

SIEMENS

SIMATIC

S7-1200 S7-1200 入门指南

入门指南

前言

快速回顾

1

安装

2

创建简单锁存电路

3

完成用户程序

4

使用监视表格进行监视

5

11/2009

A5E02486796-01

法律资讯

警告提示系统

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。

 危险
表示如果不采取相应的小心措施， 将会 导致死亡或者严重的人身伤害。
 警告
表示如果不采取相应的小心措施， 可能 导致死亡或者严重的人身伤害。
 小心
带有警告三角，表示如果不采取相应的小心措施，可能导致轻微的人身伤害。
小心
不带警告三角，表示如果不采取相应的小心措施，可能导致财产损失。
注意
表示如果不注意相应的提示，可能会出现不希望的结果或状态。

当出现多个危险等级的情况下，每次总是使用最高等级的警告提示。如果在某个警告提示中带有警告可能导致人身伤害的警告三角，则可能在该警告提示中另外还附带有可能导致财产损失的警告。

合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

按规定使用 Siemens 产品

请注意下列说明：

 警告
Siemens 产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到 Siemens 推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

商标

所有带有标记符号 © 的都是西门子股份有限公司的注册商标。标签中的其他符号可能是一些其他商标，这是出于保护所有者权利的 目地由第三方使用而特别标示的。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

前言



在致力于实现“全集成自动化”(TIA, Totally Integrated Automation)的过程中,作为 SIMATIC 中的一员, S7-1200 产品系列和 STEP 7 Basic 编程软件包为您提供了解决创建自动化应用的工具。S7-1200 紧凑型控制器设计紧凑、成本低廉且功能强大,是控制小型应用的完美解决方案。

本文档中的示例将引导您初步了解 S7-1200 系列,在尽可能短的时间内教会您基本技巧。

说明

查看《S7-1200 可编程控制器系统手册》(*S7-1200 Programmable Controller System Manual*) 获取安全准则

本快速入门指南仅仅让用户快速了解 S7-1200 系列。在进行本指南中的练习之前,请查看《SIMATIC S7-1200 可编程控制器系统手册》(*SIMATIC S7-1200 Programmable Controller System Manual*) 中的安全准则和其它信息。

部件列表

说明	数量	订货号
CPU 1212C AC/DC/RLY	1	6ES7212-1BD30-0XB0
仿真器: SIM 1274 (数字量输入)	1	6ES7274-1XF30-0XA0
标准以太网 CAT5 电缆	1	6XV1850-2GH20
STEP 7 Basic 编程软件	1	6ES7822-0AA00-0YA0

该快速入门套件还包括 S7-1200 文档集 CD。

说明

必须为 CPU 提供电源线。

CPU 1212C AC/DC/RLY 需要使用 120/240 VAC。必须提供电源线并将其连接到 CPU。有关其它 CPU 的电源要求和安装过程，请参考《S7-1200 可编程控制器系统手册》(S7-1200 Programmable Controller System Manual)。

联系信息

更多相关信息，请参考《S7-1200 可编程控制器系统手册》(S7-1200 programmable controller system manual)。可使用客户支持网站 (<http://www.siemens.com/automation/support-request>) 搜索具体的产品信息，或与技术支持代表联系。

如需了解任何技术问题、培训或订购 S7 产品方面的信息，请与西门子经销商或销售部联系。西门子销售代表受过技术培训并掌握有关您的操作、过程和行业以及您使用的各种西门子产品的专业知识，所以他们能够最快速高效地回答您可能遇到的任何问题。



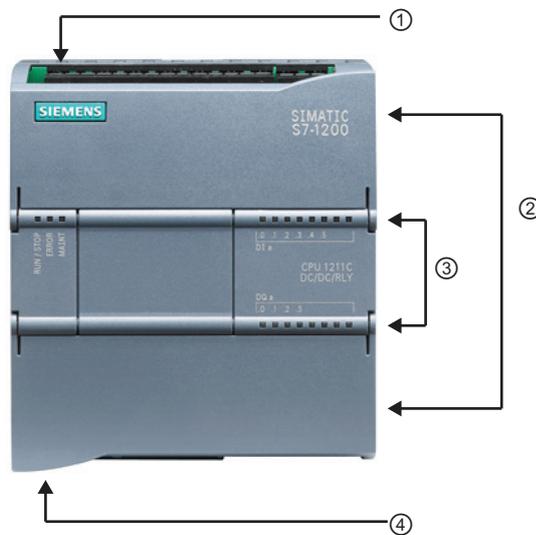
目录

前言	3
1 快速回顾	7
1.1 有关 CPU 工作原理的基本信息	8
1.1.1 每个扫描周期均执行的任务	8
1.1.2 CPU 的操作模式	9
1.1.3 以位、字节、字等形式存储的数字数据	10
1.1.4 S7-1200 支持的数据类型	11
1.1.5 S7-1200 的存储区	12
1.2 从示意图到用户程序	14
1.2.1 将示意图转换为 LAD 指令	15
1.2.2 基本电路	17
1.3 STEP 7 Basic 的基本功能	19
1.3.1 在您需要时提供的帮助	20
1.3.2 从在线帮助中打印	21
1.3.3 插入指令	22
1.3.4 在编辑器之间拖放	23
1.3.5 检测 CPU 的组态	24
1.3.6 更改工作区的设置	25
2 安装	27
2.1 请先阅读以下内容以获取安全信息	27
2.2 尺寸和准则	28
2.3 连接硬件	29
2.4 安装 STEP 7 Basic	30
3 创建简单锁存电路	31
3.1 为用户程序创建项目	32
3.2 在用户程序中创建一个简单程序段	33
3.3 为指令输入变量和地址	35
3.4 组态 CPU	37
3.5 将用户程序下载到 CPU	40
3.6 测试示例用户程序的运行	41
3.7 回顾第一个练习中的任务	42

4	完成用户程序.....	43
4.1	添加接通延迟定时器，以延迟启动输出	43
4.2	接通 5 秒延迟的输出	46
4.3	编辑用户程序使其变为另一种结构	49
5	使用监视表格进行监视.....	53
5.1	创建监视表格.....	54
5.2	监视 CPU 中的数据值	55
5.3	在 LAD 编辑器中监视状态	56
5.4	将输入强制设置为特定值.....	56
5.5	转到在线.....	58
5.6	基于良好的开始	59
	索引.....	61

快速回顾

S7-1200 CPU 具有集成电源和各种板载输入与输出电路，是功能强大的控制器。下载程序后，CPU 监视输入并根据用户程序逻辑更改输出，用户程序逻辑可以包含布尔逻辑、计数器、定时器和复杂数学运算。



- ① 电源连接器
 - ② 可拆卸用户接线连接器（门后面）
 - ③ 板载 I/O 的状态 LED
 - ④ PROFINET 连接器（CPU 的底部）
- 为了与编程设备通信，CPU 提供了内置 PROFINET 端口。借助 PROFINET 网络，CPU 可以与 HMI 面板或其它 CPU 通信。

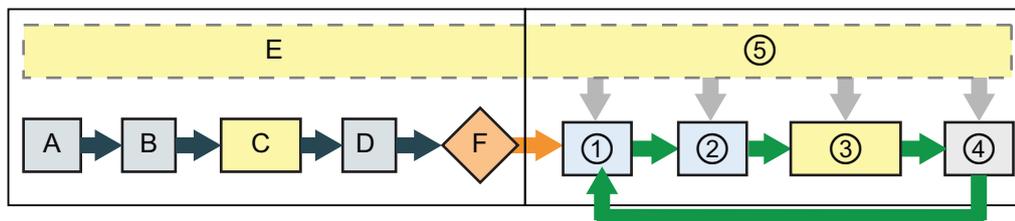
特征	CPU 1212C
物理尺寸	90 mm x 100 mm x 75 mm
<ul style="list-style-type: none"> • 工作存储器 • 装载存储器 • 保持性存储器 	<ul style="list-style-type: none"> • 25 KB • 1 MB • 2 KB
<ul style="list-style-type: none"> • 板载数字量 I/O • 板载模拟量 I/O 	<ul style="list-style-type: none"> • 8 路输入和 6 路输出 • 2 路输入
<ul style="list-style-type: none"> • 过程映像（输入） • 过程映像（输出） • 位存储器 (M) 	<ul style="list-style-type: none"> • 1024 个字节 • 1024 个字节 • 4096 个字节

1.1 有关 CPU 工作原理的基本信息

1.1.1 每个扫描周期均执行的任务

扫描周期使用户程序在某个给定的周期内执行时保持逻辑一致，避免物理输出点出现抖动，否则可能多次改变过程映像输出区中的状态。每个扫描周期都包括写入输出、读取输入、执行用户程序指令以及执行系统维护或后台处理。

在默认条件下，所有数字量和模拟量 I/O 点都使用被称作“过程映像”的内部存储区与扫描周期同步更新 I/O。过程映像包含 CPU、信号板和信号模块上的物理输入（“I 存储器”）和输出（“Q 存储器”）的快照。



- STARTUP 模式**
- A 清除过程映像的输入区（I 存储器）。
 - B 使用上一个值对输出进行初始化。
 - C 执行任意启动逻辑（包含在特殊代码块内）。
 - D 将物理输入的状态复制到 I 存储器。
 - E 所有中断事件都排队等候在 RUN 模式期间进行处理。
 - F 启用将过程映像的输出区（Q 存储器）写入到物理输出。
- RUN 模式**
- ① 将 Q 存储器写入物理输出。
 - ② 将物理输入的状态复制到 I 存储器。
 - ③ 执行用户程序逻辑。
 - ④ 执行自检诊断。
 - ⑤ 在扫描周期的任何时段处理中断和通信。

1.1.2 CPU 的操作模式

CPU 有以下三种操作模式：STOP 模式、STARTUP 模式和 RUN 模式。CPU 前面的状态 LED 指示当前操作模式。

- 在 STOP 模式下，CPU 不执行任何程序，用户可以下载项目。
- 在 STARTUP 模式下，CPU 会执行任何启动逻辑（如果存在）。在 STARTUP 模式下不处理任何中断事件。
- 在 RUN 模式下，重复执行扫描周期。在程序循环阶段的任何时刻都可能发生和处理中断事件。

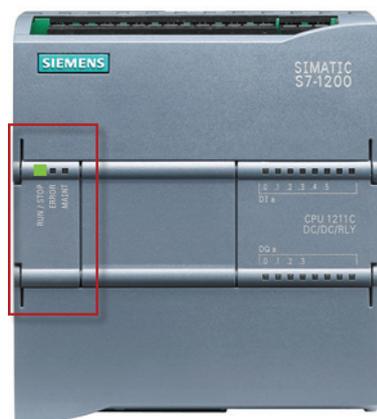
说明

CPU 处于 RUN 模式下时，无法下载任何项目。只有在 CPU 处于 STOP 模式时，才能下载项目。

该 CPU 没有用于更改操作模式（STOP 或 RUN）的物理开关。在设备配置中组态 CPU 时，应在 CPU 属性中组态启动行为。STEP 7 Basic 提供了用于更改在线 CPU 操作模式 (页 58) 的操作面板。



使用操作员面板上的按钮更改操作模式（STOP 或 RUN）。操作员面板还提供了用于复位存储器的 MRES 按钮。



RUN/STOP 指示灯的颜色表示 CPU 当前的操作模式：

- 黄色表示 STOP 模式。
- 绿色表示 RUN 模式。
- 闪烁表示 STARTUP 模式。

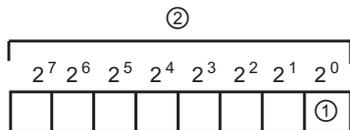
1.1.3 以位、字节、字等形式存储的数字数据



^{2⁰}
① 数字系统内的最小信息单位为“位”（对于“二进制数”）。一个位只能可以存储一种状态，即“0”（假或非真）或“1”（真）。

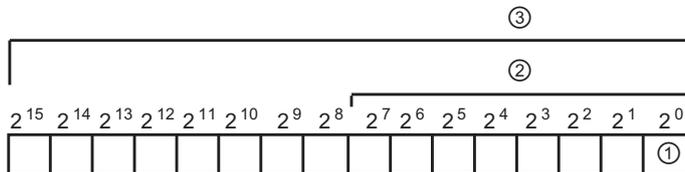
灯开关是只有两种状态的“二进制”系统示例。灯开关决定是“点亮”还是“熄灭”状态，并且该“值”可存储为一位。灯开关的数字值回答了以下问题：“灯是点亮的吗？”如果灯点亮（“真”），则该值为 1；如果灯熄灭（“假”），则该值为 0。

CPU 将数据位编成组。8 位 ① 一组称为一个字节 ②。



组中的每一位都通过具有自身地址的单独位置来精确定义。每一位都具有一个字节地址以及 0 到 7 的位地址。

2 个字节一组称为一个“字”。4 个字节一组称为一个“双字”。



- ③ 字
- ② 字节
- ① 位

使用二进制数字系统（基数为 2）来计数。字可以表示从 -32768 到 +32767 的整数。2¹⁵ 的位用来表示负数（当位置 2¹⁵ 的值为“1”时，该值为负数）。

说明

CPU 还支持 8 个字节一组构成“长实数”数据类型 (LReal)，用来存储极大或极精确的值。LReal 的范围是：+/-2.23 x 10⁻³⁰⁸ 到 +/-1.79 x 10³⁰⁸

1.1.4 S7-1200 支持的数据类型

数据类型不仅指定数据元素的大小，而且还指定数据内各个位的结构。

数据类型	大小	范围	常量输入示例
Bool (布尔型)	1 位	0 到 1	TRUE, FALSE, 0, 1
Byte (字节)	8 位 (1 个字节)	16#00 到 16#FF	16#12, 16#AB
Word (字)	16 位 (2 个字节)	16#0000 到 16#FFFF	16#ABCD, 16#0001
DWord (双字)	32 位 (4 个字节)	16#00000000 到 16#FFFFFFFF	16#02468ACE
Char (字符)	8 位 (1 个字节)	16#00 到 16#FF	'A', 't', '@'
SInt (短整型)	8 位 (1 个字节)	-128 到 127	123, -123
USInt (无符号短整型)	8 位 (1 个字节)	0 到 255	123
Int (整型)	16 位 (2 个字节)	-32,768 到 32,767	123, -123
UInt (无符号整型)	16 位 (2 个字节)	0 到 65,535	123
DInt (双整型)	32 位 (4 个字节)	-2,147,483,648 到 2,147,483,647	123, -123
UDInt (无符号双整型)	32 位 (4 个字节)	0 到 4,294,967,295	123

1.1 有关 CPU 工作原理的基本信息

数据类型	大小	范围	常量输入示例
Real (实型或浮点型)	32 位 (4 个字节)	+/-1.18 x 10 ⁻³⁸ 到 +/-3.40 x 10 ³⁸	123.456、-3.4、- 1.2E+12、 3.4E-3
LReal (长实型)	64 位 (8 个字节)	+/-2.23 x 10 ⁻³⁰⁸ 到 +/-1.79 x 10 ³⁰⁸	12345.123456789 -1.2E+40
Time (时间)	32 位 (4 个字节)	T#-24d_20h_31m_23s_648ms to T#24d_20h_31m_23s_647ms 存储为: -2,147,483,648 ms to +2,147,483,647 ms	T#5m_30s 5#-2d T#1d_2h_15m_30x_45m s
String (字符串)	可变	0 到 254 字节字符	'ABC'
DTL ¹ (长型日期和时间)	12 个字节	最小值: DTL#1970-01-01-00:00:00.0 最大值: DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16- 20:30:20.250

- ¹ DTL 数据类型是 12 个字节的结构，用于以预定义的结构保存有关日期和时间的信息。可以在块的临时存储器或者 DB 中定义 DTL。

1.1.5 S7-1200 的存储区

说明

使用符号地址代替绝对地址

STEP 7 Basic 简化了符号编程。用户为数据地址创建符号名称或“变量”，作为与存储器地址和 I/O 点相关的 PLC 变量或在代码块中使用的局部变量。要在用户程序中使用这些变量，只需输入指令参数的变量名称。为了更好地理解 CPU 构建存储区及其寻址的方式，以下段落将对 PLC 变量所引用的“绝对”寻址进行说明。

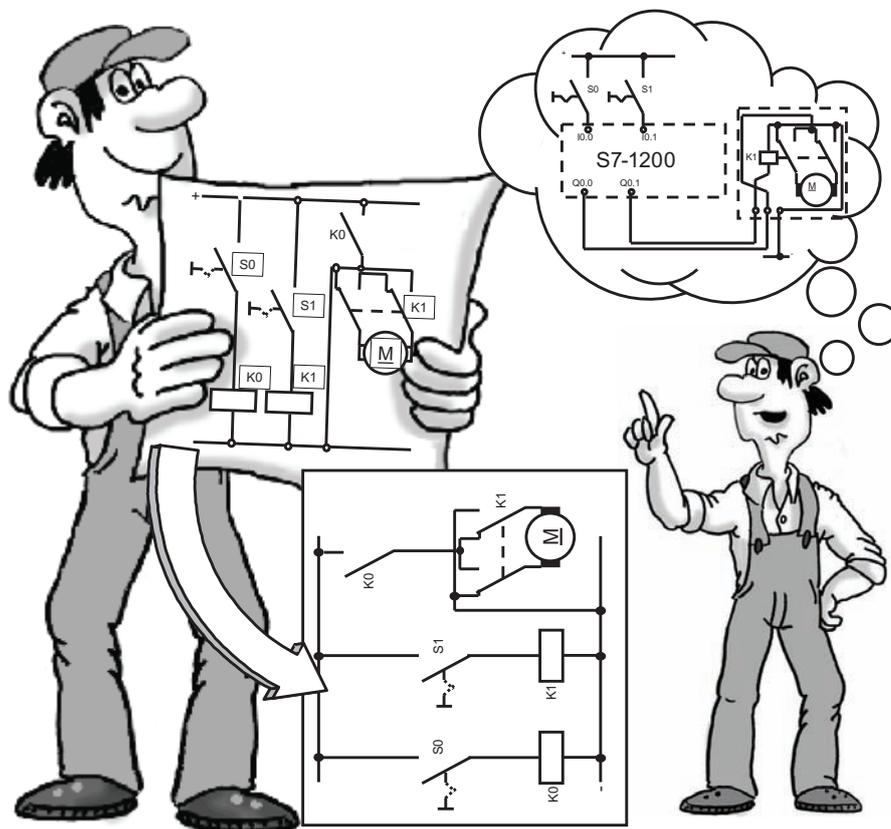
CPU 提供了各种专用存储区，其中包括输入 (I)、输出 (Q)、位存储器 (M)、数据块 (DB) 以及本地或临时存储器 (L)。用户程序对这些存储区中所存储的数据进行访问（读取和写入）。每个存储单元都有唯一的地址。用户程序使用这些地址访问存储单元中的信息。

存储区	说明
I 过程映像输入	CPU 在扫描周期开始时将物理输入的状态复制到 I 存储器。要立即访问或强制物理输入，请在地址或变量后面添加“:P”（例如，“Start:P”或 I0.3:P）。
Q 过程映像输出	CPU 在扫描周期开始时将 Q 存储器的状态复制到物理输出。要立即访问或强制物理输出，请在地址或变量后面添加“:P”（例如，“Stop:P”或 Q0.3:P）。
M 位存储器	用户程序读取和写入 M 存储器中所存储的数据。任何代码块均可访问 M 存储器。可以组态 M 存储器中的地址以在上电循环后保留数据值。
L “临时”存储器	只要调用代码块，CPU 就会分配要在执行块期间使用的临时或本地存储器 (L)。代码块执行完毕后，CPU 将重新分配本地存储器，以用于执行其它代码块。
DB 数据块	使用 DB 存储器存储各种类型的数据，其中包括操作的中间状态或 FB 的其它控制信息参数，以及许多指令（如定时器和计数器）所需的数据结构。可以指定数据块为读/写访问还是只读访问。可以按位、字节、字或双字访问数据块存储器。读/写数据块可以进行读访问和写访问。只读数据块只允许进行读访问。

不管使用变量（例如，“Start”或“Stop”）还是绝对地址（例如，“I0.3”或“Q1.7”），对输入 (I) 或输出 (Q) 存储区的引用都会访问过程映像而非物理输出。要立即访问或强制用户程序中的物理（外围设备）输入或输出，请在引用后面添加“:P”（例如，“Stop:P”或“Q0.3:P”）。

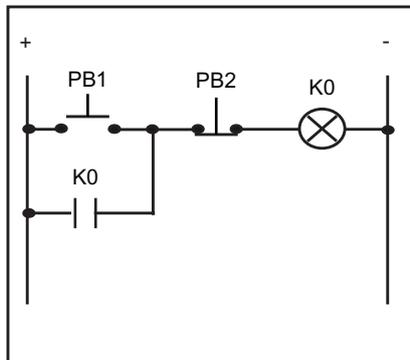
1.2 从示意图到用户程序

要将电路图转换为 PLC 程序，只需将电路图向左旋转 90°。随后电源线会显示在左边，而接地线显示在右边。电路的开关元件则显示在中间。



机器（例如，时间继电器或触发器）的电路逻辑以前是通过将开关、辅助接触器或控制接触器连接在一起来实现，而现在均由 PLC 来处理。输入端的控制元件（例如，输入开关或选择开关）和输出端的电源接触器（例如，电机接触器、极性换向器或阀）无法由 PLC 替换。

1.2.1 将示意图转换为 LAD 指令



让我们进一步看一下 PLC 程序在梯形图 (LAD) 中的结构。

这种表示法近似于电路图。

数字控制逻辑仅使用 0 和 1 来表示数据状态。状态“0”等同于“假”条件，而状态“1”等同于“真”条件。在电路中，电流决定二进制状态：有电流等于 1，无电流等于 0。在每次扫描开始时，CPU 会将输入的状态存储为 0（假）或 1（真）。



开关闭合允许电流通过。在这种情况下，电路状态等于“1”。
“1”=“真”= 有电流



开关断开不允许电流通过。在这种情况下，电路状态等于“0”。
“0”=“假”= 无电流

特定的 LAD 指令对应于示意图中的基本电路。



输入传感器扫描的**正逻辑**（“常开触点”）：

- 24V = 高电平 =“1”
- 0V = 低电平 =“0”



是否有电流流过？如果有，则扫描结果为“真”（或“1”）。



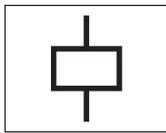
输入传感器扫描的**负逻辑**（“常闭触点”）：

- 0V = 低电平 = “1”
- 24V = 高电平 = “0”

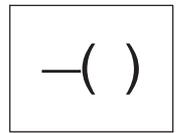


是否有电流流过？如果**无**电流流过物理触点，则扫描结果为“真”（或“1”）。信号流然后将流过 LAD 触点。

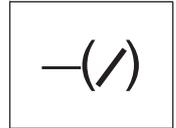
请牢记，触点闭合时**无**电流，因为这是“正常”情况。启动触点时，触点将打开并发出触点已激活的信号



输出线圈： 如果“真”值（电流或“信号流”）被传送到线圈，则 CPU 将通过接通线圈来激活线圈（设置为“1”）。



可以在程序段中间插入输出线圈，而不仅局限于程序段末尾。此外，CPU 还支持取反线圈，该线圈反映信号流反向流入线圈。



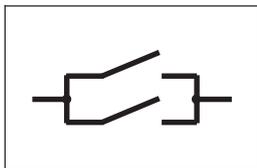
可组合多个电路来执行逻辑运算。例如，将两个触点串联可以创建两个输入的 AND 运算。



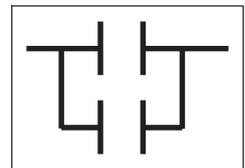
第一个开关和第二个开关必须闭合，才能通电。在 LAD 中，通过按序组合两个指令来表示 AND 运算。



对于 OR 逻辑，可将两个触点并联。



为了传递电流，必须闭合第一个开关“或”第二个开关。要创建并联电路（OR 逻辑），可在 LAD 程序段中创建分支。



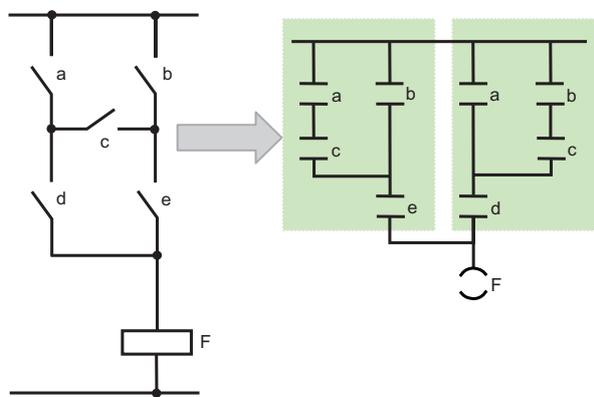
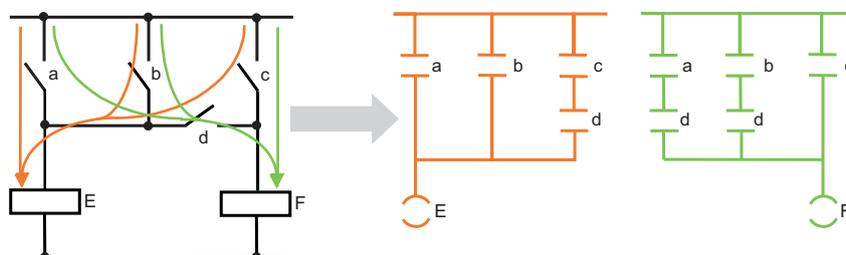
1.2.2 基本电路



如果要从接触器技术转换到 PLC 技术，您可能会遇到无法直接转换为梯形图表示的开关组合。

（为便于比较，在垂直方向上排列 LAD 程序段。）

桥接电路： 下图显示了一个简单的桥接电路（左侧）。您将使用右侧所示的两个 LAD 程序段来实现该简单桥接电路。每个程序段处理桥接电路中两个可能的电流通路之一。



按照此处所示，还可以实现复杂的桥接电路。

第一个程序段相当于“a-c-并联-b”通路。第二个程序段相当于“b-c-并联-a”通路。

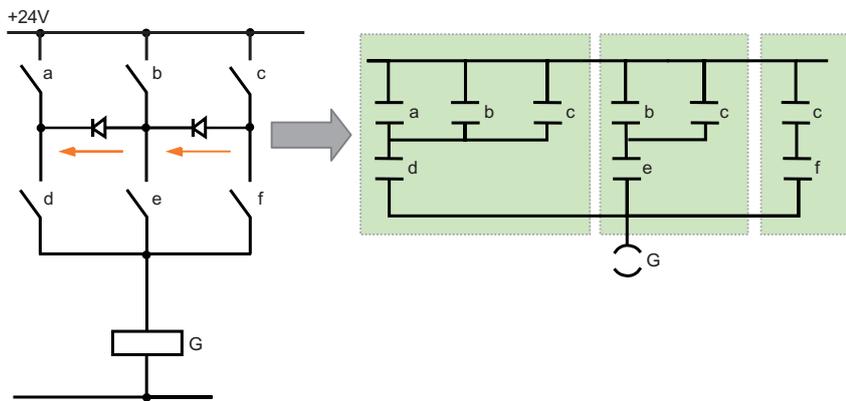
说明

在新项目中，尽可能避免在电路图中使用桥接电路！从一开始就想到“用 LAD 表示”！



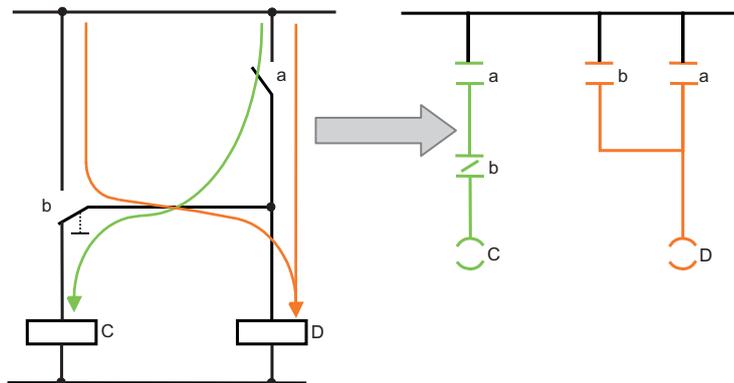
二极管电路： 当“旧”电路图中使用了二极管时，将二极管转换成梯形图项并不完全是件容易的事。因为二极管原则上表示连接线，但仅在一个方向上传输电流，因此此处采用了与桥接电路类似的解决方案。为便于与电路图进行比较，再次在垂直方向上排列梯形图。

例如，以下电路具有三条电流通路： 通过开关“d”、通过开关“e”或通过开关“f”。



通过二极管的电流只能从“b”流向“d”或从“c”流向“e”。用 LAD 表示此电路时，这三条电流通路将形成三个用框标识的子程序段。由于开关“d”、“e”和“f”连接在输出“G”所在的电源线上，因而这三个子程序段也连接成一个程序段。

转换开关： 在将电路图转换成 LAD 时，转换开关同样也不应该带来任何问题。通常，可以将一个转换开关转换成具有相同输入地址的一个常闭触点和一个常开触点。（在以下示例中，电流通路以高亮形式显示。）



转换开关 b 变成与“a”串联的一个常闭触点作用于输出“C”，或者变成与“a”并联的一个常开触点作用于“D”。

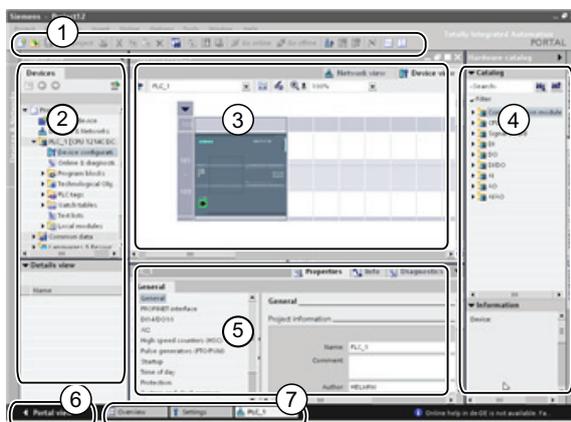
1.3 STEP 7 Basic 的基本功能

STEP 7 Basic 提供了一个用户友好的环境，供用户开发控制器逻辑、组态 HMI 可视化 and 设置网络通信。为帮助用户提高生产率，STEP 7 Basic 提供了两种不同的项目视图：根据工具功能组织的面向任务的门户集（门户视图），或项目中各元素组成的面向项目的视图（项目视图）。请选择能让您的工作最高效的视图。只需通过单击就可以切换门户视图和项目视图。



门户视图提供项目任务的功能视图，并根据要完成的任务来组织工具。用户可以很容易地确定如何继续以及选择哪个任务。

- ① 不同任务的门户
- ② 所选门户的任务
- ③ 所选操作的选择面板
- ④ 切换到项目视图



项目视图提供了访问项目中任意组件的途径。

- ① 菜单和工具栏
- ② 项目浏览器
- ③ 工作区
- ④ 任务卡
- ⑤ 巡视窗口
- ⑥ 切换到门户视图
- ⑦ 编辑器栏

由于这些组件组织在一个视图中，所以您可以方便地访问项目的各个方面。例如，巡视窗口显示了用户在工作区中所选对象的属性和信息。当用户选择不同的对象时，巡视窗口会显示用户可组态的属性。巡视窗口包含用户可用于查看诊断信息和其它消息的选项卡。

编辑器栏会显示所有打开的编辑器，从而帮助用户更快速和高效地工作。要在打开的编辑器之间切换，只需单击不同的编辑器。还可以将两个编辑器垂直或水平排列在一起显示。通过该功能可以在编辑器之间进行拖放操作。

1.3.1 在您需要时提供的帮助

为帮助用户获得更多信息或快速高效地解决问题，STEP 7 Basic 提供了智能的按需即点帮助。将光标悬停在软件界面的元素上会显示工具提示。界面中的某些工具提示通过“层叠”提供更多信息，甚至包括指向在线信息系统中特定主题的链接。工具提示旁的黑色三角形表示有更多信息。

只需单击即可获取帮助！从 Portal 视图，选择“起始”(Start) 并单击“帮助”(Help) 命令。从项目视图，在“帮助”(Help) 菜单中选择“显示帮助”(Show help) 命令。



该信息系统会在一个不会遮挡工作区域的窗口中打开。

要停坞帮助窗口并显示目录，请单击“显示/隐藏目录”(Show/hide contents) 按钮。随后可以调整帮助窗口的大小。

如果 STEP 7 Basic 已最大化，则单击“显示/隐藏目录”(Show/hide contents) 按钮将不会停坞帮助窗口。

单击“向下还原”(Restore down) 按钮可停坞帮助窗口。



随后可以移动和调整帮助窗口的大小。

1.3.2 从在线帮助中打印



要从信息系统中打印，请单击帮助窗口中的“打印”(Print)按钮。



通过“打印”(Print)对话框可以选择要打印的主题。请确保面板显示了主题。然后可以选择要打印的任何其它主题。单击“打印”(Print)按钮将所选主题发送到打印机。

1.3.3 插入指令

指令树中的拖放指令

STEP 7 Basic 提供了包含各种程序指令的任务卡。这些指令按功能分组。

要创建程序，可将指令从任务卡拖动到程序段中。



使用“收藏夹”工具栏插入常用指令



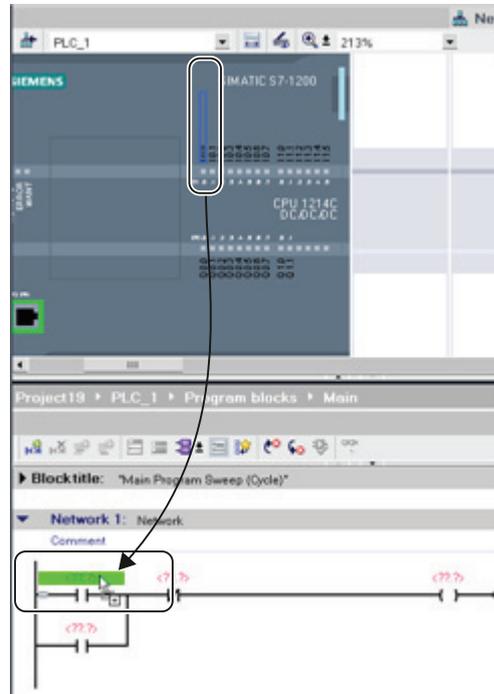
STEP 7 Basic 提供了“收藏夹”(Favorites) 工具栏，用户可通过该工具栏快速访问常用的指令。

STEP 7 Basic 在 LAD 编辑器和“指令”(Instructions) 任务卡中均显示“收藏夹”(Favorites)。只需单击指令的图标即可将其插入程序段！（在使用“指令”(Instructions) 任务卡中的“收藏夹”(Favorites) 时，也可双击。）

用户可以通过添加新指令方便地自定义“收藏夹”(Favorites)。只需将指令拖放到“收藏夹”(Favorites)。现在单击即可执行该指令！



1.3.4 在编辑器之间拖放



为帮助用户快速方便地执行任务，STEP 7 Basic 允许用户将元素从一个编辑器拖放到另一个编辑器中。例如，可以将 CPU 的输入拖动到用户程序中指令的地址上。（必须放大至少 200% 才能选中 CPU 的 I/O。）

请注意，变量名称不仅会在 PLC 变量表中显示，还会在 CPU 上显示。

要一次显示两个编辑器，请使用“拆分编辑器”(Split editor) 菜单命令或工具栏中的相应按钮。



要在已打开的编辑器之间切换，请单击编辑器栏中的图标。



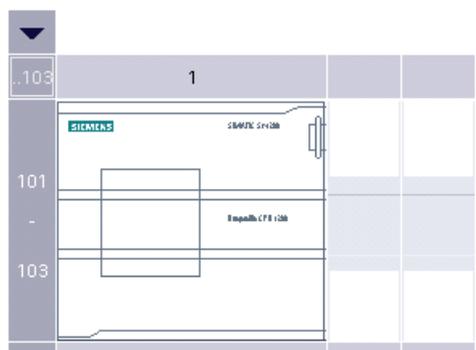
1.3.5 检测 CPU 的组态



如果用户有一个可连接到编程设备的物理 CPU，则很容易上传硬件配置。用户必须先将 CPU 连接到编程设备，并且必须创建一个新项目。创建未指定的 CPU 后，可以上传在线 CPU 的硬件配置。

在程序编辑器中，从“在线”(Online) 菜单中选择“硬件检测”(Hardware detection) 命令。

在设备配置编辑器中，选择用于检测所连设备配置的选项。



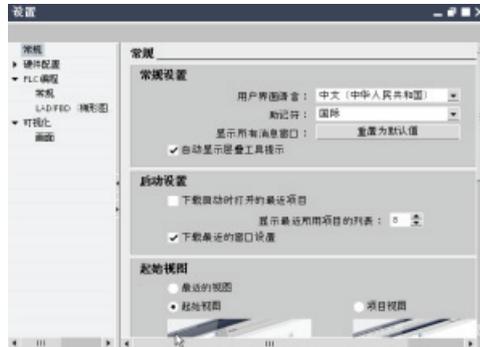
未指定该设备。

- 请使用 [硬件目录](#) 指定 CPU。
- 或 [检测](#) 相连设备的组态。

说明

从在线对话框中选择 CPU 之后，STEP 7 Basic 会上传 CPU 以及所有模块（SM、SB 或 CM）的组态。如果已为 CPU 分配 IP 地址，则会上传该 IP 地址。但是，不会上传其它设置（例如，模拟量 I/O 的属性）。必须转到“设备配置”(Device configuration) 手动组态 CPU 和各模块的设置。

1.3.6 更改工作区的设置



用户可以选择各种各样的设置，例如界面的外观、语言或保存工作的目录。

在“选项”(Options) 菜单中选择“设置”(Settings) 命令更改这些设置。

安装

2.1 请先阅读以下内容以获取安全信息

本《S7-1200 入门指南》(*Getting Started with S7-1200*) 提供了 S7-1200 的快速简介，但并不能代替《S7-1200 可编程控制器系统手册》(*S7-1200 Programmable Controller System Manual*)。在安装 CPU 1212C AC/DC/RLY 之前，请查看《S7-1200 可编程控制器系统手册》(*S7-1200 Programmable Controller System Manual*) 中所述的安全准则和其它信息。

说明

此手册中的信息描述了与 S7-1200 快速入门套件一同提供的 CPU 1212C AC/DC/RLY。有关任何其它 S7-1200 CPU 的信息，请参考《S7-1200 可编程控制器系统手册》(*S7-1200 Programmable Controller System Manual*)。

必须遵守如下基本准则，在安装或拆卸任何电气设备之前，请确保已关闭该设备的电源。同时，还要确保已关闭所有相关设备的电源。

警告

安装或拆卸已通电的 S7-1200 或相关设备可能会导致电击或意外设备操作。

如果在安装或拆卸过程中没有断开 S7-1200 或相关设备的所有电源，则可能会由于电击或意外设备操作而导致死亡、人员重伤和/或财产损失。

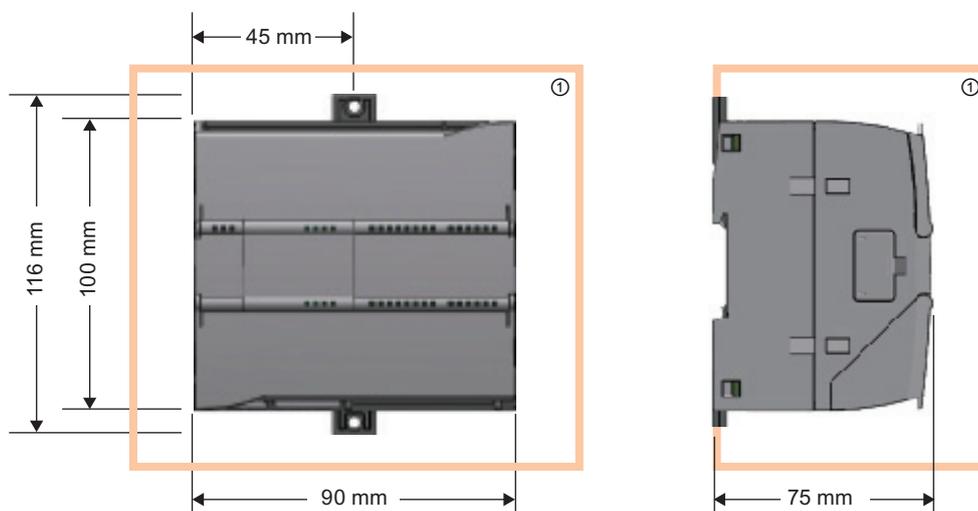
务必遵守适当的安全预防措施，确保在尝试安装或拆卸 S7-1200 CPU 或相关设备前断开 S7-1200 的电源。

警告

不要在危险场所 (Class I DIV 2 或 Class I Zone 2) 使用输入仿真器。

输入仿真器未获准在 Class I DIV 2 或 Class I Zone 2 危险场所使用。如果在 Class I DIV 2 或 Class I Zone 2 场所使用，开关存在潜在的打火危险/爆炸危险。

2.2 尺寸和准则



① 发热空隙： 25 mm



规划安装时，务必注意以下准则：

- 将设备与热辐射、高压和电噪声隔离开。
- 留出足够的空隙以便冷却和接线。必须在设备的上方和下方留出 25 mm 的发热区以便空气自由流通。

有关安装的具体要求和准则，请参考《S7-1200 可编程控制器系统手册》(S7-1200 Programmable Controller System Manual)。

警告

安装或拆卸已通电的 S7-1200 或相关设备可能会导致电击或意外设备操作。

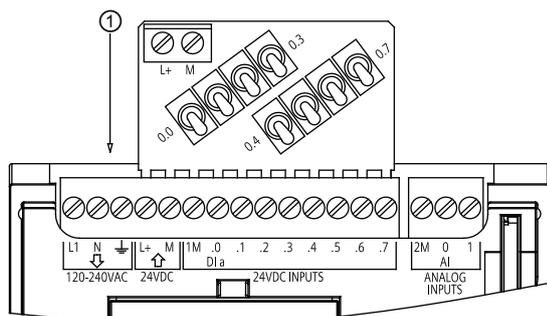
如果在安装或拆卸过程中没有断开 S7-1200 或相关设备的所有电源，则可能会由于电击或意外设备操作而导致死亡、人员重伤和/或财产损失。

务必遵守适当的安全预防措施，确保在尝试安装或拆卸 S7-1200 CPU 或相关设备前断开 S7-1200 的电源。

2.3 连接硬件

插入输入仿真器

将仿真器连接到 24 VDC 和输入端子。请确保仿真器上的所有开关均设为“关闭”位置。

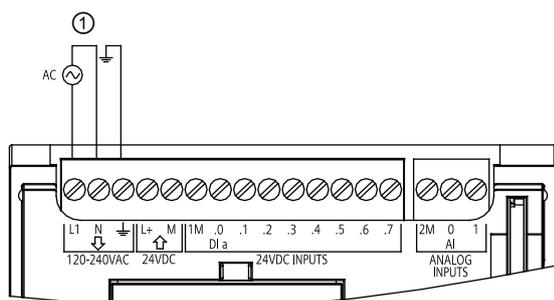


- ① 插入仿真器。
1. 松开端子上的螺钉。
 2. 插入仿真器。
 3. 拧紧端子上的所有螺钉。

将电源连接到 CPU

小心

S7-1200 快速入门套件提供了 CPU 1212C AC/DC/RLY，该 CPU 使用 120/240 VAC。要使用其它 CPU，请参考《S7-1200 可编程控制器系统手册》(S7-1200 Programmable Controller System Manual) 以获取有关如何将电源连接到特定 CPU 的说明。



- ① 将电源连接到 CPU。
- 该 CPU 需要使用 120 到 240 VAC 的电源。将线缆的电源线插入 L1 和 N 端，接地线插入接地端。拧紧端子螺钉后，即可将线缆插入电源插座。

连接 PROFINET 电缆

PROFINET 电缆是带有 RJ45 接口的标准 CAT5 以太网电缆，用于连接 CPU 与计算机或编程设备。



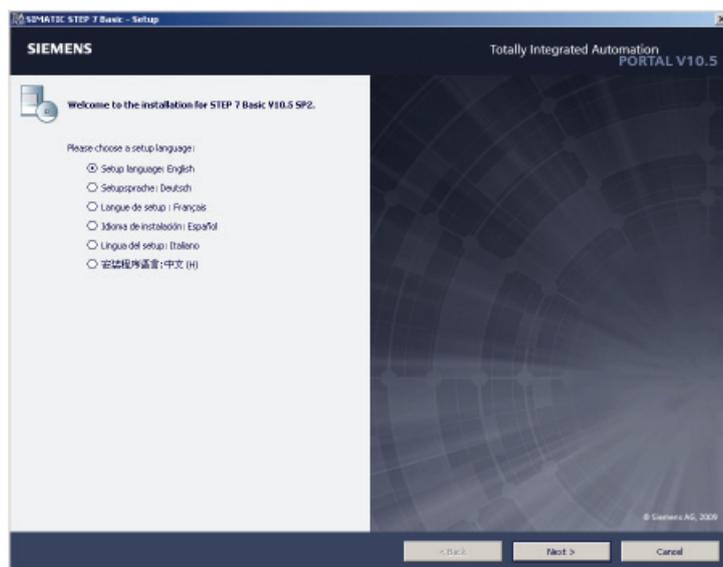
① 将 PROFINET 电缆的一端插入 CPU。

将电缆的另一端插入计算机或编程设备的以太网端口。

2.4 安装 STEP 7 Basic

STEP 7 Basic 的安装磁盘包含以下 SIMATIC 软件：

- 用于 S7-1200 CPU 的 STEP 7 Basic
- 用于 SIMATIC HMI 精简系列面板的 WinCC Flexible Basic
- 用于对 SIMATIC 软件产品进行授权的授权管理器

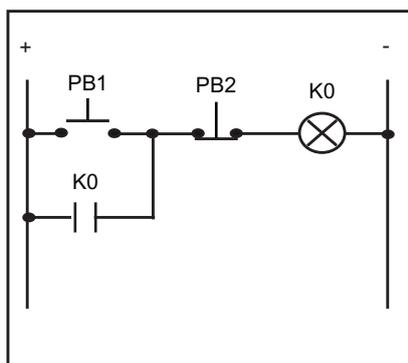


将磁盘插入计算机驱动器。安装程序应自动启动。（如果程序未自动启动，请参考安装磁盘的说明。）

选择安装对话框的语言，然后只需按照指示操作。

您可以选择以多种语言安装此软件。通常，仅安装您要使用的语言。但是，您也可以安装任何或所有其它语言。请注意，安装多种语言将需要更多磁盘空间。

创建简单锁存电路



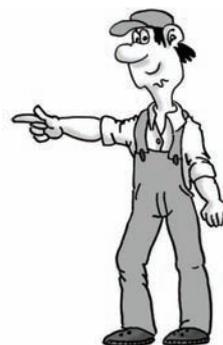
应熟悉以下标准锁存功能：

- 只要按钮输入 PB1 闭合 (ON)，输出 K0 就会激活 (ON)。
- 由于锁存电路使用 K0 的状态，因而 PB1 释放 (OFF) 后 K0 仍保持激活 (ON)。
- 按下按钮输入 PB2 将禁用 K0 (OFF)。
- K0 保持 OFF 状态，直到按钮输入 PB1 再次闭合 (ON)。

将 K0 用作 PB1 的并联输入可确保电路在 K0 断开前“锁存”为接通。

我们的第一个练习说明如何通过执行以下任务在 LAD 中创建锁存电路：

- 创建项目
- 插入 LAD 指令以创建小型用户程序
- 将 LAD 指令与 CPU 的 I/O 相关联
- 组态 CPU
- 将用户程序下载到 CPU
- 测试用户程序的运行



3.1 为用户程序创建项目



双击桌面图标启动 STEP 7 Basic。

STEP 7 Basic 打开后，单击“开始”(Start) 中的“创建新项目”(Create new project)。输入项目名称并单击“创建”(Create)。

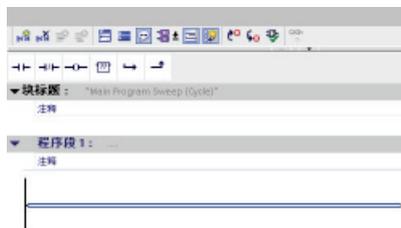


STEP 7 Basic 随即显示“新手上路”(First steps)。

单击“创建 PLC 程序”(Create a PLC program) 创建新用户程序。

STEP 7 Basic 将为用户程序创建“Main”代码块并打开“PLC 编程”(PLC Programming)。

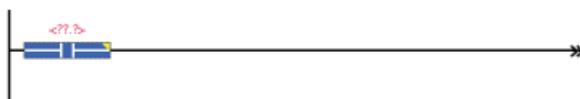
双击“主”块打开程序编辑器



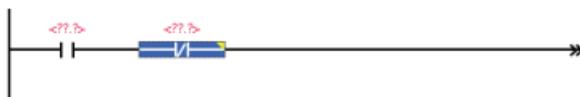
STEP 7 Basic 切换到“项目”视图并显示“主”块的程序段。

3.2 在用户程序中创建一个简单程序段

要启用锁存电路，可使用常开触点。常开触点在开关接通时提供信号流（电流）。在“收藏夹”(Favorites) 中单击常开触点以将触点插入程序段中。



要禁用锁存电路，可使用常闭触点。常闭触点在开关接通前提供信号流（电流）。接通常闭触点将中断信号流。



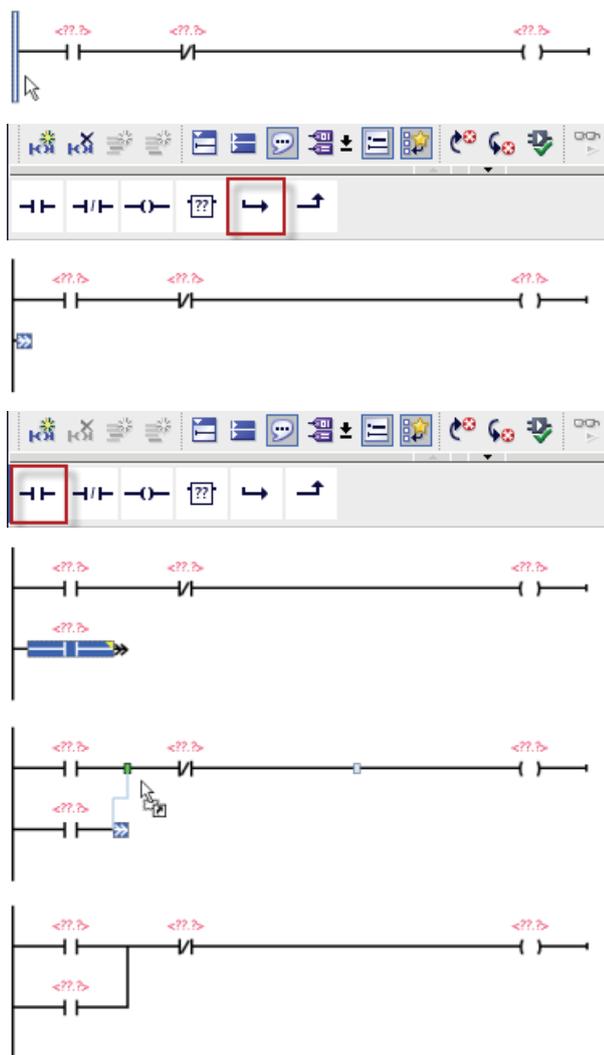
在“收藏夹”(Favorites) 中单击常闭触点以将触点插入程序段中。

信号流流过两个触点来为线圈通电。



单击线圈以将线圈插入程序段中。

3.2 在用户程序中创建一个简单程序段



要在“On”开关释放后将线圈“锁存”为通电，可创建一个并联分支。

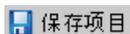
1. 选择程序段的电源线。
2. 在“收藏夹”(Favorites) 中单击“打开分支”(Open branch) 以从电源线打开分支。

在“收藏夹”(Favorites) 中单击常开触点以将触点插入分支中。

通过将分支末端拖动到程序段来闭合分支。

连接程序段中两个触点之间的分支可确保下列情况：

- 通过线圈的功率可在第一个开关释放（闭合）后流向线圈。
- 常闭触点可断开电路并使线圈断电。

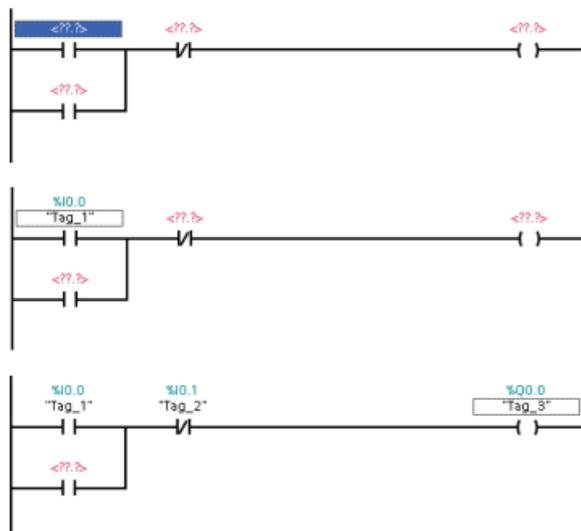


单击工具栏中的“保存项目”(Save project) 按钮保存工作。

现在，可以创建将用户程序指令与用户程序的输入和输出相关联的“变量”。



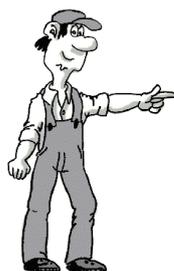
3.3 为指令输入变量和地址



下一步是将触点和线圈与 CPU 的输入和输出关联。为这些地址创建“PLC 变量”。

1. 选择第一个触点并双击操作数 (“<???.?>”)。
2. 输入地址“%I0.0”为该输入创建默认变量。
3. 为常闭触点输入地址“%I0.1”。
4. 为线圈输入一个输出地址 (“%Q0.0”)。

您可以轻松重命名 STEP 7 Basic 创建的默认变量名称。



只需右键单击指令（触点或线圈），然后从快捷菜单中选择“重命名变量”(Rename tag) 命令。



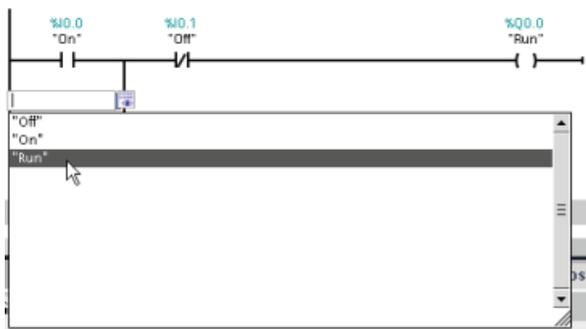
为三个指令输入以下名称：

- 将“Tag_1”(I0.0) 改为“On”。
- 将“Tag_2”(I0.1) 改为“Off”。
- 将“Tag_3”(Q0.0) 改为“Run”。

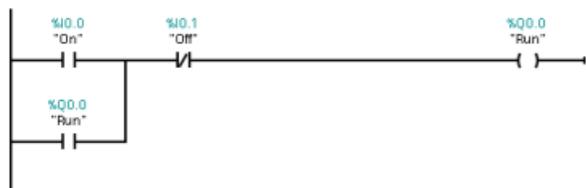


STEP 7 Basic 将变量保存在变量表中。可直接从变量表输入指令的变量地址。

3.3 为指令输入变量和地址



1. 选择分支中的触点。
2. 单击字段旁边的图标或输入“r”或“o”以显示变量表中的条目。
3. 从变量表的变量列表中选择“Run”。



现在，锁存电路已完成。



 保存项目

单击工具栏中的“保存项目”(Save project) 按钮保存工作。程序现在已准备就绪，可下载到 CPU。但是，必须先和设备配置中为该项目组态一个 CPU，之后才能下载用户程序。

3.4 组态 CPU

上传 CPU 的组态

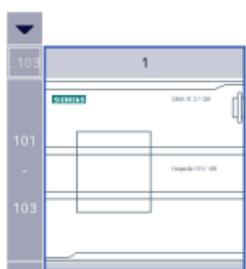


可以使用“设备组态”(Device Configuration) 的“检测 CPU”(Detect CPU) 功能上传 CPU 的硬件配置。

1. 在“项目树”(Project tree) 中，展开“PLC”容器。
2. 双击“设备组态”(Device configuration) 以显示 CPU。

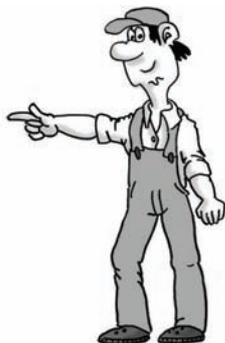
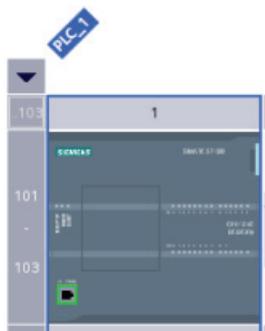
打开 LAD 编辑器时，STEP 7 Basic 已创建好“未指定的”CPU。

现在，可以在未指定的 CPU 上单击“检测”(Detect) 链接，以连接在线 CPU。



STEP 7 Basic 将“检测”与该计算机连接的所有 CPU。选择 CPU，然后单击“加载”(Load) 按钮将 CPU 组态加载到项目。

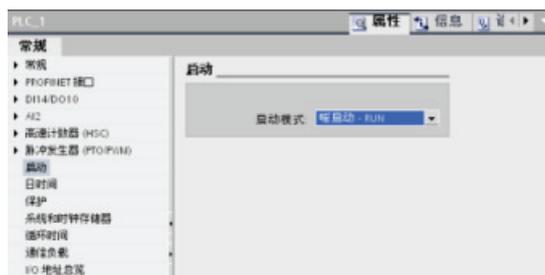
3.4 组态 CPU



STEP 7 Basic 将在设备组态中显示该 CPU。

组态 CPU 的属性

可以通过属性来组态 CPU 的工作参数。例如，加电循环后的默认启动设置可使 CPU 切换到 STOP 模式。



在本练习中，会更改启动属性：

1. 选择要在巡视窗口中显示其属性的 CPU。
2. 在巡视窗口中，选择“属性”(Properties) 选项卡，然后选择“启动”(Startup)，以显示 CPU 启动模式的选项。
3. 选择“暖启动 - RUN”(Warm restart - RUN)。

此时已将 CPU 组态为加电循环后始终执行暖启动并切换为 RUN 模式。

正如您在上传 CPU 组态时所见，CPU 并没有预分配的 IP 地址。您需要为每个 CPU 分配 IP 地址。



1. 选择 CPU 上的 PROFINET 端口，那么仅显示 PROFINET 接口的属性。（也可以在 CPU 的“常规”(General) 属性中选择“PROFINET 接口”(PROFINET interface)。）
2. 在巡视窗口中，选择“以太网地址”(Ethernet addresses)。

在“IP 协议”(IP protocol) 区域中会显示 STEP 7 Basic 创建的默认的 IP 地址。

本示例将采用默认 IP 地址。根据应用及安装的要求，可能需要为 CPU 组态具体的网络地址。请参阅系统手册以了解详细信息，也可咨询当地的网络专家。

将组态下载到 CPU

现在可以下载组态：

1. 选择 CPU。
2. 单击工具栏中的“下载”(Download) 按钮。

连接到 CPU 后，STEP 7 Basic 会显示“加载预览”(Load preview) 对话框。单击“加载”(Load)，将设备组态下载到 CPU 中。下载完成后，STEP 7 Basic 将显示“加载结果”(Load results) 对话框。单击“完成”(Finish)。

现在已将 CPU 组态为使用默认 IP 地址，并在加电循环后切换到 RUN 模式。此时可以下载用户程序。

在项目中保存您的工作



现在可以保存您的工作。只需单击工具栏中的“保存项目”(Save project) 按钮即可。

3.5 将用户程序下载到 CPU

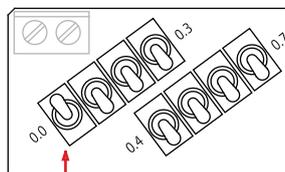
 现在可以下载用户程序。打开程序编辑器，然后单击“下载”(Download) 按钮即可。



连接到 CPU 后，STEP 7 Basic 会显示“加载预览”(Load preview) 对话框。单击“加载”(Load)，将用户程序下载到 CPU 中。在单击“完成”(Finish) 前，请先选择“全部启动”(Start all) 以确保将 CPU 切换到 RUN 模式。

3.6 测试示例用户程序的运行

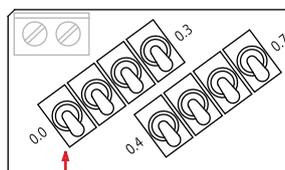
- ① 接通“On”开关 (I0.0)。
“Start”(I0.0) 和“Run”(Q0.0) 的状态 LED 将点亮。



①



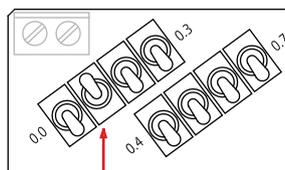
- ② 断开“On”开关 (I0.0)。
“On”(I0.0) 的状态 LED 将熄灭，但“Run”(Q0.0) 的状态 LED 仍保持点亮。



②



- ③ 接通“Off”开关 (I0.1)。
“Off”(I0.1) 的状态 LED 将点亮，同时“Run”(Q0.0) 的状态 LED 将熄灭。



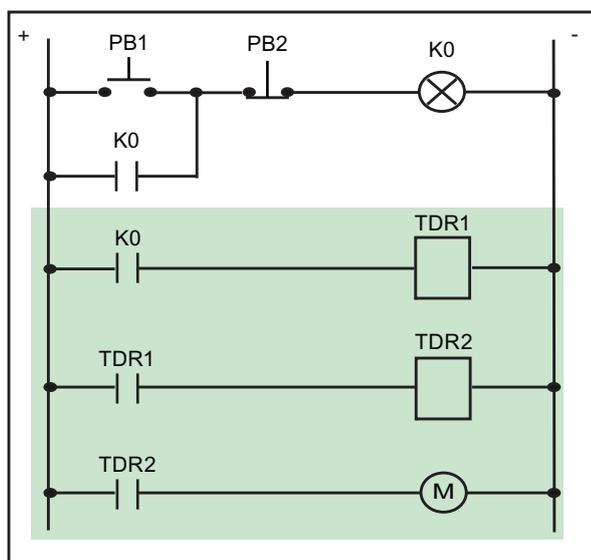
③



3.7 回顾第一个练习中的任务

恭喜您！通过执行以下任务，您已将一个简单的电路转换成 LAD 指令。

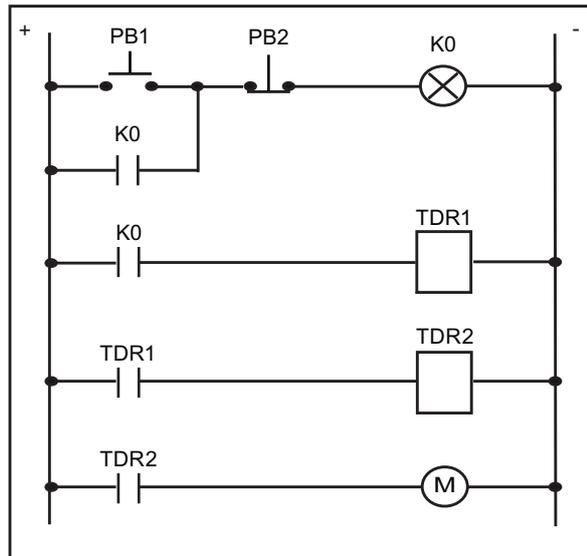
- 使用当前路径为这些指令创建逻辑流程
- 插入触点和线圈，以创建锁存电路
- 创建变量，用以链接这些指令和电路的输入与输出
- 将 CPU 的组态上传到项目中
- 下载并测试用户程序



在下面的练习中，将在此简单的用户程序中添加其它程序段。



完成用户程序



在接下来的练习中，将通过插入接通延迟定时器和关断延迟定时器，完成用户程序，以包含其余部分电路。

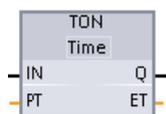


说明

S7-1200 不会限制用户程序中定时器或计数器的数量。每个定时器的数据都存储在数据块中，因此用户程序的大小只受 CPU 装载存储器容量的限制。

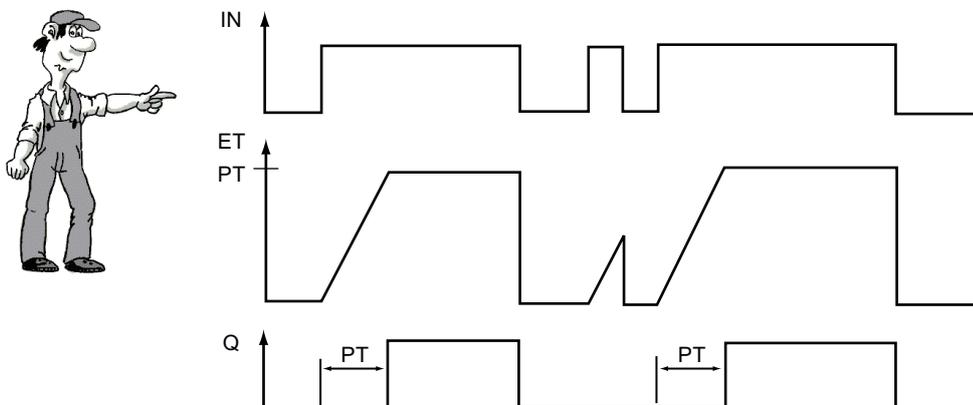
4.1 添加接通延迟定时器，以延迟启动输出

在本练习中，将通过插入一个接通延迟定时器以便在预设的时间段后接通输出，从而扩展锁存电路。



通过 TON 指令在预设延迟后接通输出 (Q)。定时器使用存储在数据块中的结构来保存定时器数据。在编辑器中设置定时器指令时即可分配该数据块。

4.1 添加接通延迟定时器，以延迟启动输出



定时器开启后 (IN = 1)，接通延迟定时器会等待定时器中预设好的时间 (PT) 后，再接通其输出 (Q = 1)。只要输入保持为接通状态 (IN = 1)，输出始终为接通状态 (Q = 1)。接通延迟定时器使用数据块 (DB, Data Block) 保存定时器的数据。在程序段中插入 TON 指令时即可分配该数据块。

在本练习中，为预设的时间参数 (PT) 输入一个常数。

在用户应用中，也可输入存储了预设值的 Time (4 字节有符号值) 存储器地址。这样，用户程序就可以在必要时根据操作条件更改预设值。还可在 Time (4 字节有符号值) 存储器地址中存储经过的时间 (ET)，用户程序中的其它元素也可对该地址进行访问。

插入接通延迟定时器

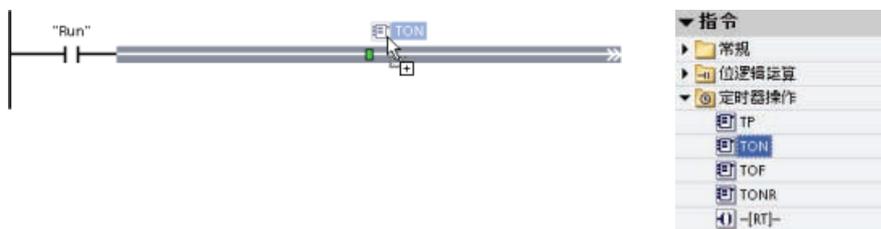
在用户程序中的第二个程序段使用 TON 指令，可在锁存电路的“Run”输出接通后开启 5 秒延迟。



- 首先，输入将激活该定时器的触点。
1. 选择用户程序中的第二个程序段。
 2. 同锁存电路 (页 33) 执行的操作一样，在“收藏夹”(Favorites) 中单击常开触点以插入指令。
 3. 对于指令地址，请选择“Run”变量。（就前一个练习 (页 35) 而言，可以键入“r”或单击变量图标以显示变量列表。）

4.1 添加接通延迟定时器，以延迟启动输出

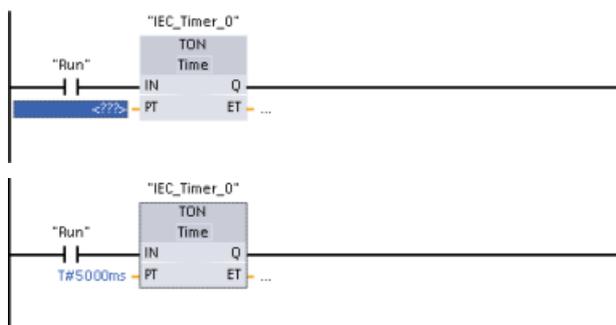
在“指令”(Instruction) 任务卡中，展开“定时器”(Timers) 文件夹，然后将 TON 定时器拖动到程序段中。



将 TON 指令拖放到程序段后，将自动创建一个用于存储定时器数据的单个背景数据块 (DB, Data Block)。

单击“确定”(OK) 创建 DB。

现在，创建一个 5 秒延迟。



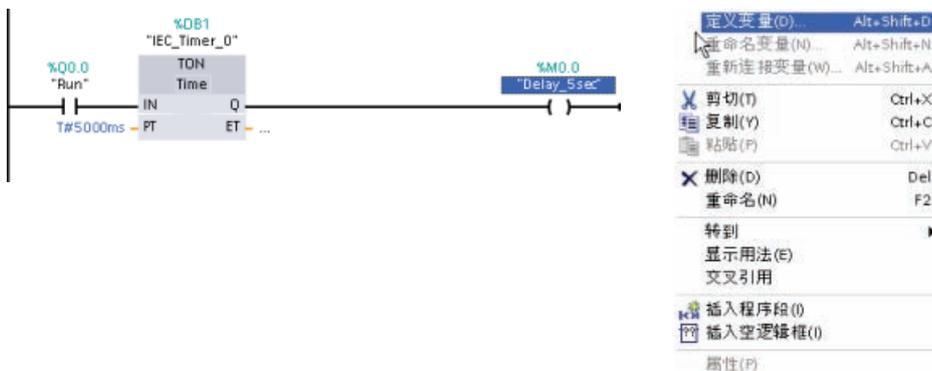
1. 双击预设时间 (PT, Preset Time) 参数。
2. 输入常数值“5000”（即 5000 ms 或 5 秒）。
也可以输入“5s”表示 5 秒。
（输入“5h”表示 5 小时，输入“5m”表示 5 分钟。）

STEP 7 Basic 将该常量格式化为“T#5000ms”

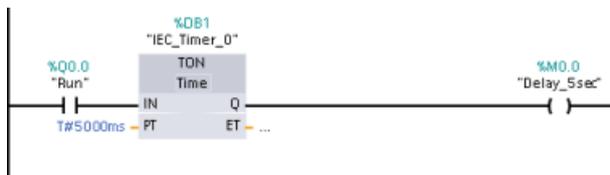
4.2 接通 5 秒延迟的输出

现在，插入一个将于 5 秒钟（TON 指令的预设值）后接通的线圈。

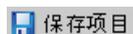
在本练习中，输入“M0.0”作为地址。然后将该值存储在位存储 (M) 区。并将该变量重新命名为“Delay_5sec”。



此时已创建了一个接通延迟定时器，将在 5 秒后接通“Delay_5sec”位。



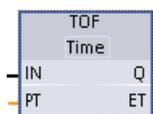
在项目中保存您的工作



现在可以保存您的工作。只需单击工具栏中的“保存项目”(Save project) 按钮即可。

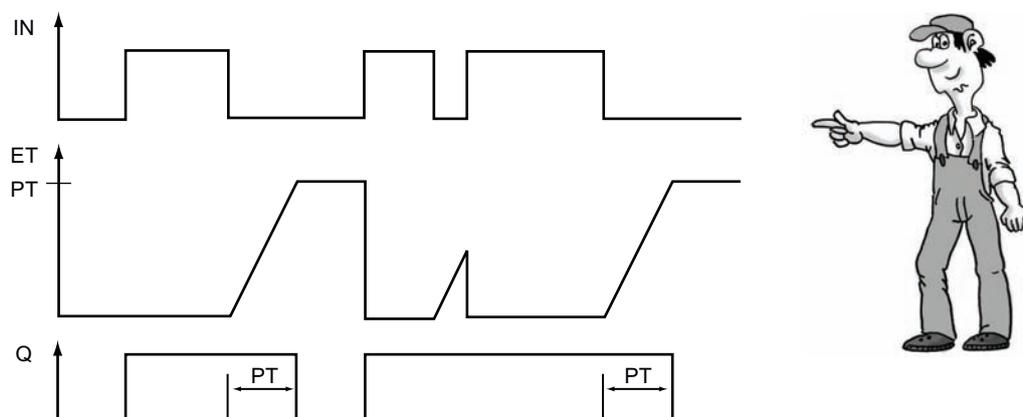
4.2 接通 5 秒延迟的输出

接下来，将关断延迟定时器 (TOF) 添加到用户程序的第三个程序段。



使用 TOF 指令，可使输出 (Q) 的接通状态在输入 (IN) 断开后保持一段预设的时间。TOF 指令由 IN 的上升沿启动后将接通 Q。TOF 定时器在 IN 的下降沿启动。经过了预设时间 (PT) 后，TOF 指令将断开 Q。

定时器开启后 (IN = 1)，将接通关断延迟定时器的输出 (Q = 1)。经过预设的时间段 (PT) 后，将断开关断延迟定时器的输出。



与接通延迟定时器类似，关断延迟定时器使用数据块 (DB, Data Block) 保存定时器数据。在程序段中插入 TOF 指令时即可分配该数据块。

与接通延迟定时器一样，为预设的时间参数 (PT) 输入一个常数。

也可输入存储预设值的 Time (4 字节有符号值) 存储器地址。这样，用户程序就可以在必要时根据操作条件更改预设值。还可以将经过的时间 (ET) 存储为可被用户程序中的其它元素访问的 Time (4 字节有符号值) 存储器地址。

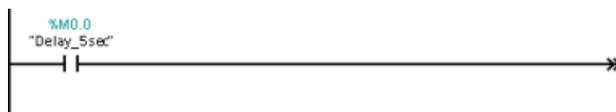
使用关断延迟定时器使输出的接通状态持续 5 秒

用户程序中的第三个程序段将使用 TOF 指令，以便在 TON 定时器 (页 43) 开启后启动并持续 5 秒钟。



可使用“Delay_5sec”变量激活关断延迟定时器。

1. 选择用户程序中的第三个程序段。
2. 在“收藏夹”(Favorites) 中单击常开触点以插入指令。
3. 选择“Delay_5sec”变量。(可以键入“d”显示变量列表，也可以单击变量图标来显示。)



“Run”输出接通后，“Delay_5sec”位将接通 5 秒钟。

4.2 接通 5 秒延迟的输出

要插入 TOF 定时器，请在“指令”(Instruction) 任务卡中双击 TOF 定时器。同 TON 定时器一样，STEP 7 Basic 将为定时器数据创建一个 DB。（单击“确定”(OK) 创建定时器并插入 TOF 指令。）

在 PT 参数中输入 5000（即 5000 ms 或 5 秒）作为 TOF 指令的预设值。



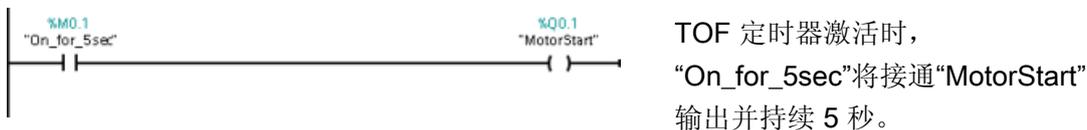
现在，创建一个要在激活 TOF 指令时接通的线圈。然后输入“M0.1”作为地址。并将该变量重命名为“On_for_5sec”。



TON 定时器接通“Delay_5sec”时，TOF 定时器将接通“On_for_5sec”位并持续 5 秒。

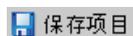
要完成用户程序，需要用两个元素创建一个程序段：

- 插入常开触点。对于地址，选择变量“On_for_5sec”。
- 插入线圈。输入“Q0.1”作为地址。将该变量重命名为“MotorStart”。



TOF 定时器激活时，“On_for_5sec”将接通“MotorStart”输出并持续 5 秒。

在项目中保存您的工作



现在可以保存您的工作。只需单击工具栏中的“保存项目”(Save project) 按钮即可。

下载和测试定时器指令

 现在可以下载用户程序。打开程序编辑器，然后单击“下载”(Download) 按钮即可。



STEP 7 Basic 在连接到 CPU 后便显示“加载预览”(Load preview) 对话框。单击“加载”(Load)，将用户程序下载到 CPU 中。在单击“完成”(Finish) 前，选择“全部启动”(Start all) 以确保 CPU 切换到 RUN 模式。

现在已准备就绪，可进行测试用户程序：

1. 确保所有输入均断开。
2. 接通输入 I0.0。
 - 5 秒后，将点亮输出 Q0.1 的状态 LED 灯。
3. 断开输入 I0.0，并接通输入 I0.1。
 - 5 秒后，将熄灭输出 Q0.1 的状态 LED 灯。

4.3 编辑用户程序使其变为另一种结构

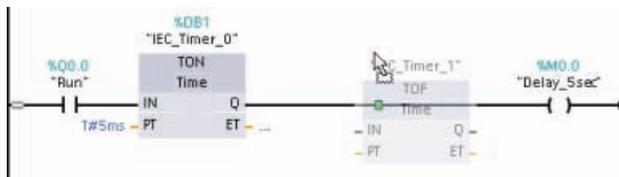


创建的程序段与电路图相一致。这有助于他人了解用户程序的逻辑结构以便进行其它操作，例如，故障排除或后续修改。

另外，您还可以将定时器和输出“MotorStart”合并到一个程序段中。接下来的练习将说明编辑 LAD 程序是一件很容易的事。

4.3 编辑用户程序使其变为另一种结构

首先，从程序段 3 中拖出 TOF 定时器，然后将其放入程序段 2 中。



现在 TON 定时器的输出与 TOF 定时器的输入直接相连。



启动 TOF 定时器不再需要“Delay_5sec”输出（M 存储器中的某一位）。只需重命名“Delay_5sec”输出即可，而无需删除该输出。

1. 双击“Delay_5sec”变量，然后单击“变量”(Tag) 图标以显示变量列表。
2. 从变量列表中选择“MotorStart”变量。

现在，已将程序段 2、3、4 的逻辑结构合并到一个程序段中。

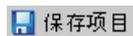


程序段 3:

现在，删除程序段 3 和程序段 4:

1. 选择程序段标题后，单击右键以显示上下文菜单。
2. 从上下文菜单中选择“删除”(Delete) 命令。

在项目中保存您的工作



现在可以保存您的工作。只需单击工具栏中的“保存项目”(Save project) 按钮即可。

 保存已修改的用户程序后，单击“下载”(Download) 按钮。



在“加载预览”(Load preview) 对话框中，单击“加载”(Load) 将用户程序下载到 CPU 中。

选择“全部启动”(Start all) 确保 CPU 切换到 RUN 模式，然后单击“完成”(Finish)。

回顾关于定时器的练习

在本练习中，我们通过添加三个程序段完成用户程序。此外还学习了如何通过拖放操作进行元素编辑。



在接下来的练习中，我们将学习如何通过监视表格来监视用户程序的运行。

4.3 编辑用户程序使其变为另一种结构

使用监视表格进行监视



测试锁存电路(页 41)时, 使用了仿真器上的物理开关, 并监视 CPU 前面板上指示灯的亮灭情况。

在本练习中, 将使用 STEP 7 Basic 的在线功能监视用户程序的运行情况。

什么是“监视表格”?

在 CPU 执行用户程序时, 用户可以通过监视表格监视或修改变量值。

	名称	地址	显示格式	监视值	修改值
1	"On"	%I0.0	布尔型	<input type="checkbox"/> FALSE	
2	"Off"	%I0.1	布尔型	<input type="checkbox"/> FALSE	
3	"Run"	%Q0.0	布尔型	<input type="checkbox"/> FALSE	



使用“修改”(Modify) 功能可以更改变量的值。但是, “修改”(Modify) 功能对输入 (I) 或输出 (Q) 不起作用, 这是因为 CPU 会更新 I/O, 并在读取已修改的值之前覆盖所有的已修改值。

监视表格提供了可用于修改 I/O 值的“强制”(Force) 功能。在本练习中, 我们将学习在接通锁存电路中如何强制输入。

5.1 创建监视表格

5.1 创建监视表格



要创建监视表格：

1. 展开“监视表格”(Watch tables) 容器。
2. 双击“添加新监视表格”(Add new watch table) 打开一个新的监视表格。

在“名称”(Name) 字段中单击，并输入变量。键入一个字符，并从列表中选择变量。



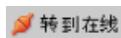
创建监视表格后，可以转到在线状态，以监视用户程序的执行。



5.2 监视 CPU 中的数据值



要监视这些变量，必须在线连接到 CPU。只需单击工具栏中的“转到在线”(Go online) 按钮。



连接到 CPU 后，STEP 7 Basic 将工作区的标题变为橙色。

项目树显示离线项目和在线 CPU 的比较结果。绿色圆点表示 CPU 与项目同步，即二者都具有相同的组态和用户程序。

监视表格中将显示各变量。

	名称	地址	显示格式	监视值	修改值
1	"On"	%I0.0	布尔型		
2	"Off"	%I0.1	布尔型		
3	"Run"	%Q0.0	布尔型		



要监视用户程序的执行并显示变量的值，请单击工具栏中的“全部监视”(Monitor all) 按钮。“监视值”(Monitor value) 字段中将显示每个变量的值。

	名称	地址	显示格式	监视值	修改值
1	"On"	%I0.0	布尔型	FALSE	
2	"Off"	%I0.1	布尔型	FALSE	
3	"Run"	%Q0.0	布尔型	FALSE	

5.3 在 LAD 编辑器中监视状态

还可以在 LAD 编辑器中监视各变量的状态。使用编辑器栏显示 LAD 编辑器。使用编辑器栏，可以在打开的编辑器之间切换视图，而无需打开或关闭编辑器。

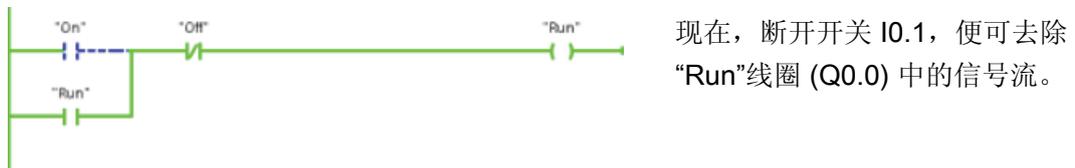


 在 LAD 编辑器的工具栏中，单击“接通/断开监视”(Monitoring on/off) 按钮，以显示用户程序的状态。



当仿真器上的所有开关都断开时，请注意输入“On”不是绿色，这是因为它也是断开的（或为“假”）。另外请注意，也没有流向“Off”触点的信号流。然而，常闭触点“Off”本身却为绿色。“Off”为绿色表示，其本身并不是产生信号流，而是表示如果有信号流入“Off”触点，那么信号流将通过“Run”线圈。

使用仿真器，接通 I0.0 的开关，并监视整个程序段中的信号流。现在，断开 I0.0，并查看锁存电路的工作方式！



5.4 将输入强制设置为特定值

监视表格提供了“强制”功能，能够将与外围设备输入或外围设备输出地址对应的输入或输出点的值改写成特定的值。CPU 在执行用户程序前将此强制值应用到输入过程映像，并在将输出写入到模块前将其应用到输出过程映像中。



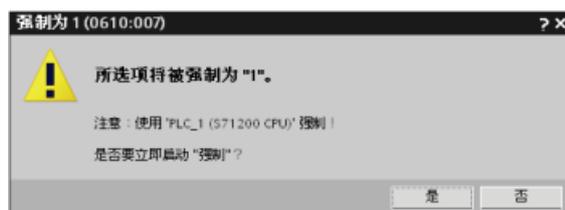
由于无法强制输入（或“I”地址），因而必须更改输入“On”的地址以访问外围设备输入。在“On”的“地址”(Address) 或“名称”(Name) 单元格中，在地址中添加“:P” (“On:P”)。

	名称	地址	显示格式	监视值
1	"On" P	%I0.0 P	布尔型	<input type="checkbox"/> FALSE
2	"Off"	%I0.1	布尔型	<input type="checkbox"/> FALSE
3	"Run"	%Q0.0	布尔型	<input type="checkbox"/> FALSE

 现在，单击“显示/隐藏强制列”(Show/hide force columns) 按钮，显示“强制值”(Force value) 列。

	名称	地址	显示格式	监视值	强制值	F
1	"On" P	%I0.0 P	布尔型	FALSE		
2	"Off"	%I0.1	布尔型	FALSE		
3	"Run"	%Q0.0	布尔型	FALSE		

右键单击“强制值”(Force value) 单元格以显示上下文菜单。然后选择“强制为 1”(Force to 1) 命令，将“On:P”(I0.0) 设置为 1 或“真”。



由于强制值存储在 CPU 中而不是监视表格中，因而 STEP 7 Basic 会要求确认是否将强制值设置为 1。单击“是”(Yes) 进行确认。

 使用“启动或替换强制”(Start or replace forcing) 按钮将“On:P”输入的值强制设置为 1 (或“真”)。单击“停止强制”(Stop forcing) 按钮将“On:P”的值重置为 0 (或“假”)。

	名称	地址	显示格式	监视值	强制值	F
1	"On" P	%I0.0 P	布尔型	TRUE	TRUE	<input checked="" type="checkbox"/>
2	"Off"	%I0.1	布尔型	FALSE		
3	"Run"	%Q0.0	布尔型	TRUE		

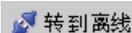
请注意，当仿真器上的所有开关都断开时，即使“On:P”为 0 (“假”)，“Run”输出仍为“真”(1 或接通)。

说明

在监视表格中强制了输入或输出时，强制操作将变成用户程序的一部分。如果关闭 STEP 7 Basic，在清除那些被强制的元素之前，它们相对于 CPU 正在执行的用户程序而言仍为激活状态。要清除这些被强制的元素，必须使用 STEP 7 Basic 连接到在线 CPU，然后使用监视表格断开或停止对这些元素的强制功能。

5.5 转到在线

现在，单击“转到离线”(Go offline) 按钮以断开与 CPU 的连接。



5.5 转到在线

最后的这些信息与 CPU 操作面板有关。请记住，CPU 并不提供从 STOP 模式切换到 RUN 模式的物理开关。需使用 STEP 7 Basic 更改 CPU 的"操作模式。

首先，访问在线 CPU。



1. 展开“在线访问”(Online access) 文件夹，然后展开连接到 CPU 的网络适配器。
2. 双击“更新可访问的节点”(Update accessible nodes) 以查找 CPU。
3. STEP 7 Basic 显示 CPU 后，展开该 CPU。
4. 双击“在线和诊断”(Online & diagnostics) 以显示在线工具。

“在线工具”(Online tools) 任务卡中包含有可用于监视 CPU 性能的大量工具，例如，诊断缓冲区、扫描时间和存储器使用情况的测量以及 CPU 操作面板。



使用 CPU 操作面板可更改"操作模式。单击“RUN”或“STOP”按钮可更改 CPU 的"操作模式。

请注意！单击“MRES”按钮将复位存储器并删除 CPU 的工作存储器！（虽然 MRES 不会删除用户程序，但会将非保持性存储区设置为 CPU 的默认组态。）

5.6 基于良好的开始

恭喜您！您现已能成功使用 S7-1200 CPU 和 STEP 7 Basic 编程软件！



您现在已掌握了创建项目的基本技能,但仍需进一步学习。

可以参考《S7-1200 简明手册》(*S7-1200 Easy Book*)和《S7-1200 可编程控制器系统手册》(*S7-1200 Programmable Controller System Manual*)。这两本手册介绍了 S7-1200 功能的更多信息。

关于常见问题解答和其它各类相关文档,可以访问客户支持网站(<http://www.siemens.com/automation/support-request>)。

5.6 基于良好的开始

索引

"

"操作模式, 58

A

AND 逻辑, 15, 33

C

CPU

"操作模式, 58

M 存储器, 7

MRES, 58

位存储器 (M), 7

保存, 34, 36, 39, 46, 48, 50

保持性存储器, 7

在线, 55

安全准则, 27, 28

工作存储器, 7

强制, 56

操作员面板, 9

操作模式, 9

操作面板, 58

支持的定时器和计数器数量, 43

数据类型, 11

未指定的 CPU, 24, 37

概述, 7

物理尺寸, 7, 28

监视, 55

程序执行, 8

装载存储器, 7

输入和输出, 7

过程映像, 7

连接电源, 29

I

I 存储器, 12

变量, 35

寻址外围设备输入, 56

强制, 56

更改变量, 49

监视, 55

监视表格, 55, 56

输入地址, 35

L

LAD

从电路图转换, 14, 15

变量, 35

插入指令, 33, 44, 47

新分支, 33

状态, 55, 56

监视, 55

编辑指令, 49

LED, 7

M

M 存储器, 7

变量, 44

更改变量, 49
输入地址, 44

O

OR 逻辑, 15, 33

P

PLC

M 存储器, 7
MRES, 58
位存储器 (M), 7
保存, 34, 36, 39, 46, 48, 50
保持性存储器, 7
工作存储器, 7
强制, 56
支持的定时器和计数器数量, 43
数据类型, 11
未指定的 CPU, 24, 37
概述, 7
物理尺寸, 7, 28
监视, 55
程序执行, 8
装载存储器, 7
输入和输出, 7
过程映像, 7

PROFINET, 30

Q

Q 存储器, 12
变量, 35
更改变量, 49

R

RUN 模式, 9, 58
强制, 56
程序执行, 8

S

S7-1200

MRES, 58
保存, 34, 36, 39, 46, 48, 50
安全准则, 27, 28
强制, 56
操作面板, 58
物理尺寸, 28
监视, 55

S7-1200 支持的定时器和计数器数量, 43

STARTUP 模式

强制, 56
程序执行, 8

STEP 7

MRES, 58
保存, 34, 36, 39, 46, 48, 50
启动, 30
安装, 30
强制, 56
操作员面板, 9
操作面板, 58
数据类型, 11
检测 CPU, 24, 37
监视, 55
监视表格, 54
设置, 25
语言, 25
门户视图, 19
项目视图, 19

STEP 7 的外观, 25

STEP 7 的设置, 25

STOP 模式, 9, 58

强制, 56

T

TIA Portal

门户视图, 19

项目视图, 19

上

上传, 37

以

以太网通信, 30

仿

仿真器

安装, 29

测试, 41

位

位存储器 (M), 7

保

保存, 34, 36, 39, 46, 48, 50

保存项目, 33

保持性存储器, 7

信

信息系统, 20

停坞, 20

打印, 21

显示目录和索引, 20

关

关断延迟 (TOF) 指令, 46

删

删除程序段, 49

发

发现 CPU, 24, 37

变

变量

I 存储器, 35

M 存储器, 44

变量表, 35, 49

强制, 56

更改地址, 49

状态, 55, 56

监视, 55

启

启动 STEP 7 Basic, 30

在

在线

MRES, 58

强制, 56

操作员面板, 9

操作面板, 58

- 监视表格, 55
- 连接, 58
- 在线帮助, 20
 - 停坞, 20
 - 打印, 21
 - 扩展帮助窗口, 20
 - 显示目录和索引, 20

基

- 基本电路
 - 二极管电路, 17
 - 桥接电路, 17
 - 转换开关, 17

存

- 存储区, 7
 - I 存储器, 12
 - M 存储器, 12
 - Q 存储器, 12
 - 外围设备 I/O, 12
 - 寻址外围设备输入, 56
 - 过程映像, 12
- 存储器复位 (MRES), 9, 58

安

- 安全准则, 27, 28
- 安装, 28, 29, 30
 - 安全准则, 27, 28
- 安装准则, 27, 28

定

- 定时器
 - TOF (关断延迟), 46
 - TON (接通延迟), 43

- 支持的定时器和计数器数量, 43

尺

- 尺寸, 28

工

- 工作存储器, 7

帮

- 帮助, 20
 - 停坞, 20
 - 打印, 21
 - 扩展, 20
 - 显示目录和索引, 20

强

- 强制, 56
 - I 存储器, 56
 - 外围设备输入, 56

打

- 打印帮助主题, 21

扫

- 扫描周期, 8
 - 强制, 56

拖

- 拖放
 - LAD 程序段之间, 49
 - 从指令任务卡, 44

在编辑器之间, 23

指令, 22

指

指令

TOF (关断延迟定时器), 46

TON (接通延迟定时器), 43

在编辑器之间拖放, 23

强制, 56

指令任务卡, 47

插入, 22, 33

收藏夹, 22, 33

新分支, 33

更改指令地址, 49

状态, 55, 56

监视, 55

线圈, 33

编辑程序段, 49

触点, 33

接

接通延迟 (TON) 指令, 43

操

操作模式, 8, 9

操作面板, 9, 58

收

收藏夹工具栏, 22, 33

数

数据类型, 11

文

文档, 20

显

显示目录和索引 (在线帮助), 20

更

更改 STEP 7 的设置, 25

更改变量地址, 49

未

未指定的 CPU, 24, 37

检

检测 CPU, 24, 37

物

物理尺寸, 28

状

状态 LED, 7

用

用户界面

门户视图, 19

项目视图, 19

电

电路

二极管电路, 17

桥接电路, 17

- 转换开关, 17
- 电路图
 - 二极管电路, 17
 - 桥接电路, 17
 - 转换为 LAD, 14, 15
 - 转换开关, 17

监

- 监视
 - LAD 状态, 55
 - 创建监视表格, 54
 - 强制, 56
 - 监视表格, 54, 55, 56
- 监视表格
 - 创建, 54
 - 强制, 53, 56
 - 监视, 53, 55

程

- 程序段
 - 删除程序段, 49
 - 添加分支, 33
- 程序段的分支, 33

编

- 编程
 - 删除程序段, 49
 - 在编辑器之间拖放, 23
 - 收藏夹, 22
 - 程序执行, 8
 - 编辑, 49

装

- 装载存储器, 7

- MRES, 58

转

- 转换电路图, 14, 15

输

- 输入仿真器
 - 安装, 29
 - 测试, 41
- 输入和输出, 7, 12
 - 强制, 56
 - 扫描周期, 8
 - 监视, 55

过

- 过程映像, 7, 12
 - 强制, 56
 - 状态, 55, 56
 - 监视, 55

锁

- 锁存电路, 33

门

- 门户视图, 19

项

- 项目视图, 19

