

通讯协议

1. 通讯参数

波特率：9600

校验位：None

数据位：8

停止位：1

2. 建立连接

要和仪表连接时，首先要建立连接，获取波长信息，发下面消息格式的帧：

名称	数据范围（hex）	数据类型(Byte)
起始标识	0xAA	1
帧长	4	1
功能号	0x01	1
结束标识	0x55	1

设备接到消息后，响应下面消息帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	6+2n	1
功能号	0x01	1
功率计波长1（高位）	0x00~0xFF	2
功率计波长1（低位）	0x00~0xFF	
激光器波长1（高位）	0x00~0xFF	1
激光器波长1（低位）	0x00~0xFF	1
结束标识	0x55	1

3. 读取当前功率值

读取测量值发下面消息格式的帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	4	1

功能号	0x02	1
结束标识	0x55	1

设备接到消息后，响应下面消息帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	8	1
功能号	0x02	1
测量数据	70~70	4
结束标识	0x55	1

4. 切换功率计波长

设置波长发送下面消息格式的帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	5	1
功能号	0x03	1
波长编号（从0开始）	0x00~0xFF	1
结束标识	0x55	1

设备接到消息后，响应下面消息帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	4	1
功能号	0x03	1
结束标识	0x55	1

5. 读取保存值

当需要读取保存的值时，发下面消息格式的帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1

帧长	4	1
功能号	0x05	1
结束标识	0x55	1

设备接到消息后，把每一组数据组成一帧发送，发送完成后再发结束帧。

数据帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	22	1
功能号	0x05	1
序号高位	序号从0开始	1
序号地位		1
波长（高位）	0x00~0xFF	2
波长（低位）	0x00~0xFF	
功率值（float） ^a	70~-70	4
参考值（float） ^a	70~-70	4
单位（dBm/db）	0~1	1
年（2000年的后两位）	0~99	1
月	0~12	1
日	0~31	1
小时	0~24	1
分钟	0~60	1
结束标识	0x55	1

结束帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	4	1
功能号	0x05	1
结束标识	0x55	1

6. 删除指定记录

删除自定记录发下面消息格式的帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	6	1
功能号	0x06	1
指定数（高位）	0x00~0xFF	1
指定数（低位）	0x00~0xFF	1
结束标识	0x55	1

设备接到消息后，响应下面消息帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	4	1
功能号	0x06	1
结束标识	0x55	1

7. 删除所有记录

删除所有记录发下面消息格式的帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	4	1
功能号	0x07	1
结束标识	0x55	1

设备接到消息后，响应下面消息帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	4	1
功能号	0x07	1
结束标识	0x55	1

8. 校准当前波长功率

当需要校准时，发下面消息格式的帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	8	1
功能号	0x08	1
校准值 (float) ^a	0x00~0xFF	4
结束标识	0x55	1

设备接到消息后，响应下面消息帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	4	1
功能号	0x08	1
结束标识	0x55	1

9. 设置时间

设置时间发送下面的数据帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	9	1
功能号	0x09	1
年（2000年的后两位）	0~99	1
月	0~12	1
日	0~31	1
小时	0~24	1
分钟	0~60	1
结束标识	0x55	1

设备接到消息后，响应下面消息帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1

帧长	4	1
功能号	0x09	1
结束标识	0x55	1

10.数据错误处理

如果设备接收的数据不正确时，响应下面的消息帧：

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	4	1
功能号	功能号取反	1
结束标识	0xBB	1

11.响应按键命令码

所有按键设备发送消息帧和响应消息帧全由起始标识、帧长、功能号、结束标识组成。

用户只需修改“功能号”即可响应相应的按键状态（以背光按键为例）格式如下

发送一帧数据如下：AA 04 16 55

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	4	1
功能号	0x16	1
结束标识	0x55	1

设备接到消息后，响应下面消息帧：AA 04 16 55

名称	数据范围	数据类型(Byte)
起始标识	0xAA	1
帧长	4	1
功能号	0x16	1
结束标识	0x55	1

下面列出不同模式的按键状态响应“功能号”：

按键状态	功能号	数据类型(Byte)
MODE	0X0D	1
OPM_LAMDA	0X0E	1
LD_LAMDA	0X0F	1
UNITS	0X10	1
LASER	0X11	1
REF	0X13	1
ZERO	0X14	1
背光	0X16	1
SAVE	0X17	1
AUTO	0X19	1
HZ	0X1B	1
POWER_OFF	0X1E	1

（要注意响应按键的条件：即发送一次MODE命令会在功率计状态和光源状态切换）。

12.注释说明

a:在对浮点数（float）发送需要转化成整型（unsigned char），我们采用了下面的处理方式，用户可通过调用下面两个函数得到自己需要的数据。

1、把float变成4个unsigned char型保存在数组中

```
void FtoC(float Fdata,unsigned char *Cdata)
```

```
{
    unsigned char *a;
    a=(unsigned char *)& Fdata;
    Cdata[0]=a[0];
    Cdata[1]=a[1];
    Cdata[2]=a[2];
    Cdata[3]=a[3];
}
```

2、把4个unsigned char的数组转换float，返回转换后的数据

```
float CtoF(unsigned char *Cdatas)
```

```
{  
    float a;  
    unsigned char i,*px;  
    void *pf;  
    px=Cdatas;  
    pf=&a;  
    for(i=0;i<4;i++)  
    {  
        *((char *)pf+i)=*(px+i);  
    }  
    return a;  
}
```