

# 代理深圳西门子PLC模块 一级代理商

产品名称	代理深圳西门子PLC模块 一级代理商
公司名称	浔之漫智控技术（上海）有限公司西门子一级代理商
价格	99.00/件
规格参数	西门子PLC代理商:西门子触摸屏代理商 西门子授权一级代理商:西门子CPU代理商 西门子模块:西门子PLC模块代理
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	15618722057 15618722057

## 产品详情

为兼顾RS-485/RS-422 接口中半双工和全双工的要求本转换器采用MAX3491 作为RS-485/RS-422 接口电路其主要指标为+3V 至+3.6V 单电源工作工作电流1mA,驱动60负载时半双工时两个120 终端匹配电阻的并联值峰值电流可达 $I_3=3V/60=50mA$ 半双工和全双工工作方式是通过跳线器来设置的见图34.4. 微处理器在本转换器中微处理器所要完成的任务很简单仅需要几根I/O 线即可实现参数的设置和发送使能的自动控制实际选择中采用Microchip公司的PIC12C508A其主要指标为工作电流 $I_4<1.0mA$ (工作电压3V 频率4MHz),6 条I/O 线512kByte 的ROM 其中GP0 GP1 GP4和GP5四个引脚设定对应于16 种常用波特率300 600 1200至38.4Kbps等8 种以及900 1800 至115.4Kbps等8种的延时时间GP3对应于10 位或11位串行数据格式GP2为TXD输入用来检测UART何时发送和停止数据GP1为复用输出引脚用来控制MAX3491的发送使能控制端GP0也为复

本转换器的\*大电流总和 $<I_1+I_2+I_3+I_4=0.12+2.0+ 50.0+1.0=53.12mA$  小于DC-DC转换器的\*小输出电流5 4.4mA 因而通过RS-232信号线为本电路供电是可行的实际上由于输入电源端的储能电容E1 和TXD 为负电平时能够为电路补充一定的功率所以设计上留有较大的电源功率裕量5. 软件设计本转换器的软件设计较为简单微处理器复位后将所有的I/O 口设为输入并读入所有的I/O 状态保存到寄存器将GP2 和GP3 改设为输出状态并输出低电平使RS-485/RS-422 接口处于禁止发送允许接收的状态CPU 根据GPIO 的初始状态确定出用户设定的通讯波特率和串行数据格式从而预置内部的延时设定CPU 检测到UART开始通讯后打开发送使能经内部预置延时后开始在一个位宽时间内检测是否有下一个起始位到来如检测到则重新延时等待否则关闭发送使能结束当前通讯重新检测UART的起始位对于半双工通讯方式允许发送使能前应该关闭接收使能而在发送使能关闭后才打开接收使能对于全双工通讯方式其接收使能可以不受此信号控制而可以直接通过跳线接地始终允许接收6. 结论在本RS-232到RS-485/RS-422接口的智能转换器设计中除了本身这个产品具有较高的应用价值外文中所涉及的RS-232信号线供电方案由于其高效率大电流输出能力在许多基于RS-232 接口的应用中都能够很好地满足应用另外这种智能控制RS-485/RS-422 接口的收发使能的思想在扩展基于RS-485/RS-422 接口的网络分支及延伸通讯距离都能够得到很好的应用

要单片机完成一项基本任务，必须将任务分解成一些具体步骤，再要求它去逐项执行每个步骤，还要对它下命令。该命令在单片机术语中称为“指令”(Instruction)。完成一项任务所需的所有指令的有序集合就称为“程序”(Programm)。这些指令要预先一条一条顺序地放到单片机的程序存储器中，单片机在运行时，片中的CPU从程序存储器中逐条有序取出指令，执行指令，并将有关指令执行完毕，即可完成既定任务。不同种类的单片机有不同的一套命令(即所谓“指令系统”)。PIC系列的单片机其指令系统与51系列的不同。PIC16F84有30余条指令构成的指令系统。每条指令由14位(bit)构成，这些位是二进制码的0和1，如果要使16F84端口B的B0位输出高电平，以点亮一只发光二极管LED，而B口的其余各位仍保持低电平，则需要使单片机执行下列各条指令(机器码)：11000000000000 00000001100110 11000000000001 00000010000110 10100000000100 早先的技术人员就是用这样的二进制码来编写程序的。上列程序，看起来像天书，很费解，但它能指挥单片机的运作。因为单片机实际上是一种复杂的数字逻辑电路。我们都知道，要数字电路运作，必须相应输入高、低电平，对正逻辑而言，高电平为1，低电平为0。上述指令顺序在不同的数位上出现的0和1，经译码后，即可完成各种不同的运作，逐步完成单片机所要执行的任务，如点亮一个LED。上述各条指令的写法，虽然是面向单片机，是用来直接指示单片机该如何运作的。因此，这种由0、1组成的指令称为机器语言。实际上，这种由二进制码构成的指令集不但难读懂，而且用来编程也有困难。因为程序往往不是从头到尾顺序执行，有时还需中途转移到其它单元执行一段程序后再返回来。而指令是一条一条顺序存放在存储器各个单元内的。因此，如果要转移，需指明具体转到哪个单元，即要写出该单元的地址。但在编写程序时，该程序有多长，具体要放到哪些单元中，都是未知数，又怎能具体指明要转到哪个单元呢？由于用机器语言会使程序难写、难读，后来一种新型的语言形式——汇编语言就问世了。使用这种语言写程序较方便，也比较容易读懂。不过，和机器语言一样，不同类型的单片机有不同的汇编语言。就如不同地区的人有不同的方言一样。在汇编语言中，转移地址是用符号来表示的。现在，我们把上面由机器语言写成的程序改写成由汇编语言构成的程序：

```
movlw B ' 00000000 ' tris PORT B movlw B ' 00000001 ' movwf PORT B fin : goto fin
```

[NextPage] 上列各条指令实际上是英语缩写和一些数组成的。如第一条中的movlw就是move Literal to w的缩写，其意义为照原样移入工作寄存器W，而“原样”就是后接引号内的数字‘00000000’。引号前的B表示后续的是二进制数。第二句是将W内的数复制到B口的三态(tri - state)控制寄存器中，以设定B口为输出，然后将00000001送入W中，再复制到B口；\*后执行一条无限循环语句以保持B口的状态不变。从单片机外部看去，16F84的第6脚(即B0)维持为高电平，以点亮LED。由上例可知，汇编语言较之机器语言要好懂得多。同时，\*后一句自身循环也是一种转移语句，转移目的地就是此句所在单元地址，如用机器语言，就难以标出具体地址，而用汇编语言助记符fin即可替代