RUNPOWER®

蓝普锋科技

专注PLC研发及产业化

蓝普锋 PLC 快速入门手册

版权申明

本手册所载的所有材料或内容受版权法的保护，所有版权由蓝普锋拥有，但注明引用其他方的内容除外。蓝普锋公司对其发行的或与合作公司共同发行的包括但不限于产品或服务的全部内容及蓝普锋网站上的材料拥有版权等知识产权，受法律保护。

未经本公司书面许可，任何单位及个人不得以任何方式或理由对手册内容的任何部分进行使用、复制、修改、抄录、传播或与其它产品捆绑使用、销售。

凡侵犯本公司版权等知识产权的，本公司必依法追究其法律责任。

蓝普锋会不断改进和创新核心技术，手册内容请以最新版为准。手册中的例子仅用于举例说明，我们不承担根据本手册及本手册的实例进行的实际生产所造成的结果。

北京蓝普锋科技有限公司保留该手册的全部权利。

Microsoft ®及 Windows ®为美国 Microsoft Corporation 在美国及其他国家的商标或注册商标。

RunProV3、RunPro、RUNPOWER 和蓝普锋的字样均为北京蓝普锋科技有限公司的注册商标。

其他的产品名、公司名分别为各公司的商标或注册商标。

目 录

前言

可编程控制器简介

可编程控制器（Programmable Logic Controller）简称为 PLC。它是在 1969 年以来，在继电-接触器控制技术和计算机技术的基础上发展起来的一种专门用于工业自动化的新型控制设备。PLC 具有典型的计算机结构，以微处理器为核心，以编程的方式进行逻辑控制、计数和算术运算等，并通过数字量和模拟量的输入和输出来完成各种生产过程的控制，具有响应时间快、控制精度高、可靠性好、易于与计算机连接、体积小、质量小、低功耗和维护方便等诸多优点。

北京蓝普锋科技有限公司基于二十多年的自动化产品设计、开发和应用经验，自主研制了高性能、双机热备、高速响应的 RPC3000 系列大型 PLC、RPC2000 系列中小型 PLC 和多款行业专用控制器，为高端控制器国产替代解决方案提供更加安全、稳定、可靠、开放的 PLC 产品。RPC 系列产品具有高性能的 CPU、先进的总线架构、高可靠冗余、高精度 I/O、标准化及开放式通讯接口等特点，性能达到了国际同类主流产品的先进水平。目前，公司研制的 PLC 产品被广泛应用于电力、煤炭矿山、石化、市政水务、地铁、供热、冶金、环保、节能、建筑、水利、农业等多个行业。

随着中国工业智能制造的持续深入发展，PLC 作为智能制造环节中上承信息化传输，下接底层设备控制及数据采集、计算的中枢，采用具有自主技术的国产设备具有非常重要的战略意义。蓝普锋公司将致力于国家产业升级和关键部件、核心技术国产化的需求研发，助力“中国制造”，不断完善国产化替代方案，全面实现关键技术国产化。

为使广大用户快速掌握 PLC 技术，学会用 PLC 构建自动控制解决方案，本手册以 RPC 系列 PLC 为例，为读者呈现出用 PLC 实现典型控制方案的完整过程，主要包括以下内容：

- 软件的安装及使用、控制程序的编制方法；
- 输入、输出信号的接线方法；
- 程序的下载和调试；
- 典型 PLC 控制案例实现过程。

编程前准备

- RPC 系列 PLC 控制器一套（包括电源模块、CPU 模块、I/O 模块等扩展模块）；
- 24V 开关电源 1 块（用于为 I/O 模块供电）‘
- RJ45 接口网线 1 根（用于 PC 机与 PLC 之间的通讯）；
- 输入/输出控制设备（如：按钮、各类传感器、继电器、LED 指示灯等）。

1. PLC 编程软件使用简介

1.1 软件获取

蓝普锋 RPC 系列 PLC 软件免费下载链接为：http://www.runpower.cn/rjxz.php，用户可根据需要下载安装。

为适应不同用户需求，蓝普锋公司提供了可适用于不同 Windows 版本和机型的 PLC 编程软件，支持 WIN7 及以上版本用户安装，分为 32 位和 64 位等不同版本。用户可根据所使用电脑的配置信息选择合适的软件版本下载。

鼠标右键单击“我的电脑”/“属性”选项卡，在打开的页面中可查看“系统”/“系统信息”，确认所使用的 Windows 版本信息，如图 1.1 所示。

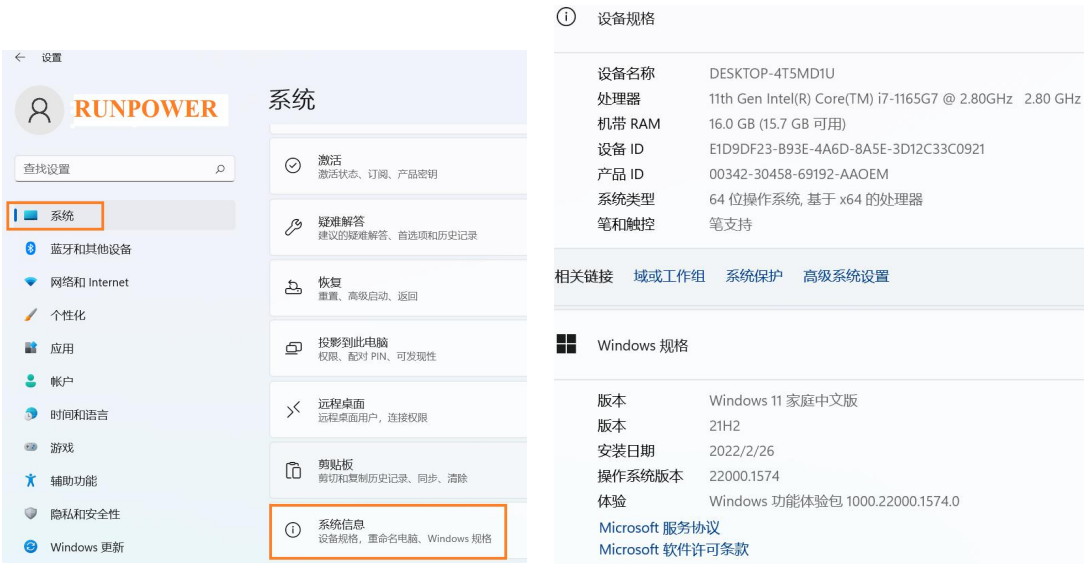


图 1.1 Windows 版本信息查询

1.2 软件安装

RPC 系列 PLC 编程软件安装注意事项如下：

- 安装环境：Windows7 SP1 及以上版本，推荐使用 Windows10 操作系统。
- 解压软件：需将下载的安装包解压缩后再安装。
- 管理员权限配置：软件安装完毕后要先设置管理员权限运行程序，再安装库文件后方可正常使用。

1.2.1 编程软件安装

以 Window10 系统 64 位 PC 机为例，选择安装的软件版本为“SP17 64 位 PLC 编程软件”。安装方法如下：

双击解压后的文件夹中的 CODESYS 64 3.5.17.30.exe 文件开始安装，出现如图 1.2 所示提示框，选择“Install”进行安装。

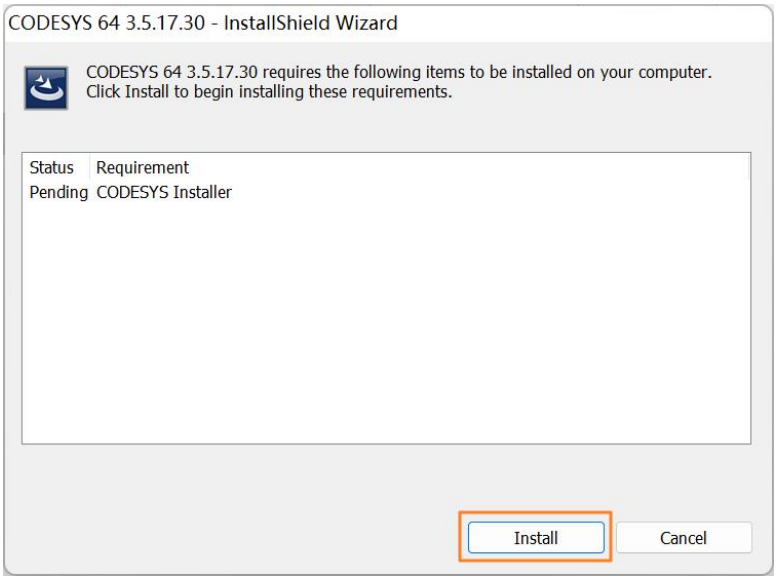


图 1.2 CODESYS 安装界面 1

在出现的安装向导对话框中，点击“Next”进入下一步，如图 1.3 所示。

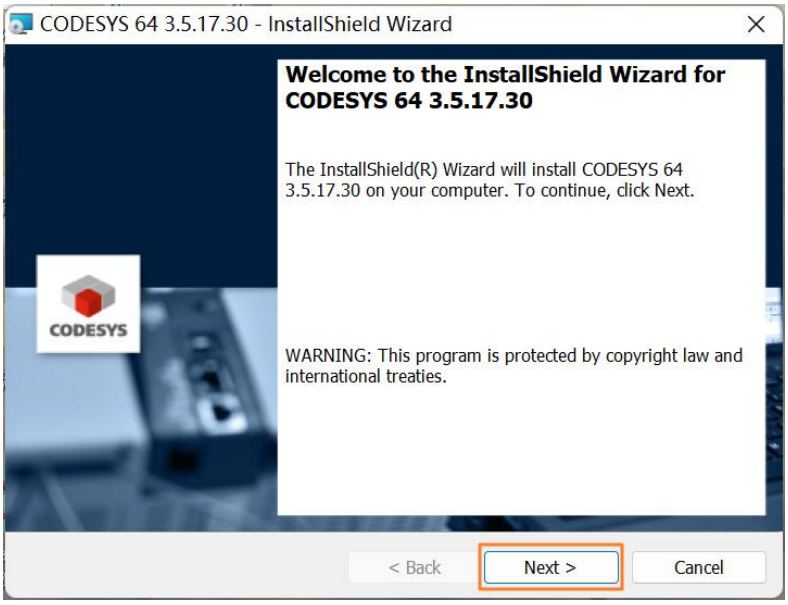


图 1.3 CODESYS 安装界面 2

当安装进程中出现图 1.4 所示“协议许可”对话框，选择“I accept the terms in the license agreement”（接受协议）选项，然后点击“Next”进入下一步。当出现图 1.5 所示“阅读重要信息”对话框，选择“I have read the information”（已阅读）选项，然后点击“Next”进入下一步。

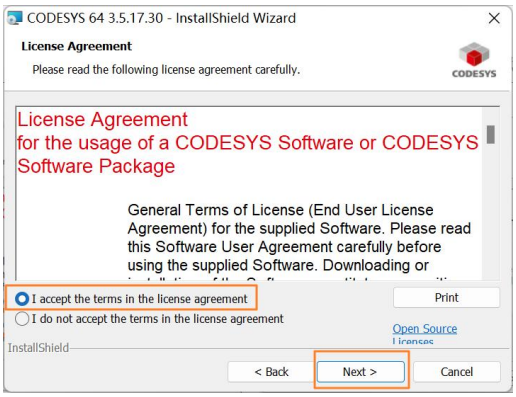


图 1.4 CODESYS 安装界面 3

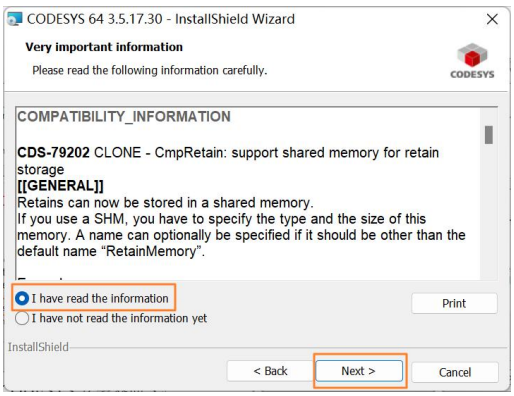


图 1.5 CODESYS 安装界面 4

按照安装向导的提示进行安装，至图 1.6 所示的“Setup Type（安装类型）”选项，选择“Complete”（完整安装）选项。

安装程序执行至图 1.7 所示的提示框时，安装进程将持续约十分钟，请耐心等待。

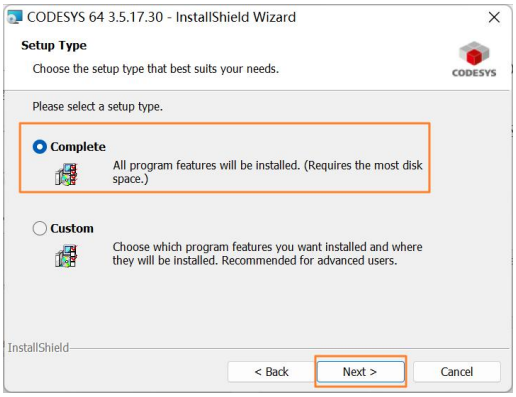


图 1.6 CODESYS 安装界面 5

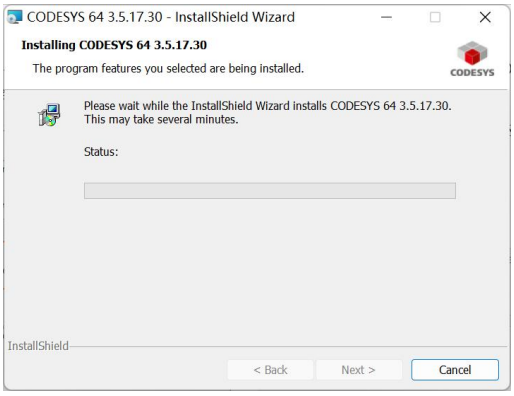


图 1.7 CODESYS 安装界面 6

安装即将结束时会出现图 1.8 所示的结束安装提示框，点击其右下角的“Finish”按钮即可完成软件的安装。安装完成后，电脑桌面上会出现 CODESYS 快捷方式图标  。

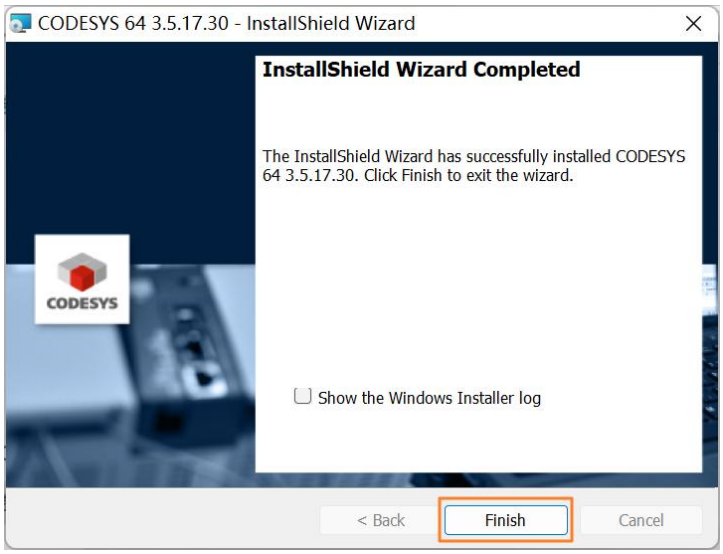


图 1.8 CODESYS 安装结束界面

1.2.2 库文件安装

1.2.2.1 管理员权限配置

软件安装完成后，在初次使用之前，需进行管理员权限配置，再进行设备库和指令库的安装，将 RPC 系列 PLC 的设备库文件和指令库文件加载到 CODESYS 程序中。如图 1.9 所示，右键单击桌面上的 CODESYS 快捷方式图标，点击“属性（R）”。在打开的图 1.10 所示的软件属性对话框中单击“兼容性”，并勾选“以管理员身份运行此程序”，再点击“确定”，即完成管理员权限配置。

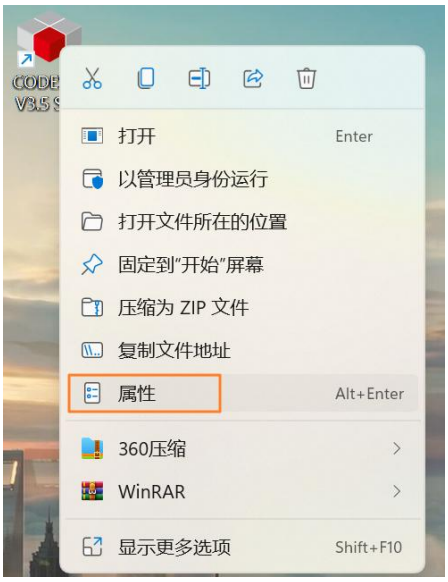


图 1.9 快捷方式右击菜单栏

图 1.10 管理员权限配置

1.2.2.2 设备库安装

双击桌面上的 CODESYS 快捷方式图标打开程序，点击菜单栏“工具/设备存储库...”，如图 1.11 所示。

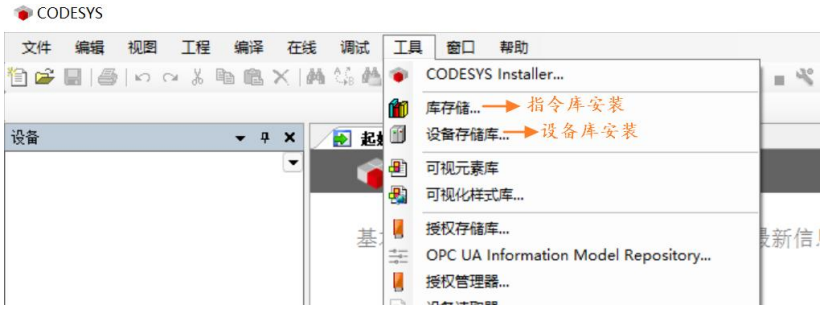


图 1.11 设备库安装（1）

在图 1.12 所示的设备存储库弹出框中点击“安装（I）…”，按照已下载的设备库文件

存储路径打开相应文件夹，选择全部安装文件，如图 1.13 所示。再点击图中右下角的“打开（O）”按钮，即可完成设备库文件的安装，并自动退回到“设备存储库”对话框，同时显示已安装的设备库文件，如图 1.14 所示。

说明：如果打开文件夹后不显示设备库文件内容，可以修改图 1.13 右下角的打开文件属性，选择“所有支持的描述文件”。

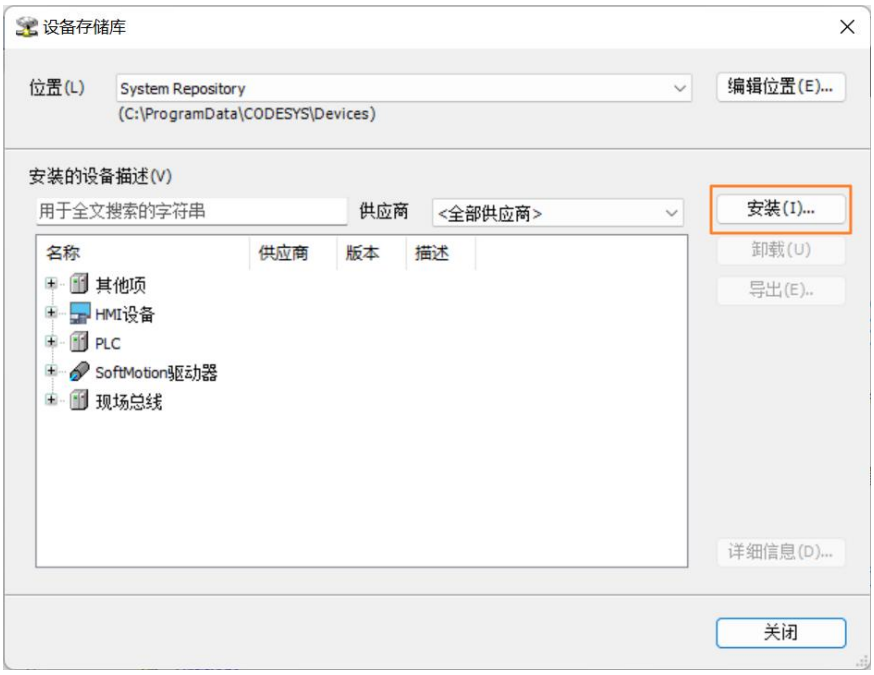


图 1.12 设备库安装（2）

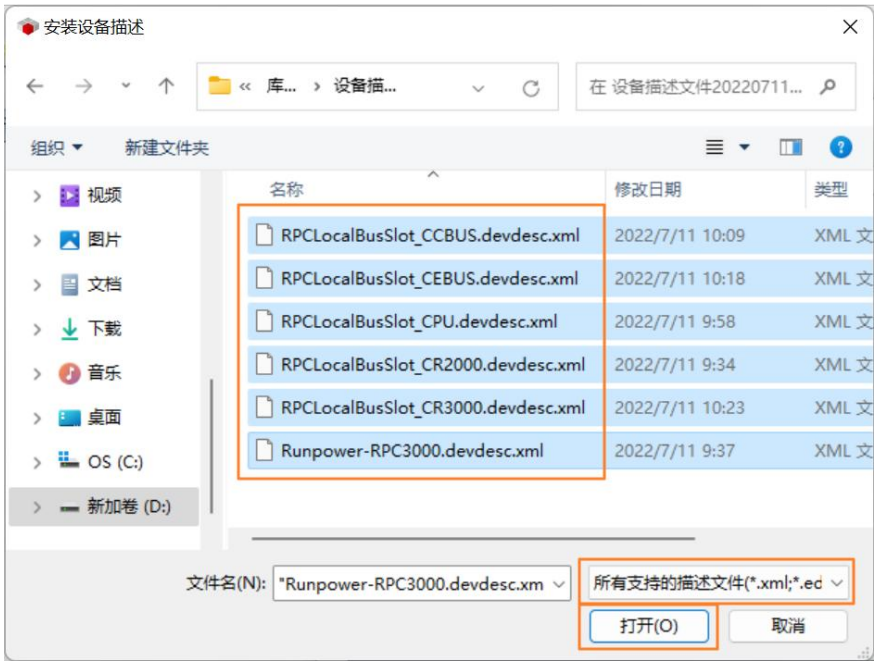


图 1.13 设备库安装（3）

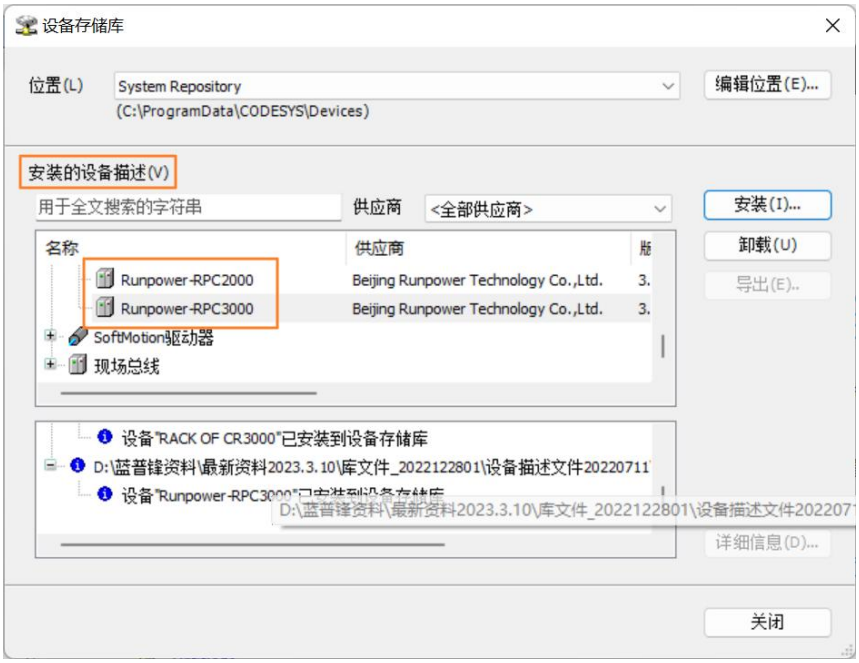


图 1.14 设备库安装（4）

1.2.2.3 指令库安装

双击桌面上的 CODESYS 快捷方式图标打开程序，点击图 1.11 中菜单栏“工具/库存储...”，弹出“库存储”对话框，点击“安装...”，如图 1.15 所示。

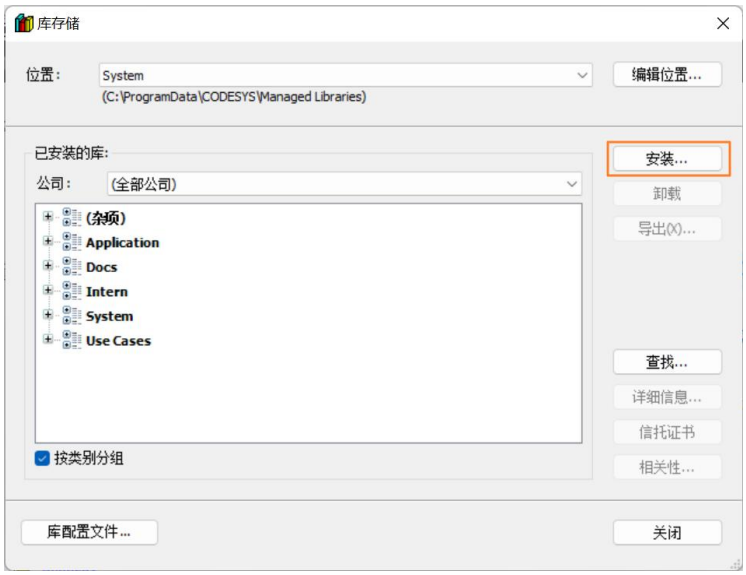


图 1.15 指令库安装（1）

按照已下载的指令库文件存放路径找到指令库所在文件夹，如图 1.16 所示，选择全部安装文件，点击右下角的“打开（O）”按钮，即可完成指令库文件的安装，并自动退回到“库存储”对话框，同时显示已安装的指令库文件，如图 1.17 所示。点击“关闭”即可退回到程序打开的初始界面。

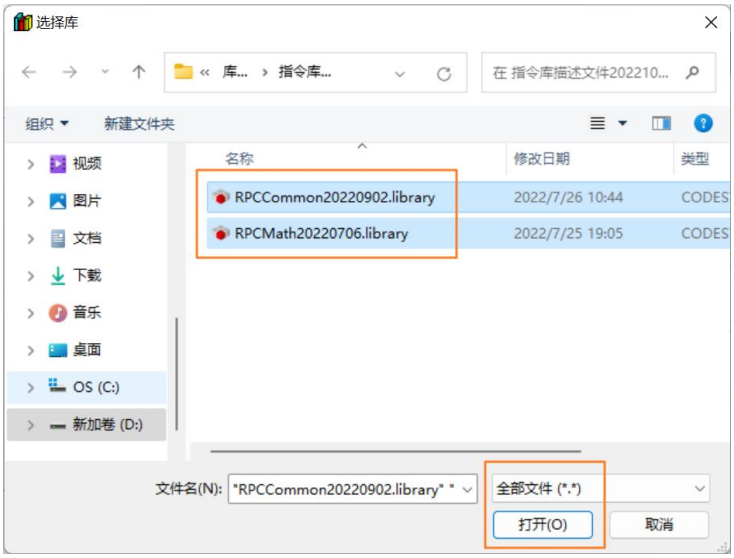


图 1.16 指令库安装（2）

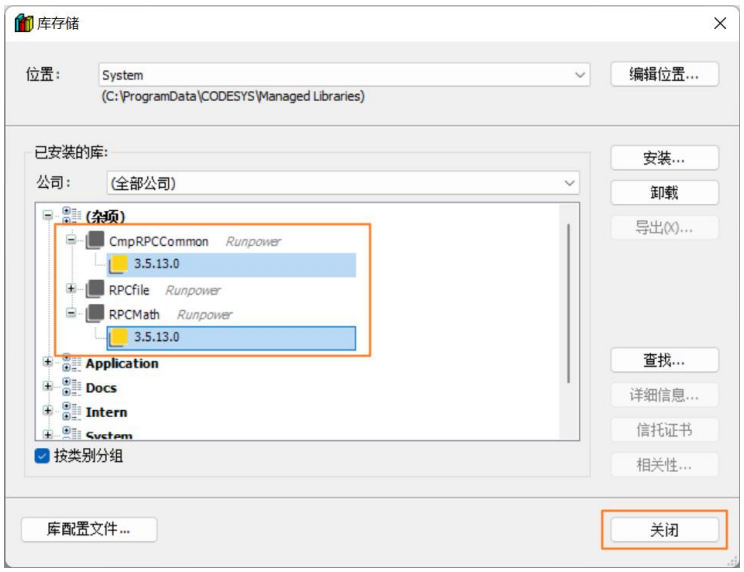


图 1.17 指令库安装（3）

1.2.3 软件卸载

如果计算机中已经安装了低版本的 CODESYS 软件，当安装新版本的软件时，需要将已安装的旧版本的软件卸载。以安装软件版本（CODESYS64 3.5.17.30）为例，卸载方法如下：

如图 1.18 所示，在 Windows 系统依次点击“开始”/“设置”/“应用”/“应用和功能”，选择 CODESYS64 3.5.17.30，点击程序右侧的选项 “┇” 并选择“卸载”便可根据安装向导的提示卸载该程序。卸载进程约需十分钟，请耐心等待。

图 1.18 卸载 CODESYS

卸载完成后，需删除安装路径下的残留文件夹选项，如图 1.19 所示。

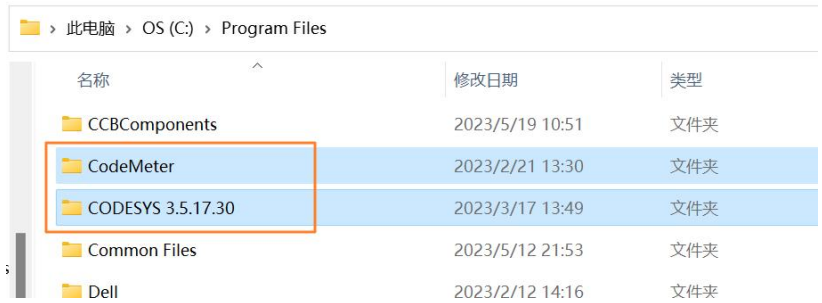


图 1.19 删除卸载残留文件

1.3 基本编程操作

根据控制系统要求进行硬件选型并且完成软件安装后，即可开始进行工程的创建。创建工程的过程就是搭建控制系统软件平台的过程，需要根据硬件配置情况进行设备组态、程序的编制以及任务配置等工作。最后工程编译完成无误后，可根据需要进行仿真调试或者将工程下载至 PLC，完成工程在线调试和运行。

1.3.1 新建工程

双击桌面上的 CODESYS 快捷方式图标打开应用程序后，用户可通过以下三种方式之一新建工程，如图 1.20 所示。

① 点击“起始页”中的“新建工程…”选项。

② 打开菜单栏“文件/新建工程”。

③ 快捷键“Ctrl+N” 。

以上新建工程操作后，会弹出“新建工程”对话框，如图 1.21 所示。

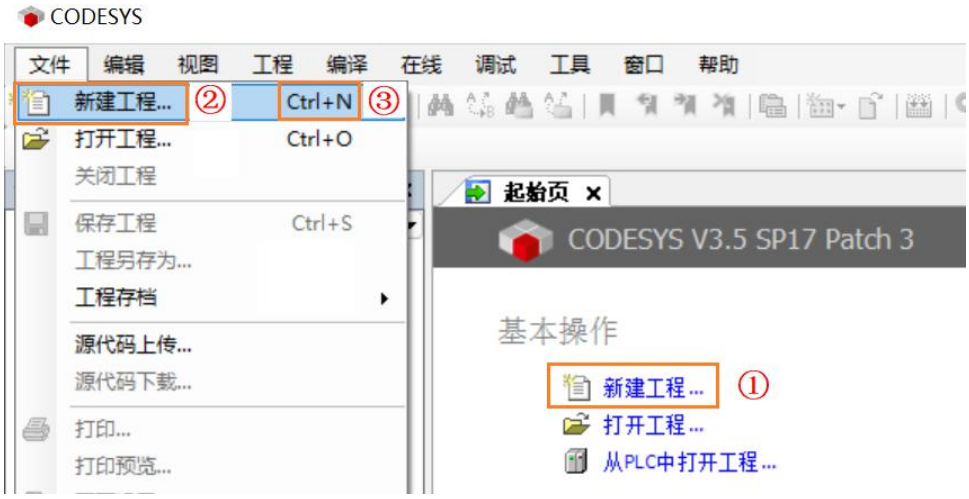


图 1.20 新建工程（1）

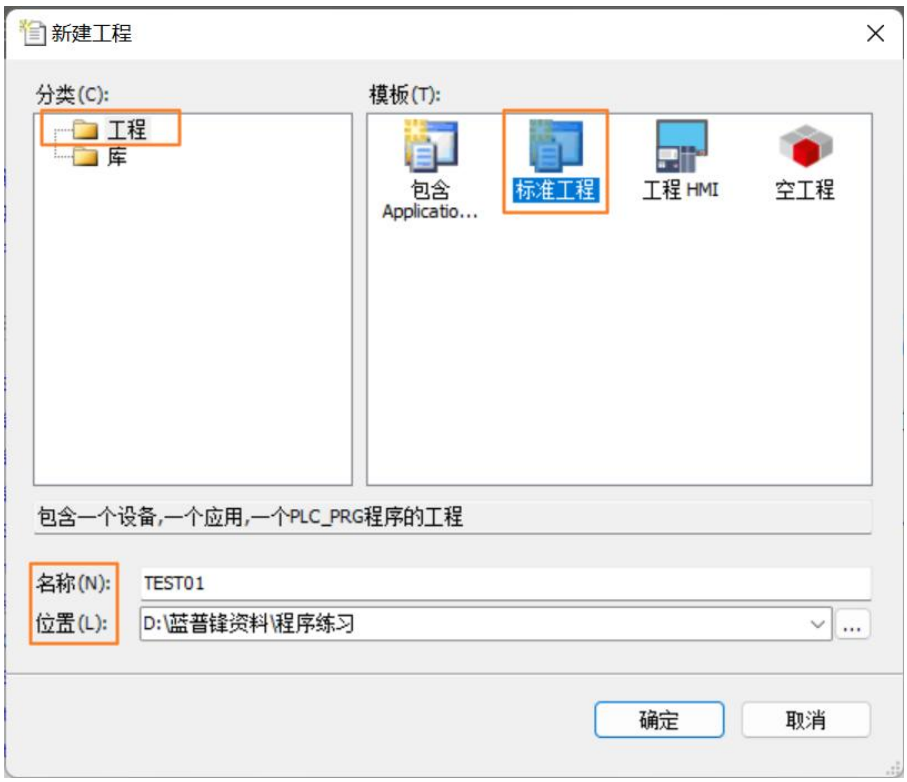


图 1.21 新建工程（2）

说明：用户可根据习惯切换语言设置，单击菜单“工具/选项…”，在弹出的如图 1.22 所示的“选项”对话框中单击左侧“语言设置”选项，在右侧语言设置界面设置“用户界面语言”和“帮助语言”，可在各种语言之间进行切换，单击确定后，重新启动 CODESYS 软件即可切换为用户需要的语言。

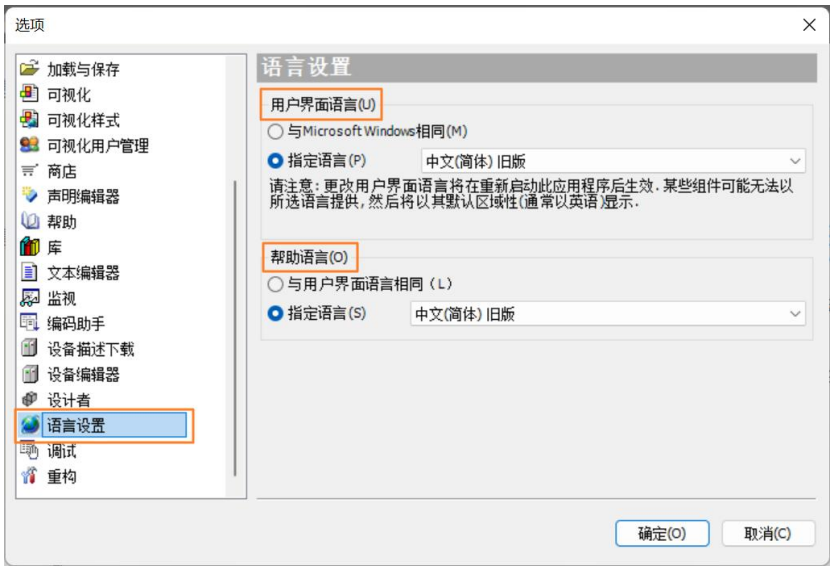


图 1.22 语言设置界面

在图 1.21 左侧“分类（C）”栏选择“工程（Projects）”，右侧“模板（T）”栏选择“标准工程”选项。通过“名称（N）”对工程进行命名，通过“位置（L）”/“...”进行保存路径选择。点击“确定”，弹出“标准工程”对话框，如图 1.23、1.24 所示。

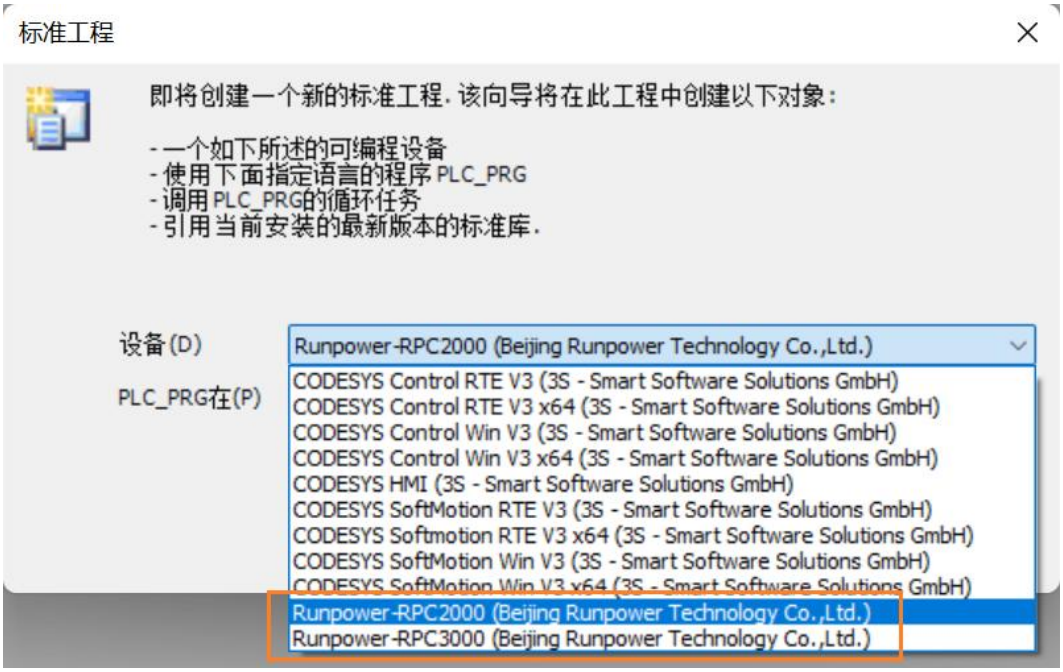


图 1.23 选择设备

点击图 1.23 中的“设备（D）”栏，从下拉列表中列出的可选设备中选择与控制系统 CPU 号相匹配的设备。本手册适用于 RPC2000 系列 PLC 和 RPC3000 系列 PLC 初学者，用户可根据需要选择“Runpower-RPC2000/3000”设备。

图 1.24 选择编程语言

点击图 1.24 中的“PLC_PRG 在（P）”选项，在下拉列表中列出了软件所支持的六种编程语言，用户可任选一种语言，例如选择“梯形逻辑图（LD）”作为编程语言，点击“确定”，即可新建名称为*.project 的工程文件，并自动进入工程开发界面。

1.3.2 打开工程

打开已创建的工程文件有以下三种方式：

① 在工程文件存储位置，直接双击打开*.project 文件。

② 双击桌面 CODESYS 图标打开程序，点击“起始页”中的“打开工程…”选项，如图 1.25 所示。

③ 双击桌面 CODESYS 图标，打开应用程序，选择菜单“文件/打开工程”或者用快捷键“Ctrl+O”。

通过②和③两种方式打开工程操作后，均会弹出“打开工程”对话框，选择工程文件所在的文件夹，打开相应的*.project 文件。

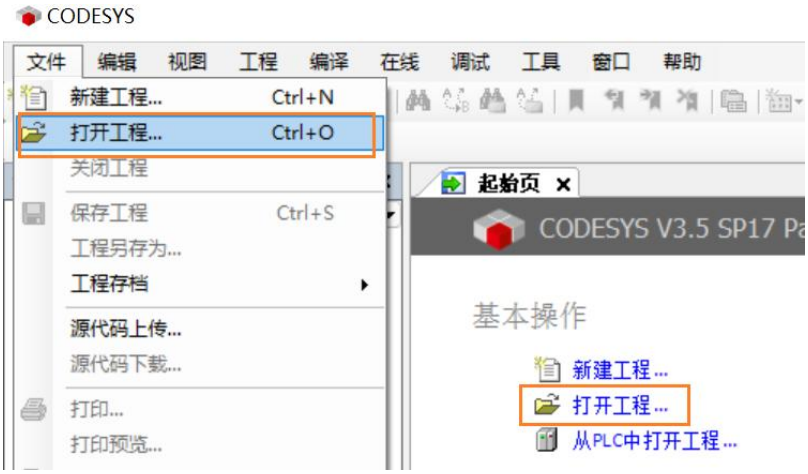


图 1.25 打开工程

1.3.3 软件结构介绍

打开新建的标准工程界面如图 1.26 所示。单击菜单“视图”，可以打开相应的视图界面，默认为打开工程的设备视图。在工程中可同时打开多个视图，单击底部的切换按钮可在不同视图间切换。图中左侧视图为默认打开的设备树视图，在设备树中，通过各类“设备”对象表示该工程的硬件设备系统。该视图由多个节点组成，左侧的“+”和“-”按钮分别用于展开和收回该节点内容。设备树视图中的各个节点介绍如下：

- “Device”——对应工程中的控制器（RPC 系列 PLC 为 Runpower-RPC3000/2000）。图 1.26 所示的工程（工程名称：电机控制）对应的控制器为 Runpower-RPC3000。工程的具体硬件构成需通过设备组态进行手动添加，将在后续章节详细介绍。
- “PLC 逻辑”——对应控制器编程的相关内容，由各种 PLC 应用程序（Application）构成。
- “Application”——由三部分构成，分别是“库管理器”、PLC 程序“PLC_PRG”和“任务配置”。其中，库管理器用于存放标准指令库和管理用户手动添加的各种设备库和指令库；“PLC_PRG”为系统自动添加的默认名称为 PLC_PRG 的程序，用来编制 PLC 控制程序，用户可修改该程序名称和属性，也可以通过右键单击“Application/添加对象”，选择添加 POU（Program Organization Unit）来创建新的程序、功能块、或者函数；“任务配置”用来设置 PLC 程序的调用方式，工程创建后系统会自动生成一个名为“MainTask”的任务，任务中包含了默认的程序“PLC_PRG”，所有用户新建的程序都必须在任务配置中进行调用，否则无法正常运行。

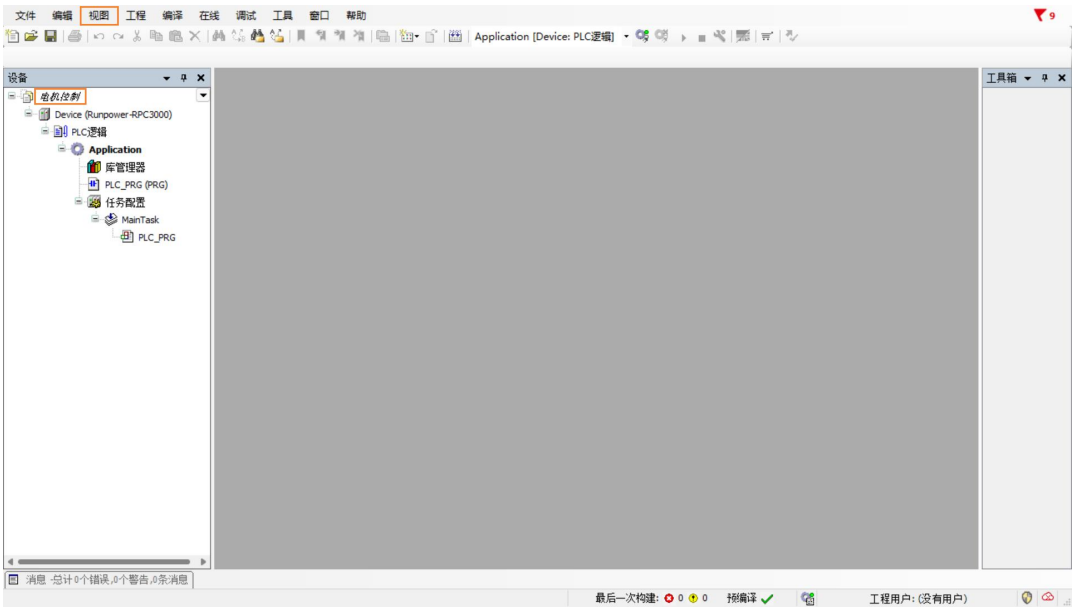


图 1.26 打开工程文件界面

1.3.4 设备组态

新建工程之后，需进行设备组态（硬件配置），使工程文件（*.project）中的设备型号与

实际控制系统中所选用的控制器型号相匹配。用户可根据实际控制系统功能需求选用 RPC 系列控制器及其扩展模块搭建控制系统。

对于 RPC3000 系列 PLC，可选 CPU 模块及扩展模块如表 1.1 所示。

对于 RPC2000 系列 PLC，可选 CPU 模块及扩展模块如表 1.2 所示。

表 1.1 RPC3000 系列 PLC 可选设备列表

可选设备	设备型号	设备规格
CPU 模块	RPC3101	600MHZ，4M 程序，8M 数据，1-以太网，2-串口，支持 CC、CR 总线，DC24V 供电
	RPC3105	800MHZ，8M 程序，32M 数据，2-以太网，2-串口，支持 CC、CR、CH、CE 总线，DC24V 供电
	RPC3111（冗余）	1GHZ，16M 程序，64M 数据，2-以太网，2-串口，支持 CC、CR、CH、CE 总线，支持冗余，DC24V 供电
背板及电源模块	RACK004/006/008/011/015	4/6/8/11/15 槽背板 可安装 RPC3000 系列 PLC 控制器、电源及扩展模块
	RPC3910	交流电源模块：30W 开关电源，AC220V 输入
	RPC3921	直流电源模块：30W 开关电源，DC24V 输入
通讯模块	RPC3700	CC 总线本地机架扩展模块
	RPC3701-M	CE 总线远程机架扩展主站模块；2-RJ45 接口
	RPC3701-S	CE 总线远程机架扩展从站模块；2-RJ45 接口
	RPC3703	Profinet 通讯从站模块；2-RJ45，共享一个 IP 地址，集成交换机功能
	RPC3704	EtherCAT 通讯从站模块；2-RJ45，分别为输入与输出，用于多个模块手拉手连接使用
	*RPC3705	Ehernet/IP 通讯从站模块
	RPC3706	CANopen 通讯，单路主站模块；1-CAN 接口，CANH 接 CAN 高，CANL 接 CAN 低，CGND 接屏蔽
	RPC3707	CAN2.0 通讯，单路自由口模块；1-CAN 接口，CANH 接 CAN 高，CANL 接 CAN 低，CGND 接屏蔽
	RPC3708	通讯扩展模块，支持通讯可编程；含有 3 路串口（RS485）和 1 路以太网（10M/100M）
	RPC3720	冗余同步模块；含 0.5 米 SPF 接口光纤
	RPC3770	CR 总线远程通讯从站模块
可选设备	名称/型号	I/O 规格
数字量扩展模块（DI/DO）	RPC3211	16-DI，DC24V 供电，漏型/源型
	RPC3212	32-DI，DC24V 供电，漏型/源型
	RPC3214	8-DI，AC220V 供电，单通道隔离

	RPC3216	1-SSI/增量编码器输入，2-DO，继电器输出
	RPC3218	16-DI，漏型/源型，事件记录（SOE）
	RPC3221	16-DO，DC24V 供电，晶体管输出，源型
	RPC3222	32-DO，DC24V 供电，晶体管输出，源型
	RPC3224	8-DO，AC220V，继电器输出，单通道负载 5A
模拟量扩展模块（AI/AO）	RPC3310	8-AI：4～20mA/0～20mA/0～10V 可选，16 位 A/D，精度 0.2%，差分输入，8 通道刷新时间 10ms
	RPC3311	8-热电偶（TC），J、K、E、N、T、B、R、S 型，-80mV~+80mV，精度 0.2%，8 通道刷新时间 20ms
	RPC3312	6-热电阻 RTD，Cu50、PT100，精度 0.2%，6 通道刷新时间 20ms
	RPC3313	16-AI：4～20mA/0～20mA 可选，16 位 A/D，精度 0.5%，单端输入，16 通道刷新时间 20ms
	RPC3320	4-AO：电流型/电压型（0～20mA/0～10V 可选，精度 0.5%）
	RPC3321	8-AO：电流型/电压型（0～20mA/0～10V 可选，精度 0.5%）

表 1.2 RPC2000 系列 PLC 可选模块列表

可选设备	型号	供电	I/O 规格
CPU 模块	RPC2116	DC24V	24-I/O：14-DI（DC24V），10-DO（（DC24V）晶体管）；1-RS232，1-RS485
	RPC2117	AC220V	24-I/O：14-DI（DC24V），10-DO（继电器）；1-RS232，1-RS485
	RPC2117A	AC220V	21-I/O：10-DI（DC24V），8-DO（继电器）；2-AI，1-AO（模拟量 0～10V/0～20mA 可选、精度 1%）；1-RS232，1-RS485
	RPC2117N	AC220V	16-I/O：10-DI（DC24V），6-DO（继电器）；1-RS232，1-RS485，1-以太网
	RPC2117R	AC220V	3-RS485，2-独立以太网
	RPC2711	AC220V	控制及保护单元，8-DI、6-DO、3-交流电压、3-交流电流、1-零序电压、1-零序电流、1-系统电压、1-附加直流、1-电阻值、1-AI，2-RS485
可选设备	名称/型号	I/O 规格	
数字量扩展模块（DI/DO）	RPC2211	16-DI，DC24V	
	RPC2221	16-DO（晶体管），DC24V	
	RPC2223	16-DO（继电器，输出电压为 DC24V 或 AC220V）	
	RPC2231	8-DI，DC24V；8-DO（继电器，输出电压为 DC24V 或 AC220V）	
模拟量扩	RPC2310	4-AI： 4～20mA/0～20mA/0～10V 可选，12A/D，精度 0.5%，差分输入，	

展模块（AI/AO）		4 通道刷新时间 10ms
	RPC2311	4-热电偶：J、K、E、N、T、B、R、S 型，-80mV~+80mV，精度 0.2%，4 通道刷新时间 20ms
	RPC2312	4- RTD（热电阻）：Cu50、PT100，精度 0.2%，4 通道刷新时间 20ms
	RPC2313	8-AI：4～20mA/0～20mA/0～10V 可选，12A/D，精度 0.5%，单端输入，8 通道刷新时间 20ms
	RPC2314	8-热敏电阻输入：R25℃为 10K、B 值可选的 NTC，12A/D，精度 0.5%，8 通道刷新时间 20ms
	RPC2320	2-AO：0～20mA/0～10V 可选，精度 1%
	RPC2321	4-AO：0～20mA，精度 1%
	RPC2330	4-AI：0～10V/4～20mA/0～20mA 可选，12A/D，精度 0.5%，单端输入，4 通道刷新时间 10ms；1-AO：输出 0～10V/0～20mA 可选，精度 1%
可选设备	名称/型号	模块说明
通讯扩展模块	RPC2401	Profibus-DP 协议从站接口模块
	RPC2403	Ethernet 通讯接口模块
	RPC2404	RS485 串口扩展模块
	RPC2410	多协议网关模块，1 路以太网口，1 路 RS232 或 RS485，2 路 RS485
电力监控及保护模块	RPC2730	四回路测量保护模块，12 路交流电流，4 路电阻和 1 路系统电压
	RPC2731	电参数采集模块，3 路交流电压，3 路交流电流

例如：构建一个包含数字量和模拟量输入/输出控制功能的小型控制系统，若选用 RPC3000 系列 PLC，新建工程时（如图 1.23）可选择 Runpower-RPC3000 设备，设备组态中需配置背板、电源模块及输入、输出模块等扩展模块，硬件配置如表 1.3 所示。

表 1.3 RPC3000 控制系统硬件配置表

背板型号	RACK OF CPU					
背板位置	1 号槽	2 号槽	3 号槽	4 号槽	5 号槽	6 号槽
模块型号	RPC3910	RPC3101	RPC3211	RPC3221	RPC3310	RPC3321
说明	电源模块	CPU 模块	数字量输入（DI）模块	数字量输出(DO)模块	模拟量输入（AI）模块	模拟量输出(AO)模块

若选用 RPC2000 系列 PLC，新建工程时可选择 Runpower-RPC2000 设备，软件中设备组态会自动添加 RPC2117N 型 CPU 模块，用户可根据实际系统配置情况添加其他扩展模块。在构建小规模逻辑控制系统时，选用的 CPU 为 RPC2117N（自带 10-DI，6-DO 继电器输出），在设备组态中只需添加模拟量输入/输出模块 RPC2330 即可，硬件配置如表 1.4 所示，

表 1.4 RPC2000 控制系统硬件配置表

PLC 设备	Runpower-RPC2000		
可选模块	RPC2117N（默认）	RPC2330	表 1.2 所列模块
说明	CPU（含 I/O）	4AI，1AO	根据需要选择

在软件设备组态中，用户可添加或更改 CPU 模块的型号，并根据系统配置依次添加表中所列其他扩展模块设备。

1.3.4.1 CPU 型号设置及更改

进行设备组态之前，首先检查设备树视图“Device”中所列设备是否与实际 PLC 选用的 CPU 型号相符，例如：在新建工程时选择了设备型号为 Runpower-RPC3000，则 Device 会显示设备型号如图 1.27 所示。如果要更改 CPU 型号，可在图 1.27 中右键单击“Device（Runpower-RPC3000）”，选择“更新设备…”选项，从弹出的“更新设备”对话框的设备列表中选择实际的 PLC 型号即可，如图 1.28 所示。例如，要将 RPC3000 更改为 RPC2000，单击选择“Runpower-RPC2000”，再单击右下角的“更新设备”按钮即可完成 CPU 型号的更改。

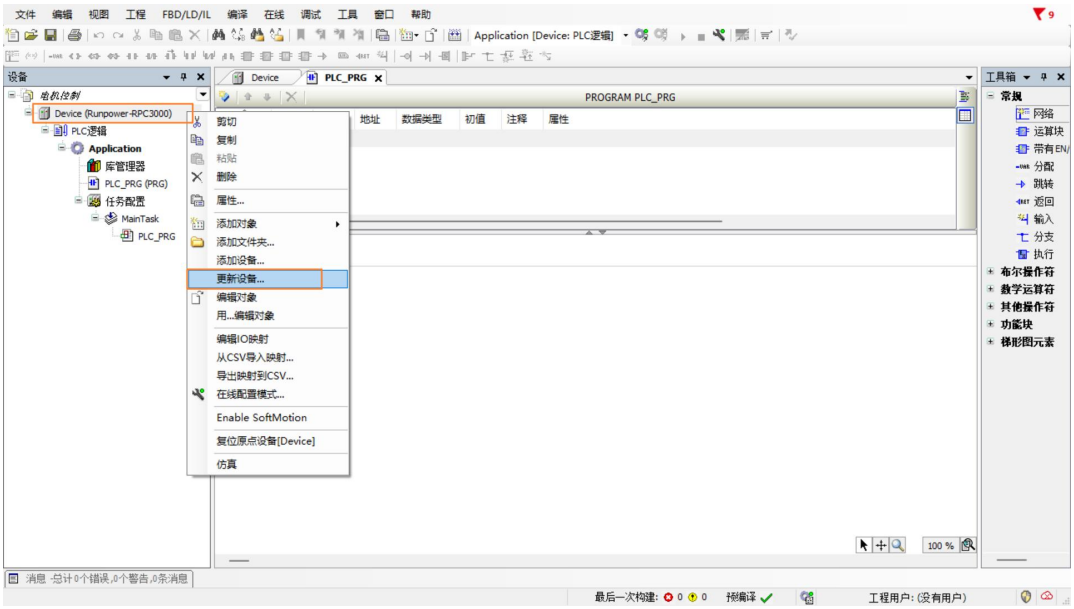


图 1.27 更改 CPU 型号

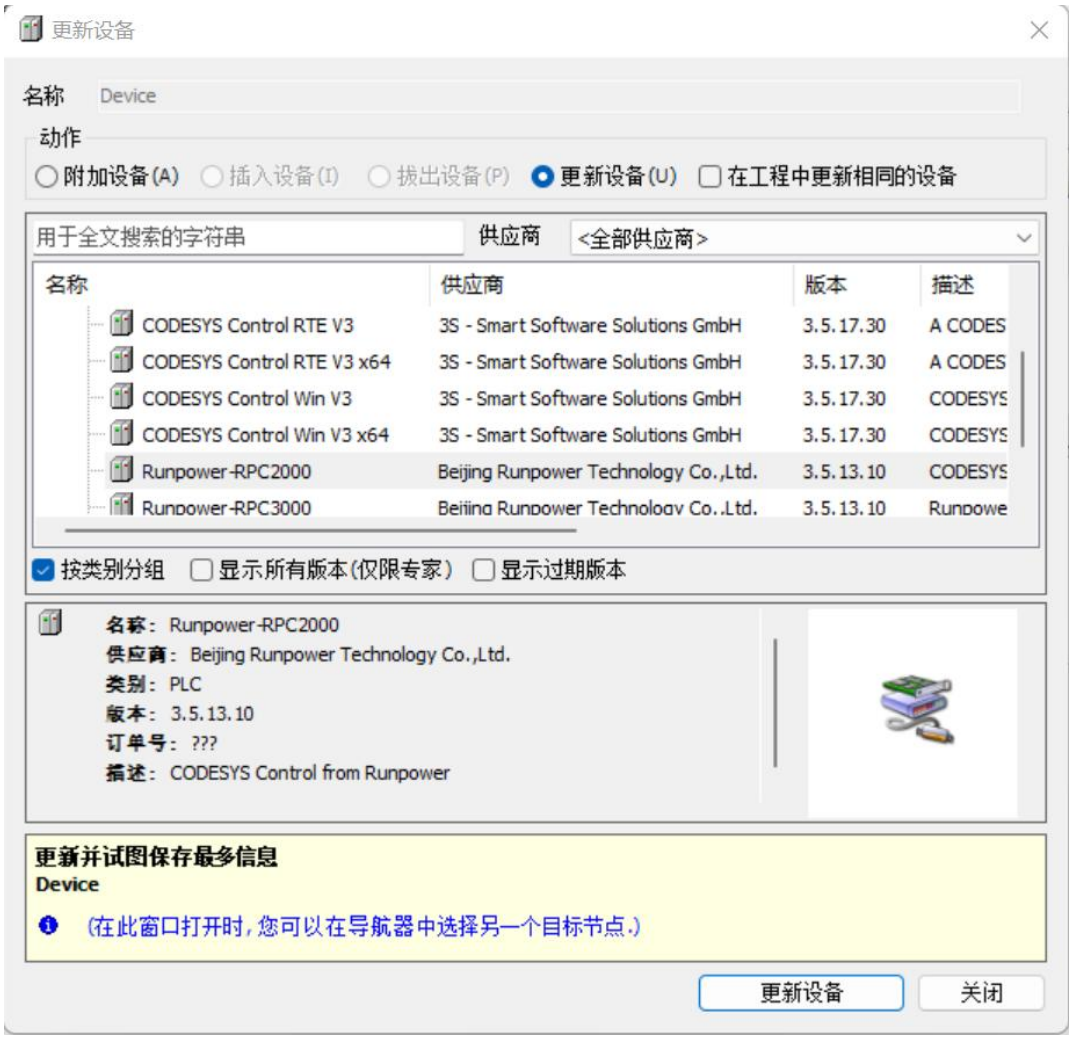


图 1.28 更新设备对话框

1.3.4.2 组态添加背板

对于 Runpower-RPC3000 设备，由于硬件系统中的控制器模块、电源模块、通讯模块、输入/输出等扩展模块均需安装在背板上（背板内置多种总线，最大为 15 槽），因此在设备组态时首先需添加背板。

在设备树视图中右键单击“Device（Runpower-RPC3000）”选择“添加设备…”，弹出右侧添加设备对话框，单击可选设备名称，根据需要选择背板为“RACK OF CPU”，如图 1.29 所示，下方会显示该设备的简介（包含了设备的名称、供应商和设备描述等内容），单击底部的“添加设备”按钮，再单击“关闭”按钮，背板添加成功，如图 1.30 所示。其中，背板的 1 号和 2 号槽被自动添加了电源模块和 CPU 模块。如果不慎选错背板型号，可以右键单击背板进行删除操作，或者直接单击选中背板，按“Delete”键删除。

说明：RPC2000 系列 PLC 无需添加背板，在设备组态时直接添加其他扩展模块即可。

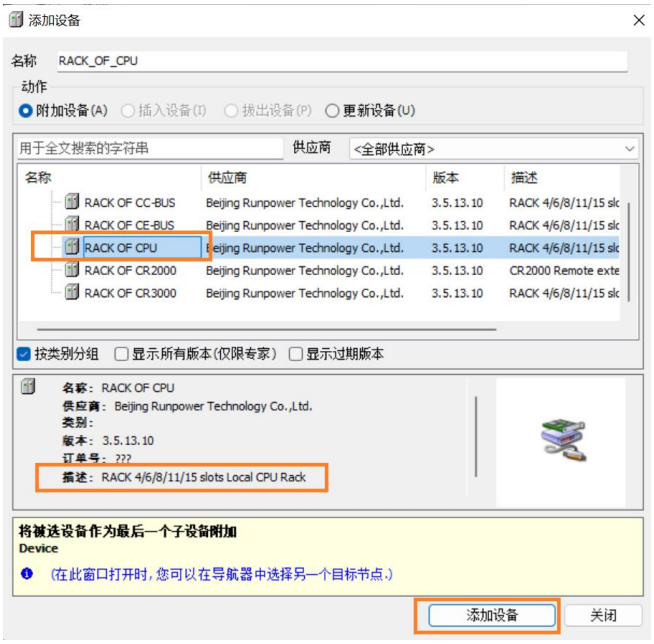


图 1.29 添加背板

1.3.4.3 添加其他模块

鼠标右键单击背板上的空白槽位，在弹出的如图 1.31 所示的“插入设备”对话框中选择所需添加的设备，单击底部的“插入设备”按钮，即可完成该槽位的设备添加。按照表 1.3 中所列设备依次完成 3~6 号槽位设备的添加，就完成了工程的设备组态，最后单击“关闭”按钮，在设备树中会显示已添加的设备，如图 1.32 所示。

对于 Runpower-RPC2000 系列 PLC 的设备组态，无需添加背板，其他模块添加方法同上。

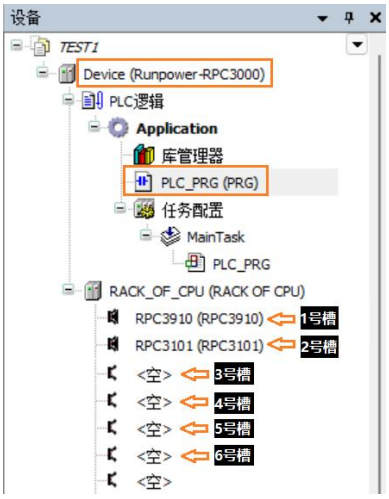


图 1.30 设备组态（1）

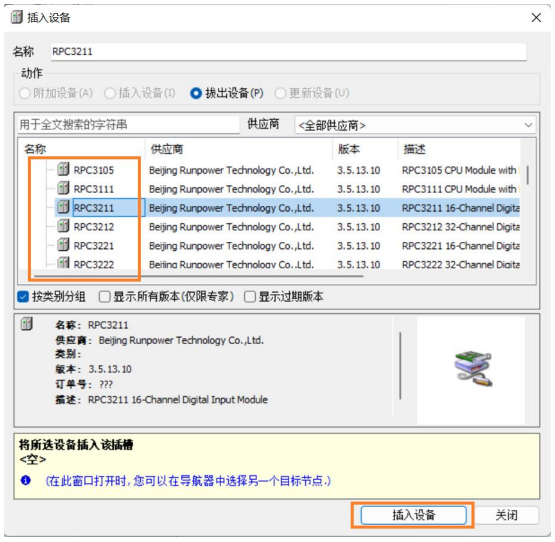


图 1.31 设备组态（2）

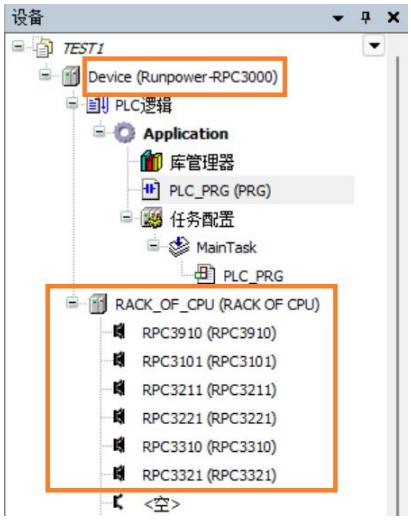


图 1.32 设备组态（3）

1.4 编制 PLC 控制程序

双击设备树视图中的“PLC_PRG(PRG)”，出现如图 1.33 所示的编程界面，其中上部为变量声明区，下部为程序编辑区。用户可单击菜单下的工具栏添加各种触点等指令，也可从右侧工具箱选择所需的指令，并拖动该指令到程序编辑区光标所在位置进行指令添加。

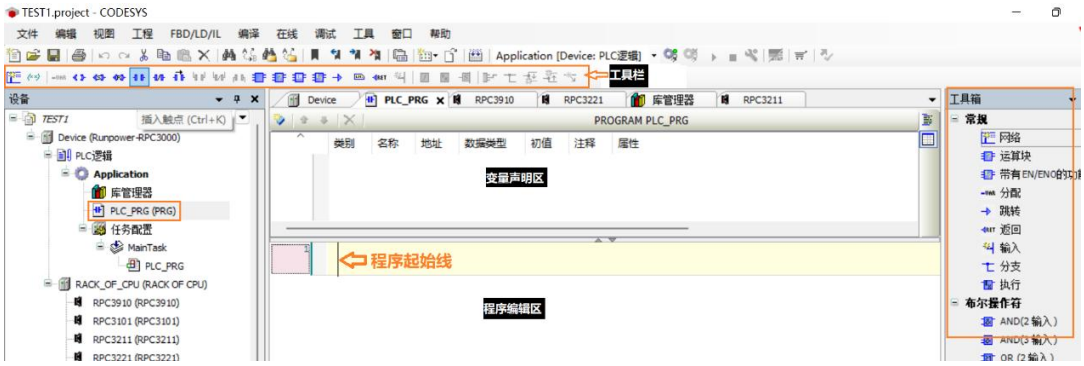


图 1.33 程序编辑界面

1.4.1 输入输出信号分配

电机启停控制电路如图 1.34 所示，采用 PLC 程序代替“控制电路”部分进行电机启停控制，PLC 控制梯形图程序如图 1.35 所示。

PLC 输入信号（DI）为：停止按钮 SB1——%IX0.1；

启动按钮 SB2——%IX0.2；

热继电器（过载保护）FR——%IX0.3。

PLC 输出信号（DO）为：电机控制 KM——%QX0.0。

其中%IX0.1、%IX0.2、%IX0.3 和%QX0.0 对应接入 PLC 输入/输出模块的外部信号的实

际通道地址，用户可自行分配，详细说明可参考后续章节中的介绍。

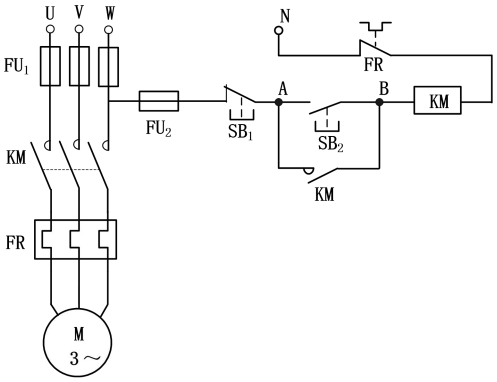


图 1.34 电机启停控制电路

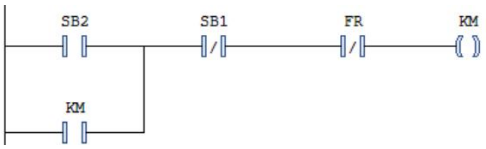


图 1.35 电机启停梯形图程序

1.4.2 梯形图编程方法

梯形图程序中的指令（常开、常闭触点、线圈、定时器等）与继电器的常开、常闭触点、线圈等表示符号相似，编程方式相对容易且易于被工程人员接受，因此，在 PLC 逻辑控制编程中，梯形图最为常见。本节以梯形图方式为例介绍控制程序的编制方法。

1.4.2.1 添加程序段

RPC 系列 PLC 梯形图程序编制方法完全相同，本节以 RPC3000 为例介绍 RPC 系列 PLC 梯形图程序编程方法。

打开图 1.36 所示编程界面。编程区域为空白，需首先添加“网络”。单击工具栏左侧的“添加网络”图标，或者用鼠标拖动右侧工具箱中的“网络”至程序编辑区，添加程序段。也可以通过在程序编辑区右键单击“插入网络”选项，添加程序段。此时工具栏其他图标均为灰色不可用状态。

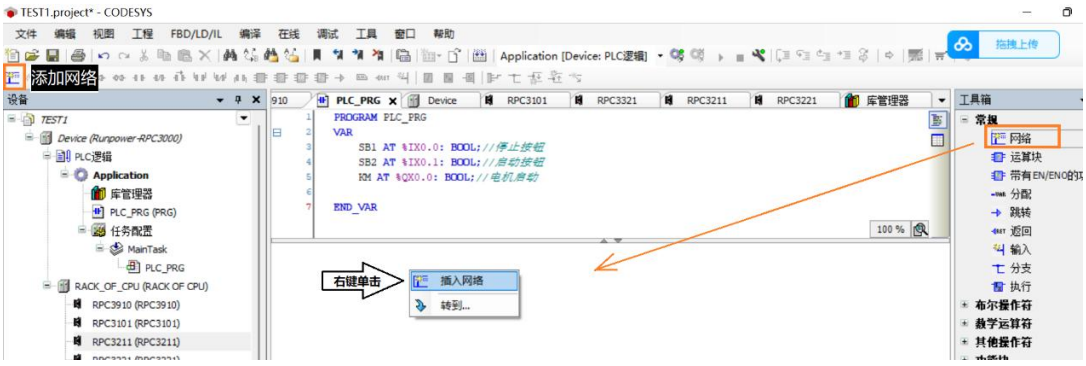


图 1.36 梯形图编程界面

添加程序网络之后，单击选中程序网络区域时，工具栏变为亮色可用，可单击调用工具栏指令进行编程。

1.4.2.2 添加指令

图 1.35 所示的梯形图程序编程方法如下：单击工具栏中的图标“ ”，插入一个常开触点指令，删除触点上方的“???”，并输入常开点名称“SB2”。由于变量 SB2 在程序中未被事先声明，会弹出“自动声明”对话框，如图 1.37 所示，默认变量范围为“var”（局部变量），类型为“BOOL”（布尔型，取值为 true/false）。该变量是来自 PLC 输入模块的指令，需按输入/输出信号分配情况设置与输入模块相应通道对应的地址——%IX0.2。最后点击“确定”关闭对话框，完成变量的自动声明。程序段 1 的起始线上接入了变量 SB2 的常开触点。

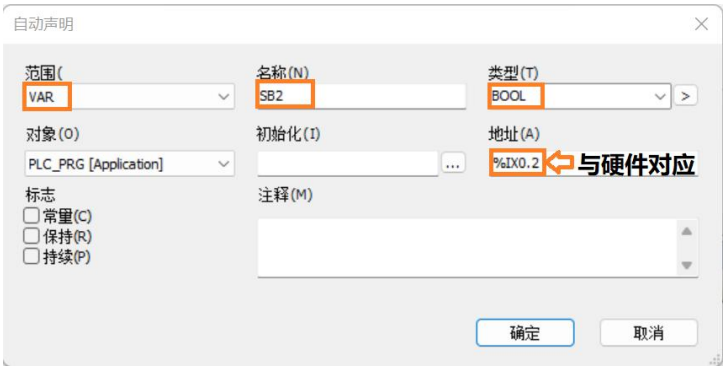


图 1.37 自动声明变量对话框

（说明：用户可查询 DI/DO 模块的 I/O 映射地址，根据模块地址分配情况进行 I/O 变量的地址分配。如图 1.38 所示，在设备树视图中双击模块“RPC3211”，在打开的模块视图中查询模块参数介绍和 I/O 映射情况，单击“Local Bus Interface I/O 映射”，可显示模块每一个通道的地址以及变量类型，此地址与模块所连接的外部信号一一对应，每一个地址对应一位二进制变量，此地址不可修改。关于变量的按位、字节、字和双字的存储及其地址对应关系会在后续章节中详细介绍。）

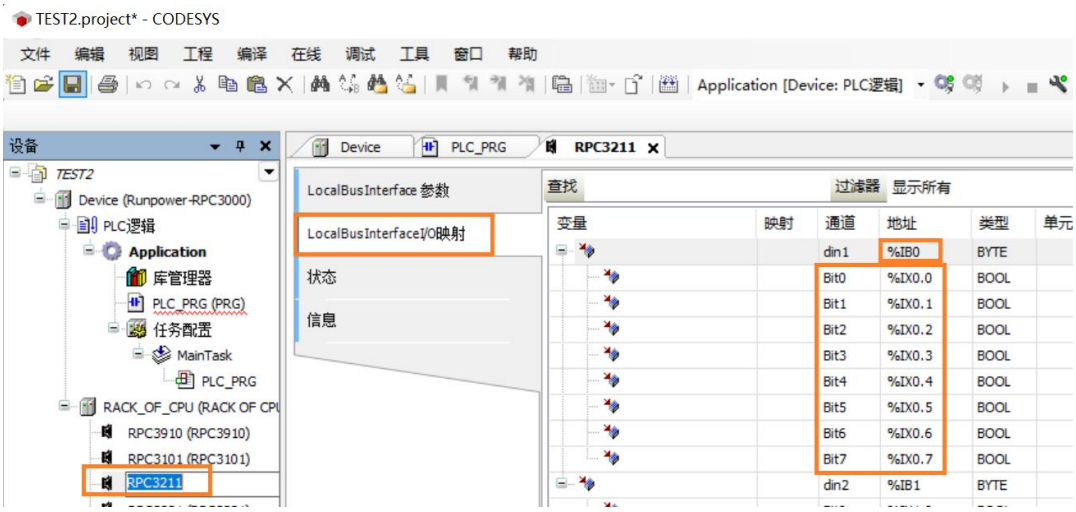


图 1.38 输入模块 I/O 映射情况

继续单击工具栏中的图标，依次输入变量 SB1 的常闭触点 、变量 FR 的常闭触点 、变量 KM 的线圈 和变量 KM 的常开触点 （自锁触点，与 SB2 的常开触点并联），完成图 1.35 梯形图程序的编制。在编程过程中会自动声明变量 SB1、FR、KM，程序中所有变

量都显示在变量声明区，如图 1.39 所示，用户可单击右侧的图标进行变量“文本视图”和“表格视图”切换显示。

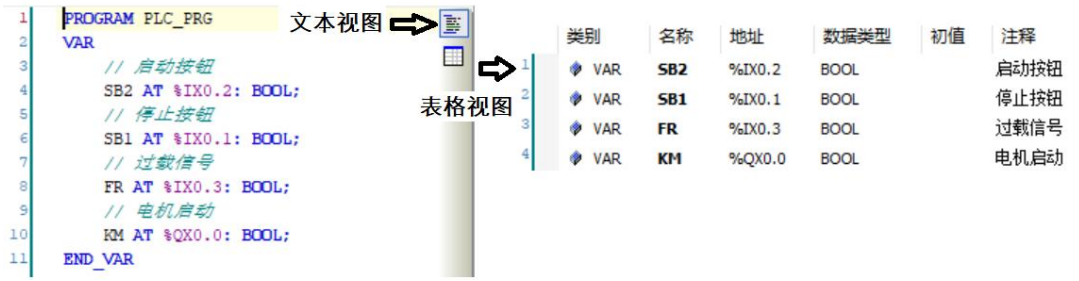


图 1.39 变量声明区显示切换

1.4.2.3 变量分类

相对于常量（数值不变的量）来说，变量是指在程序执行过程中，其数值会发生变化的量。变量可以用来表示一个数值、一个字符串值或一个数组等。在 PLC 的工程中，变量用于存储 PLC 的输入、输出数据，或存储程序运行的中间结果。

按照应用范围不同，变量可分为局部变量和全局变量。局部变量只能在定义该变量的程序中有效，而全局变量则可以在整个工程中被调用。

按照能否掉电保持，变量分为保持型变量和非保持型变量。

按照数据类型，变量可以分为标准类型和用户自定义类型。其中标准类型包括布尔型（BOOL）、整型（INT）、字型（WORD）、实型（REAL）、字符串型（STRING）以及时间型（TIME）等。自定义类型包括结构体（STRUCT）和枚举(ENUM)。

变量一般由字母、数字和下划线组成，必须以一个字母或者单一的下划线开始，随后是一定数量的字母（不区分大小写）、数字或下划线，最后以字母或数字结束。用户自定义的变量名称最好能直观反映该变量的功能，例如 Start_1、Stop_2（可分别表示启动和停止指令）。此外，还可以用中文进行变量名的命名。点击菜单“工程/工程设置/编译选项”，勾选“允许标识符使用 unicode 字符”后，可以进行中文变量名称声明。

1.4.2.4 变量声明

工程中用到的变量必须声明才可以使用，声明方式如下：

- 编程时自动声明：如图 1.37 所示，可在编程时直接输入变量名称，未声明的变量会弹出“自动声明”对话框，用户编辑变量类型和地址等相关信息即可。
- 先声明后使用：用户可在编程界面的变量声明区提前添加变量信息进行变量定义，在编程时直接输入变量名调用即可。

1.4.2.5 全局变量声明

全局变量可以先声明后使用，也可以在编程时自动声明。手动提前声明全局变量需添加“全局变量列表”。右键单击设备树视图中的“Application”，依次选择“添加对象”-“全局变量列表”，在弹出的对话框中输入要定义的全局变量列表名称，例如“GVL1”，完成全局变量列表的添加，如图 1.40、1.41 所示。

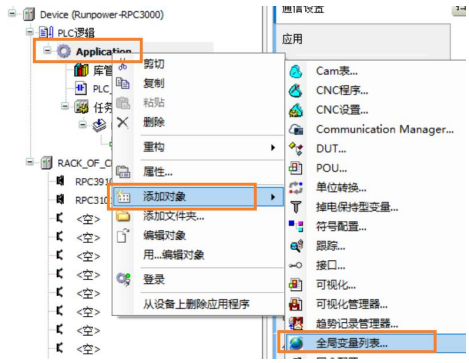


图 1.40 创建全局变量列表（1）

图 1.41 创建全局变量列表（2）

双击设备树视图中“Application”下新建的全局变量列表名称“GVL1”，打开该全局变量列表，进入全局变量声明区。右键单击变量声明区空白处，选择“插入”选项，如图 1.42 所示。

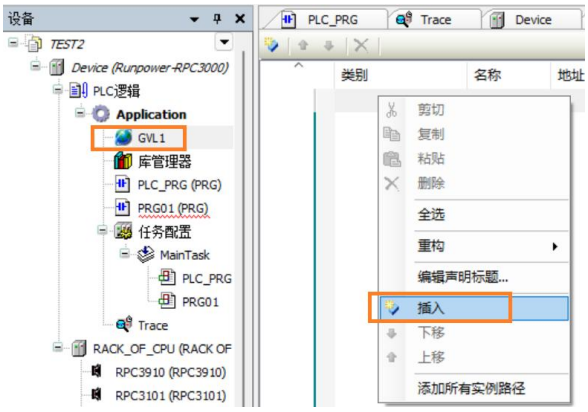


图 1.42 声明全局变量（1）

在插入的变量声明行中，输入需要定义变量的名称、地址、数据类型、初值、注释等选项，如图 1.43 所示。

图 1.43 声明全局变量（2）

双击列表中某一变量的“数据类型”所在列的表格，再依次单击“>”和“输入助手”，在弹出的“输入助手”对话框可选择需要声明的变量类型，如图 1.44 所示。

图 1.44 数据类型选择输入助手

添加完全局变量之后，需要修改某全局变量的名称时，在输入新的变量名称后，会弹出图 1.45 所示的“自动重构：重命名”对话框，单击“确定”按钮即可。

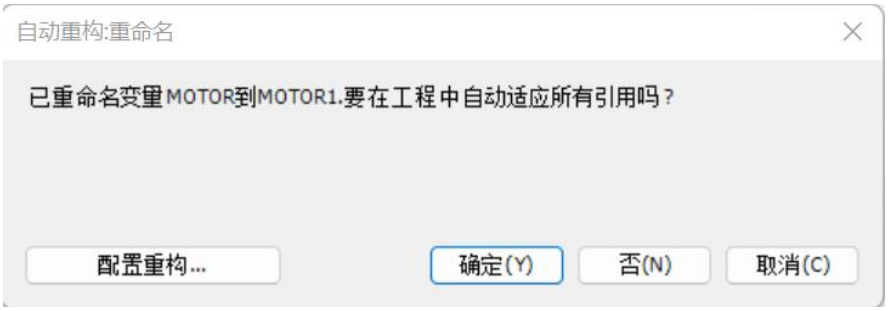


图 1.45 自动重构对话框

全局变量声明之后，在编程时可直接输入变量名称调用全局变量，例如调用图 1.43 中的全局变量，输入变量名称“START1”后，会自动显示为“GVL1.START1”。

1.4.2.6 变量存储区

RPC 系列 PLC 的存储区被划分为不同的存储区域，用来存储不同的变量。数据存储区共分为五类：I 区（输入区）、Q 区（输出区）、M 区（可寻址寄存器区）、N 区（随机寄存器区）、R 区（掉电保持区），如表 1.5 所示。

表 1.5 PLC 的存储区划分

存储区名称	存储数据类型	是否寻址访问	寻址格式	可否赋值与强制	是否可以掉电保持
输入存储区（I）	输入 PLC 的外部信号 DI、AI	是	%IXm.n、%IBm、%IWm、%IDm	是	否
输出存储区（Q）	PLC 输出的外部信号 DO、AO	是	%QXm.n、%QBm、%QWm、%QDm	是	否
M 存储区	中间数据	是	%MXm.n、%MBm、%MWm、%MDm	是	否
N 存储区	中间数据	否	系统自动分配	是	否
R 存储区	掉电保持	否	系统自动分配	是	是

1.4.2.7 地址寻址方式

RPC 系列 PLC 的 I 区、Q 区和 M 区是按地址寻址方式访问的，这些存储区都有唯一的、明确的地址。用户可以通过地址寻址方式，即直接使用该存储区地址的方式，来读取和设置该存储区的值，如图 1.46 所示。

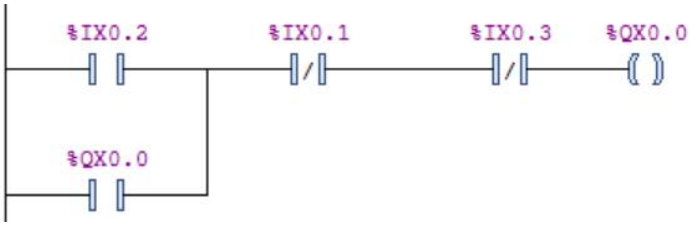


图 1.46 直接地址访问变量

寻址变量的地址格式为“%+内存区范围+数据格式+地址编号”。其中，地址均以 “%”开始，内存区范围包括 I、Q、M，分别表示输入（Input）、输出（Output）和中间（Memory location）存储区。数据格式以 X（位）、B（字节）、W（字）、D（双字）来表示。

I 区、Q 区和 M 区的存储格式是一致的，此处以 M 区为例说明，如表 1.6 和表 1.7 所示。RPC 系列 PLC 所有直接寻址的存储区，是按字节（BYTE）为单位存储的，每 1 个字节包含 8 个二进制位（BIT），每 2 个字节组成 1 个字（WORD-16 位），每 2 个字组成 1 个双字（DWORD-32 位）。存储区通过寻址方式访问，可以按位、字节、字、双字访问。

表 1.6 M 存储区格式 1

位	%MX103.7	.....	%MX103.0	%MX102.7	.....	%MX102.0	%MX101.7	.....	%MX101.0	%MX100.7	.....	%MX100.0
字节	%MB103			%MB102			%MB101			%MB100		
字	%MW51						%MW50					
双字	%MD25											

表 1.7 M 存储区格式 2

位	%MX107.7	.....	%MX107.0	%MX106.7	.....	%MX106.0	%MX105.7	.....	%MX105.0	%MX104.7	.....	%MX104.0
字节	%MB107			%MB106			%MB105			%MB104		
字	%MW53						%MW52					
双字	%MD26											

例如：%MX100.0 表示%MB100 的第 0 位（低位），%MX100.7 表示%MB100 的第 7 位（高位）。

%MB100 表示由%MX100.0～%MX100.7 共八个二进制位组成的字节。

%MW50 表示由%MB100 和%MB101 两个字节组成的字，其中%MB101 为高位。

%MD25 表示由%MW50 和%MW51 两个字组成的双字，其中%MW51 为高位。

这些存储区地址中的 BOOL、BYTE、WORD 和 DWORD 类型的数据，可以采用寻址的方式直接访问，无需定义变量。INT、REAL 等其他数据类型，需要使用变量的方式访问。

需要注意的是，按不同数据类型访问，数据存储区可能是重叠的。诸如%MW50（2#0000 0000 0000 0011）的数值为 3，则%MB100（2#0000 0011）的值也为 3，%MX100.0（2#1）的值为 TRUE，%MX100.1（2#1）的值也为 TRUE。又或者在程序中强制%MX100.0 为 TRUE，因为%MX100.0 是%MB100、%MW50 和%MD25 的第一个位，则同时，这些存储地址的值也被强制为 1。

对于 I 区和 Q 区，也是同样的存储方式。I 区和 Q 区的地址和实际 DI、DO、AI、AO 模块的相应的通道号所连接的信号是一一对应的。

例如在前述章节“输入/输出信号分配”中：

SB2（地址为%IX0.2）——表示输入存储区（I 区）的 1 位二进制数据，此数据按位存取，其值为 0 或 1。该变量的地址编号为 0.2，表示该变量映射到输入模块，通道号为 0.2。

KM（地址为%QX0.0）——表示输出存储区（Q 区）的 1 位二进制数据，此数据按位存取，其值为 0 或 1。该变量的地址编号为 0.0，表示该变量映射到输出模块，通道号为 0.0。

1.5 程序下载

1.5.1 设置 IP 地址

工程文件的下载过程就是建立 PC 机与 PLC 之间直接联系的过程。下载前，先用以太网线将 PLC 与 PC 机连接好，并确保二者处于同一网段。RPC3000 系列 PLC 出厂默认的 IP 地址为“192.168.1.4”，需将编程所用电脑 IP 地址设置为“192.168.1.XXX”，例如：可将 PC 机 IP 地址设置为“192.168.1.11”。

说明：RPC2000 系列 PLC 出厂默认的 IP 地址为“192.168.0.20”，需将编程电脑的 IP 地址设置为“192.168.0.XXX”，例如：192.168.0.11。

1.5.2 建立 PC 机与 PLC 的通讯

CODESYS 启动后，网关服务程序会自动启动。双击设备树中“Device（Runpower-RPC3000/2000）”图标打开设备窗口，网关运行情况如图 1.47 所示，右下角为绿色图标，表示网关已启动。

图 1.47 网关运行情况

在 PLC 与 PC 机联网之前，需在程序中建立 PC 机与 PLC 之间的通讯。以 RPC3000 为例，首先确认 PC 机与 PLC 之间以太网连接正常，然后接通 PLC 电源。打开工程，双击设备树视图“Device (Runpower-RPC3000)”图标，打开设备窗口，如图 1.48 所示，单击“扫描网络”选项卡寻找可用设备。弹出如图 1.49 所示的“选择设备”窗口，在该窗口显示了与 PC 机在同一网段中的设备。单击选中该设备，右侧信息栏会显示该设备的基本信息。单击图 1.49 网关（Gateway-1）下的“MyDevice [0301.5000]”图标，再单击“确定”，或者直接双击该图标完成设备选择。

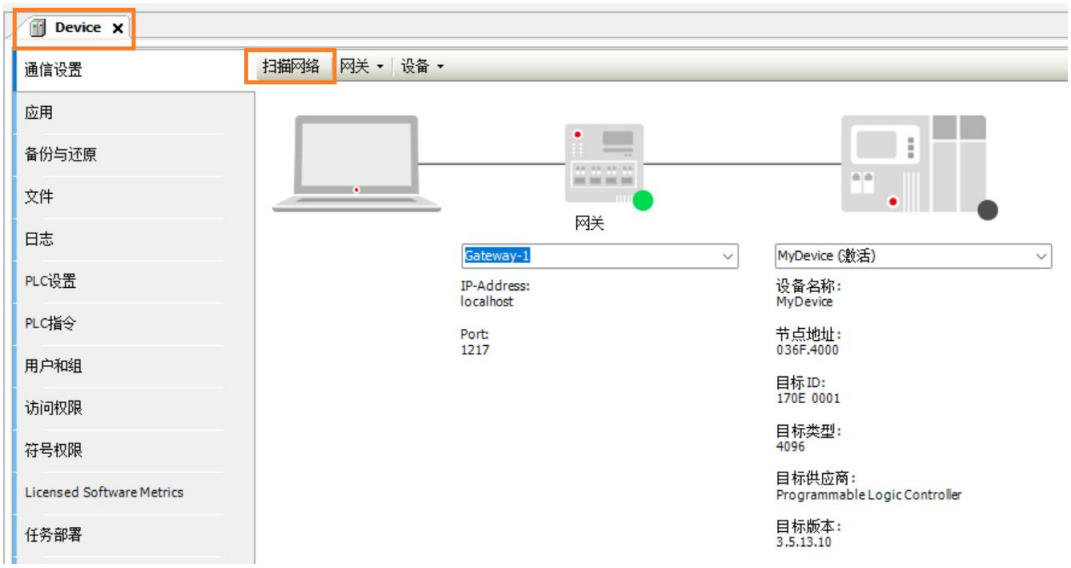


图 1.48 扫描网络

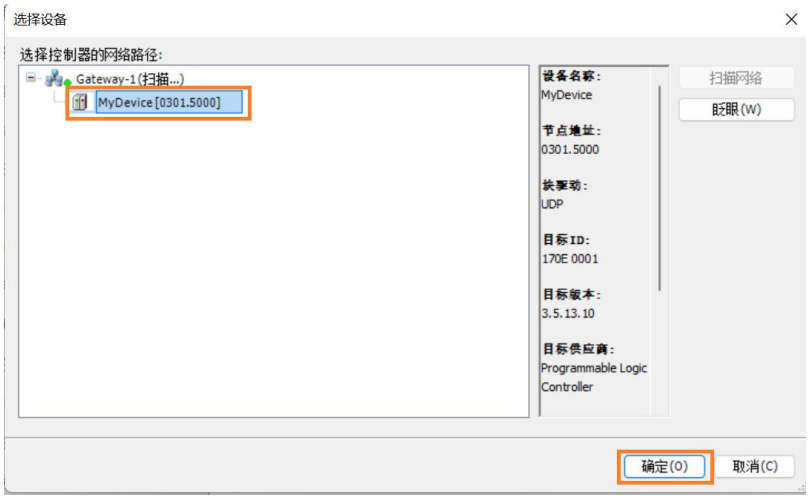


图 1.49 选择设备窗口

这时，设备窗口中“MyDevice（激活）”图标右下角的状态指示灯显示为绿色，表示 PLC 与 PC 机联网正常，如图 1.50 所示。

图 1.50 设备连接成功界面

1.5.3 编译程序

按照前面章节中的控制要求编制好控制程序后，单击工具栏中的“生成代码”图标，对程序进行编译。也可以单击菜单栏的“编译”，选择“生成代码”，或者直接按快捷键 F11，完成编译。编译的错误或者警告信息会在消息框中显示。如出现错误提示，可返回到编程窗口修改程序后再进行编译，直至程序编译正确无误为止。

1.5.4 登录设备和下载程序

登录是指 PC 机与 PLC 联网之后，从 PC 机登录至 PLC 设备，并自动下载程序的过程。单击菜单栏的“在线”，选择“登录”，或者直接按快捷键 Alt+F8，首次登录会弹出如图 1.51 所示对话框。点击“是”，完成在线登录到 PLC 的操作。登录使得 CODESYS 应用程序与目

标设备 PLC 建立起连接，并进入在线状态。如果不能正确登录，请检查设备窗口中设备通信设置是否正确配置、应用程序是否编译无误。登录后，系统会自动选择程序下载。

图 1.51 下载询问对话框（1）

如果用户修改程序后再次登录，会弹出如图 1.52 所示对话框，可根据需要选择登录下载方式进行登录下载。

图 1.52 下载询问对话框（2）

登录后，如图 1.53 所示。设备树视图中“Device”显示为 Device [连接的] (Runpower-RPC3000)，表明 PC 机与 PLC 成功连接，程序成功下载。但是，此时“Application”显示为 **Application [停止]**，表示应用程序尚未启动运行，而是处于停止状态。

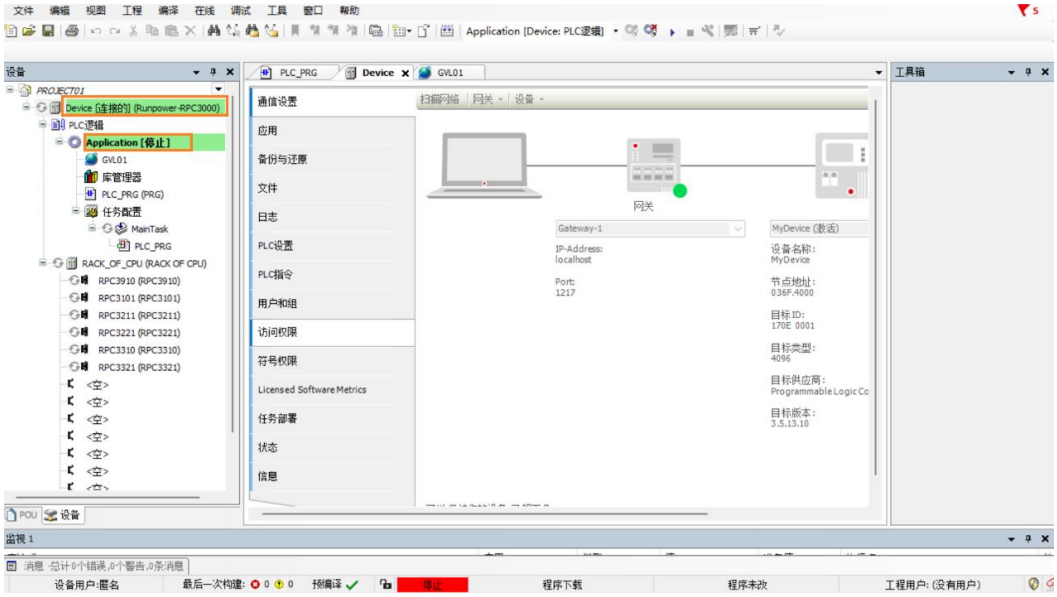


图 1.53 登录到 PLC 设备

1.6 程序调试

1.6.1 启动程序运行

设备登录后，用户可通过以下三种方式启动程序，如图 1.54 所示。①点击菜单栏的“调试”，选择“启动”；②点击工具栏的 图标；③快捷键 F5。

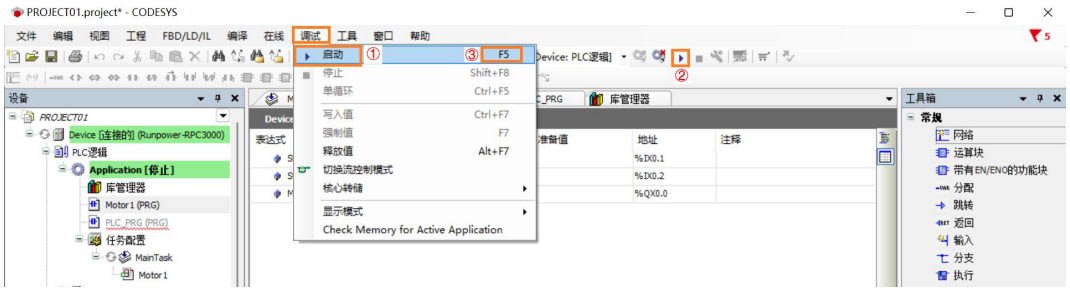


图 1.54 启动程序

程序启动后的界面如图 1.55 所示。设备树中“Application”显示为“[运行]”状态，组态的设备也全部显示为运行状态。双击打开正在运行的程序 PLC_PRG，可以看到梯形图程序也显示为在线运行状态，图中变量 SB2 和 KM 的常开触点显示为断开状态，变量 SB1 和 FR 的常闭触点显示为闭合状态，变量 KM 的输出线圈显示为 0 状态。在程序的变量声明区，可以显示变量当前的值。

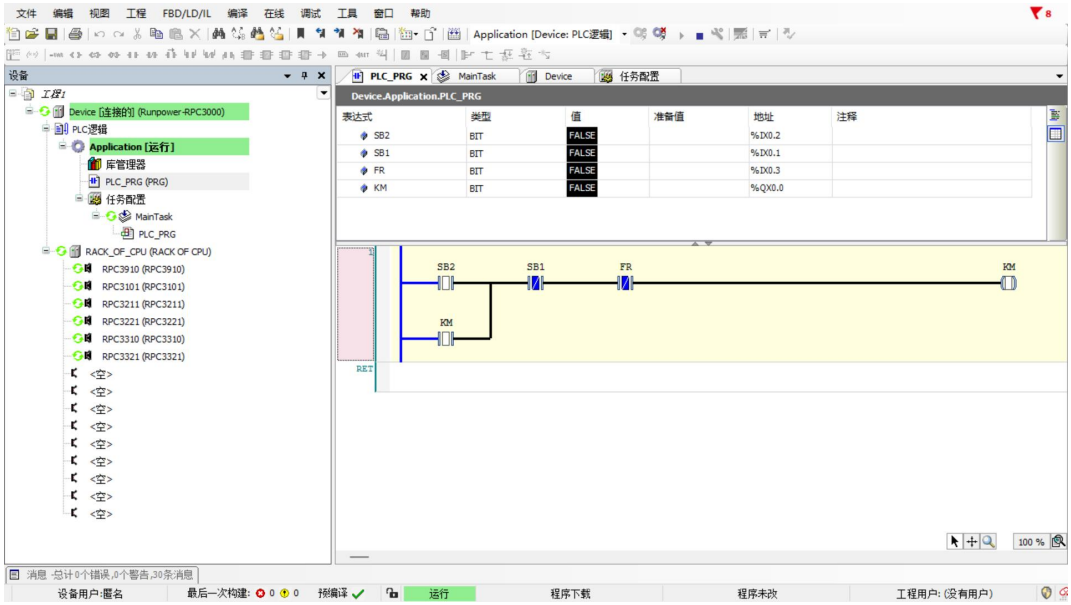


图 1.55 程序在线运行界面

此时，PLC 输入、输出模块的液晶显示屏 **RUN**（显示模块运行状态）和 **IBS**（显示模块总线通讯状态）指示灯点亮，如图 1.56 所示。

图 1.56 硬件运行状态指示

在该例程中，变量 SB1、SB2、FR 和 KM 分别通过地址%IX0.1、%IX0.2、%IX0.3 和%QX0.0 映射到输入模块和输出模块相应通道的信号。在线运行时，需要将外部信号（例如：按钮、传感器等）接入 DI 模块来改变输入存储区变量 SB1、SB2 和 FR 的值。输出存储区变量 KM 的值取决于程序逻辑运算的结果，通过 DO 模块接通输出电路完成控制。关于输入/输出模块的接线方法，将会在后续章节中详细介绍。

在 RPC3000 系列 PLC 控制系统中，接通、断开一次启动按钮 SB2，PLC 输入模块 RPC3211 液晶显示器指示灯 0.2 会闪亮一次，同时程序中变量 SB2 的常开触点也会通-断一次，如图 1.57 所示。表明输入信号已经通过输入模块的 0.2 号端子映射到 PLC 内部相应存储区（I 区的地址）。在程序运行状态下，根据 PLC 逻辑运行结果，变量 KM 的线圈显示图标为，其实时值为 TRUE，同时，PLC 输出模块液晶显示器显示 0.0，表示 PLC 输出通道 Q0.0 所处外部执行机构电路接通，如图 1.58 所示。此时，接通停止按钮 SB1，变量 KM 的线圈值变为 FALSE，输出液晶面板上的指示灯 0.0 熄灭。可以看出，在在线运行状态下，程序中各变量的值会在程序界面和变量定义界面中实时显示，如图 1.59 所示。

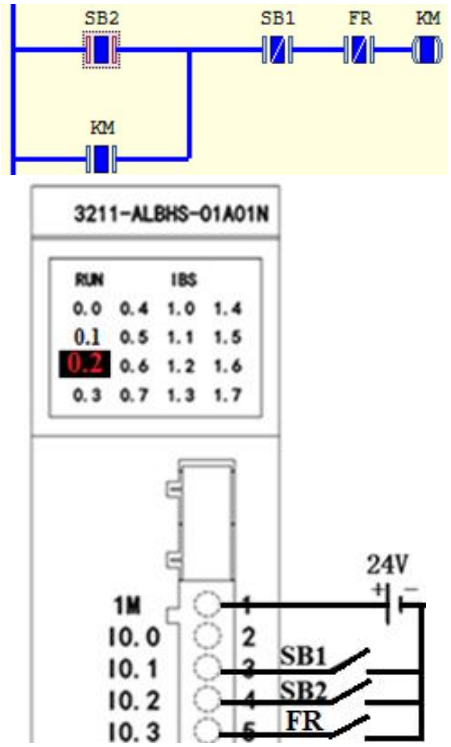


图 1.57 输入信号指示

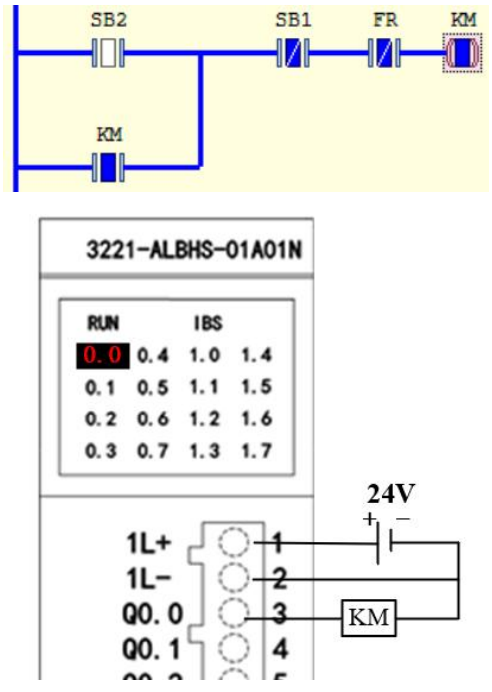


图 1.58 输出信号指示

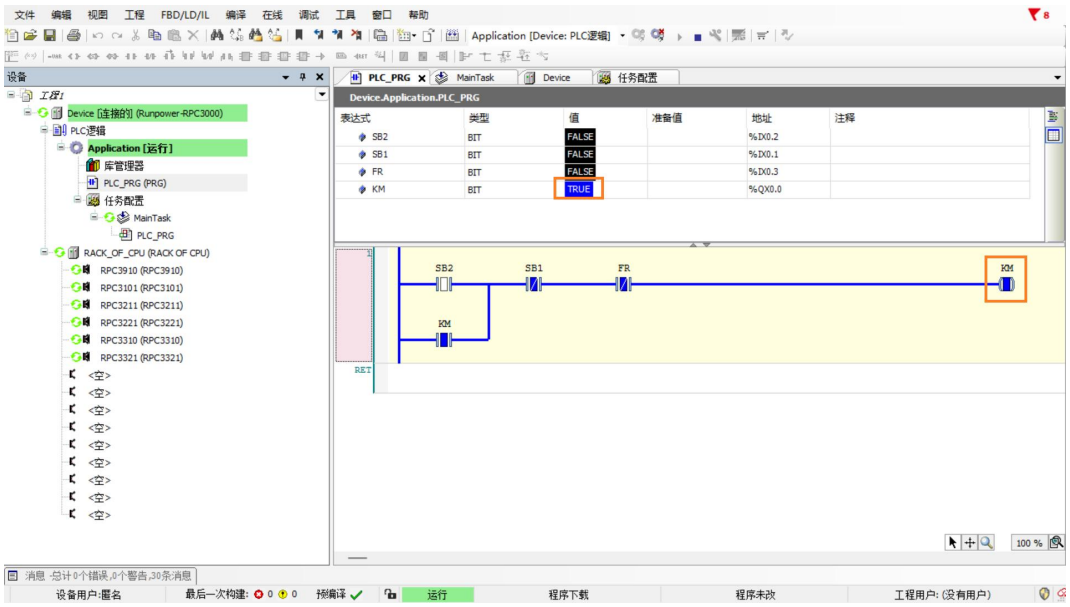


图 1.59 程序运行变量值显示

1.6.2 变量写入与强制

在程序在线调试模式下，用户可通过变量写入或强制来对变量进行赋值，如图 1.60 所示。对于局部变量的赋值，可在程序中双击某一变量，或者在变量声明区某变量的“准备值”处单击，该变量的准备值会进行切换，例如程序中的“SB2”变量，其值会在“TRUE”、“FALSE”和“无”三者之间进行切换，用户可根据调试需要进行赋值选择。

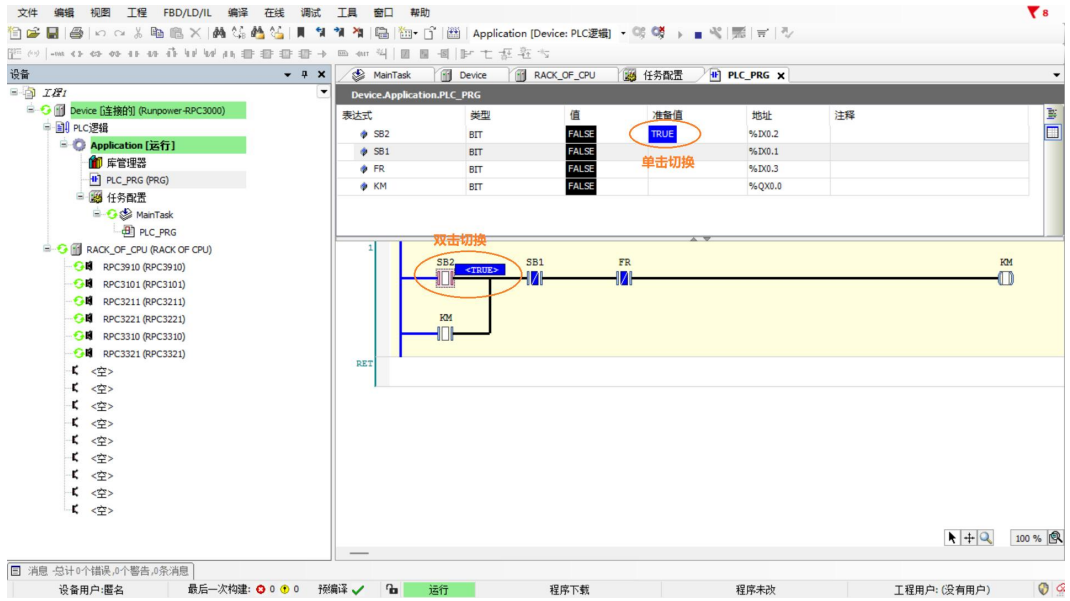


图 1.60 局部变量赋值准备

确定好将要给变量的赋值之后，鼠标右键单击选择“写入‘Device.Application’的所有值”或者“强制‘Device.Application’的所有值”，可将准备好的值写入变量，或者强制赋值给变量，如图 1.61 所示。也可以通过单击菜单“调试”，选择“写入值”（Ctrl+F7），或者选择“强制

值”（F7）。

图 1.61 变量写入/强制/释放操作

变量“强制”之后，在程序变量位置会有F标志，变量声明区该变量值处也会有强制标志“F”，如图 1.62 所示。这里“写入值”与“强制值”的区别在于，进行“写入”操作之后可以继续重新为变量再“写入”新的值，而“强制”之后，则需要先释放掉之前强制的值，才能重新写入新的值。单击菜单“调试/释放值”（或者快捷键 Alt+F7），或者在编程窗口右键单击选择“释放‘Device.Application’的所有值”，可以释放掉已经强制的变量的值。

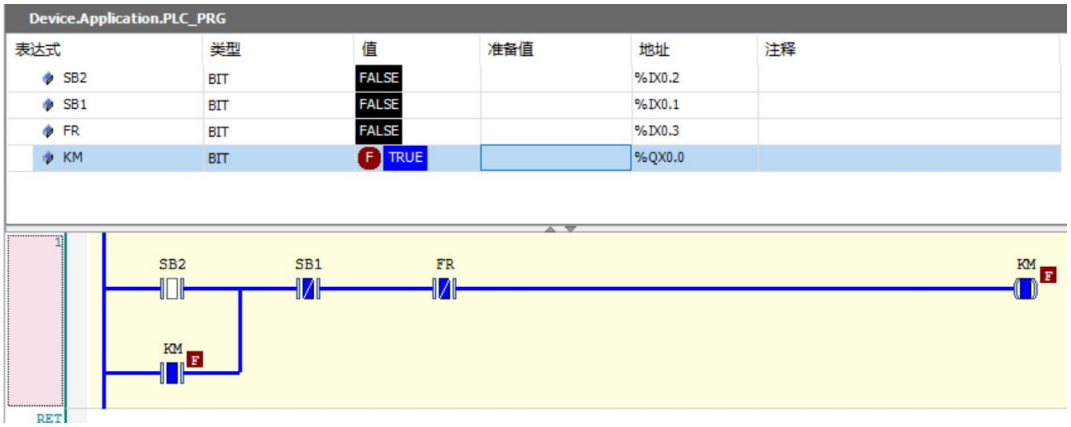


图 1.62 变量强制指示

对于全局变量的赋值可以直接在程序中双击变量进行赋值切换，也可以打开全局变量列表，修改准备值，如图 1.63 所示。赋值操作方法同上。

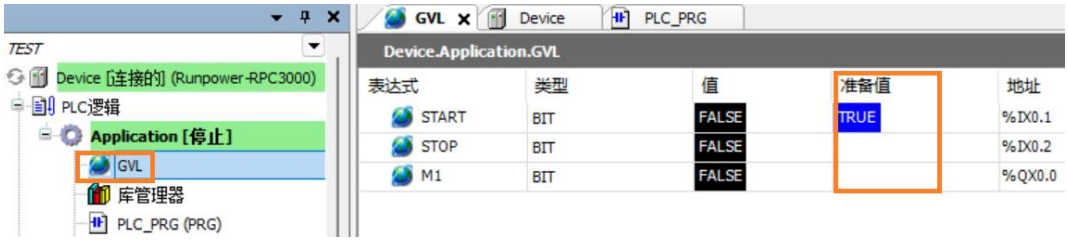


图 1.63 全局变量赋值准备

1.6.3 停止程序

在程序启动状态下，用户可通过以下三种方式停止程序，如图 1.64 所示。①点击菜单栏的“调试”，选择“停止”；②点击工具栏的图标■；③快捷键 Shift+F8。此时程序退回到登录未运行状态。再点击菜单“在线”，选择“退出”，断开 PC 机与 PLC 的连接。

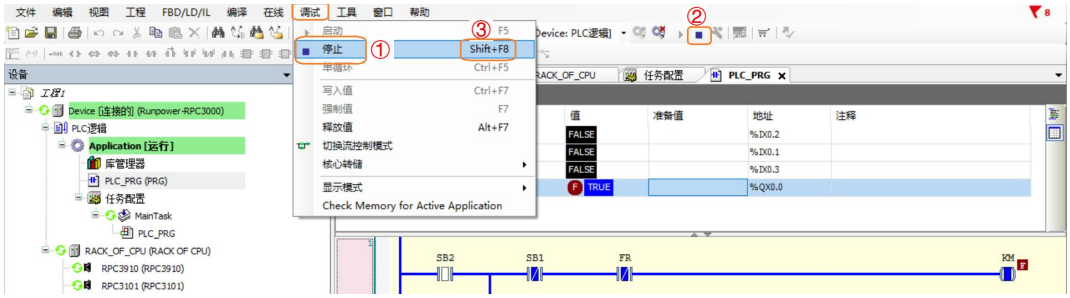


图 1.64 停止程序操作

此外，用户也可以直接单击菜单“在线/退出”，或按快捷键 Ctrl+F8，直接将程序退出登录，如图 1.65 所示。

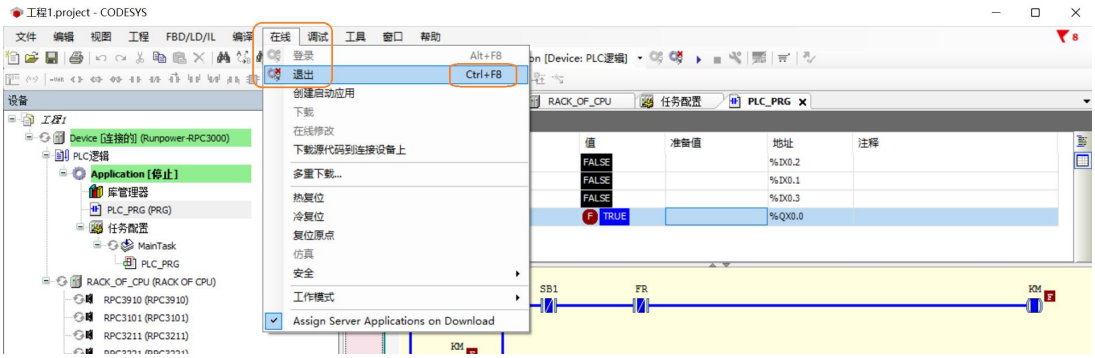


图 1.65 退出登录操作

1.7 程序仿真运行

由于 CODESYS 集成了仿真器功能，用户也可以不连接硬件，直接在 PC 机上进行程序的离线仿真调试，这样可以提高程序的调试效率，缩短开发周期。

1.7.1 设置仿真模式

在工程中单击菜单“在线/仿真”，如图 1.66 所示，“仿真”选项前面会显示已勾选的图标☑。在全屏显示状态下，页面底部也会出现红色“仿真”提示，表示程序进入仿真状态。此时，可直接下载程序至仿真器。

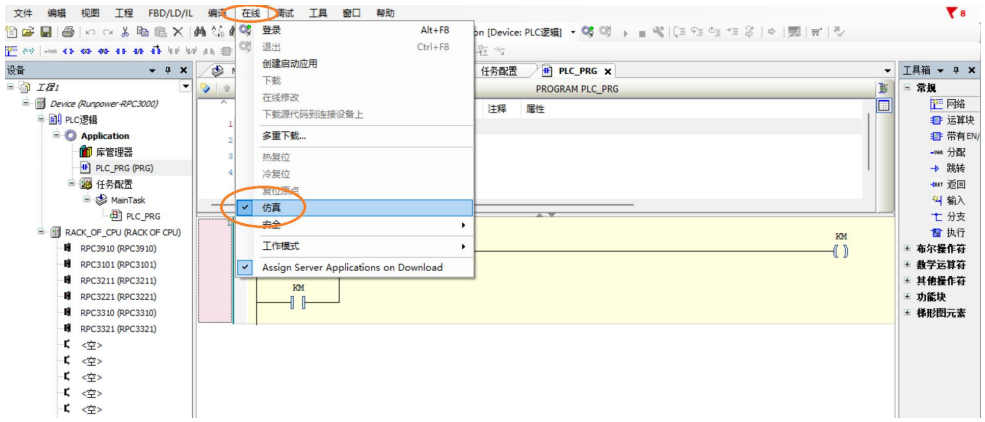


图 1.66 选择仿真模式

单击菜单栏的“在线/登录”（快捷键 Alt+F8），可登录至仿真器。下载程序之前，会弹出如图 1.67 所示对话框，询问是否创建仿真应用程序并下载，单击“是”即可完成下载。单击菜单栏的“在线/退出”（快捷键 Ctrl+F8），可退出仿真器。

图 1.67 下载询问对话框

登录至仿真后程序界面如图 1.68 所示，设备树中“Device”显示为“[连接的]”（表示已连接仿真器），Application 显示为“[停止]”（表示程序尚未运行），底部状态栏也会显示程序当前的状态。

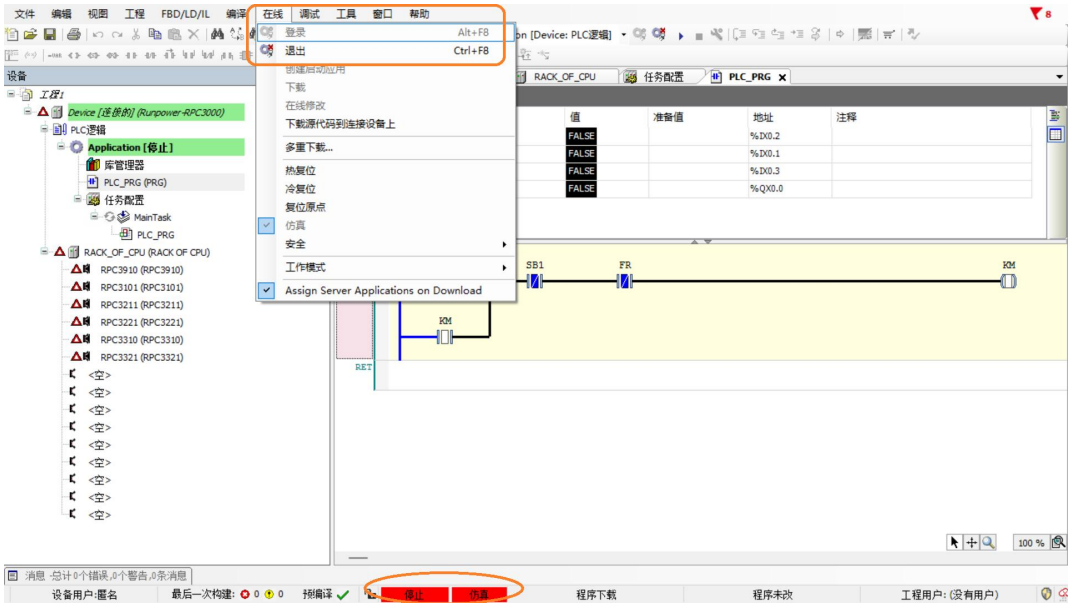


图 1.68 登录至仿真界面

1.7.2 仿真调试

仿真运行与在线运行时程序的调试方法相同，单击菜单“调试/启动”（快捷键 F5），或者单击▶图标，均可启动程序进入仿真运行，仿真运行界面如图 1.69 所示。在编程窗口可以对变量进行写入或强制操作来进行调试。

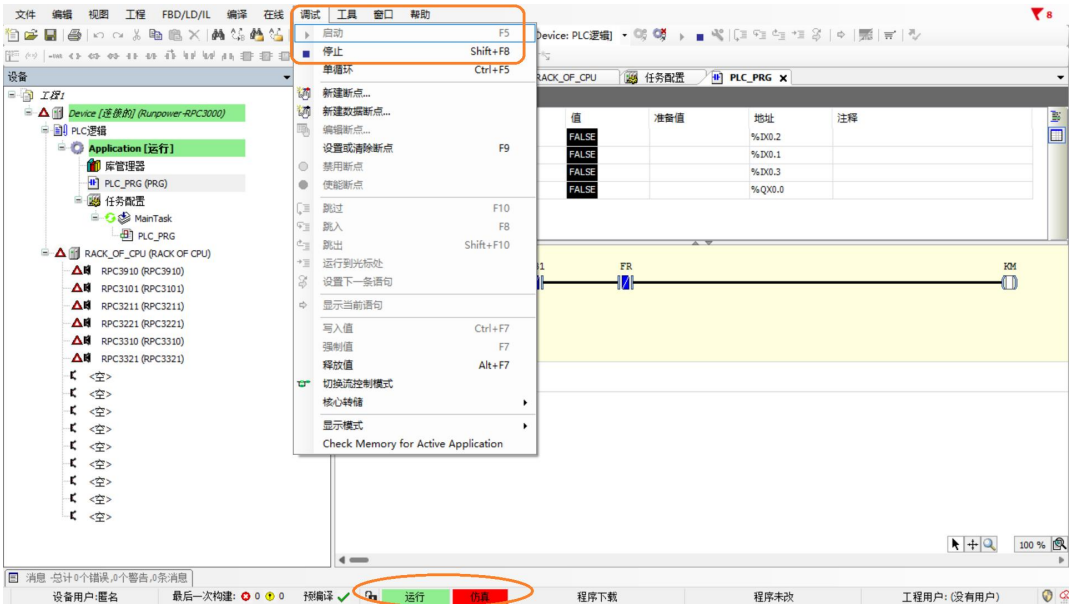


图 1.69 仿真调试程序界面

1.7.3 退出仿真模式

仿真调试结束后，用户需要手动退出仿真模式，否则下次运行该程序时仍然默认为仿真模式。可通过点击菜单“在线/仿真”，单击“仿真”选项前的✔，取消仿真勾选，退出仿真运行模式。

1.8 数据类型

无论是常量还是变量，都必须使用数据类型。数据类型决定了所占用的存储空间及将存储何种类型的值。数据类型的标准化是编程语言开放性的重要标志，RPC 系列 PLC 编程软件的数据类型完全符合 IEC61131-3 标准，将数据类型分为标准数据类型、扩展数据类型和自定义数据类型。

1.8.1 标准数据类型

在编程软件中可以查到所支持的标准数据类型，具体方法如下：选择“编辑”菜单下的“自动声明...”，弹出“自动声明”对话框，单击“类型”输入框后面的下拉箭头“▼”即可。编程软件支持的标准数据类型及范围如表 1.8 所示。

表 1.8 标准数据类型及范围

类型	类型名称	数据下限	数据上限	存储空间	备注
BOOL	布尔型	0	1	1bit	
BYTE	字节型	0	255	8 Bit	
WORD	字型	0	65535	16 Bit	
DWORD	双字型	0	4294967295	32Bit	
SINT	短整型	-128	127	8 Bit	
USINT	无符号短整型	0	255	8 Bit	
INT	整型	-32768	32767	16 Bit	
UINT	无符号整型	0	65535	16 Bit	
DINT	双整型	-2147483648	2147483647	32 Bit	
UDINT	无符号双整型	0	4294967295	32 Bit	
REAL	实数型	-3.402823E+38	3.402823466E+38	32Bit	单精度浮点数
TIME	时间型			32Bit	示例： Time1 : TIME := t#21s;
TOD	时刻型				示例： Tod1：TOD：= TOD#00:00:00；
DATE	日期型				示例： Date1：DATE：= D#2018-11-21;
DT	日期时刻型				示例： DT1 ： DT ： = dt#2018-11-7-3:24:15;
STRING	字符串型				示例： Str:STRING(9):=‘Run’;

1.8.2 指针

指针属于标准的扩展数据类型。所谓指针就是一个地址，用于存放某一变量地址的变量就称为指针变量，指针变量是一类特殊的变量。RPC 系列 PLC 中，所有的存储区都占用 CPU 的地址，这个地址被称为绝对地址。不管是 I 区、Q 区、M 区还是 N 区和 R 区，都占用 CPU 的一部分地址。而 I 区、Q 区和 M 区通过寻址访问的地址，诸如%MW200，可以认为是相对地址。指针是指数据区的绝对地址而不是相对地址。

相邻的两个相对地址，其绝对地址也是相邻的。因为 RPC 系列 PLC 的存储区是按字节

存储的，因此假如%MB200 的绝对地址为 pt，则%MB201 的绝对地址为 pt+1，%MB210 的绝对地址就是 pt+10。

在编程时，利用指针的这种关系，可以很方便地实现一些比较复杂的功能。在使用之前，同样需要定义指针变量。指针变量的定义与其他数据类型定义类似，只是其数据类型为 POINTER TO <数据类型>。

指针定义的语法格式：

<指针名> : POINTER TO <数据类型/功能块>;

例如：

```
pt1:POINTER TO WORD;          (*定义一个字型数据的指针 pt1*)
Var_word1:WORD :=100;         (*定义字型变量 Var_word1，使其等于 100*)
Var_word2:WORD;               (*定义字型变量 Var_word2*)
pt1 := ADR(Var_word1);        (*取出 Var_word1 变量的地址，将地址值赋给 pt1*)
Var_word2:= pt1^;     (*将指针 pt1 所指地址的值赋给 Var_word2，Var_word2=100*)
```

举例说明指针的用法：

在 M 数据区的%MB100 开始的地址中，存放了 100 个 BYTE 型变量，需要将这些值转移至%MB300 开始的 100 个字节内，进行批量赋值。因为数据区比较长，采用指针方式，可以很方便地实现数据区的转移。

这里需要定义两个指针变量 pt1 和 pt2，一个指向%MB100，另一个指向%MB300。这里还需要用到一个取地址指令 ADR 和读数据指令^(读取指针指向的存储区内数据)，关于这两个指令的详细信息，请查看指令说明。

变量定义如下：

```
VAR
    m: INT;
    pt1: POINTER TO BYTE;
    pt2: POINTER TO BYTE;
END_VAR
```

其中变量 m 用于传输 100 个字节。

在这里采用 ST 方式编写程序，关于 ST 语言编程，请参见软件手册。

采用一个 FOR 循环语句，每次循环，pt1 和 pt2 的值均加 1（因为是 BYTE 类型指针），然后将 pt1 的值赋值给 pt2 即可。

具体程序如下：

```
pt1:=ADR(%MB100);
pt2:=ADR(%MB300);
FOR m:=1 TO 100 BY 1 DO
pt2^:=pt1^;
pt1:=pt1+1;
pt2:=pt2+1;
END_FOR
```

1.8.3 数组

RPC 系列 PLC 编程软件对数据的管理功能是非常强大的，不但支持多种数据类型，也支持多维数组。数组属于自定义数据类型，使用数组可以批量处理相同类型的数据，提高数据处理的效率。在处理多个相同数据类型的变量时，可以定义 1 个数组，就不需要分别定义每个变量了。在编程时，可以根据标准数据类型来定义一维、二维和三维数组。数组可以采用自动定义，也可以在变量声明区手动定义。

数组的标识符为 ARRAY。数组定义的语法格式：

<数组名> : ARRAY [<L1>..<U1>, <L2>..<U2>, <L3>..<U3>] OF <基本数据类型>;

其中 L1、L2 和 L3 表示字段范围的最小值，U1、U2 和 U3 表示字段范围的最大值，字段范围必须是整数。若是一维数组，则只需设置 L1 和 U1 即可；若是二维数组，则需要设置 L1、U1 和 L2、U2；若是三维数组，则 L1、U1、L2、U2 和 L3、U3 均需定义。

在数组定义的同时，给数组中的元素赋值称为初始化数组。在数组定义时，可以初始化数组中所有元素，也可以不进行初始化。

举例 1：数组的完全初始化

Arr1:ARRAY [1..7] OF BYTE:=[0,1,2,3,4,5,6];

Arr2:ARRAY [1..2,1..3] OF INT := [1,2,3,3(5)]; （*即 1,2,3,5,5,5 的缩写形式*）

Arr3:ARRAY [1..2,1..2,1..2] OF INT :=[2(0),3(4),1,2,3]; （*即 0,0,4,4,4,1,2,3 的缩写形式*）

举例 2：数组的部分初始化

Arr1:ARRAY [1..10] OF BYTE:= [1,2,3,4,5];

对于那些没有预先赋值的元素，按照标准数据类型的缺省初始值进行初始化。在此例中，元素[6]到[10]被初始化为 0。

1.8.4 结构体

结构体也是一种自定义数据类型。在一些实际应用场合，需要用到一些配方数据。每组配方包含很多参数，而这些参数的数据类型是不同的，诸如可能包含 REAL 型、WORD 型或 TIME 型等，那么使用自定义数据类型就能非常方便地实现配方数据的管理。鼠标右键点击设备树视图中的“Application”，依次选择“添加对象”/“DTU..”，弹出“添加 DTU”对话框，如图 1.70 所示。

图 1.70 添加 DTU

设置结构名称，其命名规则同变量命名规则，在编程时该名称就可以作为一个结构用来表示这个数据类型。类型选择“结构”，点击“打开”，在编辑窗口中会弹出结构编辑窗口，如图 1.71 所示。

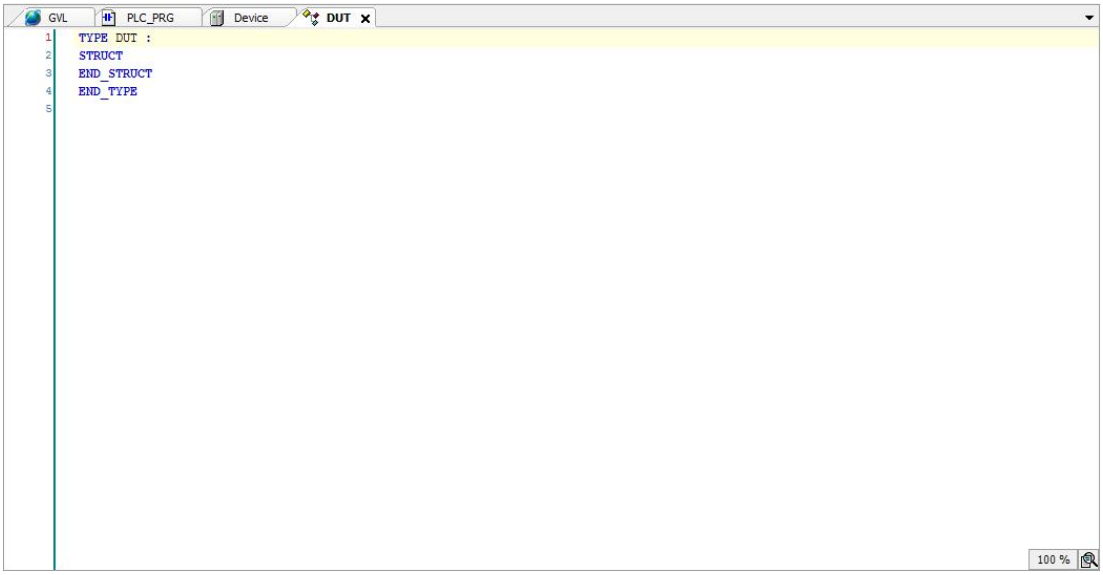


图 1.71 结构编辑窗口

结构变量以关键字 TYPE 和 STRUCT 开始，以关键字 END_STRUCT 和 END_TYPE 结束。

定义结构的语法格式：

TYPE <结构名>:

STRUCT

<变量声明 1>
<变量声明 2>
…
<变量声明 n>
END_STRUCT
END_TYPE

<结构名>是一种可以在整个工程中被识别的数据类型，可以像标准数据类型一样引用，但不可以指定结构中变量的地址（即变量名之后不允许使用“AT”来指定该变量的地址）。举例：

定义名为 Polygonline 的结构：

TYPE Polygonline:

STRUCT

Start:ARRAY [1..2] OF INT;

Point1:ARRAY [1..2] OF INT;

Point2:ARRAY [1..2] OF INT;

Point3:ARRAY [1..2] OF INT;

Point4:ARRAY [1..2] OF INT;

End:ARRAY [1..2] OF INT;

END_STRUCT

END_TYPE

结构体初始化及变量定义：

Poly_1:polygonline:= (Start:=1,2,Point1:=3,4,Point2:=5,6,Point3:=6,5,

Point4:=4,3,End :=2,1);

Start1: INT;

结构体调用：

Start1:=Poly_1.Start[1];

结构成员的访问

<结构名>.<结构成员名>

举例：如果结构名为“Poly_1”，其中一个成员名为“Start”，则可以用下面的形式访问：Poly_1. Start

1.9 指令系统

1.9.1 添加指令库文件

在可编程控制器中，使 CPU 完成某种操作或实现某种功能的命令及多个命令的组合称为指令，指令的集合称为指令系统。指令系统是可编程控制器硬件和软件的桥梁，是可编程

控制器程序设计的基础。RPC 系列 PLC 编程软件提供了大量的可供用户调用的指令，用户可通过查阅本手册“附录 A”（指令速查表），来了解指令的详情。在 1.2.2.3 节中安装指令库文件之后，初次使用指令时，需提前添加相应的指令库到工程文件的库管理器中。

在设备树视图中，双击 库管理器，可查看已添加的指令库文件，如图 1.72 所示。单击“添加库”和“删除库”可以添加和删除指令库文件。

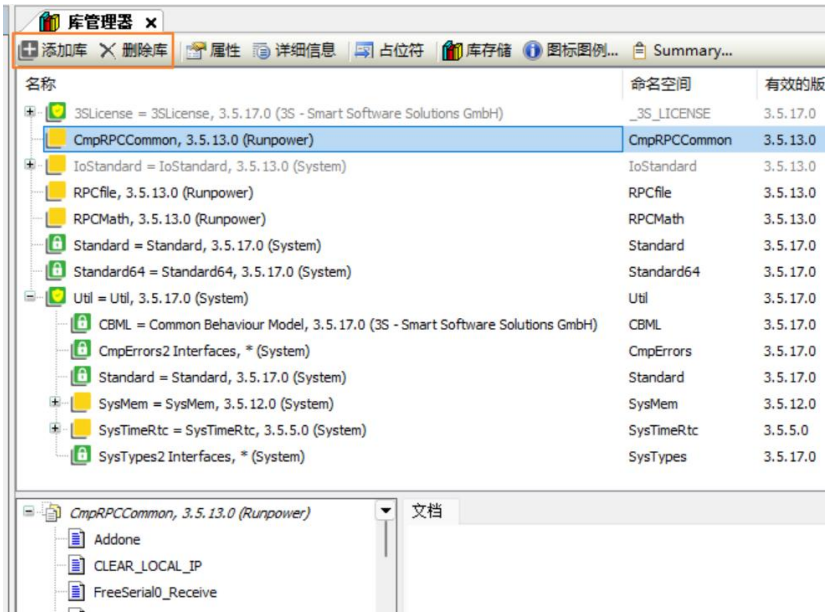


图 1.72 查看已添加的指令库文件

如图 1.73 所示，单击“添加库”后，弹出“添加库”对话框，用户可根据需要选择指令库文件，单击“确定”即可添加到应用程序的库管理器中。

图 1.73 添加指令库至应用程序

1.9.2 位逻辑指令

■ 常开触点与常闭触点

梯形图元素常开触点（符号：）在寄存器（软继电器线圈）状态为 1 时闭合，为 0 时断开。常闭触点（符号：）与常开触点相反。

梯形图编程可通过单击编程窗口的工具栏图标或者在右侧工具箱窗口，展开“梯形图元素”拖动触点至编程界面的连接点处。

■ 输出线圈

在梯形图程序中，线圈（符号：）位于一行程序的最右侧，输出的是输入程序的逻辑运算结果。线圈输出指令将线圈的状态写入软继电器，线圈通电时写入 1，断电时写入 0。如果是映射到 Q 区域的输出变量，CPU 将输出的值传送给对应的过程映像输出。

梯形图编程可通过单击编程窗口的工具栏图标或者在右侧工具箱窗口，展开“梯形图元素”拖动线圈至编程界面的连接点处。

■ 置位和复位线圈

置位线圈（S）（Set，置位或置 1）将指定的变量置位（变为 1 状态并保持）；复位线圈（R）（Reset，复位或置 0）将指定的变量复位（变为 0 状态并保持）。

图 1.74 置位/复位线圈

在图 1.74 的程序段中，当置位指令的输入变量 M1 由状态 0 变为 1 时，置位指令将输出寄存器 Q0 接通为 ON 并保持。当复位指令的输入变量 M2 由状态 0 变为 1 时，复位指令将输出寄存器 Q0 断开并保持。

梯形图编程可通过在右侧工具箱窗口，展开“功能块”拖动相应的功能块至编程界面的连接点处。

■ 边沿检测（触发器）指令

触发器功能块输入有三种方式：（1）在程序编辑区需要插入功能块的位置右键单击后选择 插入运算块；（2）在工具栏选择“插入运算块”图标 ；（3）在右侧工具箱展开“常规”选项，将 运算块 拖入指定位置后，单击功能块内部的 。以上三种操作后会出现如图 1.75 所示的输入助手窗口。在 Standard（标准指令库）中可选择 Trigger（触发器）。

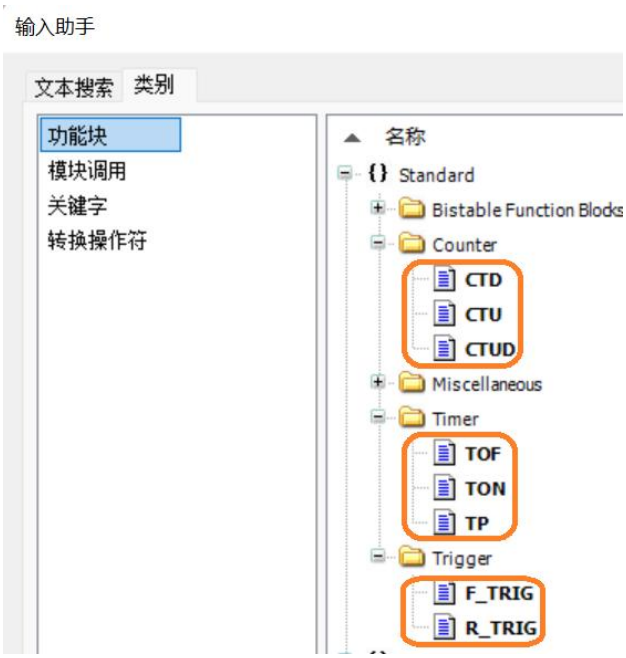


图 1.75 功能块选择输入助手

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），如果检测信号出现上升或下降变化则相应检测指令输出为 1，并且接通一个扫描周期。

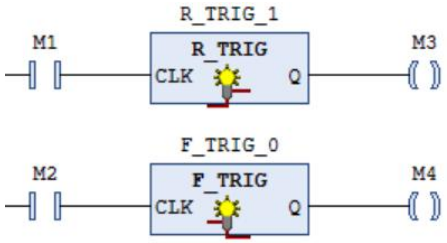


图 1.76 边沿检测指令

R_TRIG 用于检测上升沿，在图 1.76 的程序段中，功能块 R_TRIG_1 用于检测变量 M1 由 0→1 的变化。当功能块的 CLK 端检测到上升沿时，其 Q 端输出 1，保持一个扫描周期后变为 0；如果 CLK 端状态未发生变化，Q 输出一直为 0。

F_TRIG 用于检测下降沿，功能块 F_TRIG_0 用于检测变量 M2 由 1→0 的变化。当功能块的 CLK 端检测到下降沿时，其 Q 端输出 1，保持一个扫描周期后变为 0；如果 CLK 端状态未发生变化，Q 输出一直为 0。

1.9.3 定时器指令

定时器功能块输入方法同上，如图 1.75，在 Standard（标准指令库）中可选择 Timer（定时器）。定时器分为脉冲定时器（TP）、通电延时定时器（TON）和断电延时定时器（TOF）。

■ 脉冲定时器（TP）

如图 1.77 所示。IN 为定时器输入端，PT 为定时器设定端，用来设定定时时间。当 IN 输入端变量 M1 状态从 0 变为 1 时，启动定时器，在定时时间内，无论输入端如何变化，输出 Q 都保持为 1，已耗时间由 ET 显示。ET 时间达到 PT 设定的时间后，输出 Q 变为 0，

ET 显示定时时间值。当 IN 输入端变量 M1 状态为 0 时，定时器为复位，已耗时间被清零。在启动输入 IN 检测到新的信号上升沿时，该定时器功能将再次启动。

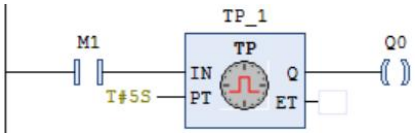


图 1.77 脉冲定时器指令

■ 通电延时定时器（TON）

如图 1.78 所示，IN 为定时器输入端，PT 为定时器设定端，用来设定定时时间。当 IN 输入端变量 M1 状态从 0 变为 1 时，启动定时器，开始计时，已耗时间由 ET 显示。当 ET 等于 PT 的设定值时，输出 Q 变为 1 状态，只要输入 IN 仍为“1”，输出 Q 就保持置位。当 IN 输入端的变量 M1 从 1 变为 0 时，定时器复位，ET 被清零，输出 Q 变为 0 状态。

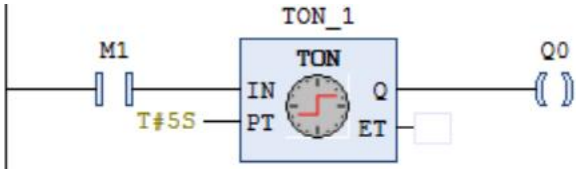


图 1.78 通电延时定时器指令

■ 断电延时定时器（TOF）

如图 1.79 所示，IN 为定时器输入端，PT 为定时器设定端，用来设定定时时间。当 IN 输入端变量 M1 状态从 0 变为 1 时，输出 Q 变为 1 状态。 当 IN 输入端变量 M1 状态从 1 变为 0 时，启动定时器，开始计时，已耗时间由 ET 显示。当 ET 等于 PT 的设定值时，输出 Q 变为 0 状态。

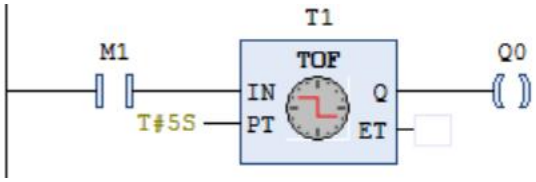


图 1.79 断电延时定时器指令

1.9.4 计数器指令

计数器功能块输入方法同上，如图 1.75，在 Standard（标准指令库）中可选择 Counter（计数器）。计数器指令分为减计数器（CTD）、增计数器（CTU）和增减计数器（CTUD）三类。

■ 减计数器（CTD）

图 1.80 所示为减计数器指令程序段。CTD 为减计数器，包括 3 个输入端，2 个输出端。LOAD 为计数器的复位端，PV 为计数设定值，CD 为输入端，CV 端显示计数当前值，Q 为输出端。所谓减计数，就是当输入端 CD 的状态从 0 变为 1 时，CV 端的值会减 1，直至减为 0 时，输出端 Q 的状态才变为 1 的计数方式。首次调用时，需先将 CTD 计数器复位，即使 LOAD 端信号由 0→1→0，功能块才生效，CV 端的初始值显示为 PV 端设定的值。

当变量 M1 出现一次上升沿时，CV 端的值就会减 1，当 CD 端出现的上升沿次数达到

计数预置值 PV 设定的次数（本例中为 5 次）时，CV 端变量的值减为 0，输出端 Q 保持为 1 状态，直到复位信号 LOAD 端的输入变量 M2 由 0→1 时，输出端 Q 复位为 0，CV 端变量的值也重新被预置了与 PV 相等的值。只要 LOAD 端的变量 M2 值为 1，输入端的上升沿就不起作用。

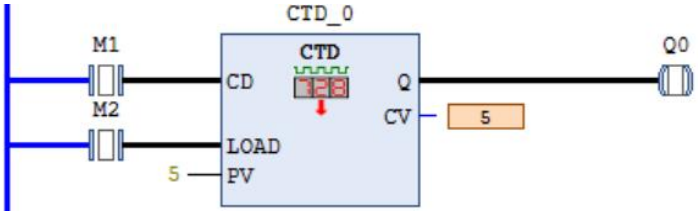


图 1.80 减计数器指令

■ 增计数器（CTU）

图 1.81 所示为增计数器指令程序段。CTU 为增计数器，包括 3 个输入端，2 个输出端。RESET 为增计数器的复位端，PV 为计数设定值，CU 为输入端，CV 端变量显示计数当前值，Q 为输出端。所谓增计数，就是当输入端 CU 状态从 0 变为 1 时，CV 的值从 0 开始加 1，直至等于 PV 的值时，输出端 Q 状态变为 1。首次调用时，复位端 RESET 状态为 0，功能块才生效，CV 的初始值为 0，Q 端初始状态为 0。

当输入端的变量 M1 出现一次上升沿时，CV 的值就会加 1，当 CU 端出现的上升沿次数达到计数预置值 PV 设定的次数（本例中为 3 次）时，Q 端状态变为 1。只要输入端出现上升沿，CV 会一直增加直至溢出，输出 Q 将保持为 1 状态，直到复位信号 RESET 的输入变量 M2 由 0→1 时，输出端 Q 复位为 0，CV 也恢复为 0。只要 RESET 端的变量 M2 的状态为 1，输入端的上升沿就不起作用。

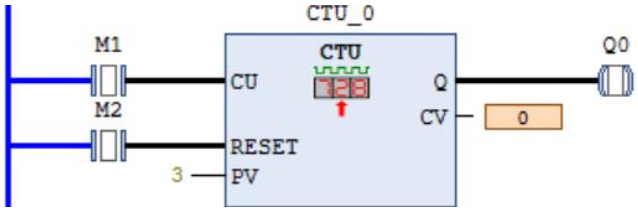


图 1.81 增计数器指令

■ 增减计数器（CTUD）

图 1.82 所示为增减计数器指令程序段。CTUD 为增减计数器，包括 5 个输入端，3 个输出端。RESET 和 LOAD 分别为增、减计数的复位端，PV 为计数设定值，CU 和 CD 分别为增、减计数的输入端，CV 端变量显示计数当前值，QU、QD 分别为增、减计数器输出端。所谓增减计数，就是当输入端 CU 或 CD 出现上升沿（从 0 变为 1）时，CV 的值会分别随之增加 1 和减小 1，当 CU 和 CD 端同时出现上升沿时，CV 的值不变。当 CV 的值大于等于 PV 的设定值时，QU 端状态为 1；当 CV 的值减为 0 时，QD 输出为 1 状态。当增计数复位信号 RESET 由 0→1 时，CV 端的值被复位为 0，QD 输出为 1；当减计数复位信号 LOAD 由 0→1 时，CV 端的值被复位为减计数的设定值，即 PV 的值，此时，输出端 QU 为 1。

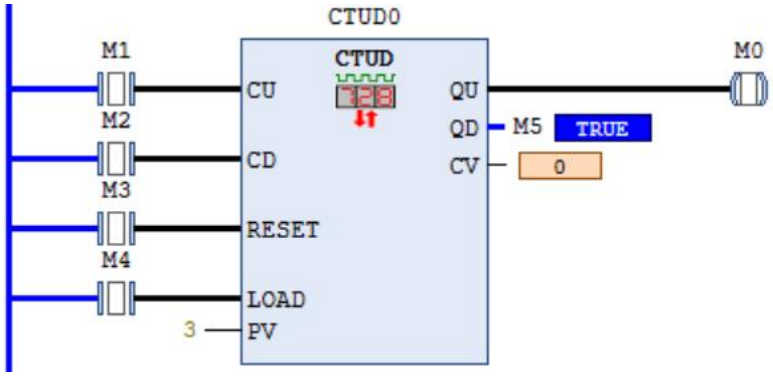


图 1.82 增减计数器指令

1.9.5 比较指令

比较指令用于比较两个相同数据类型操作数的大小，当比较的结果为“真”时，输出的 BOOL 型数据为 TRUE，否则为 FALSE。比较指令包括：等于（EQ）、不等于（NE）、大于（GT）、大于等于（GE）、小于（LT）、小于等于（LE）。两个输入的操作数可以为常量或变量，数据类型可以为 BYTE、DATE、DATE_AND_TIME(DT)、DINT、DWORD、INT、REAL、SINT、STRING、BOOL、TIME、TIME_OF_DAY(TOD)、UDINT、UINT、USINT、WORD、WSTRING。

梯形图编程可单击编程窗口右侧工具箱，展开“数学运算符”调用比较指令功能块。

1.9.6 运算指令

- 赋值指令（MOVE）

  赋值指令 MOVE 用于将一个常量或者变量的值赋给另外一个变量。

  梯形图编程可单击编程窗口右侧工具箱，展开“其他操作符”调用赋值指令功能块。

- 算术运算

算术运算包括加（ADD）、减（SUB）、乘（MUL）、除（DIV）和取余（MOD）指令，用于对输入的变量进行简单的数学运算。梯形图编程可单击编程窗口右侧工具箱，展开“数学运算符”调用 ADD、SUB、MUL、DIV 功能块，也可以通过单击工具栏“插入运算块”图标，在输入助手对话框查找并选择相应的关键字（例如：MOD），添加术运算指令。

1.9.7 移位指令

移位指令分为按位移动和循环移动两类。其功能是将操作数按移位指令的规定移动 n 位，将结果送入返回值。输入和输出的数据类型为 BYTE、INT、WORD、DWORD、SINT、UINT、DINT、UDINT、USINT。

梯形图编程可在右侧工具箱展开“常规”选项，将 运算块 拖入指定位置后，单击功能块内部的，或者单击工具栏“插入运算块”图标，在输入助手对话框查找并选择相应的关键字（例如：SHL 等），添加移位指令块。

■ 左移指令（SHL）

左移指令用于对操作数进行按位左移，右边空缺位自动补 0，不用处理左边移出的位。

■ 循环左移指令（ROL）

循环左移指令是对操作数进行按位循环左移，左边移出的最高位依次送到右边最低位。

■ 右移指令（SHR）

右移指令是对操作数进行按位右移，对于无符号数据类型，左边空缺位自动补 0，不用处理右边移出的位。对于有符号数据类型，如 INT 类型数据，每右移一位，左侧会补充一位原来的符号位，不用处理右边移出的位。

■ 循环右移指令（ROR）

循环右移指令是对操作数进行按位循环右移，右边移出的最低位依次送到左边的最高位。

图 1.83 为移位程序举例。程序将 2 个二进制数据常量分别赋值给字节型变量 MB100 和 MB101。变量 MW50（声明为 WORD 类型变量）由 MB101 和 MB100 构成，移位前的值为 2#1000001010001001。经过左移指令 SHL 左移 2 位后，结果存入变量 W01（WORD 型），其值为 2#0000101000100100。经过 ROL 指令循环左移 2 位结果存入变量 W02（WORD 型），其值为 2#0000101000100110。

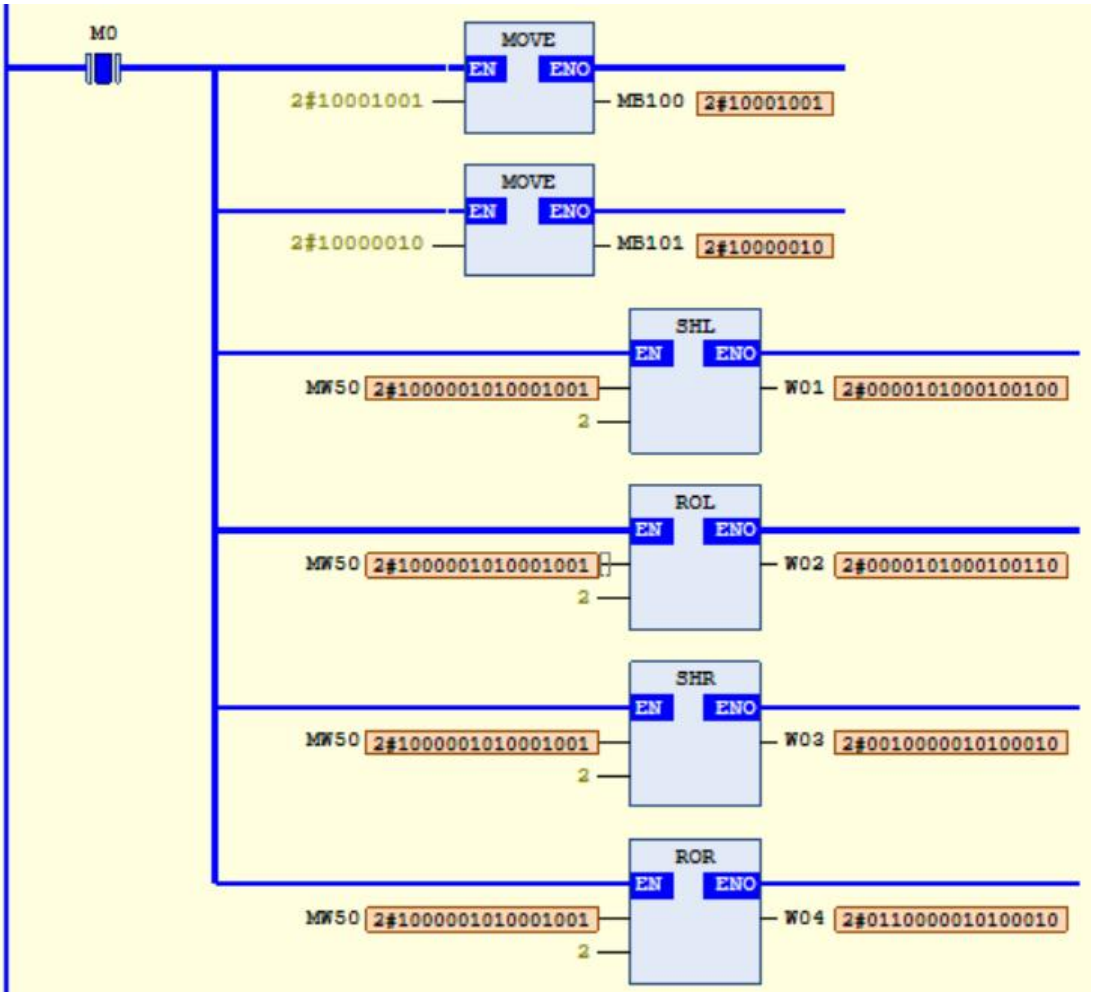


图 1.83 无符号数据移位指令程序举例

在执行右移指令时， MW50（WORD 型变量）为无符号数据类型，数据向右移动之后最左侧会补充 0，经过 SHR 指令右移 2 位结果存入变量 W03（WORD 型），其值为

2#0010000010100010。经过 ROR 循环右移 2 位结果存入变量 W04（WORD 型），其值为 2#0110000010100010。

类别	名称	地址	数据类型
VAR	**M0**		BOOL
VAR	**MB100**	%MB100	BYTE
VAR	**MB101**	%MB101	BYTE
VAR	**MW50**	%MW50	WORD
VAR	**W01**		WORD
VAR	**W02**		WORD
VAR	**W03**		WORD
VAR	**W04**		WORD

图 1.84 移位程序变量定义

如果将变量 MW50 定义为 INT（有符号数据类型，最高位为符号位）型变量，在执行右移指令时，数据每向右移动一位最左侧会补充进一位相应的符号位，则经过 SHR 指令右移 2 位后，MW50（2#1000001010001001）变为 2#1110000010100010。向右移动 2 位，最左侧补充了 2 个符号位“1”，如图 1.85 所示。

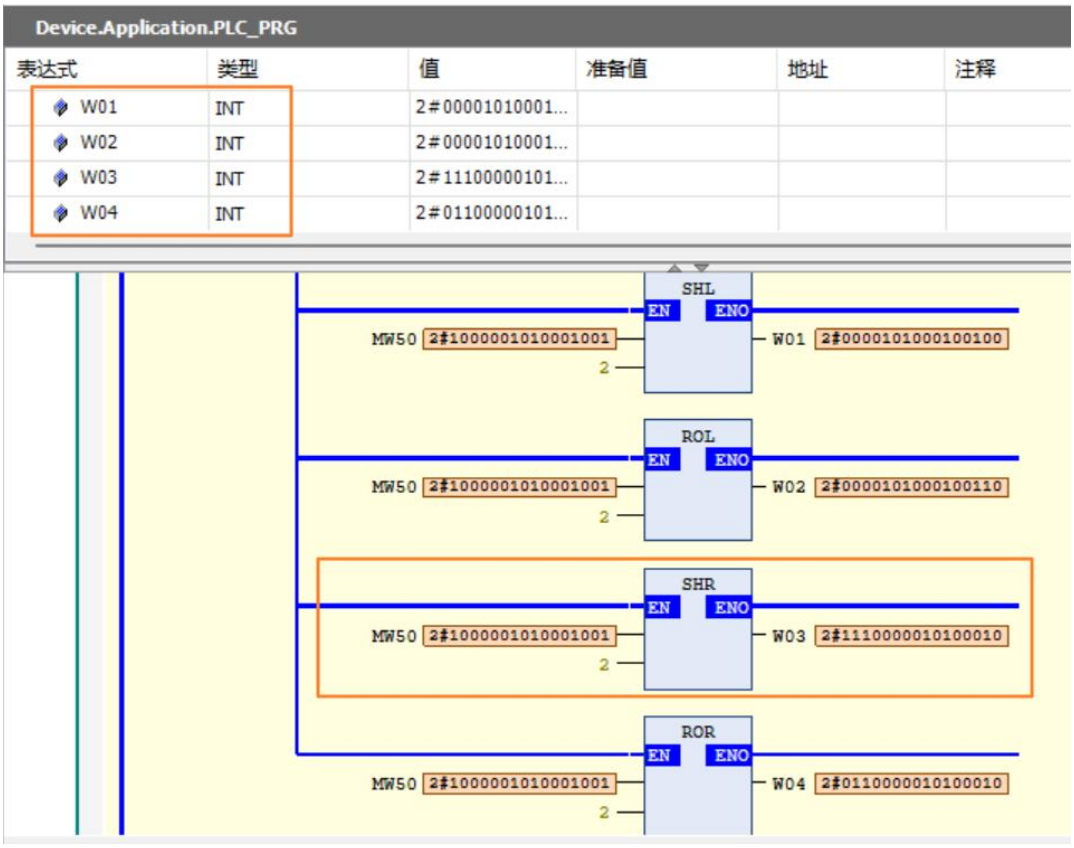


图 1.85 有符号数据移位指令程序举例

2. 硬件介绍

目前，蓝普锋公司自主研制的主流 RPC 系列 PLC 产品包括 RPC2000 和 RPC3000。其中 RPC2000 系列 PLC 是蓝普锋公司在多年 PLC 行业应用和产品设计、开发经验积累的基础上，自主研发、自主生产的一款高性能、高品质的 PLC 产品。产品广泛用于各类工业和民用领域，如：电力、煤炭、石油、环保、节能、市政、交通、机械、空调、供水、地铁、热网等，是设备配套和小型自动化工程的首选控制产品。RPC2000 系列 PLC 硬件分为 CPU 模块和扩展模块，模块均采用导轨式安装，接线端子可插拔。模块具有良好的环境适应性，电磁兼容性好，抗干扰能力强。

RPC3000 系列大型 PLC 充分融合先进、实用的计算机、信号处理、嵌入式软件、通讯、自动控制等技术，在产品性能、可靠性等方面达到了国际先进水平。该系列 PLC 具有性能优异、稳定可靠、功能丰富、集成度高、扩展性好、体积小巧、易于使用等特点，可为不同工业领域提供个性化的解决方案，广泛应用于市政、交通、环保、电力、煤炭、水利、农业等多种行业。相对于传统 PLC 而言，RPC3000 系列 PLC 充分融合了 PLC 和 DCS 的优点，结合了工厂自动化和过程自动化的应用需求，既体现了 PLC 标准化、集成化、开放化、离散控制速度快、成本相对较低的特点，又综合了 DCS 强大的模拟量控制功能、冗余、热插拔、高可靠性等优点。RPC3000 系列 PLC 的硬件系统包含控制器模块、I/O 模块、通讯模块、电源模块和背板等。所有模块均支持热插拔，在背板上的安装位置可以自由选择，无槽位限制。

前述章节中，表 1.1 和表 1.2 分别列出了部分 RPC3000/2000 系列 PLC 模块。

2.1 硬件选型

用户可根据控制规模（I/O 点数）和应用场景要求，选择适合的 CPU 及其扩展模块组建控制系统。蓝普锋公司基于多年总线设计经验，按照不同应用场景设计了四种不同类型的总线结构，分别为 CC 总线（柜内总线）、CH 总线（高速总线）、CE 总线（实时远程总线）和 CR 总线（通用远程总线）。蓝普锋工程师将会根据用户需求从可靠性、安全性、经济性等多角度为用户设计完美的解决方案。详细信息可参考硬件选型表，其中详细介绍了 RPC2000/3000 系列 PLC 选型的多种技术方案。

2.2 I/O 信号接线方法

2.2.1 RPC3000 系列模块接线方法

本手册所述例程选用的 RPC3000 系列 PLC 构成的控制系统如图 2.1 所示。各模块固定在背板上，背板如图 2.2 所示。在背板上每个插槽均配置有通讯接口，所有模块直接插在插槽上即可组成控制系统。用户只需完成输入、输出模块的外部信号接线即可。

在控制系统中，CPU 选择了 RPC3101，RUN 为运行指示灯，BUS 为通讯指示灯，BAT 为电池状态指示灯，FORCE 为强制状态指示灯，FAULT 为故障指示灯。RJ45 接口可用来

与触摸屏、上位机、云服务器等通讯，支持 MODBUS TCP 协议/自由协议，允许 8 个 client 同时链接。RS485 通讯接口可用来与触摸屏、上位机、第三方仪表等通讯，支持 MODBUS RTU 协议/自由协议。

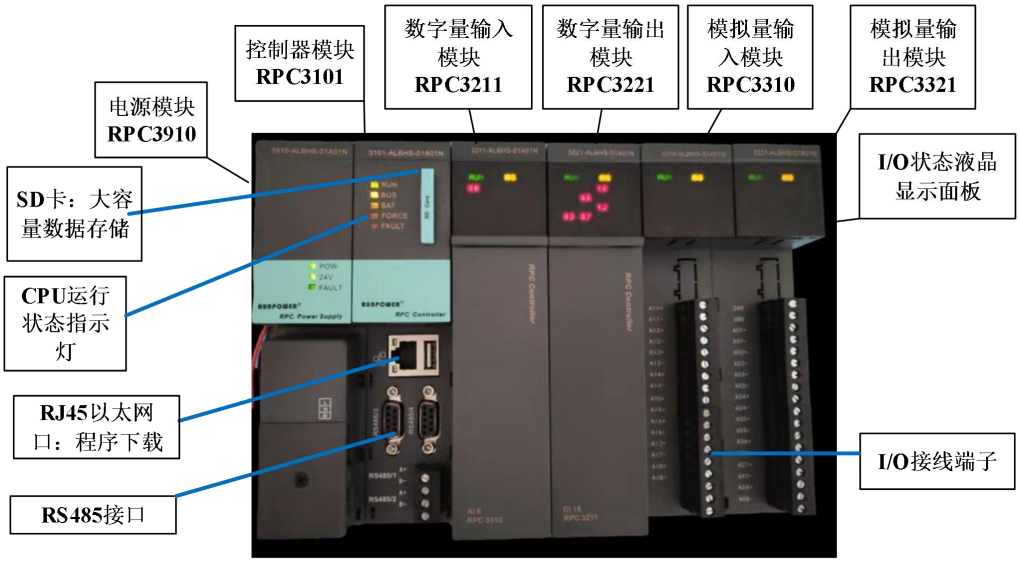


图 2.1 RPC3000 控制系统硬件构成

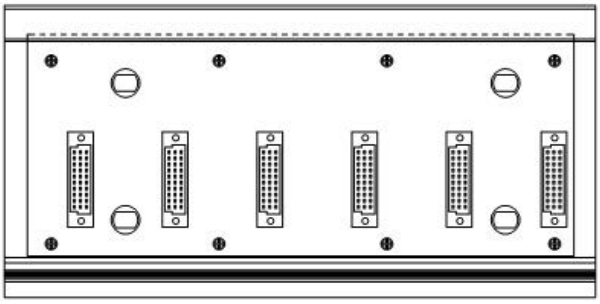


图 2.2 PLC 背板

2.2.1.1 DI 信号接线

RPC3211 集成 16 路数字量输入通道，输入信号的额定电压为 DC24V。PLC 的输入信号一般为位置开关、传感器、手动开关等，用于控制 PLC 内部软继电器的线圈，输入信号可以为常开点或常闭点，具体接线方式如图 2.3 所示。其中，1M 为 IX0.0~IX0.7 的公共端，2M 为 IX1.0~IX1.7 的公共端，漏型输入公共端需接 24V 电源正极，源型输入公共端需接 24V 电源负极。

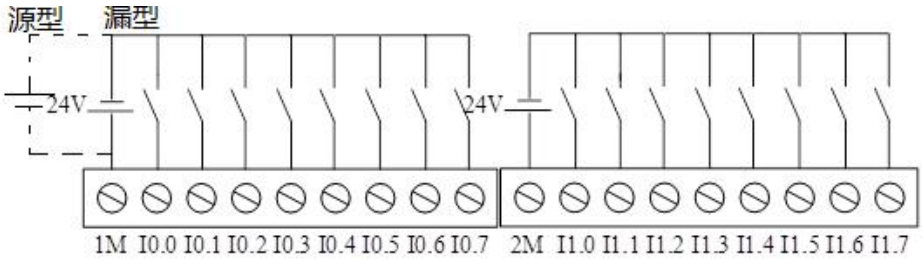
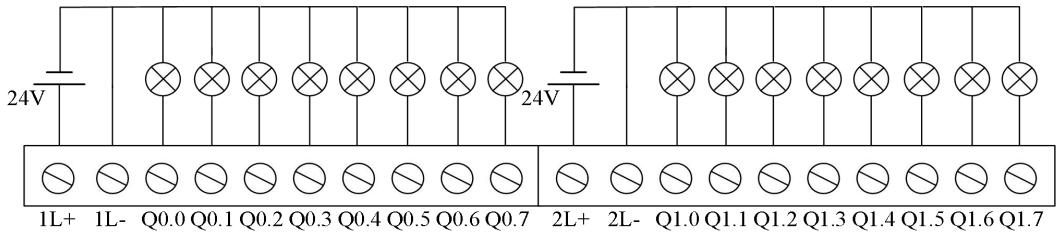


图 2.3 模块 RPC3211 接线示意图

2.2.1.2 DO 信号接线

RPC3221 集成 16 路晶体管输出通道，输出信号的额定电压为 DC24V。PLC 的输出信号一般为继电器线圈、电磁铁、指示灯、蜂鸣器等，具体接线方式如图 2.4 所示。其中，1L+、1L-为 Q0.0~Q0.7 的电源，2L+、2L-为 Q1.0~Q1.7 的电源，需分别接 24V 电源正、负极。



2.2.1.3 AI 信号接线

RPC3310 集成 8 路差分模拟量输入通道，各通道间隔离，用于检测模拟量输入信号 4～20mA、0～20mA 或 0～10V，接线方式如图 2.5 所示。

2.2.1.4 AO 信号接线

RPC3321 集成 8 路模拟量输出通道，用于输出模拟量信号 0～20mA 或 0～5V，接线方式如图 2.6 所示。

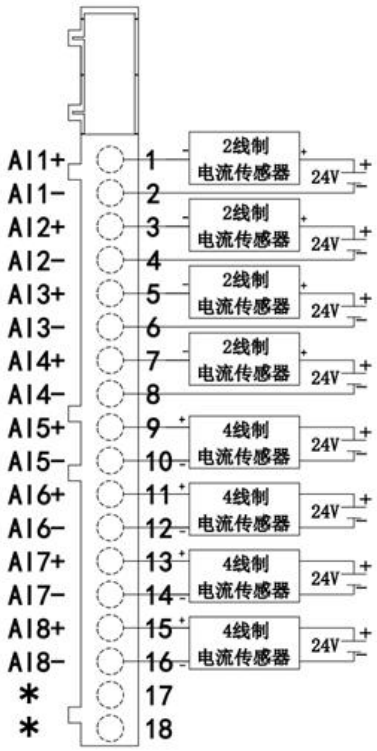


图 2.5 模块 RPC3310 接线示意图

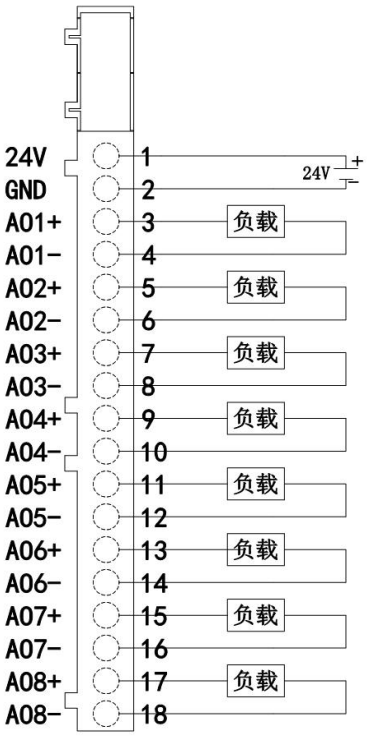


图 2.6 模块 RPC3321 接线示意图

2.2.2 RPC2000 系列模块接线方法

2.2.2.1 RPC2117N 接线说明

RPC2117N 为集成 16 点 I/O（10 路 DC24V 输入通道及 6 路继电器输出通道）的 RPC2000 系列 PLC 的 CPU 模块。模块外观如图 2.7 所示，其输入信号和输出信号接线分别如图 2.8 和图 2.9 所示。

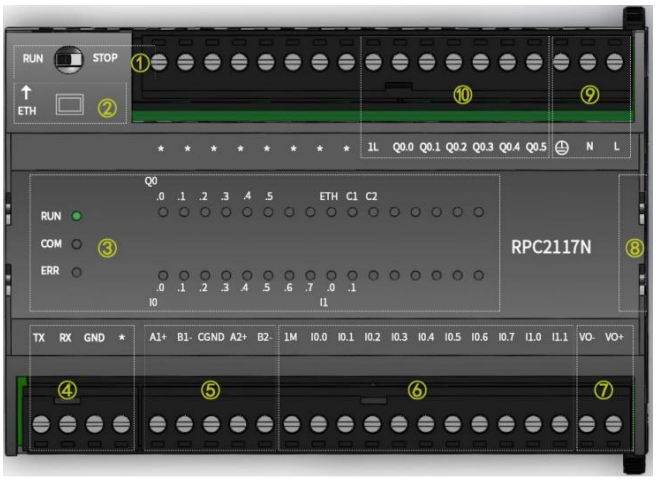


图 2.7 RPC2117N 外观图

① 拨码开关：RUN、STOP 分别用于控制 PLC 程序运行、停止；

② 以太网通讯接口：RJ45 接口为程序下载接口，默认 IP 地址为 192.168.0.20，此接口可用来与触摸屏、上位机、云服务器等通讯，支持 MODBUS TCP 协议；

③ 指示灯：RUN、COM、ERR 分别用于显示模块运行、RS232 通讯、故障情况，I0、Q0 处的指示灯分别用于指示各通道输入、输出状态，ETH 、C1、C2 分别用于指示以太网接口、两个 RS485 接口通讯状态（闪烁表示对应接口有数据收发）；

④ RS232 通讯接口：可用来与触摸屏、上位机等通讯，支持 MODBUS RTU 主从站/自由协议；

⑤ RS485 通讯接口：2 个接口可用来与触摸屏、上位机、第三方仪表等通讯，支持 MODBUS RTU 主从站/自由协议；

⑥ 模块输入端子：由公共端 1M 及输入点 I 构成，可采用源型/漏型接法；

⑦ 模块外供 DC24V 电源。VO+、VO-分别为 DC24V 的正、负接线端子；

⑧ 扩展接口：用于扩展 RPC2000 系列扩展模块；

⑨ 模块供电电源：L、N、⏚分别为 AC220V 电源的火线、零线、保护接地接线端子；

⑩ 模块输出端子：由电源驱动端 1L 及输出点 Q 构成，端子为继电器输出。

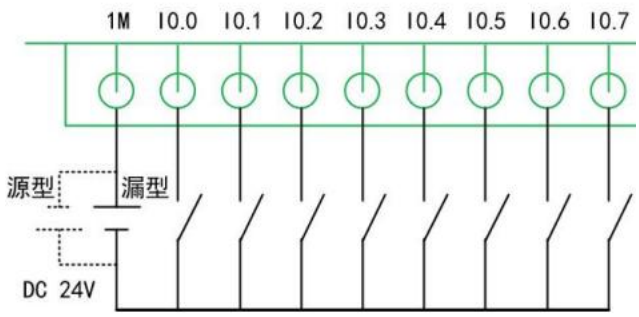


图 2.8 RPC2117N 输入通道接线方式

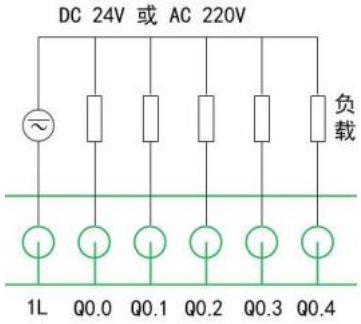


图 2.9 RPC2117N 输出通道接线方式

2.2.2.2 RPC2330 硬件说明

RPC2330 集成 4 路模拟量输入通道（4~20mA、 0~20mA 或 0~10V）及 1 路模拟量输出通道（0~20mA 或 0~10V）。模块外观如图 2.10 所示。

① 4 组模拟量信号输入端子。例如：RA、A+、A-为 1 组模拟量输入端子，当输入为电压信号时，A+、A-分别连接电压信号的输入端；当输入为电流信号时，需要将 RA 与 A+短接，作为电流信号的流入端，A-作为电流信号的流出端。其余各组（B、C、D）与 A 组相同。

② 24V+和 24V-为模块模拟量输出电源供电接入端子。V0、I0、M0 为一组模拟量输出端子，M0 为公共端，输出电压信号时通过 V0 与 M0，V0 为电压正极；输出电流信号时，通过 I0 与 M0，I0 为电流流出，“*”表示通道无实际物理连接。

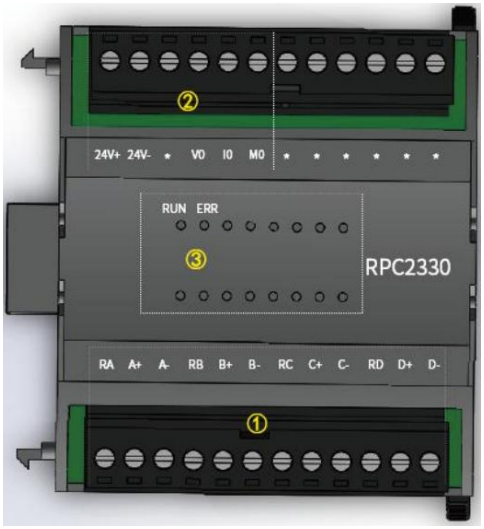


图 2.10 RPC2330 外观图

3. PLC 控制工程实例

用 PLC 实现控制的一般步骤为：

- 确定控制方案，选择合适的 PLC 设备（电源、CPU、输入/输出模块、通讯模块等）；
- 确定输入、输出信号，进行 I/O 信号分配；
- 完成 I/O 信号接线；
- 根据硬件选型进行设备组态；
- 编制控制程序；
- 登录下载进行在线调试或仿真调试。

3.1 用 PLC 实现运料装置自动往返控制

在工业生产中，经常需要实现正、反两个方向的运动控制，例如：机床主轴的正、反转，电梯的升、降运动，工作台的前、后移动等。在继电-接触器控制电路中，通过控制两个接触器主触点的接通来控制接入电机三相电源的相序，从而实现控制电机正反转。其控制电路的控制逻辑部分可通过 PLC 编程来实现。

3.1.1 控制工艺要求及原理图

运料装置自动往返控制是电机正反转控制的一种常见应用，如图 3.1 所示。生产实践要求运料小车将加工好的工件从一端送到另一端，通过控制三相异步电动机的正、反转来实现小车自动前往装料和卸料处并进行装、卸料工作。

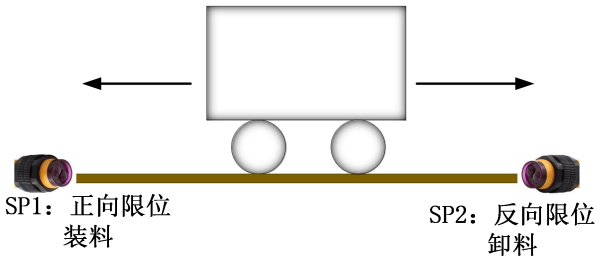


图 3.1 运料小车往返控制示意图

电气控制原理图如图 3.2 所示。小车正反转分别由交流接触器 KM_1 和 KM_2 控制，熔断器 FU 用于短路保护，热继电器 FR 用于过载保护。KA 为中间继电器，KT 为时间继电器，SP 为接近开关，SB 为按钮开关。

电路工作过程如下：（以小车向左行驶为例）

按启动按钮 SB_2，KM_1 线圈接通，KM_1 主触点闭合，小车电机接通三相电源（相序为 U、V、W）开始正转（向左行驶）。当小车向左行驶到 SP_1（装料）位置处，SP_1 常开点闭合，KA_1 线圈通电，KA_1 常闭点断开，小车停止运行，开始装料。同时，中间继电器 KA_1 常开点闭合，时间继电器 KT_1 线圈通电，进行装料计时，计时时间结束（小车装料结束），KT_1 常开点闭合，使得 KM_2 线圈带电, KM_2 主触点闭合，小车电机接通三相电源（相序为 W、V、U）开始反转（向右行驶）。运行至 SP_2（卸料）处，SP_2 常开点闭合，KA_2 线圈通电，KA_2

常闭点断开，小车停止运行，开始卸料。同时时间继电器 KT_2 线圈通电开始计时，计时时间结束，KT_2 常开点闭合，KM_1 线圈通电，小车卸料结束自动开始向左运行，如此往复。KM_1、KM_2 的常闭触点为互锁点，可防止两个接触器线圈同时通电发生两相短路事故。

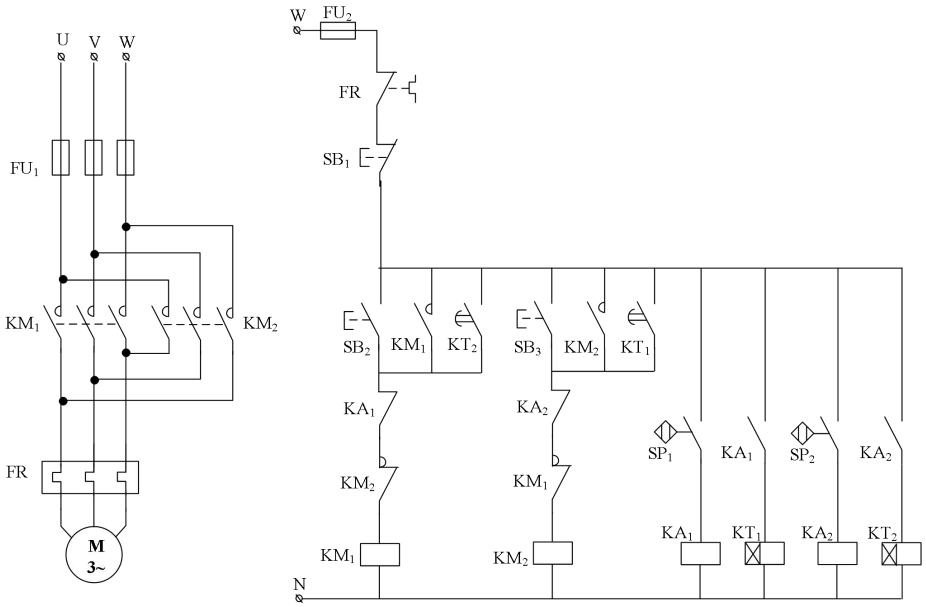


图 3.2 运料小车自动往返控制电气原理图

3.1.2 PLC 选型及 I/O 分配

用 PLC 代替继电-接触器控制系统实现电机正反转控制，主要是采用 PLC 梯形图程序代替图 3.2 右侧的控制电路部分。根据控制要求，选用表 1.3 中列出的 RPC3000 系列模块可完成该控制任务，I/O 信号分配见表 3.1。（**说明：若采用 RPC2000 系列 PLC，I/O 信号分配见表 3.2**）

表 3.1RPC3000 系列 PLC I/O 分配表

输入信号	停止	正转启动	反转启动	正向限位	反向限位	过载保护
名称	SB1	SB2	SB3	SP1	SP2	FR
DI 端子号	I0.1	I0.3	I0.5	I1.1	I1.3	I1.5
输出信号	电机正转	电机反转	-	-	-	-
DO 端子号	Q0.1	Q0.3	-	-	-	-

表 3.2 RPC2000 系列 PLC I/O 分配表

输入信号	停止	正转启动	反转启动	过载保护	正向限位	反向限位
名称	SB1	SB2	SB3	FR	SP1	SP2
DI 端子号	I0.0	I0.2	I0.4	I0.6	I1.0	I1.1
输出信号	电机正转	电机反转	-	-	-	-
DO 端子号	Q0.1	Q0.3	-	-	-	-

3.1.3 控制系统接线

3.1.3.1 RPC3000 系列 PLC 控制系统接线

RPC3000 系列 PLC 组成的控制系统输入、输出信号接线图如图 3.3 所示。PLC 的输入、输出模块需外接 DC24V 电源供电，漏型输入公共端 1M、2M 接 24VDC 的正极。实物接线图如图 3.4、3.5 所示。其中，输入信号接线完成后，可上电检验信号接线是否正确。即：PLC 通电后，接通按钮开关，输入模块液晶显示面板上对应通道的 LED 指示灯会被点亮，表明接线正确无误，如图 3.4 所示。

由于 PLC 的输出模块输出为直流 24V 电压，而主电路中需要控制的接触器为交流接触器，因此在 PLC 输出电路及主电路中，需加装中间继电器 KA1、KA2，由 PLC 的输出端 Q0.1 和 Q0.3 分别控制 KA1、KA2 的线圈，如图 3.5 所示。由 KA 的常开触点（图 3.6 中 KA1、KA2）再分别控制 KM1、KM2 的线圈（图 3.6 中 A1、A2 为接触器线圈接线端子）接通交流 220V 电压。KM 的主触点控制三相电源按不同相序接通从而控制电机正、反向转动。KA 的常闭触点为防止正、反向同时接通的互锁触点。

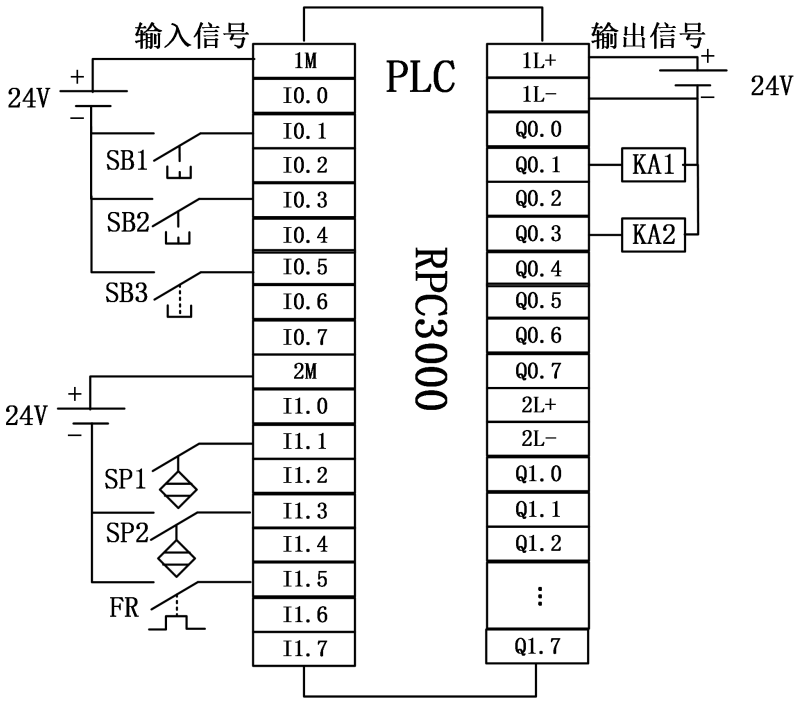


图 3.3 RPC3000PLC 控制系统接线图

图 3.4 输入信号接线测试

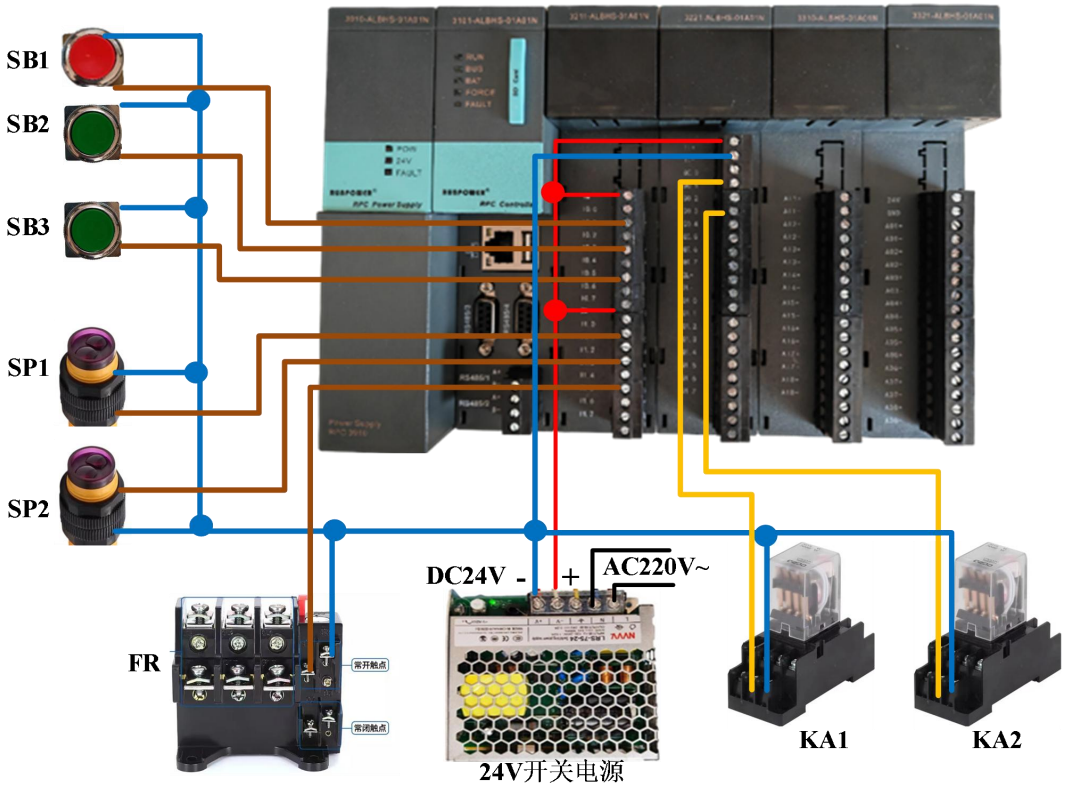


图 3.5 RPC3000 控制系统实物接线图（PLC 控制电路）

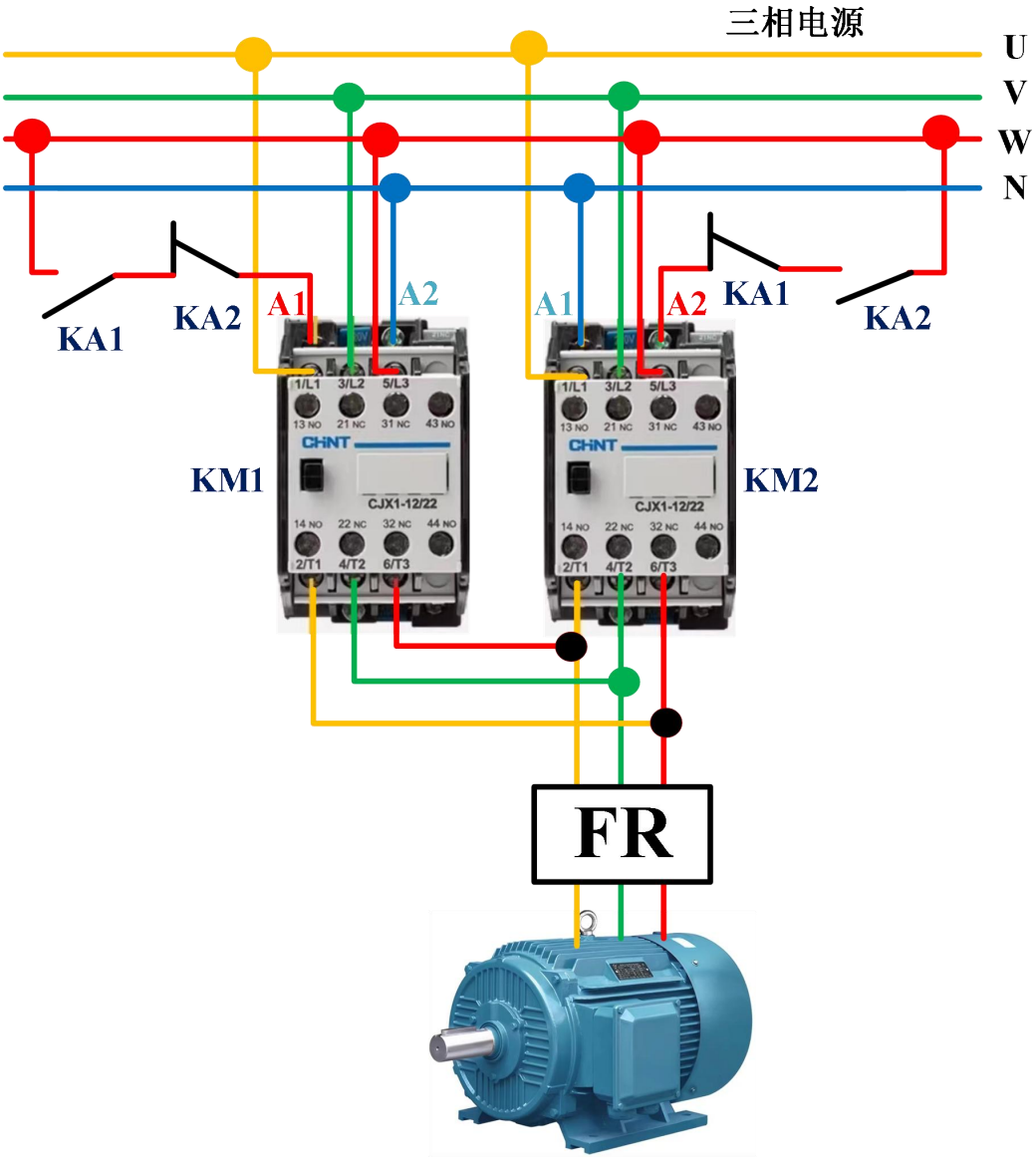


图 3.6 RPC3000 控制系统实物接线图（主电路）

3.1.3.2 RPC2000 系列 PLC 控制系统接线

若选用 RPC2000 控制系统，PLC 的输入、输出信号接线图如图 3.7 所示。PLC 的输入、输出模块需要外部供电，因此需接入 24V 直流电源。实物接线图如图 3.8 所示。输入信号接线完成后，可以上电检验信号接线是否正确，PLC 和 24V 电源接通后，操作按钮开关，CPU 模块面板上对应的输入信号指示灯会点亮，表明接线正确无误。由于 PLC 的输出模块可以输出交流 220V 电压，因此输出电路可直接控制 KM1、KM2 的线圈（图 3.8 中 A1、A2 为接触器线圈接线端子）接通交流 220V 电压。KM 的主触点控制三相电源按不同相序接通从而控制电机正、反向转动。KM 的常闭触点为防止正、反向同时接通的互锁触点。

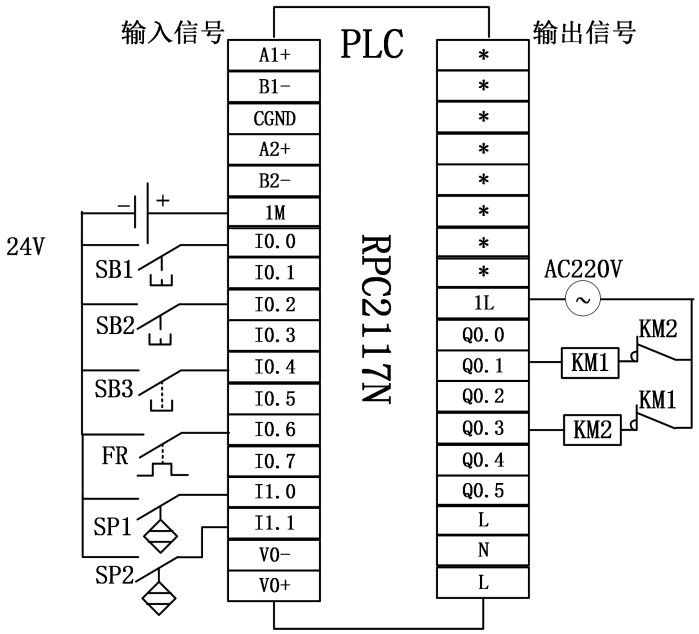


图 3.7 RPC2000 控制系统接线图

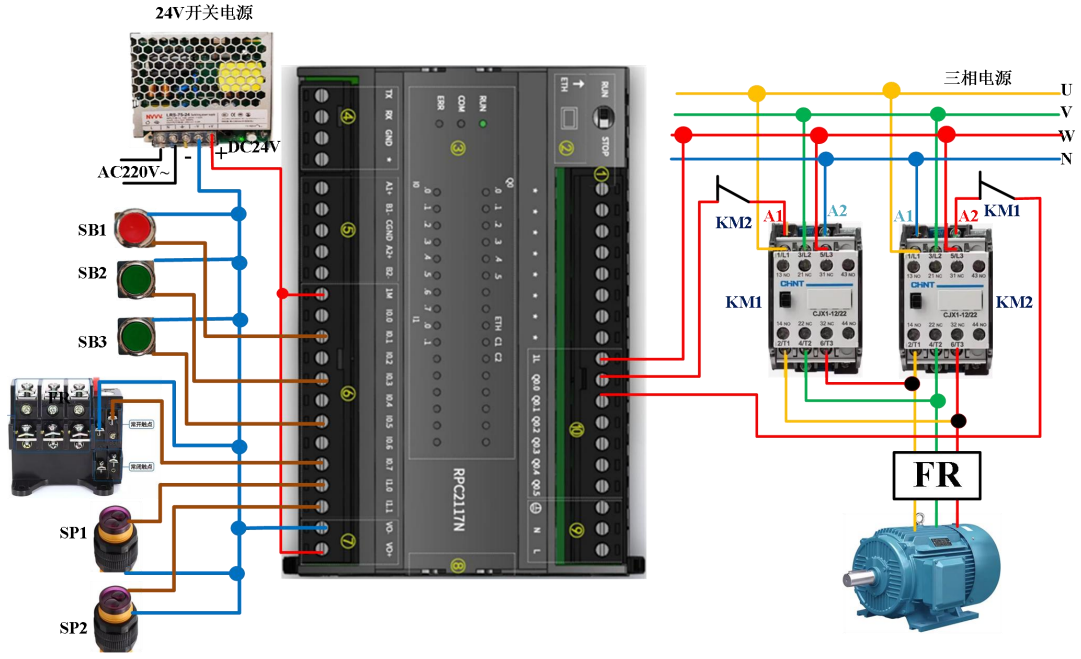


图 3.8 RPC2000 控制系统实物接线图

3.1.4 设备组态及编程

本节以 RPC3000 为例说明 RPC 系列 PLC 自动往返控制程序编程方法。按前面章节所述的方法新建工程，并完成设备组态。双击设备树中 PLC_PRG (PRG) 图标进行梯形图程序的编制，程序的变量声明如图 3.9 所示。在编程时，变量可以先声明后调用，也可在编程的同时自动声明变量。添加变量时，要输入变量名称、地址（I、Q、M 类型变量）、数据类型以及

注释（可选）等信息。需要注意的是，与 PLC 输入、输出模块相对应的变量需要输入相应的地址，如图 3.9 的“地址”列。

	类别	名称	地址	数据类型	初值	注释
1	VAR	Stop	%IX0.1	BOOL		停止
2	VAR	Start1	%IX0.3	BOOL		正向启动
3	VAR	Start2	%IX0.5	BOOL		反向启动
4	VAR	Motor_F	%QX0.1	BOOL		电机正转
5	VAR	Motor_B	%QX0.3	BOOL		电机反转
6	VAR	SP_F	%IX1.1	BOOL		正向限位
7	VAR	SP_B	%IX1.3	BOOL		反向限位
8	VAR	FR_P	%IX1.5	BOOL		过载保护
9	VAR	T1		TON		
10	VAR	M1		BOOL		
11	VAR	M2		BOOL		
12	VAR	T2		TON		

图 3.9 变量声明

编制的梯形图程序如图 3.10 所示。在梯形图程序中，最左侧的一条线为程序的起始线，可以等效为电源的“火线”，每行程序都是由起始线开始。程序中没有电流的流动，各指令之间构成逻辑运算关系。例如：触点之间串联表示逻辑“与”，触点之间并联表示逻辑“或”。程序采用从左到右、从上到下的循环扫描的方式执行。每一个触点只有在它本身为接通状态，并且在它之前的逻辑运算结果为“1（TRUE）”时，才能将“1”传递到后面的程序运算中。一行程序如果以线圈 ( ) 结束，会对线圈之前的程序段运算结果进行输出。

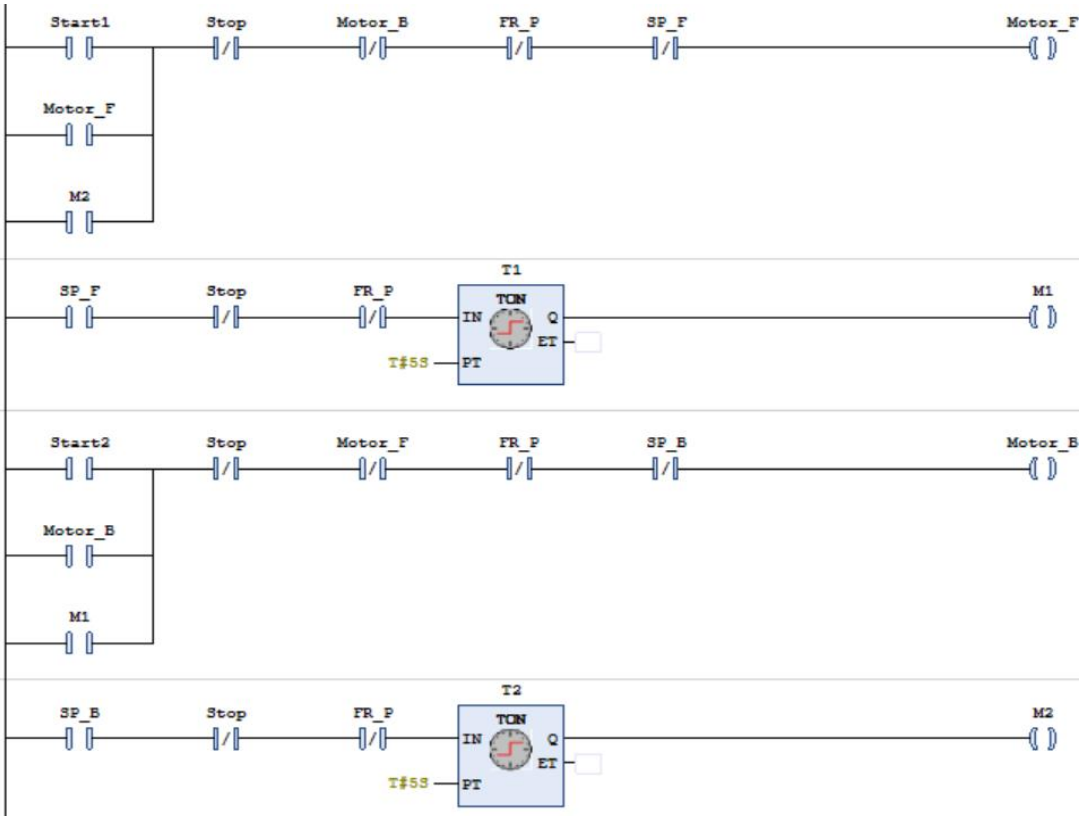


图 3.10 自动往返控制梯形图程序

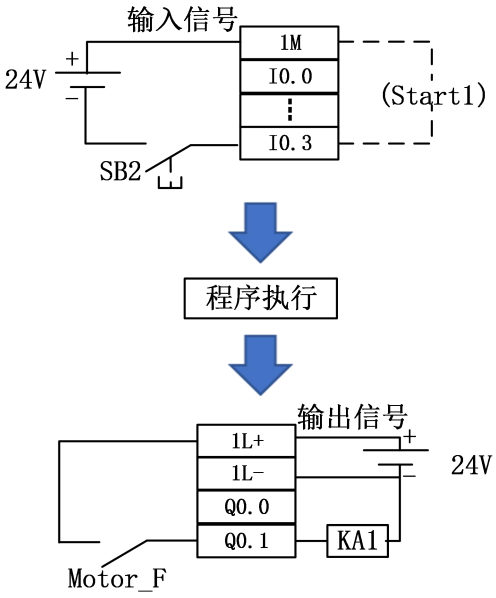
3.1.5 程序分析

PLC 的梯形图程序与继电接触器控制电气原理图十分相似，在程序执行过程中，将输入、输出信号等效成“软继电器”进行分析更易于理解。对于 RPC 系列 PLC，接线图 3.3 或图 3.7 中的每一个输入信号都用于控制程序中输入变量 I 的线圈，如图 3.11 所示。

例如：程序中变量 Start1 可以等效为一个软继电器，其映射到输入模块的地址为%IX0.3，接入的输入信号为按钮 SB2，按钮 SB2 的开合控制着“输入继电器”Start1 的线圈。因此，当按钮 SB2 断开时，变量 Start1 的线圈为断电状态，变量的值为 FALSE。只有当按钮 SB2 接通时，变量 Start1 的线圈才为通电状态，变量的值为 TRUE。

同样，输出变量也可以等效为软继电器——输出继电器。例如变量 Motor_F 为控制电机正转的输出继电器，其映射到输出模块的地址为%QX0.1，接到输出模块上的信号为 KA1 的线圈。因此，程序运行结果，即变量 Motor_F 线圈的值（TRUE/FALSE）控制输出继电器常开点的通断，从而控制输出回路 KA1 线圈的通电情况。在图 3.6 中，由于 PLC 输出电压为直流 24V，因此输出信号需先控制中间继电器 KA1，再由 KA1 常开触点控制交流接触器 KM 的线圈，来控制三相异步电机的正反转。此外，也可以采用具有 AC220V 输出的控制模块直接控制接触器线圈的通、断电。

在图 3.8 所示的 RPC2000 控制系统接线图中，RPC2117N 自带 6 路继电器输出（AC220V），输出信号可直接控制接触器线圈通断，从而控制电机正反转。



3.1.6 下载调试

程序编制完成并且 PLC 的硬件接线也全部完成之后，可以将程序下载至 PLC 进行在线调试运行。以 RPC3000 系列 PLC 为例，步骤如下：

- PLC 上电——接通开关电源，且保证与 PC 机以太网连接正常，此时 PLC 模块液

晶面板显示 RUN 与 IBS，如图 3.4 所示，表明 PLC 模块与 PC 机通讯正常。

- 连接 PC 机与 PLC 设备——打开 PLC 工程，编译程序后，双击设备树视图中的“Device（Runpower-RPC3000）”，在打开的设备视图中单击“扫描网络”，在打开的“选择设备”窗口双击“MyDevice”。
- 登录到 PLC——单击菜单“在线/登录”。
- 调试运行——单击菜单“调试/启动”，操作输入按钮等信号，观察程序运行及输出结果是否正确，并进行相应调整，直至达到控制要求，调试结束。

启动程序在线运行界面如图 3.12 所示。在线运行状态，程序中左侧起始线会变成蓝色，表示程序开始初始化运行。所有输入模块接入的信号所对应的输入继电器线圈的值都处于“FALSE”状态，即：常开点显示为断开，常闭点显示为闭合状态。每行程序的输出线圈也处于“FALSE”状态。

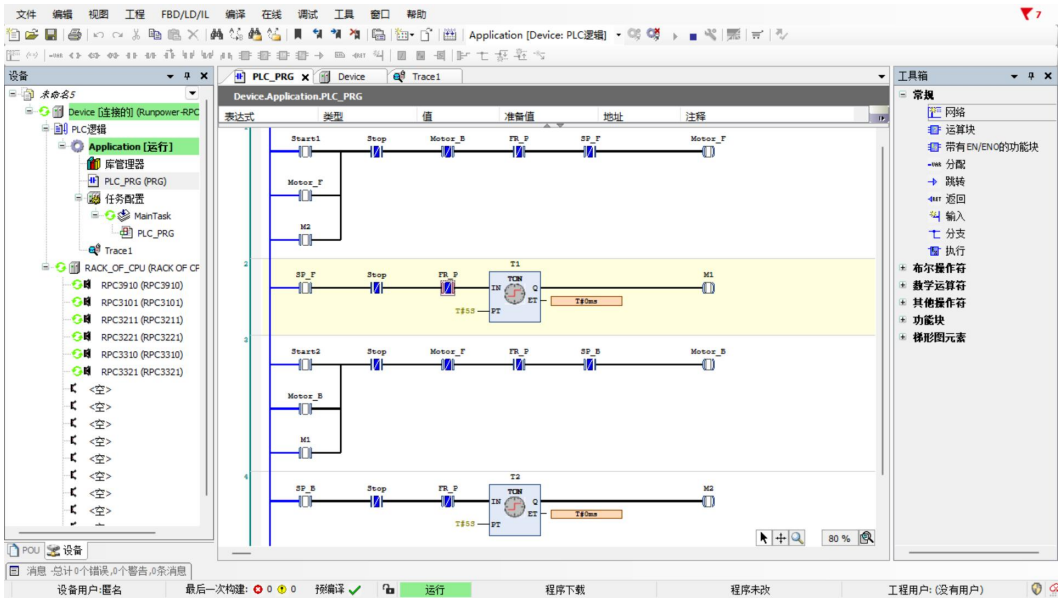


图 3.12 PLC 程序在线运行界面

在线运行状态下，程序运行过程如下：

- 正向启动：按一次启动按钮 SB2（%IX0.3），程序段 1 中变量 Start1（正向启动）常开点状态为 1，变量 Motor_F（电机正向启动）的线圈 Motor_F 输出为 1，电机正向启动运行并自锁，图 3.1 所示小车开始向左行驶。
- 开始装料：当行驶至左侧装料处接近开关 SP1（%IX1.1）处，程序段 1 处变量 SP_F 常闭点 SP_F 状态为 0，变量 Motor_F 的线圈 Motor_F 状态为 0，小车正向运行停止并开始装料；程序段 2 处变量 SP_F 常开点 SP_F 状态为 1，定时器 T1 开始进行装料计时，装料定时时间为 5 秒。
- 自动反转：5 秒定时时间结束后，装料结束。程序段 2 输出的内部变量 M1（软继电器）的线圈 M1 状态为 1，程序段 3 处 M1 的常开点 M1 状态为 1，使得变量 Motor_B（电机反向启动）的线圈 Motor_B 状态为 1，电机开始反转并自锁，

小车向右行驶去卸料处。

- 开始卸料：当行驶至右侧卸料处接近开关 SP2（%IX1.3）处，程序段 3 处变量 SP_B 常闭点  状态为 0，变量 Motor_B（反向）的线圈  状态为 0，小车停止运行并开始卸料；程序段 4 处变量 SP_B 常开点  状态为 1，定时器 T2 开始进行卸料计时，卸料定时时间为 5 秒。
- 自动返回：5 秒定时时间结束后，卸料结束。程序段 4 输出的内部变量 M2（软继电器）的线圈  状态为 1，程序段 1 处 M2 的常开点  状态为 1，使得变量 Motor_F 的线圈状态为 1，电机又开始正转。如此循环往复。
- 停止电机：按下停止按钮 SB1（%IX0.1），电机正常停止运行。如果发生过载，热继电器过载保护 FR%IX1.5）动作会使电机停止运行。

3.1.7 仿真调试

在工程界面选择菜单“在线/仿真/登录”，再选择菜单“调试/启动”，进行程序在线运行界面，如图 3.13 所示。

在仿真运行模式下，按照程序逻辑，按顺序对变量进行“写入”、 “强制”或“释放”操作，可以验证程序执行是否符合控制要求。

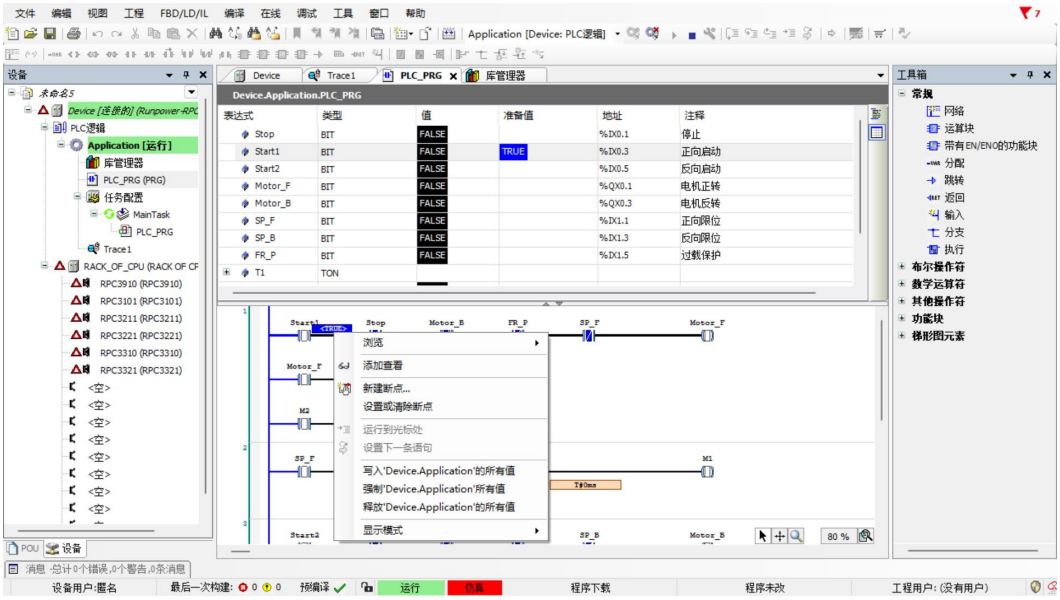


图 3.13 仿真运行界面

附录 A 指令速查表

指令类型	指令说明	指令助记符	所在库
赋值指令	赋值指令	MOVE	内置（无需添加）
算术运算指令	加法指令	ADD	内置
	减法指令	SUB	
	乘法指令	MUL	
	除法指令	DIV	
	取余指令	MOD	
选择指令	二选一指令	SEL	内置
	多选一指令	MUX	
	取最小值指令	MIN	
	取最大值指令	MAX	
	极限值指令	LIMIT	
比较指令	小于指令	LT	内置
	小于等于指令	LE	
	大于指令	GT	
	大于等于指令	GE	
	不等于指令	NE	
	等于指令	EQ	
移位指令	右移指令	SHR	内置
	循环右移指令	ROR	
	左移指令	SHL	
	循环左移指令	ROL	
逻辑运算指令	取非指令	NOT	内置
	与指令	AND	
	或指令	OR	
	异或指令	XOR	
初等数学运算指令	平方根指令	SQRT	内置
	绝对值指令	ABS	
	指数指令	EXP	
	自然对数指令	LN	
	常用对数指令	LOG	

	幂指令	EXPT	
	正弦指令	SIN	
	反正弦指令	ASIN	
	余弦指令	COS	
	反余弦指令	ACOS	
	正切指令	TAN	
	反正切指令	ATAN	
地址运算指令	数据类型大小指令	SIZEOF	内置
	取地址指令	ADR	
	取地址内容指令	^	
转换指令	布尔类型转换指令	BOOL_TO_<TYPE>	内置
	字节类型转换指令	BYTE_TO_<TYPE>	
	无符号短整型转换指令	USINT_TO_<TYPE>	
	短整型转换指令	SINT_TO_<TYPE>	
	字类型转换指令	WORD_TO_<TYPE>	
	双字类型转换指令	DWORD_TO_<TYPE>	
	无符号整数类型转换指令	UINT_TO_<TYPE>	
	整数类型转换指令	INT_TO_<TYPE>	
	无符号双整数类型转换指令	UDINT_TO_<TYPE>	
	双整数类型转换指令	DINT_TO_<TYPE>	
	日期类型转换指令	DATE_TO_<TYPE>	
	时间类型转换指令	TIME_TO_<TYPE>	
	日期时间类型转换指令	DT_TO_<TYPE>	
	时钟类型转换指令	TOD_TO_<TYPE>	
	实数类型转换指令	REAL_TO_<TYPE>	
	字符类型转换指令	STRING_TO_<TYPE>	
	截短转换指令	TRUNC	
初始化操作指令	初始化操作指令	INI	内置
调用指令	调用指令	CAL	内置
位/字节操作指令	位赋值指令	PUTBIT	Util.compiled-library
	位拆分指令	UNPACK	

	位整合指令	PACK	
	位提取指令	EXTRACT	
BCD 码转换指令	整型转 BCD 码指令	INT_TO_BCD	Util.compiled-library
	BCD 码转整型指令	BCD_TO_INT	
信号发生器指令	典型周期信号发生器	GEN	Util.compiled-library
	脉冲信号发生器	BLINK	
函数操纵器指令	整型限速	RAMP_INT	Util.compiled-library
	实型限速	RAMP_REAL	
	特征曲线	CHARCURVE	
高等数学运算指令	整型统计指令	STATISTICS_INT	Util.compiled-library
	实型统计指令	STATISTICS_REAL	
	平方偏差指令	VARIANCE	
	积分指令	INTEGRAL	
	微分指令	DERIVATIVE	
控制器指令	比例微分控制器	PD	Util.compiled-library
	比例积分微分控制器	PID	
	比例积分微分控制器	PID_FIXCYCLE	
模拟量处理指令	上下限报警	LIMITALARM	Util.compiled-library
	滞后	HYSTERESIS	
字符串处理指令	左边取字符串指令	LEFT	Standard.compiled-library
	中间取字符串指令	MID	
	右边取字符串指令	RIGHT	
	取字符串长度指令	LEN	
	删除字符指令	DELETE	
	插入字符串指令	INSERT	
	替换字符串指令	REPLACE	
	查找字符串指令	FIND	
	合并字符串指令	CONCAT	
计数器指令	递减计数器	CTD	Standard.compiled-library
	递增计数器	CTU	
	递增递减计数器	CTUD	
定时器指令	实时时钟	RTC	Standard.compiled-library
	普通定时器	TP	
	通电延时定时器	TON	

	断电延时定时器	TOF	
双稳态指令	置位优先双稳态触发器	SR	Standard.compiled-library
	复位优先双稳态触发器	RS	
触发器指令	下降沿检测触发器	F_TRIG	Standard.compiled-library
	上升沿检测触发器	R_TRIG	
数学指令	16 进制数转换为工程量数据	HEX_ENGIN	RPCMath.compiled-library
	工程量数据转换为 16 进制数据	ENGIN_HEX	
	产生 Modbus CRC 校验码	Modbus_CRC	
	数据传送指令	Block_Move	
	RPC 比例积分微分控制器	RPCPID	
RS485 口通讯指令	设置 RS485 口通讯参数	SET_COMM_PRMT	RPCCommon.compiled-library
	读取 RS485 口通讯参数	GET_COMM_PRMT	
	设置 RS485 口从站通讯地址	SET_COMM_ADDRESS	
	读取 RS485 口从站通讯地址	GET_COMM_ADDRESS	
	RS485_1 口 Modbus-RTU 主站功能块	ModbusSerial1_Master	
	RS485_2 口 Modbus-RTU 主站功能块	ModbusSerial2_Master	
	RS485_1 口自由协议通讯数据发送	FreeSerial1_Send	
	RS485_1 口自由协议通讯数据接收	FreeSerial1_Receive	
	RS485_2 口自由协议通讯数据发送	FreeSerial2_Send	
	RS485_2 口自由协议通讯数据接收	FreeSerial2_Receive	
以太网口通讯指令	设置以太网口 TCP/IPv4 参数	SET_LOCAL_IP	RPCCommon.compiled-library
	读取以太网口	GET_LOCAL_IP	

	TCP/IPv4 参数		
	清除以太网口 TCP/IPv4 参数	CLEAR_LOCAL_IP	
	以太网口 Modbus-TCP 主站功能块	ModbusTCP_Master	
	以太网口自由协议套接字打开	FreeTCP_Open	
	以太网口自由口关闭	FreeTCP_Close	
	以太网口自由协议通讯数据发送	FreeTCP_Send	
	以太网口自由协议通讯数据接收	FreeTCP_Receive	
硬件设备指令	设置实时时钟（普通数据格式）	SET_HD_RTC_X	RPCCommon.compiled-library
	设置实时时钟（类格林威治时间格式）	SET_HD_RTC	
	读取实时时钟日期、时间、星期	GET_HD_RTC	
特殊指令	读取内置保持电池状态	GET_BAT_STATUS	RPCCommon.compiled-library
	读取指定模块状态	GET_IO_STATUS	
	读取冗余状态	GET_REDU_STATUS	
	读取任务运行信息	GET_TASK_INFO	

参考文献

[1] 马立新, 陆国君. 开放式控制系统编程技术——基于 IEC 61131-3 国际标准[M]. 北京：人民邮电出版社 2018.8.
[2] RPC3000 系列 PLC 软件手册[G]. 北京蓝普锋科技有限公司, 2022.
[3] RPC3000 系列可编程控制器使用手册[G]. 北京蓝普锋科技有限公司, 2022.
[4] RPC3000 系列 PLC 产品手册[G]. 北京蓝普锋科技有限公司.
[5] 李沛，宋晓娜，侯晓霞等. 电工电子技术实践[M]. 北京：清华大学出版社，2020.4.

2023

北京蓝普锋科技有限公司
Beijing Runpower Technology Co.,Ltd.
地址：北京市昌平区东小口都市芳园嘉湖园22号楼
E-mail:Service@runpower.cn
电话：010-62740825
技术热线：18519861720
销售热线：18510991991