

西门子模块6ES7253-1AA22-0XA0支持验货

产品名称	西门子模块6ES7253-1AA22-0XA0支持验货
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支持验货

可编程控制器PLC的应用绝大部分都是通过编写梯形图的方式实现程序控制的，但对于智能低压开关系统来讲，其分、断的控制往往要求由主控操作室远程控制来完成。按传统的方法，主控室要对某一个开关柜进行操作，须先发一个控制命令，PLC接收到该控制命令后，再由PLC程序执行该命令，该段程序必须用梯形图或逻辑指令编程，当开关柜数目巨大或要求多路设备同时控制时，PLC的程序编写就比较复杂，尤其是采用脉冲型输出控制而不是用电平保持型控制时更是如此。因为开关柜的分、合控制须两路继电器输出分别控制“分闸”和“合闸”，而不是由单路继电器保持输出控制，使继电器输出与开关柜分合一致。考虑这些因素，本文采用串行通讯实现PLC的控制，PLC本机无须任何梯形图或逻辑指令，这样不仅实现简单而且方便可靠。

一、低压监控系统设计（一）系统设计要求依据用户的实际要求，智能低压开关柜不仅能在当地手动进行分、合，而且具有系统零位、时控和远方控制的功能，其控制回路的电气控制图如图1所示。图1中SV为转换开关，当1、2通时，为当地时控方式；9、10通时为手动方式；5、6通时为远程控制。当开关柜工作在远方控制方式时，当PLC输出触点#107-#113闭合时，KM线圈加载得电，KM常开触点闭合，并形成回路使KM线圈始终得电，此时PLC输出触点#107-#113释放，当PLC输出触点#105-115吸合时，KA线圈得电，KA常闭继电器断开，使KM线圈回路失电，KM常开触点断开，当KM线圈常开触点断开后，PLC#1052115输出触点释放，开关维持断开状态。图中#901-#903作为PLC的输入触点。（二）设计方案简述本系统针对低压系统的实际要求设计，实现开关柜的远程控制。主站通过电力专用modem与远方子站RTU进行双向通讯，主站发出控制命令，子站接收主站控制命令并转发到PLC，并将PLC的执行结果及时反馈给主站，控制时间在1~2s完成。监控系统分为三级：主站、变电站当地子站、PLC和仪表设备。主站借助GPS时钟jingque定时，操作员可将任何时候的停电和启动设置进主站，由主站自动发出控制命令，该监控系统极适合路灯及景观灯的控制。图1 控制回路的电气控制原理图二、通讯协议及处理方法（一）主站与子站的通讯协议主站采用电力系统远动循环规约与子站进行通信，但远动循环规约中仅有主站对子站的单点遥控命令，为此主站端必须对CDT规约进行扩展，我们借用遥信报文的格式，将主站端需要控制的低压开关状态，用遥信报文下发到子站，为此仅需将下发的遥信报文中的开关状态内容分析出来，分别放置在yk_road[0]、yk_road[1]、yk_road[2]、yk_road[3]四个字节中，支持对32路低压断路器的控制。报文由同步字、控制字和信息字三部分组成，其格式如下：同步字为6个字节：0dbh, 09h, 0dbh, 09h, 0dbh, 09h；控制字为6个字节：控制字节、帧类别、帧长度、源站址、目的站址、CRC校验；信息字为帧长度×6个字节，本文因为控制点数在32个以内，所以信息字

仅为6个字节：功能码、1到8组开关柜控制状态、9到16组开关柜控制状态、17到24组开关柜控制状态、备用开关柜控制状态、CRC校验。整个控制报文18个字节，这样既符合CDT规约的通信格式，又不增加RTU的通信报文分析负担。在该规约处理中，关键为实时判断出同步字头，然后根据帧类别区别遥控选择命令、遥控执行命令、遥控撤消命令、系统对时和成组开关控制命令，分析出遥控数据存放在yk_road[0]~yk_road[3]中，并置相应标志，再由远动测量装置将命令进行分析转换发给PLC。实现该规约的分析方法voidinterruptCntReach1()函数如下：voidinterruptCntReach1()/*串口1每收到一个字节中断一次的中断处理程序*/{cntreach[1]++;/*每接收一个字节，中断接收计数器加1*/stat=sio_read(p[1],data,1);/*从串口1缓冲区读入一个字节数据放到data[0]中*/if(ok[1]==0){/*如未判断到同步字节，以下分析同步字头*/switch(tp[1]){case0:if(data[0]==0xd7)tp[1]++;break;case1:if(data[0]==0x09)tp[1]++;else tp[1]=0;break;case2:if(data[0]==0xd7)tp[1]++;else tp[1]=0;break;case3:if(data[0]==0x09)tp[1]++;else tp[1]=0;break;case4:if(data[0]==0xd7)tp[1]++;else tp[1]=0;break;case5:if(data[0]==0x09){ok[1]=1;cntreach[1]=0;}else tp[1]=0;break;default:break;}if((ok[1]==1)&&(cntreach[1]!=0))/*已判断到同步字头后，以下分析报文内容*/{Rx_buf[1][cntreach[1]-1]=data[0];if(cntreach[1]==3)LGn[1]=Rx_buf[1][2];if(cntreach[1]>=(LGn[1]*6+6)){step_flag[1]=1;/*置报文同步标志*/CODE[1]=Rx_buf[1][1];/*帧类别*/switch(CODE[1]){case0x61:.../*遥控选择命令处理*/case0xc2:.../*遥控执行命令处理*/case0xb3:.../*遥控撤消命令处理*/case0x7a:.../*系统对时,*/case0xf4:if(Rx_buf[1][6]==0xf0)/*分析出遥控数据，并置相应标志*/if(Rx_buf[1][2]==1){yk_road[0]=Rx_buf[1][7];yk_road[1]=Rx_buf[1][8];yk_road[2]=Rx_buf[1][9];yk_road[3]=Rx_buf[1][10];yk.road=1;yx1_state[0]=1;}break;default:break;}reset_point(1);/*复位各种标志的函数*/}sio_flush(p[1],0);} (二) 子站与松下PLC的通信协议松下PLC通信协议MEWTOCOL2COM为厂家专门设计的协议，采用ASCII码的方式传送，用应答方式实现PC机与松下Fp1PLC之间的通信。由子站端RTU发出/命令信息0，FP1PLC响应，其基本格式如图2。其命令代码Commandcode由2~3个ASCII字符组成，共26种，主要有：RCS为读一个触点数据；WCS为写一个触点数据；RCP为读多个触点数据；WCP为写多个触点数据；RS为读定时器和计数器的设定值；WS为写定时器和计数器的设定值；RCC为将多个触点数据以字方式读入；WCC为将多个触点数据以字方式写入；因此，通过对这26种命令组合分析，可以完全构成由远程对PLC进行控制，而PLC本身无需编写任何梯形图。通过分析测试，例如对PLC下发ASCII数据串“%01#RCCX00000001*\xod”，共19个字节，其中X是输入数据类型，Y是输出数据类型，数据串末尾是回车符，必须用\xod代替，其中报文可以不计算BCH校验码，用**代替，该报文实现了以字方式读01号PLC中的输入从0000到0001数据，共32路输入数据。PLC收到该报文后，立即回17个字节的数据串，前6个数据为/%01#RC0，后8个数据为2个字的输入点数据，每个字数据用4个字节，每个字符可表示4位输入点的状态，设计了Read_fp1_bbbbb()函数实现报文分析转换方法。对PLC的输出控制用WCC命令实现，其基本数据串格式为01#WCCY000000030000000000000000**/\xod”，该报文实现了以字方式置01号PLC中的输出点从0000到0003数据，控制数据内容为16个字节，每个字符可表示4位输出点的状态，共64路输出数据。因为每个低压开关柜需要“开”、“断”2路输出控制，实际此命令只可控制32路开关。为了系统可靠，低压开关设计为脉冲控制型，为此必须用通信命令来实现PLC输出触点的同时输出，延时一段后再对PLC进行求反输出控制。设计Write_fp1_output()函数实现了具体报文转换成输出数据下发给PLC的方法。

(三) 控制的实现 我们为系统操作员在主站端设计了两种控制方式，第1种为手动遥控：在后台的图形界面中，点击进入相应的变电站，再进入路灯和景观灯所在的控制界面，界面中每一组路灯对应变电站的一个低压抽屉柜，操作员用鼠标选择路灯和景观灯的“开”、“灭”状态，再将所有的开关状态组成字节数据，用CDT报文下发给变电站的RTU。RTU收到路灯控制命令后，将报文解析和重组，用MEWTOCOL2COM通信协议将控制数据传送给PLC。PLC将按照系统操作员在后台的设置执行控制命令，RTU将控制结果及时反馈给主站端，主站端将这一控制过程作为一条记录存入数据库，可以事后查询。第2种控制方式为时控：操作员用鼠标选择在××年××月××日××时××分或星期几时的路灯和景观灯的/开0、/灭0状态，将这些数据存入数据库，当设定时间到达时系统自动发出这些数据，系统可以根据RTU的反馈信息确定控制是否成功，并可决定是否重发。因为对PLC的控制是在子站RTU完成的，为控制的实时性，我们采用C语言编程实现了对PLC的通信控制，确保PLC中无需编写任何梯形图逻辑。由于系统主站与子站之间的通信是基于十六进制数据的CDT规约，而MEWTOCOL2COM通信协议是基于ASCII码的数据通信，为此设计了两个函数用于基于ASCII字符数据的发送和接收。其

中发送数据函数 Sio_putb () , 其使用格式见 Read_fp1_bbbbb () 程序。另一个数据接收中断处理函数为本设计的关键, 为方便和可靠, 本文设计的中断触发信号为“ 串口收到回车符 (十六进制的 0dH) ”, 确保 PLC 每应答完一帧完整的报文, RTU 才进入中断处理, 并在中断处理程序中完成报文数据的分析, 大大提高了快速性和可靠性。这种方法在基于 ASCII 字符的通信中极为有效。本文 RTU 有 6 个串口, 串口 1 为与主站的全双工通信通道, 串口 6 为与 PLC 通信通道, 用 sio_term_irq (p[6], TermCome, 0x0d) ; 函数指明串口 6 只有收到回车符 (0x0d) 才响应中断 void interrupt Term2Come () , 其他的为变电站内的微机保护监控装置、发电机、直流电源屏、变压器通信通道, 本文不做赘述。其中断函数初始化及中断处理程序参见文献。

1 引言 普通的计算机一般都配有并行接口(就是通常用的打印机接口), 它是通过 25 芯的 D 型接头连接打印机与计算机的, 计算机的并行接口不仅可以向打印机发送数据, 还可以从打印机接收数据(打印的工作状态)。因此, 利用计算机的并行接口, 你可以去控制外部继电器的吸合, 从而控制外部设备, 同样可以利用它来检测外部设备的状态。不需添加任何硬件。从而使一台普通的计算机具有“ PLC ”的开关量的功能。本文分析的系统, 把普通并口实现了一个 4 输入 8 输出数字量的功能, 切不需任何投资。们先对打印机的并行口做一下细致的分析。配有一个并口的计算机经系统引导后, 初始化过程把并行端口配置成 LPT1, 这个端口又分配了数据地址 378H, 状态地址 379H。其中 378H 端口做输出端口, 379H 端口做输入端口。

2 并口数据采集控制原理

(1) 端口 378H, 378H 的管脚见表 1。在计算机通电后的自然状态, 378H 端口的每个数据位为 1, 表 1 中每个管脚对地(25脚)为高电平 5V。即 378H 地址的内容为 FF(十进制是 255), 当控制 378H 的内容为 1(十进制)二进制码是 00000001 时, 即 2 脚对地输出高电平, 其它对地是 0 电平。如控制 378H 的内容为 2(十进制)二进制码是 00000010, 即 3 脚对地输出高电平, 其它对地是 0 电平。如控制 378H 的内容为 3(十进制)二进制码是 00000011, 即 2、3 脚对地输出高电平, 其它对地是 0 电平。依次类推。

当某管脚对地是高电平 5V 时, 由于每条数据线的输出电流高达 20mA, 可以接 5V 的直流继电器, 通过继电器的触点去控制其它的设备, 如电磁阀、电机等。图 1 是用打印机并行接口实现的继电器控制电路, 此电路稍加修改即可用于其它设备的开关控制。(2) 端口 379H, 379H 的管脚见表 2。

在计算机通电后的自然状态, 379H 端口的数据位第 7 位是 0, 其余每个数据位为 1, 即上述每个管脚对地(25脚)为高电平 5V。即 379H 地址的内容为 127(十进制)二进制码是 01111111。当只把某管脚(如 10 脚)通过 500 的电阻接到 25 脚, 则它所对应的数据位是 0, 其它的数据位不变, 则 379H 的内容为 63(十进制)二进制码是 00111111。当只把某管脚(如 12 脚)通过 500 的电阻接到地 25 脚, 则它所对应的数据位是 0, 其它的数据位不变, 则 379H 的内容为 95(十进制)二进制码是 01011111 依次类推。图 2 是用打印机并行接口实现的开关检测控制电路, 当 K 断开时, 379H 为 127(十进制), 当 K 闭合时, 379H 为 111(十进制)二进制数是 01101111, 即 13 脚所连接的第 4 数据位为 0。

3 软件实现 VB 是一种可视化的编程语言, 自面世以来, 因简单易用倍受广大编程人员的青睐, 作为一种非常方便的 bbbbbb95 应用程序开发平台, 可以实现 bbbbbb 的绝大部分功能。VB 中没有专门的对并口的操作控件, 要实现对接口的读写, 需要对 I/O 进行操作, 但对硬件进行访问以及利用各种控制接口板进行数据采集, 仅靠 VB 是无法解决的。解决这样的问题可以间接使用动态链接库 DLL(Dynamic bbbb Library)来实现。程序人员利用 DLL 可以实现应用程序共享代码和资源, DLL 与用户的程序完全分开。动态链接库(DLL)虽然是一个可执行模块, 但它并不可以直接运行, 它只是提供一群函数供 bbbbbb 应

用程序或其它动态链接库调用，以完成其特殊任务。由于动态链接库是在应用程序运行期间被连接起来的，故称为动态链接库。本设计用VC++设计了一个读写IO端口的动态链接库IO.DLL。笔者利用VC++5.0开发的DLL的完整的程序清单如下： DLL源文件IO.CPP#include

4 结束语 并口25管脚中，作输出用的是2~9共8个管脚，作输入用的10、12、13、15共4个管脚，25脚地，利用并口可方便的实现了4入8出的开关量控制。在上位机中采用VB等语言编程，利用计算机的并口，巧妙地实现PLC的开关量功能，而且可利用VB等语言的界面设计的功能，使系统做到监控一体。且系统不需额外的投资，使用灵活简单，可靠性高。