目录

[1.CMD：BCS\_BLE\_EVT\_TYPE结构的ID //十进制 4](#_Toc14185)

[2.CTRL：BCS\_BLE\_CTRL\_TYPE结构的ID//十进制 4](#_Toc27894)

[3.预留：表示固件端未实现功能 4](#_Toc21486)

[4.数据为小端收发（如有大小端错误请指正） 4](#_Toc7390)

[1.提醒类指令(用于测试) 4](#_Toc19061)

[2.功能类指令 5](#_Toc25254)

[3.获取固件基础信息 5](#_Toc18596)

[4.获取gsensor三轴坐标 5](#_Toc5803)

[5.设置用户信息 6](#_Toc18817)

[6.设置/获取公英制 6](#_Toc17269)

[7.设置日期时间 6](#_Toc8869)

[note:已转换时区后的时间戳 6](#_Toc20637)

[8.设置/获取时间格式 6](#_Toc3082)

[9.设置/获取温度格式 7](#_Toc20982)

[10.设置语言ID 7](#_Toc32519)

[11.设置/获取表盘ID 7](#_Toc1280)

[12.获取电池电量 7](#_Toc28290)

[13.设置目标步数 8](#_Toc5866)

[14.获取分时步数/卡路里 8](#_Toc10589)

[15.获取历史七天运动数据 8](#_Toc24493)

[1.有数据返回 8](#_Toc429)

[2.没有要同步的数据 8](#_Toc25359)

[16.设置/获取消息通知开关（ios） 8](#_Toc26169)

[17.消息推送（安卓） 9](#_Toc11903)

[18.设置亮屏时长 10](#_Toc18764)

[19.获取手环配置 10](#_Toc9016)

[20.设置/获取睡眠数据 12](#_Toc18159)

[21.获取运动记录 12](#_Toc30215)

[22.设置/获取运动数据 13](#_Toc19961)

[23.设置天气预报 13](#_Toc15010)

[24.设置实时天气 15](#_Toc27291)

[25.设置抬腕亮屏 15](#_Toc16001)

[26.设置勿扰模式 16](#_Toc1618)

[27.设置久坐提醒 16](#_Toc22406)

[28.设置喝水提醒 16](#_Toc20581)

[29.设置洗手提醒 17](#_Toc11526)

[30.设置日程提醒 17](#_Toc23385)

[31.设置/获取闹钟 17](#_Toc30187)

[32.获取体温/心率/血压/血氧 18](#_Toc7939)

[33.获取7天的全天心率/全天血氧/全天体温 19](#_Toc15619)

[34.设置全天测量开关 19](#_Toc19483)

[1：打开 19](#_Toc1214)

[35.设置hr/bp/bo2数据（单次） 20](#_Toc1408)

[36.拍照 20](#_Toc11753)

[37.音乐控制（安卓）固件端请求 20](#_Toc29840)

[38.手环配置1(结构格式，固定后不要乱改) 21](#_Toc24868)

[note:返回固定132个字节 22](#_Toc6460)

[39.设置手机系统类型 23](#_Toc31014)

[40. 同步MAC地址（安卓) 23](#_Toc31683)

[41.APP 请求同步基础数据 23](#_Toc7183)

[42. 查找手机 24](#_Toc21813)

[43. 全天血压 24](#_Toc25685)

[44.交友码 25](#_Toc11570)

[note : len < 230Byte 25](#_Toc15888)

[45.收款码 25](#_Toc7812)

[note : len < 230Byte 25](#_Toc945)

[46.女性健康 25](#_Toc16760)

[47.android 挂断电话或接听电话 26](#_Toc32108)

[48.解除绑定（固件端清除所有数据） 26](#_Toc29227)

[49.心电同步 26](#_Toc16592)

[50.同步固件端支持功能+设备屏幕信息 26](#_Toc6697)

[51.同步3.0 MAC（该指令不需APP发指令），ios不需要处理 26](#_Toc6769)

[52.吃药提醒 27](#_Toc8205)

[53.GPS 运动 27](#_Toc14734)

[54.海拔 27](#_Toc29950)

[55.十米葫芦(体感游戏) 27](#_Toc22026)

[56.获取SD歌曲名称列表（仅支持杰里二代) 28](#_Toc2887)

[57.获取SD电子书名称列表（仅支持杰里二代) 28](#_Toc23090)

[58.获取SD卡剩余空间（仅支持杰里二代) 29](#_Toc13970)

[59.获取SD录音名称列表（仅支持杰里二代) 29](#_Toc1917)

[60.设置用户名称（仅支持杰里二代) 29](#_Toc24137)

[61.获取用户头像KEY（仅支持杰里二代) 29](#_Toc18987)

[99.OTA 大数据到手表(每包数据不能超过230Byte) 29](#_Toc3814)

[5帧 主动切换 byte7= 0x15， 30](#_Toc23551)

[一.准备升级 31](#_Toc17800)

[二.升级开始（CTRL = EVT\_TYPE\_OTA\_DATA） 31](#_Toc28253)

[1.APP发送数据 31](#_Toc31678)

[三.异常处理，APP均需要处理（CTRL = EVT\_TYPE\_OTA\_END） 32](#_Toc21359)

[步骤一：立即停止原有发送任务包，发送一个pack\_id + 1 的包 32](#_Toc18068)

[步骤二：开启5秒钟定时器，等待修复包OTA\_TYPE\_REPAIR，在5秒钟内 32](#_Toc20844)

[100.手表作为master，发送大文件（仅支持杰里二代) 33](#_Toc19261)

[EVT\_TYPE\_MASTER\_START = 203,/\* master 开始指令 \*/ 33](#_Toc31026)

[EVT\_TYPE\_MASTER\_DATA = 204,/\* master 数据指令 \*/ 33](#_Toc7625)

[EVT\_TYPE\_MASTER\_END = 205,/\* master 结束 \*/ 33](#_Toc25232)

[异常类型：(1byte) 33](#_Toc710)

[MASTER\_TYPE\_FAIL\_1 = 1,//未找到对应文件 33](#_Toc1007)

[MASTER\_TYPE\_FAIL\_2 = 2,//未接收到确认开始指令 33](#_Toc18494)

[传输文件类型：(1byte) 33](#_Toc30482)

[MASTER\_TYPE\_RECORD = 0,//传输录音文件 33](#_Toc23887)

[流程：FILE\_OFFSET（4byte）,SIZE （4byte） 33](#_Toc28960)

[1.APP-->手表（把需要的文件name发给固件） 33](#_Toc17925)

[CMD = EVT\_TYPE\_MASTER\_START 33](#_Toc10879)

[2.手表->APP 33](#_Toc22273)

[//成功找到对应文件 33](#_Toc31063)

[CMD = EVT\_TYPE\_MASTER\_START 33](#_Toc14774)

[CMD = EVT\_TYPE\_MASTER\_END 34](#_Toc12546)

[3.APP-->手表（APP已确认接收到目标文件大小，并发送需要的数据偏移） 34](#_Toc24072)

[CMD = EVT\_TYPE\_MASTER\_DATA 34](#_Toc29639)

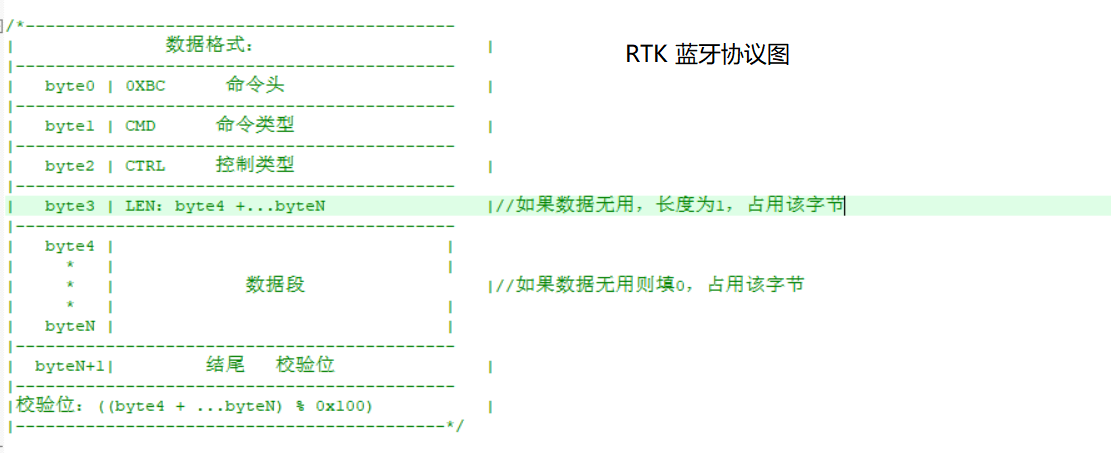
[4.手表->APP(循环发送，直到发送完成) 34](#_Toc15938)

[CMD = EVT\_TYPE\_MASTER\_DATA 34](#_Toc21787)

[5.APP-->手表（文件接收完成） 34](#_Toc8549)

[CMD = EVT\_TYPE\_MASTER\_DATA 34](#_Toc28409)

[此时FILE\_OFFSET总数据大小 34](#_Toc27563)



参考蓝牙协议.h

1.CMD：BCS\_BLE\_EVT\_TYPE结构的ID //十进制

2.CTRL：BCS\_BLE\_CTRL\_TYPE结构的ID//十进制

3.预留：表示固件端未实现功能

4.数据为小端收发（如有大小端错误请指正）

//详细指令介绍 app -> 手表

1.提醒类指令(用于测试)

EVT\_TYPE\_ALERT\_FIND\_WATCH = 0, /\* 查找手环 \*/

EVT\_TYPE\_ALERT\_LONG\_SIT = 1, /\* 久坐提醒 \*/

EVT\_TYPE\_ALERT\_STEP\_GOAL = 2, /\* 目标达成 \*/

EVT\_TYPE\_ALERT\_SLEEP\_GOAL = 3, /\* 目标达成 \*/

EVT\_TYPE\_ALERT\_DRINK\_WATER = 4, /\* 喝水提醒 \*/

EVT\_TYPE\_ALERT\_WASH\_HAND = 5, /\* 洗手提醒 \*/

EVT\_TYPE\_ALERT\_HEART\_RATE = 6, /\* 心率报警 \*/

EVT\_TYPE\_ALERT\_LOW\_POWER = 7, /\* 低电提醒 \*/

EVT\_TYPE\_ALERT\_RECHARGEING = 8, /\* 充电提醒 \*/

EVT\_TYPE\_ALERT\_CHARGE\_FULL = 9, /\* 充满电提醒 \*/

EVT\_TYPE\_ALERT\_ALARM = 10, /\* 闹钟提醒 \*/

EVT\_TYPE\_ALERT\_MSG = 11, /\* 消息提醒 \*/

EVT\_TYPE\_ALERT\_MENSTRUAL = 12, /\* 经期提醒 \*/

EVT\_TYPE\_ALERT\_SCHEDULE = 13, /\* 日程提醒 \*/

EVT\_TYPE\_ALERT\_UPGRADE = 14, /\* 升级图标 \*/

2.功能类指令

EVT\_TYPE\_MES\_HR = 29,/\* 心电测量 \*/

type ：1：开始 0：停止

发送：|0xBC|CMD|CTRL(02)|01|type|校验值|

EVT\_TYPE\_MES\_HR = 30,/\* 心率测量 \*/

EVT\_TYPE\_MES\_BP = 31,/\* 血压测量 \*/

EVT\_TYPE\_MES\_BO2 = 32,/\* 血氧测量 \*/

EVT\_TYPE\_SHUTDOWN = 33,/\* 关机 \*/

EVT\_TYPE\_REBOOT = 34,/\* 重启 \*/

EVT\_TYPE\_RESET = 35,/\* 恢复出厂 \*/

EVT\_TYPE\_LOW\_POWER\_SHUTDOWN = 36,/\* 低电关机 \*/

发送：|0xBC|CMD|CTRL|01|00|校验值|(发送后手表不返回任何消息)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

3.获取固件基础信息

EVT\_TYPE\_PRODUCT\_ID = 60,/\*产品ID\*/

EVT\_TYPE\_TP\_VER = 61,/\* TP版本 \*/

EVT\_TYPE\_FIRMWARE\_VER = 62,/\* 固件版本 \*/

EVT\_TYPE\_UI\_VER = 63,/\* UI版本 \*/

发送：|0xBC|CMD|CTRL|01|00|校验值|

例如：返回数据长度为 5 则接收格式如下

接收：|0xBC|CMD|CTRL(0x03)|0x05|data1|data2|data3|data4|data5|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

4.获取gsensor三轴坐标

EVT\_TYPE\_GSENSOR\_XYZ = 64,/\* GSENSOR坐标 \*/

发送：|0xBC|CMD|CTRL|01|00|校验值|

note：返回固定6字节数据，则接收格式如下(x,y,z)

接收：|0xBC|CMD|CTRL(0x03)|0x06|data1|data2|data3|data4|data5|data6|校验值

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

5.设置用户信息

EVT\_TYPE\_USER\_INFO = 65,/\* 用户信息 \*/

uint8\_t sex; //性别

uint8\_t height; //身高

uint8\_t weight; //体重

uint8\_t age; //年龄

uint8\_t day; //日

uint8\_t month; //月

uint16\_t year; //年

note：发送固定8字节数据，则接收格式如下

发送：|0xBC|CMD|CTRL|08|性别|身高|体重|年龄|日|月|年(2字节)|校验值|(发送后手表不返回任何消息)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

6.设置/获取公英制

EVT\_TYPE\_UNIT\_SYSTEM = 66,/\* 单位制度 \*/

note: 0:公制，1：英制（手表默认公制）

发送：|0xBC|CMD|CTRL|01|00|校验值| 公制

发送：|0xBC|CMD|CTRL|01|01|校验值| 英制

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

7.设置日期时间

EVT\_TYPE\_DATE\_TIME = 67,/\* 日期、时间 \*/

note:已转换时区后的时间戳

发送：|0xBC|CMD|CTRL|08|时间戳（4个字节）|时区1（2字节）|时区2（2字节）|校验值|

“时区”单位使用分钟，带正负号

(发送后手表不返回任何消息)

接收：|0xBC|CMD|CTRL|01|00|校验值| //表示请求同步时间

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

8.设置/获取时间格式

EVT\_TYPE\_TIME\_MODE = 68,/\* 时间格式 \*/

note: 0:24h制，1：12h制（手表默认24h制）

发送：|0xBC|CMD|CTRL|01|00|校验值| 24h制

发送：|0xBC|CMD|CTRL|01|01|校验值| 12h制

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

9.设置/获取温度格式

EVT\_TYPE\_TEMP\_UNIT = 69,/\* 温度格式 \*/

note: 0:摄氏度制，1：华氏度制（手表默认摄氏度制）

发送：|0xBC|CMD|CTRL|01|00|校验值| 摄氏度制

发送：|0xBC|CMD|CTRL|01|01|校验值| 华氏度制

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

10.设置语言ID

EVT\_TYPE\_LANGUAGE = 70,/\* 语言 \*/

note: ID 查看language.h

发送：|0xBC|CMD|CTRL|01|ID|校验值|

(发送后手表不返回任何消息)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

11.设置/获取表盘ID

EVT\_TYPE\_TIME\_STYLE = 71,/\* 表盘ID \*/

例如：3个固定表盘 ，（1个自定义表盘，1个表盘市场，共用一个位置，既：同时只能存在一个）

//固定表盘不可被替换

CUR\_ID :当前表盘ID（0~3），TOTAL\_ID:当前总表盘个数（4）

发送：|0xBC|CMD|CTRL|01|CUR\_ID|校验值|

设置(发送后手表不返回任何消息)

返回：|0xBC|CMD|CTRL|02|CUR\_ID|TOTAL\_ID|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

12.获取电池电量

EVT\_TYPE\_BAT = 72,/\* 电量 \*/

发送：|0xBC|CMD|CTRL|01|00|校验值|

note：固定返回5个byte有效数据

接收：|0xBC|CMD|CTRL|05|data1|data2|data3|data4|data5|校验值|

data1= 电量百分比（0~99），data2~data5 = 电压值 mv

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

13.设置目标步数

EVT\_TYPE\_TARGET\_STEP = 73,/\* 目标步数 \*/

note：4字节32bit值

发送：|0xBC|CMD|CTRL|04|步数值4字节|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

14.获取分时步数/卡路里

EVT\_TYPE\_HOUR\_STEP = 74,/\* 分时计步 \*/

note：96字节每【步数2byte】【卡路里2byte】,共24组

发送：|0xBC|CMD|CTRL|0x01|00|校验值|

接收：|0xBC|CMD|CTRL(0x03)|len|96字节|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

15.获取历史七天运动数据

EVT\_TYPE\_HISTORY\_SPORT\_DATA = 75,/\* 历史七天运动数据（24小时分时计步、全天运动数据） \*/

uint32\_t sport\_timestamp;//该时间戳为每天的 23:59:55

uint32\_t sport\_steps;//全天总步数 ，最大值为99999

uint32\_t sport\_distance;//全天总距离 (分米)

uint32\_t sport\_kcal;//全天总卡路里 (kcal ) 10 sport\_kcal = 1kcal

uint32\_t sport\_time;//全天运动时长 (秒)

uint16\_t per\_hour\_step[24];//24小时分时步数

uint16\_t per\_hour\_kcal[24];//24小时分时卡路里

发送：|0xBC|CMD|CTRL|01|00|校验值|

1.有数据返回

返回：|0xBC|CMD|CTRL(0x03)|data..................|校验值|

2.没有要同步的数据

返回：|0xBC|CMD|CTRL(0x03)|0x01|00|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

16.设置/获取消息通知开关（ios）

EVT\_TYPE\_NOTIFICATIONS = 76,/\* 消息通知 \*/

状态：1：打开 0 ：关闭

note:数据共32byte

switch\_msg\_other : BYTE1

switch\_msg\_call : BYTE2

switch\_msg\_sms : BYTE3

switch\_msg\_wechat : BYTE4

switch\_msg\_qq : BYTE5

switch\_msg\_facebook : BYTE6

switch\_msg\_messenger : BYTE7

switch\_msg\_twitter : BYTE8

switch\_msg\_instagram : BYTE9

switch\_msg\_skype : BYTE10

switch\_msg\_whatsapp : BYTE11

switch\_msg\_line : BYTE12

switch\_msg\_email : BYTE13

switch\_msg\_weibo : BYTE14;

switch\_msg\_snapchat : BYTE15;

switch\_msg\_gmail : BYTE16;

switch\_msg\_linkedin : BYTE17;

switch\_msg\_talk : BYTE18;

switch\_msg\_telegram : BYTE19;

BYTE16 ~ BYTE32 :没用到，补0

设置发送：|0xBC|CMD|CTRL(02)|0x20|data1|data2........data32|校验值|

获取发送：|0xBC|CMD|CTRL(01)|0x01|00|校验值|

获取接收：|0xBC|CMD|CTRL(03)|0x20|data1|data2........data32|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

17.消息推送（安卓）

EVT\_TYPE\_ALERT\_MSG = 11

typedef enum {

MSG\_OTHER = 0,

MSG\_CALL,

MSG\_MISSCALL,

MSG\_SMS,

MSG\_WECHAT,

MSG\_QQ,

MSG\_FACEBOOK,

MSG\_MESSENGER,

MSG\_TWITTER,

MSG\_INSTAGRAM,

MSG\_SKYPE,

MSG\_WHATSAPP,

MSG\_LINE,

MSG\_EMAIL,

MSG\_WEIBO,

MSG\_SNAPCHAT,

MSG\_GMAIL,

MSG\_LINKEDIN,

MSG\_TALK,

MSG\_TELEGRAM,

MSG\_PLAYER = 20, //歌词，安卓使用

MSG\_REMOVE\_CALL = 0xFF,

} phone\_app\_id\_t;

typedef struct bcs\_message\_t{

phone\_app\_id\_t type; //消息类型

uint8\_t title\_len; //实际title长度

uint8\_t content\_len; //实际content长度

uint8\_t data[220]; //data规则为："消息标题: 消息内容" 。

//中间的冒号和空格需要APP主动加上

//消息内容+标题总大小220字节

来电话："联系人名称: 联系人电话号码"（名称超过18byte，请主动截断）

}

消息组装: 消息类型+标题长度+内容长度+消息标题+: +消息内容

歌词组装: 消息类型+标题长度+内容长度+播放状态+音量+歌名+歌词

播放状态：0暂停 1播放

音量：0~9 十个等级

发送：|0xBC|CMD|CTRL|LEN|DATA|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

18.设置亮屏时长

EVT\_TYPE\_DISPLAY\_TIME = 77,/\* 亮屏时长 \*/

note：例如：亮屏5秒，0x05，表示亮屏时间（秒，一个字节）

发送：|0xBC|CMD|CTRL|01|0x05|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

19.获取手环配置

EVT\_TYPE\_BAND\_CONFIG = 78,/\* 手环配置 \*/

//用户数据格式

typedef struct user\_data

{

uint8\_t sex; //性别

uint8\_t height; //身高

uint8\_t weight; //体重

uint8\_t age; //年龄

uint8\_t day; //日

uint8\_t month; //月

uint16\_t year; //年

}user\_data\_t,\*user\_data\_p;

note:(此结构不要随意改动),没有结构版本，后期有需要再加

typedef struct watch\_config\_info

{

uint32\_t TimeStamp; //时间戳(app 转换时区后的)

user\_data\_t user\_data; //用户信息 身高体重等

uint8\_t IsBind; //绑定状态 (暂时预留)

uint8\_t ResetfalgCount ; //异常重启次数

uint8\_t Language; //当前语言ID

uint8\_t CurrentDialID; //当前表盘号

uint8\_t TotleDialCount; //总共表盘个数（除自定义表盘）

uint8\_t BackLightLevel; //背光等级 (1~6,6个等级)

uint8\_t TimeFormat; //时间格式（24小时/12小时）

uint8\_t UnitFormat; //单位格式（0：公制或 1：英制）

uint8\_t TempFormat; //温度格式（0：摄氏度:1：华氏度）

uint8\_t DisplayTime; //亮屏时长（最大 255 秒）

uint8\_t VibrationLevel; //震动等级 0:不震动 1:弱 2:强

uint8\_t PowerSave; //省电模式（震动 + 背光 + 亮屏时长）

uint8\_t Weather; //天气开关 （0：关 1：开）

uint8\_t AllDayHr; //全天HR开关（0：关 1：开）

uint8\_t AllDayBo2; //全天BO2开关 （0：关 1：开）

uint8\_t AllDayTemp; //全天TEMP开关 （0：关 1：开）

uint32\_t MsgNotifyFlag; //消息和电话通知(每个 Bit 代表一个开关，bit0 代表电话通知)

uint32\_t Passward; //密码

uint8\_t AppSystemType; //系统类型 1 ：ios 2 ：安卓

uint8\_t AppBtMac[6]; //手机系统蓝牙 MAC

} watch\_config\_info\_t,\*watch\_config\_info\_p;

note:此api 只用于蓝牙连接成功后，获取数据。不能通过此接口设置数据。接收数据对应结构

发送：|0xBC|CMD|CTRL|01|00|校验值|

接收：|0xBC|CMD|CTRL|data1...........|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

20.设置/获取睡眠数据

EVT\_TYPE\_SLEEP\_DATA = 79,/\* 睡眠数据 \*/

固定睡眠区间：22:00 ~8:00

睡眠状态类型：

SLEEP\_TYPE\_WAKE = 0,//清醒

SLEEP\_TYPE\_LIGHT = 1,//浅睡

SLEEP\_TYPE\_DEEP = 2,//深睡

发送：|0xBC|CMD|CTRL|01|00|校验值| //请求同步睡眠数据

返回：

1.没有要同步睡眠数据：|0xBC|CMD|CTRL|01|00|校验值|

2.有数据：|0xBC|CMD|CTRL|len|清醒时间4字节|浅睡时间4字节|深睡时间4字节|入睡时间戳4字节|醒来时间戳4字节|睡眠状态|时间|......|校验值|

note：请根据入睡时间戳，自行对应哪天数据

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

21.获取运动记录

EVT\_TYPE\_SPORT\_RECORD = 80,/\* 运动记录 \*/

运动类型 参考 sport.h

uint8\_t SportType; //运动类型

uint8\_t SportHr; //运动hr

uint8\_t SportAvgHr; //运动平均hr

uint8\_t SportCustom; //该字节作为预留

uint32\_t StartTimeStamp; //开始时间戳

uint32\_t EndTimeStamp; //结束时间戳

uint32\_t SportTime; //运动时长（秒）

uint32\_t SportSteps; //运动步数

uint32\_t SportKcal; //运动卡路里

uint32\_t SportDistance; //运动距离

note：支持获取最近10次运动记录，会同步所有没有同步的记录数据，

如果app ，需要排序，请以结束时间戳进行排序(固件端使用结束时间戳排序)

发送：|0xBC|CMD|CTRL|01|00|校验值|

返回：

有运动记录：|0xBC|CMD|CTRL(0x03)|len||运动记录数据|校验|

没有最近记录：|0xBC|CMD|CTRL(0x03)|01|00|校验|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

22.设置/获取运动数据

EVT\_TYPE\_SPORT\_DATA = 81,/\* 运动数据 \*/

uint32\_t sport\_steps;

uint32\_t sport\_distance;

uint32\_t sport\_kcal;

uint32\_t sport\_time;

note：每个值为4个字节，同时收发16字节数据

发送：|0xBC|CMD|CTRL|0x10|sport\_steps|sport\_distance|sport\_kcal|sport\_time|校验值|

返回：

|0xBC|CMD|CTRL(0x03)|0x10|sport\_steps|sport\_distance|sport\_kcal|sport\_time|校验值| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

23.设置天气预报

EVT\_TYPE\_WEATHER\_FORECAST = 82,/\* 天气预报 \*/

//数据类型

typedef enum {

WIND\_NONE = 0, //无

WIND\_EAST = 1, //东

WIND\_SOUTHEAST = 2, //东南

WIND\_SOUTH = 3, //南

WIND\_SOUTHWEST = 4, //西南

WIND\_WEST = 5, //西

WIND\_NORTHWEST = 6, //西北

WIND\_NORTH = 7, //北

WIND\_NORTHEAST = 8, //东北

} wind\_type;

//天气预报  
typedef struct bcs\_weather\_forecast {  
uint8\_t week; //星期  
uint8\_t type; //天气类型  
int8\_t temp\_high; //高温  
int8\_t temp\_low; //低温  
uint16\_t pressure; //大气压 hpa  
uint8\_t ult\_level; //紫外线等级  
uint8\_t humidity; //湿度  
uint8\_t wind\_dir; //风向  
uint8\_t wind\_lvl; //风等级   
uint16\_t visibility; //能见度 米  
uint16\_t precipitation; //降雨量 毫米  
uint8\_t reserve; //预留  
uint8\_t reserve1; //预留1  
} weather\_forecast\_t, \*weather\_forecast\_p;

0x00 未知

0x01 晴天

0x02 多云

0x03 阴天

0x04 阵雨

0x05 雷阵雨、雷阵雨伴有冰雹

0x06 小雨

0x07 中雨、大雨、暴雨

0x08 雨加雪、冻雨

0x09 小雪

0x0a 大雪、暴雪

0x0b 沙尘暴、浮尘

0x0c 雾、雾霾

发送：|0xBC|CMD|CTRL|len|data\*7|校验值|

返回：此处返回表示请求（如果手表已有天气预报数据，则不会主动请求）

|0xBC|CMD|CTRL(0x03)|01|01|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

24.设置实时天气

EVT\_TYPE\_REAL\_TIME\_WEATHER = 83,/\* 实时天气 \*/

//数据类型

uint8\_t type;//天气类型

int8\_t cur\_temp;//当前温度

note:温度为范围（50 ~ -50摄氏度），只发摄氏度即可，每个值为一个字节

天气类型，与天气预报中一致

发送：|0xBC|CMD|CTRL|02|type|cur\_temp|校验值|

返回：此处返回表示请求（如果手表已有实时天气数据，则不会主动请求）

|0xBC|CMD|CTRL(0x03)|01|01|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

25.设置抬腕亮屏

EVT\_TYPE\_RAISE\_WRIST = 84,/\* 抬腕亮屏 \*/

//数据类型

uint8\_t state; //开关状态 1：打开 0：关闭

uint8\_t begin\_hour;

uint8\_t begin\_min;

uint8\_t end\_hour;

uint8\_t end\_min;

note: 5个字节数据

发送：|0xBC|CMD|CTRL|05|state|begin\_hour|begin\_min|end\_hour|end\_min|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

26.设置勿扰模式

EVT\_TYPE\_DISTURB = 85,/\* 勿扰模式 \*/

//数据类型

uint8\_t state; //开关状态 1：打开 0：关闭

uint8\_t begin\_hour;

uint8\_t begin\_min;

uint8\_t end\_hour;

uint8\_t end\_min;

note: 5个字节数据

发送：|0xBC|CMD|CTRL|05|state|begin\_hour|begin\_min|end\_hour|end\_min|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

27.设置久坐提醒

EVT\_TYPE\_LONG\_SIT = 86,/\* 久坐提醒 \*/

//数据类型

uint8\_t state; //开关状态 1：打开 0：关闭

uint8\_t interval; //周期间隔(分钟)

uint8\_t begin\_hour;

uint8\_t begin\_min;

uint8\_t end\_hour;

uint8\_t end\_min;

note: 6个字节数据

发送：|0xBC|CMD|CTRL|06|state|interval|begin\_hour|begin\_min|end\_hour|end\_min|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

28.设置喝水提醒

EVT\_TYPE\_DRINK\_WATER = 87,/\* 喝水提醒 \*/

//数据类型

uint8\_t state; //开关状态 1：打开 0：关闭

uint8\_t interval; //周期间隔(分钟)

uint8\_t begin\_hour;

uint8\_t begin\_min;

uint8\_t end\_hour;

uint8\_t end\_min;

note: 6个字节数据

发送：|0xBC|CMD|CTRL|06|state|interval|begin\_hour|begin\_min|end\_hour|end\_min|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

29.设置洗手提醒

EVT\_TYPE\_WASH\_HAND = 88,/\* 洗手提醒 \*/

//数据类型

uint8\_t state; //开关状态 1：打开 0：关闭

uint8\_t interval; //周期间隔(分钟)

uint8\_t begin\_hour;

uint8\_t begin\_min;

uint8\_t end\_hour;

uint8\_t end\_min;

note: 6个字节数据

发送：|0xBC|CMD|CTRL|06|state|interval|begin\_hour|begin\_min|end\_hour|end\_min|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

30.设置日程提醒

EVT\_TYPE\_SCHEDULE = 89,/\* 日程提醒 \*/

该指令预留

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

31.设置/获取闹钟

EVT\_TYPE\_ALARM = 90,/\* 闹钟 \*/

//数据类型

/\*

\_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_

| | | | | | | | |

| |周六|周五|周四|周三|周二|周一|周日|

|\_\_\_\_|\_\_\_\_|\_\_\_\_|\_\_\_\_|\_\_\_\_|\_\_\_\_|\_\_\_\_|\_\_\_\_|

\*/ uint8\_t state; //开关状态

alarm\_remind\_t type; //0 ：没有 1 : 单次，2 ：每天，3自定义

uint8\_t week\_day; //自定义7天，低7位表示(如上图)

uint8\_t hour; //(0~23,24小时制)

uint8\_t min; //(0~59)

note: 6个字节数据 ,最大4个闹钟alarm\_id （0~3）

//设置或获取单个闹钟数据

发送：|0xBC|CMD|CTRL|06|alarm\_id|state|type|week\_day|hour|min|校验值|

接收：|0xBC|CMD|CTRL(0x03)|06|alarm\_id|state|type|week\_day|hour|min|校验值|

//获取全部闹钟数据（当前共四个）

发送：|0xBC|CMD|CTRL|01|FF|校验值|

接收：|0xBC|CMD|CTRL(0x03)|15|FF|

state|type|week\_day|hour|min| //闹钟1

state|type|week\_day|hour|min| //闹钟2

state|type|week\_day|hour|min| //闹钟3

state|type|week\_day|hour|min|校验值| //闹钟4

//设置全部闹钟数据（当前共四个，如果只有一个闹钟，其他数据补0）

例如：已存在闹钟1和闹钟2，此时用户删除闹钟1；请把闹钟2的数据copy到闹钟1，其他数据赋值0。

发送：|0xBC|CMD|CTRL|15|FF|

state|type|week\_day|hour|min| //闹钟1

state|type|week\_day|hour|min| //闹钟2

state|type|week\_day|hour|min| //闹钟3

state|type|week\_day|hour|min|校验值| //闹钟4

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

32.获取体温/心率/血压/血氧

EVT\_TYPE\_TEMP = 91,/\* 体温 \*/

EVT\_TYPE\_HR = 92,/\* 心率 \*/

EVT\_TYPE\_BP = 93,/\* 血压 \*/

EVT\_TYPE\_BO = 94,/\* 血氧 \*/

note：同步单次测量的数据

发送：|0xBC|CMD|CTRL|01|00|校验值|

/\* 体温 \*/返回：|0xBC|CMD|CTRL(0x03)|02|10倍体温值|校验值|

note：体温36.5，则会传365，体温37.0，则会传370.

/\* 心率 \*/返回：|0xBC|CMD|CTRL(0x03)|01|心率值|校验值|

/\* 血压 \*/返回：|0xBC|CMD|CTRL(0x03)|02|高压值|低压值|校验值|

/\* 血氧 \*/返回：|0xBC|CMD|CTRL(0x03)|01|血氧值|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

33.获取7天的全天心率/全天血氧/全天体温

EVT\_TYPE\_HR\_DAY = 95,/\* 全天心率 \*/

EVT\_TYPE\_BO\_DAY = 96,/\* 全天血氧 \*/

EVT\_TYPE\_TEMP\_DAY = 97,/\* 全天体温 \*/

note1: date = 0 : 今天

date = 1: 昨天

date = 2: 前天（data：0~6）

note2: 每半小时一个数据(综合考虑)，全天数据每天48个字节数据

分别对应时间轴：（00:00 ，00:30......23:00,23:30）

发送：|0xBC|CMD|CTRL|01|date |校验值|

/\* 全天心率 \*/返回：|0xBC|CMD|CTRL(0x03)|0x30|全天心率48字节|校验值|

/\* 全天血氧 \*/返回：|0xBC|CMD|CTRL(0x03)|0x30|全天血氧48字节|校验值|

/\* 全天体温 \*/返回：|0xBC|CMD|CTRL(0x03)|0x30|全天体温48字节|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

34.设置全天测量开关

EVT\_TYPE\_ALL\_DAY\_FALG = 98,/\* 全天测量开关 \*/

开关类型

switch\_hr = 1;

switch\_bo2 = 2;

switch\_temp = 3;

开关状态

1：打开

0：关闭

note:2个字节

发送：|0xBC|CMD|CTRL|02|开关类型|开关状态|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

35.设置hr/bp/bo2数据（单次）

EVT\_TYPE\_HR\_BP\_BO2 = 99,/\* 测试设置hr/bp/bo2 \*/

//数据类型

uint8\_t hr; //心率

uint8\_t spo2; //血氧

uint8\_t sbp; //收缩压100~120（mmHg）

uint8\_t dbp; //舒张压60~80（mmHg）

note: 4个字节数据

发送：|0xBC|CMD|CTRL|04|hr|spo2|sbp|dbp|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

36.拍照

EVT\_TYPE\_TAKE\_PHOTO = 100,/\* 拍照 \*/

控制类型

CAMERA\_TYPE\_CLOSE = 0,/\* 关闭 \*/

CAMERA\_TYPE\_START = 1,/\* 拍照 \*/

CAMERA\_TYPE\_OPEN = 2,/\* 打开 \*/

note: 1个字节数据

发送：|0xBC|CMD|CTRL|01|控制类型|校验值|

接收：|0xBC|CMD|CTRL(0x03)|01|控制类型|校验值|(表示请求)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

37.音乐控制（安卓）固件端请求

EVT\_TYPE\_CTRL\_MUSIC = 101,/\* 音乐控制 \*/

enum BCS\_PLAYER //控制类型（1byte）

{

PLAYER\_TYPE\_PAUSE = 0,/\* 暂停 \*/

PLAYER\_TYPE\_PLAY = 1,/\* 播放 \*/

PLAYER\_TYPE\_PREV = 2,/\* 上一曲 \*/

PLAYER\_TYPE\_NEXT = 3,/\* 下一曲 \*/

PLAYER\_TYPE\_VOLUME\_UP = 4,/\* 音量加 \*/

PLAYER\_TYPE\_VOLUME\_DOWN = 5,/\* 音量减 \*/

};

note：后期如有歌词等信息同步，后续增加控制类型表示同步

接收：|0xBC|CMD|CTRL(0x03)|01|控制类型|校验值|(表示请求)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

38.手环配置1(结构格式，固定后不要乱改)

EVT\_TYPE\_BAND\_CONFIG1 = 102,/\* 手环配置1 \*/

//全天的运动数据格式

typedef struct all\_sport\_data

{

uint32\_t SportSteps;//运动步数[4byte]

uint32\_t SportKcal;//运动卡路里[4byte]

uint32\_t SportDistance;//运动距离[4byte]

uint32\_t Sporttime;//运动时间[4byte]

uint32\_t TargetSteps; //目标步数[4byte]

uint16\_t PerHourStep[24];//24小时分时步数[48byte]

}all\_sport\_data\_t,\*all\_sport\_data\_p;

//结构性数据存储

typedef struct watch\_cycle\_info

{

all\_sport\_data\_t all\_sport\_data; //全天运动数据[68byte]

alarm\_t alarm[4]; //闹钟[20byte]

remind\_cycle\_t long\_sit; //久坐提醒[8byte]

remind\_cycle\_t drink\_water; //喝水提醒[8byte]

remind\_cycle\_t wash\_hand; //洗手提醒[8byte]

range\_switch\_t lift\_hand; //翻腕亮屏[6byte]

range\_switch\_t disturb; //勿扰模式[6byte]

medic\_switch\_t medic; //吃药提醒[8byte]

}watch\_cycle\_info\_t,\*watch\_cycle\_info\_p;

发送：|0xBC|CMD|CTRL|01|00|校验值|

note:返回固定132个字节

接收：|0xBC|CMD|CTRL(0x03)|7c|data1........data132|校验值|

(1).前68字节，在all\_sport\_data\_t该结构中

(2).闹钟固定 4 个，每个结构5 byte，共20byte。与30条 30.设置闹钟相互验证。

/\*

\_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_

| | | | | | | | |

| |周六|周五|周四|周三|周二|周一|周日|

|\_\_\_\_|\_\_\_\_|\_\_\_\_|\_\_\_\_|\_\_\_\_|\_\_\_\_|\_\_\_\_|\_\_\_\_|

\*/

typedef struct bcs\_alarm

{

uint8\_t state; //开关状态

alarm\_remind\_t type; //0 ：没有 1 : 单次，2 ：每天，3自定义

uint8\_t week\_day; //自定义7天，低7位表示(如上图)

uint8\_t hour; //(0~23,24小时制)

uint8\_t min; //(0~59)

}alarm\_t,\*alarm\_p;

(3).久坐/喝水/洗手提醒，是同一个结构，每个占8字节，共24个字节

typedef struct bcs\_remind\_cycle

{

uint8\_t state; //开关状态 1：打开 0：关闭

uint8\_t interval; //周期间隔(分钟)

uint8\_t begin\_hour; //23

uint8\_t begin\_min; //50

uint8\_t end\_hour; //23

uint8\_t end\_min; //30

uint8\_t test1;

uint8\_t test2;

} remind\_cycle\_t,\*remind\_cycle\_p;

note:每个结构前六个字节是有用的，test1与test2对于app无用。

与 26条，27条，28条 相互印证

(4).翻腕亮屏/勿扰模式是同一个结构，每个占6字节，共12个字节

typedef struct bcs\_range\_switch

{

uint8\_t state; //开关状态 1：打开 0：关闭

uint8\_t allow; //在区间内

uint8\_t begin\_hour; //23

uint8\_t begin\_min; //50

uint8\_t end\_hour; //23

uint8\_t end\_min; //30

} range\_switch\_t,\*range\_switch\_p;

note:与24条，25条 相互印证

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

39.设置手机系统类型

EVT\_TYPE\_PHONE\_SYSTEM\_TYPE = 103,/\* 设置手机系统类型 \*/

type (1byte) ：(1 ：ios) (2 : 安卓)

发送：|0xBC|CMD|CTRL(0x02)|01|type |校验值|

没有返回

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

40. 同步MAC地址（安卓)

EVT\_TYPE\_APP\_MAC = 104,/\* 安卓同步mac \*/

发送：|0xBC|CMD|CTRL(0x02)|06|mac1|mac2|mac3|mac4|mac5|mac6|校验值|

没有返回

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

41.APP 请求同步基础数据

EVT\_TYPE\_APP\_REQUEST\_SYNC = 105,/\* APP请求同步基础数据同步 \*/

//同步基础数据格式

typedef struct basic\_syn\_data

{

uint32\_t Steps; //运动步数 4Byte

uint32\_t Kcal; //运动卡路里4Byte

uint32\_t Distance; //运动距离4Byte

uint8\_t Hr; //心率1Byte

uint8\_t Spo2; //血氧1Byte

uint8\_t Sbp; //收缩压（mmHg）1Byte

uint8\_t Dbp; //舒张压（mmHg）1Byte

uint8\_t Ecg; //心电

uint8\_t test3; //预留

int16\_t Temp; //体温(2byte)(实际温度\*10倍的值)

}basic\_syn\_data\_t,\*basic\_syn\_data\_p;

note:此结构目前 20字节

发送：|0xBC|CMD|CTRL(0x01)|01|00校验值|

接收：|0xBC|CMD|CTRL(0x03)|0x10|data1........data16|校验值|

42. 查找手机

EVT\_TYPE\_FIND\_PHONE = 106,/\*查找手机 \*/

type： 1 ： 开始查找 2 ：结束查找

接收：|0xBC|CMD|CTRL(0x03)|01|type|校验值|(表示请求)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

43. 全天血压

EVT\_TYPE\_BP\_DAY = 107,/\*全天血压\*/

note1: date = 0 : 今天

date = 1: 昨天

date = 2: 前天（data：0~6）

note2: 每半小时一组数据【高血压|低血压】全天数据每天96个字节数据

分别对应时间轴：（00:00 ，00:30......23:00,23:30）

发送：|0xBC|CMD|CTRL|01|date |校验值|

/\* 全天血压 \*/返回：|0xBC|CMD|CTRL(0x03)|0x30|data1...data96|校验值|

时间00:00 = data1：高血压 data2：低血压

时间00:30 = data3：高血压 data4：低血压

时间01:00 = data5：高血压 data6：低血压

.....

44.交友码

EVT\_TYPE\_ADD\_FRIEND = 108,/\*交友码\*/

type：

APP\_WECHAT = 0,

APP\_QQ,

APP\_SKYPE,

APP\_FACEBOOK,

APP\_TWITTER,

APP\_WHATS,

APP\_LINE,

APP\_INS,

APP\_MESSENGER,

APP\_SNAPCHAT,

note : len < 230Byte

发送：|0xBC|CMD|CTRL(0x02)|len|type|data1....data n|校验值|

接收：|0xBC|CMD|CTRL(0x03)|01|type||校验值| //确认接收成功

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

45.收款码

EVT\_TYPE\_RECEIPT\_CODE = 109,/\*收款码\*/

type：

PAYMENT\_WEPAY = 0,

PAYMENT\_ALIPAY,

PAYMENT\_QQ,

PAYMENT\_PAYPAL,

note : len < 230Byte

发送：|0xBC|CMD|CTRL(0x02)|len|type|data1....data n|校验值|

接收：|0xBC|CMD|CTRL(0x03)|01|type||校验值| //确认接收成功

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

46.女性健康

EVT\_TYPE\_WOMEN\_HEALTH = 110,/\*女性健康\*/

NOTE: 共14个字节

data1：功能开关状态 0 ：关闭 1：打开

data2：类型 1：经期 2：备孕 3：孕中（目前只支持经期，所以 ：1可以写死）

data3：提前天数 （提前1天 就是 1）（提前5天就是5）（不提前 则为 0）

data4：提醒开关 0 ：关闭 1：打开

data5：经期持续时间（天）

data6：周期 （天）

data7 ~ data10：上一次时间戳

data11 ~ data14：提醒时间（提醒当天的秒数，从0点开始，例如提醒时间为 01:00 ，此值为 3600）

发送：|0xBC|CMD|CTRL(0x02)|0E|data1....data14|校验值|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

47.android 挂断电话或接听电话

EVT\_TYPE\_ANDROID\_PHONE\_CTRL: /\* 安卓电话挂断/接听 \*/

发送：|0xBC|CMD|CTRL(0x02)|len|0|校验值| //挂断

发送：|0xBC|CMD|CTRL(0x02)|len|1|校验值| //接听

接收：|0xBC|CMD|CTRL(0x03)|len|0|校验值| //挂断

接收：|0xBC|CMD|CTRL(0x03)|len|1|校验值| //接听

48.解除绑定（固件端清除所有数据）

EVT\_TYPE\_UNBIND = 112, /\* 解除绑定\*/

发送：|0xBC|CMD|CTRL(0x02)|01|00|校验值|

49.心电同步

EVT\_TYPE\_HR\_ECG = 120, /\* 心电指令\*/

type ：0 : 测量中 1:未佩戴 2: 测量完成结果 3.固件主动停止测量

接收：|0xBC|CMD|CTRL(0x03)|02|type|ecg\_data|校验值|

50.同步固件端支持功能+设备屏幕信息

EVT\_TYPE\_DEVICE\_FUN = 121,

结构在文件中“设备支持功能结构.h”

发送：|0xBC|CMD|CTRL(0x01)|01|00|校验值|

接收：|0xBC|CMD|CTRL(0x03)|len|设备支持功能结构数据|校验值|

51.同步3.0 MAC（该指令不需APP发指令），ios不需要处理

EVT\_TYPE\_WATCH\_CALL\_INFO = 122, /\* 同步3.0MAC +名称\*/

note：mac 固定为6byte，len = mac + name。

接收：|0xBC|CMD|CTRL(0x03)|len|mac+name|校验值|

52.吃药提醒

EVT\_TYPE\_MEDIC = 123, /\* 吃药提醒 \*/

{

uint8\_t state; //开关状态 1：打开 0：关闭

uint8\_t week\_day; //星期（遵守闹钟星期规则）

uint8\_t data[6]; //3组 [hour,min,hour,min,hour,min,]

}

note: 固定8 byte,固件初始值 【0,0,0xff,0xff,0xff,0xff,0xff,0xff】

接收：|0xBC|CMD|CTRL(0x02)|len|数据|校验值|

53.GPS 运动

EVT\_TYPE\_GPS\_SPORT = 124, /\* GPS运动 \*/

详情查看 “<GPS运动.h>”

54.海拔

EVT\_TYPE\_ALTITUDE = 125, /\* 海拔 \*/

note：海拔值 4个字节 int32

发送：|0xBC|CMD|CTRL(0x02)|04|海拔值|校验值|

55.十米葫芦(体感游戏)

EVT\_TYPE\_MOTION\_GAME = 126, /\* 体感游戏 \*/

发送：|0xBC|CMD|CTRL(0x02)|01| 0 |校验值| //开始游戏

发送：|0xBC|CMD|CTRL(0x02)|01| 1 |校验值| //结束游戏

发送：|0xBC|CMD|CTRL(0x02)|01| 2 |校验值| //暂停游戏

Type：0：游戏开始 1：游戏结束 2：游戏暂停 3：传输游戏数据

接收：|0xBC|CMD|CTRL(0x03)|01| Type |校验值| //游戏状态

接收：|0xBC|CMD|CTRL(0x03)|len| Type + 游戏数据 |校验值| //游戏数据

游戏数据：

{

int16\_t x; //2byte

int16\_t y; //2byte

int16\_t Speed; //2byte

int16\_t X\_Throw; //2byte

int16\_t Y\_Throw; //2byte

int16\_t Speed\_Throw; //2byte

int8\_t Count\_Throw; //1byte

int8\_t test; //预留1byte

int16\_t X\_Move\_Distance; //2byte

int16\_t Y\_Move\_Distance; //2byte

int16\_t X\_gravity; //2byte

int16\_t Y\_gravity; //2byte

int16\_t Z\_gravity; //2byte

}

Note: 游戏数据固定24 byte

56.获取SD歌曲名称列表（仅支持杰里二代)

EVT\_TYPE\_SD\_MUSIC\_LIST = 127, /\*本地音乐列表名称\*/

note：cur\_id: 该条列表所在id，total\_id:总列表数量，name:文件名称（unicode）

note：cur\_id: (2byte),total\_id:(2byte), del\_id(2byte),name: (200byte 以内)

type = 0 //删除

发送：|0xBC|CMD|CTRL(0x01)|01| 00 |校验值|

发送：|0xBC|CMD|CTRL(0x02)|len|type| del\_id|name |校验值|

接收：|0xBC|CMD|CTRL(0x03)|len| cur\_id|total\_id|name|校验值|

57.获取SD电子书名称列表（仅支持杰里二代)

EVT\_TYPE\_SD\_BOOK\_LIST = 128, /\*本地电子书列表名称\*/

note：cur\_id: 该条列表所在id，total\_id:总列表数量，name:文件名称（unicode）

note：cur\_id: (2byte),total\_id:(2byte), del\_id(2byte),name: (200byte 以内)

type = 0 //删除

发送：|0xBC|CMD|CTRL(0x01)|01| 00 |校验值|

发送：|0xBC|CMD|CTRL(0x02)|len|type| del\_id|name |校验值|

接收：|0xBC|CMD|CTRL(0x03)|len| cur\_id|total\_id|name|校验值|

58.获取SD卡剩余空间（仅支持杰里二代)

EVT\_TYPE\_SD\_FREE\_SPACE = 129, /\*SD卡剩余空间\*/

note：space（4byte，单位 KB）

发送：|0xBC|CMD|CTRL(0x01)|01| 00 |校验值|

接收：|0xBC|CMD|CTRL(0x03)|len| space|校验值|

59.获取SD录音名称列表（仅支持杰里二代)

EVT\_TYPE\_SD\_RECORD\_LIST = 130, /\*本地录音列表名称\*/

note：cur\_id: 该条列表所在id，total\_id:总列表数量，name:文件名称（unicode）

note：cur\_id: (2byte),total\_id:(2byte), del\_id(2byte),name: (200byte 以内)

time\_s (时长秒，4byte)

type = 0 //删除

发送：|0xBC|CMD|CTRL(0x01)|01| 00 |校验值|

发送：|0xBC|CMD|CTRL(0x02)|len|type| del\_id|name |校验值|

接收：|0xBC|CMD|CTRL(0x03)|len| cur\_id|total\_id|time\_s|name|校验值|

60.设置用户名称（仅支持杰里二代)

EVT\_TYPE\_USER\_NAME = 131, /\*用户名称\*/

note: name (100Byte)

发送：|0xBC|CMD|CTRL(0x02)|len| name |校验值|

61.获取用户头像KEY（仅支持杰里二代)

EVT\_TYPE\_USER\_HEAD\_LOGO\_KEY = 132, /\*户头像KEY\*/

发送：|0xBC|CMD|CTRL(0x01)|01| 00|校验值|

接收：|0xBC|CMD|CTRL(0x03)|len| KEY(4B)|校验值|

99.OTA 大数据到手表(每包数据不能超过230Byte)

EVT\_TYPE\_OTA\_START = 200,/\* OTA 开始指令 \*/

EVT\_TYPE\_OTA\_DATA = 201,/\* OTA 数据指令 \*/

EVT\_TYPE\_OTA\_END = 202,/\* OTA 结束 \*/

note：联系人格式（total\_size = 条数 \* 40B）

每条联系人固定为40byte，人名 ： 20字节，号码 ： 20字节

|len(1Byte)| 19字节人名，UTF-8编码

|len(1Byte)| 19字节电话号码，字符数字如：+138则为0x2b0x310x330x38

note:自定义表盘格式，total\_size:头文件格式+背景图+ 缩略图

【头文件格式 | 背景图（不压缩）| 缩略图（不压缩）】

头文件格式：【byte1|byte2|byte3|byte4|byte5|byte6|byte7|byte8

|byte9|byte10|byte11|byte12|byte13|byte14|byte15|byte16】

byte1: 时间位置 0 :上方 1 :下方

byte2: 时间上方位置显示0 :关闭 1 :心率2 :睡眠 3 :步数4 :日期

byte3: 时间下方位置显示0 :关闭 1 :心率2 :睡眠 3 :步数4 :日期

byte4: 字体颜色R

byte5: 字体颜色G

byte6: 字体颜色B

byte7: 视频表盘字节（2~5，固定最多5帧)

byte8: 预留2

byte9+byte10 : 屏幕 高

byte11+ byte12 : 屏幕 宽

byte13+byte14 : 预览图 高

byte15+ byte16 : 预览图 宽

byte7: bit0~bit3表示帧数，bit4 = 1 固件主动切换，bit4 = 0 固件自动切换

例：5帧 自动切换 byte7= 0x05，

5帧 主动切换 byte7= 0x15，

【头文件格式 | 背景图1（不压缩）.......背景图5（不压缩）| 缩略图（不压缩）】

流程如下：

ota\_type：//uint8\_t

(0 ： 自定义表盘传输)

(1 ： 表盘传输)

(2： 联系人传输)

(3： 全部UI传输)

(4 ： 游戏)

(5： 核酸码)

(6： 运动轨迹)

(7： 开机LOGO)（预留)

(8： 内置表盘)（安卓工具)

(9： 升级字库)（安卓工具)

(10： 固件升级)（仅支持杰里二代)

(11: 用户头像)（仅支持杰里二代)

头像格式：[宽(2B)+高(2B)+图片key(4B)+data]

SD卡：

(50: 音乐)（仅支持杰里二代)

(51:电子书)（仅支持杰里二代)

total\_size://4字节，数据总大小（字节）

total\_pack\_count：总包数，2字节

一.准备升级

1.APP首先发送开始OTA（CTRL = EVT\_TYPE\_OTA\_START）

传文件时：文件名称(unicode，长度200byte，带文件格式)

发送：|0xBC|CMD|CTRL(0x02)|len|ota\_type(1byte)|total\_size(4byte)|total\_pack\_count(2byte)|文件名称(unicode)|校验值|

2.等待手表返回，已经做好准备ota\_type升级(准备时间与数据大小成正比， 实测3M，擦除时间为 6.5 秒)

接收：|0xBC|CMD|CTRL(0x03)|07|ota\_type(1byte)|total\_size(4byte)|total\_pack\_count(2byte)|校验值|

3.校验接收的数据，与发送的完全一致，则开始发送数据，否则重新发送

二.升级开始（CTRL = EVT\_TYPE\_OTA\_DATA）

1.APP发送数据

发送：|0xBC|CMD|CTRL(0x02)|len|data1|data2|.......|data n|校验值|

(data1 + data2) :当前包ID ，data3~datan 为实际数据（实际数据按照4的倍数，最后一包，不用管是不是4的倍数），len = data1 ~ datan

2.循环60ms发一包，（如果总包数 100，则当前包ID范围（0~99））

三.异常处理，APP均需要处理（CTRL = EVT\_TYPE\_OTA\_END）

enum BCS\_OTA\_END\_TYPE

{

OTA\_TYPE\_SUCCESS = 0,//升级成功

OTA\_TYPE\_FAIL\_1 = 1,//上一包序号与下一包序号不连续

OTA\_TYPE\_FAIL\_2 = 2,//多次写入，均失败

OTA\_TYPE\_FAIL\_3 = 3,//未接收到开始指令，就接收数据

OTA\_TYPE\_FAIL\_4 = 4,//超时时间内未正确接收到数据

OTA\_TYPE\_FAIL\_5 = 5,//主动结束升级

OTA\_TYPE\_REPAIR = 6,//修复包id序号

OTA\_TYPE\_FAIL\_7 = 7,//表盘太大，无法升级

//SD传输异常

OTA\_TYPE\_SD\_SUCCESS = 20,//检测成功(该条状态，不发送)

OTA\_TYPE\_SD\_FAIL\_1 = 21,//删除临时文件失败

OTA\_TYPE\_SD\_FAIL\_2 = 22,//获取剩余空间失败

OTA\_TYPE\_SD\_FAIL\_3 = 23,//剩余空间不足

OTA\_TYPE\_SD\_FAIL\_4 = 24,//创建临时文件失败

OTA\_TYPE\_SD\_FAIL\_5 = 25,//当前文件已经达到预设最大数量

OTA\_TYPE\_SD\_FAIL\_6 = 26,//多次fwrite失败

OTA\_TYPE\_SD\_FAIL\_7 = 27,//文件重命名失败

};

//接收：|0xBC|CMD|CTRL(0x03)|03|OTA\_TYPE\_FAIL\_2(1byte)|00|00|校验值|

//接收：|0xBC|CMD|CTRL(0x03)|03|OTA\_TYPE\_FAIL\_3(1byte)|00|00|校验值|

//接收：|0xBC|CMD|CTRL(0x03)|03|OTA\_TYPE\_FAIL\_4(1byte)|00|00|校验值|

//接收：|0xBC|CMD|CTRL(0x03)|03|OTA\_TYPE\_FAIL\_5(1byte)|00|00|校验值|

接收到以上指令，立即结束升级，升级失败

//接收：|0xBC|CMD|CTRL(0x03)|03|OTA\_TYPE\_SUCCESS(1byte)|00|00|校验值|

接收到以上指令，立即结束升级，升级成功

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//接收：|0xBC|CMD|CTRL(0x03)|03|OTA\_TYPE\_FAIL\_1(1byte)|pack\_id(2byte)|校验值|

接收到以上指令，接收数据时，丢包了，最后正确接收到的包pack\_id.

步骤一：立即停止原有发送任务包，发送一个pack\_id + 1 的包

步骤二：开启5秒钟定时器，等待修复包OTA\_TYPE\_REPAIR，在5秒钟内

OTA\_TYPE\_FAIL\_2 = 2,//多次写入，均失败

OTA\_TYPE\_FAIL\_3 = 3,//未接收到开始指令，就接收数据

OTA\_TYPE\_FAIL\_4 = 4,//超时时间内未正确接收到数据

OTA\_TYPE\_FAIL\_5 = 5,//主动结束升级

接收到以上类型，则弹出升级失败

//接收：|0xBC|CMD|CTRL(0x03)|03|OTA\_TYPE\_REPAIR(1byte)|pack1|pack2|校验值|

\*修复包id 等于 pack\_id + 1，就继续循化发送 数据.

\*超过5秒，未有修复包到来，直接发送主动结束升级

OTA\_TYPE\_FAIL\_5 = 5,//主动结束升级

//发送：|0xBC|CMD|CTRL(0x02)|03|OTA\_TYPE\_FAIL\_5(1byte)|00|00|校验值|（必须发）

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

如果蓝牙不断连，则APP一定会收到异常处理的其中一条

100.手表作为master，发送大文件（仅支持杰里二代)

EVT\_TYPE\_MASTER\_START = 203,/\* master 开始指令 \*/

EVT\_TYPE\_MASTER\_DATA = 204,/\* master 数据指令 \*/

EVT\_TYPE\_MASTER\_END = 205,/\* master 结束 \*/

异常类型：(1byte)

MASTER\_TYPE\_FAIL\_1 = 1,//未找到对应文件

MASTER\_TYPE\_FAIL\_2 = 2,//未接收到确认开始指令

传输文件类型：(1byte)

MASTER\_TYPE\_RECORD = 0,//传输录音文件

流程：FILE\_OFFSET（4byte）,SIZE （4byte）

1.APP-->手表（把需要的文件name发给固件）

CMD = EVT\_TYPE\_MASTER\_START

发送：|0xBC|CMD|CTRL(0x02)|len|传输文件类型|name |校验值|

2.手表->APP

//成功找到对应文件

CMD = EVT\_TYPE\_MASTER\_START

接收：|0xBC|CMD|CTRL(0x03)|len| 传输文件类型|SIZE|校验值|

//没有找到对应文件

CMD = EVT\_TYPE\_MASTER\_END

接收：|0xBC|CMD|CTRL(0x03)|len|异常类型|校验值|

3.APP-->手表（APP已确认接收到目标文件大小，并发送需要的数据偏移）

CMD = EVT\_TYPE\_MASTER\_DATA

发送：|0xBC|CMD|CTRL(0x02)|len|FILE\_OFFSET |校验值|

4.手表->APP(循环发送，直到发送完成)

CMD = EVT\_TYPE\_MASTER\_DATA

接收：|0xBC|CMD|CTRL(0x03)|len| FILE\_OFFSET|data|校验值|

5.APP-->手表（文件接收完成）

CMD = EVT\_TYPE\_MASTER\_DATA

此时FILE\_OFFSET总数据大小

发送：|0xBC|CMD|CTRL(0x02)|len|FILE\_OFFSET |校验值|