

西门子PLC模块核心供应商6ES7291-8BA20-0XA0

产品名称	西门子PLC模块核心供应商6ES7291-8BA20-0XA0
公司名称	湖南西控自动化设备有限公司
价格	.00/件
规格参数	西门子:授权代理商 S7-200:一级代理商 德国:售后保障服务
公司地址	中国（湖南）自由贸易试验区长沙片区开元东路1306号开阳智能制造产业园（一期）4#栋301
联系电话	17838383235 17838383235

产品详情

ST语言：电气工程师迈向高工资的必备技能

引言：

ST（Structured Text）语言是一种用于编写工业自动化系统中的可执行程序的编程语言。它具有丰富的运算符和表达式，以及多样化的程序控制语句，给工程师们提供了强大而灵活的工具。本文将带你初步了解ST

语言的基础，从基础概念的了解，增加对ST语言的兴趣，从而为编写程序提供一个可选的解决思路。

2

ST 语言简介：

ST 语言是一种结构化的文本编程语言，它基于 Pascal 语言的语法，并借鉴了 C 和 Ada 语言的特性。ST 语言可用于编写 PLC（可编程逻辑控制器）等自动化设备的程序。它具备代码的模块化、易读性和可维护性，使得工程师能够高效地开发、调试和维护自动化系统。

3

运算符介绍：

ST 语言中有多种运算符，用于执行各种数学和逻辑操作。以下是一些常用的运算符：

3.1

算术运算符：+（加）、-（减）、*（乘）、/（除）、**（幂运算）、MOD（取余）等。

示例：

```
a := 10; b := 2; result_1 := a + b; // 结果为 12 result_2 := a
-b; // 结果为 8 result_3 := a * b; // 结果为 20 result_4 := a / b;
// 结果为 5 result_5 := a ** b; // 结果为 100 result_6 := a MOD b;
// 结果为 0
```

3.2

赋值运算符：=

示例：

```
a := 10; b := a; // 将 a 的值赋给 b
```

3.3

关系运算符：=（等于）、<>（不等于）、>（大于）、<（小于）、>=（大于等于）、<=（小于等于）等。

示例：

```
a := 10; b := 5; result_1 := a = b; // 结果为 FALSE result_2 :=  
a <> b; // 结果为 TRUE result_3 := a > b; // 结果为 TRUE result_4  
:= a < b; // 结果为 FALSE result_5 := a >= b; // 结果为 TRUE result_  
6 := a <= b; // 结果为 FALSE
```

3.4

逻辑运算符：AND（与）、OR（或）、NOT（非）等。

示例：

```
a := TRUE; b := FALSE; result_1 := a AND b; // 结果为 FALSE  
result_2 := a OR b; // 结果为 TRUE result_3 := a AND NOT b; // 结果  
为 TRUE
```

4

表达式介绍：

ST 语言中的表达式用于计算和产生值。以下是一些常见的表达式：

4.1

赋值表达式：

赋值表达式用于将一个值赋给一个变量或对象。例如：a := 10;

4.2

算术表达式：

算术表达式用于执行数学运算。例如：`result := a + b;`

4.3

关系表达式：

关系表达式用于比较两个值之间的关系。例如：`result := a > b;`

4.4

逻辑表达式：

逻辑表达式用于判断逻辑条件是否成立。例如：`result := a AND b;`

5

程序控制语句介绍：

ST

语言提供了丰富的程序控制语句，用于控制程序的流程和执行顺序。以下是一些常用的程序控制语句：

5.1

条件语句 IF：

IF 语句根据条件的真假来执行不同的代码块。示例：

```
IF a > 10 THEN // 执行某些操作 ELSE // 执行其他操作 END_IF
```

5.2

循环语句 FOR：

FOR 语句用于重复执行一个代码块，直到达到指定的条件。示例：

```
FOR i:= 1 TO 10 DO // 执行循环体操作 END_FOR
```

5.3

循环语句 WHILE：

WHILE 语句根据条件的真假来重复执行一个代码块。示例：

```
WHILE condition DO // 执行循环体操作 END_WHILE
```

5.4

循环语句 REPEAT UNTIL：

REPEAT UNTIL 语句在条件为真之前重复执行一个代码块。示例：

```
REPEAT // 执行循环体操作 UNTIL condition
```

5.5

CASE OF 语句：

CASE OF 语句根据表达式的不同取值执行相应的代码块。示例：

```
CASE expression OF value1: // 执行操作1
value2: // 执行操作2 ELSE // 执
行其他操作 END_CASE
```

5.6

RETURN 语句：

RETURN 语句用于在函数或函数块中返回一个值并终止执行。示例：

```
FUNCTION ExampleFunction : INT // 执行一些操作 RETUR
N 42; END_FUNCTION
```

5.7

EXIT 语句：

EXIT 语句用于提前终止循环或程序块的执行。示例：

```
FOR i:= 1 TO 10 DO IF i= 5 THEN EXIT;
// 提前退出循环 END_IF // 执行循环体操作 END_FOR
```

6

总结

当你学习 ST 语言时，这是一个基于结构化文本的编程语言，特别适用于工业自动化领域。ST 语言有以下几个重要的特点和要点：

结构化：ST 是一种结构化编程语言，它使用代码块、条件语句和循环语句来组织程序。这样可以使程序更加清晰和易于阅读。

数据类型：ST 支持多种数据类型，包括基本数据类型（例如整数、浮点数和布尔值）以及复杂数据类型（例如数组和结构体）。你可以使用这些数据类型来存储和操作数据。

变量和常量：在 ST 语言中，你可以声明和使用变量和常量。变量是用于存储和操作数据的内存空间，而常量是在程序执行期间不会改变的值。

运算符和表达式：ST 提供了各种运算符（例如算术运算符、逻辑运算符和关系运算符），你可以使用这些运算符来执行各种数学和逻辑操作。

条件语句：ST 提供了条件语句（例如 IF-THEN-ELSE 语句和 CASE 语句），用于根据条件的真假来执行不同的代码块。这使得你可以根据特定的情况采取不同的操作。

循环语句：ST 提供了循环语句（例如 FOR 循环、WHILE 循环和 REPEAT UNTIL 循环），用于重复执行一段代码块，直到满足特定的条件为止。

函数和函数块：ST 支持函数和函数块的定义和使用。你可以将一段可重复使用的代码封装到函数或函数块中，并通过调用函数来执行这段代码。

调试和错误处理：ST 支持调试和错误处理的机制。你可以使用调试工具来检测和修复程序中的错误，并使用异常处理机制来处理异常情况。

总的来说，ST 语言是一种强大的编程语言，特别适用于工业自动化领域的程序开发。通过学习 ST 语言，你将能够编写出高效、可靠且易于维护的工控程序。祝你在 ST 语言的学习和实践中取得成功！