SIEMENS/西门子S7-300传感器信号模块6ES7338-4BC01-0AB0

产品名称	SIEMENS/西门子S7-300传感器信号模块6ES7338-4BC01-0AB0
公司名称	上海乘晖科技集团有限公司
价格	.00/台
规格参数	西门子:西门子PLC总代理 西门子PLC:西门子PLC总代理商 德国:西门子PLC一级代理商
公司地址	上海市奉贤区驰华路775号2幢
联系电话	18674345958 18674345958

产品详情

寄存器寻址指针 [AR1/2,P#byte.bit] 这种结构中，P#byte.bit如何参与运算，出*终址呢？

例如：寄存器寻址指针是：[AR1，P#2.6]，我们分AR1=26.4和DBX26.4两种情况来分析。

AR1：26.2

当AR1等于DBX26.4，

+ P#：2.6

= DBX29.7 这是区域间寄存器间接寻址*终确切址数值单元

前面介绍，我们知道，要正确运用寄存器寻址，*重要是对寄存器AR赋值。同样，区分是区域内区域间寻址，也是看AR中赋值。

对AR赋值通常有下面几个方法：

例如：

LAR1

例如：

LAR1

例如：

使用P#这个32位“常数"指针赋值AR。

详解西门子间接寻址（三）

我们先看一段示例程序：

T MW 100 // 将16位整数100传入MW100

结果演变过程就是：8H=1000B=1.0

OPN DB [MW 100] // OPN DB100

T MW[MD2] // T MW1

= M [MD 2] // =M1.0

这个例子中，我们中心思想其实就是：将DB100.DBW1中内容传送到MW1中。这里我们使用了存储器间接寻址两个指针——单字指针MW100用于指DB块编号，双字指针MD2用于定DBW和MW存储区字址。

事实上，从这个例子中心思想来看，根本没有必要如此复杂。但为什么要用间接寻址呢？

例子告诉我们，它*终执行是把DB某个具体字数据传送到位存储区某个具体字中。这是针对数据块1001数据字传送到位存储区第1字中具体操作。我们现需要对同样数据块多个字（连续不连续）进行传送呢？直接方法，就是一句一句写这样具体操作。有多少个字传送，就写多少这样语句。毫无疑问，不知道间接寻址道理，也应该明白，这样编程方法是不合理。而使用间接寻址方法，语句就简单多了。

【示例程序结构分析】

========== 输入1：指数据块编号变量

|| T MW 100

|| L DW#16#8

==========操作主体程序

L DBW [MD 2]

结论：对间接寻址指针内容修改，就完成了主体程序执行结果变更，这种修改是可以是动态和静态。

正是对真正目标程序（主体程序）不做任何变动，而寻址指针是这个程序中一要修改方，可以认为，寻址指针是主体程序入口参数，就好比功能块输入参数。可使程序标准化，具有移植性、通用性。

让我们以一个具体应用，来完善这段示例程序吧：

设计完成这个任务程序之前，我们先了解一些背景知识。

【数据对象尺寸划分规则】

<p helvetica="" neue",="" pingfang="" sc",="" hiragino="" sans="" gb",="" microsoft="" yahei="" ui",="" yahei",="" arial,="" sans-serif;="" letter-spacing:="" 0.544px;="" background-color:="" rgb(255,="" 255,="" 255);="" box-sizing:="" border-box="" !important;="" overflow-wrap:="" break-word="" !important;"="" style="margin-top: 0px; margin-bottom: 24px; padding: 0px; border: 0px; color: rgb(102, 102, 102); font-family: "Microsoft YaHei"; font-size: 14px; white-space: normal; background-color: rgb(255, 255, 255); outline: 0px; max-width: ****; clear: both; min-height: 1em;">数据对象尺寸分为：位（BOOL）、字节（BYTE）、字（WORD）、双字（DWORD）。
这似乎是个简单概念，但，MW10=MB10+MB11，那么是说，MW11=MB12+MB13？你回答是肯定，我建议你继续看下去，不要跳过，这里疏忽，会导致*终程序错误