

SIEMENS西门子湖南省岳阳市（授权）伺服电机一级代理商——西门子华中总代理

产品名称	SIEMENS西门子湖南省岳阳市（授权）伺服电机一级代理商——西门子华中总代理
公司名称	广东湘恒智能科技有限公司
价格	.00/件
规格参数	
公司地址	惠州大亚湾澳头石化大道中480号太东天地花园2栋二单元9层01号房
联系电话	15915421161 15903418770

产品详情

1.1如何获得指针或者间接寻址有关的信息?

指针的类型包括16位指针、32位指针、Pointer（6Byte）和Any（10Byte）。16位指针用于定时器、计数器、程序块的寻址；32位指针用于I/Q/M/L/数据块等存储器中位、字节、字以及双字的寻址，其中第0~2位表示位地址（0~7）、第3~18位为字节地址，其余位未定义；Pointer和Any一般应用在复杂数据类型（比如Date_and_Time/Array/String等）在FB、FC之间的传递。而Any可以看做是对Pointer的延伸，因为由10Byte组成的Any中Byte4~Byte9就是一个Pointer。

1.2为什么语句 LAR1

P##PointerInput 在一个函数(FC)中是无效的,然而，同样的语句在一个功能块(FB)中是有效的?

在FC被调用时，复杂数据类型例如指针是被复制到调用者的临时变量区中，在FC内部对此L区地址直接取址放入到地址寄存器AR1或AR2是不被编译器规则接受的（导致MC7寄存器信息过长），也就是说在FC内部通过P#进行地址寄存器取址仅能支持Temp临时变量。因此如果需要在FC中操作指针等复杂输入输出变量地址需要使用累加器进行中转。考虑到程序的一致性、遵守编译器规则和STL手册中LAR1指令说明，建议用户使用如下指令操作：L P##PointerInputLAR1

1.3 STEP 7 中哪些操作会覆盖DB/DI寄存器或者地址寄存器AR1/AR2的内容?

下面说明了可能引起DB/DI寄存器或者地址寄存器AR1/AR2内容改变的一些操作：

- DB寄存器和AR1受到影响的操作1. 使用完整的DB路径（如L DB20.Val）或者调用FC/FB时使用DB块完整地址作为其参数，则DB寄存器内容被覆盖。例如在OB1中调用FC1后，DB寄存器变成20。OPN DB1Call FC1Input(bit)：DB20.DBX0.2因此在编程的时候，OPN 指令打开数据块，通过DBX x.y的方式访问其中内容，但是如果在打开数据块后DB寄存器的内容被修改了，则DBX x.y的方式访问变

量则会访问到错误的地址。可以通过使用符号寻址的方式或者使用完整路径编程避免，当然重新使用OPN指令也是可以的。2. 调用FC时使用string, array, structure, UDT作为其形参或者调用FB时使用string, array, structure 或者UDT作为其in out形参，在FC/FB程序中访问这些地址则AR1寄存器内容被覆盖，因此当使用AR1进行间接寻址时需要注意AR1内容的正确性。

· AR2地址寄存器和DI寄存器在FB中作为参数和静态变量的基址寻址使用。AR2和DI如果被修改，会影响FB的参数访问，如果希望在FB中使用DI寄存器或者地址寄存器AR2，必须预先保存它们中的内容，并在使用后恢复它们，例如：TAR2 #AR2_SAVE; //AR2寄存器状态保存到#AR2_SAVE
DINO;T #DB2_SAVE;
VE; //DI寄存器状态保存到#DB2_SAVE

User Program

```
LAR2 #AR2_SAVE;  
//AR2寄存器恢复到使用前状态OPN DI[#DB2_SAVE];  
//DI寄存器恢复到使用前状态
```

1.4 如何得到多重背景FB中的变量在背景DB里的juedui偏移量呢？

可以用下面的方法处理：TAR2 (得到多重背景FB在背景DB里的偏移地址)AD DW#16#00FFFFFF (屏蔽掉存储区ID，可参考32位指针格式)L P##Variable (得到变量在多重背景FB里的地址)+D (多重背景FB的偏移地址与变量在多重背景FB里地址相加，即得到实际juedui偏移量)LAR1上述语句就是就得到了变量在背景DB中的juedui偏移量，从而供后续程序处理。

1.5如何在程序中使用ANY型指针？

简要说明如下：L P##Input //指向存储地址指针Input首地址//这个参数是一个Any类型，P##Input指向参数Input的值所在地址，这就是指针的指针LAR1
//装载到地址寄存器AR1中。L W[AR1,P#4.0] //打开DB块//由Any类型结构知道Any类型的Byte4、Byte5存放的数据块号T #BLOCK_NOOPN DB[#BLOCK_NO]
//如果是DB块，打开指定的DB块。L W [AR1,P#2.0]
//判断ANY指针中数据长度//Any类型的Byte2、Byte3是重复系数，如P#DB1.DBX0.0 Byte 8后面的Byte 8_001:T #DATA_LEN
//通常此处做loop循环！！L D[AR1,P#6.0] //找出需要计算数据区的开始地址//Any类型Byte6~Byte9是32位区域地址理解Pointer、Any的类型的数据结构，对于正确使用指针有很大帮助。