

通化西门子PLC模块代理商

产品名称	通化西门子PLC模块代理商
公司名称	浔之漫智控技术（上海）有限公司-西门子PLC
价格	.00/件
规格参数	品牌:西门子 用途:工业 产地:德国
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	18717946324 18717946324

产品详情

通化西门子PLC模块代理商

我公司主营西门子各系列PLC（S7-200 SMART S7-300 S7-400）触摸屏 变频器（MM系列 G120 G120C G110）伺服（V80 V60）数控备件（PCU50 NCU CCU 轴卡）等价格优势产品为西门子原装正版产品 我公司售出的产品按西门子标准质保 产品本身有质量问题 质保一年 公司秉承：以信待人 以诚待人 质量如生命 客户至上的经营理念 竭诚为您服务 您的肯定是我们大的动力 我们将期待与您长期持久的合作

存储器间接寻址的地址给定格式是：地址标识符+指针。指针所指示存储单元中所包含的数值，就是地址的确切数值单元。存储器间接寻址具有两个指针格式：单字和双字。单字指针是一个16bit的结构，从0-15bit，指示一个从0-65535的数值，这个数值就是被寻址的存储区域的编号。双字指针是一个32bit的结构，从0-2bit，共三位，按照8进制指示被寻址的位编号，也就是0-7；而从3-18bit，共16位，指示一个从0-65535的数值，这个数值就是被寻址的字节编号。

指针可以存放在M、DI、DB和L区域中，也就是说，可以用这些区域的内容来做指针。

单字指针和双字指针在使用上有很大区别。下面举例说明：

L DW#16#35 //将32位16进制数35存入ACC1

T MD2 //这个值再存入MD2，这是个32位的位存储区域

L +10 //将16位整数10存入ACC1，32位16进制数35自动移动到ACC2

T MW100 //这个值再存入MW100，这是个16位的位存储区域

OPN DBW[MW100] //打开DBW10。这里的[MW100]就是个单字指针，存放指针的区域是M区，MW100中的值10，就是指针间接的地址，它是个16位的值！-----

LL#+10 //以32位形式，把10放入ACC1，此时，ACC2中的内容为：16位整数10

T MD104 //这个值再存入MD104，这是个32位的位存储区域 A I[MD104] //对I1.2进行与逻辑操作！

=DIX[MD2] //赋值背景数据位DIX6.5！-----

A DB[MW100].DBX[MD2] //读入DB10.DBX6.5数据位状态 =Q[MD2] //赋值给Q6.5 -----

A DB[MW100].DBX[MD2] //读入DB10.DBX6.5数据位状态 =Q[MW100] //错误！！没有Q10这个元件
从上面系列举例我们至少看出来一点：单字指针只应用在地地址标识符是非位的情况下。的确，单字指针前面描述过，它确定的数值是0-65535，而对于byte.bit这种具体位结构来说，只能用双字指针。这是它们的区别，单字指针的另外一个限制就是，它只能对T、C、DB、FC和FB进行寻址，通俗地说，单字指针只可以用来指代这些存储区域的编号。相对于单字指针，双字指针就没有这样的限制，它不仅可以对地址进行寻址，还可以对BYTE、WORD、DWORD寻址，并且没有区域的限制。不过，有得必有失，在对非位的区域进行寻址时，必须确保其0-2bit为全0！

总结一下：单字指针的存储器间接寻址只能用在地址标识符是非位的场合；双字指针由于有位格式存在，所以对地址标识符没有限制。也正是由于双字指针是一个具有位的指针，因此，当对字节、字或者双字存储区地址进行寻址时，必须确保双字指针的内容是8或者8的倍数。

现在，我们来分析一下上述例子中的A I[MD104] 为什么后是对1.2进行与逻辑操作。

通过L L#+10，我们知道存放在MD104中的值应该是：

MD104：0000 0000 0000 0000 0000 0000 0000 1010

当作为双字指针时，就应该按照3-18bitbyte，0-2bitbit来确定终指令要操作的地址，因此：

0000 0000 0000 0000 0000 0000 0000 1010 = 1.2

详解西门子间接寻址<2> 【地址寄存器间接寻址】在先前所说的存储器间接寻址中，间接指针用M、DB、DI和L直接，就是说，指针指向的存储区内容就是指令要执行的确切地址数值单元。但在寄存器间接寻址中，指令要执行的确切地址数值单元，并非寄存器指向的存储区内容，也就是说，寄存器本身也是间接的指向真正的地址数值单元。从寄存器到得出真正的地址数值单元，西门子提供了两种途径：

1、区域内寄存器间接寻址 2、区域间寄存器间接寻址 地址寄存器间接寻址的一般格式是：

【地址标识符】【寄存器,P#byte.bit】，比如：DIX[AR1,P#1.5] 或 M[AR1,P#0.0]。【寄存器,P#byte.bit】统称为：寄存器寻址指针，而【地址标识符】在上帖中谈过，它包含【存储区符】+【存储区尺寸符】。但在这里，情况有所变化。比较一下刚才的例子：DIX [AR1,P#1.5] X [AR1,P#1.5] DIX可以认为是我们通常定义的地址标识符，DI是背景数据块存储区域，X是这个存储区域的尺寸符，指的是背景数据块中的位。但下面一个示例中的M呢？X只是了存储区域的尺寸符，那么存储区域符在哪里呢？毫无疑问，在AR1中！DIX [AR1,P#1.5] 这个例子，要寻址的地址区域事先已经确定，AR1可以改变的只是这个区域内的确切地址数值单元，所以我们称之为：区域内寄存器间接寻址方式，相应的，这里的[AR1,P#1.5] 就叫做区域内寻址指针。X [AR1,P#1.5] 这个例子，要寻址的地址区域和确切的地址数值单元，都未事先确定，只是确定了存储大小，这就是意味着我们可以在不同的区域间的不同地址数值单元以给定的区域大小进行寻址，所以称之为：区域间寄存器间接寻址方式，相应的，这里的[AR1,P#1.5] 就叫做区域间寻址指针。

既然有着区域内和区域间寻址之分，那么，同样的AR1中，就存有不同的内容，它们代表着不同的含义。【AR的格式】地址寄存器是专门用于寻址的一个特殊指针区域，西门子的地址寄存器共有两个：AR1和AR2，每个32位。当使用在区域内寄存器间接寻址中时，我们知道这时的AR中的内容只是指明数值单元，因此，区域内寄存器间接寻址时，寄存器中的内容等同于上帖中提及的存储器间接寻址中的双字指针，也就是：其0-2bit，bit位，3-18bitbyte字节。其第31bit固定为0。