设备 OTA 升级集成开发指导（套餐四）

设备 OTA 升级集成开发指导（套餐四）

文档版本 01

发布日期 2021-11-22

华为终端有限公司

商标声明


、华为，以上为华为公司的商标（非详尽清单），未经华为公司书面事先明示许可，任何第三方不得以任何形式使用。

注意

华为会不定期对本文档的内容进行更新。

本文档仅作为使用指导，文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为终端有限公司

地址： 广东省东莞市松山湖园区新城路 2 号

网址： https://consumer.huawei.com

目 录

1 概述

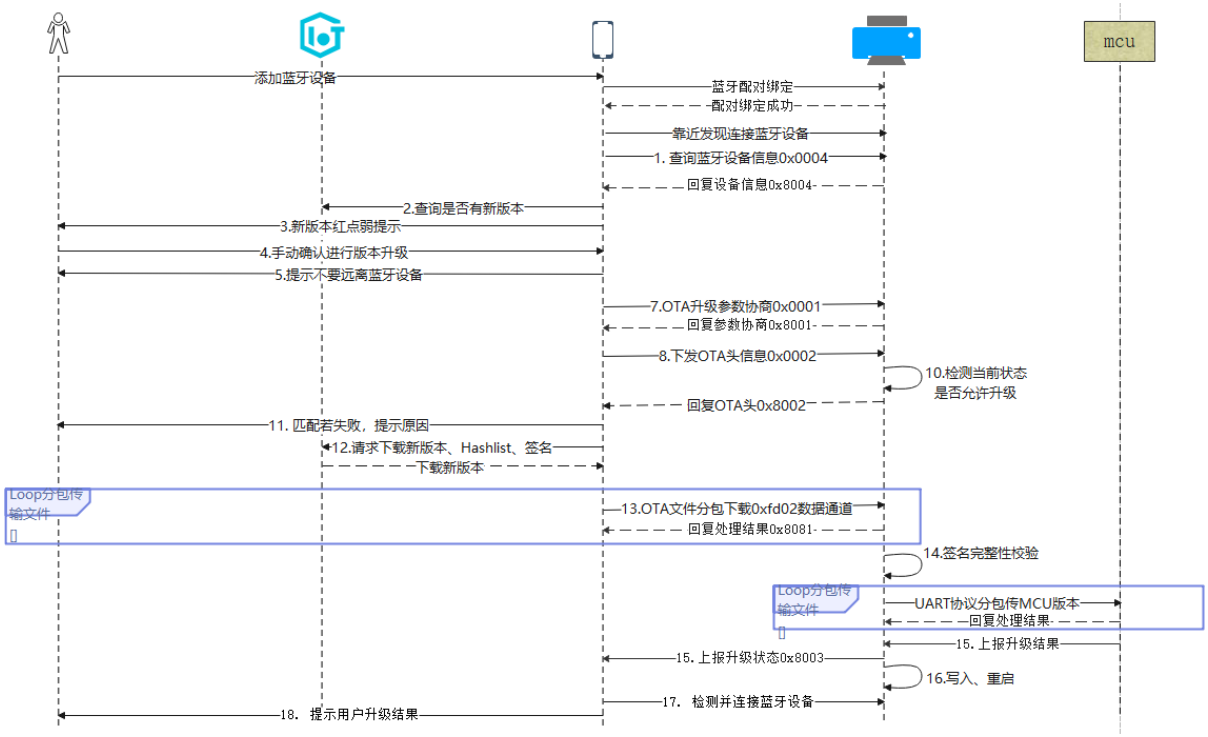
简介

本文档为 HarmonyOS Connect 生态产品合作伙伴提供 BLE 设备 OTA 升级集成指导，旨在帮助伙伴快速熟悉 OTA 升级开发流程。

方案介绍

用户在使用智慧生活 APP 进入设备详情页时，页面会从 OTA 云服务器/设备端获取设备的运行版本和最新版本做比较，如果发现设备运行版本低于最新版本，则在页面弱提示用户有最新固件可升级。用户在确认升级后，页面跳转到固件升级也从云端获取最新版本，并发送给蓝牙设备进行升级。

图1-1 BLE 设备 OTA 升级整体流程



1、BLE OTA 升级时序如上图，主要分为三个步骤：新版本的检测，下载和校验，版本的升级。

2、仅 Owner 用户支持 OTA 升级，被分享的用户 member 不支持升级。

1.1 新版本检测

手机Owner连接到BLE设备后，定期获取设备信息，并从云侧获取是否有最新版本，如有则提示向用户做出弱提示有新版本。

用户在手机侧手动确认进行版本升级后，再进入到下载升级环节。

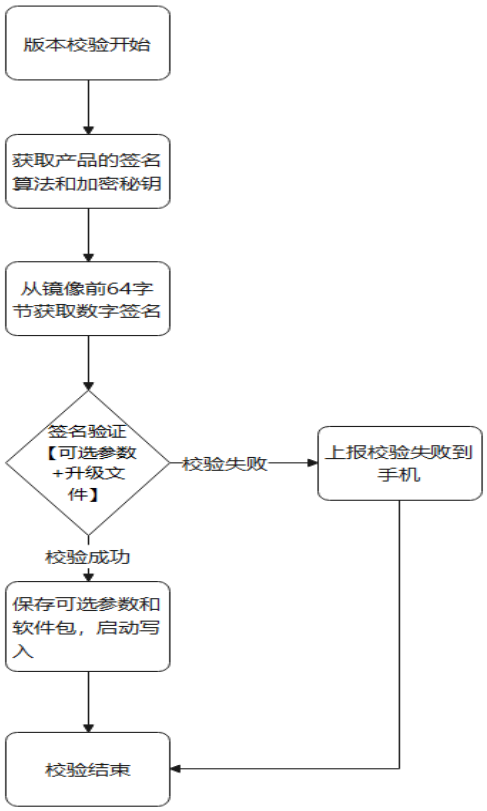
1.2 下载和校验

用户确认升级后，手机侧将OTA头信息发给设备，设备校验版本号，并且检测设备当前是否允许升级（设备当前电量是否充足、设备当前能否重启等等，由厂家实现该接口）。设备将校验结果返回给手机，如失败侧返回原因； 如成功，则开始下载文件。

手机侧通过HTTP协议从云端获取设备Hash list、签名文件、版本文件，通过BLE OTA协议发送给设备。数据帧接收不完整或者超时，则上报给手机提示升级失败。数据帧接收完整则进行版本校验。

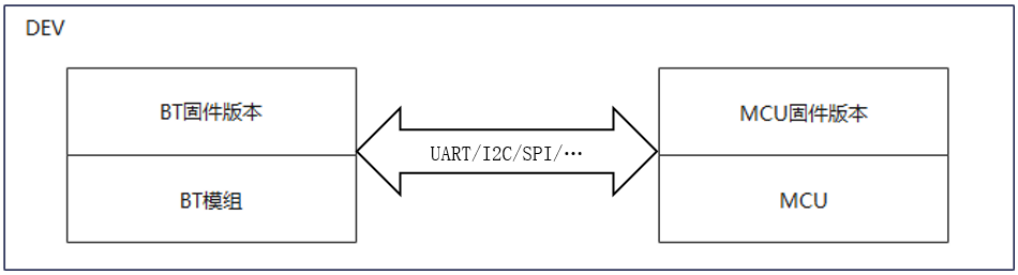
由模组程序完成接收到文件的数字签名SHA512校验，MCU不做数字签名校验。

图1-2 版本校验流程



1.3 新版本升级

蓝牙设备模块存在仅有模组，或模组+MCU两种架构，如下图所示。在模组+MCU硬件架构中，模组收到升级文件后，需将MCU版本通过通用串行总线传输给MCU；仅有BT模组的不涉及内部传输。



根据传输的文件头信息，获取BT模组版本和MCU版本（可选）。模组程序在收到模组固件版本后，保存在本地，收到MCU版本后，通过设备内部的串行总线协议转发给MCU。

如果MCU和模组都要升级，先升级模组，不重启，然后升级MCU，待模组和MCU都升级成功后才能重启设备。如果模组或MCU某一个升级失败，则要回退已升级的部分，并且上报升级失败。

图1-3 MCU 升级流程

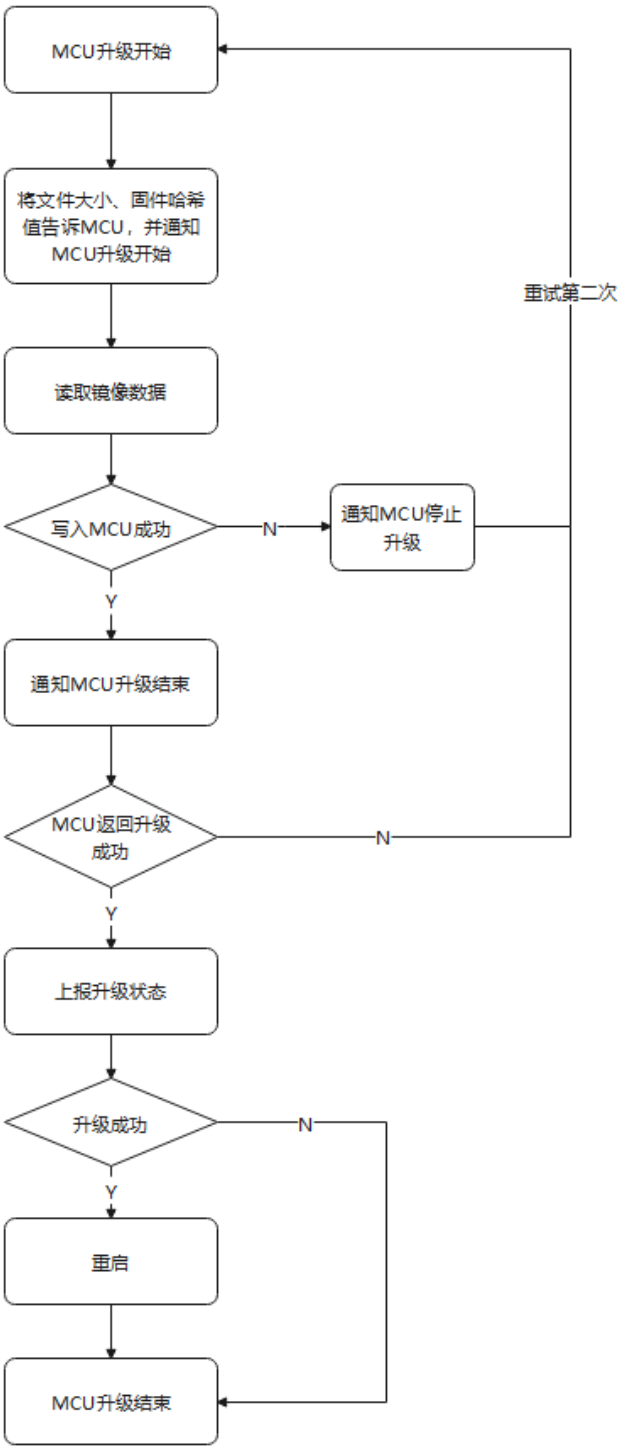
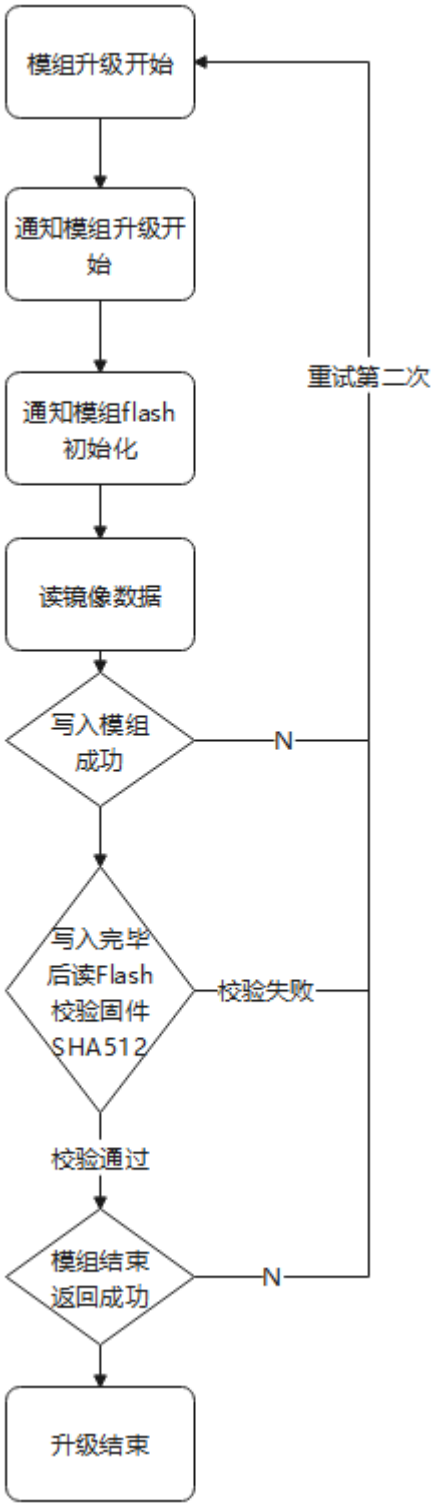


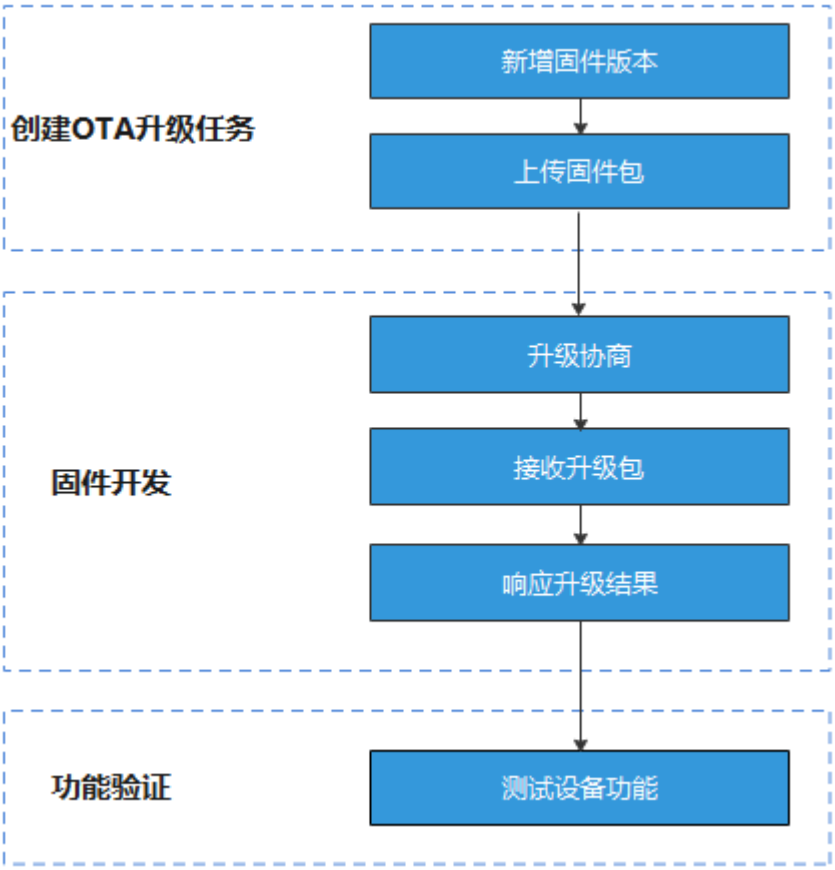
图1-4 模组升级流程



BLE 设备 OTA 升级开发流程

BLE 设备 OTA 升级需要先在 devicepartner 平台创建 OTA 升级任务，用户在设备详情页确认升级后，跳转到升级页面进行升级；设备侧需要开发实现与智慧生活 APP 的升级交互流程、接收升级包，并响应升级结果。

图1-5 BLE 设备 OTA 升级开发流程



2 创建 OTA 升级任务

参考 OTA 升级，在伙伴网站创建 OTA 升级任务。

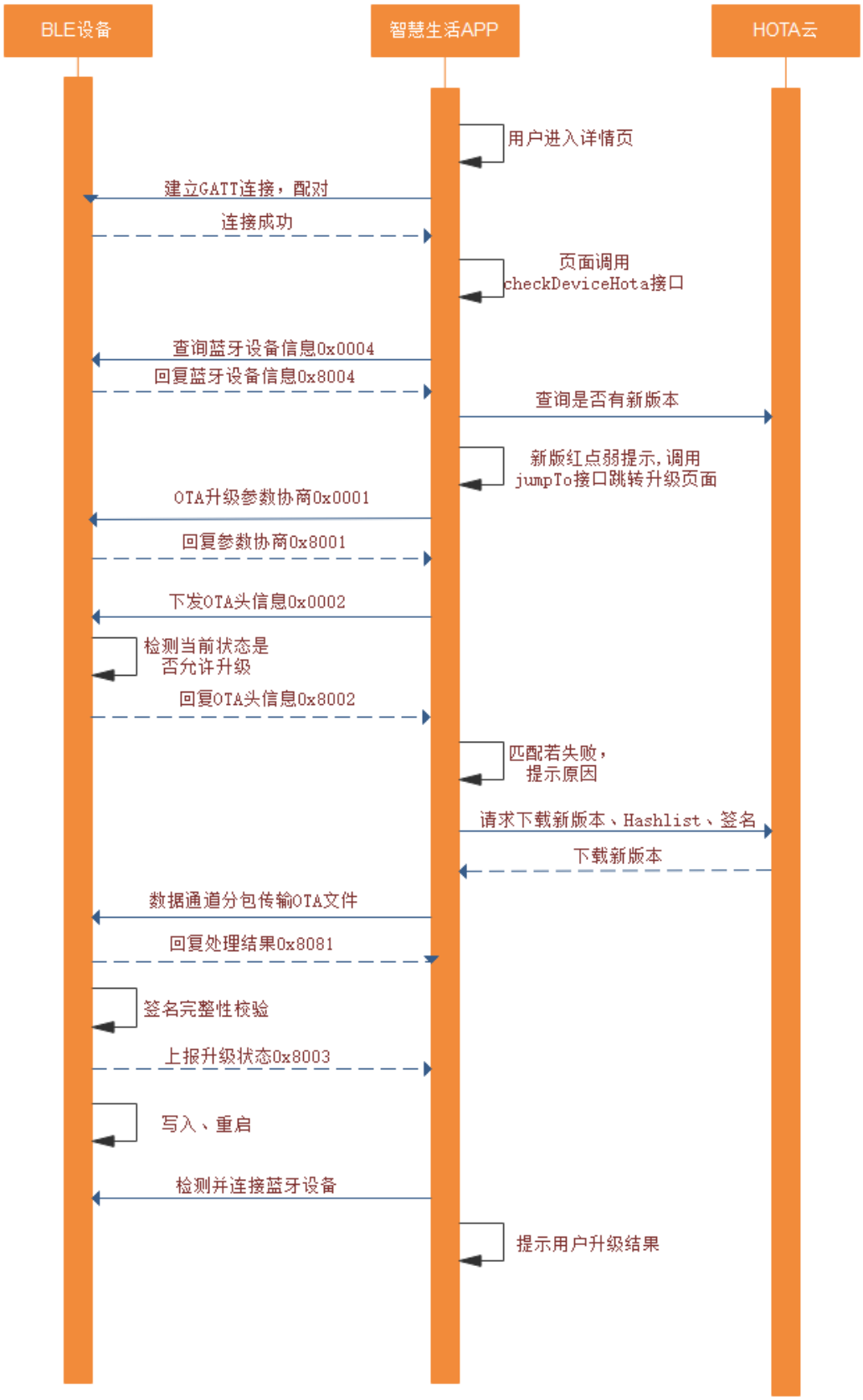
说明

当前 BLE 设备不支持线上创建 OTA 升级任务，请联系华为支持人员创建任务

3 固件开发

BLE 设备 OTA 升级时，设备与智慧生活 APP 的交互流程包括建立连接、控制指令交互、接收数据几个部分，详细流程如下：

图3-1 BLE 设备 OTA 升级流程



3.1 建立连接

OTA Service包含两个characteristic，otactrl（协议控制通道）和otadata（协议数据通道）

表3-1 BLE 通用 UUID 字段描述

服务实例 sid	serviceUuid	属性 characteristics	characteristic Uuid	属性类型 Character Type	操作权限 Permissions
ota	15f1e610-a277-43fc-a484-dd39ef8a9100				
		otaCtrl	15f1e611-a277- 43fc-a484-dd39ef8a9100	characteristic.otaCtrl	indicate/write
		otaData	15f1e612-a277- 43fc-a484-dd39ef8a9100	characteristic.otaData	write

OTA升级文件通过拆包后从数据通道发送，otadata数据通道用不带应答的write，以提高传输效率。Otactrl控制通道使用带应答的写和带应答的通知。

3.2 控制指令交互

OTA升级控制指令通过控制指令通道发送，下行命令（app ---> device）/上行状态（device--->app），

汇总如下：

Field	Bytes	Note	取值	
CMD	2	命令	0x0001	App -> Device查询设备端最大缓冲数据分段个数
			0x0002	App -> Device下发OTA头信息，查询是否支持升级
			0x0004	App -> Device下发查询设备信息
			0x8000	Device ->App出错应答
			0x8001	Device ->App上报设备端最大缓冲数据分段个数
			0x8002	Device ->App上报OTA匹配结果
			0x8081	Device ->App数据分段接收应答
			0x8003	Device ->App上报升级结果
			0x8004	Device ->App上报设备信息
PayLoad	n	命令负载		

3.2.1 下行命令出错应答（device--->app）

通过控制指令通道发送。

样例：0x80 0x00 0x01

Field	Bytes	Note	取值	
CMD	2	命令字	0x8000	
Errno	1	错误码	0x01	无法识别的CMD
			0x02	解密失败
			0x03	错误的命令参数

app 任何时候收到 0x8000，即结束升级并上报错误原因。

3.2.2 获取设备端最大缓冲数据分段个数

通过控制通道发送。

App下发OTA升级文件是以轮为单位。一轮所有分段连续下发后，App会等待Device通过0x8081上报正确收到的包。最大缓冲数据分段个数为一轮中连续下发的分段个数，最大不得超过64。该值应答暂时无效，默认为10。

App ----> Device (0x0001)

无Payload

Device ---->App (0x8001)

样例：0x80 0x01 0x0a

Field	Bytes	Note	取值
MaxSegCacheNum	1	设备最大缓冲数据包的个数。	1~64

3.2.3 查询设备信息

通过控制通道发送。

App下发查询设备信息。设备上报模组和MCU的硬件信息、版本信息等。如果BLE设备里无MCU，MTU相关的Value填写0，手机侧收到后即知道设备是否存在MCU。

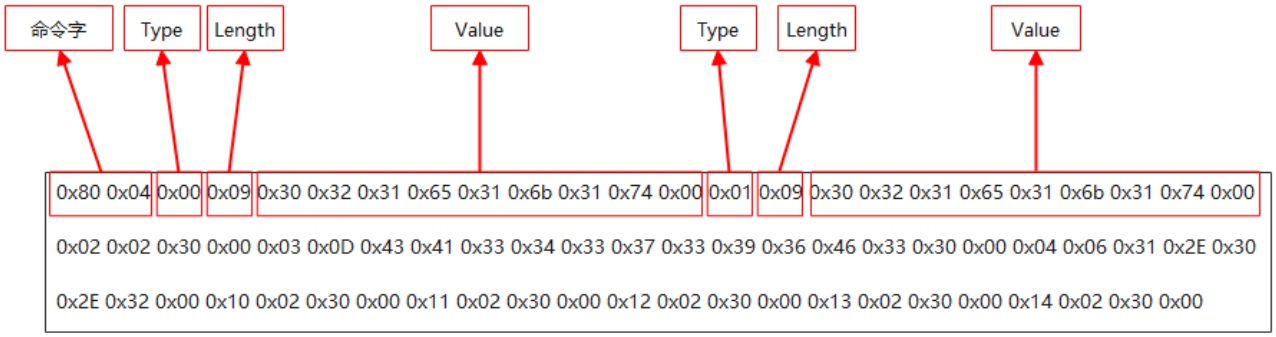
App ----> Device (0x0004)

无Payload

Device ---->App (0x8004)

设备向APP上报消息的Payload为TLV格式，每个Value值以0x00填充结尾。

样例：



Type	Length	Value	Description
0x00	1Byte	string	Module DeviceName 产品id的数字前加0，大写字母转换为小写后，前加1，例如2EKT，转换成021e1k1t，然后再转换成ASCII码，并在最后补上00，最终的Value值为0x30 0x32 0x31 0x65 0x31 0x6b 0x31 0x74 0x00
0x01	1Byte	string	Module SubDeviceName 值和Module DeviceName一样
0x02	1Byte	string	Module DeviceType 填0
0x03	1Byte	string	Module DeviceId 设备的sn，和deviceinfo上报的sn保持一致
0x04	1Byte	string	Module Firmware Version 设备固件版本号和deviceinfo上报的fwv保持一致
0x10	1Byte	string	MCU DeviceName 值同Module DeviceName，不涉及填0
0x11	1Byte	string	MCU SubDeviceName 值同Module SubDeviceName，不涉及填0
0x12	1Byte	string	MCU DeviceType 填0
0x13	1Byte	string	MCU DeviceId 值同Module DeviceId，不涉及填0
0x14	1Byte	string	MCU Firmware Version 设备MCU版本号，不涉及填0

3.2.4 下发 OTA 头信息

通过控制通道发送。

App ----> Device (0x0002)

升级开始，App给设备下发OTA头部信息，设备收到后，匹配升级条件，如果条件匹配通过0x8002应答success，如果匹配失败则通过0x8002返回相应错误码。

样例：

0x00 0x02 0x00 0x05 0x32 0x45 0x4b 0x54 0x00 0x01 0x06 0x31 0x2E 0x30 0x2E 0x33 0x00 0x02 0x04 0x00 0x02 0x08 0x40 0x03 0x01 0x00 0x04 0x04 0x00 0x00 0x00 0x00

Type	Length	Value	Description
0x00	1Byte	string	Product ID 产品ID
0x01	1Byte	string	Firmware Version 固件版本号
0x02	1Byte	Unsigned int	Firmware size 升级包大小
0x03	1Byte	string	MCU Version MCU版本号
0x04	1Byte	Unsigned int	MCU Size MCU升级包大小

Device ----> App (0x8002)

Field	Bytes	Note	取值	
Errno	1	返回值	0x00	OTA头匹配成功可以升级（有新版本，手机侧需要提示红点）
			0x01	OTA头匹配与运行版本相同，不需要升级（无新版本）
			0x02	Product ID不匹配
			0x03	Sw Version 错误
			0x04	Firmware Version 错误
			0x05	Hw Version 不匹配
			0x06	OTA文件大小超过最大值
			0x07	当前设备状态无法升级?

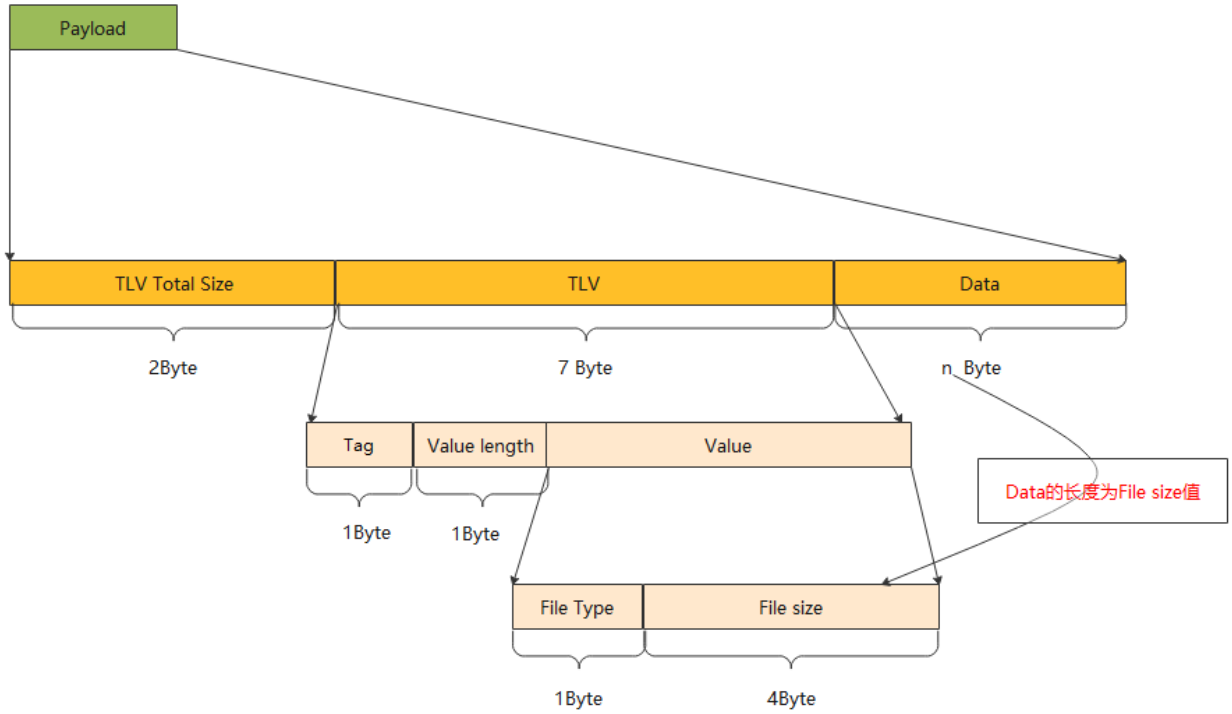
3.2.5 OTA 数据包（app ---> device）

通过数据通道发送。

OTA升级时，APP会向设备分包传送升级文件，每个包由2Byte的Seg Index和160 Bytes的Seg Payload组成

Field	Bytes	Note	取值
Seg Index	2	数据包分段索引	0~最大分段号
Seg Payload	160	分段数据	

图 1 Payload 数据包结构



Payload 数据包格式：

TLV总长度	TLV	文件内容
TLV Total Size	TLV 定义	Data
2 Byte	7Byte	Data的长度为File size值

每个TLV的格式：

Type	Value Length	Value	Description
Tag	1Byte	0x01	File information Tag
L	1Byte	0x05	Value Length
V	1Byte	0：Module Hash List 1：Module signature 2：Module File 3：MCU Hash List 4：MCU signature 5：MCU File	File Type
	4Byte	Unsigned int	File size

设备开始接收文件包时，接收到第一个文件时，根据File Type判断本次接收的文件类型，根据File size得到本次接收文件的总长度，从而计算出本次共需要接收n个包，最后一个包的Seg Payload长度。

由于第一个包TLV占用了9Byte的长度，所以Data的长度是151Byte，后面每个包的Data长度都是160Byte，最后一个包长度以实际计算得出。

样例如下：

如果文件的分包数大于10个，会按轮进行发包，每轮10个包，每轮发送完成、或整个文件发送完成后，APP会等待设备侧响应0x8081 。

3.2.6 数据接收应答

通过控制通道发送。

样例：0x80 0x81 0x00 0x00 0xE0 0x00 0x00 0x00 0x00 0x00 0x00 0x00

Device----> App(0x8081)

Field	Bytes	Note
Expected Start Seg Index	2	本轮期待收到的第一个Index
Received Seg Index Bitmap	8	本轮收到的所有分段Seg Index的BitMap数组。固定为8字节。最高位最左字节的最左的bit表示第一个包有无正确收到，收到为1，未收到为0。例如，一轮10个包，全部收到就是： FF C0 00 00 00 00 00 00.第一个包没有收到，其余9个包都收到就是07 C0 00 00 00 00 00 00。第三个包和第四个包没有收到，其余都收到就是 CF C0 00 00 00 00 00 00 APP侧在收到Device应答后，如果有未正确收到的包，则进行补发；如果全面正确，则进行下一轮发送。

3.2.7 升级状态上报

Device----> App(0x8003)

通过控制通道发送。

OTA数据接收完之后，设备端进行签名校验未通过上报0x02错误码。

后续过程中出现错误，上报0x01错误码。
如果成功，上报0x00错误码，然后重启。

Field	Bytes	Note	取值	
Errno	1	错误码	0x00	成功可以升级
			0x01	写flash错误
			0x02	签名校验失败

3.2.8 协议处理逻辑

1、device 收到 cmd(0x0002)后，对 product id 和版本号进行匹配，规则如下：

A）Product ID 与设备的 Product ID 相同
B）SW Version 大于设备当前 SW Version
C）Totle Size 小于等于设备为运行程序分配的最大 flash 空间。

满足以上条件，设备可以进行升级。

2、写数据使用带回应的写进行发送，通知数据使用 Indication（带回应）进行发送，双方通过协议层的应答确定数据是否需要重发。

4、App 端每一轮下发 10 个数据包分段，之后等待设备端应答（0x8081）。

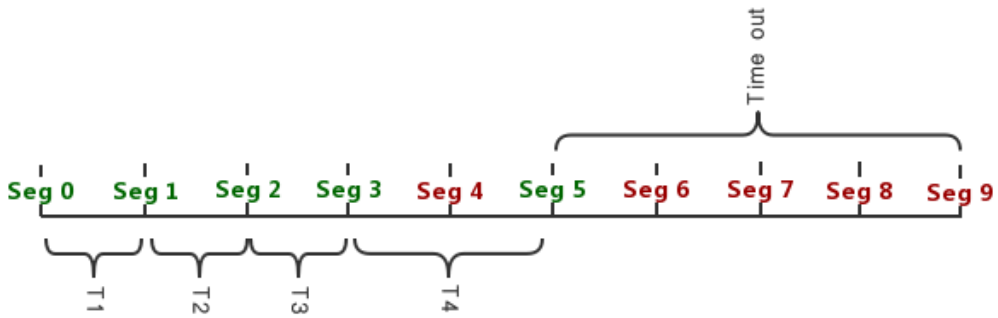
5、Device 发送数据包应答的时机

A）收到本轮发送的最后一个数据包，向 app 应答。

A）收到本次文件发送的最后一个数据包，向 app 应答。

B）超时应答，一直未收到本轮最后一个包，设置超时时间，到期后应答 app。

6、本轮数据包接收超时时间计算方法原则上根据之前收到的包与包之间的间隔的最大值来计算超时时间，每收到一个新的分段，应该重新评估本轮数据分段接收的超时时间。



（绿色为本轮收到的包，红色是未收到的包）

如上图，收到了 seg 0、1、2、3、5，计算收到 seg 9 的超时时间

Timeout = max（T1，T2，T3，T4/2） * (9 - 5 + n)（n 待定）

先计算接收到的包与包之间最大时间间隔，本轮追后收到的是 Seg 5，后面还有 4 个包，理论上要接收到 Seg 9 还需等待四倍最大时间间隔，再追加 n 个包的时间间隔的冗余，得到发送数据应答包的超时时间。

如果第一轮还未收到两个包，无法计算时最大包与包的时间间隔，最大包与包的时间间隔设置为200ms，新一轮开始时，最大包与包的时间间隔沿用上一轮的值，收到新一轮数据分段的第二个分段后再更新最大包与包的时间间隔。

7、如果有设备有模组和MCU，则由模组程序完成接收到文件的数字签名SHA512校验，MCU不做数字签名校验。

8、MCU与模组都要升级成功才能上报升级成功。

4 验证功能

4.1 测试 OTA 升级功能

步骤 1 打开设备卡片，进入设备控制界面。

步骤 2 固件升级组件弱提醒升级，点击进入升级页面进行升级，如下图所示：

----结束

5 参考

- 原子化服务开发指南
- 华为智能硬件合作伙伴 > 常见问题 > HarmonyOS Connect 生态产品