OPT 光源控制器编程手册

版本号 V1.0.14

版本维护记录

版本号	时间	备注
1.0.0	2014-09-17	由奥普特自动化科技有限公司开发、维护
1.0.1	2014-11-14	由奥普特自动化科技有限公司开发、维护
1.0.5	2014-11-19	由奥普特自动化科技有限公司开发、维护
1.0.6	2014-12-03	修订了一定的程序错误、漏洞（详情见 ReadMe.txt 文件）
1.0.7	2015-03-07	增加了 OPTController ConnectionResetBySN 函数
1.0.8	2015-03-09	增加了 OPTController_ConnectionResetByIP／OPTController_SetEtheConnectionHeartBeat 函数
1.0.9	2015-03-20	改善了心跳机制，解决了拔开网线会卡死的问题
1.0.10	2015-04-09	修复了修改 IP 的功能以解决一些情况下修改 IP 失败的问题
1.0.12	2016-10-13	修复了一定的程序错误、漏洞；增加了搜索以太网在线设备功能、软件触发功能、keepalive 功能等
1.0.13	2017-07-03	增加了 OPTController_ReadCurrentStepIndex, OPTController_ResetCurrentStepIndex 等功能
1.0.14	2018-08-05	添加了 OPTController SetOutPutVoltage,OPTController ReadOutputVoltage,OPTController SetTimeUnit,OPTController ReadTimeUnit 等功能；修改了 OPTController_ReadMaxCurrent 函数参数，OPTController_SetBootProtection 改为 OPTController_SetBootState

1. 简介概述

本手册为 OPT 光源控制器（V3.0 以上版本通讯协议）编程说明文档。该光源控制器同时支持串口和网口通讯（我们推荐使用后种通讯方式）。

1.1. 编程配置

控制器设有一个默认的 IP 地址：192.168.1.16，且该地址可有路由器动态分配。当被控光源设备的 IP 地址与控制器的 IP 地址不在同一个网段时，也就是 IP 地址不是 192.168.1.X 的形式，例如 192.168.24.X（X 为 0-255 的任意整数），需要对控制器 IP 地址做相应的改变（例如：192.168.24.X1）。若使用了不支持 DHCP 服务的交换机（交换机不能动态改变 IP 地址），我们在示例程序中集成了相应的改变 IP 地址的工具。

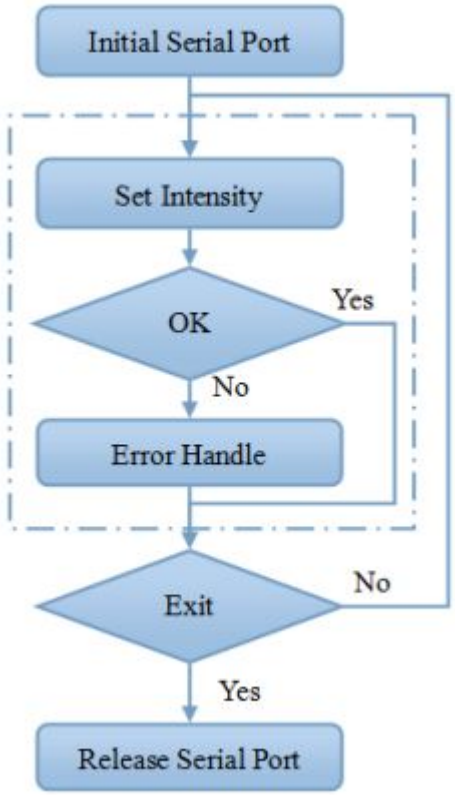
在使用控制器时，需要注意一下事项：

1) 一个控制系统中只能有一个控制器。
2) 需防止 IP 地址冲突。一个设备（包括控制器和被控光源设备）一个 IP 地址。否则控制器与被控光源设备的连接将无法建立。
3) 目前控制器不支持无线连接。

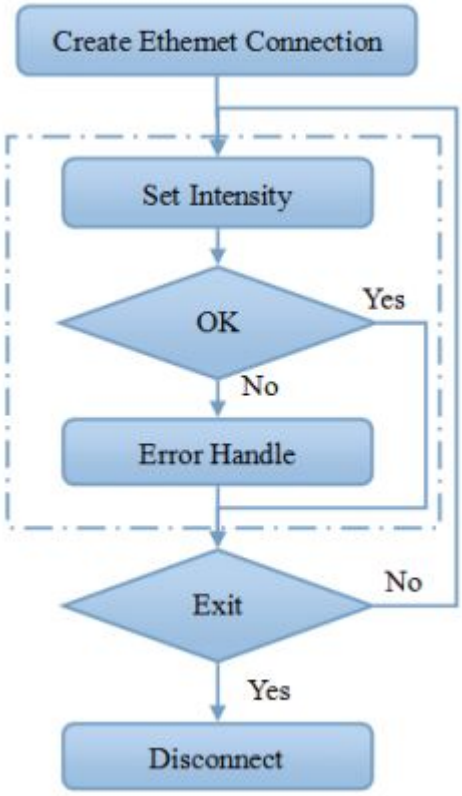
1.2. 控制器默认设置

1. 波特率：9600
2. 控制器 IP 地址是动态分配的
3. 通讯指令中无校验字
4. 支持掉电备份
5. 支持通讯中带返回值

1.3. 编辑流程



(a) 串口编程流程图实例



(b) 网口编程流程图实例

图（a）和图（b）中，两类通讯模式编程流程图（此处我们简单的以设置光源亮度为例）。所有在虚线框中的步骤均由功能函数实现。

1.4. 示例程序

我们建议在两个连续的读写操作之间给出 20ms 的时间间隔以便控制器对指令做出相应的处理反应。

1.4.1. C#示例程序

```
using System ;
using System . Collections . Generic ;
using System . Linq ;
using System . Text ;
namespace OPTController
{
    Class Program
    {
        static private int IntensityValue = 0;
```

```
static void Main(string [] args)
{
    OPTControllerAPI OptController = new OPTControllerAPI();
    do
    {
        // OptController.InitSerialPort("COM1")
        //create an Ethernet connection by IP address, e.g. "192.168.1.16"
        If(0 != OptController.CreateEthernetConnectionByIP("192.168.1.16"))
        {
            Console.WriteLine("Connection failed");
            Break;
        }

        //Set the intensity 255 to all channels
        If(0 != OptController.SetIntensity(0,   255))
        {
            Console.WriteLine("Failed to set intensity for the all channel");
            Break;
        }

        //Set the intensity 255 to the 1st channels
        If(0 != OptController.SetIntensity(1,   255))
        {
            Console.WriteLine("Failed to set intensity for the 1st channel");
            Break;
        }

        //Read the intensity of the 1st channel
        If(0 != OptController.ReadIntensity(1 , ref(IntensityValue )))
        {
            Console.WriteLine(IntensityValue);
            Console.WriteLine("Failed to read intensity of the 1st channel");
            Break;
        }

        //turn off all channels
        If(0 != OptController.TurnOffChannel(0))
        {
            Console.WriteLine("Failed to turn off the all channel");
            Break;
        }
```

```
                    //turn on all channels
                    If(0 != OptController.TurnOnChannel(0))
                    {
                        Console.WriteLine("Failed to turn on the all channel");
                        Break;
                    }
                }while(false);
                // destroy the connection
                int    ret = 0;
                ret = OptController.DestroyEthernetConnect();
                if(0 != ret)
                {
                    Console.WriteLine("Failed to destroy the connection");
                }
                else
                {
                    Console.WriteLine("DONE");
                }
                Console.ReadKey();
            }
        }
    }
```

注意：请勾选"Allow unsafe code" (ProjectName — Properties — Build)。

1.4.2. VC++ 示例程序

```
// connect to controller
OPTController_InitSerailPort(W2A(strCOMName . GetBuffer ( 0 )),
&m_OPTControllerHandle) ;
//OPTController_CreateEthernetConnectionBySN(W2A(strCOMName.GetBuffer(0)),
&m_OPTControllerHandle) ;
// OPTController_DestroyEthernetConnection ( m_OPTControllerHanlde ) ;

// turn on the 1st channel
OPTController_TurnOnChannel ( m OPTControllerHanlde , 1) ;
// turn off the 1st channel
OPTController_TurnOffChannel ( m OPTControllerHanlde , 1) ;

// Set the intensity 255 to the 3 rd channel
OPTController_SetIntensity ( m_OPTControllerHanlde , 3 , 255);

// destroy the connection with the controller
// OPTController_DestroyEthernetConnection ( m_OPTControllerHanlde ) ;
```

```
OPTController_RealeaseSerialPort(m_OPTControllerHanlde);
```

1.4.3. VB 示例程序

```
‘Create a connection to the controller
Dim IPAddress As String
IPAddress = ” 1 9 2 . 1 6 8 . 1 8 . 2 0 ”
Dim controllerHandle As Integer
OPTControllerAPI . OPTController_CreateEthernetConnectionByIP ( IPAddress , controllerHandle)

‘Turn on/ off the 1st channel
OPTControllerAPI . OPTController_TurnOnChannel ( controllerHandle , 1 )
OPTControllerAPI . OPTController_TurnOffChannel (controllerHandle , 1 )

‘Set intensity 255 to the 1st channel
OPTControllerAPI . OPTController_SetIntensity (controllerHandle, 1 , 255 )

‘Read the intensity of the 1st channel ( channel range 1 to 16) . Before you read the intensity , you need to delay
Dim nIntensity As Integer
Threading . Thread . Sleep (100)
OPTControllerAPI . OPTController_ReadIntensity (controllerHandle , 1 , nIntensity )

‘Disconnect the controller
OPTControllerAPI . OPTController_DestroyEthernetConnection ( controllerHandle )
```

2. 函数说明

2.1. 打开一个串口

函数名： long OPTController_InitSerialPort(char* comName, OPTController_Handle *controllerHandle);

函数描述：打开一个可用的串口。

输入参数：

comName --串口名称，例如：“COM1”。

输出参数：

controllerHandle --控制器句柄。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_INITSERIAL_FAILED 或 OPT_ERR_SERIALPORT_UNOPENED（见错误码表 1）。

参见：释放串口。

2.2. 释放一个串口

函数名：long OPTController_ ReleaseSerialPort (OPTController_ Handle controllerHandle);

函数描述：释放一个串口。

输入参数：

controllerHandle --控制器句柄。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_RELEASESERIALPORT_FAILED （见错误码表 1）。

参见：打开串口。

2.3. 创建一个网口连接（通过 IP 地址）

函数名： long OPTController_CreateEthernetConnectionByIP(char *serverIPAddress, OPTController_Handle *controllerHandle);

函数描述：通过 IP 地址创建一个网口连接。

输入参数：

serverIPAddress --被控光源设备 IP 地址，可设为：“192.168.1.16”。

输出参数：

controllerHandle --控制器句柄。

返回值：

- 操作成功：OPT_SUCCEED；

- 操作失败：OPT_ERR_ CREATEETHECON _FAILED （见错误码表 1）。

备注：将控制器作为客户端连接被控光源设备。连接前请检查并保证控制器已接入局域网。

参见：断开网口连接。

2.4. 创建一个网口连接（通过 SN 序列号）

函数名：long OPTController_CreateEthernetConnectionBySN(char *serialNumber, OPTController_Handle *controllerHandle);

函数描述：通过 SN 序列号创建一个网口连接。

输入参数：

serialNumber --被控光源的序列号，如“AA53017016”。

输出参数：

controllerHandle --控制器句柄。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_ CREATEETHECON _FAILED （见错误码表 1）。

备注：

- 将控制器作为客户端连接被控光源设备。连接前请检查并保证控制器已接入局域网；
- 我们推荐通过 SN 序列号创建一个网口连接，因为 IP 地址易于动态改变。我们在开发包中提供了工具 SearchForControllers.exe 来检测序列号。

参见：断开网口连接。

2.5. 断开网口连接

函数名：long OPTController_DestroyEthernetConnection(OPTController_Handle controllerHandle);

函数描述：断开一个存在的网口连接。

输入参数：

controllerHandle --控制器句柄。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_ DESTROYETHECON_FAILED （见错误码表 1）。

参见：建立网口连接。

2.6. 打开通道

函数名：long OPTController_TurnOnChannel(OPTController_Handle controllerHandle, int channelIndex);

函数描述：打开一个或所有通道。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --需打开的通道的序号，通道序号取值范围：[0 – 16]（十进制形式，0 代表所有通道，1-16 代表对应通道的通道序号）。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_TURNONCH_FAILED 或 OPT_ERR_CHINDEX_OUTRANGE（见错误码表 1）。

参见：关闭通道。

2.7. 打开多个通道

函数名：long OPTController_TurnOnMultiChannel(OPTController_Handle controllerHandle, int* channelIndexArray, int length);

函数描述：打开多个指定的通道。

输入参数：

- controllerHandle --控制器句柄；
- channelIndexArray --一个保存了所有待打开通道序号的数组，通道序号取值范围：[1 – 16]（十进制形式，1-16 代表对应通道的通道序号）；
- length --序号数组长度。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_TURNONCH_FAILED 或 OPT_ERR_CHINDEX_OUTRANGE（见错误码表 1）。

参见：关闭多个通道。

2.8. 关闭通道

函数名： long OPTController_TurnOffChannel(OPTController_Handle controllerHandle, int channelIndex);

函数描述：关闭一个或所有通道。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --需关闭的通道的序号，通道序号取值范围：[0 – 16]（十进制形式，0 代表所有通道，1-16 代表对应通道的通道序号）。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_TURNOFFCH_FAILED 或 OPT_ERR_CHINDEX_OUTRANGE （见错误码表 1）。

参见：打开通道。

2.9. 关闭多个通道

函数名： long OPTController_TurnOffMultiChannel(OPTController_Handle controllerHandle, int* channelIndexArray, int length);

函数描述：关闭多个指定的通道。

输入参数：

- controllerHandle --控制器句柄；

- channelIndexArray --一个保存了所有待关闭通道序号的数组，通道序号取值范围：[1 – 16]（十进制形式，1-16 代表对应通道的通道序号）；
- length --序号数组长度。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_TURNOFFCH_FAILED 或 OPT_ERR_CHINDEX_OUTRANGE （见错误码表 1）。

参见：打开多个通道。

2.10. 设置亮度

函数名： long OPTController_SetIntensity(OPTController_Handle controllerHandle, int channelIndex, int intensity);

函数描述：为指定的某个通道或所有通道设置亮度。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --需设置亮度的通道的序号，通道序号取值范围：[0 – 16]（十进制形式，0 代表所有通道，1-16 代表对应通道的通道序号）；
- intensity --所设亮度值，亮度值范围：[0 – 255]（十进制）。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_INTENSITY_FAILED，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE （见错误码表 1）。

参见：读取亮度。

2.11. 设置多个亮度

函数名： long OPTController_SetMultiIntensity(OPTController_Handle controllerHandle, IntensityItem* intensityArray, int length);

函数描述：为指定的某一个通道或多个通道设置亮度。

输入参数：

- controllerHandle --控制器句柄；
- intensityArray --一个包含了多个亮度值和对应通道序号的数组，亮度值取值范围：[0 – 255]（十进制）；
- length --亮度数组的长度。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_INTENSITY_FAILED 或 OPT_ERR_PARAM_OUTRANGE （见错误码表 1）。

参见：读取亮度。

2.12. 读取亮度

函 数 名： long OPTController_ReadIntensity(OPTController_Handle controllerHandle, int channelIndex, int *intensity);
函数描述：读取指定通道的亮度。
输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[1 – 16]（十进制，1-16 代表对应通道的通道序号）。

输出参数：intensity --读取到的亮度值。
返回值：

- 操作成功：OPT_SUCCEED；
- 操 作 失 败 ： OPT_ERR_READ_INTENSITY_FAILED ， OPT_ERR_CHINDEX_OUTRANGE （见错误码表 1）。

参见：设置亮度和设置多个亮度。

2.13. 设置触发脉宽

函 数 名： long OPTController_SetTriggerWidth(OPTController_Handle controllerHandle, int channelIndex, int triggerWidth);
函数描述：为一个或所有通道设置触发脉宽。
输入参数：

- controllerHandle --控制器句柄；
- channelIndex --需设置触发脉宽的通道的序号，通道序号取值范围：[0 – 16]（十进制形式，0 代表所有通道，1-16 代表对应通道的通道序号）；
- triggerWidth --触发脉宽值，触发脉宽取值范围：[1 – 1023]，单位是：1ms，详细范围请参见说明书。

返回值：

- 操作成功：OPT_SUCCEED；
- 操 作 失 败 ： OPT_ERR_SET_TRIGGERWIDTH_FAILED ， OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE （见错误码表 1）。

参见：读取触发脉宽。

2.14. 设置多个触发脉宽

函 数 名： long OPTController_SetMultiTriggerWidth(OPTController_Handle controllerHandle, TriggerWidthItem* triggerWidthArray, int length);
函数描述：为多个通道设置触发脉宽。
输入参数：

- controllerHandle --控制器句柄；
- triggerWidthArray --一个包含了多个触发脉宽和对应通道序号的数组，触发脉宽取值范围：[1 – 1023](十进制，单位是：1ms)，详细范围请参见说明书；

- length --触发脉宽值数组长度。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_TRIGGERWIDTH_FAILED，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE （见错误码表 1）。

参见：读取触发脉宽。

2.15. 读取触发脉宽

函数名： long OPTController_ReadTriggerWidth(OPTController_Handle controllerHandle,int channelIndex, int* triggerWidth);

函数描述：读取指定通道的触发脉宽。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[1 – 16]（十进制，1-16 代表对应通道的通道序号）。

输出参数：triggerWidth --读取到的触发脉宽值。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_TRIGGERWIDTH_FAILED，OPT_ERR_CHINDEX_OUTRANGE （见错误码表 1）。

参见：设置触发脉宽和设置多个触发脉宽。

2.16. 设置高亮触发脉宽

函数名： long OPTController_ SetHBTriggerWidth(OPTController_Handle controllerHandle,int channelIndex, int HBTriggerWidth);

函数描述：为一个或所有通道设置高亮触发脉宽。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --需设置高亮触发脉宽的通道的序号，通道序号取值范围：[0 – 16]（十进制形式，0 代表所有通道，1-16 代表对应通道的通道序号）；
- HBTriggerWidth --高亮触发脉宽值，取值范围：[1 – 500]，单位是：0.01ms。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_HBTRIGGERWIDTH_FAILED，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE（见错误码表 1）。

参见：读取高亮触发脉宽。

2.17. 设置多个高亮触发脉宽

函数名：long OPTController_SetMultiHBTriggerWidth(OPTController_Handle controllerHandle, HBTriggerWidthItem* HBtriggerWidthArray, int length);

函数描述：为多个通道设置触发脉宽。

输入参数：

- controllerHandle --控制器句柄；
- HBtriggerWidthArray --一个包含了多个高亮触发脉宽和对应通道序号的数组，触发脉宽取值范围：[1 – 500](十进制，单位是：0.01ms)；
- length --高亮触发脉宽值数组长度。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_ HBTRIGGERWIDTH _FAILED, OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE （见错误码表 1）。

参见：读取高亮触发脉宽。

2.18. 读取高亮触发脉宽

函数名：long OPTController_ReadHBTriggerWidth(OPTController_Handle controllerHandle,int channelIndex,int* HBTriggerWidth);

函数描述：读取指定通道的触发脉宽。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[1 – 16]（十进制，1-16 代表对应通道的通道序号）。

输出参数：HBTriggerWidth --读取到的高亮触发脉宽值。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_TRIGGERWIDTH_FAILED，OPT_ERR_CHINDEX_OUTRANGE （见错误码表 1）。

参见：设置触发脉宽和设置多个触发脉宽。

2.19. 返回值设置

函数名：long OPTController_EnableResponse(OPTController_Handle controllerHandle, bool isResponse);

函数描述：设置是否需要返回值。

输入参数：

- controllerHandle --控制器句柄；
- isResponse --"true”表示设置返回值，“false”表示无返回值。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_UNKOWN （见错误码表 1）。

2.20. 校验字设置

函数名：long OPTController_EnableCheckSum(OPTController_Handle controllerHandle, bool isCheckSum);

函数描述：设置是否需要校验字。

输入参数：

- controllerHandle --控制器句柄；
- isCheckSum --"true"表示设置校验字，“false”表示无校验字。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_UNKOWN （见错误码表 1）。

2.21. 设置断电备份

函数名：long OPTController_EnablePowerOffBackup(OPTController_Handle controllerHandle, bool isSave);

函数描述：设置是否打开断电备份功能。

输入参数：

- controllerHandle --控制器句柄；
- isSave --"true"表示断电备份，“false”表示不备份。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_UNKOWN （见错误码表 1）。

2.22. 读取序列号

函数名：long OPTController_ReadSN(OPTController_Handle controllerHandle, char *SN);

函数描述：读取控制器序列号。

输入参数：controllerHandle --控制器句柄。

输出参数：SN --读取到的序列号。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_SN_FAILED （见错误码表 1）。

2.23. 配置网口 IP 信息

函数名：long OPTController_SetIPConfiguration(OPTController_Handle controllerHandle, char *IP, char *subnetMask, char *defaultGateway);

函数描述：配置网口 IP 信息。

输入参数：

- controllerHandle --控制器句柄；
- IP --配置网口 IP 地址；
- subnetMask --配置网口子网掩码；
- defaultGateway --配置网口默认网关。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败： OPT_ERR_SET_IPCONFIG_FAILED（见错误码表 1）。

2.24. 读取 IP 配置信息

函数名： long OPTController_ReadIPConfig(OPTController_Handle controllerHandle, char *IP, char *subnetMask, char *defaultGateway);
函数描述：读取控制器 IP 配置信息，仅对以太网控制器有效。
输入参数：controllerHandle --控制器句柄。
输出参数：

- IP --读取到的 IP 地址；
- subnetMask --读取到的子网掩码；
- defaultGateway --读取到的默认网关。

返回值：

- 操作成功：OPT_SUCCEED;
- 操作失败：OPT_ERR_READ_IPCONFIG_FAILED （见错误码表 1）。

2.25. 设置最大电流

函数名： long OPTController_SetMaxCurrent (OPTController_Handle controllerHandle, int channelIndex, int current);
函数描述：为指定的通道设置最大电流。
输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0 – 16]（十进制，0 代表所有通道，1-16 代表对应通道的通道序号）；
- current --所设最大电流，最大电流取值范围：[1-200](十进制，单位：10mA)。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_MAXCURRENT_FAILED （见错误码表 1）。

参见：读取最大电流。

2.26. 设置多个最大电流

函数名： long OPTController_SetMultiMaxCurrent(OPTController_Handle controllerHandle, MaxCurrentItem *maxCurrentArray, int length);
函数描述：为指定的一个或者多个通道设置最大电流。
输入参数：

- controllerHandle --控制器句柄；
- maxCurrentArray --一个包含了多个最大电流值和对应通道序号的数组，最大电流取值范围：[1-200]（十进制，单位：10mA）；
- length --最大电流数组长度。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_MAXCURRENT_FAILED （见错误码表 1）。

参见：读取最大电流。

2.27. 读取最大电流

函数名： long OPTController_ReadMaxCurrent (OPTController_Handle controllerHandle, int channelIndex, int mode,int *value);

函数描述：读取指定的通道最大电流。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[1– 16]（十进制，1-16 代表对应通道的通道序号）；
- mode -- 读取电流模式，读取电流模式范围[0- 2]，0：读取手动 设置的当前电流值；1：读取当前电流值；2：读取电压值。

输出参数：value --读取到的值。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_MAXCURRENT_FAILED （见错误码表 1）。

参见：设置最大电流和设置多个最大电流。

2.28. 设置输出端电压

函数名： long OPTController_SetOutputVoltage(OPTController_Handle controllerHandle, int channelIndex,int voltage);

函数描述：设置指定通道的输出端电压。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0 -4]（十进制，0 代表所有通道）；
- voltage --设置的电压值。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_VOLTAGE_FAILED （见错误码表 1）。

参见：读取输出端电压。

2.29. 读取输出端电压

函数名： long OPTController_ReadOutputVoltage(OPTController_Handle controllerHandle, int channelIndex,int *voltage);

函数描述：读取指定通道的输出端电压。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[1-4]（十进制）。

输出参数：

- voltage --读取的电压值。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_VOLTAGE_FAILED （见错误码表 1）。

参见：设置输出端电压。

2.30. 读取 MAC 地址

函数名：long OPTController_ReadMAC(OPTController_Handle controllerHandle, char *MAC);
函数描述：读取控制器的 MAC 地址。
输入参数：controllerHandle --控制器句柄。
输出参数：MAC --读取到的 MAC 地址。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_ READ_MAC _FAILED （见错误码表 1）。

2.31. 设置控制器触发极性

函数名： long OPTController_SetTriggerActivation(OPTController_Handle controllerHandle,int channelIndex, int triggerActivation);
函数描述：设置控制器触发极性。
输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0 – 16]（十进制，0 代表所有通道，1-16 代表对应通道的通道序号）；
- triggerActivation --所设控制器极性的代号，取值范围：[0-3]（0 代表跟随正触发，1 代表跟随负触发，2 代表下降沿触发，3 代表上升沿触发），网口控制器器只支持上升沿触发和下降沿触发两种触发模式，默认是上升沿触发。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_TRIGGERACTIVATION_FAILED （见错误码表 1）。

参见：读取控制器触发极性。

2.32. 读取控制器触发极性

函数名： long OPTController_ReadTriggerActivation(OPTController_Handle controllerHandle, int channelIndex, int *triggerActivation);
函数描述：读取控制器触发极性。
输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[1 – 16]（十进制，1-16 代表对应通道的通道序号）。

输出参数：triggerActivation --读取到的控制器极性的代号，0 代表跟随正触发，1 代表跟随负触发，2 代表下降沿触发，3 代表上升沿触发。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_TRIGGERACTIVATION_FAILED （见错误码表 1）。

参见：设置控制器触发极性。

2.33. 设置控制器工作模式

函 数 名 ： long OPTController_SetWorkMode(OPTController_Handle controllerHandle, int workMode);

函数描述：设置控制器工作模式。

输入参数：

- controllerHandle --控制器句柄；
- workMode --所设控制器极性的代号，取值范围：[0-3]（0 代表常亮模式，1 代表常用触发模式，2 代表高亮触发模式，3 代表切换为硬件工作模式设置）。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_WORKMODE_FAILED （见错误码表 1）。

参见：读取控制器工作模式。

2.34. 读取控制器工作模式

函 数 名 ： long OPTController_ReadWorkMode(OPTController_Handle ControlHandle, int *workMode);

函数描述：读取控制器工作模式。

输入参数：controllerHandle --控制器句柄；

输出参数：workMode --所设控制器极性的代号，取值范围：[0-3]（0 代表常亮模式，1 代表常用触发模式，2 代表高亮触发模式，3 代表切换为硬件工作模式设置）。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_WORKMODE_FAILED （见错误码表 1）。

参见：设置控制器工作模式。

2.35. 设置外部触发频率上限值

函 数 名 ： long OPTController_SetOuterTriggerFrequencyUpperBound(OPTController_Handle controllerHandle, int channelIndex,int maxFrequency);

函数描述：设置控制器外部触发频率上限值，只针对频闪方式控制器有效。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0 – 16]（十进制，0 代表所有通道，1-16 代表对应通道的通道序号）；
- maxFrequency --设置外部频率上限值，范围：[1-900]，单位：1Hz。

返回值：

- 操作成功：OPT_SUCCEED；

- 操作失败：OPT_ERR_SET_OUTERTRIGGERFREQUENCYUPPERBOUND_FAILED（见错误码表 1）。

2.36. 读取外部触发频率上限值

函数名： long OPTController_ReadOuterTriggerFrequencyUpperBound(OPTController_Handle controllerHandle, int channelIndex, int *maxFrequency);
函数描述：读取控制器外部触发频率上限值，只针对频闪方式控制器有效。
输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[1 – 16]（十进制，1-16 代表对应通道的通道序号）。

输出参数：maxFrequency --读取到的外部频率上限值。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_OUTERTRIGGERFREQUENCYUPPERBOUND_FAILED（见错误码表 1）。

2.37. 自动检测一次负载电流

函数名：long OPTController_AutoDetectLoadOnce(OPTController_Handle controllerHandle);
函数描述：自动检测一次控制器负载电流。
输入参数：controllerHandle --控制器句柄。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_WRITE_FAILED（见错误码表 1）。

2.38. 设置自动频闪频率

函数名：long OPTController_SetAutoStrobeFrequency(OPTController_Handle controllerHandle, int channelIndex, int frequency);
函数描述：设置控制器自动频闪频率，只针对频闪方式控制器有效。
输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0 – 16]（十进制，0 代表所有通道，1-16 代表对应通道的通道序号）；
- frequency --设置自动频闪频率值，范围：[15-1000]，单位：1Hz。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_AUTOSTROBEFREQUENCY_FAILED （见错误码表 1）。

2.39. 读取自动频闪频率

函 数 名 ： long OPTController_ReadAutoStrobeFrequency(OPTController_Handle controllerHandle, int channelIndex, int *frequency);
函数描述：读取自动频闪频率，只针对频闪方式控制器有效。
输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[1 – 16]（十进制， 1-16 代表对应通道的通道序号）。

输出参数：frequency --设置自动频闪频率值，范围：[15-1000]，单位：1Hz。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_AUTOSTROBEFREQUENCY_FAILED （见错误码表 1）。

2.40. 动/静态 IP 切换

函 数 名 ： long OPTController_EnableDHCP(OPTController_Handle controllerHandle, BOOL bDHCP);
函数描述：是否启动 DHCP 协议，启动为动态 IP，关闭为静态 IP。
输入参数：

- controllerHandle --控制器句柄；
- bDHCP --“TRUE”表示动态 IP，“FALSE”表示静态 IP。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_DHCP_FAILED （见错误码表 1）。

2.41. 设置负载模式

函 数 名 ： long OPTController_SetLoadMode(OPTController_Handle controllerHandle, int channelIndex, int loadMode);
函数描述：设置负载模式。
输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0 – 16]（十进制，0 代表所有通道， 1-16 代表对应通道的通道序号）；
- loadMode --负载模式代号，0 为自动检测负载大小模式，1 为手动设置最大电流模式。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_LOADMODE_FAILED (见错误码表 1)。

2.42. 读取控制器属性

函 数 名 ： long OPTController_ReadProperties(OPTController_Handle controllerHandle, int property, char *value);
函数描述：读取控制器的属性。
输入参数：

- controllerHandle --控制器句柄；
- property --所需读取控制器属性代号。1 为控制器型号，2 位控制器固件版本。

输出参数：value --读取到的属性。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_PROPERTY_FAILED （见错误码表 1）。

2.43. 读取控制 DLL 版本号

函数名：long OPTController_GetVersion(char *version);
函数描述：读取控制器 DLL 版本号。
输出参数：version --读取到的版本号。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_UNKNOWN （见错误码表 1）。

2.44. 通过序列号重置控制器连接

函数名：long OPTController_ConnectionResetBySN(char *serialNumber);
函数描述：通过 UDP 的方式重置控制器连接，使序列号为 serialNumber 的控制器断开连接。
输入参数：serialNumber --需重置的控制器的序列号。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_CONNECTION_RESET_FAILED （见错误码表 1）。

备注：在现有控制器固件版本条件下，重置操作需耗时 150ms。
参见：通过 IP 地址重置控制器连接。

2.45. 通过 IP 地址重置控制器连接

函数名：long OPTController_ConnectionResetByIP(char *serverIPAddress);
函数描述：通过 UDP 的方式重置控制器连接，使 IP 地址为 serverIPAddress 的控制器断开连接。
输入参数：serverIPAddress --需重置的控制器的 IP 地址。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_CONNECTION_RESET_FAILED （见错误码表 1）。

参见：通过序列号重置控制器连接。

2.46. 设置心跳包功能

函 数 名 ： long OPTController_SetEthernetConnectionHeartBeat(OPTController_Handle controllerHandle, unsigned timeout);
函数描述：控制器连接之后设置发送心跳包的功能，只对网口控制器有效，SDK 默认设置心跳包的时间是 5s。
输入参数：

- controllerHandle --控制器句柄；
- timeout --心跳超时时间，范围：[1-65535]，单位：1s，当值为 0 时，心跳包功能失效（即不发送心跳包）；当值大于 0 时，建议每隔 timeout/2 秒发送一次心跳包。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_HEARTBEAT_FAILED （见错误码表 1）。

2.47. 控制器是否连接

函数名：long OPTController_IsConnect(OPTController_Handle controllerHandle);
函数描述：检查控制器是否连接。
输入参数：controllerHandle --控制器句柄；
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_UNKNOWN （见错误码表 1）。

2.48. 获取通道状态

函 数 名 ： long OPTController_GetChannelState(OPTController_Handle controllerHandle, int channelIndex, int *state);
函数描述：获取指定通道的状态。
输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[1– 16]（十进制， 1-16 代表对应通道的通道序号）。

输出参数：state --读取到的通道状态，0 代表已连接光源；1 代表没有连接光源；2 代表短路保护；3 代表过压保护；4 代表过流保护。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_GET_CHANNELSTATE_FAILED （见错误码表 1）。

2.49. 以太网在线设备搜索

函数名：long OPTController_GetControllerListOnEthernet(char *snList);
函数描述：搜索在线控制器，得到它们的序列号。
输出参数：snList --获取到的在线设备的序列号，序列号之间用逗号（“，”）分开，例

如：“AA53017016,AA54278910”。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_GETCONTROLLERLIST_FAILED （见错误码表 1）。

2.50. 设置 Keepalive 相关参数（控制器固件需 V3.2.7 及以上）

函数名： long OPTController_SetKeepaliveParameter(OPTController_Handle controllerHandle, int keepalive_time,int keepalive_intvl, int keepalive_probes);

函数描述：设置 keepalive 功能的参数。

输入参数：

- controllerHandle --控制器句柄；
- keepalive_time --允许的持续空闲时间，范围：[1-65535]，单位 1s，默认设置为 5s；
- keepalive_intvl --keepalive 包的发送周期，范围：[1-65535]，单位 1s，默认设置为 3s；
- keepalive_probes --keepalive 包的发送次数，范围：[1-65535]，单位 1s，默认设置为 9 次。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_KEEPALIVEPARAMETERS_FAILED（见错误码表 1）。

2.51. 开启/关闭控制器 keepalive 功能（控制器固件需 V3.2.7 及以上）

函数名： long OPTController_EnableKeepalive(OPTController_Handle controllerHandle, BOOL enable);

函数描述：启用或禁用 keepalive 功能。

输入参数：

- controllerHandle --控制器句柄；
- enable --“TRUE”表示开启控制器 keepalive 功能，”FALSE”表示关闭控制器 keepalive 功能。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_ENABLE_KEEPALIVE_FAILED （见错误码表 1）。

2.52. 软件触发（控制器固件需 V3.3.1 及以上）

函数名： long OPTController_SoftwareTrigger(OPTController_Handle controllerHandle, int channelIndex, int time);

函数描述：在指定的时间，设置指定通道的软件触发。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0– 16]（十进制，0 代表所有通道，1-16 代表对应通道的通道序号）；
- time --照射持续时间，范围：[1-3000]，单位：10ms。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SOFTWARETRIGGER_FAILED （见错误码表 1）。

2.53. 多通道软件触发（控制器固件需 V3.3.1 及以上）

函数名： long OPTController_MultiSoftwareTrigger(OPTController_Handle controllerHandle, SoftwareTriggerItem* softwareTriggerArray, int length);

函数描述：为指定的多通道设置软件触发。

输入参数：

- controllerHandle --控制器句柄；
- softwareTriggerArray --一个包含了多个照射持续时间和对应通道序号的数组，照射持续时间的范围：[1-3000]，单位：10ms；
- length --软件触发数组长度。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SOFTWARETRIGGER_FAILED，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE （见错误码表 1）。

参见：软件触发。

2.54. 读取可编程触发的总步骤数（控制器固件需 V3.3.1 及以上）

函数名： long OPTController_ReadStepCount(OPTController_Handle controllerHandle, int moduleIndex, int* count);

函数描述：读取指定模组的可编程触发总步骤数。

输入参数：

- controllerHandle --控制器句柄；
- moduleIndex --模组的序号，模组序号取值范围：[1 - 4]（十进制，1-4 代表对应模组序号（模组 1 包括 1-4 通道、模组 2 包括 5-8 通道、模组 3 包括 9-12 通道、模组 4 包括 13-16 通道））。

输出参数：count --特定通道的可编程触发总步骤数。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READSTEPCOUNT_FAILED，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE （见错误

码表 1）。

2.55. 设置可编程触发模式（控制器固件需 V3.3.1 及以上）

函 数 名： long OPTController_SetTriggerMode(OPTController_Handle controllerHandle,int moduleIndex, int mode);

函数描述：设置指定模组的可编程触发模式。

输入参数：

- controllerHandle --控制器句柄；
- moduleIndex --模组的序号，模组序号取值范围：[1 - 4]（十进制，1-4 代表对应模组序号（模组 1 包括 1-4 通道、模组 2 包括 5-8 通道、模组 3 包括 9-12 通道、模组 4 包括 13-16 通道））；
- mode --触发模式，取值范围：[1-2]。1 代表普通触发模式，2 代表可编程触发模式。

返回值：

- 操作成功：OPT_SUCCEED；
- 操 作 失 败 ： OPT_ERR_SOFTWARETRIGGER_FAILED ， OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE（见错误码表 1）。

2.56. 读取可编程触发模式（控制器固件需 V3.3.1 及以上）

函 数 名： long OPTController_ReadTriggerMode(OPTController_Handle controllerHandle, int modulelIndex, int *mode);

函数描述：读取指定模组的可编程触发模式。

输入参数：

- controllerHandle --控制器句柄；
- moduleIndex --模组的序号，模组序号取值范围：[1 - 4]（十进制，1-4 代表对应模组序号（模组 1 包括 1-4 通道、模组 2 包括 5-8 通道、模组 3 包括 9-12 通道、模组 4 包括 13-16 通道））。

输出参数：mode --获取到的触发模式。1 代表普通触发模式，2 代表可编程触发模式。

返回值：

- 操作成功：OPT_SUCCEED；
- 操 作 失 败 ： OPT_ERR_READTRIGGERMODE_FAILED ， OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE（见错误码表 1）。

2.57. 设置可编程触发当前步骤序号（控制器固件需 V3.3.1 及以上）

函数名： long OPTController_SetCurrentStepIndex(OPTController_Handle controllerHandle, int

moduleIndex, int curStepIndex);

函数描述：设置指定模组的可编程触发模式。

输入参数：

- controllerHandle --控制器句柄；
- moduleIndex --模组的序号，模组序号取值范围：[1 - 4]（十进制，1-4 代表对应模组序号（模组 1 包括 1-4 通道、模组 2 包括 5-8 通道、模组 3 包括 9-12 通道、模组 4 包括 13-16 通道））；
- curStepIndex --指定通道的当前步骤序号，取值范围：[1- 64]。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SETCURRENTSTEPINDEX_FAILED，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE（见错误码表 1）。

2.58. 读取可编程触发当前步骤序号（控制器固件需 V3.3.1 及以上）

函数名：long OPTController_ReadCurrentStepIndex(OPTController_Handle controllerHandle, int moduleIndex, int* curStepIndex);

函数描述：读取指定模组的可编程触发模式。

输入参数：

- controllerHandle --控制器句柄；
- moduleIndex --模组的序号，模组序号取值范围：[1 - 4]（十进制，1-4 代表对应模组序号（模组 1 包括 1-4 通道、模组 2 包括 5-8 通道、模组 3 包括 9-12 通道、模组 4 包括 13-16 通道））。

输出参数：curStepIndex --获取到的指定通道的当前步骤序号，范围：[1- 64]。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READCURRENTSTEPINDEX_FAILE，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE (见错误码表 1)。

2.59. 复位当前模组的可编程触发（控制器固件需 V3.3.1 及以上）

函数名：long OPTController_ResetSEQ(OPTController_Handle controllerHandle,int moduleIndex);

函数描述：复位当前模组的可编程触发。

输入参数：

- controllerHandle --控制器句柄；

- moduleIndex --模组的序号，模组序号取值范围：[1 - 4]（十进制，1-4 代表对应模组序号（模组 1 包括 1-4 通道、模组 2 包括 5-8 通道、模组 3 包括 9-12 通道、模组 4 包括 13-16 通道））。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_RESETSEQ_FAILED ，OPT_ERR_CHINDEX_OUTRANGE 或者 OPT_ERR_PARAM_OUTRANGE（见错误码表 1）。

2.60. 设置可编程触发表（控制器固件需 V3.3.1 及以上）

函 数 名 ： long OPTController_SetSeqTable(OPTController_Handle controllerHandle,int seqCount,int moduleIndex,int *triggerSource,int *intensity,int *pulseWidth);

函数描述：设置指定模组的可编程触发模式。

输入参数：

- controllerHandle --控制器句柄；
- seqCount --SEQ 数目，取值范围是：[1- 64]（十进制）；
- moduleIndex --模组的序号，模组序号取值范围：[1 - 4]（十进制，1-4 代表对应模组序号（模组 1 包括 1-4 通道、模组 2 包括 5-8 通道、模组 3 包括 9-12 通道、模组 4 包括 13-16 通道））；
- triggerSource --触发源数据，范围为[1-16]（模组序号为 1 时，表示触发源范围为[1-4]；模组序号为 2 时，表示触发源范围为[5-8]；模组序号为 3 时，表示触发源范围为[9-12]；模组序号为 4 时，表示触发源范围为[13-16]）；
- Intensity --光源强度值，范围为[0-255]；
- pulseWidth --脉冲宽度值，范围为[1-1023]。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SETSEQTABLEDATA_FAILED ，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE（见错误码表 1）。

2.61. 读取可编程触发表（控制器固件需 V3.3.1 及以上）

函 数 名 ： long OPTController_ReadSeqTable(OPTController_Handle controllerHandle,int moduleIndex,int * seqCount,int *triggerSource,int *intensity,int *pulseWidth);

函数描述：读取指定模组的可编程触发模式。

输入参数：

- controllerHandle --控制器句柄；
- moduleIndex --模组的序号，模组序号取值范围：[1 - 4]（十进制，1-4 代表对应模组序号（模组 1 包括 1-4 通道、模组 2 包括 5-8 通道、模组 3 包括 9-12 通道、模组 4 包括 13-16 通道））。

输出参数：

- seqCount --SEQ 数目，取值范围是：[1- 64]（十进制）；
- triggerSource --触发源数据，范围为[1-16]（模组序号为 1 时，表示触发源

范围为[1-4];模组序号为 2 时，表示触发源范围为[5-8];模组序号为 3 时，表示触发源范围为[9-12];模组序号为 4 时，表示触发源范围为[13-16])；

- Intensity --光源强度值,范围为[0-255]；
- pulseWidth --脉冲宽度值,范围为[1-1023]。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READSEQTABLEDATA_FAILED，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE（见错误码表 1）。

2.62. 设置触发延时（控制器固件需 V3.3.1 及以上）

函数名：long OPTController_SetTriggerDelay(OPTController_Handle controllerHandle,int channelIndex,int triggerDelay);

函数描述：设置指定通道的触发延时。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0– 16]（十进制，0 代表所有通道，1-16 代表对应通道的通道序号）；
- triggerDelay --触发延时，取值范围是：[0-65000]（十进制），单位 1us。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SETTRIGGERDELAY_FAILED，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE（见错误码表 1）。

2.63. 获取触发延时（控制器固件需 V3.3.1 及以上）

函数名：long OPTController_GetTriggerDelay(OPTController_Handle controllerHandle,int channelIndex,int *triggerDelay);

函数描述：获取指定通道的触发延时。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[1– 16]（十进制， 1-16 代表对应通道的 通道序号）。

输出参数：triggerDelay --获取到的触发延时，取值范围是：[0-65000]（十进制），单位 1us。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_GET_TRIGGERDELAY_FAILED，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE（见错误码表 1）。

2.64. 设置多通道触发延时（控制器固件需 V3.3.1 及以上）

函数名： long OPTController_SetMultiTriggerDelay(OPTController_Handle controllerHandle, TriggerDelayItem *triggerDelayArray, int length);
函数描述：设置指定的一个或者多个通道的触发延时。
输入参数：

- controllerHandle --控制器句柄；
- triggerDelayArray --一个包含了多个触发延时时间和对用通道序号的数组，触发延时取值范围：[0-65000]（十进制），单位：1us；如果触发延时小于 0，设置触发延时为 0；
- length --触发延时数组长度。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SETMULTITRIGGERDELAY_FAILED，OPT_ERR_CHINDEX_OUTRANGE 或 OPT_ERR_PARAM_OUTRANGE（见错误码表 1）。

参见：设置触发延时。

2.65. 获取控制器的通道数

函数名： long OPTController_GetControllerChannels(OPTController_Handle controllerHandle, int *channels);
函数描述：获取控制器的通道数。
输入参数：controllerHandle --控制器句柄。
输出参数：channels --获取到的控制器通道数。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_CHANNELS_FAILED（见错误码表 1）。

2.66. 读取保活开关状态

函数名： long OPTController_ReadKeepaliveSwitchState(OPTController_Handle controllerHandle, int *state);
函数描述：读取控制器的保活开关状态。
输入参数：controllerHandle --控制器句柄。
输出参数：state --获取到的控制器保活开关状态，0 代表关，1 代表开。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_KEEPALIVE_STATE_FAILED（见错误码表 1）。

2.67. 读取连续保活时间

函数名： long OPTController_ReadContinuousKeepaliveTime(OPTController_Handle controllerHandle, int *time);

函数描述：读取控制器的连续保活时间。
输入参数：controllerHandle --控制器句柄。
输出参数：time --获取到的控制器连续保活时间。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_KEEPALIVE_CONTINUOUS_TIME_FAILED（见错误码表 1）。

2.68. 读取探测包发送次数

函数名：long OPTController_ReadPacketDeliveryTimes(OPTController_Handle controllerHandle, int *times);
函数描述：读取控制器的探测包发送次数。
输入参数：controllerHandle --控制器句柄。
输出参数：times --获取到的控制器探测包发送次数。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_DELIVERY_TIMES_FAILED（见错误码表 1）。

2.69. 读取探测包发送间隔时间

函数名：long OPTController_ReadIntervalTimeOfPropPacket (OPTController_Handle controllerHandle, int *time);
函数描述：读取控制器的探测包发送的间隔时间。
输入参数：controllerHandle --控制器句柄。
输出参数：time --获取到的控制器探测包发送的间隔时间。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_INTERVAL_TIME_FAILED（见错误码表 1）。

2.70. 读取输出板的版本号

函数名：long OPTController_ReadOutputBoardVision(OPTController_Handle controllerHandle, char *vision);
函数描述：读取输出板的版本。
输入参数：controllerHandle --控制器句柄。
输出参数：vision --获取到的控制器输出板的版本号。
返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_OUTPUTBOARD_VISION_FAILED（见错误码表 1）。

2.71. 读取负载检查模式

函数名：long OPTController_ReadLoadDetectMode(OPTController_Handle controllerHandle,int channelIndex,int *mode);

函数描述：读取负载检查模式。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[1– 16]（十进制， 1-16 代表对应通道的通道序号）。

输出参数：mode --获取到的控制器负载模式，0 代表自动检查，1 代表手动设置最大电流值。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_DETECT_MODE_FAILED（见错误码表 1）。

2.72. 设置开机状态

函数名：long OPTController_SetBootState(OPTController_Handle controllerHandle,int channelIndex, int mode);

函数描述：设置指定通道的常亮模式开机状态。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0– 16]（十进制，0 代表所有通道，1-16 代表对应通道的通道序号）；
- mode --设置的开机保护模式，0 代表常亮模式，1 代表常灭模式。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_BOOT_STATE_MODE_FAILED（见错误码表 1）。

2.73. 读取各模块的开机状态

函数名：long OPTController_ReadModelBootState(OPTController_Handle controllerHandle,int channelIndex,int *state);

函数描述：读取指定通道模块的开机状态。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0– 16]（十进制，0 代表所有通道，1-16 代表对应通道的通道序号）。

输出参数：state --读取到的开机状态，0 代表常亮模式，1 代表常灭模式。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_MODEL_BOOT_MODE_FAILED（见错误码表 1）。

2.74. 设置时间单元

函数名：long OPTController_SetTimeUnit(OPTController_Handle controllerHandle,int channelIndex,int timeUnit);

函数描述：普通触发模式时间单位切换。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0– 16]（十进制，0 代表所有通道，1-16 代表对应通道的通道序号）；
- timeUnit --时间单位，范围:[0-3]，0 为 1us,1 为 10us,2 为 1ms,3 代表 100ms。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_SET_TIMEUNIT_FAILED （见错误码表 1）。

2.75. 读取时间单元

函 数 名 ： long OPTController_ReadTimeUnit(OPTController_Handle controllerHandle,int channelIndex,int *timeUnit);

函数描述：读取指定通道的时间单元。

输入参数：

- controllerHandle --控制器句柄；
- channelIndex --通道的序号，通道序号取值范围：[0– 16]（十进制，0 代表表所有通道，1-16 代表对应通道的通道序号）。

输出参数：timeUnit --时间单位，范围:[0-3]，0 为 1us,1 为 10us,2 为 1ms,3 代表 100ms。

返回值：

- 操作成功：OPT_SUCCEED；
- 操作失败：OPT_ERR_READ_TIMEUNIT_FAILED （见错误码表 1）。

附录

A. 常见问题解答（FAQ）

A.1 为什么连续操作时控制器响应不正确？

我们建议在两个连续的写读操作之间给出 20ms 的时间间隔以便控制器对指令做出相应的处理。

A.2 网口连接是否有较长的延时？

没有，如果在 50ms 内还未能建立连接（即，连接超时时间为 50ms），那么连接失败。

A.3 为什么在操作成功的情况下控制器返回错误码？

请检查是否允许返回值（设置允许返回值，请参见章节 2.19）。

A.4 为什么控制器无法找到 PC 串口或无法建立串口连接？

如果 PC 安装了 WIN7china.com 操作系统，请以管理员身份运行。

A.5 为什么 Demo 程序无法打开或者 SDK 无法被调用（提示系统错误）？

请安装 VS2008 runtime library 解决此问题。

A.6 为什么改变 MAC 地址后控制器无法正确连接？

改变 MAC 地址后请重启计算机。

A.7 控制器为什么要加入心跳包功能？

如果网口连接在非正常情况下（比如程序异常关闭、网络物理原因断开等）断开而控制器网口仍然通电，控制器无法判断该连接已经失效。控制器加入心跳包功能后，客户端每个一段时间（心跳包超时时间的一半）就会向控制器发送心跳包。控制器在收到心跳包后检查连接是否有效。在 1.0.8 版本之后，默认情况下，创建网口连接时，自动激活心跳包功能。设置心跳包详见章节 2.43。

A.8 怎么发送有效的心跳包？

一般使用线程在指定时间间隔内（建议心跳时间的 1/2。比如设置心跳为 5s，发送心跳

包时间 2.5s）发送心跳指令（命令字为 0x46）。需要强调的是，必须要求控制器能在心跳时间间隔内接受到心跳包，否则控制器会因为收不到心跳包而断线。如果发送心跳包的线程优先级过低，一旦有其他资源在一段时间内过多的占用 CPU 资源，那么心跳包就不能准时发送。类似地，定时器在线程中的优先级较低，极有可能造成心跳包无法准时发送出去。

A.9 为什么通信过程中，偶尔出现指令失效的现象？

控制器对每条指令都有一个响应时间，在一条指令的响应时间内，如果有另一个指令到达，控制器就会放弃当前指令。因此未来确保在通信过程中控制器能有效地响应每一条指令，做如下建议：

- 在比较好使的功能之后设置延时；
- 对每条指令设置返回值以判断控制器对该指令是否处理完毕；
- 通信方式必须是同步阻塞模式。

A.10 在网口通讯中，当有多个控制器时，为什么通过 SN（或 IP 地址）连接的控制器不正确，即所连接的与选择的控制器的 SN（或 IP）不一致？

请检查是否有多个控制器共相同的 IP 现象。需注意每个控制器出厂时都有一个默认的静态 IP：192.168.1.16。在局域网内的控制器，请确保相互之间 IP 地址的唯一性。

A.11 为什么可以搜索到控制器，却无法连接？

请检查客户端和控制器是否在同一网段上。

B 错误码宏定义

表 1：错误码宏定义

宏名称	错误码	备注
OPT_SUCCEED	0	操作成功
OPT_ERR_INVALIDHANDLE	3001001	无效的句柄
OPT_ERR_UNKNOWN	3001002	未知错误
OPT_ERR_INITSERIAL_FAILED	3001003	初始化串口失败
OPT_ERR_RELEASESERIALPORT_FAILED	3001004	释放串口失败
OPT_ERR_SERIALPORT_UNOPENED	3001005	试图访问一个不存在的串口
OPT_ERR_CREATEETHECON_FAILED	3001006	创建网口连接失败
OPT_ERR_DESTROYETHECON_FAILED	3001007	释放网口连接失败
OPT_ERR_SN_NOTFOUND	3001008	SN 未找到
OPT_ERR_TURNONCH_FAILED	3001009	打开指定通道失败
OPT_ERR_TURNOFFCH_FAILED	3001010	关闭指定通道失败
OPT_ERR_SET_INTENSITY_FAILED	3001011	设置亮度失败
OPT_ERR_READ_INTENSITY_FAILED	3001012	读取亮度失败
OPT_ERR_SET_TRIGGERWIDTH_FAILED	3001013	设置触发脉宽失败
OPT_ERR_READ_TRIGGERWIDTH_FAILED	3001014	读取触发脉宽失败
OPT_ERR_READ_HBTRIGGERWIDTH_FAILED	3001015	读取高亮触发脉宽失败
OPT_ERR_SET_HBTRIGGERWIDTH_FAILED	3001016	设置高亮触发脉宽失败
OPT_ERR_READ_SN_FAILED	3001017	读取控制器序列号失败
OPT_ERR_READ_IPCONFIG_FAILED	3001018	读取控制器 IP 配置失败
OPT_ERR_CHINDEX_OUTRANGE	3001019	通道序号越界
OPT_ERR_WRITE_FAILED	3001020	写操作失败
OPT_ERR_PARAM_OUTRANGE	3001021	参数越界
OPT_ERR_READ_MAC_FAILED	3001022	读取控制器 MAC 地址失败
OPT_ERR_SET_MAXCURRENT_FAILED	3001023	设置最大电流失败
OPT_ERR_READ_MAXCURRENT_FAILED	3001024	读取最大电流失败
OPT_ERR_SET_TRIGGERACTIVATION_FAILED	3001025	设置触发极性失败
OPT_ERR_READ_TRIGGERACTIVATION_FAILED	3001026	读取触发极性失败
OPT_ERR_SET_WORKMODE_FAILED	3001027	设置工作模式失败
OPT_ERR_READ_WORKMODE_FAILED	3001028	读取工作模式失败
OPT_ERR_SET_BAUDRATE_FAILED	3001029	设置波特率失败
OPT_ERR_SET_CHANNELAMOUNT_FAILED	3001030	设置通道数失败
OPT_ERR_SET_DETECTEDMINLOAD_FAILED	3001031	设置检测最小负载失败
OPT_ERR_READ_OUTERTRIGGERFREQUENCYUPPERBOUND_FAILED	3001032	读取外部触发频率上限值失败

OPT_ERR_SET_AUTOSTROBEFREQUENCY_FAILED	3001033	设置自动频闪频率失败
OPT_ERR_READ_AUTOSTROBEFREQUENCY_FAILED	3001034	读取自动频闪频率失败
OPT_ERR_SET_DHCP_FAILED	3001035	动/静态 IP 模式切换失败
OPT_ERR_SET_LOADMODE_FAILED	3001036	设置负载模式失败
OPT_ERR_READ_PROPERTY_FAILED	3001037	读取控制属性失败
OPT_ERR_CONNECTION_RESET_FAILED	3001038	重置连接失败
OPT_ERR_SET_HEARTBEAT_FAILED	3001039	设置发送心跳包失败
OPT_ERR_GETCONTROLLERLIST_FAILED	3001040	获取设备列表失败
OPT_ERR_SOFTWARETRIGGER_FAILED	3001041	设置软触发失败
OPT_ERR_GET_CHANNELSTATE_FAILED	3001042	获取通道状态失败
OPT_ERR_SET_KEEPALIVEPARAMETERS_FAILED	3001043	设置保活参数失败
OPT_ERR_ENABLE_KEEPALIVE_FAILED	3001044	开启/关闭保活功能失败
OPT_ERR_READSTEPCOUNT_FAILED	3001045	读取可编程触发步骤数失败
OPT_ERR_SETTRIGGERMODE_FAILED	3001046	设置可编程触发模式失败
OPT_ERR_READTRIGGERMODE_FAILED	3001047	读取可编程触发模式失败
OPT_ERR_SETCURRENTSTEPINDEX_FAILED	3001048	设置可编程触发当前步骤失败
OPT_ERR_READCURRENTSTEPINDEX_FAILED	3001049	读取可编程触发当前步骤失败
OPT_ERR_RESETCURRENTSTEPINDEX_FAILED	3001050	复位可编程触发当前步骤失败
OPT_ERR_SETTRIGGERDELAY_FAILED	3001051	设置触发延时失败
OPT_ERR_GET_TRIGGERDELAY_FAILED	3001052	获取触发延时失败
OPT_ERR_SETMULTITRIGGERDELAY_FAILED	3001053	设置多通道触发延时失败
OPT_ERR_SETSEQTABLEDATA_FAILED	3001054	设置 SEQ 数据表失败
OPT_ERR_READSEQTABLEDATA_FAILED	3001055	读取 SEQ 数据表失败
OPT_ERR_READ_CHANNELS_FAILED	3001056	读取控制器通道数失败
OPT_ERR_READ_KEEPALIVE_STATE_FAILED	3001057	读取保活开关状态失败
OPT_ERR_READ_KEEPALIVE_CONTINUOUS_TIME_FAILED	3001058	读取连续保活时间失败
OPT_ERR_READ_DELIVERY_TIMES_FAILED	3001059	读取探测包发送次数失败
OPT_ERR_READ_INTERVAL_TIME_FAILED	3001060	读取探测包发送时间间隔失败
OPT_ERR_READ_OUTPUTBOARD_VISION_FAILED	3001061	读取输出板版本号失败
OPT_ERR_READ_DETECT_MODE_FAILED	3001062	读取负载检测模式失败
OPT_ERR_SET_BOOT_PROTECTION_MODE_FA	3001063	设置开机保护模式失败

ILED		
OPT_ERR_READ_MODEL_BOOT_MODE_FAILED	3001064	读取各模块的开机状态失败
OPT_ERR_SET_OUTERTRIGGERFREQUENCYUPPERBOUND_FAILED	3001065	设置尾部触发频率上限值失败
OPT_ERR_SET_IPCONFIG_FAILED	3001066	设置网口配置失败
OPT_ERR_SET_VOLTAGE_FAILED	3001067	设置电压失败
OPT_ERR_READ_VOLTAGE_FAILED	3001068	读取电压失败
OPT_ERR_SET_TIMEUNIT_FAILED	3001069	设置时间单元失败
OPT_ERR_READ_TIMEUNIT_FAILED	3001070	读取时间单元失败

注：英文缩略词参见表 2。

C 英文缩写表

表 2：英文缩写表

缩写	英文	中文
CH	channel	通道
CON	connection	连接
CONFIG	configuration	配置
CUR	current	电流
ERR	error	错误
HB	high brightness	高亮
PARAM	parameter	参数
SN	serial number	序列号
UDP	User Datagram Protocl	用户数据报协议