SDA 机组
通信协议

版本 V1.0

目录

一、通信简介

1.1、RS485 串行通讯板技术指标

电源：用插接端子取自 K200

电耗：20mA

储存条件：-10~70℃，＜80%rH，无冷凝

工作条件：0~65℃，＜80%rH，无冷凝

尺寸：48×45mm

防护等级：IP00

环境污染：正常

表面温度极限：同工作温度

防电击等级：可装入Ⅰ级或Ⅱ级设备

阻热及阻燃类别：D 类

材料绝缘：250V

串行输出：3 线螺接端子，线径 0.2~1.5 mm^2

标准：光电隔离型异步 RS485

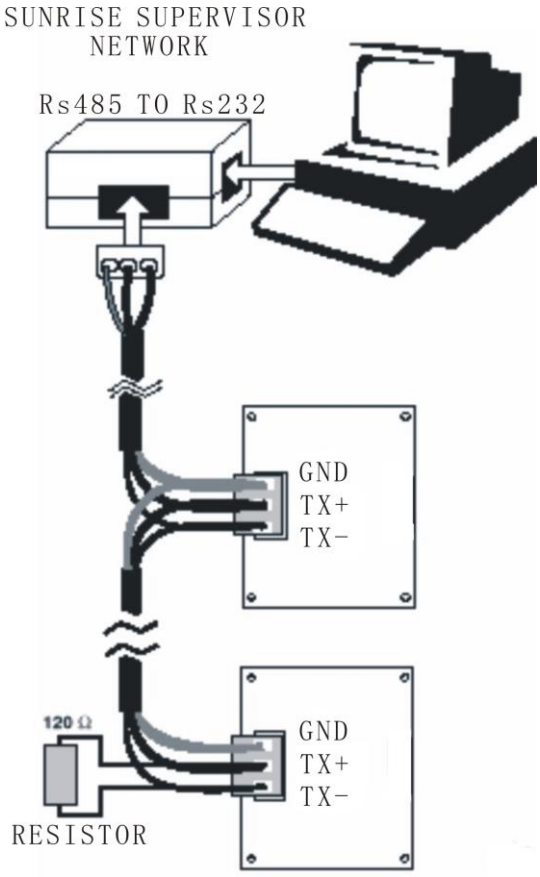
最高速率：19200 波特率

最大设备数：200

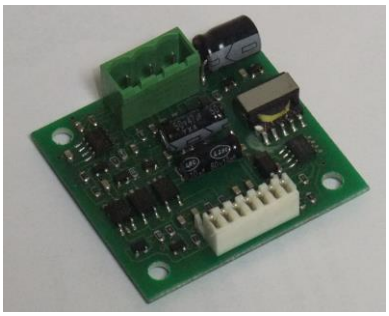
距监控设备最长距离：1km

电缆：1 对双绞线及屏蔽，美国线规 20/22 号，线间电容＜90pF/m(即 BELDEN8761-8762 电缆)

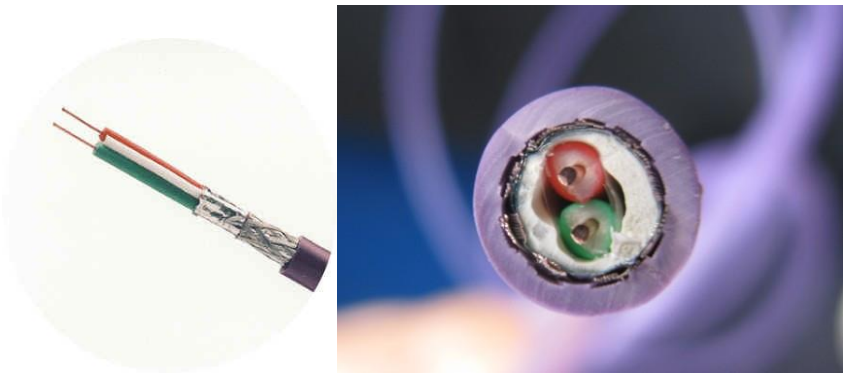
电击保护：本设备仅提供 K400 电源与串行线路间的功能性绝缘，因而 K400 必须采用安全型变压器。



通信接口板照片：



推荐连接电缆照片：



1.2、控制器参数设定

K200 系列控制器通信：将 RS485 通信接口板插入控制器主板的 7 芯插针上，通信协议选择采用 MODBUS-RTU。

注意任何对控制器的硬件操作必须在空调主机断电的条件下操作！

控制器通电后，如需实现监控，必须设置几项参数：

1.2.1、按一下 MENU 键，并通过上下键选择后进入“用户参数”->“密码：22”->“通信协议选择”，选择协议 2，即 MODBUS 协议

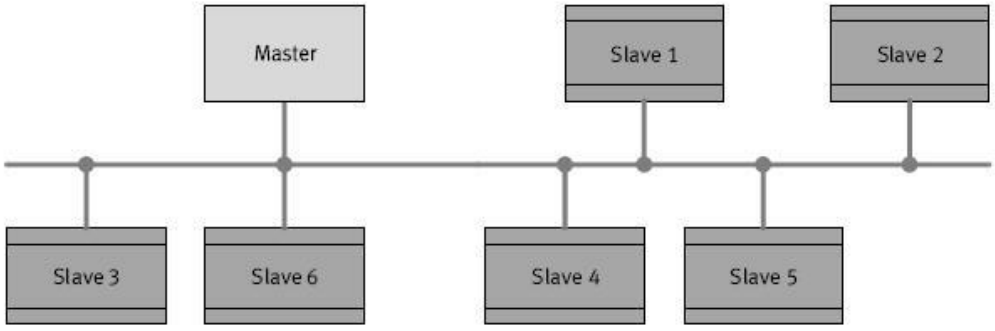
1.2.2、“用户参数”->“密码：22”->“机组群控地址”，设置机组在 485 网络中的地址，同一网络中不能有相同的地址，否则整个网络将无法通信。

1.2.3、“用户参数”->“密码：22”->“波特率选择”，1 代表 1200，2 代表 2400，3 代表 4800，4 代表 9600，5 代表 19200。默认值为 5（19200），强烈建议用户选择此波特率。

通信参数改变后，整个控制系统需断电 2 秒，重新上电后参数生效。

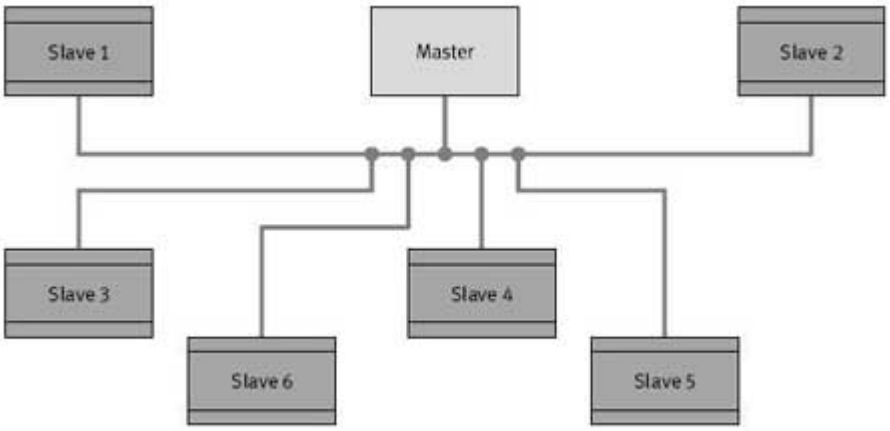
1.3、RS485 网络拓扑结构

RS485 总线只能采用总线制拓扑结构。



正确

SDA 通信协议



错误

二、Modbus 协议介绍

2.1、Modbus 协议简介

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识别使用的消息结构,而不管它们是经过何种网络进行通信的。它描述了一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。

当在一 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。

2.1.1、在 Modbus 网络上转输

标准的 Modbus 口是使用一 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，即仅一设备（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作出相应反应。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，从设备返回一消息作为回应，如果是以广播方式查询的，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、一错误检测域。

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、和一错误检测域。如果在消息接收过程中发生一错误，或从设备不能执行其命令，从设备将建立一错误消息并把它作为回应发送出去。

2.1.2、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”。如果一控制器发送一消息，它只是作为主设备，并期望从从设备得到回应。同样，当控制器接收到一消息，它将建立一从设备回应格式并返回给发送的控制器。

2.1.3、查询----回应周期

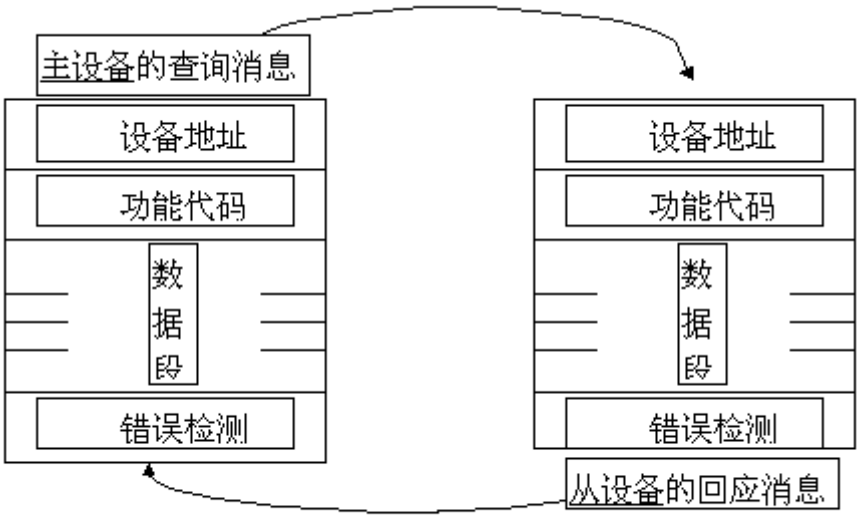


图 1 主—从 查询—回应周期表

（a）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。

（2）回应

如果从设备产生一正常的回应，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：象寄存器值或状态。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。

2.2、Modbus 传输方式

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，在一个 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。本协议中采用 RTU 模式。

RTU 模式如下:

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

图 2 RTU 模式

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。

在其它网络上（象 MAP 和 Modbus Plus）Modbus 消息被转成与串行传输无关的帧。

当控制器设为在 Modbus 网络上以 RTU（远程终端单元）模式通信，在消息中的每个 8Bit 字节包含两个 4Bit 的十六进制字符。这种方式的主要优点是：在同样的波特率下，可比 ASCII 方式传送更多的数据。

数据格式：

- 8 位二进制，十六进制数 0...9，A...F
- 消息中的每个 8 位域都是一个两个十六进制字符组成

每个字节的 Bit 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：

- CRC(循环冗长检测)

2.3、Modbus 消息帧

两种传输模式中（ASCII 或 RTU），传输设备以将 Modbus 消息转为有起点和终点的帧，这就允许接收的设备在消息起始处开始工作，读地址分配信息，判断哪一个设备被选中（广播方式则传给所有设备），判知何时信息已完成。部分的消息也能侦测到并且错误能设置为返回结果。

使用 RTU 模式，消息发送至少要以 3.5 个字符时间的停顿间隔开始。在网络波特率下多样的字符时间，这是最容易实现的(如下图的 T1-T2-T3-T4 所示)。传输的第一个域是设备地址。可以使用的传输字符是十六进制的 0...9,A...F。网络设备不断侦测网络总线，包括停顿间隔时间内。当第一个域（地址域）接收到，每个设备都进行解码以判断是否发往自己的。在最后一个传输字符之后，一个至少 3.5 个字符时间的停顿标定了消息的结束。一个新的消息可在此停顿后开始。

整个消息帧必须作为一连续的数据流传输。如果在帧完成之前有超过 1.5 个字符时间的停顿时间，接收设备将刷新不完整的消息并假定下一字节是一个新消息的地址域。同样地，如果一个新消息在小于 3.5 个字符时间内接着前个消息开始，接收的设备将认为它是前一消息的延续。这将导致一个错误，因为在最后的 CRC 域的值不可能是正确的。一典型的消息帧如下所示：

起始位	设备地址	功能代码	数据	CRC 校验	结束符
T1-T2-T3-T4	8Bit	8Bit	n 个 8Bit	16Bit	T1-T2-T3-T4

图 3 RTU 数据帧

2.3.1、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作出回应。

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。

2.3.2、如何处理功能域

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。

例如：一从主设备发往从设备的消息要求读一组保持寄存器，将产生如下功能代码：

0 0 0 0 0 0 1 1 （十六进制 03H）

对正常回应，从设备仅回应同样的功能代码。对异议回应，它返回：

1 0 0 0 0 0 1 1 （十六进制 83H）

除功能代码因异议错误作了修改外，从设备将一独特的代码放到回应消息的数据域中，这能告诉主设备发生了什么错误。

主设备应用程序得到异议的回应后，典型的处理过程是重发消息，或者诊断发给从设备的消息并报告给操作员。

2.3.3、数据域

数据域是由两个十六进制数集合构成的，范围 00...FF。根据网络传输模式，这可以是由一对 ASCII 字符组成或由一 RTU 字符组成。

从主设备发给从设备消息的数据域包含附加的信息：从设备必须用于进行执行由功能代码所定义的所为。这包括了象不连续的寄存器地址，要处理项的数目，域中实际数据字节数。

例如，如果主设备需要从设备读取一组保持寄存器（功能代码 03），数据域指定了起始寄存器以及要读的寄存器数量。如果主设备写一组从设备的寄存器(功能代码 10 十六进制)，数据域则指明了要写的起始寄存器以及要写的寄存器数量，数据域的数据字节数，要写入寄存器的数据。

如果没有错误发生，从从设备返回的数据域包含请求的数据。如果有错误发生，此域包含一异议代码，主设备应用程序可以用来判断采取下一步行动。

在某种消息中数据域可以是不存在的（0 长度）。例如，主设备要求从设备回应通信事件记录（功能代码 0B 十六进制），从设备不需任何附加的信息。

2.3.4、错误检测域

标准的 Modbus 网络有两种错误检测方法。错误检测域的内容视所选的检测方法而定。当选用 RTU 模式作字符帧，错误检测域包含一 16Bits 值(用两个 8 位的字符来实现)。错误检测域的内容是通过对消息内容进行循环冗长检测方法得出的。CRC 域附加在消息的最后，添加时先是低字节然后是高字节。故 CRC 的高位字节是发送消息的最后一个字节。

2.3.5、字符的连续传输

当消息在标准的 Modbus 系列网络传输时，每个字符或字节以如下方式发送（从左到右）：

最低有效位……最高有效位

使用 RTU 字符帧时，位的序列是(无奇偶校验)：

启始位	1	2	3	4	5	6	7	8	停止位	停止位

图 4. 位顺序（RTU）

表 1 ModBus 功能码

功能码	名称	作用
01	读取线圈状态	取得一组逻辑线圈的当前状态（ON/OFF)
02	读取输入状态	取得一组开关输入的当前状态（ON/OFF)
03	读取保持寄存器	在一个或多个保持寄存器中取得当前的二进制值
04	读取输入寄存器	在一个或多个输入寄存器中取得当前的二进制值
05	强置单线圈	强置一个逻辑线圈的通断状态
06	预置单寄存器	把具体二进值装入一个保持寄存器
15	强置多线圈	强置一串连续逻辑线圈的通断
16	预置多寄存器	把具体的二进制值装入一串连续的保持寄存器

ModBus 网络只是一个主机，所有通信都由他发出。网络可支持 247 个之多的远程从属控制器，但实际所支持的从机数要由所用通信设备决定。采用这个系统，各 PC 可以和中心主机交换信息而不影响各 PC 执行本身的控制任务。表 2 是 ModBus 各功能码对应的数据类型。

表 2 ModBus 功能码与数据类型对应表

代码	功能	数据类型
01	读	位
02	读	位
03	读	整型、字符型、状态字、浮点型
04	读	整型、状态字、浮点型
05	写	位
06	写	整型、字符型、状态字、浮点型
08	N/A	重复“回路反馈”信息
15	写	位
16	写	整型、字符型、状态字、浮点型
17	读	字符型

三、通信参数

通信对应参数列表:

K200-MODBUS 通讯协议参数表			
寄存器地址	参数描述		备注
40001	开关机	=1 开机 =0 关机	R/W
40002	设定温度	单位:0.1℃	R/W
40003	设定湿度	单位:1%	R/W
40004	系统状态	详见状态和告警位说明	R
40005	回风温度	单位:0.1℃	R
40006	回风湿度	单位:1%	R
40007	告警位1	详见状态和告警位说明	R
40008	告警位2	详见状态和告警位说明	R
不支持40009以后的地址访问			

状态和告警位说明

40004	40007	40008	位序号
--未开放受保	压缩机 1 高压	加湿器电流过大	Bit15
--未开放受保护	压缩机 1 低压	加湿器缺水	Bit14
--未开放受保护	气流丢失	无加湿电流	Bit13
--未开放受保护	风机过载	--未开放受保护	Bit12
--未开放受保护	加热器过载	--未开放受保护	Bit11
--未开放受保护	空气过滤网	--未开放受保护	Bit10
--未开放受保护	高温告警	溢流告警	Bit9
--未开放受保护	低温告警	用户告警	Bit8
--未开放受保护	高湿告警	烟雾告警	Bit7
--未开放受保护	低湿告警	--未开放受保护	Bit6
除湿	回风温度探头故障	--未开放受保护	Bit5
加湿	送风温度探头故障	--未开放受保护	Bit4
制冷	回风湿度探头故障	--未开放受保护	Bit3
制热	室外温度探头故障	压缩机 2 高压	Bit2
风机	送风温度告警	压缩机 2 低压	Bit1
开关机	--未开放受保护	水流开关告警	Bit0

注:

1) 上表中 40004 状态定义中：1 表示运行，0 表示停止；
   上表中 40007、40008 告警定义中：1 表示有告警，0 表示无告警。
2) 本协议中使用到 Modbus 功能码： **读寄存器(03)和写寄存器(06)**。
   举例如下：
   读(功能码 03)回风湿度：“01 03 00 05 00 01 94 0B”

其中:“01”：通信地址；

“03”：读寄存器；

“00 05”：寄存器地址，（两个字节，需要回风湿度寄存器地址 40006 减去 40001，即 40006 -40001 = 00 05）；

“00 01”：要读取的寄存器个数；

“94 0B”：CRC16 校验(“01 03 00 05 00 01”的 CRC16 校验值为“94 0B”)。

写(功能码 06)设定温度：“ 01 06 00 01 00 FA 58 49”

其中：“01”：通信地址；

“06”：写寄存器；

“00 01”：寄存器地址，（两个字节，需要设定温度寄存器地址 40002 减去 40001，即 40002 -40001 = 00 01）；

“00 FA”：写入寄存器数据（设定的温度数据 25.0℃，0xFA (十六进制) = 250（十进制））；

“58 49”：CRC16 校验(“01 06 00 01 00 FA”的 CRC16 校验值为“58 49”)。