ASR1601 FOTA



**(c) Copyright, ASR Microelectronics, Inc., 2015-present.**

**This software, document, web pages, or material (the "Work") is copyrighted by its respective copyright owners.  The Work may be confidential and proprietary.  The Work may be further protected by one or more patents and be protected as a part of a trade secret package.**

**No part of the Work may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without prior written consent of the copyright owner. Any other reproduction in any form without the permission of the copyright owner is prohibited.**

**All Work are protected by the copyright laws of all relevant jurisdictions, including protection under the China and US copyright laws, and may not be reproduced, distributed, transmitted, displayed, published, or broadcast without the prior written permission of the copyright owner.**

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Author | Description |
| 0.1 | 2020-02-10 | huaiwenshi | Initial draft |
| 0.2 | 2020-02-28 | huaiwenshi | Support mini system ota |
| 0.3 | 2020-03-12 | huaiwenshi | Support full system ota |
| 0.4 | 2020-05-07 | huaiwenshi | Update mini system ota |
| 0.5 | 2020-05-21 | huaiwenshi | Update mini system ota |
| 0.6 | 2020-06-04 | huaiwenshi | FBFMake\_CF\_1.4.exe support fast mode |
| 0.7 | 2020-07-07 | huaiwenshi | FBFMake\_CF\_1.5.exe support multi partition dfota |
| 0.8 | 2020-09-16 | huaiwenshi | Support PRO FoTA |
| 0.9 | 2020-09-29 | huaiwenshi | FBFMake\_CF\_1.6.exe support multi partition dfota |
| 1.0 | 2022-07-15 | huaiwenshi | Mini ota optimized |
| 1.1 | 2023-02-16 | huaiwenshi | Mini ota optimized |

# 

# Table of Contents

Revision History 1

Table of Contents 2

1. Introduction 3

1.1. Request & Purpose 3

1.2. Definitions &Abbreviations 3

1.3. Reference 3

2. 全系统差分升级方案 4

2.1. 全系统差分升级制作升级包 4

2.2. 差分包的下载 5

2.3. OTA升级状态查询命令 6

2.4. OTA升级示例 6

2.5. 注意事项 6

3. 全系统完整包升级方案 7

3.1. 全系统完整包升级制作升级包 7

3.2. 差分包的下载 8

3.3. OTA升级状态查询命令 8

3.4. OTA升级示例 8

3.5. 注意事项 8

4. 最小系统的升级 8

4.1. 最小系统升级方案思路 8

4.2. 使用工具adiff.exe制作差分包 9

4.2.1. 生成两个升级包的方法 9

4.2.2. 生成一个升级包的方法 11

4.2.3. 支持没有updater分区的差分包制作方法 11

4.2.4. 切割大小的设置 11

4.2.5. 扩展分区用于fota升级 11

4.2.6. 不支持掉电保护 12

4.3. 升级包的下载步骤 12

4.3.1. 两个升级包的下载步骤 12

4.3.2. 一个升级包的下载步骤 12

4.4. 注意事项 13

5. 新的OTA方案（PRO 版本） 14

5.1. Pro升级方案介绍 14

5.2. 使用工具adiff.exe制作差分包 14

5.3. 升级包的下载步骤 15

5.4. 注意事项 15

# Introduction

## Request & Purpose

文档主要介绍ASR1601/1602/1603/1606/1608/1609 系统的Fota方案，供开发测试人员参考.

## Definitions &Abbreviations

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Reference

1. …

# 全系统差分升级方案

## 全系统差分升级制作升级包

运行fbfmake\_fast.bat快速制作差分包，生成差分文件fbf\_dfota.bin

运行fbfmake\_dfota.bat制作差分包，生成差分文件fbf\_dfota.bin

fbfmake\_fast.bat主要执行的命令为*FBFMake\_CF\_V1.6-150.exe -f config -d 0x10000 -a a -b b -o fbf\_dfota.bin -q*

-o :代表输出的差分文件，为fbf\_dfota.bin

-f :制作差分包需要的配置文件，config文件中的内容如下，通过image\_list支持多个分区的同时差分，里面的内容可以根据具体使用进行增减。

[Image\_List]

Number\_of\_Images = 4

1\_Image\_Enable = 1

1\_Image\_Image\_ID = 0x30

1\_Image\_Path = system.img

1\_Image\_Flash\_Entry\_Address = 0x00024000

1\_Image\_ID\_Name = 1

2\_Image\_Enable = 1

2\_Image\_Image\_ID = 0x31

2\_Image\_Path = app.bin

2\_Image\_Flash\_Entry\_Address = 0x006DD000

2\_Image\_ID\_Name = 2

3\_Image\_Enable = 1

3\_Image\_Image\_ID = 0x32

3\_Image\_Path = cusapp.bin

3\_Image\_Flash\_Entry\_Address = 0x008DD000

3\_Image\_ID\_Name = 3

4\_Image\_Enable = 1

4\_Image\_Image\_ID = 0x33

4\_Image\_Path = logoapp.bin

4\_Image\_Flash\_Entry\_Address = 0x009DD000

4\_Image\_ID\_Name = 4

-d ：代表差分升级使用的切割块大小，一般为0x10000, 64K

-a :（after）设置升级后的image文件路径，比如a文件夹下会放入升级后版本的system.img

-b :（before）设置升级前的image文件路径，比如b文件夹下会放入升级前版本的system.img

-q: 可选参数，如果选择该参数，差分将选择fast模式，制作差分包的时间将大大缩小，但是该模式可能会导致差分包比非fast模式要大，在大部分情况下fast模式和非fast模式输出差分包大小是基本相同。

## 差分包的下载

生成的差分文件fbf\_dfota.bin放入服务器端，使用AT命令可以下载升级测试

OTA升级命令：

该命令是用于开启OTA升级服务，支持FTP和HTTP两种方式升级；

|  |  |
| --- | --- |
| AT$MYFOTA=<channel>,<mode>,<destination\_ip/url>,<username>,<password> | <CR><LF>OK<CR><LF> |
| <CR><LF>ERROR:<err><CR><LF> |

1. <channel>：0-5，通道号；
2. <mode>：0-1；
   * 0：FTP方式
   * 1：HTTP方式
3. <destination\_ip:port/url>：远程站点IP或者URL，IP地址为点分十进制表示法形式:XXX.XXX.XXX.XXX；URL: ASCII字符，最大长度为255个字节；如<port>省略，则默认ftp端口为21,http端口为80
4. <username>：登录用户名，ASCII字符，最大长度为128个字节；
5. <password>：登录密码，ASCII字符，最大长度为128个字节；

AT$MYFOTA =1,0,”172.22.44.123/fbf\_dfota.bin”,asr,fota // FTP方式打开OTA升级

OK //OTA升级启动成功

AT$MYFOTA =1,1,”172.22.44.123/fbf\_dfota.bin”,asr,fota // HTTP方式打开OTA升级

OK //OTA升级启动成功

## OTA升级状态查询命令

|  |  |
| --- | --- |
| 命令 | 返回值 |
| AT$MYFOTA? | <CR><LF>$MYFOTA:<Status><CR><LF> |

|  |  |
| --- | --- |
| Status代码 | 状态描述 |
| 0-99 | 下载文件并校验的进度值 |
| 100 | OTA文件下载检查并保存成功 |
| 1001 | 域名不存在 |
| 1002 | 域名解析超时 |
| 1003 | 域名解析未知错误 |
| 1004 | 连接服务器失败 |
| 1005 | 用户名密码认证失败 |
| 1006 | 文件不存在 |
| 1007 | 文件大小异常 |
| 1008 | 文件接收失败 |
| 1009 | 文件校验失败 |
| … | 备用 |

## OTA升级示例

AT$MYFOTA =1,0,”172.22.44.123/fbf.bin”,asr,fota ==>开始FOTA升级

OK

AT$MYFOTA?

$ MYFOTA: 100 ==>FOTA升级查询的结果

## 注意事项

1.Flash中的fotaparam区域要求必须要等于或者大于12K，最好设置为12K；

2.Flash中存放差分包文件的fotapkg区域大小需要满足能够存放差分包文件；

# 全系统完整包升级方案

## 全系统完整包升级制作升级包

运行fbfmake\_fota.bat，生成差分文件fbf\_fota.bin

fbfmake\_fota.bat 主要执行的命令为*FBFMake\_CF\_V1.6-150.exe -o fbf.bin -f config -a a -b a*

-o :代表输出的差分文件，为fbf\_fota.bin

-f :制作完整包升级需要的配置文件，config文件中的内容如下，通过image\_list支持多个分区的同时升级，里面的内容可以根据具体使用进行增减。

[Image\_List]

Number\_of\_Images = 4

1\_Image\_Enable = 1

1\_Image\_Image\_ID = 0x30

1\_Image\_Path = system.img

1\_Image\_Flash\_Entry\_Address = 0x00024000

1\_Image\_ID\_Name = 1

2\_Image\_Enable = 1

2\_Image\_Image\_ID = 0x31

2\_Image\_Path = app.bin

2\_Image\_Flash\_Entry\_Address = 0x006DD000

2\_Image\_ID\_Name = 2

3\_Image\_Enable = 1

3\_Image\_Image\_ID = 0x32

3\_Image\_Path = cusapp.bin

3\_Image\_Flash\_Entry\_Address = 0x008DD000

3\_Image\_ID\_Name = 3

4\_Image\_Enable = 1

4\_Image\_Image\_ID = 0x33

4\_Image\_Path = logoapp.bin

4\_Image\_Flash\_Entry\_Address = 0x009DD000

4\_Image\_ID\_Name = 4

-a :（after）设置升级后的image文件路径，比如a文件夹下会放入升级后版本的system.img

-b :（before设置升级前的image文件路径，比如b文件夹下会放入升级前版本的system.img

当-a和-b指向的是同一个文件夹，里面放的是要升级的system.img，这时就是代表全系统完整升级，和差分升级通过制定路径进行区分

## 差分包的下载

同2.2节

## OTA升级状态查询命令

同2.3节

## OTA升级示例

同2.4节

## 注意事项

1.Flash中的fotaparam区域要求必须要等于或者大于12K，最好设置为12K；

2.Flash中存升级文件的fotapkg区域大小需要满足能够存放下完整系统升级包；

# 最小系统的升级

## 最小系统升级方案思路

系统image分为可连网的最小系统和非最小系统，使用制作工具生成两个升级文件，一个是最小系统差分文件，一个是非最小系统升级文件。

升级时首先下载最小系统差分文件，下载完成后重启进入updater升级。

完成最小系统的升级后重启运行的就是最新的可连网的最小系统，然后再下载非最小系统升级文件，该文件是全包非差分文件，升级时直接覆盖非最小系统区域，下载完成后校验，校验通过后整个升级完成。

## 使用工具adiff.exe制作差分包

### 生成两个升级包的方法

**adiff.exe system\_old.img system\_new.img system\_patch.bin**

**<Options>**

**[-a1 <app1\_partition\_name> <app1\_file\_name> [-a2 <app2\_partition\_name> <app2\_file\_name>]]**

**<Options>**

**[-s size]**

**<Options>**

**[-u updater.bin]**

**<Options>**

**[-ext flash partition name]**

**<Options>**

**[-m]**

**<Options>**

**[-npop]**

前三个参数是必选项， –a1是可选项， –a2是可选项

**system\_old.img**参数是板子中已经烧写的system文件, 如1601升级包system.img重命名为system\_old.img;

**system\_new.img**参数是要升级到的system文件, 如1601升级包system.img重命名为system\_new.img;

**system\_patch.bin**参数是生成的升级文件名称；

**-a1**代表当系统中存在和system同级的客户分区app1，如下图所示，分区app1\_partition\_name名称假设为app1，当要升级这个分区时可以使用这个参数，并将要升级的app1\_file\_name文件放在**adiff.exe**同一路径下， app1\_file\_name文件在工具制作时会被放入**system\_patch.bin\_2**中，在第二阶段升级；

（app1分区名称客户可以在aboot工具下修改，这种分区主要是给客户增加一些特殊的分区，然后也能够升级，使整个最小系统升级更灵活）

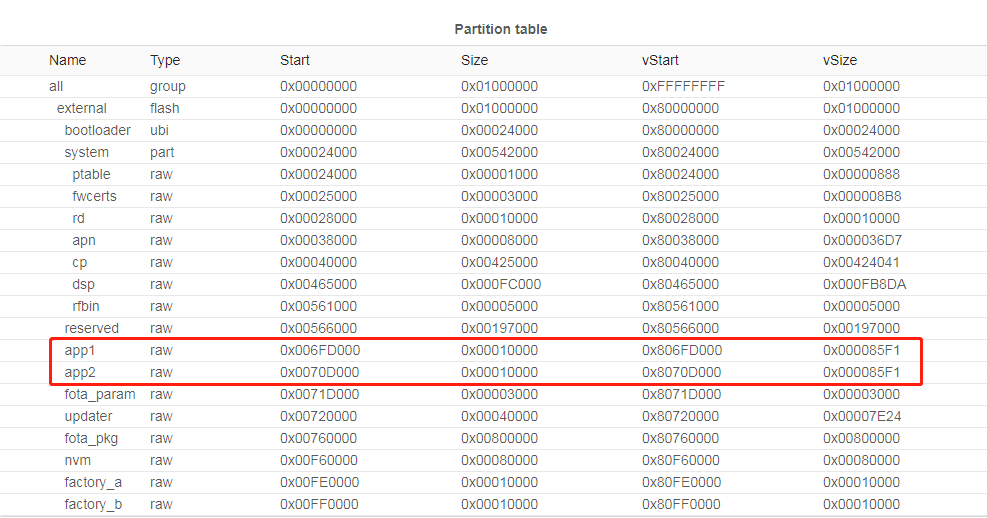
-a1也可以指定为updater分区，这样在第二阶段升级就可以升级updater，命令如下：

adiff.exe system\_old.img system\_new.img system\_patch.bin –a1 updater updater.bin

**app1\_partition\_name**代表分区的分名称；

**app1\_file\_name**代表要升级的分区文件；

**-a2**同上**a1**，最小系统升级最多支持同时升级两个app分区；（注意adiff只能最多添加两个app分区用于升级）



运行后会产生两个升级包，**system\_patch.bin\_1**代表最小系统差分升级文件，**system\_patch.bin\_2**代表非最小系统升级文件

**[-m]**

参考4.2.2

**[-u updater.bin]**

从adiff 5.6版本开始支持，参考4.2.3

**[-s size]**

参考4.2.4。

**[-ext flash partition name]**

从adiff 5.6.4版本开始支持，参考4.2.5。

**[-npop]**

No power off protection，不支持掉电保护，参考4.2.6。

### 生成一个升级包的方法

参考4.2.1中方法会生成两个升级包，可以带上参数-m生成一个升级包，如下

**adiff.exe system\_old.img system\_new.img system\_patch.bin –m**

默认生成两个升级文件，客户可以通过设置该参数将两个升级包合并在一起，组成一个升级包，在升级过程中，第一阶段下载文件的第一部分，在第二阶段下载文件的第二部分，通过断点续传功能下载部分数据，HTTP和FTP方式都可以支持，一个升级包的下载步骤可以参考4.3.2

### 支持没有updater分区的差分包制作方法

为了节省flash空间，可以删除updater分区，删除updater分区后可以将updater.bin和差分包拼接在一起，生成升级包；下载升级包后，boot33可以找到升级包里的updater.bin，从而运行updater.bin进行差分还原；

参考4.2.1中方法，在制作差分包时将updater.bin合入差分包，可以带上参数-u生成，如下

**adiff.exe system\_old.img system\_new.img system\_patch.bin –u updater.bin**

### 切割大小的设置

有时制作差分包时报空间不足，我们可以通过-s来设置切割块大小，调小后看看是否能解决空间不足的问题

-s代表对非压缩区域进行切割大小设置，切割概念参考设计文档，切割size越小，升级时需要的备份size越小，但是差分包效果变差，差分包可能会变大，8M的flash默认是1M，4M的flash默认是256KB，2M的flash默认是128KB。

-s设置的太小，但是非压缩区很大时可能差分包时报错，因为切割块太多导致头部存放的切割信息放不下

如下, **–s 80000**按照512KB切割

**adiff.exe system\_old.img system\_new.img system\_patch.bin –s 80000**

### 扩展分区用于fota升级

当FOTA时空间不足，导致制作差分包失败时。可以利用reserved后的分区配合fota升级。

注意1：ext指定用于fota的分区将和reserved组成连续的地址空间用于升级，所以要求指定的用来fota的分区和reserved分区地址必须连续

注意2：如果ext制定用于fota的分区是一个客户的app分区，那么在制作差分包时需要将该分区的固件加入升级包，差分包工具识别不出每个分区是否必须要有固件（如NVM里是不需要固件的），所以如果没有将固件加入升级包就存在customer app分区破坏的风险，工具不能提前做出提示，需要客户在自己注意；如下将customer\_app分区扩展用来fota升级，customer\_app固件需要加入升级包

**adiff.exe system\_old.img system\_new.img system\_patch.bin –ext customer\_app –a1 customer\_app customer\_app.bin**

注意3：ext指定用于fota的分区在升级下载固件过程中不能有其他程序对该分区进行擦写操作；

### 不支持掉电保护

制作升级包时，工具会检测当前空间是否满足，差分包的size+备份size能否满足当前设备的空间，备份size用于断掉保护，当设置这个参数说明不需要断电保护，那么工具在制作差分包时只需要检测差分包的size能否在当前设备中放的下；

## 升级包的下载步骤

### 两个升级包的下载步骤

1.首先输入命令**AT$MYMINISYS=1**，系统重启后进入最小系统运行，重启后可以使用**AT$MYMINISYS?**查询状态，返回**$MYMINISYS ：0**是非最小系统，返回**$MYMINISYS ：1，1**代表处于最小系统的第一个阶段，需要下载第一个升级文件，**$MYMINISYS ：1，2**代表处于最小系统的第二个阶段，需要下载第二个升级文件

2.使用2.3节介绍的OTA下载命令AT$MYFOTA，先下载最小系统差分升级文件**system\_patch.bin\_1**，下载后校验成功后会自动重启，重启后执行updater升级最小系统；

3.待最小系统升级后，自动重启后继续运行新的最小系统，此时再使用2.3节介绍的OTA下载命令AT$MYFOTA下载非最小系统升级文件**system\_patch.bin\_2**；下载校验成功整个升级完成，自动重启运行全系统

### 一个升级包的下载步骤

1.直接使用2.3节介绍的OTA下载命令AT$MYFOTA，后续FOTA会自动根据当前处于的最小系统状态进行自动下载升级包；

2.升级过程可以使用**AT$MYMINISYS?**查询状态，返回**$MYMINISYS ：1，1**代表处于最小系统的第一个阶段，需要下载第一个升级文件，**$MYMINISYS ：1，2**代表处于最小系统的第二个阶段；

3.升级完成后自动退出最小系统，使用**AT$MYMINISYS?**查询状态，返回**$MYMINISYS ：0**是正常运行的非最小系统；

## 注意事项

1.Flash中的fotaparam区域要求必须要等于或者大于12K，最好设置为12K；

2.Flash中不需要预留fotapkg区域；

3.系统需要支持可连网的最小系统；

# 新的OTA方案（PRO 版本）

## Pro升级方案介绍

由于从SDK008开始之后的版本，image压缩采用LZMA压缩，如果继续使用第2章中的全系统差分方案，那么差分文件会很大，PRO版本是全系统差分方案的升级版本。

## 使用工具adiff.exe制作差分包

**adiff.exe -p system\_old.img system\_new.img system\_patch.bin**

**<Options>**

**[-a1 <app1\_partition\_name> <old app1\_file\_name> <new app1\_file\_name>]**

**[-a2 <app2\_partition\_name> <old app2\_file\_name> <new app2\_file\_name>]**

**[-l fs|fp: the diff file in file system or fota\_pkg, default is fota\_pkg]**

前四个参数是必选项， –a1是可选项， –a2是可选项

**-p** 参数代表使用PRO方案进行差分包制作

**system\_old.img**参数是板子中已经烧写的system文件, 如1601升级包system.img重命名为system\_old.img;

**system\_new.img**参数是要升级到的system文件, 如1601升级包system.img重命名为system\_new.img;

**system\_patch.bin**参数是生成的升级文件名称；

**-a1**代表当系统中存在和system同级的客户分区app1，如下图所示，分区app1\_partition\_name名称假设为app1，当要升级这个分区时可以使用这个参数，并将要升级前后的old app1\_file\_name和new app1\_file\_name文件放在**adiff.exe**同一路径下；

（app1分区名称客户可以在aboot工具下修改，这种分区主要是给客户增加一些特殊的分区，可以随着system.img一起差分升级，使整个系统升级更灵活）

**app1\_partition\_name**代表分区的分名称；

**old app1\_file\_name**代表升级的旧分区文件；

**new app1\_file\_name**代表升级的新分区文件；

**-a2**同上**a1**，升级最多支持同时升级两个app分区；

**-l fs|fp代表差分文件存放位置，选择fp，代表差分文件将会放在fota\_pkg分区；选择fs，代表差分文件将会放在文件系统中；通过这个指定，工具可以检测出升级条件是否能够满足，如果不满足工具将不能制作差分包。**

## 升级包的下载步骤

同第2章升级方案下载步骤相同

## 注意事项

1.Flash中的fotaparam区域要求必须要等于或者大于12K，最好设置为12K；

2.Flash中必须要预留fotapkg分区，当差分包放在fotapkg中，fotapkg必须要大于2M；如果差分包放在文件系统中，fotapkg必须要大于1M；