

第5章

PLC系统设计选型



PLC 是工业机器人系统集成的控制核心,主要用来协调工作站中各元器件的动作和功能,实现集成系统的自动化运行。

5.1 PLC 概述

PLC 以其结构紧凑、应用灵活、功能完善、操作方便、速度快、可靠性高、价格低等优点,已经越来越广泛地应用于自动化控制系统中,并且在自动化控制系统中起着非常重要的作用,已成为与分布式控制系统并驾齐驱的主流工业控制系统。世界上有 200 多个 PLC 生产厂家,如美国的 AB 公司、莫迪康公司、GE 公司,德国的西门子公司,日本的欧姆龙公司、三菱电机公司以及中国的浙江浙大中控信息技术有限公司等。对于不同的工业控制需求,应当选择合适的 PLC。

1. 控制系统任务的分析

随着 PLC 功能的不断完善,几乎可以用 PLC 完成所有的工业控制任务。但是,是否选择 PLC 控制,选择单台 PLC 控制、还是多台 PLC 的分散控制或分级控制,还应根据系统所需完成的控制任务、对被控对象的生产工艺及特点进行详细分析,特别是从以下几方面考虑。

控制规模: 一个控制系统的控制规模可用该系统的输入、输出设备总数来衡量,控制规模较大时,特别是开关量控制的输入、输出设备较多且连锁控制较多时,最适合采用 PLC 控制。

工艺复杂程度: 当工艺要求较复杂时,用继电器系统控制极不方便,而且造价也相应提高,甚至会超过 PLC 控制的成本,因此,采用 PLC 控制将有更大的优越性。特别是工艺要求经常变动或控制系统有扩充功能的要求时,则只能采用 PLC 控制。

可靠性要求: 虽然有些系统不太复杂,但对可靠性、抗干扰能力要求较高时,也需采用 PLC 控制。20 世纪 70 年代,一般认为 I/O 总数在 70 点左右时,可考虑 PLC 控制;到了 20 世纪 80 年代,一般认为 I/O 总数在 40 点左右就可以采用 PLC 控制;目前,由于 PLC 性

价比的进一步提高,当 I/O 点总数在 20 点甚至更少时,就趋向于选择 PLC 控制了。

数据处理速度:当数据的统计、计算规模较大,需要很大的存储器容量,且要求很高的运算速度时,可考虑带有上位计算机的 PLC 分级控制;如果数据处理程度较低,而主要以工业过程控制为主时,采用 PLC 控制将非常适宜。

2. PLC 选型原则

选择能满足控制要求的适当型号的 PLC 是应用设计中至关重要的一步。目前,国内外 PLC 生产厂家生产的 PLC 品种已达数百个,其性能各有特点。所以,在设计时,首先要尽可能考虑采用与单位正在使用的同系列的 PLC,以便于学习和掌握;其次是备件의通用性,可减少编程器的投资。此外,还要充分考虑下面因素,以便选择最佳型号的 PLC:

(1) 输入、输出设备的数量和性质根据系统的控制要求,详细给出 PLC 所有输入量和输出量的情况,包括:

有哪些开关量输入?电压分别是多少?尽量选择直流 24V 或交流 220V。

有哪些开关量输出?要求驱动功率为多少?一般的 PLC 输出驱动能力约 2A,如果容量不够,可以考虑输出功率的扩展,如在输出端接功率放大器、继电器等。

有哪些模拟量输入、输出?具体参数如何?PLC 的模拟量处理能力一般为 1~5V、0~10V,或 4~20mA。

在确定了 PLC 的控制规模后,一般还要考虑一定的余量,以适应工艺流程的变动及系统功能的扩充,一般可按 10% 的余量来考虑。另外,还要考虑 PLC 的结构,如果规模较大,以选用模块式的 PLC 为好。

(2) PLC 的特殊功能要求。控制对象不同会对 PLC 提出不同的控制要求。如用 PLC 替代继电器完成设备的生产过程控制、上下限报警控制、时序控制等,只需 PLC 的基本逻辑功能即可。对于需要模拟量控制的系统,则应选择配有模拟量输入/输出模块的 PLC,PLC 内部还应具有数字运算功能。对于需要进行数据处理和信息管理的系统,PLC 则应具有图表传送、数据库生成等功能。对于需要进行高速脉冲计数的系统,PLC 还应具有高数计数功能,且应了解系统所需的最高计数频率。有些系统,需要进行远程控制,就先配置具有远程 I/O 控制的 PLC。还有一些特殊功能,如温度控制、位置控制、PID 控制等,如果选择合适的 PLC 及相应的智能控制模块,将使系统设计变得非常简单。

(3) 被控对象对响应速度的要求。各种型号的 PLC 的指令执行速度差异很大,其响应时间也各不相同。一般来讲,不论哪种 PLC,其最大响应时间都等于输入、输出延迟时间及 2 倍的扫描时间三者之和。对于大多数被控对象来说,PLC 的响应时间都是能满足要求的,但对于某些要求快速响应的系统,则必须考虑 PLC 的最大响应时间是否满足要求。

(4) 用户程序存储器所需容量的估算。用户程序存储器的容量以地址(或步)为单位,每个地址可以存储一条指令,用户所需程序存储器的容量在程序编好后可以准确地计算出来,但在设计开始时往往办不到,通常需要进行估算,一般粗略的方法是

$$(I/O) \text{总数} \times (10 \sim 20) = \text{指令步数}$$

如果系统中含有模拟量,可以按每个模拟量通道相当于 16 个 I/O 点来考虑。比较复杂的系统,应适当增加存储器的容量,以免造成麻烦。

5.2 PLC 选型实践

5.2.1 PLC 系统设计

系统设计包括硬件设计和软件设计。所谓硬件设计,是指 PLC 及外围线路的设计,而软件设计即 PLC 程序的设计,包括系统初始化程序、主程序、子程序、中断程序、故障应急措施和辅助程序等。

1. 硬件设计

在硬件设计中,要进行输入设备的选择(如操作按钮、转换开关及模拟量的输入信号等)、执行元件(如接触器、电磁阀、信号灯等),以及控制台、柜的设计等。应根据 PLC 使用手册的说明,对 PLC 进行输入/输出通道分配及外部接线设计。在进行 I/O 通道分配时应做出 I/O 通道分配表,表中应包含 I/O 编号、设备代号,名称及功能,且应尽量将相同类型的信号、相同电压等级的信号排在一起,以便于施工。对于较大的控制系统,为便于软件设计,可根据工艺流程,将所需的计数器、定时器及内部辅助继电器也进行相应的分配,这些工作完成之后,就可以进行软件设计了。

2. 软件设计

软件设计的主要方法是先编写工艺流程图,将整个流程分解为若干步,确定每步的转换条件,配合分支、循环、跳转及某些特殊功能便可很容易地转为梯形图了。在编写梯形图时,经验法是非常重要的方法。因此,在平时要多注意积累经验。

软件设计可以与现场施工同步进行,即在硬件设计完成以后,同时进行软件设计和现场施工,以缩短施工周期。

系统调试分为两个阶段:第一阶段为模拟调试,第二阶段为联机调试。

当 PLC 的软件设计完成之后,应首先在实验室进行模拟调试,检查是否符合工艺要求。模拟调试可以根据所选机型,外接适当数量的输入开关作为模拟输入信号,通过输出端子的 LED,可观察 PLC 的输出是否满足要求。

当现场施工和软件设计都完成以后,就可以进行联机统调了。在统调时,一般应首先屏蔽外部输出,再利用编程器或编程软件的监控功能,采用分段分级调试方法,通过操作外部输入器件检查外部输入量是否连接无误,然后再利用 PLC 的强迫置位/复位功能逐个运行输出部件。

系统调试完成以后,为防止程序遭到破坏和丢失,可通过存储设备将程序保存起来。

5.2.2 CODESYS 软 PLC 的设计

CODESYS 软件由德国 Smart software solution GmbH 公司所开发,是可编程逻辑控制 PLC 的完整开发环境(CODESYS 是 Controlled Development System 的英文缩写),在 PLC 程序员编程时,CODESYS 为强大的 IEC 语言提供了一个简单的方法,系统的编辑器和调试器的功能是建立在高级编程语言的基础上(如 Visual C++)。

CODESYS 包括 PLC 编程、可视化 HMI、安全 PLC、控制器实时核、现场总线及运动控制,是一个完整的自动化软件。其功能强大,易于开发,可靠性高,开放性好并且集成了 PLC、可视化、运动控制及安全 PLC 的组件。

软件 PLC 综合了计算机和 PLC 的开关量控制、模拟量控制、数学运算、数值处理、网络通信、PID 调节等功能,通过一个多任务控制内核,提供强大的指令集、快速而准确的扫描周期、可靠的操作和可连接各种 I/O 系统及网络的开放式结构。所以,软件 PLC 提供了与硬件 PLC 同样的功能,同时又提供了 PC 环境。软 PLC 与硬 PLC 相比,还具有如下的优点:

(1) 具有开放的体系结构。软 PLC 具有多种 I/O 端口和各种现场总线接口,可在不同的硬件环境下使用,突破传统 PLC 对硬件的高度依赖,解决了传统 PLC 互不兼容的问题。

(2) 开发方便,可维护性强。软 PLC 是用软件形式实现硬 PLC 的功能,软 PLC 可以开发更为丰富的指令集,以方便实际工业的应用;并且软 PLC 遵循国际工业标准,支持多种编程语言,开发更加规范方便,维护更简单。

(3) 能充分利用 PC 机的资源。现代 PC 机强大的运算能力和飞速的处理速度,使得软 PLC 能对外界响应迅速作出反应,在短时间内处理大量的数据。利用 PC 机的软件平台,软 PLC 能处理一些比较复杂的数据和数据结构,如浮点数和字符串等。PC 机大容量的内存,使得开发几千个 I/O 端口简单方便。

(4) 降低对使用者的要求,方便用户使用。由于各厂商推出的传统 PLC 的编程方法差别很大,并且控制功能的完成需要依赖具体的硬件,工程人员必须经过专业的培训,掌握各个产品的内部接线和指令的使用。软 PLC 不依赖具体硬件,编程界面简洁友好,降低了使用者的入门门槛,节约培训费用。

(5) 打破了几家垄断的局面,有利于降低成本,促进软 PLC 技术的发展。

要实现软 PLC 控制功能,必须具有三个主要部分,即开发系统、对象控制器系统及 I/O 模块。开发系统主要负责编写程序,对软件进行开发。对象控制器及 I/O 模块是软 PLC 的核心,主要负责对采集的 I/O 信号进行处理、逻辑控制及信号输出的功能。

CODESYS 从架构上基本上可以分为三层,即应用开发层,通信层和设备层,如图 5-1 所示。

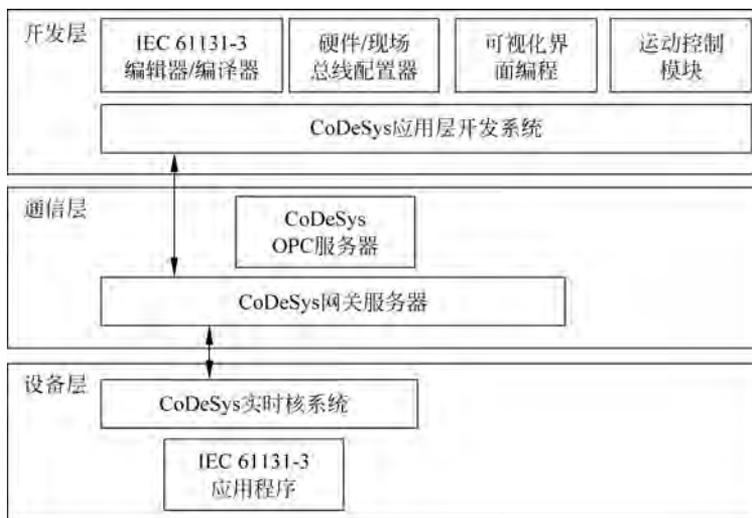


图 5-1 CODESYS 架构

软 PLC 开发系统实际上就是带有调试和编译功能的 PLC 编程软件,具备如下功能:编程语言标准化,遵循 IEC61131-3 标准,支持多语言编程(共有 5 种编程方式:IL,ST,LD,FBD 和 SFC),编程语言之间可以相互转换;丰富的控制模块,支持多种 PID 算法(如常规

PID 控制算法、自适应 PID 控制算法、模糊 PID 控制算法、智能 PID 控制算法等),还包括目前流行的一些控制算法,如神经网络控制;开放的控制算法接口,支持用户嵌入自己的控制算法模块;仿真运行,实时在线监控,在线修改程序和编译;网络功能;支持基于 TCP/IP 网络,通过网络实现 PLC 远程监控,远程程序修改等。

搬运码垛机器人工作站系统采用 CODESYS 软 PLC 利用 Windows 实时核作为 CPU,搭载倍福 I/O 硬件模块实现码垛系统设备的 I/O 控制。

CODESYS 软件安装步骤如下:

(1) 关闭杀毒软件,解压安装包,打开解压后的安装包,双击 `setup_codesysv3.5p10Patch2`,如图 5-2、图 5-3 所示。

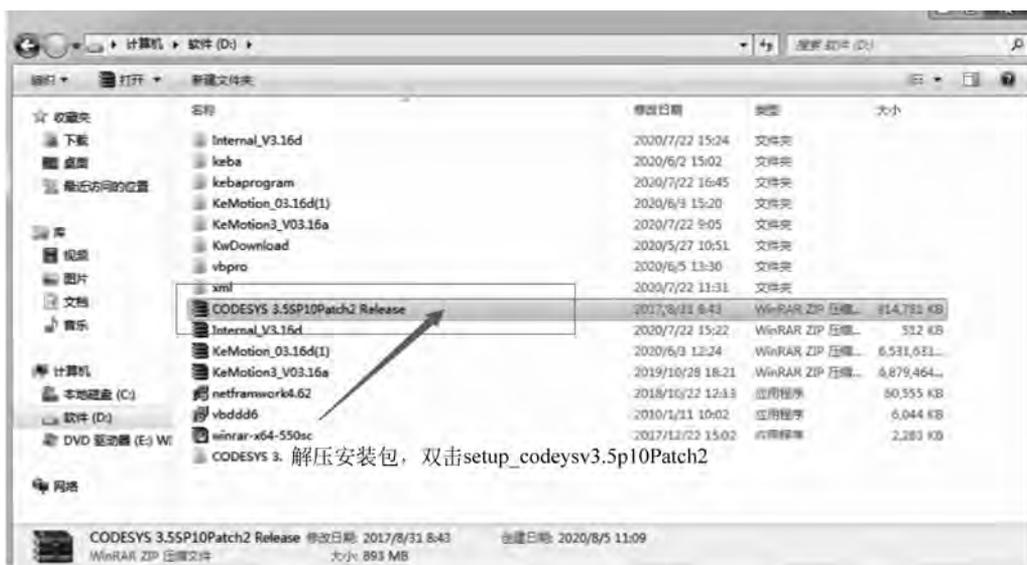


图 5-2 CODESYS 安装包解压



图 5-3 CODESYS 安装包安装

(2) 在图 5-4 所示界面,单击 Install,在图 5-5 单击 Next。

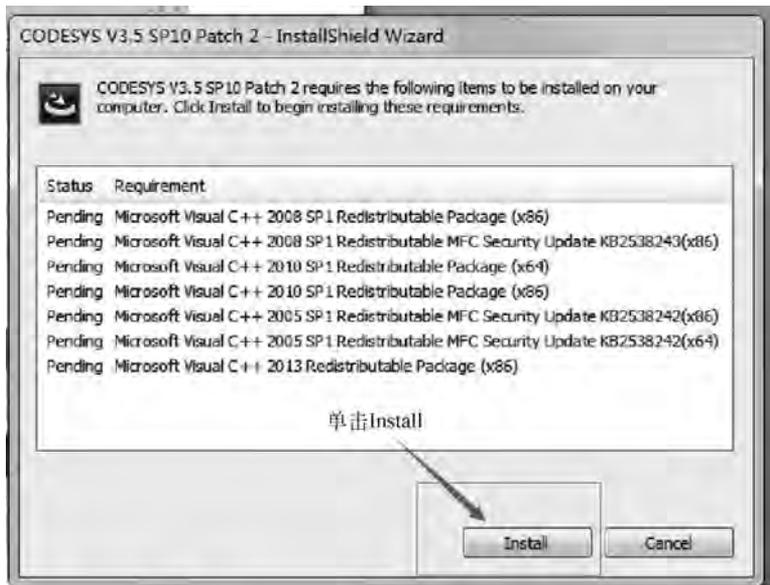


图 5-4 安装界面

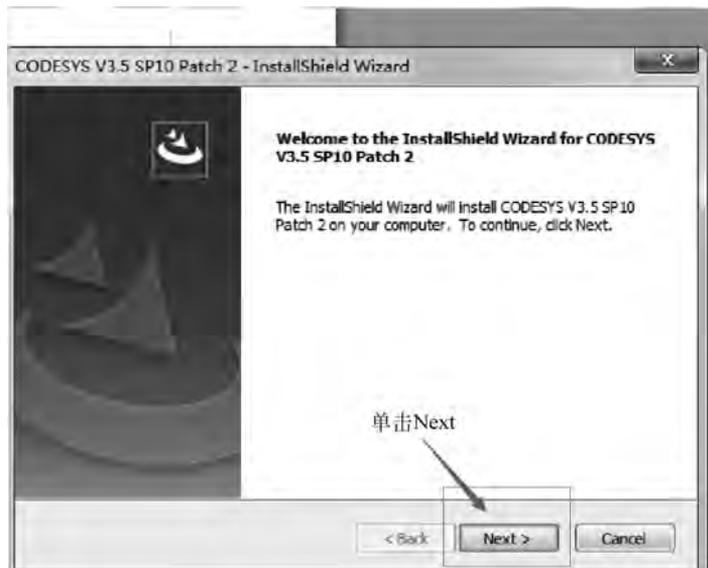


图 5-5 操作 Next 界面

(3) 如图 5-6 所示,选择并接受安装条款;如图 5-7 所示,可以安装在默认目录,也可以单击 Browse 浏览安装在其他目录下,但是要尽量保证安装目录是英文的。

(4) 选择要安装的 CODESYS 相关附属软件,如图 5-8 所示,全选并单击 Next。选择默认 CODESYS 文件夹并单击 Next,如图 5-9 所示。确认选择安装文件无误单击 Next,否则单击 Back,如图 5-10 所示。



图 5-6 接受安装条款界面

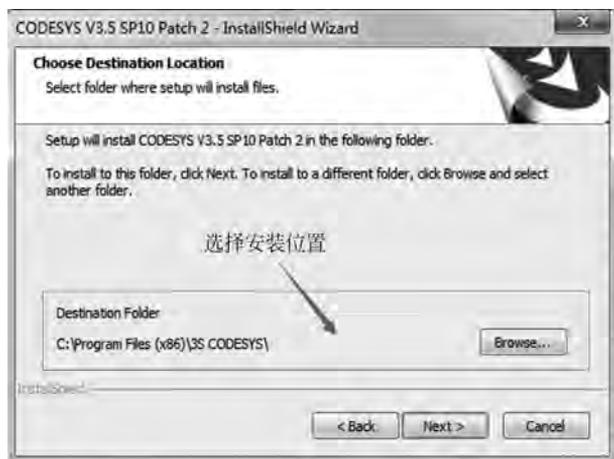


图 5-7 选择安装位置界面



图 5-8 选择要安装的附属软件界面

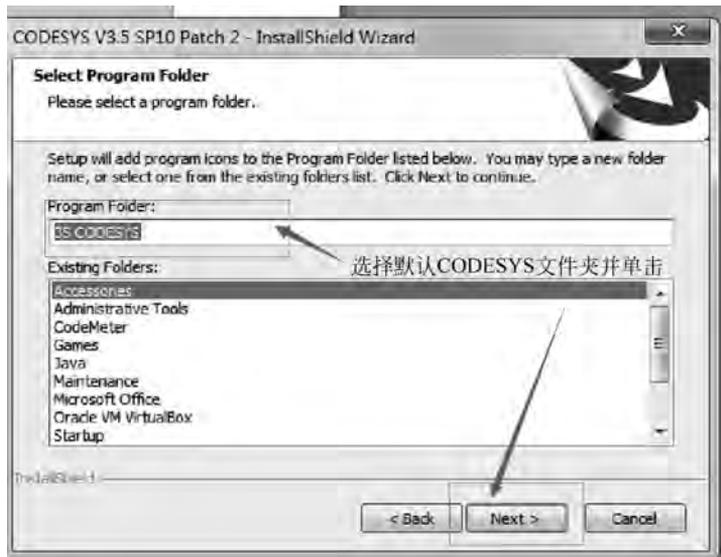


图 5-9 选择默认 CODESYS 文件夹界面

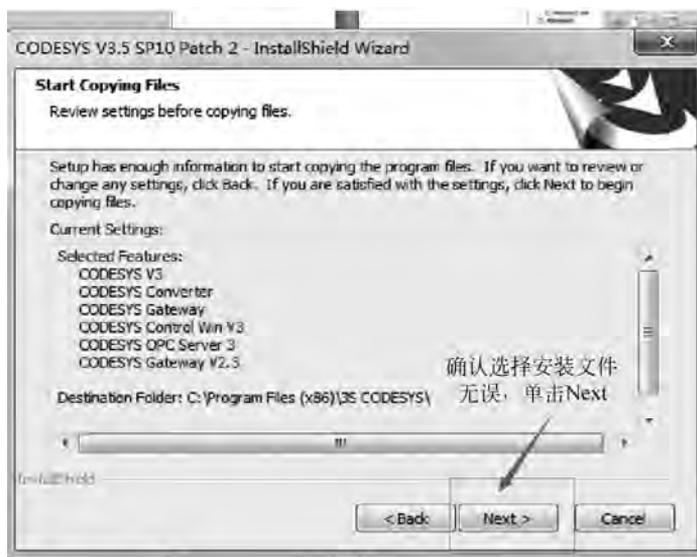


图 5-10 确认选择安装文件界面

(5) 经过 5~10min 的安装过程,会出现图 5-11 所示的界面,单击接受→Next。如图 5-12 所示,单击 Finish 完成软件安装。

(6) 注意安装完成后会在显示屏幕右下角出现如图 5-13 所示的三个图标,表示安装完成。CODESYS 软件程序编译、下载方法如下:

程序编写完成后,在下载之前需要对程序进行编译。编译命令对编写的程序进行语法检查,并且只编译添加到任务中的程序。编译指令不生成任何代码,只针对 POU 的语法进行检查。直接执行设备登录命令,系统也将默认执行编译指令(等同于先手动执行编译命令),再编译检查没有语法错误后执行连接登录指令。CODESYS 编译菜单如图 5-14 所示。



图 5-11 安装过程界面

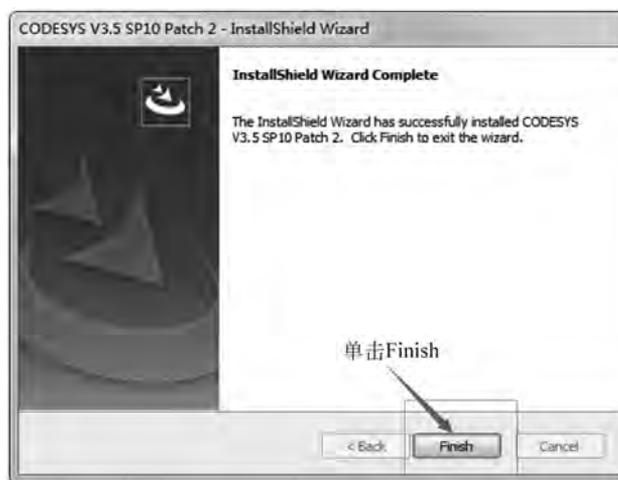


图 5-12 完成软件安装界面



图 5-13 安装完成图标

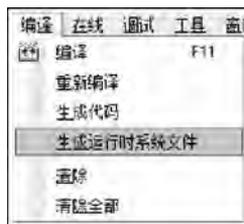


图 5-14 编译菜单

- (1) 编译：对当前的应用进行编译。
- (2) 重新编译：如果需要对已经编译过的应用再次编译，可以通过重新编译指令进行。
- (3) 生成代码：执行此命令后生成当前应用的机器代码，执行登录命令时，生成代码默认执行。

(4) 清除：删除当前应用的编译信息，如果再次登录设备时需要重新生成编译信息。

(5) 清除全部：删除工程中所有的编译信息。执行编译命令后可以看到，添加到任务里面的“PLC_PRG”显示为蓝色，没有添加到任务中的则显示为灰色，如图 5-15 所示。编译指令不会对灰色的 POU 进行语法检查，因为该程序单元没有处于活动状态，编译指令只对处于活动状态的 POU 进行语法检查。如果在编译的过程中发现需要运行的程序单元显示为灰色，可以检查该程序单元是否已经被成功地添加到了所需要运行的任务当中。



图 5-15 编译信息

编译命令执行完成之后可以在消息栏看到编译生成的信息，其中可以看到编译的程序是否有错误或者警告，以及错误和警告的数量。如果有错误和警告产生可以通过消息窗口进行查看和查找，根据提示信息对程序进行修改。

登录下载。登录使应用程序与目标设备建立起连接，并进入在线状态。能正确登录的前提条件是要正确配置设备的通信设置并且应用程序必须是无编译错误的。对于以当前活动应用登录，生成的代码必须没有错误并且设备通信设置必须配置正确。登录后，系统会自动选择程序下载。下载命令在线模式下有效，包括对当前应用程序的编译和生成目标代码两部分。除了语法检查（编译处理）外，还生成应用目标代码并装载到 PLC。

(1) 在线修改后登录：选择此选项后，项目的更改部分被装载到控制器中。使用“在线修改后登录”操作，可以防止控制器进入 STOP 状态。

(2) 登录并下载：选择“登录并下载”后，将整个项目重新装载到控制器中。其与“在线修改后登录”最大的区别，是当完成下载后，控制器会停留在 STOP 模式，等待用户发送 RUN 指令，或重启控制器程序才会运行。

(3) 没有变化后的登录：登录时，不更改上次装载到控制器中的程序。当完成下载后，需要将程序每次启动时运行，还需要单击“创建启动应用”，已编译的项目在控制器上以这种方式创建引导，即控制器再启动后，可以自动装入项目程序运行。引导项目的存储方式取决于目标系统。

5.2.3 CODESYS 基础编程及应用

通过对 CODESYS 编程软件中变量、编程指令、编程语言、程序编译下载的深入学习，掌握 CODESYS 基础编程方式方法并完成简单工程应用实例。

1. 了解全局变量和局部变量区别

变量的范围确定其在哪个程序组织单元中是可用的，范围可能是全局或局部。每个变量的范围由它被声明的位置和声明所使用的变量关键字所定义。

全局变量：在程序组织单元之外定义的变量称为外部变量，外部变量是全局变量。全局变量可以为本文件中其他程序组织单元所共用。全部程序可共享同一数据，它甚至能与其他网络进行数据交换。其原理示意图如图 5-16 所示。图中，bIn1 能同时为程序 A 和程序 B 共用。

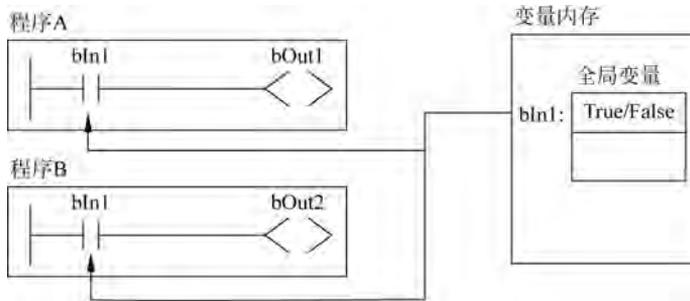


图 5-16 全局变量原理示意图

一个系统中不能有相同名称的两个全局变量。所有的全局变量都在全局变量列表中进行声明，全局变量提供了两个不同程序和功能块之间非常灵活地交换数据的方法。全局变量的关键字如下：

```
VAR_GLOBAL
<全局变量声明>
END_VAR
```

可以通过添加全局变量列表实现全局变量的添加。鼠标选中 Application, 右键选择“添加对象”→“全局变量列表”，系统则会自动弹出全局变量列表，用户只需输入列表名称，单击“确认”即可，具体步骤请参考图 5-17。



(a) 添加全局变量列表

(b) 输入列表名称

图 5-17 全局变量列表添加

局部变量：在一个程序组织单元(POU)内定义的变量都为内部变量，它只在该程序组织单元内有效，这些变量称为“局部变量”，其结构原理图如图 5-18 所示。

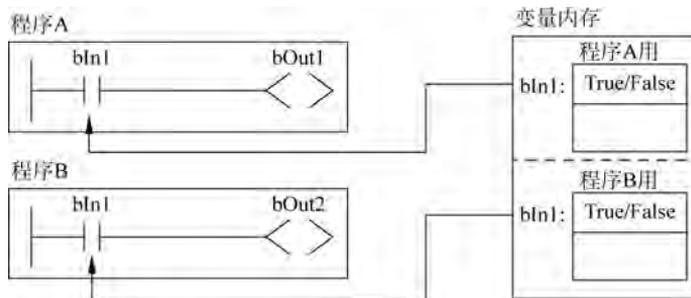


图 5-18 局部变量结构原理图

使用局部变量后，在执行多个独立的程序时，编程时无须理会其他独立程序中的同名变量，它们之间没有映射关系，互不影响。局部变量的关键字如下：

```
VAR
<局部变量声明>
END_VAR
```

2. CODESYS 相关编程指令

基础编程指令：

(1) 位逻辑指令。位逻辑指令处理布尔值“1”和“0”的逻辑变化。CODESYS 提供的位逻辑指令包括基本的逻辑运算、置位/复位优先触发器及上升/下降沿检测指令，如表 5-1 所示。

表 5-1 位逻辑指令的图形化与文本化指令表

位逻辑指令	图形化语言	文本化语言	说明
位逻辑指令		AND	与
		OR	或
		NOT	非
		XOR	异或
		SR	置位优先触发器
		RS	复位优先触发器
		R_TRIG	上升沿触发
		F_TRIG	下降沿触发

(2) 基本逻辑指令。基本位逻辑指令包括有“与”“或”“非”“异或”。在 CODESYS 中，从功能上分可以分为按位逻辑运算及布尔逻辑运算。按位逻辑运算：对两个整型数据的相应位逐一进行布尔逻辑运算，并返回兼容的整数结果。布尔逻辑运算：对两个布尔类型数据执行逻辑运算。

① 按位“与”AND。功能：按位“与”运算指令是比较两个整数的相应位。当两个数是“11”时，返回相应的结果位是“1”；当两个数是“00”或者其中一位是“0”时，则返回相应的结果位是“0”。

② 布尔“与”AND。功能：布尔“与”运算用于计算两个布尔表达式的“与”结果。当两个布尔表达式都为真时，则返回为真；其中只要有一个为假时，则返回为假。

③ 按位“或”OR。功能：按位“或”运算指令是比较两个整数的相应位。当两个数的对应位有一个是“1”或者是“11”时，返回相应的结果位为“1”；当两个整数的相应位都是“0”时，则返回相应的结果为“0”。

④ 布尔“或”OR。功能：布尔“或”运算指令用于计算两个布尔表达式的“或”结果。当两个布尔表达式中有一个表达式为真时，则结果为真；当两个布尔表达式都是假时，则结果为假。

⑤ 按位“非”NOT。功能：对逻辑串进行取反，将当前的值由“0”变“1”，或由“1”变“0”。按位“非”运算指令是将变量或常量逐一取非。

⑥ 布尔“非”NOT。功能：布尔“非”运算指令用于计算单个布尔表达式的结果。当输入为真时，结果为假；当输入为假时，结果为真。

(3) 置位优先与复位优先触发器指令。在继电器系统中，一个继电器的若干对触点是同时动作的。在 PLC 中，指令是一条一条执行的，指令的执行是有先后次序的，没有“同时”执行的指令。

所以线圈格式的置位、复位指令有优先级。SR 触发器与 RS 触发器的置位输入和复位输入在同一条指令里，置位和复位输入谁在指令输入端的下面谁后执行。SR 触发器为“置位优先”型触发器，当置位信号(SET1)和复位信号(RESET)同时为“1”时，触发器最终为置位状态；RS 触发器为“复位优先”型触发器，当置位信号(SET)和复位信号(RESET1)同时为“1”时，触发器最终为复位状态。置位优先 SR 与 RS 复位优先触发器指令参数如表 5-2 所示，指令表详见表 5-3。

表 5-2 置位优先 SR 与 RS 复位优先触发器指令

功能名	FB	ST	说明
置位优先触发器 SR 与复位优先触发器 RS		SR	置位优先触发器
		RS	复位优先触发器

表 5-3 置位优先 SR 与复位优先 RS 触发器指令参数

名称	定义	数据类型	说明
SET1	输入变量	BOOL	置位优先命令
SET	输入变量	BOOL	置位命令
RESET1	输入变量	BOOL	复位优先命令
RESET	输入变量	BOOL	复位命令
Q1	输出变量	BOOL	输出

置位优先触发器 SR 功能：置位双稳态触发器，置位优先。

逻辑关系： $Q1 = (\text{NOT RESET AND } Q1) \text{ OR SET1}$ ，其中 SET1 为置位信号，RESET 为复位信号。当 SET1 为“1”时，不论 RESET 是否为“1”，Q1 输出都为“1”；当 SET1 为“0”

时,如果 Q1 输出为“1”,一旦 RESET 为“1”,Q1 输出立刻复位为“0”。如果 Q1 输出为“0”,不论 RESET 为“1”或者“0”,Q1 输出保持为“0”。其时序图如图 5-19 所示。

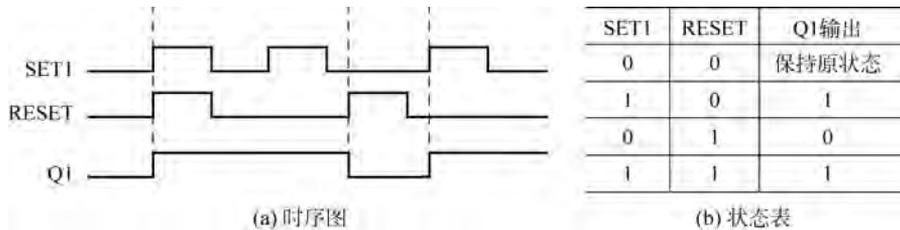


图 5-19 SR 置位优先触发器时序

复位优先触发器 RS 功能: 复位双稳态触发器,复位优先。逻辑关系: $Q1 = \text{NOT RESET1 AND } (Q1 \text{ OR SET})$,其中 SET 为置位信号,RESET1 为复位信号。当 RESET1 为“1”时,不论 SET 是否为“1”,Q1 输出都为“0”;当 RESET1 为“0”时,如果 Q1 输出为“0”,一旦 SET 为“1”,Q1 输出立刻置位为“1”。如果 Q1 输出为“1”,不论 SET 为“1”或者“0”,Q1 输出保持为“1”。其时序图如图 5-20 所示。

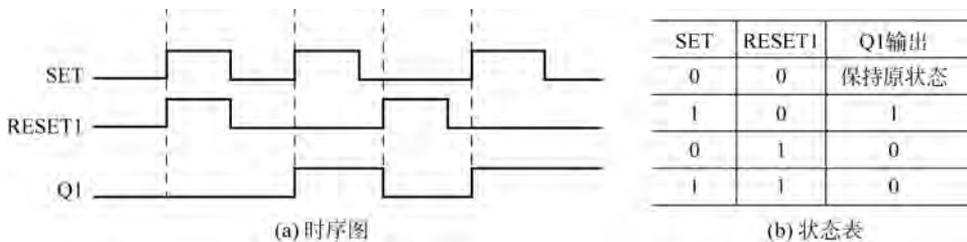


图 5-20 RS 复位优先触发器时序

(4) 边沿检测指令。边沿检测指令用来检测 BOOL 信号的上升沿(信号由 0→1)和下降沿(信号由 1→0)的变化,如图 5-21 所示。在每个扫描周期中把信号状态和它在前一个扫描周期的状态进行比较,若不同则表明有一个跳变沿。因此,前一个周期里的信号状态必须被存储,以便能和新的信号状态相比较。

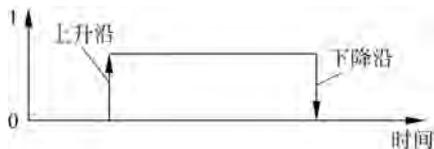


图 5-21 边沿信号

上升沿检测 R_TRIG 功能: 用于检测上升沿。当 CLK 从“0”变为“1”时,该上升沿检测器开始启动,Q 输出先由“1”然后输出变为“0”,持续一个 PLC 运算周期;如果 CLK 持续保持为“1”或者“0”,Q 输出一直保持为“0”。采集 bInput 信号的上升沿,程序如图 5-22 所示。

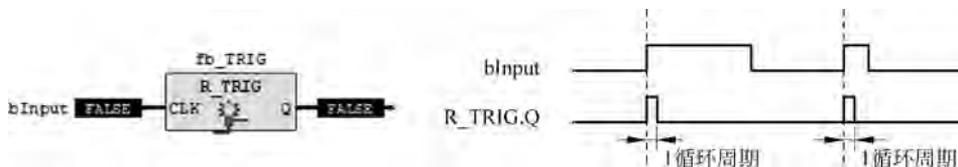


图 5-22 上升沿触发程序及时序

下降沿检测 F_TRIG 功能：用于检测下降沿。当 CLK 从“1”变为“0”时，该下降沿检测器开始启动，Q 输出先由“1”然后输出变为“0”，持续一个 PLC 运算周期；如果 CLK 持续保持为“1”或者“0”，Q 输出一直保持为“0”。采集 bInput 信号的下降沿，当 bInput 由 True 变为 False 时，功能块 F_TRIG.Q 会根据下降沿的触发事件给出相应输出，输出时间维持在一个周期。程序如图 5-23 所示。

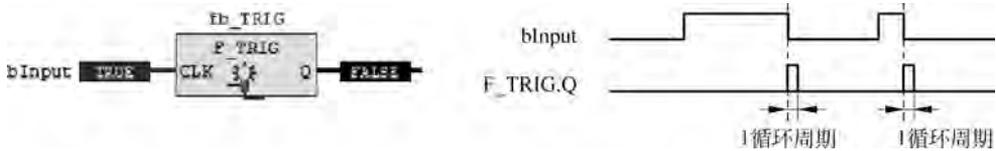


图 5-23 下降沿触发程序及时序

3. 梯形图编程语言

梯形图是国内应用最为广泛的编程语言，是传统 PLC 使用得最多的图形编程语言，也被称为 PLC 的第一编程语言。根据梯形图中各触点的状态和逻辑关系，求出与图中各线圈对应的编程元件的状态，称为梯形图的逻辑解算。梯形图中的某些编程元件沿用了继电器这一名称，如线圈、触点等，但是它们不是真实的物理继电器，而是一些存储单元（软继电器），每一软继电器与 PLC 存储器中映像寄存器的一个存储单元相对应。该存储单元如果为 TRUE 状态，则表示梯形图中对应软继电器的线圈“通电”，其常开触点接通，常闭触点断开，称这种状态是该软继电器的 TRUE 或 ON 状态。如果该存储单元为 FALSE 状态，对应软继电器的线圈和触点的状态与上述的相反，称该软继电器为 FALSE 或 OFF 状态。使用中也将这些“软继电器”称为编程元件。

梯形图程序执行顺序：梯形图的执行过程是按照从左至右、从上到下的顺序进行，如图 5-24 所示。

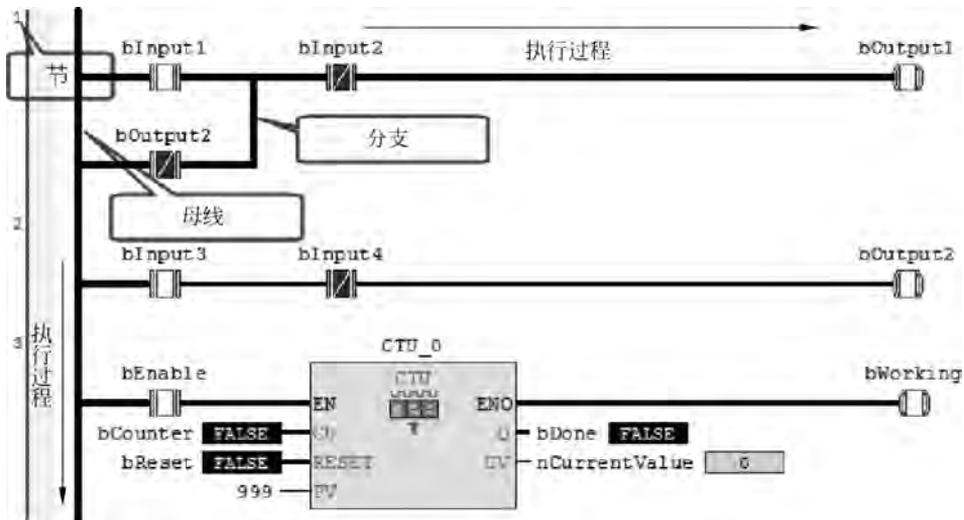


图 5-24 梯形图执行过程

执行过程包括：

- (1) 母线：梯形图采用网络结构，一个梯形图的网络以左母线为界。在分析梯形图的

逻辑关系时,为了借用继电器电路图的分析方法,可以想象左右两侧母线(左母线和右母线)之间有一个左正右负的直流电源电压,母线之间有“能流”从左向右流动。右母线不显示。

(2) 节:节是梯形图网络结构中最小单位,从输入条件开始,到一个线圈的有关逻辑的网络称为一个节。在编辑器中,节垂直排列。在 CODESYS 中,每个节通过左侧的一系列节号标示,包含输入指令和输出指令、逻辑式、算术表达式、程序、功能或功能块调用指令、跳转或返回指令所构成。要插入一个节,可以使用命令插入节或从工具箱拖动它。一个节所包含的元素都可以通过在编辑器中拖放来进行复制或移动。梯形图执行时,从标号最小的节开始执行,从左到右确定各元素的状态,并确定其右侧连接元素的状态,逐个向右执行,操作执行的结果由执行控制元素输出,然后进行下一节的执行过程。

(3) 能流:加粗线即为能流,可以理解为一个假想的“概念电流”或“能流”(PowerFlow)从左向右流动,这一方向与执行用户程序时的逻辑运算的顺序是一致的。能流只能从左向右流动。利用能流这一概念,可以帮助我们更好地理解和分析梯形图。

(4) 分支:当梯形图中有分支出现时,同样依据从上到下、从左至右的执行顺序分析各图形元素的状态,对垂直连接元素根据上述有关规定确定其右侧连接元素的状态,从而逐个从左向右、从上向下执行求值过程。在梯形图中,没有反馈路径的求值不是很明确。其所有外部输入值与这些有关的触点必须在每个梯级以前被求值。

CODESYS 的主要的图形符号包括:

触点类:常开触点、常闭触点、正转换读出触点、负转换触点。

线圈类:一般线圈、取反线圈、置位(锁存)线圈、复位去锁线圈、保持线圈、置位保持线圈、复位保持线圈、正转换读出线圈、负转换读出线圈。

功能和功能块:包括标准的功能和功能块以及用户自己定义的功能块,如图 5-25 所示。

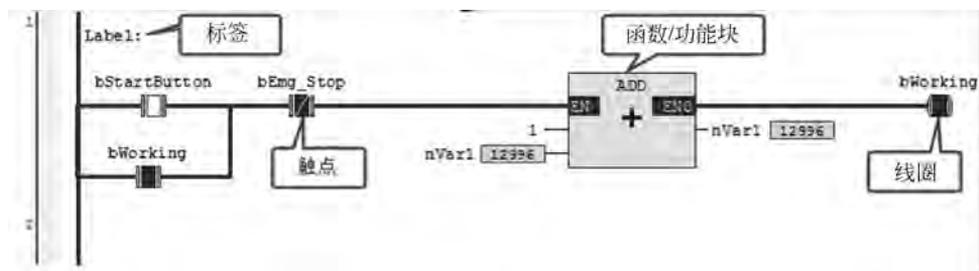


图 5-25 梯形图编辑器

① 电源轨线:梯形图电源轨线(Power Rail)的图形元素亦称为母线。其图形表示是位于梯形图左侧,也可称其为左电源母线。图 5-26 为左母线的图形表示。

② 连接元素:在梯形图中,各图形符号用连接元素连接,连接元素的图形符号有水平线和垂直线,它是构成梯形图的最基本元素。图 5-27 是水平和垂直连接元素的图形表示。

③ 触点:触点是梯形图的图形元素。梯形图的触点沿用了电气逻辑图的触点术语,用于表示布尔型变量的状态变化。触点是向其右侧水平连接元素传递一个状态的梯形图元素。触点可以分为常开触点和常闭触点。常开触点指在正常工况下,触点断开,其状态为 FALSE。常闭触点指在正常工况下,触点闭合,其状态为 True,如表 5-4 所示。

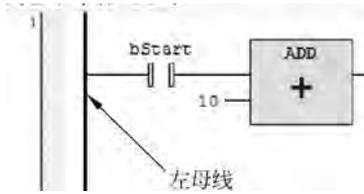


图 5-26 左母线图形表示



图 5-27 水平和垂直连接元素的图形

表 5-4 触点元素的图形符号与说明

类型	图形符号	说明
常开触点		如果该触点对应当前布尔变量值为 True 时,则该触点吸合,如触点左侧连接元素的状态为 True 时,则状态 True 传递至该触点右侧,使右侧连接元素的状态为 True;反之,当布尔变量值为 False 时,右侧连接元素状态 False
常闭触点		如果该触点对应当前布尔变量值为 False 时,则该闭触点处于吸合状态,如触点左侧连接元素的状态为 True 时,则状态 True 被传递至该触点右侧,使右侧连接元素的状态为 True;反之,当布尔变量值为 True 时,触点断开,则右侧连接元素状态为 False
插入右触点		可以进行多个触点的串联,在右侧插入触点。多个串联的触点都为吸合状态时,最后一个触点才能传输 True
插入并联下常开触点		可以进行多个触点的并联,在触点下侧并联插入常开触点。两个并联触点中只需一个触点为 True,则平行线传输 True
插入并联下常闭触点		可以进行多个触点的并联,在触点下侧并联插入常闭触点。常闭触点默认为吸合状态,如该触点对应当前布尔变量值为 False,左侧连接元素的状态为 True 时,则该并联触点右侧传输 True
插入并联上常开触点		可以进行多个触点的并联,在触点上侧并联插入常开触点。两个并联触点中只需一个触点为 True,则平行线传输 True

④ 线圈：线圈是梯形图的图形元素。梯形图中的线圈沿用了电气逻辑图的线圈术语，用于表示布尔型变量的状态变化。根据线圈的不同特性，可以分为瞬时线圈和锁存线圈，锁存线圈分为置位线圈和复位线圈。如表 5-5 所示。

表 5-5 线圈元素的图形符号与说明

类型	图形符号	说明
线圈		左侧连接元素的状态被传递到有关的布尔型变量和右侧连接元素,如果线圈左侧连接元素的状态为 True,则线圈的布尔变量为 True;反之,线圈为 False
置位线圈		线圈中有一个 S。当左侧连接元素的状态为 True 时,该线圈的布尔型变量为置位并且保持,直到有复位线圈的复位
复位线圈		线圈中有一个 R。当左侧连接元素的状态为 True 时,该线圈的布尔型变量为复位并且保持,直到有置位线圈的置位

⑤ 函数及功能块调用：与节点和线圈一起，用户也可以插入功能块和程序。在网络中，它们必须有带布尔值的一个输入和一个输出，并可在相同位置上像接点那样使用，也就是说在 LD 网络的左侧。

4. 对象控制器系统及 I/O 模块

这两部分是软 PLC 的核心,完成输入处理、程序执行、输出处理等工作。通常由 I/O 接口、通信接口、系统管理器、错误管理器、调试内核和编译器组成。

I/O 接口:可与任何 I/O 信号连接,包括本地 I/O 和远程 I/O,远程 I/O 主要通过现场总线 InterBus,ProfiBus,CAN 等实现。

通信接口:通过此接口使运行系统可以和开发系统或 HMI 按照各种协议进行通信,如下载 PLC 程序或进行数据交换。

系统管理器:处理不同任务和协调程序的执行。

错误管理器:检测和和处理程序执行期间发生的各种错误。

调试内核:提供多个调试函数,如强制变量、设置断点等。

编译器:通常开发系统将编写的 PLC 源程序编译为中间代码,然后运行系统的编译器将中间代码翻译为与硬件平台相关的机器码存入控制器。

搬运码垛机器人工作站系统总线配置采用 EtherCAT 技术,通过自动连接检测使设备部件的热插拔成为可能。EtherCAT 技术对线型、树状、星状或菊花链状拓扑结构都可以实现。设备的连接或断开由总线管理器管理,也可以由从站设备自动实现。若用一条线缆连接 EtherCAT 主站上另一个(标准的)以太网端口,就简单而经济地实现了网络冗余。EtherCAT 拥有多种机制,支持主站到从站、从站到从站以及主站到主站之间的通信。

① **主站配置:**首先在设备下添加 EtherCAT 主站,单击“设备”,右键选择“添加设备”,当弹出添加设备窗口后,选择 EtherCAT→“主站”→EtherCAT Master,单击“添加设备”。主站添加后,可以通过单击选中主站从而对其进行配置。在 EtherCAT“主站”选项卡中,配置主站、分布式时钟、冗余及从站相应设置,如图 5-28 所示。



图 5-28 添加 EtherCAT 主站

② 从站配置：安装 EtherCAT 从站设备描述文件。为了在设备目录中插入和配置 EtherCAT 设备，主站和从站必须使用硬件提供的设备描述文件，通过“设备库”对话框安装（标准的 CODESYS 设置自动地完成）。主站的设备描述文件（*.devdesc.xml）定义了可以插入的从站，从站为“xml”文件格式（文件类型：EtherCAT XML 设备描述配置文件）。

在“工具”菜单栏中通过“设备库”→“安装”选择“EtherCAT XML 设备描述配置文件（*.xml）”，找到该文件路径，选择“打开”进行安装。

添加从站设备通过选择“插入设备”，系统会自动弹出从站设备添加框，根据实际连接的从站进行添加。此外，也可以通过选中主站选项卡，单击鼠标，选择“扫描设备”进行实际从站自动扫描搜索，如图 5-29 所示。

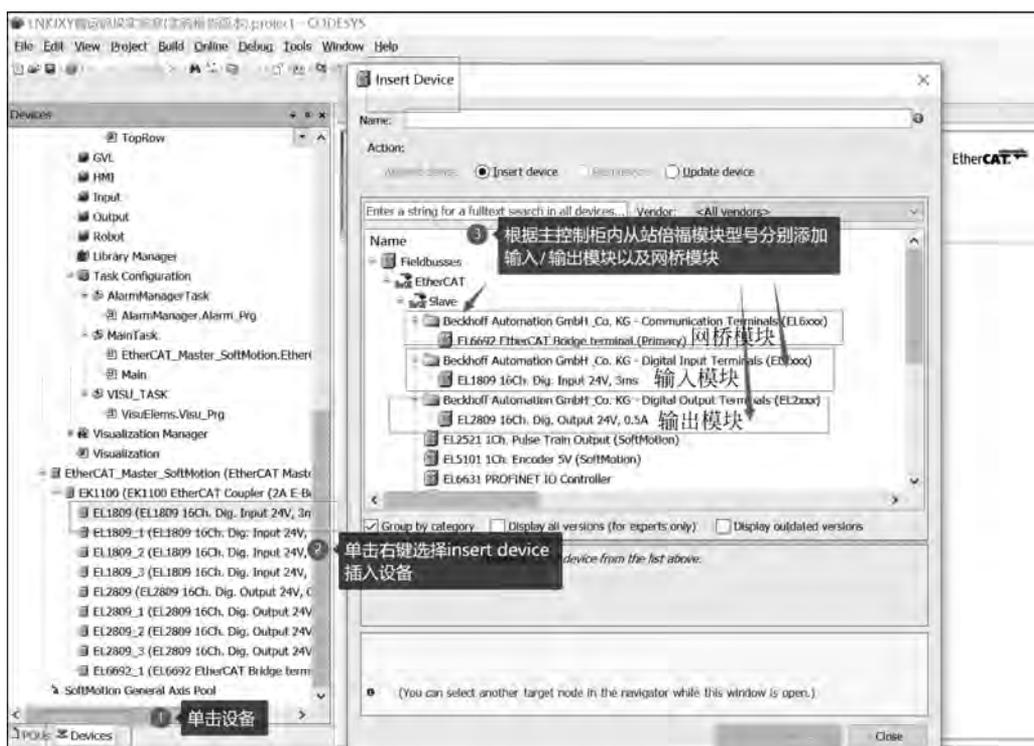


图 5-29 添加 EtherCAT 从站模块

③ EtherCAT I/O 匹配：按照图纸 I/O 接线图，将输入/输出点分别关联至已经组态好的 I/O 模块相应地址，具体步骤如图 5-30 和图 5-31 所示。

5. HMI 人机界面可视化编辑

HMI 人机界面可视化编辑的学习是了解 CODESYS 软 PLC 上位监控软件界面创建及编辑的方法，并绘制码垛系统 HMI 人机界面。

下面以创建“机器人 Robot”界面为例进行说明如何创建 HMI 人机界面。

(1) 右键单击 Mainview，选择“添加对象”，单击视图，如图 5-32(a)所示。

(2) 如图 5-32(b)所示，添加视图，输入人机界面名称“Robot”，确认打开，则完成机器人的可视化编辑器。

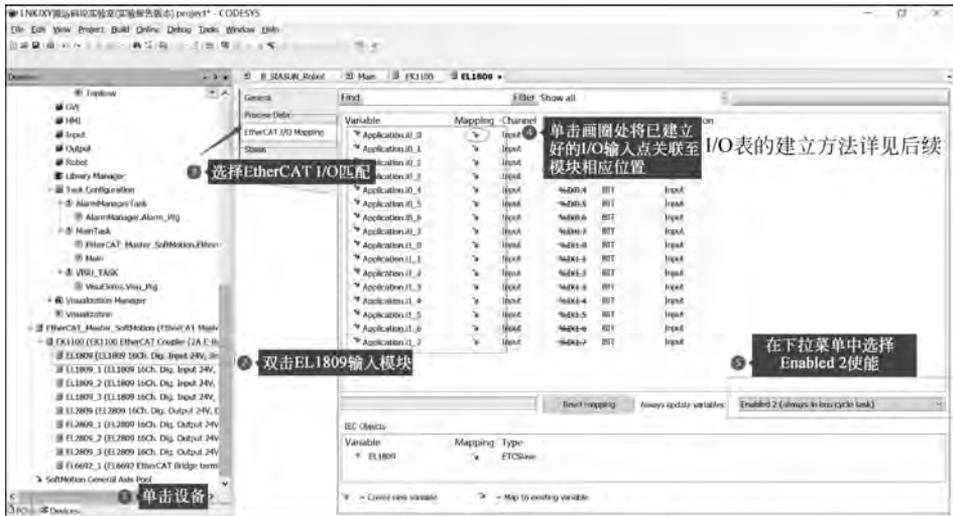


图 5-30 EtherCAT 输入 I/O 匹配

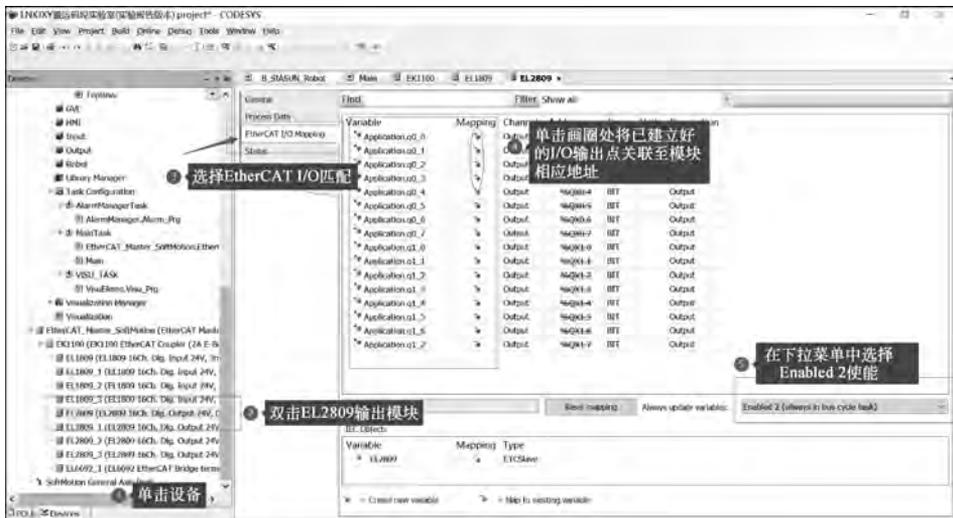


图 5-31 EtherCAT 输出 I/O 匹配



图 5-32 创建“机器人 Robot”界面

接下来绘制 Robot 人机界面可视化界面：

(1) 如图 5-33 所示,单击工具箱并找到指示灯元件图库,拖曳指示灯至 Robot 界面可视化编辑器窗口。



图 5-33 可视化编辑器指示灯工具

(2) 如图 5-34 所示,右击指示灯图标,选择指示灯背景颜色为蓝色,并关联机器人执行模式 Execute_Mode 变量至指示灯属性。

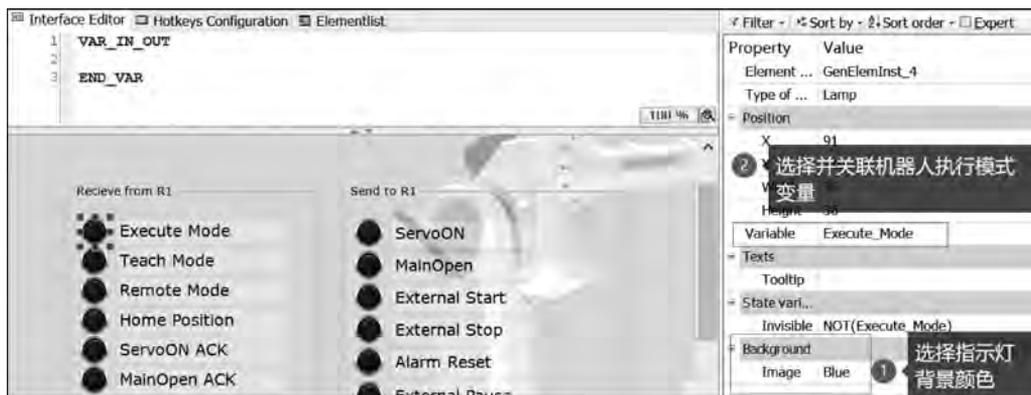


图 5-34 关联指示灯变量

- (3) 按照以上步骤重复创建机器人监控变量指示灯,并关联机器人变量至相应指示灯。
 (4) 如图 5-35 所示,单击工具箱并选择基础工具,拖曳矩形文本至可视化编辑器窗口。



图 5-35 可视化编辑器矩形文本工具

- (5) 按照步骤(4)重复创建其他机器人监控变量指示灯文本,如图 5-36 所示。

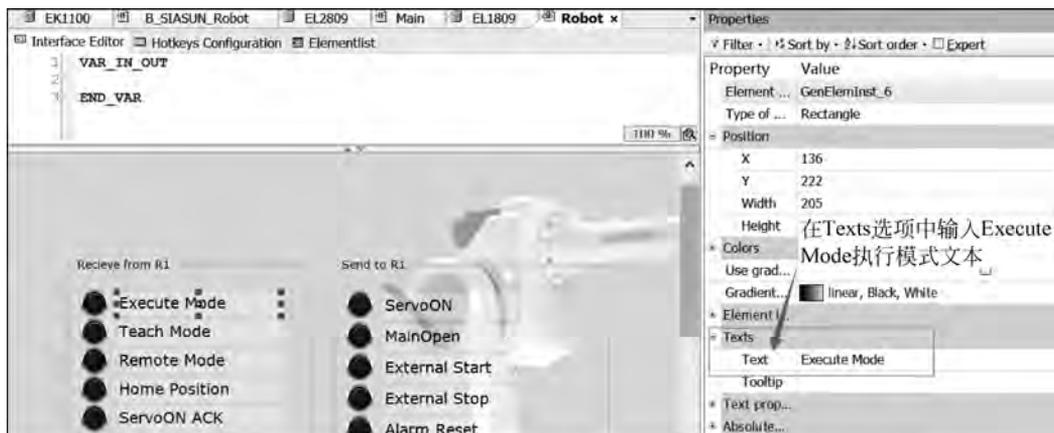


图 5-36 输入指示灯相关文本

(6) 按钮创建及关联方法在此不再赘述。

(7) 按照以上步骤完成图 5-37 所示 Robot 人机界面可视化编辑绘制。

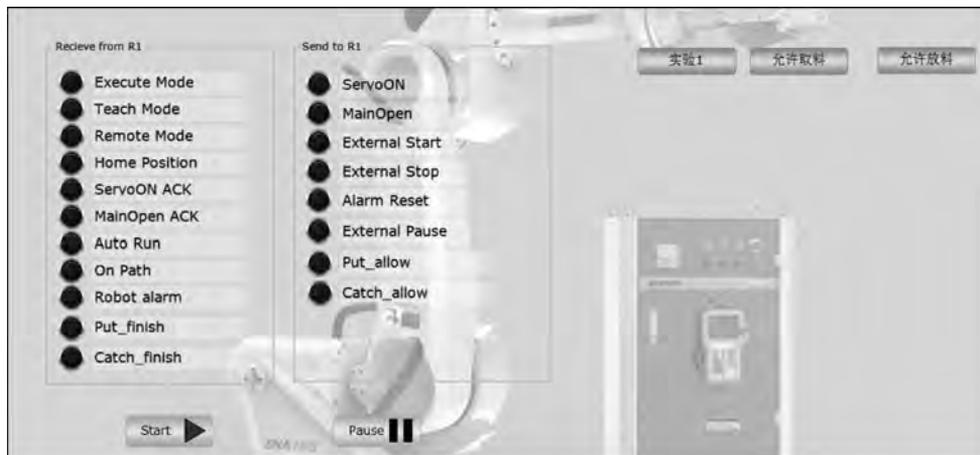


图 5-37 Robot 人机界面可视化编辑

绘制 transfer 输送线可视化界面方法如下：

在 Devices 栏双击 MainView 文件夹，选择 transfer 界面，在打开的空白界面中按照图 5-38 所示分别添加五条输送线变频器的正转控制按钮、反转控制按钮、变频器复位按钮、低速选择按钮、高速选择按钮，运行以及报警指示灯，并关联相关属性变量完成 transfer 可视化界面绘制。



图 5-38 transfer 界面绘制

绘制 Start 启动可视化界面的方法如下：

① 如图 5-39 所示，右击 Mainview，创建 Start 启动可视化界面；

② 如图 5-40 所示，单击 ToolBox，选择当前工程；拖曳工艺流程截图至 Start 启动界面可视化窗口；



图 5-39 新建 Start 可视化界面



图 5-40 添加工艺流程图片

③ 如图 5-41 所示,在工艺流程图背景基础上,按照码垛系统实验室实际布置的传感器及磁性开关位置,分别添加传感器及磁性开关可视化监控界面指示灯及矩形文本注释,更加直观地实现现场传感器设备的在线监控;具体添加方法及变量关联方法按照上文介绍方式逐一添加并完成 Start 启动界面的绘制。



图 5-41 Start 启动可视化界面

设置 Start 启动界面为开机启动画面。如图 5-42 所示,单击 Device 设备,选择目标 TargetVisu 界面设置,单击 Start Visualization,关联 Start 启动界面为开机画面。

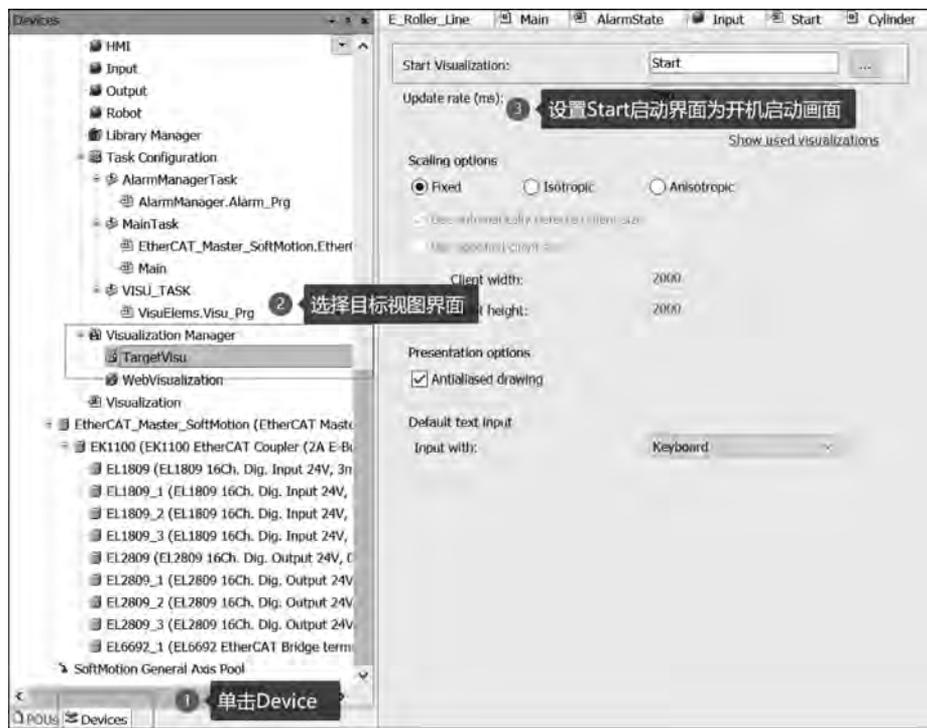


图 5-42 开机启动界面设置

可视化界面注释中英文切换: ①如图 5-43 所示,单击 POU; ②选择 Global 文本列表; ③在 ch 中文注释列分别输入注释的中文含义。

添加可视化界面背景图片: ①单击 Devices; ②选择图片库; ③输入图片 ID 名称; ④浏览并添加背景图片; ⑤添加完成后,Link type 列显示连接到文件,如图 5-44 所示。

浏览添加背景图片: ①浏览并选择相应图片; ②单击 OK,如图 5-45 所示。

综合控制方案软 PLC 控制器的硬件平台主要可以分为如下三部分:

(1) 基于嵌入式控制器的控制系统。嵌入式控制器是一种超小型计算机系统,一般没有显示器,软件平台是嵌入式操作系统(如 Windows CE、VxWorks 和 QNX 等)。软 PLC 的实时控制核被安装到嵌入式控制系统中,以保证软 PLC 的实时性,开发完的系统通过串口或以太网将转换后的二进制码写入对象控制器中。

(2) 基于工控机(IPC)或嵌入式控制器(EPC)的控制系统。该方案的软件平台可以采用 Windows 操作系统(Windows XP Embedded、Windows 7 等),通用 I/O 总线负责将远程采集的 I/O 信号传至控制器进行处理,软 PLC 可以充当开发系统的角色及对象控制器的角色。目前市场上越来越多的用户更倾向于直接使用面板型工控机进行控制的方案,这样的方案直接集成了 HMI,开发系统及对象控制器的功能,大大降低了成本。

(3) 基于传统硬 PLC 的控制系统。此方案中,PLC 开发系统一般在普通 PC 上运行,而传统硬 PLC 只是作为一个硬件平台,将软 PLC 的实时核安装在传统硬 PLC 中,将开发系统编写的系统程序下载到硬 PLC 中,其与控制系统图的区别是将图中的嵌入式控制器替换成传统硬 PLC。

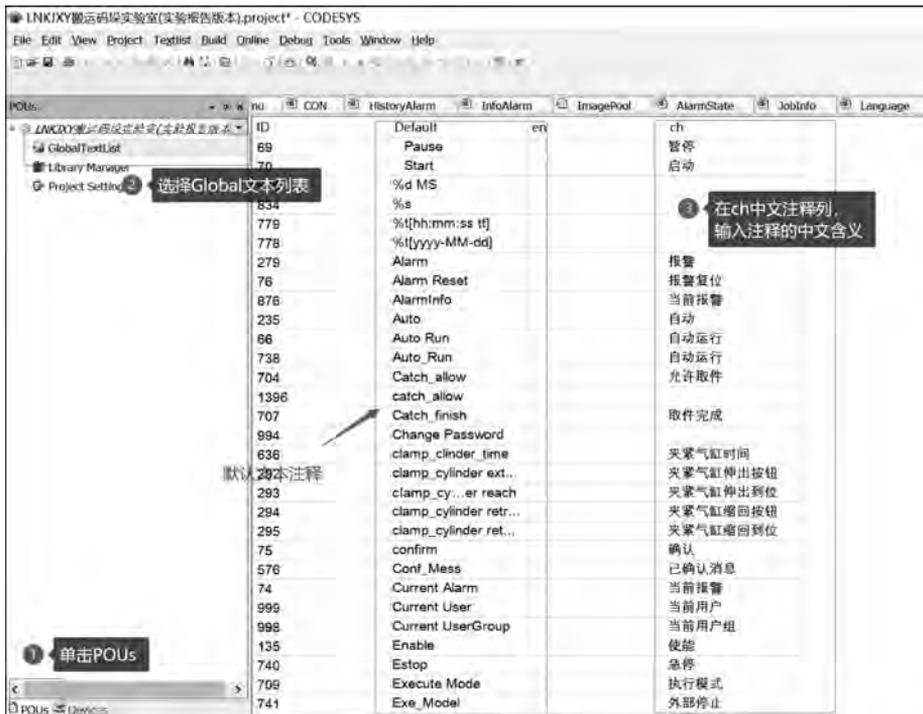


图 5-43 可视化界面中英文注释



图 5-44 背景图片添加



图 5-45 浏览添加背景图片

6. CODESYS 软 PLC 码垛工作站编程

了解码垛工作站编程思路及程序架构,熟悉 Main 主程序、FB 功能块、FC 程序块创建及编写方式方法,利用所学的 CODESYS 基础编程知识完成码垛工作站编程工作。学习新建 CODESYS 工程项目的步骤,熟悉码垛工作站系统 PLC 程序架构,掌握 CODESYS 软 PLC 的 FB 功能块形参变量的创建方法以及输送线、气缸、机器人等 FB 功能块程序的编程方法。

(1) 新建 CODESYS 工程项目。如图 5-46 所示,按照图片所示步骤,可以完成工程项目创建。

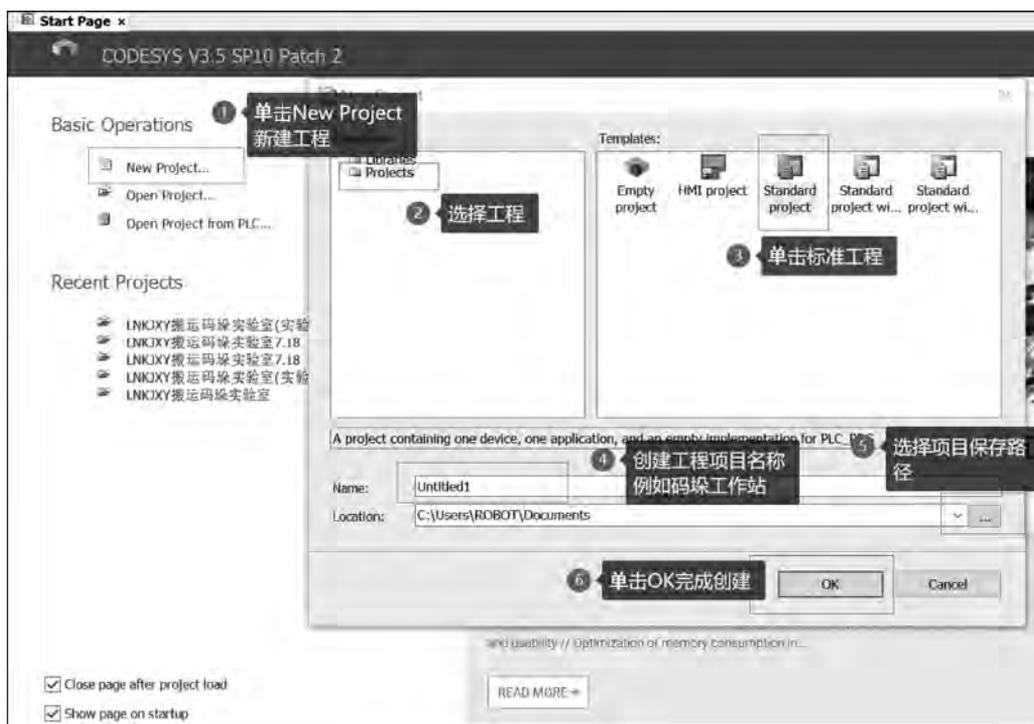


图 5-46 新建 CODESYS 工程项目步骤

(2) CODESYS 工程程序架构。如图 5-47(a)所示,Device 设备树涵盖了整个工程项目的所有设备信息,Program 文件夹包含 FB 功能块、Main 主程序等程序段,图 5-47(b)与电气图纸中倍福模块组态分布保持一致,模块所有信息与控制柜内各模块完全对应,正确的配置保证了组态的顺利完成。



图 5-47 CODESYS 工程程序架构

(3) 创建变量表。新建码垛工作站工程第一步需要根据电气图纸 PLC 输入/输出模块接线图,创建 GVL 全局变量表(图 5-48)、Input 和 Output 变量表(图 5-49),以及机器人通信 I/O 变量表(图 5-50)。

VAR_GLOBAL	system_auto_mode_A	BOOL	系统自动模式
VAR_GLOBAL	system_hand_mode_A	BOOL	系统手动运行
VAR_GLOBAL	system_Start_Button_A	BOOL	系统启动按钮
VAR_GLOBAL	system_pause_Button_A	BOOL	系统暂停按钮
VAR_GLOBAL	system_auto_run_A	BOOL	系统自动运行
VAR_GLOBAL	system_auto_pause_A	BOOL	系统自动停止
VAR_GLOBAL	system_alarm	BOOL	系统报警
VAR_GLOBAL	T	WORD	1200
VAR_GLOBAL	g_MainVisuSwitch	INT	主界面切换
VAR_GLOBAL	firstYL	BOOL	初始位有料
VAR_GLOBAL	lastYL	BOOL	末端位有料
VAR_GLOBAL	R_start	BOOL	机器人启动

图 5-48 GVL 全局变量表

创建 I/O 变量表有两种方式,可在右上角进行切换,在此不再赘述。

(4) 编写 FB 功能块。编写系统控制 FB 块程序如图 5-51 所示。编写系统控制功能块程序,完成启动、暂停、停止按钮及按钮灯,以及塔灯各状态相关逻辑。

编写新松机器人 FB 块程序:首先按照图 5-52 所示步骤创建新松机器人 FB 块形参变量表;然后用上面新建的机器人 FB 块形参变量按照图 5-53 所示步骤编写机器人 FB 块梯形图逻辑程序。

Line	VAR_GLOBAL	Line	VAR_GLOBAL
2	i0_0: BOOL; //启动按钮	2	q0_0: BOOL; //启动按钮灯
3	i0_1: BOOL; //暂停按钮	3	q0_1: BOOL; //暂停按钮灯
4	i0_2: BOOL; //复位按钮	4	q0_2: BOOL; //复位按钮灯
5	i0_3: BOOL; //急停按钮	5	q0_3: BOOL; //预留按钮灯
6	i0_4: BOOL; //预留按钮	6	q0_4: BOOL; //指示灯绿
7	i0_5: BOOL; //变频器5运行中	7	q0_5: BOOL; //指示灯黄
8	i0_6: BOOL; //变频器5故障	8	q0_6: BOOL; //指示灯红
9	i0_7: BOOL; //变频器4运行中	9	q0_7: BOOL; //蜂鸣器
10	i1_0: BOOL; //变频器4故障	10	q1_0: BOOL; //变频器5反转
11	i1_1: BOOL; //变频器3运行中	11	q1_1: BOOL; //变频器5正转
12	i1_2: BOOL; //变频器3故障	12	q1_2: BOOL; //变频器5速度高
13	i1_3: BOOL; //变频器2运行中	13	q1_3: BOOL; //变频器5速度低
14	i1_4: BOOL; //变频器2故障	14	q1_4: BOOL; //变频器5报警复位
15	i1_5: BOOL; //变频器1运行中	15	q1_5: BOOL; //变频器4正转
16	i1_6: BOOL; //变频器1故障	16	q1_6: BOOL; //变频器4反转
17	i2_0: BOOL; //线体1起始位传感器	17	q1_7: BOOL; //变频器4速度1
18	i2_1: BOOL; //线体1末端位传感器	18	q2_0: BOOL; //变频器4速度2
19	i2_3: BOOL; //线体2末端位传感器	19	q2_1: BOOL; //变频器4报警复位
20	i2_5: BOOL; //线体3移载完全进入	20	q2_2: BOOL; //变频器3正转
21	i2_6: BOOL; //线体3末端位传感器	21	q2_3: BOOL; //变频器3反转
22	i2_7: BOOL; //线体4加紧位传感器	22	q2_4: BOOL; //变频器3速度1
23	i3_1: BOOL; //线体4末端位传感器	23	q2_5: BOOL; //变频器3速度2
24	i3_2: BOOL; //线体2侧推气缸缩磁环	24	q2_6: BOOL; //变频器3报警复位
25	i3_3: BOOL; //线体2侧推气缸伸磁环	25	q2_7: BOOL; //变频器2反转
26	i3_4: BOOL; //移载机气缸伸磁环	26	q3_0: BOOL; //变频器2正转
27	i3_5: BOOL; //移载机气缸缩磁环	27	q3_1: BOOL; //变频器2速度1
28	i3_6: BOOL; //线体4阻挡气缸伸磁环	28	q3_2: BOOL; //变频器2速度2
29	i3_7: BOOL; //线体4阻挡气缸缩磁环	29	q3_3: BOOL; //变频器2报警复位
30	i4_0: BOOL; //线体4放行气缸伸磁环	30	q3_4: BOOL; //变频器1反转
31	i4_1: BOOL; //线体4放行气缸缩磁环	31	q3_5: BOOL; //变频器1正转
32	i4_2: BOOL; //线体4定位气缸伸磁环	32	q3_6: BOOL; //变频器1速度1
33	i4_3: BOOL; //线体4定位气缸缩磁环	33	q3_7: BOOL; //变频器1速度2
34	// 线体2始端传感器	34	q4_0: BOOL; //变频器1报警复位
35	i4_4: BOOL;	35	q4_1: BOOL; //阻挡伸电磁阀
36	// 挡停工位阻挡气缸下降到位	36	q4_2: BOOL; //阻挡缩电磁阀
37	i4_5: BOOL;	37	q4_3: BOOL; //阻挡伸电磁阀1
38	// 挡停工位有料传感器	38	q4_4: BOOL; //阻挡缩电磁阀1
39	i4_6: BOOL;	39	q4_5: BOOL; //定位伸电磁阀
40	// 挡停工位阻挡气缸上升到位	40	q4_6: BOOL; //定位缩电磁阀
41	i4_7: BOOL;	41	q4_7: BOOL; //侧推伸电磁阀
42	// 线体1始端传感器2	42	q5_0: BOOL; //侧推缩电磁阀
43	i5_1: BOOL;	43	q5_1: BOOL; //移载机伸电磁阀
44	END_VAR	44	q5_2: BOOL; //移载机缩电磁阀
45		45	// 挡停工位阻挡气缸上升
		46	q5_3: BOOL;

图 5-49 Input 输入变量表与 Output 输出变量表

Scope	Name	Address	Data type	Initialization	Comment	Attributes
1	VAR_GLOBAL	Execute_Mode	BOOL		机器人执行模式	
2	VAR_GLOBAL	Teach_Mode	BOOL		机器人示教模式	
3	VAR_GLOBAL	Remote_Mode	BOOL		机器人远程模式	
4	VAR_GLOBAL	Home_Position	BOOL		机器人原点位置	
5	VAR_GLOBAL	ServoON_ACK	BOOL		机器人伺服上电确认	
6	VAR_GLOBAL	MainOpen_ACK	BOOL		机器人主作业打开确认	
7	VAR_GLOBAL	Auto_Run	BOOL		机器人运行中	
8	VAR_GLOBAL	On_Path	BOOL		机器人在路径上	
9	VAR_GLOBAL	Alarm_ing	BOOL		机器人报警中	
10	VAR_GLOBAL	estop	BOOL		机器人急停中	
11	VAR_GLOBAL	pause_ing	BOOL		机器人暂停中	
12	VAR_GLOBAL	catch_finish	BOOL		取料完成	
13	VAR_GLOBAL	put_finish	BOOL		放料完成	
14	VAR_GLOBAL	ServoON	BOOL		机器人外部问题上电	
15	VAR_GLOBAL	MainOpen	BOOL		机器人主作业打开	
16	VAR_GLOBAL	external_Start	BOOL		机器人外部启动	
17	VAR_GLOBAL	external_Pause	BOOL		机器人外部暂停	
18	VAR_GLOBAL	external_Stop	BOOL		机器人外部急停	
19	VAR_GLOBAL	Alarm_Reset	BOOL		机器人报警复位	
20	VAR_GLOBAL	catch_allow	BOOL		允许取料	
21	VAR_GLOBAL	put_allow	BOOL		允许放料	
22	VAR_GLOBAL	catch_allow_Test	BOOL		允许取料1(Test)	
23	VAR_GLOBAL	put_allow_Test	BOOL		允许放料1(Test)	

图 5-50 机器人通信 I/O 变量表

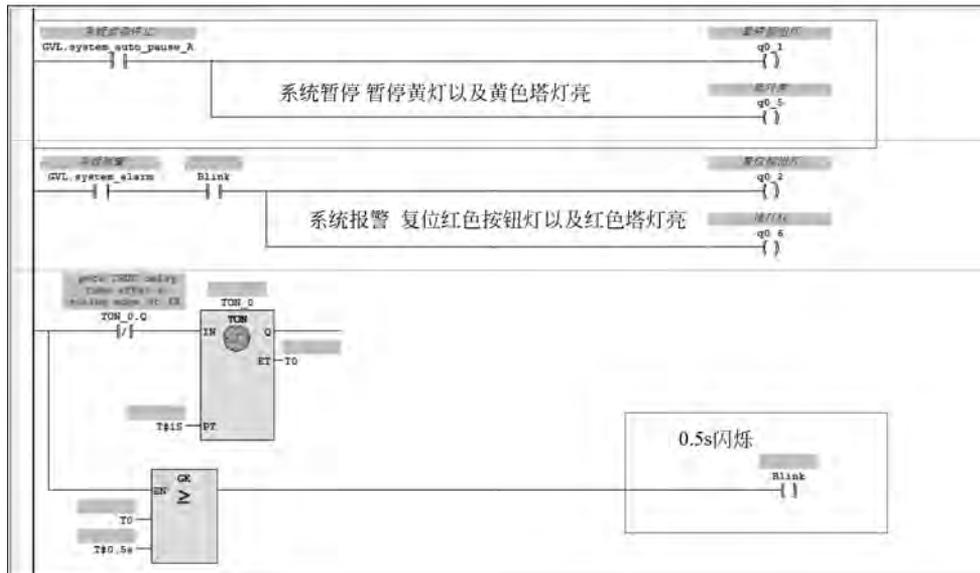


图 5-51 系统控制功能块逻辑程序

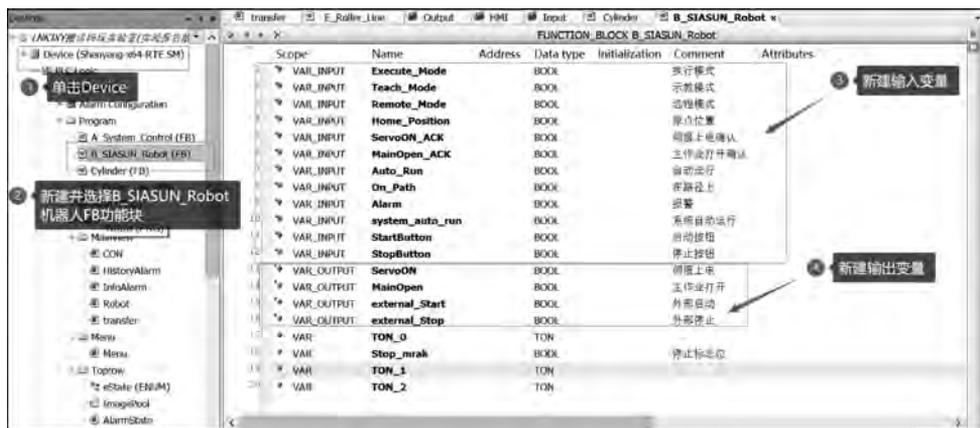


图 5-52 创建新松机器人 FB 功能块变量表

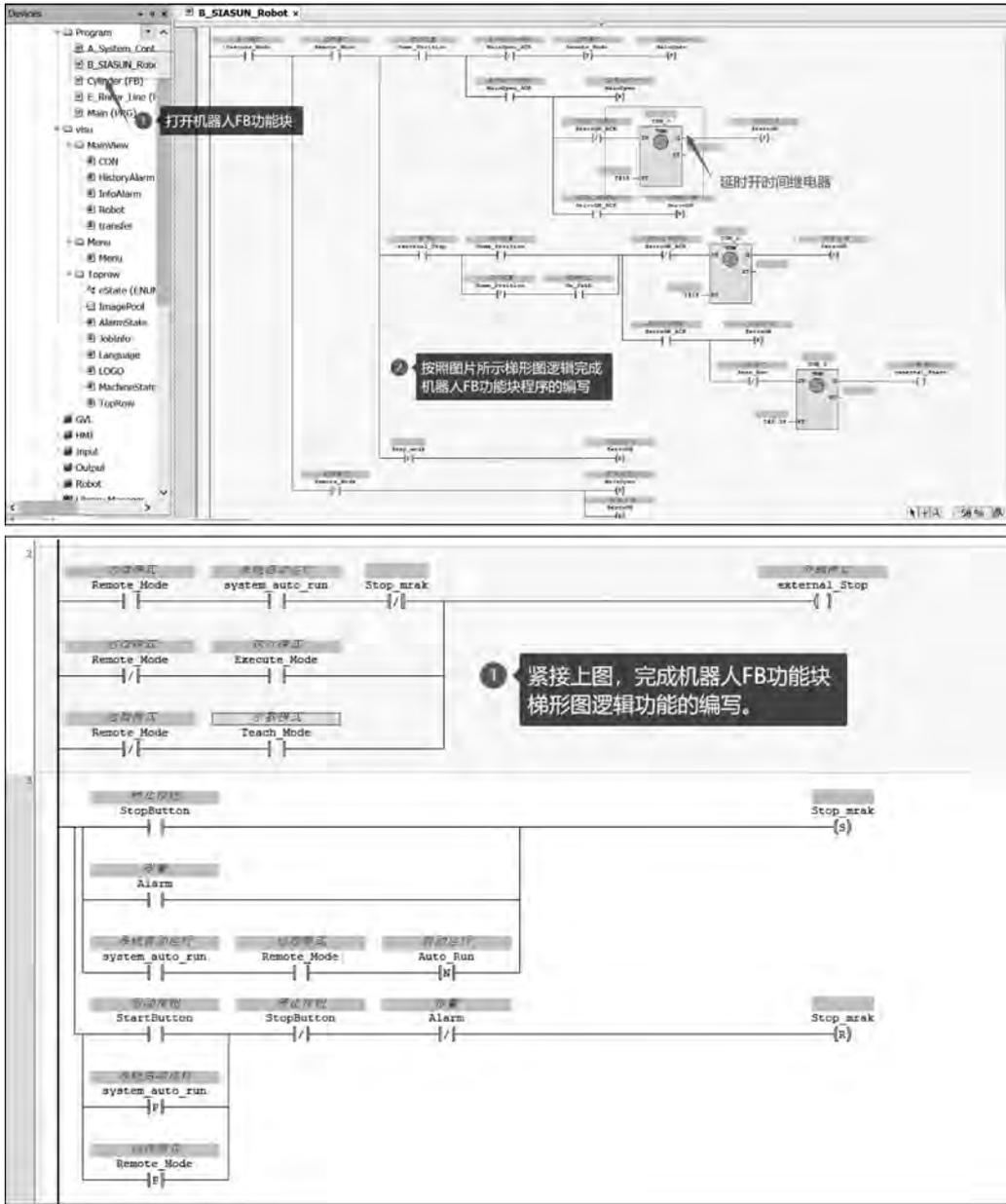


图 5-53 新松机器人 FB 功能块程序

编写气缸 FB 功能块程序：首先按照图 5-54 所示步骤创建气缸 FB 功能块形参变量表；然后用上面新建的机器人 FB 块形参变量按照图 5-55 所示步骤编写气缸 FB 功能块梯形图逻辑程序。

编写线体 FB 功能块程序：首先按照图 5-56 所示步骤创建线体 FB 功能块形参变量表；然后用线体 FB 功能块形参变量按照图 5-57 所示步骤编写线体 FB 功能块梯形图逻辑程序。

编写 Main 主程序段逻辑。码垛工作站系统自动运行需要控制的设备分别有：

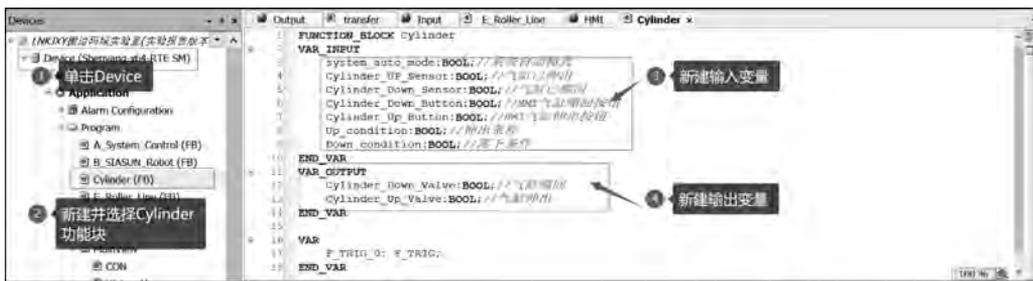


图 5-54 气缸 FB 功能块形参变量表

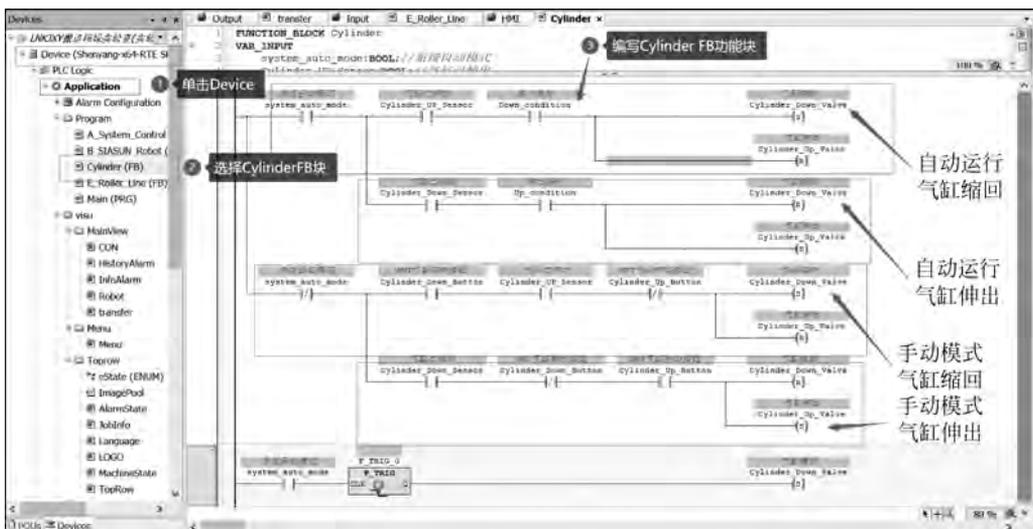


图 5-55 气缸 FB 功能块程序

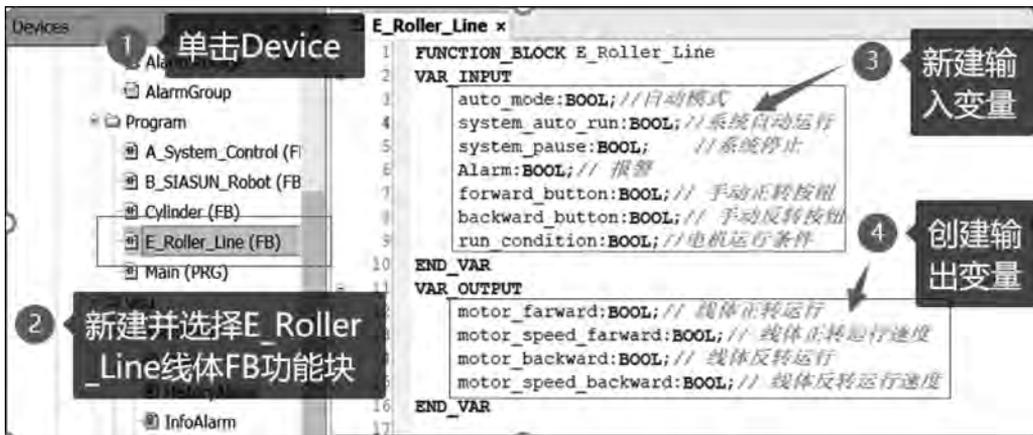


图 5-56 线体 FB 功能块变量表



图 5-57 线体 FB 功能块逻辑程序

① 新松机器人一台。CODESYS 软 PLC 需要在自动运行过程中,调度机器人完成码垛及拆垛作业。

② 输送线体变频器五台。在传感器、气缸或者衔接的其他变频器达到固定的某种状态时启停变频器,在此仅以输送线 1 为例介绍相关程序编写方法,其他输送线体程序请参照 CODESYS 最终程序或者实验报告。

③ 气缸电磁阀若干。在自动运行过程中,PLC 需要在传感器或者变频器反馈信号达到某种固定的状态时伸出或者缩回气缸,在此仅以侧推气缸 1 为例介绍相关程序编写方法,其他气缸程序请参照 CODESYS 最终程序或者实验报告。

输送线体主程序。如图 5-58 所示,E_Roller_Line 中的 run condition 引脚关联输送线 1 自动运行的触发条件,系统自动运行时输送线 1 即自动运行,当输送线 2 始端传感器被触发,将标志位 M_mark1 置位,输送线 1 停止,当工件到达侧推气缸工位并被推出至侧推气缸伸出到位置,磁性开关信号将 M_mark1 标志位复位,输送线 1 继续运行。

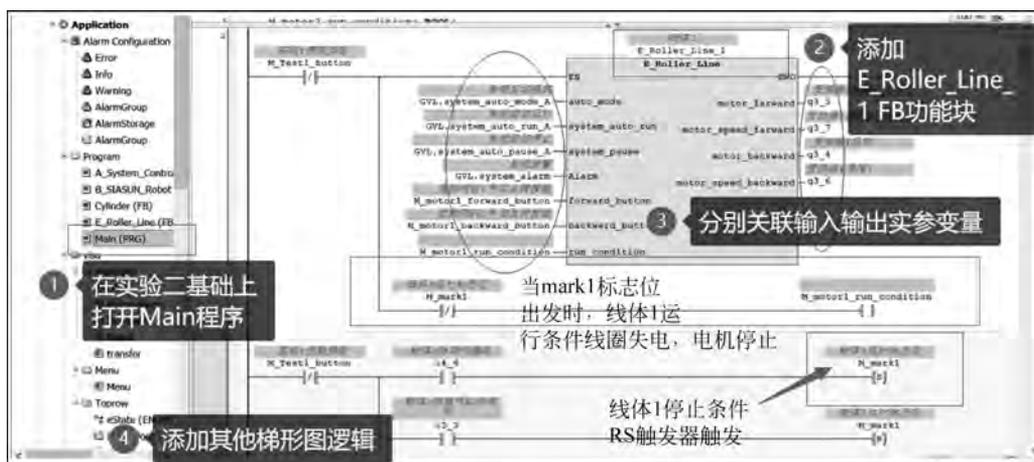


图 5-58 输送线体 1 主程序

侧推气缸主程序。如图 5-59 所示, Cylinder 中的 Up_condition 引脚、Down_condition 引脚关联气缸 1 自动伸出、自动缩回的触发条件, 系统自动运行时, 当输送线 2 末端传感器被触发并且移栽机气缸处于落下状态时, 满足侧推气缸伸出条件, 侧推气缸动作伸出, 当侧推气缸伸出到位磁环触发延时 1s, 满足侧推气缸缩回条件, 侧推气缸动作缩回, 当下一个工件到达末端传感器被触发并且移栽机气缸处于落下状态时, 侧推气缸再次伸出, 由此往复。

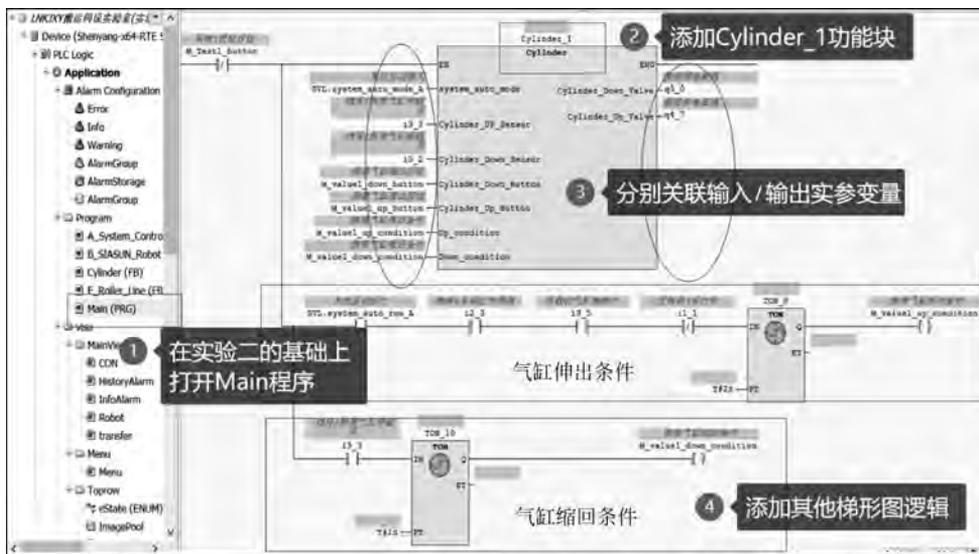


图 5-59 侧推气缸主程序

机器人调度主程序。如图 5-60 所示, 系统自动运行时, 当挡停工位有料传感器被触发并且线体 4 夹紧位传感器被触发时, 满足阻挡气缸伸出条件, 阻挡气缸动作伸出。当线体 4 末端位传感器有料并且挡停工位无料时, 满足阻挡气缸缩回条件, 阻挡气缸动作缩回, 当下一个工件到达挡停工位且满足条件时, 阻挡气缸再次伸出, 如此往复。阻挡气缸自动运行程序编写步骤同侧推气缸。



图 5-60 机器人调度主程序