

SIEMENS西门子 中国襄阳市智能化工控设备代理商

产品名称	SIEMENS西门子 中国襄阳市智能化工控设备代理商
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 模块: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

复制保护附加安全特性允许捆绑程序块，以用于特定存储卡或 CPU。该特性对于保护您的知识产权特别有用。当您将程序块与特定设备捆绑在一起时，就会将程序或代码块限制为仅用于特定存储卡或 CPU。该特性允许您以电子方式（如通过 Internet 或通过电子邮件）或通过发送内存匣分布程序或代码块。复制保护可用于 OB(页 101)、FB (页 103) 和 FC (页 103)。S7-1200 CPU 支持三种类型的块保护：与 CPU 的序列号进行绑定 与存储卡的序列号进行绑定 与强制性密码动态绑定 使用代码块的“属性”(Properties) 任务卡将块捆绑到特定 CPU 或存储卡。1. 打开代码块之后选择“保护”(Protection)。2. 在“复制保护”(Copy protection) 任务下的下拉列表中，选择要使用的复制保护的类型。3. 对于与 CPU 或存储卡序列号的绑定，可以在下载时插入序列号，也可以输入存储卡或 CPU 的序列号。说明序列号区分大小写。对于与强制性密码的动态绑定，定义下载或复制块所必须使用的密码。随后下载带有动态绑定的块时，必须输入可用于下载块的密码。请注意，复制保护密码和专有技术保护(页 95)密码是两个不同的密码。轻松创建设备配置 5.8 防止对 CPU 或代码块的访问非常容易入门手册 98 设备手册, 01/2015, A5E02486780-AG 入门手册 设备手册, 01/2015, A5E02486780-AG 99 简化了编程 66.1 轻松设计用户程序 创建处理自动化任务的用户程序时，需要将程序指令插入代码块（OB、FB 或 FC）中。为用户程序选择结构类型根据实际应用要求，可选择线性结构或模块化结构来创建用户程序。线性程序按顺序逐条执行处理自动化任务的所有指令。通常，线性程序将所有程序指令都放入一个程序循环 OB (如 OB 1) 中以循环执行该程序。模块化程序调用可执行特定任务的特定代码块。要创建模块化结构，需要将复杂的自动化任务划分为与过程所执行的功能任务相对应的更小的次级任务。每个代码块都为各个次级任务提供程序段。通过从另一个块中调用其中一个代码块来构建程序。线性结构：模块化结构：通过设计 FB 和 FC 执行通用任务，可创建模块化代码块。然后可通过由其它代码块调用这些可重复使用的模块来构建用户程序。调用块将设备特定的参数传递给被调用块。当一个代码块调用另一个代码块时，CPU 会执行被调用块中的程序代码。执行完被调用块后，CPU 会继续执行调用块，并继续执行该块调用之后的指令。简化了编程 6.1 轻松设计用户程序 入门手册 100 设备手册, 01/2015, A5E02486780-AG 还可以将 OB 分配给中断事件。发生该事件时，CPU 会执行相关 OB

中的程序代码。OB的执行完成后，CPU会在发生中断事件的时间点在用户程序中恢复执行，而该点可能在扫描过程中的任何位置。A 调用块（或中断的块）B 调用的 FB 或 BC（或中断 OB）程序执行用于触发其它块执行的指令（或中断事件）程序执行块结束（返回到调用块）可嵌套块调用以实现更加模块化的结构。在以下示例中，嵌套深度为 3：程序循环 OB 加 3 层对代码块的调用。循环开始

嵌套深度通过创建可在用户程序中重复使用的通用代码块，可简化用户程序的设计和实现。可为标准任务创建能够重复使用的代码块，如用于控制泵或电机。也可以将这些通用代码块存储在可由不同的应用或解决方案使用的库中。将用户程序构建到与功能任务相关的模块化组件中，可使程序的设计更易于理解和管理。模块化组件不仅有助于标准化程序设计，也有助于使更新或修改程序代码更加快速和容易。简化了编程6. 创建模块化组件可简化程序的调试。通过将整个程序构建为一组模块化程序段，可在开发每个代码块时测试其功能。

利用与特定功能任务相关的模块化设计，可以减少对已完成的应用程序进行调试所需的时间。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 以最低优先级等级执行，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。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 newblock) 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最小编号（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。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。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。数据块为程序数据提供了便捷的存储方式在用户程序中创建数据块 (DB) 以存储代码块的数据。用户程序中的所有程序块都可访问全局 DB 中的数据，而背景 DB 仅存储特定功能块 (FB) 的数据。用户程序可将数据存储在 CPU 的专用存储区中，如输入 (I)、输出 (Q) 和位存储器 (M)。此外，可使用数据块 (DB) 快速访问存储在程序本身中的数据。当数据块关闭或相关代码块的执行结束时，DB 中存储的数据不会被删除。有两种类型的 DB：全局 DB 存储程序中代码块的数据。任何 OB、FB 或 FC 都可访问全局 DB 中的数据。背景 DB 存储特定 FB 的数据。背景 DB 中数据的结构反映了 FB 的参数 (Input、Output 和 InOut) 和静态数据。FB 的临时存储器不存储在背景 DB 中。尽管背景 DB 反映特定 FB 的数据，然而任何代码块都可访问背景 DB 中的数据。简化了编程 6.1 轻松设计用户程序入门手册 106 设备手册, 01/2015, A5E02486780-AG6.1.4

创建新代码块要在程序中添加新的代码块，请按以下步骤操作：1. 打开“程序块”(Program blocks) 文件夹。2. 双击“添加新块”(Add new block)。3. 在“添加新块”(Add new block) 对话框中单击要添加的块的类型。例如，单击“功能(FC)”图标来添加 FC。4. 从下拉菜单中为代码块选择编程语言。5. 单击“确定”(OK) 将块添加到项目中。选择“添加新对象并打开”(Add new and open) 选项（默认），让 STEP 7 在编辑器中打开新创建的块。创建可重复使用的代码块使用项目浏览器中“程序块”(Program blocks) 下的“添加新块”(Add newblock) 对话框创建 OB、FB、FC 和全局 DB。创建代码块时，需要为块选择编程语言。无需为 DB 选择语言，因为它仅用于存储数据。选中“添加新对象并打开”(Add new and open) 复选框（默认），在项目视图中打开代码块。可存储想要在库中重复使用的对象。每个项目都有一个与之相连的项目库。除项目库外，您还可以创建可在多个项目中使用的任意数量的全局库。由于库彼此兼容，因此可以复制库要素并将其从一个库移动到另一个库。库可用于创建块的模板：首先将块粘贴到项目库中，随后在其中进一步开发块。最后，将块从项目库复制到全局库。可将全局库共享给正在使用项目的其他同事。他们可使用块并根据需要进一步调整块以满足各自的需求。有关库操作的详细信息，请参见 STEP 7 在线帮助库主题。从一个代码块调用另一个代码块用户可以很容易地使用用户程序中的任何代码块 (OB、FB 或 FC) 调用 CPU 中的 FB 或 FC。1. 打开将调用另一个块的代码块。2. 在项目树中，选择要调用的代码块。3. 将该块拖到所选网络中以创建一个对代码块的调用。说明用户程序无法调用 OB，因为 OB 是由事件驱动的 (页 62)。CPU 接收到事件时，将开始执行 OB 进行响应。