

# 松原西门子PLC模块代理商

产品名称	松原西门子PLC模块代理商
公司名称	浔之漫智控技术-西门子PLC代理商
价格	.00/件
规格参数	
公司地址	上海市松江区石湖荡镇塔汇路755弄29号1幢一层A区213室
联系电话	15221406036

## 产品详情

在使用西门子S7-200时，在不增加其它通讯接口模块时，紫金桥软件可以通讯PPI协议与S7-200直接通讯。但这种通讯方式由于受通讯协议本身限制，在通讯数据较多时，通讯速度相对较慢，使用Modbus协议则通讯速度可以提高约一倍。下面就介绍一下怎么使用Modbus协议建立S7-200与紫金桥软件之间的通讯：

### 硬件连接

在使用Modbus协议时，计算机与S7-200之间通讯直接使用PPI通讯电缆即可。但如果通讯距离较远，或者需要将多个S7-200连接到一个通讯总线上时，我们可以通过如下方法配制通讯链路。

### 选配一个计算机通讯端口

由于S7-200通讯端口物理层使用的是RS-485通讯规范，因此我们需要在计算机端增加一个RS-485通讯端口，才能与计算机通讯建立通讯。如果计算机闲置的串口，我们可以选配一个RS-232转RS-484转换器即可；如果没有闲置的串口，我们通过在计算机中增加一个RS-485通讯卡也可以；现在很多计算机都有USB口，我们也可以在计算机上外接一个USB转RS-485转换器。

### 连接通讯电缆

S7-200的通讯端口是一个9孔（famel）D型插头，针脚分布如下所示：

针脚

信号

地线 (RS-485 逻辑地)

2

24 V 地线 (RS-485 逻辑地)

3

信号B (RxD/TxD+)

4

RTS (TTL level)

5

地线 (RS-485 逻辑地)

6

(空)

7

24 V 电源

8

信号A (RxD/TxD-)

9

通讯选择

S7-200通讯端口（端口0）与RS-485板卡或RS-485/RS-232转换器之间接线，如下图所示：

## S7-200编程及设置

在缺省情况下S7-200的通讯端口是不支持Modbus协议的，要想实现Modbus通讯必需在PLC的主程序模块中调用Modbus通讯子程序。Modbus通讯子程序可以从“STEP 7-Micro/WIN Add-On: Instruction Library (STEP 7-Micro/WIN附件：指令库)”中获得。在安装了“STEP 7-Micro/WIN附件：指令库”后，在导航树“指令/库”下面我们可以找到“Modbus Protocol”。在其下面包含了MBUS\_INIT和MBUS\_SLAVE两个子程序，MBUS\_INIT用于对Modbus通讯进行初始化，MBUS\_SLAVE用于在指定端口上提供Modbus从站通讯服务。下在介绍如何在主程序中调相关子程序及环境参数设置：

### 调用Modbus通讯初始化命令

首先为MBUS\_INIT命令建立一个触发条件（只触发一次），如：SM0.1；从导航树“指令/库/ Modbus Protocol”下面，将MBUS\_INIT指令拖拽到主程序块中。再就是正确设置MBUS\_INIT各项调用参数和执行结

果输出地址，我们可以建立一下如下图所示初始化调用过程：

Mode:协议类型，1-Modbus协议；0-PPI协议。

Addr: PLC地址，1~247，。

Baud: 通讯波特率，1200、2400、4800、9600、19200、38400、57600或11520。

Parity: 校验方式，0-无校验；1-奇数校验；2-偶数校验。

Delay:信息结束超时时间，0~32767,有线连接设置为0即可。

MaxIQ:0~128,映射在离散输入寄存器或离散输出寄存器中的I或Q数。建议为：128。

MaxAI:0~32；映射在模拟输入寄存器中AIW数；CPU 221为0，CPU 222为16，CPU 224、226和226XM为32。

MaxHold:V内存映射在保持寄存器中的寄存器数。

HoldStart:V内存的映射时的起始地址。

Done:初始化指完成时，输出为开状态；开关量（Bit）。

Error:初始化错误代码，请参阅Modbus从机协议执行错误代码；输出为字节。

#### 调用Modbus从机通讯命令

初始化完成后，就可以调用Modbus从机通讯命令（MBUS\_SLAVE）了。通常Modbus从机通讯命令在主程序块的每个执行周期都要初执行（始终保持通讯状态），因为可以用一个常开量作为命令的触发条件，如：SM0.0。从导航树“指令/库/Modbus Protocol”下面，将MBUS\_SLAVE指令拖拽到主程序块中。调用过程如下图所示：

Done: 当MBUS\_SLAVE指令响应对Modbus请求时，“Done”为开状态。没通讯请求时“Done”为关状态。

Error:指令执行结果。只有“Done”为开状态时，此状态有效。请参阅Modbus从机协议执行错误代码；输出为字节。

#### 库存储区分配

在完成前面两个指令调用后，还要为库指令使用的符号分配内存。当库指令被插入到主程序块中，在导航树“程序块”下会出现一个“库”节点。在“库”节点上点击鼠标右键，在弹出菜单中选择“库存储区”，进入“库存储区分配”对话框。

在地址框中输入分配内存的开始地址，或者通过点击“建议地址”按钮自动分配内存。注意，分配的内存不要与已使用的内存重叠。

完成PLC编程和配制后，还要对紫金桥软件进行配制，这样才能获取、处理、查测、存储和显示相应的数据。组态过程分为两个步骤：（1）设备组态，即建立一个与PLC相对应的逻辑设备；（2）点组态，即建立过程数据库点的过程参量与PLC中变量的对应关系，从而利用点的自身功能进行处理、检测和存储。当然要将过程数据在计算机上展现给用户，还要进行交互界面（窗口）组态。但这一过程对所有应用是相同这，这里就不介绍了。

## 设备组态

设备组态就根据PLC使用的通讯协议及在网上的节点地址、端口属性等对通讯参数进行定义的过程。

进入紫金桥开发环境，将导航树切换到“数据库”，在导航树上选择“设备驱动/MODBUS/MODBUS（ASCII&RTU串口通讯）”，单击鼠标右键，在单出菜单中选择“增加设备驱动”会弹出发下所示对话框：

在对话框中输入设备名称、设备地址（与PLC中MBUS\_INIT初始化的Addr参数一致），选择串口号，并且设置（点击“设置”按钮）端口属性，使用它与PLC中MBUS\_INIT命令设置的属性一致。点击“下一步”：

在Modbus设备定义对话框中将协议类型选择为“RTU”，存储器类型选择“32位”（即长整和浮点型4字节的数据的传送格式为高位在前，低位在后）。单击“完成”按钮，则设备组态完成。

## 点组态及I/O连接

完成了点组态还要进行点组态I/O连接，这样才能将PLC中的变量映射到紫金桥软件中。

进入紫金桥开发环境，将导航树切换到“数据库”，在导航树上双击“点组态”，进入点组窗口：

选择点类型，建点，设置常规参数是点组态的基本操作，对所有应用是相同的，这里就不详细说明了。下面重点介绍一下I/O连接及与变量对应关系。

这里以模拟I/O点的I/O连接为例。新建一个模拟I/O点后，输入点名，切换到“数据连接”属性页：

从该属性的设备列表框中选择在上一步骤中建立的设备（如：MOD），点击“增加连接项”按钮，进入“Modbus组点对话框”：

这里的“内存区”、“偏置”和“数据格式”是与PLC中部分内存类型和地址相对应的。对于在Modbus协议中不能对应的PLC内存类型，如果需要上传到过程数据库中，可以先它们先移动到V内存区，再通过V内存区读取。S7-200内存区与紫金桥I/O连接对应关系如下所示：

## S7-200地址

## 紫金桥软件I/O连接

内存区

偏置

Q0.0

DO离散输出量

0

Q0.1

1

Q0.2

2

.....(Qm.n)

.....(m\*8+n)

Q15.6

126

Q15.7

127

I0.0

DI离散输入量

0

I0.1

1

I0.2

2

.....(Im.n)

.....(m\*8+n)

I15.6

126

I15.7

127

AIW0

AR输入寄存器

0

AIW2

1

AIW4

2

.....(AIW2n)

..... ( n )

AIW62

31

Hold Start ( 如 : VB0 )

0

Hold Start+2 ( 如 : VB2 )

1

Hold Start+4 ( 如 : VB4 )

2

.....

.....

Hold Start+(HoldMax-1)\*2

HoldMax-1

在完成存储区和地址组态，还要注意一点就是数据类型一致性，只有类型一致才能从PLC上获得正确数

据。我们可以从“数据格式”中选择格式。对于字寄存器我们也可以将它拆分为位，从而获得某个位的状态。

注：对新组态的数据库点只有重新启动后才会生效。

以上是对通过Modbus协议在紫金桥软件中获取S7-200 PLC数据的使用方法的简单介绍，希望对大家使用紫金桥有所帮助

## 1.前言

可编程控制器（PLC）作为一种高性能的工业现场控制装置，已广泛地用于工业控制的各个领域。目前，工业自动控制对PLC的网络通信能力要求越来越高，PLC与上位机之间、PLC与PLC之间都要能够进行数据共享和控制。

飞剪控制系统要求在远离PLC的控制室里，实时监控电机、供纸、刀辊等设备。上位机为普通PC机，下位机为SIEMENS S7-222 PLC。在实际开发中，采用自由口通信模式，自定义PC与PLC的通信协议，用Step7编写PLC端的通信程序，而在PC端用VC6.0实现串行通信的控制和监控界面的显示。

## 2.通信方式及原理

S7-200系列PLC通信方式有三种：一种是点对点（PPI）方式，用于与西门子公司的PLC编程器或其它产品通信，其通信协议是不公开的。另一种为DP方式，这种方式使得PLC可通过Profibus的DP通信接口接入现场总线网络，从而扩大PLC的使用范围。后一种方式是自由口（FreePort）通信方式，由用户定义通信协议，实现PLC与外设的通信。本系统中采用自由口通信方式。它是S7-200系列PLC一个很有特色的功能。这种方式不需要增加投资，具有较好的灵活性，适合小规模控制系统。自由口通信在物理接口上要求双方都使用RS485接口，波特率高为38400bps。虽然PC机的标准串口为RS232，但西门子公司提供的PC/PPI电缆带有RS232 /RS485电平转换器，因此在不增加任何硬件的情况下，可以很方便地将PLC和PC机互联。

### 2.1自由口模式的初始化

PLC的自由口模式通信编程首先是对串口初始化。对S7-200PLC的初始化是通过特殊存储字节SMB30（端口0）写入通信控制字，来设置通信的波特率、奇偶校验、停止位和数据位数。显然，这些设定必须与上位机设定值相一致。另外还可选择通信模式和主从站模式，各具体存储位内容可参考SIMATIC S7-200系统手册。

### 2.2自由口模式下收发数据

初始化自由端口通信模式后，就可以进行数据的收发。

#### （1）发送数据指令 XMT

格式：XMT Table, Port。可以用XMT指令发送数据，XMT指令激活发送缓冲区（从Table开始的变量存储区）中的数据。数据缓冲区的个数指明了要发送的字节数，Port指明了用于发送的端口，缓冲区多可以有255个字符。在发完缓冲区的后一个字符时，会产生一个中断（对端口0为中断事件9）。本例的XMT缓冲区的格式如表1。其中，状态字节表示PLC是否正确接收了上位机所传数据；上传数据为PLC上传给PC的数据，需将9字节的16进制数编码为18字节的ASIIIC码，所以字节数为18；BCC为上传数据的异或和，同样将16进制数编码为ASIIIC码；结束字符的值为26。

## 表1 发送缓冲区表2 接收缓冲区

RCV Table, Port。用 RCV指令接收多为255个字符的数据，这些字符存储在缓冲区中。在接收到结束字符时，会产生一个中断（对端口 0为中断事件23）。本例的RCV缓冲区的格式如表2。其中，命令类型表示上位机让PLC执行什么操作，如读或写；目标站号是分配给PLC的一个代号；起始地址是PLC要进行读写的存储区的起始地址；读写字节数是PLC接到命令后，对存储区进行读写的字节数，本例中多写入16字节、读出9字节；写入数据是上位机要写入PLC的数据，对于读命令不起作用；BCC是从命令类型开始到写入数据为止的43字节数据的异或和。从目标站号到BCC这几项内容，都是把16进制数编码为ASCII码来表示的。

### 3.自由口通信程序设计

通信程序的设计需遵循一定的规则，如中断通信处理程序要短小精悍、要避免XMT与RCV指令同时在一个端口执行等。整个PLC通信程序包括主程序、通信初始化子程序、校验子程序、读写数据子程序和发送完成、接收完成中断服务程序。

#### 3.1主程序

通信主程序是PLC实现接收、发送功能的主框架。其主要流程为查询接收是否完成，校验，再根据命令类型执行读、写等操作。它的作用是控制程序的主流程，校验、读写等具体工作由相应的子程序完成。流程如图1。

#### 3.2通信初始化程序

通信初始化子程序设置自由口通信的有关参数，对接收信息控制寄存器SMB87写入控制字，定义起始字符、结束字符和接收超时。设好自由口模式的这些参数后，还要连接中断事件和中断服务程序，并打开中断。后，把接收、发送缓冲区写入初值即可。

#### 3.3校验子程序

每次PLC接收完1帧数据，就调用此子程序进行校验。进入子程序后，先清除接收完成标志位，再计算所接收数据的校验和BCC。如果正确，还要检验结束字符是否为‘G’。不是的话，说明数据报文长度不对或传输过程中发生了错误，需要向上位机返回相应的出错信息。流程如图2。

图1 主程序流程图图2 校验子程序流程图

#### 3.4读、写数据子程序

这两个子程序的任务是把PLC存储区中的数据发给上位机或把上位机传来的数据写入PLC存储区。二者的流程相似，只是数据流向不同。进入子程序后，先停止接收，然后完成数据传输，后发送应答报文。不同之处就是应答报文中的状态字节：读操作时是1、写操作时是2。

#### 3.5接收、发送完成中断服务程序

当PLC接收到结束字符后产生中断（事件号9）或数据发送结束后产生中断（事件号23），这两个服务程



序被执行。接收完，先把接收完标志置1，然后再次启动接收。发送完，先清除校验正确标志，再把接收缓冲区中的结束字符和计算出的接收BCC结果清零，后再次启动接收。

由于是半双工通信，因此PLC无论是发送和接收完数据后，都必须将通信口设置成接收状态。否则，PLC就接收不到任何数据了。

#### 4.上位机的通信编程

上位机通过RS232口与PLC进行通信，bbbbbs环境串口通信程序利用VC6编写。VC6编写串口通信程序通常有MSCOMM控件和通信API两种方法。二者各有优缺点。MSCOMM控件封装了微机串口通信的基本功能，使用者只需设置一些基本参数，就可以通过串口收发数据了。这种方法简单，易于编程人员使用，现在已有很多例子供参考。用通信API编写串口程序相对复杂一些。开发者要直接使用bbbbbs提供的一组API函数来完成上述控件封装好的功能。所以使用API编程比使用控件更复杂，但同时也更灵活。通信控件已经封装好的功能是无法改变的，而使用API就能针对通信协议编写效率更高的代码。

在飞剪控制系统的上位机程序中，使用通信API编写了串口读写的模块。接收时，程序要查找起始字符‘g’，以确定1帧数据的开始;再根据下一个状态字节判断通信的正确性;后，把长度为23字节的数据帧接收好，并准备接收下一帧。发送过程不用判断数据内容，执行发送函数即可。需要注意的是：由于PLC通信口是半双工的，所以在PLC向上位机上传数据时，上位机等1帧数据接收完毕，再执行发送操作，以避免收发冲突。

图3 上位机串口通信流程图

#### 5.结束语

本系统取PC机和PLC各自的特点，实现了对飞剪系统的实时监控。通过利用PLC（下位机）的自由口通信协议和上位机的VC开发工具，可以方便地开发出PC机和PLC通信应用软件。这种方法节省投资，对小规模的系统极具现实意义。系统具有实时性好、速度快、可靠性高、操作方便等优点，达到了预期的效果。经现场调试及运行表明，该系统适合于飞剪系统的实时监控。