



昆仑通态
kunluntongtai

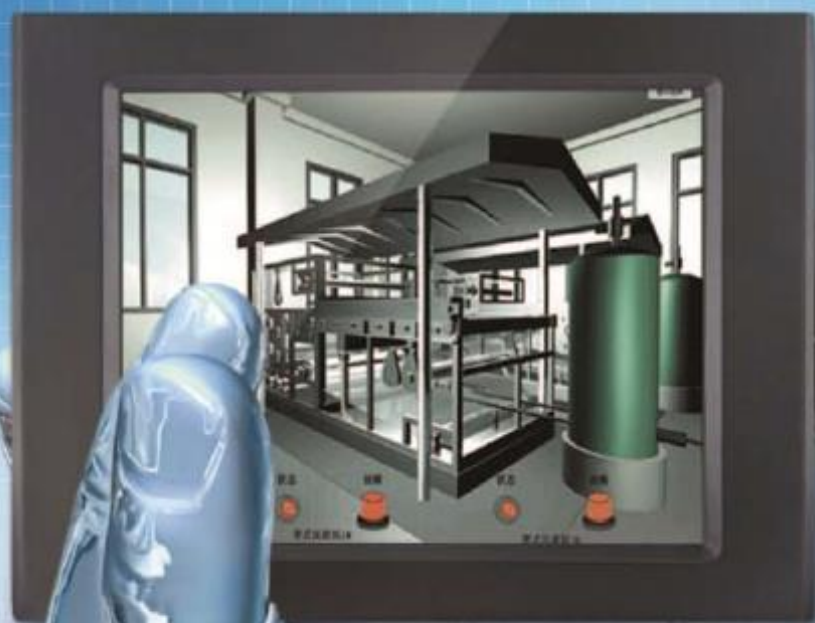
McgsPro使用教程

mcgsTpc

嵌入式一体化触摸屏

MCGS

全中文工控组态软件



目录

目录.....	1
内容简介.....	8
名词说明.....	9
※入门篇※.....	11
第 1 章 McgsPro 组态软件概述.....	12
1.1 软件简介.....	12
1.2 产品结构.....	12
第 2 章 McgsPro 组态软件基础应用.....	13
2.1 软件下载.....	13
2.2 安装和卸载.....	13
2.2.1 软件安装.....	13
2.2.2 软件卸载.....	14
2.3 运行和退出 McgsPro.....	15
2.3.1 运行 McgsPro.....	15
2.3.2 退出 McgsPro.....	15
2.4 新建工程.....	16
2.5 工程下载和上传.....	17
2.5.1 模拟运行.....	17
2.5.2 U 盘下载和上传工程.....	18
2.5.3 USB 从口下载和上传工程.....	20
2.5.4 网线下下载和上传工程.....	21
2.6 运行环境升级.....	22
2.6.1 U 盘升级运行环境.....	22
2.6.2 网线升级运行环境.....	22
2.7 工程兼容性.....	23
2.7.1 McgsPro 软件打开 McgsPro 工程.....	23
2.7.2 McgsPro 软件打开 MCGSE 工程.....	23
2.8 工程备份.....	24
2.9 软件帮助.....	24
2.9.1 McgsPro 帮助手册.....	24
2.9.2 设备帮助.....	24
第 3 章 McgsPro 简单工程制作.....	26
3.1 工程组成.....	26
3.2 工程需求分析.....	26
3.3 控制系统分析.....	27
3.3.1 系统组成.....	27
3.3.2 软硬件选型.....	27
3.3.3 PLC 地址和 McgsPro 工程画面规划.....	27
3.4 PLC 工程组态.....	27
3.5 McgsPro 工程组态.....	28
3.5.1 创建工程.....	28
3.5.2 设备窗口组态.....	29
3.5.3 实时数据库组态.....	32

3.5.4 将变量和通道相连接	34
3.5.5 用户窗口组态	35
3.5.6 模拟运行	42
3.5.7 下载工程到 TPC	42
※进阶篇※	44
第 1 章 McgsPro 软件功能	45
1.1 界面布局	45
1.2 菜单栏	45
1.2.1 文件菜单	45
1.2.2 编辑菜单	46
1.2.3 查看菜单	46
1.2.4 排列菜单	47
1.2.5 工具菜单	48
1.2.6 窗口菜单	50
1.2.7 帮助菜单	51
1.3 工具栏	51
1.3.1 工作台窗口工具条	51
1.3.3 动画组态窗口工具条	51
1.3.4 动画组态窗口绘图编辑条	51
1.3.5 脚本程序窗口工具条	52
1.4 工作台	52
1.4.1 主控窗口介绍	53
1.4.2 设备窗口介绍	55
1.4.3 用户窗口介绍	59
1.4.4 实时数据库介绍	60
1.4.5 运行策略介绍	67
第 2 章 窗口画面应用实例	72
2.1 工程名片	72
2.1.1 封面窗口和启动窗口设置	72
2.1.2 二维码展示信息	75
2.2 数据显示和输入	76
2.2.1 数据显示	76
2.2.2 数据输入	78
2.3 输入键盘设置	80
2.3.1 系统默认键盘	80
2.3.2 自定义键盘	81
2.4 公共窗口功能	87
2.5 条件满足自动弹窗	88
2.5.1 条件满足弹窗功能组态	89
第 3 章 动画功能应用实例	93
3.1 模拟仪表显示	93
3.2 动画功能组态	96
3.2.1 GIF	96
3.2.2 闪烁	97
3.2.3 可见度	98
3.2.4 跑马灯	99
3.3 多状态显示	99
3.3.1 动画显示	99
3.3.2 动画按钮	101

3.4 下拉式菜单选择功能	105
3.5 构件或窗口的属性和方法	107
3.5.1 功能概述	107
3.5.2 构件属性	107
3.5.3 窗口属性	107
3.5.4 构件方法	108
3.5.5 窗口方法	108
3.6 构件或窗口的事件	109
3.6.1 功能概述	109
3.6.2 构件事件	109
3.6.3 窗口事件	112
3.6.4 样例演示	113
3.6.5 运行展示	114
第 4 章 报警功能应用实例	116
4.1 报警功能介绍	116
4.2 实时报警组态	117
4.2.1 位报警组态	117
4.2.2 上下限报警	119
4.2.3 多状态报警	122
4.2.4 报警弹窗	123
4.3 历史报警功能	126
4.3.1 历史报警组态	126
4.3.2 历史报警删除	127
4.3.3 历史报警导出	127
4.4 报警信息显示和操作	127
4.4.1 报警信息翻页	127
4.4.2 报警值修改	129
4.5 报警信息配置	131
4.5.1 进入报警统一配置	131
4.5.2 设置统一报警配置	132
4.5.3 报警信息导出和导入	133
第 5 章 趋势曲线和数据处理	135
5.1 实时趋势曲线	135
5.1.1 绝对时钟趋势曲线	136
5.1.2 相对时钟趋势曲线	140
5.2 历史数据功能	140
5.2.1 历史数据存盘	140
5.2.2 历史数据显示	143
5.2.3 历史数据统计	149
5.2.4 历史数据导出	152
5.2.5 历史数据删除	154
5.3 数据处理	155
5.3.1 数据前处理	155
5.3.2 数据后处理	156
5.4 变量初值保存功能	156
5.4.1 设置初值属性	156
5.4.2 初值存盘函数	157
第 6 章 多重复制应用实例	158
6.1 单一控件的多重复制	158
6.2 单元的多重复制	164

第 7 章 指针功能应用实例	168
7.1 地址偏移应用样例	168
7.2 变量指针应用样例	169
第 8 章 配方功能应用实例	171
8.1 组态配方组	171
8.2 配方构件控制生产工艺	175
8.2.1 组态配方构件	175
8.2.2 配方构件方法应用	177
8.3 配方函数控制生产工艺	179
8.3.1 配方的加载和关闭	179
8.3.2 选择与修改配方	180
8.3.3 配方数据的上传与下载	180
8.3.4 配方的导入导出	181
8.3.5 配方的保存	182
8.4 顺序控制器功能	182
第 9 章 生产工艺流程预览应用实例	193
9.1 XY 曲线的基本属性和标注属性	193
9.2 XY 曲线的曲线属性	194
9.2.1 XY 曲线_历史数据	194
9.2.2 XY 曲线_CSV	195
9.2.3 XY 曲线_字符串	198
9.2.4 趋势图_历史数据	198
9.2.5 趋势图_CSV	199
9.2.6 趋势图_字符串	199
9.3 XY 曲线的方法函数	200
9.4 XY 曲线的注意事项	201
第 10 章 脚本程序功能介绍	202
10.1 脚本知识概要	202
10.1.1 编辑环境介绍	202
10.1.2 语言要素介绍	203
10.2 基本语句规则及应用	205
10.2.1 赋值语句	206
10.2.2 条件语句	206
10.2.3 循环语句	209
10.2.4 注释语句	209
10.2.5 跳出语句	210
10.2.6 退出语句	210
10.2.7 声明语句	210
10.3 脚本查错	210
第 11 章 脚本函数应用实例	211
11.1 使用帮助引导	211
11.2 运行环境函数	212
11.2.1 数组操作函数	212
11.2.2 运行策略函数	213
11.2.3 窗口操作函数	216
11.2.4 系统操作函数	218
11.2.5 操作日志函数	220
11.3 数据对象操作函数	220

11.3.1 数据设置函数	220
11.3.2 组对象操作函数	221
11.3.3 历史数据操作函数	222
11.3.4 保存初值（掉电保持）函数	227
11.3.5 报警操作函数	227
11.4 用户权限函数	229
11.5 时间函数	230
11.5.1 常用函数介绍	230
11.5.2 样例演示	230
11.6 数学函数	233
11.7 字符串函数	234
11.7.1 进制转换函数	234
11.7.2 字符串操作函数	235
11.7.3 数据解析函数	236
11.8 系统变量	237
11.8.1 常用系统变量介绍	237
11.8.2 样例演示_显示当天日期	237
11.8.3 样例演示_显示当天星期	238
11.9 文件操作函数	239
11.9.1 常用函数介绍	239
11.9.2 样例演示_文件（夹）的创建、查找和删除	240
11.9.3 样例演示_文件写入和文件读取	242
11.10 内存操作函数	244
11.10.1 常用函数介绍	244
11.10.2 样例演示	245
11.11 计时器函数	246
11.11.1 常用函数介绍	246
11.11.2 样例演示_启动、停止、获取当前值、获取状态	247
11.11.3 样例演示_获取当前值、重置当前值	249
11.11.4 样例演示_锁定屏幕 N 秒	250
第 12 章 多语言功能应用实例	254
12.1 多语言：中文、英语	254
12.2 多语言：中文、自定义语种	258
12.3 多语言的导入导出	260
第 13 章 安全功能应用实例	261
13.1 工程运行安全机制	261
13.1.1 安全属性	261
13.1.2 用户权限设置	262
13.1.3 操作日志记录	268
13.2 工程师权益保护	274
13.2.1 工程密码设置	274
13.2.2 TPC 系统密码设置	275
13.2.3 工程文件保护	275
13.2.4 工程上传	276
13.2.5 工程运行期限	277
第 14 章 打印功能应用实例	283
14.1 建立设备通道	283
14.2 组态画面	285
14.3 设置炜煌打印机	285
14.4 运行工程	286

第 15 章 McgsPro 监控应用.....	287
15.1 设备驱动	287
15.1.1 添加设备驱动	287
15.1.2 增加设备驱动	288
15.1.3 安装设备驱动	289
15.1.4 查找设备驱动	289
15.2 设置设备参数	290
15.2.1 父设备参数	290
15.2.2 子设备参数	291
15.2.3 设备命令	292
15.3 添加通道并连接变量	295
15.3.1 添加通道	295
15.3.2 快速连接变量	296
15.3.3 其他连接变量的方式	296
15.4 McgsPro 与西门子 200PPI 串口通讯及监控	298
15.4.1 硬件连接	298
15.4.2 TPC 通讯设置	299
15.4.3 PLC 通讯设置	300
15.4.4 组态画面	300
15.5 McgsPro 与三菱 FX3U 网口通讯及监控	302
15.5.1 硬件连接	303
15.5.2 TPC 通讯设置	303
15.5.3 PLC 通讯设置	306
15.5.4 组态画面	309
15.6 模拟通讯	310
15.6.1 硬件连接	310
15.6.2 通讯设置	310
15.6.3 组态画面	311
15.6.4 模拟运行	312
15.7 通过通道导入导出替换 PLC	312
15.7.1 西门子 SMART200 设备工程情况	312
15.7.2 西门子 SMART200 与三菱 FX3U 通道对应关系	314
15.7.3 通过设备通道的导入导出进行设备替换	314
15.8 TPC 系统控制	317
15.8.1 TPC 系统控制功能介绍	317
15.8.2 TPC 系统控制的设备命令	318
15.8.3 TPC 系统控制应用举例	319
15.9 模拟设备	320
15.10 屏与屏之间的数据同步	321
15.10.1 硬件连接	321
15.10.2 组态画面	321
15.10.3 通讯设置	322
15.10.4 运行展示	326
15.11 PLC 通过 TPC1 同 TPC2 传输数据	326
15.11.1 硬件连接	326
15.11.2 组态设备	326
15.11.3 通讯设置	328
15.11.4 组态 PLC	328
15.11.5 运行展示	329
15.12 通讯常见问题分析处理	330
15.12.1 通讯与否	330
15.12.2 驱动最新	330
15.12.3 通讯状态	330

15.12.4 注意事项	330
※硬件篇※	331
第 1 章 TPC 产品介绍	332
第 2 章 TPC 系统功能	333
2.1 TPC 系统功能	333
2.1.1 TPC 产品信息	333
2.1.2 TPC 系统设置	333
2.2 TPC 常见异常	334
※辅助工具※	337
第 1 章 TPC 辅助工具	338
1.1 运行环境升级包	338
1.2 U 盘功能包	338
1.3 NK 升级包	338
1.4 特殊功能包	339
附录 1 McgsPro 支持的驱动	340
附录 2 样例工程	342

内容简介

McgsPro 组态软件是深圳昆仑通态科技有限责任公司（以下简称“昆仑通态”），为 G 系列、E 系列和 K 系列等基于 Linux 系统的触摸屏开发的专用人机界面组态编辑软件，该软件为用户提供了一个简单易用、功能齐全的开发环境，可满足不同行业用户的使用需求。

关于本教程

本教程以实际应用为导向，按照循序渐进的方式，系统地介绍了 McgsPro 组态软件的使用方法及功能应用。教程共分为五个部分：

- 入门篇：McgsPro 组态软件从安装到入门
- 进阶篇：McgsPro 组态软件功能详解
- 硬件篇：适配 McgsPro 组态软件的 TPC 产品介绍
- 辅助工具：TPC 系统维护工具和特殊功能部署工具介绍
- 附录：TPC 支持的通讯协议列表，以及其他补充资料

本教程涉及的内容可能会由于组态软件版本更新、TPC 产品改进等原因发生变化，请以最新版本教程内容为准，如有改动恕不另行通知。

联系我们

公 司：深圳昆仑通态科技有限责任公司

地 址：深圳市龙岗区雪岗路 2018 号天安云谷一期 3 栋 B 座 2401 室

官 网：www.mcgs.cn

电 话：4006007062

名词说明

为了便于读者对本教程的理解，本节对教程中出现的专有名词进行说明。

专有名词	含义
昆仑通态	深圳昆仑通态科技有限责任公司
HMI	人机界面,又称用户界面或使用界面,是系统和用户之间进行交互和信息交换的媒介,它实现信息的内部形式与人类可以接受形式之间的转换。凡参与人机信息交流的领域都存在着人机界面。
TPC	昆仑通态生产的 HMI 人机界面,提供工程运行的硬件平台
McgsPro/ McgsPro 组态软件	昆仑通态开发的安装、运行于电脑端的组态软件,工程师在此完成 HMI 工程开发和调试工作
系统 NK	系统 NK 可以简单的看作 TPC 的操作系统, McgsPro 运行环境就运行在 TPC 系统 NK 中,标准 TPC 产品自带系统 NK
McgsPro 运行环境	用于在 TPC 上运行用户开发的 McgsPro 工程,标准 TPC 产品自带 McgsPro 运行环境
工作台	工作台是进入工程之后看到的第一个窗口,它包含 5 个窗口页:主控窗口、设备窗口、用户窗口、实时数据库、运行策略。工作台相当于一个大的容器,可以放置一个主控窗口、一个设备窗口和多个用户窗口,负责这些窗口的管理和调度,并调度用户策略的运行
主控窗口	是组态工程结构的主框架,它位于控制台的首位,用户可在主控窗口内设置系统运行流程及特征参数
设备窗口	在设备窗口中建立系统与外部硬件设备的连接关系,使系统能够从外部设备读取数据并控制外部设备的工作状态,实现对工业过程的实时监控
用户窗口	用来放置图元、图符和动画构件等各种图形对象,通过对图形对象的组态设置,建立与实时数据库的连接,来完成图形界面的设计工作。
实时数据库	是整个软件的核心,从外部硬件采集的数据送到实时数据库,再通过【用户窗口】更改数据库的值,最后由【设备窗口】输出到外部硬件。
运行策略	是用户为实现对系统运行流程自由控制所组态生成的一系列功能块的总称。
图元	是构成图形对象的最小单元。多种图元的组合可以构成新的、复杂的图形对象。
图符	多个图元对象按照一定规则组合在一起所形成的图形对象,称为图符对象。图符对象是作为一个整体而存在的,可以随意移动和改变大小
动画构件	是将工程监控作业中经常操作或观测用的一些功能性器件软件化,做成外观相似、功能相同的构件,存入组态软件的“工具箱”中,供用户在图形对象组态配置时选用,完成一个特定的动画功能。
数据对象	相当于全局变量和组对象,在所有的程序段共用,在实时数据库中定义
变量	包括全局变量、局部变量、系统变量
系统变量	McgsPro 定义的内部数据对象作为系统内部变量,在脚本程序中可自由使用。在使用系统变量时,变量的前面必须加“\$”符号
局部变量	只能在当前脚本中使用的变量,用脚本“Dim...As...”语句声明
组对象	一种特殊类型的数据对象,类似于编程语言中的数组和结构体,用于把相关的多个数据对象集合在一起,作为一个整体的定义和处理
局部数组变量	只能在当前脚本中使用,支持整数数组、浮点数数组、字符串数组、字节型数组四种数据类型,用脚本“Dim...As...”语句声明
父设备	父设备可以看作硬件接口

子设备	子设备放在父设备下，用于与该父设备对应的接口所连接的设备进行通讯
属性	通过属性，可以对构件或用户窗口有一个基本的描述和设置
方法	方法相当于针对构件和用户窗口的脚本操作函数
事件	事件就是当用户在对构件或在窗口中进行某些操作时，该构件或用户窗口会根据用户不同的操作进行相应的处理
时间值	所有的时间在电脑端的模拟将以 1970 年 1 月 1 日 8 时 0 分 0 秒为基线，而人机界面端的实际运行将以 1970 年 1 月 1 日 0 时 0 分 0 秒为基线，时间值为以基线作为基准增加的秒数
存盘	在运行过程中，根据设定周期或通过脚本调用将指定数据的值写入磁盘文件进行保存的过程，就是历史数据存盘
主口	USB-A 口，作用：USB 打印，U 盘上传/下载工程，数据导入/导出，运行 U 盘功能包
从口	USB-B 口，作用：上传/下载工程

※入门篇※

第 1 章 McgsPro 组态软件概述

本章主要讲解 McgsPro 组态软件产品构成。

1.1 软件简介



图 1.1-1 McgsPro 组态软件界面

McgsPro 组态软件拥有强大的界面显示组态功能、丰富的功能模块、良好的开放性、强大的数据库、可编程的脚本命令、完善的安全机制、便捷的仿真功能，可适应不同用户的需要。广泛应用于电力设备、纺织机械、生产设备、铁路行业、橡胶机械、中央空调、印刷机械、重工机械等行业。

1.2 产品结构

McgsPro 组态软件是昆仑通态研发的人机界面产品的一部分。人机界面产品由以下四个部分组成：

- McgsPro 组态软件：安装、运行于 PC 端，自动化工程师在此完成 McgsPro 工程开发和调试工作；
- McgsPro 运行环境：用于运行用户开发的 McgsPro 工程，标准 TPC 产品自带 McgsPro 运行环境；
- TPC 系统 NK：系统 NK 可以简单的看作 TPC 的操作系统，McgsPro 运行环境就运行在 TPC 系统 NK 中，标准 TPC 产品自带系统 NK；
- TPC：昆仑通态生产的 HMI 人机界面，提供工程运行的硬件平台。

第2章 McgsPro 组态软件基础应用

本章主要介绍 McgsPro 组态软件的下载、安装、卸载、运行、工程下载、工程上传、工程兼容性、工程恢复，以及软件帮助等基础应用。

2.1 软件下载

McgsPro 组态软件可在昆仑通态官网：www.mcgs.cn → 【下载中心】下载，或拨打热线电话 4006007062 获取。

2.2 安装和卸载

2.2.1 软件安装

- 1、支持的操作系统：
 - Windows 7 (中文简体)
 - Windows 8 (中文简体)
 - Windows 8.1 (中文简体)
 - Windows 10 (中文简体)
- 2、McgsPro 安装步骤如下：
 - ① 关闭电脑中的安全软件（杀毒软件和防火墙软件）；
 - ② 解压 McgsPro 组态软件压缩包；
 - ③ 双击【Setup.exe】，运行安装程序；
 - ④ 进入欢迎界面，如图 2.2-1 所示，点击【下一步】：

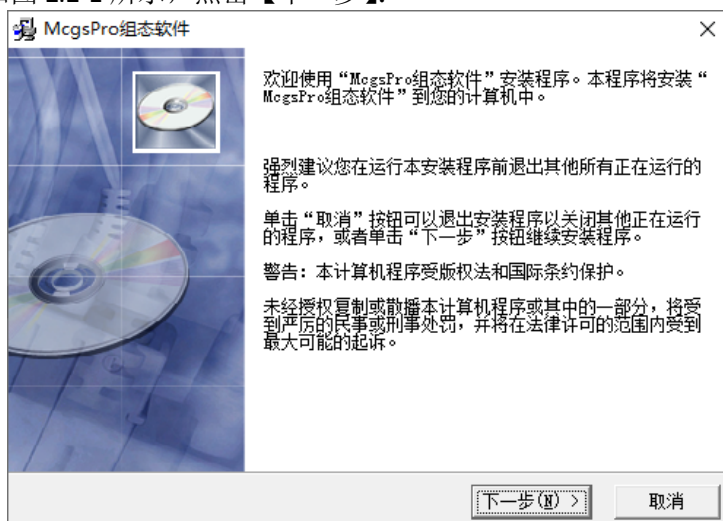


图 2.2-1 欢迎界面

- ⑤ 如图 2.2-2 所示，阅读“自述文件”后，点击【下一步】：



图 2.2-2 自述文件

- ⑥ 如图 2.2-3 所示，McgsPro 默认安装到“D:\McgsPro”。如需指定安装路径，请单击【浏览】更改。设置完成后，点击【下一步】：

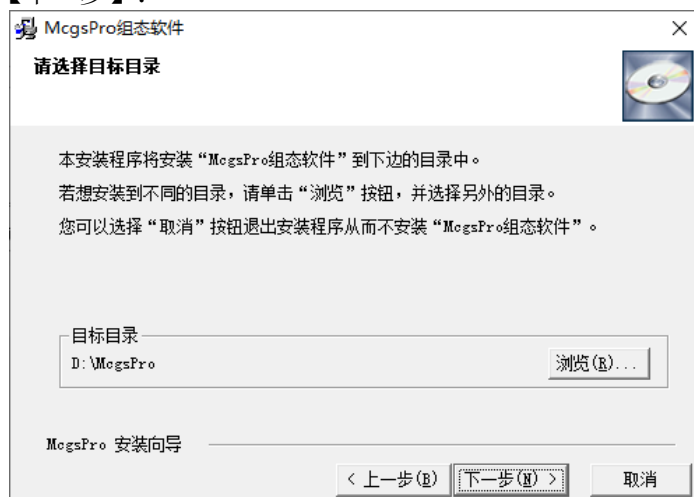


图 2.2-3 选择安装路径

- ⑦ 完成安装步骤⑥，再次点击【下一步】，McgsPro 软件启动安装。安装完成后，自动在桌面生成图 2.2-4 所示的快捷方式：“McgsPro 组态软件”、“McgsPro 模拟器”。

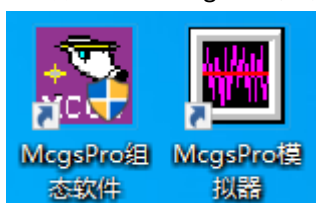


图 2.2-4 McgsPro 快捷方式

在上述安装过程中，用户单击【取消】，可退出安装。

2.2.2 软件卸载

执行卸载前，请先退出 McgsPro 组态软件。卸载步骤如下：

- ① 在电脑端依次打开【控制面板】→【卸载程序】或【程序和功能】；
- ② 在【卸载或更改程序】界面，单击选中【McgsPro 组态软件】→ 点击鼠标右键 →【卸载/更改】；
- ③ 如图 2.2-5 所示，卸载方式选择【自动】，点击【下一步】，等待 McgsPro 卸载完成。

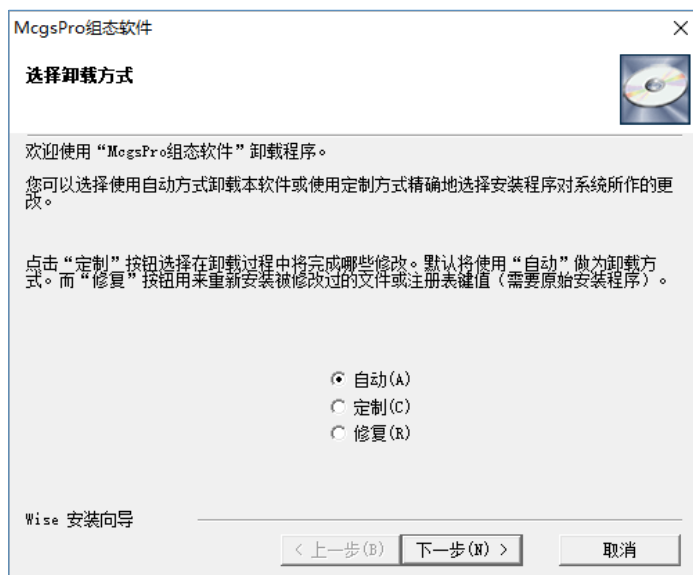


图 2.2-5 卸载 McgsPro 组态软件

2.3 运行和退出 McgsPro

2.3.1 运行 McgsPro

方法 1：从“开始”菜单启动，步骤如下：

【开始】→【所有程序】→【McgsPro 组态软件】→【McgsPro 组态软件】快捷方式

方法 2：双击“McgsPro 组态软件”快捷方式，如图 2.3-1 所示：



图 2.3-1 运行 McgsPro 组态软件

- 首次启动 McgsPro 组态软件，默认打开如图 2.3-2 所示的样例演示工程：

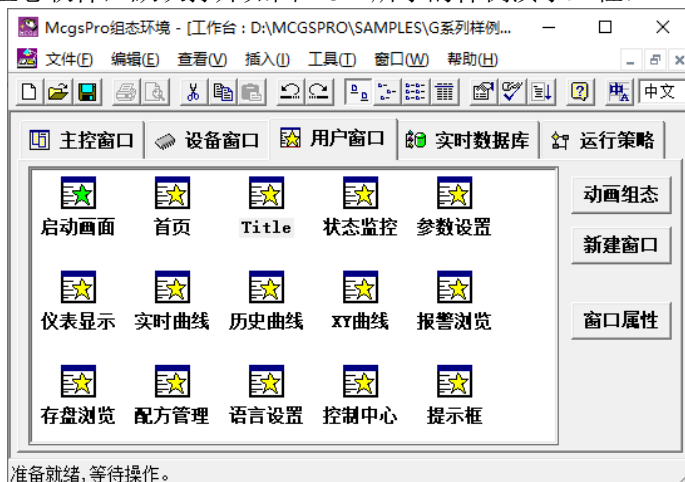


图 2.3-2 样例演示工程

- 非首次启动 McgsPro 组态软件，将自动打开上次操作的工程项目。

2.3.2 退出 McgsPro

启动 McgsPro 组态软件后，可使用下列任意一种方法退出：

- 单击 McgsPro 组态软件主界面右上角的【关闭】按钮；
- 如图 2.3-3 所示，单击 McgsPro 组态软件菜单栏的【文件】→【退出】。



图 2.3-3 退出 McgsPro

注意：退出 McgsPro 组态软件前，请确保工程已保存。

2.4 新建工程

新建工程步骤如下：

- ① 运行 McgsPro 组态软件；
- ② 依次点击菜单栏【文件】→【新建工程】，或者直接在工具栏单击图标 ，进行新建工程；
- ③ 如图 2.4-1 所示，依次设置新工程的 HMI 配置、组态配置参数（包括网格、构件风格、工程旋转），设置完成后，点击【确定】，完成新建工程操作。

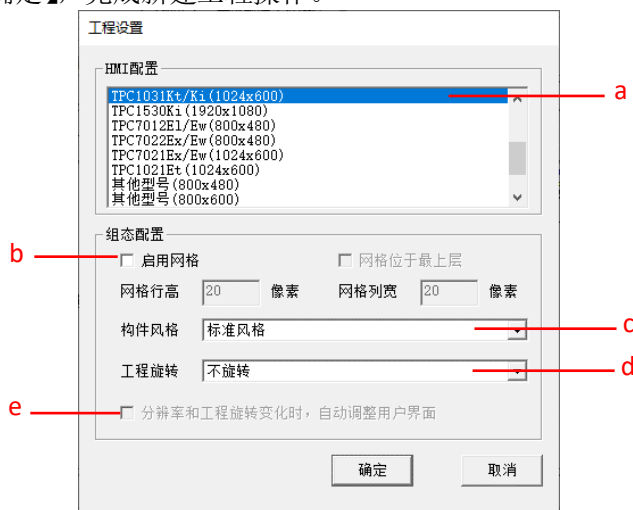


图 2.4-1 新建工程设置

- a. 选择 TPC 型号分辨率（如未找到或不清楚 TPC 型号，请选择“其他型号”中对应的分辨率）；
- b. 启用网格（网格仅组态时显示，运行时不显示）；
- c. 选择默认的构件绘制风格；
- d. 设置工程运行时画面旋转角度，支持不旋转、旋转 90 度、180 度、270 度；
- e. 分辨率自动调整设置，通过【文件】→【工程设置】修改工程设置时有效。修改工程分辨率或旋转角度时勾选此项，可实现构件大小自动调整。

注意：

- 工程新建完成后，图 2.4-1 中所有的参数设置，均可通过【文件】→【工程设置】重新修改。
- 新建工程名称由 McgsPro 组态软件自动命名为【新建工程 N】（N 为 ≥ 0 的整数）；
- 新建工程的路径与上一次运行 McgsPro 组态软件打开的工程路径一致。用户可以通过【文件】→【工程另存为】，重命名工程或将工程保存到指定路径。

2.5 工程下载和上传

McgsPro 组态软件可以在电脑端模拟运行工程，还可以通过 U 盘、USB 从口线（USBTYPE-B4Pin）、网线的方式下载工程到 TPC 中运行，以及从 TPC 中上传工程。注意，下载工程到 TPC 会清除原 TPC 中的工程。

通过菜单栏【工具】→【下载工程】，或者直接点击工具栏中的图标，进入图 2.5-1 所示的下载配置界面。在图中的【下载选项】，选择是否清除 TPC 中的用户数据，以及是否允许从 TPC 中上传工程。

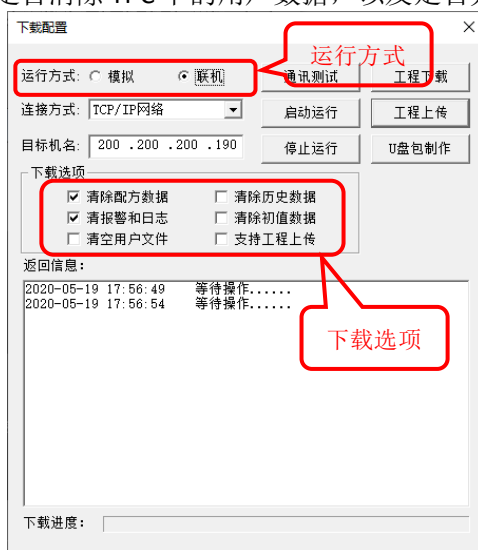


图 2.5-1 下载配置

图 2.5-1【下载选项】功能解释：

- 清除配方数据：更新工程时，清除 TPC 中保存的配方数据；
- 清除历史数据：更新工程时，清除 TPC 中保存的历史数据；
- 清除报警和日志：更新工程时，清除 TPC 中保存的历史报警数据和操作日志数据；
- 清除初值数据：更新工程时，清除 TPC 中保存的初值数据；
- 清空用户文件：更新工程时，清除 TPC 中用户文件区内的保存的用户文件（可通过菜单栏【工具】→【TPC 文件操作】、【U 盘包制作】→【附带文件功能】、或函数“!FileCopy”和“!FileDelete”，修改用户文件区中的文件）；
- 支持工程上传：允许用户上传本次更新到 TPC 中的工程。

2.5.1 模拟运行

使用 McgsPro 组态软件完成工程组态工作后，图 2.5-1 中运行方式选择【模拟】，再依次点击【工程下载】→【启动运行】，可将工程下载到模拟器中模拟运行，样例演示工程模拟运行画面如图 2.5-2 所示：



图 2.5-2 模拟运行画面

2.5.2 U 盘下载和上传工程

适用范围：所有型号的 TPC 均支持 U 盘下载或上传工程，U 盘必须为 FAT32 格式。

● U 盘下载工程

- ① 准备一只 FAT32 格式的 U 盘，插入电脑；
- ② 运行 McgsPro 组态软件，打开工程，进入下载配置界面；
- ③ 如图 2.5-3 所示，单击【U 盘包制作】→【U 盘功能包内容选择对话框】，功能包目录选择 U 盘目录，点击【确定】，完成 U 盘功能包制作；（可将多个不同名称的工程通过 U 盘包的方式制作到同一个 U 盘中）

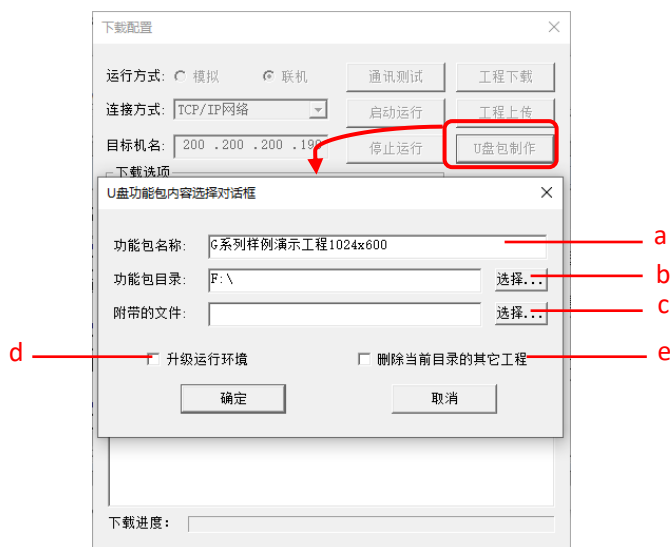


图 2.5-3 U 盘功能包制作

名词解释

- a. 编辑功能包名称
 - b. 选择 U 盘功能包制作路径（可选择非 U 盘路径，只需制作完成后手动拷贝“tpcbakup”到 U 盘根目录即可）
 - c. 选择附带的文件（附带文件会在下载工程时拷贝到用户文件区，用户文件区文件操作方法将在后续章节进行讲解。）
 - d. 打包当前版本 McgsPro 组态软件对应的运行环境到 U 盘功能包，供升级时使用
 - e. 选择是否删除“功能包目录”指向路径中的其他工程
- ④ 将 U 盘插入 TPC 的 USB 主口，TPC 弹出图 2.5-4 所示的提示界面，依次点击【是】→【用户工程更新】



图 2.5-4 准备更新工程

- ⑤ 如图 2.5-5 所示，在下载工程列表中单击选择需要下载的工程，点击【开始下载】，等待工程下载完成后重启 TPC 即可。

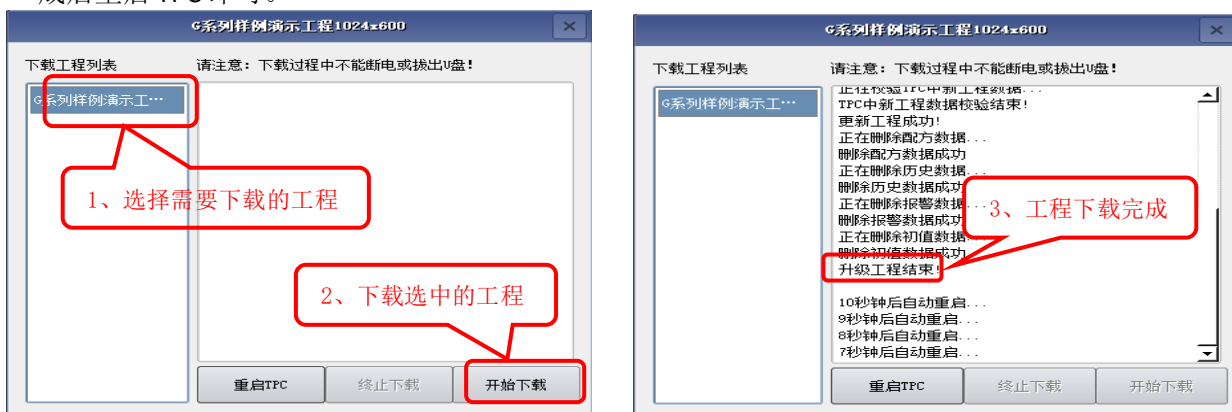


图 2.5-5 U 盘更新工程

● U 盘上传工程

前提：下载工程时勾选了图 2.5-1 所示的【支持工程上传】。

- ① 首先完成“U 盘下载工程”步骤①、②、③；
- ② 将 U 盘插入 TPC 的 USB 主口，TPC 弹出图 2.5-6 所示的提示界面，依次点击【是】→【上传工程到 U 盘】



图 2.5-6 准备上传工程到 U 盘

- ③ 如图 2.5-7 所示在 TPC 中工程支持上传的前提下，点击【上传】，工程将被上传到“U 盘 /tpcbakup/McgsTcpProject.mcsp”

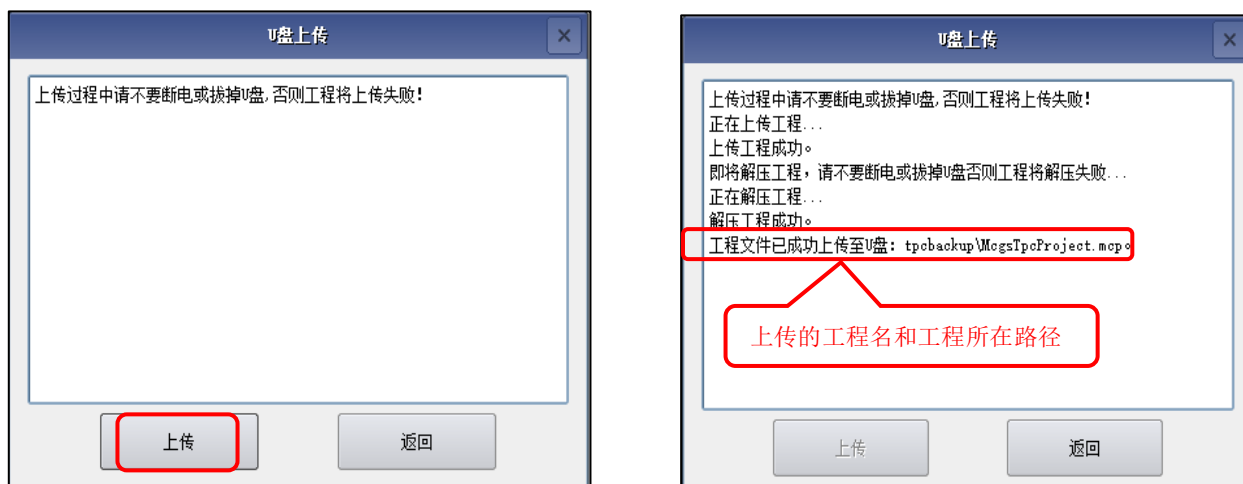


图 2.5-7 完成上传工程到 U 盘

2.5.3 USB 从口下载和上传工程

使用 USB 从口下载和上传工程，需要事先在【TPC 系统设置】中将 USB 模式设置为【从口模式】，设置方法请参考硬件篇章节 2.1.2。

● USB 从口下载工程

- ① 使用图 2.5-8 所示的从口线连接电脑和 TPC；



图 2.5-8 USB 从口线

- ② 运行 McgsPro 组态软件，打开工程，进入下载配置界面。如图 2.5-9 所示，运行方式：【联机】，连接方式【USB 通讯】。设置完成，点击【通讯测试】正常后，再点击【工程下载】即可。

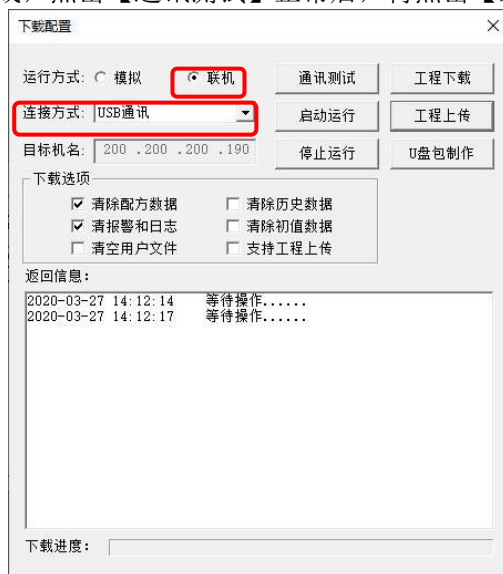


图 2.5-9 USB 从口下载工程设置

● USB 从口上传工程

前提：下载工程时勾选了图 2.5-1 所示的【支持工程上传】。

- ① 使用从口线连接电脑和 TPC；
- ② 运行 McgsPro 组态软件，任意打开一个工程。点击【菜单栏】→【工具】→【上传工程】，进入图 2.5-10 所示的“上传工程”界面；（也可点击图 2.5-9 中的【工程上传】，进入“上传工程”界面）

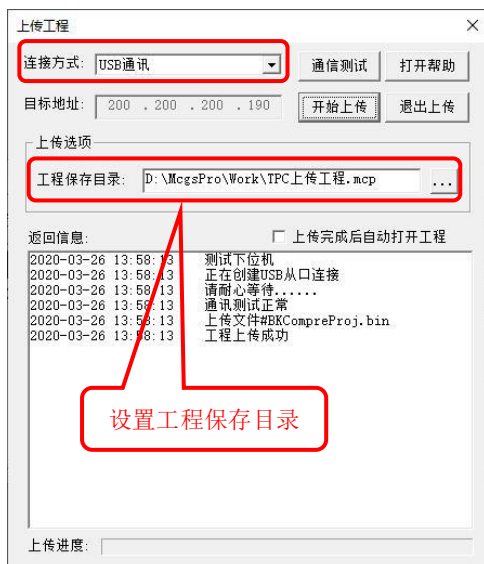


图 2.5-10 USB 从口上传工程

- ③ 在图 2.5-10 所示界面，连接方式选择【USB 通讯】，设置【工程保存目录】，最后进行通讯测试，完成后点击【开始上传】即可。

2.5.4 网线下载和上传工程

网线上传和下载工程，需要将电脑和 TPC 的 IP 地址修改为同一网段。TPC 的 IP 地址修改方法请参考硬件篇章第 2.1.2。

● 网线下载和上传工程

通过网线下载和上传工程操作步骤与“USB 从口下载和上传工程”类似。区别在于，下载和上传工程时，连接方式选择：【TCP/IP 网络】；目标机名：输入 TPC 的 IP 地址，如图 2.5-11 所示：

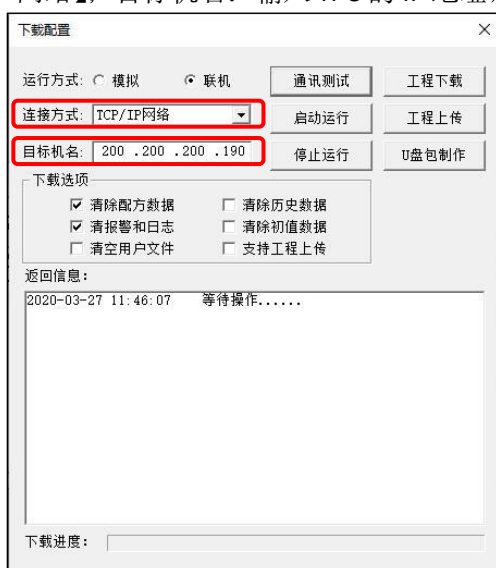


图 2.5-11 网线下载工程设置

2.6 运行环境升级

TPC 的运行环境可以看作运行在 TPC 中 McgsPro 软件平台，运行环境升级指的就是升级运行在 TPC 中的 McgsPro 软件版本，用户可以使用 U 盘或网线更新 TPC 运行环境。

- McgsPro 组态软件版本查看方法：打开组态软件 → 菜单栏 → 【帮助】 → 【关于】；
- TPC 运行环境版本查看方法：重启 TPC → 进度条界面按住 TPC 液晶面板 → 进入系统配置。

注意：

- ① 高版本的运行环境可以兼容低版本 McgsPro 软件组态的工程；
- ② 低版本的运行环境可能无法兼容高版本 McgsPro 软件组态的工程。

2.6.1 U 盘升级运行环境

有 2 种 U 盘包可以实现 TPC 运行环境升级功能，分别是：U 盘功能包、McgsPro 运行环境升级包。（U 盘必须使用 FAT32 格式）

- U 盘功能包

U 盘功能包：指入门篇章章节 2.5.2 制作的 U 盘功能包，需要用户在制作 U 盘功能包时，勾选【升级运行环境】，如图 2.6-1 所示：



图 2.6-1 打包运行环境到 U 盘包

注意：使用打包了运行环境的 U 盘功能包更新工程时，当 U 盘功能包中的运行环境版本高于 TPC 时，会自动触发 TPC 运行环境升级提示，由用户选择是否升级。

- McgsPro 运行环境升级包

跟随 McgsPro 组态软件一起发布的运行环境 U 盘升级包，该升级包提供运行环境升级功能，可无视 TPC 当前运行环境版本，强制将 TPC 运行环境升级到目标版本。

解压“运行环境升级包.rar”，拷贝解压后的“tpcbakcup”文件夹到 U 盘，将 U 盘插入待升级的 TPC，按照提示即可完成运行环境升级，详情请参考辅助工具章节 1.1。

2.6.2 网线升级运行环境

网线升级运行环境要求：电脑端安装的 McgsPro 组态软件版本不低于 3.3.1.3010 SP1，TPC 自身运行环境版本不低于 3.3.1.3955 SP1。网线升级运行环境操作步骤如下：

- ① 按照章节 2.5.4 中网线下载工程操作方法，用网线连接电脑与 TPC，进行工程下载；
- ② 当组态环境版本高于 TPC 运行环境版本，系统自动弹出图 2.6-2 所示的运行环境升级确认框，点击【确定】，自动将触摸屏运行环境升级后再下载工程；点击【取消】不升级运行环境，直接下载工程。

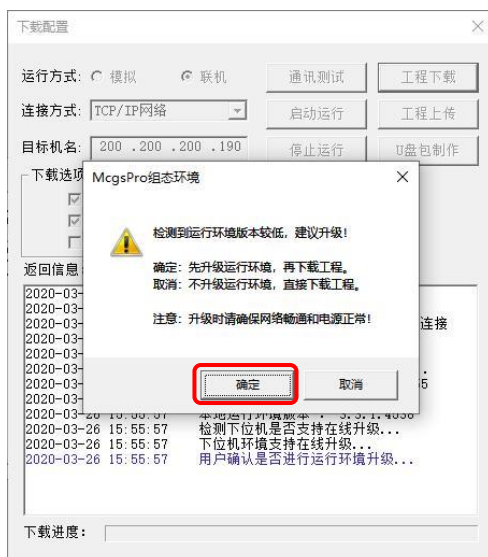


图 2.6-2 网线更新运行环境

2.7 工程兼容性

2.7.1 McgsPro 软件打开 McgsPro 工程

McgsPro 软件版本向上兼容，遵循如下原则：

- 老版本 McgsPro 组态软件开发的工程可以用新版本 McgsPro 组态软件打开

注意：

新版本软件打开旧版本软件组态的工程，会在工程所在目录生成 1 个旧版工程备份文件，文件命名方式：“旧工程名称 + 旧工程软件版本 + 序号 + .old”。手动修改旧版工程备份文件后缀名为“.mcp”，可以恢复旧工程。

例如：使用 McgsPro 3.3.1 SP2 打开 3.3.1 SP1.3 组态的工程，原工程名“测试工程.MCP”，备份文件如图 2.7-1 所示：

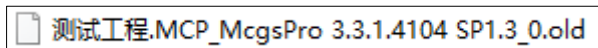


图 2.7-1 自动备份的工程文件示例

- 新版本 McgsPro 组态软件开发的工程无法用老版本 McgsPro 组态软件打开

2.7.2 McgsPro 软件打开 MCGSE 工程

McgsPro 支持打开 MCGSE 嵌入版组态的工程。操作步骤如下：

- ① 如图 2.7-2 所示，首先运行 McgsPro，点击菜单栏图标 ；
- ② 打开的文件类型选择【所有文件】；
- ③ 选择需要打开的 MCGSE 嵌入版工程（后缀：MCE）；
- ④ 单击按钮【打开】，按照提示打开 MCE 工程。

注意：McgsPro 打开 MCGSE 开发的工程，仍然会在工程所在目录生成 1 个旧版工程备份文件，文件命名方式：“旧工程名称 + 旧工程软件版本 + 序号 + .old”（参考图 2.7-1）。手动修改旧版工程备份文件后缀名为“.MCE”，可以恢复旧工程。

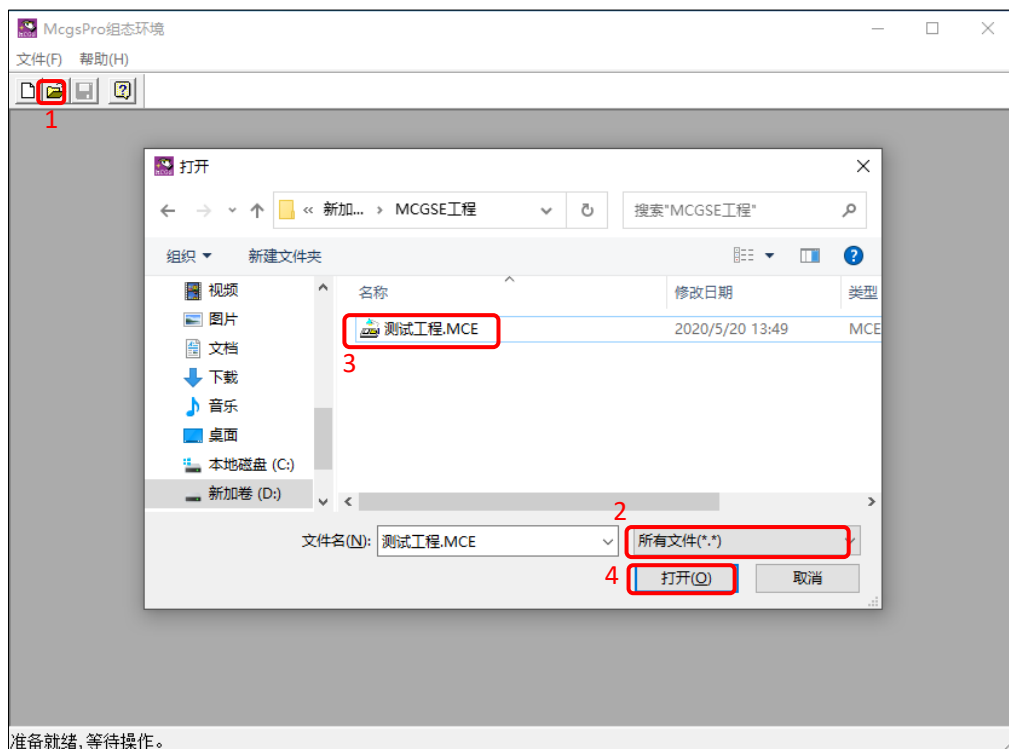


图 2.7-2 McgsPro 打开 MCGSE 工程

2.8 工程备份

用户在工程开发过程中，每次打开工程时，McgsPro 组态软件都会对工程进行一次备份操作，路径为 McgsPro 安装目录下：“McgsPro\Program\bakprj_MCP\工程名称”。

如图 2.8-1 所示，备份内容包括：bakprj.ini（存储原工程路径）、工程文件（命名：工程名 + 打开时间 + .bak）。将备份的“工程文件”后缀名修改为“.mcp”，再使用 McgsPro 组态软件打开，可以恢复工程。

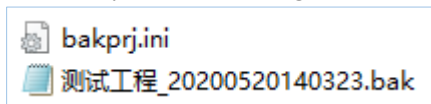


图 2.8-1 McgsPro 备份内容

注意：该备份方式是以工程名进行备份，同名工程会覆盖。

2.9 软件帮助

McgsPro 组态软件提供 McgsPro 帮助手册和设备帮助。

2.9.1 McgsPro 帮助手册

提供 McgsPro 组态软件功能和使用方法帮助。在已运行 McgsPro 组态软件的前提下，可通过以下三种方法打开 McgsPro 帮助手册：

- 按下键盘上的【F1】键；
- 点击 McgsPro 组态软件菜单栏的【帮助】；
- 通过构件属性设置对话框右下角的【帮助】按钮，直接打开对应构件帮助文档。

2.9.2 设备帮助

提供 McgsPro 驱动设备帮助，主要介绍 TPC 与不同驱动设备通讯时硬件接线、McgsPro 端通讯参数设置、驱动通道信息、驱动设备命令、通讯故障分析处理、以及 PLC 端通讯参数设置等。驱动设备帮助可以协助用户快速在 TPC 与下位机之间建立通讯，或者根据“通讯状态”处理可能出现的通讯故障。

以“Siemens_1200”驱动为例，其设备帮助打开方法如下：

- ① 运行 McgsPro 组态软件，新建一个工程，双击图 2.9-1 所示的【设备窗口】：

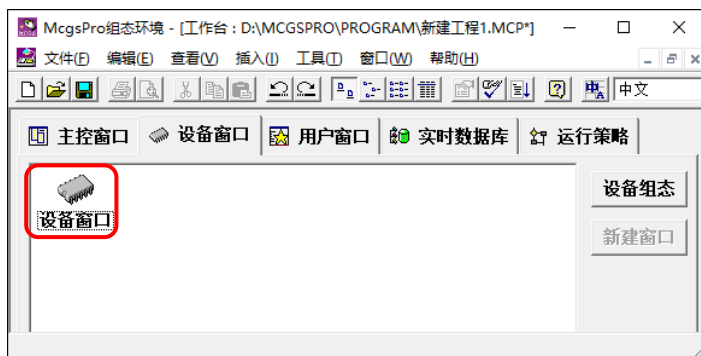


图 2.9-1 设备窗口

- ② 在设备窗口中依次添加“通用 TCP/IP 父设备”和“Siemens_1200”驱动，如图 2.9-2 所示：

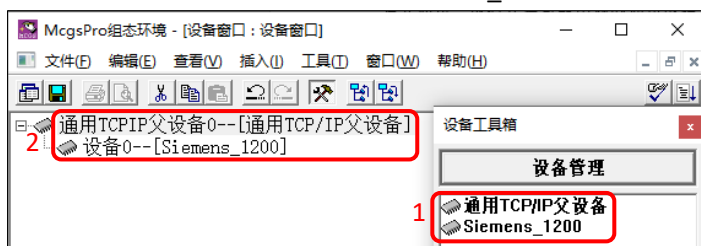


图 2.9-2 添加设备驱动

- ③ 双击添加的“Siemens_1200”，进入其设备编辑窗口，点击【打开设备帮助】，即可进入“Siemens_1200”驱动的设备帮助，如图 2.9-3 所示：

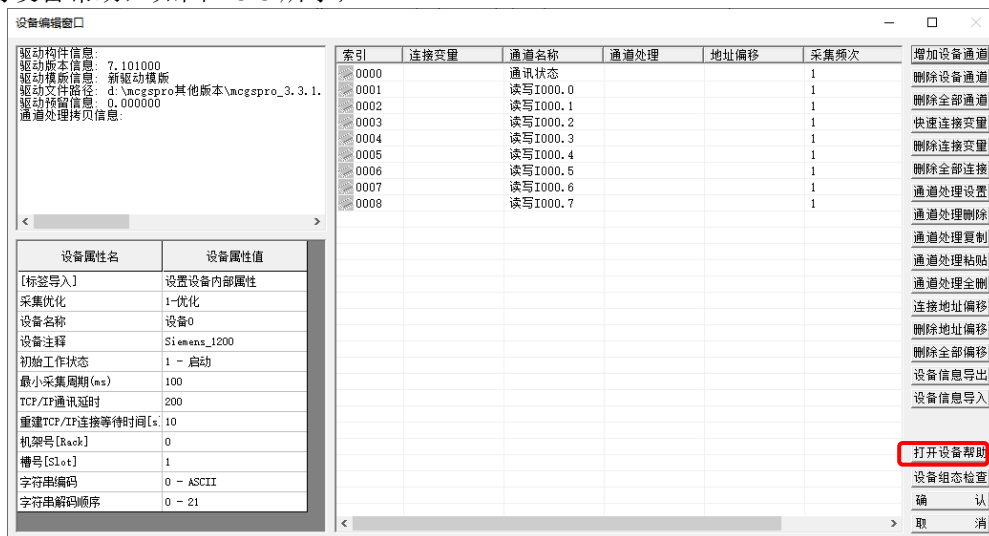


图 2.9-3 打开设备帮助

注意：不同驱动设备打开的设备帮助是不一样的，通过“Siemens_1200”只能打开“Siemens_1200”驱动对应的设备帮助，如需参考其他驱动的设备帮助，请按照上述步骤，选择需要的驱动设备进行操作。

第3章 McgsPro 简单工程制作

本章以【液位保持系统】为例，演示简单 McgsPro 工程制作过程。让大家对使用 McgsPro 的组态流程有一个直观的认识，以及对软件界面、软件基本功能和组态方式有初步的印象。

在实际应用中，用户可以选择先组态设备通道，建立 TPC 与外部设备的连接，再组态画面；或者先组态画面，再组态设备通道。本章样例使用第一种组态流程。

工程组态流程：

1. 建立新工程
2. 组态设备窗口
3. 组态实时数据库
4. 将变量与通道相连
5. 组态用户窗口
6. 模拟运行测试
7. 下载工程到 TPC

3.1 工程组成

McgsPro 组态软件由主控窗口、设备窗口、用户窗口、实时数据库和运行策略五个部分构成，如图 3.1-1 所示：

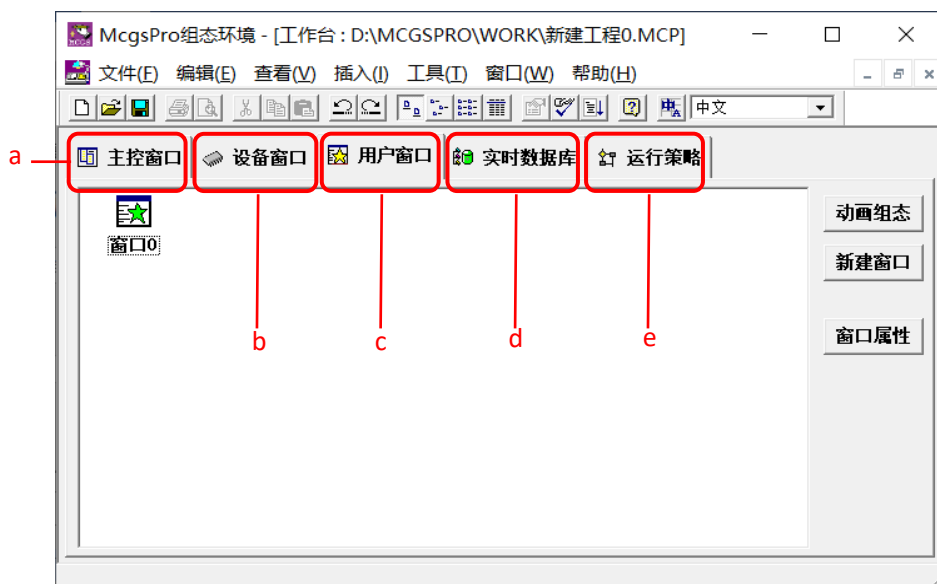


图 3.1-1 McgsPro 组态软件界面

a.主控窗口；b.设备窗口；c.用户窗口；d.实时数据库；e.运行策略

3.2 工程需求分析

现有一个蓄水池，它有一根进水管和一根出水管。进水管和出水管上各有一个水泵和一个电动阀门，由 PLC 控制。用一个液位计将蓄水池中的水位数据实时传递给 PLC，根据下游管网用水需要，PLC 对进水泵和阀门进行控制。当蓄水池液位过高或过低时，系统报警。

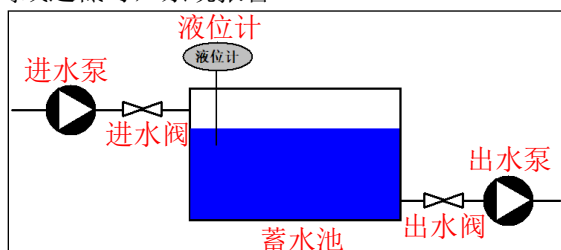


图 3.2-1 工程工艺流程图

3.3 控制系统分析

3.3.1 系统组成

根据工程需求，我们采用上位机、PLC 和检测仪表等构建该控制系统。

PLC 主要完成数据（液位计）采集、控制设备（水泵、电动阀）启动/停止等任务；上位机则使用 TPC 通过以太网与 PLC 进行通信，提供直观的人机交互界面，方便用户操作。

3.3.2 软硬件选型

根据以上系统分析，需配备以下设备和参数：

- 软件环境：McgsPro 组态软件
STEP 7 MicroWIN SMART
- 硬件环境：运行 Windows 7/8/8.1/10 操作系统的个人电脑一台
G 系列 TPC 一台（本例中选用 TPC1071Gi）
工业 PLC 一款（本例中选用西门子 Smart200）
电源（DC24V 电源一个，为 TPC 供电；AC220V 电源一个，为 PLC 供电）
网线（用于 TPC、PLC 和电脑三者之间通讯）

3.3.3 PLC 地址和 McgsPro 工程画面规划

● PLC 地址规划

序号	设备	地址	McgsPro 对应变量名	备注
1	进水泵	I0.0	进水泵启停状态	
2	出水泵	I0.1	出水泵启停状态	
3	进水阀	I0.2	进水阀开闭状态	
4	出水阀	I0.3	出水阀开闭状态	
5	进水泵	Q0.0	启停进水泵	
6	出水泵	Q0.1	启停出水泵	
7	进水阀	Q0.2	开关进水阀	
8	出水阀	Q0.3	开关出水阀	
9	液位计	AIW0	液位值	低于 1 米下限位报警，高于 9 米上限位报警

● McgsPro 工程画面规划

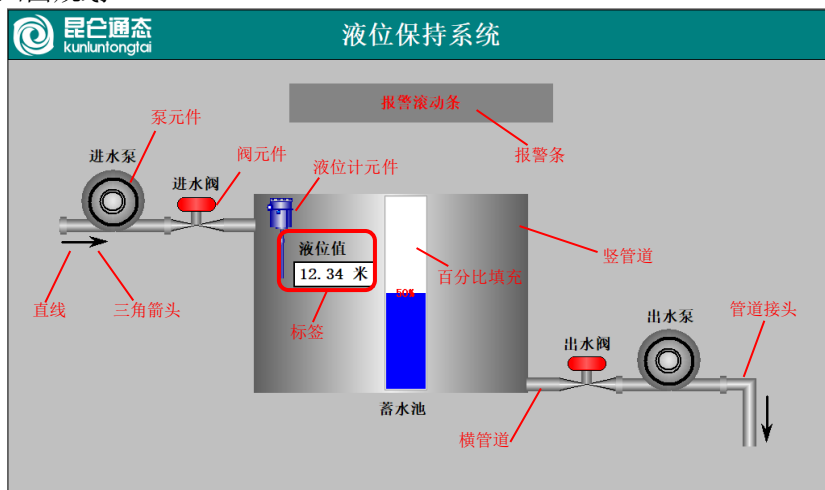


图 3.3-1 工程画面规划

3.4 PLC 工程组态

为了进行 TPC 与 PLC 之间的通讯，我们需要对 PLC 的通讯参数进行设置。

在本工程中，我们使用 STEP 7 MicroWIN SMART 软件对 Smart200 的通讯参数进行设置。打开软件后，首

先新建工程，选择合适的 PLC 后，点击项目列表中“通信”弹出通信窗口，点击“查找 CPU”按钮，找到 CPU 后，点击“确定”。如图：

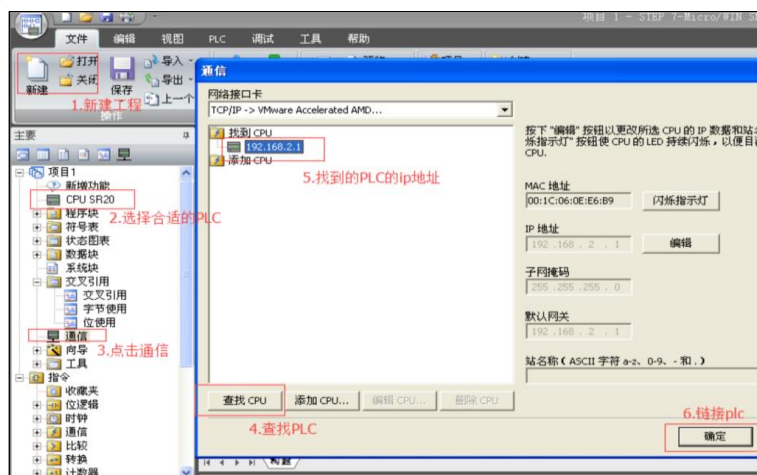


图 3.4-1 在 STEP 7 MicroWIN SMART 软件中添加 PLC

点击项目列表下“通信”，在通信窗口点击“设置”按钮设置 IP 地址等。如图：

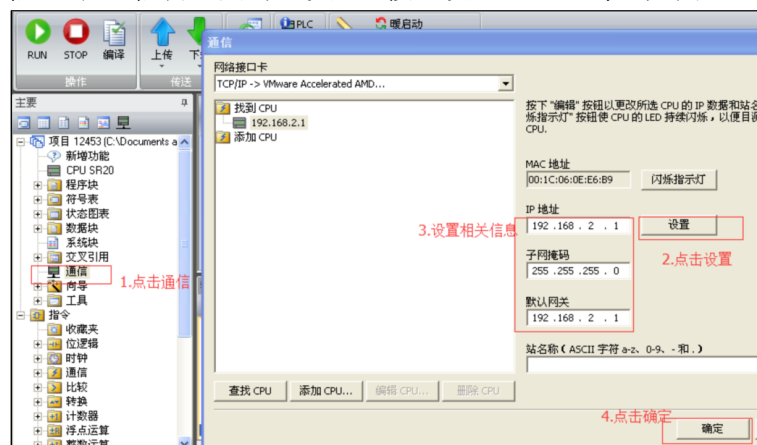


图 3.4-2 设置 PLC 的 IP 地址

注意：本节教程只介绍 PLC 端通讯参数设置方法，需要客户自行组态 PLC 程序。

3.5 McgsPro 工程组态

3.5.1 创建工程

- 启动 McgsPro 组态软件



图 3.5-1 工程开始界面

- 创建新工程

① 单击工具栏的  图标，新建工程

② 选择 TPC 型号

③ 进行组态配置，本样例采用默认设置

新建的工程会出现在安装目录下 Work 文件夹中，以“新建工程 N”命名。

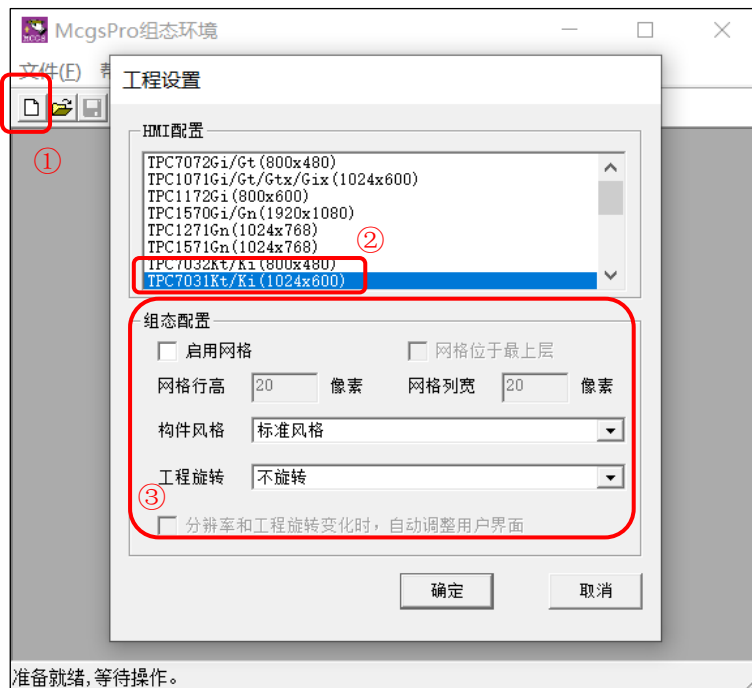


图 3.5-2 新建工程设置界面

3.5.2 设备窗口组态

- 打开设备窗口

① 选择【工作台】的【设备窗口】页面

② 双击下方【设备窗口】图标，进入设备窗口设置界面

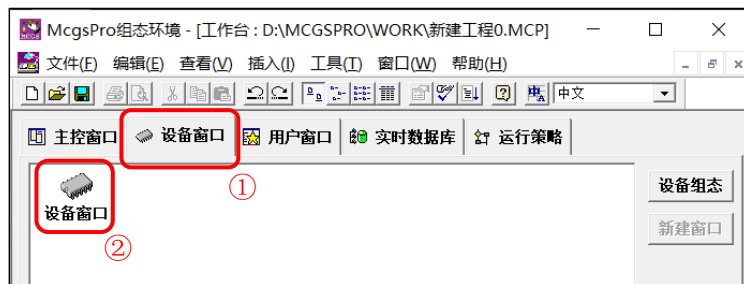


图 3.5-3 工作台界面

- 添加 Smart200 驱动

① 工具栏中的  图标，打开【设备工具箱】

② 双击【设备工具箱】中的【通用 TCP/IP 父设备】，在设备窗口设置界面添加 Smart200 驱动的父设备

③ 双击【设备工具箱】中的【西门子_Smart200】添加子设备

④ 弹出提示窗口，询问“是否使用‘西门子_Smart200’驱动的默认通讯参数设置 TCP/IP 父设备参数？”，选择“是”，在【通用 TCP/IP 父设备】下添加 Smart200 驱动

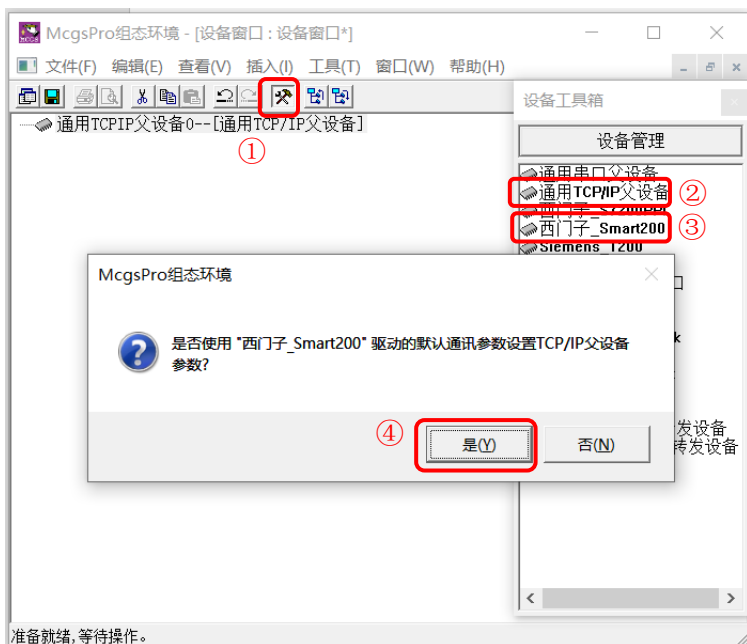


图 3.5-4 添加 Smart200 驱动

● 增加设备通道

- ① 双击添加好的 Smart200 驱动，弹出【设备编辑窗口】界面
- ② 点击右侧的【删除全部通道】按钮，删除默认通道

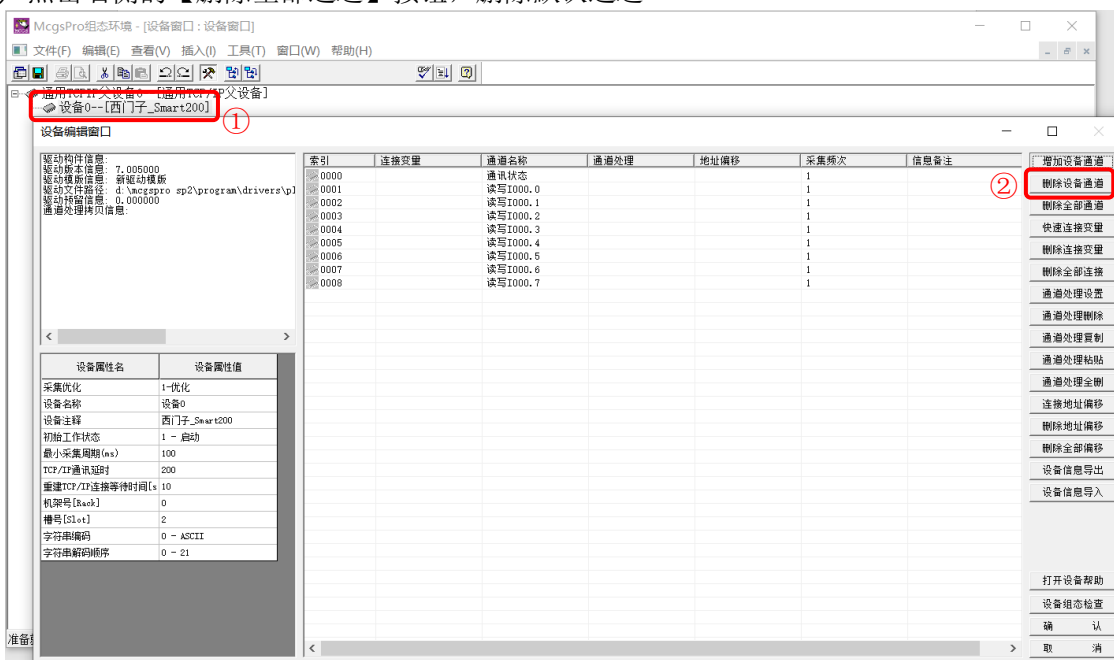


图 3.5-5 设备编辑窗口界面

- ③ 点击【添加设备通道】按钮，弹出添加设备通道界面
- ④ 选择【通道类型】为【I 输入继电器】
- ⑤ 设置【数值类型】为【通道的第 00 位】
- ⑥ 设置【通道地址】为【0】(即通道开始的地址)
- ⑦ 设置【通道个数】为【4】
- ⑧ 设置【读写方式】为【只读】，点击【确认】创建通道



图 3.5-6 添加设备通道之一

- ⑨ 用同样的方式建立 4 个【通道类型】为【Q 输出继电器】，【数值类型】为【通道的第 00 位】，【通道地址】为【0】，【通道个数】为【4】，设置【读写方式】为【只写】的通道



图 3.5-7 添加设备通道之二

- ⑩ 再建立 1 个【通道类型】为【V 数据寄存器】，【数值类型】为【32 位 浮点数】，【通道地址】为【0】，【通道个数】为【1】的通道



图 3.5-8 添加设备通道之三

- ⑪ 创建好通道后，【设备编辑窗口】如图 3.5-9 所示。点击【确认】保存及关闭窗口。

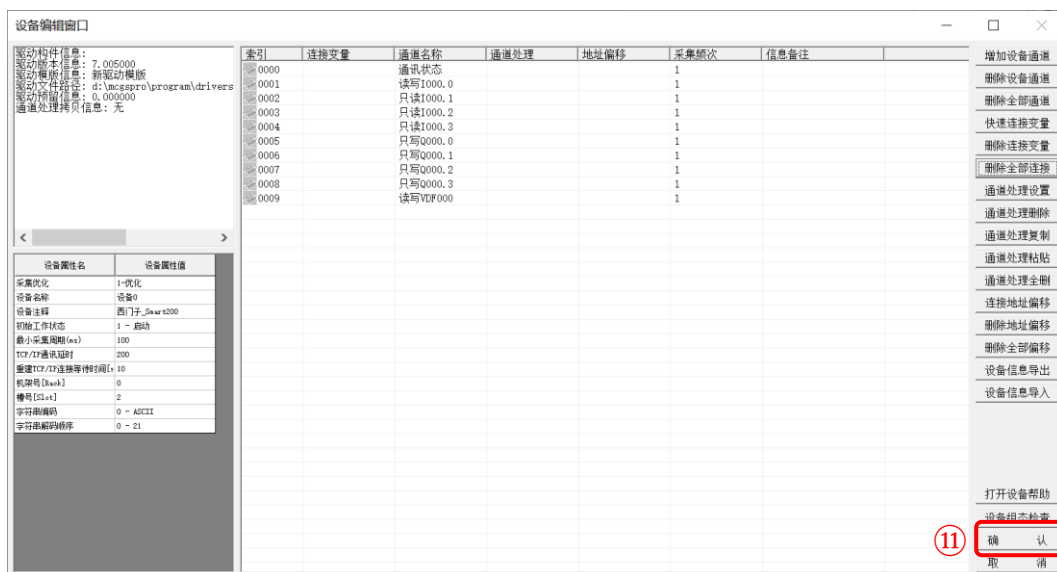


图 3.5-9 通道添加完成

● 设置父设备通讯参数

- ① 双击【通用 TCP/IP 父设备 0—[通用 TCP/IP 父设备]】，弹出【通用 TCP/IP 设备属性编辑】界面。
- ② 在【基本属性】功能页中，设置 TCP 与 PLC 的通讯参数。

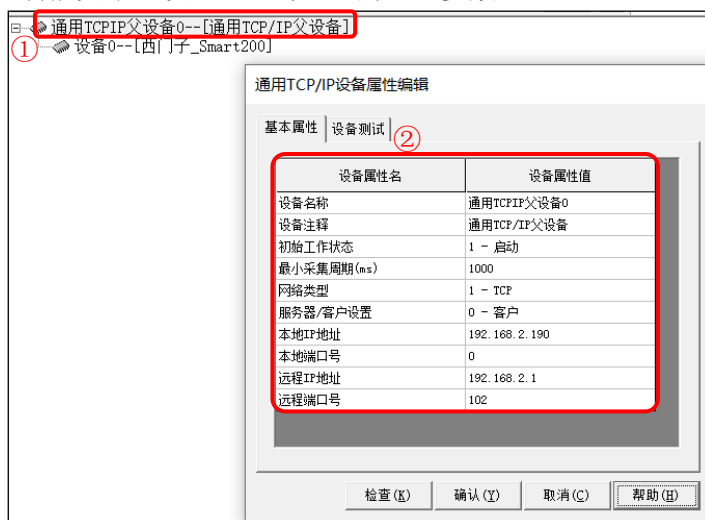


图 3.5-10 通用父设备设置界面

3.5.3 实时数据库组态

● 在实时数据库中新增 9 个整数变量：

- ① 选择【工作台】里的【实时数据库】
- ② 点击右侧的【新增对象】，此时会在左侧系统默认自带的四个变量下方，新增一个变量【Data1】
- ③ 双击【Data1】，弹出【数据对象属性设置】界面
- ④ 在其【基本属性】功能页中，修改【对象名称】为【进水泵启停状态】
- ⑤ 选择【对象类型】为【整数】

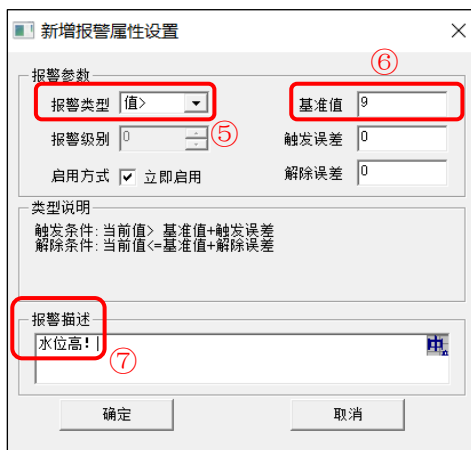


图 3.5-13 新增报警属性设置界面

- ⑧ 用同样的方法，为变量【液位值】追加一条【报警类型】为【值<】，【基准值】为【1】，【报警描述】为【水位低!】的报警。

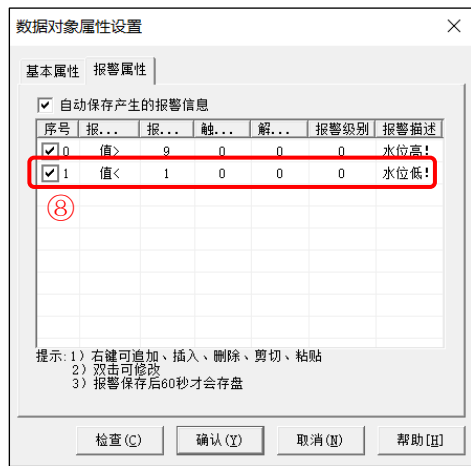


图 3.5-14 追加第二条报警属性

3.5.4 将变量和通道相连接

- ① 回到设备窗口，双击 Smart200 驱动，弹出【设备编辑窗口】界面
- ② 在通道【读写 I000.0】的左侧连接变量列右击鼠标，弹出【变量选择】界面
- ③ 在界面下方的列表中选择【进水泵启停状态】
- ④ 点击【确认】关闭界面

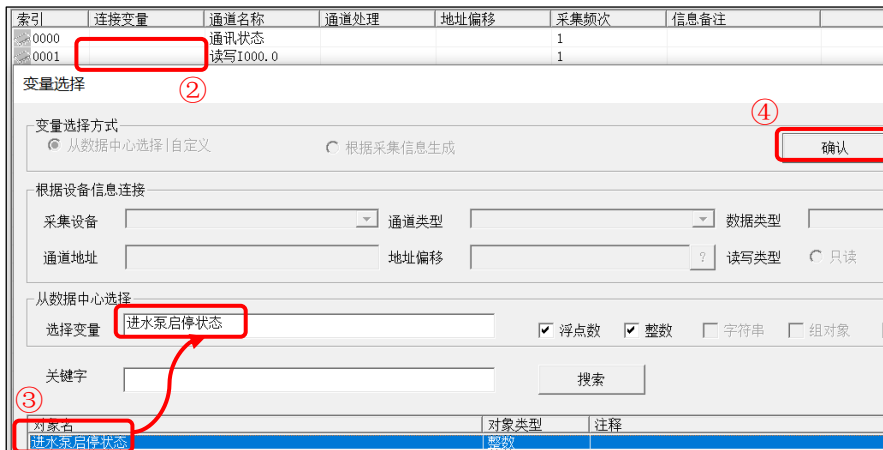


图 3.5-15 将变量和通道相连接一

⑤ 用同样的办法将剩下的通道和变量一一连接，连接完成后如图 3.5-16 所示。

索引	连接变量	通道名称
0000		通讯状态
0001	进水泵启停状态	读写I000.0
0002	出水泵启停状态	只读I000.1
0003	进水阀开闭状态	只读I000.2
0004	出水阀开闭状态	只读I000.3
0005	启停进水泵	只写Q000.0
0006	启停出水泵	只写Q000.1
0007	开关进水阀	只写Q000.2
0008	开关出水阀	只写Q000.3
0009	液位值	读写VDF000

图 3.5-16 将变量和通道相连接二

3.5.5 用户窗口组态

- 进入【动画组态窗口】开始组态画面
 - ① 点击菜单栏的【窗口】按钮
 - ② 在弹出的下拉菜单中选择【工作台：xxx】，进入工作台界面
 - ③ 选择【用户窗口】界面
 - ④ 双击【窗口0】，进入【动画组态窗口】界面



图 3.5-17 进入用户窗口

- 设置窗口画面标题

- ① 点击【工具箱】中的  图标，将鼠标在画面上拖拽出标签构件到合适的位置和大小。【工具箱】可以通过工具栏中的  图标进行打开或者关闭

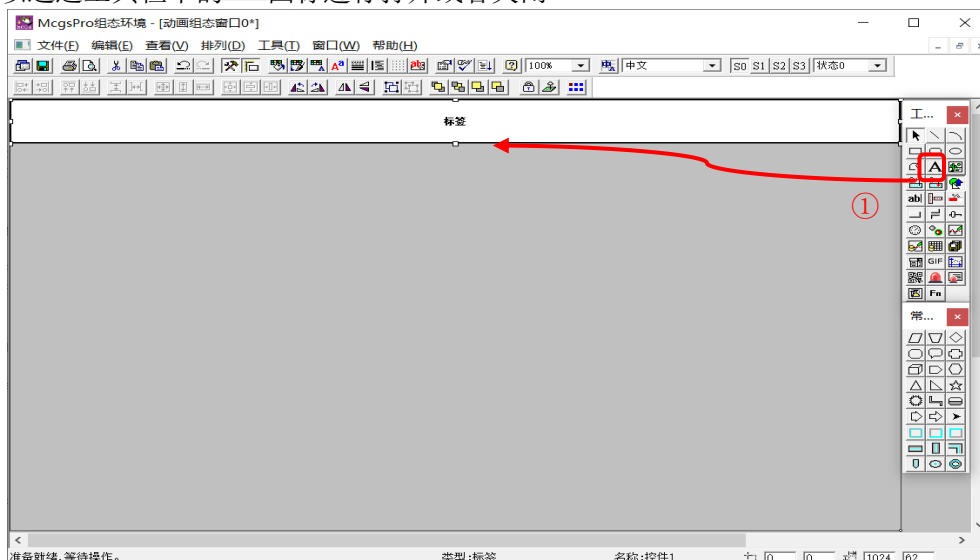


图 3.5-18 设置窗口画面标题

- ② 双击标签构件，弹出【标签动画组态属性设置】界面
- ③ 点击【填充颜色】下拉菜单，选择需要的颜色，本样例使用【青色】



图 3.5-19 设置标签填充颜色

- ④ 设置【字符颜色】为【白色】
- ⑤ 设置字体为【宋体】【粗体】【小一】

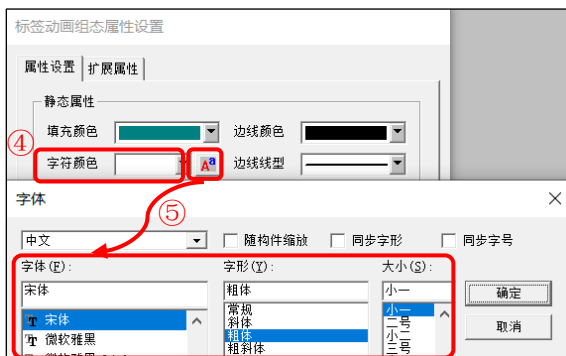


图 3.5-20 设置标签字体

- ⑥ 设置线型为最细



图 3.5-21 设置标签边线线型

- ⑦ 选择【扩展属性】功能页
- ⑧ 在【文本内容输入框】中输入“液位保持系统”



图 3.5-22 设置标签文本显示内容

● 绘制蓄水池和水管

- ① 点击【工具箱】中的  图标，打开【常用符号】工具箱
- ② 选择【常用符号】工具箱中的  和  图标，拖拽到合适的位置和大小

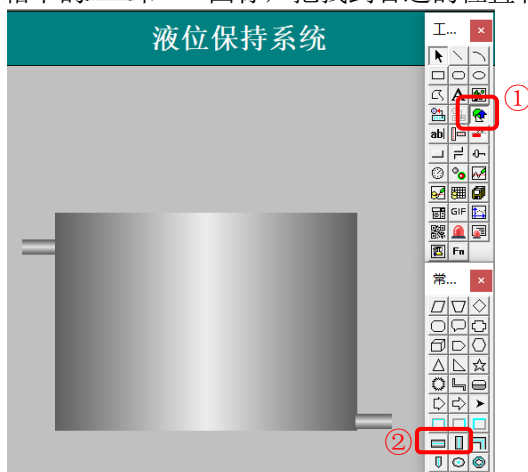


图 3.5-23 绘制蓄水罐和水管

● 从图库中添加【电动阀】

- ① 点击【工具箱】中的【插入元件】，弹出【元件图库管理】界面
- ② 在【类型】中选择【公共图库】
- ③ 在下方的树形表中，选中【阀】
- ④ 选择需要的阀门，本样例选择【阀 52】，点击确定
- ⑤ 【阀】出现在画面左上角，拖拽到合适的位置

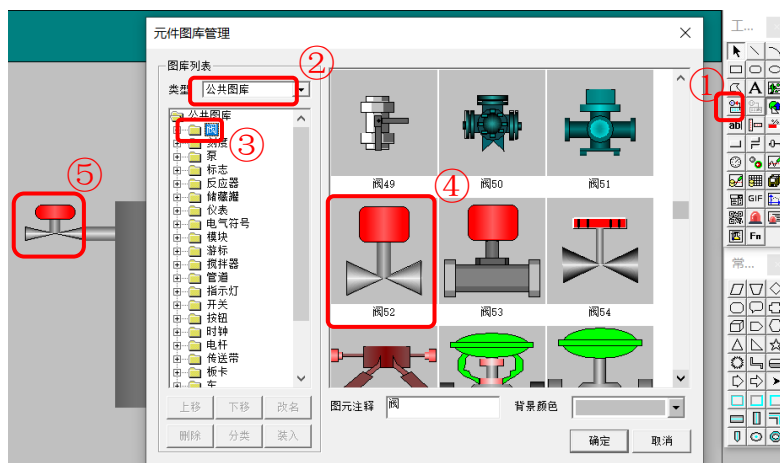


图 3.5-24 从图库中添加阀门

- 用同样的方法从图库中再添加一个【电动阀】以及其他元件【公司 LOGO】【水泵】和【液位计】

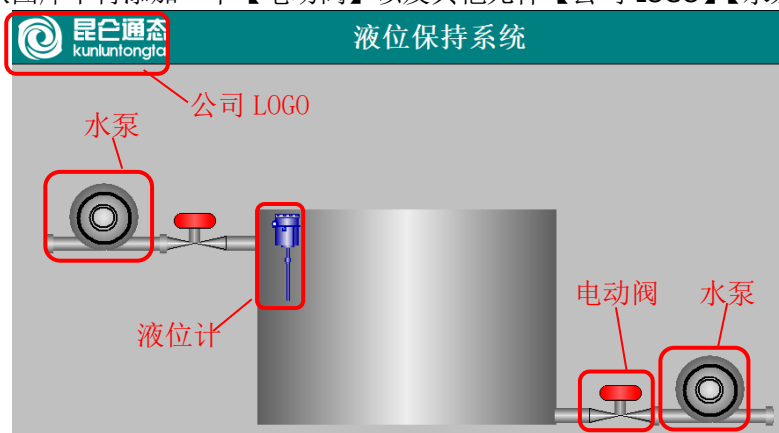


图 3.5-25 从图库中添加其他元件

- 【报警条】【箭头】【百分比填充】和【弯管】构件的绘制方法同绘制蓄水池和水管，其中【箭头】是由【直线】和【三角箭头】组合而成

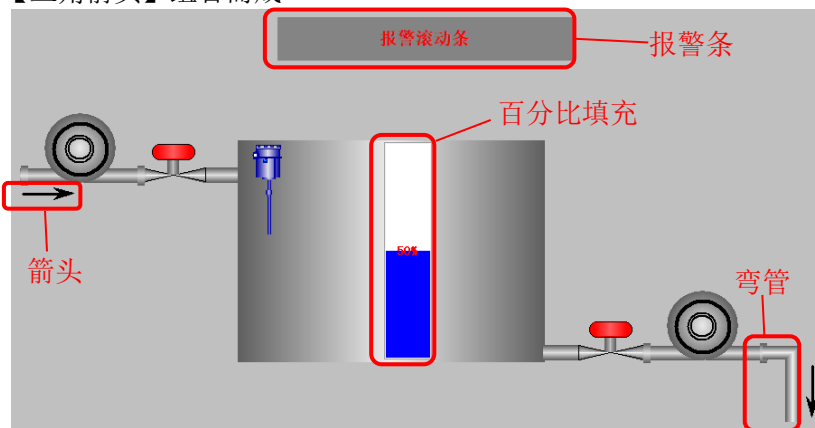


图 3.5-26 添加其他构件完善画面

- 关联【百分比填充】构件的【操作属性】表达式为【液位值】，设置其 0%对应的值为【0】，100%对应的值为【10】



图 3.5-27 设置百分比填充构件操作属性

● 设置【报警条】构件属性

- ① 双击【报警条】构件，弹出【报警条属性设置】界面
- ② 不对【基本属性】功能页做修改，当报警条不关联【报警对象】时，表示报警条显示所有的报警信息



图 3.5-28 报警条基本属性功能页

- ③ 选择【显示格式】功能页
- ④ 勾选【显示内容】中所有的显示
- ⑤ 设置【日期时间】格式

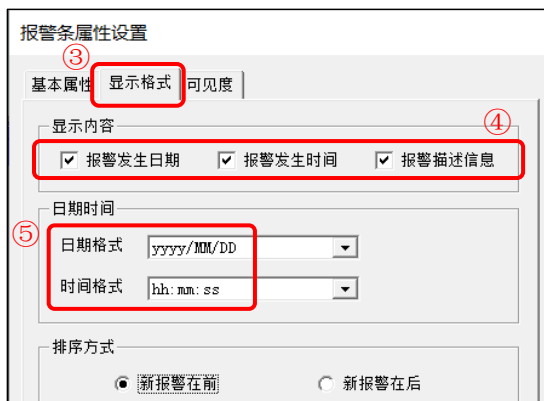


图 3.5-29 设置报警条显示格式功能页

- 使用【标签】构件给图中的构件和元件标注名称。【标签】构件的绘制方法同设置窗口画面标题，选

择不同的【填充颜色】和【边线颜色】绘制不同的效果

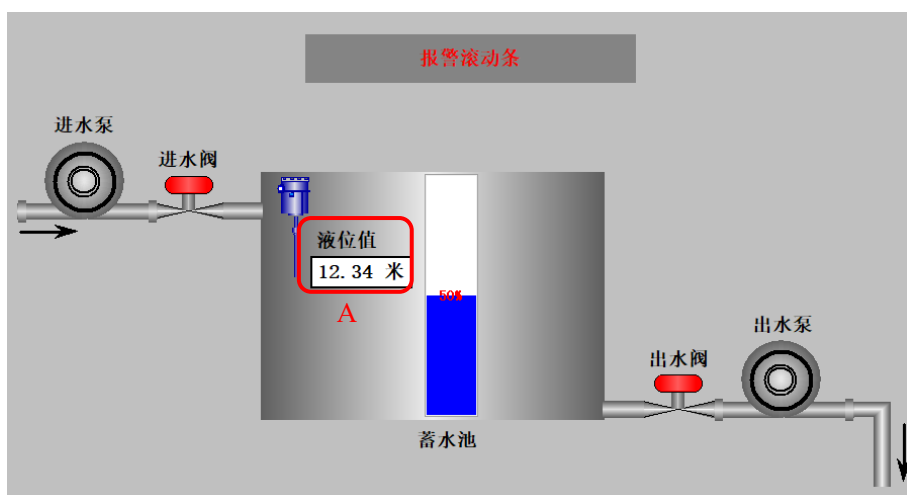


图 3.5-30 添加标签构件

其中,标签构件 A, 需要设置显示输出表达式, 使其能实时显示储水罐的液位值。

- ① 双击上图中的标签构件 A, 弹出【标签动画组态属性设置】界面
- ② 勾选【属性设置】功能页中,【输入输出连接】下【显示输出】前的复选框, 出现【显示输出】功能页
- ③ 选择【显示输出】功能页
- ④ 关联表达式为【液位值】
- ⑤ 设置【单位】为【米】
- ⑥ 选择【输出值类型】为【数值量输出】, 确认

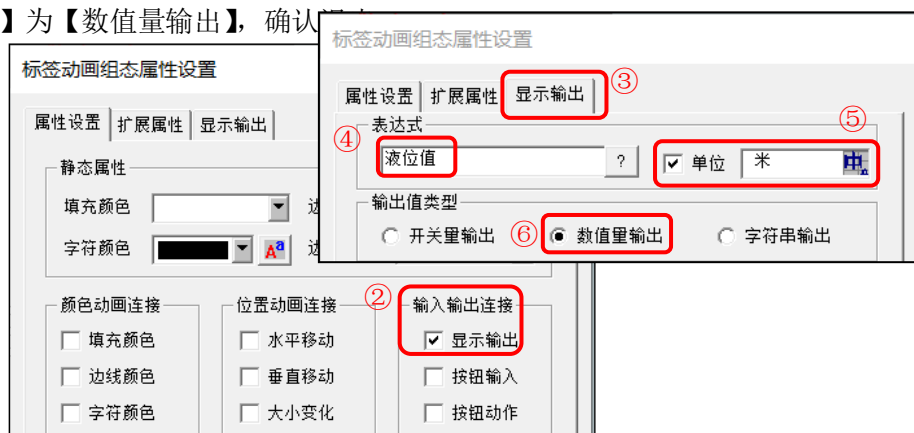


图 3.5-31 设置标签 A 显示输出

- 将【进水阀】与变量关联
 - ① 双击【进水阀】元件, 弹出【单元属性设置】界面
 - ② 在【变量列表】属性功能页中, 选中【数据操作对象】, 点击右侧出现的[?], 弹出【变量选择】界面
 - ③ 双击变量【开关进水阀】进行选择
 - ④ 用同样的方法设置【表达式】为【进水阀开闭状态】
 - ⑤ 也可以在【变量关联】处双击进入编辑状态, 然后直接输入想要关联的变量的名称。

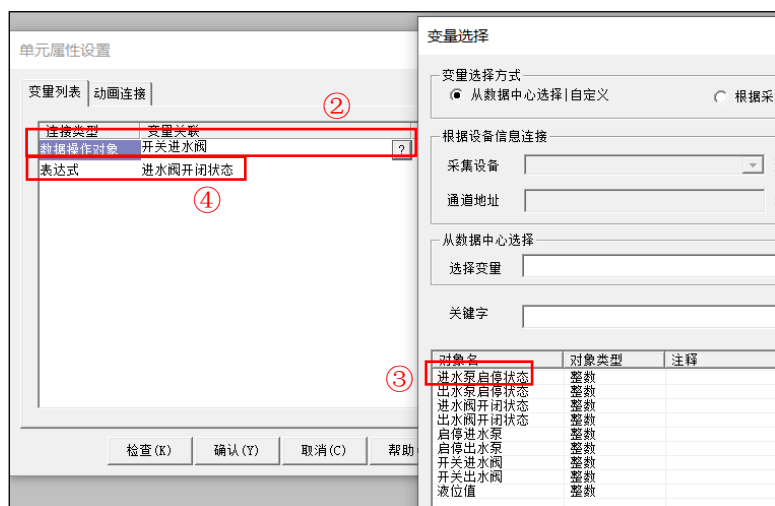


图 3.5-32 设置进水阀关联变量

- 用同样的方法将【出水阀】与变量关联
 - ① 出水阀的【数据操作对象】关联变量【开关出水阀】
 - ② 出水阀的【表达式】关联变量【出水阀开闭状态】
- 将【进水泵】与变量关联
 - ① 双击【进水泵】，弹出【动画组态属性设置】界面
 - ② 勾选【基本属性】功能页中【颜色动画连接】中的【填充颜色】前的复选框，出现【填充颜色】功能页
 - ③ 选中【填充颜色】功能页，设置表达式为【进水泵启停状态】，【填充颜色连接】中，【分段点 0】的颜色设置为【红色】，【分段点 1】的颜色设置为【绿色】
 - ④ 勾选【基本属性】功能页中【输入输出连接】中的【按钮动作】前的复选框，出现【按钮动作】功能页
 - ⑤ 选【按钮动作】功能页，勾选【数据对象值操作】前的复选框，在第一个下拉菜单中选择【取反】，在第二个输入框中关联变量【启停进水泵】，点击【确定】按钮

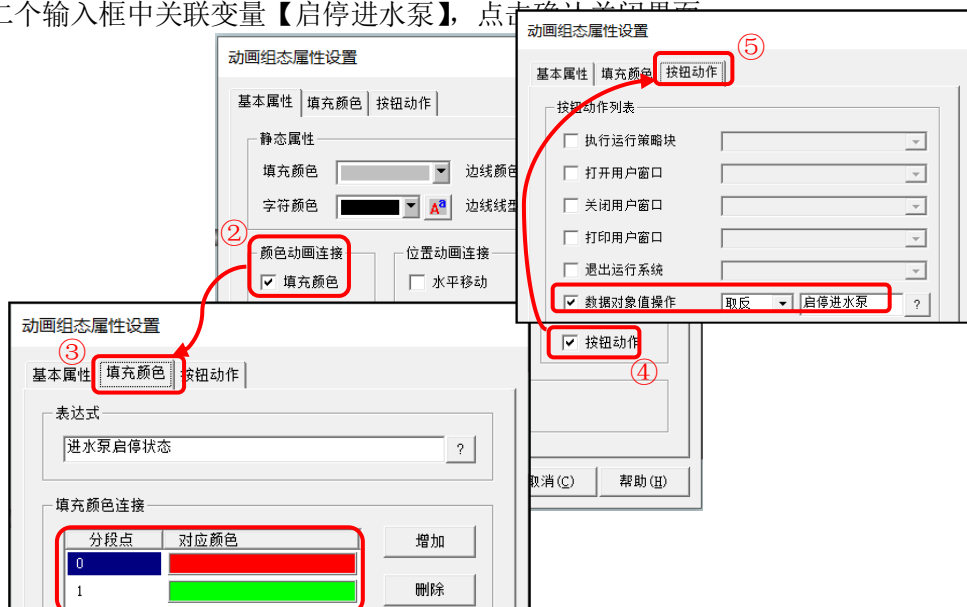


图 3.5-33 设置进水阀填充颜色和按钮动作

- 用同样的方法设置元件【出水泵】中，【填充颜色】关联的表达式为【出水泵启停状态】，【按钮动作】中【数据对象值操作】为【取反】【启停出水泵】

3.5.6 模拟运行

在调试阶段，我们可以通过电脑和 PLC 直接连接，实现模拟运行在线调试。如果是串口通讯，在父设备中设置串口号即为电脑对应的串口号。本样例是网口通讯，电脑的 IP 地址设置为【设备窗口】中父设备的本地 IP。

模拟运行前需要先保存工程。

- ① 首先用网线将电脑与 PLC 连接，然后点击工具栏中的图标，弹出【下载配置】界面
- ② 选择【运行方式】为【模拟】
- ③ 点击【工程下载】按钮
- ④ 下载成功后，点击【启动运行】按钮，弹出模拟运行窗口

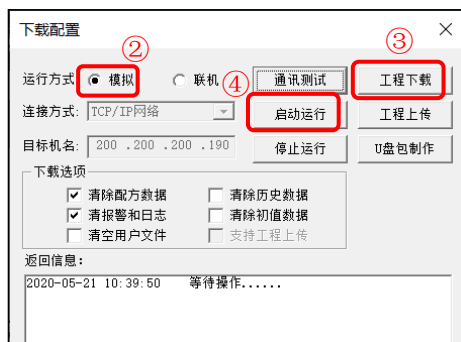


图 3.5-34 模拟运行下载设置界面

模拟运行窗口如图 3.5-35 所示：

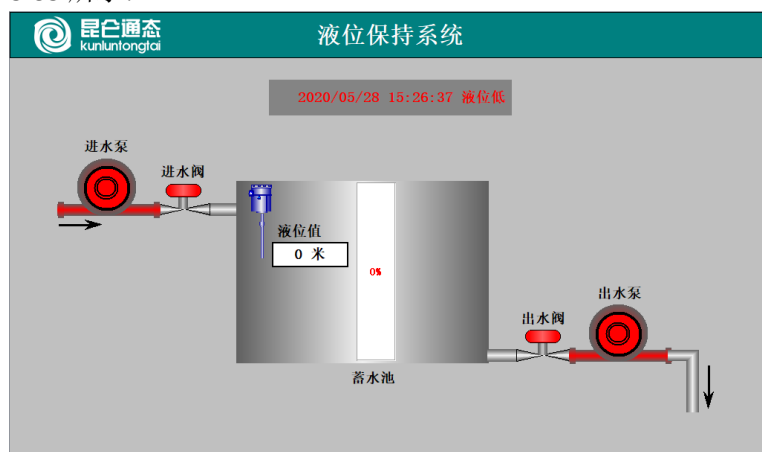


图 3.5-35 模拟运行窗口

3.5.7 下载工程到 TPC

工程组态完成后，我们将工程下载到 TPC 中。并通过网线连接 TPC 和 PLC，使 TPC 和 PLC 进行通讯，从而使 TPC 能显示和控制 PLC 中的数据。本样例我们使用 TCP/IP 下载的方式对工程进行下载。

- ① 将电脑与 TPC 用网线连接，然后点击工具栏中的图标，弹出【下载配置】界面
- ② 选择【运行方式】为【联机】
- ③ 选择【连接方式】为【TCP/IP】
- ④ 设置【目标机名】，该 IP 地址必须与 TPC 的 IP 地址一致，电脑的 IP 地址需与 TPC 在同一个网段。TPC 的 IP 地址设置方式将在后续章节：[硬件篇 2.1 TPC 系统功能讲解](#)。
- ⑤ 按照实际需求对【下载选项】进行勾选
- ⑥ 点击【工程下载】按钮
- ⑦ 下载成功后，用网线将 TPC 与 PLC 相连。点击【启动运行】按钮，TPC 中下载好的工程开始运行

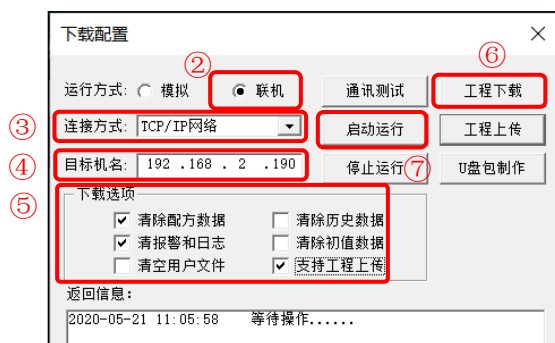


图 3.5-36 联机运行下载设置界面

※进阶篇※

第 1 章 McgsPro 软件功能

本章简要介绍 McgsPro 的组态结构。

1.1 界面布局

打开 McgsPro 工程后，启动界面如图 1.1-1 所示：

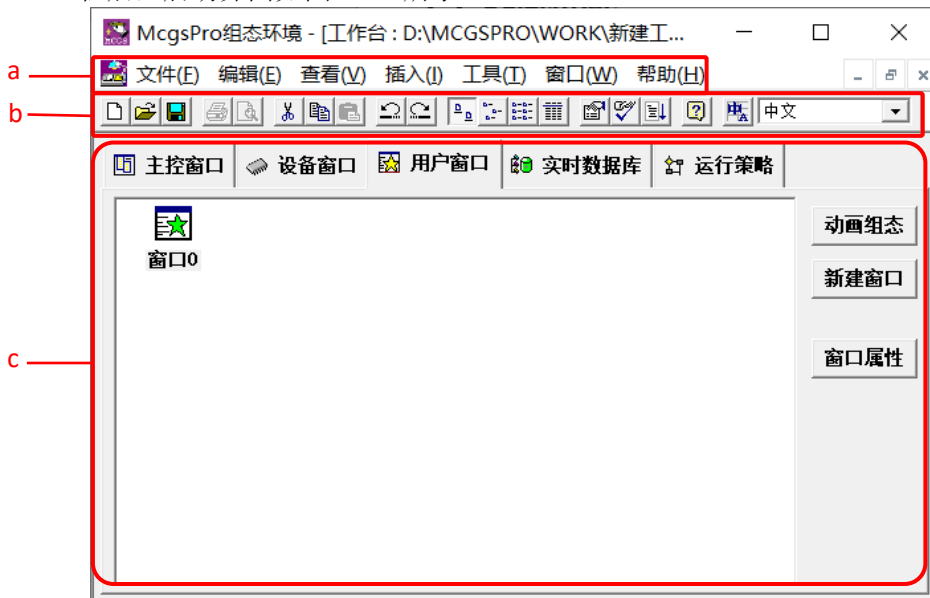


图 1.1-1 McgsPro 组态软件界面

a.菜单栏；b.工具栏；c.工作台（工作区）

其中，菜单栏和工具栏的选项内容，会根据所处窗口的不同有细微的差别。

1.2 菜单栏

1.2.1 文件菜单

文件(F)	编辑(E)	查看(V)	排列(D)
新建工程(N)			Ctrl+N
打开工程(O)			Ctrl+O
关闭窗口			
保存窗口			Ctrl+S
工程另存为(A)...			
自动保存设置...			
打印设置(U)...			
打印预览(V)...			
打印(P)...			Ctrl+P
工程设置...			
1 新建工程0.MCP			
2 新建工程5.MCP			
3 新建工程3.MCP			
4 新建工程2.MCP			
退出(X)			

图 1.2-1 文件菜单界面

- **【自动保存设置】**：设置是否启用自动保存功能，默认打开。保存时间间隔默认 3 分钟，时间可设置，范围为 3~120 分钟。
- **【打印预览】**：预览画面打印效果，只在动画组态窗口有效。

- **【打印】**: 设置打印机名称和属性、打印范围、打印份数。只在动画组态窗口有效。
- **【工程设置】**: 修改 HMI 配置（即 TPC 型号）和组态配置（包括网格相关设置、构件风格、工程选择、用户界面是否随分辨率和工程选择变化）。
- 最近打开的四个工程，选择工程名可退出当前工程并打开该工程。
- **【退出】**: 退出软件。

1.2.2 编辑菜单



图 1.2-2 编辑菜单界面

- **【属性】**: 打开当前选中项的属性设置界面。在动画组态窗口中，鼠标未选中任何项时，会打开当前窗口属性设置界面。
- **【事件】**: 打开当前选中项的事件组态界面。在动画组态窗口中，鼠标未选中任何项时，会打开当前窗口事件组态界面。
- **【插入元件】**: 从对象元件库中读取存盘的图形对象。
- **【保存元件】**: 把当前用户窗口中选中的图形对象存入对象元件库中。

1.2.3 查看菜单



图 1.2-3 查看菜单界面

- **【工作台】**: 进入工作台的其他界面。
- **【工具条】**: 打开或关闭工具条。
- **【状态条】**: 打开或关闭状态条。
- **【全屏显示】**: 进入全屏显示，按“ESC”键退出。
- **【视图缩放】**: 设置动画组态窗口的缩放比例为 50%~400%。
- **【绘图工具条】**: 打开或关闭绘图工具条。
- **【绘图编辑条】**: 打开或关闭绘图编辑条。
- **【辅助提示】**: 设置是否在构件左上方显示“构件名称”和“构件引用变量”。

1.2.4 排列菜单



图 1.2-4 排列菜单界面

【排列】菜单仅在动画组态窗口使用，在其他窗口中处于隐藏状态。

- **【构成图符】**：将多个图元对象组合在一起，可以构成一个图符对象。图元和图符对象也可以构成图符。构成的图符可以看作是一个整体而存在的，可以等比例缩放，可以设置事件。
- **【分解图符】**：将合成的图符分解开。
- **【合成单元】**：将图元、图符、构件合成单元，合成单元可以等比例缩放，不能设置事件。
- **【分解单元】**：将合成的单元分解开。
- **【最前面】**：把被选中的图形对象放在所有对象前。
- **【最后面】**：把被选中的图形对象放在所有对象后。
- **【前一层】**：把被选中的图形对象向前移一层。
- **【后一层】**：把被选中的图形对象向后移一层。
- **【对齐】**：当选中多个图形对象时，可以把当前对象（最后选中的对象）作为基准，对被选中的多个图形对象进行相对位置和大小关系调整：
 - **【左对齐】**：以当前对象为基准，左边界对齐；
 - **【右对齐】**：以当前对象为基准，右边界对齐；
 - **【上对齐】**：以当前对象为基准，上边界对齐；
 - **【下对齐】**：以当前对象为基准，下边界对齐；
 - **【纵向等间距】**：被选中的多个图形对象沿 Y 方向等距离分布；
 - **【横向等间距】**：被选中的多个图形对象沿 X 方向等距离分布；
 - **【图元等高宽】**：所有选中对象的高度和宽度相等；
 - **【图元等高】**：所有选中对象的高度相等；
 - **【图元等宽】**：所有选中对象的宽度相等；
 - **【中心对中】**：所有选中对象的中心点重合；
 - **【纵向对中】**：所有选中对象的中心点 X 坐标相等；
 - **【横向对中】**：所有选中对象的中心点 Y 坐标相等。
- **【旋转】**：
 - **【左旋 90 度】**：把被选中的图形对象左旋 90 度；
 - **【右旋 90 度】**：把被选中的图形对象右旋 90 度；
 - **【左右镜像】**：把被选中的图形对象沿 X 方向翻转；
 - **【上下镜像】**：把被选中的图形对象沿 Y 方向翻转。
- **【锁定】**：固定对象的位置和大小，使用户不能对其进行移动和修改，避免编辑时，因误操作而破坏组态完好的图形。
- **【固化】**：图形对象被固化后，用户就不能选中它，从而也不能对其进行各种编辑工作。在组态过程中，一般把作为背景用途的图形对象加以固化，以免影响其它图形对象的编辑工作。

- **【激活】**：将固化的图形对象激活。
- **【多重复制】**：快速创建一批同质不同地址(变量)的动画构件。具体设置方法将在后续章节：进阶篇-第6章 多重复制应用实例中讲解。

1.2.5 工具菜单

组态检查	F4
下载工程	F5
上传工程	F6
模拟运行	F7
组态对象浏览	
数据对象名替换	
文本查找替换	
使用计数检查	
清除未用变量	
TPC文件操作	
工程运行期限	
工程文件压缩	
工程密码设置	
工程字体管理	
工程文件保护	
元件图库管理	
动画构件管理	
用户权限管理	
配方组态设计	
报警统一配置	
多语言	
操作日志设置	

图 1.2-5 工具菜单界面

- **【组态检查】**：进行工程正确性检查。
- **【下载工程】**：进入下载配置界面，可以设置运行方式、连接方式、目标机名，进行通讯测试、工程下载、启动运行等操作。
- **【上传工程】**：进入上传工程界面，可以设置连接方式、目标地址，进行通信测试、开始上传、退出上传等操作。
- **【模拟运行】**：将组态好的工程在 PC 上进行模拟运行，用户可根据模拟运行效果增减其中内容。它帮助用户测试组态工程的设计和构造。
- **【组态对象浏览】**：展示工程中设备窗口、用户窗口、变量、用户策略、配方以及引用和被引用的拓扑关系。在此基础上帮助用户快速查找包含某关键字的对象，并在对象树上快速跳转，且可以定位指定对象所在的具体位置。
- **【数据对象名替换】**：替换所有出现该对象名称的关联对象、表达式、实时数据库变量、脚本等处的该对象名为另一个对象名。替换和被替换的变量名都不能为空，若被替换的变量不存在，则会提示“指定的变量名不存在”。
- **【文本查找替换】**：定位用户窗口、构件、策略中的表达式或脚本中指定文本所在位置，同时附带文本替换的功能。如快速定位“!OpenSubWnd”函数在哪些地方使用过。
- **【使用计数检查】**：统计当前工程实时数据库中的变量。

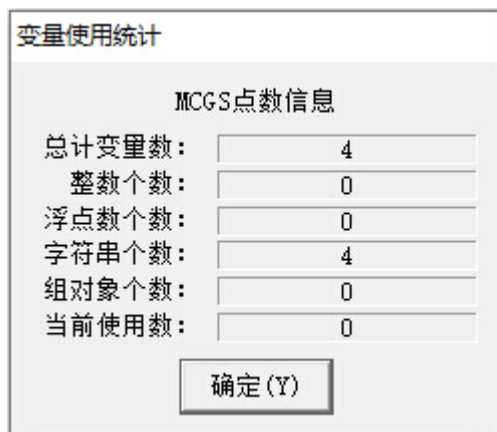


图 1.2-6 变量使用统计界面

- **【清除未用变量】:** 删除“实时数据库”中未使用的变量，减小工程体积，使工程更整洁。
- **【TPC 文件操作】:** 提供本地文件与用户文件区文件的直接操作。界面主要展示网络连接状态、本地目录文件及文件夹列表、用户文件区文件及文件夹列表、传输过程进度及日志信息等。

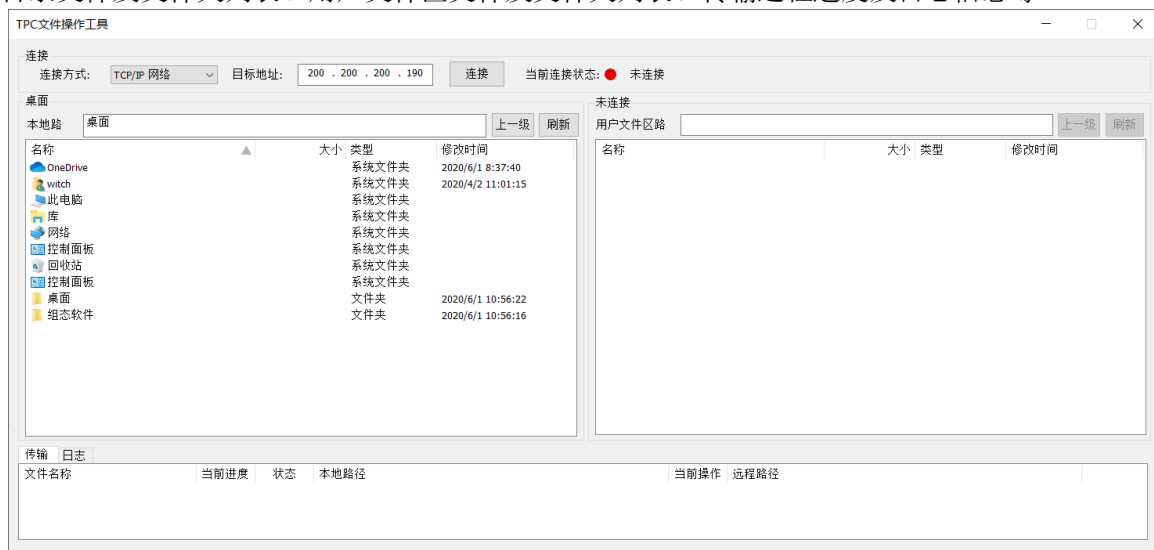


图 1.2-7 TPC 文件操作工具界面

- **【工程运行期限】:** 启用了分期功能，当工程用户无法一次结清工程款时，启用此功能，可协助客户收取工程尾款，防止业主赖账，保护客户权益。
- **【工程文件压缩】:** 如果工程中使用了大量 bmp 格式图片，使得工程体积很大，可使用本功能减小工程体积。它有两种压缩方式，均不会影响实际显示效果：
 - ①将工程中所有位图裁剪为实际显示大小，当图片实际大小大于显示大小时，该操作可以减少运行环境的内存占用和磁盘占用；
 - ②将工程中所有 BMP 图片转换为 JPG 格式，该操作会减少运行时的磁盘占用。
- **【工程密码设置】:** 当使用 McgsPro 组态软件来打开设置了工程密码的工程时，首先弹出输入框要求输入工程的密码，如果密码不正确则不能打开该工程，从而起到保护劳动成果的作用。
- **【工程字体管理】:** 查看工程中使用的字体并可以对使用的字体进行更改。
- **【工程文件保护】:** 将工程限制在指定的触摸屏上使用。
- **【元件图库管理】:** 管理 McgsPro 组态软件内的构件风格、构件模版、用户元件、标准图片。所有可组态背景图的构件，均支持将图库中的构成图符和标准图片作为背景显示。构件模版是构件从工具箱拖出来的默认样式。用户元件是用户保存的图库资源。标准图片是 bmp、jpg、png、ico、svg 格式图片。
- **【动画构件管理】:** 增加和删除动画工具箱中的动画构件。
- **【用户权限管理】:** 编辑用户和用户组。在用户权限设置窗口中，把对应的用户或用户组选中，则只有该用户或该用户组成员能对该项工作进行操作。

- **【配方组态设计】**：进行配方组以及配方的新建、编辑、删除等操作。对话框左侧是所有配方组列表，右侧可以编辑配方组相关属性，主要编辑配方组各个元素关联的变量以及变量对应的标题。

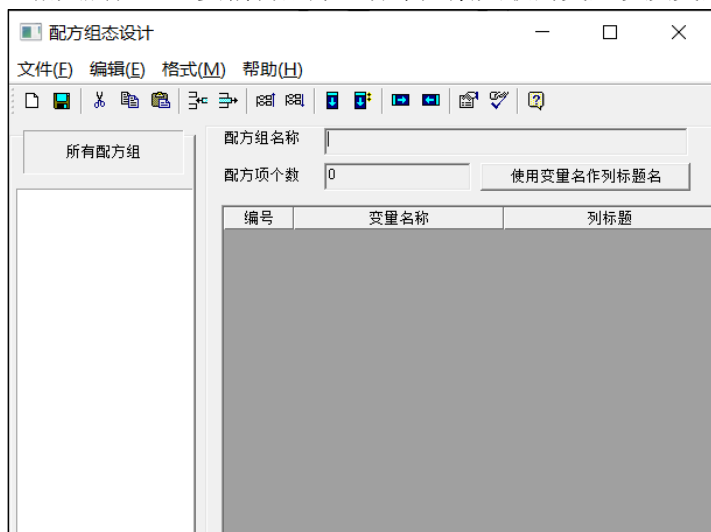


图 1.2-8 配方组态设计界面

- **【报警统一配置】**：统一对报警属性进行配置。如果已在实时数据库中设置过数据对象报警属性，则**【报警统一配置】**界面的表格中将会显示已经设置过的所有报警信息，否则表格中没有报警信息。在变量名下方的行双击，可添加一行报警，然后在该行进行关联变量，修改变量类型、报警类型、基准值、报警描述等设置。

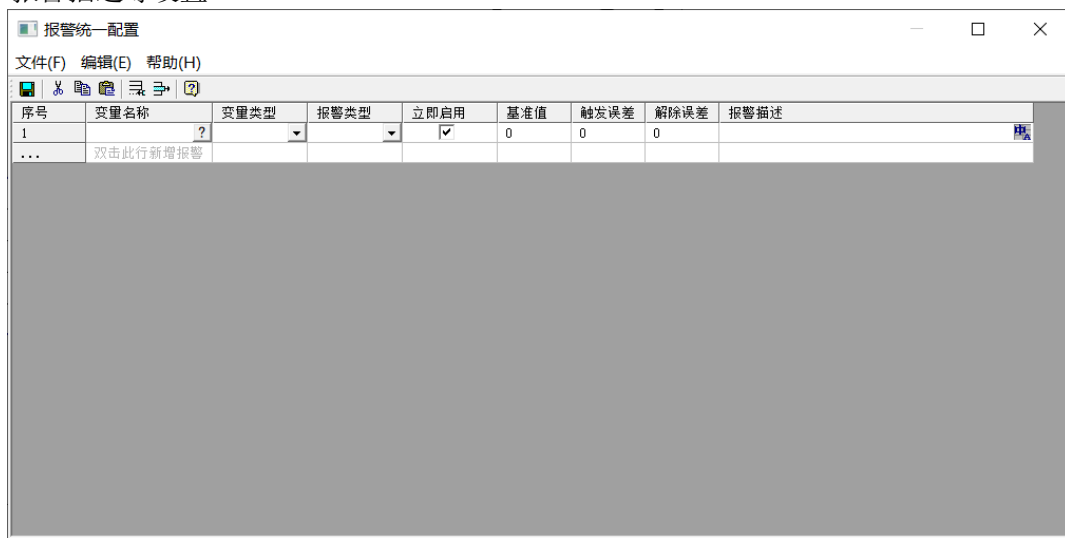


图 1.2-9 报警统一配置界面

- **【多语言】**：当同个工程在多个国家使用时，项目就需要使用多种语言来创建。通过多语言配置，可将工程中显示的文字在数个语言间切换。
- **【操作日志设置】**：在这里开启或关闭操作日志记录功能。在启用该功能后，可以在下方启停指定构件的日志、对日志描述文本进行编辑，日志描述文本支持多语言。

1.2.6 窗口菜单



图 1.2-10 窗口菜单界面

- 设置各个窗口的位置关系为【层叠】【水平平铺】【垂直平铺】。
- 对已打开的窗口进行选择

1.2.7 帮助菜单

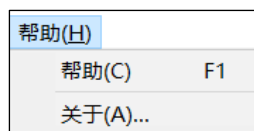


图 1.2-11 帮助菜单界面

- 【帮助】：进入 McgsPro 帮助系统。
- 【关于】：提示 McgsPro 版本编号，版权所属方及联系方式。

1.3 工具栏

工具栏中是常用功能的快捷图标。将鼠标悬停在某个快捷图标上，可出现文字提示，点击该快捷图标，实现提示对应功能。工具条在不同窗口会有个性化差异。

1.3.1 工作台窗口工具条

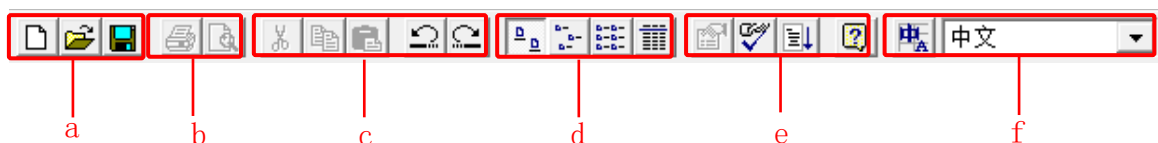


图 1.3-1 工作台窗口工具条

- a.新建/打开/保存
- b.打印/打印预览
- c.剪切/拷贝/粘贴/撤销/恢复
- d.大图标/小图标/列表/详细信息
- e.显示属性/组态检查/下载运行/帮助
- f.多语言/当前语言

1.3.2 设备窗口工具条

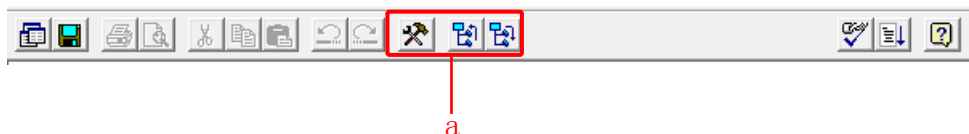


图 1.3-2 工作台窗口工具条

- a.（设备）工具箱/向上移动/向下移动

1.3.3 动画组态窗口工具条

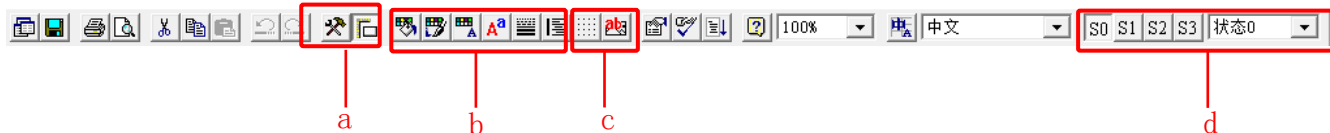


图 1.3-3 动画组态窗口工具条

- a.工具箱/编辑条
- b.填充颜色/边线颜色/字符颜色/字体/线型/对齐方式
- c.背景网格/辅助提示
- d.分段点选择/当前状态

1.3.4 动画组态窗口绘图编辑条

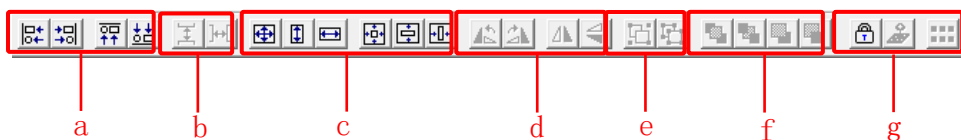


图 1.3-4 动画组态窗口绘图编辑条

- a. 左边界对齐/右边界对齐/顶边界对齐/底边界对齐
- b. 纵向等间距/横向等间距
- c. 等高宽/等高/等宽/中心对齐/纵向对中/横向对中
- d. 左旋 90 度/右旋 90 度/Y 反转/X 反转
- e. 构成图符/分解图符
- f. 置于最前面/置于最后面/向前一层/向后一层
- g. 锁定 or 解锁/固化/多重复制

1.3.5 脚本程序窗口工具条

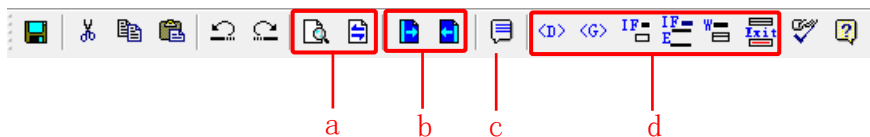


图 1.3-5 脚本程序窗口工具条

- a. 查找/替换
- b. 缩进/退格
- c. 注释
- d. 声明局部变量/声明数据对象/语句块 IF~THEN/语句块 IF~ELSE/语句块 WHILE/语句块 EXIT

1.4 工作台

工作台是我们进入工程之后看到的第一个窗口，它包含 5 个窗口页：主控窗口、设备窗口、用户窗口、实时数据库、运行策略。工作台相当于一个大的容器，可以放置一个主控窗口、一个设备窗口和多个用户窗口，负责这些窗口的管理和调度，并调度用户策略的运行。

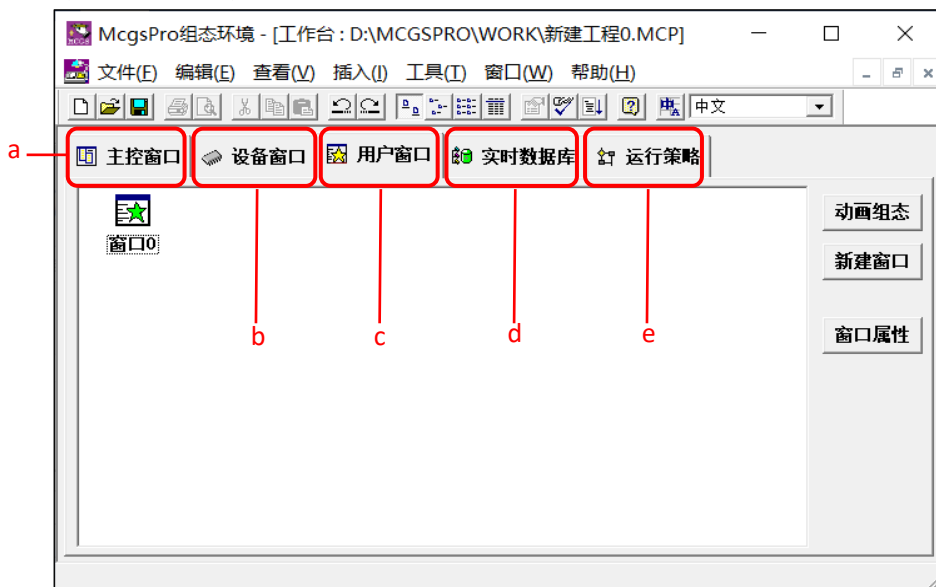


图 1.4-1 McgsPro 组态软件界面

a. 主控窗口：构造了应用系统的主框架。用于对整个工程相关的参数进行配置，可设置封面窗口、运行工程的权限、启动画面、内存画面、磁盘预留空间等。

b. 设备窗口：是应用系统与外部设备联系的媒介。专门用来放置不同类型的设备构件，实现对外部设备的操作和控制。设备窗口通过设备构件把外部设备的数据采集进来，送入实时数据库，或把实时数据库中的数据输出到外部设备。

c.用户窗口：实现了应用系统数据和流程的“可视化”。工程里所有可视化的界面都是在用户窗口里面构建的。用户窗口中可以放置三种不同类型的图形对象：图元、图符和动画构件。通过在用户窗口内放置不同的图形对象，用户可以构造各种复杂的图形界面，用不同的方式实现数据和流程的“可视化”。

d.实时数据库：是应用系统的核心。实时数据库相当于一个数据处理中心，同时也起到公共数据交换区的作用。

e.运行策略：是对应用系统运行流程实现有效控制的手段。运行策略本身是系统提供的一个框架，其里面放置由策略条件构件和策略构件组成的“策略行”，通过对运行策略的定义，使系统能够按照设定的顺序和条件操作任务，实现对外部设备工作过程的精确控制。

其中，【实时数据库】是整个软件的核心，从外部硬件采集的数据送到实时数据库，再通过【用户窗口】更改数据库的值，最后由【设备窗口】输出到外部硬件。

用户窗口中的动画构件关联实时数据库中的数据对象，动画构件按照数据对象的值进行相应的变化，从而达到“动”起来的效果。

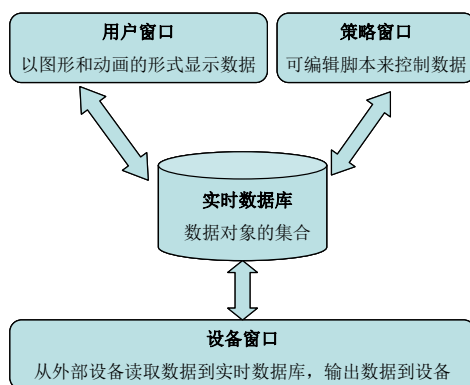


图 1.4-2 McgsPro 组态软件原理图

1.4.1 主控窗口介绍

- 打开【主控窗口】设置界面：
 - ①选择工作台的主控窗口页；
 - ②双击下方的【主控窗口】图标，弹出【主控窗口】设置界面。



图 1.4-3 打开主控窗口

- 选择【基本属性】标签按钮，即进入【基本属性】设置窗口页。主控窗口的基本属性决定了工程在启动时的总体概貌及外观。

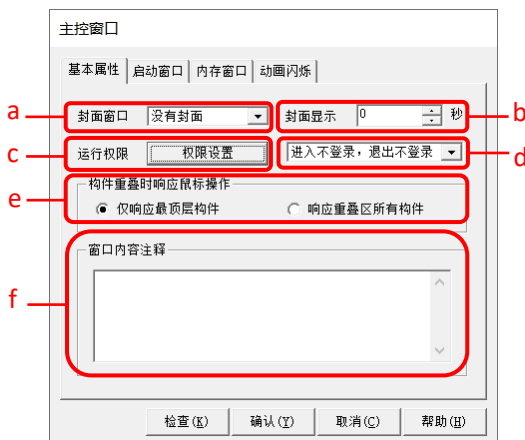


图 1.4-4 主控窗口-基本属性功能页

- a. 【封面窗口】：在下拉菜单中选择相应的窗口作为封面窗口。

- b. **【封面显示】**: 设置封面持续显示的时间, 以秒为单位。运行时, 鼠标点击窗口任何位置, 封面自动消失。当封面时间设置为 0 时, 封面将一直显示, 直到鼠标单击窗口任何位置时, 封面方可消失。
 - c. **【运行权限】**: 点击“权限设置”按钮后弹出**【用户权限设置】**界面。在该界面可将进入或退出工程的权限赋予某个用户组。无此权限的用户组中的用户, 不能进入或退出该工程。当选择“所有用户”时, 相当于无限制。此项措施对防止无关人员的误操作, 提高系统的安全性起到重要的作用。
 - d. 选择 c 项设置的权限的具体内容。选项包括:
 - 进入不登录, 退出登录。即当用户启动运行环境时, 不必登录, 退出运行环境时, 需登录。
 - 进入登录, 退出不登录。即当用户启动运行环境时, 需登录, 退出时不必登录。
 - 进入不登录, 退出不登录。即进入或退出运行环境时, 都不必登录。
 - 进入登录, 退出登录。即进入或退出运行环境时, 都需要登录。
 如: 在 c 项中设置了拥有权限的用户组为“管理员组”, d 项设置“进入登录, 退出不登录”, 那么, 进入工程时, 必须在弹出的对话框中输入正确的“管理员组”的用户名和密码, 才能进入工程, 否则, 不能进入工程。而退出时不做此要求。
 - e. **【构件重叠时响应鼠标操作】**: 当多个动画构件存在重叠区域时, 鼠标点击重叠区域的表现由该设置项决定。可设置为**【仅响应最顶层构件】**或者**【响应重叠区所有构件】**, 该设置项对整个组态工程的所有用户窗口生效。
 - f. **【窗口内容注释】**: 起到说明和备忘的作用, 对应用工程运行时的外观不产生任何影响。
- 选择**【启动窗口】**标签按钮, 进入**【启动窗口】**设置窗口页。



图 1.4-5 主控窗口-启动窗口功能页

图中左侧为**【用户窗口列表】**, 列出了所有定义的用户窗口名称。右侧为**【启动窗口】**列表, 它是启动时自动打开的用户窗口的列表。

点击“增加”按钮或用鼠标双击左侧列表内指定的用户窗口, 可以把该窗口移动到右侧。

点击“删除”按钮或用鼠标双击右侧列表内指定的用户窗口, 可以将该窗口移动到左侧。

注意: 只能设置一个用户窗口为启动窗口。

- 选择**【内存窗口】**标签按钮, 进入**【内存窗口】**设置窗口页。



图 1.4-6 主控窗口-内存窗口功能页

工程运行过程中，打开一个用户窗口时，系统首先把窗口的数据从磁盘调入内存，然后再执行窗口打开指令，这样一个打开窗口的过程可能比较缓慢，满足不了某些工程的需要。为了加快用户窗口的打开速度，运行环境提供了一种直接从内存中打开窗口的机制，即把用户窗口数据预先装入内存，每次打开该窗口时直接从内存中打开，节省了打开窗口的时间。这些被预装到内存中的窗口，被称为【内存窗口】。注意：组态时内存窗口不宜设置过多，否则会降低 TPC 运行性能。

图 1.4-6 左侧为所有定义的【用户窗口列表】，右侧为【内存窗口】列表，及启动时装入内存中的用户窗口列表。

点击“增加”按钮或用鼠标双击左侧列表内指定的用户窗口，可以把该窗口移动到右侧，成为始终位于内存中的用户窗口。

点击“删除”按钮或用鼠标双击右侧列表内指定的用户窗口，可以将该窗口移动到左侧。

- 选择【动画闪烁】标签按钮，进入【动画闪烁】设置窗口页。该页面主要设置与动画闪烁周期有关的时间参数。



图 1.4-7 主控窗口-动画闪烁功能页

快速闪烁周期：其值在 100~1000ms（毫秒）之间；

中速闪烁周期：其值在 200~2000ms（毫秒）之间；

慢速闪烁周期：其值在 150~2000ms（毫秒）之间。

小于周期范围的值会被系统强制转换为周期最小值，大于周期范围的值会被系统强制转换为周期最大值。

1.4.2 设备窗口介绍

设备窗口是 McgsPro 组态软件的重要组成部分，在设备窗口中建立系统与外部硬件设备的连接关系，使系统能够从外部设备读取数据并控制外部设备的工作状态，实现对工业过程的实时监控。

- 打开【设备窗口】设置界面：
 - ①选择工作台的主控窗口页
 - ②双击下方的【设备窗口】图标，弹出【设备窗口】设置界面

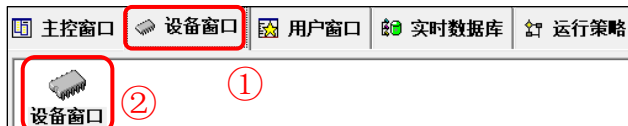


图 1.4-8 打开设备窗口

- 【设备窗口】设置界面由【设备组态画面】和设备【工具箱】两部分组成。设备组态画面用于配置该工程需要通讯的设备。设备工具箱里是常用的设备。在设备工具箱里的设备名称上双击，可以把设备添加到设备组态画面。
- 要添加或删除设备工具箱中的设备驱动时，可点击设备工具箱顶部的【设备管理】按钮。打开【设备管理】窗口。在【设备管理】窗口左侧的【可选设备】区域的树形目录中找到需要的设备，双击或点击下方【增加】按钮即可添加到【选定设备】区域。选中【选定设备】区域里的设备，点击窗口左下方的【删除】按钮可删除该设备。如图 1.4-9 所示。【工具箱】中的设备就是【选定设备】区域中的设备。

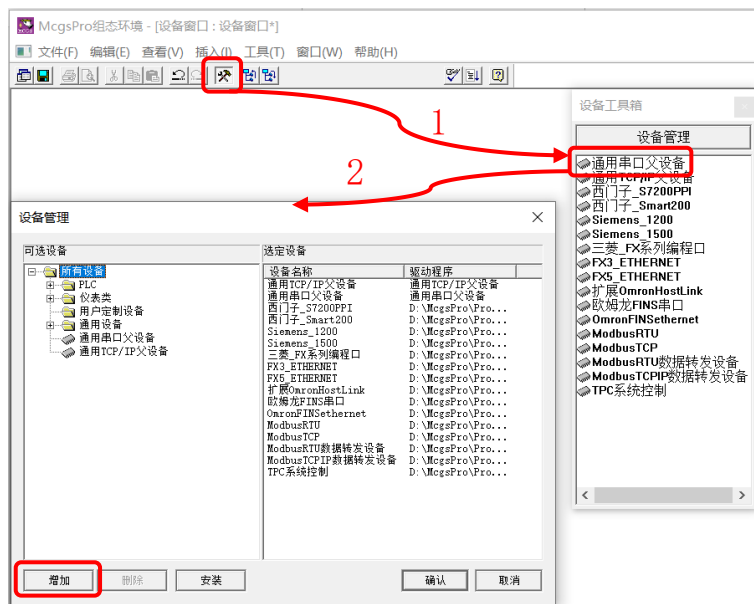


图 1.4-9 在设备窗口增加设备

- 如果要使用系统目前没有的设备，单击【安装】按钮，系统弹出对话框询问是否需要安装新增的驱动程序，选择“是”，指明驱动程序所在的路径，进行安装，安装完毕，新的设备将显示在设备管理窗口左侧列表的【用户定制设备】目录下。接下来就可以进行新设备的登记工作了。
- McgsPro 组态软件中把设备分成两个层次：父设备和子设备。父设备可以看作硬件接口。子设备放在父设备下，用于与该父设备对应的接口所连接的设备进行通讯。在设备组态画面双击父设备或子设备可以设置通讯参数。如图 1.4-10 所示。

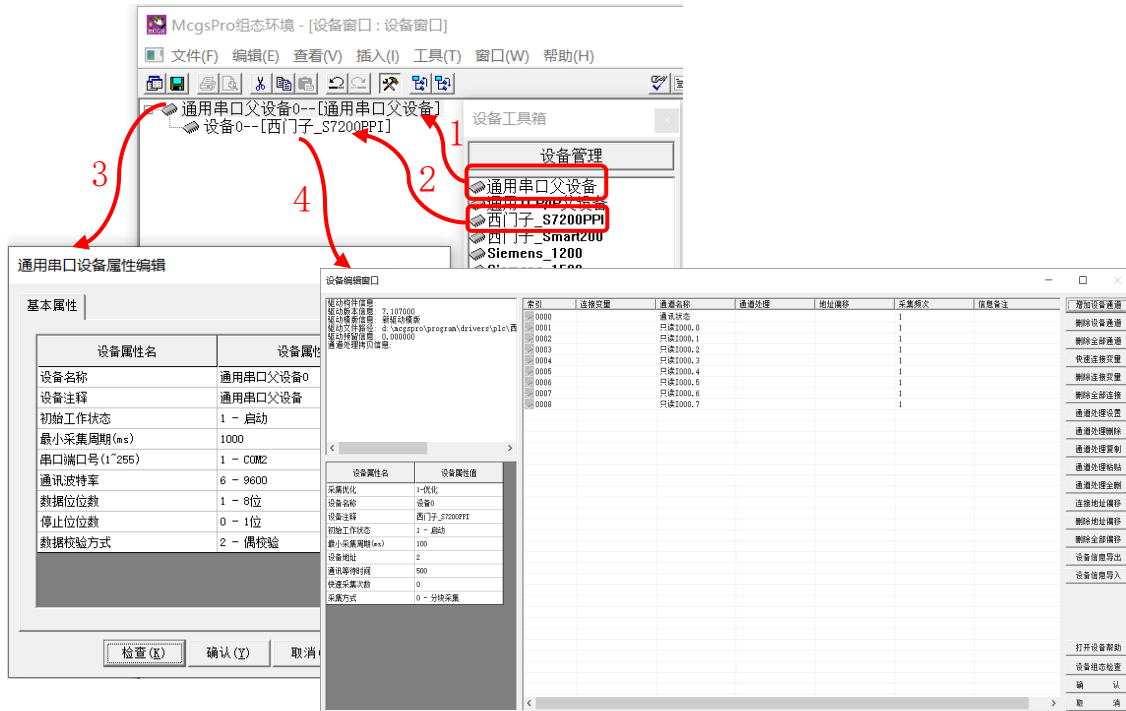


图 1.4-10 打开父设备属性编辑窗口及子设备编辑窗口

- 串口父设备里可设置串口号、波特率、数据位、停止位、校验方式。TCP/IP 父设备可设置本地 IP 地址、本地端口号、远程 IP 地址、远程端口号等。
- 子设备的设备编辑窗口分为三个区域：驱动信息区、设备属性区和通道连接区。驱动信息区里显示的是该设备驱动版本、路径等信息。设备属性区可设置采集周期、设备地址、通讯等待时间等通讯参数。通道连接区用于构建下位机寄存器与 McgsPro 组态软件变量之间的映射。



图 1.4-11 子设备编辑窗口

(1) 驱动信息

在这个信息栏中包括了驱动的版本信息，模版信息，驱动文件路径，驱动预留信息，通道处理拷贝信息。

(2) 设备属性

要使 McgsPro 组态软件能正确操作 PLC 设备，必须按如下的步骤来使用和设置本构件的属性：

采集优化：设置为优化时，在进入 McgsPro 运行环境时，驱动设备将只采集界面、脚本、策略使用到的通道和拥有存盘、报警属性的通道，以提高采集效率；设置为不优化时，驱动设备将采集全部通道。

设备名称：可根据需要来对设备进行重新命名，但不能和设备窗口中已有的其它设备构件重名。

初始工作状态：设置为启动时，在进入 McgsPro 运行环境时，自动开始对设备进行操作，设置为停止时，不对设备进行操作，但可以使用设备操作函数和策略在 McgsPro 运行环境中启动或停止设备。

最小采集周期：指系统对设备构件的读写操作的最快时间周期，单位为毫秒。一般在静态测量时设为 1000ms，在快速测量时设为 200ms。

(3) 通道信息

通道信息内容是通道连接区左侧的表格部分，内容包括：索引，连接变量，通道名称，通道处理，地址偏移，采集周期，信息注释。

选中某一行，在连接变量列双击左键或者单击右键：打开通道连接变量选择窗口进行变量选择，可以选择多个。

选中某一行，在通道处理列双击左键或者单击右键：打开通道处理设置窗口。

选中某一行，在地址偏移列双击左键或者单击右键：打开通道连接变量选择窗口进行变量选择，只能选择一个。

(4) 功能按钮

【增加设备通道】：增加后通道后立即反映到通道信息表格中，如图 1.4-12。

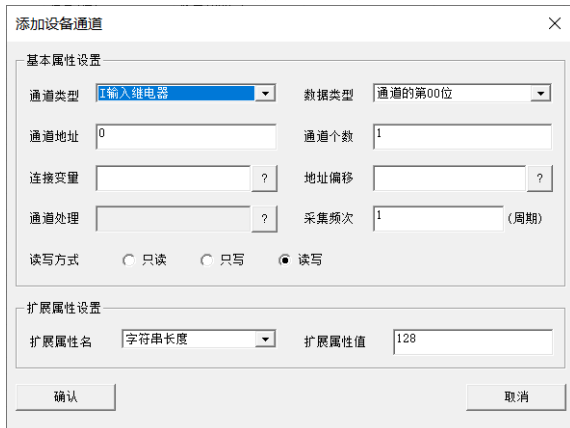


图 1.4-12 添加设备通道

【删除设备通道】：删除选中通道信息表格中选中的一个或多个通道。

【删除全部通道】：删除选中通道信息表格中所有的通道内容，通讯状态除外。

【快速连接变量】：为通道信息表格的通道连接变量提供一种方便快捷的连接方式,可实现多通道连接,如图 1.4-13。

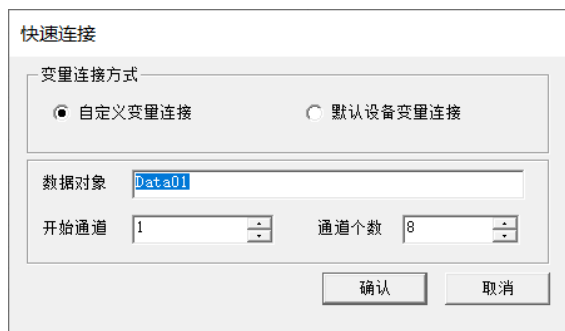


图 1.4-13 快速连接变量

有两种连接方式：【自定义变量连接】和【默认设备变量连接】，如果所定义的变量没有在实时数据库中定义，则在点击设备组态窗口下面的确认按钮时会给出提示，自动把所有变量添加到实时数据库中。

自定义变量连接：输入变量名称，从【开始通道】处开始连接变量，根据通道个数添加相应个数通道的变量连接，如从 1 通道开始添加 11 个通道的连接，变量从 Data01 开始起，通道 1，……11 对应的连接变量依次为 Data01，……Data11。

默认设备变量连接：所有通道连接的变量统一被替换成一种格式的变量，格式为“设备名+读写方式+地址”。

注意：如果通道中原来就连接有变量，则再次连接时清除原来的连接变量，重新连接新的变量。

【删除连接变量】：选中通道信息表格中一行或多行（不管有没有连接变量都可以），点击该功能按钮即可删除选中通道连接的变量。

【删除全部连接】：删除通道信息表格中的所有通道连接的变量。

【通道处理设置】：对从设备中采集到的数据或输出到设备的数据进行前处理，以得到实际需要的工程物理量。如从 AD 通道采集进来的数据一般都为电压 mV 值，需要进行量程转换或查表计算等处理才能得到所需的物理量。

【通道处理删除】：删除选中通道中的通道处理方法。

【通道处理复制】：只对选中的通道中索引号最小的通道处理进行复制，且只复制其通道处理方法，内容注释不复制。

【通道处理粘贴】：把复制的通道处理方法粘贴到选中的一个通道中，通道处理注释默认为“#通道处理：处理方法的序号”。

【通道处理全删】：删除通道信息栏中所有通道的通道处理。

【连接地址偏移】：选中通道信息表格中一个或多个通道，点击该功能按钮为通道连接地址偏移。

【删除地址偏移】：选中通道信息表格中一个或多个通道，点击该功能按钮即可删除选中通道连接的地址偏移。

【删除全部偏移】：删除通道信息表格中的所有通道连接的地址偏移。

【设备信息导出】：该功能可以把通道信息表格的内容以.CSV 格式导出到指定的位置，.CSV 格式可以使用 Microsoft Office 提供的 Excel 和记事本打开。注意编辑文件时不可更改文件的以下内容：组态设备名称，驱动库文件路径，驱动构件名称，驱动构件版本。否则导入文件时将不成功。导出内容包括：通道号,变量名,变量类型,通道名称,读写类型,寄存器名称,数据类型,寄存器地址。

【设备信息导入】：使用该功能可以从外界导入编辑好或保存好的通道信息内容，方便使用者的组态。导入内容包括：变量名，变量类型，通道名称，读写类型，寄存器名称，数据类型，寄存器地址。

使用导入导出时应注意以下 3 个方面：

- 导入文件时必须保证要导入文件的组态名称，驱动库文件路径，驱动构件名称，驱动构件版本和当前设备组态窗口的驱动信息内容保持一致，并且要导入的文件在导入时没有被使用 Excel 打开，否则导入不成功。
- 变量类型为三种，Integer，Single，String。
- 通道名称在导出文件中有，实际导入中无作用，可有可无。

【打开设备帮助】：打开对应设备的帮助内容。

【设备组态检查】：进行工程正确性检查。

【确认】：保存在设备组态窗口中进行的操作，并进行正确性检查。

【取消】：不保存设备组态窗口中进行的所有的操作。

1.4.3 用户窗口介绍

- 用户窗口主界面的右侧有三个按钮：每点击一次【新建窗口】按钮可以新建一个窗口，【窗口属性】用于打开已选中窗口的属性设置，可以将窗口设置为启动窗口、用户窗口和子窗口，其中启动窗口标注为绿色五角星，用户窗口为黄色五角星，子窗口为紫色五角星。双击窗口图标或者选中窗口之后点击“动画组态”按钮可以进入该窗口的编界面。如图 1.4-14 所示。

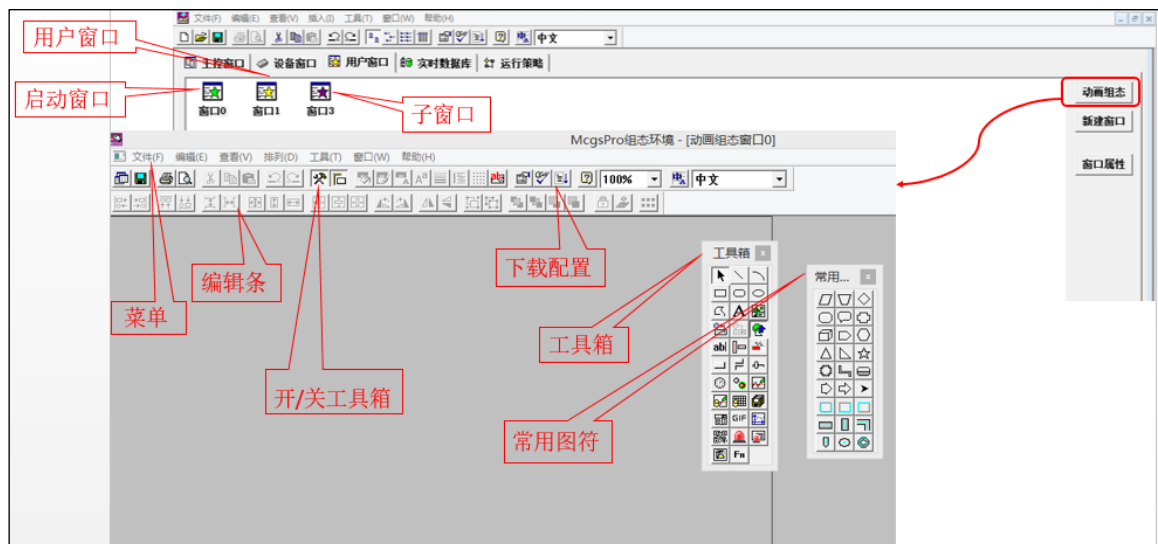


图 1.4-14 动画组态窗口界面

- 窗口编辑界面的主要部分是【工具箱】和【窗口编辑区域】。工具箱有我们画面组态要使用的所有构件。窗口编辑区域用于绘制画面，运行时我们能看到的画面都需在这里添加。在工具箱单击所需要的构件，然后在窗口编辑区域中按住鼠标左键拖动将该构件添加到画面中。
- 工具箱中常用的构件有：标签、输入框、标准按钮、二维码、配方、位图、开/关常用图符、GIF、XY曲线、键盘按钮等。如图 1.4-15 所示。

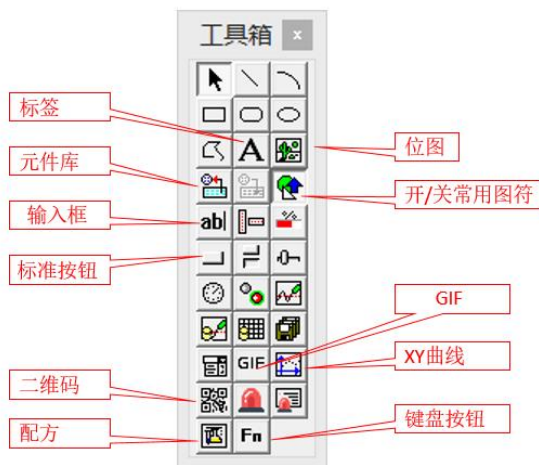


图 1.4-15 工具箱

- 将构件添加到窗口编辑区域之后，双击该构件可打开该构件的属性。构件的作用不同，属性设置界面有很大的差异。可点击属性设置界面的右下角的“帮助”按钮查看构件属性设置详细说明。如图 1.4-16 所示。



图 1.4-16 标准按钮构件属性设置界面

1.4.4 实时数据库介绍

实时数据库是 McgsPro 组态软件的核心，是应用系统的数据处理中心。系统各部分均以实时数据库作为公用区进行数据交换，实现各个部分地协调运作。

1.4.4.1 数据对象类型

McgsPro 中数据对象主要有整数、浮点数、字符串和组对象，每种数据类型的属性不同用途也不同。

(1) 整数数据对象

整数数据对象的数值范围是：-2147483648 到 2147483647。整数数据对象通常用与外部设备的数字量输入输出通道连接，用来表示某一设备当前的状态或记录设备的当前整型值。整数数据对象也用于表示 McgsPro 中某一对象的状态，如一个图形对象的可见度状态。

整数数据对象可以设置状态报警（开关量报警、正跳变报警、负跳变报警）、位报警（位==报警、位 ON→OFF 报警、位 OFF→ON 报警）、值报警（值==报警、值>报警、值≥报警、值<报警、值≤报警）。

(2) 浮点数数据对象

浮点数数据对象的取值范围是：-1.79E+308 到+1.79E+308。浮点数数据对象除了存储数值和参与数据运算外，还提供报警信息，并与外部设备的模拟量输入输出通道连接。

浮点数数据对象有限值报警属性（下下限、下限、上限、上上限、上偏差、下偏差），当对象的值超出报警限值时，产生报警；当对象的值在报警限值以内，报警结束；浮点数数据对象还可以设置值报警（值==报警、值>报警、值≥报警、值<报警、值≤报警）。

(3) 字符串数据对象

字符串数据对象是存放文字信息的单元，用于描述外部对象的状态特征，由多个字符构成，如果字符串作为初值保存，最大允许长度为 8KB；如果该对象作为历史数据存储，最大允许长度约 32KB，其它情况无长度限制。

(4) 组对象数据对象

组对象是 McgsPro 引入的一种特殊类型的数据对象，类似于编程语言中的数组和结构体，用于把相关的多个数据对象集合在一起，作为一个整体的定义和处理。在实际的工程中，描述一个锅炉的工作状态有温度、压力、液面高度、流量等多个物理量。为便于处理，定义一个“锅炉”组对象与实际的物理对象进行对应，其内部成员则由上述物理量对应的数据对象组成。这样，在对“锅炉”对象进行处理（如进行组态存盘、曲线显示、报警显示）时，只需指定组对象的名称“锅炉”，就包括了对其所有成员的处理。

组对象只是在组态时对某一类对象的一种整体表示，实际操作则是针对某一个成员进行的。如在报警显示动画构件中，指定要显示的报警数据对象为“锅炉”，则该构件显示组对象包含的各个数据对象在运行时产生的所有报警信息。

把一个对象定义成组对象后，还必须设置组对象包含的成员，如图 1.4-17 所示，在“数据对象属性设置”对话框内，专门有“组对象成员”标签页，用于设置组对象的成员。对话框的左边为数据对象成员的列表，右边为组对象成员的列表，利用属性页中的“增加”按钮，可以将左边指定的对象添加到组对象中成员中；也可

以利用“删除”按钮删除指定的组对象的成员。



图 1.4-17 组对象属性设置界面

1.4.4.2 添加数据对象

定义数据对象时，在组态环境工作台窗口，选择“实时数据库”标签，进入实时数据库窗口页，显示出已定义的数据对象，分别是 InputSTime, InputEtime, InputUser1, InputUser2。数据对象可以使用大图标、小图标、列表和详细信息方式进行显示，可以使用名称顺序或者类型顺序来显示变量。也可以剪切、拷贝、粘贴指定的数据对象。

数据对象可有两种方式进行添加：单个数据对象添加和成组数据对象添加。

(1) 单个数据对象添加：点击【新增对象】按钮，新增默认名称为【Data1】的【浮点数】对象。若先选中一个数据类型为【字符串】的数据对象【A1】，再点击【新增对象】按钮，则会智能增加一个数据类型同为【字符串】的数据对象【A2】。

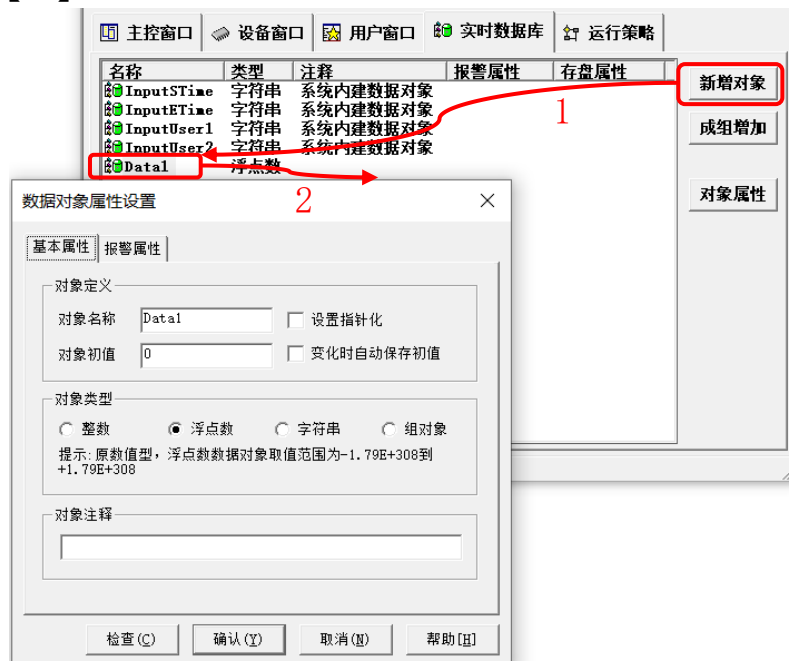


图 1.4-18 新增数据对象流程

(2) 成组数据对象添加：批量生成多个名称编号依次递增、数据类型相同的数据。选择“成组增加”按钮，弹出“批量添加数据对象”对话框，一次定义多个数据对象如图 1.4-19 所示，其中对象名称一栏，代表该组对象的主体部分，而“起始索引值”则代表第一个成员的“索引代码”，其他数据对象的主体名称相同，索引代码依次递增。批量添加的数据对象，其他特性如数据类型、工程单位都是一样的。批量修改具有相同

属性的数据对象时，可选中需要修改的对象，再选择对象属性统一修改。

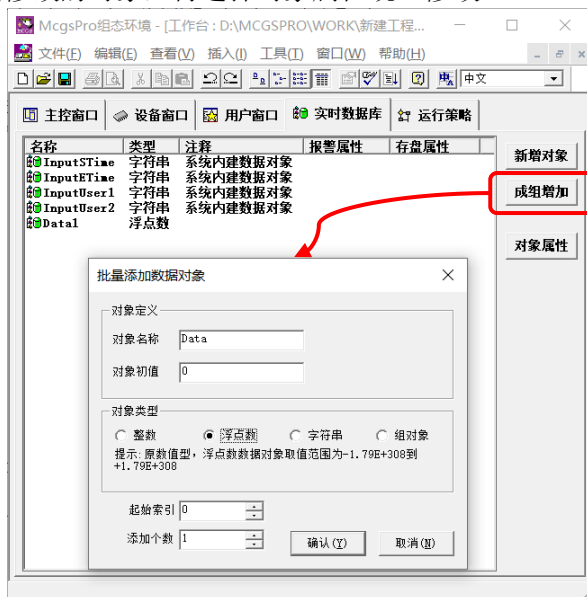


图 1.4-19 成组增加数据对象流程

1.4.4.3 数据对象配置

● 整数数据对象

整数数据对象有基本属性、存盘属性、报警属性，可以在【数据对象属性设置】对话框中设置对应的属性。

(1) 【基本属性】页用于设置数据对象的基本属性，见图 1.4-20。

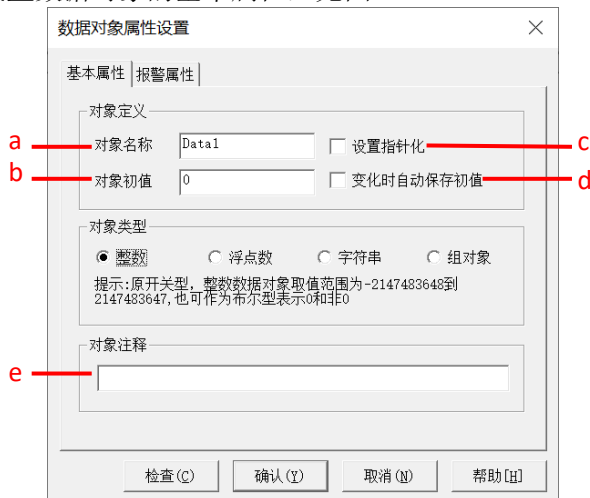


图 1.4-20 数据对象-基本属性页

- 对象名称：用于显示和修改数据对象的名称，指定的数据对象名称不能以“!”“\$”开头，不能使用加减乘除等运算符，不能使用大于等于小于等逻辑运算符
- 对象初值：用于在数据对象初始化的时候，赋初值给数据对象
- 设置指针化：可将整数和浮点数数据对象设置为指针化数据对象
- 变化时自动保存初值：添加初值属性，初值改变后 60 秒才会刷盘
- 对象注释：用于对该对象的进行注释和说明

(2) 报警属性页

报警属性页用于设置数据对象的报警属性，在报警属性表格中右键进行报警属性插入、追加、删除、剪切及粘贴，双击已有报警表项可以进行修改，整数对象有状态报警、位报警和值报警，当对象的值触发相应的报警条件时，将产生报警，见图 1.4-21。

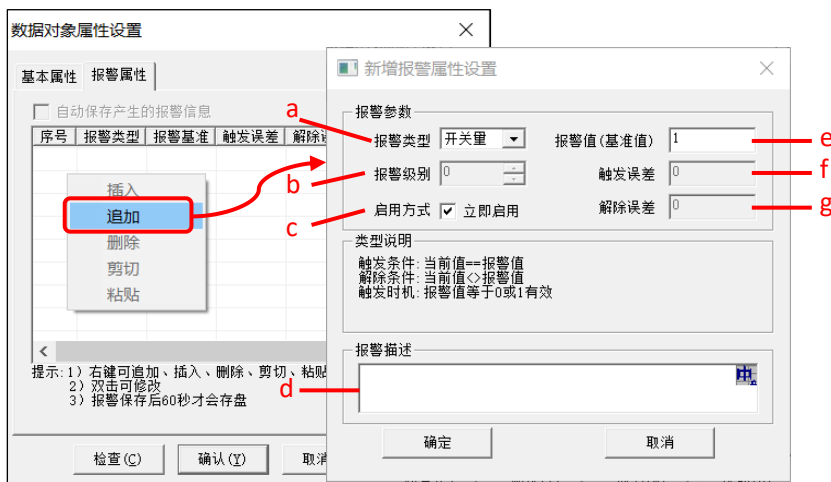


图 1.4-21 整数-报警属性功能页

- a. 报警类型：设置报警属性的类型，原报警类型（开关量、跳变、限值、偏差报警）不可重复，新报警类型(位值报警)可重复
 - b. 报警级别：设置报警优先级，当前无效（保留）
 - c. 启用方式：选中表示当前设置的报警会立即生效
 - d. 报警描述：用于描述该项报警的注释型信息，所有类型的报警都有“报警注释”
 - e. 报警值(基准值)：报警参数的参照值，限值与开关量报警称作报警值，位报警称作指定位
 - f. 触发误差：触发报警参数，部分报警类型此值无效，偏差报警称作报警值，位报警称作指定值
 - g. 解除误差：解除报警误差参数值，部分报警类型此值无效
- 浮点数数据对象

浮点数数据对象有基本属性和报警属性，可以在【数据对象属性设置】对话框中设置对应的属性。

(1) 基本属性页

其中基本属性页与整数基本属性页相同，详细说明见整数数据对象基本属性页。

(2) 报警属性页

浮点数数据对象支持限值报警和值报警，即当浮点数数据对象的值满足报警条件时，将会触发报警。浮点数数据对象的报警属性页见图 1.4-22。报警详细设置方式与整数报警设置相同。

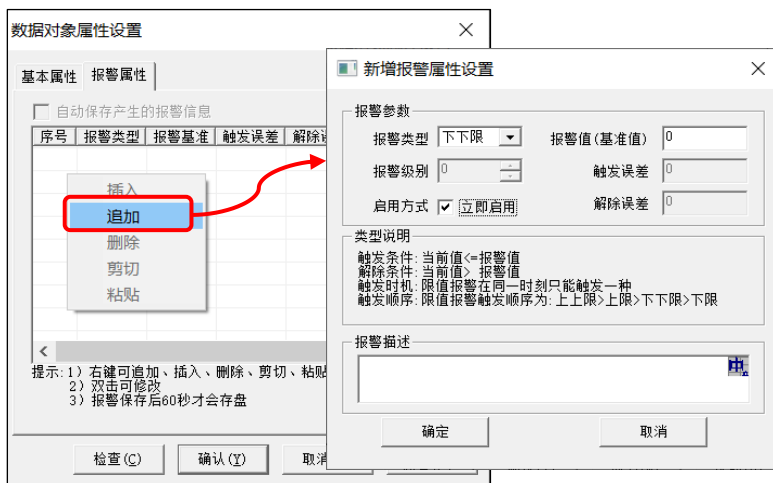


图 1.4-22 浮点数-报警属性功能页

注意：浮点数数据对象限值报警触发顺序为上上限、上限、下下限和下限；偏差报警触发顺序为上偏差、下偏差。限制报警同一时刻只能触发一种限制报警，偏差报警同一时刻只能触发一种偏差报警。

● 字符串数据对象

字符串数据对象有基本属性页，不具备存盘属性和报警属性功能。其中基本属性页的设置同整数数据对象的基本属性设置，详见整数数据对象的基本属性页设置。

● 数据组数据对象

组对象拥有基本属性设置和存盘属性设置，不具备报警属性设置。其中基本属性设置与整数数据对象的基本属性设置一致，详见整数的数据对象的基本属性描述。组对象成员设置方法参考组对象说明，详细界面见图 1.4-17。

(1) 存盘属性页

组对象的存盘属性页主要用于设置组对象是否存盘以及存盘的周期，具体见图 1.4-23 所示。

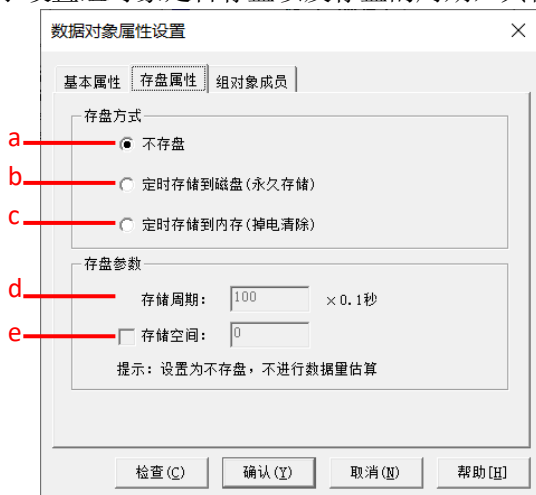


图 1.4-23 组对象-存盘属性功能页

- a. 不存盘：勾选则表示在运行环境中禁止存盘
 - b. 定时存储到磁盘：历史数据保存到磁盘上，断电后数据依然存在
 - c. 定时存储到内存：历史数据保存到缓存中，断电后数据丢失
 - d. 存储周期： $=0$ ，表示手动调用脚本本存盘，可在运行环境调整；
 >0 ，按设定间隔保存数据，最小值为 100 毫秒，可在运行环境调整
 - e. 存储空间：当历史数据达到指定大小或磁盘空间不足时，最早的数据将会被删除
- 注意：存储周期单位为 0.1 秒；当存储方式为存储到磁盘时，空间最大为 2000MB，当存储方式为存储到缓存时，空间最大为 1024KB，当存储方式为存储到缓存时，如果空间指定不为 0 则最小空间为 256KB。

1.4.4.4 数据对象存盘

数据对象存盘有三种：历史数据存盘、报警数据存盘和初值数据存盘。数据对象存盘的详细内容将在后续章节：进阶篇 5.2 历史数据功能进行介绍。

1.4.4.5 变量报警

● 名词解释

当前值：变量实时值

基准值：报警值计算时的标准参数值

报警值：当前值相对于基准值达到报警值时会触发报警（此值部分报警有效）

触发误差值：触发报警计算时相对于基准值的误差值

解除误差值：解除报警计算时相对于基准值的误差值

● 状态报警

报警类型		状态	计算方法	备注
开关量报警	报警值为 0	触发	当前值 == 0	
		解除	当前值非 0	
	报警值为非 0	触发	当前值非 0	
		解除	当前值 == 0	
正跳变报警		触发	当前值 0-->非 0	忽略初值报警
		解除	当前值非 0-->0	
负跳变报警		触发	当前值非 0-->0	忽略初值报警

报警类型	状态	计算方法	备注
	解除	当前值 0→非 0	

● 位报警

报警类型	状态	计算方法	备注
位报警==	触发	指定位 == 指定值	
	解除	指定位 <> 指定值	
位报警 ON→OFF	触发	指定位 1 → 0	忽略初值报警
	解除	指定位 0 → 1	
位报警 OFF→ON	触发	指定位 0 → 1	忽略初值报警
	解除	指定位 1 → 0	

● 限值报警

报警类型	状态	计算方法	备注
上上限报警	触发	当前值 ≥ 报警值	
	解除	当前值 < 报警值	
上限报警 (无上上限报警)	触发	当前值 ≥ 报警值	
	解除	当前值 < 报警值	
上限报警 (有上上限报警)	触发	报警值 ≤ 当前值 < 上上限报警值	
	解除	当前值 < 报警值 或 当前值 ≥ 上上限报警值	
下下限报警	触发	当前值 ≤ 报警值	
	解除	当前值 > 报警值	
下限报警 (无下下限报警)	触发	当前值 ≤ 报警值	
	解除	当前值 > 报警值	
下限报警 (有下下限报警)	触发	下下限报警值 < 当前值 ≤ 报警值	
	解除	当前值 > 报警值 或 当前值 ≤ 下下限报警值	

● 偏差报警

报警类型	状态	计算方法	备注
下偏差报警	触发	当前值 ≤ 基准值 - 报警值	
	解除	当前值 > 基准值 - 报警值	
上偏差报警 (下偏差优先级高于上偏差优先级)	触发	当前值 ≥ 基准值 + 报警值	
	解除	当前值 < 基准值 + 报警值	

● 值报警

报警类型	状态	计算方法	备注
值报警==	触发	基准值 - 触发误差值 ≤ 当前值 且 当前值 ≤ 基准值 + 触发误差值 (即报警触发区间范围:[基准值 - 触发误差值, 基准值 + 触发误差值])	
	解除	当前值 < 基准值 - 解除误差值 或 当前值 > 基准值 + 解除误差值 (即报警解除区间范围:(-∞, 基准值 - 触发误差值) ∪ (基准值 + 触发误差值, +∞))	
值报警<>	触发	当前值 < 基准值 - 触发误差值 或 当前值 > 基准值 + 触发误差值	

报警类型	状态	计算方法	备注
		(即报警触发区间范围: $(-\infty, \text{基准值} - \text{触发误差值}) \cup (\text{基准值} + \text{触发误差值}, +\infty)$)	
	解除	基准值 - 解除误差值 $\leq$ 当前值 且 当前值 $\leq$ 基准值 + 解除误差值 (即报警触发区间范围: $[\text{基准值} - \text{触发误差值}, \text{基准值} + \text{触发误差值}]$)	
值报警 $\leq$	触发	当前值 $\leq$ 基准值 + 触发误差值	
	解除	当前值 $>$ 基准值 + 解除误差值	
值报警 $<$	触发	当前值 $<$ 基准值 + 触发误差值	
	解除	当前值 $\geq$ 基准值 + 解除误差值	
值报警 $\geq$	触发	当前值 $\geq$ 基准值 + 触发误差值	
	解除	当前值 $<$ 基准值 + 解除误差值	
值报警 $>$	触发	当前值 $>$ 基准值 + 触发误差值	
	解除	当前值 $\leq$ 基准值 + 解除误差值	

1.4.4.6 变量选择

在组态过程中,为了能够准确地输入数据对象的名称,经常需要从已定义的数据对象列表中查询或确认,同时也可能会根据需要添加一些数据对象。为了方便用户使用,提供两种变量选择方式:【从数据中心选择|自定义】和【根据采集信息生成】。

在数据对象的许多属性设置窗口中,对象名称或表达式输入框的右端,都带有一个“?”号按钮(图 1.4-24),单击该按钮,会弹出如下图 1.4-24 所示的对话框。

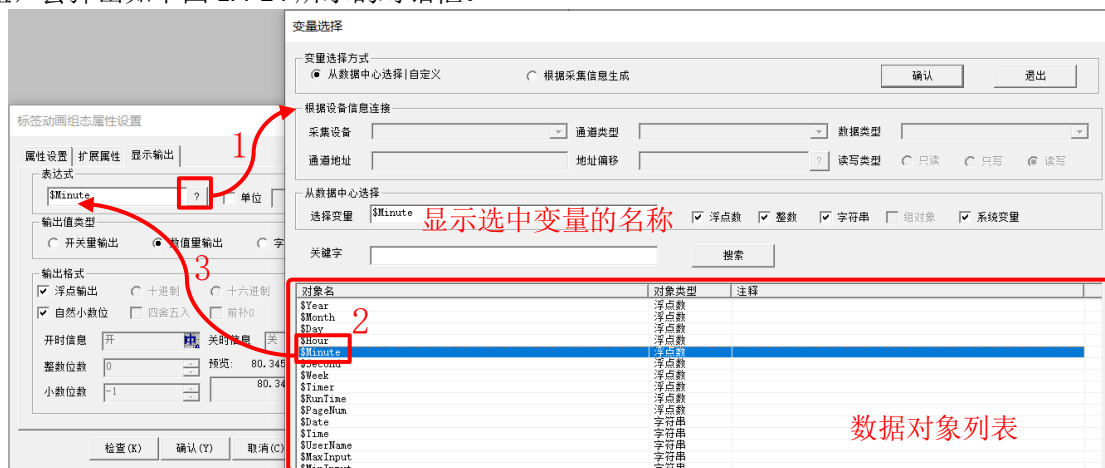


图 1.4-24 变量选择对话框

● 从数据中心选择|自定义

当数据变量选择方式使用【从数据中心选择|自定义】时,【根据设备信息连接】部分不能使用,【从数据中心选择】以及【数据对象列表】可用,单击数据对象列表中要选择的数据对象,点击【确认】按钮后对应的数据对象名称会自动添加到“?”号按钮左边的输入框内。另外双击列表中的指定数据对象也可实现这种功能。

部分构件只能选择一个数据变量。

自定义变量是指在【选择变量】后面的输入框中输入一个实时数据库中不存在的变量名称,确认后自动添加到“?”号按钮左边的输入框内,在输入框构件属性窗口点击确认时,系统会提示用户添加该数据对象到实时数据库中。

(1) 从数据中心选择

【从数据中心选择】模块,【选择变量】输入框能够显示选中变量的名称,也可以直接在此处输入变量名。根据输入框右侧复选框的值筛选数据对象类型,设置变量类型后,数据对象列表内容将同步更新。如:勾选整数和字符串复选框时,数据对象列表只显示整数和字符串变量。

(2) 关键字搜索

关键字搜索就是用户在使用【从数据中心选择】时，在【关键字】输入框中输入用户所要查找的信息内容后，点击【搜索】按钮，数据对象列表同步更新显示查找结果。如图 1.4-25 所示

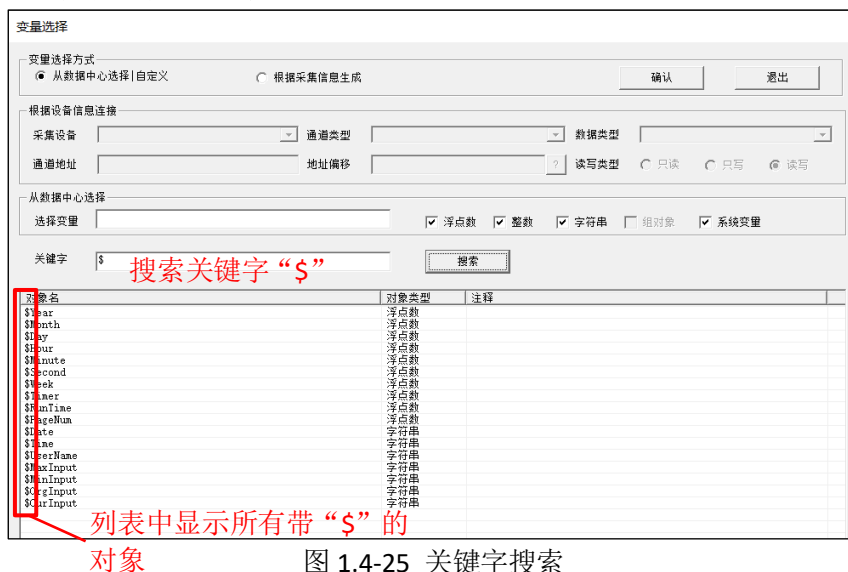


图 1.4-25 关键字搜索

● 根据采集信息生成

当数据变量选择方式使用【根据采集信息生成】时，首先，要确保在设备窗口中使用了设备，否则该功能不可用；其次，设备窗口要关闭，如果没有关闭设备窗口，在选择该方式时，则会弹出对话框询问是否自动关闭，如果不关则不能使用该方式。选择此方式时【从数据中心选择】不可用。在【根据设备信息连接】中，选择采集设备，然后选择该设备的通道类型，数据类型，通道地址，地址偏移，读写类型。点击“确认”按钮后，自动按“设备名+读写方式+地址”格式生成变量添加到“？”号按钮左边的输入框内，且自动在设备编辑窗口中添加对应的通道和变量。如果添加的通道在设备窗口中添加过并且连接过变量，会给出提示以防误操作。



图 1.4-26 根据采集信息生成数据变量

1.4.5 运行策略介绍

运行策略是用户为实现对系统运行流程自由控制所组态生成的一系列功能块的总称。

运行策略能够按照预设的顺序和条件操作实时数据库，控制用户窗口状态，修改设备运行数据，提高控制过程的实时性和有序性。

根据运行策略的不同作用和功能，McgsPro 组态件把运行策略分为后台任务、启动策略、退出策略、循环策略、用户策略、报警策略、事件策略及热键策略八种。

每种策略都由一系列功能模块组成。McgsPro 运行策略窗口中“启动策略”、“退出策略”、“后台任务”为系统固有的三个策略块，用户策略、循环策略、报警策略、事件策略、热键策略由客户根据工程需要自行定义，如图 1.4-27 所示。

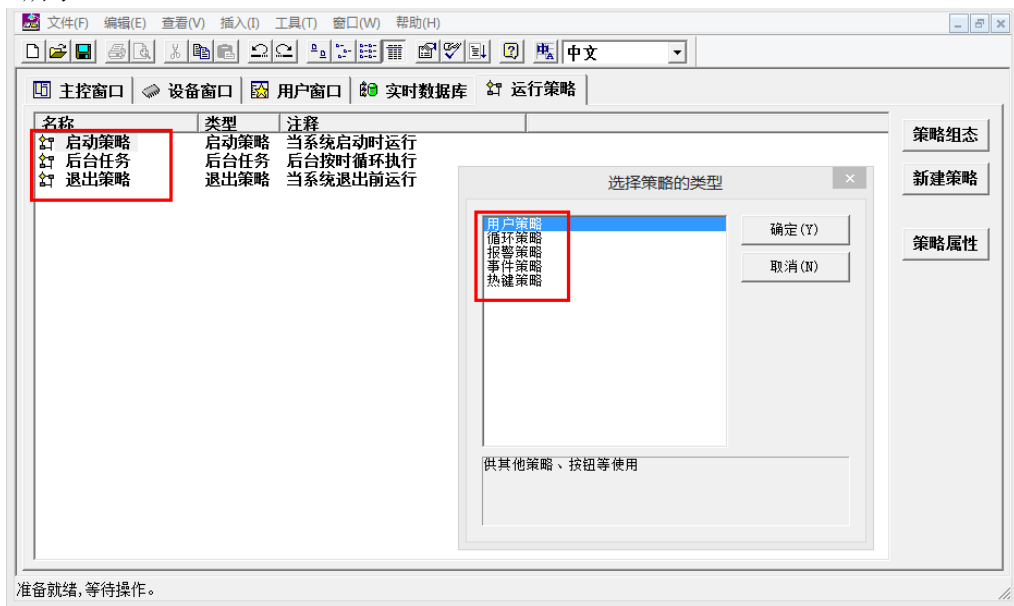


图 1.4-27 策略窗口

用户根据自己的需要，建立对应的策略后，点击“新增策略行”或者通过右键操作来增加策略行，新增的策略行如图 1.4-28 所示。



图 1.4-28 策略行建立

双击【策略构件】可打开脚本编辑窗口。详细的脚本功能将在后续章节：[进阶篇-第 11 章 脚本函数应用实例介绍](#)。

每一个策略行中，只有当策略执行条件成立时，系统才能对策略行中的脚本程序进行操作。通过对策略条件的设定（默认情况下，执行条件部分为真），用户可以自主的控制策略在什么时候、什么条件、什么状态下执行，如图 1.4-29 所示。



图 1.4-29 策略执行条件



图 1.4-30 启动策略

1.4.5.1 启动策略

启动策略为系统固有策略，在系统开始运行时被自动调用一次。启动策略属性设置如图 1.4-30 所示，其操作如下：

- 策略名称：输入启动策略的名字，由于系统必须有一个启动策略，所以此名字不能改变
- 策略内容注释：用于对策略加以注释

1.4.5.2 退出策略

退出策略为系统固有策略，在退出系统时自动被调用一次。退出策略属性设置如图 1.4-31 所示。

- 策略名称：退出策略名字，由于系统必须有一个退出策略，所以此名字不能改变
- 策略内容注释：用于对策略加以注释

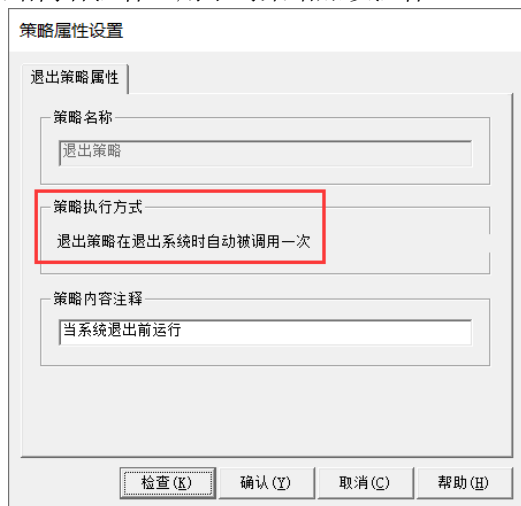


图 1.4-31 退出策略



图 1.4-32 后台任务

1.4.5.3 后台任务

后台任务为系统固有策略，在系统运行时按照设定的时间循环运行。在一个应用系统中，只能有一个后台任务。后台任务属性设置如图 1.4-32 所示。

- 策略名称：名称固定为后台任务，一个应用系统必须有一个后台任务策略
- 策略执行方式：定时循环：按设定的时间间隔循环执行，直接用 ms 来设置循环时间。最小循环时间间隔为 100ms，当设定值小于 100ms 时按 100ms 计算。实际运行过程中循环间隔有约 20ms 的误差值。
- 策略内容注释：用于对策略加以注释

1.4.5.4 循环策略

循环策略由用户在组态时创建，在系统运行时按照设定的时间循环运行。在一个应用系统中，用户可以定义多个循环策略。循环策略属性设置如图 1.4-33 所示。

- 策略名称：输入循环策略的名称
- 策略执行方式：
 - ① 定时循环：按设定的时间间隔循环执行，直接用 ms 来设置循环时间。最小循环时间间隔为 100ms，当设定值小于 100ms 时按 100ms 计算。
 - ② 固定时刻：策略在固定的时刻执行。
- 策略内容注释：用于对策略加以注释

图 1.4-33 循环策略

图 1.4-34 用户策略

1.4.5.5 用户策略

用户策略由用户在组态时创建，在系统运行时通过按钮、脚本调用。用户策略属性设置如图 1.4-34 所示，其操作如下：

- 策略名称：输入用户策略的名称。
- 策略内容注释：用于对策略加以注释。

1.4.5.6 报警策略

报警策略由用户在组态时创建，当指定数据对象的某种报警状态发生时，报警策略被系统自动调用一次。报警策略属性设置如图 1.4-35 所示，其操作如下：

- 策略名称：输入报警策略的名称。
- 策略执行方式：
 - ① 对应数据对象：用于与实时数据库的数据对象连接。
 - ② 对应执行条件：报警产生时执行一次/报警结束时执行一次/报警应答时执行一次。
- 策略内容注释：用于对策略加以注释。

图 1.4-35 报警策略

图 1.4-36 事件策略

1.4.5.7 事件策略

事件策略由用户在组态时创建，当对应数据对象的某种事件状态产生时，事件策略被系统自动调用一次。事件策略属性设置如图 1.4-36 所示。

- 策略名称：输入事件策略的名称。
- 策略执行方式：

- ① 对应数据对象：用于关联事件对应的数据对象。
- ② 执行条件：
 - 数据对象对应的事件内容有五种：
 - 数据对象的值正跳变(0 到 1)
 - 数据对象的值负跳变(1 到 0)
 - 数据对象的值正负跳变(0 到 1 再到 0)
 - 数据对象的值负正跳变(1 到 0 再到 1)。
 - 数据对象的值有改变时
- 策略内容注释：用于对策略加以注释。

1.4.5.8 热键策略

热键策略由用户在组态时创建，当用户按下对应的热键时执行一次。

热键策略属性设置如图 1.4-37 所示

- 策略名称：输入热键策略的名称。
- 热键：输入对应的热键（可以是组合键）。
- 策略内容注释：用于对策略加以注释。

图 1.4-37 热键策略

第2章 窗口画面应用实例

人机界面的窗口显示画面承担着与工程用户直接进行交互的重任。本章通过 McgsPro 组态软件展示常用的几种窗口画面显示方式，让用户快速掌握常用的窗口画面功能组态方法。

如图 2-1 所示，用户可以通过【新建窗口】，并在窗口属性设置中设置用户窗口属性（如窗口名等）。

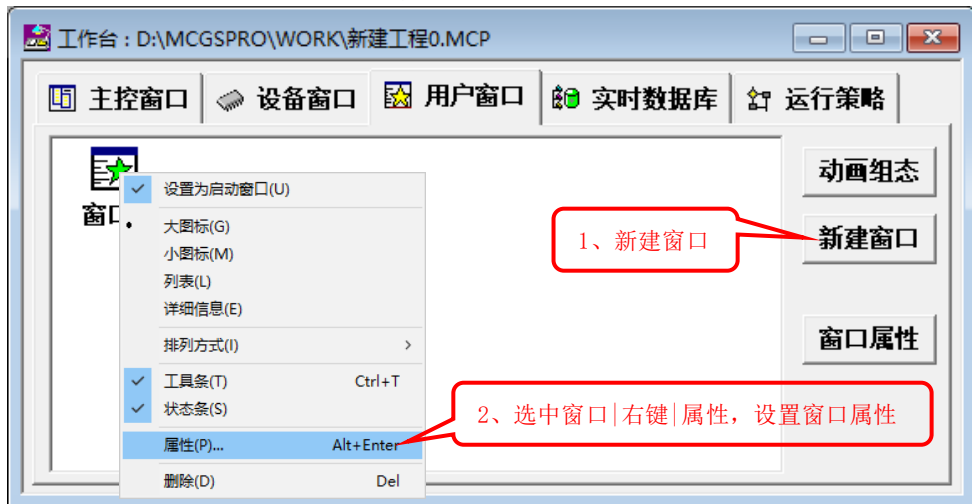


图 2-1 进入窗口属性设置界面的方法

注意：窗口属性设置进入方法 1：窗口0 → 右键 → 属性；方法二：双击窗口0 → 组态画面空白处右键 → 属性。

2.1 工程名片

TPC 工程除了必要的控制功能，还可以看作一张企业名片，为企业本身提供一定的宣传作用。对此，本节主要介绍工程封面窗口、启动窗口，以及二维码展示信息功能。

2.1.1 封面窗口和启动窗口设置

封面窗口和启动窗口功能：对于 McgsPro 工程，用户窗口画面启动顺序依次为：封面窗口 → 启动窗口。如工程未设置【封面窗口】，工程启动时将直接显示【启动窗口】。

注意：如图 2.1-1 所示，McgsPro 禁止将在窗口属性设置中勾选了【作为子窗口使用】的用户窗口，设置成封面窗口或启动窗口。

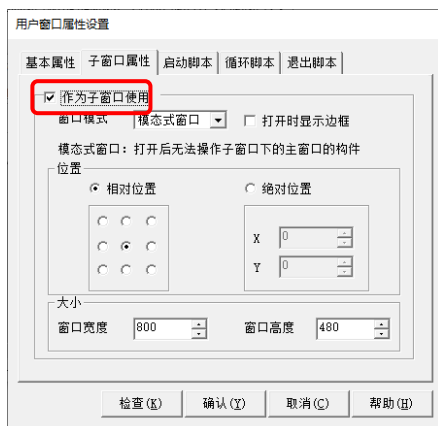


图 2.1-1 子窗口属性设置

● 封面窗口

封面窗口作为工程第一个运行的用户窗口，其显示时间可以手动指定。封面窗口运行时，手动点击封面窗口画面，或者等待封面窗口显示时间耗尽，工程将自动打开【启动窗口】。

封面窗口设置方法如下：

- ① 运行 McgsPro 组态软件，在工作台下的【用户窗口】中新建 2 个窗口画面：窗口 0、窗口 1。如图 2.1-2 所示：

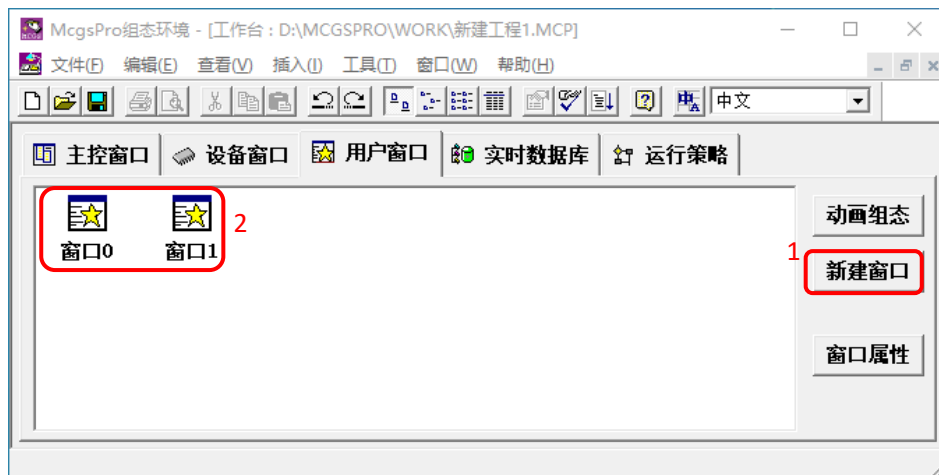
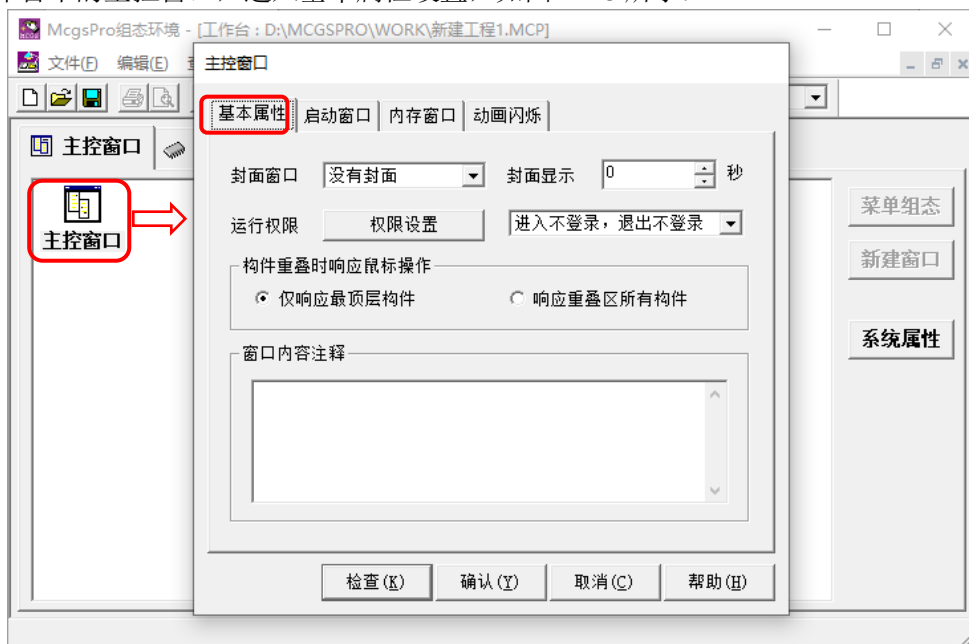


图 2.1-2 新建窗口画面

- ② 双击工作台下的主控窗口，进入基本属性设置，如图 2.1-3 所示：



2.1-3 主控窗口基本属性设置

- ③ 如图 2.1-4 所示，在【主控窗口】|【基本属性】功能页中，选择封面窗口，并设置封面显示时间即可（封面时间设置为 0，封面将一直显示，直到单击窗口任何位置时，封面方可消失）。封面窗口：窗口 0，封面显示：3 秒；

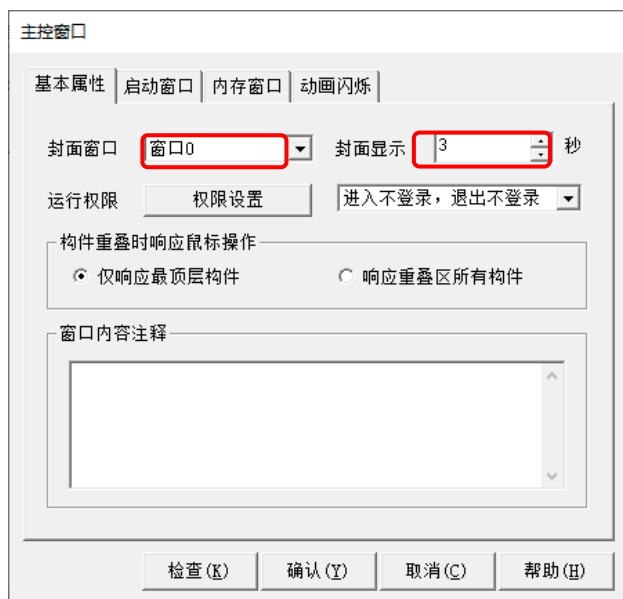


图 2.1-4 设置封面窗口

注意：图 2.1-4 中设置的封面窗口（窗口 0），其窗口显示内容需要用户自行组态。除了不能设置为子窗口外，封面窗口具备普通窗口所有功能。

● 启动窗口

启动窗口：作为封面窗口展示完成后启动的用户窗口，也需要用户手动进行设置。可通过以下 2 种方法设置启动窗口。

方法一：

- ① 在图 2.1-3 中的主控窗口属性设置，选择【启动窗口】功能页；
- ② 选择【用户窗口列表】→【窗口 1】→【增加】，将窗口 1 设置成启动窗口，如图 2.1-5 所示：



图 2.1-5 方法一：设置启动窗口

方法二：

直接在【工作台】→【用户窗口】中，选中【窗口 1】→【右键】→【设置为启动窗口】，如图 2.1-6 所示：

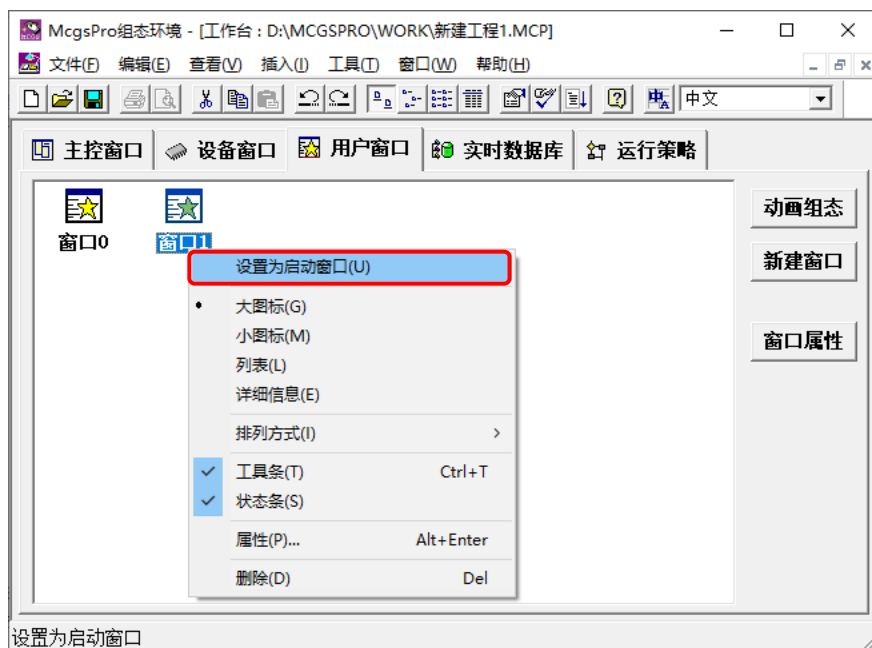


图 2.1-6 方法二：设置启动窗口

注意：设置的启动窗口，其窗口显示内容需要用户自行组态。除了不能设置为子窗口外，启动窗口具备普通窗口所有功能。

2.1.2 二维码展示信息

使用 McgsPro 组态软件的用户，可以通过 McgsPro 自带的二维码构件，将字符信息以二维码的方式进行展示。用户可以在工程运行时，使用手机中具备“扫一扫”功能的 app 或者直接使用二维码扫码枪，扫描并获取其展示的二维码信息。

二维码设置方法如下：

- ① 双击图 2.1-2 中需要设置二维码功能的窗口画面，进入窗口动画组态；
- ② 打开工具箱，在工具箱中选择“二维码”构件，并在窗口画面中绘制二维码构件，操作步骤如图 2.1-7 所示：

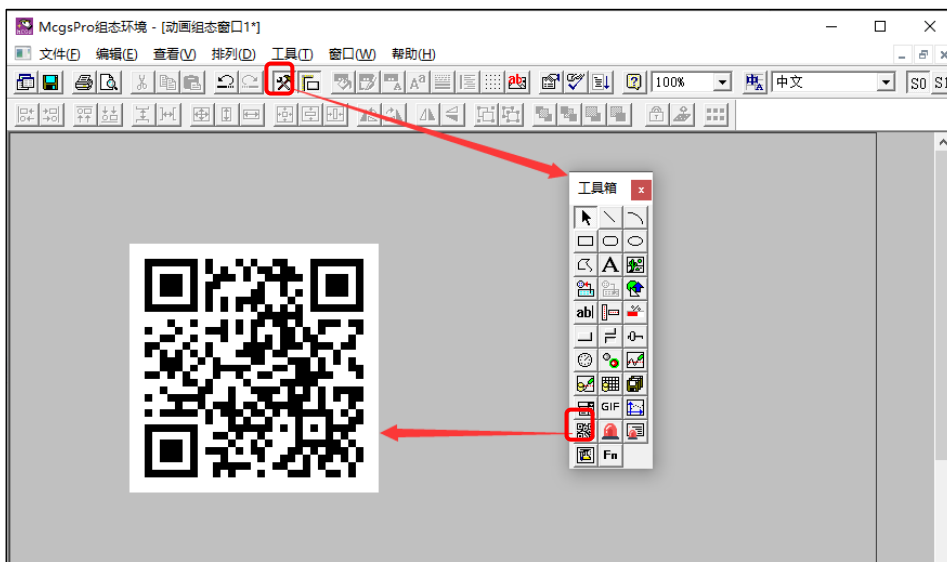


图 2.1-7 绘制二维码构件

- ③ 鼠标双击图 2.1-7 中的二维码构件，进入图 2.1-8 所示的二维码构件属性设置界面；

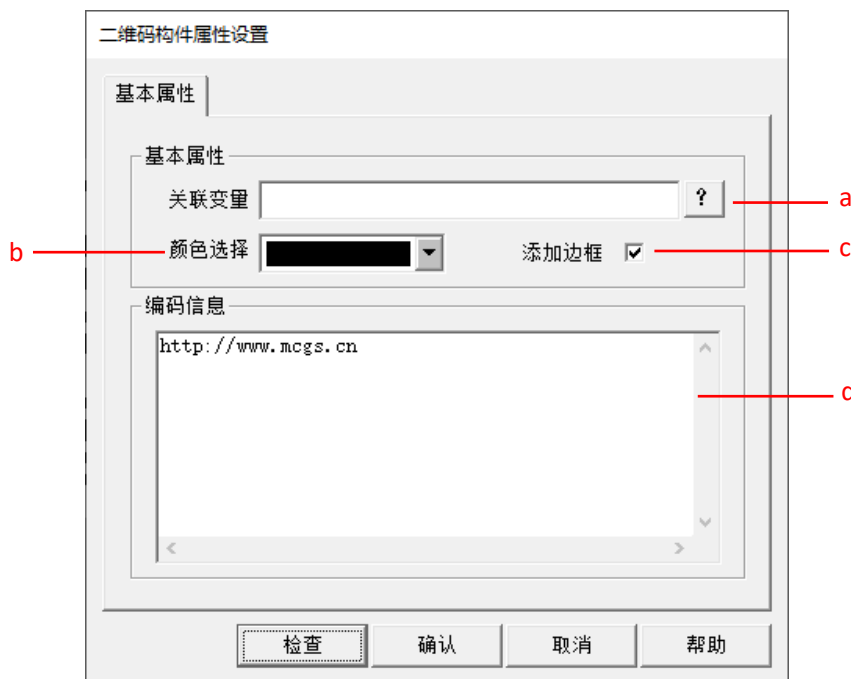


图 2.1-8 二维码构件属性设置

二维码属性配置信息

a. 关联变量：设置二维码的显示信息，如果该位置关联了变量，工程运行时，构件将以二维码的方式显示该变量的值；

b. 颜色选择：设置二维码构件的显示颜色；

c. 添加边框：选择是否显示二维码构件的白色边框；

d. 编码信息：设置静态文本，当未在【关联变量】处关联变量，构件在工程运行时将以二维码的方式显示【编码信息】中的内容。另外，组态时构件将始终显示【编码信息】中的内容。

注意：二维码构件可以通过方法函数 `SetFixRate(整数 1, 整数 2)` 设置纠错率。详细的构件方法函数的内容将在后续章节：[3.5 构件或窗口的属性和方法](#)进行介绍。

2.2 数据显示和输入

数据的显示和输入是用户进行工程组态时常用的功能，McgsPro 组态软件提供了专用的标签构件和输入框构件，实现数据的显示和输入功能。

2.2.1 数据显示

使用图 2.2-1 红框所示的标签构件可以显示工程中变量的值。



图 2.2-1 工具箱：标签构件

● 标签显示数据

- ① 首先打开用户窗口，在图 2.2-1 中选择点击【鼠标左键】，选择标签构件，并在用户窗口中绘制；

- ② 双击绘制的标签构件，进入【属性设置】。如图 2.2-2 所示，勾选【输入输出连接】→【显示输出】，进入“显示输出”选项卡：



图 2.2-2 标签属性设置

- ③ 如图 2.2-3 所示，通过点击图中的【？】，实现在表达式位置关联变量，即可实现工程运行时显示所关联变量的值。

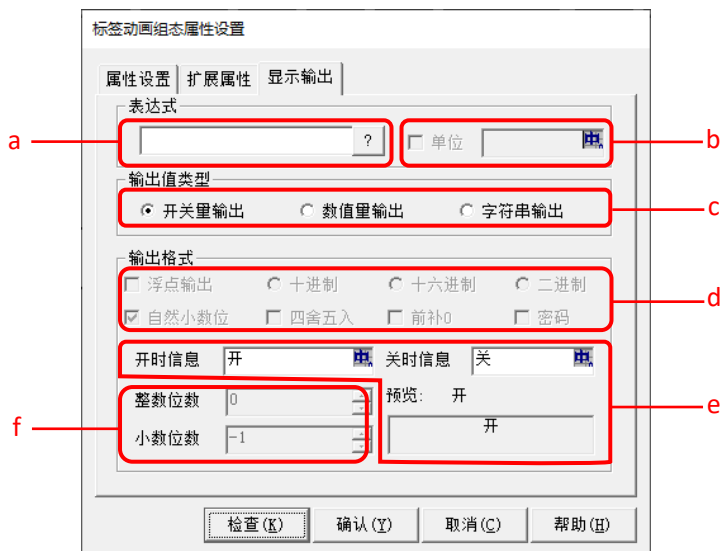


图 2.2-3 标签显示输出设置

显示输出配置信息

a. 表达式：工程运行时，标签构件将显示表达式中关联变量的值，支持的变量类型包括：整数、浮点数、字符串，或者是它们的表达式；

b. 单位：工程运行时，显示数据单位。只有当【表达式】中关联的变量是整数或浮点数，该选项才呈可用状态；

c. 输出值类型：【表达式】关联整数或浮点数，输出值类型才可选择【开关量输出】和【数值量输出】，【表达式】关联字符串，输出值类型只能选择【字符串输出】；

d. 输出格式：以设定的格式显示【表达式】中关联的变量，数据格式包括浮点输出、十进制、十六进制、二进制、自然小数位、四舍五入、前补 0、密码。

浮点输出：以浮点数方式显示【表达式】关联变量；

十进制、十六进制、二进制：以十进制、十六进制、二进制的方式显示【表达式】关联变量；

自然小数位：对显示的小数位的格式不做特殊要求，让系统自行决定小数位精度；取消勾选此项后，可手动指定“f”项中的【整数位数】和【小数位数】；

四舍五入：【表达式】关联变量的小数位数超过设定位数，选择直接舍去或进行四舍五入；

前补 0：【表达式】关联变量的整数位数低于设定值，将自动进行补 0；

密码：以 “*” 代替显示【表达式】中关联的变量。

e. 开时信息、关时信息：标签显示配置项“c”中的输出类型选择【开关量输出】时，当【表达式】关联的变量值为非 0，标签将显示“开时信息”中的文本，当【表达式】关联变量的值为 0，标签将显示“关时信息”中的文本。

2.2.2 数据输入

McgsPro 提供的“标签”和“输入框”构件都可以实现数据输入功能，下面首先利用标签构件完成数据输入。

● 标签输入数据

- ① 如图 2.2-1 所示，在“工具箱”中选择标签构件，并绘制；
- ② 双击绘制的标签构件进入其【属性设置】。如图 2.2-4 所示，勾选【输入输出连接】→【按钮输入】，点击进入“按钮输入”选项卡；



图 2.2-4 标签属性设置

- ③ 如图 2.2-5 所示，在标签构件的【按钮输入】→【对应数据对象的名称】中，点击“?”关联变量，工程运行时，可以通过点击标签构件，将数据输入到关联变量。



图 2.2-5 标签按钮输入设置

按钮输入配置信息

- a. 输入变量关联：支持整数、浮点数、字符串变量；
- b. 输入值类型设置：选择“开关量输入”，工程运行时点击标签构件，数据将以开关量（0 或非 0）的方式输入到关联变量；选择“数值量输入”，数据将以数值的方式输入到关联的变量；选择“字符串输入”，数据将以字符串的方式输入到关联变量；
- c. 输入格式：变量关联“整数”且输入类型为“数值量输入”，可指定输入数据进制（十进制、二进制、十六进制），变量关联字符串或自然小数格式浮点数，可指定输入数据格式为“密码”，实现以掩码“*”的方式输入；

提示信息：进行数据输入时键盘顶部显示的提示信息；

开时信息、关时信息：开关量输入时键盘上的提示文字配置；

最小值、最大值：运行时允许输入的数据范围，仅当输入类型选择“数值量输入”时有效。

● 输入框输入数据

- ① 如图 2.2-6 所示，在工具箱中选择输入框构件，并在用户窗口中绘制输入框构件；



图 2.2-6 工具箱：输入框构件

- ② 双击绘制的输入框构件，进入图 2.2-7 所示的【操作属性】，用户只需要在【操作属性】→【对应数据对象名称】中，点击  按钮关联变量，并设置好输入格式，即可在工程运行时通过点击输入框，将数据输入到关联的变量。

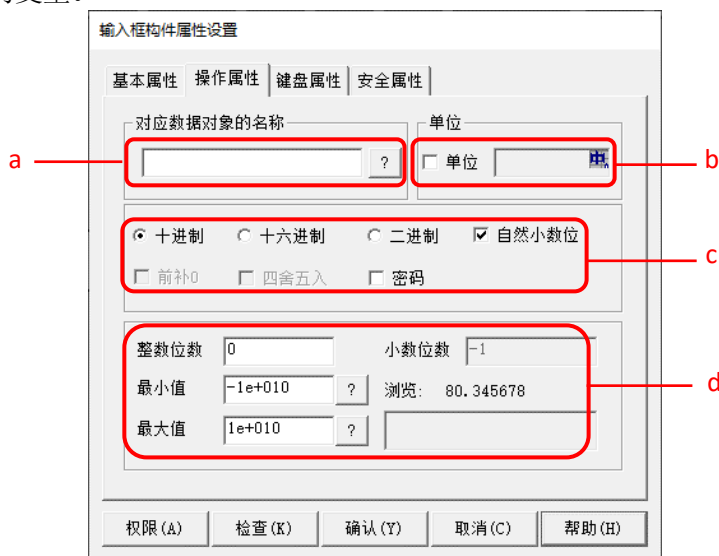


图 2.2-7 输入框操作属性设置

输入框操作属性配置信息

- a. 对应数据对象的名称：工程运行时通过点击输入框将数据输入到关联的变量，支持：整数、浮点数、字符串变量；
- b. 单位：勾选并设置后，工程运行时会自动在数据后追加显示单位，仅当关联变量为整数或浮点数时有效；
- c. 数据格式设置：

十进制、十六进制、二进制：以十进制、十六进制、二进制的方式显示和输入数据；

自然小数位：勾选后，系统对小数位数不做格式限制；不勾选，则需要在配置项“d”中手动指定小数位数；

前补 0：工程运行时，输入的整数位数小于配置项“d”中设置值，系统将自动进行补 0；

四舍五入：取消勾选自然小数位后，当输入数据小数位数超过配置项“d”中的设置值，选择是否进行四舍五入；

密码：选择是否以“*”的方式输入和显示数据；

d. 设置数据的整数位数、小数位数、允许输入的最小值和最大值、以及显示效果预览。

2.3 输入键盘设置

具有输入功能的构件（如输入框、标签构件）均可设置键盘类型，McgsPro 支持 3 种键盘输入方式：

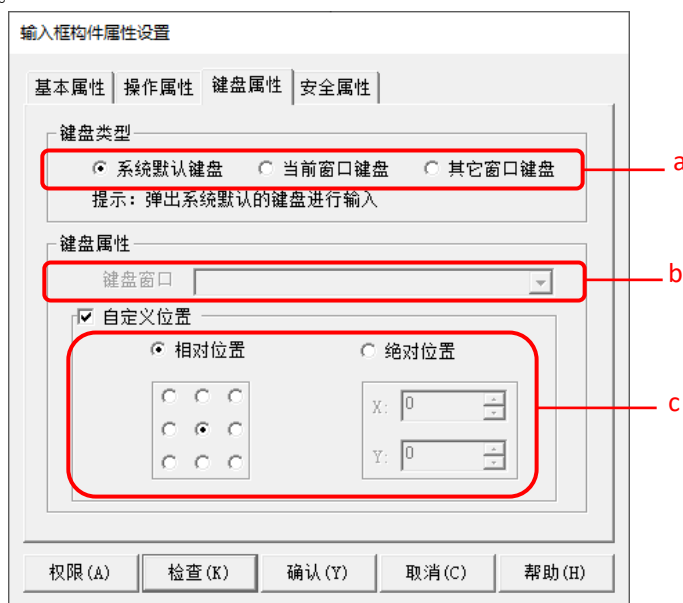
- 外接 USB 键盘实现数据输入：工程运行时激活构件的输入功能，即可通过外接 USB 键盘进行输入，无需额外设置；
- 系统默认键盘：系统键盘，输入构件默认选择此键盘；
- 自定义键盘：用户在工程组态时自定义的输入键盘。

本节主要讲解系统默认键盘、自定义键盘的使用方法。

2.3.1 系统默认键盘

以输入框构件为例，系统默认键盘设置步骤如下：

参考进阶篇章 2.2.2，双击绘制的输入框构件，进入属性设置界面。如图 2.3-1 所示，在【键盘属性】功能页，勾选【系统默认键盘】，工程运行时，点击该输入框，系统将根据用户在【操作属性】中关联的变量类型，弹出对应形态的输入键盘。



键盘属性配置信息

a. 弹出的【键盘类型】设置：

【系统默认键盘】→ 系统键盘；

【当前窗口键盘】→ 自定义键盘，系统不弹出键盘（键盘按钮就在输入框构件所在的用户窗口）；

【其他窗口键盘】→ 自定义键盘，系统弹出【键盘窗口】中指定的用户窗口（键盘按钮就在弹出的指定窗口中）；

b. 【键盘窗口】：仅键盘类型选择【其他窗口键盘】时有效。工程运行时点击该输入框构件，将弹出此处选择的用户窗口。

c. 弹出的键盘位置设置：

【相对位置】：弹出键盘相对于 TPC 液晶显示屏的位置；

【绝对位置】：按照用户设置的 XY 坐标弹出键盘；

2.3.2 自定义键盘

自定义键盘可以用来替代系统默认键盘，其本质上是【键盘按钮】放置在指定窗口，然后通过输入框等构件去调用这个窗口实现自定义键盘功能。自定义键盘中的【当前窗口键盘】和【其他窗口键盘】区别详见图 2.3-1 所示的键盘属性配置信息。

假设用户需要使用自定义键盘对一个浮点数进行数据输入，用户理想中的键盘外观如图 2.3-2 所示：

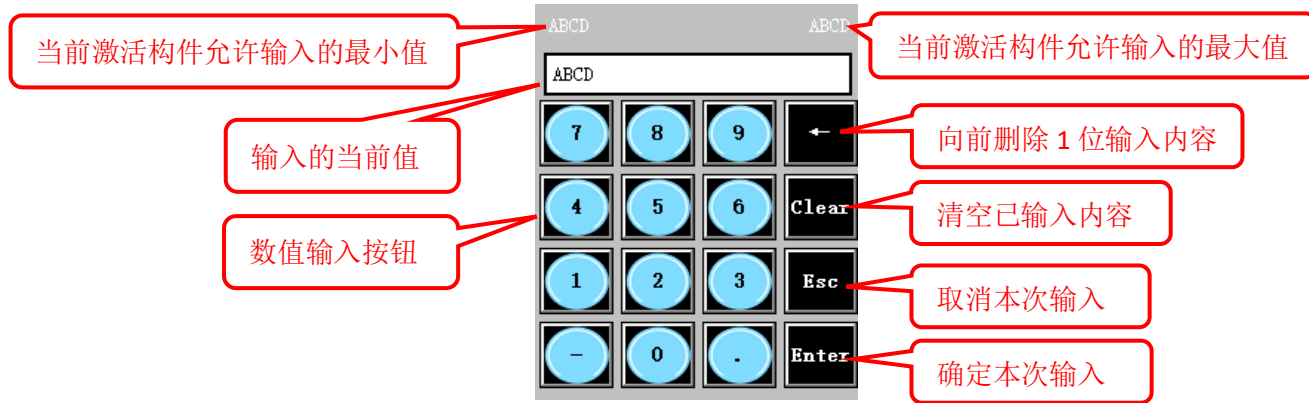


图 2.3-2 自定义键目标外观

2.3.2.1 发送指令组态

- ① 如图 2.3-3 所示，首先新建 2 个用户窗口，【数值键盘】窗口作为自定义键盘窗口；【测试窗口】窗口用于绘制输入框构件，测试自定义键盘弹出功能。注意将【测试窗口】设置为启动窗口（启动窗口设置参考进阶篇章节 2.1.1）：

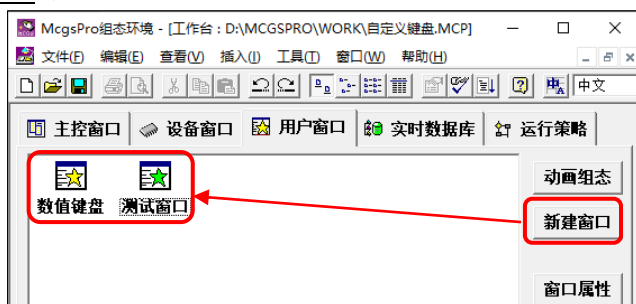


图 2.3-3 新建窗口

- ② 双击进入新建的【数值键盘】窗口组态界面。如图 2.3-4 所示，从【工具箱】中添加一个【键盘按钮】构件：

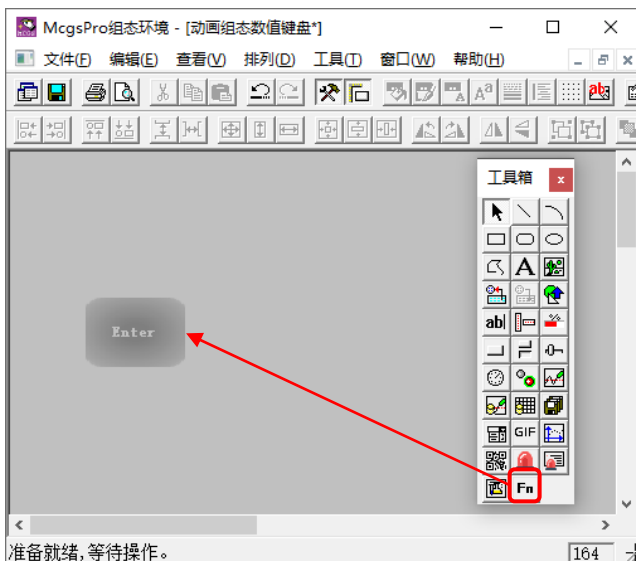


图 2.3-4 添加键盘按钮

- ③ 先添加 1 个输入“0”的按钮：双击绘制的键盘按钮构件，在基本属性功能页如图 2.3-5 所示，设置键盘按钮外观；

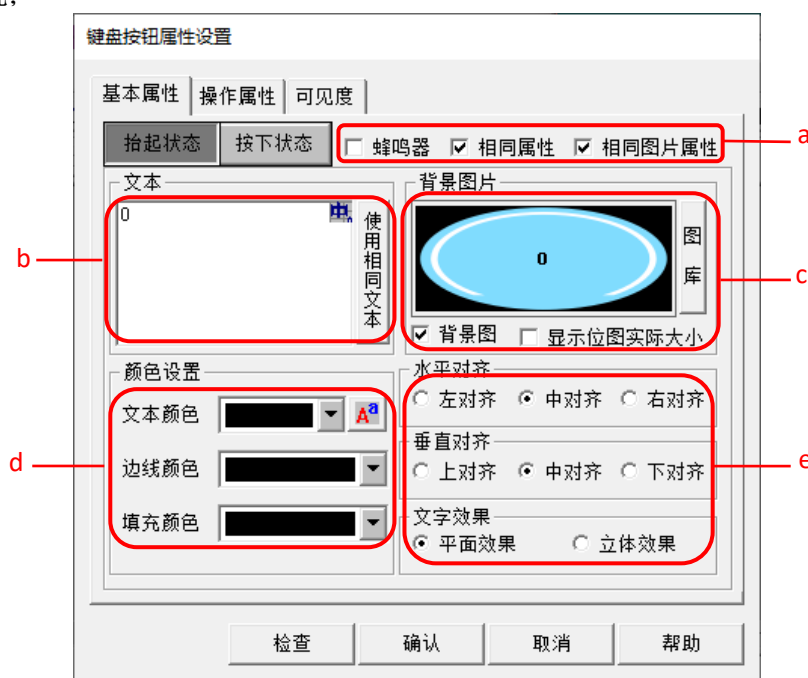


图 2.3-5 键盘按钮外观设置

基本属性配置信息

- a. 设置点击按钮是否触发蜂鸣；设置按钮【抬起状态】和【按下状态】的显示文本和背景图保持一致；
b. 设置抬起/按下状态的文本内容；

c. 设置构件背景图片属性，通过【图库】选择背景图或插入自定义背景图，本节所示图片插入方法如图 2.3-5 所示；

注意：通过图 2.3-6 中的【装入】按钮，可装载用户自定义图片，支持图片格式：bmp、jpg、png、ico、svg。

- d. 设置文本颜色和字体大小，以及构件边线颜色和填充颜色；
e. 设置文本对齐方式和显示效果。

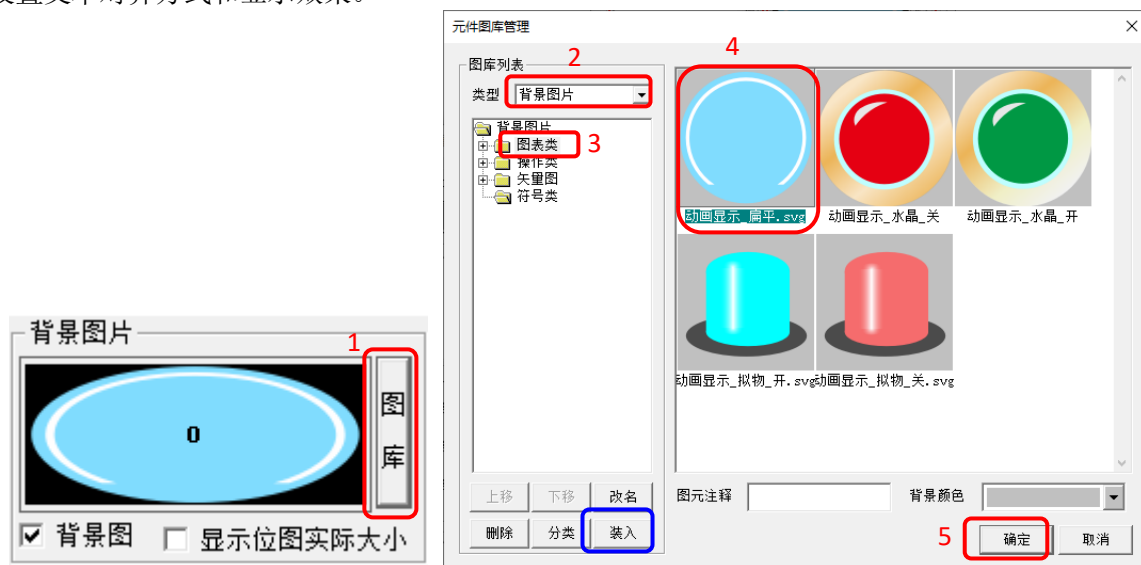


图 2.3-6 插入背景图

- ④ 点击构件【操作属性】，进入操作属性选项卡，设置该键盘按钮构件发送的指令内容。由于我们需要添加可输入“0”的键盘按钮，可选择【字符码】→【字符】→“0”，如图 2.3-7 所示：



图 2.3-7 键盘按钮操作属性

操作属性配置信息

- a. 选择功能码或字符码指令的发送时机；
b. 功能码：选择按下按钮时发送的功能码，功能码与字符码互斥；（功能码是具体的某项功能，不是字符）

【Enter】：确定本次输入；
【Backspace】：向前删除 1 位输入内容；
【Clear】：清空已输入内容；
【Esc】：取消本次输入；
【Delete】：向后删除 1 位输入内容；
【Left】：光标向左移动 1 个位置；
【Right】：光标向右移动 1 个位置。

- c. 字符码：选择按下按钮时发送的字符，字符码与功能码互斥，本次我们的字符码选择阿拉伯数字“0”；

- ⑤ 完成步骤 4，点击【确认】，按照需要调整构件大小，效果如图 2.3-8 所示：



图 2.3-8 键盘按钮：0

- ⑥ 重复步骤 2～步骤 5，依次添加字符码：“1”、“2”、“3”、“4”、“5”、“6”、“7”、“8”、“9”、“-”、“.”，以及添加功能码：“Backspace”、“Clear”、“Esc”、“Enter”。调整构件位置和大小，如图 2.3-9 所示：

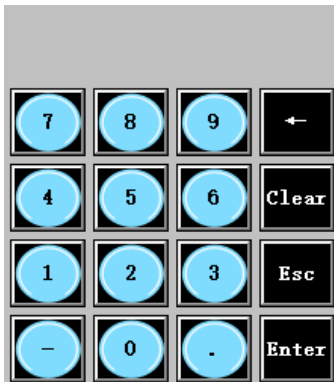


图 2.3-9 添加字符码和功能码

2.3.2.2 激活的输入键盘属性显示

完成上一节的发送指令组态操作后，该自定义键盘已经可以实现数值输入功能。同时用户还可以通过

McgsPro 提供的系统变量显示数据输入的过程值，以及键盘输入允许输入的最小值和最大值，设置方法如下：

- ① 在【数值键盘】窗口中添加 1 个标签构件，勾选【显示输出】，如图 2.3-10 所示：



图 2.3-10 标签：显示输出

- ② 如图 2.3-11 所示，在【显示输出】→【表达式】，输入系统变量“\$CurInput”，【输出值类型】选择字符串，设置完成后点击【确认】。此时运行工程，通过输入构件激活【数值键盘】窗口，该标签构件将显示：激活的输入键盘输入的当前值。

注意：标签显示输出配置信息可参考进阶篇章节 2.2.1。

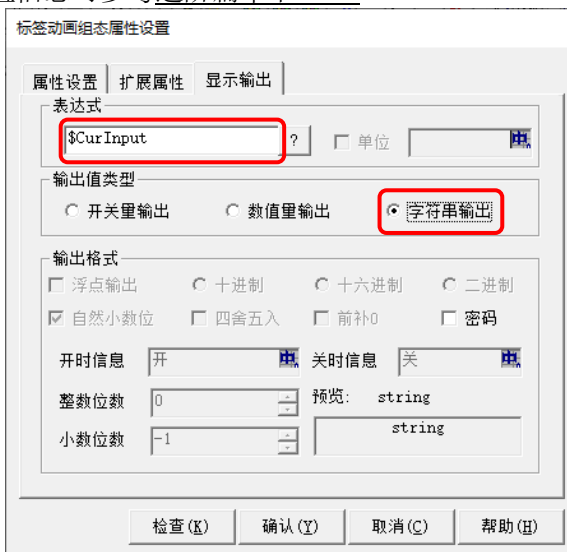


图 2.3-11 系统变量关联

- ③ 按照同样的方式，再添加 2 个标签构件，重复步骤①～步骤②，分别在其在【显示输出】→【表达式】，输入以下字符串：“Min:”+\$MinInput; “Max:”+\$MaxInput。并调整标签构件大小、填充色等外观，效果如图 2.3-12 所示：

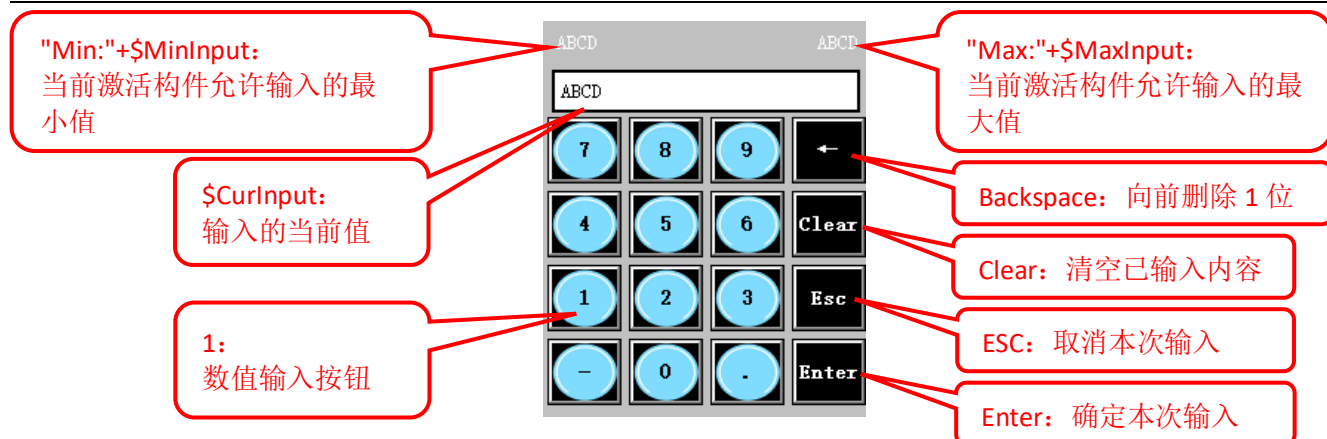


图 2.3-12 完成自定义键盘

2.3.2.3 自定义键盘弹出方式

自定义键盘的弹出方式可以自行定义，大多数用户一般是选择将其单独作为一个子窗口弹出，设置方法如下：

- ① 在自定义键盘所在的窗口【数值键盘】空白处双击，进入用户窗口属性设置；
- ② 如图 2.3-13 所示，勾选【作为子窗口使用】，根据需要设置窗口模式【模态式窗口】，并设置子窗口打开的大小（子窗口打开范围应包含所有构件）；

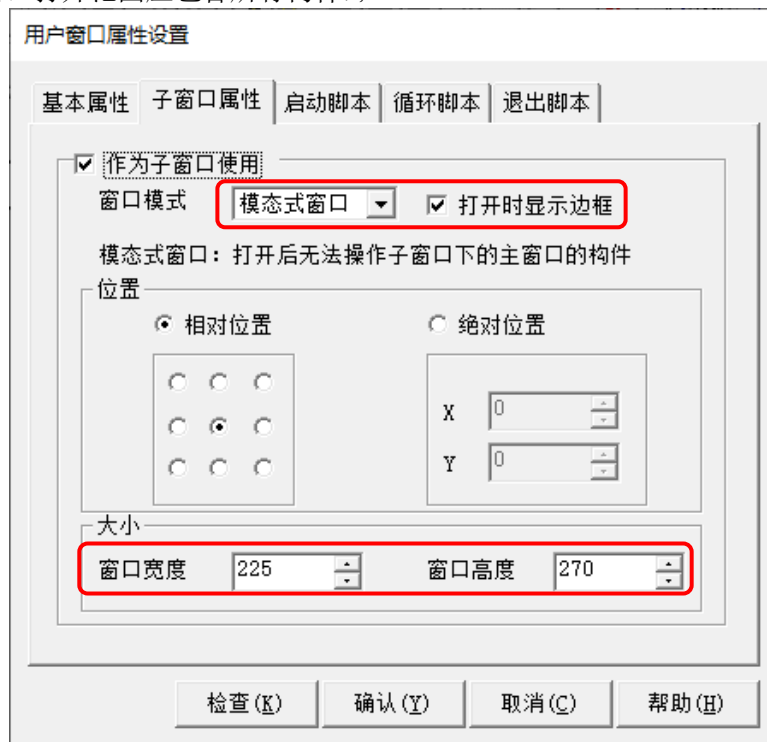


图 2.3-13 窗口打开方式

- ③ 设置完成，窗口【数值键盘】如图 2.3-14 所示：



图 2.3-14 子窗口模式打开自定义键盘

2.3.2.4 测试自定义键盘

- ① 在工程的【实时数据库】新建一个浮点数变量，变量名：“测试”；
- ② 双击图 2.3-2 所示的【测试窗口】，进入其组态画面。在该窗口中添加一个输入框构件，在其操作属性关联变量“测试”；设置【最小值】：-500；最大值：500；进入【键盘属性】选项卡，选择键盘窗口为【数值键盘】。如图 2.3-15 所示：

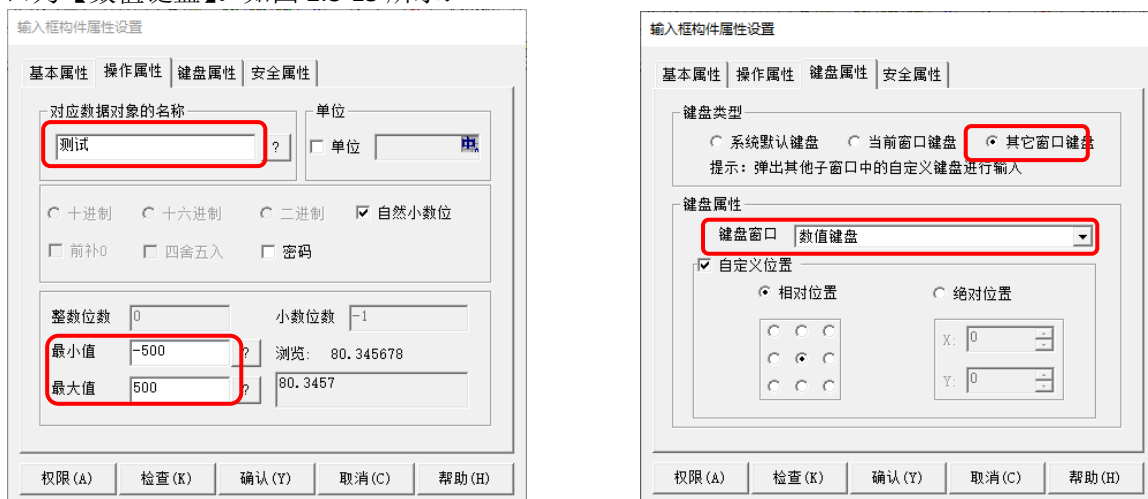


图 2.3-15 输入框键盘设置

- ③ 设置完成后，保存并模拟运行工程，点击图 2.3-16 设置的输入框构件，将弹出 2.3-15 所示的自定义键盘窗口：

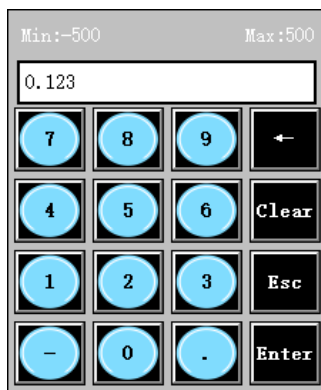


图 2.3-16 自定义键盘窗口

注意：用户可通过自定义键盘进行字符串输入方式与之类似，只需要将图 2.3-6 中的字符码修改为对应的字符即可，字符可以是汉字、字母（区分大小写）、标点、特殊符号。

2.4 公共窗口功能

“公共窗口”可以看作被其他窗口引用的模板窗口。与公共窗口概念相对应的是“宿主窗口”，即引用公共窗口的用户窗口。同一个工程，任何一个用户窗口都可以被指定为另一个用户窗口的公共窗口，公共窗口中所有的构件均可被宿主窗口继承，并显示在宿主窗口的最底层。

比如，用户需要在每个用户窗口的右上角显示系统时间，用户可以选择手动在每个窗口的右上角添加时间显示构件；也可选择新建一个窗口，将时间构件放在新建窗口右上角，其他窗口直接指定新建窗口为公共窗口即可。操作步骤如下：

- ① 运行 McgsPro，新建工程，再新建 2 个用户窗口，如图 2.4-1 所示：

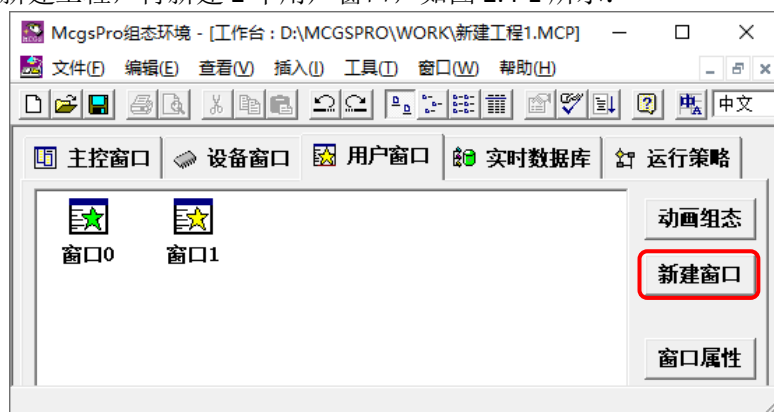


图 2.4-1 新建用户窗口

- ② 本节教程选择“窗口 0”作为宿主窗口，“窗口 1”作为公共窗口。双击窗口 1，进入窗口 1 组态画面；
- ③ 按照图 2.4-2 所示步骤，在“窗口 1”画面右上角添加一个时钟，添加完成后保存工程；

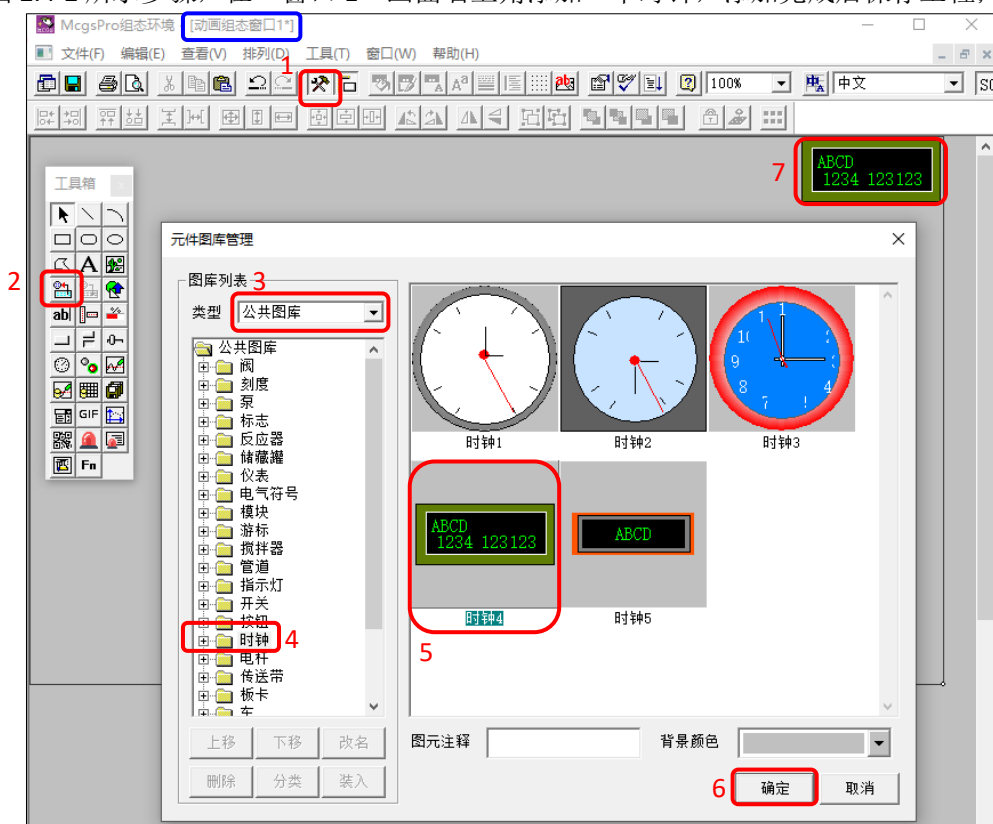


图 2.4-2 添加时钟

- ④ 如图 2.4-3 所示，在窗口属性设置界面依次执行操作：【基本属性】→【公共窗口】→ 选择“窗口 1”→【确定】。将“窗口 1”设置成“窗口 0”的公共窗口模板。

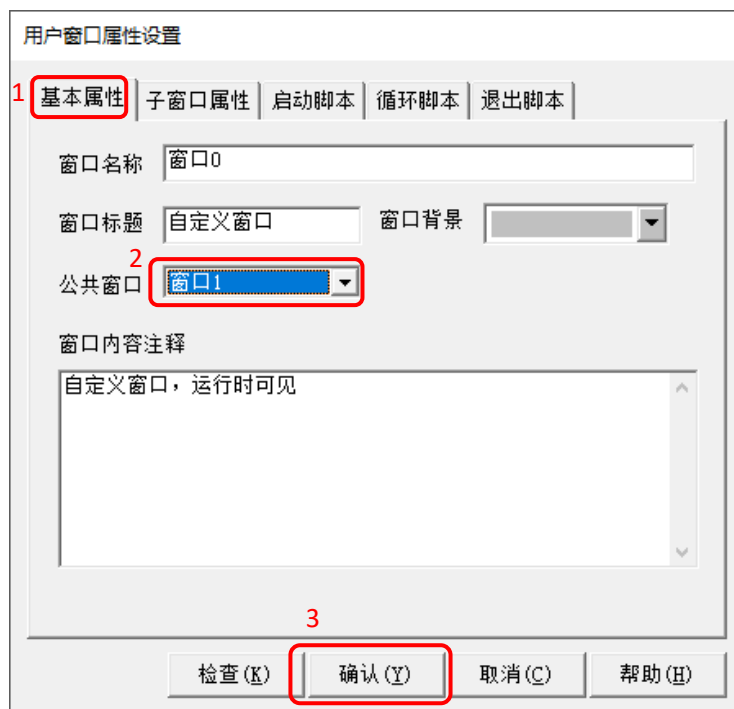


图 2.4-3

- ⑤ 完成图 2.4-3 所示操作后，“窗口 0”显示如图 2.4-4 所示。可以看到在宿主窗口“窗口 0”的右上角，显示了指定的公共窗口“窗口 1”中组态的时钟显示构件。

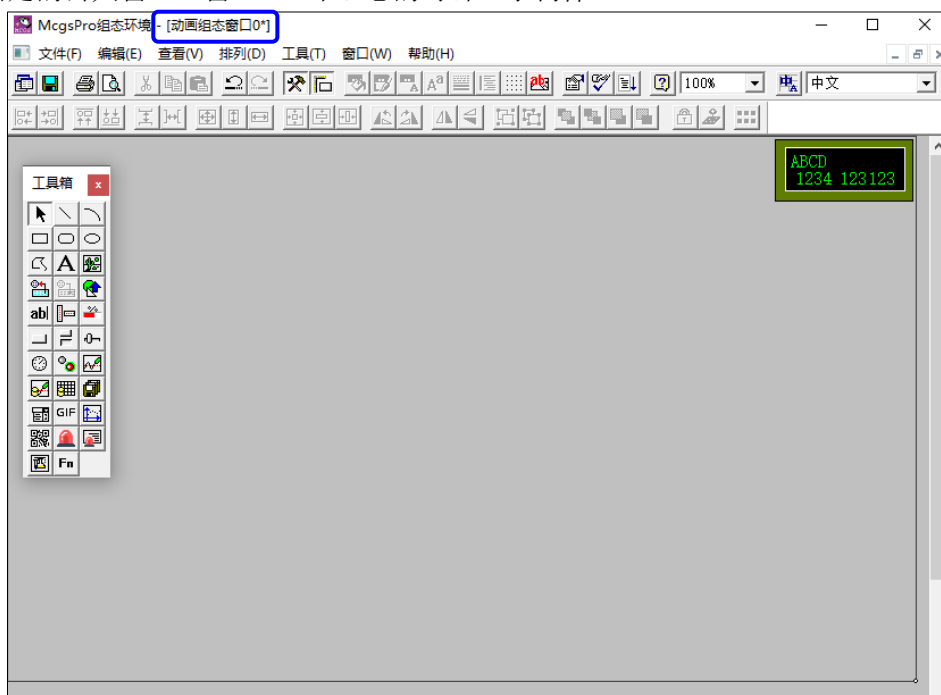


图 2.4-4 窗口 0 画面展示

公共窗口设置注意事项：

- 公共窗口中的构件、窗口脚本，可在宿主窗口中正常显示或执行；
- 公共窗口中的构件显示在宿主窗口的底层，且只能在公共窗口中编辑；
- 公共窗口的背景色不会在宿主窗口中显示；
- 1 个公共窗口可被多个宿主窗口引用，但 1 个宿主窗口只能引用 1 个公共窗口。

2.5 条件满足自动弹窗

现场经常会有客户要求在不正常情况下，触摸屏可自动弹出提示窗口或者执行某些指令。其本质是在工程运行时监视指定的对象（变量或组对象），当监视对象产生报警、数据跳变、数据变化或者满足其他指定的条件时，触发预置动作。动作内容用户可自行定义，如打开某个窗口，或执行某条指令。

2.5.1 条件满足弹窗功能组态

McgsPro 提供运行策略模块供用户使用，不同的运行策略对应不同的触发方式，本节教程采用事件策略。当现场“阀门开关状态”发生正跳变时（即阀门由关闭变为打开状态：0 → 非0），触摸屏以子窗口的方式弹出提示。

- ① 新建工程，如图 2.5-1 所示，在工程的实时数据库中新建整数变量“阀门开关状态”；

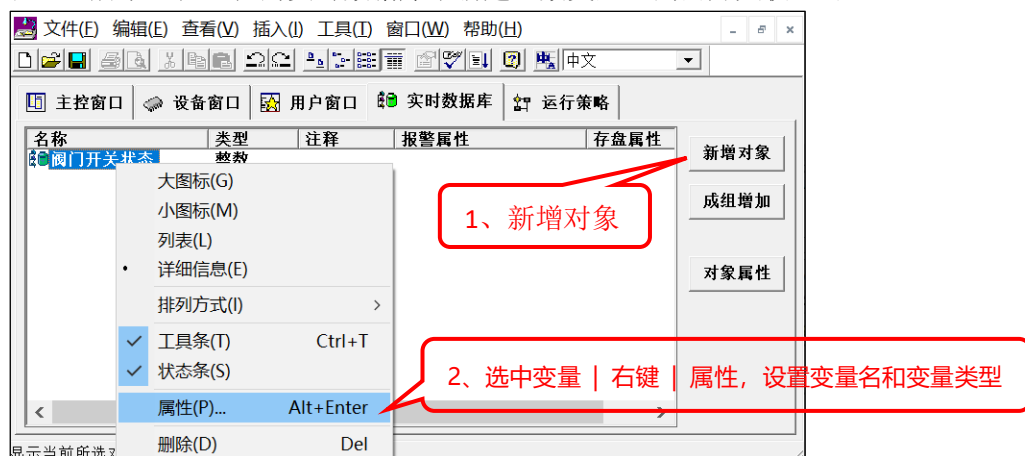


图 2.5-1 新增整数变量

- ② 在【用户窗口】新建 2 个窗口画面，进入“窗口属性设置”修改窗口名称，如图 2.5-2 所示：

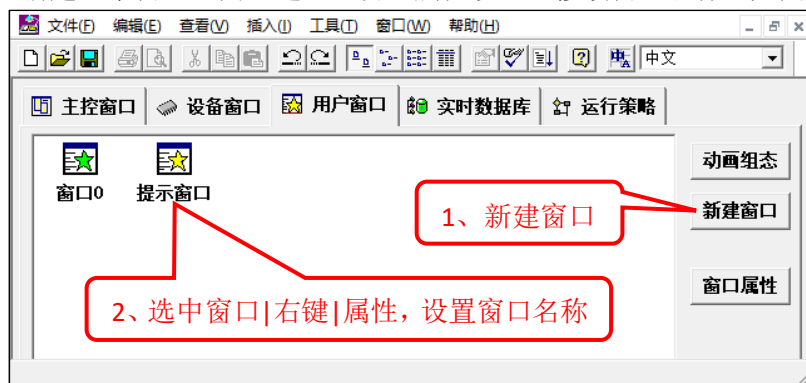


图 2.5-2 新建用户窗口

- ③ 进入运行策略模块，新建一个事件策略，如图 2.5-3 所示：

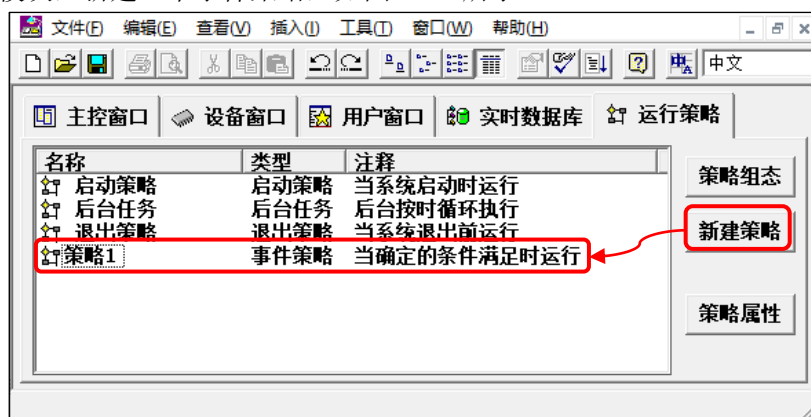


图 2.5-3 新建事件策略

④ 双击图 2.5-3 新建的【策略 1】，进入策略 1 的“策略组态”界面；

⑤ 在策略组态界面双击 ，进入“属性设置”。如图 2.5-4 所示，通过【策略执行方式】→【数据对象】→【?】，关联需要监视的变量“阀门开关状态”；【执行条件】选择“正跳变时，执行一次”（变量值由 0 → 非 0）；设置完成点击“确认”。

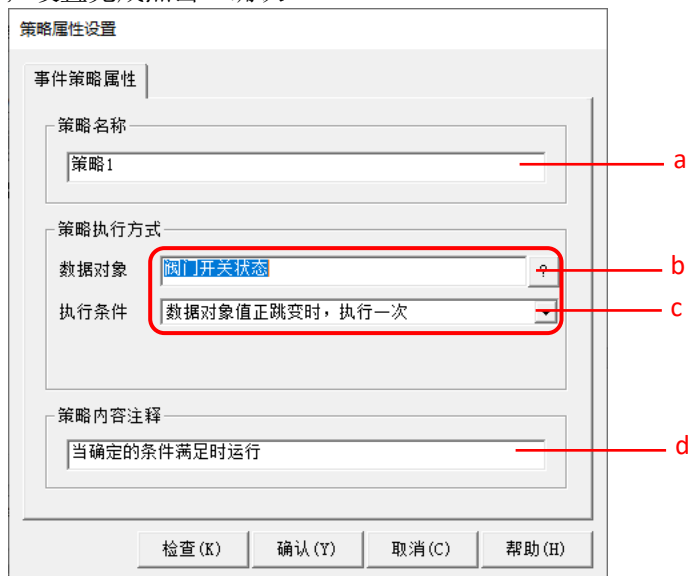


图 2.5-4 设置策略执行方式

事件策略配置信息

a. 策略名设置

b. 关联被监视的数据对象，可以是变量或表达式

c. 设置策略的执行条件，当监视的“数据对象”变化情况满足“执行条件”，就执行一次该策略中的“策略行”

正跳变：0 → 非 0

负跳变：非 0 → 0

正负跳变、负正跳变：McgsPro 不支持正负跳变和负正跳变，考虑版本兼容性故保留该选项有改变：数据对象值发生变化

d. 设置策略内容注释

⑥ 如图 2.5-5，在策略组态界面空白处点击【右键】→【新增策略行】。“策略 1”满足触发条件，将执行策略行中的脚本程序。

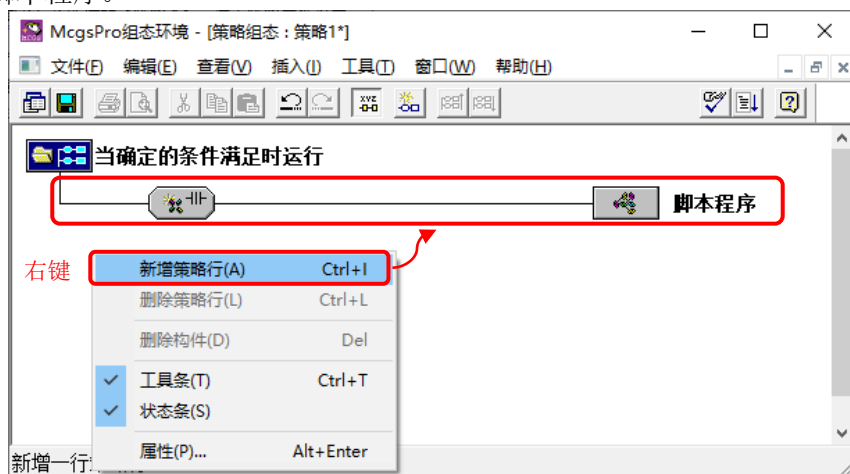


图 2.5-5 新增策略行

⑦ 双击新增策略行中的脚本程序图标 ，如图 2.5-6，在脚本程序编辑页面右侧对象树，依次展开【用户窗口】→【提示窗口】(图 2.5-2 新建的用户窗口)→【方法】，双击“open”，将其添加到脚

本编辑框。设置完成点击  检查脚本，通过后点击  保存，并关闭脚本程序编辑页面。

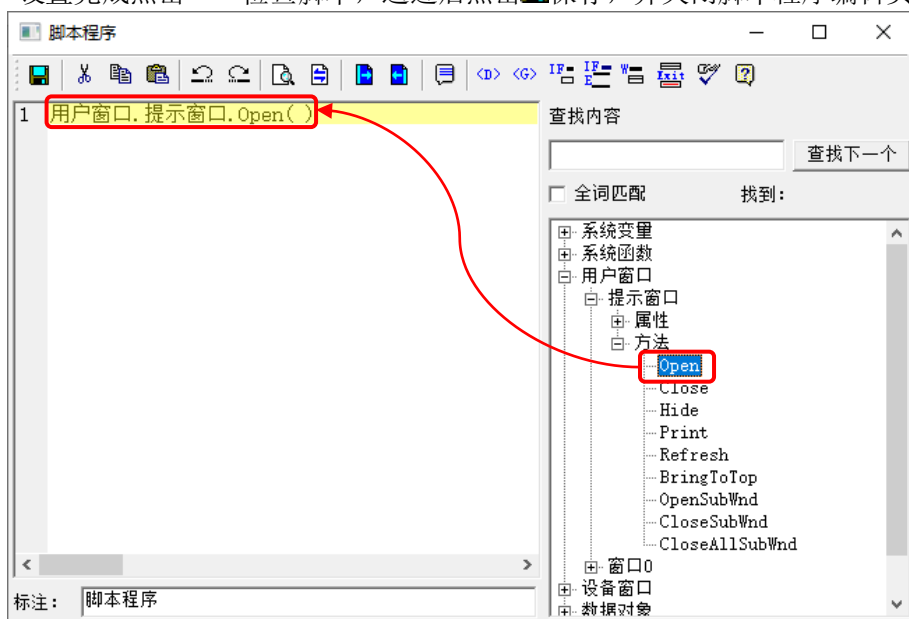


图 2.5-6 添加打开窗口脚本

注意：此时执行该策略行，将打开用户窗口：【提示窗口】



- ⑧ 回到工作台用户窗口模块，双击  提示窗口 进入其组态画面。进行以下操作：
- A、按照需要添加提示内容，用于提示窗口弹出后显示；
 - B、按照需要添加一个标准按钮构件，按照需要修改构件显示文本。双击标准按钮构件 → 【脚本程序】 → 【抬起脚本】，添加关闭【提示窗口】脚本“用户窗口.提示窗口.Close()”，添加方法参考图 2.5-6；
 - C、在【提示窗口】画面空白处双击 → 【用户窗口属性设置】 → 【子窗口属性】，将其设置为子窗口，子窗口大小自行定义；（子窗口设置方法详见进阶篇章节 2.3.2.4）
- 完成步骤 A、B、C 后，提示窗口画面效果如图 2.5-7 所示：



图 2.5-7 提示窗口组态



- ⑨ 最后在工作台双击启动窗口 **窗口0**，并在【窗口0】添加输入框，关联变量【阀门开关状态】(用于工程运行时演示提示窗口弹出效果)，输入框变量关联方法参考进阶篇章 [2.2.2](#)。
- ⑩ 完成以上步骤，运行工程，通过【窗口0】中的输入框将变量“阀门开关状态”先赋值“0”，再赋值为非0，可触发事件策略“策略1”中组态的脚本“用户窗口.提示窗口.Open()”，打开【提示窗口】，如图 2.5-8 所示：

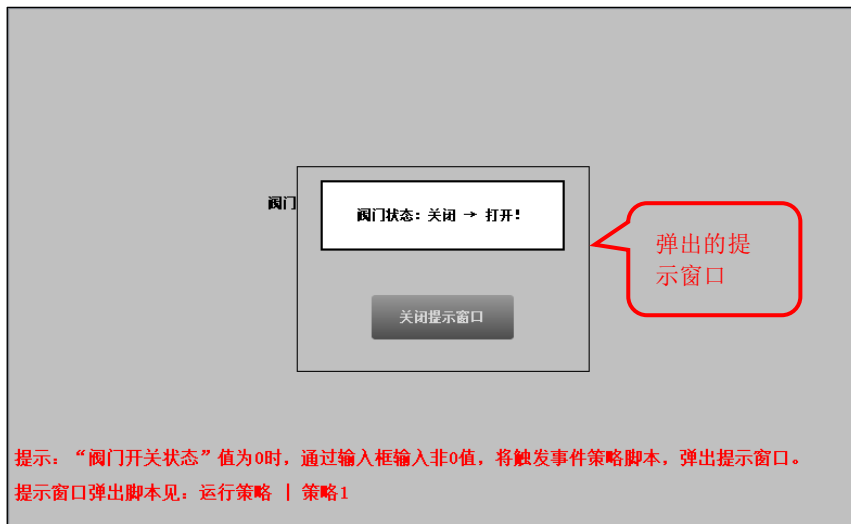


图 2.5-8 提示窗口弹出

注意：

- 用户如果希望满足条件后，TPC 可执行特定的脚本程序，都可以通过类似的方法实现。只需要用户选择满足自身要求的策略，并在该策略的【策略行】→【脚本程序】，添加待执行的脚本即可，策略行添加方法参考本节教程步骤⑤；
- 不同的策略具备不同的触发条件，McgsPro 提供八种策略供用户使用，详细内容请参考章节：[进阶篇 1.4.5 运行策略介绍](#)。

第3章 动画功能应用实例

本章主要介绍几个简单常用构件的使用方法和应用实例，以及构件或窗口的属性和方法功能、事件功能。

3.1 模拟仪表显示

旋转仪表构件是模拟旋转式指针仪表的一种动画图形，用其显示所连接的整数和浮点数变量的值。旋转仪表构件的指针随变量值的变化而不断改变位置，指针所指向的刻度值即为所连接的变量的当前值。

本节将使用软件自带的功能，组态一个美观大方、富有科技感的旋转仪表用于设备压力值的显示。

- ① 将一个旋转仪表构件拖拽到画面合适的位置和合适的大小（注意：图中的旋转仪表不是默认样式，是为了更好的区分仪表的不同区域，特地做了设置）。标准的旋转仪表构件本身由 A.圆边，B.标注，C.次划线，D.主划线，E.警示色环，F.指针，G.轴心，H.背景图，I.扇形等 9 部分组成，如图 3.1-1 所示。
- ② 双击旋转仪表构件，弹出【旋转仪表属性设置】界面，打开【基本属性】功能页，如图 3.1-1 所示。

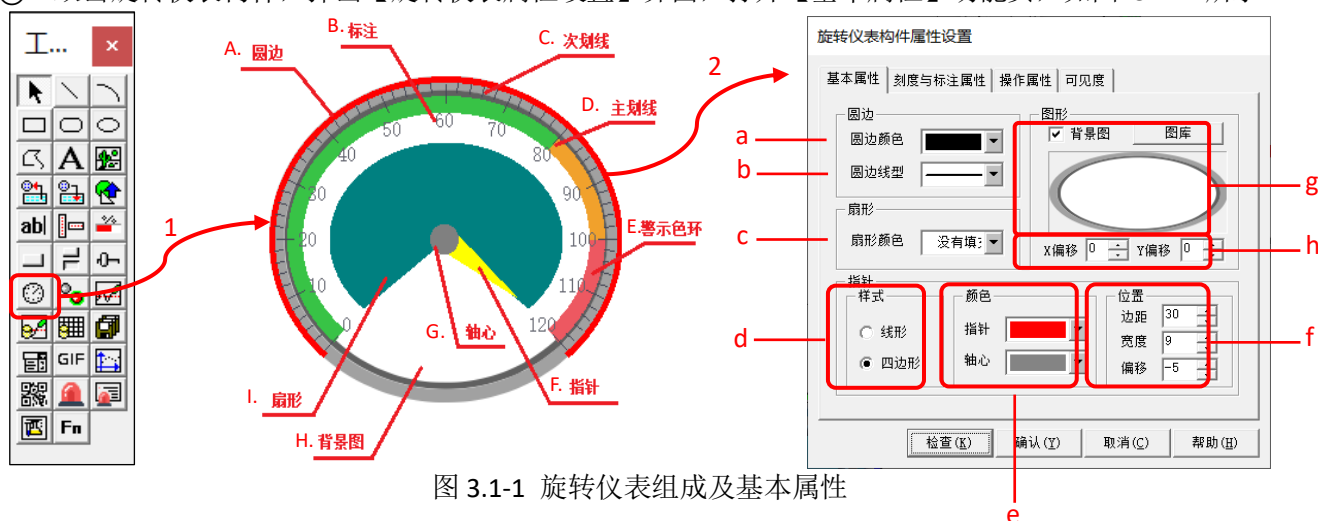


图 3.1-1 旋转仪表组成及基本属性

基本属性配置信息

基本属性功能页界面如图 3.1-1 所示。

- a. 圆边颜色：配置圆边的颜色
 - b. 圆边线型：配置圆边的线型
 - c. 扇形颜色：配置扇形的颜色
 - d. 指针样式：可选择线型和四边形两种
 - e. 指针颜色：可分别配置指针和轴心的填充颜色
 - f. 指针位置：可通过配置指针的边距，宽度，以及偏移长度来确定指针的位置，边距是指针到矩形绘制区域的内切椭圆的距离，偏移长度是指偏移轴心的距离，可根据实际情况调整这三个参数来确定指针的位置和合适的大小
 - g. 背景图：选择之后可以添加背景图片，支持 bmp, jpg, png, svg, ico 五种格式
 - h. X(Y)偏移：位图中心坐标的 X(Y)坐标减去旋转仪表中心坐标的 X(Y)坐标的值
- ③ 点击【图库】按钮打开【元件图库管理】。在【图库类型】处选择【水晶风格】。在下方的目录树中选择【仪表】。选择【PNG_表盘 2】。点击确定，关闭界面。



图 3.1-2 从库中选择仪表背景

- ④ 将【主划线】数目设置为【10】，【标注颜色】设置为【白色】以突出字体显示。

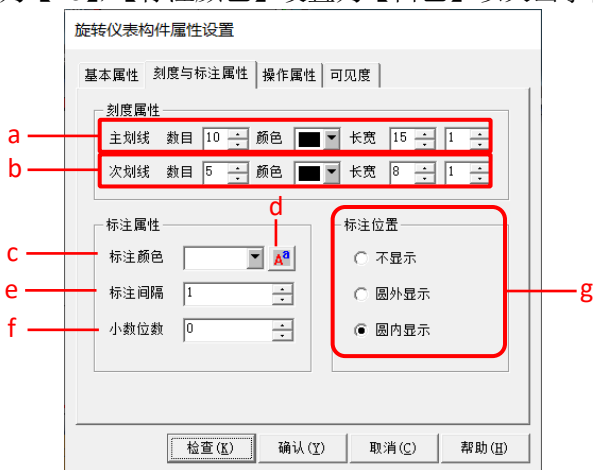


图 3.1-3 刻度与标注属性界面

刻度与标注属性配置信息

- 主划线：配置主划线的数目，颜色，长，宽
 - 次划线：配置次划线的数目，颜色，长，宽
 - 标注颜色：配置标注文字的颜色
 - 标注字体：配置标注的字体，字形，大小
 - 标注间隔：配置标注的间隔
 - 小数位数：配置标注显示浮点数时，小数显示的位数
 - 标注位置：可以选择标注显示在圆内，圆外，不显示
- 在实时数据库中新增浮点型变量【压力值】。然后在【操作属性】功能页中，将【表达式】设置为变量【压力值】。
 - 将【偏移范围】中的【对应值】设置为 1000，它表示表盘的最大显示值为 1000。
 - 把色环的大小调到合适的数值，本样例设置色环【宽度】为 6，【边距】为 0。
 - 根据工程需要设置【提示值】和【警告值】，本样例将其设置为 700 和 900。
 - 根据工程需要设置色环【颜色】，本样例设置如图 3.1-4 所示。



图 3.1-4 操作属性界面

操作属性配置信息

- a. 表达式：旋转仪表构件所对应的浮点数变量，变量的值和指针的位置成一一对应的关系
- b. 起点位置：指针偏转角度范围的确定，可基于起点位置逆时针和顺时针角度设置来确定，即最大角度=逆时针角度+顺时针角度，可选择 12 点，3 点，6 点，9 点，分别对应 90 度，0 度，270 度，180 度，可根据实际情况选择不同的起点位置
- c. 偏移范围：偏移范围是基于起点位置设置指针的偏转角度范围，同时可以设置逆时针角度和顺时针角度的数据值，即边界值
- d. 是否显示（警示色环）：选择后，才可以正常绘制警示色环
- e. （警示色环）大小：可设置警示色环的宽度和边距
- f. （警示色环）界限：这里的目的是将色环分为三个级别，分别表示正常、提示和警告。一般而言，[最小值，提示值]区间为正常色环，[提示值，警告值]区间为提示色环，[警告值，最大值]为警告色环
- g. （警示色环）颜色：可以分别设置三个级别的颜色

⑩ 点击确认保存设置并关闭设置界面。

⑪ 把一个标签构件拖拽到指针的上方，将它的【文本内容输入】设置为【Pa】，【填充颜色】设置为【没有填充】，【边线颜色】设置为【没有边线】，字体大小设置为【四号】。

⑫ 为了为大家演示旋转仪表的运行效果，我们在旋转仪表的下方放置一个【输入框】构件，并将其【操作属性】功能页中的【对应数据对象的名称】关联为变量【压力值】。

设置好的旋转仪表如图 3.1-5 所示：

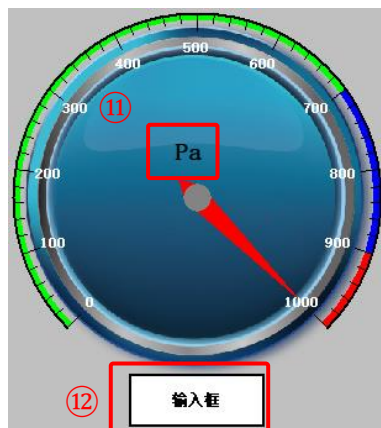


图 3.1-5 旋转仪表组态效果图

⑬ 保存工程后，下载工程，然后启动运行。在输入框中输入一个 0-1000 的值，可以看到指针会随之指到相应的位置。

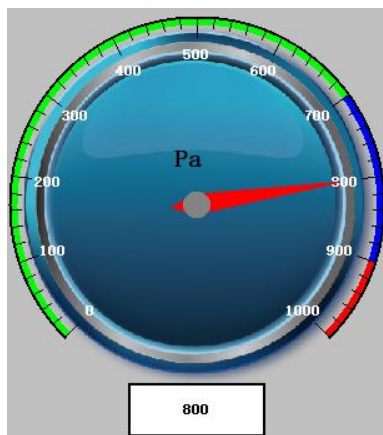


图 3.1-6 旋转仪表运行效果图

3.2 动画功能组态

3.2.1 GIF

当用户需要使用一个现有的动画效果时，可以使用 GIF 构件加载 GIF 动画，运行时可以多次显示或者循环显示。下面以加载一个风扇动画为例为大家进行讲解。



图 3.2-1 GIF 构件属性设置界面

- ① 将 GIF 构件拖拽到画面中合适的位置和合适的大小，双击该构件，打开【GIF 构件属性设置】界面。如图 3.2-1 所示。
- ② 点击【添加 GIF 图片】按钮。如图 3.2-1 所示。
- ③ 在弹出的对话框中，选择自定义 GIF 动画，完成后点击【打开】按钮，添加成功。如图 3.2-2 所示。
- ④ 添加 GIF 动画后，属性设置界面上会出现图片预览，供用户查看效果。

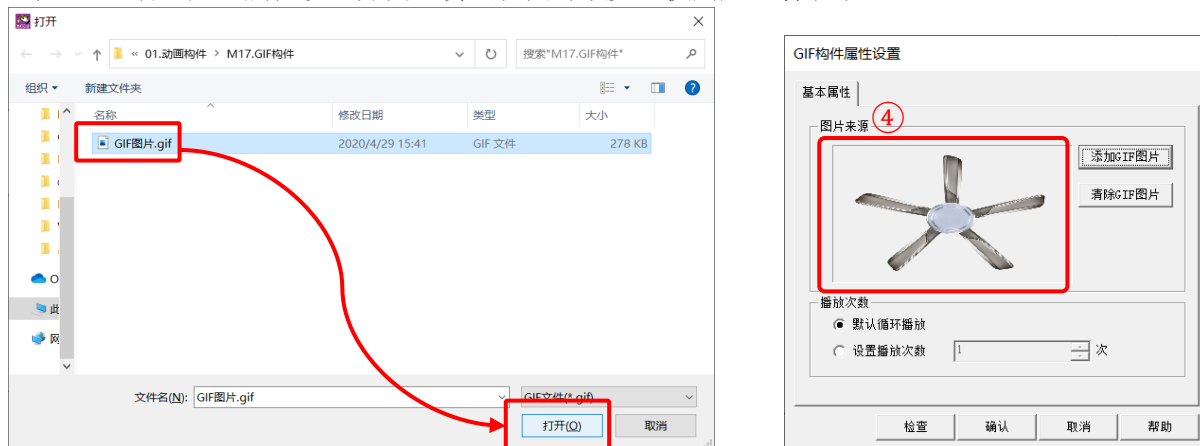


图 3.2-2 从文件选择 GIF 动画及预览效果

- ⑤ 在【播放次数】处，可以选择是【默认循环播放】还是【设置播放次数】，设置播放次数的范围是：1~999999999。本次我们选择默认设置。
- ⑥ 点击属性设置界面下方的【确认】按钮后，GIF 动画添加成功，效果如图 3.2-3 所示。

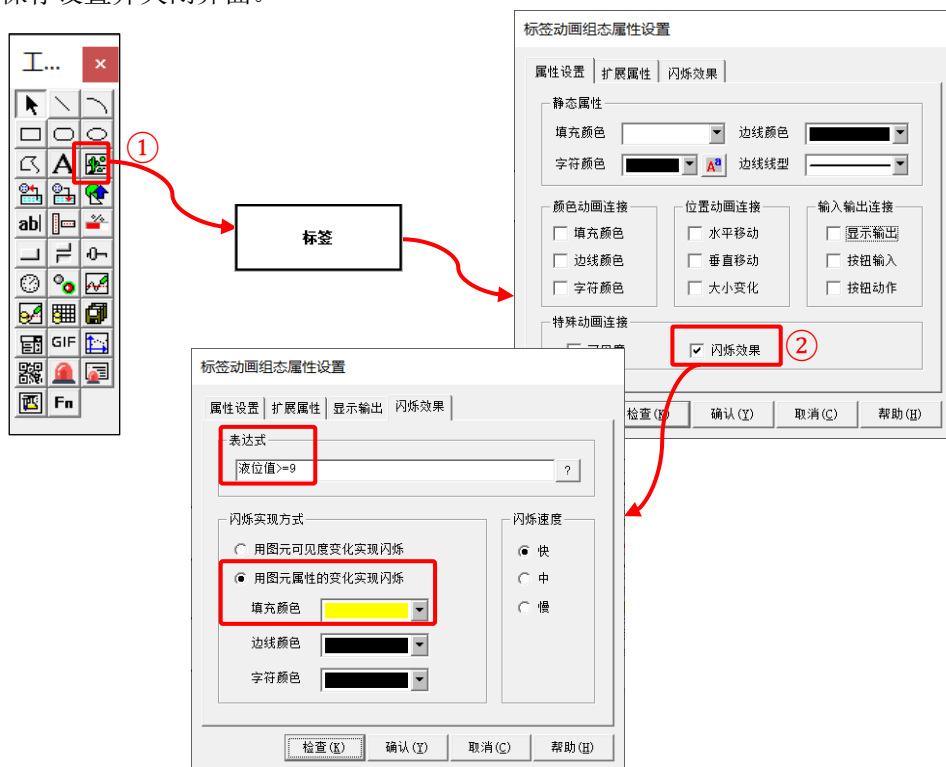


图 3.2-3 GIF 动画运行效果

3.2.2 闪烁

假设有一个水池的液位高度为 10 米，当它的液位超过 9 米时，画面中会弹出一个显示文字【液位高!!】的提示框。并且提示框的背景颜色不断闪烁，而文字也会由右向左移动。下面将为大家进行组态讲解。本节 3.2.2 至 3.2.4 使用的是同一个工程。

- 用一个标签构件关联变量【液位值】，当【液位值】 ≥ 9 时，标签闪烁，当【液位值】 < 9 时，标签停止闪烁。
 - ① 首先在实时数据库中新建一个浮点型变量【液位值】，然后拖拽一个【标签】构件到画面中。双击该构件打开【标签动画组态属性设置】界面。
 - ② 勾选【闪烁效果】前的复选框，点选出现的【闪烁效果】功能页。设置表达式为【液位值 ≥ 9 】，选择【闪烁实现方式】为【用图元属性的变化实现闪烁】，将【填充颜色】设置为【黄色】。点击确认保存设置并关闭界面。



3.2-4 闪烁功能设置流程图

- 设置完标签构件的闪烁效果后，我们还需要设置一个输入框，用它来模拟设置变量【液位值】的值。
 - ① 将一个【输入框】构件拖拽到画面中合适的位置，双击【输入框】构件，弹出【输入框构件属性设置】界面。
 - ② 在【操作属性】功能页中关联【对应数据对象的名称】为【液位值】。点击确认关闭界面。



3.2-5 输入框关联变量流程图

● 运行展示

保存工程后，下载工程，然后启动运行。将输入框中的值设置为 10，则标签构件的填充颜色在黄色和白色之间来回切换，实现闪烁效果。

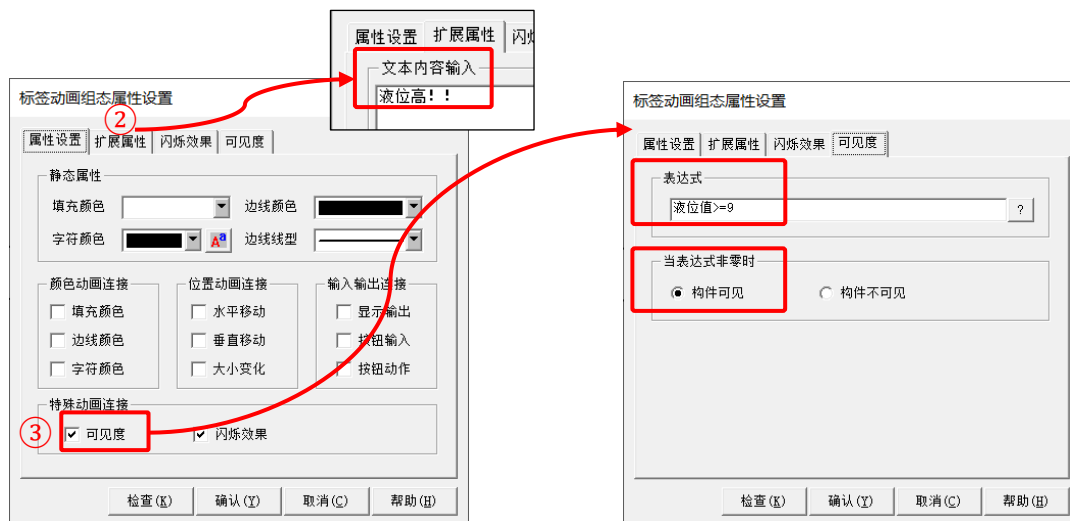


3.2-6 闪烁功能运行效果

3.2.3 可见度

● 用标签构件显示文字【液位高!!】，当【液位值】>=9 时，标签出现，当【液位值】<9 时，标签消失。

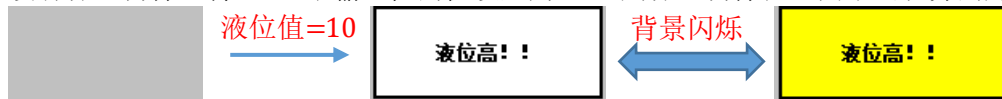
- ① 双击 3.2.2 中的标签构件，弹出【标签动画组态属性设置】界面。
- ② 选择【扩展属性】功能页，在【文本内容输入框】中输入文字【液位高!!】
- ③ 勾选【可见度】前的复选框，点选出现的【可见度】功能页。设置表达式为【液位值】>=9，选择【当表达式非零时】【构件可见】。



3.2-7 可见度功能设置流程图

● 运行展示

保存工程后，下载工程，然后启动运行。因为【液位值】的默认初始值是 0，所以画面刚开始运行时画面中没有标签构件。将 3.2.2 中输入框的值设置为 10，则标签构件在画面中出现并闪烁。



3.2-8 可见度功能运行效果

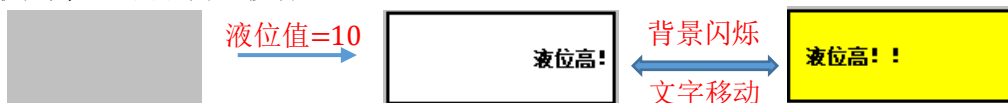
3.2.4 跑马灯

- 为了使提示内容更醒目，可以给文字设置跑马灯效果，使提示文字滚动显示。
双击 3.2.3 中的标签构件，弹出【标签动画组态属性设置】界面。勾选【扩展属性】功能页中【跑马灯】前的复选框，设置【滚动方向】为【向左】，【滚动步进】为【10】。点击确认保存设置并关闭设置界面。



3.2-9 跑马灯功能设置流程图

- 运行展示
保存工程后，下载工程，然后启动运行。因为【液位值】的默认初始值是 0，所以画面刚开始运行时画面中没有标签构件。将 3.2.2 中输入框的值设置为 10，则标签构件在画面中出现并闪烁，且标签构件中的文字【液位高!!】由右向左移动。



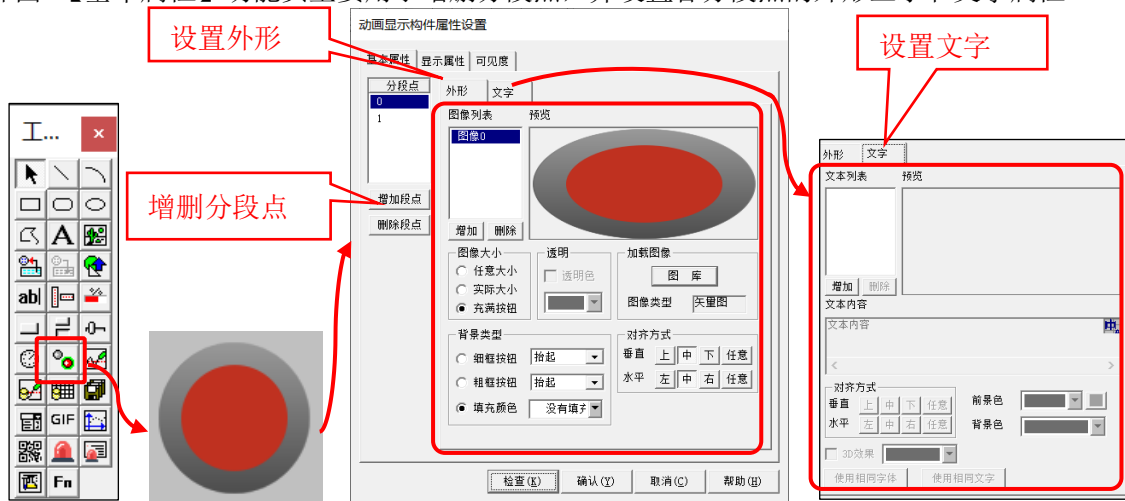
3.2-10 跑马灯功能运行效果

3.3 多状态显示

3.3.1 动画显示

下面以指示灯为例，讲解多状态显示，使用到的构件是【动画显示】构件。

- 将动画显示构件拖拽到画面合适的位置和合适的大小。双击该构件，打开【动画显示构件属性设置】界面。【基本属性】功能页主要用于增删分段点，并设置各分段点的外形显示和文字属性。



3.3-1 动画显示构件基本属性功能页

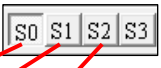
- ⑦ 设置完成后，在组态状态下，通过工具栏上的【分段点选择】图标可以预览每个分段点的状态。



图 3.3-5 分段点组态预览效果

- ⑧ 在实时数据库中添加整数型变量【设备显示】。
⑨ 关联【动画显示】构件 → 【显示属性】 → 【显示变量】为变量【设备显示】，类型为【数值显示】。

显示属性的配置信息

显示属性用于控制动画显示的切换，显示属性页见图 3.3-6。

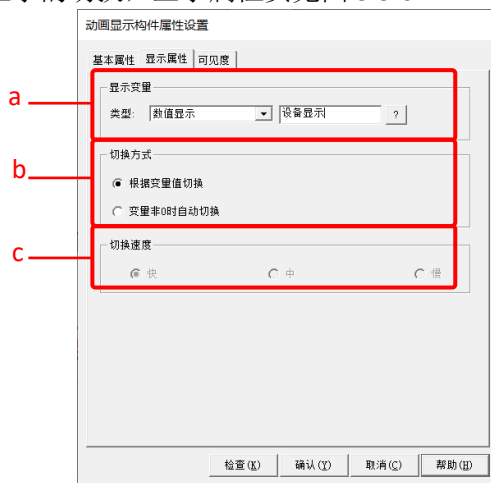


图 3.3-6 动画显示显示属性界面

- a. 显示变量：动画显示构件通过关联显示变量实现图像的切换显示。显示变量的类型包括：浮点数、整数、整数的位。当显示变量为整数的位时，位的范围只能为 0-31，每个位对应的分段点只能为两个。
- b. 切换方式：可用两种不同的方法来实现动画显示效果。一种是根据变量值切换，当表达式的值发生变化时，构件用表达式的值来寻找对应的分段点，如找不到对应的分段点，则构件分段点不会变化。另一种是构件根据变量是否非 0，按设定的频率，自动循环显示各分段点的图像和文字。当变量的值为非 0 时，开始切换显示。当变量的值为 0 时，停止切换显示。
- c. 自动切换显示的速度：当设置切换方式为【变量非 0 时自动切换】时，在此处设定切换图像的频率。
- ⑩ 设置完动画显示构件后，我们还需要组态一个输入框，用它来模拟设置变量【设备显示】的值。
- ⑪ 在画面中插入一个输入框构件，关联它的【操作属性】 → 【对应数据对象名称】为变量【设备显示】。
- ⑫ 保存工程后，下载工程，然后启动运行。在【输入框】构件中，依次输入 0、1、2。可以看到，【动画显示】构件的状态发生相应变化。如图 3.3-7 所示。



图 3.3-7 动画显示运行效果

3.3.2 动画按钮

设备启停控制在实际应用中非常常见，当我们点击按钮后，希望按钮的状态也发生对应的变化。下面就

以电机的启停控制为例为大家讲解按钮的多状态控制及显示，使用到的构件是【动画按钮】。

- ① 在实时数据库中新建一个整数型变量【设备开关】
- ② 【动画按钮】构件的基本属性设置方式同 3.3.1【动画显示】构件的基本属性设置方式。本节样例使用默认设置。如图 3.3-8 至图 3.3-9 所示。

基本属性配置信息

动画按钮的【基本属性】主要用于增减分段点的数量和设置每个分段点对应的外观特征，基本属性页中【分段点 0】的【外形】及【文字】如图 3.3-8 所示。

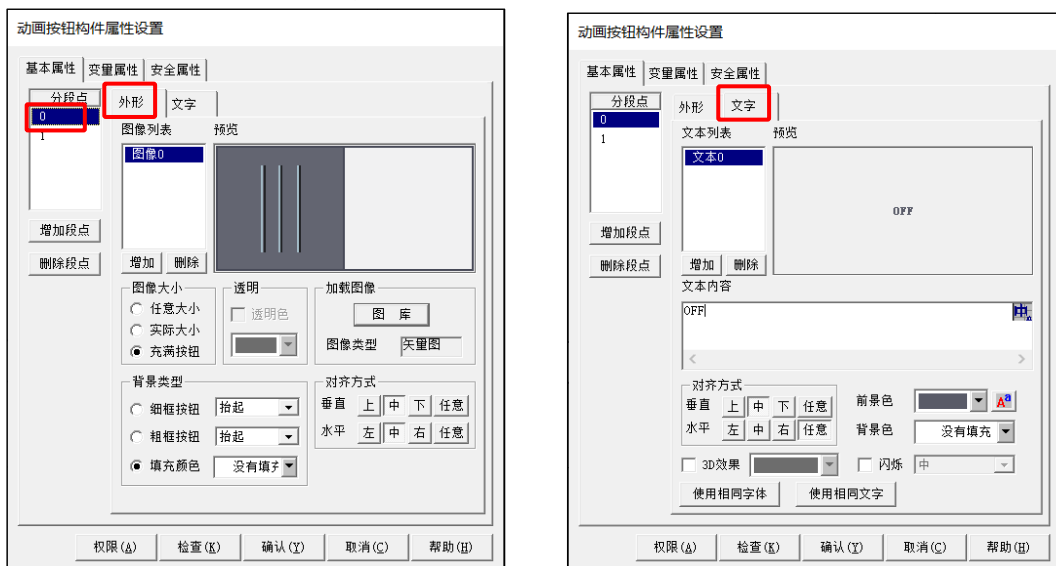


图 3.3-8 分段点 0 的外形及文字设置

基本属性页【分段点 1】的【外形】及【文字】如图 3.3-9 所示。

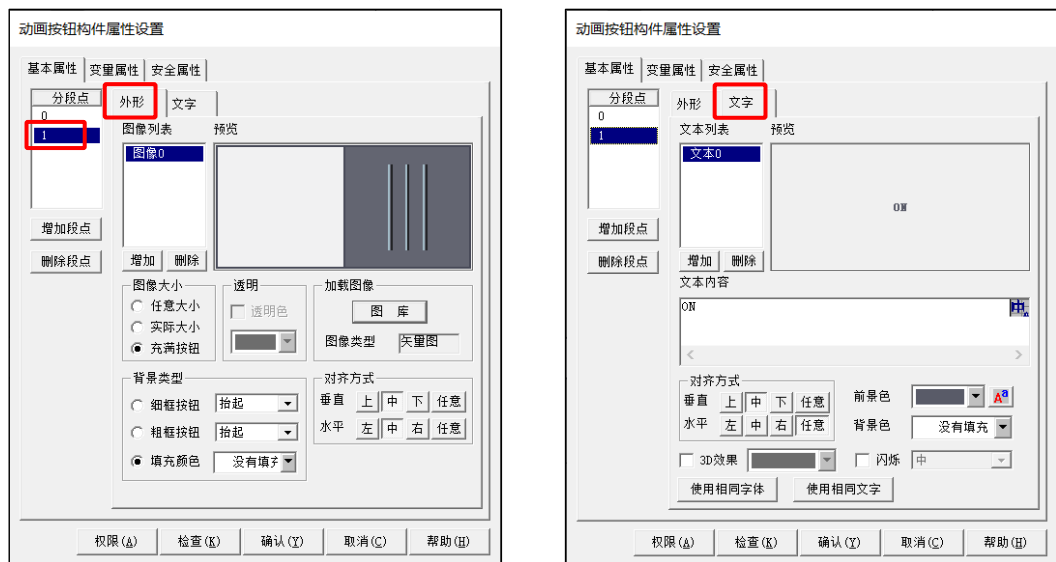


图 3.3-9 分段点 1 的外形及文字设置

- ③ 在【变量属性】功能页中勾选【与设置变量使用同一变量】前的复选框
- ④ 选择【设置变量】的【类型】为【数值操作】
- ⑤ 关联【设置变量】为变量【设备开关】
- ⑥ 选择【执行操作】为【切换分段点】

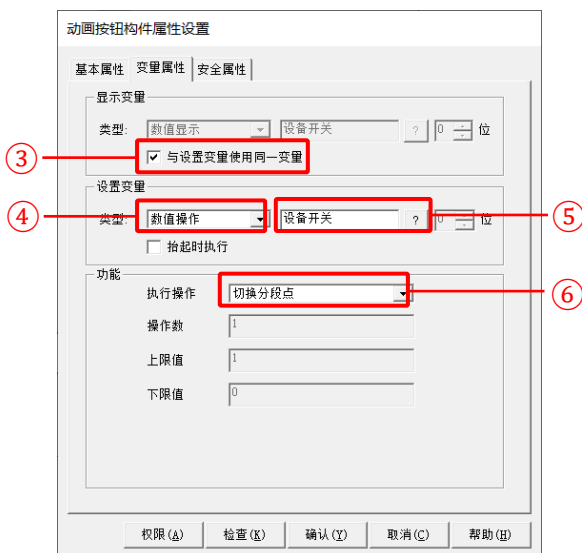


图 3.3-10 动画按钮变量属性界面

变量属性配置信息

动画按钮构件可以关联两个变量：**【显示变量】**和**【设置变量】**。显示变量和设置变量间没有必然的联系。如果只有显示变量，构件没有按钮动作；如果只有设置变量，构件只有按钮动作，没有变量显示的功能；如果设置变量和显示变量关联同一个变量，构件执行按钮动作的同时改变自身的显示状态。变量属性页如图 3.3-11 所示。

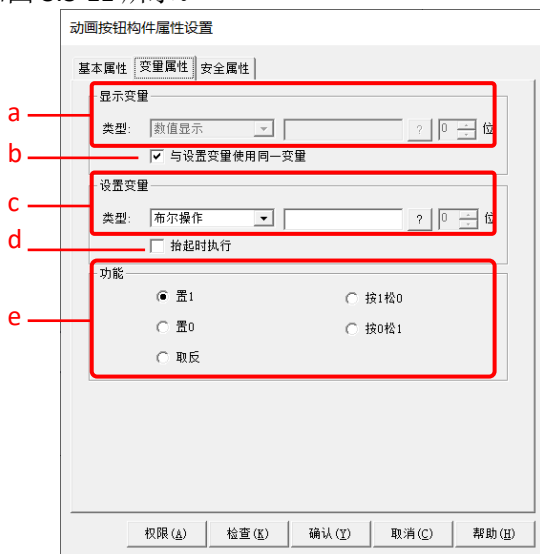


图 3.3-11 动画按钮变量属性界面

- 显示变量：**用于显示构件的段点状态，通过显示变量值的变化使构件可以在多个段点之间进行切换，显示变量类型为：整数、浮点数、整数的位。当显示整数的位时，位的范围为 **0-31**，每一位的分段点只能为两个。
- 与设置变量使用同一变量：**默认设置变量与显示变量为同一个变量，如果需要关联不同变量，需要取消此设置。
- 设置变量：**用于构件执行按钮动作。每按一次动画按钮构件就执行一次操作，设置变量的操作类型为整数、浮点数、整数的位。选择布尔操作和数值操作时，变量类型为整数、浮点数。选择位操作时，变量只能是整数，位的范围为 **0-31**。
- 抬起时执行：**选中后，表示按钮在鼠标抬起时触发变量事件，默认不勾选，表示变量事件在按钮按下时执行，按 **1 松 0**、按 **0 松 1**、递加、递减、循环加、循环减、周期加、周期减操作无此选项。
- 布尔操作的功能：**布尔操作包括置 **1**、置 **0**、取反、按 **1 松 0**、按 **0 松 1**，执行布尔操作后变量值只能是 **1** 或者是 **0**，其中按 **1 松 0** 和按 **0 松 1** 不支持弹框确认安全控制。

- f. 位操作的功能：包括置 1、置 0、取反、按 1 松 0 和按 0 松 1。此操作只针对整数数据的位操作，其中按 1 松 0 和按 0 松 1 不支持弹框确认安全控制。如图 3.3-12。



图 3.3-12 动画按钮位操作属性

- g. 数值操作的功能：数值操作包括切换分段点、设置常量、加、减、递加、递减、循环加、循环减、周期加、周期减功能，其中递加、递减、循环加、循环减、周期加、周期减支持长按操作，但不支持弹窗确认操作。如图 3.3-13。



图 3.3-13 动画按钮数值操作属性

若在【安全属性】功能页中设置了长按安全确认或弹框确认后，以下几个操作需要在长按安全确认或弹框确认成功后，才执行。

切换分段点：动画按钮执行切换到下一个分段点状态。

设置常量：将设置变量的值变为操作数的值。

加：将设置变量的值加上一个操作数，当结果大于等于上限值时，变量值变为上限值。

减：将设置变量的值减去一个操作数，当结果小于等于下限值时，变量值变为下限值。

迟滞时间：仅当操作为递加、递减、循环加、循环减、周期加、周期减时可用，按钮经过长按安全确认后，再经过此时间才被认为是长按操作。

执行间隔：长按操作时，每次操作进行的间隔时间。

递加：经过迟滞时间后长按，每隔一个执行间隔，设置变量会加上一个操作数，当结果大于等于上限值时，变量值变为上限值。

递减：经过迟滞时间后长按，每隔一个执行间隔，设置变量会减去一个操作数，当结果小于等于下限值时，变量值变为下限值。

循环加：经过迟滞时间后长按，每隔一个执行间隔，设置变量会加上一个操作数，当结果大于上限值时，变量值变为上限值，下一次操作变成下限值，再下一次操作继续加上操作数。

循环减：经过迟滞时间后长按，每隔一个执行间隔，设置变量会减去一个操作数，当结果小于下限值时，变量值变为下限值，下一次操作变成上限值，再下一次操作继续减去操作数。

周期加：经过迟滞时间后长按，每隔一个执行间隔，设置变量加上一个操作数，当结果大于上限值时，重置变量为上限值，之后操作变为减去操作数，当结果小于下限值时，重置变量为下限值，之后操作变为加上操作数。

周期减：经过迟滞时间后长按，每隔一个执行间隔，设置变量减去一个操作数，当结果小于下限值时，重置变量为下限值，之后操作变为加上操作数，当结果大于上限值时，重置变量为上限值，之后操作变为减去操作数。

- ⑦ 保存工程，下载工程后，点击启动运行。用鼠标单击动画按钮构件，构件会如图 3.3-14 所示，随动作在两个状态下来回切换。



图 3.3-14 动画按钮运行效果

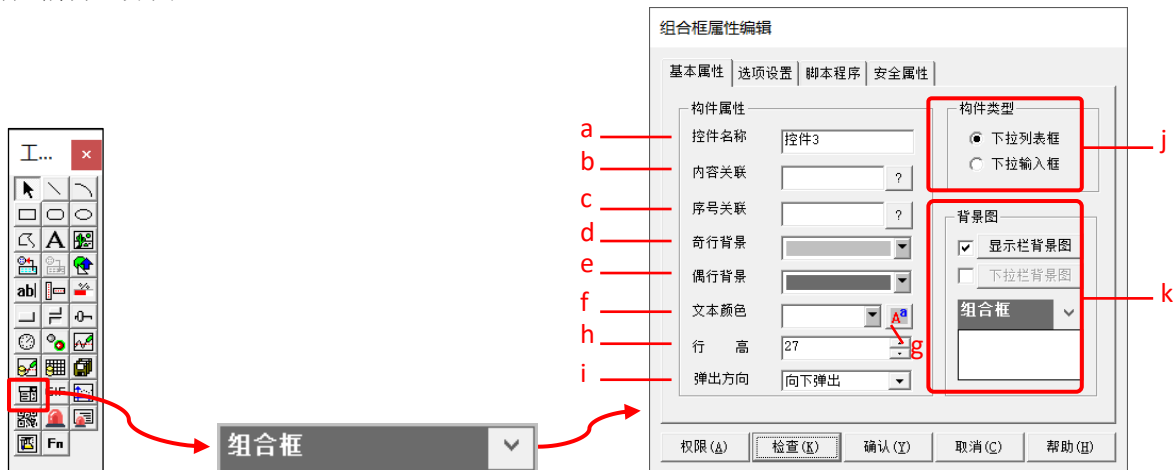
3.4 下拉式菜单选择功能

组合框构件可以实现以下拉列表的方式进行数据选择，避免重复性输入。McgsPro 的组合框构件包括了 2 种类型，本节以第二种为例进行讲解：

- A. 下拉输入框：既可以选择下拉列表中的项，又可以直接输入列表项；
- B. 下拉列表框：只能选择下拉列表中的项。

本节以传送带高中低速调节为例，介绍下拉式菜单选择功能，用到的构件是【组合框】。

- ① 在实时数据库中新建一个字符串型变量【速度内容】，一个整型变量【速度序号】。
- ② 从【工具箱】中拖拽【组合框】构件到画面中合适的位置和合适的大小。双击该构件，弹出【组合框属性编辑】界面。



3.4-1 组合框基本属性页面

基本属性配置信息

- a. 控件名称：设置组合框构件的名称；
- b. 内容关联：设置输出到实时数据库变量的名称，可设置整数、浮点数、字符串变量；
- c. 序号关联：设置选项 ID 号关联的实时数据库变量。选择不同下拉选项后，关联变量值相应改变；或者关联变量值改变后，选项随之改变；
- d. 奇行背景：设置组合框构件编辑显示部分背景颜色及下拉列表奇行颜色；
- e. 偶行背景：设置组合框下拉列表偶行背景颜色；
- f. 文本颜色：设置组合框构件编辑显示部分文字颜色；
- g. 文本字体：设置组合框构件编辑显示部分和下拉菜单部分文字字体；
- h. 行高：设置显示栏和下拉列表行高；

- i. 弹出方向：指定运行时下拉列表弹出方向。
 - j. 构件类型：McgsPro 系列产品仅支持下拉列表框和下拉输入框类型。下拉输入框的编辑显示区域可以点击输入，下拉列表框则不能输入，只能通过下拉选项选择。
 - k. 背景图：显示栏和下拉栏可关联背景图。
- ③ 在基本属性功能页中的内容关联处关联变量【速度内容】，序号关联处关联变量【速度序号】。如图 3.4-2 所示。
 - ④ 然后在选项设置功能页选择【静态选项】，删除表中多余的项目行。将序号【00】的项目内容设置为【低速】，序号【01】的项目内容设置为【中速】，序号【02】的项目内容设置为【高速】。如图 3.4-3 所示。点击确定保存设置并关闭界面。

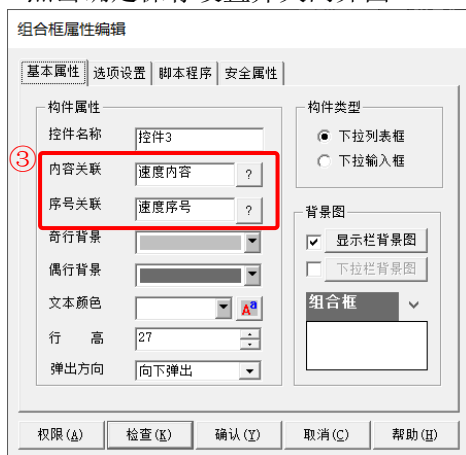


图 3.4-2 组合框基本属性界面



图 3.4-3 组合框选项设置界面

选项设置配置信息

在选项设置栏中每一行配置一个下拉选项，并对应一个序号。选项设置支持多语言，下拉项最多配置 8000 项。

- a. 静态选项：设置下拉列表选项为固定内容；
- b. 动态选项：设置下拉列表选项从关联的字符串变量中读取，按“,”“\”“\r”“\n”或“\r\n”进行分割。

脚本程序配置信息

脚本程序可以设置组合框构件选项发生变化后，执行组态的脚本程序，如图 3.4-4 所示。

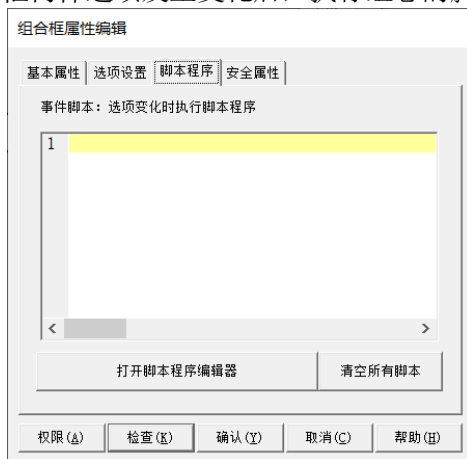


图 3.4-4 组合框脚本程序界面

- ⑤ 拖拽两个输入框构件到窗口中，其中一个的【操作属性】功能页中的【对应数据对象的名称】关联变量【速度内容】，另一个输入框关联变量【速度序号】。
- ⑥ 保存工程后，下载工程，然后启动运行。
- ⑦ 点击【组合框】，在下拉菜单中选择【中速】。可以看到，【组合框】的内容由【低速】变成了【中速】，

而右侧【输入框】中的内容也发生了对应的变化。直接在【输入框】中输入对应的内容，【组合框】中的内容也会发生对应变化。如图 3.4-5 所示。

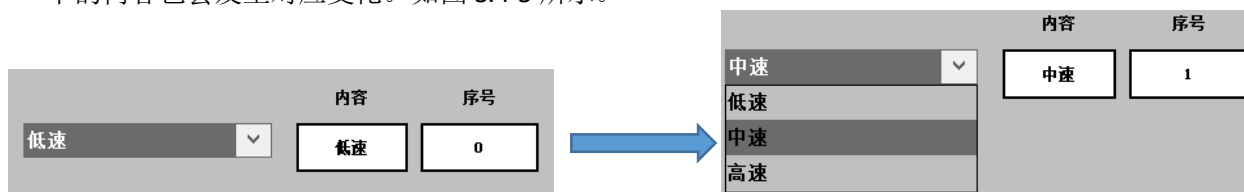


图 3.4-5 组合框运行效果

3.5 构件或窗口的属性和方法

3.5.1 功能概述

构件或用户窗口的属性和方法功能，可以在工程的运行过程中方便灵活的改变构件或用户窗口的属性和状态。如图 3.5-1 所示，在脚本程序中，使用操作符“.”，可以在脚本程序或使用表达式的地方，调用构件对象相应的属性。例如：窗口 0.控件 10.Left 可以获得构件 10 的 X 坐标。

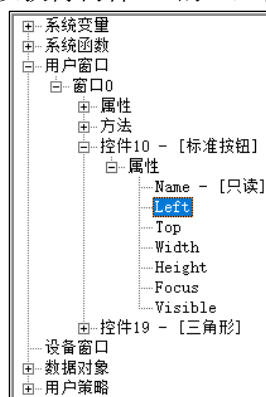


图 3.5-1 用户构件属性

3.5.2 构件属性

每一个动画构件都有以下基本属性，通过这些基本属性，可以对构件有一个基本的描述和设置。

- Name: 构件名字
- Left: 构件的 X 坐标
- Top: 构件的 Y 坐标
- Width: 构件的宽度
- Height: 构件的高度
- Focus: 构件获得焦点
- Visible: 构件的可见度
- 有的构件除了具有公共属性外，还具有自己特有的属性。

如【滑动输入块】构件还具有 Value 属性。

Value: 滑动输入器当前值

3.5.3 窗口属性

- Name: 窗口的名字
- Left: 此函数在 McgsPro 中无效，考虑兼容性问题，故保留
- Top: 此函数在 McgsPro 中无效，考虑兼容性问题，故保留
- Width: 此函数在 McgsPro 中无效，考虑兼容性问题，故保留
- Height: 此函数在 McgsPro 中无效，考虑兼容性问题，故保留
- Visible: 窗口的可见度

实例：用户窗口.窗口 1.Visible = 1，设置“窗口 1”的 Visible 属性值为 1，即为可见
ret = 用户窗口.窗口 1.Visible, ret = “窗口 1”的 Visible 属性值

注意事项： 在子窗口中无效

- Caption: 此函数在 McgsPro 中无效，考虑兼容性问题，故保留

3.5.4 构件方法

构件方法相当于针对构件的脚本操作函数。构件不具有公共方法，本节以【GIF】构件为例，讲解构件方法的使用。

【GIF】构件具有以下两个方法：

Start(): 开始播放

Stop(): 停止播放

我们使用 3.2.1 中用 GIF 加载风扇的样例工程。

- ① 组态风扇开始转动按钮：拖拽一个【标准按钮】构件到画面中合适的位置和合适的大小。双击该构件，弹出【标准按钮构件属性设置】界面。将它【基本属性】中的【文本】输入框中的内容修改为【风扇启动】。然后点选【脚本程序】功能页，点击下方的【打开脚本程序编辑器】按钮，弹出【脚本程序】编辑窗口。依次打开右侧对象树中的【用户窗口】→【窗口 0】→【控件 0-[GIF[3.31]]】-【方法】，双击子目录中的【Start】，则左侧的脚本编辑框中出现一行脚本【窗口 0.控件 0.Start】。保存脚本后关闭【脚本程序】编辑窗口，点击确认保存修改。



图 3.5-2 构件方法设置流程图

- ② 组态风扇停止转动按钮：用同样的方法组态一个【标准按钮】构件，使它的显示文本为【风扇停止】，抬起脚本为【窗口 0.控件 0.Stop()】。

- ③ 保存工程后，下载工程，然后启动运行。可以看到，由于风扇的 GIF 动画选择了【默认循环播放】，画面初始运行时，风扇处于转动状态。而点击【风扇停止】按钮，则风扇停止转动；点击【风扇启动】按钮，风扇又开始转动。



图 3.5-3 构件方法功能运行效果图

3.5.5 窗口方法

窗口方法的设置与构件方法相同。使用窗口方法时需注意：有的方法函数需要设置参数。

- Open(): 打开窗口

实例： 用户窗口.窗口 1.Open(), 打开窗口名为“窗口 1”的窗口

- Close(): 关闭窗口
- Hide(): 隐藏窗口
- Print(): 打印当前窗口

注意事项: 此函数功能暂未实现

- Refresh(): 刷新当前窗口
- BringToTop(): 此函数在 McgsPro 系列产品中无效, 考虑兼容性问题, 故保留
- OpenSubWnd(参数 1, 参数 2, 参数 3, 参数 4, 参数 5, 参数 6): 显示子窗口

参 数 值:

参数 1, 用户窗口名

参数 2, 整数, 打开子窗口相对于本窗口的 X 坐标 iLeft

参数 3, 整数, 打开子窗口相对于本窗口的 Y 坐标 iTop

参数 4, 整数, 打开子窗口的宽度 iWidth

参数 5, 整数, 打开子窗口的高度 iHeight

参数 6, 整数, 打开子窗口的类型

0 位: 是否模态模式打开, 使用此功能, 须通过调用 CloseSubWnd 或 CloseAllSubWnd 来关闭此子窗口, 子窗口外的构件对鼠标操作不响应

1 位: 是否菜单模式打开, 使用此功能, 一旦在子窗口之外鼠标按下, 则子窗口关闭

2 位: 是否显示水平滚动条, 使用此功能, 可以显示水平滚动条 (不支持)

3 位: 是否垂直显示滚动条, 使用此功能, 可以显示垂直滚动条 (不支持)

4 位: 是否显示边框, 选择此功能, 在子窗口周围显示细黑线边框

5 位: 是否自动跟踪显示子窗口, 选择此功能, 在当前鼠标位置上显示子窗口。选用此功能则忽略 iLeft, iTop 的值。如果鼠标在系统窗口内部且当前鼠标位置下无法完整的显示子窗口, 则子窗口会自动调整位置使其显示完整。同理当鼠标在系统窗口外部时, 子窗口也会自动调整位置显示在窗口内部并显示完整。

6 位: 是否自动调整子窗口的宽度和高度为缺省值, 使用此功能则忽略 iWidth 和 iHeight 的值

实例:

(1) !OpenSubWnd(窗口 1,0,0,400,240,1)在位置 (0,0) 打开大小为 400*240, 子窗口名为“窗口 1”的模态子窗口

(2) !OpenSubWnd(窗口 1,0,0,400,240,2)在位置 (0,0) 打开大小为 400*240, 子窗口名为“窗口 1”的菜单子窗口

(3) !OpenSubWnd(窗口 1,0,0,400,240,34)在位置 (0,0) 打开大小为 400*240, 子窗口名为“窗口 1”的菜单子窗口, 并自动跟随鼠标显示

- CloseSubWnd(参数 1): 关闭子窗口

参 数 值: 参数 1, 子窗口的名字

实 例: 用户窗口.窗口 1.CloseSubWnd(窗口 2), 关闭窗口名为“窗口 2”的子窗口

- CloseAllSubWnd(): 关闭当前标准窗口中的所有子窗口

实 例: 用户窗口.窗口 1.CloseAllSubWnd(), 关闭当前标准窗口的所有子窗口

3.6 构件或窗口的事件

3.6.1 功能概述

事件就是当用户在对构件或在窗口中进行某些操作时, 该构件或用户窗口会根据用户不同的操作进行相应的处理。例如当用户在窗口中用鼠标单击窗口时, 就会触发用户窗口的 Click 事件, 同时执行在 Click 事件中定义的一系列操作。

事件是所有构件都具备的公共特性。

3.6.2 构件事件

在对动画构件进行右键操作时, 在弹出的右键菜单中点击【事件】选项, 打开事件组态窗口, 如图 3.6-1 所示。

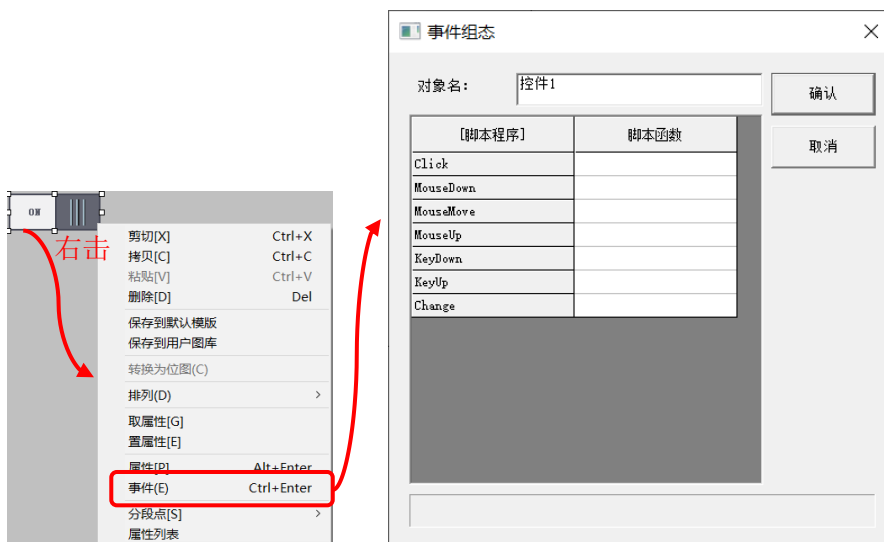


图 3.6-1 事件组态界面

双击某一个事件，即可进入相应的组态画面，对该事件进行设置。

- Click-鼠标单击，见图 3.6-2，点击【事件连接脚本】按钮，将出现【脚本程序】编辑器，可以编辑鼠标单击所要连接的脚本。

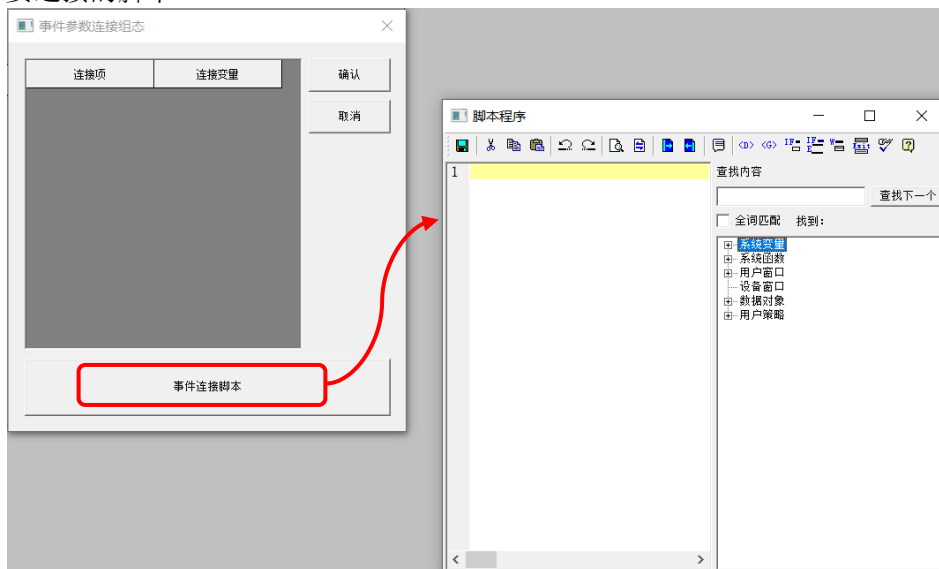


图 3.6-2 Click 事件参数连接组态对话框

- MouseDown-鼠标按下，见图 3.6-3。鼠标按下事件具有四个参数：

参 数 1: 鼠标按下时的鼠标按键信息

= 1 时，表示左键按下

= 2 时，表示右键按下

= 4 时，表示中键按下

参 数 2: 鼠标按下时的键盘信息

= 1 时，表示 Shift 键按下

= 2 时，表示 Ctrl 键按下

= 4 时，表示 Alt 键按下

参 数 3: 鼠标按下时的 X 坐标

参 数 4: 鼠标按下时的 Y 坐标



图 3.6-3 MouseDown 事件参数连接组态对话框



3.6-4 MouseMove 事件参数连接组态对话框

● MouseMove-鼠标移动，见图 3.6-4。鼠标移动事件具有四个参数：

参 数 1： 鼠标按下时的鼠标按键信息

= 1 时，表示左键按下

= 2 时，表示右键按下

= 4 时，表示中键按下

参 数 2： 鼠标按下时的键盘信息

= 1 时，表示 Shift 键按下

= 2 时，表示 Ctrl 键按下

= 4 时,表示 Alt 键按下

参 数 3： 鼠标的 X 坐标

参 数 4： 鼠标的 Y 坐标

● MouseUp-鼠标抬起，见图 3.6-5。鼠标抬起事件具有四个参数：

参 数 1： 鼠标按下时的鼠标按键信息

= 1 时，表示左键按下

= 2 时，表示右键按下

= 4 时，表示中键按下

参 数 2： 鼠标按下时的键盘信息

= 1 时,表示 Shift 键按下

= 2 时,表示 Ctrl 键按下

= 4 时,表示 Alt 键按下

参 数 3： 鼠标抬起时的 X 坐标

参 数 4： 鼠标抬起时的 Y 坐标



图 3.6-5 MouseUp 事件参数连接组态对话框



3.6-6 KeyDown 事件参数连接组态对话框

- **KeyDown**-按下按键，见图 3.6-6。按键按下事件具有两个参数：
参 数 1：整数，按下按键的 AscII 码
参 数 2：整数，0~7 位是按键的扫描码，目前按键的扫描码恒为 0。
- **KeyUp**-按键抬起，见图 3.6-7。按键抬起事件具有两个参数：
参 数 1：整数，按下按键的 AscII 码
参 数 2：整数，0~7 位是按键的扫描码，目前按键的扫描码恒为 0。



图 3.6-7 KeyUp 事件参数连接组态对话框



图 3.6-8 Change 事件参数连接组态对话框

- 有的构件除了具有公共事件外，还具有自己特有的事件。
如【输入框】构件还具有【Change】事件。见图 3.6-8。

Change

事件意义：当输入框内容改变时触发。

注意事项：只有通过键盘输入操作改变输入框内容才会触发

3.6.3 窗口事件

除了构件，McgsPro 的用户窗口也包括事件：



图 3.6-9 窗口事件对话框

窗口事件中的【Click】【MouseDown】【MouseMove】【MouseUp】【KeyDown】【KeyUp】事件请参考 3.6.2 构件事件。

下面，进行窗口特有事件的介绍。

- **Load**：窗口装载时触发。
- **Unload**：窗口关闭时触发。
- **Resize**：无实际意义，预留功能。

3.6.4 样例演示

组态工程实现：当鼠标移动到主界面按钮上时，按钮左侧的箭头高亮显示；当鼠标从主界面按钮上移开后，按钮左侧的箭头变暗。

- ① 在实时数据库中新建一个整数型变量【主界面高亮】
- ② 在画面中用【常用符号】中的【等腰三角形】画一个细长的等腰三角形当作箭头。然后选中该等腰三角形，点击工具栏中的【右旋 90 度】按钮将它顺时针旋转 90，如图 3.6-10 所示。

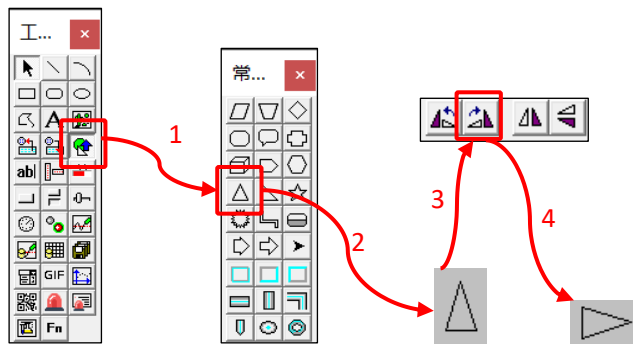


图 3.6-10 三角形组态流程图

- ③ 拖拽一个【标准按钮】构件到箭头的右侧，将按钮【基本属性】中的文本修改为【主界面】。然后用鼠标同时选中按钮和三角形，点击工具栏上的【纵向对中】按钮使它们对齐。

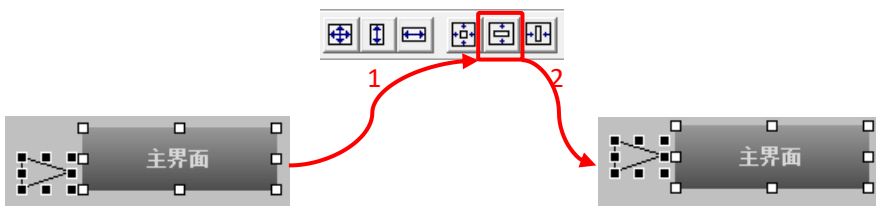


图 3.6-11 对齐按钮和三角形

- ④ 双击三角形，弹出【动画组态属性设置】界面，勾选【填充颜色】前的复选框，点击出现的【填充颜色】功能页。将【表达式】关联变量【主界面高亮】，将下方【分段点 0】的对应颜色设置为【灰色】，【分段点 1】的对应颜色设置为【黄色】。点击确认关闭组态界面。

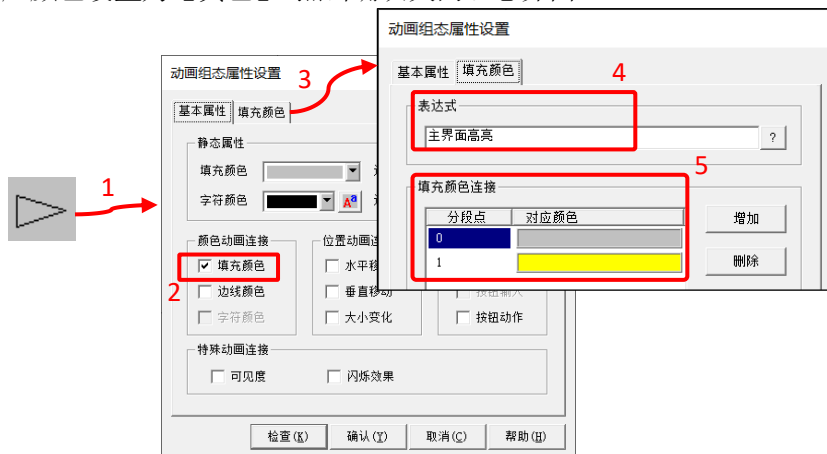


图 3.6-12 组态三角形填充颜色

- ⑤ 组态【按钮】构件【事件】脚本：右击按钮，在弹出的菜单中选择【事件】，弹出【事件组态】窗口。双击【MouseMove】行，弹出【事件参数连接组态】窗口。点击下方的【事件连接脚本】按钮，弹出【脚本程序】编辑窗口。在右侧的对象树中，点击【数据对象】，在子目录中双击出现的【主界面高亮】，则变量【主界面高亮】出现在左侧的脚本编辑框中。在最后一行脚本中的变量【主界面高亮】后，紧跟着输入脚本“=1”。输入完成后，点击保存，并关闭【脚本程序】编辑窗口，然后连着点击两次确认按钮，关闭并保存刚才所做的修改。

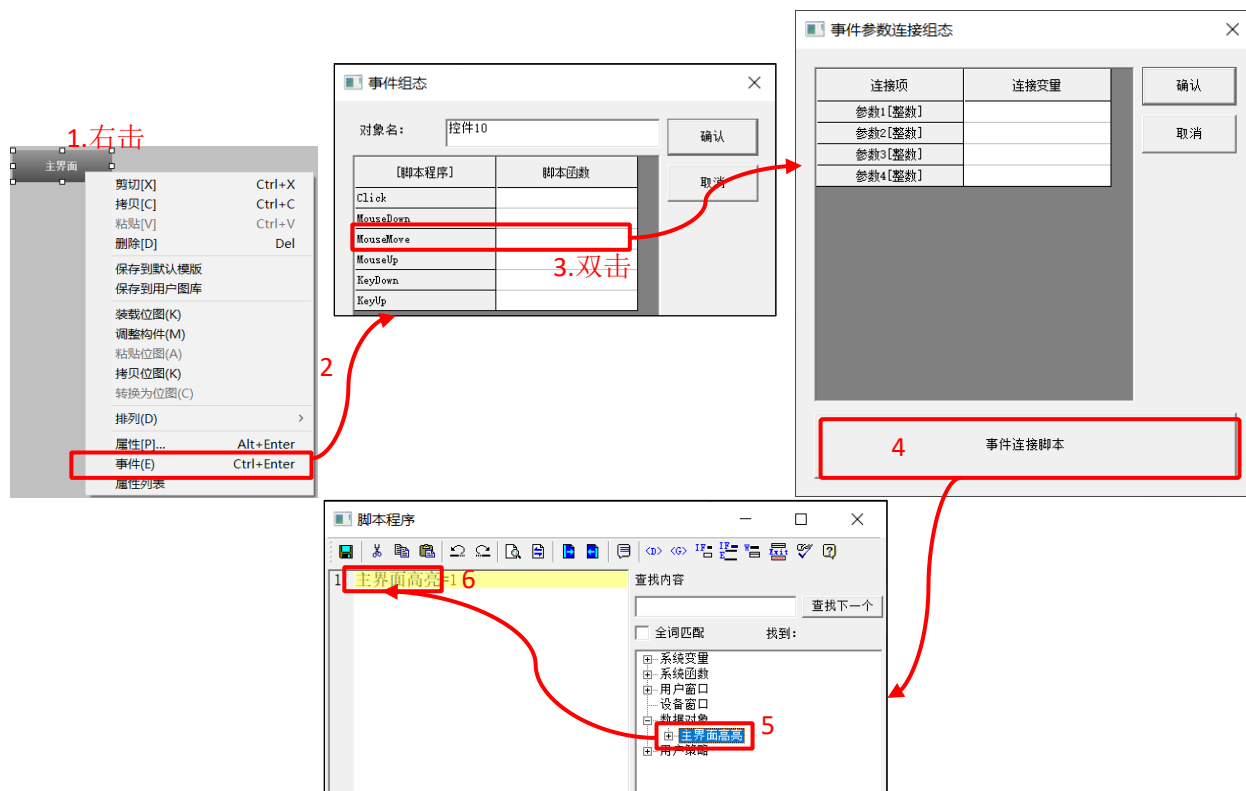


图 3.6-13 组态按钮构件事件脚本流程图

- ⑥ 组态【窗口事件】脚本：在窗口的空白处右击，在弹出的菜单中选择【事件】。用组态【按钮】构件【事件】脚本同样的方法组态【窗口事件】脚本，将脚本写为“主界面高亮=0”。

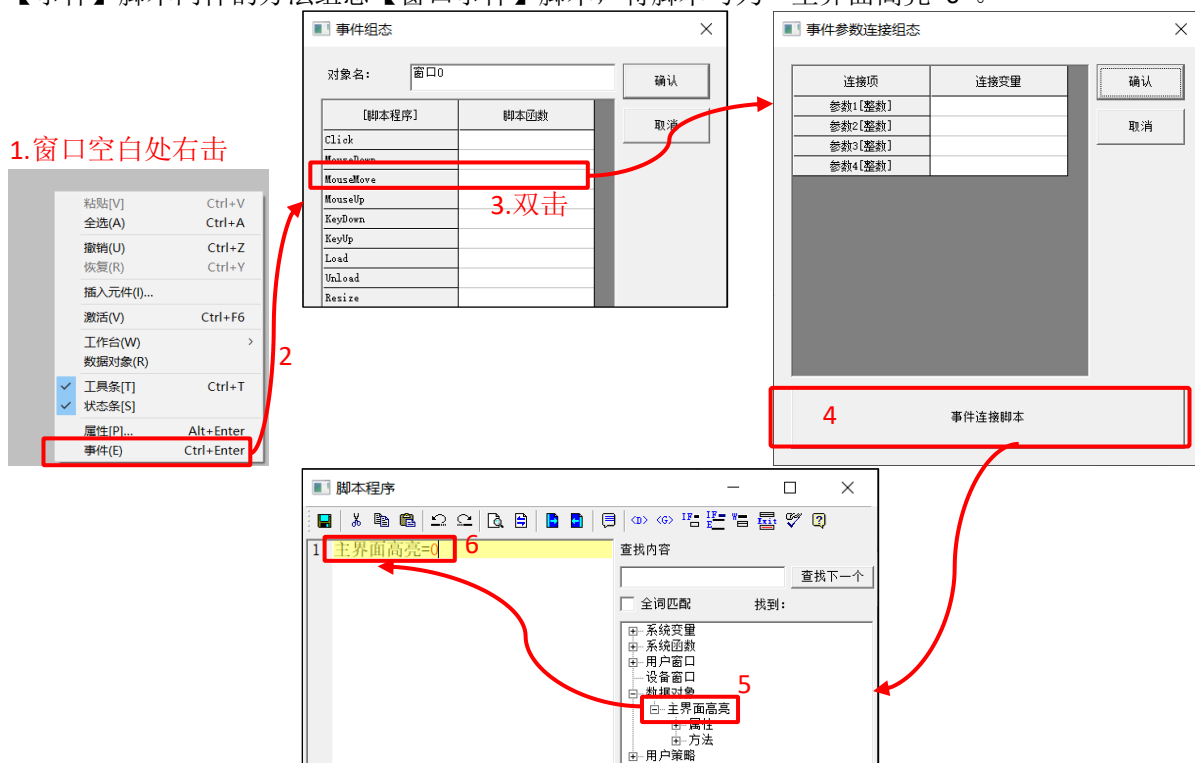


图 3.6-14 组态窗口事件脚本流程图

3.6.5 运行展示

保存工程后，下载工程，然后启动运行。

可以看到，当我们将鼠标移动到窗口空白处时，三角形箭头呈灰色；当我们将鼠标移动到按钮上时，三角形箭头呈黄色；从按钮上上移开鼠标后，三角形又变成了灰色。



图 3.6-15 事件功能运行效果图

第4章 报警功能应用实例

McgsPro 组态软件提供实时报警和历史报警功能，通过组态报警功能，用户可以更好地掌握现场设备运行情况，保证产品生产安全。

4.1 报警功能介绍

首先介绍 McgsPro 组态软件报警实现流程：从 PLC 等外部设备读取的数据传送给实时数据库中对应的数据对象，在用户窗口显示对应数据对象（即变量）的值，也就是显示了当前 PLC 中的值；用户只需要判断实时数据库中数据对象的值是否满足报警的条件，如果满足即产生报警；报警流程如图 4.1-1 所示：

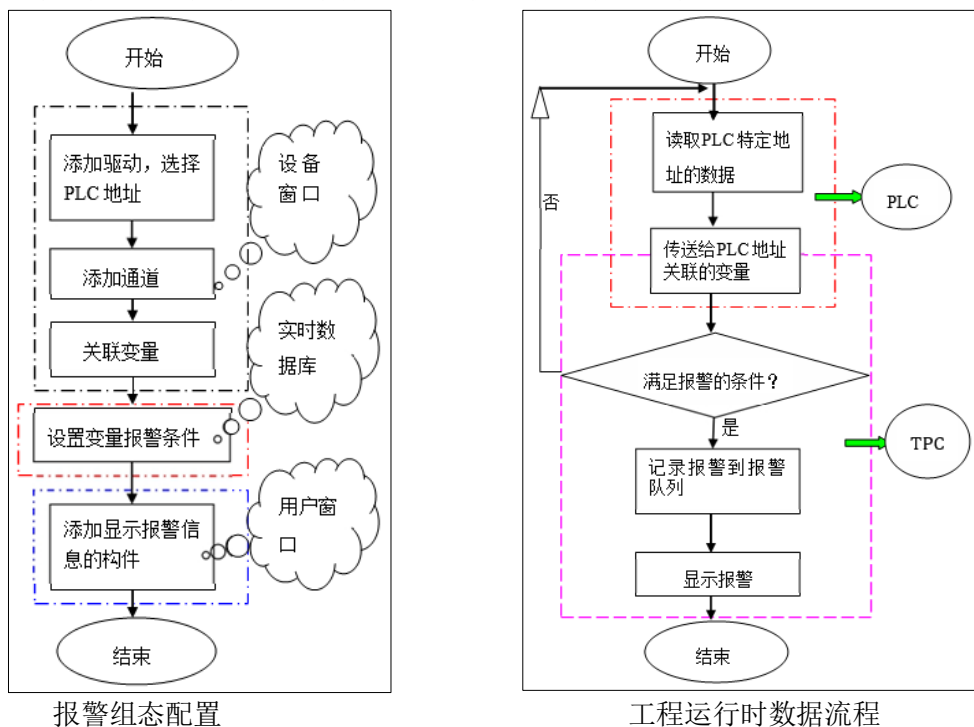


图 4.1-1 报警流程

图 4.1-1 是实现报警功能的组态流程，首先要确定所用的硬件设备，例如 PLC 型号，在 McgsPro 组态软件的设备窗口添加正确的驱动构件，添加 PLC 通道地址，并且在通道中关联变量 → 进入实时数据库中设置报警属性 → 在用户窗口用报警构件显示报警信息。如图 4.1-2 所示，McgsPro 组态软件提供了报警条构件（只支持实时报警），报警浏览构件（支持实时和历史报警），显示报警信息。



图 4.1-2 支持报警显示的构件

4.2 实时报警组态

重启 TPC 后，实时报警信息会被清空，实时报警组态流程如下：

现场用户可能需要实时对下位机数据进行监控，并显示其报警信息，我们通过一个样例来学习报警的各种表现形式。以 S7-200PLC 为例，图 4.1-3 是该样例的运行效果。



图 4.1-3 报警运行效果图

图 4.1-3 显示了常用的四种基本报警形式：

a. 位报警：当 PLC 的地址 M12.3 状态为 1 时提示水满了，此报警信息通过“报警条构件”在屏幕上滚动显示。

b. 上下限报警：当 PLC “V 寄存器” 的字地址 49 的值超过 10~30 的范围，提示温度太高或太低，通过“报警浏览构件”以列表显示。

c. 多状态报警：当 PLC “V 寄存器” 的字地址 200 值非 0 时表示不同的故障，在画面上显示异常报警信息。

各种故障信息如下：

V200 的值	含义
0	正常
1	故障信息 1
2	故障信息 2
3	故障信息 3
4	故障信息 4

d. 报警弹窗：当 “M 寄存器” 的地址 12.3 发生报警后立即弹出一个子窗口，显示当前报警信息。

4.2.1 位报警组态

图 4.1-3 报警形式 “a”：PLC 地址 M12.3 的值为 “1” 时提示 “水满了”，并且滚动显示。

方案：地址 M12.3 报警内容固定，直接设置对应变量的报警属性，然后在用户窗口用报警条构件（走马灯）显示。

- ① 添加位通道：在设备窗口，双击 “西门子_S7200PPI” 驱动进入 “设备编辑窗口”。如图 4.2-1 所示，先通过 “删除设备通道” 按钮，删除默认的通道；再点击 “增加设备通道” 按钮，弹出 “添加设备通道” 对话框，选择通道类型 “M 寄存器”，数据类型 “通道的第 03 位”，通道地址为 “12”，通道个数为 “1”，读写方式选择 “读写”。设置完成点击 “确认”。可以看到设备编辑窗口增加通道成功。

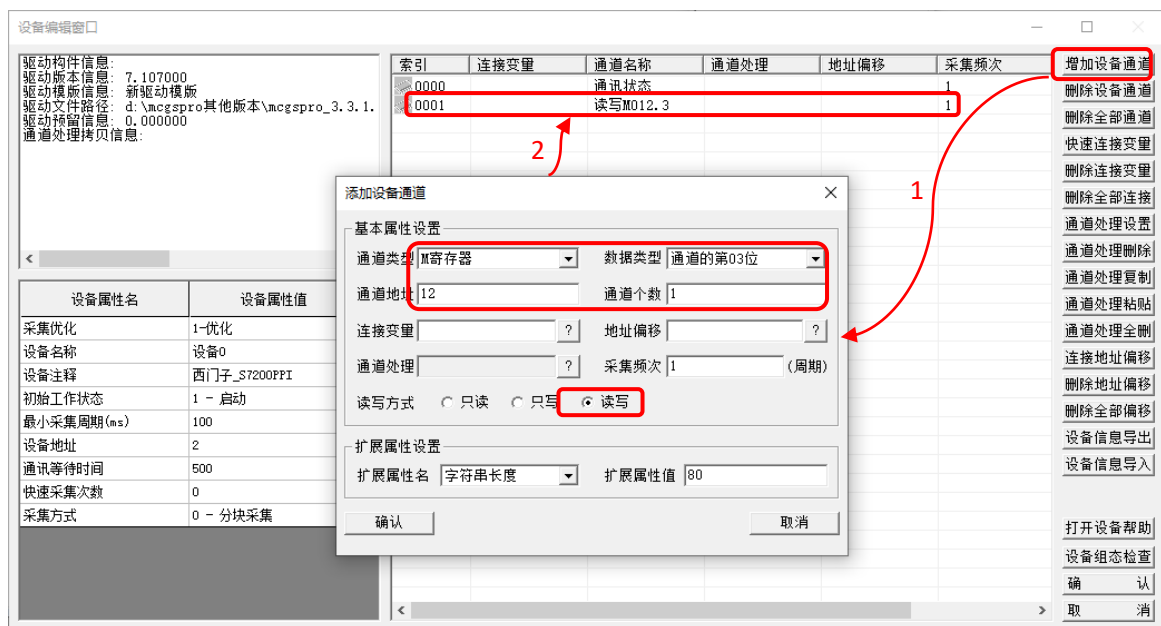


图 4.2-1 增加设备通道 M012.3

- ② 通道关联变量：如图 4.2-2 所示，点击“快速连接变量”→“默认设备变量连接”→“确认”。自动生成变量名“设备 0_读写 M012_3”。在设备编辑窗口点击“确认”，最后在系统弹出“添加数据对象”的提示框，选择“全部添加”，McgsPro 自动在“实时数据库”添加整数变量“设备 0_读写 M012_3”。

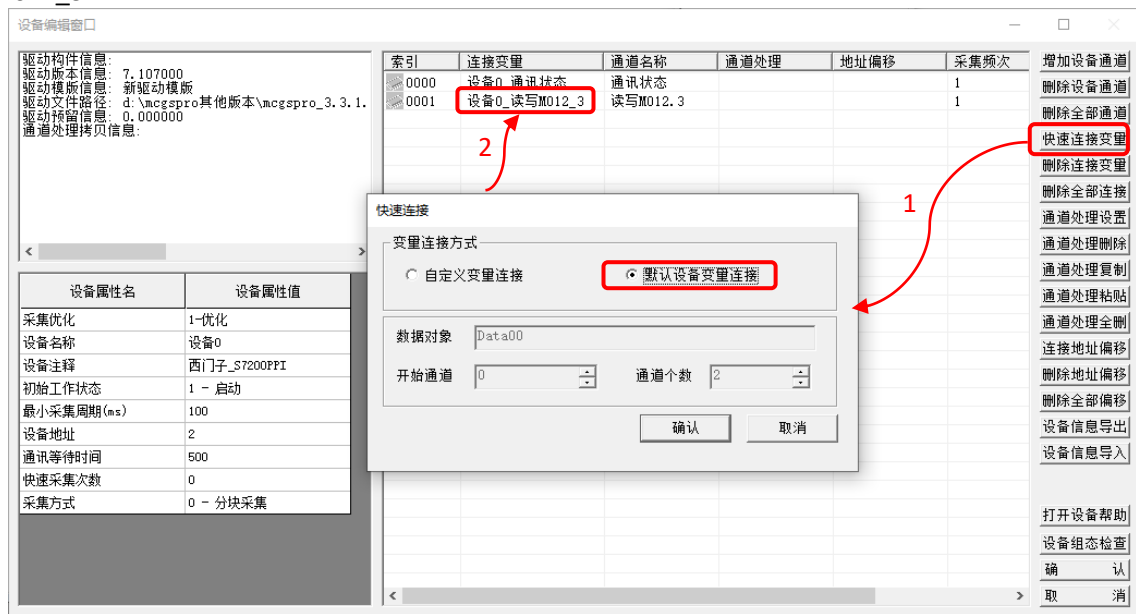
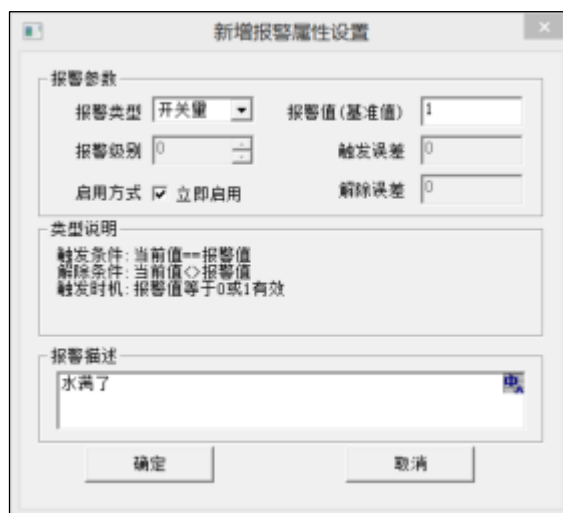


图 4.2-2 连接变量

- ③ 进入实时数据库，如图 4.2-3 所示，双击整数变量“设备 0_读写 M012_3”→“报警属性”→ 表格空白处右键 → 追加，在修改报警属性设置对话框中，报警类型选择“开关量”，报警值“1”，报警描述为“水满了”，设置完成点击“确认”。



进入报警属性设置



设置开关量报警

图 4.2-3 设置开关量报警

注意：只有变量类型为“整数”，才支持选择“开关量”报警类型。

- ④ 设置报警条构件：在用户窗口中新建“窗口 0”，双击进入“窗口 0”组态界面，点击工具箱中的 ，绘制报警条构件，双击绘制的报警条构件 → “报警条属性设置”，“报警对象”不关联对象（“报警对象”位置不关联任何数据对象时，该报警条将显示实时数据库中所有变量产生的实时报警信息），按照需要设置其外观显示、滚动设置如图 4.2-4 所示：“显示格式”可设置报警信息的显示内容）



图 4.2-4 报警条设置

- ⑤ 显示数据：在工具箱点击  添加一个“标签”，双击标签勾选显示输出。在显示输出属性页，点击  选择变量“设备 0_读写 M012_3”，以开关量输出。另外添加一个“标签”，输入“显示注水状态”。参照图 4.2-5 中效果设置标签颜色和字体颜色。
- ⑥ 查看效果：组态完成后，连接 PLC，下载工程运行查看效果：当 PLC 有报警产生时（变量“设备 0_读写 M012_3”的值等于图 4.2-3 设置的报警值），报警信息显示如图 4.2-5 所示：

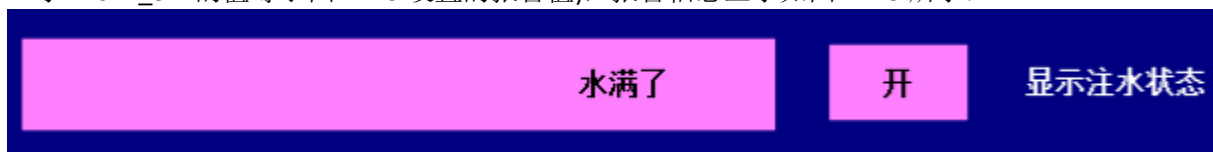


图 4.2-5 报警条显示报警信息

4.2.2 上下限报警

图 4.1-3 报警形式“b”：PLC 中“V 寄存器”地址 49 的值超出 10~30 的范围时，以列表形式报警显示温度太高或太低。

方案：设置“V 寄存器”地址 49 对应变量的报警属性，在用户窗口用报警浏览构件显示。

- ① 添加字通道：在设备窗口，双击“西门子_S7200PPI”驱动进入“设备编辑”窗口。如图 4.2-6 所示，单击“增加设备通道”按钮，进入“添加设备通道”对话框，选择通道类型为“V 寄存器”，数据类型为“16 位无符号二进制”，通道地址为“49”，通道个数为“1”，读写方式为“读写”。设置完成后点击“确认”按钮。

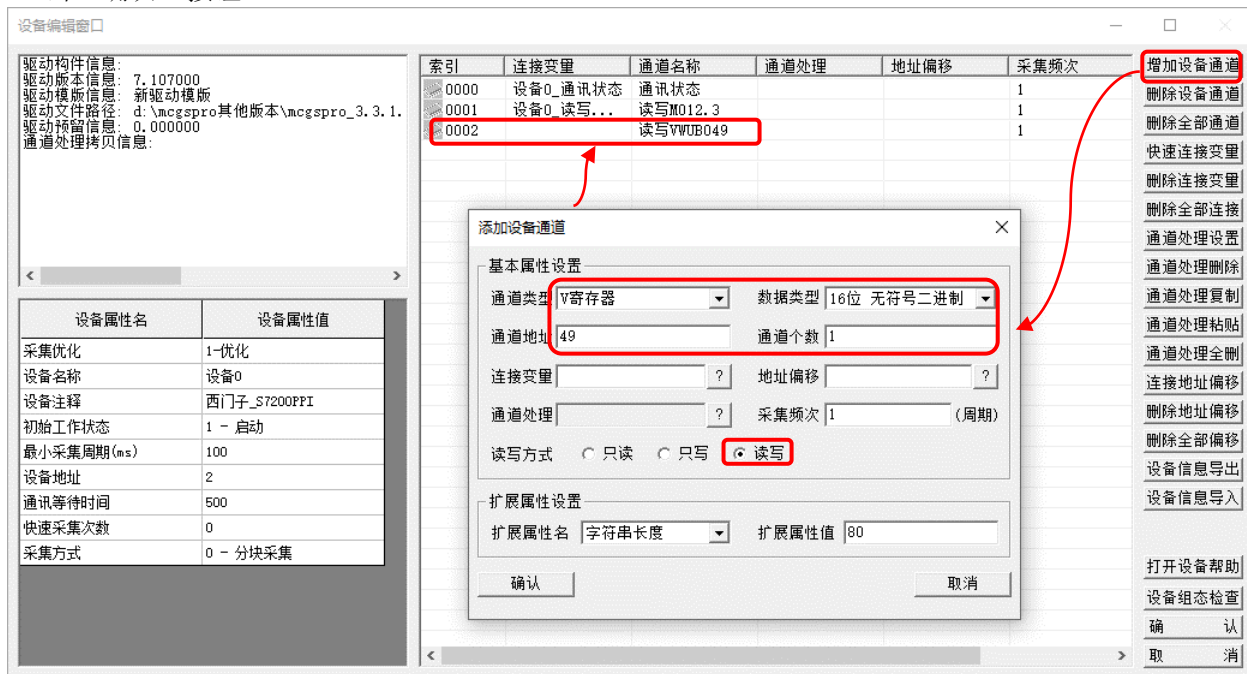


图 4.2-6 增加设备通道：VWUB049

- ② 通道关联变量：如图 4.2-7 所示，点击“快速连接变量”→“默认设备变量连接”→“确认”。自动生成变量名“设备 0_读写 VWUB049”。在设备编辑窗口点击“确认”，最后在系统弹出“添加数据对象”的提示框，选择“全部添加”，McgsPro 自动在“实时数据库”添加浮点数变量“设备 0_读写 VWUB049”。



图 4.2-7 连接变量

- ③ 进入实时数据库，双击浮点数变量“设备_0_读写_VWUB049”→“报警属性”→表格空白处右键→追加，在修改报警属性设置对话框中，报警类型选择“上限”，设置报警值为“30”，报警描述为“温度太高了”，设置完成点击“确认”；按照同样的方式，追加“下限”报警，并设置报警值和报警描述，如图 4.2-8 所示：

新增报警属性设置

报警参数

报警类型: 上限 报警值(基准值): 30

报警级别: 0 触发误差: 0

启用方式: ☒ 立即启用 解除误差: 0

类型说明

触发条件: 当前值<=报警值
解除条件: 当前值>报警值
触发时机: 限值报警在同一时刻只能触发一种
触发顺序: 限值报警触发顺序为: 上上限>上限>下下限>下限

报警描述

温度太高了

确定 取消

设置上限报警

新增报警属性设置

报警参数

报警类型: 下限 报警值(基准值): 10

报警级别: 0 触发误差: 0

启用方式: ☒ 立即启用 解除误差: 0

类型说明

触发条件: 当前值<=报警值
解除条件: 当前值>报警值
触发时机: 限值报警在同一时刻只能触发一种
触发顺序: 限值报警触发顺序为: 上上限>上限>下下限>下限

报警描述

温度太低了

确定 取消

设置下限报警

图 4.2-8 设置上下限报警

注意：只有变量类型为“浮点数”，才支持选择“上限”和“下限”报警类型。

- ④ 设置报警浏览构件：进入“窗口 0”组态画面，点击工具箱中的图标 ，在“窗口 0”中绘制报警浏览构件，再双击绘制的报警浏览构件，进入“报警浏览构件属性设置”对话框。
- ⑤ 如图 4.2-9 所示：在数据来源页，数据类型选择“实时报警数据”，报警对象留空（“报警对象”不关联任何对象时，该报警浏览构件将显示实时数据库中所有变量产生的报警信息）。
- ⑥ 如图 4.2-10 所示：在显示属性页，按照需要勾选显示内容：“日期”、“时间”、“对象名”、“报警类型”、“报警值”、“报警描述”，并设置合适的列宽；其他功能页用户根据自己的需要进行设置。设置完成点击“确认”。

报警浏览构件属性设置

基本属性 数据来源 显示属性 报警输出

数据类型

☒ 实时报警数据(R) ☐ 历史报警数据(H)

报警对象:

排序方式

☒ 新报警在上 ☐ 新报警在下

实时报警

报警应答: 双击应答 报警闪烁: 无

时间选择

☒ 最近一天 ☐ 最近一周 ☐ 最近一月 ☐ 全部

☐ 自定义 时间格式: 2008-08-08 08:08:08

开始时间:

结束时间:

☐ 自动刷新历史数据

检查 确认 取消 帮助

图 4.2-9 数据来源设置

报警浏览构件属性设置

基本属性 数据来源 显示属性 报警输出

显示内容及显示列宽

☐ 编号 40

☒ 日期 90

☒ 时间 70

☒ 对象名 100

☒ 报警类型 70

☐ 报警事件 70

☒ 报警值 70

☐ 界限值 70

☒ 报警描述 100

☐ 响应时间 70

☐ 结束时间 70

表格样式

☒ 网格类型 ☐ 无网格类型

☐ 边框类型 ☐ 无线网格类型

日期格式: yyyy/mm/dd 时间格式: hh:mm:ss

行列显示

行数: 3 行高: 27 起始行: 0

预览(P)

检查 确认 取消 帮助

图 4.2-10 显示属性设置

- ⑦ 显示数据：在工具箱点击  添加一个“标签”，双击标签勾选显示输出。在显示输出属性页，点击

? 选择变量“设备 0_读写 VWUB049”，以数值量输出。另外添加一个“标签”，输入“显示当前温度”。参照图 4.2-11 效果设置标签颜色和字体颜色。

- ⑧ 查看效果：组态完成后，连接 PLC，下载工程运行查看效果：当 PLC 有报警产生时（变量“设备 0_读写 VWUB049”的值超出图 4.2-8 设置的范围），报警信息显示如图 4.2-11 所示：

实时报警					显示当前温度
日期	时间	对象名	报警值	报警描述	
2019/08/06	13:49:12	设备0_读写VWUB049	10	温度太低了	10℃

图 4.2-11 报警浏览显示报警信息

4.2.3 多状态报警

图 4.1-3 报警形式“c”：PLC 中“V 寄存器”地址 200 输出的值不同时，提示不同的故障信息。

方案：采用数值报警，用“值==”的报警类型实现当“V 寄存器”地址 200 输出的值不同时，提示对应的故障信息。

- ① 添加字通道：在设备窗口，双击“西门子_S7200PPI”驱动进入“设备编辑”窗口。如图 4.2-12 所示，单击“增加设备通道”按钮，进入“添加设备通道”对话框，选择通道类型为“V 寄存器”，数据类型为“16 位无符号二进制”，通道地址为“200”，通道个数为“1”，读写方式为“读写”。设置完成后点击“确认”按钮。



图 4.2-12 增加设备通道：VWUB200

- ② 通道关联变量：如图 4.2-13 所示，点击“快速连接变量”→“默认设备变量连接”→“确认”。自动生成变量名“设备 0_读写 VWUB200”。在设备编辑窗口点击“确认”，最后在系统弹出“添加数据对象”的提示框，选择“全部添加”，McgsPro 自动在“实时数据库”添加浮点数变量“设备 0_读写 VWUB200”。



图 4.2-13 连接变量

- ③ 进入实时数据库，如图 4.2-14 所示，双击浮点数变量“设备0_读写VWUB200”→“报警属性”→表格空白处右键→追加，报警类型选择“值==”。按照进阶篇章 4.2.2 的方式追加报警，效果如图 4.2-15 所示：



图 4.2-14 追加报警



图 4.2-15 效果展示

- ④ 设置报警浏览构件和显示数据：参见进阶篇章 4.2.2 的方式设置报警浏览构件和显示数据，组态完成后，连接 PLC，下载运行查看效果，当 PLC 有报警产生时，报警信息显示如图 4.2-16 所示：



图 4.2-16 多状态报警信息显示

4.2.4 报警弹窗

图 4.1-3 报警形式“d”：当“M 寄存器”的地址 12.3 发生报警后立即弹出一个子窗口，显示当前报警信息。

方案：用子窗口弹出来实现，运用报警策略来及时判断报警是否发生，并设置子窗口显示的大小和坐标。

- ① 添加子窗口：在工作台界面切换到用户窗口，新建“窗口 1”。
- ② 设置显示信息：双击进入“窗口 1”组态画面，如图 4.2-17 所示，通过“插入元件”，在对象元件库插入“标志 24”。



图 4.2-17 插入警示标志

- ③ 再添加一个“标签”，设置文本内容为“水满了！”，调整标签构件的背景颜色，然后把这两个构件放到“窗口 1”左上角合适的位置，效果如图 4.2-18 所示：



图 4.2-18 报警窗口信息

- ④ 设置窗口 1 弹出方式：如图 4.2-19 所示，在“窗口 1”组态画面空白处双击，进入“子窗口属性”，勾选“作为子窗口使用”，并设置子窗口弹出位置和大小，设置完成后点击“确认”。

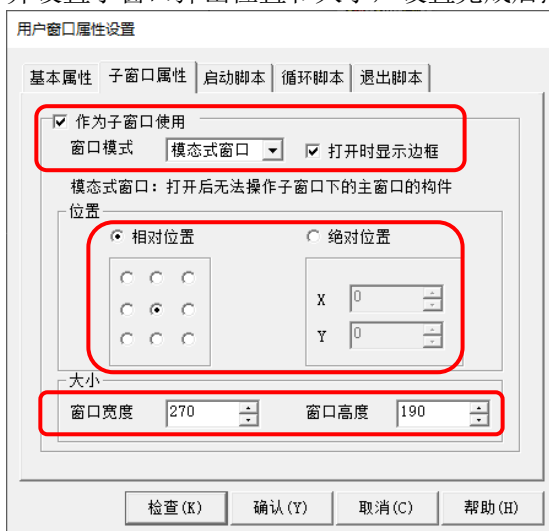


图 4.2-19 子窗口弹出设置

- ⑤ 组态窗口弹出功能：在工作台界面切换到“运行策略”，点击“新建策略”按钮，在“选择策略的类

型”对话框中选择“报警策略”，确定后回到运行策略窗口，双击新建的报警策略进入策略组态窗口，在策略组态窗口空白处点击鼠标右键 → “新增策略行”，如图 4-2-20 所示



图 4.2-20 新增策略行

- ⑥ 设置策略执行方式：双击上图中的  进入“策略属性设置”对话框，数据对象关联“设备 0_读写 M012_3”，执行条件选择“报警产生时，执行一次”，如图 4.2-21 所示：



图 4.2-21 策略执行方式

- ⑦ 添加窗口 1 打开脚本：如图 4.2-22 所示，进入“脚本程序”编辑框 → 右侧对象树 → 用户窗口 → 窗口 1 → 方法 → 双击“open”，将窗口 1 打开脚本添加到左侧的脚本程序编辑框，添加完成后检查并保存。

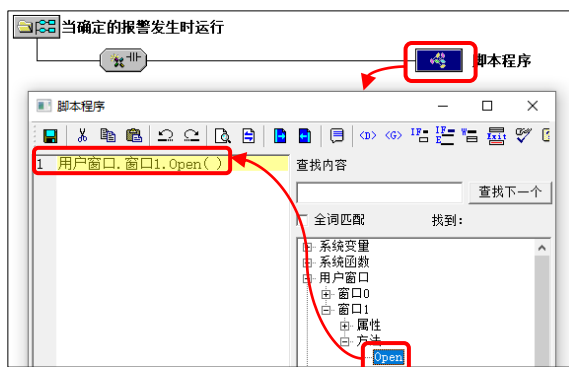


图 4.2-22 添加窗口 1 打开脚本

- ⑧ 采用同样的方式再次新建一个报警策略，数据对象关联“设备 0_读写 M012_3”，执行条件选择“报警结束时，执行一次”，脚本程序为“用户窗口.窗口 1.Close()”。
- ⑨ 效果演示：组态完成后，连接 PLC，当“M 寄存器”的地址 12.3 发生报警时，在就会在当前的用户窗口中，以子窗口打开“窗口 1”；报警结束则会自动关闭“窗口 1”。

4.3 历史报警功能

4.3.1 历史报警组态

现场用户如需要保存报警记录，要求即使重启 TPC，仍然可以进行查看。对于该情况，工程组态人员可以在组态时保存产生的报警信息，并通过报警浏览构件显示历史报警。

保存历史报警信息操作方法如下：

- ① 在“实时数据库”双击需要保持历史报警记录的变量，进入“报警属性”设置界面（参考进阶篇章 4.2.1）。如图 4.3-1 所示，在“追加”报警内容后，勾选“自动保存产生的报警信息”。

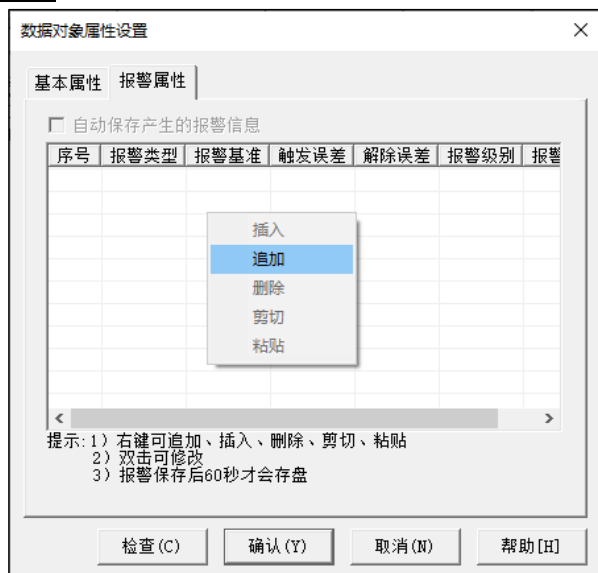


图 4.3-1 报警属性设置界面



图 4.3-2 报警浏览构件

- ② 双击进入“窗口 0”组态画面，如图 4.3-2 所示，在工具箱中选择“报警浏览”构件，并绘制。
- ③ 双击绘制的“报警浏览构件”，如图 4.3-3 所示：在数据来源页，数据类型选择“历史报警数据”，报警对象留空（“报警对象”不关联任何对象时，该报警浏览构件将显示实时数据库中，所有勾选了“自动保存产生的报警信息”变量的历史报警数据）。
- ④ 如图 4.3-4 所示：在显示属性页，按照需要勾选显示内容：“编号”、“日期”、“时间”、“对象名”、“报警值”、“报警描述”，并设置合适的列宽；其他功能页用户根据自己的需要进行设置。设置完成点击“确认”。

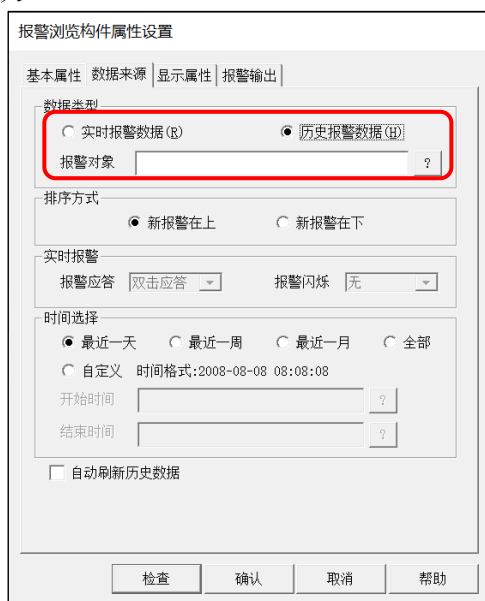


图 4.3-3 数据来源设置

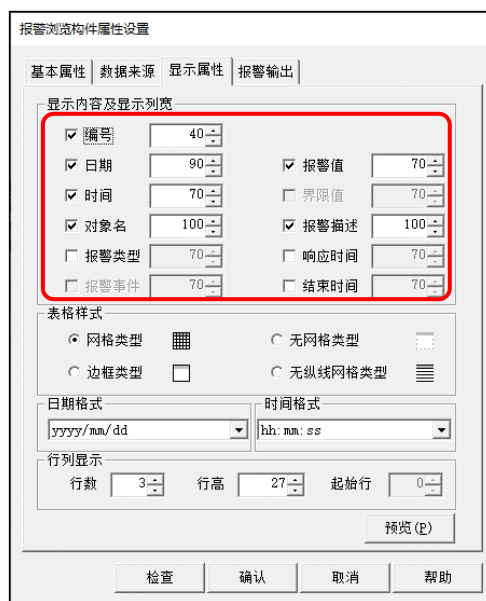


图 4.3-4 显示属性设置

- ⑤ 完成上述步骤后，即可运行工程，当变量“设备 0_读写 M012_3”产生报警后，TPC 将在步骤 3 添加到报警浏览构件中显示自动保存产生的报警信息。

4.3.2 历史报警删除

McgsPro 提供三种方法删除历史报警数据：

- 方法 1：下载工程时，在下载配置界面删除历史报警数据，如图 4.3-5。



图 4.3-5 下载工程删除历史报警

- 方法 2：工程运行时，历史报警数据存储达到 1M 时，自动删除最老的 20%数据。
- 方法 3：工程运行时，通过执行事先组态的函数“!ClearHistoryAlarmData()”，清空所有历史报警数据。
!ClearHistoryAlarmData()：删除所有的历史报警数据

4.3.3 历史报警导出

用户可在工程运行时，执行事先组态的函数“ExportHisDataToCSV”，将触摸屏中记录的历史报警数据以 CSV 格式导出到触摸屏的用户文件区或者 U 盘中，用户可在电脑端使用 EXCEL 打开导出的历史报警数据。

使用函数“ExportHisDataToCSV”导出触摸屏中的历史报警数据方法，可参考进阶篇章 5.2.4。

注意：使用函数“ExportHisDataToCSV”导出历史报警数据，组对象名称必须填写“Mcgs_HistoryAlarm”

4.4 报警信息显示和操作

4.4.1 报警信息翻页

McgsPro 中的“报警浏览构件”除了显示报警信息外，用户还可以使用报警浏览构件的方法函数，实现刷新报警数据和翻页等功能。操作步骤如下：

- ① 首先进行报警功能组态，并使用图 4.4-1 中的“报警浏览”构件显示报警信息，详细组态过程请参考进阶篇章 4.3。



图 4.4-1 报警浏览构件

- ② 打开“报警浏览”构件所在的用户窗口，并在该窗口中添加 1 个图 4.4-1 所示的标准按钮构件。
- ③ 点击鼠标左键选中“报警浏览”构件，记住该构件的名称以及所在窗口。如图 4.4-2 所示的报警浏览构件位于“窗口 0”，名称为“控件 0”。

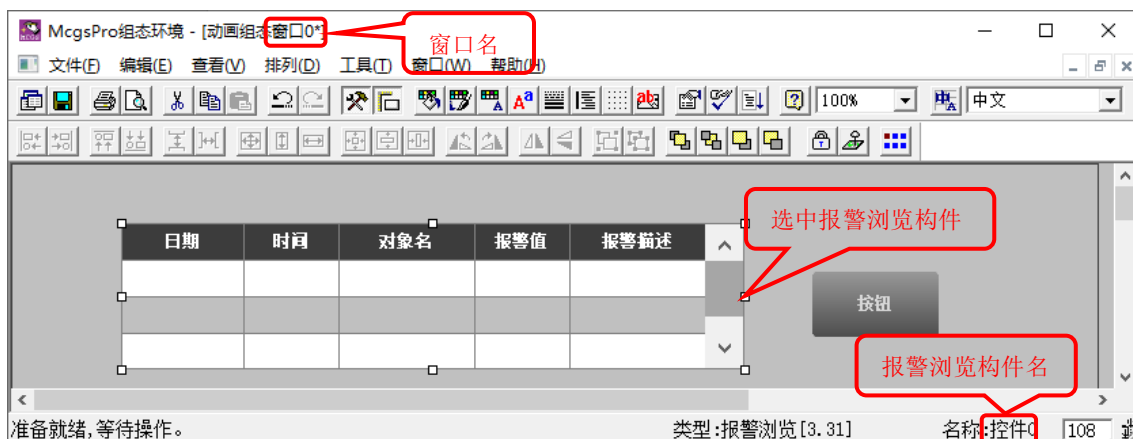


图 4.4-2 构件基础信息

- ④ 双击图 4.4-2 中的标准按钮构件，如图 4.4-3 所示，在其“基本属性”页修改显示文本，如修改为“下一页”。

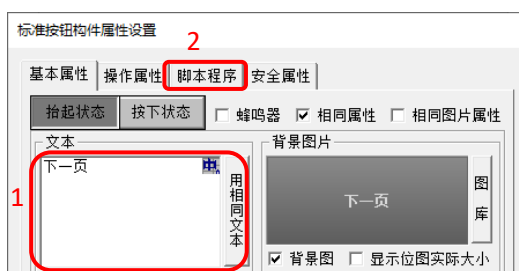


图 4.4-3 显示文本设置

- ⑤ 点击图 4.4-3 中的“脚本程序”，进入脚本程序功能页 → 点击“打开脚本程序编辑器”。按照步骤 3 展示的构件名称和所在窗口名称，在图 4.4-4 右侧对象树依次展开“窗口 0” → “控件 0” → “方法”，双击方法函数“PageDown”，将其添加到左侧的脚本程序编辑框。

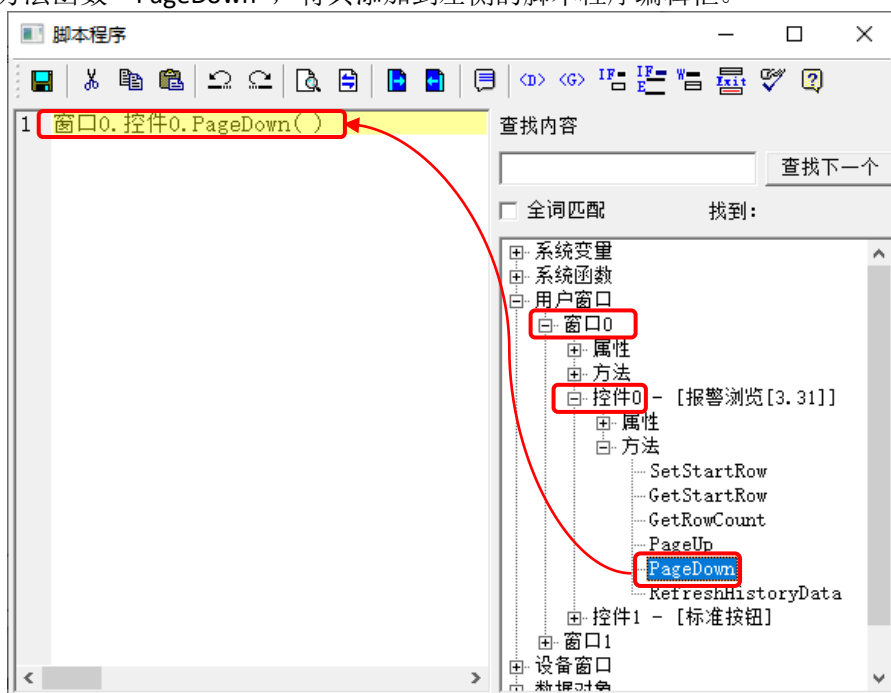


图 4.4-4 报警浏览构件方法函数

报警浏览构件方法函数解释：

- SetStartRow: 设置报警浏览构件显示起始行；
- GetStartRow: 获取报警浏览构件当前起始行；

- GetRowCount: 获取报警浏览构件报警信息总条数;
- PageUp: 向上翻页;
- PageDown: 向下翻页;
- RefreshHistoryData: 刷新历史报警信息。

注意: 不同构件的方法函数存在差异, 不能直接套用。

- ⑥ 完成步骤 5, 检查并保存脚本, 点击“确认”。将工程下载到 TPC 并运行, 当报警信息条数超过报警浏览构件当前显示页, 就可以通过点击“下一页”标准按钮, 实现对窗口 0 中的报警浏览构件进行翻页操作。

4.4.2 报警值修改

用户可以通过以下方法修改报警值:

方法 1: 工程组态时直接在“实时数据库”中找到对应的变量, 双击变量进入其“报警属性”设置页面设置报警值, 再将工程重新下载到 TPC。参考进阶篇章节 4.2 设置变量报警功能部分。此方法适合工程组态人员使用。

方法 2: 在工程运行时, 通过事先组态的函数(SetAlmValue)修改变量报警值。采用此方法, 即使不懂组态的人员也可以很方便地修改变量报警值。

本节我们以“方法 2”为例进行讲解。操作步骤如下:

目标: 以进阶篇章节 4.2.2 为例, 在工程运行时, 将变量“新报警值”的值赋给变量“设备 0_读写 VWUB049”的上限报警值, 并且要求修改的报警值在重启 TPC 后仍然有效。

- 查看报警序号: 首先按照进阶篇章节 4.2.2 进行变量上下限报警组态, 组态完成后, 变量“设备 0_读写 VWUB049”报警属性显示如图 4.4-5 所示, 可以看到该变量上限报警序号为“0”, 下限报警序号为“1”。

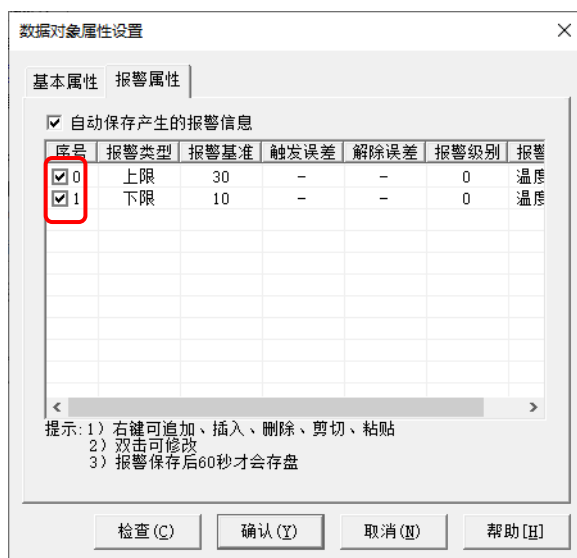


图 4.4-5 报警属性

- 添加输入框: 在实时数据库新增浮点数变量“新报警值”, 并在用户窗口“窗口 0”中添加一个输入框。
- 输入框关联变量: 如图 4.4-6 所示, 双击添加的输入框, 在其“操作属性”页面关联浮点数变量“新报警值”。



图 4.4-6 关联变量“新报警值”

- 注意步骤 1 的操作，变量“设备 0_读写 VWUB049”上限报警序号为“0”。
- 添加报警设置脚本和初值保存脚本：在“窗口 0”添加一个标准按钮构件，按照需要修改按钮的文本内容，并在构件的“脚本程序”页面添加以下脚本
`!SetAlmValue(设备0_读写 VWUB049,0,新报警值,12)` '将变量“新报警值”设置为变量“设备 0_读写 VWUB049”的上限报警
`!SaveSingleDataInit(新报警值)` '对变量“新报警值”保存初值
`!FlushDataInitValueToDisk()` '将初值数据刷盘保存

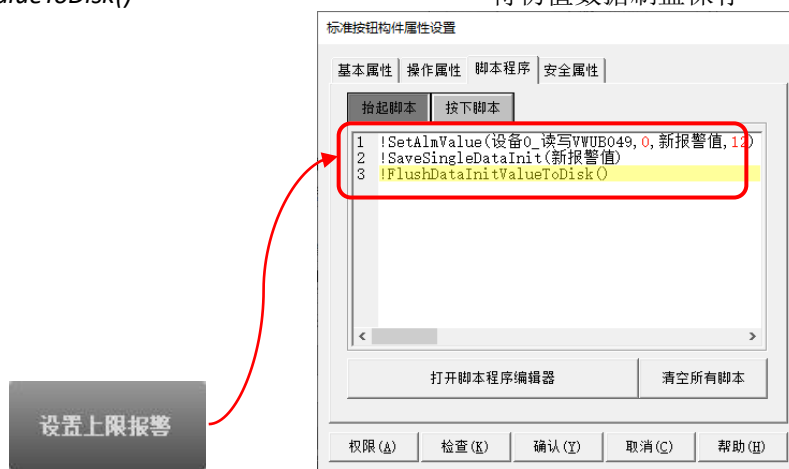


图 4.4-7 添加脚本

报警值设置函数意义：

`!SetAlmValue(DataName,AlarmIndex,Value,Flag)`

DataName，数据对象名，只支持整数变量和浮点数变量

AlarmIndex，整数，指定报警序号

Value，浮点数，新的报警值

Flag，浮点数，标志要读取何种报警属性值，具体意义如下：

- = 9，报警使能值
- = 10，报警存储状态值
- = 11，报警打印值（目前备用状态）
- = 12，报警基准值（开关量与限值报警的报警值，位报警不能修改）
- = 13，报警触发误差值（位报警的指定值，偏差报警的报警值）
- = 14，报警解除误差值（部分报警类型此值无意义）
- = 15，报警优先级值（目前备用状态）

- = 16, 报警状态值（不可修改）
- = 17, 报警类型值（不可修改）
- 启动策略设置：由于 TPC 重启后，变量报警值会恢复成组态时的设置值，所以需要步骤 5 对变量“新报警值”进行初值保存，并在启动策略中使用再次对变量进行报警值修改操作，保证重启 TPC 后，变量“设备 0_读写 VWUB049”的报警值与设置值一致。如图 4.4-8 所示，启动策略中新增一个策略行，并添加以下脚本
`!SetAlmValue(设备 0_读写 VWUB049,0,新报警值,12)` 将变量“新报警值”设置为变量“设备 0_读写 VWUB049”的上限报警

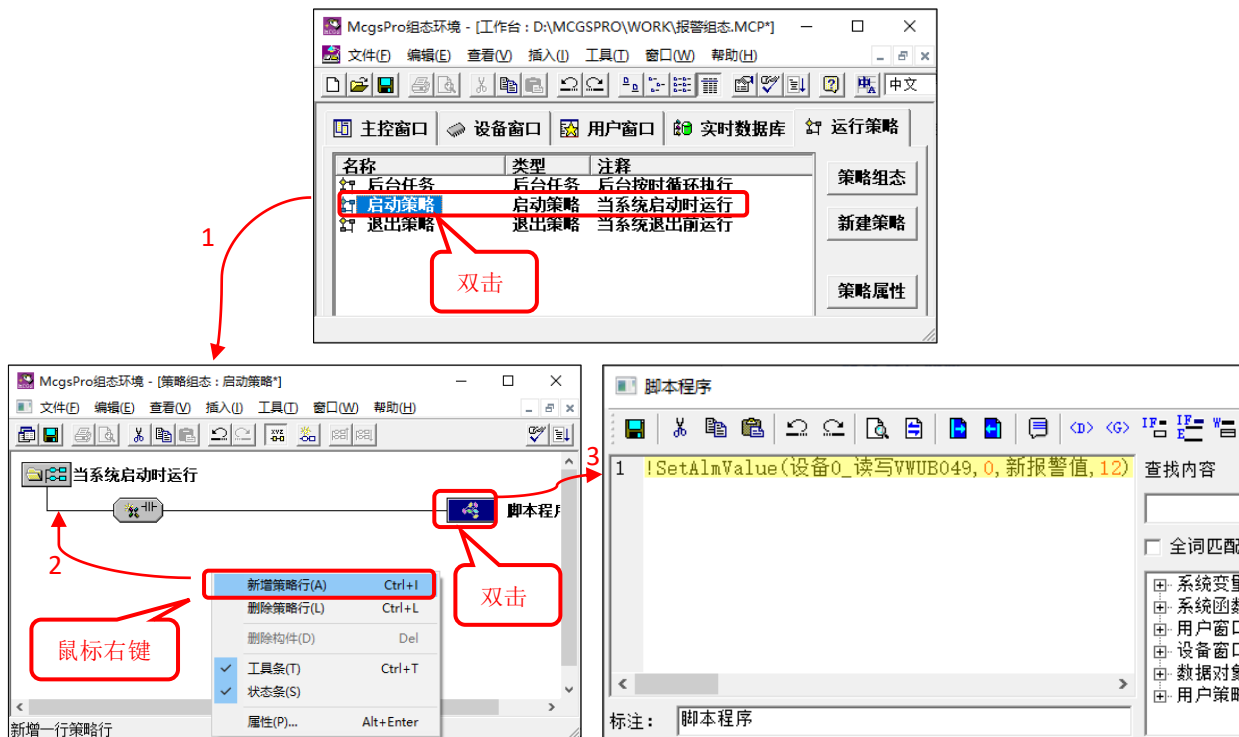


图 4.4-8 启动策略添加报警设置脚本

- 将工程下载到 TPC 正常运行，可以通过输入框向变量“新报警值”赋值，再通过图 4.4-7 中组态的“设置上限报警”按钮，将变量“新报警值”的当前值设置为变量“设备 0_读写 VWUB049”的上限报警值。
- 修改报警值后，重启 TPC，在工程启动过程会首先触发“启动策略”，将变量“新报警值”的当前值设置为变量“设备 0_读写 VWUB049”的上限报警值。（注意：工程下载到 TPC 后首次启动，也会触发“启动策略”，并执行启动策略中的报警值修改脚本）

4.5 报警信息配置

McgsPro 提供 2 种方法供用户对变量进行报警信息组态配置。

方法 1：直接在“实时数据库”中双击对应变量的“报警属性”功能页，配置报警信息，进阶篇章 4.2 就是采用的该方法。

方法 2：通过 McgsPro 提供的“报警统一配置”功能，批量进行变量报警信息配置。

本节教程以方法 2 为例，讲解“报警统一配置”使用方法。

4.5.1 进入报警统一配置

如图 4.5-1，通过 McgsPro 组态软件菜单栏 → “工具” → “报警统一配置”，进入报警统一配置界面

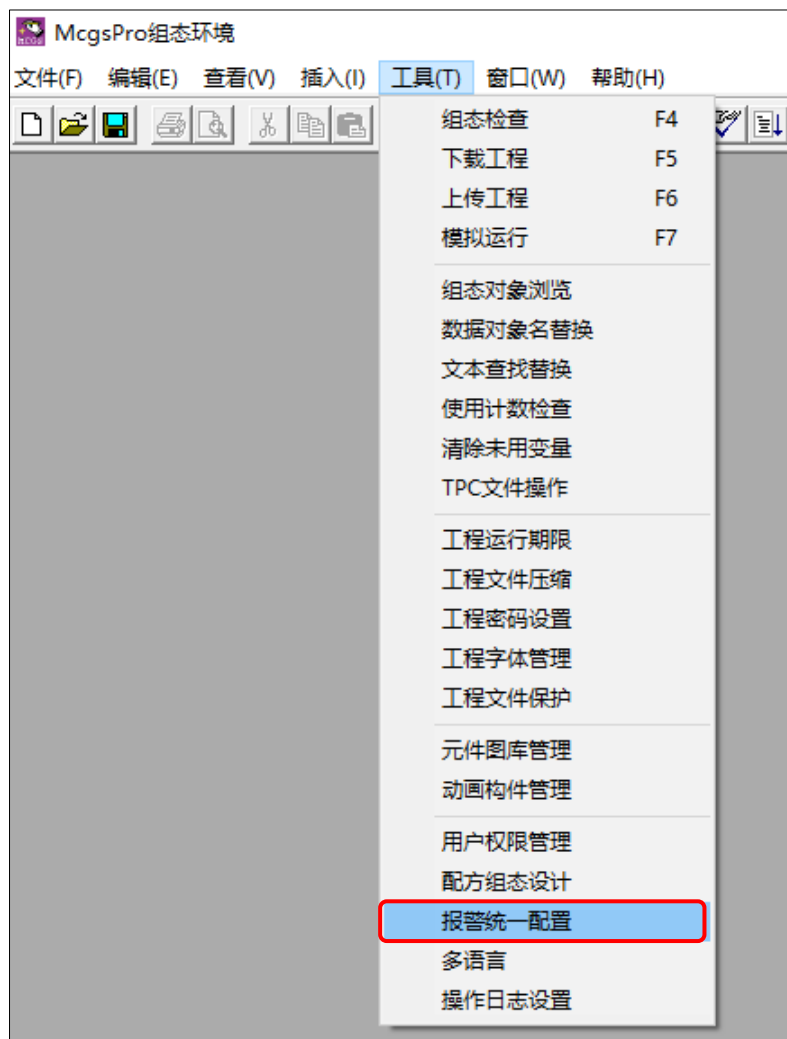


图 4.5-1 打开报警统一配置

报警统一配置界面打开后，将显示当前工程中已有的报警信息，如图 4.5-2 所示：

报警统一配置									
文件(F) 编辑(E) 帮助(H)									
序号	变量名称	变量类型	报警类型	立即启用	基准值	触发误差	解除误差	报警描述	
1	设备0_读写M012_3	整数	开关量	☒	1	-	-	水满了	
2	设备0_读写VWUB049	浮点数	上限	☒	30	-	-	温度太高了	
3	设备0_读写VWUB049	浮点数	下限	☒	10	-	-	温度太低了	
...	双击此行新增报警								

图 4.5-2 报警信息统一配置

4.5.2 设置统一报警配置

● 新增报警

可通过以下三种方式新增报警行：

- ① 通过图 4.5-3 中的“编辑”→“添加一行”，实现新增报警行。

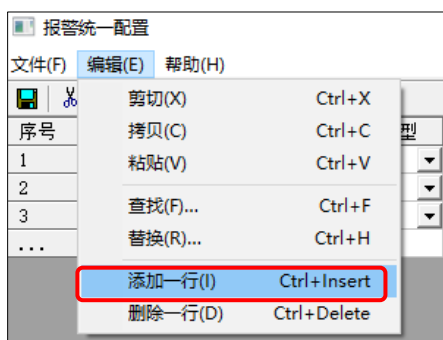


图 4.5-3 菜单栏新增报警行

- ② 通过图 4.5-4 中的工具栏图标 , 实现新增报警行。

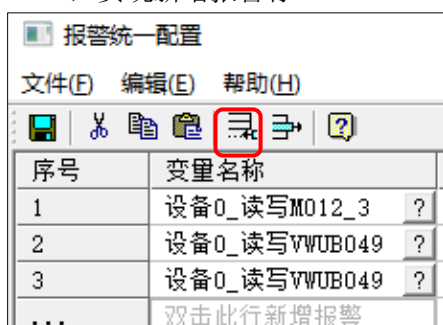


图 4.5-4 工具栏新增报警

- ③ 通过图 4.5-5 中的“双击此行新增报警”，实现新增报警行。

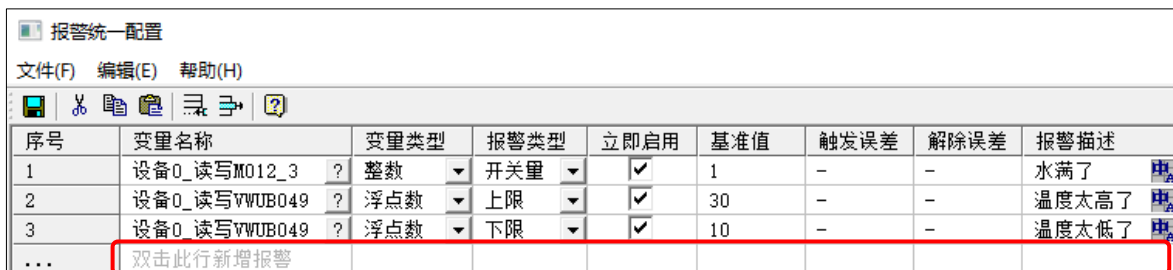


图 4.5-5 双击新增报警行

● 编辑报警信息

如图 4.5-6，通过“报警统一配置”页面，设置变量名称、变量类型、报警类型、是否立即启用、基准值、触发误差、解除误差、报警描述。

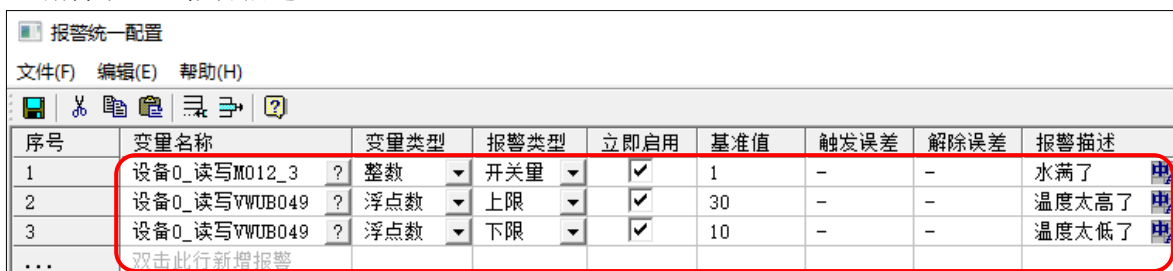


图 4.5-6 编辑报警信息

4.5.3 报警信息导出和导入

利用 McgsPro 的“报警统一配置”功能，可以将报警信息以 xml 格式导出，支持在电脑上使用 EXCEL 打开并进行编辑后，再重新导入工程，减少用户重复工作量。

● 导出报警信息

如图 4.5-7，通过报警统一配置界面的菜单栏 → “文件” → “导出”，可以弹出文件保存选择对话框，用

户手动指定导出的文件路径和文件名后，即可导出报警信息。

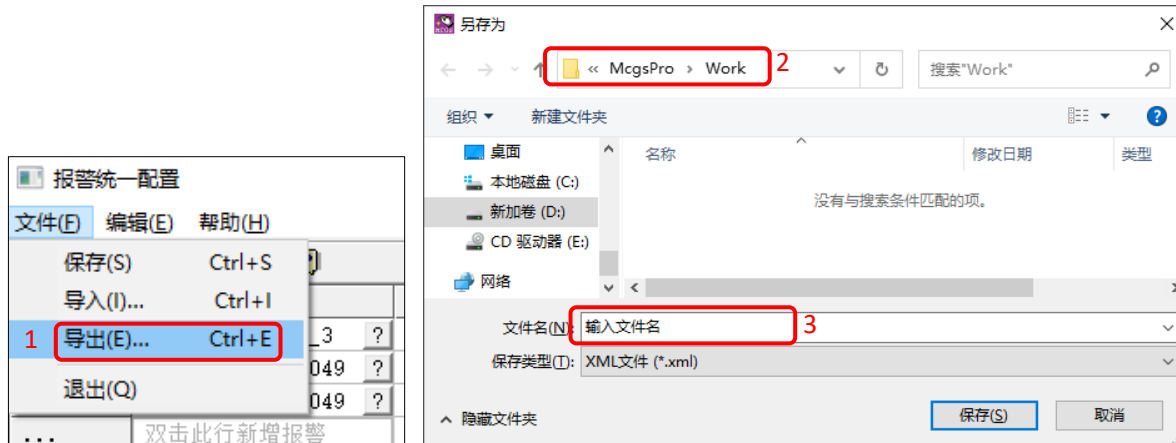


图 4.5-7 导出报警信息

● 导出文件编辑

如图 4.5-8 所示，使用 Excel 将导出的 xml 文件打后进行编辑。注意：完成编辑后仍然需要保存为 XML 格式。

ID	变量名称	变量类型	报警类型	可用性	基准值	触发误差	解除误差	多语言ID	多语言类型名称	报警描述
1	设备0_读写M000_0	开关型	开关量	1	1	-	-	-1	中文	温度过高!
2	设备0_读写M000_1	开关型	开关量	1	1	-	-	-1	中文	压力超标!
3	设备0_读写M000_2	开关型	开关量	1	1	-	-	-1	中文	水满了!
4	设备0_读写VWUB100	数值型	上限	1	50	-	-	-1	中文	上限报警!
5	设备0_读写VWUB200	数值型	值==	1	2	0	0	-1	中文	故障代码01
6	设备0_读写VWUB200	数值型	值==	1	2	0	0	-1	中文	故障代码02
7	设备0_读写VWUB200	数值型	值==	1	3	0	0	-1	中文	故障代码03
8	设备0_读写VWUB200	数值型	值==	1	4	0	0	-1	中文	故障代码04
9	设备0_读写VWUB200	数值型	值==	1	5	0	0	-1	中文	故障代码05

图 4.5-8 编辑报警信息文件

● 导入报警

如图 4.5-9，通过报警统一配置界面的菜单栏 → “文件” → “导入”，弹出“旧报警”操作对话框，选择“是”，导入前会清空“报警统一配置”中已有的报警信息；选择“否”，不清空“报警统一配置”中的报警信息，选择“取消”，则放弃导入操作。



图 4.5-9 导入报警信息

第 5 章 趋势曲线和数据处理

TPC 的一个重要应用就是显示实时趋势和历史趋势曲线，以及对接收到 PLC 数据进行处理。本章教程主要讲解 McgsPro 的趋势曲线显示和数据处理功能。

5.1 实时趋势曲线

通过 McgsPro 提供的实时曲线构件，可以实现工程运行时显示指定变量的实时趋势，实时曲线构件通过 McgsPro 组态软件工具箱进行添加，如图 5.1-1：

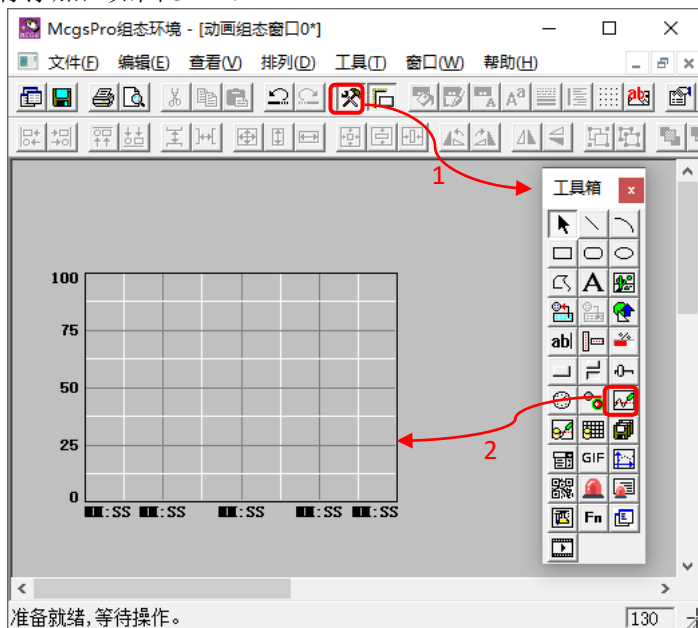


图 5.1-1 添加实时曲线构件

实时趋势曲线组态流程如图 5.1-2 所示：

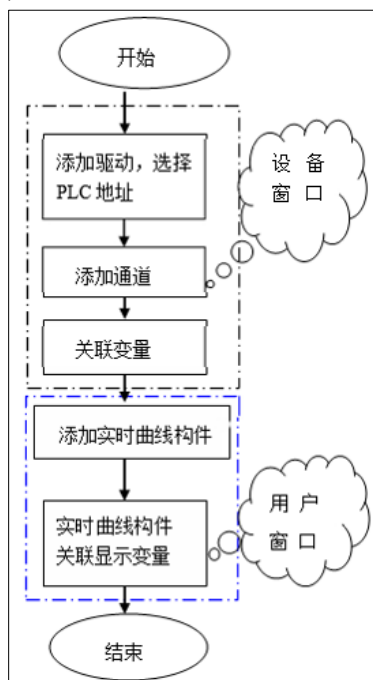


图 5.1-2 实时趋势曲线组态

本节教程将讲解实时曲线构件支持的 2 种趋势曲线类型：

绝对时钟趋势曲线：Y 轴为目标变量的值，X 轴为系统时间，显示变量相对时间的变化曲线，是最常用的实时曲线类型。

相对时钟趋势曲线：Y 轴为目标变量的值，X 轴为其他变量，显示一个变量相对于另一个变量的变化曲线。

5.1.1 绝对时钟趋势曲线

以“模拟设备”为例，通过实时曲线构件显示模拟设备通道地址变量“温度”的趋势曲线，要求曲线的 Y 轴为“温度”值，X 轴为系统时间，并可通过按钮清除实时曲线构件中的曲线。我们通过组态一个样例来学习实时趋势曲线，图 5.1-3 是该样例的运行效果。

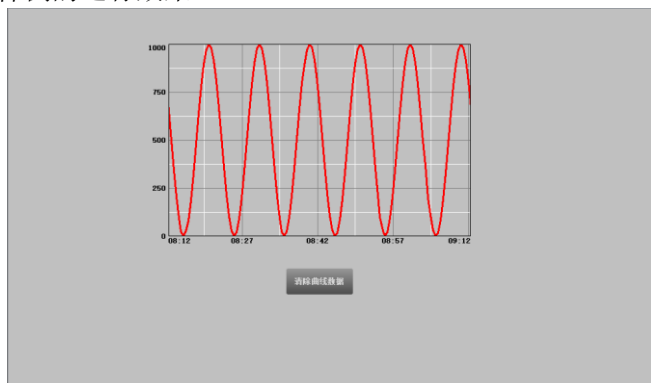


图 5.1-3 实时趋势曲线运行效果

方案：在设备窗口中添加驱动“模拟设备”，在“模拟设备通道 1”关联“温度 01”变量，然后进入用户窗口添加实时曲线构件，并在构件的画笔属性页关联变量“温度 01”，工程运行时即可显示“模拟设备通道 1”的实时曲线。

实时趋势曲线组态：

- ① 添加模拟设备：如图 5.1-4，新建工程，进入工程的“设备窗口”，将“设备管理”→“通用设备”→“模拟数据设备”→“模拟设备”，添加到设备工具箱，再将“模拟设备”添加到设备窗口。

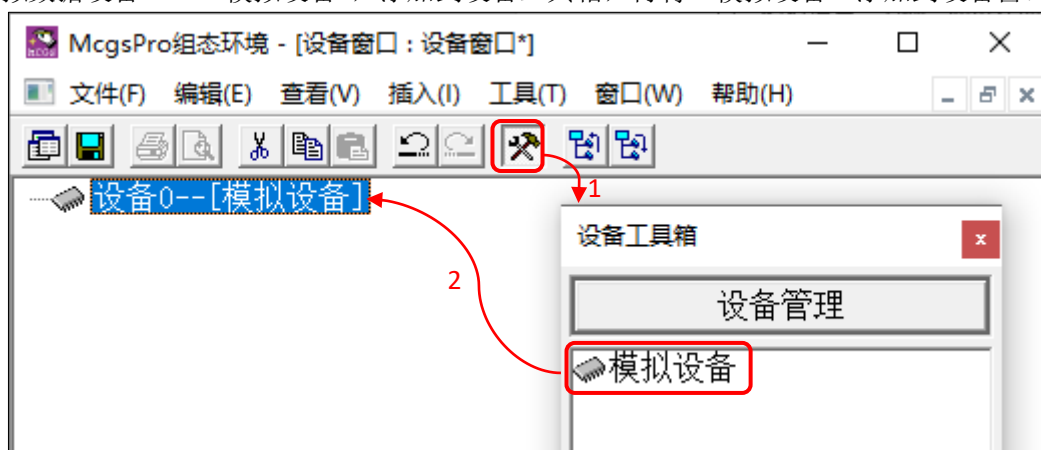


图 5.1-4 添加模拟设备

- ② 通道连接变量：双击上图“设备 0--[模拟设备]”，进入图 5.1-5 所示的设备编辑窗口，点击“快速连接变量”，数据对象为“温度 01”，开始通道“1”，通道个数“1”，设置完成，点击“确认”，完成连接变量操作。



图 5.1-5 连接变量

- ③ 添加变量：点击图 5.1-5 右下方的确认按钮，在弹出的“添加变量”对话框中，选择“全部添加”，变量“温度 01”将自动添加到实时数据库。
- ④ 绘制实时曲线构件：在工程的用户窗口中新建“窗口 0”，双击进入“窗口 0”组态界面，按照图 5.1-6 所示步骤，在窗口 0 中绘制“实时曲线构件”。

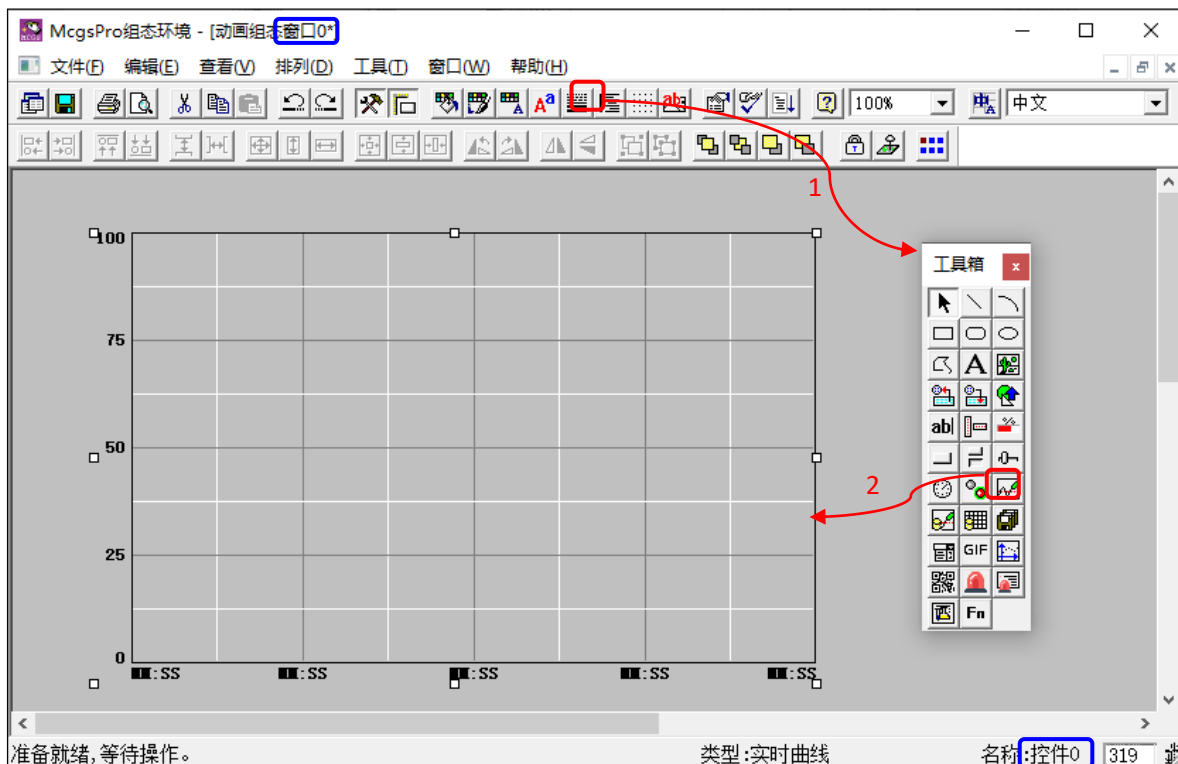


图 5.1-6 绘制实时曲线构件

- ⑤ 设置曲线类型：双击绘制的实时曲线构件，如图 5.1-7，在“基本属性”页设置构件显示外观，曲线类型选择：绝对时钟趋势曲线（X 轴为系统时间）



图 5.1-7 设置曲线类型

- ⑥ 设置曲线标注：在“标注属性”页，按照需要设置 X 轴和 Y 轴的标注属性。本次我们将 X 轴时间单位设置为“秒钟”，X 轴长度设置为“60”，Y 轴最大值设置为：1000.0。

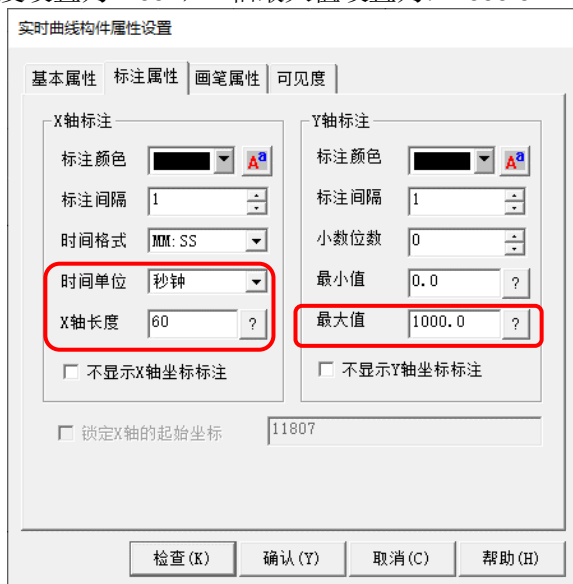


图 5.1-8 设置标准属性

注意：“锁定 X 轴的起始坐标”功能，只有当曲线类型选择“绝对时钟趋势曲线”，且 X 轴标注中的“时间单位”选择“小时”，此功能才可用。

- ⑦ 设置画笔属性：“画笔属性”是实时曲线构件的核心功能设置，如图 5.1-9，用户通过点击“画笔属性”中的  关联需要显示的曲线变量，还可设置曲线颜色和线型。我们在“曲线 1”关联变量“温度 01”，并设置曲线颜色和线型，完成设置后点击“确认”保存。

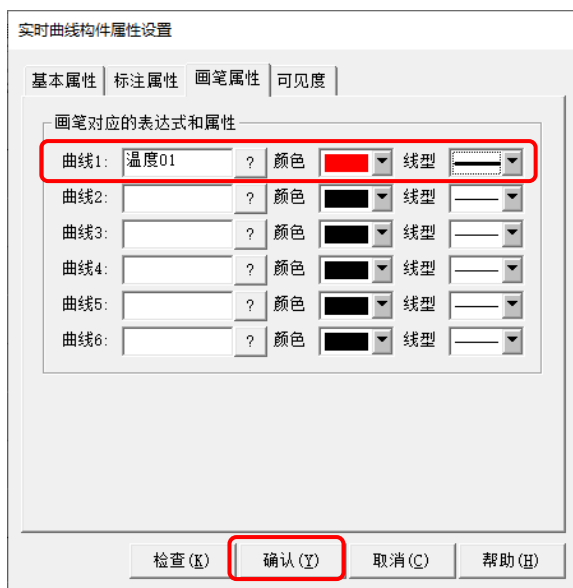


图 5.1-9 设置画笔属性

- ⑧ 曲线清除功能说明：McgsPro 提供构件方法函数，我们可以采用实时曲线构件的方法函数实现曲线清除功能。
- ⑨ 添加曲线清除功能：从图 5.1-6 可以看到，绘制的实时曲线构件位于“窗口 0”，构件名为“控件 0”。接下来在“窗口 0”中绘制一个标准按钮构件，双击标准按钮构件修改其显示文本为“清除曲线数据”，再进入按钮“脚本程序”→“打开脚本程序编辑器”→“用户窗口”→“窗口 0”→“控件 0”→“方法”→“ClearData”，将清除曲线数据函数“ClearData”添加到脚本编辑框，效果如图 5.1-10 所示。

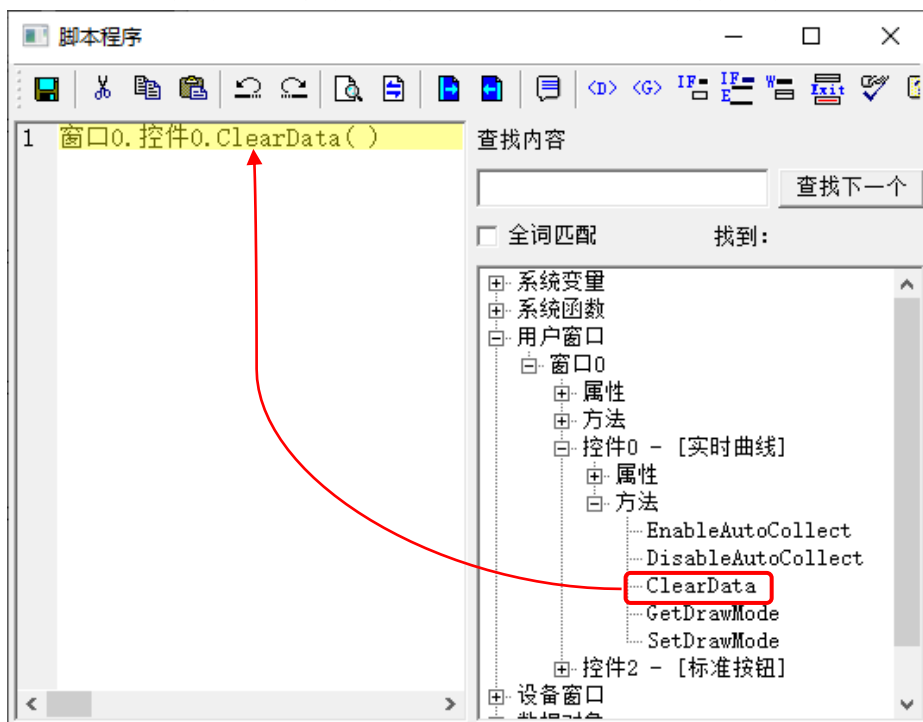


图 5.1-10 添加方法函数“ClearData”

- ⑩ 点击保存步骤 9 添加到方法函数。将工程下载到 TPC 或者直接模拟运行，“模拟设备”将对通道 1 关联的变量“温度 01”，以正弦波的方式赋值 0~1000，实时曲线构件记录的“温度 01”数值曲线如图 5.1-11 所示：

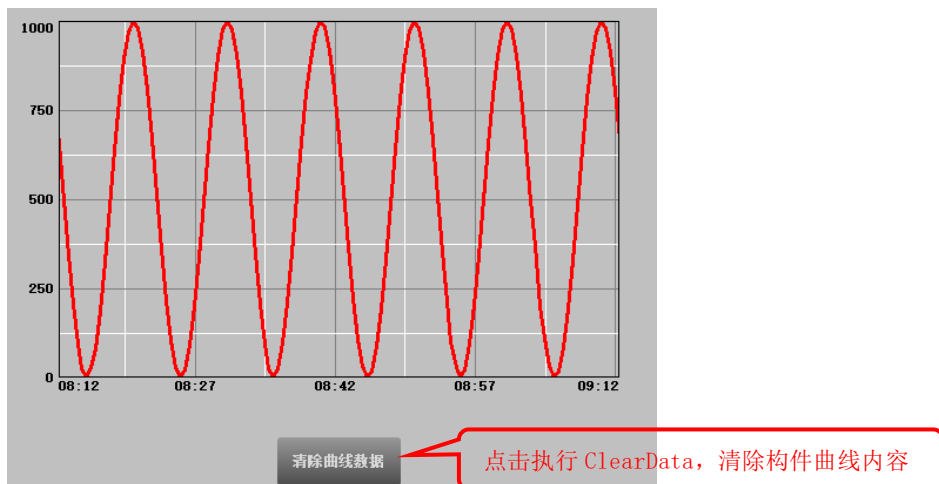


图 5.1-11 实时曲线构件运行效果

注意:

- 每个实时曲线构件最多可显示 6 条曲线;
- 每条曲线最多支持 300 个数据点;
- 重启 TPC, 实时曲线构件记录的曲线内容会清空。

5.1.2 相对时钟趋势曲线

McgsPro 中实时曲线构件的相对时钟趋势曲线功能已经由 XY 曲线构件代替, 在设置界面中保留该功能只是为了保持对低版本软件的兼容性, 建议在 McgsPro 中使用 XY 曲线构件代替实时曲线构件的相对时钟趋势曲线功能。

5.2 历史数据功能

本节教程主要介绍历史数据存盘、显示, 以及历史数据操作。

5.2.1 历史数据存盘

McgsPro 中的历史数据以组对象为单位进行存盘, 用户只需要在“实时数据库”中将需要存盘的变量(变量可以是整数、浮点数、字符串)添加到组对象, 再对组对象进行存盘设置即可。下位机(如 PLC)传入触摸屏的数据存盘流程如图 5.2-1 所示:

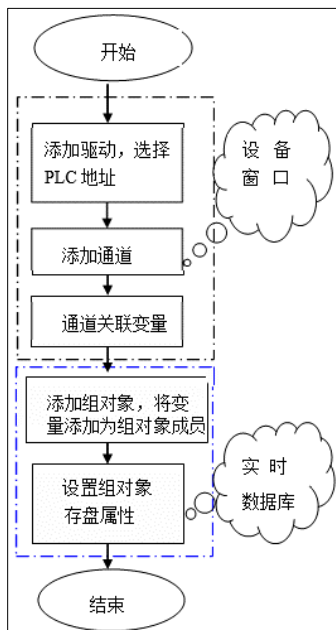


图 5.2-1 设置历史数据存盘

本节以“模拟设备”为例，演示下位机历史数据存盘设置方法。假设用户需要保存下位机数据“温度 01”和“压力 01”，要求每隔 3 秒保存一次数据，且数据重启后不会丢失，可进入如下操作：

方案：在设备窗口中添加“模拟设备”驱动，关联变量“温度 01”和“压力 01”，在实时数据库新建组对象“储水罐数据”，设置组对象存盘周期为 3 秒，并将数据存储在磁盘，随后将“温度 01”和“压力 01”添加为组对象“储水罐数据”成员。

历史数据存盘组态：

- ① 添加模拟设备驱动：新建工程，在设备窗口中添加驱动“模拟设备”，

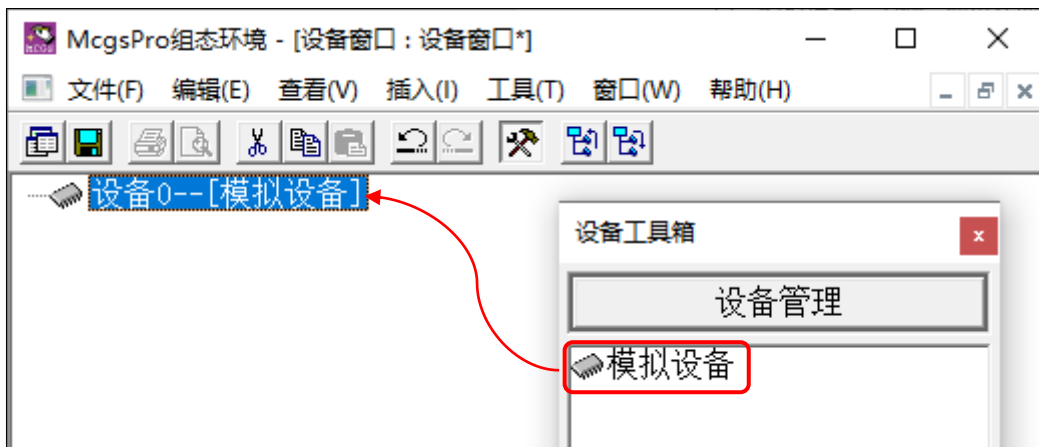


图 5.2-2 添加模拟设备

- ② 通道连接变量：双击图 5.2-2 中“设备 0--[模拟设备]”，进入图 5.2-3 所示的设备编辑窗口，点击“快速连接变量”，数据对象为“温度 01”，开始通道“1”，通道个数“1”，设置完成，点击“确认”；按照同样的操作，在通道 2 关联变量“压力 01”。



图 5.2-3 连接变量

- ③ 添加变量：点击上图右下方的确认按钮，在弹出的“添加变量”对话框中，选择“全部添加”，变量“温度 01”、“压力 01”将自动添加到“实时数据库”。
- ④ 新增对象：关闭“设备窗口”，进入“实时数据库”，如图 5.2-4，点击“新增对象”，新增一个对象。

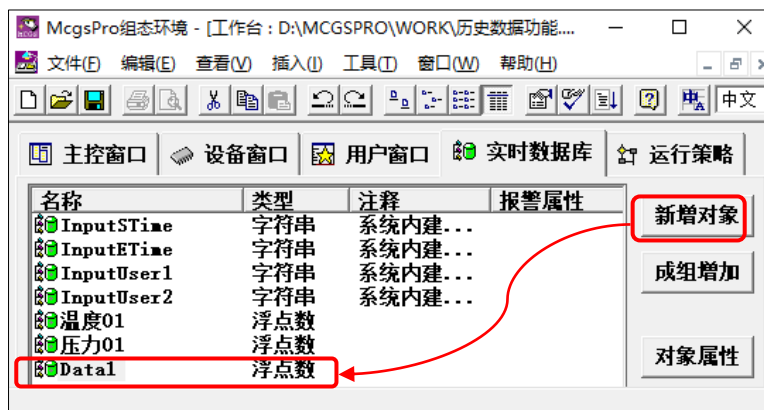


图 5.2-4 新增对象

- ⑤ 设置对象名和类型：双击上图新增的对象“Data1”，修改对象名称为“储水罐数据”，修改对象类型为“组对象”。

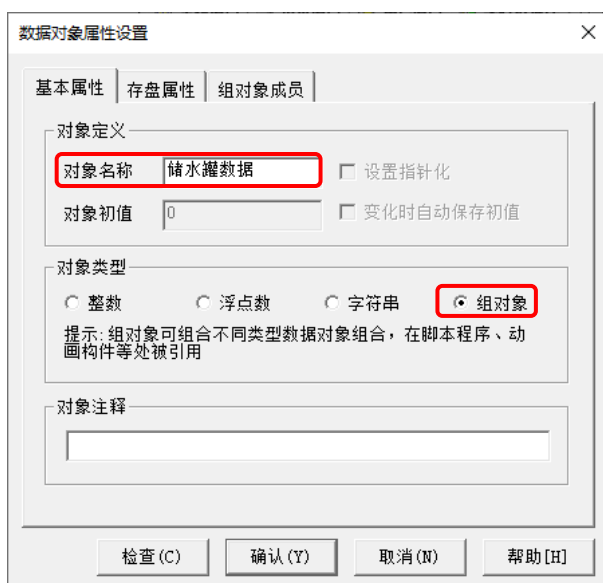


图 5.2-5 设置对象名称和类型

- ⑥ 设置组对象存盘属性：如图 5.2-6，存盘方式选择“定时存储到磁盘”，存盘周期设置为“3 秒”。

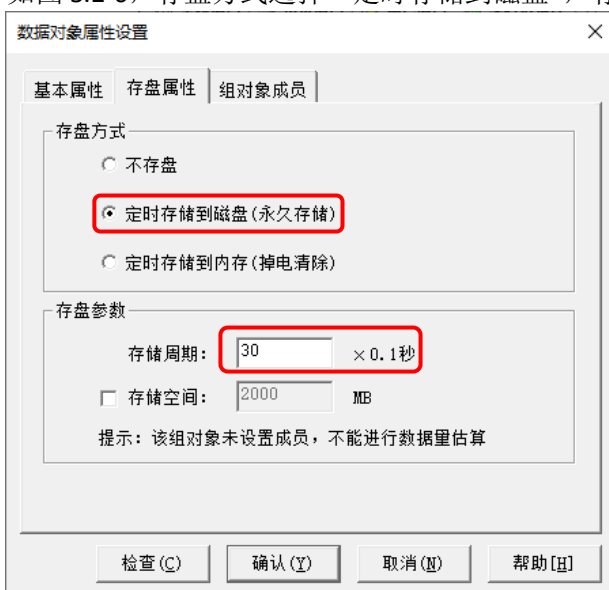


图 5.2-6 设置组对象存盘属性

存盘属性配置信息：

- 不存盘：不进行历史数据存盘。
 - 定时存储到磁盘：历史数据会存储到磁盘。重启 TPC，历史数据仍然存在。（历史存储到磁盘有 60 秒写入间隔）
 - 定时存储到内存：历史数据存储到系统内存，重启后数据清空。
 - 存储周期：设置组对象历史数据存储周期，设置为“0 秒”，则组对象不自动进行周期存盘，但可通过执行存盘函数“!SaveData(组对象名)”，实现存盘功能，工程运行时，每执行 1 次该函数，可对指定的组对象存盘 1 次。
 - 存储空间：系统根据添加的组对象成员计算出的历史数据存储条数。
- ⑦ 添加组对象成员：如图 5.2-7，将变量“温度 01”、“压力 01”添加到右侧的组对象成员列表，对该组对象进行存盘，就是对该组对象中所有成员进行存盘。



图 5.2-7 添加组对象成员

- ⑧ 完成以上步骤，历史数据数据存盘功能组态完成。工程运行时，TPC 将按照设置的“存储周期”对该组对象（成员）进行历史数据存盘。

注意：TPC 磁盘剩余可用空间不足 10%或 5M 时会自动删除最早的 20%，一个组对象每次最多删除 12.5M 历史数据。

5.2.2 历史数据显示

本节继续使用章节 5.2.1 组态的工程，分别以曲线和表格的方式显示历史数据供用户查看，运行效果如图 5.2-8 所示：

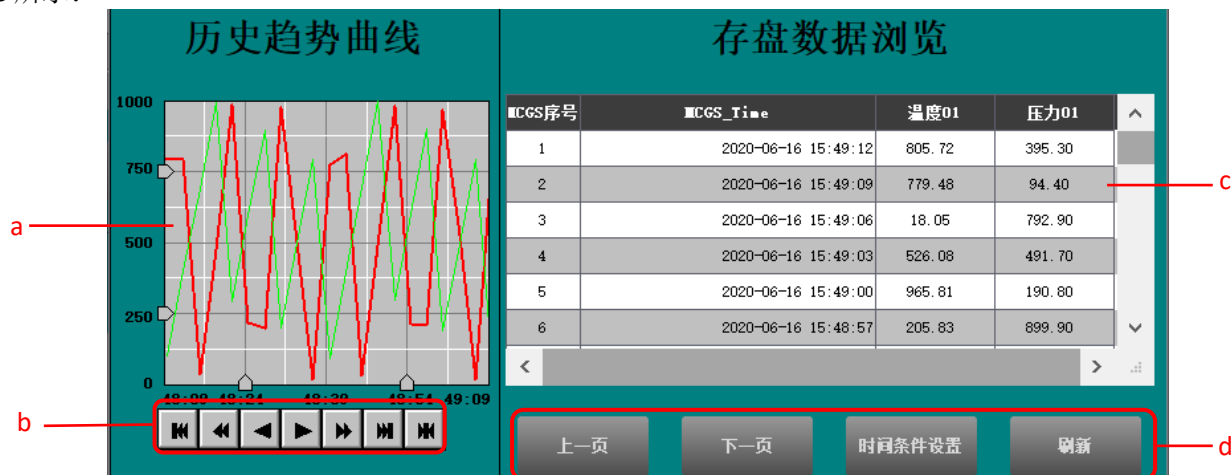


图 5.2-8 历史数据显示

窗口画面功能解释：

- 用历史曲线构件显示历史数据；
- 历史曲线构件翻页操作按钮；
- 用存盘浏览构件显示历史数据；
- 存盘浏览构件部分常用方法函数，实现功能：上翻页、下翻页、显示指定时间段数据、数据刷新。

5.2.2.1 趋势曲线显示历史数据

方案：利用历史曲线构件，显示历史数据趋势曲线，操作步骤如下：

- 进入窗口组态：在章节 5.2.1 所组态工程的基础上，进入工作台，双击“窗口 0”进入画面组态窗口，窗口背景颜色可自行定义。
- 绘制历史曲线构件：如图 5.2-9，在工具箱中选择“历史曲线”构件，并绘制，

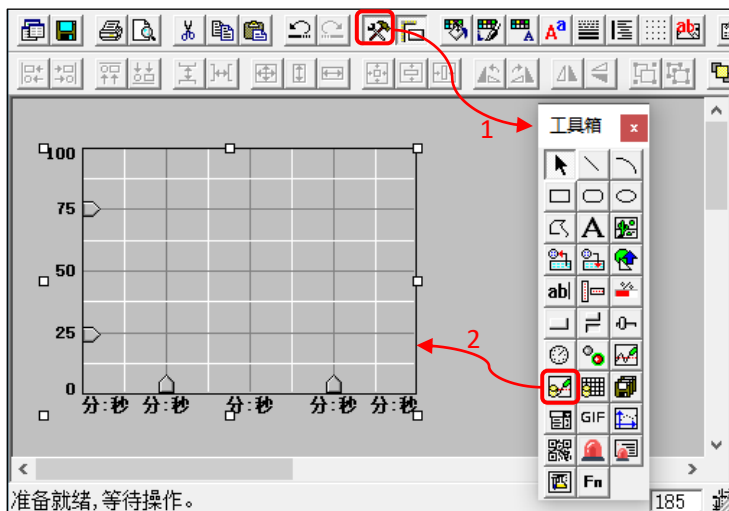


图 5.2-9 绘制历史曲线构件

- 选择数据来源：双击上图绘制历史曲线构件，首先在其“数据来源”页，选择想要显示的变量所在的组对象，如图 5.2-10，数据来源选择“储水罐数据”。

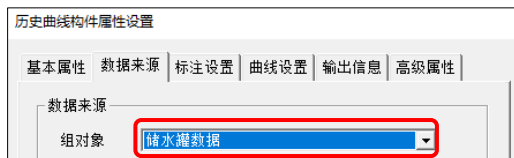


图 5.2-10 选择数据来源

- 标注设置：如图 5.2-11，按照需要设置曲线的 X 轴（时间）标注和曲线的起始点。

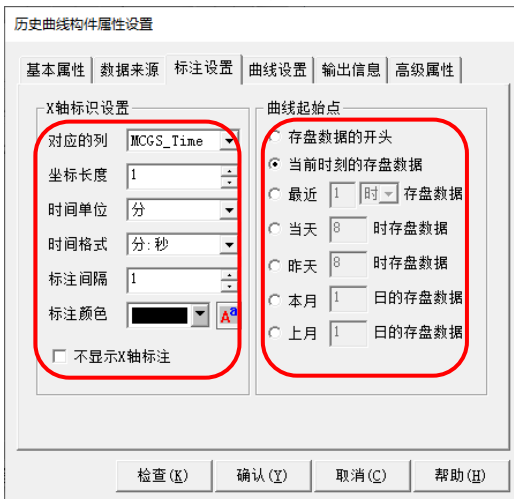


图 5.2-11 标注设置

注：X 轴标识：设置历史曲线构件 X 轴标注长度范围、间隔等；
曲线起始点：设置历史曲线构件 X 轴的起始时间点。

- ⑤ 曲线设置：在“曲线标识”选择待配置的曲线，曲线内容选择“数据来源”对应组对象中的成员，并设置曲线线型、颜色、单位、小数位数、坐标等，如图 5.2-12 所示。

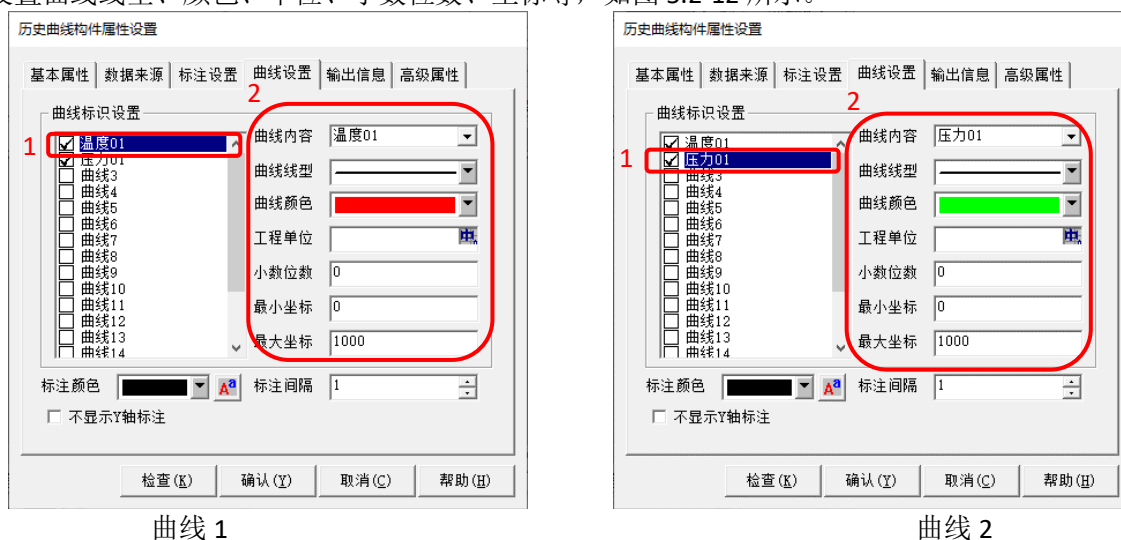


图 5.2-12 曲线设置

- ⑥ 历史曲线构件其他属性设置：“基本属性”主要设置构件的显示外观；“输出信息”设置选择对应曲线时在输出信息窗口展示该曲线信息；“高级属性”设置构件运行时功能按钮显示、曲线信息显示、曲线刷新周期。

这几项设置不影响历史趋势的显示，本节教程“高级属性”设置如图 5.2-13。

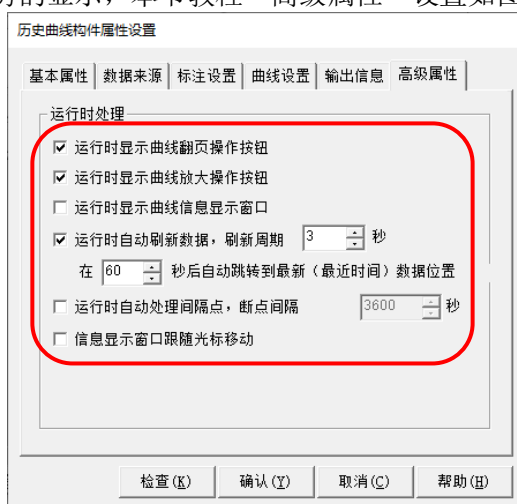


图 5.2-13 高级属性

- ⑦ 运行工程，历史曲线构件自动以高级属性中设置的周期“3 秒”，刷新曲线，效果如图 5.2-14。

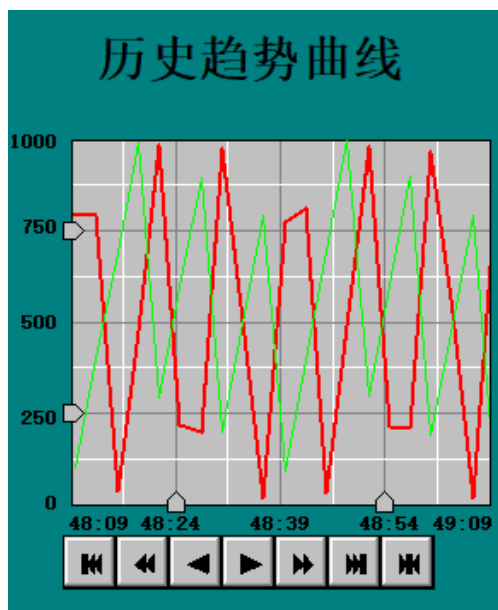


图 5.2-14 运行效果：历史曲线构件

5.2.2.2 存盘浏览构件显示历史数据

方案：利用存盘浏览构件，以表格的方式显示 TPC 中存储的历史数据。操作步骤如下：

- ① 进入窗口组态：在章节 5.2.2.1 所组态工程的基础上，进入工作台的“用户窗口”，双击“窗口 0”进入其组态画面。
- ② 绘制存盘浏览构件：如图 5.2-15，在工具箱中选择“存盘浏览构件”，并绘制。



图 5.2-15 绘制存盘浏览构件

- ③ 构件外观设置：双击上图绘制的存盘构件，在其“基本属性”页设置构件的外观显示属性。
- ④ 选择数据来源：进入其“数据来源”页，选择想要显示的变量所在的组对象，如图 5.2-16 所示，数据来源选择“组对象”→“储水罐数据”。

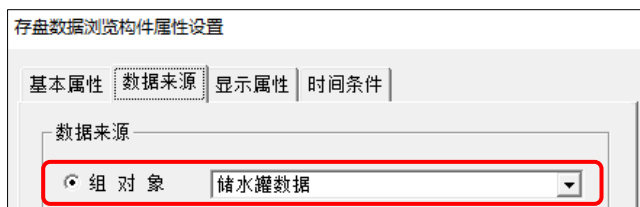


图 5.2-16 选择数据来源

数据来源配置信息解释：

- 组对象：存盘浏览构件将显示指定组对象成员变量的历史数据；（历史数据保存功能需要单独组态）
- 报警浏览：存盘浏览构件将显示变量的历史报警数据；（历史报警功能需要单独组态）

- 操作日志：存盘浏览构件将显示工程操作日志。（操作日志功能需要单独组态）
- ⑤ 设置显示属性：如图 5.2-17，在构件“显示属性”页，点击“复位”，存盘浏览构件自动获取组对象“储水罐数据”的成员列表，用户还可以在成员列表中设置数据的显示格式等。

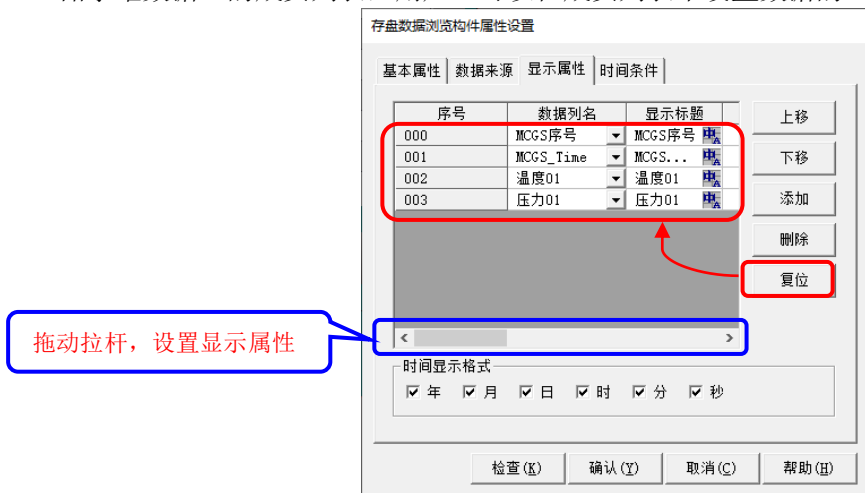


图 5.2-17 设置显示属性

注意：如果用户重新选择了“数据来源”，则必须在“显示属性”页重新点击“复位”，获取新的显示成员列表。

- ⑥ 设置被浏览数据的排序方式和时间范围：在“时间条件”页，如图 5.2-18 所示，选择历史数据的排序方式，以及允许的时间范围。设置完成后，点击“确认”即可。

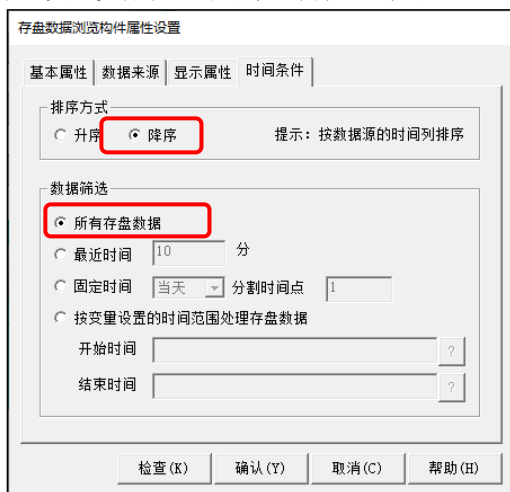


图 5.2-18 时间条件设置

时间条件配置信息解释：

- 排序方式：根据“数据来源”的时间列进行升序或者降序排列数据。
 - 数据筛选：根据“数据来源”的时间列筛选需要显示的数据。
 - 所有存盘数据：显示所有时间段的数据。
 - 最近时间：显示最近“XX”分钟内的数据。
 - 固定时间：显示指定的时间点之后的数据。
 - 按变量设置的时间范围处理存盘数据：显示指定时间段的数据。时间格式为“yyyy-MM-dd hh:mm:ss”，例：“2017-01-01 00:00:00”。
- ⑦ 报警浏览构件历史数据显示功能组态完成，用户还可以选择使用报警浏览构件提供的方法函数，实现在工程运行时，执行“上一页”、“下一页”、“显示指定时间段数据”、“数据刷新”等功能。
- ⑧ 如图 5.2-19 所示，在“窗口 0”添加一个标准按钮构件，修改标准按钮构件的显示文本为“上一页”，再进入其“脚本程序”页，点击“打开脚本程序编辑器”，在右侧对象树依次展开“用户窗口”→

“窗口 0” (存盘浏览构件所在的窗口名) → “控件 1” (存盘浏览构件名称) → “方法”，双击其中的“PageUp”，将上翻页函数添加到脚本编辑框。

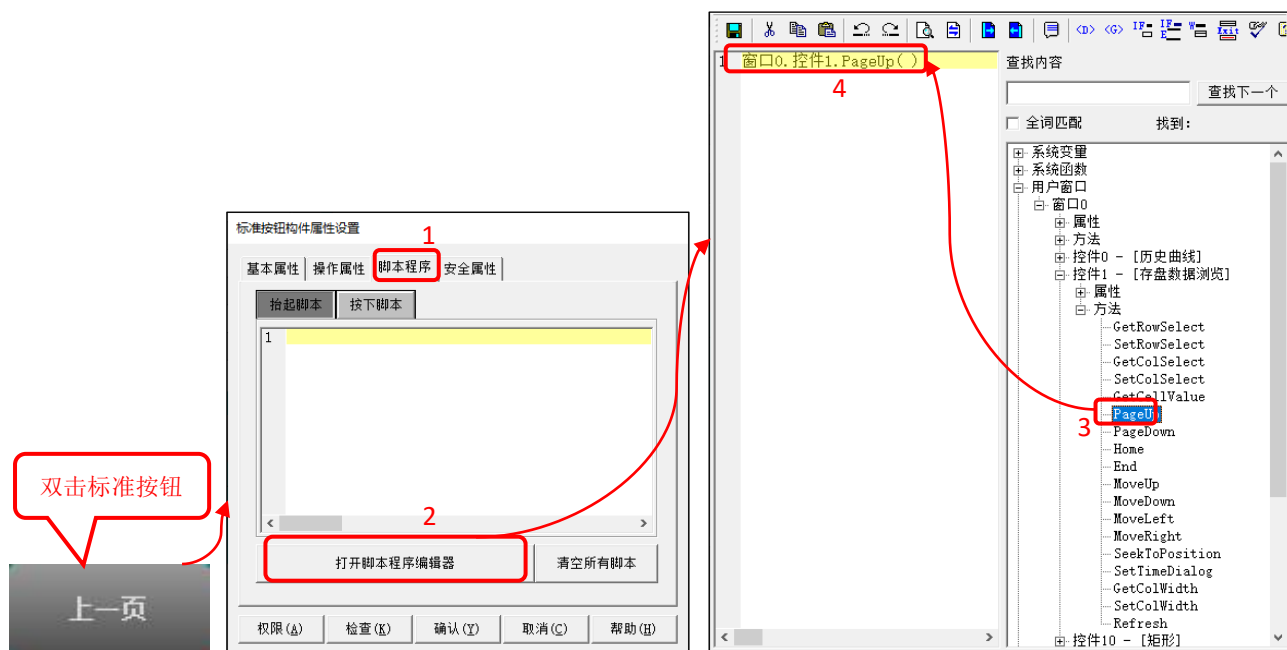


图 5.2-19 添加存盘浏览构件方法函数

⑨ 按照步骤 8 操作，再次绘制标准按钮构件，添加需要的方法函数，完成后如图 5.2-20 所示



图 5.2-20 存盘浏览构件方法函数

- 上一页：窗口 0.控件 1.PageUp() ‘构件向上翻页
- 下一页：窗口 0.控件 1.PageDown() ‘构件向下翻页
- 时间条件设置：窗口 0.控件 1.SetTimeDialog() ‘打开构件时间条件设置对话框
- 刷新：窗口 0.控件 1.Refresh() ‘刷新构件数据

⑩ 运行工程，存盘浏览构件显示历史数据如图 5.2-21 所示。工程运行时，用户可通过方法函数进行上下翻页、打开数据时间筛选对话框、刷新历史数据操作。

存盘数据浏览				
■CGS序号	■CGS_Time	温度01	压力01	^
1	2020-06-16 15:49:12	805.72	395.30	
2	2020-06-16 15:49:09	779.48	94.40	
3	2020-06-16 15:49:06	18.05	792.90	
4	2020-06-16 15:49:03	526.08	491.70	
5	2020-06-16 15:49:00	965.81	190.80	
6	2020-06-16 15:48:57	205.83	899.90	▼
< >				
上一页 下一页 时间条件设置 刷新				

图 5.2-21 运行效果：存盘浏览构件

注意：历史曲线构件也可以添加方法函数。

5.2.3 历史数据统计

本节主要讲解历史数据统计最大值统计方法。其他统计数据的求和、平均值、最大值、最小值、首记录纸、未记录值同理。

方案：利用报表构件显示历史数据的最大值，操作步骤如下：

- ① 新建窗口：使用章节 5.2.2.2 组态的工程，进入工程工作台的“用户窗口”，新建“窗口 1”（注意在“窗口 0”中添加标准按钮构件跳转到“窗口 1”）
- ② 绘制报表构件：双击新建的“窗口 1”进入其组态画面（可按照需要设置“窗口 1”背景色），如图 5.2-22 所示，在工具箱中选择“报表”构件，并绘制。

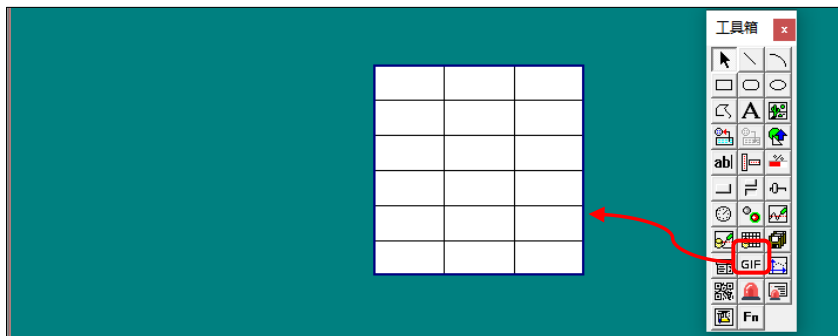


图 5.2-22 绘制报表构件

- ③ 设置列名：双击报表构件，在分别双击第 1 行当单元格，设置列名，分别输入“时间”、“温度 01_max”、“压力 01_min”，如图 5.2-23 所示。

	C1	C2	C3
R1	时间	温度01_max	压力01_min
R2			
R3			
R4			
R5			
R6			

图 5.2-23 设置列名

注意，报表构件列宽和行高均可自行调整，可参考 EXCEL 的列宽和行高调整方法。

- ④ 添加数据连接：如图 5.2-24，按住鼠标左键 → 框选第 2 行单元格 → 移动鼠标至框选的单元格 → 右键 → 点击“添加数据连接”。

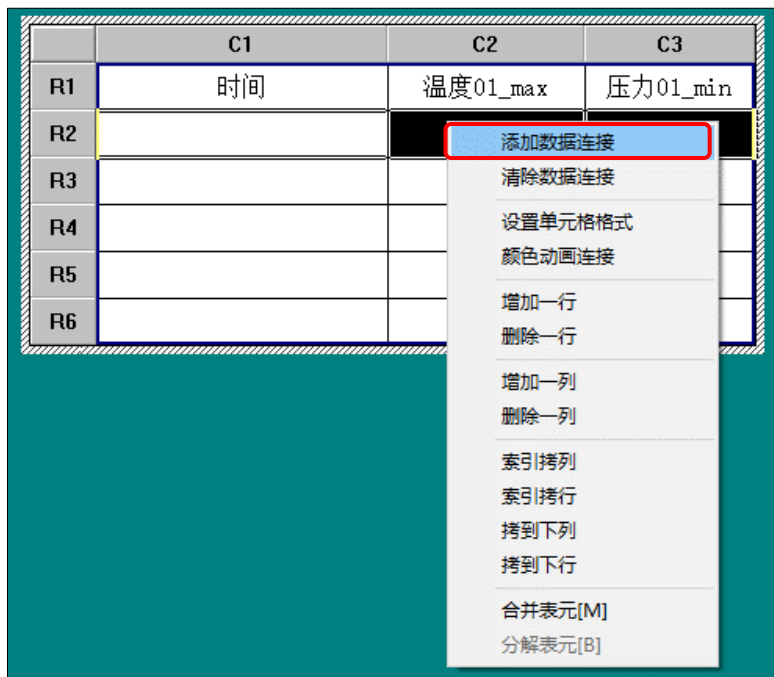


图 5.2-24 添加数据连接

- ⑤ 选择数据来源：如图 5.2-25，打开“添加数据连接”后，在“数据来源”页选择“历史数据统计”。



图 5.2-25 选择数据来源

- ⑥ 设置显示属性：如图 5.2-26，在“显示属性”页，组对象名选择“储水罐数据” → 点击“复位” → 按照需要修改“显示内容”。

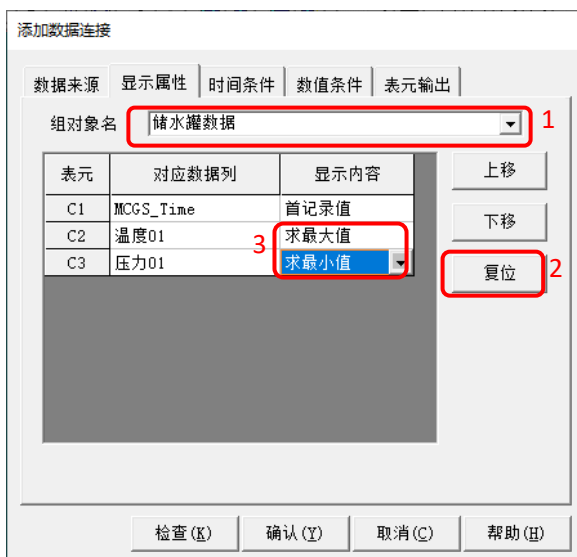


图 5.2-26 显示属性设置

“显示内容”支持：求和、平均值、最大值、最小值、首记录值、末记录值。

⑦ 其他功能页可按照需要自行设置：

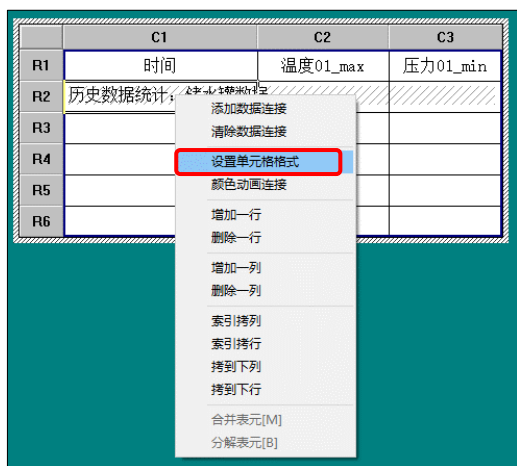
时间条件：使构件只显示指定时间段数据；

数值条件：以某列或某几列数据为筛选条件，使构件只显示满足条件的数据；

表元输出：将指定单元格的数据输出到关联的变量。

⑧ 完成以上操作后，点击“确认”。

⑨ 数据格式设置：如图 5.2-27 所示，选择单元格“R2C1”→“右键”→“设置单元格格式”→“数据格式”，显示类型选择“日期时间”，设置完成点击“确认”。



进入单元格格式设置



设置显示类型

图 5.2-27 数据格式设置

⑩ 运行效果展示：将工程下载到 TPC 或直接模拟运行，运行如图 5.2-28 所示：

历史数据统计		
时间	温度01_max	压力01_min
2020-06-17 18:18:09	983.822	0
历史数据: “温度 01” 的最大值		历史数据: “压力 01” 的最小值

图 5.2-28 工程运行效果

5.2.4 历史数据导出

工程运行时，用户可以使用“!ExportHisDataToCSV”函数，将保存的历史数据导出到 U 盘，或者触摸屏的“U 盘文件区”。将触摸屏中保存的历史数据导出到 U 盘，操作步骤如下：

方案：工程运行时，执行“!ExportHisDataToCSV”函数，将指定组对象的历史数据导出到用户 U 盘，并显示历史数据导出条数和导出状态。

- ① 新增对象：使用章节 5.2.3 工程，在“实时数据库”中新增 2 个整数对象“导出进度”、“状态控制”，如图 5.2-29。

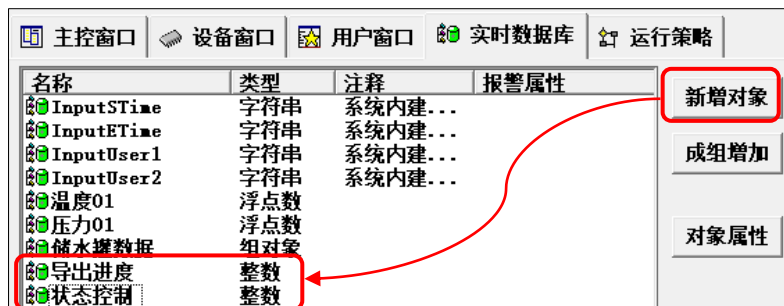


图 5.2-29 新增对象

- ② 历史导出按钮组态：进入“窗口 1”组态画面，绘制 1 个标准按钮构件，修改标准按钮构件的显示文本，并在其“脚本程序”页添加如下脚本函数：

!ExportHisDataToCSV("储水罐历史数据/历史数据.csv","储水罐数据","", "2020/6/1 0:0:0", "2025/6/1 0:0:0", 20000, 1, "", 导出进度, 状态控制)

意义：将组对象“储水罐数据”中的“温度 01”和“压力 01”的历史数据以覆盖的方式导出到“U 盘 / 储水罐历史数据 / 历史数据.csv”，导出数据时间范围为 2020/6/1 0:0:0~2025/6/1 0:0:0，最多导出 20000 条数据。

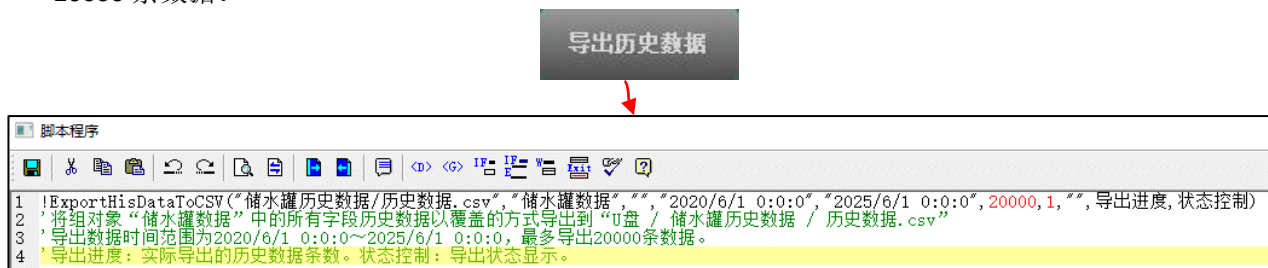


图 5.2-30 历史数据导出按钮组态

ExportHisDataToCSV 函数解释：

例：!ExportHisDataToCSV(文件名,组对象名,字段名,开始时间,结束时间,最大记录数,导出模式,导出参数,进度指示数据对象名,控制数据对象名)

- 文件名：导出文件路径，目录不存在自动创建，以“\$MCGS_DIR_USER”开头表示导出到用户分区，其它则为 U 盘
- 组对象名：字符串，导出的历史数据的组对象名字

- 字段名：字符串，用逗号分隔的要导出的字段名，如果字符串内容为空字符串“”，则导出所有字段
 - 开始时间：字符串，导出的历史数据开始时间段，格式为“YYYY-MM-DD HH:MM:SS”或“YYYY/MM/DD HH:MM:SS”
 - 结束时间：字符串，导出的历史数据结束时间段，格式为“YYYY-MM-DD HH:MM:SS”或“YYYY/MM/DD HH:MM:SS”
 - 最大记录数：整数，希望导出的历史数据条数，超过该记录数就返回，停止导出
 - 导出模式：整数，“=1”表示覆盖现有文件；“=2”表示追加到文件最后
 - 导出参数：字符串，暂时为空，保留以后使用
 - 进度指示数据对象名：必须为整数数据对象，导出过程中该对象值显示当前已导出的数据条数，如果导出过程出现异常则返回错误码，错误码含义如下：
 - = -1001, 进度或控制数据对象类型不正确
 - = -1004, 组对象名不存在或组对象不具有存盘属性
 - = -1020, 导出的开始时间大于结束时间
 - = -1021, U 盘没有插入
 - = -1022, 同一时间只允许一个导出任务
 - = -1023, 记录读取的条数为 0
 - = -1024, 文件操作失败
 - = -1025, 导出路径为空
 - = -1026, 导出路径不合法
 - = -1027, 时间格式不正确
 - = -1028, 不支持的导出模式
 - 控制数据对象名：整数数据对象,控制导出执行，启动导出时会自动设置该数据对象为 0，导出过程中想取消导出则可以设置该值为<0 的任何值，导出函数运行结束后会自动设置对象值为 1
- ③ 导出进度和状态控制显示：在“窗口 1”添加 2 个输入框，分别关联变量“导出进度”和“状态控制”，用于在工程运行时显示历史数据导出进度和状态。如图 5.2-31 所示：

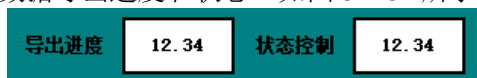
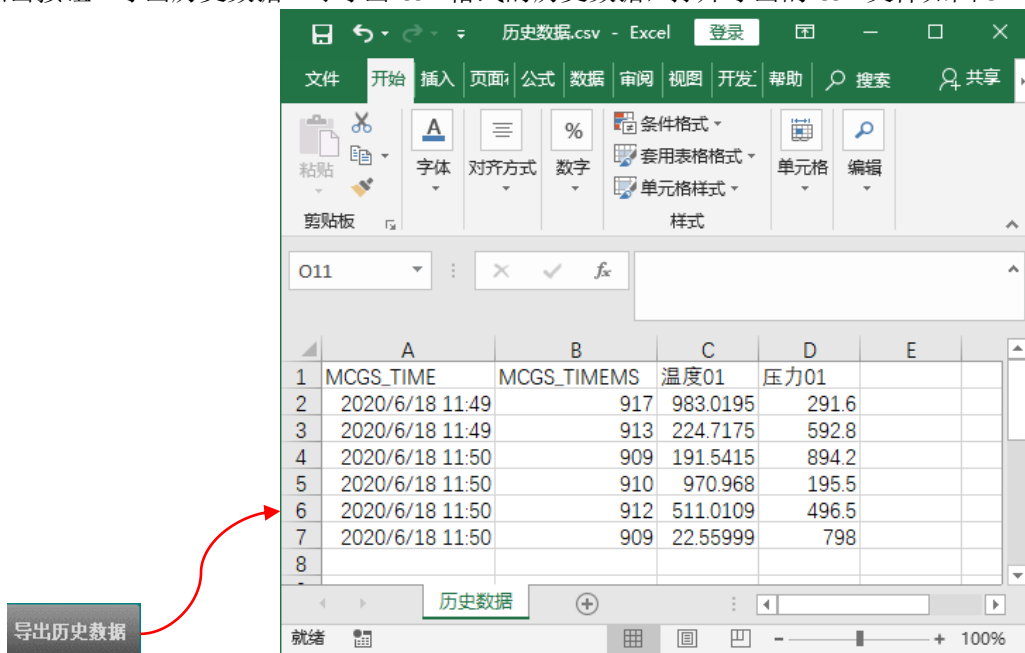


图 5.2-31 数据导出进度和状态控制

- ④ 运行工程：保证组态时设置的导出时间段正确，将工程下载到 TPC 或模拟运行，有历史数据保存情况下，点击按钮“导出历史数据”可导出 CSV 格式的历史数据，打开导出的 CSV 文件如图 5.2-32 所示：



5.2-32 导出历史数据

注意：

- 数据导出起止时间可以是变量，可在工程运行时手动指定导出时间段
- U 盘格式必须为 FAT32
- 模拟运行时导出的文件路径为：McgsPro 组态软件安装目录 \Program\export\ 用户设置的导出路径

5.2.5 历史数据删除

McgsPro 提供三种方法删除历史数据：

- 方法 1：下载工程时删除历史数据，如图 5.2-33。



图 5.2-33 下载界面删除历史数据

- 方法 2：工程运行时，磁盘剩余空间不足 10%或 5M 时，系统自动删除存储的历史数据，1 个组对象每次触发最多会删除 12.5M 历史数据。
- 方法 3：工程运行时，执行函数!DelAllSaveDat 或!DelAllSaveDat 删除历史数据。

本节教程主要讲解“方法 3”，通过函数删除历史数据，操作步骤如下：

方案：在标准按钮中组态!DelAllSaveDat 或!DelAllSaveDat 函数，工程运行时点击按钮，删除历史数据。

- ① 绘制标准按钮：使用章节 5.2.4 组态的工程，进入用户窗口“窗口 1”，绘制 1 个标准按钮构件。
- ② 添加历史数据删除函数：双击标准按钮构件，进入“脚本程序”页，输入脚本函数“!DelAllSaveDat(储水罐数据)”，如图 5.2-34 所示：

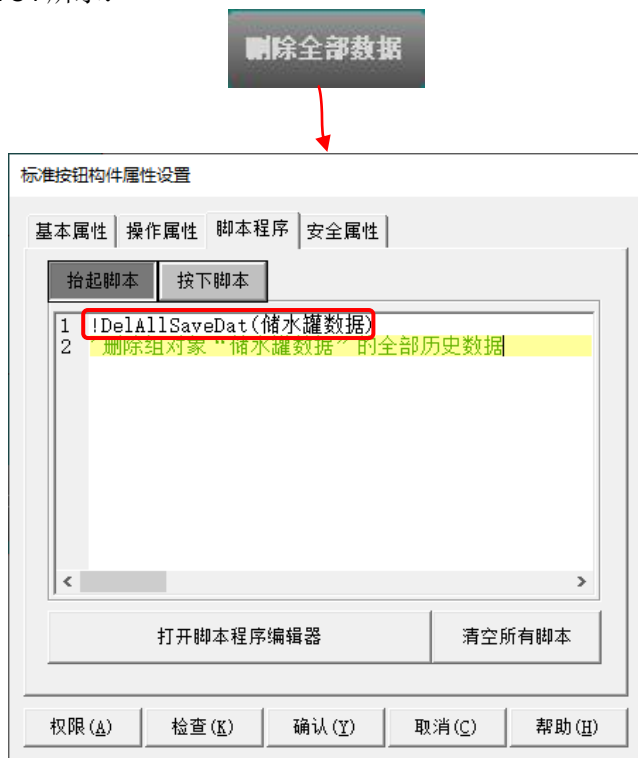


图 5.2-34 添加历史数据删除函数

- ③ 运行工程：完成上述操作后，保存工程，并下载到 TPC 或模拟运行。在有历史数据记录时，点击该按钮，可删除组对象“储水罐数据”已保存的历史数据。

函数解释：

- **!DelSaveDat(DataName,Num)**
函数意义：删除组对象 DataName 对应的存盘数据中 Num 小时之前的存盘数据
- **!DelAllSaveDat(DataName)**
函数意义：删除组对象 DataName 对应的所有存盘数据

5.3 数据处理

对于 McgsPro 组态软件，数据处理分为：数据前处理和数据后处理。通过选择合适的数据处理方式，使采集到数据满足现场用户需求。

5.3.1 数据前处理

数据前处理：指“通道处理设置”功能，在工程“设备窗口”中双击添加的驱动设备，进入其设备编辑窗口，鼠标选中想要进行通道处理的设备通道，再点击右侧的“通道处理设置”，即可进入图 5.3-1 所示的通道处理设置界面。使用“通道处理”功能，下位机地址数据在传入 McgsPro 实时数据库变量之前就完成了处理，关联该通道的“连接变量”将直接显示“通道处理”后的数据。

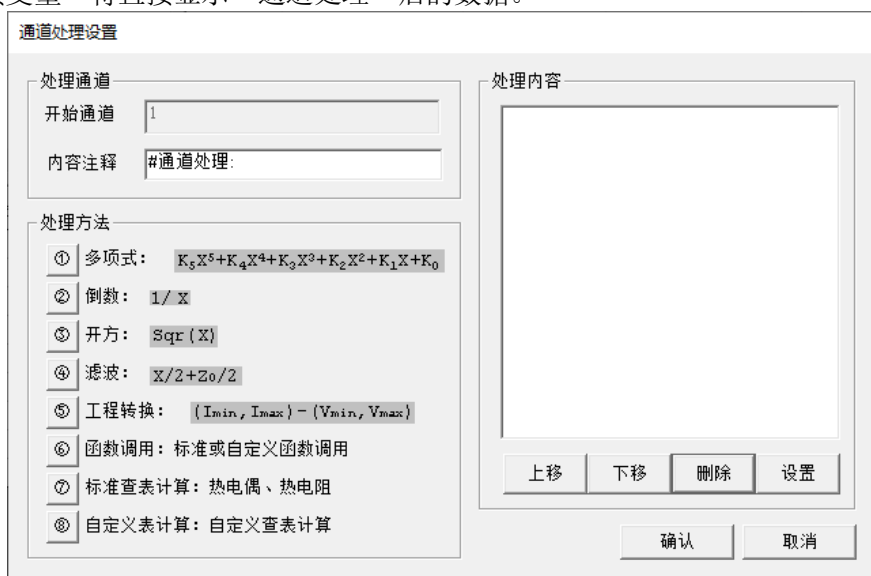


图 5.3-1 通道处理设置

McgsPro 提供 8 中通道处理方法：

- 多项式：对指定通道进行多项式运算；
- 倒数：求指定通道的倒数；
- 开方：对通道值进行开方；
- 滤波：中值滤波，本次输入信号的 1/2+上次输入信号的 1/2；
- 工程转换：对通道值进行线性（倍数）缩放；
- 函数调用：对指定的多个通道值进行统计计算，包括：求和、求平均值、求最大值、求最小值、求标准方差标；
- 标准查表计算：提供 8 中常用热电偶和 Pt100 热电阻查表计算；
- 自定义查表计算：自定义一个表，在每一行输入对应值后，指定查表基准。

注意：以上处理方法只有“工程转换”支持逆向运算，即选择了“工程转换”，从 PLC 地址中读取数据或写入数据时，都会执行“工程转换”。其他 7 中通道处理方法只在从 PLC 地址读取数据时会进行相应的运算。

本节教程以西门子 200PLC 为例，假设用户采集的 PLC 地址“V100”的数据，属于现场控制设备的压力原始信号，范围为 1000~5000mV，而压力传感器的量程为 0~2Mpa。对此，用户可选择使用通道处理中的“工程转换”对数据进行线性缩放，使 TPC 最终显示正确的压力数据。

方案：双击“设备窗口”中的西门子_200PPI 驱动，找到地址 V100，进入“通道处理”设置界面，选择“工程转换”，设置输入最大值、最小值，以及工程最大值、最小值。

- ① 进入工程的设备窗口画面，双击  进入设备窗口，再双击驱动设备“西门子_200PPI”进入设备

编辑窗口。

- ② 如图 5.3-2 所示：首先在设备窗口编辑窗口选中需要进行通道处理的通道“读写 VWUB100”→“通道处理设置”→“工程转换”→“工程量转换”，按照需要，设置输入最小值为 1000，输入最大值为 5000，工程最小值为 0，工程最大值为 2，设置完成依次点击“确认”保存。

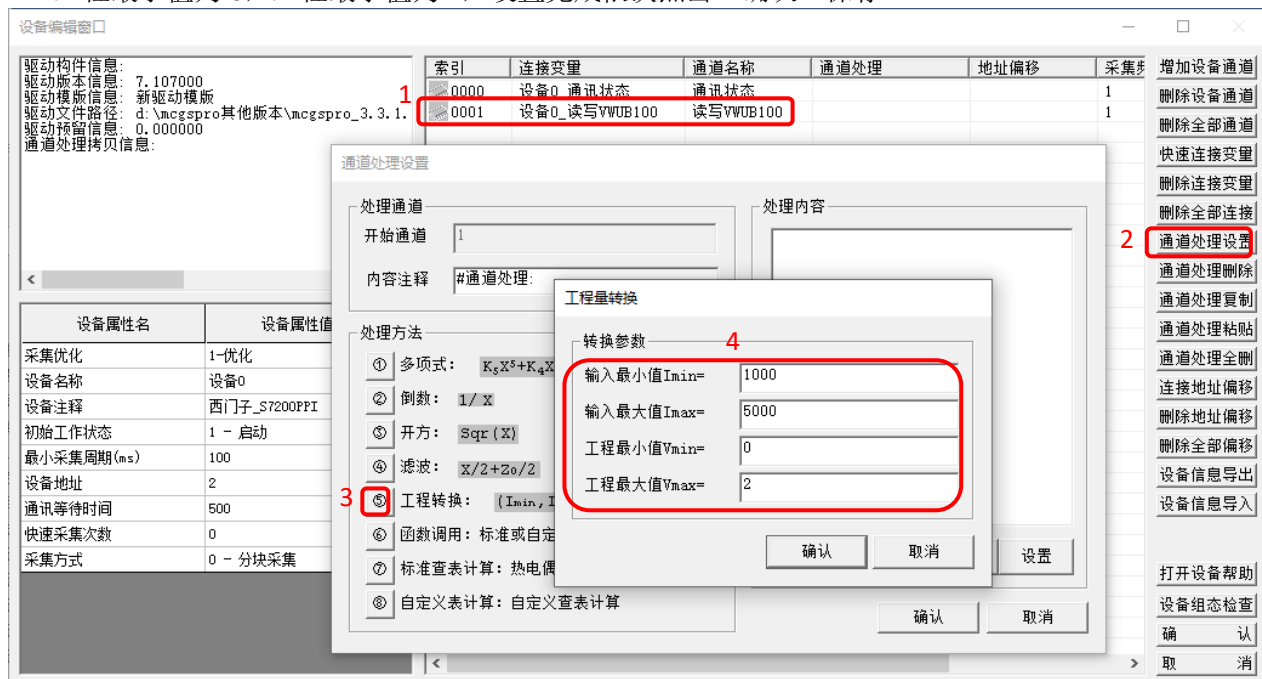


图 5.3-2 工程转换设置

- ③ 完成步骤 2 操作后，在用户窗口中添加 1 个输入框，关联图 5.3-2 所示的连接变量“设备 0_读写 VWUB100”，保存工程，连接触摸屏和 PLC，并将工程下载到 TPC 中运行，效果如下：
- 通道“读写 VWUB100”的值小于 1000，变量“设备 0_读写 VWUB100”值为 0；
 - 通道“读写 VWUB100”的值大于 1000，小于 5000，变量“设备 0_读写 VWUB100”值对应 0~2；
 - 通道“读写 VWUB100”的值大于 5000，变量“设备 0_读写 VWUB100”值为 2。

5.3.2 数据后处理

数据后处理：本质上是对工程“实时数据库”中变量数据的处理，本章节介绍的实时趋势曲线、历史数据功能（包括历史数据存盘、显示、数据统计、导出、删除）均属于数据后处理的范畴。关于数据后处理中统计数据的处理，用户可参考进阶篇章 5.2.3。

5.4 变量初值保存功能

一般来说，工程启动时，实时数据库中数据对象的初始值是组态时设置的初始值。若想重启后数据对象的值恢复为重启前的值，就需要将数据对象的值先保存起来，以备重启时调用。变量的初值保存有两种方式，一种是设置初值属性，另一种是使用初值存盘函数!SaveSingleDataInit()。不论用哪种方式保存初值，都会在 60 秒后才会刷盘将初值存入磁盘，除非保存初值后马上调用函数!FlushDataInitValueToDisk()，把所有需要保存初值的数据对象的设定初值立即写入磁盘。

5.4.1 设置初值属性

当对象类型是【整数】【浮点数】或【字符串】时，可以选择数据对象的【基本属性】功能页中的【变化时自动保存初值】。勾选此复选框后，当数据对象的值发生变化时，首先自动保存初值到缓存，等待 60 秒后，系统再将初值刷入磁盘。若还没有到 60 秒屏幕就意外掉电了，将丢失 60 以内修改的初值。

- ① 进入实时数据库，新建一个数据对象【A】。将它的【数据类型】设为【浮点数】，勾选【变化时自动保存初值】前的复选框。点击确认保存设置并关闭设置界面。如图 5.4-1 所示。



图 5.4-1 数据对象 A 基本属性设置



图 5.4-2 数据对象 B 基本属性设置

- ② 进入窗口 0，在一个【输入框】构件的【操作属性】功能页中关联变量【A】。下载工程并运行后，将【输入框】的值修改为 3。等待 60 秒后，将屏幕断电重启，可以看到输入框中的值依然是 3，保存了刚才修改的初值。

5.4.2 初值存盘函数

函数!SaveSingleDataInit(DataName)的作用是把数据对象的当前值设置为初始值，以便 McgsPro 组态软件下次启动时这些数据对象能自动恢复其值。

- ① 进入实时数据库，新建一个数据对象【B】。将它的【数据类型】设为【整数】。点击确认保存设置并关闭设置界面。如图 5.4-2 所示。
- ② 在窗口 0 中新建一个【输入框】构件，在其【操作属性】功能页中关联变量【B】。
- ③ 然后新建一个标准按钮构件。如图 5.4-3 所示将按钮的【基本属性】中的文本设置为【保存初值】，【脚本程序】中的抬起脚本设置为【!SaveSingleDataInit(B)】。按钮的功能是，当按钮抬起时，执行脚本【!SaveSingleDataInit(B)】，即将【B】的初值存入缓存，等待 60 秒后，初值被刷入磁盘中。此时，若立即执行脚本【!FlushDataInitValueToDisk()】，则不必等待 60 秒，初值将被立即刷入磁盘。

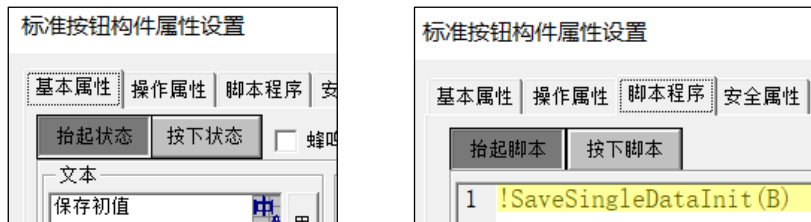


图 5.4-3 按钮属性设置

第6章 多重复制应用实例

在组态画面过程中，经常需要快速创建一批同质不同地址(变量)的动画构件，此时可使用多重复制功能。多重复制构件可配置项目有普通文本、多语言文本、变量名称、表达式。

- 多语言文本：属性值使用了多语言项内容，自动创建或关联对应的多语言项内容。没使用多语言项内容，依旧保持不使用多语言项内容。

- 变量名称：变量名称不存在时，在实时数据库中自动创建对应变量。

- 表达式：表达式中存在未知变量时，在实时数据库中自动创建对应变量。

多重复制功能仅针对变化项目配置中已勾选的项目进行内容变化分发，其他未勾选的，保持跟基础构件内容一致。

6.1 单一控件的多重复制

使用多重复制组态一个包含 100 个电机状态指示灯报警界面。

- 首先使用多重复制功能复制出 100 个【标签】构件作为电机序号标注，使其样式保持统一，文字显示为【1#电机】 - 【100#电机】。
- ① 将一个【标签】构件拖拽到画面中合适的位置和合适的大小。双击该构件，弹出【标签动画组态属性设置】界面，在【属性设置】功能页中，设置其【填充颜色】为【没有填充】，【边线颜色】为【没有边线】。在【扩展属性】功能页中的【文本内容输入】框中，输入【1#电机】。点击确认保存设置并关闭设置界面。



图 6.1-1 标签构件动画组态属性设置界面

- ② 将【标签】构件拖拽到画面的左上角，为接下来的多重复制留下足够的空间。
- ③ 选择【标签】构件，点击工具栏中的【多重复制】图形按钮，弹出【多重复制构件】设置界面。
- ④ 因为要组态 100 组电机指示灯，所以将【水平方向个数】和【垂直方向个数】各设置为 10。
- ⑤ 然后在【水平间隔像素】和【垂直间隔像素】中输入数值，点击下方的【预览】按钮，观察多重复制后矩阵的排列效果。注意需要预留出指示灯的位置。
- ⑥ 勾选【变化项目配置】列中的【显示文本[标签][控件 0]】前的复选框，这样右侧【详细配置】中的修改才会生效。
- ⑦ 在【自动递增】中自动生成的【内容】中，{R}为可递增变量，默认出现在文字最后，将它修改到需要的地方，如图 6.1-2 所示。起始值修改为【1】。修改完后，下方的【列表索引】中的预览内容也会随之改变。点击确定保存设置并关闭设置界面。

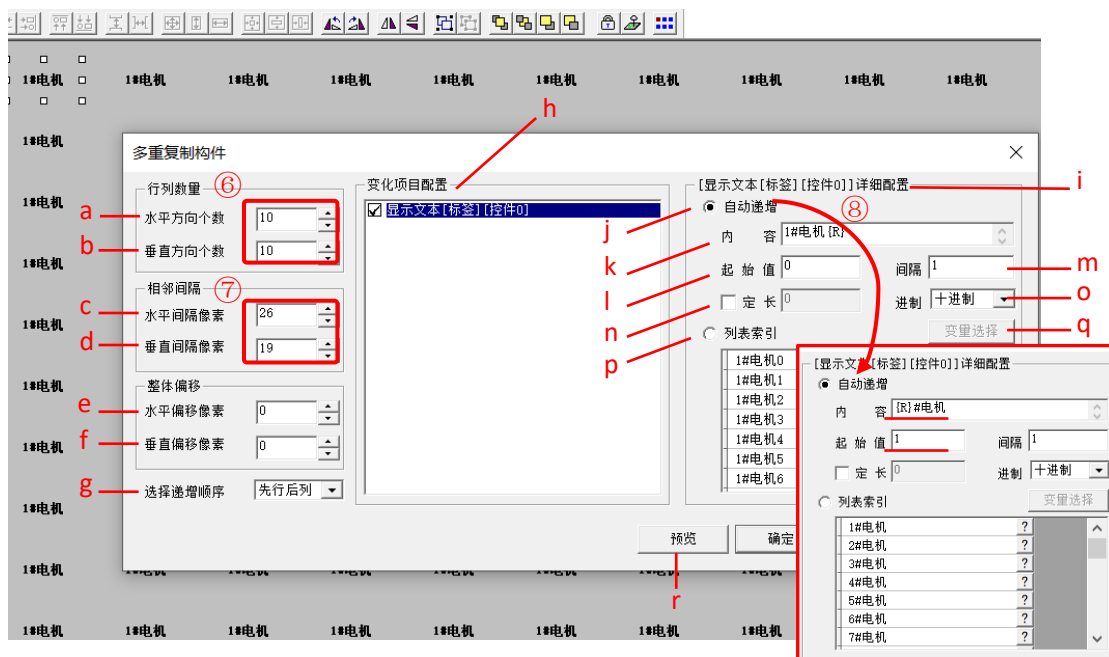


图 6.1-2 多重复制构件界面

多重复制配置信息

- a. 水平方向个数：构件矩阵在水平方向上的个数[1,30]。
- b. 垂直方向个数：构件矩阵在垂直方向上的个数[1,30]。
- c. 水平间隔像素：构件矩阵在水平方向上的间隔像素[0,500]。
- d. 垂直间隔像素：构件矩阵在垂直方向上的间隔像素[0,500]。
- e. 水平偏移像素：构件矩阵在水平方向上的整体偏移像素[-500,500]。
- f. 垂直偏移像素：构件矩阵在垂直方向上的整体偏移像素[-500,500]。
- g. 选择递增顺序：先行后列(优先水平方向分配)、先列后行（优先垂直方向分配）。
- h. 变化项目配置：构件属性的普通文本、多语言文本、变量、表达式。
勾选代表矩阵构件对应属性内容按照预定义规则进行规律变化。
不勾选代表矩阵构件对应属性内容保持不变。

所有的变化项目后方的中括号中标有所属控件的类型和控件名,如:显示文本[标签][控件0]。

- i. 详细配置：配置某一项具体变化项目，包含自动递增、列表索引两种方式。
- j. 自动递增：按照自动递增的一些预置规则进行内容变化。（列表索引下的表格可预览配置对应的矩阵内容）。
- k. 内容：基础构件对应属性下的值，{R}这三个字符代表变化部分。
- l. 起始值：配置{R}部分的起始数字，默认为 0，原值末尾是数字时，起始值默认为该数字。例如，“文 3”起始值就是 3。
- m. 间隔：配置{R}部分的数字间隔，默认为 1。
- n. 定长：配置{R}部分的数字整体长度，默认不勾选，原值末尾是数字时，定长长度为该数字长度。例如，“文 0003”定长就是 4。
- o. 进制：配置{R}部分的变化，起始值、间隔、定长按照八进制/十进制/十六进制进行规律变化。默认为十进制。
- p. 列表索引：开启列表索引下的表格内容编辑。
- q. 变量选择：在列表索引选中时，可对表格内容做整体替换。（多选）
- r. 预览：预览构件矩阵的个数、间隔像素、整体偏移像素。

⑧ 点击确定，关闭设置界面后，画面如图 6.1-3 所示。

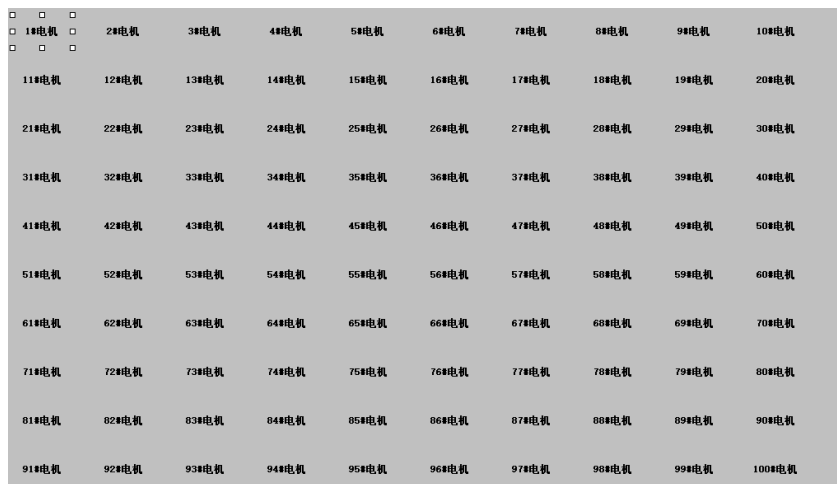


图 6.1-3 标签构件多重复制效果

- 然后使用多重复制功能复制出 100 个【椭圆】图元，作为电机报警状态指示灯。
- ① 在实时数据库中新建一个浮点数变量【电机 1 报警状态】。注意，变量名中不能出现【#】字符。
- ② 将一个【椭圆】拖拽到画面中【1#电机】标签构件的右侧，拖拽成一个正圆。也可以直接在右下角的长度、宽度输入框中输入相同的数值使之成为一个正圆。

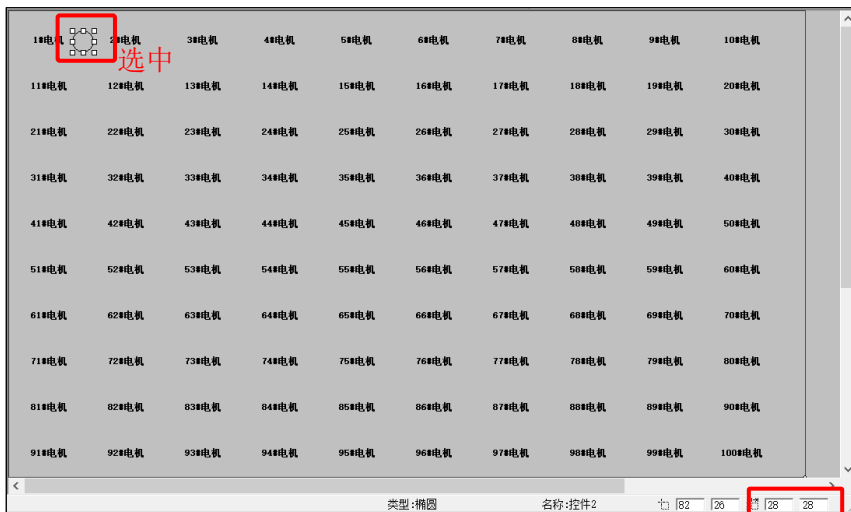


图 6.1-4 将椭圆设置成正圆

输入长、宽值

- ③ 双击【椭圆】构件，弹出【动画组态属性设置】界面。勾选【填充颜色】前的复选框，在出现的【填充颜色】功能页中的表达式处关联变量【电机 1 报警状态】。下方的【填充颜色连接】保持默认不变。表示当表达式为 1 时，填充颜色红色，为 0 时填充颜色为绿色。点击确认保存设置并关闭设置界面。



图 6.1-5 椭圆图元组态

- ④ 接着选中【椭圆】构件，点击工具栏中的【多重复制】图形按钮，弹出【多重复制构件】设置界面。
- ⑤ 同样的，在【水平方向个数】和【垂直方向个数】处输入 10，然后在【水平间隔像素】和【垂直间隔像素】处输入数值，点击下方的【预览】按钮，使所有的【椭圆】构件都在之前复制的【标签】构件的右侧方。
- ⑥ 勾选【变化项目配置】列下方的【表达式[椭圆][控件 2]】前的复选框。
- ⑦ 修改【自动递增】中的【内容】为【电机{R}报警状态】，【起始值】为【1】。下方【列表索引】中也会出现相应的预览变化。



图 6.1-6 对椭圆图元进行多重复制

- ⑧ 点击确定保存设置并关闭设置界面。此时，由于我们没有提前创建变量，系统会弹出提示框询问是否添加，选择【确定】，在实时数据库中自动添加变量。

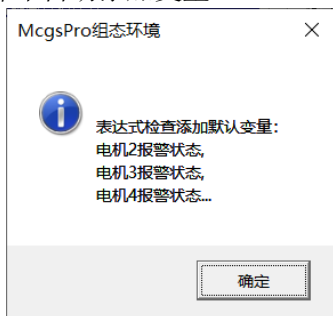


图 6.1-7 提示添加未定义变量

- ⑨ 关闭设置界面后，画面如图 6.1-8 所示。

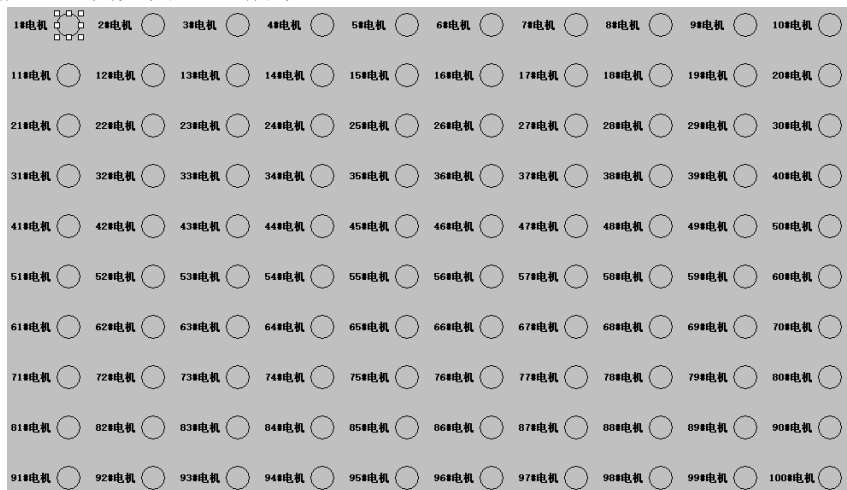


图 6.1-8 椭圆图元多重复制效果

- ⑩ 此时，所有的标签构件右侧的椭圆中关联的表达式都和标签构件中电机的编号一一对应。任意双击打开一个椭圆构件查看效果。



图 6.1-9 检查多重复制结果

- ⑪ 此时的实时数据库如图 6.1-10 所示。

名称	类型	注释	报警属性
电机1报警状态	浮点数		
电机2报警状态	浮点数		
电机3报警状态	浮点数		
电机4报警状态	浮点数		
电机5报警状态	浮点数		
电机6报警状态	浮点数		
电机7报警状态	浮点数		
电机8报警状态	浮点数		
电机9报警状态	浮点数		
电机10报警状态	浮点数		
电机11报警状态	浮点数		
电机12报警状态	浮点数		
电机13报警状态	浮点数		
电机14报警状态	浮点数		
电机15报警状态	浮点数		
电机16报警状态	浮点数		
电机17报警状态	浮点数		
电机18报警状态	浮点数		
电机19报警状态	浮点数		
电机20报警状态	浮点数		
电机21报警状态	浮点数		

图 6.1-10 实时数据库

- 组态完画面和实时数据库变量后，还需将实时数据库中变量与 PLC 通道相连。本样例以 Smart200 为例。

- ① 双击【Smart200】驱动，进入【设备编辑窗口】。点击【删除全部通道】删除默认通道。再点击【添加设备通道】按钮 → 选择通道类型为【M 内部继电器】 → 设置通道个数为【100】。

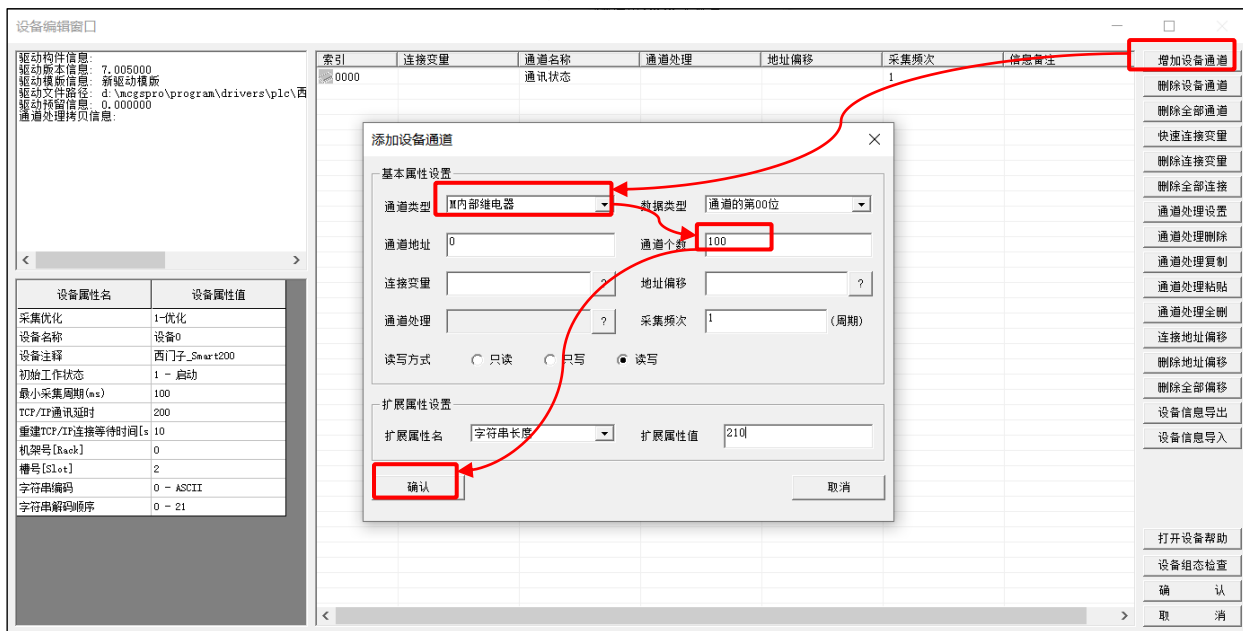


图 6.1-11 设备编辑窗口

- ② 然后在第一个通道的【连接变量】列双击 → 在弹出的【变量选择】界面中按住 Shift 选择从【电机 1 报警状态】到【电机 100 报警状态】 → 确认。

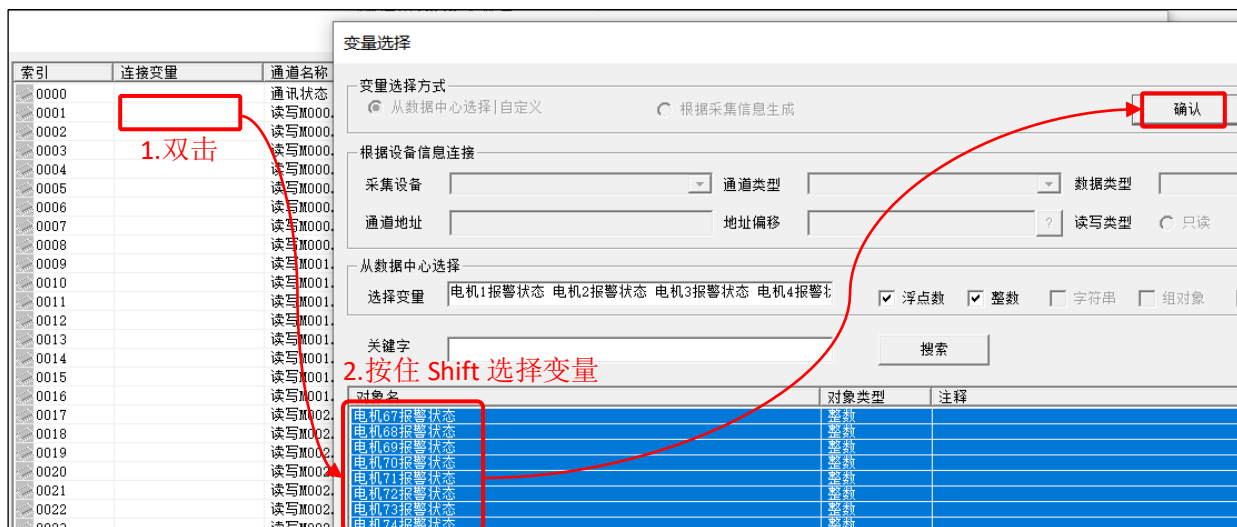


图 6.1-12 连接变量

- ③ 连接变量完成后，【设备编辑窗口】如图 6.1-13 所示。点击【确认】保存设置并关闭设置界面。

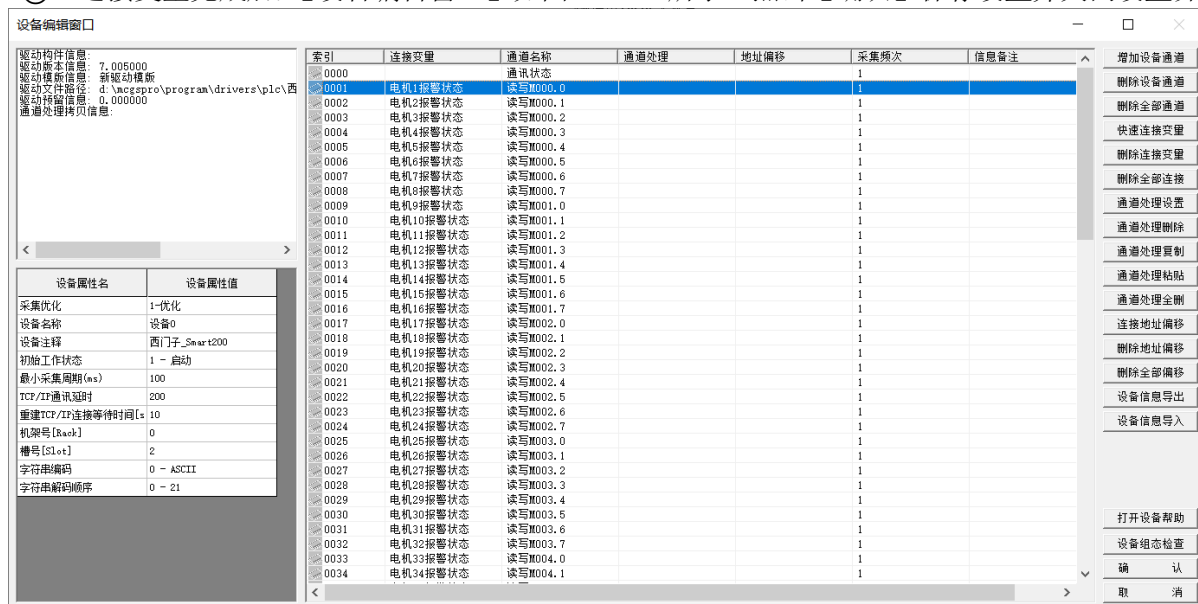


图 6.1-13 连接变量结果

6.2 单元的多重复制

在 McsgPro 中也可以将几个控件组合成单元后再统一多重复制，这样就节约了控件一个个单独排版的时间。本节使用【列表索引】方式来组态。

- 在实时数据库中新建 20 个浮点数值变量【电流 1】至【电流 20】，20 个整数型变量【启停 1】至【启停 20】。

- 进入实时数据库 → 成组增加，设置【对象名称】为【电流】，【对象类型】为【浮点数】，【起始索引】为【1】，【添加个数】为【20】个。点击确认保存设置并关闭设置界面。如图 6.2-1 所示。

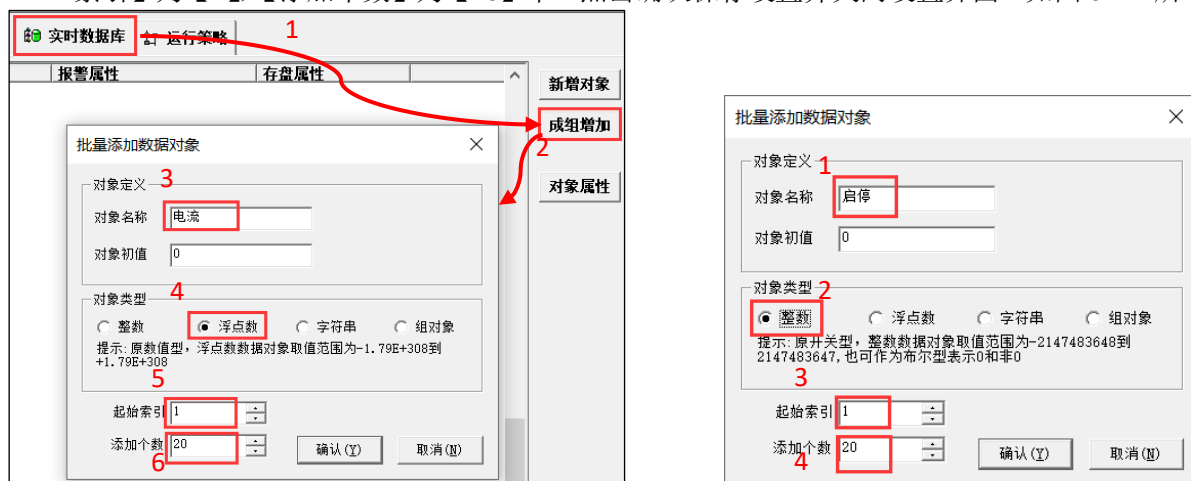


图 6.2-1 成组增加变量

- 用同样的方法新建 20 个整数型变量【启停 1】至【启停 20】。点击【成组增加】，设置【对象类型】为【整数】，【起始索引】为【1】，【添加个数】为【20】个。点击确认保存设置并关闭设置界面。如图 6.2-1 所示。
- 设置完成后实时数据库如图 6.2-2 所示：

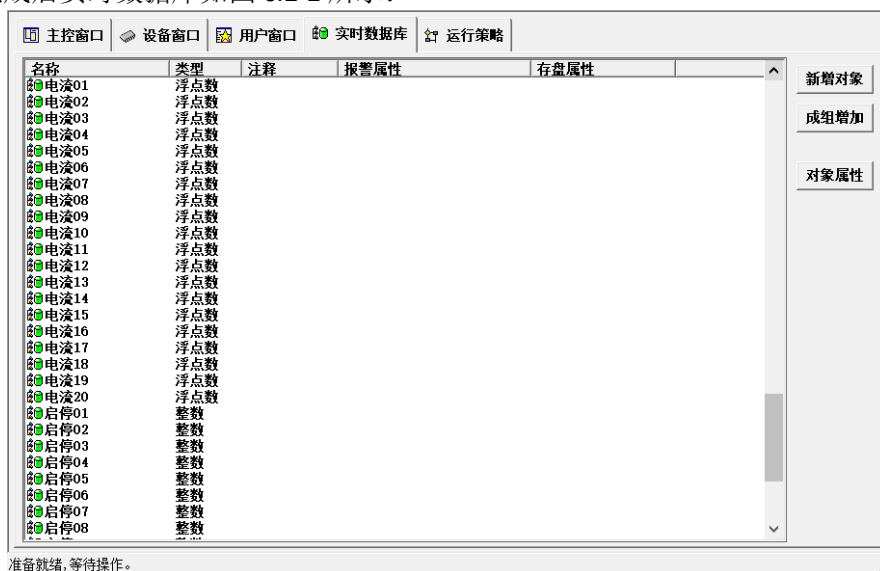


图 6.2-2 实时数据库

- 组态一个画面，显示 1#至 20#电机的电流，并能在画面中控制所有电机的启动和停止。

- 组态如图 6.2-3 所示的画面，我们将对红线框出来的 4 个组态内容进行多重复制。其他内容，在多重复制时，不在【变化项目配置】中勾选，将保持不变。

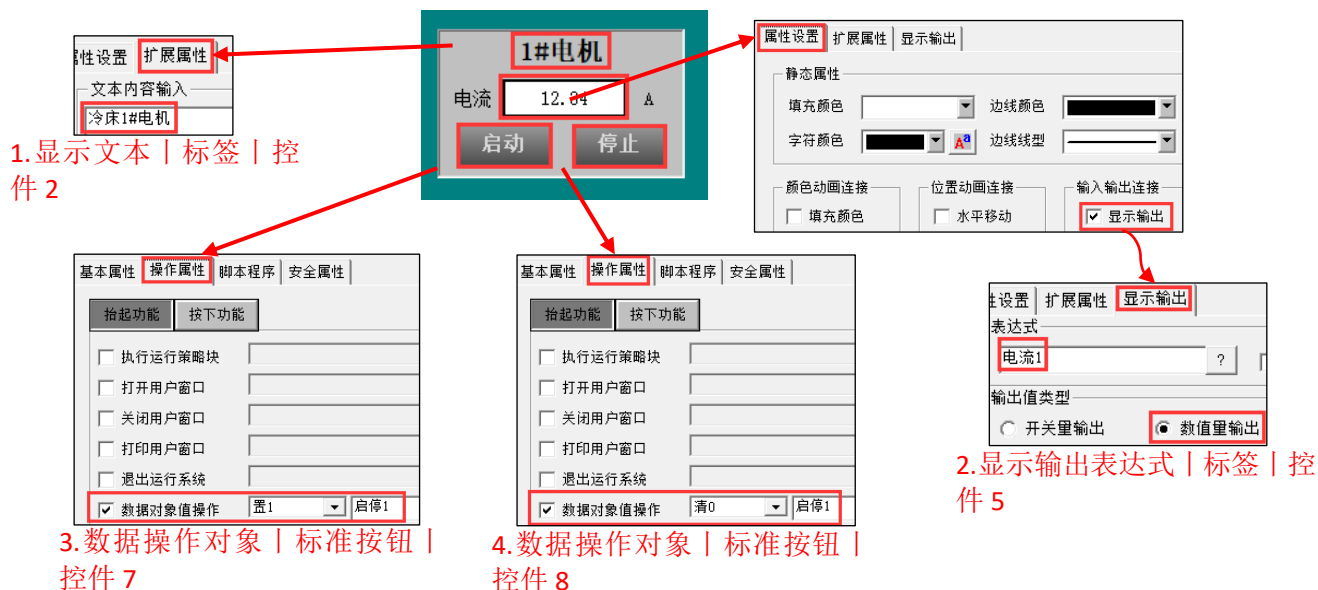


图 6.2-3 单元中需要多重复制的项

② 选中所有控件后，在控件上点击鼠标右键 → 排列 → 合成单元。



图 6.2-4 合成单元

- ③ 选中合成后的单元，再点击工具栏中的【多重复制】图形按钮，弹出【多重复制构件】设置界面。
- ④ 将【水平方向个数】设置为【5】，【垂直方向个数】设置为【4】。然后结合下方的【预览】按钮，调节【水平间隔像素】和【垂直间隔像素】的值，使 20 个单元在画面中均匀分布。
- ⑤ 勾选【变化项目配置】列表中【显示文本[标签][控件 2]】前的复选框，按图 6.2-5 进行设置。



图 6.2-5 多重复制单元流程一

- ⑥ 再勾选【显示输出表达式[标签][控件 5]】前的复选框 → 列表索引 → 变量选择，弹出【变量选择】窗口。按住 Shift 键后选中【电流 1】至【电流 20】，点击【确认】关闭窗口。

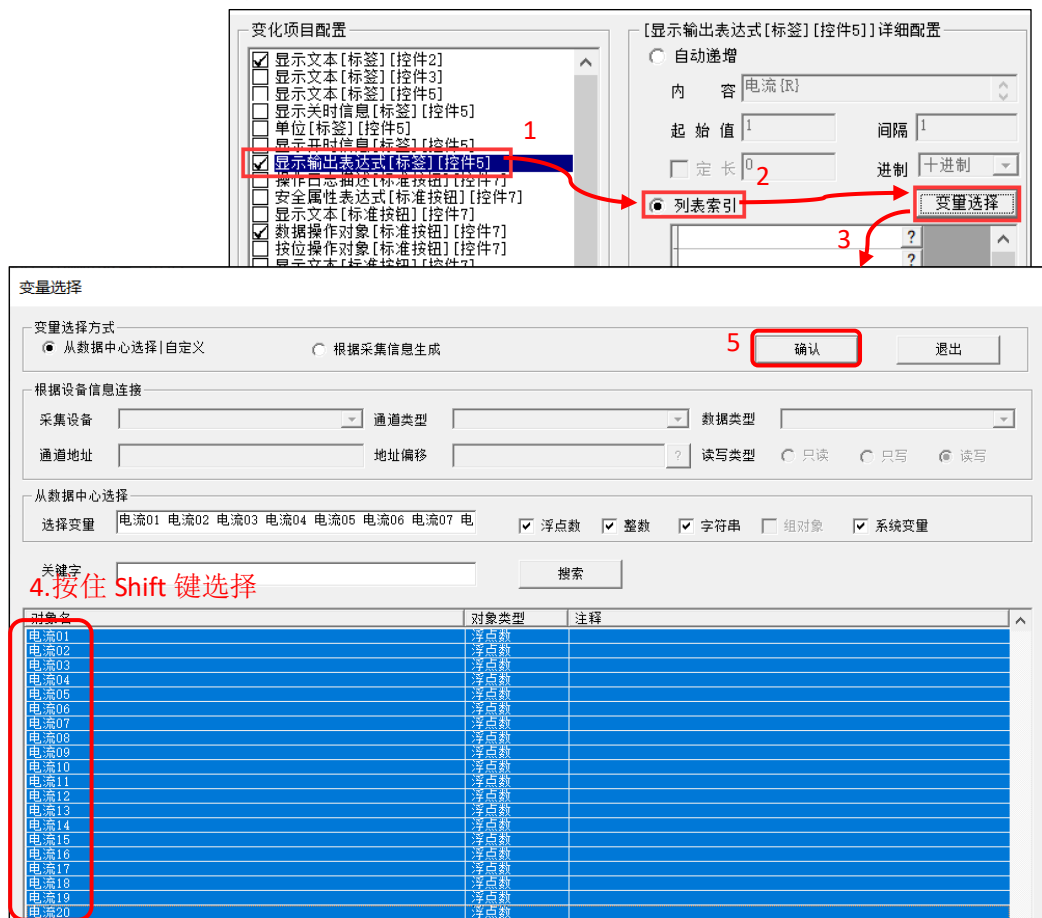


图 6.2-6 多重复制单元流程二

- ⑦ 用同样的方法配置剩下的两个变化项目。配置完成后如图 6.2-7 所示。

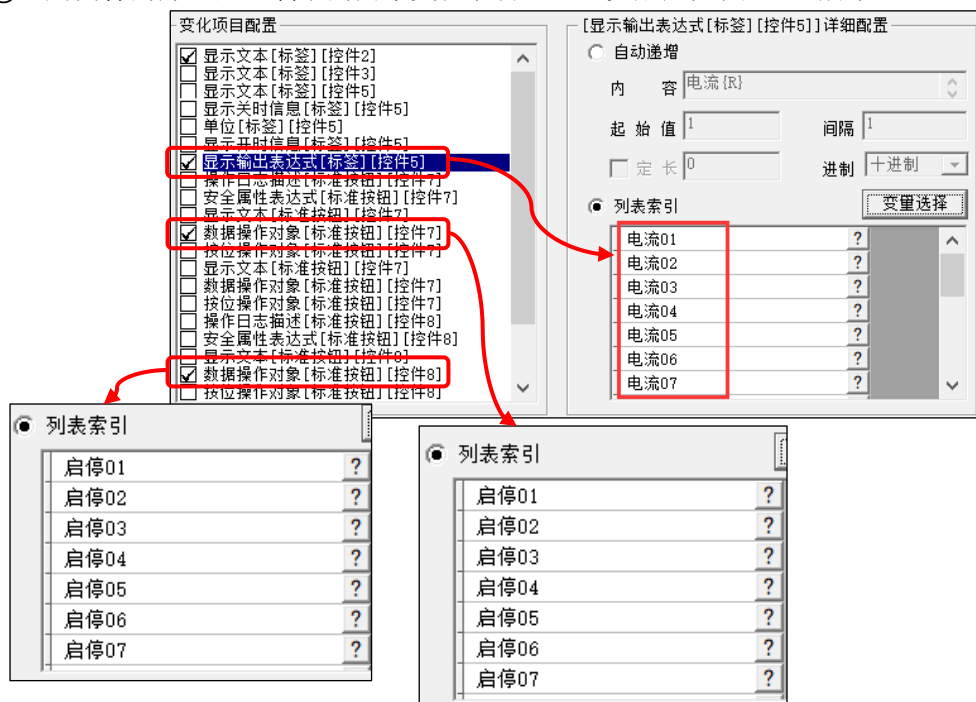


图 6.2-7 多重复制单元流程三

⑧ 点击确认保存设置并关闭设置界面。设置完成后的画面如图 6.2-8 所示：



图 6.2-8 多重复制单元效果

- 然后再按照 6.1 的方法创建通道并将变量与通道相连。

在实际使用中，还会出现窗口内容相同，变量地址有规律递增的情况。此时可以通过多重复制功能，先将所有窗口内容在同一个窗口中复制出来，直接关联好新窗口中的变量。再将各个新窗口的内容，剪切到对应的窗口中去。

第 7 章 指针功能应用实例

McgsPro 的指针有两种，一种是设备通道里的地址偏移，一种是实时数据库中的变量指针。

7.1 地址偏移应用样例

地址偏移的作用是：通过设备驱动的地址偏移，让设备通道关联的【连接变量】采集偏移后 PLC 地址的值。

偏移后地址计算公式：偏移后地址=当前通道地址+地址偏移数值。地址偏移数值可为负数，偏移后的地址数值最小值为 0。

本节样例使用西门子 SMART200 PLC 与 McgsPro 进行通讯。假设在 PLC 的 VW90 中的值为 900，VW100 中的值为 1000，VW110 中的值为 1100。在 McgsPro 中组态变量【DATA1】连接 PLC 的 VW100，然后通过设置【偏移量】，使画面可以显示 PLC 的 VW90 和 VW110 中的值。

- ① 在实时数据库中新建两个整数型变量【DATA1】和【偏移量】。
- ② 双击西门子 SMART200 驱动，打开【设备编辑窗口】，删除所有默认通道。点击【添加设备通道】按钮，弹出【添加设备通道】界面。如图 7.1-1 所示，设置通道类型为【V 数据寄存器】，数据类型为【16 位 无符号二进制】，通道地址为【100】，连接变量为【DATA1】，地址偏移为变量【偏移量】，其他设置保存默认不变。点击确认。

图 7.1-1 地址偏移设置

地址偏移设置方法

- a. 选中通道表格中一个或多个通道，点击“连接地址偏移”按钮打开设备通道窗口设置地址偏移。多个通道可共用同一地址偏移，也可使用不同的地址偏移。
- b. 选中通道表格中一个通道，在地址偏移列双击左键或者单击右键，打开通道连接变量选择窗口进行变量选择。
- c. 选中通道表格中一个通道，在通道名称列双击左键或者单击右键，打开设备通道窗口设置地址偏移。

- ③ 设置完成后【设备编辑窗口】如图 7.1-2 所示

索引	连接变量	通道名称	通道处理	地址偏移
0000		通讯状态		
0001	DATA1	读写VWUB100		偏移量

图 7.1-2 添加地址偏移完成

- ④ 进入用户【窗口 0】，组态一个【输入框】构件，在它的【操作属性】功能页中关联变量【偏移量】。组态一个【标签】构件，勾选它的【显示输出】前的复选框，然后在【显示输出】功能页中关联变量【DATA1】，并选择其【输出值类型】为【数值量输出】。
- ⑤ 保存工程后，下载工程，然后启动运行。当变量【偏移量】=0 时，变量【DATA1】的值为字地址 VW100

的值 1000。用【输入框】设置【偏移量】=-10，则变量【DATA1】的值变为 VW90 的值 900。同样的，用【输入框】设置【偏移量】=10，则变量【DATA1】的值变为 VW110 的值 1100。如图 7.1-3 所示。

偏移地址:	0	偏移地址:	-10	偏移地址:	10
连接变量DATA1:	1000	连接变量DATA1:	900	连接变量DATA1:	1100

图 7.1-3 地址偏移功能运行效果

7.2 变量指针应用样例

有的项目需要显示多路相同设备的状态，但限于触摸屏大小不能在一个画面中全部显示。这种情况下，一般是使用多个画面来显示多路设备的状态，但通过数据对象指针化的方式可以减少画面数量，实现在一个画面中即可显示全部设备状态的效果。

比如我们需要显示 100 路设备的温度，但在一个画面中不能全部显示，在没有指针化功能时，只能通过多个画面切换来观察 100 路设备的温度。而通过指针化功能，我们可以将 100 路设备的温度通过一个【指针数据对象（温度指针）】和【一个索引数据对象（温度索引）】关联在一起，然后通过调整温度索引的值，使画面中的温度指针发生对应的变化，即可观察所有设备的温度值。为了方便理解，本节样例只演示 6 个设备的温度通过指针化功能实现在屏幕上显示。使用指针化功能显示更多设备的温度的组态方式以此类推。

除数据组对象外，其它三种数据对象均可以配置指针化属性。当数据对象配置为指针化时，其属性将只能配置指针化属性，其它诸如报警、存储等属性都将不能配置。指针化数据对象的使用与普通数据对象相同，但在运行时会受到索引数据对象的值的束缚。

指针化配置有以下两种方式：

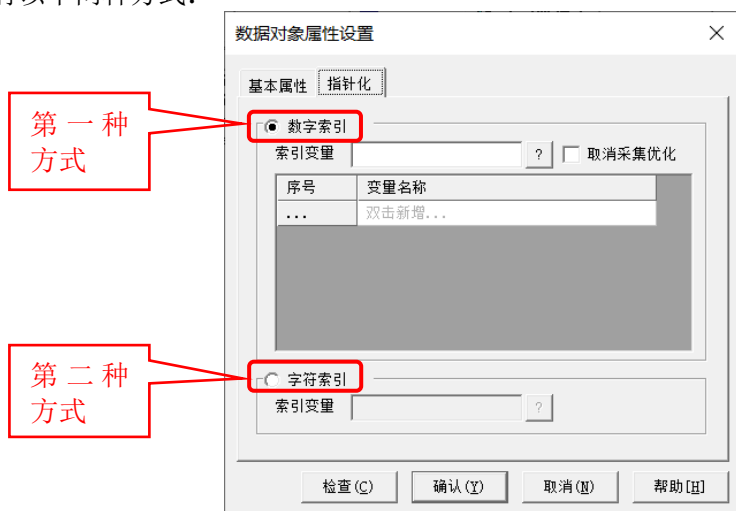


图 7.2-1 数据对象-指针化功能页

- 第一种是数字索引，即索引列表方式：

- ① 在实时数据库中新建 8 个整数变量，名称依次为【温度指针】【温度索引】【设备 0 温度】至【设备 5 温度】。双击【温度指针】，勾选【设置指针化】，在【索引变量】处关联变量【温度索引】。然后双击下方列表的空白行增加一行，在列表中依次关联变量【设备 0 温度】至【设备 5 温度】。当需要关联的变量比较多时，可以使用右键的【导出数据】功能将数据导出为 xml 格式后，用 Excel 打开修改并再次保存为 xml 格式后使用右键的【导入数据】功能导入。



图 7.2-2 数字索引指针变量-设置流程图

- ② 设置完成后，当【温度索引】=【0】时，【温度指针】=序号【0】对应的变量【设备0温度】，以此类推。

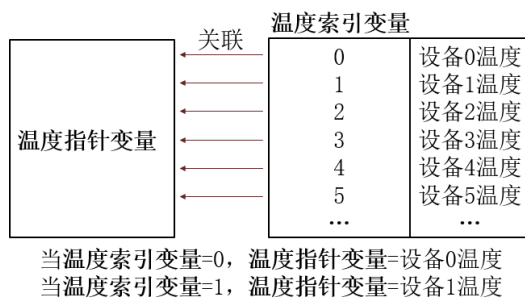


图 7.2-3 数字索引指针变量-效果

注意：数字索引变量的数据类型只能是整数或浮点数。当索引数据对象的字符内容指定的数据对象不存在或不正确时，指针数据对象操作将无效。

- 第二种是字符索引，即使用数据对象名索引的方式：



图 7.2-4 字符索引指针变量-设置流程图

相当于，字符串类型的【温度索引】变量=【设备0温度】的名字后，【温度指针】变量=【设备0温度】的值。

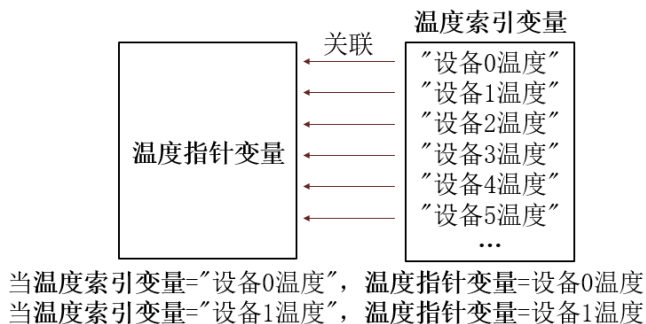


图 7.2-5 字符索引指针变量-效果

注意：字符串索引变量的数据类型只能是字符串。当索引数据对象的字符内容指定的数据对象不存在或不正确时，指针数据对象操作将无效。

第 8 章 配方功能应用实例

在制造领域，配方是用来描述生产一件产品所用的不同配料之间的比例关系，是生产过程中一些变量对应的参数设定值的集合。例如面包厂生产面包时有一个配料配方。此配方列出所有要用来生产面包的配料（如水、面粉、糖、盐、蜂蜜等），而不同口味的面包会有不同的配料用量。例如甜面包会使用更多的糖，而低糖面包则使用更少的糖。在组态软件中，每一种口味的面包原料的用量信息称为一个配方，所有面包配方合起来称为配方组。可以把配方组想象成一张表格，表格的每一列就是一种原料，而每一行就是一个配方，单元格的数据则是每种原料的具体用量。

配方	糖	盐	面粉	水	蜂蜜
甜面包配方	80	10	80	30	10
低糖面包配方	30	5	80	30	0
无糖面包配方	10	5	80	30	0

图 8-1 配方表格

在 McgsPro 系列产品配方功能可通过配方选择对话框，配方编辑对话框以及各种脚本操作来完成对配方数据的增、删、改、查等操作。下面对配方组态中涉及到的名词进行说明：

- 配方组：配方组即同类配方所组成的集合，这一类配方拥有相同的配方成分，不同的是配方成分的数量不同，在配方组中可以使用配方名称来区分这些不同的配方。上表整体就为一个配方组。

- 配方项：配方项就是配方的配方成分，如图 8-1 中糖、盐等。上表每一列即一个配方项。

- 配方：配方组中的数据成员，每个配方的配方成分一致，数量不同。上表每一行就表示一个配方。

- 变量名称：配方项关联的变量名称。

- 列标题：配方项用于展示(配方构件、配方编辑时)的标题文本，支持多语言设置；在配方的导入和导出时，作为匹配依据，并且区分语言，即在英文状态下无法导入中文状态导出的配方。上表第一行即为列标题。

8.1 组态配方组

使用配方功能需要先完成配方组的组态设计，然后才能使用配方构件或配方函数对配方进行显示和编辑。

- 使用 McgsPro 的配方功能组态一个面包配方组，配方组包含有三个配方：甜面包配方、低糖面包配方、无糖面包配方，每个配方有 5 个配方项：糖、盐、面粉、水和蜂蜜。
- ① 创建关联配方项的变量：在实时数据库中新建 5 个浮点数型变量，名称依次为【糖】【盐】【面粉】【水】【蜂蜜】。

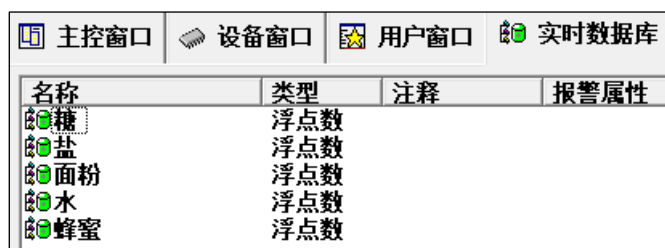


图 8.1-1 创建关联配方项的变量

- ② 进入【配方组态设计】界面：点击菜单栏 → 工具 → 配方组态设计，进入【配方组态设计】界面。

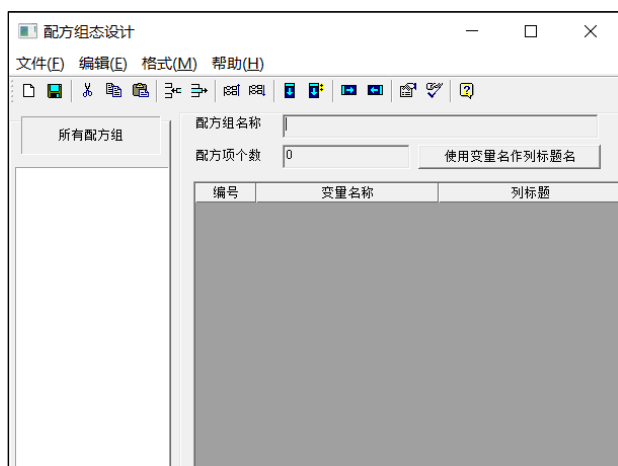


图 8.1-2 配方组态设计界面

- ③ 新建一个配方组并改名：用鼠标点击工具栏上的【新建】图标，增加一个配方组。然后右击新增加的配方组 → 配方组改名，将新配方组的名称改为【面包】。

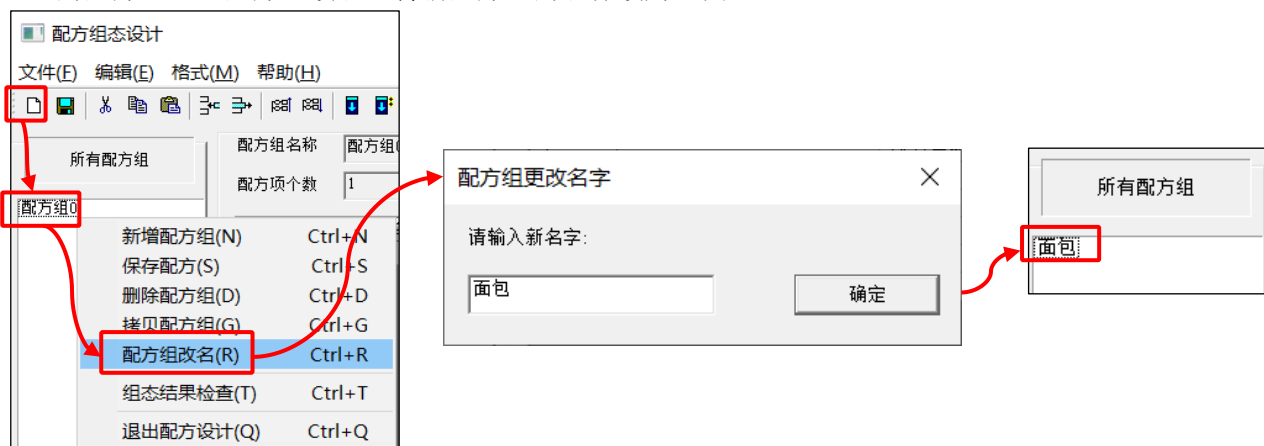


图 8.1-3 新建一个配方组并改名

- ④ 给配方组添加配方项：点击工具栏中的【增加一行】图标增加一行配方项 → 点击 按钮弹出【变量选择】界面 → 按住 Shift 键选择对象【糖】【盐】【面粉】【水】【蜂蜜】 → 确认。可以看到，刚才选择的对象都出现在了配方项列表中。

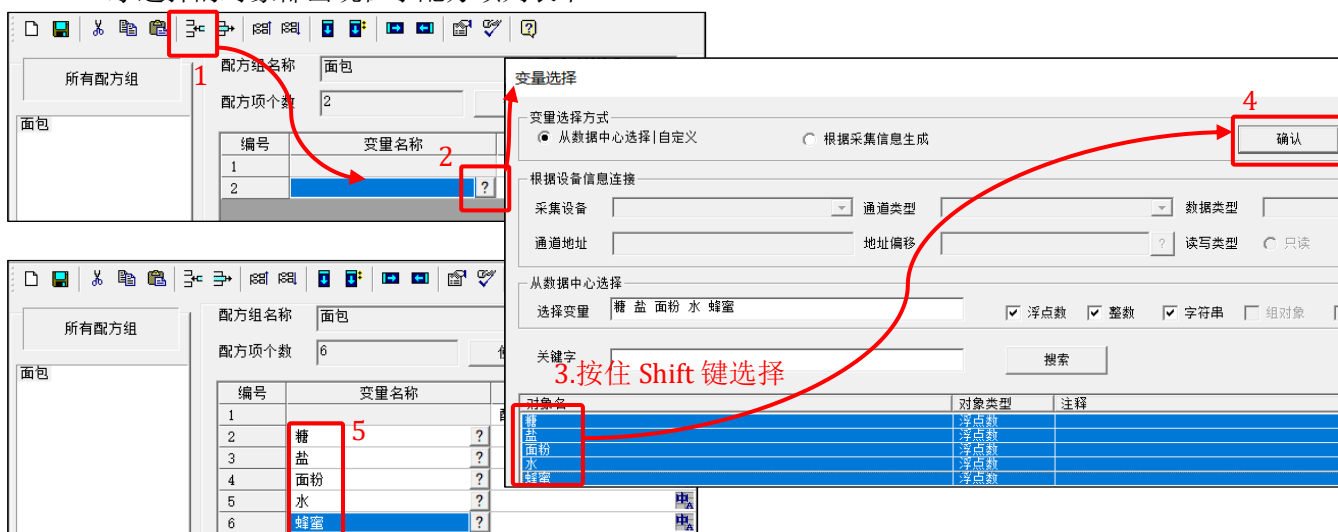


图 8.1-4 给配方组添加配方项

- ⑤ 然后点击配方项列表上方的【使用变量名做列标题名】按钮，将变量名称添加为配方的列标题名。此

时，相当于组态好了配方组的如图 8.1-5 红框所示部分的内容：

配方	糖	盐	面粉	水	蜂蜜
甜面包配方	80	10	80	30	10
低糖面包配方	30	5	80	30	0
无糖面包配方	10	5	80	30	0

图 8.1-5 配方组列表

- ⑥ 双击配方组名【面包】或点击上方工具栏中的【编辑配方】图标 → 点击三次【增加一行】图标，在下方的列表中输入如图 8.1-6 所示的内容 → 点击【保存】图标保存设置后关闭界面。

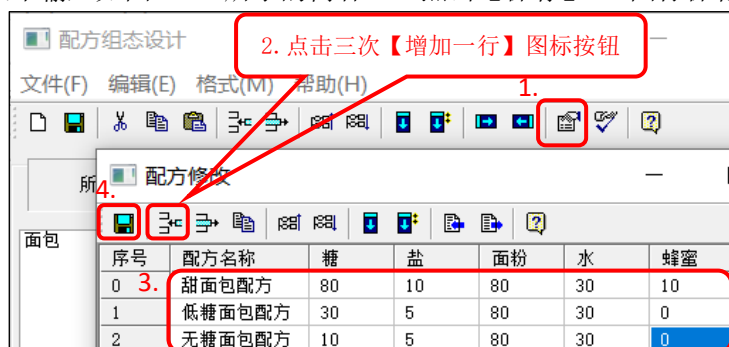


图 8.1-6 增加三条配方

● 配方的导入导出

在组态过程中，McgsPro 组态软件支持将配方组以 CSV 格式导出到电脑中，然后使用 Excel 打开表格进行修改。修改完成后将表格再次保存为 CSV 格式后导入到 McgsPro 组态软件中。

- ① 将配方组导出到 CSV 文件：双击配方组 → 点击【导出】图标 → 选择保存路径 → 保存。

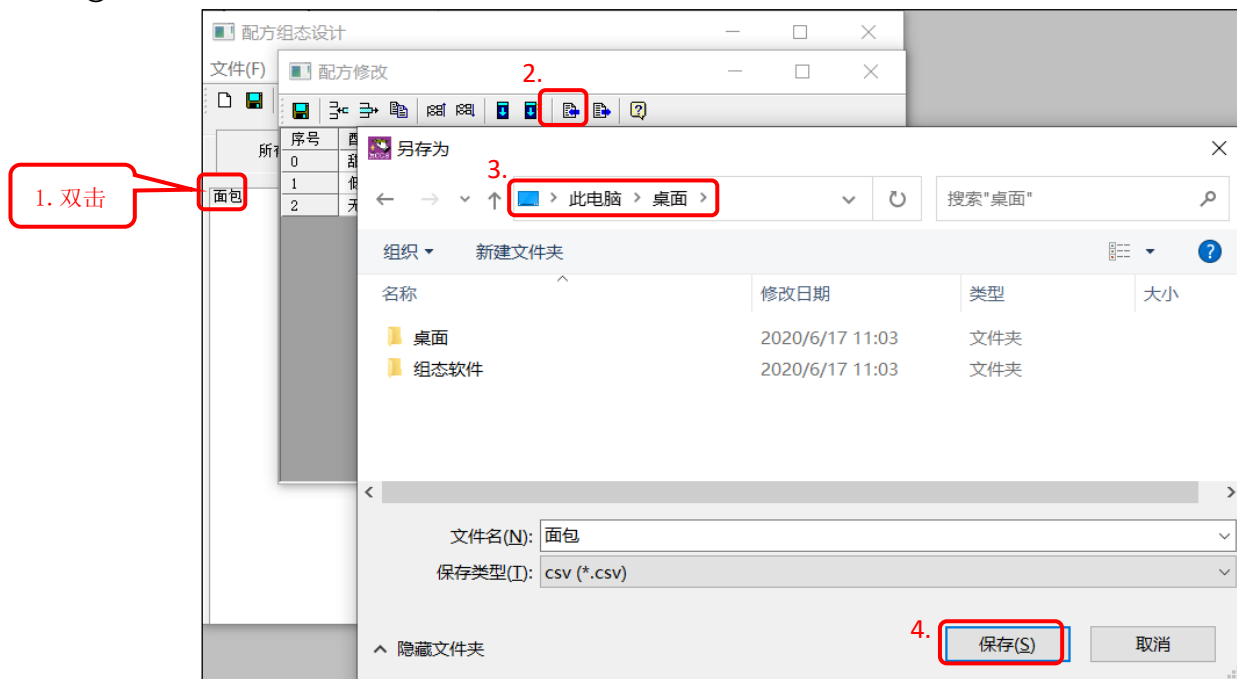


图 8.1-7 导出配方

- ② 使用 Excel 打开导出的 CSV 文件按需要对配方组进行修改。修改完成后依然保存为 CSV 格式。本样例将【无糖面包配方】中【糖】的含量从 10 修改为 0，然后依然保存为 CSV 文件。

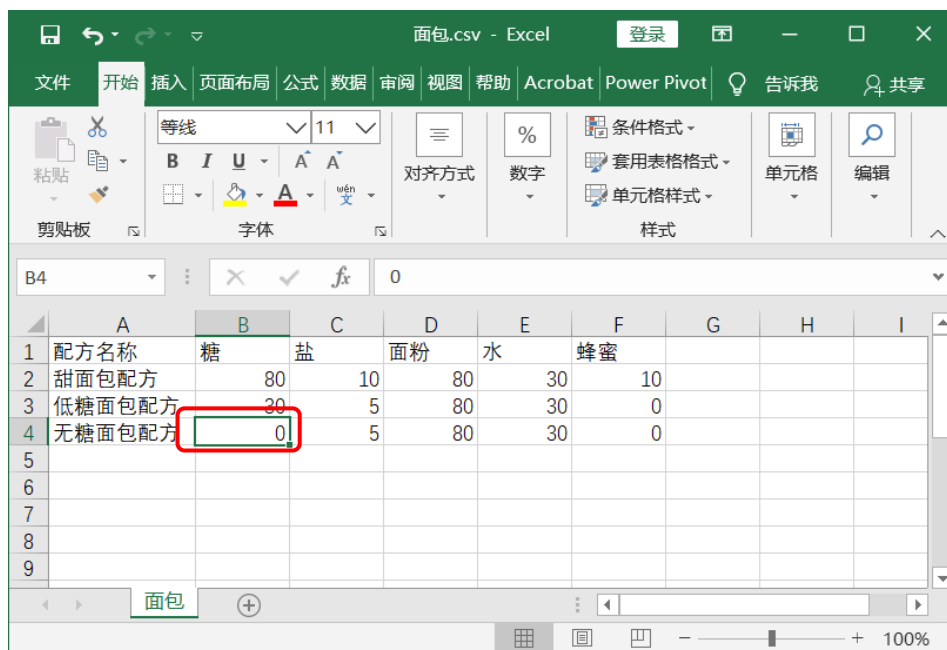


图 8.1-8 使用 Excel 修改配方

- ③ 点击【配方修改】窗口中工具栏上的【导入】图标，在弹出的窗口中选择刚才修改好的 CSV 文件后点击【打开】。

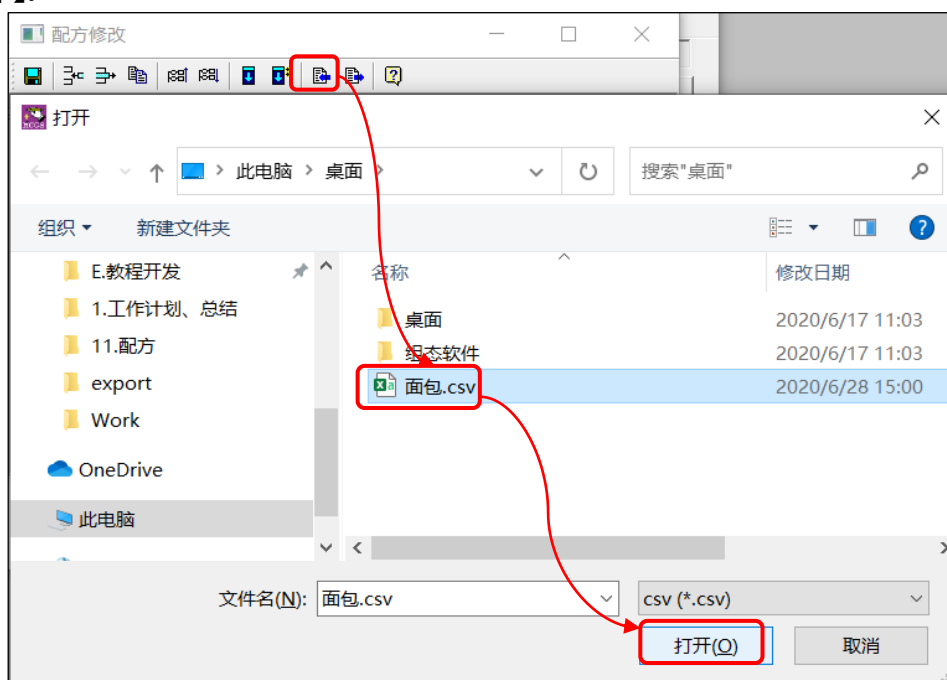


图 8.1-9 导入配方

- ④ 可以看到，导入后的配方组中，【无糖面包配方】中【糖】的含量变成了 0。

序号	配方名称	糖	盐	面粉	水	蜂蜜
0	甜面包配方	80	10	80	30	10
1	低糖面包配方	30	5	80	30	0
2	无糖面包配方	0	5	80	30	0

图 8.1-10 导入配方效果

8.2 配方构件控制生产工艺

组态完配方组后，还需要对配方组进行显示和编辑，本节介绍使用【配方】构件来控制面包生产工艺的组态过程。

8.2.1 组态配方构件

- ① 从工具栏中将【配方】构件拖拽到画面中合适的位置和合适的大小。

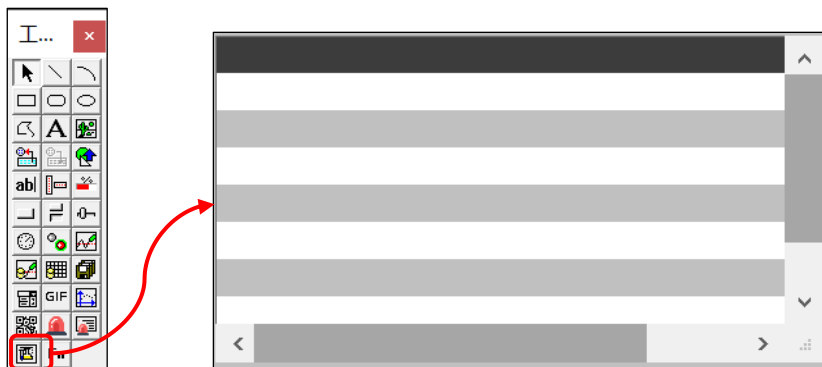


图 8.2-1 在画面中添加配方构件

- ② 设置配方构件【基本属性】：双击【配方】构件，打开【配方构件属性设置】界面。按需求设置【基本属性】功能页中的内容。为了使标题行更醒目，本节样例将【背景颜色】设置为【青色】，将【字符颜色】设置为【白色】，其他设置保持默认不变。

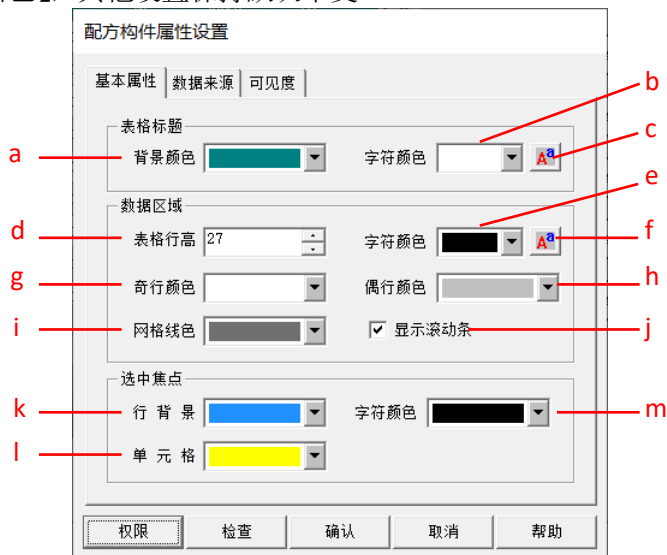


图 8.2-2 配方构件基本属性功能页

基本属性配置信息：

- a. 表格标题-背景颜色：标题行背景颜色；
- b. 表格标题-字符颜色：标题行字符颜色；

- c. 表格标题-字符设置：标题行字符、字形、大小设置；
- d. 数据区域-表格行高：表格每行高度，包括标题行；
- e. 数据区域-字符颜色：表格除标题行的字符颜色；
- f. 数据区域-字符设置：标题行字符、字形、大小设置；
- g. 数据区域-奇行颜色：表格奇数行背景颜色；
- h. 数据区域-偶行颜色：表格偶数行背景颜色；
- i. 数据区域-网格线色：表格线颜色；
- j. 数据区域-是否显示滚动条：运行时构件是否绘制滚动条；
- k. 选中焦点-行背景：焦点行背景颜色；
- l. 选中焦点-单元格：焦点单元格背景颜色；
- m. 选中焦点-字符颜色：焦点行字符颜色。

- ③ 设置配方构件【数据来源】：然后点击进入【数据来源】功能页，在【配方组】处的下拉菜单中选择 8.1 中设置好的【面包】配方组。勾选【全显示】，【全编辑】，调整【列宽】到合适的宽度使字体能全部显示。点击确认保存设置并关闭设置界面。

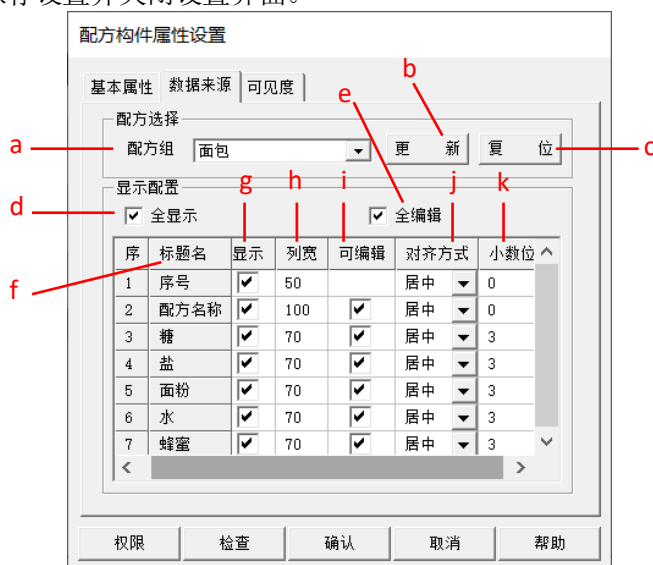


图 8.2-3 配方构件数据来源功能页

数据来源配置信息：

- a. 配方组：下拉框选择构件关联配方组。
 - b. 更新：修改配方组后需点击更新按钮使配方组最新。
 - c. 复位：关联配方显示配置初始化。
 - d. 全显示：全部列均显示或不显示。
 - e. 全编辑：除序号列外全部列在运行时可以或者不可以点击编辑。
 - f. 标题名：配方项显示名称，多语言时根据语言切换，无法修改。
 - g. 显示：控制指定列是否显示。
 - h. 列宽：控制指定列列宽。
 - i. 可编辑：控制指定列（序号列除外）是否可在运行时点击编辑。
 - j. 对齐方式：控制指定列文本对齐方式（左对齐/居中/右对齐）。
 - k. 小数位数：控制指定列数值显示时的小数位数。
- ④ 设置完成后，配方构件在画面中显示如图 8.2-4 所示。如果配方组内容显示不全，可以选中构件后，拖拽边框上出现的白色方框进行调整。

序号	配方名称	糖	盐	面粉	水	蜂蜜
0	name	0.000	0.000	0.000	0.000	0.000
1	name	0.000	0.000	0.000	0.000	0.000
2	name	0.000	0.000	0.000	0.000	0.000
3	name	0.000	0.000	0.000	0.000	0.000
4	name	0.000	0.000	0.000	0.000	0.000
5	name	0.000	0.000	0.000	0.000	0.000
6	name	0.000	0.000	0.000	0.000	0.000
7	name	0.000	0.000	0.000	0.000	0.000

图 8.2-4 配方构件组态完成效果图

- ⑤ 运行演示：保存工程后，下载工程，然后启动运行。如图 8.2-5 所示，配方构件显示了 8.1 中组态的配方组。鼠标选中的行变成了 8.2.1 中设置的蓝色，鼠标选中的单元格变成了设置的黄色。除了第一行的列标题和第一列的序号外，表格中的内容都可以通过双击表格弹出的键盘进行修改。通过右侧和下方的滚动条可以拉动

序号	配方名称	糖	盐	面粉	水	蜂蜜
0	甜面包配方	80.000	10.000	80.000	30.000	10.000
1	低糖面包配方	30.000	5.000	80.000	30.000	0.000
2	无糖面包配方	10.000	5.000	80.000	30.000	0.000

图 8.2-5 配方构件运行效果

8.2.2 配方构件方法应用

在运行状态下，常常也需要对配方组进行操作。本节重点介绍如何使用配方构件方法函数对运行状态下的配方组进行操作。

8.2.2.1 新增/删除一条配方。

● 新增一条配方

- ① 组态拖拽一个【标准按钮】构件到画面中，双击该构件，在其【基本属性】功能页中将文本修改为【新增】。进入【脚本程序】功能页，点击下方的【打开脚本程序编辑器】弹出【脚本程序】编辑窗口。在右侧的对象树中依次点击：用户窗口 → 窗口 0 → 控件 0[配方] → 方法，双击【Add】，则左侧的脚本编辑框中出现一行脚本【窗口 0.控件 0.Add】。点击保存后关闭窗口。



图 8.2-6 配方构件方法函数

配方构件方法函数解释：

- a. Refresh(): 刷新配方构件数据
- b. SetEditable(列索引, 编辑标志): 设置配方构件指定列是否可编辑
- c. SetColumnWidth(列索引, 列宽): 设置配方构件指定列列宽
- d. SelectCell(行索引, 列索引): 设置构件选中配方行与单元格
- e. GetName(): 获取当前配方的配方名称
- f. EditCell(): 编辑构件当前选中单元格
- g. PageUp(): 上翻页
- h. PageDown(): 下翻页
- i. Save(): 配方数据存盘
- j. ReadFromData(): 读取配方关联变量当前数据写入构件光标所在配方数据
- k. WriteToData(): 将构件光标所在配方数据写入配方关联变量中
- l. Add(): 在配方末尾新增配方
- m. Insert(): 配方构件光标所在配方前插入配方
- n. Copy(): 配方构件光标所在配方拷贝后插入到下一条配方, 并定位
- o. MoveUp(): 配方构件光标所在配方前移一个位置
- p. MoveDown(): 配方构件光标所在配方后移一个位置
- q. Delete(): 删除配方组处于构件光标位置的配方

- r. Import(): 从 csv 文件中导入配方数据
 - s. Export(): 导出配方数据到 csv 文件
- ② 保存工程后下载工程, 然后启动运行。通过点击【新增】按钮, 可以在配方构件中新增一行, 然后双击表格进行修改。



序号	配方名称	糖	盐	面粉	水	蜂蜜
0	甜面包配方	80.000	10.000	80.000	30.000	10.000
1	低糖面包配方	30.000	5.000	80.000	30.000	0.000
2	无糖面包配方	10.000	5.000	80.000	30.000	0.000
3		0.000	0.000	0.000	0.000	0.000

图 8.2-7 在配方构件中新增一行

- 删除一条配方: 用和新增一条配方同样的方法, 建一个【标准按钮】构件, 将它的文本修改为【删除】, 设置抬起脚本为【窗口 0.控件 0.Delete ()】。它的作用是删除配方组处于构件光标位置的配方。

8.2.2.2 配方数据的上传与下载

- 从 PLC 上传数据到配方: 在用户窗口中新建一个【标准按钮】构件, 将它的文本修改为【上传】, 用 8.2.2.1 中的步骤, 设置抬起脚本为【窗口 0.控件 0.ReadFromData()】。在运行状态下点击该按钮, 可读取配方关联变量当前数据写入构件光标所在配方数据中。
- 将配方数据下载到 PLC: 在用户窗口中新建一个【标准按钮】构件, 将它的文本修改为【下载】, 用 8.2.2.1 中的步骤, 设置抬起脚本为【窗口 0.控件 0.WriteToData()】。在运行状态下点击该按钮, 可将构件光标所在配方数据写入配方关联变量中。

8.2.2.3 配方的导入导出

在运行状态下, 也可以将配方导出到 CSV 文件, 使用 Excel 打开修改后, 保存为 CSV 文件再导入到配方中。导入导出文件名为配方组名称, 屏幕上文件路径为 U 盘路径, Windows 模拟运行环境上, 文件应位于../安装目录/Program/export 目录下。

- 配方的导出: 在用户窗口中新建一个【标准按钮】构件, 将它的文本修改为【导出】, 用 8.2.2.1 中的步骤, 设置抬起脚本为【窗口 0.控件 0.Export ()】。
- 配方的导入: 在用户窗口中新建一个【标准按钮】构件, 将它的文本修改为【导入】, 用 8.2.2.1 中的步骤, 设置抬起脚本为【窗口 0.控件 0.Import ()】。导入文件名为配方组名称, 文件中数据结构需要与配方结构一致, 否则会导入失败。

8.2.2.4 配方数据的保存

- 在用户窗口中新建一个【标准按钮】构件, 将它的文本修改为【保存】, 用 8.2.2.1 中的步骤, 设置抬起脚本为【窗口 0.控件 0.Save()】。工程运行时按下按钮, 系统将会把配方数据存盘到屏。掉电时没有保存的修改内容将会丢失。

8.3 配方函数控制生产工艺

在当前画面中没有配方构件时, 将不能使用配方构件方法函数, 且初始时配方处于未加载状态, 此时可以使用配方脚本函数对运行环境中的配方进行操作。本节继续使用 8.1 中组态好的面包配方组工程。

8.3.1 配方的加载和关闭

- !RecipeLoad(配方组名称): 加载指定配方组

实 例：!RecipeLoad("面包")

注意事项：!RecipeLoad 函数的调用和配方构件的存在均可以使配方变为已加载状态，只有当配方组处于已加载时，其他的配方函数才可以正确执行。配方已加载情况下再次加载不影响已加载状态。

● !RecipeClose(配方组名称)： 关闭配方组

实 例：!RecipeClose("面包")

注意事项：函数可以使配方关闭，离开已加载状态，但是不会保存配方数据到磁盘文件。

8.3.2 选择与修改配方

● !RecipeLoadByDialog(配方组名称,对话框标题)： 弹出配方选择对话框选择配方。

实 例： !RecipeLoadByDialog("面包","配方选择"), 效果如图 8.3-1 所示。



图 8.3-1 配方选择窗口

● 除了使用上图中的【编辑配方】按钮进入【配方编辑窗口】，还可以使用脚本函数实现该功能。

!RecipeModifyByDialog(配方组名称)： 弹出配方编辑对话框供配方编辑。

实 例：!RecipeModifyByDialog("面包"), 效果如图 8.3-2 所示，在此窗口中可实现配方的增加、删除、拷贝、上移、下移、存盘等操作。

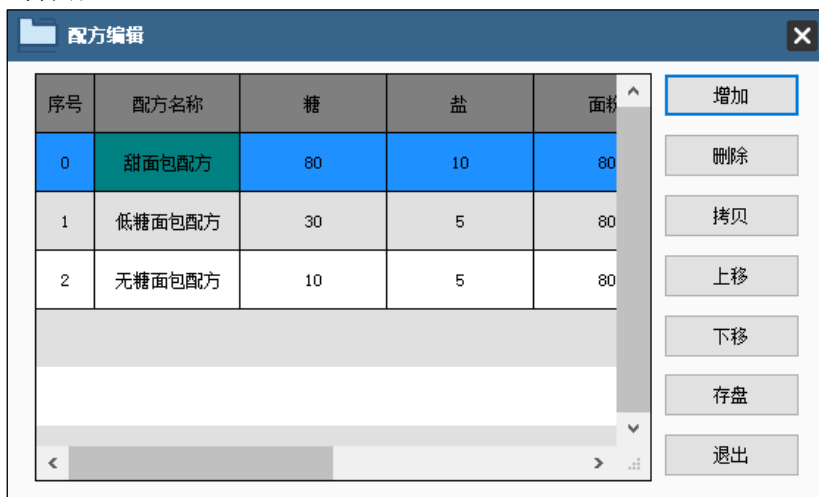


图 8.3-2 配方编辑窗口

8.3.3 配方数据的上传与下载

● 上传：

!RecipeGetValueFrom(配方组名称,组对象)： 将组对象中变量的当前值写入到配方组当前配方中

实 例：在实时数据库中新建一个组对象【配方组】，将配方项关联的变量【糖】【盐】【面粉】【水】【蜂蜜】设置为该组对象的成员。脚本!RecipeGetValueFrom("面包",组对象)的运行效果是：将组对象【配方组】中

的变量【糖】【盐】【面粉】【水】【蜂蜜】的当前值写入到配方组【面包】的当前配方中。

● 下载:

!RecipeGetCurrentValue(配方组名称): 获取当前配方数据写入到关联变量中

实 例: 脚本!RecipeGetCurrentValue("面包")的运行效果是: 获取配方组【面包】的当前配方数据写入到关联的变量【糖】【盐】【面粉】【水】【蜂蜜】中。

8.3.4 配方的导入导出

● 导出:

!RecipeToCsv(配方组名称,文件名,导出列名,起始编号,期望导出条数,实际导出条数,导出参数): 导出指定配方数据到 CSV 文件。

参数说明: 配方组名称, 字符串, 指定配方组

文件名, 字符串, csv 文件名, 不含"\"、"/"、".."等非法字符

导出列名, 字符串, 需要导出的列的集合, 英文逗号分割, 空表示全部列

起始编号, 整数, 从起始编号开始导出配方数据

期望导出条数, 整数, 希望导出配方的数目

实际导出条数, 整数, 实际导出的配方数目, 此为输出参数

导出参数, 整数

BIT0 表示导出数据的方式

BIT0 = 0: 代表保存现有数据, 并追加到文件尾部

BIT0 = 1: 代表删除文件中现有数据, 写入导出数据

BIT1 表示控制导出数据的内容

BIT1 = 0: 代表包含配方组成员字段名称

BIT1 = 1: 代表不包含配方组成员字段名称

实 例: 脚本!RecipeToCsv("面包","面包.csv","",0,100,实际导出条数,3)的运行效果是: 将配方组【面包】的数据, 导出到 U 盘目录下名为【面包.CSV】的文件中, 导出全部列, 从编号 0 开始导出, 最多导出 100 条, 把实际导出条数保存的变量【实际导出条数】中。导出时, 删除文件中现有的数据, 写入导出数据, 导出数据不包含配方组成员字段名称。

注意事项: 导出路径为 U 盘目录, 导出配方时建议只插入一个 U 盘, 避免系统找不到 U 盘。Windows 模拟运行环境导出目录为../安装目录/Program/export。

● 导入:

!CsvToRecipe(配方组名称,文件名,导入列名,期望导入条数,实际导入条数,导入参数): 从 Csv 文件中导入数据到配方组

参数说明: 配方组名称, 字符串, 指定配方组

文件名, 字符串, csv 文件名, 不含"\"、"/"、".."等非法字符

导入列名, 字符串, 需要导入的列的集合, 英文逗号分割, 空表示全部列

期望导入条数, 整数, 希望导入配方的数目

实际导入条数, 整数, 实际导入的配方数目, 此为输出参数

导入参数, 整数

BIT0 表示导入数据的方式

BIT0 = 0: 代表更新导入文件数据, 已有配方更新数据, 没有的配方新增配方数据

BIT0 = 1: 代表删除配方现有数据, 导入文件数据

BIT1 表示导入文件是否有文件列标题

BIT1 = 0: 代表文件中不包含列标题字段

BIT1 = 1: 代表文件中包含列标题字段

实 例: 脚本!CsvToRecipe("面包","面包.csv","",100,实际导入条数,2)的作用是: 将文件名为【面包.CSV】的文件数据导入到配方组【面包】中, 导入配方的全部列, 最多导入 100 行, 将实际导入条数保存到变量【实际导入条数】中。导入时, 更新已有配方数据, 新增没有的配方数据, 且导入文件包含文件列标题。

注意事项: 导入配方时建议只插入一个 U 盘, 避免系统找不到 U 盘。Windows 模拟运行环境导入目录为../安装目录/Program/export 中。此外, 目前配方导入不支持分列导入, 在写配方导入时导入列名请保持为空列名, 始终导入全部列。

8.3.5 配方的保存

!RecipeSave(配方组名称): 将当前配方组数据写入磁盘文件中保存

实 例: 配方返回值 = !RecipeSave("面包")

注意事项: 未保存配方掉电时, 未保存内容将会丢失。

8.4 顺序控制器功能

顺序控制器: 通过执行事先配置好的动作, 控制设备的运行, 使产品生产流程安全可靠。本节教程我们以点胶机为例学习顺序控制功能。如图 8.4-1, 以配方构件为主, 通过配方构件脚本函数, 模拟点胶机的动作, 按照预置动作步骤, 完成点胶。



图 8.4-1 点胶机 控制画面

● 点胶机控制画面功能解释

- 点胶机预置指令: 由配方构件显示的点胶机动作指令。
- 点胶操作: 提供点胶机指令内容设置功能, 点击按钮可插入对应的点胶指令。
- PLC 通道值显示: 显示当前点胶机正在执行的控制指令。
- 点胶机流程控制: 增删、保存、导入、导出点胶指令, 以及执行选中的指令, 或者执行全部指令。

● 点胶机操作流程:

工程运行时, 用户首先通过“d. 点胶机流程控制”插入指令行; 然后在“a. 点胶机预置指令”选中插入的指令行; 再通过“b. 点胶操作”设置选中指令行当命令内容; 最后点击途中的“保存”按钮, 保存命令内容。完成以上操作, 点击“指令选中指令”或“执行全部指令”, 将指定的命令内容和动作内容下载到 PLC。

下面介绍顺序控制器点胶机点胶功能组态:

方案: 本节教程下位机采用三菱 FX0N PLC, 通讯协议采用“三菱 FX 系列串口”。使用 McgsPro 提供的配方功能存储点胶机控制指令, 通过配方构件显示当前的预置指令, 并用配方构件方法函数完成预置指令插入、修改指令内容、保存, 并支持将预置的控制指令下载到 PLC。

首先进行配方功能组态:

① 新建变量: 通过 McgsPro 组态软件新建一个工程, 进入工程的“实时数据库”, 如图 8.4-2, 新增 5 个字符串变量: “命令内容”、“动作内容 1”、“动作内容 2”、“动作内容 3”、“动作内容 4” (这 5 个字符串变量用于配方项关联, 以及在配方构件中显示控制指令); 以及新增整数变量“命令内容_数值”, 浮点数变量“动作内容 1_数值”、“动作内容 2_数值”、“动作内容 3_数值”、“动作内容 4_数值”, 用来保存转换成数值的命令和动作内容。另外, 再新增 1 个整数变量“使能”, 在工程运行时, 通过该变量控制按钮的使能功能, 防止现场用户误操作。

名称	类型	注释	报警属性
InputSTime	字符串	系统内建...	
InputETime	字符串	系统内建...	
InputUser1	字符串	系统内建...	
InputUser2	字符串	系统内建...	
命令内容	字符串		
动作内容1	字符串		
动作内容2	字符串		
动作内容3	字符串		
动作内容4	字符串		
命令内容_数值	整数		
动作内容1_数值	浮点数		
动作内容2_数值	浮点数		
动作内容3_数值	浮点数		
动作内容4_数值	浮点数		
使能	整数		

图 8.4-2 新建配方项变量

② 进入配方组态：如图 8.4-3，通过软件菜单栏“工具”→“配方组态设计”，进入配方组态设计界面

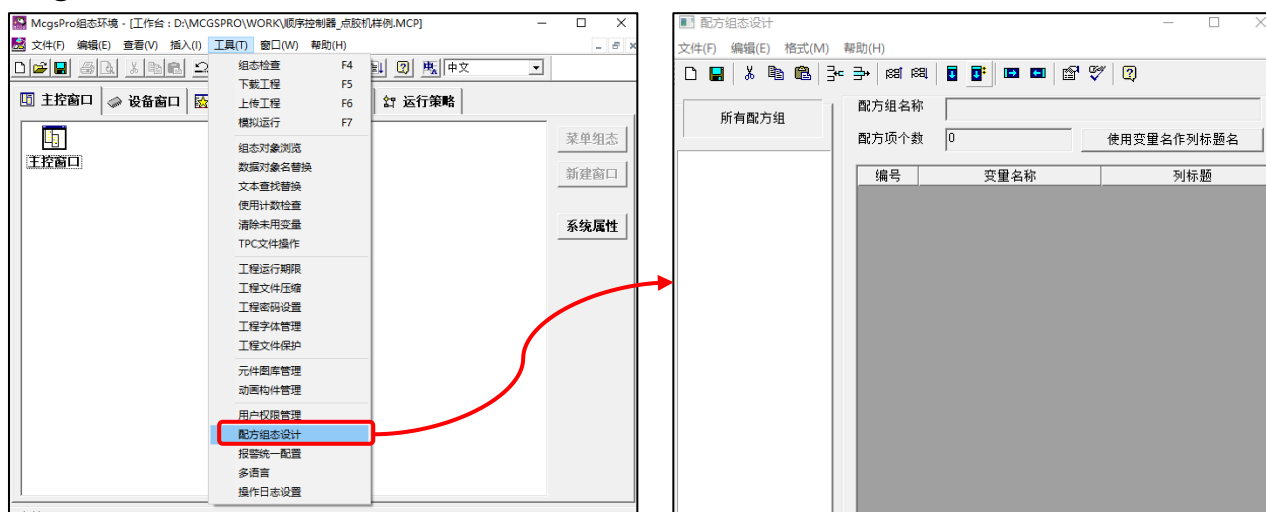


图 8.4-3 进入配方组态设计

③ 配方功能组态：如图 8.4-4，在配方组态设计页面新增配方组“点胶机工艺”，再新增配方项，并在新增的配方项中关联变量“命令内容”、“动作内容 1”、“动作内容 2”、“动作内容 3”、“动作内容 4”。配方组态完成，保存并关闭配方组态设计界面。

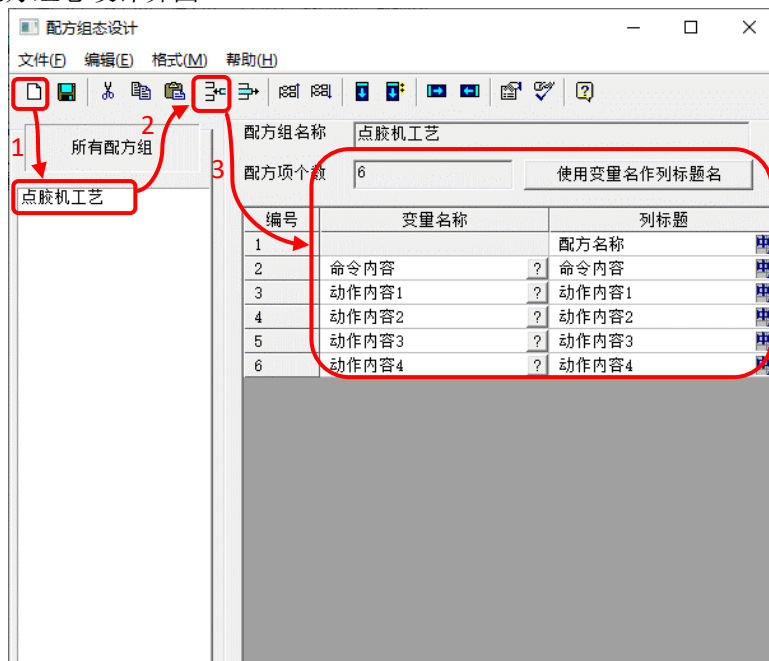


图 8.4-4 配方功能组态

④ 添加驱动：如图 8.4-5，双击工作台中的“设备窗口”，在设备窗口界面添加“通用串口父设备”和“三联 FX 系列串口”驱动，驱动添加方法参考入门篇章 3.5.2。



图 8.4-5 添加驱动：三菱 FX 系列串口

⑤ 增加设备通道：双击上图的“三菱 FX 系列串口”，进入图 8.4-6 所示的设备编辑窗口，通过“增加设备通道”按钮，新增 PLC 通道，选择 D 寄存器，通道地址地址 0、1、3、5、7，用于接收触摸屏发送的点胶控制指令。

注意：增加设备通道前，请先通过“删除设备通道”，删除用户不需要的默认通道。

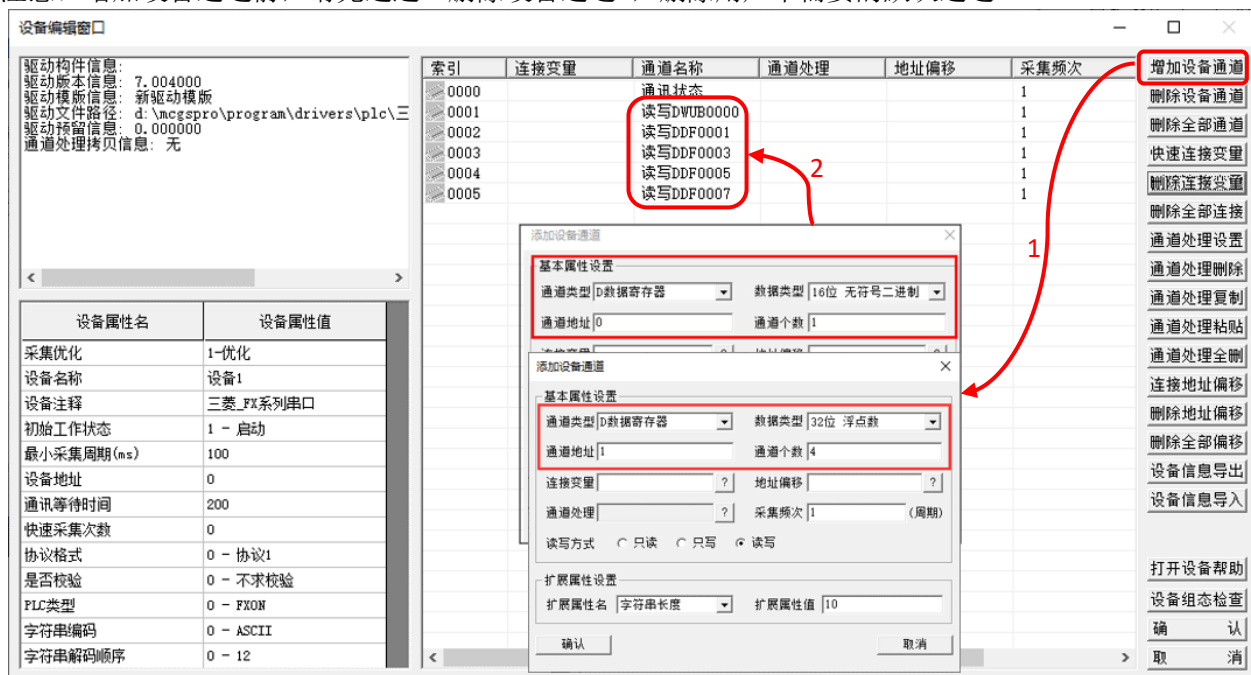


图 8.4-6 增加设备通道

⑥ 快速连接变量：如图 8.4-7，通过【快速连接变量】→【默认设备变量连接】，为设备通道中关联变量。

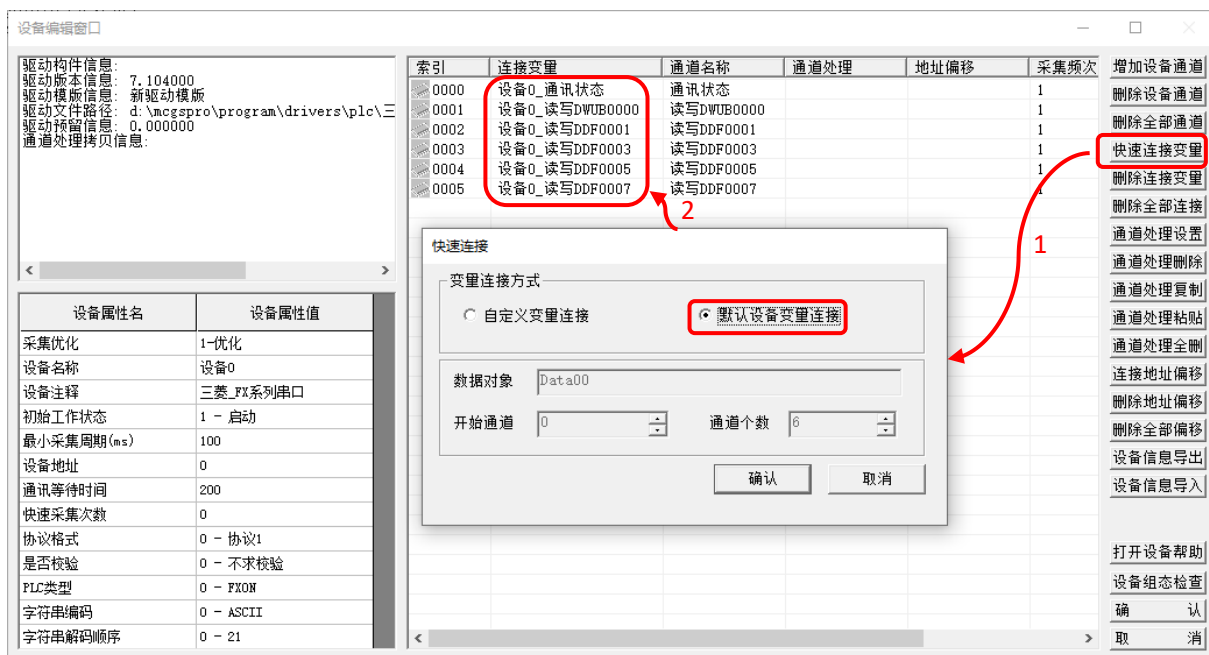


图 8.4-7 快速连接变量

⑦ 添加变量: 完成上图设置后, 点击设备编辑窗口右下方的确认按钮, 弹出图 8.4-8 所示的“添加变量”界面, 点击“全部添加”, 将“连接变量”添加到实时数据库。

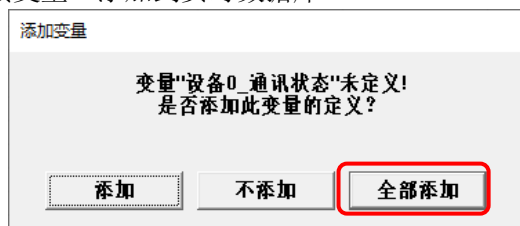


图 8.4-8 添加变量

完成以上操作, 下面进行配方构件组态:

- ① 新建窗口: 首先进入工作台“用户窗口”, 新建一个窗口, 并修改窗口名为“点胶操作”。
- ② 窗口画面组态: 双击窗口“点胶操作”, 进入其窗口组态画面, 通过工具箱添加配方构件、输入框构件、标签构件, 组态如图 8.4-9 所示画面。



图 8.4-9 窗口画面组态

③ 配方构件设置：双击上图绘制的配方构件，按照图 8.4-10 设置配方构件。



图 8.4-10 配方构件设置

完成配方构件组态，下面进行点胶操作组态：

① 点胶操作组态：如图 8.4-11 所示，依次组态图中所示的点胶命令。



图 8.4-11 点胶命令展示

注意：工程运行时，选中配方构件显示的配方，点击“点胶操作”中的命令，可将对应的点胶命令内容写入所选配方。

② 点胶开始：在“点胶开始”按钮的脚本程序中，添加图 8.4-12 所示脚本，用于插入“点胶开始”命令。

功能介绍	点胶开始
编写脚本	<pre>1 插入动作：“点胶开始” 2 命令内容 = “点胶开始”，给命令赋值 3 动作内容1 = “”，给动作内容1赋值 4 动作内容2 = “”，给动作内容2赋值 5 动作内容3 = “”，给动作内容3赋值 6 动作内容4 = “”，给动作内容4赋值 7 点胶操作.控件3.ReadFromData()，将动作内容写入配方</pre>
子窗口脚本	

图 8.4-12 点胶命令组态：点胶开始

③ X 轴运动：在“X 轴运动”按钮的脚本程序中，添加图 8.4-13 所示脚本，用于打开子窗口“X 轴”。并

在子窗口“X 轴”中组态如图 8.4-13 所示的窗口画面和脚本。

功能介绍	X轴运动
编写脚本	1 用户窗口.X轴.Open() '打开X轴运动设置子窗口
子窗口脚本	 <pre> 1 '插入动作: "X轴运动" 2 3 命令内容 = "X轴运动" '给命令赋值 4 动作内容1 = "X: "+IFormat(动作内容1_数值,"0.00") '格式化"动作内容1_数值"的值,合成字符串写入"动作内容1" 5 动作内容2 = "速度: "+IFormat(动作内容2_数值,"0.00") '格式化"动作内容2_数值"的值,合成字符串写入"动作内容2" 6 动作内容3 = "" '给动作内容3赋值为空 7 动作内容4 = "" '给动作内容4赋值为空 8 9 点胶操作.控件3.ReadFromData() '将动作内容写入配方 10 iCloseAllSubWnd() '关闭子窗口 </pre>

图 8.4-13 点胶命令组态: X 轴运动

注意: 组态“X 轴运动”需要新建用户窗口(窗口名称如图 8.4-13 所示:“X 轴”),并将其设置为子窗口,子窗口设置方法参考进阶篇章章节 4.2.4。

④ Y 轴运动: 在“Y 轴运动”按钮的脚本程序中,添加图 8.4-14 所示脚本,用于打开子窗口“Y 轴”。并在子窗口“Y 轴”中组态如图 8.4-14 所示的窗口画面和脚本。

功能介绍	Y轴运动
编写脚本	1 用户窗口.Y轴.Open() '打开Y轴运动设置子窗口
子窗口脚本	 <pre> 1 '插入动作: "Y轴运动" 2 3 命令内容 = "Y轴运动" '给命令赋值 4 动作内容1 = "Y: "+IFormat(动作内容1_数值,"0.00") '格式化"动作内容1_数值"的值,合成字符串写入"动作内容1" 5 动作内容2 = "速度: "+IFormat(动作内容2_数值,"0.00") '格式化"动作内容2_数值"的值,合成字符串写入"动作内容2" 6 动作内容3 = "" '给动作内容3赋值为空 7 动作内容4 = "" '给动作内容4赋值为空 8 9 点胶操作.控件3.ReadFromData() '将动作内容写入配方 10 iCloseAllSubWnd() '关闭子窗口 </pre>

图 8.4-14 点胶命令组态: Y 轴运动

⑤ X+Y 轴运动: 在“X+Y 轴运动”按钮的脚本程序中,添加图 8.4-15 所示脚本,用于打开子窗口“XY 轴”。并在子窗口“XY 轴”中组态如图 8.4-15 所示的窗口画面和脚本。

功能介绍	X+Y轴运动
编写脚本	1 用户窗口.XY轴.Open() '打开XY轴运动设置子窗口
子窗口脚本	 <pre> 1 '插入动作: "XY轴运动" 2 3 命令内容 = "XY轴运动" '给命令赋值 4 动作内容1 = "X: "+IFormat(动作内容1_数值,"0.00") '格式化"动作内容1_数值"的值,合成字符串写入"动作内容1" 5 动作内容2 = "Y: "+IFormat(动作内容2_数值,"0.00") '格式化"动作内容2_数值"的值,合成字符串写入"动作内容2" 6 动作内容3 = "速度: "+IFormat(动作内容3_数值,"0.00") '格式化"动作内容3_数值"的值,合成字符串写入"动作内容3" 7 动作内容4 = "" '给动作内容4赋值为空 8 9 点胶操作.控件3.ReadFromData() '将动作内容写入配方 10 iCloseAllSubWnd() '关闭子窗口 </pre>

图 8.4-15 点胶命令组态: X+Y 轴运动

⑥ 延时：在“延时”按钮的脚本程序中，添加图 8.4-16 所示脚本，用于打开子窗口“延时”。并在子窗口“延时”中组态如图 8.4-16 所示的窗口画面和脚本。

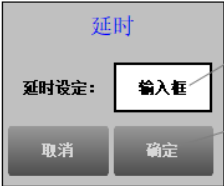
功能介绍	延时
编写脚本	<pre> 1 用户窗口.延时.Open() '打开“延时”设置子窗口 </pre>
子窗口脚本	 <pre> 1 插入动作：“延时” 2 3 命令内容 = “延时(ms)” '给命令赋值 4 动作内容1 = “延时(ms)” + IFormat(动作内容1_数值,“0.00”) '格式化“动作内容1_数值”的值，合成字符串写入“动作内容1” 5 动作内容2 = “” '给动作内容2赋值为空 6 动作内容3 = “” '给动作内容3赋值为空 7 动作内容4 = “” '给动作内容4赋值为空 8 9 点胶操作.控件3.ReadFromData() '将动作内容写入配方 10 ICloseAllSubWnd() '关闭子窗口 </pre>

图 8.4-16 点胶命令组态：延时

⑦ XY 轴回原：在“XY 轴回原”按钮的脚本程序中，添加图 8.4-17 所示脚本，用于插入“XY 轴回原”命令。

功能介绍	XY轴回原
编写脚本	<pre> 1 插入动作：“XY轴回原” 2 命令内容 = “XY轴回原” '给命令赋值 3 动作内容1 = “” '给动作内容1赋值 4 动作内容2 = “” '给动作内容2赋值 5 动作内容3 = “” '给动作内容3赋值 6 动作内容4 = “” '给动作内容4赋值 7 点胶操作.控件3.ReadFromData() '将动作内容写入配方 </pre>
子窗口脚本	

图 8.4-17 点胶命令组态：XY 轴回原

⑧ Z 轴运动：在“Z 轴运动”按钮的脚本程序中，添加图 8.4-18 所示脚本，用于打开子窗口“Z 轴”。并在子窗口“Z 轴”中组态如图 8.4-18 所示的窗口画面和脚本。

功能介绍	Z轴运动
编写脚本	<pre> 1 用户窗口.Z轴.Open() '打开Z轴运动设置子窗口 </pre>
子窗口脚本	 <pre> 1 插入动作：“Z轴运动” 2 3 命令内容 = “Z轴运动” '给命令赋值 4 动作内容1 = “Z” + IFormat(动作内容3_数值,“0.00”) '格式化“动作内容1_数值”的值，合成字符串写入“动作内容1” 5 动作内容2 = “速度” + IFormat(动作内容4_数值,“0.00”) '格式化“动作内容2_数值”的值，合成字符串写入“动作内容2” 6 动作内容3 = “” '给动作内容3赋值为空 7 动作内容4 = “” '给动作内容4赋值为空 8 9 点胶操作.控件3.ReadFromData() '将动作内容写入配方 10 ICloseAllSubWnd() '关闭子窗口 </pre>

图 8.4-18 点胶命令组态：Z 轴运动

⑨ Z 轴回原：在“Z 轴回原”按钮的脚本程序中，添加图 8.4-19 所示脚本，用于插入“Z 轴回原”命令。

功能介绍	Z轴回原
编写脚本	<pre> 1 '插入动作: "Z轴回原" 2 命令内容 = "Z轴回原" '给命令赋值 3 动作内容1 = "" '给动作内容1赋值 4 动作内容2 = "" '给动作内容2赋值 5 动作内容3 = "" '给动作内容3赋值 6 动作内容4 = "" '给动作内容4赋值 7 点胶操作.控件3.ReadFromData() '将动作内容写入配方 </pre>
子窗口脚本	

图 8.4-19 点胶命令组态: Z 轴回原

⑩ 相对运动: 在“相对运动”按钮的脚本程序中, 添加图 8.4-21 所示脚本, 用于打开子窗口“相对运动”。并在子窗口“相对运动”中组态如图 8.4-20 所示的窗口画面和脚本。

功能介绍	相对运动
编写脚本	<pre> 1 用户窗口. 相对运动. Open() '打开“相对运动”设置子窗口 </pre>
子窗口脚本	 <pre> 1 '插入动作: "相对运动" 2 命令内容 = "相对运动" '给命令赋值 3 动作内容1 = "X:" + IFormat(动作内容1_数值, "0.00") '格式化“动作内容1_数值”的值, 合成字符串写入“动作内容1” 4 动作内容2 = "Y:" + IFormat(动作内容2_数值, "0.00") '格式化“动作内容2_数值”的值, 合成字符串写入“动作内容2” 5 动作内容3 = "Z:" + IFormat(动作内容3_数值, "0.00") '格式化“动作内容3_数值”的值, 合成字符串写入“动作内容3” 6 动作内容4 = "速度:" + IFormat(动作内容4_数值, "0.00") '格式化“动作内容4_数值”的值, 合成字符串写入“动作内容4” 7 8 点胶操作.控件3.ReadFromData() '将动作内容写入配方 9 !CloseAllSubWnd() '关闭子窗口 10 </pre>

图 8.4-20 点胶命令组态: 相对运动

⑪ 绝对运动: 在“绝对运动”按钮的脚本程序中, 添加图 8.4-21 所示脚本, 用于打开子窗口“绝对运动”。并在子窗口“绝对运动”中组态如图 8.4-21 所示的窗口画面和脚本。

功能介绍	绝对运动
编写脚本	<pre> 1 用户窗口. 绝对运动. Open() '打开“绝对运动”设置子窗口 </pre>
子窗口脚本	 <pre> 1 '插入动作: "绝对运动" 2 命令内容 = "绝对运动" '给命令赋值 3 动作内容1 = "X:" + IFormat(动作内容1_数值, "0.00") '格式化“动作内容1_数值”的值, 合成字符串写入“动作内容1” 4 动作内容2 = "Y:" + IFormat(动作内容2_数值, "0.00") '格式化“动作内容2_数值”的值, 合成字符串写入“动作内容2” 5 动作内容3 = "Z:" + IFormat(动作内容3_数值, "0.00") '格式化“动作内容3_数值”的值, 合成字符串写入“动作内容3” 6 动作内容4 = "速度:" + IFormat(动作内容4_数值, "0.00") '格式化“动作内容4_数值”的值, 合成字符串写入“动作内容4” 7 8 点胶操作.控件3.ReadFromData() '将动作内容写入配方 9 !CloseAllSubWnd() '关闭子窗口 10 </pre>

图 8.4-21 点胶命令组态: 绝对运动

⑫ 点胶停止: 在“点胶停止”按钮的脚本程序中, 添加图 8.4-22 所示脚本, 用于插入“点胶停止”命令。

功能介绍	点胶停止
编写脚本	<pre> 1 插入动作: "点胶停止" 2 命令内容 = "点胶停止" 给命令赋值 3 动作内容1 = "" 给动作内容1赋值 4 动作内容2 = "" 给动作内容2赋值 5 动作内容3 = "" 给动作内容3赋值 6 动作内容4 = "" 给动作内容4赋值 7 点胶操作.控件3.ReadFromData() 将动作内容写入配方 </pre>
子窗口脚本	

图 8.4-22 点胶命令组态：点胶停止

完成点胶操作组态，下面组态配方操作功能：

① 配方构件方法函数：通过配方构件方法函数实现配方操作，用户可以在工程运行时插入、新增、删除、保存、导入、导出配方。本期教程需要添加的方法函数如图 8.4-23 所示。配方构件方法函数添加方法请参考：进阶篇 8.2.2 配方构件方法应用。



图 8.4-23 组态配方构件方法函数

完成配方操作功能组态，下面进行点胶指令执行组态：

① 由于点胶指令中有延时功能，所以点胶脚本需要添加到策略中，本期教程我们在用户策略中添加点胶指令执行脚本，再通过标准按钮构件执行用户策略。本期教程需组态图 8.4-24 所示的“执行选中指令”、“执行全部指令”功能。



图 8.4-24 组态执行点胶指令功能

② 执行选中指令：用户需要在“运行策略”中新建用户策略，策略名命名为“执行选中指令”。并在策略“执行选中指令”中新增一个策略行，再添加图 8.4-26 所示脚本，最后通过图 8.4-25 所示的标准按钮构件“执行选中指令”调用该策略，实现工程运行时点击按钮“执行选中指令”，即可将当前配方构件中选择的指令下载到 PLC。



图 8.4-25 标准按钮：执行选中指令

```

1 使能 = 1 '按钮使能控制, 0允许控制, 1禁止控制
2
3 点胶操作.控件3.WriteToData() '将当前选中的配方构件中的控制指令写入“动作内容”
4 IF 命令内容 = "点胶开始" THEN 命令内容.数值 = 1
5 IF 命令内容 = "X轴运动" THEN 命令内容.数值 = 2
6 IF 命令内容 = "Y轴运动" THEN 命令内容.数值 = 3
7 IF 命令内容 = "Z轴运动" THEN 命令内容.数值 = 4
8 IF 命令内容 = "XY轴运动" THEN 命令内容.数值 = 5
9 IF 命令内容 = "延时(ms)" THEN 命令内容.数值 = 6
10 IF 命令内容 = "XY轴回原" THEN 命令内容.数值 = 7
11 IF 命令内容 = "Z轴回原" THEN 命令内容.数值 = 8
12 IF 命令内容 = "Z轴运动" THEN 命令内容.数值 = 9
13 IF 命令内容 = "绝对运动" THEN 命令内容.数值 = 10
14 IF 命令内容 = "点胶停止" THEN 命令内容.数值 = 11
15 '判断字符串变量“命令”，给“命令内容_数值”赋固定的数值
16
17 '判断命令内容，命令内容为延时，执行延时命令；命令内容为非延时，则执行对应的动作命令
18 IF 命令内容.数值 = 5 THEN
19     动作内容1.数值 = !Val(!MidEx(动作内容1, !StrReverseFind(动作内容1, ":")+2))
20     !Sleep(动作内容1.数值)
21     '判断“命令内容”为延时，将“命令内容_数值”写入PLC，并延时设定时间，再进行下一步动作
22 else
23     动作内容1.数值 = !Val(!MidEx(动作内容1, !StrReverseFind(动作内容1, ":")+2))
24     动作内容2.数值 = !Val(!MidEx(动作内容2, !StrReverseFind(动作内容2, ":")+2))
25     动作内容3.数值 = !Val(!MidEx(动作内容3, !StrReverseFind(动作内容3, ":")+2))
26     动作内容4.数值 = !Val(!MidEx(动作内容4, !StrReverseFind(动作内容4, ":")+2))
27     设备0.读写DWF0000 = 命令内容.数值
28     设备0.读写DWF0001 = 动作内容1.数值
29     设备0.读写DWF0003 = 动作内容2.数值
30     设备0.读写DWF0005 = 动作内容3.数值
31     设备0.读写DWF0007 = 动作内容4.数值
32     '判断“命令内容”非延时，将“命令内容_数值”写入PLC，同时将动作内容写入PLC
33 endif
34
35 使能 = 0 '按钮使能控制, 0允许控制, 1禁止控制
        
```

图 8.4-26 “执行选中指令”策略脚本内容

③ 执行全部指令：用户需要在“运行策略”中新建用户策略，策略名命名为“执行全部指令”。并在策略“执行全部指令”中新增一个策略行，再添加图 8.4-28 所示脚本，最后通过图 8.4-27 所示的标准按钮构件“执行选中指令”调用该策略，实现工程运行时点击按钮“执行全部指令”，即可将当前配方中的全部指令依次下载到 PLC。



图 8.4-27 标准按钮：执行全部指令

```

1 使能 = 1 '按钮使能控制, 0允许控制, 1禁止控制
2
3 DIM i as INTEGER '定义整数变量i
4 DIM j as INTEGER '定义整数变量j
5 i = RecipeGetCount("点胶机工艺") '获取点胶机工艺的配方个数
6
7 IF i > 0 THEN
8     j = 0
9     '启动While循环, 按照序号为0、1、2...的顺序依次将对应配方的数据复制到配方项关联的变量上
10    WHILE j < i
11        RecipeLoadByNum("点胶机工艺", j)
12        命令内容_数值 = 0
13        IF 命令内容 = "点胶开始" THEN 命令内容_数值 = 1
14        IF 命令内容 = "X轴运动" THEN 命令内容_数值 = 2
15        IF 命令内容 = "Y轴运动" THEN 命令内容_数值 = 3
16        IF 命令内容 = "XY轴运动" THEN 命令内容_数值 = 4
17        IF 命令内容 = "延时(ms)" THEN 命令内容_数值 = 5
18        IF 命令内容 = "X轴回原" THEN 命令内容_数值 = 6
19        IF 命令内容 = "Y轴回原" THEN 命令内容_数值 = 7
20        IF 命令内容 = "Z轴回原" THEN 命令内容_数值 = 8
21        IF 命令内容 = "相对运动" THEN 命令内容_数值 = 9
22        IF 命令内容 = "绝对运动" THEN 命令内容_数值 = 10
23        IF 命令内容 = "点胶停止" THEN 命令内容_数值 = 11
24        '判断字符串变量“命令”，给“命令内容_数值”赋固定的数值
25
26        '判断命令内容，命令内容为延时，执行延时命令；命令内容为非延时，则执行对应的动作命令
27        IF 命令内容_数值 = 5 THEN
28            动作内容1_数值 = Val( Mid$ (动作内容1, StrReverseFind(动作内容1, ":")+2) )
29            ISleep(动作内容1_数值)
30            '判断“命令内容”为延时，将“命令内容_数值”写入PLC，并延时设定时间，再进行下一步动作
31        else
32            动作内容1_数值 = Val( Mid$ (动作内容1, StrReverseFind(动作内容1, ":")+2) )
33            动作内容2_数值 = Val( Mid$ (动作内容2, StrReverseFind(动作内容2, ":")+2) )
34            动作内容3_数值 = Val( Mid$ (动作内容3, StrReverseFind(动作内容3, ":")+2) )
35            动作内容4_数值 = Val( Mid$ (动作内容4, StrReverseFind(动作内容4, ":")+2) )
36            设备0_读写DWUB0000 = 命令内容_数值
37            设备0_读写DDF0001 = 动作内容1_数值
38            设备0_读写DDF0003 = 动作内容2_数值
39            设备0_读写DDF0005 = 动作内容3_数值
40            设备0_读写DDF0007 = 动作内容4_数值
41            '判断“命令内容”非延时，将“命令内容_数值”写入PLC，同时将动作内容写入PLC
42        endif
43        j = j + 1 'While循环次数判定
44    ENDWHILE
45    ENDIF
46 使能 = 0 '按钮使能控制, 0允许控制, 1禁止控制
    
```

图 8.4-28 “执行全部指令”策略脚本内容

完成点胶指令执行组态，下面进行使能功能组态：

① 使能组态：使能功能组态不属于必要功能，主要为了防止用户在点胶指令执行过程时误操作。参考图 8.4-26、图 8.4-28，脚本的开头将变量“使能”赋值为 1，脚本执行结束后将变量“使能”赋值为 0，我们可以在用户窗口“点胶操作”中所有标准按钮构件的“安全属性”页面关联变量“使能”，使点胶指令在执行时，对应的标准按钮构件失效。如图 8.4-29 所示。



图 8.4-29 使能功能组态

完成点击质量执行组态，下面组态 PLC 通道值显示：

① PLC 通道值显示组态：如图 8.4-30 所示，在通道值所在行的标签构件中勾选“显示输出”，并依次关联变量“设备 0_读写 DWUB0000”、“设备 0_读写 DDF0001”、“设备 0_读写 DDF0003”、“设备 0_读写 DDF0005”、“设备 0_读写 DDF0007”，方便工程运行时显示当前 PLC 接收的指令数据。

PLC地址	命令内容 DWUB0000	动作1 DDF0001	动作2 DDF0003	动作3 DDF0005	动作4 DDF0007
通道值	0	0	0	0	0

图 8.4-30 PLC 通道值显示组态

② 以上操作步骤全部完成，即可连接触摸屏和 PLC，再将工程下载到触摸屏中运行。通过点击“插入”或“新增”按钮，在配方构件中插入配方（指令），再选中插入的指令，点击点胶操作中的点胶指令，写入用户需要的点胶指令（如点胶开始、X 轴运动等），最后点击“保存”按钮保存指令内容，再点击“执行选中指令”或“执行全部指令”，将指令下方到 PLC，效果如图 8.4-31 所示。

通讯状态 0

点胶机工艺流程设置

序号	命令内容	动作内容1	动作内容2	动作内容3	动作内容4
0	点胶开始				
1	延时(ms)	延时(ms): 1000.00			
2	X轴运动	X: 2.00	速度: 3.00		
3	延时(ms)	延时(ms): 2000.00			
4	绝对运动	X: 20.00	Y: 15.00	Z: 10.00	速度: 8.00
5	延时(ms)	延时(ms): 3000.00			
6	XY轴回原				
7	延时(ms)	延时(ms): 1500.00			
8	Z轴回原				
9	延时(ms)	延时(ms): 800.00			
10	点胶停止				

点胶操作

点胶开始

X轴运动

Y轴运动

X+Y轴运动

延时

XY轴回原

Z轴运动

Z轴回原

相对运动

绝对运动

点胶停止

PLC地址	命令内容 DWUB0000	动作1 DDF0001	动作2 DDF0003	动作3 DDF0005	动作4 DDF0007
通道值	0	0	0	0	0

插入

新增

删除

保存

导入

导出

执行选中指令

执行全部指令

公司信息

图 8.4-31 点胶机工艺流程效果展示

第 9 章 生产工艺流程预览应用实例

若用户需要在产品正式生产前，先进行轨迹、图形等曲线的预览，减少实际生产试错的成本，可以使用 McgsPro 提供的 XY 曲线构件功能。

9.1 XY 曲线的基本属性和标注属性

- 首先介绍 XY 曲线构建的属性设置功能，其基本属性设置如图 9.1-1 所示：

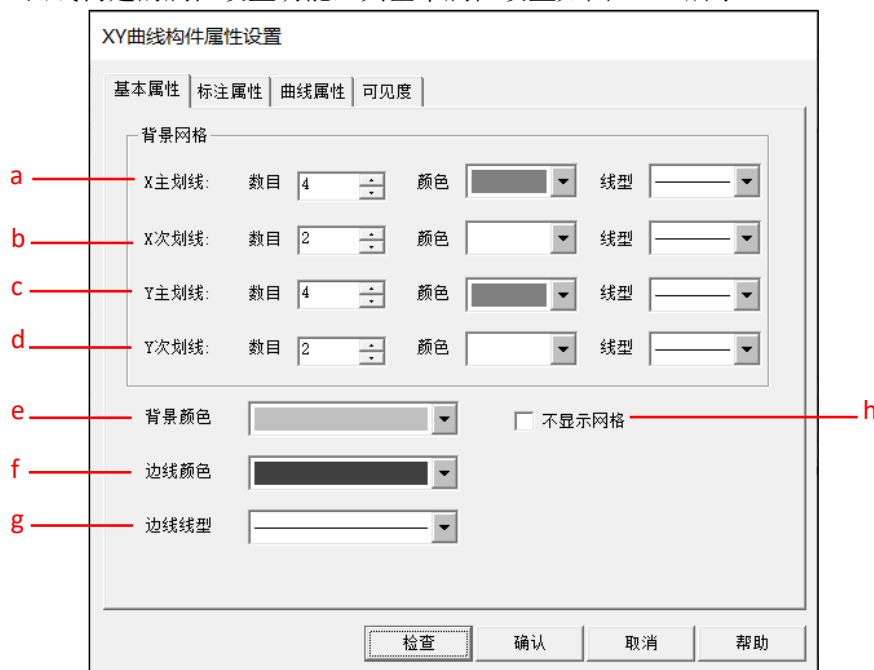


图 9.1-1 XY 曲线构件基本属性

基本显示配置信息：

- X 主划线：设置 X 主划线的数量、颜色和线型，取值范围为[1， 20]，超出范围时，检查不通过；
 - X 次划线：设置 X 次划线的数量、颜色和线型，取值范围为[1， 20]，超出范围时，检查不通过；
 - Y 主划线：设置 Y 主划线的数量、颜色和线型，取值范围为[1， 20]，超出范围时，检查不通过；
 - Y 次划线：设置 Y 次划线的数量、颜色和线型，取值范围为[1， 20]，超出范围时，检查不通过；
 - 背景颜色：设置构件的背景颜色，若选择没有填充时，则构件背景与窗体背景保持一致；
 - 边线颜色：设置构件的边线颜色，若选择没有边线时，则构件不绘制边线；
 - 边线线型：设置构件的边线线型，包括有 6 种不同的线宽。
 - 不显示网格：当勾选时，XY 主次划线设置不生效，构件不绘制 XY 主次划线；
- XY 曲线构建标注属性设置如图 9.1-2 所示：



图 9.1-2 XY 曲线构件标注属性

标注属性配置信息：

- a. X 轴标注颜色：设置 X 轴标注颜色；
- b. X 轴标注字体：设置 X 轴标注的字体；
- c. X 轴标注间隔：设置 X 轴标注间隔，取值范围为[1，20]；
- d. X 轴标注小数位数：设置 X 轴标注的小数位数，取值范围为[0，6]；
- e. X 轴最小值：设置 X 轴最小值，取值范围为[-FLT_MAX，FLT_MAX]，FLT_MAX=3.402823466e+38F，下同；
- f. X 轴最大值：设置 X 轴最大值，取值范围为[-FLT_MAX，FLT_MAX]；
- g. 不显示 X 轴坐标标注：当勾选时，X 轴标注设置无效，构件不显示 X 轴标注；
- h. Y 轴标注颜色：设置 Y 轴标注颜色；
- i. Y 轴标注字体：设置 Y 轴标注的字体；
- j. Y 轴标注间隔：设置 Y 轴标注间隔，取值范围为[1，20]；
- k. Y 轴标注小数位数：设置 Y 轴标注的小数位数，取值范围为[0，6]；
- l. Y 轴最小值：设置 Y 轴最小值，取值范围为[-FLT_MAX，FLT_MAX]；
- m. Y 轴最大值：设置 Y 轴最大值，取值范围为[-FLT_MAX，FLT_MAX]；
- n. 不显示 Y 轴坐标标注：当勾选时，Y 轴标注设置无效，构件不显示 Y 轴标注。

9.2 XY 曲线的曲线属性

XY 曲线构件有两个功能，一是用于绘制二维坐标数据<x, y>，便于直观地观察 y 随 x 的变化情况，二是 X 轴按某一正整数递增，反映一组数据的变化趋势。曲线数据有三种来源：历史数据，CSV（文件或字符串），字符串；支持 8 条曲线同时绘制，每条曲线最大可加载并显示 20480 个数据点。

9.2.1 XY 曲线_历史数据

XY 曲线历史数据通过关联存盘的组对象，设置起点时间与终点时间限制曲线显示的时间段，X 值和 Y 值均选择组对象里的成员，运行时通过“窗口 0.控件 0.Refresh()”刷新脚本可将构件绘制的曲线显示出来。

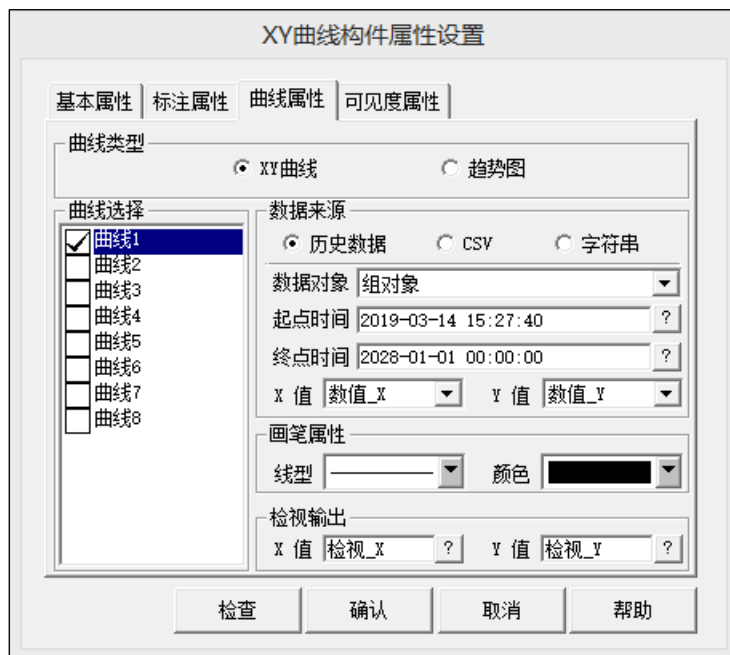


图 9.2-1 数据来源为 XY 曲线_历史数据

检视输出：当检视输出 X 值和 Y 值分别关联上变量（开关型或者数值型），运行时将指针定位在 XY 曲线构件任意位置，将会出现横向和纵向交叉的黑直线，交叉点的 XY 值会输出给对应的变量，并且可以通过标签或输入框显示出来。这里检视输出默认显示 6 位有效数字。

注意：红线为绘制的历史曲线，黑线为检视输出的基准线。

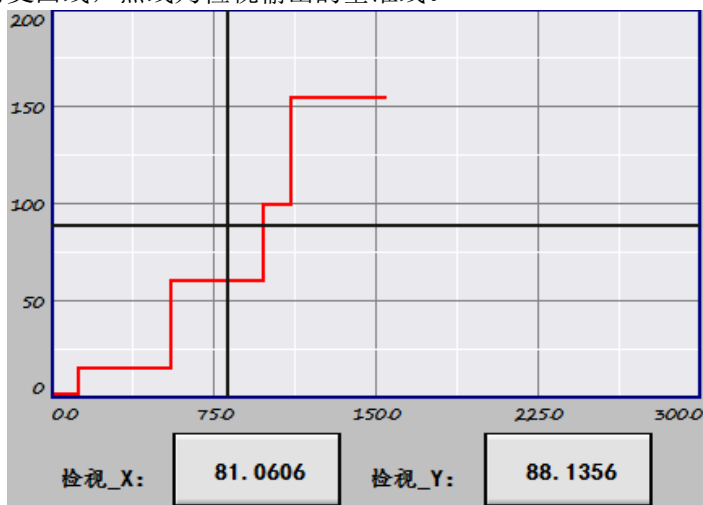


图 9.2-2 数据来源为 XY 曲线_历史数据效果图

9.2.2 XY 曲线_CSV

XY 曲线 CSV 中包含文件和字符串，文件中文件路径可以关联字符型变量，变量内容为 CSV 文件的路径。也可手动填入，此时有两种情况，第一种为“\$MCGS_DIR_USER/文件名.csv”，模拟器时对应路径为安装路径下/data/user_dir/文件名.csv；第二种为“/文件名.csv”或者“文件名.csv”，触摸屏路径为 U 盘/文件名.csv，模拟器时对应路径为安装路径下/export/文件名.csv。起始行号、X 所在列、Y 所在列这三项可写常量或者关联整数型的变量（关联整数型变量运行时需输入数据才可绘制出曲线）。这里将参数设置为固定路径和常量。

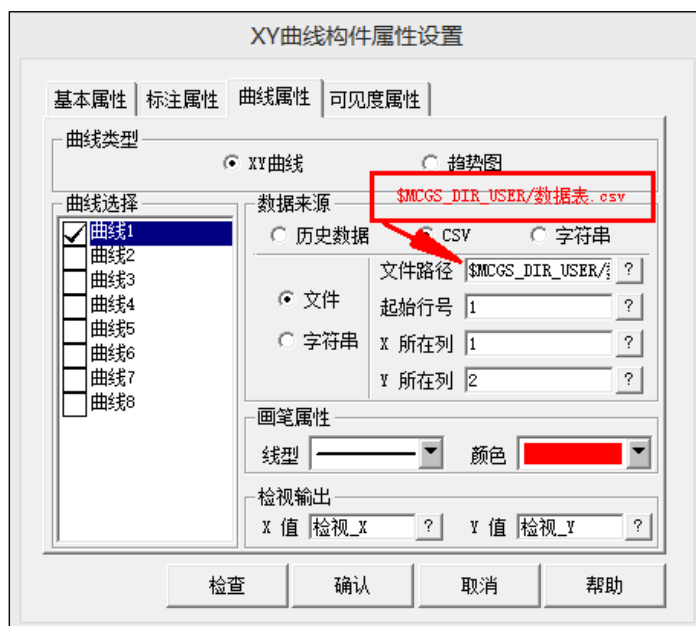


图 9.2-3 数据来源为 XY 曲线_CSV_文件

在“组态软件安装路径\Program\data\user_dir”文件夹下建立“数据表.csv”文件。表格内的内容设置方式如下：

A	B
1	1
20	200
50	500
70	700
80	800
100	1000

图 9.2-4 数据表.CSV

此时运行工程，将会绘制出如下的曲线（若起始行号、X 所在列、Y 所在列关联的是变量，那么在设置好对应参数后，需用“窗口 0.控件 0.Refresh()”来刷新曲线）。

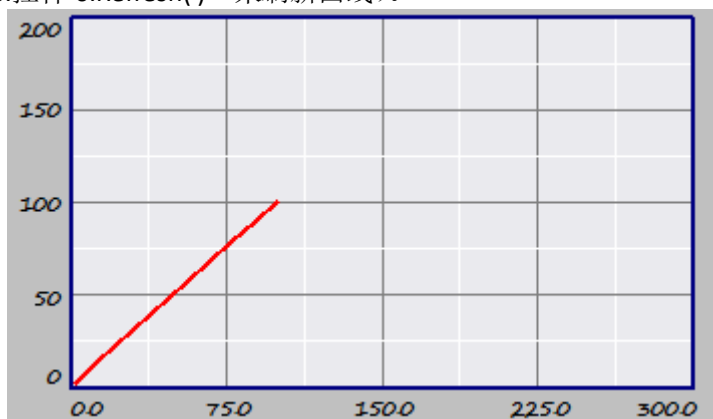


图 9.2-5 数据来源为 XY 曲线_CSV_文件效果图

XY 曲线/CSV/字符串中，数据变量关联字符型变量“csv_字符”，起始行号、X 所在列、Y 所在列按照下图设置，将数据变量关联变量“csv_字符”，该字符型变量中的内容可由关联了变量“csv_字符”的输入框输入。



图 9.2-6 数据来源为 XY 曲线_CSV_字符串

注意：运行时弹出的软键盘里面没有换行符，所以只能通过复制表格里面的数据生成曲线，无法直接手动在输入框输入数据。

	A
1	20,40
2	30,140
3	31,140
4	50,40
5	51,40
6	80,140
7	81,140
8	90,40

图 9.2-7 输入框的数据表格

通过编写脚本的方式亦可绘制曲线，McgsPro 支持使用“\N”或!I2Ascii(13)+!I2Ascii(10)表示换行。注意，若脚本使用“\N”换行，则需要加双引号，若使用!I2Ascii(13)+!I2Ascii(10)，则不用加双引号。

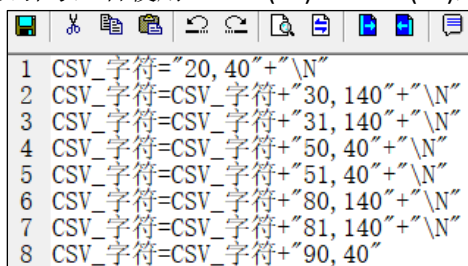


图 9.2-8 使用“\N”给脚本换行

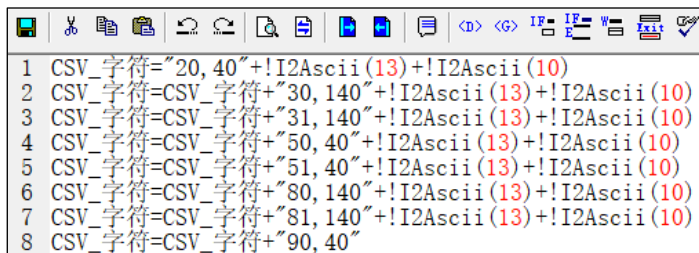


图 9.2-9 使用!I2Ascii(13)+!I2Ascii(10)给脚本换行

9.2.3 XY 曲线_字符串

在XY曲线中，勾选“字符串”，X值和Y值可以取自不同的字符串，也可以取自同一字符串，用户可根据需求来配置。这里使用取自同一字符串，关联“XY字符串”变量。

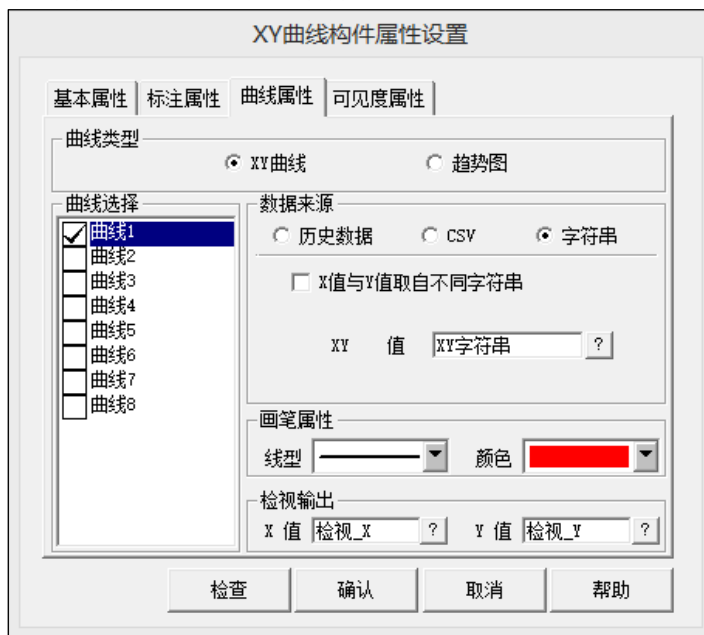


图 9.2-10 数据来源为XY曲线_字符串

用户可以直接在输入框中输入使用英文逗号相隔开的一串数字，第一个数字是X的值，第二个数字是Y的值，以此类推。也可以通过编写脚本的方式来实现此功能，数字之间使用“，”或!I2Ascii(13)+!I2Ascii(10)来分隔。

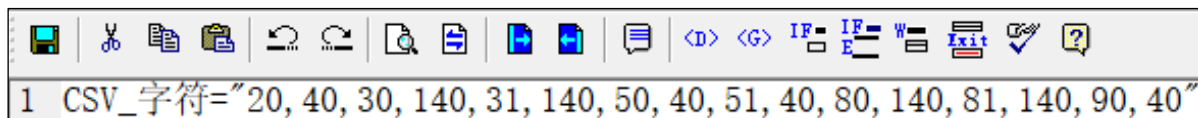


图 9.2-11 使用“，”分隔字符串

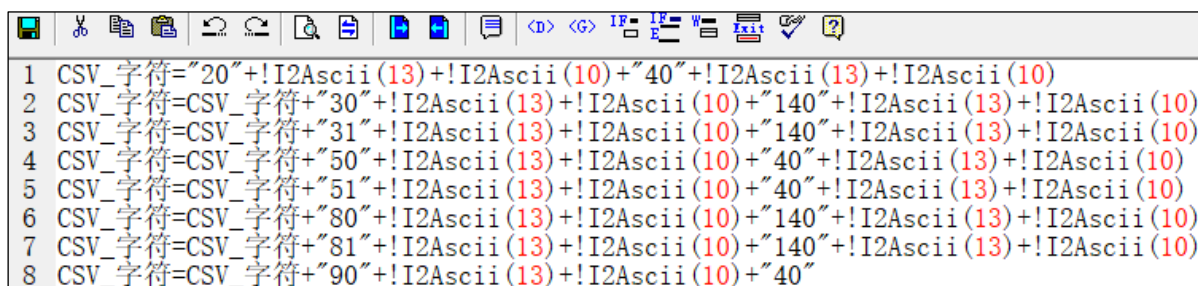


图 9.2-12 使用!I2Ascii(13)+!I2Ascii(10)分隔字符串

9.2.4 趋势图_历史数据

趋势图/历史数据与XY曲线/历史数据区别在于，前者引入了新的概念--步进，后者的X值只能关联组对象内的变量，步进则没有这方面的限制，可任意关联一个整数型变量，也可输入常量，Y值为组对象内的变量。

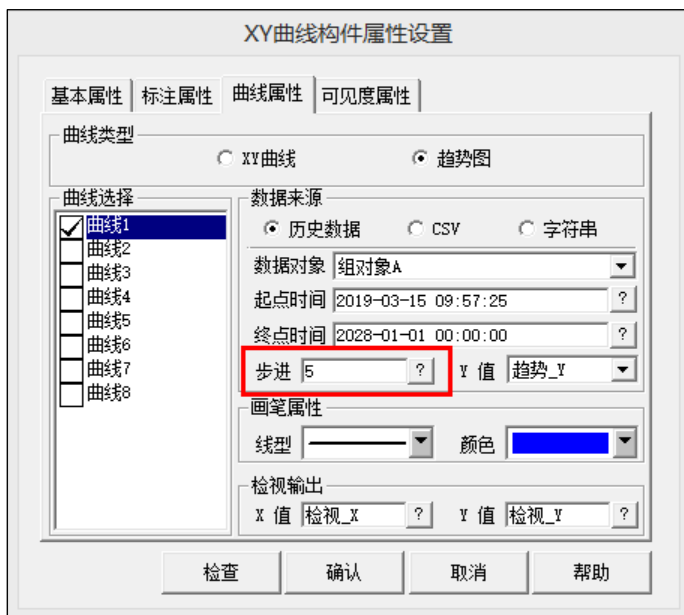


图 9.2-13 数据来源为趋势图_历史数据

检视输出：当检视输出 X 值和 Y 值分别关联上变量（整数型或者浮点数值型），运行时将指针定位在 XY 曲线任意位置，出现纵向的黑直线，与曲线交叉点的 XY 值会输出给对应的变量，并且可通过标签或输入框显示出来，检视输出默认显示整数。

注意：蓝线为绘制的历史曲线，黑线为检视输出的基准线。

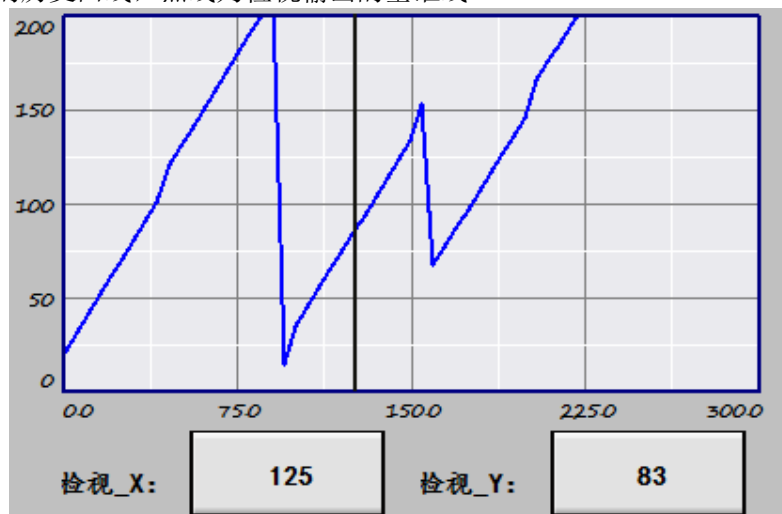


图 9.2-14 数据来源为趋势图_历史数据效果图

9.2.5 趋势图_csv

数据来源为趋势图_csv 与 XY 曲线_csv 做法相同，这里不作介绍，用户可参见“XY 曲线_csv”。

9.2.6 趋势图_字符串

趋势图中字符串的步进值可输入常量或者关联整数型变量，Y 值根据步进值变化。Y 值的设置请参见“XY 曲线_字符串”。

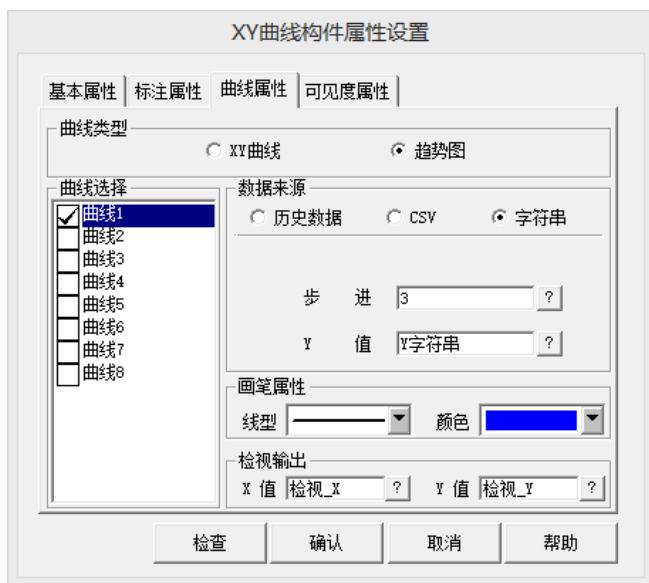


图 9.2-15 数据来源为趋势图_字符串

9.3 XY 曲线的方法函数

报警浏览构件方法函数解释：

- **Refresh():** 刷新，在运行过程中若是数据改变，可调用此函数刷新 XY 曲线构件的显示。
组态方法：新建一个【标准按钮】构件，将它的【基本属性】中的【文本】设置为【刷新】，抬起脚本设置为【窗口 0.控件 0.Refresh()】。注意，其中【窗口 0】是 XY 曲线构件所在窗口名，【控件 0】是 XY 曲线构件的名称，点击 XY 曲线构件后，右下角的信息栏显示构件名称。构件方法的详细设置步骤请参考章节：进阶篇 3.5.4 构件方法。

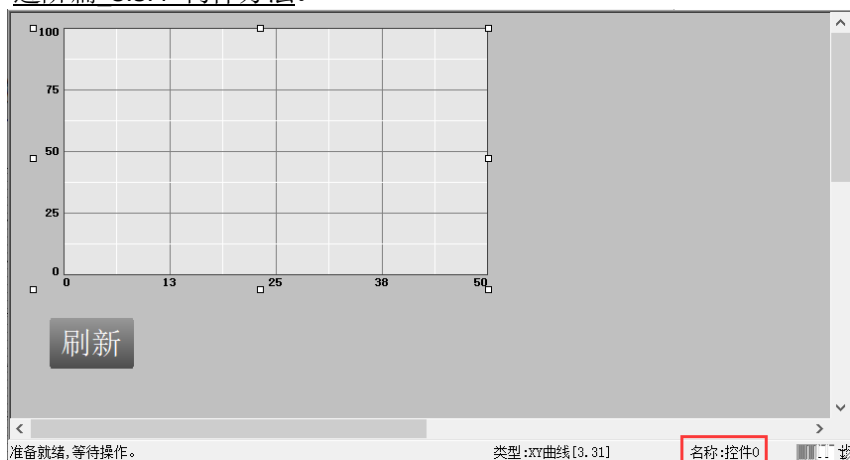


图 9.3-1 获取控件名称

- **ClearData(Index):** 清除曲线数据
注意事项: 调用该函数只是清除构件上的显示，并没有清除实际存储的数据
- **AddXYDataFromFile(Index, Count, Xcolumn, Ycolumn, StrFilePath, StartLine, XStep):** 从文件中加载数据
- **AddXYDataFromCSVString(Index, Count, Xcolumn, Ycolumn, StrCSVString, StartLine, XStep):** 从 CSV 字符串中加载数据
- **AddXYDataFromGroupObject(Index, Count, GroupObject, VarX, VarY, StrStartTime, XStep):** 从存盘数据中加载数据
- **AddXYDataFromString(Index, Count, StrVarX, StrVarY, StrVarXY, XStep):** 从字符串中加载数据
- **SetTrendRange(Index, Min, Max):** 设置特定曲线的上下限值
- **GetTrendRange(Index, Min, Max):** 取特定曲线的上下限值
- **SetTrendVisible(Index, Flag):** 置特定曲线的可见度
- **GetTrendVisible(Index):** 取定曲线的可见度

- **GetXRange(Min,Max):** 取 X 轴最小值和最大值
- **SetXRange(Min,Max):** 置 X 轴的最小值和最大值
- **ShowYaxis(Index):** 显示指定曲线的 Y 轴坐标标注
- **SetDrawMode (Index,Mode):** 设置曲线绘制模式

9.4 XY 曲线的注意事项

- XY 曲线的曲线是可以来回变化的，而趋势图的曲线是单调往右变化。
- 步进值若设置为常量，不能为负数。
- 曲线的 X 轴固定不动，若需要移动，可使用 **SetXRange(Min, Max)**函数来设置 X 轴的最大值和最小值。

```

1 趋势 Y=趋势 Y+10
2 趋势图=趋势图+10
3
4 IF 趋势图>100 THEN
5     窗口6.控件1.SetXRange(趋势图-100, 趋势图)
6 ENDIF
7
8 窗口6.控件1.Refresh()
    
```

图 9.4-1 SetXRange(Min, Max)函数的使用

第 10 章 脚本程序功能介绍

10.1 脚本知识概要

脚本程序是组态软件中的一种内置编程语言引擎。在 McgsPro 组态件中，脚本语言是一种语法上类似 Basic 的编程语言，有些 HMI 软件中也叫做“宏指令”。

脚本可以在运行策略、窗口启动脚本、窗口循环脚本、窗口退出脚本、窗口事件脚本、构件事件脚本、按钮脚本中运用。

10.1.1 编辑环境介绍

脚本程序编辑窗口是用户进行脚本编写最主要的地方，操作界面分为以下几个部分：

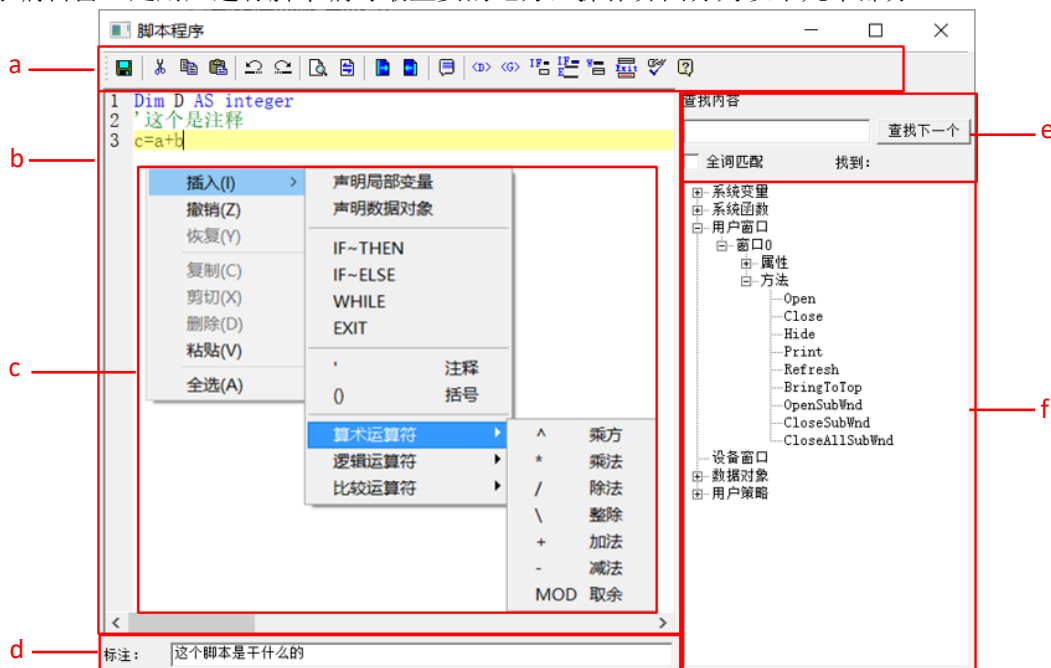


图 10.1-1 脚本程序编辑窗口

- 工具条按钮：请参考章节：[进阶篇 1.3.5 脚本程序窗口工具条](#)。
- 脚本编辑框：脚本编辑框是用户进行脚本编写最主要的地方，所有脚本内容均显示在此处。脚本编辑框支持自动完成功能。自动完成功能主要是在用户输入内容时自动提示与用户输入相关的内容，辅助用户快速完成输入。比如 McgsPro 中以“!AS”开头的函数有两个，当用户在脚本编辑框中输入“!AS”时会自动提示用户匹配对象（如图 10.1-2 所示），用户这时可以按上下键选择想输入的内容后按 enter 键直接快速输入。

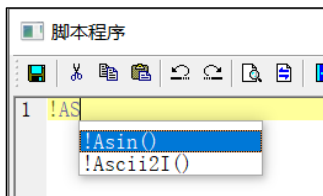


图 10.1-2 脚本自动完成

当用户通过自动未完成提示快速输入系统函数后，系统函数需要填入参数，此时用户可以将鼠标移动到已经输入的系统函数上，此时编辑框会自动对此脚本函数的功能、参数类型和返回值类型进行提示，如图 10.1-3 所示。

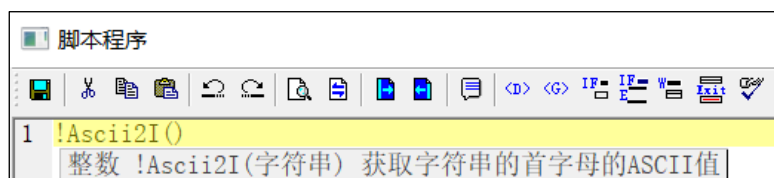


图 10.1-3 函数提示

- c. 右键菜单：脚本编辑框中点击鼠标右键弹出操作菜单，可以进行一些语法或操作符的快捷输入。
- d. 脚本注释框：策略特有标注信息，注释脚本的作用等。
- e. 查找功能区：查找范围为系统变量、系统函数、窗口名称、构件名称、设备名称、变量名称、策略名称等。注：不包含属性、方法。分为非全词匹配和全词匹配两种。
- f. 对象树：对象树以树结构的形式，列出了工程中所有的窗口、策略、设备、变量、系统支持的各种方法、属性以及各种函数，用户可以在此快速查找。对象树只有在脚本编辑窗口中有效。

10.1.2 语言要素介绍

10.1.2.1 变量

- 数据对象：相当于全局变量，在所有的程序段共用。它可以在实时数据库中定义，也可以通过脚本框上方的【声明数据对象】<G> 图标按钮定义，如图 10.1-4 所示。脚本编辑中可以用数据对象的名称来读写数据对象的值，也可以对数据对象的属性进行操作，支持整数、浮点数、字符串三种数据对象。

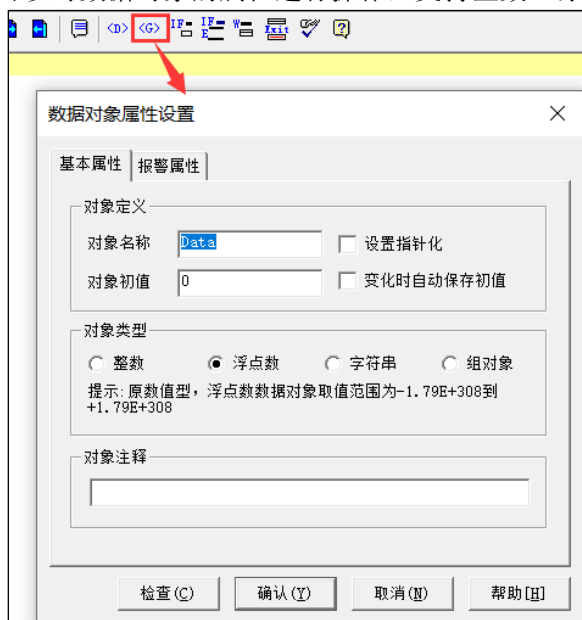


图 10.1-4 在脚本程序编辑窗口中声明数据对象

- 局部变量：支持整数、浮点数、字符串、字节型四种数据类型，只能在当前脚本中使用。可直接输入脚本“Dim...As...”语句对局部变量进行声明，或使用工具栏上的【声明局部变量】<D> 图标按钮进行声明。需要注意的是，声明语句不能嵌套在其他任何语句中。

```

17 Dim D1 AS integer' 定义一个整数型变量D1
18 D1=123
19
20 Dim D2 AS float' 定义一个浮点型变量D2
21 D2=12.3
22
23 Dim D3 AS string' 定义一个字符串型变量D3
24 D3="昆仑通态"
25
26 Dim D4 AS byte' 定义一个字节型变量D4
27 D4=1
    
```

图 10.1-5 在脚本程序编辑窗口中声明局部变量

- 局部数组变量：支持整数数组、浮点数数组、字符串数组、字节型数组四种数据类型，只能在当前脚

本中使用, 声明方法同【局部变量】。定义数组变量最大长度是 65535, 其访问元素的方式为 array[index], 其中“array”为数组变量、“index”为访问元素的位置 (从 1 开始), 如图 10.1-6 所示。

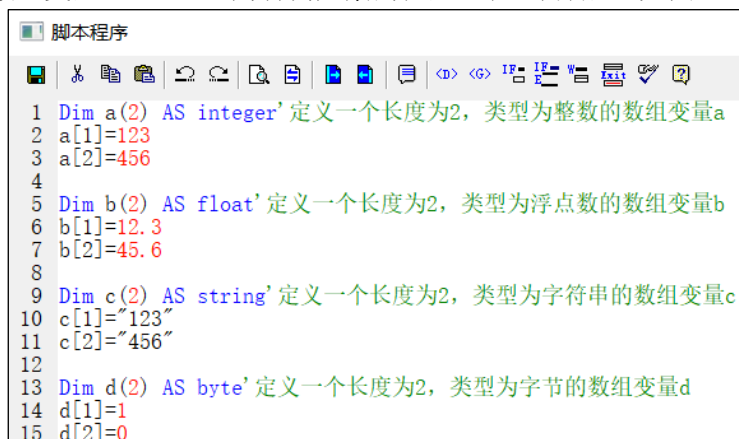


图 10.1-6 在脚本程序编辑窗口中声明局部数组变量

10.1.2.2 常量

- 整数常量: 类似 123 的整数
- 十六进制整数常量: 类似 0x123 的十六进制整数
- 浮点数常量: 带小数点的数值, 如: 12.45
- 字符串常量: 英文双引号内的字符串, 如: "OK", "正常"
- 字符串常量内支持以下转义:
 - “\n”: 代表换行
 - “\r”: 需要和“\n”一起使用 (“\r\n”), 在微软环境中代表换行
 - “\””: 代表双引号
 - “\t”: 代表制表符
 - “\\”: 代表反斜杠自身其他以“\”开始的字符将作为非法字符串。

10.1.2.3 系统变量

McgsPro 组态定义的内部数据对象作为系统内部变量, 在脚本程序中可自由使用, 在使用系统变量时, 变量的前面必须加“\$”符号, 如 \$Date。系统变量为只读类型, 在脚本编辑环境中, 点击展开“对象树”中的“系统变量”, 再双击变量名称即可添加到编辑框中。系统变量的详细内容将在后续章节: [进阶篇 11.8 系统变量](#)进行介绍。

10.1.2.4 属性和方法

属性和方法函数的详细内容请参考章节: [进阶篇 3.5 构件或窗口的属性和方法](#)。

10.1.2.5 表达式

由变量、括号和各种运算符组成的运算式称为表达式, 表达式的计算结果称为表达式的值。例如: a=b+c; str="aa"+"bb"。

10.1.2.6 运算符

分类	名称	运算符
算术运算符	乘方	^
	乘法	*
	除法	/
	整除	\
	加法	+
	减法	-
	取余	MOD

分类	名称	运算符
逻辑运算符	与	AND
	非	NOT
	或	OR
	异或	XOR
比较运算符	大于	>
	大于等于	>=
	等于	=
	小于等于	<=
	小于	<
	不等于	<>

10.1.2.7 优先级

优先级	运算符
高	()
	^
	* / \ MOD
	+ -
	< > <= >= = <>
	NOT
	AND OR XOR
低	

图 10.1-7 运算符优先级

10.1.2.8 注意事项

所有能够在脚本中使用的对象名称（如窗口、变量、策略），不能用以下的关键字命名：

- if、then、else、endif、while、break、endwhile、and、or、xor、not、exit、break、byte、integer、single、float、string。
- 不能以数字开头
- 不能含有以下的特殊字符：
- `~!@#\$%^&*()-=+\|}][{\'";/?.>,<~!·#¥%.....—*（）——+|？，。《》/。
- 不能为空
- 不能为“_”

10.2 基本语句规则及应用

脚本程序是为了实现流程的控制及对象操作处理，包括如下语句：

赋值语句、条件语句、循环语句、退出语句、注释语句、声明语句、跳出语句，如图 10.2-1 所示。



图 10.2-1 语句种类

10.2.1 赋值语句

赋值语句的格式为：数据对象 = 表达式。

赋值号（等号）左边必须是能够读写的数据对象。

赋值号的右边为表达式，表达式的数据类型必须与左边数据对象的值的类型相符合，否则系统会提示“类型不匹配”的错误信息。

假设变量A为整数变量，B为浮点数变量，C为浮点数变量，D为字符串变量		错误赋值格式	不常用格式
A=1	设置整数变量A为ON	A=B=10	A=10:B=10
B=5	设置浮点数变量B为5	A=10;B=10	
B=C/2	把C的值除以2后赋值给B	A=10,B=10	
D= ""	设置字符串变量D为空	\$Year=1	
D= "深圳"	设置字符串变量D显示内容为深圳		

图 10.2-2 应用举例

10.2.2 条件语句

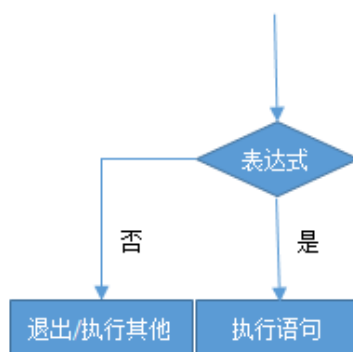


图 10.2-3 条件语句逻辑

“IF”语句的表达式一般为逻辑表达式，当表达式的值为非0时，条件成立，执行“Then”后的语句，否则，条件不成立，将不执行条件块中包含的语句，开始执行条件块后面的语句。

条件语句有如下三种形式：

形式一：

If 〔表达式〕 Then 〔赋值语句或退出语句〕

形式二：

If 〔表达式〕 Then

〔语句〕

EndIf

形式三：

If 〔表达式〕 Then

〔语句〕

Else

〔语句〕

EndIf

条件语句中的四个关键字“if”、“Then”、“Else”、“Endif”不分大小写。如拼写不正确，检查程序会提示出错信息。

条件语句允许多级嵌套，即条件语句中可以包含新的条件语句。

值为字符串的表达式不能作为“if”语句中的表达式。

- 使用条件语句组态脚本实现：点击启动按钮后，传送带上的物体向左传送。
 - ① 首先在实时数据库中新建两个整数型变量【传送带移动】和【传送带启动】。
 - ② 进入用户窗口【窗口0】，点击【工具栏】→【插入元件】→在弹出的元件图库管理窗口中选择类型为【公共图库】→【传送带】→【传输带5】→确定。将传送带拖拽到画面合适的位置。

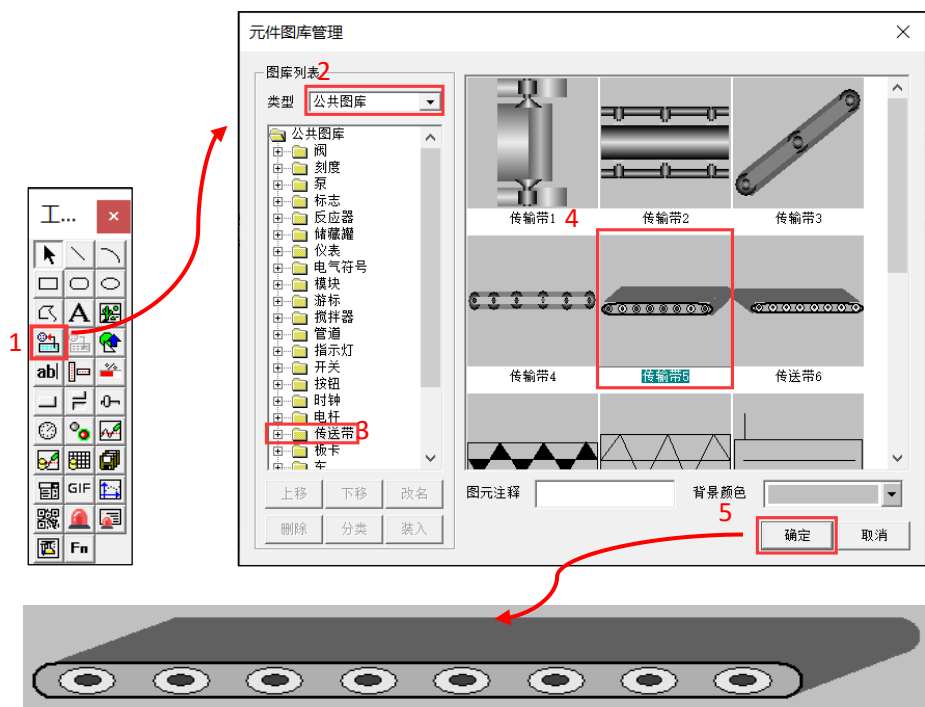


图 10.2-4 新建一个传送带

- ③ 点击【工具栏】中的【常用符号】→【立方体】→拖拽到传送带靠右端合适的位置。

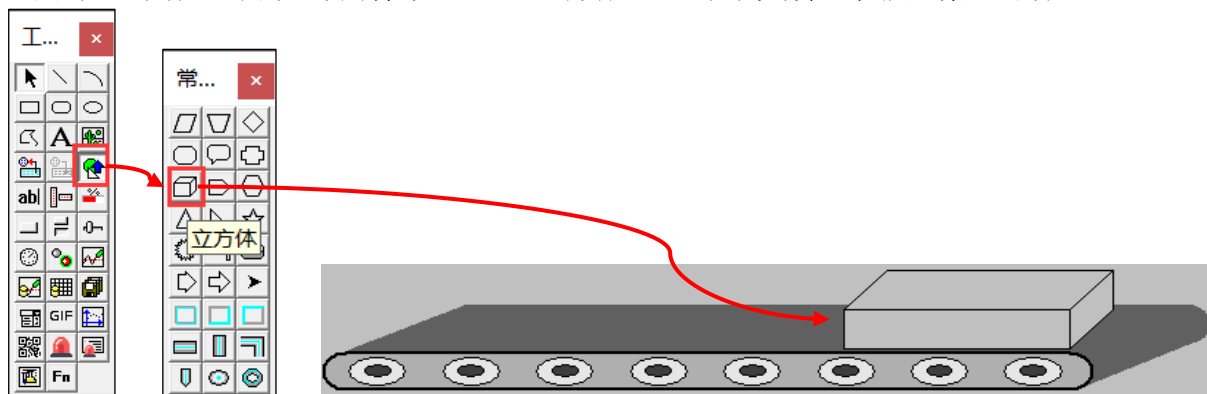


图 10.2-5 新建传送带上运载的物件

- ④ 双击画面窗口中的【立方体】，在弹出的动画组态属性设置界面中点击【基本属性】→勾选【水平移动】前的复选框→点击出现的【水平移动】功能页→设置【表达式】为【传送带移动】，【最小移动偏移量】为0，对应的表达式的值为【0】，【最大移动偏移量】为-300，对应的表达式的值为【100】。注意：偏移量是以组态时图形对象所在的位置为基准（初始位置），单位为像素点，向左为负方向，向右为正方向。通过“最小移动偏移量”及其“表达式的值”和“最大移动偏移量”及其“表达式的值”可以得到“表达式”值与偏移量之间的线性映射关联，当“表达式”为某一个值时，即可根据此线性关系计算出实际的偏移量从而定位图形。

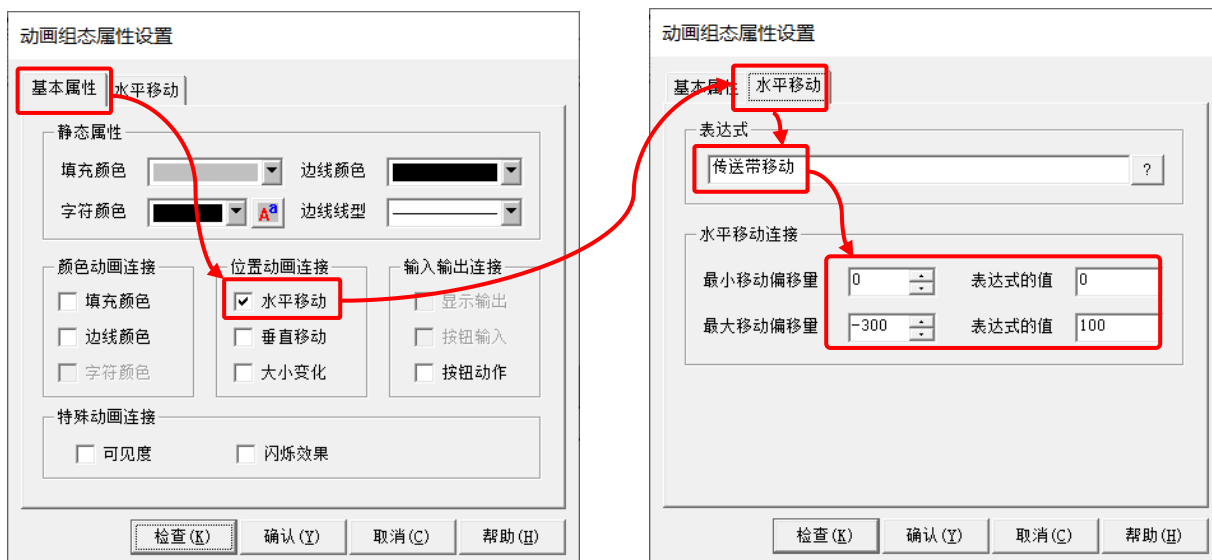


图 10.2-6 组态传送带所运载物件的水平移动

- ⑤ 从工具箱拖拽一个【标准按钮】构件到窗口画面中。双击该构件，弹出标准按钮构件属性设置窗口 → 【基本属性】 → 修改文本为【启动】 → 【操作属性】 → 关联【数据对象值操作】【置1】为【传送带启动】 → 确认。

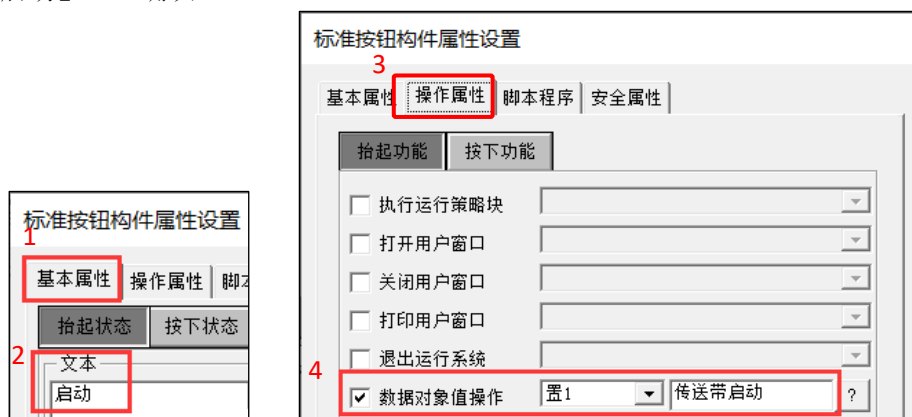


图 10.2-7 组态启动按钮

- ⑥ 进入【运行策略】窗口 → 【新建策略】 → 【循环策略】 → 双击出现的策略 1，进入策略组态窗口。

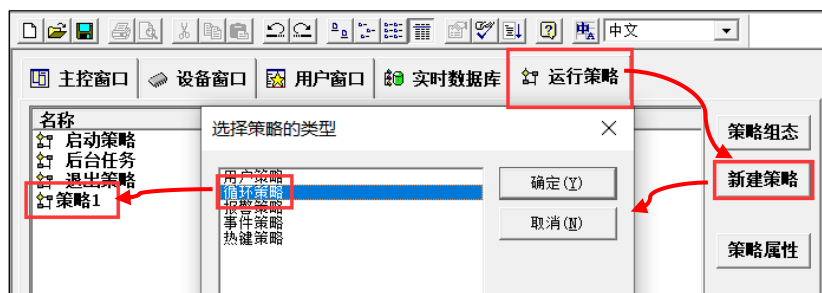


图 10.2-8 新建循环策略 1

- ⑦ 在策略组态窗口中，修改策略 1 的【循环时间】为 100ms。然后新增一行策略行，编辑策略脚本如图 10.2-9 所示。然后保存脚本，关闭脚本程序编辑窗口，保存策略 1。

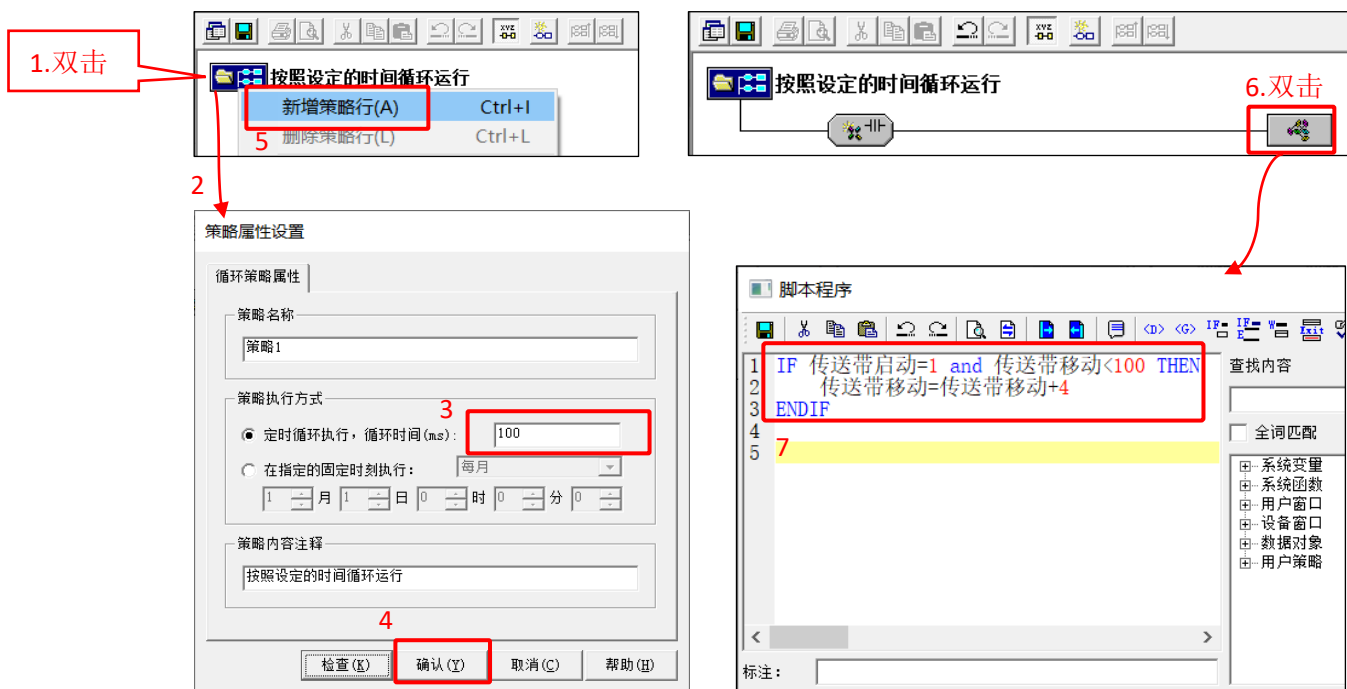


图 10.2-9 组态循环策略

⑧ 保存工程后下载工程，然后启动运行。点击【启动按钮】，传送带上的物料由右向左移动。

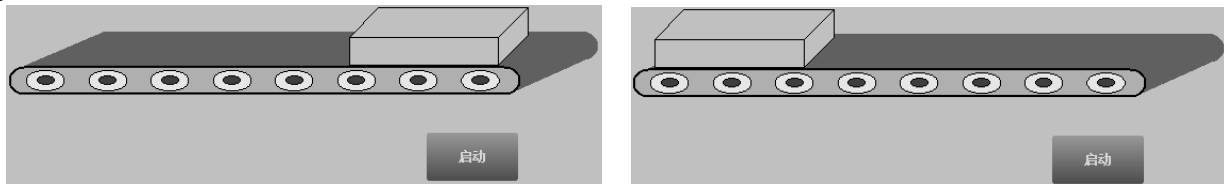


图 10.2-10 模拟运行效果

10.2.3 循环语句

循环语句为 WHILE 和 ENDWHILE，其结构为：

WHILE 〔条件表达式〕

...

ENDWHILE

当条件表达式成立时（非零），循环执行 WHILE 和 ENDWHILE 之间的语句。直到条件表达式不成立（为零）时退出。

- 通过以下脚本使 MAX 变量的值为数组中的最大值。

```
1 Dim 数组(100) AS float
2 Dim i AS integer
3 Dim MAX AS float
4
5 i=1
6 MAX=数组[1]
7
8 WHILE i<=100
9     IF MAX < 数组[i] THEN
10         MAX = 数组[i]
11     ENDIF
12 i=i+1
13 ENDWHILE
14
15 ' 经过这段代码计算后，MAX的值将为所有数组中的最大值
```

图 10.2-11 循环语句应用实例

10.2.4 注释语句

图 10.2-11 中的绿色文字就是注释语句，它在脚本运行过程中不执行，仅起到解释作用，可通过在语句前

输入英语单引号或点击工具栏的【注释】图标按钮实现。

10.2.5 跳出语句

跳出语句为“Break”，用于跳出当前循环，必须在循环语句中使用。

10.2.6 退出语句

退出语句为“EXIT”，用于中断脚本程序的运行，停止执行其后面的语句。一般在条件语句中使用退出语句，以便在某种条件下，停止并退出脚本程序的执行。

10.2.7 声明语句

脚本格式为“Dim...As...”，具体用法请参考章节：[进阶篇 10.1.2.1 变量](#)。

10.3 脚本查错

脚本程序编制完成后，可点击工具栏上的【检查】图标按钮对程序代码进行检查，以确认脚本程序的编写是否正确。检查过程中，如果发现脚本程序有错误，则会返回相应的信息，以提示可能的出错原因，帮助用户查找和排除错误。如图 10.3-1 所示：

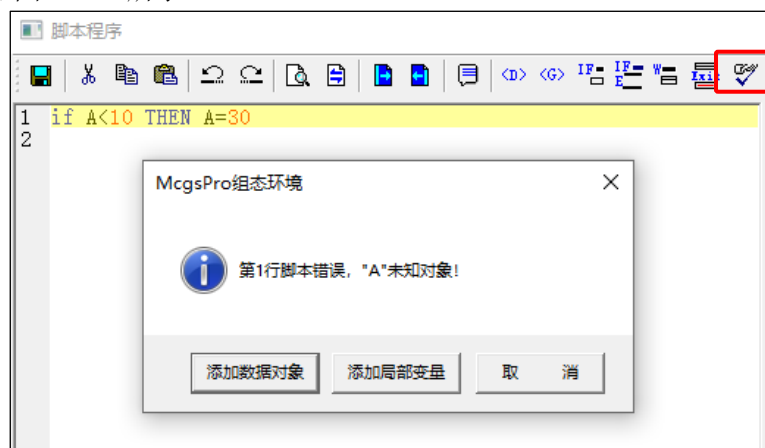


图 10.3-1 脚本检查报错

注：脚本程序里面未通过语法检查也能保存，但是无法下载。

常见的提示信息有：

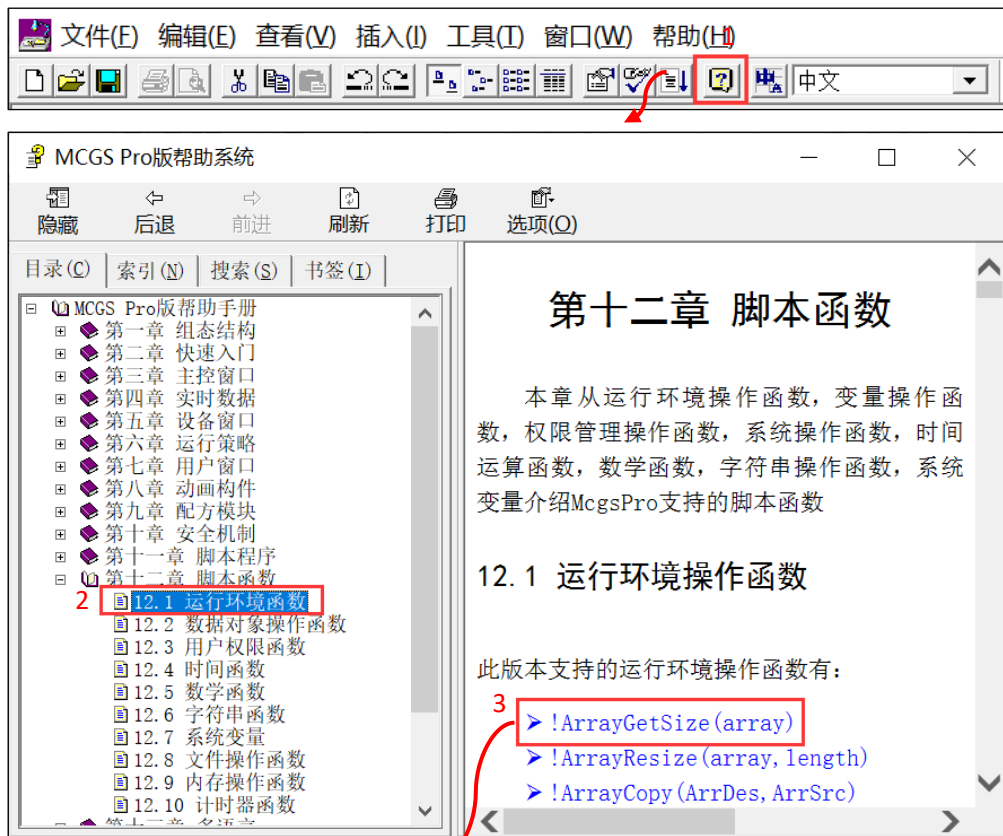
- 未识别的标识符：该标识符不是局部变量和全局可访问的对象
- 未识别的类型：变量声明时没有输入正确的类型
- 未填入参数：函数调用（需要参数）没有填入参数
- 需要一个逻辑值：在条件语句和循环语句中的条件判断中的表达式的值，必须是字节型、整数、浮点数，而不能是字符串和数组
- 需要引用参数：函数的当前参数类型为引用
- 需要一个数据对象：函数的当前参数类型为数据对象
- 局部变量声明错误：局部变量声明的格式错误
- 局部数组变量声明错误：局部数组变量声明的格式错误
- 局部数组变量声明长度非法：局部数组声明的初始长度错误，须在 0-65535 之间
- 下标访问类型不匹配：访问数组的元素时，下标类型错误；必须是字节/整数，最好是整数
- 无副作用的表达式：除了函数调用，一般的表达式需要一个赋值语句来使用表达式的结果
- 类型不匹配：一般是表达式的操作类型不匹配，如整数+字符串
- 参数个数不匹配：函数的参数个数不匹配
- 参数类型不匹配：函数的参数类型不匹配

第 11 章 脚本函数应用实例

本章主要介绍 McgsPro 的功能函数及其应用实例。McgsPro 的功能函数可分为运行环境函数，数据对象操作函数，用户权限函数，时间函数，数学函数，字符串函数，系统变量，文件操作函数，内存操作函数和计时器函数。

11.1 使用帮助引导

脚本函数的详细介绍和使用说明，可参考 McgsPro 组态软件帮助。以运行环境函数为例，点击工具栏上  按钮图标-目录-第十二章，找到运行环境函数，可查看各个函数的意义、参数、返回值、实例等的介绍。



!ArrayGetSize(array)

函数意义：获取数组的大小

返回值：整数

参 数：array，字节型数组/整型数组/浮点型数组/字符串数组

实 例：!ArrayGetSize(array)，获取数组array的大小

图 11.1-1 McgsPro 帮助使用方法

11.2 运行环境函数

McgsPro 中，涉及运行环境操作的函数大致可以分为 5 类：数组操作函数、运行策略函数、窗口操作函数、系统操作函数、操作日志函数。接下来为大家进行常用函数的介绍。

11.2.1 数组操作函数

11.2.1.1 常用函数介绍

下面 7 个函数是针对数组的操作函数，通过这些函数可以对数组进行设置大小、获取大小、复制、追加、添加元素等操作。数组的介绍请参考章节：[进阶篇 10.1.2.1 变量](#)。

!ArrayGetSize(array): 获取数组 array 的大小。

!ArrayResize(array,length): 设置数组 array 的大小为 length。

!ArrayCopy(ArrDes,ArrSrc): 把一个数组 ArrSrc 的内容复制到另外一个数组 ArrDes 中去，两者必须是同一类型，目标数组的大小也跟源数组一致。

!ArrayAppend(Arr1,Arr2): 把一个数组 Arr2 追加到另外一个数组 Arr1 的后面。要求两个数组类型一致，数组大小变为两个数组大小之和。

!ArrayIntAdd(Arr,n): 在一个整型数组 Arr 后面添加一个整数元素 n，数组大小加 1。

!ArraySingleAdd(Arr,x): 给一个浮点数组 Arr 添加一个浮点数元素 x，数组大小加 1。

!ArrayStringAdd(Arr,str): 给一个字符串数组 Arr 添加字符串元素 str，数组大小加 1。

11.2.1.2 样例演示

本节主要介绍：如何获取当前数组的大小且以及将数组大小重新设置。

- ① 使用 McgsPro 新建一个工程，进入用户窗口 0。然后新建一个按钮，将它的显示文本设为【获取数组大小】，脚本函数设置为：

Dim array(5) AS integer '定义一个长度为 5 的整数型数组

size=!ArrayGetSize(array) '获取数据【array】的大小，并将值赋给变量【size】



图 11.2-1 获取数组大小功能按钮设置步骤

由于没有提前在实时数据库中定义变量【size】，所以点击【确认】按钮时，会弹出对话框提示【size】是位置对象，询问【是否增加次对象？】。选择【是】，则弹出【数据对象属性设置】界面，界面的【对象名称】处自动填写未定义的变量名，选择【对象类型】为【整数】，点击确认即可。此时实时数据库中自动生成了一个名为【size】的【整数】型变量，如图 11.2-2 所示。

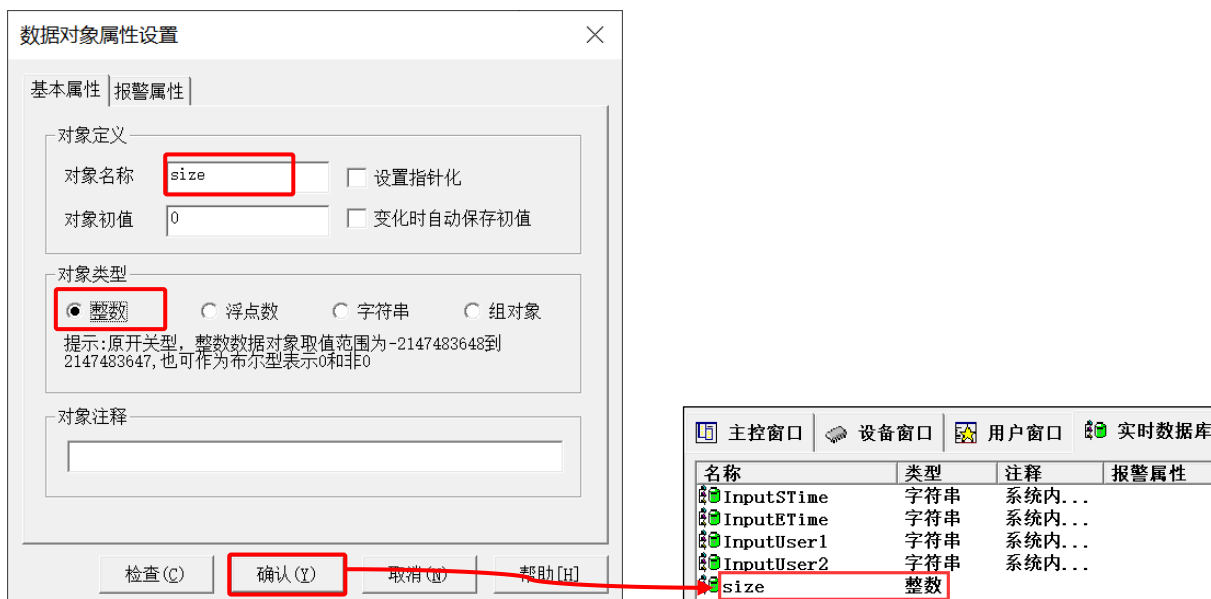


图 11.2-2 自动生成未定义变量

- ② 用同样的办法，再次新建一个按钮，将它的文本显示设置为【设置数组大小】，脚本函数设置为：

```
Dim array(5) AS integer    '定义一个长度为5的整数型数组
!ArrayResize(array, 7)    '将数组 array 的长度设置为7
size=!ArrayGetSize(array) '获取数据【array】的大小，并将值赋给变量【size】
```

由于数组是局部变量，只能在当前脚本中使用，所以在上一个按钮中定义过的数组在新按钮中已经失效，需要重新定义。

- ③ 然后新建一个输入框构件，用它显示【size】的值。
④ 组态完成后，保存工程，然后下载工程，下载成功后启动运行。

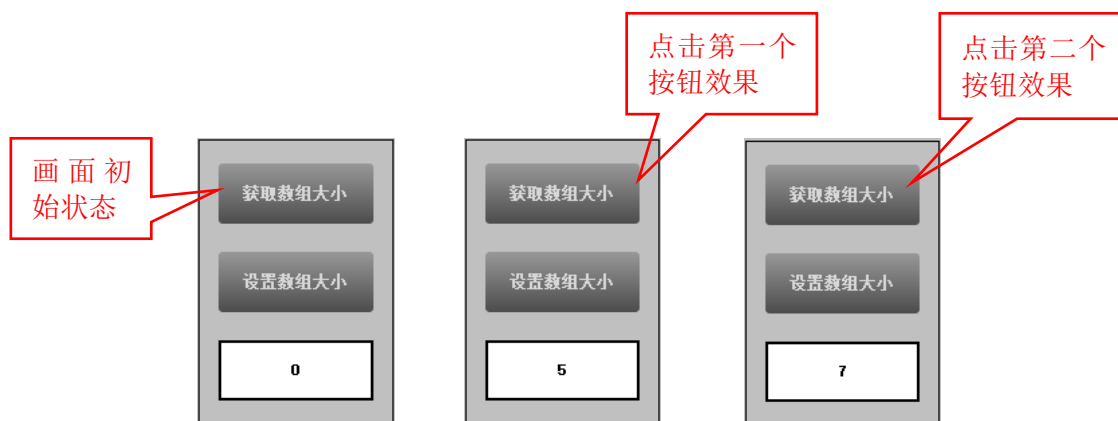


图 11.2-3 画面运行效果

11.2.2 运行策略函数

11.2.2.1 常用函数介绍

下面 3 个函数是针对运行策略的操作函数：

!ChangeLoopStgy(StgyName,n): 改变循环策略的循环时间。

!SetStgy(StgyName): 执行 StgyName 指定的运行策略。该函数是异步执行的策略，即：使用该函数时，函数中的策略 StgyName 和函数所在行的下一行脚本程序可以同时执行。

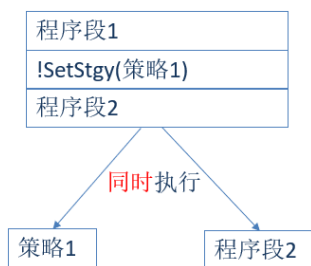


图 11.2-4 函数!SetStgy 执行机制

!SetStgyMode(StgyName): 以模态方式执行策略。该函数是同步执行的策略，即：使用该函数时，必须先执行完函数中的策略 StgyName 才会执行函数所在行的下一行脚本程序。

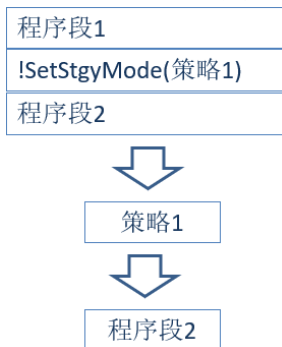


图 11.2-5 函数!SetStgyMode 执行机制

11.2.2.2 样例演示

组态一个样例，演示函数!SetStgy(StgyName)和!SetStgyMode(StgyName)运行时的区别。

- ① 进入实时数据库，新建整数型变量【A】和【B】。

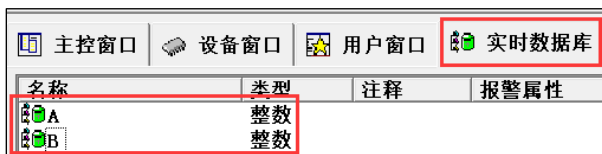


图 11.2-6 新建变量

- ② 进入运行策略窗口，新建一个【用户策略】。

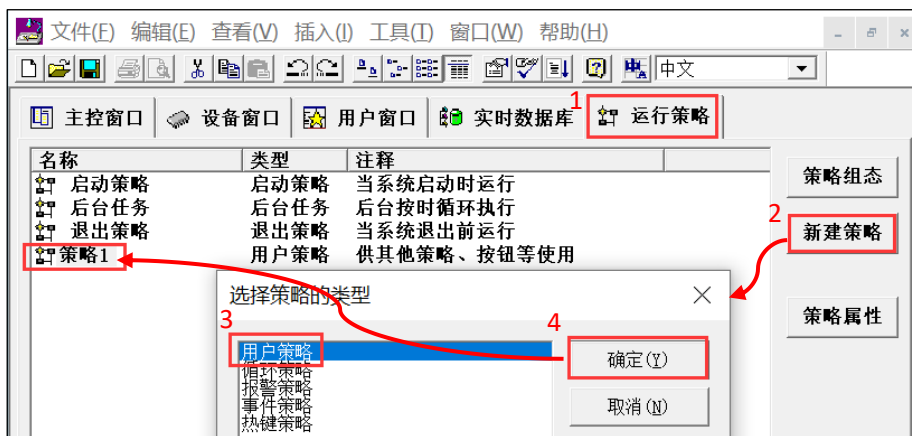


图 11.2-7 新建用户策略

- ③ 双击新建的用户策略【策略1】，将它的名字修改为【等待 3S 策略】，脚本设置为：
 A=300 ‘将 300 赋值给变量【A】’
 !Sleep(3000) ‘暂停 3000 毫秒’

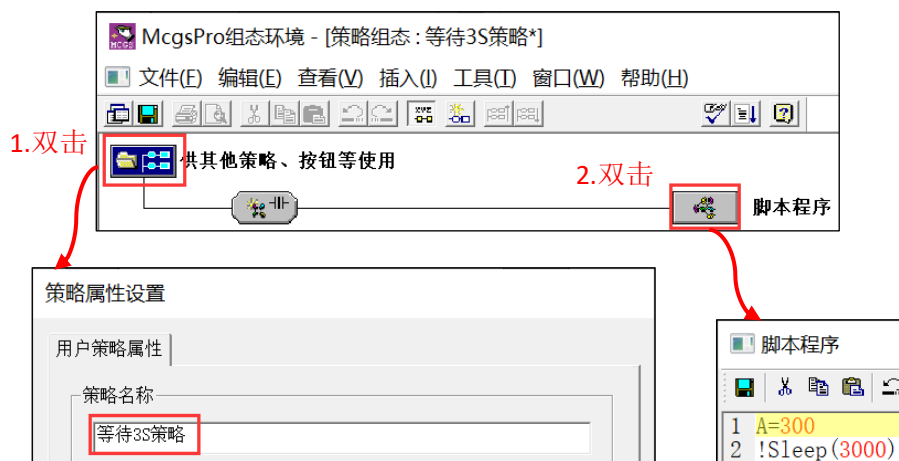


图 11.2-8 用户策略设置

- ④ 用同样的方法再建立两个用户策略，策略名分别为【异步策略】和【同步策略】。其中【异步策略】的脚本为：

‘执行策略【等待3S策略】的同时将500赋值给变量【B】’

!SetStgy(等待3S策略)

B=500

【同步策略】的脚本为：

‘执行策略【等待3S策略】完毕后再将500赋值给变量【B】’

!SetStgyMode(等待3S策略)

B=500

- ⑤ 进入用户窗口，新建一个按钮，将它的显示文本设置为【执行异步策略】，操作属性设置为【执行运行策略块】【异步策略】。

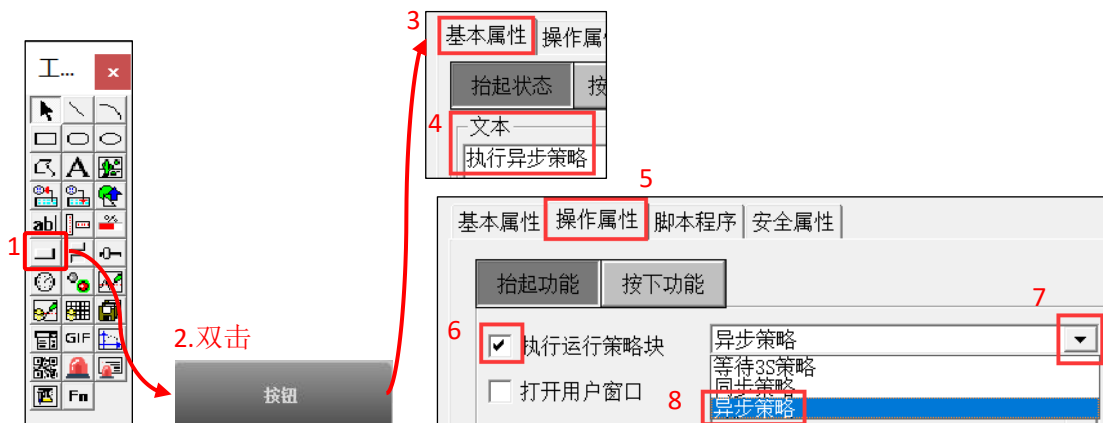


图 11.2-9 执行异步策略功能按钮设置

- ⑥ 用同样的方法再建立一个按钮，将它的显示文本设置为【执行同步策略】，操作属性设置为【执行运行策略块】【同步策略】。

- ⑦ 再次建立一个按钮，将它的显示文本设置为【Clear】，脚本设置为：

‘将A和B的值清零’

A=0

B=0

- ⑧ 然后建立两个标签构件，分别将它们的显示输出关联变量【A】和【B】，输出值类型都选择为【数值量输出】。关联变量【A】的标签属性如图 11.2-10 所示。

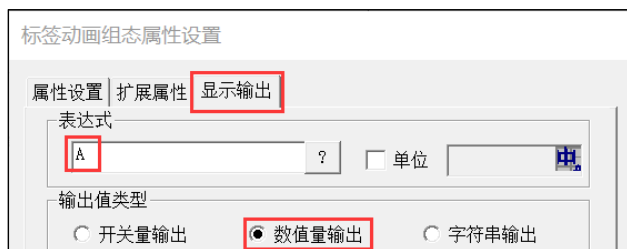


图 11.2-10 标签构件显示输出设置

⑨ 组态完成后，保存工程，然后下载工程，下载成功后启动运行。

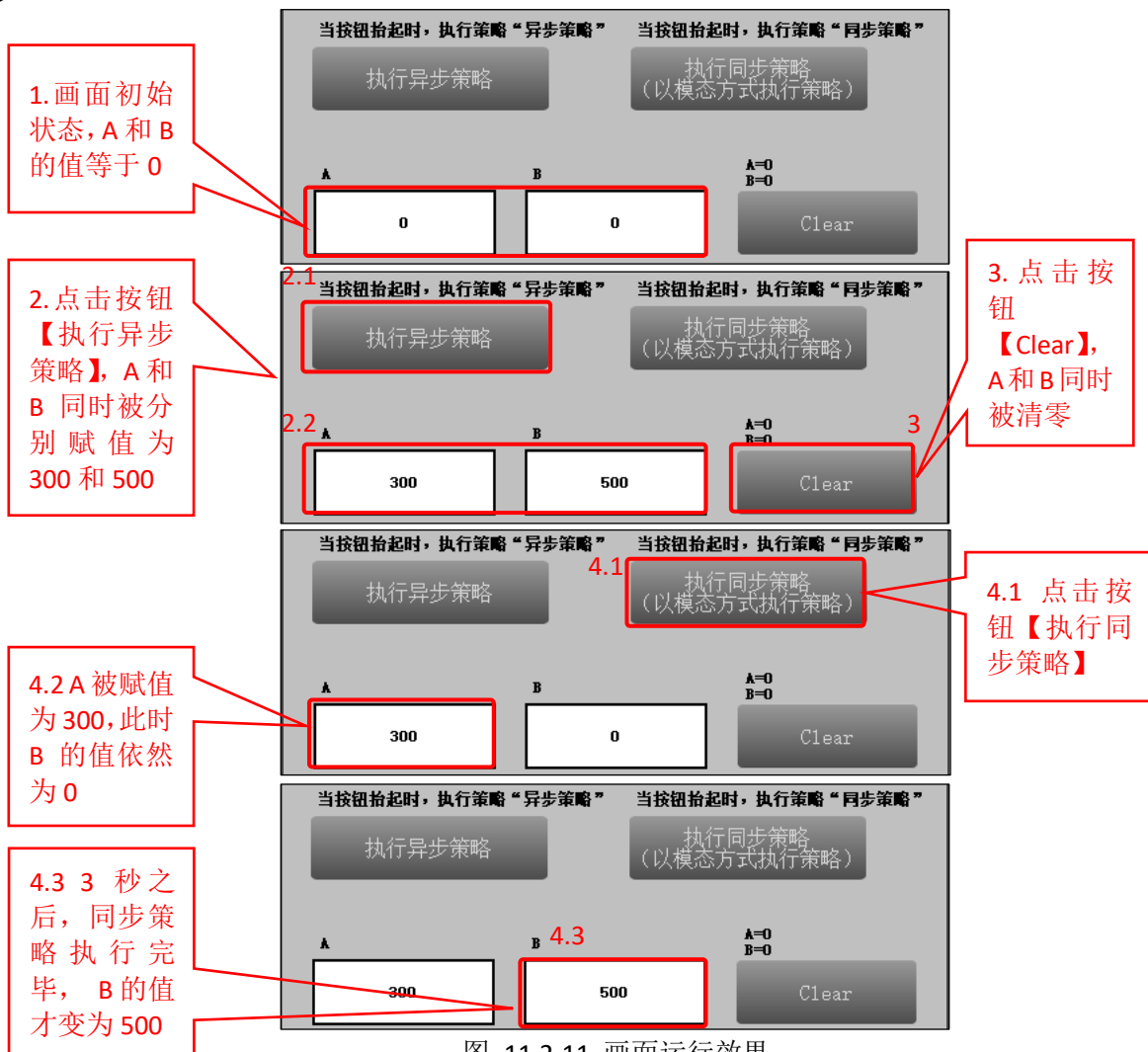


图 11.2-11 画面运行效果

11.2.3 窗口操作函数

11.2.3.1 常用函数介绍

以下 7 个函数为针对窗口的操作函数，通过这些函数可以对窗口执行关闭、显示、获取状态、获取名字等操作。

!CloseAllWindows(WndName): 关闭所有窗口。如果在字符串“WndName”中指定了一个窗口，则首先关闭其他所有窗口，再打开这个窗口。如果“WndName”为空，则关闭所有窗口。

!CloseAllSubWnd(): 关闭当前标准窗口中的所有子窗口。

!CloseSubWnd(WndObject): 关闭窗口名为“WndObject”的子窗口。

!OpenSubWnd(参数 1, 参数 2, 参数 3, 参数 4, 参数 5, 参数 6): 在当前窗口中打开子窗口。

!GetWindowName(Index): 按用户窗口组态时排列的顺序获得用户窗口的名字。

!GetWindowState(WndObject): 根据用户窗口名获取窗口工作状态。

!SetWindow(WndObject,Op): 按照名字操作用户窗口，如打开、关闭、打印。

说明：当函数!*OpenSubWnd*的【参数6】为1时表示以【模态模式】打开子窗口，为2时表示以【菜单模式】打开子窗口。【参数6】的更多信息，请查看 McgsPro 组态软件帮助。

11.2.3.2 样例演示

本节样例主要演示使用函数打开模态式子窗口和菜单式子窗口的区别，以及如何关闭子窗口。

① 新建两个窗口，分别为【窗口0】和【窗口1】。

② 在【窗口0】中新建两个按钮，将它们的显示文本分别修改为【以模态模式打开子窗口】和【以菜单模式打开子窗口】。

【以模态模式打开子窗口】按钮中的脚本程序为：

!OpenSubWnd(窗口1,0,0,400,240,1) ‘在当前窗口的坐标值为(0,0)的位置，以模态模式打开一个长宽为(400,240)的名为【窗口1】的子窗口。以模态方式打开子窗口时，子窗口外的构件对鼠标操作不响应，且鼠标在窗口外点击时窗口不会关闭，必须调用函数 *CloseSubWnd* 或 *CloseAllSubWnd* 来关闭此子窗口。

【以菜单模式打开子窗口】按钮中的脚本程序为：

!OpenSubWnd(窗口1,0,0,600,350,2) ‘在当前窗口的坐标值为(0,0)的位置，以菜单模式打开一个长宽为(600,350)的名为【窗口1】的子窗口。以菜单方式打开子窗口时，一旦在子窗口之外按下鼠标，则子窗口关闭。

③ 进入【窗口1】，新建一个按钮，将它的显示文本修改为【关闭子窗口】，脚本程序为：

!CloseAllSubWnd() ‘关闭所有子窗口。

将【窗口1】的窗口背景颜色设置为【浅蓝色】以突出打开子窗口时的效果。然后将按钮拖拽到本窗口坐标(400,240)以内，避免以模态方式打开子窗口或无法关闭子窗口。

④ 组态完成后保存工程，然后下载工程，下载成功后点击启动运行。启动时画面如图 11.2-12 所示：

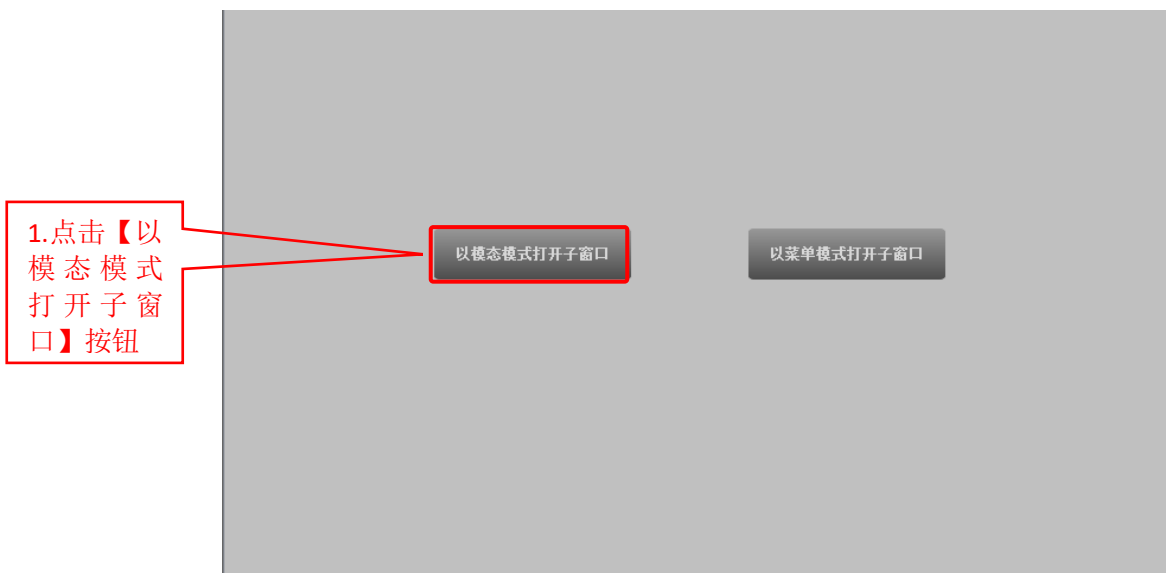


图 11.2-12 启动画面

⑤ 点击【以模态模式打开子窗口】按钮，系统按照按钮中函数的设置，在当前窗口的坐标值为(0,0)的位置，以长宽(400,240)的显示大小打开【窗口1】。此时用鼠标点击子窗口外的画面系统没有任何反应，点击子窗口内部的【关闭子窗口】按钮后，子窗口关闭。

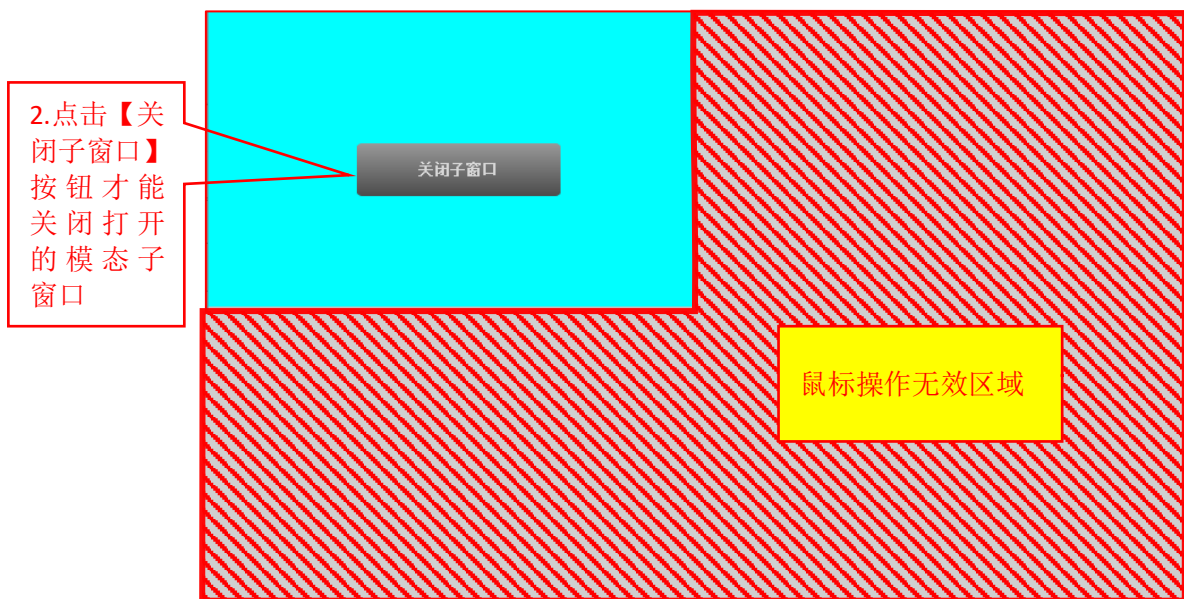


图 11.2-13 以模态模式打开子窗口画面

- ⑥ 然后点击【以菜单模式打开子窗口】按钮，系统按照按钮中函数的设置，在当前窗口的坐标值为(0,0)的位置，以长宽(600,350)的显示大小打开【窗口1】。此时用鼠标点击子窗口外的画面，子窗口关闭。也可以直接点击子窗口内的【关闭子窗口】按钮关闭该子窗口。

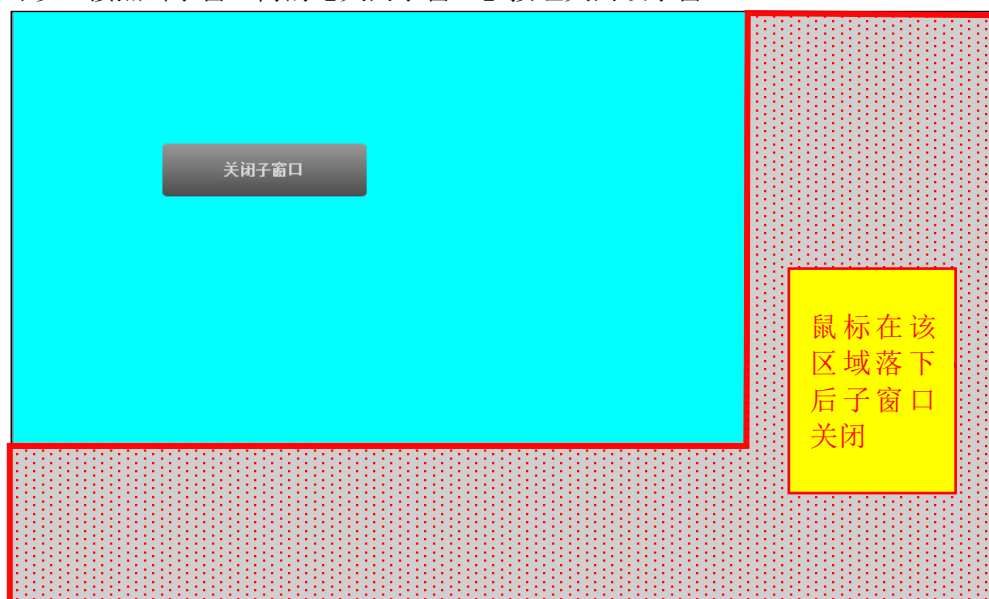


图 11.2-14 以菜单方式打开子窗口画面

11.2.4 系统操作函数

11.2.4.1 常用函数介绍

下面 15 个函数为系统相关的操作函数，通过这些函数可以执行获取当前工程运行期限状态、最后鼠标操作时机等系统相关的操作。

!GetExpiryStatus (): 获取当前工程的运行期限状态。0 为工程未到期；1 为已到期。

!GetLastMouseActionTime (): 获取最后一次鼠标的动作发生的时间。

!SetDevice(DevName,DevOp,CmdStr): 按照设备名字对设备进行操作。该函数的详细介绍和应用举例请参考章节：进阶篇 15.2.3 设备命令。

!Beep (): 发出嗡鸣声。

!SendKeys(string): 将一个或多个按键消息发送到活动窗口，如同在键盘上进行输入一样。

!SetTime(n1,n2,n3,n4,n5,n6): 设置当前系统时间。

!GetCurrentLanguageIndex ()：获取当前使用的语言的索引值。

!SetCurrentLanguageIndex(整数)：过索引项设定当前语言环境。

!GetLocalLanguageStr(整数)：获得指定自定义 ID 对应的当前语言的内容。

!GetLanguageNameByIndex(整数)：根据语言索引值返回语言名称。

!PrintToFile(folder, filename, flag)：截屏输出到文件。

!SetDialogBy9Palace(对话框索引, 对话框大小等级, 九宫格模拟弹出方位)：根据需要调整键盘或指定模态对话框的大小和九宫格弹出位置。

!SetDialogByXYPosition(对话框枚举, 对话框大小等级, X 轴弹出方位, Y 轴弹出方位)：根据需要调整键盘或指定模态对话框的大小和 XY 轴弹出位置。

!PrinterSetup ()：调用打印设置。

!Sleep(interval ()：暂停指定的时间，单位毫秒。

11.2.4.2 样例演示

在工程运行过程中，往往会出现需要将当前画面截屏并保存下来的情况，此时可以使用函数 **!PrintToFile**。

- ① 在 McgsPro 组态软件中，进入希望截图的窗口画面，增加一个按钮，将按钮的显示文本显示为【打印屏幕】，脚本设置为：

!PrintToFile("Folder", "Name", 0) ‘截屏内容保存为 Name.jpg 文件，存储到 U 盘的 Folder 目录下。

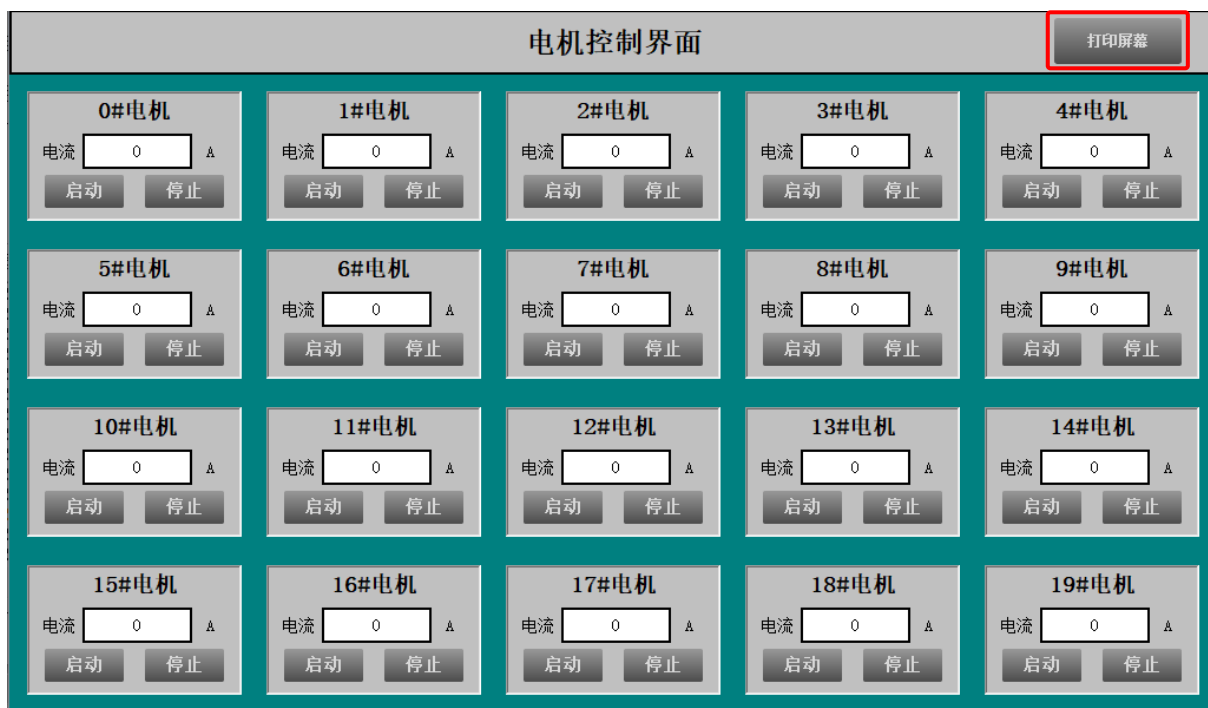


图 11.2-15 打印屏幕功能按钮设置

- ② 此时在工程运行过程中点击此按钮已经可以实现截屏的目的了，但是为了美观和符合实际，我们不希望屏幕中出现【打印屏幕】按钮。此时可以选中【打印屏幕】按钮，点击工具栏上的【置于最后】 图标按钮，将【打印屏幕】按钮放置到最后一层，被上层构件遮挡住。设置步骤和效果如图 11.2-16 所示。



图 11.2-16 遮蔽不希望被截屏的按钮

- ③ 然后进入工作台 → 主控窗口 → 双击 → 选择【响应重叠区所有构件】。

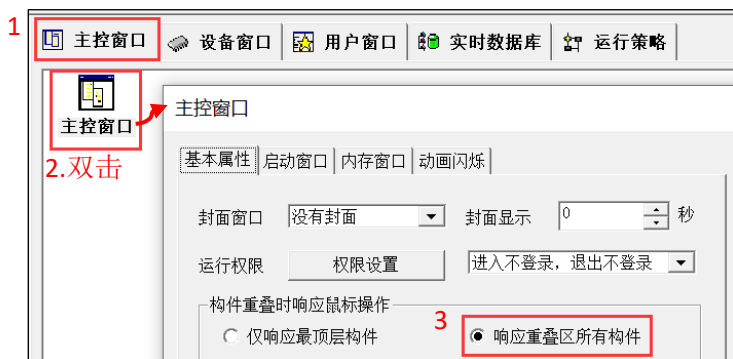


图 11.2-17 设置相应重叠区域所有构件

④ 此时在运行工程，即可凭记忆点击按钮所在位置，实现截屏并保存到 U 功能。

如果将函数设置为!PrintToFile("\$MCGS_DIR_USER\Folder","Name",0)，则可以将截屏文件 Name.jpg 保存到 TPC 的用户自定义分区的 Folder 目录下。

11.2.5 操作日志函数

11.2.5.1 常用函数介绍

以下 4 个函数为与操作日志相关的函数，通过这些函数可以执行操作日志的导出、清除、开启、关闭。

!ExportOperationLogToCSV(文件名, 开始时间, 结束时间, 导出模式): 导出操作日志到 csv 文件。

!OperationLogClear(): 清除所有的操作日志。

!OperationLogDisable(): 关闭操作日志功能，最小执行间隔 3 秒，频繁调用无效。

!OperationLogEnable(): 开启操作日志功能，最小执行间隔 3 秒，频繁调用无效。

11.2.5.2 样例演示

关于操作日志函数的样例演示将在后续章节：进阶篇 13.1.3.2 对日志进行操作进行介绍。

11.3 数据对象操作函数

数据对象操作函数主要分为数据设置函数、组对象操作函数、历史数据操作函数、初值保存函数和报警操作函数。

11.3.1 数据设置函数

11.3.1.1 常用函数介绍

!DataGetInt (DatName, Value) : 通过数据对象名【DatName】读取整数数据对象的值【Value】。

!DataSetInt (DatName, Value) : 通过数据对象名【DatName】写入整数数据对象的值【Value】。

11.3.1.2 样例演示

① 在实时数据库中新建两个整数型变量【A】和【B】

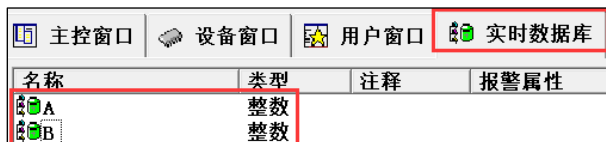


图 11.3-1 新建变量

② 进入用户窗口，新建两个按钮，将它们的显示文本分别设置为【获取变量数据】和【设置变量数据】。

其中，【获取变量数据】按钮的脚本程序是：

!DataGetInt("A",B) ‘将变量 A 的值赋值给变量 B’

【设置变量数据】按钮的脚本程序是：

!DataSetInt("A",B) ‘将变量 B 的值赋值为变量 A’

③ 然后新建两个输入框构件，在它们的操作属性处分别关联变量【A】和【B】，用于函数参数的赋值和

运行结果显示。

- ④ 组态完成后，保存工程，然后下载工程，下载成功后启动运行。将变量【A】的值设置为【9】。然后点击【获取变量数据】按钮，可以看到，变量【B】的值变成【9】。

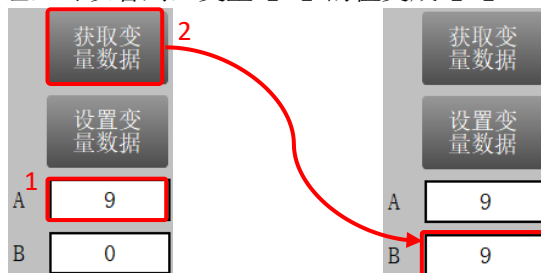


图 11.3-2 !DataGetInt("A",B)的运行效果

- ⑤ 然后将【B】设置为5，点击【设置变量数据】按钮，可以看到，【A】的值也变成了【5】。

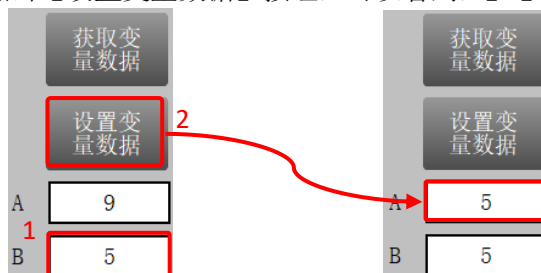


图 11.3-3 !DataSetInt("A",B)的运行效果

11.3.2 组对象操作函数

当变量是组对象的成员时，也可以使用组对象操作函数来获取和设置变量的值。

11.3.2.1 常用函数介绍

!GroupGetInt(组对象名,成员编号,返回值)：读取【组对象名】中由【成员编号】指定的成员的值给【返回值】。

!GroupSetInt(组对象名,成员编号,设置值)：设置【组对象名】中由【成员编号】指定的成员的值【设置值】。

11.3.2.2 样例演示

- ① 在实时数据库中新建两个整数型变量【A】和【B】，一个组对象【GROUP】。

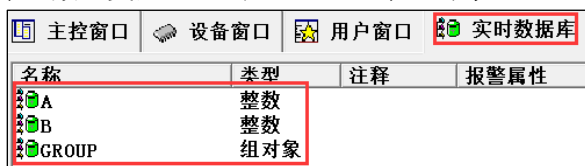


图 11.3-4 新建变量

- ② 将【A】设置为【GROUP】的组对象成员。组对象成员的编号是从0开始的，设置完后，【A】的编号为0。

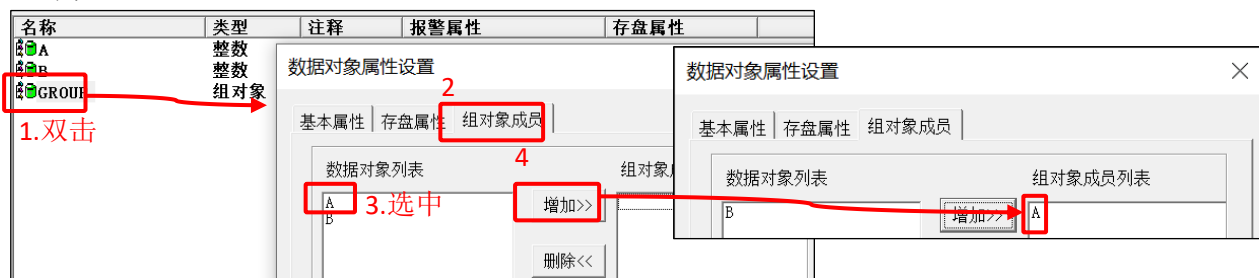


图 11.3-5 设置组成员

- ③ 在用户窗口中新建两个按钮，将它们的显示文本设置为【获取组对象成员数据】和【设置组对象成员数据】。
其中，【获取组对象成员数据】按钮的脚本程序是：
`!GroupGetInt(Group,0,B)` 将组对象【GROUP】的编号为【0】的组成员（即A）的值赋值给【B】
【设置组对象成员数据】按钮的脚本程序是：
`!GroupSetInt(Group,0,B)` 设置组对象【GROUP】的编号为【0】的组成员（即A）的值赋值为【B】
- ④ 然后新建两个输入框构件，在它们的操作属性处分别关联变量【A】和【B】，用于函数参数的赋值和运行结果显示。
- ⑤ 组态完成后，保存工程，然后下载工程，下载成功后启动运行。将变量【A】的值设置为【22】。然后点击【获取组对象成员数据】按钮，可以看到，变量【B】的值变成【22】。

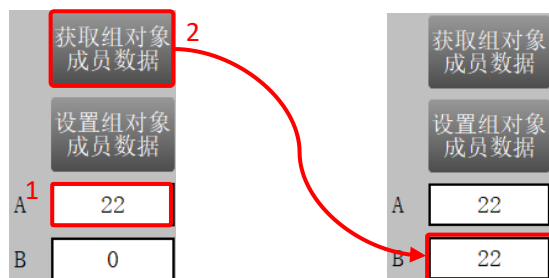


图 11.3-6 !GroupGetInt(Group,0,B)的运行效果

- ⑥ 然后将【B】设置为 66，点击【设置组对象成员数据】按钮，可以看到，【A】的值也变成了【66】。

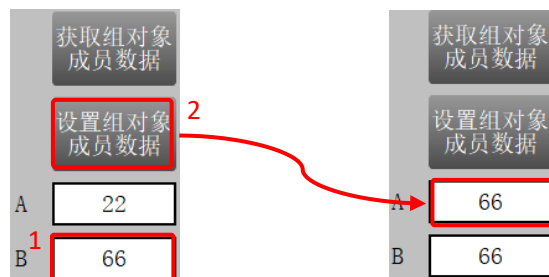


图 11.3-7 !GroupSetInt(Group,0,B)的运行效果

11.3.3 历史数据操作函数

以下 4 个函数为历史数据操作函数。

11.3.3.1 常用函数介绍

● 函数的意义

!SaveData(DataName): 把组对象 DataName 所有成员的当前值存入存盘数据库中，此组对象必须具有存盘属性，即需在其【存盘属性】设置界面勾选“定时存储到磁盘”或“定时存储到内存”，否则会操作失败。

!FreshDataSave(): 将已存盘的历史数据写入磁盘，最长等待时间 3s。

!DelAllSaveDat(DataName): 删除组对象 DataName 对应的所有存盘数据(包括内存中和磁盘中的数据)。

注意：此函数不能用来删除报警存盘数据。

!ExportHisDataToCSV(文件名,组对象名,字段名,开始时间,结束时间,最大记录数,导出模式,导出参数,进度指示数据对象名,控制数据对象名): 以 CSV 格式导出指定时间段的历史存盘数据或历史报警数据到 U 盘。

● 存盘的定义和机制

McgsPro 在运行过程中，根据设定周期或通过脚本调用将指定数据的值写入磁盘文件进行保存的过程，就是历史数据存盘。历史数据存盘只对具有存盘属性的组对象起作用。



图 11.3-8 存盘属性

存盘方式分类

选择【定时存储到内存】时，历史数据不会被存储到磁盘，掉电后数据清除。

选择【定时存储到磁盘】有两种历史数据存盘方式：

1. 定时周期存盘：存盘周期大于 0 秒，按照设置的存盘周期循环存盘。
2. 函数触发存盘：存盘周期等于 0 秒，需要使用 `!SaveData` 脚本函数存盘。

实现机制

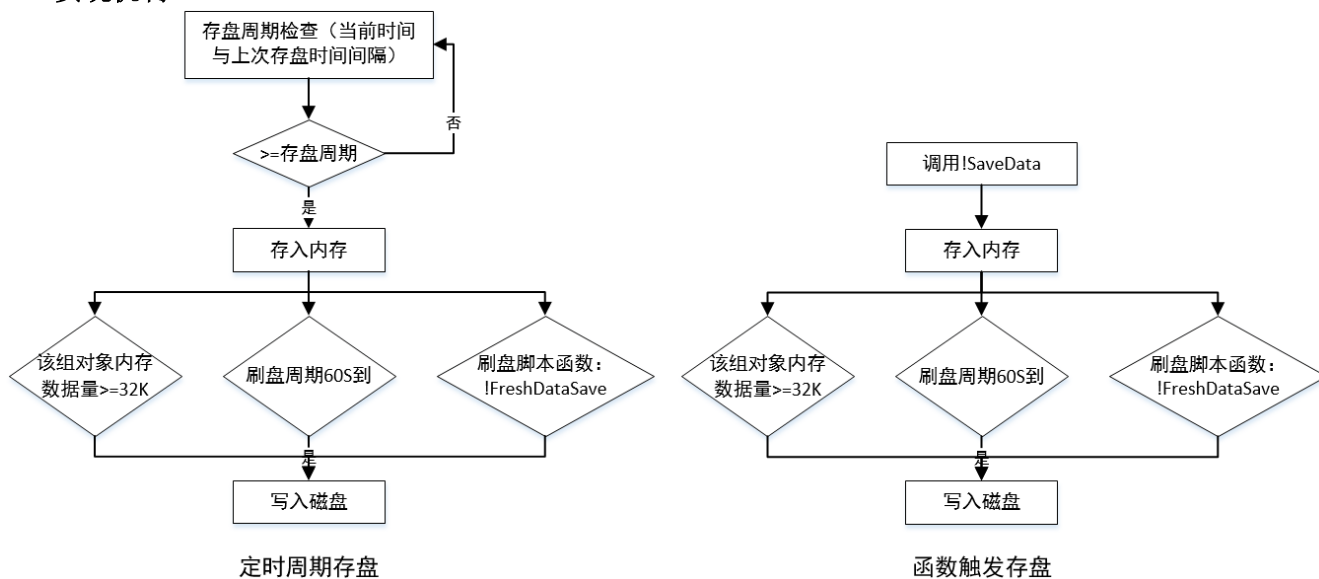


图 11.3-9 存盘机制

历史数据都是先保存在内存中，在满足一定的条件（三种：该组对象内存数据量 $\geq 32\text{KB}$ ；刷盘周期 60S 时间到；调用了刷盘函数 `!FreshDataSave`）时，才写入到磁盘中实现永久保存。定时周期存盘和函数触发存盘的内部实现机制差不多，只是它们的触发方式不同而已。定时周期存盘是按照设置的存盘周期循环保存数据，而函数触发存盘是由函数脚本触发的。

11.3.3.2 样例演示_历史数据保存

● 定时周期存盘

定时周期存盘组态非常简单，只要按需求设置存盘组对象的存盘周期即可。

例如：实时数据库中有三个数值型变量 Data01、Data02、Data03，需要记录它们每 1 秒的数据。则组态步骤为：

- 1.在实时数据库中，将这三个变量加入到一个组对象（Group1）中。
- 2.设置组对象的存盘属性为【定时存储到磁盘】，存储周期为 10*0.1 秒。

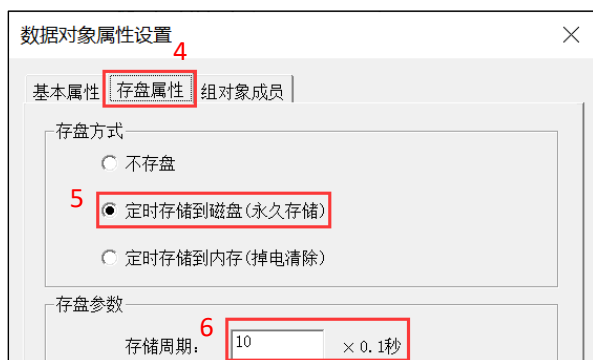
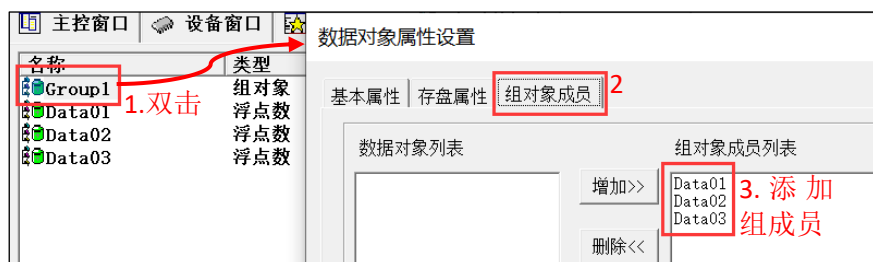


图 11.3-10 定时周期存盘

● 函数触发存盘

函数触发存盘需要调用函数!SaveData 实现存盘操作。

例如：有三个数值型成员 Data04、Data05、Data06，需要在数据对象 Data00 变化的时候进行存盘。则组态步骤为：

- 1.在组态环境中，将这三个变量加入到一个组对象（Group2）中。
- 2.设置组对象的存盘属性为【定时存储到磁盘】，存盘周期为 0 秒。
- 3.在运行策略窗口添加事件策略【策略 1】。

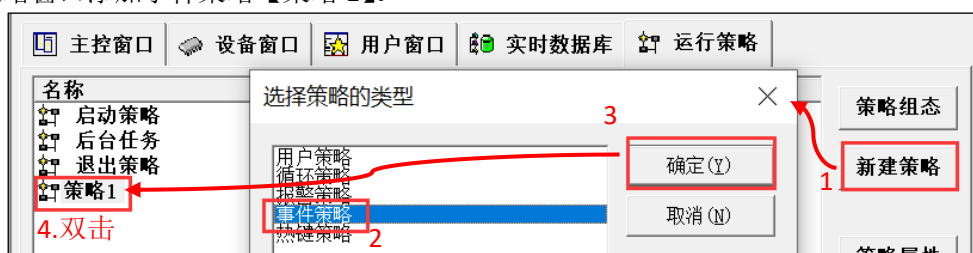


图 11.3-11 新增事件策略

- 4.向该事件策略【新增策略行】，设置事件策略属性，并加入存盘脚本程序。

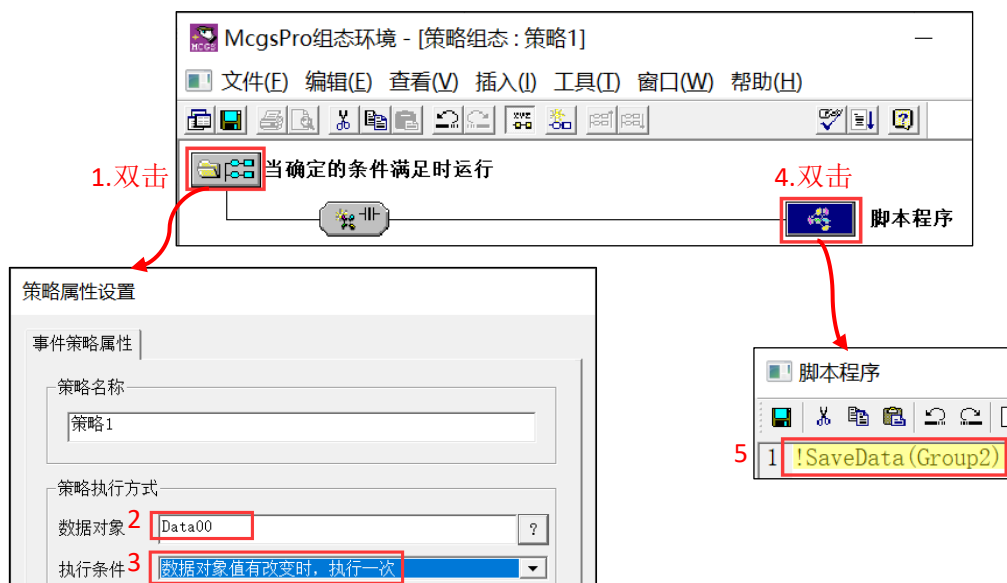


图 11.3-12 设置策略属性和存盘脚本

从存盘的运行机制可以知道，存盘动作产生后，并没有立即将保存的历史数据存储到磁盘文件中，而是暂存在内存里，等到组对象内存数据 $\geq 32\text{KB}$ 或刷盘周期 60S 时间到之后，才会被系统刷入磁盘中永久保存。为了防止断电丢失内存中的历史数据，可以在脚本 `!SaveData(Group2)` 下再增加一行刷盘脚本程序 `!FreshDataSave()`，在存盘后立即将数据存入磁盘。

请注意：频繁调用 `!FreshDataSave()` 脚本，会对系统性能造成影响。该脚本一般在使用触发存盘时才使用，若都是定时周期存盘，不建议使用。

另外，建议在断电前安全退出运行环境，这样会将内存中的所有历史数据保存到磁盘文件中。可以添加一个按钮，将它的操作属性设置为【退出运行系统】，点击该按钮退出运行环境后再断电。

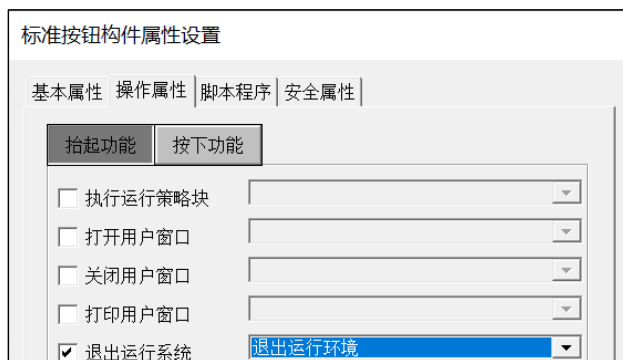


图 11.3-13 使用按钮安全退出运行环境

11.3.3.3 历史数据的显示和统计

历史数据存盘，也是为了以后可以分析这些数据，所以就必然要查看这些数据或者形象地表示这些数据。主要有三种构件用来查看分析历史数据：

1. 存盘数据浏览构件：主要用于对历史数据的直观浏览；
2. 历史曲线构件：主要用于形象的表达历史数据变化趋势；
3. 报表构件：主要用于统计分析历史数据。

存盘数据浏览构件

以前面提到的 Group1 为例，设置存盘数据浏览构件的数据来源为 Group1，在显示属性中设置序号、时间和成员值，运行工程，就可以浏览 Group1 的所有历史数据了，如图 11.3-14 所示。

■CGS序号	■CGS_Time	Data01	Data02	Data03
1	2020-08-07 13:35:30	0.00	0.00	0.00
2	2020-08-07 13:37:32	30.00	20.00	10.00
3	2020-08-07 13:37:41	66.00	69.00	99.00
4	2020-08-07 13:37:48	55.00	32.00	99.00
5	2020-08-07 13:37:56	55.00	36.00	58.00
6	2020-08-07 13:38:07	65.00	54.00	0.00
7	2020-08-07 13:38:22	2.00	38.00	77.00
8	2020-08-07 13:38:38	44.00	89.00	78.00

图 11.3-14 存盘浏览构件

历史曲线构件

仍然以 Group1 为例，设置历史曲线的存盘数据为 Group1，并设置其成员 Data01、Data02、Data03 对应三条曲线，假如它们的数值分别按照正弦、方波、三角曲线的方式变化，则最后历史曲线显示的效果如图 11.3-15 所示。

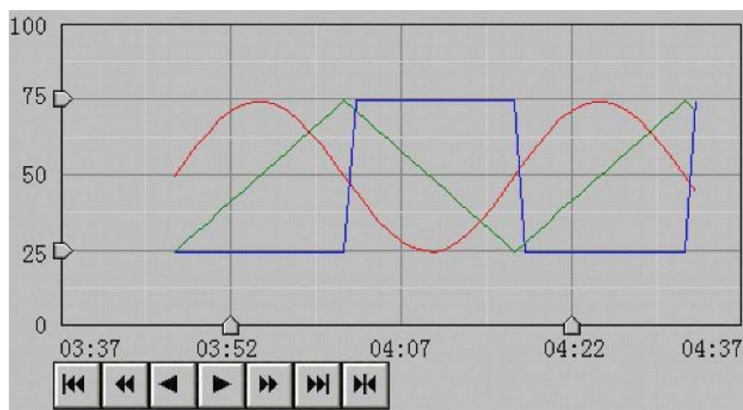


图 11.3-15 历史曲线构件

使用存盘数据浏览构件和历史曲线构件显示历史数据的详细步骤请参考章节：[进阶篇 5.2.2.2 存盘浏览构件显示历史数据](#)。

报表构件

比如可以分别对 Group1 和 Group2 的成员各自求和，显示效果如图 11.3-16 所示。

Data01	Data02	Data03	求和
25.2182	75	47.895	148.113
Data04	Data05	Data06	求和
16.1	26.1	126.1	168.3

图 11.3-16 报表构件

使用报表构件统计历史数据的详细步骤请参考章节：[进阶篇 5.2.3 历史数据统计](#)。

11.3.3.4 历史数据的导出和删除

导出

有时候需要将历史数据以更好的格式导出到电脑中，以便更方便地查看分析其数据，这时可以使用脚本函数!ExportHisDataToCSV 将组对象的历史数据导出到 CSV 文件中，使用 Excel 软件方便地查看数据。

详细介绍请参考章节：[进阶篇 5.2.4 历史数据导出](#)。

删除

McgsPro 提供三种方法删除历史数据，详细介绍请参考章节：[进阶篇 5.2.5 历史数据删除](#)。

11.3.4 保存初值（掉电保持）函数

一般来说，工程启动时，实时数据库中变量的初始值是组态时设置的初始值。若想重启后变量的值恢复为重启前的值，就需要将变量的值先保存起来，使用的功能是【保存初值】功能。

!SaveSingleDataInit(DataName): 把数据对象的当前值设置为初始值。

!FlushDataInitValueToDisk(): 把保存在内存中的初值写入磁盘，调用后即刻刷盘。

函数的详细介绍请参考章节：[进阶篇 5.4 变量初值保存功能](#)。

11.3.5 报警操作函数

11.3.5.1 常用函数介绍

报警相关的函数需要指定对象是整数或浮点数，并且需要指定报警序号的报警已设置。以下为 3 个常用的报警相关的函数。

!GetAlmValue(DataName,AlarmIndex,Value,Flag): 读取数据对象 DataName 的报警序号为 AlarmIndex 的由 Flag 指定的报警属性值，赋值给 Value。

!SetAlmValue(DataName,AlarmIndex,Value,Flag): 设置数据对象 DataName 的报警序号为 AlarmIndex 的由 Flag 指定的报警属性值。

!ClearHistoryAlarmData(): 删除所有的历史报警数据。

说明：当函数 *!GetAlmValue* 和 *!SetAlmValue* 的参数【Flag】为 12 时表示【报警基准值】，参数【Flag】的更多信息，请查看 McgsPro 组态软件帮助。

11.3.5.2 样例演示_读取和设置报警基准值

- ① 在实时数据库中新建三个浮点型变量【A】【B】和【温度】。

名称	类型
A	浮点数
温度	浮点数
B	浮点数

图 11.3-17 新建变量

- ② 给变量【温度】添加一条报警：当【温度】的值 ≤ 10 时报警，报警内容为【温度过低】。

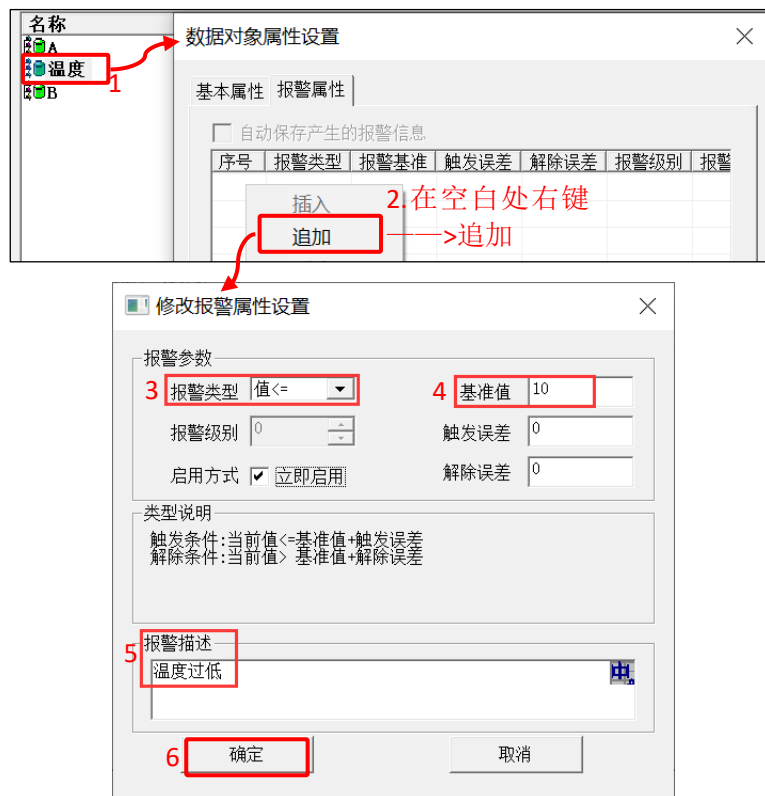


图 11.3-18 设置一条报警

- ③ 添加的报警序号为【0】。



图 11.3-19 报警序号

- ④ 在用户窗口中新建两个按钮，将它们的显示文本分别设置为【获取序号 0 报警基准值】和【设置序号 0 报警属性】。

其中，按钮【获取序号 0 报警基准值】中的脚本程序为：

`!GetAlmValue(温度,0,A,12)` '获取变量【温度】的序号为【0】的那条报警的报警基准值给变量【A】'

按钮【设置序号 0 报警属性】中的脚本程序为：

`!SetAlmValue(温度,0,B,12)` '设置变量【温度】的序号为【0】的那条报警的报警基准值为变量【B】的值'

- ⑤ 新建 3 个输入框构件，分别在它们的【操作属性】处关联变量【A】【B】和【温度】。
⑥ 新建一个报警浏览构件。选择【数据来源】为【实时报警数据】。

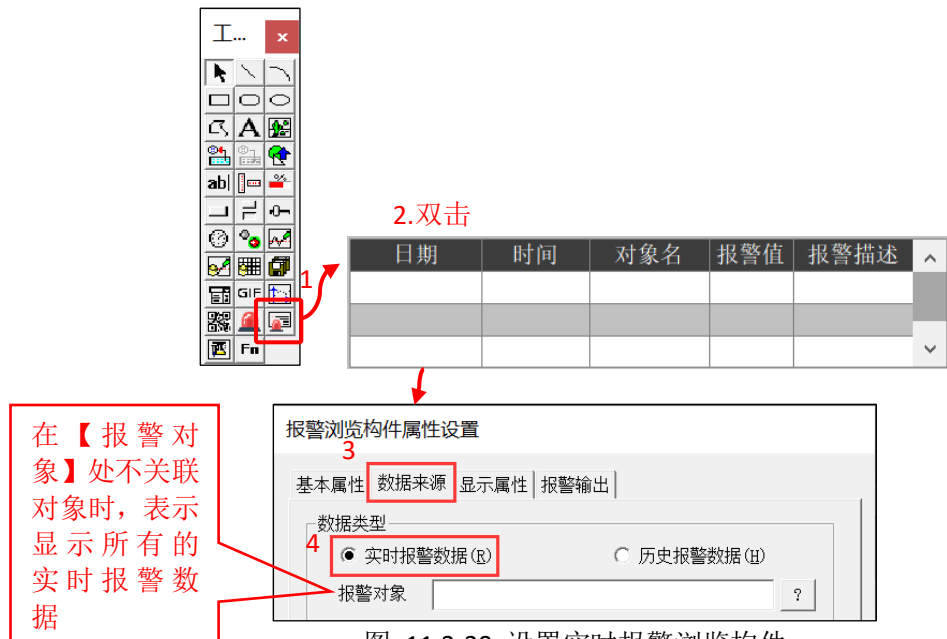


图 11.3-20 设置实时报警浏览构件

- ⑦ 组态好工程后保存工程，然后下载工程，下载成功后启动运行。由于变量【温度】的默认初始值为【0】，小于设置的报警基准值【10】，所以在工程运行的最初，报警浏览构件即显示了一条报警消息。

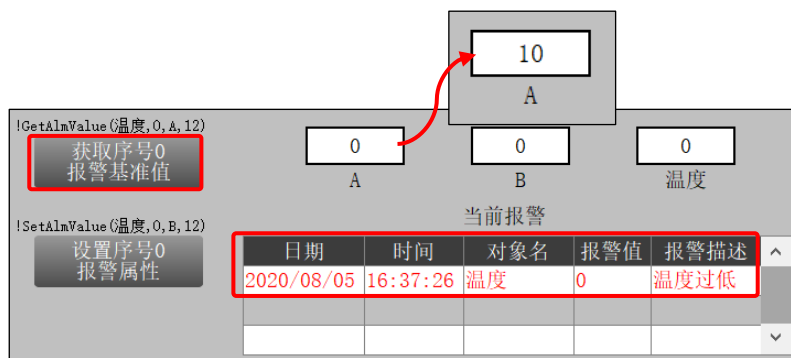


图 11.3-21 运行演示一

- ⑧ 点击【获取序号 0 报警基准值】按钮，变量【A】的值变为【温度】第 0 条报警的报警基准值【10】。如图 11.3-21 所示。
- ⑨ 将变量【B】的值设为【-2】，然后点击【设置序号 0 报警属性】按钮，则【温度】第 0 条报警的报警基准值被设置成了【-2】。此时由于【温度】的值【0】，已经不再小于新的基准值【-2】，故报警消除，报警浏览构件中的报警消息也随之消失。



图 11.3-22 运行演示二

11.3.5.3 样例演示_删除历史报警数据

- ① 回到 McgsPro 组态软件，在 11.3.5.3 样例工程的用户窗口中，再添加一个【报警浏览】构件，设置它的【数据来源】为【历史报警数据】。

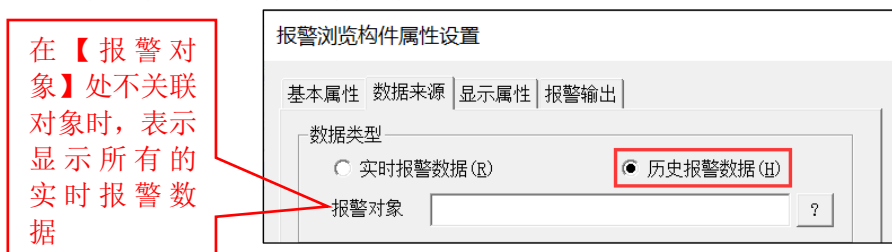


图 11.3-23 设置历史报警浏览构件

- ② 在窗口空白处双击鼠标，设置窗口循环脚本为 `Refresh()`，循环时间为 1000ms。
- ③ 新增一个按钮，将它的显示文本设置为【清除所有历史报警】，脚本程序为：
`!ClearHistoryAlarmData()` ‘删除所有历史报警数据’
- ④ 组态好工程后保存工程，然后下载工程，下载成功后启动运行。注意，由于重新下载了工程，此时【温度】的报警基准值依然是【10】，初始值为【0】。通过输入框设置【温度】的值依次为【22（大于 10）】→【2（小于 10）】→【33（大于 10）】→【3（小于 10）】。可以看到设置了【数据来源】为【历史报警数据】的【报警浏览】构件上出现了 3 条报警信息，即显示了所有出现过的报警。

历史报警					
日期	时间	对象名	报警值	报警描述	结束时间
2020/08/05	17:30:34	温度	3	温度过低	
2020/08/05	17:30:28	温度	2	温度过低	2020/08/05 17:30:31
2020/08/05	17:30:22	温度	0	温度过低	2020/08/05 17:30:26

图 11.3-24 运行演示一

- ⑤ 然后点击【清除所有历史报警】，构件中的历史报警消失。

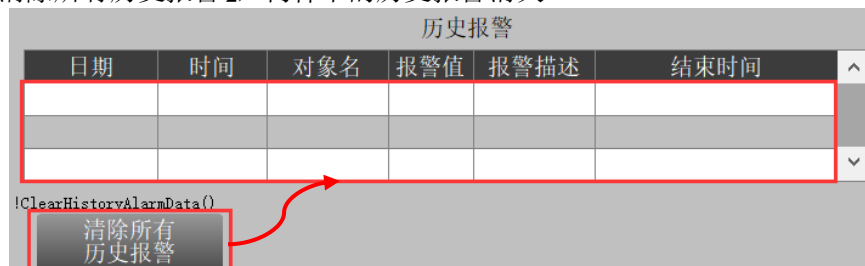


图 11.3-25 运行演示二

11.4 用户权限函数

在 McgsPro 组态软件中，可以通过脚本函数来实现用户的登录、退出、登录用户名获取、登录用户组名获取、打开用户管理对话框、登录用户密码修改。相关函数的详细内容将在后续章节：进阶篇 13.1.2.4 进行介绍。

11.5 时间函数

McgsPro 中，所有的时间在电脑端的模拟将以 1970 年 1 月 1 日 8 时 0 分 0 秒为基线，而人机界面端的实际运行将以 1970 年 1 月 1 日 0 时 0 分 0 秒为基线，时间值为以基线作为基准增加的秒数。

涉及时间操作的函数有以下几类：

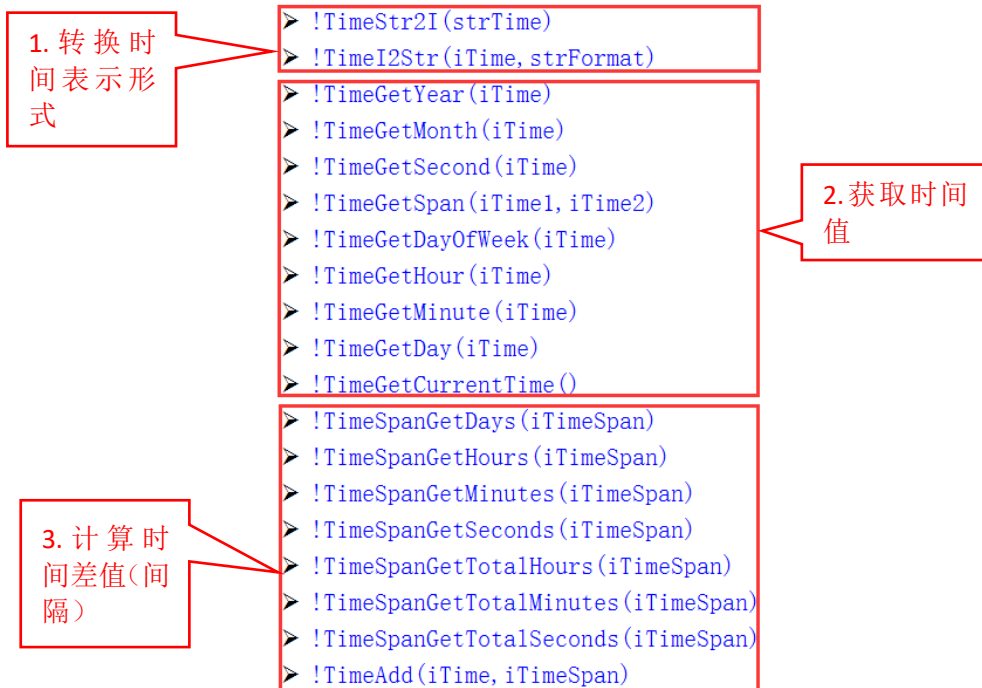


图 11.5-1 时间函数

11.5.1 常用函数介绍

!TimeI2Str(iTime, strFormat): 将时间值转换为字符串表示的时间。

!TimeGetCurrentTime(): 获取当前时间值。

!TimeAdd(iTime, iTimeSpan): 向时间 iTime 中加入由 iTimeSpan 指定的秒数。

11.5.2 样例演示

本节样例介绍如何使用时间函数推算：当前时间再过数分钟后的时间是多少。

- ① 首先在实时数据库中新建三个整数型变量【iTime】【addTime】和【iTime2】用来表示当前时间值、增加的时间和增加后的时间值，两个字符串型变量【strTime】和【strTime2】用来表示当前时间和增加后的时间。

名称	类型	注释	报警属性
iTime	整数	当前时间值	
iTime2	整数	增加后的时间值	
strTime	字符串	当前时间	
strTime2	字符串	增加后的时间	
addTime	整数	增加的分钟数	

图 11.5-2 新建变量

- ② 在用户窗口中，新建一个输入框构件，将它的显示输出关联变量【iTime】。

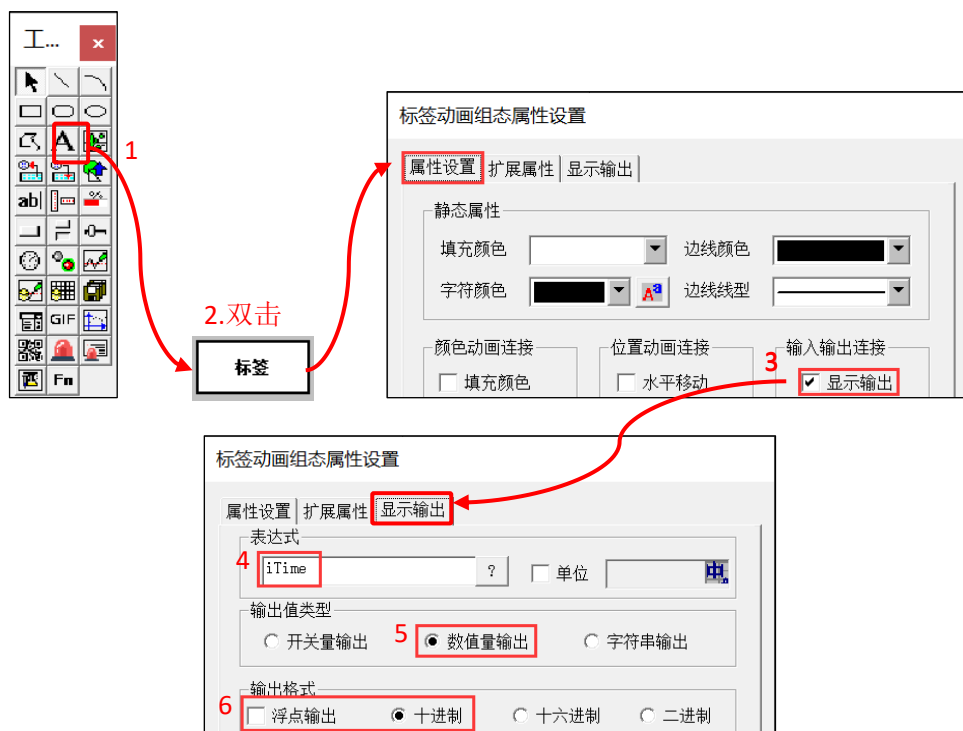


图 11.5-3 设置标签构件步骤

- ③ 用同样的方法再新建三个标签构件，使它们分别关联变量【iTime2】、【strTime】和【strTime2】。由于【strTime】和【strTime2】是字符串型变量，所以在第 4 步关联变量后选择输出值类型为【字符串输出】，其他使用系统默认设置，点击确认即可。

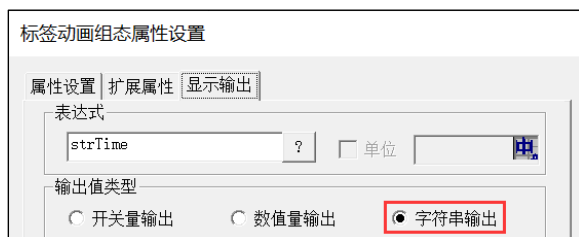


图 11.5-4 设置其它标签构件

- ④ 然后新建一个输入框构件，在它的操作属性中关联变量【addTime】。

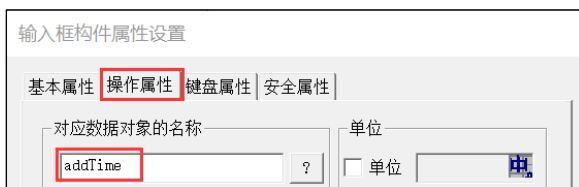


图 11.5-5 设置输入框构件关联变量

- ⑤ 给输入框添加事件脚本，实现功能：当输入框内容发生变化时，执行脚本给当前时间值加上增加的时间，得到加时的时间值，并转换成字符串表示。设置步骤如图 11.5-6 所示。

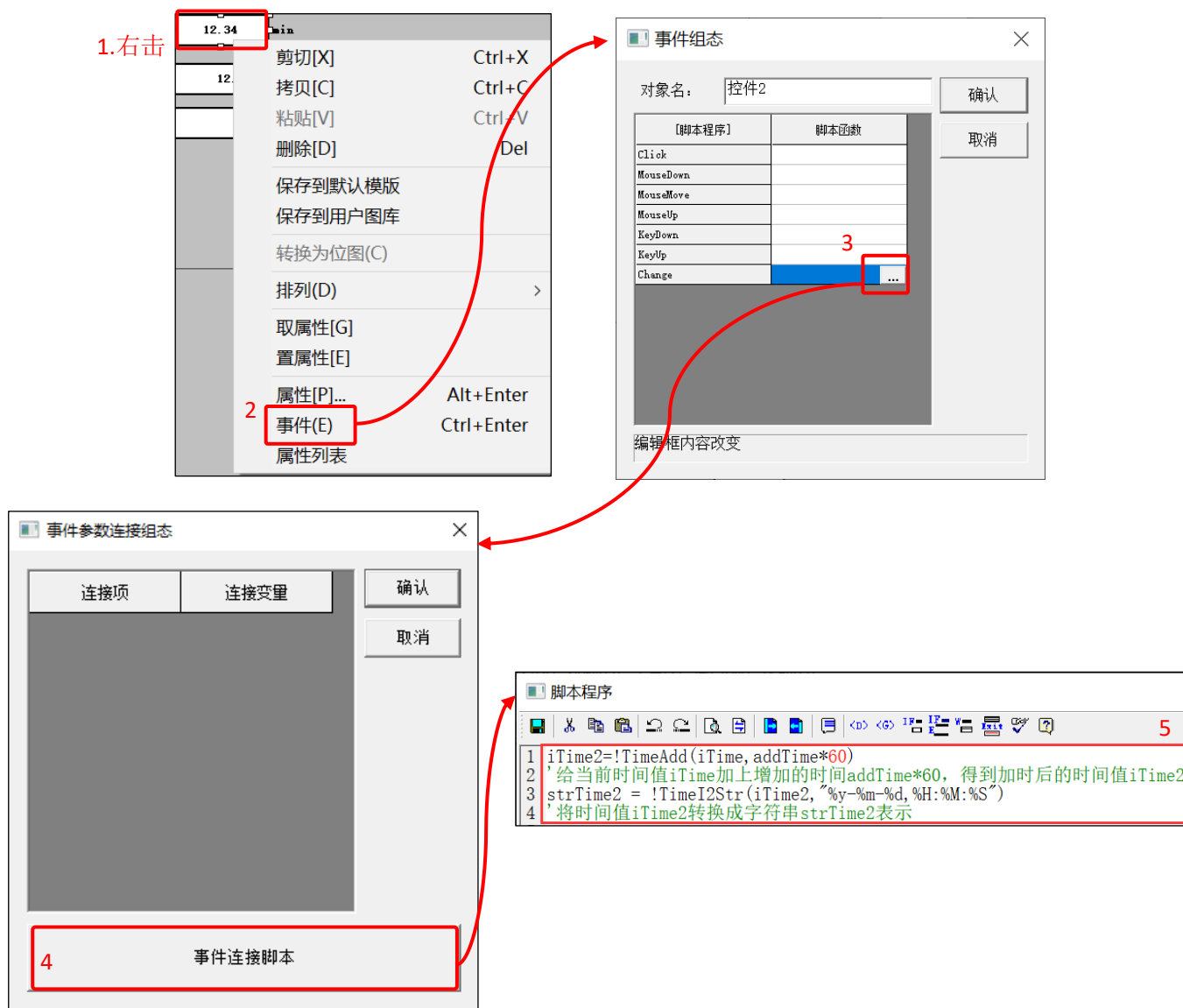


图 11.5-6 设置输入框构件事件脚本

- ⑥ 然后在窗口空白处双击，添加启动脚本：获取当前时间值【iTime】，并转换为字符串【strTime】表示。

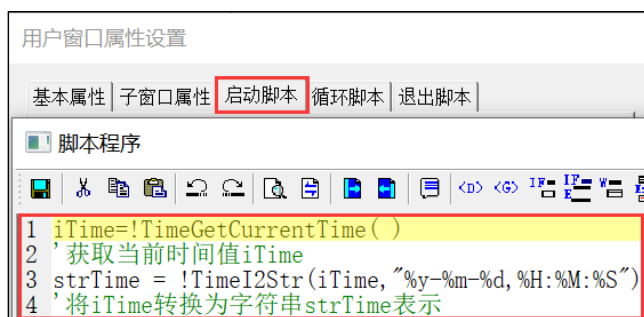


图 11.5-7 启动脚本

- ⑦ 保存工程后下载工程，然后启动运行。屏幕显示当前时间值【iTime】和当前时间【strTime】。在关联【addTime】的输入框中输入 30，系统自动计算出 30 分钟后的时间值【iTime2】和时间【strTime2】。

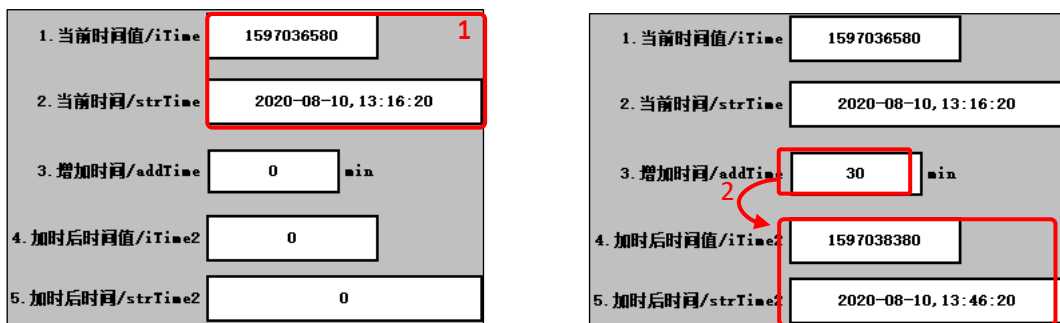


图 11.5-8 运行演示

11.6 数学函数

数学函数一共包含 31 个函数，主要分为三角函数、反三角函数、按位运算函数、取整函数和其他数学函数。

● 函数介绍及举例

三角函数：

!Sin(x)：正弦函数。

!Cos(x)：余弦函数。

!Tan(x)：正切函数。

!Cot(x)：余切函数。

!Sec(x)：正割函数。

!Csc(x)：余割函数。

请注意：三角函数中的参数 x 均为弧度。

三角函数举例：

!Sin(x)：正弦函数。

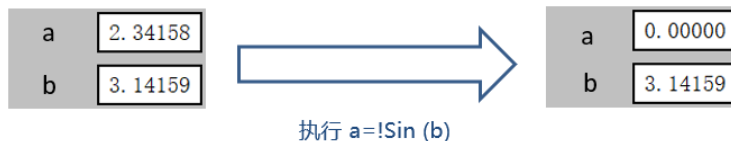


图 11.6-1 三角函数应用举例

反三角函数

!Asin(x)：反正弦函数。

!Acos(x)：反余弦函数。

!Atn(x)：反正切函数。

!Acot(x)：反余切函数。

请注意：反三角函数中的参数 x 均为弧度。

反三角函数举例：

!Asin(x)：反正弦函数。

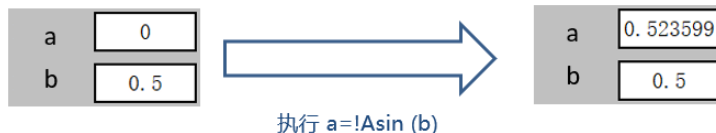


图 11.6-2 反三角函数应用举例

按位运算函数

!BitAnd(x,y)：按位与。

!BitOr(x,y)：按位或。

!BitXor(x,y)：按位异或。

!BitClear(x,y)：对 x 的位 y 进行清 0。

!BitSet(x,y)：对 x 的位 y 进行置 1 操作。

!BitTest(x,y)：检测 x 对应二进制数的位 y 是否为 1。

!BitLShift(x,y): 将参数 x 左移 y 位。

!BitRShift(x,y): 将参数 x 右移 y 位。

!BitNot(x): 按位取反。

按位运算函数举例:

!BitAnd(x,y): 按位与。

十进制的 3 对应二进制为 011
十进制的 4 对应二进制为 100
执行按位与后, 二进制就为
000, 对应十进制也为 0, 所以
a 为 0

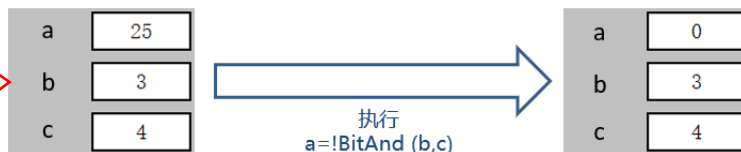


图 11.6-3 按位运算函数应用举例

取整函数:

!Ceil(x): 向上取整。

!Floor(x): 向下取整。

取整函数举例:

!Ceil(x): 向上取整。

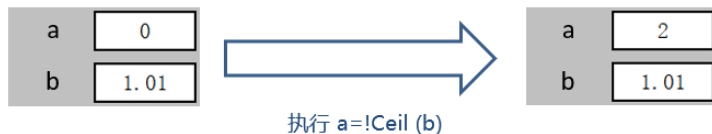


图 11.6-4 取整函数应用举例

其他数学函数:

!Log(x): 以 e 为底数的对数函数。

!Log2(x): 以 2 为底的对数。

!Log10(x): 以 10 为底的对数。

!ToDeg(x): 将弧度转换为角度。

!ToRad(x): 将角度转换为弧度。

!Sqr(x): 平方根函数。

!Exp(x): 以 e 为底数的指数函数。

!Abs(x): 绝对值函数。

!Sgn(x): 符号函数。

!Rand(x,y): 生成随机数, 随机数的范围在 x 和 y 之间。

其他数学函数举例:

!Rand(x,y): 生成随机数, 随机数的范围在 x 和 y 之间。

注意事项:

(1) 当 $x = y$ 时, 在 $(x, x+1)$ 之间生成随机数。

(2) 当 $x > y$ 时, 在 (y, x) 之间生成随机数。

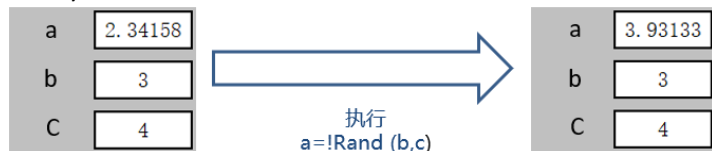


图 11.6-5 随机函数运行效果

11.7 字符串函数

字符串函数一共包含三大类, 分别为“进制转换函数”、“字符串操作函数”和“数据解析函数”。本教程主要介绍 6 个具有代表性的函数, 其他函数的使用请参考 McgsPro 帮助文档。

11.7.1 进制转换函数

- 函数介绍:

!Hex2I(s): 把十六进制字符串转换为十进制数值。

应用举例：十六进制数 11 转换为十进制数是？

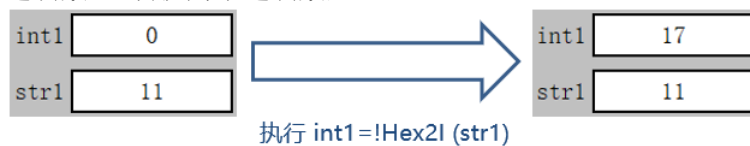


图 11.7-1 进制转换函数应用举例一

请注意：字符串 s 应为 ‘0’ ~ ‘9’、‘a’ ~ ‘f’、‘A’ ~ ‘F’ 或 ‘-’ 字符组成的字符串，‘-’ 在开头才有效，除 ‘-’ 外最多 8 个字符。若指定字符串不是以 0-9、a-f、A-F、‘-’ 开头，则函数执行返回值为 0，表示转换失败。

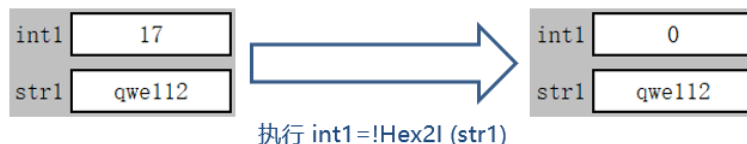


图 11.7-2 进制转换函数应用举例二

- 函数介绍：

!I2Hex(s): 把十进制数值转换为十六进制字符串。

应用举例：十进制数 17 转换为十六进制数是？

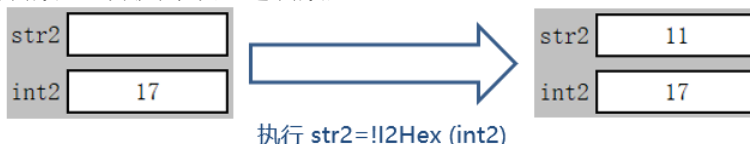


图 11.7-3 进制转换函数应用举例三

11.7.2 字符串操作函数

- 函数介绍：

!Mid(str,n,k): 从字符串变量 str 左边第 n 个字符起，取 k 个字符。

应用举例：从字符串 “aBcDeFg” 左边第 4 个字符起，取 2 个字符，赋值给变量 str3。

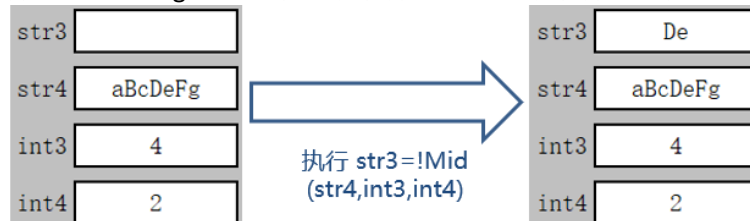


图 11.7-4 字符串操作函数应用举例一

请注意：（1）当 n 以后的字符个数小于 k 时，函数将返回 n 以后的所有字符串；（2）当 k=-1 时，返回 n 以后的所有字符；（3）当 n<1 时，返回空字符串。

- 函数介绍：

!StrComp(str1,str2) : 比较字符串变量 str1 和 str2 是否相等，不区分大小写字母。当返回值为 0，则表示两个字符串相等，反之表示不相等。

应用举例：

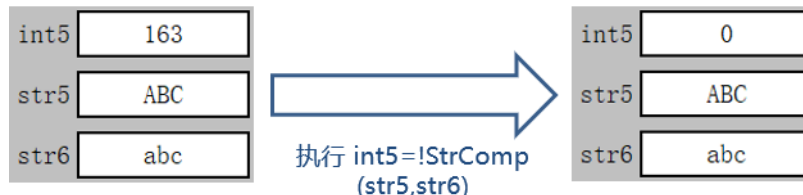


图 11.7-5 字符串操作函数应用举例二_两字符串相同

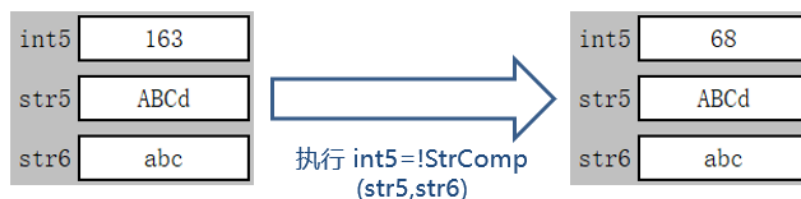


图 11.7-6 字符串操作函数应用举例三_两字符串不同

11.7.3 数据解析函数

● 函数介绍

!SetIntToByteArray (ByteArray,start,n): 对字节数组 ByteArray 中从 start 位置开始的内容进行填充, 填充的长度为 4 字节, 填充的内容为 n。

!SvrGetIntFromByteArray (ByteArray,start,n,Flag): 将字节数组 ByteArray 中从 start 位置开始的内容输出到变量 n 中, 输出的格式为 4 字节整数, 字节数组位置计数从 1 开始。

应用举例:

在实时数据库中新建 5 个整数型变量【in6】至【int10】。

进入用户窗口, 新建 6 个输入框构件, 将其中 5 个输入框构件的操作属性分别关联【in6】至【int10】, 进制设为【十进制】。【int6】的操作属性如图 11.7-7 所示, 其他变量以此类推。



图 11.7-7 输入框构件一操作属性

再将一个输入框构件的操作属性关联【int6】, 进制设为【二进制】。



图 11.7-8 输入框构件二操作属性

新建一个按钮, 将它的脚本函数设置为:

```
1 DIM BA(5) as BYTE
2 ' 定义一个组成员个数为5的字节型数组BA
3 !SetIntToByteArray(BA,2,int6)
4 ' 将int6分为4个字节, 从低位到高位依
5 ' 次将4个字节赋值给BA[2]至BA[5]
6 int7=BA[2]
7 ' 将BA[2]的值赋给int7
8 int8=BA[3]
9 ' 将BA[3]的值赋给int8
10 int9=BA[4]
11 ' 将BA[4]的值赋给int9
12 int10=BA[5]
13 ' 将BA[5]的值赋给int10
```

图 11.7-9 按钮脚本

【int6】的字节 1 至 4 分别被赋值为 BA[2]至 BA[5]。由于 BA[2]至 BA[5]不能直接关联输入框，为了观察函数运行效果，我们又将 BA[2]至 BA[5]赋值给 int7 至 int10 以显示脚本函数运行结果。

保存工程后下载工程，然后启动运行。在以十进制显示的关联【int6】的输入框内任意输入一个数字，可以看到以二进制显示【int6】的输入框显示了该数据的二进制形式。为了便于观察，我们用红线将该二进制值分割成 4 个字节，最高位的第 4 个字节不够 8 位的看作用由 0 补齐。

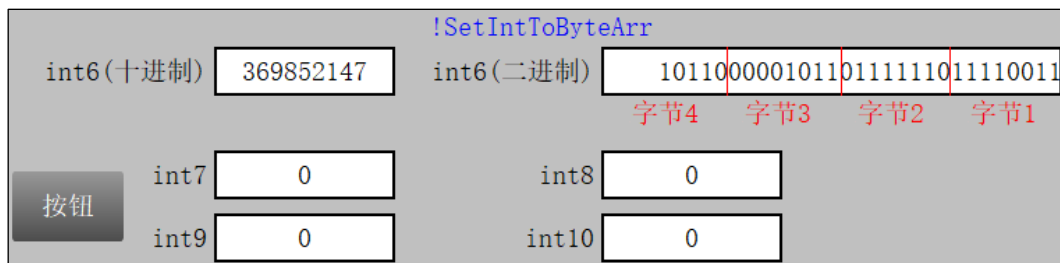


图 11.7-10 运行演示一

点击按钮运行脚本。关联【int7】的输入框显示字节 1 的十进制形式，以此类推。

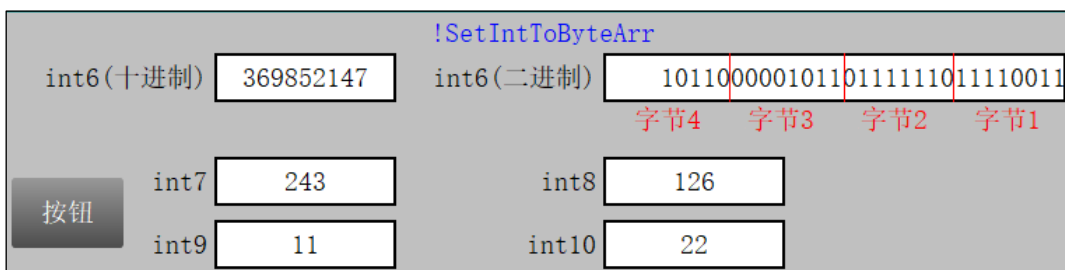


图 11.7-11 运行演示二

11.8 系统变量

McgsPro 内部定义了一些变量，我们称之为 McgsPro 系统变量。在进行组态时，可直接使用这些系统变量。为了和用户自定义的变量相区别，系统变量的名称一律以“\$”符号开头。McgsPro 系统变量多数用于读取系统内部设定的参数，它们只有值的属性，没有最大值、最小值及报警属性。

11.8.1 常用系统变量介绍

\$Date: 读取当前时间：“日期”，字符串格式为：(年-月-日)，年用四位数表示，月日用两位数表示，如 1997-01-09。

\$Time: 读取当前时间：“时刻”，字符串格式为：(时:分:秒)，时、分、秒均用两位数表示，如 20:12:39。

\$Week: 读取计算机系统内部的当前时间：“星期”(1~7)。

\$RunTime: 读取应用系统启动后所运行的秒数，小数部分表示毫秒值。

\$UserName: 在程序运行时记录当前用户的名字。若没有用户登录或用户已退出登录，“\$UserName”为空字符串。

11.8.2 样例演示_显示当天日期

本节样例介绍使用标签构件显示当天的日期。

- ① 进入用户窗口，新建一个标签构件，将它的显示输出关联变量【\$Date】，输出值类型设为【字符串输出】。

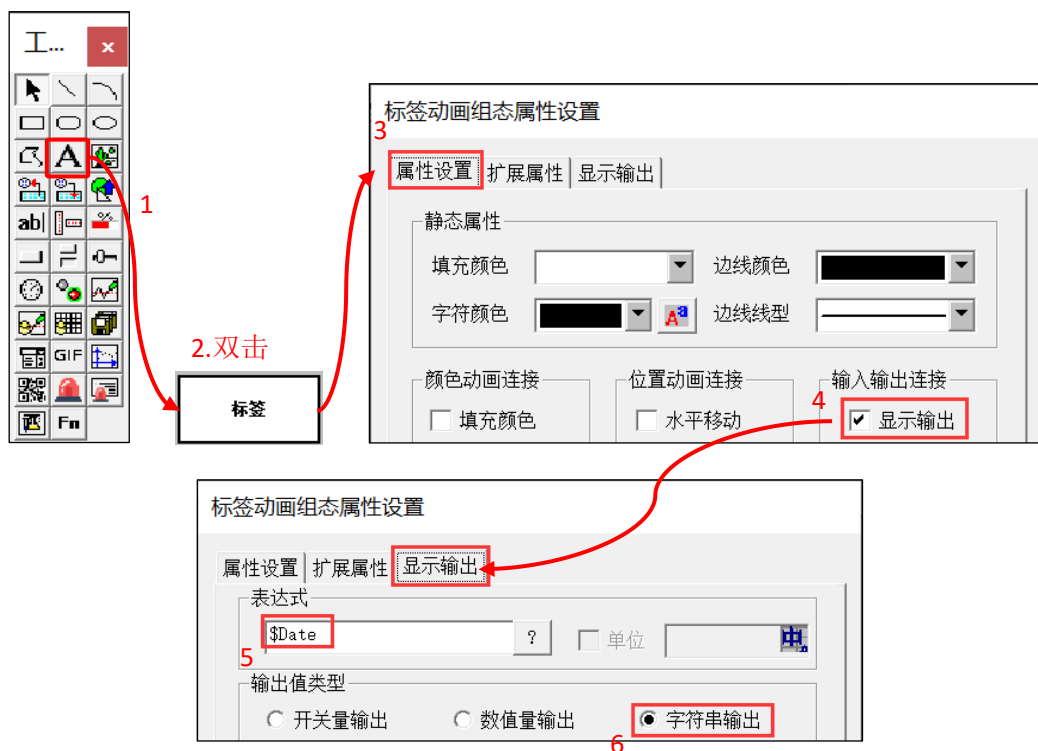


图 11.8-1 标签构件设置

- ② 保存工程后下载工程，然后启动运行。画面中标签构件显示系统当前日期。

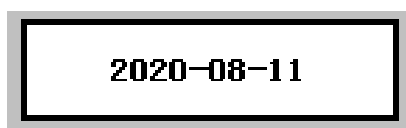


图 11.8-2 运行演示

11.8.3 样例演示_显示当天星期

McgsPro 中的系统变量\$Week 显示的值的范围为 1~7，对应星期一~星期天。如果需要标签构件直接显示星期，可以用一个简单的脚本程序来实现。

首先在实时数据库中新建一个字符串型变量【星期显示】。

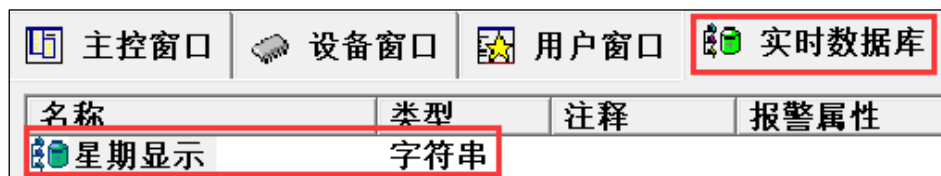


图 11.8-3 新建变量

然后进入工作台 → 运行策略 → 后台任务，后台任务的默认循环周期为 60000ms，用户可根据需要自行设置脚本执行周期。给后台任务添加一条策略行，编辑它的脚本如图 11.8-4 所示。



图 11.8-4 设置后台策略

进入用户窗口，将一个标签构件的显示输出关联变量【星期显示】，输出值类型设为【字符串输出】。保存工程后下载工程，然后启动运行。画面启动后，标签中显示当天的星期。

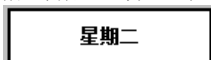


图 11.8-5 运行演示

11.9 文件操作函数

11.9.1 常用函数介绍

以下几个函数为文件操作函数，通过文件操作函数可以在人机界面系统中对文件进行创建、复制、删除、查找、移动等操作。

!GetFreeDiskSpace(): 读取空余存盘空间。

!CreateDirectory(strDirPath): 创建指定路径的文件夹目录。

!FileFind(strFilename,nameArr,sizeArr,attribArr): 查找指定路径下及其子文件下指定条件的文件或文件夹。

!FileDelete(strFilename): 将指定的文件或目录删除。

!FileWriteByStr(strFileName,strValue,mode,encode): 将字符串按照指定编码方式写入指定文件的指定位置。

!FileReadByStr(strFilename,filePos,strLength,strResultRef,encode): 从指定文件指定位置开始，按照指定编码方式读取指定字节的字符串，或一整行，或全部字符串并将结果保存到 strResultRef 变量中。

说明：

1.文件操作涉及路径均以“\$MCGS_DIR_USER/”开始表示用户目录，其他情况则表示 U 盘目录，且不能存在操作系统根目录的写法（即“./”、“/.”、“/./”）。

举例说明：

- ① 如路径为 test.csv，表示 U 盘目录下 test.csv 文件；a/test.csv，表示 U 盘目录下文件夹 a 中的 tes.csv 文件
- ② 使用电脑模拟时[../安装目录/Program/data/user_dir]表示用户目录，[../安装目录/Program/export]表示 U 盘目录。
- 2.文件操作涉及的路径或文件名，在模拟环境下不区分大小写；实际触摸屏运行环境下严格区分大小写。
- 3.若某个路径下存在文件时，在该路径不能创建同名文件夹。

11.9.2 样例演示_文件（夹）的创建、查找和删除

11.9.2.1 创建文件夹

首先在实时数据库中新建两个字符串型变量【strDirPath】和【DC】。

然后在用户窗口中新建一个标准按钮构件，将它的显示文本修改为【创建文件夹】，脚本程序设置如图 11.9-1 所示。

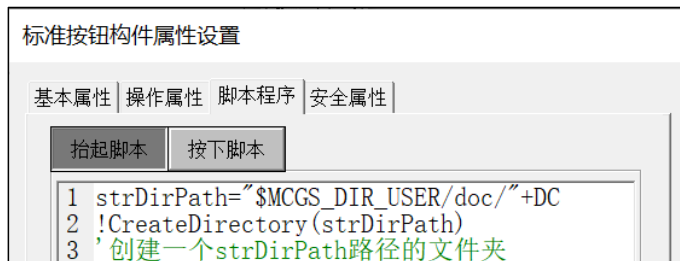


图 11.9-1 创建文件夹功能按钮设置

再新建一个输入框构件，将它的操作属性关联变量【DC】，作为文件夹名称的输入入口。

11.9.2.2 查找文件

在实时数据库中新建 5 个字符串型变量【str000】至【str004】。

然后在用户窗口中新建一个标准按钮构件，将它的显示文本修改为【查找文件】，脚本程序设置如图 11.9-2 所示。

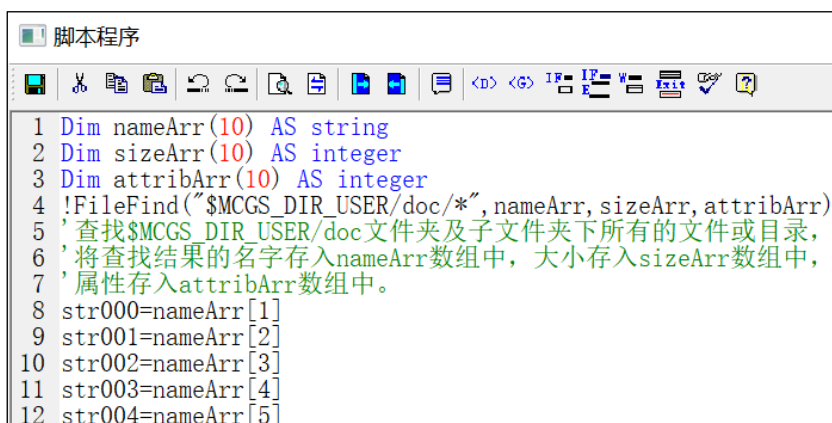


图 11.9-2 查找文件夹功能按钮设置

再创建 5 个标签构件，分别将它们的显示输出关联变量【str000】至【str004】，输出值类型为【字符串输出】。关联变量【str000】的标签构件的显示属性设置如图 11.9-3 所示，其他标签构件以此类推。

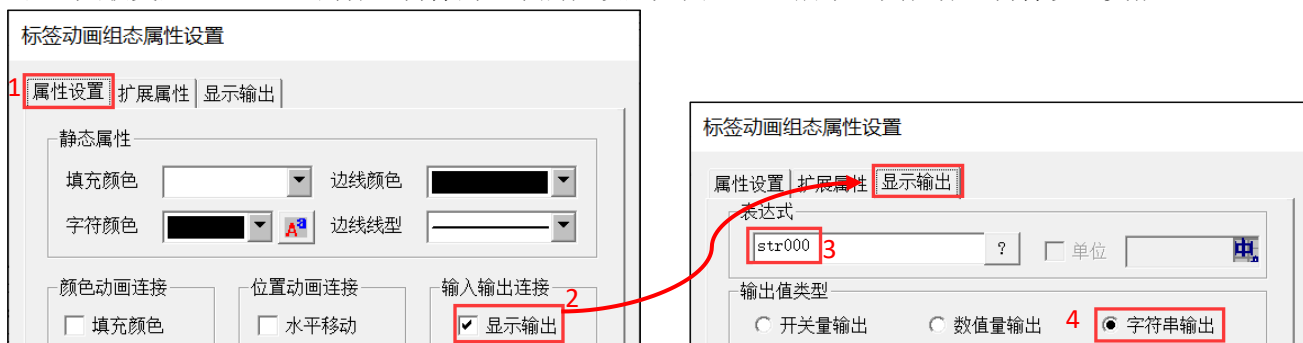


图 11.9-3 将标签关联变量

11.9.2.3 删除文件

首先在实时数据库中新建量给字符串型变量【strFilename】和【DC1】。

然后在用户窗口中新建一个标准按钮构件，将它的显示文本修改为【删除文件】，脚本程序设置如图 11.9-4 所示。

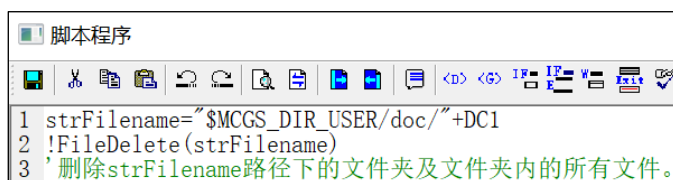


图 11.9-4 删除文件功能按钮设置

再新建一个输入框构件，将它的操作属性关联变量【DC1】

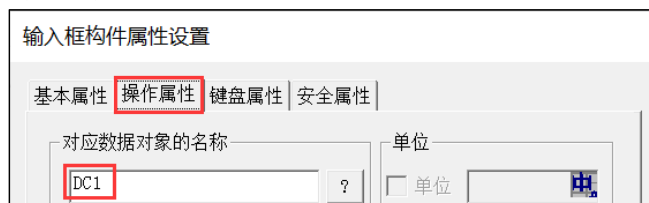


图 11.9-5 输入框构件设置

11.9.2.4 模拟运行演示

组态完成后保存工程，然后下载工程，下载成功后启动运行。在关联【DC】的输入框内输入想要创建的文件夹的名称，然后点击【创建文件夹】按钮。



图 11.9-6 运行演示_创建文件夹

此时的安装目录/Program/data/user_dir 下，创建了一个名为【doc】的文件夹，文件夹中又创建子文件夹【aaa】。



图 11.9-7 运行演示_创建文件夹效果

在【doc】文件夹中新建数个文件（夹），如图 11.9-8 所示。

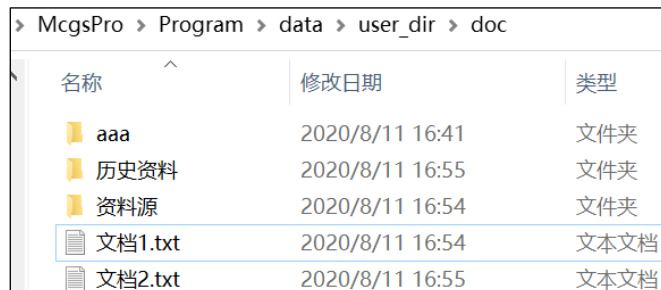


图 11.9-8 运行演示_手动在【doc】文件夹中新建数个文件（夹）

回到模拟运行界面，点击【查找文件】按钮，系统查找到【doc】文件夹中的文件，由关联好变量的标签构件显示。



图 11.9-9 运行演示_查找文件效果

在关联【DC1】的输入框中输入【资料源】，点击【删除文件】按钮。可以看到该文件夹被删除。

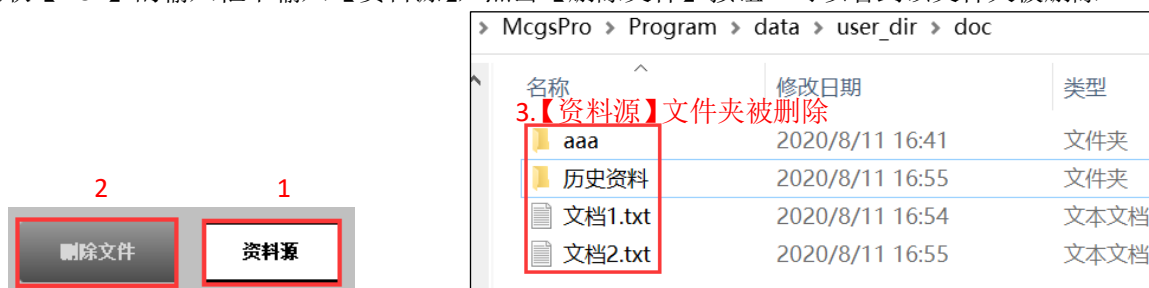


图 11.9-10 运行演示_删除文件效果

11.9.3 样例演示_文件写入和文件读取

继续使用 11.9.2 中的样例。

11.9.3.1 创建并写入

首先在实时数据库中新建一个字符串型变量【strValue】。

然后在用户窗口中新建一个标准按钮，将它的显示文本修改为【创建并写入】，脚本程序设置如图 11.9-11 所示。

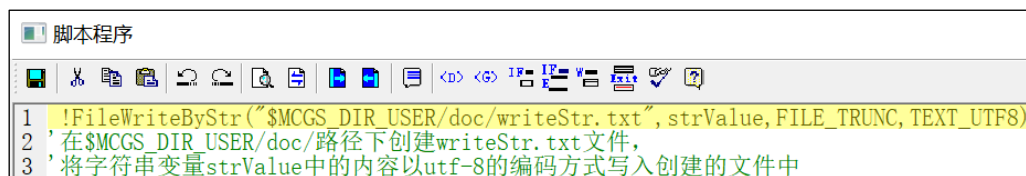


图 11.9-11 创建并写入功能按钮设置

再新建一个输入框构件，将它的操作属性关联变量【strValue】。

11.9.3.2 追加写入

再在用户窗口中新建一个标准按钮，将它的显示文本修改为【追加写入】，脚本程序设置如图 11.9-12 所示。

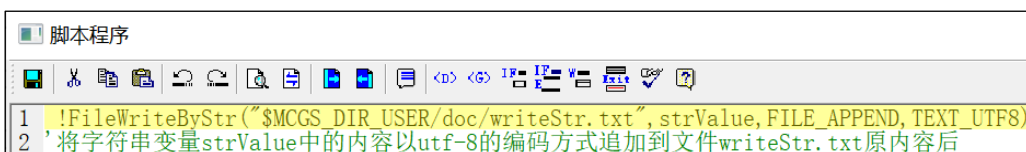


图 11.9-12 追加写入功能按钮设置

11.9.3.3 读取内容

在实时数据库中新建一个字符串型变量【strResultRef】。

然后在用户窗口中新建一个标准按钮，将它的显示文本修改为【读取】，脚本程序设置如图 11.9-13 所示。

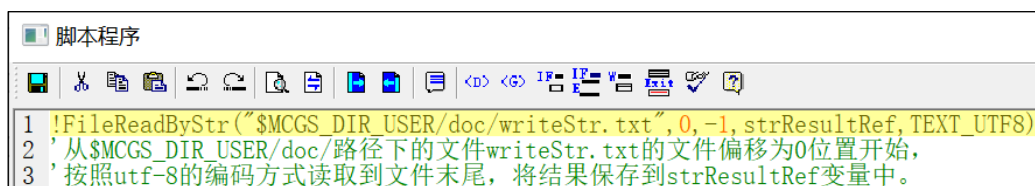


图 11.9-13 读取内容功能按钮社会组

再新建一个输入框构件，将它的操作属性关联变量【strResultRef】。

11.9.3.4 模拟运行演示

组态完成后保存工程，然后下载工程，下载成功后启动运行。在关联【strValue】的输入框内输入想要创建的文件内容，比如【昆仑通态】，然后点击【创建并写入】按钮。



图 11.9-14 运行演示_创建文件并写入内容

此时的安装目录/Program/data/user_dir/doc 下，创建了一个名为【writeStr.txt】的文件。使用记事本打开该文件，可以看到文件内容如图 11.9-15 所示。

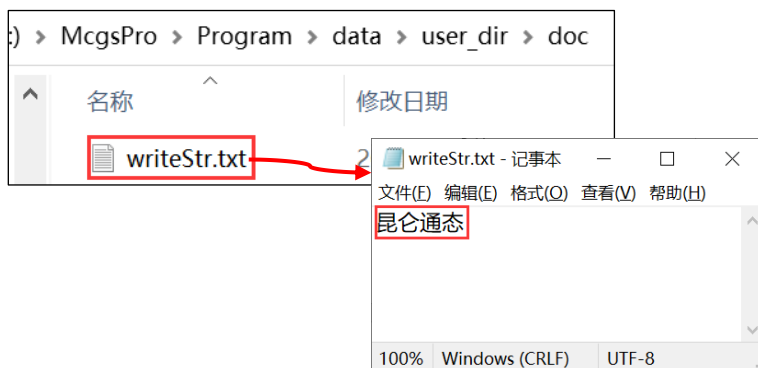


图 11.9-15 运行演示_打开创建的文件

点击【读取】按钮，将【writeStr.txt】文件中的内容读取到模拟屏上，如图 11.9-16 所示。



图 11.9-16 运行演示_读取内容

再在关联【strValue】的输入框内输入想要追加的文件内容，比如【创世界精品】，然后点击【追加写入】按钮。点击【读取】，可以看到系统在上一步写入的内容后面增加了新的内容。

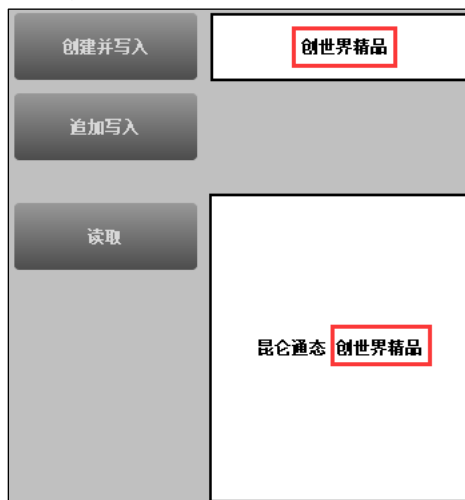


图 11.9-17 运行演示_追加写入

11.10 内存操作函数

11.10.1 常用函数介绍

以下 4 个函数为内存操作函数，通过内存操作函数可以在人机界面系统内存中进行创建缓冲区、清除缓冲区数据、释放缓存、获取缓存大小、加载字符串到缓冲区等操作。

!BufferCreate(bufID,bufSize,byteOrder,encode): 创建一个 ID 为 bufID，大小 bufSize，字节排序方式为 byteOrder，编码格式为 encode 的缓冲区。

!BufferSetInt(bufID,pos,intValue,type): 将 type 类型的整型值 intValue 写入 ID 为 bufID 的缓冲区的 pos 位置。

!BufferGetInt(bufID,pos,intValueRef,type): 获取 ID 为 bufID 的缓冲区 pos 位置的 type 型整型值，赋给变量 intValueRef。

!BufferWriteToCsvStr(bufID,bufPos,formatStr,count,csvStrRef): 将 ID 为 bufID 的缓冲区，bufPos 位置开始的数据，按 formatStr 格式导出 count 行到 csvStrRef 字符串变量中。

说明：

- 1.缓冲区个数：最多 128 个。
- 2.缓冲区总大小：小于等于 1M。
- 3.缓冲区 ID：0-127。
- 4.涉及路径标识时：

\$MCGS_DIR_USER: 表示用户分区目录

非\$开头: U 盘分区目录, 如 DOC/a.CSV, 表示 U 盘中 DOC 文件夹下的 a.csv 文件

11.10.2 样例演示

① 创建缓冲区:

在用户窗口中新建一个标准按钮, 将它的显示文本修改为【创建缓冲区】, 脚本程序设置如图 11.10-1 所示。



图 11.10-1 创建缓冲区功能按钮设置

② 将整数写入缓冲区:

在实时数据库中新建一个整数型变量【A】。

然后在用户窗口中新建一个标准按钮, 将它的显示文本修改为【将整数写入缓冲区】, 脚本程序设置如图 11.10-2 所示。

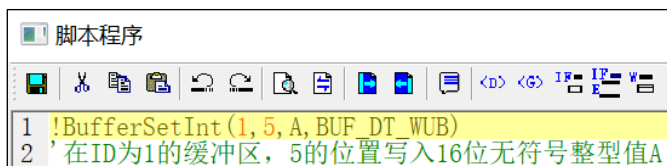


图 11.10-2 将整数写入缓冲区功能按钮设置

再新建一个输入框构件, 将它的操作属性关联变量【A】, 格式为【十进制】。

③ 从缓冲区读取整数:

在实时数据库中新建一个整数型变量【B】。

然后在用户窗口中新建一个标准按钮, 将它的显示文本修改为【从缓冲区读取整数】, 脚本程序设置如图 11.10-3 所示。

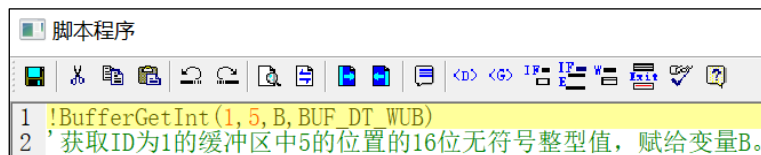


图 11.10-3 从缓冲区读取整数功能按钮设置

再新建一个标签构件, 将它的显示输出关联变量【B】, 输出值类型为【数值量输出】。

④ 缓冲区数据写到字符串:

在实时数据库中新建一个字符串型变量【csvStrRef】。

然后在用户窗口中新建一个标准按钮, 将它的显示文本修改为【缓冲区数据写到字符串】, 脚本程序设置如图 11.10-4 所示。

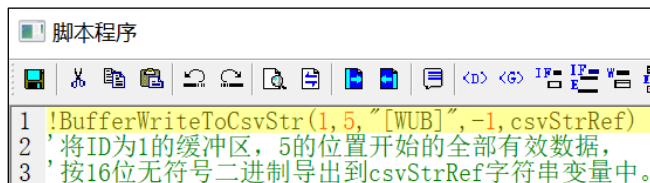


图 11.10-4 缓冲区数据写到字符串功能按钮设置

再新建一个标签构件, 将它的显示输出关联变量【csvStrRef】, 输出值类型为【字符串输出】。

⑤ 模拟运行演示:

组态完成后保存工程, 然后下载工程, 下载成功后启动运行。

首先点击【创建缓冲区】按钮, 在内存中新建一个 ID 为 1 的缓冲区。



图 11.10-5 运行演示_创建缓冲区

在关联变量【A】的输入框中任意输入一个数字，比如【789】，点击按钮【将整数写入缓冲区】，将【789】存入缓冲区指定位置。然后点击【从缓冲区读取整数】按钮，将缓冲区指定位置内容赋值给变量【B】，可以看到关联变量【B】的标签构件的显示内容为【789】。

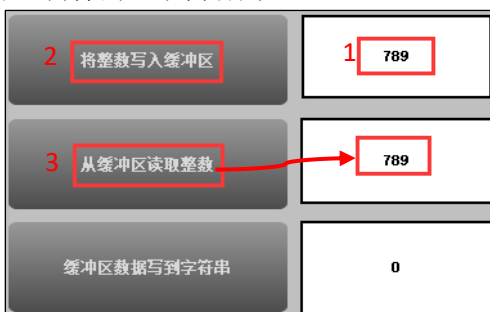


图 11.10-6 运行演示_读写缓冲区内容

点击【缓冲区数据写到字符串】按钮，将缓冲区的指定位置内容赋值给字符串变量【csvStrRef】，可以看到关联变量【csvStrRef】的标签构件的显示内容为【789】。

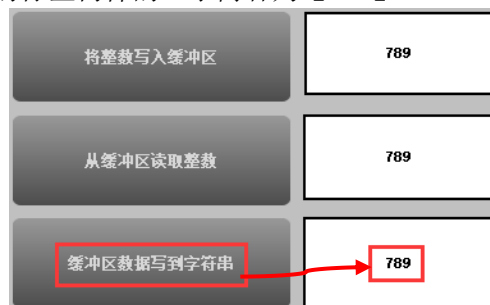


图 11.10-7 运行演示_缓冲区数据写到字符串

11.11 计时器函数

McgsPro 可用的系统计时器范围为 0 到 127，即系统内嵌 128 个系统计时器。用户可以随意使用其中的任意一个。

计时器返回时间值为浮点数，单位为秒，小数位表示毫秒。因为采用浮点数表示，随着数值增大会略有误差。

11.11.1 常用函数介绍

!TimerRun(计时器号): 启动计时器开始工作。

!TimerStop(计时器号): 停止计时器工作。

!TimerReset(计时器号,数值): 设置计时器的当前值。

!TimerValue(计时器号): 取计时器的当前值。

!TimerState(计时器号): 取计时器的工作状态。

!TimerSetLimit(计时器号,上限值,参数 3): 设置计时器的上限值及运行模式。

!TimerSetOutput(计时器号,变量): 将计时器的值输出给参数二。

!TimerWaitFor(定时器,数值): 等待定时器工作到参数二指定的值后, 脚本程序才向下执行。

11.11.2 样例演示_启动、停止、获取当前值、获取状态

- ① 首先在实时数据库中新建 1 个浮点数值型变量【计时器状态】。
- ② 进入用户窗口 0, 在窗口中新建 2 个标准按钮构件, 将它们的显示文本分别设置为【启动计时器】和【停止计时器】。

其中, 【启动计时器】按钮的脚本为:

!TimerRun(0) '启动计时器0

【停止计时器】按钮的脚本为:

!TimerStop(0) '停止计时器0

- ③ 双击窗口空白处, 设置窗口的循环脚本时间设置为 100ms, 循环脚本设置如图 11.11-1 所示:

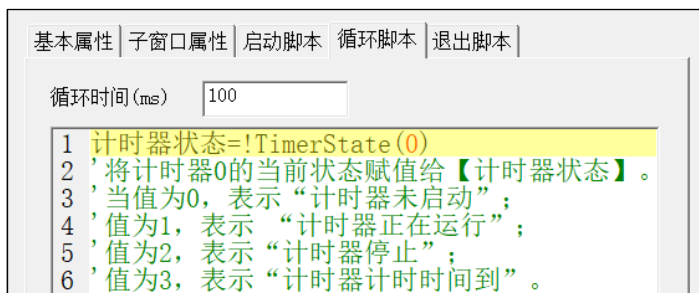


图 11.11-1 窗口循环脚本设置

- ④ 新建一个【动画显示】构件。双击进入基本属性 → 分段点 0 → 外形 → 图像 0 → 点击图库按钮 → 扁平风格 → 图标 → 白色背景_按钮抬起 → 确定。

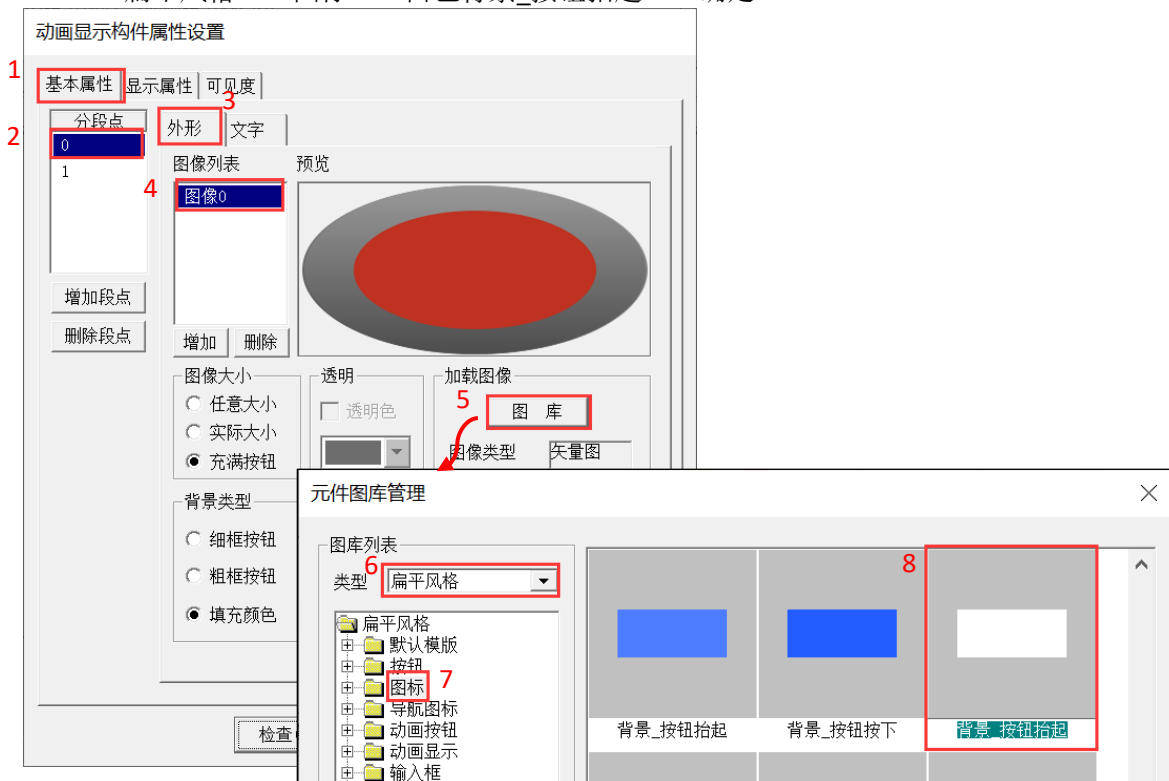


图 11.11-2 动画显示分段点 0 图像设置

然后点击基本属性分段点 0 中的文字 → 点击增加按钮 → 选中出现的文本 0 → 修改文本内容为【计时器未启动】。

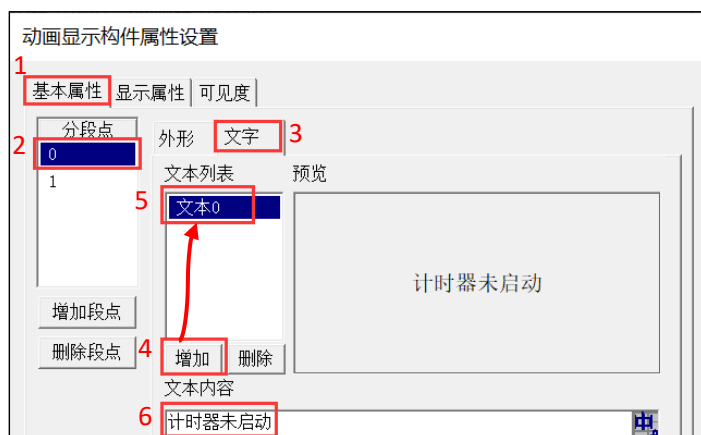


图 11.11-3 动画显示分段点 0 文本设置

点击基本属性中的【增加段点】按钮增加 2 个分段点【分段点 2】和【分段点 3】。按照前面的步骤，将【分段点 1】至【分段点 3】的外形统一设置成和【分段点 0】一致，文本显示分别为【计时器正在运行】【计时器停止】【计时器时间到】。

然后进入【显示属性】功能页，设置显示变量类型为【数值显示】，关联的变量为【计时器状态】，切换方式为【根据变量值切换】。



图 11.11-4 动画显示显示属性设置

- ⑤ 保存工程后下载工程，然后启动运行。此时画面中关联【计时器状态】的动画显示构件显示内容为【计时器未启动】。



图 11.11-5 运行演示_刚开始运行时的画面

点击【启动计时器】按钮，计时器状态变为【计时器正在运行】。

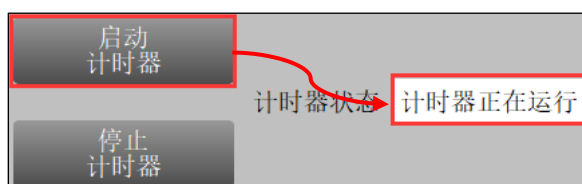


图 11.11-6 运行演示_启动计时器

点击【停止计时器】按钮，计时器状态变为【计时器停止】。

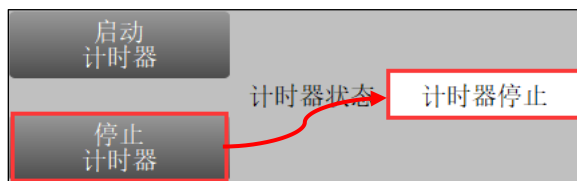


图 11.11-7 运行演示_停止计时器

11.11.3 样例演示_获取当前值、重置当前值

继续使用 11.11.2 中的样例工程。

- ① 首先在实时数据库中新建 2 个浮点型变量【当前值】和【重置当前值】。
- ② 进入用户窗口 0，双击窗口空白处，将窗口的启动脚本设置如图 11.11-8 所示：

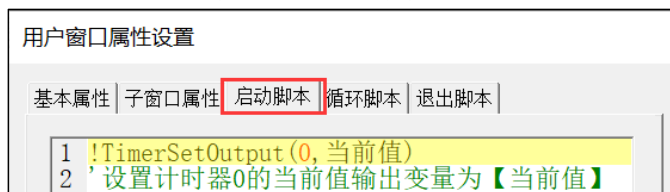


图 11.11-8 窗口启动脚本

- ③ 新建一个标签构件，将它的显示输出关联变量【当前值】，单位设置为【秒】，输出值类型为【数值量输出】，取消勾选输出格式中的【浮点输出】和【自然小数点】，默认输出格式为【十进制】。



图 11.11-9 标签构件显示属性设置

- ④ 新建一个标准按钮构件，将它的显示文本设置为【重置当前值】，脚本设置为：
!TimerReset(0, 重置当前值) '将计时器0的当前值重置为变量【重置当前值】的值'
- ⑤ 新建一个输入框构件，将它的操作属性关联变量【重置当前值】，单位设为【秒】。



图 11.11-10 输入框属性设置

- ⑥ 保存工程后下载工程，然后启动运行。点击【启动计时器】按钮，关联变量【当前值】的标签构件的显示值以每秒加一的速度递增。

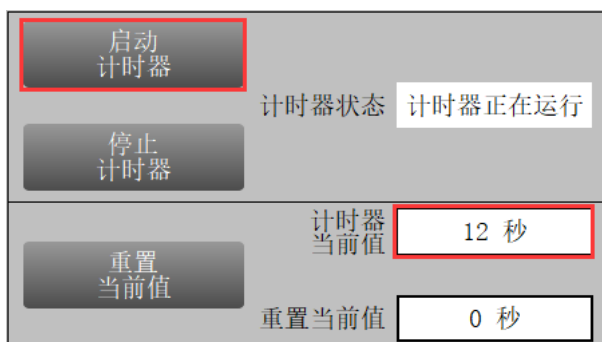


图 11.11-11 运行演示_启动计时器

- ⑦ 将关联变量【重置当前值】设为【0】，点击【重置当前值】按钮，可以看到，关联变量【当前值】的标签的显示内容变成了【0】，重置成功。

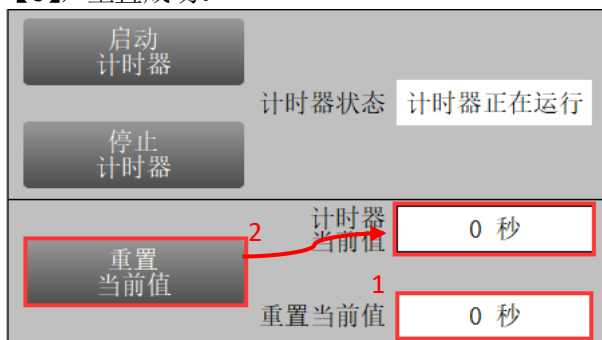


图 11.11-12 运行演示_重置计时器值

11.11.4 样例演示_锁定屏幕 N 秒

继续使用 11.11.3 中的样例工程。

- ① 在实时数据库中新建一个浮点数变量【锁定时间】，将它的初值设定为【5】。



图 11.11-13 新建变量并设定其初值

- ② 进入工作台 → 用户窗口 → 新建窗口。

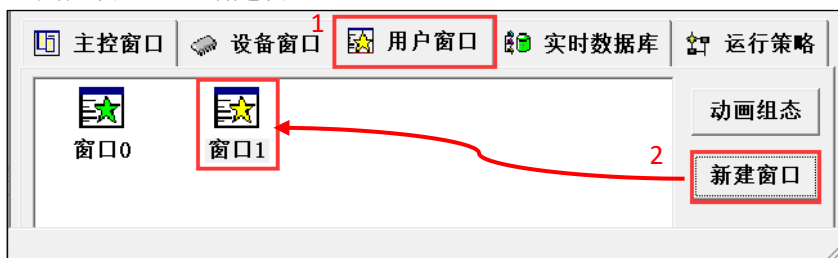


图 11.11-14 新建窗口

- ③ 双击新建的窗口 1 进入窗口。双击窗口空白处，弹出【用户窗口属性设置】界面，将窗口名称修改为【屏幕锁定】。窗口背景设置为【浅蓝色】。

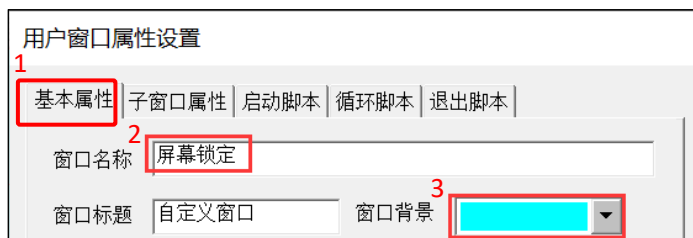


图 11.11-15 设置窗口基本属性

- ④ 进入【子窗口属性】功能页，勾选【作为子窗口使用】前的复选框，设置窗口宽度为【160】，窗口高度为【110】，其他设置保持默认不变。



图 11.11-16 设置窗口子窗口属性

- ⑤ 进入【循环脚本】功能页，设置循环时间为【100】ms，循环脚本程序如图 11.11-17 所示，它的作用是计时器 0 运行 t 秒后，关闭子窗口【屏幕锁定】。

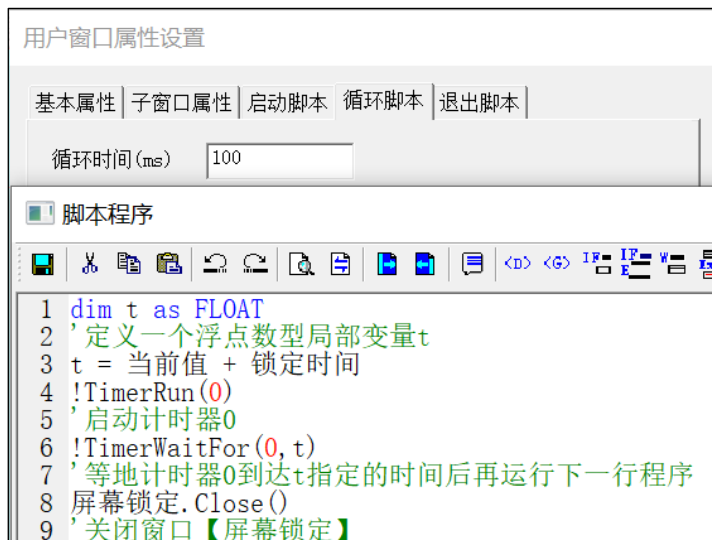


图 11.11-17 设置窗口循环脚本

- ⑥ 新建一个标签构件，将它的文本设置为【屏幕锁定中！请稍等！】，填充颜色为【没有填充】和边线颜色为【没有边线】。

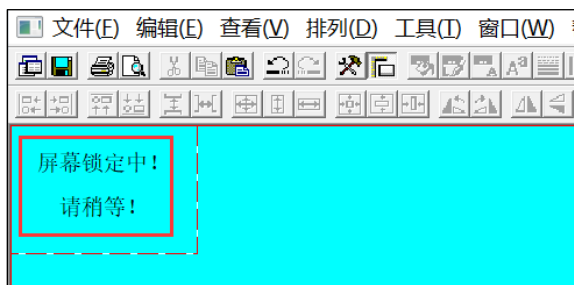


图 11.11-18 新建提示内容标签

- ⑦ 进入用户窗口 0，新建一个按钮，将按钮的显示文本设置为【锁定屏幕】，操作属性设置为打开用户窗口【屏幕锁定】。

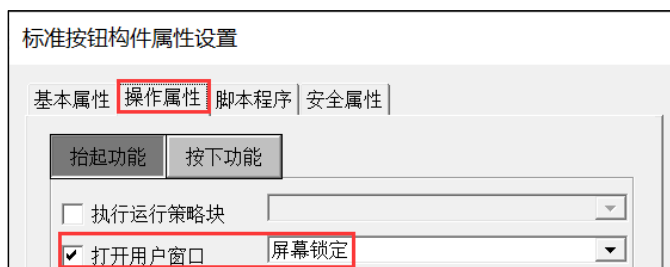


图 11.11-19 设置按钮属性

- ⑧ 新建一个【输入框】构件，将它的操作属性关联变量【锁定时间】，单位设置为【秒】。



图 11.11-20 设置输入框属性

- ⑨ 保存工程后下载工程，然后启动运行。由于已经将变量【锁定时间】的初值设置为【5】，故画面刚开始运行时，关联变量【锁定时间】的输入框显示的值即为【5】。点击【锁定屏幕】按钮，屏幕正中间弹出子窗口【屏幕锁定】。由于我们将子窗口默认设置为【模态窗口】，此时点击子窗口外的构件操作无效。

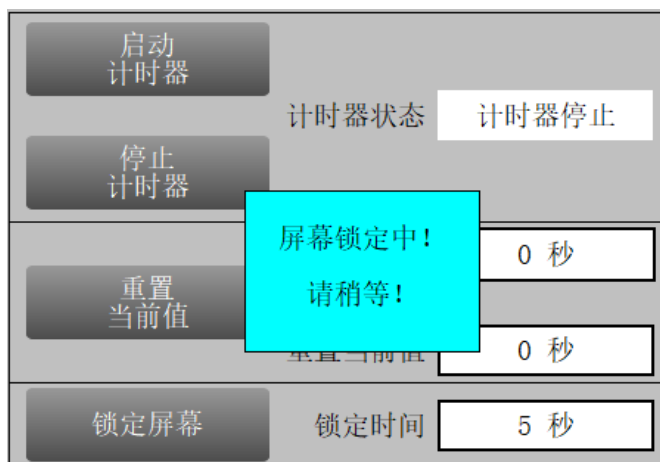


图 11.11-21 运行演示_锁定屏幕

- ⑩ 5 秒之后，计时器等待时间到，子窗口自动关闭。我们也可以根据需求通过输入框设置屏幕的锁定时

间。

启动 计时器	计时器状态	计时器正在运行
停止 计时器		
重置 当前值	计时器 当前值	6 秒
	重置当前值	0 秒
锁定屏幕	锁定时间	5 秒

图 11.11-22 运行演示_锁屏时间到

第 12 章 多语言功能应用实例

当用户需要在多个国家使用同一工程，或将工程提供给工厂中使用不同语言的人员操作时，工程可使用 McgsPro 的多语言功能。

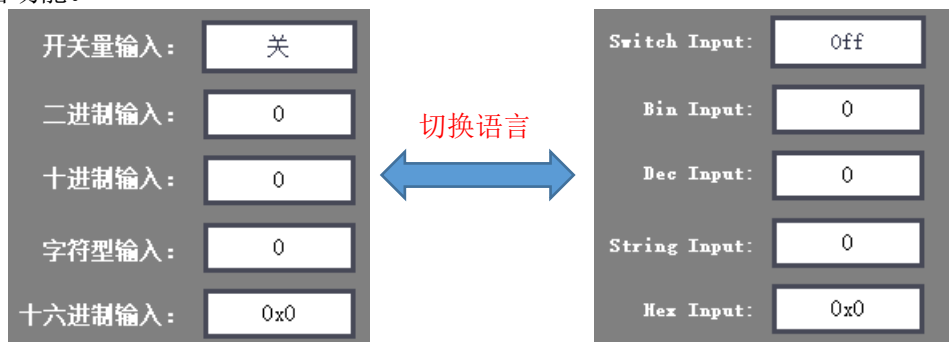


图 12-1 多语言功能切换效果

多语言支持的模块有内置语言和组态语言。

- ① 内置语言：系统设置界面，内置键盘（数值、字符、开关），提示窗口，时间设置窗口等，这些窗口显示的固定字符串为内置语言。
- ② 组态语言：各动画构件的显示内容、报警的描述信息以及用户与用户组的描述内容。

12.1 多语言：中文、英语

本节将组态一个使用了多语言功能的标签，同时组态两个具备切换语言功能的按钮，当点击按钮时，标签显示的语言在中文和英语之间切换。

- 使用系统默认语言（中文）组态工程。

- ① 将一个标签按钮从工具箱拖拽到画面中合适的位置和大小，此时的多语言配置按钮是默认状态：不使用多语言。



图 12.1-1 文本的多语言默认状态

- ② 拖拽两个标准按钮拖拽到标签构件的下方，将其中一个按钮的文本设置为【中文】，抬起脚本设置为【!SetCurrentLanguageIndex(0)】，如图 12.1-2 所示。!SetCurrentLanguageIndex(语言 ID)的作用是通过语言 ID 设定当前语言环境，本例中括号内的参数【0】是中文的语言 ID，各语言的语言 ID 可从【语言选择对话框】中查看，如图 12.1-9。将另一个按钮的文本设置为【英文】，抬起脚本设置为【!SetCurrentLanguageIndex(1)】，如图 12.1-3 所示。它的作用是将当前语言环境设置为英语，括号内的参数【1】是英语的语言 ID。

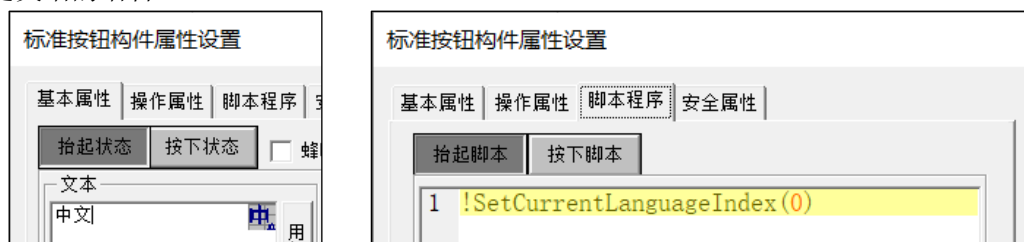


图 12.1-2 设置切换中文按钮

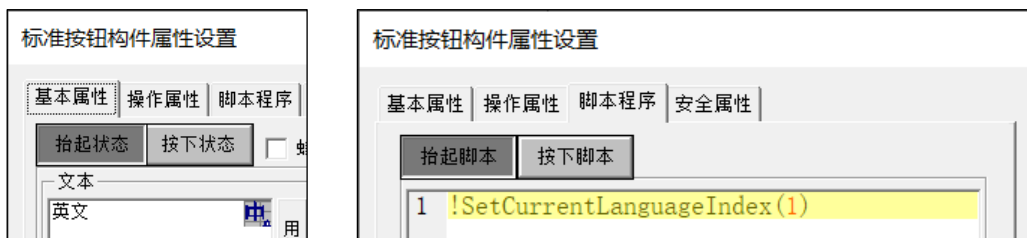


图 12.1-3 设置切换英语按钮

③ 组态好的画面如图 12.1-4 所示：



图 12.1-4 默认语言画面

● 设置工程语言并编辑工程多语言内容

① 点击菜单栏 → 工具 → 多语言或工具栏上的【多语言】图标按钮进行多语言设置。

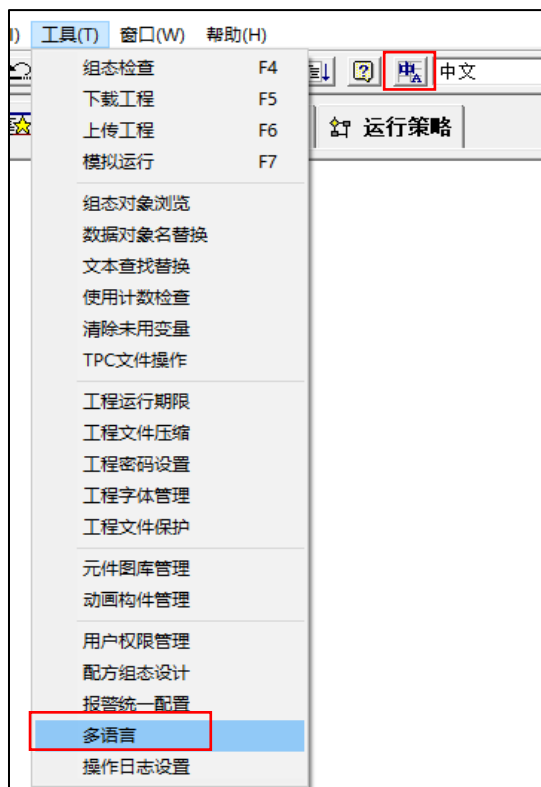


图 12.1-5 进入多语言设置界面

② 如果是第一次设置，会弹出对话框询问是否进行文本收集，如图 12.1-6 所示，选择【是】。文本收集功能是指当前组态工程中，部分构件或功能模块的某些文本项目，未使用多语言，可通过文本收集功能统一收集这些未使用多语言的文本，并自动给它们分配多语言 ID，使之成为多语言项目并可在多语言中心进行统一编辑，无需逐个构件或功能模块进行编辑。

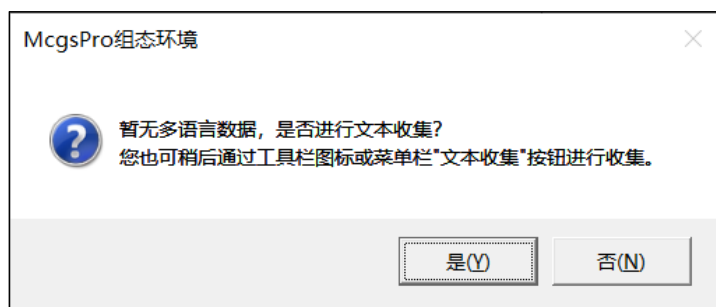


图 12.1-6 询问是否进行文本收集的对话框

③ 在【多语言文本收集】窗口选择【全选】 → 确定 → 是。

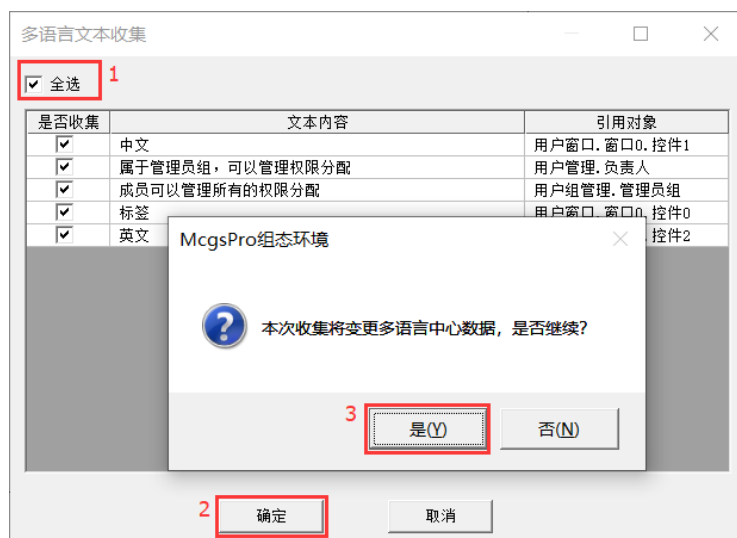


图 12.1-7 多语言文本收集窗口

④ 弹出的【多语言配置】窗口中，此时只有语言 ID 为 0 的【中文】列，该列中包含了系统默认生成的文本和我们刚才组态的构件中的文本，每条文本的左侧都显示了该文本对应的【文本 ID】，比如文本【中文】的文本 ID 为【0】，【标签】的文本 ID 为【3】，【英文】的文本 ID 为【4】。然后点击工具栏上的【打开语言选择对话框】 图标按钮。

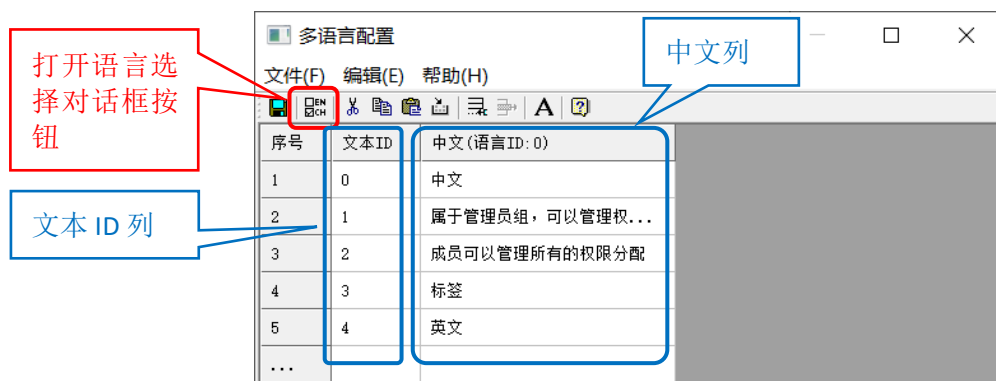


图 12.1-8 多语言配置窗口

⑤ 在弹出的窗口中勾选需要用到的其他语言，本例选择语言 ID 为 1 的英语 → 确定。在该窗口中可以选择工程的启动语言。下载工程后、工程未启动时，屏幕的【SETTING 界面】的语言也会变成该语言。



图 12.1-9 语言选择对话框

⑥ 可以看到，此时的多语言配置窗口中，多了一列【英语】列。

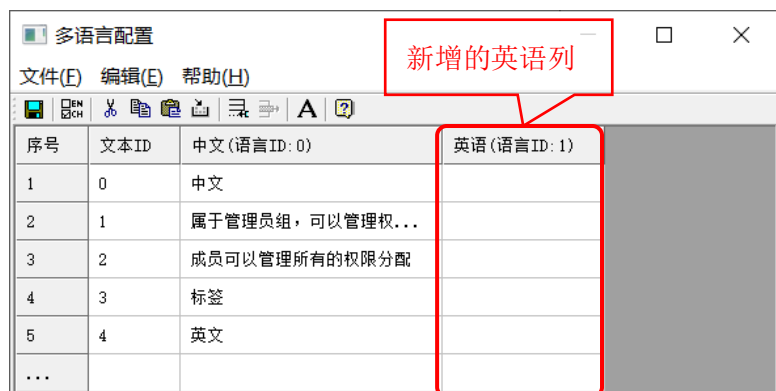


图 12.1-10 新增的英语列

⑦ 手动在英语列中输入中文文本对应的英语文本 → 保存 → 关闭。

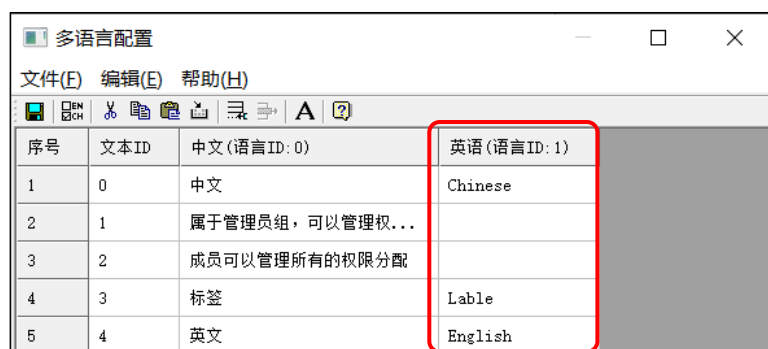


图 12.1-11 手动翻译英语

⑧ 组态完成后，标签、按钮中的文本已经具备了多语言功能。

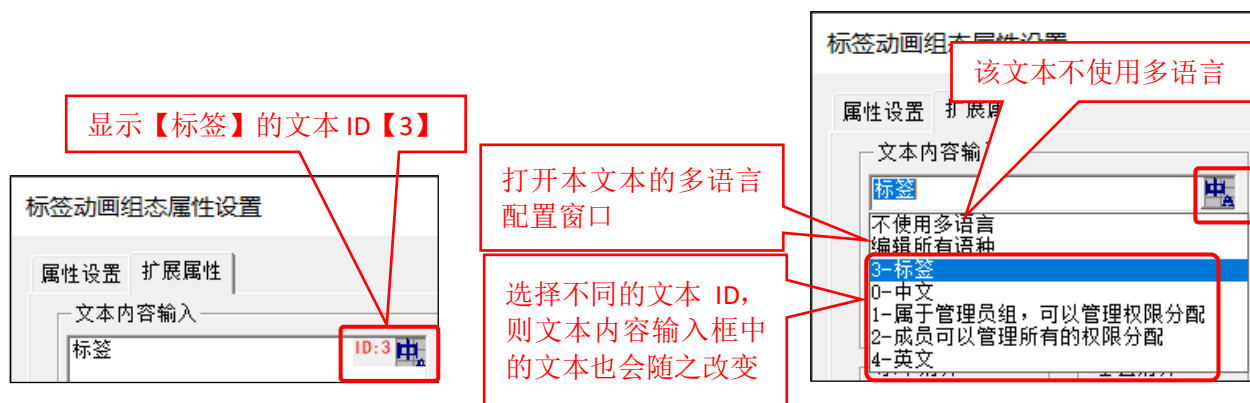


图 12.1-12 标签的多语言

⑨ 保存工程后下载工程，然后启动运行。刚开始运行时，画面默认显示中文，点击切换英语的按钮后，画面中设置了多语言功能的文本切换成英语；点击中文按钮后，文本又被切换成中文。

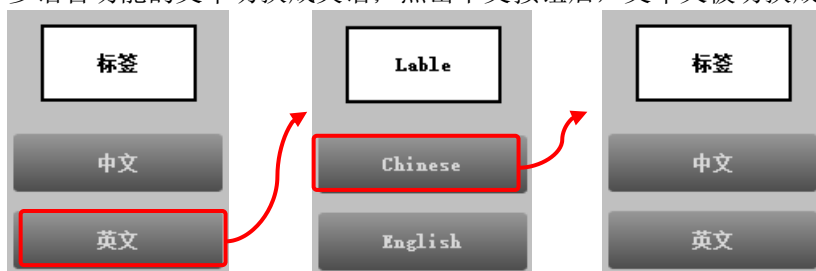


图 12.1-13 多语言工程运行效果

12.2 多语言：中文、自定义语种

McgsPro 组态软件的多语言功能提供了 12 种语言可供选择，当这 12 种语言不能满足用户使用需求时，用户也可以通过【增加语言种类】来自定义一个新语言。本节介绍如何增加一个语言种类。

- ① 在【语言选择对话框】界面点击【增加语言种类】按钮，进入【增加语言种类】设置界面。在【增加语种】处给新语言命名，然后按该语言的具体情况，选择【工程语种设置】和【内置语种设置】中的【语言字体】和【缩放比例】→ 确定。【语言字体】是 Windows 自带字体，用户根据添加的自定义语种情况自行选择。【使用继承语种属性】被勾选时，新增语种的字体与本对话框设置的字体一致，字形、字号设置将与表格中选择的“继承语种”保持一致；如果构件通过字体属性页重新设置了字体、字形和字号，则以重新设置之后的属性为准。

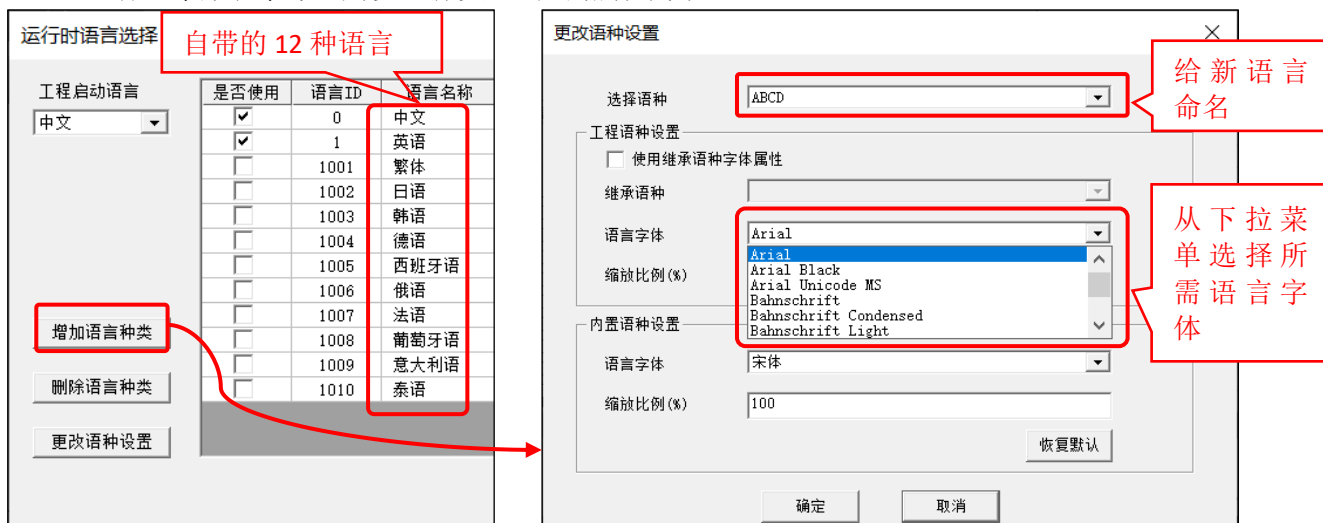


图 12.2-1 增加语言种类流程

- ② 设置完成后，新增加的语言出现在【语言选择对话框】列表中。还可以通过【删除语言种类】按钮对

除【工程启动语言】外的语言进行删除。也可以通过【更改语种设置】按钮对语言设置进行修改。



图 12.2-2 增加语言种类设置效果

- ③ 设置完成后，【多语言配置】窗口中增加了一列语言【ABCD】，需要手动在该列输入对应的文本。

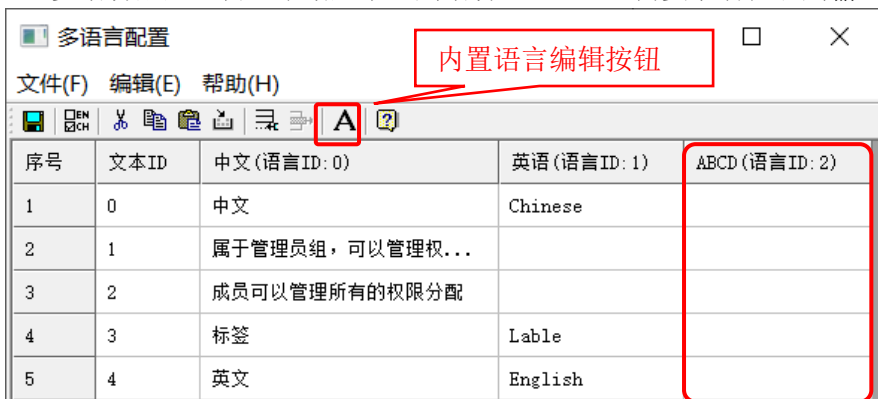


图 12.2-3 增加语言种类在多语言配置窗口的效果

- ④ 点击工具栏上的【内置语言编辑】按钮，进入内置语言编辑窗口。将【中文】列的文本翻译成新语言【ABCD】后，在对应位置进行输入，然后保存并关闭。

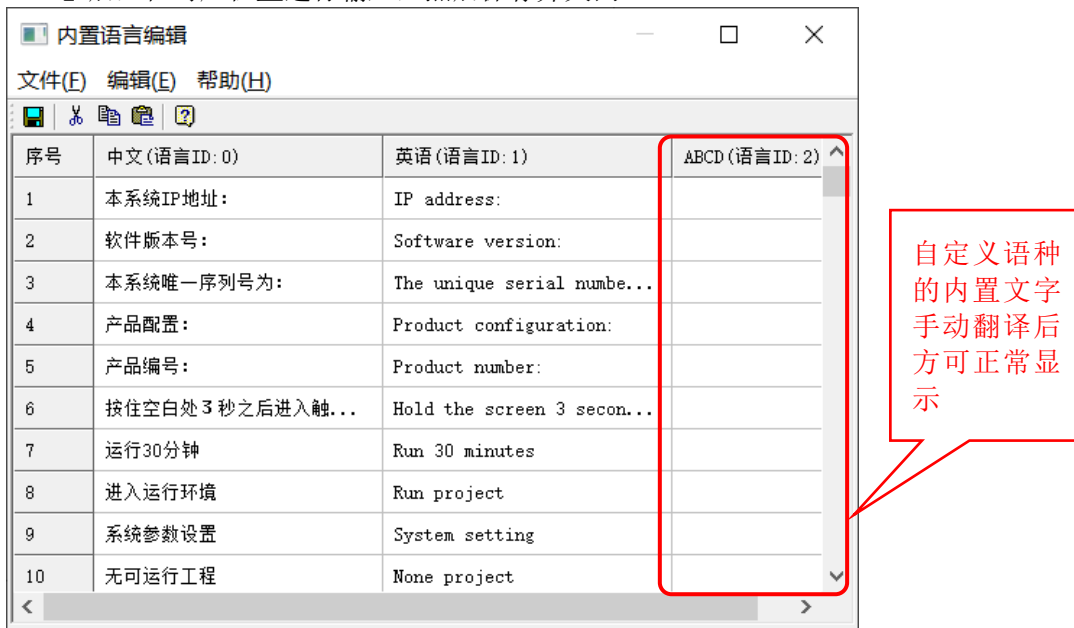


图 12.2-4 翻译内置语言为新语言

12.3 多语言的导入导出

当需要对多语言文本进行批量编辑时，可以使用多语言的【导入\导出】功能，将多语言文本导出为 XML 格式后，使用 Excel 打开并编辑。

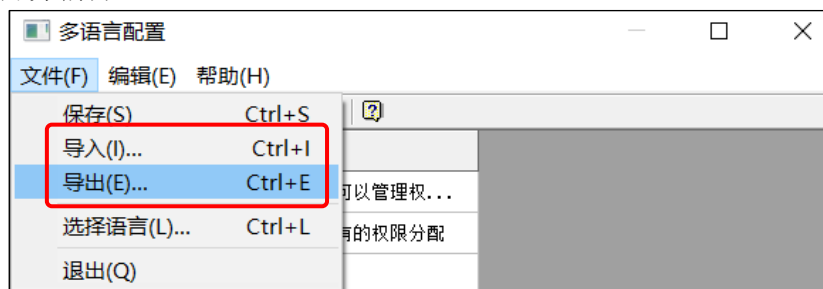


图 12.3-1 多语言的导入导出界面

- ① 导出的内容进行编辑时不可修改其格式，否则会导入失败；
- ② 导入文件的格式必须是 xml(utf8)，列标题(语言名称)不能是纯数字或特殊符号，需遵守 xml 节点命名规则；
- ③ 使用 Excel 编辑 xml 文件时插入 xml 节点，需按住最后一条数据右下角往下拖动的方式增加（如图 12.3-2 所示红框部分）；

	A	B
1	ID	中文
2	0	属于管理员组，可以管理权限分配
3	1	成员可以管理所有的权限分配
4		
5		
6		
7		
8		

图 12.3-2 多语言导出为 xml 格式

- ④ 导出时已默认在 xml 文件之后追加 10000 个空白节点，方便用户编辑。

第 13 章 安全功能应用实例

McgsPro 提供了一套完善的安全机制，包括工程运行安全机制和工程师权益保护机制，本章将对其进行详细介绍。

13.1 工程运行安全机制

工程运行安全机制可保证工程运行时的安全，如防止误操作、操作追责等。

13.1.1 安全属性

输入框、组合框、标准按钮、动画按钮构件具有安全属性。其中，输入框的安全属性只具有使能控制功能，而组合框、标准按钮和动画按钮构件的安全属性除了使能功能以外还具有安全控制功能。

13.1.1.1 使能控制

以输入框构件为例讲解使能控制功能。

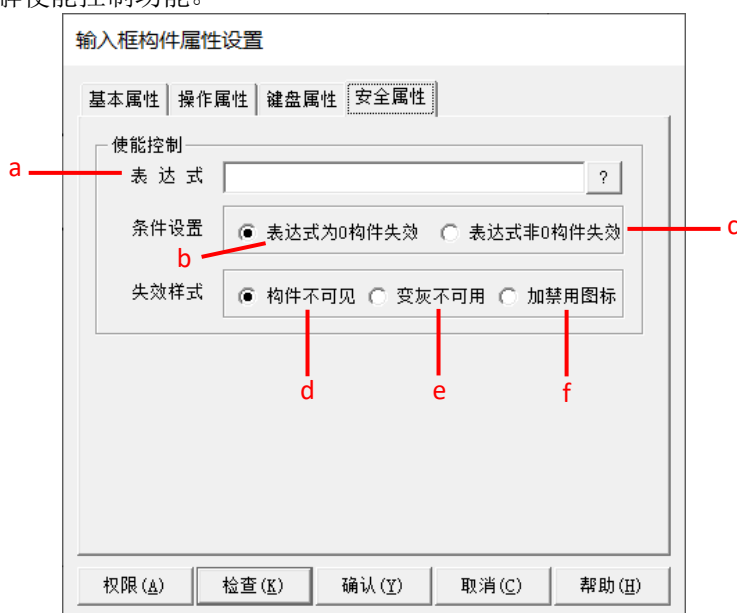


图 13.1-1 输入框构件的安全属性

- a. 表达式：用表达式控制构件是否可操作（即是否使能）。若不设置任何表达式，则运行时，构件始终处于可操作状态。
- b. 表达式为 0 构件失效（即不可操作）：当表达式的值为 0 时，按钮失效，否则，按钮有效。
- c. 表达式非 0 构件失效：当表达式的值为非 0 时，按钮失效，否则，按钮有效。
- d. 构件不可见：当构件失效时，隐藏构件。
- e. 变灰不可用：当构件失效时，构件文本为灰色。
- f. 加禁用图标：当构件失效时，构件上会出现一个禁用图标。

● 组态 3 个输入框构件展示不同的【失效样式】。

- ① 将输入框构件的安全属性表达式关联变量【输入框安全属性】，选择【条件设置】为【表达式为 0 时构件失效】。复制 3 份后，分别设置其【失效样式】为【构件不可见】【变灰不可用】【加禁用图标】。然后新建一个【标准按钮】构件，双击 → 操作属性 → 数据对象值操作 → 取反 → 输入框按钮属性。
- ② 保存工程后下载工程，然后启动运行。由于变量【输入框安全属性】默认初始值为 0，所以刚开始运行时，输入框处于失效状态。点击按钮，变量【输入框安全属性】取反，变成 1，则构件使能，变成了可操作状态，如图 13.1-2 所示。

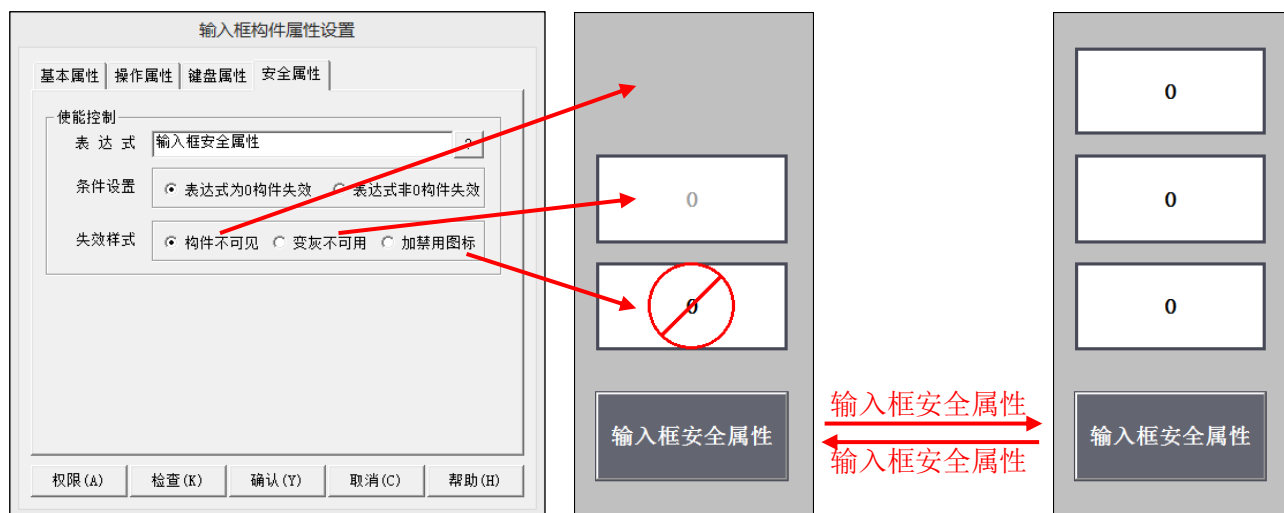


图 13.1-2 输入框安全属性页和使用效果对比图

13.1.1.2 安全控制

安全控制功能可以防止误操作。以标准按钮构件为例讲解安全控制功能。它的使能控制功能与输入框构件一致，不再赘述。

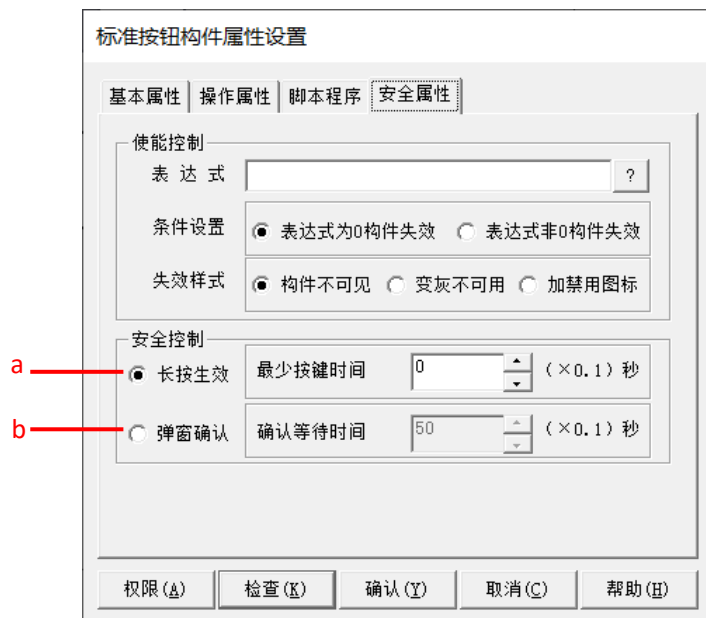


图 13.1-3 标准按钮构件的安全属性

- 长按生效：按下按钮时长超过最少按键时间后按钮动作才生效；最少按键时间取值为 0~1000，单位 0.1 秒。
- 弹窗确认：按下按钮弹出确认框，点击确认后按钮动作生效，点击取消后按钮动作无效；超过确认等待时间没有点击确认则确认框消失并且按钮动作无效；确认等待时间取值为 10~1000，单位 0.1 秒；当操作属性为数据对象值操作的“按 1 松 0”和“按 0 松 1”操作时，安全控制不可选择弹窗确认。

13.1.2 用户权限设置

添加用户权限既可以避免没有操作能力的人对画面进行误操作，又可以在出现了异常状况后，通过操作日志追查当时的操作人员和行为动作。

用户权限设置是采用用户组和用户的概念来进行操作权限的控制，类似于 Windows 开机时的用户登录。在 McgsPro 组态软件中可以定义多个用户组，每个用户组中可以包含多个用户，同一个用户可以隶属于多个用户组。操作权限是以用户组为单位来进行分配的，即某些用户组有权限操作某种功能，而某个用户能否对这个功能进行操作取决于该用户所在的用户组是否具备对应的操作权限。在 McgsPro 组态软件中，通过“用

户权限管理”编辑用户和用户组。

13.1.2.1 添加用户

点击菜单栏 → 工具 → 用户权限管理，打开【用户管理器】对话框，如图 13.1-4 所示。

在用户管理器窗口中，上半部分为已建用户的用户名列表，下半部分为已建用户组的列表。当用鼠标激活用户名列表时，在窗口底部显示有“新增用户”、“复制用户”、“删除用户”等对用户操作的按钮；当用鼠标激活用户组名列表时，在窗口底部显示有“新增用户组”、“删除用户组”等对用户组操作的按钮。



图 13.1-4 用户管理器

单击新增用户按钮，可以添加新的用户名，选中一个用户时，点击“属性编辑”或双击该用户，会出现用户属性设置窗口，在该窗口中，用户可对用户名称、用户描述、用户密码及该用户隶属于哪个用户组进行设置，如图 13.1-5 所示。



图 13.1-5 用户属性设置

13.1.2.2 添加用户组及成员管理

用鼠标激活用户组列表，单击新增用户组按钮，可以添加新的用户组，选中一个用户组时，点击“属性编辑”或双击该用户组，会出现用户组属性设置窗口，在该窗口中，用户可对用户组名称、用户组描述及该用户组包括哪些用户或用户组进行设置，如图 13.1-6 所示。

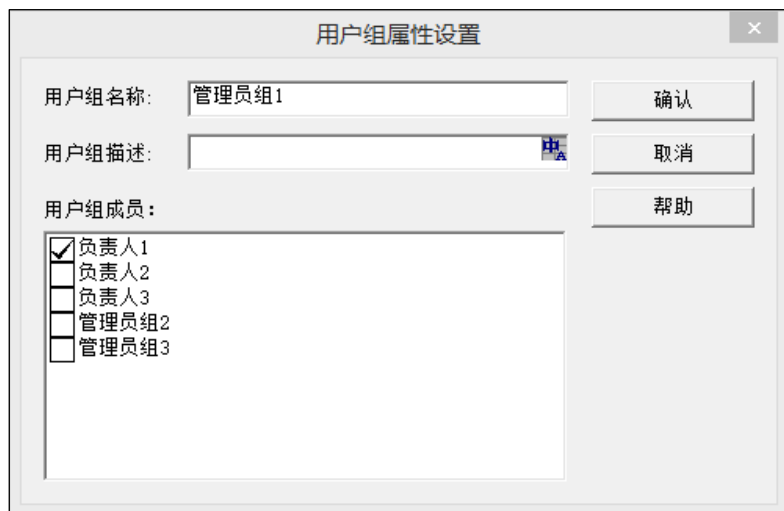


图 13.1-6 用户组属性设置

13.1.2.3 USB 密钥

- ① USB 密钥的功能为预先设定好用户的登录信息，生成密钥文件。设置步骤为：打开用户管理器，选择一个用户名，点击 USB 密钥，如图 13.1-7 所示。



图 13.1-7 USB 密钥制作

- ② 电脑上插上 U 盘，选择导出为 U 盘路径即可在 U 盘上生成密钥，密钥文件为“McgsUserKey”。设置界面如图 13.1-8 所示。

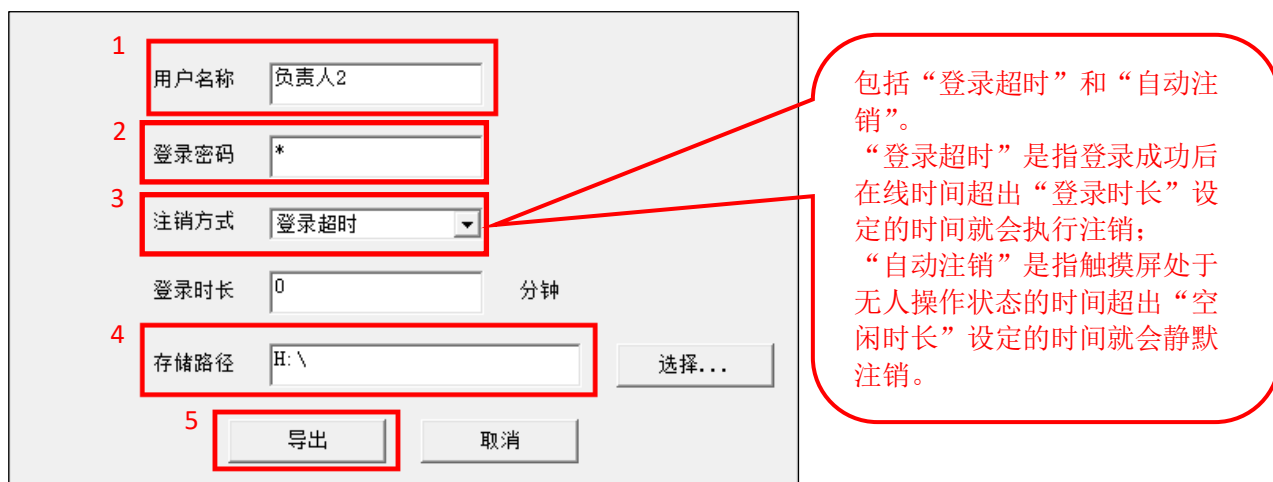


图 13.1-8 USB 密钥生成

- ③ 在工程运行中，当导出过 USB 密钥的用户需要登录时，可以在屏的 USB 口插入 U 盘，然后点击登录界面的【USB 登录】按钮即可登录，无需再输入密码。

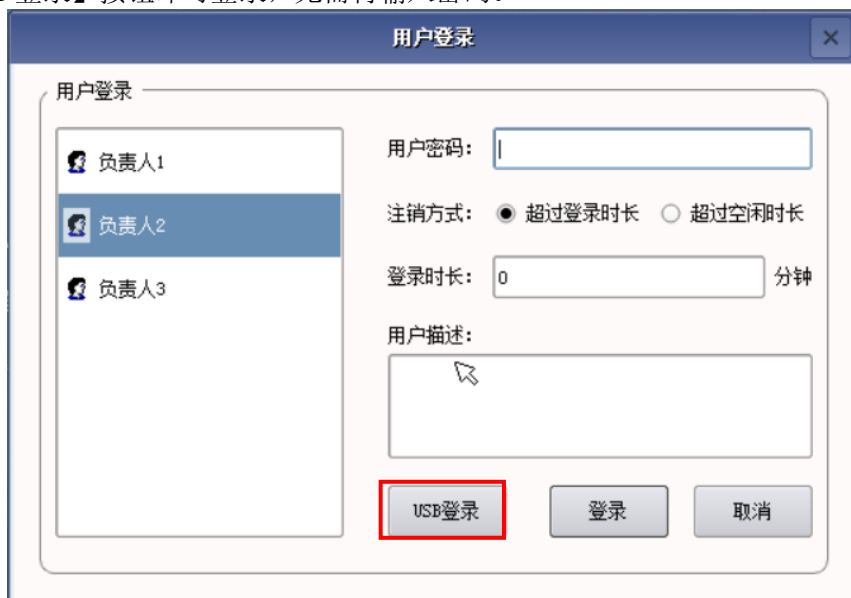


图 13.1-9 用户登录界面

13.1.2.4 用户权限函数

- 在 McgsPro 组态软件中，可以通过脚本函数来实现用户的登录、退出、登录用户名获取、登录用户组名获取、打开用户管理对话框、登录用户密码修改等操作。

!LogOn(): 弹出登录对话框

!LogOff(): 注销当前用户

!GetCurrentUser(): 读取当前登录用户的用户名

!GetCurrentGroup(): 读取当前登录用户的所在用户组名

!Editusers(): 弹出用户管理窗口

!ChangePassword(): 弹出改变密码窗口，供当前登录的用户修改密码。

- 以使用密码打开窗口为例进行用户权限函数的讲解

例一：启动画面的密码保护

本节主要讲解使用用户权限功能给启动画面设置密码，只有当用户输入正确的密码后，才能进入工程。

- ① 进入工作台的【主控窗口】界面，双击【主控窗口】图标或点击右侧的【系统属性】按钮进入【主控窗口】设置界面 → 基本属性，选择运行权限为【进入登录，退出不登录】，点击【权限设置】按钮，勾选希望拥有该权限的用户组，点击确认。

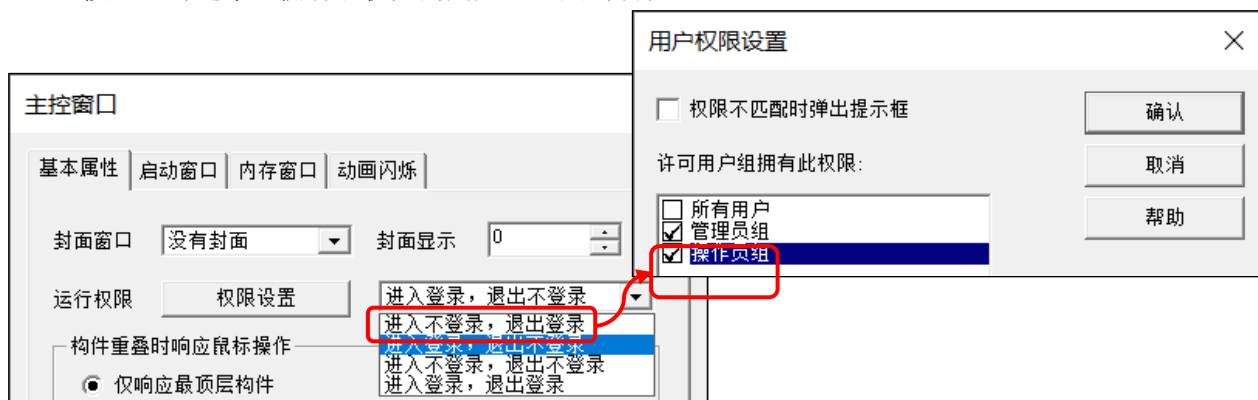


图 13.1-10 用户权限设置步骤

- ② 保存工程后下载工程，然后启动运行，在工程刚开始运行时，系统就会弹出用户登录框，只有选择了登录权限的用户组中的用户成功登录后，才能进入工程。

例二：子画面的密码保护

组态一个按钮，当用户点击按钮后，弹出用户登录界面。此时，若登录成功，则打开子窗口，若登录不成功，不执行任何操作。

- ① 组态好两个画面，一个命名为【主画面】，一个命名为【子画面】。
- ② 在主画面窗口中新建一个按钮，将它的文本设置为【密码正确后打开子窗口】，将它的抬起脚本设置为【if !logon()=0 then !OpenSubWnd(子窗口,0,0,300,150,1)】，如图 13.1-11 所示。脚本的作用是：当按钮抬起时，弹出用户登录窗口，若登录成功，打开【子窗口】，若登录不成功，不执行任何操作。

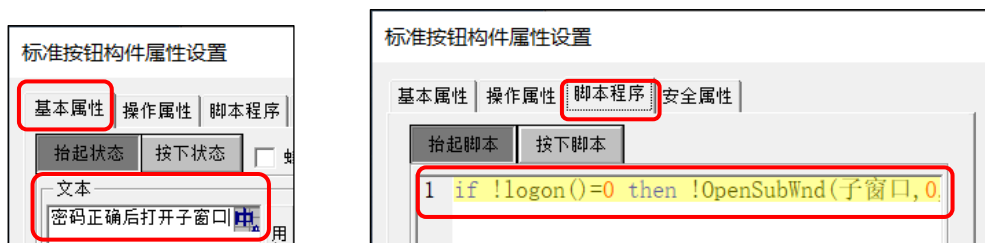


图 13.1-11 密码正确后打开子窗口按钮设置

- ③ 然后在【子窗口】页面中新建一个按钮，将它的文本设置为【关闭子窗口】，抬起脚本设置为【!CloseSubWnd(子窗口)】，如图 13.1-12 所示。点击按钮后，将关闭【子窗口】。

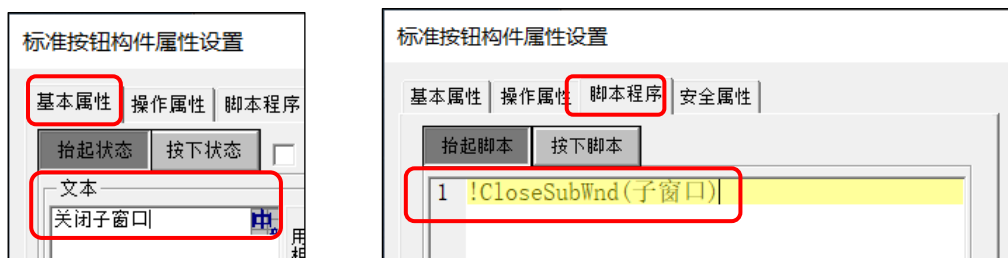


图 13.1-12 关闭子窗口按钮设置

- ④ 保存工程后下载工程，然后启动运行。点击【密码正确后打开子窗口】按钮，系统弹出用户登录窗口，选择用户后输入登录密码即可打开子窗口，若密码不正确，提示密码输入错误，子窗口不打开。



图 13.1-13 运行演示

13.1.2.5 构件操作权限

另外，部分构件也可以设置用户操作权限，只有设置了拥有用户操作权限的用户组才能操作该构件，否则操作无反应或提示权限不匹配。用户操作权限在工程运行时才能体现出来。能进行用户操作权限设置的构件有标准按钮、输入框、动画按钮、组合框、滑动输入器。

以【标准按钮】构件为例，进行用于操作权限设置的步骤如图 13.1-14 所示。

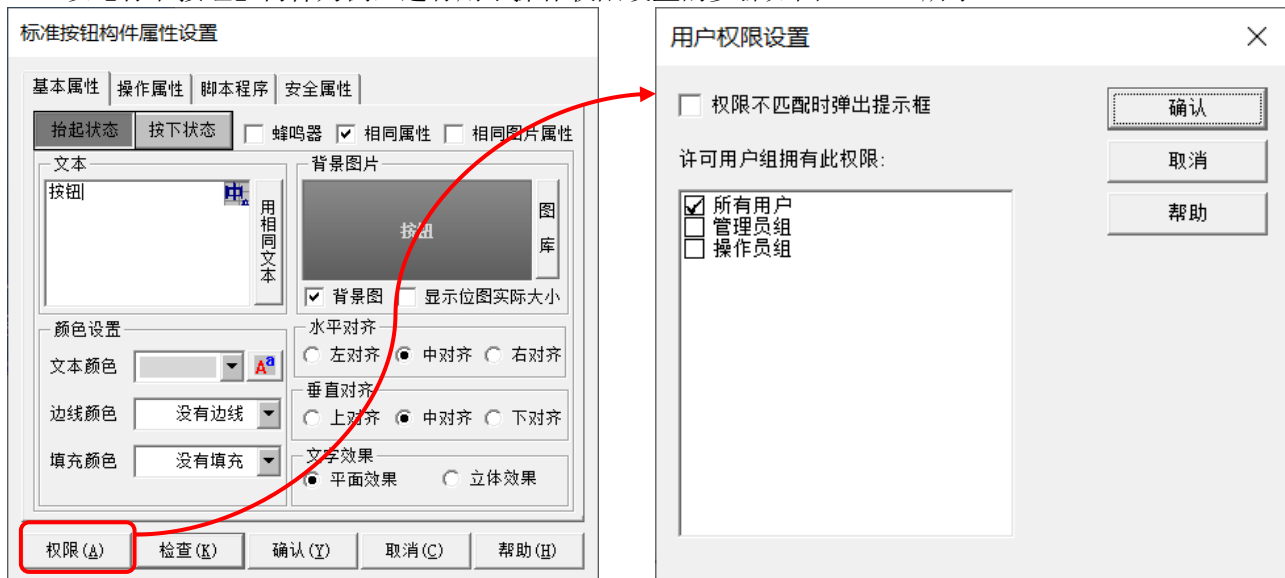


图 13.1-14 按钮的用户操作权限设置

13.1.3 操作日志记录

在用户的使用过程中，通过对动画构件进行日志组态和编辑，可实时记录用户的某种动作（如鼠标点击、内容编辑、鼠标事件等）。当出现异常状况时，通过翻阅和分析日志记录，可以得知操作过程是否符合流程规范，便于问题查找和定位。

操作日志标准内容如图 13.1-16 所示。

日期	用户	窗口	构件	动作	描述
2019/5/5 9:00	负责人	主页	控件0	点击	打开用户窗口并执行脚本
2019/5/5 9:05	操作员1	配方管理	控件1	编辑	修改配方内容

图 13.1-15 操作日志标准内容示例

13.1.3.1 组态显示操作日志的工程

本节将组态一个启用了操作日志的工程，并由存盘浏览构件显示操作日志。

- ① 在实时数据库新建两个整数型对象【电机启动】和【电机停止】。
- ② 组态两个按钮，一个按钮【控件 0】设置基本属性 → 文本【电机启动】，操作属性 → 数据对象值操作 → 【置 1】并关联对象【电机启动】，如图 13.1-7 所示。另一个按钮【控件 1】设置基本属性 → 文本【电机停止】，操作属性 → 数据对象值操作 → 【置 1】并关联对象【电机停止】。

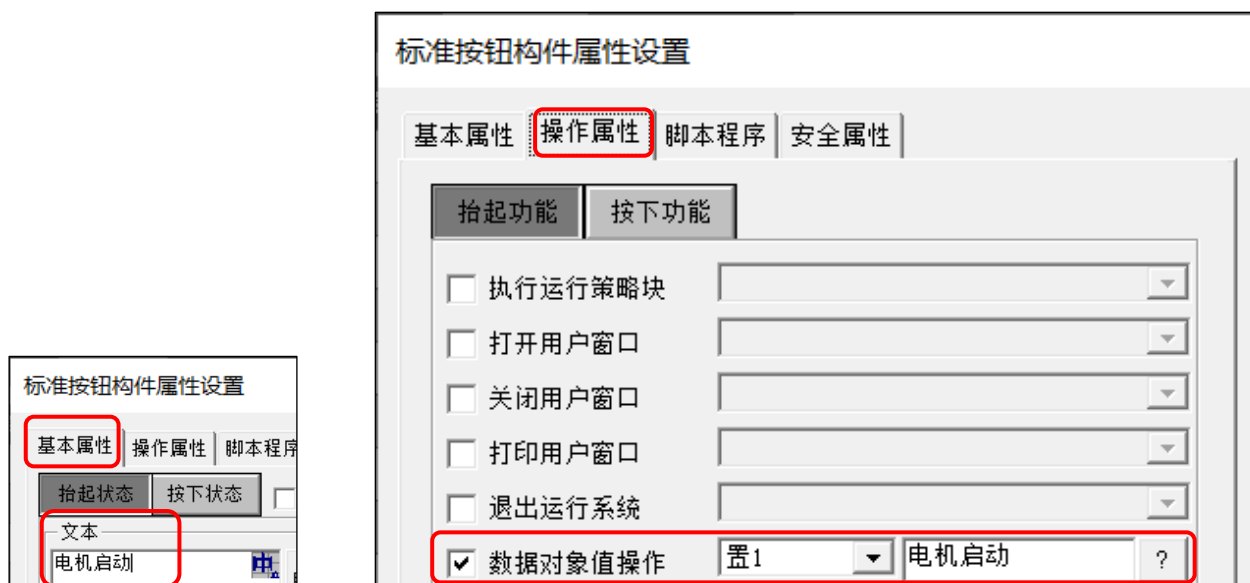


图 13.1-16 电机启动按钮设置

- ③ 点击菜单栏中的【工具】 → 【操作日志设置】，弹出【操作日志】设置界面。

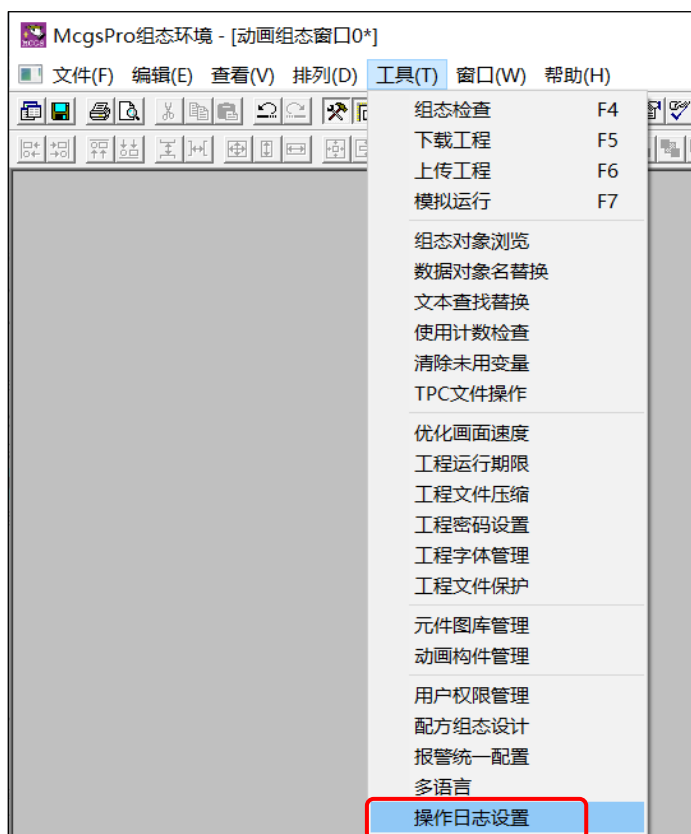


图 13.1-17 打开操作日志

打开后界面如图 13.1-18 所示。【操作日志】设置界面是操作日志的编辑环境。用户可以在这里开启或关闭操作日志记录功能。在启用该功能时，可以开启指定构件的日志，和对日志描述文本进行编辑，日志描述文本支持多语言。

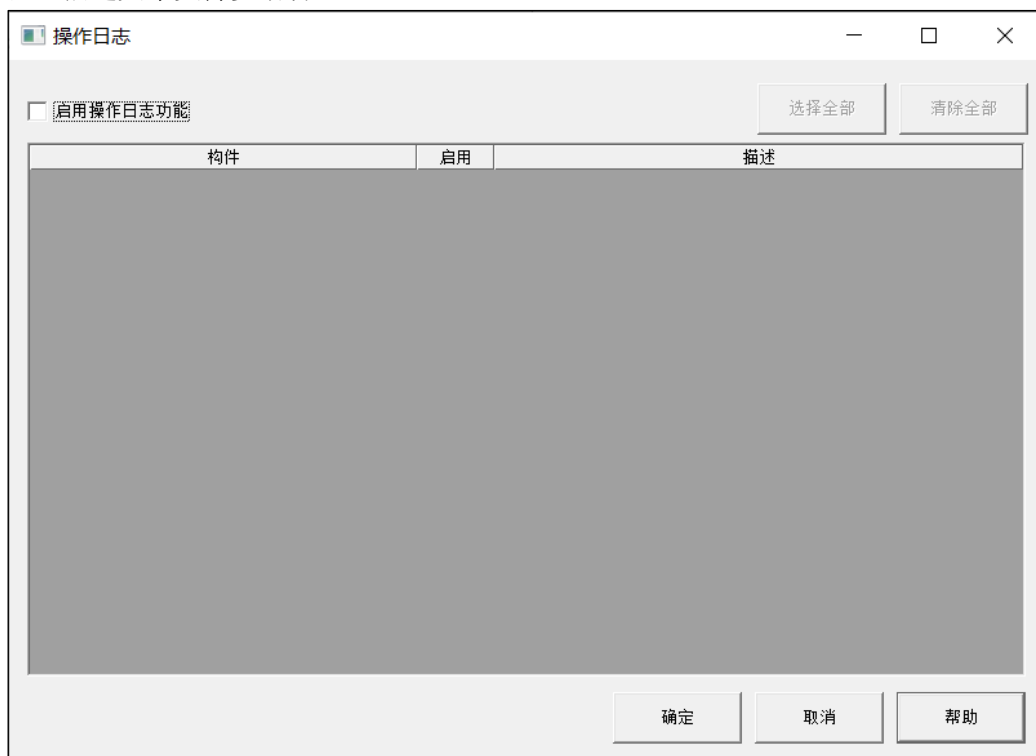


图 13.1-18 操作日志设置界面

- ④ 勾选【启动操作日志功能】，点击【选择全部】则下方的所有构件前的复选框都被选中。设置【控件

0】的描述为【电机启动】，【控件 1】的描述为【电机停止】。点击【确认】保存设置并关闭窗口。

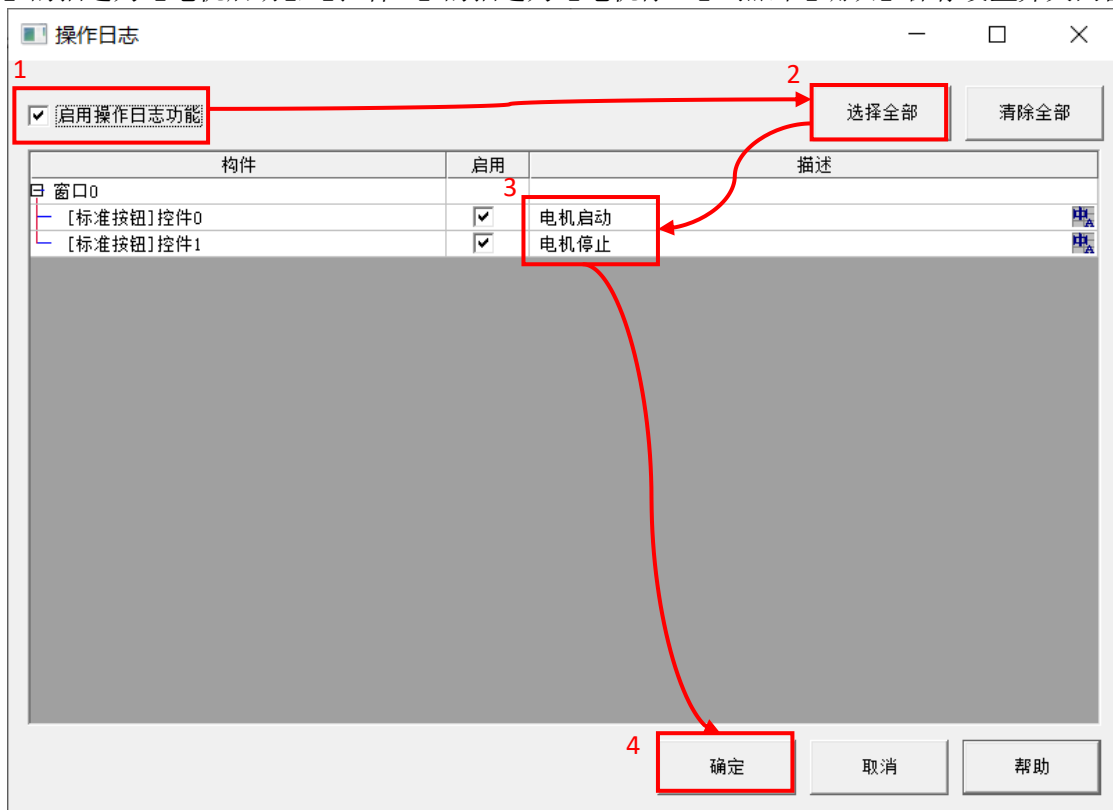


图 13.1-19 操作日志设置流程

- ⑤ 从工具箱拖拽一个【存盘浏览构件】到画面中合适的位置和大小，双击打开【存盘数据浏览构件属性设置】界面。设置其【数据来源】为【操作日志】。

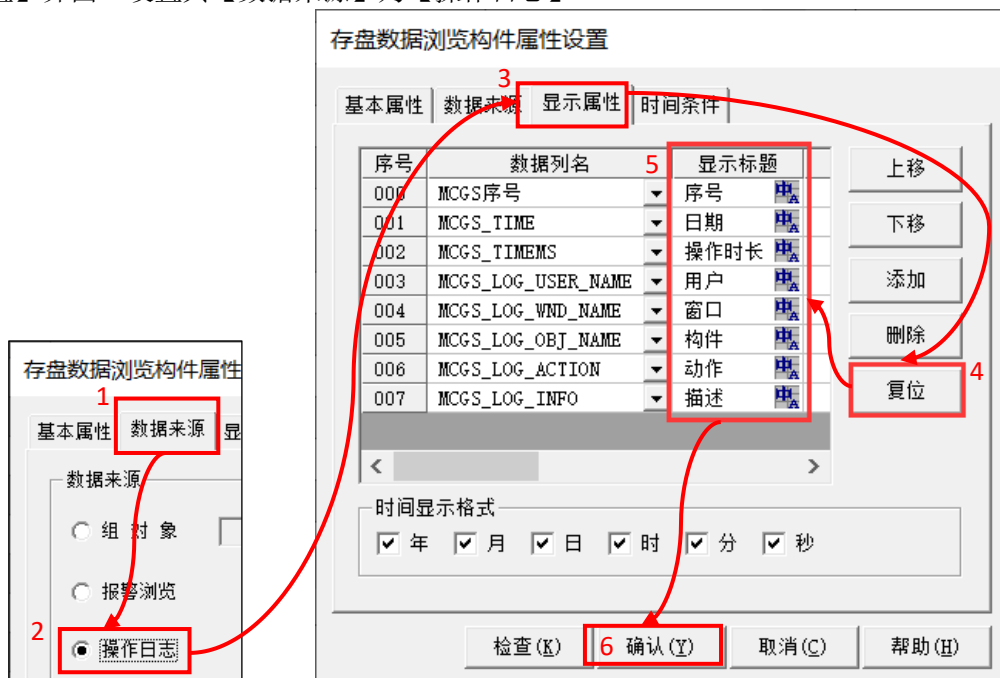


图 13.1-20 存盘浏览构件设置流程

- ⑥ 进入【显示属性】功能页，点击【复位】按钮，然后修改【显示标题】列如图 13.1-20 所示。确认关闭界面。
- ⑦ 用鼠标选中【存盘数据浏览】构件，可以从窗口下方看到它的名称是【控件 4】。

- ⑧ 在窗口空白处双击鼠标，弹出【用户窗口属性设置】界面，设置循环脚本为：**窗口0.控件4.Refresh()**，循环时间为 1000ms。它的作用是每隔一秒将【存盘数据浏览】构件刷新一次。



图 13.1-21 设置循环脚本

- ⑨ 按照 13.1.2 的方法新建两个用户【管理员 1】和【操作员 1】，分别隶属于【管理员组】和【操作员组】。
- ⑩ 然后进入工作台的【主控窗口】属性设置界面，设置运行权限为【进入登录，退出登录】。接着点击【权限设置】按钮，进入【用户权限设置】窗口，勾选【所有用户】前的复选框。

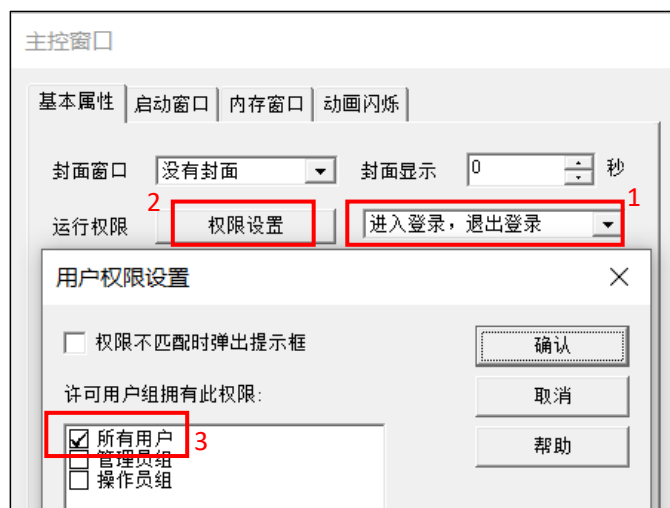


图 13.1-22 设置用户权限

- ⑪ 保存工程后下载工程，然后启动运行。工程启动后弹出登录界面，选择【操作员 1】，输入对应的密码，点击【登录】按钮，进入工程画面。



图 13.1-23 用户登录界面

- ⑫ 点击【电机启动】按钮，【存盘数据浏览】构件中依次显示出序号为 1-3 的三行数据。数据由左往右依次是动作【鼠标按下操作】【鼠标抬起操作】【鼠标点击操作】的【序号】【日期】【操作时长】，操作的【用户】，动作所在的【窗口】，动作的【构件】名以及动作的【描述】内容。按下【电机停止】按钮也同样如此。

序号	日期	操作时长	用户	窗口	构件	动作	描述
1	2020-08-04 16:03:05	747.00	管理员1	窗口0	控件0	鼠标按下操作	电机启动
2	2020-08-04 16:03:05	851.00	管理员1	窗口0	控件0	鼠标抬起操作	电机启动
3	2020-08-04 16:03:05	851.00	管理员1	窗口0	控件0	鼠标点击操作	电机启动
4	2020-08-04 16:10:30	866.00	管理员1	窗口0	控件1	鼠标按下操作	电机停止
5	2020-08-04 16:10:30	953.00	管理员1	窗口0	控件1	鼠标抬起操作	电机停止
6	2020-08-04 16:10:30	953.00	管理员1	窗口0	控件1	鼠标点击操作	电机停止

图 13.1-24 操作日志运行效果

13.1.3.2 对日志进行操作

本节主要介绍使用函数对操作日志进行开启、关闭、清除、导出操作。

- ① 回到 McgsPro 组态软件中，在 13.1.3.1 工程的【存盘数据浏览】构件下方新建 4 个按钮，将它们的显示文本分别设置为【开启日志功能】【关闭日志功能】【清除日志】【日志导出】。

其中【开启日志功能】按钮的脚本程序为：

`!OperationLogEnable()` ‘开启日志功能’

【关闭日志功能】按钮的脚本程序为：

`!OperationLogDisable()` ‘关闭日志功能’

【清除日志】按钮的脚本程序为：

`!OperationLogClear()` ‘清除日志’

【日志导出】按钮的脚本程序为：

!ExportOperationLogToCSV("log.csv", !TimeStr2I("2020-08-01 00:00:00"), !TimeStr2I("2020-08-31 23:59:59"), 2)

将操作日志在 2020 年 8 月 1 日 00:00:00 到 2020 年 8 月 31 日 23:59:59 时间段内的数据导出至 U 盘根目录，存储文件名为【log.csv】，如果文件已存在，则导出数据追加至文件末尾。

- ② 组态完成后保存工程，下载工程，下载成功后启动运行。
- ③ 点击【电机启动】按钮，下方的【存盘数据浏览】构件中出现操作日志数据。此时点击【关闭日志功能】按钮，再次点击【电机启动】按钮，则【存盘数据浏览】构件中没有新的数据出现。只有当点击【开启日志功能】按钮后，再次对【电机启动】按钮，新的操作日志才会出现。



图 13.1-25 关闭日志功能效果



图 13.1-26 开启日志功能和导出日志效果

- ④ 给 TPC 插上 U 盘，点击【日志导出】按钮，则按钮脚本中划定的时间范围内的操作日志被导出到 U 盘，导出文件名为【log.csv】。
- ⑤ 点击【清除日志】按钮，操作日志被清除，【存盘数据浏览】构件中显示的日志内容也随之清空。

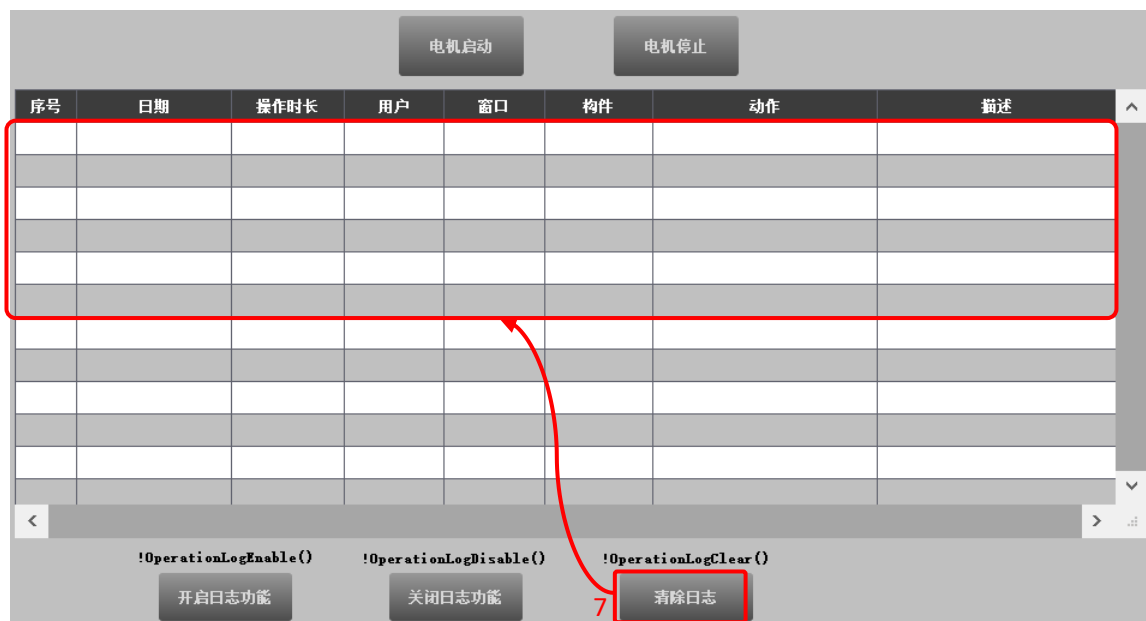


图 13.1-27 清除日志效果

13.2 工程师权益保护

工程师权益保护机制可以保护工程师的开发成果。

13.2.1 工程密码设置

给工程设置了密码后，需要输入密码才能打开工程进行组态。设置步骤如下：

- ① 点击菜单栏 → 工具 → 工程密码设置。

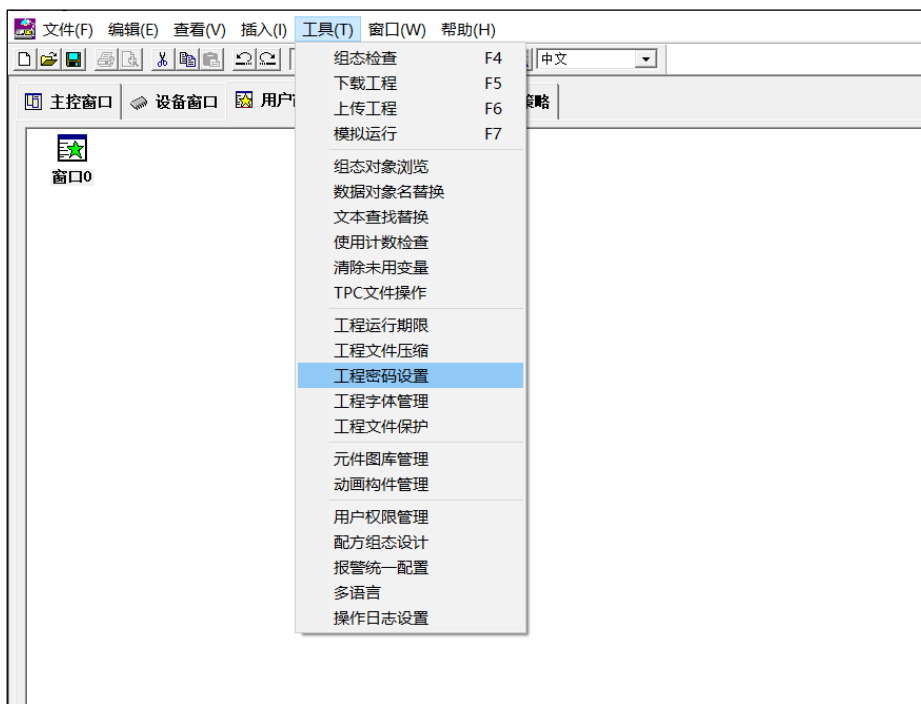


图 13.2-1 进入工程密码设置

- ② 弹出【修改工程密码】窗口，如图 13.2-2 所示。

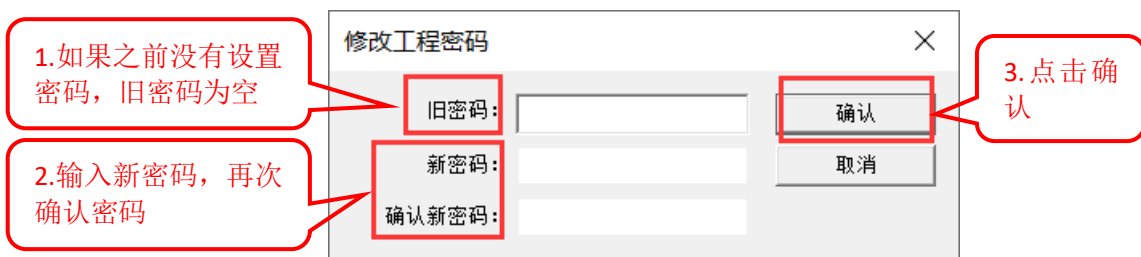


图 13.2-2 修改工程密码窗口

③ 再次打开工程，就会提示输入工程密码。

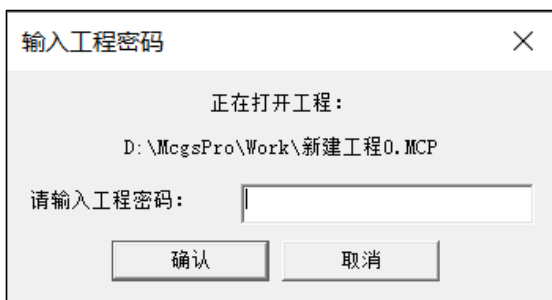


图 13.2-3 输入工程密码窗口

13.2.2 TPC 系统密码设置

TPC 系统密码的作用是：设置了该密码后，以下操作均需要验证 TPC 系统密码：

- ① 再次打开 TPC 系统设置界面；
 - ② 工程上传和下载；
 - ③ 使用 U 盘部署或卸载某些特殊功能，以及升级 NK 版本或运行环境。
- 进入 TPC 系统设置界面的方法，将在后续章节：[硬件篇 2.1.1](#) 和 [2.1.2](#) 进行介绍。



图 13.2-4 TPC 系统密码设置界面

13.2.3 工程文件保护

使用了工程文件保护功能后，工程只能在具有相同识别码的触摸屏上使用。触摸屏的识别码由设备商 U 盘工具设置，设备商 U 盘工具需要联系触摸屏供应商获取。设置步骤如下：

- ① 点击菜单栏 → 工具 → 工程文件保护。

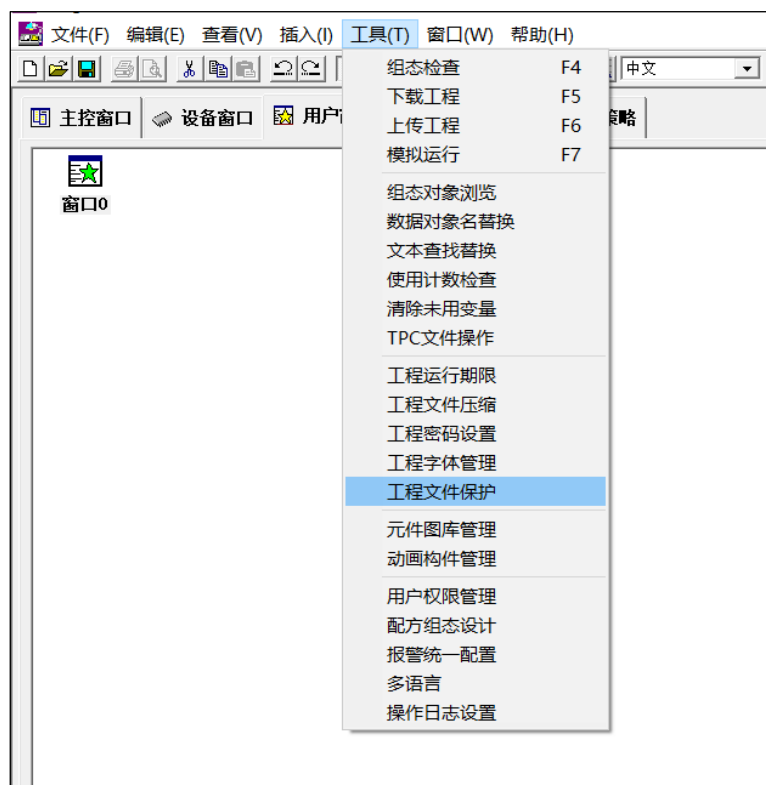


图 13.2-5 进入工程文件保护

② 弹出【工程文件保护】设置界面。

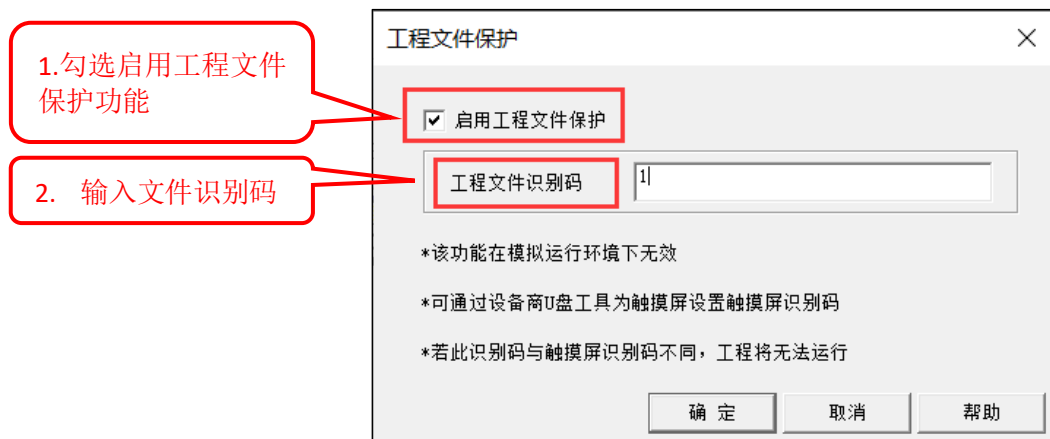


图 13.2-6 工程文件保护设置界面

③ 设置完成后，工程只能在具有相同识别码的触摸屏上使用。

13.2.4 工程上传

在下载配置界面的下载选项中，有一个【支持工程上传】的选项。若在下载工程到屏幕的时候，勾选了该选项，则用户可以将工程从屏幕上传的电脑，若没有勾选此选项，工程则不能上传到电脑。

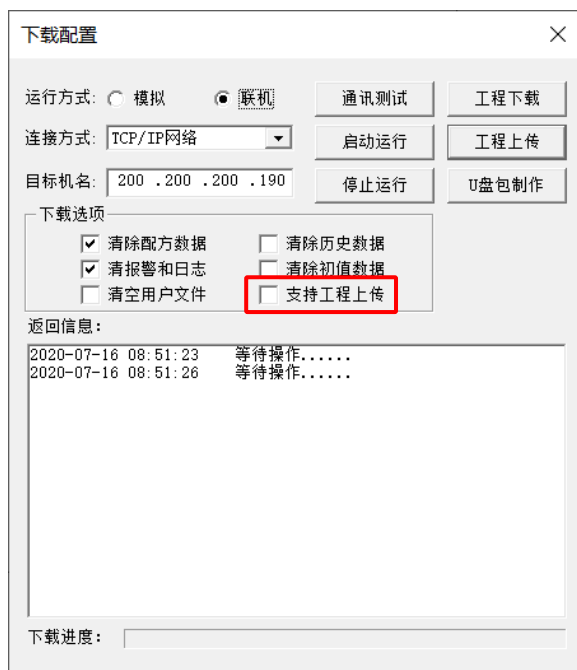


图 13.2-7 工程是否上传选择

13.2.5 工程运行期限

工程运行期限功能主要用于实现分期付款。在组态环境配置了运行期限功能的工程，下载到触摸屏之后，运行到当期的截止日期之前，会根据组态设置的提前天数弹出警告窗口，提醒用户输入当期的激活密码，以激活下一分期。警告窗口为模态窗口，弹出时触摸屏无法对其之外的任何窗口和控件进行响应。该窗口显示当前分期的情况，便于用户提前了解工程运行期限，对生产活动作出及时安排。

● 进入【工程运行期限】设置界面

① 点击菜单栏 → 工具 → 工程运行期限。

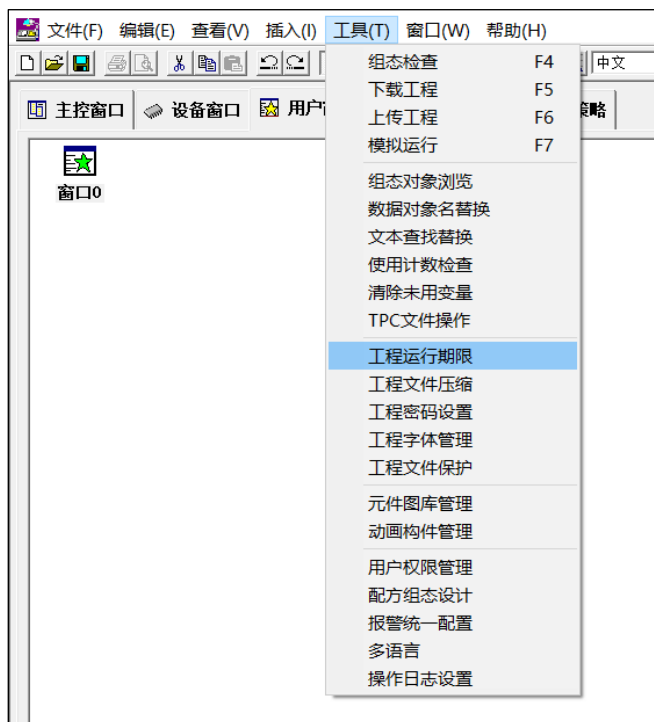


图 13.2-8 进入工程运行期限界面

② 弹出【运行期限】设置界面，如图 13.2-9 所示。

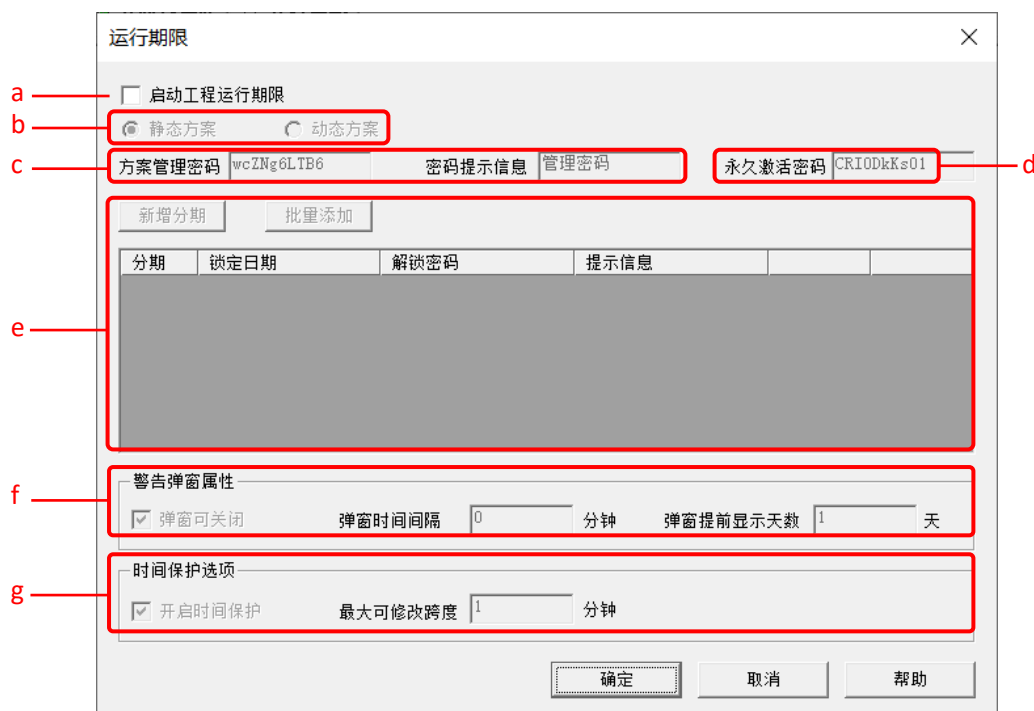


图 13.2-9 运行期限设置界面

运行期限设置界面配置信息：

- 设置是否启动工程运行期限。
- 运行期限分为静态方案和动态方案，并且两种方案互斥。
- 方案管理密码：使用“设备商 U 盘工具包”去管理期限时需输入的密码，可以设置密码提示信息，输入界面可见。
- 永久激活密码：在永久激活密码输入框内输入永久密码，即可一次性解锁所有分期。
- 分期密码列表。
- 警告弹窗属性：可设置弹窗的属性，包括弹窗是否可关闭、间隔时间和提前显示天数。
- 时间保护选项：勾选后，运行环境对系统时间的修改作一定的保护，避免用户通过修改系统时间来规避期限。

● 组态静态方案

在明确了分期支付期限的情况下，可选择静态方案。

- ① 勾选【启动工程运行期限】前的复选框激活工程运行期限功能，然后选择【静态方案】。

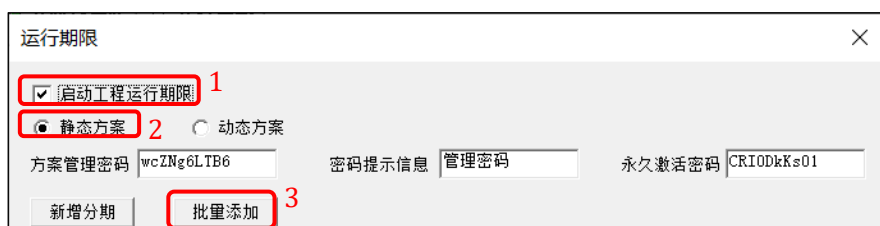


图 13.2-10 组态静态方案

静态方案中新增分期的方式有两种：【新增分期】和【批量添加】。若点击【新增分期】，会按照当日日期增加一组分期，再点击时逐月增加，最多可设置 48 期。系统默认生成的锁定日期和解锁密码，用户均可修改（密码最大长度为 20 个字符）。本节样例使用【批量添加】的方式进行设置。

- ② 点击【批量添加】，按照需要的分期期数、第一期截止日期和分期间隔生成一批期限。

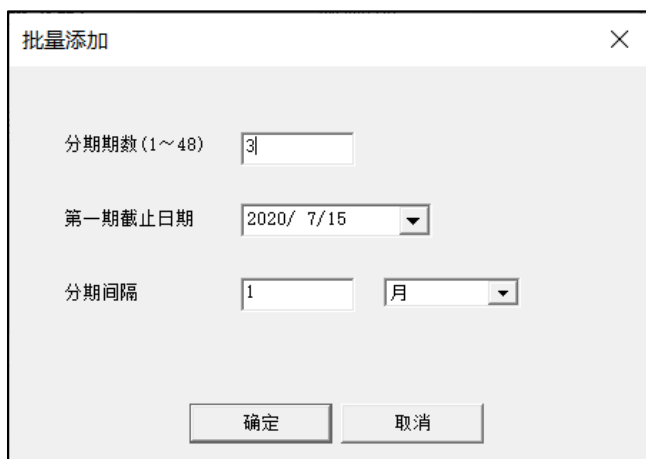


图 13.2-11 批量添加静态分期期限

生成后效果如图 13.2-12 所示。

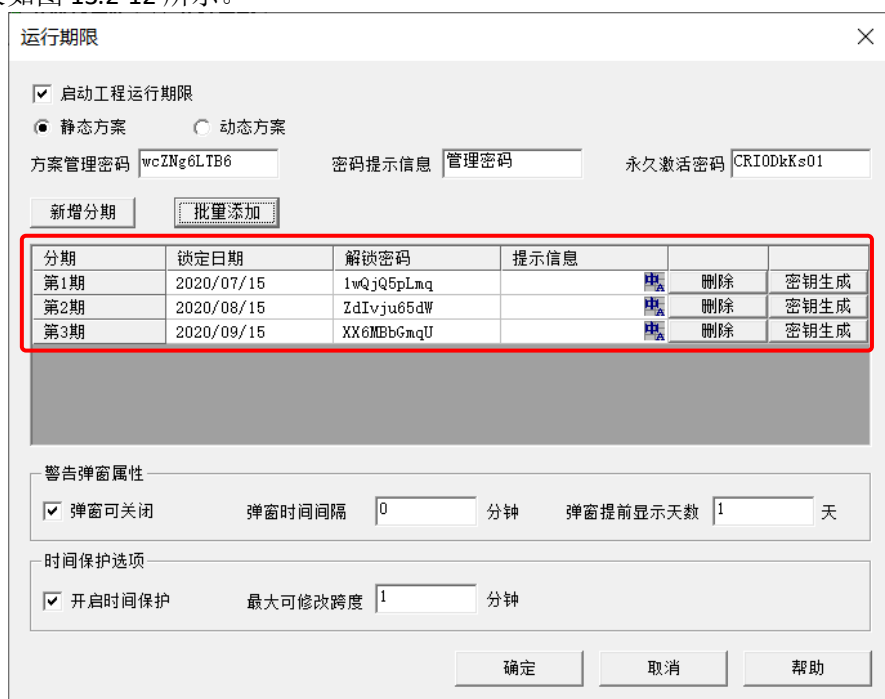


图 13.2-12 静态分期设置效果图

- ③ 为了方便使用，McgsPro 引入了【USB 激活】功能，用户可点击【密钥生成】按钮弹出【运行期限密钥生成】提示框，导出密钥文件到 U 盘上。



图 13.2-13 USB 密钥生成步骤

④ 按工程需要设置【警告弹窗属性】。



图 13.2-14 警告弹窗属性设置

【弹窗可关闭】选项控制运行到期弹窗能否关闭；【弹窗时间间隔】控制运行到期弹窗弹出的时间，若设置 1 分钟，在关闭弹窗 1 分钟后会再次弹出此弹窗（排除已激活的情况）；【弹窗提前显示天数】设置 1 天，表示弹窗会在到期前 1 天弹出，提醒用户工程到期了。

⑤ 然后设置【时间保护选项】。



图 13.2-15 时间保护选项设置

用户使用时间保护选项一般是防止终端使用者通过修改系统时间来规避运行期限。比如，勾选【时间保护选项】，设置最大可修改跨度为 1 分钟（设置不得超过 1440 分钟），下载到 TPC 后，用户修改系统时间最多能往前修改 1 分钟，往后无限制。修改的时间跨度超过 1 分钟时，会弹出如图 13.2-16 所示的警告，点击继续，则时间修改成功。此时进入运行环境还是会弹出系统时间异常的警告，无法正常运行工程。用户不勾选“时间保护选项”则无此限制。

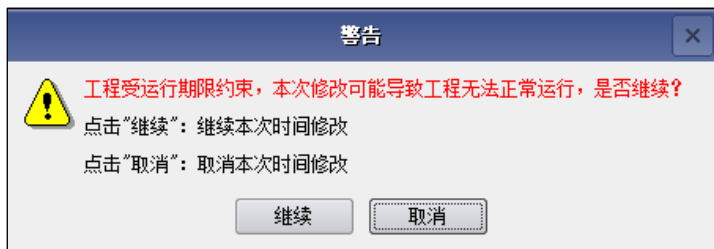


图 13.2-16 修改 TPC 时间警告

⑥ 组态完成后点击确定，保存设置并关闭设置界面。

⑦ 模拟期限到期运行：工程启动后弹出【运行到期】提示框，如图 13.2-17 所示。用户在密码输入框内，输入组态时设定的对应分期的解锁密码，确认后即可激活该分期，工程可正常使用至下一分期的锁定日期。或插入含当期 USB 密钥的 U 盘，待识别后点击【USB 激活】按钮，亦可激活分期。

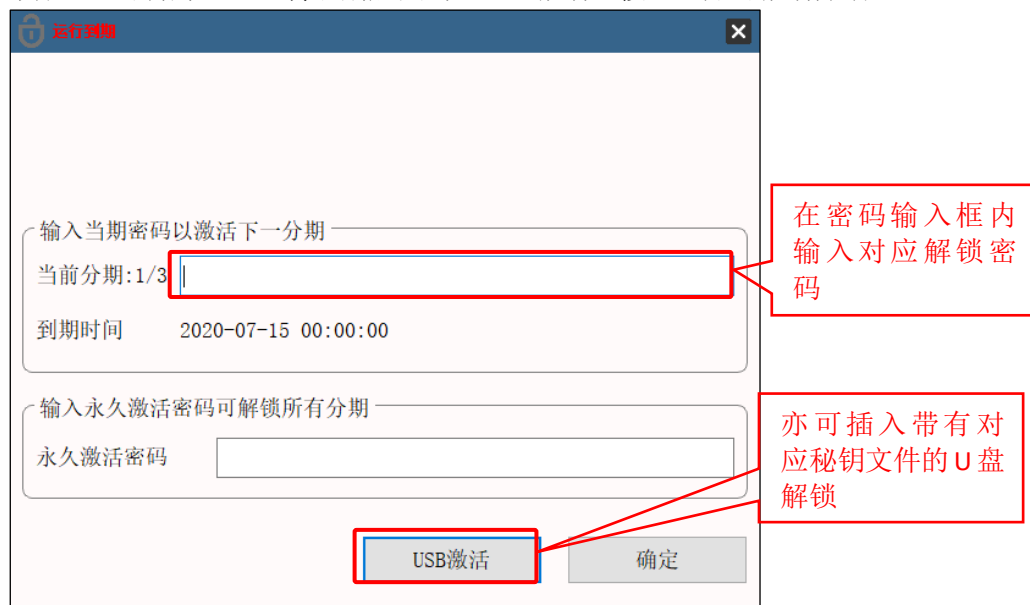


图 13.2-17 静态方案运行到期提示框

● 组态动态方案

在分期支付期限和批次不明确的情况下，可选择动态方案。

注意，切换方案会有“切换方案会清空列表内容，是否继续？”的提示，请谨慎选择。

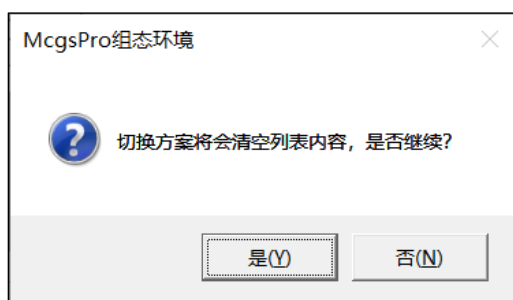


图 13.2-18 切换方案提示框

设置分期方案为【动态方案】后，动态方案分期期数固定为 1 期，分期结束时间默认为当日日期，用户可根据需要自行修改。而分期密码是软件自动生成的长度为 40 个字符的随机密码，用户无法修改。与静态方案一样，用户可导出 USB 密钥，通过 U 盘来解锁分期。而【警告弹窗属性】和【时间保护选项】的定义和设置方法与静态方案一致，此处不再赘述。

第一次设置完成后将工程下载到触摸屏中使用。注意：下载工程时不勾选下载配置界面的【支持工程上传】选项，并且不将工程文件交给用户。

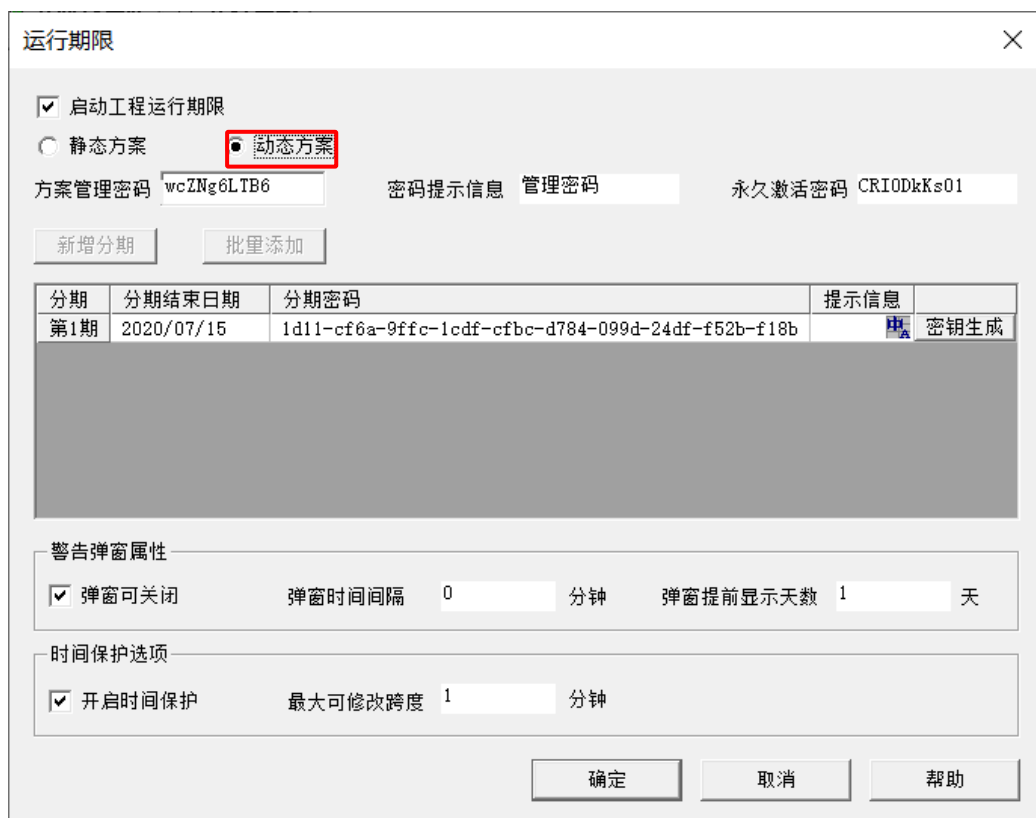


图 13.2-19 动态方案设置界面

当设定的期限到期时，在组态状态下将【分期结束日期】设置成希望延长到的日期，然后将将此日期和软件新生成的 40 位密码提供给工程用户，此时不许要重新下载工程。工程用户在弹窗的日期输入框输入组态设定的截止日期，在密码输入框输入 40 位动态密码，验证通过即可继续正常使用至新设置的截止日期，如图 13.2-20 所示。

图 13.2-20 动态方案运行到期提示框

注：动态方案的分期密码是系统生成的随机密码，与分期结束时间和方案管理密码有关，分期结束时间或方案管理密码改变，分期密码随之改变，此时需要重新下载工程。

第 14 章 打印功能应用实例

本章主要介绍 TPC 与炜煌微型打印机的连接与使用。

14.1 建立设备通道

首先在 McgsPro 中建立设备通道。

- ① 新建工程 → 进入工作台 → 设备窗口 → 双击  图标或点击右侧的【设备组态】按钮 → 进入设备窗口
- ② 点击工具栏的  图标打开工具箱，若工具箱中没有【微型打印机】驱动，则点击工具箱中的【设备管理】按钮 → 目录树-所有设备-通用设备-微型打印机 → 双击将其添加到右侧【选定设备】窗口中 → 确定。

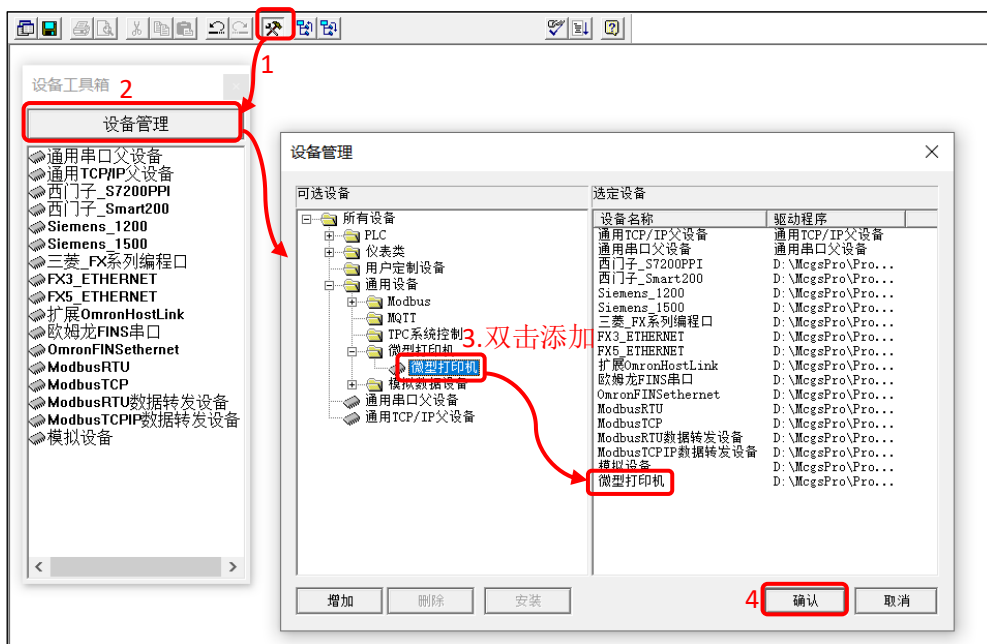


图 14.1-1 在工具箱中添加微型打印机驱动

- ③ 然后依次双击工具栏中的【通用串口父设备】和【微型打印机】添加驱动。

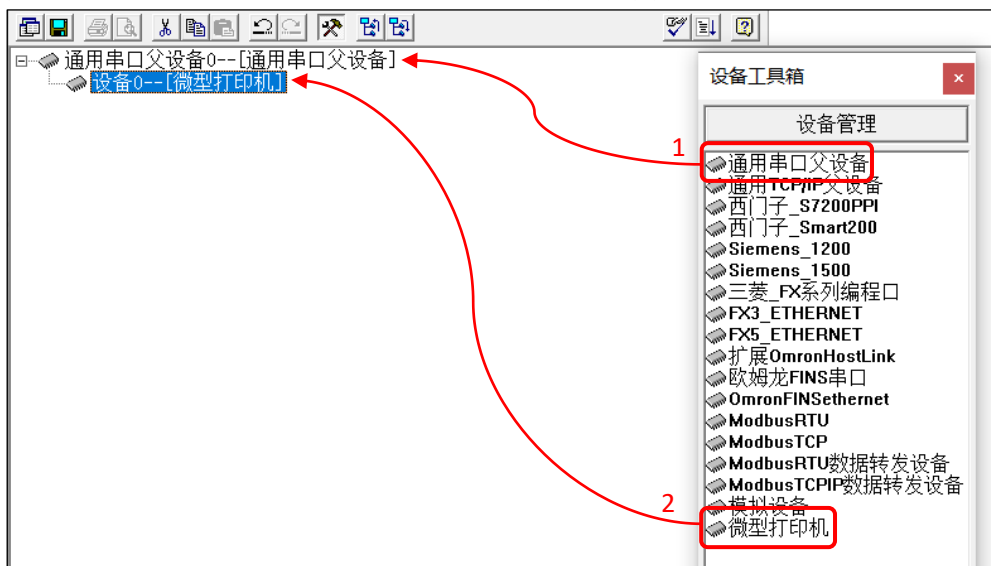


图 14.1-2 在设备窗口中添加微型打印机驱动

- ④ 双击父设备打开【通用串口设备属性编辑】界面，设置串口通讯参数，如图 14.1-3 所示。

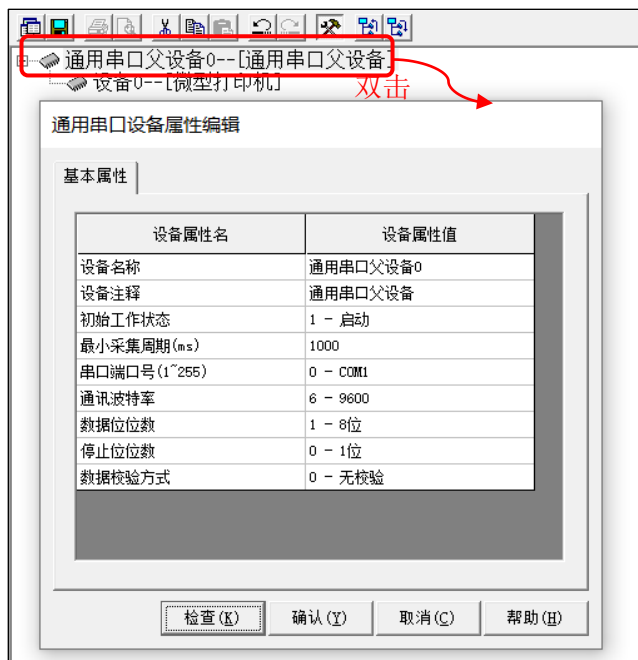


图 14.1-3 串口设备属性设置界面

- ⑤ 双击子设备打开【设备编辑窗口】，微型打印机子设备参数设置如下：

设备属性名	设备属性值
采集优化	1-优化
a — 设备名称	设备0
设备注释	微型打印机
b — 初始工作状态	1 - 启动
c — 最小采集周期(ms)	100
d — 设备类型	0 - 针式(有缓冲)
e — 通讯等待时间(ms)	100
f — 打印机超时值(s)	30
g — 曲线颜色深度(0~255)	10
h — 位图打印速率	50

图 14.1-4 子设备参数

- 设备名称：**可根据需要来对设备进行重新命名，但不能和设备窗口中已有的其它设备构件同名。
- 初始工作状态：**用于设置设备的起始工作状态，设置为启动时，在进入 MCGS 运行环境时，MCGS 即自动开始对设备进行操作，设置为停止时，MCGS 不对设备进行操作。但可以用 MCGS 的设备操作函数和策略在 MCGS 运行环境中启动或停止设备。
- 最小采集周期：**为运行时，MCGS 对设备进行操作的时间周期，单位为毫秒，默认为 100ms。由于此驱动没有需要采集的通道，另外为了便于观察设备命令的返回值，建议在测量时设置为 5000ms。
- 设备类型：**默认为针式（有缓冲）。
 - 0-针式(有缓冲)：主要适用于行点数不超过 255，且带有缓冲区的打印机。每行点数 240。如果打印机支持曲线打印命令则可完成曲线位图的打印功能。
 - 1 - 针式(无缓冲)：主要适用于行点数不超过 255，不带缓冲区的打印机。每行点数 240。如果打印机支持曲线打印命令则可完成曲线位图的打印功能。
 - 2 - 热敏：主要适用于行点数超过 255，且带有缓冲区的打印机。主要特点是打印曲线命令使用高低位格式。每行点数 384，如果打印机支持曲线打印命令则可完成曲线位图的打印功能。
- 通讯等待时间：**设备进行一次通讯的最长时间，单位为毫秒。在通讯等待时间内，如果通讯还没有完成，则报错。因此，建议通讯时间较长的设备，通讯等待时间可设长一点。默认为 200 毫秒。
- 打印机超时值：**此属性只适用于 1 - 针式（无缓冲）打印机，单位为秒。如果打印机执行单次打印

的时间超过此值，通讯状态将返回失败，默认为 30s。

- g. **曲线颜色深度**：此属性只适用于打印曲线位图设备命令，默认值为 10。在触摸屏使用打印曲线功能时，此值建议设置为 60。如果打印出的图形颜色过深，请减小此值；同理，如果打印出的图形颜色过浅，甚至缺少部分图形，请增大此值，或者考虑加宽曲线的粗细。具体请根据实际情况自行调整。
- h. **位图打印速率**：此属性只适用于打印曲线位图设备命令，默认值为 50。如果打印过程中发现打印出现乱码，可以适当增加此值，以保证正确打印曲线位图。

14.2 组态画面

组态一个工程：当点击按钮时，打印机打印从输入框输入的字符。

- ① 在实时数据库中新建一个字符串数据对象【str】。
- ② 进入用户窗口 0，新建一个输入框构件，关联数据对象【str】。如图 14.2-1 所示。

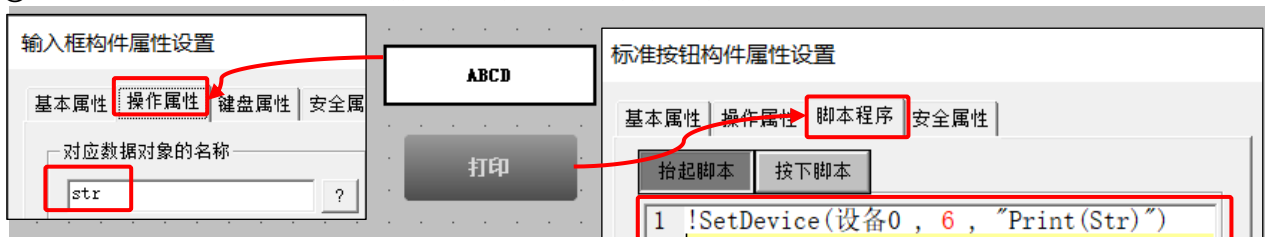


图 14.2-1 组态画面

- ③ 在按钮的脚本框中输入脚本【!SetDevice(设备 0, 6, "Print(Str)")】。如图 14.2-1 所示。

14.3 设置炜煌打印机

炜煌打印机的串行接口与 RS-232C 标准兼容，因此可直接将打印机与 TPC 相接。串行连接方式面板式和平台式插座引脚序号如图 14.3-1 和图 14.3-2 所示：

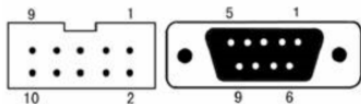


图 14.3-1 炜煌打印机面板式通讯接口定义



图 14.3-2 平台式通讯接口定义

串行接口引脚定义如下：

10针面板式	9孔面板式	平台式	信号	方向	说明
5	3	2	TxD	入	打印机从计算机接收数据
6	8	5	CTS	出	该信号高电平是，表示打印机正“忙”不能接收数据，而当该信号低电平时，表示打印机“准备好”，可以接收数据
2	6	6	DSR	出	该信号为“SPACE”状态表示打印机“在线”
9	5	7	GND	—	信号地
1	1	8	DCD	出	同信号CTS
10	—	—	+5v	入	直流+5V 3A电源输入类

图 14.3-3 串行接口引脚定义

炜煌微型打印机按键如图 14.3 所示：



图 14.3-4 炜煌微型打印机按键

修改打印机波特率和通讯模式：

- ① 按住任意键后给打印机上电，打印机会将当前串口设置状态打印出来。
- ② 按 LF 键切换打印机波特率。每按一次 LF 键，打印机立即打印出显示其当前波特率的状态报告。打印机出厂设置波特率为 9600。
- ③ 按 SEL 键切换串口的工作方式。出厂时默认设置为方式 1。
 串行工作方式 1: 一帧信息为 10 位，1 位起始位，8 位数据位，1 位停止位。
 串行工作方式 3: 一帧信息为 11 位，1 位起始位，8 位为数据位，1 位校验位，1 位停止位。
- ④ 设置完成后重启打印机。
 本样例对打印机通讯参数不做修改。
 连接完成后，打印机上电待命。

14.4 运行工程

- ① 保存工程并下载工程，然后启动运行。
- ② 在输入框中输入字符【打印测试】。
- ③ 然后点击按钮，打印机打印出输入的字符。

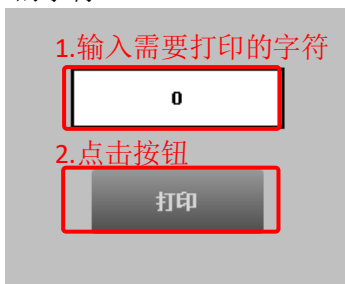


图 14.4-1 打印输入的字符

第 15 章 McgsPro 监控应用

本章将对使用 McgsPro 与设备通讯进行讲解和应用举例。

15.1 设备驱动

设备驱动的作用是建立系统与外部硬件设备的连接关系，使系统能够从外部设备读取数据并控制外部设备的工作状态，实现对工业过程的实时监控。本节主要介绍设备驱动方面的内容。

15.1.1 添加设备驱动

- ① 双击工作台中的  图标进入设备组态界面。



图 15.1-1 进入设备窗口

- ② 点击工具栏中的  图标打开的【设备工具箱】。【设备工具箱】中有两个特殊的设备 → 【通用 TCP/IP 父设备】和【通用串口父设备】。除个别驱动外，大部分驱动都是添加在父设备下，称之为 → 【子设备】。如果子设备是网口通讯，就选用【通用 TCP/IP 父设备】，子设备是串口通讯，就选用【通用串口父设备】。

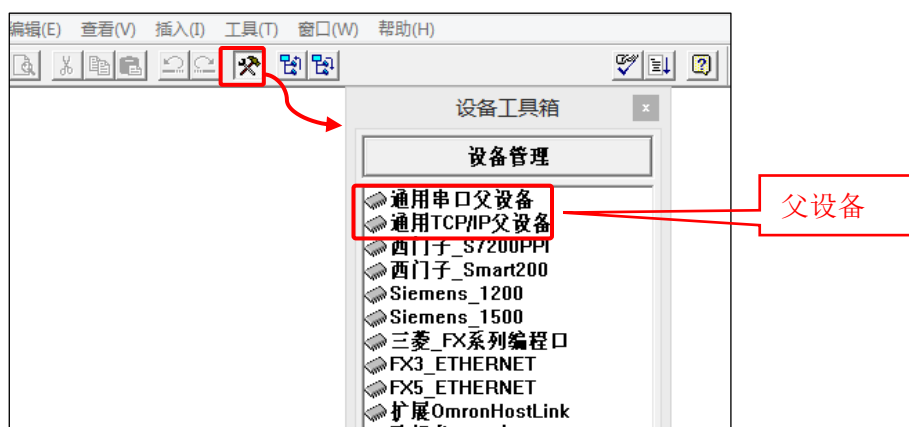


图 15.1-2 父设备

以添加 ModbusRTU 的驱动为例。首先双击设备工具箱中的【通用串口父设备】，然后双击【ModbusRTU】或直接拖拽【ModbusRTU】到父设备下。在弹出的对话框中选择【是】，添加完成。

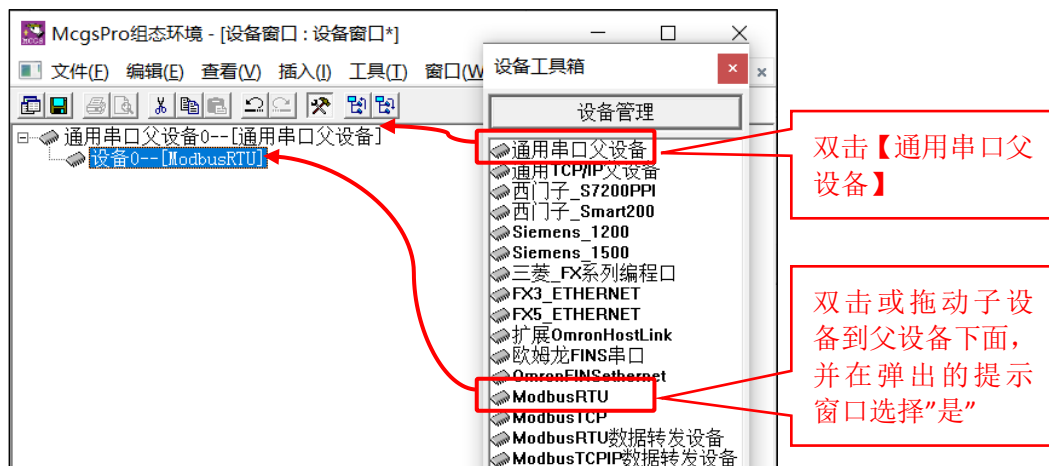


图 15.1-3 添加 ModbusRTU 驱动

网口设备的添加步骤相同。

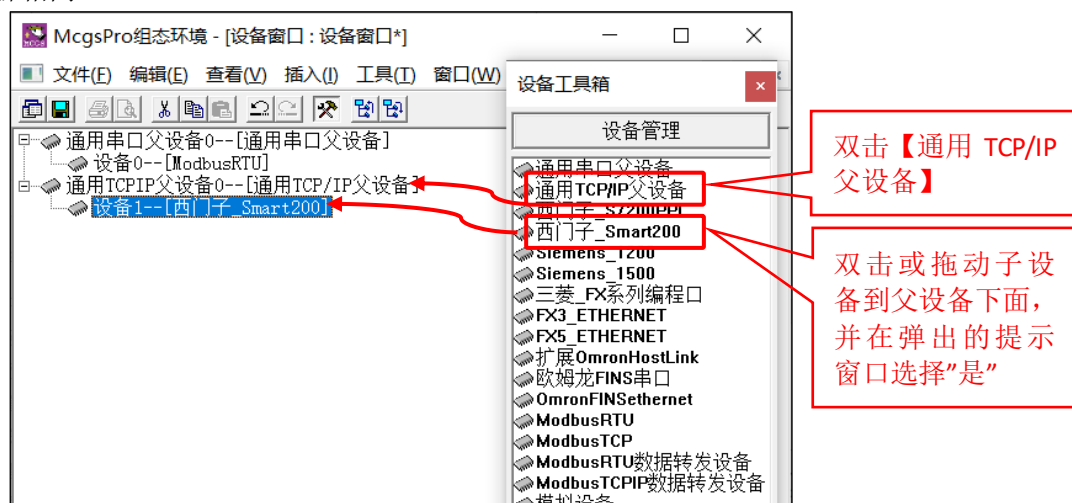


图 15.1-4 添加西门子_Smart200 网口驱动

15.1.2 增加设备驱动

若【设备工具箱】中没有需要的驱动，可自行增加，步骤如下：

- ① 点击【设备工具箱】的【设备管理】按钮。



图 15.1-5 点击设备管理按钮

- ② 在打开的【设备管理】界面中，选中左侧目录树中需要的驱动，双击或点击下方的【增加】按钮，可以将该驱动添加到右侧【选定设备】窗口中，确定后可添加到【设备工具箱】中。

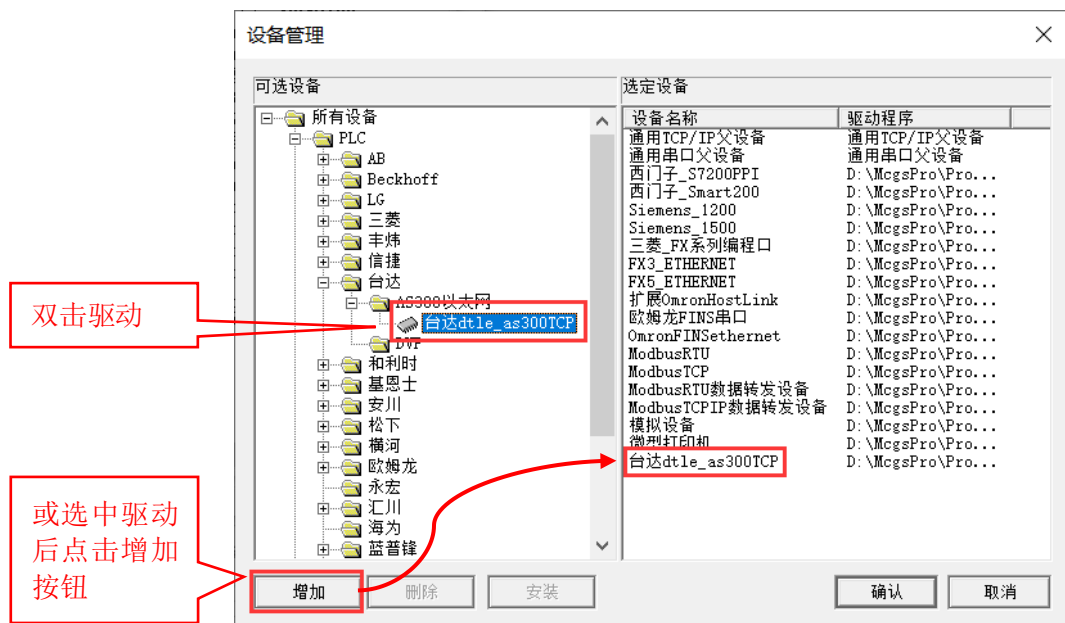


图 15.1-6 给工具箱添加驱动

15.1.3 安装设备驱动

若左侧的目录树中没有需要的驱动，则可点击下方的【安装】按钮，指定新增驱动程序所在的路径后确定安装。

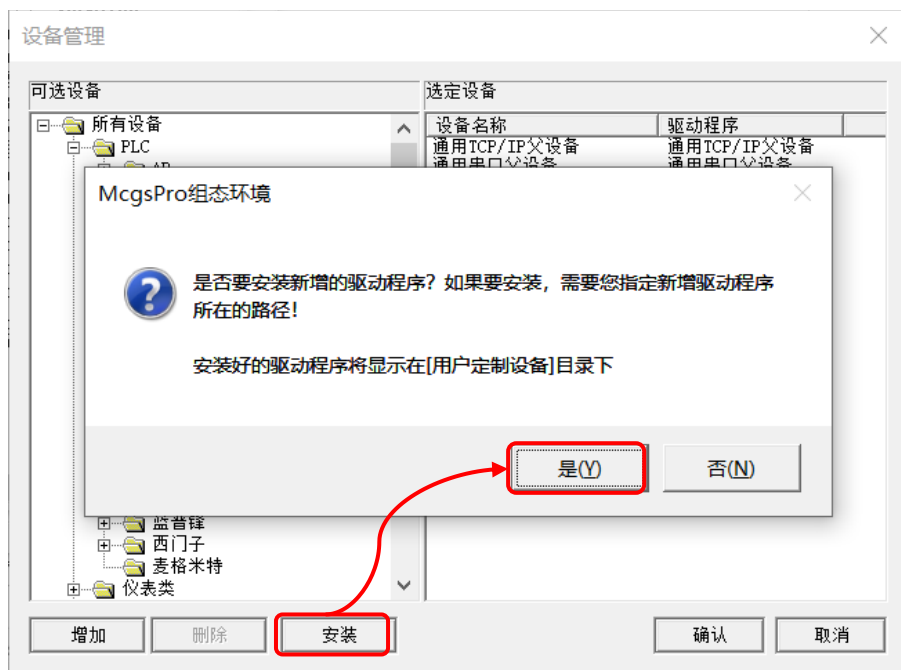


图 15.1-7 安装软件中没有的驱动

15.1.4 查找设备驱动

当用户工程已经加载的驱动无法找到时（如：驱动存放位置不在以前添加的位置），双击驱动会弹出驱动搜索功能界面。

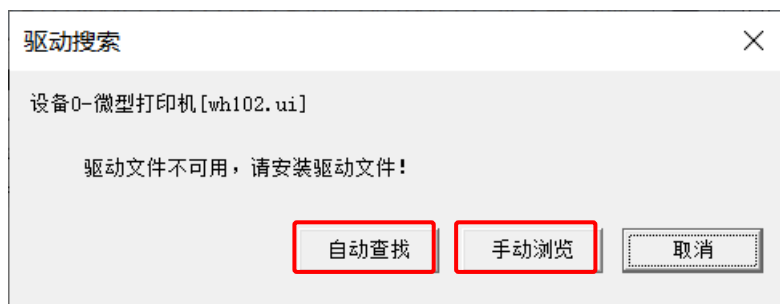


图 15.1-8 查找丢失的驱动

若驱动文件已放在了【(软件安装路径)\Program\Drivers】文件夹下，则可以选择【自动查找】功能来搜索驱动，此时驱动将使用新的加载路径。

若驱动文件未放在【(软件安装路径)\Program\Drivers】文件夹下，自动搜索功能搜索不到匹配的驱动，此时用户可点击【手动浏览】按钮，在弹出的窗口中选择驱动所在路径下对应的驱动文件，此时驱动将使用用户选择的驱动文件进行加载。

15.2 设置设备参数

添加好设备驱动后，还需要对父设备和子设备的参数进行设置，才能实现与外部设备间的正常通讯。

15.2.1 父设备参数

首先需要对父设备参数进行设置。

- 设置通用串口父设备

双击【通用串口父设备】，打开【通用串口设备属性编辑】界面。按照所通讯设备的要求来设置串口端口号、通讯波特率等参数。如设置不正确，会导致无法正常通讯。【通用串口父设备】属性编辑界面如图 15.2-1 所示。



图 15.2-1 通用串口设备属性编辑界面

- a. **初始工作状态**：指运行时设备的初始工作状态，如父设备处于停止状态，则父设备下挂接的所有子设备都处于停止状态。
- b. **最小采集周期**：运行时，对设备进行定时操作的时间周期，单位为毫秒。
- c. **串口端口号**：可选范围 COM1~COM254，默认值 COM2。
- d. **通讯波特率**：可选范围 9600,19200,38400 等，默认值 9600。

- e. **数据位位数**: 可选范围 7、8，默认值 8。
- f. **停止位位数**: 可选范围 1、1.5、2，默认值 1。
- g. **奇偶校验位**: 可选范围无校验、奇校验、偶校验、标志位、空格位，默认值无校验。
- **设置通用 TCP/IP 父设备**
 双击【通用 TCP/IP 父设备】，打开【通用 TCP/IP 设备属性编辑】界面。同样的，按所通讯设备要求设置 IP 地址、端口号等参数。【通用 TCP/IP 父设备】属性编辑界面如图 15.2-2 所示。



图 15.2-2 通用 TCP/IP 父设备编辑界面

- a. **初始工作状态**: 指运行时设备的初始工作状态，如父设备处于停止状态，则父设备下挂载的所有子设备都处于停止状态。
 - b. **最小采集周期**: 运行时，对设备进行定时操作的时间周期，单位为毫秒。
 - c. **网络类型**: 可选择 UDP 或 TCP 中任意一种网络，服务器与客户端应使用同一种网络类型。
 - d. **服务器/客户设置**: 设置本工作站为服务器或客户端。
 - e. **本地 IP 地址**: 指触摸屏在 TCP/IP 网络中的 IP 地址。
 - f. **本地端口号**: 指触摸屏所使用的网络 TCP/IP 端口的地址。
- 注意**: 若无特殊要求 TCP 客户端的本地端口号建议设置为 0，为随机绑定本地端口号。
- g. **远程 IP 地址**: 指和触摸屏进行通讯的远程设备的 IP 地址或域名地址。
 - h. **远程端口号**: 指远程设备使用的网络 TCP/IP 端口的地址。

15.2.2 子设备参数

不同子设备的参数配置不同，可在在设备帮助中查看相关说明。

- ① 双击子设备，进入【设备编辑窗口】。【设备编辑窗口】的详细说明请参考：[进阶篇 1.4.2 设备窗口介绍](#)。

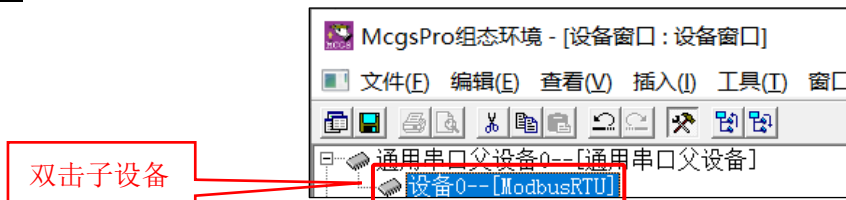


图 15.2-3 双击子设备

- ② 点击【打开设备帮助】按钮，弹出该子设备对应的驱动帮助，包含【硬件配置】【参数设置】【设备属性】【通道信息】【设备命令】【通讯状态】【附录】（在附录中会有 PLC 端如何设置步骤，可供参考）

等信息，点击帮助页上方的蓝色字体超链接可快速定位，用户可在需要的时候自行查看相关内容。



图 15.2-4 设备编辑窗口

15.2.3 设备命令

McgsPro 除了可以在组态过程中对父设备或子设备的参数进行设置外，也可以在工程运行过程中，通过设备命令，对父设备或子设备参数进行设置。

● 函数介绍和应用举例

设备命令的格式如下：

!SetDevice(DevName,DevOp,CmdStr)

参数含义：DevName，字符型，设备名

DevOp，数值型，设备操作码

CmdStr，字符串，设备命令，只有当 DevOp=6 时 CmdStr 才有意义

DevOp 取值范围及相应含义：

= 1，启动设备

= 2，停止设备

= 3, 测试设备的工作状态

= 4, 启动设备工作一次

= 5, 改变设备的工作周期, CmdStr 包含新的工作周期, 单位 ms

= 6, 执行指定的设备命令, CmdStr 中包含指定命令的格式

其中 DevOp 值为 6 时, CmdStr 可以支持多种函数, 如父设备的参数修改, 子设备中通道数值读写等。

通用 TCP/IP 父设备对应的设备命令和用法如下:

设备命令	命令格式	命令举例
获取以太网信息 GetIpInfo()	GetIpInfo(以太网信息)	例: !SetDevice(通用 TCPIP 父设备 0,6,"GetIpInfo(IpInfo)") 获取“通用 TCPIP 父设备 0”的以太网信息
修改本地 IP 和端口号 SetLocal()	SetLocal(IP 地址,端口号)	例: !SetDevice(通用 TCPIP 父设备 0,6,"SetRemote(200.200.200.190,0)") 设置“通用 TCPIP 父设备 0”本地 IP 为 200.200.200.190, 本地端口号为 0
修改远程 IP 和端口号 SetRemote()	SetRemote(IP 地址,端口号)	例: !SetDevice(通用 TCPIP 父设备 0,6,"SetRemote(200.200.200.191,0)") 设置“通用 TCPIP 父设备 0”远程 IP 为 200.200.200.191, 远程端口号为 0

通用串口父设备对应的设备命令和用法如下:

设备命令	命令格式	命令举例
修改串口端口号 ChangePort()	ChangePort(端口号)	例: !SetDevice(通用串口父设备 0,6,“ChangePort(1)”) 设置“通用串口父设备 0”的串口端口号为 1-COM2
修改通讯参数 ChangePortParam()	ChangePortParam(波特率,数据位,停止位,校验位,返回值)	例: !SetDevice(通用串口父设备 0,6,"ChangePortParam(9600,8,1,0,nReturn) “) 设置“通用串口父设备 0”波特率为 9600,数据位 8,停止位 1,校验方式无校验。函数返回值放入 nReturn

通过设备命令控制子设备:

以 ModbusRTU 为例, 介绍如何利用设备命令控制子设备的启停。

① 首先在设备窗口中新建如图 15.2-5 所示的驱动:

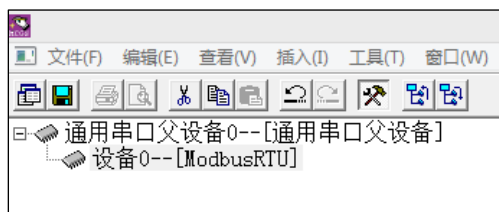


图 15.2-5 添加 ModbusRTU 驱动

② 然后在用户窗口 0 中新建两个按钮, 将按钮 1 的显示文本设置为【启动设备 0】, 脚本编辑为【!SetDevice(设备 0,1,"")】, 如图 15.2-6 所示。

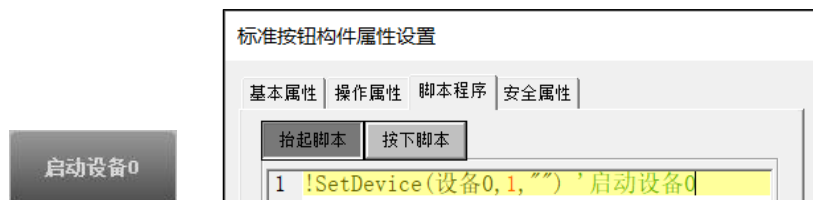


图 15.2-6 按钮 1 设置

- ③ 将按钮 2 的显示文本设置为【停止设备 0】，脚本编辑为【!SetDevice(设备 0,2,"")】，如图 15.2-7 所示。

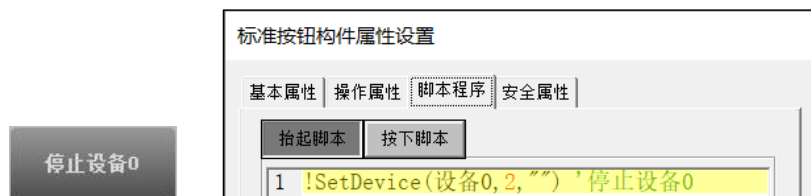


图 15.2-7 按钮 2 设置

- ④ 工程运行时，点击按钮 1，则设备 0 启动；点击按钮 2，则设备 0 停止。

- 通过设备命令，对数据进行读写操作。

设备命令除了可以在运行过程中对父设备或子设备的参数进行设置外，还可以对子设备中通道数值进行读写操作。不同子设备的函数不同，可以在每个驱动的帮助中心查看。子设备常用函数如下：

Read（寄存器名称，寄存器地址，数据类型=读取值）：按照指定数据格式读取寄存器某一地址数值。

Write（寄存器名称，寄存器地址，数据类型=写入值）：将数值以指定数据格式写入寄存器某一地址中。

ReadP：格式 1: ReadP(寄存器名称，寄存器起始地址，数据类型，操作个数 n，读取值 1，读取值 2,..., 读取值 n)

格式 2: ReadP(寄存器名称，寄存器起始地址，数据类型，操作个数 n，读取值 1，读取值 2,..., 读取值 n, 返回值)

从寄存器指定地址开始，按照指定数据类型连续读取 n 个数值，将读取值分别存放入变量。

WriteP：格式 1: WriteP(寄存器名称，寄存器起始地址，数据类型，操作个数 n，写入值 1，写入值 2,..., 写入值 n)

格式 2: WriteP(寄存器名称，寄存器起始地址，数据类型，操作个数 n，写入值 1，写入值 2,..., 写入值 n, 返回值)

从寄存器指定地址开始，按照指定数据类型连续写入 n 个数值。

ReadPV：格式 1: ReadPV(寄存器名称，寄存器起始地址，数据类型，操作个数 n，读取值)

格式 2: ReadPV(寄存器名称，寄存器起始地址，数据类型，操作个数 n，读取值，返回值)

从寄存器指定地址开始，按照指定数据类型连续读取 n 个数值，将读取值分别放入以“读取值”为起始，连续 n 个变量中。

WritePV：格式 1: WritePV(寄存器名称，寄存器起始地址，数据类型，操作个数 n，写入值)

格式 2: WritePV(寄存器名称，寄存器起始地址，数据类型，操作个数 n，写入值，返回值)

读取以“写入值”为起始，连续 n 个的变量数值，将读取值从寄存器指定地址开始，按照指定数据类型分别写入 n 个寄存器中。

ReadBlock：格式 1: ReadBlock(寄存器名称，寄存器起始地址，[数据类型 1][数据类型 2][数据类型 n]，操作个数 n，写入变量)

格式 2: ReadBlock(寄存器名称，寄存器起始地址，[数据类型 1][数据类型 2][数据类型 n]，操作个数 n，写入变量, 返回值)

从寄存器指定地址开始，按照指定数据类型格式连续读取 n 组数据，将读取值以特定 CSV 格式存入字符变量中。

WriteBlock：格式 1: WriteBlock(寄存器名称，寄存器起始地址，[数据类型 1][数据类型 2][数据类型 n]，操作个数 n，写入值)

格式 2: WriteBlock(寄存器名称，寄存器起始地址，[数据类型 1][数据类型 2][数据类型 n]，操作个数 n，写入值，返回值)

将字符串变量中的数据，以指定格式解析并按照指定数据类型格式，写入寄存器指定地址开始的连续地址中。

数据读写举例：

!SetDevice(设备 0,6,"Read(4,10,WUB=Data00;4,40,DF=Data01)")

释义：读取寄存器 4 区地址 10 的 16 位无符号值，和地址 40 的浮点数值，放入 MCGS 变量 Data00,Data01 中。

!SetDevice(设备 0,6,"Write(4,10,WUB=Data00;4,40,DF=Data01)")

释义：将 Data00,Data01 的值分别以 16 位无符号和浮点数写入 4 区寄存器地址 10 和 40 中。

15.3 添加通道并连接变量

设置完设备参数后，还需要将设备通道与实时数据库中的变量连接，通道数据才可通过变量被用户窗口中的构件和策略脚本等所使用。

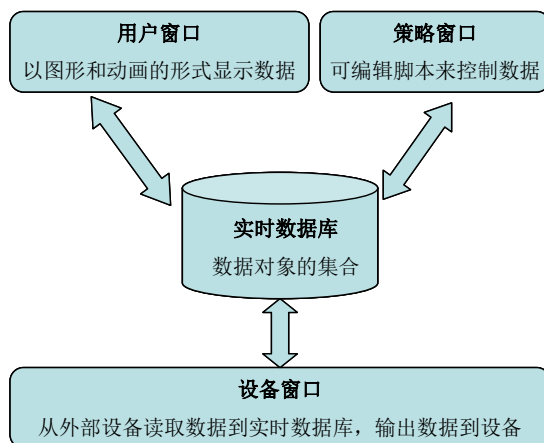


图 15.3-1 McgsPro 组态软件原理图

15.3.1 添加通道

- ① 双击【ModbusRTU】子设备进入其设备编辑窗口，点击右侧的【增加设备通道】按钮。选择【通道类型】为【3 区输入寄存器】，【数据类型】为【16 位无符号二进制】，【通道地址】为【0】，【通道个数】为【1】个。其他设置保持默认不变。点击【确认】。



图 15.3-2 添加通道

- ② 组态结果如图 15.3-3 所示。

索引	连接变量	通道名称	通道处理	地址偏移	采集频次	信息备注	操作
0000		通讯状态			1		增加设备通道
0001		只读10001			1		删除设备通道
0002		只读10002			1		删除全部通道
0003		只读10003			1		快速连接变量
0004		只读10004			1		删除连接变量
0005		只读10005			1		删除全部连接
0006		只读10006			1		通道处理设置
0007		只读10007			1		通道处理删除
0008		只读10008			1		
0009		只读3WUB0001			1		

图 15.3-3 添加通道结果

15.3.2 快速连接变量

- ① 点击【设备编辑窗口】右侧的【快速连接变量】按钮，弹出【快速连接】界面。
- ② 此时，若选择【自定义变量连接】，则会按照在该界面的设置生成连接变量；若选择【默认设备变量连接】，则会以【设备名称_读写方式_通道名称】为规则生成所有通道的连接变量（无论通道是否已连接变量）。本例选择使用【默认设备变量连接】，点击确认，关闭【快速连接】界面。注意，此时只是在当前界面的【连接变量】列填入了变量的名称，实时数据库中还未添加这些变量。
- ③ 然后点击确认，关闭【设备编辑窗口】。系统弹出对话框告知变量未定义，询问是否添加变量，选择【全部添加】，则系统自动在实时数据库中添加未定义的变量。

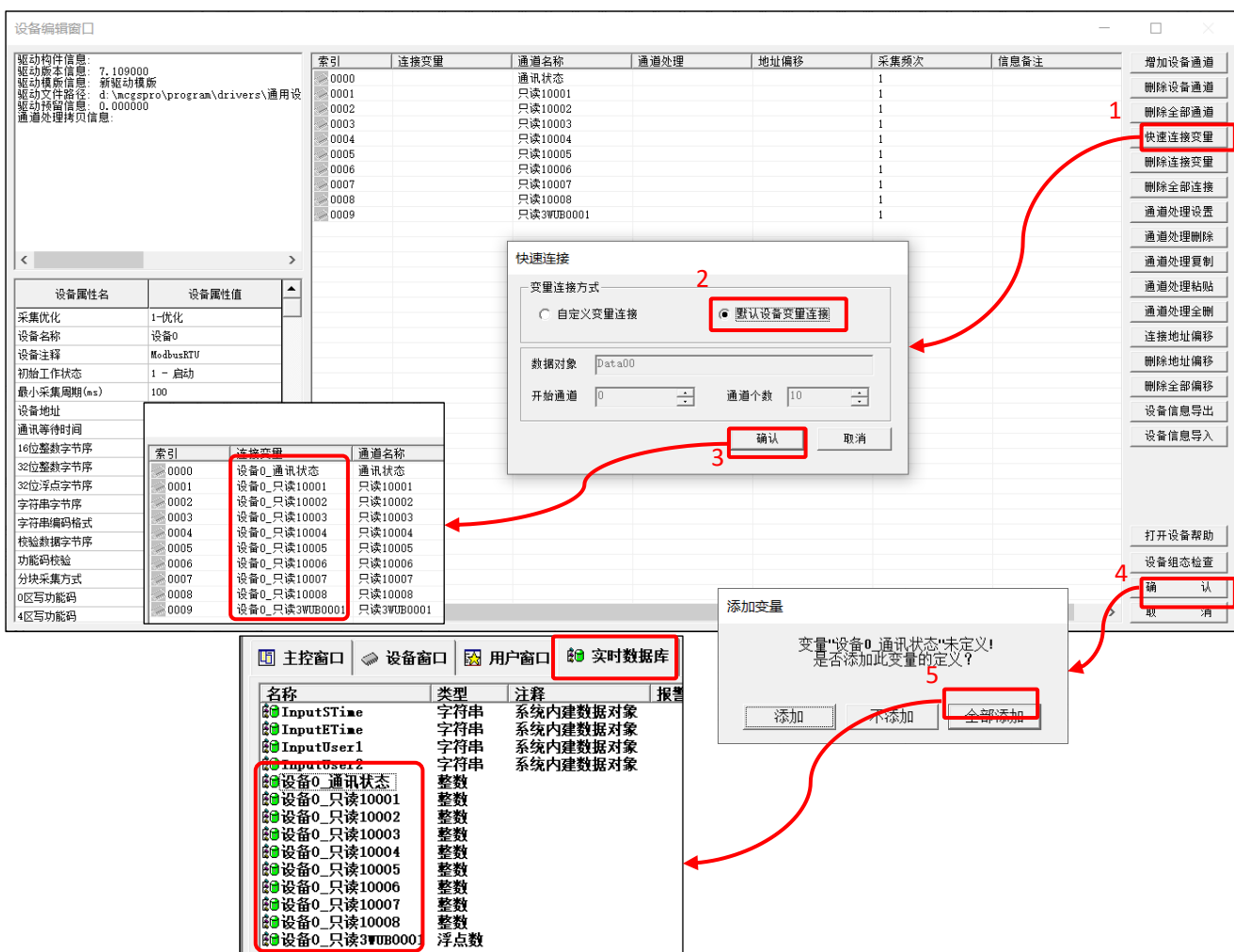


图 15.3-4 快速连接变量

15.3.3 其他连接变量的方式

方法一：若在建立通道前，已经在实时数据库中定义了变量，则可以在建立完通道后，在【连接变量】列点击右键，弹出【变量选择】界面。然后双击选择需要的变量即可。

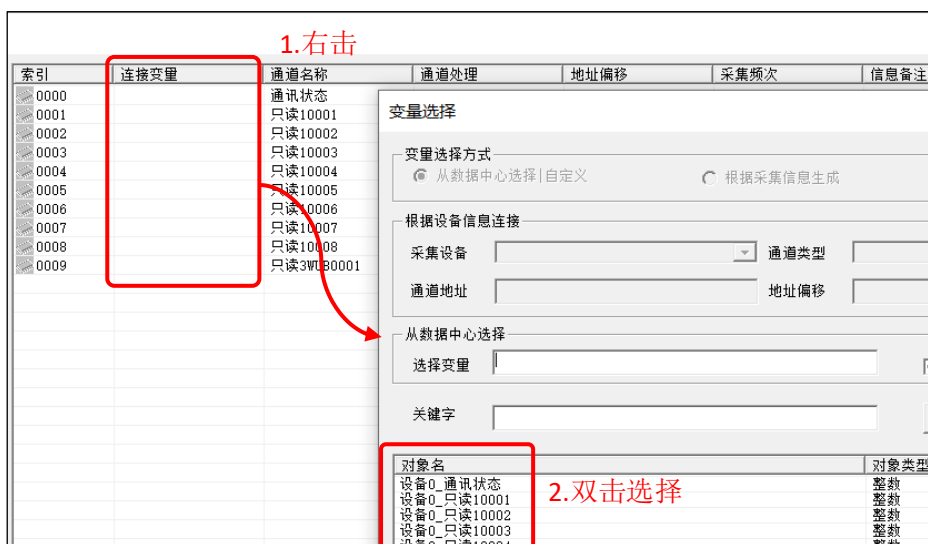


图 15.3-5 其他连接变量方法一

方法二：也可以在【添加设备通道】时，点击【连接变量】右侧的图标按钮，弹出【变量选择】设置界面，双击选择已经定义好了的变量如【Data1】。可以看到，【Data1】出现在了【连接变量】右侧的输入框中。也可以直接在该输入框处输入变量名。然后点击确认关闭【添加设备通道】界面。

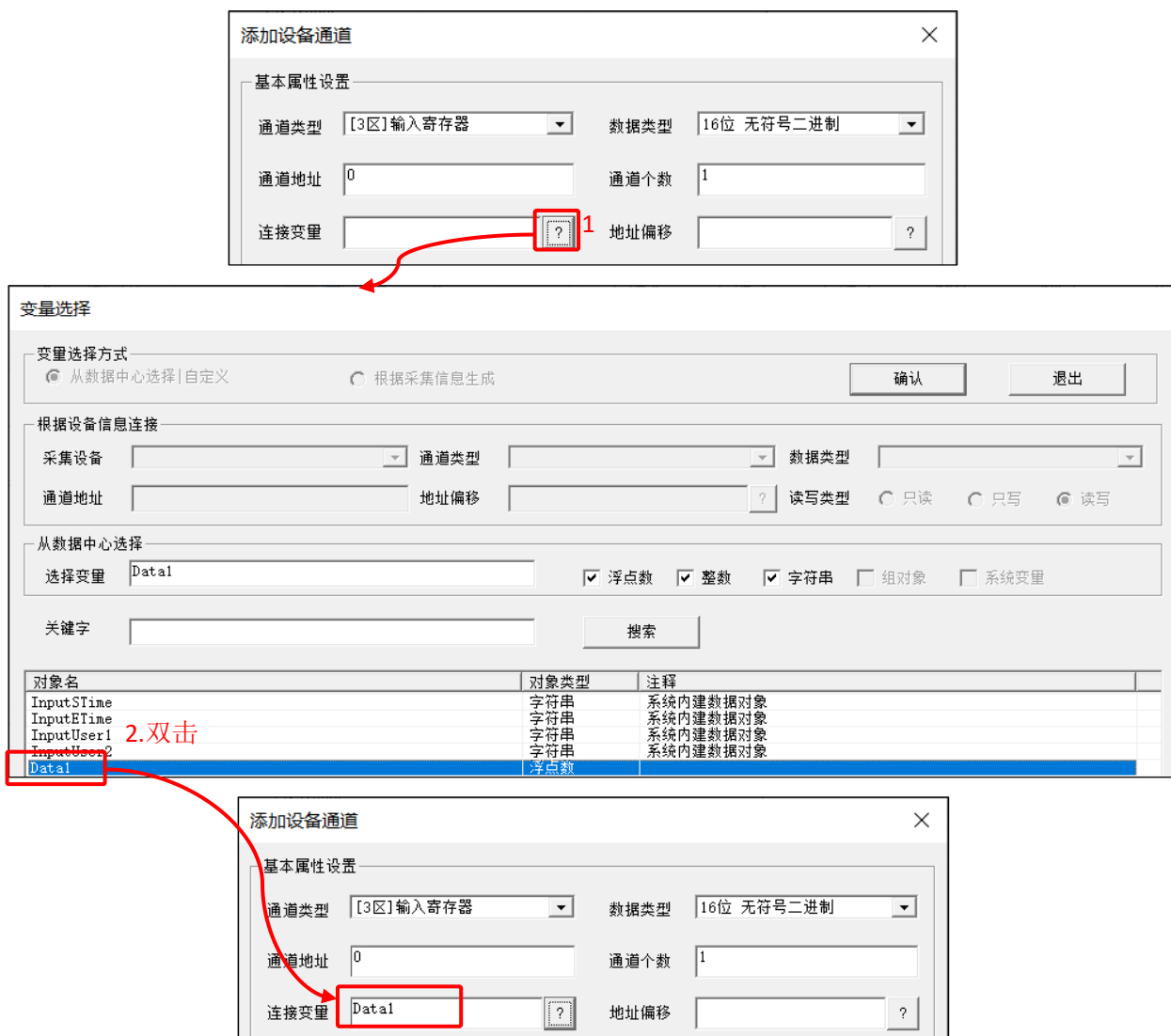


图 15.3-6 其他连接变量方法二

组态结果如图 15.3-7 所示。



图 15.3-7 连接结果

15.4 McgsPro 与西门子 200PPI 串口通讯及监控

本节样例实现的功能是：使用 TPC 接收西门子_S7200PPI 发来的信号值，当信号值为 1 时，画面中的指示灯变绿；当信号值为 0 时，指示灯变红。使用到的地址为 I0.0，通讯方式为 RS485。

15.4.1 硬件连接

MCGS 软件与设备通讯之前，必须保证通讯连接正确。

通讯连接方式：

- (1)采用标准串口型号的西门子 PC/PPI 电缆。
- (2)TPC 触摸屏的 RS485 接口的 A 正 B 负与 PLC 编程口 3 正 8 负连接，如图 15.4-1 所示。

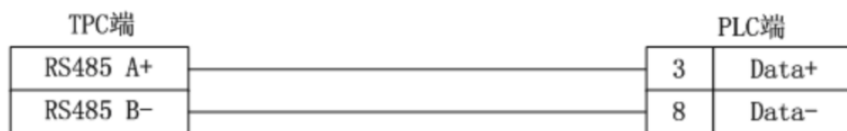


图 15.4-1 接线图

TPC 的 9 针串口引脚定义如图 15.4-2 所示。

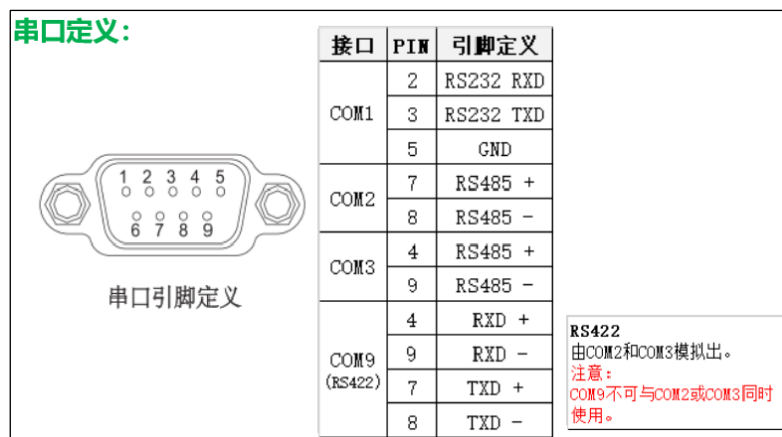


图 15.4-2 TPC9 针串口引脚定义

15.4.2 TPC 通讯设置

15.4.2.1 父设备参数设置

- ① 在 McsgPro 软件中，进入设备窗口，将【西门子_S7200PPI】驱动挂载到【通用串口父设备】下，如图 15.4-3 所示。

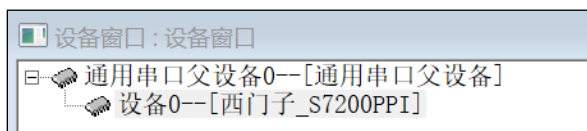


图 15.4-3 添加驱动

- ② 双击【通用串口父设备】，进入其属性编辑界面，参数设置如图 15.4-4 所示。



图 15.4-4 父设备参数设置

15.4.2.2 子设备参数设置

【西门子_S7200PPI】子设备参数设置如图 15.4-5 所示，一般保持默认即可。

设备属性名	设备属性值
采集优化	1-优化
设备名称	设备0
设备注释	西门子_S7200PPI
初始工作状态	1 - 启动
最小采集周期(ms)	100
设备地址	2
通讯等待时间	500
快速采集次数	0
采集方式	0 - 分块采集

图 15.4-5 子设备参数设置

15.4.3 PLC 通讯设置

进入 Step7 - Micro/WIN，点击系统块打开通信端口设置，设置通信中的地址（默认 2）和波特率（9.6Kbps）和 McgsPro 中父设备通讯参数一致。

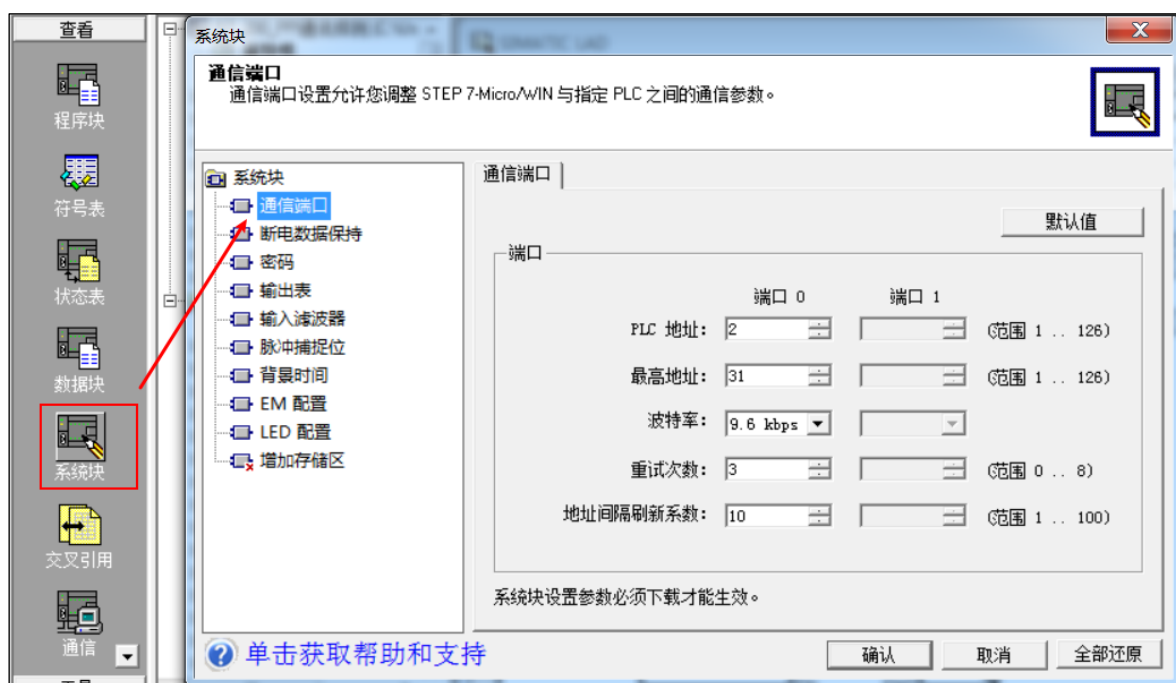


图 15.4-6 PLC 通讯参数设置

设置完后，将程序下载到 PLC。

15.4.4 组态画面

- ① 在 McgsPro 中，依次进入工作台 → 用户窗口 → 窗口 0。
- ② 点击工具箱中的插入元件，弹出【元件图库管理】界面。
- ③ 依次选择图库类型为【公共图库】 → 指示灯 → 指示灯 14 → 确定。
- ④ 指示灯 14 出现在用户窗口的左上角，将它拖拽到合适的位置和大小。

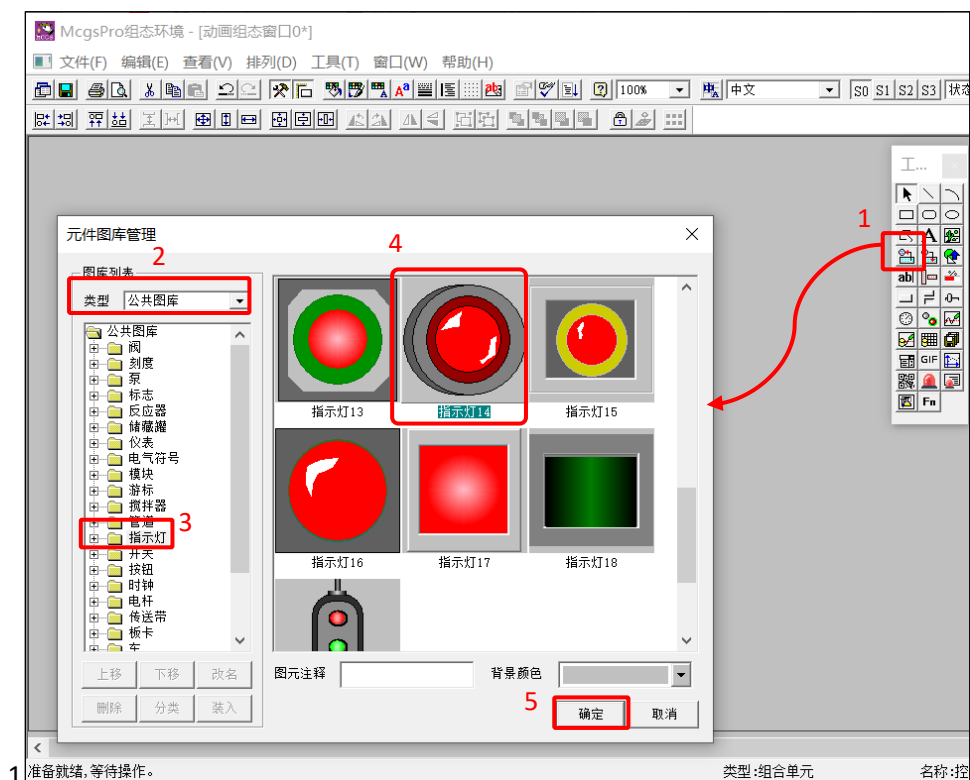


图 15.4-7 从添加指示灯

- ⑤ 双击指示灯，弹出【单元属性设置】界面。选中【变量列表】页的【表达式】，点击出现的[?]图标，弹出【变量选择】界面。

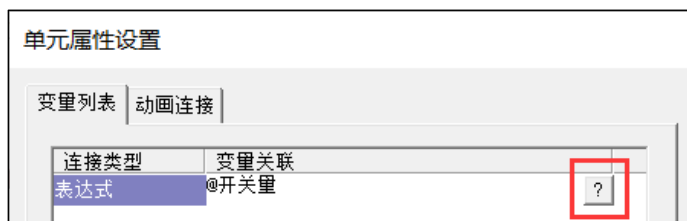


图 15.4-8 关联变量一

- ⑥ 选择变量选择方式为【根据采集信息生成】。采集设备为【设备 0[西门子_S7200PPI]】，通道类型为【I 寄存器】，数据类型为【通道的第 00 位】，通道地址为【0】，其他设置保持不变。点击确认。



图 15.4-9 关联变量二

- ⑦ 设置完成后，表达式右侧的输入框中出现了上一步定义的变量，与此同时，该变量也在实时数据库中自动生成。

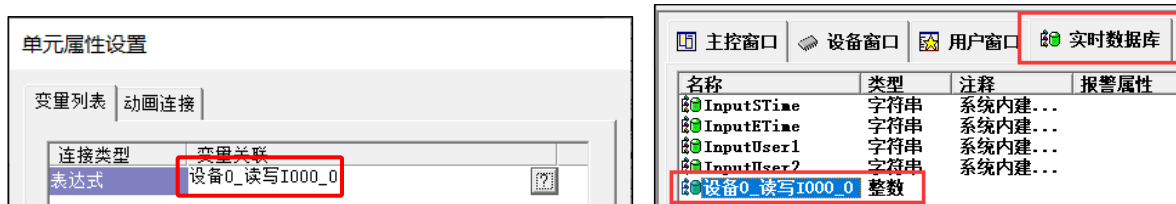


图 15.4-10 实时数据库自动生成变量

- ⑧ 设置完成后，保存工程。用 USB 线将电脑与 TPC 相连，将 TPC 上电。然后单击工具栏的下载运行图标按钮，弹出【下载配置界面】。
- ⑨ 选择【运行方式】为【联机】，连接方式为【USB 通讯】。然后单击【工程下载】按钮，下载成功后点击【启动运行】。则 TPC 上开始运行我们刚才组态好的画面。

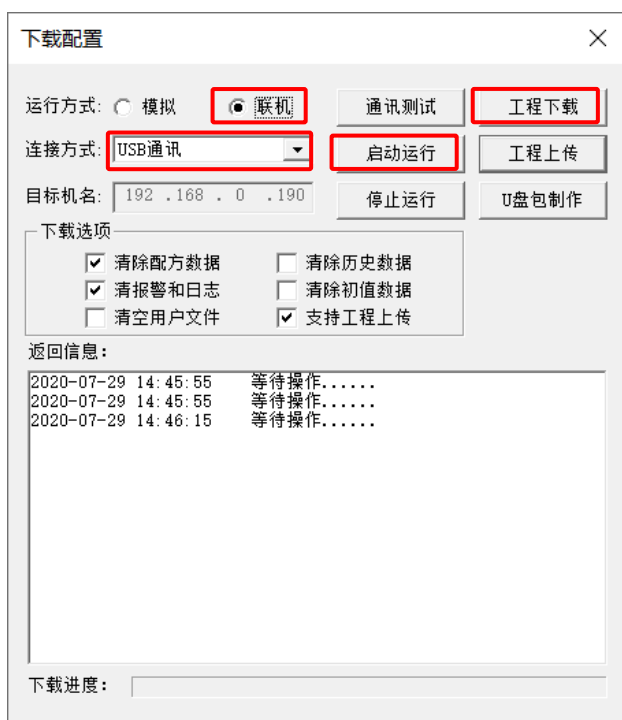


图 15.4-11 下载工程

- ⑩ 此时，若 PLC 的 I0.0 传来信号 1，则指示灯变为绿色，传来 0，则指示灯变为红色。

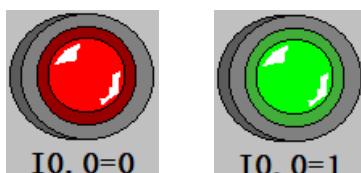


图 15.4-12 运行效果

15.5 McgsPro 与三菱 FX3U 网口通讯及监控

本节样例实现的功能是：在 TPC 画面中，第一次按下【按钮】时，发送一个信号值【1】给三菱 FX3U PLC，使 PLC 的 Y0.0 置 1；再次按下【按钮】时，发送信号值【0】给 PLC，使 PLC 的 Y0.0 置 0。本节样例的通讯方式为以太网。

15.5.1 硬件连接

通讯连接方式：采用 RJ-45 网线

15.5.2 TPC 通讯设置

15.5.2.1 父设备参数设置

在 McsgPro 软件中，进入设备窗口，将【FX3_ETHERNET】驱动挂载到【通用 TCP/IP 父设备下】，如图 15.5-1 所示。

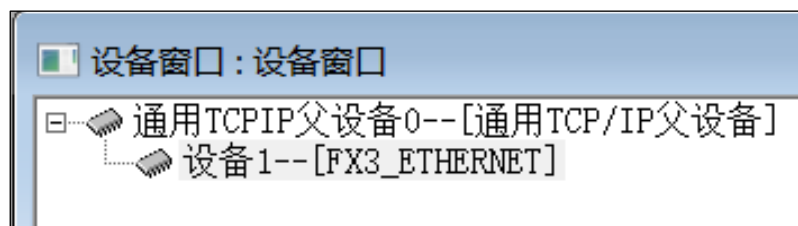


图 15.5-1 添加驱动

双击【通用 TCP/IP 父设备】，进入其属性编辑界面，通讯参数设置如图 15.5-2 所示。其中，本地 IP 地址设置成 TPC 的 IP 地址，远程 IP 地址设置成 PLC 的 IP 地址，两个 IP 地址应在同一网段。本例中本地 IP 地址为【200.200.200.190】，PLC 的 IP 地址为【200.200.200.200】。



图 15.5-2 设置父设备属性

15.5.2.3 设置子设备

双击子设备，打开 FX3_ETHERNET 的设备编辑窗口。点击【增加设备通道】按钮，在弹出的【添加设备通道】界面中，设置通道类型为【Y 输出<8 进制地址>】，通道地址为【0】，通道个数为【1】。其他设置保持默认不变，点击确认。

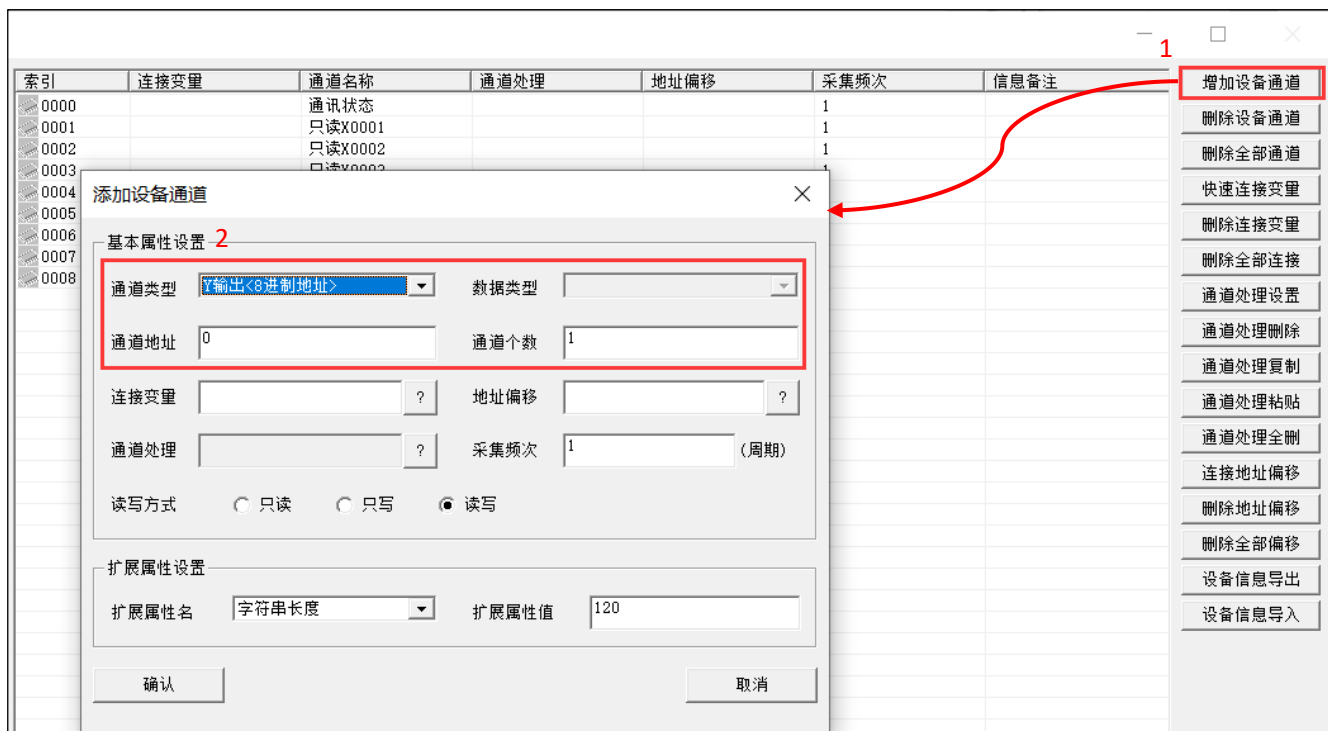


图 15.5-3 添加设备通道

设置完成之后，【设备编辑窗口】中将新增一条通道。点击【快速连接变量】按钮，在弹出的【快速连接】窗口中，选择变量连接方式为【自定义变量连接】，输入数据对象为【开关】，设置开始通道为【9】，通道个数为【1】。点击确认。

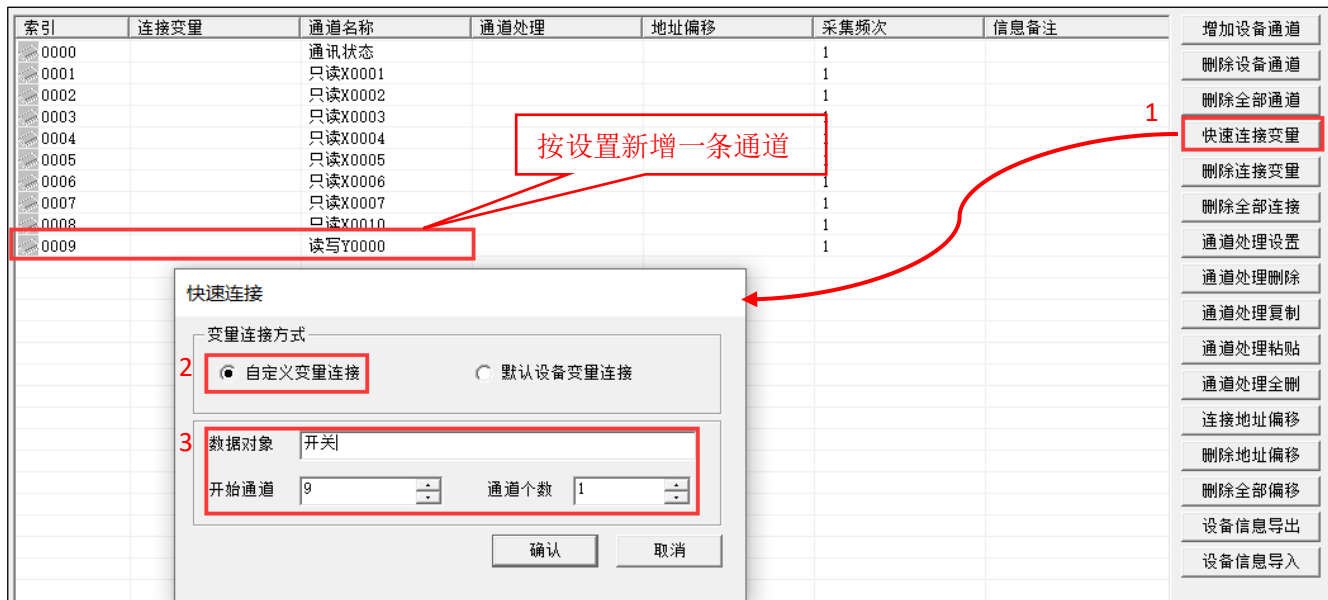


图 15.5-4 快速连接变量

点击设备编辑窗口的【确认】按钮，弹出对话框提示变量为定义，询问是否添加，选择【添加】。



图 15.5-5 自动在实时数据库中添加变量的定义

此时，实时数据库中也自动添加了一个名为【开关 00】的变量。

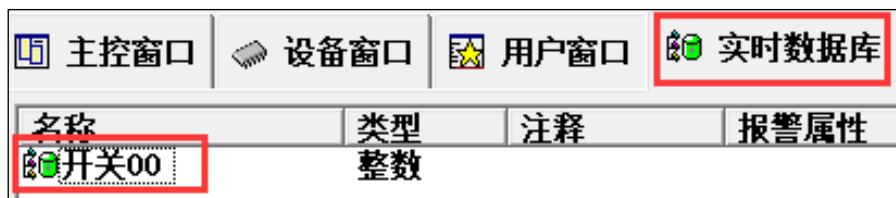


图 15.5-6 实时数据库

15.5.2.4 设置 TPC 的 IP 地址

启动 TPC，在读条时按住 TPC 液晶显示屏，进入 TPC 系统配置界面。



图 15.5-7 进入 TPC 系统配置界面

点击【系统参数设置】->【网络】修改 TPC IP 地址和组态时的本地 IP 地址设置一致。



图 15.5-8 设置 TCP/IP 地址

15.5.3 PLC 通讯设置

使用三菱 GX Works2/GX Developer 编程软件设置 PLC 通讯参数步骤如下：

15.5.3.1 新建工程

打开软件，工程 → 新建，在弹出的【新建】界面中设置 PLC 系列、型号。

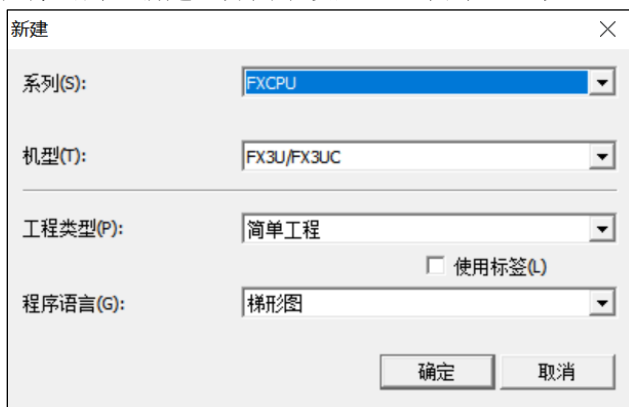


图 15.5-9 新建 PLC 工程

15.5.3.2 设置 PLC 参数

- ① 打开工程 → 参数 → PLC 参数，打开【FX 参数设置】界面。

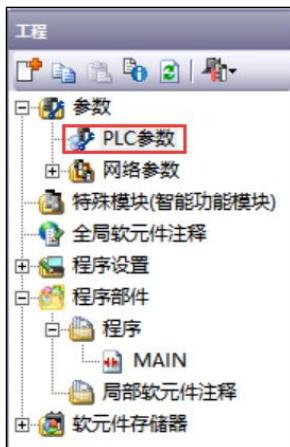


图 15.5-10 打开参数设置界面

- ② 点击【以太网端口设置】功能页，选择使用 CH 为【CH2】，按照图 15.5-11 所示设置 IP 地址，然后点

击【打开设置】按钮。弹出【以太网端口打开设置】界面。



图 15.5-11 设置 IP 地址

③ 按照图 15.5-12 所示进行设置。

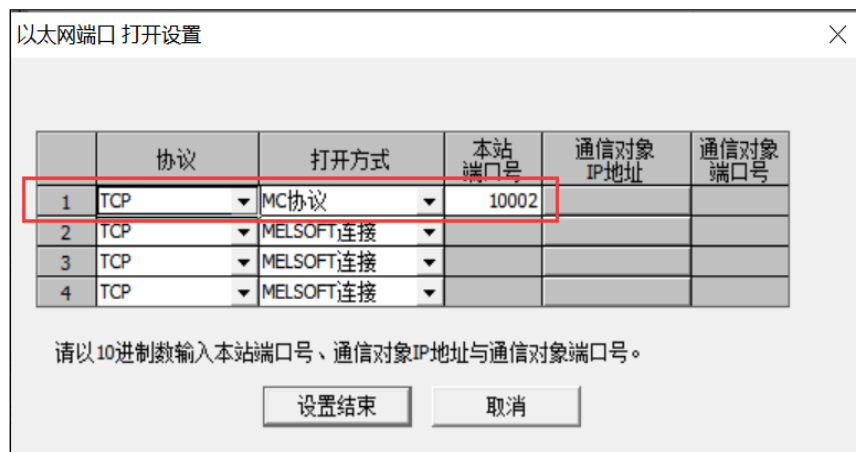


图 15.5-12 设置以太网端口

④ 点击【设置结束】按钮，结束以太网端口设置。

15.5.3.3 连接目标设置

① 点击连接目标 → 当前连接目标 → Connection1 来设置连接目标。

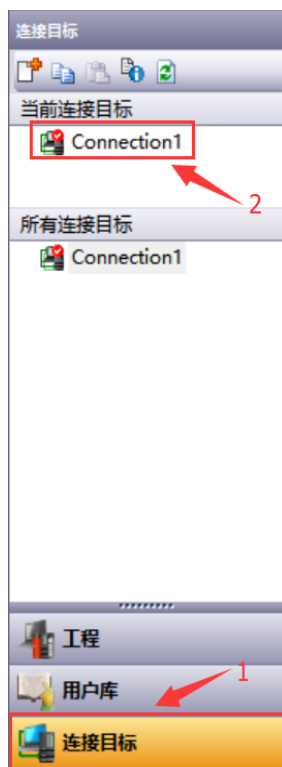


图 15.5-13 连接目标设置一

- ② 按照以下步骤对连接目标进行设置。设置步骤 1 和步骤 3 时，在弹出的对话框中直接点击【确认】即可。在步骤 2 中设置 IP 地址与 McgsPro 中父设备参数中的【远程 IP 地址】一致。

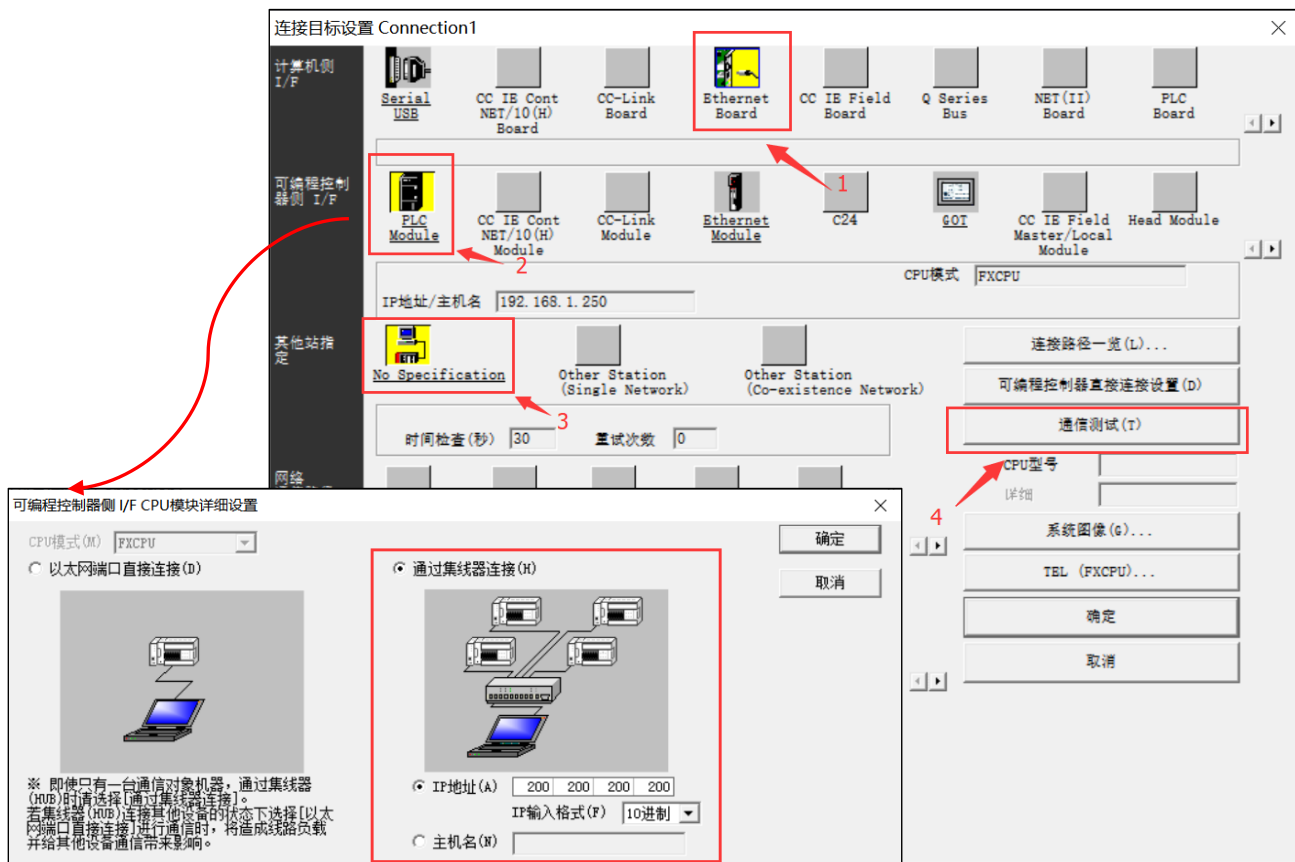


图 15.5-14 连接目标设置二

- ③ 按照顺序设置完成后，点击【通信测试】按钮，按钮下方 的 CPU 型号右侧显示 PLC 型号。
- ④ 设置结束，下载至 PLC。

15.5.4 组态画面

在 McgsPro 中进入用户窗口 0，从工具箱拖拽一个【按钮】构件到画面中合适的位置。双击该按钮，进入【操作属性】功能页，勾选【数据对象值操作】前的复选框，选择操作效果为【取反】，关联变量【开关 00】。

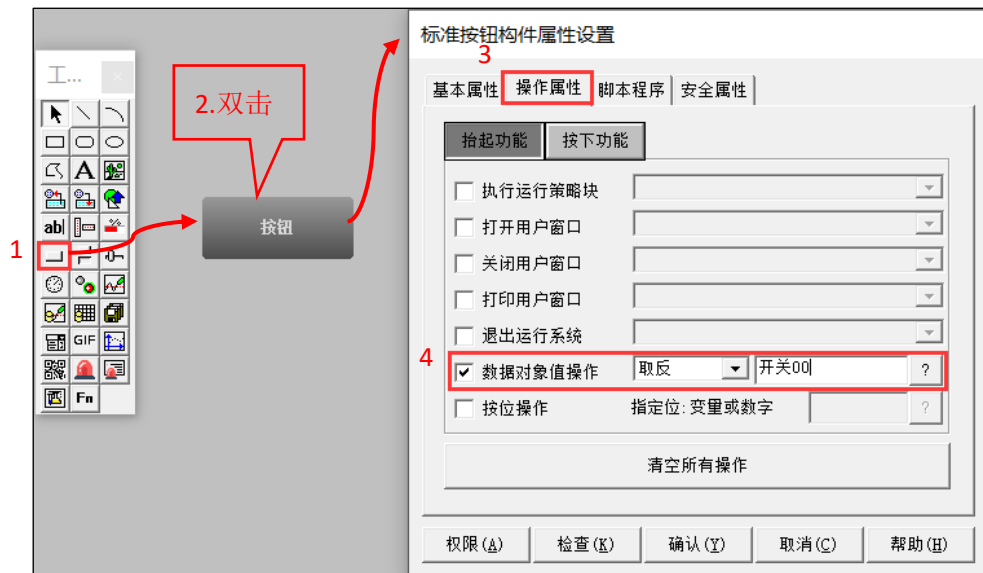


图 15.5-15 组态画面

设置完成后，保存工程。然后点击工具栏的下载运行图标按钮，弹出【下载配置界面】。

选择【运行方式】为【联机】，连接方式为【TCP/IP 网络】，将 TPC 的地址输入到【目标机名】中。然后点击【工程下载】按钮，下载成功后点击【启动运行】。则 TPC 上开始运行我们刚才组态好的画面。

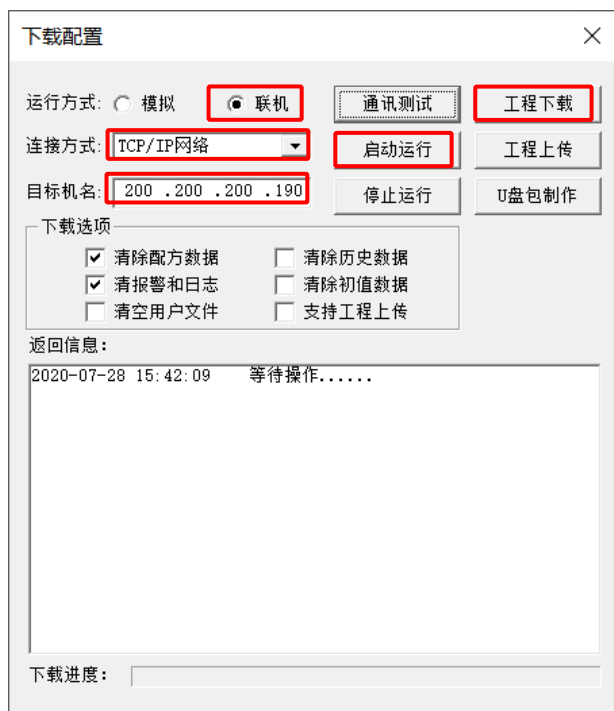


图 15.5-16 下载运行

点击 TPC 画面中的按钮，PLC 的 Y0.0 灯被点亮，再点击一次按钮，Y0.0 熄灭。

15.6 模拟通讯

在工程调试的过程中，可能会遇到暂时没有 TPC 的情况，此时可以使用 McgsPro 自带的模拟器与外部设备进行通讯调试。本节主要介绍使用模拟器与支持 ModbusRTU 协议的设备进行调试。

15.6.1 硬件连接

McgsPro 与设备通讯之前，必须保证通讯连接正确。TPC 与设备之间采用标准的 RS485 或 RS232 通讯。本节样例使用 RS485 通讯，用 USB 转 485 的电缆将电脑与设备相连。

然后进入电脑的【设备管理器】查看端口号，如图 15.6-1 所示，本节样例中的模拟器的端口号为【COM3】。

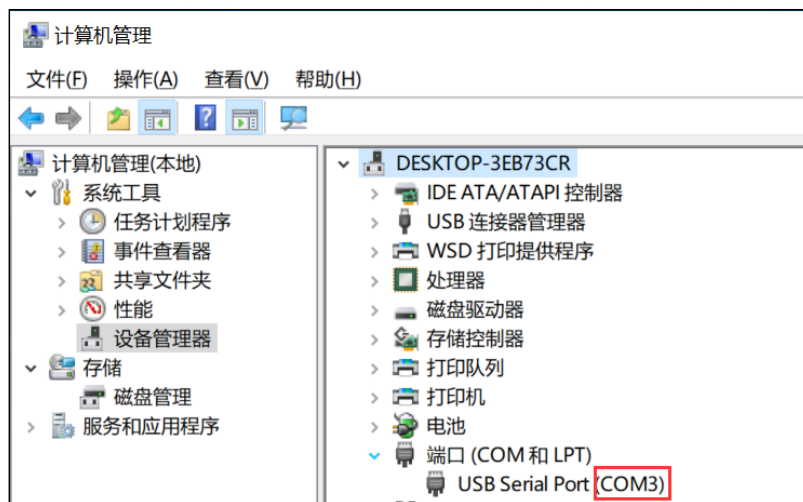


图 15.6-1 查看模拟器端口号

15.6.2 通讯设置

- ① 组态在 McgsPro 中新建一个工程，进入设备窗口，将【ModbusRTU】挂载到【通用串口父设备】下。

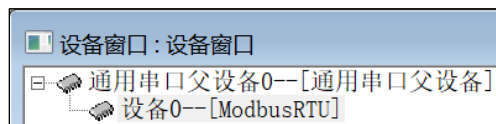


图 15.6-2 添加驱动

- ② 双击父设备，设置父设备参数如图 15.6-3 所示。



图 15.6-3 设置父设备参数

父设备通讯参数设置应与设备的通讯参数相同，否则无法正常通讯，默认为 9600、8、1、E(偶校验)。设备通讯参数的具体设置请参见对应设备手册。

- ③ 双击【ModbusRTU】，进入子设备【设备编辑窗口】。子设备参数如图 15.6-4 所示。

设备属性名	设备属性值
采集优化	1-优化
设备名称	设备0
设备注释	ModbusRTU
初始工作状态	1 - 启动
最小采集周期(ms)	100
设备地址	1
通讯等待时间	200
16位整数字节序	0 - 12
32位整数字节序	0 - 1234
32位浮点字节序	0 - 1234
字符串字节序	0 - 21
字符串编码格式	0 - ASCII
校验数据字节序	0 - LH[低字节, 高字节]
功能码校验	0 - 校验功能码
分块采集方式	0 - 按最大长度分块
0区写功能码	1 - 0x0F
4区写功能码	0 - 0x06
1区最大块长	120
0区最大块长	120
3区最大块长	120
4区最大块长	120
通讯间隔字节数	4

图 15.6-4 设置子设备参数

在【设备编辑窗口】中添加如图 15.6-5 所示的通道，再点击【快速连接变量】按钮 → 选择【变量连接方式】为【默认设备变量连接】 → 确认。设置完成后设备的通道和连接变量情况如图 15.6-5 所示。

索引	连接变量	通道名称
0000	设备0_通讯状态	通讯状态
0001	设备0_只读10001	只读10001
0002	设备0_读写00001	读写00001
0003	设备0_只读3WUB0001	只读3WUB0001
0004	设备0_读写4WUB0001	读写4WUB0001

图 15.6-5 建立通道和变量连接

然后点击【设备编辑窗口】的确认按钮，弹出对话框提示变量未定义，询问是否添加定义，选择【全部添加】。此时实时数据库中自动生成了上一步骤中连接的变量，如图 15.6-6 所示。

名称	类型	注释	报警属性
设备0_只读10001	整数		
设备0_读写00001	整数		
设备0_只读3WUB0001	浮点数		
设备0_读写4WUB0001	浮点数		
设备0_通讯状态	整数		

图 15.6-6 实时数据库

15.6.3 组态画面

进入用户窗口 0，组态如图 15.6-7 所示的画面。

设备0_通讯状态	12.34
设备0_只读10001	12.34
设备0_读写00001	12.34
设备0_只读3WUB0001	12.34
设备0_读写4WUB0001	12.34

图 15.6-7 组态画面

画面右侧列的输入框中均关联了它们左侧列标签构件显示的变量。图 15.6-8 为关联变量【设备 0_通讯状态】的输入框组态情况，其他输入框以次类推。

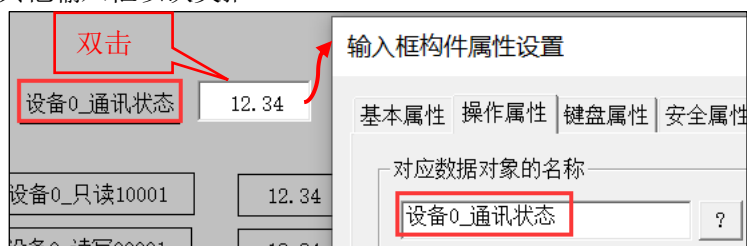


图 15.6-8 构件关联变量情况

15.6.4 模拟运行

组态完成后保存工程，然后模拟运行。步骤如图 15.6-9 所示。

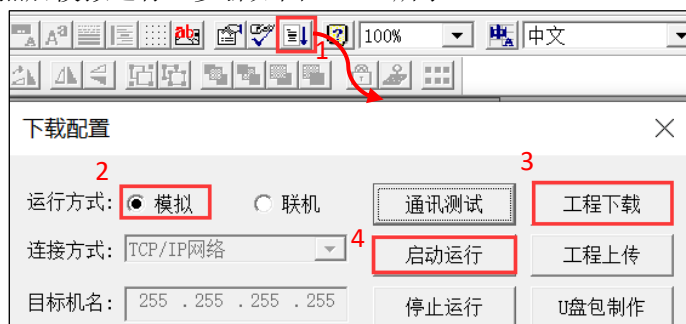


图 15.6-9 模拟运行

模拟运行窗口如图 15.6-10 所示。如果模拟器与设备通讯成功，通讯参数将显示【0】。如果通讯参数不为 0，则可以通过参考设备帮助中通讯参数值的具体含义进行排错。

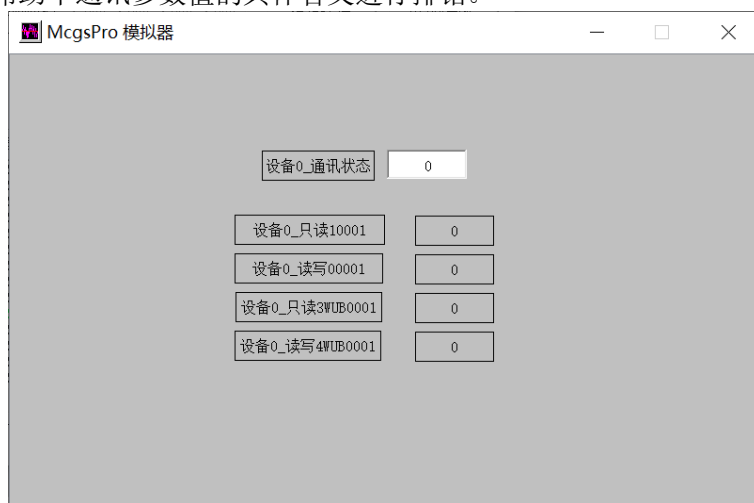


图 15.6-10 模拟器运行效果

15.7 通过通道导入导出替换 PLC

由于在 McgsPro 中，设备通道首先与实时数据库中的变量连接，再通过变量被用户窗口中的构件和策略脚本等所使用。所以当用户需要替换外部设备时，只需要将变量与原外部设备通道间的连接断开，再重新添加变量与新设备通道之间的连接，即可不用重新组态画面、脚本等，便简单、快速地更换了外部设备。

本节以将外部设备从西门子_SMART200 替换为三菱_FX3U 为例进行讲解。

15.7.1 西门子_SMART200 设备工程情况

现有一个使用 McgsPro 组态的控制冷床电机启停并显示电机状态的工程，工程画面如图 15.7-1 所示：



图 15.7-1 现有工程画面

画面中的构件都已经关联好了实时数据库中的变量。冷床 1#电机相关各个构件的变量关联情况如图 15.7-2 所示，其他电机关联变量情况以此类推。

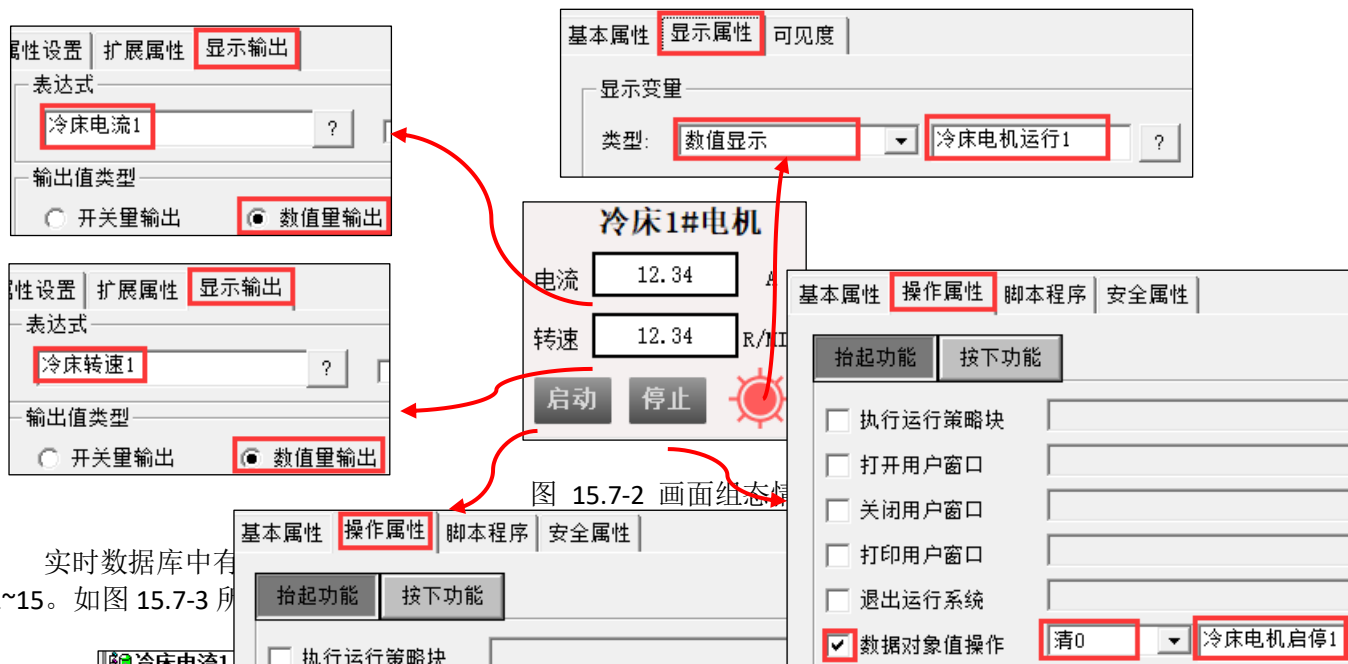


图 15.7-2 画面组态属性

实时数据库中有 1~15。如图 15.7-3 所示。

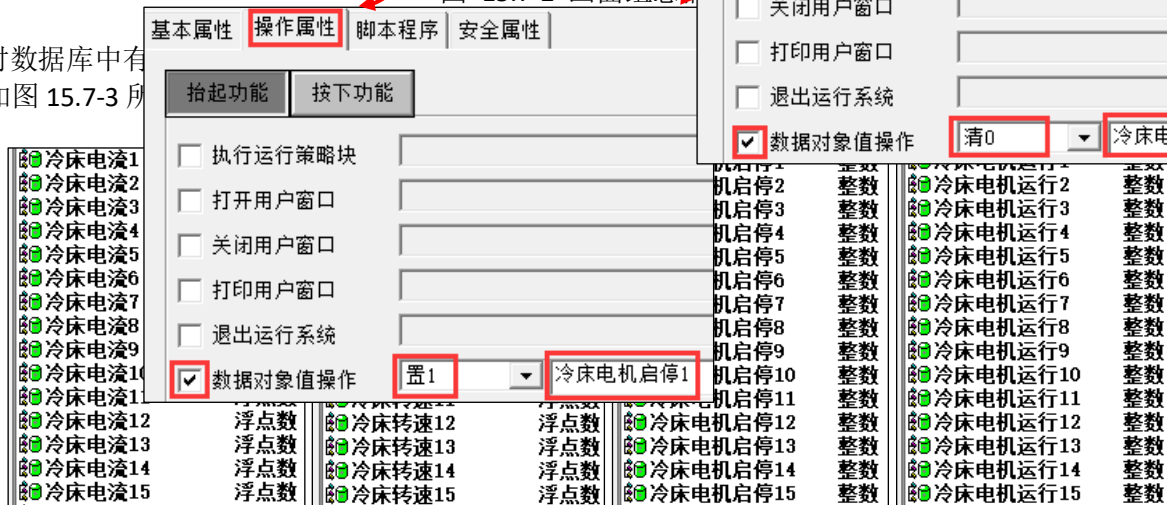


图 15.7-3 实时数据库情况

设备窗口中父设备为【通用 TCP/IP 父设备】，子设备为【西门子_Smart200】。

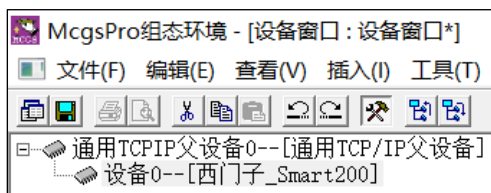


图 15.7-4 设备窗口情况

双击子设备，进入【设备编辑窗口】。西门子_SMART200 的通道【I000.0~I001.6】连接变量【冷床电机启停 1~15】；通道【Q000.0~Q001.6】连接变量【冷床电机运行 1~15】；通道【VDF000~VDF056】连接【冷床电流 1~15】；通道【VDF060~VDF116】连接【冷床转速 1~15】。如图 15.7-5 所示。

索引	连接变量	通道名称	索引	连接变量	通道名称
0001	冷床电机启停1	读写I000.0	0031	冷床电流1	读写VDF000
0002	冷床电机启停2	读写I000.1	0032	冷床电流2	读写VDF004
0003	冷床电机启停3	读写I000.2	0033	冷床电流3	读写VDF008
0004	冷床电机启停4	读写I000.3	0034	冷床电流4	读写VDF012
0005	冷床电机启停5	读写I000.4	0035	冷床电流5	读写VDF016
0006	冷床电机启停6	读写I000.5	0036	冷床电流6	读写VDF020
0007	冷床电机启停7	读写I000.6	0037	冷床电流7	读写VDF024
0008	冷床电机启停8	读写I000.7	0038	冷床电流8	读写VDF028
0009	冷床电机启停9	读写I001.0	0039	冷床电流9	读写VDF032
0010	冷床电机启停10	读写I001.1	0040	冷床电流10	读写VDF036
0011	冷床电机启停11	读写I001.2	0041	冷床电流11	读写VDF040
0012	冷床电机启停12	读写I001.3	0042	冷床电流12	读写VDF044
0013	冷床电机启停13	读写I001.4	0043	冷床电流13	读写VDF048
0014	冷床电机启停14	读写I001.5	0044	冷床电流14	读写VDF052
0015	冷床电机启停15	读写I001.6	0045	冷床电流15	读写VDF056
0016	冷床电机运行1	读写Q000.0	0046	冷床转速1	读写VDF060
0017	冷床电机运行2	读写Q000.1	0047	冷床转速2	读写VDF064
0018	冷床电机运行3	读写Q000.2	0048	冷床转速3	读写VDF068
0019	冷床电机运行4	读写Q000.3	0049	冷床转速4	读写VDF072
0020	冷床电机运行5	读写Q000.4	0050	冷床转速5	读写VDF076
0021	冷床电机运行6	读写Q000.5	0051	冷床转速6	读写VDF080
0022	冷床电机运行7	读写Q000.6	0052	冷床转速7	读写VDF084
0023	冷床电机运行8	读写Q000.7	0053	冷床转速8	读写VDF088
0024	冷床电机运行9	读写Q001.0	0054	冷床转速9	读写VDF092
0025	冷床电机运行10	读写Q001.1	0055	冷床转速10	读写VDF096
0026	冷床电机运行11	读写Q001.2	0056	冷床转速11	读写VDF100
0027	冷床电机运行12	读写Q001.3	0057	冷床转速12	读写VDF104
0028	冷床电机运行13	读写Q001.4	0058	冷床转速13	读写VDF108
0029	冷床电机运行14	读写Q001.5	0059	冷床转速14	读写VDF112
0030	冷床电机运行15	读写Q001.6	0060	冷床转速15	读写VDF116

图 15.7-5 通道和变量连接情况

15.7.2 西门子_SMART200 与三菱_FX3U 通道对应关系

替换外部设备时，需要理清原设备的通道和新设备的通道之间的对应关系。在本例中，在西门子_SMART200 中使用到的通道和三菱_FX3U 通道之间的对应关系如图 15.7-6 所示。

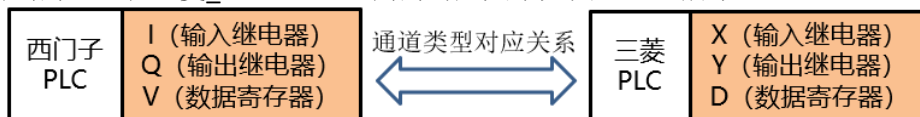


图 15.7-6 西门子 PLC 与三菱 PLC 通道类型对应关系

然后根据新设备的地址规划进行地址分配。本例中，我们将西门子_SMART200 的通道【I000.0~I001.6】替换成三菱_FX3U 的通道【X0100~X0116】；通道【Q000.0~Q001.6】替换成三菱_FX3U 的通道【Y0220~Y0236】；通道【VDF000~VDF056】替换成三菱_FX3U 的通道【DDF0560~DDF0588】；通道【VDF060~VDF116】替换成三菱_FX3U 的通道【DDF0590~DDF0618】。

15.7.3 通过设备通道的导入导出进行设备替换

- ① 进入西门子_SMART200 的【设备编辑】窗口，点击右侧的【设备信息导出】按钮，在弹出的【另存为】

对话框中，输入文件名【西门子_Smart200.csv】，默认保存类型为 CSV。选择存储路径为桌面，然后点击保存。系统弹出设备信息导出成功的提示窗口，点击确定关闭即可。

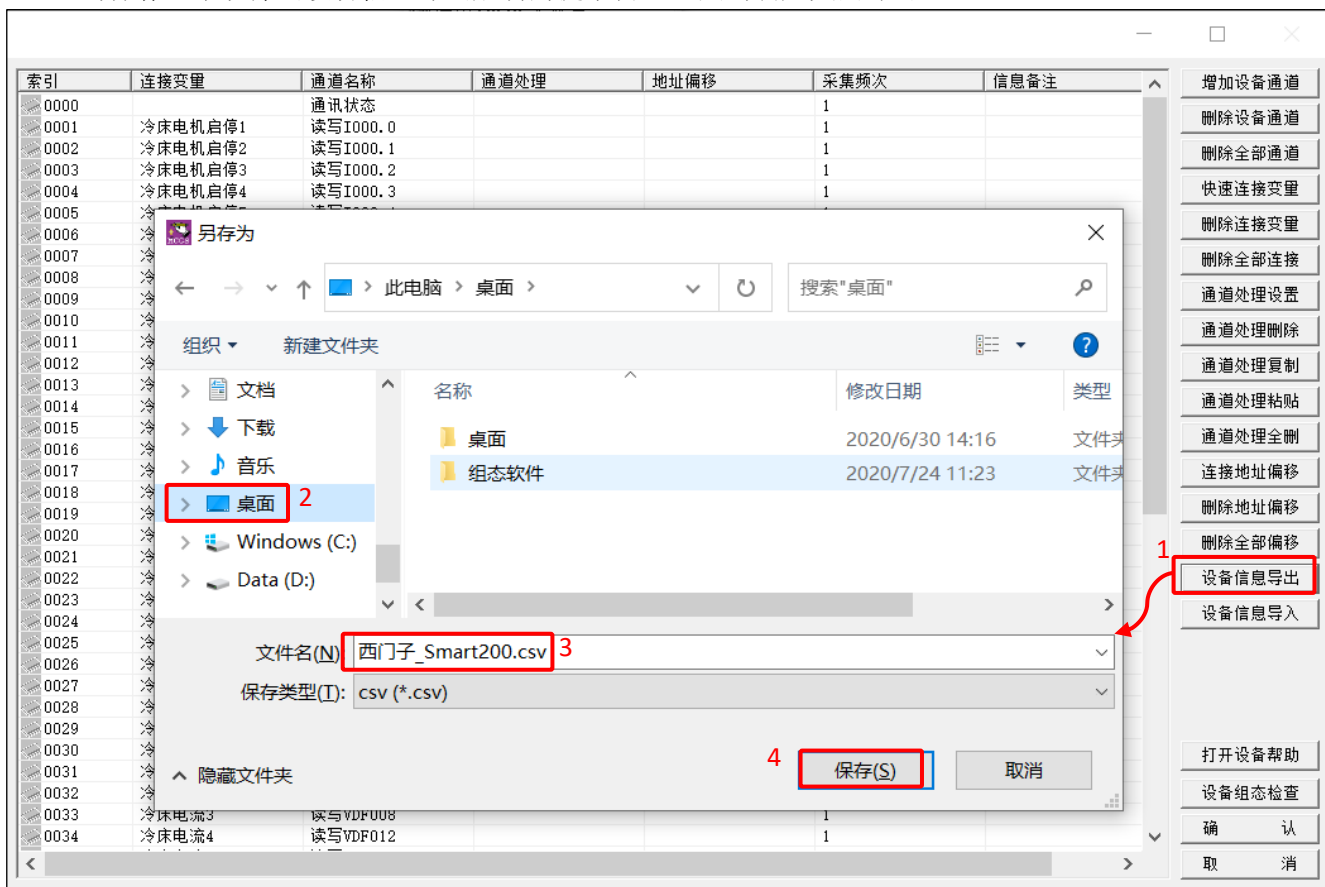


图 15.7-7 设备信息导出

- ② 在设备窗口中，新建一个【通用 TCP/IP 父设备】，在它的下方添加【FX3_ETHERNET】子设备，双击打开。然后添加设备通道【X0100~X0116】【Y0220~Y0236】和【DDF0560~DDF0618】。添加完成后，点击【设备信息导出】，导出文件名为【FX3_ETHERNET.csv】，保存路径为桌面。
- ③ 用 EXCEL 打开刚才导出的文件【西门子_Smart200.csv】和【FX3_ETHERNET.csv】，将【西门子_Smart200.csv】中变量名列的文字，复制到【FX3_ETHERNET.csv】中对应通道前的变量名列。复制完成后保存文件为 CSV 格式。

通道号	变量名	变量类型	通道名称
0	冷床电机启停1	INTEGER	读写I000.0
1	冷床电机启停2	INTEGER	读写I000.1
2	冷床电机启停3	INTEGER	读写I000.2
3	冷床电机启停4	INTEGER	读写I000.3
4	冷床电机启停5	INTEGER	读写I000.4
5	冷床电机启停6	INTEGER	读写I000.5
6	冷床电机启停7	INTEGER	读写I000.6
7	冷床电机启停8	INTEGER	读写I000.7
8	冷床电机启停9	INTEGER	读写I001.0
9	冷床电机启停10	INTEGER	读写I001.1
10	冷床电机启停11	INTEGER	读写I001.2
11	冷床电机启停12	INTEGER	读写I001.3
12	冷床电机启停13	INTEGER	读写I001.4
13	冷床电机启停14	INTEGER	读写I001.5
14	冷床电机启停15	INTEGER	读写I001.6

通道号	变量名	变量类型	通道名称
0		INTEGER	只读X0100
1		INTEGER	只读X0101
2		INTEGER	只读X0102
3		INTEGER	只读X0103
4		INTEGER	只读X0104
5		INTEGER	只读X0105
6		INTEGER	只读X0106
7		INTEGER	只读X0107
8		INTEGER	只读X0110
9		INTEGER	只读X0111
10		INTEGER	只读X0112
11		INTEGER	只读X0113
12		INTEGER	只读X0114
13		INTEGER	只读X0115
14		INTEGER	只读X0116

图 15.7-8 复制变量

- ④ 回到【FX3_ETHERNET】的【设备编辑窗口】，点击【设备信息导入】按钮，在桌面上选择保存好的【FX3_ETHERNET.csv】文件，点击【打开】按钮进行设备信息的导入。

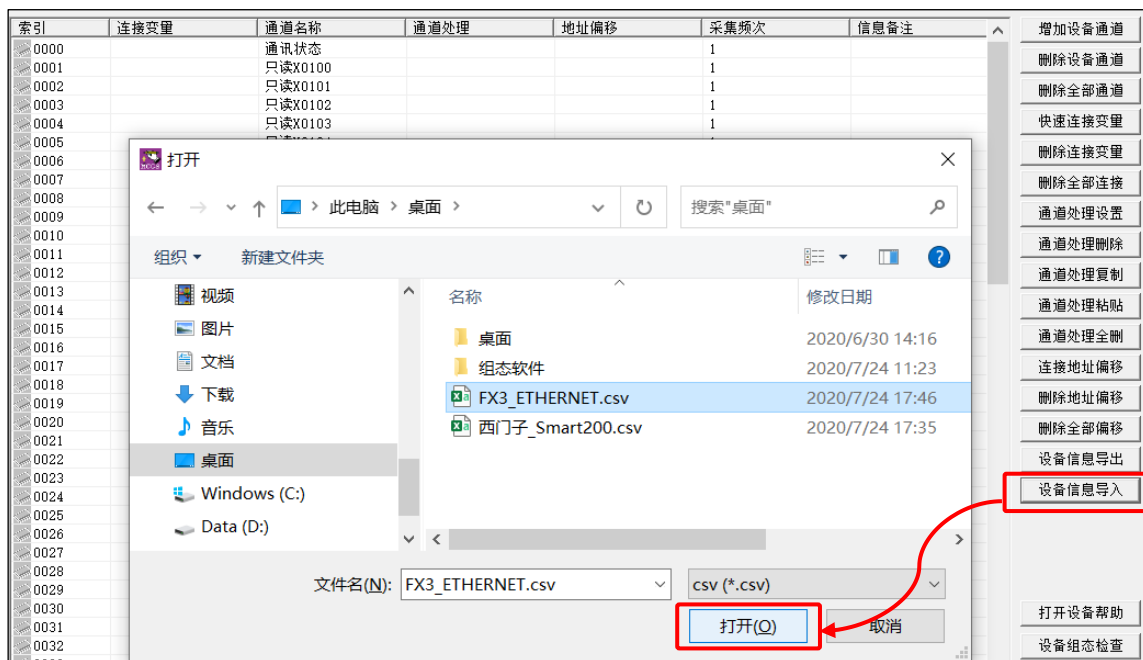


图 15.7-9 设备信息导入

- ⑤ 此时系统会弹出对话框提示【导入文件将自动删除设备中全部通道】询问【是否继续导入？】，选择【是】。

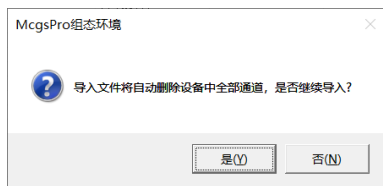


图 15.7-10 导入提示

- ⑥ 导入成功后，【FX3_ETHERNET】的【设备编辑窗口】如图 15.7-11 所示。

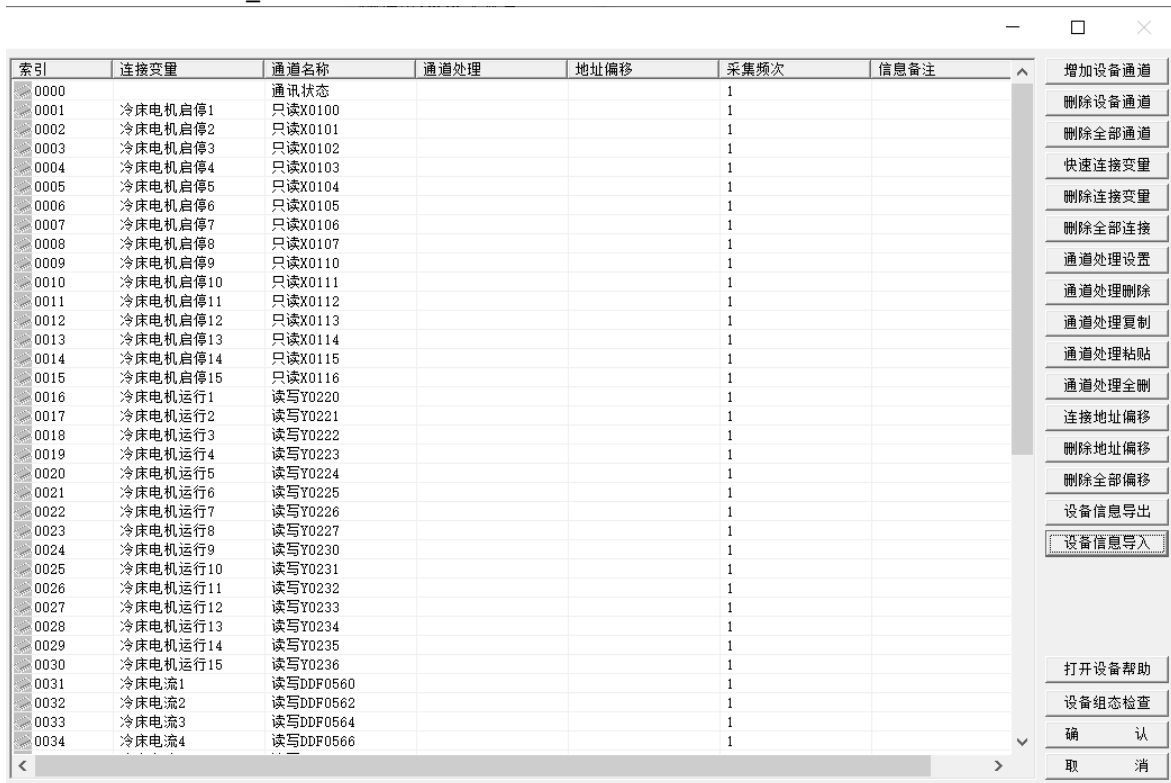


图 15.7-11 导入成功效果

- ⑦ 点击【确认】保存设置并关闭界面。
- ⑧ 双击【FX3_ETHERNET】的父设备，弹出【通用 TCP/IP 设备属性编辑】界面，将设备参数按 TPC 和当前外部设备的参数进行设置。



图 15.7-12 按新 PLC 设置新父设备参数

- ⑨ 添加完成新设备后，删除原设备驱动的父设备和子设备。

15.8 TPC 系统控制

本驱动构件用在工程运行过程中控制 TPC 的蜂鸣、背光及 IP 地址设置。

本驱动为独立设备，只对本机进行操作不涉及通信。

本节主要介绍 TPC 系统控制驱动的功能和使用方法。

15.8.1 TPC 系统控制功能介绍

首先添加一个 TPC 系统控制驱动。

- ① 进入工作台 → 设备窗口 → 工具箱 → 设备管理，在【设备管理】界面中左侧的【可选设备】目录树中找到【TPC 系统控制】。双击该驱动，将其添加到右侧的选定设备窗口中，然后点击确认保存设置并关闭窗口。

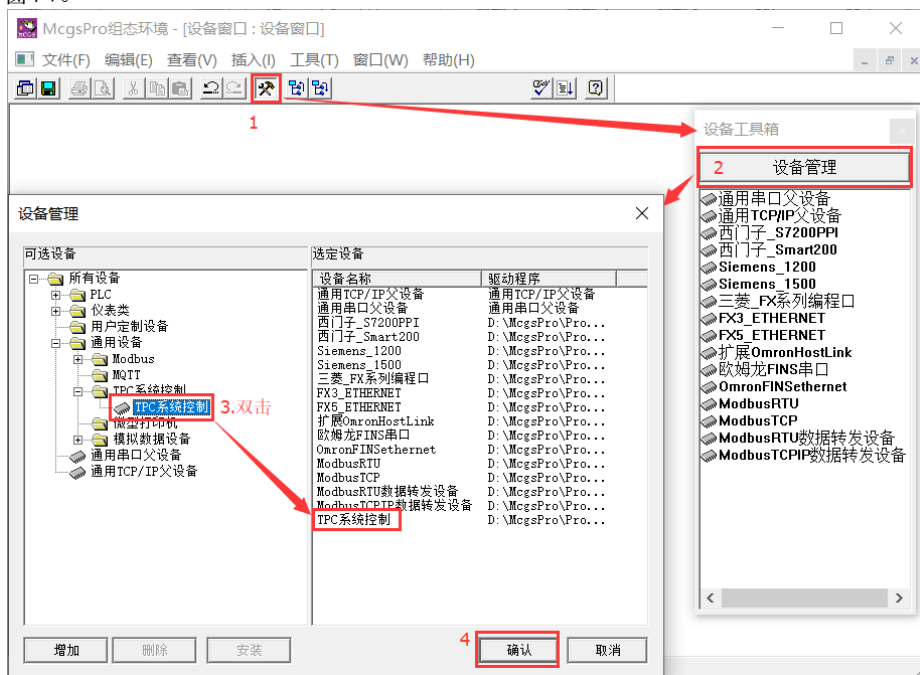


图 15.8-1 添加驱动

- ② TPC 系统控制驱动不需要挂载在父设备下，直接双击工具箱中的【TPC 系统控制】驱动，将它添加到设备窗口左侧的空白处。
- ③ 双击【TPC 系统控制】设备，打开【设备编辑窗口】。如图 15.8-2 所示。

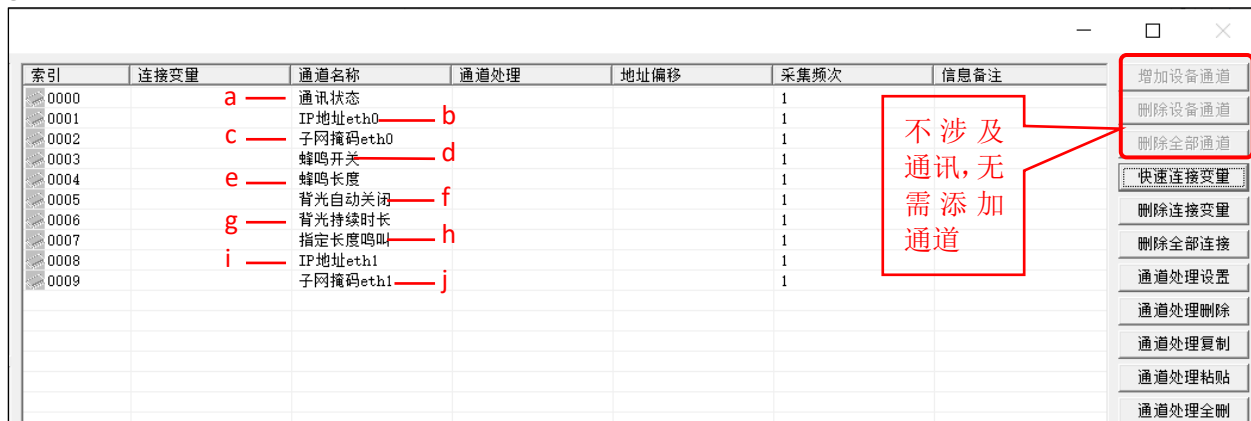


图 15.8-2 设备通道展示

TPC 系统控制设备编辑窗口配置信息：

- a. 通讯状态：为 0 表示当前驱动正常；为-1 表示驱动加载失败。
- b. IP 地址 eth0：设置、获取本机 IP 地址。
- c. 子网掩码 eth0：设置、获取本机子网掩码。
- d. 蜂鸣开关：蜂鸣器的开关，1 为开，0 为关。
- e. 蜂鸣长度：设置系统蜂鸣器鸣叫声音长短。重启后生效。
- f. 背光自动关闭：背光灯自动关闭是否生效，1 为可用，0 为不可用。
- g. 背光持续时长：背光灯在开启了自动关闭功能的状态下，点亮持续时长后熄灭。
- h. 指定长度鸣叫：产生指定长度的即时鸣叫，单位为 MS。
- i. IP 地址 eth1：针对双网口 TPC(单网口此通道无效)。
- j. 子网掩码 eth1：针对双网口 TPC(单网口此通道无效)。

注意：本驱动用于无网络机型时，IP 地址显示为空，其他功能可正常使用。

15.8.2 TPC 系统控制的设备命令

我们也可以通过设备命令对 TPC 进行设置。相关设备命令如下：

设备命令	命令介绍	
BACKLIGHTON	格式	BACKLIGHTON()
	用途	打开背光灯
	例	!SetDevice(设备 0, 6, "BACKLIGHTON()") 打开背光灯
BACKLIGHTOFF	格式	BACKLIGHTOFF()
	用途	关闭背光灯
	例	!SetDevice(设备 0, 6, "BACKLIGHTOFF()") 关闭背光灯
REFRESH	格式	REFRESH()
	用途	重新获取当前所有通道值
	例	!SetDevice(设备 0, 6, "REFRESH()") 获取一次当前所有通道值，对状态进行刷新。
REBOOT	格式	REBOOT()
	用途	重新启动操作系统
	例	!SetDevice(设备 0, 6, "REBOOT()") 重新启动操作系统

15.8.3 TPC 系统控制应用举例

本节介绍通过 TPC 系统控制驱动，设置单网口 TPC 的 IP 地址。

- ① 按照 15.8.1 中的步骤添加 TPC 系统控制子设备。双击子设备进入【设备编辑窗口】。在通道【IP 地址 eth0】左侧的连接变量处右击，弹出【变量选择】界面，在该界面的【选择变量】右侧的输入框中输入【IP 地址】，点击确认关闭窗口。

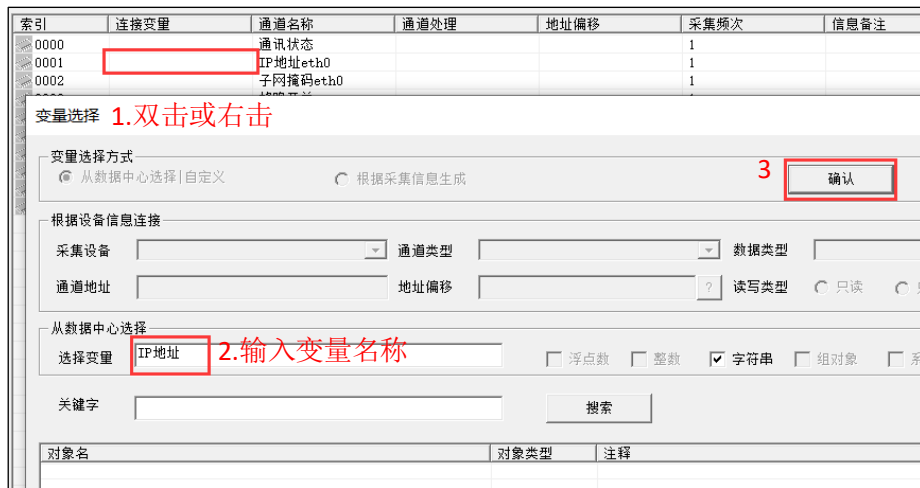


图 15.8-3 连接变量

- ② 点击【确认】按钮后弹出对话框提示变量未定义，询问是否添加，选择【添加】。

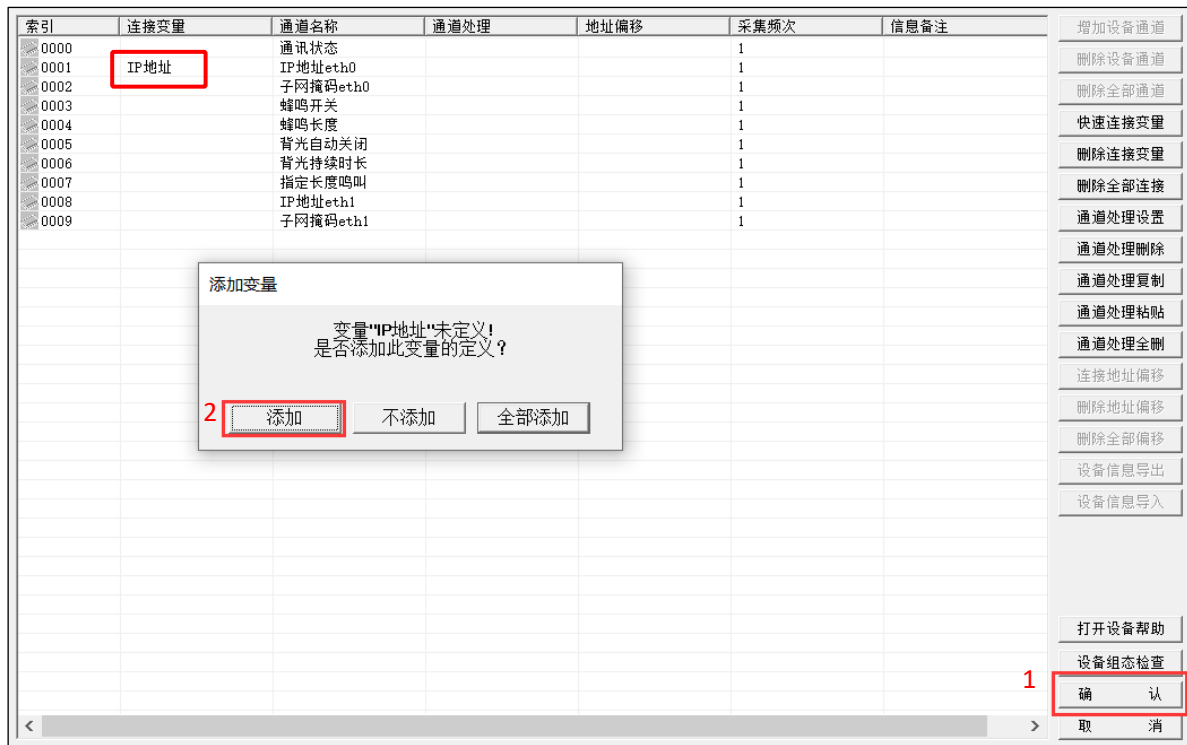


图 15.8-4 添加变量定义

此时，实时数据库中出现一个字符串型变量【IP 地址】。

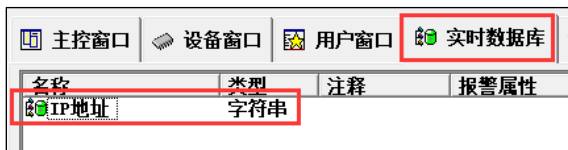


图 15.8-5 实时数据库中自动生成变量

- ③ 进入用户窗口，添加一个输入框构件，关联它操作属性中的数据对象为【IP 地址】。



图 15.8-6 组态画面

- ④ 保存工程后下载工程，然后启动运行。可以看到输入框中显示当前 IP 地址【200.200.200.190】，我们将其重新设置为【200.200.200.191】。

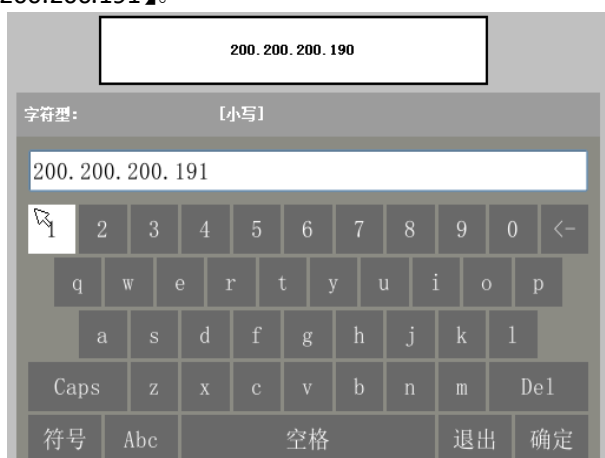


图 15.8-7 修改 IP 地址

- ⑤ 重启 TPC，在读条界面按住屏幕，进入【启动页面配置界面】，点击【系统参数设置】按钮，进入【网络】设置界面。可以看到 TPC 的 IP 地址变成了【200.200.200.191】。



图 15.8-8 IP 地址已修改

15.9 模拟设备

模拟设备可根据设置的参数产生一组模拟曲线的数据，以供用户调试工程使用。本设备可以产生标准的正弦波、方波、三角波、锯齿波信号，其幅值和周期都可以任意设置。本节主要介绍模拟设备的设置方法。

使用 15.5.1 中的方法在设备窗口中添加一个模拟设备。请注意：模拟设备也不需要挂载在父设备下。

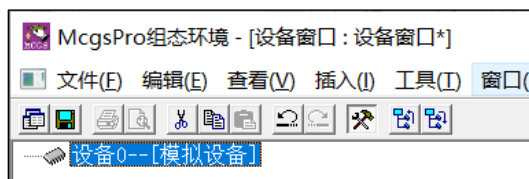


图 15.9-1 添加驱动

双击【模拟设备】弹出【设备编辑窗口】。

选择左下角的【内部属性】行，点击出现的...图标，弹出模拟设备的【内部属性】设置界面。

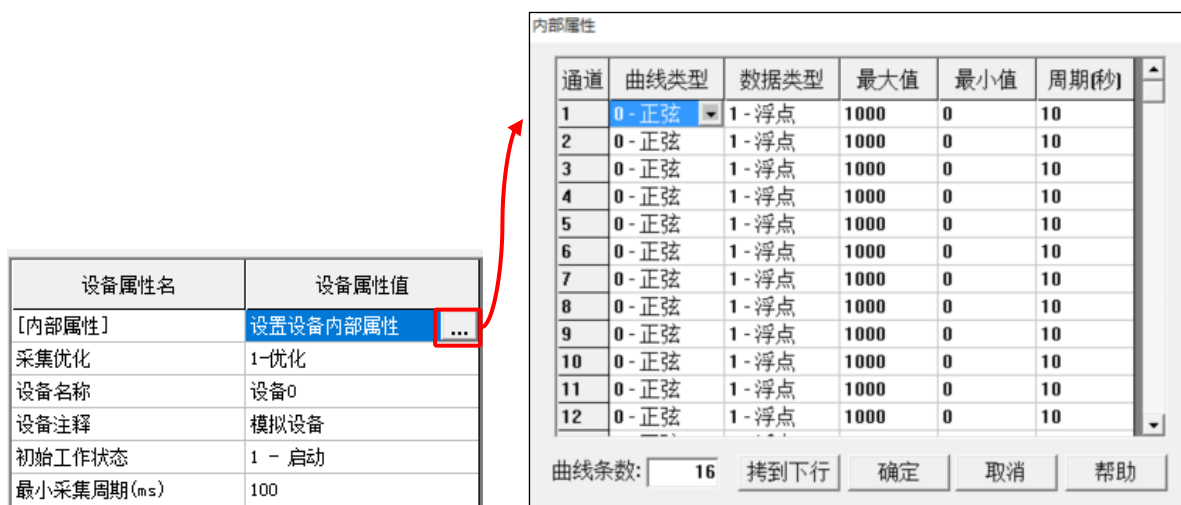


图 15.9-2 设置通道的曲线参数

在此处可以设置模拟设备的通道数量（即曲线条数），每条通道的曲线类型、数据类型、曲线的最大值、最小值和周期（单位为秒）。每个模拟设备可以产生多条曲线，每条曲线都可设置成不同的参数。本例保持默认不变。

具体操作如下

（1）用鼠标左键单击曲线类型下的某行，即可设置该行曲线的类型，单击数据类型下的某行，即可设置该行曲线的数据类型，可直接输入曲线的最大值，最小值，循环周期。

（2）在曲线条数输入框可以输入自己需要的曲线数。

（3）按拷到下行按钮，可以把鼠标所在行的所有数据拷贝到下一行。

（4）用鼠标选取多行，按 Ctrl+F5 可将所选取的最上一行的数据拷贝到选的其他各行。

15.10 屏与屏之间的数据同步

本节主要介绍使用 McgsPro 组态软件中的【ModbusTCP/IP 数据转发设备】、【ModbusTCP】两个驱动实现屏 TPC1 与屏 TPC2 之间的数据同步。

15.10.1 硬件连接

使用交换机和网线，将组态用的电脑和两个屏幕相连。

15.10.2 组态画面

使用 McgsPro 新建一个工程，将工程另存为【TPC1】。

在实时数据库中新建整数型变量【data0】。

进入用户窗口 0，新建一个按钮，将它的显示文本设置为【取反】，操作设置为点击后取反，关联的变量为【data0】。设置步骤如图 15.10-1 所示。

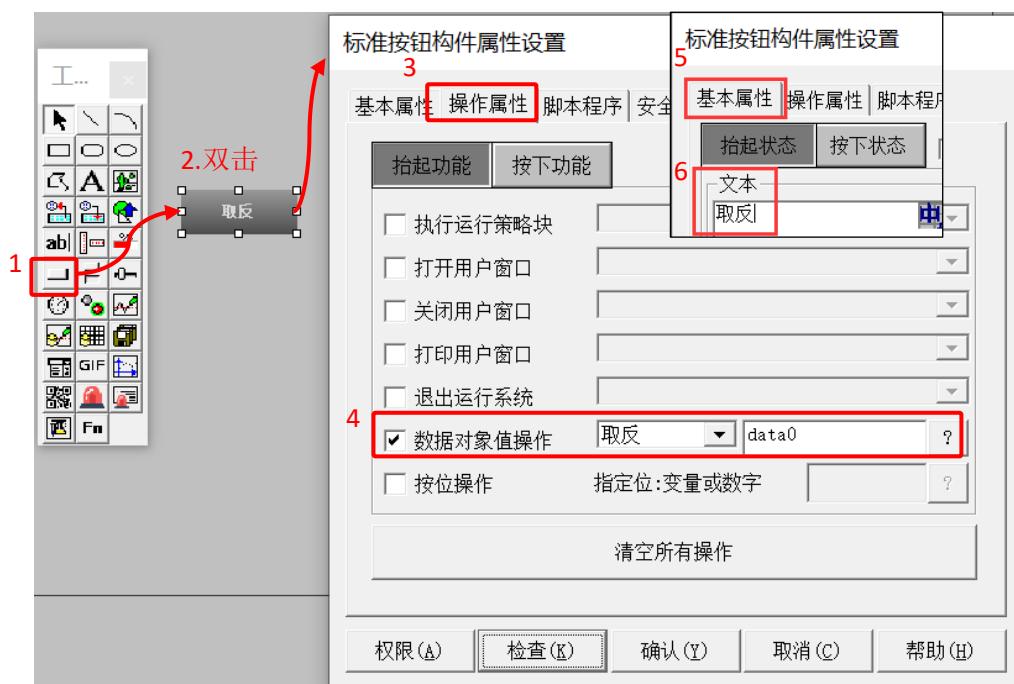


图 15.10-1 组态画面

在按钮左侧新建一个标签构件，将它的显示输出关联变量【data0】，当变量值为1时，显示信息为【开】，当变量值为0时，显示信息为【关】。设置步骤如图 15.10-2 所示。

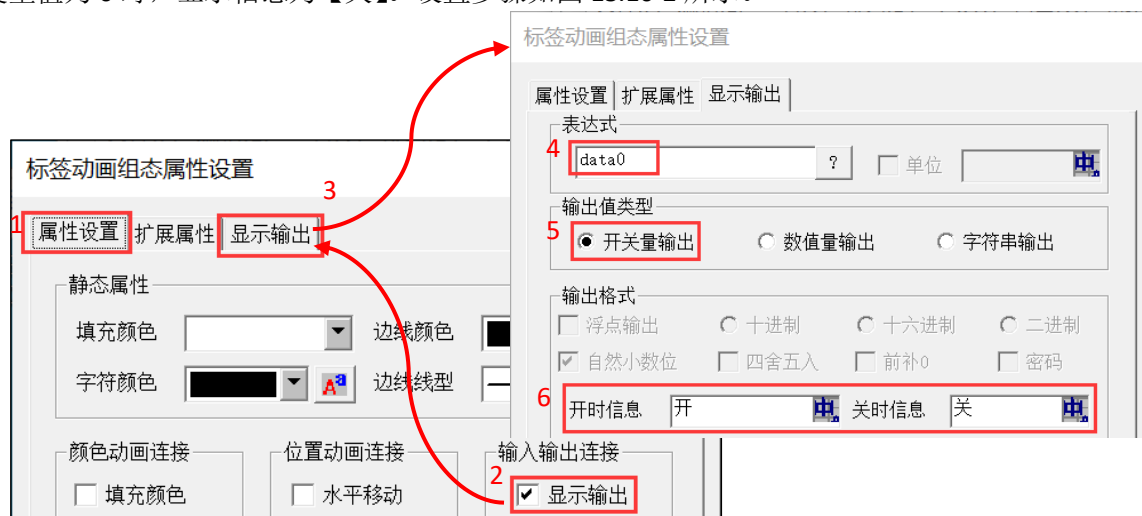


图 15.10-2 设置标签构件

再在实时数据库中新建一个整数型变量【通讯状态】。然后回到用户窗口0中，新建一个标签构件，按照关联【data0】的步骤将标签和变量【通讯状态】相关联，输出值类型设为【数值量输出】。

组态完画面后，保持工程，然后将工程另存为【TPC2】。

15.10.3 通讯设置

在本例中，TPC1 的 IP 地址为【200.200.200.190】，TPC2 的 IP 地址为【200.200.200.191】。

15.10.3.1 组态 TPC1 中的设备

- ① 用 McgsPro 打开工程【TPC1】，进入其设备窗口，添加【ModbusTCPIP 数据转发设备】驱动到设备窗口中，该驱动没有父设备。

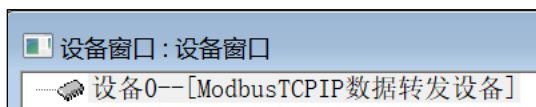


图 15.10-3 添加驱动

② 双击设备驱动进入【设备编辑窗口】。设备参数设置如图 15.10-4 所示。

设备属性名	设备属性值
采集优化	1-优化
设备名称	设备0
设备注释	ModbusTCP/IP数据转发设备
初始工作状态	1 - 启动
最小采集周期(ms)	1000
TCP/UDP	0 - TCP
IP 地址	0.0.0.0
端口号	502
客户端生存周期(s)	20
数据读取时间间隔(ms)	5
最大并行客户数	5
16位整数组码顺序	0 - 12
32位整数组码顺序	0 - 1234
32位浮点数组码顺序	0 - 1234
字符串字节序	1 - 21
字符串编码格式	0 - ASCII
设备站号	1

图 15.10-4 子设备属性

当该设备的 IP 地址为【0.0.0.0】时，表示绑定本机（TPC）的所有 IP 地址，即使本机的 IP 改变，工程 IP 也可不做修改。当然也可以直接将此处设置为 TPC1 的 IP 地址【200.200.200.190】。

③ 新建通道【读写 00001】，将它与变量【data0】连接。设置步骤如图 15.10-5 所示。



图 15.10-5 添加通道并连接变量

在通道【通讯状态】左侧的【连接变量】列右击或双击，弹出【变量选择】界面，双击变量【通讯状态】，如此，通道【通讯状态】便与变量【通讯状态】相连接。

组态完成后，保存工程，然后将该工程下载到 TPC1 中。下载配置界面设置如图 15.10-6 所示。其中，

【200.200.200.190】是 TPC1 的 IP 地址。下载成功后点击【启动运行】按钮使工程运行。

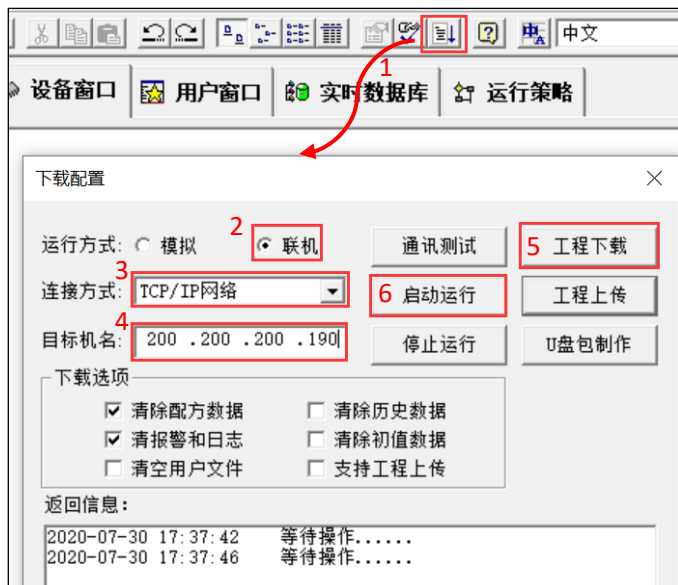


图 15.10-6 下载工程

此时由于 TPC2 还没有组态好，连接了【通讯状态】的标签显示为【-2】，表示【通讯端口打开失败】。通讯参数显示的数字所代表的意义，可以在设备的帮助中找到。

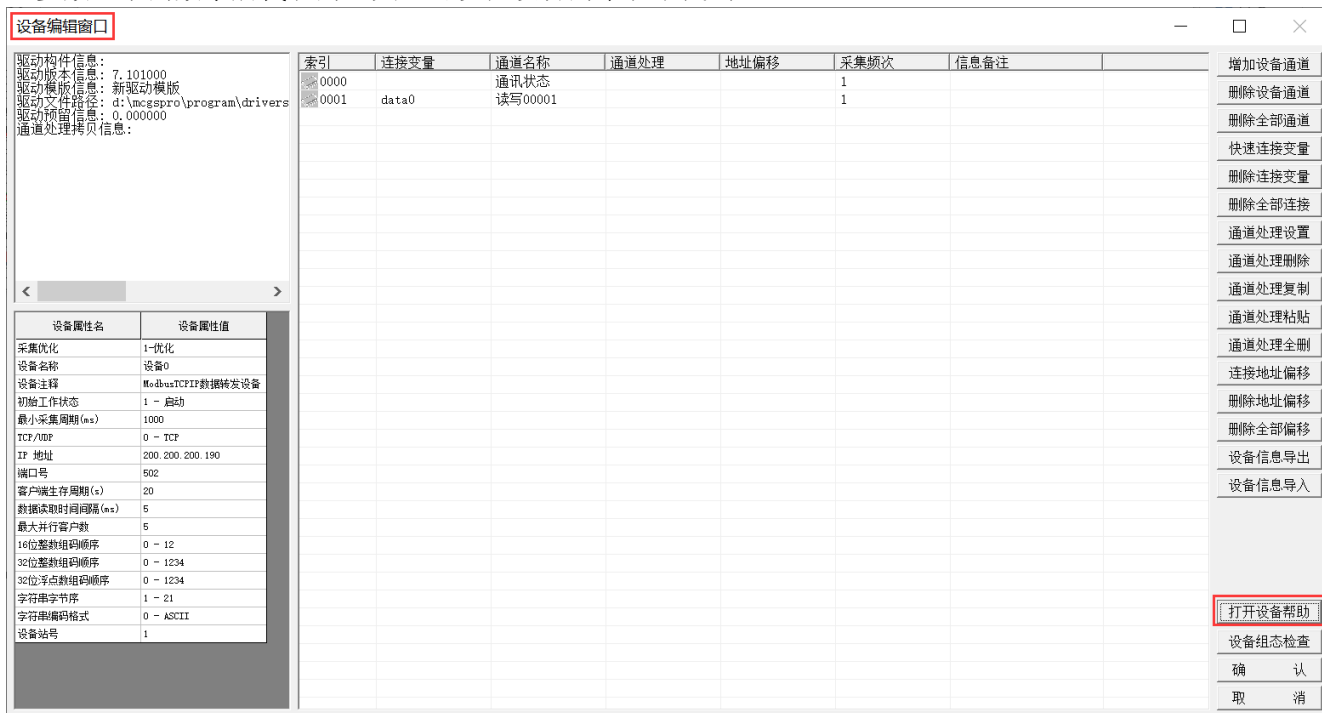


图 15.10-7 设备帮助的打开方法

15.10.3.2 组态 TPC2 中的设备

用 McgsPro 打开工程【TPC2】，进入其设备窗口，将【ModbusTCP】驱动挂载到【通用 TCP/IP 父设备】下。

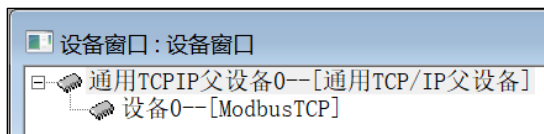


图 15.10-8 添加驱动

双击父设备，设置参数如图 15.10-9 所示。其中【200.200.200.191】是本机（TPC2）的 IP 地址，

【200.200.200.190】是远程设备（TPC1）的IP地址，

通用TCP/IP设备属性编辑

基本属性 | 设备测试

设备属性名	设备属性值
设备名称	通用TCP/IP父设备0
设备注释	通用TCP/IP父设备
初始工作状态	1 - 启动
最小采集周期(ms)	1000
网络类型	1 - TCP
服务器/客户设置	0 - 客户
本地IP地址	200.200.200.191
本地端口号	0
远程IP地址	200.200.200.190
远程端口号	502

检查(K) 确认(Y) 取消(C) 帮助(H)

图 15.10-9 设置父设备参数

双击【ModbusTCP】，进入子设备【设备编辑窗口】。子设备的参数设置如图 15.10-10 所示。

设备属性名	设备属性值
采集优化	1-优化
设备名称	设备0
设备注释	ModbusTCP
初始工作状态	1 - 启动
最小采集周期(ms)	100
设备地址	1
通讯等待时间	200
16位整数字节序	0 - 12
32位整数字节序	0 - 1234
32位浮点字节序	0 - 1234
字符串字节序	0 - 21
字符串编码格式	0 - ASCII
分块采集方式	0 - 按最大长度分块
0区写功能码	2 - 默认值
4区写功能码	2 - 默认值
1区最大块长	120
0区最大块长	120
3区最大块长	120
4区最大块长	120
是否校验事务ID	0 - 校验
是否校验数据长度	0 - 校验
64位浮点字节序	0 - 12345678

图 15.10-10 设置子设备参数

参考 15.3.3 中的步骤，将通道【通讯状态】与变量【通讯状态】连接，再新建一个和 TPC1 中同样地址的通道【读写 00001】，将它连接变量【data0】。设置完成后【设备编辑窗口】的通道和变量连接情况如图 15.10-11 所示。然后点击确认关闭界面。

索引	连接变量	通道名称
0000	通讯状态	通讯状态
0001	data0	读写00001

图 15.10-11 通道和变量连接情况

组态完成后，保存工程，然后将工程下载到 TPC2，工程下载成功后点击启动运行。可以看到，此时显示 TPC1 和 TPC2 通讯参数的标签都显示为【0】，表示通讯正常。

15.10.4 运行展示

在 TPC1 的画面中点击取反按钮，则按钮前的标签的显示内容从【关】变成【开】，且在 TPC2 画面中按钮前的标签显示内容也跟随变化，从【关】变成【开】。由于在通道设置时选择通道读写方式为【读写】，所以数据的转发是双向的，即在 TPC2 画面中操作时，TPC1 画面中的内容也跟随变化。

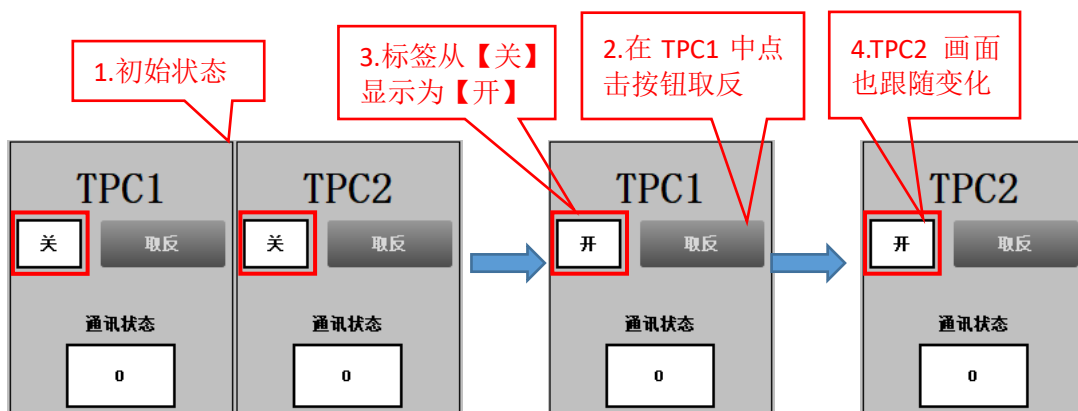


图 15.10-12 运行效果

说明：数据转发是通过相同的通道实现的数据转发，所以即使在两个工程中，通道【读写 00001】连接了不同名称的变量，也可以实现数据同步。比如在本例的两个工程中，通道【00001】分别连接了变量【A】和变量【B】，那么在 TPC1 中的变量【A】发生变化后，TPC2 中的变量【B】也会随之变化，反之亦然。

15.11 PLC 通过 TPC1 同 TPC2 传输数据

本节主要讲解内容是：PLC 不直接与 TPC2 相连，而是通过 TPC1 的数据转发功能，将 PLC 的数据传递给 TPC2。本例将在 15.10 样例的基础上进行组态，使用的 PLC 是 Siemens_1200，其 IP 地址为【200.200.200.200】。

15.11.1 硬件连接

使用交换机和网线，将组态用的电脑和两个屏幕以及 PLC 相连。

15.11.2 组态设备

使用 McgsPro 进入工程【TPC1】，在它的设备窗口中将【Siemens_1200】驱动挂载到【通用 TCP/IP 父设备】下，如图 15.11-1 所示。

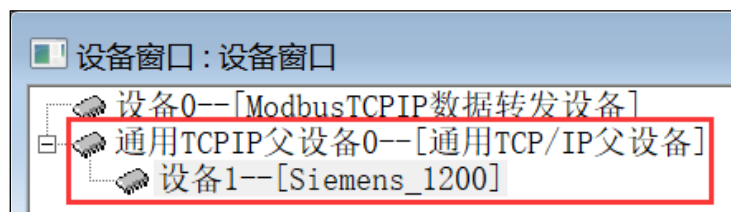


图 15.11-1 添加驱动

双击【通用 TCP/IP 父设备】，父设备参数设置如图 15.11-2 所示。其中【200.200.200.190】为 TPC1 的 IP 地址，【200.200.200.200】为 PLC 的 IP 地址。



图 15.11-2 设置父设备参数

双击【Siemens_1200】进入子设备的【设备编辑窗口】，子设备参数如图 15.11-3 所示。

设备属性名	设备属性值
[标签导入]	设置设备内部属性
采集优化	1-优化
设备名称	设备1
设备注释	Siemens_1200
初始工作状态	1 - 启动
最小采集周期(ms)	100
TCP/IP通讯延时	200
重建TCP/IP连接等待时间[s]	10
机架号[Rack]	0
槽号[Slot]	1
字符串编码	0 - ASCII
字符串解码顺序	0 - 21

图 15.11-3 设置子设备参数

在【设备编辑窗口】中添加通道【读写 Q000.0】并将它连接变量【data0】,设置步骤如图 15.11-4 所示。

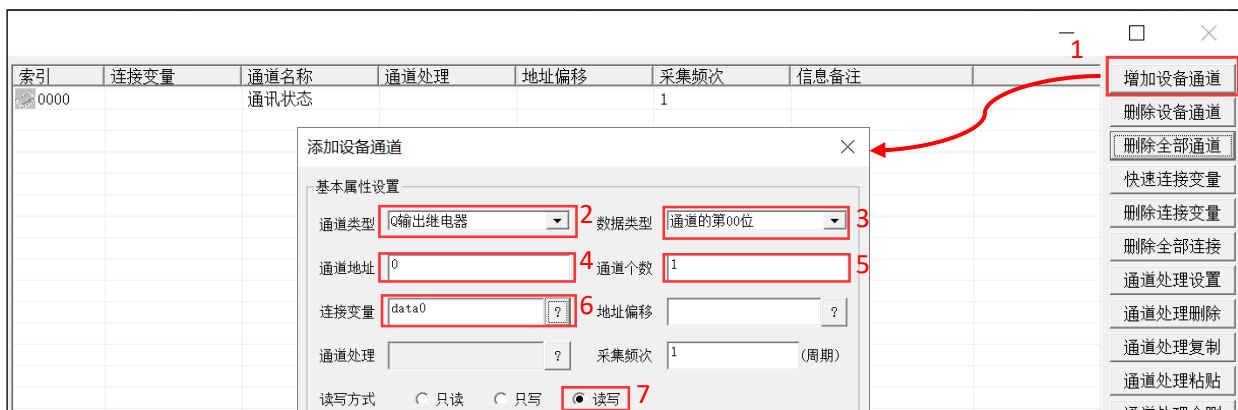


图 15.11-4 添加通道并连接变量

将通道【通讯状态】连接变量【PLC_通讯状态】。

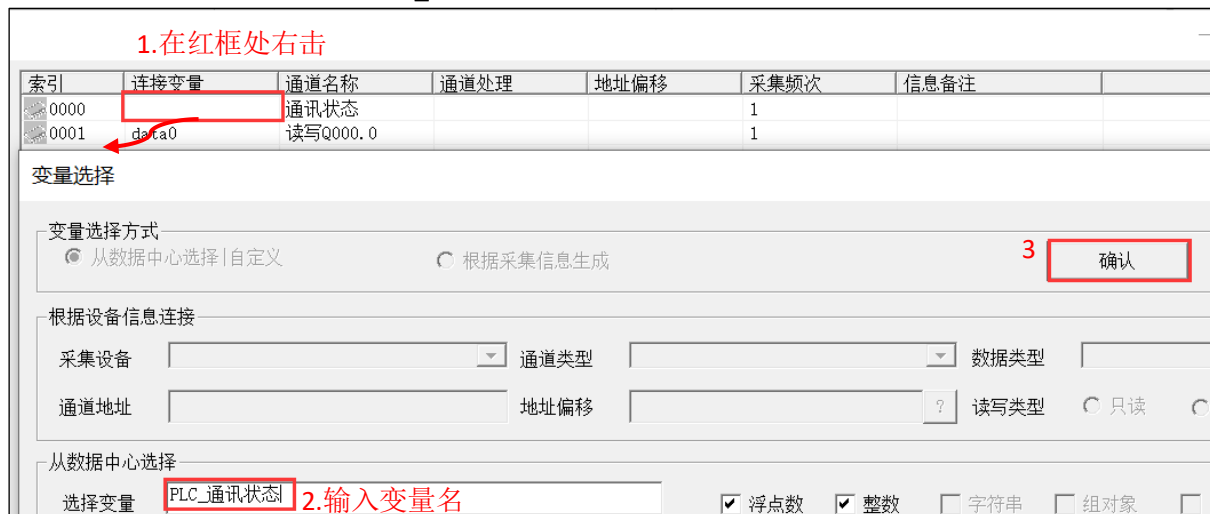


图 15.11-5 给设备通讯状态连接变量

连接完成后，通道与变量的连接情况如图 15.11-6 所示。

索引	连接变量	通道名称
0000	PLC_通讯状态	通讯状态
0001	data0	读写Q000.0

图 15.11-6 通道与变量的连接情况

点击【设备编辑窗口】的【确认】按钮，弹出对话框提示【变量“PLC_通讯状态”未定义!】，询问【是否添加此变量的定义?】，选择【添加】。

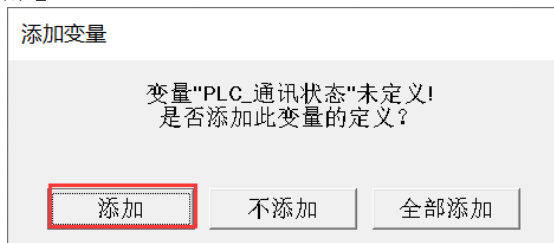


图 15.11-7 自动添加变量定义

此时实时数据库中自动添加一个整数型变量【PLC_通讯状态】。

名称	类型	注释	报警属性
通讯状态	整数		
data0	整数		
PLC_通讯状态	整数		

图 15.11-8 实时数据库情况

15.11.3 通讯设置

然后在用户窗口 0 中，新建一个标签，将它的显示输出关联新增加的变量【PLC_通讯状态】，输出值类型设为【数值量输出】。

组态完成后下载工程到 TPC1，然后启动工程。TPC2 的工程不做修改。

15.11.4 组态 PLC

使用 TIA Portal V15.1 设置 PLC 的 IP 地址。设置步骤如图 15.11-9 所示。



图 15.11-9 设置 PLC 参数一

勾选【属性】->【常规】->【防护与安全】->【连接机制】->【允许来自远程对象的 PUT/GET 通信访问】。

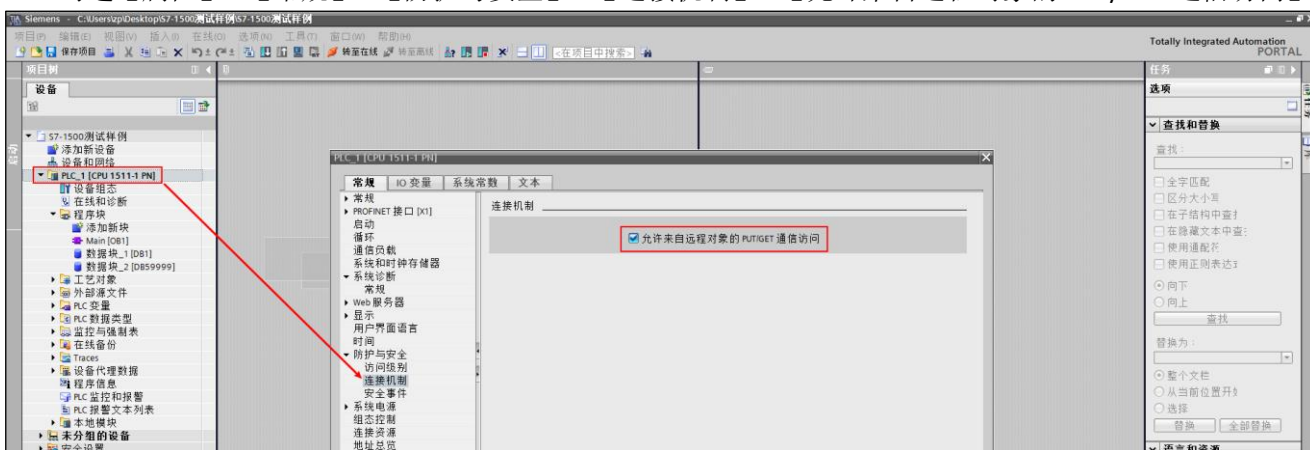


图 15.11-10 设置 PLC 参数二

设置完成后将程序下载到 PLC。

15.11.5 运行展示

在 TPC1 的画面中点击取反按钮，则按钮前的标签的显示内容从【关】变成【开】，此时 TPC2 画面中标签显示内容跟随变化，从【关】变成【开】，PLC 的输出点 Q0.0 的灯也随之变亮。由于在通道设置时选择通道读写方式为【读写】，所以 PLC 和两个 TPC 的数据可以相互传递，即其中一个设备的数据发生变化，另两个设备的数据也跟随变化。

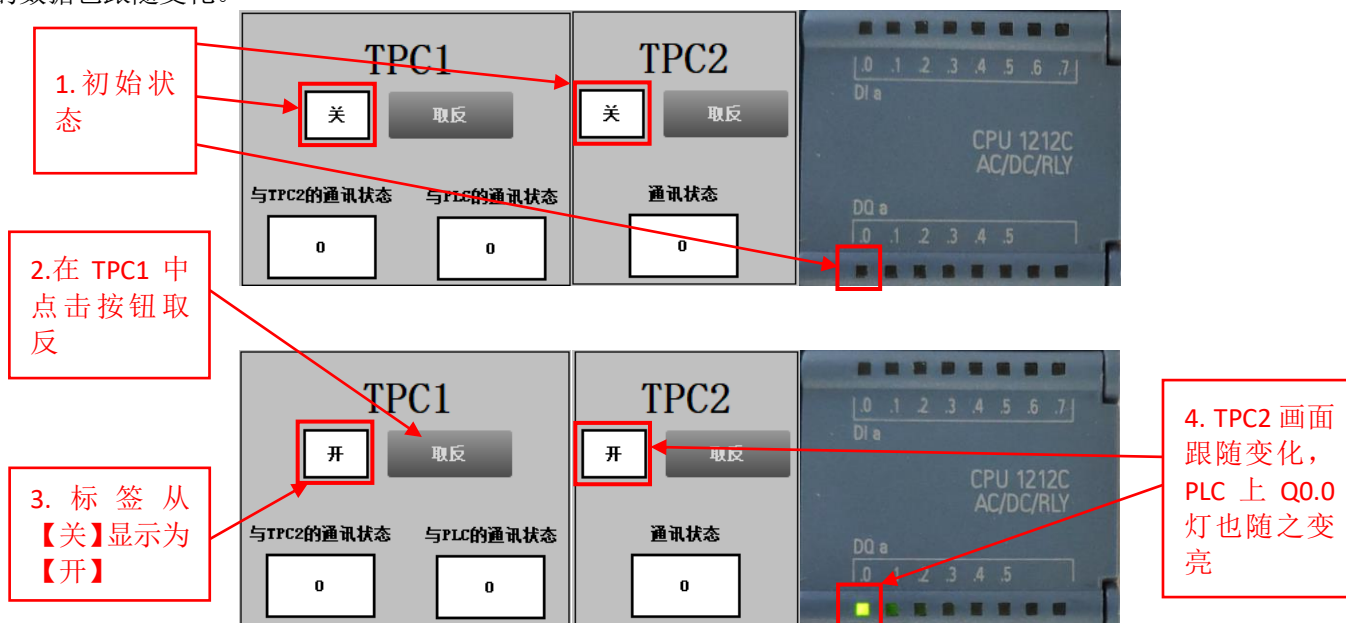


图 15.11-11 运行效果

15.12 通讯常见问题分析处理

15.12.1 通讯与否

当不确定通讯是否正常时候，建议都添加通讯状态来准确判断。

15.12.2 驱动最新

理论上通讯状态不为 0，都建议确认驱动是否为最新版本，可与 400 客服热线确认。

15.12.3 通讯状态

通讯状态不为 0，打开设备编辑窗口的设备帮助，里面有每个驱动通讯状态非 0 的排查措施。

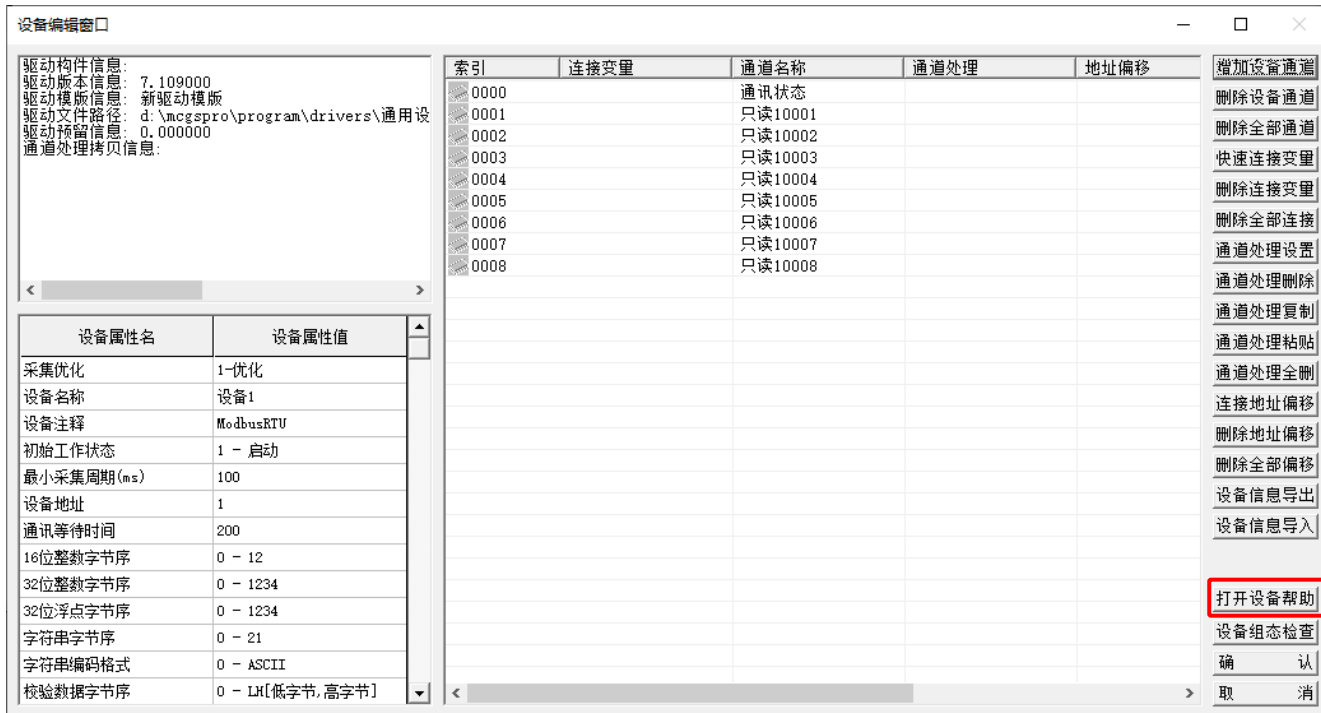


图 15.12-1 打开设备帮助

15.12.4 注意事项

- 绝大部分客户问题主要集中在 3 个，组态错误，参数硬件，地址超范围，故建议不好排查通讯问题时候，都按照对应通讯测试工程去测试通讯，通讯测试工程请联系 400 客服热线获取。
- 通讯测试工程包含 MCGS 工程，PLC 程序，接线帮助等，故用通讯测试工程完成通讯，基本可排除组态错误和参数硬件问题，剩下的主要地址超范围问题排查就有了方向。

※硬件篇※

第 1 章 TPC 产品介绍

昆仑通态生产的基于 Linux 的 TPC 产品均通过了 CE、FCC、RoSH 认证。本章节主要以 TPC1071Gi 为例，进行 TPC 产品的硬件介绍。其他型号的 TPC 会有外观材料、接口数量等方面的差异。详细硬件资料请参考昆仑通态选型手册（访问昆仑通态官网或致电 400 获取）。

- TPC 外观和接口
- TPC1071Gi 外观和接口如图 1.1-1 和图 1.1-2 所示。
- 正面



图 1.1-1 TPC 正面图

背面

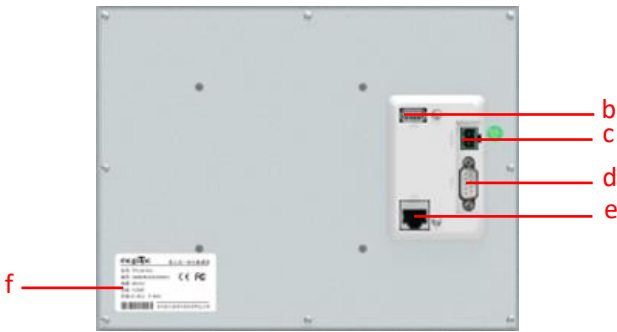
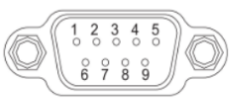


图 1.1-2 TPC 背面图

- a. 显示/触控区域； b.USB 主口（Type-A，作用：USB 打印，上传/下载工程，数据导入/导出）； c.电源 DC24V±20%；
- d .DB9 公头（作用：与下位机通讯）：（COM1、COM2、COM3）或（COM1、COM9），针脚定义如图 1.1-3 所示：



串口引脚定义

接口	PIN	引脚定义
COM1	2	RS232 RXD
	3	RS232 TXD
	5	GND
COM2	7	RS485 +
	8	RS485 -
COM3	4	RS485 +
	9	RS485 -
COM9 (RS422)	4	RXD +
	9	RXD -
	7	TXD +
	8	TXD -

RS422 由COM2和COM3模拟出。

注意：COM9不可与COM2或COM3同时使用。

图 1.1-3 TPC 引脚定义

- e .LAN 网口（作用：与下位机通讯，网络打印，上传/下载工程）； f .TPC 产品标签

第2章 TPC 系统功能

本章节主要介绍 TPC 的系统时间、IP 地址等系统功能设置，以及 TPC 常见异常处理。

2.1 TPC 系统功能

2.1.1 TPC 产品信息

进入 TPC 系统配置界面可显示 TPC 产品信息。

- TPC 系统配置界面进入方法：如图 2.1-1 所示，启动 TPC 时按住 TPC 液晶显示屏，进入 TPC 系统配置界面。



图 2.1-1 进入系统配置界面

- 系统配置界面如图 2.1-2 所示：

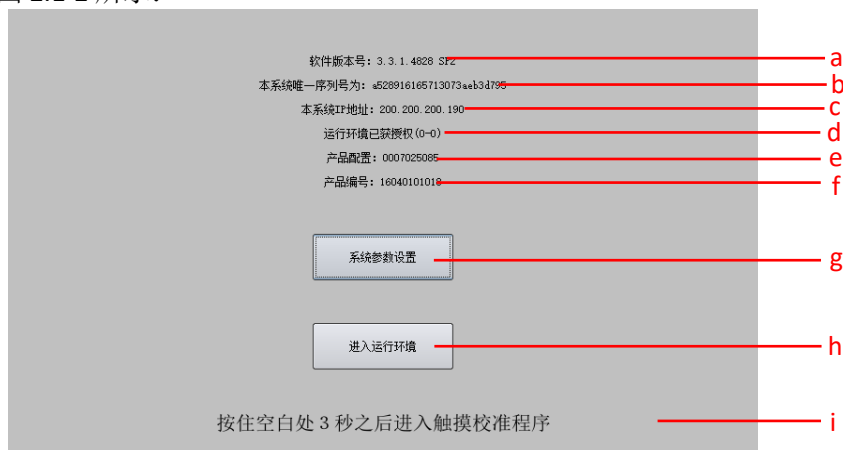


图 2.1-2 系统配置界面

a. TPC 运行环境版本 b. TPC 序列号 c. IP 地址（双网口 TPC 会显示 2 个 IP 地址） d. 授权状态
e.f. 产品配置编号 g. 系统参数设置 h. 运行 TPC 中的工程 i. 提示：按住页面空白处 3 秒，进行触摸校准

2.1.2 TPC 系统设置

通过 TPC 系统设置，可以查看 TPC 系统信息、设置系统时间、IP 地址等参数。

点击图 2.1-2 中的“g. 系统参数设置”进入 TPC 系统设置界面。如图 2.1-3 所示：



图 2.1-3 TPC 系统设置

- 系统：查看 TPC 系统 NK 版本，以及 TPC 磁盘、内存使用情况；

- 注意：系统版本可以使用 NK 升级包进行升级，请参考 TPC 辅助工具章节 1.3
- 背光：设置 TPC 液晶屏的背光时间。分辨率为 1920*1080 的 TPC，还支持背光亮度调节；
 - 蜂鸣：设置点击 TPC 液晶显示面板时，蜂鸣器鸣叫持续时间；
 - 触摸：选择是否显示鼠标指针，以及进行触摸校准；
 - 时间：设置 TPC 的系统时间；
 - USB：进行 USB 主从口切换，只有同时具备 USB 主口（Type-A）和 USB 从口（Type-B）的 TPC 才有该选项；
- 注意：TPC 重启后默认为主口模式，进行主从口切换时，主口和从口禁止插入任何设备，否则可能导致切换失败；
- 网络：设置 TPC 的 IP 地址、掩码、网关、DNS、启用/停用 DHCP。
- 注意：必须将 TPC 连上支持 DHCP 的路由器，才能启用 DHCP 模式，实现 TPC 自动获取 IP 地址；
- 单网口 TPC 默认 IP 地址：LAN 200.200.200.190；
- 双网口 TPC 默认 IP 地址：LAN1 200.200.200.190；LAN2 192.168.0.1。LAN1 与 LAN2 完全独立，互不干扰。

2.2 TPC 常见异常

① 修改 TPC 系统时间导致工程无法正常运行

现象：修改系统时间，弹出图 2.2-1 所示警告，【继续】修改后，工程无法正常运行；

原因：工程组态人员在当前工程中组态了运行期限功能，并开启了时间保护功能，用户无法随意修改系统时间；

处理：a.输入期限密码解锁所有工程期限；b.参考章节：[进阶篇 13.2.5 工程运行期限](#)，重新组态工程，取消时间保护功能，再将新工程下载到 TPC（[重新组态并下载工程会清空 TPC 中的已有工程](#)）。

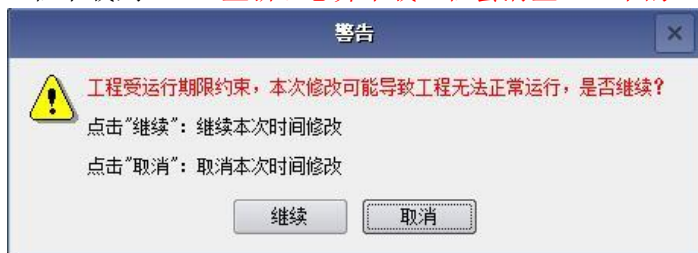
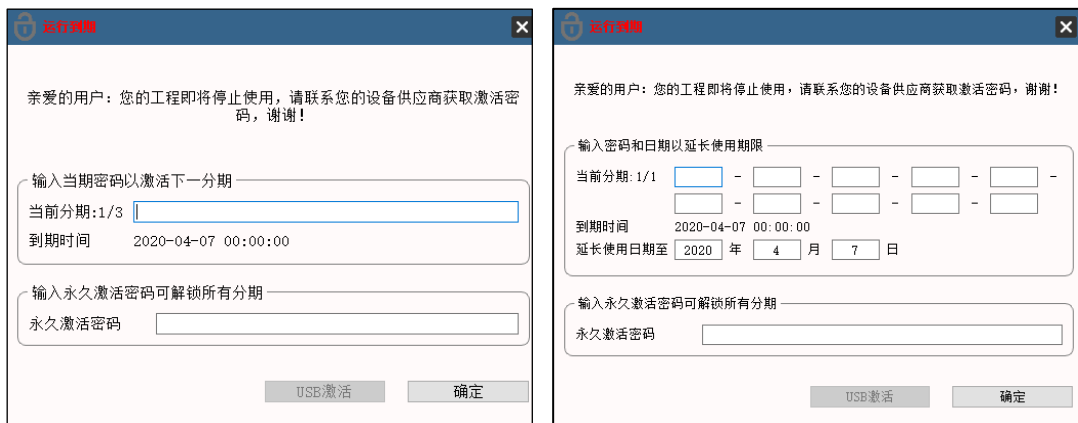


图 2.2-1 系统时间约束

② TPC 运行过程弹出【运行到期】提示框

现象：TPC 运行过程突然弹出图 2.2-2 所示的运行到期提示框，要求输入激活密码，否则工程无法正常使用；



静态期限弹窗

动态期限弹窗

图 2.2-2 工程运行到期弹窗

原因：工程组态人员在当前工程中组态了运行期限功能，导致工程运行到期后工程被锁定；

处理：a.输入当期激活密码或永久激活密码，解锁工程期限；b. 参考章节：[进阶篇 13.2.5 工程运行期限](#)，

重新组态工程，取消期限功能，再将新工程下载到 TPC（重新组态并下载工程会清空 TPC 中的已有工程）。

③ TPC 运行工程提示“用户工程文件已受保护”

现象：TPC 运行工程，弹出图 2.2-3 所示提示框“用户工程文件已受保护”，工程无法运行；

原因：工程组态人员在当前工程中启用了【工程文件保护】功能，而 TPC 却没有设置正确的触摸屏识别码；

处理：参考章节：[进阶篇 13.2.3 工程文件保护](#)，重新组态工程，关闭【工程文件保护】，重新下载工程（**重新组态并下载工程会清空 TPC 中的已有工程**）；。



图 2.2-3 工程文件保护弹窗

④ TPC 运行工程提示“系统时间异常”

现象：TPC 运行工程，如图 2.2-4 所示，弹窗提示“TPC 系统时间异常”，TPC 系统时间从 1970 年 0 时 0 分 0 秒重新计时。修改系统时间并重启 TPC，系统时间恢复成 1970 年 0 时 0 分 0 秒；

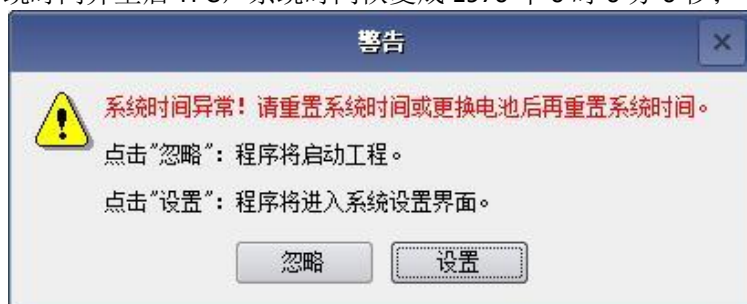


图 2.2-4 系统时间异常

原因：TPC 主板上的纽扣电池松动或电量耗尽；

处理：如图 2.4-5 所示，购买新 CR2032 纽扣电池，卸下 TPC 后盖进行替换，并固定好新电池。再重新进入 TPC 系统设置修改系统时间，系统时间修改方法参考[硬件篇章 2.1.2](#)。



图 2.4-5

注意：如果该处理办法无效，则可能是触摸屏故障，建议从购买渠道进行返修。

⑤ TPC 主从口切换无效

现象：显示 TPC 主从口切换成功，但实际上无法通过主口或从口连接 TPC；

原因：进行主从口切换时，主口或从口上插入了 U 盘或者从口线等设备；

处理：进行主从口切换时，主口和从口禁止插入任何设备，见[硬件篇章 2.1.2](#)。

⑥ DHCP 启用失败

现象：TPC 的 DHCP 功能启用失败，无法自动获取 IP 地址；

原因：TPC 网口未连接支持 DHCP 的路由器情况下，无法启用 DHCP 功能；

处理：TPC 网口连接支持 DHCP 的路由器后，再进入 TPC 系统设置启用 DHCP 功能，见[硬件篇章 2.1.2](#)。

⑦ TPC 密码遗失

现象：无法进行任何需要输入 TPC 密码的操作，比如更新工程、更新运行环境、更新系统 NK、进入 TPC 系统设置；

处理：TPC 密码遗失后无法找回，只能使用专用工具对 TPC 恢复出厂设置，该操作会清空 TPC 中包括用户工程在内的所有用户数据，该工具可致电 400 热线电话获取。

注意：此处的 TPC 密码指的是硬件篇章节 2.1.2讲解的【TPC 系统设置】→【密码】，不是工程密码或者工程用户登录密码。

※辅助工具※

第 1 章 TPC 辅助工具

用户可以使用昆仑通态提供的专用 U 盘包对 TPC 进行维护，主要功能包括：TPC 运行环境升级、TPC 系统 NK 升级、TPC 系统功能维护等。

除“运行环境升级包”、“U 盘功能包”之外，“NK 升级包”和“特殊功能包”需要向当地代理索取，各地代理联系方式可致电 400 获取。

1.1 运行环境升级包

TPC 的运行环境可以看作运行在 TPC 中 McgsPro 软件平台，运行环境升级指的就是升级运行在 TPC 中的 McgsPro 软件版本。用户可在重启 TPC 时，按照提示按住 TPC 液晶显示屏，进入系统配置界面查看 TPC 运行环境版本。

如图 1.1-1 所示，McgsPro 运行环境升级包一般跟随 McgsPro 组态软件一起发布。该升级包可强制将 TPC 运行环境升级到目标版本，无视 TPC 当前运行环境版本。

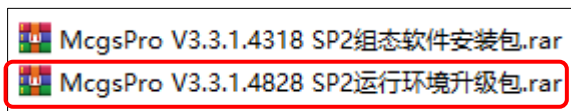


图 1.1-1 运行环境升级包

操作步骤：

- ① 解压图 1.1-1 所示的运行环境升级包，拷贝解压文件“tpcbakcup”到 U 盘；
- ② 将 U 盘插入 TPC，等待 TPC 弹出图 1.1-2 所示对话框，按照需要选择升级包语言（仅 3.3.1.4828 SP 及之后版本可选择语言），点击【是】进行运行环境升级，升级完成后拔出 U 盘，重启 TPC 即可。

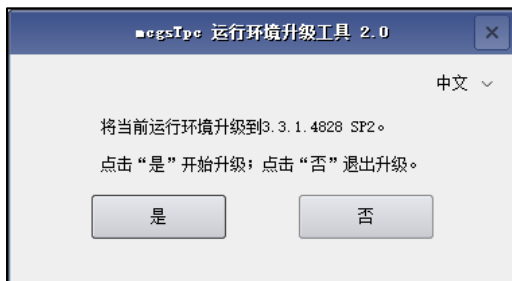


图 1.1-2 U 盘更新运行环境

注意：新版本运行环境可兼容旧版本工程；旧版本运行环境可能不兼容新版本工程。

1.2 U 盘功能包

此处的 U 盘功能包，指在电脑端运行 McgsPro 组态软件制作的 U 盘功能包，该功能包可用于更新 TPC 工程和运行环境。使用方法请参考[入门篇章节 2.5.2](#)。

1.3 NK 升级包

TPC 系统 NK：标准 TPC 产品自带 TPC 系统 NK 环境，可以简单的看作 TPC 的操作系统。

昆仑通态会根据需要更新迭代 TPC 的 NK 版本。对于有更新 TPC 系统 NK 版本需求的客户，可向代理索取 NK 升级包。

《U 盘升级包_G 系列_NK302.02_V1.0_20200420.rar》，该升级包可将 G 系列符合要求的 TPC 系统 NK 版本升级到 NK_302.02，我们以该升级包为例讲解，操作步骤如下：

- ① 解压“U 盘升级包_G 系列_NK302.02_V1.0_20200420.rar”，如图 1.3-1，先阅读“使用前必读.txt”，然后将“tpcbakcup”拷贝到 U 盘根目录；

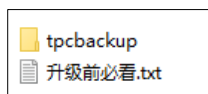


图 1.3-1

- ② 将 U 盘插入符合“使用前必读.txt”要求的 TPC，按照提示启动升级流程，NK 升级时 TPC 会多次重启，不要拔出 U 盘或断电；
- ③ 用户必须等 TPC 弹出图 1.3-2 所示对话框，才可拔出 U 盘，并重启 TPC。

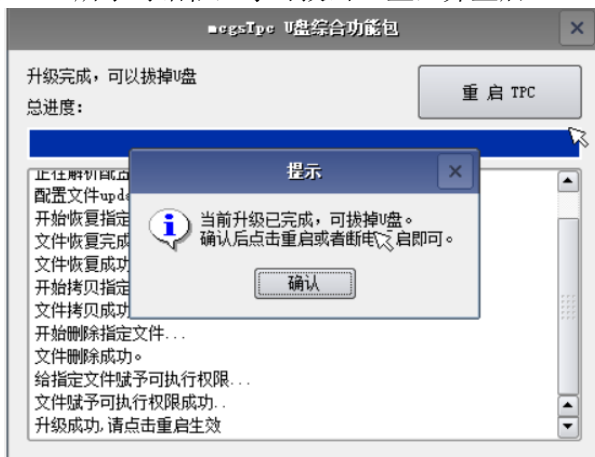


图 1.3-2 升级成功提示框

注意事项：

- U 盘必须采用 FAT32 格式，否则 TPC 无法识别 U 盘；
- NK 升级属于 TPC 系统升级，升级过程严禁拔出 U 盘或者给断电，否则可能导致 TPC 无法启动（如意外拔出了 U 盘或断电，建议不要修改 U 盘文件，直接插入 U 盘重启 TPC 继续升级流程，此为补救办法，不一定有效）；
- 标准版本 NK 升级会备份包括 TPC 工程在内的所有用户数据和用户文件，如用户 TPC 中保存的数据较多，将会大大增加“文件完整性校验”、“数据备份”和“数据恢复”耗时，请用户耐心等待。

1.4 特殊功能包

针对一些用户可能需要在 TPC 中实现某些特殊功能的需求，昆仑通态提供了对应的功能包供用户使用，如 VNC 功能包、MySQL 客户端部署包等。这些功能包需要向代理索取，且不具备普适性，对 TPC 型号有一定限制，用户在使用时需要先阅读图 1.3-1 所示的“使用前必读.txt”，确认目标 TPC 是否支持部署。特殊功能包使用方法与普通 U 盘功能包类似，用户参考 [TPC 辅助工具章节 1.3](#) 所述的 NK 升级包使用方法即可。

附录 1 McgsPro 支持的驱动

下表中列出了 McgsPro 当前版本支持的驱动。同时客户的非标设备、自研设备也可向代理申请驱动定制。

序号	驱动名称
1	Beckhoff_ADS
2	LG_MasterK 系列编程口
3	LG-MasterK 系列 Cnet 模块
4	ModbusAscii
5	ModbusRTU
6	ModbusRtuBaseZero
7	ModbusRTU 数据转发设备
8	ModbusTCP/IP 数据转发设备
9	ModbusTCP
10	MQTT
11	AB_DF1_ETHERNET
12	AB-半双工
13	AB-全双工
14	CompactLogixTCP
15	安川 MEMOBUS 扩展以太网
16	安川 MEMOBUS 扩展以太网_3000
17	安川 MP 控制器 MEMOBUS 以太网
18	岛电数字控制器
19	VIGOR_VS
20	VIGOR 系列 PLC
21	HaiWell_PLC
22	LE_ModbusRTU
23	和利时 PLC-ModbusRTU
24	横河西仪 FA-M3 TCP 以太网
25	汇川 AM600-ModbusTCP
26	H3U_MODBUSTCP
27	基恩士 KV5500_以太网
28	基恩士 PLC_KV1000
29	麦格米特 EC 系列 plc
30	OmronNj
31	欧姆龙 E5□N 系列仪表
32	扩展 OmronHostLink
33	欧姆龙 FINS 串口
34	欧姆龙 FINS 以太网
35	FX3_ETHERNET
36	FX5_ETHERNET
37	三菱_FX 系列串口
38	PLC_MELSEC_Q_TCPIP
39	三菱_Q 系列编程口
40	三菱_Q 系列串口
41	三菱_FX 系列编程口
42	松下 FP7 以太网

序号	驱动名称
43	松下 FP 系列通讯口
44	松下 FP 系列以太网
45	台达 dtle_as300TCP
46	台达 DVP 系列 PLC
47	微型打印机
48	西门子 CP443-1 以太网模块
49	西门子 S7_300_400MPI
50	西门子_S7200PPI
51	西门子_Smart200
52	西门子 S7200_CP243-1 以太网
53	Siemens_1200
54	Siemens_1500
55	XINJE_XL5E_ModbusTCP
56	信捷 XC 系列 PLC
57	信捷 ModbusRTU
58	永宏 PLC
59	宇光智能仪表

附录 2 样例工程

教程中使用到的样例工程清单列表如下，下载地址请拨打 400 热线电话获取。

序号	样例名称
1	入门篇：第 3 章_液位保持系统.MCP
2	进阶篇：2.3.2_自定义键盘.MCP
3	进阶篇：2.5_异常弹窗提示.MCP
4	进阶篇：3.1_旋转仪表.MCP
5	进阶篇：3.2.1_GIF.MCP
6	进阶篇：3.2.2-3.2.4_闪烁+可见度+跑马灯.MCP
7	进阶篇：3.3.1_动画显示.MCP
8	进阶篇：3.3.2_动画按钮.MCP
9	进阶篇：3.4_组合框构件_演示工程.MCP
10	进阶篇：3.5.4_GIF 构件方法.MCP
11	进阶篇：3.6.4_构件事件.MCP
12	进阶篇：5.1_实时趋势曲线.MCP
13	进阶篇：5.2_历史数据功能.MCP
14	进阶篇：第 6 章_多重复制.MCP
15	进阶篇：7.1_地址偏移.MCP
16	进阶篇：7.2_变量指针.MCP
17	进阶篇：8.1-8.2_配方构件+方法.MCP
18	进阶篇：8.3_配方函数.MCP
19	进阶篇：8.4_工艺顺控器应用示例.MCP
20	进阶篇：第 9 章_XY 曲线.MCP
21	进阶篇：10.1.2_声明脚本.MCP
22	进阶篇：10.2.2_传送带移动（IF）.MCP
23	进阶篇：11.10_内存操作函数.MCP
24	进阶篇：11.11_计时器函数.MCP
25	进阶篇：11.11_计时器函数 ALL.MCP
26	进阶篇：11.2.1_数组操作函数.MCP
27	进阶篇：11.2.1_设置数组大小，获取数组大小.MCP
28	进阶篇：11.2.2_运行策略.MCP
29	进阶篇：11.2.3_窗口操作函数.MCP
30	进阶篇：11.2.4_打印屏幕.MCP
31	进阶篇：11.2.4_系统操作函数.MCP
32	进阶篇：11.2.5_操作日志函数.MCP
33	进阶篇：11.3.1_数据设置函数+组对象操作函数.MCP
34	进阶篇：11.3.3_历史数据操作函数+保存初值.MCP
35	进阶篇：11.3.5_报警操作函数.MCP
36	进阶篇：11.5_时间函数.MCP
37	进阶篇：11.6_数学函数.MCP
38	进阶篇：11.7_字符串操作函数.MCP
39	进阶篇：11.8_系统变量函数.MCP
40	进阶篇：11.9_文件操作函数.MCP
41	进阶篇：12.1_多语言.MCP
42	进阶篇：第 12 章_多语言.MCP

43	进阶篇：13.1.2_用户权限（密码正确打开子窗口）.mcp
44	进阶篇：13.1.3_操作日志.MCP
45	进阶篇：第14章_打印功能应用实例.MCP
46	进阶篇：15.10_1_ModbusTCP/IP 转发附 PLC.mcp
47	进阶篇：15.10_2_ModbusTCP/IP 转发附 PLC.mcp
48	进阶篇：15.6_模拟通讯.mcp
49	进阶篇：15.7_替换 PLC.MCP
50	进阶篇：15.8_TPC 系统控制.MCP
51	进阶篇：15.9_模拟设备.MCP