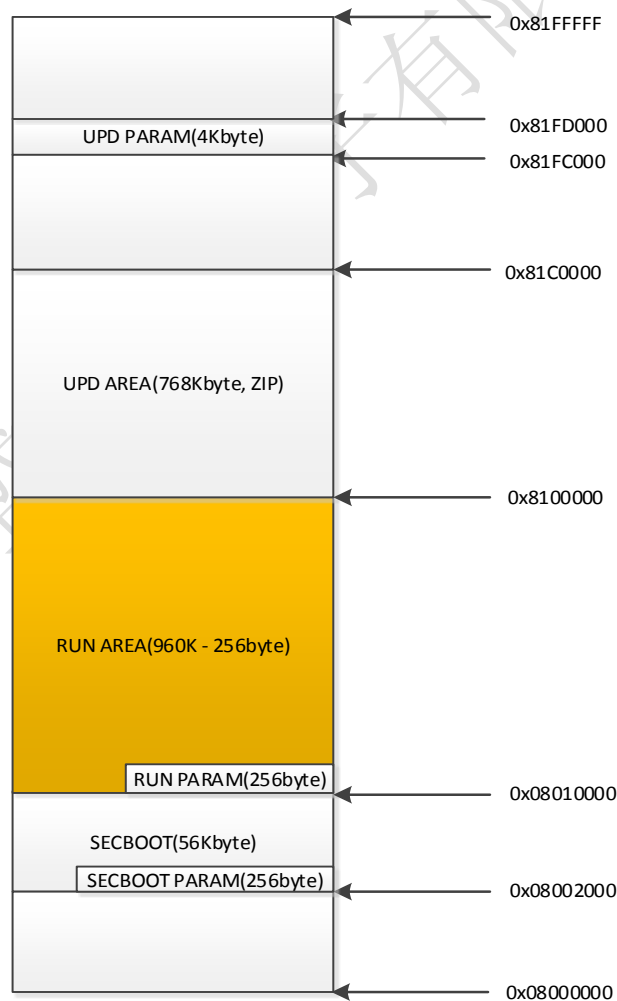


图 6-1

2) 依据用户编译的 WM_W600_GZ.img 确定的所用升级区空间

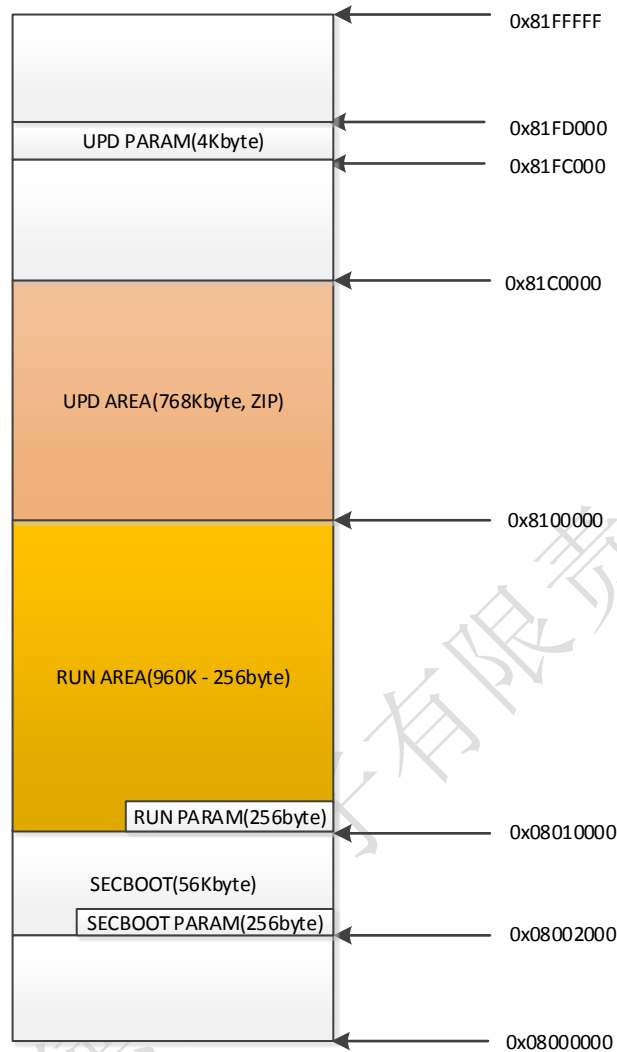


图 6-2

- 3) 依据 WM_W600_SEC.img 和 WM_W600_GZ.img 的大小按照 FLASH 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 W60X SDK 的宏定义确定新的用户空间起始地址
查找 SDK 中函数 tls_flsh_layout_init, 当前布局如下图:

```

/**Run-time image area size*/
CODE_RUN_AREA_LEN = (960*1024 - 256);

/**Upgrade image area*/
CODE_UPD_START_ADDR = (CODE_RUN_START_ADDR + CODE_RUN_AREA_LEN);
CODE_UPD_AREA_LEN = (768*1024);

/**Area can be used by User*/
USER_ADDR_START = (CODE_UPD_START_ADDR + CODE_UPD_AREA_LEN);
TLS_FLASH_PARAM_DEFAULT = (USER_ADDR_START);
USER_AREA_LEN = (240*1024);
USER_ADDR_END = (USER_ADDR_START + USER_AREA_LEN - 1);
    
```

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" 3 1 "..\Bin\version.txt" 100000 10100 "..\Bin\WM_W600.bin"

makeimg.exe "..\Bin\WM_W600.bin" "..\Bin\WM_W600_SEC.img" 3 0 "..\Bin\version.txt" 100000 10100

makeimg.exe "..\Bin\WM_W600.bin" "..\Bin\WM_W600.img" 3 0 "..\Bin\version.txt" 100000 10100

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

注意：makeimg 和 makeimg_all 的使用位置

(1) Keil 的 Image 生成脚本位置：G3.02.00\Tools\Keil\W600Project\makeimg.bat

(2) GCC 的 Image 生成脚本位置：G3.02.00\Tools\GNU\createimg.sh

6.2.2.2 举例

如果用户编译的 IMAGE 大小为

WM_W600_SEC.img: 760Kbyte

WM_W600_GZ.img: 570Kbyte

把 IMAGE 的大小向上取 64Kbyte 的整数倍 (重要), 则

运行区空间: 768Kbyte

升级区空间: 576Kbyte

配置步骤如下:

1) 用户的新空间如图黄色部分

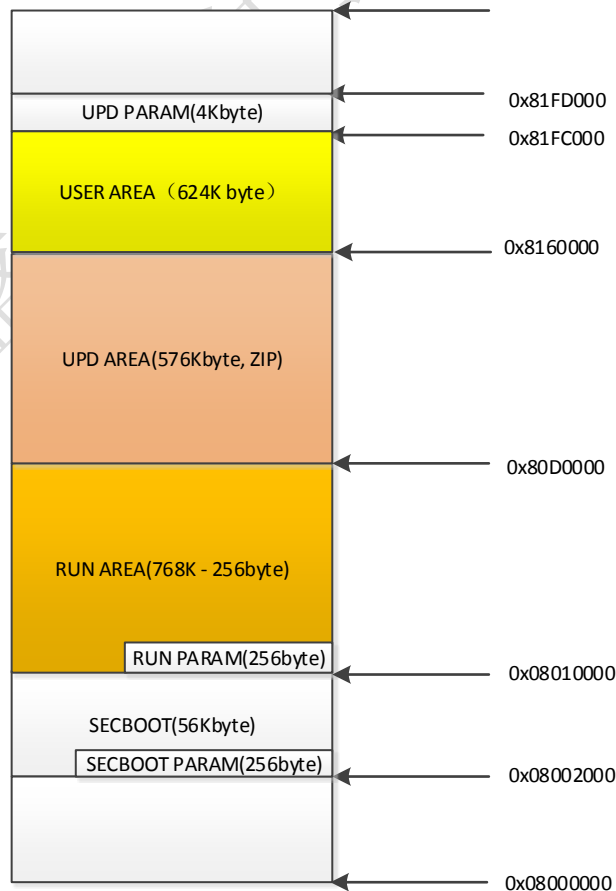


图 6-3

2) 新的代码空间调整为:

查找 SDK 中函数 `tls_fl_layout_init`, 针对 2M Flash 布局类型为 3 的参数值修改, 结果如下图:

```

/**Run-time image area size*/
CODE_RUN_AREA_LEN           = (768*1024 - 256);

/**Upgrade image area*/
CODE_UPD_START_ADDR         = (CODE_RUN_START_ADDR + CODE_RUN_AREA_LEN);
CODE_UPD_AREA_LEN           = (576*1024);

/**Area can be used by User*/
USER_ADDR_START              = (CODE_UPD_START_ADDR + CODE_UPD_AREA_LEN);
TLS_FLASH_PARAM_DEFAULT     = (USER_ADDR_START);
USER_AREA_LEN                 = (624*1024);
USER_ADDR_END                 = (USER_ADDR_START + USER_AREA_LEN - 1);

```

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" 3 1 "..\Bin\version.txt" D0000 10100 "..\Bin\WM_W600.bin"`

`makeimg.exe "..\Bin\WM_W600.bin" "..\Bin\WM_W600_SEC.img" 3 0 "..\Bin\version.txt" D0000 10100`

`makeimg.exe "..\Bin\WM_W600.bin" "..\Bin\WM_W600.img" 3 0 "..\Bin\version.txt" D0000 10100`

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

注意: `makeimg` 和 `makeimg_all` 的使用位置

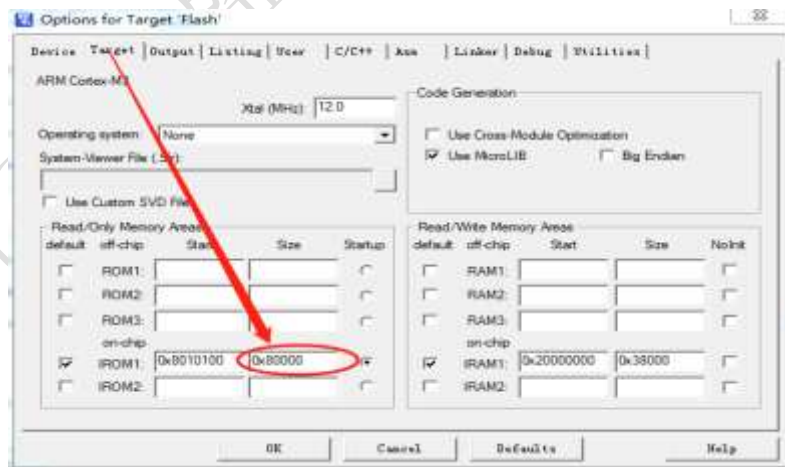
(1) Keil 的 Image 生成脚本位置: `G3.02.00\Tools\Keil\W600Project\makeimg.bat`

(2) GCC 的 Image 生成脚本位置: `G3.02.00\Tools\GNU\createimg.sh`

4) 如果 Code Size 超过设定值, 还需调整编译器的链接脚本

注意: Code Size 不能超过 Flash 的前 1M 空间范围

(1)Keil 编译工程, 调整 Size 如图



GCC 编译工程, 调整链接脚本 `link_w600.ld`, 如图

```
link_w600.ld
7 */
8 MEMORY
9 {
10 FLASH (rx) : ORIGIN = 0x8010100, LENGTH = 768K /* 768KB-256B */
11 RAM (rwx) : ORIGIN = 0x20000000, LENGTH = 0x38000 /* 224KB */
12 }
13
14 /* Linker script to place sections and symbol values. Should be used together
15 * with other linker script that defines memory regions FLASH and RAM.
16 * It references following symbols, which must be defined in code:
17 * Reset_Handler : Entry of reset handler
18 *
```

- 5) 重新编译 SDK 生成 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。