



## LED 分析仪编程手册

版本号	修订人员	修订时间	修订内容
V1.5	NEWTON	2018-08-18	增加 r_chroma,flick_mode 等指令
V19.3	NEWTON	2019-03-03	增加 offset,r_cd_mm 等指令
V19.6	NEWTON	2019-06-06	增加 adc_samp,flick_chroma 等指令
V19.9	NEWTON	2019-09-09	增加 Red_6,system 等指令
V20.10	NEWTON	2020-10-01	增加 sdcm,flick_flow 等指令
V21.05	NEWTON	2021-05-01	增加 flick_edge 等指令
V21.06	NEWTON	2021-06-01	增加目录页
V22.03	NEWTON	2022-02-01	增加 net 网络通信说明
新版本兼容旧版本--当前 V22.03			

## ---目录---

### 目录

新版本兼容旧版本--当前 V22.03 .....	1
---目录--- .....	2
一，通信概述： .....	5
二，指令格式： .....	5
三，指令详解： .....	5
---读取仪器 IDN 信息(idn): .....	5
---查询仪器状态指令(state): .....	5
---写入模块的地址(w_id) .....	5
---读取模块的地址(r_id): .....	5
---写入模块的波特率(w_baud): .....	5
---读取模块的波特率(r_baud): .....	6
---系统复位初始化(w_system_reset): .....	6
---用户参数保存参数到 flash(save_to_flash): .....	6
---恢复出厂默认设置(default): .....	6
---配置系统的 ADC 采样模式(w_system_samp): .....	6
---读取系统的 ADC 采样模式(r_system_samp): .....	7
---写入每个通道的采样时间索引号(w_ft): .....	7
---读取每个通道的采样时间的索引号(r_ft): .....	7
---写入每个通道的增益值索引号(w_gain): .....	7
---读取每个通道的增益索引号(r_gain): .....	7
---读取每个通道的增益值(r_gainx) : .....	7
---写入可见光亮度(照度)补偿系数(w_k_Lux): .....	8
---读取可见光亮度(照度)补偿系数(r_k_lux): .....	8
---读取可见光强度(亮度/照度/流明)数据(r_lux): .....	8
---写入非可见光功率补偿系数(w_k_uw): .....	8
---读取非可见光功率补偿系数(r_k_uw): .....	8
---读取光功率辐射强度数据(r_uw_cm): .....	8
---读取流明数据(r_lm): .....	8
---读取发光强度值(坎德拉-流明)数据(r_cd_lm): .....	9
---读取发光亮度值(坎德拉/平方米)(r_cd_mm): .....	9
---读取传感器 Raw Data ADC 数据(rgbw): .....	9
---读取 RGBI(U8)颜色亮度数据(r_rgb): .....	9
---读取每个通道的 HSLI 数据(r_hsl): .....	9
---读取(CIE-1931 色坐标)xy 数据(r_xy): .....	10
---读取每个通道的 Yxy 数据(r_Yxy): .....	10
---读取(CIE-1976 色坐标)uv 数据(r_uv): .....	10
---读取 CCT(相关色温)数据(r_cct): .....	10
---读取色温黑体偏离量(r_cctd): //V1.5 版本 .....	10
---读取主波长数据(r_doWave): .....	10

---读取主波长+饱和度+亮度(r_wavesi): (V1.4 版本) .....	11
---读取 r_chroma(色度)数据: (V1.5 版本及以后) .....	11
---选择待测 LED 的颜色类型(w_target_type): .....	11
---读取待测 LED 的类型(r_target_type): .....	11
---写入色容差类型(w_sdcn_type): .....	12
---读取色容差的类型(r_sdcn_type): .....	12
---读取色容差的数据(r_sdcn_data): .....	12
---读取 flick 阈值的参考模式(r_flick_mode): .....	13
---写入 flick 的亮灭阈值(w_flick_limit): .....	13
---读取 flick 的亮灭阈值(r_flick_limit): .....	13
---启动 flick 捕获功能(w_flick_ts): .....	13
---读取 flick 周期(r_flick_ts): .....	13
---启动 LED 流水灯捕获功能(w_flick_flow): //V20.123 版本 .....	13
---启动 LED 边沿捕获功能(w_flick_edge): //V21.051 版本 .....	14
---读取 LED 多次亮灭边沿时间戳(r_flick_edge): //V21.051 版本 .....	14
---读取 flick 的照度数据(r_flick_lx): //V1.5 版本 .....	15
---读取 flick 的色度数据(r_flick_chroma): .....	15
---读取 flick 的 rgbi 数据(r_flick_rgb): .....	15
---读取 flick 的 Raw Data ADC 数据(r_flick_rgbc): .....	16
---读取被测 LED 的亮灭状态(r_led_ch1) .....	16
---写入 LED 发射光源的亮灭状态(w_led_disp) .....	16
---读取 LED 发射光源的亮灭状态(r_led_disp) .....	16
---使能 Offset 组别(w_offset_en) .....	16
---写入 Offset 组别的补偿值 dx(w_offset_dx) .....	16
---写入 Offset 组别的补偿值 dy(w_offset_dy) .....	17
---写入 Offset 组别的亮度值系数 kl(w_offset_kl) .....	17
---保存 offset 所有数据到 flash (w_offset_save) .....	17
---清零 offset 所有数据(w_offset_clear) .....	17
---读取 offset 使能组别(r_offset_en) .....	17
---读取 offset 补偿值 dx (r_offset_dx) .....	17
---读取 offset 补偿值 dy(r_offset_dy) .....	18
---读取 offset 补偿值 kl(r_offset_kl) .....	18
---读取 DI 单通道输入状态(r_inbit) .....	18
---读取 DI1-8 通道输入状态 U8(r_in_u8) .....	18
---写入 DO 多个通道输出状态(w_outbit) .....	18
---写入 net 的工作模式(w_net_mode) .....	19
---读取 net 的工作模式(r_net_mode) .....	19
---写入 net 的仪器的源 IP 地址(w_net_sip) .....	19
---读取 net 的仪器的源 IP 地址(r_net_sip) .....	19
---写入 net 的上位机的目标 IP 地址(w_net_dip) .....	19
---读取 net 的上位机的目标 IP 地址(r_net_dip) .....	19
---写入 net 的网关 Gateway IP 地址(w_net_gip) .....	19

---读取 net 的网关 Gateway IP 地址(r_net_gip) .....	19
---写入 net 的子网掩码 Subnet Mask IP 地址(w_net_mip) .....	20
---读取 net 的子网掩码 Subnet Mask IP 地址(r_net_mip) .....	20
---写入 net 的 LED 仪器的源端口号(w_net_sport) .....	20
---读取 nnet 的 LED 仪器的源端口号(r_net_sport) .....	20
---写入 net 的上位机服务器的目标端口号(w_net_dport) .....	20
---读取 net 的上位机服务器的目标端口号(r_net_dport) .....	20
---读取 net 的 MAC 地址(r_net_mac) .....	20
---读取 net 的所有配置参数(r_net_all) .....	20
---试运行 net 网口通信连接(w_net_run) .....	20
四：指令表汇总： .....	21
五：软件开发流程如下 .....	24
---单色常亮 LED 数据采集流程 .....	24
---双色常亮 LED 数据采集流程 .....	24
---单色闪烁 Flick-LED 数据采集流程 .....	25
六：测试用例-指令读写实例 .....	27
---读取 1CH 的 RED-LED 的 RGBI/HSLI(ID=1): .....	27
---读取 1CH 的黄绿灯的主波长(ID=1): .....	27
---读取 1CH 的白光灯的 xy 色坐标(ID=1): .....	27
---读取 1CH 的 RED-LED-flick 闪烁频率(ID=1): .....	27
七：编程注意事项-必看！ .....	28
1, 配置参数保存 save_to_flash .....	28
2, 增益 gain 和采样时间 ft .....	28
3, 采样模式 w_system_samp: .....	28
4, LED 光源类型选择 w_target_type .....	28
5, 频率流水灯捕获指令 flick/flow: .....	28
6, r_rgb 与 r_chroma 颜色识别指令: .....	28
7, Net 通信说明: .....	28
八：手册声明： .....	29

### 一，通信概述：

- 1,USB,RS485 两种接口,都遵循 RS232 硬件协议，8b-byte，1b-stop，none 校验；
- 2,USB,RS485 两种接口通信协议一样，字符串通信；
- 3,网口 net 通信，底层是 TCP/IP 协议，但顶层通信指令和串口指令完全一样；

### 二，指令格式：

0.假设当前模块的地址为“001”，且系统中只有一个该模块；

- 1.主机发送指令格式：“:001\*\*\*\*\*\n”---( “:”+“ID”+“功能指令”+“\n” );
- 2.模块返回指令格式：“:001#####\r\n”---( “:”+“ID”+“返回信息”+“\n” );
- 3,其中“:”为起始字符，“\r\n”为结束字符(至少要有有一个结束符'\n')，ID 为 3 位整数(%03d)，字符串内无空格；
- 4,所有指令全为英文字符串,无 CRC 校验;
- 5,当发送错误命令时，模块统一返回":001ERR\_CMD\r\n"或者无任何字符返回
- 6,ID="000"是广播地址，所有模块都会应答;
- 7,RS485 通信时，发送和读取需预留总线转换时间(>3ms);
- 8,数据读写指令全部支持任意单通道和连续多通道操作，通道号必须递增格式；

### 三，指令详解：

#### ---读取仪器 IDN 信息(idn):

询问模块的型号信息；

发送指令格式：“:001idn\n”

返回指令格式：“:001#####\r\n”

指令解释：返回的指令以实际字符为准，

(具体根据模块反馈信息而定,该信息内容没有特定的格式规律至少有“HanOpticSens”关键字)

#### ---查询仪器状态指令(state):

询问模块的工作状态是 idle 还是 busy;

发送指令格式：“:001state\n”

返回指令格式：“:001idle\r\n”或者为“:001busy\r\n”;

指令解释：当模块 busy 时，模块不会执行新发来的指令；

#### ---写入模块的地址(w\_id)

发送指令格式：“:000w\_id=002\n”或“:001w\_id=002\n”(该参数立即生效，自动保存在 flash)

返回指令格式：“:002w\_id=002\n”;

指令解释：

写入 ID 成功后，返回指令立即采用新的 ID 返回信息；

该指令会擦写 flash，寿命小于 10 万次，不要在主程序内频繁操作；

#### ---读取模块的地址(r\_id):

发送指令格式：“:000r\_id\n”或“:001r\_id\n”

返回指令格式：“:001r\_id=001\n”;

指令解释：模块返回自己当前的 ID="001”;

#### ---写入模块的波特率(w\_baud):

发送指令格式：“:001w\_baud1=6\n”或 “:001w\_baud2=6\n”(该参数立即生效，自动保存在 flash)

返回指令格式：“:001w\_baud1=6\n”或 “:001w\_baud2=6\n”

指令解释:

baud1 代表 USB-RS232 接口, baud2 代表 RS485 接口;

模块先按照原来的 baud 返回指令, 再配置自己的 baud;

“=”后面的值是波特率编号(0-9),默认值是 6;

baud[10]={2400,4800,9600,19200,38400,57600,115200,230400,460800,921600};

如果只有一个通信接口, 只能用 baud1;

该指令会擦写 flash, 寿命小于 10 万次, 不要在主程序内频繁操作;

**---读取模块的波特率(r\_baud):**

发送指令格式: “:001r\_baud1\n”或 “:001r\_baud2\n”

返回指令格式: “:001r\_baud1=6\n”或 “:001r\_baud2=6\n”

指令解释: baud1 代表 USB-RS232, baud2 代表 RS485(RS232), 返回接口波特率编号(0-9);

**---系统复位初始化(w\_system\_reset):**

发送指令格式: “:001w\_system\_reset\n”

返回指令格式: “:001w\_system\_reset\n”

指令解释:

让产品在上电状态下从新复位, 并不是恢复出厂设置;

在外置 MIC 探头系列产品中可以在探头热插拔后, 从新恢复工作状态;

**---用户参数保存参数到 flash(save\_to\_flash):**

发送指令格式: “:001save\_to\_flash\n”(该指令立即生效-不建议频繁执行)

返回指令格式: “:001save\_to\_flash\n”;

指令解释:

把 gain,ft,target\_type,system\_samp,net 等用户可以修改的参数保存到 flash, 掉电不丢失;

该指令会擦写 flash, 寿命小于 10 万次, 不要在主程序内频繁操作;

**---恢复出厂默认设置(default):**

发送指令格式: “:001default\n” (该指令立即生效-不建议频繁执行)

返回指令格式: “:001default\n”;

指令解释:

把 id,baud,gain,ft,target\_type,system\_samp,net 等参数全部恢复出厂默认, 并保存到 flash, 掉电不丢失; 该指令会擦写 flash, 寿命小于 10 万次, 不要在主程序内频繁操作;

**---配置系统的 ADC 采样模式(w\_system\_samp):**

发送指令格式: “:001w\_system\_samp=1\n”

返回指令格式: “:001w\_system\_samp=1\n”

指令解释: **(V19.6-该参数立即生效, 暂存于 RAM 中, 也可以保存, 掉电不丢失)**

“=”后面值是 adc\_samp 工作模式, 0:连续采样模式 continuous, 1:单次采样模式 single sample;

**0:** 所有通道都处于同步连续采样模式, 数据更新周期等于采样时间(FT);

但不管 FT 是多大, 当模块接收到串口指令后, 会立即返回数据, 但返回的是上一个采样周期的数据, 所以读取时要注意: 必须让 LED 点亮时间大于 2 倍的 FT, 再去读取数据, 才能保证读取到的是 LED 点亮稳定后的数据; 这个指令的优点在于串口返回数据不等待, 可节约上位机等待时间; 对于 LED 常亮型或者对测试时间要求不高的话, 建议采用该模式;

**1:** 发送一次读取指令, 传感器立即从新采样一次数据, 模块串口需等待 FT 时间后才能返回采集到的数据;虽然串口等待, 但可以测量到短暂的脉冲光源: 点亮 LED 后, 立即发送一次读取, 只要 LED 点亮时间比 FT 时间长, 那么读取到的就是 LED 点亮的稳定数据;

---读取系统的 ADC 采样模式(r\_system\_samp):

发送指令格式: ":001r\_system\_samp\n"

返回指令格式: ":001r\_system\_samp=0\n"

指令解释:

"="后面的值是 adc\_samp 的工作模式, 0:连续采样模式 continuous(默认), 1:单次采样模式 single sample;

---写入每个通道的采样时间索引号(w\_ft):

发送指令格式: ":001w\_ft01-20=1\n"(该参数立即生效, 暂存在于 RAM 中)

返回指令格式: ":001w\_ft01-20=1\n"

指令解释:

**相关读写指令全部支持任意单通道和连续多通道操作, 格式必须严格执行:**

**"01-20"代表从 01(%02d)通道到 20(%02d)通道, 必须是从小到大, 或者是相等(通道最大值为 20);**

"="后面的值是 ft 编号(0-5-11), 编号越大, ft 越大, 采样速度越慢, 读取到的 rgbw 原始 ADC 数据越大;

在频闪测试时, 根据实际频率大小选择合适的 ft; (某些产品的 ft 索引号最多是 5);

---读取每个通道的采样时间的索引号(r\_ft):

发送指令格式: ":001r\_ft03-06\n"

返回指令格式: ":001r\_ft=1,1,1,1,\n"

指令解释:

"="后面的值依次是 03-06 通道的 ft 索引号;

---读取每个通道的真实采样时间(r\_ftms):

发送指令格式: ":001r\_ftms01-02\n"

返回指令格式: ":001r\_ftms=20,123,\n"

指令解释:

"="后面的值依次是 01-02 通道 ft 真实 ms 时间, 该值与写入的 ft 索引号对应, 实际数据以模块返回的数据为准, 对频闪测试有参考意义;

---写入每个通道的增益值索引号(w\_gain):

发送指令格式: ":001w\_gain01-20=1\n"(该参数立即生效, 暂存在于 RAM 中)

返回指令格式: ":001w\_gain01-20=1\n"

指令解释:

"="后面的值是 gain 编号(0-4-11), 编号越大, gain 越大, 读取到的 rgbw 原始 ADC 数据越大; 修改该参数, 是为了让传感器工作在最佳量程范围, 或者提高灵敏度; (某些产品的 gain 索引号最多是 4);

---读取每个通道的增益索引号(r\_gain):

发送指令格式: ":001r\_gain03-06\n"

返回指令格式: ":001r\_gain=1,1,1,1,\n"

指令解释:

"="后面的值依次是 03-06 通道的增益编号;

---读取每个通道的增益值(r\_gainx) :

发送指令格式: ":001r\_gainx01-02\n"

返回指令格式: ":001r\_gainx=5,1,\n"

指令解释:

“=”后面的值依次是 01-02 通道的增益值,该值与写入的 gain 索引号对应,实际数据以模块返回的数据为准;

//////////光强数据-//////////

---写入可见光亮度(照度)补偿系数(w\_k\_lux):

发送指令格式:“:001w\_k\_lux01-08=1.001\n”

返回指令格式:“:001w\_k\_lux=1.001\n”

指令解释:

“=”后面的值依次是 01-02 通道的补偿系数,精确到 0.001,对于亮度一致性测试有用;

---读取可见光亮度(照度)补偿系数(r\_k\_lux):

发送指令格式:“:001r\_k\_lux01-02\n”

返回指令格式:“:001r\_k\_lux=1.001,1.001\n”

指令解释:

“=”后面的值依次是 01-02 通道的补偿系数,精确到 0.001;

---读取可见光强度(亮度/照度/流明)数据(r\_lux):

发送指令格式:“:001r\_lux01-02\n”

返回指令格式:“:001r\_lux=123.12,234.12,\n”

指令解释:

“=”后面的值依次是 01-02 通道的照度值,数据格式是浮点字符串%f,

在测量照度的产品中,该值单位是 lx;在光纤系列产品中,该值代表正比于 LED 发光强度的一个参考值,需要二次标定才能得到真实的照度/亮度/流明值;

---写入非可见光功率补偿系数(w\_k\_uw):

发送指令格式:“:001w\_k\_uw01-08=1.001\n”

返回指令格式:“:001w\_k\_uw=1.001\n”

指令解释:

“=”后面的值依次是 01-08 通道的补偿系数,精确到 0.001,主要用于非可见光 IR/UV 强度测量;

---读取非可见光功率补偿系数(r\_k\_uw):

发送指令格式:“:001r\_k\_uw01-02\n”

返回指令格式:“:001r\_k\_uw=1.001,1.001\n”

指令解释:

“=”后面的值依次是 01-08 通道的补偿系数,精确到 0.001;

---读取光功率辐射强度数据(r\_uw\_cm):

发送指令格式:“:001r\_uw\_cm01-02\n”

返回指令格式:“:001r\_uw\_cm=123.1,234.2,\n”

指令解释:

“=”后面值依次是 01-02 通道的光功率精确到 0.1%FS,该数据只用于非可见光功率判断;

---读取流明数据(r\_lm):

发送指令格式:“:001r\_lm01-02-1.0\n”

返回指令格式:“:001r\_lm=1.123,0.123,\n”

指令解释: //在积分球探头内测量, 数据比较准确

"="后面的值依次是 01-02 通道的流明 LM"(%,0.3f),(%,0.3f),";

"1.0(%)是补偿系数(相乘关系), 非光纤流明测量探头, 在出厂前已校准, 默认是 1.0;

---读取发光强度值(坎德拉-流明)数据(r\_cd\_lm):

发送指令格式: ":001r\_cd\_lm01-01-00-20.5\n"//只适用于标准朗伯体光源

返回指令格式: ":001r\_cd\_lm=1.123,2.123,\n"

指令解释:

"00(%,0.2d)"是探头相对于 LED 法线的平面夹角 $\theta$ (0-89°C),

"20.5(%,0.1f)"是探头距离 LED 的直线距离 d(单位:mm);

"="后面的值依次是 01-01 通道的探测角度的坎德拉,以及 LED 的最大流明

("%,0.3f,%,0.3f,"cd,lm);

为了保证测量数据准确性, 尽量保证 LED 向空间无遮拦,无反射,无聚散透镜,自由发散,

尽量在 LED 法线方向上测量, 即夹角等于 0; 探头尽量远离光源,  $d > 10$  倍光源直径;

---读取发光亮度值(坎德拉/平方米)(r\_cd\_mm):

发送指令格式: ":001r\_cd\_mm01-02\n"

返回指令格式: ":001r\_cd\_mm=123,125,\n"

指令解释: // 只适用于 V19.3 之后版本

"="后面的值依次是 01-02 通道的亮度值, 单位是"坎德拉/平方米"(cd/m<sup>2</sup>),

为了保证测量数据绝对值准确性, 外置探头尽量靠近面光源, 如果光源面积小于探头感光面积, 读取到的亮度值小于理论值; 采用光纤导光测量时, 该数据等于 r\_lux 指令, 该指令适合非光纤类产品;

//////////RGB 颜色数据-适合 RGB 系列产品//////////

---读取传感器 Raw Data ADC 数据(rgbw):

发送指令格式: ":001rgbw01-02\n"

返回指令格式: ":001rgbw=123,234,345,678,11123,11234,11345,22678,\n"

指令解释:

"="后面的值依次是 01-02 通道的"r(%,d),g(%,d),b(%,d),w(%,d)", 属于传感器原始 ADC 数据,rgbw 的比例代表颜色, 绝对值代表相对强度; 实际编程时, 不建议用该数据作为判断标准, 但该数据可以反应传感器工作状态是否合理, 数据太小,接近传感器灵敏度下限, 导致不准确, 过大可能会造成数据饱和, 数据失真;

距离数据饱和比例处在 1%-60%是最佳工作区间; 修改 gain 或 ft 参数, 会更改 Raw Data;

---读取 RGBI(U8)颜色亮度数据(r\_rgb):

发送指令格式: ":001r\_rgb01-01\n"

返回指令格式: ":001r\_rgb=255,244,105,50.00\n"

"="后面的值依次是 01-01 通道的"r(%,d),g(%,d),b(%,d),i(%,0.2f)", 该 RGB 范围为 0-255;

I 是相对于最大测量强度的百分比(0.00-100.00), I 可以作为相对亮度的数据(对比测试比较实用),

但同一光强下, 修改 Gain 参数, 会改变 I 的值;I=100.00%时, 说明传感器已完全饱和;

该 RGBI 仅是颜色亮度的相对分量值, 没有标准可以参考, 只能一致性相对比较;

该 RGB 与相机的 RGB 和显示器图片 RGB 数据原理还是存在一定的差异, 不能完全对等;

---读取每个通道的 HSLI 数据(r\_hsl):

发送指令格式: “:001r\_hsli01-02\n”

返回指令格式: “:001r\_hsli=300,80,40,10.01,300,80,40,10.01,\n”

指令解释:

“=”后面的值依次是 01-02 通道的“H(%0.0f),S(%0.0f),L(%0.0f),I(%0.2f),”;

H(0-360), S(0%-100%), L(0%-100%), I(0.00%-100.00%) ,

I 是相对于最大测量强度的百分比, I=100%时, 说明传感器已完全饱和;

该 HSLI 仅是颜色亮度的相对分量值, 没有标准可以参考, 只能一致性相对比较, 其中 L 与 HSV 种的 V 相同;

该 HSL 与相机的 HSL 和显示器图片 HSL 数据原理还是存在一定的差异, 不能完全对等;

//////////CIE-Chroma 色度数据//XYZ/MIC/True 系列产品//////////

---读取(CIE-1931 色坐标)xy 数据(r\_xy):

发送指令格式: “:001r\_xy01-02\n”

返回指令格式: “:001r\_xy=0.3333,0.4333,0.3666,0.3111,\n”

指令解释:

“=”后面的值依次是 01-02 通道的“x(0.4F%),y(0.4F%),”, 该 xy 代表 CIE-1931 色度图坐标;

---读取每个通道的 Yxy 数据(r\_Yxy):

发送指令格式: “:001r\_Yxy01-02\n”

返回指令格式: “:001r\_Yxy=323.5,0.2345,0.3145,678.5,0.5234,0.1434,\n”

指令解释: “01-02 代表从 01(%02d)通道到 02(%02d)通道, 必须是从小到大, 或者是相等(通道最大值为 8(16)); “=”后面的值依次是 01-02 通道的“Yxy(f%),(f%),(f%),”, Y 等于照度值 Lux, xy 是色坐标;

---读取(CIE-1976 色坐标)uv 数据(r\_uv):

发送指令格式: “:001r\_uv01-02\n”

返回指令格式: “:001r\_uv=0.3333,0.4333,0.6666,0.1111,\n”

指令解释:

“=”后面的值依次是 01-02 通道的“u(0.4F%),v(0.4F%),”, 该 uv 代表 CIE-1976 色度图坐标;

---读取 CCT(相关色温)数据(r\_cct):

发送指令格式: “:001r\_cct01-02\n”

返回指令格式: “:001r\_cct=5438,6457,\n”

指令解释:

“=”后面的值依次是 01-02 通道的“CCT(d%),(d%),”, 该 CCT 数据根据 CIE-1931(xy)色度图坐标计算得出;

---读取色温黑体偏离量(r\_cctd): //V1.5 版本

发送指令格式: “:001r\_cctd01-01\n”

返回指令格式: “:001r\_cctd=5438,0.00601,\n”

指令解释:

“=”后面的值依次是 01-01 通道的“CCT(0%0.0f),duv(%0.6f),”, 该数据根据 CIE-1960(uv)色度图坐标计算得出;

---读取主波长数据(r\_doWave):

发送指令格式: “:001r\_dowave01-02\n”

返回指令格式: “:001r\_dowave=438.5,617.5,\n”

指令解释:

“=”后面的值依次是 01-02 通道的“(%,0.1f),(%,0.1f),”;

---读取主波长+饱和度+亮度(r\_wavesi): (V1.4 版本)

发送指令格式: “:001r\_wavesi01-01\n”

返回指令格式: “:001r\_wavesi=555.5,99.9,123.4,\n”

指令解释:

“=”后面的值依次是 01-01 通道的“(%,0.1f),(%,0.1f),”, 555.5 代表主波长, 单位 nm, 99.9%代表饱和度, 范围 0%-100%; 123.4 是照度, 单位 lx, 与指令 r\_lux 读取数据相同;

---读取 r\_chroma(色度)数据: (V1.5 版本及以后)

发送指令格式: “:001r\_chroma01-01\n”

返回指令格式: “:001r\_chroma=1000.0,0.3333,0.4444,555.5,85.2,6500,0.00123,\n”

指令解释:

“=”后面的值依次是 01-01 通

道“lux(%,0.1f),x(%,0.4f),y(%,0.4f),dowave(%,0.1f),duty(%,0.1f),cct(%,0.0f),fd(%,0.5f),”

该指令可以高效获取 LED 的色度数据;

lux=1000.0, 与 r\_lux 指令相同;

x=0.3333,y=0.4444, 与 r\_xy 相同, 是 CIE1931 的 xy 坐标;

dowave=555.5nm, duty=85.2%, 与 r\_wavesi 功能相似;

cct=6500K, 与 r\_cct 指令类似;

fd=0.00123, 该浮点数是预留参数, 不同产品该值不一样, 默认是 CCT 中的 duv 参数, 但启用 SDCM 测量数据时, fd 是 SDCM 数据(支持 V20.101 及以后);

---选择待测 LED 的颜色类型(w\_target\_type):

发送指令格式: “:001w\_target\_type01-02=0\n”(立即生效, 暂存于 RAM 中, 只适用 XYZ 及以上产品)

返回指令格式: “:001w\_target\_type01-02=0\n”

指令解释:

“=”后面的值是类型编号(0-7), 默认值是 0, 任何模式都可以读取出来色度值, 但为了获取更准确的色度参数, 需要选择合适的模式; 只有 XYZ 及以上系列产品支持该指令;

0: 白平衡模式, 任何光源都可以采用该模式, 但不一定是最优配置, 比较适合白光;

3: 白色照明 LED, 色温 CCT 区间(2000K-20000K)之间最准确, 比如汽车 LED 白光照明灯;

4: 单芯灯珠发“橙色(600nm 左右)”或“黄绿(570nm)”或“青色(500nm 左右),”

以及任意其中两芯或三芯混合而成的灯光, 比如路由器等消费类电子产品指示灯;

5: RGB 单色灯 Red(630nm 左右), Green(525nm 左右), Blue(470nm 左右),

以及任意其中两芯或三芯混合而成的灯光; 比如 RGB 氛围灯;

6: Red 单色灯, 主波长在(600-690nm)之间的红灯;

1, 2, 7 预留;

**备注:** 在一些特殊光源测量时, 会占用其中某些模式, 以配套编程手册为准;

该指令目的只在于修正色度数据的准确度;

---读取待测 LED 的类型(r\_target\_type):

发送指令格式: “:001r\_target\_type01-02\n”

返回指令格式: “:001r\_target\_type=0,0\n”

指令解释:

“=”后面的值是 LED 类型配置的编号;

---写入色容差类型(w\_sdcn\_type):

发送指令格式: “:001w\_sdcn\_type01-02=0\n”

返回指令格式: “:001w\_sdcn\_type01-02=0\n”

指令解释: (立即生效, 暂存于 RAM 中, 只适用 XYZ 及以上版本 V20.101)

“=”后面的值是类型编号(0-99),默认值是 0,disable,

类型编号	功能描述	备注
0	Disabale, 不启用 SDCM 测量功能	
1	预留, 暂时不用	后续可能会添加成全自动匹配模式
2	AUTO-SDCM_ErpF	
3	AUTO-SDCM_ANSI	
4	AUTO-SDCM_customized	可根据客户的要求定制
5,6,7,8,9	预留	
10-19	SDCM_ErpF (0,1,2,3,4,5,6,7,8,9,)	
20-29	SDCM_ANSI (0,1,2,3,4,5,6,7,8,9,)	
30-39	SDCM_customized (0,1,2,3,4,5,6,7,8,9,)	

---读取色容差的类型(r\_sdcn\_type):

发送指令格式: “:001r\_sdcn\_type01-02\n”

返回指令格式: “:001r\_sdcn\_type=0,0\n”

指令解释:

“=”后面的值是 sdcn 类型配置的编号;

---读取色容差的数据(r\_sdcn\_data):

发送指令格式: “:001r\_sdcn\_data01-02\n”

返回指令格式: “:001r\_sdcn\_data=3.2,2.3\n”

指令解释:

“=”后面的值是 01-02 通道的 sdcn 数据, 精确度 0.1f%;

---读取色容差+光照度的数据(r\_sdcn\_lux):

发送指令格式: “:001r\_sdcn\_lux01-01\n”

返回指令格式: “:001r\_sdcn\_lux=100.1,3.2,21\n”

指令解释:

“=”后面是 01 通道的数据, 100.1 是 lux 照度数据, 3.2 是 SDCM 值, 21 是 SDCM 的光源对标类型 ANSI 3000K;

//////////flick 指令通用, 但不同型号仪器的最小采样速度不同//////////

---写入 flick 阈值的参考模式(w\_flick\_mode):

发送指令格式: “:001w\_flick\_mode01-08=0\n”(该参数立即生效, 暂存于 RAM 中)

返回指令格式: “:001w\_flick\_mode01-08=0\n”;

指令解释:

"="后面的模式索引范围是 0-9(%d),默认值是 0; 0:LUX / 1:R / 2:G / 3:B / 4:W;

该模式参数,是选择采用哪个数据与亮灭阈值(flick\_limit)进行对比, lux 数据参考 r\_lux,  
R-G-B-W 是传感器原始 ADC 数据,适用于多色 LED 交替闪烁,以及多色交替呼吸灯的场合;

---读取 flick 阈值的参考模式(r\_flick\_mode):

发送指令格式: ":001r\_flick\_mode01-02\n"

返回指令格式: ":001r\_flick\_mode=0,0\n";

指令解释:

"="后面的模式索引范围是 0-9(%d); 0:LUX / 1:R / 2:G / 3:B / 4:W;

---写入 flick 的亮灭阈值(w\_flick\_limit):

发送指令格式: ":001w\_flick\_limit01-08=10\n"(该参数立即生效,暂存于 RAM 中)

返回指令格式: ":001w\_flick\_limit01-08=10\n";

指令解释: "="后面的阈值范围是 0-1000000(%d),必须大于 0,默认值是 20;

当采集到的选定数据(见 flick\_mode)小于该值时,认为是灯灭,大于该阈值,灯点亮,该值是判断频闪测试的亮灭阈值,可独立配置每个通道;

---读取 flick 的亮灭阈值(r\_flick\_limit):

发送指令格式: ":001r\_flick\_limit01-02\n"

返回指令格式: ":001r\_flick\_limit=20,20,\n";

指令解释: "="后面是返回的阈值(0-1000000:%d);

---启动 flick 捕获功能(w\_flick\_ts):

发送指令格式: ":001w\_flick\_ts01-02=09\n"

返回指令格式: ":001w\_flick\_ts01-02=09\n";

指令解释: "09"的取值范围是 01-50 秒,设置采样时间长度,单位是秒;

须提前设置好相应通道的硬件参数(ft&gain&w\_flick\_limit),该指令下达后,模块立即执行频闪测试,测试时间持续 9s(设定值);测试期间,模块不响应其他指令,但可以发送 state 指令"查询模块状态,待模块返回"idle",才可发送新的读写指令;模块支持所有通道同时进行频闪测试;测量结果会一直保持在 RAM 中,直到下次指令下达从新转换更新;

---读取 flick 周期(r\_flick\_ts):

发送指令格式: ":001r\_flick\_ts01-02\n"(数据一直保持,直到下次测试更新)

指令解释: 读取 01-02 通道的频闪测试结果;

返回指令格式: ":001r\_flick\_ts=1.00,990,1000,500,5, 2.00,490,501,100,8\n";

指令解释:

"="后面值分别代表(频率 1.00Hz,亮-亮间隔 990ms,灭到灭间隔 1000ms,持续点亮时间 500ms,亮脉冲次数 5)

Fre1(Hz)"%0.2f",Tup-up1(ms)%d ", Tdw-dw1(ms)"%d ",Tduty1(ms)"%d",Cnt1\_Puls"%d",

Fre2(Hz)"%0.2f",Tup-up2(ms)%d ", Tdw-dw2(ms)"%d ",Tduty2(ms)"%d",Cnt1\_Puls"%d",

//////////flow,edge-指令通用,但不同型号仪器的最小采样速度不同//////////

---启动 LED 流水灯捕获功能(w\_flick\_flow): //V20.123 版本

发送指令格式: ":001w\_flick\_flow01-16=03\n" //只适合顺序点亮 LED 流水检测,例如汽车流水转向灯

指令解释: 同时启动 1-16chI 流水捕获功能,捕获时间是 3 秒;

返回指令格式: ":001w\_flick\_flow01-16=03\n";

指令解释: "03"的取值范围是 01-55 秒,设置捕获时间长度,单位是秒;

须提前设置好相应通道的硬件参数(ft&gain&w\_flick\_limit), 该指令下达后, 模块立即执行 flow 测试, 测试时间持续 3s(设定值); 测试期间, 模块不响应其他指令, 但可以发送 state 指令"查询模块状态, 待模块返回"idle",才可发送新的读写指令; 模块支持所有通道同时进行 flow 测试; 测量结果会一直保持在 RAM 中, 直到下次指令下达从新转换更新; 该指令收到后, 会先立即返回":001w\_flick\_flow01-16=03\n",再进行 flow 测量;

在多个模块 RS485 并行测量多个流水灯时, 可以发送把地址改为 0, 例如发送":000w\_flick\_flow01-16\n"指令, 同时启动多个仪器捕获, 由于是 485 并联, 仪器会同时响应广播地址 0, 只有 ID=1 的仪器会正常返回值":000w\_flick\_flow01-16\n", 其他 ID 仪器不回复任何指令, 以免造成总线冲突;

**---读取 LED 流水灯亮熄灭时间戳(r\_flick\_flow): //V20.123 版本**

发送指令格式: ":001r\_flick\_flow01-02\n"(数据一直保持, 直到下次测试更新)

指令解释: 读取 01-02 通道的 flow 测试结果;

返回指令格式: ":001r\_flick\_flow=1000,1500,500,1050,1450,400,\r\n";

指令解释:

"="后面值是 ms, 全部整数, 返回数据个数 CNT=CHL\*3;

1000: 代表 1CH 第一次被点亮的时间,

1500: 1CH 点亮后第一次熄灭的时间,

500: 是预留的参数, 暂定是前后时间差 1500-1000=500;

1050: 2CH 第一次被点亮的时间,

1450: 2CH 点亮后第一次熄灭的时间,

400: 是预留的参数, 暂定是前后时间差 1450-1050=400;

从数据上来看, 第 2CH 的点亮时间比第 1CH 晚了 1050-1000=50ms, 但第 2CH 的熄灭亮时间比第 1CH 早了 50ms(1450-1500 = -50ms);说明流水灯是从 1 到 2 顺序点亮, 然后再反序熄灭; 测量的时间参考 0 点, 是发送 w\_flick\_flow 启动指令后, 仪器返回 w\_flick\_flow 后, 计为 0ms;

**---启动 LED 边沿捕获功能(w\_flick\_edge): //V21.051 版本**

发送指令格式: ":001w\_flick\_edge01-16=03\n"

指令解释: 同时启动 1-16ch 边沿捕获功能, 捕获时间是 3 秒;

适用多次流水或多次闪烁的场合,比如汽车迎宾灯,流水转向灯,是频闪中最通用的一条指令;

返回指令格式: ":001w\_flick\_edge01-16=03\n";

指令解释: "=03"的取值范围是 01-55 秒, 设置捕获时间长度, 单位是秒;

须提前设置好相应通道的硬件参数(ft&gain&w\_flick\_limit), 该指令下达后, 模块立即执行 edge 测试, 测试时间持续 3s(设定值); 测试期间, 模块不响应其他指令, 但可以发送 state 指令"查询模块状态(查询频率过快, 会降低底层采集速度), 待模块返回"idle",才可发送新的读写指令; 模块支持所有通道同时进行 edge 测试; 测量结果会一直保持在 RAM 中, 直到下次指令下达从新转换更新; 该指令收到后, 会先立即返回":001w\_flick\_edge01-16=03\n",再进行 flow 测量;

在多个模块 RS485 并行测量多个流水灯时, 可以发送把地址改为 0, 例如发

送":000w\_flick\_edge01-16\n"指令, 同时启动多个仪器捕获, 由于是 485 并联, 仪器会同时响应广播地址 0, 只有 ID=1 的仪器会正常返回值":000w\_flick\_edge01-16\n", 其他 ID 仪器不回复任何指令, 以免造成总线冲突;

**---读取 LED 多次亮灭边沿时间戳(r\_flick\_edge): //V21.051 版本**

发送指令格式: ":001r\_flick\_edge01-02=2\n"(数据一直保持, 直到下次测试更新)

指令解释:

读取 01-02 通道的 edge 测试结果,"=2"是每个通道读取 EDGE 次(2 次)点亮熄灭的时间戳,最多是 10 次;

返回数据个数 CNT 等于通道数 CHL 乘以边沿次数 EDGE,再乘以 2(上升沿和下降沿),

$CNT=CHL*EDGE*2$ ;

测量的时间参考 0 点,是发送 w\_flick\_flow 启动指令后,仪器返回 w\_flick\_flow 后,计为 0ms;

返回指令格式: ":001r\_flick\_edge=1000,1100,1200,1300,2000,2100,2200,2300,\r\n";

指令解释:

"="后面值是 ms,全部整数,前面 4 个数据是 1CHL 时间戳,后面 4 个数据是 2CHL 时间戳,

1000: 1CHL 第一次点亮的时间,

1100: 1CHL 点亮后第一次熄灭的时间,

1200: 1CHL 第二次点亮的时间,

1300: 1CHL 第二点亮后再次熄灭的时间,

2000: 2CHL 第一次点亮的时间,

2100: 2CHL 点亮后第一次熄灭的时间,

2200: 2CHL 第二次点亮的时间,

2300: 2CHL 第二点亮后再次熄灭的时间,

//////////下面是读取 flick 期间捕获的光度数据//////////

---读取 flick 的照度数据(r\_flick\_lx): //V1.5 版本

发送指令格式: ":001r\_flick\_lx01-02\r\n"(数据一直保持,直到下次测试更新)

返回指令格式: ":001r\_flick\_lx=123,234,\r\n"

指令解释:

"="后面的值依次是 01-02 通道的"lx1(%0.0f),lx2(%0.0f),";该数据是 LED 点亮期间 lux 的最大值,该值的大小可以反映 LED 的光照度(光亮度);

---读取 flick 的色度数据(r\_flick\_chroma):

发送指令格式: ":001r\_flick\_chroma01-01\r\n" (V19.6 版-数据一直保持,直到下次测试更新)

返回指令格式: ":001r\_flick\_chroma=1000.0,0.3333,0.4444,555.5,85.2,6500,0.00123,\r\n"

指令解释:

"="后面的值依次是 01-01 通

道"lux(%0.1f),x(%0.4f),y(%0.4f),dowave(%0.1f),duty(%0.1f),cct(%0.0f),fd(%0.5f),"

该指令可以高效获取 LED 的色度数据:

lux=1000.0,与 r\_lux 指令相同;

x=0.3333,y=0.4444,与 r\_xy 相同,是 CIE1931 的 xy 坐标;

dowave=555.5nm,duty=85.2%,与 r\_wavesi 功能相似;

cct=6500K,与 r\_cct 指令类似;

fd=0.00123,该浮点数是预留参数,不同产品该值不一样,默认是 CCT 中的 duv 参数;

---读取 flick 的 rgbi 数据(r\_flick\_rgb):

发送指令格式: ":001r\_flick\_rgb01-01\r\n" (V19.6 版-数据一直保持,直到下次测试更新)

返回指令格式: ":001r\_flick\_rgb=255,244,105,50.00\r\n"

"="后面的值依次是 01-01 通道的"r(%d),g(%d),b(%d),i(%0.2f),";该 RGB 范围为 0-255;

I 是相对于最大测量强度的百分比(0.00-100.00),I 可以作为相对亮度的数据(对比测试比较实用),

但同一光强下，修改 Gain 参数，会改变 I 的值;I=100.00%时，说明传感器已完全曝光；

该指令与 r\_rgb 指令读取效果一样；

---读取 flick 的 Raw Data ADC 数据(r\_flick\_rgbc):

发送指令格式: “:001r\_flick\_rgbc01-01\n”(V19.6 版-数据一直保持，直到下次测试更新)

返回指令格式: “:001r\_flick\_rgbc=123,234,345,678,\n”

指令解释:

“=”后面的值依次是 01-01 通道的“r(%d),g(%d),b(%d),w(%d),”，属于传感器 RGBW 原始 ADC 数据;该数据是 LED 点亮期间 RGBW 的最大值，该值的大小可以表征 LED 的光强，比例可以定性判断 LED 的颜色；

与 rgbw 指令读取结果一样；

////////////////////////////////////

---读取被测 LED 的亮灭状态(r\_led\_ch1)

发送指令格式: “:001r\_led\_ch101-02\n” //可快速判断 LED 亮灭状态，也可用于数码管的段显识别

返回指令格式: “:001r\_led\_ch1=0,1,\n”

指令解释:

发送: “01 代表 01(%02d)通道 CH1;

返回: 1 代表点亮，0 代表熄灭；

可以选择某一种光强数据与 limit\_flick 数据比较，比较规则和 flick 逻辑一样，参看 flick\_mode 指令；

---写入 LED 发射光源的亮灭状态(w\_led\_disp)

发送指令格式: “:001w\_led\_disp01-01=1\n” //只适用于自带 LED 发射端口的产品

返回指令格式: “:001w\_led\_disp01-01=1\n”

指令解释:

发送: “01 代表 01(%02d)led 输出通道 CH1，最多是 8CH;

---读取 LED 发射光源的亮灭状态(r\_led\_disp)

发送指令格式: “:001r\_led\_disp01-02\n” //只适用于自带 LED 发射端口的产品

返回指令格式: “:001r\_led\_disp=0,1,\n”

指令解释:

发送: “01 代表 01(%02d)led 输出通道 CH1;

返回: 1 代表点亮，0 代表熄灭；

////////////////////////////////////offset 功能-----适用 V19.3 及以后版本////////////////////////////////////

---使能 Offset 组别(w\_offset\_en)

发送指令格式: “:001w\_offset\_en01-02=8\n” 或 “:001w\_offset\_en01-02=0\n”

返回指令格式: “:001w\_offset\_en01-02=8\n” 或 “:001w\_offset\_en01-02=0\n”

指令解释:

发送: 01-02(%02d)是 led 连续通道 CH1-CH2;

8(%d):使能 CH1-CH2 的第 8 组别的 offset 参数，如果该值是 0，就是关闭 CH1-CH2 的 offset 功能;

---写入 Offset 组别的补偿值 dx(w\_offset\_dx)

发送指令格式: “:001w\_offset\_dx01-08=0.0001\n”

返回指令格式: “:001w\_offset\_dx01-08=0.0001\n”

指令解释:

发送: 01(%02d)是 led 通道 CH1;

08(%02d)是第一通道,第 8 组的 dx, 一共支持设置 8 组;

测量时可以随意打开其中一组 dx 参与运算, 也可以关闭所有组的 dx;

0.0001 是 dx 的数据,取值范围是( -1.0000 ~ 1.0000 ),在模块内部是与 x 坐标相加关系;

该参数使能后可以自动修正主波长 dwave 和色温 CCT 等色度参数;

---写入 Offset 组别的补偿值 dy(w\_offset\_dy)

发送指令格式: “:001w\_offset\_dy01-08=0.0001\n”

返回指令格式: “:001w\_offset\_dy01-08=0.0001\n”

指令解释:

发送: 01(%02d)是 led 通道 CH1;

08(%02d)是第一通道,第 8 组的 dy, 一共支持设置 8 组;

测量时可以随意打开其中一组 dx 参与运算, 也可以关闭所有组的 dy;

0.0001 是 dy 的数据,取值范围是( -1.0000 ~ 1.0000 ),在模块内部是与 y 坐标相加关系;

该参数使能后可以自动修正主波长 dwave 和色温 CCT 等色度参数;

---写入 Offset 组别的亮度值系数 kl(w\_offset\_kl)

发送指令格式: “:001w\_offset\_kl01-08=1.001\n”

返回指令格式: “:001w\_offset\_kl01-08=1.001\n”

指令解释:

发送: 01(%02d)是 led 通道 CH1;

08(%02d)是第一通道,第 8 组的 kl, 一共支持设置 8 组;

测量时可以随意打开其中一组 kl 参与运算, 也可以关闭所有组的 kl;

1.001 是 kl 的数据,取值范围(0.001-32.000 ),在内部与 lux 相乘关系,该参数只能改变照度 lux 数据;

---保存 offset 所有数据到 flash (w\_offset\_save)

发送指令格式: “:001w\_offset\_save\n”

返回指令格式: “:001w\_offset\_save\n”

指令解释: 该指令只能保存 offset 相关参数, 其他保存指令也不能保存 offset 参数;

该指令会擦写 flash, 寿命小于 10 万次, 尽量不要在主程序内频繁操作;

---清零 offset 所有数据(w\_offset\_clear)

发送指令格式: “:001w\_offset\_clear\n”

返回指令格式: “:001w\_offset\_clear\n”

指令解释: 清除 offset 相关所有通道所有组别数据, en 组别=0, dx=0,dy=0,kl=1.0;

---读取 offset 使能组别(r\_offset\_en)

发送指令格式: “:001r\_offset\_en01-02\n”

返回指令格式: “:001r\_offset\_kl=8,8\n”

指令解释:

发送: 01-02(%02d)是 led 连续通道 CH1-CH2;

8,8:使能了 CH1-CH2 的第 8 组别的 offset, 如果该值是 0,就是关闭了 CH1-CH2 的 offset 功能;

---读取 offset 补偿值 dx (r\_offset\_dx)

发送指令格式: “:001r\_offset\_dx01-02\n”

返回指令格式: “:001r\_offset\_dx= -0.0001\n”

指令解释:

发送: 01(%02d)是 led 通道 CH1;

02(%02d)是读取第一通道,第 2 组的 dx, 一共支持设置 8 组,-0.0001 是 dx 的数据;

**---读取 offset 补偿值 dy(r\_offset\_dy)**

发送指令格式: “:001r\_offset\_dy01-02\n”

返回指令格式: “:001r\_offset\_dy= -0.0001\n”

指令解释:

发送: 01(%02d)是 led 通道 CH1;

02(%02d)是读取第一通道,第 2 组的 dy, 一共支持设置 8 组,-0.0001 是 dy 的数据;

**---读取 offset 补偿值 kl(r\_offset\_kl)**

发送指令格式: “:001r\_offset\_kl01-02\n”

返回指令格式: “:001r\_offset\_kl= 1.001\n”

指令解释:

发送: 01(%02d)是 led 通道 CH1;

02(%02d)是读取第一通道,第 2 组的 kl, 一共支持设置 8 组,1.001 是 kl 的数据;

////////DIO 指令仅限选配 DIO 模块使用, 部分 LBB 系列仪器可选配 16DO,8DI 模块////////

**---读取 DI 单通道输入状态(r\_inbit)**

发送指令格式: “:001r\_inbit01\n”

返回指令格式: “:001r\_inbit=0\n”或“:001r\_inbit=1\n”

指令解释:

发送: “01 代表 01(%02d)通道(DIN1),每次读取一路输入点;

返回: 1 代表有输入(光耦导通), 0 代表无输入(光耦无导通);

**---读取 DI1-8 通道输入状态 U8(r\_in\_u8)**

发送指令格式: “:001r\_in\_u8\n”

返回指令格式: “:001r\_in\_u8=128\n”

指令解释:

发送: “u8”代表 DIN1-DIN8 从低到高组成的 uint8 字节数据(模块默认支持 8DI);

返回: DIN1-DIN8 依次由低到高位组合成一个字节数据, 128(1000 0000)代表 DIN8 有输入, 其他无输入;

**---写入 DO 多个通道输出状态(w\_outbit)**

发送指令格式: “:001w\_outbit01-02=1\n”

返回指令格式: “:001w\_outbit01-02=1\n”

指令解释:

发送: “01-02 代表从 01(%02d)通道到 02(%02d)通道, 必须是从小到大, 或者是相等(通道最大值为 16);

=后面的“1”代表打开 DO1-DO2, “0”代表关闭通道 DO1-DO2

返回: 如果指令正常, 返回值就和发送字符一样;

**---写入 DO-16 通道输出 U16(w\_out\_u16)**

发送指令格式: “:001w\_out\_u16=4\n”

返回指令格式: “:001w\_out\_u16=4\n”

指令解释:

发送: “u16”代表 DO1-DO16 从低到高组成的 uint16 整形数据;

=后面的“4”(0000 0000 0000 0100)代表打开 DO3, 其他通道全部关闭,

返回: 如果指令正常, 返回值就和发送字符一样;

//////////2022 年 2 月之后的仪器才支持 net 指令////////

#### ---写入 net 的工作模式(w\_net\_mode)

发送指令格式: “:001w\_net\_mode=1\n”

返回指令格式: “:001w\_net\_mode=1\n”

发送指令解释: net\_mode 取值范围是 0:Disable,1:TCP\_Server,1:TCP\_Client,3:UDP;

默认是关闭网口的, 一旦打开网口功能, 仪器初始化会有 5 秒钟网络连接时间, 如果需要网口通信, 建议 LED 分析仪工作在 1:TCP\_Server 模式下; 在没有启动网口之前, 所有 net 参数都是通过 USB/RS485/RS232 串口协议配置的, 所有 net 参数都是可以通过 save\_to\_flash 指令保存的;

返回指令解释: 如果指令正常, 返回值就和发送字符一样;

#### ---读取 net 的工作模式(r\_net\_mode)

发送指令格式: “:001r\_net\_mode\n”

返回指令格式: “:001r\_net\_mode=1\n”

#### ---写入 net 的仪器的源 IP 地址(w\_net\_sip)

发送指令格式: “:001w\_net\_sip=192,168,0,100\n”

返回指令格式: “:001w\_net\_sip=192,168,0,100\n”

发送指令解释: sip 是 LED 分析仪网口的 IP 地址, 该网络所有 ip 满足 IPV4 格式;

为了兼容其他解析数字指令, IP 地址数字间隔是用逗号(',')分割, 不是用点('.');

返回指令解释: 如果指令正常, 返回值就和发送字符一样;

#### ---读取 net 的仪器的源 IP 地址(r\_net\_sip)

发送指令格式: “:001r\_net\_sip\n”

返回指令格式: “:001r\_net\_sip=192,168,0,100,\r\n”

#### ---写入 net 的上位机的目标 IP 地址(w\_net\_dip)

发送指令格式: “:001w\_net\_dip=192,168,0,10\n”

返回指令格式: “:001w\_net\_dip=192,168,0,10\n”

发送指令解释: dip 是上位机电脑的网口的 IP 地址,

只有 LED 分析仪网口工作在 TCP\_Client 模式下, dip 配置才能用到;

返回指令解释: 如果指令正常, 返回值就和发送字符一样;

#### ---读取 net 的上位机的目标 IP 地址(r\_net\_dip)

发送指令格式: “:001r\_net\_dip\n”

返回指令格式: “:001r\_net\_dip=192,168,0,10,\r\n”

#### ---写入 net 的网关 Gateway IP 地址(w\_net\_gip)

发送指令格式: “:001w\_net\_gip=192,168,0,1\n”

返回指令格式: “:001w\_net\_gip=192,168,0,1\n”

发送指令解释: gip 是网口的网关 Gateway IP 地址,

返回指令解释: 如果指令正常, 返回值就和发送字符一样;

#### ---读取 net 的网关 Gateway IP 地址(r\_net\_gip)

发送指令格式: “:001r\_net\_gip\n”

返回指令格式: “:001r\_net\_gip=192,168,0,1,\r\n”

---写入 net 的子网掩码 Subnet Mask IP 地址(w\_net\_mip)

发送指令格式: “:001w\_net\_mip=255,255,255,0\n”

返回指令格式: “:001w\_net\_mip=255,255,255,0\n”

发送指令解释: gip 是网口的网关 Gateway IP 地址,

返回指令解释: 如果指令正常, 返回值就和发送字符一样;

---读取 net 的子网掩码 Subnet Mask IP 地址(r\_net\_mip)

发送指令格式: “:001r\_net\_mip\n”

返回指令格式: “:001r\_net\_mip=255,255,255,0,\r\n”

---写入 net 的 LED 仪器的源端口号(w\_net\_sport)

发送指令格式: “:001w\_net\_sport=8000,8001,8002,8003,8004,8005,8006,8007\n”

返回指令格式: “:001w\_net\_sport=8000,8001,8002,8003,8004,8005,8006,8007\n”

发送指令解释: sport 是 LED 分析仪网口端口号, 在 TCP\_Server 模式下, 支持 8 个同时通信; 在 TCP\_Client 模式下, 只采用第一个端口号(8000)访问上位机服务器;

返回指令解释: 如果指令正常, 返回值就和发送字符一样;

---读取 nnet 的 LED 仪器的源端口号(r\_net\_sport)

发送指令格式: “:001r\_net\_sport\n”

返回指令格式: “:001r\_net\_sport=8000,8001,8002,8003,8004,8005,8006,8007,\r\n”

---写入 net 的上位机服务器的目标端口号(w\_net\_dport)

发送指令格式: “:001w\_net\_dport=10000\n”

返回指令格式: “:001w\_net\_dport=10000\n”

发送指令解释: dport 是 PC 上位机作为 TCPServer 时端口号, 只有 LED 分析仪在 TCP\_Client 模式下, 该端口号才有作用;

返回指令解释: 如果指令正常, 返回值就和发送字符一样;

---读取 net 的上位机服务器的目标端口号(r\_net\_dport)

发送指令格式: “:001r\_net\_dport\n”

返回指令格式: “:001r\_net\_dport=10000\n”

---读取 net 的 MAC 地址(r\_net\_mac)

发送指令格式: “:001r\_net\_mac\n”

返回指令格式: “:001r\_net\_mac=101,102,103,104,105,106,\r\n”

指令解释: mac 格式也是按照十进制字符串返回的, 不是按照 16 进制字符串格式;

---读取 net 的所有配置参数(r\_net\_all)

发送指令格式: “:001r\_net\_all\n”

返回指令格式: “:.....\r\n”

指令解释: 该指令适合在串口调试助手下, 手动读取网络配置信息,

网口功能启动后, LED 分析仪在复位初始化时, 会从 USB 接口打印出来网口所有的配置信息;

---试运行 net 网口通信连接(w\_net\_run)

发送指令格式: “:001w\_net\_run\n”

返回指令格式: “:001w\_net\_run\n”

指令解释: 网线插上后, net 参数配置完成后, 可以先发送该指令试运行一下网络连接, 该指令发送后, LED 分析仪上的 D9 指示灯会连续闪烁 4 秒连接网络, 期间 USB 接口谁自动打印网络参数信息, RJ45 上的橙,绿色灯也会点亮; 一旦网络通信连接正常, 说明配置参数正确, 那就可以 save\_to\_flash 保存起来, 后续上电后网口会自动工作在网络连接状态;

#### 四: 指令表汇总:

串口指令表(地址为 001)			
指令分类	指令关键字串	功能描述	备注
状态查询	":001state\n"	查询模块状态 ":001idle\r\n"或者":001busy\r\n"	“
模块信息	:001idn	读取模块产品信息	
模块 ID	:000r_id	读取模块的 ID	不管 RS232 或者 RS485 接口, 都要用模块地址(保持协议格式不变)
	:001w_id	写入模块 ID(%03d),立即生效并自动保存该参数	
波特率 Baud	w_baud1	设置 USB-RS232 波特率, 立即生效并自动保存该参数	Baud 范围 0-9
	w_baud2	设置 RS485 端口波特率, 立即生效并自动保存该参数	{2400,4800,9600,19200,38400,57600,115200,230400,460800,921600}
配置 ADC 采样模式	w_system_samp	配置 ADC 采样模式 0: 连续模式; 1: 单次采样(V19.6 版)	
读取 ADC 采样模式	r_system_samp	读取 ADC 采样模式 (V19.6 版)	
复位系统	w_system_reset	系统从新复位初始化, 不是恢复出厂设置(V19.3 版)	
参数保存	save_to_flash	把用户配置在 RAM 中参数保存到 flash, 掉电不丢失, 不建议频繁保存	
恢复出厂设置	default	所有用户配置的参数恢复出厂设置	
采样周期	w_ft	设置传感器采样周期, 所有通道同步采样, 立即执行, 暂存在 RAM	ft 越大, 数据越大, 采样周期越长
	r_ft	读取传感器采样时间索引号	
	r_ftms	读取传感器采样时间 ms (V1.5 版本)	
ADC 增益	w_gain	设置传感器硬件增益, 立即执行, 暂存在 RAM	增益越大, 数据越大
	r_gain	读取传感器增益索引号	
	r_gainx	读取传感器增益实际值 (V1.5 版本)	

RGB 数据 (只有 RGB 和 XYZ 系列 才有此功 能)	r_rgbw	读取传感器 RGBW 原始 ADC 值	
	r_u8_rgb	读取经过换算到 0-255 范围的 RGB (V1.4 版本-不建议采用)	
	r_rgb_i	读取换算到 0-255 范围的 RGB 和相对功率强度 I(0-100%)	
	r_hsl_i	读取色度/饱和度/亮度/相对亮度 I(0-100%)	
Lux 照度/功 率	w_k_lux	写入 lux 补偿系数,立即执行, 暂存于 RAM 中	
	r_k_lux	读取 lux 补偿系数 k(%0.3f):	
	w_k_uw	写入光功率补偿系数,立即执行, 暂存于 RAM 中	
	r_k_uw	读取光功率补偿系数 k(%0.3f)默认为 1.0:	
	r_lux	读取照度值 lux, 或者是亮度值 mcd/m^2,	
	r_uw_cm	读取光功率 uw/cm^2,一般用于表征非可见光, 某些产品不适用	
	r_cd_mm	读取亮度 cd/m^2,面光源亮度数据, 适合非光纤类产品 (V19.3 版本)	
流明 lm	r_lm	读取积分球探头的流明值, 流明探头出厂前已校准, 与 lux 成正比关系	积分球反射率和面积 cm^2
坎德拉 cd	r_cd_lm	读取 LED 的坎德拉 cd 和流明 (适用于朗伯体光源)	探测角度和距离
Flick 频闪功能			
	w_flick_mode	设置 LUX-R-G-B-W 为对比对象- (V1.5 版本)立即执行	
	r_flick_mode	读取 flick 对比通道模式- (V1.5 版本)	
	w_flick_limit	设置 flick 亮度门限值, 立即执行, 暂存于 RAM 中	
	r_flick_limit	读取 flick 亮度门限值	阈值 001-999lux
	w_flick_ts	设置并立即执行 flick 测试- (V1.5 版本)	
	r_flick_ts	读取 flick 结果(Cnt,fre,Tup,Tdw,Tduty,LUXmax)- (V1.5 版本)	
	r_flick_lx	读取 flick 期间的点亮时的 lux 数据- (V1.5 版本)	
	r_flick_chroma	读取 flick 期间的点亮时的 chroma 数据- (V19.6 版本)	
	r_flick_rgb_i	读取 flick 期间的点亮时的 rgb_i 数据- (V19.6 版本)	
	r_flick_rgbc	读取 flick 期间的点亮时的 rgbc 原始数据- (V19.6 版本)	
	w_flick_flow	设置并立即执行 flick_flow 测试- (V20.101 版本)	
	r_flick_flow	读取 flick_flow 结果 (V20.101 版本)	
	w_flick_edge	设置并立即执行 flick_flow 测试- (V21.051 版本)	
	r_flick_edge	读取 flick_edge 结果 (V21.051 版本)	
读取 LED 亮	r_led_ch1	快速读取每个通道是否有 led 亮灭, 适合数码管测量	

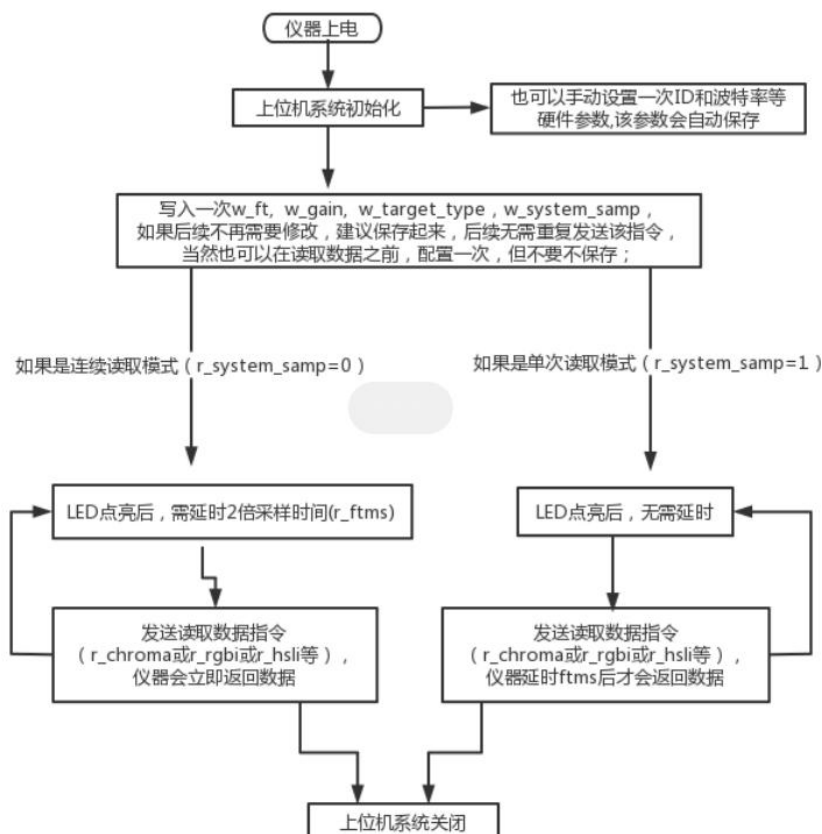
灭状态			
chroma 色度 数据 (只有 XYZ 模块读取的 数据是准确 的, 为了兼容通 信协议, 这 些指令也适 用于 RGB 系 列产品, 但 获取的数据 不准确)	w_target_type	选择待测目标 LED 的类型, 立即执行, 暂存于 RAM	选择合适的类型可以提高色坐标的测量精度
	r_target_type	读取当前的 led 测试类型	
	r_Yxy	读取(CIE1931)三刺激色坐标和亮度	
	r_xy	读取 xy(CIE1931)色坐标数据	
	r_uv	读取 uv(CIE1976)色坐标数据	
	r_cct	读取 CCT(CIE1931)色温数据	
	r_cctd	读取 CCT 及 Duv 数据 (V1.5 版本)	
	r_dowave	读取主波长(CIE1931)数据	
	r_wavesi	读取主波长, 色纯度(色饱和度), lux 数据 (V1.5 版本)	
	r_chroma	读取 lux, x, y, dowave, duty%, cct, duv 色度数据; (V1.5 版本)	
	w_sdcn_type	选择 sdcn 类型, 立即执行, 暂存于 RAM	
	r_sdcn_type	读取当前通道的 sdcn 类型	
LED 发射光源	r_sdcn_data	读取读取当前通道的 sdcn	
	r_sdcn_lux	读取读取当前通道的 lux, sdcn, sdcn_type	
DIO 读写 (配 DIO 的 模块才可使 用)	w_led_disp	打开 LED 的亮灭状态, 只适合带 LED 发射端口的产品	
	r_led_disp	读取 LED 的亮灭配置状态, 只适合带 LED 发射端口的产品	
	r_inbit	读取一位输入信号	光耦输入
	r_in_u8	读取 8 位输入组成一个字节	
Offset (V19.3 版 本)	w_outbit	写入指定地址的 DO 状态	ULN2803-NPN 晶体管输出 (200MA)
	w_out_u	写入 16 位 DO 状态	
	w_offset_en	读写某些通道的某个组别功能	r_offset_en
	w_offset_dx	读写 cie1931-x 的 offset_dx 参数	r_offset_dx
	w_offset_dy	读写 cie1931-x 的 offset_dy 参数	r_offset_dy
	w_offset_kl	读写 lux 的 offset_kl 参数	r_offset_kl
	w_offset_save	保存所有 offset 相关参数	
	w_offset_clear	清零所有 offset 相关参数	
Net 网络参			
	w_net_mode	读写网口的工作模式	r_net_mode
	w_net_sip	读写仪器的源 ip 地址	r_net_sip

数配置指令	w_net_dip	读写上位机的目标 ip 地址, 一般是电脑的 ip	r_net_dip
适用于	w_net_gip	读写网络的网关 Gateway IP 地址	r_net_gip
2022-2 月之	w_net_mip	读写网络的子网掩码 Subnet Mask IP 地址	r_net_mip
后带 LAN 网	w_net_sport	读写仪器网口的源端口号, 支持 8 个 ports	r_net_sport
口的新仪器	w_net_dport	读写上位机的网口目标端口号, 只有一个	r_net_dport
	r_net_mac	读取网口的 MAC 地址	
	w_net_run	软件初始化一次网口, 但网络参数并没有保存到 flash	
	r_net_all	读取所有网络配置信息, 只用在串口调试助手上	
强烈建议采用官方调试软件, 通过 USB 或 RS485 配置 net 参数, 方便省事!			

## 五：软件开发流程如下

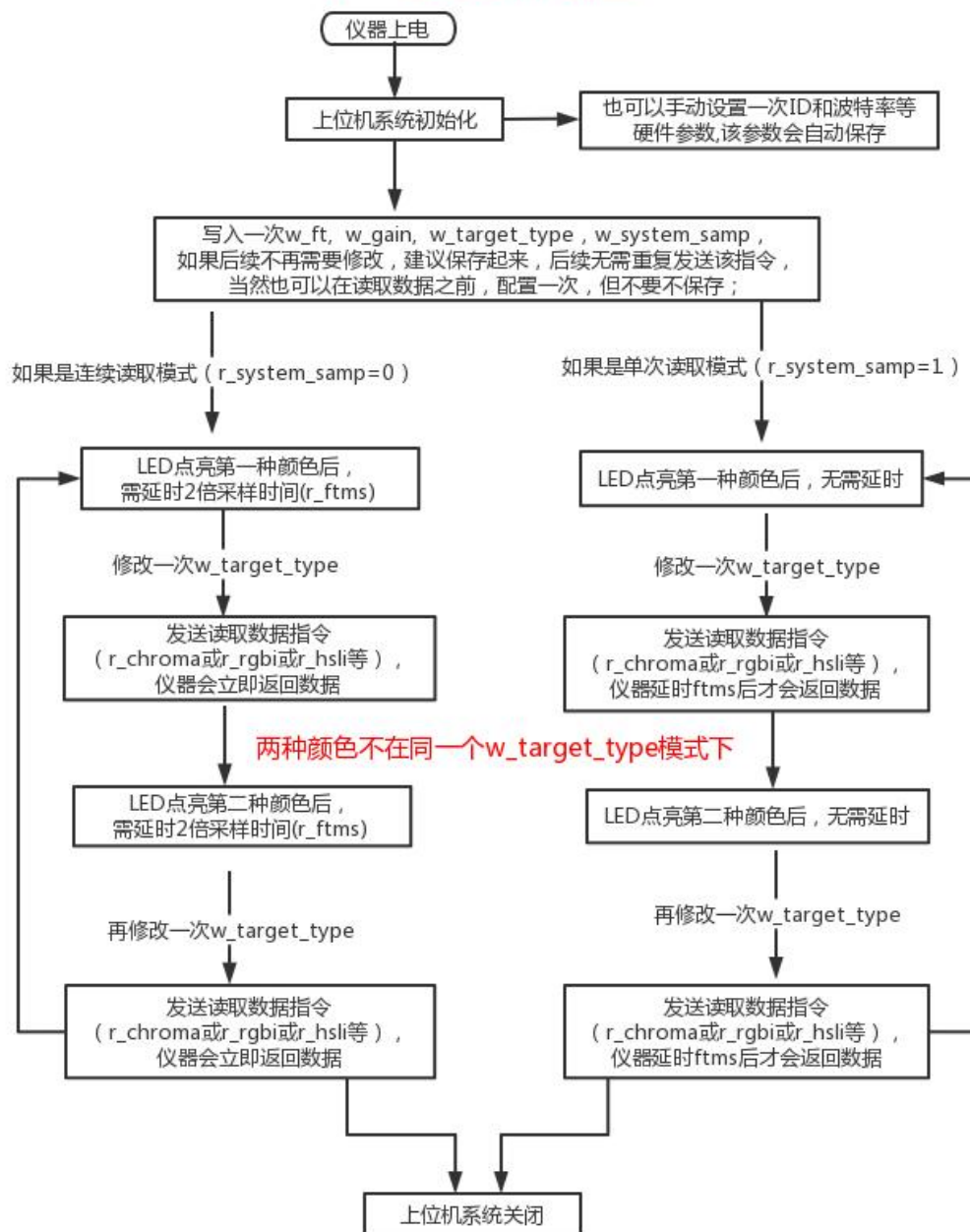
### ---单色常亮 LED 数据采集流程

#### 单色常亮LED的光学数据采集流程



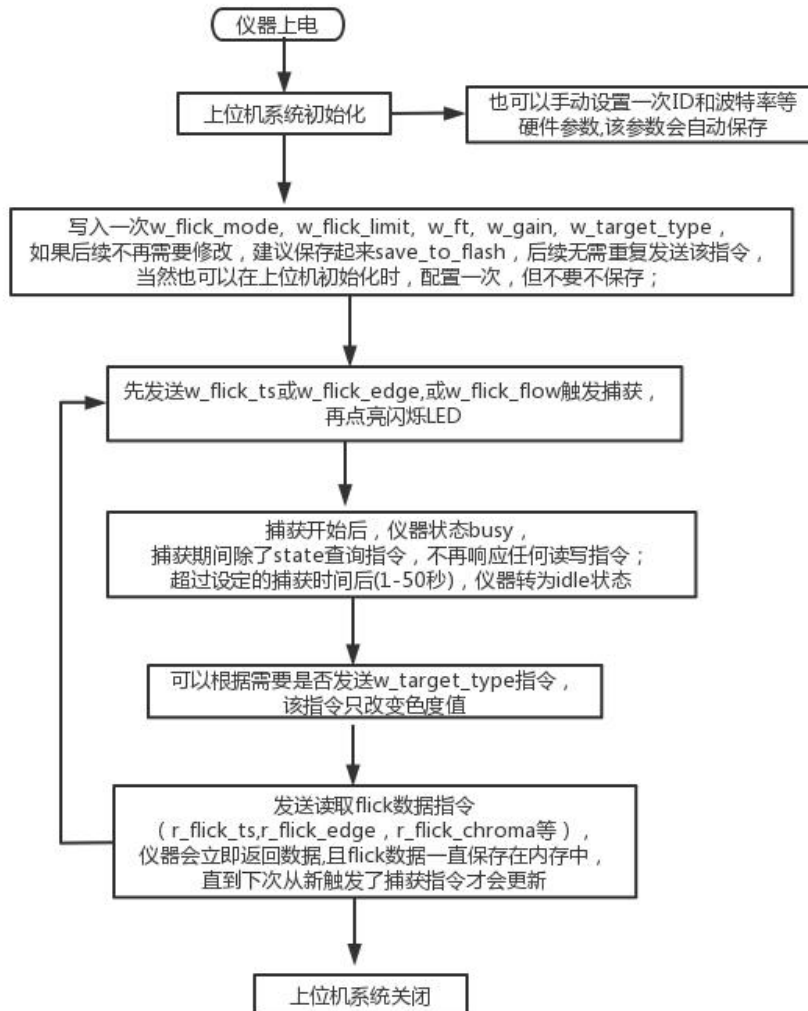
### ---双色常亮 LED 数据采集流程

## 双色常亮LED的光学数据采集流程 -(XYZ系列仪器及以上)



---单色闪烁 Flick-LED 数据采集流程

## 单色闪烁Flick-LED的光学数据采集流程



## 六：测试用例-指令读写实例

---读取 1CH 的 RED-LED 的 RGBI/HSI(ID=1):

序号	发送指令(C 语言格式)	返回指令	备注
1	“:001w_gain01-01=4\n” “:001w_ft01-01=4\n”	“:001w_gain01-01=4\n” “:001w_ft01-01=4\n”	光越弱, gain, ft 要配置越大, 每款传感器的 gain 和 ft 范围不一样, 但仪器内部有软件防呆功能, 写入非法的序号, 主机不会生效; 一般情况下, 出厂默认配置就可以满足大部分测量要求, 除非 LED 特殊, 一般不需要频繁修改这两个硬件参数
2	“:001r_rgb01-01\n”	“:001r_rgb=255,50,10,1.23\n”	RGB/HSI 系列仪器建议用其中一条指令判断颜色亮度即可, 不建议用 CIE1931-xy 专业光学参数
3	“:001r_hsi01-01\n”	“:001r_hsi=5,95,55,1.23\n”	

---读取 1CH 的黄绿灯的主波长(ID=1):

序号	发送指令(C 语言格式)	返回指令	备注
1	“:001w_gain01-01=4\n” “:001w_ft01-01=4\n”	“:001w_gain01-01=4\n” “:001w_ft01-01=4\n”	可以配置一次保存起来(save_to_flash), 不用频繁发送;
2	“:001w_target_type01-01=4\n”	“:001w_target_type01-01=4\n”	选择黄绿色的 LED 读取模式 4, 也可以配置一次保存起来(save_to_flash), 不用频繁发送;
3	“:001r_wavesi01-01\n”	“:001r_wavesi=570.5,80,345.6”	也可以用“:001r_chroma01-01\n”读取

---读取 1CH 的白光灯的 xy 色坐标(ID=1):

序号	发送指令(C 语言格式)	返回指令	备注
1	“:001w_gain01-01=4\n” “:001w_ft01-01=4\n”	“:001w_gain01-01=4\n” “:001w_ft01-01=4\n”	可以配置一次保存起来(save_to_flash), 不用频繁发送;
2	“:001w_target_type01-01=0\n”	“:001w_target_type01-01=0\n”	选择白光的 LED 读取模式 0, 也可以用 3, 建议用 0
3	“:001r_Yxy01-01\n”	“:001r_Yxy=123.4,0.3333,0.3333”	也可以用“:001r_chroma01-01\n”读取

---读取 1CH 的 RED-LED-flick 闪烁频率(ID=1):

序号	发送指令(C 语言格式)	返回指令	备注
1	“:001w_gain01-01=4\n” “:001w_ft01-01=4\n”	“:001w_gain01-01=4\n” “:001w_ft01-01=4\n”	Ft 的设置采样时间要小于二分之一 LED 频闪期间的点亮时间
2	“:001w_flick_mode01-01=0\n” “:001w_flick_limit01-01=20\n”	“:001w_flick_mode01-01=0\n” “:001w_flick_limit01-01=20\n”	如果单色 LED 是固定的闪烁状态, 可以一次配置后, 保存起来(save_to_flash), 不用频繁发送;
3	“:001w_flick_ts01-01=5\n”	“:001w_flick_ts01-01=5\n”	根据需要也可以用“:001w_flick_edge01-01=3\n”
4	“:001r_flick_ts01-01\n”	“:001r_flick_ts=1.0,1000,1000,500,5\n”	上位机必须延时等待捕获时间到期后, 才可以发送读取指令, 不建议在捕获期间连续发送 state 查询指令, 这样会影响一点主机的处理速度
5	“:001r_flick_rgb01-01\n”	“:001r_flick_rgb=255,50,10,1.23\n”	

## 七：编程注意事项-必看！

### 1，配置参数保存 save\_to\_flash

出厂默认配置基本可以满足大部分测量场合，相关参数都是可以程控配置的，配置完成后都是可以选择保存到 flash 或 eeprom，掉电不丢失，但不建议频繁保存该指令，次数太多会擦坏内部 flash 或 eeprom；

### 2，增益 gain 和采样时间 ft

其中 w\_gain 和 w\_ft 指令是可以配置光学传感器的 ADC 采集增益和采样时间，rgbw/r\_adc/r\_spect 等指令可以读取最原始的 ADC 值，相同光强下，w\_gain 写入参数越大，ADC 值越大，w\_ft 写入参数越大，ADC 值越大，但不能让 ADC 值饱和，ADC 是否饱和要看 r\_rgbi 中的 i(0%-100%)来判断，一旦 ADC 饱和，那么测量的光学数据将失真，但 gain 和 ft 太小导致 ADC 过小，又会降低光学数据的分辨率；一般来说，i 的区间取 1%-60%是最合适；Gain 和 ft 基本不影响 lux 的数值，只是改变了 lux 的分辨率；可以在初始化设置一次 gain 和 ft，或者直接提前配置保存好，掉电不丢失；

### 3，采样模式 w\_system\_samp:

**连续模式 0(默认模式):**仪器内所有通道会按照指定的 FT 时间自动刷新数据，采样起始时间不受上位机软件控制，LED 点亮后，要上位机软件延时 2 倍的 ftms 时间再去读取数据，才能保证是在 LED 点亮后更新出来的完整光强数据，这个好处在于，接口通信发送读取指令后，仪器立即返回数据，不用再等待一个采样时间周期后收到仪器返回的数据，可以连续发送多条读取指令，读取同一时刻测量的不同的光学参数；

**单次模式 1:**仪器内所有通道会等待上位机指令软件触发采集，LED 点亮后，上位机可以理解发送数据读取指令，仪器收到指令后会立即开始采集数据，等待 FT 时间后，再发送采集到的数据；这个好处在于，在只读取一条光学数据时，可以缩短数据获取时间；可以在初始化设置一次模式，或者直接提前配置保存好，掉电不丢失；

### 4，LED 光源类型选择 w\_target\_type

这条指令的目的是提高光学参数的准确度，虽然每种模式都可以测量任何光源，但选择合适的类型，可以大大提高光学数据的准确度，一些低版本的产品不支持(亮度系列 LTD/RGB 系列)；如果是多色灯，可以在读取光学数据之前实时发送对应的类型，如果是单色灯，可以在初始化设置一次，或者直接提前配置保存好，掉电不丢失；

### 5，频率流水灯捕获指令 flick/flow:

需要关注亮灭的阈值 w\_flick\_limit，和采样时间 ft，limit 设置在亮灭亮度值的中间偏低位置，ft 一定要小于 LED 闪烁周期的一半以内，ft 越小，周期分辨率越高，但 ADC 越小，在不影响光学数据分辨率的情况下，尽量减小 ft；一旦捕获指令执行完成后，在频闪期间捕获的数据，都会一直暂存于内存中，可以随时去读，直到下次从新触发捕获指令后，才会被更新改变；

### 6，r\_rgbi 与 r\_chroma 颜色识别指令:

r\_rgbi 指令中的 rgb 是颜色比例值，并不代表亮度信息，i 是仅代表相对光强度值，rgb 也没有任何光度参考标准，只能定性判断 LED 颜色和亮度，一般以待测合格样品的 rgbi 为标准值，比如一个红光的 rgb(255,100,30),r=255，但 gb 并不会等于 0；

r\_chroma 中的参数全部是国际照明协会规定的 CIE-1931 的色度值，可以定量判断 LED 的色度和亮度，具体参数代表的含义，需要提前了解一些最基本的光度学概念；

### 7，Net 通信说明:

---

支持网口通信的 LED 分析仪，底层是 TCP/IP 硬件协议层，但顶层应用层的通信指令还是和串口通信指令完全一样，因此本文档上的字符串通信指令全部兼容网口通信指令；

#### 八：手册声明：

我司 LED 系列产品开发资料会不断更新，我们会尽可能兼容之前的协议，如有差错，请以最新手册作为开发依据；开发资料编写如发现错误，欢迎指正修改；如有疑问，请及时联系我司技术人员协助解答；最终解释权归我司所有！