

产品详情6ES7952-1AS00-0AA0西门子SIMATIC S7，RAM 存储卡 针对 S7-400，

产品名称	产品详情6ES7952-1AS00-0AA0西门子SIMATIC S7，RAM 存储卡 针对 S7-400，
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:西门子授权代理商 S7-400:核心供货商 德国:现货
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

SCL与STL的区别是什么？SCL常见问题及解决办法！

一、SCL 与 STL

SCL (Structured Control

Language) 语言，也就是结构化控制语言，它其实和IEC标准制定的ST语言是一样的。S7-SCL 可以编译成 STL, 虽然其代码量相对于 STL 编程有所增加，但我们更关心的是程序结构和程序的总体效率。类似于计算机行业的发展，汇编语言已经被舍弃，取而代之的是 C/C++等语言。S7-SCL 对工程设计人员要求较高，需要其具有一定的计算机语言的知识 and 编程技巧。

STL是STEP7时代推出的语言，它不是STRUCT TEXT LANGUAGE的缩写，所以它和ST语言完全没关系。STL语言是一种汇编语言，是面向机器底层的语言，可以直接操作寄存器，可读性不强，和80C51单片机的机器语言是一个层面。所以它是一种低级语言，而ST和SCL都是中语言。

二、S7-SCL 常见问题

1) 如何规范地的开发一个 S7-SCL 程序？

解答：开发一个 S7-SCL 程序，应当遵循如下流程：

规划需要的块类型，即程序的整体结构；

规划子任务（FB,FC 等等）；

定义各个子任务的之间输入/输出接口；

定义各个子任务在原文件中的顺序及调用关系；

定义符号表

编译检查；

下载调试。

2) S7-SCL 支持哪些 STEP7 块，与其它语言有什么关系？

解答：S7-SCL 支持如下

STEP7 的块：OB,FC,FB,DB,UDT。

在 S7 程序中，S7-SCL 块可以与其它STEP7 编程语言生成的块互相调用。

S7-SCL 可以被编译成 STL，S7-SCL 生成的块也可以作为库文件被其它语言引用。

由于 S7-SCL 程序由 ASCII 文本构成，所以它非常容易被导入或导出。

3) 为什么双击打开 S7-SCL 类型的程序后，显示的格式却是 STL 格式？

解答：S7-SCL 程序编译后生成的执行代码实际上为 STL 格式，如下图中的 FC1, 虽然其标示为 SCL 格式，但如果用户将 Source 目录中的 FC1 的源文件删除后，再双击打开 FC1 后，显示的将为 STL 格式。

4) 什么是 OK Flag？

解答：在 S7-SCL 程序中，OK 是一个布尔类型的特殊本地变量。它用来显示块执行是否正确。

OK 标志的使用与 STL 编程中对于 FC/FB 的 ENO 处理（STL 是用状态字的 BR 位来保存状态）非常相似。

在程序的开始，可以将 OK 标志赋值为 TRUE,如果程序执行有错误（例如 0 作为除数），则当程序退出时，用户通过程序将 OK 标志改写为 FALSE，此数值将被存储在输出参数 ENO 当中。

OK 标志是一个系统变量，不需要定义，如果用户希望在程序中使用此标志，可以在 S7-SCL编程菜单 Options>Customize>Compiler 中，应当选择 Set OK flag.项。否则 ENO 将无法被评估。

OK 标志使用的例子：

```
OK:= TRUE;
```

```
Division:= 1 / IN;
```

```
IF OK THEN // 除法操作正常;
```

```
// 此处可以添加在除法操作正常情况下，其它的用户程序ELSE //除法操作不正常;
```

```
// :END_IF;
```

5) 在 STL 中可以读取一个组织块的信息 (例如 OB1 中的循环时间)，在 S7-SCL 中是否存在同样的可能？

解答：STEP7 中的任意 OB 块在生成之后，其内部都自动生成一些特有的临时变量，例如 OB1 中的变量 OB1_PREV_CYCLE 即为 CPU 上个扫描周期所用的时间。在 S7-SCL 编辑器中，通过“Insert>Block Template > OB”创建一个 OB 模板。

在缺省情况下，此模板的临时变量区域将创建一个 ARRAY OF BYTE 类型的临时变量。此区域与 OB 块的临时变量区域是相对应的，但由于没有变量名称标识，并且数据类型也不相同，所以用户使用起来非常不方便。如果用户想要使用 S7-SCL 格式的 OB 块内的特有变量，请按下列步骤进行：

在路径 STEP7 的安装路径 \Step7\S7DATA\S7wiz 下，包含所有 OB 块的 STL 格式模板，可以使用写字板打开这些模板 (例如 OB31.awl)，并复制相应变量定义。

创建一个新的 S7-SCL 源文件并插入刚才复制的变量定义。

调整 S7-SCL 源文件中的语法结构；

在这个程序的开始部分，可以设置 "Title" 和 "Version"，例如 Title=' ' 和 Version='0.1'；

在 BEGIN 语句后面插入一个分号，至此，即完成格式修改。另外，也可以打开本文中的 S7-SCL 例子目录中的 OB_Templ 项目，其中包括了 44 个 S7-SCL 源文件格式的组织块。

6) 如何理解 FC/SFC 的输出参数 RET_VAL (返回值) ?

解答：对于此问题，如果读者对 PASCAL 语言或者编程语言非常熟悉，可能此问题就非常容易解释。许多 SFC (系统功能) 都有输出参数 RET_VAL (返回值)，它提供一个可供评估的错误代码。STEP 7 在线帮助中提供有更多关于系统功能和输出参数 RET_VAL 的信息。

通过在 SIMATIC 管理器中选中 SFC 然后按 “ F1 ” 键可以获取相关的在线帮助信息。同样，一个 FC 也可以返回一个结果，下面给出了一些例子，说明了如何在 S7-SCL 中调用带有返回值 (RET_VAL) 的功能 (FC 或 SFC)。

定义 FC1, 函数类型为 INT (返回值)

调用 FC1, 得到返回值：

调用一个系统功能 (SFC24)：当调用系统功能 SFC 24 “ TEST_DB ” (测试数据块) 时，得到关于 CPU 主内存中一个数据块的信息。SFC24 确定指定 DB 的数据字节数，并检查该 DB 是否受到写保护。选定 DB 所包含的数据字节数通过参数 “ DB_LENGTH ” 输出，参数 “ WRITE_PROT ” 包含关于选定 DB 的写保护 ID 的信息 (FALSE 表示没有写保护)。

7) 在 S7-SCL 程序中，PLC 中的地址标识与 STL/LAD 中的地址标识有何区别，使用中有何注意事项？

解答：在 S7-SCL 程序中，PLC 中的地址标识与 STL/LAD 中的地址标识基本相同，方便了编程人员快速掌握 S7-SCL 语言。下图为 PLC 中的 S7-SCL 程序地址标识格式，图中一些格式有些特殊，实际上，下面的格式在 S7-SCL 中都是正确的。

```
DB1.D0.0:=DB1.DBX0.1;
```

```
DB1.DB1:=DB1.DBB1;
```

```
DB1.DW2:=DB1.DBW2;
```

DB1.DD4:=DB1.DBD4;

QX0.0:=Q0.0;

IX0.0:=I0.0;

MX0.0:=M0.0;

所以说，在一般情况下，编程人员依旧可以 STL 的标识方式用于 S7-SCL。注意：如果数据块编号及地址都使用了间接寻址方式，则必须使用 S7-SCL 特定的标识方式，例如如下语句：

STATUS_1:=WORD_TO_BLOCK(INDEX).DW [COUNTER]; (此格式正确)

STATUS_1:=WORD_TO_BLOCK(INDEX).DBW [COUNTER]; (此格式错误)

上图为 S7-SCL 中的 PLC 地址格式。x 可以为 0 至 65535 之间的数字（地址），y 可以为 0 至 7 之间的数字（位地址），对于其它 FB,OB,DB, T,C 等等的标识格式，S7-SCL 格式基本与 STL 格式相同，此处不做说明。

8) 为什么用户人工输入的程序在编译时经常提示语法错误？

解答：S7-SCL 程序和所有的语言一样，有着自己特定的语法，而很多编程人员容易在人工输入程序时忽视语法格式，这样就造成编译错误。因此强烈建议编程人员使用 S7-SCL 的模板向导。在菜单 Insert 中，提供了 Block Call, Block Template, Control Structure, 三个向导，用户可以通过使用此向导，提高输入效率。例如在下图中指出了两个常见错误，用户如果使用向导，就不容易出错误：

9) S7-SCL 程序中如何使用多重背景？

解答：S7-SCL 程序中可以使用多重背景，下面的程序列出了正确格式，声明完成后，在程序中即可调用：

Supply1 : FB10; //将 Supply1 定义为 FB10 类型；

Supply2,Supply3,Supply4 : FB100;

//将 Supply2,Supply3,Supply4 定义为 FB100 类型。

10) 如何在 S7-SCL 程序中实现数据块间接寻址？

解答：S7-SCL 程序中对数据块间接寻址提供了良好的支持，编程也很简单。

```
STATUS_1:= DB11.DW[COUNTER]; //字节间接寻址；
```

```
STATUS_2:= DB12.DX[WNO, BITNO]; //位间接寻址，用户改变 WNO,BITNO 数值即可；
```

```
STATUS_1:= Database1.DW[COUNTER] ; // Database1 为 DB 类型的本地变量；
```

```
STATUS_2:= Database2.DX[WNO, BITNO];
```

```
STATUS_1:=WORD_TO_BLOCK_DB(INDEX).DW[COUNTER];
```

//INDEX 被定义为 BLOCK_DB 数据类型，COUNTER 为整数数据类型，这样可以实现数据块编号，字节地址同时间接寻址，此功能是一种功能很强间接寻址方式。

```
STATUSBYTE :=DB101.DB10; //字节直接寻址格式；
```

```
STATUS_3 :=DB30.D1.1; //位直接寻址格式；
```

```
MEASVAL :=DB25.DW20; //字直接寻址格式；
```

```
STATUSBYTE :=Status_data.DB10;
```

```
STATUS_3:="New data ".D1.1;
```

```
MEASVAL :=Measdata.DW20;
```

```
STATUS_1 :=WORD_TO_BLOCK_DB (INDEX).DW10;
```

11) 在 S7-SCL 程序中调用 FC/FB 与在 STL/LAD 中调用 FB/FC 有何区别？

解答：在 STL/LAD 程序中调用 FB 可以不把参数填写完整，但在 S7-SCL 程序中调用 FB 时，必须把 FB 参数填写完整；对于 FC 的调用 S7-SCL 与 STL/LAD 区别不大。强烈建议编程人员使用 S7-SCL 的模板向导调用 FB。

12) 为了将 S7-SCL 中的位数据类型 (BYTE, WORD 或 DWORD) 转换到 “ REAL ” 数据类型时，需要使用哪种数据类型转换程序？

解答：为了将 S7-SCL 中的位数据类型 (BYTE, WORD 或 DWORD) 转换到数字数据类型 “ REAL ” 时，需要使用以下指令：

从 BYTE 到 REAL：“DINT_TO_REAL(DWORD_TO_DINT(BYTE_TO_DWORD(value)))”

从 WORD 到 REAL：“DINT_TO_REAL(DWORD_TO_DINT(WORD_TO_DWORD(value)))”

从 DWORD 到 REAL：“DINT_TO_REAL(DWORD_TO_DINT(value))”

注意事项：用户在将 BYTE, WORD 或 DWORD 数据类型转换到 “ REAL ” 数据类型时，S7-SCL 中常见的错误可以参考下列：

从 WORD 到 REAL：“DWORD_TO_REAL(WORD_TO_DWORD(value))” (错误程序)。

如果执行了这一转换，会得到错误的结果。原因：DINT_TO_REAL 函数将把源数据转换并按照 IEEE REAL 的格式存储到目的变量，而 DWORD_TO_REAL 仅仅是从源数据拷贝位串至目的变量。因此，为了获得正确的 “ REAL ” 数，必须总是通过中间步骤 “ DWORD_TO_DINT ” 和 “ DINT_TO_REAL ” 实现。

13) 在 S7-SCL 中如何区分某个变量名是本地变量，还是符号名？

解答：S7-SCL 中本地变量与符号名的引用有如下区别：

如果引用中只有变量名，则此变量为本地变量，如：Motor_Status:= Motor_1；

如果引用中变量名称上包括 “ ”，则此变量为符号名，如：Motor_Status:= “ Motor_1 ”。

14) 如何访问一个字符串中的单个字符？

解答：对于字符串中的单个字符，使用如下的访问格式，是错误的：

MB10 := str[5] 这样的表达式无法实现将字符串的第 5 个字符送至 MB10 中。

方法：可以使用 IEC 函数 MID 来得到字符串的某个部分，例子如下：

VAR

str : STRING[20];

END_VAR

MB10:=CHAR_TO_BYTE(String_to_char(MID(IN:=str,L:=1,P:=5)));

15) 如何加密我的 S7-SCL 程序？

解答：S7-SCL 程序加密与 STL 程序加密方法几乎一样，用户可以在程序中加入关键字：

KNOW_HOW_PROTECT，再进行编译即可，下图中的 FC1 在编译后，即出现了加锁保护的图标。

注意：提供给终用户的项目中应当删除 S7-SCL 源代码，否则用户在双击加密的块后（例如下图中的 FC1），依然可以关联打开 S7-SCL 源代码。

注意：提供给终用户的项目中如果删除 S7-SCL 源代码，用户将只能看到 STL 格式的程序，稍微复杂些的 S7-SCL 程序在 STL 格式下几乎是无法阅读的。此方法类似于加密的效果。

16) 如何对 S7-SCL 程序中的数据块及静态变量初始化？

解答：S7-SCL 程序中对数据块及静态变量初始化和 STL 语言格式类似，下面的程序列出了正确格式：

DAT1 : REAL := 100.5;

A1 : INT := 10;

A2 : STRING[6] := 'FACTOR';

A3 : ARRAY[1..12] OF REAL := 0.0, 10(100.0), 1.0;

