

SE18E 以太网转八路并行串口服务器

用户手册

版本 1.8 2017 年 1 月
深圳市南彬电子有限公司

版本信息

版本号	修改时期	修改内容	修改人
V1.0	2016.2	初始文档	R&D
V1.2	2016.3	串口设置改为 UDP	R&D
V1.6	2016.10	优化 TCP/IP 协议栈	R&D
V1.8	2017.1	优化 Rx 缓存	R&D

SE18E 以太网转八路串口服务器用户手册

1. 产品概述

SE18E 采用 32 位 1G MIPS 高速并行处理器，无中断和操作系统，八路串口并行运行，每路独立收发数据，能保证每路数据的实时和准确。

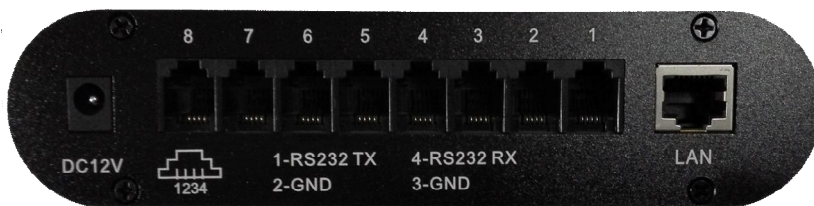
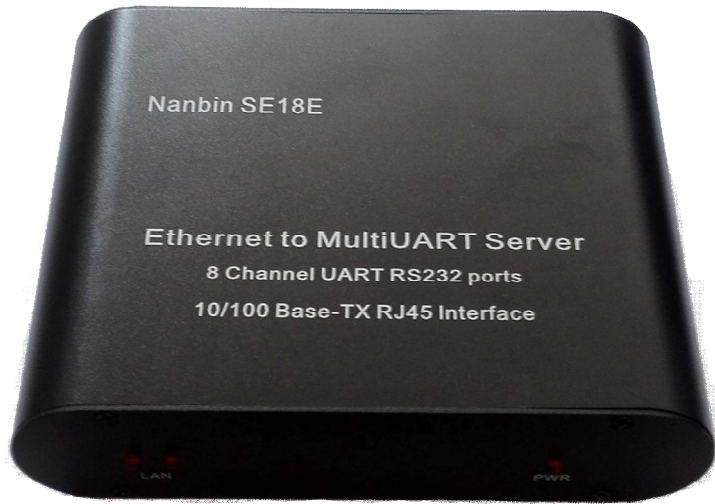
SE18E 的 TCP/IP 协议栈针对 RS232、RS485 串口通信模式优化，并有流量控制功能，能保证在工业环境下的准确传输。

SE18E 的上位机编程可采用 socket 库函数和虚拟串口两种模式。socket 库函数编程非常简单，源码开放、稳定、跨平台，通用性好。

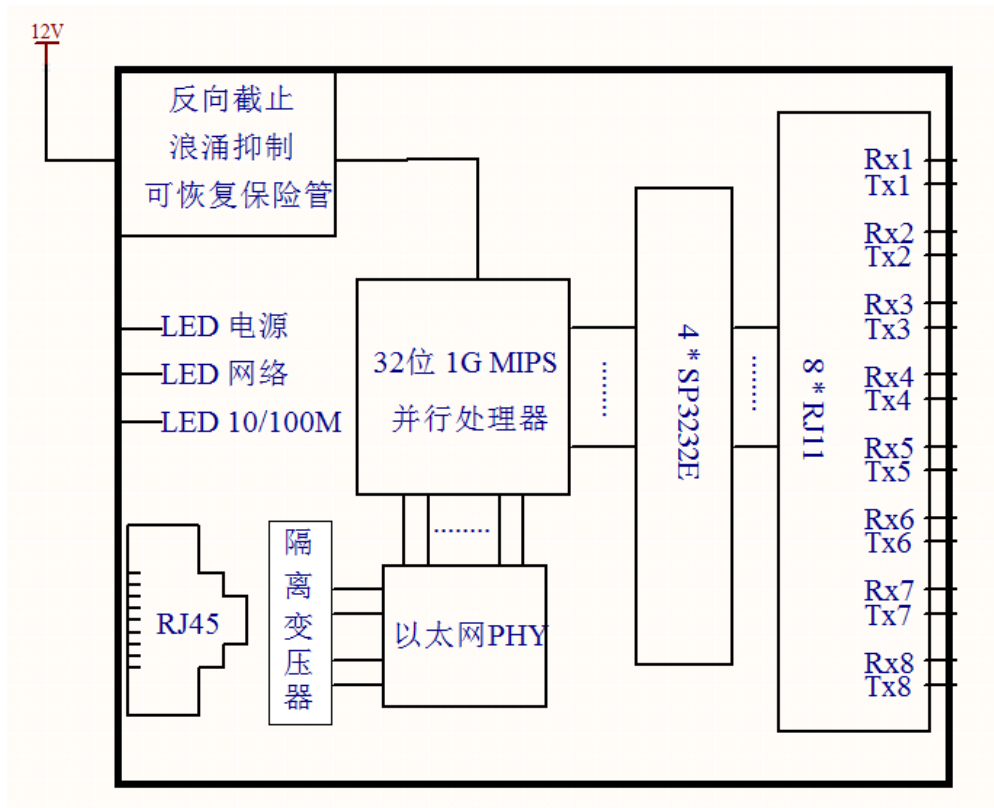
1.1 产品特点

- ◆ 局域网可挂接任意多个本设备（缺省设定为 5 个）
- ◆ 支持 DHCP 自动分配 IP 地址，也可远程设定固定 IP
- ◆ 上位机自动搜索接入设备，即插即用
- ◆ 上位机可远程设置设备的 IP 地址、串口配置和设备标识符
- ◆ 上位机可通过“IP+端口”方式与设备任意一个串口双向通讯
- ◆ 上位机也可用 Webserver 设置串口配置
- ◆ 上位机采用库函数编程。库函数采用基本 socket 编写，开源，Windowst 各本版本/Linux 通用，稳定、简单易用。
- ◆ 如需虚拟串口，请联系客服。
- ◆ 串口一可输入控制字符以恢复 DHCP 动态 IP，防止错误 IP 地址造成死锁
- ◆ 稳定可靠的自有产权 TCP/IP 协议栈，针对串口通讯优化
- ◆ 可提供上位机 MFC 编程测试程序参考代码
- ◆ 设定固定 IP 地址可接受外网访问。请参考端口映射或外网穿透的实现方法

1.2 产品外观



1.3 功能框图



1.4 产品配套

- ◆ Windows 编程库函数
- ◆ Windows 测试程序（用于测实际波特率和误码率）

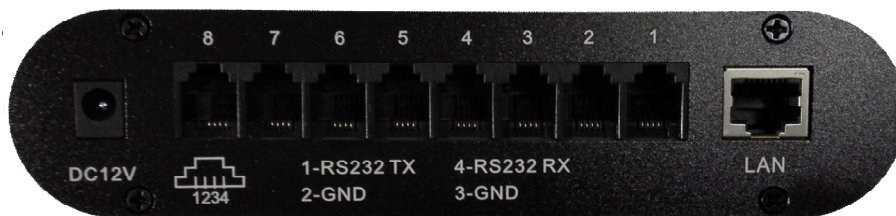
2. 产品规格

规格	参数
处理器	32 位 1G MIPS 并行处理器
网 口	一路 10/100 以太网口 RJ45
	内置 1500V 隔离变压器
串 口	八路并行 RS232 串口 全双工 无 RTS、CTS 等
	SP3232E 接口芯片 15KV ESD 短路保护
	最大双工波特率 56KB (注 1)
	八个 RJ11 母座接口 四针
工作电源	12V (注 2) 内置反向及瞬态过压保护、可恢复保险管
电源功耗	< 5W
存贮温度	-45° C 至 85° C
工作温度	-20° C 至 85° C
工作湿度	0 至 90%
外观尺寸	165*130*40mm

注 1: 本产品出厂均经八路并行 56KB 收发三小时无误码检测合格

注 2: 输入电压 5 (最低) 至 16V, 推荐 12V@1A。本产品不配电源适配器, 敬请注意

3. 硬件接口



- RS232 串口采用 4P RJ11，连接牢固易于操作，8K/15KV ESD 静电保护。

连线为：（顶视图）

- . 左 1 — RS232 Tx
 - . 左 2 — GND
 - . 左 3 — GND
 - . 左 4 — RS232 Rx
- 100M 以太网口采用 8P RJ45，内置隔离变压器。
 - DC5 电源插座，12V 输入，内置反向及瞬态过压保护、可恢复保险管。

4. 流量控制

- 本设备通过 UDP 设置或读取设备的参数。
- 本设备通过 TCP 从上位机接收数据并经串口 Tx 发送。TCP 接收窗口会暂时关闭直至 Tx 数据发送完毕，以保证 TCP 数据不会溢出。

如上机持续发送数据，应避免数据堵塞。请考虑以下建议：

- (1) 每 1K 字节发送速率应不高于串口波特率的 1/10(按 8 位字长计算)
 - (2) 适当增加 socket 的发送缓存 (Windows 一般默认值是 8KB)
 - (3) 适当延长 send() 的 timeout 时间
- 每路串口的 Rx 缓存 10KB。在 56K 波特率下相当于可承受 1 秒多的断网时间。
 - 串口 Rx 触发 TCP 发送的条件：
 - (1) Rx 缓存达到 400 字节；
 - (2) 或 200ms 没有收到新的数据
 - 串口 Rx 缓存溢出，会在 Rx 数据流中插入“overflow”字符串。
 - 如需要在串口数据流中加入 XON/XOFF 握手协议，请与客服联系。

5. TCP/UDP 端口占用

	SE18E	上位机	工作模式	功 能
UDP	15534	15533	广播/点对点	搜索/IP/UID
UDP	23	Any	点对点	串口配置
TCP	46~53	Any	点对点	串口通讯

- ◆ 上位机 UDP 广播地址 : 192.168.1.255
S2E 设备 UDP 广播地址: 255.255.255.255
- ◆ TCP 端口与 S2E 串口的对应关系:
46—串口 0 53—串口 7

6. UID 和固定 IP

- ◆ 上位机可为每一台 S2E 设备设定一个十个中文字符的 UID 用于识别设备。需确保每一个 UID 是唯一的。
- ◆ 上位机可为每一台 S2E 设备设定固定 IP 用于识别设备或用于端口映射/外网穿透。需确保固定 IP 不能与网络中其它设备 IP 地址一样；也必须确保固定 IP 地址网段是正确的。如把 192.168.1.xxx 设成 168.192.1.xxx 则 S2E 永远不能上网。如错误发生，可用串口向 S2E 串口一发送“S2EUartResetToZero!@”字符串，即可将设备恢复至 DHCP 自动分配 IP 地址。

7.Windows 编程库函数

7.1 库函数简介

- Windows 库函数采用 socket 基本编程，简单、稳定，通用性好。
- 包括以下函数：

序号	函数名	功 能
1	S2E_WSASStartup()	打开 WinSock 库
2	S2E_WSACleanup()	关闭 WinSock 库
3	S2E_SearchBoard()	搜索局域网中的 SE18E 设备
4	S2E_SetIPAddr()	设置 SE18E 设备 IP 地址
5	S2E_SetUID()	设置 SE18E 设备 UID
6	S2E_GetUartConfig()	读取 SE18E 设备一路串口的配置
7	S2E_GettingAllUart()	读取 SE18E 设备全部八路串口的配置
8	S2E_SetUartConfig()	设置 SE18E 设备一路串口的配置
9	S2E_SettingAllUart()	设置 SE18E 设备全部八路串口的配置
10	S2E_SettingAllUart()	将串口配置存入 flash

7.2 上位机与串口的数据通讯

在调用以上库函数搜索到 SE18E 设备并完成 IP、UID 和串口配置后，即可用 socket 的 send() 和 recv() 直接读/写一个串口。

7.3 编程例子

(1). 搜索设备

```
#include "S2EWinLib.h"

Int          result;

S2E_BOARD_INFO  board[MaxBoardNum];

result=S2E_SearchBoard(m_BoardInfo, MaxBoardNum);
```

说明:

Result 返回搜索到的 SE18E 设备数;

SE18E 的 IP 地址和 UID 包括在 boad[i]中, 其中:

- boad[i]. IPaddr 是 IP 地址
- boad[i]. ID 是 UID 字符串
- boad[i]. Active==1 表示设备有效

(2). 设置固定 IP 地址

```
#include "S2EWinLib.h"
#include <string. h>
int     result;
int     ifound;
char *   target="风机一号";
struct  in_addr curIP;
struct  in_addr newIP;
    //如果新 IP 地址是全 0, 则恢复 DHCP
newIP.S_un.S_addr = inet_addr("192.168.1.109");
    //取得目标设备 IP 地址, 也可用 IP 地址确定目标设备
ifound=0;
for(int i=0;i< MaxBoardNum;i++)
{
    if (boad[i]. Active==1 )
    {
        if(strcmp (target,board[i]. ID) ==0)
        {
            ifound=i;
            curIP=boad[i]. IPAddr.sin_addr;
            Break;
        }
    }
}
if(0==ifound) return 0;
S2E_SetIPAddr(curIP, newIP);
```

(3). 设置设备新 UID

```
#include "S2EWinLib.h"
struct in_addr  curtIP;
char *          pNewID="1 栋 302 室";
struct in_addr  curIP;

curIP = boad[i]. IPaddr.sin_addr;
S2E_SetUID(curIP, pNewID, strlen(pNewID));
```

(4). 设置串口配置

```
#include "S2EWinLib.h"
struct          in_addr  curtIP;
S2E_UARTCONFIG UartSetting;
struct in_addr  curIP;

curIP = boad[i]. IPaddr.sin_addr;
UartSetting. Parity=0;          //无奇偶校验
UartSetting. stop_bits=0;      //一个停止位
UartSetting. Baud=9600;        //波特率
UartSetting. char_len=8;       //字长为 8

S2E_SettingUart(curIP, 0, &UartSetting);
```

说明: 参数的取值范围:

- ◆ Parity (奇偶校验)
 - 0 (None/无), 1 (Odd/偶),
 - 2 (Even/奇), 3 (Mark), 4 (Space)
- ◆ stop_bits (停止位数)
 - 0 (一个停止位), 1 (二个停止位)
- ◆ Baud (波特率)
 - 150, 300, 600, 1200, 2400, 4800, 7200,
 - 9600, 14400, 19200, 28800, 38400, 57600, 115200
- ◆ char_len (字长)
 - 5, 6, 7, 8, 9

(5). 串口读写实例

```
struct in_addr    curIP;
SOCKADDR_IN      addrsockC;
Socket           mySock;
int              result, timeout;
char *           pSendstr=" this is firt test" ;
char             RecvBuf[600];
int              Uartchannel;
int              iLenOfRecv, iLenOfSend;

mySock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
timeout = 1000; //设置收发 timeout
result = setsockopt(mySock, SOL_SOCKET, SO_SNDTIMEO,
(char*)&timeout, sizeof(timeout));
result = setsockopt(m_socket, SOL_SOCKET, SO_RCVTIMEO,
(char*)&timeout, sizeof(timeout));
int disable = 0;
int len = sizeof(disable); //可以打开或关掉心跳包
result = setsockopt(m_socket, SOL_SOCKET, SO_KEEPALIVE,
(char*)&disable, len);

addrsockC.sin_family = AF_INET;
addrsockC.sin_port = htons(Uartchannel + 46);
addrsockC.sin_addr = curIP;

int nret = connect(m_socket, (sockaddr*)&addrsockC,
sizeof(addrsockC));

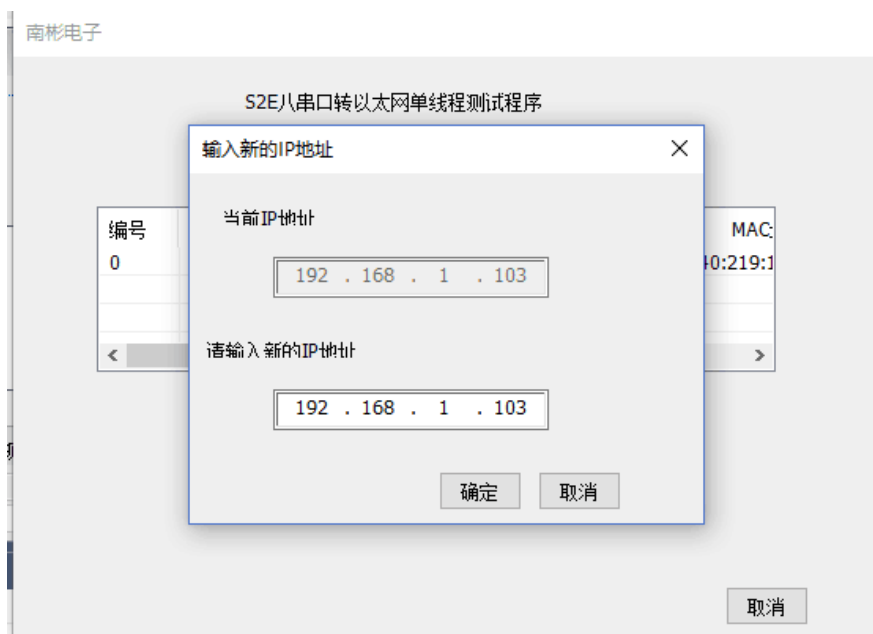
iLenOfSend= send(mySock, pSendstr, strlen(pSendstr), 0);
Sleep(100);
iLenOfRecv = recv(mySock, RecvBuf, 600, 0);
```

说明:

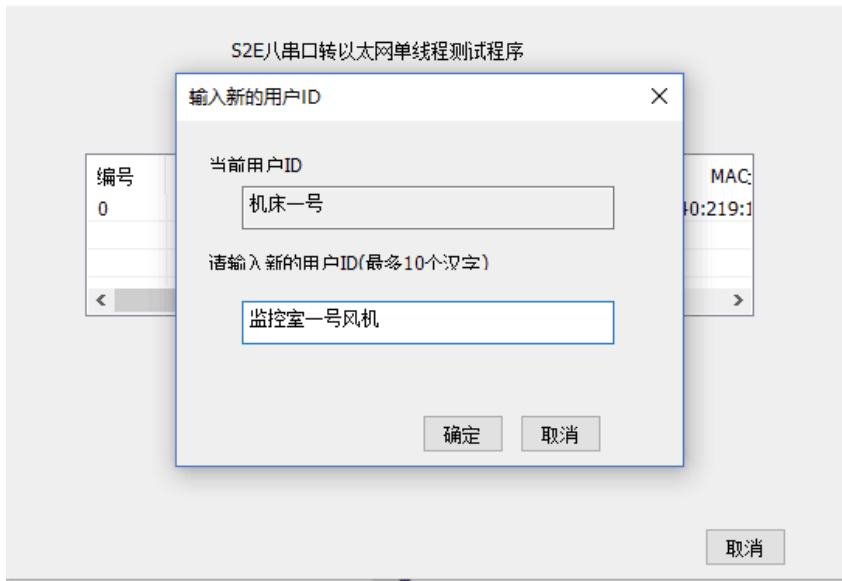
- ◆ Send() timeout 事件发生时，数据也有可能发送出去，可适当延长 timeout 时间以避免 timeout 事件发生；
- ◆ 测式软件的多线程方式可供参考；

8.测试软件图例

南彬电子



南彬电子



串口参数

×



异或，英文为exclusiveOR，或缩写成XOR异或（xor）是一个数学运算符。它应用于逻辑运算。异或的数学符号为“ \oplus ”，计算机符号为“xor”。其运算法则为： $a\oplus b=(a\wedge b)\vee(a\wedge\neg b)\vee(\neg a\wedge b)$ 。异或的结果是真，假异或真的结果是假，假异或假的结果是真。就是说两个值不相同，则异或结果为真。反之，为假。不同为1，相同为0。异或也叫半加运算，其运算法则相当于不带进位的二进制加法：二进制下用1表示真，0表示假，则异或的运算法则为： $0\oplus 0=0, 1\oplus 0=1, 0\oplus 1=1, 1\oplus 1=0$ （同知，异或为1），这些法则与加法是相同的，只是不带进位。异或顺序中有二种运算符：XOR、XOR、 \oplus 。使用方式如下： $7=7\oplus x\oplus x$

文章总字节数是 599 字节
最大测试长度600字节

5E 流控驱动串口异码率测试

串口0	串口1	串口2	串口3	串口4	串口5	串口6	串口7
测试长度 :599 字节	测试长度 :599 字节	测试长度 :599 字节	测试长度 :599 字节	测试长度 :599 字节	测试长度 :599 字节	测试长度 :599 字节	测试长度 :599 字节
发送次数 :155030	发送次数 :155024	发送次数 :155008	发送次数 :155316	发送次数 :154999	发送次数 :153886	发送次数 :153600	发送次数 :155001
接收次数 :154920	接收次数 :154914	接收次数 :154898	接收次数 :153406	接收次数 :154889	接收次数 :153476	接收次数 :153490	接收次数 :154891
总字节数 :92797	总字节数 :92793	总字节数 :92783	总字节数 :91890	总字节数 :92778	总字节数 :91932	总字节数 :91940	总字节数 :92779
测试用时 :271分钟	测试用时 :271分钟	测试用时 :271分钟	测试用时 :271分钟	测试用时 :271分钟	测试用时 :271分钟	测试用时 :271分钟	测试用时 :271分钟
错误字节 :0	错误字节 :0	错误字节 :0	错误字节 :0	错误字节 :0	错误字节 :0	错误字节 :0	错误字节 :0
波特率(%) :57600	波特率(%) :57600	波特率(%) :57600	波特率(%) :57600	波特率(%) :57600	波特率(%) :57600	波特率(%) :57600	波特率(%) :57600
波特率(实) :56979	波特率(实) :56977	波特率(实) :56975	波特率(实) :56426	波特率(实) :56971	波特率(实) :56452	波特率(实) :56457	波特率(实) :56972

连接成功

连接成功

连接成功

连接成功

连接成功

连接成功

连接成功

连接成功

启动测试

停止测试

取消