

Truhigh P600 PLC 系统

软件手册

V2.0



前言

用途

该手册叙述了如何使用 Truhigh P600 PLC 系统的组态软件。
用户根据本手册可以使用 Truhigh P600 PLC 系统编写用户程序。

基础知识要求

用户要熟悉本手册，需要具有自动化技术的一般知识，熟悉 windows 操作。

Truhigh P600 编程软件的特点

Truhigh P600 软件系统采用德国科威软件有限公司的控制器运行引擎 ProConOS 和 IEC 61131-3 编程工具 Multiprog。ProConOS 基于 Microsoft 的 .NET 技术，采用预编译器(AOT)，直接生成机器代码，从而保证控制器的实时性，同时也大大减少了程序所占用的存储空间。编程系统 Multiprog 采用简单易用的用户界面，提供大量的成熟功能，可进行便捷的工程创建、快速的应用开发以及程序仿真、下载、监控和试运行。



系统包含以下功能：

- ◆ 包含库管理的工程浏览器
- ◆ 工程向导和编辑向导
- ◆ 方便图形 LD/FBD 代码开发的先进梯形图编辑器
- ◆ 带语法高亮显示和智能感知的文本编辑器
- ◆ 基于网格的变量编辑器，可方便、无误地对变量及 FB 实例声明
- ◆ 编辑和故障排除模式下均带有交叉引用窗口
- ◆ 经过优化的代码编译，只编译工程中被修改的部分
- ◆ 集成的 DCS/PLC 仿真带扩展仿真模式
- ◆ 逻辑分析器、断点、地址调试、单步模式、变量和配方覆盖和强制等故障排除和调试功能
- ◆ 监视窗口连续监视实例信息
- ◆ 详尽的上下文帮助系统
- ◆ 用户界面语言：英语、德语、中文和日语

MULTIPROG EXPRESS IEC 61131 程系统包含了下列编程语言：

- ◆ 指令表(IL)
- ◆ 结构化文本(ST)
- ◆ 梯形图(LD)
- ◆ 功能块图(FBD)
- ◆ 顺序功能图(SFC)

计算机硬件要求

处理器	推荐 Pentium 4 2 GHz 及以上
内存	Win XP: 512MB Win 7 / Win 10: 2GB
硬盘存储	10GB 的空闲存储空间
通信	TCP/IP
操作系统	Win XP SP3, Win Vista SP2, Win7 (32 Bit), Win10(32Bit/64Bit) Internet Explorer 6.0 或更高版本

系统限制

工程树中的节点数目	8000
工程树中的配置/资源数目	1 / 1
每个资源中的程序实例数目	15
每个资源的任务数	5
每个任务中的程序实例数目	15
每个 POU 的全局变量/局部变量数	5000 / 1500
包含的库数目	32
每个工程中 POU（包括库的 POU）的数目	256
每个工程支持的 I/O 数量	128 Kbytes
I/O 组数目	200

目录

前言	I
目录	III
第 1 章 如何开始	1
1.1 软件安装	2
1.2 TRUHIGH P600 组态软件	5
1.3 连接 TRUHIGH P600 PLC	6
1.4 修改 TRUHIGH P600 PLC 的 IP 地址	6
1.5 创建一个工程	8
1.5.1 启动 MULTIPROG 编程软件	8
1.5.2 工程向导	9
1.5.3 TRUHIGH P600 PLC 的配置	14
1.5.4 创建一个程序	15
1.6 下载用户程序	17
1.7 在线监视用户程序	19
第 2 章 TRUHIGH P600 系统的工作原理	20
2.1 TRUHIGH P600 系统如何执行用户编写的程序	21
2.2 TRUHIGH P600 系统的数据存取	21
2.3 TRUHIGH P600 系统的保存数据	22
2.4 数据类型	22
2.4.1 基本数据类型	23
2.4.2 类属数据类型	24
2.4.3 用户自定义数据类型	24
2.4.4 常量数据的表示	25
第 3 章 MULTIPROG 软件	26
3.1 用户界面	27
3.1.1 菜单栏	28
3.1.2 工具栏	54
3.1.3 工作区	55
3.1.4 交叉参考窗口	55
3.1.5 定制用户界面	58
3.2 工程树——硬件	59
3.2.1 物理硬件	59

3.2.2 配置	59
3.2.3 资源	59
3.2.4 TASKS	61
3.2.5 GLOBAL_VARIABLES——全局变量	62
3.2.6 IO-CONFIGURATION——IO 配置	68
3.3 工程树——工程	69
3.3.1 库	69
3.3.2 数据类型	72
3.3.3 逻辑 POU	72
 第 4 章 硬件组态	 75
4.1 CPU 编辑	77
4.2 本地 IO 组态	78
4.3 作为标准 MODBUS 从站组态 MBS	80
4.4 PROFINET 组态	87
 第 5 章 用户程序编辑	 94
5.1 POU 编程	95
5.1.1 变量工作单	95
5.1.2 新建逻辑 POU	98
5.1.3 逻辑 POU 的类型	100
5.2 IL 指令表编程语言	107
5.2.1 创建一个 IL 程序	107
5.2.2 IL 的语句	110
5.2.3 IL 的操作符	110
5.3 ST 结构化文本编程语言	112
5.3.1 创建一个 ST 程序	112
5.3.2 ST 的语句	116
5.3.3 ST 的表达式	117
5.4 FBD 功能块图编程语言	118
5.4.1 创建一个 FBD	118
5.5 LD 梯形图编程语言	121
5.5.1 创建一个 LD 程序	121
5.5.2 在 LD 中插入一个 FB	123
5.6 SFC 顺序功能图编程语言	126
5.6.1 创建一个 SFC 程序	127
5.6.2 SFC 的动作限定符	139
5.7 各类 POU 的调用	140
5.8 编译工程	140
5.8.1 编译器命令	141
5.8.2 “制作”命令	142
5.9 工程仿真	143

第 6 章 OPC 通讯	145
6.1 OPC 服务器设置	146
6.2 测试 OPC 服务器	149
6.3 上位机通过 OPC 连接 TRUHIGH P600 系统	151
6.3.1 IFIX 5.5 的 OPC 设置.....	151
6.3.2 在 IFIX 5.5 数据库中添加变量.....	158
6.4 IFIX 画面组态	166
6.5 上位机 WINCC 通过 OPC 连接 TRUHIGH P600 系统	167
6.5.1 WINCC 的 OPC 变量添加.....	167
6.5.2 WINCC 的画面组态.....	170

第 1 章 如何开始

本章叙述软件安装步骤，如何连接 Truhigh P600 PLC，如何利用编程软件创建一个程序，如何下载并运行这个程序。

本章分为以下八节：

- 1.1 软件安装
- 1.2 Truhigh P600 系统组态软件
- 1.3 连接 Truhigh P600 控制器
- 1.4 修改 Truhigh P600 控制器的 IP 地址
- 1.5 创建一个工程
- 1.6 下载用户程序
- 1.7 在线监视用户程序

1.1 软件安装

1. 双击 Truhigh P600 安装软件包，进入安装欢迎界面（图 1-1-1），点击下一步。



图 1-1-1

2. 进入版权提示界面（图 1-1-2），如果接受授权协议，点击“我接受”按钮，继续安装；否则点击“取消”，退出安装。

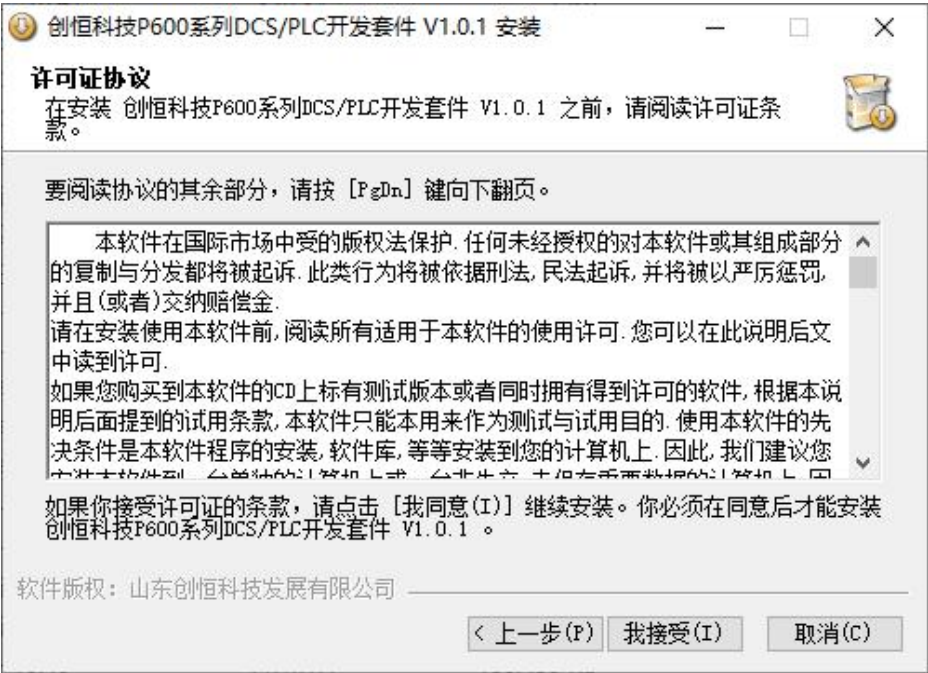


图 1-1-2

3. 进入安装目录选择界面（图 1-1-3），选择安装目标文件夹，然后点击“安装”按钮。

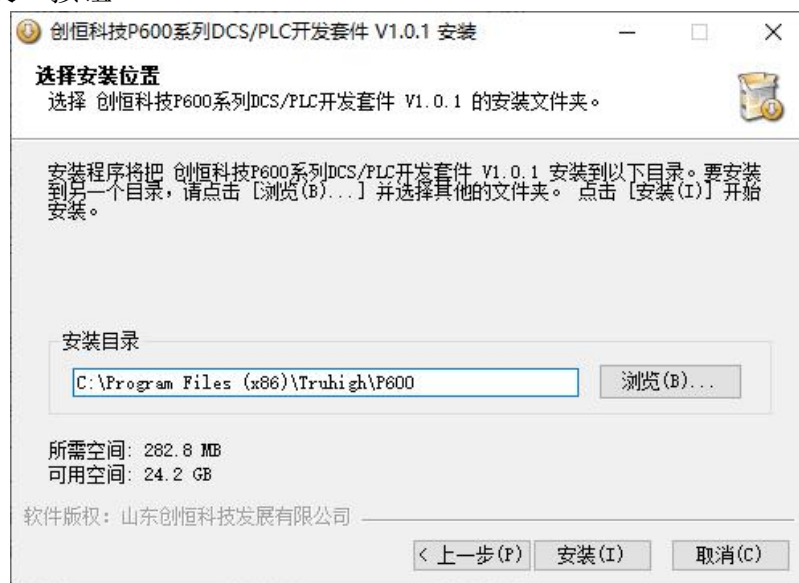


图 1-1-3

4. 进入“Npcap”插件安装界面（图 1-1-4）。

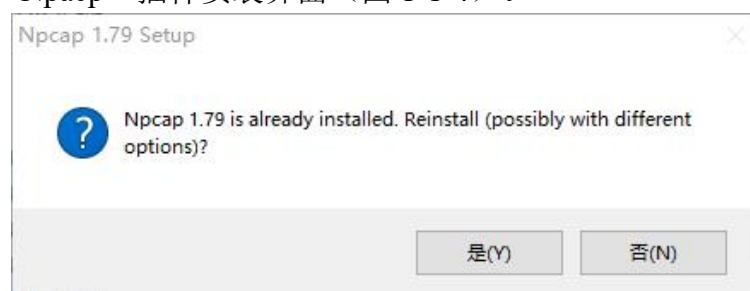


图 1-1-4

点击“是”。

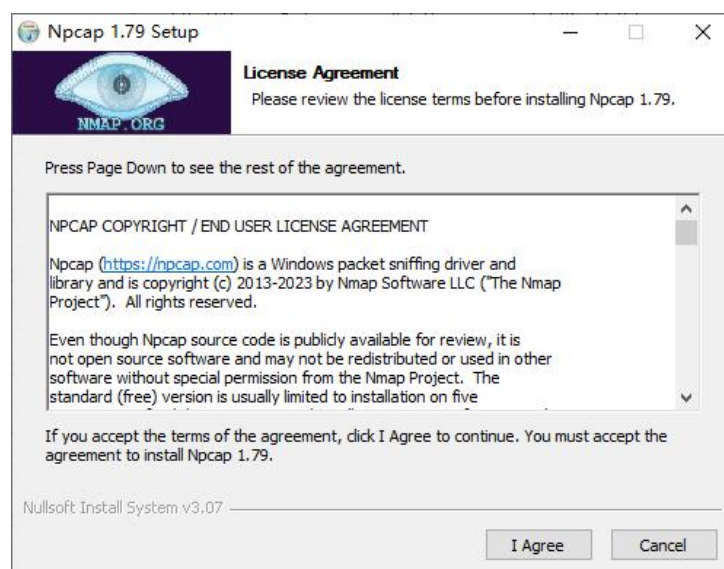


图 1-1-5

选择“I Agree”。

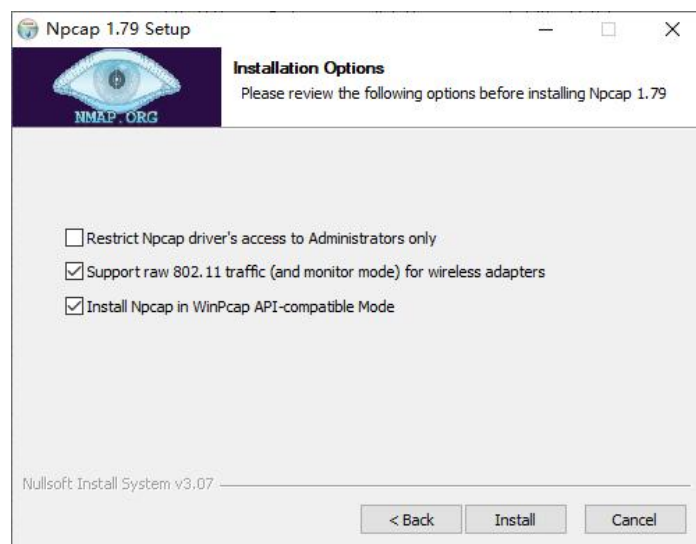


图 1-1-6

选择“Install”，等待安装完成，选择“Next”。

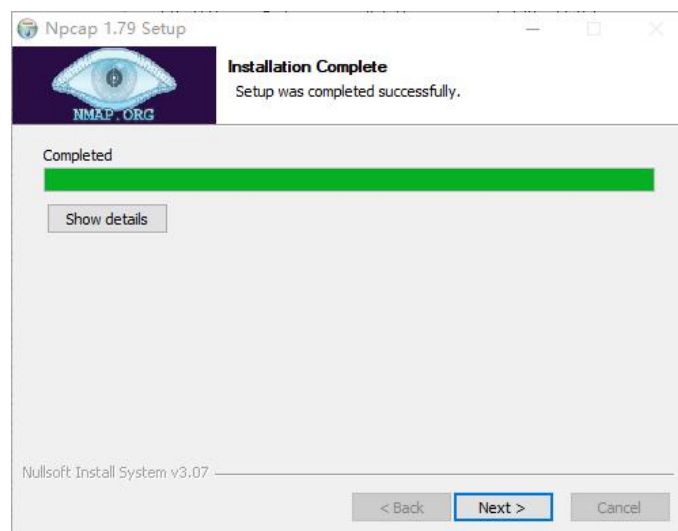


图 1-1-7

5. 进入“正在安装”界面（图 1-1-4）。

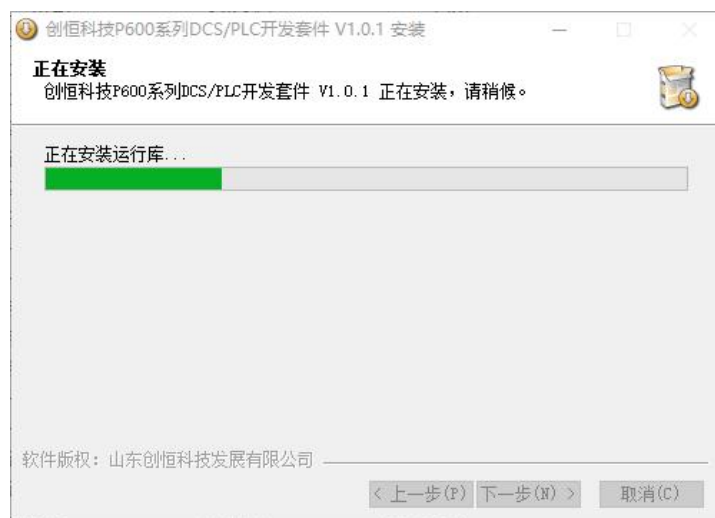


图 1-1-8

6. 程序安装完成，点击“完成”按钮，退出安装程序。



图 1-1-9

1.2 Truhigh P600 组态软件

软件安装完成后，在“开始”->“所有程序”->“TruHighP600”菜单中生成的快速菜单，如图 1-2-1。

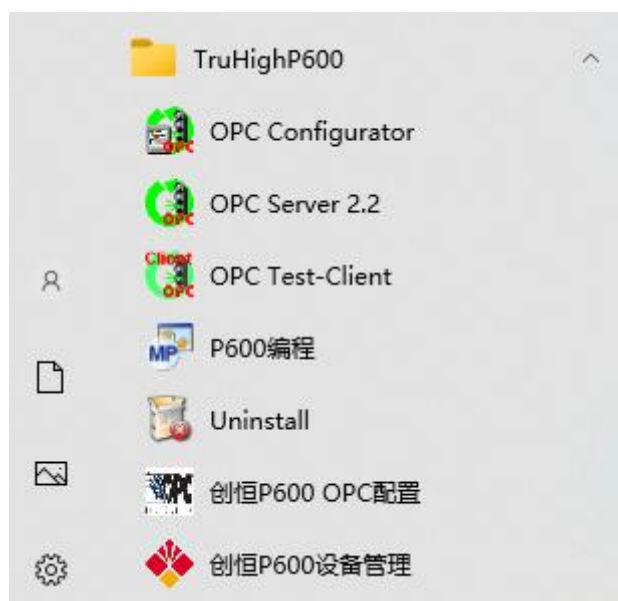


图 1-2-1

- **P600编程**：Multiprog Express 是集成开发环境，用于控制程序的编写、仿真、下载等。
- **OPC Configurator**：OPC.OUT.PCOS.21 OPC Server 配置程序，用于配置 OPC 服务器与 Truhigh P600 PLC 数据交换的参数。

-  OPC Server 2.2 : OPC.OUT.PCOS.21 OPC Server 用于组态软件和硬件平台的数据交互。
-  OPC Test-Client : OPC.OUT.PCOS.21 OPC 客户端，用于测试连接 OPC 服务器。
-  创恒P600设备管理 : Truhigh P600 管理器用于查看和修改设备系统参数。
-  Uninstall : 用于卸载 Truhigh P600 组态软件。

1.3 连接 Truhigh P600 PLC

连接 Truhigh P600 PLC 需要做两个工作：

- 一是连接电源到 Truhigh P600 PLC；
- 二是用过网线连接计算机与 Truhigh P600 PLC。

连接电源：

PLC 供电是 220VAC/24VDC 供电。

连接网线：

用直连网线（或通过交换机）连接计算机和 Truhigh P600 PLC。

1.4 修改 Truhigh P600 PLC 的 IP 地址

打开设备管理工具。



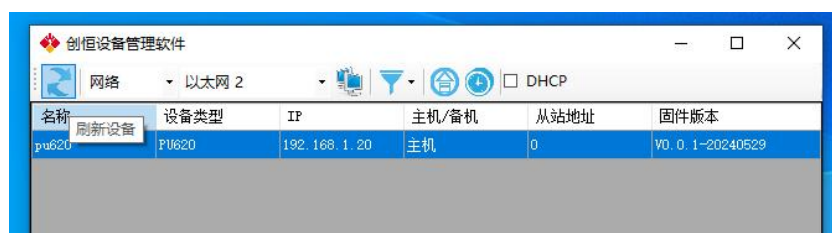
主界面如下：

名称	设备类型	IP	主机/备机	从站地址	固件版本
pu620-创恒	PU620	192.168.0.30	主机	0	V0.0.2-20240512
pu610	PU620	192.168.0.50	主机	0	V0.0.1-20240522
pu620	PU620	192.168.1.20	主机	0	V0.0.1-20240529
ci610-创恒	CI610	192.168.0.31	主机	2	V0.4.0-20231120
ci610-dev4	CI610	192.168.0.54	主机	4	V0.4.0-20231120
ci610-dev5	CI610	192.168.0.55	主机	5	V0.4.0-20231120
ci610-dev2	CI610	192.168.0.52	主机	2	V0.4.1-20240328
ci610-dev3	CI610	192.168.0.53	主机	3	V0.4.1-20240328

设备数量: 8

: 刷新设备;
 : 刷新通讯接口;
 : 过滤显示设备;
 : 批量升级固件;
 : 同步时间;
 ☐ DHCP: 开启 DHCP。

当鼠标光标放在相应图标上会显示提示信息，如下图：



选择对应的物理网卡，点击“刷新设备”，会在列表框里面显示找到的 PLC 设备。

右键点击相应设备，选择“网络”：



在弹出的界面中修改相应的 IP 信息。点击“确定”，保存信息。

修改IP

MAC: 00E062410335

IP: 192.168.1.20

子网掩码: 255.255.255.0

网关: 192.168.1.1

确定 取消

1.5 创建一个工程

1.5.1 启动 MULTIPROG 编程软件

启动 Multiprog 编程软件有两种方法：

- 一是通过“开始菜单”启动 Multiprog 编程软件。

打开“开始菜单” -> “所有程序” -> “TruHighP600” -> “P600 编程”，运行 Multiprog 编程软件。

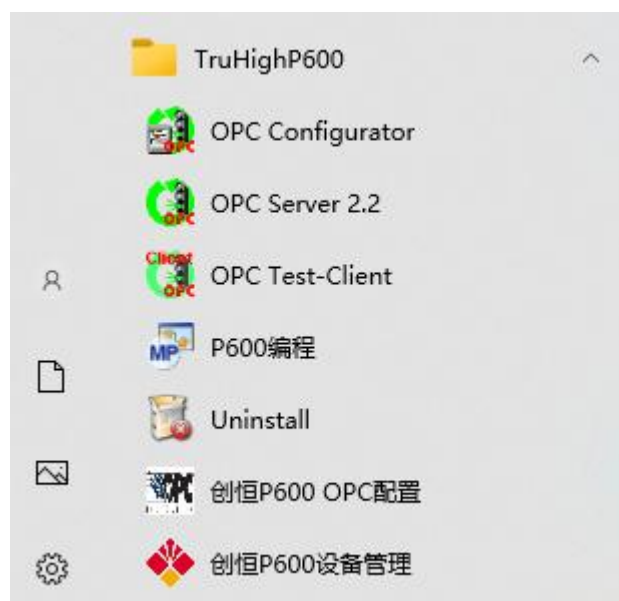


图 1-5-1

- 二是通过桌面快捷方式启动 Multiprog 编程软件。



图 1-5-2

Multiprog 编程软件界面如图 1-5-3。

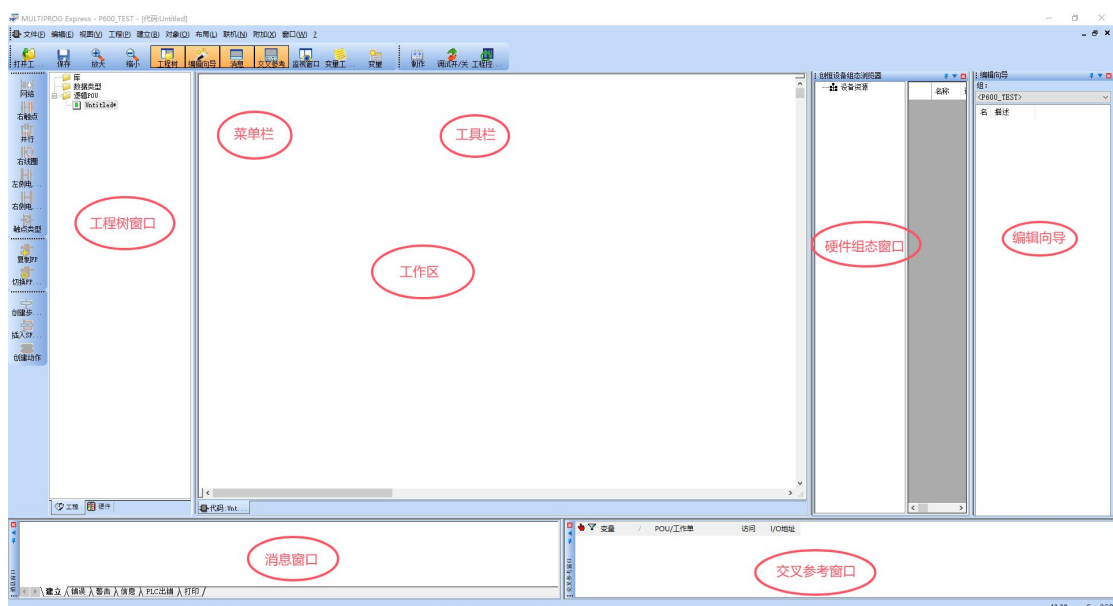


图 1-5-3

1.5.2 工程向导

1. 在工程浏览器中点击“文件>新建工程”，弹出工程向导第 1 步，填写工程名称，修改工程的存储位置。然后点击下一步，如图 1-5-5。



图 1-5-5

2. 设置第一个 POU 名称，选择需要的编程语言，如图 1-5-6。



图 1-5-6

3. 设置配置名称和内核类型，如图 1-5-6。

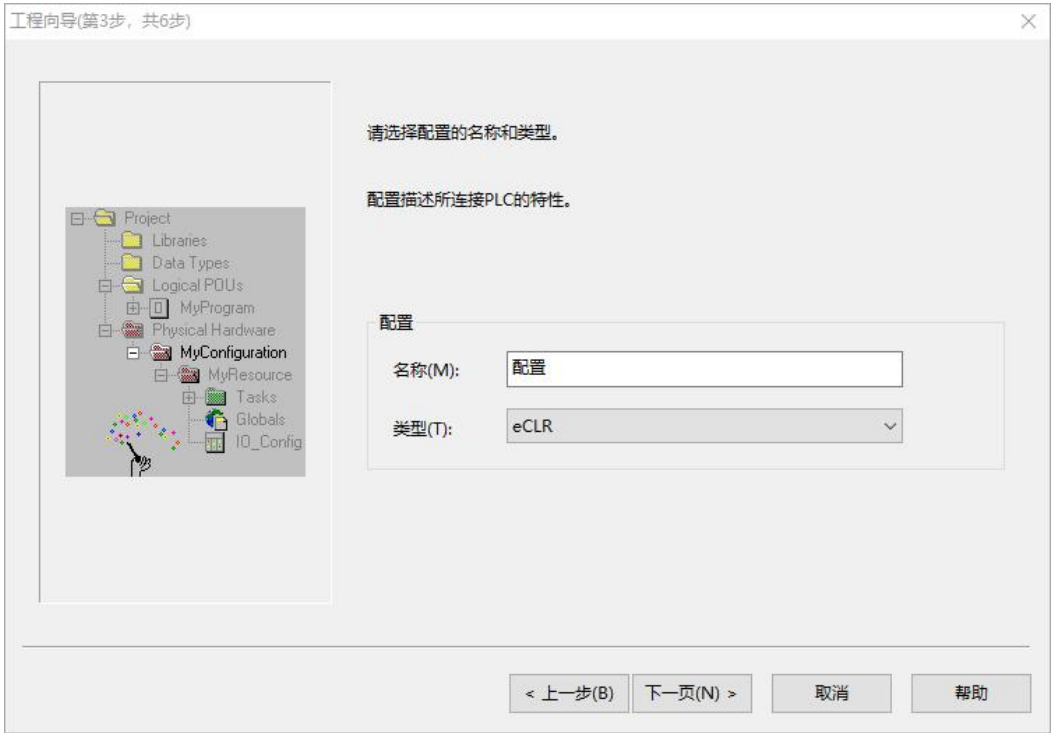


图 1-5-6

4. 点击下一步，出现工程向导第 4 步，选择 Truhigh_P600 如图 1-5-7。

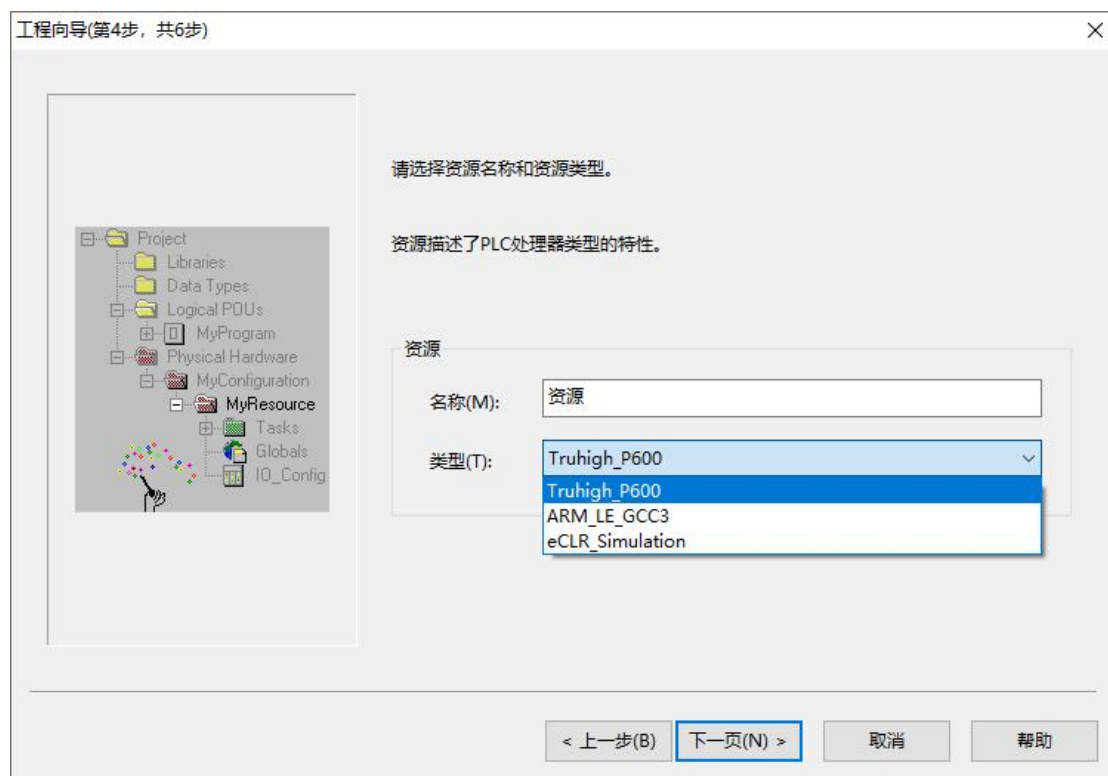


图 1-5-7

5. 点击下一步，出现工程向导第 5 步，如图 1-5-8。

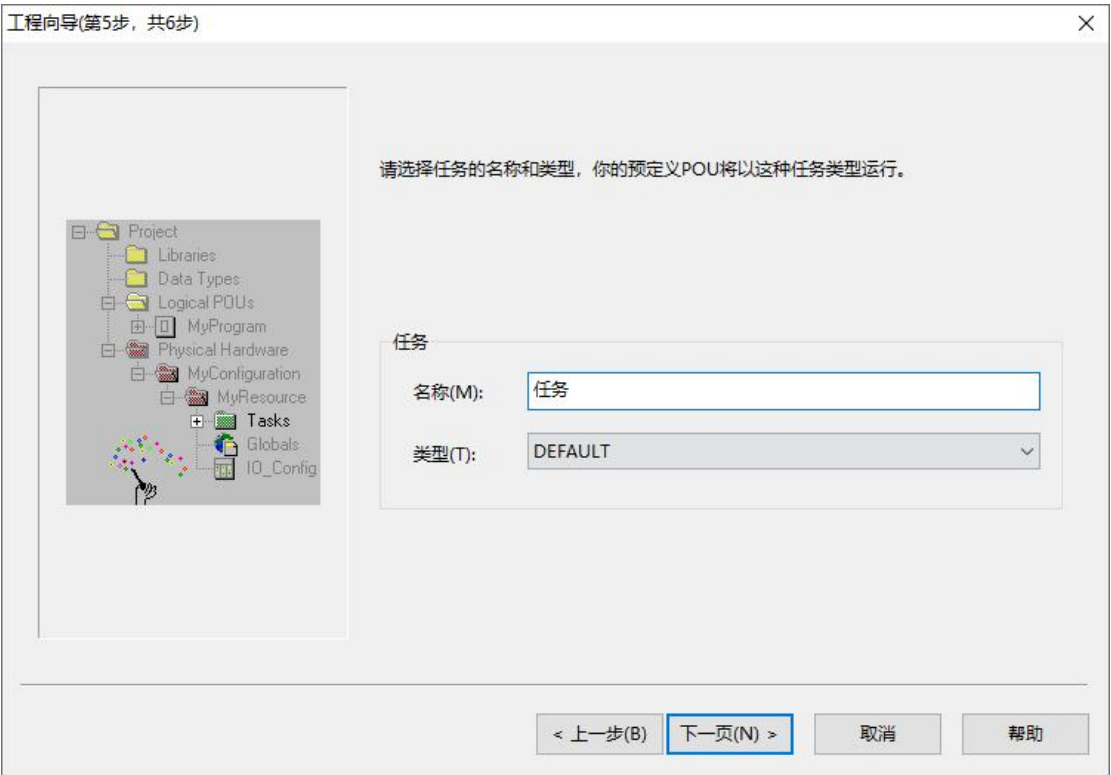


图 1-5-8

6. 点击下一步，出现工程向导第 6 步，如图 1-5-9。

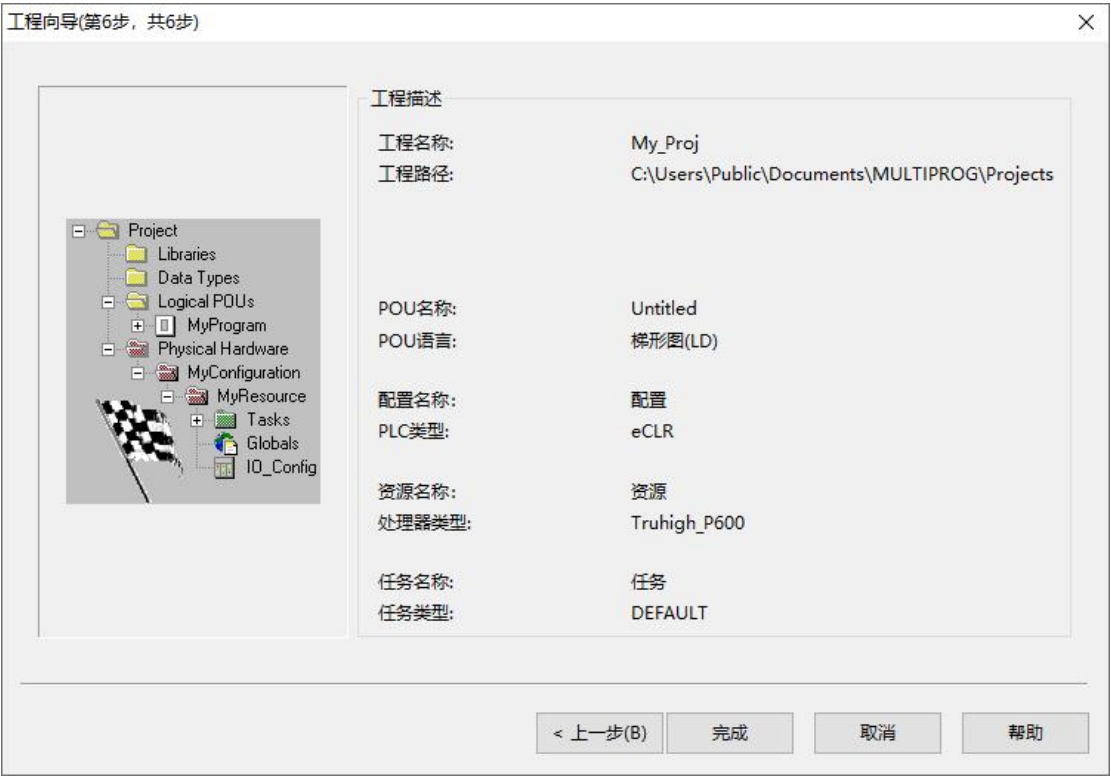


图 1-5-9

7. 在这里，描述了将要创建的工程文件，即前几步中所选的内容。点击完成，就创建了一个初步的工程了，如图 1-5-10。

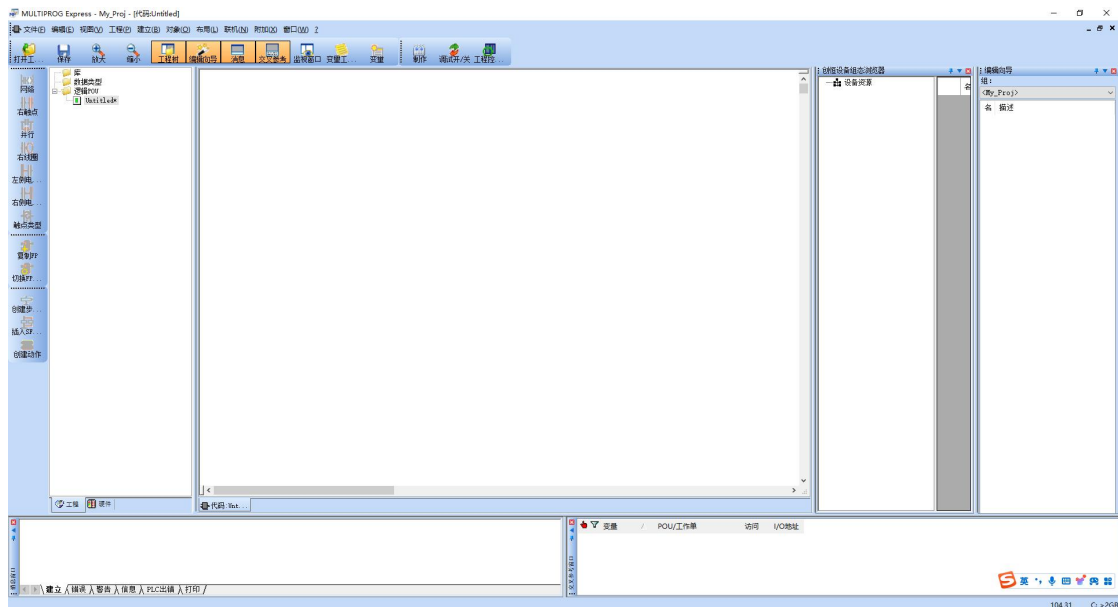


图 1-5-10

8. 在工程树中也出现工程选项卡和硬件选项卡，如图 1-5-11。

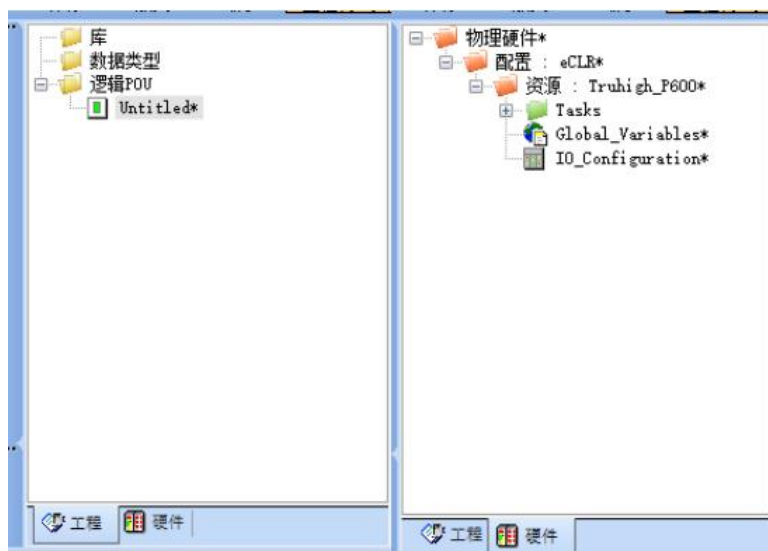


图 1-5-11

1.5.3 Truhigh P600 PLC 的配置

在工程树目录下选择“硬件”选项卡，在“资源”目录上点击右键，选择“属性”在“PLC/处理器”选项卡，确认是否是“Truhigh_P600”，如果不是需要重新选择，如图 1-5-12。

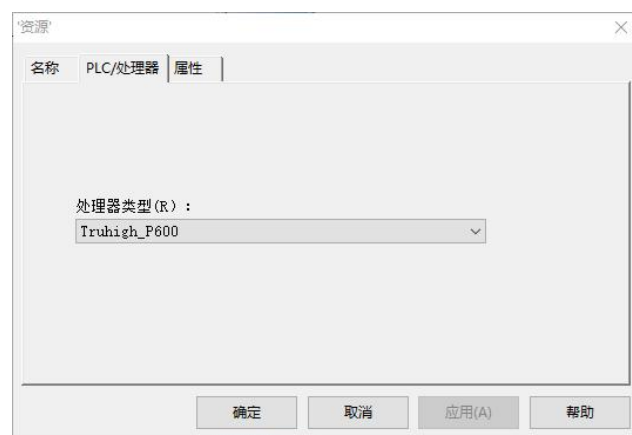


图 1-5-12

在工程树目录下选择“硬件”选项卡，在“资源”目录上点击右键，选择“设置”，修改通讯参数为 Truhigh P600 PLC 的 IP 地址，如：192.168.1.100，如图 1-5-13。



图 1-5-13

设置的 IP 地址要与 Truhigh P600 PLC 的 IP 地址一致，否则无法下载用户程序。修改控制器 IP 地址方法参见 1.4 章节。

1.5.4 创建一个程序

1. 创建一个工程后，点击 MULTIPROG 编程软件的编辑区域，然后点击编辑区域左侧的网络图标 ，在编辑区域出现一个梯形图的简单网络，左侧是一个常开触点，变量名是 C000，右侧是一个线圈，变量名是 C001，如图 1-5-14。



图 1-5-14

2. 双击常开触点 C000，出现触点/线圈属性对话框，在 I/O 地址 (S) 栏，输入 %IX6000.0，表示 PLC 的第一个数字量输入通道，点击确定，如图 1-5-15。

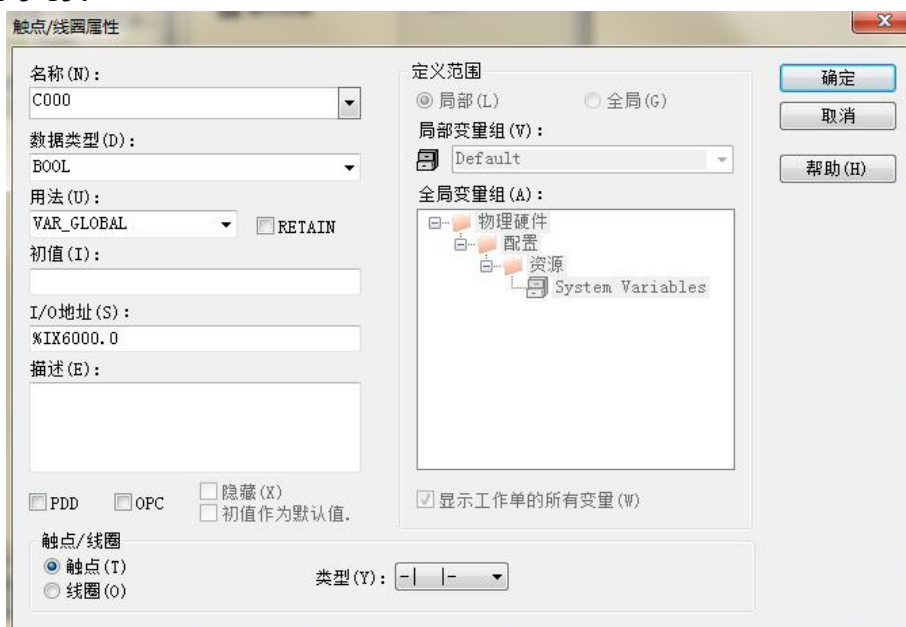


图 1-5-15

3. 双击梯形图中线圈 C001，出现触点/线圈属性对话框，在 I/O 地址 (S) 栏，输入 %QX6000.0，表示 PLC 的第一个数字量输出通道，点击确定，如图 1-5-16。

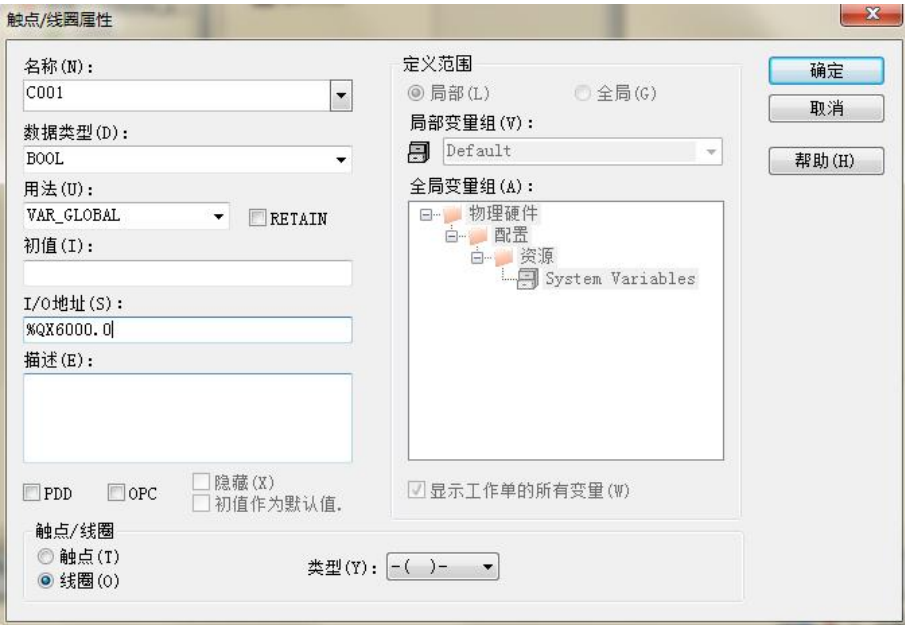


图 1-5-17

4. 然后在 MULTIPROG 编程软件的工具栏中点击  制作按钮，编译程序。
5. 编译过程中，在左下角的消息窗口显示编译过程，编译通过后，显示编译结果。如果出现“0 个错误, 0 个警告”，说明编译成功，否则再根据错误提示修改程序，直到没有错误和警告（警告不影响程序下载），如图 1-5-18。

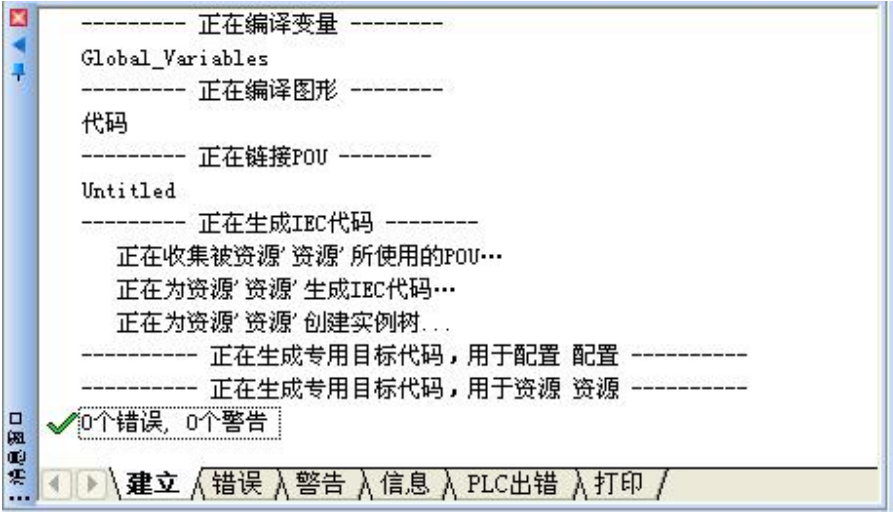


图 1-5-18

1.6 下载用户程序

1. 程序编译成功之后，就可以下装控制器了。在 MULTIPROG 编程软件的工具栏中点击工程控制对话框



，出现“资源”对话框，如图 1-6-1。



图 1-6-1

2. 点击“更多（M）”按钮，弹出对话框中，如图 1-6-2。



图 1-6-2

在下装选项中，勾选上“作为引导工程永久驻留（P）”，如果需要上传程序，则下载时必须勾选上“包括源代码（S）”，然后点击关闭，回到资源对话框。

3. 点击“下装（D）”，程序下载完成后，点击“冷启（O）”，如图 1-6-3。



图 1-6-3

4. 此时，PLC 开始运行用户程序，如图 1-6-4。



图 1-6-4

1.7 在线监视用户程序

MULTIPROG 支持在线监视调试用户程序，可以在 MULTIPROG 编程软件的工具栏中点击 调试开/关 。

1. 如果触点 C000 和线圈 C001 断开，则梯形图中为蓝色，如图 1-7-1。



图 1-7-1

双击梯形图中的 C000，在弹出的窗口中点击 强制 (O)，可以强制 PLC 的数字量 C000 为 1，继而驱动数字量输出 C001，如图 1-7-2。



图 1-7-2

如果 C000 不是 IO 点，而是中间变量，则强制功能不可用，可以使用 覆盖 (W)。

触点或者线圈导通之后，显示红色，如图 1-7-3。



图 1-7-3

第 2 章 Truhigh P600 系统的工作原理

Truhigh P600 系统根据用户编写的程序, 执行逻辑运算, 顺序控制, 定时, 计数与算术操作等面向用户的指令, 并通过数字或模拟式输入/输出控制各种类型的机械或生产过程。

本章节分为一下四节:

- 2.1 Truhigh P600 系统如何执行用户编写的程序
- 2.2 Truhigh P600 系统的数据存取
- 2.3 Truhigh P600 系统的保存数据
- 2.4 数据类型

2.1 Truhigh P600 系统如何执行用户编写的程序

Truhigh P600 系统是采用“顺序扫描，不断循环”的方式进行工作的。即在 Truhigh P600 系统运行时，CPU 根据用户按控制要求编制好并存于用户存储器中的程序，按指令步序号（或地址号）作周期性循环扫描，如无跳转指令，则从第一条指令开始逐条顺序执行用户程序，直至程序结束。然后重新返回第一条指令，开始新一轮新的扫描。在每次扫描过程中，还要完成对输入信号的采样和对输出状态的刷新等工作。

Truhigh P600 系统的一个扫描周期必经输入采样、程序执行和输出刷新三个阶段。

- 输入采样阶段：首先以扫描方式按顺序将所有暂存在输入锁存器中的输入端子的通断状态或输入数据读入，并将其写入各对应的输入状态寄存器中，即刷新输入。随即关闭输入端口，进入程序执行阶段。
- 程序执行阶段：按用户程序指令存放的先后顺序扫描执行每条指令，经相应的运算和处理后，其结果再写入输出状态寄存器中，输出状态寄存器中所有的内容随着程序的执行而改变。
- 输出刷新阶段：当所有指令执行完毕，输出状态寄存器的通断状态在输出刷新阶段送至输出锁存器中，并通过一定的方式（继电器、晶体管或晶闸管）输出，驱动相应输出设备工作。

2.2 Truhigh P600 系统的数据存取

对于 IO 数据的存取必须指定地址，地址以 IEC61131-3 规定的“%”开头，后续是位置前缀，大小前缀，然后用整数表示字节地址，如%IX0.0 表示输入映射区第 1 个字节的第 1 个位。

中间变量可以不指定地址，也可以指定 M 存储区，如表 2-1。

表 2-1

序号	前缀		定义	约定类型	举例
0	地址前缀	%	地址标识符		
1	位置前缀	I	输入映射区		%IX0.0 第 1 路 DI, %IW64 第 1 路 AI
2		Q	输出映射区		%QX0.0 第 1 路 DO, %QW64 第 1 路 AO
3		M	中间变量映射区		
4	大小前缀	X	位	BOOL	%MX3.0.0
5		无	位	BOOL	%IX0.0 即 %I0.0
6		B	字节（8 位）	BYTE	%MB3.0
7		W	字（16 位）	WORD	%MW3.0
8		D	双字（32 位）	DWORD	%MD3.0
9		L	长型（64 位）	LREAL	%ML3.0

2.3 Truhigh P600 系统的保存数据

- Truhigh P600 系统具有 2M 程序存储空间。
- Truhigh P600 系统具有 2M 数据存储空间。
- Truhigh P600 系统具有 2000 字节的数据保存空间。

2.4 数据类型

数据类型决定了变量的格式、位数、可能值的范围和初始值。

IEC 61131-3 区分下面的数据类型：

- 基本数据类型
- 类属数据类型
- 用户自定义数据类型

2.4.1 基本数据类型

基本数据类型是不可以再分解为其它类型的数据类型，常用的数据类型如表 2-2 所示。

表 2-2

数据类型	描述	大小	范围	默认初始值
BOOL	布尔	1	0...1	0
SINT	短整数	8	-128...127	0
INT	整数	16	-32,768...32,767	0
DINT	双整数	32	-2,147,483,648 至 2,147,483,647	0
USINT	无符号短整数	8	0 至 255	0
UINT	无符号整数	16	0 至 65,535	0
UDINT	无符号双整数	32	0 至 4,294,967,295	0
REAL	实数	32	-3.402823466 E+38 (约 7 位数字) 至 -1.175494351 E-38 (约 7 位数字) 以及 +1.175494351 E-38 (约 7 位数字) 至 +3.402823466 E+38 (约 7 位数字)	0.0
LREAL	长实数	64	~ -1.798 E+308 (约 15 位数字) 至 ~ -2.225 E-308 (约 15 位数字) 以及 ~ +2.225 E-308 (约 15 位数字) 至 ~ +1.798 E+308 (约 15 位数字)	0.0
TIME	持续时间	32	0... 4,294,967,295 毫秒	t#0s
BYTE	长度为 8 的位串	8	0...255 (16#00...16#FF)	0
WORD	长度为 16 的位串	16	0...65,535 (16#00...16#FFFF)	0
DWORD	长度为 32 的位串	32	0...4,294,967,295 (16#00....16#FFFFFFFF)	0

2.4.2 类属数据类型

类属数据类型是包括了基本数据类型的分级组的数据类型。
如 ANY_INT 包括 DINT、INT、SINT、UDINT、UINT 和 USINT 等基本数据类型。

类属数据类型如图 2-4-1。

ANY					
ANY_NUM			ANY_BIT	STRING	TIME
ANY_REAL	ANY_INT		BOOL		
REAL	SINT	USINT	BYTE		
LREAL	INT	UINT	WORD		
	DINT	UDINT	DWORD		

图 2-4-1

2.4.3 用户自定义数据类型

用户自定义的数据类型必须在某个数据类型工作单中声明，可以通过使用一个“TYPE ...END_TYPE ”声明块来完成。用户自定义数据类型是一种衍生数据类型。

衍生数据类型可以是结构，或者是数组。

- 数组：
数组是一个单一的数据类型对象的集合，同基本的数据一样具有唯一的名字，对单个的对象并没有被命名，但是用户可以通过它在数组的位置对它进行访问。

声明数组的例子如下：

```
TYPE
  DATA2:ARRAY [0..100] OF REAL;
END_TYPE
```

- 结构：
结构是多个不同的数据类型对象的集合，同基本数据一样它有唯一的名字，结构的成员是基本数据类型或数组类型，也可以是另一个结构，或者嵌套。

声明一个结构的例子如下：

```
TYPE
  DATA3 :
    STRUCT
      X1:INT;
      X2:REAL;
      X3:BOOL;
    END_STRUCT;
END_TYPE
```

- 字符串数据类型：
用户自定义的字符串数据类型是具有可变数目的字符的字符串。声明一个用户自定义的字符串时，其长度被放在该数据类型后面的括号内。

用户定义字符串数据类型声明的例子。

```
TYPE
  STRING10 :STRING(10) ;
END_TYPE
```

在这个例子中，字符串的长度是 10 个字符。
限制：最短的合理字符串长度为 1，最长的合理字符串长度为 32766。

2.4.4 常量数据的表示

表 2-3

数据类型	描述	位长度	表示方法举例
BOOL	布尔	1	BOOL#0
SINT	短整型	8	SINT#-128
INT	整型	16	INT#-32,768
DINT	双整型	32	DINT#-2,147,483,648
USINT	无符号短整型	8	USINT#255
UINT	无符号整型	16	UINT#65,535
UDINT	无符号双整型	32	UDINT#4,294,967,295
REAL	实数	32	REAL#3.1415629
LREAL	长实数	64	LREAL#3.1415629
TIME	时间	32	T#10MS、T#10S、T#10M、 T#10H、T#10D、T#1D_10H
DATE	日期		D#2011-07-24
TIME OF DATE	时刻		TOD#15:23:45,55
TIME AND DATE	日期和时刻		DT#2011-07-24-15:23:45,55
BYTE	字节	8	BYTE#16#FF
WORD	字	16	WORD#16#FFFF)
DWORD	双字	32	DWORD#16#FFFFFFFFF)
STRING	字符串		'A'

第 3 章 MULTIPROG 软件

MULTIPROG 编程软件模型用分层结构表示，软件模型描述了各个部分之间的相互关系，包括配置、资源、任务、程序组织单元、全局变量、IO 配置等。编程工程中可将一个复杂的程序分解成为若干个小程序或者模块，也可以讲多个独立的程序同时下载、运行，或者将程序分成多个任务去执行，提高了程序的模块化和运行效率。

本章分为以下三节：

- 3.1 用户界面
- 3.2 工程树——硬件
- 3.3 工程树——工程

3.1 用户界面

本程序用户界面由多个控件组成，这些控件可被显示或隐藏(通过‘查看’菜单和工具栏图标)，并且可以根据需要来安排，如：解锁、移动和重新锁定它们。

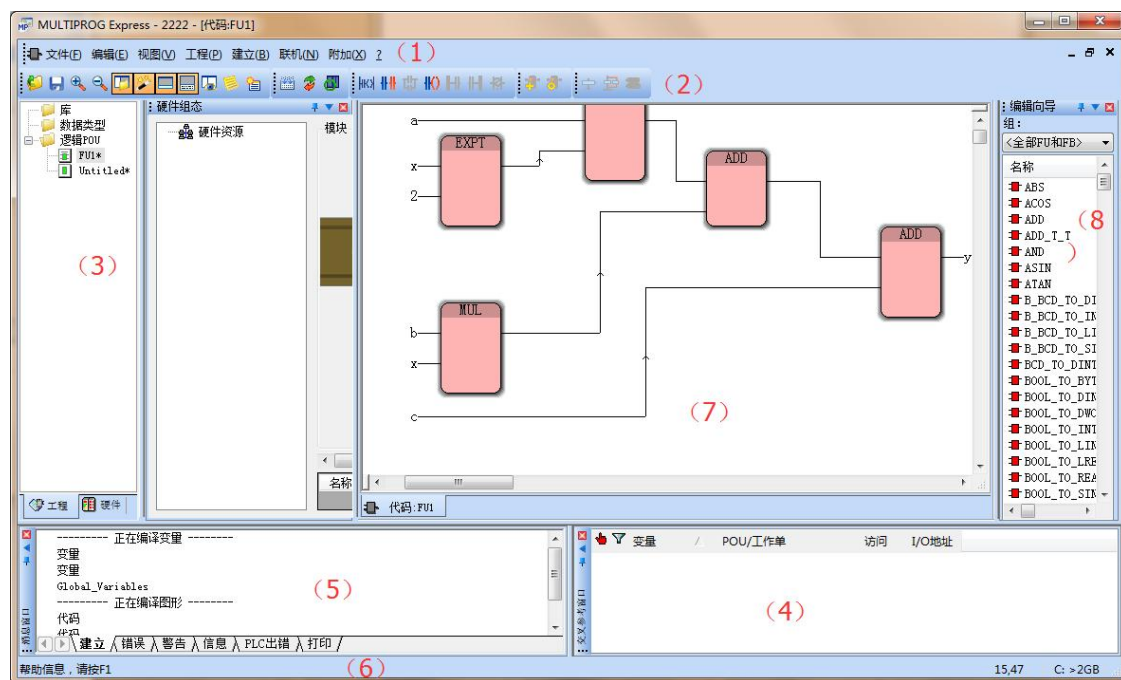


图 3-1-1

- (1) 菜单
- (2) 工具栏
- (3) 工程树
- (4) 交叉参考窗口
- (5) 消息窗口
- (6) 状态栏
- (7) 工作区
- (8) 编辑向导

没有出现在下面的屏幕拷贝中的控件包括“监视窗口”和“逻辑分析器”。

3.1.1 菜单栏

菜单栏里面包含了 Multiprog 几乎所有的命令，说明如下。

注意，取决于正在处理的程序部分或编辑器，某些特定的菜单项可能无法访问。

3.1.1.1 文件

‘文件’菜单可用于创建、保存 和 压缩/解压缩工程。它也包含用于打印、打印设置和打印预览等。



图 3-1-2

➤ 文件/新建工程

此菜单命令使用工程向导，创建新的工程。点击“新建工程”后，弹出“新建工程向导”对话框，如图 3-1-3。

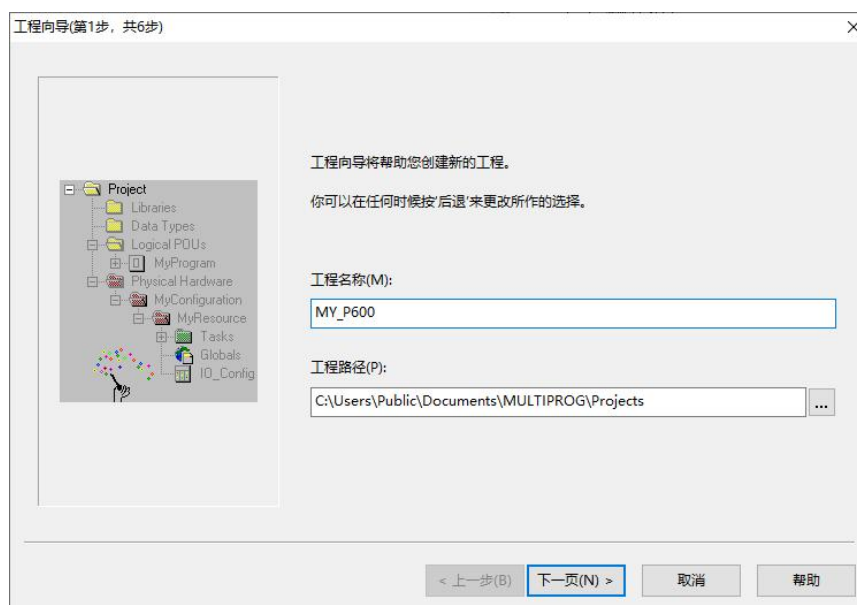


图 3-1-3

➤ 文件/打开工程/解压缩工程

“打开工程/解压缩工程”菜单命令可以打开一个工程或者解压缩一个工程。

✧ 打开一个工程：

点击“打开工程/解压缩工程”菜单命令，弹出“打开/解压缩工程”对话框。

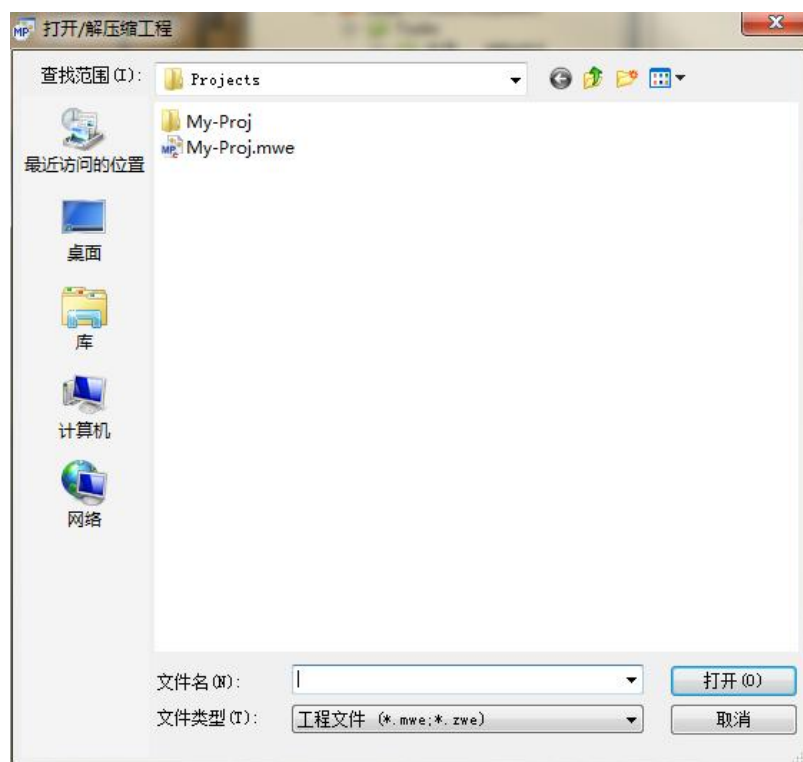


图 3-1-4

浏览查找所需的工程或归档文件(*.mwe 或 *.zwe), 并点击‘打开’。

每个工程都由一个 *.mwe 文件和一个相应文件夹(包含了与工程有关的各种文件)组成。

✧ 解压缩归档文件:

选择菜单项‘文件 > 打开工程/解压工程...’, 出现‘打开/解压工程’。在‘文件类型’列表框中选择‘压缩了的工程文件(*.zwe)’选项。

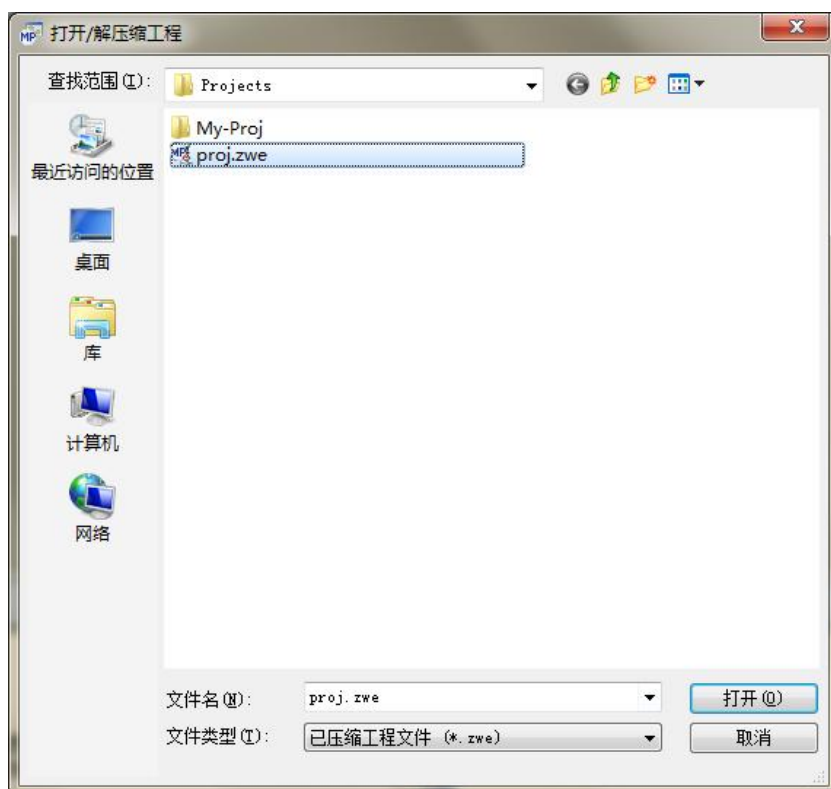


图 3-1-5

点击‘打开’按钮来启动解压缩。出现‘解压工程’对话框。

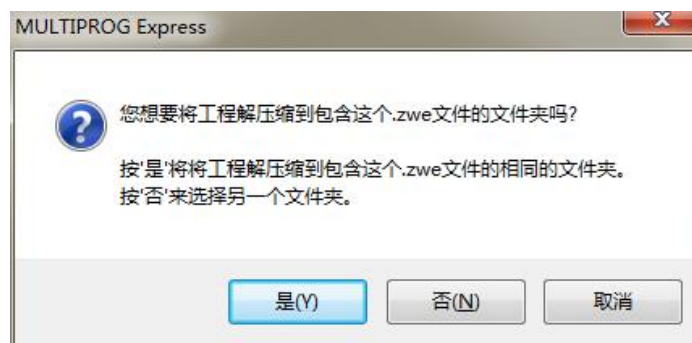


图 3-1-6

在‘解压工程’对话框中，按：

- ✓ ‘是’：将工程(归档文件)解压到(对话框中列出的)一个 zwe 文件的子目录中，其名称等同于 zwe 文件。另外，解压过程完成之后，带有 zwe 文件的目录将包含工程文件(*.mwe)和一个相应的文件夹，该文件夹包含工程相关文件(见下面的注释)。例如，如果您解压归档文件"Archive.zwe"，那么将创建工程文件"Archive.mwe"和文件夹"Archive"。
- ✓ ‘否’：选择另一个位置用于解压工程。在这种情况下，出现带有默认工程目录的文件选择对话框。可以确定默认工程名称(与 zwe 文件同名)和默认目录，也可以浏览您计算机上其它目录，输入一个新的工程名称(*.mwe)，并确定对话框。
- ✓ ‘取消’：忽略解压过程。

如果这个含有要被解压 zwt 文件的目录已经包含了一个具有相同文件

名称的工程，那么会出现一个对话框，该对话框询问您是否想要以压缩的工程来覆盖现有工程。点击‘是’，用压缩的归档文件工程数据来覆盖已有工程，或者点击‘否’。在第二种情况下，出现‘工程解压为’对话框，在此，如有需要，您必须选择解压工程的其它工程名称和其它目录。

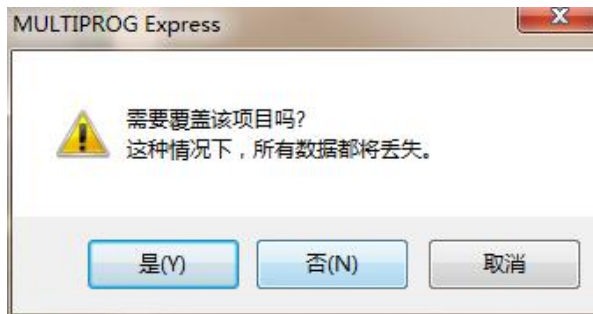


图 3-1-7

如果该归档文件包含了一个已存在于你的计算机上的压缩了的固件库，则会出现一个消息窗口，询问你是要用压缩的 FW 库覆盖现存 FW 库，还是要为将被解压的 FW 库定义一个新名称。

在消息对话框中：

- ✓ - 输入一个新的 FW 库名称，并点击‘确定’，会将该库解压缩到一个新名称下。
- ✓ - 点击‘跳过’，如果您不想解压缩这个固件库。
- ✓ - 点击‘跳过全部’，如果您不想解压缩任何 FW 库。



提示：

在解压缩了工程的一个固件库之后，您必须重新连接该工程与所解压缩的库。

解压缩归档文件之后，必须使用‘重建工程’。

➤ 文件/工程另存为/压缩为

“工程另存为/压缩为”文件菜单命令可以将当前工程保存到一个不同的文件名以及/或者一个不同的目录，或者将当前工程压缩为归档文件。

✧ 工程另存为：

1. 点击“工程另存为/压缩为”文件菜单命令，弹出“工程另存为/压缩为”对话框；

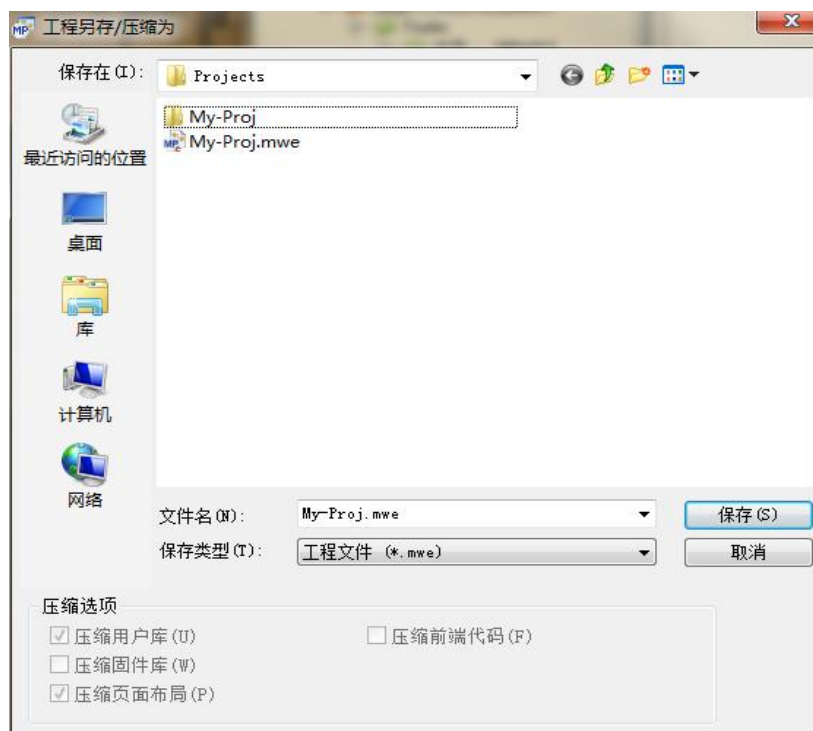


图 3-1-8

2. 选择一个目录，输入一个文件名，并点击‘保存’。

✧ 压缩为：

要想创建工程备份，可以将工程压缩为归档文件。这个压缩了的归档文件包含了属于该工程的所有文件，即：工程文件‘projectname.mwe’本身、代码本体文件、变量声明以及一些用于将工程从归档文件恢复正常所需的内部文件。

1. 点击“工程另存为/压缩为”文件菜单命令，弹出“工程另存为/压缩为”对话框。为激活可用的压缩选项，首先要在‘文件类型’列表框中选择‘压缩了的工程文件 (*.zwe)’一项。

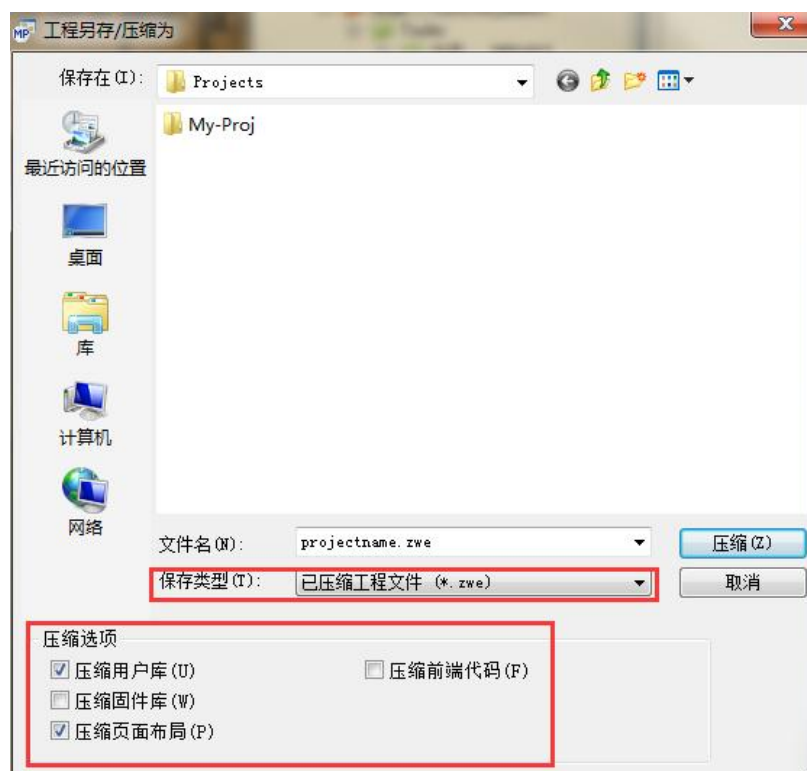


图 3-1-9



提示:

基本的压缩操作能将下列文件压缩为归档文件：所有的 POU、数据类型工作单和所有的配置数据。结果是一个很小的归档文件。

当压缩工程时，可以将库、前端的代码(独立于硬件的编译输出)和页面布局，连同您的工程文件，一起压缩到同一个归档文件中。为此，‘工程另存为/压缩为’对话框提供了几个选项，可以结合压缩操作来选择它们。

2. 点击‘压缩’按钮。压缩过程完成的消息会显示在一个消息框内。



图 3-1-10

➤ 文件/关闭工程

点击“关闭工程”菜单命令，可以关闭当前编辑的工程。如果工程为保存，会弹出提示保存的对话框。



图 3-1-11

➤ 文件/删除工程

可以删除一个保存的工程。

✧ 点击“删除工程”菜单命令，弹出“删除工程”对话框。

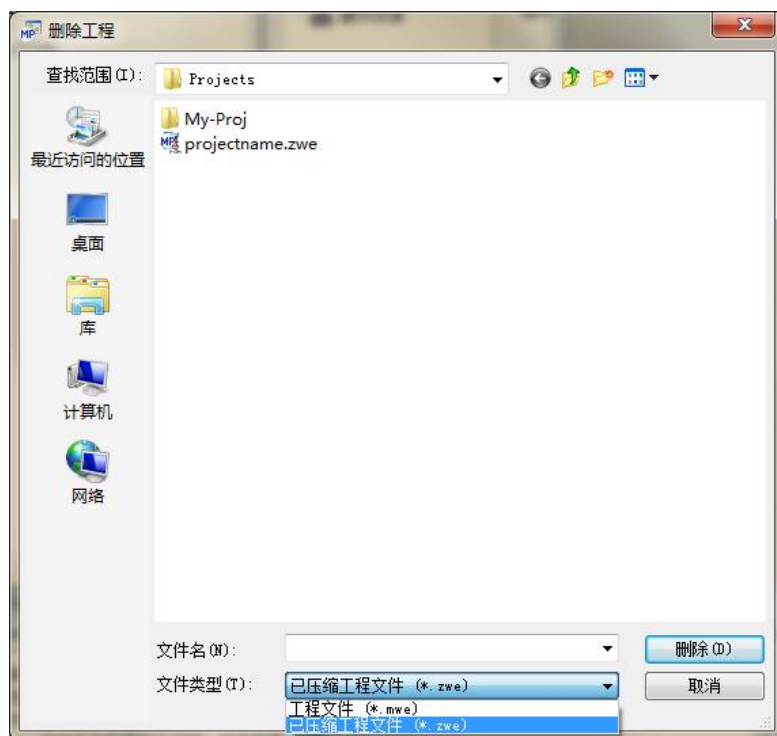


图 3-1-12



提示:

如图要删除工程文件，在文件类型中选择：工程文件 (*.mwe)。

如果要选择压缩工程，在文件类型中选择：已压缩工程文件(*.zwe)。

✧ 点击删除按钮。

➤ 文件/保存

点击“保存”菜单命令，可以保存当前编辑的已修改工作单。

➤ 文件/保存全部

点击“保存全部”菜单命令，可以全部的已修改工作单。

➤ 文件/关闭

点击“关闭”菜单命令，可以关闭当前编辑的工作单。

➤ 文件/打印

点击“打印”菜单命令，可以打开“打印”对话框，可以设置打印机，打印范围和打印页数。

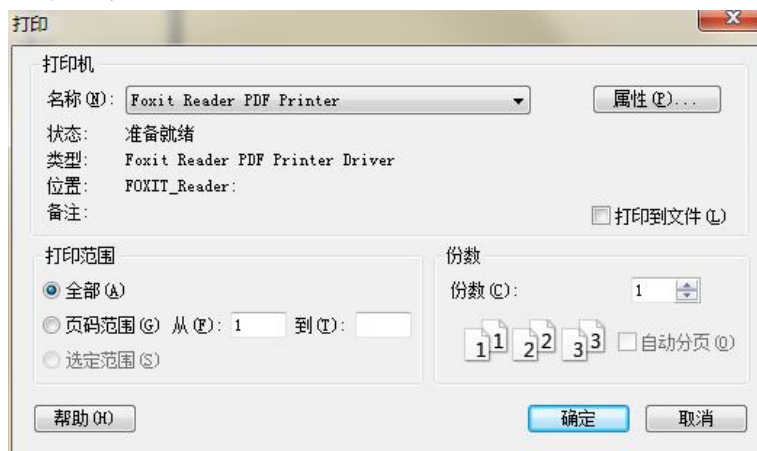


图 3-1-13

➤ 文件/打印预览

点击“打印预览”菜单命令，可以打开预览打印的内容，样式。

➤ 文件/打印设置

点击“打印设置”菜单命令，可以打开“打印设置”对话框，在对话框可以选择打印机，设置纸张类型和方向。



图 3-1-14

➤ 文件/打印工程

点击“打印工程”菜单命令，可以打开“打印工程”对话框。



图 3-1-15

对话框元素	含义
范围 ‘全部’或‘选定的’	指定将被打印的部分： ‘全部’- 如果激活了这个单选按钮，则要打印工程树中的整个工程(包括所有子树)。 ‘所选择的’- 如果激活了这个单选按钮，则只打印那些在调用此对话框之前所标记的工程树中的节点。
页码范围 ‘全部’或‘页码范围’	指定将要打印的页码： ‘全部’- 如果激活该单选按钮，那么将打印工作单的全部页面。 ‘页码范围’- 输入您想要打印工作单的页码。
‘打印’区域中的复选框	指定将要打印工程中的哪些部分。
变量显示模式(LD)	如果您的编程系统中安装了固定格式梯形图编辑器，那么才可以使用该对话框区域。
打印	使用该对话框中所作的设置，开始打印处理。
保存设置	通过按下这个按钮，系统保存工程的打印设置。当下次打开打印对话框时，当关闭工程时自动显示保存的这些设置。

➤ 文件/关闭

关闭当前编辑的工程。

3.1.1.2 编辑

‘编辑’菜单包含了编辑所必需的全部命令，如标记、选择不同工作模式或剪切和粘贴。另外，它提供了在文本工作单和图形工作单中搜索和替换文本串的功能。依赖于活动的编辑器，该菜单也包括用于处理对象（例如选择、移动、使能分支模式、创建图片等等）的命令。



图 3-1-16

- 编辑/撤销
点击“撤销”菜单命令，撤销在编辑器中进行的最后的编辑步骤。
- 编辑/恢复
点击“恢复”菜单命令，恢复在撤销的操作。
- 编辑/剪切
点击“剪切”菜单命令，将选择的内容移动到剪贴板。
- 编辑/复制
点击“复制”菜单命令，将选择的内容复制到剪贴板。

➤ 编辑/粘贴

点击“粘贴”菜单命令，将剪贴板的内容复制到指定位置。

➤ 编辑/插入

在工程树中选择不同的目录，点击“插入”菜单命令弹出不同的“插入对话框”。例如在“逻辑 POU”上点击“插入”命令弹出插入 POU 的对话框。



图 3-1-17

➤ 编辑/查找

点击“查找”菜单命令，打开“查找/替换”对话框。

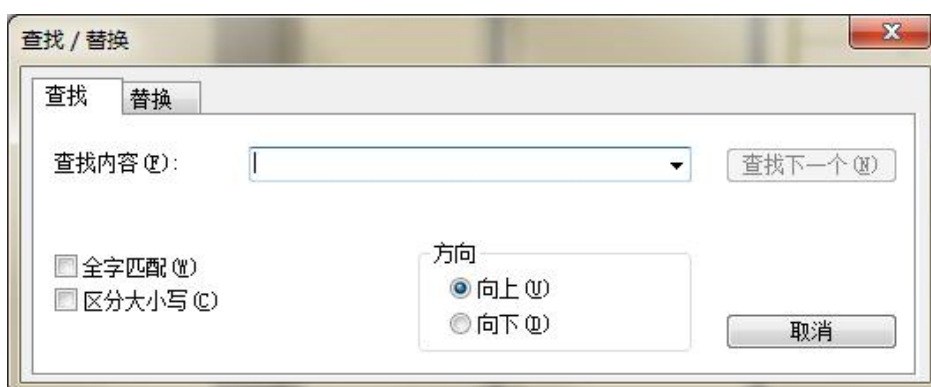


图 3-1-18

➤ 编辑/替换

点击“替换”菜单命令，打开“查找/替换”对话框。



图 3-1-19

➤ 编辑/查找上一个（局部）

➤ 编辑/查找下一个（局部）

如果想再次搜索相同的文本元素(搜索方向既可以是‘向上’，也可以是‘向下’)，但是‘查找/替换’对话框已关闭，不必重新打开此对话框。只需在‘编辑’菜单中，选择‘(局部)查找上一个’或‘(局部)查找下一个’菜单项之一。在‘查找内容’对话框区域中所键入的搜索字符串被再次搜索，并在活动的工作单内被标记了出来。

➤ 编辑/全局查找

点击“全局查找查找”菜单命令，打开“查找/替换”对话框。查找可以选择查找的类型、变量、代码工作单和描述。



图 3-1-20

➤ 编辑/全局替换

点击“全局替换”菜单命令，打开“查找/替换”对话框。可以替换类型、变量、代码工作单和描述。

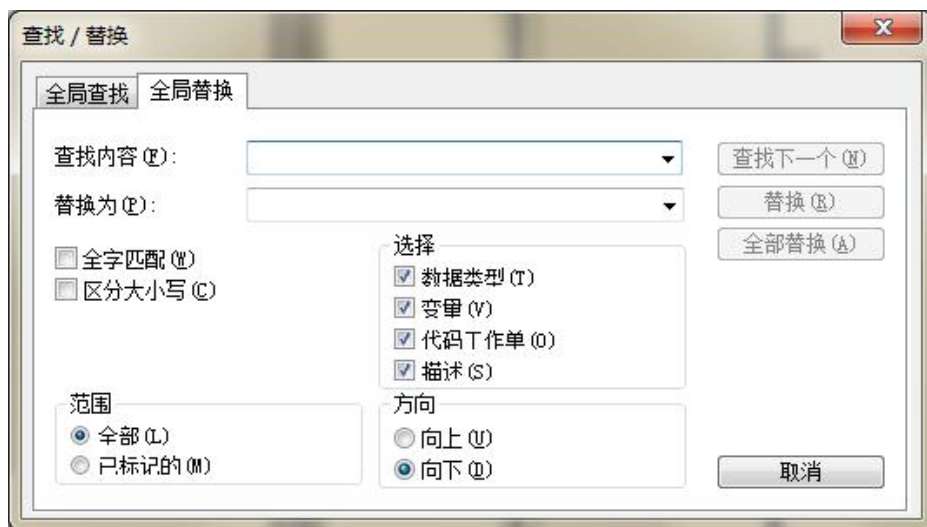


图 3-1-21

➤ 编辑/查找上一个（全局）

➤ 编辑/查找下一个（全局）

如果想再次搜索相同的文本元素(搜索方向既可以是‘向上’，也可以是‘向下’)，但是‘查找/替换’对话框已关闭，不必重新打开此对话框。只需在‘编辑’菜单中，选择‘(局部)查找上一个’或‘(局部)查找下一个’菜单项之一。在‘查找内容’对话框区域中所键入的搜索字符串被再次搜索，并在活动的工作单内被标记了出来。

➤ 编辑/更新 FB/FU

修改了 FB/FU 之后，必须在调用工作单之后更新这个块。

3.1.1.3 视图

‘视图’菜单用于隐藏或显示用户界面的不同窗口和控件（工程树窗口、消息窗口、交叉参考窗口、监视窗口、逻辑分析器、编辑向导）以及状态栏，并用于切换工作区(如果已经实施)。而且，还有一些命令可用于打开变量表格工作单(它们与当前活动的代码工作单相关)，并用于打开全局变量表格工作单。



图 3-1-22

- 视图/工程树窗口
用于显示或隐藏“工程树窗口”。



图 3-1-23

- 视图/消息窗口
用于显示或隐藏“消息窗口”。

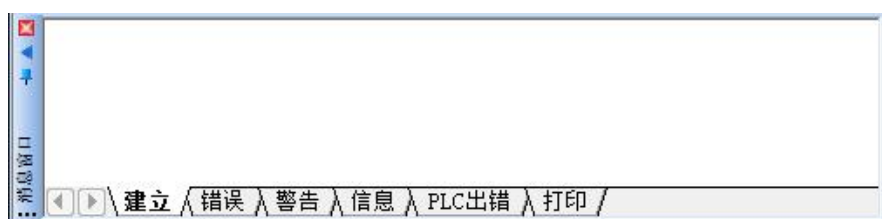


图 3-1-24

- 视图/编辑向导
用于显示或隐藏编辑向导。



图 3-1-25

- 视图/交叉参考窗口
用于显示或隐藏“交叉参考窗口”。



图 3-1-26

- 视图/监视窗口
用于显示或隐藏“监视窗口”。



图 3-1-27

- 视图/逻辑分析器
用于显示或隐藏“逻辑分析器”。



图 3-1-28

- 视图/状态栏
用于显示或隐藏“状态栏”。

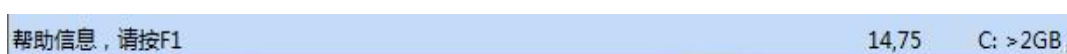


图 3-1-29

- 视图/创恒设备组态浏览
用于显示或隐藏“硬件组态”。



图 3-1-30

- 视图/打开变量工作单
用于在程序代码区打开变量工作单。
- 视图/全局变量声明
用于打开全局变量表。

3.1.1.4 工程

‘工程’菜单用于插入(多个)新的 POU 以及一个数据类型工作单，并用于声明库。

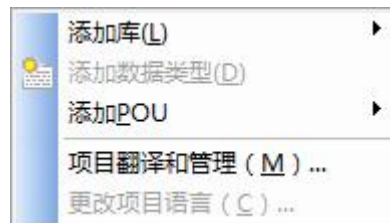


图 3-1-31

➤ 工程/添加库

‘库’子树含有您的工程中所包含的所有库。它们既可以是用户自定义库，也可以是固件库。

固件库是那些包含了由 PLC 制造商所提供的 POU 的库。固件库的文件扩展名为*.fwl。

✧ 打开“添加库”->“固件库”

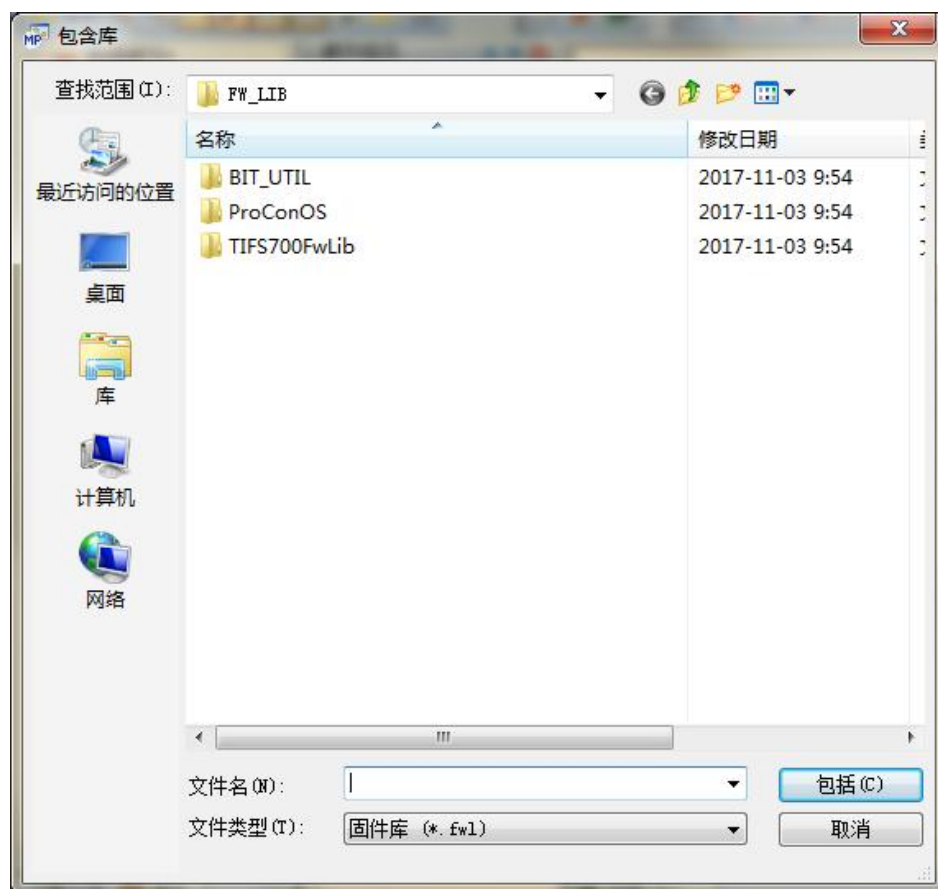


图 3-1-32

用户库是“普通的”工程，其 POU 将被重复使用。作为用户库而包括进来的工程中的程序、功能块、功能和用户自定义数据类型都可以在当前工程中重复使用。

✧ 打开“添加库”->“用户库”

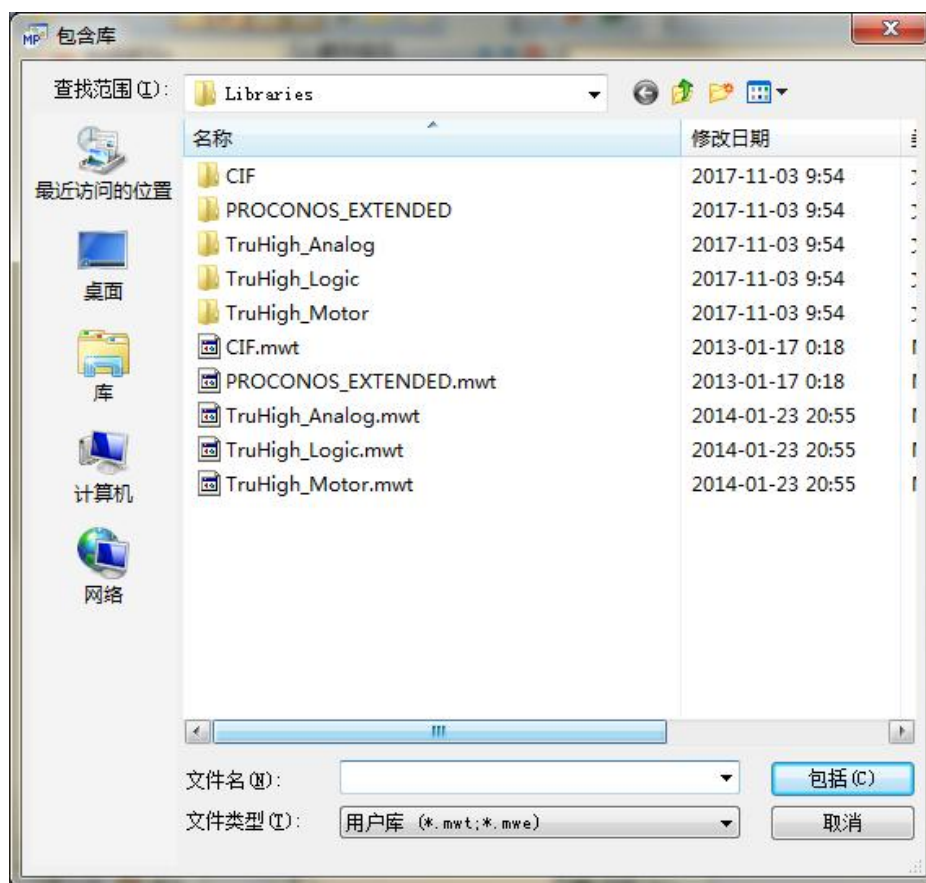


图 3-1-33

用户库的扩展名为*.mwe（通过 Express 版本创建工程）和*.mwt（通过完全版编程系统创建工程）。

- 工程/添加数据类型
点击“添加数据类型”，弹出插入数据类型对话框。



图 3-1-34

➤ 工程/添加 POU

点击“添加 POU/功能块 或者 功能 或者 程序”，弹出插入对话框。



图 3-1-35

➤ 工程/项目翻译和管理

可以添加语言、删除语言、对翻译文件进行导入、导出。



图 3-1-36

- 工程/更改项目语言
添加语言之后，可以更改项目语言。

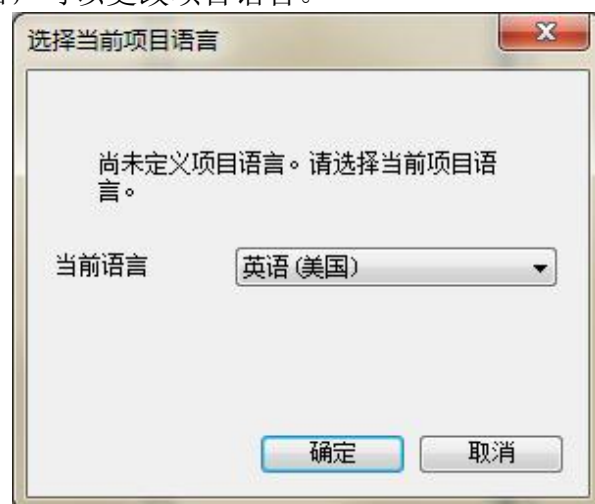


图 3-1-37

3.1.1.5 建立

‘建立’菜单由不同的命令组成，用于启动编辑之后的编译，用于显示编译过程中检测出的错误，用于建立交叉参考，以及用于去除未使用的局部变量和 FB 实例的声明。



图 3-1-38

➤ 建立/制作

点击“制作”命令，已更改的工作单会被编译、连接，并将产生更改后的 PLC 代码。在成功地执行完这几步之后，所修改的工程就具备了下装到 PLC 的条件。

➤ 建立/修补 POU

修补 POU 意味着：在一步之内，编译你在工程中所做的修改、产生相关的代码、并自动下装到 PLC。修补过程中，PLC 继续运行，即当编译和下装所修改内容时，不中止 PLC 上的代码执行。

➤ 建立/编译工作单

‘编译工作单’菜单项命令编译一个工作单，以便在编辑过程中检查语法。使用‘编译工作单’不产生代码。

➤ 建立/重建工程

使用‘重建工程’命令，将编译和连接所有工作单。检查出的错误和警告显示在消息窗口。在通过语法检查之后，会自动产生 IEC 代码和专用于具体 PLC 的代码。这样，工程就具备了下装到 PLC 上的条件。

仅当‘制作’产生了编译错误，或者解压缩了没有前端代码的工程，才应该使用‘重建工程’命令。

➤ 建立/停止编译

建立/终止编译命令。

➤ 建立/转到下一个错误

转到下一个错误。

- 建立/转到上一个错误
转到上一个错误。
- 建立/建立交叉参考
点击“建立交叉参考”菜单命令，在“交叉参考列表”中显示当前工程内使用的所有变量、功能块、动作、转换、步、跳转、标号以及连接符。这对于调试和错误隔离来说是一个很有用的工具。
- 建立/去除未使用的变量和 FB 实例
使用“去除未使用的变量和 FB 实例”菜单命令可以删除未使用的变量和 FB 实例。

3.1.1.6 联机

‘联机’菜单提供了用于调试工程和调用资源控制对话框的命令。



图 3-1-39

➤ 联机/调试

点击“调试”菜单命令，可以打开或者关闭在线调试功能。

➤ 联机/工程控制

点击“工程控制”菜单命令，可以打开控制 PLC 下载、启动、停止等功能的对话框。



图 3-1-40

➤ 联机/下装修改部分

点击“下载修改部分”菜单命令，PLC 会保持其状态的情况下修改 PLC 程序。

➤ 联机/取消下载

点击“取消下载”菜单命令，可以取消 PLC 的下载。

➤ 联机/显示活动步

在联机调试 SFC 程序时，点击“显示活动步”菜单命令，可以显示当前活动步。

➤ 联机/逻辑分析器

‘逻辑分析器’含有一些用于控制逻辑分析器的命令。

3.1.1.7 附加

在‘附加’菜单中，可以启动可选的已经实现的工具，例如：页面布局编辑器，‘选项’菜单项允许您定制编程系统的外观。

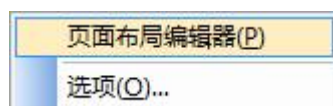


图 3-1-41

➤ 附加/页面布局编辑器

页面布局编辑器是用于创建和编辑页面布局的工具，页面布局用于打印工程文档。一个页面布局代表一个工作单内容打印模板。

➤ 附加/选项

可以根据您的个人需要，整理或隐藏(自动隐藏功能)窗口和控制(编辑向导、工程树窗口、交叉参考窗口等等)。



图 3-1-42

3.1.1.8 帮助

‘?’菜单包含了用于调用帮助的所有命令。



图 3-1-43

- ?/内容
可以打开帮助系统。
- ?/关于 IEC61131
查看 IEC 61131 基础说明和标准。
- ?/标准 FB/FU 帮助
可以查看 IEC 61131 标准所提供的函数和功能块的描述。
- ?/PLC 帮助
这个帮助部分提供了对所有 PLC 类型都有效的通用信息。

3.1.2 工具栏

通过工具栏图标快速地访问频繁使用的功能，这些功能也可以通过菜单或者通过预定义的快捷键来调用。



图 3-1-44

将光标悬浮在一个符号上的同时，那个符号的简短描述就马上出现状态栏中。该描述也以工具提示的形式显示。

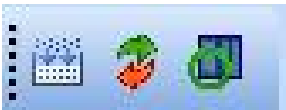


图 3-1-45



图 3-1-46

每个工具栏都可以被拉出来，放到别处，以及移动，如下面的图所显示：双击点状线，将工具栏从界面拉出来。

要更改窗口的尺寸，请将光标放在窗口边框上，按住鼠标按钮，并拖拽窗口边框。您可以将窗口移动到屏幕上您喜欢的任何位置，这要通过将光标放到标题栏上，按住鼠标按钮，并移动窗口来完成。

工具栏样式如图 3-1-49：

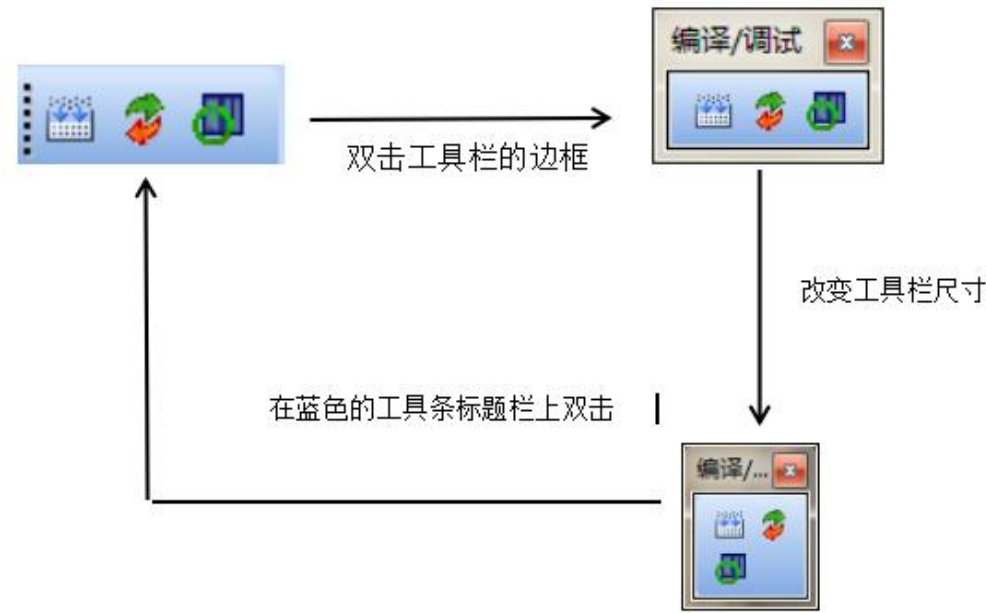


图 3-1-47

3.1.3 工作区

工作单是通过双击工程树中的相关工作单图标打开工作区。

这些工作单既可以使用图形编辑器来编辑，也可以使用文本编辑器来编辑。声明包含在变量表格工作单中。



图 3-1-48

提示：

默认情况下，工作区视图被设置为‘工作簿样式’：如果打开了多个工作单，则每个已打开工作单都被分配了一个页面标签。要激活一个特定工作单，请点击相应标签，或者通过反复按动 <CTRL>+<TAB>去浏览已经打开的工作单。

要整理多个工作单窗口，可以使用‘窗口’菜单中的多个命令，例如：‘窗口 > 层叠’或‘窗口 > 平铺’。

要最大化工作区(当使用小显示器时很有用)，您可以隐藏窗口，这既可以通过点击适当的工具栏按钮，也可以通过激活自动隐藏功能，如调整窗口主题中所述。

3.1.4 交叉参考窗口

交叉参考列表包含了当前工程内使用的所有变量、功能块、动作、转换、步、跳转、标号以及连接符。这对于调试和错误查找来说是一个很有用的工具。

每个 POU 包含了它自己的局部数据。这意味着，如果您打开某一特定工作单，交叉参考列表内的局部变量随之更新。

➤ 打开“交叉参考”

如果交叉参考窗口不可见，请点击‘交叉参考’图标 。

交叉参考窗口的列(栏目)	栏目内容的含义
--------------	---------

变量	显示变量、FB、SFC 步、转换等的相应符号和名称。如果某个过滤器是活动的，则这一点会在列标题栏中由下面的符号显示出来：  列标题还指明交叉参考列表当前是否最新。这或者由一个红手符号来表示，或者由一个绿手符号来表示。  意味着交叉参考是最新的。  表示自从上次建立了交叉参考之后，已经在某个工作单中做了修改。 对于结构和数组分别显示每个单一元素。
‘POU/ 工作单’	声明和使用了该变量/FB 的 POU 名称。
‘访问’	对该变量的访问是‘读’或者是‘写’。
‘I/O 地址’	物理地址。

可以直接访问使用或声明了某个特定变量的相应工作单。而且，如果你在一个编辑器中或者在变量工作单中标记了一个变量，该变量也会在交叉参考窗口中被标记出来。

可以按字母顺序将列表款目分类。每一列都可被用作分类准则。条目可以按升序和降序来分类。

可以根据个人的偏好来修改列的顺序。修改操作可以通过拖拽来完成：点击将被移动的列头。在将列头拖拽到目标位置的过程中，要保持鼠标键处于按下状态。释放鼠标键，以便将该列放置在此位置。

为了能够快速浏览所显示的对象，每一项都用一个代表其类型的符号来显示。下面的列表列出了‘变量’列中所显示的各个项及其相应符号：



 变量	POU/工作单	访问	I...~
 A00_IOC_D0710_02_CH01	配置.资源.Global_Variables		%QX6000.1
 NS_G	SFC_PRO.NS_GREEN.NS_GREEN	写	%QX6000.1
 NS_G	SFC_PRO.NS_GREEN_F.NS_GREEN_F	写	%QX6000.1
 A00_IOC_DI710_01_CH02	配置.资源.Global_Variables		%IX6000.2
 A00_IOC_D0710_02_CH02	配置.资源.Global_Variables		%QX6000.2
 NS_Y	SFC_PRO.NS_YELLOW.NS_YELLOW	写	%QX6000.2
 A00_IOC_DI710_01_CH03	配置.资源.Global_Variables		%IX6000.3

图 3-1-49

-   局部变量
-   局部变量的声明

- ✧  全局变量
- ✧  使用了‘VAR_GLOBAL’的全局变量声明
- ✧  输入型变量 (VAR_IN)
- ✧  输出型变量 (VAR_OUT)
- ✧  输入/输出型变量 (VAR_IN_OUT)
- ✧  功能块
- ✧  未引用过的变量
- ✧  跳转
- ✧ **LBL** 标号
- ✧  连接符

➤ 如何建立交叉参考

- ✧ 右键点击交叉参考窗口，并选择上下文菜单项‘建立交叉参考’。就自动地创建了交叉参考列表。
- ✧ 或者通过按<F12>，创建了交叉参考列表。
- ✧ 或者在‘建立’菜单或编辑器的上下文菜单中，选择‘建立交叉参考’。

➤ 使用交叉参考

利用交叉参考列表，既可以访问使用了某个变量的代码工作单，也可以访问声明该变量的相应变量工作单。

可以按字母顺序来分类诸列表条目。每一列都可被用作分类准则。条目可以按升序和降序来分类。

有几种可能来访问代码工作单并标记变量：

- ✧ 在交叉参考窗口中双击所需的行。
- ✧ 在交叉参考窗口中右键点击所需的行，并选择上下文菜单项‘转到位置’。

3.1.5 定制用户界面

要定制用户界面，请在‘附加’子菜单中选择‘选项’菜单项。出现‘选项’对话框，它被分成多个页面。

对话框页面...	目的
‘常规’	使用该列表框，您可以设置用户界面启动时的默认语言。 标记"隐藏外部变量"复选框意味着：使用 IEC 变量声明关键字 VAR_EXTERNAL 声明的外部变量是不可见的。 通过按‘复位工作空间’按钮，可以将用户界面中的打开的窗口、可见菜单和工具栏的布置复位到交付状态。 在组合框"题目"中，您可以选择一个预定义的用户界面外观。
工具栏	“大按钮”：通过复选/不复选“大按钮”复选框，在大按钮和小按钮间切换。
交叉参考	用于定义交叉参考窗口中的哪些列可见、哪些列将随着工程打印出来。 网格线： 如果该复选框已被标记，则交叉参考窗口中会显示出网格线。 在打印过程中，使能整词自动换行： 如果已经激活，则使能了交叉参考窗口内容的多行打印： 区域中的内容将被折成多行。
页面布局	用于定义不同的默认页面布局。
图形编辑器	用于定义图形工作单的默认尺寸、图形编辑器中的网络布局的默认设置，以及规定是否应该在插入 FU/FB 时，其内部要有一个图片。
逻辑分析器颜色	这个标签用于指定逻辑分析器窗口中所显示的不同曲线的颜色。 因为逻辑分析器可以同时显示多达 10 个变量的曲线，所以，对于所定制颜色的使用，会对所测量结果的阅读提供便利。
变量表格	用于定义变量表格工作单中哪些列可见、哪些列将被打印。要显示/打印某一特定的列，请用鼠标左键点击相关的复选框来标记之。 在打印过程中，使能整词自动换行： 如果激活了这个选项，则会使能变量表格工作单的单元格内部的多行打印： 必要时，单元内容将被自动截断，并在下一行中写出。这样，一个单独的单元可以展开到整个页面。

3.2 工程树——硬件

打开已经建立的工程，在左侧的工程树中点击硬件，出现硬件目录树。展开硬件目录树，可以显示工程的模型结构。用户可以分层自的设置、查看，如图 3-2-1。



图 3-2-1

3.2.1 物理硬件

‘物理硬件’树包含了配置元素，反映了符合 IEC 61131-3 的配置元素的结构。‘物理硬件’可以插入‘配置’，并且只能有一个配置存在。

3.2.2 配置

‘配置’是‘物理硬件’下一层，负责管理他的下一层‘资源’，目前 MULTIPROG Express 版本仅支持一个‘资源’。可以删除‘资源’，也可以从其他工程拷贝一个‘资源’。

在‘配置’上点击右键，选择‘属性’，可以查看‘配置’的信息。此选择不可以修改。

3.2.3 资源

‘资源’是硬件树中‘配置’的下一层。负责管理 Truhigh P600 系统的控制器，包括处理器类型，处理器的通讯地址等信息。

➤ 选择控制器类型

在‘资源’上点击右键，选择“属性”，选择‘PLC/处理器’选项卡，可以查看修改处理器类型，如图 3-2-2。

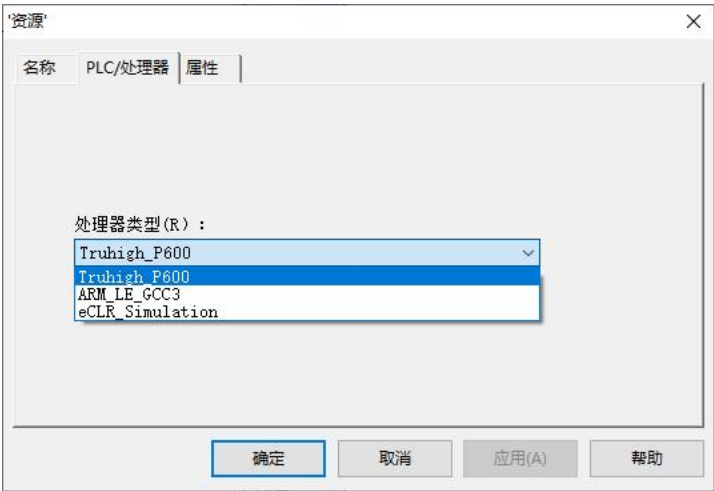


图 3-2-2

 提示:

处理器类型有 Truhigh_P600、eCLR_Simulation 和 ARM_GCC3 三种类型。
Truhigh_P600: 适用于 Truhigh P600 系列 CPU
ARM_GCC3: 适用于 ARM 平台控制器
eCLR_Simulation: 仿真控制器。

➤ **设置 IP 地址**

在资源目录上点击右键，选择设置，修改通讯参数为 Truhigh P600 PLC 的 IP 地址，如：192.168.0.100，如图 3-2-3。这里设置的 IP 地址要和 Truhigh P600 PLC 的 IP 地址一致，否则无法下载用户程序。修改 IP 地址见 1.4 章节。

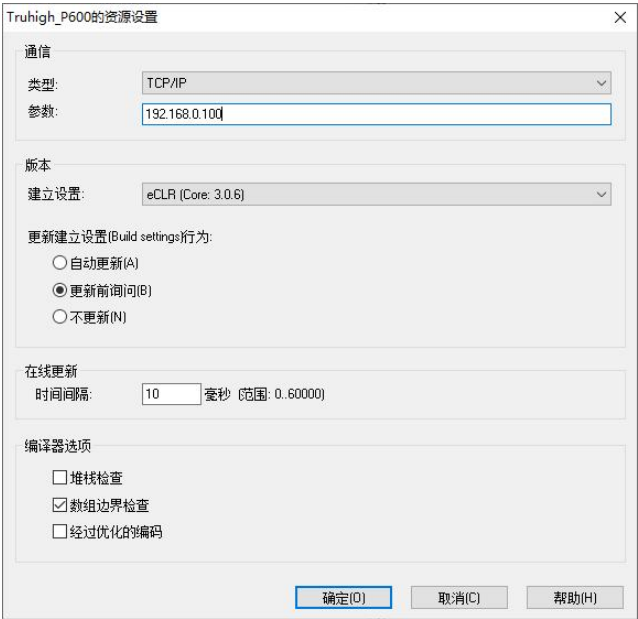


图 3-2-3

如果选择 eCLR_Simulation 控制器进行仿真，在通讯参数中选择：‘仿真 1’，版本-建立设置：eCLR_Simulation (Core: 3.0.0)。点击确定，如图 3-2-4。

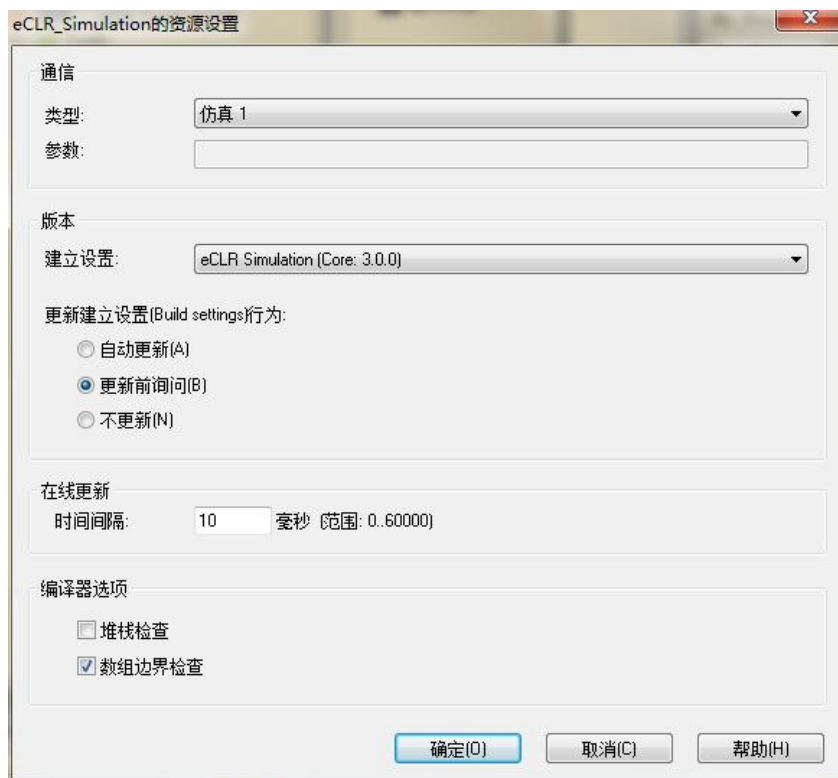


图 3-2-4

3.2.4 Tasks

‘Tasks’是‘资源’的下一层，‘Tasks’下可以插入多个任务（最多 16 个），可以是循环扫描任务，也可以是周期扫描任务。

创建一个工程后，MULTIPROG 自动生成了一个‘Tasks’和这个目录下的一个任务。用户可以插入一个新的任务。

任务类型分为：DEFAULT 和 CYCLIC，即默认任务和循环任务。



提示：

- ✧ 默认任务：默认任务执行完成之后，系统等待一个所定义的空闲时间，如果接下来还是没有具有更高优先级的事件要处理，则该默认任务被自动再次执行。
- ✧ 循环任务：以固定的时间间隔执行它所关联的程序。

当使用一个循环任务时，必须要定义执行任务的时间间隔以及所关联的程序。固定的程序调用时间间隔很重要，例如当程序是控制循环中的一部分时。在这里，时间间隔会影响控制器特性。

如果设置了 CYCLIC 循环任务，则需要设置时间间隔、优先权和监视定时时间，右击‘任务’，选择设置，如图 3-2-5。



图 3-2-5

- ✧ 间隔：以毫秒为单位，指定执行循环扫描型任务要用的时间间隔。时间间隔值也可以作为一个浮点值插入。这允许定义 1 毫秒以下的时间间隔。该时间值应该是目标设备的定时器分辨率的倍数。仅可用于循环扫描型任务。
- ✧ 优先级：指定调用任务的优先权。0 代表最高优先权，而 31 代表最低优先权。具有最高优先级的任务首先被调用。
- ✧ 监视定时器时间：指定一个任务执行一个周期（或被调用一次）所允许的时间。如果超出此监视定时器时间，则会产生一个监视定时器错误。通过这种方法，监视定时器监控着任务的执行以及任务在指定时间内的完成情况。监视定时器(watchdog)在每个系统周期内都执行。

对于循环扫描型任务，其监视定时器时间通常应该比系统周期时间短。监视定时器时间的有效范围：1 - 5000 ms。

激活监视定时器 选中该选项，激活任务监视定时器，并在“任务监视定时器时间”输入时间设置。如果未选中，任务监视定时器不会对当前的任务进行监视。

插入新的任务后，右击新插入的任务名，选择‘程序实例’，指定程序实例名和实例类型。实例类型是在 POU 中插入的程序（PROGRAM）。

一个工程中，只允许有一个 DEFAULT 类型的任务，可以有多个 CYCLIC 任务。在一个任务中，可以插入多个程序实例，多个程序实例按照顺序执行。

3.2.5 Global_Variables——全局变量

‘Global_Variables’在‘资源’下一层，与‘Tasks’并列，‘Global_Variables’是一个全局变量表。

全局变量包括：IO 变量表、系统变量、用户自定义变量。

✧ IO 变量表是添加硬件之后自动添加 IO 的变量。

✧ 系统变量是 MULTIPROG 软件提供的变量。

✧ 用户自定义变量是编写用户程序是定义的全局变量。

双击 ‘Global_Variables’，可以打开全局变量表，查看、修改、添加全局变量。

➤ 如何插入变量

1. 添加变量组：先点击变量组，如 ‘System Variables’ 组，然后点击右键，选择 ‘插入组’。双击名称，可以修改变量组名。如修改为 “AI”。
2. 添加变量：点击 “AI” 组，然后邮件选择 ‘插入变量’，如图 3-2-6。添加了变量名为 NewVar1 的变量。

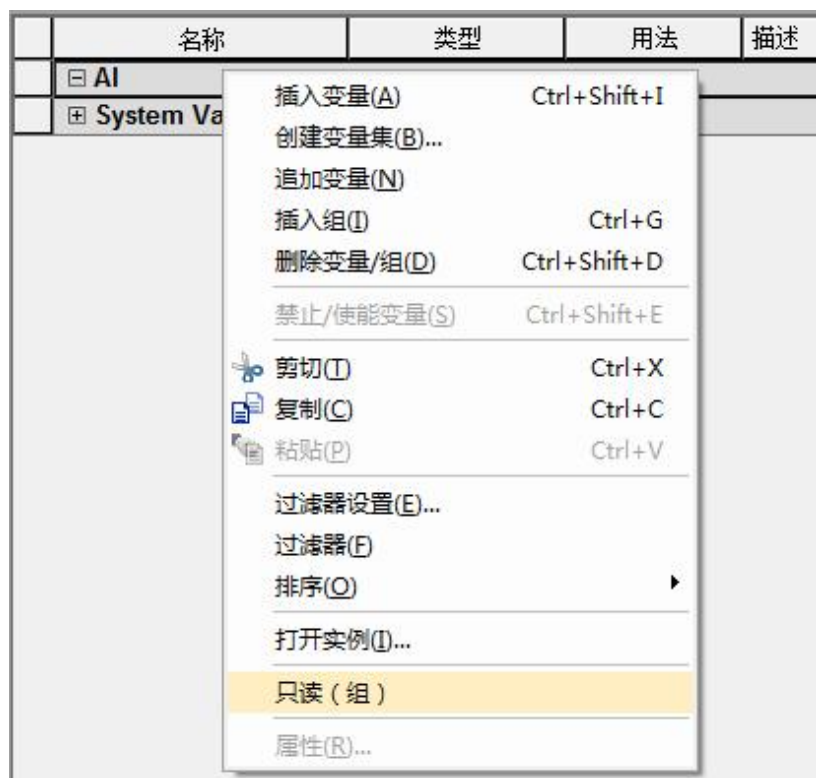


图 3-2-6

3. 在新建的变量上点击右键，然后选择 ‘属性’。可以查看修改变量属性，如图 3-2-7。



图 3-2-7

对话框元素	含义
名称	显示变量/实例的名字。
数据类型	在指定变量的数据类型。数据类型决定变量可以具有哪种数值，并且它们定义初值、可能值的范围以及所占的二进制位数。
用法	全局变量为：VAR_GLOBAL。
RETAIN	表示(将要)声明一个保持型变量。保持型变量是那些即使在关掉电源的情况下仍保存值的变量。在进行一个暖启动的情况下，将使用变量的最后值。
初值 / 默认的隐藏值	指定初值。 在 PLC 程序中，第一次使用某个变量时，会使用其初值。初值必须符合所选择的数据类型。
I/O 地址	指定变量地址，也可以不指定地址。
描述	用户定义的注释。
PDD	指明变量已写入过程数据目录（PDD）中。 只有当您想访问 PLC 上的某个地址对应的变量名称时，才需要用到它。
OPC	指明变量已写入*.csv 文件中。 只有当您想通过使用诸如 OPC 服务器等来访问变量，以便运行某个可视化客户机程序时，才需要用到它。
隐藏	仅仅可用于功能或功能块 POU 的输入和输出变量(使用

	VAR_INPUT 和 VAR_OUTPUT 关键字声明的)。 如果标记了这个复选框,则在另一个图形工作单中调用(插入)这个用户 POU 时,特定的输入/输出变量将不能作为块的形式参数出现。 对于隐藏的形式参数,可以指定一个默认的隐藏值(见表的下一个行)。
初值作为默认值	仅仅可用于隐藏的形式参数(VAR_INPUT/VAR_OUTPUT 声明),如上面的表行中所述。 如果这个复选框被标记,‘初值’域中所输入的值被用作‘默认隐藏值’(即使该输入域相应地被重新命名 - 见上面的‘初值’域的描述)。

➤ 创建变量集

如果要新建的变量比较多,而且变量安装数字顺序排列,如 M_STA01、M_STA02、M_STA03、M_STA04、M_STA05 等 5 个变量时,可以使用创建变量集,如图 3-2-8。

图 3-2-8

对话框元素	含义
名称	指定了在变量表格中如何命名新变量。必须将新变量名称前缀输入到'名称'输入区域中,且该前缀后跟随字符'#'(例如, 'NewVar#')。输入字符'#',使能列表框'起始'和'结束'。 范例: 名称前缀'NewVar#'以及起始号码'0'和结束号码'3'创建了四个变量,名为'NewVar0'、'NewVar1'、...、'NewVar3'。 如果您没有输入字符'#',那么将仅有一个带有特定名称的变量被插入到变量表格中。 ‘预览’框显示了在变量表格中如何命名变量。
起始/结束	指定要被插入到变量表格中的变量号码,以及如何给变量编号。变量名称由名称前缀和一个号码组成,该号码范围

	从'起始'号码至'结束'号码。 范例： 名称前缀'NewVar#'以及起始号码'0'和结束号码'3'创建了四个变量， 名为'NewVar0'、 'NewVar1'、 ... 、 'NewVar3'。 只有在名称前缀含有'#'字符的情况下， 这个区域才能访问。
如 果 需 要， 填 充 前 导 '0'	如果您要同时插入多于 10 个变量， 那么使用前导零填补变量名称(例如， 用'NewVar01'、 'NewVar02'代替'NewVar1'、 'NewVar2')。 只有在名称前缀含有'#'字符的情况下， 这个复选框才能访问。
I/O 地址	指定要被分配给新变量的地址前缀。地址前缀必须以下面的格式输入： '%位置前缀尺寸前缀地址' (例如， '%MX1')。通过将字符'#'附加到地址前缀， 起始号码和增量能够被相应的列表框另外指定。 如果您没有输入字符'#'， 那么所有新变量将拥有相同 I/O 地址。 '预览'框显示了哪个 I/O 地址将被指派到新变量。如果您输入了一个错误的地质前缀， 那么将显示<无效地址>。
起 始 / 增 量	指定地址值的起始号码和增量。 范例： 地址前缀'%MX1#'， 起始号码'1'和增量'2'会产生 I/O 地址'%MX11'、 '%MX13'、 '%MX15'、 '%MX17'.... ‘起始’只有在下列情况下才被使能： 如果地址前缀包含字符'#'， 并且复选框‘使用名称的定义范围’没有被标记。只有地址前缀和名称前缀都包含字符'#'的情况下， ‘增量’才被使能。
使 用 名 称 的 起 始 值	如果想要使用'名称'组框中指定的起始号码作为 I/O 地址的起始号码， 请标记该复选框。在这种情形下， 'I/O 地址'组框中的起始号码为不活动状态。
用法	指定变量关键字(根据 IEC 61131-3 标准)。 取决于(将被)声明的变量所在的 POU， 列表框中包含了各种变量声明关键字。 在程序类 POU 中， 只能选择 VAR, VAR_EXTERNAL 和 VAR_EXTERNAL_PG。 在功能块类 POU 中， 该列表框包含了 VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT 和 VAR_EXTERNAL 等关键字。 在功能类 POU 中， 只能选择 VAR 和 VAR_INPUT 关键字。 如果在对话框‘选项 常规’内， 标记了复选框‘隐藏外部变量’， 则无法在这里选择关键字 VAR_EXTERNAL 。而是可以使用 VAR_GLOBAL。见这个表格上方的注解‘隐藏的外部变量’。
RETAIN	指明(将要)声明的保持型变量。保持型变量是那些即使在

	关掉电源的情况下仍保存值的变量。在进行一个暖启动的情况下，将使用变量的最后值。 RETAIN 关键字可以与 VAR、VAR_OUTPUT、和 VAR_GLOBAL 等关键字组合使用。不能用 VAR_INPUT 和 VAR_EXTERNAL 等关键字来声明保持型变量。 当声明保持型本地变量或直接表示的变量(%Q或%M)时，您应该注意 I/O 配置中的保持设置！ 在一个保持型输出变量(声明时带有位置前缀%Q...)的情况下，您必须确保在 I/O 配置中对相应 OUTPUT 组的‘保持’复选框也要进行设置。这就需要将输出组定义为保持型区域。 在(声明带有位置前缀%M...)保持型标志变量的情况下，您必须在相关的 VARCONF 声明中相应地选择‘保持’复选框。 只有在两个‘保持’复选框(在变量声明和 I/O 配置中)都被选择的情况下，该变量才会在断电情况下也会保持，并且在热启动期间也不会被初始化。 只有在两个‘保持’复选框(在变量声明和 I/O 配置中)都被选择的情况下，该变量才会在断电情况下也会保持，并且在热启动期间也不会被初始化。 请注意观察那些关于途径的信息，编程系统在下装或修补工程并执行一个 PLC 暖启动时，将处理变化了的或者新插入的保持型变量。
数据类型	指定变量的数据类型。数据类型决定变量可以具有哪种数值，并且它们定义初值、可能值的范围以及所占的二进制位数。
初值	指定初值。在 PLC 程序中，第一次使用某个变量时，会使用其初始值。 初值必须符合所选择的数据类型。
描述	用户定义的注释。将字符'#'输入到'描述'区域中，另外，每个新变量的注释将包含新变量的当前号码，该号码在'名称'组框中指定。
PDD	指明变量已写入过程数据目录（PDD）中。只有当您想访问 PLC 上的某个地址对应的变量名称时，才需要用到它。
OPC	指明变量已写入*.csv 文件中。只有当您想通过使用诸如 OPC 服务器等来访问变量，以便运行某个可视化客户机程序时，才需要用到它。

3.2.6 IO-Configuration——IO 配置

‘IO-Configuration’ 在 ‘资源的下一层’，与 ‘Tasks’、‘Global_Variables’ 并列。

I/O 配置一般包含了对 I/O 模块的声明，例如：模块的逻辑地址(开始和结束地址)，对设备的声明(驱动程序名称或存储器地址)，等等。当所连接的过程运行时，PLC 通过 I/O 模块接收输入信号。依照 PLC 程序处理这些输入信号。在一个工作周期之后，PLC 将输出信号传送到 I/O 模块。PLC 处理这些信号的方式，是通过 I/O 配置设置的，如图 3-2-9。

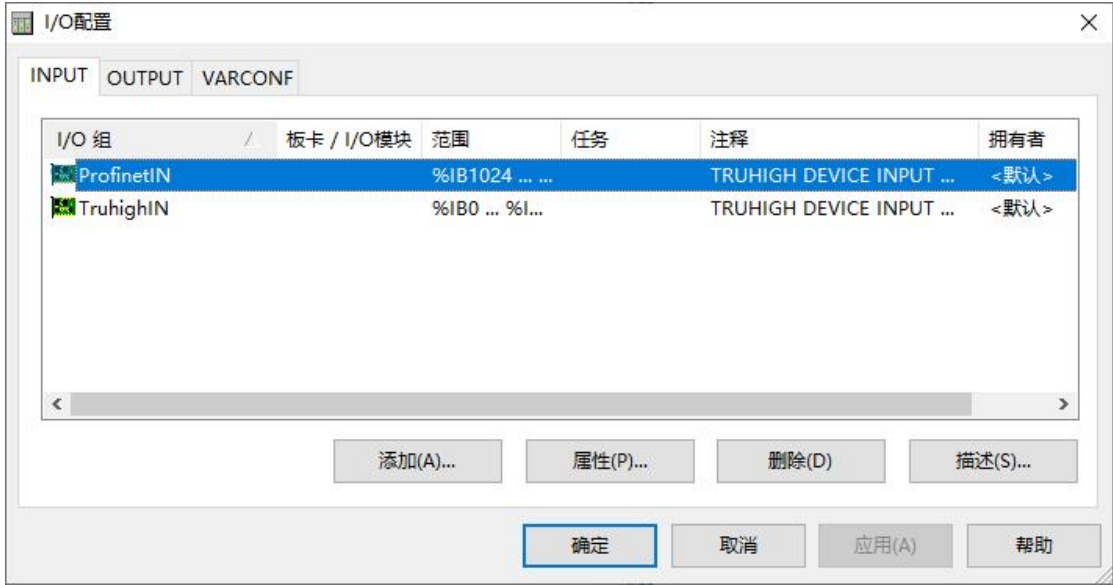


图 3-2-9

3.3 工程树——工程

在工程树中选择工程选项卡，可以查看工程的程序结构。工程包含库、数据类型、逻辑 POU 三部分，如图 3-3-1。

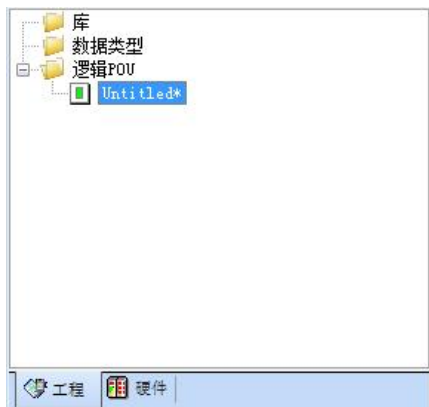


图 3-3-1

在‘库’子树中，可以将库包括到工程中。

在‘数据类型’文件夹中，可以添加一个工作单，以便声明用户自定义数据类型。

在‘逻辑 POU’子树中，保存了 POU，并且可以添加新的 POU。

3.3.1 库

库提供了功能块、功能、程序和数据类型。在包括一个库之后，您可以使用当前正在编辑的工程中被包含的元素。

3.3.1.1 库类型

固件库是那些包含了由 PLC 制造商所提供的 POU 的库。固件库的文件扩展名为*.fwl。

Truhigh P600 系统提供了 2 个固件库：BIT_UTIL.FWL 和 PROCONOS.FWL，库文件说明详见《Truhigh P600 功能块手册》。

用户库是那些您以前已创建的、并想重新使用其中的 POU 的工程。用户库的文件扩展名为*.mwt。

Truhigh P600 系统提供了 3 个用户库：TruHigh_Analog.mwt，TruHigh_Logic.mwt 和 TruHigh_Motor.mwt，库中包含的功能块、功能说明见《Truhigh P600 功能块手册》。

3.3.1.2 库表示法

在工程树的‘库’文件夹中，固件库由一个带有钥匙符号的书形图标来表示。这些图标不能被扩展。所包含的功能和功能块可以从编辑向导中的相应库‘组’插入。

用户库由一个(不带钥匙符号的)书形图标来表示。它们可以被扩展(通过双击该图标或者左键点击符号‘+’)以显示所包含的 POU。可以查看所包含的 POU，但是不能改变它们。要想在工程中使用这些用户库 FU/FB，请从编辑向导插入它们(类似于固件块)。

3.3.1.3 插入固件库：

1. 在“库”上点击右键，选择插入-固件库。

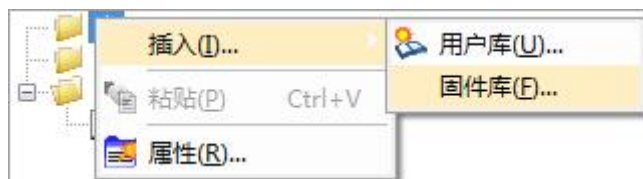


图 3-3-2

2. 在打开的“包含库”对话框中选择要添加的库文件，然后点击包括。在编辑向导中就可以看到添加的库。

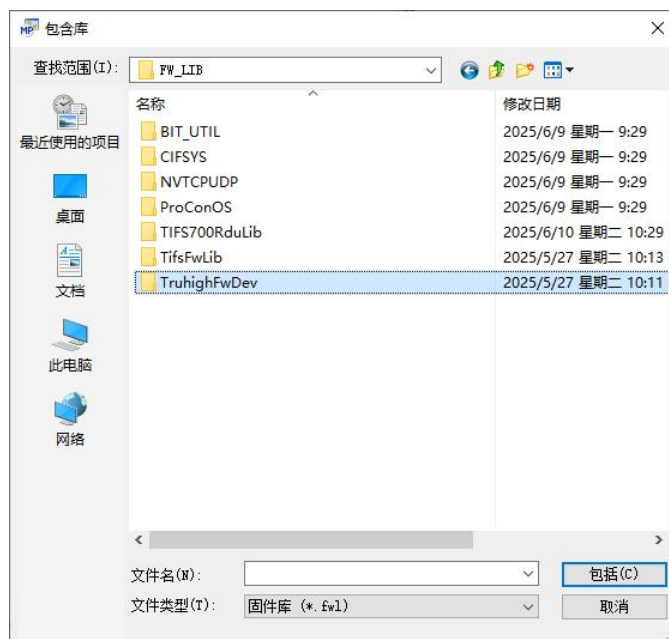


图 3-3-2

3.3.1.4 插入用户库：

在“库”上点击右键，选择插入-用户库。

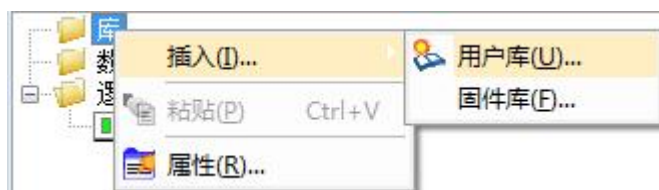


图 3-3-3

在打开的“包含库”对话框中选择要添加的库文件，然后点击包括。

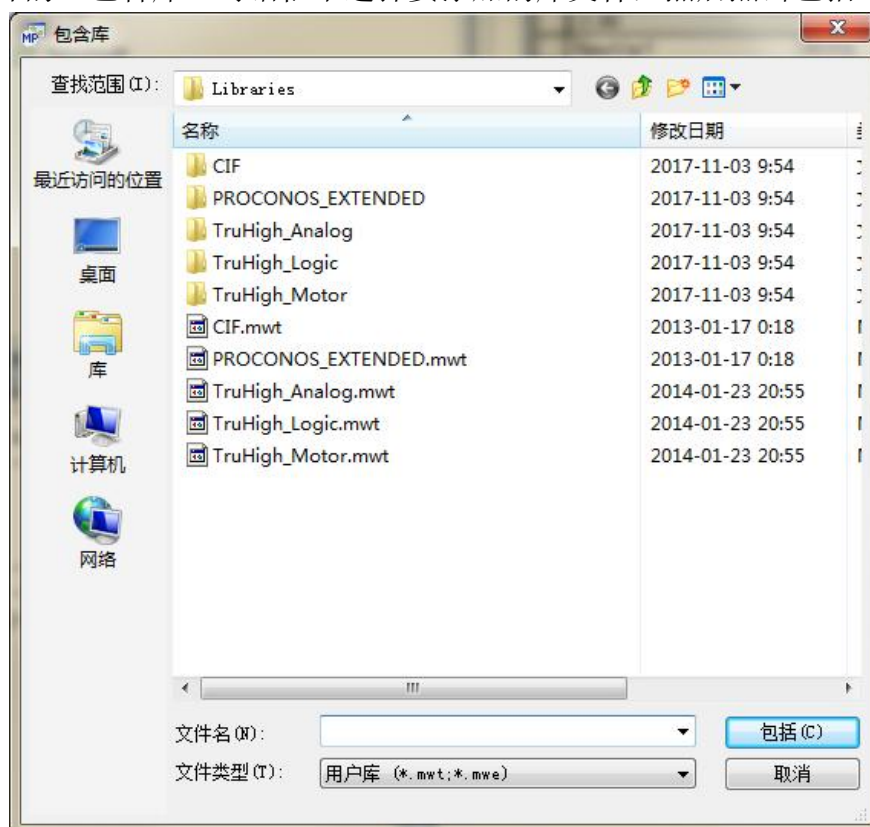


图 3-3-4

在编辑向导中就可以看到添加的库。



图 3-3-5

3.3.2 数据类型

如果用户要使用用户自定义数据类型(如: 数组、结构, 等等), 则这些数据类型必须要在某个数据类型工作单中声明。这些工作单位于工程树的‘数据类型’文件夹中。

系统允许每个工程有一个数据类型工作单。这样, 如果已经存在一个数据类型工作单, 则用于添加数据类型的所有命令和工具栏图标都处于不活动状态。

插入一个自定义数据类型: 鼠标右键点击‘数据类型’文件夹图标, 并选择上下文菜单项‘插入 > 数据类型’。在出现的对话框内输入一个工作单名称, 例如 ARR1, 并点击‘确定’, 以将这个工作单插入到工程树中。

编辑自定义数据类型: 双击打开新建的工作单 ARR1, 键入一下字符:

```
TYPE
DATA1:ARRAY [1..100] OF INT;
END_TYPE
```

以上代码定义了一个包含 100 个类型是 INT 的数组, 数组名称是 DARA1

3.3.3 逻辑 POU

根据 IEC 61131-3, 程序组织单元或 POU 是 PLC 程序的语言元素。它们是包含了程序代码的小的、独立的软件单位。POU 的名称在工程内必须是唯一的。

三种不同的 POU 类型可用: 程序、功能块、功能。

每种 POU 都由两个不同部分组成：变量声明部分和代码本体部分。两者都被命名为'工作单'。在声明部分，声明了所有局部变量。一个 POU 的指令或代码部分包含指令，该指令是用所希望的编程语言编写的。

3.3.3.1 功能

功能是带有多个输入参数和恰好一个输出参数的 POU。调用带有相同值的功能总是返回相同的结果。返回值可以是简单数据类型。在一个功能内，可以调用另外的功能，但不能调用功能块或程序。不允许递归调用。功能的缩写为 FU。

IEC 61131-3 表述了标准功能。依赖于所使用的硬件和 PLC 类型不可能使用所有的标准功能，也不可能提供所有的固件功能。

下面是一个标准 IEC 61131-3 功能列表：

- 类型转换功能，如 INT_TO_REAL
- 数值功能，如：ABS 和 LOG
- 标准算术运算功能，如：ADD 和 MUL
- 位串功能，如：AND 和 SHL
- 选择和比较功能，如：SEL 和 GE
- 字符串功能，如：RIGHT 和 INSERT
- 时间数据类型功能，如带有 TIME 数据类型的 SUB

3.3.3.2 功能块

功能块是带有多个输入/输出参数和内部存储单元的 POU。功能块的返回值取决于其内部存储单元的值。功能块内可以调用另外的功能块或功能，但不能调用程序。不允许递归调用。功能块的缩写为 FB。

IEC 61131-3 表述了标准功能块。依赖于所使用的硬件和 PLC 类型不可能使用所有的标准功能块，也不可能提供所有的固件功能块。所有的 FB(IEC 定义的，库件 FB 和用户定义的 FB POU)可以很容易地被插入到一个正在使用编辑向导调用的 POU 的代码体中。

每个被调用的 FB 必须在调用 POU 的局部变量工作单中被声明。将调用其它 POU 中的一个 FB 命名为实例化，具体功能块使用可参考《Truhigh P600 功能块手册》。

下面是一个标准 IEC 61131-3 功能块列表：

- 双稳态元素，如 SR 和 RS
- 边沿检测功能块，如：R_TRIG 和 F_TRIG

- 计数器，如：CTU 和 CTD
- 定时器功能块，如：TON 和 TOF

3.3.3.3 程序

程序 POU 通常包含了功能/功能块调用的一个逻辑组合。程序的行为和用途类似于功能块。程序具有输入和输出参数，而且可以具有内部存储区。不允许递归调用。

在创建一个工程时，MULTIPROG 自动声明了一个程序，用户声明一个新的程序时，MULTIPROG 同事生成这个程序的变量工作单，并把它加载到‘Tasks’目录下的第一个任务中。

程序一定要被关联到任务上。下图显示了一个带有两个程序的默认任务：

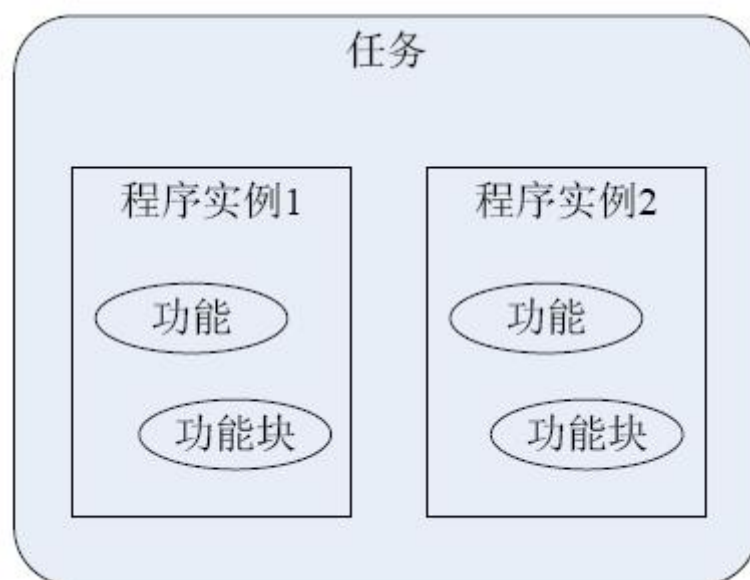


图 3-3-6

第 4 章 硬件组态

硬件组态用于在工程师站上定义应用系统的硬件配置。
Truhigh P600 系列 PLC 支持本地 IO 模块、Modbus、Profinet 等设备。

本章分为下面几个章：

- 4.1 CPU 编辑
- 4.2 本地 IO 组态
- 4.3 作为标准 Modbus 从站组态
- 4.4 Profinet 远程站组态

■ 系统概述

- Truhigh P600 PU620 控制器可以添加本地 IO、Profinet 远程 IO、作为 Modbus 从站等。

4.1 CPU 编辑

1. 新建一个工程，选择资源类型为 Truhigh_P600 的工程。选择菜单栏-视图-创恒设备组态浏览器，进入硬件组态界面，右键点击设备资源，选择网卡-本地连接，选择本机可用的网卡。



图 4-1-1

2. 然后再右键点击设备资源，选择添加设备->PU620，如图 4-1-2。

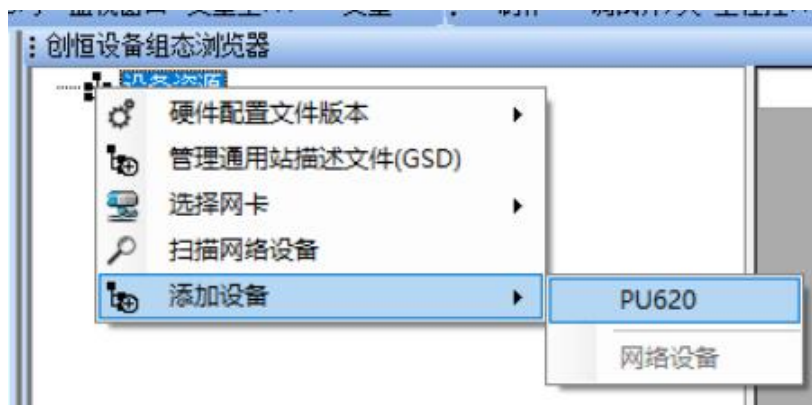


图 4-1-2

3. CPU 出现在了硬件资源目录下，如图 4-1-3。



图 4-1-3

4. 右键点击 PU620 在弹出的菜单中选中“编辑”。



图 4-1-4

5. 进入 CPU 编辑界面，如图 4-1-5：

图 4-1-5

- 名称：可修改 PLC 的显示名称；
- IP 地址：填写待连接 PLC 的 IP 地址；
- 子网掩码：填写待连接 PLC 的子网掩码；
- 网关：填写待连接 PLC 的网关；
- 启用 OPCUA：选择是否启动 OPCUA 服务，固定端口号 48400

4.2 本地 IO 组态

1. 右键点击“本地 IO”，在弹出来的界面中选择“添加”，如图 4-2-1：



图 4-2-1

2. 然后在弹出的“添加 IO”界面添加所需模块，并可根据需求更改模块位置，如图 4-2-2。



图 4-2-2

3. 根据现场实际需要，添加 IO 模块，如图 4-2-3。

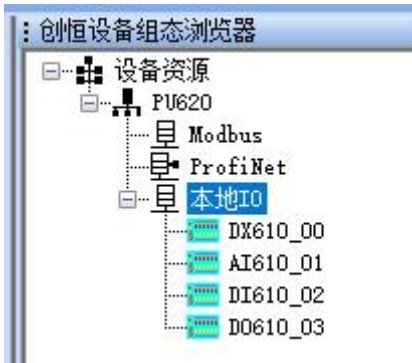


图 4-2-3

4. 右键添加之后的模块，选择编辑，弹出 IO 模块配置信息，可修改通道类型和模块位置，如图 4-2-4。



图 4-2-4

5. 在工程树中点击“硬件”，双击“Global_Variables”就可以打开全局变量表，添加的模块通道变量自动添加到了全局变量表中，如图 4-2-5。



图 4-2-5

4.3 作为标准 MODBUS 从站组态 MBS

Truhigh P600 PLC 可以配置为标准 Modbus 从站与标准 Modbus 主站通讯。

- 1) 右键选择 MBS->启用：启用作为 Modbus 从站功能。

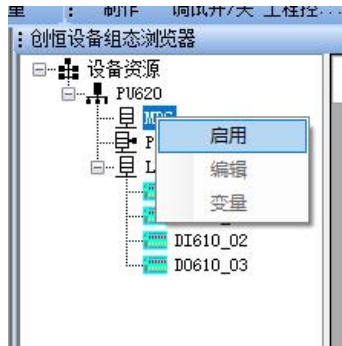


图 4-3-1

- 2) 右键选择 MBS->编辑：修改从站通讯参数。

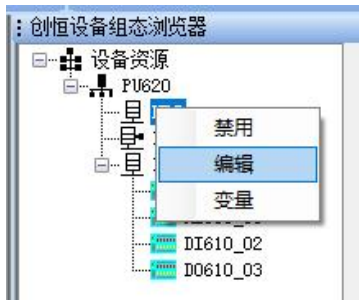


图 4-3-2

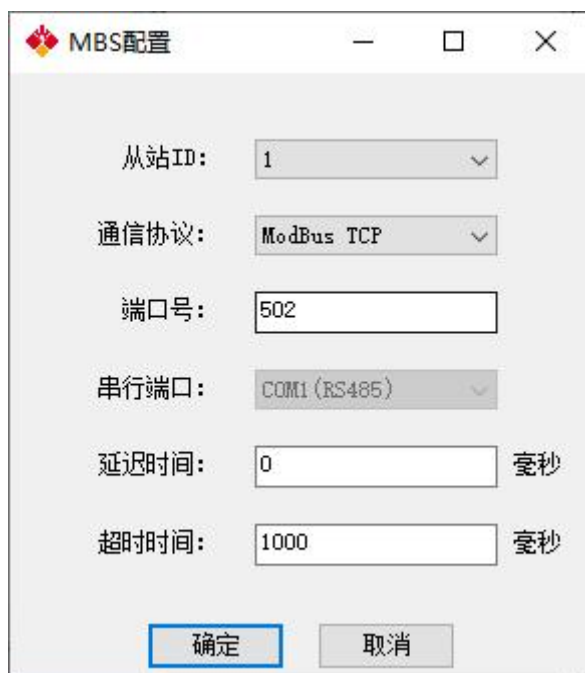


图 4-3-3

从站 ID:PU620 作为从站时的 Modbus 地址 1-255;

延迟时间: 响应主站延时返回时间, 单位毫秒;

超时时间: 判断主站连接超时时间, 单位毫秒;

通信协议: Modbus TCP/RTU;

作为 Modbus TCP 从站时参数:

端口号: 网络端口号;

作为 Modbus RTU 从站时参数:

串行端口: 可选择那个串口作为外部通讯。

3) 右键选择“变量”, 弹出变量添加界面:

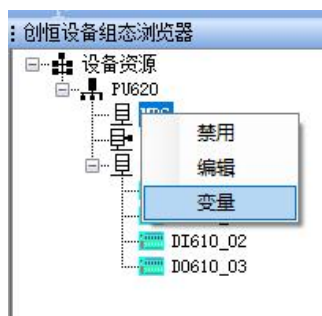


图 4-3-4

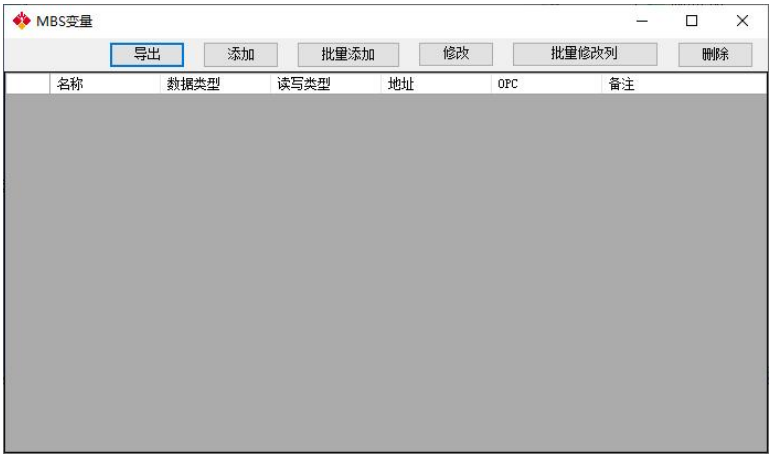


图 4-3-5

导出：可把当前配置的变量点表导出到 csv 格式表格中，以便其他应用使用或者查看点表信息。

添加：添加单个变量；

批量添加：添加多个有规律名称的变量；

修改：修改所选中的变量属性；

批量修改列：拖选变量后点击批量修改列；

删除：删除所选中的变量；

4) 导出界面如下：

可把当前配置的变量点表导出到 csv 格式表格中，以便其他应用使用或者查看点表信息。

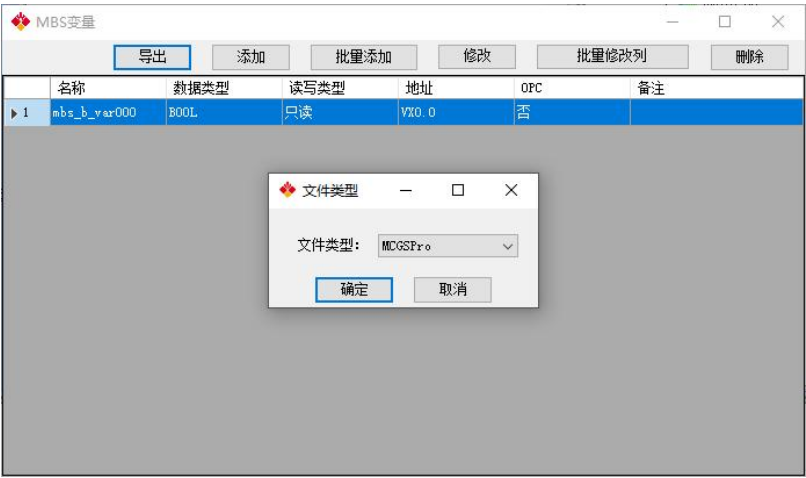


图 4-3-6

5) 单个变量添加和修改界面如下：



图 4-3-7

名称：变量在全局变量表内显示名称；
数据类型：变量数据类型；
读写类型：主站对 PU620 的操作类型，读写/只读；
地址偏移：从站变量的 V 区地址；
 BOOL 数据地址范围为 0-125，位偏移为 0-8；
 非 BOOL 数据地址范围为 0-4000，位偏移为 0；
位偏移：针对 BOOL 类型数据在地址内的位偏移（0-8）；
OPC：是否作为 OPC 变量；
例如：添加以下变量时地址偏移和位偏移分别为
AA:地址偏移为 0，位偏移为 0，对应 Modbus 地址为线圈区 48000；
BB:地址偏移为 0，位偏移为 1，对应 Modbus 地址为线圈区 48001；
CC:地址偏移为 0，位偏移为 2，对应 Modbus 地址为线圈区 48002；
DD:地址偏移为 0，位偏移为 0，对应 Modbus 地址为保持区 6000；
EE:地址偏移为 2，位偏移为 0，对应 Modbus 地址为保持区 6001；
FF:地址偏移为 6，位偏移为 0，对应 Modbus 地址为保持区 6003；

MBS变量						
刷新全局变量 添加 批量添加 修改 删除						
	名称	数据类型	读写类型	地址	OPC	备注
▶ 1	AA	BOOL	读写	VX0.0	否	
2	BB	BOOL	读写	VX0.1	否	
3	CC	BOOL	读写	VX0.2	否	
4	DD	INT	读写	VW0	否	
5	EE	REAL	读写	VDP2	否	
6	FF	INT	读写	VW6	否	

图 4-3-8

V 区变量地址与 Modbus 地址对应关系如下：

x 为地址偏移，y 为位偏移。BOOL 类型数据为线圈寄存器，其他为保持寄存器。

数据类型		Modbus 地址	通信功能码	系统显示地址
线圈(布尔数据)	BOOL	$x*8+y+48000$	读 1，写 5/15	VXx.y
16 位 无符号二进制	UINT	$x/2+6000$	读 3，写 6/16	VWUx
16 位 有符号二进制	INT	$x/2+6000$	读 3，写 6/16	VWx
16 位 4 位 BCD	WORD	$x/2+6000$	读 3，写 6/16	VWDx
32 位 无符号二进制	UDINT	$x/2+6000$	读 3，写 6/16	VDUx
32 位 有符号二进制	DINT	$x/2+6000$	读 3，写 6/16	VDx
32 位 8 位 BCD	DWORD	$x/2+6000$	读 3，写 6/16	VDDx
32 位 浮点数	REAL	$x/2+6000$	读 3，写 6/16	VDFx
ASCII 字符串	STRING	$x/2+6000$	读 3，写 6/16	VBSx

说明：ASCII 字符串变量固定为 80 个字节长度。

- 6) 批量添加 V 区变量
批量添加主界面如下：

MBS变量

	名称	数据类型	读写类型	地址	位地址	OPC	备注
*1		BOOL	只读		0	☐	

删除

导入

批量生成

确定

取消

图 4-3-9

批量添加分为批量生成和导入两种方法：
批量生成：单机批量生成按钮，弹出批量生成规则界面，如下：



图 4-3-10

名称：变量名称，使用#来表示号码插入位置；
起始/终止：号码的起始和结束（包含结束号码）；
必要时填充前导“0”：是否在号码前填充“0”；
数据类型：所批量添加的变量数据类型；
起始：变量起始地址；
读写类型：只读/读写；
OPC：所添加变量是否具有 OPC 属性。
 选择确定，按照上述配置添加的变量如下：

	名称	数据类型	读写类型	地址	位地址	OPC	备注
1	MASTER_V_000	INT	读写	0	0	☐	
2	MASTER_V_001	INT	读写	2	0	☐	
3	MASTER_V_002	INT	读写	4	0	☐	
4	MASTER_V_003	INT	读写	6	0	☐	
5	MASTER_V_004	INT	读写	8	0	☐	
6	MASTER_V_005	INT	读写	10	0	☐	
7	MASTER_V_006	INT	读写	12	0	☐	
8	MASTER_V_007	INT	读写	14	0	☐	
9	MASTER_V_008	INT	读写	16	0	☐	
10	MASTER_V_009	INT	读写	18	0	☐	
11	MASTER_V_010	INT	读写	20	0	☐	
12	MASTER_V_011	INT	读写	22	0	☐	
13	MASTER_V_012	INT	读写	24	0	☐	
14	MASTER_V_013	INT	读写	26	0	☐	
15	MASTER_V_014	INT	读写	28	0	☐	
16	MASTER_V_015	INT	读写	30	0	☐	
17	MASTER_V_016	INT	读写	32	0	☐	

图 4-3-11

在批量变量列表里面可以单独查看、修改某个变量，修改完成后点击确定则再全局变量表里面生成相应的变量，同时显示在从站 V 区变量表里面，如下图：

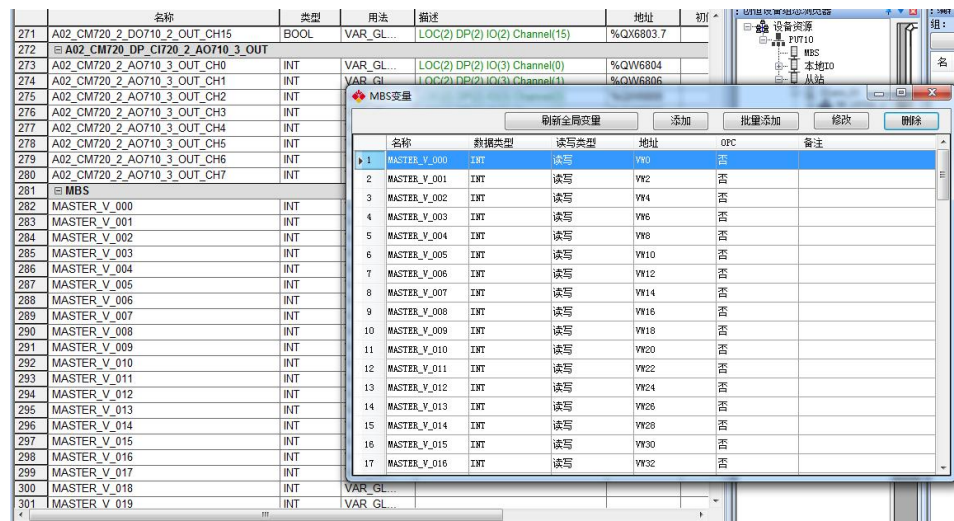


图 4-3-12

导入：点击“导入”按钮，选择导入文件类型



图 4-3-13

选择合法的变量点表文件：

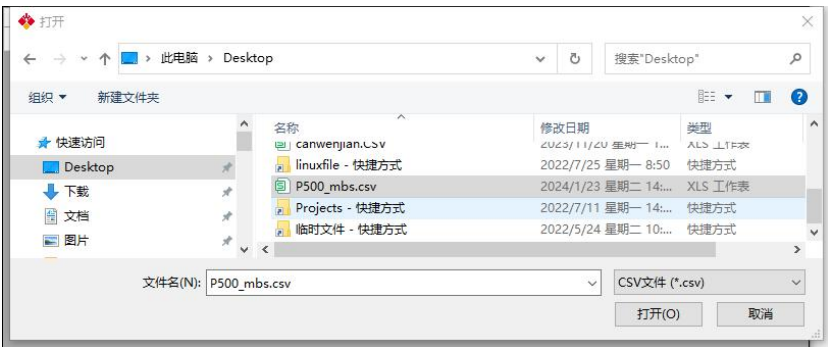


图 4-3-14

显示导入的变量点表：

MBS变量							
	名称	数据类型	读写类型	地址	位地址	OPC	备注
1	DJ_1_MF	BOOL	读写	0	0	☐	
2	DJ_1_MR	BOOL	读写	0	1	☐	
3	DJ_1_JT	BOOL	读写	0	2	☐	
4	YB_1_DRR	BOOL	读写	0	3	☐	
5	YB_1_DRO	BOOL	读写	0	4	☐	
6	YB_1_STO	BOOL	读写	0	5	☐	
7	DJ_1_MF_JC	BOOL	读写	0	6	☐	
8	DJ_1_MR_JC	BOOL	读写	0	7	☐	
9	DJ_1_JT_JC	BOOL	读写	1	0	☐	
10	YB_1_RN	BOOL	读写	1	1	☐	
11	YB_1_ALM	BOOL	读写	1	2	☐	
12	DJ_2_MF	BOOL	读写	1	3	☐	
13	DJ_2_MR	BOOL	读写	1	4	☐	
14	DJ_2_JT	BOOL	读写	1	5	☐	
15	YB_2_DRR	BOOL	读写	1	6	☐	
16	YB_2_DRO	BOOL	读写	1	7	☐	
17	YB_2_STO	BOOL	读写	2	0	☐	

图 4-3-15

并且可对点表查看修改，点击“确定”按钮，会自动逐条添加到全局变量表。打开全局变量表如下：

名称	类型	用法	描述
A00_AI610_01			
A00_DI610_02			
A00_DO610_03			
MBS			
mbs_b_var000	BOOL	VAR_GLOBAL	
DJ_1_MF	BOOL	VAR_GLOBAL	
DJ_1_MR	BOOL	VAR_GLOBAL	
DJ_1_JT	BOOL	VAR_GLOBAL	
YB_1_DRR	BOOL	VAR_GLOBAL	
YB_1_DRO	BOOL	VAR_GLOBAL	
YB_1_STO	BOOL	VAR_GLOBAL	
DJ_1_MF_JC	BOOL	VAR_GLOBAL	
DJ_1_MR_JC	BOOL	VAR_GLOBAL	
DJ_1_JT_JC	BOOL	VAR_GLOBAL	
YB_1_RN	BOOL	VAR_GLOBAL	
YB_1_ALM	BOOL	VAR_GLOBAL	
DJ_2_MF	BOOL	VAR_GLOBAL	
DJ_2_MR	BOOL	VAR_GLOBAL	
DJ_2_JT	BOOL	VAR_GLOBAL	

图 4-3-16

4.4 Profinet 组态

1、添加 Profinet 从站设备的 GSD 文件
在“创恒设备组态浏览器”窗口中，右键“设备资源”，选择“管理通用站描述文件(GSD)”。在弹出窗口中，根据存放 GSD 文件位置，选择合适的 GSD 文件进行安装。

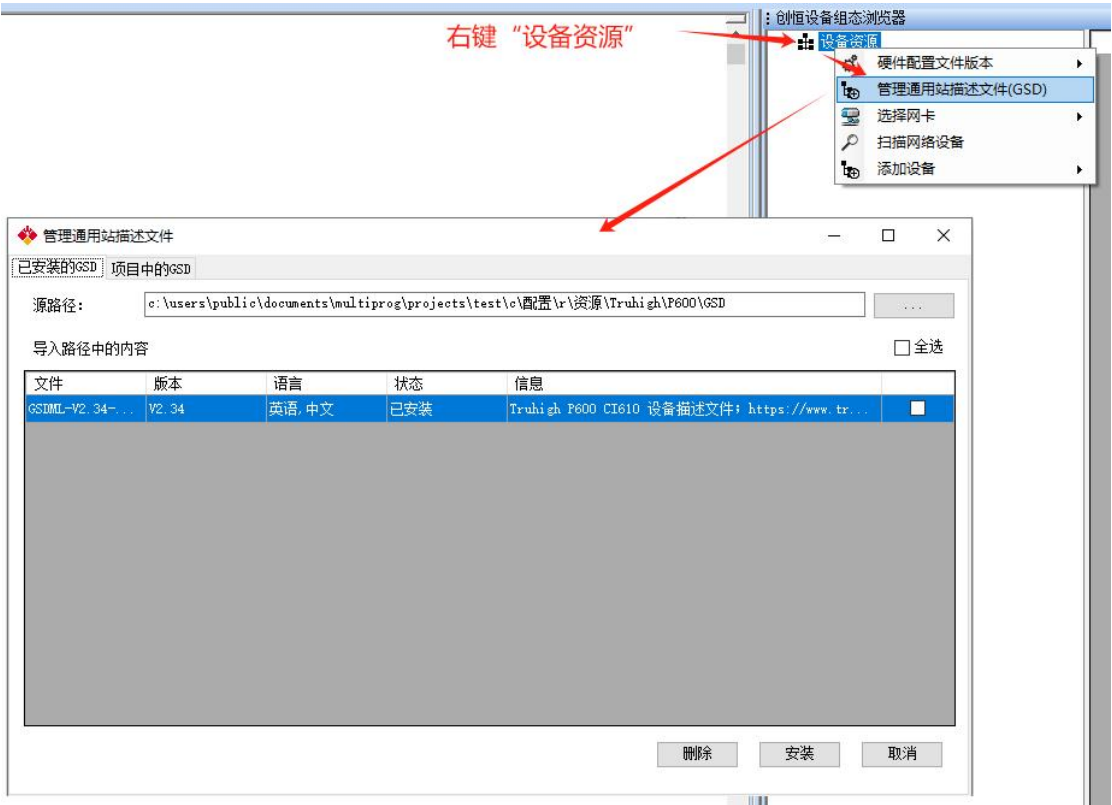


图 4-4-1

注意：对于同一 PC 同一设备的相同版本 GSD 文件安装一次即可。

2、添加控制器设备

右键“设备资源”，右键选择“添加设备”。根据需求，可直接添加“PU620”类型的设备作为控制器，也可通过设备扫描，将发现的设备添加为控制器。



图 4-4-2

添加完成后，设备列表如下：

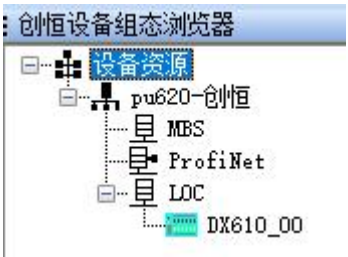


图 4-4-3

注意：添加扫描发现的设备，将自动填充相应的设备信息。

3、配置控制器参数

右键控制器名称，选择“编辑”，弹出“控制器设备”配置窗口。根据实际情况，配置控制器名称、IP 参数、OPCUA。

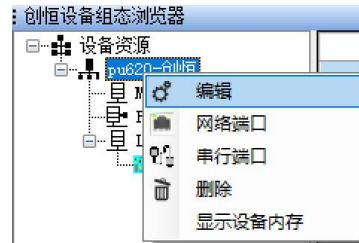


图 4-4-4



图 4-4-5

4、添加 Profinet 从站设备

右键控制器设备下的“Profinet”项，选择“添加”，出现两种选择。继续选择添加，是将已添加的 GSD 文件中的设备作为 profinet 从站设备；选择“网络设备”下的具体设备，则是将当前网络中发现的 profinet 从站设备加入到控制器组态中，并自动填充从站信息。



图 4-4-6

添加之前，两者都需要选择 GSD 文件版本。

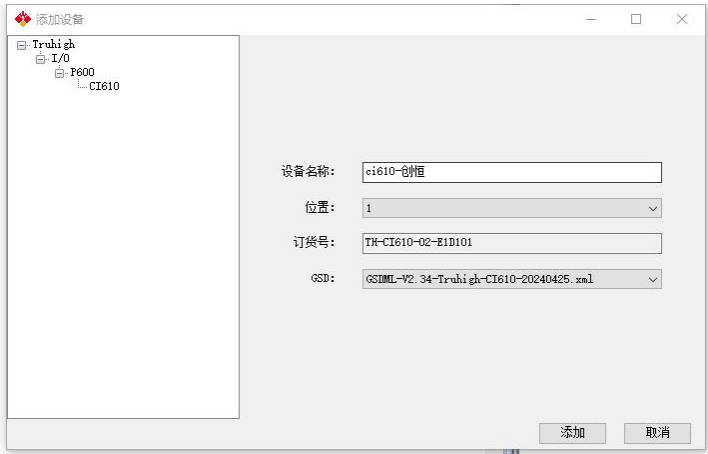


图 4-4-7

添加成功后设备组态如下：

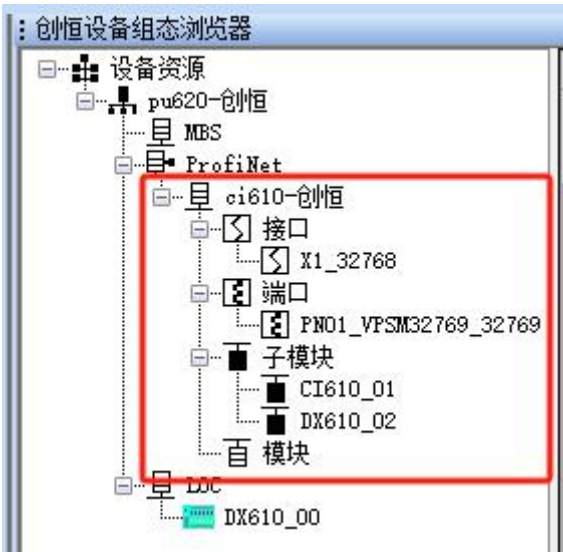


图 4-4-8

5、配置 Profient 从站

双击需要配置的从站设备，弹出配置界面，根据组态需求，填充设备名称与 IP 参数。

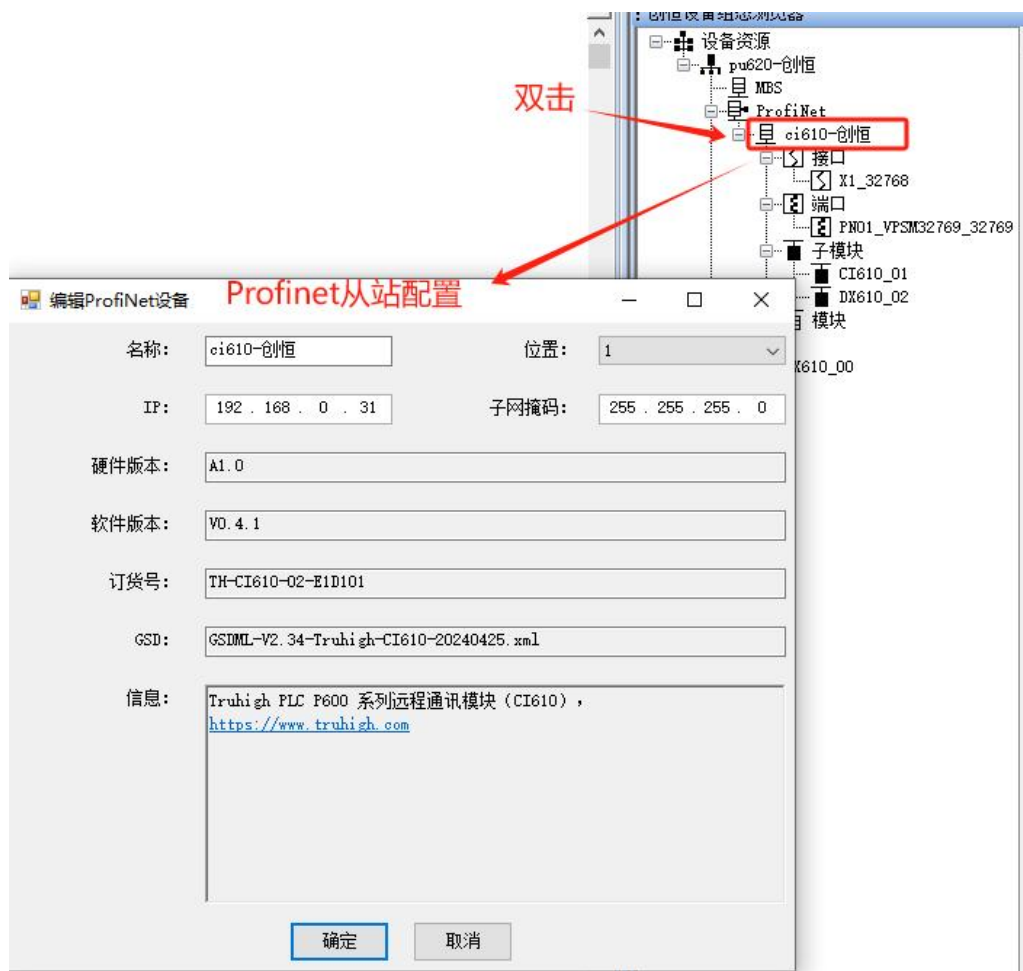


图 4-4-9

双击从站“接口”下的“X1_32768”，弹出“接口子模块”配置界面，配置 profinet 从站更新时间（数据轮询时间）与更新周期，更新时间 $\times$ 更新周期 = 看门狗时间。控制器在看门狗时间内，未收到从站数据，表示从站丢失。

6、向 Profinet 从站添加模块

双击相应 Profinet 从站设备下的“模块”，弹出“模块添加界面”，选择合适的模块指定相应位置进行添加。

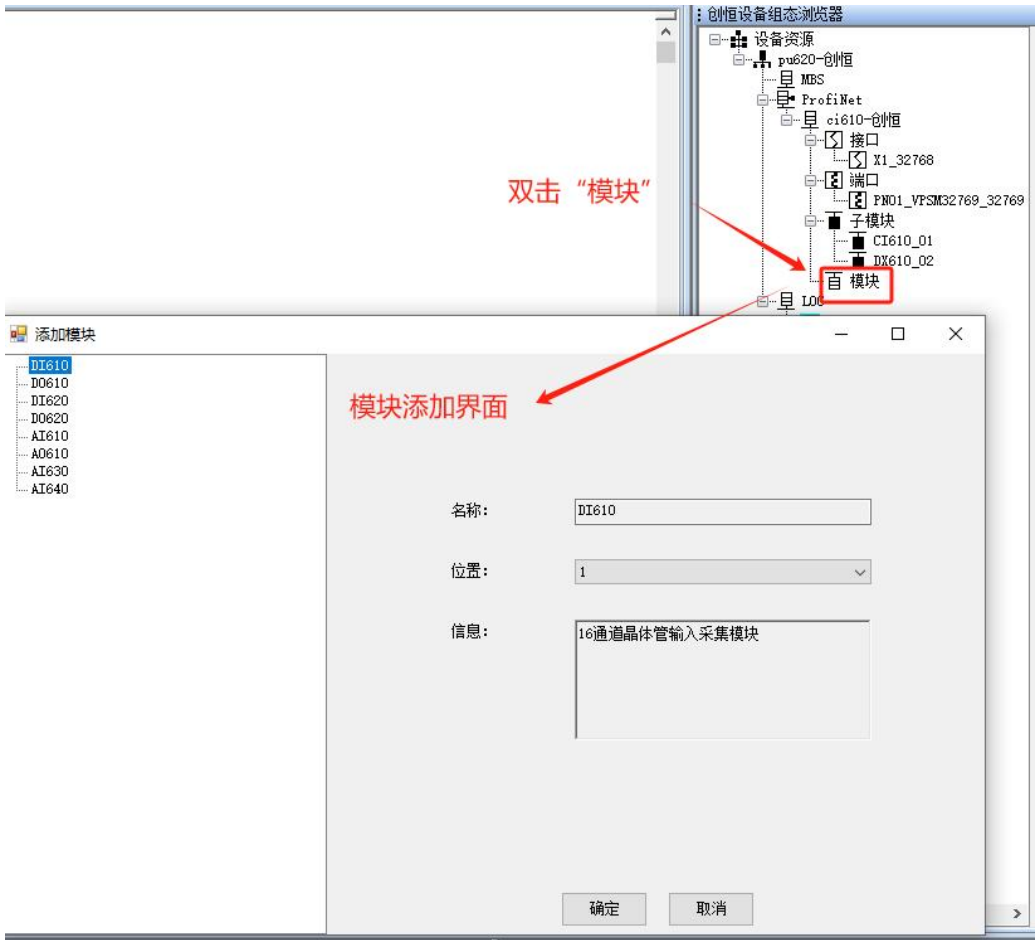


图 4-4-10

添加“DI610”、“DO610”、“AI610”、“AO610”后，组态如下：

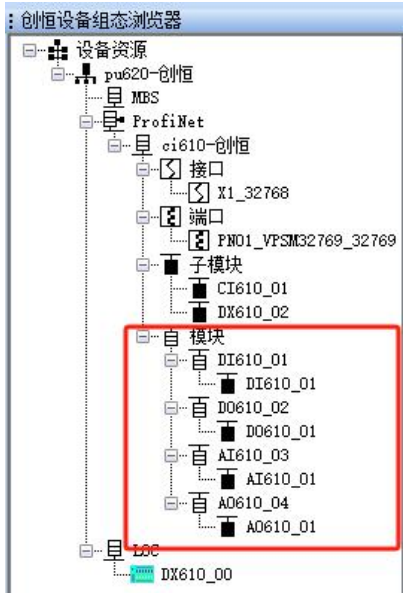


图 4-4-11

7、配置模块参数

以“AO610”模块为例，右键“AO610_04”模块下的子模块“AO610_01”，选择“参数”选项，弹出参数配置界面，如下：

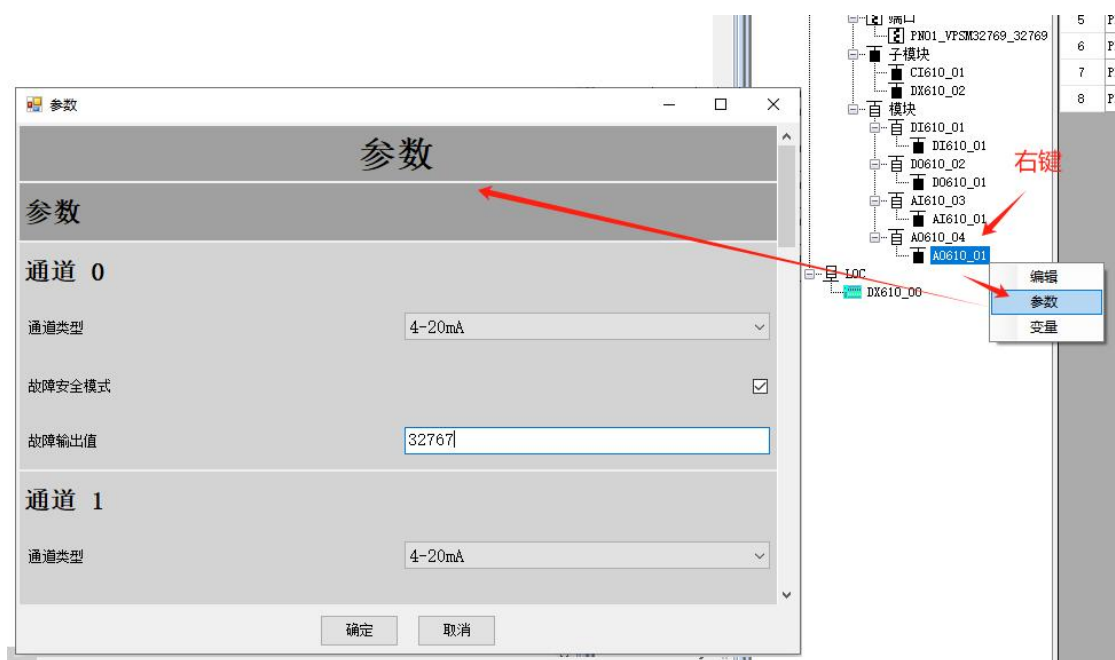


图 4-4-12

根据组态需求，可配置 AO 模块的通道类型与故障输出功能。

第 5 章 用户程序编辑

MULTIPROG 支持 IL、FBD、ST、LD 和 SFC 五种编程语言，其中 IL 和 ST 属于文本语言，FBD、LD 和 SFC 属于图形语言。一个完整的程序包括数据变量和程序代码两部分，数据变量要在变量工作单中声明，代码要用 IL、FBD、ST、LD、SFC 中的一种或者几种的组合来编写。

本章分为以下九节：

- 5.1 POU 编程
- 5.2 IL 指令表编程语言
- 5.3 ST 结构化文本编程语言
- 5.4 FBD 功能块图编程语言
- 5.5 LD 梯形图编程语言
- 5.6 SFC 顺序功能图编程语言
- 5.7 各类 POU 的调用
- 5.8 工程编译
- 5.9 工程仿真

5.1 POU 编程

Truhigh P600 系列 PLC 使用 MULTIPROG 编程软件，其编程语言和程序结构符合 IEC61131-3 编程系统，在 IEC61131-3 中，程序和工程的建立是在程序组织单元（POU）中完成的。

POU 是程序组织单元（Program Organization Unit）的缩写，是 MULTIPROG 软件的基本构成元件，用户基于程序组织单元编写应用程序。

POU 分为程序（Program）、功能块（Function Block）和函数（Function）三种类型。其中程序是主体，程序可以调用功能块和函数，构成用户的应用程序。

5.1.1 变量工作单

变量工作单是程序组织单元 POU 所对应的变量表。打开任意 POU 后，点击工具栏上的变量工作单按钮, 打开变量工作单，可以查看和编辑 POU 内的变量，如图 5-1-1。

名称	类型	用法	描述
1			
C000	BOOL	VAR_EX...	
C001	BOOL	VAR_EX...	

图 5-1-1

右键点击变量，选择属性，可以查看修改变量的属性，如图 5-1-2。

属性

名称(N):
C000

数据类型(D):
BOOL

用法(U):
VAR_EXTERNAL

初值(I):

I/O地址(S):

描述(E):

☐ PDD

☐ OPC

☐ 隐藏(X)

☒ 初值作为默认值。

确定

取消

帮助(H)

图 5-1-2

对话框'属性'也可以用来同时编辑几个声明中的属性。为此，首先您必须在变量表格中选择要选的声明，然后调用对话框。在这种情况下，'名称'域包括未被改变的入口'<<变量采集>>'。

如果已经用不同值(例如，使用'VAR'和'VAR_EXTERNAL'或者类型'CTD'和'CTU')标记了声明，那么适当的对话框域包括入口'<<不同值>>'。在改变这些值之前，您应该早考虑到改变的结果。

如果为已标记的声明(例如，为一个声明复选框'OPC'被设置和为其它声明不被设置)复选框被设置不同，那么它们出现在亮灰色中。为了所有已选择的声明激活或者不激活它们，多次点击相关的复选框，如图 5-1-3。



图 5-1-3

下面介绍变量参数：

对话框元素	含义
名称	显示变量/实例的名字。 当调用对话框'属性'时，在变量表格中，如果几个变量被标记，那么'名称'域处于不活动状态并且显示入口'<<变量采集>>'。
数据类型	在变量情况下：指定变量的数据类型。数据类型决定变量可以具有哪种数值，并且它们定义初值、可能值的范围以及所占的二进制位数。
用法	在变量情况下：指定变量关键字。 在程序类 POU 中，只能选择 VAR, VAR_EXTERNAL 和 VAR_EXTERNAL_PG。 在功能块类 POU 中，该列表框包含了 VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT 和 VAR_EXTERNAL 等关键字。 在功能类 POU 中，只能选择 VAR 和 VAR_INPUT 关键字。
RETAIN	在变量情况下：

	表示(将要)声明一个保持型变量。保持型变量是那些即使在关掉电源的情况下仍保存值的变量。在进行一个暖启动的情况下，将使用变量的最后值。
初值 /默认 的 隐藏值	在变量情况下： 指定初值。 初值可以赋给那些具有 VAR、VAR_GLOBAL、VAR_INPUT 和 VAR_OUTPUT 等用法的变量。初值不能用于 VAR_EXTERNAL 声明。 初值必须符合所选择的数据类型。 在带有默认隐藏值的隐藏形式参数情况下(见下面对‘初值作为默认值’复选框的描述)，这个对话框区域指定了默认使用的隐藏值，它是在您将这个 POU 插入到另一个图形工作单时被默认地使用的。 在 FB 实例情况下： 不适用于 FB 实例。该域处于不活动状态。
I/O 地址	在变量情况下： 指定变量地址，如果你想声明一个指定了地址的变量。 在 FB 实例情况下： 不适用于 FB 实例。该域处于不活动状态。
描述	用户定义的注释。
PDD	在变量情况下：指明变量已写入过程数据目录（PDD）中。 只有当您想访问 PLC 上的某个地址对应的变量名称时，才需要用到它。 在 FB 实例情况下： 不适用于 FB 实例。该复选框处于不活动状态。
OPC	在变量情况下：指明变量已写入*.csv 文件中。 只有当您想通过使用诸如 OPC 服务器等来访问变量，以便运行某个可视化客户机程序时，才需要用到它。 在 FB 实例情况下： 不适用于 FB 实例。该复选框处于不活动状态。
隐藏	仅仅可用于功能或功能块 POU 的输入和输出变量(使用 VAR_INPUT 和 VAR_OUTPUT 关键字声明的)。 如果标记了这个复选框，则在另一个图形工作单中调用(插入)这个用户 POU 时，特定的输入/输出变量将不能作为块的形式参数出现。
初 值 作 为 默 认 值	仅仅可用于隐藏的形式参数(VAR_INPUT/VAR_OUTPUT 声明)，如上面的表行中所述。 如果这个复选框被标记，‘初值’域中所输入的值被用作‘默认隐藏值’(即使该输入域相应地被重新命名 - 见上面的‘初值’域的描述)。

5.1.2 新建逻辑 POU

在工程树目录下单机工程，在“逻辑 POU”上点击右键，选择插入，点击程序，如图 5-1-4，后者按下了‘INS’键。

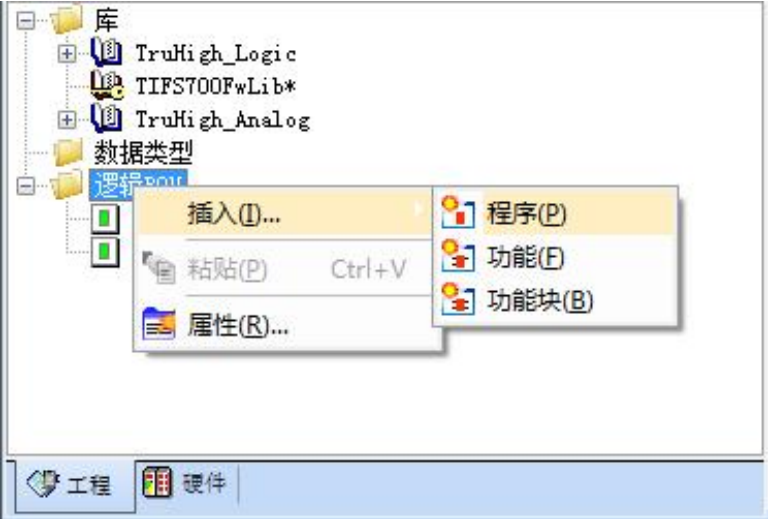


图 5-1-4

执行插入“程序”，弹出插入对话框。



图 5-1-5

该对话框用于插入一个新的 POU，并指定其属性

对话框元素	含义
名称	指定对象名称。最多 24 个字符。
类型	指定要插入对象的类型。取决于您想要插入新对象的位置，一些单选按钮可能处于不活动状态。例如，您不能在 POU 内插入程序，或者在 FBD 功能块内插入动作工作单。
语言	说明要插入对象的编程语言。如果您想在某个 POU 内插入代码本体工作单，则可用编程语言已经由该 POU 语言所确定。 取决于您想要插入新对象的位置，当插入一个新的工作单时，有些单选按钮可能是不活动的。例如，您不能在 POU 内插入数据类型工作单。
返回值的 数据类型	指定返回值的数据类型。只有在功能情况下才是有效的。当功能返回一个值时，该函数名称就是该输出值。因此，在功能 POU 中没有要声明的输出变量（使用 VAR_OUTPUT 声明关键字）。
使用保留	使用用于修补 POU 的 POU 保留存储区。‘使用保留’的值是在‘数据区域’对话框中输入的，该对话框是通过点击您 PLC 的‘资源设置...’对话框中的‘数据区域...’按钮来调用的。
模式	指明新对象是在所标记的对象之前（单选按钮‘插入’）还是之后（单选按钮‘追加’）插入。
PLC 类型	指定 PLC 的类型。该列表框的默认表目（在打开对话框时就被选定）是在工程属性对话框中设置的。
处理器类型	指定处理器类型。使用默认值。

设置完成后，单击“确认”按钮，完成 POU 的创建，并在工作区域打开此 POU 的编辑窗口。

5.1.3 逻辑 POU 的类型

POU 有 3 种类型：程序(Program)、功能块(Function Block)、函数(Function)。

➤ 程序 (Program)

程序是为了完成某项任务而编写的语句序列，或一组指令的集合。程序是惟一可执行的应用程序的主体，程序可以调用功能块、函数和其他程序。程序可以通过任务组态来激活，也可以通过其他程序来调用。但是，这种调用与函数以及功能块的调用是不同的，这种调用导致值被改变时，改变的是被调用程序本身的值，而函数和功能块被调用时，改变的是调用它的程序实例中的值。

➤ 功能块 (Function Block)

功能块是预先编好的、实现某种运算的程序。功能块本身不能单独执行，可以由程序和功能块调用功能块执行。功能块的输入量可以是一个或多个，输出量可以是一个或多个执行结果。和函数不同，功能块本身没有返回值。功能块声明以关键字

➤ 函数 (Function)

函数是预先编好的、实现某种运算的程序。函数本身不能单独执行，只能由程序或功能块调用执行。函数的输入量可以是一个或多个，输出量只有一个执行结果。函数在被执行时，会对一系列特定的输入产生一个唯一数据类型的输出结果，这个输出结果被赋给函数本身。

5.1.3.1 程序

1. 新建程序

在“插入”对话框中，选择类型为“程序(G)”，名称为 PRO1，语言为 ST，编程语言详见《编程语言》章节，如图 5-1-6。



图 5-1-6

2. 声明变量

	名称	类型	用法	描述
	Default			
	a	REAL	VAR_EX...	
	b	REAL	VAR_EX...	
	c	REAL	VAR_EX...	

图 5-1-7

3. 编写程序

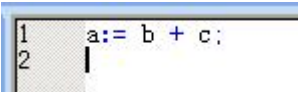


图 5-1-8

4. 程序实例化

程序可以在资源内部被实例化。程序实例的创建只需将一个程序关联到一个任务，并在相应的对话框中输入实例名称即可。

将一个新的程序 POU 插入到工程树的“逻辑 POU”文件夹中之后，系统将自动创建相关的程序实例，并将其插入到硬件树的‘任务’文件夹中。通过这种方法，保证了您的工程中每个已经开发的程序都由 PLC 实际执行。然而，也可能您从‘任务’文件夹中删除了某个实例，那么必须要手工地重新插入它，否则程序不执行。



图 5-1-9

5.1.3.2 功能块

1. 新建功能块

在“插入”对话框中，选择类型为“功能块(U)”，名称为 FB1，语言为 ST，编程语言详见《编程语言》章节，如图 5-1-10。



图 5-1-10

2. 功能块声明变量如下：

名称	类型	用法	描述
Default			
a	REAL	VAR_OUTPUT	
b	REAL	VAR_INPUT	
c	REAL	VAR_INPUT	
m	BOOL	VAR_OUTPUT	

图 5-1-11

3. 功能块编写程序如下：

```
1  a:= b + c;  
2  m:=(a = b);  
3
```

图 5-1-12

4. 功能块实例化：

因为功能块总是具有一个内部存储区，所以，每当该功能块被用于一个不同的存储区域时，有必要将它们的值保存起来。

实例化意味着，一个功能块被定义一次而被使用多次。每当用编辑向导插入一个功能块时，都会自动创建实例名称。

通过 LD 语言编写的程序（例如程序名为：LD_PRO）调用功能块：
程序 LD_PRO 变量声明如下：

名称	类型	用法	描述
Default			
V000	REAL	VAR_EXTERNAL	
V001	REAL	VAR_EXTERNAL	
V002	REAL	VAR_EXTERNAL	
FB1_1	FB1	VAR	
V003	BOOL	VAR_EXTERNAL	

图 5-1-13

5. 在编辑向导中找到<My_Proj>中，找到自定义的功能块 FB1。

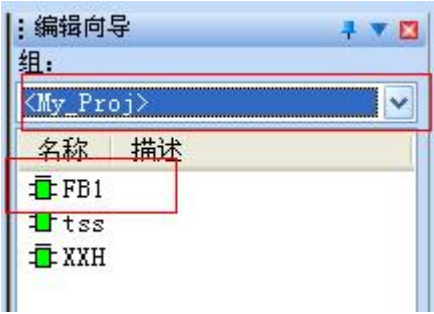


图 5-1-14

6. 然后把功能块拖动到程序 LD_PRO 编辑区，然后出现变量属性对话框，

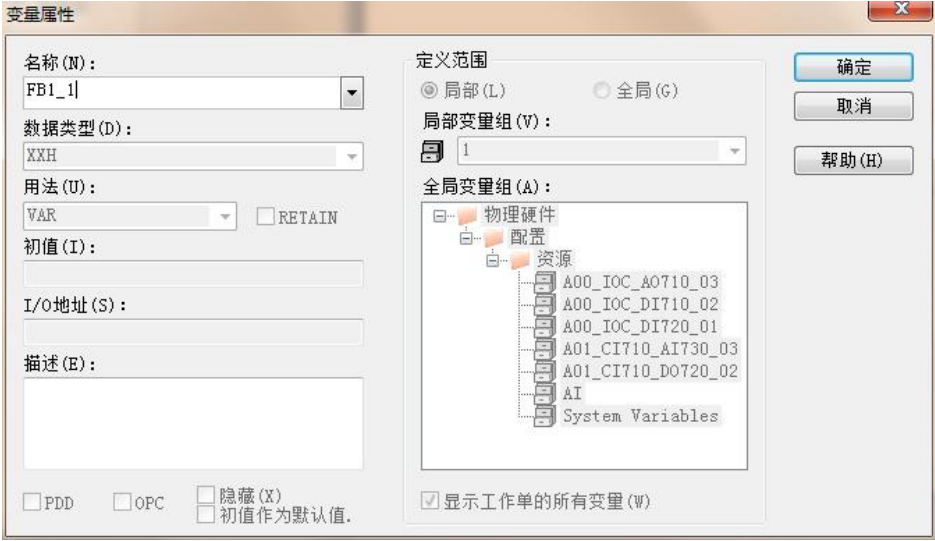


图 5-1-15

7. 定义名称，然后点击确定，如图 5-1-16。

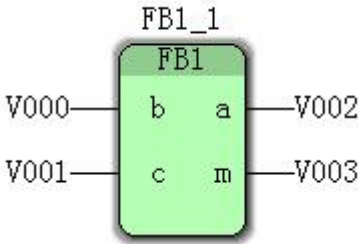


图 5-1-16

下面是一个标准 IEC 61131-3 功能块列表：

- 双稳态元素，如 SR 和 RS
- 边沿检测功能块，如：R_TRIG 和 F_TRIG
- 计数器，如：CTU 和 CTD
- 定时器功能块，如：TON 和 TOF

功能列表详见《Truhigh P600 功能块手册》。

5.1.3.3 功能

功能是不包括任何内部状态信息的 POU。对于带有相同输入参数功能的调用，总会产生相同的输出值，功能只有一个输出。

1. 新建功能

在“插入”对话框中，选择类型为“功能 (B)”，名称为 FU1，语言为 ST，编程语言详见《编程语言》章节，如图 5-1-17。



图 5-1-17

 提示:

函数不能选择 SFC 语言。

功能的返回值，就是该功能的输出值。因此，在功能 POU 中没有要声明的输出变量（使用 VAR_OUTPUT 声明关键字）。

2. 声明变量如下:

	名称	类型	用法	描述
	Default			
	a	REAL	VAR_INPUT	
	b	REAL	VAR_INPUT	

图 5-1-18

3. 编写程序如下:

```
1  FU1 := a + b;  
2
```

图 5-1-19

4. 功能调用

通过 LD 语言编写的程序（例如程序名为：LD_PRO）调用功能：
程序 LD_PRO 变量声明如下：

名称	类型	用法	描述
Default			
V000	REAL	VAR_EXTERNAL	
V001	REAL	VAR_EXTERNAL	
V002	REAL	VAR_EXTERNAL	
FB1_1	FB1	VAR	
V003	BOOL	VAR_EXTERNAL	
XXH_1	XXH	VAR	
V004	REAL	VAR_EXTERNAL	
V005	REAL	VAR_EXTERNAL	
V006	REAL	VAR_EXTERNAL	

图 5-1-20

5. 在编辑向导中找到<My_Proj>中，找到自定义的功能 FU1。



图 5-1-21

6. 把功能 FU1 拖到程序 LD_PRO。

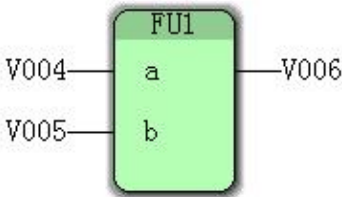


图 5-1-22

下面是一个标准 IEC 61131-3 功能列表：

- 类型转换功能，如 INT_TO_REAL
 - 数值功能，如：ABS 和 LOG
 - 标准算术运算功能，如：ADD 和 MUL
 - 位串功能，如：AND 和 SHL
 - 选择和比较功能，如：SEL 和 GE
 - 字符串功能，如：RIGHT 和 INSERT
 - 时间数据类型功能，如带有 TIME 数据类型的 SUB
- 功能列表详见《Truhigh P600 功能块手册》。

5.2 IL 指令表编程语言

指令表编程语言的基本语句是指令表，它是一种底层语言，采用面向机器的操作符，比较容易转换成可编程控制器的机器代码。但是指令表语言不适合阅读，不适合做大型复杂的控制任务。

5.2.1 创建一个 IL 程序

使用 IL 编程，用户可在编程区键入代码，对于指令，可以直接键入或者从‘编辑向导’中拖拽指令到编辑区。

下面写一个加法运算的程序为例，说明 IL 的编程过程。

1. 创建一个工程。
2. 插入 IL 语言程序

右键工程树中的‘逻辑 POU’，选择插入-程序，弹出的对话框中键入程序名，如 IL_PRO，如图 5-2-1。点击确定，在逻辑 POU 中出现了建立的‘IL_PRO’程序。



图 5-2-1

3. 双击打开 ‘IL_PRO’ 程序，在编辑向导中选择 ‘<全部 FU 和 FB>’，如图 5-2-2。

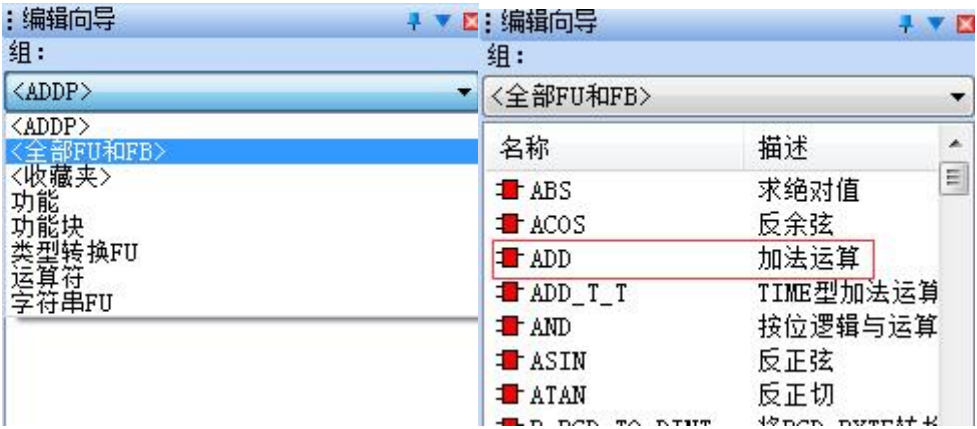


图 5-2-2

4. 双击编辑向导中的 ‘ADD’ 函数，在编辑区域出现了加法运算符，然后把程序补充完整，如图 5-2-3。

```
1 LD (* IN1 作为 ANY_NUM *)
2 ADD (* IN2 作为 ANY_NUM *)
3 ST (* 结果 作为 ANY_NUM *)
4 |
```

图 5-2-3

5. 打开变量工作单，新建变量 IN1、IN2、OUT，类型为 REAL，用法为 VAR，如图 5-2-4。

名称	类型	用法
Default		
IN1	REAL	VAR
IN2	REAL	VAR
OUT	REAL	VAR

图 5-2-4

6. 点击制作，并下载程序，然后打开调试开关。分别在变量 IN1、IN2、和 OUT 上右键点击 ‘添加到监视窗口’，然后在打开监视窗口，如图 5-2-5。

变量	值	类型	实例
IN1	0.0000000	REAL	配置.资源.任务.IL_PRO.IN1
IN2	0.0000000	REAL	配置.资源.任务.IL_PRO.IN2
OUT	0.0000000	REAL	配置.资源.任务.IL_PRO.OUT

图 5-2-5

7. 分别双击监视窗口中的 IN1、IN2 变量，写入 10.0 和 15.0，OUT 的值则为 25.0，如图 5-2-6、5-2-7。



图 5-2-6

变量	值	类型	实例
IN1	10.0000000	REAL	配置.资源.任务.IL_PRO.IN1
IN2	15.0000000	REAL	配置.资源.任务.IL_PRO.IN2
OUT	25.0000000	REAL	配置.资源.任务.IL_PRO.OUT

图 5-2-7

至此一个完整用 IL 语言编写的程序就完成了。

5.2.2 IL 的语句

在 IL 语句中包含的元素有：指令、标号、修饰符、操作数、算术运、注释。

一个用文本语言 IL 编写的代码包括了一个指令序列。每条指令都从一个新行开始。每一行的前面显示了一个行号。每一行从一个操作符开始，后面跟着一个操作数。而且，可以用修饰符。IL 中的操作数可以是变量、直接量或功能块实例名称。操作数是与指令中的操作符或功能一起来使用的。由于语法突出显示，以不同颜色来显示变量元素：操作符及其修饰符为蓝色，变量和操作数为黑色，注释为绿色，如图 5-2-8。

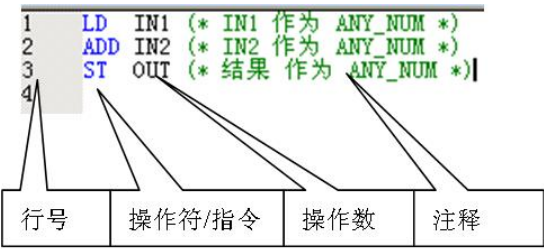


图 5-2-8

5.2.3 IL 的操作符

表 5-1

操作符	修饰符	操作数	描述
LD	N	ANY	将当前结果设定为等于操作数
ST	N	ANY	将当前结果保存到操作数位置
S	1	BOOL	如果当前结果为 1，则将布尔操作数设定为 1
R	1	BOOL	如果当前结果为 1，则将布尔操作数设定为 0
AND	N, (	ANY_BIT	布尔与
OR	N, (	ANY_BIT	布尔或
XOR	N, (	ANY_BIT	布尔异或
ADD	(	ANY_NUM	加法运算
SUB	(	ANY_NUM	减法运算
MUL	(	ANY_NUM	乘法运算
DIV	(	ANY_NUM	除法运算
GT	(	ANY_NUM + ANY_BIT	比较： >
GE	(	ANY_NUM + ANY_BIT	比较： >=

EQ	(	ANY_NUM + ANY_BIT	比较： =
NE	(	ANY_NUM + ANY_BIT	比较： <>
LE	(	ANY_NUM + ANY_BIT	比较： <=
LT	(	ANY_NUM + ANY_BIT	比较： <
JMP	C, N	LABEL	跳转到标号
CAL	C, N	NAME	调用功能块
RET	C, N	1	从所调用的功能或功能块返回

修饰符中 N 代表取反，C 代表只有在当前运算结果为真是才执行。

5.3 ST 结构化文本编程语言

结构化文本编程语言是高层语言编程语言，类似于 Pascal 编程语言。它用高度压缩的方式提供大量抽象语言来描述复杂的控制系统。

用文本语言 ST 编写的代码本体由语句和表达式组成。表达式是一个对语句的执行返回一个值的结构。表达式由操作符和操作数组成。

5.3.1 创建一个 ST 程序

编辑 ST 语言代码本体工作单有两种方式：使用文本编辑器输入语句和表达式，或使用编辑向导插入它们。推荐使用向导插入，因为它更方便且能防止错误。使用 ST 语言时，编辑向导包含了许多标准关键字、功能和功能块可供插入。

下面创建一个小信号切除的程序：当输入信号 IN 小于 10.0 的时候，输出 OUT 等于 0.0，当输入 IN 大于或者等于 10.0 的时候，输出 OUT 等于输入 IN。

1. 在工程树中逻辑 POU 上右键点击插入，创建一个 ST 语言的程序 ST_PRO，如图 5-3-1。



图 5-3-1

2. 双击打开新建的程序 ST_PRO，点击工具栏的变量工作单，创建变量 IN 类型为：REAL，用法为 VAR。变量 OUT 类型为 REAL，用法为 VAR，如图 5-3-2。

	名称	类型	用法	描述
1	OUT	REAL	VAR	
	IN	REAL	VAR	

图 5-3-2

3. 打开编辑区，在编辑向导中，打开‘组’列表框，并选择‘关键字’，如图 5-3-3。



图 5-3-3

4. 将关键字 ELSE 拖动到编辑区，如图。标记绿色占位符，并用实际的运算符(变量和值)替换它们。

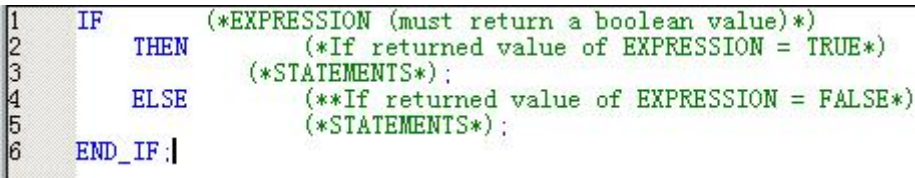


图 5-3-4

5. 然后补充语句，如图 5-3-5，然后点击保存。

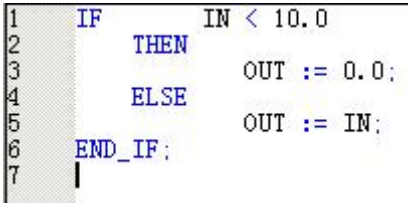


图 5-3-5

6. 点击工具栏上的制作按钮 ，在没有错误和警告的情况下，点击工具栏上的工程控制对话框按钮 ，然后在工程控制对话框中点击下载。下载完成之后，打开工具栏上的调试开关 ，如图 5-3-6。

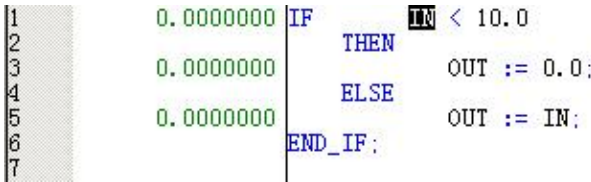


图 5-3-6

7. 分别选择变量 IN 和 OUT，右击，选择添加到监视窗口，如图 5-3-7。

变量	值	类型	实例
IN	0.0000000	REAL	配置.资源.任务.ST_PRO. IN
OUT	0.0000000	REAL	配置.资源.任务.ST_PRO. OUT

图 5-3-7

8. 双击监视窗口中的变量 IN，写入 15.0，然后点击覆盖，如图 5-3-8。可以在监视窗口查看 OUT 的值为 15.0，如图 5-3-9。



图 5-3-8

变量	值	类型	实例
IN	15.0000000	REAL	配置.资源.任务
OUT	15.0000000	REAL	配置.资源.任务

图 5-3-9

9. 双击监视窗口中的变量 IN，写入 5.0，然后点击覆盖，如图 5-3-10。可以在监视窗口查看 OUT 的值为 0.0，如图 5-3-11。



图 5-3-10

变量	值	类型	实例
IN	5.0000000	REAL	配置.资源.任务
OUT	0.0000000	REAL	配置.资源.任务

图 5-3-11

至此，一个完整的 ST 语言编写的程序就完成了。

5.3.2 ST 的语句

在结构化文本编程语言 ST 中，程序由语句组成，语句由表达式和关键字组成。如一条赋值语句，它将位于右侧的表达式的值复制给位于左侧的变量，如下面的范例所示：

`variableName := expression;`

对于赋值语句来说，很重要的一点是位于左侧的变量与位于右侧的表达式值要具有相同的数据类型。否则，必须使用类型转换。

在用 ST 语言编辑程序时，除了赋值语句，还可以用更多的语句（如选择语句、循环语句或返回语句）。语句的关键字、语句的使用举例及其含义在下表中。

表 5-2

关键字	举例	描述
RETURN	RETURN;	返回语句:返回语句退出被调用的功能、功能块或程序，并返回到调用 POU。
IF	IF a < b THEN c:=1; ELSIF a=b THEN c:=2; ELSE c:=3; END_IF;	选择语句:仅当所关联的布尔表达式‘a<b’的值为 TRUE 时，才执行第一组语句。如果该条件为 FALSE，则不执行任何语句或执行跟在 ELSE 后面的那组语句。
CASE	CASE f OF 1: a:=3; 2..5: a:=4; 6: a:=2; b:=1; ELSE a:=0; END_CASE;	选择语句:根据 CASE 关键字后面的表达式的值，执行一组语句。变量或表达式‘f’必须为 INT 数据类型。
FOR	FOR a:=1 TO 10 BY 3 DO f[a] :=b; END_FOR;	循环语句:一组语句被反复执行，同时，将变量‘a’从 1 开始，以 3 递增，到 10 结束。起始点由赋给控制变量‘a’的值指明。终值有跟在‘TO’后面的值指明，而递增量由‘BY’后面的值表明。所有值必须具有 ANY_INT 数据类型。 注意：如果‘BY’没有出现，就使用默认值‘1’。在这种情况下，所有值必须是 INT 数据类型。
WHILE	WHILE b > 1 DO b:= b/2; END_WHILE;	循环语句:一组语句被反复执行，直到所关联的布尔表达式‘b>1’的值为 FALSE。语句的条件在循环的开始处执行。如果条件值为 FALSE，则不执行循环体。

REPEAT	REPEAT a := a*b; UNTIL a < 10000 END_REPEAT;	循环语句.一组语句被反复执行,直到所关联的布尔表达式‘a<10000’的值为TRUE。语句的条件在循环的结束处执行。如果条件值为FALSE,则循环至少被执行一次。
EXIT	FOR a:=1 TO 2 DO IF flag THEN EXIT; END_IF SUM:= SUM + a END_FOR	退出语句。退出语句可被用于中止循环语句的执行。

5.3.3 ST 的表达式

表达式是一个对语句的执行返回一个值的结构,它由运算符和操作数组成。在表达式里,运算符根据其优先级而被应用到操作数。在表达式里也可以调用功能。操作数可以是直接量、变量或功能名称。

在 ST 代码本体中,可以使用下列运算符:

表 5-3

操作符	举例	例子的值	描述	优先级
()	(2+3)*(4+5)	45	括号	最高
**	3.0**4	81.0	乘幂运算	1
-	-10	-10	求相反数	1
NOT	NOT TRUE	false	按位取反	1
*	10*3	30	乘法运算	1
/	6/2	3	除法运算	1
MOD	17 MOD 10	7	模数运算	1
+	2+3	5	加法运算	1
-	4-2	2	减法运算	1
<,>, <=,>=	4 > 12	false	比较	1
=	T#26h = T#1d2h	true	相等	1
<>	8 <> 16	true	不相等	1
&, AND	TRUE & FALSE	false	布尔与	1
OR	TRUE OR FALSE	true	布尔或	1
XOR	TRUE XOR FALSE	true	布尔异或	最低

5.4 FBD 功能块图编程语言

功能块图编程语言 FBD 是 IEC 标准的基础，一个功能块图编程语言的程序是将各种功能块连接起来，编程语言的元素是函数、功能块和连接符号。

下边用 FBD 编写一个用来计算一元二次方程 $y=aX^2+bX+c$ 。

5.4.1 创建一个 FBD

1. 在工程树中逻辑 POU 上右键点击插入，创建一个 FBD 语言的程序 FBD_PRO，如图 5-4-1。



图 5-4-1

2. 打开新建的程序 FBD_PRO，打开变量工作单，添加变量 y、x、a、b、c，为 REAL 类型，用法为 VAR，如图 5-4-2。

	名称	类型	用法	描述
	Default			
	a	REAL	VAR	
	b	REAL	VAR	
	c	REAL	VAR	
	y	REAL	VAR	
	x	REAL	VAR	

图 5-4-2

3. 在编辑向导中找到 ADD、MUL 和 EXPT 三个功能块，将他们拖拽到编辑区，用线链接起来，如图 5-4-3。

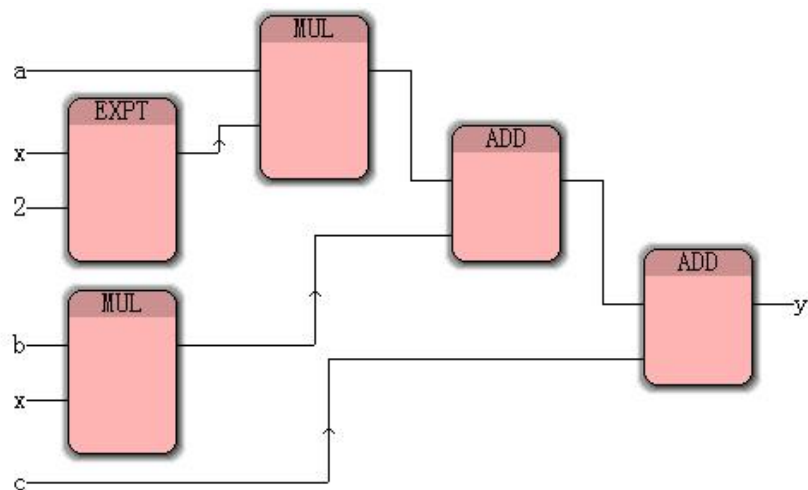


图 5-4-3

4. 点击工具栏上的制作按钮 ，在没有错误和警告的情况下，点击工具栏上的工程控制对话框按钮 ，然后在工程控制对话框中点击下载。下载完成之后，打开工具栏上的调试开关 ，如图 5-4-4。

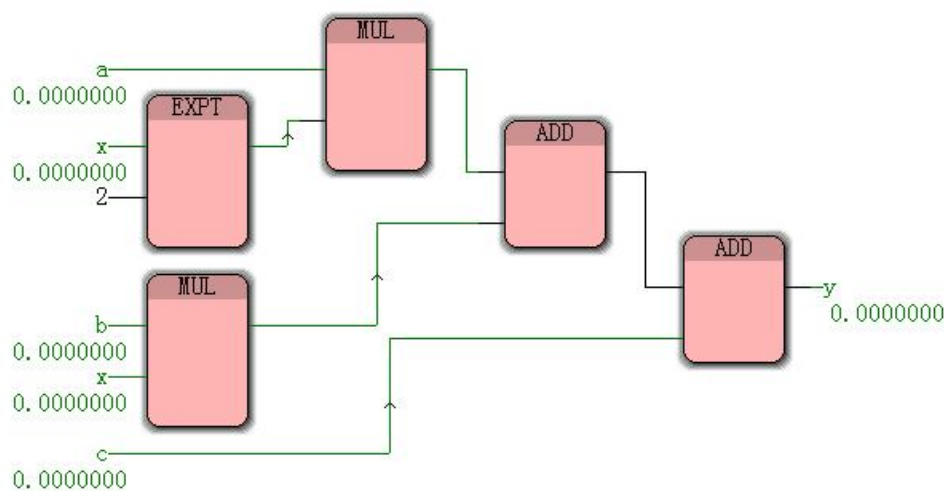


图 5-4-4

5. 分别把变量 a、b、c、x、y 添加的监视窗口，然后打开监视窗口，如图 5-4-5。

变量	值	类型	实例
a	0.0000000	REAL	配置.资源.任务.FBD_PRO1.a
b	0.0000000	REAL	配置.资源.任务.FBD_PRO1.b
c	0.0000000	REAL	配置.资源.任务.FBD_PRO1.c
x	0.0000000	REAL	配置.资源.任务.FBD_PRO1.x
y	0.0000000	REAL	配置.资源.任务.FBD_PRO1.y

图 5-4-5

6. 给变量赋值 a=2, b=3, c=4, x=5, 则可以看到结果 y=81, 如图 5-4-6。

变量	值	类型	实例
a	2.0000000	REAL	配置.资源.任务.FBD_PR01.a
b	3.0000000	REAL	配置.资源.任务.FBD_PR01.b
c	4.0000000	REAL	配置.资源.任务.FBD_PR01.c
x	5.5000000	REAL	配置.资源.任务.FBD_PR01.x
y	81.0000000	REAL	配置.资源.任务.FBD_PR01.y

图 5-4-6

FBD 编程语言和 LD 编程语言相似。
至此，一个完整的 FBD 编写的程序就完成了。

5.5 LD 梯形图编程语言

梯形图编程语言 LD 是历史最久远的一种编程语言，梯形图源于电气系统的逻辑控制图，逻辑图采用继电器、触点、线圈和逻辑关系图等表示它们之间逻辑关系。梯形图编程语言采用的图形元素有梯形图网络，电源轨线、连接导线、触点、线圈和功能等。

LD 网络可以与 FBD 对象相结合。通过这种方法，可以调用功能和功能块，并且可以将其连接到变量。另外，可以使用跳转、标号、连接符和返回。

5.5.1 创建一个 LD 程序

1. 在工程树中逻辑 POU 上右键点击插入，创建一个 LD 语言的程序 LD_PRO，如图 5-5-1。



图 5-5-1

2. 点击编辑区域左侧的网络图标，在编辑区域出现一个梯形图的简单网络，左侧是一个常开触点，变量名是 C000，右侧是一个线圈，变量名是 C001，如图 5-5-2。



图 5-5-2

3. 双击常开触点 C000，出现触点/线圈属性对话框，在 I/O 地址 (S) 栏，输入 %IX0.0，表示 PLC 本机的第一个数字量输入通道，点击确定，如图 5-5-3。



图 5-5-3

4. 双击梯形图中线圈 C001，出现触点/线圈属性对话框，在 I/O 地址 (S) 栏，输入%QX0.0，表示 PLC 本机的第一个数字量输出通道，点击确定，如图 5-5-4。



图 5-5-4

当变量 C000 接通闭合是，C001 也线圈也闭合。
至此一个简单的 LD 梯形图就编写完了。

5.5.2 在 LD 中插入一个 FB

在使用 LD 编程时，有时会需要一个功能块，而编辑向导中没有，此时需要自定义一个功能块，下面说明如何在 LD 中插入一个小信号切除的功能块 XXH。

1. 首先插入一个名为 XXH、ST 语言编写的功能块，如图 5-5-5。



图 5-5-5

2. 和编辑程序一样编辑功能块，打开新建的功能块 XXH 后，在变量工作单中添加变量 IN 为 REAL，用法为 VAR_INPUT，变量 CQ 为 REAL，用法为 VAR_INPUT，，变量 OUT 为 REAL，用法为 VAR_OUTPUT，如图 5-5-6。

	名称	类型	用法	描述
	Default			
	CQ	REAL	VAR_INPUT	
	OUT	REAL	VAR_OUTPUT	
	IN	REAL	VAR_INPUT	

图 5-5-6

3. 然后在编辑区键入代码如图 5-5-7：

```
1 IF      IN < CQ
2 THEN
3     IN := 0.0;
4 ELSE
5     IN := OUT;
6 END_IF;
```

图 5-5-7

保存编辑之后的 XXH 功能块。

4. 打开一个 LD 编写的程序，如之前建立的 LD_PRO，打开编辑相当，选择组<MY_Proj>，可以看到自定义的功能块 XXH，如图 5-5-8。

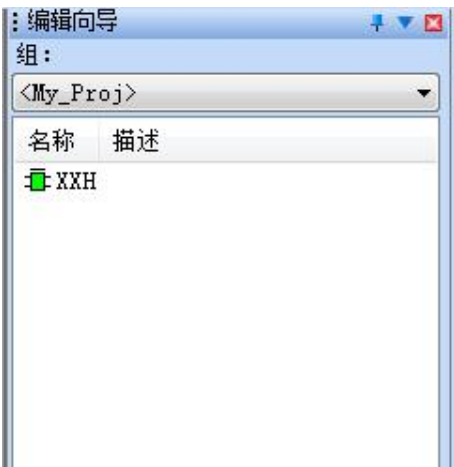


图 5-5-8

5. 把自定义的功能块 XXH 拖拽到编辑区，如图 5-5-9。

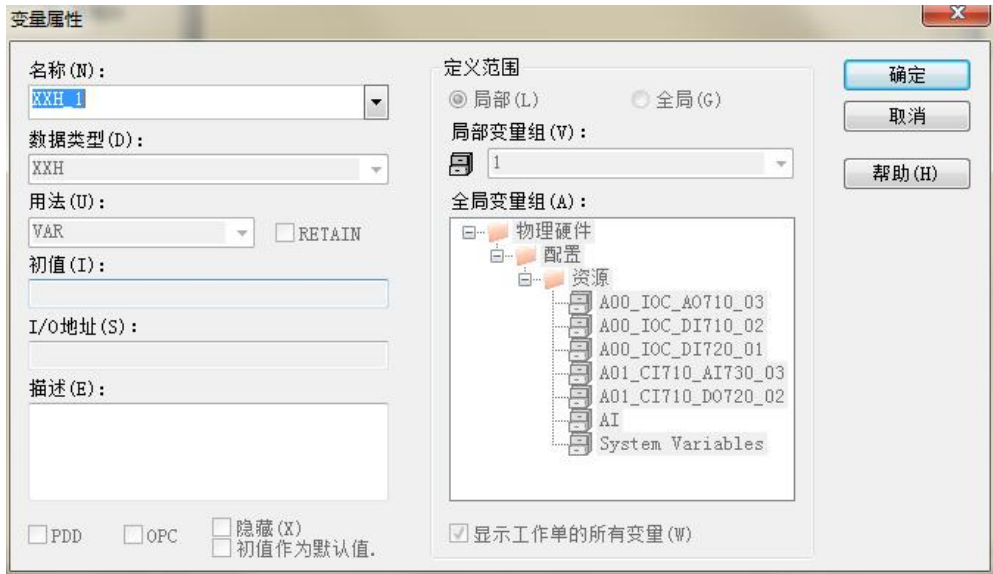


图 5-5-9

6. 在程序编辑区分别给插入的功能块定义变量，如图 5-5-10。

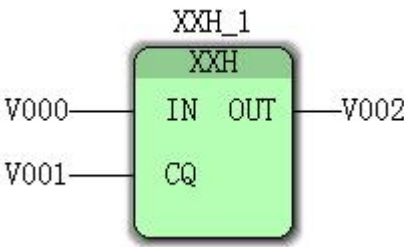


图 5-5-10

7. 点击工具栏上的制作按钮，在没有错误和警告的情况下，点击工

具栏上的工程控制对话框按钮，然后在工程控制对话框中点击下载。下载完成之后，打开工具栏上的调试开关。把变量 V000、V001、V002 分别添加到监视窗口，如图 5-5-11。

变量	值	类型	实例
V000	0.0000000	REAL	配置.资源.任务.LD_PRO.V000
V001	0.0000000	REAL	配置.资源.任务.LD_PRO.V001
V002	0.0000000	REAL	配置.资源.任务.LD_PRO.V002

图 5-5-11

在调试对话框可以赋值测试程序。

5.6 SFC 顺序功能图编程语言

图形语言 SFC 是由步和转换组成，这些步和转换是通过有方向的链接连接在一起的。可以将一个或多个动作块关联到一个 SFC 步。

动作块由一个动作和相应的动作限定符组成。动作限定符说明动作是如何受与其相关联的步的影响的。当 SFC 步变为活动状态时，所关联的动作会根据动作限定符而被执行。动作既可以是一个布尔变量，也可以是一个 IL/ST/LD/FBD 代码体(被称为'明细')。

转换代表了通过下一步继续处理的情况。如果一个转换变为 TRUE，则前一个步被再执行一次，并且后一个的步成为活动状态。转换既可以是一个布尔变量，也可以是使用 FBD 或 LD 编写的直接连接的布尔表达式。也可以在另一个被称为明细的代码本体中，编辑将要执行的代码。

所连接对象的集合被称为 SFC 网络。一个 SFC 网络必须具有一个初始步，当调用该 SFC POU 时，这个初始步是第一个要被执行的步。可以在 SFC 网络内插入并行分支(同步地执行)或选择分支(可选择地执行)。

5.6.1 创建一个 SFC 程序

创建一个红绿灯的控制程序为例，说明 SFC 编程语言的用法。

5.6.1.1 创建一个 SFC 网络

1. 插入一个工程，选 SCF 语言，程序名为 SFC_PRO。进入 SFC_PRO 的

编辑区，然后点击编辑区左侧的‘创建步转换序列’, 在编辑区就出现了一个 SCF 的‘步’ S001，如图 5-6-1。

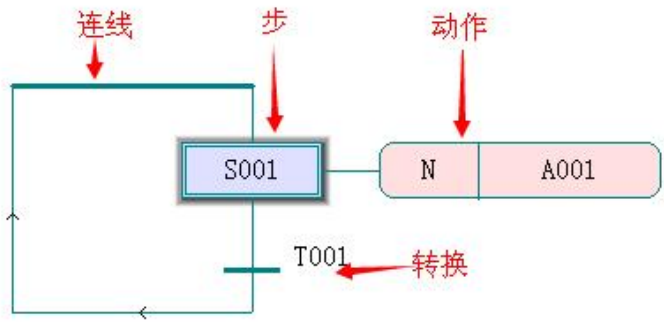


图 5-6-1

2. 上图中，每个‘步’、‘转换’和‘动作’都有唯一的名称。接着点击‘转换’ T001，在点击‘创建步转换序列’，则在这个步的下面增加了一个‘步’ S002，如图 5-6-2。

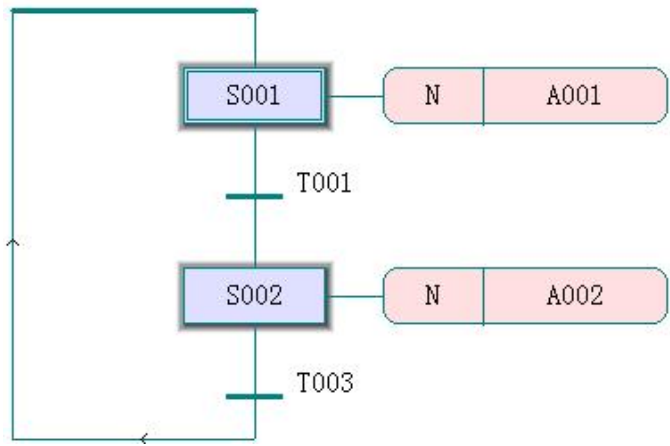


图 5-6-2

3. 点击‘步’ S002，再点击‘插入 SFC 分支’，则在‘步’ S002 的右侧插入了一个分支，分支带有一个‘步’ S003，如图 5-6-3。

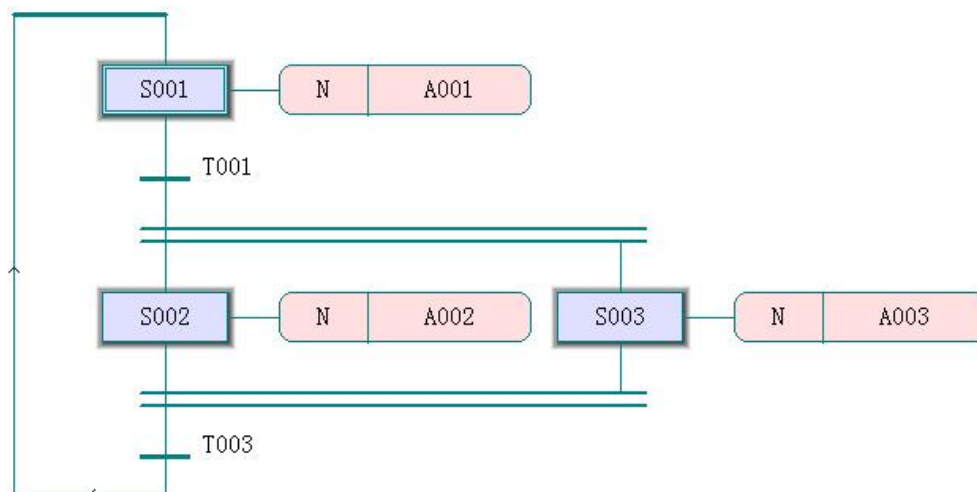


图 5-6-3

4. 点击‘步’ S003，然后点击‘创建步转换序列’，以此添加‘步’ S004、S005，如图 5-6-4。

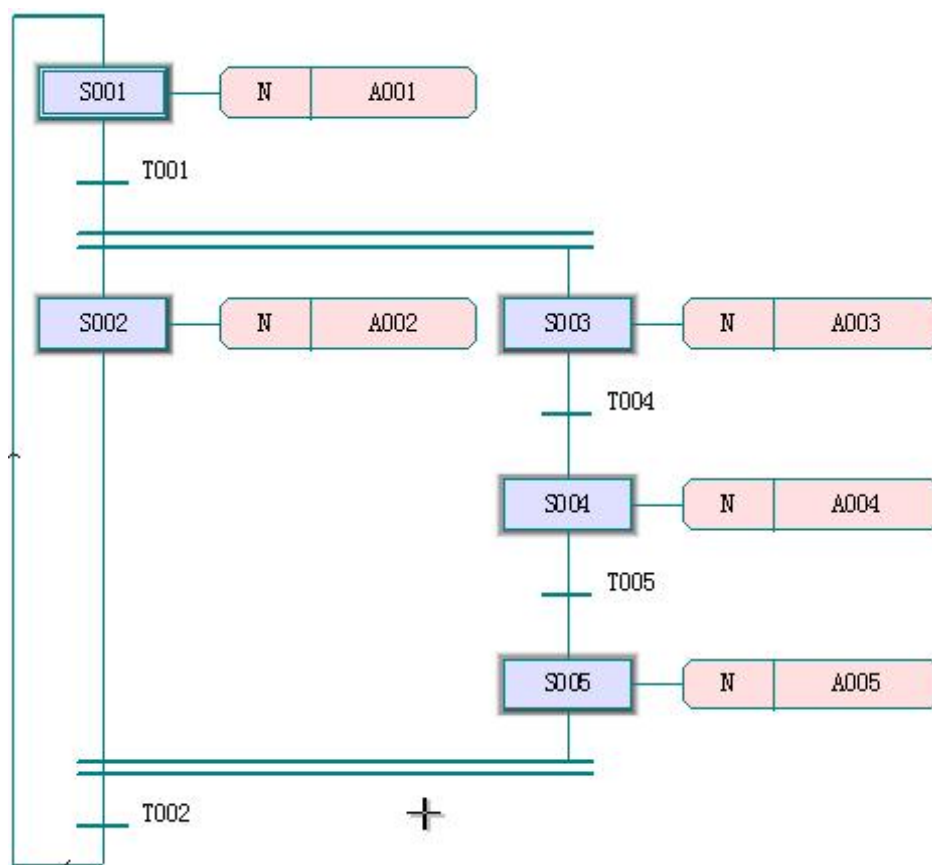


图 5-6-4

5. 然后点击‘转换’ T002, 点击‘插入步转换序列’, 插入一个步 S006, 如图 5-6-5。

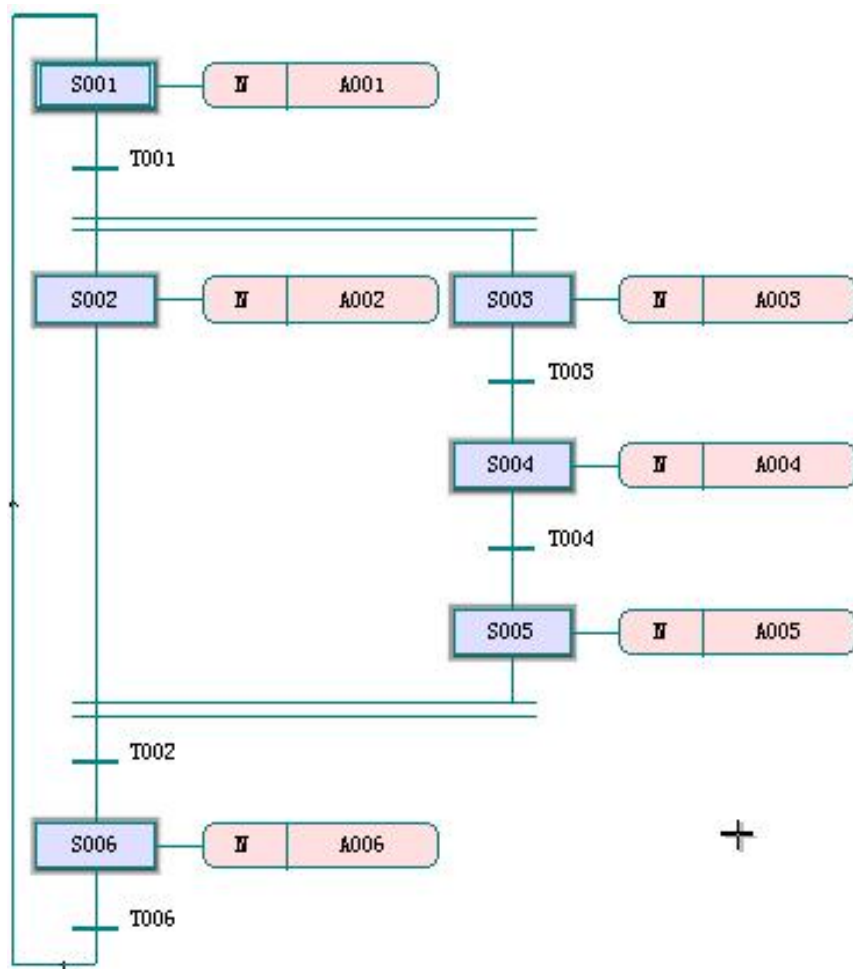


图 5-6-5

6. 然后点击‘步’ S006，插入分支‘步’ S007、S008、S009，如图 5-6-6。

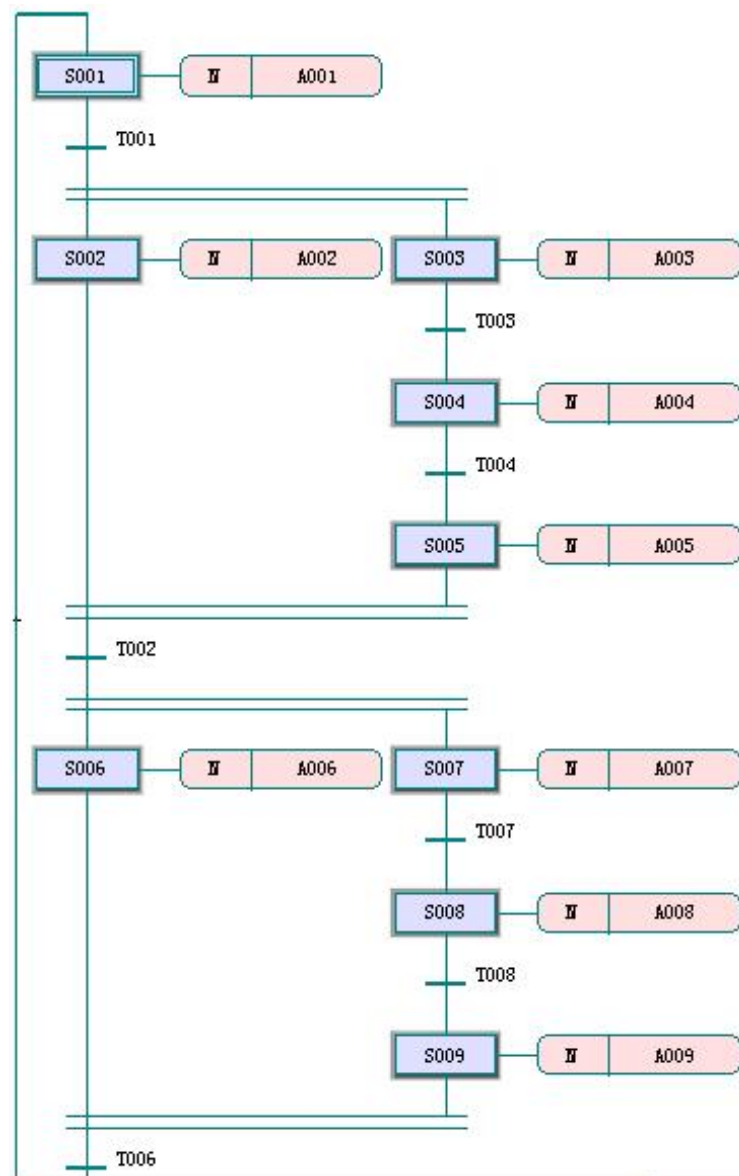


图 5-6-6

至此一个 SFC 网络建立完毕。

5.6.1.2 编写程序

➤ 转换条件

SFC 步的状态可以是活动的，也可以是不活动的。活动意味着该步实际上正被 PLC 处理。关联到该步上的动作被执行。如果一个步变为不活动的，则再执行一次该动作。一个没有相关动作块的步则等待后继的条件变为真。

SFC 步的状态由变量‘步骤名称.x’表示。如果该步是活动的，则‘步骤名字.x’为 TRUE。这个变量也称为步骤标志。

1. 双击‘转换’ T001，在弹出的对话框中选择 LD 语言，如图 5-6-7，然后点击确定，进入 T001 的编辑状态。



图 5-6-7

2. 在‘T001’转换中，编辑程序如图 5-6-8。S001 是起始步，当全局变量 C000 为 1 的时候，T001 为 1，进入步 S002、S003。



图 5-6-8

3. 打开 T003，编写程序如图 5-6-9，当‘步’ S003 开始执行 8S 之后，开

始执行‘步’ S004。

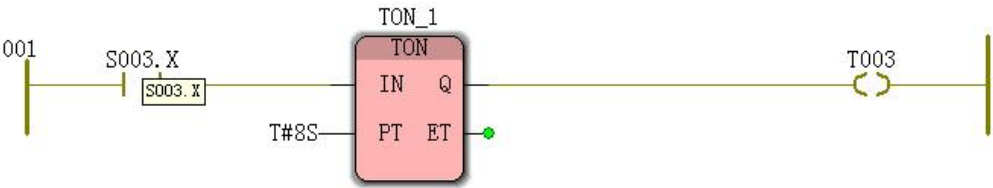


图 5-6-9

4. 打开 T004，编写程序如图 5-6-10，当‘步’ S004 开始执行 3S 之后，转换 T004 置 1，进入‘步’ S005。

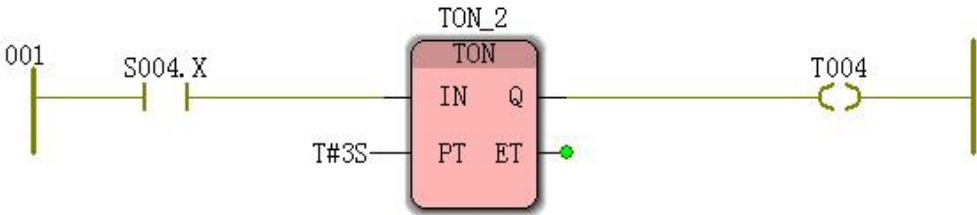


图 5-6-10

5. 双击打开 T002，编写程序如图 5-6-11，当‘步’ S005 执行 2 秒之后，开始执行‘步’ S006、S007。

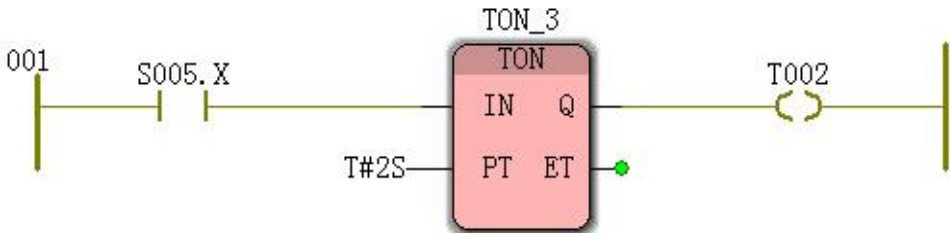


图 5-6-11

6. 打开 T006，编写程序如图 5-6-12，当‘步’ S007 开始执行 8S 之后，转换 T006 置 1，进入‘步’ S008。

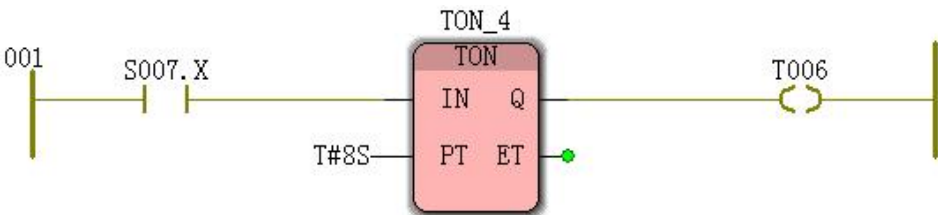


图 5-6-12

7. 打开 T007，编写程序如图 5-6-13，当‘步’ S008 开始执行 3S 之后，转换 T007 置 1，进入‘步’ S009。

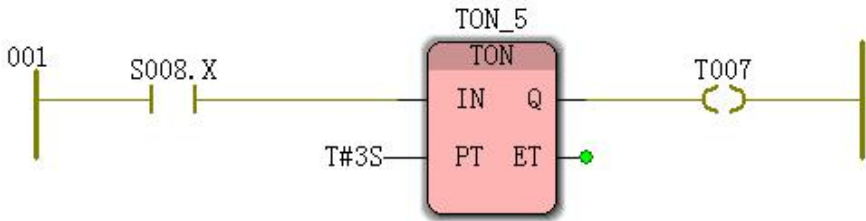


图 5-6-13

8. 打开 T005，编写程序如图 5-6-14，当‘步’ S009 开始执行 2S 之后，转换 T005 置 1。

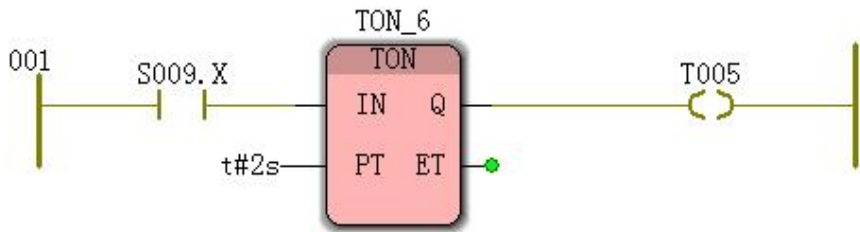


图 5-6-14

至此，SCF 编程的转换条件编程完毕。

➤ 动作：

1. 删除‘动作’ A001。
2. 将双击动作‘A002’名称更改为 NS_RED,(*南北红灯*)，选择动作类型为明细，限定符为 N，如图 5-6-15。

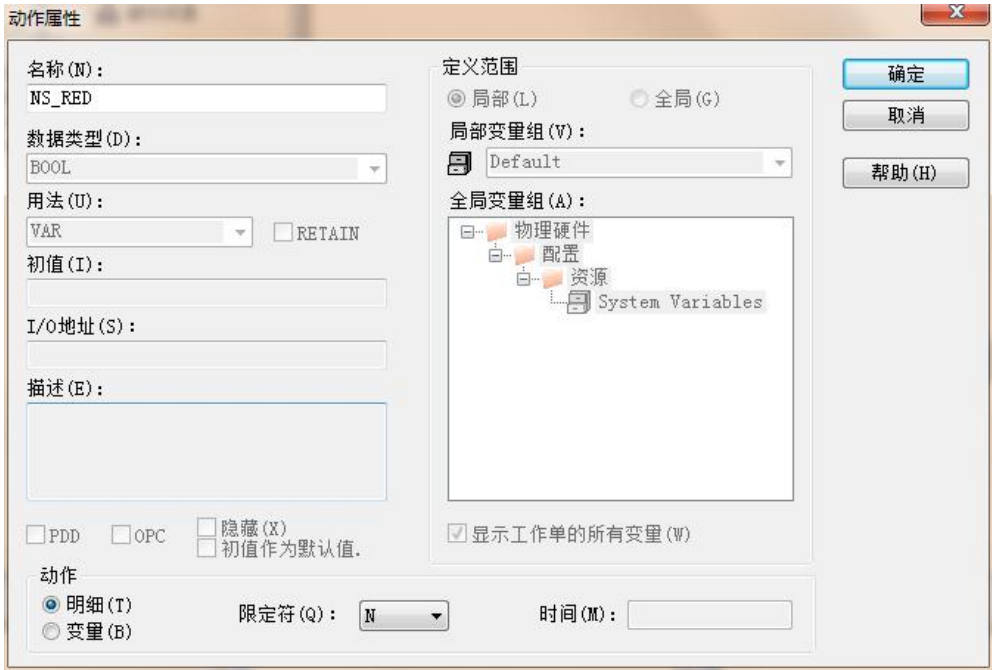


图 5-6-15

3. 将双击动作 ‘A003’ 名称更改为 EW_GREEN,(*东西绿灯*), 选择动作类型为明细, 限定符为 N;
4. 将双击动作 ‘A004’ 名称更改为 EW_GREEN_F,(*东西绿灯闪*), 选择动作类型为明细, 限定符为 N;
5. 将双击动作 ‘A005’ 名称更改为 EW_YELLOW,(*南北黄灯*), 选择动作类型为明细, 限定符为 N;
6. 将双击动作 ‘A006’ 名称更改为 EW_RED,(*东西红灯*), 选择动作类型为明细, 限定符为 N;
7. 将双击动作 ‘A007’ 名称更改为 NS_GREEN,(*南北绿灯*), 选择动作类型为明细, 限定符为 N;
8. 将双击动作 ‘A008’ 名称更改为 NS_GREEN_F,(*南北绿灯闪*), 选择动作类型为明细, 限定符为 N;
9. 将双击动作 ‘A009’ 名称更改为 NS_YELLOW,(*南北黄灯*), 选择动作类型为明细, 限定符为 N。
10. 双击动作 ‘NS_RED’, 选择语言为 LD, 点击确定, 进入动作编辑界面。



图 5-6-16

在编辑区编辑程序如下：



图 5-6-17

其中，变量 NS_R 的地址为%QX6000.0，表示步 S002 激活时，NS_R

南边方向亮红灯。

11. 双击动作 ‘EW_GREEN’，选择语言为 LD，点击确定，进入动作编辑界面，在编辑区编辑程序如下：



图 5-6-18

其中，变量 EW_G 的地址为%QX6000.5，表示步 S003 激活时，EW_G 东西方向的绿灯亮。

12. 双击动作 ‘EW_GREEN_F’，选择语言为 LD，点击确定，进入动作编辑界面，在编辑区编辑程序如下：

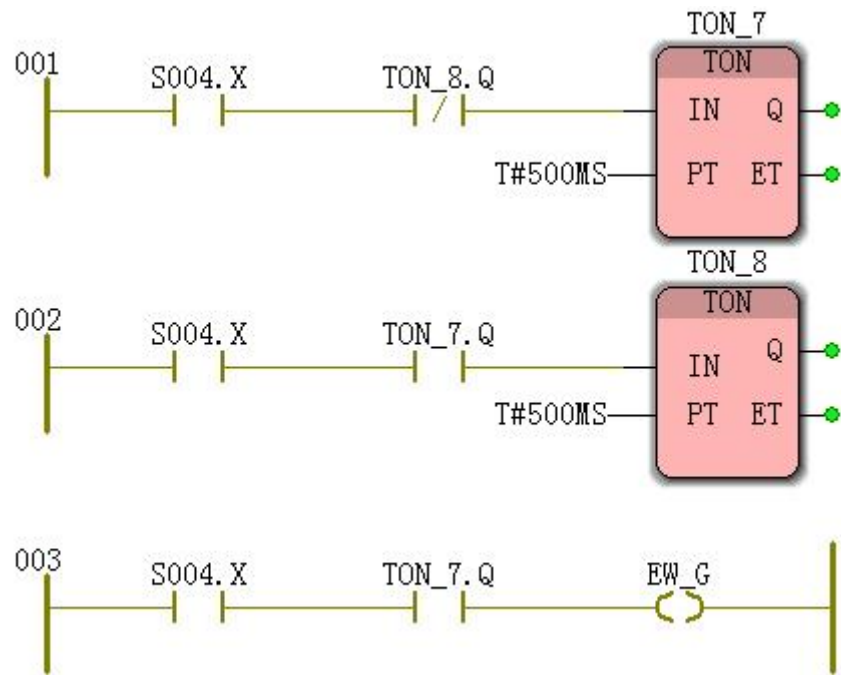


图 5-6-19

表示当 ‘步’ S004 激活时，EW_G 东西方向的绿灯闪。

13. 双击动作 ‘EW_YELLOW’，选择语言为 LD，点击确定，进入动作编辑界面，在编辑区编辑程序如下：



图 5-6-20

其中，线圈 EW_Y 地址为%QX6000.6，当‘步’ S005 激活时，EW_Y 东西方向的黄灯亮。

14. 双击动作 ‘EW_RED’，选择语言为 LD，点击确定，进入动作编辑界面，在编辑区编辑程序如下：



图 5-6-21

其中，线圈 EW_R 地址为%QX6000.4，当‘步’ S006 激活时，EW_R 东西方向的红灯亮。

15. 双击动作 ‘EW_GREEN’，选择语言为 LD，点击确定，进入动作编辑界面，在编辑区编辑程序如下：



图 5-6-22

其中，线圈 NS_G 地址为%QX6000.1，当‘步’ S007 激活时，NS_G 南边方向的绿灯亮。

16. 双击动作 ‘EW_GREEN_F’，选择语言为 LD，点击确定，进入动作编辑界面，在编辑区编辑程序如下：

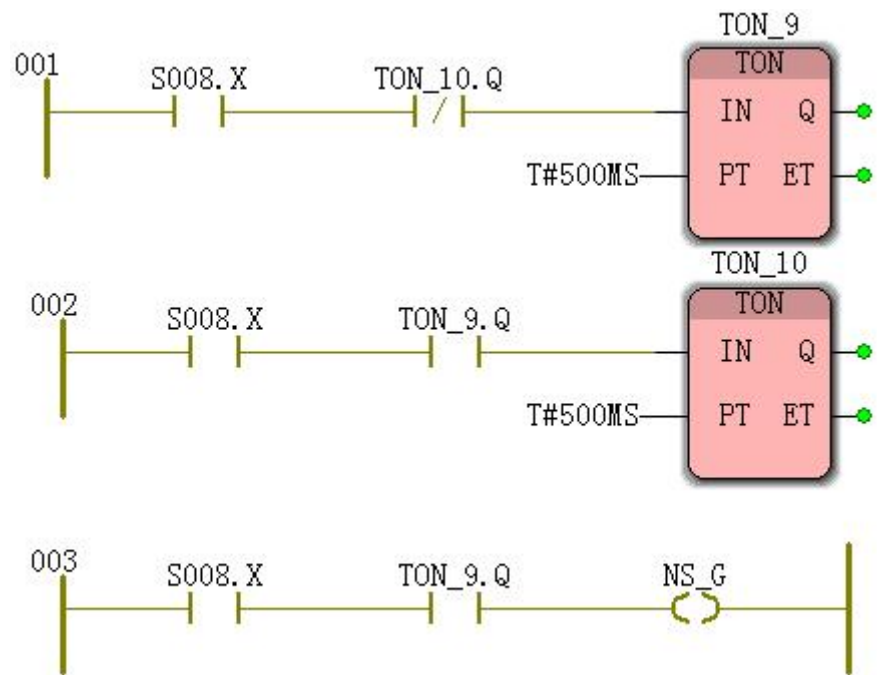


图 5-6-23

其中，线圈 NS_G 地址为%QX6000.1，当‘步’S008 激活时，NS_G 南边方向的绿灯闪亮。

17. 双击动作 ‘NS_YELLOW’，选择语言为 LD，点击确定，进入动作编辑界面，在编辑区编辑程序如下：



图 5-6-24

其中，线圈 NS_Y 地址为%QX6000.2，当‘步’S009 激活时，NS_Y 南边方向的黄灯亮。

至此，一个完整的红绿灯程序就编写完了。

➤ 调试程序

点击工具栏上的制作按钮，在没有错误和警告的情况下，点击工具栏上的工程控制对话框按钮，然后在工程控制对话框中点击下载。

下载完成之后，打开工具栏上的调试开关。可以监视 SCF 程序。

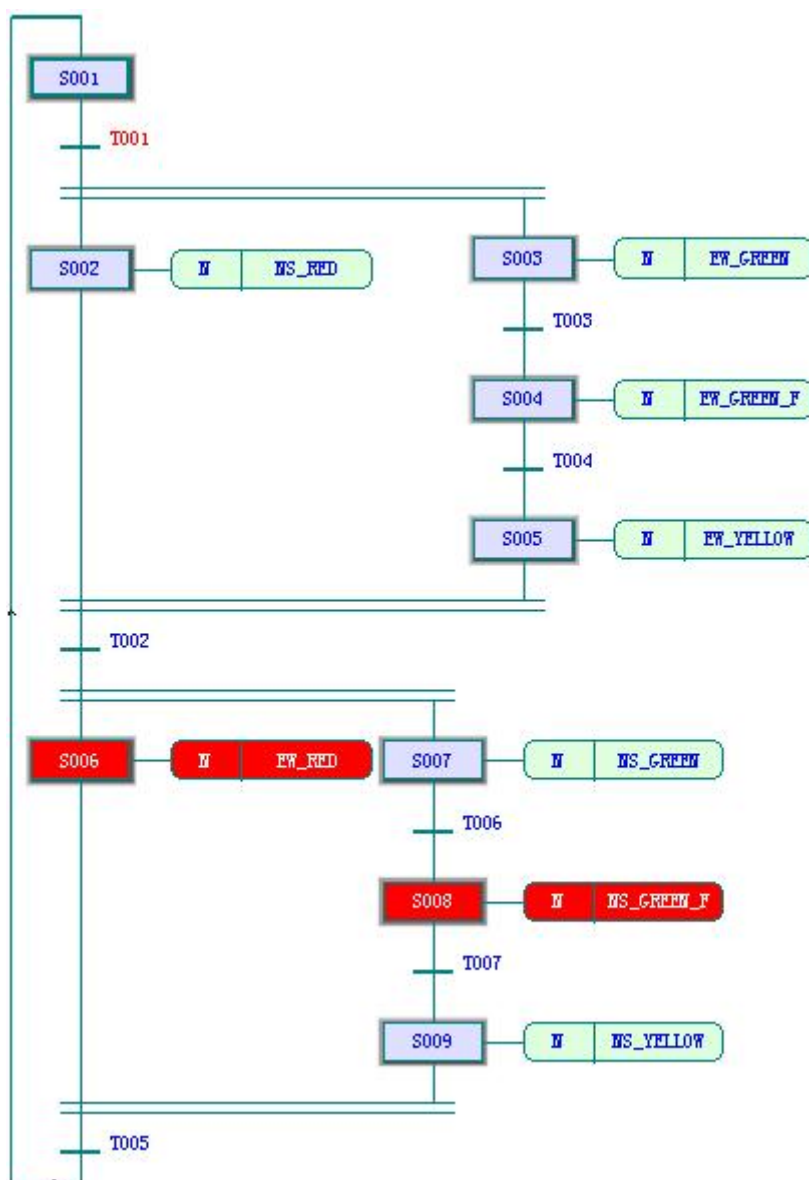


图 5-6-25

也可以打开全局数据库，查看信号灯变量的状态。

5.6.2 SFC 的动作限定符

限定符	描述	功能
N	不保存的	只要步是活动的，就执行动作代码本体或置位布尔变量。
R	超越复位	不再执行动作代码本体或者复位布尔变量。在使用‘S’限定符之前，必须要置位动作。
S	置位（保存的）	执行动作代码本体或者置位布尔变量。一旦(相关联的)步变为活动状态，就保存这个(置位)状态。该置位状态，仅能通过使用‘R’限定符，将该动作关联到另一个不同的步上，来显式地复位。
L	限时的	只要步是活动的，就执行动作代码本体或置位布尔变量，但最多能持续所设定的时间间隔这样一段时间。
D	延时的	在所设定的延迟时间流逝之后，再执行动作代码本体或置位布尔变量。只要步是活动的，动作就保持活动状态。如果步处于活动状态的时间比所设定的延迟时间短，则动作不会变为活动状态。
P	脉冲	步一变为活动状态，动作代码本体就执行一个操作周期，或布尔变量就置位一个操作周期。
SD	保存并延时	在步被激活之后，当又流逝了所设定的延迟时间之时，就执行动作代码本体或保存并置位布尔变量，即使步又变为不活动状态。该动作在被复位之前，将一直保持活动状态。如果步处于活动状态的时间比所设定的延迟时间还短，则该动作无论如何也会变为活动状态的。
DS	延迟并保存	在步被激活之后，当又流逝了所设定的延迟时间之时，就执行动作代码本体或保存并置位布尔变量。该动作在被复位之前，将一直保持活动状态。如果步处于活动状态的时间比所设定的延迟时间短，则动作不会变为活动状态。
SL	保存并限时	只要步是活动的，就在一个固定时间间隔内，执行动作代码本体或置位并保存布尔变量。如果步处于活动状态的时间比该时间间隔短，则该动作也无论如何会在整个时间间隔内处于活动状态。如果在该时间间隔内，动作被复位，该动作马上就会变为不活动状态。

5.7 各类 POU 的调用

POU 的调用要遵循以下原则：

- ✧ 程序可以调用功能、功能块，程序通过任务配置来调用。
- ✧ 功能块可以调用功能和其他功能块。
- ✧ 功能可以调用功能。
- ✧ POU 可以调用其他的 POU，但不允许递归调用。任何一个 POU 只有经过触发才能够开始运算。

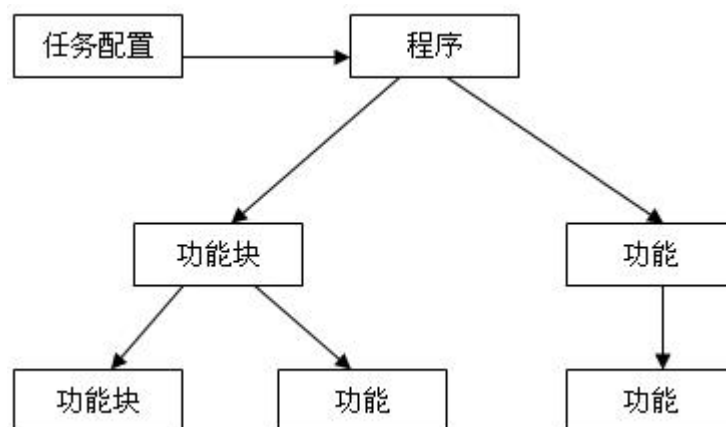


图 5-7-1

5.8 编译工程

在编辑了工程之后，您必须编译它。编译意味着将工作单中的内容翻译并转换为可以被您的 PLC 执行的机器专用代码。编译过程由这样几个步骤完成：

1. 编译（即语法检查）不同工作单。
2. 所编译的工作单被连接在一起，并且产生 IEC 中间代码。
3. 产生必须要下装到 PLC 中的 PLC 代码。

当编译时，消息窗口显示编译过程。任何检测到的错误和警告（例如，语法错误、存储器或者文件问题）以及其它信息也被显示在适当的消息窗口页面中。你可以利用消息窗口通过双击错误消息来访问可疑的代码本体工作单。

5.8.1 编译器命令

当完成编辑时，有几种可能来编译工程。在‘建立’子菜单中提供了相应的菜单项，也可以在工具栏上选择。

- ‘制作’ - 当完成编辑时，这是编译工程的标准模式。菜单项可被用于编译所编辑的所有工作单。在工程树中，用星号标记这些工作单。在使用‘制造’之后，产生 PLC 特定代码并且工程准备下载到 PLC。
- 修补 POU’ - 这个命令用于仅仅编译所更改的内容，例如调试了工程之后可能会做一些修改。当 PLC 保持运行状态，即不中止 PLC 上的程序的执行的情况下，更改内容被自动下装到 PLC。所以在将相应的工作单切换到联机模式之后，你能立即观察所做的更改。在修补一个 POU 时，不必停止 PLC 上的程序执行。如果您的 PLC 支持，您也可以使用‘下装更改部分’特征，将程序中修改的部分发送并应用到目标系统上。
- ‘编译工作单’ - 此菜单项用于编译一个单一的编辑完的工作单。选择这个菜单项意味着，当前代码本体工作单及相关的变量工作单内的语法错误将被编译器检查。所有检查出来的错误和警告都会显示在消息窗口。通过双击错误或警告能打开相关的工作单，在这里检查到错误。使用‘编译工作单’不产生代码。

当关闭或保存一个工作单时，系统会自动编译这个工作单。

另外，变量工作单也会被编译。

由于这个自动编译功能，每个用户自定义功能或功能块，都会在保存了相应工作单之后，马上可以在编辑向导中使用。

- ‘重建工程’ - 此菜单项用于第一次编译整个工程或者在用户库被更改的情况下。它应该仅仅被用于这样几种情况：‘制作’产生了编译错误、你已经对没有前端代码的工程进行了解压缩、或者在一个已被声明的用户库中做了改动。

使用‘重建工程’命令，将编译和连接所有工作单。检查出的错误和警告显示在消息窗口。在通过语法检查之后，会自动产生 IEC 代码和专用于具体 PLC 的代码。这样，工程就具备了下载到 PLC 上的条件。

5.8.2 “制作” 命令

点击工具栏图标 ‘制作’  或者<F9>快捷键，或选择菜单项 ‘建立 > 制作’ 。

编译过程被显示在消息窗口的 ‘建立’ 页面内。编译过程中所检测出的错误和警告信息被记录在消息窗口的相应页面内。

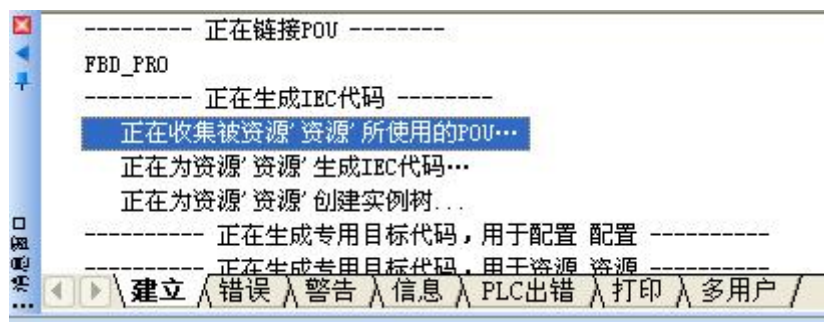


图 5-8-1

使用 ‘制作’ 菜单项/图标或者<F9>快捷键，已更改的工作单会被编译、连接，并将产生更改后的 PLC 代码。在成功地执行完这几步之后，所修改的工程就具备了下装到 PLC 的条件

5.9 工程仿真

如果不连接 PLC，也可以通过仿真控制器进行仿真调试程序。

1. 新建一个工程，在工程向导第 4 步中选择资源类型为：eCLR_Simulation，如图，直到完成。

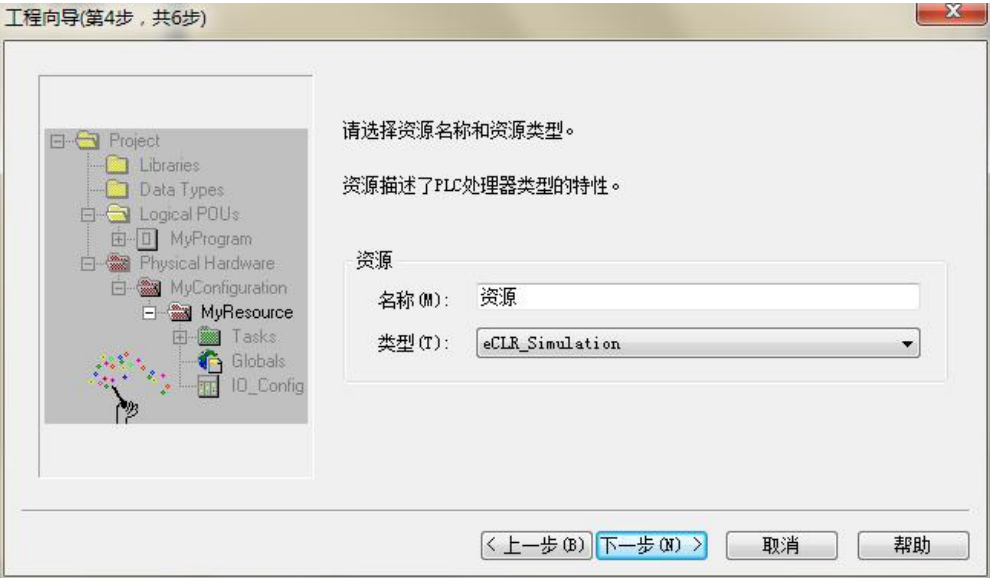


图 5-9-1

也可以在工程树-硬件选择卡-资源属性中修改控制器类型为：eCLR_Simulation，如图 5-9-2。



图 5-9-2

2. 选择处理器类型后，选择工程树-硬件选择卡-资源-右键选择设置，设置版本：eCLR_Simulation（Core: 3.0.0），通讯类型选择为：仿真 1，如图 5-9-3。

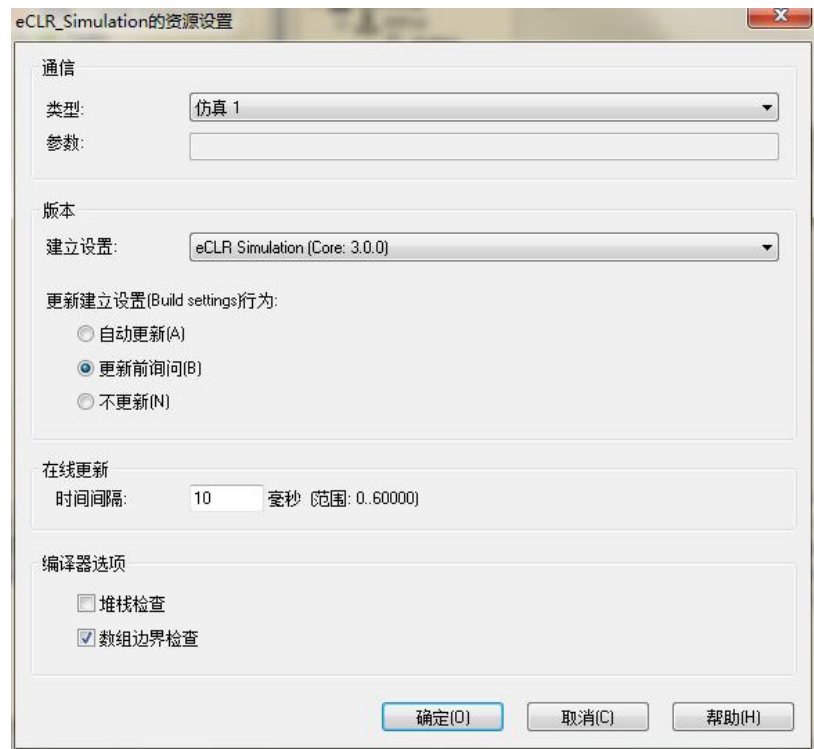


图 5-9-3

3. 点击制作按钮 ，如果出现：0 错误，0 警告时，可以点击工程控制按钮 ，激活 EasySim eCLR 1 。
4. 点击工程控制对话框的下载、冷启，就可以对程序进行仿真测试了。
5. 右键点击任务栏的 EasySim eCLR 1 ，选择 显示 I/O（I），打开仿真控制软件。如果硬件中包括了 IO 模块，可以对 IO 模块进行模拟操作。详细操作参见 PLC 帮助文档（EasySim 控制器仿真程序）。

第 6 章 OPC 通讯

Truhigh P600 系列 PLC 系统提供了 OPC 数据接口，可以与多种上位机组态软件进行通讯，方便工艺参数监控和设备操作。

本章分为一下四节：

- 6.1 OPC 服务器设置
- 6.2 测试 OPC 服务器
- 6.3 上位机通过 OPC 连接 Truhigh P600 系统
- 6.4 IFIX 画面组态

6.1 OPC 服务器设置

1. 全局变量数据库中的变量，选择有 OPC 属性的变量，都可以通过 OPC 与上位机通讯，如图 6-1-1。

	名称	类型	用法	描述	地址	初值	保持	PDD	OPC	隐藏	初...	默
1	System Variables											
25	NewGroup											
26	C000	BOOL	VAR_GL...				☐	☐	☒	☐	☐	
27	NS_R	BOOL	VAR_GL...	南北红灯	%QX6000.0		☐	☐	☒	☐	☐	
28	NS_G	BOOL	VAR_GL...	南北绿灯	%QX6000.1		☐	☐	☒	☐	☐	
29	NS_Y	BOOL	VAR_GL...	南北黄灯	%QX6000.2		☐	☐	☒	☐	☐	
30	EW_R	BOOL	VAR_GL...	东西红灯	%QX6000.3		☐	☐	☒	☐	☐	
31	EW_G	BOOL	VAR_GL...	东西绿灯	%QX6000.4		☐	☐	☒	☐	☐	
32	EW_Y	BOOL	VAR_GL...	东西黄灯	%QX6000.5		☐	☐	☒	☐	☐	

图 6-1-1

2. 在电脑开始菜单栏里面选择 TruHighP600-> OPC Configurator，如图 6-1-2。

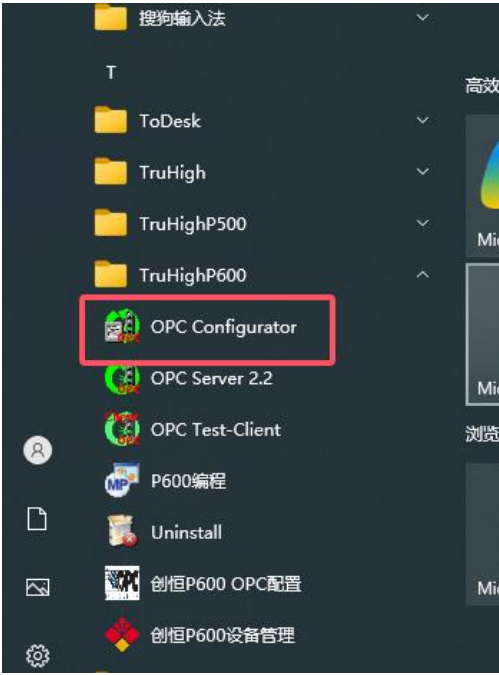


图 6-1-2

3. 打开 OPC Configurator 窗口，在 OpcProject 上右击选择“New Resource”，新建数据源，如图 6-1-3。

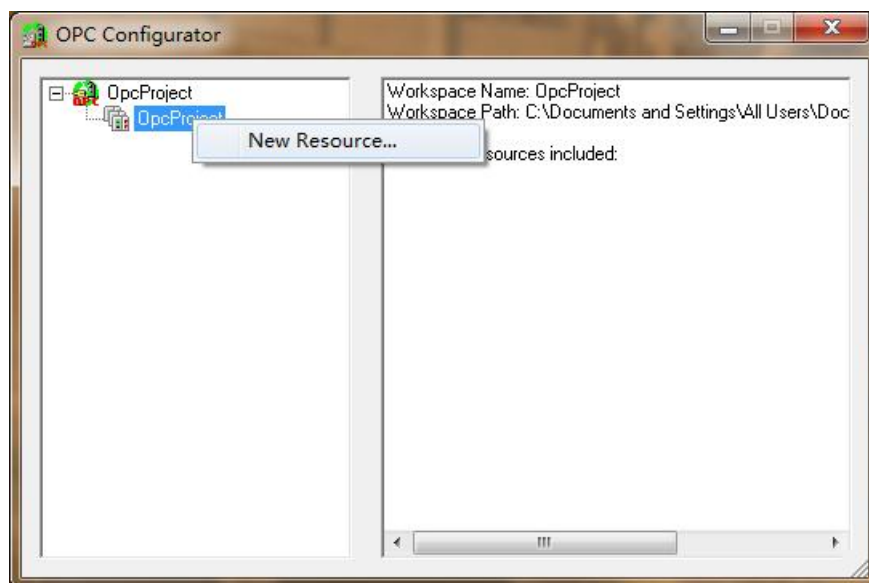


图 6-1-3

4. 在点击新建的数据源“NewResource”，在“Resource type”中选择 eCLR。

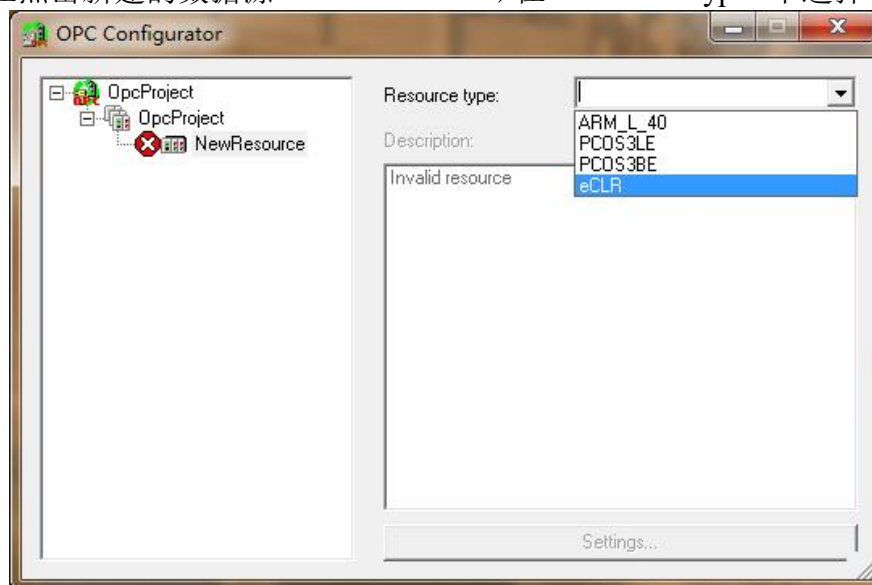


图 6-1-4

5. 这里可以对数据源重命名，例如命名为 PU620。

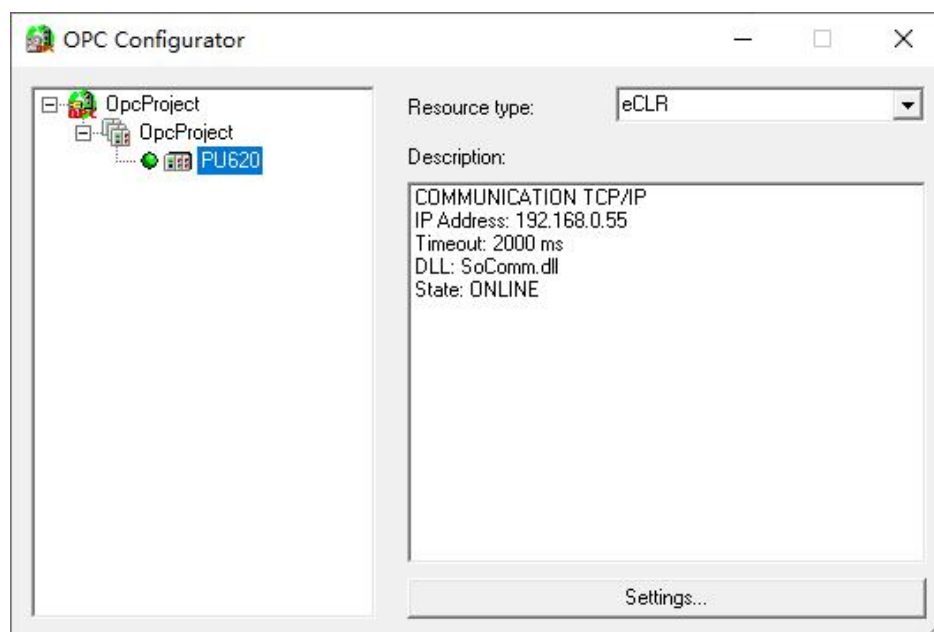


图 6-1-5

6. 点击“Setting...”按钮，打开 OPC 设置，如图 6-1-6，“Communication path”选择为“TCP/IP”这里的 IP 地址是指 PLC 的地址，超时时间 Timeout 默认 2000ms。

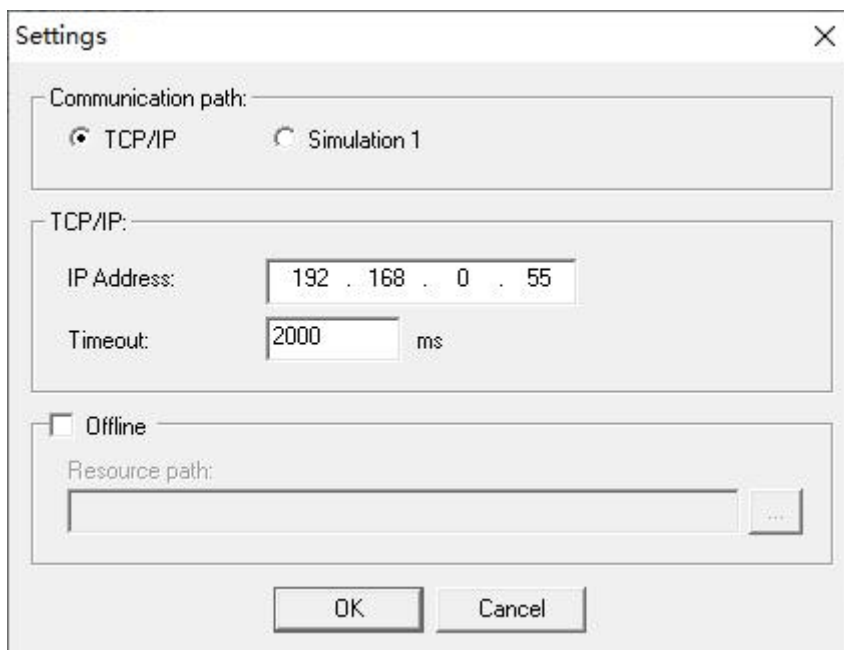


图 6-1-6

设置完毕后点击 OK，然后关闭 OPC Configurator。

6.2 测试 OPC 服务器

1. 在开始菜单选择 TruHighP600-> OPC Test-Client，如图 6-2-1，打开 OPC 测试客户端，界面如图 6-2-2。

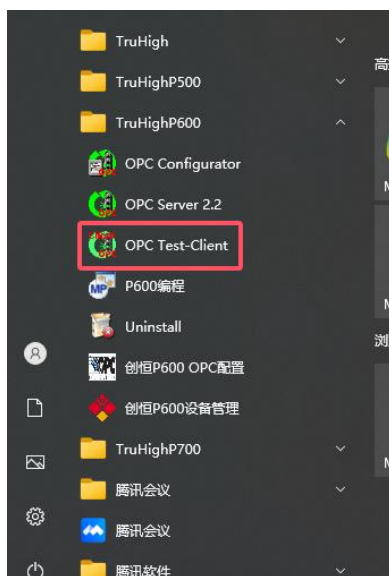


图 6-2-1

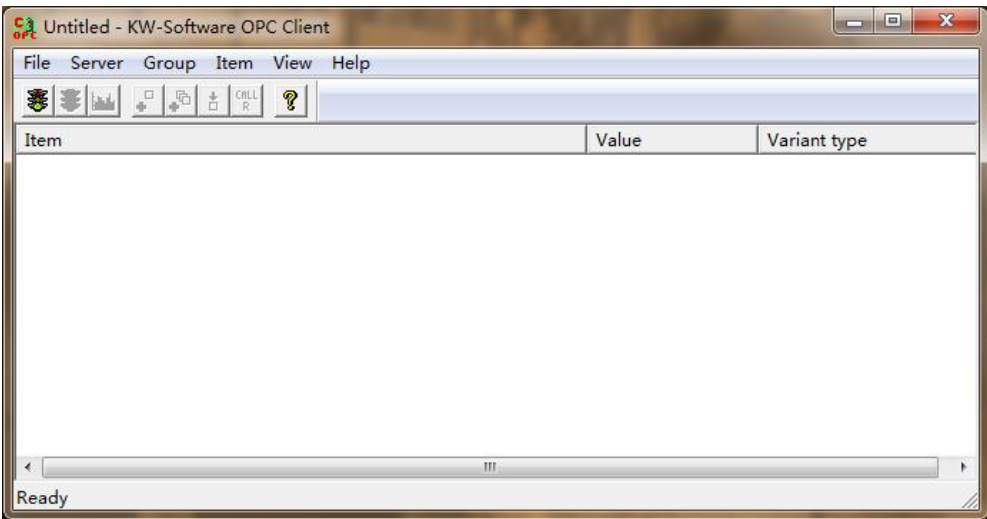


图 6-2-2

2. 点击工具栏上的“connect to server”图标，连接 OPC 服务器。



图 6-2-3

然后工具栏变为：



图 6-2-4

3. 点击 Add all items” 按钮，添加所以 OPC 数据。



图 6-2-5

4. 这样就可以把用户在 MULTIPROG 数据库中设置为 OPC 变量的数据都读出来，如图 6-2-6。

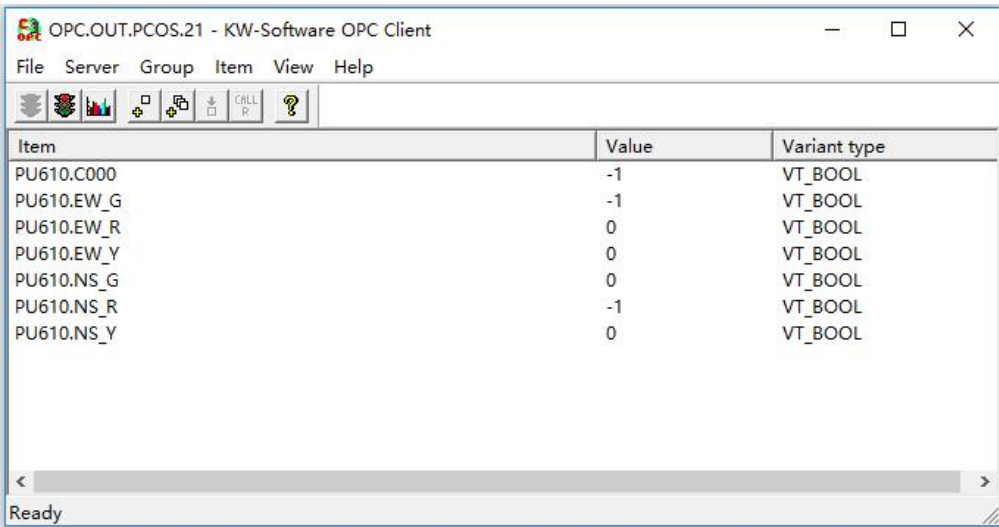


图 6-2-6

- 对于可写的变量，我们可以通过工具栏上的“Write item”按钮进行修改。

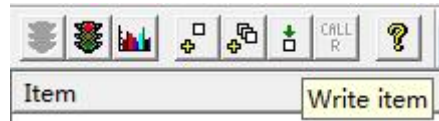


图 6-2-7

我们给变量变量置 1，通过监视程序可以查看修改的变量。

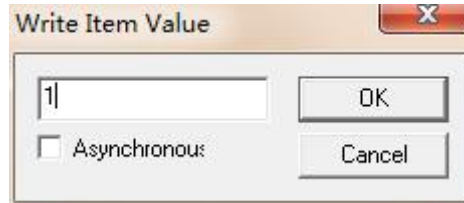


图 6-2-8

此时，Truhigh P600 系统的 OPC 服务器配置完成了。

6.3 上位机通过 OPC 连接 Truhigh P600 系统

以 IFIX5.5 上位机为例，说明上位机通过 OPC 连接 Truhigh P600 系统。

CPU 运行程序为红绿灯控制，程序见 5.6 章节 《SFC 顺序功能图编程语言》。

6.3.1 IFIX 5.5 的 OPC 设置

- 首先运行 IFIX 5.5，然后在工程树中双击打开系统配置-系统配置应用，如图 6-3-1，主界面如图 6-3-2。



图 6-3-1



图 6-3-2

2. 点击菜单栏-配置-SCADA 配置，在“I/O 驱动器名称”中选择可用的驱动器“OPC-OPC Client v7.42a”，然后点击确定，再在 SCADA 配置对话框中点击添加，然后点击确定关闭 SCADA 配置对话框，如图 6-3-3。



图 6-3-3

3. 然后双击驱动器 OPC 图标，如图 6-3-4。



图 6-3-4

4. 在弹出的“/I/O Driver Server Connection”对话框中选择“Use Local OPC Client Driver”然后点击“Connect”按钮，如图 6-3-5。

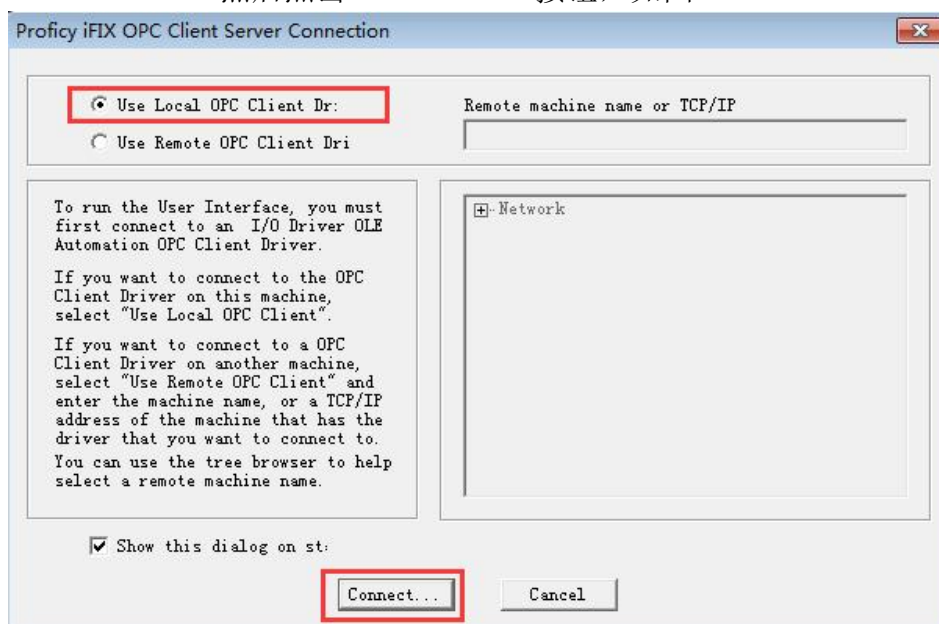


图 6-3-5

5. 然后出 OPC 的配置窗口，如图 6-3-6。

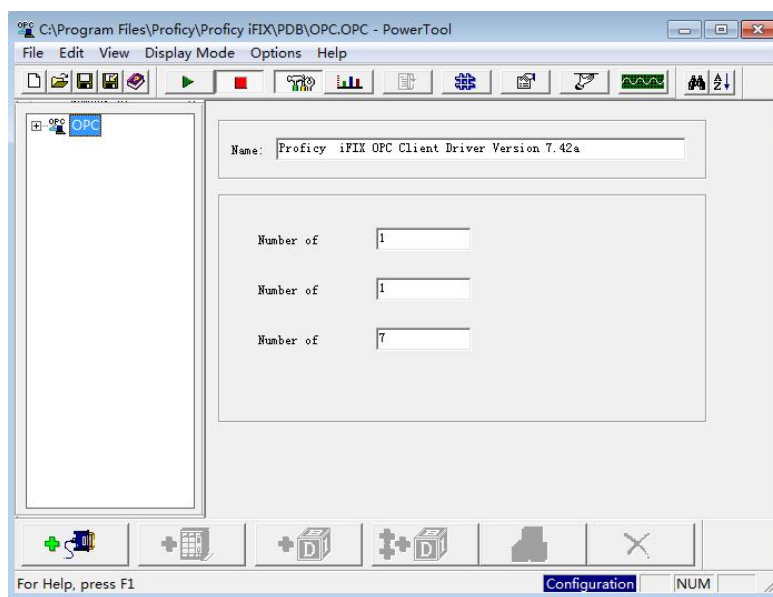


图 6-3-6

6. 然后点击“add OPC server”按钮，在出现对对话框中选择“Use Local OPC Server”，然后点击“Next”，进入下一步。

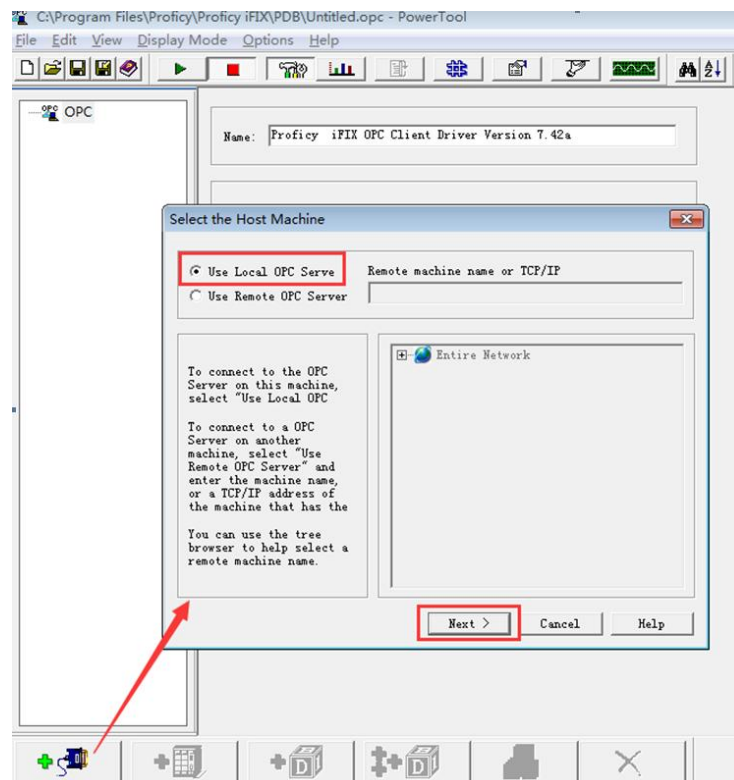


图 6-3-7

7. 进入 Select an OPC Server 窗口，选择 “OPC.OUT.PCOS.21”，然后单击 “Finish”。

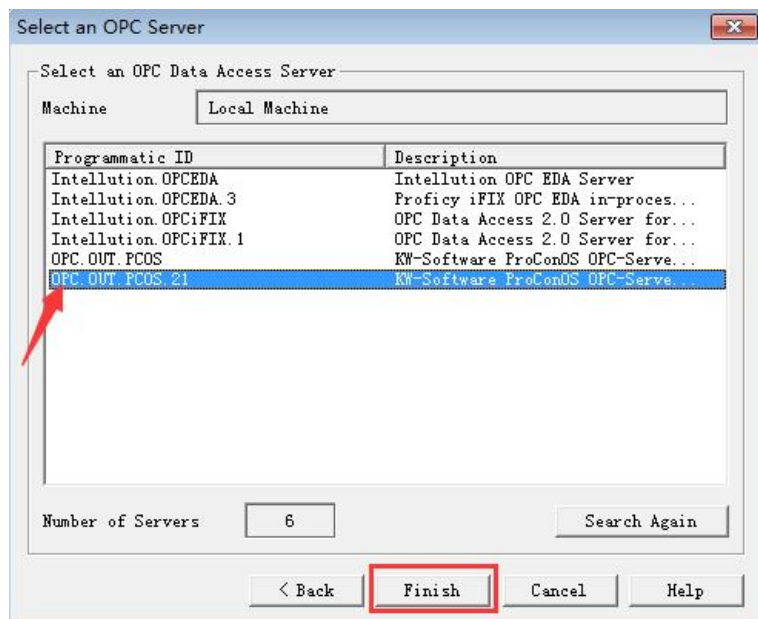


图 6-3-7

8. 在 OPC 配置窗口出现了 Server1，然后单击 “add Group” 按钮。

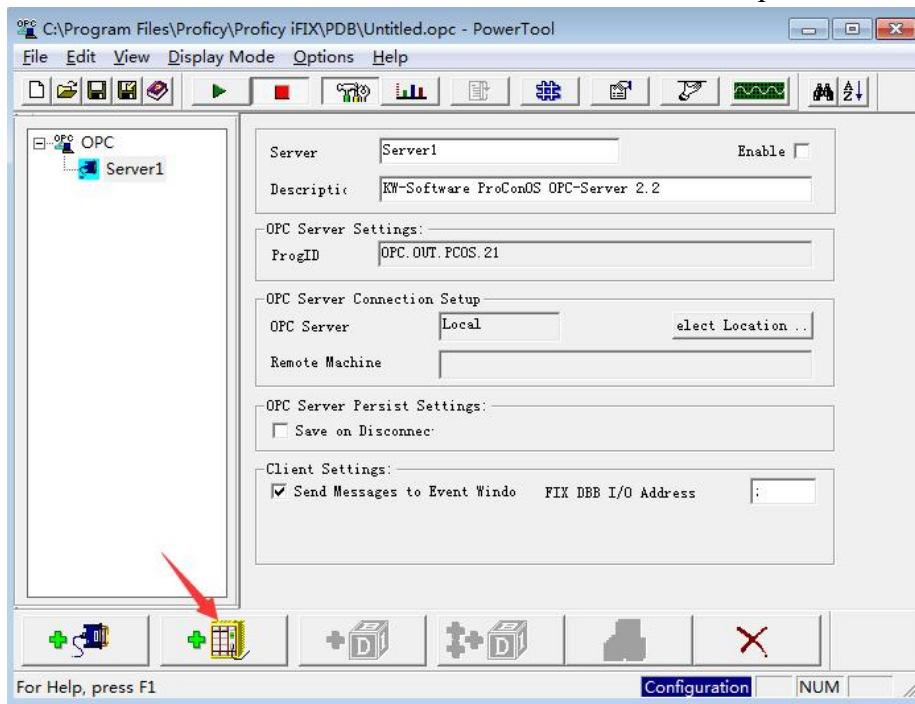


图 6-3-8

9. 然后点击“add Multiple Items”按钮。

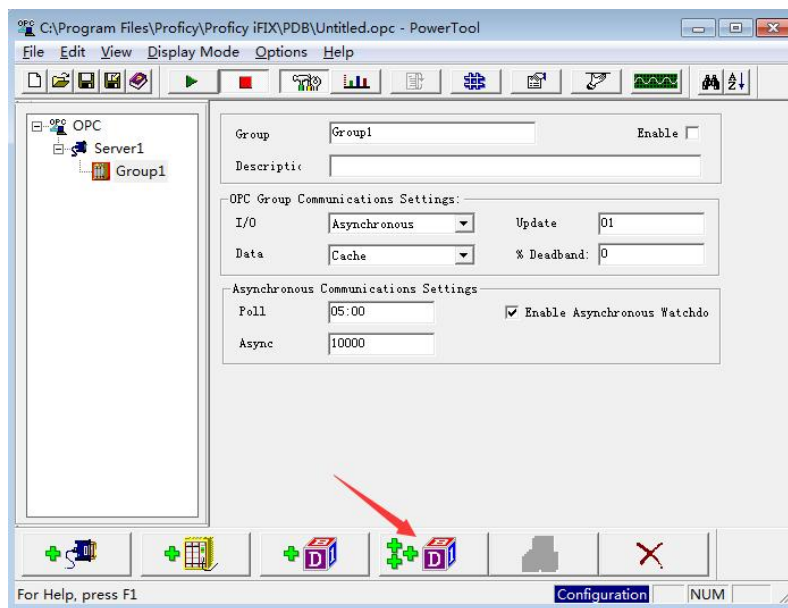


图 6-3-9

10. 弹出询问是否连接 OPC Server 对话框，点击“是”。



图 6-3-10

11. 然后在弹出的界面可以看到 OPC 服务器里的变量，然后把变量全选后点击 OK，添加到 OPC 配置当中，如图 6-3-11。

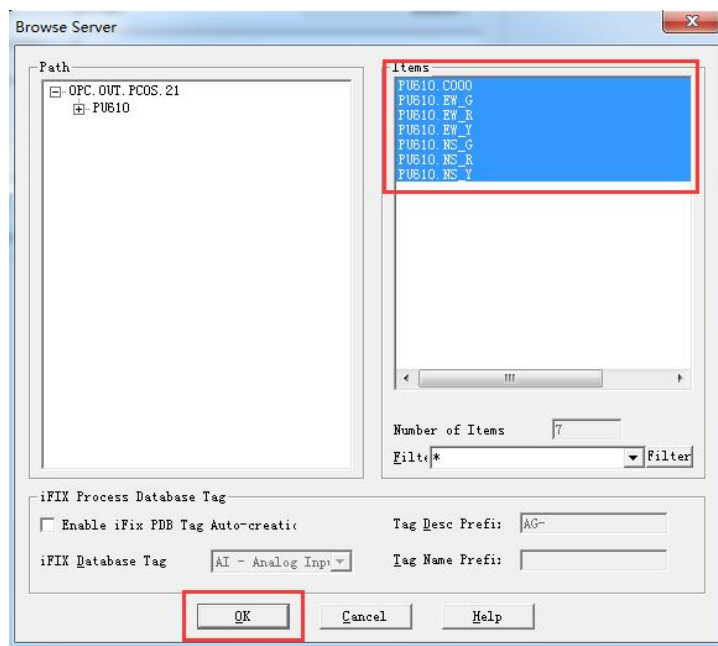


图 6-3-11

12. 在配置中每个选项都把“Enable”勾选上，如图 6-3-12。

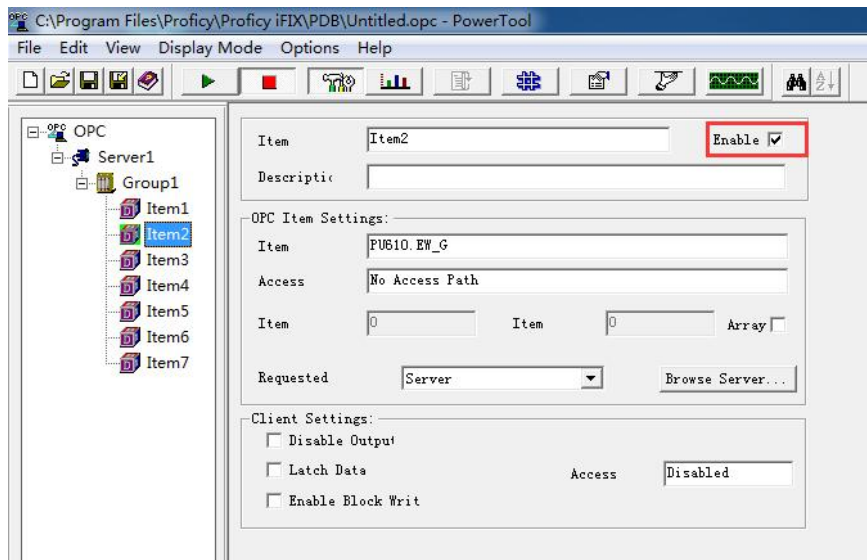


图 6-3-12

13. 然后点击菜单栏 File->SaveAs... 把配置文件另存到 C:\Program Files\Proficy\Proficy iFIX\PDB\目录下，命名为 OPC.OPC，点击工具栏



中的“Setup”按钮，选择第二个选项卡“Default Path”，设置如图 6-3-13。

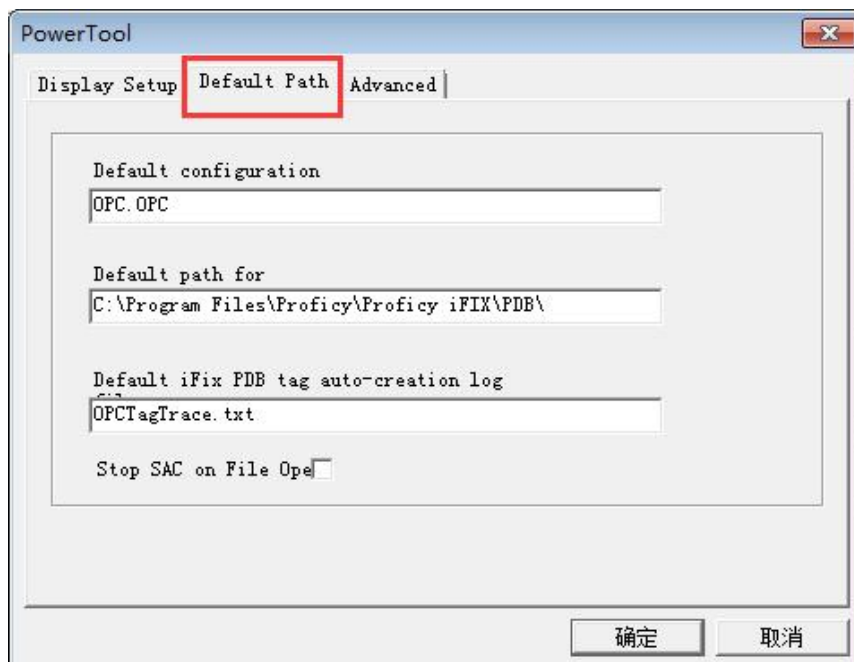


图 6-3-13



14. 然后点击启动按钮，配置完成之后关闭窗口。

6.3.2 在 IFIX 5.5 数据库中添加变量

1. 打开 IFIX 工作台，如图 6-3-14，在工程树中选择 FIX-I/O 驱动器-发现与配置，然后双击打开，如图 6-3-15。



图 6-3-14

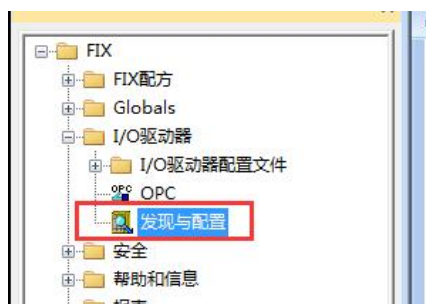


图 6-3-15

2. 在发现代理选择向导对话框中点击下一步。



图 6-3-16

3. 选择 OPC Discovery Agent, 然后点击下一步。



图 6-3-17

4. 选择“使用本地 OPC 服务器”，然后点击下一步。

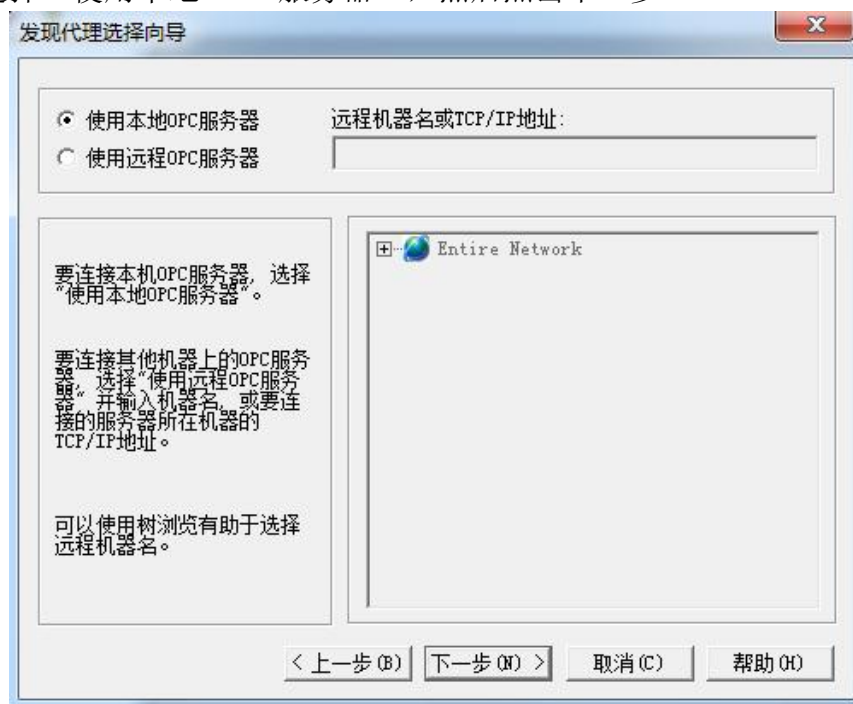


图 6-3-18

5. 选择 OPC 服务：OPC.OUT.PCOS.21，然后点击下一步。



图 6-3-19

6. 点击完成。



图 6-3-20

7. 然后在发现与配置软件总可以看到 OPC 服务器中的变量了，选择需要导入数据库中的变量。

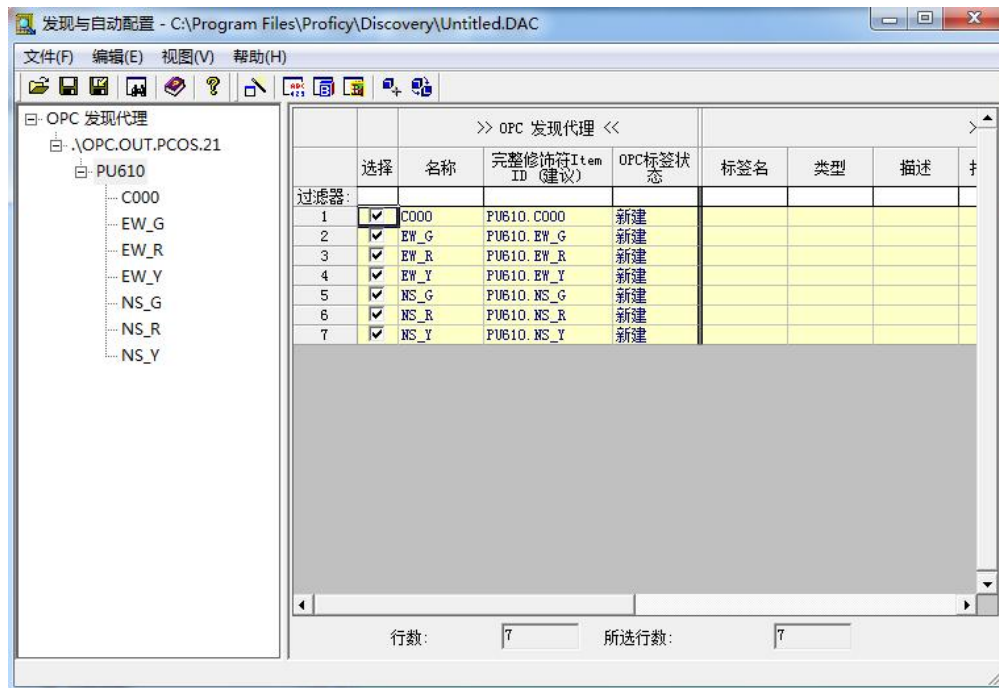


图 6-3-21

8. 在选择的变量上右键，选择“IFIX-所有向导”。

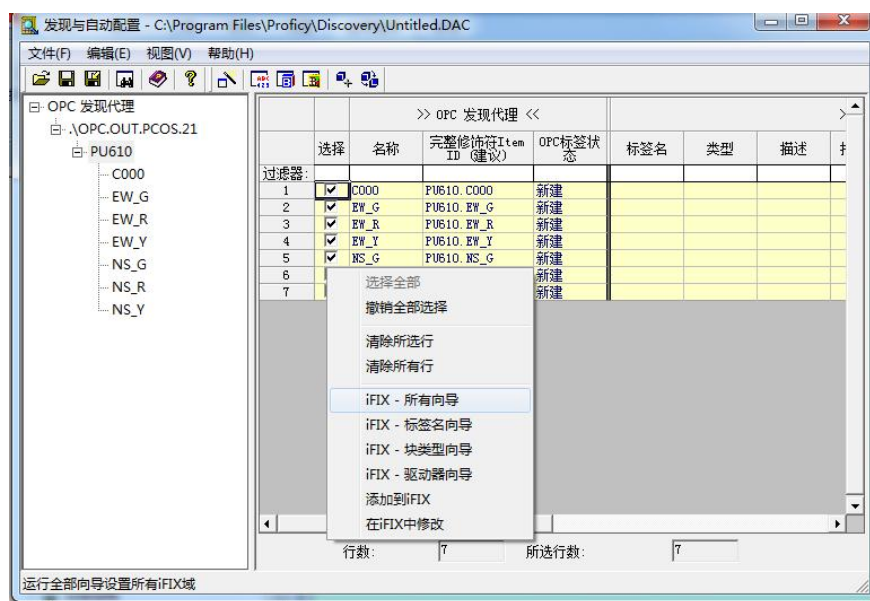


图 6-3-22

9. 弹出标签名和扫描向导对话框，默认所有参数，然后点击下一步。



图 6-3-23

10. 示例中所以参数都是开关量，块类型中选择：DR-数字寄存器，然后点击下一步。



图 6-3-24

11. 选择驱动器，点击下一步。



图 6-3-25

12. 在 iFIX OPC 客户端选择中。展开 Server1，点击 Group1。然后点击下一步。



图 6-3-26

13. 然后点击完成按钮。

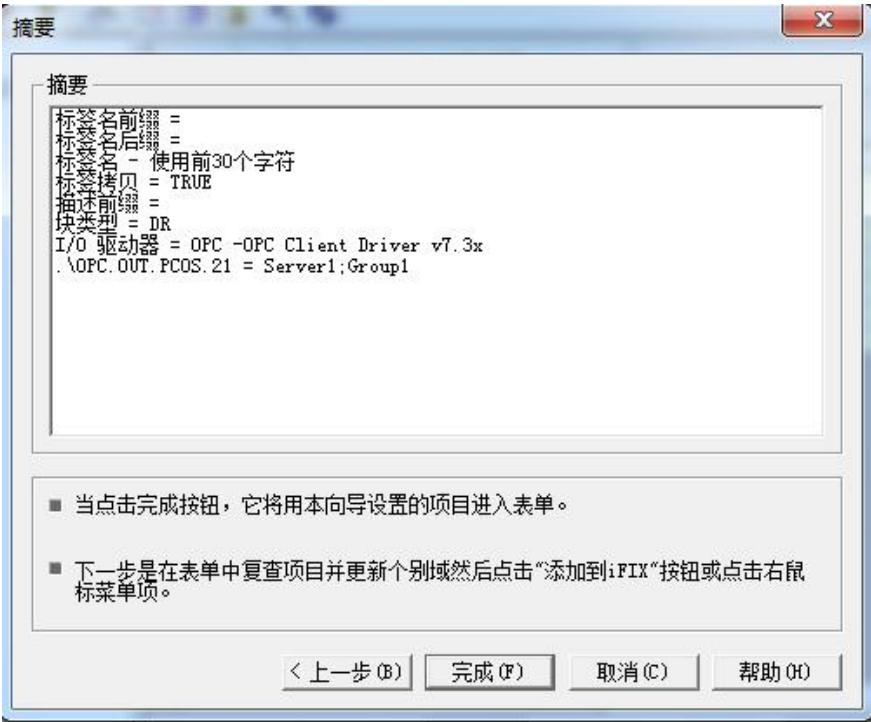


图 6-3-27

在列表中即可显示添加对应的好的变量；

>> OPC 发现代理 <<				>> Proficy HMI/SCADA - iFix <<								
	选择	名称	完整修饰符Item ID (建议)	OPC标签状态	标签名	类型	描述	扫描时间	驱动器	I/O 地址	硬件选项	信号条件
过滤器:												
1	☒	C000	PUB10.C000	新建	C000	DR	PUB10.C000		OPC	Server1:Gr		
2	☒	EW_G	PUB10.EW_G	新建	EW_G	DR	PUB10.EW_G		OPC	Server1:Gr		
3	☒	EW_R	PUB10.EW_R	新建	EW_R	DR	PUB10.EW_R		OPC	Server1:Gr		
4	☒	EW_Y	PUB10.EW_Y	新建	EW_Y	DR	PUB10.EW_Y		OPC	Server1:Gr		
5	☒	NS_G	PUB10.NS_G	新建	NS_G	DR	PUB10.NS_G		OPC	Server1:Gr		
6	☒	NS_R	PUB10.NS_R	新建	NS_R	DR	PUB10.NS_R		OPC	Server1:Gr		
7	☒	NS_Y	PUB10.NS_Y	新建	NS_Y	DR	PUB10.NS_Y		OPC	Server1:Gr		

图 6-3-28

14. 然后再从变量上右击，在弹出的菜单中选择“添加到 IFIX”。

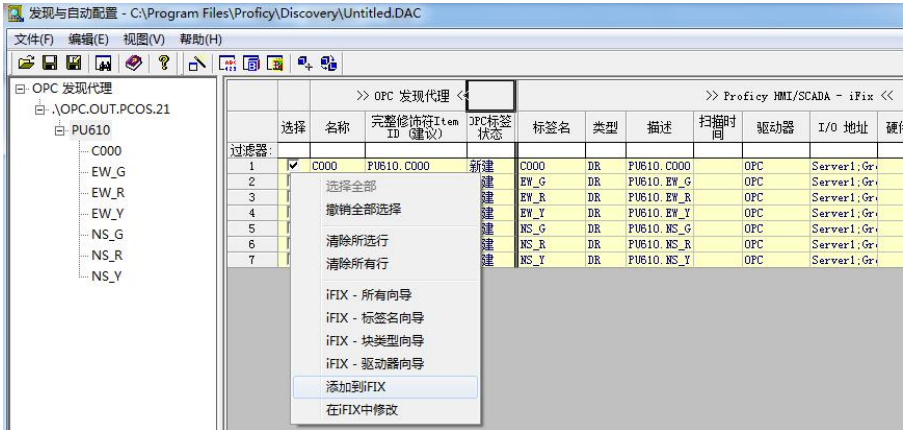


图 6-3-29

15. 所选数据添加到数据库中。

>> OPC 发现代理 <<					>> Proficy HMI/SCADA - iFix <<								
	选择	名称	完整修饰符Item ID (建议)	OPC标签状态	标签名	类型	描述	扫描时间	驱动器	I/O 地址	硬件选项	信号条件	最后动作
过滤器:													
1	☒	C000	PU610.C000	新建	C000	DR	PU610.C000		OPC	Server1;Gr			确定
2	☒	EW_G	PU610.EW_G	新建	EW_G	DR	PU610.EW_G		OPC	Server1;Gr			确定
3	☒	EW_R	PU610.EW_R	新建	EW_R	DR	PU610.EW_R		OPC	Server1;Gr			确定
4	☒	EW_Y	PU610.EW_Y	新建	EW_Y	DR	PU610.EW_Y		OPC	Server1;Gr			确定
5	☒	NS_G	PU610.NS_G	新建	NS_G	DR	PU610.NS_G		OPC	Server1;Gr			确定
6	☒	NS_R	PU610.NS_R	新建	NS_R	DR	PU610.NS_R		OPC	Server1;Gr			确定
7	☒	NS_Y	PU610.NS_Y	新建	NS_Y	DR	PU610.NS_Y		OPC	Server1;Gr			确定

图 6-3-30

16. 打开 IFIX 数据库，通过向导添加的 OPC 服务器的变量如图 6-3-31、图 6-3-32。

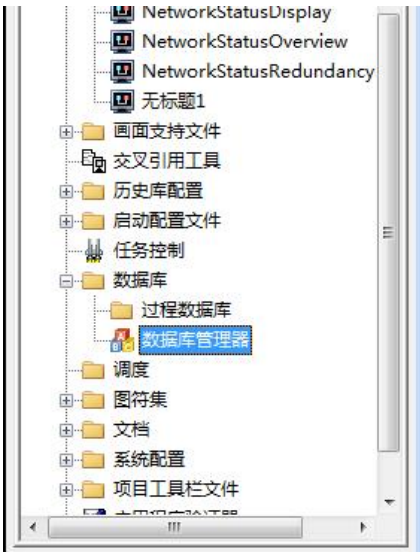


图 6-3-31

Proficy iFIX 数据库管理器 - [FIX : 8 列]									
首页 视图 剪贴板 过程数据库 数据块 编辑 诊断 驱动程序									
	标签名	类型	描述	扫描时	I/O	I/O地址			当前值
1	C000	DR	PU610.C000	----	OPC	Server1;Group1;PU610.C000;No Acces			OPEN
2	EW_G	DR	PU610.EW_G	----	OPC	Server1;Group1;PU610.EW_G;No Acces			CLOSE
3	EW_G1	DR	PU610.EW_G	----	OPC	Server1;Group1;PU610.EW_G;No Acces			CLOSE
4	EW_R	DR	PU610.EW_R	----	OPC	Server1;Group1;PU610.EW_R;No Acces			OPEN
5	EW_Y	DR	PU610.EW_Y	----	OPC	Server1;Group1;PU610.EW_Y;No Acces			OPEN
6	NS_G	DR	PU610.NS_G	----	OPC	Server1;Group1;PU610.NS_G;No Acces			OPEN
7	NS_R	DR	PU610.NS_R	----	OPC	Server1;Group1;PU610.NS_R;No Acces			CLOSE
8	NS_Y	DR	PU610.NS_Y	----	OPC	Server1;Group1;PU610.NS_Y;No Acces			CLOSE

图 6-3-32

6.4 IFIX 画面组态

数据库变量添加完毕后，编辑画面。
新建一个画面，保存名为：红绿灯。

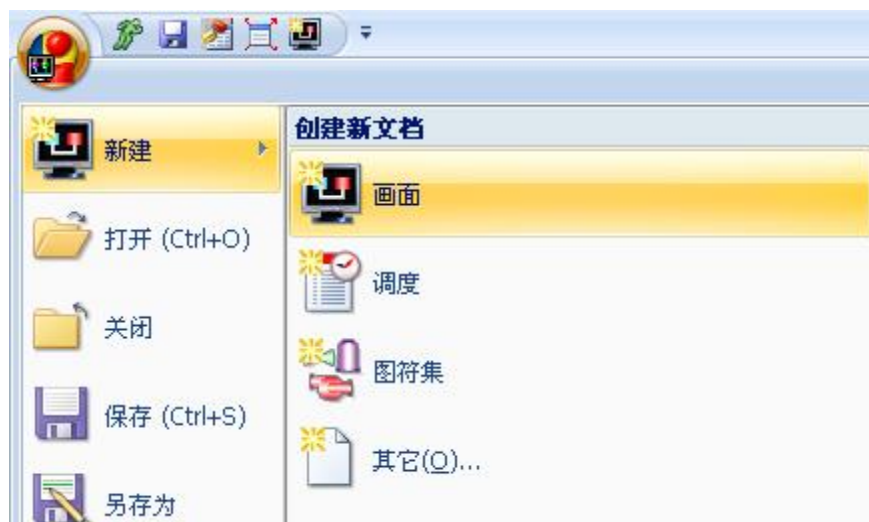


图 6-4-1

对应设置好变量及控件，运行画面如图 6-4-2:

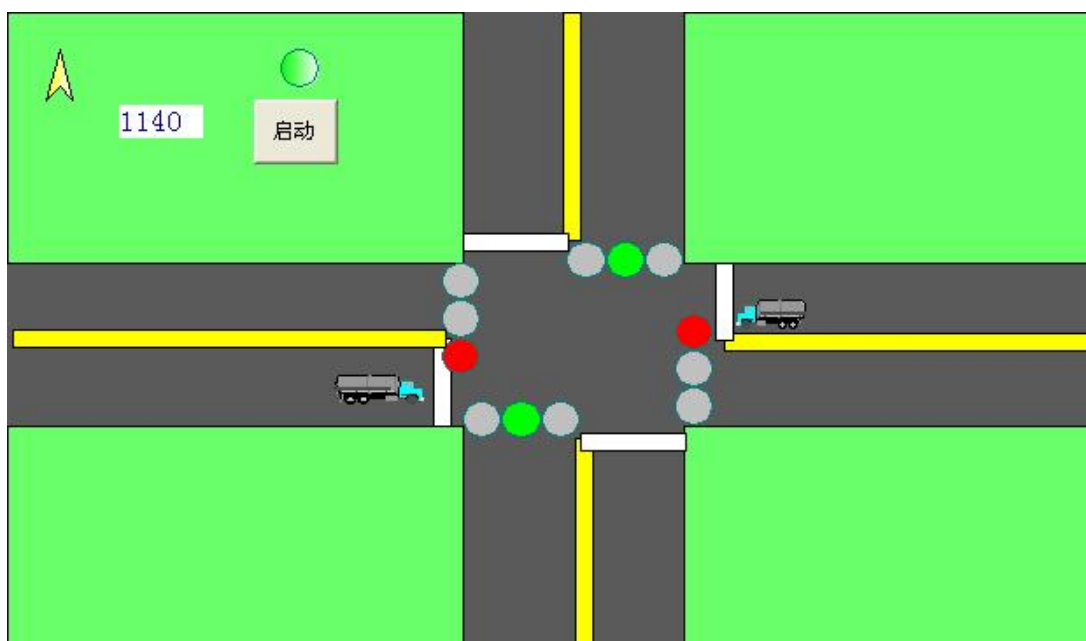


图 6-4-2

6.5 上位机 WINCC 通过 OPC 连接 Truhigh P600 系统

6.5.1 WINCC 的 OPC 变量添加

打开 wincc 变量管理，在右键菜单中选择添加新的驱动器程序，选择 OPC，如下图所示：

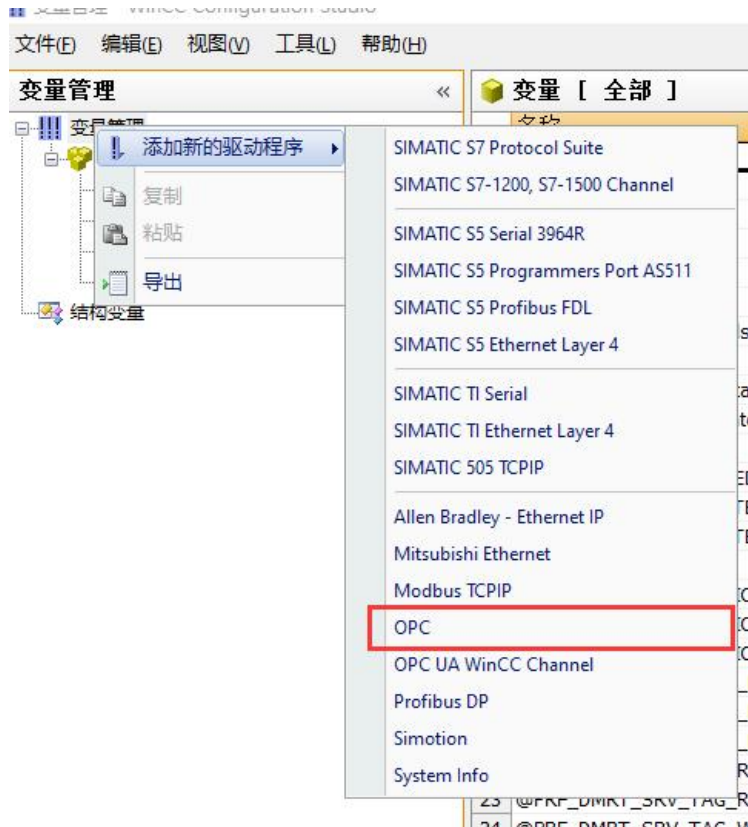


图 6-5-1

右键选择 OPC Groups，在弹出的列表中选择系统参数，如下图所示：



图 6-5-2

弹出 OPC 条目管理器，然后在 LOCAL 分支中找到“OPC.OUT.PCOS.21”，如果找不到则，选择视图->更新列表。

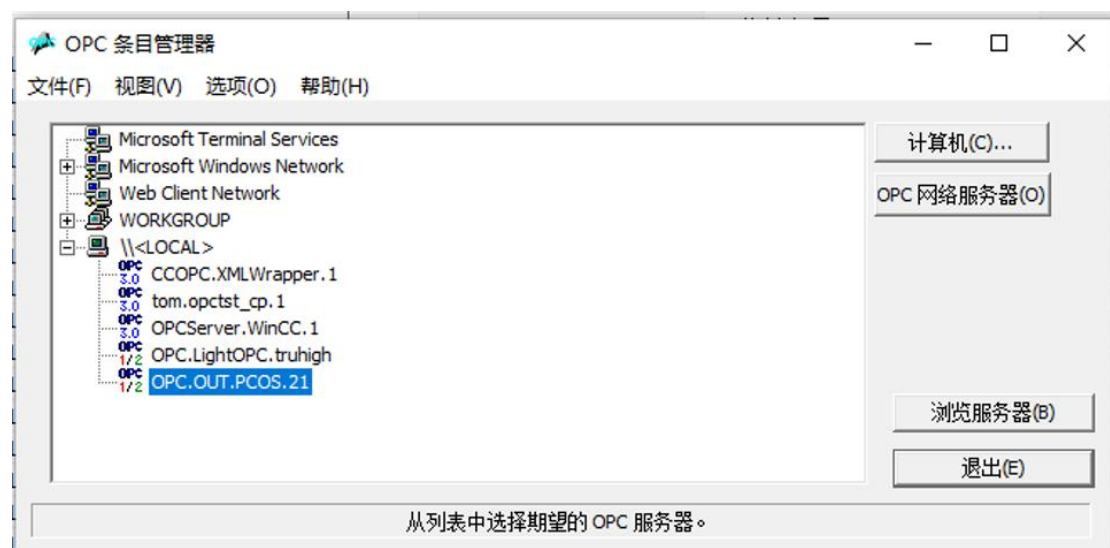


图 6-5-3

下一步，选择浏览服务器，弹出过滤标准窗口，如下图：



图 6-5-4

选择下一步，弹出条目添加窗口，如下图：

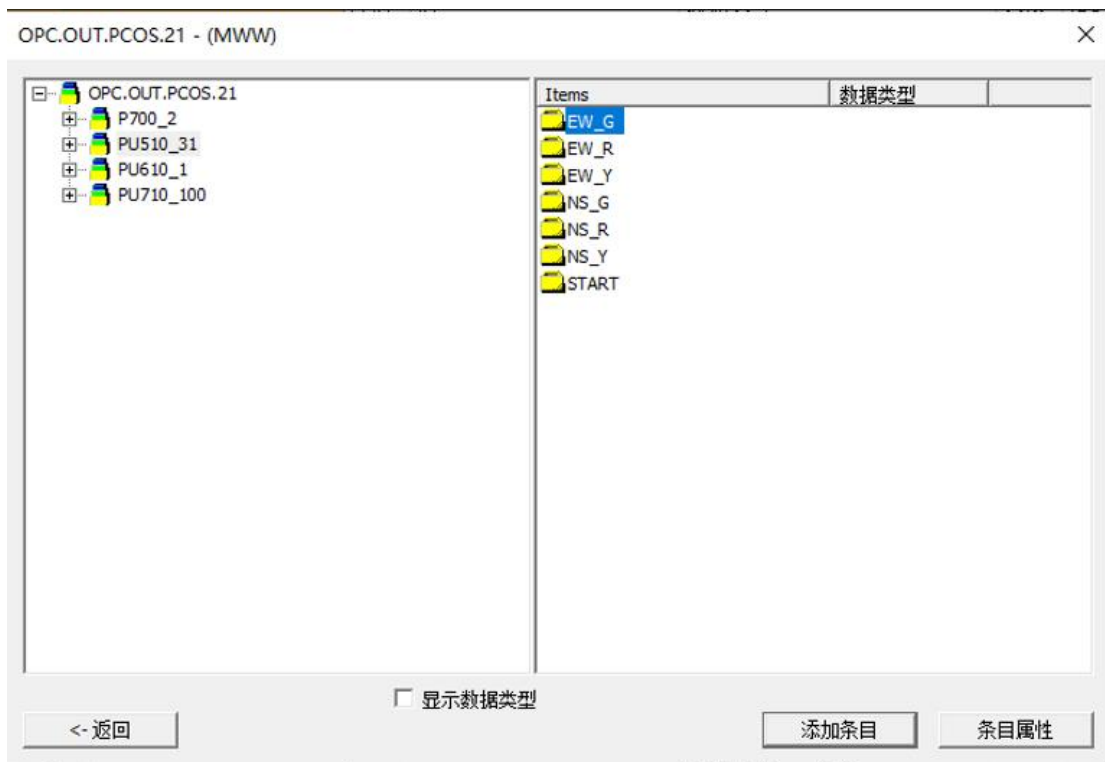


图 6-5-5

选择需要添加的变量条目，点击添加条目，选择添加到的组，然后点击完成，如下图。

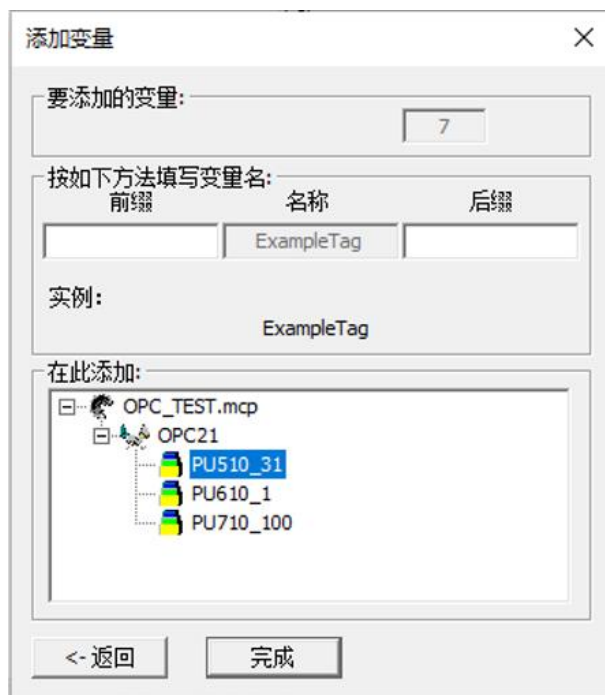


图 6-5-6

添加完成后在变量表里面可以看到已经添加的变，如下图：

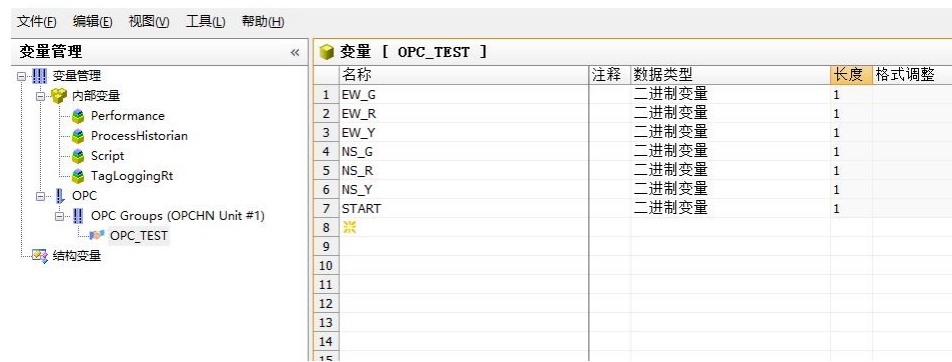


图 6-5-7

6.5.2 WINCC 的画面组态

变量添加完成后，添加画面，双击图形编辑器，进入图形界面设计，并且关联控件对象与 OPC 变量，详细操作参考 WINCC 相关文档，在此不再详述。

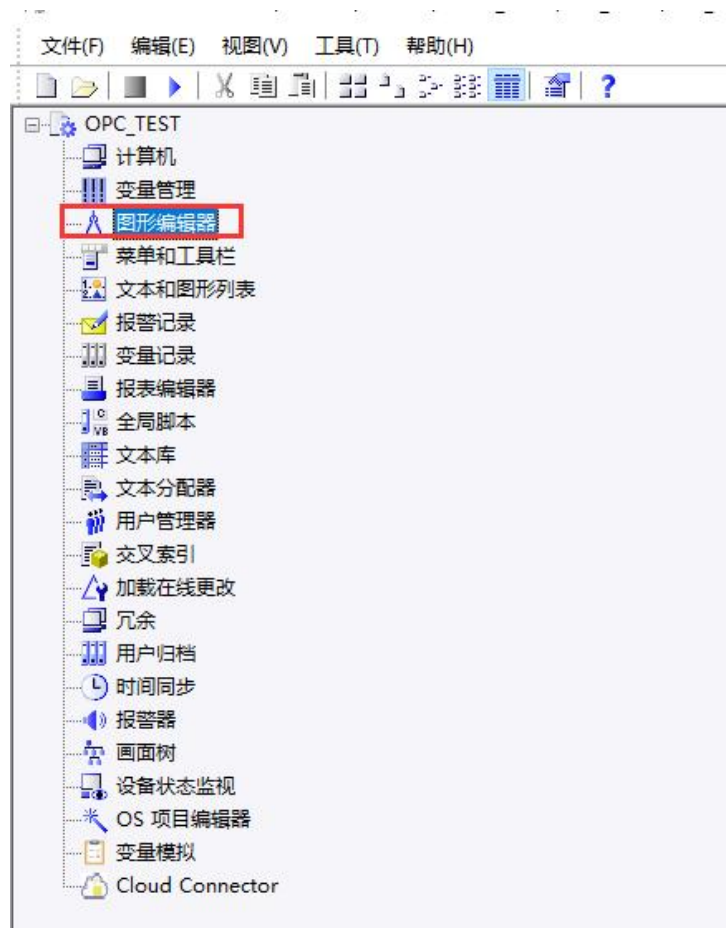


图 6-5-8