

# SIEMENS西门子 S-1FL2低惯量型电机 1FL2104-4AF01-1HC0

产品名称	SIEMENS西门子 S-1FL2低惯量型电机 1FL2104-4AF01-1HC0
公司名称	浔之漫智控技术(上海)有限公司
价格	.00/件
规格参数	西门子:原装正品 驱动器电机电缆:假一罚十 德国:现货包邮
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	15801815554 15801815554

## 产品详情

### 全局解释器变量 (S7-1500T) 说明

全局解释器变量对解释器程序的每个程序组织单元均可见。语法 VAR\_IPR/END\_VAR 用于声明全局解释器变量。该声明块必须是主程序的一部分。可根据需要选用。示例

以下示例说明了如何使用全局解释器变量：访问

可从主程序以及从主程序调用的所有函数访问这些变量。全局解释器变量无法在函数中定义。

无法从用户程序访问全局解释器变量。输入参数 (S7-1500T) 说明

形式输入参数被分配函数的当前输入值（进入函数的数据流）。调用函数时，输入参数应用当前输入值。语法 函数的输入参数在声明块 VAR\_INPUT/END\_VAR

中声明。该声明块必须是函数的一部分。可根据需要选用。示例

以下示例说明了如何使用输入参数：访问

只可从函数的代码部分访问这些参数，该部分的声明单元中对参数进行了声明。有关参数使用和相关数据交换的更多信息，请参见“函数参数 (页 114)”。输出参数 (S7-1500T) 说明

形式输出参数用于传输输出值（从块到外部的数据流）。输出参数将当前输出值传送到调用程序组织单元（主程序或指令）。可向其中写入数据或从中读取数据。语法 输出函数的参数在

VAR\_OUTPUT/END\_VAR 声明块中声明。该声明块必须是函数的一部分。可根据需要选用。示例 以下示例说明了如何使用输出参数：可以从函数的代码部分访问这些参数，该部分的声明块中对参数进行了声明。有关参数使用和相关数据交换的更多信息，请参见“函数参数 (页 114)”。常量的类别

(S7-1500T) 说明 常量是数据元素，其值在程序执行期间无法更改。MCL

支持使用以下常量集，具体取决于访问级别：局部常量：仅对声明它们的程序组织单元可见。

全局解释器常量：对解释器程序的每个程序组织单元均可见。

除了变量和函数参数之外，赋值和表达式也使用常量。

常量有一个符号名称。常量的符号名称必须在单独的声明块中单独进行声明。运算符概述 (S7-1500T)

MCL 中的表达式包括运算符。运算符是一种规则，可以从一个操作数创建新操作数。MCL 支持以下几类运算符：赋值运算符 算术运算符 关系运算符 逻辑运算符 元素选择运算符 声明常量 (S7-1500T) 说明

常量在程序组织单元的单独的声明部分中按类型声明。所需的值设置是通过值分配（符号集“:=”）进行的，遵循数据类型规范。初始化时，不能指定任何表达式作为初始化值。只允许使用值或常数。可存在以下声明部分：VAR CONSTANT/END\_VAR 声明局部常数。VAR\_IPR CONSTANT/END\_VAR 声明全局解释器常量。全局解释器常量无法在函数中定义。

下表显示了可以在不同的程序组织单元中声明哪些类型的常量：初始化变量和函数参数 (S7-1500T) 说明  
在声明单元中声明变量和函数参数（函数的输入和输出参数）时，相应变量和参数会收到一个起始值。初始化通过遵循数据类型规范的赋值（符号集“:=”）运算来执行。变量和参数的初始化要初始化变量或参数，必须为变量赋值：赋值运算符 (S7-1500T) 说明

赋值运算符将表达式的值赋给变量。之前的值被覆盖。该变量仅在声明后才存在。更多信息，请参见“声明变量和函数参数 (页 62)”部分。语法

表达式在运算符赋值符号的右侧进行计算。结果存储在变量中，变量的名称位于赋值运算符（目标变量）的左侧。下表列出了赋值运算符的语法规则：组合赋值运算符 (S7-1500T) 说明 MCL 提供了使用组合赋值运算符的选项，即与算术运算符组合的赋值。

在这种情况下，赋值运算在相应的算术运算之后执行。该运算的第一个操作数是运算符左侧的变量，第二个操作数是右侧的表达式。赋值规则 (S7-1500T) 说明 以下规则适用于赋值：

如果变量具有简单数据类型，则在以下条件下可以进行赋值： -

表达式和变量的结果必须具有相同的数据类型。 -

计算得出的表达式的数据类型可以隐式转换为变量的数据类型，前提是所涉及的数据类型允许这样做。

如果变量是复杂数据类型或系统预定义的数据类型，则赋值表达式必须是相同的数据类型。

如果变量是一个 Array，表达式也必须是 Array，并且数据类型必须与 Arrayelement 相同，并具有相同的字段边界。使用聚合指定结构和数组的值

可以使用聚合来指定结构和数组的值。这在以下情况下受支持：变量声明内的初始化 系统函数中的参数

直接赋值给同一类型的一个变量

如果在不同上下文中跨聚合指定结构和数组的值，会导致出现运行时错误（例如在表达式中，函数中的参数等）。规则 数学表达式中运算符的应用顺序取决于它们的优先级。

有关运算符优先级的更多信息，请参见“运算符优先级 (页 81)”部分。

除法运算符（'/'、'MOD'）假定第二个操作数不为零。如果一个数字的类型为

ANY\_INT（整数），另一个数字的类型为 LREAL（长实数），结果将始终是类型 LREAL。

包含数字的算术表达式示例 假设 i 和 j 是整型变量（例如，数据类型为 DINT），值分别为 11 和

-3，以下示例显示整数表达式及其关联值：关系运算符 (S7-1500T) 说明

关系表达式是由关系运算符构成的 BOOL 类型的表达式。语法 关系运算符比较两个操作数的值并返回 BOOL 值作为结果。以这种方式获得的结果为表示 TRUE 或 FALSE 属性的值，

关系运算符的含义如下表所示：1)

执行比较时，采用两个操作数都可以隐式转换成的最不常用的数据类型。 2) 第一个操作数的数据类型。

规则 创建关系表达式时必须遵守以下规则：

操作数应括在括号内，以确保执行逻辑运算的指令是唯一的。

可以根据布尔逻辑规则连接逻辑表达式来创建查询，例如：逻辑运算符 (S7-1500T) 说明 使用

AND、&、XOR 和 OR 逻辑运算符，可以连接一般数据类型 ANY\_BIT（BOOL 或 DWORD）

的操作数和表达式。逻辑运算符 NOT 可以对 ANY\_BIT 数据类型的操作数和表达式取反。规则

逻辑表达式：仅 BOOL 数据类型的操作数。运算符对以下真值表中指定的操作数有效。逻辑

表达式的结果是 1 (TRUE) 或 0 (FALSE)。位串表达式：数据类型 DWORD

的操作数。运算符对以下真值表中给出的操作数有效：元素选择运算符 (S7-1500T) 说明

“.”运算符用于选择结构的元素。它只能用于 STRUCT 数据类型的变量。语法

要访问（写入或读取）结构的特定元素，需要使用“.”符号。结构化变量的父元素后面必须跟有

符号“.”。在该符号之后是子元素，必须遵循以下结构：运算符的优先级 (S7-1500T) 说明

表达式的解释指令取决于以下因素：所用运算符的优先级 先左后右规则 括号的使用 规则

表达式按照一定的规则进行处理：运算符执行与其优先级相对应（见下表）

优先级相同的运算符从左到右执行 标识符开头的减号表示乘以 -1

括号覆盖运算符优先级指令，意味着括号内的内容具有最高优先级

左括号的数量必须等于右括号的数量

IF 指令 (S7-1500T) 说明

IF 指令是条件程序分支。执行顺序取决于给定的条件（一个或多个选项）。执行条件指令会强制检查指定的逻辑表达式。如果逻辑表达式的值为 TRUE，表示满足条件，将执行指令部分。如果值为 FALSE，则表示不满足条件，不执行指令部分。执行顺序 以下内容适用于 IF 指令：

如果第一个表达式的值为 TRUE，则指令部分在 THEN 之后执行。然后程序在 END\_IF 之后继续执行。

如果第一个表达式的值为 FALSE，将对 ELSIF 分支中的表达式进行求值。如果 ELSIF 的一个分支中的布尔表达式为 TRUE，将执行 THEN 后面的指令部分。然后程序在 END\_IF 之后继续执行。

如果 ELSIF 分支中没有任何一个布尔表达式的值为 TRUE，将执行 ELSE 后面的指令序列（如果没有 ELSE 分支，则不再执行指令）。然后程序在 END\_IF 之后继续执行。可以编程任意数量的 ELSIF 指令。

请注意，ELSIF 和 ELSE 指令是可选指令。使用一个或多个 ELSIF 分支而不是一系列 IF 指令的一个好处是，不需要对有效表达式后面的逻辑表达式进行求值。这有助于减少程序的处理时间，并防止执行不需要的程序例程。CASE 指令从 n 个可选项中选择一个程序部分。该选择基于选择表达式的当前值。

该选择是从值列表中进行的，可将程序部分分配给每个值或每组值。执行顺序 以下内容适用于 CASE 指令：将计算 CASE 关键字后面的表达式的值，该表达式必须包含 DINT 或 UDINT 类型的值，用于与值列表进行比较。

然后检查选择值是否列在值列表中。列表中的每个值均对应一个选择表达式所允许的值。

如果存在匹配，则执行程序的相应部分，然后程序在 END\_CASE 之后继续执行。如果没有匹配但存在 ELSE 分支，则执行该分支。否则，程序在 END\_CASE 之后继续执行。值列表 值列表包含选择表达式所允许的值。下图显示了值列表表述的可能变体：表述值列表时请注意以下事项：

每个值列表的开头可以为常量（值）、常量列表（值 1、值 2、值 3，...）或常量序列（值 1 到值 2 - 值 1.. 值 2）。值列表中的值必须是整型常量。

一个值列表中的同一个值可能出现在多个值列表中。但是，仅考虑第一次出现的情况。FOR 指令 (S7-1500T) 说明 作为重复指令，FOR

指令在一次循环中执行一系列指令，并在每次运行时为变量分配值（运行变量）。运行变量必须是 DINT 类型的可写变量。如果编程阶段已执行循环的数量已知，将使用 FOR 指令。在包含 FOR 的循环的定义中，加入了运行变量的起始值和结束值的规范。起始值和结束值表

达式必须与运行变量具有相同的数据类型。如果不知道运行次数，则 WHILE 或者 REPEAT 指令更合适。执行顺序 以下内容适用于 FOR 指令：

在循环开始时，运行变量设置为起始值，每次循环运行后，运行变量都会增加（正增量）或减少（负增量）指定的增量，直至达到结束值。每次运行期间都会检查是否满足以下条件： - 运行变量的值 结束值（如果增量为正），或 - 运行变量的值 结束值（如果增量为负）。

如果条件满足，则执行指令部分。如果不满足条件，则跳过循环从而跳过指令序列，程序在 END\_FOR 之后继续执行。如果循环因上一点所描述的情况而无法执行，则运行变量将保留其当前值。