# 测力通信协议文档V0.5

文档版本历史

|  |  |  |
| --- | --- | --- |
| 版本号 | 修改内容 | 修改时间 |
| V0.1 | 首次完成初稿 | 2018-2-16 |
| V0.2 | 加入枚举设备文件列表命令 | 2018-2-17 |
| V0.3 | 增加包头和包长度 | 2018-2-22 |
| V0.4 | 修改波形包结构 | 2018-2-22 |
| V0.5 | 修改时间包,参数结构。命令号 | 2018-2-22 |
|  |  |  |

## 协议概述

为了通信协议的简单性，传输层采用UDP通信，在应用层收到的每个数据包就应该是一个完整的协议包，不需要应用层再进行分包处理，但是因为UDP层没有对每个包的确认机制，并且因为我们的物理网络是wifi，UDP更容易出现丢包的的情况，因此需要在应用层对每个包进行确认。

* 1. 专用术语

S：服务端，也就是PC程序。

C: 客户端，也就是各个测力设备。

* 1. 数据校验

通信数据采用1字节和校验的方式进行校验。

* 1. 通用协议格式

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 包头 | 包长度 | 设备编号 | 序列号 | 会话号 | 命令号 | 数据包 | 校验和 |
| 2字节 | 2字节 | 4字节 | 2字节 | 2字节 | 1字节 | n字节 | 1字节 |

1.3.1字段介绍

包头：固定为:0xFE 0x7F

包长度:指示整个包的长度，从包头到数据包结尾

设备编号：每个设备唯一编号，应用程序用它来唯一区分是哪个设备，因为用IP不准确

序列号：为了解决UDP传输过程中出现的重复包和丢包现象，应用层加入序号和字段，用来唯一区分一条数据，序列号需要由发送方维护，每次发送不同数据包时递增该序号，到65535后，恢复到0.从新开始。回应包的序列号必须和对应的接收包序号一致。

会话号：对多个设备同一次发送的命令需要进行区分，以便后期的多设备同步处理。目前不需要。

命令号：0x0-0x7F 是从S->C 回应包最高位置1

校验和：计算从设备编号到数据包结束所有数据的校验和

大小端：所有的数据采用小端对齐的方式，以便结构体直接赋值。

## 协议内容

### 2.1 启动发送波形命令

通知设备开始发送最后一次采集的波形数据。

命令编号: 0x1

方向：S->C

数据内容

### 2.2 设备回应启动发送波形命令

命令编号: 0x81

方向：C->S

数据内容：

|  |
| --- |
| 总包数 |
| 4Bytes |

### 2.3 设备发送波形数据

设备收到波形发送命令后，开始发送波形，一次可以发送该设备n个通道

m个样本点的波形数据，UDP协议的时候不要超过64K。以太网MTU的大小

1500，大于1500后会导致分片。

同一次波形数据的多个波形包的会话编号需要一致。

命令编号: 0x2

方向：C->S

数据内容：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 采样位数 | 通道数 | 总包数 | 起始包数 | 样本数 | n包数据 |
| 1Byte | 1Byte | 4bytes | 4bytes | 2Bytes | 采样位数\*通道数\*样本数 |

### 2.4 上位机回应设备的波形数据

当起始包序号为0时表示第一包数据

当（起始包序号+样本数）== 总包数时 表示发送完成

命令编号: 0x82

方向：S->C

数据内容：

### 2.5 设备主动上报心跳包和注册包

为了维护设备在线状态，设备需要定时发送心跳数据给设备，同时该心跳包也作为上线注册包，上位机在没有收到注册包之前，不能主动下发命令给设备

命令编号: 0x3

方向：C->S

数据内容：

### 2.6 上位机回应心跳包

上位机回应设备心跳包中包含了上位机的系统时间，设备可以用于校时

命令编号: 0x83

方向：S->C

数据内容：

|  |
| --- |
| 系统时间戳（到秒） |
| 6Bytes(0xYY 0xMM 0xDD 0xHH 0xMM 0xSS) |

### 2.7 参数读取包

上位机主动读取设备的参数

命令编号: 0x4

方向：S->C

数据内容：无

### 2.8 设备回应参数读取包

设备参数是一个设备端和PC端通用的结构体，以1byte小端对齐。

命令编号: 0x84

方向：C->S

数据内容：

|  |
| --- |
| 参数结构体 |
| Nbytes |

/\*IP地址\*/

typedef struct{

unsigned char addr1;

unsigned char addr2;

unsigned char addr3;

unsigned char addr4;

} sIP\_ADDR;

/\*本机MAC地址\*/

typedef struct{

char addr1;

char addr2;

char addr3;

char addr4;

char addr5;

char addr6;

} sMAC\_ADDR;

/\*本机IP地址\*/

typedef struct{

sIP\_ADDR ipaddr; /\*本机IP地址\*/

sIP\_ADDR SubnetMask; /\*本机掩码地址\*/

sIP\_ADDR GateWay; /\*本机网关地址\*/

} Glocal\_IP\_ADDR;

/\*服务器IP地址，和端口号\*/

typedef struct{

sIP\_ADDR ipaddr; /\*服务器IP地址\*/

int port; /\*服务器端口\*/

} sServerADDR;

typedef struct

{

INT16U mWorkMode; //采集方式

INT16U mWetUp; // 重量上限值

INT16U mWetDown; //重量下限值

Glocal\_IP\_ADDR Local\_IP; //IP 地址

sServerADDR Server\_ip; //服务器IP

INT8U mWifiSSID[12]; //ssid 名字

INT8U mWifiPass[12]; //pass 密码

unsigned char m\_year; //年

unsigned char m\_month; //月

unsigned char m\_day; //日

unsigned char m\_hour; //时

unsigned char m\_minute; //分

unsigned char m\_seconds; //秒

INT8U checksum;

}PARA\_PLATFORM;

### 2.9 参数写入包

上位机主动写入设备的参数

设备参数是一个设备端和PC端通用的结构体，以1byte小端对齐。

命令编号: 0x5

方向：S->C

数据内容：

|  |
| --- |
| 参数结构体 |
| Nbytes |

### 2.10 设备回应参数读取包

命令编号: 0x85

方向：C->S

数据内容：

### 2.11 设备复位包

PC端请求设备复位

命令编号: 0x6

方向：S->C

数据内容：

|  |
| --- |
| 复位延迟时间（秒） |
| 1Byte |

### 2.12 回应设备请求包

设备在收到复位命令n秒后复位设备，设备需要在n秒内回应复位请求

命令编号: 0x86

方向：C->S

数据内容：无

### 2.13 枚举设备SD内的文件列表

传入一个目录名，返回该目录下的所有文件。可从根目录查询起

命令编号: 0x7

方向：S->C

数据内容：

|  |
| --- |
| 目录 |
| 变长字符串，以0结尾 |

### 2.14 返回指定目录下的文件列表

返回指定目录下的文件列表，包含目录文件

命令编号: 0x87

方向：C->S

数据内容：

|  |  |  |
| --- | --- | --- |
| 文件个数 | 文件属性列表 | 文件名字符串列表 |
| 2字节 | N\*1字节  每个文件用一个字节描述  0x1：普通文件  0x2：目录文件 | 文件1，文件2，文件3，目录4  文件名字符串之间用逗号分隔 |