

# RJ242 用户手册



2017 年 12 月 9 日 修订

# 目录

概述.....	1
第一章 购入检查及产品技术指标.....	3
1.1 购入检查.....	3
1.2 铭牌及型号说明.....	3
第二章 产品安装.....	4
2.1 安装环境要求.....	4
2.2 配电注意事项.....	4
第三章 产品规范.....	5
3.1 产品外观及部件名称.....	5
3.2 产品尺寸（mm）.....	5
3.3 电气参数.....	6
3.4 LAN 口.....	6
3.5 485 口.....	6
第四章 设备接口.....	7
4.1 电源接口.....	7
4.2 LAN 接口.....	7
4.3 485 口.....	8
4.4 指示灯.....	8
第五章 设备接口.....	9
5.1 软件安装.....	9
5.2 RJ242 界面介绍.....	10
5.3 各参数介绍.....	14
5.4 MODBUS TCP<==>MODBUS RTU 协议的说明.....	16
第六章 PC 连接.....	17
6.1 PC 的连接.....	17

## 概述

RJ242 针对现场一主多从网络的联网需求而设计。在原有 485 网络中增加一网口主机，便于远程数据采集监控，并解决了原 485 网络不允许许多主机并存的矛盾。

RJ242A 中，网口接主机。

RJ242B 中，网口接从机。

更加详细的介绍见后面章节。



RJ242A 典型应用

RJ242A 远程主机



RJ242B 典型应用

RJ242B 远程从机



RJ242A&B 组合典型应用



RJ242A&B 组合远程控制

# 第一章 购入检查及产品技术指标

## 1.1 购入检查

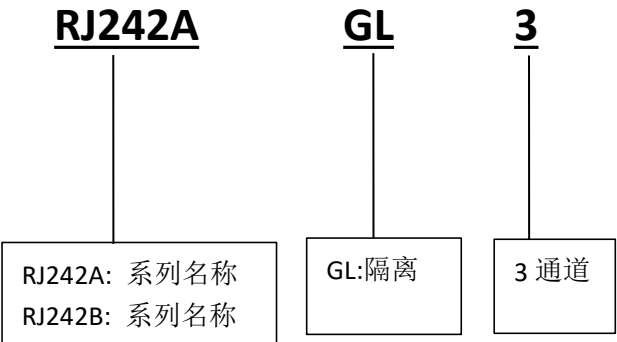
- 1.开箱前请检查产品的包装是否因运输不慎而造成损伤。
- 2.产品的规格、型号是否与所订购机种相符。
- 3.本产品出厂时，均对产品进行严格的测试和品质检查，请检查有无合格证、保修卡。
- 4.检查产品外观有无任何损坏。如有，请不要使用。

## 1.2 铭牌及型号说明

### 1.铭牌说明



### 2.型号说明



## 第二章 产品安装

### 2.1 安装环境要求

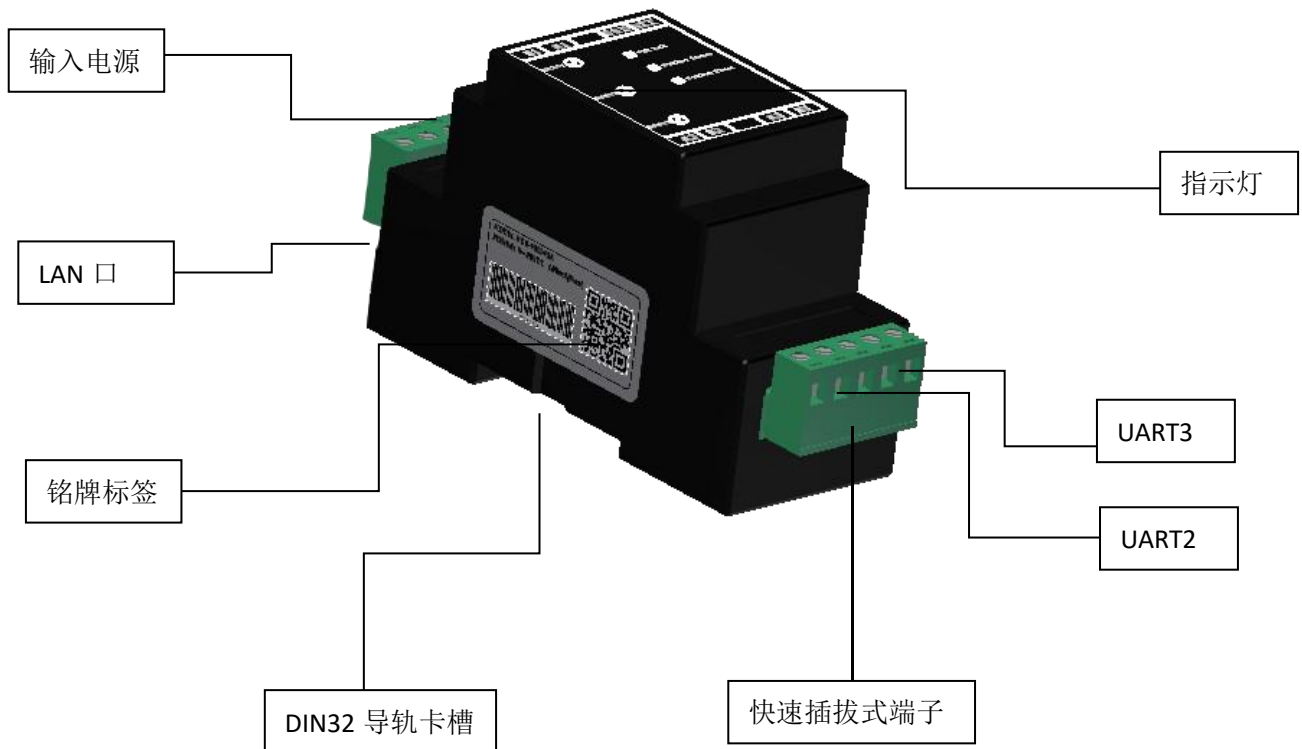
1. 环境温度  $-30^{\circ}\text{C}$ — $+70^{\circ}\text{C}$ ，通风良好，无结露场合(湿度小于90%RH)。
2. 避免震动，阳光直晒，远离强电磁干扰源。
3. 防止水滴、蒸汽、粉尘、灰尘、金属细粉末的侵入。
4. 远离油、盐及腐蚀性气体。
5. 禁止使用在易燃性、爆炸性气体、液体或固体危险场合。

### 2.2 配电注意事项

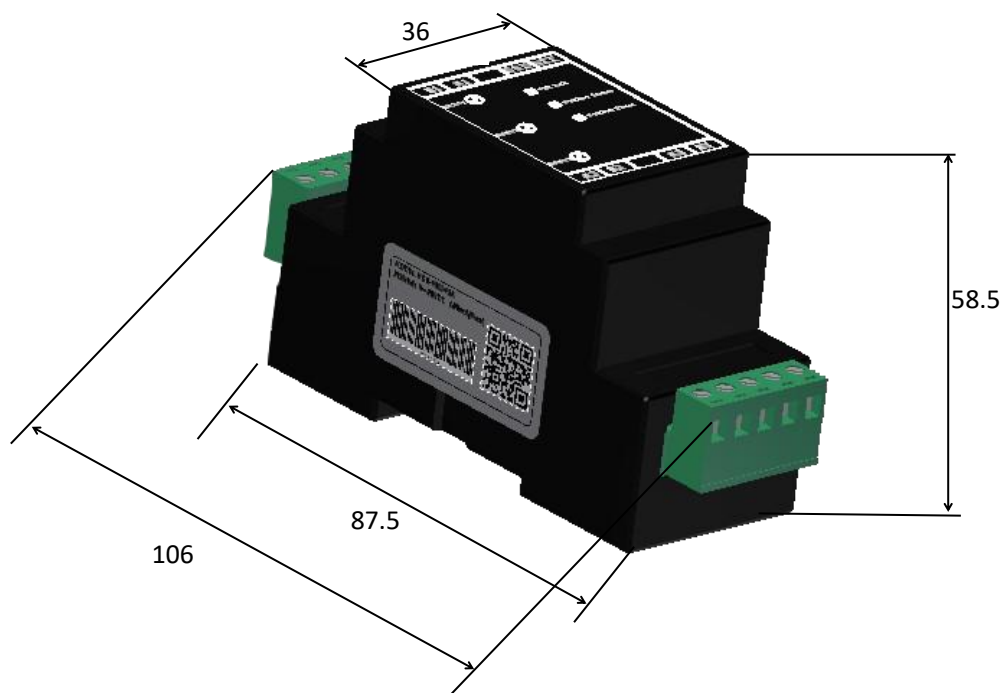
1. 电源端输入电源务必在 DC9V–35V 之间。
2. 绝对禁止将电源通入 A, B，否则有损坏通讯芯片的可能。
3. 通讯电缆尽量选择屏蔽双绞线，网口选择超五类网线。
4. 为尽量减少电磁干扰的影响，当使用的电磁接触器及继电器等距离本设备较近时，应考虑为其加装浪涌吸收装置。
5. 通讯线缆应采用屏蔽双绞线，如果通讯距离过长，可考虑在首末端加装 120 欧姆 1/8W 终端电阻。
6. 通讯电缆走线时应与动力电缆及其他强电分开走线，以减少不必要的干扰。如果信号电缆必须穿越动力电缆时，则应正交穿越。
7. 主电源配电线径应不小于 0.5 平方。

## 第三章 产品规范

### 3.1 产品外观及部件名称



### 3.2 产品尺寸 (mm)



### 3.3 电气参数

除非特别说明，下表所列参数是指  $T_{amb} = 25^{\circ}\text{C}$  时的值。

参数名称	符 合	数 值	单 位
电源电压	VCC	+9 ~ +35	V
功 耗	PD	小于 2.16	W

### 3.4 LAN 口

10/100M 自适应，RJ45 接口，1.5KV 电磁隔离。

### 3.5 485 口

- 位置：URAT2,URAT3 口;
- 信号类型：RS-485;
- 信号线：RS-485(A,B);
- 波特率：1200 ~115200bps;
- 数据位：7,8;
- 停止位：1,2;
- 校验：N,E,O,MARK;
- AB 接口保护：200ma 自恢复保险丝，TVS 600W

隔离版的如型号内有 GL 字样的产品，其 485 信号的电源采用内部隔离电源模块供电，外部 AB 信号，通过磁隔离芯片耦合到内电路，具有非常好的抗干扰性能，适合于电磁环境比较恶劣的场合。



## 第四章 设备接口

### 4.1 电源接口

该设备电源为直流 9~35V，通过 2P 可插拔式端子接入，请采用合适的直流电源供电，接线参考面板上的 VCC GND,设备带有防反接功能，反接时将不能正常工作。



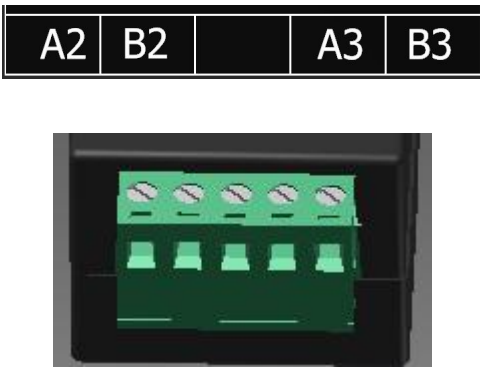
### 4.2 LAN 接口

LAN 接口为 10/100M 自适应以太网接口，接线图如下

管脚	信号
1	Tx+
2	Tx-
3	Rx+
6	Rx-

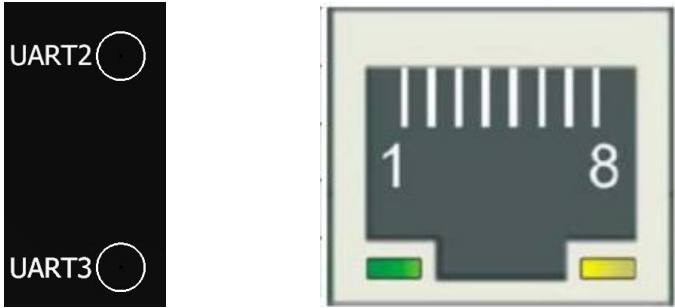
4.3 485 口

该设备具有 2 只 485 口，通过 1 只 5P 的可插拔式端子接入，接线如下所示



4.4 指示灯

当 485 口有数据通信时，相应的指示灯会亮起。



RJ45 插座上绿色 LED 灯	点亮：连接的是 100M 以太网 熄灭：连接的 10M 的以太网
RJ45 插座上黄色 LED 灯	闪烁：有数据传输

## 第五章 设备接口

### 5.1 软件安装

此配置软件采用 C#编写，运行架构为.net framework 4.0，如果没有，程序将无法运行，需要先安装此运行环境。一般下载后，直接点 DZ485.exe，如果没有出现如下页面，再进入文件夹 src,运行。 .net framework 进行安装，安装后即可打开，打开页面如下，点 RJ242,即可打开配置界面。



## 5.2 RJ242 界面介绍

运行界面如下，下面将逐部分介绍。



①设备出厂时参数如下：

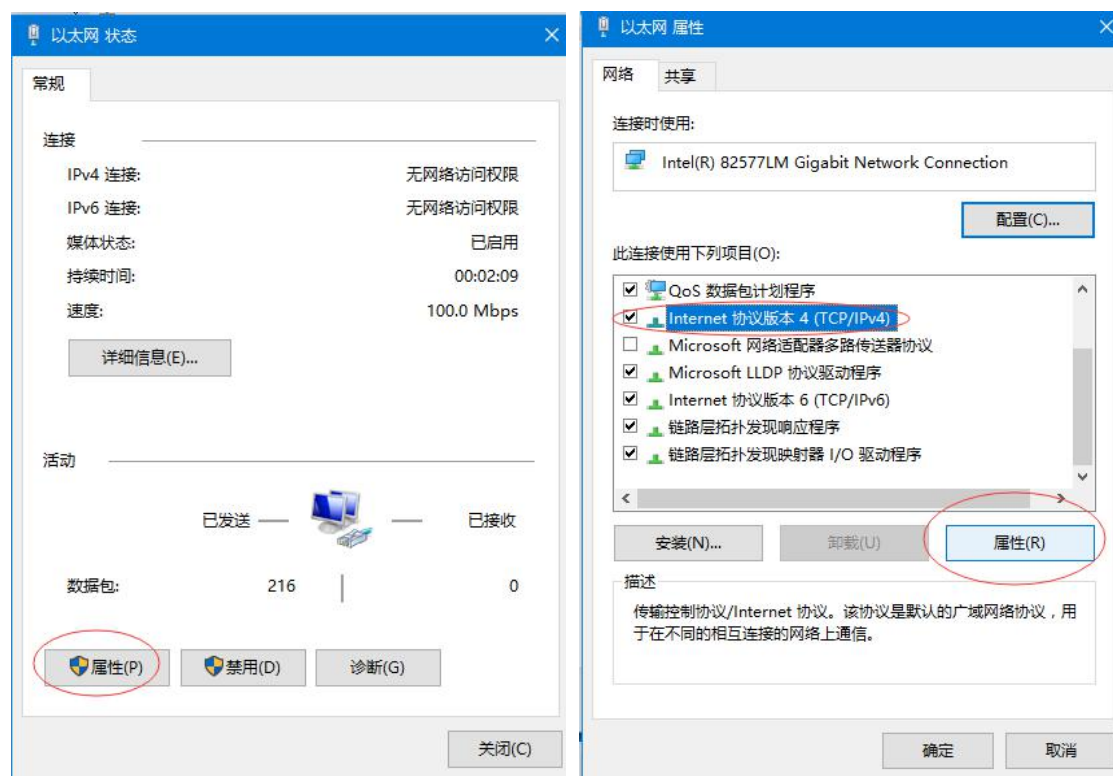
模块 IP	192.168.1.199
子网掩码	255.255.255.0
默认网关	192.168.1.1

②准备一根网线，最好用交叉线。初次配置，我们需要将模块和电脑直接连接。后面使用的话，一般路由器都有网线翻转功能，和路由连接也可以用平行线。和电脑连接最好用交叉线，如果电脑支持翻转，也可以用平行线。

③插入网线，接通电源，物理连通后，计算机和模块会看到绿灯亮，黄灯闪。电脑上会有连接标识。如下图



点开以太网，属性，如下左图



左图

右图

选择 IPV4，点属性，如上右图。

出现如下页面：

设置 IP,子网掩码，默认网关，使电脑和模块在一个网段内，如下图所示。然后点确定。



然后运行 ping 命令。

开始->运行->cmd,然后输入下命令

ping 192.168.1.199

```
C:\Users\Administrator>ping 192.168.1.199

正在 Ping 192.168.1.199 具有 32 字节的数据:
来自 192.168.1.199 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.1.199 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.1.199 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.1.199 的回复: 字节=32 时间<1ms TTL=128

192.168.1.199 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

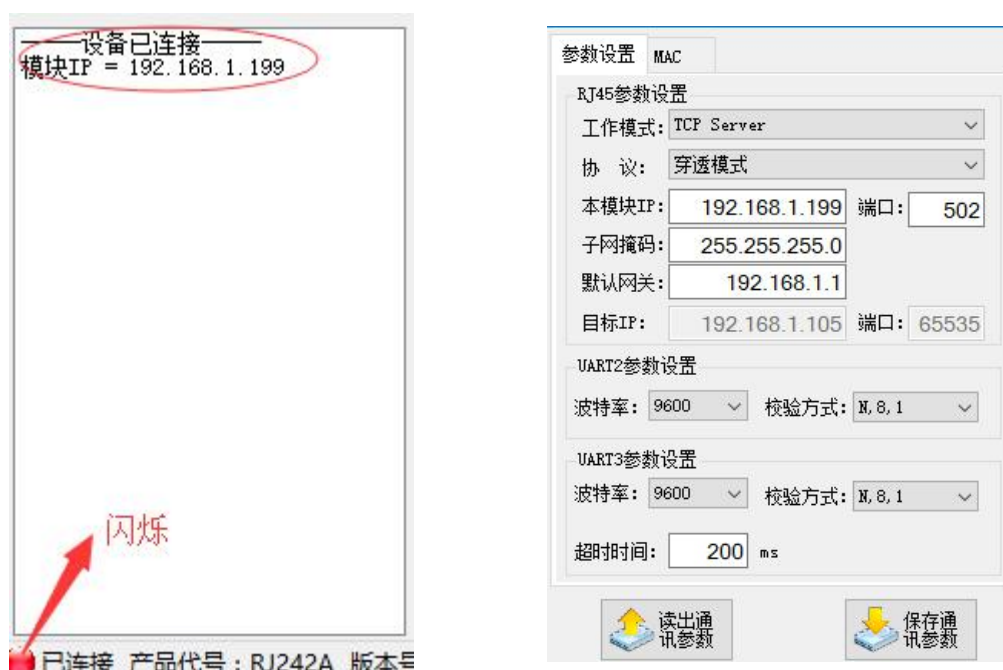
到此，表示我们已经完成准备工作，可以进入配置软件进行参数的配置了。

④参看 5.1 的步骤打开软件

点搜索模块，模块会自动上报自己的 IP，如下图，



然后点**连接模块**，显示设备已连接，且左下角灯会闪烁，表示有数据交互。同时模块内参数会同步到软件上并显示在右边。如下图



我们根据自己使用的实际情况进行相关参数配置，配置完成后，点保存参数即可。保存时，设备会重启，软件会掉线，等 10 秒左右，可以再次连接。

可以在连接后，随便动一下右边的数据，然后点读出参数。来确认参数是否已经更改成功。

用同样的方法可以设置更改 MAC 地址。一般不用管，出厂时，我们会设置好。如下图。然后点保存参数即可。到此所有的操作流程完毕。

## 5.3 各参数介绍



读出当前所有口的通讯参数。

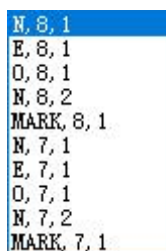


对当前设备的参数进行保存。

2 个 485 口，非常简单，只需要设置波特率和校验方式即可。

**波特率：**该端口获取或者发送数据时的波特率。

**校验方式：**设置该端口的通信编码的相关参数。支持下图内所有的校验方式。



**超时时间：**此参数的功能为如果 3 口连接的总线上有从机不能正常响应，最多等待的时间，目的在于防止因为某个从机的异常或缺失，导致整体通信的异常。

网口的设置稍多一些，先介绍简单一点的。

**本模块 IP：**这个就不用多说了

**端口号：**用于通信的端口号



**子网掩码：**略

**默认网关：**略

以上就是最基本的网络参数，我们用电脑，除了端口号，想必都很熟练了。

**目标 IP：**用于设置该模块用于主动建立连接的目标 IP。

**目标端口号：**用于设置该模块用于主动建立连接的目标端口号。

下面就是几个稍微复杂一点参数

**工作模式：**

有三种工作模式，TCP Server,TCP Client,UDP Server;

**TCP Server：**该模块作为 TCP 服务器，支持客户端的最大连接数量为 1 个。一旦客户端和该模块建立连接，则其它客户端将无法连接到该模块，直到连接的客户端主动退出，其它客户才有机会建立连接。

**TCP Client：**此模式下，该模块会根据目标 IP 以及目标端口号进行主动握手连接，连接好后，就可以进行通信。具有断开重连功能。

**UDP Server：**UDP server 是指在普通 UDP 的基础上不验证来源 IP 地址，收到 UDP 数据包后将目标 IP 改为数据来源 IP，类似 TCP server 的功能。

在此模式下，模块默认记录一个目标 IP，当串口有数据时，向记录的 IP 发送数据，同时，模块处于服务器地位，接受网络中发给模块的数据包，并随时调整目标 IP 为数据来源的 IP。

协议：无协议模式和 MODBUS RTU/TCP 协议。

**无协议模式：**网口接收到数据不进行处理就转给 485 口输出，485 口收到的数据不进行处理就转给网口输出。

**MODBUS TCP<==>MODBUS RTU 协议：**网口接收到的数据递交给 485 口时，先按 MODBUS TCP 协议将数据提取出来，然后按 MODBUS RTU 协议打包转给 485 口。当收到 485 口的数据时，先按 MODBUS RTU 协议提取数据，然后按 MODBUS TCP 协议打包从网口转出。

## 5.4 MODBUS TCP<==>MODBUS RTU 协议的说明

RJ242A 版为，网口接主机，URAT2 接主机，URAT3 接从机。

RJ242A 在使用 MODBUS 协议时，需将工作模式设置为 TCP Server 或者 UDP Server。

RJ242B 版为，网口接从机，URAT2 接主机，URAT3，接主机。

RJ242B 在使用 MODBUS 协议时，需将工作模式设置为 TCP Client 或者 UDP Client。

## 第六章 PC 连接

### 6.1 PC 的连接

PC 的连接比较丰富，可以用组态软件，也可以用 OPC 采集之后用高级语言调用 IOCOMP 来做，也可以自己用串口控件开发，这些要么成本较高，要么比较复杂。推荐大家用开源的一个 C#类，NMODBUS 来做，使用非常简单。

以下内容引用网上公开资料。

需要连续读取数据时，可以开一个线程。写入数据可以用事件触发写。

1、将 NModbus 类库导入工程中，添加引用、命名空间。工程属性必须配置为.NET 4.0。

2、创建 SerialPort 类的一个实例，配置参数，打开串口，如：

```
public SerialPort port = new SerialPort("COM1");//创建串口
port.BaudRate = 9600;//配置
port.DataBits = 8;
port.Parity = Parity.None;
port.StopBits = StopBits.One;
port.Open();//打开串口
```

3、调用相应方法创建 Modbus 主站，如：

```
IModbusSerialMaster master = ModbusSerialMaster.CreateRtu(port);
//创建 Rtu 传输模式、通过串口 port 通信的主站 master
IModbusSerialMaster master = ModbusSerialMaster.CreateAscii(port);
//ASCII 传输模式
```

4、配置参数，如：

```
master.Transport.ReadTimeout = 1000;//读取串口数据超时为 1000ms
master.Transport.WriteTimeout = 1000;//写入串口数据超时
master.Transport.Retries = 3;//重试次数
master.Transport.WaitToRetryMilliseconds = 250;//重试间隔
```

5、调用相应方法执行功能，如：

```
try
{
```

```

master.WriteMultipleRegisters(slaveId, startAddress, registers);
} 有以下几种功能:
//读线圈, 参数: 从站地址(8 位)、起始地址(16 位)、数量(16 位); 返回布
尔型数 组
bool[] ReadCoils(byte slaveAddress, ushort startAddress, ushort
numberOfPoints);
//读输入离散量, 参数: 从站地址(8 位)、起始地址(16 位)、数量(16 位);
返回布 尔型数组
bool[] ReadInputs(byte slaveAddress, ushort startAddress, ushort
numberOfPoints);
//读保持寄存器, 参数: 从站地址(8 位)、起始地址(16 位)、数量(16 位);
返回
16 位整型数组
ushort[] ReadHoldingRegisters(byte slaveAddress, ushort startAddress, ushort
numberOfPoints);
//读输入寄存器, 参数: 从站地址(8 位)、起始地址(16 位)、数量(16 位);
返回
16 位整型数组
ushort[] ReadInputRegisters(byte slaveAddress, ushort startAddress, ushort
numberOfPoints);
//写单个线圈, 参数: 从站地址(8 位), 线圈地址(16 位), 线圈值(布尔型)
void WriteSingleCoil(byte slaveAddress, ushort coilAddress, bool value);
//写单个寄存器, 参数: 从站地址(8 位), 寄存器地址(16 位), 寄存器值(16
位)
void WriteSingleRegister(byte slaveAddress, ushort registerAddress, ushort
value);
//写多个寄存器, 参数: 从站地址(8 位), 起始地址(16 位), 寄存器值(16 位
整型 数组)
void WriteMultipleRegisters(byte slaveAddress, ushort startAddress, ushort[]
data);
//写多个线圈, 参数: 从站地址(8 位), 起始地址(16 位), 线圈值(布尔型数
组)
void WriteMultipleCoils(byte slaveAddress, ushort startAddress, bool[] data);
//读写多个寄存器, 参数: 从站地址(8 位), 读起始地址(16 位), 数量(16 位),
写 起始地址(16 位), 写入值(16 位整型数组); 返回 16 位整型数组
ushort[] ReadWriteMultipleRegisters(byte slaveAddress, ushort
startReadAddress,
ushort numberOfPointsToRead, ushort startWriteAddress, ushort[] writeData);
6、使用 catch 语句捕捉异常: 如果执行没有出错, 则不抛出异常, 如果是
执行读操作则能得到相应的 返回值。 响应超时会抛出 TimeoutException
类型的异常; 接收到从站的异常响应时会抛出 SlaveException

```

