

西门子模块6ES7253-1AA22-0XA0实体经营

| | |
|------|--------------------------------|
| 产品名称 | 西门子模块6ES7253-1AA22-0XA0实体经营 |
| 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

西门子模块6ES7253-1AA22-0XA0实体经营

随着DCS系统在化工自动化领域的广泛应用，其功能已不仅仅是完成控制和操作，在很大程度上还要整合其它各辅助系统的运行信息和生产数据，并将这些数据提供给操作人员和管理者。因此，DCS系统如何实现与第三方系统间通讯进行数据交换，越来越被更多的用户重视并广泛应用。大庆炼化公司30万吨/年聚丙烯装置DCS控制系统是CS3000系统，有5套PK包机组采用西门子独立的S7400PLC系统，并与CS3000进行MODBUS通讯。在DCS与PLC通讯时，DCS只能作为主设备，PLC作为从设备。MODBUS为单主站网络协议，所以系统中只能有一个MODBUS主站，并且只能实现主站和从站的数据交换，从站之间不能进行数据交换。西门子CP340/CP341/CP440/CP441-1/CP441-2模块是S7300/400系列PLC中的串行通讯模块，该模块具有串行通讯口。可以使用这种通讯模块实现S7300/400与其他串行通讯设备的数据交换，例如打印机、扫描仪、仪表、MODBUS主从站、Data Highway站、变频器，USS站等。下面以CP341与ALR121为例介绍其通讯功能，其网

络框图如图1所示。

一、通讯的硬件配置

1、CS3000系统的ALR121通讯卡 ALR121通讯卡支持RS 422/485 通讯，其接口定义和通讯接线如图2所示。RS485接线采用普通的3线接法。

2、西门子CP341通讯卡 CP341通讯模块是西门子S7-300/400系列PLC中的串行通讯模块，具有1个串行通讯口（RS232C或RS485/422）。可以使用这种通讯模块实现S7300/400与其它串行通讯设备的数据交换其物理接口如图3所示。

图1 CP341与ALR121通讯网络示意图 3、CP341 与 ALR121连接进行RS485通讯时，将CP341的4、8、11端子与ALR121的TX+、TX -、SG端子连接，接线图如图4所示。

图2 ALR121硬件接口 图3 CP341硬件接口 图4 RS485通讯CP341 与 ALR121接线图

二、软件组态

下面以聚丙烯PK301机组S7-414H型PLC与CS3000系统间通讯为例，介绍CP341和ALR121如何进行通讯组态和编程的。

1、PLC 软件编程 首先安装STEP 7 V5.3 + SP2版编程软件和CP341模板所带的软件驱动程序。在硬件配置中定义CP341通讯卡，如图5所示。在标记CP 341 选择"Edit > bbbbbb Properties"，或者双击CP 341，显示图6.0，不必在"General"，"Addresses"和"Basic bbbbbbeters"标签中做任何设置。点击" bbbbbbeters"按钮来参数化波特率9600，偶校验等参数，这里非常重要的一点就是PLC作为MODBUS通讯的从站地址"10"的定义，如图6.1所示。然后再设定MODBUS从站的Function Code地址与PLC中M，I，Q等地址的对应关

系。图6.2是定义MODBUS通讯RS485协议的定义，图6.3是定义调用的DB60和DB61数据块。图5 CP341通讯卡硬件组态 图6.0 CP341属性显示MODBUS通讯连接 图6.1 CP341通讯从站端口定义 图6.2 CP341中定义RS485通讯 图6.3 CP341中定义调用的DB数据块 2、CS3000系统组态 在CS3000组态中，首先要对ALR121的定义。定义为冗余型通讯卡，通讯端口为PORT2。如图7所示。然后定义端口PORT2的通讯参数，波特率9600BPS，偶校验，响应时间等，如图8所示。在定义完通讯卡属性后，重要的一项工作就是在ALR121中定义通讯点的地址。如图9所示。其中关键参数是：SIZE：44 PORT：2 STATION：10 DEVICE&ADDRESS：A30061（数字量）DEVICE&ADDRESS：A30080（模拟量）DATA TYPE：bbbb（32 - Bit Floating）LABEL：%%TT3501（仅以模拟量温度指示TT3501为例）DEVICE & ADDRESS的地址A30080是PLC编程者提供的MODBUS通讯地址的起始地址，以后的排列顺序必须与PLC提供的地址表相一致。"Device & Address"在MODBUS PLC的地址结构如下：<function code> + <device type> + <address of the device> "xxxx"是MODBUS PLC通讯的起始地址，范围是1 - 65535 (1H to FFFFH)，对于PK301机组模拟量通讯的起始地址就是A30080，数字量通讯的起始地址就是A30061。图7 ALR121通讯卡属性定义 图8 ALR121通讯卡通讯端口定义 图9 ALR121中定义通讯点的地址 对于数字量通讯，在定义完硬件地址A30061等参数后，要记住%WW0109地址，然后如图10在FCS0101的SWITCH中定义通讯点变量详细地址。注意，由于1Word=8Byte = 16Bit，因此%WW0109对应%WB10901~%WB109016，可以定义16个数字量。按照PLC程序提供的通讯变量顺序定

义地址，如图11所示。其中Tag Name是在CS3000系统中调用的变量名称，如变量TT3506AS就是温度传感器故障报警信息，在DCS流程图中的调用见图12所示。图10 在SWITCH中定义通讯点变量 图11数字量通讯点地址分配 图12 TT3506AS在DCS流程图中调用 对于模拟量，当对PLC通讯到DCS的数据进行显示时，利用CS3000系统的FUNCTION BLOCK建立PVI显示即可，例如TI3501组态如图13所示。图13 FUNCTION BLOCK的PVI组态 结论 聚丙烯装置运行几年来，PLC与DCS通讯性能稳定，技术可靠，已经在数据统计分析等方面体现出明显的优势，已经成为当今自动化应用领域的发展方向。当然PLC与DCS通讯的手段和策略有很多种，需要广大的用户不断总结和探索，本文所阐述的不当之处还请同行人士批评指正。

初学PLC梯形图编程，应要遵循一定的规则，并养成良好的习惯。下面以三菱FX系列PLC为例，简单介绍一下PLC梯形图编程时需要遵循的规则，希望对大家有所帮助。有一点需要说明的是，本文虽以三菱PLC为例，但这些规则在其它PLC编程时也可同样遵守。 一，梯形阶梯都是始于左母线，终于右母线（通常可以省掉不画，仅画左母线）。每行的左边是接点组合，表示驱动逻辑线圈的条件，而表示结果的逻辑线圈只能接在右边的母线上。接点不能出现在线圈右边。如下图（a）应改为（b）：

二，接点应画在水平线上，不应画在垂直线上，如下图（a）中的接点X005与其它接点间的关系不能识别。对此类桥式电路，应按从左到右，从上到下的单向性原则，单独画出所有的去路。如图（b）所示：

三，并联块串联时，应将接点多的去路放在梯形图左方（左重右轻原则）；串联块并联时，应将接点多的并联去路放在梯形图的上方（上重下轻的原则）。这样做，程序简洁，从而减少指令的扫描时间，这对于一些大型的程序尤为重要。如下图所示：

四，不宜使用双线圈输出。若在同一梯形图中，同一组件的线圈使用两次或两次以上，则称为双线圈输出或线圈的重复利用。双线圈输出一概梯形图初学者容易犯的毛病之一。在双线圈输出时，只有后一次的线圈才有效，而前面的线圈是无效的。这是由PLC的扫描特性所决定的。 PLC的CPU采用循环扫描的工作方式。一般包括五个阶段（如图所示）：内部诊断与处理，与外设进行通讯，输入采样，用户程序执行和输出刷新。当方式开关处于STOP时，只执行前两个阶段：内部诊断与处理，

与外设进行通讯。

1, 输入采样阶段 PLC顺序读取每个输入端的状态,并将其存入到我们称之为输入映像寄存器的内在单元中。当进入程序执行阶段,如输入端状态发生改变,输入映像区相应的单元信息并不会跟着改变,只有在下一个扫描周期的输入采样阶段,输入映像区相应的单元信息才会改变。因此,PLC会忽视掉小于扫描周期的输入端的开关量的脉冲变化。 2, 程序执行阶段 PLC从程序0步开始,按先上后下,先左后右的顺序扫描用户程序并进行逻辑运算。PLC按输入映像区的内容进行逻辑运算,并把运算结果写入到输出映像区,而不是直接输出到端子。 3, 输出刷新阶段 PLC根据输出映像区的内容改变输出端子的状态。这才是PLC的实际输出。 以上简单说明了PLC的工作原理,下面我们再以实例说明为什么编写梯形图程序,不宜重复使用线圈。如下图所示,设输入采样时,输入映像区中X001=ON, X002=OFF, Y003=ON, Y004=ON被实际写入到输出映像区。但继续往下执行时,因X002=OFF,使Y003=OFF,这个后入为的结果又被写入输出映像区,改变原Y003的状态。所以在输出刷新阶段,实际外部输出Y003=OFF, Y004=ON。许多新手就碰到过这样的问题,为什么X001已经闭合了,而Y003没有输出呢?逻辑关系不对。其实就是因为双线圈使用造成的。

[bbb=][/bbb]

注意:我们所说的是不宜(好不要)使用双线圈,双线圈使用并不是禁止的,在一些特殊的场合也可以使用双线圈,这时就需要你有较丰富的编程经验和技巧了。下面我们会谈到这一点。但对于初学者还是不要冒这个险。其实,从以上的例子可以看出,重复利用线圈之所以会造成Y003的输出混乱,是由于程序是从上到下顺序执行的缘故造成的。但如果我们可以改变程序执行的顺序,保证在任何时刻两个线圈只有一个驱动逻辑发生,就可以使用双线圈。其中,常用的方法就是使用跳转指令。如下图所示:[bbb=][/bbb]

程序分析:M0闭合,程序跳至P0处(不执行X001语句),M0常闭断开,CJ P1不会发生,执行下一语句。此时,Y003将X002状态进行驱动。M0断开时,程序顺序执行并按X001的状态对T003进行驱动,M0常闭闭合,跳至P1按X003状态对Y004进行驱动,即跳过了X002驱动Y003的语句。可见,在同一时刻,Y003驱动只有一个可以发生。此时,双线圈利用是可以的。但在梯形图编程时,我们还是尽量避免使用双线圈,而引入辅助继电器是一个常用的方法。如下图所示:图(b)中,X001和X002接点控制辅助继电器M000,X003~X005接点控制辅助继电器M001,再由两个继电器M000,M001接点的并联组合去控制线圈Y000。这样逻辑关系没变,却把双线圈变成单线圈。