

使用指南

CMT453x 固件升级使用指南

简介

本文档将介绍CMT453x 固件升级的原理和应用例程，帮助用户了解固件升级的流程，便于快速开发。

CMT453x固件升级功能具有以下特点：

- 支持串口升级固件。
- 支持蓝牙升级固件。
- 支持蓝牙MTU 20-244字节。
- 为保证安全，蓝牙升级使用ECC数字签名校验固件合法性。
- 为加快升级的速度，和支持版本可回退的功能，我们提供蓝牙“双Bank”升级。
- 提供蓝牙“单Bank”升级方案，以满足用户程序占用空间较大的情况。
- 系统自动选择“双Bank”升级还是“单Bank”升级，用户无需判断。
- 升级速度可以通过手机APP调节，以便兼容各厂商手机蓝牙。

目录

简介	1
1 实例演示	4
1.1 JLINK烧录演示	4
1.2 串口升级演示	7
1.3 蓝牙“双BANK”升级演示	9
1.4 蓝牙“单BANK”升级演示	12
2 FLASH内存分布	14
3 数据结构	16
3.1 BOOTSETTING.....	16
4 升级流程	21
4.1 串口升级流程	21
4.2 蓝牙双BANK升级流程.....	21
4.3 蓝牙单BANK升级流程.....	22
5 升级命令	23
5.1 串口升级命令	23
5.1.1 Usart dfu.....	23
5.1.2 Ping.....	24
5.1.3 Init packet.....	24
5.1.4 Packet header.....	24
5.1.5 Packet	25
5.1.6 Postvalidate	25
5.1.7 Activate&Reset	25
5.2 蓝牙升级命令	26
5.2.1 更新BLE连接间隙命令	29
5.2.2 更新BLE MTU 命令	29
5.2.3 查询版本号和升级方式命令.....	29
5.2.4 创建发送dfu_setting 和签名文件命令.....	30
5.2.5 创建发送固件数据命令	30
5.2.6 FLASH固件整体校验命令.....	31
5.2.7 激活分区表和复位命令	31
5.2.8 跳入ImageUpdate	31
5.3 错误代码	32
6 工具讲解	33
6.1 JLINK工具	33
6.2 HPUTIL工具	33
6.3 ANDROIDUTIL工具	33
7 例程讲解	34
7.1 MASTERBOOT讲解	34
7.2 APPUSART讲解	36

7.3 APPOTA讲解	38
7.4 IMAGEUPDATE讲解	41
8 加密讲解	43
9 版本历史	44
10 声明	45

1 实例演示

1.1 JLINK烧录演示

进入.\CMT453x_SDK\utilities\dfu\Image\JLINKProgrammingDemo目录。

双击JLINKProgramming.bat文件，查看命令行窗口输出。

第一步：HPUtl工具生成BootSetting.bin，命令行显示Bootsetting created successfully!说明文件生成成功。

```
=====BootSetting.bin=====
00000000: 66 01 47 D3 FF FF FF FF 00 40 00 01 60 54 00 00 f.G.....@..`T..
00000010: C5 16 ED C4 01 00 00 00 01 00 00 00 FF FF FF FF .....
00000020: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000030: 00 00 02 01 2C 55 00 00 C0 68 7B 9D 01 00 00 00 ....,U...h{.....
00000040: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000050: FF FF FF FF FF FF FF FF 00 C0 03 01 24 39 00 00 .....$9..
00000060: A2 44 D9 6D 01 00 00 00 FF FF FF FF FF FF FF FF .D.m.....
00000070: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000080: AE 1C 41 A5 F4 35 DD 3D 89 C8 00 D8 0F 8D 2A C2 ..A..5.=.....*.
00000090: 63 3A 02 37 24 5D 2D DB F0 46 A1 6A 5E 43 26 44 c:..7$]-..F.j^C&D
000000A0: 73 20 7D 16 86 EA 41 6B A3 8D 0D 60 DA 61 CD 98 s }...Ak...`.a..
000000B0: 53 D5 22 A5 14 6A EE 64 BB B4 7E 40 39 A6 B5 29 S.."..j.d...~@9..)
Bootsetting created successfully!
SEGGER J-Link Commander V6.32 (Compiled Apr 20 2018 17:25:19)
DLL version V6.32, compiled Apr 20 2018 17:25:02
```

第二步：JLink连接并检测设备

```
Connecting to target via SWD
Found SW-DP with ID 0x0BB11477
Scanning AP map to find all available APs
AP[1]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x04770021)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FF000
CPUID register: 0x410CC200. Implementer code: 0x41 (ARM)
Found Cortex-M0 r0p0, Little endian.
FPUnit: 4 code (BP) slots and 0 literal slots
CoreSight components:
ROMTbl[0] @ E00FF000
ROMTbl[0][0]: E000E000, CID: B105E00D, PID: 000BB008 SCS
ROMTbl[0][1]: E0001000, CID: B105E00D, PID: 000BB00A DWT
ROMTbl[0][2]: E0002000, CID: B105E00D, PID: 000BB00B FPB
Cortex-M0 identified.
Reset delay: 0 ms
Reset type NORMAL: Resets core & peripherals via SYSRESETREQ & VECTRESET bit.
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
```

第三步：JLink擦除芯片Flash。

```
Erasing device (CMT4531)...  
J-Link: Flash download: Total time needed: 1.278s  
Restore: 0.006s)  
Erasing done.
```

第三步：烧录MasterBoot.bin到芯片Flash。

```
Downloading file [Image\MasterBoot.bin]...  
J-Link: Flash download: Bank 0 @ 0x01000000: 1 range affected (8192 bytes)  
J-Link: Flash download: Total time needed: 1.109s (Prepare: 0.051s, Compare: 0.138s,  
Restore: 0.021s)  
O.K.
```

第四步：烧录Bootsetting.bin到芯片Flash。

```
Downloading file [Image\Bootsetting.bin]...  
J-Link: Flash download: Bank 0 @ 0x01000000: 1 range affected (4096 bytes)  
J-Link: Flash download: Total time needed: 0.790s (Prepare: 0.049s, Compare: 0.139s,  
Restore: 0.023s)  
O.K.
```

第五步：烧录APP1.bin到芯片Flash。

```
Downloading file [Image\APP1.bin]...  
J-Link: Flash download: Bank 0 @ 0x01000000: 1 range affected (24576 bytes)  
J-Link: Flash download: Total time needed: 2.449s (Prepare: 0.052s, Compare: 0.138s,  
Restore: 0.022s)  
O.K.
```

第六步：烧录APP2.bin到芯片Flash。

```
Downloading file [Image\APP2.bin]...  
J-Link: Flash download: Bank 0 @ 0x01000000: 1 range affected (24576 bytes)  
J-Link: Flash download: Total time needed: 2.509s (Prepare: 0.053s, Compare: 0.138s,  
Restore: 0.022s)  
O.K.
```

第七步：烧录ImageUpdate.bin到芯片Flash。

```
Downloading file [Image\ImageUpdate.bin]...  
J-Link: Flash download: Bank 0 @ 0x01000000: 1 range affected (16384 bytes)  
J-Link: Flash download: Total time needed: 1.849s (Prepare: 0.049s, Compare: 0.139s,  
Restore: 0.023s)  
O.K.
```

第八步：复位芯片。

```
Reset delay: 0 ms
Reset type NORMAL: Resets core & peripherals via SYSRESETREQ & VECTRESET bit.
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
```

JLINK依次执行，擦除，写入MasterBoot.bin, Bootsetting.bin, APP1.bin, APP2.bin, ImageUpdate.bin, 复位命令，完成批量烧录工作，如果烧录不了，可能是因为芯片进入DEEP SLEEP，可以尝试先执行JLINKProgramming.bat，再给芯片上电。

MasterBoot.bin

由[MasterBoot Keil例程](#)生成，编译完成之后，系统将会自动将所生成的固件复制到dfu\image目录下，具体可查看例程设置 keil → option for target → User → Run #2。

Bootsetting.bin

由HPUtil python 工具生成，[bootsetting数据结构查看第三章第一节](#)。

APP1.bin和APP2.bin

由[AppOTA例程](#)生成，但生成APP1.bin和APP2.bin所选的程序起始地址需要分别相应的调整，其起始地址值分别为0x1004000和0x1020000，具体请参考[FLASH内存分布](#)。生成路径查看例程 keil → option for target → User → Run #2。

ImageUpdate.bin

由[ImageUpdater例程](#)生成，生成路径查看例程 keil → option for target → User → Run #2。

Bin文件的运行地址和功能描述在[第二章](#)有详细介绍。

使用任意文本工具打开JLINKProgramming.bat文件。

```
set JLink_path=.\JLink\JLink_V632\JLink.exe
set JLink_script_path=.\JLink\JLink_Script\download.jlink
set HPUtil_path=.\HPUtil\HPUtil.exe

::Creating bootsetting file
::bootsetting.bin path
set output_bootsetting=.\Image\bootsetting.bin
::bank1 parameters, nonoptional
set bank1_start_address=0x1004000
set bank1_version=0x00000001
set bank1_bin=.\Image\APP1.bin
set bank1_activation=yes
::bank2 parameters, optional
set bank2_start_address=0x1020000
set bank2_version=0x00000001
set bank2_bin=.\Image\APP2.bin
set bank2_activation=no
::ImageUpdate parameters, optional
set image_update_start_address=0x0103C000
set image_update_version=0x00000001
::set image_update_bin=.\Image\ImageUpdate.bin
set image_update_activation=no
::ImageUpdate parameters, optional
set public_key_file=.\keys\public_key.bin
```

用户可以修改bank1_activation=no, bank2_activation=yes, 以设置上电默认启动bank2的程序。可以通过串口接芯片PB1（115200 8N1）查看打印看效果，也可以通过LED1和LED2闪烁频率看效果，APP1以100毫秒闪烁，APP2以500毫秒闪烁，蓝牙搜索相应的广播名称HOPERF_APPx (x分别为1, 2)。

用户还可以修改bank1_activation=no, image_update_activation=yes, 看串口打印效果，或者蓝牙搜索ImageUpdate设备广播看效果。

需要注意的是，activation只有一个程序可以使能。

1.2 串口升级演示

进入CMT453x_SDK\utilities\dfu\Image\UartProgrammingDemo目录。

双击JLINKProgramming.bat文件，查看命令行窗口输出。

从[Bootsetting.bin](#)可以看出，芯片中只有bank1和bank2有程序，且bank1程序激活。

```
=====BootSetting.bin=====
00000000: F0 53 32 7C FF FF FF FF 00 40 00 01 9C 0F 00 00 .S2|.....@.....
00000010: FC E8 69 E6 01 00 00 00 01 00 00 00 FF FF FF FF ..i.....
00000020: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000030: 00 00 02 01 9C 0F 00 00 E3 47 01 4E 01 00 00 00 .....G.N....
00000040: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000050: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000060: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000070: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000080: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000090: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
000000A0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
000000B0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
Bootsetting created successfully!
```

运行`JLINKProgramming.bat`脚本，通过JLink将MasterBoot.bin, Bootsetting.bin, APP1.bin和APP2.bin依次烧录至设备。

其APP1.bin和APP2.bin由[AppUsart例程](#)生成，生成路径查看例程 keil → option for target → User → Run #2。

```
Downloading file [Image\MasterBoot.bin]...
J-Link: Flash download: Bank 0 @ 0x01000000: 1 range affected (8192 bytes)
J-Link: Flash download: Total time needed: 1.113s (Prepare: 0.046s, Compare: 0.139s,
ore: 0.022s)
O.K.

Downloading file [Image\Bootsetting.bin]...
J-Link: Flash download: Bank 0 @ 0x01000000: 1 range affected (4096 bytes)
J-Link: Flash download: Total time needed: 0.789s (Prepare: 0.051s, Compare: 0.138s,
ore: 0.022s)
O.K.

Downloading file [Image\APP1.bin]...
J-Link: Flash download: Bank 0 @ 0x01000000: 1 range affected (4096 bytes)
J-Link: Flash download: Total time needed: 0.946s (Prepare: 0.048s, Compare: 0.139s,
ore: 0.023s)
O.K.

Downloading file [Image\APP2.bin]...
J-Link: Flash download: Bank 0 @ 0x01000000: 1 range affected (4096 bytes)
J-Link: Flash download: Total time needed: 0.937s (Prepare: 0.047s, Compare: 0.138s,
ore: 0.021s)
O.K.
```

文本工具打开`UartFirmwareUpdate.bat`文件。


```
set HPUTil_path=..\..\HPUTil\HPUTil.exe
:: APP.bin is the update firmware
:: --app_start_address is the address of update firmware
:: --app_version is the version of update firmware
:: --serial_port is the serial port number
set app_bin=.\Image\APP2.bin
set app_start_address=0x1020000
set app_version=0x01020304
set serial_port=COM5
set serial_baudrate=115200

%HPUTil_path% ius serial %app_bin%
--app_start_address %app_start_address%
--app_version %app_version%
--serial_port %serial_port%
--serial_baudrate %serial_baudrate%
--force_update false
```

修改serial_port=串口号（通过查看设备管理器得到），USB转串口线接芯片PB6（芯片TX）和PB7（芯片RX），保存关闭。

双击UartFirmwareUpdate.bat文件，查看命令行窗口打印结果。

```
D:\SDKs\CMT\CMT453x_SDK_v1.3.1_alpha\CMT453x_SDK_v1.3.1\utilities\dfu\Image\UartProgrammingDemo>echo off
baudrate --> 115200
Serial Image Update In Progress [#####] 100%
3996 bytes Image Update Complete in 2.06s
```

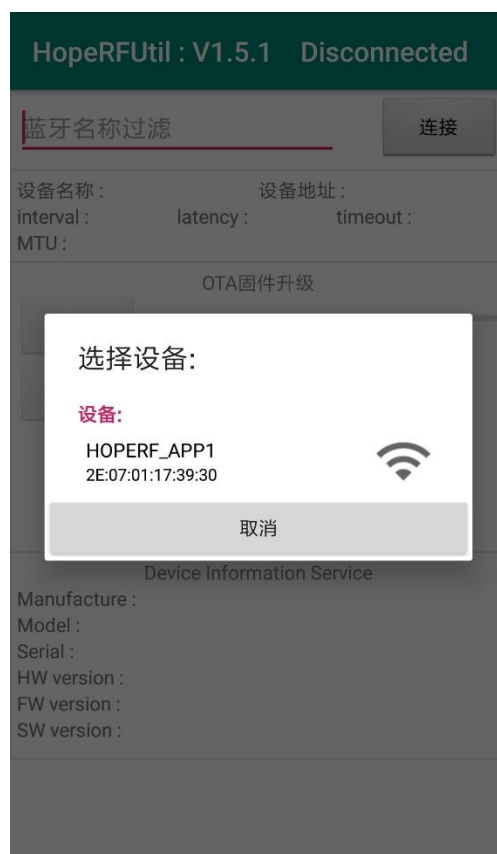
JLINK默认烧录bank1程序执行（PB0脚会被拉高），串口默认升级到bank2程序执行（PA6脚会被拉高），用户也可以修改app_bin=.\Image\APP1.bin，app_start_address=0x01004000，使串口默认升级bank1程序执行。

1.3 蓝牙“双Bank”升级演示

进入 CMT453x_SDK\utilities\dfu\HPAndroidUtil\ 目录，手机安装 AndroidUtil.apk 文件。再进入 CMT453x_SDK\utilities\dfu\Image\DualBankUpdateDemo\目录，双击JLINKProgramming.bat，烧录固件到芯片。

再进入CMT453x_SDK\utilities\dfu\Image\DualBankUpdateDemo\Image\目录，将ota_dual_bank.zip文件拷贝到手机内部存储设备上。

点击连接，再点击选择HOPERF_APP1设备

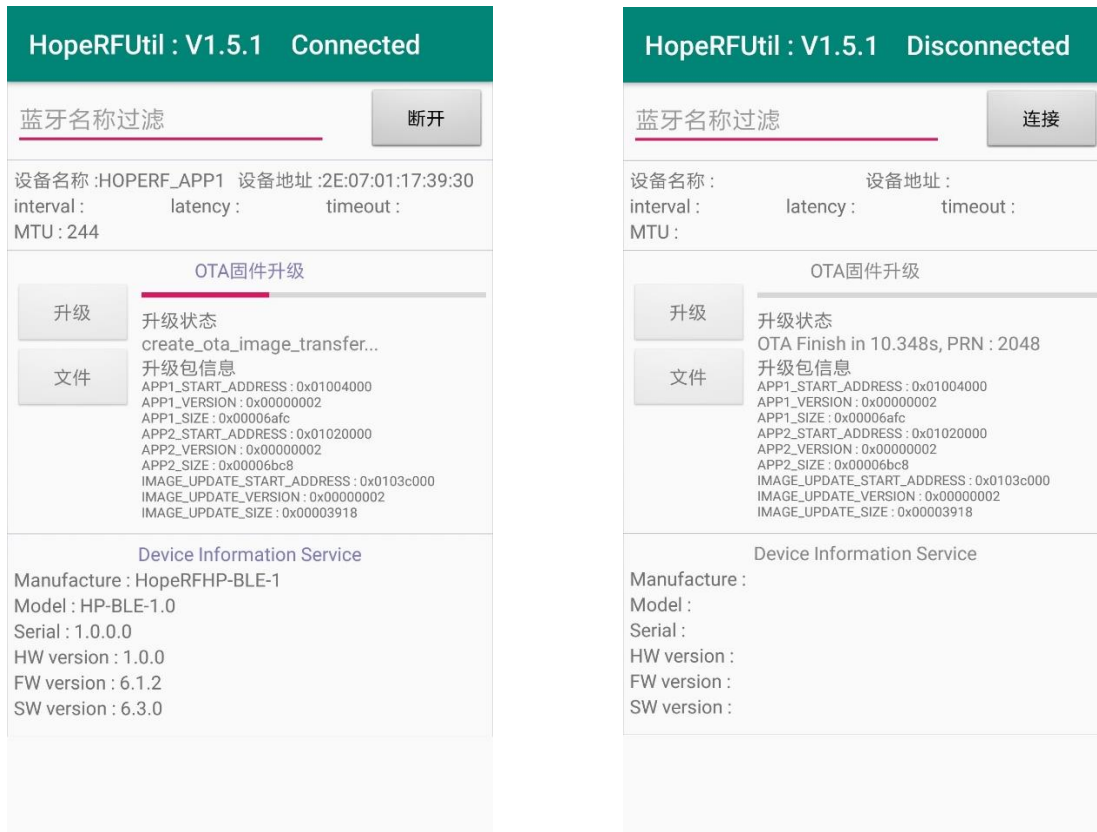


等待右上角蓝牙状态变成Connected，再点击文件，选择ota_dual_bank.zip文件。

升级包信息中可以查看到升级包的大小，点击升级，观察升级状态改变，进度条更新。



升级完成之后，蓝牙将自动断开，升级状态显示升级耗时。



升级过程中所使用的蓝牙升级包ota_dual_bank.zip，用户可能通过双击GenerateUpdateImage.bat生成，观察命令行窗口信息。

```
=====dfu_setting.dat=====
00000000: 41 68 05 13 00 40 00 01 FC 6A 00 00 62 72 6D 14 Ah...@...j..brm.
00000010: 02 00 00 00 00 00 02 01 C8 6B 00 00 D3 91 E2 E6 .....k.....
00000020: 02 00 00 00 00 C0 03 01 18 39 00 00 58 B7 27 2F .....9..X.'/
00000030: 02 00 00 00 C0 CA EE 5A 79 70 69 0E 29 80 B3 3A .....Zypi.)..:
00000040: 85 6C 3A 63 F1 34 6B 85 3B 9F 74 AC BE 33 F3 4D .l:c.4k.;.t..3.M
00000050: 23 D7 36 5B 0A B8 8D 0A C1 55 DD 9B D0 75 FE B5 #.6[.....U...u..
00000060: CB 14 69 0F 0C 4E 13 17 5B 34 06 47 00 19 D9 26 ..i..N..[4.G...&
00000070: 3A 36 40 9A :6@.

=====config.txt=====
APP1_START_ADDRESS : 0x01004000
APP1_VERSION : 0x00000002
APP1_SIZE : 0x00006afc
APP2_START_ADDRESS : 0x01020000
APP2_VERSION : 0x00000002
APP2_SIZE : 0x00006bc8
IMAGE_UPDATE_START_ADDRESS : 0x0103c000
IMAGE_UPDATE_VERSION : 0x00000002
IMAGE_UPDATE_SIZE : 0x00003918
```

该批处理文件会按照文件内配置的参数和Image文件夹下的bin文件，制作蓝牙升级包，命令行窗口同时显示dfu_setting.dat文件数据，和Config.txt文件数据。

1.4 蓝牙“单Bank”升级演示

进入CMT453x_SDK\utilities\dfu\HPAndroidUtil\目录，手机安装AndroidUtil.apk文件。

再进入CMT453x_SDK\utilities\dfu\Image\SingleBankProgrammingDemo\目录，双击JLINKProgramming.bat，烧录固件到芯片。

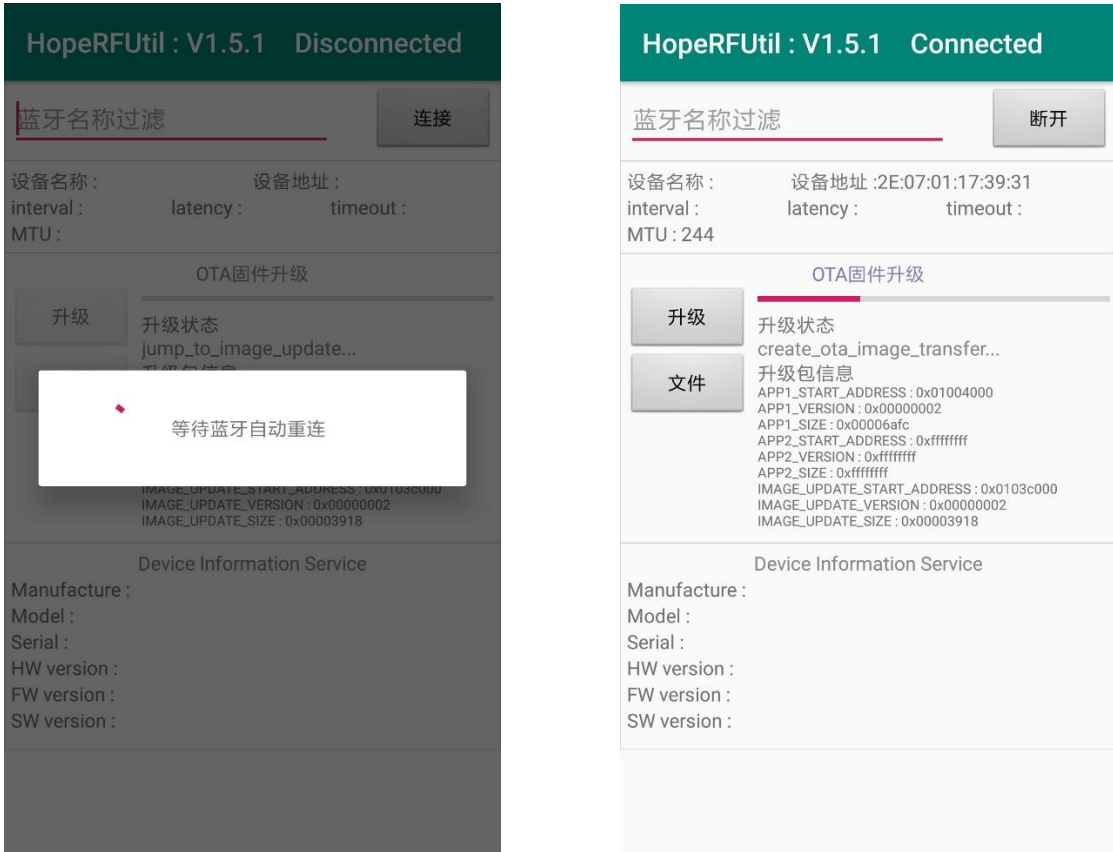
再进入CMT453x_SDK\utilities\dfu\Image\SingleBankProgrammingDemo\Image\目录，将ota_single_bank.zip文件拷贝到手机内部存储设备上。

点击连接，再点击选择HOPERF_APP1设备。

等待右上角蓝牙状态变成Connected，再点击文件，选择ota_single_bank.zip文件。升级包信息中可以查看到升级包的大小。



点击升级，观察升级状态改变，进度条更新。单bank升级过程中，蓝牙会断开重连一次，界面会弹出对话框要用户等待蓝牙自动重连。



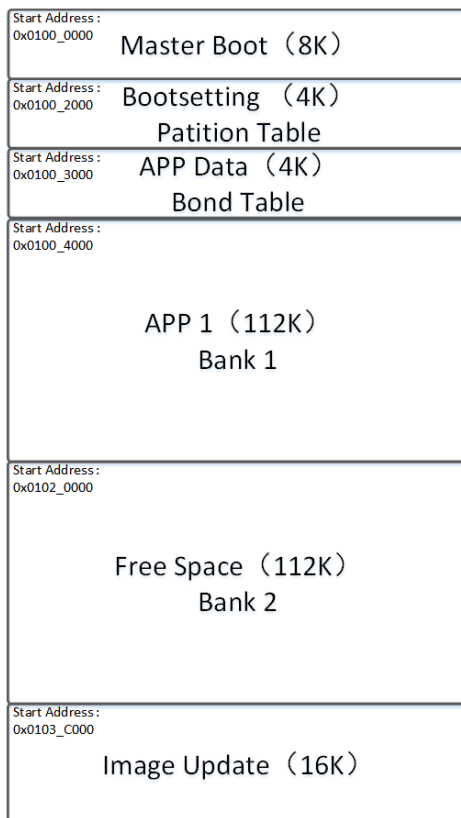
蓝牙重连后升级继续，升级结束显示升级耗时。



2 FLASH内存分布

CMT453x芯片FLASH地址范围 0x0100_0000 - 0x0103_FFFF，可用空间 256K字节。

程序和数据FLASH分布如下图：



名称	CMT453x FLASH地址	说明
Master Boot (程序)	0x0100_0000 - 0x0100_1FFF (8K)	上电后程序入口； 读取分区表，程序跳转； 串口升级功能；
Bootsetting (数据) (Patition Table)	0x0100_2000 - 0x0100_2FFF (4K)	FLASH分区表；
APP Data (数据) (Bond Table) (User Data)	0x0100_3000 - 0x0100_3FFF (4K)	APP数据存储区； 存储绑定列表（5个设备约500字节）； 剩余空间存储用户自定义数据；
APP 1 (程序) (Bank 1)	0x0100_4000 - 0x0101_FFFF (112K)	用户程序1存储区域；
Free Space/APP2 (程序) (Bank 2)	0x0102_0000 - 0x0103_BFFF (112K)	预留空间； 或者用户程序2存储区域；
Image Update (程序)	0x0103_C000 - 0x0103_FFFF (16K)	预留空间； 或者“单BANK”升级使用；

MasterBoot (程序)

提供程序跳转入口功能，和串口升级功能。

Bootsetting (数据)

提供程序跳转，和升级共享数据。

APP Data (数据)

用户数据存储区域，如果需要更大空间，建议使用外接存储器，或者修改FLASH内存分布（具体修改方法请联系技术支持）。

APP1 (程序)

用户编译在Bank1的应用程序，编译时需要指定正确的起始地址。

APP2 (程序)

用户编译在Bank2的应用程序，编译时需要指定正确的起始地址。

ImageUpdate (程序)

“单bank”升级使用程序。ImageUpdate是一个最小的支持蓝牙OTA的程序，它将接收主机端所发送的更新固件，并写入应用程序所在的flash区域以完成程序更新。

3 数据结构

3.1 Bootsetting

大小（字节）	名称	说明
4	Bootsetting CRC	Bootsetting以下数据校验值
4	MasterBoot Force Update	强制MasterBoot串口升级 1： 串口升级 默认： 0xFFFFFFFF
4*10	Bank 1 分区	4字节start address表示Bank1分区应用程序起始地址 4字节size程序大小 4字节crc程序校验值 4字节version程序版本号 4字节activation程序激活码： 1激活， 其它 4*5字节reserve保留
4*10	Bank 2 分区	4字节start address表示Bank2分区应用程序起始地址 4字节size程序大小 4字节crc程序校验值 4字节version程序版本号 4字节activation程序激活码： 1激活， 其它 4*5字节reserve保留
4*10	Image Update 分区	4字节start address表示ImageUpdate程序的起始地址 4字节size程序大小 4字节crc程序校验值 4字节version程序版本号 4字节activation程序激活码： 1激活， 其它 4*5字节reserve保留

64	公开密钥	用于ECC签名验证。
----	------	------------

Bootsetting crc = CRC32 (MasterBoot Force Update + Bank 1 分区 + Bank 2 分区 + Image Update 分区 + 公开密钥)

MasterBoot Force Update: MasterBoot程序判断这个变量，决定是否进入串口升级模式。

Bank 1 分区: 记录APP1程序的起始地址，程序大小，程序CRC32校验值，以及程序的激活状态。

Bank 2 分区: 记录APP2程序的起始地址，程序大小，程序CRC32校验值，以及程序的激活状态。

Image Update 分区: 记录Image Update程序的起始地址，程序大小，程序CRC32校验值，以及程序的激活状态。

公开密钥: HPUTIL工具生成，升级验签使用。

reserve: 保留字段，用于扩展。

在代码中，结构体定义如下:

```
typedef struct
{
    uint32_t start_address;
    uint32_t size;
    uint32_t crc;
    uint32_t version;
    uint32_t activation;
    uint32_t reserve[5];
}HP_Bank_t;

typedef struct
{
    uint32_t crc;
    uint32_t master_boot_force_update;
    HP_Bank_t app1;
    HP_Bank_t app2;
    HP_Bank_t ImageUpdate;
    uint8_t public_key[64];
}HP_Bootsetting_t;
```

3.2 init packet

串口升级Init packet使用

大小 (字节)	名称	说明
---------	----	----

4	CRC	数据校验值
4	App_start_address	新固件首地址
4	App_size	新固件大小（单位字节）
4	App_crc	新固件校验值
4	App_version	新固件版本
4*10	Reserve	保留

在代码中，结构体定义如下：

```
typedef struct
{
    uint32_t crc;
    uint32_t app_start_address;
    uint32_t app_size;
    uint32_t app_crc;
    uint32_t app_version;
    uint32_t reserve[10];
}_init_pkt;
```

3.3 dfu_setting

HPUTIL.exe制作升级包自动生成，蓝牙升级中使用，用于对升级固件的签名验证，和升级固件的完整性校验。

大小（字节）	名称	说明
4	CRC	DFU_SETTING数据校验值
4*4	APP 1 参数	4字节start address程序起始地址 4字节size程序大小 4字节crc程序校验值 4字节version程序版本号
4*4	APP 2 参数	4字节start address程序起始地址 4字节size程序大小 4字节crc程序校验值 4字节version程序版本号
4*4	Image Update 参数	4字节start address程序起始地址 4字节size程序大小 4字节crc程序校验值 4字节version程序版本号
64	Signature	HASH数据+私钥生成签名文件

CRC = CRC32（APP 1 参数 + APP 2 参数 + Image Update 参数 + Signature）

APP 1 参数：升级固件APP1程序的起始地址，程序大小，程序CRC，以及程序版本号。

APP 2 参数：升级固件APP1程序的起始地址，程序大小，程序CRC，以及程序版本号。

Image Update 参数：升级固件Image Update程序的起始地址，程序大小，程序CRC，以及程序版本号。

Signature = ECC_ECDSA_SHA256_NIST256P（APP 1 参数 + APP 2 参数 + Image Update 参数）。

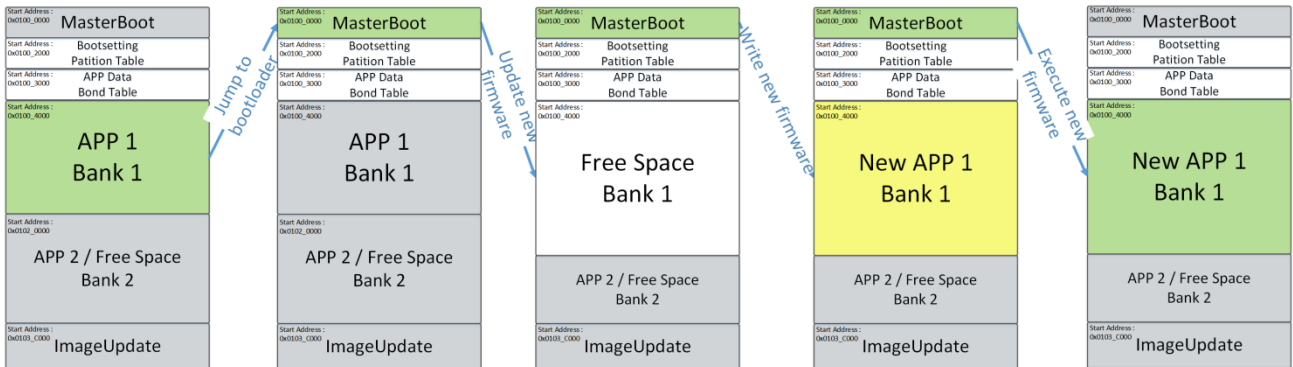
在代码中，结构体定义如下：

```
typedef struct{
    uint32_t start_address;
    uint32_t size;
    uint32_t crc;
    uint32_t version;
```

```
}Dfu_setting_bank_t;  
  
typedef struct{  
  
    uint32_t crc;  
    Dfu_setting_bank_t app1;  
    Dfu_setting_bank_t app2;  
    Dfu_setting_bank_t image_update;  
    uint8_t signature[64];  
}Dfu_setting_t;
```

4 升级流程

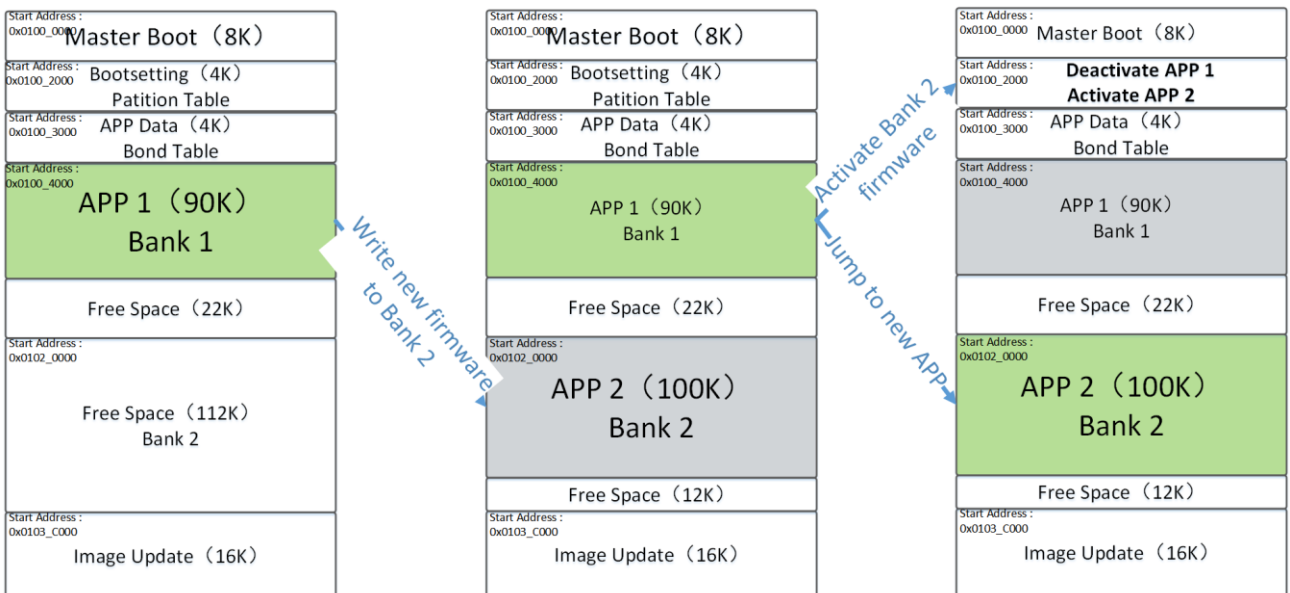
4.1 串口升级流程



用户的应用程序将Bootsetting数据中的MasterBootForceUpdate变量置位，软件复位，进入MasterBoot程序。MasterBoot程序检查MasterBoot Force Update被置位，进入串口升级流程。

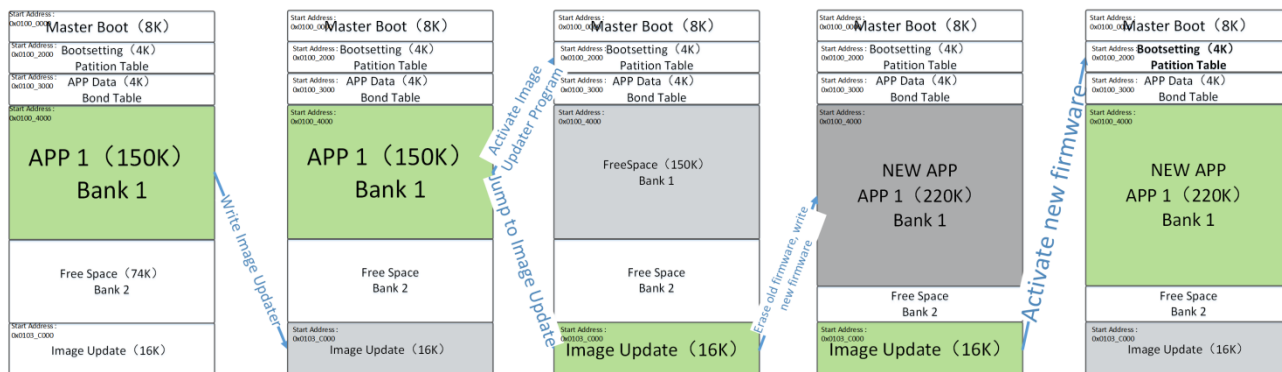
MasterBoot通过串口接收新固件，擦除bank 1区域原有固件，并写入新固件，全部固件写完，激活新固件，最后跳入新固件执行。

4.2 蓝牙双bank升级流程



“双bank”升级采用APP1升级APP2，或者APP2升级APP1的方法，升级速度更快，更稳定。但是，用户程序只能使用一半FLASH区域，制作升级包时需要多生成一个bank2的bin文件。

4.3 蓝牙单bank升级流程



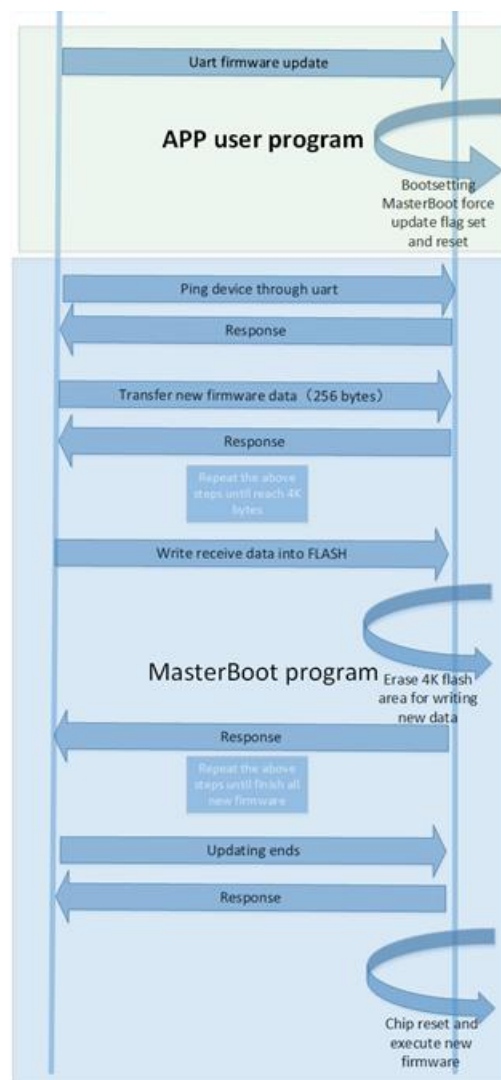
“单bank”升级采用ImageUpdate程序升级APP1程序，优点：用户程序可以使用全部FLASH区域，缺点：升级速度慢，因为蓝牙需要断开重连导致连接不稳定，以及升级失败没有备份。

5 升级命令

5.1 串口升级命令

以下是串口升级的流程图：

1. 首先PC上位机通过发送“USART DFU”命令至设备端要求其进入Masterboot串口升级模式。Masterboot也可能通过其他方式，例如在设备端判断外部按键，来进入Masterboot串口升级模式。从机在收到上位机发送的“USART DFU”命令后，将Bootsetting中的“Master Boot Force Update”置1，并复位芯片。
2. 芯片复位后将执行Master Boot段代码，因为“Master Boot Force Update”置1，程序将等待上位机通过串口发送命令以执行下一步的升级动作。



接下来具体介绍串口升级过程中所涉及的命令。

5.1.1 USART DFU

PC上位机命令芯片进入串口升级模式，芯片会复位进入MasterBoot串口升级模式。

名称	大小（字节）	数值	说明
串口头	1	0xAA	（PC->芯片）
命令号	1	0x07	强制进入串口升级
参数	3	0x01,0x02,0x03	防止误判

名称	大小（字节）	数值	说明
串口头	1	0xAA	（芯片->PC）
响应号	1	0x07	

5.1.2 Ping

名称	大小（字节）	数值	说明
串口头	1	0xAA	（PC->芯片）
命令号	1	0x01	PC上位机尝试与芯片通信

名称	大小（字节）	数值	说明
串口头	1	0xAA	（芯片->PC）
响应号	1	0x01	

5.1.3 Init packet

名称	大小（字节）	数值	说明
串口头	1	0xAA	（PC->芯片）
命令号	1	0x02	整个新固件头文件
INIT PACKET	Init packet 大小		详情见：init packet

名称	大小（字节）	数值	说明
串口头	1	0xAA	（芯片->PC）
响应号	1	0x02	
错误码	1		详情见：错误代码

5.1.4 Packet header

名称	大小（字节）	数值	说明
串口头	1	0xAA	（PC->芯片）
命令号	1	0x03	一包数据头文件
OFFSET	4		当前升级文件偏移位置
SIZE	4		将要发送升级数据大小
CRC	4		将要发送升级包数据校验值

名称	大小（字节）	数值	说明
----	--------	----	----

串口头	1	0xAA	(芯片->PC)
响应号	1	0x03	
错误码	1		详情见：错误代码

5.1.5 Packet

名称	大小（字节）	数值	说明
串口头	1	0xAA	(PC->芯片)
命令号	1	0x04	升级包数据
数据	<=256-3		按照包头偏移的升级包数据

名称	大小（字节）	数值	说明
串口头	1	0xAA	(芯片->PC)
响应号	1	0x04	
错误码	1		详情见：错误代码

5.1.6 Postvalidate

名称	大小（字节）	数值	说明
串口头	1	0xAA	(PC->芯片)
命令号	1	0x05	校验接收数据

名称	大小（字节）	数值	说明
串口头	1	0xAA	(芯片->PC)
响应号	1	0x05	
错误码	1		详情见：错误代码

5.1.7 Activate&Reset

名称	大小（字节）	数值	说明
串口头	1	0xAA	(PC->芯片)
命令号	1	0x06	激活新固件，软件复位

名称	大小（字节）	数值	说明
串口头	1	0xAA	(芯片->PC)
响应号	1	0x06	
错误码	1		详情见：错误代码

5.2 蓝牙升级命令

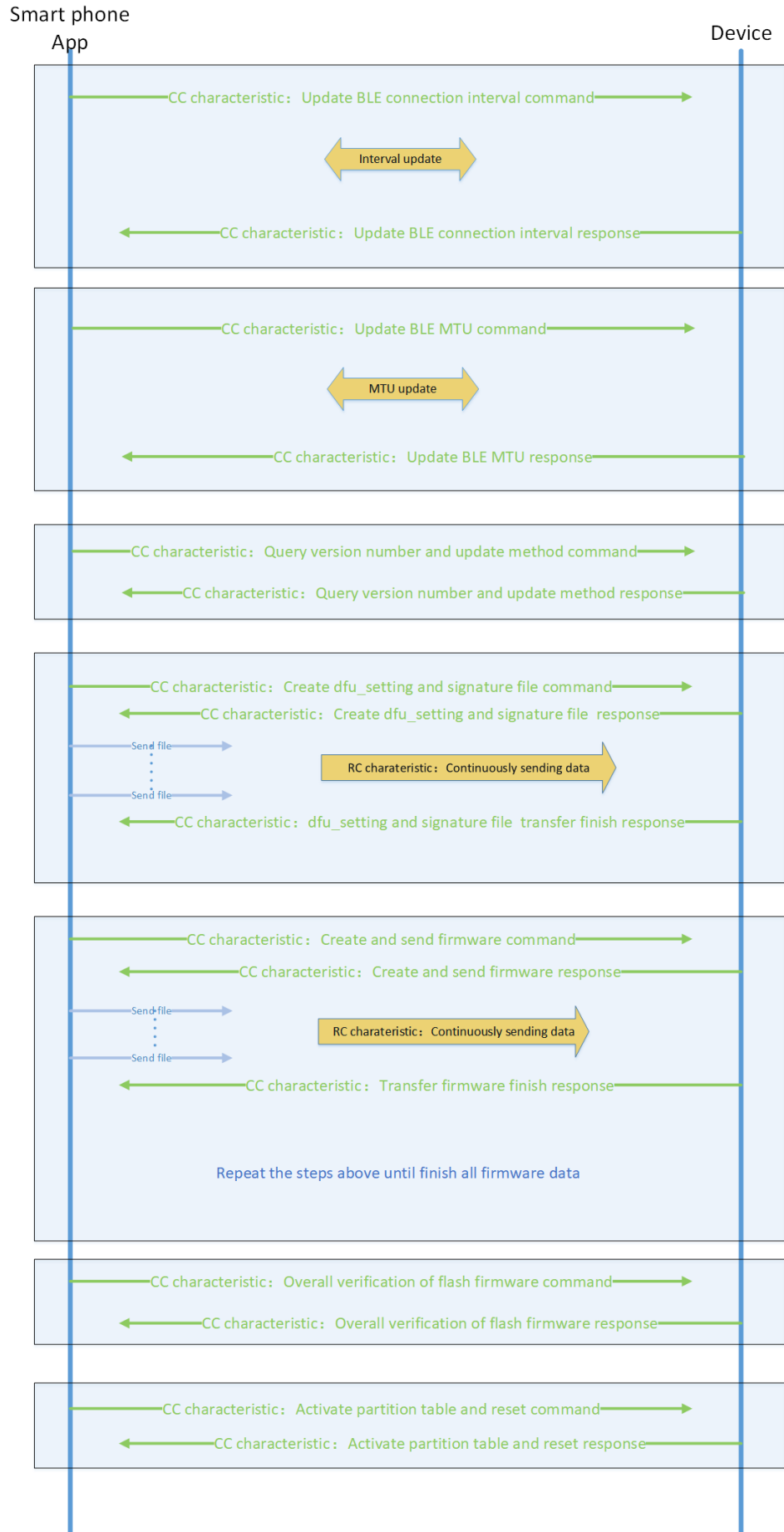
手机端通过Command Characteristic (CC) 特征值发送命令，并通过CC特征值接收响应。

升级包数据流通过RC特征发送给设备。

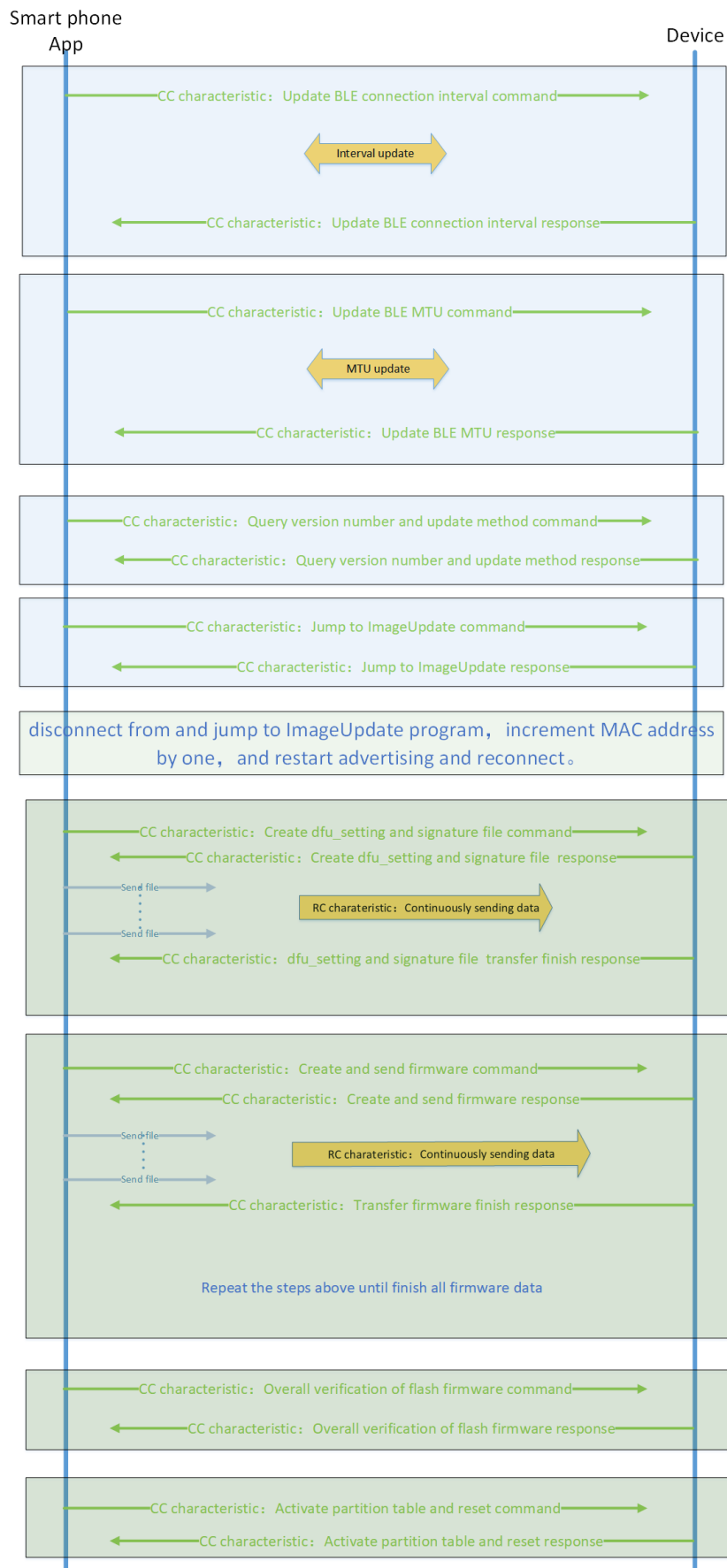
- 服务和特征值

类型	名称	UUID（16进制）	属性	MTU	功能说明
服务	IUS(Image Update Service)	11-11-11-11-11-11-11-11-11-11-11-00-01-11-11	Primary		固件升级服务
特征	RC(Receive Characteristic)	11-11-11-11-11-11-11-11-11-11-11-00-02-11-11	Write Without Response	20-244	固件接收特征
特征	CC(Command Characteristic)	11-11-11-11-11-11-11-11-11-11-11-00-03-11-11	Notify, Write	20	命令收发特征

- 蓝牙双bank升级



- 单bank升级



5.2.1 更新BLE连接间隙命令

- 升级时减小BLE连接间隔，加快BLE传输速度，手机将连接间隔参数发给从设备，由从设备发起连接间隔参数更新命令。
- 原因是：不确定Android和IOS是否开放了连接间隔参数更新的上层命令。
- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
命令号	1	1	（手机->设备）
最小连接间隙	2		单位： 1.25 MS
最大连接间隙	2		单位： 1.25 MS
SLAVE LATENCY	2		从设备少响应个数
连接超时	2		单位： 10 MS

- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
响应号	1	1	（设备->手机）
错误码	1		详情见：错误代码

5.2.2 更新BLE MTU命令

- 加大RC特征MTU，手机将新MTU大小发给从设备，由从设备发起MTU更新命令。
- 原因是：不确定IOS是否开放了MTU更新的上层命令。
- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
命令号	1	2	（手机->设备）
新MTU大小	2		根据手机型号定义

- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
响应号	1	2	（设备->手机）
错误码	1		详情见：错误代码

5.2.3 查询版本号和升级方式命令

- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
命令号	1	3	（手机->设备）
新APP1固件大小	4		单位（字节）
新APP1固件大小	4		单位（字节）
新IMAGE UPDATE固件大小	4		单位（字节）
新IMAGE UPDATE固	4		

件版本			
-----	--	--	--

- 命令格式：（CC特征）

名称	大小（字节）	数值		说明
响应号	1	3		（设备->手机）
APP1版本号	4			分区表Bank 1 Version
APP2版本号	4			分区表Bank 2 Version
IMAGE UPDATE版本号	4			分区表IMAGE UPDATE Version
升级方式	1	value	meaning	设备读取分区表， 计算剩余空间是否能容纳 新固件， 如果能容纳选择双Bank升 级， 如果不能容纳选择单Bank 升级。
		1	选择APP1	
		2	选择APP2	
		3	选择IMAGE UPDATE	
		4	跳入IMAGE UPDATE	

5.2.4 创建发送dfu_setting和签名文件命令

- 设备端对接收到的`dfu_setting`文件进行签名验证，判断签名文件是否合法。
- 升级结束后设备端可以使用`dfu_setting`中的分区表更新自己本地分区表。
- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
命令号	1	4	（手机->设备）
新DFU SETTING大小	4		设备端通过RC特征接收完该大小数据后， 通过CC特征通知手机接收完毕。

- 新Bootsetting和签名文件接收完毕命令
- 设备端需要对Bootsetting进行CRC32完整性校验，另外需要对电子签名进行合法性检验。
- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
响应号	1	4	（设备->手机）
错误码	1		详情见：错误代码

5.2.5 创建发送固件数据命令

- 手机端通知设备端RC特征将要接收到的数据偏移值，数据大小，以及数据CRC校验值。

- 设备端RC特征接收完毕后，通过CC特征通知手机数据是否成功接收。
- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
命令号	1	5	（手机->设备）
固件数据偏移地址	4		
固件数据传输大小	4		小于等于2048
固件数据校验CRC	4		

- 固件数据接收完毕命令
- 接收完毕后，设备端需要对接收到的数据进行CRC验证，验证通过写入FLASH，如果达到4K偏移地址，需要对FLASH向后擦除4K。
- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
响应号	1	5	（设备->手机）
错误码	1		详情见：错误代码

5.2.6 FLASH固件整体校验命令

- 设备端对拷贝的固件进行CRC校验，与新分区表中的固件CRC进行对比，返回结果给手机。
- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
命令号	1	6	（手机->设备）

- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
响应号	1	6	（设备->手机）
错误码	1		详情见：错误代码

5.2.7 激活分区表和复位命令

- 设备端修改本地分区表对应的固件为激活状态，响应手机命令，再执行软件复位。
- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
命令号	1	7	（手机->设备）

- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
响应号	1	7	（设备->手机）
错误码	1		详情见：错误代码

5.2.8 跳入ImageUpdate

- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
命令号	1	8	（手机->设备）

- 命令格式：（CC特征）

名称	大小（字节）	数值	说明
响应号	1	8	（设备->手机）
错误码	1		详情见：错误代码

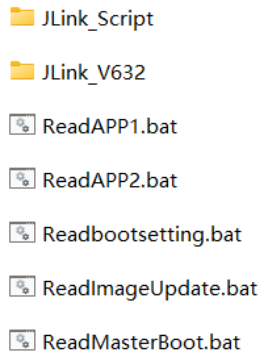
5.3 错误代码

数值	说明
0	成功
1	参数错误
2	CRC错误
3	电子签名错误

6 工具讲解

6.1 JLINK工具

进入CMT453x_SDK\utilities\dfu\JLink\目录



JLink_V632文件夹中有SEGGER公司制作的JLINK工具，用户需要安装JLINK驱动才能使用。

JLink_Script文件夹中有JLink脚本文件，用户可以直接双击ReadXX.bat文件读取芯片FLASH中的数据。

JLink还有芯片FLASH擦除和编程功能，前面章节中使用的JLINKProgramming.bat文件就是实现擦除和编程功能。

6.2 HPUTIL工具

进入CMT453x_SDK\utilities\dfu\HPUtil\文件夹。

- HPUtil.exe Windows平台执行程序。

此工具实现的功能有：

- 制作[bootsetting.bin](#)文件
- 串口升级PC上位机
- 蓝牙升级包打包工具
- ECC密钥生成，数字签名工具

6.3 AndroidUtil工具

进入CMT453x_SDK\utilities\dfu\HPAndroidUtil\文件夹。

- AndroidUtil.apk 安卓安装包。

此工具实现功能：

- 蓝牙OTA升级。

7 例程讲解

进入CMT453x_SDK_v1.3.1\projects\cmt453x_EVAL\ble_dfu目录。

- └─ app_ota
- └─ app_usart
- └─ common
- └─ image_update
- └─ masterboot

7.1 MasterBoot讲解

MasterBoot是系统上电之后所支持的第一个程序，它通过读取Bootsetting区域的信息，以判断是否跳转到Bank1, Bank2 或ImageUpdate区域执行相应的程序，或者停留在MasterBoot以通过UART来接收串口命令并执行升级动作。

其程序代码片段如下：

```
int main(void)
{
    masterboot();

    dfu_leds_config();
    dfu_led_on(LED1_GPIO_PORT, LED_GPIO1_PIN);
    dfu_led_on(LED2_GPIO_PORT, LED_GPIO2_PIN);
    HP_SCHED_INIT(256, 16);
    hp_dfu_serial_init();

    while(1)
    {
        app_sched_execute();
        __WFE();
        __SEV();
        __WFE();
    }
}
```

```
static void masterboot(void)
{
    if(hp_bootsetting.crc == dfu_crc32((uint8_t *)&hp_bootsetting.crc + 4, sizeof(HP_Bootsetting_t) - 4))
    {
```

```
    if (hp_bootsetting.master_boot_force_update != HP_BOOTSETTING_MASTER_BOOT_FORCE_UPDATE_YES)
    {
        if (hp_bootsetting.app1.activation == HP_BOOTSETTING_ACTIVATION_YES &&
hp_bootsetting.app1.start_address == HP_APP1_START_ADDRESS)
        {
            if (hp_bootsetting.app1.crc == dfu_crc32((uint8_t *)((uint32_t
*)hp_bootsetting.app1.start_address), hp_bootsetting.app1.size))
            {
                hp_dfu_boot_jump(hp_bootsetting.app1.start_address);
            }
        }
        //.....
    }
}
}
```

- **masterboot()**: 读取bootsetting分区表，直接跳入被激活且校验完整的固件。
- 点亮两个灯示意。
- 初始化简单调度。
- 初始化串口。
- 等待串口中断接收数据。

如果在masterboot()中未检测到激活的应用程序，MasterBoot将等待串口端所传送的命令，在接收到串口数据之后，将会解析、处理命令并对上位机作出响应。

```
static void sched_evt(void * p_event_data, uint16_t event_size)
{
    switch(*(uint8_t *)p_event_data)
    {
        case SCHED_EVT_RX_DATA:{
            if(m_buffer[0] == DFU_SERIAL_HEADER)
            {
                switch(m_buffer[1]){

                    case DFU_SERIAL_CMD_Ping:{
                        dfu_serial_cmd_ping();
                    }break;
                    case DFU_SERIAL_CMD_InitPkt:{
                        dfu_serial_cmd_init_pkt();
                    }break;
                    case DFU_SERIAL_CMD_Pkt_header:{
                        dfu_serial_cmd_pkt_header();
                    }break;
                }
            }
        }
    }
}
```

```
        case DFU_SERIAL_CMD_Pkt:{
            dfu_serial_cmd_pkt();
        }break;
        case DFU_SERIAL_CMD_PostValidate:{
            dfu_serial_cmd_postvalidate();
        }break;
        case DFU_SERIAL_CMD_ActivateReset:{
            dfu_serial_cmd_activate_reset();
        }break;
        case DFU_SERIAL_CMD_JumpToMasterBoot:{
            dfu_serial_cmd_jump_to_master_boot();
        }break;
    }
}
}break;
}
```

7.2 AppUsart讲解

- 串口升级APP 1程序。

```
int main(void)
{
    PWR->VTOR_REG = CURRENT_APP_START_ADDRESS | 0x80000000;

    dfu_leds_config();
    if(CURRENT_APP_START_ADDRESS == HP_APP1_START_ADDRESS){
        dfu_led_on(LED1_GPIO_PORT, LED_GPIO1_PIN);
    }else if(CURRENT_APP_START_ADDRESS == HP_APP2_START_ADDRESS){
        dfu_led_on(LED2_GPIO_PORT, LED_GPIO2_PIN);
    }

    HP_SCHED_INIT(256, 16);

    Qflash_Init();
    dfu_usart1_interrupt_config();
    dfu_usart1_enable();
    while(1)
    {
        app_sched_execute();
        __WFE();
        __SEV();
    }
}
```

```
    __WFI();  
}  
}
```

- 判断当前处于bank 1还是bank 2中，bank1则亮LED1，bank2则亮LED2。
- 初始化串口。
- 等待串口中断接收数据。
- 处理串口命令，响应PC上位机，写入强制串口升级标志，服务跳入MasterBoot程序。

```
static void sched_evt(void * p_event_data, uint16_t event_size)  
{  
    switch(*(uint8_t *)p_event_data)  
    {  
        case SCHED_EVT_RX_DATA:{  
            if(m_buffer[0] == 0xAA)  
            {  
                switch(m_buffer[1]){  
                    case DFU_SERIAL_CMD_JumpToMasterBoot:{  
                        if(m_buffer[2] == 0x01 && m_buffer[3] == 0x02 && m_buffer[4] == 0x03)  
                        {  
                            uint8_t cmd[] = {0xAA,DFU_SERIAL_CMD_JumpToMasterBoot,0};  
                            serial_send_data(cmd, sizeof(cmd));  
  
                            if(hp_dfu_boot_force_usart_dfu() == false){  
                                uint8_t cmd[] = {0xAA,DFU_SERIAL_CMD_JumpToMasterBoot,2};  
                                serial_send_data(cmd, sizeof(cmd));  
  
                                }  
                            }else  
                            {  
                                uint8_t cmd[] = {0xAA,DFU_SERIAL_CMD_JumpToMasterBoot,1};  
                                serial_send_data(cmd, sizeof(cmd));  
  
                                }  
                            }break;  
                        }  
                    }  
                }break;  
            }  
        }  
    }  
}
```

7.3 AppOTA讲解

首先判断当前程序处理bank1还是bank2，并相应的作闪灯提示。

```
int main(void)
{

    HP_LOG_INIT();

    if(CURRENT_APP_START_ADDRESS == HP_APP1_START_ADDRESS){
        HP_LOG_INFO("application 1 start new ...\r\n");
    }else if(CURRENT_APP_START_ADDRESS == HP_APP2_START_ADDRESS){
        HP_LOG_INFO("application 2 start new ...\r\n");
    }

    dfu_leds_config();

    for(uint8_t i=0;i<10;i++)
    {
        dfu_led_toggle(LED1_GPIO_PORT, LED_GPIO1_PIN);
        if(CURRENT_APP_START_ADDRESS == HP_APP1_START_ADDRESS){
            dfu_delay_ms(100);
        }else if(CURRENT_APP_START_ADDRESS == HP_APP2_START_ADDRESS){
            dfu_delay_ms(500);
        }
        dfu_led_toggle(LED2_GPIO_PORT, LED_GPIO2_PIN);
    }
    dfu_led_off(LED1_GPIO_PORT, LED_GPIO1_PIN);
    dfu_led_off(LED2_GPIO_PORT, LED_GPIO2_PIN);

    app_ble_init();

    while(1)
    {
        rwip_schedule();
        hp_sleep();
    }
}
```

初始化蓝牙协议栈。

- 配置蓝牙MAC地址。
- 配置蓝牙广播名称。
- 添加IUS (Image Update Server)服务。

```
void app_ble_init(void)
{
    struct hp_stack_cfg_t app_handler = {0};
    app_handler.ble_msg_handler = app_ble_msg_handler;
    app_handler.user_msg_handler = app_user_msg_handler;
    app_handler.lsc_cfg          = BLE_LSC_LSI_32000HZ;
    //initialization ble stack
    hp_ble_stack_init(&app_handler);

    app_ble_gap_params_init();
    app_ble_sec_init();
    app_ble_adv_init();
    app_ble_prf_init();
    //start adv
    hp_ble_adv_start();
}
```

- 处理Command Characteristic (CC) 命令

```
void hp_dfu_ble_handler_cc(uint8_t const *input, uint8_t input_len, uint8_t *output, uint8_t
*output_len)
{
    switch(input[0]){

        case OTA_CMD_CONN_PARAM_UPDATE:{
            struct gapc_conn_param conn_param;
            conn_param.intv_min = input[1]<<8 | input[2];
            conn_param.intv_max = input[3]<<8 | input[4];
            conn_param.latency = input[5]<<8 | input[6];
            conn_param.time_out = input[7]<<8 | input[8];
            app_env.manual_conn_param_update = 1;
            hp_ble_update_param(&conn_param);
            *output_len = 0;
        }break;
        case OTA_CMD_MTU_UPDATE:{
            app_env.manual_mtu_update = 1;
            hp_ble_mtu_set(input[1]<<8 | input[2]);
            *output_len = 0;
        }break;
        // *****
    }
}
```

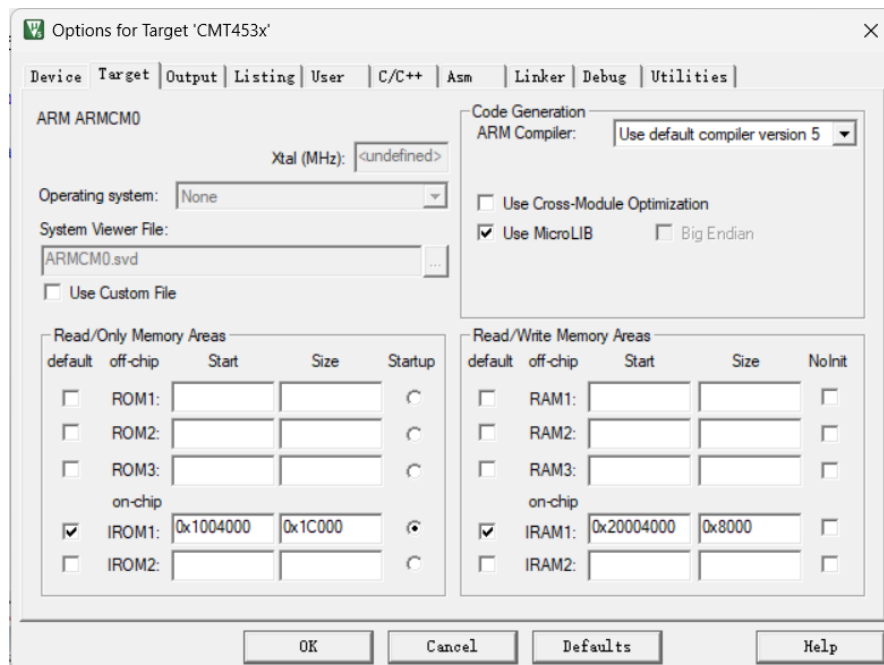
- 处理Receive Characteristic (RC) 特征值相关命令和数据

```
void hp_dfu_ble_handler_rc(uint8_t const *input, uint32_t input_len)
{
    switch(m_rc_state){

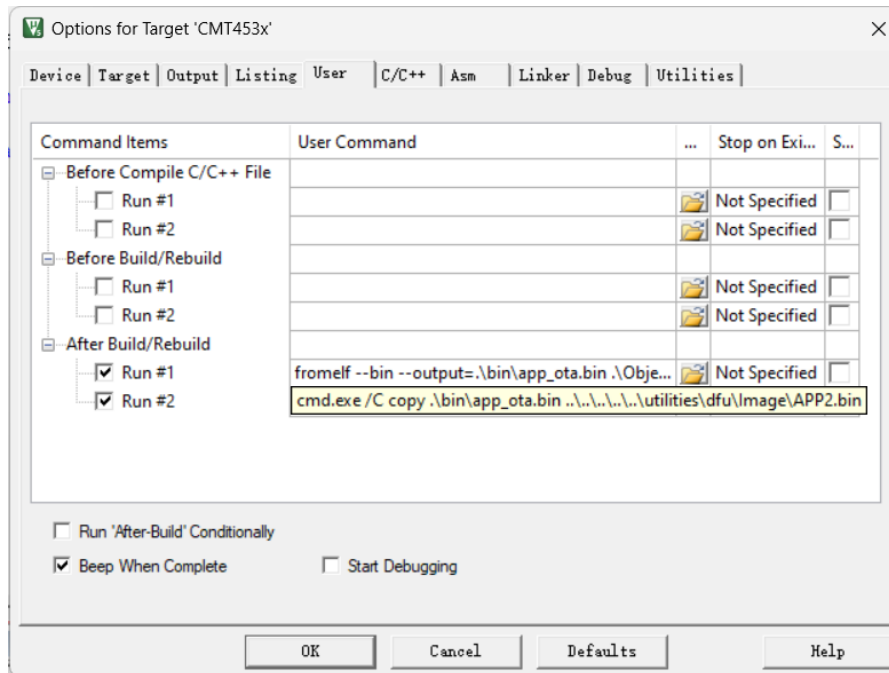
        case OTA_RC_STATE_DFU_SETTING:{
            memcpy(m_buffer + rc_mtu_offset, input , input_len);
            rc_mtu_offset += input_len;
            if(rc_mtu_offset >= m_ota_setting_size){
                rc_mtu_offset = 0;
                m_rc_state = OTA_RC_STATE_NONE;
                memcpy(&m_dfu_setting, m_buffer, sizeof(Dfu_setting_t));
                uint32_t crc = dfu_crc32((uint8_t *)&m_dfu_setting.crc + 4, sizeof(Dfu_setting_t) - 4);
                if(crc == m_dfu_setting.crc){
                    uint8_t error = 0;

                    // *****
                }
            }
        }
    }
}
```

- 制作升级包，需要用户修改keil的Options for Target中的IROM1。如果使用DualBank模式OTA，则需要准备APP1.bin, APP2.bin两个升级固件。其程序的起始地址分别为0x0100_4000和0x0102_0000，对应flash空间的Bank1和Bank2。最大Size都设置为0x1C000。



编译完成之后，Keil将会自动将编译生成的固件复制到.\utilities\dfu\Image目录下，并命名为APPx.bin。在编译APP1和APP2的工程时，请相应地修改目标固件的名称。



7.4 ImageUpdate讲解

ImageUpdate用于单Bank升级流程，它是一个最小化实现的蓝牙应用程序，用于同手机建立连接，并接收手机端所下发的命令及升级固件，在接收到升级固件后，ImageUpdate将固件直接写入原有应用程序的地址空间。

程序首先调用bootsetting_reset()以设置bootsetting为默认值，例如，如果bank1的地址区域内存在固件，那么bootsetting_reset()将bank1激活标志设置为1，这样下次重启之后，系统将自动从MasterBoot跳转到Bank1执行。

```
int main(void)
{
    bootsetting_reset();

    app_init();
    prf_init(RWIP_INIT);

    while(1)
    {
        rwip_schedule();
    }
}
```

使用Image Update server profile

```
/* Device name */
```

```
#define CUSTOM_DEVICE_NAME          "ImageUpdate"
#define CUSTOM_BLE_MAC_ADDRESS      "\x8A\x22\x77\x44\x55\x66"

//Enable Image Update Server
#define CFG_APP_HP_IUS              1
```

- 蓝牙升级命令的处理，及升级固件的接收处理请参考AppOTA讲解过程。

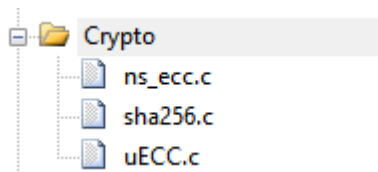
8 加密讲解

在hp_ble_dfu.c文件中通过OTA_ECC_ECDSA_SHA256_ENABLE来使能升级验证签名功能。

- 在AppOTA和ImageUpdate程序中将OTA_ECC_ECDSA_SHA256_ENABLE宏定义置1，使能升级验签功能。

```
/* Private define -----*/  
#define OTA_ECC_ECDSA_SHA256_ENABLE 1  
  
#if OTA_ECC_ECDSA_SHA256_ENABLE  
#include "hp_ecc.h"  
#endif
```

- 工程中包含以下文件，实现验签接口。



- 当嵌入端接收到dfu_setting数据后使用以上方法进行验签。

```
#if OTA_ECC_ECDSA_SHA256_ENABLE  
uint8_t raw_data[sizeof(Dfu_setting_bank_t)*3];  
memcpy(raw_data,&m_dfu_setting.app1,sizeof(Dfu_setting_bank_t));  
memcpy(raw_data+sizeof(Dfu_setting_bank_t),&m_dfu_setting.app2,sizeof(Dfu_setting_bank_t));  
memcpy(raw_data+sizeof(Dfu_setting_bank_t)*2,&m_dfu_setting.image_update,sizeof(Dfu_setting_bank_t));  
uint8_t hash_digest[32];  
if(ERROR_SUCCESS == hp_lib_ecc_hash_sha256(raw_data, sizeof(Dfu_setting_bank_t)*3, hash_digest)){  
    if(ERROR_SUCCESS != hp_lib_ecc_ecdsa_verify(hp_bootsetting.public_key, hash_digest, 32,  
m_dfu_setting.signature)){  
        error = 3;  
    }  
}else{  
    error = 3;  
}  
#endif
```

- 制作升级包，使用ECC对固件CRC，固件大小，和固件其它参数进行签名加密，并保存在dfu_setting中，嵌入端收到dfu_setting后，使用自己已知的公有密钥进行验签，如果验签成功，才能开始真正升级（即擦除FLASH和写入FLASH）。
- 只对固件独有信息进行加密签名，这样加快嵌入端的验签速度，也能够较好的保护固件的升级。

9 版本历史

日期	版本	修改
2021.07.29	V1.0	初始版本
2021.12.22	V1.1	根据SDK更新app_ota工程的代码截图，更新常见问题处理描述。
2023.06.06	V1.2	更新文档结构。

10 声明

免责声明

华普微电子股份有限公司保留在不另行通知的情况下，更改产品以提升其可靠性、功能或设计的权利。本公司亦不承担因使用此处所述产品或电路而引致的任何责任。

关于涉及生命维持设备的应用

华普微电子股份有限公司的产品并不适用于生命维持设备、装置或系统，因为这些产品的故障可能会导致人身伤害。使用或销售本产品作上述用途的客户须自行承担风险，并同意就因使用或销售不当而引致的任何损害，向本公司作出全面赔偿。

联系方式

深圳市华普微电子股份有限公司

地址：深圳市南山区西丽街道万科云城三期8栋A座30层

电话：+86-0755-82973805

邮箱：sales@hoperf.com

网址：<http://www.hoperf.com>