EIO 远程 IO 联网产品技术配置手册

(Ver2.9)

一、 技术特点及应用方式

1、 EIO 技术特点

EIO 是同时集成 I/O 控制和 RS232/RS485 串口服务器的以太网设备，同时具备开关量输出、开关量采集、串口服务器等功能，可同时替代 I/O 卡和串口服务器。

支持 Socket、虚拟串口两种用户通讯接口，用户可以按照 Socket 标准，通过 TCP/IP 连接与 EIO 进行通讯。也可以通过 VSPM 虚拟串口软件，将 EIO 虚拟成普通串口设备，可以有效的降低软件编写难度。EIO 采用国际通用的 Modbus TCP 作为通讯协议，可以与各类组态软件无缝结合。

首创 EIO Link 技术，互联 2 个 EIO 设备，可以将远程的开关量状态传输到控制中心，并完整重现，控制中心的开关量状态也可以传输到现场，2 个 EIO 的串口也可以进行透明数据传输，整个控制过程无需电脑，完全由 2 个 EIO 实现。

EIO 设备具备光电隔离、ESD 防护等多用防护措施，可以稳定的工作在恶劣环境中。

二、 EIO Modbus TCP、RTU 报文详解

1、 Modbus TCP 简介

Modbus TCP 是在 Modbus 协议基础上所发展而来，目的是为了使 Modbus 更好的在以太网&TCP/IP 环境下进行传输，Modbus TCP 保留了 Modbus 的全部功能，并扩展了一些数据结构。

1）Modbus 报文格式

Modbus TCP 报文

传输 ID	数据长度	子设备 ID	功能码	Modbus 数据区
5 字节	1 字节	1 字节	1 字节	EIO 使用最大 128 字节

Modbus RTU 报文

设备地址码	功能码	Modbus 数据区	CRC16 校验
1 字节	1 字节	EIO 使用最大 128 字节	2 字节

- 传输 ID

  Modbus TCP 有效，用户指定的传输 ID，默认为全 0。

- 数据长度

  Modbus TCP 有效，包括子设备、功能码和有效 Modbus 数据的以字节为单位的数据长度。

- 子设备 ID 或设备地址码

  设备地址码。

- 功能码

  Modbus 功能码。

- Modbus 数据区

  有效的 Modbus 数据，包括寄存器地址、寄存器偏移参数或控制参数、写

出数据等。

- CRC16 校验

  Modbus RTU 有效，为 2 个字节的 CRC16 校验码。

2）功能码

EIO 支持的功能码（Modbus TCP、RTU），表中数据均用 16 进制表示。

功能码	寄存器地址	寄存器偏移参数或控制参数	说明
0x01	0x0A	要返回输入状态的路数。	读开关量/TTL 输入状态，返回 8bit 数据，用来表示 8 路输入状态，0-为断开或高电平，1-为接通或低电平。
0x02	0x13	同上	同上
0x03	0x14	1-255，对应 1-255 路 ADC 转换数据。	读保持寄存器,对于带 ADC 转换的 EIO 有效,不带 ADC 功能 EIO,一直返回 0。
0x05	0x1E-0x25 每个地址对应一个输出	开关量/TTL 输出控制 接通或低电平：FF 00 断开或高电平：00 00	单独设置一个开关量/TTL 输出状态。
0x0F	0x64	要设置的输出开关量/TTL 电平输出数量。 对于 8 端口设备，此数值为 1-8。	设置一组线圈(开关量)状态 接收到的 2(或更高)字节数据的数据位对应开关的输出控制状态，0-为断开或高电平，1-为接通或低电平。

3）命令执行方式

上位机一次性发送上述的 Modbus TCP、RTU 报文，EIO 接收到报文后，如果在指定时间没有收到新数据，EIO 开始分析报文并执行相应功能，如果接收到的是错误报文，EIO 将功能码高位置 1，然后原样返回报文。

4）Modbus TCP、RTU 参数说明

寄存器及寄存器偏移参数或控制参数为 16bit 数据（2 字节），并且低位字节在后高位字节在前。

2、 EIO Modbus TCP、RTU 报文

1) 读开关量/TTL 输入状态

功能码：0x01

地址：0x0A

参数：1-EIO 支持的最大开关量路数

A) Modbus TCP

读开关量输入报文：

传输 ID	数据长度	子设备 ID	功能码	寄存器地址	寄存器地址偏移
00 00 00 00 00	06	01	01	00 0A	00 01

读 8 路开关量输入命令：

发送报文：00 00 00 00 00 06 01 01 00 0A 00 01

返回报文：00 00 00 00 00 04 01 01 01 42

返回报文说明：

传输 ID	数据长度	子设备 ID	功能码	返回数据个数	返回数据
00 00 00 00 00	04	01	01	01	42

B) Modbus RTU

读开关量输入报文：

设备地址	功能码	寄存器地址	寄存器地址偏移	CRC16 校验
01	01	00 0A	00 01	DD C8

发送报文：01 01 00 0A 00 01 DD C8

返回报文：01 01 01 42 A0 14

返回报文说明：

设备地址	功能码	返回数据个数	返回数据	CRC16 校验码
01	01	01	42	A0 14

C) 结果说明

返回数据个数指出共有多少个有效数据字节返回，8 端口设备用 1 个字节表示 8 路数据，如果的硬件输入端口为接通状态，那么对应的位被置 1，例如，返回数据为 0x42=0100 0010，说明 2、7 号输入通道为接通或低电平状态。

2) 读开关量/TTL 输出状态

功能码：0x01

地址：0x64

参数：1-EIO 支持的最大开关量输出路数

A) Modbus TCP

读开关量输出状态报文：

传输 ID	数据长度	子设备 ID	功能码	寄存器地址	寄存器地址偏移
00 00 00 00 00	06	01	01	00 64	00 01

读 8 路开关量输出状态命令：

发送报文：00 00 00 00 00 06 01 01 00 64 00 01

返回报文：00 00 00 00 00 04 01 01 01 D5

返回报文说明：

传输 ID	数据长度	子设备 ID	功能码	返回数据个数	返回数据
00 00 00 00 00	04	01	01	01	D5

B) Modbus RTU

读开关量输出状态报文：

设备地址	功能码	寄存器地址	寄存器地址偏移	CRC16 校验
01	01	00 64	00 01	BC 15

读 8 路开关量输出状态命令：

发送报文：01 01 00 64 00 01 BC 15

返回报文：01 01 01 D5 90 17

返回报文说明：

设备地址	功能码	返回数据个数	返回数据	CRC16 校验
01	01	01	D5	90 17

返回数据个数指出共有多少个有效数据字节返回，8 端口设备用 1 个字节表示 8 路数据，如果的硬件输出端口为接通状态，那么对应的位被置 1，例如，返回数据为 0xD5=1101 0101，说明 1、3、5、7、8 输出端口为接通状态（继电器接通）。输出状态为 EIO 保存的值，如果硬件损坏，这个值将无法反映输出的实际情况。

3） 读输入寄存器状态

功能码：0x02

地址：0x13

地址偏移：1

与功能码 0x01 完全相同

A） Modbus TCP

读开关量输入报文：

传输 ID	数据长度	子设备 ID	功能码	寄存器地址	寄存器地址偏移
00 00 00 00 00	06	01	01	00 13	00 01

读 8 路开关量输入命令：

发送报文：00 00 00 00 00 06 01 01 00 13 00 01

返回报文：00 00 00 00 00 04 01 01 01 42

返回报文说明：

传输 ID	数据长度	子设备 ID	功能码	返回数据个数	返回数据
00 00 00 00 00	04	01	01	01	42

B) Modbus RTU

读开关量输入报文：

设备地址	功能码	寄存器地址	寄存器地址偏移	CRC16 校验
01	01	00 13	00 01	0C 0F

发送报文：01 01 00 13 00 01 0C 0F

返回报文：01 01 01 42 A0 14

返回报文说明：

设备地址	功能码	返回数据个数	返回数据个数	CRC16 校验码
01	01	01	42	A0 14

结果说明

返回数据个数指出共有多少个有效数据字节返回，8 端口设备用 1 个字节表示 8 路数据，如果的硬件输入端口为接通状态，那么对应的位被置 1，例如，返回数据为 0x42=0100 0010，说明 2、7 号输入通道为接通或低电平状态。

4) 读保持寄存器

功能码：0x03

地址：0x15

地址偏移：1-模拟量最大输入路数

此寄存器只用于返回模拟量采集结果。

A) Modbus TCP

读保持寄存器报文：

传输 ID	数据长度	子设备 ID	功能码	寄存器地址	寄存器地址偏移
00 00 00 00 00	06	01	03	00 15	00 01

发送报文：00 00 00 00 00 06 01 03 00 15 00 01

返回报文：00 00 00 00 00 05 01 03 02 00 00

返回报文说明：

传输 ID	数据长度	子设备 ID	功能码	返回数据个数	结果数据 1	结果数据 2
00 00 00 00 00	05	01	03	02	00	00

B) Modbus RTU

读保持寄存器报文：

设备地址	功能码	寄存器地址	寄存器地址偏移	CRC16 校验
01	03	00 15	00 01	95 CE

发送报文：01 03 00 15 00 01 95 CE

返回报文：01 03 02 00 00 B8 44

返回报文说明：

设备地址	功能码	返回数据个数	结果数据 1	结果数据 2	CRC16 校验
01	03	02	00	00	B8 44

结果说明

返回数据个数指出共有多少有效数据返回，此命令返回数据为 16bit，所以用 2

个字节表示，如果设备不支持 ADC 功能的，总是返回 0。

返回结果为高 8 位字节在前，低 8 位在后，如果转换成 Word 类型数据，需要交换高低字节。

例如返回完整的数据报文(先收到 01，最后收到 A3)：01 03 02 01 A3

模拟量的值为：0x01A3=419

5) 单独设置一个开关量/TTL 输出状态

功能码：0x05

地址：0x1E 至 0x1E 加设备最大输出路数（0 起）。

A) Modbus TCP

设置第 8 路输出为接通或 TTL 为低电平报文：

传输 ID	数据长度	子设备 ID	功能码	寄存器地址	控制数据
00 00 00 00 00	06	01	05	00 25	FF 00

发送报文：00 00 00 00 00 06 01 05 00 25 FF 00

返回报文：00 00 00 00 00 06 01 05 00 25 FF 00

返回报文说明:

传输 ID	数据长度	子设备 ID	功能码	寄存器地址	控制数据
00 00 00 00 00	06	01	05	00 25	FF 00

设置第 8 路输出为断开/TTL 为高电平报文：

传输 ID	数据长度	子设备 ID	功能码	寄存器地址	控制数据
00 00 00 00 00	06	01	05	00 25	00 00

发送报文：00 00 00 00 00 06 01 05 00 25 00 00

返回报文：00 00 00 00 00 06 01 05 00 25 00 00

说明：

传输 ID	数据长度	子设备 ID	功能码	寄存器地址	控制数据
00 00 00 00 00	06	01	05	00 25	00 00

结果说明：EIO 原样返回接收到的控制报文。

B) Modbus RTU

设置第 8 路输出为接通或 TTL 为低电平报文：

设备地址	功能码	寄存器地址	控制数据	CRC16 校验
01	05	00 25	FF 00	9D F1

发送报文：01 05 00 25 FF 00 9D F1

返回报文：01 05 00 25 FF 00 9D F1

设置第 8 路输出为断开/TTL 为高电平报文：

设备地址	功能码	寄存器地址	控制数据	CRC16 校验
01	05	00 25	FF 00	DC 01

发送报文：01 05 00 25 00 00 DC 01

返回报文：01 05 00 25 00 00 DC 01

C) 结果说明

如果控制成功，EIO 原样返回接收到的控制报文。

6）设置多个开关量/TTL 输出状态

功能码：0x0F

地址：0x64

参数：1 至最大路数的输出

控制数据：数据位为 1，接通对应的开关量输出或设置 TTL 为低电平，数据位为 0，断开对应的开关量输出或设置 TTL 为高电平。

A）Modbus TCP

设置全部 8 路输出为接通/设置 TTL 为低电平报文：

传输 ID	数据长度	子 设 备 ID	功能码	寄存器地址	要控制的路数	控 制 数 据个数	控制数据
00 00 00 00 00	08	01	0F	64	00 08	01	FF

发送报文：00 00 00 00 00 08 01 0F 00 64 00 08 01 FF

返回报文：00 00 00 00 00 08 01 0F 00 64 00 08 01 FF

设置全部 8 路输出为断开/设置 TTL 为高电平报文：

传输 ID	数据长度	子设备 ID	功能码	寄存器地址	要控制的路数	控 制 数 据个数	控制数据
00 00 00 00 00	08	01	0F	64	00 08	01	FF

发送报文：00 00 00 00 00 08 01 0F 00 64 00 08 01 00

返回报文：00 00 00 00 00 08 01 0F 00 64 00 08 01 00

B) Modbus RTU

设置全部 8 路输出为接通/设置 TTL 为低电平报文：

设备地址	功能码	寄存器地址	要控制的路数	控 制 数 据 个数	控制数据	CRC16 校验
01	0F	64	00 08	1	FF	CF 1D

发送报文：01 0F 00 64 00 08 01 FF CF 1D

返回报文：01 0F 00 64 00 08 01 FF CF 1D

设置全部 8 路输出为断开/设置 TTL 为高电平报文：

设备地址	功能码	寄存器地址	要控制的路数	控 制 数 据 个数	控制数据	CRC16 校验
01	0F	64	00 08	1	00	8F 5D

发送报文：01 0F 00 64 00 08 01 00 8F 5D

返回报文：01 0F 00 64 00 08 01 00 8F 5D

C) 结果说明

如果控制成功，EIO 原样返回接收到的控制报文。

3、 EIO 与组态软件

EIO 支持 Modbus TCP、RTU 通讯协议，兼容各类组态软件，只要在组态软件中定义好设备的寄存器地址等参数就可以将 EIO 添加到系统中。

4、 EIO 的编程接口

1) Modbus TCP Socket 方式

EIO Modbus TCP 工作在 Server 模式，监听在：<EIO IP 地址：502 >端口。用户可以使用任何支持 Socket 的开发环境（Delphi、VC、VB、。net、GCC 等）来编写应用程序。

应用程序使用 Socket 标准与设备 Modbus TCP 端口建立 TCP/IP 连接，然后就可以通过此连接与设备进行通讯。

2) 虚拟串口方式

随机带的虚拟串口软件(VSPM)，可以将 EIO 虚拟成本机虚拟串口上的一个 RS232 设备。

在这种情况下，控制程序可以像操作普通 RS232 设备一样控制以太网控制器，而不再需要考虑网络部分。

控制方式：以太网控制器<--TCP/IP 协议-->OS<-->socket<-->VSPM(虚拟成 RS232 设备)<-->控制程序

3) EIO 的 RS232/RS485 方式

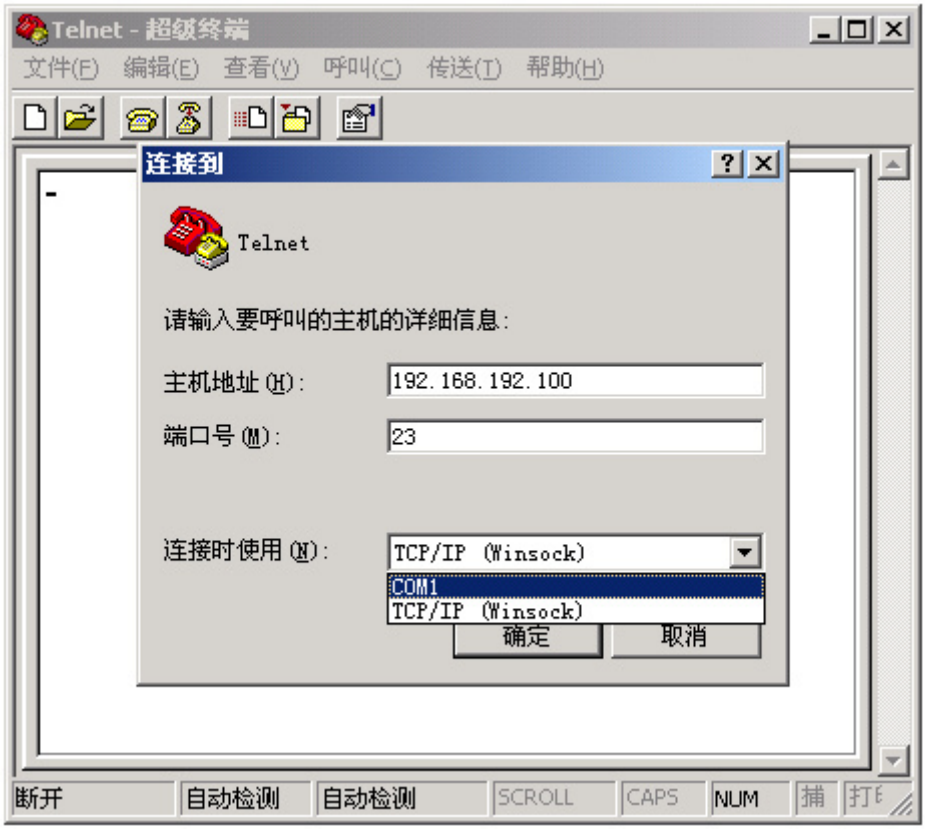
一些定制型号的 EIO 产品可以通过 RS232/RS485 端口进行控制通讯。请以说明书为准。

三、 EIO 管理方式

EIO 支持 Telnet 协议，可以通过 Windows 超级终端或 VSPM 虚拟串口集成的 Telnet 管理器来远程登录管理 EIO，也可以通过 EIO 的 RS232 管理口在本地管理 EIO。

1、 使用 Windows 超级终端进行参数管理

1）启动“Windows 超级终端”并建立一个新连接，这里用 Telnet 作为会话名称

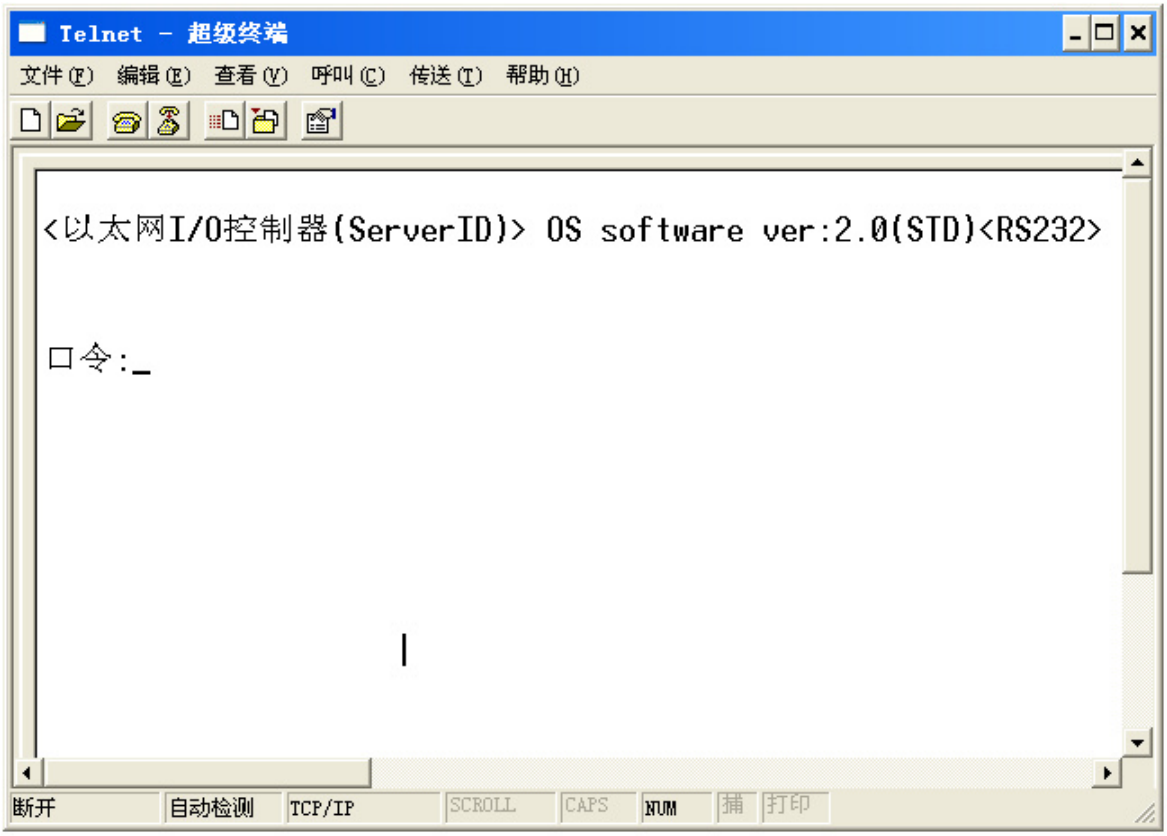


连接时使用项为：TCP/IP（Winsock）

设置好连接后，按“确定”，超级终端将建立与设备的 Telnet 连接

2）登录控制器

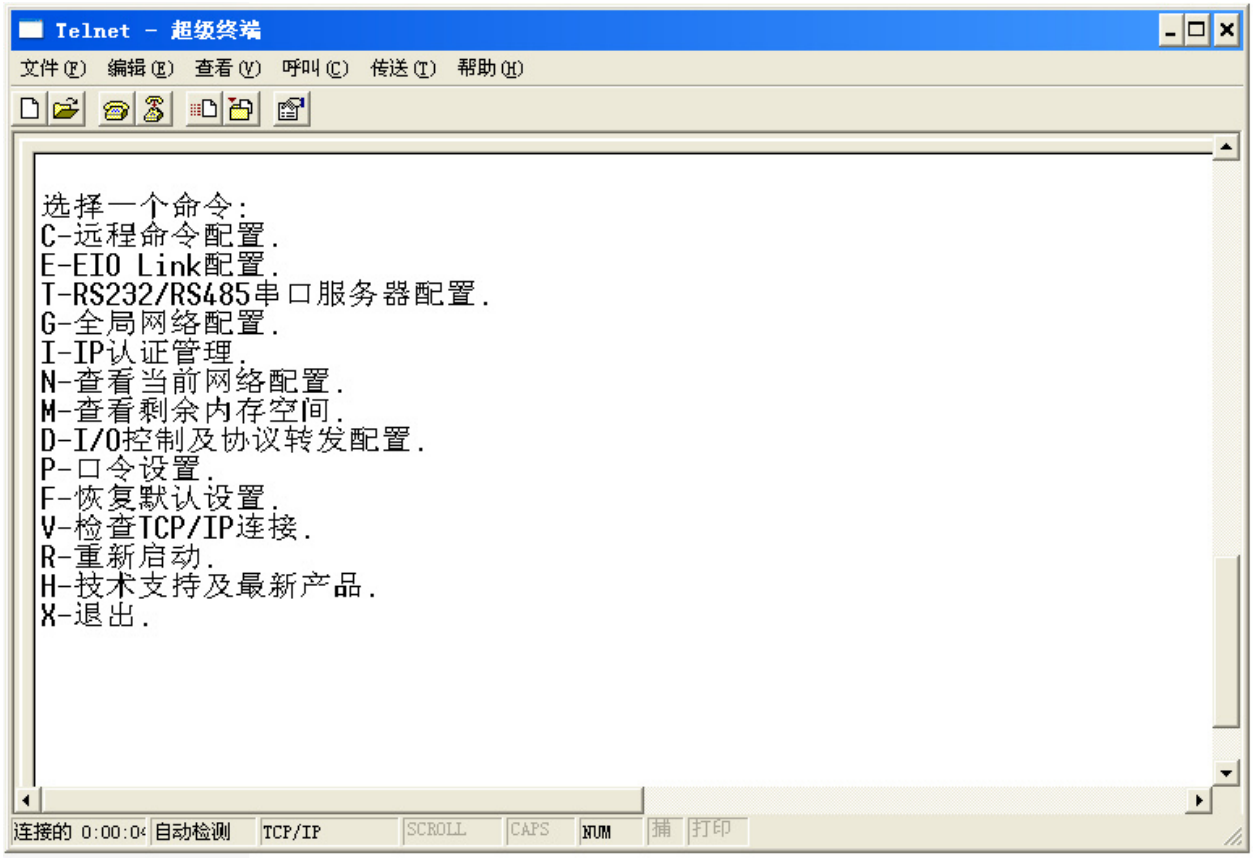
连接建立后，按空格键，将出现登录提示，然后输入管理员口令（默认为：admin）。



如果无法连接服务器，请检查是否存在下列问题

- 检查网络物理连接是否正常。
- 检查防火墙是否放行对外的 23 端口连接。
- 超级终端的主机地址、端口号是否正确。
- 如果开启了 IP 认证，本机的 IP 地址是否在认证表内。
- 如果遗失了控制器 IP 地址，可以使用 VSPM 虚拟串口的<设备探测器>搜索到设备。或者使用回复默认值按钮将设备回复到默认设置。

3）成功登录后，将出现下面的功能菜单。



第 1 个字母或数字为菜单选择键。

2、 使用 VSPM 虚拟串口软件的 Telnet 管理器

1）启动虚拟串口软件。

在设备探测器中选择一个 EIO 设备，选择<管理>-><使用 Telnet 管理此设备>。

2）Telnet 登录。

连接成功后，将出现登录提示，请输入管理员口令登录

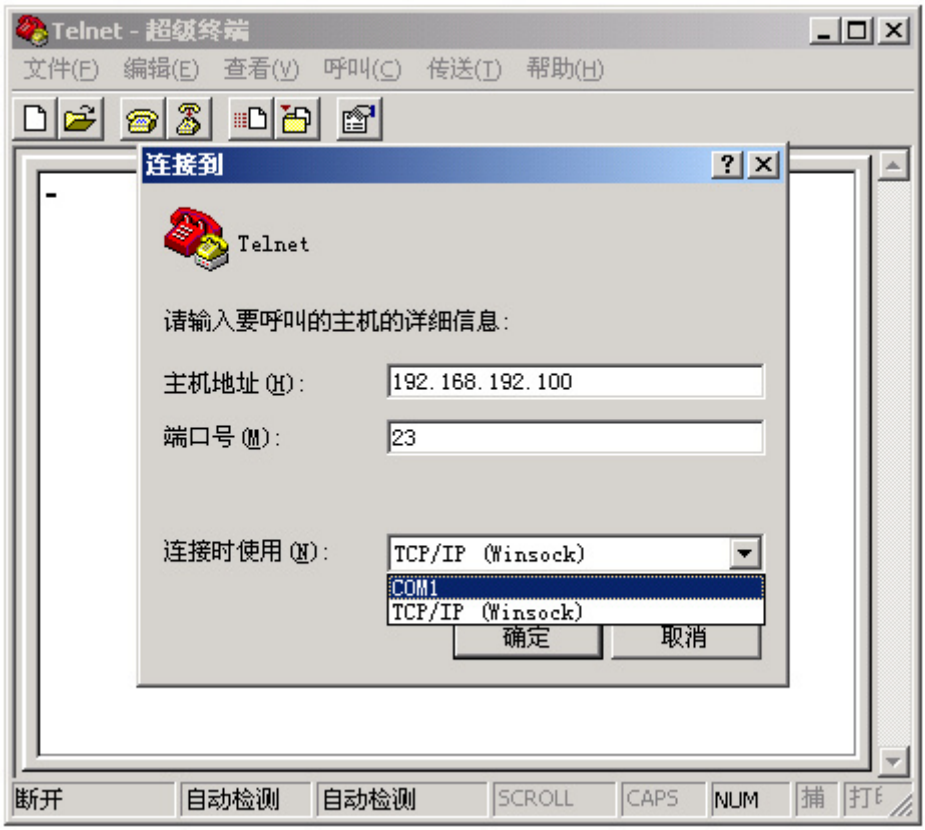
3）成功登录后的界面


Telnet管理
Telnet
主机地址：
192.168.192.100
主机端口：
23
重新连接
<以太网I/O控制器(ServerID)> OS software ver:2.2(EIO)<RS232>
a
口令:*****口
选择一个命令:
C-远程命令配置.
E-EIO Link配置.
T-RS232/RS485串口服务器配置.
G-全局网络配置.
I-IP认证管理.
N-查看当前网络配置.
M-查看剩余内存空间.
D-I/O控制及协议转发配置.
P-口令设置.
F-恢复默认设置.
V-检查TCP/IP连接.
R-重新启动.
H-技术支持及最新产品.
X-退出.

3、 使用 Windows 超级终端通过管理口进行管理

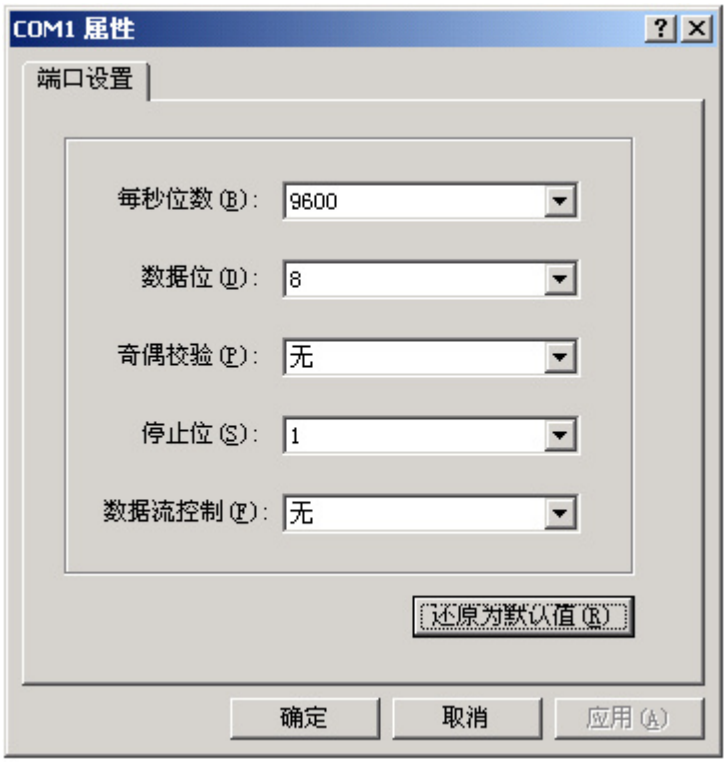
大部分型号的 EIO 产品支持一个 RS232 管理串口，可以通过管理口，在现场设置设备各种参数。管理口速率为 9600bps，无校验，8 数据位，1 停止位。

1）启动“Windows 超级终端”并建立一个新连接，这里用 Cfg 作为会话名称



连接时使用项为：COM1

然后按照下图配置串口



设置好连接后，按“确定”，然后执行呼叫菜单，超级终端将建立与设备管理口建立 RS232 连接。

2）登录 EIO

连接后，在超级终端内按任意键，会出现提示，按’1’键就可以进入登录提示。默认口令为 admin。

四、 控制器参数配置

1、 主菜单功能列表

下面两种主菜单的相同菜单项配置方法完全一样。

EIO-STD 支持以太网的 EIO 主菜单

C-远程命令配置。
E-EIO Link 配置。
T-RS232/RS485 串口服务器配置。
G-全局网络配置。
I-IP 认证管理。
N-查看当前网络配置。
M-查看剩余内存空间。
D-I/O 控制及协议转发配置。
P-口令设置。
F-恢复默认设置。
V-检查 TCP/IP 连接。
R-重新启动。
H-技术支持及最新产品。
X-退出。

EIO-RTU 不支持以太网 EIO 主菜单

选择一个命令:
T-Modbus RTU 设置。
M-查看剩余内存空间。
D-I/O 控制及协议转发配置。
P-口令设置。
F-恢复默认设置。
R-重新启动。
H-技术支持及最新产品。
X-退出

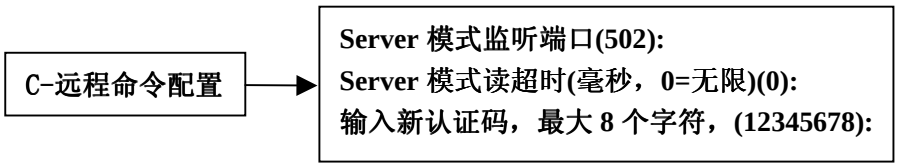
操作默认规则：

- 菜单的第一个字母或数字为选择此菜单功能键。
- 如果不录入数据，直接按回车为跳过参数。
- 只有当一个菜单项目都走完后，设置的参数才会被存储。
- 如果录入参数值错误，可以用 Backspace 键来重新录入参数值。
- 如果达到超时时间，没有操作，设备将中断 Telnet 连接。

2、 远程命令配置

1）远程命令配置菜单结构

远程命令配置菜单结构



2）Server 模式监听端口

参数值：1-65535

默认值：502（Modbus TCP 标准端口）

说明：EIO 通过此端口接收远程控制命令，对于 Modbus TCP 协议为 502 端口。

3）Server 模式读超时

参数值：0-65535

默认值：0（无限）

说明：如果在超时时间内没有数据传输，EIO 将中断此 TCP/IP 连接。

4）输入新认证码

参数值：最大 8 个数字

默认值：12345678

说明：对于 Modbus TCP 此参数无效，只有使用 EIO 特殊控制协议时才有效，只有符合此认证码的控制数据才会被 EIO 接收执行。

3、 EIO Link 配置

```
选择 EIO Link 工作模式:
1-启用 EIO Link Server 模式
2-启用 EIO Link Client 模式
3-禁用 EIO Link
```

如果准备将 2 个 EIO 设备使用 EIO Link 技术连接起来，那么就需要将一个 EIO 设置为 EIO Link Server 模式，另一个设置为 EIO Link Client 模式。

1) EIO Link Server 模式参数设置

在选择了<启用 EIO Link Server 模式>后，需要配置如下参数。

- Server 下 EIO Link 监听端口(5100):

监听其他 Client 模式下的 EIO 设备发起的 EIO Link 的 TCP/IP 连接。

默认值：5100

- Server 下 EIO Link 端口连接超时(秒)(0):

如果在超过此时间还没有数据传输，设备将中断此连接。

默认值：0（无超时）

- EIO Link 状态同步间隔(毫秒)(5000):

EIO 将以此参数为间隔，向其他 EIO 发出同步数据，此同步数据为 EIO 的开关量输入状态，其他 EIO 收到此数据后，将使用此数据设置自身的开关量输出状态。只有与其他 EIO 建立了 EIO Link TCP/IP 连接后，才会发送状态同步数据。

2) EIO Link Client 模式参数设置

- Client 下 EIO Link 要连接的远程 EIO 设备地址(192.168.192.100)

  默认为：192.168.192.100

  EIO 要连接远程 Server 模式 EIO 的 IP 地址。

- Client 下 EIO Link 要连接的远程 EIO 设备端口(5100)：

  默认为：5100

  EIO 要连接远程 Server 模式 EIO 的端口。

- Client 下 EIO Link TCP/IP 连接超时，达到此超时后，将中断连接(秒)(0)

  默认为：0 无超时

  如果超过此超时，TCP/IP 连接上没有数据接收，将中断此 TCP/IP 连接。

- Client 下 EIO Link TCP/IP 连接重试间隔(毫秒)(5000)：

  如果 EIO 尝试连接远程 EIO 失败，将等待此参数指定的时间，然后重试。

- EIO Link 状态同步间隔(毫秒)(5000)：

  EIO 将以此参数为间隔，向其他 EIO 发出同步数据，此同步数据为 EIO 的开关量输入状态，其他 EIO 收到此数据后，将使用此数据设置自身的开关量输出状态。只有与其他 EIO 建立了 EIO Link TCP/IP 连接后，才会发送状态同步数据。

4、 RS232/RS485 串口服务器配置。

EIO 的串口服务器端口映射在设备的 TCP/IP 6020 端口，远程主机与 EIO 建立连接后，就可以实现 RS232/RS485<->以太网&TCP/IP 的透明数据转发。

EIO 的 RS232/RS485 端口参数可以与 VSPM 虚拟串口参数同步，无须手工设置。

1）选择工作模式

```
请输入
选择工作模式:
1-Server 模式
2-Client 模式
3-UDP 广播模式
```

EIO 的串口服务器功能支持 TCP/IP Server、Client 和 UDP 广播模式，前两种工作模式使用 TCP/IP 传输数据，后一种使用 UDP 广播包来传输数据。

2）Server 模式参数

当模式选择为<1-Server 模式>时，需要设置如下参数。

配置参数项	值	说明
串口（N）对应的 TCP/IP 端口	0<值<65536 默认值： 串口 A-D 对应 6020-6023	串口 N 对应的 TCP/IP 监听端口。EIO 将监听此端口并等待连接，一旦建立 TCP/IP 连接，此连接将与串口 N 进行双向数据转发通讯。
串口（N）的 TCP/IP 读超时	以秒为单位，值<65536，0 为无超时。 默认值：0，无超时	如果 TCP/IP 连接在指定时间内没有数据，EIO 将中断此连接。

3) Client 模式参数

工作在 TCP/IP 客户端模式，根据设置的远程 IP 或域名，主动连接远程服务器。

- 心跳包过滤

心跳包过滤(应用于 UART 转以太网)： 1-禁用心跳包过滤 2-启用心跳包过滤,心跳包数据将不会被转发

由 Server 端主机定时发送特定的 6 字节心跳包，设备通过心跳包数据，检测 TCP/IP 连接状态。如果设备在接收超时时间内没有收到心跳数据，将判定此 TCP/IP 连接为死连接，并中断重连。

如果选择<禁用心跳包过滤>，此心跳包数据将被转发到串口。

如果选择<启用心跳包过滤>，串口服务器将根据设置，过滤掉心跳包，不转发心跳包数据。

默认设置为<1-禁用心跳包过滤>。

VSPM 虚拟串口软件通过插件，可以自动发送心跳包。其他软件或非 PC 设备，需要由相应软件发送心跳包。

- 心跳包定义

请输入心跳包字节(6 字节)(000102030405):0A0B0C0D0E0F

如果选择了<1-启用心跳包过滤>，将提示输入 6 字节心跳包定义，默认为：00-01-02-03-04-05。

按照 16 进制，连续输入 12 个数字，两个数字为 1 个字节。如果收到的

数据与这 6 个字节完全相同，设备将拦截此数据，不做转发。

当模式选择为<2-Client 模式>时，需要设置如下参数。

配置参数项	值	说明
远程服务器 IP 地址	有效的 IP 地址 默认值：192.168.192.10	可以分别为每个扩展串口设置不同的远程服务器 IP 和端口。 EIO 以“尝试连接服务器间隔”值为间隔，尝试连接“远程服务器 IP 地址”和“远程服务器端口”。 如果成功建立 TCP/IP 连接，此连接将与串口 N 进行双向数据转发通讯。
远程服务器端口	0<值<65536 默认值： 串口 A-D 对应 6050-6053	
尝试连接服务器间隔	以毫秒为单位，100<值<65536 默认值：5000ms	
串口的 TCP/IP 读超时	以秒为单位 值<65536，0 为无超时。 默认值：0，无超时	如果 TCP/IP 连接在指定时间内没有数据，EIO 将中断此连接。

4）UDP 模式参数

当模式选择为<3-UDP 广播模式>时，需要设置如下参数。

配置参数项	值	说明
UDP 发送地址(0-为广播地址)(255。255。255。255)	有效的 IP 地址，输入 0 为广播地址。 默认值：255。255。255。255	EIO 使用<UDP 发送地址>和<UDP 发送端口>，发送数据，使用 UDP 接收端口接收广播数据。 UDP 广播模式下，EIO 接收到的网络数据，将被转发到 EIO 所有串口。
UDP 发送端口(7102)	0<值<65536 默认值：7102	
UDP 接收端口(7101)	0<值<65536 默认值：7102	

5）通用串口参数设置

配置参数项	值	说明
串口速度(Bps)	300，1200，2400，4800，9600，19200，38400，57600，115200， 默认值：9600	此参数为扩展串口的实际运行参数 在重新启动串口服务器后生效。 《VSPM 虚拟串口软件》
串口数据位	5，6，7，8 默认值：8	

串口校验位	无校验(0)，奇校验(1)，偶校验(2) 默认值：无校验(0)	虚拟 COM 口的配置参数，并不与串口服务器的扩展串口参数进行同步。 更详细信息请参见配置要点。
串口停止位	1 停止位，2 停止位 默认值：1 停止位	

6）通用串口数据接收模式

- 流转发模式，收到即转发

此模式为默认设置，在此模式下，当串口服务器从串口收到数据时，不做等待及缓冲，直接转发收到的数据到 TCP/IP 连接，这些数据由 VSPM 虚拟串口软件完成数据包重组，所以不会出现拆包现象。此方式工作速度快、效率高，并可兼容绝大多数应用场合。

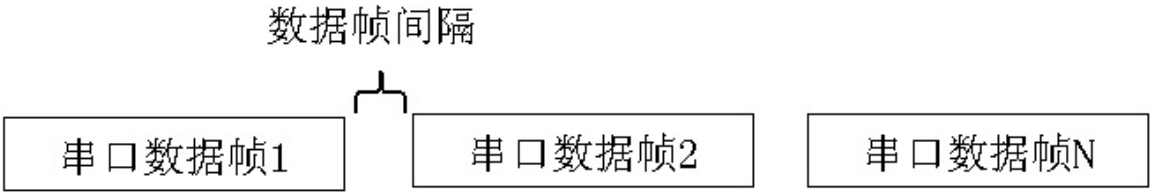
在应用软件使用 Socket 直连方式与串口服务器通讯， 应用软件必须自己完成包重组工作，否则将出现拆包现象，其表现为从 TCP/IP 连接无法一次收全一个串口数据帧。此问题是串口通讯模式与 TCP/IP 通讯模式存在的差异所导致。

如果应用软件无法完成包重组工作，那可以使用下面的自适应数据帧工作模式。

- 自适应数据帧，存贮转发模式，最大支持 1024 字节的串口数据帧，推荐用于 Socket 方式。

此工作模式下，串口服务器将在接收到一个完整的数据帧后，再转发这个完整的数据帧到 TCP/IP 连接。该模式由串口服务器完成串口数据帧重组。

串口服务器根据数据帧之间的间隔来判断是否收到了一个完整的数据帧。



当在指定帧间隔时间内，没有接收到新的串口数据，就认为收到了一个完整的数据帧，此数据帧将被一次性转发到 **TCP/IP** 连接。

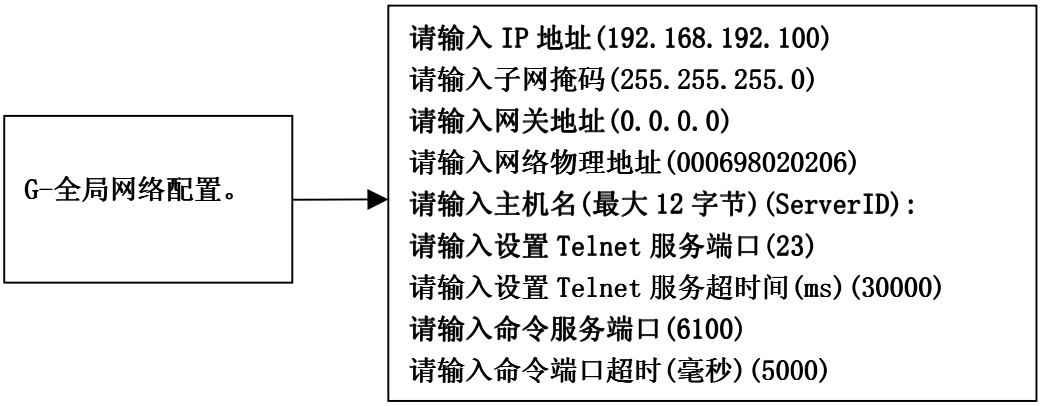
<接收数据帧间隔>可以用来指定串口数据帧之间的间隔值，默认为 **20** 毫秒。

- Modbus TCP 至 Modbus RTU 协议双向转换模式

实现 Modbus TCP 与 Modbus RTU 的双向数据转换。网口处理 Modbus TCP 报文，RS232/RS485 端口处理 Modbus RTU 报文。Modbus TCP 默认端口为 502 端口。

5、 全局网络配置

全局网络配置菜单结构

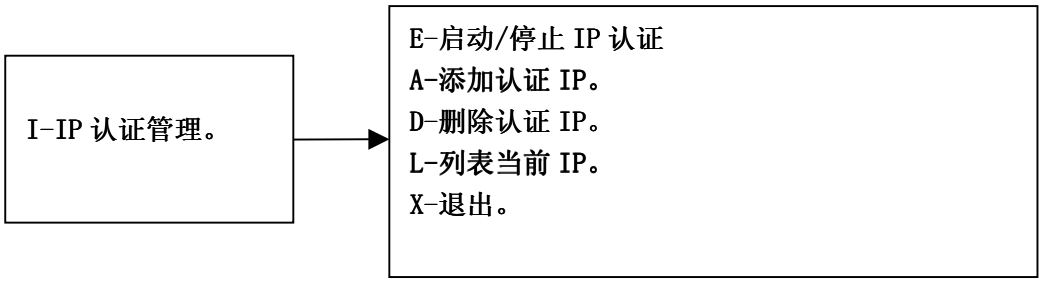


配置参数项	值	说明
IP 地址	有效的 IP 地址 默认值：192.168.192.100	IP 地址及网络物理地址在同一个物理网段中必须唯一。 如果串口服务器不仅仅工作在本地的局域网内或运行在 NAT 环境下，就必须设置正确的网关。 这些参数将在重新启动后生效。
子网掩码	有效的子网掩码地址 默认值：255.255.255.0	
网关	有效的网关地址 默认值：无	
网络物理地址	6 字节的网络物理地址 默认值：00069XXXXXX	
主机名	12 个字节的主机名，用于标识串口服务器。	
Telnet 服务端口	0<值<65536 默认值：23	服务器的 Telnet 服务将通过“Telnet 服务端口”提供，如果在“Telnet 服务超时”指定的时间内，没有数据，服务器将中断连接。
Telnet 服务超时	以毫秒为单位 值<65536，0 为无超时。 默认值：30000	
命令服务端口	0<值<65536 默认值：6100	通过此端口来完成一些配置操作，比如虚拟串口软件同步串口参数等。

命令端口超时	以毫秒为单位 值<65536，0 为无超时。 默认值：5000	命令端口超时。

6、 IP 认证管理。

IP 认证管理菜单结构



1）E-启动/停止 IP 认证

用来控制是否启动 IP 认证功能，如果启用了 IP 认证，那么只有在 IP 认证表中的 IP 才允许访问本设备。

```
e

启用 IP 认证(否):(y-是|n-否)
n

已经保存指定值:否
重新启动后，配置生效。
*****按任意键继续*****
```

2）A-添加认证 IP。

此功能将首先列表当前 IP 认证表内容，然后接收一个 IP，如果 IP 已经存在，将提示“要添加的 IP 已经存在。”否则将提示添加成功。

```
a

<IP 认证表>
1-192.168.192.1
<结束>
请输入要添加的 IP 地址:192.168.192.2
IP:192.168.192.2 添加成功。
*****按任意键继续*****
```

3）D-删除认证 IP。

此功能将首先显示一个 IP 列表，从中选择一个 IP 即可完成删除。

```
<IP 认证表>
1-192.168.192.2
2-192.168.192.1
<结束>
请选择要删除的 IP 地址:
2

IP:192.168.192.1 已经被删除。
*****按任意键继续*****
```

4）L-列表当前 IP。

显示当前 IP 表内容。

```
l

<IP 认证表>
1-192.168.192.2
<结束>
*****按任意键继续*****
```

7、 查看当前网络配置

按“N”键，服务器将返回当前系统的 IP、子网掩码、网关、网络物理地址和是否启用了 IP 认证等相关网络信息。

例如：

```
n

网络物理地址:00-06-98-02-02-06
IP 地址:192.168.192.100
子网掩码:255.255.255.0
网关:0.0.0.0
启用 IP 认证:否
*****按任意键继续*****
```

8、 查看剩余内存空间。

按“M”键，服务器将返回当前系统以字节为单位的剩余内存。

例如：

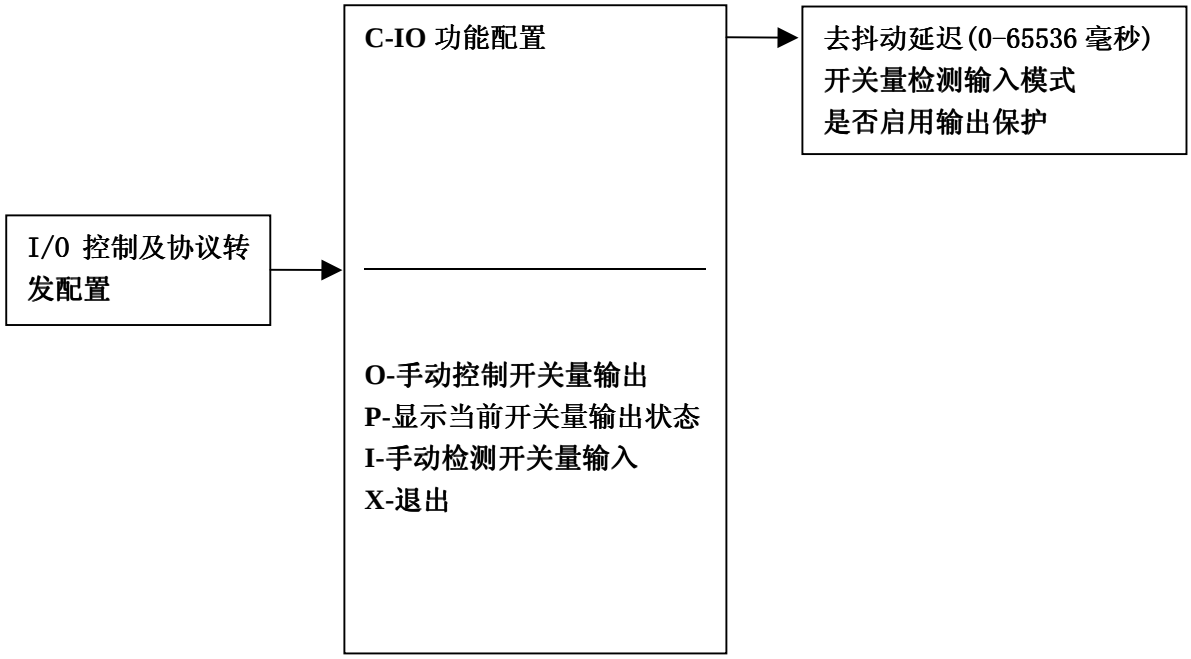
```
m

可用内存:22236
*****按任意键继续*****
```

9、 I/O 控制及协议转发配置。

1）远程命令配置菜单结构

EIO I/O 控制及协议转发配置菜单结构



2） IO 功能配置

配置参数项	值	说明
去抖动延迟	范围：0-65536 毫秒 默认值：20	输入可能会产生抖动，产生大量无用信号，延迟一段时间等待抖动停止后，再进行输入检测，可以去掉这些无用信号。
开关量检测输入模式	1-轮询模式。 2-中断模式，适用于按钮输入，不支持并发输入。 默认值：轮询模式	轮询模式：在 EIO Link 模式下可以同时响应多个开关量输入。 中断模式：在 EIO Link 模式下，如果 1 个开关量输入接通，设备将不再响应其他开关量变化，直到此开关量断开，才会响应其他开关量输入。
是否启用输出保护	1-启用输出保护，当通讯出现异常时，输出进入指定的安全状态。 2-禁用输出保护。 默认值：禁用	如果启用了输出保护，就需要指定一个输出状态值，当设备启动后或控制通讯出现异常时，设备将输出设置为指定的状态值。 状态值的位置 1，对应的输出为接通或低电平。

3） 手动控制开关量输出、显示当前开关量输出状态

A） 手动控制开关量输出

选择“0”为手动开关量输出，此功能需要输入一个字节的控制数据，控制数据位为 1，接通对应的开关量输出或设置 TTL 为低电平，数据位为 0，断开对应的开关量输出或设置 TTL 为高电平。

此功能可以用来检查控制器继电器输出部分工作是否正常。

```
o

请输入开关量输出控制字节(0-全接通，255-全断开):
192
已经保存指定值:192
*****按任意键继续*****
```

B) 显示当前开关量输出状态

注意：此状态值为控制器记录值，若硬件存在问题，此值与硬件输出将能不完全一致。

```
p
开关量输出状态:255

按<回车>退出，其他键继续检测...
```

C) 手动检测开关量输入

此功能可以用来检查控制器开关量检测部分是否工作正常。

```
i

当前的开关量输入状态:255

按<回车>退出，其他键继续检测...
```

10、 口令设置

此口令为管理员口令，最大 8 个字节，如果遗失了此口令，可以通过恢复默认值功能，恢复为默认口令：admin。

```
p

请输入新管理员口令(admin):123456
新口令: 123456 已经被接受，请保管好此口令，按任意键继续...
```

11、 恢复默认设置

恢复设备参数为默认值，恢复默认值以后，必须重新启动才可生效。

12、 检查 TCP/IP 连接

此功能通过发送“test connect”字符串，检查所有已经建立的转发连接。此功能主要用于诊断网络故障。

13、 重新启动

重新启动设备。

14、 Modbus RTU 设置

EIO-RTU 设备具备此菜单项。进入此菜单后，将首先配置串口参数，配置过程与<RS232/RS485 串口服务器配置>完全相同，但是没有网络相关参数。

1) Modbus RTU 设备地址，0-为广播地址(1)

参数值：0-255

默认值：1

说明：设备只接收符合此地址码的 Modbus RTU 报文，如果设置为 0，则为接收所有报文。

EIO 为免跳线设计，在此处设置设备地址码，在 Modbus RTU RS485 通讯模式下必须设置地址码，并且每个设备的地址码不能相同（广播地址除外）。

2) Modbus RTU 帧超时(毫秒)(500)：

参数值：1-65536

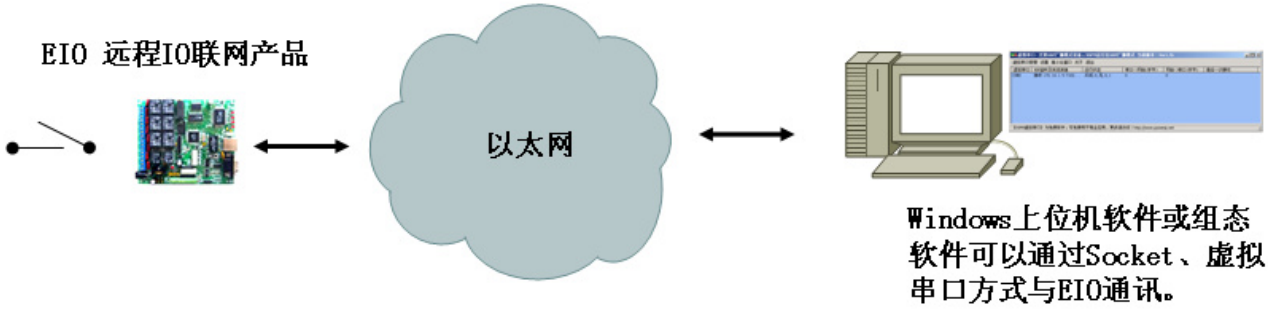
默认值：500

说明：设备以此参数为依据，判断一个 Modbus RTU 帧是否结束。如果设置值较大，可以提高兼容性，设置值较小可以提高设备报文处理反映速度。

15、 技术支持及最新产品

五、 技术要点及应用

1、 EIO 与上位机的工作模式



2、 两个 EIO 设备透传工作模式

EIO-A的开关量（或TTL）输入状态将被复制到EIO-B开关量（或TTL）输出

EIO-B的开关量（或TTL）输入状态将被复制到EIO-A开关量（或TTL）输出

EIO-A与EIO-B的RS232/RS485串口服务器端口也可以设置为透明传输模式。

3、 EIO 的串口服务器性能

1 个 RS232 或 RS485 端口，每端口最高速度：115200bps，数据位：5、6、7、8，

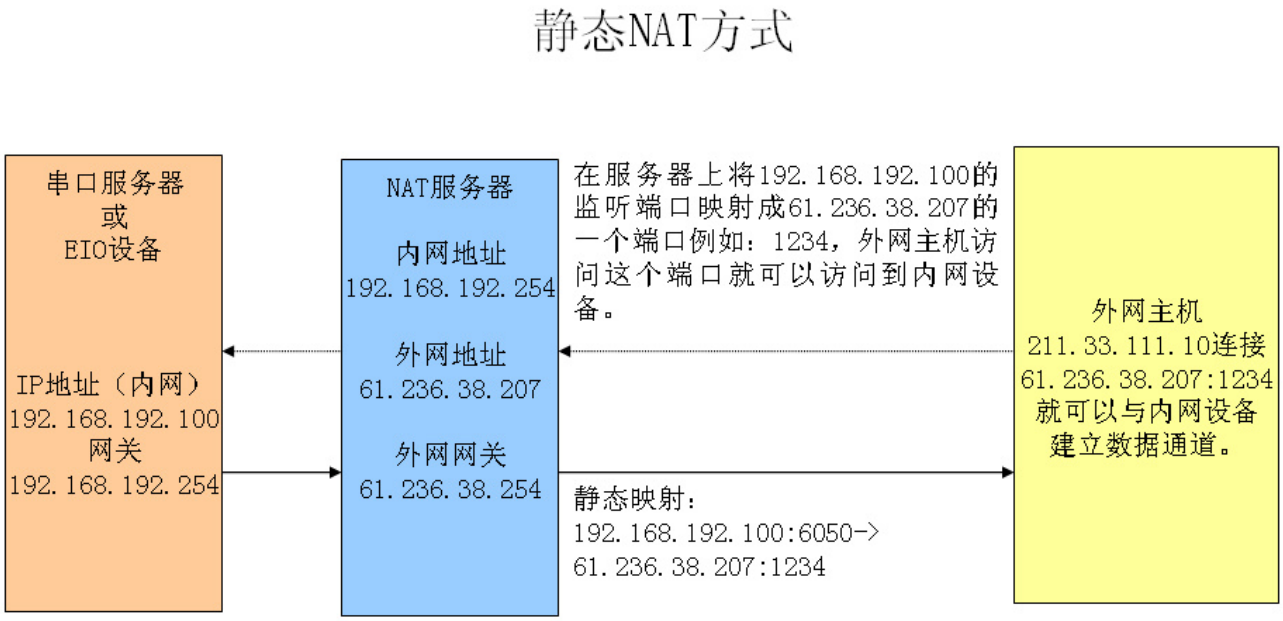
停止位：1、2，校验位：无、奇、偶、标记。

串口服务器端口参数可以与虚拟串口自动同步，无须手工设置。

4、 加密模块

如果需要提高传输的安全性，可以加配 RC6 64bit/128bit 加密模块，对命令进行加密传输。

5、 NAT 环境配置



6、 Modbus RTU CRC16 算法 C 代码

```
// CRC 高位字节值表
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
```

```
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
} ;

// CRC 低位字节值表

static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
```

```
0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,
0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,
0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
0x43, 0x83, 0x41, 0x81, 0x80, 0x40
} ;


unsigned short CRC16(unsigned char *puchMsg,  unsigned short usDataLen)
{
// 要进行 CRC 校验的消息 ，消息中字节数
```

```
//高 CRC 字节初始化
unsigned char uchCRCHi = 0xFF;
// 低 CRC 字节初始化
unsigned char uchCRCLo = 0xFF;

//CRC 循环中的索引
unsigned uIndex ;
//传输消息缓冲区
while (usDataLen--)
{
//计算 CRC
uIndex = uchCRCHi ^ *puchMsg++ ;
uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex];
uchCRCLo = auchCRCLo[uIndex] ;
};
return (uchCRCHi << 8 | uchCRCLo) ;
};
```

六、产品定制

我们拥有从硬件、嵌入式软件到应用软件的完整的研发能力，可以为您提供以下服务。

- OEM、ODM 生产。
- 按需定制嵌入式软件、硬件功能。
- 设计专用硬件数字接口、通讯接口。
- 扩展其他硬件专用芯片，如高精度 A/D 采集、D/A 输出等。
- 为周边产品嵌入专用控制协议，如各种读卡器、各类控制设备等。
- 应用软件设计。