

光源控制器 DLL 函数说明书

Developer's Guide

Made By Shuangyi

光源控制器 DLL 函数说明书

光源控制器 DLL 函数说明书.....	2
1.List of Function 函数列表说明.....	4
1.1 获取/查询/释放串口端口资源&光源控制模块资源.....	4
1.2 光源控制器的相关命令.....	5
1.3 光源控制器的编程流程图.....	6
1.3.1 打开串口.....	7
1.3.2 设定模块 ID&连接光源控制器模块.....	7
1.3.3 参数设定.....	8
1.3.4 打开通道开关&PWM 调节.....	8
1.3.5 保存参数到 Flash 中.....	9
1.3.6 关闭通道开关&释放端口.....	9
2.1 获取/查询/释放串口端口资源.....	10
2.1.1 串口连接.....	10
2.1.2 释放端口.....	10
2.1.3 设置串口号.....	10
2.1.4 获得当前使用串口号.....	10
2.2 光源控制器模块相关参数.....	11
2.2.1 光源控制器模块初始化.....	11
2.1.2 连接光源控制器模块.....	11
2.1.3 查询光源控制器模块状态信息.....	11
2.1.4 更改光源控制器模块 ID 号.....	11
2.1.5 得到当前光源控制器模块 ID 号.....	11
2.1.6 获得链路上所有连接的模块 ID 号.....	12
2.3 光源控制器的相关函数.....	12
2.3.1 设置通道亮度值.....	12
2.3.2 得到通道亮度值.....	12
2.3.3 设置通道触发延迟时间.....	12
2.3.4 得到通道触发延迟时间.....	13
2.3.5 设置通道频闪保持时间.....	13
2.3.6 得到通道频闪保持时间.....	13
2.3.7 设置光源控制模式.....	14
2.3.8 得到光源控制模式.....	14
2.3.9 设置光源开关状态.....	14
2.3.10 载入配置文件.....	15
2.3.11 保持配置文件.....	15
2.3.12 保存到 Flash.....	15
2.3.13 软件切换通道.....	15
2.3.14 设置通讯模式.....	16
2.3.15 同步设置 4 通道亮度值.....	17
2.3.16 同步设置 4 通道开关.....	17

2.3.17 同步通道开关状态.....	18
2.3.18 设置通讯波特率.....	18
2.3.19 设置通道 PWM 和通道状态.....	19
2.3.20 设置通道电流输出值.....	19
2.3.21 得到通道电流输出值.....	19
修订记录.....	21

1.List of Function 函数列表说明

根据所需各项功能提供相应的接口。

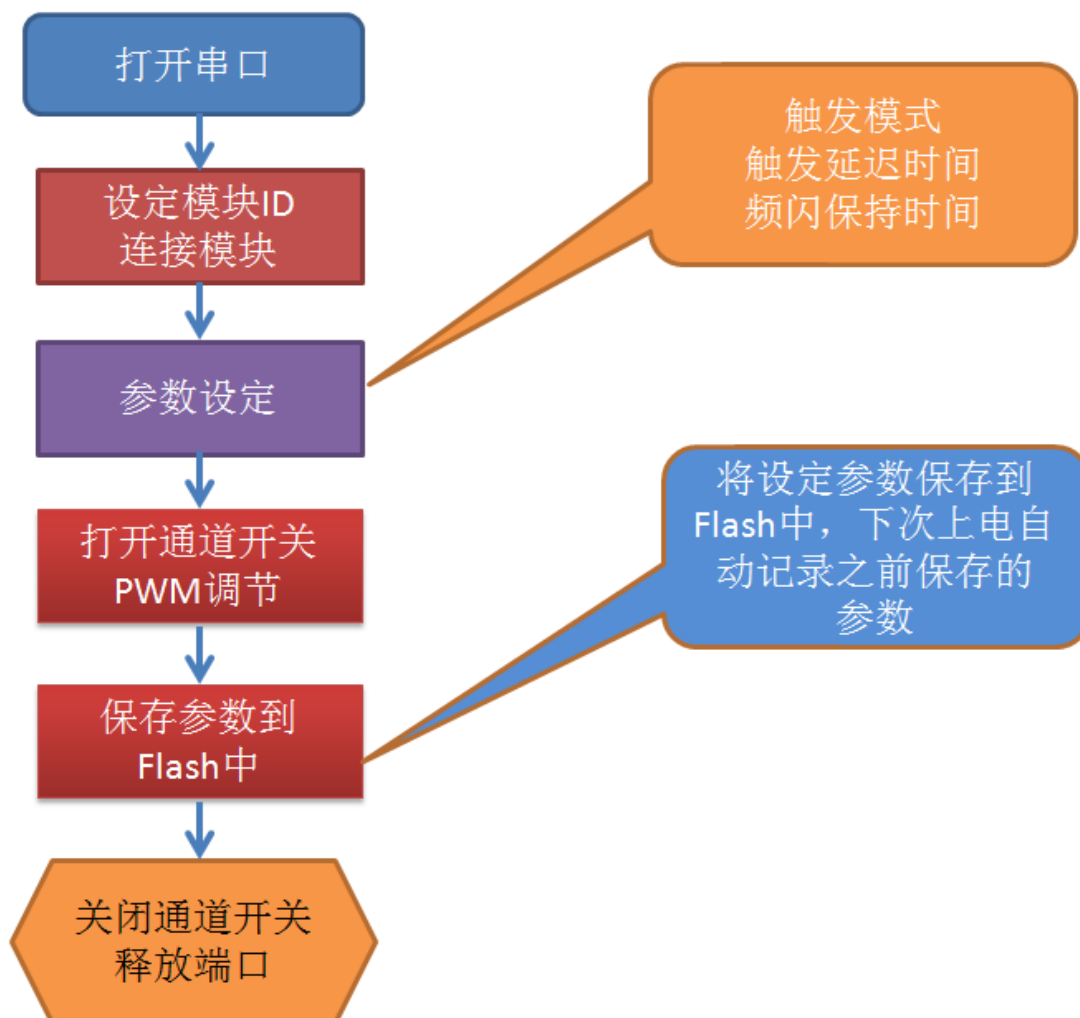
1.1 获取/查询/释放串口端口资源&光源控制模块资源

Function Name(函数名称)	说明
SY_MVD_ComPort_Connect	串口连接
SY_MVD_Light_Disconnect	释放端口
SY_MVD_Light_Set_PortNum_Config	设置串口 ID 号
SY_MVD_Light_Get_PortNum_Config	得到当前使用串口 ID 号
SY_MVD_Light_Slave_Init	光源控制器模块初始化
SY_MVD_Light_Slave_Connect	光源控制器模块连接
SY_MVD_Slave_ConnectSts	模块连接状态查询
SY_MVD_Light_Set_SlaveIP	改变模块 ID 号
SY_MVD_Light_Get_SlaveIP	得到当前模块 ID 号
SY_MVD_Light_Slave_Init	模块初始化
SY_MVD_Light_Get_AllSlaveIP	查询连接的模块 ID

1.2 光源控制器的相关命令

Function Name (函数名称)	说明
SY_MVD_Light_Set_Intensity	设置亮度
SY_MVD_Light_Get_Intensity	得到亮度值
SY_MVD_Light_SetCHMode	光源通道模式(PWM 或者触发模式)
SY_MVD_Light_GetCHMode	得到光源通道模式(PWM 或者触发模式)
SY_MVD_Light_Set_TriggerDelay	光源通道触发延迟时间
SY_MVD_Light_Get_TriggerDelay	得到光源通道触发延迟时间
SY_MVD_Light_Set_StrobeDuration	光源通道频闪保持时间
SY_MVD_Light_Get_StrobeDuration	得到光源通道频闪保持时间
SY_MVD_Light_SelChn	软件切换模块的通道
SY_MVD_Light_SaveParamToFlash	参数保存到 Flash
SY_MVD_LoadLightParam	将参数存到文件中
SY_MVD_SaveLightParam	从文件中读取参数
SY_MVD_Light_SetChStsOnOff	PWM 模式下打开或者关闭某个通道, 或者频闪
SY_MVD_Light_Set_AllIntensity	同步设定 4 个通道的 PWM 值
SY_MVD_Light_Set_AllChStsOnOff	同步设定打开或者关闭 4 个通道
SY_MVD_Light_Get_AllChStsOnOff	获得设定 4 个通道的状态
SY_MVD_Light_Set_Baudrate	设定控制器的波特率
SY_MVD_Light_Set_Intensity_ChanelSts	独立控制 4 个通道 PWM 值和 4 个通道开关命令
SY_MVD_Light_Set_CurrentOut	设置通道电流输出值
SY_MVD_Light_Get_CurrentOut	得到当前通道电流输出值

1.3 光源控制器的编程流程图



程序框架图

1.3.1 打开串口

Function Name (函数名称)	说明
SY_MVD_Light_Set_PortNum_Config	设置串口 ID 号
SY_MVD_ComPort_Connect	打开串口连接

```
//串口号为 3
int m_ComPort = 3;
SY_MVD_Light_Set_PortNum_Config(m_ComPort);
if (!SY_MVD_ComPort_Connect())
{
    MessageBox("串口连接失败!");
    return;
}
else
{
    MessageBox("串口连接成功!");
}
```

1.3.2 设定模块 ID&连接光源控制器模块

Function Name (函数名称)	说明
SY_MVD_Light_Slave_Connect	光源控制器模块连接
SY_MVD_Slave_ConnectSts	模块连接状态查询

```
//模块 ID 默认为 10
int m_ModuleIP;
m_ModuleIP = 10;
if (!SY_MVD_Light_Slave_Connect(m_ModuleIP))
    return;
if (SY_MVD_Slave_ConnectSts())
    MessageBox("光源控制器模块连接成功!");
else
    MessageBox("光源控制器模块连接失败!");
```

1.3.3 参数设定

Function Name (函数名称)	说明
SY_MVD_Light_SetCHMode	光源通道模式(PWM 或者触发模式)
SY_MVD_Light_Set_TriggerDelay	光源通道触发延迟时间
SY_MVD_Light_Set_StrobeDuration	光源通道频闪保持时间

```
//设定通道 0 参数，模式为 PWM 常亮模式，触发延迟时间为 10000us
//频闪保持时间为 20000us
int m_ModuleIP;
m_ModuleIP = 10;
SY_MVD_Light_SetCHMode(m_ModuleIP,0,0);
SY_MVD_Light_Set_TriggerDelay(m_ModuleIP,0,10000);
SY_MVD_Light_Set_StrobeDuration(m_ModuleIP,0,20000);
```

1.3.4 打开通道开关&PWM 调节

Function Name (函数名称)	说明
SY_MVD_Light_SetChStsOnOff	PWM 模式下打开或者关闭或频闪
SY_MVD_Light_Set_Intensity	设置亮度
SY_MVD_Light_Set_AllChStsOnOff	同步设定打开或者关闭 4 个通道
SY_MVD_Light_Set_AllIntensity	同步设定 4 个通道的 PWM 值

```
//通道 0 打开开关，并设置通道 0 亮度值为 100
SY_MVD_Light_SetChStsOnOff(m_ModuleIP,0,1);
SY_MVD_Light_Set_Intensity(m_ModuleIP, 0, 100); //设置通道 0 的光源亮度

//同时打开 4 个通道
SY_MVD_Light_Set_AllChStsOnOff(m_ModuleIP,1);

//同时设定 4 个通道的亮度值,亮度值分别为通道 0:100; 通道 1:99;通道 2:98;通道 3:97;
int chvalue0 = 100;
int chvalue1 =99;
int chvalue2 =98;
int chvalue3 =97;
SY_MVD_Light_Set_AllIntensity(m_ModuleIP,chvalue0,chvalue1,chvalue2,chvalue3);
```


1.3.5 保存参数到 Flash 中

Function Name (函数名称)	说明
SY_MVD_Light_SaveParamToFlash	参数保存到 Flash

```
//将参数表保存到 Flash 中  
int m_ModuleIP;  
m_ModuleIP = 10  
SY_MVD_Light_SaveParamToFlash(m_ModuleIP);
```

1.3.6 关闭通道开关&释放端口

Function Name (函数名称)	说明
SY_MVD_Light_SetChStsOnOff	PWM 模式下打开或者关闭
SY_MVD_Light_Set_AllChStsOnOff	同步设定打开或者关闭 4 个通道
SY_MVD_Light_Disconnect	释放端口

```
//通道 0 关闭  
SY_MVD_Light_SetChStsOnOff(m_ModuleIP,0,0);  
  
//同时关闭 4 个通道  
SY_MVD_Light_Set_AllChStsOnOff(m_ModuleIP,0);  
  
//释放端口  
SY_MVD_Light_Disconnect();
```

2.函数说明

2.1 获取/查询/释放串口端口资源

2.1.1 串口连接

BOOL SY_MVD_ComPort_Connect()

说明：串口端口连接

输出参数：返回：TRUE ：成功； FALSE ：失败

输入参数：无

2.1.2 释放端口

BOOL SY_MVD_Light_Disconnect()

说明：释放端口

输出参数：TRUE ：成功； FALSE ：失败

输入参数：（无）

2.1.3 设置串口号

BOOL SY_MVD_Light_Set_PortNum_Config (int PortNum)

说明：设置串口端口号

输出参数；返回 True 成功；返回 False 表示端口设置失败

2.1.4 获得当前使用串口号

BOOL SY_MVD_Light_Get_PortNum_Config (int *PortNum)

说明：得到当前使用串口 ID 号

2.2 光源控制器模块相关参数

2.2.1 光源控制器模块初始化

BOOL SY_MVD_Light_Slave_Init (int SlaveIP)

说明：光源控制器模块初始化，执行此函数后，光源控制器所有参数为出厂设定值。

输入参数：SlaveIP：模块 ID 号，控制器的 ID（1~32）

输出参数：TRUE：成功； FALSE：失败

2.1.2 连接光源控制器模块

BOOL SY_MVD_Light_Slave_Connect ()

说明：光源控制器模块连接

2.1.3 查询光源控制器模块状态信息

BOOL SY_MVD_Slave_ConnectSts()

说明：查询控制器模块在线状态

输出参数：TRUE：成功； FALSE：失败

输入参数：（无）

2.1.4 更改光源控制器模块 ID 号

BOOL SY_MVD_Light_Set_SlaveIP (int SlaveIP,int NewIP)

说明：改变模块 ID 号

输出参数：TRUE：新端口获取成功； FALSE：新端口获取失败

输入参数：SlaveIP：原来的模块 ID 号，控制器的 ID（1~32）

输入参数：NewIP：更改后的模块 ID 号，控制器的 ID（1~32）

2.1.5 得到当前光源控制器模块 ID 号

BOOL SY_MVD_Light_Get_SlaveIP (int *SlaveIPIndex)

说明：得当前使用的模块 ID 号

输出参数：TRUE：获取成功； FALSE：获取失败

输入参数：SlaveIPIndex：当前使用的模块 ID 号

2.1.6 获得链路上所有连接的模块 ID 号

BOOL SY_MVD_Light_Get_AllSlaveIP (int *AllIP , int *IPcount)

说明：获得链路上所有连接的模块 ID 号。

输出参数：TRUE ： 获取成功； FALSE ： 获取失败

输入参数：AllIP： 总共允许接驳的模块数量号，可以接驳 63 个模块；

输入参数：IPcount： 得到当前连接模块的数量；

2.3 光源控制器的相关函数

2.3.1 设置通道亮度值

BOOL SY_MVD_Light_Set_Intensity(int SlaveIP,int ChNum, int Intesity)

说明：设定光源控制器某个通道的亮度值

输出参数：TRUE ： 设置成功； FALSE ： 失败

输入参数：

SlaveIP： 控制器的 ID（1~32）

ChNum： 控制器的通道号（0~3）

Intesity： 通道的亮度值（0~255）

2.3.2 得到通道亮度值

BOOL SY_MVD_Light_Get_Intensity(int SlaveIP,int ChNum, int *Intesity)

说明：查询光源控制器某个通道的亮度值

输出参数：Intesity： 通道的亮度值（0~255）

输入参数：

SlaveIP： 控制器的 ID（1~32）

ChNum： 控制器的通道号（0~3）

2.3.3 设置通道触发延迟时间

BOOL SY_MVD_Light_Set_TriggerDelay (int SlaveIP,int ChNum , int StrobeTime)

说明：设置触发延迟时间

输出参数：TRUE ： 操作成功； FALSE ： 失败

输入参数：

SlaveIP： 控制器的 ID（1~32）

ChNum： 控制器的通道号（0~3）

StrobeTime： 频闪延迟时间值（1~4096），单位为 5 微秒，即最小触发延迟时间为 5us，最大触发延迟时间为 4096*5=20480 微秒。

2.3.4 得到通道触发延迟时间

BOOL SY_MVD_Light_Get_TriggerDelay(int SlaveIP,int ChNum, int *StrobeTime)

说明：得到触发延迟时间

输出参数：TRUE：操作成功； FALSE：失败

输入参数：

SlaveIP：控制器的 ID（1~32）

ChNum：控制器的通道号（0~3）

StrobeTime：频闪延迟时间值（1~4096），单位为 5 微秒，即最小触发延迟时间为 5us，最大触发延迟时间为 4096*5=20480 微秒。

2.3.5 设置通道频闪保持时间

BOOL SY_MVD_Light_Set_StrobeDuration (int SlaveIP,int ChNum, int HoldTime)

说明：设置频闪保持时间

输出参数：TRUE：操作成功； FALSE：失败

输入参数：

SlaveIP：控制器的 ID（1~32）

ChNum：控制器的通道号（0~3）

HoldTime：保持时间值（1~20），单位为毫秒，即 1ms~20ms。

2.3.6 得到通道频闪保持时间

BOOL SY_MVD_Light_Get_StrobeDuration(int SlaveIP,int ChNum, int *HoldTime)

说明：得到频闪保持时间

输出参数：TRUE：操作成功； FALSE：失败

输入参数：

SlaveIP：控制器的 ID（1~32）

ChNum：控制器的通道号（0~3）

HoldTime：保持时间值（1~20），单位为毫秒，即 1ms~20ms。

2.3.7 设置光源控制模式

BOOL SY_MVD_Light_SetCHMode(int SlaveIP, int ChNum, int Mode)

说明：光源通道模式，Strobe 或者 PWM 模式等

输出参数：TRUE：操作成功； FALSE：失败

输入参数：

SlaveIP：控制器的 ID（1~32）

ChNum：控制器的通道号（0~4）

Mode：模式有 5 种模式

工作模式 0，将该通道设置为软件触发常亮模式。可由按键触发和通信触发。

工作模式 1，将该通道设置为软件触发闪光模式。可由按键触发和通信触发。

工作模式 2，将该通道设置为外部触发闪光模式，上升沿触发。只能由输入端口触发。

工作模式 3，将该通道设置为外部触发闪光模式，下降沿触发。只能由输入端口触发。

工作模式 4，将该通道设置为外部开关量触发模式，输入信号高电平有效，光源常亮；信号无效时候，光源关闭。

2.3.8 得到光源控制模式

BOOL SY_MVD_Light_GetCHMode(int SlaveIP, int ChNum, int *Mode)

说明：得到光源通道模式，Strobe 或者 PWM 模式等

输出参数：TRUE：操作成功； FALSE：失败

输入参数：

SlaveIP：控制器的 ID（1~32）

ChNum：控制器的通道号（0~4）

Mode：模式有 5 种模式

工作模式 0，将该通道设置为软件触发常亮模式。可由按键触发和通信触发。

工作模式 1，将该通道设置为软件触发闪光模式。可由按键触发和通信触发。

工作模式 2，将该通道设置为外部触发闪光模式，上升沿触发。只能由输入端口触发。

工作模式 3，将该通道设置为外部触发闪光模式，下降沿触发。只能由输入端口触发。

工作模式 4，将该通道设置为外部开关量触发模式，输入信号高电平有效，光源常亮；信号无效时候，光源关闭。

2.3.9 设置光源开关状态

BOOL SY_MVD_Light_SetChStsOnOff(int SlaveIP, int ChNum, int CHstatus)

说明：PWM 模式下打开或者关闭某个通道，或者频闪

输出参数：TRUE：设置成功； FALSE：失败

输入参数：

SlaveIP：控制器的 ID（1~32）

ChNum：控制器的通道号（0~3）

CHstatus：2：频闪； 1：打开； 0：关闭

2.3.10 载入配置文件

BOOL SY_MVD_LoadLightParam(int SlaveIP,char *FilePath)

说明：从 ini 文件中读取当前光源控制&COM 参数

输出参数：TRUE ： 操作成功； FALSE ： 失败

输入参数：

SlaveIP: 控制器的 ID (1~32)

FilePath: ini 文件的路径

2.3.11 保持配置文件

BOOL SY_MVD_SaveLightParam(int SlaveIP,char *FilePath)

说明：当前光源控制&COM 参数存入 ini 文件

输出参数：TRUE ： 操作成功； FALSE ： 失败

输入参数：

SlaveIP: 控制器的 ID (1~32)

FilePath: ini 文件的路径

2.3.12 保存到 Flash

BOOL SY_MVD_Light_SaveParamToFlash (int SlaveIP)

说明：将设置的参数存入 Flash

输出参数：TRUE ： 操作成功； FALSE ： 失败

输入参数：

SlaveIP: 控制器的 ID (1~32)

2.3.13 软件切换通道

BOOL SY_MVD_Light_SelChn (int SlaveIP, int ChNumSet)

说明：软件切换模块的通道

输出参数：TRUE ： 操作成功； FALSE ： 失败

输入参数：

SlaveIP: 控制器的 ID (1~32)

ChNumSet: 控制器的通道号 (0~3)

2.3.14 设置通讯模式

BOOL SY_MVD_Light_Set_CommMod (int SlaveIP, int CommMod)

说明：软件切换模块的通讯方式

输出参数：TRUE ：操作成功； FALSE ：失败

输入参数：

SlaveIP：控制器的 ID（1~32）

CommMod：光源控制器模块通讯模式

光源控制器模块通讯模式	描述
0	采用 RS-232 通讯模式
1	采用 RS-485 通讯模式

2.3.15 同步设置 4 通道亮度值

BOOL SY_MVD_Light_Set_AllIntensity(int SlaveIP,int CH1Intensity,int CH2Intensity,int CH3Intensity,int CH4Intensity);

说明：同步设定 4 个通道的 PWM 值

输出参数：TRUE ：操作成功； FALSE ：失败

输入参数：

SlaveIP: 控制器的 ID (1~32)

CH1Intensity: 通道 1 的亮度值 (0~255)

CH2Intensity: 通道 2 的亮度值 (0~255)

CH3Intensity: 通道 3 的亮度值 (0~255)

CH4Intensity: 通道 4 的亮度值 (0~255)

2.3.16 同步设置 4 通道开关

BOOL SY_MVD_Light_Set_AllChStsOnOff(int SlaveIP,int CHstatus);

说明：同步设定打开或者关闭 4 个通道

输出参数：TRUE ：操作成功； FALSE ：失败

输入参数：

SlaveIP: 控制器的 ID (1~32)

CHstatus:

光源控制器所有通道状态 (CHstatus)	描述
0	所有通道关闭
1	所有通道打开

2.3.17 同步通道开关状态

BOOL SY_MVD_Light_Get_AllChStsOnOff(int SlaveID,int *pstatus);

说明：获得 4 个通道的开关状态

输出参数：TRUE ： 操作成功； FALSE ： 失败

输入参数：

SlaveID: 控制器的 ID (1~32)

pstatus: 通道开关状态

pstatus 各个 bit 定义如下所示

位	定义	描述
0	通道 1 状态	值为 1 时候表示通道 1 状态是打开的；值为 0 时候表示通道 1 状态是关闭的
1	通道 2 状态	值为 1 时候表示通道 2 状态是打开的；值为 0 时候表示通道 2 状态是关闭的
2	通道 3 状态	值为 1 时候表示通道 3 状态是打开的；值为 0 时候表示通道 3 状态是关闭的
3	通道 4 状态	值为 1 时候表示通道 4 状态是打开的；值为 0 时候表示通道 4 状态是关闭的

2.3.18 设置通讯波特率

BOOL SY_MVD_Light_Set_Baudrate(int SlaveID,int Baudrate_type);

说明：设置通讯波特率

输出参数：TRUE ： 操作成功； FALSE ： 失败

输入参数：

SlaveID: 控制器的 ID (1~32)

Baudrate_type: 波特率速率

Baudrate_type(波特率)	描述
0	2400bps
1	9600bps
2	38400bps
3	115200bps

2.3.19 设置通道 PWM 和通道状态

BOOL SY_MVD_Light_Set_Intensity_ChanelSts(int SlaveIP,int CHSts,int CH1Intesity,int CH2Intesity,int CH3Intesity,int CH4Intesity);

说明：独立控制 4 个通道 PWM 值和 4 个通道开关命令

输出参数：TRUE：操作成功； FALSE：失败

输入参数：

SlaveIP：控制器的 ID（1~32）

CHSts:分别控制 4 个通道的开关状态。如 CHSts =0X01，则表示通道 0 打开，其余通道关闭；如 CHSts =0X05,则表示通道 0 和通道 2 同时打开，其余通道关闭。

CH1Intesity:通道 1 的 PWM 值

CH2Intesity:通道 2 的 PWM 值

CH3Intesity:通道 3 的 PWM 值

CH4Intesity:通道 4 的 PWM 值

2.3.20 设置通道电流输出值

BOOL SY_MVD_Light_Set_CurrentOut(int SlaveIP,int ChNum, int CurrentOut)

说明：设定光源控制器某个通道的电流输出值，恒流源光源控制器专用。

输出参数：TRUE：设置成功； FALSE：失败

输入参数：

SlaveIP：控制器的 ID（1~32）

ChNum：控制器的通道号（0~3）

CurrentOut：通道的电流输出值，根据恒流源的光源控制器实际输出来设定。

2.3.21 得到通道电流输出值

BOOL SY_MVD_Light_Get_CurrentOut (int SlaveIP,int ChNum, int * CurrentOut)

说明：查询光源控制器某个通道的电流输出值，恒流源光源控制器专用。

输出参数：CurrentOut：通道的电流输出值。

输入参数：

SlaveIP：控制器的 ID（1~32）

ChNum：控制器的通道号（0~3）

2.3.22 同时设置 4 通道电流输出值

BOOL SY_MVD_Light_Set_AllCurrentOut(int SlaveIP,int CH1CurrentOut,int CH2CurrentOut,int CH3CurrentOut,int CH4CurrentOut);

说明：独立控制 4 个通道电流输出值和 4 个通道开关命令

输出参数：TRUE：操作成功； FALSE：失败

输入参数：

SlaveIP: 控制器的 ID (1~32)

CH1CurrentOut:通道 1 的电流输出值

CH2CurrentOut:通道 2 的电流输出值

CH3CurrentOut:通道 3 的电流输出值

CH4CurrentOut:通道 4 的电流输出值

2.3.23 同时设置 4 通道电流输出值和通道状态

BOOL SY_MVD_Light_Set_AllCurrentOut(int SlaveIP,int CH1CurrentOut,int CH2CurrentOut,int CH3CurrentOut,int CH4CurrentOut);

说明：独立控制 4 个通道电流输出值和 4 个通道开关命令

输出参数：TRUE：操作成功； FALSE：失败

输入参数：

SlaveIP: 控制器的 ID (1~32)

CHSts:分别控制 4 个通道的开关状态。如 CHSts =0X01, 则表示通道 0 打开, 其余通道关闭; 如 CHSts =0X05,则表示通道 0 和通道 2 同时打开, 其余通道关闭。

CH1CurrentOut:通道 1 的电流输出值

CH2CurrentOut:通道 2 的电流输出值

CH3CurrentOut:通道 3 的电流输出值

CH4CurrentOut:通道 4 的电流输出值

修订记录

Rev	Data	Author	Description
1.0	20150121	Shuangyi	初稿
1.1	20150206	Shuangyi	增加 SY_MVD_Light_Set_Current 函数
1.2	20150315	Shuangyi	增加 SY_MVD_Light_SetCHMode 函数 增加 SY_MVD_Light_Set_TriggerDelay 增加 SY_MVD_Light_Set_StrobeDuration
1.2	20150530	Shuangyi	增加 SY_MVD_Light_Get_AllSlaveIP 查询连接的模块 ID
1.3	20150601	Shuangyi	增加 SY_MVD_Light_SelChn, 软件切换模式
1.4	20150701	Shuangyi	增加光源频闪电流峰值函数 SY_MVD_Light_Set_PeakCurrent SY_MVD_Light_Get_PeakCurrent
1.5	20150721	Shuangyi	函数更名为: SY_MVD_Light_xxxxxxx
1.5	20150722	Shuangyi	修复 SY_MVD_Light_SetCHMode 设置模式为 3 的异常状况
1.6	20160622	Shuangyi	增加设置和查询通讯模式函数 SY_MVD_Light_Set_CommMod SY_MVD_Light_Get_CommMod
1.7	20160623	Shuangyi	增加设置和查询设置输出电压函数 SY_MVD_Light_Set_OutVolt SY_MVD_Light_Get_OutVolt
1.8	20170811	Shuangyi	增加设置开关量触发模式(模式 4) SY_MVD_Light_SetCHMode
1.9	20171207	Shuangyi	*增加一条指令同时设置 4 个同道的 PWM 值 SY_MVD_Light_Set_AllIntensity *增加一条指令同时控制 4 个通道的开关状态 SY_MVD_Light_Set_AllChStsOnOff *增加一条读取 4 个通道开关状态的命令 SY_MVD_Light_Get_AllChStsOnOff *修改波特率设置命中波特率的值 SY_MVD_Light_Set_Baudrate *删除不使用的命令
2.0	20171216	Shuangyi	增加编程流程图
3.0	20180317	Shuangyi	将触发延迟和频闪保持时间设定值先定为 1ms~20ms
3.1	20180427	Shuangyi	独立控制 4 个通道 PWM 值和 4 个通道开关命令
3.2	20180730	Shuangyi	增加电流输出函数 SY_MVD_Light_Set_CurrentOut SY_MVD_Light_Get_CurrentOut
3.3	20180801	Shuangyi	增加一条同时设置 4 个通道恒流电流值的命令。 增加了同时设置 4 通道电流和开关的命令