二代

OTA 协议说明

Yichip Microelectronics

Revision History

Version	Date	Author	Description
1.0	2020-01-19	Bob.wen	Initial version
1.1	2020-02-25	Bob.wen	change version

目录

1 概述

本文重点说明 YiChip 对于二代 OTA 协议。

1.1 术语说明

OTA：Over The Air，空中升级。
Normal 固件：正常使用的固件。

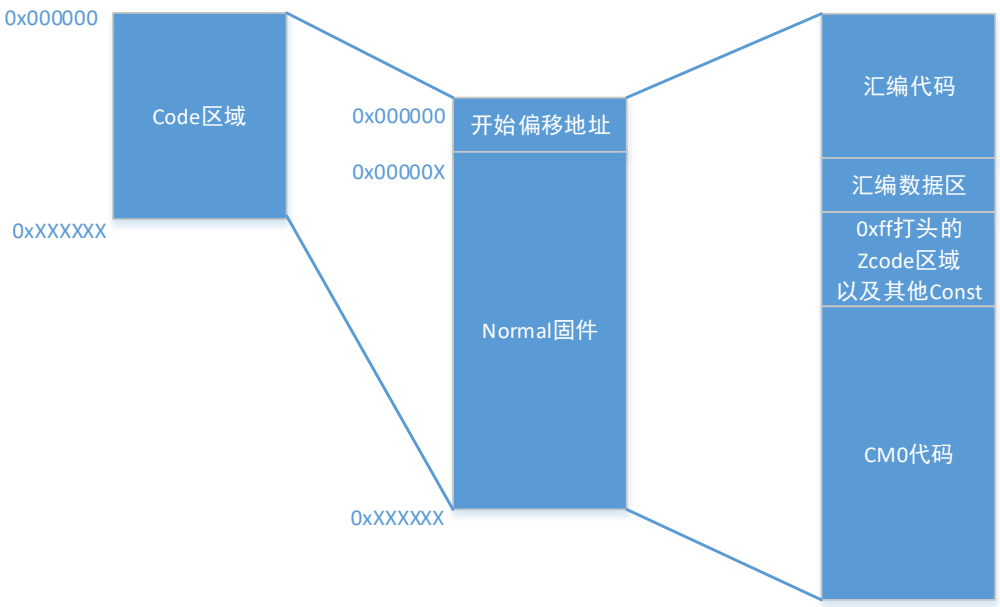
1.2 简介

本文所介绍的 OTA 协议主要针对于需要进行空中升级的设备。

1.3 技术关键点

单固件升级方案：一个固件是正常使用的版本称之为 Normal 固件，Normal 固件随着用户需求不同不断调整。

2 Flash 结构说明



3 二代 OTA-升级协议说明

3.1 基本数据协议

本文所设计的 OTA 协议大多数场景下都是 1 发 1 收的，也就是从 APP 端发送一个 Cmd 过去，设备端也会回复一个与之对应的 Evt。部分交互不需要设备回复。

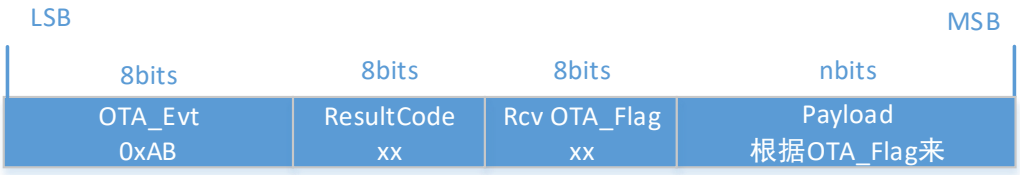
3.1.1 APP 端发送给设备端的协议

所有从 APP 端发送给设备端的数据都是由一个字节的 0xBA 打头，后面接着一个字节的 OTA_Flag，而后是 Payload 部分，Payload 的长度和意义跟随 OTA_Flag 来。

3.1.2 设备端发送给 APP 端的协议

所有从 APP 端发送给设备端的数据都是由一个字节的 0xAB 打头，后面接着一个字节的 ResultCode，而后是一字节的 OTA_Flag，而后是 Payload 部分，Payload 的长度和意义跟随 OTA_Flag 来。

ResultCode 代表当前指令的执行状态，0x00 代表执行成功。



3.2 交互指令

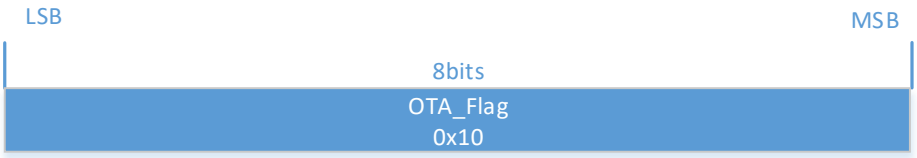
3.2.1 PORTOCOL_VERSION_REQUEST(0x10)

协议版本获取，获取当前设备支持的协议版本，用于 APP 端发送给设备端，让设备返回当前支持的 OTA 协议版本。

该指令通常第一次收发使用。

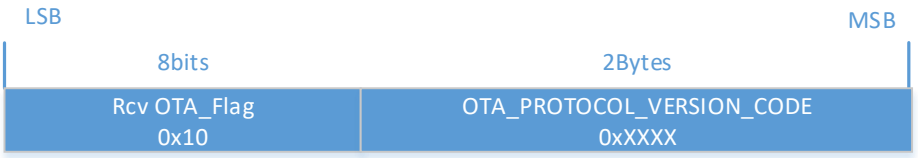
3.2.1.1 发送数据包格式

该指令只有一个 OTA_Flag，没有 Payload，设备收到后会返回当前支持的 version 版本。格式如下图：



3.2.1.2 回复数据包格式

回复的数据中会带有 2 个字节的 OTA_PROTOCOL_VERSION_CODE 信息，格式如下图：

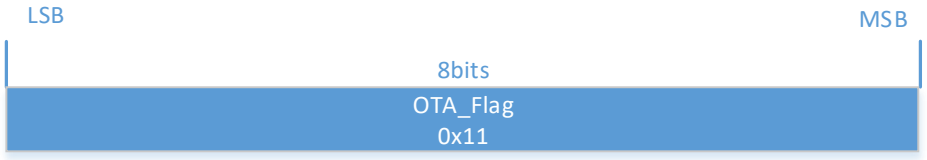


3.2.2 BUCK_SIZE_REQUEST (0x11)

BUCK Size 获取，用于获取当前设备支持的 BUCK 缓存区大小。

3.2.2.1 发送数据包格式

该指令只有一个 OTA_Flag，没有 Payload，设备收到后会返回当前支持的 BUCK Size 信息。格式如下图：



3.2.2.2 回复数据包格式

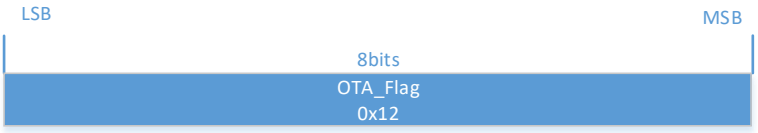
回复的数据中会带有 2 个字节的 BUCK_SIZE 信息，和 2 字节的 PACKET_MAX_LEN 信息。其中 BUCK_SIZE 代表缓存区大小，各个芯片可以根据自身的 RAM 空间来设置。PACKET_MAX_LEN 表示单笔数据包允许的最大长度（主要考虑 IPC 场景），格式如下图：

3.2.3 WORK_MODE_REQUEST (0x12)(未使用)

获取当前设备的工作模式，用于确认设备处于 Normal 工作模式还是 OTA 工作模式。

3.2.3.1 发送数据包格式

该指令只有一个 OTA_Flag，没有 Payload，设备收到后会返回当前的工作模式信息。格式如下图：



3.2.3.2 回复数据包格式

回复的数据中会带有 1 个字节当前工作模式，0x00 代表 Normal 模式，0x01 代表 OTA 模式。

APP 端如果要升级 Normal 固件，必须检查设备当前的工作模式，如果设备正在 Normal 模式下，要先让设备进入 OTA 工作模式，如果设备已经在 OTA 模

式，就直接进行升级；

如果要升级 OTA 固件，如果设备正在 Normal 模式下，直接升级，如果设备在 OTA 模式下，提示用户必须先升级 Normal 固件（考虑 Normal 固件异常情况）。

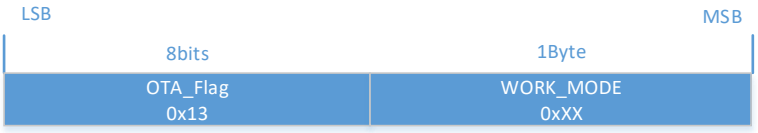
格式如下图：

3.2.4 SWITCH_WORK_MODE_REQUEST (0x13) (未使用)

命令设备切换工作模式。

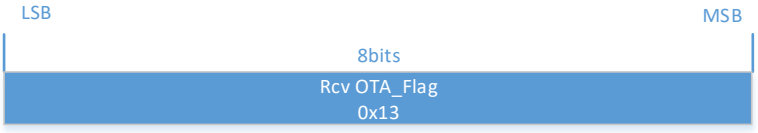
3.2.4.1 发送数据包格式

该指令包含一字节的 WORK_MODE，WORK_MODE 为要切换的模式，0x00 表示切换为 Normal 固件，0x01 表示切换为 OTA 固件。设备收到后会修改开始偏移地址，并重启。格式如下图：



3.2.4.2 回复数据包格式

回复的数据只有一个 Flag，格式如下图：



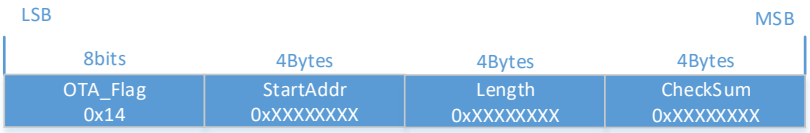
3.2.5 FLASH_CHECKSUM_REQUEST (0x14) (未使用)

检查 Flash 的 CheckSum，用于判断当前设备中 Flash 代码和指定区域的代码是否一致。

3.2.5.1 发送数据包格式

该指令包含多个信息，APP 端将要检查的 Flash 开始地址，Flash 长度以及本地计算的 CheckSum（累加求和）发送给设备。格式如下图：

☆StartAddr 表示开始偏移地址。



3.2.5.2 回复数据包格式

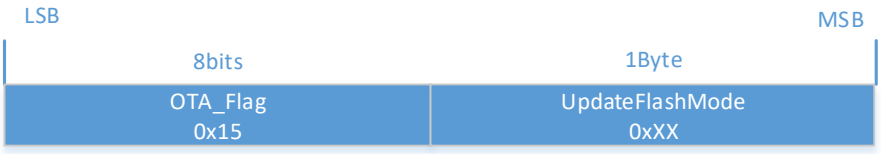
回复的数据包含设备自身根据开始地址和长度计算的 CheckSum 值。计算判断结果通过 ResultCode 返回，格式如下图：

3.2.6 START_REQUEST (0x15)

OTA 开始命令请求，用于请求开始 OTA 行为。

3.2.6.1 发送数据包格式

UpdateFlashMode 用于通知设备要升级哪块区域的数据，0x00 代表升级 Normal 固件。格式如下图：



3.2.6.2 回复数据包格式

回复的数据包含收到的 UpdateFlashMode 信息，格式如下图：

3.2.7 DATA_WRITE_CMD (0x16)

数据发送请求，用于 APP 发送数据给设备端，该命令不要求对方回复。

3.2.7.1 发送数据包格式

PacketIndex 为当前发送的包序号，Length 为后续的 DataPayload 长度，DataPayload 为要烧录的 Flash 数据，该包的最大长度不应大于 PACKET_MAX_LEN。格式如下图：

3.2.7.2 回复数据包格式

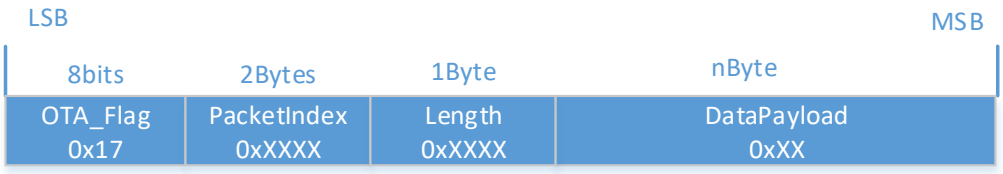
该命令无需回复。

3.2.8 DATA_WRITE_REQUEST (0x17)

数据发送请求，用于 APP 发送数据给设备端，该命令不要求对方回复。

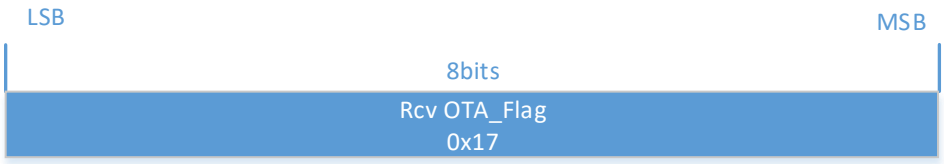
3.2.8.1 发送数据包格式

数据包格式和 DATA_WRITE_CMD 基本相同，但是需要**注意**的是，多个 DATA_WRITE_CMD 的 DataPayload 加一个 DATA_WRITE_REQUET 的 DataPayload 组成一个 BUCK，BUCK 大小应等于小于 BUCK_SIZE。**（前面的 BUCK 必须是等于 BUCK_SIZE，最后一个 BUCK 可以小于 BUCK_SIZE!!!）** 格式如下图：



3.2.8.2 回复数据包格式

回复格式如下图：

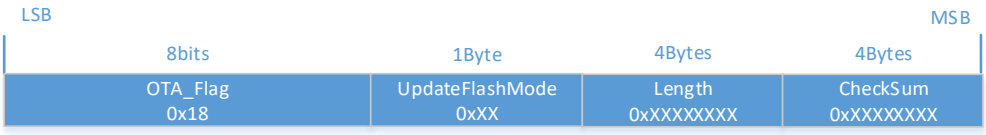


3.2.9 END_REQUEST (0x18)

结束请求，每个固件升级完毕后，需要发送给对端。

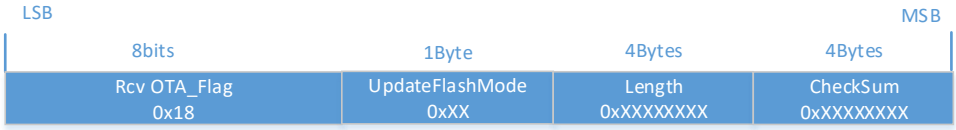
3.2.9.1 发送数据包格式

数据包格式和 START_REQUET 基本相同，加了一个对之前所有发送数据的 CheckSum，但是需要注意的是，设备收到该命令后重启（除了提示文件升级）。格式如下图：



3.2.9.2 回复数据包格式

回复格式如下图：



4 常量说明

4.1 ResultCode

每个交互，设备都会返回 ResultCode 信息，用于标识该指令是否执行成功。下面列举支持的 ResultCode 种类，除了 0x00 外，都代表升级失败。

属性值	意义	常用于
0x00	成功	所有
~~0x01~~	~~所需切换的工作模式不支持~~	~~SWITCH_WORK_MODE_REQUEST~~
~~0x02~~	~~所需切换的工作模式异常~~	~~SWITCH_WORK_MODE_REQUEST~~
0x03	不支持的固件类型	START_REQUEST
0x04	PacketIndex 不匹配	~~DATA_WRITE_CMD~~/DATA_WRITE_REQUEST
0x05	PacketLengh 溢出	~~DATA_WRITE_CMD~~/DATA_WRITE_REQUEST
0x06	BuckSize 溢出	~~DATA_WRITE_CMD~~/DATA_WRITE_REQUEST
0x07	Flash 写入异常	~~DATA_WRITE_CMD~~/DATA_WRITE_REQUEST
0xff	未知错误	所有

~~WORK_MODE~~

~~工作模式常量。~~

~~属性值~~	~~意义~~
~~0x00~~	~~Normal 模式~~
~~0x01~~	~~OTA 模式~~

~~4.3~~ 4.2 UpdateFlashMode

UpdateFlashMode 用于通知设备要升级哪块区域的数据

属性值	意义
0x00	升级 Normal 固件
~~0x01~~	~~升级 OTA 固件~~
~~0x02~~	~~升级提示音固件~~

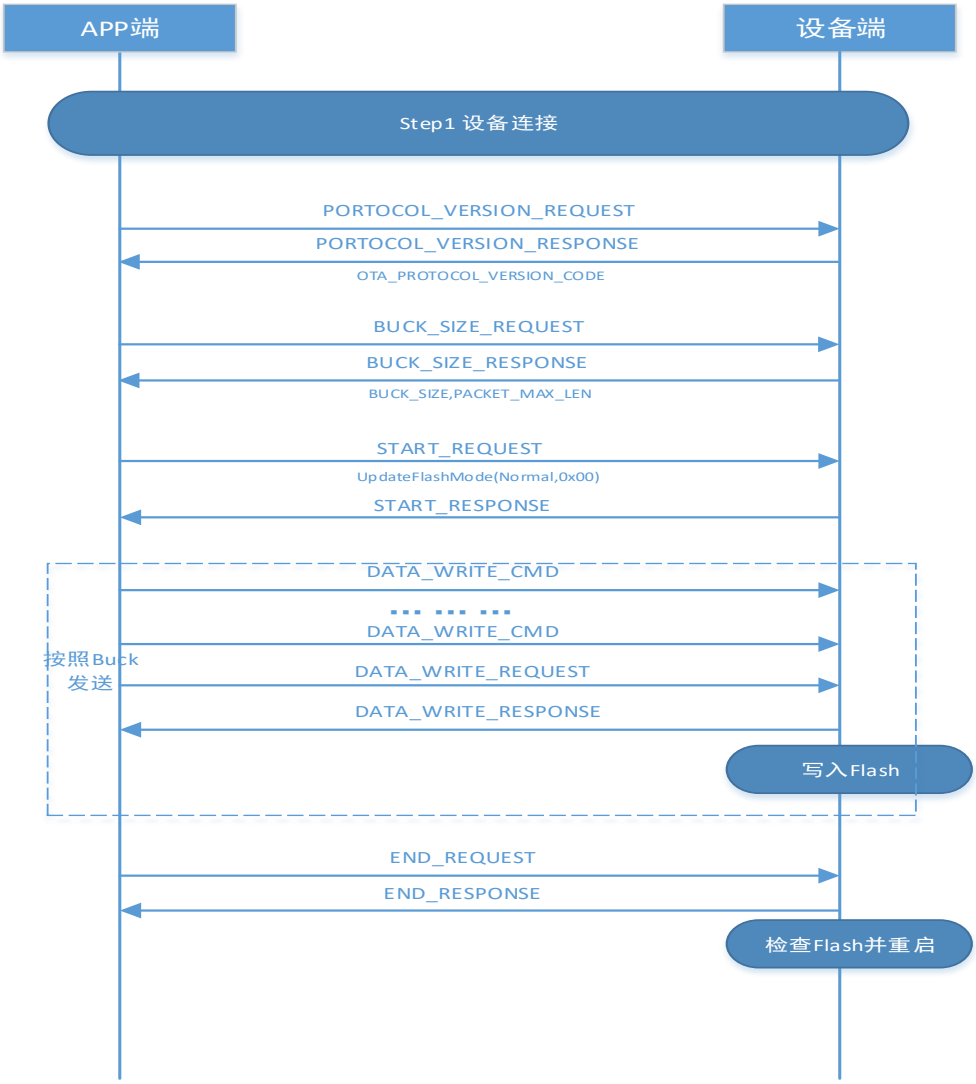
5 升级流程示意图

下面给出几个常见的升级场景示意图。

5.1 “背靠背”升级方式

5.1.1 二代协议（112D 耳机 & 跳绳 & YC3020）

升级流程图如下；APP 首先获取固件升级协议版本号，如果为二代背靠背方式，则在获取到 buck size 和 max packet length 后，直接发起升级。升级完成后校验重启，切换新固件。


APP端
设备端
Step1 设备连接
PORTOCOL_VERSION_REQUEST
PORTOCOL_VERSION_RESPONSE
OTA_PROTOCOL_VERSION_CODE
BUCK_SIZE_REQUEST
BUCK_SIZE_RESPONSE
BUCK_SIZE,PACKET_MAX_LEN
START_REQUEST
UpdateFlashMode(Normal,0x00)
START_RESPONSE
DATA_WRITE_CMD
DATA_WRITE_CMD
按照Buck发送
DATA_WRITE_REQUEST
DATA_WRITE_RESPONSE
写入Flash
END_REQUEST
END_RESPONSE
检查Flash并重启