**MiniLed接口说明文档**

**武汉普赛斯仪表有限公司**

**声明：**本文件所有权和解释权归武汉普赛斯仪表有限公司所有，未经武汉普赛斯仪表有限公司书面许可，不得复制或向第三方公开。

修订历史记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版次** | **发布日期** | **AMD** | **修订者** | **说明** |
| v0.0.1 | 2022.07.29 | A | Lsb | 初稿 |
| V0.0.2 | 2022.08.26 | A | Lsb | 新增GetLastScpiEexcStatu接口 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

（A-添加，M-修改，D-删除）

# 一、接口说明

* 1. 初始化动态库

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_Init |
| 功能 | 初始化动态库 |
| 输入参数 | 无 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_Init();

* 1. 卸载动态库

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_Uninit |
| 功能 | 释放动态库 |
| 输入参数 | 无 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_Uninit(void);

* 1. 网口连接设备

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_OpenNetDevice |
| 功能 | 连接设备 |
| 输入参数 |  |
| strDevIP | 设备IP，端口：9009 |
| 输出参数 | 无 |
| 返回值 | >=0,表示成功，返回设备ID，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_OpenNetDevice(const char\* strDevIP,int port);

* 1. 串口连接设备

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_OpenUartDevice |
| 功能 | 连接设备 |
| 输入参数 |  |
| strUartName | 串口名称 |
| nBaudRate | 波特率 115200 |
| 输出参数 | 无 |
| 返回值 | >=0,表示成功，返回设备ID，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_OpenUartDevice(const char\* strUartName, int nBaudRate);

* 1. 断开设备连接

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_CloseUartDevice |
| 功能 | 关闭设备连接 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_CloseUartDevice(int nDevID);

* 1. 获取设备信息

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_GetDevIDN |
| 功能 | 获取设备IDN |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| sIDN[] | 返回的IDN字符串，外部申请空间 |
| nSize | 对应数组大小 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_GetDevIDN(int nDevID, char sIDN[], int \*nSize);

* 1. 复位设备

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_ResetDev |
| 功能 | 设备初始化，该指令清除设备所有指令设置的电压电流相关信息，恢复至设备上电初始设置值。 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_ResetDev(int nDevID);

* 1. 查询设备状态

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_GetLastScpiEexcStatu |
| 功能 | 可检测设备是否离线，后续拓展查询上一次指令是否异常。 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_GetLastScpiEexcStatu(int nDevID);

* 1. 触发模式设置

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_SetLedTestMode |
| 功能 | 设置触发模式 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| mode | MiniLed\_TEST\_MODE\_MAN=0x01, 手动  MiniLed\_TEST\_MODE\_AUTO=0x02, 自动 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetTestMode(int nDevID, MiniLed\_TEST\_MODE mode);

* 1. 触发线设置

|  |  |
| --- | --- |
| 函数名 | PSSMiniledSDK\_SetTrigLoad |
| 功能 | 设置Trig |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nLineID | 只能设置为0-16的值 |
| bInput | true表示输入 false表示输出 |
| eMiniLedTrigType | trig类型，参考MiniLed\_Trig\_Event\_Type\_Tag |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetTrigLoad(int nDevID, int nLineID, bool bInput, MiniLed\_Trig\_Edge\_Type eEdgeType,MiniLed\_Trig\_Event\_Type eMiniLedTrigType, int nWaitTime, int nDuration);

* 1. 清理触发线

|  |  |
| --- | --- |
| 函数名 | PSSMiniledSDK\_ClearTrig |
| 功能 | 清除Trig |
| 输入参数 |  |
|  |  |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_ClearTrig(int nDevID);

* 1. 测试项添加

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_SetTestItemCfg |
| 功能 | 设置测试项参数,该函数调用后，会将之前的设置清空 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| eMiniLedTestProjectType | 测试项类型，参考MiniLed\_Test\_Item\_Type |
| fSrcVal[], fLmtVal | ■测试项为VF时，fSrcVal为正向电流设置值，一次传入两个值，fLmtVal为限值电压值，单位均为(A/V);  ■测试项为VR时，fSrcVal为反向电流设置值，一次传入一个值，fLmtVal为限值电压值，单位均为(A/V);  ■测试项为IR时，fSrcVal为反向电压设置值，一次传入一个值，fLmtVal为限值电流值，单位均为(A/V);  ■测试项为LPSP时，fSrcVal为反向电压设置值，一次传入两个值，fLmtVal为控制板电流量程，单位均为(A/V); |
| nSrcValCount | 设置值数组大小 |
| nSampleDelay | 采样延时,单位S |
| fNplc | NPLC设定值 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetTestItemCfg(int nDevID,MiniLed\_Test\_Item\_Type eMiniLedTestProjectType,double fSrcVal[],int nSrcValCount, double fLmtVal, double fSampleDelay,fNplc);

* 1. 测试项追加

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_AppendTestItemCfg |
| 功能 | 追加一条测试项参数,该函数调用后，会将之前的设置基础上追加一条测试项 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| eMiniLedTestProjectType | 测试项类型，参考MiniLed\_Test\_Item\_Type |
| fSrcVal[],  fLmtVal | ■测试项为VF时，fSrcVal为正向电流设置值，一次传入两个值，fLmtVal为限值电压值，单位均为(A/V);  ■测试项为VZ时，fSrcVal为反向电流设置值，一次传入一个值，fLmtVal为限值电压值，单位均为(A/V);  ■测试项为IR时，fSrcVal为反向电压设置值，一次传入一个值，fLmtVal为限值电流值，单位均为(A/V);  ■测试项为LPSP时，fSrcVal为反向电压设置值，一次传入两个值，fLmtVal为控制板电流量程，单位均为(A/V); |
| nSrcValCount | 设置值数组大小 |
| nSampleDelay | 采样延时,单位S |
| fNplc | NPLC设定值 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_AppendTestItemCfg(int nDevID, MiniLed\_Test\_Item\_Type eMiniLedTestProjectType,double fSrcVal[],int nSrcValCount, double fLmtVal, double fSampleDelay,fNplc);

* 1. 通道设置

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_SetTestChannelList |
| 功能 | 设置当前测试的通道列表，设备总共4个通道，可以设置指定通道启动测试，该接口如不调用，默认测试四个通道 |
|  |  |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nChannelID | 指定测试通道ID列表，通道号从1开始 |
| nChannelCount | 指定的测试通道数量 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetTestChannelList(int nDevID,int nChannelID[],int nChannelCount);

* 1. 启动所有业务测试（指令触发）

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_TestStart |
| 功能 | 启动业务测试 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_TestStart(int nDevID);

* 1. 使用指令触发设置（指令触发）

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_TestStart |
| 功能 | 使用指令触发设置 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetTrigDebugMode(int nDevID,bool bOn);

* 1. 启动电测试（指令触发）

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_TestStart |
| 功能 | 新版miniled模拟触发线启动电测试 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_TestStartElect(int nDevID);

* 1. 启动光测试（指令触发）

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_TestStart |
| 功能 | 新版miniled模拟触发线启动光测试 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_TestStartLightON(int nDevID);

* 1. 读取业务测试数据（字符串）

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_GetTestDataStr |
| 功能 | 读取业务测试数据 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nData | 接收的数据 |
| nSize | 接受数据的长度 |
| nTimeout | 超时时间 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_GetTestDataStr(int nDevID, char nData[],int \*nSize, int nTimeout=2000);

* 1. 读取业务测试数据（结构体）

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_GetTestResult |
| 功能 | 获取测试结果，该接口返回的测试结果为所有测试项的全部测试数据 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| struMiniLedTestItemResult | 以数组形式返回全部数据结果，参考MiniLedTestItemResult说明，数组空间外部申请 |
| nTestItemCount | 当前测试项数量，对应数组大小 |
| nTimeOut | 超时返回时间,单位ms |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_GetTestResult(int nDevID, MiniLedTestItemResult struMiniLedTestItemResult[], int nTestItemCount, int nTimeOut = 2000);

* 1. 关闭光测试（指令触发）

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_TestStart |
| 功能 | 新版miniled模拟触发线关闭光测试 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_TestStartLightOFF(int nDevID);

* 1. 发送指令（SCPI）

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_SendData |
| 功能 | 发送指定数据 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nData | 待发送的数据 |
| nTimeout | 超时时间 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SendData(int nDevID, char nData[],int nSize, int nTimeout = 1000);

* 1. 读取数据(SCPI)

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_RecvData |
| 功能 | 读取缓存数据 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nData | 接收的数据 |
| nSize | 接受数据的长度 |
| nTimeout | 超时时间 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_RecvData(int nDevID, char nData[],int \*nSize, int nTimeout=2000);

* 1. 源设置

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_SetSource |
| 功能 | 源设置 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nChannelID | 模拟板通道ID，通道号从1开始 |
| bIsVSrc | true,表示电压源；false，表示电流源 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetSource(int nDevID, int nChannelID,

bool bIsVSrc);

* 1. 源值设置

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_SetSourceValue |
| 功能 | 源值设置 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nChannelID | 模拟板通道ID，通道号从1开始 |
| bIsVSrc | true,表示电压源；false，表示电流源 |
| fValue | 源值,电压单位V，电流单位A |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetSourceValue(int nDevID,int nChannelID,bool bIsVSrc, double fValue);

* 1. 限值设置

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_SetLimitValue |
| 功能 | 限值设置 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nChannelID | 模拟板通道ID，通道号从1开始 |
| bIsVSrc | true,表示电压源；false，表示电流源 |
| fLimitValue | 设备限值值,电压单位V，电流单位A |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetLimitValue(int nDevID,int nChannelID,bool bIsVSrc, double fLimitValue);

* 1. 源量程设置

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_SetAnlgBoardSourceRange |
| 功能 | 设置模拟板源量程 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nChannelID | 模拟板通道ID，通道号从1开始 |
| bIsVSrc | true,表示电压源；false，表示电流源 |
| fRange | 量程，单位A |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetAnlgBoardSourceRange(int nDevID, int nChannelID,bool bIsVSrc, double fRange);

* 1. 限量程设置

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_SetAnlgBoardLimitRange |
| 功能 | 设置模拟板限值量程 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nChannelID | 模拟板通道ID，通道号从1开始 |
| bIsVLmt | true,表示限值电压；false，表示限值电流 |
| fRange | 量程，单位A |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetAnlgBoardLimitRange(int nDevID, int nChannelID,bool bIsVLmt, double fRange);

* 1. NPLC设置

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_SetNPLC |
| 功能 | 设置模拟板和控制板NPLC |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nChannelID | 0表示控制板ID,1-4表示模拟板通道ID |
| bIsVSrc | true,表示设置电压；false，表示设置电流 |
| fNPLC | 取值范围为 0.01~10，其中 0.01 为最小 NPLC，10 为最大 NPLC,设备会根据用户输入值匹配最佳 NPLC 值 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetNPLC(int nDevID, int nChannelID,bool bIsVSrc, double fNPLC);

* 1. 设置设备输出状态

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_SetOutPut |
| 功能 | 设置设备输出状态 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| bOn | true 设备开输出 false 设备关输出 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_SetOutPut(int nDevID, bool bOn);

* 1. 读取电压电流值

|  |  |
| --- | --- |
| 函数名 | PSSMiniLedSDK\_ReadData |
| 功能 | 读取电流电压值 |
| 输入参数 |  |
| nDevID | PSSMiniLedSDK\_OpenUartDevice返回值 |
| nChannelID | 模拟板通道ID，通道号从1开始 |
| sTestData | 电压电流值 格式3.0,0.2 |
| nSize | sTestData长度 |
| 输出参数 | 无 |
| 返回值 | 0表示成功，错误时返回错误码 |

int \_\_stdcall PSSMiniLedSDK\_ReadData(int nDevID,int nChannelID, double \*Vx,double \*Ix, int nTimeOut = 10000);

# 二、结构体

#define MAX\_TEST\_CHANNEL\_NUM 4

#define MAX\_TEST\_POINT\_COUNT 16

/\*----------------------------- MiniLed项目测试项类型定义------------------------------------\*/

typedef enum MiniLed\_Test\_Item\_Type\_Tag{

MiniLed\_Test\_Item\_Type\_VF = 0x01, // 正向电压测试

MiniLed\_Test\_Item\_Type\_VR = 0x02, // 反向击穿电压测试

MiniLed\_Test\_Item\_Type\_IR = 0x03, // 反向泄露电流测试

MiniLed\_Test\_Item\_Type\_LPSP = 0x04, // 光功率和光谱测试

MiniLed\_Test\_Item\_Type\_VZ = 0x05 // 反向击穿电压测试

}MiniLed\_Test\_Item\_Type;

typedef enum MiniLed\_Trig\_Event\_Type\_Tag{

MiniLed\_Trig\_Event\_Type\_ELEC = 0x01, //电信号

MiniLed\_Trig\_Event\_Type\_LIGHT = 0x02, //光测试完成

MiniLed\_Trig\_Event\_Type\_READY = 0x03, //设备就绪信号

MiniLed\_Trig\_Event\_Type\_USER = 0x04, //led测试完成信号

MiniLed\_Trig\_Event\_Type\_LIGHT\_ON = 0x05, //光测试启动

MiniLed\_Trig\_Event\_Type\_LIGHT\_OFF = 0x06 //光测试停止

}MiniLed\_Trig\_Event\_Type;

typedef enum MiniLed\_Trig\_Edge\_Type\_Tag{

MiniLed\_Trig\_Edge\_Type\_RISE = 0x01, //上升沿触发

MiniLed\_Trig\_Edge\_Type\_FALL = 0x02, //下降沿触发

MiniLed\_Trig\_Edge\_Type\_EITH = 0x03, //边沿触发

}MiniLed\_Trig\_Edge\_Type;

typedef enum MiniLed\_TEST\_MODE\_Tag{

MiniLed\_TEST\_MODE\_MAN = 0x01, //手动

MiniLed\_TEST\_MODE\_AUTO = 0x02, //自动

}MiniLed\_TEST\_MODE;

/\* 通用测试项返回值，主要应用与VF,VZ和IR测试项

\*\* VF返回正向电压值，VR表示反向击穿电压值，IR表示泄露电流值

\*\* 返回值数量与设置时，一次设置的电位数量相对应，目前VF一次设置两个

\*\* 点，返回两个值，两个峰值差；VZ,IR一次设置一个点，返回一个值,单位为A/V\*/

typedef struct ComTestItemResultTag{

double fVal[MAX\_TEST\_POINT\_COUNT];

int nValidCount;

}ComTestItemResult, \*LPComTestItemResult;

/\*------------------------------ LPSP测试项返回值-------------------------------------------\*/

typedef struct LPSPTestItemResultTag{

double fMaxPower; // 光功率，单位为A

}LPSPTestItemResult, \*LPSPspTestItemResult;

/\*------------------------------ 测试结果数据定义--------------------------------------------\*/

typedef struct MiniLedTestItemResultTag{

MiniLed\_Test\_Item\_Type eMiniLedTestItemType; // 测试项类型

ComTestItemResult struComTestItemResutlt[MAX\_TEST\_CHANNEL\_NUM]; // 对应VF,VR,IR测试类型

LPSPTestItemResult struLPSPTestItemResult[MAX\_TEST\_CHANNEL\_NUM]; // 对应LPSP测试类型

}MiniLedTestItemResult, \*LPMiniLedTestItemResult;

# 三、接口返回错误代码

PssRstSuccess = 0, /\* 成功 \*/

PssRstParameterError = -1, /\* 参数输入错误 \*/

PssRstDevCreateError = -2, /\* 设备构造失败 \*/

PssRstDeviceNotInit = -3, /\* 设备未初始化\*/

PssRstScpiClientNotFound = -4, /\* 未找到scpiClient对象 \*/

PssRstNotImpl = -5, /\* 未实现 \*/

PssRstOutputFail = -6, /\* 设备Output失败 \*/

PssRstDeviceNotConnect = -7, /\* 设备未连接 \*/

PssRstSetSrcValError = -8, /\* 源值设置失败 \*/

PssRstSetSrcRangeError = -9, /\* 源量程设置失败 \*/

PssRstSetLmtValError = -10, /\* 限值设置失败 \*/

PssRstSetLmtRangeError = -11, /\* 限量程设置失败 \*/

PssRstSetSrcError = -12, /\* 源设置失败 \*/

PssRstSetNPLCError = -13, /\* 设置NPLC指令失败 \*/

PssRstSetMeasureSrcError = -14, /\* 设置测试电压电流失败 \*/

PssRstReadError = -15, /\* 读取设备数据失败 \*/

PssRstAlreadyConnect = -16, /\* 设备已经初始化 \*/

PssRstRecvFromIOError = -17, /\* 接收IO数据失败 \*/

PssRstIOEmptyError = -18, /\* IO 为空 \*/

PssRstOptTimeoutError = -19, /\* 超时错误 \*/

PssRstWriteToIOError = -20, /\* 向IO写入数据失败 \*/

PssRstReadFromIOError = -21, /\* 读取IO数据失败 \*/

PssRstMallocFailed = -22, /\* 内存申请失败 \*/

PssRstSetSensFuncError = -23, /\* 限设置失败 \*/

PssRstBufIsNotEnough = -24, /\* 数据不足以支持后续解析 \*/

PssRstResponseArrived = -25, /\* 响应到达 \*/

PssRstResponseMismatch = -26, /\* 请求与响应不匹配 \*/

PssRstSet24WError = -27, /\* 设置2/4线失败 \*/

PssRstSetFBError = -28, /\* 设置前后面板失败 \*/

PssRstSetMasterError = -29, /\* 主从机设置失败 \*/

PssRstSetTrigDirError = -30, /\* 设置Trig方向失败 \*/

PssRstSetScanSrcError = -31, /\* 扫描源设置失败 \*/

PssRstSetScanStartError = -32, /\* 设置扫描起点失败 \*/

PssRstSetScanStopError = -33, /\* 设置扫描终点失败 \*/

PssRstSetScanPointError = -34, /\* 设置扫描点数失败 \*/

PssRstSetCustomValError = -35, /\* 设置自定义数据失败 \*/

PssRstOverFlowError = -36, /\* 溢出错误 \*/

PssRstRepeatError = -37, /\* 重复调用或设备错误 \*/

PssRstSetValOverFlow = -38, /\* 设置值超过极限值 \*/

PssRstScpiFormatError = -39, /\* scpi格式错误 \*/

PssRstScpiCmdError = -40, /\* scpi指令错误 \*/

PssRstSetResolutinError = -41, /\* 设置分辨率出错 \*/

PssRstCheckError = -42, /\* 检查出错 \*/

PssRstOverFlow = -43, /\* 缓存溢出 \*/

PssRstGPIBOpenError = -44, /\* GPIB打开失败 \*/

PssRstGPIBNotOpen = -45, /\* GPIB未打开 \*/

PssRstGPIBWriteError = -46, /\* GPIB写失败 \*/

PssRstGPIBReadError = -47, /\* GPIB读失败 \*/

PssRstSocketBindError = -48, /\* socket绑定错误 \*/

PssRstSocketConnectError = -49, /\* socket连接至服务端错误 \*/

PssRstUnknowPssBinFrm = -50, /\* 错误的PssBin协议帧 \*/

PssRstAnlgOptError = -51, /\* 模拟板底层操作出错 \*/

PssRstPwrOptError = -52, /\* 电源板底层操作出错 \*/

PssRstFileError = -53, /\* 文件异常错误 \*/

PssRstUnknowDevice = -54, /\* 未知设备类型 \*/

PssRstCalError = -55, /\* 校准失败 \*/

PssRstUartConfigError = -56, /\* 串口配置失败 \*/

PssRstOverFlowOfIndex = -57, /\* 索引溢出 \*/

PssRstIOBrokenErr = -61,

PssRstSysCallError = -62, /\* system call error \*/