

# 智能串口数据交换器

## SS-430

### 产品手册

REV 2.0



上海泗博自动化技术有限公司  
Sibotech Automation Co., Ltd

技术支持热线:021-5102 8348  
E-mail: support@sibotech.net

# 目 录

一、引言 .....	2
1.1 关于说明书 .....	2
1.2 版权信息 .....	2
1.3 相关产品 .....	2
1.4 术语 .....	2
1.5 更改记录 .....	2
二、产品概述 .....	3
2.1 产品功能 .....	3
2.2 产品特点 .....	3
2.3 技术指标 .....	4
三、产品外观 .....	5
3.1 产品外观 .....	5
3.2 指示灯 .....	6
3.3 配置开关 .....	6
3.4 通信端口 .....	6
3.4.1 端口定义 .....	6
3.4.2 RS-485 端口技术规格 .....	8
四、使用方法 .....	9
4.1 配置模块 .....	9
4.2 软件配置 .....	9
4.3 运行 .....	10
4.3.1 数据交换 .....	10
4.3.2 自定义协议 .....	11
4.3.3 通用模式 .....	14
五、安装 .....	19
5.1 机械尺寸 .....	19
5.2 安装方法 .....	19
六、运行维护及注意事项 .....	20
附录：MODBUS协议 .....	21

# 一、引言

## 1.1 关于说明书

本说明书描述了网关 SS-430 的各项参数，具体使用方法和注意事项，方便工程人员的操作运用。在使用网关之前，请仔细阅读本说明书。

## 1.2 版权信息

本说明书中提及的数据和案例未经授权不可复制。

**SiboTech** 是上海泗博自动化技术有限公司的注册商标。

## 1.3 相关产品

本公司其它相关产品包括：

MD21：Modbus/DeviceNet 网关

PM-120：Modbus/Profibus-DP 网关

获得以上两款产品的说明，请访问公司网站[www.sibotech.net](http://www.sibotech.net)，或者拨打技术支持热线：021-5102 8348

## 1.4 术语

Modbus：MODICON 公司设计的一种通信协议

RS232、RS485、RS422：串口的硬件规范

SS-430：智能串口数据交换器

RS-25：RS232/RS485 转换器

## 1.5 更改记录

◆ 2009 年 10 月修订 Rev1.2:

[1] 删除旧的网站地址；

[2] 增加自定义协议报文示例。

◆ 2010 年 03 月修订 Rev2.0:

[1] 增加协议类型。

注：原有协议类型：Modbus 主站、Modbus 从站、自定义

现有协议类型：Modbus 主站、Modbus 从站、自定义、通用模式

通用模式可分为通用模式-问答式和通用模式-接收式

## 二、产品概述

### 2.1 产品功能

1. 突破 Modbus 主站只能有一个的限制，可以使两个 Modbus 主站同时访问一段 Modbus 总线（SS-430 两个端口配置为从站，各连接一个主站；其它从站设备连接于第三个端口）；
2. 整合多个 Modbus 协议的设备，使多达近百个 Modbus 设备如同一个设备那样被访问；
3. 无需 PLC、PC 等主站设备，即可使 Modbus 从站设备互相交换数据；
4. 无需开发 Modbus 协议，即可使用简单的协议轻松连接 Modbus 设备。

### 2.2 产品特点

▼**应用广泛：**凡具有 RS232/RS485/RS422 接口的设备都可以使用本产品实现数据的交换和传递。如：具有 RS485 接口的变频器、电机启动保护装置、智能高低压电器、电量测量装置、各种变送器、智能现场测量设备、仪表、PLC 及 PC 等等。

▼**配置简单：**用户不必了解 Modbus 技术细节，只需参考本手册，根据要求使用网关配置软件 GT-123 可轻松完成 SS-430 的配置，不需要复杂编程，即可在短时间内实现连接通信。

## 2.3 技术指标

[1] SS-430 具有四个串口，串口 I 为 RS232 标准，串口 II 为 RS485 标准，串口 III 为 RS232 标准，串口 IV 为 RS485/RS422 标准。

[2] 串口参数：

- ① 工作方式：半双工；
- ② 波特率：300、600、1200、2400、9600、19.2K、38.4K、57.6、115.2Kbps 可选；
- ③ 数据位：8 位；
- ④ 校验位：无、奇、偶、标志、空格可选；
- ⑤ 停止位：1、2 位可选。

[3] Modbus 主站：

- ① 支持的功能码：01H、02H、03H、04H、05H、06H、0FH、10H 号功能；
- ② 支持的格式：RTU 格式和 ASCII 格式；
- ③ 具有的功能：写命令连续输出、禁止输出或逢变输出的可选择功能；
- ④ 每个主站最多可配置 48 条 Modbus 命令。

[4] Modbus 从站：

- ① 支持的功能码：03H、04H、06H、10H 号功能；
- ② 支持的格式：RTU 格式和 ASCII 格式。

[5] SS-430 数据输入/输出数量：

- ①  $\text{Input Bytes} + \text{Output Bytes} \leq 2\text{KBytes}$
- ②  $\text{Max Input Bytes} \leq 1\text{KBytes}$
- ③  $\text{Max Output Bytes} \leq 1\text{KBytes}$

[6] 供电：24VDC(11V ~ 30V)

[7] 功耗：<140mA (DC24V)

[8] 工作环境温度：-20 ~ 60℃；工作环境湿度：5 to 95%（无凝露）

[9] 内置静电防护：15 KV ESD；通信端口隔离：3KV

[10] 机械尺寸：100mm x 70mm x 25mm [不包括导轨连接器]

[11] 安装：35mm 导轨

[12] 防护等级：IP20

## 三、产品外观

### 3.1 产品外观

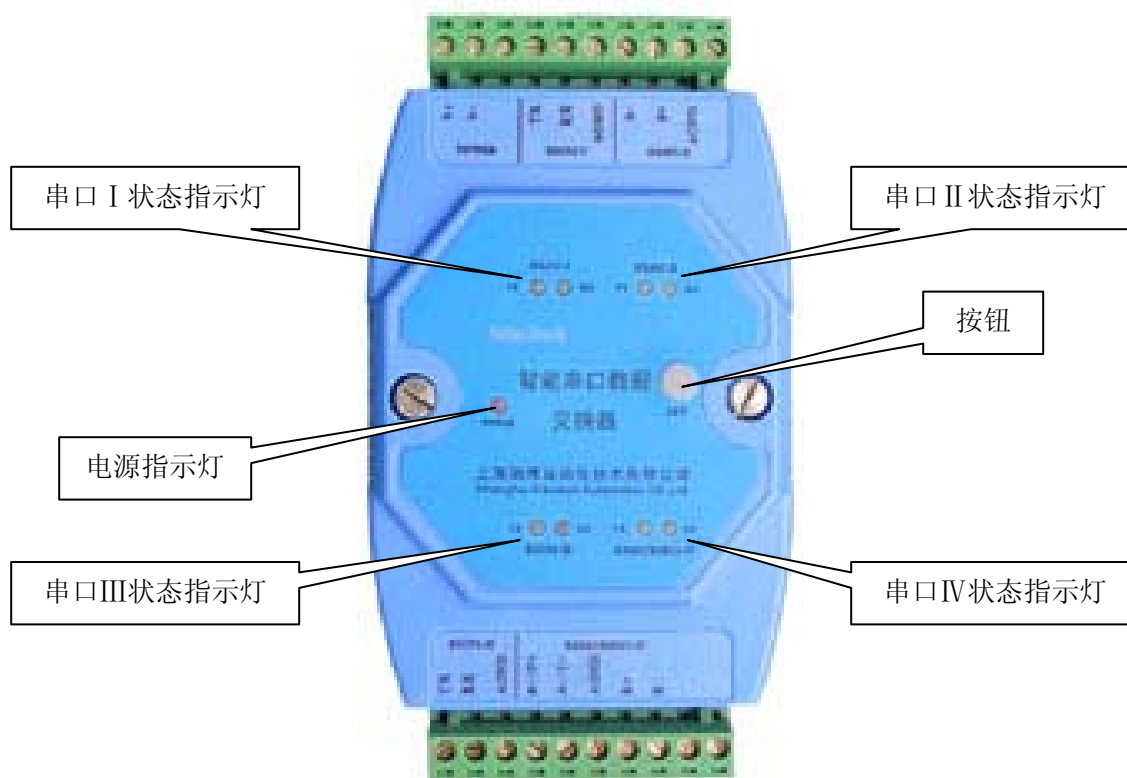


图 1 产品外观

## 3.2 指示灯

指示灯	状态	含义
POWER	红灯常亮	有电源
	红灯常灭	无电源
TX RS232 - I	绿灯闪烁	正在发送数据
	绿灯常灭	无发送
RX RS232 - I	绿灯闪烁	正在接收数据
	绿灯常灭	无接收
TX RS485 - II	绿灯闪烁	正在发送数据
	绿灯常灭	无发送
RX RS485 - II	绿灯闪烁	正在接收数据
	绿灯常灭	无接收
TX RS232 -III	绿灯闪烁	正在发送数据
	绿灯常灭	无发送
RX RS232 -III	绿灯闪烁	正在接收数据
	绿灯常灭	无接收
TX RS485/RS422 -IV	绿灯闪烁	正在发送数据
	绿灯常灭	无发送
RX RS485/RS422 -IV	绿灯闪烁	正在接收数据
	绿灯常灭	无接收

## 3.3 配置开关

SS-430 上电后双击 SET 按钮，进入配置模式。用户可以通过四个串口中的任意一个串口将 SS-430 连接到 PC（如果用 RS485 或 RS422 端口需使用转换器连接 PC），并使用网关配置软件 GT-123 配置 SS-430。

## 3.4 通信端口

### 3.4.1 端口定义

#### (1) RS232/485 & 电源端（上端）

引脚 (从左到右)	功能
1	24V+
2	24V-
3	无连接
4	TX, RS232 数据发送, 连接到用户设备 RS232 的 RXD
5	RX, RS232 数据接收, 连接到用户设备 RS232 的 TXD
6	RS232 GND
7	RS485_A, 数据-
8	RS485_B, 数据+
9	RS485 GND
10	无连接



(2) RS232/422/485 端 (下端)



引脚（从左到右）	功能
1	TX_232，连接到用户设备 RS232 的 RXD
2	RX_232，连接到用户设备 RS232 的 TXD
3	RS232 GND
4	B+/T+，连接到 RS485 B+ 或 RS422 T+
5	A-/T-，连接到 RS485 A- 或 RS422 T-
6	RS485/RS422 GND
7	R+，连接到 RS422 R+
8	R-，连接到 RS422 R-
9	无连接
10	无连接

### 3.4.2 RS-485 端口技术规格

站点数：每分段 32 个站（不带中继），可多到 127 个站（带中继）；

连接方式：10 针可插拔端子

注：总线的最远两端可配置一个总线终端电阻  $120\Omega$  1/4W 以减少信号反射。

## 四、使用方法

### 4.1 配置模块

使用 SS-430 的几个步骤：

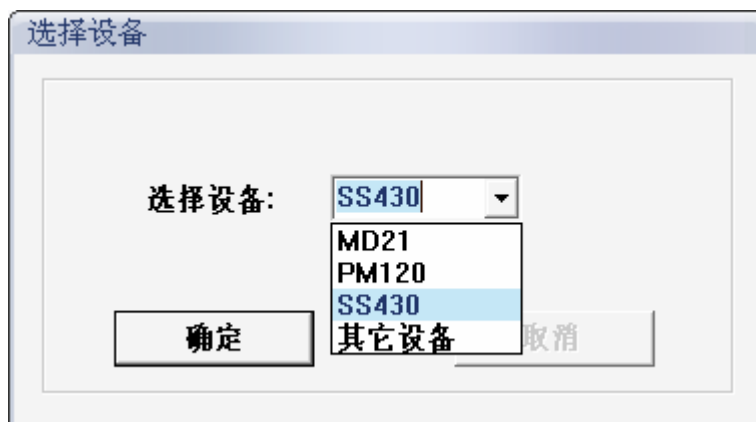
- 按照说明书完成硬件接线；
- 将四个串口中的一个串口连接至 PC 串口（PC 串口为 RS232 标准，采用模块的 RS232 口）；
- 给 SS-430 上电，然后**双击** SET 按钮，SS-430 进入配置模式；
- 使用网关配置软件 GT-123 对 SS-430 进行配置，配置完成后关闭 SS-430 电源；
- 给 SS-430 重新上电，即进入正常运行状态。

### 4.2 软件配置

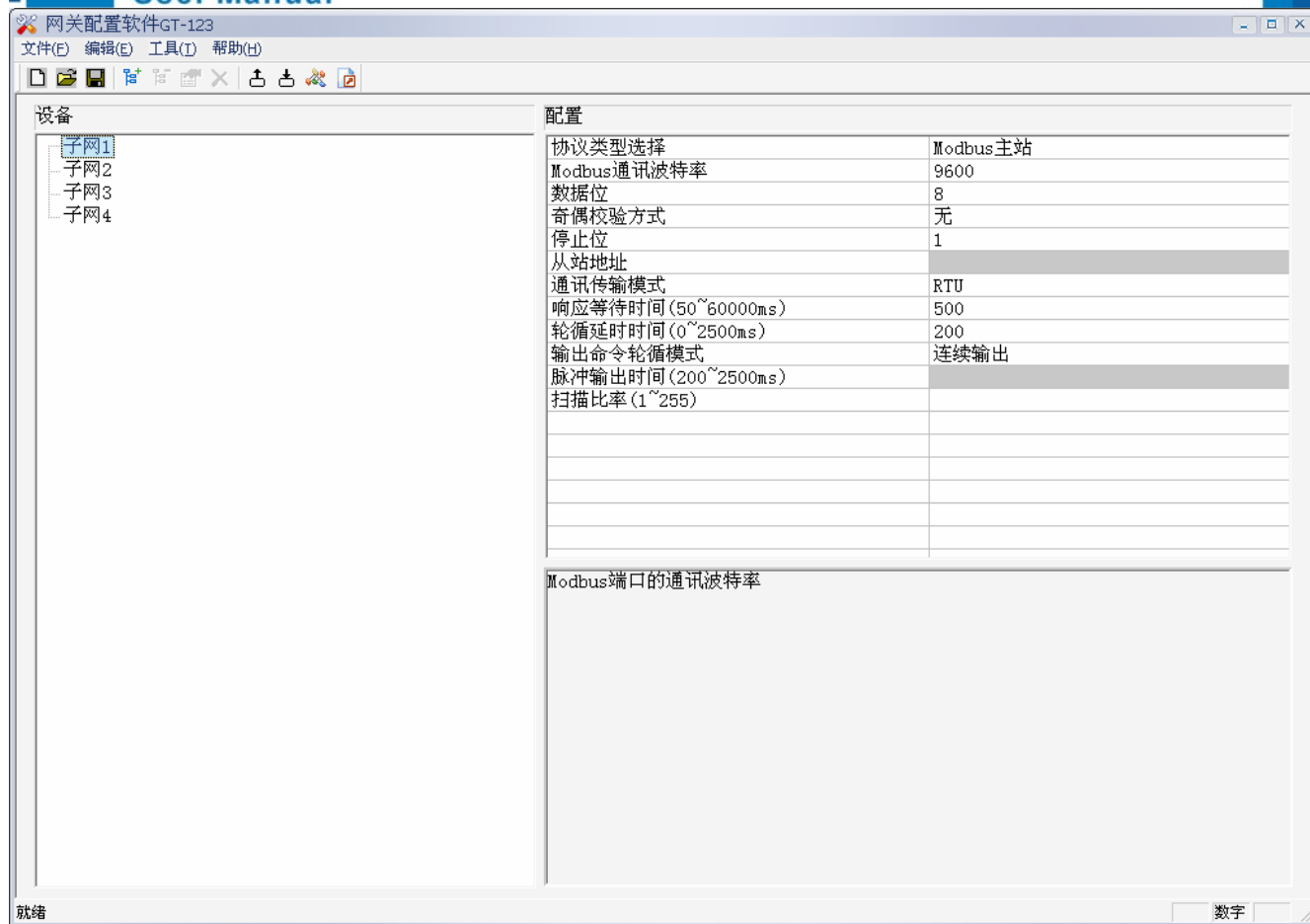
配置模块需要使用光盘中的 GT-123。用户需要安装 GT-123。

用户使用网关配置软件 GT-123 可以轻松完成 SS-430 的配置，包括通信波特率、奇偶校验、停止位、通信协议选择及协议参数等，并可对网关内存映射数据进行冲突检测。

安装完成后，启动软件，选择设备类型：



然后进入主菜单



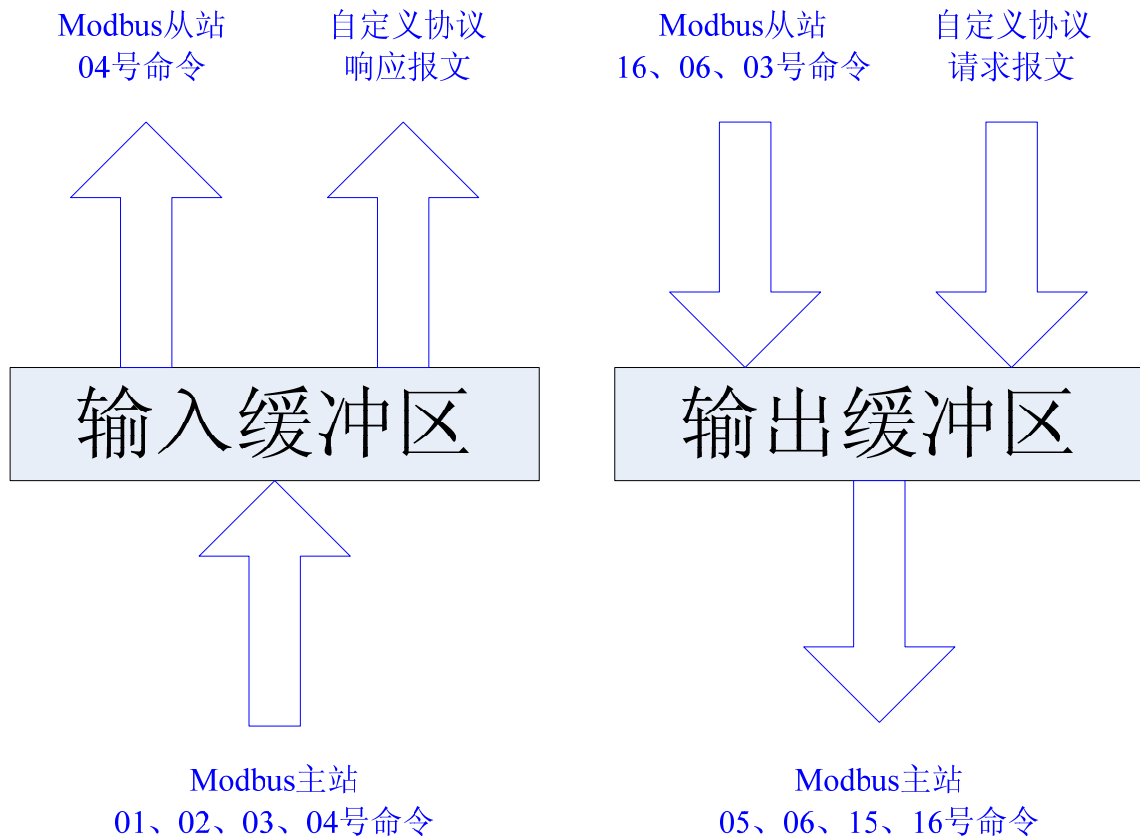
网关配置软件 GT-123 的详细使用方法详见 GT-123 软件说明书。

## 4.3 运行

### 4.3.1 数据交换

SS-430 的多串口之间的数据转换通过“映射”关系来建立。在 SS-430 中有两块数据缓冲区，一块是输入缓冲区（1K 字节），另一块是输出缓冲区（1K 字节）。

Modbus 主站读取命令将读取的数据写入到网络输入缓冲区，供 Modbus 从站或自定义从站侧读取。Modbus 主站写寄存器类的命令从输出缓冲区取数据，通过写命令输出到相应的 Modbus 设备。如下图所示：



输入缓冲区地址范围：0x0000 ~ 0x03FF；

输出缓冲区地址范围：0x4000 ~ 0x43FF。

输入输出缓冲区起始地址：

	输入缓冲区	输出缓冲区
内存映射起始地址	0000H	4000H
Modbus 从站 PLC 地址	30001	40001
Modbus 从站协议地址	0000H	0000H

每个 Modbus 主站都可配置 **48** 条命令，每条命令可以读取一组连续的 Modbus 寄存器。

映射关系在配置模式下用户可自行组态。

## 4.3.2 自定义协议

### (1) 定义

用户设备作为通信的发起方，在请求帧中发送输出数据。

SS-430 作为通信的响应方，在响应帧中发送输入数据。

通信为点对点。

请求帧中相邻字节间的时间间隔不能超过 50ms，否则 SS-430 将放弃这帧数据。

每条有效请求帧 SS-430 应在 200ms 内开始作出响应。

支持通信波特率范围 300 ~ 115200bps，8 位数据位，奇偶校验位（无、奇、偶、标志、空格），1 或 2 位停止位。

## （2）通信报文格式

### ① 请求帧报文格式

[输出数据长度] [输出数据地址高字节] [输出数据地址低字节] [输入数据长度] [输入数据地址高字节] [输入数据地址低字节] [输出数据 1] ..... [输出数据 n] [校验]。

数据个数 n 等于输出数据长度。

### ② 响应帧报文格式

正确响应：

[输入数据长度] [输入数据地址高字节] [输入数据地址低字节] [输入数据 1] ..... [输入数据 n] [校验]。

数据个数 n 等于输入数据长度。

错误响应：

[0x00] [0xFF] [0xFF] [错误代码] [校验]

## （3）校验方式

所有数据的 8 位累加和，忽略溢出位。即：

[报文校验码] = [输出数据长度] + [输出数据地址高字节] + [输出数据地址低字节] + [输入数据长度] + [输入数据地址高字节] + [输入数据地址低字节] + [输出数据 1] + ..... + [输出数据 n]。

[响应校验码] = [输入数据长度] + [输入数据地址高字节] + [输入数据地址低字节] + [输出数据 1] + ..... + [输出数据 n]。

## （4）错误代码含义

错误代码	含义
0x01	输出数据长度错误
0x02	累加和校验错误
0x03	输出数据地址错误或输出数据区域不合法
0x04	输入数据地址错误或输入数据区域不合法

## (5) 报文示例

若读入输入数据 50 个字节，输出数据 32 个字节。

现在用户要输出数据为全 0，并读取所有输入数据，示例如下：

[以下均为 16 进制数]

请求帧报文：

[20]	[40 00]	[32]	[00 00]	[00.....00]	[92]
输出数据长度	输出起始地址	要读取的输入数据长度	输入起始地址	32 个输出数据	校验（累加和）

响应帧报文：

[32]	[00 00]	[00.....00]	[92]
输入数据长度	输入起始地址	50 个输出数据	校验（累加和）

这里的输出地址和输入地址，是 SS-430 内存映射地址。

### 4.3.3 通用模式

#### (1) 定义

SS-430 通用模式协议报文可以根据用户自由设定，解决了 Modbus 标准协议和 Modbus 非标准协议设备之间的通信问题。在通用模式下有两种工作方式：问答式、接收式（注：在配置列表中有 3 个选项，其中发送式不可用）。问答式工作机制与 Modbus 通信协议相似，采用请求响应的方式通信，每个子网在通用模式下最多可配置 30 条命令；接收式只接收存储数据，接收完数据后不作任何回应，例如和条形码扫描器设备通信等。

#### (2) 通用模式-问答式

使用通用模式-问答式之前用户需要配置通用模式-问答式的请求报文和响应报文。

帧头：16 进制输入，最大字节数 8

数据：16 进制输入，每个选项占两个字节

常量：16 进制输入，最大字节数 8

校验：无校验、CRC 校验、LRC 校验、和校验

帧尾：16 进制输入，最大字节数 3

在 RTU 格式下发送顺序：帧头、数据、常量、校验、帧尾

在 RTU 格式下接收顺序：帧头、数据、常量、校验、帧尾

在 ASCII 格式下发送顺序：帧头、常量、数据、校验、帧尾

在 ASCII 格式下接收顺序：帧头、常量、数据、校验、帧尾

**例如配置 Modbus 命令，RTU 传输格式：**

**请求：**

从站地址： 01

功能码： 03

寄存器地址 H： 00

寄存器地址 L： 00

数据数量 H： 00

数据数量 L： 02

CRC 校验 H: C4

CRC 校验 L: 0B

报文: 01 03 00 00 00 02 C4 0B

**响应:**

从站地址: 01

功能码: 03

数据个数: 04

数据: 00

数据: 00

数据: 00

数据: 00

CRC 校验 H: FA

CRC 校验 L: 33

报文: 01 03 04 00 00 00 00 FA 33

则配置如下图:



节点-1 命令-1

请求

☒ 帧头 01 03 00 00 00 02

☐ 数据

起始地址 0000 字节数 0010

位数 0016 位移量 0

☐ 常量

☒ 校验 CRC校验

☐ 帧尾 0d 0a

响应

☒ 帧头 01 03 04

☒ 数据

起始地址 0000 字节数 0004

位数 0016 位移量 0

☐ 常量

☒ 校验 CRC校验

☐ 帧尾 0d 0a

提示

确定 取消

注：在 RTU 传输格式下，校验支持：无校验、CRC 校验、和校验

例如配置 Modbus 命令，ASCII 传输格式：

：(3A)

地址

功能代码

数据数量

数据 1

.....

数据 n

LRC 高字节

LRC 低字节

回车 (0D)

换行 (0A)

示例配置如下图：

The screenshot shows a configuration window titled '节点-1 命令-1'. It is divided into three main sections: '请求' (Request), '响应' (Response), and '提示' (Prompt). Each section has a set of checkboxes and input fields.

**请求 (Request) Section:**

- ☒ 帧头: 3a
- ☐ 数据: 起始地址 0000, 字节数 0010, 位数 0016, 位移量 0
- ☒ 常量: 01 03 00 00 00 02
- ☒ 校验: LRC校验
- ☒ 帧尾: 0d 0a

**响应 (Response) Section:**

- ☒ 帧头: 3a
- ☒ 数据: 起始地址 0000, 字节数 0004, 位数 0016, 位移量 0
- ☒ 常量: 01 03 04
- ☒ 校验: LRC校验
- ☒ 帧尾: 0d 0a

**提示 (Prompt) Section:**

Empty text box.

Buttons at the bottom: 确定 (OK) and 取消 (Cancel).

注：在 ASCII 传输格式下，校验支持：无校验、LRC 校验、和校验

### (3) 通用模式-接收式

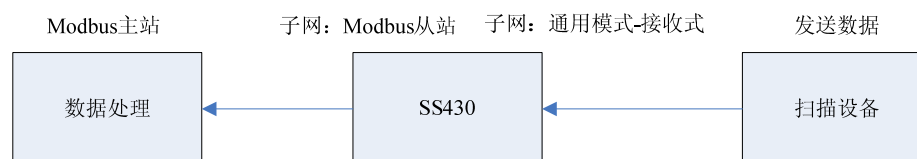
通用模式-接收式只接收数据不作应答，可用于接收条形码扫描设备的数据，通用模式-接收式每个子网有 16 组数据接收缓存区，每组接收数据缓存区大小为 255 个字节。

配置界面如下图：

协议类型选择	通用模式
Modbus通讯波特率	19200
数据位	8
奇偶校验方式	无
停止位	1
从站地址	
通讯传输模式	RTU
响应等待时间(300~60000ms)	300
轮循延时时间(0~2500ms)	0
输出命令轮循模式	连续输出
脉冲输出时间(200~2500ms)	
扫描比率(1~255)	10
通信方式	接收式
断帧超时时间	3
校验方式	无校验

通信实现方法：SS-430 一个子网配置为通用模式-接收式，连接到条形码扫描设备上；SS-430 另一个子网配置为 Modbus 从站，连接到带 Modbus 主站的网关上。Modbus 主站用 04H 功能码读取从寄存器 0 开始的数据，寄存器 0 为事物序列号，每次读取一条新的报文，事务序列号加 1，事务序列号从 0~255 循环。寄存器 1~寄存器 n 为接收条形码扫描设备的数据，具体读取的数据个数由 Modbus 主站配置来决定，最大有效字节数为 255。

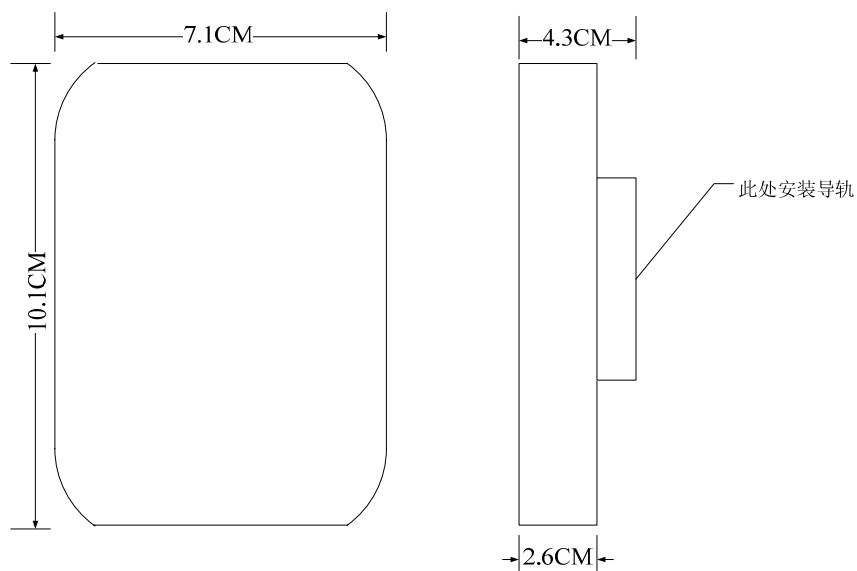
连接示意图如下：



## 五、安装

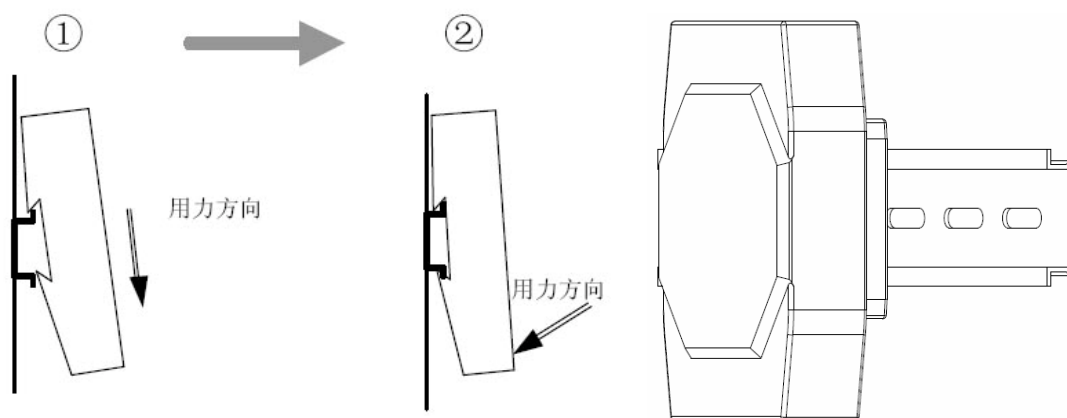
### 5.1 机械尺寸

尺寸：100mm x 70mm x 25mm [不包括导轨连接器]



### 5.2 安装方法

35mm DIN 导轨安装



## 六、运行维护及注意事项

- ✧ 模块需防止重压，以防面板损坏。
- ✧ 模块需防止撞击，有可能会损坏内部器件。
- ✧ 供电电压控制在说明书的要求范围内，以防模块烧坏。
- ✧ 模块需防止进水，进水后将影响正常工作。
- ✧ 上电前请检查接线，有无错接或者短路。

## 附录：Modbus 协议

### Modbus-RTU 协议：

说明：与本产品通讯的设备必须带有 Modbus 接口，同时设备 Modbus 协议必须符合下面的规定，本公司提供用户定制服务。

#### 1. 协议概述

物理层：传输方式：RS485

通讯地址：0-247

通讯波特率：可设定

通讯介质：屏蔽双绞线

传输方式：主从半双工方式。

协议在一根通讯线上使用应答式连接（半双工），这意味着在一根单独的通讯线上信号沿着相反的两个方向传输。首先，主计算机的信号寻址到一台唯一的终端设备（从机），然后，在相反的方向上终端设备发出的应答信号传输给主机。

协议只允许在主计算机和终端设备之间，而不允许独立的设备之间的数据交换，这就不会在使它们初始化时占据通讯线路，而仅限于响应到达本机的查询信号。

一个数据帧格式：

1 位起始位，8 位数据，1 位停止位。

一个数据包格式

地址	功能码	数据	校验码
8-Bits	8-Bits	N x 8-Bits	16-Bits

协议详细定义了校验码、数据序列等，这些都是特定数据交换的必要内容。

当数据帧到达终端设备时，它通过一个简单的“口”进入寻址到的设备，该设备去掉数据帧的“信封”（数据头），读取数据，如果没有错误，就执行数据所请求的任务，然后，它将自己生成的数据加入到取得的“信封”中，把数据帧返回给发送者。返回的响应数据中包含了以下内容：终端从机地址 (Address)、被执行了的命令(Function)、执行命令生成的被请求数据(Data)和一个校验码(Check)。发生任何错误都不会有成功的响应。

#### 地址（Address）域

地址域在帧的开始部分，由 8 位（0 ~ 255）组成，这些位标明了用户指定的终端设备的地址，该设备将接收来自与之相连的主机数据。每个终端设备的地址必须是唯一的，仅仅被寻址到的终端会响应包含了该地址的查询。当终端发送回一个响应，响应中的从机地址数据便告诉了主机哪台终端正与之进行通信。

#### 功能（Function）域

功能域代码告诉了被寻址到的终端执行何种功能。表 1-1 列出了所有的功能码、它们的意义及它们的初始功能。

表 1-1 功能码

代码	意义	行为
03	读数据	获得一个或多个寄存器的当前二进制值
06	预置单寄存器	放置一个特定的二进制值到一个单寄存器中
16	预置多寄存器	放置特定的二进制值到一系列多寄存器中

## 数据域

数据域包含了终端执行特定功能所需要的数据或者终端响应查询时采集到的数据。这些数据的内容可能是数值、参考地址或者极限值。例如：功能域码告诉终端读取一个寄存器，数据域则需要指明从哪个寄存器开始及读取多少个数据，内嵌的地址和数据依照类型和从机之间的不同能力而有所不同。

## 错误校验域

该域允许主机和终端检查传输过程中的错误。有时，由于电噪声和其它干扰，一组数据在从一个设备传输到另一个设备时在线路上可能会发生一些改变，出错校验能够保证主机或者终端不去响应那些传输过程中发生了改变的数据，这就提高了系统的安全性和效率，出错校验使用了 16 位循环冗余的方法。

[注] 发送序列总是相同的 – 地址、功能码、数据和与方向相关的出错校验。

## 错误检测

循环冗余校验（CRC）域占用两个字节，包含了一个 16 位的二进制值。CRC 值由发送设备计算出来，然后附加到数据帧上，接收设备在接收数据时重新计算 CRC 值，然后与接收到的 CRC 域中的值进行比较，如果这两个值不相等，就发生了错误。

CRC 运算时，首先将一个 16 位的寄存器预置为全 1，然后连续把数据帧中的 8 位字节与该寄存器的当前值进行运算，仅仅每个字节的 8 个数据位参与生成 CRC，起始位和终止位以及可能使用的奇偶位都不影响 CRC。

在生成 CRC 时，每个 8 位字节与寄存器中的内容进行异或，然后将结果向低位移位，高位则用“0”补充，最低位（LSB）移出并检测，如果是 1，该寄存器就与一个预设的固定值进行一次异或运算，如果最低位为 0，不作任何处理。

上述处理重复进行，知道执行完了 8 次移位操作，当最后一位（第 8 位）移完以后，下一个 8 位字节与寄存器中的当前值进行异或运算，同样进行上述的另一个 8 次移位异或操作，当数据帧中的所有字节都作了处理，生成的最终值就是 CRC 值。

生成一个 CRC 的流程为：

预置一个 16 位寄存器为 0FFFFH（全 1），称之为 CRC 寄存器。

把数据帧中的第一个 8 位字节与 CRC 寄存器中的低字节进行异或运算，结果存回 CRC 寄存器。将 CRC 寄存器向右移一位，最高位填以 0，最低位移出并检测。

如果最低位为 0：重复第三步（下一次移位）。

如果最低位为 1：将 CRC 寄存器与一个预设的固定值（0A001H）进行异或运算。

重复第三步和第四步直到 8 次移位。这样处理完了一个完整的八位。

重复第 2 步到第 5 步来处理下一个八位，直到所有的字节处理结束。

最终 CRC 寄存器得值就是 CRC 的值。

## 2. 应用层功能详解

第一章已经简述了协议和数据帧，使用此软件的程序员可以使用下述的方法以便通过协议正确的建立他们的特定应用程序。

本章所述协议将尽可能的使用如图 2-1 所示的格式，（数字为 16 进制）。

地址	功能码	变量起始地址高字节	变量起始地址低字节	变量的个数高字节	变量的个数低字节	校验码低字节	校验码高字节
03H	03H	00H	01H	00H	03H	55H	E9H

图 2-1 协议例述  
读数据（功能码 03）  
查询

图 2-2 的例子是从 03 号从机读 3 个采集到的基本数据 U1,U2,U3, U1 的地址为 0001H, U2 的地址为 0002H, U3 的地址为 0003H,

地址	功能码	变量起始地址高字节	变量起始地址低字节	变量的个数高字节	变量的个数低字节	校验码低字节	校验码高字节
03H	03H	00H	01H	00H	03H	55H	E9H

图 2-2 读 Uca 和 Ia 的查询数据帧  
响应

响应包含从机地址、功能码、数据的数量和 CRC 错误校验。

图 2-3 的例子是读取 U1,U2,U3 的响应。

地址	功能码	变量的总字节数	变量值高字节	变量值低字节	变量值高字节	变量值低字节	变量值高字节	变量值低字节	校验码低字节	校验码高字节
03H	03H	06H	01H	7CH	01H	7DH	01H	7CH	F9H	9BH

图 2-3 读 U1,U2,U3 的响应数据帧

## 2. 2 预置多寄存器（功能码 10） 查询

功能码 10H 允许用户改变多个寄存器的内容，设备可从任何地址开始设置最多 16 个变量的值。控制器是以动态扫描方式工作的，任何时刻都可以改变寄存器内容。

图 2-4 是修改 3 号从站设备的负载监控 1 和负载监控 2 的动作及延时时间的设定值，其中负载监控 1 的动作设定值地址为 2AH,延时时间的设定值为 2BH,负载监控 2 的动作设定值地址为 2CH,延时时间的设定值为 2DH。

地址	功能码	变量起始地址高字节	变量起始地址低字节	变量的个数高字节	变量的个数低字节	变量的总字节数	变量值高字节	变量值低字节	变量值高字节	变量值低字节	变量值高字节	变量值低字节	变量值高字节	变量值低字节	校验码低字节	校验码高字节
03H	10H	00H	2AH	00H	04H	08H	07H	D0H	00H	0AH	07H	0D0H	00H	0AH	25H	7CH

图示 2-4 修改负载监控 1 和负载监控 2 的动作值及延时时间的设定值  
响应

地址	功能码	变量起始地址高字节	变量起始地址低字节	变量的个数高字节	变量的个数低字节	校验码低字节	校验码高字节
03	10H	00H	2AH	00H	04H	EBH	8DH



图示 2-5 修改负载监控 1 和负载监控 2 的动作值及延时时间的设定值的响应

### 2. 3 预置单寄存器（功能码 06）

查询

功能码 06 允许用户改变单个寄存器的内容,DAE 系统内部的任何单寄存器都可以使用此命令来改变其值。既然仪器是以动态扫描方式工作的,任何时刻都可以改变单寄存器内容。

下面的例子是请求 03 号从机修改过载动作设定值 Ir1, Ir1 地址是 002EH.

地址	功能码	变量起始 地址高字节	变量起始 地址低字节	变量值 高字节	变量值低 字节	校验码 低字节	校验码 高字节
03H	06H	00H	2EH	07H	0D0H	EBH	8DH

图示 2-6 修改过载动作设定值 Ir1

响应

对于预置单寄存器请求的正常响应是在寄存器值改变以后将接收到的数据传送回去。

地址	功能码	变量起始 地址高字节	变量起始 地址低字节	变量值高 字节	变量值低 字节	校验码 低字节	校验码 高字节
03H	06H	00H	2EH	07H	0D0H	EBH	8DH

图示 2-7 图示 2-6 修改过载动作设定值 Ir1