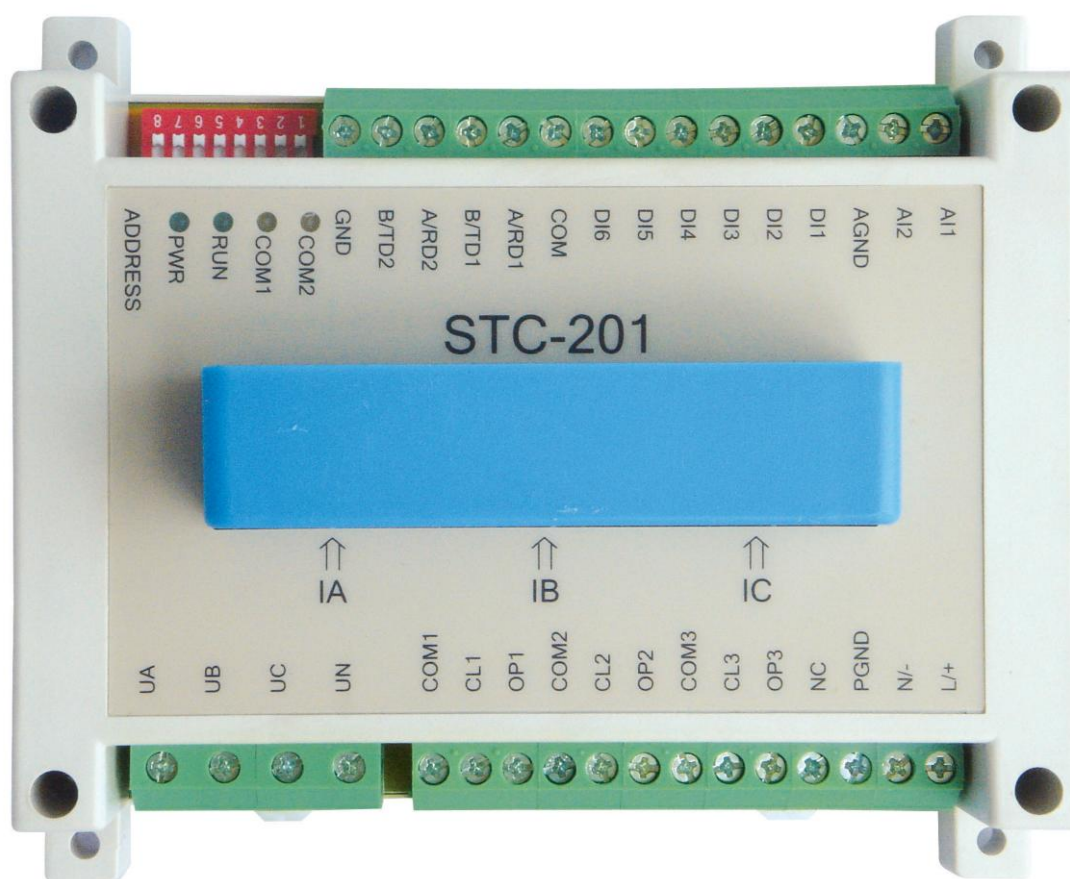


STC-201 微型 RTU

用户手册(V1.0)



北京易控微网科技有限公司

2007.1

1. 产品介绍	3
1.1. 产品设计说明	3
1.2. 功能特点	3
1.3. 系统参数	3
1.3.1. 直流模拟量输入	3
1.3.2. 交流模拟量输入	4
1.3.3. 开关量输入	4
1.3.4. 脉冲量输入（与开关量输入共用）	4
1.3.5. 开关量输出	5
1.3.6. 通信接口	5
1.4. 安装使用环境	5
2. 原理说明	5
2.1. 开关量输入	5
2.2. 开关量输出	6
2.3. 交流输入	6
2.4. 直流输入	7
2.5. 温度测量	7
3. 通信协议说明	8
3.1. MODBUS 规约简介	8
3.1.1. 在 Modbus 网络上转输	8
3.1.2. 在其它类型网络上转输	9
3.1.3. 查询—回应周期	9
3.2. 帧格式说明	9
3.3. 数据定义	11
4. 安装说明	23
5. 典型方案	26
6. 订货要求	28

1. 产品介绍

1.1. 产品设计说明

STC-201 是我公司针对用户对于工厂、企事业、机场、楼宇等单位高低压电力线路智能配电监控、工厂能源节能分析等存在众多需要电力监控的场合而专门开发的一种高性能、高集成度、高稳定性微型远动装置。一个装置实现一条三相回路的遥信、遥测、遥控，可以通过 485 总线组成分散式的测控系统，选择双 485 接口，还可以实现总线冗余（通道冗余）的高可靠性系统。STC-201 可以选择交流 220V 遥信或直流 110V 或 220V 遥信，具备 SOE，以及开关量可以作为脉冲输入实现抄表，在实际的工程运用中具有极强的优势。

1.2. 功能特点

- 三表法准确测量三相交流电压、电流、有功、无功、频率、功率因数、零序电流等电力参量，可以测量变频器输出。
- 具有 3 路独立的开关量输出，可以作为遥控、跳闸或者告警
- 6 路开关量输入（可以接入交流 220V 或直流 220V 等信号），也可以作为脉冲量输入，实现脉冲累计
- 2 路直流采样，可以接各种变送器输出的标准信号（0-5V 或 0-20mA 等）
- 两路隔离独立串行通信接口，支持 MODBUS 规约，支持通信总线冗余
- FFT 算法，可计算 1—7 次谐波
- 内置铁电存储器，实现重要参数掉电保存
- 一条三相电力线路的测控或一个风机水泵的监控，用一个模块实现
- 具有良好的宽温工作性能，适合真正分散式的测控需要

1.3. 系统参数

1.3.1. 直流模拟量输入

容量：	2 路/单元
输入信号：	0-5V 或 0-20mA 或 4-20mA
转换速度：	5 μ S
分辨率：	12 位 A/D
计算方法：	以电网频率为参考，每周波采 16 点，计算平均值作为测量值

精度： 0.2% FS $\pm$ 1 字

1.3.2. 交流模拟量输入

测量方法： 三表法或者二表法，角型或者星型接法

容量： 一条线路

测量对象： 三相相电压或者线电压，三相电流

输入电压： 0—100V 0—400V 长时间允许 50%过载，保证
150%线性范围

输入电流： 0—5A 长时间允许 800%过载，保证 120%线性范围

采样速度： 每周波采样 16 个点

计算量： UA,UB,UC(或者 UAB,UAC,UBC),IA,IB,IC,PA,PB,PC,
QA,QB,QC,P,Q,F,COS,3I0；各次谐波的实部、虚部；负序电流

数据刷新速度： 40MS

测量交流量频率： 20~65HZ 在 45~55HZ 间保证精度，在其他频率范围内
误差可能有所超差

频率测量精度： $\pm 0.01\text{HZ} \pm 1$ 字

精度： 0.5% FS ± 2 字 (不包括谐波、3I0，负序电流)

1.3.3. 开关量输入

容量： 6 路/单元

额定输入： 输入直流 12—48V 或直流 110V/220V，或交流 220V

输入方式： 光耦隔离，有信号输入时为“1”，无信号输入时为“0”

扫描方式： 中断方式

SOE 分辨率： 1ms

1.3.4. 脉冲量输入（与开关量输入共用）

容量： 6 路/单元

额定输入： 输入直流 12—48V 或直流 110V/220V，或交流 220V

输入方式： 光耦隔离

扫描方式： 中断方式

最高计数频率： 1000Hz

最小脉冲宽度： 0.2ms

1.3.5. 开关量输出

容量： 3 路/单元，每路提供一个常开或者常闭触点，每路相互独立
触点容量： 250V/3A 交流

1.3.6. 通信接口

容量： 2 路
接口方式： 2 路隔离 RS485 接口或 RS232 接口
规约： MODBUS 规约或者其他规约

1.4. 安装使用环境

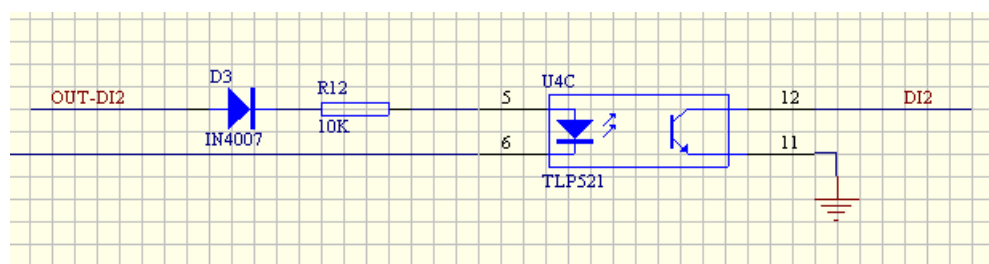
安装方式： 卡式导轨安装或者底部螺丝固定
温度范围： $-10^{\circ}\text{C} \sim 55^{\circ}\text{C}$
存贮温度： $-20^{\circ}\text{C} \sim 70^{\circ}\text{C}$
相对湿度： $<85\%$ ($20 \pm 5^{\circ}\text{C}$ 条件)
大气压力： $86 \sim 108\text{Kpa}$
安装尺寸： $145 \times 90 \times 93\text{mm}$ 或 $145 \times 90 \times 55$
电 源： 交流 $85 \sim 265\text{V}$ $50\text{HZ} \sim 60\text{HZ}$
或者用户要求
功 耗： 小于 2W
工 作 环 境： 无爆炸,无腐蚀性气体及导电尘埃,无严重霉菌存在,无剧烈振动,无冲击源;如果需要在此类环境下工作,请采取相应的防护措施。

2. 原理说明

STC-201 单元采用 16 位超低功耗单片机，单片机内部集成了几乎计算机所能外围电路，设计该产品时没有进行任何的扩展。提高了系统的可靠性和抗干扰能力，缩小了产品的体积。

2.1. 开关量输入

系统的数字输入采用光电隔离，原理图如下：



输入信号为高时，光耦导通，DI2=0。

单片机监测到信号的变化，产生中断，对其对应的计数器进行计数；单片机记下产生中断的时间，并启动一个软定时器计数。如果没有到达设定的时间信号又发生变换，则清除中断时间和事件记录。如果在设定的时间内信号没有发生变化，则作为输入改变，并记录改变的时间作为 SOE 记录。

开关量的输入是无源的，具有一个公共端子，使用时需要外接电源。每路输入的输入电流不超过 3ma，功率消耗很小。

一般情况下，输入的公共端子应该接开关量输入电源的负端，输入电源的正端接待测开关量的一端，另外一端接单元的输入端。

开关量输入不允许交流电源作为开关量输入电源，这有可能导致严重的错误。

2.2. 开关量输出

开关量输出为继电器输出，响应的时间较慢，不适合需要输出快速改变の場合。

开关量输出有返校输入，如果开关量输出回路不正确，则会返回错误，保证了系统的可靠性。但是无法校验继电器的输出触点。

继电器的触点是直接连接到输出端子的，没有经过任何的防弧、防短路措施，使用时请多加注意，必要时需要增加保险丝和电弧吸收回路等。继电器的输出触点主要用于控制中间继电器或者交流接触器，一般不要用来直接控制功率负载，不适合控制有非常大冲击电流的负载，如开关电源等。

输出一共 3 路，每路提供的是继电器的一副触点，一个为常开触点，一个为常闭触点，二者共用一个公共端。

三路开关量输出之间没有任何的电气连接，每路都可以单独使用。

2.3. 交流输入

交流输入包括 A、B、C 三相电压和电流。电流是直接把线穿入小型电流互感器的圆孔。电压则采用 4 个各端子。分别为 UA,UB,UC,UN；其中 UA 为 A 相输入；UB 为 B 相输入；UC 为 C 相输入；UN 接零线（在三角形接法时，

可以不接)。

输入的交流电压信号通过小型的 PT (电压互感器), 变换为交流 0.5V 的信号, 经过滤波处理, 滤除干扰信号, 然后进行电平平移, 使得原来的交流信号, 叠加 1/2 的 VREF, 直接送到 A/D 转换, 进行采样。

输入的电流信号, 通过导线穿入小 CT (电流互感器), CT 的输出接一个精密电阻, 变换成电压信号, 经过滤波处理, 滤除干扰信号, 然后进行电平平移, 使得原来的交流信号, 叠加 1/2 的 VREF, 直接送到 A/D 转换, 进行采样。

采样好的信号存入单片机的 RAM 中供软件处理。

在软件中, 我们每个周波采样 16 个点, 根据采样定理, 可以计算出输入信号的 8 次谐波。但是在应用中对奇次谐波更为关心。在数字信号处理中, 由于电网的频率是在变化的, 如果采样频率不是电网频率的整数倍, 就会有所谓的频谱泄漏问题, 详细内容请参考有关书籍。在该问题上我们采用了我们的提出软件跟踪算法, 效果非常优异。

对于 6 路输入信号, 进行 FFT 变换, 得出各次谐波的幅值和相角, 并且计算零序电流和负序电流。计算的方法和 FFT 变换请参考有关数字信号处理的书籍, 不是本手册的内容。

计算的结果存入 RAM 中, 供通信程序、保护程序等其他程序使用。

用户可以通过相应的软件得到这些数据。

2.4. 直流输入

直流输入一个是两路, 可以作为电压或者电流输入。作为电压输入时, 其输入阻抗为 170K; 作为电流输入时, 其输入阻抗为 250 欧姆。

输入的直流信号为了避免电网的干扰, 在软件设计时, 系统跟踪电网的频率。每个电网周波也作 16 点采样, 对采样值进行平均运算, 运算结果作为输入值。这样大大提高了系统对电网干扰的抑制能力。

2.5. 温度测量

温度测量可以采用温度变送器, 把热电阻或者热电偶的信号变为直流量送直流输入测量。

环境温度的测量是通过单片机内部的温度二极管进行测量的, 精度可以达到 ± 2 度, 这样非常方便测量电动机的温升。

3. 通信协议说明

我们的 STC-201 采用 MODBUS 规约,原因是该规约是开放的,文本容易得到,有着众多厂商的支持,协议本身也非常的简单。

MODBUS 规约是 MODICOM 公司开发的,版权归其所有。

我们的接口采用 RS485 接口,比 RS232 具有更高的通信速率和更远的通信距离。

根据我们设备的情况,我们仅仅实现了 MODBUS 的一个小型子集,没有完全实现其所有内容,已经能够满足我们所有的需要。

3.1. MODBUS 规约简介

MODBUS 规约是 MODICOM 公司开发的一个为很多厂商支持的开放规约 Modbus 协议是应用于电子控制器上的一种通用语言。通过此协议,控制器相互之间、控制器经由网络(例如以太网)和其它设备之间可以通信。它已经成为一通用工业标准。有了它,不同厂商生产的控制设备可以连成工业网络,进行集中监控。

此协议定义了一个控制器能认识使用的消息结构,而不管它们是经过何种网络进行通信的。它描述了控制器请求访问其它设备的过程,如果回应来自其它设备的请求,以及怎样侦测错误并记录。它制定了消息域格局和内容的公共格式。

当在 Modbus 网络上通信时,此协议决定了每个控制器须要知道它们的设备地址,识别按地址发来的消息,决定要产生何种行动。如果需要回应,控制器将生成反馈信息并用 Modbus 协议发出。在其它网络上,包含了 Modbus 协议的消息转换为在此网络上使用的帧或包结构。这种转换也扩展了根据具体的网络解决节地址、路由路径及错误检测的方法。

3.1.1. 在 Modbus 网络上转输

标准的 Modbus 口是使用 RS-232C 兼容串行接口,它定义了连接口的针脚、电缆、信号位、传输波特率、奇偶校验。控制器能直接或经由 Modem 组网。控制器通信使用主—从技术,即仅设备(主设备)能初始化传输(查询)。其它设备(从设备)根据主设备查询提供的数据做出相应反应。典型的主设备:主机和可编程仪表。典型的从设备:可编程控制器。

主设备可单独和从设备通信,也能以广播方式和所有从设备通信。如果单独通信,从设备返回消息作为回应,如果是广播方式查询的,则不作任何回应。Modbus 协议建立了主设备查询的格式:设备(或广播)地址、功能代码、所有要发送的数据、错误检测域。

从设备回应消息也由 Modbus 协议构成，包括确认要行动的域、任何要返回的数据、和错误检测域。如果在消息接收过程中发生错误，或从设备不能执行其命令，从设备将建立错误消息并把它作为回应发送出去。

3.1.2. 在其它类型网络上转输

在其它网络上，控制器使用对等技术通信，故任何控制都能初始和其它控制器的通信。这样在单独的通信过程中，控制器既可作为主设备也可作为从设备。提供的多个内部通道可允许同时发生的传输进程。

在消息位，Modbus 协议仍提供了主—从原则，尽管网络通信方法是“对等”。如果控制器发送消息，它只是作为主设备，并期望从从设备得到回应。同样，当控制器接收到消息，它将建立一从设备回应格式并返回给发送的控制器。

3.1.3. 查询—回应周期

.查询

查询消息中的功能代码告之被选中的从设备要执行何种功能。数据段包含了从设备要执行功能的任何附加信息。例如功能代码 03 是要求从设备读保持寄存器并返回它们的内容。数据段必须包含要告之从设备的信息：从何寄存器开始读及要读的寄存器数量。错误检测域为从设备提供了一种验证消息内容是否正确的方法。

.回应

如果从设备产生正常的回应，在回应消息中的功能代码是在查询消息中的功能代码的回应。数据段包括了从设备收集的数据：像寄存器值或状态。如果有错误发生，功能代码将被修改以用于指出回应消息是错误的，同时数据段包含了描述此错误信息的代码。错误检测域允许主设备确认消息内容是否可用。

3.2. 帧格式说明

控制器能设置为两种传输模式（ASCII 或 RTU）中的任何一种在标准的 Modbus 网络通信。用户选择想要的模式，包括串口通信参数（波特率、校验方式等），在配置每个控制器的时候，在 Modbus 网络上的所有设备都必须选择相同的传输模式和串口参数。

ASCII 模式

:	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	换 行
---	--------	-----------	-----------	---------	-----	----------	-------------	-------------	--------	--------

RTU 模式

地址	功能代码	数据数量	数据 1	...	数据 n	CRC 高字节	CRC 低字节
----	------	------	------	-----	------	---------	---------

所选的 ASCII 或 RTU 方式仅适用于标准的 Modbus 网络，它定义了在这些网络上连续传输的消息段的每一位，以及决定怎样将信息打包成消息域和如何解码。

当控制器设为在 Modbus 网络上以 ASCII（美国标准信息交换代码）模式通信，在消息中的每个 8Bit 字节都作为两个 ASCII 字符发送。这种方式的主要优点是字符发送的时间间隔可达到 1 秒而不产生错误。

代码系统

- 十六进制，ASCII 字符 0...9, A...F
- 消息中的每个 ASCII 字符都是一个十六进制字符组成

每个字节的位

- 1 个起始位
- 8 个数据位，最小的有效位先发送
- 1 个奇偶校验位，无校验则无
- 1 个停止位（有校验时），2 个 Bit（无校验时）

错误检测域

- LRC(纵向冗长检测)

地址域

消息帧的地址域包含两个字符（ASCII）或 8Bit（RTU）。可能的从设备地址是 0...247（十进制）。单个设备的地址范围是 1...247。主设备通过将要联络的从设备的地址放入消息中的地址域来选通从设备。当从设备发送回应消息时，它把自己的地址放入回应的地址域中，以便主设备知道是哪一个设备做出回应。

地址 0 是用作广播地址，以使所有的从设备都能认识。当 Modbus 协议用于更高水准的网络，广播可能不允许或以其它方式代替。

功能域

消息帧中的功能代码域包含了两个字符（ASCII）或 8Bits（RTU）。可能的代码范围是十进制的 1...255。当然，有些代码是适用于所有控制器，有此是应用于某种控制器，还有些保留以备后用。

当消息从主设备发往从设备时，功能代码域将告之从设备需要执行哪些行为。例如去读取输入的开关状态，读一组寄存器的数据内容，读从设备的诊断状态，允许调入、记录、校验在从设备中的程序等。

当从设备回应时，它使用功能代码域来指示是正常回应(无误)还是有某种错误

发生（称作异议回应）。对正常回应，从设备仅回应相应的功能代码。对异议回应，从设备返回一等同于正常代码的代码，但最重要的位置为逻辑 1。

我们目前所支持的功能码非常有限，主要包括：

- 01 READ COIL STATUS
- 02 READ INPUT STATUS
- 03 READ HOLDING REGISTERS
- 04 READ INPUT REGISTERS
- 05 FORCE SINGLE COIL
- 06 PRESET SINGLE REGISTERS
- 24 READ FIFO QUEUE

3.3. 数据定义

STC-201 通信数据定义：我们采用的 MODICOM 公司的 MODBUS 规约的 ASCII 或 RTU 方式。ASCII 波特率 9600BPS，1 位起始位，8 位数据位，1 位停止位，无校验。RTU 波特率 9600BPS，1 位起始位，8 位数据位，1 位停止位

帧格式：

ASCII 方式：

上位机发送：

： 单元地址 功能码 起始地址 读取点数 LRC 校验 CR LF

： 为帧起始标志

单元地址占用两个字节为 ASCII 字符举例：实际单元地址为 1F 对应的地址字符为 0X31,0X46

功能码占用两个字节为 ASCII 字符举例：功能码为 01 对应的功能码字符为 0X30,0X31

起始地址占用四个字节为 ASCII 字符举例：实际起始地址为 0002 对应的地址字符为 0X30,0X30,0X30,0X32，对应的实际开关量或者模拟量为第三个输入或者输出。

读取点数占用四个字节为 ASCII 字符 举例：读取点数为 0002 对应的字符为 0X30,0X30,0X30,0X32

LRC 校验为和校验，占用两个字节。计算方法可以参考如下的 C 源程序，需要注意的是首先计算 LRC 的值,然后把结果转换成为相应的 ASCII 字符串。比如 LRC 结果为 0X5F，则在规约中 LRC 的值为 0X35,0X46

```

unsigned char lrc(unsigned char *str,int lenth)
{
    unsigned char tmp;
    tmp=0;
    while (lenth-- )
    {
        tmp+= *str++;
    }
    return ((unsigned char)(-((char)tmp)));
}

```

CR=0X0D

LF=0X0A 为帧结束标志

在通信中，假设是各种不同类型的量的第一个起始地址为 0000，按照顺序排列的，不同的功能码实际读取的是不同类型的量。上位机按照程序可以顺序读取所有的量。

下位机响应：

：单元地址 功能码 字节数 数据 LRC 校验 CR LF

（除了字节数，其他的量定义与上面的内容完全相同。

字节数指的是数据的字节数，不包括帧起始标志、单元地址、功能码、LRC 校验、帧结束标志。字节数占用两个字节。比如：数据是两个 32 位整数，则字节数为 0x30,0x38,不是实际通信数据占用的字节数 16。）

RTU 方式：

发送：

单元地址 功能码 起始地址 读取点数 CRC 校验

帧定界采用 3.5 倍字符时间没有收到字符作为帧开始或者帧结束

CRC 校验计算：

```

const 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
```

};

```
const unsigned 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 crc(unsigned char *puchMsg , unsigned short usDataLen)
{
    unsigned char uchCRCHi = 0xFF ; /* high byte of CRC initialized */
    unsigned char uchCRCLo = 0xFF ; /* low byte of CRC initialized */
    unsigned uIndex ; /* will index into CRC lookup table */
    while (usDataLen--)/ * pass through message buffer */
    {
        uIndex = uchCRCHi ^ *puchMsg++ ; /* calculate the CRC */
        uchCRCHi = uchCRCLo ^ uchCRCHi[uIndex];
        uchCRCLo = uchCRCLo[uIndex] ;
    }
    return (uchCRCHi << 8 | uchCRCLo) ;
}
```

接收:

单元地址 功能码 字节数 数据 CRC 校验

功能码: 01

数据起始地址: 00

数据长度: 不大于 08 (因为只有三个输出继电器)

说明: 读取输出继电器的状态。

读到的 8 位数据的低三位为继电器状态, 第 2 位对应继电器的第三路,
第 1 位对应第二路, 第 0 位对应第一路。

位内容 0 表示继电器不吸合 1 表示继电器吸合

功能码: 02

数据起始地址: 00

数据长度: 不大于 08 (因为只有 6 路输入)

说明: 读取输入开关量的状态。

读到的 8 位数据的低六位为输入状态, 第 5 位对应输入开关量的第 6 路, 第 4 位对应第 5 路, ……., 第 0 位对应第 1 路, 位内容 0 表示输入没有加上电源, 1 表示输入已加上电源

输入的量改变状态需要经过 0.1S 滤波抗干扰后才会变化。如果用户想知道改变

的时刻，需要读取系统的 FIFO 中的 SOE 事件记录

输入量的每次改变都会累积到该路对应的 32 位计数器上，对于计数，输入的最高频率可以是 1KHz

功能码： 03

数据起始地址：00—80

数据长度：ASCII 不大于 32 RTU 不大于 64 个（因为通信缓冲区的限制）

说明： 读取保持寄存器的值。

读取的是 16 位整数或者无符合整数。

数据定义：

地址	数据内容
0000	系统实际时间的低 16 位 （0000—0020 为无符号整数）
0001	系统实际时间的高 16 位
0002	第一路开关量 32 位计数器低 16 位
0003	第一路开关量 32 位计数器高 16 位
0004	第二路开关量 32 位计数器低 16 位
0005	第二路开关量 32 位计数器高 16 位
0006	第三路开关量 32 位计数器低 16 位
0007	第三路开关量 32 位计数器高 16 位
0008	第四路开关量 32 位计数器低 16 位
0009	第四路开关量 32 位计数器高 16 位
000A	第五路开关量 32 位计数器高 16 位
000B	第五路开关量 32 位计数器低 16 位
000C	第六路开关量 32 位计数器高 16 位
000D	第六路开关量 32 位计数器低 16 位
000E	A 相有功电度 32 位计数器低 16 位
000F	A 相有功电度 32 位计数器高 16 位
0010	A 相无功电度 32 位计数器低 16 位
0011	A 相无功电度 32 位计数器高 16 位
0012	B 相有功电度 32 位计数器低 16 位
0013	B 相有功电度 32 位计数器高 16 位
0014	B 相无功电度 32 位计数器低 16 位
0015	B 相无功电度 32 位计数器高 16 位
0016	C 相有功电度 32 位计数器低 16 位
0017	C 相有功电度 32 位计数器高 16 位

0018	C 相无功电度 32 位计数器低 16 位
0019	C 相无功电度 32 位计数器高 16 位
001A	总有功电度 32 位计数器低 16 位
001B	总有功电度 32 位计数器高 16 位
001C	总无功电度 32 位计数器低 16 位
001D	总无功电度 32 位计数器高 16 位
001E	保留
001F	保留
0020	A 相电压系数, 用于系统调校
0021	A 相电流系数, 用于系统调校
0022	B 相电压系数, 用于系统调校
0023	B 相电流系数, 用于系统调校
0024	C 相电压系数, 用于系统调校
0025	C 相电流系数, 用于系统调校
0026	A 相有功系数, 用于系统调校
0027	A 相无功系数, 用于系统调校
0028	B 相有功系数, 用于系统调校
0029	B 相无功系数, 用于系统调校
002A	C 相有功系数, 用于系统调校
002B	C 相无功系数, 用于系统调校
002C	串口 1 设置
002D	串口 2 设置
002C~002D 内容用 16 进制表示为 0XPQRS QR 保留	
P 高 4 为奇偶校验设置。P=D(13)为 1,8,E, P=1 为 1,8,N	
S=0	波特率 9600
1	波特率 300
2	波特率 600
3	波特率 1200
4	波特率 2400
5	波特率 4800
6	波特率 9600
7	波特率 19200
8	波特率 38400
9	波特率 57600

A 波特率 76800

B 波特率 115200

C ~F 9600

通电前把地址开关拨成地址 0,这时通讯地址为 1,波特率为 9600 1, 8, N

002E 直流 0 系数

002F 直流 1 系数

电度数 = (低 16 位 + 高 16 位 $\times 65536$) / 1000 $\times$ 电压互感器变比 $\times$ 电流互感器变比

功能码: 04

数据起始地址: 00--127

数据长度: ASCII 不大于 32 RTU 不大于 64 个 (因为通信缓冲区的限制)

说明: 读取输入数据。

读出值为 16 位整数

数据定义:

地址 功能

0000 A 相电压 1000 为 100.0V

0001 B 相电压 1000 为 100.0V

0002 C 相电压 1000 为 100.0V

0003 A 相电流 500 对应 5A

0004 B 相电流 500 对应 5A

0005 C 相电流 500 对应 5A

0006 零序电流 500 对应 5A

0007 A 相有功 500 对应 500W

0008 B 相有功 500 对应 500W

0009 C 相有功 500 对应 500W

000A A 相无功 500 对应 500VAR

000B B 相无功 500 对应 500VAR

000C C 相无功 500 对应 500VAR

000D 三相有功 500 对应 500W

000E 三相无功 500 对应 500VAR

000F 功率因数 值为 1000 对应 1.000

0010 频率 值为 5000 对应 50Hz

0011	直流 1	值为 5000/2000	对应 5.00V 或 20mA
0012	直流 2	值为 5000/2000	对应 5.00V 或 20mA
0013	环境温度	值为 1000	对应 100 摄氏度
0014	机内电源电压	值为 3300	对应 3.3V
0015	A 相电压一次谐波实部		
0016	A 相电压一次谐波虚部		
0017	B 相电压一次谐波实部		
0018	B 相电压一次谐波虚部		
0019	C 相电压一次谐波实部		
001A	C 相电压一次谐波虚部		
001B	A 相电流一次谐波实部		
001C	A 相电流一次谐波虚部		
001D	B 相电流一次谐波实部		
001E	B 相电流一次谐波虚部		
001F	C 相电流一次谐波实部		
0020	C 相电流一次谐波虚部		
0021	A 相电压二次谐波实部		
0022	A 相电压二次谐波虚部		
0023	B 相电压二次谐波实部		
0024	B 相电压二次谐波虚部		
0025	C 相电压二次谐波实部		
0026	C 相电压二次谐波虚部		
0027	A 相电流二次谐波实部		
0028	A 相电流二次谐波虚部		
0029	B 相电流二次谐波实部		
002A	B 相电流二次谐波虚部		
002B	C 相电流二次谐波实部		
002C	C 相电流二次谐波虚部		
002D	A 相电压三次谐波实部		
002E	A 相电压三次谐波虚部		
002F	B 相电压三次谐波实部		
0030	B 相电压三次谐波虚部		

0031 C 相电压三次谐波实部
0032 C 相电压三次谐波虚部
0033 A 相电流三次谐波实部
0034 A 相电流三次谐波虚部
0035 B 相电流三次谐波实部
0036 B 相电流三次谐波虚部
0037 C 相电流三次谐波实部
0038 C 相电流三次谐波虚部

0039 A 相电压四次谐波实部
003A A 相电压四次谐波虚部
003B B 相电压四次谐波实部
003C B 相电压四次谐波虚部
003D C 相电压四次谐波实部
003E C 相电压四次谐波虚部
003F A 相电流四次谐波实部
0040 A 相电流四次谐波虚部
0041 B 相电流四次谐波实部
0042 B 相电流四次谐波虚部
0043 C 相电流四次谐波实部
0044 C 相电流四次谐波虚部

0045 A 相电压五次谐波实部
0046 A 相电压五次谐波虚部
0047 B 相电压五次谐波实部
0048 B 相电压五次谐波虚部
0049 C 相电压五次谐波实部
004A C 相电压五次谐波虚部
004B A 相电流五次谐波实部
004C A 相电流五次谐波虚部
004D B 相电流五次谐波实部
004E B 相电流五次谐波虚部
004F C 相电流五次谐波实部
0050 C 相电流五次谐波虚部

- 0051 A 相电压六次谐波实部
- 0052 A 相电压六次谐波虚部
- 0053 B 相电压六次谐波实部
- 0054 B 相电压六次谐波虚部
- 0055 C 相电压六次谐波实部
- 0056 C 相电压六次谐波虚部
- 0057 A 相电流六次谐波实部
- 0058 A 相电流六次谐波虚部
- 0059 B 相电流六次谐波实部
- 005A B 相电流六次谐波虚部
- 005B C 相电流六次谐波实部
- 005C C 相电流六次谐波虚部

- 005D A 相电压七次谐波实部
- 005E A 相电压七次谐波虚部
- 005F B 相电压七次谐波实部
- 0060 B 相电压七次谐波虚部
- 0061 C 相电压七次谐波实部
- 0062 C 相电压七次谐波虚部
- 0063 A 相电流七次谐波实部
- 0064 A 相电流七次谐波虚部
- 0065 B 相电流七次谐波实部
- 0066 B 相电流七次谐波虚部
- 0067 C 相电流七次谐波实部
- 0068 C 相电流七次谐波虚部

- 0069 A 相功率因数
- 006A B 相功率因数
- 006B C 相功率因数
- 006C 正序电流
- 006D 负序电流

- 0069 --007f 保留

功能码： 05

数据起始地址： 0000—0002

数据长度：

说明： 设置单个继电器。

ASCII 方式

继电器吸合：

上位机发送：

： 单元地址 05 起始地址 FF00 LRC CR LF

下位机响应：

： 单元地址 05 起始地址 FF00 LRC CR LF

详细说明参考 MODBUS 规约。

继电器释放：

上位机发送：

： 单元地址 05 起始地址 FF00 LRC CR LF

下位机响应：

： 单元地址 05 起始地址 FF00 LRC CR LF

详细说明参考 MODBUS 规约。

RTU 方式

继电器吸合：

上位机发送：

单元地址 05 起始地址 FF00 CRC

下位机响应：

单元地址 05 起始地址 FF00 CRC

详细说明参考 MODBUS 规约。

继电器释放：

上位机发送：

单元地址 05 起始地址 FF00 CRC

下位机响应：

单元地址 05 起始地址 FF00 CRC

功能码： 06

数据起始地址： 0000—0080

数据长度：

说明： 设置保持寄存器的值。

设置的是 16 位整数。

各个寄存器的说明参考上面读取保持寄存器

ASCII 方式

上位机发送:

: 单元地址 06 起始地址 数据 LRC CR LF

下位机响应:

: 单元地址 06 起始地址 数据 LRC CR LF

数据为 16 位整数

详细说明参考 MODBUS 规约。

RTU 方式:

上位机发送:

单元地址 06 起始地址 数据 CRC

下位机响应:

单元地址 06 起始地址 数据 CRC

数据为 16 位整数

详细说明参考 MODBUS 规约。

功能码: 24

数据起始地址: 0000—0011

数据长度:

说明: 读取 FIFO 的值。每次读取一个事件, 如果没有事件则产生错误。

读取的是 16 位整数。

用于 SOE 读取, 每个 SOE 包括 4 个寄存器, 每个寄存器都是 16 位, 第一个为动作的点编号, 第二个为动作类型, 第三个为动作时间的高 16 位, 第四个动作时间的低 16 位。

动作类型取值 01 为 SOE 事件为从 0—>1 事件 也就是输入信号从失电变为得电

动作类型取值 00 为 SOE 事件为从 1—>0 事件 也就是输入信号从得电变为失电

ASCII 方式:

读取:

上位机发送:

: 单元地址 功能码 FIFO 地址 LRC CR LF

FIFO 地址一般设为 0000

下位机响应:

: 单元地址 功能码 数据长度 FIFO 数据 LRC CR LF

数据长度为 16 位数据, 占 4 个字节

FIFO 长度为 16 位数据, 占 4 个字节

FIFO 数据每个数据 16 位数据, 占 4 个字节

RTU 方式:

上位机发送:

板地址 功能码 FIFO 地址 CRC

FIFO 地址一般设为 0000

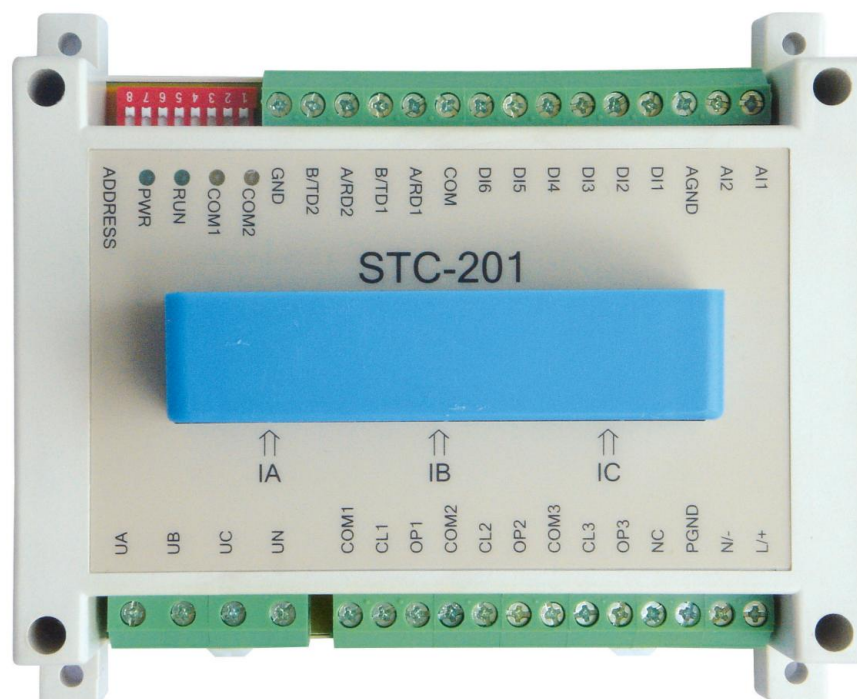
下位机响应:

板地址 功能码 数据长度 FIFO 长度 数据 CRC

数据长度为 16 位, FIFO 长度为 16 位 数据是 16 位数据

4. 安装说明

STC-201 安装非常简单。首先是固定 STC-201, 安装的方法: 如果是有 IEC 标准导轨, 直接卡装在导轨上即可。如果没有导轨, 需要用四个螺丝把 STC-201 安装在机柜上。



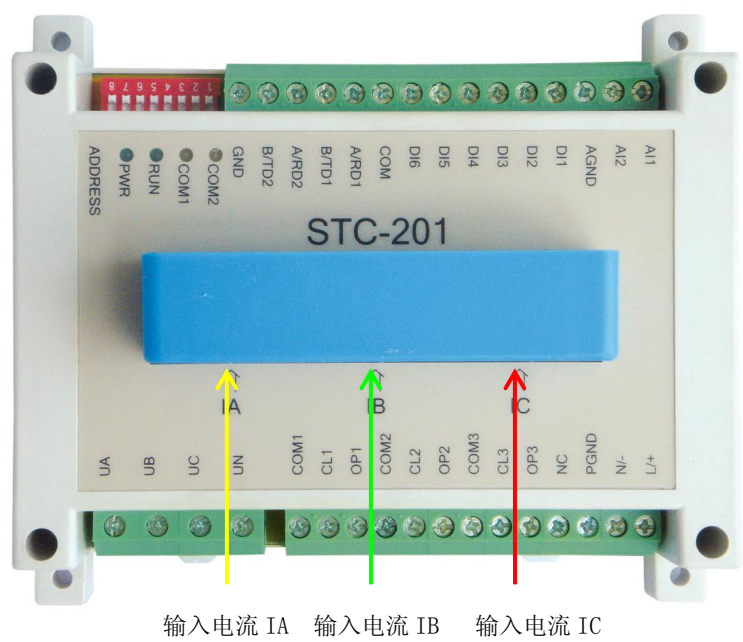
参照下面说明接线端子定义和接线

如上图所示，端子定义按序对应关系如下：

UA	A 相电压输入
UB	B 相电压输入
UC	C 相电压输入
UN	星形接法是接零线，三角形接法时不接线
COM1	第一组继电器输出，COM1 为公共端，CL1 为常闭触点，OP1 为常开触点
CL1	
OP1	
COM2	第二组继电器输出，COM2 为公共端，CL2 为常闭触点，OP2 为常开触点
CL2	
OP2	
COM3	第三组继电器输出，COM3 为公共端，CL3 为常闭触点，OP3 为常开触点
CL3	
OP3	
PGND	保护地，接机壳或大地
N/—	
L/+	
AI1	第一路模拟量输入
AI2	第二路模拟量输入
AGND	模拟信号地
DI1	开关量和脉冲量输入 1
DI2	开关量和脉冲量输入 2
DI3	开关量和脉冲量输入 3
DI4	开关量和脉冲量输入 4
DI5	开关量和脉冲量输入 5
DI6	开关量和脉冲量输入 6
COM	开关量和脉冲量公共端
A/RD1	第一路 485 A 线或第一路 232 接口 RXD 线

B/TD1	第一路 485	B 线或第一路 232 接口 TXD 线
A/RD2	}	第二路 485 A 线或第二路 232 接口 RXD 线
B/TD2		第二路 485 B 线或第二路 232 接口 TXD 线
GND		数字地 RS485 或者 RS232 地

为电流输入穿入互感器小孔方向



如果电压或者电流的方向接错，会引起下列故障：单相功率计算值为负值，可能引起负序电流和零序电流计算错误。

8 位拨码开关 选择本模块的地址（也就是 MODBUS 规约中的单元地址），便于组网。拨码开关标有数字‘1’的为最低位，标有‘8’的为最高位，ON 的位置为‘0’，单元地址按照二进制表示。仅仅使用最低的五位，高三位系统保留。

比如：1 ON 2ON 3 OFF 4 ON 5 ON 对应的单元地址为 04 。

0 0 1 0 0

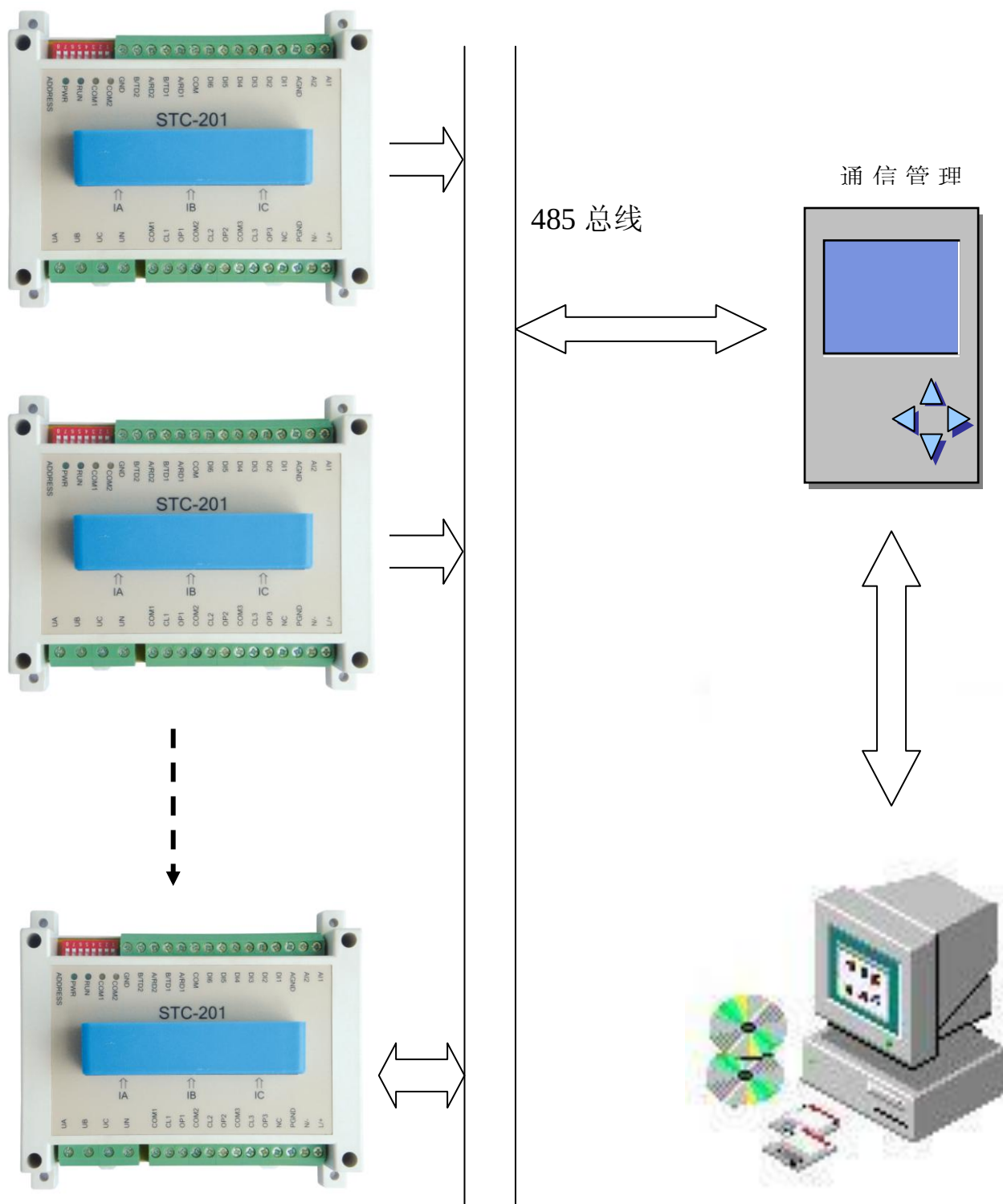
单元地址：00 或者 FF 是广播地址，请不要使用，否则会引起系统故障。

- 由于端子的比较小，安装时需要比较小的一字起子或者仪表起子
- 导线的线径除了电流引线外，不要超过 2.5 平方毫米，否则不易安装
- 如果工作场合干扰严重，请把 STC-201 放入铁制的机箱内，并对电源加电源滤波器
- 如果需要 IPC55 的工作条件，可以把 STC-201 安装在 IPC55 防护的机箱内

485 总线

计算机可以采用一般的工业控制计算机，软件可以采用多种工控软件，（如国产的组态王，昆仑通态、国外的 INTOUCH 等）或其他的软件。

STC-201 通过通信管理机与计算机连接，可以组成大型的测控系统



6. 订货要求

为了避免差错和不必要地麻烦，在订货前请准确提供如下信息：

工作电源：☐220V 交流 ☐110V 交流 ☐24V DC ☐5V DC
 ☐220V 直流 ☐110V 直流 ☐12V DC ☐48V DC
直流输入 1：☐0-5V ☐0-20mA
直流输入 2：☐0-5V ☐0-20mA

额定输入交流电压：☐ 100V ☐400V
额定输入交流电流：☐ 5A ☐其他请注明

是否测量变频器输出：☐是 ☐否

开关量输入信号电压：☐DC12/24V ☐AC220V ☐DC110/220V
是否需要脉冲量保存：☐是 ☐否

规约：☐MODBUS ASCII ☐MODBUS RTU
通信接口：☐ 2 路 RS485 接口 ☐1 路 RS485 接口，1 路 RS232 接口

台数：

注意！

本产品不防爆，不允许在有爆炸危险的环境下使用，若要用必须安装于相应等级防爆箱内；本品不允许在医院、核电、航空等可能对生命有危险或者有着巨大安全风险的场合使用；本公司不对任何使用者或任何第三方遭受的间接的、偶然的、特殊的、相应而生的信用或投保的损害，包括利润、收入、数据或使用的损失负责。

版权声明：

本文档版权归易控微网公司所有，并受到法律保护。

公司地址

公司地址：北京市海淀区紫竹院路广源闸 5 号广源大厦 312

邮政编码：100081

电话：010-59790086 传真：010-68703551

网址：<http://www.tengcon.com>

技术：tech@tengcon.com 技术支持：support@tengcon.com

市场：market@tengcon.com 销售：sales@tengcon.com

开户行：华夏银行北京紫竹桥支行

帐户：801924026

公司所在位置地图：

