

# SIEMENS

## SINUMERIK 802S/C/D

### PLC 子程序库应用指南

技术手册

2004.03 版本

制造商文件

# SINUMERIK® 文献

## 版本说明

以下是当前版本及以前各版本的简要说明。

每个版本的状态由“附注”栏中的代码指明。

在“附注”栏中的状态码分别表示：

- A** ....新文件
- B** ....没有改动，但以新的订货号重印
- C** ....有改动，并重新发行

版本	订货号	附注
2004.03	??????	A

## 注册商标

SIMATIC®, SIMATIC HMI®, SIMATIC NET®, SIMODRIVE®, SINUMERIK® 和 SIMOTION® 均为西门子公司的注册商标。

本文件中的其他名称也可能是商标，任何第三人擅自使用此商标将会侵犯注册商标所有人的权利。

### ©西门子股份公司版权所有 2003年

没有明确的书面许可，任何人不得翻印、传播和使用本文献及其中的内容，违者将负责赔偿损失。西门子公司享有所有版权及相关权利，包括专利权或实用新型的申请注册权。

### 责任免除

经过审查，本文献的内容与其描述的软件和硬件相符合。但是仍可能存在一些差异。因此我们不能保证它们完全一致。我们会定期审查本文献，并在下一个版本中作出必要的修改。欢迎提出改进意见和建议。

© Siemens AG, 2003  
如有技术改动，恕不提前通知。

# 目录

<b>1. 欢迎使用Micro/WIN</b> .....	<b>1-1</b>
1.1. STEP 7-Micro/WIN 32窗口组件.....	1-1
1.2. 如何使用联机帮助.....	1-3
1.3. 如何定制STEP 7-Micro/WIN 32的外观.....	1-4
<b>2. 编程概念</b> .....	<b>2-1</b>
2.1. 控制程序如何作业.....	2-1
2.2. 编址概述.....	2-2
2.3. 如何组织程序.....	2-3
2.4. 项目组件和如何作业.....	2-4
2.5. 控制程序元素.....	2-4
<b>3. 如何输入梯形图逻辑程序</b> .....	<b>3-1</b>
3.1. 如何创建项目.....	3-1
3.2. 梯形图逻辑元素和如何作业.....	3-2
3.3. 如何在LAD中输入指令.....	3-3
3.4. 如何在LAD中输入地址.....	3-6
3.5. 如何在LAD中输入程序注解.....	3-8
3.6. 如何在LAD中编辑程序元素.....	3-9
3.7. 如何使用查找、替换和转入功能.....	3-11
3.8. 程序编辑器如何在LAD中显示输入错误.....	3-13
3.9. 如何在LAD中编译.....	3-14
3.10. 如何保存您的作业.....	3-14
<b>4. 如何建立通讯和下载程序</b> .....	<b>4-1</b>
4.1. 通讯概述.....	4-1
4.2. 如何测试通讯网络.....	4-2
4.3. 如何下载程序.....	4-3
4.4. 如何纠正编译错误和下载错误.....	4-4
<b>5. 如何监控和调试程序</b> .....	<b>5-1</b>
5.1. 停止和运行PLC操作模式.....	5-1
5.2. 调试和监控特征概述.....	5-3
5.3. 如何在状态图中显示状态.....	5-6
5.4. 如何在程序编辑器中显示状态.....	5-11
5.5. 如何执行有限次数的扫描通讯概述.....	5-16
<b>6. 如何管理项目</b> .....	<b>6-1</b>
6.1. 如何打印.....	6-1
6.2. 如何移动、拷贝、重新命名和邮寄项目.....	6-2
6.3. 如何从PLC上载.....	6-3
6.4. 如何在程序之间拷贝程序段.....	6-4
6.5. 使用较早版本的STEP 7-Micro/WIN项目.....	6-5
<b>7. PLC 指令</b> .....	<b>7-1</b>
7.1. 位逻辑指令.....	7-1
7.1.1. 常开接点.....	7-1
7.1.2. 常闭接点.....	7-2
7.1.3. 输出线圈.....	7-3
7.1.4. 置位线圈.....	7-4

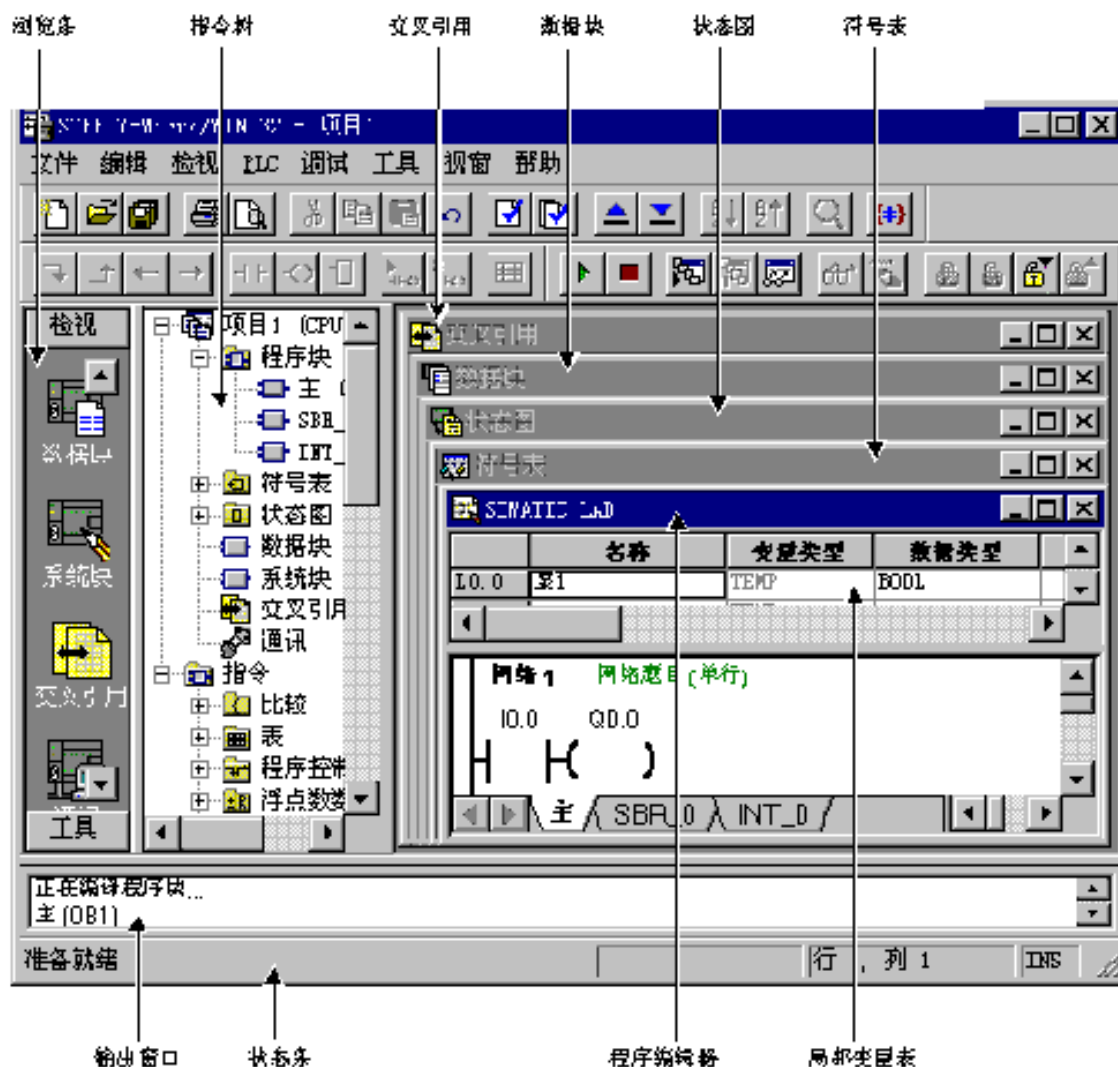
7.1.5.	复位线圈 .....	7-5
7.1.6.	上升沿检测接点 .....	7-6
7.1.7.	下降沿检测接点 .....	7-7
7.1.8.	NOT接点 .....	7-8
7.2.	定时器、计数器指令 .....	7-10
7.2.1.	保留性打开延迟计时器 TONR .....	7-10
7.2.2.	延时定时器 TON .....	7-11
7.2.3.	加计数器 .....	7-12
7.2.4.	加减计数器CIUD .....	7-13
7.3.	比较类指令 .....	7-14
7.3.1.	无符号字节比较 .....	7-14
7.3.2.	带符号字比较 .....	7-15
7.3.3.	带符号双字比较 .....	7-16
7.3.4.	带符号实数比较 .....	7-17
7.4.	算术类指令 .....	7-18
7.4.1.	字加减法 .....	7-18
7.4.2.	双字加减法 .....	7-19
7.4.3.	字乘法 .....	7-20
7.4.4.	浮点加减法 .....	7-21
7.4.5.	平方根 .....	7-22
7.4.6.	举例 .....	7-23
7.5.	逻辑类指令 .....	7-24
7.5.1.	字节与或异或 .....	7-24
7.5.2.	字节非 .....	7-26
7.5.3.	字与或异或 .....	7-27
7.5.4.	字非 .....	7-29
7.5.5.	双字与或异或 .....	7-30
7.5.6.	双字非 .....	7-32
7.6.	逻辑类指令 .....	7-33
7.6.1.	字节增减量 .....	7-33
7.6.2.	字增减量 .....	7-34
7.6.3.	双字增减量 .....	7-35
7.7.	数据转换类指令 .....	7-36
7.7.1.	双整数到实数转换 .....	7-36
7.7.2.	TRUNC指令 .....	7-37
7.8.	程序控制类指令 .....	7-38
7.8.1.	跳转至标签和标签 .....	7-38
7.8.2.	子程序调用 .....	7-39
7.9.	移位类指令 .....	7-40
7.9.1.	字节左右移位 .....	7-40
7.9.2.	字左右移位 .....	7-41
7.9.3.	双字左右移位 .....	7-42
7.10.	赋值类指令 .....	7-43
7.10.1.	字节赋值 .....	7-43
7.10.2.	字赋值 .....	7-44
7.10.3.	双字赋值 .....	7-45

7.10.4. 浮点赋值 .....	7-46
7.10.5. 字节转换 .....	7-46
<b>8. 编程举例 .....</b>	<b>8-1</b>
8.1. 电机Y- 起动 .....	8-1
8.2. 二进制分频器（触发电路） .....	8-4
8.3. 如何追踪一台设备运行了多长时间 .....	8-7
<b>9. SINUMERIK 802S/C base line PLC 子程序库说明 .....</b>	<b>9-1</b>
9.1. 概述 .....	9-1
9.1.1. 子程序库的构成 .....	9-1
9.1.2. 重要事项 .....	9-2
9.2. 子程序库说明 .....	9-11
9.2.1. 子程序 32 – PLC_INI（PLC 初始化） .....	9-11
9.2.2. 子程序 33 – EMG_STOP（急停处理） .....	9-12
9.2.3. 子程序 34-X_CROSS（点动键布局控制） .....	9-14
9.2.4. 子程序 35 – SPINDLE（用于开关主轴，或单、双极性的模拟主轴） .....	9-15
9.2.5. 子程序36 – MINI_HHU（为西门子小型手持单元设计6FX2006-1BG00） .....	9-20
9.2.6. 子程序 38 - MCP_NCK（MCP信号处理） .....	9-22
9.2.7. 子程序39-HMI_HW（根据操作面板信号选择手轮） .....	9-23
9.2.8. 子程序 40-AXES_CTL（进给轴主轴使能控制） .....	9-24
9.2.9. 子程序 41 - GEAR_CHG（主轴换挡控制） .....	9-27
9.2.10. 子程序 44 - COOLING（冷却液控制） .....	9-31
9.2.11. 子程序 45 - LUBRICAT（润滑控制） .....	9-33
9.2.12. 子程序 46 -TURRET1（霍尔元件刀架控制） .....	9-36
9.2.13. 子程序 48 - TOOL_DIR（换刀方向确定） .....	9-40
9.2.14. 子程序 49 - LOCK_UNL（卡紧放松控制） .....	9-41
9.2.15. 子程序51 - Trg_key_OR（进给/主轴超调） .....	9-43
9.3. PLC实例应用程序说明 .....	9-48
9.3.1. 主要功能 .....	9-48
9.3.2. 输入输出定义 .....	9-50
9.3.3. MCP按键布局 .....	9-52
9.3.4. 实例程序的PLC参数定义 .....	9-54
9.3.5. 实例程序结构说明 .....	9-55
9.3.6. 实例程序SAMPLE的调试 .....	9-59
<b>10. SINUMERIK 802D PLC 子程序库说明 .....</b>	<b>10-1</b>
10.1. 概述 .....	10-1
10.1.1. 子程序库的内容 .....	10-1
10.1.2. 重要须知 .....	10-2
10.1.3. 子程序库中提供的子程序表 .....	10-9
10.2. 子程序库说明 .....	10-10
10.2.1. 子程序 32 – PLC_INI（PLC初始化） .....	10-10
10.2.2. 子程序 33 – EMG_STOP（急停处理） .....	10-11
10.2.3. 子程序 34 – MCP_802D（802D机床控制面板MCP信号传递） .....	10-14
10.2.4. 子程序 35 – SPD_OVR（利用按键选择主轴倍率） .....	10-19
10.2.5. 子程序36 – MINI_HHU .....	10-21
10.2.6. 子程序37 – MCP_SIMU（机床操作面板MCP仿真） .....	10-23
10.2.7. 子程序38 – MCP_NCK（MCP和HMI信号处理） .....	10-24

10.2.8.	子程序39 – HMI_HW (根据HMI接口信号选择手轮)	10-26
10.2.9.	子程序40 – AXES_CTL (主轴和进给轴控制)	10-27
10.2.10.	子程序41 – GEAR_CHG (主轴换挡)	10-30
10.2.11.	子程序43 – MEAS_JOG	10-34
10.2.12.	子程序44 – COOLING (冷却控制)	10-35
10.2.13.	子程序45 – LUBRICATE (导轨润滑控制)	10-37
10.2.14.	子程序46 – TURRET1 (霍尔元件刀架控制)	10-39
10.2.15.	子程序47 – TURRET2 (二进制编码器刀架换刀控制)	10-42
10.2.16.	子程序48 – TOOL_DIR (判断就近换刀的方向)	10-45
10.2.17.	子程序49 – LOCK_UNL (锁紧/放松控制)	10-47
10.2.18.	子程序50 – MGZ_SRCH (搜索目标刀具所在的刀套位置)	10-48
10.2.19.	子程序51 – MGZ_RNEW (刷新刀套表)	10-49
10.2.20.	子程序52 – MGZ_INI (初始化刀套表)	10-50
10.2.21.	子程序63: TOGGLE	10-51
10.3.	PLC实例应用程序	10-52
10.3.1.	MCP仿真应用程序	10-52
10.3.2.	车床PLC应用程序实例	10-54
10.3.3.	铣床PLC应用程序实例	10-56
10.4.	附录 – 更改说明	10-59
<b>11.</b>	<b>附录</b>	<b>11-1</b>
11.1.	附录一: PLC数据类型和地址定义	11-1
11.2.	附录二: PLC指令表	11-3

# 1. 欢迎使用Micro/WIN

## 1.1. STEP 7-Micro/WIN 32窗口组件



**菜单栏** 允许您使用鼠标或键击执行操作。您可以在工具菜单中增加自己的工具，使其定制。

**工具条** 为最常用的STEP 7-Micro/WIN 32操作提供便利的鼠标访问。您可以定制每个工具条的内容和外观。

**浏览条** 显示编程特性的按钮控制群组：

“视图”一选择此一分类，为程序块、符号表，状态图，数据块，系统块，交叉参考及通讯显示按钮控制。

注释：当浏览条包含的对象因为当前窗口尺寸无法显示时，浏览条显示滚动按钮，使您能向上或向下移动至其他对象。

<b>指令树</b>	<p>提供所有项目对象和为当前程序编辑器提供的所有指令的树型视图。</p> <p>您可右击树中“项目”部分的文件夹，插入附加程序组织单元（POUs）；您可以右击单个POU，打开、删除、编辑其属性表，或重新命名子程序。</p> <p>您可以右击树中“指令”部分的一个文件夹或个别指令，以便隐藏整个树。</p> <p>您一旦打开指令文件夹，就可以拖放个别指令或双击，按照需要自动将所选指令插入程序编辑器窗口中的光标位置。</p>
<b>局部变量表</b>	<p>包含您对局部变量所作的分配（即子程序）。在局部变量表中创建的变量使用暂时内存；地址分配由系统处理；变量的使用仅限于创建此变量的POU。</p>
<b>程序编辑器窗口</b>	<p>包含用于该项目的编辑器的局部变量表和程序视图。</p> <p>如果需要，您可以拖动分割条，展开程序视图，并覆盖局部变量表。当您在主程序一节（OB1）之外，创建子程序时，制表符出现在程序编辑器窗口的底部。可单击制表符，在子程序和OB1之间移动。</p>
<b>输出窗口</b>	<p>在您编译程序时提供讯息。当输出窗口列出程序错误时，可双击错误讯息，会在程序编辑器窗口中显示适当的网络。</p>
<b>状态条</b>	<p>提供您执行STEP 7-Micro/WIN 32时的操作状态信息。</p>
<b>交叉参考</b>	<p>允许您检视程序的交叉参考和元件使用信息。</p>
<b>符号表 / 全局变量表窗口</b>	<p>允许您指定和编辑全局符号（即可在任何POU中使用的符号值，不只是创建符号的POU）。您可以创建多个符号表。</p> <p>在符号表 / 全局变量表中还有一个为系统定义符号的制表符，您可以在程序中使用该制表符。</p>
<b>状态图窗口</b>	<p>允许您将程序输入、输出或变量置入图表中，以便追踪其状态。</p> <p>您可以创造多个状态图，以便从程序的不同部分检视元件。每个状态图在状态图窗口中有自己的制表符。</p>
<b>数据块/数据初始化程序窗口</b>	<p>允许您显示和编辑数据块内容。</p>



## 1.2. 如何使用联机帮助

- 上下文相关的帮助**            对您希望获得帮助的标题，选择菜单项目或打开对话框，按“F1”键存取该标题的上下文相关帮助。（在某些情形下，可按“Shift”和“F1”键存取帮助标题。）
- 从菜单获得帮助**            STEP 7-Micro/WIN32中的“帮助”菜单提供下列选项：
- 目录和索引允许您借助目录浏览程序（显示哪一本书包含何种标题）或可搜索索引浏览该帮助系统。
  - 这是什么？提供接口元素定义。通过同时按Shift和F1键，您也能存取“这是什么？”帮助。光标变为一个问号；用它在您希望获得帮助的项目上单击。
  - 关于列出STEP 7-Micro/WIN32的产品和版权信息。
- 目录浏览程序**            STEP 7-Micro/WIN32中，从菜单条选择：帮助 > 目录和索引，调出“目录”浏览程序。  
从帮助标题内，单击“帮助标题”按钮，调出“目录”浏览程序。
- 索引**                        从按字母顺序排列的标题关键字列表，选择“目录”浏览程序中的“索引”标签。
- 如何打印**                    可从“目录”浏览程序或单独的标题打印标题：
- 欲打印书中的所有标题，从“目录”浏览程序选择书标题，并使用位于浏览程序底部的“打印”按钮。
  - 欲从“目录”浏览程序打印个别标题，选择标题，并使用位于浏览程序底部的“打印”按钮。
  - 欲从单个标题内打印，使用位于标题上方的“打印”按钮。

### 1.3. 如何定制STEP 7-Micro/WIN 32的外观

#### 说明

STEP 7-Micro/WIN32提供许多存取和显示信息的方法。

为了简化程序设计，您可以不显示导航条和输出窗口。您可以将在程序设计时需要的窗口盖住或最小化，例如局部变量表和符号表，仅在必要时调出。这为最基本的内容：指令树和程序编辑器窗口腾出最大的空间。

注释：当您编译时，输出窗口自动恢复。

以下是一些安排STEP 7-Micro/WIN32工作区不同组件的提示：

- 检视或隐藏各种窗口组件从菜单条选择“检视”，并选择一个对象，将其标选符号在打开和关闭之间切换。  
带标选符号的对象是当前在STEP7Micro/WIN32环境中打开的对象。
- 级联窗口：从菜单条选择窗口 > 级联、窗口 > 垂直或窗口 > 水平。
- 最小化、恢复、最大化或关闭窗口：  
使用位于每个窗口标题条中的最小化、恢复、最大化和关闭按钮。  
请注意，当您最大化窗口时，按钮显示在STEP 7-Micro/WIN32主窗口按钮下方的菜单条区内。当您最大化窗口时，窗口会盖住您已经打开的任何其他窗口显示，但最大化窗口不会关闭其他窗口。
- 使用制表符检视窗口的不同组件：  
诸如程序编辑器、状态图、符号表和数据块的窗口可以有多个制表符。例如，程序编辑器窗口包含的制表符允许您在主程序（OB1）、子程序和中断例行程序之间浏览。

**注释：**当您显示若干窗口时，可能没有足够的空间显示每个窗口包含的所有制表符。您可以用制表符旁的一对滚动箭头，从第一个制表符查找到最后一个制表符。（制表符旁的一对滚动箭头的功能与可能提供的允许您水平滚动窗口内容的滚动箭头不同。）

- 更改尺寸或移除局部变量表：  
将光标放置在程序编辑器和局部变量表的分隔条上方，拖动光标，以便增加或减少局部变量表的尺寸。如果程序不包含要求您定义任何局部变量的子程序或中断例行程序，则拖动程序编辑器，使之完全盖住局部变量表。（因为局部变量表是程序编辑器窗口的一部分，您无法取消局部变量表。）
- 移动或隐藏工具条：  
根据默认值，文件、调试和程序工具条出现在STEP 7-Micro/WIN32的菜单条下方。然而，您可将光标放在工具条区域内，移动该工具条。如果您将工具条拖至STEP 7-Micro/WIN32中任何窗口的边框附近，工具条将移至该窗口的边框，否则工具条成为一个独立的、自由漂浮的工具条。当工具条独立时，单击工具条标题条中的“X”按钮，隐藏工具条。您可以选择工具>定制，并从“定制”对话框“工具条”标签选择适当的复选框（文件、调试、阶梯、FBD、STL）恢复工具条。

## 2. 编程概念

### 2.1. 控制程序如何作业

#### 说明

当您将程序下载至PLC并且将PLC放置挨“运行”模式时，PLC的中央处理器（CPU）按下列顺序执行程序：

- CPU读取连接至可编程控制器的所有输入状态。该数据存储在输入内存区，即程序图像输入寄存器。
- CPU使用这些输入评估（或执行）控制程序逻辑。
- 当程序经过评估，CPU将程序逻辑结果存储在输出内存区，即程序图像输出寄存器。
- 在程序结束时，CPU将数据从程序图像输出寄存器写入至栏输出。
- 重复任务循环。

CPU任务的这个循环执行被称为扫描循环。S7-200CPU中的扫描循环包括下列任务：

- 读取数字输入
- 执行程序
- 处理任何通信请求
- 执行CPU自测试诊断程序
- 写入至数字输出

## 2.2. 编址概述

<b>绝对和符号地址</b>	您可以用绝对或符号方式识别程序中的指令操作数。绝对引用使用内存区和位或字节位置识别地址。符号引用使用字母数字字符组合识别地址。
<b>程序编辑器显示地址举例</b>	I0.0绝对地址由内存区和地址数目指定（SIMATIC程序编辑器）"INPUT1"全局符号名称（SIMATIC或IEC程序编辑器）???.? 或 ????? 红色问号表示一个未定义的地址（必须在程序编译之前已定义）
<b>全局范围与局部范围</b>	在符号表中分配的符号值（全局变量表）带有全局范围，在局部变量表中分配的符号值带有局部范围。
<b>全局符号</b>	全局符号可在SIMATIC程序编辑器中使用。在SIMATIC程序中，您通过使用符号表分配全局符号。全局符号。在您使用程序符号之前不必作出符号分配；您可以在任何时间进行符号指定。
<b>局部变量</b>	局部变量可在SIMATIC程序编辑器中使用。局部变量在各自的局部变量表中分配（POU），并在局部变量被创建的POU范围内有效。每个POU有其自身独立的局部变量表。

### 举例：

您在称作SBR1的子程序局部变量表中定义一个称作INPUT1的变量。当您从SBR1内引用INPUT1时，程序编辑器将其识别为SBR1的一局部变量。然而，如果您从程序中的其他位置（例如，从OB1或二级子程序）引用INPUT1，程序编辑器不将其视作局部变量（因为它位于SBR1的外部），而将INPUT1视作未定义全局符号。

**提示：** 如果您在局部和全局水平使用一个地址的相同名称，局部使用优先。即如果程序编辑器在局部变量表中找到特定程序块的名称定义，则使用该定义。如果找不到定义，程序编辑器则检查符号表 / 全局变量表。

### 举例：

您将PumpOn定义为全局符号。您还在SBR2中（而不是在SBR1中）将其作为局部变量定义。当程序被编译时，局部定义用于SBR2中的PumpOn；全局定义被用于SBR1中的PumpOn。

- 局部变量使用临时PLC、L内存，而不要求PLC程序内存空间。仅使用局部变量参数（或无参数）的子程序是可移动子程序，能在一个以上程序中重复使用。您不一定必须使用局部变量，它是高级编程技术的一个选项。如果您想要在若干个POU中使用一个参数，最好将其定义为全局符号表 / 全局变量表中的全局符号，而不要将其定义为局部变量，因为您必须对每个POU的局部变量表分别分配。
- 局部变量使用临时内存，每次POU被调用时，要确保POU内的局部变量被初始化。不能确定局部变量在一次迭代至下一次迭代的过程中仍保持数据值。

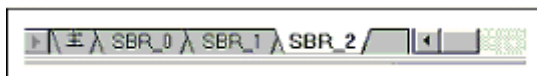
## 2.3. 如何组织程序

### 控制程序的基本元素

S7-200 CPU的控制程序由以下程序组织单位（POU）类型组成：

- 主程序：程序的主体（称为**OB1**）是您放置控制应用程序指令的位置。主程序中的指令按顺序执行，每次执行CPU的一次扫描循环。
- 子程序：子程序是指令的一个选用集，存放在分开的程序块中，仅在被主程序调用时执行。

STEP 7-Micro/WIN32通过为每个POU在程序编辑器窗口中提供分开的制表符组织程序。主程序**OB1**总是第一个制表符，后面才是您创建的子程序。



### 如何终止POU

因为程序被划分（每个POU占有一个分开的制表符），**OB1**或各种子程序的终止位置毫无成问题。编译程序根据实际情况用一个无条件**END**、**MEND**、**RET**终止每个POU。您不应该在程序提供该代码；如果在程序中加入了无条件**END**、**MEND**、**RET**，编译程序会显示一则错误。

### 子程序

子程序在您希望反复执行一个功能的位置特别有用；您只需在子程序中写入一次逻辑，然后在主程序过程中每次需要时调用子程序即可，而无需在主程序中您希望执行功能的每个位置重写该逻辑。这有若干优点：

- 您的总体代码尺寸减小。
- 因为您从主程序中移出了代码，您的扫描时间也会减少（在主程序中每次扫描循环都会自动评估，而无论是否已执行）。子程序能有条件地被调用，且在未调用时不在扫描期间评估。
- 子程序容易移动；这允许您挑出一个功能，并将其拷贝至其他的程序中，而很少或无须重新操作。

**注释：**V内存的使用限制了子程序的可移动性，因为来自一个程序的V内存地址分配可能与另一个程序中的分配发生冲突。相反，使用所有地址分配局部变量表的子程序却很容易移动，因为不必担心发生编址冲突。

## 2.4. 项目组件和如何作业

**说明** 在STEP 7-Micro/WIN32中。项目为您提供程序与所有需要与PLC交流及下载指定程序至特定PLC联系的途径。

项目包括下列基本组件：

- **程序块：**程序块由可执行代码和注解组成。可执行代码包含一个主程序（OB 1）和任意子程序。代码被编译并下载至PLC；程序注解被省略。
- **数据块：**数据块由数据（初始内存值；常数值）和注解组成。数据被编译并下载至PLC，注解被省略。
- **系统块：**系统块由配置信息组成，例如通讯参数、保留数据范围，模拟和数字输入过滤程序，用于STOP过渡的输出值和密码信息。系统块信息下载至PLC。
- **符号表：**符号表是为允许程序员使用符号编址的一种工具。符号有时对程序员而言更为方便，使程序遵循逻辑变得更加容易。下载至PLC的编译程序将所有的符号转换为绝对地址，符号表信息未被下载至PLC。
- **状态图：**状态图允许您观察在执行程序时进程值如果受到影响。状态图不下载至PLC；而仅是监控PLC（或模拟PLC）活动的一种工具。
- **交叉引用：**交叉引用窗口允许您检查列举在程序中何处使用操作数以及哪些内存区域已经被分配（位使用或字节使用）的表格。交叉引用及使用信息不下载至PLC。

**提示：**

- 您无法从PLC上载状态图、交叉引用或符号表信息，除非首先下载。
- 您在STEP 7-Micro/WIN32项目中所作的修改在您将其下载至PLC之后才会对程序产生影响。

## 2.5. 控制程序元素

**网络** 在LAD程序中，逻辑的基本元素用接点、线圈和方框表示。构成完整电路的一系列互联元素被称为网络。

**接点** 硬线输入由被称为接点的符号表示。正常状态下打开的接点在关闭时启用电流。接点亦可在正常状态下关闭。在这种情况下，打开接点时发生电流。


**线圈** 硬线输出由被称为线圈的符号表示。线圈有电流时，输出被打开。

**方框** 方框是代表在PLC内进行的复杂操作的符号。方框简化操作编程。例如，计时器和计数器由方框表示。

## 3. 如何输入梯形图逻辑程序

### 3.1. 如何创建项目

#### 打开新项目

双击STEP 7-Micro/WIN32图标 ，或从“开始”菜单选择 Simatic > STEP 7 Micro/WIN32，启动应用程序。会打开一个新的STEP 7-Micro/WIN32项目。

#### 打开现有项目

从STEP 7-Micro/WIN32中，使用文件菜单，选择下列选项之一：

- 打开—允许您浏览至一个现存的项目，并且打开该项目。
- 文件名称—如果您最近在一项目中工作过，该项目在“文件”菜单下列出，可直接选择，不必使用“打开”对话框。

您也可以使用 Windows Explorer 浏览至适当的目录，无需首先启动 STEP 7-Micro/WIN 32 即可打开您的项目。

**提示：**一旦打开一个项目，您就可以开始写入程序。开始之前，您可能需要执行下列一项或多项任务：




- 根据PLC类型进行范围检查  
为了使 STEP 7-Micro/WIN32 检查参数范围，您可以在写入程序前选择一个PLC类型。（如果您已经为项目指定了一个PLC类型，指令树用红色标记 x：显示对您的PLC无效的任何指令。）
- 设定通讯  
您可以现在设定通讯，或等到准备好下载程序时再设定通讯。
- 定制工作区域  
您可以采用多种方法定制工作区域。

## 3.2. 梯形图逻辑元素和如何作业

### LAD

阶梯逻辑（LAD）是一种与电气中继图相似的图形语言。

当您在LAD中写入程序时，您使用图形组件，并将其排列成一个逻辑网络。下列类型在您创建程序时可以使用：

- 触点 ：  
代表电源可通过的开关。电源仅在触点关闭时通过正常开启的触点（逻辑值一）；电源仅在触点打开时通过正常关闭或否定（非）触点（逻辑值零）。
- 线圈 ：  
代表由电流充电的中继或输出。
- 方框 ：  
代表当电流到达方框时执行的一项功能（如，计时器、计数器或数学运算操作）。

网络由以上元素组成并代表一个完全的线路。电源从左边的动力杆流过（在LAD编辑器中由窗口左边的一条垂直线代表）闭合触点，为线圈或方框充电。



### 3.3. 如何在LAD中输入指令

#### 输入LAD指令的方法

1. 指令树：拖放
2. 指令树：双击
3. 工具条按钮
4. 功能键
5. 插入与覆盖方式
6. 如何划线
7. 插入与覆盖模式

#### 插入与覆盖方式

STEP 7-Micro/WIN 32允许您在键盘上转换INSERT（插入）键，在两种编辑模式之间切换：

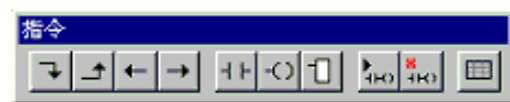
- 插入模式（按INSERT（插入）键时选择）中，如果您将一个指令放在另一个指令上，程序编辑器将现有指令移开，为新指令让出位置。
- 覆盖模式（INSERT（插入）键未按下时的默认值）中，如果您将一个指令放在另一个指令上，程序编辑器删除现有指令，并用新指令替代现有指令。

#### 覆盖模式中的参数保留

如果您用具有相同概况的另一个方框替代（覆盖）一个指令，对旧参数所作的任何赋值都会传送至新参数。即如果第二个指令与第一个指令有同样数目的电流输入、输入地址参数、电流输出和输出地址参数，则当您用第二个指令覆盖第一个指令时参数赋值被保留。

#### 划线

您可以从“程序”工具条使用水平和垂直线，或按住键盘上的CTRL键并按左、右、上或下箭头键，必要时在网络和左侧电源杆之间划线，完成网络绘制。



#### 指令树拖放

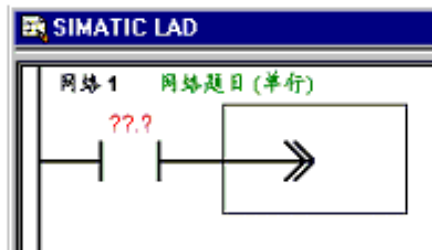
1. 选择指令。



2. 将指令拖曳至需要的位置。



3. 松开光标按钮，将指令放置在所需的位置。



**注释：**光标会自动阻止您将指令放置在非法位置（例如，放置在网络标题或另一指令的参数上）。

### 指令树双击

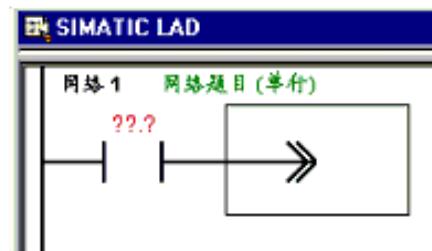
1. 在程序编辑器窗口中将光标放在所需的位置。一个选择方框在该位置周围出现。



2. 在指令树中，浏览至所需的指令并双击该指令。



双击后指令在程序编辑器窗口中出现。

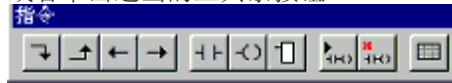


### 使用工具条

1. 在程序编辑器窗口中将光标放在所需的位置。一个选择方框在位置按钮或功能键周围出现。



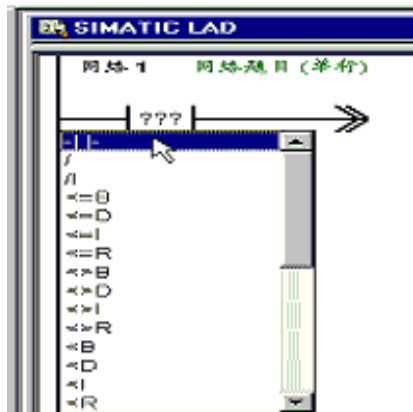
2. 或者单击适当的工具条按钮



或使用适当的功能键（F4=触点、F6=线圈、F9=方框）插入一个通用指令。

**注意：**该通用指令不能存取，并会阻止程序被编译；您必须按步骤3所示选择一个特定指令类型。

3. 出现一个下拉列表。滚动或键入开始的几个字母，浏览至所需的指令。在所需的指令上双击或使用ENTER键插入该指令。（如果此时您不选择特定的指令类型，则可返回网络，单击通用指令的记忆区域（该区域包含???, 而不是助记符），或者选择该指令并按ENTER键，将列表调回。）



### 3.4. 如何在LAD中输入地址

**说明** 当您在LAD中输入一个指令时，参数最初用问号代表，例如（???)或（????）。问号表示参数未赋值。您可以在输入元素时为该元素的参数指定个常量或绝对值、符号或变量地址或者以后再赋值。如果有任何参数未赋值，程序将不能正确编译。

**指定地址** 欲指定一个常数（例如100）或一个绝对地址（例如I0.1），只需在指令地址区域中键入所需的数值。（使用滑鼠或ENTER键选择键入地址区域。）



欲指定一个符号地址（使用诸如INPUT1的总体符号或局部变量），您必须执行下列简单的步骤：

1. 在指令的地址区域中键入符号或变量名称。
2. 如果是总体符号，使用符号表 / 总体变量表把符号名称指定给一个内存地址。

**注释：** 您不必预先定义符号，则可在程序中使用，您可以以后再定义内存地址。如果是局部变量，在程序编辑器窗口的顶端使用局部变量表。在“名称”列输入符号名称。因为编译程序会自动指定L内存地址，您不必为一个局部变量输入地址。您可以拖曳表格边缘，使局部变量表尺寸缩小到最小。使用局部变量是一种先进的编程技术。无经验的程序员应当考虑在符号表 / 总体变量表中将所有符号值指定为总体符号。

**提示** 在您输入这些地址值之后，程序编辑器自动格式化地址值。您无需键入格式化字符；在您完成键入后，这些字符会自动显示。

<b>程序编辑器</b>	I0.0	绝对地址由内存区域和地址数目决定（SIMATIC程序编辑器）
<b>显示地址举例</b>	#INPUT1	局部变量前有一个#符号（SIMATIC或IEC程序编辑器）
	INPUT1	总体符号名称（SIMATIC或IEC程序编辑器）
	??? 或 ????	红问号表示一个未定义的地址（必须在程序编译之前定义）

### 匹配地址和定义符号

当您用鼠标右键单击一个指令的参数时，弹出菜单允许您在符号表中快速定义地址，或根据已经在地址区域中键入的数值最多选择五个可能的匹配。



#### 提示:

- 使用ENTER键浏览网络的所有指令，每次一个参数，以便快速编辑所有必要的地址。
- 另外可通过用鼠标右键单击选择单个参数，并定义符号或使用弹出菜单为这些地址查找匹配。

### 有效和无效符号名称

符号名称允许包含数字字母字符和下划线，也允许包含扩展字符（ASCII 128至ASCII 255）。首字符仅限使用字母和扩展字符。

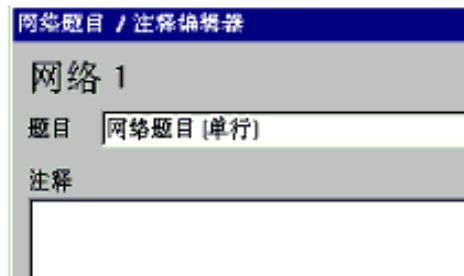
有效名称:	非法名称:
a11	1loop
a_b_1_2	!;kdl";ld
懒糖愚惚_1284938	

因此非法名称是以一个数字开始或包含不属于数字字母或扩展字符集的字符。

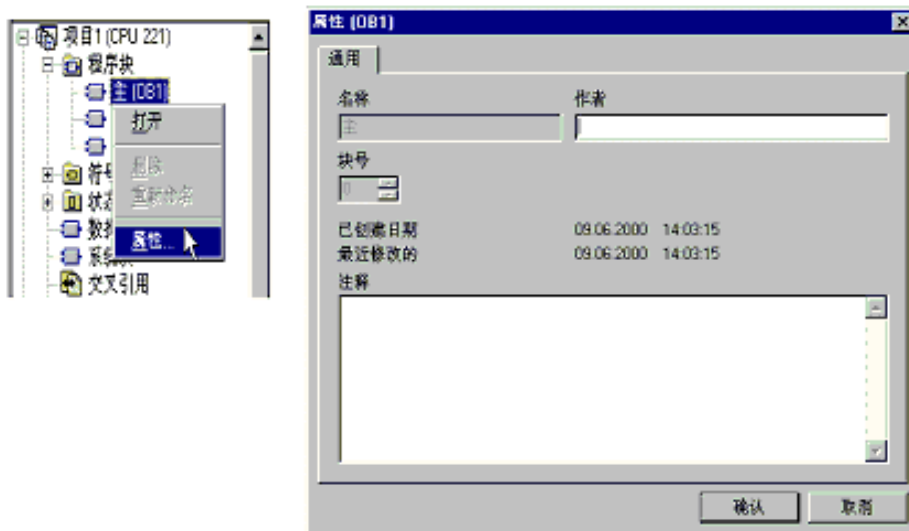
### 3.5. 如何在LAD中输入程序注解

**说明** 在LAD编辑器中有两级注解。

**网络注解** 将光标放在网络标题线上的任何地方，双击或按ENTER键，调用网络标题 / 注解编辑器。您可以输入一个标题，标识网络逻辑和关于网络内容的注解。网络标题在程序编辑器中显示；网络注解仅在网络标题 / 注解编辑器中显示，并在您打印程序注解时显示。网络标题允许的最大字符数目是127；网络注解允许的最大字符数目是1,023。



**项目组件注解** 使用指令树浏览至POU，并用鼠标右键单击该POU，调出该项目的菜单与“属性”对话框。



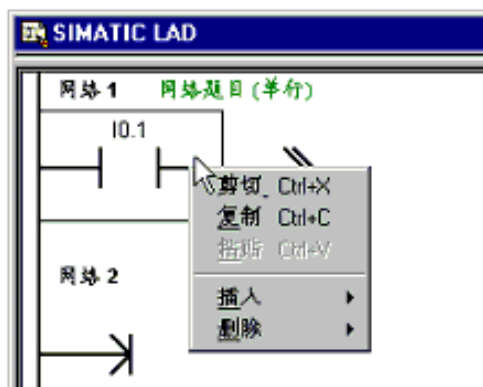
### 3.6. 如何在LAD中编辑程序元素

#### 剪切复制粘贴 或删除多个网络

通过拖曳鼠标或使用SHIFT键和UP（上）、DOWN（下）箭头键，您可选择多个相邻网络，用于剪切、复制、粘贴或删除选项。使用工具条按钮，从“编辑”菜单选择命令，或用鼠标右键单击，调出编辑选项的弹出菜单。

#### 编辑单元指令 地址和网络

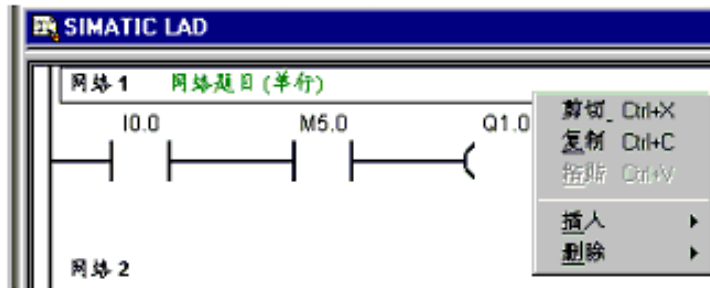
- 当您单击程序编辑器的空单元时，会出现一个方框，显示您选择的单元。您可以使用弹出菜单在空单元中粘贴一个选项，或在该位置插入一个新行、列、垂直线或网络。您也可以从空单元位置删除网络。
- 单击指令时，会在指令周围出现一个方框，显示您选择的指令。您可以使用弹出菜单在该位置剪切、拷贝或粘贴指令，以及插入或删除（适当的）行、列、垂直线或网络。



- 单击指令时，会在栏周围出现一个方框，显示您选择的参数。您可以使用弹出菜单撤消键入、剪切、拷贝、粘贴或删除信息，或快速选择栏内容（“全选”）。您也可以双击的方法全选。



- 单击网络标题时，在标题区域周围会出现一个方框。然而，弹出菜单允许您在整个网络中执行编辑，而不仅限于标题。您可以在网络上剪切、拷贝或粘贴，以及插入一个新网络或删除现有网络。



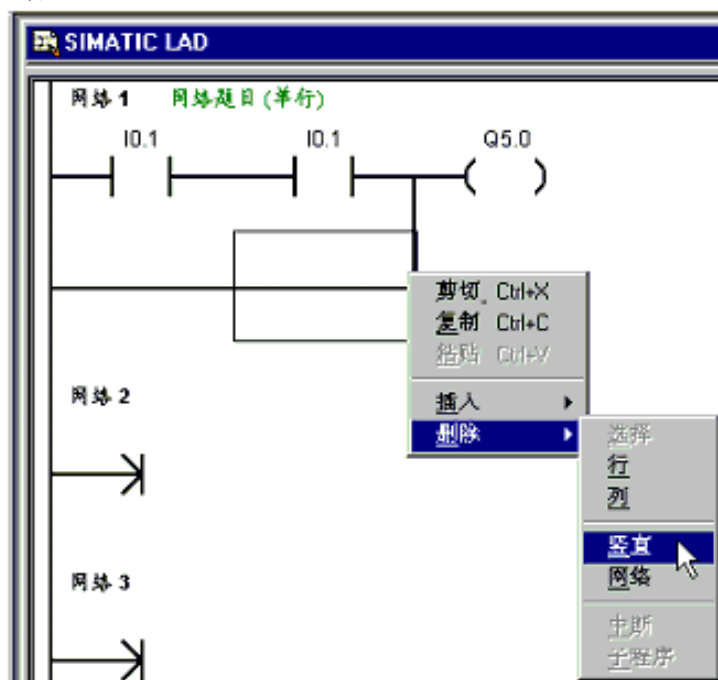
您也可以使用工具条按钮、标准窗口控制键和“编辑”菜单剪切、拷贝或粘贴选项。

### 将元素移近

您可以剪切和粘贴元素和行，删除行或列；然而，程序编辑器要求元素之间有一定的间距。在某些情形中，您不能把某个元素靠近另一个元素（例如，在方框指令之间要求有一个水平线段，并且不能移除）。

### 删除元素

您可以用DELETE（删除）或BACKSPACE（退格）键删除个别元素；您可以用“编辑”菜单或用滑鼠右键单击的方法调出弹出菜单，以便删除行、列、垂直线和网络。



**注释：** 为了正确选择一条删除的垂直线，始终将光标放在垂直线左边的单元上。



### 3.7. 如何使用查找、替换和转入功能

#### 使用“查找/替换”或“转入”

- 选择编辑 > 查找 (Edit>Find)，编辑 > 替换 (Edit>Replace) 或编辑 > 转入 (Edit>GoTo) 菜单命令
- 按 CTRL+F 执行查找，CTRL+H 执行替换或 CTRL+G 执行转入

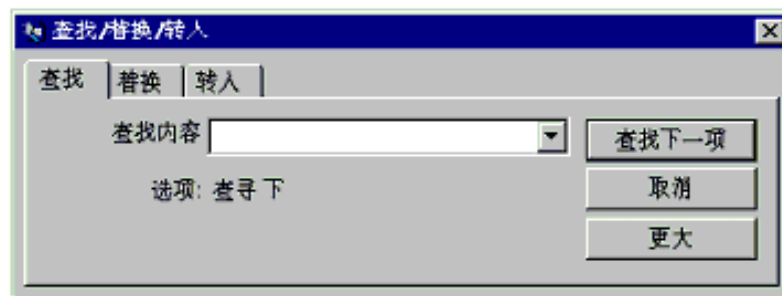
#### 何处使用

可以在程序编辑器窗口、局部变量表，符号表、状态图表、交叉引用标签和数据块中使用“查找”、“替换”和“转入”。

#### 如何操作

- “查找”功能允许您定位指定的字符串，例如操作数、网络标题或指令助记符。（“查找”不搜索网络注解，仅搜索网络标题。“查找”不搜索LAD中的网络符号信息表。）
- “替换”功能允许替换指定的字符串。（“替换”不对指令助记符操作。）
- “转入”功能允许您通过指定网络数目或您希望浏览的行的方式快速移至另一个位置。

#### 查找功能



1. 在“查找内容”栏中键入要搜索的字符串。
2. 欲移至下一个搜索字符串，单击“查找下一项”按钮。

**注释：**在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反应了操作数在代码中的存储方式。

欲进一步定义搜索，必须单击“更多”按钮，该按钮允许您为按下列方式定制搜索：

- 您可以使用“搜索”列表框选择搜索的方向。
- 可以选择“大小写匹配”复选框，仅搜索与您在“查找内容”中键入字符串大小写数值相同的字符串。
- 可以选择“全字”复选框，除去包含作为较长字的一部分的搜索短语的字符串。
- 您可以选择适当的“SPAN”复选框，对所有的POU（OB1、所有的子程序和中断例行程序）或局部变量表的所有实例、符号表或状态图表进行搜索。
- 可以指定对一定的行范围进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认范围；您也可以键入网络或行号码，作为搜索的起始和结束。
- 您可以指定是否通过选择适当的复选框搜索网络标题和 / 或程序代码。

### 替换功能



1. 在“查找内容”栏中键入要搜索的字符串。
2. 在“替换内容”栏中键入您希望用作替换搜索结果的字符串。
3. 欲查找出现的搜索字符串，单击“查找下一项”按钮。

注释：在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反应了操作数在代码中的存储方式。

4. 果您希望替换字符串，单击“替换”。如果您已经仔细地定义搜索字符串而且没有误改的可能，您可以单击“全部替换”，替换所有字符串实例，而无须逐一检验每个字符串。

欲进一步定义搜索，必须单击“更多”按钮，该按钮允许您按照下列方法定制搜索：

- 可以使用“搜索”列表框选择搜索的方向。
- 可以选择“大小写匹配”复选框，仅搜索与您在“查找内容”中键入字符串大小写数值相同的字符串。
- 可以选择“全字”复选框，除去包含作为较长字的一部分的搜索短语的字符串。
- 您可以选择适当的“SPAN”复选框，对所有的POU（OB1、所有的子程序和中断例行程序）或局部变量表的所有实例、符号表或状态图表进行搜索。
- 可以指定对一定的行范围进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认范围；您也可以键入网络或行号码，作为搜索的起始和结束。
- 您可以指定是否通过选择适当的复选框搜索网络标题和 / 或程序代码。

### 3.8. 程序编辑器如何在LAD中显示输入错误

错误一

红色文字显示非法语法。



**注释：**当您用有效数值替换非法地址值或符号时，字体自动更改为默认字体颜色（黑色，除非您已定制窗口）。

错误二

一条红色波浪线位于数值下方表示该数值或是超出范围或是对不符合此类指令。



错误三

一条绿色波浪线位于数值下方表示正在使用的变量或符号尚未定义。

STEP 7-Micro/WIN32允许您在定义变量和符号之前写入程序。您可随时将数值增加至局部变量表或符号表中。



提示



- 用鼠标右键单击，调出与全局符号最接近的五个匹配菜单。
- 使用“定义符号”对话框，解决未定义的符号。

### 3.9. 如何在LAD中编译

#### 如何编译

您可以用工具条按钮或PLC菜单进行编译。



- “编译” : 允许您编译项目的单个元素。在您选择“编译”时带有焦点的窗口程序编辑器或数据块是编译窗口；另外两个窗口不编译。
- “全部编译” : 对程序编辑器、系统块和数据块进行编译。当您使用“全部编译”命令时，哪一个窗口是焦点无关紧要。

#### 使用输出窗口 解决错误

当您编译时，“输出窗口”列出发生的所有错误。错误根据位置（网络、行和列）以及错误类型识别。您可以在错误线上双击，调出程序编辑器中包含错误的代码网络。

### 3.10. 如何保存您的作业

#### 说明

您可以使用工具条上的“保存”按钮  来保存您的作业，或从“文件”菜单选择“保存”和“另存为”选项保存您的作业。



- “保存”允许您在您的作业中快速保存所有改动。（然而，您初次保存一个项目时，会被提示核实或修改当前项目名称和目录的默认选项。）
- “另存为”允许您修改当前项目的名称和 / 或目录位置。

当您首次创建项目时，STEP 7-Micro/WIN 的默认值名称为“Project1.mwp”。您可以接受或修改该名称；如果您接受该名称，下一个项目的默认名称将自动递增为“Project2.mwp”。

STEP7Micro/WIN32项目的默认目录位置是位于“Microwin”目录中的称作“项目”的文件夹，您可以不接受该默认位置。

## 4. 如何建立通讯和下载程序

### 4.1. 通讯概述

#### 说明

如何在运行 STEP 7-Micro/WIN32 的个人计算机和PLC之间建立通讯取决于您安装的硬件。如果您仅使用PC/PPI电缆连接计算机和PLC，您只要连接电缆，接受当您安装 STEP 7-Micro/WIN32 软件时，在STEP 7-Micro/WIN32 中为个人计算机和PLC指定的默认参数即可。

您可以在任何时间建立通讯或编辑通信设定值。

以下列出为建立通讯要求的典型任务：

- 在PLC和运行STEP 7-Micro/WIN32的个人计算机之间连接一条电缆。对于简单的PC/PPI连接，将速度设为9600波特、DCE、11位。如果您在使用调制解调器或通讯卡，请参阅硬件随附的安装指令。
- 供选用：核实STEP 7-Micro/WIN32 中的PLC类型选项与您的PLC实际类型相匹配。
- 如果您使用简单的PC/PPI连接，您可以接受当您安装STEP 7-Micro/WIN32 时在“设定PG/PC接口”对话框中提供的默认通讯协议。否则，从“设定PG/PC接口”对话框为个人计算机选择一个不同的通讯协议并核实参数（站址、波特率等。）。
- 供选用：核实系统模块端口标签中的PLC配置（站址、波特率等。）。如有必要，修改和下载更改的系统模块。

## 4.2. 如何测试通讯网络

说明 测试通讯网络是否正常，有以下两种方法：

1. 在STEP 7-Micro/WIN32中，单击导航条中的“通讯”图标，或从菜单选择检视 > 通讯。



2. 从“通讯”对话框的右侧窗格，单击显示“双击刷新”的蓝色文字。



如果您成功地在网络上的个人计算机和设备之间建立了通讯，会显示一个设备列表（及其模型类型和站址）。

STEP 7-Micro/WIN32 在同一时间仅与一个PLC通讯。会在PLC周围出现一个红色方框，说明该PLC当前正在与STEP 7-Micro/WIN32 通信。





您可以双击另一个PLC，更改该PLC的通讯。

### 4.3. 如何下载程序

#### 说明

如果您已经成功地在运行STEP 7-Micro/WIN32 的个人计算机和PLC之间建立通讯，您可以将程序下载至该PLC。请遵循下列步骤。

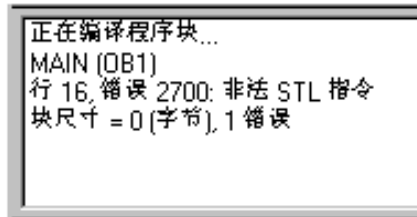
**注释：**从您的个人计算机将程序块、数据块或系统块下载至PLC时，从个人计算机下载的块内容盖写目前在PLC中的块内容（如果PLC中已经存在）。在您开始下载之前，确定您想要盖写PLC中的块。

1. 下载至PLC之前，您必须核实PLC处于“停止”模式。检查PLC上的模式指示器灯。如果PLC未设为“停止”模式，单击工具条中的“停止”按钮，或选择PLC > 停止。
2. 单击工具条中的“下载”按钮，或选择PLC > 下载。“下载”对话框出现。
3. 根据默认值，在您初次发出下载命令时，复选框“程序代码块”、“数据块”和“CPU配置”（系统块）被选择。如果您不需要下载某一特别的块，清除该复选框。
4. 单击“确定”按钮，开始下载程序。
5. 如果下载成功，一个确认方框会显示以下讯息：“下载成功。”程序继续执行步骤12。
6. 如果STEP 7-Micro/WIN32中的CPU类型数值与实际PLC不匹配，会出现一个带有以下讯息的警告方框：“为项目选择的PLC类型与远程PLC类型不匹配。继续下载吗？”
7. 欲纠正PLC类型选项，选择“否”终止下载程序。
8. 从菜单条选择PLC > 类型，调出“CPU类型”对话框。
9. 您能从下拉列表方框选择纠正类型，或单击“读取CPU”，由STEP 7-Micro/WIN32 自动读取正确的数值。
10. 单击“确定”按钮，确认PLC类型并清除对话框。
11. 单击工具条中的“下载”按钮，重新开始下载程序，或从菜单条选择PLC > 下载。
12. 一旦下载成功，在PLC中运行程序之前，您必须将PLC从“停止”模式转换回“运行”模式。单击工具条中的“运行”按钮，或选择PLC > 运行转换回“运行”模式。

#### 4.4. 如何纠正编译错误和下载错误

##### 说明

输出窗口在您编译程序或下载程序时随时自动显示编译程序信息和错误讯息。讯息通常包括发生错误的网络、列和行位置以及一个错误代码和说明。



```
正在编译程序块...
MAIN (OB1)
行 16, 错误 2700: 非法 STL 指令
块尺寸 = 0 (字节), 1 错误
```

双击错误讯息，在程序编辑器中显示出错的网络。

如果您已经关闭输出窗口，从菜单条选择视图 输出窗口，重新显示输出窗口。



## 5. 如何监控和调试程序

### 5.1. 停止和运行PLC操作模式

#### 说明

请使用下列一种方法改变PLC操作模式：

- 单击“运行”按钮，进入“运行”模式，或单击“停止”按钮，进入“停止”模式。
- 选择PLC > 运行（PLC > RUN）菜单命令，进入“运行”模式，或选择 PLC > 停止（PLC > STOP）菜单命令，进入“停止”模式。
- 用手动方式更改PLC上的模式开关。
- 在程序中插入“停止”指令。

**注释：**欲使用STEP 7-Micro/WIN 32软件控制“运行 / 停止”模式，在STEP 7-Micro/WIN 32 和PLC之间必须存在通讯路径。另外，PLC硬件模式开关必须设为“终端”或“运行”。将模式开关设为“终端”并不改变PLC的操作模式，但允许STEP 7-Micro/WIN 32更改PLC操作模式。

PLC前方的状态LED表示当前操作模式。

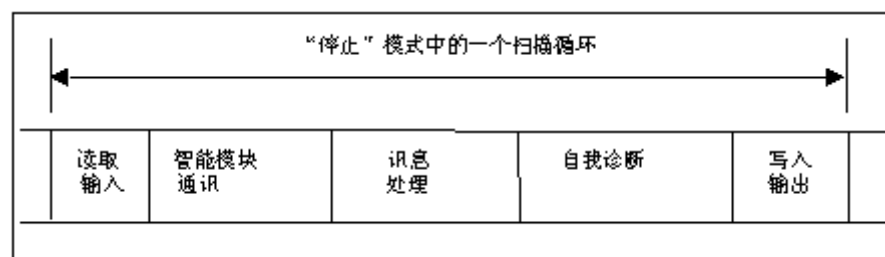
#### PLC操作模式详情

PLC具有两种操作模式：“停止”和“运行”模式。在“停止”模式中，您可以创建 / 编辑程序。在“停止”模式中不得执行程序。但在“运行”模式中可执行程序。此外，在“运行”模式中，您可以创建、编辑和监控程序操作和数据。为您提供调试协助，以便增强追踪程序操作和识别编程问题的能力。

可在“停止”模式中使用调试协助，例如第一次和多次扫描功能，并导致预定扫描次数从“停止”更改为“运行”模式。

PLC操作系统存储严重错误，并强制从“运行”模式更改为“停止”模式。如果PLC检测到严重错误，在严重错误条件依然存在时不允许从“停止”模式更改为“运行”模式。PLC操作系统功能存储非严重错误，用于检查，但不会使模式从“运行”更改为“停止”。

在停止模式中，PLC处于半闲置状态。用户程序执行被中断；执行输入更新；用户中断条件被禁用。下图描述PLC在“停止”模式中遵守的时间表。



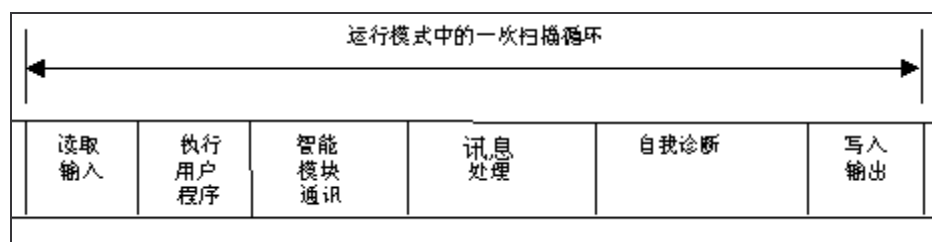
发生通讯中断时，PLC收到讯息并在适当的情况下执行请求。当PLC仍然处于“停止”模式时，图像寄存器作出I/O数值更改。一种例外的情况是对图像寄存器盖写I/O数值更改的强制功能。处于“停止”模式时，您能够载入、上载或删除用户程序内存。

一台或多台设备尝试通过通讯端口与PLC通讯时，PLC按顺序对每一请求作出应答。PLC不尝试阻止一台通讯设备的行动干扰另一台通讯设备的行动。所有必要

的防止此类干扰的措施均须由您的系统设计提供。

自我诊断检查包括定期检查操作系统EEPROM、I/O模块状态检查和对扩展I/O每次存取执行的I/O扩展总线一致性检查。

在运行模式中，PLC读取输入、执行程序、写入输出、对通讯请求作出应答、更新智能模块、执行内部管理工作以及对您的中断条件作出应答。PLC不支持用于运行”模式执行循环的固定扫描次数。根据发生的优先顺序对这些行动（中断除外）提供服务。如以下所示，此一执行循环被称为扫描循环。



每次扫描循环开始时先读取当前输入位数值，并将这些数值写入输入图象寄存器。输入位无相应的物理输入，但与具有物理输入位处于同一字节中，每次输入更新循环在图象寄存器中被设为零，除非被强制。

读取输入后，程序从第一条指令开始执行，直至遇到结束指令。遇到结束指令时，PLC检查系统的智能模块是否需要服务。如果需要，读取讯息并将讯息存储在循环的下一阶段缓冲器中。

在扫描循环的讯息处理阶段，从通讯端口接收的讯息被处理。完成的应答被置于一旁，等待在适当的时候传输给给通讯请求方。

自我诊断检查包括对操作系统EEPROM和用户程序内存以及I/O模块状态检查进行定期检查。

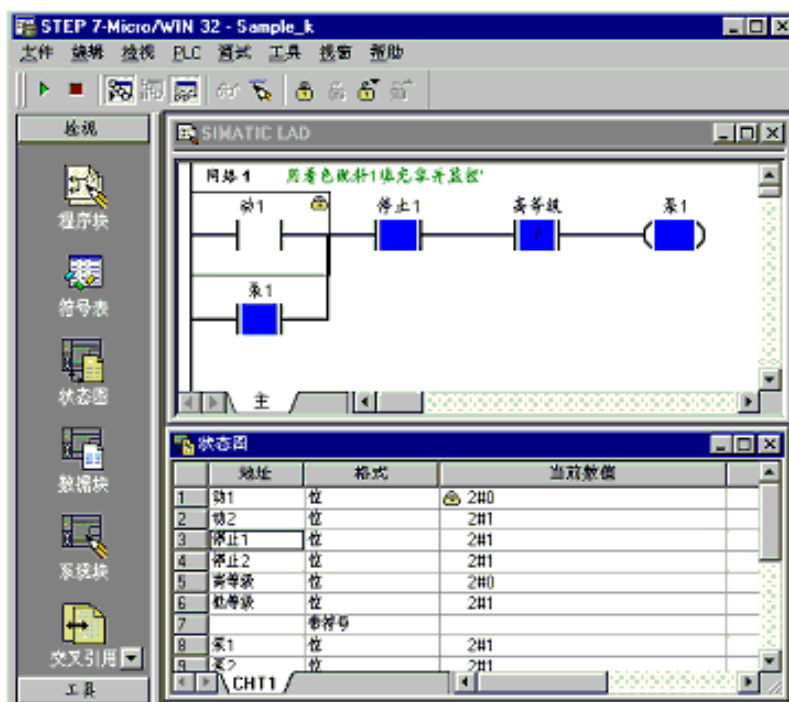
最后，输出图象寄存器数值被写入输出模块，从而完成一个扫描循环。

## 5.2. 调试和监控特征概述

**说明** 您成功地在您运行的STEP 7-Micro/WIN 32 编程设备和PLC之间建立通讯并向PLC下载程序后，您就可以利用STEP 7-Micro/WIN 32 提供的诊断特征调试新程序并监控正在运行的程序。

**什么是“状态”** “状态”这一术语指在PLC中执行您的程序时，对操作数当前数值的信息显示。您可以通过建立状态图或打开程序编辑器窗口中的程序状态显示状态信息。下图是 STEP 7-Micro/WIN 32 状态图和程序编辑器窗口中的状态信息举例。

注释：注意此处删去了不必要的项目元件（例如，指令树和输出窗口），以便有更充足的空间显示所需的元件（SIMATIC LAD程序编辑器窗口、状态图窗口、调试工具条和浏览条中的状态图图标）。改变任务时，“视图”菜单允许您定制STEP 7-Micro/WIN 32 环境，以便您仅显示您当前正在使用的项目元件。



**前提条件** 在收集状态用于监控或调试程序之前，您必须完成下列任务：

- 成功地编译程序。
- 成功地在STEP 7-Micro/WIN 32和PLC之间建立通讯。
- 成功地向PLC下载程序。

**注释：**STEP 7-Micro/WIN 32 中项目的时间标记和PLC必须相匹配，才能打开状态。时间标记比较（在您尝试打开状态时自动执行）保证STEP 7-Micro/WIN 32 中的项目状态表示正确反映PLC程序。

如果时间标记不匹配，您可以检查“时间标记不匹配”对话框中的下拉列表，了解哪些网络引起不匹配。

- 下载程序后，您还应将PLC转换回“运行”模式。否则虽然能报告操作数状态，但PLC无法执行程序，因此您将无法看到预期的逻辑操作。

**PLC模式** PLC模式决定您可以执行的监控和调试操作类型。

虽然无法在“停止”模式中执行程序，但PLC操作系统继续监控PLC（收集PLC

RAM和I/O状态)、将状态数据传送至STEP 7-Micro/WIN 32、并执行任何强制或取消强制命令。当PLC处于“停止”模式时,您可执行下列操作:

- 利用图状态或程序状态查看操作数的当前数值。(这与执行单次读取效果相同,因为程序尚未执行。)
- 利用图状态或LAD程序状态强制数值。
- 利用图状态写入数值。
- 写入或强制输出。
- 执行有限数目的扫描,并通过状态图和 / 或程序状态查看结果。

当PLC处于“运行”状态时,您不能使用“第一次扫描”或“多次扫描”特征。您可以在状态图中写入或强制数值,或利用LAD程序编辑器强制数值,正如在“停止”模式中的状况相同。您还可以执行下列操作(不适用于“停止”模式):

- 利用图状态收集连续更新。(如果需要单次更新,则必须关闭图状态,才能使用“单次读取”命令。)
- 利用程序状态收集连续更新。
- 编辑程序并利用“运行”模式特征中的“程序编辑”功能向PLC下载改动。

#### 通讯和扫描循环

PLC在连续循环中读取输入、执行程序逻辑、写入输出并执行系统操作和通讯。这一扫描循环速度极快,每秒钟执行多次。虽然STEP 7-Micro/WIN 32能够快速发出状态请求,但您应当意识到您并不能看到PLC中发生的每一事件。当您利用程序状态或图状态从PLC中的用户程序读取数据值时,您就实际上在进行数据采样。从PLC读取的状态数值更新速率取决于通讯路径的波特率。使用最大波特率可获得最快更新速率。

#### 收集状态不同方法

LAD程序编辑器中监控程序状态时,在每次扫描循环的结尾收集状态。

如果操作数受几项指令的操纵,则不向状态报告该操作数的中间数值;LAD程序状态仅在程序扫描的结尾处显示操作数数值。

您可以连续或间断(单次读取状态)收集状态。

当PLC处于“运行”模式时,打开程序编辑器窗口并打开程序状态,查看连续状态更新。请记住“连续”并不意味着实时;它只意味着编程设备快速从PLC获取状态信息并在屏幕中显示,以您的通讯方式所允许的最快速度更新显示。可能无法捕获并在屏幕中显示某些迅速变化的数值;或者这些数值可能变化太快,您无法读取。如果您将PLC转换成“停止”模式,您可以收集单次状态更新。另外,当PLC处于“停止”模式时,您可以利用“多次扫描”特征查看一次或多次扫描。您可以启用“第一次扫描”位,利用“第一次扫描”特征查看单次扫描。

当PLC处于“运行”模式时,打开状态图窗口并打开图状态,查看连续更新。如果您只希望收集一次状态更新并且不希望将PLC转换成“停止”模式,则可关闭图状态并利用“单次读取”特征。如果您将PLC转换成“停止”模式并打开图状态,这样您也可以收集单次状态更新。另外,查看状态图时您也可以利用“多次扫描”和“第一次扫描”特征。

#### 模拟程序条件

您可以通过向操作数写入新数值的方法模拟程序条件:使用状态图。

您可以使用状态图LAD程序状态强制数值

#### 检查交叉引用和元素使用状况

调试程序时,您可以决定是否增加、删除或编辑参数。

使用“交叉引用”窗口观察当前程序中如何指定参数,这样可以防止您意外地重复赋值。

另外，如果您希望在“运行”模式中执行程序编辑，并且某些网络使用正向或负向过渡（EU或ED指令），您一定要检查“交叉引用”信息中的EU/ED数目，以防重复指令号码。

### 向程序下载改动

一旦调试部分代码，如果PLC处于“停止”模式，您可以编辑程序、下载程序、然后再次监控状态，查看改动是否发生作用。

您不必进入“停止”模式就可以在“运行”模式中执行程序编辑并向PLC下载较小的改动。



---

### 警告

在“运行”模式中向PLC下载改动时，您的改动会立即影响程序操作。不得出错；程序编辑中的错误可能导致人员伤亡和 / 或设备损坏。仅限合格人员在“运行”模式中执行程序编辑。

---

### 5.3. 如何在状态图中显示状态

**说明**

向PLC下载程序后，您可以建立一个或多个状态图，监控和调试程序操作。如果PLC处于“运行”模式，程序在连续扫描的基础上执行。您可以打开图状态，连续更新图数值。作为打开图状态的替代方法，您可以使用“单次读取”功能，收集状态图数值的单次“瞬态图”。查看状态图时，您也可以使PLC进入“停止”模式，并使用“第一次扫描”或“多次扫描”功能监控有限次数扫描的程序操作。

**提示：** 请记住，打开图状态后不能再编辑图！关闭图状态，编辑图。

**状态图举例**

打开状态图与启动状态图不同。您可以打开状态图进行编辑或检查，但除非您（从“调试”菜单或工具条）发出“单次读取”命令或（从“调试”菜单或工具条）启动状态图，否则不会在“当前数值”列中显示状态信息。

利用单次读取特征（只有在关闭图状态时才能使用）检查状态图时，从PLC收集当前数值，并在“当前数值”列显示，但PLC执行程序时并不对其进行更新。

（从“调试”菜单或工具条）启动图状态时，在连续循环的基础上从PLC收集当前数值。从PLC接收到改变时，对“当前数值”列进行更新。


您可以利用“新数值”列指定（写入或强制）您确定的一个数值。您可以从“当前数值”、“格式”或“地址”列强制当前数值。

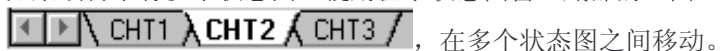
	地址	格式	当前数值	新数值
1	起始_1	位	2#1	
2	起始_2	位	2#0	
3	停止_1	位	2#1	
4	停止_2	位	2#0	
5	高位	位	2#0	
6	低位	位	2#0	
7	重设	位	2#0	
8		带符号		
9	泵_1	位	2#1	
10	泵_2	位	2#0	
11	混合器_启动	位	2#0	
12	蒸汽_阀	位	2#0	
13	排液_阀	位	2#1	
14	排液_泵	位	2#1	
15		带符号		
16	高位_已选则	位	2#1	
17	混合_定时器	带符号	+7426	
18	循环_计数器	带符号	+0	

**打开状态图与启动图状态**



打开图查看或编辑图内容。启动图收集状态信息。可使用下列一种方法打开状态图：

单击浏览条上的“状态图”  按钮。

- 选择查看 > 状态图 (View>Status Chart) 菜单命令。
- 打开指令树中的“状态图”文件夹，然后双击“图”  图标。
- 如果项目中有多个状态图，使用位于状态图窗口底部的“图”标签



请使用下列一种方法启动状态图：

- 为了连续收集状态图信息，启动状态图：使用菜单命令 **调试 > 图状态 (Debug>Chart Status)** 或使用“图状态”  工具条按钮。
- 为了获得单次“瞬态图”数值，使用“单次读取”功能：使用菜单命令 **调试 > 单次读取 (Debug>Single Read)** 或使用“单次读取”  工具条按钮。  
(如果您已经启动图状态，但“单次读取”功能被禁用。)

#### 提示：


- 打开图并不意味着您在查看状态，您必须启动图才能收集状态信息。
- 如果图空置，则启动图毫无意义：您必须首先“建立”图，方法是将程序数值（操作数）放在“地址”列，并为每个“格式”列选择数据类型。

### 使用多个状态图

可采用下列几种方法创建附加状态图：

- 从指令树，用鼠标右键单击“状态图”文件夹，并选择弹出菜单命令插入状态图。
- 打开状态图窗口，并使用“编辑”菜单或用鼠标右键单击，调出弹出菜单，选择插入 > 图 (Insert>Chart)。

#### 注释：

- 成功地插入新状态图后，在状态图窗口底部会出现新标签，，并在指令树窗口中“状态图”图标下方另外创建一个元件。若要对状态图重新命名，用鼠标右键单击指令树中的对应图标，并选择“重新命名”。欲在图之间移动，只需单击所需的图标签。
- 有时位于标签部分右侧的滚动按钮可能会遮住标签。如果看不到标签，拖曳标签区域和滚动按钮之间的分界线，以便看到更多的标签。

### 建立图

您可以在状态图中输入地址，以便监控或修改程序数值。可按位或字显示计时器和计数器数值。如果您按位显示计时器或计数器数值，则显示输出状态（输出打开或关闭）。如果您按字显示计时器或计数器数值，则使用当前数值。

欲建立状态图，请遵循下列步骤：

1. 在“地址”栏为每个所需数值输入地址。PLC内存地址范围中列出的大多数内存类型有效，但数据常量、累加器和高速计数器除外。  
(欲编辑地址单元，使用箭头键或鼠标选择希望编辑的单元。开始键入时，栏清除原有字符并接收输入的新字符。如果您单击鼠标或按“F2”键，栏会增亮，您可以使用箭头键将编辑光标移至您希望编辑的位置。)

地址	
1	I0.0

2. 如果元素是位（例如I、Q或M），则“格式”列中的格式被设为位。如果元素是字节、字或双字，从“格式”列选择单元，并双击或按 **SPACEBAR** 或 **ENTER** 键，浏览有效格式，直至出现适当的格式。

格式
位





（但是如果您已经启动图状态，“单次读取”功能则被禁用。）

- 欲连续收集状态图信息，启动图状态：使用菜单命令  
调试 > 图状态 (Debug>Chart Status) 或使用“图状态”工具条按钮。

## 写入与 强制数值

全部写入”功能允许您向程序写入一个或多个数值，模拟一种条件或一系列条件。然后您可以运行程序或使用状态图（以及程序状态（如果愿意））监控运行状况。执行程序时，您用“全部写入”功能修改的数值可能用新数值盖写。

注释：如果您向I/O点写入数值，您可能永远也看不到它作为当前数值出现在状态图中，因为下一次扫描时，PLC读取实际栏数值（而不是您在状态图中写入的数值）。使用“强制”功能向I/O指定数值更加有效。

“强制”功能允许您模拟逻辑条件（通过强制V、M、T或C内存）或物理条件（通过强制I/O点）。当您在状态图中使用“强制”功能强制一个或多个程序数值时，每次扫描循环时都会有效地重设数值。虽然在单次扫描过程中，程序可修改强制数值，但新扫描开始时，会重新应用强制数值。由于PLC执行速度很快，而从PLC向您的编程设备传输状态信息和在编程设备屏幕上显示息所需的延误时间很长，您可能无法观察到强制数值的任何改变。但是，在单个扫描过程中可以观察到此种改变。



### 警告

在程序中强制数值时，在程序每次扫描时将操作数重设为该数值，与输入 / 输出条件或其他正常情况下对操作数值有影响的程序逻辑无关。强制可能导致程序操作无法预料，无法预料的程序操作可能导致人员伤亡和 / 或设备损坏。强制功能是调试程序的辅助工具，切勿为了弥补处理设备的故障而进行强制。仅限合格人员使用强制功能。强制程序数值时，务必通知所有授权维修或调试程序的人员。

## 在状态图中 使用调试功能

您可从“调试”菜单或“调试”工具条存取“调试”功能（单次读取、全部写入、强制、取消强制、全部取消强制和读取全部强制）。



### 单次读取：

欲获得“瞬态图”，即对全部数值程序状态的一次更新，使用“单次读取”功能。在默认情况下，图状态不断地从PLC获取状态更新信息。单击状态图且图状态关闭时，则启用“单次读取”按钮。



### 全部写入：

对状态图中的“新数值”列的改动完成后，您可以使用全部写入”将所需的改动传送至PLC。



### 强制：

欲将地址强制为某一数值，您必须首先规定所需的数值，可通过读取数值（如果您希望强制当前数值）或键入数值（如果您希望将地址强制为新数值）来完成。一旦您使用强制功能，每次扫描都会将数值重新应用于该地址，直至您对该地址执行取消强制。STEP 7-Micro/WIN 32允许您LAD程序状态以及图状态强制地址。LAD可以在图状态中使用强制特征。



#### 取消强制:

选择一个地址并使“取消强制”从该特定地址移除强制。您也可以选择（单击）一个参数，然后用滑鼠右键单击该参数，调出“强制”和“取消强制”功能弹出菜单。



#### 全部取消强制:

如果您要从全部地址移除强制，使用“全部取消强制”功能。应用“全部取消强制”之前不必选择单个地址。



#### 读取全部强制:

使用“读取全部强制”特征时，状态图的“当前数值”列为已经被明确强制、隐含强制或部分隐含强制的所有地址显示一个图标。



该图标表示该地址被明确强制，不能改变该地址的数值，直至对该地址取消强制。



该图标表示该地址被隐含强制。如果较大地址的一部分被明确强制，则该地址被认为是默认强制。例如，如果VW0被强制，则VB0是隐含强制（VB0是VW0的第一字节）。

隐含强制数值本身不能被取消强制。在改变该地址数值之前，您必须对较大地址取消强制。



该图标表示该地址的一部分被部分强制。例如，如果VW0被明确强制，则VW1的一部分也被强制（VW1的第一个字节是VW0的第二个字节）。部分强制数值本身不能被取消强制。在改变该地址数值之前，您必须对数值中被强制的地址取消强制。


如果发出“读取全部强制”命令时，地址旁边的“当前数值”列中未出现这三种图标，则该地址未被强制。

## 5.4. 如何在程序编辑器中显示状态

**说明** 打开程序状态

**提示:** 向PLC下载程序时, 会提示您将PLC转换成“停止”模式。如果您希望观察程序状态的连续更新, 请务必将PLC转回至“运行”模式。您可以检查STEP 7-Micro/WIN 32状态条, 查看PLC的当前模式。

可采用下列方法之一打开程序状态:

- 选择菜单命令调试 > 程序状态 (Debug>Program Status)。
- 单击“调试”工具条中的“程序状态”  按钮。

短暂停顿后, 程序编辑器窗口将开始显示状态。




如果操作数值等于1 (位打开), 布尔指令 (触点、线圈) 则被显示为彩色块。显示非布尔操作数值, 并以通讯速度允许的最快速度更新非布尔操作数。


**提示:** 打开程序状态时, 会禁用许多其他STEP 7-Micro/WIN 32功能。例如, 您将不能编辑程序, 除非重新关闭程序状态。其他功能, 例如, 在程序编辑器之间改变视图会自动关闭程序状态, 如果您希望继续查看状态, 必须重新选择 (再次打开) 程序状态选项。


**故障排除** 如果打开程序状态时遇到故障, 请记住下列前提条件:

- 您必须首先成功地建立通讯 (以便下载程序)。
- 您必须为下载程序选择正确的PLC型号。
- 您必须成功地编译程序。
- 您必须成功地下载程序。
- 欲查看连续状态更新, PLC必须处于“运行”模式。否则, 您只能看到I/O变化 (如果有)。因为尚未执行PLC程序, 对I/O所作的变动对程序状态显示中程序逻辑不会产生您预料的影响。如果您处于尚未执行的程序区域内 (例如子程序、中断例行程序或由于JMP指令越过的区域), 则没有状态显示, 因为代码未被扫描。

**理解程序状态显示** LAD中打开程序状态时, 如果操作数值等于1 (位打开), 布尔指令被显示为彩色块。在操作数的旁边 (或操作数的位置) 显示其他操作数的当前数据值, 并随着从PLC读取的数据不断更新显示。

定图标  表示该数值已经被“明确地”或直接地强制为当前正在显示的数值。

色隐去锁定图标  表示该数值已经被“隐含地”强制, 即不对地址进行直接强制, 但内存区域落入另一个被明确地强制的较大数值内。例如, 如果您明确地强制VW0, 则VB0和VB1被隐含地强制, 因为它们包含在VW0内。

块图标  表示数值被“部分地”强制。例如, 如果您明确地强制VB1, 则VW0被部分地强制, 因为其中的一个字节VB1被强制。

**程序状态限制** 从程序编辑器窗口查看状态的优势是您能够得到有关程序当前运行状况的图形

显示。您可以从程序状态强制和取消强制数值，但您不能使用状态图中提供的所有工具（例如单次读取、全部写入、读取全部强制功能以及向您希望强制操作数键入精确数值的能力）。

调试或监控程序时，理解程序状态的限制非常重要。

#### 强制数值时的通讯延迟

由于PLC与您观察程序状态所用的编程设备之间的通讯时间延迟，您看到的操作数显示数值总是首先在PLC中发生变化，然后再在显示中发生变化。虽然更新显示所需的时间不长，但您可能对操作数发出“强制”命令，其数值已经在PLC中实际发生变化，但未在程序状态显示中发生变化。

#### 提示：

- 强制操作数时，请始终观察程序状态显示，直至出现锁定图标。一旦出现锁定图标，您就可以确认数值已经被正确强制（否则您会看到被强制的数值，并进行修改）。
- 如果您希望强制数值不断波动的操作数，请使用状态图。这样，您无需尝试从不断变化的显示“捕捉”数值，可以直接在“新数值”列键入所需的数值，然后发出“强制”命令。

“电流”着色并不总是表示电源流动！因为LAD中的程序状态仅报告扫描结尾的数值，有时不太容易解释“电流”显示的真正含义。布尔触点和线圈根据其位操数值在LAD程序状态中着色。如果位值等于1（位打开），指令接受色彩。但是，这并不意味已经执行指令。有几种情况可能引起容易造成误解的电流显示：

- 检查状态时，如果PLC处于“停止”模式，可能已启用触点，但尚未打开线圈和方框，因为程序尚未执行。
- 如果您已经发出跳转指令，您正在检查的网络可能不会显示您预期的结果，因为PLC在程序执行过程中实际上略过这些结果。
- 与此相似，检查子程序时，可能启动布尔操作数，但只有在激活子程序时才能执行子程序逻辑。如果程序尚未调用子程序，则无论指令位值如何，在这些网络中都不会执行逻辑。

欲从LAD更改为STL，只需选择菜单命令 查看 > STL (View>STL)，然后再选择 调试 > 程序状态 (Debug>Program Status) 重新打开程序状态。

完成后，选择 查看 > 梯形图 (View>Ladder) 返回LAD视图。

#### 在程序状态中强制或取消强制数值

强制功能盖写立即读取或立即写入指令。强制功能还盖写向“停止”模式过渡时配置进入某一指定数值的输出：如果CPU进入“停止”模式，输出被设为强制值，而不是配置数值。



#### 警告

在程序中强制数值时，在程序每次扫描时将操作数重设为该数值，与输入 / 输出条件或其他正常情况下对操作数值有影响的程序逻辑无关。强制可能导致程序操作无法预料，无法预料的程序操作可能导致人员伤亡和 / 或设备损坏。强制功能是调试程序的辅助工具，切勿为了弥补处理设备的故障而进行强制。仅限合格人员使用强制功能。强制程序数值时，务必通知所有授权维修或调试程序的人员。

## 如何强制和取消强制数值

强制操作数：

- 直接单击操作数（不是指令），然后从“调试”工具条单击“强制”图标。
- 直接用鼠标右键单击操作数（不是指令），并从弹出菜单选择“强制”。

对单个操作数取消强制：

- 直接单击操作数（不是指令），然后从“调试”工具条单击“取消强制”图标。
- 直接用鼠标右键单击操作数（不是指令），并从弹出菜单中选择“取消强制”。

对全部强制数值取消强制：

- 直接单击操作数（不是指令），然后从“调试”工具条单击“全部取消强制”图标。
- 直接用鼠标右键单击操作数（不是指令），并从弹出菜单选择“全部取消强制”。

仅限从状态图提供“读取全部强制”选项，不能从程序状态提供。

**注释：**程序状态允许您将操作数强制为当前数值，而图状态则允许您在“新数值”列键入所需的数值。请记住PLC收到强制命令时，会强制操作数为当前数值。这是一个快速但并非瞬时完成的程序。在您单击 STEP 7-Micro/WIN 32中的状态显示和PLC收到强制命令之间，操作数值可能发生变化。一旦在操作数旁边出现锁定图标，您可以看到进入PLC的强制数值。如果强制数值有误，您可以利用状态图在“新数值”列键入您希望使用的正确数值，并从图状态强制操作数。

## 可被强制

并非所有内存地址均可被强制，下列内存区域的内存区域可被强制：

- 任何或所有数字输入和输出（DI、DQ）
- 多达16个内存数值（V或M）和 / 或模拟输入和输出（AI、AQ）字或双字为单位强制V内存或M内存。在偶数编号字节界限上（例如AIW6或AIW14）仅以字为单位强制模拟数值。如果您对VD0（包含VB0、VB1、VB2和VB3）进行强制，则计作可强制的16个内存数值之一。如果您单独强制 VB0、VB1、VB2和VB3，则计作可强制的16个内存数值中的4个。所有强制数值均存储在CPU的永久EEPROM内存中。

## 扫描循环

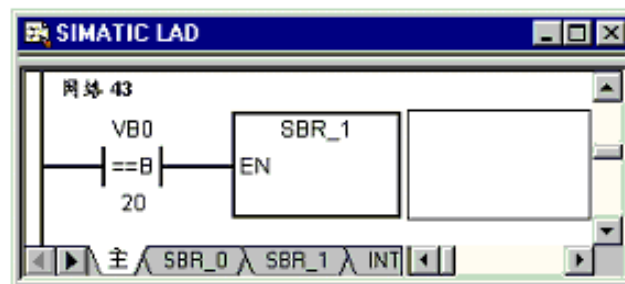
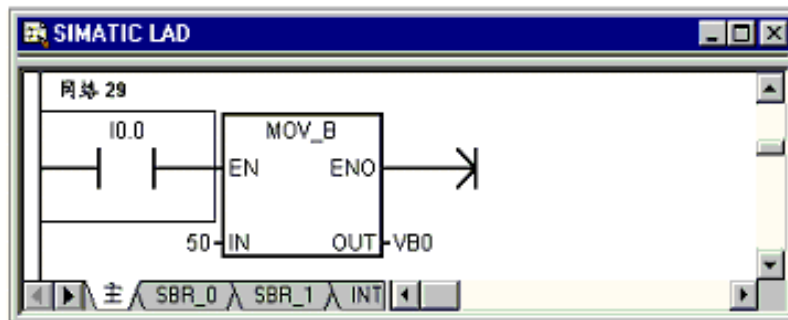
为可能在扫描循环过程中（由程序、I/O更新循环或通讯处理循环）改变强制数据，CPU会对每次扫描循环重新应用强制数值。



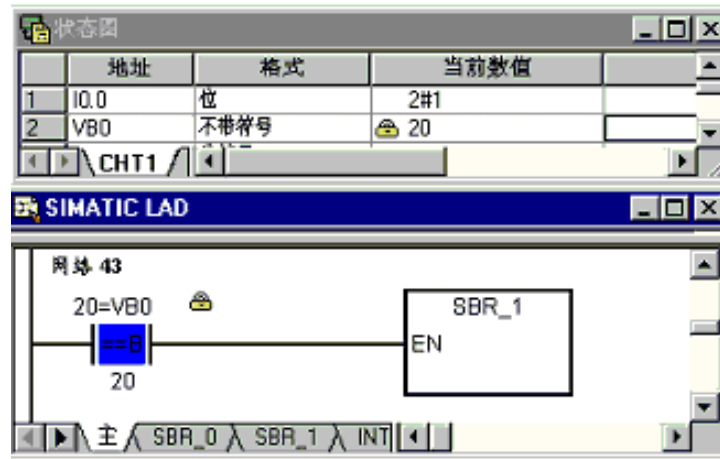
无法预测的  
强制数值程序操作

您需要理解如何对每种内存类型进行强制，才能有效地使用“强制”功能。

例如，您可能有类似以下的程序逻辑：



如果您正在查看该程序的Network 43，而且不记得在程序的其他部如何处理VB0，您可能认为为了激活SBR1可以将VB0强制为20。但请注意Network 29 MOV\_B指令将VB0数值设为50。PLC评估Network 43时，VB0包含数值50，所以不会激活SBR1。即您创建状态图，为VB0键入新数值20，并应用“强制”命令，有在评估程序逻辑后，才能在扫描结尾将VB0数值设为20。VB0 数值，在评估程序逻辑后才对V内存地址应用强制命令。



因为LAD程序状态显示扫描结尾数值，它显示的V0为强制数值20。但是当PLC评估Network 43时，V0不包含强制数值；因此强制V0并不能对SBR1产生预期的影响。

如果调试程序时发现带有强制操作数的网络不按预期设定执行，可使用编辑 > 查找（Edit>Find）命令寻找可能引用并改变操作数值的其他网络。

## 调节程序

使用菜单命令工具 > 选项（Tools>Options）并选择“状态”标签状态显示修改下列设定值：

- 缩放因素：编辑“缩放”设定值。  
 捷径：为了快速从程序状态调整“缩放”设定值，请使用组合键。  
 按CTRL键和数字小键盘上的加号，增大显示尺寸，按CTRL键和数字小键盘上的减号，缩小显示尺寸。
- 单元宽度和高度：编辑“栅格”设定值。  
 您可能需要增加单元宽度，以便显示更多未经截断的信息。  
 您可能希望降低单元高度，以便在屏幕上显示更多的网络。
- 流：您可以改变用于显示何时启用布尔操作数（位值等于1）的色彩。
- 操作数显示：您可以在指令内外显示操作数；您也可以不显示操作数名称或地址即显示状态数值。

另外，您可以重新排列STEP 7-Micro/WIN 32中的其他窗口，为程序编辑器窗口提供更大空间（或者便于与其他窗口（例如，状态图、符号表或交叉引用窗口）一起并排查看。使用“查看”和“窗口”菜单指令，并用鼠标确定尺寸、拖曳窗口，按照您的要求排列全部窗口。

## 5.5. 如何执行有限次数的扫描通讯概述

**说明** 您可以指定PLC对程序执行有限次数扫描（从1次扫描到65,535次扫描）。通过选择PLC运行的扫描次数，您可以在程序改变进程变量时对其进行监控。

第一次扫描时，SM0.1数值为1（打开）。

### 执行单次扫描

1. PLC必须处于“停止”模式。如果不是已经处于“停止”模式，将PLC转换成“停止”模式。
2. 从菜单条选择调试 > 第一次扫描。


### 执行多次扫描

1. PLC须处于“停止”模式。如果不是已经处于“停止”模式，将PLC转换成“停止”模式。
2. 欲执行多次扫描，从菜单条选择调试 > 多次扫描。

出现“执行扫描”对话框。



3. 输入所需的扫描次数数值，单击“确定”，确认您的选择并取消对话框。

**注释：** 当您准备好恢复正常程序操作时，请务必将PLC转回“运行”模式（单击工具条中的“运行”  按钮或选择PLC > 运行）。



## 6. 如何管理项目

### 6.1. 如何打印

**预览** 打印前可检视项目。选择菜单命令文件>打印预览（File>Print Preview）。

打印预览对话框提供下列选项：

- 一次检视一页程序或并排的两页程序。
- 改变POU标签，检视不同的程序块。
- 前进至下一页；后退至前一页。
- 预览时增加（放大）或减少（缩小）程序显示。

**主要打印选项** 选择菜单命令文件>打印（File>Print），存取主要“打印”对话框。

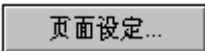
主要打印对话框提供下列选项：

- 选择打印机。
- 彩色或黑白打印。
- 打印单个项目组件内的一系列网络或行。
- 打印多个项目组件。

**打印选项** 从主打印对话框，选择选项  按钮，存取“打印选项”对话框。

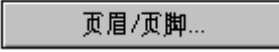
“打印选项”对话框提供下列选项：

- 打印与项目相关的属性信息。
- 打印数据块属性信息。
- 为STL项目打印行数。
- 为LAD项目调整列数（打印输出宽度）。
- 打印LAD项目的网络列。

**页面设定选项** 选择菜单命令文件>页面设定（File>Page Setup）或在主打印对话框内单击“页面设定”  按钮，存取“页面设定”对话框。

“页面设定”对话框提供下列选项：

- 调整打印输出的左、右、上、下边缘。
- 调整纸张方向（纵向或横向）。
- 选择纸张尺寸。
- 选择纸张来源。

**页首 / 页脚选项** 在“页面设定”对话框中，单击“页首 / 页脚”  按钮，存取“页首 / 页脚”对话框。

“页首 / 页脚”对话框允许您将下列项目放置在页首或页脚中：

- 项目名称
- 对象名称（例如POU、符号表、状态图或数据块）
- 日期
- 时间
- 页号
- 用户文本

您还可以决定页首或页脚项目是否应当左齐、右齐或居中，您可以为项目页脚创建注释框。

## 6.2. 如何移动、拷贝、重新命名和邮寄项目




### 说明

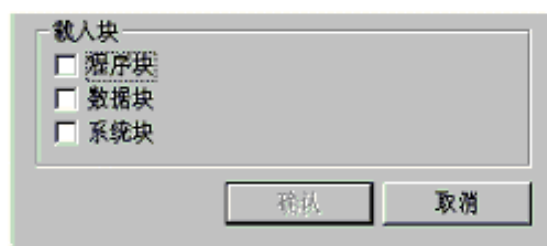
在STEP 7-Micro/WIN 32中，项目包含在带“.mwp”扩展名的单个文件内。使您能够利用标准Windows Explorer程序和电子邮件移动、拷贝、重新命名或邮寄项目。您还可以使用项目文件内的“另存为”对话框更改项目名称或目录位置。

### 6.3. 如何从PLC上载

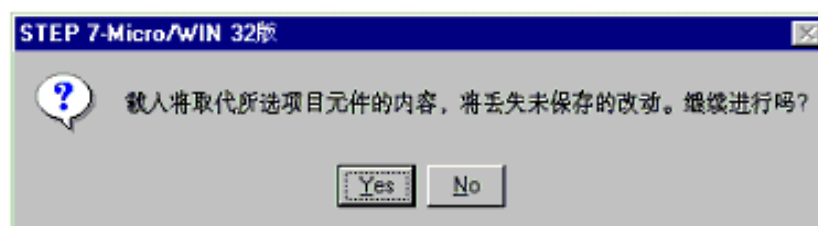
- 说明** 您可以使用工具条按钮或文件菜单，从PLC将程序上载至运行STEP 7-Micro/WIN 32的个人计算机中。
- 上载单个程序块** 可以上载程序块（OB1、子程序）、系统块和数据块；另外，您也可以仅上载二个程序块之一。PLC不包含符号或状态图信息；因此，您无法上载符号表或状态图。
- 上载至空置的新项目** 这是捕捉程序块、系统块和 / 或数据块信息的无风险方法。因为项目空置，所以无法反向损坏数据。如果您希望使用为该项目创建的状态图或符号表材料，您随时可以打开另一个STEP 7-Micro/WIN 32，并从其他项目文件拷贝该信息。
- 上载至现有项目** 如果您希望盖写自下载至PLC以来对程序进行的全部修改，这是一个好办法。如果您需要保留下载至PLC以来对程序块、系统块、和 / 或数据块所作的任何修改，则不应采用这种方法，因为上载将改写这些块内容。

#### 步骤

1. 打开STEP 7-Micro/WIN 32中的一个项目，容纳您将从PLC上载的程序块。
  - Ø 如果您希望上载至空置项目，选择文件>新或使用“新项目”工具条  按钮。
  - Ø 如果您希望上载至现有项目，选择文件>打开或使用“打开项目”工具条  按钮。
2. 选择文件>上载或使用“上载”工具条  按钮，初始化上载程序。
3. 上载”方框显示程序块、数据块和系统块复选栏。请核实已选择您希望上载的程序块复选栏，并取消选择您不希望上载的任何程序块，然后单击“确认”按钮。



4. STEP 7-Micro/WIN 32显示下列警告：



单击“是”按钮。

从PLC成功地将程序块上载至您的个人计算机时，STEP 7-Micro/WIN 32 显示一则讯息。

## 6.4. 如何在程序之间拷贝程序段

### 说明

使用“编辑”菜单命令和鼠标或标准Microsoft键组合，例如，CTRL + A 全选，CTRL+ C 拷贝，CTRL + X 剪切和CTRL + V 粘贴。  
提示：粘贴材料时，将材料插入光标当前位置上方的行内或格内。

### 拷贝网络范围或POU

- 您可以拷贝LAD或STL程序编辑器中的网络范围或全部POU。
- 只需打开程序编辑器窗口底部的标签，即可将选项粘贴至程序的不同POU中。
- 您也可以将选项粘贴至不同的项目中。

**提示：**为每个剪切 / 拷贝 / 粘贴操作所选择的网络范围必须是连续的：必须互相接触。如果您希望从POU的不同部分拷贝几个网络或网络集，则必须对每个单独的网络选项重复进行剪切 / 拷贝和粘贴操作。

### 拷贝符号表或状态图信息

- 您可以从符号表拷贝符号名称或地址信息，并粘贴至状态图的地址列。
- 您可以从符号表或状态图拷贝全部信息列，并将其粘贴至Microsoft Excel工作单，或从Excel将适当的列信息粘贴至符号表或状态图。

### 拷贝数据块信息

- 您可以拷贝数据块信息，并将其粘贴至文本编辑器。
- 您可以从文本编辑器将信息粘贴至数据块。但是，如果信息违反语法规则，数据块编辑器会出示红色警告X。

### 局部变量表

您不能在局部变量表之间成块拷贝数据，因为每个表的只读L内存分配必须保持独特。

### 两个STEP 7-Micro / WIN 32项目之间拷贝

执行下列步骤：

1. 打开两个STEP 7-Micro/WIN 32。
2. 在第一个STEP 7-Micro/WIN 32内，使用 文件 > 打开，打开您希望拷贝的项目。
3. 选择材料并使用 CTRL + C 或 编辑 > 拷贝 进行拷贝。
4. 在第二个STEP 7-Micro/WIN 32内，使用 文件 > 打开 来打开您希望粘贴材料的项目。
5. 将光标放在适当的位置，使用 CTRL + V 或 编辑 > 粘贴 来粘贴材料。

**注释：**请记住在关闭项目之前保存作业。

### 使用单个STEP 7-Micro / WIN 32在项目之间拷贝

执行下列步骤：

1. 使用 文件 > 打开，打开您希望拷贝的第一个项目。
2. 选择材料并使用 CTRL + C 或 编辑 > 拷贝 进行拷贝。
3. 使用 文件 > 关闭 来关闭第一个项目。

4. 使用 文件 > 打开 来打开第二个项目，即您希望粘贴材料的项目。
5. 将光标放在适当的位置，使用CTRL + V或 编辑 > 粘贴 来粘贴材料。

**注释：**请记住在关闭项目之前保存作业。

## 6.5. 使用较早版本的STEP 7-Micro/WIN项目

### 说明

欲使用在STEP 7-Micro/WIN或STEP 7-Micro/DOS早期版本内创建的项目，单击“打开”  按钮，或选择文件>打开，并选择需要的项目。

### 注释：

- 由STEP 7-Micro/WIN或STEP 7-Micro/DOS 早期版本创建的项目可能包含一个或多个STEP 7-Micro/WIN 32、3.0 版或更高版本不支持的逻辑结构。欲成功打开项目，您必须使用创建该项目的较早版本程序包，并按照下列步骤重新保存项目。

1. 将程序编辑器转换成STL。
2. 关闭符号编址。
3. 保存项目文件。

完成上述步骤后，即可在3.0版或更高版本中成功地打开项目。

- 您不能使用“打开”命令打开位于PLC中的项目；项目文件必须位于您的个人计算机 / 编程设备中。
- 每个STEP-7 Micro/WIN 32只能打开一个项目。您必须运行两个STEP 7-Micro/WIN 32才能同时打开两个项目。

当两个版本打开时，您可以在两个项目之间拷贝和粘贴LAD程序元件和STL文本。



## 7. PLC 指令

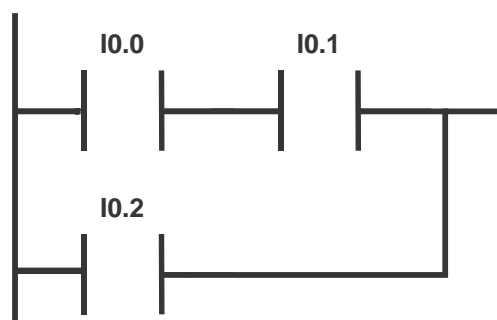
### 7.1. 位逻辑指令

#### 7.1.1. 常开接点

**说明** 可用常开接点（地址）指令检查指定的接点的信号状态。如果指定地址的信号状态是1的话，则接点是闭合，同时指令产生结果为1，如果指定地址的信号状态是0的话，则接点是打开，指令产生的结果为0。

表 7-1 常开接点（地址）元素和参数


LAD 元素	参数	数据类型	存储区	说明
	〈地址〉	BOOL	V, I, Q, M, SM, C, T	地址指明要检查信号状态的位

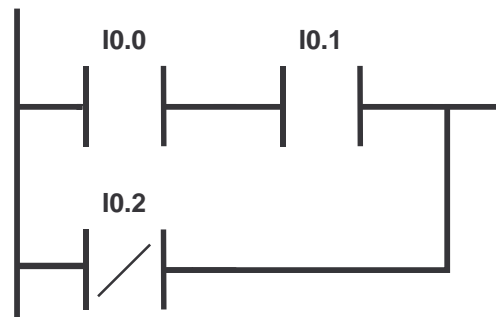


## 7.1.2. 常闭接点

**说明** 可用常闭接点（地址）指令检查指定地址的接点的信号状态。如果指定地址的信号状态是0的话，则接点是闭合的，同时指令产生结果是1。如果指定地址的信号状态是1的话，则接点是打开的，指令产生的结果是0。

表 7-2 常闭接点（地址）元素和参数

LAD 元素	参数	数据类型	存储区	说明
〈地址〉 	〈地址〉	BOOL	V, I, Q, M, SM, C, T	地址指明要检查信号状态的位



如果下列条件之一成立，则电流流通：

- 在输入 I0.0 和 I0.1 的信号状态为 1
- 或在输入 I0.2 的信号状态为 0



### 7.1.3. 输出线圈

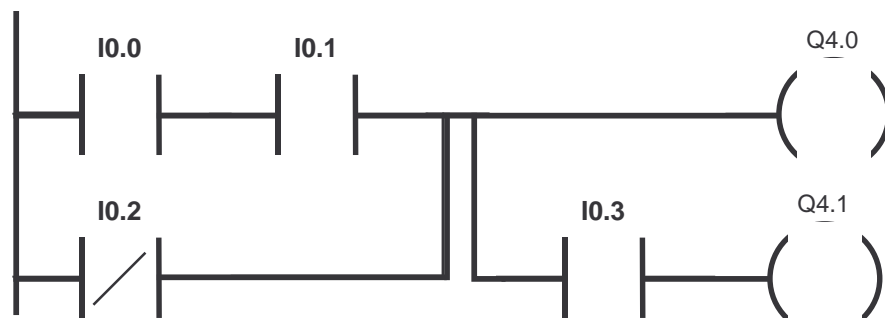
#### 说明

输出线圈指令象继电器逻辑图中的线圈一样作用，电路末端的线圈是通电还是不通电决定于下面的准则：

- 如果电流能够经电路到达线圈（即：电路的信号状态为1）的话，则线圈通电。
- 如果电流不能够经电路到达线圈（即：电路的信号状态为0）的话，则线圈不通电。

表 7-3 输出线圈元素和参数

LAD 元素	参数	数据类型	存储区	说明
〈地址〉 —( )	〈地址〉	BOOL	V, I, Q, M, C, T	地址表示要将逻辑符号串信号状态赋给的位



如果下列条件之一成立，则输出Q4.0的信号状态为1：

- 在输入I0.0和I0.1的信号状态为1
- 或在输入I0.2的信号状态为0

如果下列条件之一成立，则输出Q4.1的信号状态为1：

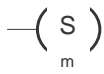
- 在输入I0.0, I0.1和I0.3的信号状态为1
- 或在输入I0.2的信号状态为0和输入I0.3信号状态为1

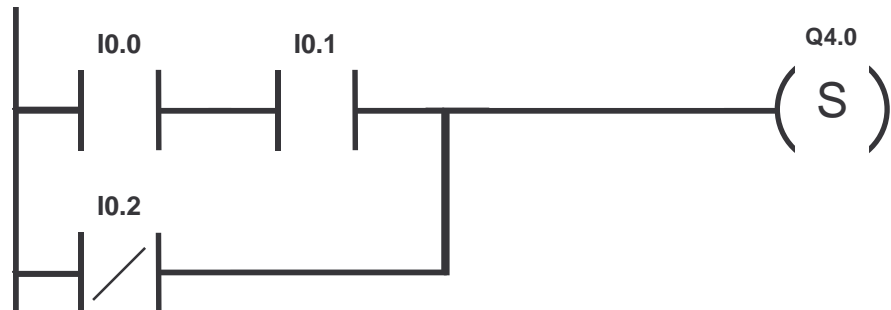
## 7.1.4. 置位线圈

## 说明

置位线圈指令在线圈前端有“1”信号或上升沿信号时，则这一指令将它指定地址的状态置“1”，如果在线圈前端有“0”信号时，则这一指令对它指定地址的状态没有影响，地址的状态保持不变。

表 7-4 置位线圈元素和参数

LAD 元素	参数	数据类型	存储区	说明
$\langle \text{地址} \rangle$ 	$\langle \text{地址} \rangle$ m=1	BOOL	V, I, Q, M, C, T	地址表示要置位的位



如果下列条件之一成立，则输出Q4.0的信号状态为1：

- 在输入I0.0和I0.1的信号状态为1
- 或在输入I0.2的信号状态为0

如果下列条件之一成立，则输出Q4.0的信号状态保持不变：

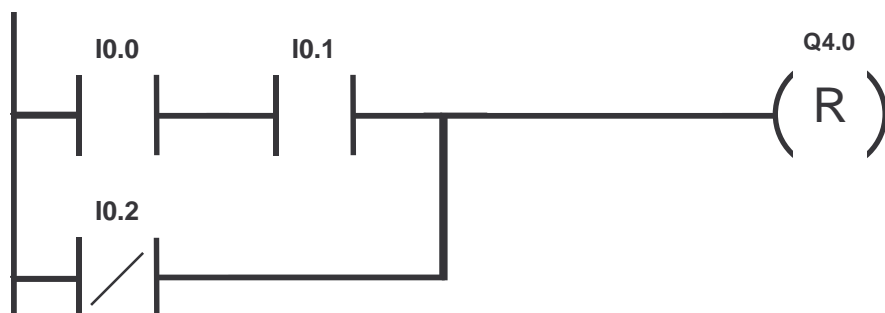
- 在输入I0.0, I0.1的信号状态为0
- 或在输入I0.2的信号状态为1

### 7.1.5. 复位线圈

**说明** 复位线圈指令在线圈前端有“1”信号或上升沿信号时，则这一指令将它指定地址的状态置“0”，如果在线圈前端有“0”信号时，则这一指令对它指定地址的状态没有影响，地址的状态保持不变。

表 7-5 置位线圈元素和参数

LAD 元素	参数	数据类型	存储区	说明
$\langle \text{地址} \rangle$ $\text{---} \left( \begin{array}{c} \text{R} \\ \text{m} \end{array} \right)$	$\langle \text{地址} \rangle$ $\text{m}=1$	BOOL	V, I, Q, M, C, T	地址表示要复位的位



如果下列条件之一成立，则输出Q4.0的信号状态为0：

- 在输入I0.0和I0.1的信号状态为1
- 或在输入I0.2的信号状态为0

如果下列条件之一成立，则输出Q4.0的信号状态保持不变：

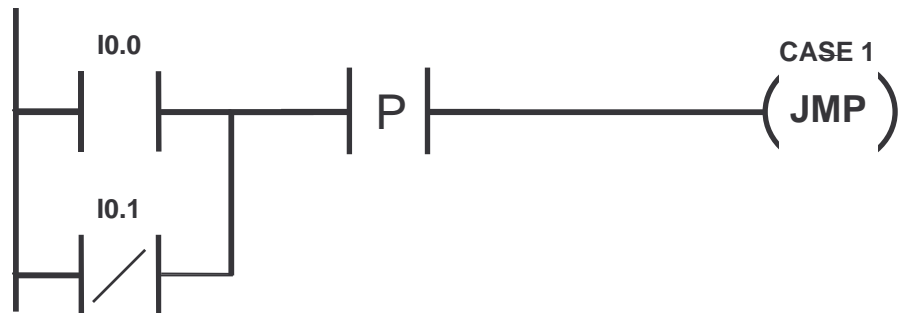
- 在输入I0.0, I0.1的信号状态为0
- 或在输入I0.2的信号状态为1

## 7.1.6. 上升沿检测接点

**说明** 上升沿检测接点允许为每次关至开转换一次扫描接通电流。

表 7-6 上升沿检测接点元素和参数

LAD 元素	参数	数据类型	存储区	说明
				在1个PLC循环周期内允许为每次关至开转换一次扫描接通电流



如果下列条件之一成立，则程序跳转至标号为CASE1处：

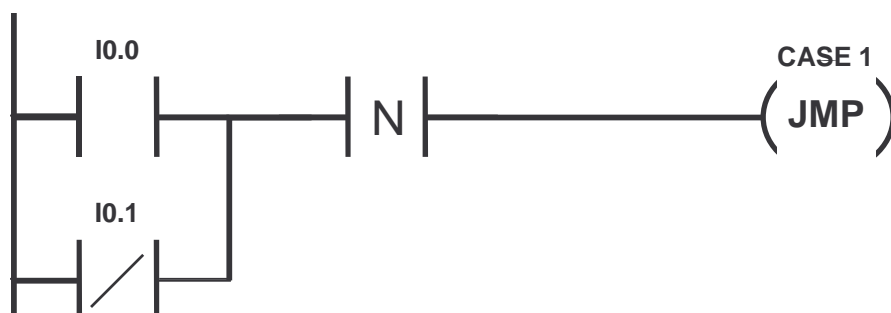
- 在输入I0.0的信号状态由0到1，即上升沿检测接点指令检测到一个上升沿信号
- 或在输入I0.1的信号状态为1到0，即上升沿检测接点指令检测到一个上升沿信号

### 7.1.7. 下降沿检测接点

**说明** 下降沿检测接点允许为每次开至关转换一次扫描接通电流。

表 7-7 下降沿检测接点元素和参数

LAD 元素	参数	数据类型	存储区	说明
				在1个PLC循环周期内允许为每次开至关转换一次扫描接通电流




如果下列条件之一成立，则程序跳转至标号为CASE1处：

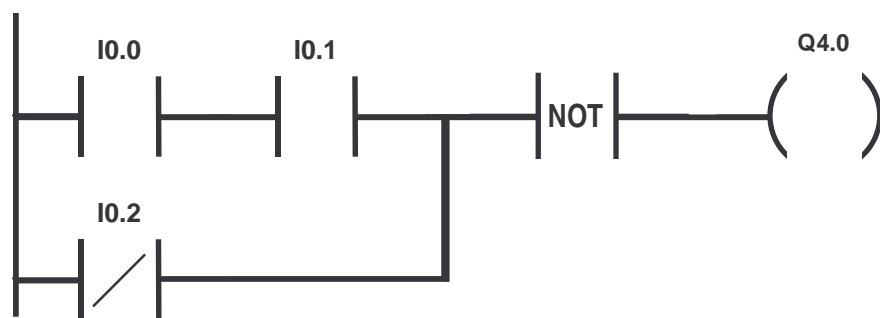
- 在输入I0.0的信号状态由0到1，即下降沿检测接点指令检测到一个下降沿信号
- 或在输入I0.1的信号状态为1到0，即下降沿检测接点指令检测到一个下降沿信号

## 7.1.8. NOT接点

**说明** NOT接点改变电流输入状态，当电流到达NOT接点时即停止。电流未到达NOT接点时，则供给电流。

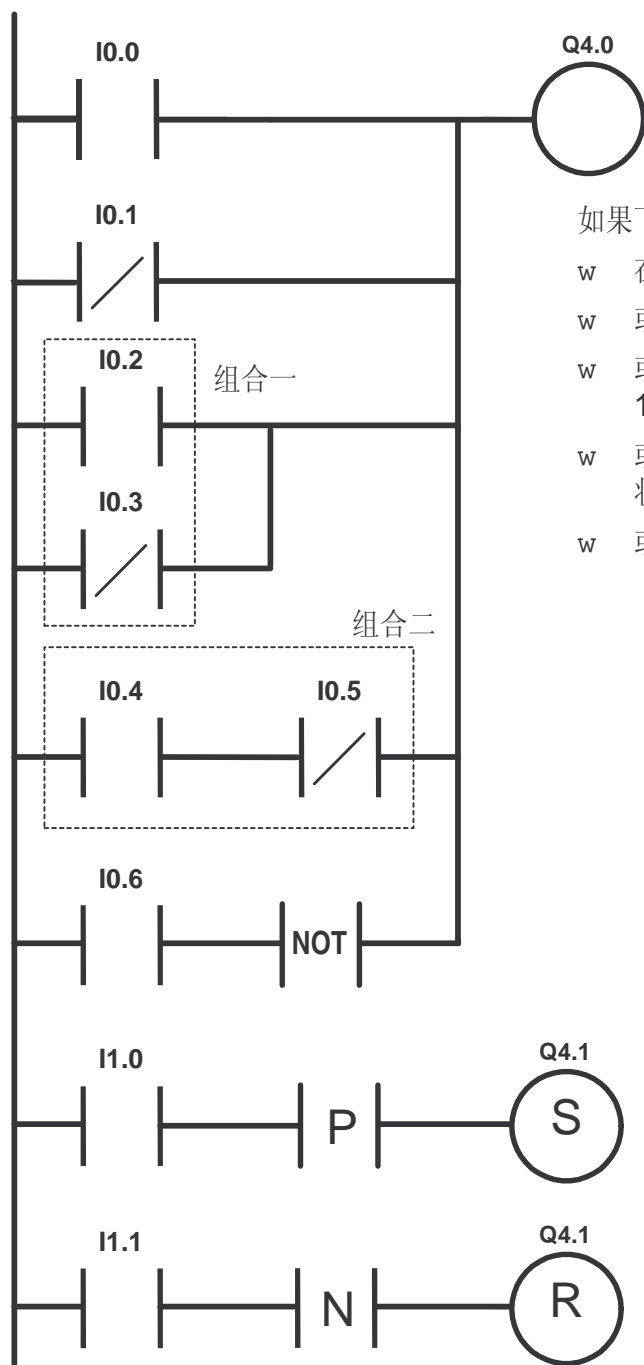
表 7-8 NOT接点元素和参数

LAD 元素	参数	数据类型	存储区	说明
				改变电流输入状态



如果下列条件之一成立，则输出Q4.0的信号状态为1：

- 在输入I0.0和I0.1的信号状态为0
- 或在输入I0.2的信号状态为1



如果下列条件之一，则输出 Q4.0 为1：

- w 在输入 I0.0 的信号状态为1
- w 或在输入 I0.1 的信号状态为0
- w 或在组合一中的 I0.2 的信号状态为1； I0.3 的信号状态为0
- w 或在组合二中的 I0.4和 I0.5 的信号状态同时为1和0
- w 或在 I0.6 的信号状态为0

如果 I1.0 的信号状态有0到1的变化，则 Q4.1 的信号状态将被置位成1的信号状态

如果 I1.1 的信号状态有1到0的变化，则 Q4.1 的信号状态将被复位成0的信号状态

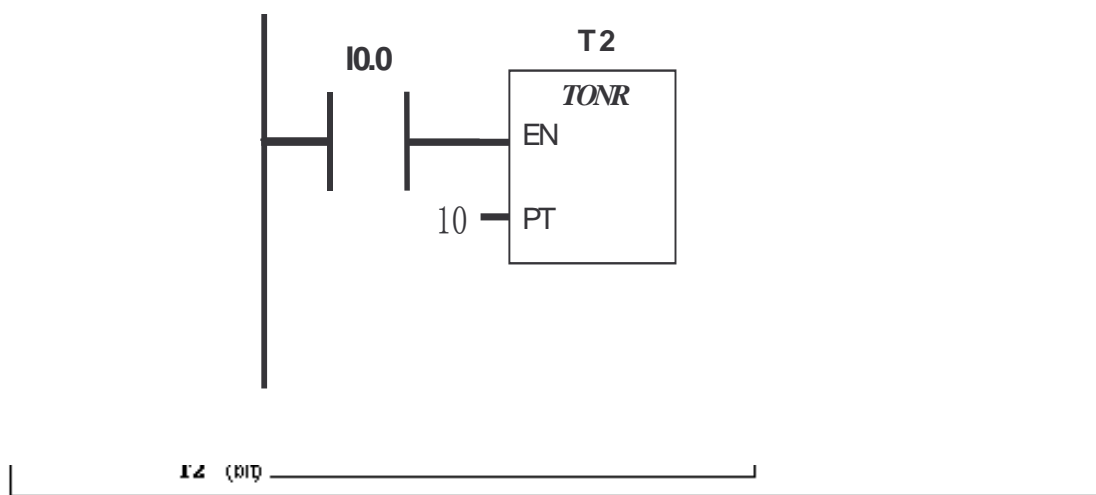
## 7.2. 定时器、计数器指令

### 7.2.1. 保留性打开延迟计时器 TONR

**说明** 保留性打开延迟计时器（TONR）指令在启用输入为ON时，开始计时。当前值（T<sub>xxx</sub>）大于或等于预设时间（PT）时，计时位为ON。当输入为OFF时，保存保留性延迟计时器当前值。您可使用保留性打开延迟计时器为多个输入ON期间累加时间。使用重设指令（R）清除保留性延迟计时器的当前值。达到预设值后，计时器继续计时，达到最大值32767时，停止计时。

表 7-9 保留性打开延迟计时器方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	T <sub>xxx</sub>	Timer	-	定时器标识号，范围0-15
	EN	BOOL	V, I, Q, M, SM, C, T	起动输入端
	PT	INT	VW, IW, QW, MW, AC, K	预置时间值



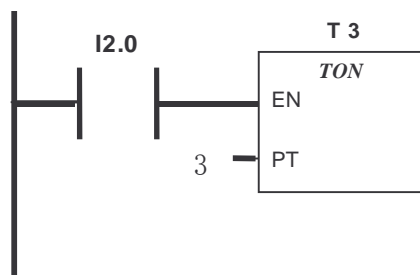


## 7.2.2. 延时定时器 TON

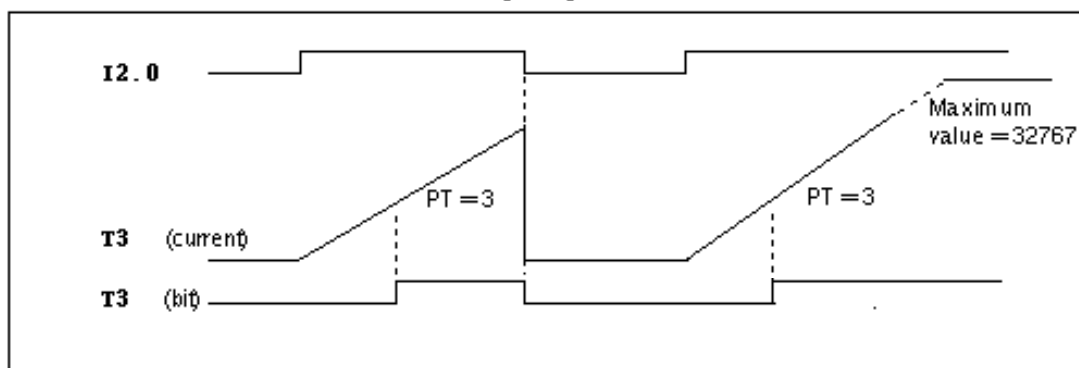
**说明** 延时定时器（TON）指令在启用输入为ON时，开始计时。当前值（Txxx）大于或等于预设时间（PT）时，计时位为ON。启用输入为OFF时，打开延时器当前值被清除。达到预设值后，计时器仍继续计时，达到最大值32767时，停止计时。

表 7-10 延时定时器方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	T <sub>xxx</sub>	Timer	-	定时器标识号，范围0-15
	EN	BOOL	V, I, Q, M, SM, C, T	起动输入端
	PT	INT	VW, IW, QW, MW, AC, K	预置时间值



Timing Diagram



### 7.2.3. 加计数器

**说明** 当加计数输入端（CTU）有上升沿输入时，计数器的当前值加1。当计数器的当前值大于预置值（PV）时，置位计数器标志。当前值达到最大允许值（32767）时，停止计数。当复位输入（R）接通时，计数器复位（当前值清零，标志位复位）。

表 7-11 加计数器方块图和参数

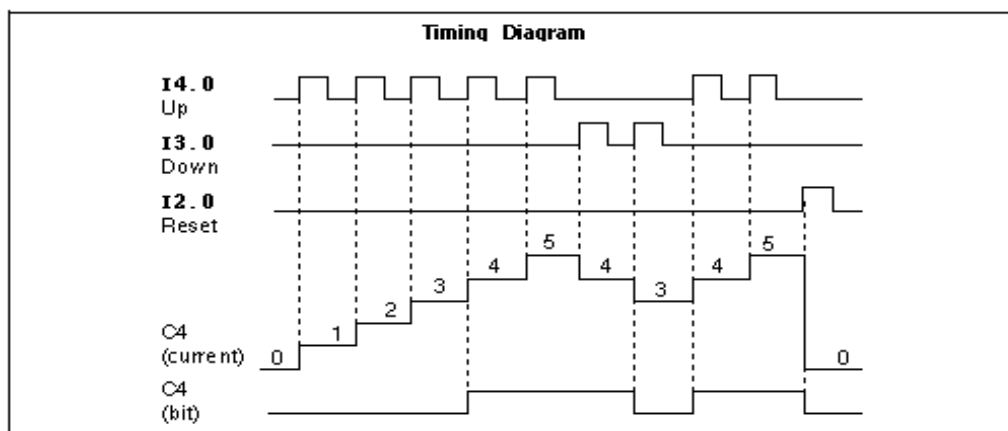
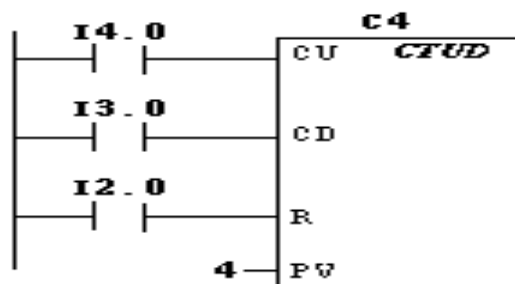
LAD 方块	参数	数据类型	存储区	说明
	<b>C<sub>xxx</sub></b>	Counter	-	计数器标识号，范围0-31
	CU	BOOL	V, I, Q, M, SM, C, T	起动输入端
	R	BOOL	V, I, Q, M, SM, C, T	复位输入端
	PV	INT	VW, IW, QW, MW, AC, K	预置计数值

### 7.2.4. 加减计数器CIUD

**说明** 当加计数输入端（CU）有上升沿输入时，加减计数器的当前值加1，当减计数输入端（CD）有上升沿输入时，计数器的当前值减1。当计数器的当前值大于预置值（PV）时，计数器标志位置位。当前值增至最大允许值（32767）时，不再增加；当前值减至最小允许值（-32768）时，不再减少。当复位输入（R）接通时，计数器复位（当前值清零，标志位复位）。

表 7-12 加减计数器方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	<b>C<sub>xxx</sub></b>	Counter	-	计数器标识号，范围0-31
	CU	BOOL	V, I, Q, M, SM, C, T	加计数起动输入端
	CD	BOOL	V, I, Q, M, SM, C, T	减计数起动输入端
	R	BOOL	V, I, Q, M, SM, C, T	复位输入端
	PV	INT	VW, IW, QW, MW, AC, K	预置计数值



### 7.3. 比较类指令

#### 7.3.1. 无符号字节比较

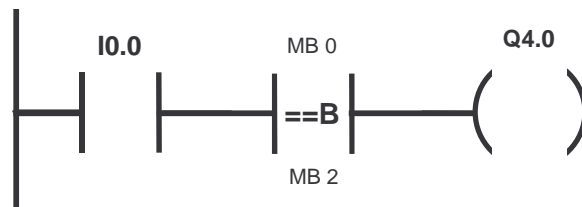
**说明** 比较字节指令用于比较两个数值：IN1与IN2。比较包括：IN1 = IN2、IN1 >= IN2、IN1 <= IN2。比较为真实时，接点打开。

表 7-13 无符号字节比较

比较类型	方块上的符号
IN1 等于 IN2	= =
IN1 大于等于 IN2	> =
IN1 小于等于 IN2	< =

表 7-14 无符号字节比较方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	IN 1	Byte	VB, IB, QB, MB, AC, K	第一个参与比较的字节
	IN 2	Byte	VB, IB, QB, MB, AC, K	第二个参与比较的字节



如果出现以下情况，则输出Q4.0的信号状态为1：

- 在输入I0.0的信号状态为0
- 且MWB0=MB2

### 7.3.2. 带符号字比较

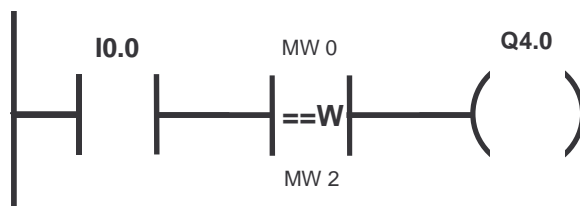
**说明** 带符号字比较指令用于比较两个数值：IN1与IN2。比较包括：IN1 = IN2、IN1 >= IN2、IN1 <= IN2。比较为真实时，接点打开。

表 7-15 带符号字比较

比较类型	方块上的符号
IN1 等于 IN2	==
IN1 大于等于 IN2	>=
IN1 小于等于 IN2	<=

表 7-16 带符号字方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	IN 1	WORD	VW, IW, QW, MW, AC, K	第一个参与比较的字
	IN 2	WORD	VW, IW, QW, MW, AC, K	第二个参与比较的字



如果出现以下情况，则输出Q4.0的信号状态为1：

- 在输入I0.0的信号状态为0
- 且MW0=MW2

### 7.3.3. 带符号双字比较

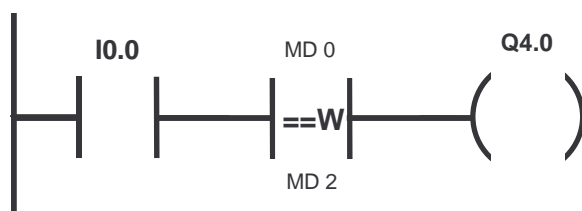
**说明** 带符号双字比较指令用于比较两个数值：IN1与IN2。比较包括：IN1 = IN2、IN1 >= IN2、IN1 <= IN2。比较为真实时，接点打开。

表 7-17 带符号双字比较

比较类型	方块上的符号
IN1 等于 IN2	==
IN1 大于等于 IN2	>=
IN1 小于等于 IN2	<=

表 7-18 带符号双字方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	IN 1	DWORD	VD, ID, QD, MD, AC, K	第一个参与比较的双字
	IN 2	DWORD	VD, ID, QD, MD, AC, K	第二个参与比较的双字



如果出现以下情况，则输出Q4.0的信号状态为1：

- 在输入I0.0的信号状态为0
- 且MD0=MD2

### 7.3.4. 带符号实数比较

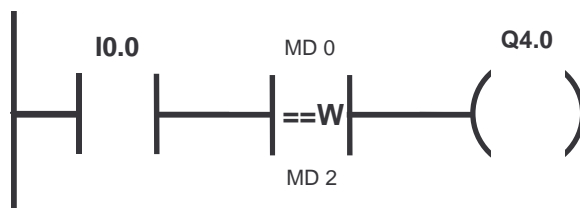
**说明** 比较实数指令用于比较两个数值：IN1与IN2。  
比较包括：IN1 = IN2、IN1 >= IN2、IN1 <= IN2。比较为真实时，接点打开。

表 7-19 带符号实数比较

比较类型	方块上的符号
IN1 等于 IN2	==
IN1 大于等于 IN2	>=
IN1 小于等于 IN2	<=

表 7-20 带符号实数方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	IN 1	DWORD	VD, ID, QD, MD, AC, K	第一个参与比较的实数
	IN 2	DWORD	VD, ID, QD, MD, AC, K	第二个参与比较的实数



如果出现以下情况，则输出Q4.0的信号状态为1：

- 在输入I0.0的信号状态为0
- 且MD0=MD2

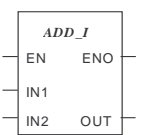
## 7.4. 算术类指令

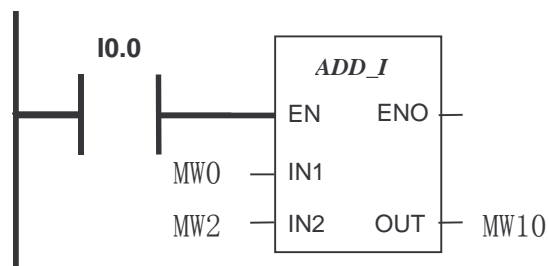
### 7.4.1. 字加减法

**说明** 字加法和字减法指令在EN=1的状态下，将两个16位字相加或相减，并产生一个16位的结果（OUT）。

在LAD中： $IN1 + IN2 = OUT$ ， $IN1 - IN2 = OUT$ 。

表 7-21 字加减法方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN1	WORD	VW, IW, QW, MW, AC, T, C, K	相加减的第一个输入
	IN2	WORD	VW, IW, QW, MW, AC, T, C, K	相加减的第二个输入
	OUT	WORD	VW, QW, MW, T, C, AC	相加减的结果



输入I0.0的1信号状态激活ADD\_I方块，  
MW0+MW2的结果放入存储器字MW10



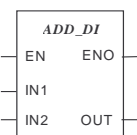
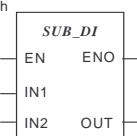
## 7.4.2. 双字加减法

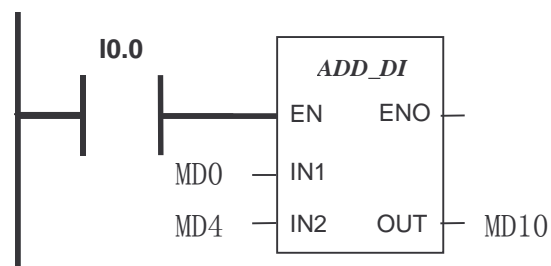
### 说明

双字加法和字减法指令在EN=1的状态下，将两个32位字相加或相减，并产生一个32位的结果（OUT）。

在LAD中：IN1 + IN2 = OUT，IN1 - IN2 = OUT

表 7-22 双字加减法方块图和参数

LAD 方块	参数	数据类型	存储区	说明
 <p>ADD_DI</p> <p>EN ENO</p> <p>IN1 IN2 OUT</p>	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
 <p>SUB_DI</p> <p>EN ENO</p> <p>IN1 IN2 OUT</p>	IN1	DWORD	VD, ID, QD, MD, AC, T, C, K	相加减的第一个输入
	IN2	DWORD	VD, ID, QD, MD, AC, T, C, K	相加减的第二个输入
	OUT	DWORD	VD, QD, MD, T, C, AC	相加减的结果



输入I0.0的1信号状态激活ADD\_DI方块，  
MD0+MD4的结果放入存储器字MD10

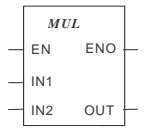
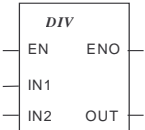
## 7.4.3. 字乘法

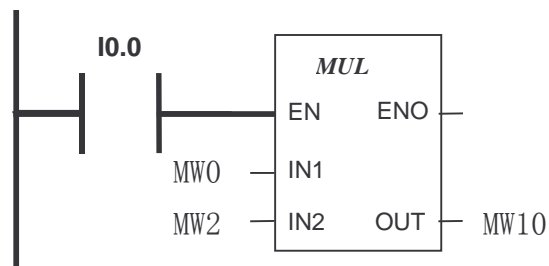
## 说明

在EN=1的状态下，字乘法指令将两个16位字相乘，并产生一个16位乘积。  
字除法指令将两个16位字相除，并产生一个16位商，不保留余数。

在LAD中：IN1 \* IN2 = OUT，IN1 / IN2 = OUT

表 7-23 字乘法除法方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN1	WORD	VW, IW, QW, MW, AC, T, C, K	相乘除的第一个输入
	IN2	WORD	VW, IW, QW, MW, AC, T, C, K	相乘除的第二个输入
	OUT	DWORD	VD, QD, MD, AC	相乘除的结果



输入I0.0的1信号状态激活MUL方块，  
MW0+MW2的结果放入存储器字MW10

### 7.4.4. 浮点加减法

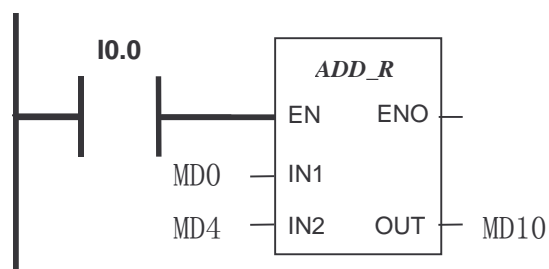
#### 说明

在EN=1的状态下，浮点加法和浮点减法指令将两个32位实数相加或相减，并产生一个32位浮点结果（OUT）。

在LAD中：IN1 \* IN2 = OUT，IN1 / IN2 = OUT

表 7-24 浮点加减法方块图和参数

LAD 方块	参数	数据类型	存储区	说明
 <p>ADD_R</p> <p>EN ENO</p> <p>IN1</p> <p>IN2 OUT</p>	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
 <p>SUB_R</p> <p>EN ENO</p> <p>IN1</p> <p>IN2 OUT</p>	IN1	Real	VD, ID, QD, MD, AC, T, C, K	相加减的第一个输入
	IN2	Real	D, ID, QD, MD, AC, T, C, K	相加减的第二个输入
	OUT	Real	VD, QD, MD, AC	相加减的结果



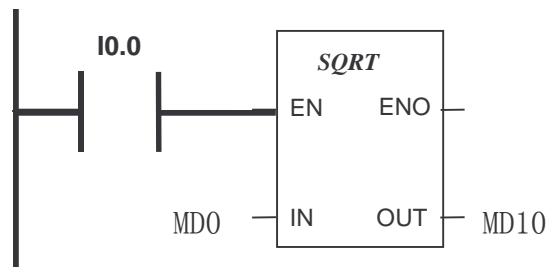
输入I0.0的1信号状态激活ADD\_R方块，  
MD0+MD4的结果放入存储器字MD10

### 7.4.5. 平方根

说明 在EN=1的状态下，平方根指令对32位实数（IN）取平方根，并产生一个32位实数结果（OUT），如以下等式所示： $\sqrt{IN}$

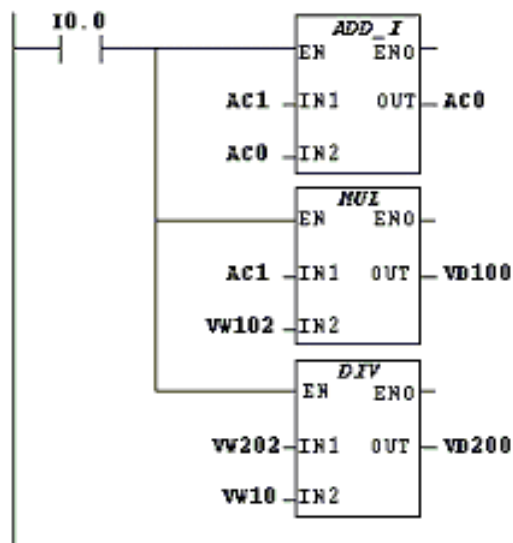
表 7-25 平方根方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	Real	VD, ID, QD, MD, AC, K	数据
	OUT	Real	VD, QD, MD, AC	数据的平方根



输入I0.0的1信号状态激活SQRT方块，MD0的平方根值存在双字存储器字MD10中

7.4.6. 举例



加法指令	乘法指令	除法指令
$\begin{array}{r} \text{AC1} \quad 4000 \\ + \\ \text{AC0} \quad 6000 \\ \hline = \\ \text{AC0} \quad 10000 \end{array}$	$\begin{array}{r} \text{AC1} \quad 4000 \\ * \\ \text{VD100} \quad 200 \\ \hline = \\ \text{VD100} \quad 800000 \end{array}$	$\begin{array}{r} \text{VD200} \quad 4000 \\ / \\ \text{VW10} \quad 41 \\ \hline = \\ \text{VD200} \quad 23 \quad 97 \\ \text{余数} \quad \text{商} \\ \text{VW202} \quad \text{VW200} \end{array}$
<p>注：VD100 包含 VW100 和 VW102 VD200 包含 VW200 和 VW202</p>		

## 7.5. 逻辑类指令

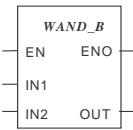
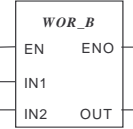
### 7.5.1. 字节与或异或

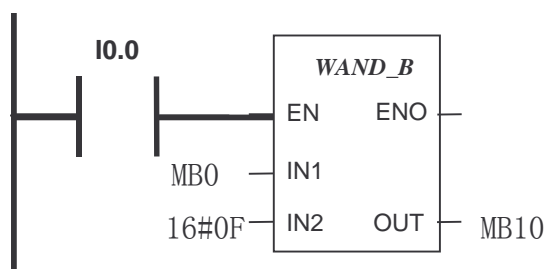
**说明**                      字节AND指令对两个输入字节的对应位进行AND操作，并用一个字节载入结果（OUT）。

字节OR指令对两个输入字节的对应位进行OR操作，并用一个字节载入结果（OUT）。

字节异OR指令对两个输入字节的对应位进行XOR操作，并用一个字节载入结果（OUT）。

表 7-26                      字节与或异或方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN1	BYTE	VB, IB, QB, MB, AC, K	逻辑操作的第一个输入
	IN2	BYTE	VB, IB, QW, MW, AC, K	逻辑操作的第二个输入
	OUT	BYTE	VB, QB, MB, AC	逻辑操作的结果

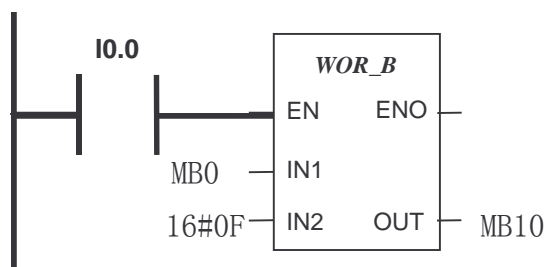


输入I0.0的1信号状态激活WAND\_B方块，只有位0至位3是重要的，存储器MB0的其余位被屏蔽。

MB0 = 0101 0101

AND = 0000 1111

MB10 = 0000 0101

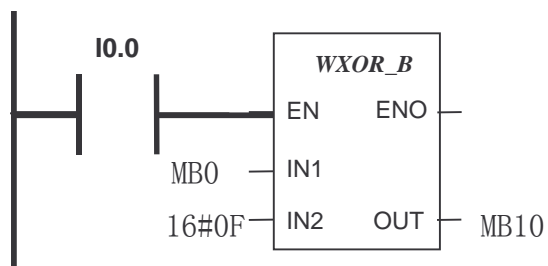


输入I0.0的1信号状态激活WOR\_B方块，位0至位3被置1，存储器MB0的其余位不变。

MB0 = 0101 0101

OR = 0000 1111

MB10 = 0000 1111



输入I0.0的1信号状态激活WXOR\_B方块，

MB0 = 0101 0101

XOR = 0000 1111

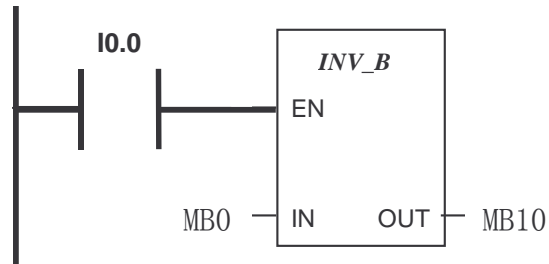
MB10 = 0101 1010

## 7.5.2. 字节非

**说明** 字节非指令对输入字节IN进行求反操作，并将结果载入字节数值OUT。

表 7-27 字节非方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	BYTE	VB, IB, QB, MB, AC, K	逻辑操作输入
	OUT	BYTE	VB, QB, MB, AC	逻辑操作的结果



输入10.0的1信号状态激活INV\_B方块，存储器MB0的位被取反。

MB0 = 0101 0101

MB10 = 1010 1010



### 7.5.3. 字与或异或

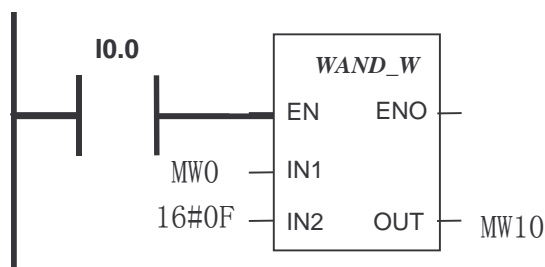
**说明** 字AND指令对两个输入字的对应位进行AND操作，并用一个字载入结果（OUT）。

字OR指令对两个输入字的对应位进行OR操作，并用一个字载入结果（OUT）。

字异OR指令对两个输入字的对应位进行XOR操作，并用一个字载入结果（OUT）。

表 7-28 字与或异或方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN1	WORD	VW, IW, QW, MW, AC, K	逻辑操作的第一个输入
	IN2	WORD	VW, IW, QW, MW, AC, K	逻辑操作的第二个输入
	OUT	WORD	VW, QW, MW, AC	逻辑操作的结果

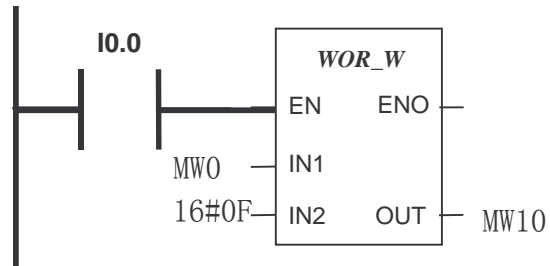


输入I0.0的1信号状态激活WAND\_W方块，只有位0至位3是重要的，存储器MWO的其余位被屏蔽。

MWO = 0101 0101 0101 0101

AND = 0000 0000 0000 1111

MW10 = 0000 0000 0000 0101

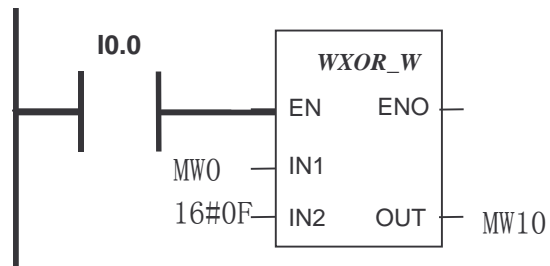


输入I0.0的1信号状态激活WOR\_W方块，位0至位3被置1，存储器MW0的其余位不变。

MW0 = 0101 01011 0101 0101

OR = 0000 0000 0000 1111

MW10 = 0101 0101 0101 1111



输入I0.0的1信号状态激活WXOR\_W方块，

MW0 = 0101 0101 0101 0101

XOR = 0000 0000 0000 1111

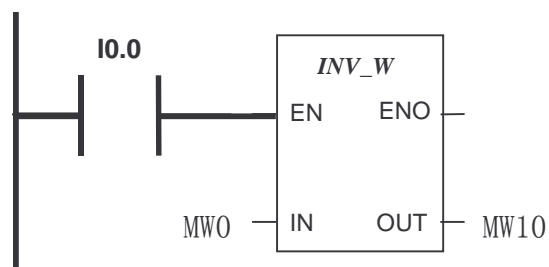
MW10 = 0101 0101 0101 1010

### 7.5.4. 字非

**说明** 字非指令对输入字IN进行求反操作，并将结果载入字数值OUT。

表 7-29 字非方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	WORD	VW, IW, QW, MW, AC, K	逻辑操作输入
	OUT	WORD	VW, QW, MW, AC	逻辑操作的结果



输入I0.0的1信号状态激活INV\_W方块，存储器MW0的位被取反。

MW0 = 0101 0101 0101 0101

MW10 = 1010 1010 1010 1010

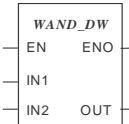
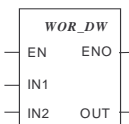
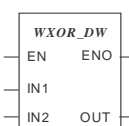
### 7.5.5. 双字与或异或

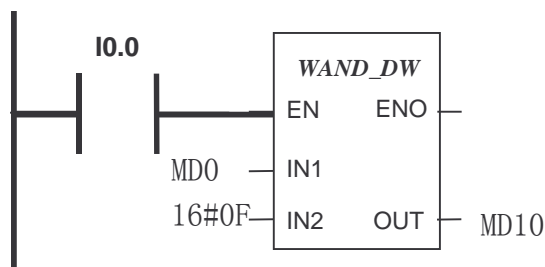
说明 双字AND指令对两个输入双字的对应位进行AND操作，并用一个双字载入结果（OUT）。

双字OR指令对两个输入双字的对应位进行OR操作，并用一个双字载入结果（OUT）。

双字异OR指令对两个输入双字的对应位进行XOR操作，并用一个双字载入结果（OUT）。

表 7-30 双字与或异或方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN1	DWORD	VD, ID, QD, MD, AC, K	逻辑操作的第一个输入
	IN2	DWORD	VD, ID, QD, MD, AC, K	逻辑操作的第二个输入
	OUT	DWORD	VD, QD, MD, AC	逻辑操作的结果

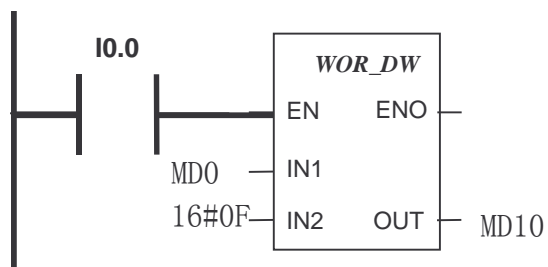


输入10.0的1信号状态激活WAND\_DW方块，只有位0至位3是重要的，存储器MD0的其余位被屏蔽。

MD0 = 0101 0101 0101 0101 0101 0101 0101 0101

AND = 0000 0000 0000 0000 0000 0000 0000 1111

MD10 = 0000 0000 0000 0000 0000 0000 0000 0101

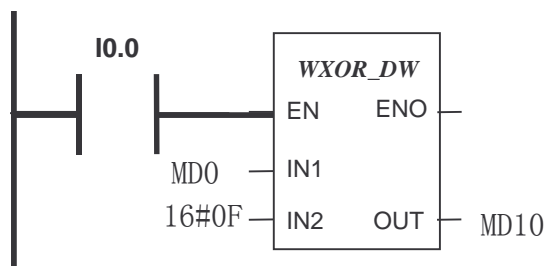


输入I0.0的1信号状态激活WOR\_DW方块，位0至位3被置1，存储器MD0的其余位不变。

MD0 = 0101 0101 0101 0101 0101 0101 0101 0101

OR = 0000 0000 0000 0000 0000 0000 0000 1111

MD10 = 0101 0101 0101 0101 0101 0101 0101 1111



输入I0.0的1信号状态激活WXOR\_DW方块，

MD0 = 0101 0101 0101 0101 0101 0101 0101 0101

XOR = 0000 0000 0000 0000 0000 0000 0000 1111

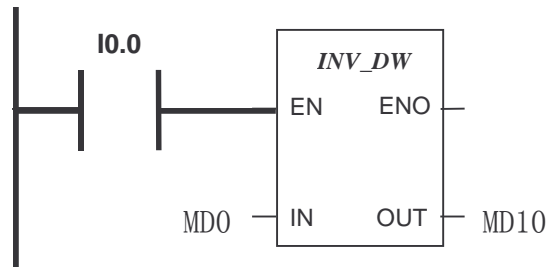
MD10 = 0101 0101 0101 0101 0101 0101 0101 1010

### 7.5.6. 双字非

**说明** 双字非指令对输入双字IN进行求反操作，并将结果载入双字数值OUT。

表 7-31 双字非方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	DWORD	VD, ID, QD, MD, AC, K	逻辑操作输入
	OUT	DWORD	VD, QD, MD, AC	逻辑操作的结果



输入I0.0的1信号状态激活INV\_DW方块，存储器M0的位被取反。

M0 = 0101 0101 0101 0101 0101 0101 0101 0101

MD10 = 1010 1010 1010 1010 1010 1010 1010 1010

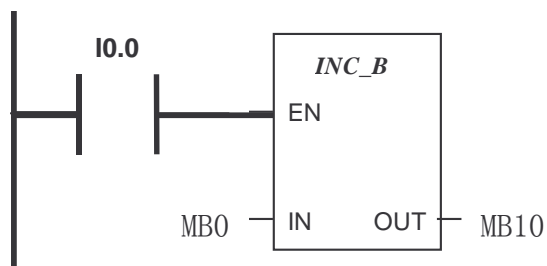
## 7.6. 逻辑类指令

### 7.6.1. 字节增减量

**说明** 字节增量和字节减量指令在输入字节（IN）上加1或减1，并将结果置入OUT指定的变量中。字节增量和减量指令不带符号。  
在LAD中：IN + 1 = OUT，IN - 1 = OUT

表 7-32 字节增减量方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	BYTE	VB, IB, QB, MB, AC, K	数据输入
	OUT	BYTE	VB, QB, MB, AC	数据结果



输入I0.0的1信号状态激活INC\_B方块，存储器MB0的值+1后存入MB10。

MB0 = 16#55, MB10 = 16#56

### 7.6.2. 字增减量

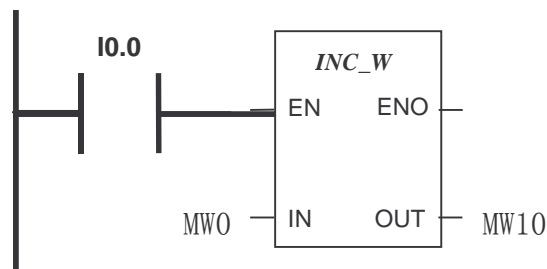
**说明**

字增量和字减量指令在输入字节（IN）上加1或减1，并将结果置入OUT指定的变量中。字节增量和减量指令不带符号。

在LAD中：IN + 1 = OUT，IN - 1 = OUT

表 7-33 字增减量方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	WORD	VW, IW, QW, MW, AC, K	数据输入
	OUT	WORD	VW, QW, MW, AC	数据结果



输入I0.0的1信号状态激活INC\_W方块，存储器MW0的值+1后存入MW10。

MW0 = 16#55， MW10 = 16#56

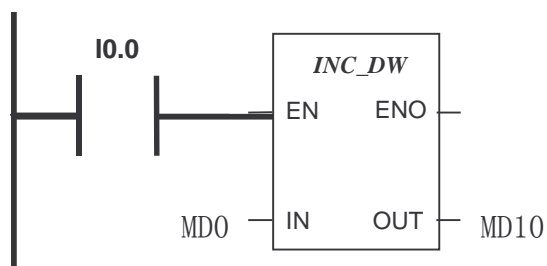


### 7.6.3. 双字增减量

说明 双字增量和双字减量指令在输入字节（IN）上加1或减1，并将结果置入OUT指定的变量中。双字增量和减量指令不带符号。  
在LAD中：IN + 1 = OUT，IN - 1 = OUT

表 7-34 双字增减量方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	DWORD	VD, ID, QD, MD, AC, K	数据输入
	OUT	DWORD	VD, QD, MD, AC	数据结果



输入I0.0的1信号状态激活INC\_DW方块，存储器MD0的值+1后存入MD10。  
MD0 = 16#55，MD10 = 16#56

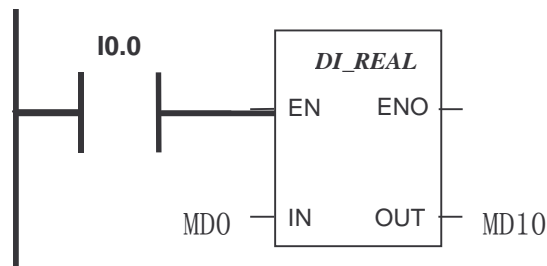
## 7.7. 数据转换类指令

### 7.7.1. 双整数到实数转换

**说明** 双整数到实数指令将32位带符号整数（IN）转换成32位实数，并将结果载入OUT指定的变量中。

表 7-35 双整数到实数转换方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	DWORD	VD, ID, QD, MD, AC, K	数据输入
	OUT	REAL	VD, QD, MD, AC	数据结果



输入I0.0的1信号状态激活DI\_REAL方块，读入存储器MD0的值，将其转换为一个浮点后存入MD10。

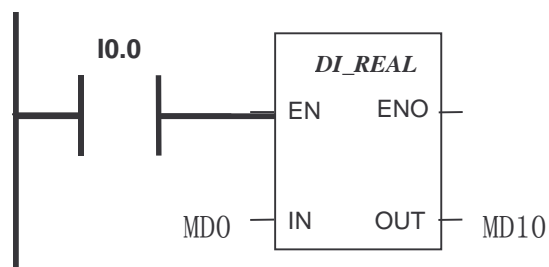
MD0 = 100, MD10 = 100.0

## 7.7.2. TRUNC指令

**说明** TRUNC指令将32位实数（IN）转换成32位带符号整数，并将结果载入OUT指定的变量中。只有实数的整数部分被转换，小数部分被丢弃。

表 7-36 TRUNC方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	REAL	VD, ID, QD, MD, AC, K	数据输入
	OUT	DWORD	VD, QD, MD, AC	数据结果



输入I0.0的1信号状态激活DI\_REAL方块，读入存储器MD0的值，将其转换为一个浮点后存入MD10。

MD0 = 100.6, MD10 = 100

## 7.8. 程序控制类指令

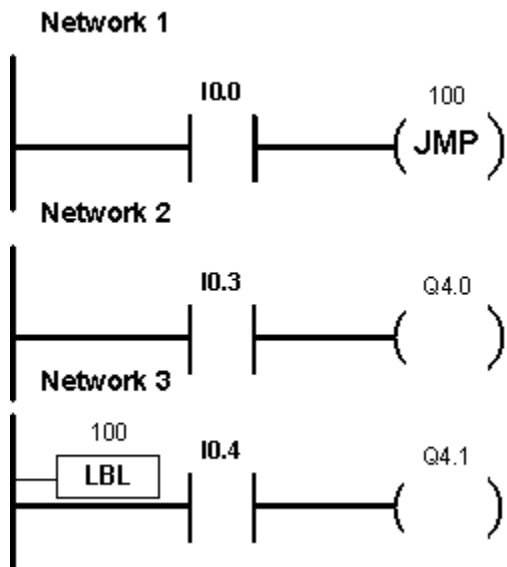
### 7.8.1. 跳转至标签和标签

**说明** 跳转至标签（JMP）指令对程序内的指定标签（n）进行分支。跳转完成时，堆栈顶值始终为逻辑1。  
 标签（LBL）指令标记跳转目的地（n）的位置。

跳转和对应的标签必须位于主程序、子程序或中断例行程序中。您不能从主程序跳转至子程序或中断例行程序中的标签，与此相似，您也不能从子程序或中断例行程序跳转至该子程序或中断例行程序之外的标签。

表 7-37 跳转至标签和标签方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	n	-	K	跳转至标签，范围（0-255）
	n	-	K	标签定义处，范围（0-255）



若I0.0=1,跳转至标签100处执行。

由于跳转，即使I0.3=1,Q4.0的输出也不执行

## 7.8.2. 子程序调用

### 说明

子程序调用指令将控制转移给子例行程序（n）。使用子例行程序指令时，有无参数均可。

每个子例行程序调用的输入 / 输出参数最大限制为16。如果您尝试下载的程序超过此一限制，会返回一则错误。

如果您为子例行程序指定一个符号名称，例如USR\_NAME，该符号名称会出现在指令树的子例行程序文件夹内。

将参数值指定给子例行程序中的局部内存时应遵守下列原则。

1. 参数值指定给局部内存的顺序由CALL指定，参数从L.0开始。
2. 一至八个连续位参数值被指定给从Lx.0开始持续至Lx.7的单字节。
3. 字节、字和双字数值被指定给局部内存时，以字节为界限（LBx、LWx或LDx）。

在带参数的子例行程序CALL指令中，参数必须与子例行程序局部变量表中定义的变量完全匹配。参数顺序必须以输入参数开始，其次是输入 / 输出参数，然后是输出参数。

表 7-38 子程序调用方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	SBR Name	-	-	子程序名称
	EN	-	-	子程序调用条件输入
	Pi1-Pin	-	-	参数输入
	Po1-Pom	-	-	参数输出
(RET)	-	-	-	条件/无条件子程序结束

## 7.9. 移位类指令

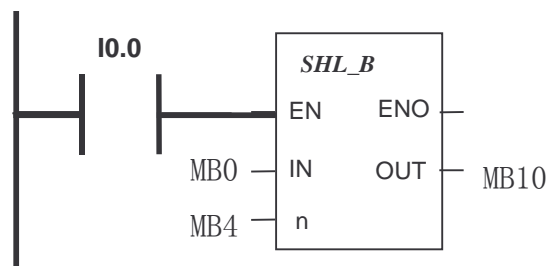
### 7.9.1. 字节左右移位

**说明** 字节向右移位和字节向左移位指令将输入字节（IN）数值向右或向左移动N位，并将结果载入输出字节（OUT）。移位指令对每个移出位补0。如果移位数目（N）大于或等于8，则数值最多被移位8次。

字节向右和向左移位操作不带符号。

表 7-39 字节左右移位方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	IN	BYTE	VB, IB, QB, MB, AC, K	移位输入数据
	N	BYTE	VB, IB, QB, MB, AC, K	移位输入位数
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	OUT	BYTE	VB, QB, MB, AC	数据输出



输入I0.0的1信号状态激活SHL\_B方块，将存储器MB0中的值左移，左移位数由存储器MB4中给出，结果存入MB10

## 7.9.2. 字左右移位

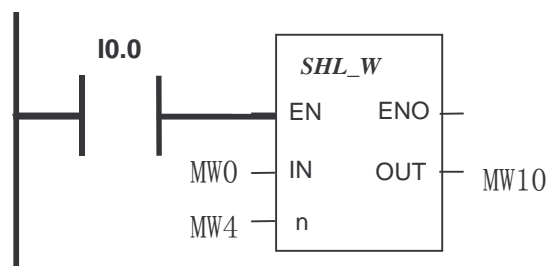
### 说明

字向右移位和字向左移位指令将输入字（IN）数值向右或向左移动N位，并将结果载入输出字（OUT）。移位指令对每个移出位补0。如果移位数目（N）大于或等于16，则数值最多被移位16次。

字向右和向左移位操作带符号。

表 7-40 字左右移位方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	IN	WORD	VW, IW, QW, MW, AC, K	移位输入数据
	N	WORD	VW, IW, QW, MW, AC, K	移位输入位数
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	OUT	WORD	VW, QW, MW, AC	数据输出



输入I0.0的1信号状态激活SHL\_W方块，将存储器MW0中的值左移，左移位数由存储器MW4中给出，结果存入MW10

### 7.9.3. 双字左右移位

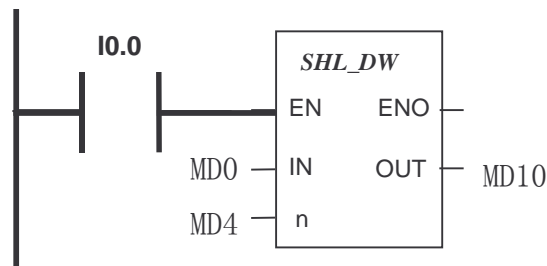
**说明**

双字向右移位和双字向左移位指令将输入双字（IN）数值向右或向左移动N位，并将结果载入输出双字（OUT）。移位指令对每个移出位补0。如果移位数目（N）大于或等于32，则数值最多被移位32次。

双字向右和向左移位操作带符号。

表 7-41 双字左右移位方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	IN	DWORD	VD, ID, QD, MD, AC, K	移位输入数据
	N	DWORD	VD, ID, QD, MD, AC, K	移位输入位数
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	OUT	DWORD	VD, QD, MD, AC	数据输出



输入I0.0的1信号状态激活SHL\_DW方块，将存储器MD0中的值左移，左移位数由存储器MD4中给出，结果存入MD10



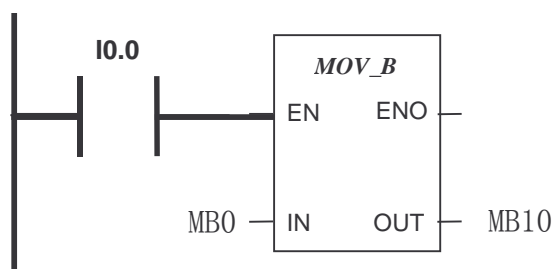
## 7.10. 赋值类指令

### 7.10.1. 字节赋值

**说明** 移动字节指令将输入字节（IN）移至输出字节（OUT）。移动不改变输入字节。

表 7-42 字节赋值方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	BYTE	VB, IB, QB, MB, AC, K	数据输入
	OUT	BYTE	VB, QB, MB, AC	数据输出



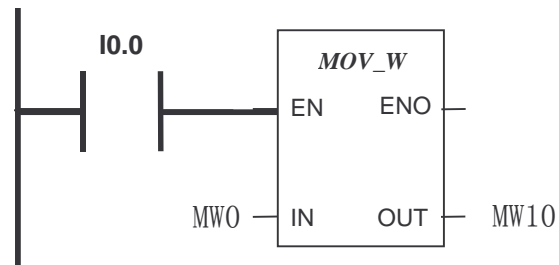
输入I0.0的1信号状态激活MOV\_B方块，将存储器MB0中的值复制到存储器MB10中。

### 7.10.2. 字赋值

**说明** 移动字节指令将输入字（IN）移至输出字（OUT）。移动不改变输入字。

表 7-43 字赋值方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	WORD	VW, IW, QW, MW, AC, K	数据输入
	OUT	WORD	VW, QW, MW, AC	数据输出



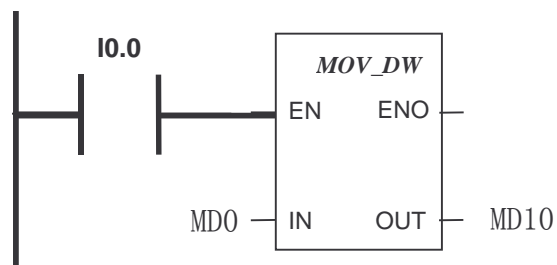
输入I0.0的1信号状态激活MOV\_W方块，将存储器MWO中的值复制到存储器MW10中。

### 7.10.3. 双字赋值

**说明** 移动双字指令将输入双字（IN）移至输出双字（OUT）。移动不改变输入双字。

表 7-44 双字赋值方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	DWORD	VD, ID, QD, MD, AC, K	数据输入
	OUT	DWORD	VD, QD, MD, AC	数据输出



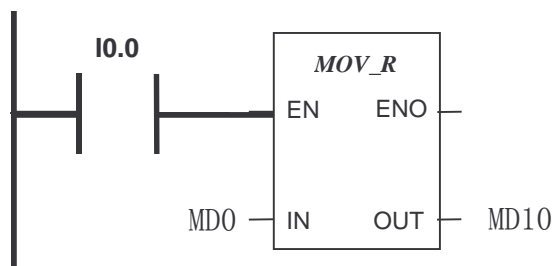
输入I0.0的1信号状态激活MOV\_DW方块，将存储器MD0中的值复制到存储器MD10中。

### 7.10.4. 浮点赋值

**说明** 移动浮点指令将32位、浮点输入双字（IN）移至输出双字（OUT）。移动不改变输入双字。

表 7-45 浮点赋值方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	DWORD	VD, ID, QD, MD, AC, K	数据输入
	OUT	DWORD	VD, QD, MD, AC	数据输出

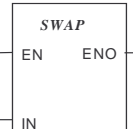


输入I0.0的1信号状态激活MOV\_D方块，将存储器MD0中的值复制到存储器MD10中。

### 7.10.5. 字节转换

**说明** 交换字节指令交换字（IN）的最高位字节和最低位字节。

表 7-46 字节转换方块图和参数

LAD 方块	参数	数据类型	存储区	说明
	EN	BOOL	V, I, Q, M, SM, C, T	使能输入
	ENO	BOOL	V, I, Q, M, SM, C, T	使能输出
	IN	WORD	VW, IW, QW, MW, AC, K	数据输入

## 8. 编程举例

### 8.1. 电机Y-Δ 起动

#### 说明

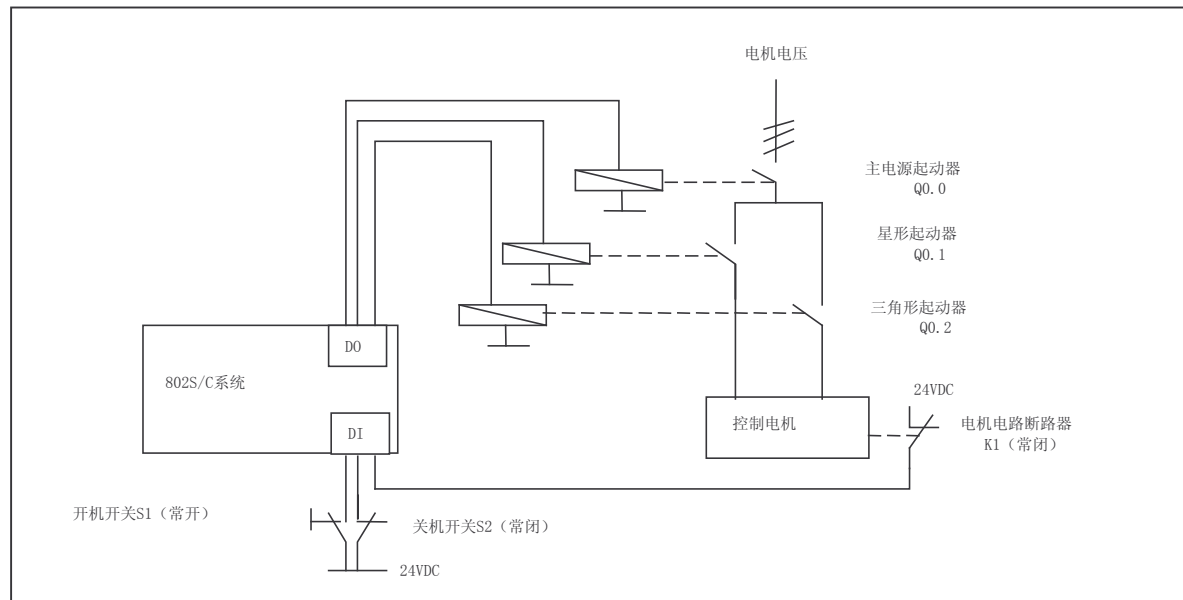
本例程序的目的是通过程序的编程来实现电机由Y方式到Δ方式的切换过程，当起动开关接通时，电机绕组星形连接运转，经过预置5秒后，电机绕组切换为三角形连接运转，并保持三角形连接连续运转工作。

当关机开关接通或电机电路断路器动作时，电机关机。当起动开关和关机开关同时被按下时，电机仍然处于停机状态。

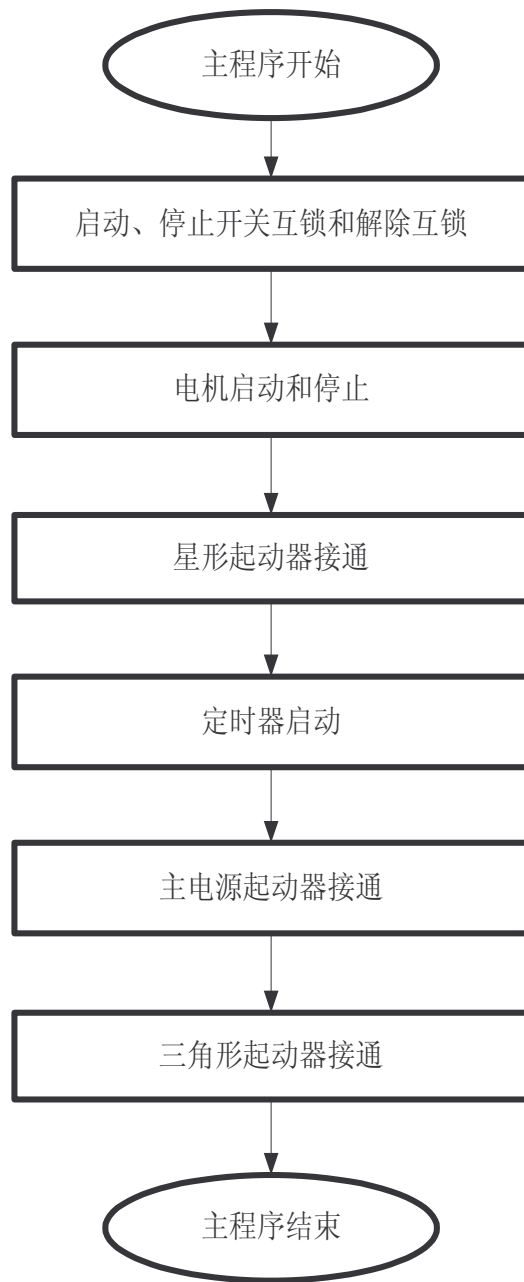
表 8-1 电机Y-Δ 起动元素

系统部件	地址	符号
电机起动按钮	I0.0	S1
电机停止按钮	I0.1	S2
电机电路断路器	I0.2	K1
电机运行主接触器	Q0.0	J1
电机运行Y接触器	Q0.1	J2
电机运行Δ接触器	Q0.2	J3

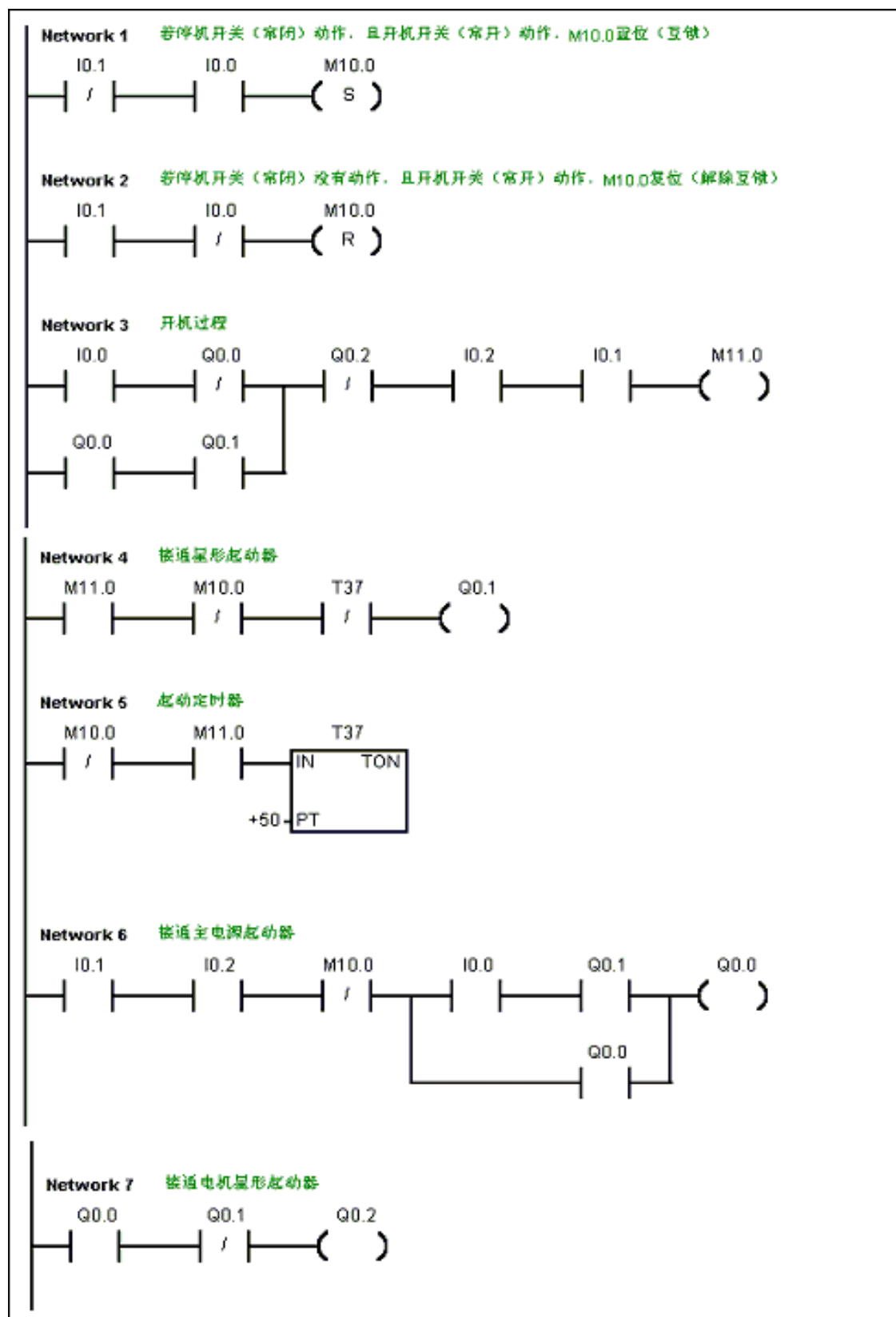
#### 例图



程序流程图



程序和注释



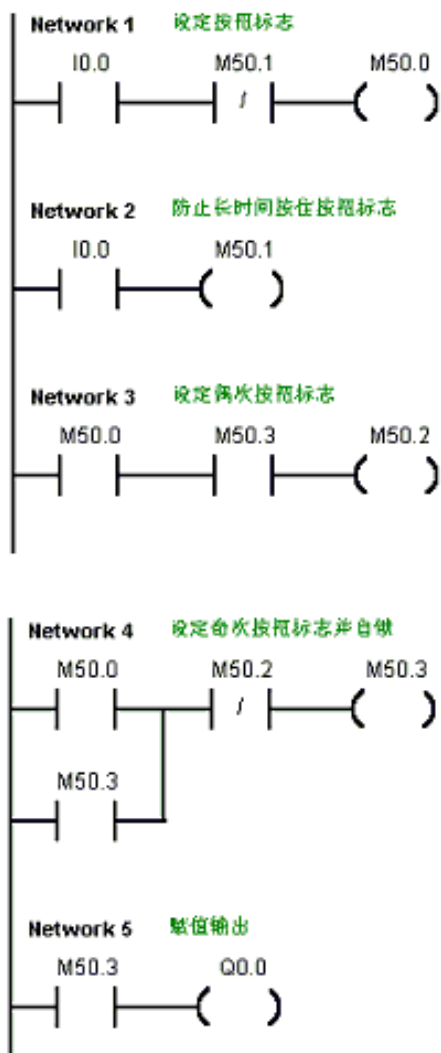
## 8.2. 二进制分频器（触发电路）

**说明** 二进制分频器（触发电路）是指通过一个输入点来控制一个输出的两种相反的状态，如通过一按键来控制一指示灯，按一下，灯被点亮，再按一下，则灯灭。这种控制方式也可被用于对润滑，冷却，卡盘等控制。下面用冷却来说明程序的编制。

表 8-2 二进制分频器（触发电路）元素

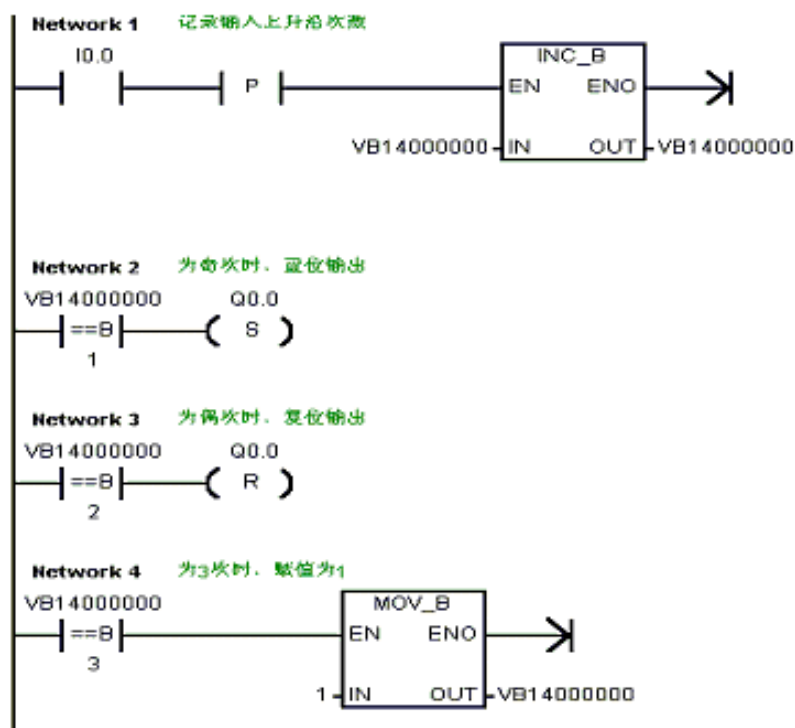
系统部件	地址	符号
冷却启动按钮	I0.0	S1
冷却启动指示灯	Q0.0	LED1

### 程序方法一

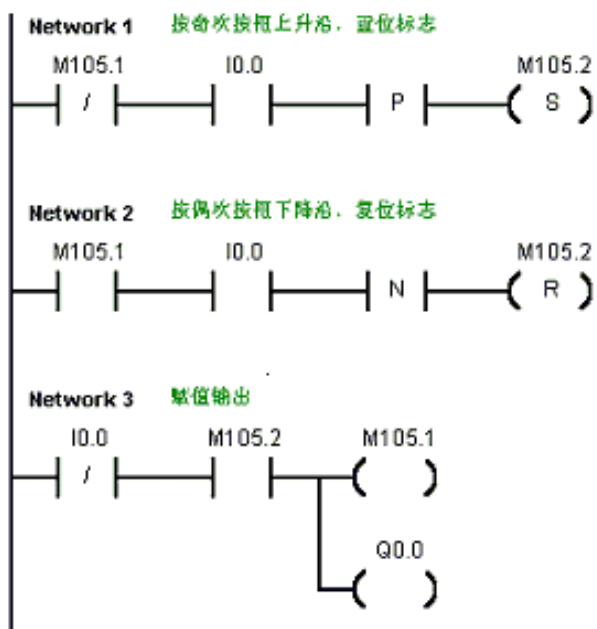




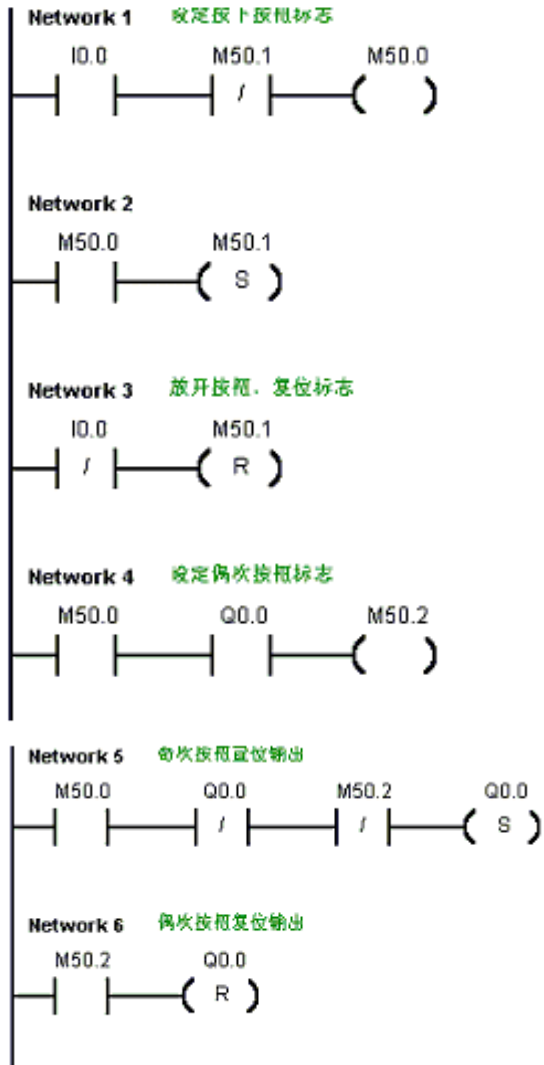
## 程序方法二



## 程序方法三



程序方法四



### 8.3. 如何追踪一台设备运行了多长时间

#### 说明

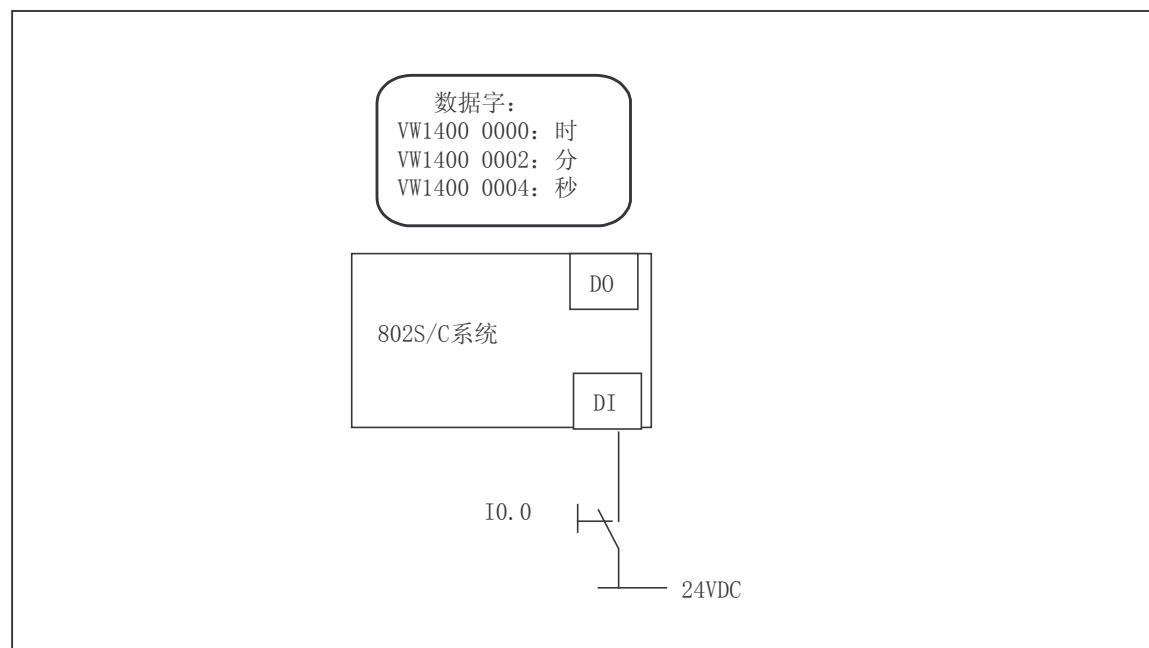
本例程序的目的是记录一台设备（制动器、开关等）运行的时间，以下前提必须满足：当设备运行时，必须给输入I0.0提供24V信号，当设备不工作时不提供电压。

当提供信号时，开始测量时间。如果没有输入信号，那么就中断时间的测量，直到重新提供输入信号为止。测量到的小时数存在VW14000000中，分钟数存在VW14000002中，秒钟数存在VW14000004中。

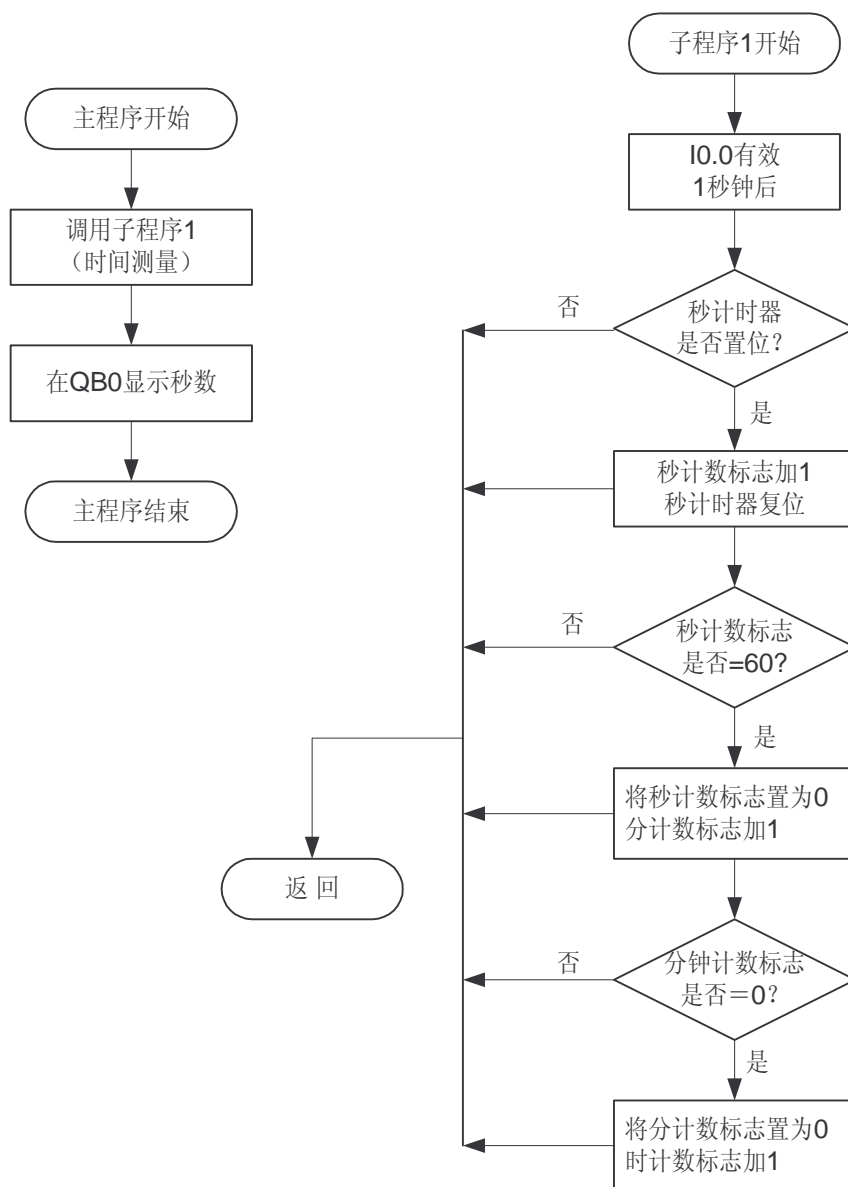
表 8-3 如何追踪一台设备运行了多长时间元素

系统部件	地址
设备运行输入信号	I0.0
设备运行小时数	VW1400 0000
设备运行分钟数	VW1400 0002
设备运行秒钟数	VW1400 0004

#### 例图

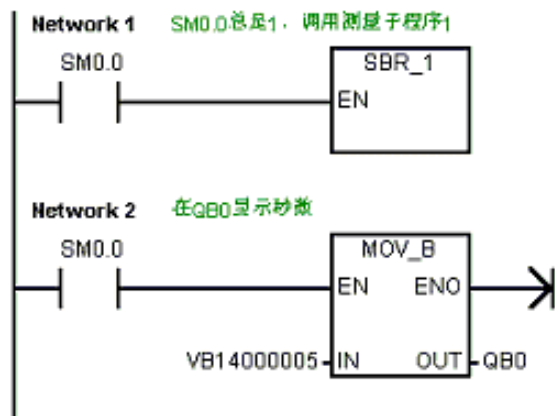


程序流程图

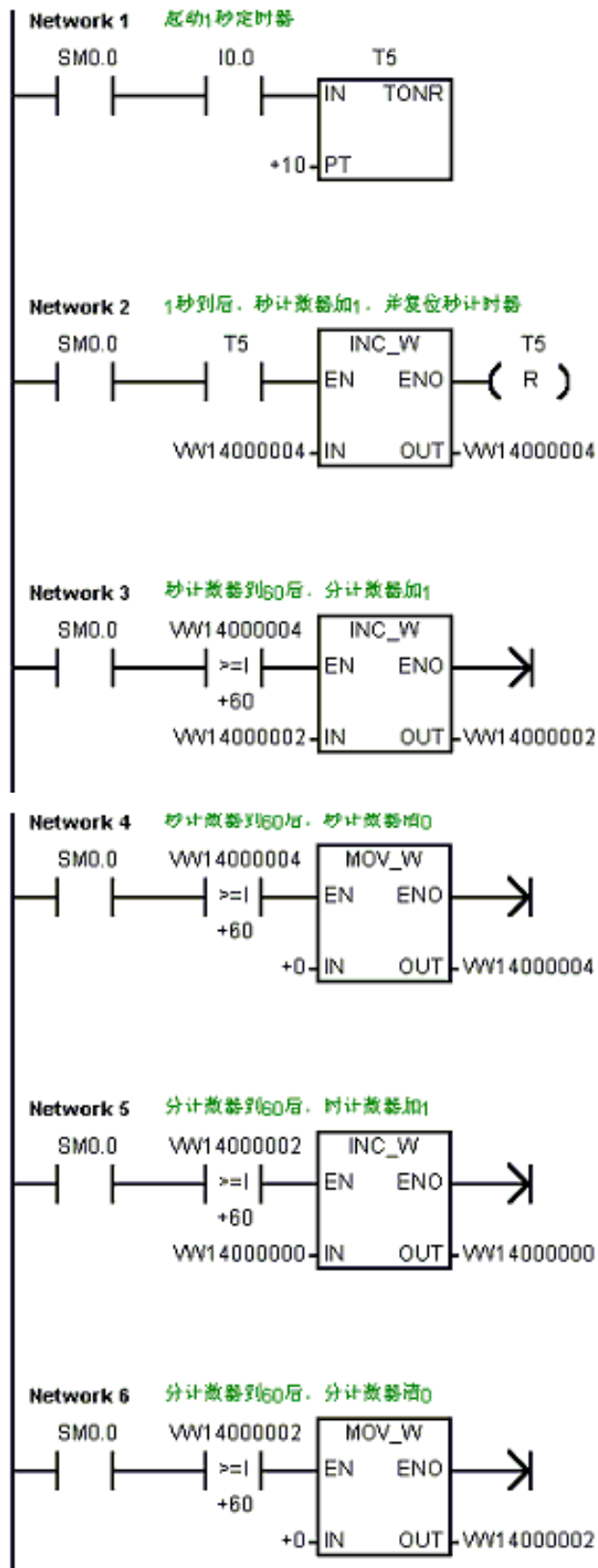


## 程序和注释

## 主程序 OB1



子程序 SBR1



## 9. SINUMERIK 802S/C base line PLC 子程序库说明

### 9.1. 概述

说明	<p>SINUMERIK 802S baseline和 SINUMERIK 802C baseline 是用于车床和铣床的紧凑型数控系统。</p> <p>SINUMERIK 802S base line 装备了步进驱动系统，而SINUMERIK 802C base line 则装备模拟伺服驱动系统。</p> <p>为使数控系统与机床的调试快捷方便，我们提供了基于 <i>Programming Tool PLC 802 V3.1</i> 以上版本的PLC子程序库。</p>
读者	本章为具有CNC和PLC基础知识的机床制造商的工程师设计。
相关资料	<p>详细信息请参阅系列资料：</p> <ul style="list-style-type: none"><li>• <i>SINUMERIK 802S/C baseline Start-up and installation Guide</i></li><li>• <i>SINUMERIK 802S/C baseline Function description</i></li><li>• <i>SimoDrive 611 planning guide</i></li></ul>

#### 9.1.1. 子程序库的构成

子程序库包括在802S/C baseline的工具盒中。

子程序库由其说明文件，和两个PLC项目文件构成：

<b>UBR_LIBRARY.PTP</b>	PLC项目文件。包括了子程序库所提供的全部子程序和空的主程序（OB1）。
<b>SAMPLE.PTP</b>	PLC项目文件。包括了全部子程序和适用于车床或铣床的主程序。

在项目文件SUBR\_LIBRARY.PTP中提供了各种基本功能的子程序，例如急停处理、坐标控制等，以及诸如冷却控制、润滑控制、简易刀架控制子程序。利用这些子程序，可以象搭积木一样，将各个所需的子程序放在主程序中，然后装入802S/C base line中。

项目文件SAMPLE.PTP是利用子程序库SUBR\_LIBRARY.PTP的子程序搭建的一个完整的应用程序。SAMPLE.PTP在供货前已经预装入系统中。对于SAMPLE.PTP的控制功能可以覆盖的那些机床，没有必须编写任何PLC应用程序，只需通过设定相关的PLC参数，即可实现所需的控制功能。当然您也可以修改SAMPLE.PTP，在其基础上增加您所需要的控制功能。

在您利用SUBR\_LIBRARY.PTP搭建自己的应用程序或修改SAMPLE.PTP之前，详细阅读本说明。

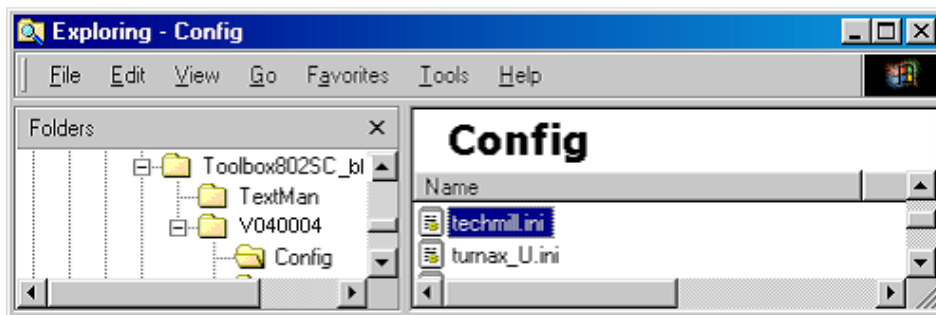
### 9.1.2. 重要事项

下面的各项内容，对您了解子程序库已经各个子程序非常重要。

#### 9.1.2.1. NC初始化

802S/C base line的缺省配置均为车床、无固定循环。

如果您要控制铣床，首先应将铣床配置文件TECHMILL.TXT由工具盘通过RS232接口利用WINPCIN工具传入系统中。



#### 9.1.2.2. 系统资源的划分

资源（数量）	为用户预留	为子程序库预留
I/O (48input/16output)	所有输入输出（单及性主轴时Q0.0/ Q0.1作为主轴方向和使能信号被占 用）	无
定时器（16）	8: T0~T7	8: T8~T15
计数器（32）	24: C0~C23	8: C24~C31
存储器（128 Bytes）	64: M0.0~M63.7	64: M64.0~M127.7
记忆存储器 (64 Bytes)	32: V14000000.0~V14000031.7	32: V14000032.0~V14000063.7
报警（32）	16: V16000000.0~V16000001.7	16: V16000002.0~V16000003.7
参数MD14510（32）	16: MD14510[0]~ MD14510[15]	16: MD14510[16]~MD14510[32]
参数MD14512（32）	16: MD14512[0]~ MD14512[15]	16: MD14512[16]~MD14512[32]
子程序（64）	32: SBR0 ~ SBR31	32: SBR32 ~ SBR63
符号表（32）	15: USR1 ~ USR15	17: USR16 ~ USR31



### 9.1.2.3. 符号命名约定

子程序库所使用符号的定义符合下列约定：

1. 前两个引导字母表示目标方向：
  - P\_ - 送至 PLC 接口
  - H\_ - 送至 HMI 接口
  - N\_ - 送至 NCK 接口
  - M\_ - 送至 MCP 接口
2. 随后的字母表示区域：
  - C\_ - 通道数据区
  - 1\_ - 轴数据区
  - M\_ - MCP数据区
3. 符号中的其他缩写：
  - HWL - 硬限位
  - HW - 手轮
  - RT - 快速移动
  - TK - 点动键
  - ACT - 有效
  - SEL - 选择

一个符合由最多11个字母构成。除下划线“\_”外，不允许特殊字符，=, +, -, , [, ]等。

## 9.1.2.4. 符号表的安排

USR	符号表名	符号表内容
1	DI_1	DI 接口 100: I0.0~I0.7
2	DI_2	DI 接口 101: I1.0~I1.7
3	DI_3	DI 接口 102: I2.0~I2.7
4	DI_4	DI 接口 103: I3.0~I3.7
5	DI_5	DI 接口 104: I4.0~I4.7
6	DI_6	DI 接口 105: I5.0~I5.7
7	DO_0	DO接口200: Q0.0~Q0.7
8	DO_1	DO接口201: Q1.0~Q1.7
16	IS_MCP	接口信号 MCP
17	IS_HMI	接口信号 HMI
18	IS_AUX	接口信号 辅助功能
19	IS_NCK	接口信号 NCK
20	IS_CHA	接口信号 通道
21	IS_X	接口信号 X 轴
22	IS_Y	接口信号 Y 轴
23	IS_Z	接口信号 Z 轴
24	IS_SP	接口信号 主轴
25		
26	MD_PLC	PLC 参数
27	ALARM	接口信号 用户报警
28	NV_MEM	保持数据
29	SPC_MEM	特殊标志位
30	SBR_MEM	子程序库使用存储器定义
31 ~ 32		子程序库备用

## 9.1.2.5. 子程序库的结构

SBR	子程序名	子程序功能描述
0 ~ 30	-	为用户预留
31	USR_INI	为用户初始化预留（由子程序32 USR_INI调用）
32	PLC_INI	PLC初始化
33	EMG_STOP	急停处理（驱动器上电下电时序控制）
34	X_CROSS	点动键布局控制（通过PLC机床参数）
35	SPINDLE	主轴控制
36	MINI_HHU	西门子手持单元（通过PLC输入）
38	MCP_NCK	机床面板和操作面板信号激活NCK功能
39	HMI_HW	操作面板手轮选择
40	AXES_CTL	坐标使能、硬限位控制、抱闸释放等
41	GEAR_CHG	铣床模拟主轴两挡变速
44	COOLING	冷却控制
45	LUBRICAT	定时定量润滑控制
46	TURRET1	简易刀架控制（4或6或8工位刀架）
48	TOOL_DIR	计算就近找刀的方向
49	LOCK_UNL	夹紧放松控制—用于铣床的刀具卡紧放松或车床卡盘控制
51	Trg_key_OR	进给/主轴由按键控制
62	FILTER	输入输出信号滤波器。仅用于SAMPLE

## 9.1.2.6. PLC 机床数据

MD14512 机床参数	USER_DATA_HEX							
	PLC 机床参数 - 十六进制							
数据号	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
14512[0]	定义有效输入位 (接口 X100, 端子号: 0~7)							
	I0.7	I0.6	I0.5	I0.4	I0.3	I0.2	I0.1	I0.0
14512[1]	定义有效输入位 (接口 X101, 端子号: 8~15)							
	I1.7	I1.6	I1.5	I1.4	I1.3	I1.2	I1.1	I1.0
14512[2]	定义输入位为常闭连接 (接口 X100, 端子号: 0~7)							
	I0.7	I0.6	I0.5	I0.4	I0.3	I0.2	I0.1	I0.0
14512[3]	定义输入位为常闭连接 (接口 X101, 端子号: 8~15)							
	I1.7	I1.6	I1.5	I1.4	I1.3	I1.2	I1.1	I1.0
14512[4]	定义有效输出位 (接口 X200, 端子号: 0~7)							
	Q0.7	Q0.6	Q0.5	Q0.4	Q0.3	Q0.2	Q0.1	Q0.0
14512[5]	定义有效输出位 (接口 X201, 端子号: 8~15)							
	Q1.7	Q1.6	Q1.5	Q1.4	Q1.3	Q1.2	Q1.1	Q1.0
14512[6]	定义输出位为低电平有效 (接口 X200, 端子号: 0~7)							
	Q0.7	Q0.6	Q0.5	Q0.4	Q0.3	Q0.2	Q0.1	Q0.0
14512[7]	定义输出位为低电平有效 (接口 X201, 端子号: 8~15)							
	Q1.7	Q1.6	Q1.5	Q1.4	Q1.3	Q1.2	Q1.1	Q1.0
14512[11]	PLC实例程序配置							
	刀架 控制 有效	模拟 主轴 换挡 控制			主轴  有效	卡紧 放松  有效	润滑  有效	冷却  有效
14512[12]	进给/主轴倍率控制方式配置							
	定义主轴倍率 转换速度	定义进给倍率 转换速度			开机 主轴 倍率 设置	开机 进给 倍率 设置		倍率控 制方式

MD14512 机床参数	USER_DATA_HEX							
	PLC 机床参数 – 十六进制							
数据号	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
14512[16]		Z 轴 旋转 监控	Y 轴 旋转 监控	X 轴 旋转 监控	主轴配置 配备 倍率 开关	单极性 模拟 主轴	主轴 使能 自动 取消	调试 过程 中
14512[17]	定义带制动装置的进给电机				定义回参考点倍率无效的轴			
		Z 轴 抱闸	Y 轴 抱闸	X 轴 抱闸		Z 轴 REF	Y 轴 REF	X 轴 REF
14512[18]	急停链 生效	定义硬限位螺距			技术设定			
		Z 单开 关硬 限位	Y 单开 关硬 限位	X 单开 关硬 限位		开机自 动润滑 一次	驱动优 化生效	

MD14510 机床参数	USER_DATA_INT
	PLC 机床参数 – 整数
数据号	字 (16位整型数)
14510[12]	定义：有关进给/主轴倍率控制的时间量设置。按住进给/主轴倍率减速键大于此设定时间值，进给/主轴倍率将直接降至0%和50%。 单位：100ms 范围：5 ~ 30 (0.5 ~ 3 秒)，若超出此范围，将默认为1.5秒。
14510[13]	定义：有关进给/主轴倍率控制的时间量设置。按住进给/主轴倍率100%键大于此设定时间值，进给/主轴倍率将直接变为100%。 单位：100ms 范围：5 ~ 30 (0.5 ~ 3 秒)，若超出此范围，将默认为1.5秒。
14510[16]	定义：机床类型 单位：- 范围：0 – 车床；1 – 铣床；> 2 无定义
14510[17]	定义：驱动器类型 单位：- 范围：0 – 步进驱动器；1 – 伺服驱动器（如SIMODRIVE 611U）；> 2 – 无定义。
14510[20]	定义：刀架刀位数 单位：- 范围：4, 6, 8 (注意：实例SAMPLE 只允许 4 / 6 工位刀架)
14510[21]	定义：换刀监控时间（换刀必须在该时间内完成） 单位：0.1 秒 范围：30 ~ 200 (3 ~ 20 秒)

MD14510 机床参数	USER_DATA_INT PLC 机床参数 – 整数
数据号	字（16位整型数）
14510[22]	定义：刀架卡紧时间 单位：0.1 秒 范围：5 ~ 30（0.5 ~ 3 秒）
14510[23]	定义：外部主轴制动时间（适于开关量控制的主轴） 单位：0.1 秒 范围：5 ~ 200（0.5 ~ 20 秒）
14510[24]	定义：导轨润滑间隔 单位：1 分钟 范围：5 ~ 300 分钟
14510[25]	定义：导轨润滑时间 单位：0.1 秒 范围：10 ~ 200（1 ~ 20 秒）
14510[26]	定义：X 轴 + 点动键的键号 单位：- 范围：22 ~ 30之间，除26以外
14510[27]	定义：X 轴 - 点动键的键号 单位：- 范围：22 ~ 30之间，除26以外
14510[28]	定义：Y 轴 + 点动键的键号 单位：- 范围：22 ~ 30之间，除26以外
14510[29]	定义：Y 轴 - 点动键的键号 单位：- 范围：22 ~ 30之间，除26以外
14510[30]	定义：Z 轴 + 点动键的键号 单位：- 范围：22 ~ 30之间，除26以外
14510[31]	定义：Z 轴 - 点动键的键号 单位：- 范围：22 ~ 30之间，除26以外

## 十六进制PLC机床参数

- (1) MD14512[16]
- Bit 0=0 – PLC 正常运行（缺省设定）
  - Bit 0=1 – 调试方式。PLC不检测馈入模块的就绪信号
  - Bit 1=0 – 无主轴命令且主轴已停止停止后按主轴停止键取消主轴使能（缺省设定）
  - Bit 1=1 – 无主轴命令且主轴停止后主轴使能自动取消
  - Bit 2=0 – 带有+/-10V给定的模拟主轴（缺省设定）
  - Bit 2=1 – 带有0~10V给定的模拟主轴
  - Bit 3=0 – MCP 上无主轴倍率开关（缺省设定）
  - Bit 3=1 – MCP 有主轴倍率开关
  - Bit 6/5/4=0 – 802S旋转监控无效（缺省设定）
  - Bit 6/5/4=1 – 802S旋转监控生效
- (2) MD14512[17]
- Bit 2/1/0=0 – 返回参考点时进给倍率有效（缺省设定）
  - Bit 2/1/0=1 – 返回参考点时进给倍率无效
  - Bit 6/5/4=0 – Z/Y/X 轴电机无抱闸（缺省设定）
  - Bit 6/5/4=1 – Z/Y/X 轴电机有抱闸（只允许一个电机带抱闸）
- (3) MD14512[18]
- Bit 1=0 – 子程序40的输入#OPTM无效（缺省设定）
  - Bit 1=1 – 子程序40的输入#OPTM 有效。#OPTM=1 – 电机抱闸释放
  - Bit 2=0 – 开机无润滑（缺省设定）
  - Bit 2=1 – 上电自动润滑一次
  - Bit 6/5/4=0 – Z/Y/X 每轴具有两个硬限位开关（缺省设定）（当Bit7=0时）
  - Bit 6/5/4=1 – Z/Y/X 每轴具有一个硬限位开关（当Bit7=0时）
  - Bit 7=0 – 硬限位采用PLC方案（缺省设定）（Bit 6/5/4 有效）
  - Bit 7=1 – 硬件方案（超程链）

**重要**

以上的PLC机床参数位必须按十六进制格式输入。请将所需的位设定转换为十六进制。

#### 9.1.2.7. 子程序库项目文件中用到的特殊标志存储器

##### 用于子程序的输入:

SM0.0 - 常“1”，符号定义ONE

M122.0 - 常“0”，符号定义ZERO（SRB33每次扫描自动清零）

##### 用于无定义的子程序输出:

M127.7 - 用于无效的输出位，符号定义NULL\_b

MB127 - 用于无效的输出字节，符号定义NULL\_B

MW126 - 用于无效的输出字，符号定义NULL\_W

MD124 - 用于无效的输出长字，符号定义NULL\_D



## 9.2. 子程序库说明

### 9.2.1. 子程序 32 – PLC\_INI (PLC 初始化)

#### 目的

该子程序在第一个PLC周期运行，设定系统接口的基本信号。

以下是在该子程序中设定的接口信号：

V32000006.7 - 进给倍率对通道生效

V380x0001.5 - 各轴的激活测量系统1

V380x0001.7 - 各轴激活进给倍率

(主轴倍率是否生效由MD14512[18] Bit0 定义)

在该子程序结束之前自动调用子程序31—用户初始化。因此用户所有有关的初始化内容都应该编在子程序31中。请参见9.3节实例程序说明。

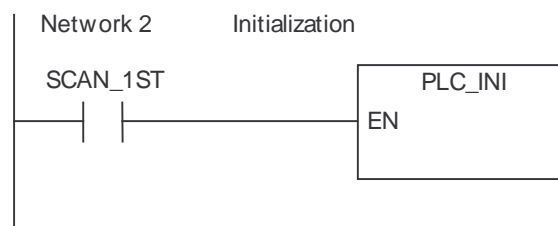
所使用的局部变量 无

占用的标志存储器 无

#### 相关的PLC机床参数

MD14512[18]Bit0=1 - 主轴倍率开关生效

#### 子程序调试实例



## 9.2.2. 子程序 33 – EMG\_STOP（急停处理）



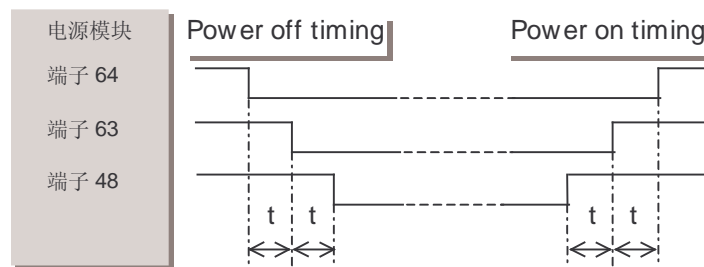
## 注意

请检查该子程序是否满足所有相关的安全要求。

## 目的

该子程序根据802S/802C baseline和802D功能说明中急停处理的时序要求进行急停处理。

对于802C baseline 该子程序处理电源馈入模块的上电和下电时序（详见611规划手册）。



急停不仅可以由急停按钮触发，而且在上电之后（端子64和9接通），一个无效的就绪信号或有效的I<sup>2</sup>t信号都可触发接通。

两个报警可以由该子程序激活：

Alarm 700016 – DRIVE NOT READY

Alarm 700017 – I<sup>2</sup>T ALARM FOR INFEEED MODULE

对于802S/802C baseline均可使用该子程序，没有任何区别。

如果控制802S base line，可将无效的输入信号接入子程序输入T\_72（常“1”无效—即SM0.0）及T\_52（常“0”无效—即），无效的输出信号连接到无效输出位即可—NULL\_b（M127.7）。

## 所使用的局部变量

## 输入：

DELAY	WORD	上电下电时序延时（单位：2*PLC 扫描周期）
EMG_K	BOOL	急停按钮（NC）
EN_K	BOOL	伺服使能键（NC）
T_72	BOOL	馈入模块状态反馈：驱动器就绪（NO）
T_52	BOOL	馈入模块状态反馈：I <sup>2</sup> t 监控报警（NO）

注：NO – 常开接法 ； NC – 常闭接法 ；

注：PLC 循环时间由机床参数MD100074（1=12ms；2=24ms）。

## 输出：

T_48	BOOL	连接611馈入模块的端子48和端子9。
T_63	BOOL	连接611馈入模块的端子63和端子9。
T_64	BOOL	连接611馈入模块的端子64和端子9。

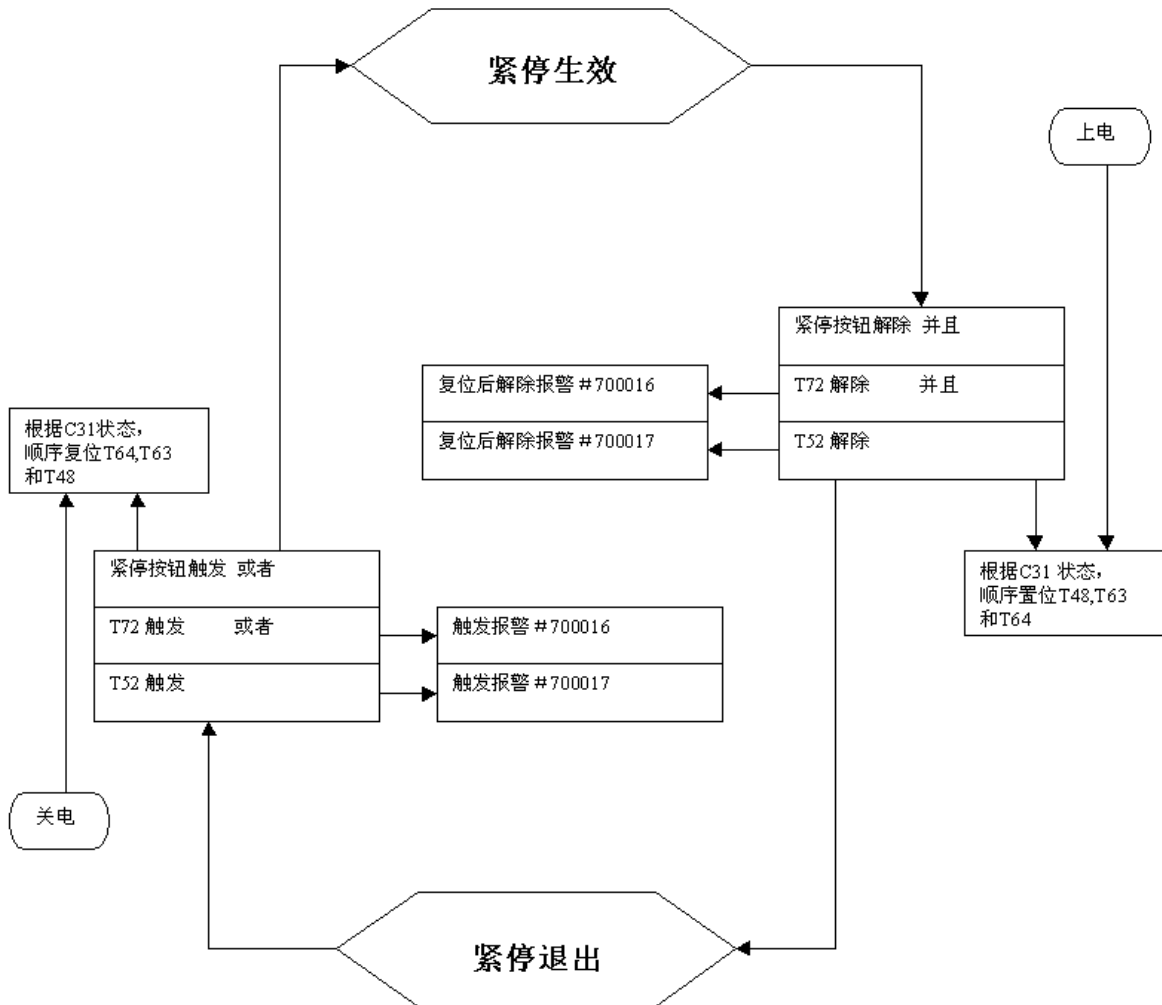
EN_L	BOOL	驱动器使能状态指示
ERR1	BOOL	错误信息：驱动器未就绪
ERR2	BOOL	错误信息：I <sup>2</sup> t监控报警

占用的标志存储器

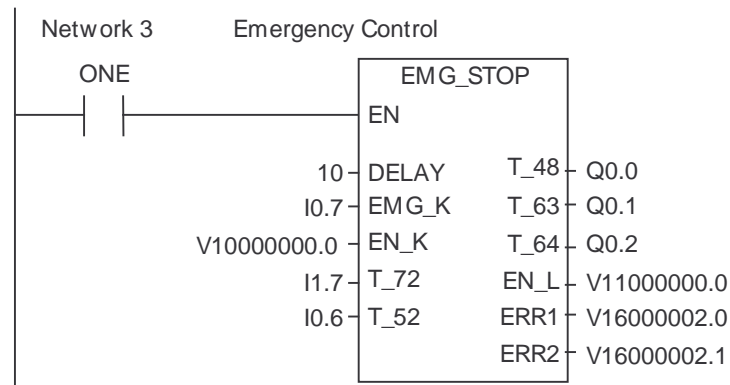
MB118		为该子程序预留
P_TON	C31	CTU 用于定时控制

相关的PLC机床参数 无

子程序流程示意图



## 子程序调试实例



## 9.2.3. 子程序 34-X\_CROSS (点动键布局控制)

**目的** 该子程序根据PLC机床参数定义的点动键号，将点动按键的状态送到NCK相应的接口。另外该子程序通过标志存储器建立一个局部接口，用于带有点动键的手持单元。  
该子程序被子程序38调用。

**所使用的局部变量** 无

**占用的标志存储器** MB107 – 用于手持单元的内部接口

**相关的PLC机床参数**

MD14510[26] – 点动键+X的键号（取值范围：16~24，不等于20）；  
MD14510[27] – 点动键-X的键号（取值范围：16~24，不等于20）；  
MD14510[28] – 点动键+Y的键号（取值范围：16~24，不等于20）；  
MD14510[29] – 点动键-X的键号（取值范围：16~24，不等于20）；  
MD14510[30] – 点动键+Z的键号（取值范围：16~24，不等于20）；  
MD14510[31] – 点动键-Z的键号（取值范围：16~24，不等于20）；

按键配置参数为按键的键号。由于K20已被定义为快速叠加键，因此该参数的取值范围为：16~24（20除外）。

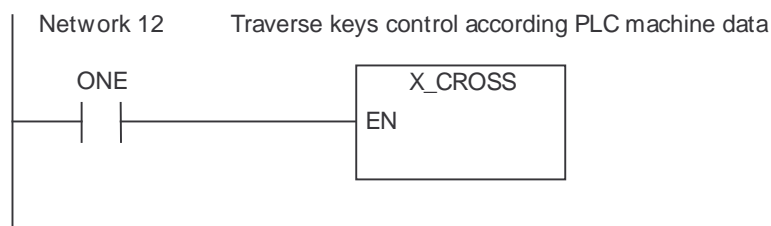


注意：按键K20已定义为快速叠加键。其他点动按键可根据机床的要求自由定义。

注意：如果没有定义点动键的机床参数，手动操作不生效。

注意：机床控制面板的布局请参考3.3节。

## 子程序调试实例



## 9.2.4. 子程序 35 – SPINDLE（用于开关主轴，或单、双极性的模拟主轴）

## 目的

该子程序用于主轴控制。

可控制以下类型的主轴：开关主轴；单极性（0~+10V 给定）模拟主轴；或双极性（±10V 给定）模拟主轴。

## 所使用的局部变量

## 输入：

DELAY	WORD	主轴制动时间（单位：100 ms）
T_64	BOOL	馈入模块端子64状态
SP_EN	BOOL	主轴运行使能（NO）
UNI_PO	BOOL	单极性主轴生效（NO）
KEYcw	BOOL	主轴手动正转键（NO）
KEYccw	BOOL	主轴手动反转键（NO）
KEYstop	BOOL	主轴手动停止键（NO）

## 输出：

SP_CW	BOOL	主轴正转输出
SP_CCW	BOOL	主轴反转输出
SP_BRAKE	BOOL	主轴制动输出
SP_LED	BOOL	主轴工作状态显示（1—主轴运行）
ERROR	BOOL	错误信息：主轴运行条件没有满足

对于不同主轴的应用说明：

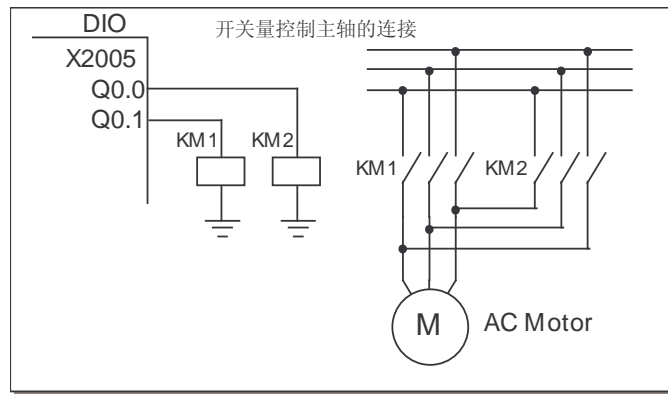
- （1）开关主轴（普通异步电机主轴，由两个接触器控制正转和反转）

## 输入：

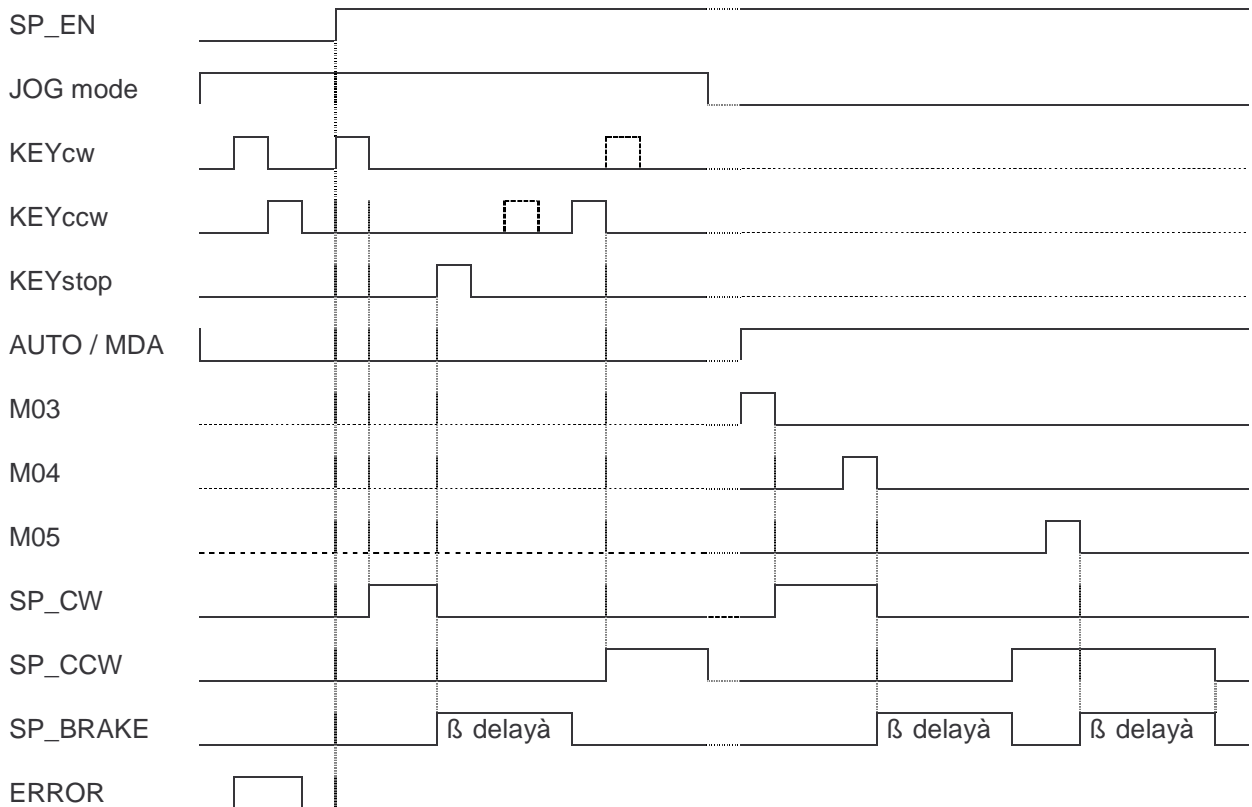
DELAY	- 根据主轴制动的实际时间设定主轴制动延时
T_64	- 通过标志存储位将急停子程序T64的输出连接到该子程序
SP_EN	- 主轴运行条件，如卡盘卡紧状态，可由子程序49引出
UNI_PO	- ZERO
KEYcw	- 来自MCP主轴正转键（V10000001.4）
KEYccw	- 来自MCP主轴反转键（V10000001.6）
KEYstop	- 来自MCP主轴停止键（V10000001.5）

## 输出：

- SP\_CW - 连接主轴正转接触器线圈
- SP\_CCW - 连接主轴反转接触器线圈
- SP\_BRAKE - 连接主轴制动装置（高电平：制动）
- SP\_LED - 主轴运行状态。  
通过存储位将主轴运行状态连接到子程序49作为互锁条件，即在主轴运行构成中卡盘不能放松。
- ERROR - 通过输出位连接指示灯或输出到接口V1600000x.x产生PLC报警



开关量控制主轴的控制时序图：



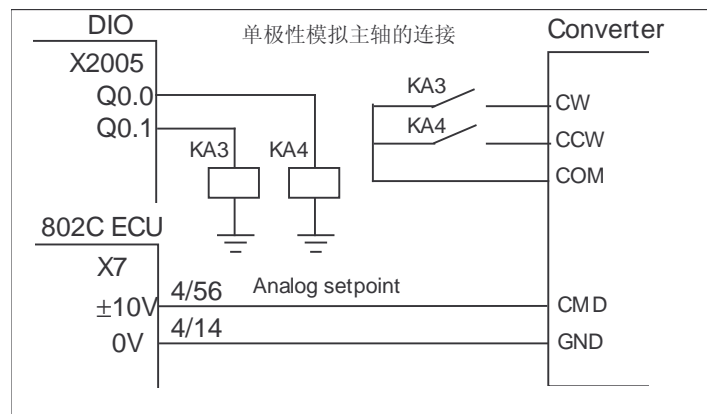
## (2) 0~10V 给定的模拟主轴（如变频器）

**输入：**

- DELAY - 根据变频器的实际制动时间设定主轴制动延时
- T\_64 - 通过标志存储位将急停子程序T64的输出连接到该子程序
- SP\_EN - 主轴运行条件，如卡盘卡紧状态，可由子程序49引出
- UNI\_PO - 常“1”位—SM0.0
- KEYcw - 来自MCP主轴正转键（V10000001.4）
- KEYccw - 来自MCP主轴反转键（V10000001.6）
- KEYstop - 来自MCP主轴停止键（V10000001.5）

**输出：**

- SP\_CW - 通过继电器将变频器的正转使能和其公共端短接
- SP\_CCW - 通过继电器将变频器的反转使能和其公共端短接
- SP\_BRAKE - 输出到空位—M127.7
- SP\_LED - 主轴运行状态。  
通过存储位将主轴运行状态连接到子程序49作为互锁条件，即在主轴运行构成中卡盘不能放松。
- ERROR - 通过输出位连接指示灯或输出到接口V1600000x.x产生PLC报警



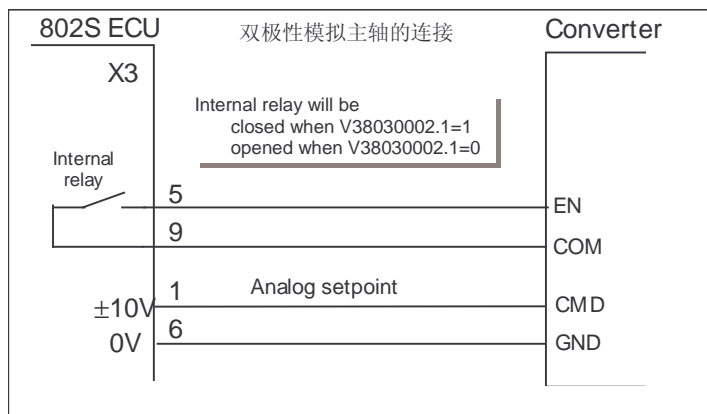
(3) ±10V给定模拟主轴（伺服主轴或双极性给定变频器）

**输入：**

- DELAY - 根据变频器的实际制动时间设定主轴制动延时
- T\_64 - 通过标志存储位将急停子程序T64的输出连接到该子程序
- SP\_EN - 主轴运行条件，如卡盘卡紧状态，可由子程序49引出
- UNI\_PO - ZERO
- KEYcw - 来自MCP主轴正转键（V10000001.4）
- KEYccw - 来自MCP主轴反转键（V10000001.6）
- KEYstop - 来自MCP主轴停止键（V10000001.5）

**输出：**

- SP\_CW - 连接到空位—M127.7
- SP\_CCW - 连接到空位—M127.7
- SP\_BRAKE - 连接到空位—M127.7
- SP\_LED - 主轴运行状态。  
通过存储位将主轴运行状态连接到子程序49作为互锁条件，即在主轴运行构成中卡盘不能放松。
- ERROR - 通过输出位连接指示灯或输出到接口V1600000x.x产生PLC报警



**注意：**当802C baseline连接双极性模拟主轴时，主轴控制单元的模拟给定信号应与由X7端口引出的电缆上标有4/56和4/14的信号线连接，其使能与标有4/65和4/9的信号线连接。

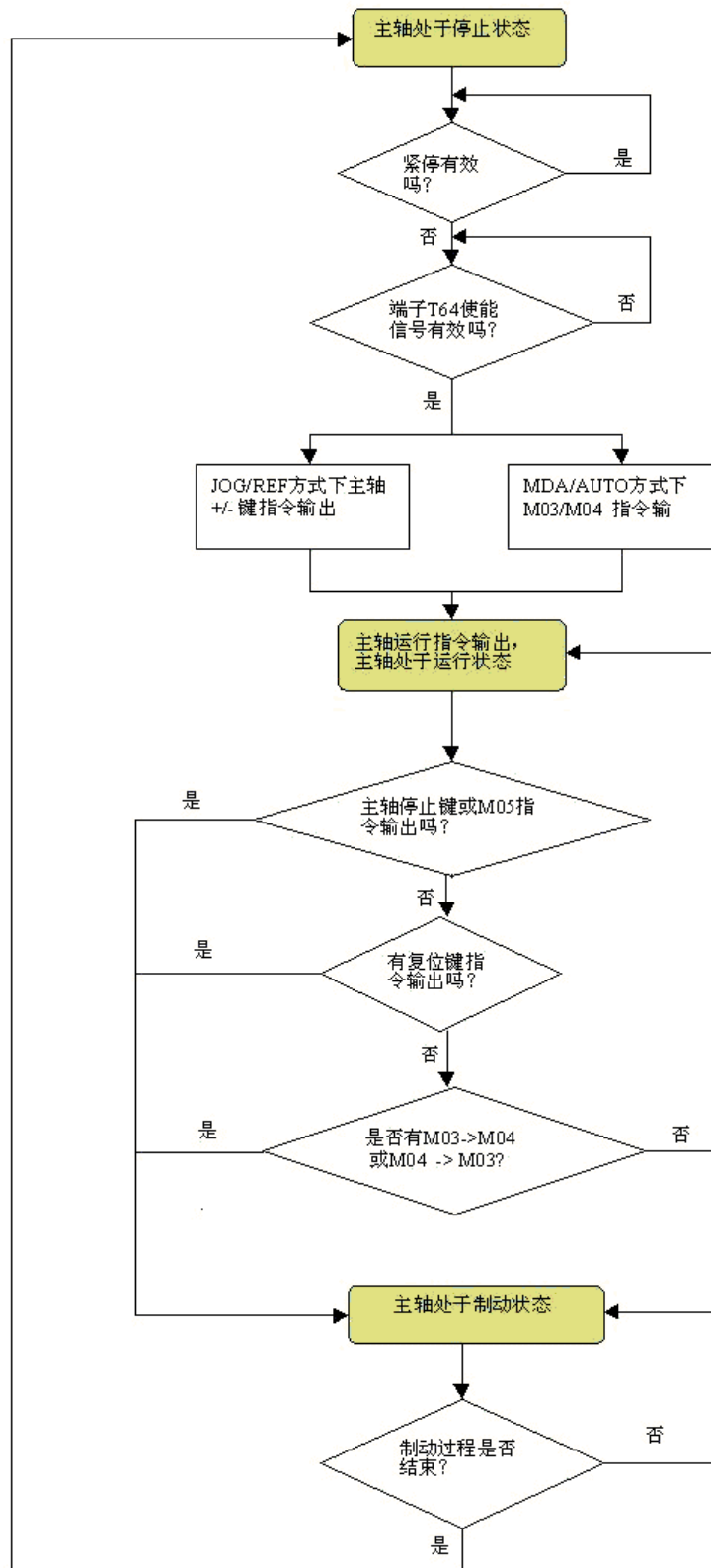
**占用的标志存储器**

- MB115 状态存储器（详见符号表USR30）
- MB116 状态存储器
- T12 主轴制动定时器

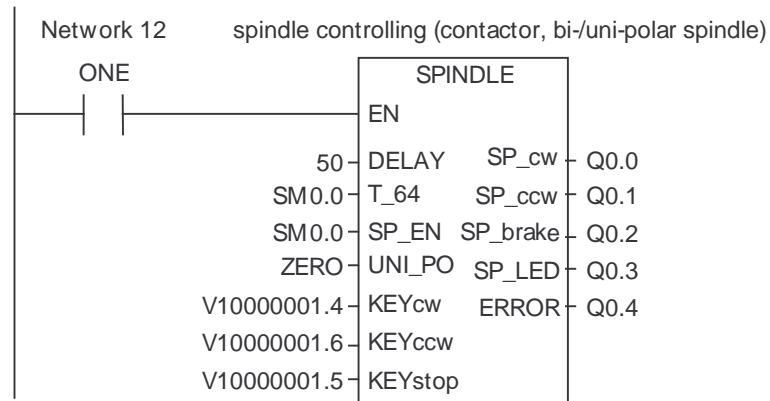
**相关的PLC机床参数** 无



子程序流程示意图



子程序调试实例



9.2.5. 子程序36 – MINI\_HHU （为西门子小型手持单元设计6FX2006-1BG00）

**目的** 该子程序的目的是将手持单元上选择的增量点动键通过内部接口送入NCK。  
 另外如果您使用了其它类型的手持单元，可在该子程序的基础上修改以满足手持单元的要求。

所使用的局部变量

输入：

NODEF	WORD	保留字	
ENABLE	BOOL	手持单元使能 (NO)	轴码: Pin8 Pin9 Pin10
AX_A	BOOL	轴选A Pin8 (NO)	X 0 1 0
AX_B	BOOL	轴选B Pin9 (NO)	Y 1 1 1
AX_C	BOOL	轴选C Pin10 (NO)	Z 0 1 1
KEY_Tp	BOOL	点动键+ (NO)	
KEY_Tn	BOOL	点动键- (NO)	
KEY_R	BOOL	快速叠加键 (NO)	
INC1	BOOL	增量 1 (NO)	
INC10	BOOL	增量 10 (NO)	
INC100	BOOL	增量 100 (NO)	

占用的标志存储器

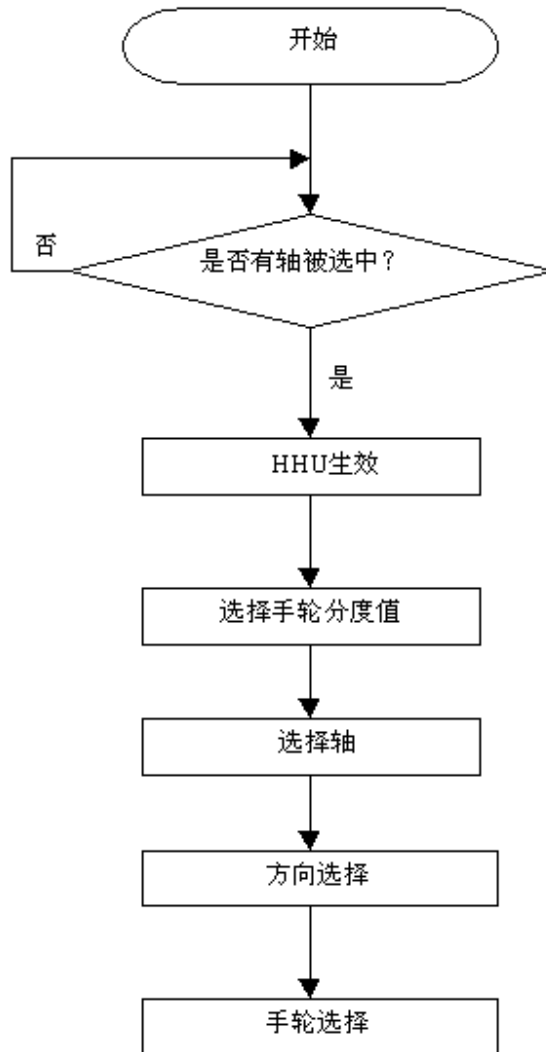
HHU_AX1P	M107.0	HHU 点动键: +X
HHU_AX1N	M107.1	HHU 点动键: -X
HHU_AX2P	M107.2	HHU 点动键: +Y
HHU_AX2N	M107.3	HHU 点动键: -Y
HHU_AX3P	M107.4	HHU 点动键: +Z
HHU_AX3N	M107.5	HHU 点动键: -Z
HHU_RAPID	M107.6	HHU 快速叠加键
HHU_ACT	M107.7	HHU 生效

**注意：** 这些标志存储器为手持单元的点动键预留。

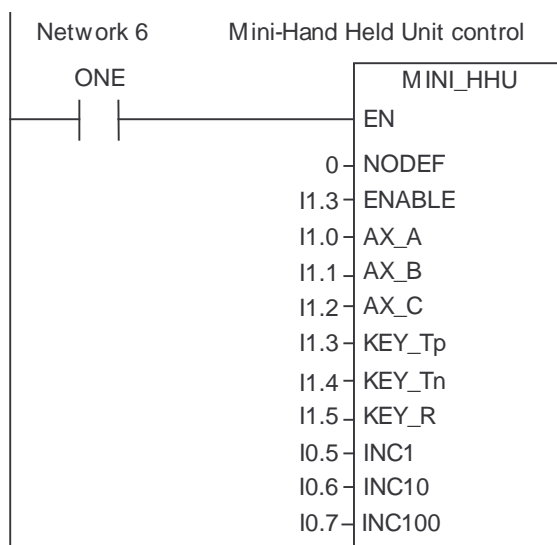
如果HHU\_ACT为“1”，PLC停止由HMI（VB19001003 & VB19001004）  
选择手轮，手轮可以通过输入信号选择。

相关的PLC机床参数 无

子程序流程示意图



子程序调试实例



9.2.6. 子程序 38 - MCP\_NCK (MCP信号处理)

**目的** 该子程序的目的是传递机床控制面板以及来自操作面板的信息到NCK接口。

主要功能有：

1. 操作方式选择
2. 增量选择
3. 操作面板信号激活（如程序控制、手轮选择）
4. 坐标手动控制（根据PLC参数控制点动方向）

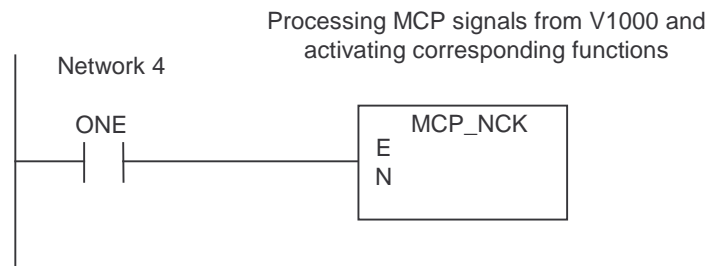
在该子程序中，子程序SBR34-X\_CROSS和子程序SBR39-HMI\_HW被调用。

**所使用的局部变量** 无

**占用的标志存储器** 无

**相关的PLC机床参数** 无

子程序调试实例



### 9.2.7. 子程序39-HMI\_HW（根据操作面板信号选择手轮）

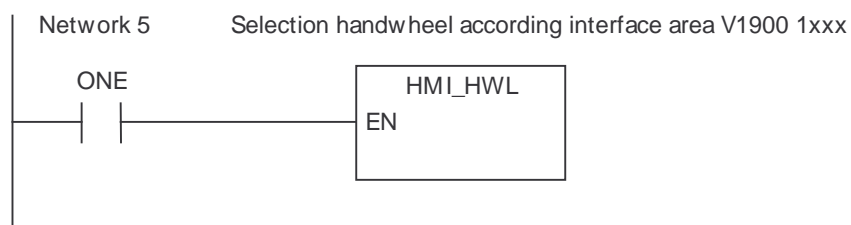
**目的** 该子程序根据操作面板信号（V19001xxx）在机床坐标系或工件坐标系下选择手轮。  
该子程序在子程序SBR38中被调用。

**所使用的局部变量** 无

**占用的标志存储器** 无

**相关的PLC机床参数** 无

#### 子程序调试实例

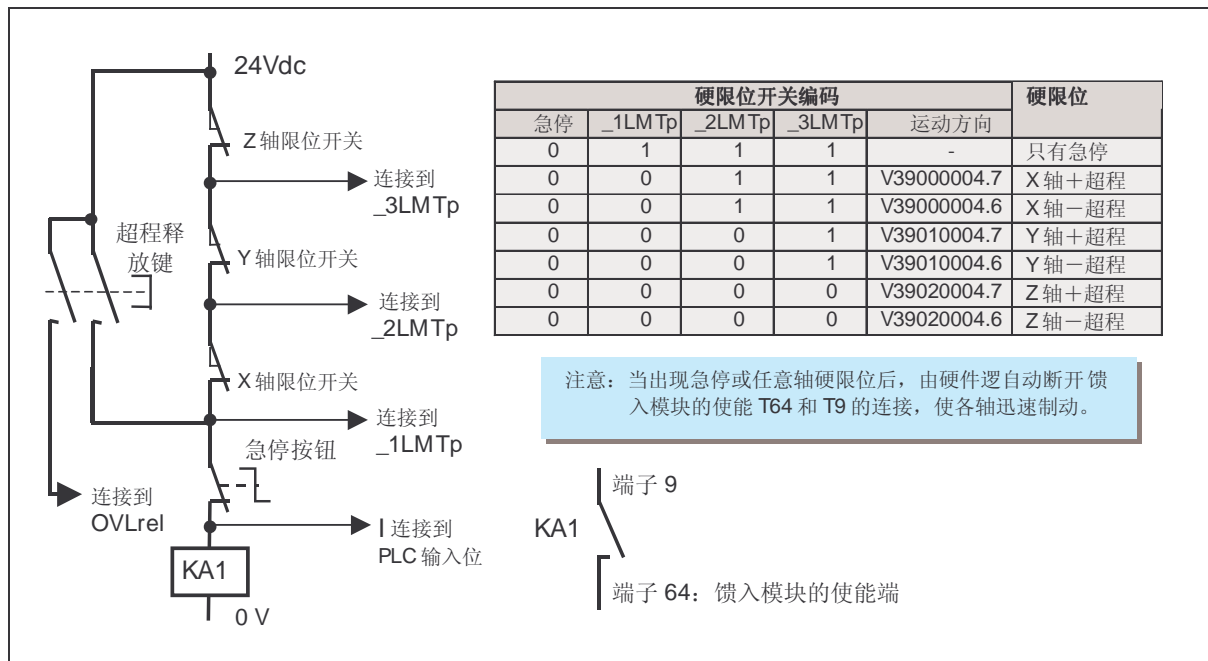


### 9.2.8. 子程序 40-AXES\_CTL (进给轴主轴使能控制)

**目的**

该子程序的目的是对NCK接口的使能 (V380x0002.1)， 监控硬限位， 参考点减速开关， 电机抱闸释放等。另外该子程序提供了两种硬限位的方案：一种是PLC方案 (MD14512[18]bit7=0)， 另一种是硬件方案 (MD14512[18]bit7=1)。PLC方案每个进给轴需配备两个 (也可以一个) 硬限位开关。当使用硬件方案 (即超程链) 时， 限位开关的的接线必须严格按照下图连接。

主轴的使能由主轴命令 (手动、M03、M04、摆动、定向) 激活。主轴使能可以在无主轴命令且主轴已经停止后按操作面板上的主轴停止键取消使能， 或通过PLC机床参数MD14512[16]Bit1选择在没有主轴命令且主轴停止后自动取消。



如果使用硬件方案，需要为该子程序提供一个超程释放的状态信号—“OVLrel” (NO)，以确保超程释放后，点动的方向仍然受到系统的控制。

**所使用的局部变量**

**输入:**

名称	数据类型	说明
NODEF	WORD	保留字
T_64	BOOL	驱动器馈入模块端子64的状态 (NO)
T_63	BOOL	驱动器馈入模块端子63的状态 (NO)
SPAD	BOOL	主轴使能自动取消 (NO)
OPTM	BOOL	驱动器优化时释放抱闸的开关 (NO)
OVLrel	BOOL	超程释放 (NO)
_1LMTp	BOOL	X轴正向限位 (NC) *
_1LMTn	BOOL	X轴负向限位 (NC)
_1REF	BOOL	X轴参考点开关 (NO)
_2LMTp	BOOL	Y轴正向限位 (NC) *
_2LMTn	BOOL	Y轴负向限位 (NC)
_2REF	BOOL	Y轴参考点开关 (NO)

_3LMTp	BOOL	Z轴正向限位 (NC) *
_3LMTn	BOOL	Z轴负向限位 (NC)
_3REF	BOOL	Z轴参考点开关 (NO)

\*注意: 如果使用PLC方案, 但当每个轴只有一个限位开关时, 或使用超程链时, 该限位开关信号应连接到正向限位输入端, 如\_1LMTp。

#### 输出:

_BRK	BOOL	抱闸制动输出 (高电平抱闸锁紧)
MSG1	BOOL	信息: 优化时抱闸已经释放
MSG2	BOOL	信息: 步进驱动器旋转监控生效, 必须返回参考点

#### 占用的标志存储器

X_M_DIR	V14000063.0	X 记忆运动方向—1: 正向; 0: 反向
Y_M_DIR	V14000063.1	Y 记忆运动方向—1: 正向; 0: 反向
Z_M_DIR	V14000063.2	Z 记忆运动方向—1: 正向; 0: 反向

#### 相关的PLC机床参数

MD14512[18] Bit 7 –	1: 超程链生效; 0: PLC硬限位方案生效
MD14512[18] Bit 6/5/4 –	1: 每轴只有一个限位开关; 0: 每轴两个限位开关
MD14512[18] Bit 1 –	1: 信号#OPTM可以释放电机制动; 0: 信号#OPTM电机制动

#### 注意:

- (1) 当对611U驱动器进行特性优化时必须释放电机的抱闸, 为此该子程序设计了优化抱闸释放开关#OPTM。该开关只有在机床参数MD14512[18]的第一位等于“1”时, 才起作用;
- (2) 急停具有最高优先级——即使抱闸由#OPTM释放, 急停仍然可以将抱闸锁紧;
- (3) 下面的报警可以由该子程序产生

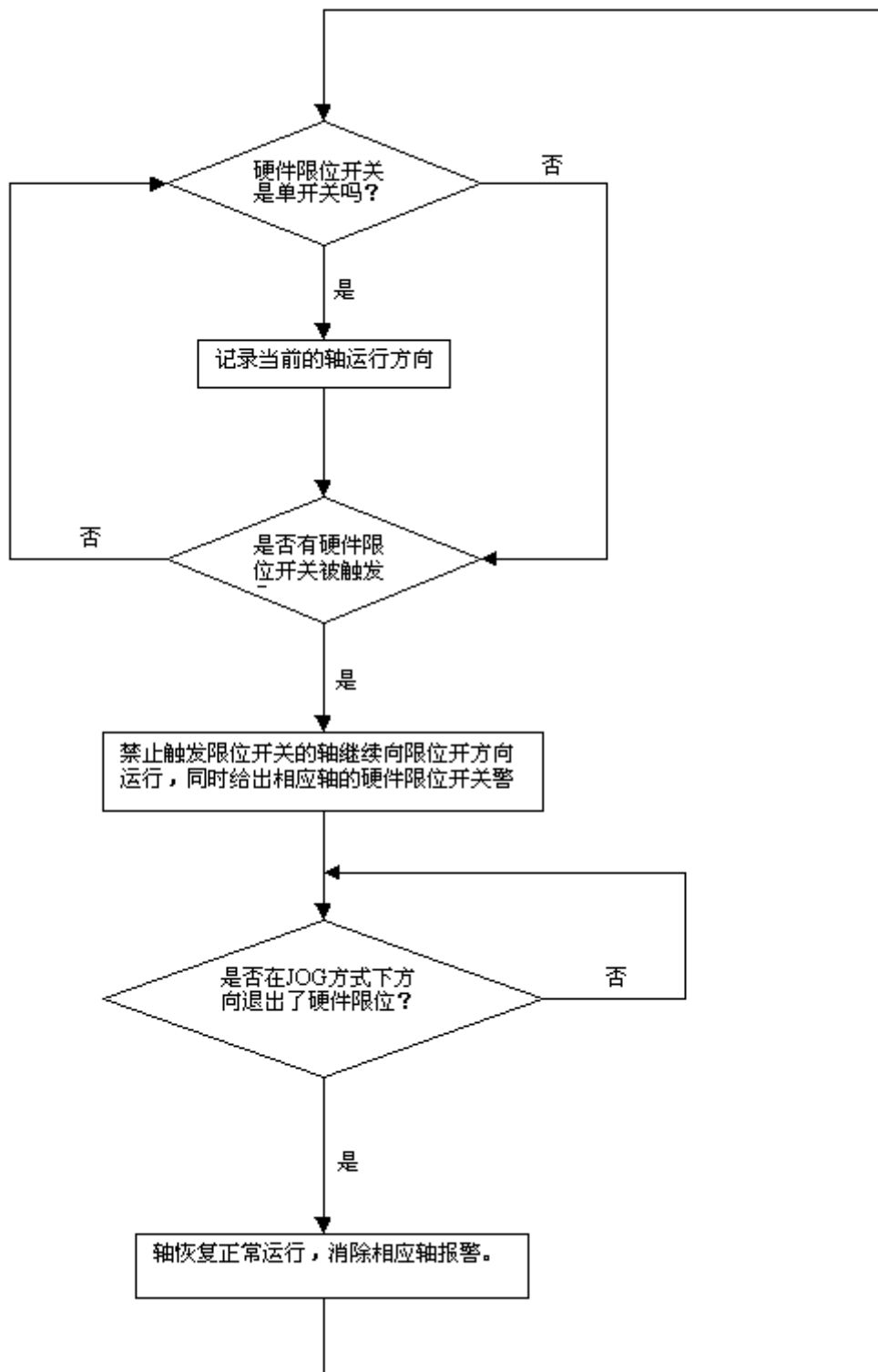
ALARM700026 – BRAKE RELEASED FOR DRIVE OPTIMIZATION



#### 重要

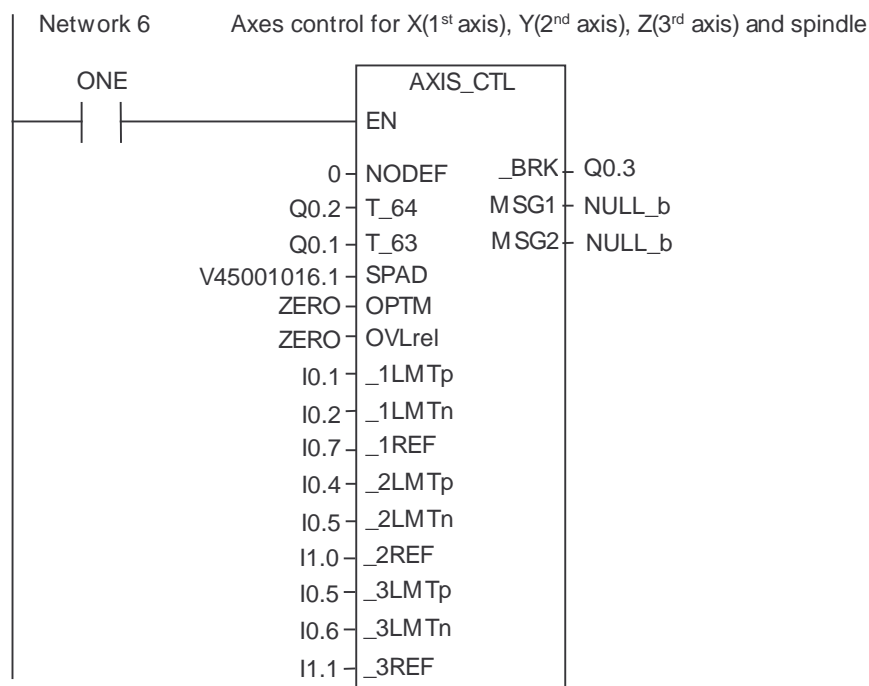
驱动器优化完毕后, 应立即将机床参MD14512[18]的第一位清零

子程序流程示意图





## 子程序调试实例



## 9.2.9. 子程序 41 - GEAR\_CHG (主轴换挡控制)

## 目的

该子程序用于模拟主轴（或具有 $\pm 10V$ 给定的变频主轴）换挡控制。

主轴具有高低两极速度，每挡均具有到位检测信号。换挡可以通过编程的速度自动激活，也可以通过M41（低档）和M42（高档）实现。该子程序可以用于铣床，也可以用于没有刀架的车床。

## 所使用的局部变量

## 输入：

D_CHG	WORD	换挡延时（单位：0.1秒）
D_MON	WORD	监控延时（单位：0.1秒）
D_S0	WORD	主轴停止延时（单位：0.1秒）
S_hold	BOOL	主轴零速信号（NO）
S_alarm	BOOL	主轴报警（NO）
LHi	BOOL	低档到位信号（NO）
HGi	BOOL	高档到位信号（NO）

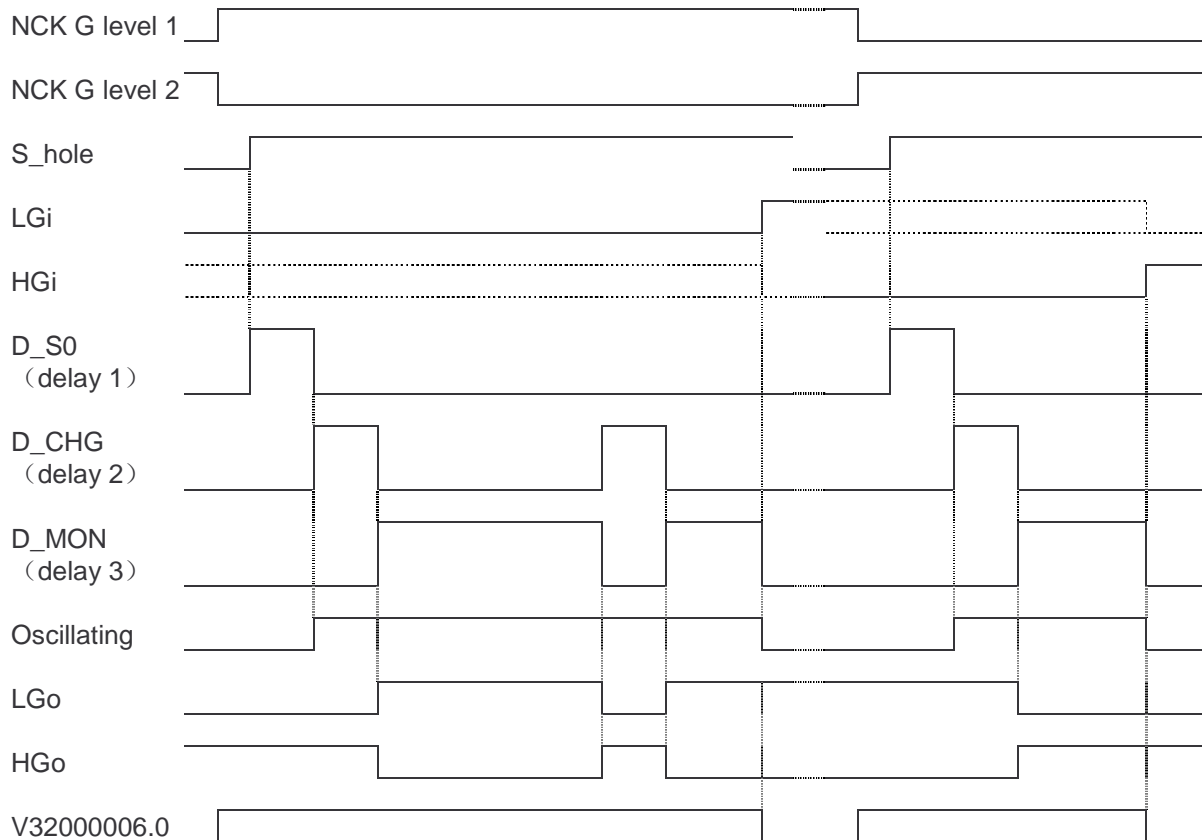
注意 1: D\_S0      -    主轴停止到主轴摆动开始的延时

注意 2: D\_CHG    -    主轴摆动开始到换挡机构动作的延时

注意 3: D\_MON    -    从换挡机构动作到换挡到位监控延时。  
如果换挡超时，自动重复换挡动作。

<b>输出:</b>			
LGo	BOOL	低档输出	
HGo	BOOL	高档输出	
LG_in	BOOL	低档到位显示	
HG_in	BOOL	高档到位显示	

**主轴自动换刀的控制时序图**



注：信号NCK G level 1 表示编程的主轴转速在第一挡范围内，或编程了M41；  
 信号NCK G level 2 表示编程的主轴转速在第二挡范围内，或编程了M42。

**占用的标志存储器**

MB117	用于记忆中间状态
T11/T14/T15	使用丝杠定时器



**重要**

由于定时器T14和T15被子程序TURRET1和GEAR\_CHG公用，所有这两个子程序不能同时使用。

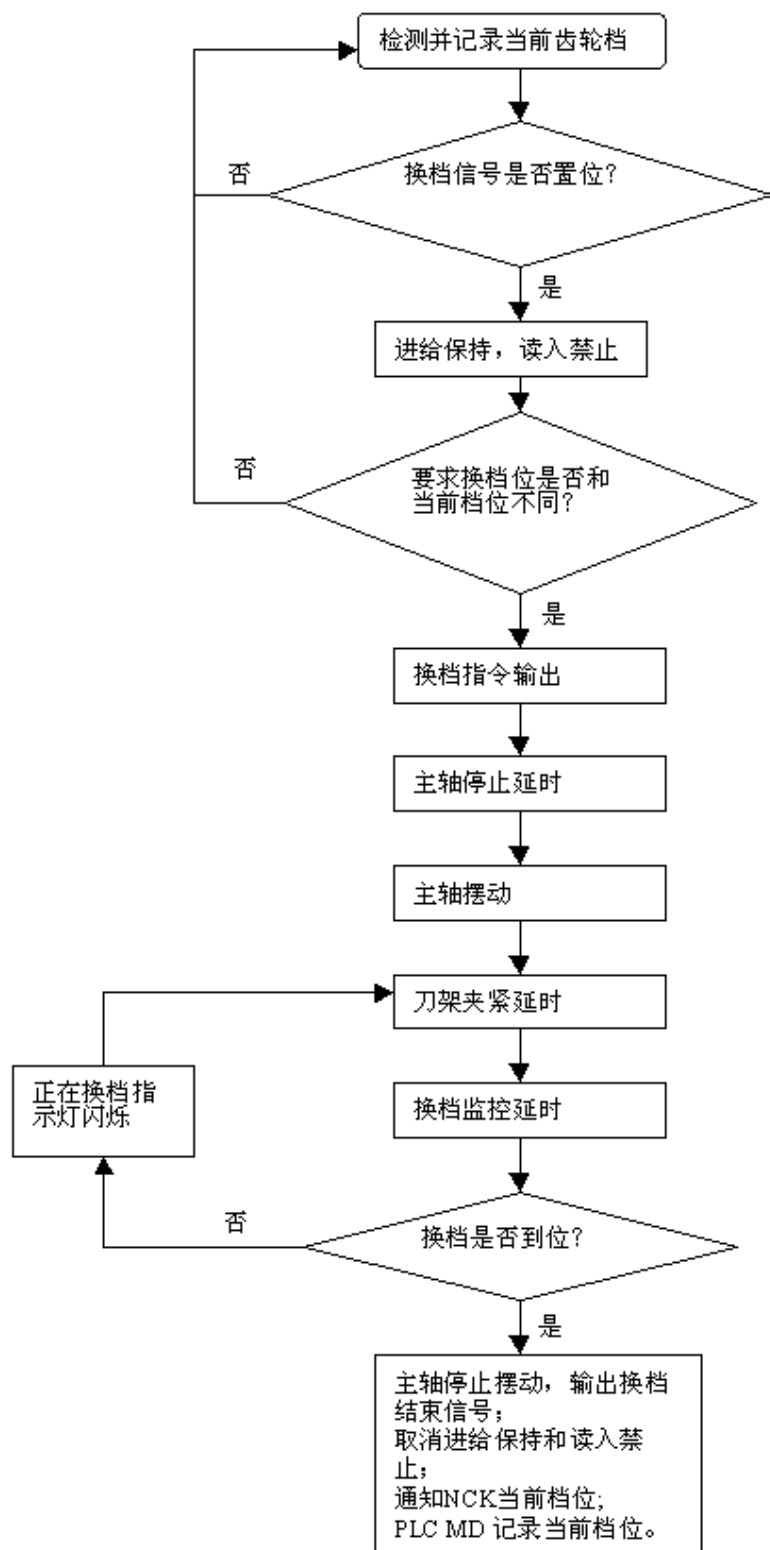
相关的PLC机床参数 无

但需要设定以下的NC参数：

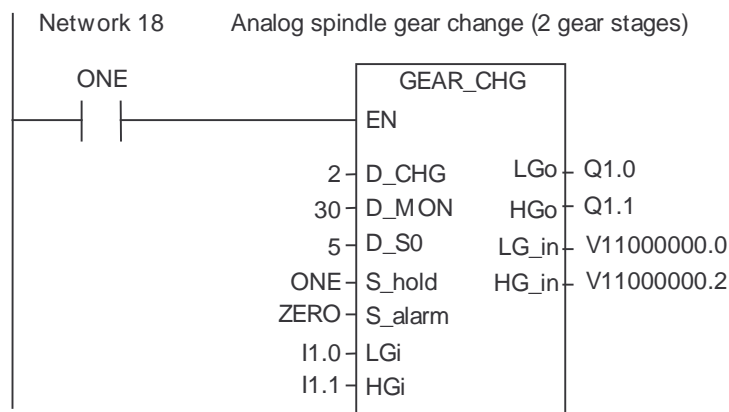
35010	GEAR_STEP_CHANGE_ENABLE	
—	主轴换挡使能	
35110	GEAR_STEP_MAX_VELO	
转/分钟	各挡的最高换挡速度，齿轮级 [0, 1...5]	
35120	GEAR_STEP_MIN_VELO	
转/分钟	各挡的最低换挡速度，齿轮级 [0, 1...5]	
35130	GEAR_STEP_MAX_VELO_LIMIT	
转/分钟	各挡的最高换挡速度限制，齿轮级 [0, 1...5]	
35140	GEAR_STEP_MIN_VELO_LIMIT	
转/分钟	各挡的最低换挡速度限制，齿轮级 [0, 1...5]	
36200	AX_VELO_LIMIT	
毫米/分钟，RPM	各挡的监控速度，齿轮级 [0, 1...5]	
35400	SPIND_OSCILL_DES_VELO	
转/分钟	主轴摆动速度	
35410	SPIND_OSCILL_ACCEL	
转/秒 <sup>2</sup>	主轴摆动加速度	
35430	SPIND_OSCILL_START_DIR	
—	主轴摆动的起始方向	
35440	SPIND_OSCILL_TIME_CW	
—	主轴正向摆动时间	
35450	SPIND_OSCILL_TIME_CCW	
—	主轴方向摆动时间	

注：详细信息请参考“系统安装调试手册”

子程序流程示意图



## 子程序调试实例



## 9.2.10. 子程序 44 - COOLING (冷却液控制)

## 目的

冷却液控制子程序可以通过机床控制面板的控制键启动或停止冷却，也可以在自动或MDA方式下利用M07或M08启动冷却、以M09停止冷却。在急停、冷却电机过载、冷却液位过低或在程序测试等情况下终止冷却输出。另外子程序设计了冷却终止输入，例如在机床防护门打开时，可停止冷却。

两个报警可以通过该子程序生成：

Alarm 700018 – COOLING MOTOR OVERLOAD

Alarm 700019 – COOLANT LEVEL LOW

## 所使用的局部变量

## 输入：

NODEF	WORD	保留字
C_key	BOOL	手动冷却启动键（触发信号）
OVload	BOOL	冷却电机过载（NC）
C_low	BOOL	冷却液位低（NC）
C_dis	BOOL	冷却输出禁止（NC）

## 输出：

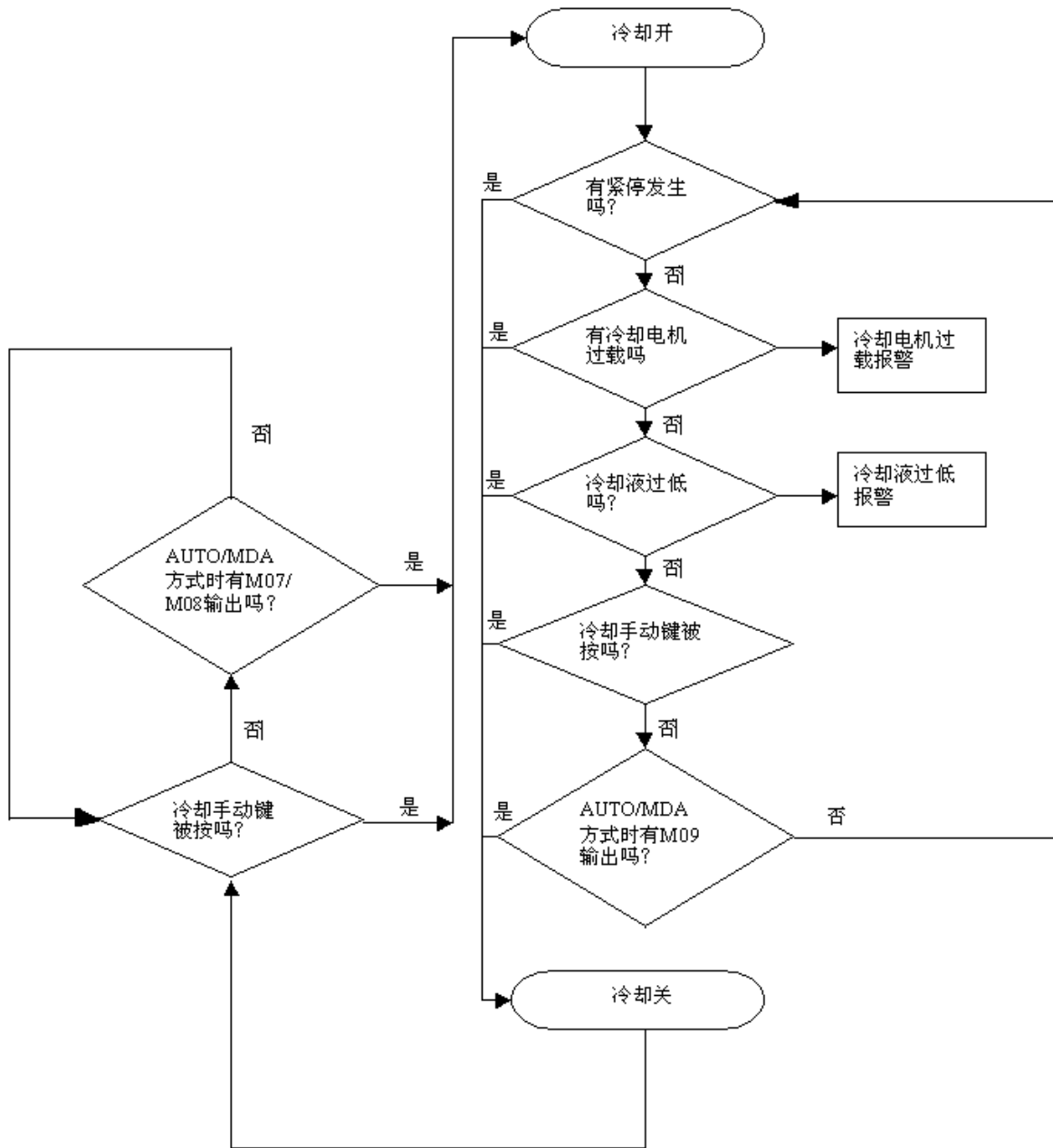
C_out	BOOL	冷却输出
C_LED	BOOL	冷却状态显示
ERR1	BOOL	错误：冷却电机过载
ERR2	BOOL	错误：冷却液位过低

## 占用的标志存储器

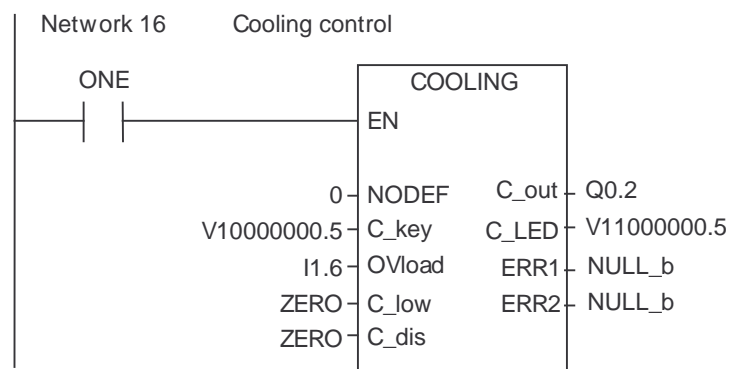
COOLon	M105.1	冷却状态
--------	--------	------

## 相关的PLC机床参数 无

子程序流程示意图



子程序调试实例



### 9.2.11. 子程序 45 - LUBRICAT (润滑控制)

#### 目的

润滑控制子程序，按每个设定的间隔自动启动一次润滑，每次润滑的时间由参数设定。通过手动按键也可以启动一次润滑。另外还可以设定的每次开机自动润滑一次。在急停、润滑电机过载、或润滑液位过低等情况下润滑停止。

两个报警可以通过该子程序生成：

Alarm 700020 – LUBRICATING MOTOR OVERLOAD

Alarm 700021 – LUBRICANT LEVEL LOW

#### 所使用的局部变量

##### 输入：

Lintv	WORD	润滑间隔（单位：1分钟）
Ltime	WORD	每次润滑时间（单位：0.1秒，最大327.67秒）
L_key	BOOL	手动润滑键（触发信号）
L1st	BOOL	上电润滑设定
Ovload	BOOL	润滑电机过载（NC）
L_low	BOOL	润滑液位过低（NC）

##### 输出：

L_out	BOOL	润滑输出
L_LED	BOOL	润滑状态显示
ERR1	BOOL	错误：润滑电机过载
ERR2	BOOL	错误：润滑液位过低

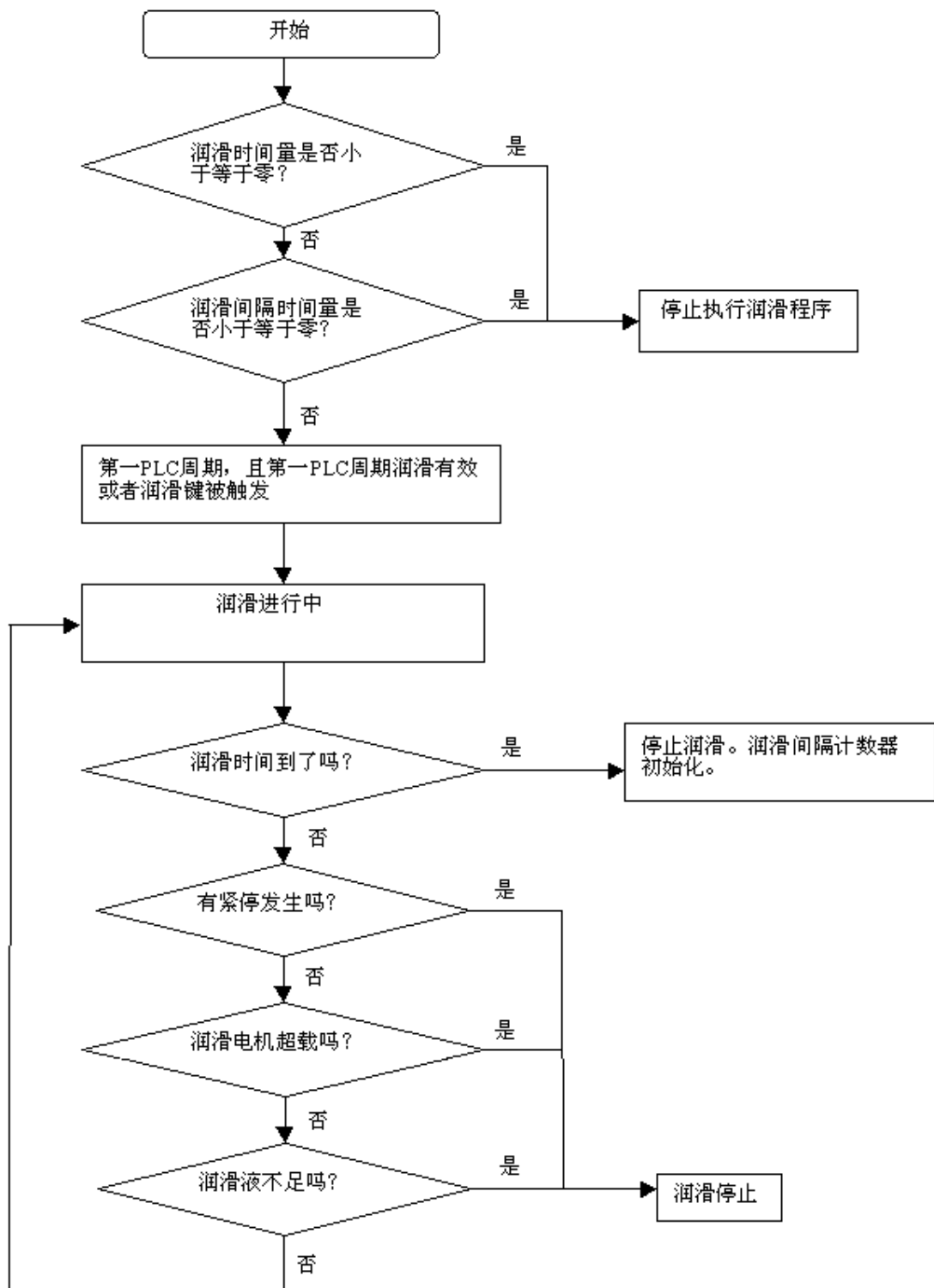
#### 占用的标志存储器

L_cmd	M105.4	润滑命令
L_interval	C30	作为润滑间隔定时器（单位：1分钟）
L_time	T13	作为润滑时间定时器（单位：0.1秒）

#### 相关的PLC机床参数

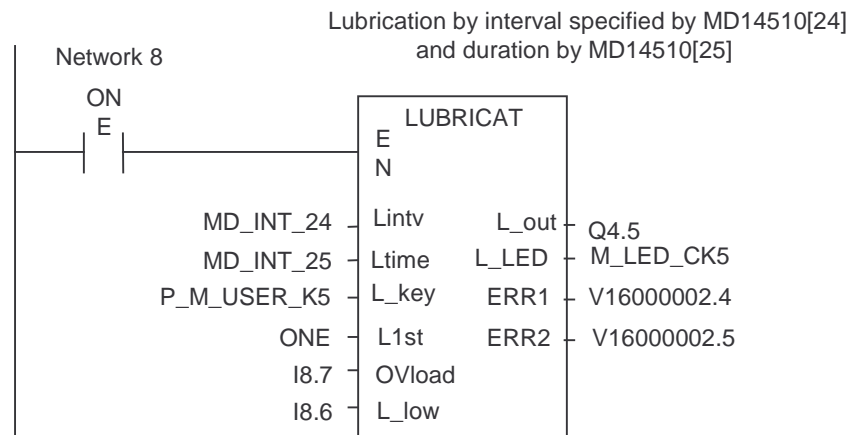
MD 14510 [24]:	润滑间隔（单位：1分钟）
MD 14510 [25]:	润滑时间（单位：0.1秒）

子程序流程示意图





子程序调试实例



**9.2.12. 子程序 46 -TURRET1 (霍尔元件刀架控制)**

**目的**

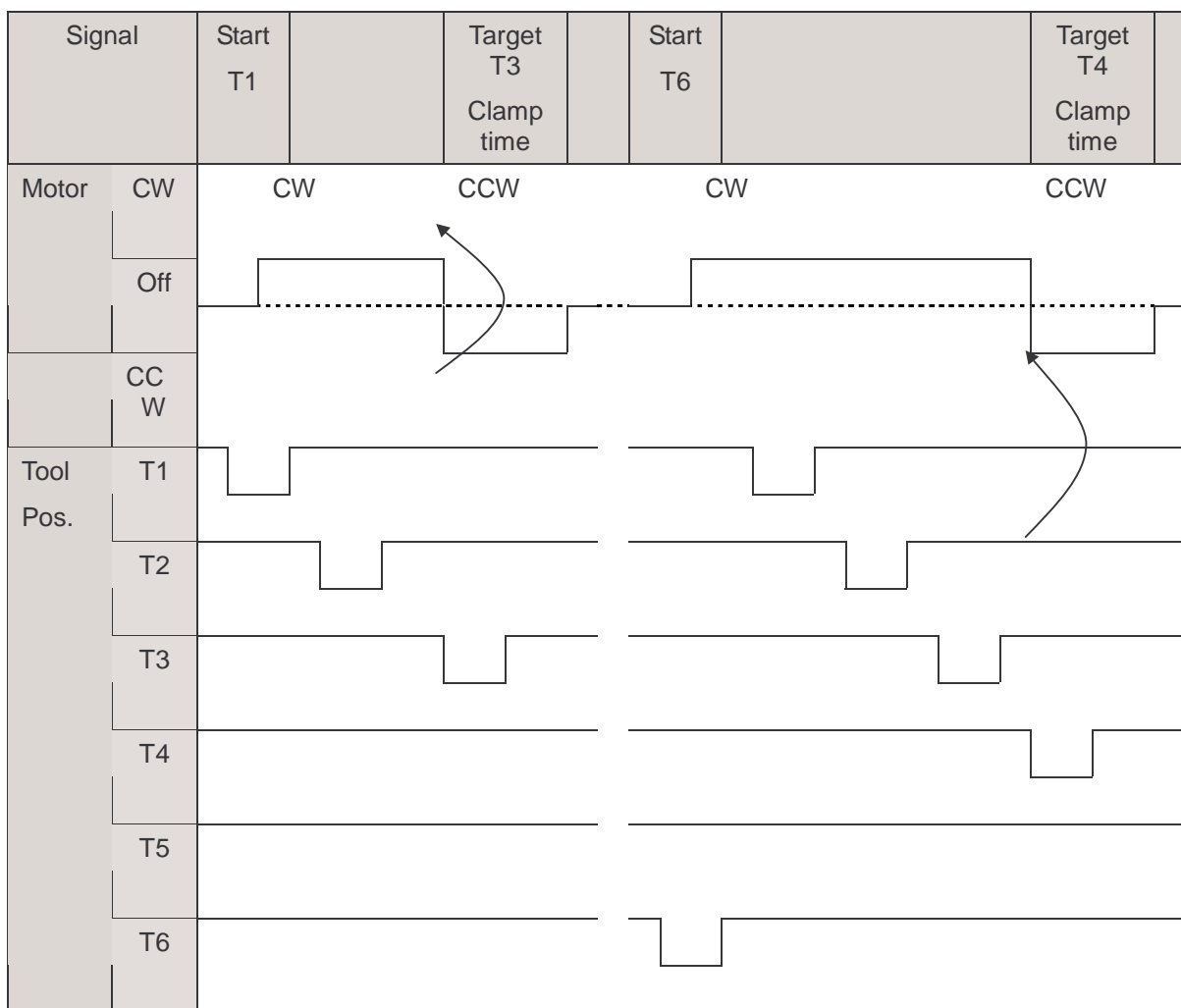
该子程序控制由霍尔元件作为位置检测的简易车床刀架。子程序的两个输出控制两个接触器实现刀架电机的正转和反转。刀架正转为寻找刀具，刀架反转为锁紧定位。

在自动或MDA方式下可以通过T编程指令启动自动换刀，也可在手动方式下，利用手动换刀键启动换刀。一个短促的按键可以换相邻的一个刀具。

在刀架转动过程中，接口信号“读入禁止”(V32000006.1)“进给保持”(V32000006.1)自动置位，直到换刀结束。这样在换刀过程没有完成时，加工程序停止等待换刀结束。

在急停、刀架电机过载、或程序测试生效等情况下，换刀被禁止。

简易刀架的换刀时序：



两个报警可以由该子程序生成

Alarm 700022 – TURRET MOTOR OVERLOAD

Alarm 700023 – PROGRAMMED TOOL NUM. > MAX. TURRET  
NUMBER

#### 所使用的局部变量

##### 输入:

Tmax	WORD	刀架刀位数（4或6或8）
C_time	WORD	刀架锁紧（反转）时间（单位：0.1秒）
M_time	WORD	刀架监控时间（单位：0.1秒）
T_01 ... T_07	BOOL	刀位检测输入（NO）
T_key	BOOL	手动换刀键（触发信号）
OVload	BOOL	刀架电机过载（NC）

##### 输出:

T_cw	BOOL	刀架定位输出
T_ccw	BOOL	刀架锁紧输出
T_LED	BOOL	刀架工作状态显示
ERR1	BOOL	错误：无刀架定位信号
ERR2	BOOL	错误：编程刀具号大于刀架刀位数
ERR3	BOOL	错误：找刀监控时间超出
ERR4	BOOL	错误：刀架电机过载

#### 占用的标志存储器

ClampTime	MD108	当前刀位寄存器
Status bits	MB112	见符号表 SBR_MEM
Status bits	MB113	见符号表 SBR_MEM
C_TIMER	T14	刀架锁紧延时
M_TIMER	T15	刀架监控延时

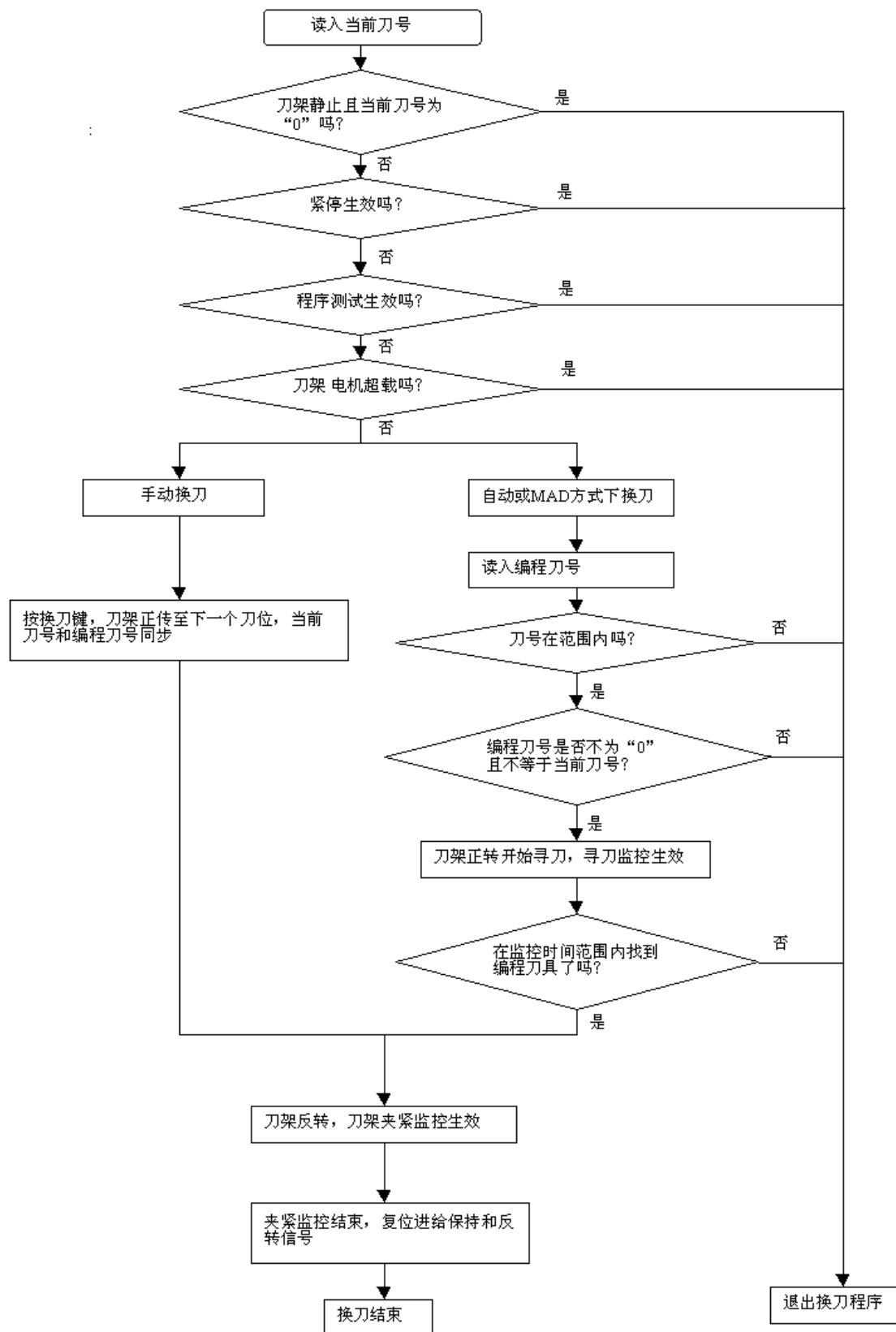
#### 相关的PLC机床参数

MD14510[20]: 刀架刀位数 4 / 6 / 8

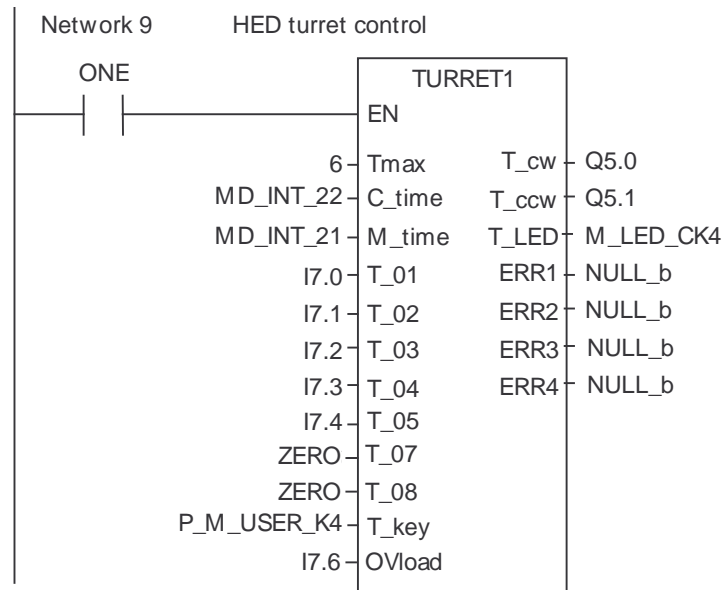
MD14510[21]: 刀架监控时间（单位：0.01s 最大. 200 单位）

MD14510[22]: 刀架锁紧时间（单位：0.01s 最大. 30 单位）

子程序流程示意图



## 子程序调试实例

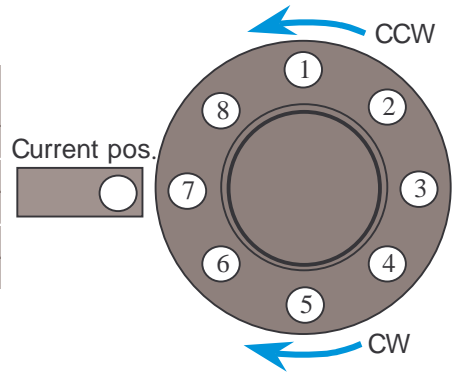


**9.2.13. 子程序 48 - TOOL\_DIR (换刀方向确定)**

目的 该子程序的目的是计算出最短路径找刀的方向以及在该方向上的预停刀位 (在最短路径方向上, 目标刀位的前一个刀位)。该子程序可以用于双向换刀的刀架或加工中心的刀库。

Examples

#	Curr. Pos.	Prog. Pos.	pre-ind Pos	Dir.
1	7	2	1	CCW
2	7	5	6	CW
3	3	8	1	CW
4	1	4	3	CCW
5	6	8	7	CCW



所使用的局部变量

输入:

Tmax            DWORD      刀位总数  
 Pnum           DWORD      编程刀号  
 Pcurr           DWORD      当前刀位

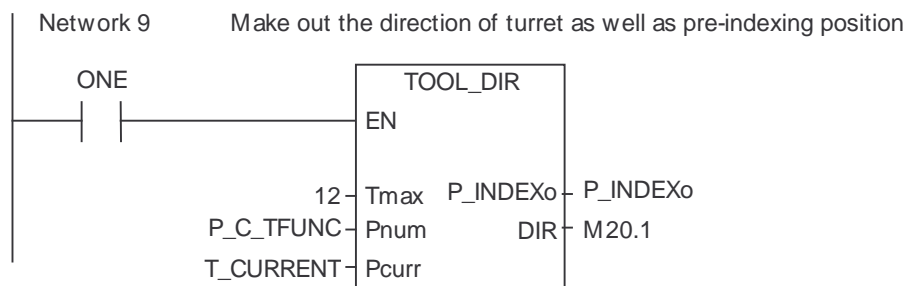
输出:

P\_INDXo        DWORD      在最短路径上的预停刀位  
 DIR            BOOL        换刀方向: 1—CW; 0—CCW

占用的标志存储器 无

相关的PLC机床参数 无

子程序调试实例



### 9.2.14. 子程序 49 - LOCK\_UNL (卡紧放松控制)

#### 目的

该子程序可以用于铣床的刀具锁紧放松，也可以用于车床卡盘的锁紧放松。锁紧放松动作的条件是程序没有运行，主轴处于静止状态。在放松时NC接口的进给保持信号被置位。该子程序支持有锁紧到位传感器的装置也支持无传感器装置。对于有传感器的锁紧装置，进给保持信号一直有效直到到位信号生效。对于无传感器的锁紧装置，在锁紧输出后自动启动延时，延时时间到清除进给保持。

两个报警可以由该子程序生成：

Alarm 700018 – UNCLAMPING NOT POSSIBLE WHILE SPINDLE RUNNING

Alarm 700019 – CLAMPING NOT IN POSITION

#### 所使用的局部变量

##### 输入：

DELAY	WORD	锁紧延时（单位：2x PLC扫描周期）
CONF	BOOL	配置：0/1 – 有/无锁紧到位传感器
KEY	BOOL	锁紧放松按键（NO）
EX_K	BOOL	外部锁紧放松开关（如脚踏开关）（NO）
S0	BOOL	主轴停止状态（NO）
CLPi	BOOL	锁紧到位传感器输入（NO）

##### 输出：

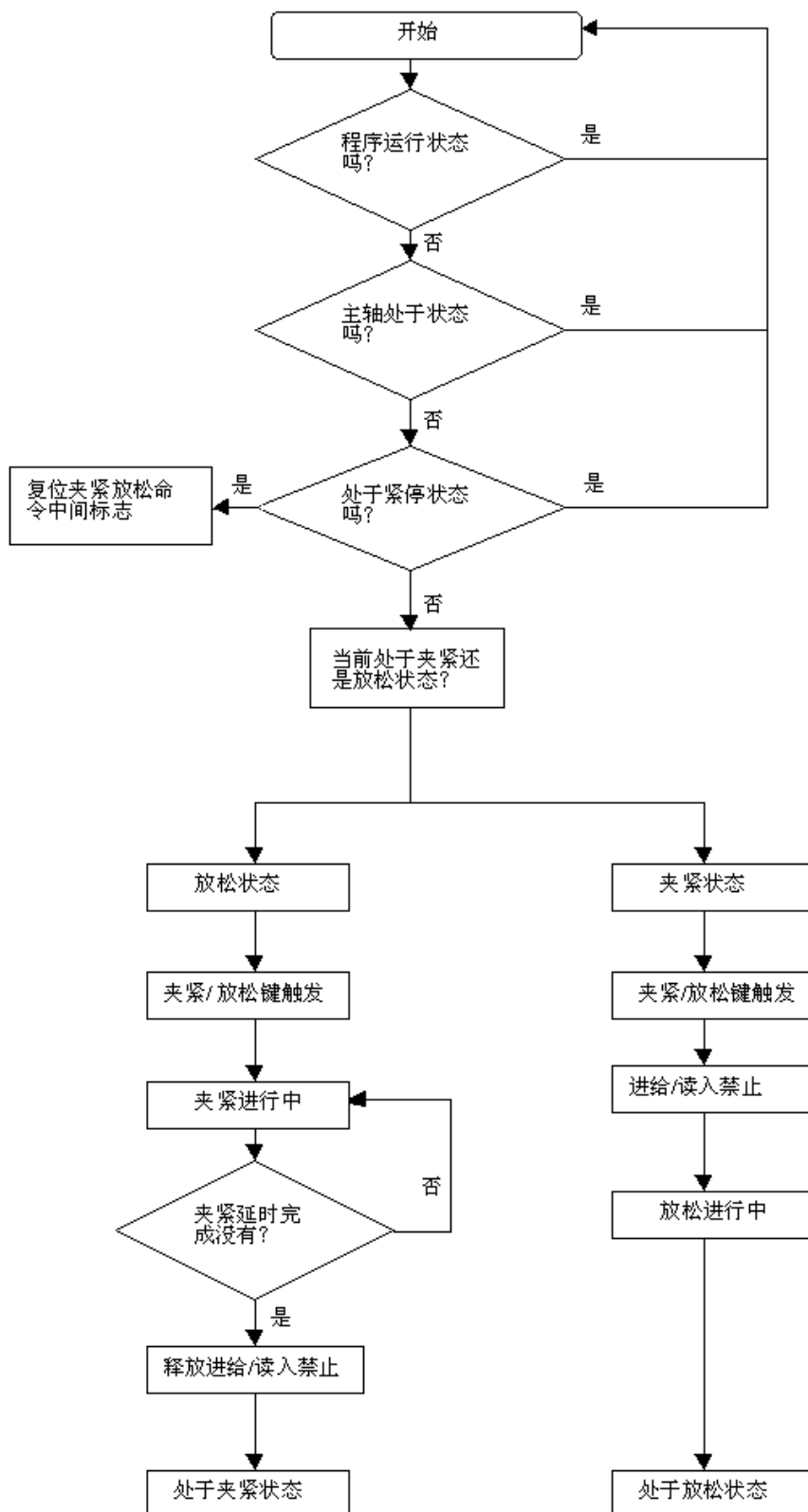
RELo	BOOL	释放输出
CLPo	BOOL	锁紧输出
C_ind	BOOL	锁紧状态显示
ERR1	BOOL	错误：主轴运行时锁紧放松无效
ERR2	BOOL	错误：锁紧没有到位

#### 占用的标志存储器

EOD	M113.5	延时结束
TR_st	M113.6	释放状态
TR_om	M113.7	释放缓冲输出

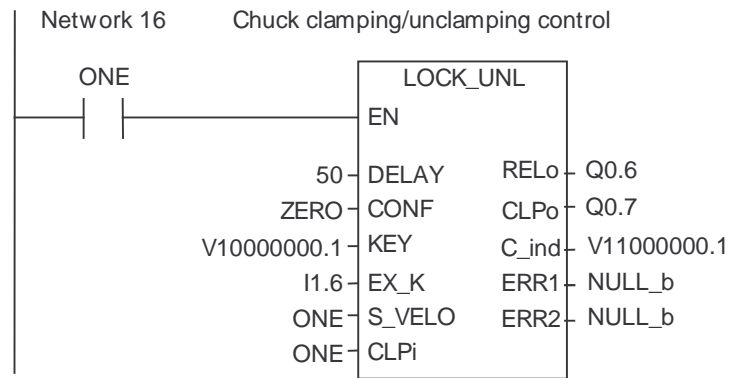
相关的PLC机床参数 无

子程序流程示意图



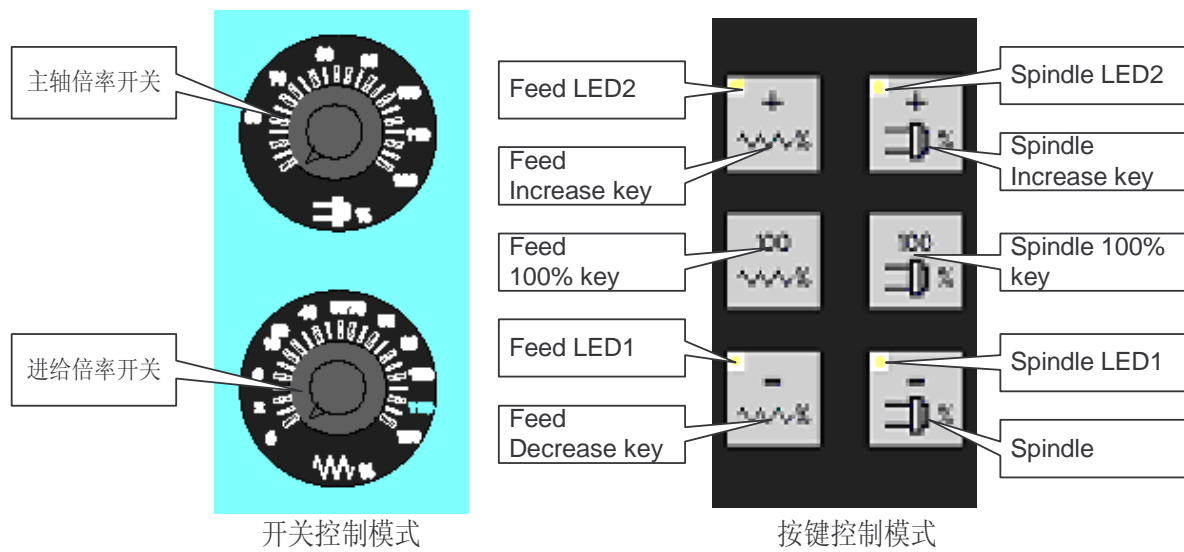


子程序调试实例



9.2.15. 子程序51 - Trg\_key\_OR (进给/主轴超调)

目的 由于802SC baseline的设计要求，此子程序实现由6个按键和4个LED来控制进给和主轴的超调。（见下图）



## 局部变量定义

名称	变量类型	数据类型	说明
LED1_F	OUT	BOOL	当前进给倍率 LED1
DecF_key	IN	BOOL	进给倍率增加键
F_100_Key	IN	BOOL	进给倍率100%键
IncF_key	IN	BOOL	进给倍率减小键
LED2_F	OUT	BOOL	当前进给倍率 LED2
LED1_S	OUT	BOOL	当前主轴倍率LED1
DecS_key	IN	BOOL	主轴倍率增加键
S_100_Key	IN	BOOL	主轴倍率100%键
IncS_key	IN	BOOL	主轴倍率减小键
LED2_S	OUT	BOOL	当前主轴倍率LED2

## 全局变量

名称	地址	说明
FO_buffer	MB80	进给倍率缓存
SO_buffer	MB81	主轴倍率缓存
FG_buffer	MB82	进给倍率gray 码缓存
SG_buffer	MB83	主轴倍率gray 码缓存
T9	T9	进给主轴倍率减小键延时器
T10	T10	进给主轴100%键延时器
F_recorder	VB14000002	当前进给倍率记录
S_recorder	VB14000003	当前主轴倍率记录

## 相关PLC机床数据

**MD14510[12]**

定义：有关进给/主轴倍率控制的时间量设置。按住进给/主轴倍率减速键大于此设定时间值，进给/主轴倍率将直接降至0%和50%。

单位：100ms

范围：5 ~ 30（0.5 ~ 3 秒），若超出此范围，将默认为1.5秒。

**MD14510[13]**

定义：有关进给/主轴倍率控制的时间量设置。按住进给/主轴倍率100%键大于此设定时间值，进给/主轴倍率将直接变为100%。

单位：100ms

范围：5 ~ 30（0.1 ~ 3 秒），若超出此范围，将默认为1.5秒。

**MD14512[12]**

Bit 0 = 0 由用户键和LED来控制进给和主轴倍率

Bit 0 = 1 由倍率开关来控制进给和主轴倍率

Bit 2 = 0 开机进给倍率为100%

Bit 2 = 1 开机进给倍率为上次关机时的倍率值

Bit 3 = 0 开机主轴倍率为100%

Bit 3 = 1 开机主轴倍率为上次关机时的倍率值

Bit4/Bit5 定义进给倍率转换速度:

Bit5	Bit4	进给倍率转换速度
0	0	标准速度: 0%, 1%, 2%, 4%, 6%, 8%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 75%, 80%, 85%, 90%, 95%, 100%, 105%, 110%, 115%, 120%.
0	1	两倍标准速度: 0%, 2%, 6%, 10%, 30%, 50%, 70%, 80%, 90%, 100%, 110%, 120%.
1	0	约三倍标准速度: 0%, 4%, 10%, 40%, 70%, 85%, 100%, 110%, 120%.
1	1	约四倍标准速度: 0%, 2%, 10%, 50%, 80%, 100%, 120%.

Bit6/Bit7 定义主轴倍率转换速度:

Bit7	Bit6	主轴倍率转换速度
0	0	标准速度: 50%, 55%, 60%, 65%, 70%, 75%, 80%, 85%, 90%, 95%, 100%, 105%, 110%, 115%, 120%.
0	1	两倍标准速度: 50%, 60%, 70%, 80%, 90%, 100%, 110%, 120%.
1	0	约三倍标准速度: 50%, 60%, 70%, 85%, 100%, 110%, 120%.
1	1	约四倍标准速度: 50%, 60%, 80%, 100%, 120%.

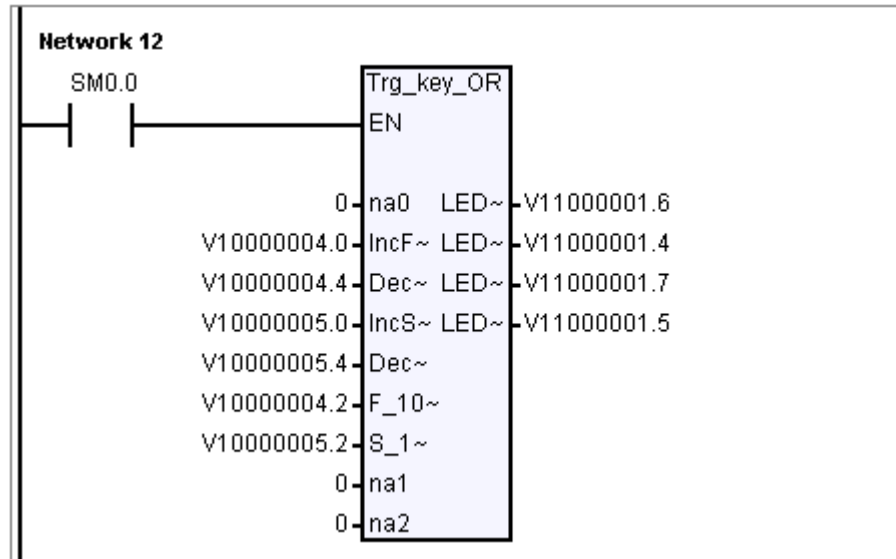
## 相关按键和LED的功能说明

名称	功能说明
Feed LED1	指示当前进给倍率值： 0%<当前进给倍率值< 100%时： 亮 当前进给倍率值= 0% 时： 闪烁（2Hz）
Feed Decrease key	每次按此键，进给倍率减小一级。 按住此键超过 PLC MD14510[12]设定的时间，进给倍率直接变为0%
Feed 100% key	按住此键超过 PLC MD14510[13]设定的时间，进给倍率直接变为100%
Feed Increase key	每次按此键，进给倍率增加一级
Feed LED2	指示当前进给倍率值： 100% <当前进给倍率值 <120%时： 亮 当前进给倍率值： =120%时： 闪烁（2Hz）
Spindle LED1	指示当前主轴倍率值： 50%< 当前主轴倍率值< 100%： 亮 当前主轴倍率值= 50%： 闪烁（2Hz）
Spindle Decrease key	每次按此键，主轴倍率减小一级。 按住此键超过 PLC MD14510[12]设定的时间，主轴倍率直接变为0%
Spindle 100% key	按住此键超过 PLC MD14510[13]设定的时间，主轴倍率直接变为100%
Spindle Increase key	每次按此键，主轴倍率增加一级
Spindle LED2	指示当前主轴倍率值： 100% <当前主轴倍率值<120%： 亮 当前主轴倍率值=120%： 闪烁（2Hz）

## 调用举例

- (1) 设置 MD14512[12] bit 0 = 0（由按键和LED控制进给主轴倍率）  
 设置 MD14512[12] bit 2 = 0（开机进给倍率为 100%）  
 设置 MD14512[12] bit 3 = 0（开机主轴倍率为100%）  
 设置 MD14512[12] bit 4/5 =00（进给倍率标准变换。）  
 设置 MD14512[12] bit6/7 =00（主轴倍率标准变换。）  
 设置 MD14510[12] = 10（按住进给/主轴倍率减小键超过1秒，进给/主轴倍率直接变为0%）  
 设置MD14510[13] = 15（按住进给/主轴100%键超过1.5秒，进给/主轴倍率直接变为100%）

(2) 在OB1 中调用此子程序



### 9.3. PLC实例应用程序说明

#### 概述

SAMPLE.PTP是利用子程序库搭建的一个完整的PLC应用程序。设计该应用程序有两个主要目的：

(1) 可以直接用于那些其功能所覆盖的机床，只需要通过修改PLC机床参数，即可实现对其功能的设定；

(2) 它可以作为应用实例。通过实例可以表明如果利用子程序库搭建自己的应用程序。当然也可以修改该实例程序已适应您的特殊要求。

与上一个版本中的集成PLC应用实例相同，实例程序的所有输入和生成都通过滤波器处理。在后面的章节里会对滤波器的工作原理进行介绍，以便您能了解并修改实例程序。当然如果您利用子程序库搭建自己的应用程序，您可以直接对输入输出寻找，而不需要滤波器。

#### 9.3.1. 主要功能

##### PLC 初始化

- 激活测量系统 1
- 通道和轴接口的进给倍率生效
- 参数有效性检测（在子程序31中实现）

##### 急停处理

- 急停按钮处理
- 611馈入模块上电下电时序控制（T48, T63, T64）
- 611馈入模块的状态监控（T72—驱动器就绪, T52—I<sup>2</sup>t报警）（这两个状态反馈信号也可以激活急停）

##### MCP和HMI信号处理





- 操作方式选择
- NC启动、停止、复位
- 主轴手动操作（主轴正转、反转和停止）
- 点动键处理（根据PLC参数）
- 由HMI接口选择手轮（SBR39 HMI\_HW）

##### 坐标控制

- 各个坐标的使能控制（包括主轴）
- 硬限位处理（单或双开关逻辑）或超程链
- 参考点开关监控
- 步进驱动器的旋转监控
- 进给电机抱闸释放

##### 接触器控制的主轴

- 主轴使能
- 主轴手动操作（手动方式下正转、反转和停止）
- 在自动或MDA方式下，可直接编程M03和M04，PLC自动在M03和M04之间加入制动输出
- 外部主轴制动控制

- 模拟主轴**
- 双极性模拟主轴 (+/- 10V)
  - 双极性主轴使能由  
802S baseline的 X3 (Pin5 和 Pin 9)  
802C baseline 的 X7 (Pin4/65 和 Pin4/9)
  - 主轴手动操作 (手动方式下正转、反转和停止)
  - 主轴程序控制 (自动和MDA方式)
- 车床刀架控制**
- 适用于4或6工位霍尔元件简易刀架
  - 刀架锁紧监控 (锁紧时间由PLC机床参数设定)
  - 刀架刀位反馈监控
  - 换刀过程监控 (如果在一定时间内没有找刀目标刀具, 自动停止)
  - 换刀时进给停止
- 冷却液控制**
- 手动方式下用户键K6启动或关闭冷却
  - 自动和MDA方式下M07、M08启动冷却, M09关闭冷却
  - 冷却液位和冷却电机过载监控
- 导轨润滑**
- 用户键K5启动润滑一次。
  - 定时定量润滑 (根据PLC参数设定的间隔和润滑时间)
- 卡紧放松控制**
- 用于车床的卡盘卡紧和放松
  - 用于铣床的刀具卡紧和放松
- 伺服驱动器优化时  
刀具抱闸的释放控制**
- 组合键由PLC机床参数激活 (只用于611U对带抱闸电机的优化)
  - 组合键:
    -  +  à 抱闸释放;
    -  +  à 抱闸锁紧
  - 抱闸释放时可以产生PLC报警以提示调试人员注意

## 9.3.2. 输入输出定义

输入信号说明		
信号名	用于车床: X2003	用于铣床: X2003
I0.0	硬限位 X+	硬限位 X+
I0.1	硬限位 Z+	硬限位 Z+
I0.2	X 参考点开关	X 参考点开关
I0.3	Z 参考点开关	Z 参考点开关
I0.4	硬限位 X - <sup>1)</sup>	硬限位 X - <sup>1)</sup>
I0.5	硬限位 Z - <sup>1)</sup>	硬限位 Z - <sup>1)</sup>
I0.6	过载 (611馈入模块的T52)	过载 (611馈入模块的T52)
I0.7	急停按钮	急停按钮
信号名	用于车床: X2004	用于铣床: X2004
I1.0	刀架信号 T1	主轴低档到位信号
I1.1	刀架信号 T2	主轴高档到位信号
I1.2	刀架信号 T3	硬限位 Y +
I1.3	刀架信号 T4	Y 参考点开关
I1.4	刀架信号 T5	硬限位 Y - <sup>1)</sup>
I1.5	刀架信号 T6	无定义
I1.6	超程释放信号 (用于超程链)	超程释放信号 (用于超程链)
I1.7	就绪信号 (611馈入模块的T72)	就绪信号 (611馈入模块的T72)
I2.0~5.7	未定义	未定义



输出信号说明		
信号名	用于车床: X2005	用于铣床: X2005
Q0.0	主轴正转 CW <sup>3)</sup>	主轴正转 CW <sup>3)</sup>
Q0.1	主轴反转 CCW <sup>3)</sup>	主轴反转 CCW <sup>3)</sup>
Q0.2	冷却控制输出	冷却控制输出
Q0.3	润滑输出	润滑输出
Q0.4	刀架正转 CW	无定义
Q0.5	刀架反转 CCW	无定义
Q0.6	卡盘卡紧	卡盘卡紧
Q0.7	卡盘放松	卡盘放松
信号名	用于车床: X2006	用于铣床: X2006
Q1.0	无定义	主轴低档输出
Q1.1	无定义	主轴高档输出
Q1.2	无定义	无定义
Q1.3	电机抱闸释放	电机抱闸释放
Q1.4	主轴制动	主轴制动
Q1.5	馈入模块端子T48	馈入模块端子T48
Q1.6	馈入模块端子T63	馈入模块端子T63
Q1.7	馈入模块端子T64	馈入模块端子T64

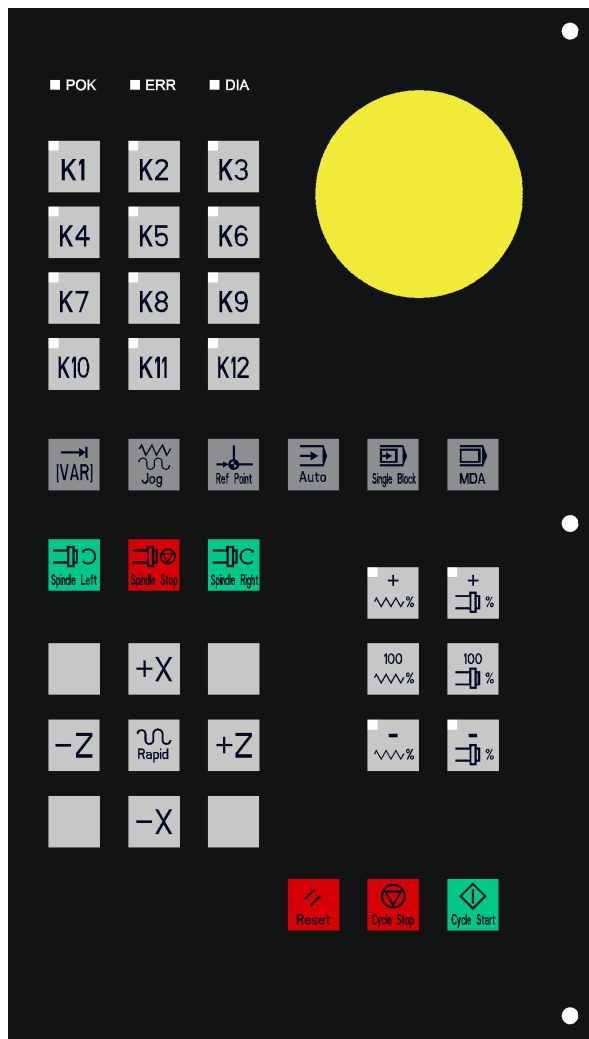
**注意:** 实例程序中所有输入信号均按常开设计。如果某个输入按常闭接线, 可通过PLC机床参数将该输入定义为负逻辑。

**注意:**

- (1) 当某轴只有一个硬限位开关时, 该输入无定义;
- (2) 当选择4工位刀架时, I1.4和I1.5无定义;
- (3) 当使用双极性主轴时, Q0.0和Q0.1无定义。

当使用单极性主轴时Q0.0和Q0.1被定义为主轴使能和方向信号, 不能再在PLC程序中使用。

### 9.3.3. MCP按键布局



## 用户键定义

用户键描述		
键号	用于车床	用于铣床
K1	驱动器使能, 或禁止	驱动器使能, 或禁止
K2	卡盘卡紧, 或放松	刀具卡紧, 或放松
K3	无定义	无定义
K4	手动换刀	无定义
K5	手动润滑启动, 或停止	手动润滑启动, 或停止
K6	手动冷却启动, 或停止	手动冷却启动, 或停止
K7~K12	未定义	未定义
指示灯	用于车床	用于铣床
LED1	驱动器已使能	驱动器已使能
LED2	卡盘已卡紧	刀具已卡紧
LED3	无定义	无定义
LED4	正在换刀	无定义
LED5	正在润滑	正在润滑
LED6	正在冷却	正在冷却
LED7~LED12	未定义	未定义

## 点动键号定义

<b>K22</b>	<b>K23</b>	<b>K24</b>
<b>K25</b>	<b>K26</b>	<b>K27</b>
<b>K28</b>	<b>K29</b>	<b>K30</b>

点动键的定义有PLC机床参数MD14510[26]~[31]定义。

**K26** 的定义为快速叠加键。

## 9.3.4. 实例程序的PLC参数定义

**PLC 机床参数** 方便为使实例程序在使用上更加灵活，实例程序的所有输入输出都进行了滤波处理。由于是实例PLC程序，PLC机床参数的定义在为预留的区域内。

MD14512		USER_DATA_HEX						
机床参数		PLC 机床参数 - 十六进制						
Index	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
14512[0]	I0.7	I0.6	I0.5	I0.4	I0.3	I0.2	I0.1	I0.0
定义有效输入位（接口 X2003，端子号：0~7）								
14512[1]	I1.7	I1.6	I1.5	I1.4	I1.3	I1.2	I1.1	I1.0
定义有效输入位（接口 X2004，端子号：8~15）								
14512[2]	I0.7	I0.6	I0.5	I0.4	I0.3	I0.2	I0.1	I0.0
定义输入位为常闭连接（接口X2003，端子号：0~7）								
14512[3]	I1.7	I1.6	I1.5	I1.4	I1.3	I1.2	I1.1	I1.0
定义输入位为常闭连接（接口X2004，端子号：8~15）								
14512[4]	Q0.7	Q0.6	Q0.5	Q0.4	Q0.3	Q0.2	Q0.1	Q0.0
定义有效输出位（接口 X2005，端子号：0~7）								
14512[5]	Q1.7	Q1.6	Q1.5	Q1.4	Q1.3	Q1.2	Q1.1	Q1.0
定义有效输出位（接口 X2006，端子号：8~15）								
14512[6]	Q0.7	Q0.6	Q0.5	Q0.4	Q0.3	Q0.2	Q0.1	Q0.0
定义输出位为低电平有效（接口X2005，端子号：0~7）								
14512[7]	Q1.7	Q1.6	Q1.5	Q1.4	Q1.3	Q1.2	Q1.1	Q1.0
定义输出位为低电平有效（接口X2006，端子号：8~15）								
14512[11]	PLC实例程序配置							
	刀架控制有效	模拟主轴换挡控制			主轴有效	卡紧放松有效	润滑有效	冷却有效

**注意：**所有参数均为“1”有效。同时还要设定子程序库的标准参数。这些参数均在子程序SBR31中进行检测。

MD14512[11] Bit 0 = 1 – 子程序 COOLING 运行有效  
 MD14512[11] Bit 1 = 1 – 子程序 LUBRICAT 运行有效  
 MD14512[11] Bit 2 = 1 – 子程序 LOCK\_UNL 运行有效  
 MD14512[11] Bit 3 = 1 – 子程序 SPINDLE 运行有效

MD14512[11] Bit 6 = 1 – 子程序GEAR\_CHG 运行有效  
 MD14512[11] Bit 7 = 1 – 子程序 TURRET1 运行有效

### 9.3.5. 实例程序结构说明

#### 9.3.5.1. 实例程序的结构

##### MAIN (OB1)

序号	SBR #	说明	
1	62	输入输出滤波 ( IWO/QW0 → MW100/MW102)	
2	32	PLC 初始化 →	SBR31 – 用户初始化
3	33	急停处理	
4	38	MCP 信号处理 →	SBR34 – 点动按控制
			SBR39 – 由HMI选择手轮
5	40	X&Y&Z&主轴使能控制	
6	44	冷却控制	
7	45	润滑控制	
8	35	主轴控制 (开关量主轴、单或双极性模拟主轴)	
9	41	刀架控制	
10	49	卡紧放松控制	
11	51	进给/主轴超调由按键控制	

##### I/O 信号处理

SAMPLE为不同的机床接线而设计，即任何输入位即可以按常开连接也可以I按常闭连接。所有输入输出均通过子程序62按PLC机床数据 MD14512[0], [1], [2], [3] 和 MD14512[4], [5], [6], [7]进行预处理。

##### I/O 滤波器原理

根据下图可以了解物理输入信号与内部缓存信号之间的关系。

SAMPLE中的所有子程序均按常开逻辑设计。在SAMPLE中M100.0表示输入位I0.0, M101.2表示I1.2; M102.3表示Q0.3, M103.4表示Q1.4依此类推。

而子程序库中的所有子程序均独立于物理输入输出。

输入信号处理 X2003:

I0.0à	逻辑“异或” MD14512[2]	逻辑“与” MD14512[0]	à M100.0
I0.1à			à M100.1
I0.2à			à M100.2
I0.3à			à M100.3
I0.4à			à M100.4
I0.5à			à M100.5
I0.6à			à M100.6
I0.7à			à M100.7

输入信号处理 X2004:

I1.0à	逻辑“异或” MD14512[3]	逻辑“与” MD14512[1]	à M101.0
I1.1à			à M101.1
I1.2à			à M101.2
I1.3à			à M101.3
I1.4à			à M101.4
I1.5à			à M101.5
I1.6à			à M101.6
I1.7à			à M101.7

输出信号处理 X2005:

M102.0à	逻辑“异或” MD14512[6]	逻辑“与” MD14512[4]	à Q0.0
M102.1à			à Q0.1
M102.2à			à Q0.2
M102.3à			à Q0.3
M102.4à			à Q0.4
M102.5à			à Q0.5
M102.6à			à Q0.6
M102.7à			à Q0.7

输出信号处理 X2006:

M103.0à	逻辑“异或” MD14512[7]	逻辑“与” MD14512[5]	à Q1.0
M103.1à			à Q1.1
M103.2à			à Q1.2
M103.3à			à Q1.3
M103.4à			à Q1.4
M103.5à			à Q1.5
M103.6à			à Q1.6
M103.7à			à Q1.7

## 实例程序的更改

以下两种情况下可以修改实例程序：

(1) 如果系统只配置了一个DIO模块，而且您需要一个子程序库没有提供的功能，您只需在子程序SBR0~SBR30中编写一个新的子程序实现所需的功能，然后在主程序中调用即可。在编写新子程序时您应该使用缓冲的输入输出，即M100.0代表I0.0，M102.0代表Q0.0。

(2) 当您的系统配置了多于一个DIO模块时，您就不能使用主程序中的滤波子程序。因为滤波子程序只对16个输入和16个输出进行缓冲。这样在您修改实例程序时，首先从主程序中删除滤波子程序FILTER，然后将主程序中所有缓冲输入输出该为直接地址。

当然您可以利用子程序库SUBR\_LIBRARY.PTP中的子程序模块搭建一个新的应用程序，这样您可以从一开始就直接使用输入输出的地址进行编程。



---

### 重要事项

在实例程序中所有的输入点均按常开时间，即高电平有效。例如，I0.7在实例程序中定义为急停，当I0.7为“1”1时，PLC激活急停。

如果实际应用中急停为常闭连接，则利用滤波器需要通过PLC参数MD14512[2]的第7位，将输入I0.7设置为常闭逻辑。

---



---

### 重要事项

如果您利用子程序库 SUBR\_LIBRARY 提供的子程序模块搭建自己的应用程序，一定要注意每个子程序输入的定义，以确定输入信号是常开还是常闭。

---

## 9.3.5.2. 实例程序中定义的用户报警

## 子程序库的标准报警信息

报警号	接口地址	报警信息	子程序
700016	V16000002.0	驱动器未就绪	33
700017	V16000002.1	馈入模拟 I <sup>2</sup> /t 报警	
700018	V16000002.2		
700019	V16000002.3		
700020	V16000002.4		
700021	V16000002.5	卡盘或刀具放松时不能启动主轴	35
700022	V16000002.6	主轴运转时不能放松卡盘或放松刀具	49
700023	V16000002.7	编程刀具号大于刀架最大刀位数	46
700024	V16000003.0	在监控时间内未能找到目标刀具	
700025	V16000003.1	刀架无位置检测信号	
700026	V16000003.2	驱动器优化时电机抱闸已释放	40
700027	V16000003.3	旋转监控生效, 请重新返回参考点	
700028	V16000003.4		
700029	V16000003.5		
700030	V16000003.6		
700031	V16000003.7		

## 实例程序定义的报警信息

报警号	接口地址	报警信息	子程序
700000	V16000000.0	初次调试! 需要设定下列PLC参数 或 按存储数据启动	31
700001	V16000000.1		
700002	V16000000.2	X+ 点动键没有定义, 请检查MD14510[26]	
700003	V16000000.3	X - 点动键没有定义, 请检查MD14510[27]	
700004	V16000000.4	Y+ 点动键没有定义, 请检查MD14510[28]	
700005	V16000000.5	Y - 点动键没有定义, 请检查MD14510[29]	
700006	V16000000.6	Z+ 点动键没有定义, 请检查MD14510[30]	
700007	V16000000.7	Z - 点动键没有定义, 请检查MD14510[31]	
700008	V16000001.0	刀架刀位数定义错, 检查MD14510[20] = 4/ 6	
700009	V16000001.1	刀架锁紧时间没有定义, 检查MD14510[21]	
700010	V16000001.2	刀架监控时间没有定义, 检查MD14510[22]	
700011	V16000001.3	主轴制动时间超出范围, 检查MD14510[23]	
700012	V16000001.4	润滑间隔超出范围, 检查MD14510[24]	
700013	V16000001.5	润滑时间超出范围, 检查MD14510[25]	
700014	V16000001.6	输入使能没有定义, 检查MD14512[0]&MD14512[1]	
700015	V16000001.7	输出使能没有定义, 检查MD14512[4]&MD14512[5]	

## 注意:

- (1) 实例程序只允许4或6工位刀架;
- (2) 实例程序中, 报警号700000~700015只在子程序31—USR\_INI中激活。而在子程序库中, 该子程序为空。



### 9.3.6. 实例程序SAMPLE的调试

系统首次上电后，出现下面的报警：

MA	RESET	AUTO	RO	700000
Number		Del cri		C95HP.MPF
700000				
初次调试！ 需要设定下列 PLC 参数： ->MD14510[16]&[26]~[30]&MD14512[0]~[7]或 按存储数据启动				
报警		维修 信息	调试	机床 参数

这时应该设定下列PLC机床参数：

- 提供MD14510[16]设定机床类型—0表示车床；1表示铣床
- 定义输入输出的使能和连接逻辑（常开或常闭）：  
MD14512[0]和[1]，[2]&[3] 定义DIO1的输入使能和输入逻辑  
MD14512[4]和[5]，[6]&[7] 定义DIO1的输出使能和输出逻辑
- 根据机床的要求设定点动操作键的布局  
MD14510[26] – X + 键  
MD14510[27] – X - 键  
MD14510[30] – Z + 键  
MD14510[31] – Z -键  
MD14510[28] – Y + 键（在MD14510[16]=1时）  
MD14510[29] – Y - 键（在MD14510[16]=1时）
- 如果您的系统配置了611伺服驱动器，而且还没有调试，驱动器的就绪信号就不会生效，导致实例程序加入急停状态且不能退出。在调试开始时可以将I1.7接高电平，或将PLC机床参数MD14512[16]的第一位设定为1，这样就可以退出急停。在驱动器调试完毕后，需将该参数位置0。
- 提供 MD14512[11] 定义使用的功能：  
Bit 7=1 车床刀架有效  
Bit 6=1 铣床主轴换挡生效  
Bit 3=1 主轴控制生效  
Bit 2=1 卡紧放松控制  
Bit 1=1 自动润滑生效  
Bit 0=1 冷却控制生效  
然后802S/C baseline上电使参数生效。这时系统会提示输入所需的PLC机床参数。
- 通过MD14512[16]/[17]/[18]对系统进行技术设定。



#### 重要事项

您还必须设定相应的NC数据，才能使系统加入正常工作状态。  
请仔细阅读802S/C baseline的安装调试手册



## 10. SINUMERIK 802D PLC 子程序库说明

### 10.1. 概述

**说明** SINUMERIK 802D 是用于车床和铣床的全数字式数控系统。  
为了将数控系统与机床联结，必须利用PLC编程工具 *Programming Tool PLC 802 V3.0*，来设计机床的电气逻辑。该工具软件使用 S7-200 STEP7-Micro/WIN32 V3.0 编程语言。  
PLC子程序库的目的是简化机床制造商PLC程序的设计任务，甚至可以直接使用所提供的实例程序。

**读者** 本章为具有CNC和PLC基础知识的机床制造厂工程师而设计。

**相关资料：** 详细信息请参阅系列资料：

- SINUMERIK 802D Start-up and installation Guide
- SINUMERIK 802D Function description
- SIMODRIVE 611U planning guide

#### 10.1.1. 子程序库的内容

子程序库包括了四个项目文件。所有提供的子程序均包括于任意一个项目文件。

**项目文件：**

SUBR_LIBRARY_802D.PTP	包含了所有提供的子程序和一个空的主程序（OB1）
MCP_SIMULATION_802D.PTP	802D机床控制面板MCP仿真
SAMPLE_TURN_802D.PTP	车床实例程序
SAMPLE_MILL_802D.PTP	铣床实例程序

随802D软件版本V2.1工具箱提供的子程序库的版本为V1.5。

本手册描述的是V2.0版子程序库。子程序库项目文件可向西门子销售代表索取，也可由西门子公司的网站直接下载：<http://www.siemens.com/sinumerik802d>



---

#### 重要事项

SINUMERIK802D PLC子程序库V2的功能是在子程序库版本V1.5的基础上设计。请仔细阅读本手册，注意一些子程序库的改动。

---



---

#### 重要事项（已经使用过V1.5版PLC子程序库的机床制造商应注意）

1. 基于PLC子程序库版本V1.5的项目文件仍然可在SINUMERIK 802D系统软件版本V2下直接使用。
  2. 欲使用新版的PLC子程序库来升级原有的项目文件，必须重新组织项目文件的主程序MAIN（OB1）。制造商自行设计的子程序可保持不变；
  3. 所有子程序相关的用户报警应联结到主程序中。
-

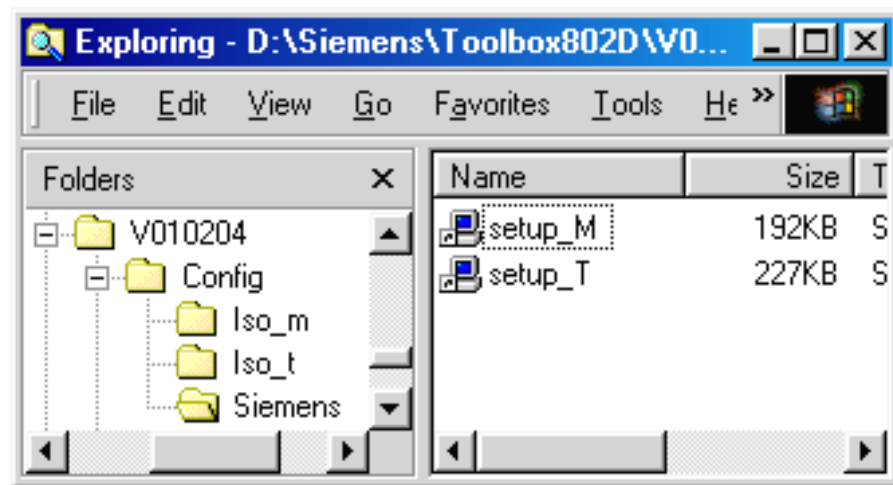
## 10.1.2. 重要须知

下面的内容对于理解和使用所提供的子程序库非常重要。

### 10.1.2.1. SINUMERIK802D系统初始化

如果欲使用PLC子程序库，使用标准的初始化文件对SINUMERIK802D进行初始化是至关重要的。启动由工具箱提供的通讯软件WINPCIN，将车床或铣床的初始化文件通过串行接口以二进制格式传送到802D系统中。

初始化文件的路径：



下载初始化文件SETUP\_T.CNF后，系统变为车床配置，具有两个进给轴和一个主轴：

轴号	轴名	轴信号接口
1	X1	V3800 xxxx
2	Z1	V3801 xxxx
3	SP	V3802 xxxx

下载初始化文件SETUP\_M.CNF后，系统变为铣床配置，具有四个进给轴和一个主轴：

轴号	轴名	轴信号接口
1	X1	V3800 xxxx
2	Y1	V3801 xxxx
3	Z1	V3802 xxxx
4	SP	V3803 xxxx
5	A1	V3804 xxxx

## 10.1.2.2. 系统资源的划分

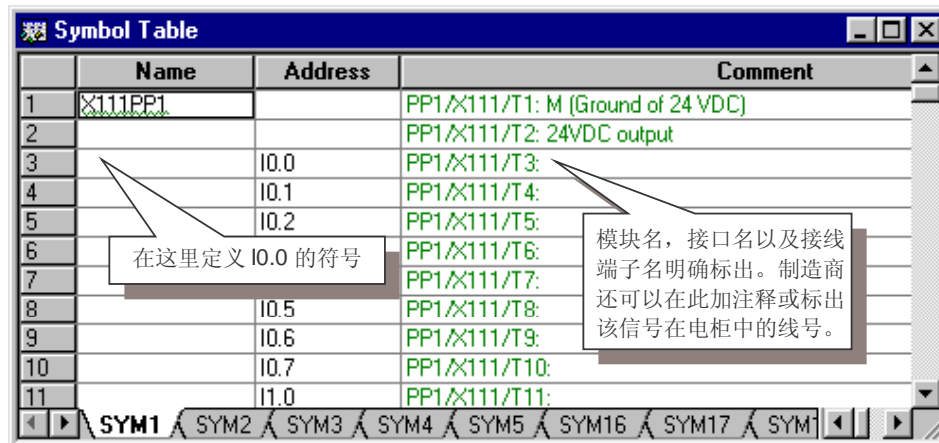
资源（量）	用于制造商	用于 PLC 子程序库
I/O（max.144 / 96）	所有输入输出	无
定时器（32）	24:T0~T23	8:T24 ~T31
计数器（32）	24:C0~C23	8:C24~C31
存储器（256 Bytes）	64: M0.0~M127.7	64: M128.0~M255.7
保持存储器 （128Bytes）	32: V14000000.0~V14000063.7	32: V14000064.0~V14000127.7
报警（64）	64: V16000000.0~V16000007.7	0:
机床数据MD14510 （32）	16: MD14510[0]~ MD14510[15]	16: MD14510[16]~MD14510[31]
机床数据MD14512 （32）	16: MD14512[0]~ MD14512[15]	16: MD14512[16]~MD14512[31]
机床数据MD14514 （8）	08: MD14512[0]~ MD14512[7]	0:
子程序（64）	32: SBR0 ~ SBR31	32: SBR32 ~ SBR63
符号表（32）	15: USR1 ~ USR15	17: USR16 ~ USR31

10.1.2.3. 符号表的结构

PLC程序库采了符号寻址设计，使得PLC应用程序易于理解。在子程序库中使用的所有地址均采用符号编程。所有接口信号均命名以符号，并安排在不同的符号表中。

符号表	表名	符号表内容
1	PP_1	PP 模块 1 I/O 由制造商进行定义
2	PP_2	PP 模块 2 I/O 由制造商进行定义
3 ~ 15		为制造商预留
16	IS_MCP	送至或来自机床控制面板MCP的信号
17	IS_HMI	送至或来自人机接口HMI的信号
18	IS_AUX	来自NCK的辅助功能
19	IS_NCK	送至或来自NCK的信号
20	IS_CHA	送至或来自通道的信号
21	IS_AX1	送至或来自坐标轴 1 的信号
22	IS_AX2	送至或来自坐标轴 2 的信号
23	IS_AX3	送至或来自坐标轴 3 的信号
24	IS_AX4	送至或来自坐标轴 4 的信号
25	IS_AX5	送至或来自坐标轴 5 的信号
26	MD_PLC	PLC 机床数据
27	ALARM	用户报警
28	NV_MEM	可保持存储器 (None Volatile Memory)
29	SPC_MEM	特殊状态字
30	SBR_MEM	实例程序和子程序库使用的全局存储器定义
31 ~ 32		实例程序和子程序库预留

符号表一和符号表二为制造商定义第一个和第二个外设模块（PP72/48）而准备。制造商可以在此定义其机床的输入输出表。



#### 10.1.2.4. 子程序库中符号命名的约定

子程序库中所使用的符号均遵循如下约定：

1. 引导字符表示接口信号的目标方向：
  - P\_ - 表示到PLC的接口信号
  - H\_ - 表示到HMI的接口信号
  - N\_ - 表示到NCK的接口信号
  - M\_ - 表示到MCP的接口信号
2. 随后的字符表示接口区：
  - C\_ - 通道接口信号区
  - 1\_ - 轴接口信号区
  - M\_ - 机床面板MCP接口信号其他缩写符号有：
  - HWL - 硬限位（取自**Hardware Limit**）
  - HW - 手轮（取自**Handwheel**）
  - RT - 快速移动（取自**Rapid Traverse**）
  - TK - 点动键（取自**Traverse key**）
  - ACT - 生效（取自**active**）
  - SEL - 已选择（取自**Selected**）
3. 符号由最多11个大写字母和数字构成（引导字符必须是字母）。除了下划线外不允许任何特殊字符，如 =, +, -, [, ] 等。

## 10.1.2.5. PLC机床数据

MD14510	USER_DATA_INT
机床数据	PLC机床数据 – 16位整型数
14510[16]	定义：机床类型 单位：- 范围：0 – 无定义；1 – 车床；2 – 铣床；>2 无定义
14510[20]	定义：刀架刀位数 单位：- 范围：4, 6, 8
14510[21]	定义：换刀监控时间（如果在监控时间内没有找到目标刀位，换刀结束） 单位：0.01 秒 范围：30 ~ 200（3 ~ 20 秒）
14510[22]	定义：刀架锁紧时间 单位：0.01 秒 范围：5 ~ 30（0.5 ~ 3 秒）
14510[23]	定义：主轴制动时间（接触器控制的主轴） 单位：0.01 秒 范围：5 ~ 200（0.5 ~ 20 秒）
14510[24]	定义：导轨润滑间隔 单位：1 分钟 范围：5 ~ 300 分钟
14510[25]	定义：导轨润滑时间 单位：0.01 秒 范围：10 ~ 200（1 ~ 20 秒）
14510[28]	定义：X 轴 + 点动键的位置，- 点动键位于对称位置。 单位：- 范围：16 ~ 24 之间，但不等于 20
14510[29]	定义：Y 轴 + 点动键的位置，- 点动键位于对称位置。 单位：- 范围：16 ~ 24 之间，但不等于 20
14510[30]	定义：Z 轴 + 点动键的位置，- 点动键位于对称位置。 单位：- 范围：16 ~ 24 之间，但不等于 20
14510[31]	定义：4 <sup>th</sup> 轴 + 点动键的位置，- 点动键位于对称位置。 单位：- 范围：16 ~ 24 之间，但不等于 20



MD14512	USER_DATA_HEX							
机床数据	PLC机床数据 – 2位十六进制数（8位二进制数）							
Index	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
14512[16]				手动键 保持方式	带有 倍率开关	外部 停止信号	使能 自动取消	调试 过程中
14512[17]	带制动装置的坐标轴			返回参考电时倍率开关对下列轴无效				
	Z轴	Y轴	X轴	4 <sup>th</sup> 轴	Z轴	Y轴	X轴	
14512[18]	定义硬限位控制逻辑			技术设定				
	超程链 生效	Z轴单 限位开关	Y轴单 限位开关	X轴单 限位开关	K1作为 驱动使能	上电自动 润滑一次	优化开关 生效	

PLC机床数据的详细解释如下：

(1) MD14512[16]

- Bit 0=0 – PLC 正常运行（出厂设定）
- Bit 0=1 – 调试方式。  
PLC不检测来自电源馈入模块72号端子的就绪信号；
- Bit 1=0 – 主轴使能取消的条件：  
无主轴命令且主轴已停止时按主轴停止键（出厂设定）；
- Bit 1=1 – 主轴使能取消的条件：  
无主轴命令且主轴已停止时，主轴使能自动取消；
- Bit 2=0 – （用于急停处理）主轴停止信号取自NC内部接口  
V390x0001.4（出厂设定）；
- Bit 2=1 – （用于急停处理）主轴停止信号取自PLC输入点；
- Bit 3=0 – 机床控制面板MCP上无主轴倍率开关（出厂设定）；
- Bit 3=1 – 机床控制面板MCP上有主轴倍率开关；
- Bit 4=0 – 主轴手动按键触发方式（按键松开主轴停止）  
（出厂设定）；
- Bit 4=1 – 主轴手动按键保持方式（按主轴停止键主轴停止）；

(2) MD14512[17]

- Bit 2/1/0=0 – 返回参考点时进给倍率有效（出厂设定）；
- Bit 2/1/0=1 – 返回参考点时进给倍率无效；
- Bit 6/5/4=0 – Z/Y/X进给电机无抱闸（出厂设定）；
- Bit 6/5/4=1 – Z/Y/X进给电机有抱闸  
（注意：只有其中一个电机允许有抱闸）

(3) MD14512[18]

- Bit 1=0 – 优化开关（PLC子程序40的输入#OPTM）无效  
（出厂设定）；
- Bit 1=1 – 优化开关（PLC子程序40的输入#OPTM）有效；  
即#OPTM=1电机抱扎释放
- Bit 2=0 – 上电后无自动润滑一次（出厂设定）；
- Bit 2=1 – 上电后自动启动润滑一次

- Bit 3=0 –（出厂设定）；
- Bit 3=1 – 802D机床控制面板MCP上的K1键用于驱动器使能；
- Bit 6/5/4=0 – Z/Y/X轴具有正负两个限位开关（条件：Bit 7=0时有效）
- Bit 6/5/4=1 – Z/Y/X轴具只有一个限位开关（条件：Bit 7=0时有效）
- Bit 7=0 – 硬限位采用PLC方案（出厂设定）（Bit 6/5/4有效）
- Bit 7=1 – 硬限位采用硬件逻辑（急停链）（Bit 6/5/4无效）



---

#### 重要事项

以上机床参数应以十六进制各式输入。注意先将位参数转换为十六进制格式后，再输入。

---

#### 10.1.2.6. 子程序库项目文件中使用的特殊地址

用于子程序的位输入

- SM0.0 – 常“1”，符号为ONE
- M251.0 – 常“0”，符号为ZERO

用于子程序的无效输出

- M255.7 – 用于无效的“位”输出，符号为NULL\_b
- MB255 – 用于无效的“字节”输出，符号为NULL\_B
- MW254 – 用于无效的“字”输出，符号为NULL\_W
- MD252 – 用于无效的“长字”输出，符号为NULL\_D

## 10.1.3. 子程序库中提供的子程序表

子程序#	子程序名	子程序说明
0 ~ 30	-	为制造商预留
31	USR_INI	为制造商初始化预留（该子程序由子程序32自动调用）
32	PLC_INI	PLC初始化
33	EMG_STOP	急停处理（包括驱动器上电和下电时序的控制）
34	MCP_802D	传送802D机床控制面板对应的 I/O 状态到接口V1000xxxx
35	SPD_OVR	利用三个按键选择主轴倍率（格雷码）并送到接口VB1000 0008
36	MINI_HHU	西门子小型手持单元（6FX2006-1BG00）控制
37	MCP_SIMU	机床控制面板MCP仿真
38	MCP_NCK	机床控制面板MCP的信号、操作面板HMI信号送至NCK接口
39	HANDWHL	由操作面板HMI在机床坐标系或工件坐标系选择手轮
40	AXIS_CTL	进给轴和主轴使能控制（包括硬限位和电机抱闸释放等）
41	SPD_GEAR	主轴两级变速控制（自动摆动）
42		子程序预留
43	MEAS_JOG	手动刀具数据测量（只用于车床）
44	COOLING	冷却控制（手动键及M代码：M07/M08/M09）
45	LUBRICATE	导轨润滑控制（每时间间隔润滑一个时间单位）
46	TURRET1	车床刀架控制（刀架类型：霍尔元件传感器、4/6/8工位）
47	TURRET2	车床刀架控制（刀架类型：4位编码器检测位置）
48	TOOL_DIR	判断就近换刀方向，并计算预停刀位
49	LOCK_UNL	卡紧放松控制（车床的卡盘或铣床的刀具）
50	MGZ_SRCH	刀库刀表初始化（用于随机换刀，刀库最多40把刀具）
51	MGZ_RNEW	在刀表中搜索编程刀具所在的刀套号
52	MGZ_INI	刷新刀表
53 ~ 62		子程序预留
63	TOGGLE	6个单键保持开关K1~K6；两个延时开关K7，K8

**注意：**子程序32至63之间所有空的子程序均为子程序库预留；  
表中用黑体字标出的子程序在所有应用场合均需要；

## 10.2. 子程序库说明

### 10.2.1. 子程序 32 – PLC\_INI (PLC初始化)

**子程序32的目的** 该子程序在第一个PLC周期 (SM0.1) 被调用。用于设置一些基本接口信号。  
在该子程序中设定了下列接口信号：

- V32000006.7 - NCK通道接口的进给倍率生效
- V380x0001.5 - 坐标轴接口的测量系统1有效
- V380x0001.7 - 坐标轴接口的进给倍率生效  
(主轴倍率是否生效由机床参数MD14512[16] Bit3确定)

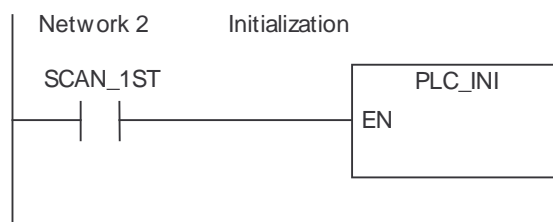
在可保持数据区定义了24个字节用于存储旋转格雷码开关的格雷码。可用于轴选、方式选择等。请参见符号表SYM28 (NV\_MEM)，地址：VB14000101 ~ VB14000124。

开关位置	存储位置	格雷码	方式选择	轴选择	进给倍率	主轴倍率
1	VB14000101	00001	REF	X轴	0%	50%
2	VB14000102	00011	MDA	X轴	1%	55%
3	VB14000103	00010	MDA	Y轴	2%	60%
4	VB14000104	00110	JOG	Y轴	4%	65%
5	VB14000105	00111	INC1	Z轴	6%	70%
6	VB14000106	00101	INC10	Z轴	8%	75%
7	VB14000107	00100	INC100	第四轴	10%	80%
8	VB14000108	01100	INC1000	第四轴	20%	85%
9	VB14000108	01101	INC10000	第五轴	30%	90%
10	VB14000110	01111	REPOS	第五轴	40%	95%
11	VB14000111	01110	AUTO		50%	100%
12	VB14000112	01010	AUTO		60%	105%
13	VB14000113	01011	REF		70%	110%
14	VB14000114	01001			75%	115%
15	VB14000115	01000			80%	120%
16	VB14000116	11000			85%	
17	VB14000117	11001			90%	
18	VB14000118	11011			95%	
19	VB14000119	11010			100%	
20	VB14000120	11110			105%	
21	VB14000121	11111			110%	
22	VB14000122	11101			115%	
23	VB14000123	11100			120%	

在该子程序结束之前，自动调用子程序31：USR\_INI。用户初始化的内容可编写在子程序31中。

局部变量定义	无
占用的全局变量	VB14000101 ~ VB14000124: 存储格雷码值 (参见符号表28 - NV_MEM)
相关PLC机床参数	MD14510[16] - 机床类型 (1: 车床; 2: 铣床) MD14512[16] Bit3 = 1 - 操作面板上配备主轴倍率开关

## 子程序调用实例



## 10.2.2. 子程序 33 – EMG\_STOP (急停处理)

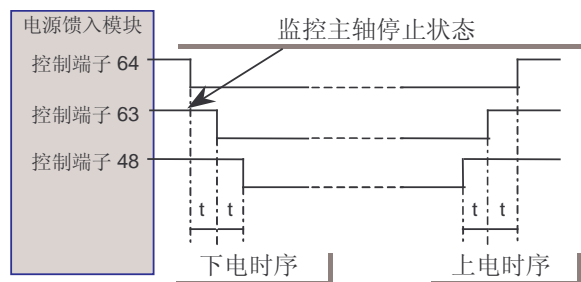


## 注意

请检查该子程序是否满足相关的安全要求。

## 子程序33的目的

该子程序根据“功能说明”中定义的时序对急停的过程，以及对驱动器电源馈入模块的上电及下电时序进行控制。有关611U和电源馈入模块的详细说明请参见611U调整手册。



该子程序的前提条件是：802D必须使用车床或铣床的标准初始化。当去掉驱动器使能时，如电源模块的端子64与端子9断开，所有进给轴和主轴进入制动状态直至完全停止。该子程序根据802D接口信号V390X0001.4 -  $n < n_{\min}$ 或外部主轴停止信号 (MD14512[16]Bit2= “1”) (如变频器) 来确定主轴停止状态。驱动器的使能和禁止信号来自MCP到PLC的接口信号V10000002.7 - 使能, V10000002.6 - 禁止。

该子程序可能激活以下两个报警：

报警700016 – 驱动器未就绪

报警700017 – 电源馈入模块I<sup>2</sup>/T报警



**重要事项**



主轴停止信号，或来自NCK接口、或来自外部，是该子程序处理上下电时序时必不可少的条件。假如再刚开始调试时，暂时不能提供主轴停止信号，可设置PLC机床参数MD14512[16] Bit0 = “1” –既调试方式，这时PLC不检查主轴停止信号。当能够提供主轴停止信号时，必须将该机床参数位设为：MD14512[16] Bit0 = “0”

**局部变量定义**

**输入：**

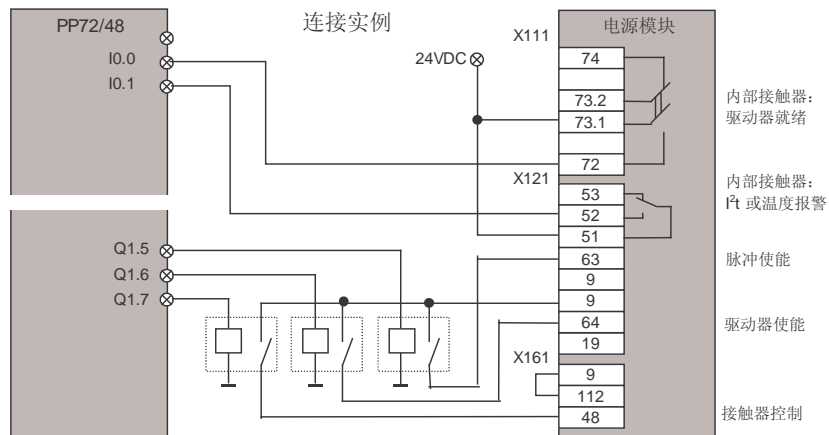
DELAY	WORD	上下电时序延时（单位：2*PLC扫描周期 <sup>4)</sup> ）
E_KEY	BOOL	急停开关（NC） <sup>3)</sup>
T_72	BOOL	驱动器就绪反馈： 来自电源模块端子T72（T73.1接24VDC）
T_52	BOOL	I <sup>2</sup> t监控信号： 来自电源模块端子T52（T51接24VDC）
HWL_ON	BOOL	任意轴硬限位有效状态（NO） <sup>1)</sup>
SpStop	BOOL	主轴停止信号（NO） <sup>2)</sup>

**注：**

- 1) 可取自子程序40（AXIS\_CTL）的一个输出，使在硬限位出现时可自动进入急停；
- 2) 当PLC机床参数MD14512[16] Bit2 = 1时，外部主轴停止信号生效；
- 3) NO – 常开信号 ； NC –常闭信号 ；
- 4) PLC扫描周期由NC机床参MD100074确定（1=12ms； 2=24ms）

**输出：**

T_48	BOOL	连接电源模块控制端子48：控制接触器
T_63	BOOL	连接电源模块控制端子63：脉冲使能
T_64	BOOL	连接电源模块控制端子64：控制器使能
EN_L	BOOL	驱动器使能状态指示
ERR1	BOOL	错误信息：驱动器未就绪
ERR2	BOOL	错误信息：I <sup>2</sup> t监控报警



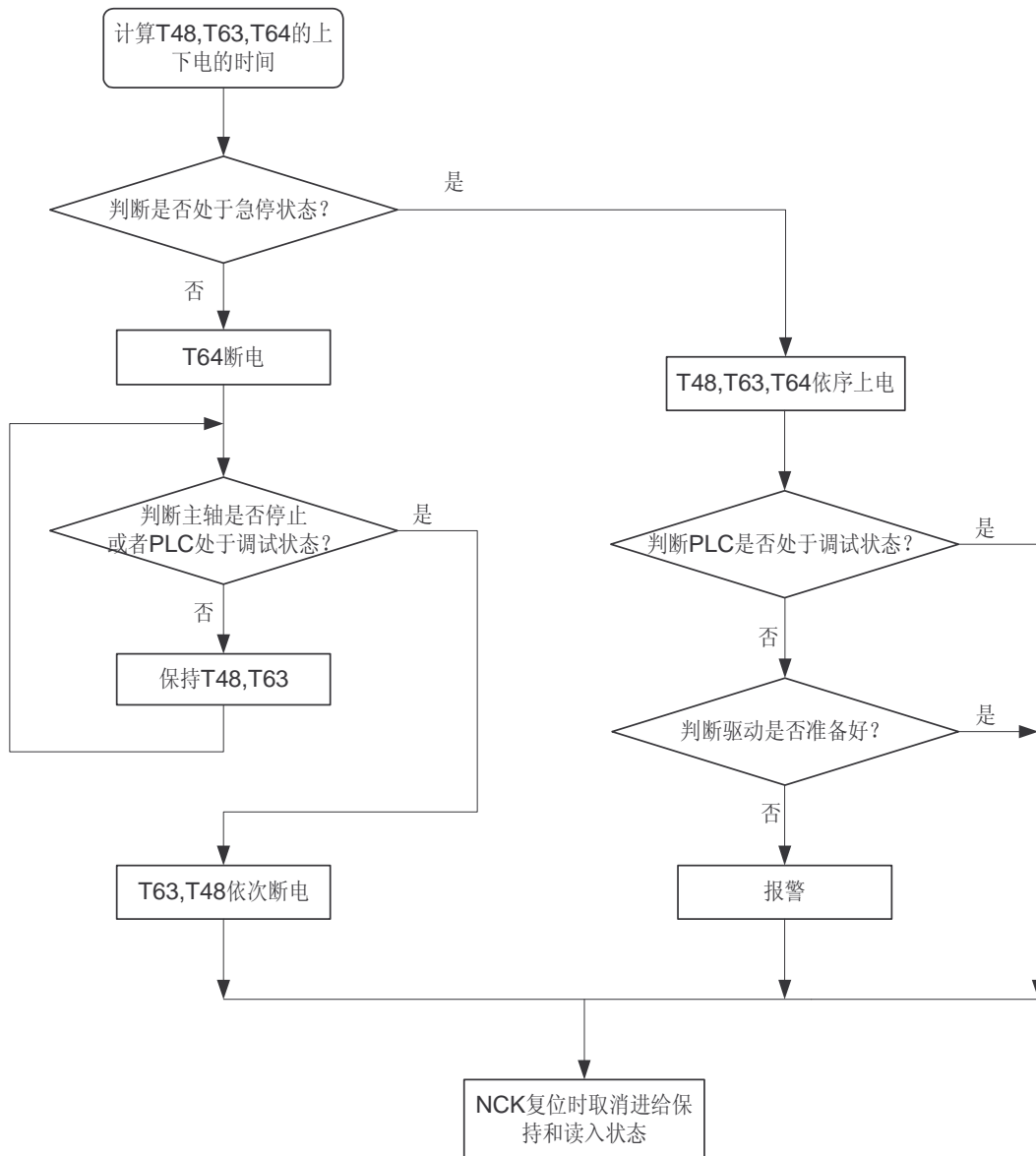
## 占用的全局变量

T_48m	M130.0	记录驱动器电源模块端子48的状态
T_63m	M130.1	记录驱动器电源模块端子63的状态
T_64m	M130.2	记录驱动器电源模块端子64的状态
D_T64m	M130.6	使能延时
SP_STOPm	M130.4	主轴静止状态
PO_END	M130.7	上电过程结束
P_TON	C31	计数器31 - CTU 用于定时

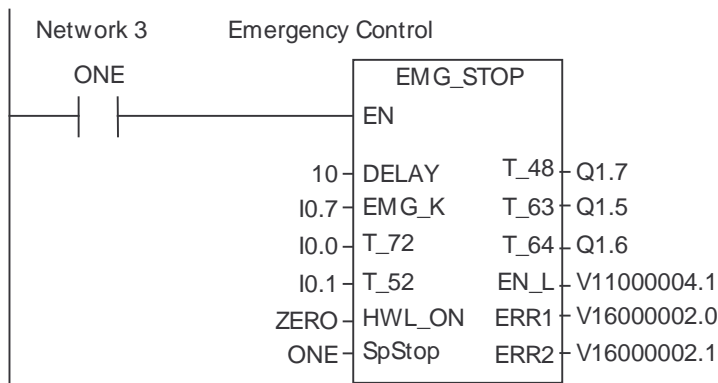
## 相关PLC机床参数

MD14510[16]	WORD	1 – 车床； 2 – 铣床
MD14512[16] Bit0	BOOL	0 – 正常运行； 1 – 调试过程中

## 子程序流程图



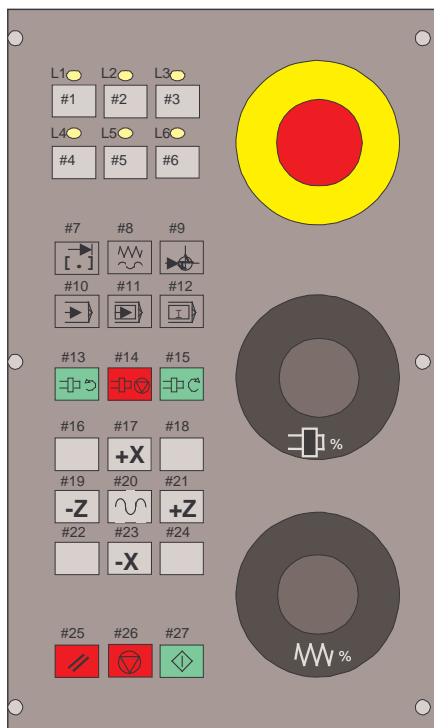
子程序调用实例



10.2.3. 子程序 34 – MCP\_802D (802D机床控制面板MCP信号传递)

子程序34的目的

子程序34将来自802D机床控制面板（订货号：6FC5 603-0AD00-0AA1）的信号，通过PP72/48的输入位传递到接口数据区V1000xxxx以备后续子程序进一步处理。如果制造商使用自制的机床控制面板，可以按该子程序的方法设计自己的面板处理子程序，将机床控制面板的信号传送到相同的数据区。802D机床控制面板的按键布局如下图所示。





802D机床控制面板具有两个50芯的扁平电缆插座用来连接PP72/48模块。每个插座具有24个数字输入和16个数字输出。它们的排列如下：

802D MCP	对应的按键
X1201	输入字节0: 对应按键 #1 ~ #8
	输入字节1: 对应按键 #9 ~ #16
	输入字节2: 对应按键 #17 ~ #24
	输出字节0: 6个对应于用户定义键的发光二极管
X1202	输入字节3: 对应按键 #25 ~ #27
	输入字节4: 对应进给倍率开关 (5 位格雷码)
	输入字节5: 对应主轴倍率开关 (5 位格雷码)
	输出字节1: 保留

两条扁平电缆可以连接PP72/48模块 (PROFIBUS总线地址9) 上任意插座:

PP72/48	输入字节	输出字节
X111	IB0, IB1, IB2	QB0, QB1
X222	IB3, IB4, IB5	QB2, QB3
X333	IB6, IB7, IB8	QB4, QB5

该子程序可能激活以下两个报警:

报警700024 – 机床控制面板故障

报警700025 – 点动按键没有定义

802D机床面板MCP或任何制造商自制的机床面板上的所有按键或按钮，必须映像到MCP信号接口：

1000	来自MCP的信号 [写/读] 接口信号：MCP 到 PLC							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
10000000	NC-停止				单段方式	请求操作方式 手动方式	MDA方式	自动方式
10000001	NC-启动	主轴正转	主轴手动 主轴停止	主轴反转	钥匙开关 保护级7	机床功能 参考点方式	再定位方式	
10000002	进给 启动	进给 停止	变增量	钥匙开关 保护级4	机床功能 增量 1000	机床功能 增量 100	机床功能 增量 10	机床功能 增量 1
10000003	复位	钥匙开关 保护级6	钥匙开关 保护级5	E	D	C	B	A
10000004	轴点动键			用户自定义键 (CK)				
	第四轴 -	第四轴 +	快速叠加	CK4	CK3	CK2	CK1	CK0
10000005	空	CK5	第三轴 -	第三轴 +	第二轴 -	第二轴 +	第一轴 -	第一轴 +
10000006	用户自由定义键				用户自由定义键			
	空	空	空	空	空	空	空	空
10000007	用户自由定义键							
	空	空	空	空	空	空	空	空
10000008	主轴倍率 (5位格雷码)							
	"0"	"0"	"0"	E	D	C	B	A

1100	送到MCP的信号 [写/读]							
	接口信号: PLC 到MCP							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
11000000	NC- 停止				单段 方式	请求操作方式 手动 方式	MDA 方式	自动 方式
11000001	NC- 启动	主轴手动 主轴 正转      主轴 停止      主轴 反转			空	机床功能 参考点 方式      再定位 方式		
11000002	进给 启动      停止		变增量	空	增量 1000	增量 100	增量 10	增量 1
11000003	空	空	空	空	空	空	空	空
11000004	用户自定义键			用户自定义键上的发光二极管LED				
	第四轴 -	第四轴 +	快速叠 加	CLED5	CLED4	CLED3	CLED2	CLED1
11000005	空	CLED6	第三轴 -	第三轴 +	第二轴 -	第二轴 +	第一轴 -	第一轴 +
11000006	用户自定义键					用户自定义键		
	空	空	空	空		空	空	空
11000007	用户自定义键							
	空	空	空	空	空	空	空	空

## 局部变量定义

## 输入:

PB_0	BYTE	对应于按键 K1 ... K8
PB_1	BYTE	对应于按键K9 ... K15和K20
PB_2	BYTE	对应于按键K16... K19和21...K24
PB_3	BYTE	对应于按键K25... K27
Fov	BYTE	对应于进给倍率
Sov	BYTE	对应于主轴倍率
Drv_En	BOOL	驱动器使能键（持信号: 1- 使能; 0: 禁止; 见机床参数MD14512[18] bit 3）
I_En	BOOL	NC启动条件

## 输出: 按键状态送到接口信号区: V1100 xxxx, 且:

LEDs	BYTE	对应于用户定义键的发光二极管L1到L6
ERR1	BOOL	错误信息: MCP故障（检查MCP硬件）
ERR2	BOOL	错误信息: 点动键没有定义

## 占用的全局变量

MB107 – 用于手持单元点动按键的内部接口  
注意: 详情参见第2.5.章

## 相关PLC机床参数

MD14510[28] – +X点动键的键号

MD14510[29] – +Y点动键的键号

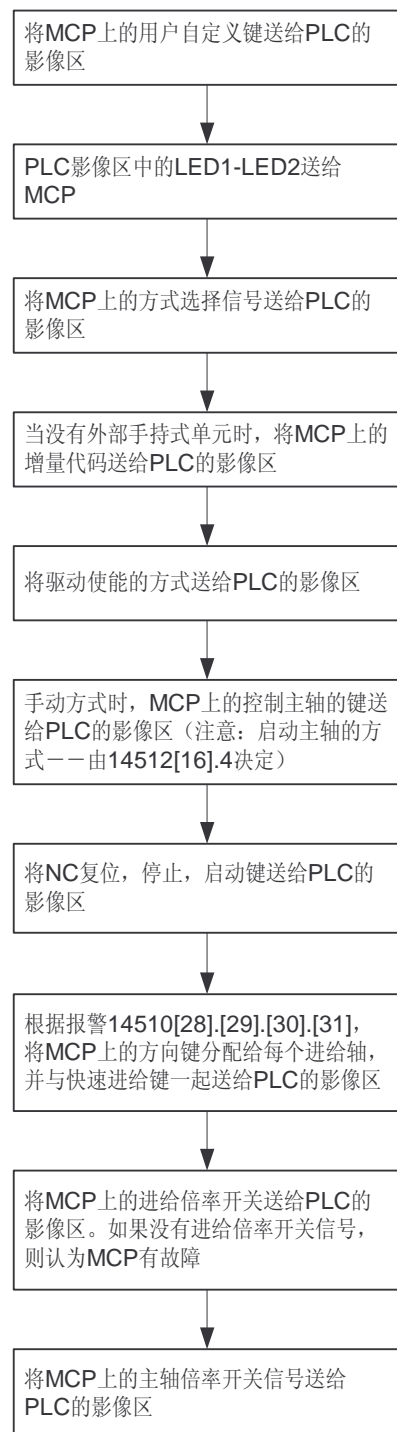
MD14510[30] – +Z点动键的键号

MD14510[31] – +A点动键的键号

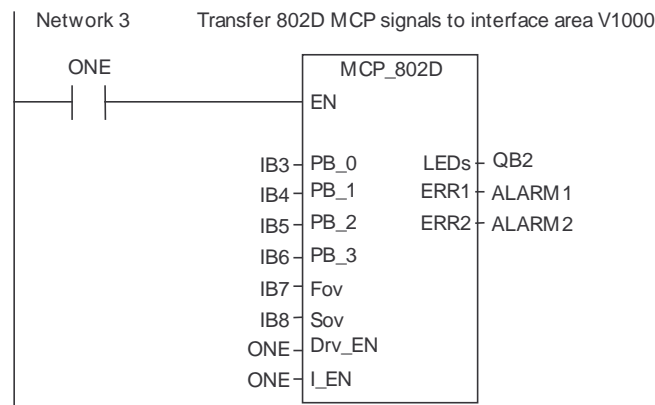
MD14512[16] Bit 4 – 主轴点动操作方式：0 - 触发信号；1 – 保持信号

**注意：** 负向点动键默认位于正向点动键的对称位置；#20按键定义为快速键；如果点动按键没有定义，不能进行点动操作；

## 子程序流程图



## 子程序调用实例



## 10.2.4. 子程序 35 – SPD\_OVR（利用按键选择主轴倍率）

## 子程序35的目的

该子程序提供了一个利用按键替代旋转格雷码倍率开关的方法。

一些制造商为了简化其机床控制面板的设计，利用三个按键对主轴倍率进行增加或减小或直接选择100%。生成的倍率码自动输出到MCP接口VB10000008。

如果Gcode=1，则选择了格雷码倍率，此时STEPi无效。共有15个格雷码对应旋转倍率开关15个位置的格雷码。

如果Gcode=0，则选择了二进制倍率，这时需要设定以下机床参数：

MD12060 OVR\_SPIND\_IS\_GRAY\_CODE = 0      (0: 二进制码; 1: 格雷码)

## 局部变量定义

## 输入:

STEPi	WORD	二进制倍率增加或减小的步长。范围: 1~10
Gcode	BOOL	1 – 格雷码; 0 – 二进制码;
OVinc	BOOL	主轴倍率增加键
OVdec	BYTE	主轴倍率减小键
OV100	BYTE	主轴倍率100%

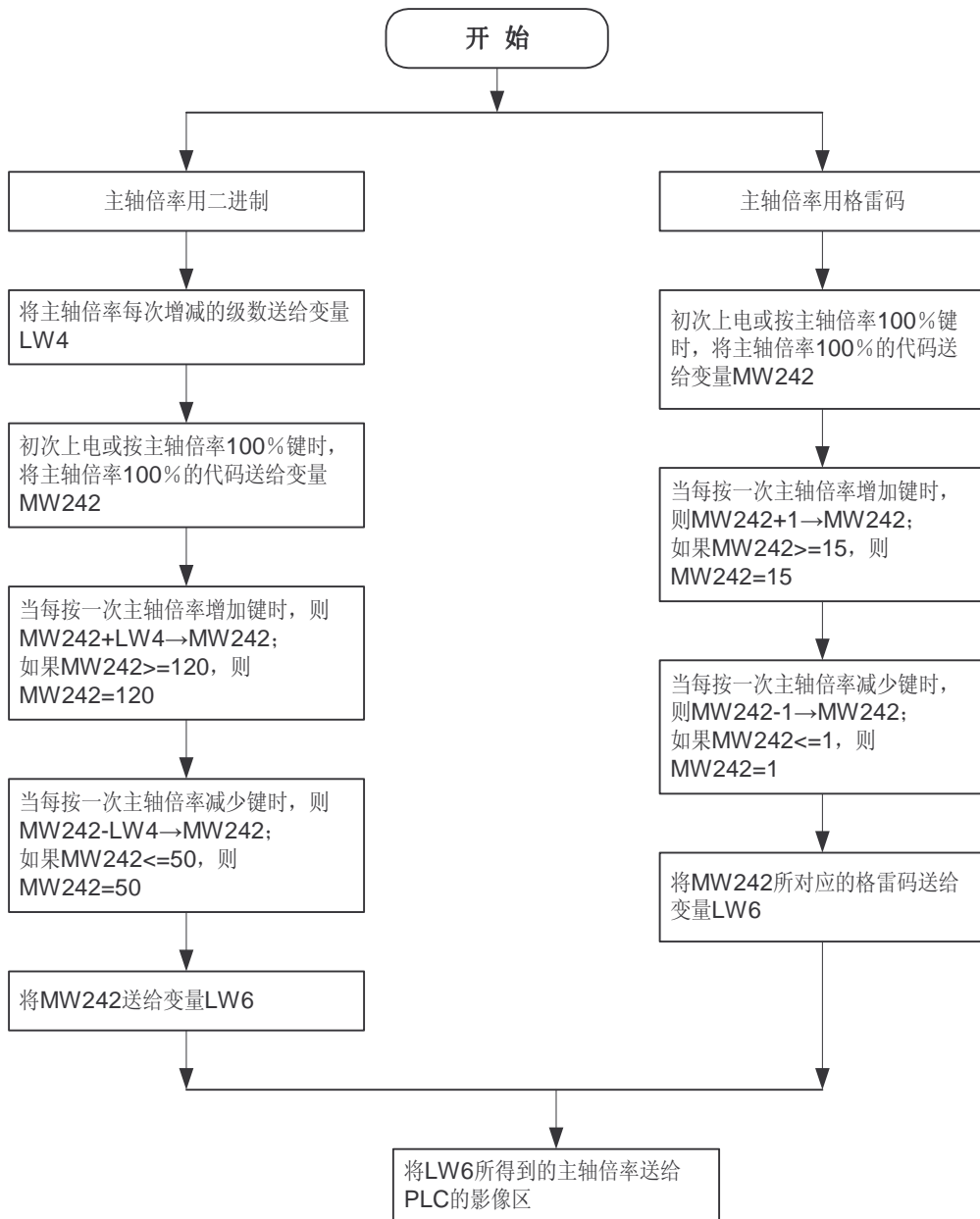
## 占用的全局变量

OV\_CNT      MW242      缓冲倍率码

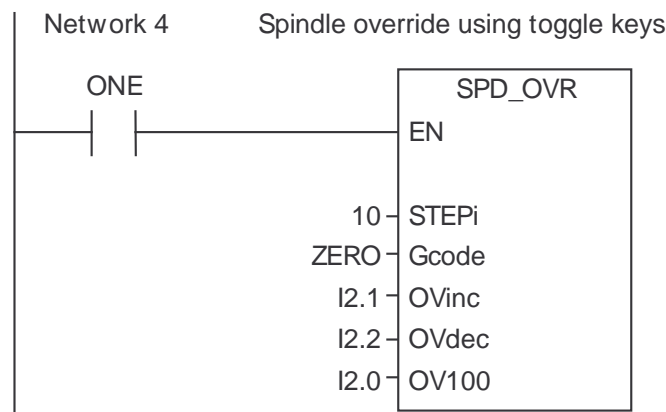
## 相关PLC机床参数

无

子程序流程图



子程序调用实例



## 10.2.5. 子程序36 – MINI\_HHU

**子程序 36 的目的** 该子程序的目的是将西门子小型手持单元（订货号：6FX2006-1BG00）上的点动键和增量通过由输入信号通过一个内部接口送到NCK。对于与其类似的手持单元，制造商可参照或修改该子程序来满足其特定的要求。手轮编码器可连接到80 2D PCU 上任意三个手轮接口之一，同时要将手轮接口的编号输入该子程序。

### 局部变量定义

（只有输入）

PORT	WORD	手轮接口号（1/2/3）	
ENABLE	BOOL	HHU 使能（NO）	轴码： Pin8 Pin9 Pin10
AX_A	BOOL	轴选 Pin8（NO）	X 0 1 0
AX_B	BOOL	轴选 Pin9（NO）	Y 1 1 1
AX_C	BOOL	轴选 Pin10（NO）	Z 0 1 1
KEY_Tp	BOOL	“正”点动键（NO）	
KEY_Tn	BOOL	“负”点动键（NO）	
KEY_R	BOOL	快速键（NO）（也用于清除HHU的增量）	
INC1	BOOL	增量1（NO）	
INC10	BOOL	增量10（NO）	
INC100	BOOL	增量100（NO）	

**占用的全局变量** 与 SBR34（X\_CROSS）中的定义相同

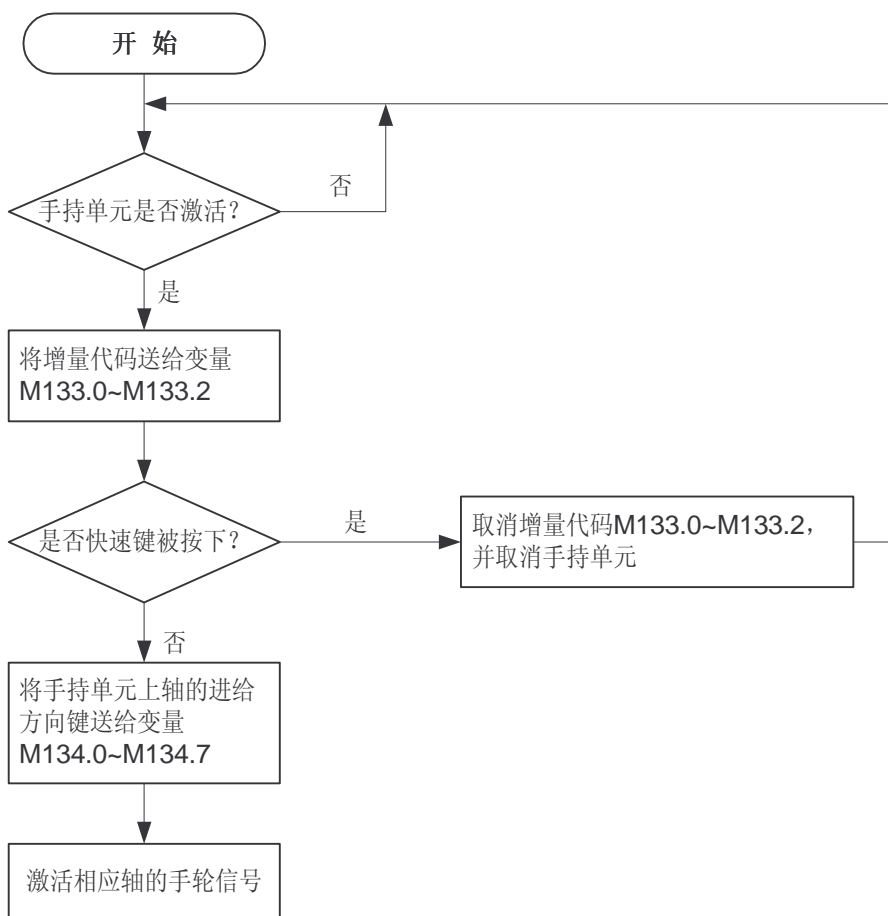
HHU_AX1P	M134.0	HHU 点动键：第一轴“正”
HHU_AX1N	M134.1	HHU点动键：第一轴“负”
HHU_AX2P	M134.2	HHU点动键：第二轴“正”
HHU_AX2N	M134.3	HHU点动键：第二轴“负”
HHU_AX3P	M134.4	HHU点动键：第三轴“正”
HHU_AX3N	M134.5	HHU点动键：第三轴“负”
HHU_AX4P	M134.6	HHU点动键：第四轴“正”
HHU_AX4N	M134.7	HHU点动键：第四轴“负”
HHU_RAPID	M135.0	HHU 快速叠加键
HHU_ACT	M135.1	HHU 使能

如果HHU\_ACT为“1”，手持单元的选择信号来自PLC的输入信号而不是来自HMI（VB19001003 & VB19001004）。

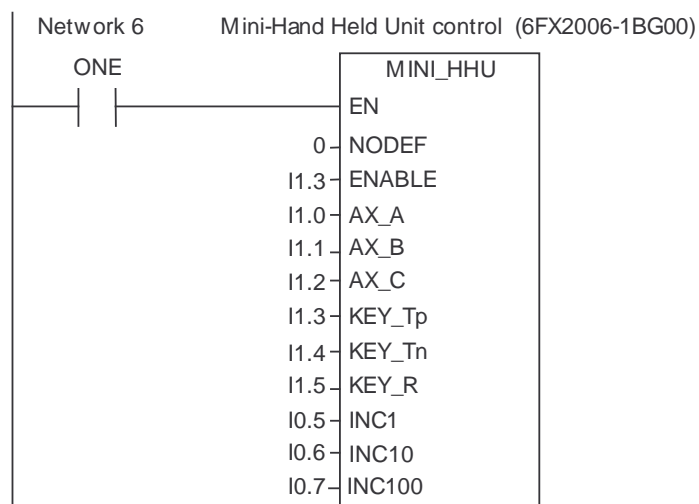
点动信号由该子程序送到以上列出的内部接口。

**相关PLC机床参数** MD14510[16] – 机床类型

子程序流程图



子程序调用实例

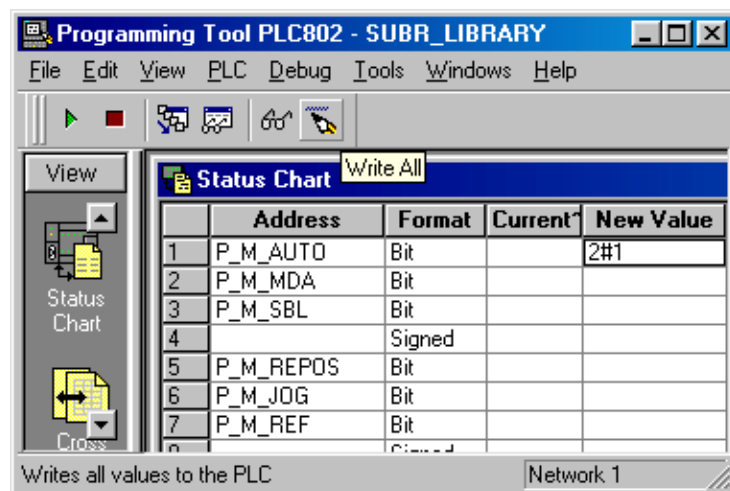




### 10.2.6. 子程序37 – MCP\_SIMU（机床操作面板MCP仿真）

#### 子程序37的目的

该子程序的目的是提供一种替代的方法，即在没有机床控制面板的情况下操作802D。其原理是利用PLC编程软件Micro/WIN中的状态表来仿真MCP的功能，如方式选择、返回参考点，NC启动和停止，等。



#### 局部变量定义

无

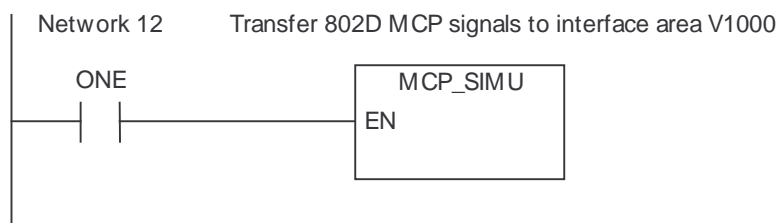
#### 占用的全局变量

SIM_CAM1	M249.0	仿真第一轴参考点碰块
SIM_CAM2	M249.1	仿真第二轴参考点碰块
SIM_CAM3	M249.2	仿真第三轴参考点碰块
SIM_T63	M249.3	仿真驱动器控制端子63
SIM_T64	M249.4	仿真驱动器控制端子64
FOV_P	M250.0	进给倍率增加
FOV100	M250.1	进给倍率100%
FOV_N	M250.2	进给倍率减少
SOV_P	M250.3	主轴倍率增加
SOV100	M250.4	主轴倍率100%
SOV_N	M250.5	主轴倍率减少
SIM_INC	M250.7	增量选择
FOV_POS	C25	用于仿真进给倍率
SOV_POS	C26	用于仿真主轴倍率

#### 相关PLC机床参数

无

#### 子程序调用实例



**10.2.7. 子程序38 – MCP\_NCK (MCP和HMI信号处理)**

**子程序 38 的目的** 该子程序的目的是将来自MCP (V1000xxxx) 和HMI (V1700xxxx, V1800xxxx 和 V1900xxxx) 接口信号送到NCK接口, 以激活如操作方式等。主要功能有:

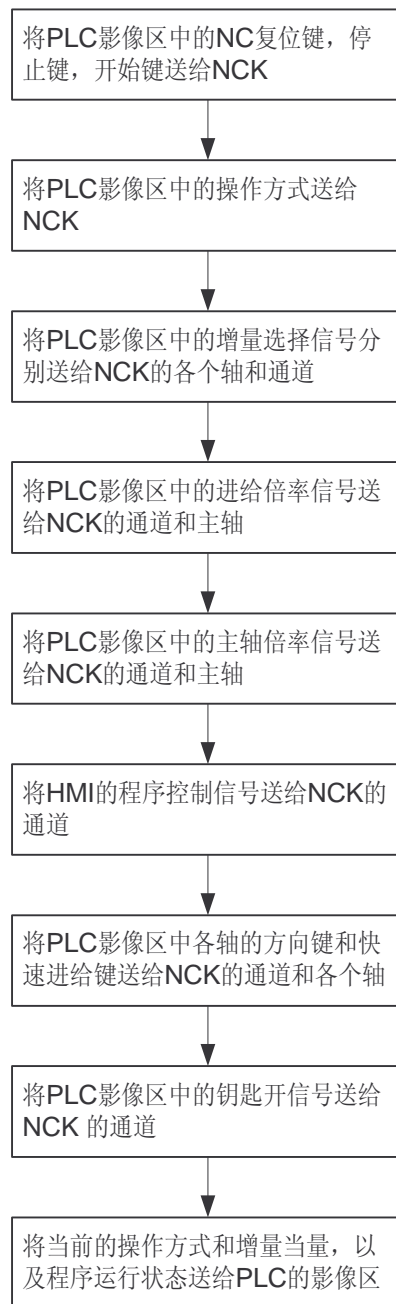
1. 选择操作方式
2. 选择增量
3. HMI信号送NCK接口 (如程序控制、手轮等)
4. 根据PLC机床参数对点动控制

**局部变量定义** 无

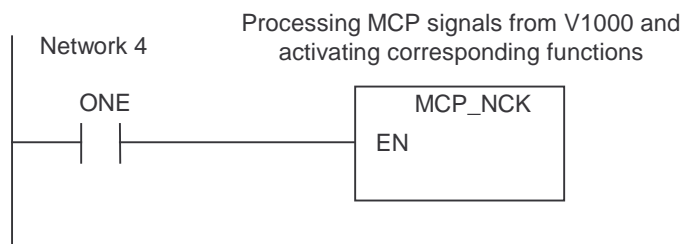
**占用的全局变量** 无

**相关PLC机床参数** MD14510[16]    WORD    1 - 车床; 2 - 铣床

## 子程序流程图



## 子程序调用实例



### 10.2.8. 子程序39 – HMI\_HW（根据HMI接口信号选择手轮）

**子程序39的目的** 该子程序的目的是根据HMI的接口信号V1900 1xxx 在机床坐标系或工件坐标系选择802D三个手轮接口中的任意一个手轮控制任意坐标轴。

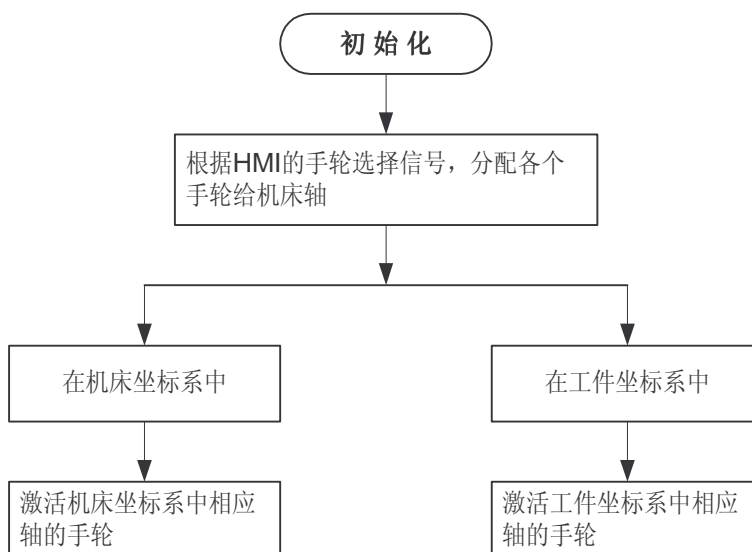
**注意：** 该子程序不能与子程序36 - MINI\_HHU一起使用。

**局部变量定义** 无

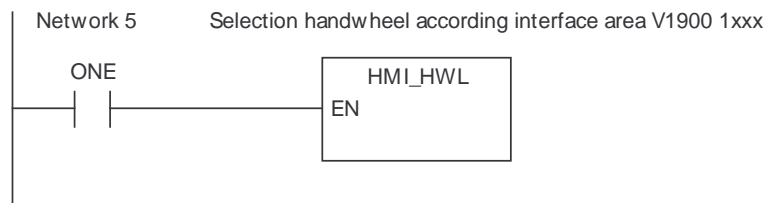
**占用的全局变量** 无

**相关PLC机床参数** 无

**子程序流程图**



**子程序调用实例**



### 10.2.9. 子程序40 – AXES\_CTL (主轴和进给轴控制)

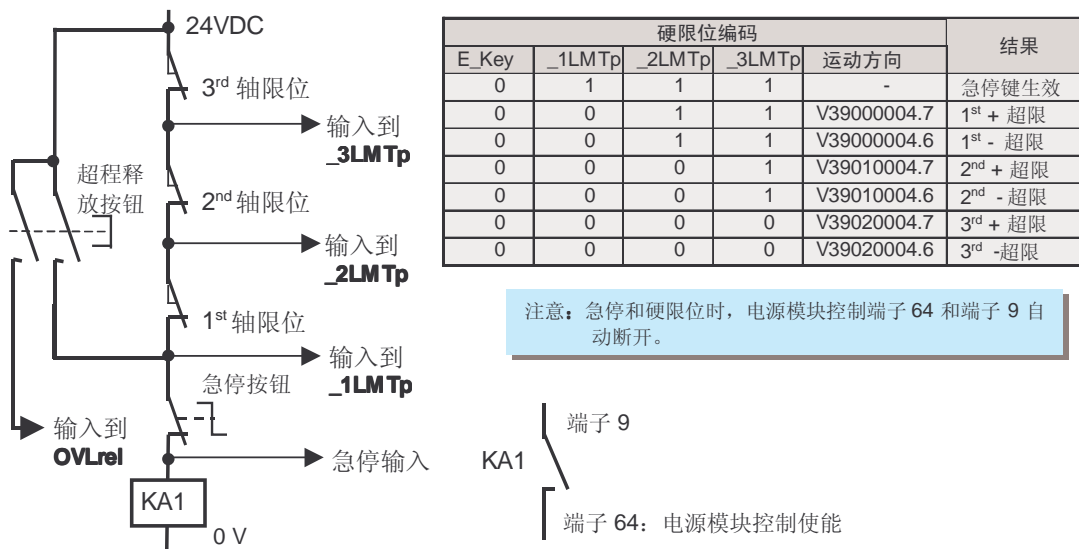
#### 子程序40的目的

该子程序的目的是对驱动器使能（V380x0002.1 & V380x4001.7）、监控硬限位和参考点碰块以及控制电机抱闸的释放等。

该子程序提供两种硬限位的控制方案：

PLC方案（MD14512[18] bit 7=0），且支持每轴双硬限位开关或单硬限位开关，和硬件逻辑方案（MD14512[18] bit 7=1），即急停链（超程链）（注意：若采用超程链，必须严格按照下图接线）。

主轴的使能条件是主轴命令，如M03等，主轴使能的取消条件可由该子程序的选择信号“SPAD”（也可通过机床参数MD14512[16]Bit1=1）选择：“0”是无主轴命令且主轴已经停止时，按MCP上的主轴停键；“1”无命令时，主轴使能自动取消。



#### 局部变量定义

##### 输入：

NODEF	WORD	保留字
T_64	BOOL	电源馈入模块控制端子64的状态
T_63	BOOL	电源馈入模块控制端子63的状态
SPAD	BOOL	选择信号： 1-主轴使能自动取消；0-主轴停止键取消使能
OPTM	BOOL	抱闸释放开关（NO），用于驱动器优化
OVLrel	BOOL	超程释放开关（NO）
_1LMTp	BOOL	第一轴硬限位开关正（NC）*
_1LMTn	BOOL	第一轴硬限位开关负（NC）
_1REF	BOOL	第一轴参考点开关（NO）
_2LMTp	BOOL	第二轴硬限位开关正（NC）*
_2LMTn	BOOL	第二轴硬限位开关负（NC）
_2REF	BOOL	第二轴参考点开关（NO）
_3LMTp	BOOL	第三轴硬限位开关正（NC）*

_3LMTn	BOOL	第三轴硬限位开关负（NC）
_3REF	BOOL	第三轴参考点开关（NO）
_4REF	BOOL	第四轴参考点开关（NO）
_5REF	BOOL	第五轴参考点开关（NO）

\*注意：如果只有一个硬限位开关，或使用超程链位时，使用硬限位正作为输入。

#### 输出：

_BRK	BOOL	抱闸释放输出（“1”有效）
HWLon	BOOL	硬限位输出（“1”有效）
MSG1	BOOL	信息：驱动器优化时抱闸已经释放

#### 占用的全局变量

X_M_DIR	V14000063.0	X轴运动方向：1：+ 方向；0：- 方向；
Y_M_DIR	V14000063.1	Y轴运动方向：1：+ 方向；0：- 方向；
Z_M_DIR	V14000063.2	Z轴运动方向：1：+ 方向；0：- 方向；

#### 相关PLC机床参数

MD14512[18] Bit 7 –	1: 超程链生效 0: PLC硬限位方案（下列参数位生效）：
MD14512[18] Bit 6/5/4 –	1: 每轴只有一个硬限位开关； 0: 每轴有两个硬限位开关；
MD14512[18] Bit 1 –	1: 子程序输入#OPTM生效： #OPTM=1 - 释放抱闸；#OPTM=0 - 抱闸锁紧； 0: 子程序输入#OPTM无效

#### 注意

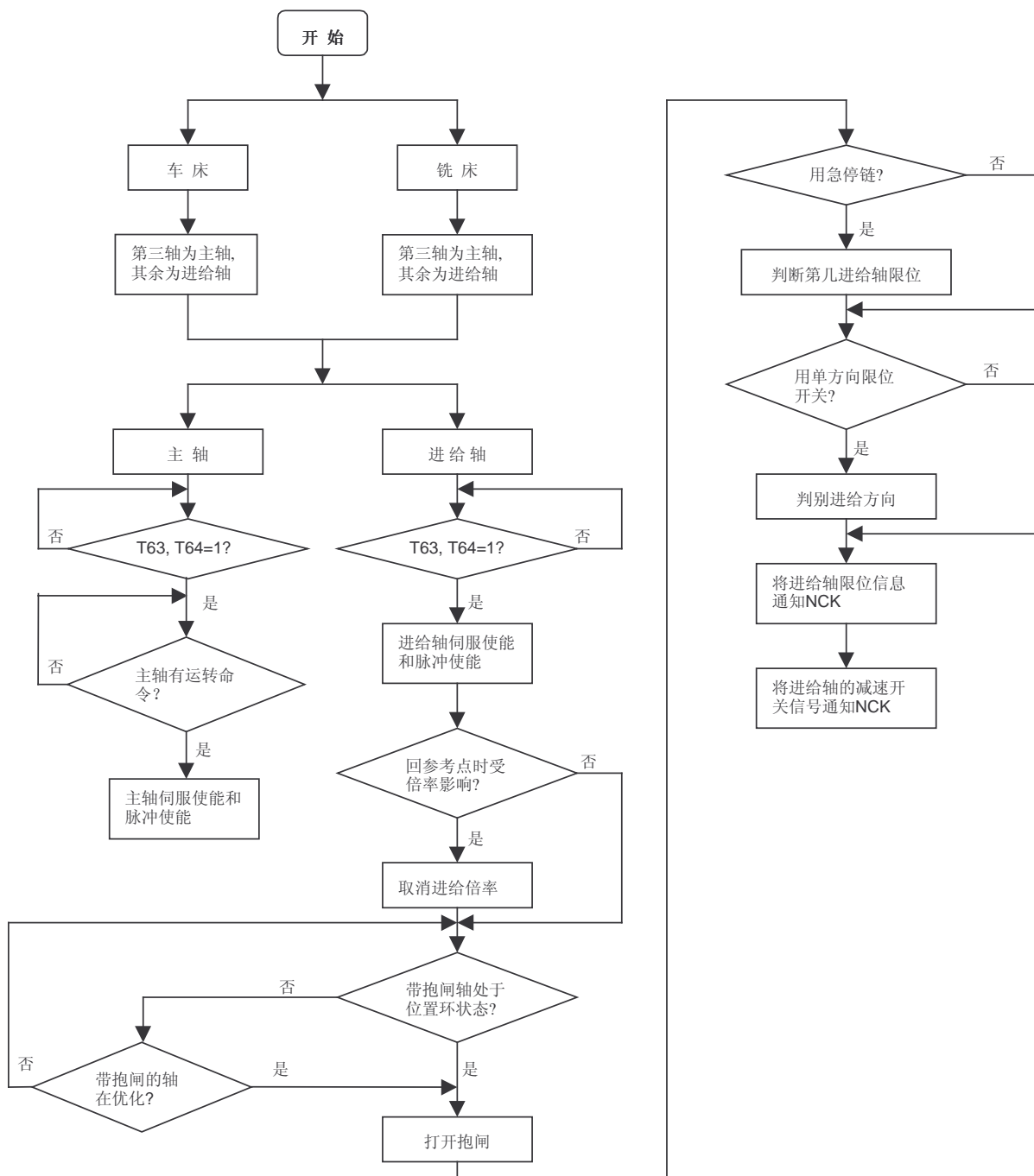
- 1) 优化是指对驱动器的控制参数自动设定。对带抱闸电机的速度环进行优化时电机的抱闸必须释放，对电流环进行优化时，电机抱闸必须锁紧。输入信号#OPTM正是用于此目的。该输入必须在机床参数MD14512[18] Bit1为“1”时才生效。
- 2) 急停（V27000000.1=1）在任何时候均可将抱闸锁紧；
- 3) 下列报警在优化时出现：报警 700026 – 驱动器优化时抱闸已释放



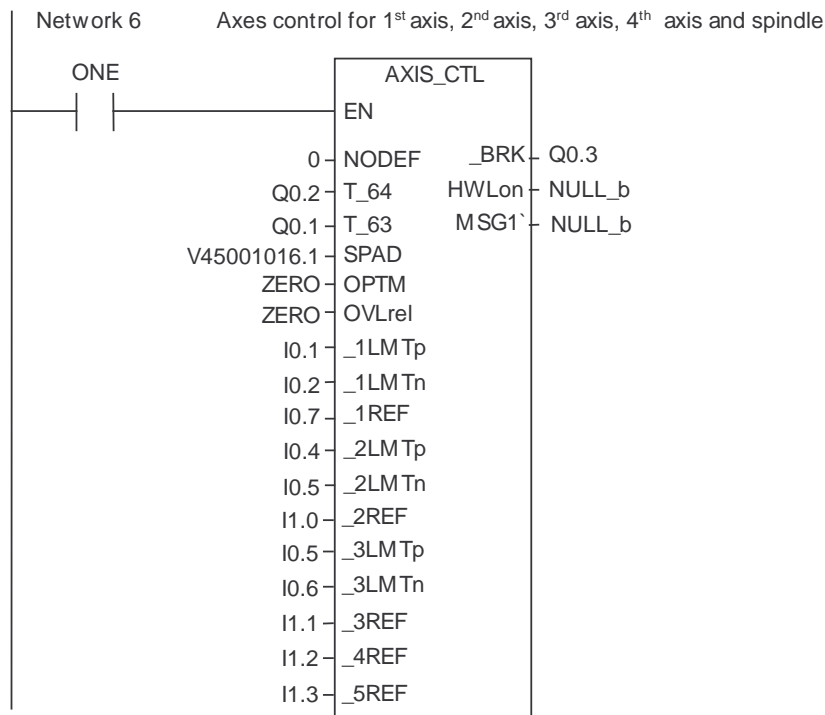
#### 重要事项

在驱动器优化完毕后，必须将机床参数MD14512[18] Bit1设置为“0”。

子程序流程图



子程序调用实例



10.2.10. 子程序41 – GEAR\_CHG (主轴换挡)

子程序41的目的

该子程序为主轴两挡换挡设计。主轴为伺服主轴，如611U控制的1PH7数字主轴，或者模拟主轴，如变频器主轴（±10V控制）。主轴上需配备两个挡位的到位检测开关。换挡动作由不同的主轴速度，或M41/M42激活。换挡过程中主轴自动摆动。

局部变量定义

输入:

D_CHG	WORD	换挡延时（单位：0.01秒）
D_MON	WORD	换挡时间监控（单位：0.01秒）
D_S0	WORD	主轴停止延时（单位：0.01秒）
S_hold	BOOL	主轴零速信号（NO）
S_alarm	BOOL	主轴报警（NO）
LGi	BOOL	低档到位检测信号（NO）
HGi	BOOL	高档到位检测信号（NO）

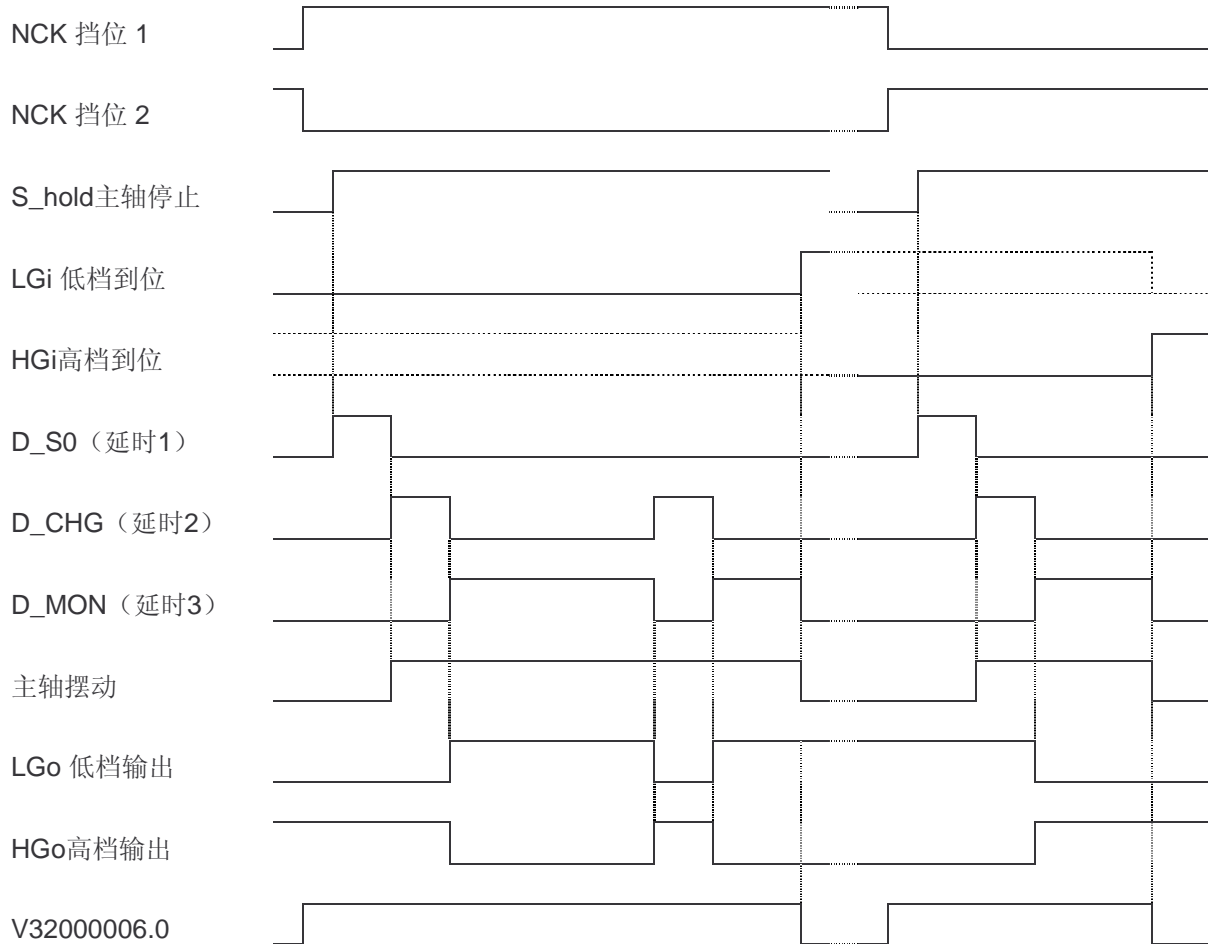
- 注1: D\_S0      – 主轴停止信号启动延时，延时后主轴开始摆动；  
 注2: D\_CHG     – 摆动开始启动延时，延时后换挡动作开始（换挡输出）；  
 注3: D\_MON     – 换挡输出启动延时，延时时间到若仍无换挡到位信号，重复换挡动作。



## 输出:

LGo	BOOL	低挡输出
HGo	BOOL	高档输出
LG_in	BOOL	低挡到位显示
HG_in	BOOL	高档到位显示

## 换挡时序图



## 注意:

NCK挡位1表示主轴的编程速度在第一挡速度范围内, 或编程了M41;

NCK挡位2表示主轴的编程速度在第二挡速度范围内, 或编程了M42;

## 占用的全局变量

MB117	8位状态信息用于换挡
T25/T26/T27	使用的定时器

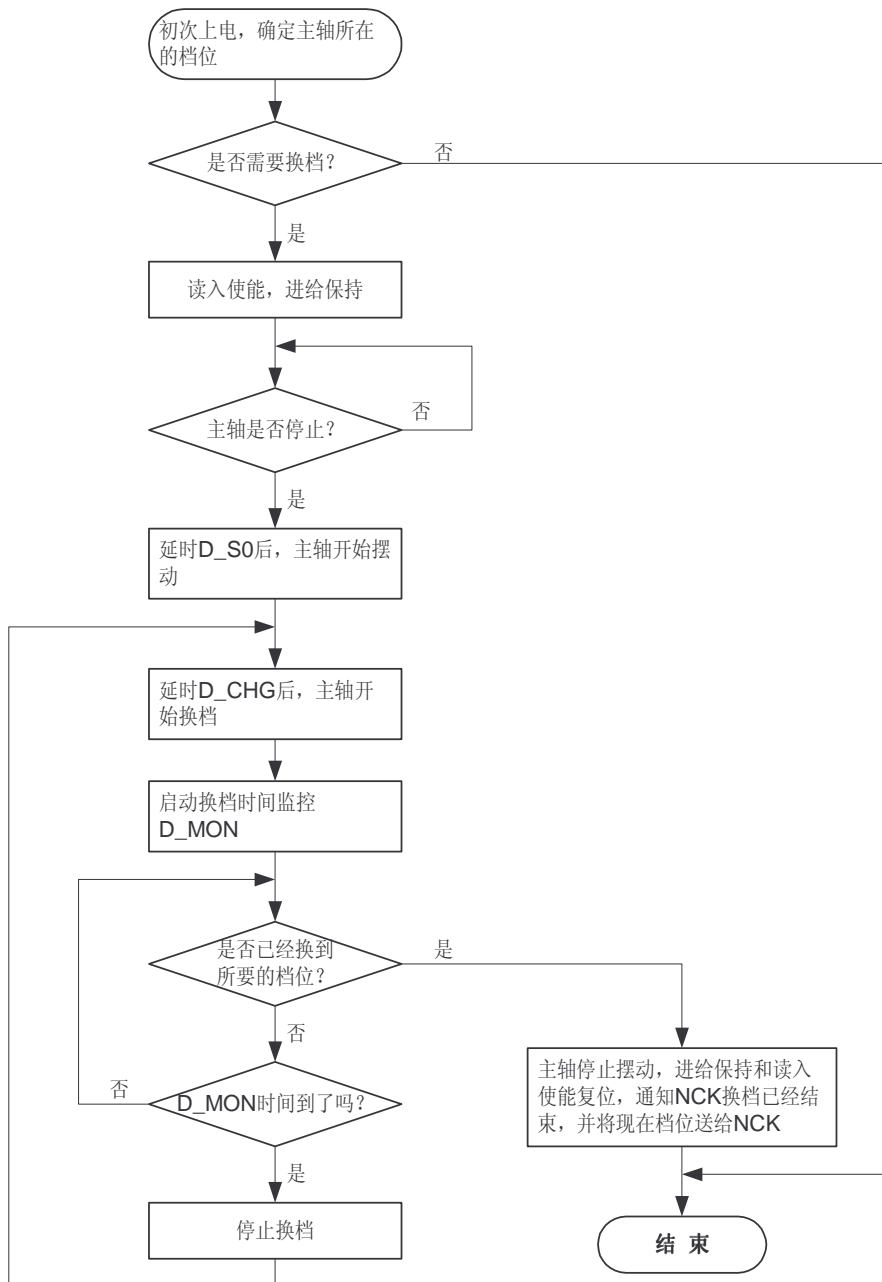
相关PLC机床参数 无

与主轴摆动相关的NC参数需要设定：

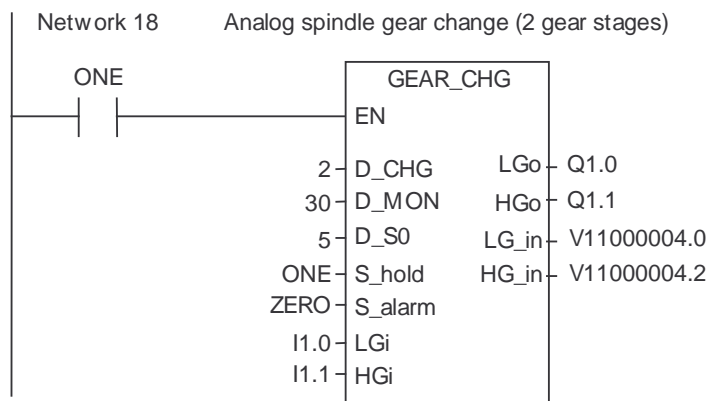
35010	GEAR_STEP_CHANGE_ENABLE	
–	1: 换挡使能	
35110	GEAR_STEP_MAX_VELO	
转/分	每挡最高速度（挡位号：[0]和[1]表示第一挡） 0,1...5	
35120	GEAR_STEP_MIN_VELO	
转/分	每挡最低速度（挡位号：[0]和[1]表示第一挡） 0,1...5	
35130	GEAR_STEP_MAX_VELO_LIMIT	
转/分	每挡最高速度极限（挡位号） 0,1...5	
35140	GEAR_STEP_MIN_VELO_LIMIT	
转/分	每挡最低速度极限（挡位号） 0,1...5	
36200	AX_VELO_LIMIT	
转/分	每挡最高监控速度（挡位号） 0,1...5	
35400	SPIND_OSCILL_DES_VELO	
转/分	摆动速度	
35410	SPIND_OSCILL_ACCEL	
转/秒 <sup>2</sup>	摆动加速度	
35430	SPIND_OSCILL_START_DIR	
–	起始摆动方向：3-正转；4-反转	
35440	SPIND_OSCILL_TIME_CW	
秒	正向摆动时间	
35450	SPIND_OSCILL_TIME_CCW	
秒	负向摆动时间	

注意：详细说明请参阅“安装调试手册”

子程序流程图



子程序调用实例



### 10.2.11. 子程序43 – MEAS\_JOG

#### 子程序43的目的

该子程序处理探头信息并且实现“手动方式测量”的功能。利用该子程序可以对探头进行校准以及刀具的测量。  
 使用该子程序的前提条件是在主程序OB1中调用子程序MCP\_NCK（SBR38）。如果在“手动方式测量”功能生效时改变操作方式，手动方式测量功能自动关闭。  
 该子程序可以激活下列报警信息：报警700031 – 手动方式测量功能生效

#### 局部变量定义

输入：ACTIVE\_TOOL DWORD      有效的刀具号 VD14000064  
 输出：MSG1                  BOOL      报警信息：手动方式测量功能生效

#### 占用的全局变量

MEAS_OPAUT	M240.0	自动方式下测量
MEAS_OPJOG	M240.1	手动方式下测量
CHL_HMI	M240.2	来自HMI信号：测量过程中方式变化
NO_KEY	M240.3	坐标轴无点动键
OUT_HMI	M240.4	来自HMI信号：操作方式自动
FDI_MEASJOG	M240.5	进给禁止 Meas_JOG
ON_MEASJOG	M240.6	启动 Meas_JOG
PROBE_ON	M240.7	探头信号释放
JOG_MEASJOG	M241.0	操作方式手动输出到 Meas_JOG
AUT_MEASJOG	M241.1	操作方式手动输出到Meas_JOG
CHL_MEASJOG	M241.2	操作方式更改禁止到 Meas_JOG
KEY_MEASJOG	M241.3	点动键 Meas_JOG
RES_MEASJOG	M241.4	复位 Meas_JOG
ESC_MEASJOG	M241.5	中断 Meas_JOG
DRY_MEASJOG	M241.6	空运行 Meas_JOG
SBL_MEASJOG	M241.7	单段 Meas_JOG

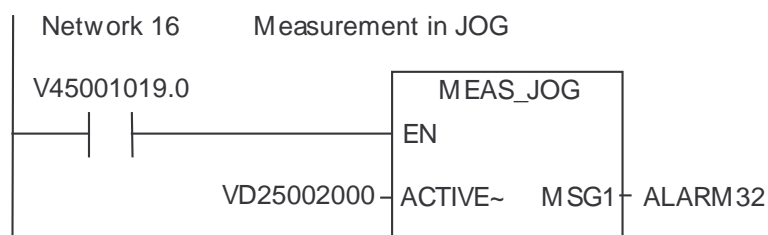
#### 相关PLC机床参数

无

#### 子程序流程图

请看802D功能描述

#### 子程序调用实例



### 10.2.12. 子程序44 – COOLING（冷却控制）

**子程序44的目的** 冷却可以通过操作面板MCP的冷却启动和停止键，或通过零件程序中的编程指令M07/M08和M09启动或停止。在急停、冷却电机过载、冷却液位低或在程序控制生效时，冷却输出禁止。

该子程序可以激活下列报警信息：

报警 700018 – 冷却泵电机过载

报警 700019 – 冷却液液位低

#### 局部变量定义

##### 输入：

NODEF	WORD	保留字
C_key	BOOL	手动操作键（触发信号）
OVload	BOOL	冷却电机过载（NC）
C_low	BOOL	冷却液液位低（NC）
C_dis	BOOL	冷却禁止（NC）

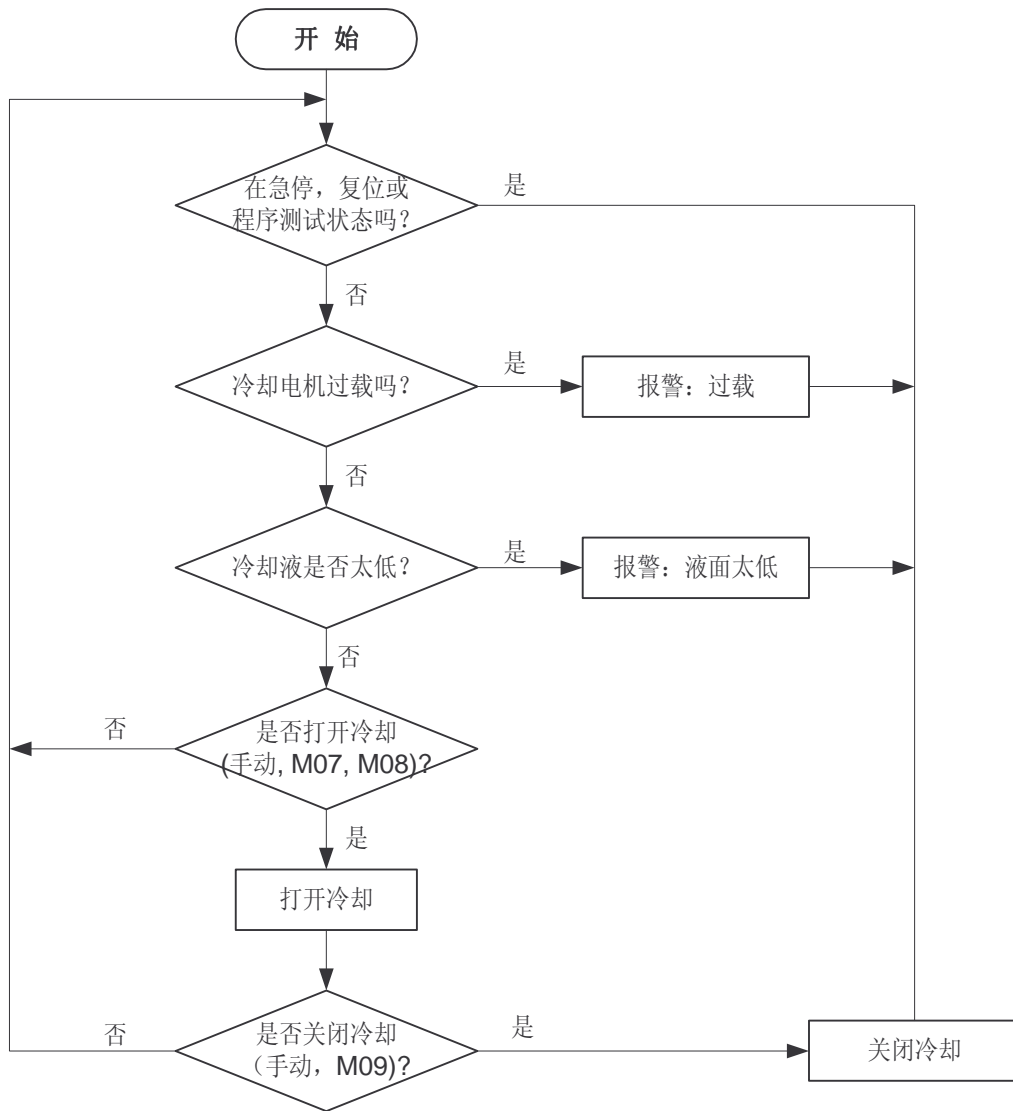
##### 输出：

C_out	BOOL	冷却输出
C_LED	BOOL	冷却输出状态显示
ERR1	BOOL	错误信息：冷却电机过载
ERR2	BOOL	错误信息：冷却液液位低

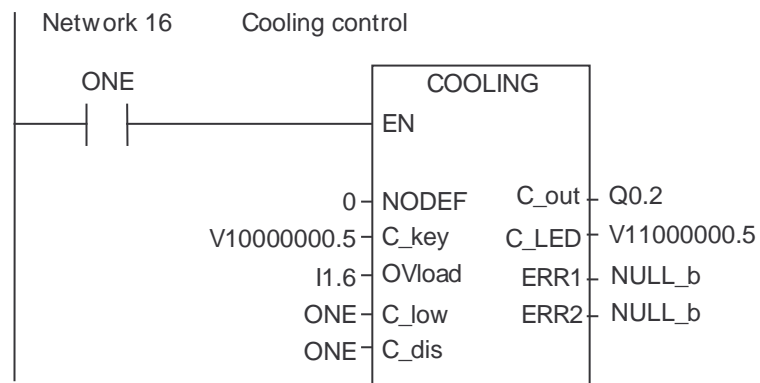
**占用的全局变量** MB151 冷却液开关状态

**相关PLC机床参数** 无

子程序流程图



子程序调用实例



### 10.2.13. 子程序45 – LUBRICATE（导轨润滑控制）

#### 子程序45的目的

导轨润滑是根据给定的时间间隔和给定的润滑时间进行控制（非坐标运动距离相关）。同时提供一个手动按键来启动润滑，并且可以在机床每次上电时自动启动润滑一次。正常情况下导轨的润滑是按规定的时间间隔周期性自动启动，每次按给定的时间润滑。在急停、润滑电机过载、润滑液位低等情况下润滑停止。

该子程序可以激活下列报警信息：

报警 700020 – 润滑电机过载

报警 700021 – 润滑液液位低

#### 局部变量定义

##### 输入：

Lintv	WORD	润滑时间间隔（单位：1分钟）
Ltime	WORD	每次润滑的时间（单位：0.01秒，最大327.67秒）
L_key	BOOL	手动润滑键（触发信号）
L1st	BOOL	方式选择：第一次PLC扫描启动一次润滑）
Ovload	BOOL	润滑电机过载（NC）
L_low	BOOL	润滑液液位低（NC）

##### 输出：

L_out	BOOL	润滑输出
L_LED	BOOL	润滑输出状态指示
ERR1	BOOL	错误信息：润滑电机过载
ERR2	BOOL	错误信息：润滑液液位低

#### 占用的全局变量

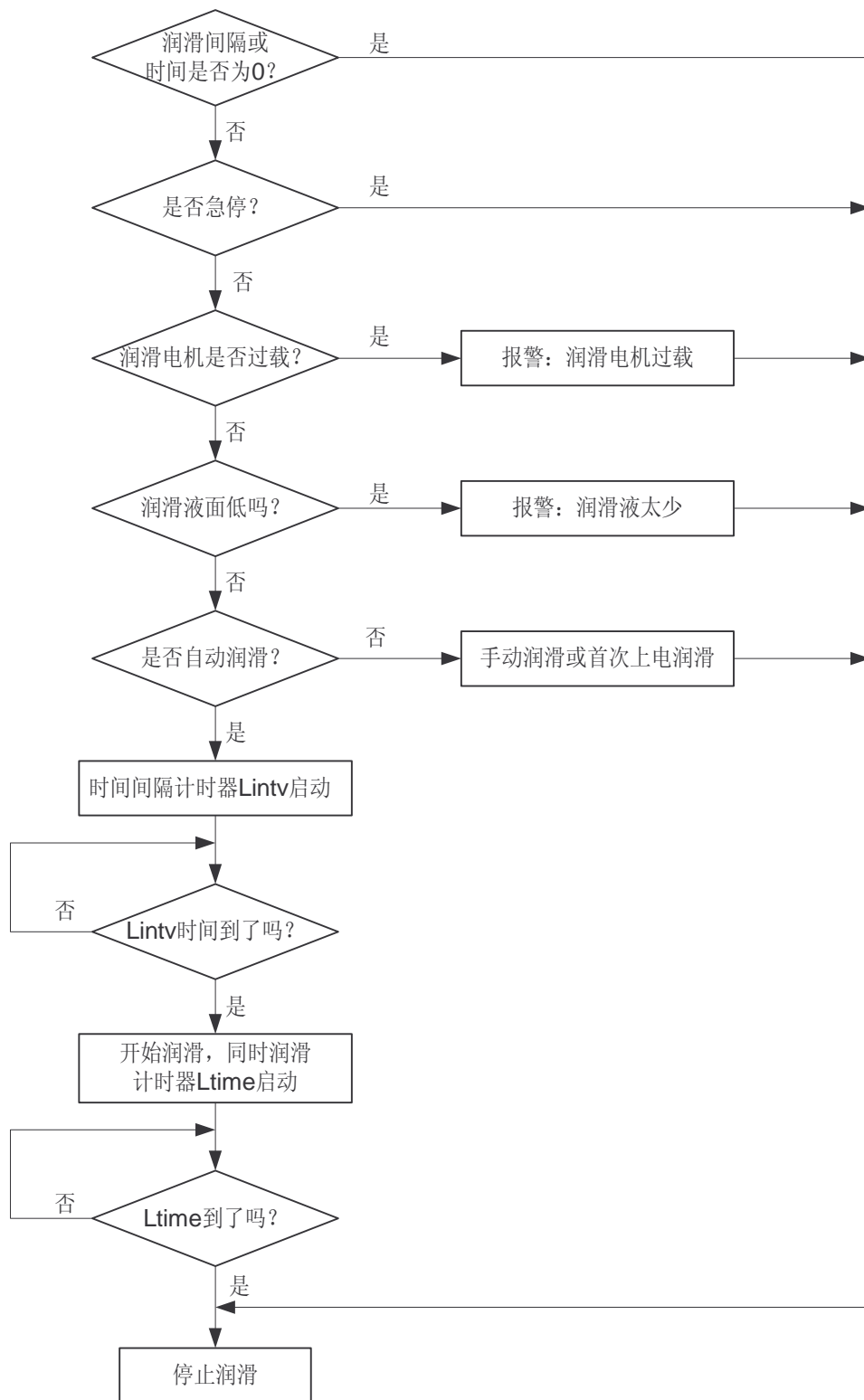
L_cmd	M152.0	润滑命令
L_interval	C24	作为润滑间隔计时器（单位：分钟）
L_time	T27	作为每次润滑的时间计时器（单位：0,01秒）

#### 相关PLC机床参数

MD 14510 [24]: 润滑间隔（单位：分钟）

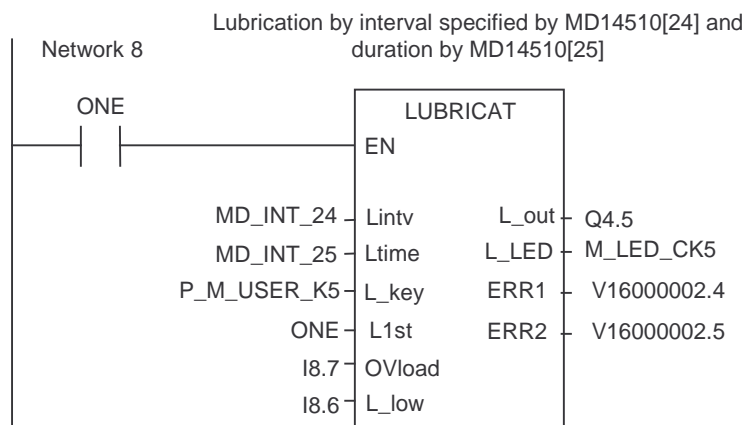
MD 14510 [25]: 每次润滑的时间（单位：0,01秒，最大327.67秒）

子程序流程图





子程序调用实例

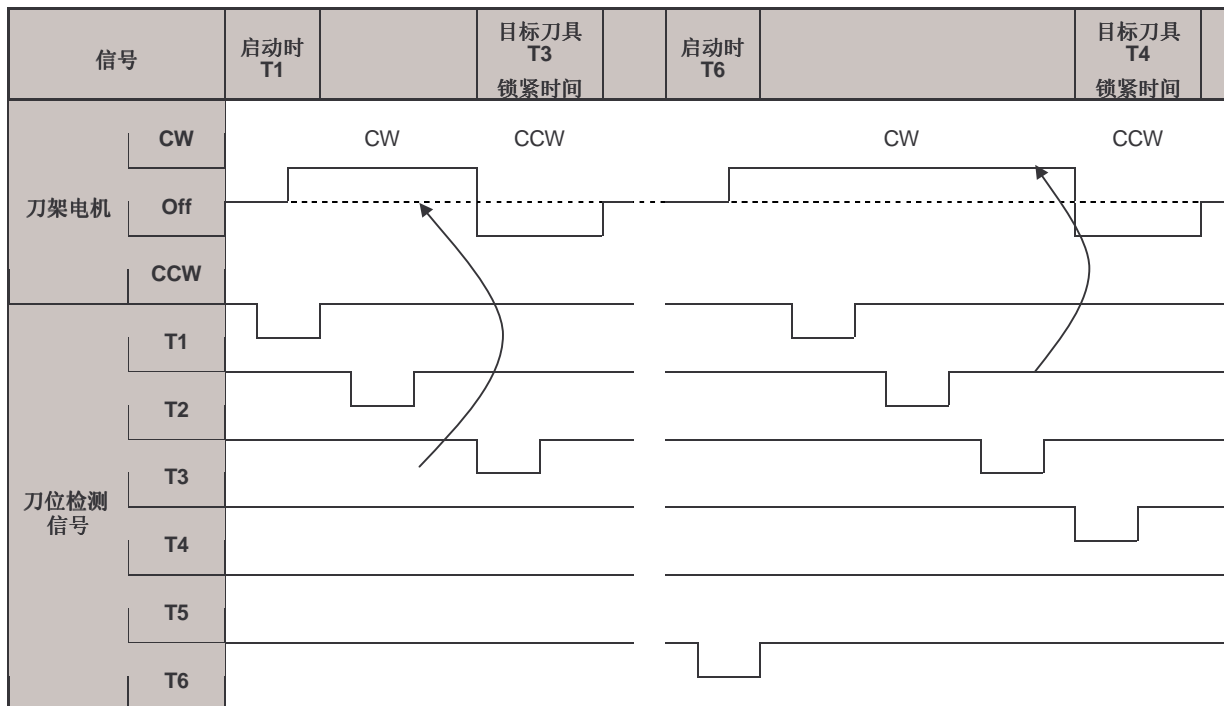


10.2.14. 子程序46 – TURRET1（霍尔元件刀架控制）

子程序46的目的

该子程序用于控制霍尔元件为刀位传感器的刀架，刀架电机由PLC控制。刀架正转找刀，目标刀具找刀后，刀架反转锁紧（反转时间可调）。在自动方式和MDA方式下，T功能启动的换刀动作。在手动方式下，短击机床面板MCP上的换刀键，可使刀架转一个刀位；长时间按换刀键可连续找刀，松开按键刀架自动锁紧。在换刀过程中NC接口信号“读入禁止”（V32000006.1）和“进给保持”（V32000006.0）置位。这样加工程序将等待换刀完成后，方可继续运行。在急停、刀架电机过载或程序测试PRT（Program Test）时，刀架转动禁止。

霍尔元件刀架换刀时序图：



该子程序可以激活下列报警信息:

- 报警 700022 – 刀架电机过载
- 报警 700023 – 编程刀具号大于刀架最大刀位数
- 报警 700024 – 在监控时间内, 没有找到目标刀具
- 报警 700025 – 刀架无刀位检测信号

### 局部变量定义

#### 输入:

Tmax	WORD	刀架最大刀位数
C_time	WORD	刀架反转锁紧时间 (单位: 0.01秒)
M_time	WORD	换刀监控视监 (单位: 0.01秒)
T_01 ... T_07	BOOL	刀位传感器 (低电平有效)
T_key	BOOL	手动换刀键 (触发信号)
OVload	BOOL	刀架电机过载 (NC)

#### 输出:

T_cw	BOOL	刀架找刀输出
T_ccw	BOOL	刀架锁紧输出
T_LED	BOOL	换刀过程状态显示
ERR1	BOOL	错误信息: 无刀位检测信号
ERR2	BOOL	错误信息: 编程刀具超出刀架范围
ERR3	BOOL	错误信息: 找刀监控时间到, 但未到刀目标刀具
ERR4	BOOL	错误信息: 刀架电机过载

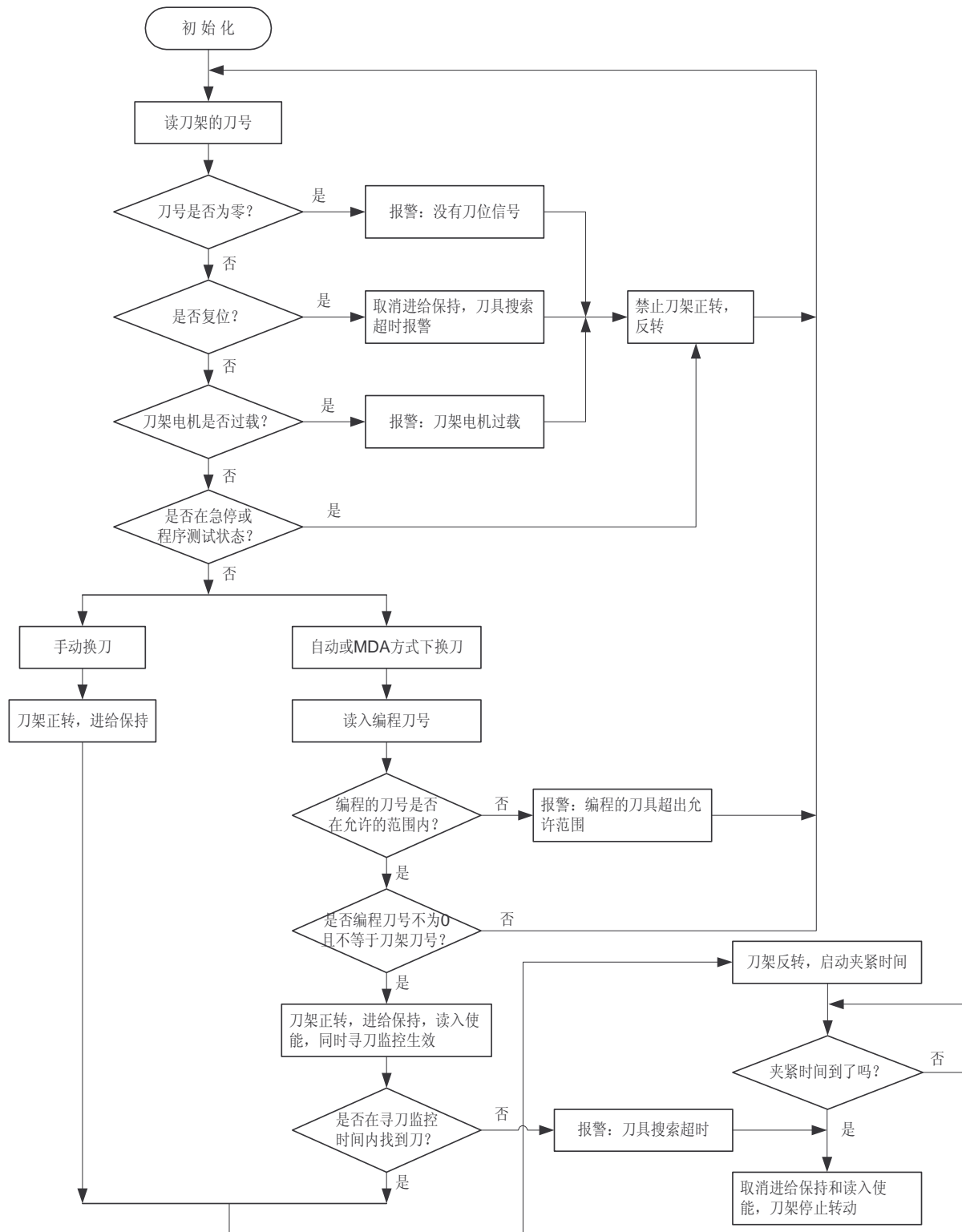
### 占用的全局变量

	MB154	换刀中间状态
	MB155	换刀中间状态
C_TIMER	T30	刀架锁紧时间定时器
M_TIMER	T31	换刀监控时间定时器

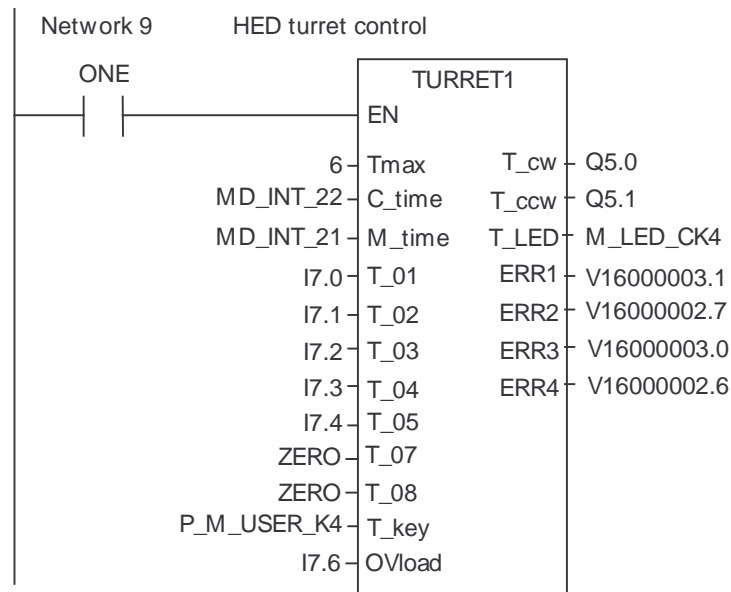
### 相关PLC机床参数

MD14510[20]:	刀架刀位数 4 / 6 / 8
MD14510[21]:	找到监控时间 (单位: 0.01秒)
MD14510[22]:	刀架锁紧时间 (单位: 0.01秒)

子程序流程图



## 子程序调用实例



## 10.2.15. 子程序47 – TURRET2（二进制编码器刀架换刀控制）

## 子程序47的目的

该子程序的目的是作为控制具有编码器刀位检测信号、双向换刀的刀架控制程序的实例。关于刀架的工作原理以及刀架的换刀时序请与刀架供货商联系。在换刀过程中NC接口信号“读入禁止”（V32000006.1）和“进给保持”（V32000006.0）置位。这样加工程序将等待换刀完成后，继续运行。

在急停、刀架电机过载或程序测试PRT（Program Test）时，刀架转动禁止。

该子程序可以激活下列报警信息：

报警700022 – 刀架电机过载

报警700023 – 编程刀具号大于刀架最大刀位数

## 局部变量定义

## 输入：

Tmax	DWORD	刀架最大刀位数
Delay	WORD	安全延时（单位：0.01秒）
T_1	BOOL	刀码 A x 1
T_2	BOOL	刀码 B x 2
T_3	BOOL	刀码 C x 4
T_4	BOOL	刀码 D x 8
Parity	BOOL	校验位
Strobe	BOOL	选通位
OVload	BOOL	刀架电机过载（NC）
T_key	BOOL	手动换刀键（NO）

**输出:**

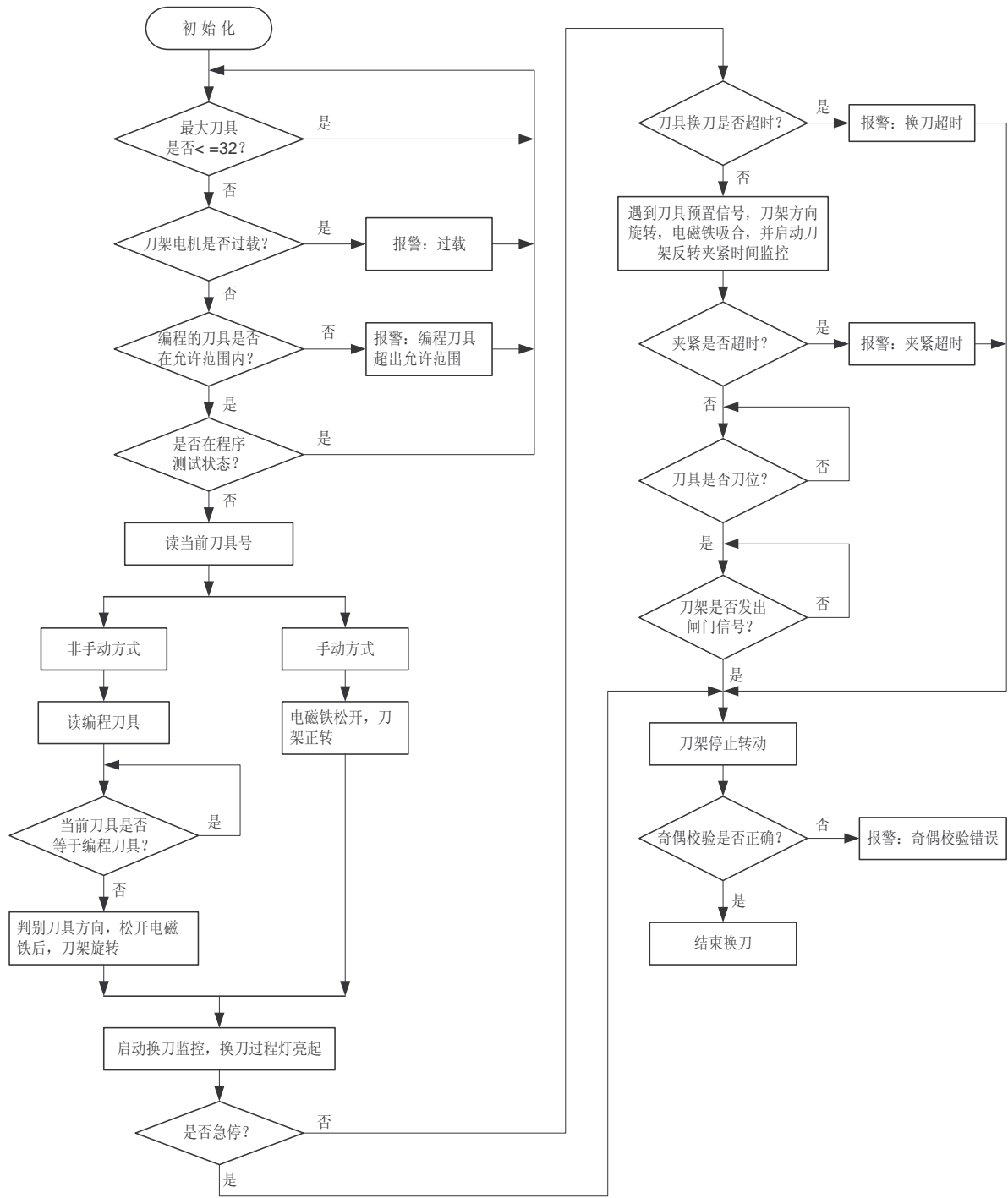
T_cw	BOOL	刀架正转输出 CW
T_ccw	BOOL	刀架反转输出 CCW
ERR1	BOOL	错误信息: 刀架电机过载
ERR2	BOOL	错误信息: 编程刀具号大于刀架最大刀位数
ERR3	BOOL	错误信息: 刀架没有锁紧 (校验和或无选通)
ERR4	BOOL	错误信息: 刀架定位超时
ERR5	BOOL	错误信息: 刀架锁紧超时

**占用的全局变量**

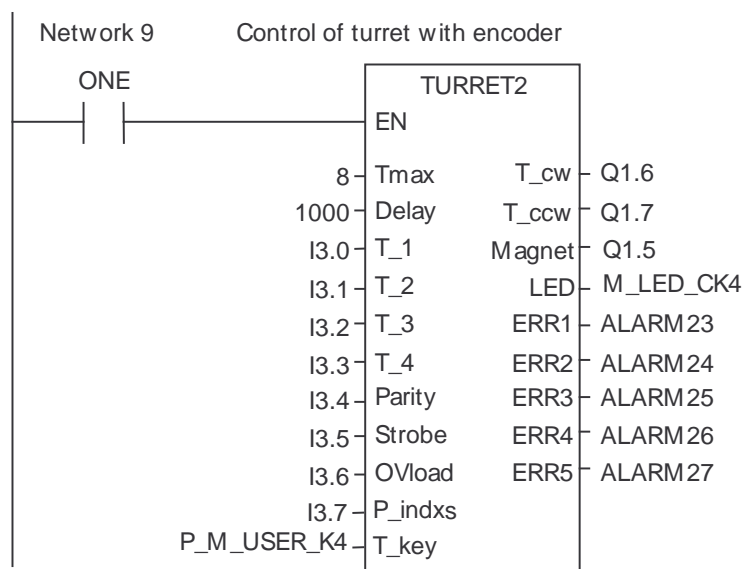
T_cw_m	M156.0	刀架正转标志 CW
T_ccw_m	M156.1	刀架反转标志CCW
T__P_INDX	MD160	手动方式下监控刀位变化的缓冲器
T_DES	MD164	目标刀号
T_DIR	M168.0	就近换刀方向
T_POS	M168.1	刀架找刀完毕到位
T_LOCK	M168.2	刀架锁紧命令
T_MAG	M168.3	用于刀架锁紧的电磁铁

**相关PLC机床参数** 无

子程序流程图



子程序调用实例



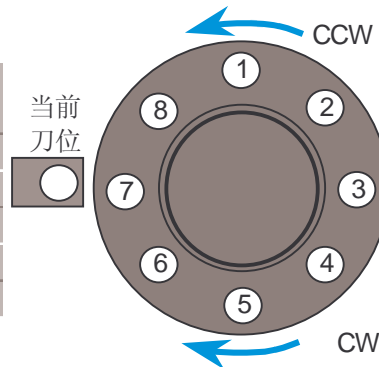
10.2.16. 子程序48 – TOOL\_DIR (判断就近换刀的方向)

子程序48的目的

该子程序的目的是判断出就近找刀的方向以及预停刀位（既在就近方向上，目标刀具的前一个刀位）。判断的条件是刀架或刀库的最大刀位数和编程的刀具号。该子程序可以用于车床的刀架或加工中心刀库的就近找刀控制。

举例

	当前刀位	编程刀号	预停刀位	方向
1	7	2	1	反 CCW
2	7	5	6	正 CW
3	3	8	1	正 CW
4	1	4	3	反 CCW
5	6	8	7	反 CCW



局部变量定义

输入:

Tmax	DWORD	刀架或刀库的最大刀位数
Pnum	DWORD	编程刀具号
Pcurr	DWORD	刀架或刀库当前位置

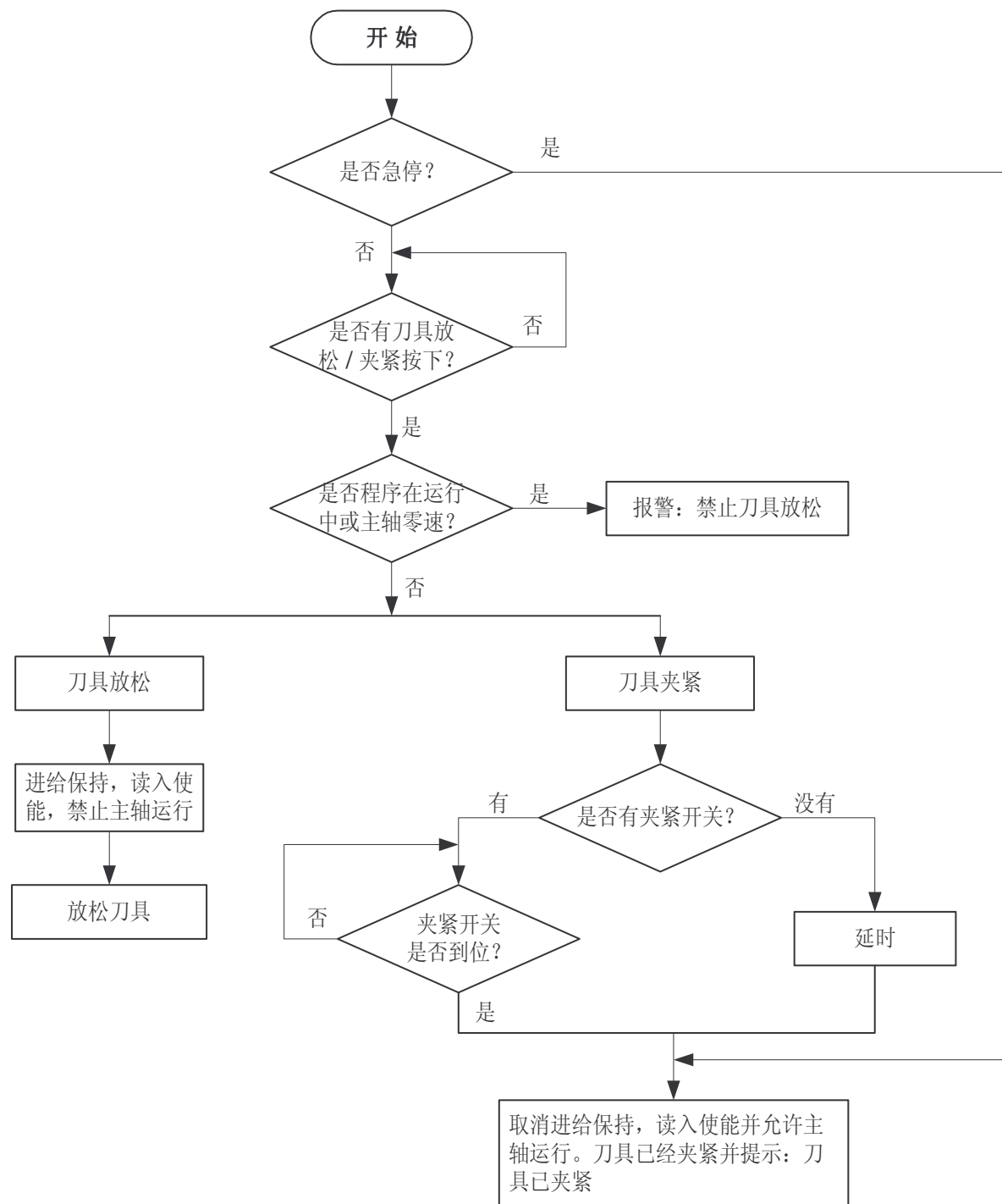
输出:

P_INDXo	DWORD	预停刀位: 在就近找刀方向上, 目标刀位的前一个刀位
DIR	BOOL	换刀方向: 1 – 正向CW; 0 – 反向CCW

占用的全局变量 无

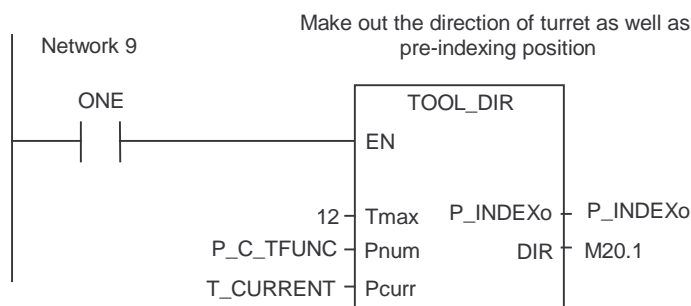
相关PLC 机床参数 无

子程序流程图





## 子程序调用实例



## 10.2.17. 子程序49 – LOCK\_UNL（锁紧/放松控制）

## 子程序49的目的

该子程序的目的是控制铣床的松刀紧刀或车床的卡盘卡紧和放松。

卡紧和放松的条件是零件程序没有执行、且主轴处于静止状态。为了保证安全在卡紧放松的过程中NC的“进给保持”信号（V32000006.0=1）置位。该子程序可支持具有到位检测开关或无检测开关的锁紧放松机构。对于配有到位检测开关的锁紧机构，只有在锁紧到位信号有效时进给保持才自动复位。对于无到位检测开关的锁紧机构，紧给保持经延时后自动复位。

该子程序可以激活下列报警信息：

- 报警700018 – 主轴运行过程中不允许进行锁紧放松操作
- 报警700019 – 锁紧未到位

## 局部变量定义

## 输入：

DELAY	WORD	锁紧延时（单位：2x PLC扫描周期）
CONF	BOOL	程序配置：1/0 – 有/ 无锁紧 到位检测开关
KEY	BOOL	手动锁紧放松键（NO）
EX_K	BOOL	外部手动锁紧放松键（NO）
S_VELO	BOOL	主轴速度状态：1- 主轴停止/ 0 – 主轴运行
CLPi	BOOL	锁紧到位传感器（NO）

## 输出：

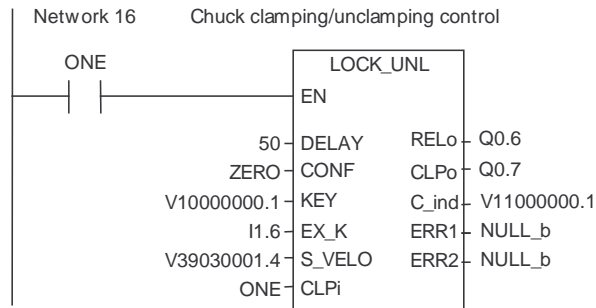
RELo	BOOL	放松输出
CLPo	BOOL	锁紧输出
C_ind	BOOL	锁紧到位状态指示
ERR1	BOOL	错误信息：主轴运行过程中禁止锁紧放松
ERR2	BOOL	错误信息：锁紧未到位

## 占用的全局变量

EOD	M153.0	延时结束
TR_st	M153.1	释放状态
TR_om	M153.2	释放缓冲输出
CLAMPING	C29	卡紧延时

相关PLC 机床参数 无

子程序调用实例



### 10.2.18. 子程序50 – MGZ\_SRCH（搜索目标刀具所在的刀套位置）

**子程序50的目的** 该子程序的目的是在刀套表中查找目标刀具所在的刀套位置。刀套表的建立和结构请参见子程序52的说明。

**局部变量定义**

**输入:**

P\_TOOL            DWORD            编程刀具号

**输出:**

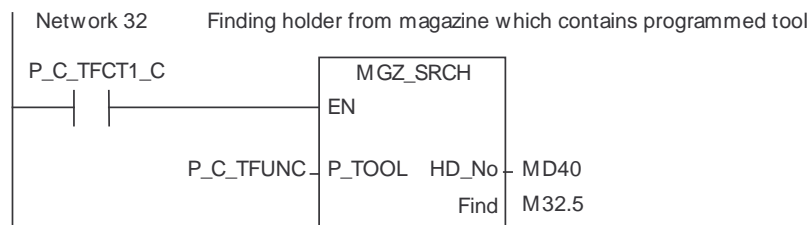
HD\_No            DWORD            编程刀具所在的刀套号

Find             BOOL             搜索结果：1-目标刀具找到；0-没有找到

占用的全局变量 无

相关PLC机床参数 无

子程序调用实例



**10.2.19. 子程序51 – MGZ\_RNEW（刷新刀套表）**

**子程序51的目的** 该子程序的目的是在换刀完毕后对刀套表进行刷新，即将主轴刀套（VB14000000）内的刀具号与目标刀套（VB140000xx）的内容（编程刀具号）交换。

**局部变量定义****输入:**

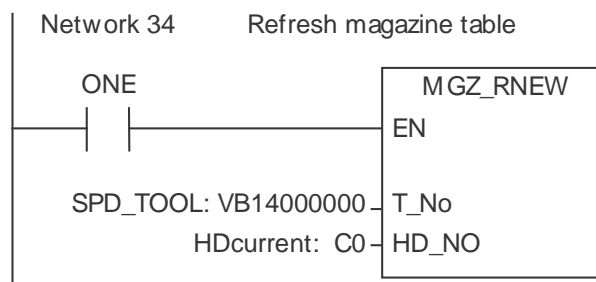
T\_No            DWORD            编程刀具号

**输出:**

HD\_No            DWORD            装有编程刀具的刀库刀套号

**占用的全局变量**    无

**相关PLC机床参数**    无

**子程序调用实例**

**10.2.20. 子程序52 – MGZ\_INI（初始化刀套表）****子程序52的目的**

为实现加工中心随机换刀的刀库和机械手的控制，PLC子程序库提供了相应的子程序。包括刀套表初始化、搜索目标刀具所在的刀套位置，以及刀套表的刷新。

子程序库定义刀套表的最大位数为40。初始化的目的是在可保持数据区VB14000000到VB14000040建立刀套表，对应刀库中的最多40把刀具。

初始化后，刀套表中每个刀套中具有与刀套号相同的刀具，并且规定主轴上无刀具。在换刀时，首先要定位（找到）装有编程刀具的刀套，刀套号用于控制刀库正转或反转。在机械手将刀套内的刀具与主轴上的刀具交换后，必须刷新刀表，即原主轴上的刀具号写入当前刀套表中，编程刀具号写入主轴刀套表。请参阅下表。

刀具在	刀套表	初始化后	T5 M06	T8 M06	T16 M6	T0 M06	T15 M06	T10 M06
SPINDLE	VB14000000	0	5	8	16	0	15	10
刀套 1	VB14000001	1	1	1	1	1	1	1
刀套 2	VB14000002	2	2	2	2	2	2	2
刀套 3	VB14000003	3	3	3	3	3	3	3
刀套 4	VB14000004	4	4	4	4	4	4	4
刀套 5	VB14000005	5	0	0	0	16	16	16
刀套 6	VB14000006	6	6	6	6	6	6	6
刀套 7	VB14000007	7	7	7	7	7	7	7
刀套 8	VB14000008	8	8	5	5	5	5	5
刀套 9	VB14000009	9	9	9	9	9	9	9
刀套 10	VB14000010	10	10	10	10	10	10	15
刀套11	VB14000011	11	11	11	11	11	11	11
刀套12	VB14000012	12	12	12	12	12	12	12
刀套13	VB14000013	13	13	13	13	13	13	13
刀套14	VB14000014	14	14	14	14	14	14	14
刀套15	VB14000015	15	15	15	15	15	0	0
刀套16	VB14000016	16	16	16	8	8	8	8

802D最多允许64把刀具，但子程序库最多允许40把刀。对于多于40把刀的刀库，需要修改下列子程序：SBR52-MGZ\_INI，SBR50-MGZ\_SRCH，和SBR51-MGZ\_RNEW。

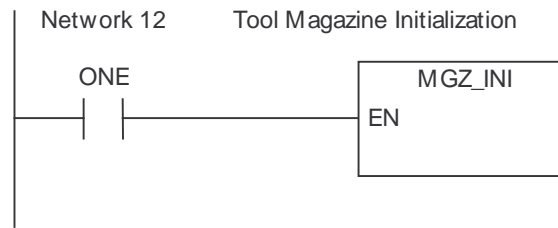
**注意：**刀库和机械手的控制逻辑因生产厂家而异，所以在设计刀库和机械手的应用程序时请仔细阅读由刀库制造商提供的使用说明和控制时序图。

局部变量定义 无

占用的全局变量 无

相关PLC机床参数 无

## 子程序调用实例



## 10.2.21. 子程序63: TOGGLE

## 子程序63的目的

保持开关：按一次开关闭合，再按一次开关关断；延时开关：按一次开关（触发信号），开关闭合且保持一定时间后自动关断。该子程序提供了六个保持开关和两个延时开关。延时时间可设定。

子程序的按键输入和输出可与任何物理输入输出连接。所有未使用的开关的输入为“ZERO”（M251.0）、输出为“NULL\_b”（M255.7）。

## 局部变量定义

## 输入:

Delay7	WORD	开关7的延时时间（单位：2*PLC周期）
Delay8	WORD	开关8的延时时间（单位：2*PLC周期）
Ki_1...Ki_8	BOOL	保持开关一的输入...保持开关八的输入

## 输出:

Ko_1...Ko_8	BOOL	保持开关一的输出...保持开关八的输出
-------------	------	---------------------

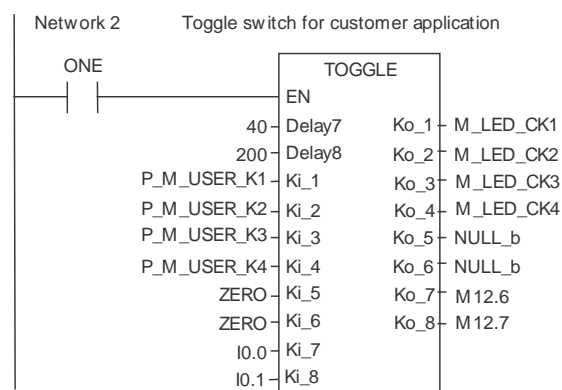
## 占用的全局变量

K1st1 ... K8st1	MB245	保持开关状态1
K1st2 ... K8st2	MB246	保持开关状态2
K1ston ... K8ston	MB247	保持开关开状态

## 相关PLC机床参数

无

## 子程序调用实例



## 10.3. PLC实例应用程序

### 10.3.1. MCP仿真应用程序

**目的** 该应用程序MCP\_SIMULATION\_802D.PTP的目的是提供一种在没有机床控制面板的情况下操作802D的方案。原理是利用子程序37，且借助PLC编程工具软件的状态表来实现方式转换、点动控制、反参考点控制、改变倍率、选择增量等操作。


#### 主程序的结构（OB1）

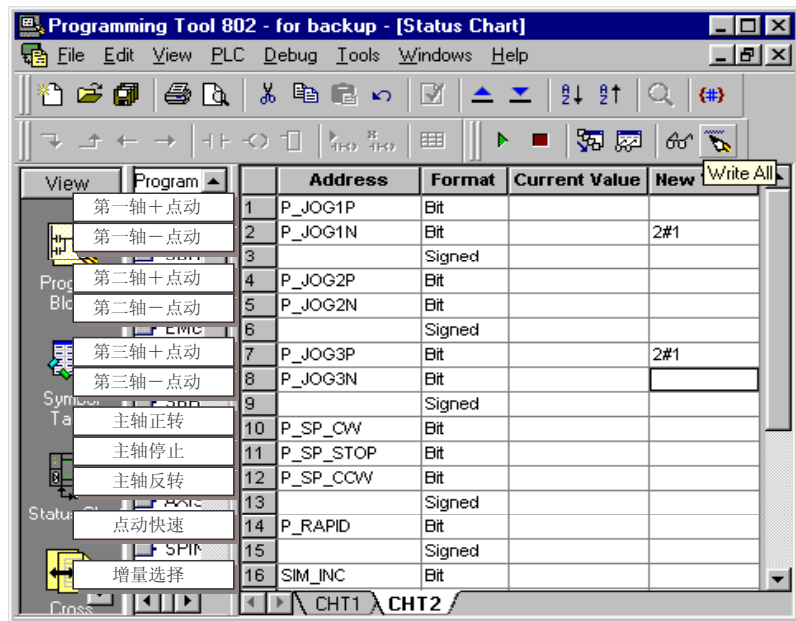
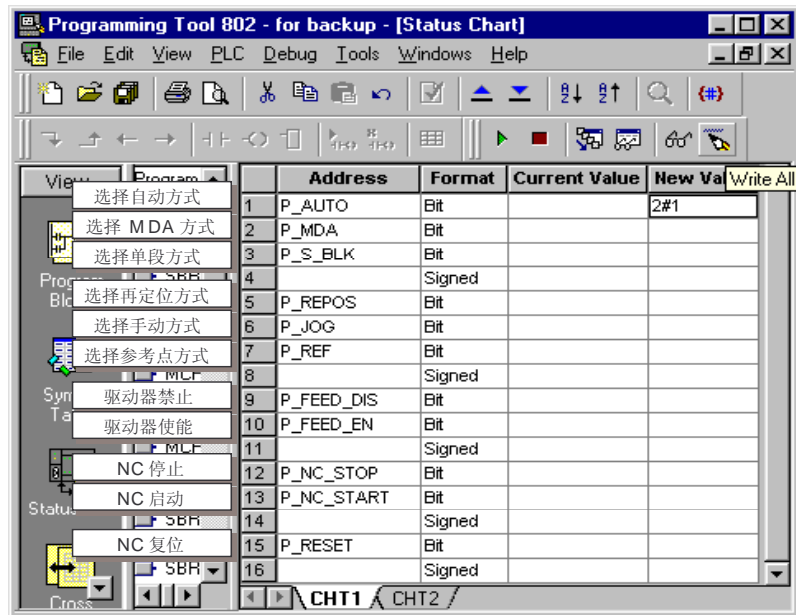
调用条件	调用的子程序	说明
第一次扫描 (SM0.1)	PLC_INI (SBR32)	PLC初始化
每次扫描 (SM0.0)	EMG_STOP (SBR33)	驱动器使能控制
每次扫描 (SM0.0)	MCP_SIMU (SBR37)	机床控制面板 MCP 仿真
每次扫描 (SM0.0)	MCP_NCK (SBR38)	传递MCP信号致NCK 接口
每次扫描 (SM0.0)	AXES_CTL (SBR39)	使能NCK 接口信号

#### 相关的PLC机床参数

MD14510[16] – 机床类型： 1-车床； 2-铣床

#### 应用

1. 根据“安装调试手册”连接802D的各部件；
2. 802D及驱动器上电。且802D上选择“STEP7-连接”，并将连接开启；
3. 启动PLC编程软件 *Programming Tool PLC V3.1*且设置通讯参数与802D匹配。
4. 打开该应用程序MCP\_SIMULATION\_802D.PTP，并下载到802D中，然后启动PLC。
5. 打开PLC编程工具的状态表“Status Chart”在所需功能的新值栏中输入“1”，然后用鼠标击图标将值写入802D。



### 10.3.2. 车床PLC应用程序实例

**目的** 车床PLC应用程序实例SAMPLE\_TURN\_802D.PTP的目的是示范如何利用PLC子程序库中的子程序组织一个PLC应用程序。另外在系统配置和输入输出定义相同的情况下，可直接使用该应用程序。

该实例应用程序适合于具有下列配置的车床：

- 两个进给轴：X和Z；每轴的正负方向各有一个硬限位开关；
- 六工位霍尔元件刀架；
- PLC控制定时定量润滑系统；
- PLC控制冷却系统；
- 802D配置：
  - 一个数字主轴：1PH7 ...
  - 一个输入输出模块PP72/48：X111和X222接口用于联接802D机床控制面板MCP

其他机床输入输出的定义：PP72/48接口X333

信号	端子号	说明	备注
M	1	24VDC地	
L+	2	24VDC输出（作为X333输入信号的公共端）	
I 6.0	3	急停按钮	常闭信号
I 6.1	4	X轴“正”向限位开关	常闭信号
I 6.2	5	X轴“负”向限位开关	常闭信号
I 6.3	6	Z轴“正”向限位开关	常闭信号
I 6.4	7	Z轴“负”向限位开关	常闭信号
I 6.5	8	X轴参考点开关	常开信号
I 6.6	9	Z轴参考点开关	常开信号
I 6.7	10		
I 7.0	11	刀位检测信号T1*	低电平有效
I 7.1	12	刀位检测信号T2	低电平有效
I 7.2	13	刀位检测信号T3	低电平有效
I 7.3	14	刀位检测信号T4	低电平有效
I 7.4	15	刀位检测信号T5	低电平有效
I 7.5	16	刀位检测信号T6	低电平有效
I 7.6	17	刀架电机过载	常闭信号
I 7.7	18		
I 8.0	19	驱动器就绪：来自电源模块端子72（73.1接24V）	
I 8.1	20	I <sup>2</sup> t报警：来自电源模块端子52（51接24V）	
I 8.2	21		
I 8.3	22		
I 8.4	23	冷却液液位过低	常闭信号
I 8.5	24	冷却泵电机过载	常闭信号
I 8.6	25	润滑油液位过低	常闭信号
I 8.7	26	润滑油泵电机过载	常闭信号
	27,28, 29,30	无定义	

\*注：刀位检测信号任何时刻只有一个有效



信号	端子号	说明	备注
Q 4.0	31	通过继电器接驱动器电源模块的端子48	
Q 4.1	32	通过继电器接驱动器电源模块的端子63	
Q 4.2	33	通过继电器接驱动器电源模块的端子64	
Q 4.3	34	X 轴电机抱闸释放	
Q 4.4	35	冷却泵	
Q 4.5	36	润滑泵	
Q 4.6	37		
Q 4.7	38		
Q 5.0	39	刀架电机正转	
Q 5.1	40	刀架电机反转	
Q 5.2	41		
Q 5.3	42		
Q 5.4	43		
Q 5.5	44		
Q 5.6	45		
Q 5.7	46		
L+	47, 48	输出信号的公共端：24VDC	
L+	49, 50	输出信号的公共端：24VDC	

- 802D机床控制面板MCP的联接：  
PP72/48的X111接口通过电缆连接到MCP的X1201  
PP72/48的X222接口通过电缆连接到MCP的X1202  
注意：PP72/48与MCP的联接与铣床实例应用程序SAMPLE\_MILL\_802D.PTP的联接不同
- 机床面板MCP上用户键的定义：

用户键1	驱动器使能键（保持按键）
用户键2	
用户键3	
用户键4	手动换刀键
用户键5	手动润滑键
用户键6	手动冷却键

### 主程序的结构（OB1）

调用条件	调用的子程序	说明
第一次扫描 (SM0.1)	PLC_INI (SBR32)	PLC初始化
每次扫描 (SM0.0)	EMG_STOP (SBR33)	急停控制
每次扫描 (SM0.0)	MCP_802D (SBR34)	802D机床控制面板信号至V1000xxxx
每次扫描 (SM0.0)	MCP_NCK (SBR38)	MCP和HMI信号传送到NCK接口
每次扫描 (SM0.0)	AXES_CTL (SBR40)	坐标使能控制、硬限位等
每次扫描 (SM0.0)	COOLING (SBR44)	冷却控制
每次扫描 (SM0.0)	LUBRICATE (SBR45)	润滑控制
每次扫描 (SM0.0)	TURRET1 (SBR44)	霍尔元件刀架控制

## 相关的PLC机床参数

MD14510[16] = 1	- 车床.
MD14510[20] = 输入值	- # 刀架刀为数 (4/6/8)
MD14510[21] = 输入值	- 换刀监控时间 (单位: 0.01S)
MD14510[22] = 输入值	- 刀架锁紧时间 (单位: 0.01S)
MD14510[24] = 输入值	- 润滑间隔 (单位: 1.0 Min)
MD14510[25] = 输入值	- 润滑时间 (单位: 0.01S)
MD14512[18] Bit3 = 1	- 机床控制面板上K1 键用于驱动器使能

## 10.3.3. 铣床PLC应用程序实例

## 目的

铣床PLC应用程序实例SAMPLE\_MILL\_802D.PTP的目的是示范如何利用PLC子程序库中的子程序组织一个PLC应用程序。另外在系统配置和输入输出定义相同的情况下, 可直接使用该应用程序。

该实例应用程序适合于具有下列配置的铣床:

- 三个进给轴: X、Y和Z; 每轴的正负方向各有一个硬限位开关;
- PLC控制定时定量润滑系统;
- PLC控制冷却系统;
- 802D配置:
  - 一个数字主轴: 1PH7 ...
  - 一个输入输出模块PP72/48: X222和X333接口用于联接802D机床控制面板MCP

其他机床输入输出的定义:

信号	端子号	说明	备注
M	1	24VDC地	
L+	2	24VDC 输出 (作为X333输入信号的公共端)	
I 0.0	3	急停按钮	常闭信号
I 0.1	4	X轴“正”向限位开关	常闭信号
I 0.2	5	X轴“负”向限位开关	常闭信号
I 0.3	6	Y轴“正”向限位开关	常闭信号
I 0.4	7	Y轴“负”向限位开关	常闭信号
I 0.5	8	Z轴“正”向限位开关	常闭信号
I 0.6	9	Z轴“负”向限位开关	常闭信号
I 0.7	10	X轴参考点开关	常开信号
I 1.0	11	Y轴参考点开关	常开信号
I 1.1	12	Z轴参考点开关	常开信号
I 1.2	13		
I 1.3	14		
I 1.4	15		
I 1.5	16		
I 1.6	17		
I 1.7	18		
I 2.0	19	驱动器就绪: 来自电源模块端子72 (73.1接24V)	常开信号

信号	端子号	说明	备注
I 2.1	20	I <sup>2</sup> /t报警：来自电源模块端子52（51接24V）	常开信号
I 2.2	21		
I 2.3	22		
I 2.4	23	冷却液液位过低	常闭信号
I 2.5	24	冷却泵电机过载	常闭信号
I 2.6	25	润滑油液位过低	常闭信号
I 2.7	26	润滑泵电机过载	常闭信号
	27,28 29,30	无定义	

信号	端子号	说明	备注
Q 0.0	31	通过继电器接驱动器电源模块的端子48	
Q 0.1	32	通过继电器接驱动器电源模块的端子63	
Q 0.2	33	通过继电器接驱动器电源模块的端子64	
Q 0.3	34	Z轴电机抱闸释放	
Q 0.4	35	冷却泵	
Q 0.5	36	润滑泵	
Q 0.6	37		
Q 0.7	38		
Q 1.0	39		
Q 1.1	40		
Q 1.2	41		
Q 1.3	42		
Q 1.4	43		
Q 1.5	44		
Q 1.6	45		
Q 1.7	46		
L+	47, 48	输出信号的公共端：24VDC	
L+	49, 50	输出信号的公共端：24VDC	

- 802D机床控制面板MCP的联接：  
PP72/48的X222接口通过电缆连接到MCP的X1201  
PP72/48的X333接口通过电缆连接到MCP的X1202  
注意：PP72/48与MCP的联接与铣床实例应用程序SAMPLE\_TURN\_802D.P  
TP的联接不同
- 机床面板MCP上用户键的定义：

用户键1	驱动器使能键（保持按键）
用户键2	
用户键3	
用户键4	
用户键5	手动润滑键
用户键6	手动冷却键

**主程序的结构 (OB1)**

调用条件	子程序名	说明
第一次扫描 (SM0.1)	PLC_INI (SBR32)	PLC初始化
每次扫描 (SM0.0)	EMG_STOP (SBR33)	急停控制
每次扫描 (SM0.0)	MCP_802D (SBR34)	802D机床控制面板信号至V1000xxxx
每次扫描 (SM0.0)	MCP_NCK (SBR38)	MCP和HMI信号传送到NCK接口
每次扫描 (SM0.0)	AXES_CTL (SBR40)	坐标使能控制、硬限位等
每次扫描 (SM0.0)	COOLING (SBR44)	冷却控制
每次扫描 (SM0.0)	LUBRICAT (SBR45)	润滑控制

**相关的PLC机床参数**

- MD14510[16] = 2 - 铣床  
 MD14510[24] = 输入值 - 润滑间隔 (单位: 1.0 Min)  
 MD14510[25] = 输入值 - 润滑时间 (单位: 0.01S)  
 MD14512[18] Bit3 = 1 - 机床控制面板上K1键用于驱动器使能

注: 首先装载铣床初始化文件setup\_M.ini (4个进给轴和一个主轴) 到802D。然后修改通道参数MD20070[5]=0去掉A1轴。

**重要事项**

必须根据“安装调试手册”设定相关的机床参数, 才能使系统正常工作。请参阅“安装调试手册”。

## 10.4. 附录 – 更改说明

- 系统资源  
64个用户报警（系统软件版本高于2.0）
- 子程序库的报警结构  
PLC子程序库中的报警不再固定在子程序的内部，而是以局部变量的形式连接到子程序的外部。因此制造商可以根据情况在PLC主程序中激活相应的报警也可以通过数字输出显示报警，如发光二极管。
- PLC机床数据  
请参阅第1.2.5.章：PLC机床数据
- 关于子程序的改动
  1. 子程序33 – EMG\_STOP:
    - 增加报警输出
  2. 子程序34 – MCP\_802D
    - 增加内部接口信号用于手持单元
    - 增加PLC机床参数用于机床控制面板上点动键布局
    - 增加报警输出
  3. 子程序40 – AXIS\_CTL
    - 增加“超程释放”输入位，用于超程链控制逻辑
    - 增加选择输入“SPAD”位，用于主轴使能的取消（自动取消，或按“主轴停止”键）
    - 抱闸输出减少为一个（通过参数配置带抱闸的坐标轴）
    - 增加报警输出
  4. 子程序44 – COOLING
    - 增加报警输出
  5. 子程序45 – LUBRICATE
    - 增加报警输出
  6. 子程序46 – TURRET1
    - 增加换刀时间监控
    - 增加换刀安全功能
    - 增加报警输出
    - 增加新的报警内容
- 新增子程序
  1. 子程序36 – MINI\_HHU  
用于西门子小型手持单元（订货号：6FX2006-1BG00）
  2. 子程序41 – SPD\_GEAR  
车床或铣床主轴换挡控制（两个齿轮级）
  3. 子程序49 – LOCK\_UNL  
用于车床卡盘的卡紧和放松，或铣床的刀具卡紧和放松控制



# 11. 附录

## 11.1. 附录一：PLC数据类型和地址定义

数据类型	BOOLEAN	机床数据位（1 或 0）
	BYTE	整数值（从-128 到127）
	DOUBLE	实数和整数值（从 $4.19 \times 10^{-307}$ 到 $1.67 \times 10^{308}$ ）
	DWORD	整数值（从 $-2.147 \times 10^9$ 到 $2.147 \times 10^9$ ）
	STRING	由带有数字和下划线的大写字母组成字符串 （最大16个字符）
	UNSIGNED WORD	整数值（从0到65536）
	SIGNED WORD	整数值（从-32768 到32767）
	UNSIGNED DWORD	整数值（从0 到 4294967300）
	SIGNED DWORD	整数值（从-2147483650 到 2147483649）
	WORD	十六进制数（从0000 到 FFFF）
	DWORD	十六进制数（从00000000 到FFFFFFFF）
	FLOAT DWORD	实数（从 $8.43 \times 10^{-37}$ 到 $3.37 \times 10^{38}$ ）

### 地址定义 PLC 地址定义及范围

操作地址符	说明	范围
V	数据	V00000000.0 到 V99999999.7
T	计时器	T0 到 T15, 单位: 100 ms
C	计数器	C0 到 C31
I	数字输入	I0.0 到 I7.7
Q	数字输出	Q0.0 到 Q7.7
M	标志位	M0.0 到 M127.7
SM	特殊标志位	SM0.0 到 SM0.6
A	ACCU（逻辑）	AC0 到 AC1（Udword）
A	ACCU（算术）	AC2 到 AC3（Dword）

### 地址区V 地址区V的组成

类型标记 (DB号)	区号 (通道号, 轴号)	分区	偏移
00	00	0	000
(00-99)	(00-99)	(0-9)	(000-999)

## 特殊标记位



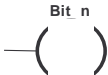
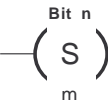
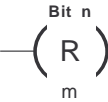


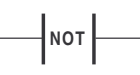
## 特殊标记位定义（只读）

位	说明
SM0.0	定义常“1”信号
SM0.1	第一次PLC循环‘1’，后面循环‘0’
SM0.2	缓冲数据丢失：只适用于第一次PLC循环（‘0’信号时数据不丢失，‘1’信号时数据丢失）
SM0.3	重新启动：第一次PLC循环‘1’，后面循环‘0’
SM0.4	60秒周期的脉冲（占空比，30秒‘0’，30秒‘1’）
SM0.5	1秒周期的脉冲（占空比，0.5秒‘0’，0.5秒‘1’）
SM0.6	PLC循环周期（交替循环‘0’和循环‘1’）



## 11.2. 附录二：PLC指令表

## 二进制位操作

二进制位操作		
指令	阶梯图符号	操作数
常开触点 n=1 闭合 n=0 断开		n: V, I, Q, M, SM, C, T
常闭触点 n=0 闭合 n=1 断开		n: V, I, Q, M, SM, C, T
位输出 前端 0, n=0 前端 1, n=1		n: V, I, Q, M, C, T
置位 前端 0, 不置位 前端 1 or ä, n 置位 1		n: V, I, Q, M, C, T m: 1
复位 前端 0, 不复位 前端 1 or ä, n 复位 0		n: V, I, Q, M, C, T m: 1
上升沿 前端 ä 闭合 (1个 PLC 周期)		
下降沿 前端 æ 闭合 (1个 PLC 周期)		
逻辑非 前端 0, 结果 1 前端 1, 结果 0		

## 定时器 / 计数器

定时器, 计数器指令		
指令	阶梯图符号	操作数
保持定时器 EN=1, 启动 EN=0, 停止 如果 $T_{value} \geq PT$ , $T_{bit}=1$		EN: V, I, Q, M, SM, C, T PT: VW, IW, QW, MW, AC, K
延时定时器 EN=1, 启动 EN=0, 复位 如果 $T_{value} \geq PT$ , $T_{bit}=1$		EN: V, I, Q, M, SM, C, T PT: W, IW, QW, MW, AC, K
加计数器 CU ä, 计数值+1 R=1, 复位 如果 $C_{value} \geq PV$ , $C_{bit}=1$		EN: V, I, Q, M, SM, C, T R: V, I, Q, M, SM, C, T PT: VW, IW, QW, MW, AC, K
加减计数器 CU ä, 计数值+1 CD ä, 计数值-1 R=1, 复位 如果 $C_{value} \geq PV$ , $C_{bit}=1$		EN: V, I, Q, M, SM, C, T R: V, I, Q, M, SM, C, T PT: VW, IW, QW, MW, AC, K

## 比较类

比较类 (带符号字节)		
指令	阶梯图符号	操作数
字节 == a = b 闭合 a ≠ b 断开		a, b: VB, IB, QB, MB, AC, K
字节 >= a >= b 闭合 a < b 断开		a, b: VB, IB, QB, MB, AC, K
字节 <= a <= b 闭合 a > b 断开		a, b: VB, IB, QB, MB, AC, K

## 比较

比较（带符号字）		
指令	阶梯图符号	操作数
字 == a = b 闭合 a ≠ b 断开		a, b: VW, IW, QW, MW, AC, K
字 >= a >= b 闭合 a < b 断开		a, b: VW, IW, QW, MW, AC, K
字 <= a <= b 闭合 a > b 断开		a, b: VW, IW, QW, MW, AC, K
比较（带符号双字）		
指令	阶梯图符号	操作数
长字 == a = b 闭合 a ≠ b 断开		a, b: VD, ID, QD, MD, AC, K
长字 >= a >= b 闭合 a < b 断开		a, b: VD, ID, QD, MD, AC, K
长字 <= a <= b 闭合 a > b 断开		a, b: VD, ID, QD, MD, AC, K
比较（带符号实数）		
指令	阶梯图符号	操作数
浮点 == a = b 闭合 a ≠ b 断开		a, b: VD, ID, QD, MD, AC, K
浮点 >= a >= b 闭合 a < b 断开		a, b: VD, ID, QD, MD, AC, K
浮点 <= a <= b 闭合 a > b 断开		a, b: VD, ID, QD, MD, AC, K

## 算术指令

算术指令		
指令	阶梯图符号	操作数
字 相加  如果 EN =1, OUT=IN1 + IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VW, IW, QW, MW, AC, T, C, K ( $<\pm 2^{15}-1$ ) OUT: VW, QW, MW, T, C, AC
长字 相加  如果 EN =1, OUT=IN1 + IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VD, ID, QD, MD, AC, K ( $<\pm 2^{31}-1$ ) OUT: VD, QD, MD, AC
浮点 相加  如果 EN =1, OUT=IN1 + IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VD, ID, QD, MD, AC, K OUT: VD, QD, MD, AC
字 相减  如果 EN =1, OUT=IN1 - IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VW, IW, QW, MW, AC, T, C, K ( $<\pm 2^{15}-1$ ) OUT: VW, QW, MW, T, C, AC
长字 相减  如果 EN =1, OUT=IN1 - IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VD, ID, QD, MD, AC, K ( $<\pm 2^{31}-1$ ) OUT: VD, QD, MD, AC
浮点 相减  如果 EN =1, OUT=IN1 - IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VD, ID, QD, MD, AC, K OUT: VD, QD, MD, AC
字 相乘  如果 EN =1, OUT=IN1 * IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VW, IW, QW, MW, AC, T, C, K ( $<\pm 2^{15}-1$ ) OUT: VD, QD, MD, AC
浮点 相乘  如果 EN =1, OUT=IN1 * IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VD, ID, QD, MD, AC, K OUT: VD, QD, MD, AC
字 相除  如果 EN =1, OUT=IN1 / IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VW, IW, QW, MW, AC, T, C, K ( $<\pm 2^{15}-1$ ) OUT: VD, QD, MD, AC

算术指令		
指令	阶梯图符号	操作数
浮点 相除  如果 EN =1, OUT=IN1 / IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VD, ID, QD, MD, AC, K OUT: VD, QD, MD, AC
平方根  如果 EN =1, OUT= $\sqrt{IN}$		EN: V, I, Q, M, SM, C, T IN1/IN2: VD, ID, QD, MD, AC, K ( $< \pm 2^{31} - 1$ ) OUT: VD, QD, MD, AC

## 逻辑指令

逻辑指令		
指令	阶梯图符号	操作数
字节 "与"  如果 EN =1, OUT=IN1 & IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VB, IB, QB, MB, AC, K ( $< 2^8$ ) OUT: VB, QB, MB, AC
字 "与"  如果 EN =1, OUT=IN1 & IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VW, IW, QW, MW, AC, K ( $< 2^{16}$ ) OUT: VW, QW, MW, AC
长字 "与"  如果 EN =1, OUT=IN1 & IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VD, ID, QD, MD, AC, K ( $< 2^{32}$ ) OUT: VD, QD, MD, AC
字节 "或"  如果 EN =1, OUT=IN1   IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VB, IB, QB, MB, AC, K ( $< 2^8$ ) OUT: VB, QB, MB, AC
字 "或"  如果 EN =1, OUT=IN1   IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VW, IW, QW, MW, AC, K ( $< 2^{16}$ ) OUT: VW, QW, MW, AC
长字 "或"  如果 EN =1,		EN: V, I, Q, M, SM, C, T IN1/IN2: VD, ID, QD, MD, AC, K ( $< 2^{32}$ )

逻辑指令		
指令	阶梯图符号	操作数
OUT=IN1   IN2		OUT: VD, QD, MD, AC
字节 "异或"  如果 EN =1, OUT=IN1 ^ IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VB, IB, QB, MB, AC, K (<math><2^8</math>) OUT: VB, QB, MB, AC
字 "异或"  如果 EN =1, OUT=IN1 ^ IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VW, IW, QW, MW, AC, K (<math><2^{16}</math>) OUT: VW, QW, MW, AC
长字 "异或"  如果 EN =1, OUT=IN1 ^ IN2		EN: V, I, Q, M, SM, C, T IN1/IN2: VD, ID, QD, MD, AC, K (<math><2^{32}</math>) OUT: VD, QD, MD, AC
字节 取反  如果 EN =1, OUT = ! IN		EN: V, I, Q, M, SM, C, T IN: VB, IB, QB, MB, AC, K (<math><2^8</math>) OUT: VB, QB, MB, AC
字 取反  如果 EN =1, OUT = ! IN		EN: V, I, Q, M, SM, C, T IN: VW, IW, QW, MW, AC, K (<math><2^{16}</math>) OUT: VW, QW, MW, AC
长字 取反  如果 EN =1, OUT = ! IN		EN: V, I, Q, M, SM, C, T IN: VD, ID, QD, MD, AC, K (<math><2^{32}</math>) OUT: VD, QD, MD, AC

## 增量减量指令

增量减量指令		
指令	阶梯图符号	操作数
字节 增量  如果 EN ä , OUT = IN + 1		EN: V, I, Q, M, SM, C, T IN: VB, IB, QB, MB, AC, K (<math>2^8</math>) OUT: VB, QB, MB, AC
字 增量  如果 EN ä , OUT = IN + 1		EN: V, I, Q, M, SM, C, T IN: VW, IW, QW, MW, AC, T, C, K (<math>2^{16}</math>) OUT: VW, QW, MW, AC
长字 增量  如果 EN ä , OUT = IN + 1		EN: V, I, Q, M, SM, C, T IN: VD, ID, QD, MD, AC, K (<math>2^{32}</math>) OUT: VD, QD, MD, AC
字节 减量  如果 EN ä , OUT = IN - 1		EN: V, I, Q, M, SM, C, T IN: VB, IB, QB, MB, AC, K (<math>2^8</math>) OUT: VB, QB, MB, AC
字 减量  如果 EN ä , OUT = IN - 1		EN: V, I, Q, M, SM, C, T IN: VW, IW, QW, MW, AC, T, C, K (<math>2^{16}</math>) OUT: VW, QW, MW, AC
长字 减量  如果 EN ä , OUT = IN - 1		EN: V, I, Q, M, SM, C, T IN: VD, ID, QD, MD, AC, K (<math>2^{32}</math>) OUT: VD, QD, MD, AC

## 数据格式转换指令

数据格式转换指令		
指令	阶梯图符号	操作数
整型 到 实型  如果 EN =1, OUT= float (IN)		EN: V, I, Q, M, SM, C, T IN: VD, ID, QD, MD, AC, K (<math>2^{32}</math>) OUT: VD, QD, MD, AC
实型 到 整型  如果 EN =1, OUT= trunc (IN)		EN: V, I, Q, M, SM, C, T IN: VD, ID, QD, MD, AC, K (<math>2^{32}</math>) OUT: VD, QD, MD, AC

## 程序控制指令

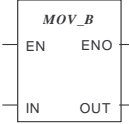
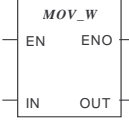
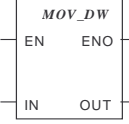
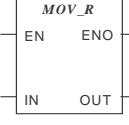
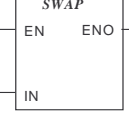
程序控制指令		
指令	阶梯图符号	操作数
转到标号  如果前端为 1		有条件
标号定义		常数 范围: 0...127
子程序调用  如果 EN=1		最多允许16个字节的局部参数
子程序返回		有条件 / 无条件



## 移位指令

移位指令		
指令	阶梯图符号	操作数
字节 左移  如果 EN ä , OUT=IN <<n 位		EN: V, I, Q, M, SM, C, T IN: VB, IB, QB, MB, AC, K (<2 <sup>8</sup> ) n: VB, IB, QB, MB, AC, K OUT: VB, QB, MB, AC
字 左移  如果 EN ä , OUT=IN <<n 位		EN: V, I, Q, M, SM, C, T IN: VW, IW, QW, MW, AC, T, C, K (<2 <sup>8</sup> ) n: VB, IB, QB, MB, AC, K OUT: VW, QW, MW, AC
长字 左移  If EN ä , OUT=IN <<n 位		EN: V, I, Q, M, SM, C, T IN: VD, ID, QD, MD, AC, K (<2 <sup>16</sup> ) n: VB, IB, QB, MB, AC, K OUT: VD, QD, MD, AC
字节 右移  如果 EN ä , OUT=IN >>n 位		EN: V, I, Q, M, SM, C, T IN: VB, IB, QB, MB, AC, K (<2 <sup>8</sup> ) n: VB, IB, QB, MB, AC, K OUT: VB, QB, MB, AC
字 右移  如果 EN ä , OUT=IN >>n 位		EN: V, I, Q, M, SM, C, T IN: VW, IW, QW, MW, AC, T, C, K (<2 <sup>8</sup> ) n: VB, IB, QB, MB, AC, K OUT: VW, QW, MW, AC
长字 右移  如果 EN ä , OUT=IN >>n 位		EN: V, I, Q, M, SM, C, T IN: VD, ID, QD, MD, AC, K (<2 <sup>16</sup> ) n: VB, IB, QB, MB, AC, K OUT: VD, QD, MD, AC

## 赋值指令

赋值指令		
指令	阶梯图符号	操作数
字节 赋值  如果 EN =1 , OUT= IN		EN: V, I, Q, M, SM, C, T IN: VB, IB, QB, MB, AC, K (<math>2^8</math>) OUT: VB, QB, MB, AC
字 赋值  如果 EN =1 , OUT= IN		EN: V, I, Q, M, SM, C, T IN: VW, IW, QW, MW, AC, T, C, K (<math>2^{16}</math>) OUT: VW, QW, MW, AC
长字 赋值  如果 EN =1 , OUT= IN		EN: V, I, Q, M, SM, C, T IN: VD, ID, QD, MD, AC, K (<math>2^{32}</math>) OUT: VD, QD, MD, AC
浮点 赋值  如果 EN =1 , OUT= IN		EN: V, I, Q, M, SM, C, T IN: VD, ID, QD, MD, AC, K (<math>2^{32}</math>) OUT: VD, QD, MD, AC
字节交换  如果 EN =1 , IN 高 & 低 字节交换		EN: V, I, Q, M, SM, C, T IN: VW, IW, QW, MW, AC, T, C, K (<math>2^{16}</math>)

SIEMENS NUMERICAL CONTROL  
LTD., NANJING, CHINA  
西门子数控（南京）有限公司

R&D Division  
No.18, Siemens Road, Jiangning Development Zone  
211100 NANJING  
People's Republic of China  
南京江宁经济开发区西门子路18号  
研发部  
邮编：211100  
电话：025—52101888  
传真：025—52101666

建议  
更正

出版/手册：  
SIUNERIK 802S/C/D PLC子程序库应用指南  
制造商文件

<p>此信来自</p> <p>姓名</p>	<p>技术手册</p> <p>订货号：</p> <p>版本：</p>
<p>公司/部门</p> <p>地址</p> <p>_____</p> <p>_____</p> <p>电话</p> <p>_____</p> <p>_____</p> <p>传真</p>	<p>当你阅读此刊物时若发现印刷错误，请在这张纸上通知我们。欢迎提出改进建议。</p>

建议和/或更正：