RTL8762C/ OTA User Manual

V1.01

2018/06/11

修订历史

Date	Version	Comments
2018/06/09	V1.0	Draft by Ken First release version
2018/06/11	V1.0.1	Modify Indications for image version.
2018/09/06	V1.0.2	Reviewed by Andy&Lory, modified by Ken.

目录

修订历史 2
1 概述 7
1.1 功能介绍 7
1.2 相关重点内容 7
2 Flash 布局介绍 7
3 Image Header 格式 9
4 OTA 打包格式及工具使用 13
4.1 支持 bank 切换 13
4.1.1 FLASH Layout 13
4.1.2 打包相关工具使用步骤---切换 BANK 14
4.2 不支持 bank 切换 17
4.2.1 FLASH Layout 17
4.2.2 打包相关工具使用步骤---不切换 BANK 18
5 OTA 协议 21
5.1 DFU Service 21
5.2 OTA Service 22
5.2.1 OTA CMD 22
5.2.2 Device Mac 23
5.2.3 Patch Version 23
5.2.4 APP Version 23
5.2.5 Patch Extension Version 24
5.2.6 Test Mode 24
5.2.7 Device Info 24
5.2.8 Image Counter 26
5.2.9 Image Version 26
5.3 OTA 流程 27
5.3.1 OTA 不带 buffer check 流程 27
5.3.2 OTA buffer check 流程 28

5.3.3 多文件更新 29
6 Master 软件的使用 30
7 参考文献 31

表目录

Table 2-1 FLASH 空间分布和功能说明 8
Table 2-2: Flash Segmentation 9
Table 3-3: Fields of OTA Header 11
Table 3-4: Image Header Field 11
Table 4-1： FLASH Layout Sample 13
Table 4-2： FLASH Layout Sample 18
Table 6-1 Dfu opcode 22
Table 6-2： Ota Characteristic 22
Table 6-3: OTA CMD characteristics 23
Table 6-4: Device Mac characteristics 23
Table 6-5: Patch Version characteristic for bee2(don’t recommend, described in image version) 23
Table 6-6: APP Version characteristic for bee2(don’t recommend, described in image version) 24
Table 6-7: Patch Extension Version characteristic 24
Table 6-8:Test Mode characteristics 24
Table 6-9: Device info characteristic for Bee2 24
Table 6-10: Device info Format For Bee2(OTA version = 1) 25
Table 6-11: Image Counter characteristics 26
Table 6-12: Image Counter characteristics 26

图目录

Figure 2-1 Flash 布局 7
Figure 2-2 OTA Bank 布局 8
Figure 3-1 OTA Header 结构 10
Figure 3-2 Image Header 结构 11
Figure 4-1 生成 Flash Layout 14
Figure 4-2 MP PACK Tool Load Flash Layout 15
Figure 4-3 生成 OTA Header 16
Figure 4-4 打包生成 PACK 17
Figure 4-5 生成 Flash Layout 19
Figure 4-6 MP PACK Tool Load Flash Layout 20

1 概述

1.1 功能介绍

本文中 OTA（Over The Air）是指通过蓝牙的传输，对 RTL8762C 系列的 FLASH 运行的 Image 和 data 进行空中升级的技术。

注: 本文同样适用于 RTL8752C 系列 IC。

1.2 相关重点内容

- Flash 布局介绍
- IMAGE 格式和布局
- OTA 打包格式及工具使用
- OTA 协议

2 Flash 布局介绍

RTL8762C 的 flash 布局如图 2-1 所示，是由“OEM Config”，“OTA Bank 0”，“OTA Bank 1”，“FLASH Transport Layer(FTL)”，“OTA TMP”以及 APP defined section 组成。访问 flash 的起始地址为 0x800000。

（注: 上述几个组成部分的定义见图 2-1）

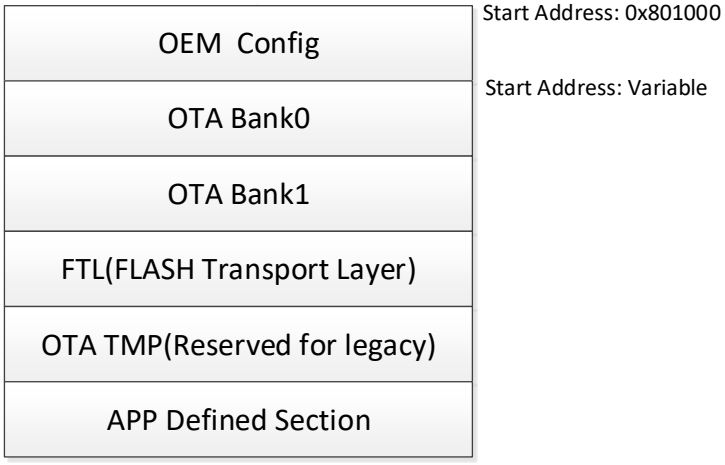


Figure 2-1 Flash 布局

Flash 中各个部分空间分布和功能说明如下表所示。

Memory Segment	Starting Address	Size (Bytes)	Functions
OEM Config	0x801000	0x1000	方案使用的 config 信息存储区域，包括蓝牙地址，AES Key，和客户可修改的 Flash 布局内容等
OTA Bank 0	可变	可变长度	如果不切换 bank，则为方案数据和代码运行区，包

	(OEM Config 里定义)	(OEM Config 里定义)	括 OTA Header，Secure boot，Patch，APP，Data1，Data2。此 OTA 备份区为 OTA_TMP。 如果切换 bank，OTA Bank0 和 OTA Bank1 互为备份区，假设 OTA Bank0 为运行区，则 OTA Bank1 为备份区.
OTA Bank 1	可变 (OEM Config 里定义)	可变长度 (OEM Config 里定义)	切换 Bank 的方式才存在，在切换 bank 的方式下功能和 Bank0 一样。 大小必须和 OTA Bank 0 一致。
FTL	可变 (OEM Config 里定义)	可变长度 (OEM Config 里定义)	以逻辑地址访问 flash 的软件技术，可以按照 word 进行任意地址读写，客户无需关注 flash 物理层操作;并且考虑到损耗均衡。
OTA_TMP	可变 (OEM Config 里定义)	可变长度 (OEM Config 里定义)	不切换 bank 方式下， 作为 OTA 备份区使用，大小必须不小于 OTA Bank0 最大的 image 大小。
APP Defined Section	可变 (OEM Config 里定义)	可变长度 (OEM Config 里定义)	Flash 剩余未划分区域， 客户可以自由使用。但不受本文涉及的 OTA 方案的管控。

Table 2-1 FLASH 空间分布和功能说明

OTA bank 内部的布局如下图所示，其中各个部分的说明如图 2-2 所示，布局中各段描述说明如表 2-2 所示。

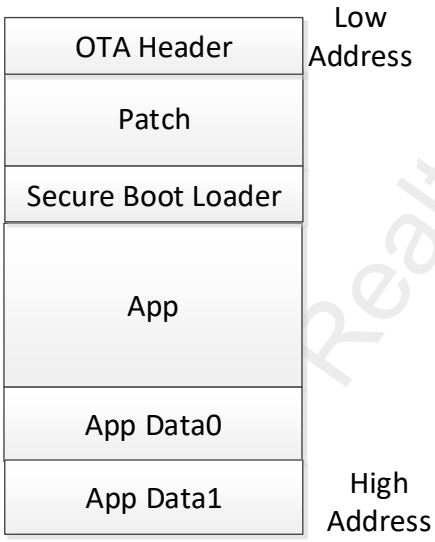


Figure 2-2 OTA Bank 布局

Memory Segment	Starting Address	Size	Functions
OTA Header	由 OEM Config 定义决定	4KB	存放 OTA Header 的版本，BANK 中存在 image 的起始地址和大小。
Secure Boot	由 OTA Header 定	Variable	启动过程中对代码安全级别检查的代码。

Loader	义决定		
Patch	由 OTA Header 定义决定	Variable	对 rom 中协议栈，系统的优化和扩展代码.
App	由 OTA Header 定义决定	Variable	开发方案的运行代码
App Data0	由 OTA Header 定义决定	Variable	开发方案中使用的数据区.
App Data1	由 OTA Header 定义决定	Variable	开发方案中使用的数据区.

Table 2-2: Flash Segmentation

3 Image Header 格式

OTA Header image 是由 1KB 的 header 和 3KB 的 dummy payload 组成。OTA Header 由 MPPackTool 生成，其中 OTA Header Image 中的 header 各字段定义如图图 3-1 所示。

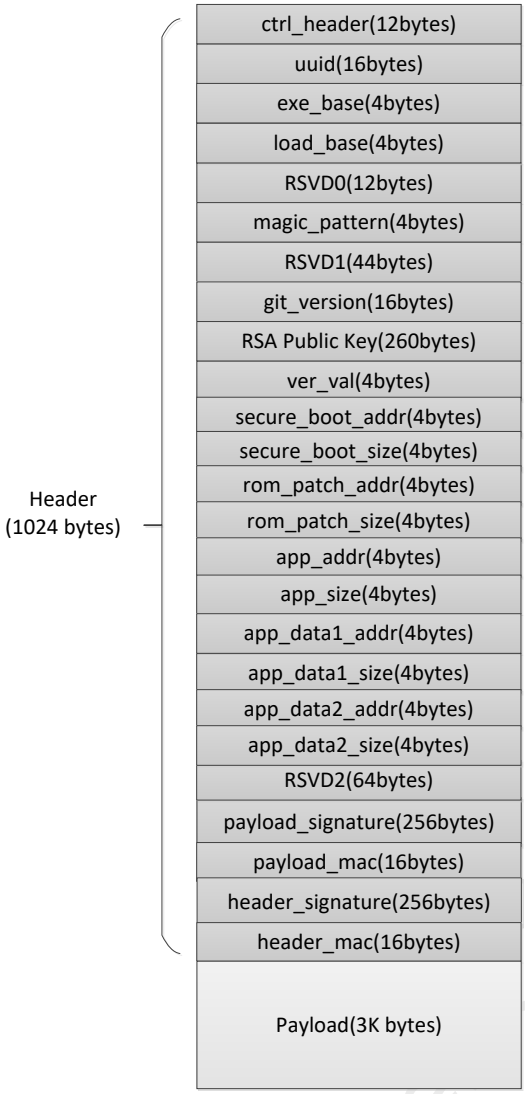


Figure 3-1 OTA Header 结构

OTA Header 中 header 部分与 OTA 相关的各字段的作用如表 3-3 所示。

Fields	Length (Byte)	Functions
ctrl_header	12	OTA Header 的控制信息
secure_boot_addr	4	Secure boot image 的起始地址
secure_boot_size	4	Secure boot image 的大小
rom_patch_addr	4	Rom patch image 的起始地址
rom_patch_size	4	Rom patch image 的大小
app_addr	4	App image 的起始地址
app_size	4	App image 的大小
app_data1_addr	4	App data1 的起始地址
app_data1_size	4	App data1 的大小
app_data2_addr	4	App data2 的起始地址

app_data2_size	4	App data2 的大小

Table 3-1: Fields of OTA Header

Patch 和 App 以及 App data 的 image 是由 1KB image header 和对应的 payload 组成。Patch 和 App 的 image header 由编译链接的时候生成，App data 的 image 的 image header 由 APP DATA 的工具添加。其中 Image header 中各字段定义如图 3-2 所示。

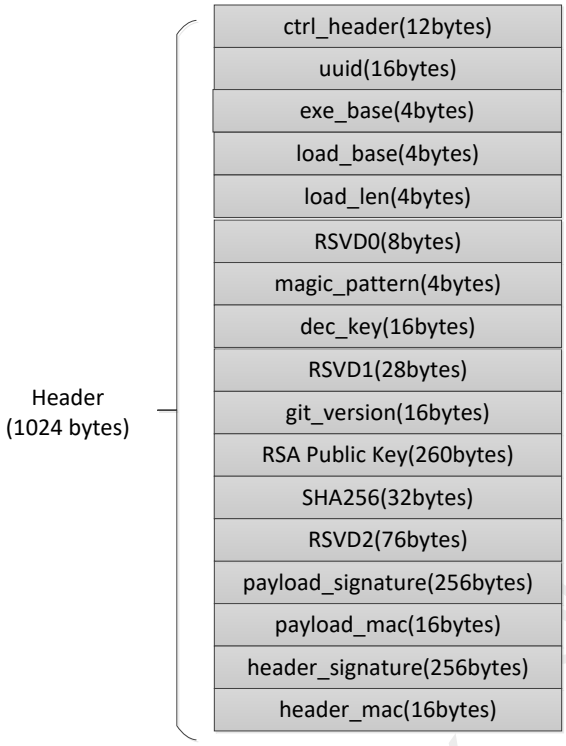


Figure 3-2 Image Header 结构

其中与 OTA 相关的字段表示的含义如表 3-4 所示。

Fields	Length(Byte)	Functions
ctrl_header	12	Image Header 的控制信息
git_version	16	版本管理的信息字段

Table 3-2: Image Header Field

Image Header 中 ctrl_header format 如下：

```
typedef struct _IMG_CTRL_HEADER_FORMAT
{
    uint8_t ic_type;
    uint8_t secure_version;
    union
    {
        uint16_t value;
        struct
        {
```

```
                uint16_t xip: 1; // payload is executed on flash
                uint16_t enc: 1; // all the payload is encrypted
                uint16_t load_when_boot: 1; // load image when boot
                uint16_t enc_load: 1; // encrypt load part or not
                uint16_t enc_key_select: 3; // referenced to ENC_KEY_SELECT
                uint16_t not_ready : 1; //for copy image in ota
                uint16_t not_obsolete : 1; //for copy image in ota
                uint16_t integrity_check_en_in_boot : 1; // enable image integrity check in boot flow
                uint16_t rsvd: 6;
            };
        } ctrl_flag;
        uint16_t image_id;
        uint16_t crc16;
        uint32_t payload_len;
    } T_IMG_CTRL_HEADER_FORMAT;
```

ic_type 表示 IC type，RTL8762C/ RTL8752C 的 ic type 值为 5,secure_version 表示启动安全检查 image 的版本。

image_id 标识不同的 image 的类型，枚举如下，其中 SCCD，OCCD，FactoryCode 是不可以 OTA 升级的。

```
typedef enum _IMG_ID
{
    SCCD            = 0x278D,
    OCCD            = 0x278E,
    FactoryCode     = 0x278F,
    OTA             = 0x2790, /* OTA header */
    SecureBoot      = 0x2791,
    RomPatch        = 0x2792,
    AppPatch        = 0x2793,
    AppData1        = 0x2794,
    AppData2        = 0x2795,
    IMAGE_MAX       = 0x2796,
} IMG_ID;
```

payload_len 表示 image 的大小，不包括 1KB 的 image header，单位是字节。

crc16 表示是进行 crc 校验还是 SHA256 校验，如果不为 0 表示进行 crc 校验，否则进行 SHA256 校验。

ctrl_flag 和 OTA 相关的位域只有 not_ready 和 not_obsolete。其中 not ready 表示 OTA 传输写入是否正确完成，默认编译的 image 中 not_ready 为 0，当 image 写到备份区的时候，先将 not_ready 置 1，直到升级传输完成，且通过完整性校验后(CRC or SHA256)，not_ready 才写为 0，表示 Image 已经 ready。not_obsolete 表示 image 是否废弃。默认编译的 image 中 not_obsolete 为 1，当支持 bank 切换的情况下,not_obsolete 在 OTA 过程中是无效的;当不支持 bank 切换的情况下， OTA 升级完成重启后，启动代码判断 not_ready 为 0 且 not_obsolete 为 1，执行将 image 从 OTA_TMP 区域搬移到相应的 image 指定区域(App,Patch,Appdata)，搬运成功后，将 OTA_TMP 区 image 中 not_obsolete 标志写为 0.

4 OTA 打包格式及工具使用

相关工具及 OTA 打包中扮演的功能：

- FlashMapGenerateTool------生成 flash map.ini 和 flash_map.h，flash_map.h 需要放到工程同级目录下参与编译， 生成 APP Image。flash map.ini 为 MPPackTool 和 MPTool 的输入文件,保证 image 和设置的所有输出地址一致。
- MPPackTool------打包 OTA 文件
- MPTool:对打包用到的 Patch 进行地址转换

4.1 支持 bank 切换

4.1.1 FLASH Layout

BANK 切换方案需要两个完全相同的 OTA bank 互为备份，所以优点就是升级完成，重启后，程序直接跳转到新的 BANK 运行， OTA 升级的切换流程非常快， 但缺点是增加了比较多的 flash 开销，所以一般情况下用 BANK 切换方案， 相对选用 flash 的 size 要比较大一点.。下面以 1Mbyte 的 flash 为例，介绍 BANK 切换方案。

下表为我们推荐的一种 Flash Layout：

sample layout for flash(total size = 1MB)	size	start addr
1) OEM Header	4K	0x801000
2) OTA Bank0	400K	0x802000
a) OTA Header	4K	
b) Secure boot loader	0K	0x803000
c) Patch code	40K	0x807000
d) APP code	160K	0x811000
e) APP data1	180K	0x839000
f) APP data2	0K	
3) OTA Bank1 (same as OTA Bank0)	400K	0x866000
a) OTA Header	4K	
b) Secure boot loader	0K	0x867000
c) Patch code	40K	0x86B000
d) APP code	160K	0x875000
e) APP data1	180K	0x89D000
f) APP data2	0K	
4) FTL	16K	0x8A1000
5) OTA Temp	0K	
6) APP Defined Section	200K	

Table 4-1：FLASH Layout Sample

注：FLASH Layout 需要根据客户实际的 Image 和 data 的大小进行合理划分。

4.1.2 打包相关工具使用步骤---切换 BANK

1. 使用 FlashMapGenerateTool 生成 flash map.ini 和 flash map.h。生成后，将 flash_map.h 拷贝到工程文件同目录，并用 keil 打开工程，编译链接生成 app_MP_sdk#####+version+MD5.bin 的文件供打包使用。切换 bank 的方式需要编译 OTA BANK0 和 OTA BANK1 的 image，通过修改工程文件同目录下的 mem_config.h 中的#define APP_BANK 实现。

```
/** @brief set app bank to support OTA: 1 is ota bank1, 0 is ota bank0 */
    #define APP_BANK                                    0
```

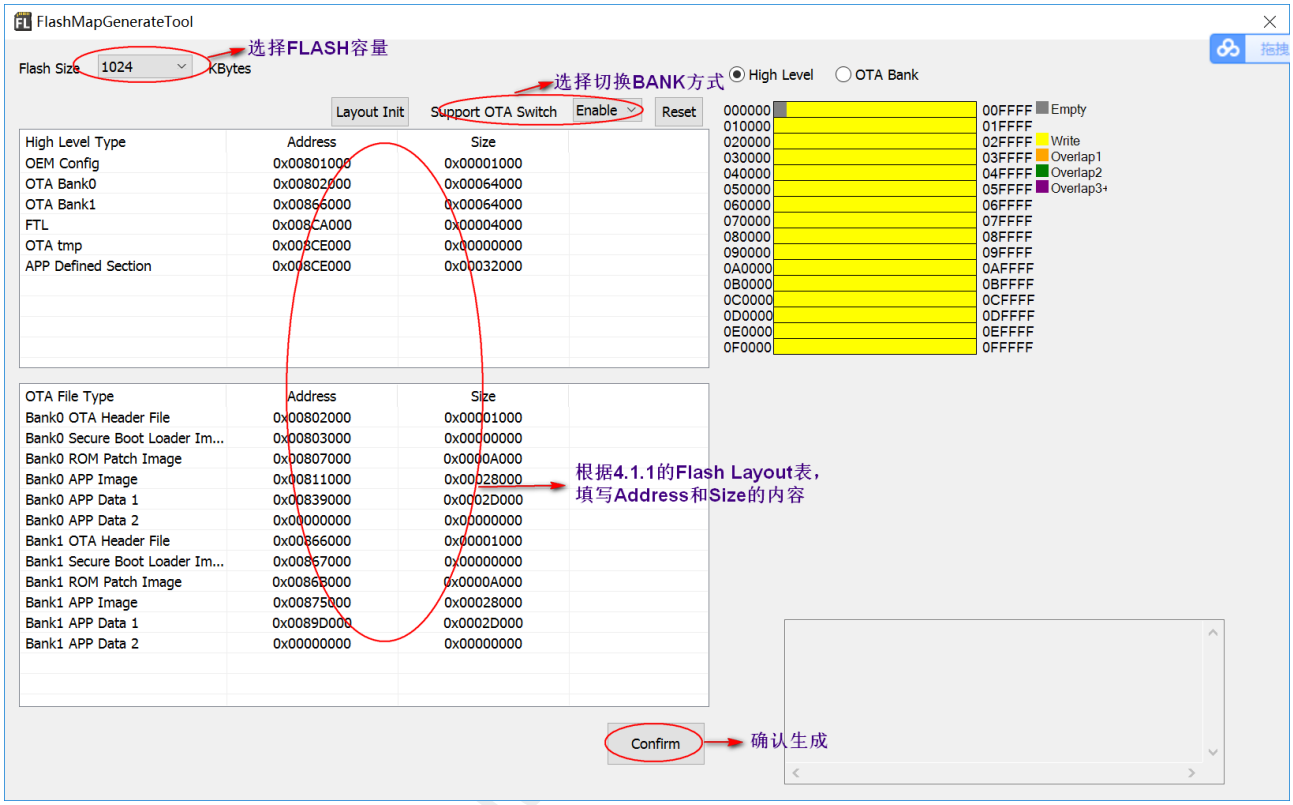


Figure 4-1 生成 Flash Layout

注：此处生成的 flash map.ini 需要和 MP 阶段使用的 flash map.ini 保持一致。

2. 打开 MP_PackTool，加载第一步生成的 flash_map.ini.，分别生成 OTA Header0，OTA Header1。

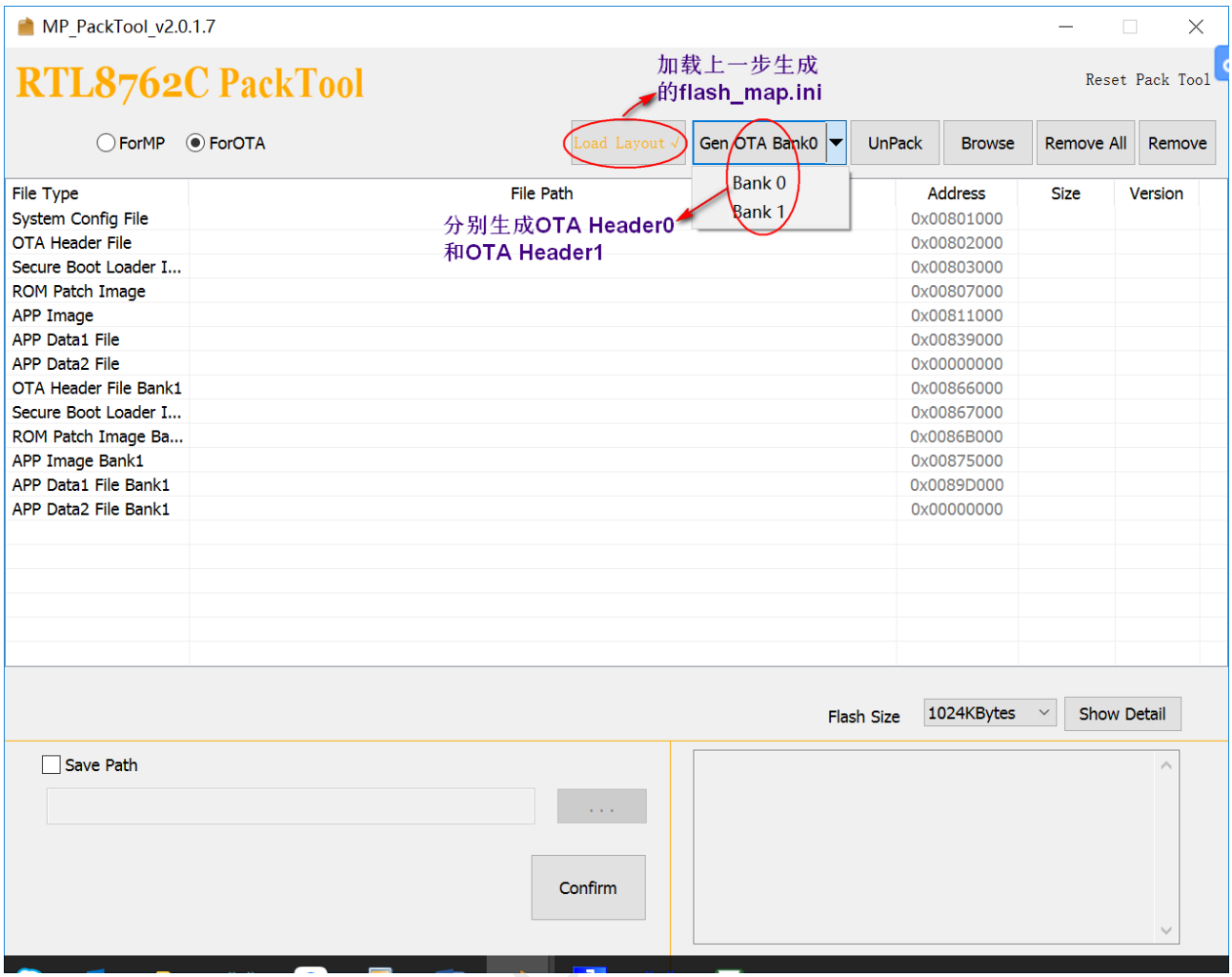


Figure 4-2 MP PACK Tool Load Flash Layout

3. 生成 OTA Header0 和 OTA Header1，图例只介绍如何生成 OTA Header0，OTA Header1 生成方式一样。

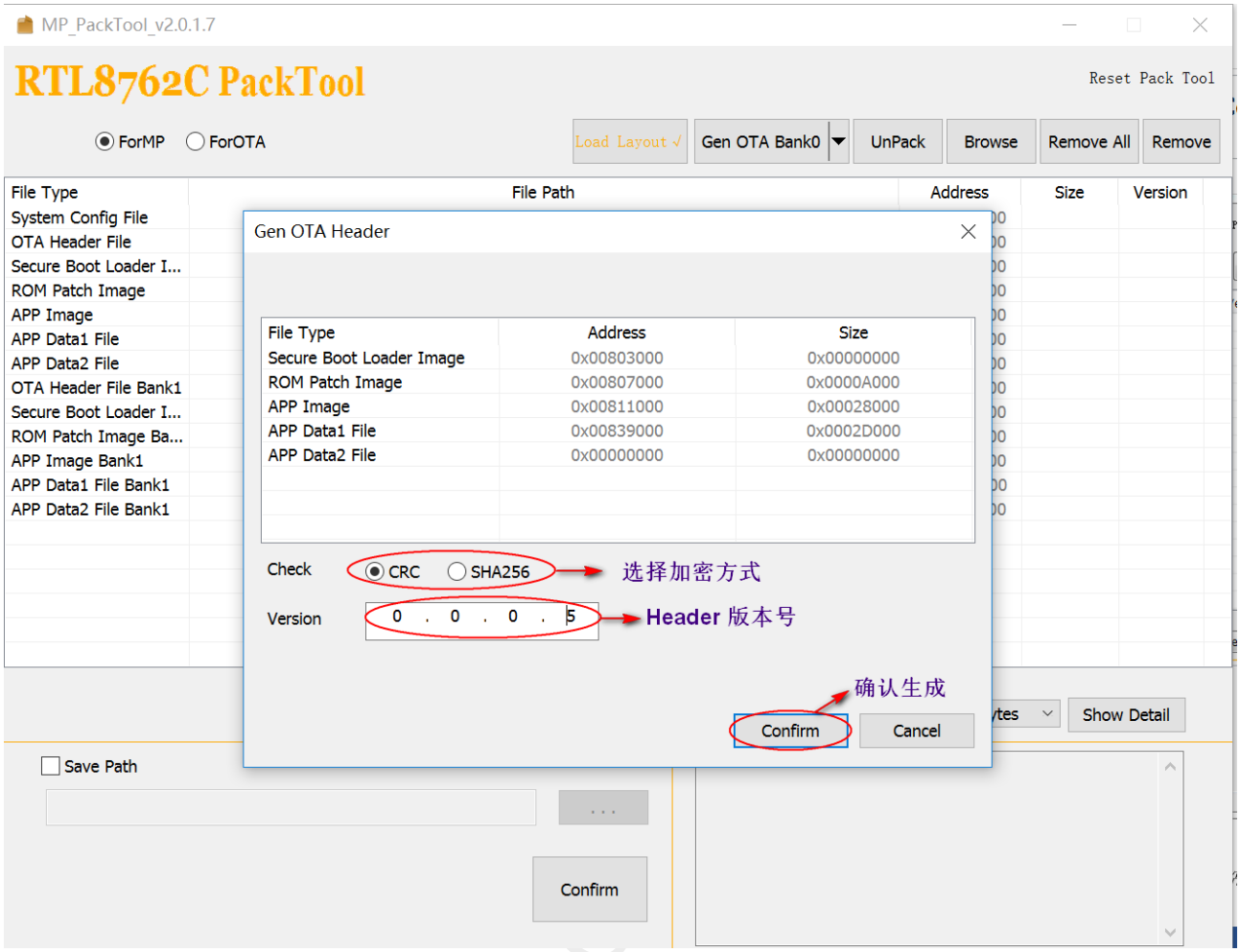


Figure 4-3 生成 OTA Header

注意：打包用的 OTA Header 的版本号，要比原来运行的版本号高，这样 OTA 升级完新 bank 才能正常生效。

4. 生成打包文件，默认在软件同目录下生成 ImgPacketFile-xxxxxx.bin，此文件为升级用的打包文件。

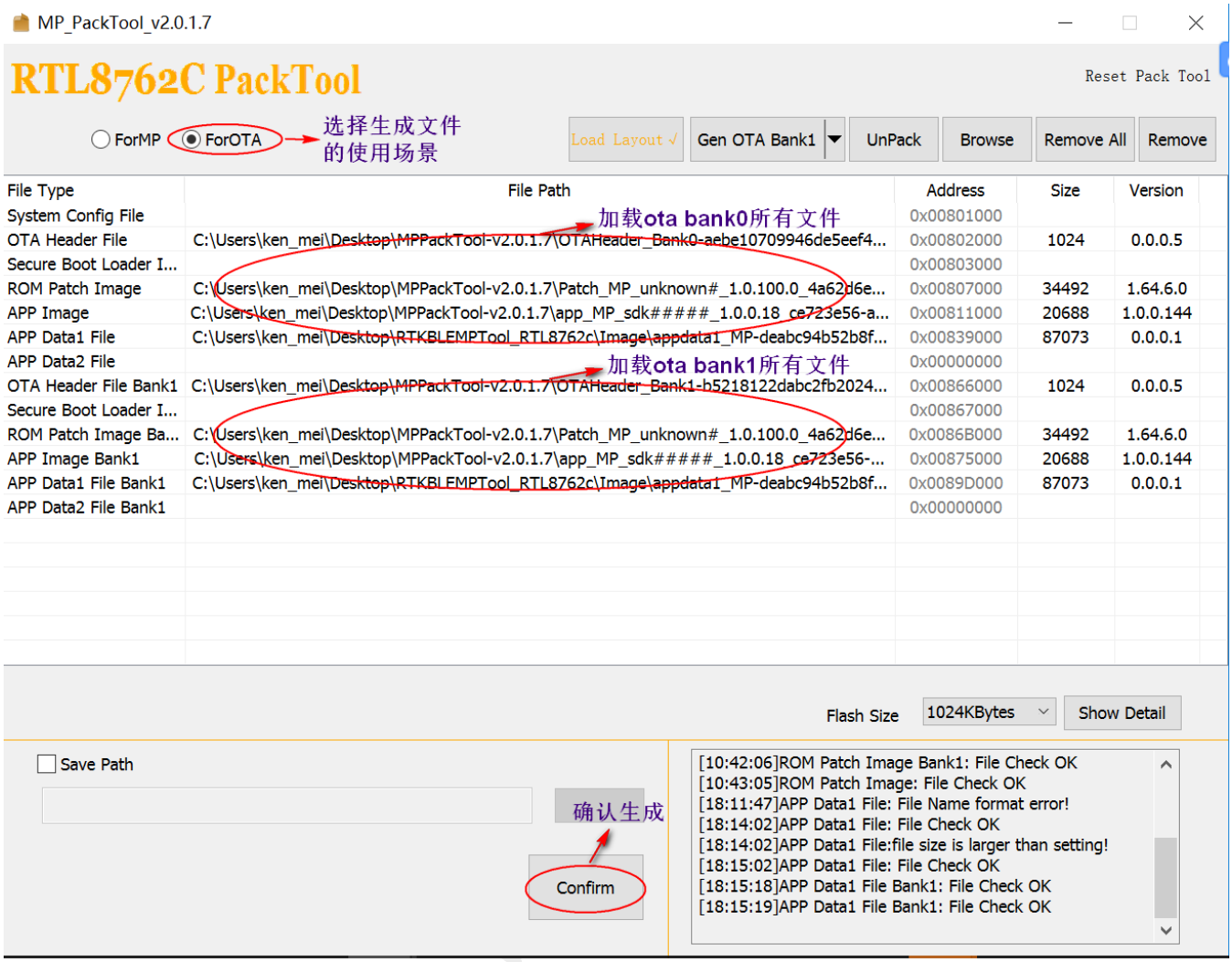


Figure 4-4 打包生成 PACK

注意：1. OTAHeader0 和 OTAHeader1 都需要打包进 PACK（和不切换 bank 的模式下不同）。

2. Flash layout 中定义到的内容都需要进行打包，缺一不可。

3. 建议 BANK0 和 BANK1 一起打进 PACK。

4. Patch img 需要经过 RTKBLEMPTool 进行地址转换。

5.APP Data 文件通过 App Data 生成脚本生成，具体参考 Bee2 Tools User Guide

4.2 不支持 bank 切换

4.2.1 FLASH Layout

不支持 BANK 切换方案和 BANK 切换方案 Flash layout 不同的地方有两点：

1. OTA BANK1 区不分配容量；
2. OTA Temp 区需要分配，且大小不小于 OTA BANK0 中容量最大的 image 的大小。

故不支持 BANK 切换方案相对节约 flash 空间， OTA 传输完成，重启 boot 程序会将 OTA Temp 区数据搬到

OTA BANK0 指定 image 区域，再重启生效，所以相对增加了升级完成的重启时间。

面以 256Kbyte 的 flash 为例，介绍不支持 BANK 切换方案。

下表为我们推荐的一种 Flash Layout：

sample layout for flash(total size = 256KB)	size	start addr
1) OEM Header	4K	0x801000
2) OTA Bank0	140K	0x802000
a) OTA Header	4K	
b) Secure boot loader	4K	0x80D000
c) Patch code	40K	0x803000
d) APP code	92K	0x80E000
e) APP data1	0K	0x825000
f) APP data2	0K	
3) OTA Bank1 (same as OTA Bank0)	0K	0x825000
4) FTL	16K	0x825000
5) OTA Temp	92K	0x829000
6) APP Defined Section	0K	

Table 4-2：FLASH Layout Sample

注：此处没有分配 APP data，FLASH Layout 需要根据客户实际的 Image 和 data 的大小进行合理划分。

4.2.2 打包相关工具使用步骤---不切换 BANK

1. 使用 FlashMapGenerateTool 生成 flash map.ini 和 flash map.h。生成后，将 flash_map.h 拷贝到工程文件同目录，并用 keil 打开工程，编译链接生成 app_MP_sdk#####+version+MD5.bin 的文件供打包使用。不切换 bank 的方式只需要编译 OTA BANK0 的 image，配置见工程文件同目录下的 mem_config.h 中的 #define APP_BANK。

```
/** @brief set app bank to support OTA: 1 is ota bank1, 0 is ota bank0 */
    #define APP_BANK                                    0
```

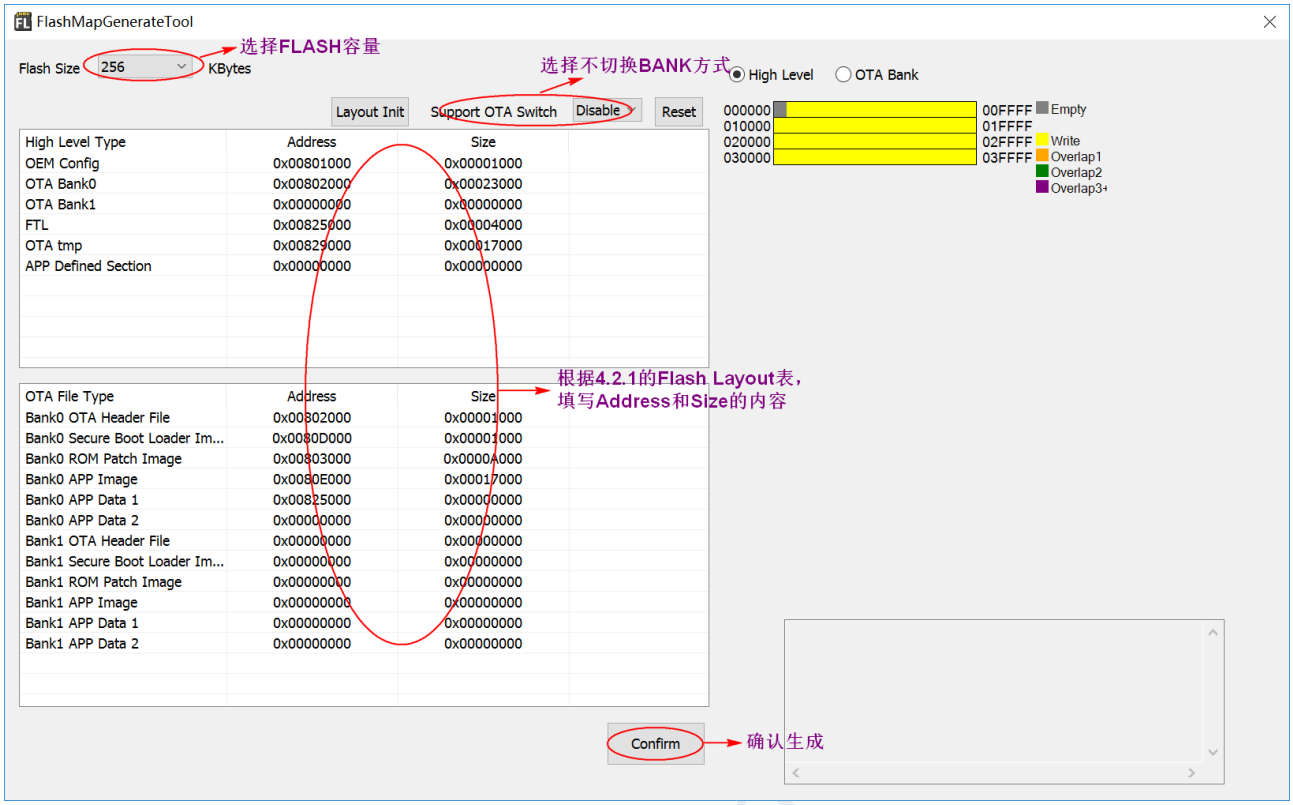


Figure 4-5 生成 Flash Layout

注：此处生成的 flash map.ini 需要和 MP 阶段使用的 flash map.ini 保持一致。

2. 打开 MP_PackTool，加载第一步生成的 flash_map.ini.，并 load 相应 image 文件。

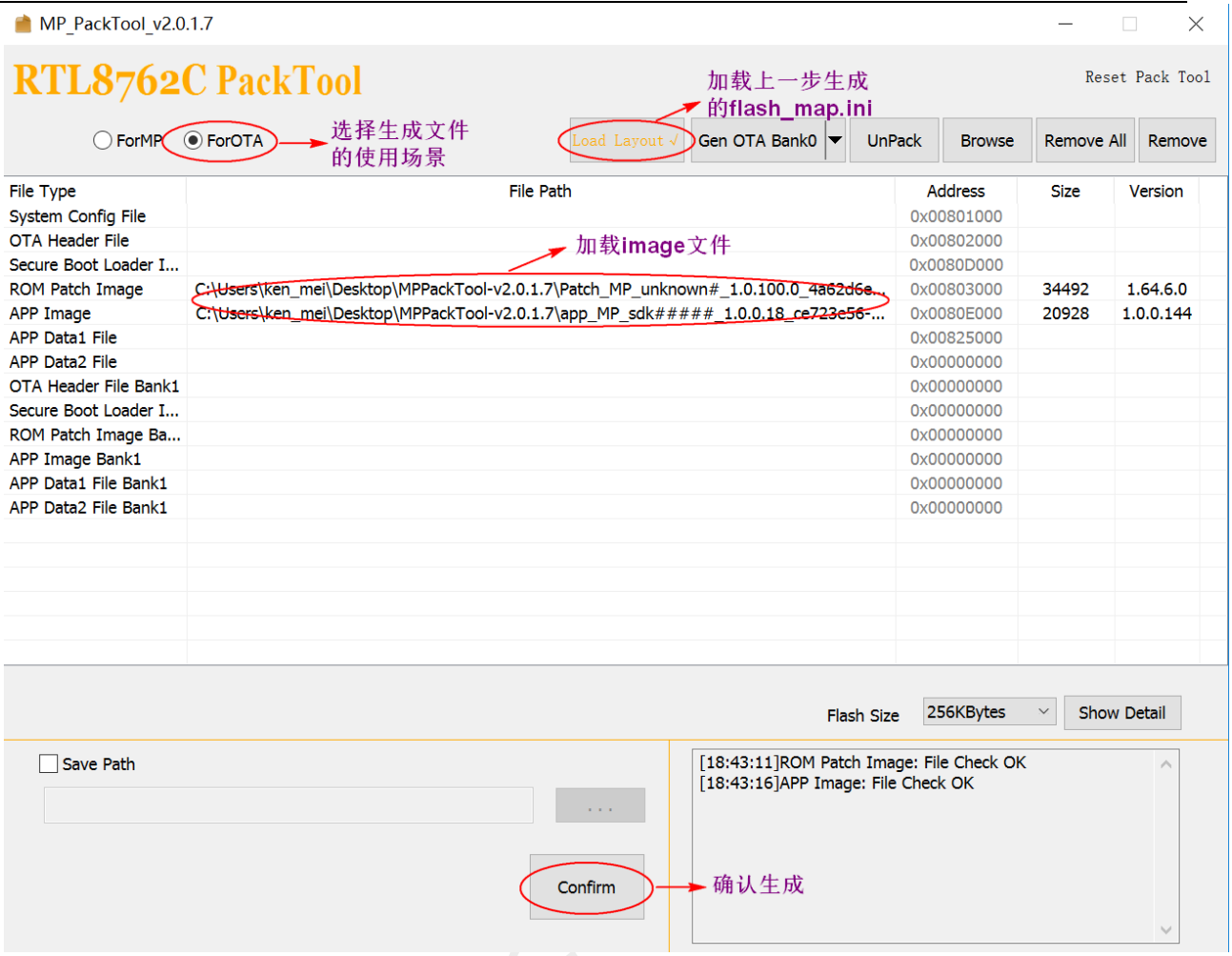


Figure 4-6 MP PACK Tool Load Flash Layout

注意：

1. *OTAHeader0 在不切换 bank 的方式不需要打包进 PACK（和不切换 bank 的模式下不同）。*
3. *Flash layout 中定义了 Secure boot loader Image 的内容，但如果没有新的 Secure boot loader Image 版本发布，建议不进行打包。*
4. *如果只需要更新 ROM Patch Image 或者 APP Image，可打包其中一个。*
5. *Patch img 一般需要经过 RTKBLEMPTool 进行地址转换，此处的 patch 地址和默认相同（0x803000），可以不转换使用。*

5 OTA 协议

5.1 DFU Service

DFU Service 的 uuid 为：{ 0x12, 0xA2, 0x4D, 0x2E, 0xFE, 0x14, 0x48, 0x8e, 0x93, 0xD2, 0x17, 0x3C, ***0x87, 0x62***, *0x00, 0x00*}.

DFU Service 定义了两个 Characteristic：

Data Characteristic 接受 image 的数据通道，属性 write no response；

Control Point Characterisic 接受控制指令的通道,属性 write/notification。

DFU Service 支持的所有 control point 如下表：

Procedure	Requirement	Properties	Parameter Description	Applicable Response Value(s)	Response Parameter
Start DFU①	M	Write	ic_type(UINT8) secure_version (UINT8) ctrl_flag.value(UINT16) image_id (UINT16) crc16((UINT16) payload_len (UINT32)	ARV	None
Receive FW image	M	Write	image_id (2byte-UINT16) nImageLength (4Byte-UINT32)	ARV	None
Validate FW	M	Write	image_id (2byte-UINT16)	ARV	None
Activate Image and Reset	M	Write	None	None	None
Reset System	M	Write	None	None	None
Report Received Image Information	M	Write	image_id(UINT16)	ARV	origin_image_version (UINT32) cur_offset (UINT32)
Connection parameter update	M	Write	connIntervalMin(UINT16) connIntervalMax (UINT16) connLatency(UINT16) supervisionTimeout (UINT16)	ARV	None
Buffer check enable	M	Write	None	ARV	Max buffer size(UINT16) Mtu size(UINT16)

Buffer check size&crc	M	Write	mBufferSize(UINT16) mCrc(UINT16)	ARV	Next send offset(UINT32)
IC type	O	Write	None	ARV	ic_type(UINT8)
Copy Img②	M	Write	image_id(UINT16) destination_addr(UINT32) copysize(UINT32)	ARV	None

Table 5-1 DFU opcode

其中①Start DFU 的参数为 img 的 ctrlheader，在接受到 ctrlheader 后，需要写进 flash 作为升级文件的一部分，所以此 12 个 bytes 和数据的传输处理一样，如果使能了 aes 加密，收到 Start DFU 的参数后，需要解密后再解析，并写入 flash 中。

②当升级 appdata 时，采用的 bank 切换方式，判断 secure version 和 APPDATA version 相同，可以使用这条命令，直接复制原来 bank 的内容到目的 bank，省去 OTA 传输环节。

在数据传输时，如启用 buffer check，buffer check 的大小必须为$(16 * 2^n)$bytes，并且要小于等于 max buffer Size(Buffer check enable 命令返回).另外如果 aes 支持的话，每 16 个 byte 都是进行 aes 加密的，收到后，需要先解密，最后的小于 16 的部分没有进行加密发送。当收满 buffer check size 后，写入 flash。

如果没有启用 buffer check，则需要一次发送 20*n(n=1,2,4,5,10) bytes，直到 rtl8762c 端收满 2000bytes，才启动写入 flash，最后一次则把文件收完就写入。如果 aes 支持的话，20*nbytes packet 数据，能被 16 整除的部分，需要加密后发送，余数小于 16bytes 则不需要加密。

只有静默升级应用方案上才需要添加 DFU Service,传统升级方案 DFU Service 在 ROM 中实现。

5.2 OTA Service

OTA Service 的 uuid 为：{ 0x12, 0xA2, 0x4D, 0x2E, 0xFE, 0x14, 0x48, 0x8e, 0x93, 0xD2, 0x17, 0x3C, ***0xFF, 0xD0***，*0x00, 0x00*}.

OTA Service 定义了如下 Characteristic：

Characteristic Name	Requirement	Mandatory Properties	Description
OTA CMD	M/O	WriteWithoutResponse	See OTA CMD
Device Mac	M	Read	See Device Mac
Patch Version	M	Read	See Patch Version
App Version	M	Read	See App Version
Patch Extension Version	O	Read	See Patch Extension Version
Test Mode	O	WriteWithoutResponse	See Test Mode
Device Info	M	Read	See Device Info
Image Counter	O	WriteResponse	See Image Counter
Image Version	M	Read	See Image Version

Table 5-2：Ota Characteristic

5.2.1 OTA CMD

UUID: 0xFFD1

该特性是允许设备进入 OTA 模式的控制端点。如果 DFU Service 跑在 ROM code，则需要此命令进入 DFU mode。

进入 ROM code 部分的代码如下：

```
dfu_switch_to_ota_mode();
WDG_SystemReset(RESET_ALL_EXCEPT_AON, DFU_SWITCH_TO_OTA);
```

如果 DFU Service 跑在 APP code，此命令则为可选的（可以用来同步 Master APK/APP，做兼容方案）。

Names	Field Requirement	Format	Value
OTA CMD	Mandatory	Uint8	1

Table 5-3: OTA CMD characteristics

5.2.2 Device Mac

UUID: 0xFFD2

该特征值用来读取 RTL8762C 设备 BDA (Bluetooth Device Address)以与 OTA 模式中的扫描 BDA 进行比较。

Name	Field Requirement	Format	Value
Device Mac	Mandatory	Uint8*6	XX:XX:XX:XX:XX:XX

Table 5-4: Device Mac characteristics

5.2.3 Patch Version

UUID: 0xFFD3

该特征值用来读取设备的 patch 版本号，该特性为兼容 Bee1 版本，在 Bee2 不推荐使用，Bee2 中版本信息在 Image version 描述。

Name	Field Requirement	Format	Value
Patch Version	Mandatory	Uint32	0xNNNNNNNN

Table 5-5: Patch Version characteristic for Bee2 (not recommend, described in image version)

5.2.4 APP Version

UUID: 0xFFD4

该特征值用来读取设备的 APP 版本号，该特性为兼容 Bee1 版本，在 Bee2 不推荐使用，Bee2 中版本信息在 Image version 描述。

Name	Field Requirement	Format	Value

APP Version	Mandatory	Uint32	0xNNNNNNNN

Table 5-6: APP Version characteristic for Bee2(Not recommend, described in image version)

5.2.5 Patch Extension Version

UUID: 0xFFD5

该特征值用来读取设备的 patch extension 版本号，该特性只有 Bee1 版本有，在 bee2 并没有此特性。

Name	Field Requirement	Format	Value
Patch extension Version	Optional	Uint16	0xNNNN

Table 5-7:**错误!未找到引用源。** Patch Extension Version characteristic

5.2.6 Test Mode

UUID: 0xFFD8

该特性是允许设备退出测试模式的控制端点，写‘1’清除测试标志，并退出 MP 模式。

Name	Field Requirement	Format	Value
Test mode	Optional	Uint8	1

Table 5-8:Test Mode characteristics

注：此特性和 OTA 无关

5.2.7 Device Info

UUID: 0xFFF1

该特征值用来读取设备的固件基本信息，描述如下：

For the other BT SoC , the characteristic is listed as below.

Name	Field Requirement	Format	Value
Device info	Mandatory	As Table 10	As Table 10

Table 5-9: Device info characteristic for Bee2.

Format	ICType	Version	Secure Version	MODE		Max Buffer Size	Reserved
	8bit	8bit	8bit	8bit		16bit	16bit
Value	BBpro:4 BEE2:5	Bit3~0: OTA		Bit 0	0:normal mode 1:Support buffer check	0xNNNN	0x00

	version = 0x1 Bit7~4: Reserved: 0x0.	Bit 1	0:Aes flag not set 1:Aes flag Set
		Bit 2	0: Only encrypt first 16 bytes of OTA data in normal mode. 1:Encrypt 16*N bytes of OTA date in normal mode
		Bit3	0: Disable Copy Image. 1: Enable Copy Image.
		Bit4	0: Update one Image at a time. 1: Update multiple Images at a time.

Format (Attach to above table)	Image Version Indicator
	32bit
Value (Attach to above table)	0xNNNNNNNN Indications for each image version. Each indication uses 2 bits. 00: image does not exist. 01: image exists in bank0, OTA should update image for bank1. 10: image exists in bank1, OTA should update image for bank0. 11: image is standalone. OTA should update image for standalone. bit[1:0]: Image 0 … bit[2N+1:2N]:Image N Image indicator for bee2 is as below:

Image 0	SOCV Config File
Image 1	System Config File
Image 2	OTA Header File
Image 3	Secure Boot Loader Image
Image 4	ROM Patch Image
Image 5	APP Image
Image 6	APP Data1 File
Image 7	APP Data2 File

Table 5-10: Device info Format For Bee2(OTA version = 1)

5.2.8 Image Counter

UUID: 0xFFF2

该特性是写入响应，告知设备接下来有多少个 image 文件需要写入设备。

Name	Field Requirement	Format	Value
Image Counter	Optional	Uint8	0xNN

Table 5-11: Image Counter Characteristics

5.2.9 Image Version

Image Version

UUID: 0xFFE0~FFEF

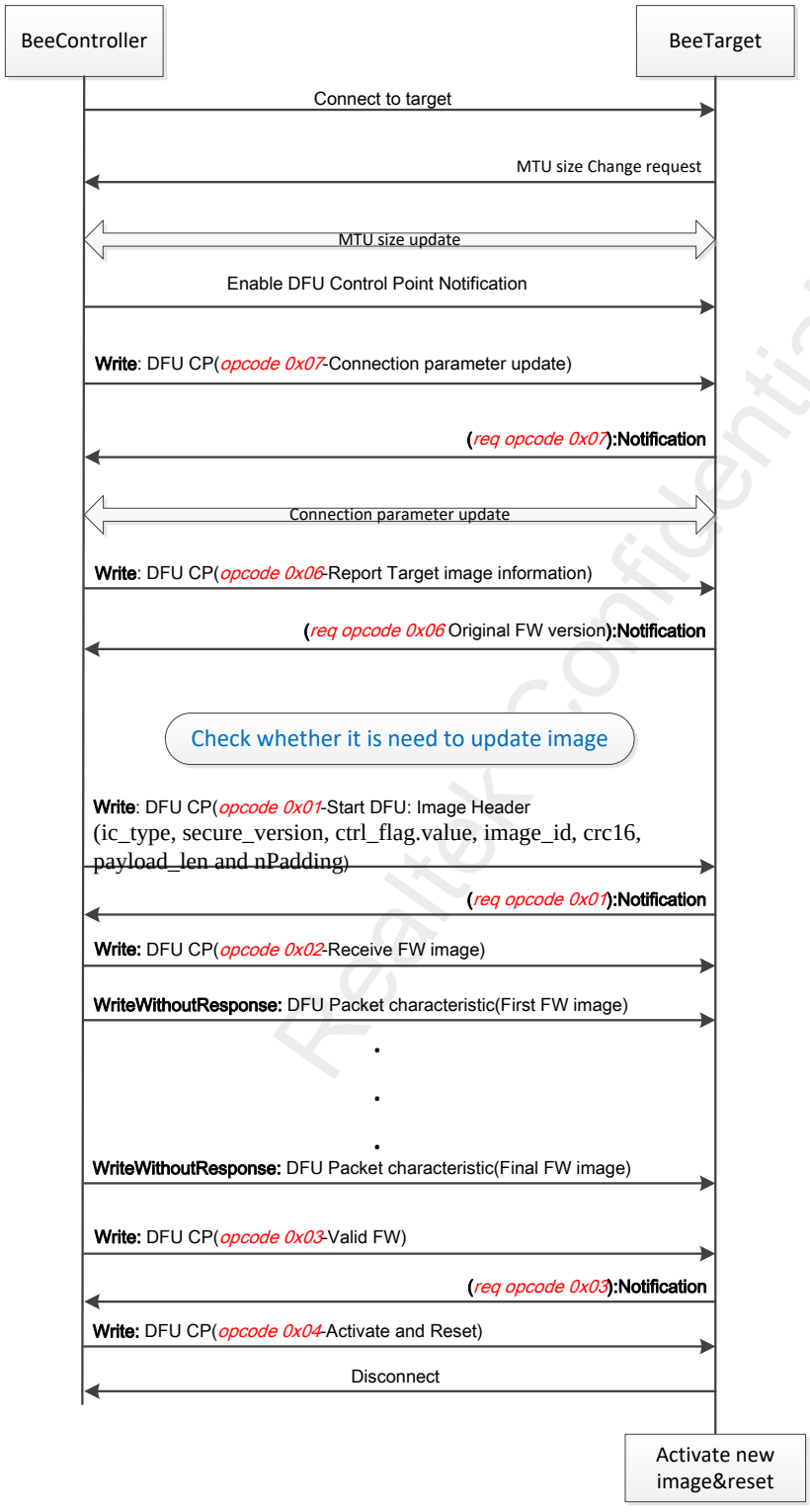
该特性是读取设备的 image 版本.每个 image 版本占用 4 个 bytes，由于受默认 MTU size（20 bytes）的限制，当 image 数量大于 5 时，则需要下一定义的特性（**UUID: 0xFFE0~FFEF**）去读取接下来的 image 版本。设备的 image 版本数量是通过 Image Version Indicator 去识别的。Image Version Indicator 在 Device Info（0xfff1）中定义。

Name	Field Requirement	Format	Value
Image Version	mandatory	Uint32*N	

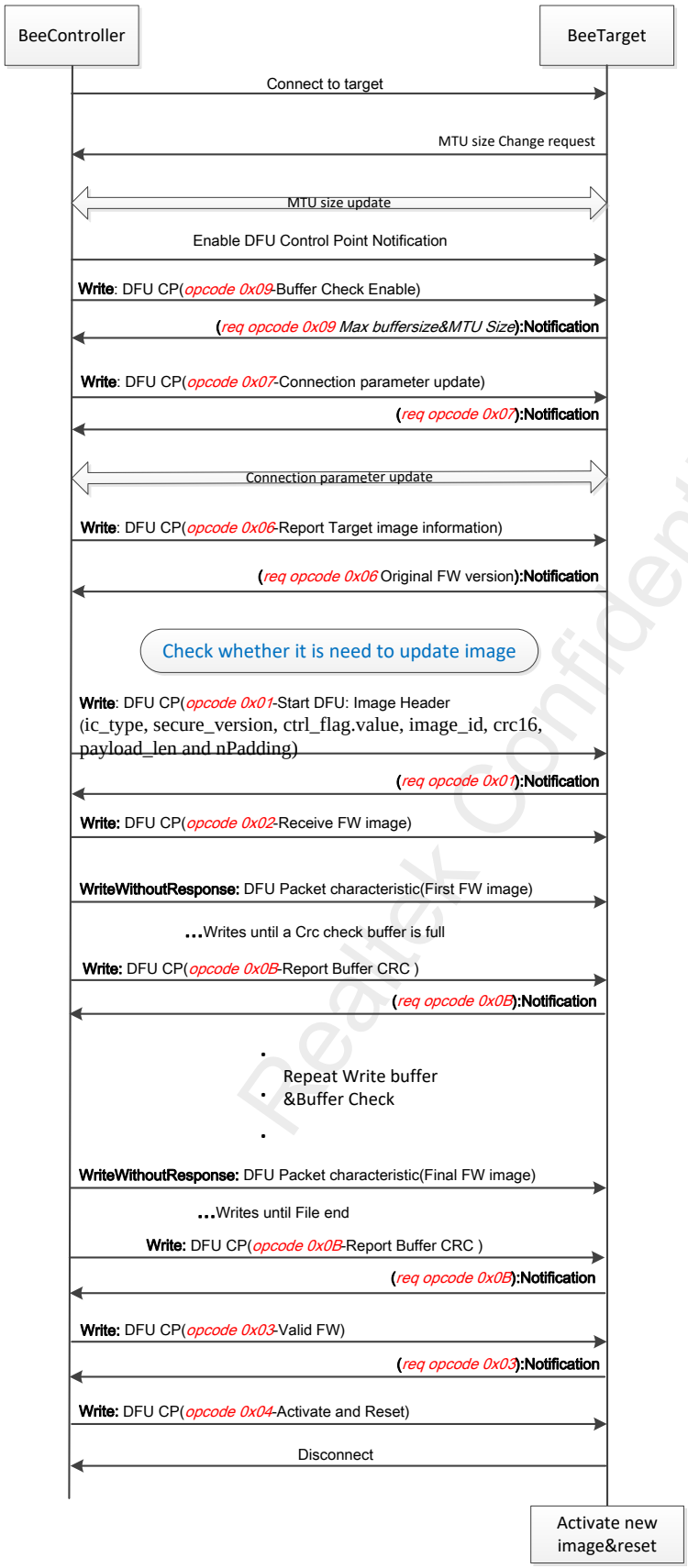
Table 5-12: Image Counter characteristics

5.3 OTA 流程

5.3.1 OTA 不带 buffer check 流程



5.3.2 OTA buffer check 流程



5.3.3 多文件更新

1. 不支持 bank 切换的情况，当打包待升级文件中包含 Patch，APP 或者 APPDATA，需要升级一个文件验证成功后，重启使之生效，再升级下一个文件。
2. 支持 bank 切换的情况。当打包待升级文件中包含 OTA Header ，Patch，APP 或者 APPDATA，需要升级验证一个文件，再升级验证下一个文件，等所有的文件都升级验证成功后，才能进行重启，否则本次升级无效。原因是切换 bank 的方式，需要 bank 区所有的文件都一起生效，才能正常运行。

6 Master 软件的使用

略

7 参考文献

[1] 《RTL8762C Device Firmware Update Profile Rom Version》

[2] 《RTL8762C Device Firmware Update Service Rom Version》