通讯 API 使用说明

一、打开PCI-114设备

int sio_open(void);

功能：打开PCI-114设备、并且初始化设备。是一切通讯操作的前提条件。

函数入口：无。

函数出口：整型数。

0 ：打开成功；

-1：打开失败。

二、设备初始化

int sio_device_init(void);

功能：初始化设备，默认由sio_open()内部调用。用户也可以在运行中调用来再次初始化设备。

函数入口：无。

函数出口：整型。

0:成功。

-1：其它错误；

-2：通讯口1事件创建错误；

-3：通讯口1线程创建错误；

-4：通讯口2事件创建错误；

-5：通讯口2线程创建错误；

-6：通讯口3事件创建错误；

-7：通讯口3线程创建错误；

-8：通讯口4事件创建错误；

-9：通讯口4线程创建错误。

三、读通讯口数据

int sio_read(int comport,unsigned char *rbuf);

功能：从通讯口接收缓冲中读取已经接收到的数据。

函数入口：comport:类型为整型数据，范围1～4，对应通讯口1～4。

Rbuf：无符号字符型数据指针；用户开辟的缓冲区的指针，用以接收所读的通讯口数据。

函数出口：整型。

-1：读错误；

≥0:为读取数据的字节数。

四、通讯中断服务程序安装函数

int sio_isr_install(int comport,LPCALLBACKFUNC lpFunc);

功能：为通讯控制函数安装中断服务程序(ISR)，在ISR中调用sio_read()函数可以读取本次接收的数据。

函数入口：comport:类型为整型数据，范围1～4，对应通讯口1～4。

LpFunc：ISR的函数指针。LpFunc的函数原型为：

Void (*LpFunc)(void);

函数出口：整型。

0:成功。
-1：其它错误；
-2：通讯口1事件创建错误；
-3：通讯口1 ISR安装错误；
-4：通讯口2事件创建错误；
-5：通讯口2 ISR安装错误；
-6：通讯口3事件创建错误；
-7：通讯口3 ISR安装错误；
-8：通讯口4事件创建错误；
-9：通讯口4 ISR安装错误。

五、RS485方向控制函数

int sio_set_485_dir(int comport,unsigned char in_out);

功能：在通讯口配置为RS485接口时，控制RS485的方向。

函数入口：comport:类型为整型数据，范围1～4，对应通讯口1～4。
in_out：类型为无符号字符，0xff表示方向为发送，0x00表示方向为接收。

函数出口：整型。
0：成功；
-1：失败。

六、DTR信号控制函数

int sio_set_DTR(int comport,UCHAR in_out);

功能：设置数据终端就绪（DTR）信号的状态。

函数入口：comport:类型为整型数据，范围1～4，对应通讯口1～4。
in_out：类型为无符号字符， 0xff高电平，0x00低电平。

函数出口：整型。
0：成功；
-1：失败。

七、RTS信号控制函数

int sio_set_RTS(int comport,UCHAR in_out);

功能：设置请求传送数据（RTS）信号的状态。

函数入口：comport:类型为整型数据，范围1～4，对应通讯口1～4。
in_out：类型为无符号字符， 0xff高电平，0x00低电平。

函数出口：整型。
0：成功；
-1：失败。

八、通讯位协议设置

int sio_set_parity(int comport,enum par_code parity,int stopbit,int databit);

功能：设置串行通讯硬件协议。

函数入口：comport:类型为整型数据，范围1～4，对应通讯口1～4。
parity：类型为par_code枚举型。
COM_NONE表示无校验，COM_EVEN表示奇校验， COM_ODD表示偶校验。
Stopbit：类型为整型数据。停止位个数，值为1～2；

Databit：类型为整型数据。数据位个数，值为5～8。

函数出口：整型。

0：成功；

-1：失败。

九、通讯波特率设置函数

int sio_set_baudrate(int comport,int band);

功能：设置通讯口的波特率。

函数入口：comport:类型为整型数据，范围1～4，对应通讯口1～4。

band：类型为整型数据，值和波特率对应如下：

值	波特率
0	300
1	600
2	1200
3	2400
4	4800
5	9600
6	19200
7	38400
8	56000
9	128000
10	256000
11	512000。

函数出口：整型。

0：成功；

-1：失败。

十、字符发送函数

int sio_send_char(int comport,unsigned char cc);

功能：发送单个字符数据。

函数入口：comport:类型为整型数据，范围1～4，对应通讯口1～4。

cc：类型为无符号字符。欲发送的字节数据。

函数出口：整型。

0：成功；

-1：失败。

十一、字符串发送函数

int sio_send_string(int comport,unsigned char *str,int len);

功能：发送字符串或以字节为单位批量发送数据。

函数入口：comport:类型为整型数据，范围1～4，对应通讯口1～4。

str：类型为无符号字符缓冲指针。存放欲发送的字符串数据。

Len: 整型数据， 字符串长度。

函数出口：整型。

0：成功；

≥0:为实际发送数据的字节数。

十二、关闭设备函数

void sio_close(void);

功能：关闭设备，销毁相关的对象句柄。

函数入口：无。

函数出口：无。

十三、例子

演示中断服务程序的安装及使用。

```
#include "sercomm.h" //必须包含此头文件
#define MAX_REC_BUFF_SIZE    512  //缓冲区大小
//-----------------------------------------------
//接收数据的处理
char rbuff1[MAX_REC_BUFF_SIZE]; //接收缓冲区
//-----------------------------------------------void SER_ISR1(void) //中断服务程
序
{
 int len=0;
 AnsiString ss1;
 AnsiString temp;
 int i=0;
 memset(rbuff1,0x00,MAX_REC_BUFF_SIZE);
 len=sio_read(1, rbuff1);
 if(len>=0){
// 数据处理
…
}
}
void DeviceOpen (void)//设备初始化的函数
{
  int rst=0;
  rst=sio_open();//设备打开并初始化
  if(rst!=0) {
   //错误处理
 }
  sio_set_485_dir(1,0x00);  //方向为接收，仅当 RS485 时调用。

}
//-----------------------------------------------

//C++builder 下发送按钮处理函数，发送 Memo 控件中的数据
void __fastcall TCommForm::BtnSendClick(TObject *Sender)
{
 AnsiString str;
 unsigned char buff[512];
 unsigned char buff1[512];
 unsigned char *ptrs=NULL;
```

```
 int slen=0;
 if(DeveiOpened){
     slen=(MemoSend->Lines->Text.Length()+1)/3;
    memcpy(buff1,MemoSend->Lines->Text.c_str(),MemoSend->Lines->Text.Length());
     ptrs=buff1;
     for(int i=0;i<slen;i++){
      sscanf(ptrs,"%02x ",&(buff[i]));
      ptrs+=3;
     }
     if(!PCOM1_Is_485) sio_send_string(1,buff,slen);
     else{
       sio_set_485_dir(1,0xff);   //send
       sio_send_string(1,buff,slen);

     }
     PCOM1_Tra_Cnt+=slen;
     StaticTXCount->Caption="TX:"+IntToStr(PCOM1_Tra_Cnt);
  }
 }
 else{
  ShowMessage("请先打开设备！");
 }
}
//---------------------------------------------
```