SYDTEK OTA 设备端使用说明

**SYDTEK的OTA协议适用于SYDTEK公司旗下所有系列的芯片，这里为了方便说明，就用SYD8801来作为例子。**

# 简介

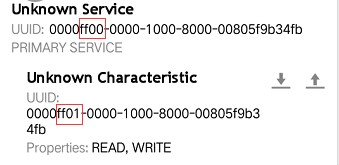
SYD8801 设备端使用 A、B 区的方式储存代码，即当前程序是在存储在 A 区，OTA 将新程序写入 B 区，然后重启系统，程序从 B 区开始执行，故中途断开连接或者中断 OTA 不会造成设备“变砖”。A、B 区随着 OTA 的次数相互切换。

设备端大部分功能已经在 ota.h、ota.c 实现，只需实现相应的服务特性读写，以及调用相关 API 即可。

# 二、OTA 升级，设备端需要条件

1、实现服务以及对应的特性读写

<1>服务 UUID 为 FF00

<2>特性 UUID 为 FF01 权限可读、可写

2、keil 工程包含 ota.c，ota.h，并调用相关函数

3、OTA 之前，最好能将蓝牙的连接间隔变小、以便提升 OTA 的速度。

三、OTA 实现

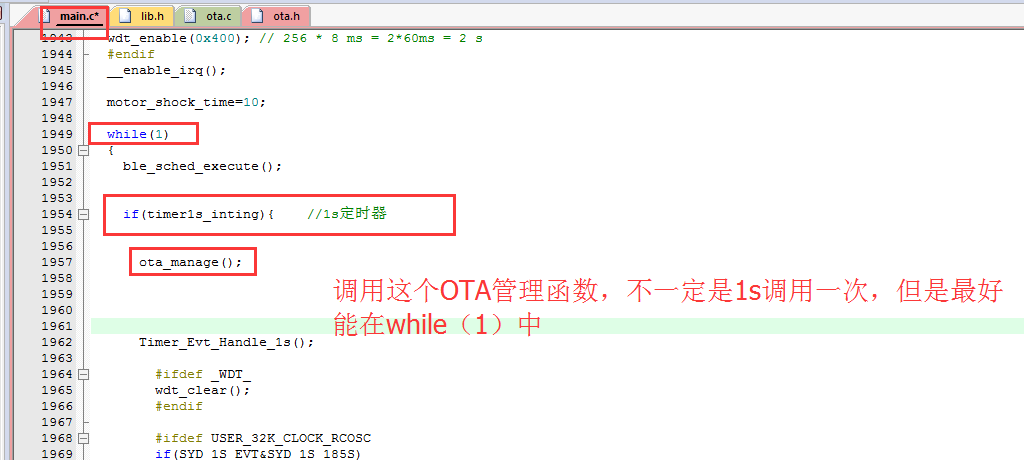
第一、实现设备端与 APP 数据接收与返回



第二、main.c 实现 ota 状态管理函数——ota\_manage



第三、在 main 函数的 while(1)调用 ota\_manage 函数

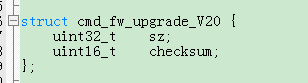


|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 日期 | 撰写人 | 版本 |
| SYD8801 OTA 设备端使用说明.PDF | 2018 年 3 月 20 日 | Bihu | 0.1 |

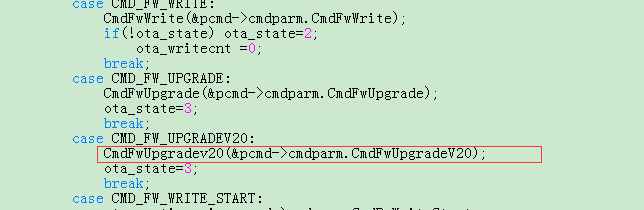
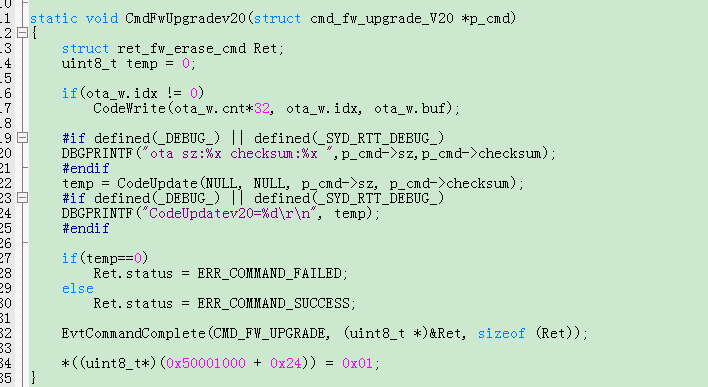
Version 2.0版本协议补充说明：

上文中所述都是针对Version 1.0版本的协议，为了弥补Version 1.0版本在空升代码量只能够达到65536Byte的局限性，这里开出Version 2.0的版本。

不能够升级大于65536Byte的代码主要是因为CMD\_FW\_UPGRADE命令中的” sz”时候“uint16\_t”类型的，所以这里增加一个新的命令CMD\_FW\_UPGRADEV20替换原来的CMD\_FW\_UPGRADE命令，这里增加一个结构体：



在参数函数里执行如下函数：

Version 3.0版本协议补充说明：

Version 2.0虽然解决了OTA空间大小的问题，但是因为他是基于Version 1.0的，所以OTA的速度依旧达不到理想的效果，特别是在SYD8821和SYD8811中代码量至少达到了248KB以上，这时候如果还用原来协议OTA的速度的体验不是很好。

这里改掉CMD\_FW\_WRITE的协议，原来的一个CMD\_FW\_WRITE命令只是发送了15个byte的数据，这里改成了一个命令发送20个byte，同时这里采用分段发送，每个段都会进行校验，如果检验不通过会重发当前段！具体修改如下：

1. 增加CMD\_FW\_WRITE\_START及其流程：

#define CMD\_FW\_WRITE\_START 0x14

struct cmd\_fw\_write\_start {

uint32\_t offset;

uint16\_t sz;

uint16\_t checksum;

};

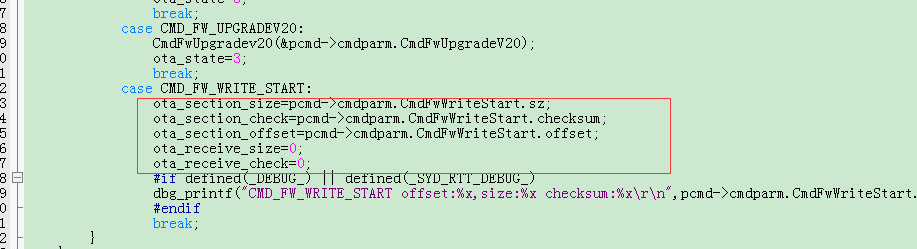
struct ret\_fw\_write\_start\_cmd {

uint8\_t status;

uint16\_t sz;

uint16\_t checksum;

};



1. 当该段数据发送完后将发起一次对设备端的读操作，读回的数据包含了设备端计算出来的校验和值。当该段落发送完成的时候则返回该段落的校验和以及大小：

