

NrClientSDK 使用手册

版本： Ver 0.4 2023-06-12

- NrClientSDK 使用手册.....1
 - 一、概述.....2
 - 二、文件.....2
 - 三、接口.....2
 - 1、初始化.....2
 - 2、增加一个网络读卡器.....2
 - 3、删除一个网络读卡器.....2
 - 4、发送声光提醒请求.....3
 - 5、发送批量声光提醒请求.....3
 - 6、发送短消息文本通知请求.....4
 - 7、打开 TELNET 调试.....4
 - 8、关闭 TELNET 调试.....4
 - 三、类型、事件、返回值定义5
 - 1、事件回调接口定义.....5
 - 2、标签结构体定义.....5
 - 3、声光提醒标签结构体定义.....5
 - 4、事件定义.....5
 - 5、返回值定义.....6

一、概述

NrClientSDK 封装了与网络读卡器通讯的协议。通过简单的接口调用可以跟读卡器通讯，获取到读卡器读到的标签信息。

二、文件

开发包包含三个文件“NrClientSDK.h”、“NrClientSDK.dll”、“NrClientSDK.lib”。

三、接口

1、初始化

使用开发包首先需要调用该接口初始化开发包。

原型：`NRCLIENTSDK_API int __stdcall NrClientSDK_Init(char * ver, NrClientSDK_Event_Callback eventCb)`

参数： ver [out] 返回的 SDK 版本号
event_cb [in] 开发包事件回调函数
返回值： 0 成功，非零 失败值

2、增加一个网络读卡器

调用该接口可以增加一个网络读卡器，开发包连接到该读卡器后，会通过事件回调接口上报读取到的标签。

原型：`NRCLIENTSDK_API int __stdcall NrClientSDK_NewReader(char * devIp)`

参数： devIp [in] 网络读卡器的 ip 地址
返回值： 0 成功，非零 失败值

注：读卡器数量最多 100 个

3、删除一个网络读卡器

调用该接口可以删除一个已经存在的网络读卡器，删除以后，开发包不再通过事件回调接口

上报读取到的标签。

原型: `NRCLIENTSDK_API int __stdcall NrClientSDK_DeLReader(char * devIp)`

参数: devIp [in] 网络读卡器的 ip 地址

返回值: 0 成功, 非零 失败值

4、发送声光提醒请求

调用该接口可以向需要声光提醒的标签发送声光提醒请求, 网络读卡器收到消息后会通过事件回调接口上报请求结果。

原型: `NRCLIENTSDK_API int __stdcall NrClientSDK_CallLabel(char * devIp, unsigned long slot, unsigned long cardNo, unsigned long mode, unsigned long keepSeconds)`

参数: devIp [in] 网络读卡器的 ip 地址

slot [in] 分站号

cardNo [in] 需要提醒的标签 ID

mode [in] 提醒模式 0 绿灯闪 1 绿灯闪+声音提醒
2 红灯闪 3 红灯闪 + 声音提醒 4 红绿灯交替闪
5 红绿灯交替闪 + 声音提醒 255 关闭声光提醒

keepSeconds [in] 声光提醒保持的秒数

返回值: 0 成功, 非零 失败值

5、发送批量声光提醒请求

调用该接口可以向需要声光提醒的多个标签发送声光提醒请求, 网络读卡器收到消息后会通过事件回调接口上报请求结果。

原型: `NRCLIENTSDK_API int __stdcall NrClientSDK_BulkCallLabel(char * devIp, unsigned long slot, unsigned long keepSeconds, LNRCLIENTSDK_CALL_CARD callCard, unsigned long cardNum)`

参数: devIp [in] 网络读卡器的 ip 地址

slot [in] 分站号

keepSeconds [in] 声光提醒保持的秒数

callCard [in] 需要提醒的标签 ID 及提醒模式, 提醒模式参考”发送声光提醒请求接口参数 mode”

cardNum [in] 需要提醒的标签数量

返回值： 0 成功，非零 失败值

6、发送短消息文本通知请求

调用该接口可以向标签发送短消息文本请求，网络读卡器收到消息后会通过事件回调接口上报请求结果。

原型： `NRCLIENTSDK_API int __stdcall NrClientSDK_SmsNotify(char * devIp, unsigned long slot, unsigned long cardNo, unsigned long code, const char * content)`

参数：

devIp	[in]	网络读卡器的 ip 地址
slot	[in]	分站号
cardNo	[in]	需要提醒的标签 ID，当 ID 为 0xFFFFFFFF 时为广播消息
code	[in]	短信编号 (0 - 65535)
content	[in]	消息文本内容 (最长 110 字节)

返回值： 0 成功，非零 失败值

7、打开 TELNET 调试

打开 TELNET 调试后，使用 TELNET 客户端可以连接到 SDK 的 TELNET 调试服务，可以使用明文查询 SDK 内部信息。（命令行的详细指令另行发布）

原型： `NRCLIENTSDK_API int __stdcall NrClientSDK_OpenTelnetDebug(unsigned short port)`

参数：

port	[in]	调试用的 TELNET 端口号
------	------	-----------------

返回值： 0 成功，非零 失败值

8、关闭 TELNET 调试

关闭 TELNET 调试后，将释放 TELNET 调试服务所占用的资源，提高运行效率。

原型： `NRCLIENTSDK_API int __stdcall NrClientSDK_CloseTelnetDebug(void)`

参数： 无

返回值： 0 成功，非零 失败值

三、类型、事件、返回值定义

1、事件回调接口定义

原型: `typedef int (__stdcall *NrClientSDK_Event_CallBack)(int evt, int result, char * devIp, LNRCLIENTSDK_CARD card)`

参数: evt 事件类型, 参考[事件定义](#)
 result 函数异步执行返回的结果值参考[返回值定义](#)
 devIp 读卡器 IP 地址
 card 标签数据

2、标签结构体定义

原型:

```
typedef struct _NRCLIENTSDK_CARD
{
    unsigned long slot;     // 分站号
    unsigned long cardNo;   // 标签卡号
    unsigned char attr;     // 标签属性
    unsigned char extData[7]; // 扩展数据
    char rssi;             // 信号场强值
}NRCLIENTSDK_CARD, *LNRCLIENTSDK_CARD
```

3、声光提醒标签结构体定义

原型:

```
typedef struct _NRCLIENTSDK_CALL_CARD
{
    unsigned long cardNo;   // 标签卡号
    unsigned char mode;     // 提醒模式
}NRCLIENTSDK_CALL_CARD, *LNRCLIENTSDK_CALL_CARD
```

4、事件定义

值	定义	说明	参数 :	参数 : devIp	参数 : card
---	----	----	------	------------	-----------

			result		
1	EVENT_CONNECT_OK	成功连接到读卡器	无	读卡器 IP 地址	无
3	EVENT_DISCONNECT	读卡器断开连接	无	读卡器 IP 地址	无
4	EVENT_CARD_READ	读卡器读到标签	无	读卡器 IP 地址	读取到的标签信息
5	EVENT_CALL_ACK	读卡器确认声光提醒	结果值	读卡器 IP 地址	呼叫的标签信息仅分站号和标签卡号有效
6	EVENT_SMS_ACK	读卡器确认短信通知	结果值	读卡器 IP 地址	接收短信的标签信息仅分站号和标签卡号有效
7	EVENT_BULK_CALL_ACK	读卡器确认批量声光提醒	结果值	读卡器 IP 地址	无

5、返回值定义

值	定义	说明
0	NRCLIENTSDK_OK	成功
1	NRCLIENTSDK_ERR_READER_NOT_CONNECT	读卡器未连接
2	NRCLIENTSDK_ERR_READER_TIMEOUT	超时
3	NRCLIENTSDK_ERR_READER_ALREADY_EXIST	读卡器已存在
4	NRCLIENTSDK_ERR_READER_READER_NOT_EXIST	读卡器未存在
5	NRCLIENTSDK_ERR_SDK_NOT_INIT	开发包未初始化
6	NRCLIENTSDK_ERR_PARAM_WRONG	参数错误
7	NRCLIENTSDK_ERR_COUNT_REACH_MAX	数量达到最大
8	NRCLIENTSDK_ERR_SLOT_NOT_EXIST	分站未连接
9	NRCLIENTSDK_ERR_READER_BUF_FULL	读卡器缓存满