

# 西门子 S7-1200扩展信号板模块 6ES7223-3AD30-0XB0总经销商

产品名称	西门子 S7-1200扩展信号板模块 6ES7223-3AD30-0XB0总经销商
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:代理经销商 S7-1200:全新原装 假一罚十 德国:正品现货 实体经营
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

西门子 S7-1200扩展信号板模块 6ES7223-3AD30-0XB0使用 OB 组织用户程序 组织块为程序提供结构。它们充当操作系统和用户程序之间的接口。OB 是由事件驱动的。事件（如诊断中断或时间间隔）会使 CPU 执行 OB。某些 OB 预定义了起始事件和行为。程序循环 OB 包含用户主程序。

用户程序中可包含多个程序循环 OB。RUN 模式期间，程序循环 OB

以\*低优先级等级执行，可被其它事件类型中断。启动 OB 不会中断程序循环 OB，因为 CPU 在进入 RUN 模式之前将先执行启动 OB。完成程序循环 OB 的处理后，CPU 会立即重新执行程序循环 OB。该循环处理是用于可编程逻辑控制器的“正常”处理类型。

对于许多应用来说，整个用户程序位于一个程序循环 OB 中。可创建其它 OB

以执行特定的功能，如用于处理中断和错误或用于以特定的时间间隔执行特定程序代码。这些 OB 会中断程序循环 OB 的执行。简化了编程 6.1 轻松设计用户程序 入门手册 102 设备手册, 01/2015,

A5E02486780-AG 使用“添加新块”(Add new block)对话框在用户程序中创建新的 OB。

总是由事件驱动中断处理。发生此类事件时，CPU 会中断用户程序的执行并调用已组态用于处理该事件的 OB。完成中断 OB 的执行后，CPU 会在中断点继续执行用户程序。CPU

按优先级确定处理中断事件的顺序。可为多个中断事件分配相同的优先级。

更多相关信息，请参见组织块(页 61)和执行用户程序(页 60)。创建附加 OB

可为用户程序，甚至为程序循环和启动 OB 事件创建多个 OB。使用“添加新块”(Add new block)对话框创建 OB 并为 OB 输入名称。如果为用户程序创建了多个程序循环 OB，则 CPU

会按数字顺序从具有\*小编号（例如 OB 1）的程序循环 OB 开始执行每个程序循环 OB。例如：

当第一个程序循环 OB（例如 OB 1）完成后，CPU 将执行下一个编号更高的程序循环 OB。简化了编程 6.1 轻松设计用户程序 入门手册 设备手册, 01/2015, A5E02486780-AG 103 组态 OB 的属性 可对 OB

的属性进行修改。例如，可组态 OB 编号或编程语言。说明

请注意，您可将局部过程映像编号分配给对应于 PIP0、PIP1、PIP2、PIP3 或 PIP4 的 OB。

如果您为局部过程映像编号输入编号，则 CPU 将创建该过程映像分区。

有关过程映像分区的说明，请参见主题“执行用户程序(页 60)”。6.1.2 FB 和 FC

使模块化任务编程变得很轻松 功能 (FC) 与子例程类似。 FC 是通常对一组输入值执行特定运算的代码块。 FC 将此运算结果存储在存储单元中。 使用 FC 可执行以下任务：

- 执行标准和可重复使用的运算，如数学计算
- 执行功能任务，如通过使用位逻辑运算进行单独控制 也可以在程序中的不同位置多次调用 FC。 此重复使用简化了对经常重复发生的任务的编程。 与 FB 不同，FC 不具有相关的背景 DB。 FC 使用其临时存储器 (L) 保存用于计算运算的数据。 不保存临时数据。 要存储数据以备 FC 执行完成后使用，可将输出值赋给全局存储单元，如 M 存储器或全局 DB。 功能块 (FB) 与带存储器的子例程类似。 FB 是可通过块参数以编程方式实现其调用的代码块。 FB 将输入 (IN)、输出 (OUT) 和输入/输出 (IN\_OUT) 参数存储在数据块 (DB) 或“背景”DB 中的变量存储器内。 背景 DB 提供与 FB 的实例（或调用）关联的一块存储区并在 FB 完成后存储数据。 简化了编程 6.1 轻松设计用户程序 入门手册 104 设备手册, 01/2015, A5E02486780-AG 用户通常使用 FB 控制在一个扫描周期内未完成其运行的任务或设备的运行。

要存储运行参数以便从一个扫描快速访问到下一个扫描，用户程序中的每一个 FB 都应具有一个或多个背景 DB。 调用 FB 时，也会打开存储块参数的值以及用于该 FB 调用或“背景”的静态局部数据的背景 DB。 这些值会在 FB 完成之后存储在背景数据块中。 可以给 FB 接口中的参数赋初值。 这些值将传送到相关的背景 DB 中。 如果未分配参数，将使用当前存储在背景 DB 中的值。 某些情况下，必须分配参数。 可将不同的背景 DB 与 FB 的不同调用进行关联。 通过背景 DB 可使用一个通用 FB 控制多个设备。 通过使一个代码块对 FB 和背景 DB 进行调用，来构建程序。 然后，CPU 执行该 FB 中的程序代码，并将块参数和静态局部数据存储在背景 DB 中。 FB 执行完成后，CPU 会返回到调用该 FB 的代码块中。 背景 DB 保留该 FB 实例的值。

通过设计用于通用控制任务的 FB，可对多个设备重复使用 FB，方法是：为 FB 的不同调用选择不同的背景 DB。 下图显示了三次调用同一个 FB 的 OB，方法是针对每次调用使用一个不同的数据块。 该结构使一个通用 FB 可以控制多个相似的设备（如电机），方法是在每次调用时为各设备分配不同的背景数据块。 每个背景 DB 存储单个设备的数据（如速度、加速时间和总运行时间）。 在此实例中，FB 22 控制三个独立的设备，其中 DB 201 用于存储第一个设备的运行数据，DB 202 用于存储第二个设备的运行数据，DB 203 用于存储第三个设备的运行数据。